

CISCO SYSTEMS



Cisco Press

Cisco 职业认证培训系列

CISCO CAREER CERTIFICATIONS

CCIE



CCIE 实验指南 (第1卷)

CCIE Practical Studies
Volume 1

网友dada147友情制作

Hands-on preparation for the CCIE Lab Exams

[美] Karl Solie, CCIE #4599 著
李津, CCIE #8794
卓林, CCIE #8867 译

人民邮电出版社

POSTS & TELECOMMUNICATIONS PRESS



CCIE 实验指南 (第1卷)

- 书中的5个实验场景模拟了CCIE考试中可能遇到的问题，完成这些实验有助于备考CCIE。
- 通过40多个LAN和WAN的实验，强化你的网络配置能力。
- 通过动手搭建模拟网络，加强CCIE考试实战能力。
- 通过练习VLAN、VTP、中继协议和生成树协议，训练配置Catalyst交换机的能力。
- 通过配置HDLC、PPP、Frame Relay、VoIP、VoFR、VoATM、ISDN和ATM，提高WAN技能。
- 深入理解并娴熟地配置几个重要的内部路由选择协议，如：RIP、IGRP、OSPF和EIGRP。
- 精通透明桥接、集成路由和桥接、源路由桥接、远程源路由桥接和DLSw+的配置技巧。

CCIE考试是思科认证考试中最难的，也是回报最高的。即使考生在专业方面非常出色，要获得这个认证也需要付出几年的精力和努力。本书既可以作为配置思科路由器的参考书籍，也可以用来准备CCIE考试，是获取CCIE认证最理想的资料。

本书从OSI参考模型的第一层往上，汇聚了所有组建复杂思科网络必需的硬件和软件部分。每一章涵盖了详细的技术或者协议，每个章节的后面包含了和考试类似的挑战实验，可以帮助你理解本章的知识点，同时也可以衡量CCIE考生掌握知识的程度。最后一章提供了5个CCIE的模拟实验。这些实验不仅测试你的知识水平，而且也考验你完成的速度，这是新的一天考试最需要注意的方面。在众多的准备CCIE考试的资料中，你将会发现本书是一本不可或缺的书。

网友dada147友情制作

ISBN 7-115-10872-2



9 787115 108722 >

CCIE

ciscopress.com

ISBN7-115-10872-2/TP·3191

定价:128.00元

人民邮电出版社
http://www.ptpress.com.cn

Cisco职业认证培训系列

CCIE实验指南 (第1卷)

[美]Karl Solie, CCIE#4599 著

李津, CCIE#8794

译

卓林, CCEI#8867

网友dada147友情制作

人 民 邮 电 出 版 社

69001500

图书在版编目 (CIP) 数据

CCIE 实验指南. 第1卷/ (美) 索利 (Solie, K.) 著: 李津, 卓林译.

—北京: 人民邮电出版社, 2002.12

ISBN 7-115-10872-2

I. C... II. ①索...②李... ③卓... III. 计算机网络—路由选择—水平考试—自学参考资料 IV. TP393

中国版本图书馆 CIP 数据核字 (2002) 第 084974 号

版权声明

Karl Solie: CCIE Practical Studies, Volume I (ISBN:1587200023)

Copyright © 2002 by Cisco Systems, Inc.

Authorized translation from the English language edition published by Cisco Press.

All rights reserved.

本书中文简体字版由美国 Cisco Press 授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

Cisco 职业认证培训系列

CCIE 实验指南 (第1卷)

- ◆ 著 [美] Karl Solie, CCIE # 4599
- 译 李 津, CCIE # 8794 卓 林, CCIE # 8867
- 责任编辑 李 际

- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
- 邮编 100061 电子函件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 读者热线 010-67132705
- 北京汉魂图文设计有限公司制作
- 北京顺义振华印刷厂印刷
- 新华书店总店北京发行所经销

- ◆ 开本: 787×1092 1/16
- 印张: 63.25
- 字数: 1 538 千字 2002 年 12 月第 1 版
- 印数: 1-3 000 册 2002 年 12 月北京第 1 次印刷

著作权合同登记 图字: 01-2001-4081 号

ISBN 7-115-10872-2/TP·3191

定价: 128.00 元

内 容 提 要

本书是第一本由 Cisco Systems 授权讲述 CCIE 实验考试的书籍。

本书主要目的是用来帮助那些 CCIE 备选者们通过完成许多实验来准备 CCIE 实验考试。这些实验是用来测试应试者在参考答案之前解决实际问题的能力。这些实验里的许多内容都是非常有难度的，如果你没有在实际环境中完成这些实验，那么在真正的 CCIE 考试中就有可能漏掉许多细节，从而导致整个考试的失败。本书结构遵从 OSI 模型及网络自低向高的构建方式。第 1 部分主要讲述物理层和路由器基本的设置和访问知识。第 2 部分“LAN 模型的建立”和第 3 部分“采用广域网互连局域网”关注数据链路层，而第 4 部分“路由选择协议”则关注第 3 层和第 4 层的内容。第 5 部分、第 6 部分和第 7 部分则分别介绍不可路由协议的传输、网络控制与网络访问、增强型网络协议。每一部分后面都有一个或者多个复杂的实验。第 8 部分讨论了现在的 CCIE 实验认证考试，提供了学习要点和建议以帮助应试者准备 CCIE 实验考试，这一部分还包括 5 套 CCIE 考试模拟题，以便给应试者一种身临其境的考试感觉。

本书主要对象是至少已经获得了 CCNA 或者 CCDA 证书的网络工程师。这些工程师们在阅读本书之前都已经对 IP 地址以及子网等知识有了一个比较深入的理解。而拥有 Cisco 路由器基础认识和基本路由器配置经验对于本书的阅读也是非常有帮助的。

关于作者

Karl Solie, CCIE #4599, 是 Comdisco 公司的首席网络工程师。Karl 在为 McDonnell、Douglas、Unisys 和 Comdisco 等公司设计和实施基于 LAN 和 WAN 的各种内部和外部网络方面积累了超过 13 年的丰富经验。在这 13 年里, Karl 设计和实施过各种类型的互连网络, 包括一些美国的大型商用或政府的 IP 或 SNA 网络。他曾经完成的网络项目范围从他所在威斯康星的哈得逊社区学校的 LAN/WAN 的设计与安装, 到亚特兰大、乔治亚以及加利福尼亚州的洛杉矶等地的政府网络工程。Karl 在加利福尼亚大学获得了法律专业的文学学士学位, 并且还在威斯康星一斯托德大学系统地学习了数学理论。

合作作者

Dan Keller, CCIE #6489, 目前在 Qwest Communications 公司担任高级网络架构工程师，主要从事 WAN 的设计。Dan 居住在加利福尼亚州的 Huntington Beach, 他完成了本书的第 7 章“WAN 协议与技术：综合业务数字网(ISDN)”。

Galina Diker Pildush, CCIE #3176, CCSI, 是 Advanced Communications Experts (ACE)公司的总裁及高级顾问，并且她还在 Cisco 全球最大的 CLP (Cisco Learning Partner, 认证和培训合作伙伴) Global Knowledge Network 公司任教，教授和设计 Cisco 网络培训课程。Galina 还是 Cisco Press 出版的《Cisco ATM Solutions》一书的作者，在本书中，她完成了第 8 章“WAN 协议与技术：异步传输模式 (ATM)”，现在她住在加拿大的多伦多。

Eric Sandberg, CCIE #4355, 目前是 Enventis 公司的高级网络工程师。Eric 在这个行业里有着 20 多年的丰富经验，现在他正利用 Cisco 的 AVVID 技术设计和实施多服务综合的网络。现在他住在明尼苏达州的 Minnetonka, Eric 完成了本书第 6 章“WAN 协议与技术：通过多协议传输语音”。

关于本书的技术审稿人

Lawrence Broadnax, Clover Technologies 公司的系统咨询工程师，从 1999 年开始，Lawrence 就已经是 CCIE (CCIE #5258) 了，他一直专注于异步传输模式(ATM)、路由器、交换机、AVVID、网络安全、VPN、IP QoS 以及语音数据包的研究。Lawrence 在 TELCO、ISP 以及 VAR 方面有着超过 12 年的网络与分布式系统实施的丰富经验。

Bill Kern, CCIE #5364, CCDP 是 Qwest Communications 公司的销售专家，在 Qwest 公司，Bill 为提供互联网接入与以网络为基础的 VPN 服务销售人员提供第 3 级技术支持。Bill 在网络领域有着 20 多年的丰富经验，包括多协议网络以及传统大型机环境下的网络设计、实施以及故障恢复。另外，Bill 还对各种路由选择协议、交换机服务以及 Cisco 网络产品中的各种工具非常熟悉。

Mike Reid, CCIE #2879, 他加入 Cisco 和 CCIE 组已经超过 4 年了。Mike 在 Halifax CCIE lab 任专职考官已经 3 年，现在负责路由与交换的 CCIE 认证工作。加入 Cisco 以前，Mike 在加拿大东部地区负责网络设计、组建以及管理，有超过 10 年的丰富经验。

John Tiso, CCIE #5162, 是属于 Cisco 银牌代理合作伙伴——NIS 公司的一名高级技术专家。John 在 Adelphi 大学获得理科学士学位，并且取得了 CCDP 认证，Cisco 的安全、语音与数据专家认证，以及 Sun、Microsoft 和 Novell 等多家公司的技术认证。到目前为止，John 已经在多家出版社出版了著作。我们可以通过电子邮箱 johnt@jtiso.com 和 John 取得联系。

Sze Jee Wong, CCIE #6791, 已经在数据通信领域工作了超过 10 年的时间。Wong 目前在 Enventis 公司担任高级网络工程师，负责帮助客户完成网络的设计和实施。除了 CCIE 之外，Wong 还获得了一系列专家认证，如 Cisco 的安全、语音与数据方面的专家认证，CCNP/DP 认证，CCNA-WAN，Microsoft 的 MCP 以及 Novell 的 CNE 等认证。

献 辞

本书要献给我的家人，特别是我的妻子 Sandra，感谢她长久以来对我的支持与信任，不仅仅是在我过去两年写作的时间里，还有在过去的 12 年里，我经常早出晚归地外出授课和在实验室里做实验，总被人叫走并经常在全国各地长期出差。另外，本书还要献给我的两个女儿：Amanda 和 Paige，对她们的爸爸长时间地关在实验室里表现的谅解和耐心。他们三人照亮了我工作前进的道路，使得我所做的一切都显得很有意义，很有价值。

致 谢

如果没有许多朋友、CCIE 们和其他专业人士的努力，本书是不可能完成的。

首先，要感谢 Cisco Press 的工作人员们。他们在过去的两年里不停地鼓励我。尤其要感谢 John Kane，他一直对我非常信任，并且是他将本书的出版变成了现实。还要感谢 Amy Lewis 在过去的两年里给予我的支持和指导。我也要感谢 Chris Cleveland，这位可敬的编辑是人们所希望的最好的合作伙伴。如果大家能够查一些 Cisco Press 出版的书籍，就会发现，以上的这些名字是多么熟悉。能够和他们合作是我的荣幸。

我还要感谢参与了本书的写作，为本书在语音、ISDN、ATM 等章节里面分别注入了非常有价值的实践经验的那些 CCIE 们，他们是：Eric Sandberg，Dan Keller 和 Galina Diker Pildush，他们都分别在语音、ISDN、ATM 等章节的完成方面做出了他们的贡献。我还要感谢那些作为本书技术审阅的 CCIE 们，他们对本书的付出和帮助是非常有益的，特别要提到的是 Bill Kern、Sze Jee Wong、John Tiso 和 Mike Reid，没有他们，本书的付梓是不可能的。

我还要感谢我在 Comdisco 公司曾经的和现在的那些同事和管理者们，他们对我的支持与帮助是难以用语言表达的。

还要感谢我的爸爸妈妈，感谢他们为我买的第 1 台电脑，感谢他们 20 年前为我买的那些视频游戏，感谢他们把我送上了这趟伟大的技术之旅，感谢他们一直以来对我的支持与关怀。

序

“没有准备，任何渴望成功的决心都是没有用的”

——H·D·梭罗

现在的社会已经发展到任何一个人、公司、行业乃至国家都越来越倚重于网络的力量，CCIE 计划的目的是为了从这些个人、公司、行业乃至国家中挑选出最优秀的互联网络专家，以促成他们的成功与发展。而选拔 CCIE 的标准之高也正如这个目标一样崇高。

获取 CCIE 证书的过程就是一个攀登 IT 行业技术最高峰的过程。尽管 CCIE 们在其准备获取 CCIE 证书过程中不可避免地要去学习大量的各种产品知识，但对产品的培训并不是 CCIE 认证的最终目的。CCIE 的目的在于寻找那些有能力驾驭各种端到端的网络中固有的错综复杂的内涵和理解潜在缺陷的专家，而不拘泥于是何种技术，是哪种产品。

要想成为 CCIE，必须首先通过一个资格考试，该考试是用于评估应试者对于目前各种相关技术和拓扑结构的知识。只有在这一考试中取得了专家级别所要求的分数，才能够有资格参加 CCIE 认证实验考试。这一由 Cisco 举办的考试与别的认证考试有很大区别。应试者必须通过一系列的关于互联网络搭建和优化的考试来表明其对这方面知识的掌握程度，这一系列的考试都是严格仿真目前实际 IT 世界里的各种网络环境，以实验能力作为考核的基础。

要想成为一名 CCIE，应试者必须付出大量精力去参加培训和准备。而且，严格的、以至于有点强制性的每两年必须再认证的措施，保证了应试者长期保持在专家水平以及维护本考试的真实与有效性。这些严格的规定也保证了获得 CCIE 的应试者是一直在从事这个行业的工作，并且是在不停的学习和进步之中。

Cisco 并不要求应试者在准备 CCIE 认证考试的过程中去接受某一特定的培训，因为这一考试的目的是为了考查应试者所掌握的专业技术知识以及在工作中获得的经验，而不是去简单地完成某项特定的工作。

本书能够让那些有志于获取 CCIE 证书的人们在准备的过程中更有效地安排时间。本书是一系列的 CCIE 实验考试指导丛书中的第一本，花了大量篇幅来讲述 CCIE 考试的复杂性以及所涉及的范围。以本书为代表的一系列丛书是专门用来帮助应试者准备 CCIE 实验认证考试的。尽管这并不能代替经验和已获取的专业技术知识，然而本书的确能够通过练习来巩固应试者已经掌握的技能 and 已经获得的知识，帮助有志于参加 CCIE 认证考试的应试者成功地取得 CCIE 认证证书。

Cisco Systems 公司 CCIE 项目组经理

Lorne Bradock

前言

在 1993 年的下半年，Cisco 推出了 Cisco 互联网络专家认证考试项目，向认证考试提出了挑战。在过去很多年里和该认证考试出现之前，人们以前所未有的应试比率来参加各种不同的认证考试。从这些众多的获得认证证书的人们中产生了一个新的词汇：书面认证。这里的“书面”是指人们不用去碰那些他们需要熟悉的设备就可以通过这些考试（paper certification）。从本质上来说，应试者们的知识都停留在书面上。现在，人们以前花了大量精力去获取的认证证书已经变得越来越常见和越来越没有意义了。

市场上已经有了足够多的这类书面认证证书，因而需要一种新的更加接近于实践的认证途径。IT 行业不仅仅需要能考核应试者的理论知识，更加需要能够衡量应试者的实际动手能力。Cisco 就是为了这一目的而举办了 CCIE 认证考试。CCIE 认证考试的开始之时正是其他认证考试的结束之际，那就是 2 小时的紧张的笔试考试。只有得到 70 分以上的分数，应试者才能成为 CCIE 备选者，才有资格参加 CCIE 的实验考试。实验考试是一个历时 8 个半小时的测试，在此期间，CCIE 备选者将要把他们所掌握的理论知识用在实验室的测试中。

本书是第一本由 Cisco Systems 授权讲述 CCIE 实验考试的书籍，由 9 位 CCIE 共同撰写，还包含了许多其他 CCIE 和作者的付出。希望这本书和它的后续书籍能够让大家对 CCIE 实验考试有更多更深入的了解，能够帮助有志于成为 CCIE 的人们更好地准备这一考试。

目的

本书的目的不在于仅仅指导大家如何通过 CCIE 实验考试，而是指导工程师们怎样在实验室环境里设计和模拟不同的 WAN 和 LAN 环境。从本质上来说，这就是 CCIE 考试的衡量

原则。希望这本书不仅仅用于学习中，而且也能够对实际工作有一定参考作用。

写作之初，我们是想在书中包括在 CCIE 考试中可能出现的每一个主题。最初的要点包括 BGP, IPX, AppleTalk 和 DECnet 等，但是很快我们就意识到很难在一本书里面包括所有可能的 CCIE 实验考试的题目。像 BGP 这样的主题，要想达到 CCIE 的程度，至少也需要 100 多页的篇幅才能够讲解清楚。不能够仅仅因为考试里面可能有这样的题目就将其作为一个主题写进这本书里。因此，我们现在已经着手写作《CCIE 实验指南（第 2 卷）》了。在这本书里将会把许多没有写进第 1 卷的内容包括进去，像 BGP, IPX、多播系统、VPN 等等。

本书读者对象

本书可以作为一本通用的用于配置 Cisco 路由器的技术参考书。它主要目的是用来帮助那些 CCIE 备选者们通过完成许多实验来准备 CCIE 实验考试。这些实验是用来测试应试者在参考答案之前解决实际问题的能力。我们真诚地建议大家完成这些实验，因为这些实验里的许多内容都是非常有难度的。如果你没有在实际环境中完成这些实验，那么在真正的 CCIE 考试中就有可能漏掉许多细节，从而导致整个考试的失败。

本书主要对象是至少已经获得了 CCNA 或者 CCDA 证书的网络工程师。这些工程师们在阅读本书之前都已经对 IP 地址以及子网等知识有了一个比较深入的理解。而拥有 Cisco 路由器基础认识和基本路由器配置经验对于本书的阅读也是非常有帮助的。

本书组织结构

本书结构遵从 OSI 模型及网络自低向高的构建方式。

第 1 部分主要讲述物理层和路由器基本的设置和访问知识。第 2 部分“LAN 模型的建立”和第 3 部分“采用广域网互联局域网”关注数据链路层，而第 4 部分“路由选择协议”则关注第 3 层和第 4 层的内容。每一部分后面都有一个或者多个复杂的实验。建议在查看实验的答案之前，应独立完成设计和实验搭建，以便对这些复杂的实验有一个清楚深入的理解。

本书的主要部分包括以下内容：

- 第 1 部分，“建立网络互联模型”——这一部分讲述了路由器的基本和高级设置，包括 16 位引导寄存器，路由器和交换机的密码恢复，控制台访问以及访问服务器的配置问题。这一部分还包括网络互联模型的建立，像帧中继交换，电缆的类型以及搭建一个复杂的模拟网络所需设备。
- 第 2 部分，“LAN 模型的建立”——这一部分包括配置 Catalyst 以太网和令牌环系列交换机的详细信息。包括 Catalyst 2900/3500、4000/5000/6000 以太网交换机和 3920 令牌环交换机等。该部分还涵盖了 VLAN、VTP、VLAN 中继协议 (Trunking Protocol)、生成树协议 (Spanning Tree Protocol) 等方面的详细内容。
- 第 3 部分，“采用广域网互联局域网”——这一部分介绍了广域网中数据链路层协议的配置问题，包括 HDLC、PPP、帧中继、语音传输，ISDN 与 ATM 等的详细配置信息。
- 第 4 部分，“路由选择协议”——这一部分主要讲述内部路由协议及其配置问题，包括 RIP、RIP v2、IGRP、OSPF 和 EIGRP 等信息。

由桥，综合路由桥接，源路由桥接，远程源路由桥接以及增强型数据链路层交换等方面的详细知识。

- 第6部分，“网络控制与网络访问”——这一部分专门讲述了配置与应用IP访问控制列表的不同方法，包括标准与扩展型访问控制列表以及动态访问控制列表的配置问题。二进制换算问题和通配符问题也在这一部分进行详细论述。
- 第7部分，“增强型网络协议”——这一部分分成3个章节来讲述Cisco路由器上较常见的一些特性，包括NAT、HSRP、NTP/SNTP等，还讲述了对每一特性的详细配置信息。
- 第8部分，“CCIE准备与自我评估”——这一部分讨论了现在的CCIE实验认证考试，提供了学习要点和建议以帮助应试者准备CCIE实验考试。这一部分还包括5套CCIE考试模拟题，以便给应试者一种身临其境的考试感觉。

本书中用到的图标



命令句法约定

本书中用到的句法规范是和Cisco的IOS中使用的完全一致的。句法参考：

- 竖画线 (|)：区分可选项，分开的部分之间是或的关系。
- 方括号 []：表明括号里的一些可选项。
- 大括号 {}：表明是必选项。

- 方括号里的大括号 **[{ }]**：表明是一个可选项里的必选项。
- **黑体字** 表明是命令字或者是输入的关键字。在一些配置实例和输出显示（非命令句法）里，黑体字表示需要手工输入的命令（比如说一个 **show** 命令）。
- *斜体字* 表示是一些需要赋予实际数值的命令参数。

目 录

第 1 部分 建立网络互联模型

第 1 章 建立网络互联模型的关键组件	5
1.1 确定建立网络互联模型的关键组件	6
1.2 访问服务器	6
1.3 建立局域网 (LAN) 模型	9
1.3.1 采用集线器与 MAU 建立 LAN 的模型	9
1.3.2 采用交换机建立 LAN 的模型	10
1.3.3 采用路由发起源或主干路由器仿真 LAN	10
1.3.4 利用以太网反接电缆建立 LAN 模型	11
1.4 广域网连接方法的仿真	12
1.4.1 采用特定反接电缆连接含内置或外置 CSU/DSU 的路由器实现 WAN 的建模	12
1.4.2 采用 V.35 DTE-DCE 电缆建立 WAN 的 模型	13
1.4.3 采用 HDLC 和 CSU/DSU 上的环路插头来 仿真 WAN	16
1.4.4 将一台 Cisco 路由器作为帧中继或 X.25 交换 机来建立 WAN 的模型	17
1.5 实验室中路由器，Cisco IOS 软件以及内存的 要求	18
1.6 测试主机与数据仿真	19
1.7 建立网络互联模型框架——关键组件的配置	20
1.7.1 获取特权访问：16 位的引导寄存器	20
1.7.2 Cisco IOS 软件的升级	36
1.7.3 访问服务器的设置与使用	41

1.7.5 配置路由发起源或主干路由器	51
1.7.6 配置模拟远程访问	52
1.7.7 设置 Microsoft Windows 95/98 网络	61
1.8 第1章实验指南: 简介	63
1.9 实验1: 密码恢复——第1部分	63
1.9.1 实验说明	63
1.9.2 实验内容	63
1.9.3 实验目的	64
1.9.4 所需设备	64
1.9.5 物理设计与实验准备	64
1.10 实验1: 密码恢复——第2部分	64
1.10.1 实验步骤	64
1.11 实验2: Catalyst 5500 交换机的密码恢复——第1部分	68
1.11.1 实验说明	68
1.11.2 实验内容	68
1.11.3 实验目的	68
1.11.4 所需设备	68
1.11.5 物理设计与实验准备	68
1.12 实验2: Catalyst 5500 交换机的密码恢复——第2部分	69
1.12.1 实验步骤	69
1.13 实验3: 升级 IOS 以及从 TFTP 服务器恢复配置——第1部分	69
1.13.1 实验说明	69
1.13.2 实验内容	70
1.13.3 实验目的	70
1.13.4 所需设备	70
1.13.5 物理设计与实验准备	70
1.14 实验3: 升级 IOS 以及从 TFTP 服务器恢复配置——第2部分	71
1.14.1 实验步骤	71
1.15 实验4: 访问服务器的配置——第1部分	74
1.15.1 实验说明	74
1.15.2 实验内容	74
1.15.3 实验目的	74
1.15.4 所需设备	75
1.15.5 物理设计与实验准备	75
1.16 实验4: 访问服务器的配置——第2部分	75
1.16.1 实验步骤	75
1.17 实验5: 帧中继交换机的配置——第1部分	77
1.17.1 实验说明	77
1.17.2 实验内容	78

1.17.3 实验目的	78
1.17.4 所需设备	78
1.17.5 物理设计与实验准备	78
1.18 实验 5: 帧中继交换机的配置——第 2 部分	79
1.18.1 实验步骤	79
1.19 实验 6: 远程访问实验室的配置——第 1 部分	81
1.19.1 实验说明	81
1.19.2 实验内容	82
1.19.3 实验目的	82
1.19.4 所需设备	82
1.19.5 物理设计与实验准备	82
1.20 实验 6: 远程访问实验室的配置——第 2 部分	83
1.20.1 实验步骤	83

第 2 部分 LAN 模型的建立

第 2 章 LAN 协议: Catalyst 以太网和令牌环交换机的配置	89
2.1 以太网: 协议发展简史	90
2.2 以太网技术概览	92
2.2.1 以太网的工作原理	92
2.3 802.1d 生成树协议 (STP)	96
2.3.1 生成树 (STP) 工作原理	96
2.3.2 STP 定时器	100
2.4 以太网交换技术	101
2.4.1 广播域与冲突域	102
2.4.2 虚拟局域网 (VLAN)	103
2.4.3 VTP 和中继协议	105
2.4.4 Catalyst 以太网交换机的设置	110
2.5 令牌环: 已有 30 年历史, 仍然在使用	149
2.6 令牌环技术概览	150
2.6.1 令牌环的工作原理	150
2.7 令牌环交换技术	152
2.8 令牌环网桥中继功能 (TrBRF) 与令牌环集中器中继功能 (TrCRF)	152
2.9 在 Catalyst 3920 上配置令牌环交换	155
2.9.1 交换机的配置界面	156
2.9.2 信息统计界面	157
2.9.3 下载/上传界面	158

2.9.4 复位界面	158
2.9.5 在 Catalyst 3920 交换机上设置 VLAN	158
2.10 更多练习：以太网/令牌环网实验	165
2.11 实验 7：以太交换、VLAN 中继和生成树根布局——第 1 部分	165
2.11.1 实验说明	165
2.11.2 实验内容	166
2.11.3 实验目的	166
2.11.4 所需设备	168
2.11.5 物理设计与实验准备	168
2.12 实验 7：以太交换、VLAN 中继和生成树根布局——第 2 部分	168
2.12.1 实验步骤	168
2.13 实验 8：用 Catalyst 3920 配置令牌环交换网络——第 1 部分	180
2.13.1 实验说明	180
2.13.2 实验内容	180
2.13.3 实验目的	180
2.13.4 所需设备	180
2.13.5 物理设计与实验准备	181
2.14 实验 8：用 Catalyst 3920 配置令牌环交换网络——第 2 部分	181
2.14.1 实验步骤	181

第 3 部分 采用广域网互连局域网

第 3 章 WAN 协议与技术：高级数据链路控制 (HDLC) 191

3.1 HDLC 的兼容性和简易性	192
3.1.1 HDLC 的设置	194
3.1.2 HDLC 的 “Big show” 和 “Big D”	195
3.1.3 show interface serial_interface 命令	195
3.1.4 show controllers serial_interface 命令	196
3.1.5 debug serial interface 命令	196
3.2 实验 9：HDLC 的配置——第 1 部分	201
3.2.1 实验说明	201
3.2.2 实验内容	201
3.2.3 实验目的	201
3.2.4 所需设备	201
3.2.5 物理设计和实验准备	201
3.3 实验 9：HDLC 的配置——第 2 部分	202
3.3.1 实验步骤	202

第4章 WAN 协议与技术：点对点协议 (PPP)	207
4.1 PPP 的多种用途	209
4.1.1 在同步串行链路上配置 PPP	209
4.1.2 在模拟拨号链路的异步端口上进行 PPP 配置	211
4.1.3 PPP 数据压缩的配置	229
4.1.4 配置多链路捆绑 PPP	230
4.1.5 PPP 的 LAPB 和 LQM 的配置	232
4.1.6 PPP 和 DDR 的 “Big show” 和 “Big D”	233
4.1.7 PPP 回拨设置	235
4.2 实验 10：在异步拨号连接上配置 PPP、PAP 和数据压缩——第 1 部分	235
4.2.1 实验说明	235
4.2.2 实验内容	235
4.2.3 实验目的	236
4.2.4 所需设备	236
4.2.5 物理设计和实验准备	236
4.3 实验 10：在异步拨号连接上配置 PPP、PAP 和数据压缩——第 2 部分	237
4.3.1 实验步骤	237
4.4 实验 11：同步链路上的 PPP、CHAP 和 LQM 配置——第 1 部分	244
4.4.1 实验说明	244
4.4.2 实验内容	244
4.4.3 实验目的	245
4.4.4 所需设备	245
4.4.5 物理设计和实验准备	245
4.5 实验 11：同步链路上的 PPP、CHAP 和 LQM 配置——第 2 部分	245
4.5.1 实验步骤	245
4.6 实验 12：同步链路的 PPP 模拟拨号备份——第 1 部分	250
4.6.1 实验说明	250
4.6.2 实验内容	251
4.6.3 实验目的	251
4.6.4 所需设备	251
4.6.5 物理设计和实验准备	251
4.7 实验 12：同步连接的 PPP 模拟拨号备份——第 2 部分	252
4.7.1 实验步骤	252
第5章 WAN 协议与技术：帧中继	261
5.1 帧中继的相关术语	261
5.2 帧中继技术概览	263
5.2.1 帧中继 LMI 的操作	264

5.3 帧中继的配置	266
5.3.1 实例：配置混合型帧中继网络	268
5.4 帧中继的“Big show”和“Big D”命令	272
5.4.1 show frame-relay pvc 命令	273
5.4.2 show frame-relay lmi 命令	274
5.4.3 show frame-relay map 命令	275
5.4.4 debug frame-relay lmi 命令	276
5.5 其他帧中继配置命令	277
5.6 帧中继流量整形的设置	278
5.6.1 实例：帧中继流量整形的配置	280
5.7 实验 13：配置帧中继网络与控制帧中继 ARP——第 1 部分	282
5.7.1 实验说明	282
5.7.2 实验内容	282
5.7.3 实验目的	282
5.7.4 所需设备	283
5.7.5 物理设计和实验准备	283
5.8 实验 13：配置帧中继网络与控制帧中继 ARP——第 2 部分	285
5.8.1 实验步骤	285
5.9 实验 14：帧中继网络、数据整形、OSPF 及 DLSw/LLC2 配置——第 1 部分	291
5.9.1 实验说明	291
5.9.2 实验内容	292
5.9.3 实验目的	292
5.9.4 所需设备	292
5.9.5 物理设计与实验准备	292
5.10 实验 14：帧中继网络、数据整形、OSPF 及 DLSw/LLC2 的配置——第 2 部分	294
5.10.1 实验步骤	294

第 6 章 WAN 协议与技术：通过多协议传输语音 301

6.1 模拟电话技术简介	302
6.1.1 电话呼叫的组件	302
6.1.2 电话信令	303
6.1.3 本地环路	305
6.1.4 语音交换机	305
6.1.5 中继	306
6.1.6 中继抢占信令的类型	308
6.1.7 中继监控	313
6.1.8 2 线到 4 线转换和回音	315
6.2 数字语音技术	315

6.2.1 模拟信号的数字化	316
6.2.2 模拟信号到数字信号的转换过程	316
6.2.3 数字语音插入	319
6.2.4 信道信令类型和帧格式	319
6.3 Cisco 语音产品	321
6.3.1 Cisco 1750	321
6.3.2 Cisco 2600	321
6.3.3 Cisco 3600	321
6.3.4 Cisco MC3810	321
6.3.5 Cisco 7200	322
6.3.6 Cisco 语音路由器的比较	322
6.4 实验 15: 通过帧中继、IP 和 ATM 传输语音	322
6.4.1 实验说明	322
6.4.2 实验内容	322
6.4.3 实验目的	323
6.4.4 所需设备	323
6.5 实验 15a: VoFR 的配置——第 1 部分	323
6.5.1 物理设计与实验准备	323
6.5.2 语音端口的配置与验证	324
6.6 实验 15a: VoFR 的配置——第 2 部分	326
6.6.1 实验步骤	326
6.7 实验 15b: VoIP 的配置——第 1 部分	330
6.7.1 所需设备	330
6.7.2 物理设计与实验准备	331
6.8 实验 15b: VoIP 的配置——第 2 部分	331
6.8.1 实验步骤	331
6.9 实验 15c: VoATM 的配置——第 1 部分	336
6.9.1 所需设备	336
6.9.2 物理设计与实验准备	337
6.10 实验 15c: VoATM 的配置——第 2 部分	337
6.10.1 实验步骤	337
6.11 实验 15d: 可选实验，私有专线自动振铃 (PLAR) 连接	341
第 7 章 WAN 协议与技术：综合业务数字网 (ISDN)	345
7.1 ISDN 的发展、组成和结构	345
7.1.1 ISDN 组件和参考点	346
7.1.2 ISDN 分层	348
7.1.3 ISDN 数据封装格式	348
7.2 ISDN 配置基础	348

7.3 按需拨号路由（DDR）的配置	350
7.3.1 第1步：ISDN 交换机类型和 SPID 信息的设置	351
7.3.2 第2步：指定用户所期望的数据	352
7.3.3 第3步：拨号信息的设置	352
7.3.4 第4步：配置高级可选参数	358
7.4 ISDN 调试的“Big show”和“Big D”命令	403
7.4.1 ISDN 的“Big show”	403
7.4.2 ISDN 的“Big D”	408
7.5 技巧和窍门	411
7.6 ISDN 实验	412
7.7 实验 16：配置 ISDN 上的 PPP 认证、回拨和多链路连接	412
7.7.1 实验 16 的解决方案	413
7.7.2 实验 16 解决方案的讨论	418
7.8 实验 17：配置 ISDN 上 OSPF 按需电路	418
7.8.1 实验 17 的解决方案	419
7.8.2 实验 17 解决方案的讨论	425
7.9 总结	426

第8章 WAN 协议与技术：异步传输模式（ATM） 429

8.1 ATM 实验学习所需的特定组件	431
8.2 RFC 2684 的配置	434
8.2.1 PVC 的配置	434
8.2.2 SVC 的配置	438
8.3 RFC 2225（经典 IP）的配置	443
8.3.1 PVC 的配置	443
8.3.2 SVC 的配置	445
8.4 实验 18：Cisco 7XXX 路由器上的 PVC，RFC 2684 的配置——第1部分	449
8.4.1 实验说明	449
8.4.2 实验内容	449
8.4.3 实验目的	450
8.4.4 所需设备	450
8.4.5 物理设计与实验准备	450
8.5 实验 18：Cisco 7XXX 路由器上的 PVC，RFC 2684 的配置——第2部分	451
8.5.1 实验步骤	451
8.6 实验 19：在 Cisco 7XXX 路由器上利用 SVC 对经典 IP，RFC 2225 进行配置——第1部分	455
8.6.1 实验说明	455
8.6.2 实验内容	456
8.6.3 实验目的	456

8.6.4 所需设备	456
8.6.5 物理设计与实验准备	456
8.7 实验 19: 在 Cisco 7XXX 路由器上利用 SVC 对经典 IP、RFC 2225 进行配置 ——第 2 部分	457
8.7.1 实验步骤	457
8.8 总结	461

第 4 部分 路由选择协议

第 9 章 距离矢量协议: 路由信息协议版本 1 和版本 2 (RIP-1 和 RIP-2)	473
9.1 RIP 技术概览	473
9.1.1 有类路由 (仅 RIP-1)	474
9.1.2 无类路由 (仅 RIP-2)	475
9.2 RIP-1 和 RIP-2 的配置	476
9.2.1 RIP-1 的配置	476
9.2.2 RIP-2 的配置	477
9.2.3 RIP 的 “Big show” 和 “Big D”	478
9.2.4 show ip protocols {summary} 命令	479
9.2.5 show ip route 命令	479
9.2.6 debug ip rip {events} 命令	480
9.3 RIP 更新信息的调整、重分布和控制	480
9.4 RIP 默认路由	484
9.5 实验 20: 集成 RIP 网络: 重分布、路由的过滤和控制——第 1 部分	485
9.5.1 实验说明	485
9.5.2 实验内容	485
9.5.3 实验目的	486
9.5.4 所需设备	486
9.5.5 物理设计与实验准备	486
9.6 实验 20: 集成 RIP 网络: 重分布、路由的过滤和控制——第 2 部分	488
9.6.1 实验步骤	488
第 10 章 距离矢量协议: 内部网关路由选择协议 (IGRP)	495
10.1 IGRP 技术概览	496
10.1.1 IGRP 的路由类型	497
10.1.2 IGRP 的度量	498
10.2 IGRP 的配置	499
10.2.1 IGRP 的 “Big show” 和 “Big D”	499

10.3 IGRP 更新信息的调整、重分布和控制	502
10.3.1 非等价路由开销的负载平衡	503
10.3.2 IGRP 的非等价路由开销的负载平衡的配置	504
10.3.3 IGRP 和 EIGRP 的集成和移植	507
10.3.4 IGRP 和默认路由	509
10.4 实验 21：配置 IGRP：默认路由、路由过滤和非等价负载平衡——第 1 部分	510
10.4.1 实验说明	510
10.4.2 实验内容	510
10.4.3 实验目的	511
10.4.4 所需设备	512
10.4.5 物理设计与实验准备	512
10.5 实验 21：配置 IGRP：默认路由、路由过滤和非等价负载平衡——第 2 部分	512
10.5.1 实验步骤	512
第 11 章 混合协议：增强型内部网关路由选择协议（EIGRP）	521
11.1 EIGRP 技术概览	522
11.1.1 EIGRP 的度量	522
11.1.2 EIGRP 的邻居路由器	528
11.1.3 EIGRP 的可靠传输协议（RTP）	529
11.1.4 扩散刷新算法（DUAL）	530
11.1.5 协议相关模块	531
11.2 水平分隔	531
11.3 EIGRP 的配置	533
11.4 EIGRP 的“Big show”和“Big D”命令	534
11.4.1 show ip eigrp neighbors 命令	534
11.4.2 show ip eigrp topology 命令	535
11.4.3 show ip protocols 命令	536
11.4.4 show ip route 命令	537
11.4.5 debug eigrp packets 命令	537
11.4.6 eigrp log-neighbor-changes 命令	538
11.5 调整 EIGRP 的更新信息	538
11.6 EIGRP 的重分布和路由控制	539
11.6.1 实例：EIGRP 重分布的应用	540
11.6.2 实例：EIGRP 路由控制的应用	544
11.7 EIGRP 的汇总	546
11.7.1 通过汇总控制查询范围以及 SIA 路由的问题	546
11.7.2 EIGRP 的自动汇总功能	548
11.7.3 EIGRP 的手动汇总或路由聚合	550
11.8 EIGRP 的默认路由	552

11.9 EIGRP 的存根路由	554
11.10 EIGRP 的等价路由开销和非等价路由开销的负载平衡	556
11.11 实验 22: 配置 EIGRP: 路由重分布、汇总以及存根路由——第 1 部分	558
11.11.1 实验说明	558
11.11.2 实验内容	558
11.11.3 实验目的	558
11.11.4 所需设备	559
11.11.5 物理设计与实验准备	559
11.12 实验 22: 配置 EIGRP: 路由重分布、汇总以及存根路由——第 2 部分	560
11.12.1 实验步骤	560
11.13 实验 23: 配置 EIGRP 网络: 默认路由、路由的管理与过滤——第 1 部分	570
11.13.1 实验说明	570
11.13.2 实验内容	570
11.13.3 实验目的	570
11.13.4 所需设备	571
11.13.5 物理设计与实验准备	571
11.14 实验 23: 配置 EIGRP 网络: 默认路由、路由的管理与过滤——第 2 部分	572
11.14.1 实验步骤	572
第 12 章 链路状态协议: 开放式最短路径优先 (OSPF)	581
12.1 OSPF 技术概览	582
12.1.1 OSPF 的 Hello 协议	583
12.1.2 OSPF 的邻居路由器和网络类型	584
12.1.3 指定路由器 (DR) 和备份指定路由器 (BDR)	584
12.1.4 OSPF 的路由器标识 (RID)	585
12.1.5 OSPF 的基本邻接关系	587
12.1.6 最短路径树 (SPF) 和 OSPF 的度量代价	590
12.1.7 OSPF 的路由器类型、区域以及 LSA	591
12.1.8 OSPF 的确认信号	595
12.1.9 OSPF 的路径类型	595
12.2 配置 OSPF	597
12.2.1 实例: 在帧中继中配置多 OSPF 区域的类型	599
12.3 OSPF 的 “Big show” 和 “Big D” 命令	609
12.3.1 show ip ospf neighbors 命令	610
12.3.2 show ip ospf database 命令	611
12.3.3 show ip ospf interface 命令	612
12.3.4 show ip route 命令	613
12.3.5 show ip ospf 命令	613
12.3.6 debug ip ospf adj 和 debug ip ospf events 命令	614

12.3.7	log-adjacency-changes/show log 命令	615
12.3.8	clear ip ospf process	615
12.4	OSPF 的存根区域配置	615
12.5	OSPF 的调整	616
12.6	减少 OSPF 的扩散	616
12.7	OSPF 重分布和路由控制	617
12.7.1	用于控制路由过滤和重分布的命令	617
12.7.2	用于改变 OSPF 的路由选择的命令	617
12.7.3	实例：路由的过滤和重分布	618
12.8	OSPF 的汇总功能	622
12.9	OSPF 的默认路由	625
12.10	OSPF 的认证	627
12.10.1	类型 I 认证方式	627
12.10.2	类型 II 认证方式	627
12.10.3	类型 I 和类型 II 认证实例	628
12.11	OSPF 按需电路和备份	629
12.11.1	坚持 OSPF 的设计规则	629
12.11.2	OSPF 按需电路	630
12.11.3	Area 0 的设计准则	630
12.12	OSPF 的虚链路	630
12.13	实验 24：配置 OSPF：多域路由、认证、路径管理和默认路由——第 1 部分	632
12.13.1	实验说明	632
12.13.2	实验内容	632
12.13.3	实验目的	632
12.13.4	所需设备	633
12.13.5	物理设计与实验准备	633
12.14	实验 24：配置 OSPF：域间路由、认证、路径管理和默认路由——第 2 部分	635
12.14.1	实验步骤	635
12.15	实验 25：配置 OSPF：多域路由、路由的重分布与汇总功能——第 1 部分	645
12.15.1	实验说明	645
12.15.2	实验内容	646
12.15.3	实验目的	646
12.15.4	所需设备	646
12.15.5	物理设计与实验准备	647
12.16	实验 25：配置 OSPF：多域路由、路由的重分布与汇总功能——第 2 部分	648
12.16.1	实验步骤	648

11.9 EIGRP 的存根路由	554
11.10 EIGRP 的等价路由开销和非等价路由开销的负载平衡	556
11.11 实验 22: 配置 EIGRP: 路由重分布、汇总以及存根路由——第 1 部分	558
11.11.1 实验说明	558
11.11.2 实验内容	558
11.11.3 实验目的	558
11.11.4 所需设备	559
11.11.5 物理设计与实验准备	559
11.12 实验 22: 配置 EIGRP: 路由重分布、汇总以及存根路由——第 2 部分	560
11.12.1 实验步骤	560
11.13 实验 23: 配置 EIGRP 网络: 默认路由、路由的管理与过滤——第 1 部分	570
11.13.1 实验说明	570
11.13.2 实验内容	570
11.13.3 实验目的	570
11.13.4 所需设备	571
11.13.5 物理设计与实验准备	571
11.14 实验 23: 配置 EIGRP 网络: 默认路由、路由的管理与过滤——第 2 部分	572
11.14.1 实验步骤	572
第 12 章 链路状态协议: 开放式最短路径优先 (OSPF)	581
12.1 OSPF 技术概览	582
12.1.1 OSPF 的 Hello 协议	583
12.1.2 OSPF 的邻居路由器和网络类型	584
12.1.3 指定路由器 (DR) 和备份指定路由器 (BDR)	584
12.1.4 OSPF 的路由器标识 (RID)	585
12.1.5 OSPF 的基本邻接关系	587
12.1.6 最短路径树 (SPF) 和 OSPF 的度量代价	590
12.1.7 OSPF 的路由器类型、区域以及 LSA	591
12.1.8 OSPF 的确认信号	595
12.1.9 OSPF 的路径类型	595
12.2 配置 OSPF	597
12.2.1 实例: 在帧中继中配置多 OSPF 区域的类型	599
12.3 OSPF 的 “Big show” 和 “Big D” 命令	609
12.3.1 show ip ospf neighbors 命令	610
12.3.2 show ip ospf database 命令	611
12.3.3 show ip ospf interface 命令	612
12.3.4 show ip route 命令	613
12.3.5 show ip ospf 命令	613
12.3.6 debug ip ospf adj 和 debug ip ospf events 命令	614

12.3.7	log-adjacency-changes/show log 命令	615
12.3.8	clear ip ospf process	615
12.4	OSPF 的存根区域配置	615
12.5	OSPF 的调整	616
12.6	减少 OSPF 的扩散	616
12.7	OSPF 重分布和路由控制	617
12.7.1	用于控制路由过滤和重分布的命令	617
12.7.2	用于改变 OSPF 的路由选择的命令	617
12.7.3	实例：路由的过滤和重分布	618
12.8	OSPF 的汇总功能	622
12.9	OSPF 的默认路由	625
12.10	OSPF 的认证	627
12.10.1	类型 I 认证方式	627
12.10.2	类型 II 认证方式	627
12.10.3	类型 I 和类型 II 认证实例	628
12.11	OSPF 按需电路和备份	629
12.11.1	坚持 OSPF 的设计规则	629
12.11.2	OSPF 按需电路	630
12.11.3	Area 0 的设计准则	630
12.12	OSPF 的虚链路	630
12.13	实验 24：配置 OSPF：多域路由、认证、路径管理和默认路由——第 1 部分	632
12.13.1	实验说明	632
12.13.2	实验内容	632
12.13.3	实验目的	632
12.13.4	所需设备	633
12.13.5	物理设计与实验准备	633
12.14	实验 24：配置 OSPF：域间路由、认证、路径管理和默认路由——第 2 部分	635
12.14.1	实验步骤	635
12.15	实验 25：配置 OSPF：多域路由、路由的重分布与汇总功能——第 1 部分	645
12.15.1	实验说明	645
12.15.2	实验内容	646
12.15.3	实验目的	646
12.15.4	所需设备	646
12.15.5	物理设计与实验准备	647
12.16	实验 25：配置 OSPF：多域路由、路由的重分布与汇总功能——第 2 部分	648
12.16.1	实验步骤	648

第5部分 不可路由协议的传输

第13章 配置桥接和增强数据链路交换 (DLSw+)	665
13.1 透明桥接 (Transparent Bridging)	666
13.1.1 透明桥接的工作	666
13.1.2 透明桥接的配置	669
13.1.3 透明桥接模型	671
13.1.4 透明桥接的检验，透明桥接和 STP 的 “Big show” 命令	673
13.2 综合路由和桥接	675
13.2.1 IRB 的注意点	675
13.2.2 配置 IRB	676
13.2.3 实例：IRB 的配置	677
13.3 源路由桥接 (SRB)	682
13.3.1 源路由桥接 (SRB) 概览	683
13.3.2 源路由桥接 (SRB) 的配置	686
13.3.3 实例：远程源路由桥接的配置	693
13.3.4 其他 SRB 功能与特性的配置	699
13.4 增强数据链路交换 (DLSw+)	703
13.4.1 DLSw+ 的特性	704
13.4.2 DLSw+ 技术概览	705
13.4.3 DLSw+ 的配置	710
13.4.4 实例：DLSw TCP 和 FST 对等体	712
13.4.5 DLSw+ 的 “Big show” 和 “Big D” 命令	717
13.4.6 DLSw+ 的高级配置	722
13.5 网桥环境下的数据过滤	735
13.5.1 对服务接入点 (SAP) 的过滤	735
13.5.2 MAC 地址的过滤	737
13.5.3 NetBIOS 名称的过滤	737
13.5.4 实例：网桥环境中的过滤	737
13.6 实验 26：透明桥接、远程源路由桥接 (RSRB) 和 LSAP 过滤——第 1 部分	738
13.6.1 实验说明	738
13.6.2 实验内容	738
13.6.3 实验目的	739
13.6.4 所需设备	740
13.6.5 物理设计与实验准备	740
13.7 实验 26：透明桥接、远程源路由桥接 (RSRB) 和 LSAP 过滤——第 2 部分	741
13.7.1 实验步骤	741

13.8 实验 27: DLSw+的 TCP、LLC2、混杂、动态以及备份对等体的配置——	
第 1 部分	751
13.8.1 实验说明	751
13.8.2 实验内容	752
13.8.3 实验目的	754
13.8.4 所需设备	754
13.8.5 物理设计与实验准备	754
13.9 实验 27: DLSw+的 TCP、LLC2、混杂、动态以及备份对等体的配置——	
第 2 部分	755
13.9.1 实验步骤	755
13.10 实验 28: DLSw+的可达性，边界对等体，按需对等体和弹性对等体的配置——	
第 1 部分	764
13.10.1 实验说明	764
13.10.2 实验内容	764
13.10.3 实验目的	765
13.10.4 所需设备	766
13.10.5 物理设计与实验准备	766
13.11 实验 28: DLSw+的可达性，边界对等体，按需对等体和弹性对等体的配置——	
第 2 部分	766
13.11.1 实验步骤	766

第 6 部分 网络控制与网络访问

第 14 章 理解 IP 访问控制列表	777
14.1 理解访问控制列表的工作方式	778
14.2 访问控制列表、反向掩码和二进制算术	779
14.3 标准访问控制列表	781
14.4 扩展访问控制列表	784
14.5 访问控制列表的显示	790
14.6 动态访问控制列表	791
14.7 命名访问控制列表	794
14.8 实验 29: 配置访问控制列表、命名访问控制列表以及 EIGRP 路由过滤——	
第 1 部分	795
14.8.1 实验说明	795
14.8.2 实验内容	795
14.8.3 实验目的	796
14.8.4 所需设备	796

14.8.5 物理设计与实验准备	796
14.9 实验 29: 配置访问控制列表、命名访问控制列表以及 EIGRP 路由过滤——	
第 2 部分	797
14.9.1 实验步骤	797
14.10 实验 30: 利用命名访问控制列表配置动态访问控制列表和数据过滤——	
第 1 部分	803
14.10.1 实验说明	803
14.10.2 实验内容	803
14.10.3 实验目的	803
14.10.4 所需设备	804
14.10.5 物理设计与实验准备	804
14.11 实验 30: 利用命名访问控制列表配置动态访问控制列表和数据过滤——	
第 2 部分	805
14.11.1 实验步骤	805

第 7 部分 增强型网络协议

第 15 章 配置网络地址转换 (NAT)	813
15.1 NAT 技术概览	813
15.1.1 NAT 的术语	814
15.2 NAT 和 RFC 1918	816
15.3 NAT 的配置	817
15.3.1 NAT 动态转换方式的配置	818
15.3.2 NAT 静态转换方式的配置	820
15.3.3 简单 IP 和端口地址转换 (PAT) 的配置	821
15.4 NAT 的 “Big show” 和 “Big D” 命令	824
15.5 NAT 转换的清除和改变	826
15.6 NAT 的局限性以及使用	826
15.7 NAT 与非标准 FTP 端口号	827
15.8 实验 31: 配置动态 NAT 与非标准 FTP 端口号的应用——第 1 部分	828
15.8.1 实验说明	828
15.8.2 实验内容	828
15.8.3 实验目的	828
15.8.4 所需设备	829
15.8.5 物理设计与实验准备	829
15.9 实验 31: 配置动态 NAT 与非标准 FTP 端口号的应用——第 2 部分	829
15.9.1 实验步骤	829

15.10 实验 32: 配置静态 NAT 和 DLSw——第 1 部分	835
15.10.1 实验说明	835
15.10.2 实验内容	835
15.10.3 实验目的	835
15.10.4 所需设备	835
15.10.5 物理设计与实验准备	836
15.11 实验 32: 配置静态 NAT 和 DLSw——第 2 部分	836
15.11.1 实验步骤	836
第 16 章 热备份路由选择协议（HSRP）的使用	843
16.1 HSRP 的概览与配置	844
16.1.1 在路由器之间配置 HSRP	846
16.2 HSRP 的“Big show”和“Big D”命令	848
16.3 实验 33: 配置 HSRP、跟踪与非对称路由——第 1 部分	849
16.3.1 实验说明	849
16.3.2 实验内容	849
16.3.3 实验目的	850
16.3.4 所需设备	850
16.3.5 物理设计与实验准备	850
16.4 实验 33: 配置 HSRP、跟踪与非对称路由——第 2 部分	851
16.4.1 实验步骤	851
第 17 章 网络时间协议（NTP）与简单网络时间协议（SNTP）的配置	861
17.1 NTP 技术概览	861
17.2 NTP 的配置	863
17.2.1 NTP 广播客户端模式的设置	863
17.2.2 NTP 静态客户模式的配置	865
17.2.3 NTP 主模式的配置	867
17.2.4 NTP 对等体关系的配置	868
17.2.5 NTP 认证以及与时间相关的选项的配置	869
17.3 简单网络时间协议（SNTP）的配置	872
17.4 NTP 和 SNTP 的“Big show”和“Big D”命令	873
17.5 实验 34: 配置 NTP 服务器、客户端和认证——第 1 部分	875
17.5.1 实验说明	875
17.5.2 实验内容	875
17.5.3 实验目的	875
17.5.4 所需设备	876
17.5.5 物理设计与实验准备	876
17.6 实验 34: 配置 NTP 服务器、客户端和认证——第 2 部分	876

17.6.1 实验步骤	876
17.7 实验 35: 配置 NTP 服务器、客户端和对等体——第 1 部分	880
17.7.1 实验说明	880
17.7.2 实验内容	880
17.7.3 实验目的	881
17.7.4 所需设备	881
17.7.5 物理设计与实验准备	881
17.8 实验 35: 配置 NTP 服务器、客户端和对等体——第 2 部分	882
17.8.1 实验步骤	882

第 8 部分 CCIE 准备与自我评估

第 18 章 CCIE 实验考试: 考试准备与 CCIE 实验室练习	889
18.1 新的一天制 CCIE 试验考试	890
18.2 怎样成为一名 CCIE	890
18.3 CCIE: 推荐读物以及知识点提纲	891
18.4 CCIE 实验考试模拟练习: “Skynet” 的配置	895
18.4.1 设备清单	895
18.4.2 实验准备工作: 帧中继交换机的配置	896
18.4.3 实验准备工作: 主干路由器的配置	897
18.4.4 计时实验考试部分	898
18.5 CCIE 实验考试的模拟练习: “Darth Reid”	902
18.5.1 设备清单	902
18.5.2 实验准备工作: 帧中继交换机的配置	902
18.5.3 实验准备工作: 主干路由器的配置	903
18.5.4 计时实验考试部分	905
18.6 CCIE 实验考试的模拟练习: “The Lab, the Bad, the Ugly”	909
18.6.1 设备清单	909
18.6.2 实验准备工作: 帧中继交换机的配置	910
18.6.3 计时实验考试部分	911
18.7 CCIE 实验考试的模拟练习: “The Enchilada”	915
18.7.1 设备清单	915
18.7.2 实验准备工作: 帧中继交换机的配置	915
18.7.3 计时实验考试部分	916
18.8 CCIE 实验考试的模拟练习: “The Unnamed Lab”	921
18.8.1 设备清单	921
18.8.2 实验准备工作: 帧中继交换机的配置	921

18.8.3 实验准备工作: 主干路由器的配置	922
18.8.4 计时实验考试部分	924

第9部分 附录

附录 A ISDN 交换机类型、原因代码以及原因代码值	931
A.1 交换机类型	931
A.2 原因代码字段	932
A.3 原因代码值	933
A.4 承载能力值	935
A.5 “处理”(Progress)字段的值	936
附录 B 简化的 OSI 参考模型	939
附录 C RFC 清单	941
附录 D 常见的电缆类型以及引脚定义	949
D.1 控制台端口与辅助端口的信号与引脚定义	949
D.2 串行电缆的部件和引脚定义	950
D.2.1 EIA-530	951
D.2.2 EIA/TIA-232	952
D.2.3 EIA/TIA-449	955
D.2.4 V.35	957
D.2.5 X.21	960
D.3 以太电缆的部件与引脚定义	962
D.4 令牌环端口引脚定义	962
D.5 异步串行端口	963
D.6 RJ-45 适配器的引脚定义	966
附录 E 参考书目	969

第1 部分

建立网络互联模型

第1章 建立网络互联模型的关键组件

第 1 章

建立网络互联模型 的关键组件

现今存在着各种各样的模型，从数学模型、统计模型到小时候可能做过的橡皮泥模型。尽管这些模型各有特点，但它们都属于同一种类。本书提出的是一种新的模型——网络互联模型。

网络互联模型可以定义为对大型互联网络的各种功能具体化、精确化的表述。和所有的模型一样，网络互联模型是大型网络的小型化。这里的精确是指模型必须要反映“真实”网络的所有元素的精确要求。例如，你不仅要设计一个 OSPF 网络，而且还要关心整个网络设计的细节问题，例如某个特定接口的 OSPF 接口类型，它们在什么区域，是否应该发送链路状态或者形成邻居关系等等。学习过程中多注意这类细节问题是非常重要的，这样才能保证“精确”。最后，这样的模型还必须实用，也就是说，能够接受实际数据和应用的测试。

正如其他的模型用于证明某个假设一样，网络互联模型用来证明大型互联网络的功能和设计理论。网络模型的设计完成之后，可以在多种网络模型上传输多种数据，通过使用测试主机和仿真数据来对网络功能进行检测。

总之，网络互联模型是大型互联网络的具体而精确的表述。本书的目的就是引导网络工程师组装和配置复杂网络互联模型所有必需的硬件和软件组件。

网络互联这一术语可以定义为网络的集合，即局域网（LAN）、广域网（WAN）通过路由器、网桥以及交换机连接在一起而形成的整个网络。要想在实验室里正确地搭建网

络互联模型，必须要正确地仿真各种不同的 LAN 和 WAN 技术。

建立网络互联模型时，必须遵循一定的逻辑顺序。就像数学一样，网络是建立在网络的基础之上的。一个人在学习代数之前，必须要先学习四则运算，在学习微积分之前，必须要先学习代数，以此类推，建立网络也遵循类似的逻辑方法。

第 1 步 首先，建立所有的 LAN 连接。

第 2 步 在初始的 LAN 连接建立起来之后，建立和配置所有的 WAN 连接。

第 3 步 建立完整的互联网络连接。这是通过在 LAN 和 WAN 上配置适当的路由选择协议来完成的。

第 4 步 最后，应用 IP 网络正常运转所要求的所有过滤、特性或者是其他的外部路由选择协议，如边界网关协议 (BGP)。

1.1 确定建立网络互联模型的关键组件

建立一个复杂、完整的网络互联模型，需要的一些关键组件包括：

- 访问服务器。
- 局域网 (LAN)：交换机/集线器和相应的电缆。
- 广域网 (WAN)：路由器和相应的电缆。
- 路由器。
- 测试主机与应用程序，最好是 Microsoft Windows 95/98/2000 或者是 Windows NT。

上面这个组件列表应该看作是功能列表而不是设备列表。设备型号并不重要，关键是设备在网络模型中的功能。仿真 WAN 有很多种方法。例如，在某些网络模型中，如果配置一个帧中继交换机，模型会更加精确真实，而在另外一些模型中则由于所用的协议无关紧要，只需要一个 WAN 连接就可以了。

上述列表中，惟一可选项就是访问服务器。不论是在实验室环境下还是在实际应用中，访问服务器都可能是很有用的。在实际应用环境中，访问服务器在有多台的路由器的环境下能够提供带外管理功能。访问服务器只需要一个调制解调器就能够作为一个中心点提供带外管理服务，而不需要很多的调制解调器通过拨号来访问路由器。而在实验室环境下，访问服务器的主要功能在于提供对多个路由器简单快捷的访问。在实验中，我们将详细讨论这些网络模型组件的功能。

1.2 访问服务器

访问服务器是主要的配置设备。可以通过这一设备使用反向 Telnet 会话来配置其他路由器和交换机。最有效的 Cisco 路由器类型是具有 SCSI-II 的 68 针异步通信口和 1 转 8 电缆的路由器，常见型号如下：

- Cisco 2509/2510
- Cisco 2511/2512

任何一个 Cisco 路由器可以在其串口上配置异步通信。像 Cisco 2522 这样具有多串口的路由器也可以用作访问服务器。不论是使用 2509 这样的带有 SCSI-II 68 针端口的路由器还是采用 Cisco 2522 这样有 10 个串口的路由器，一些特定的电缆和配置都是必需的。

最常见的用作访问服务器的路由器是 Cisco 2509/2510 和 Cisco 2511/2512。Cisco 2509 有一个以太网口和 8 个异步通信口，而 Cisco 2510 有一个令牌环口和 8 个异步通信口。这 8 个异步通信口可与一条 Cisco 的 1 转 8 电缆互联。该电缆叫做八爪电缆，电缆编号是 CAB-OCTAL-KIT，它还包括一个用于异步通信设备，如调制解调器的头。这 8 条电缆一端的 RJ-45 头用于插入实验室每台路由器的控制端口（Console）里。这里要使用 Cisco 称为反向 Telnet 会话的方法通过这些电缆来对路由器进行配置。这也许应该称内部 Telnet，配置路由器的时候，Telnet 使用的方法并没有什么正、反之分。实际上，Telnet 是作为一个协议，通过特定的内部端口或者是连线来和设备进行通信的。

注释 关于本书中提到的各种电缆线和电缆头等更为详尽的信息，请参考本书的附录 D “常见的电缆类型以及引脚定义”。

Cisco 2511 访问服务器使用一个 68 针的连接器和导出电缆，每条电缆上都有 8 个 RJ-45 端口。这些端口也可以使用 RJ-45 到 DB-25 的转换接头与异步设备相连接。Cisco 2511/2512 路由器的外形与此相同，但多了 8 个异步通信口。

图 1-1 显示了 Cisco 2511 路由器的背面结构以及 1 转 8 的电缆示意图。

访问路由器其他的形式是使用路由器上的串行接口，如 Cisco 3600 系列，带有一个 8 端口串行模块，或者 Cisco 2522 系列，带有 10 个串行接口。有很多种方法能够将串行电缆与不同的通信接头相连接来实现异步通信。从根本上说，理想组合要求一条 RS-232 串行电缆（DTE 或 DCE），一个 RJ-45 到 DB-25 转换电缆，一条正线或反线。RS232 串行电缆和正线各引脚的准确定义请参考附录 D。RJ-45 到 DB-25 的转换电缆有 3 种形式：DTE M/F，DCE M/F 和 MMOD。Cisco 的 DTE 母口转换头在其侧面标有 Terminal 的字样，MMOD（最常见的一种）则标有 MODEM 的字样。这样的通信接头也可以用在 Cisco 4000 和 7000 系列路由器上面作为控制端口，并且也能够用于控制台服务器与调制解调器的连接。

采用 RS-232 电缆作为控制台服务器的电缆有两种常见并且很廉价的电缆连接方法：

- 电缆连接方法 1——使用 Cisco RS-232 母口到 DB-60 串行电缆，标有 MODEM 字样的 RJ-45 到 DB-25 转换接头和一条 Cisco 全反线。按照顺序将电缆连接好，然后，从配置模式下的接口模式里输入 **physical-layer async** 命令。这样就可以使串口输出的通信方式强制为异步通信的方式。在下一节里，将会学习如何完整地使用这条命令来和 **transport input** 一起用反向 Telnet 会话方式连接到路由器的控制端口。
- 电缆连接方法 2——这个方法和方法 1 相似，只不过使用的电缆和转换接头不一样。该方法使用 Cisco DCE 的 RS-232 到 DB-60 串行电缆，标有 TERMINAL 的 RJ-45 到 DB-25 转换接头和 Cisco 全反线。同样，将电缆按顺序连接好，在配置模式下的串口接口模式下输入 **physical-layer async** 命令。

注释 要想使用这些电缆连接方法，串口必须是支持同/异步口。例如，Cisco 2501 上的端口只是同步口，因此上述的电缆连接方法就不适用了。

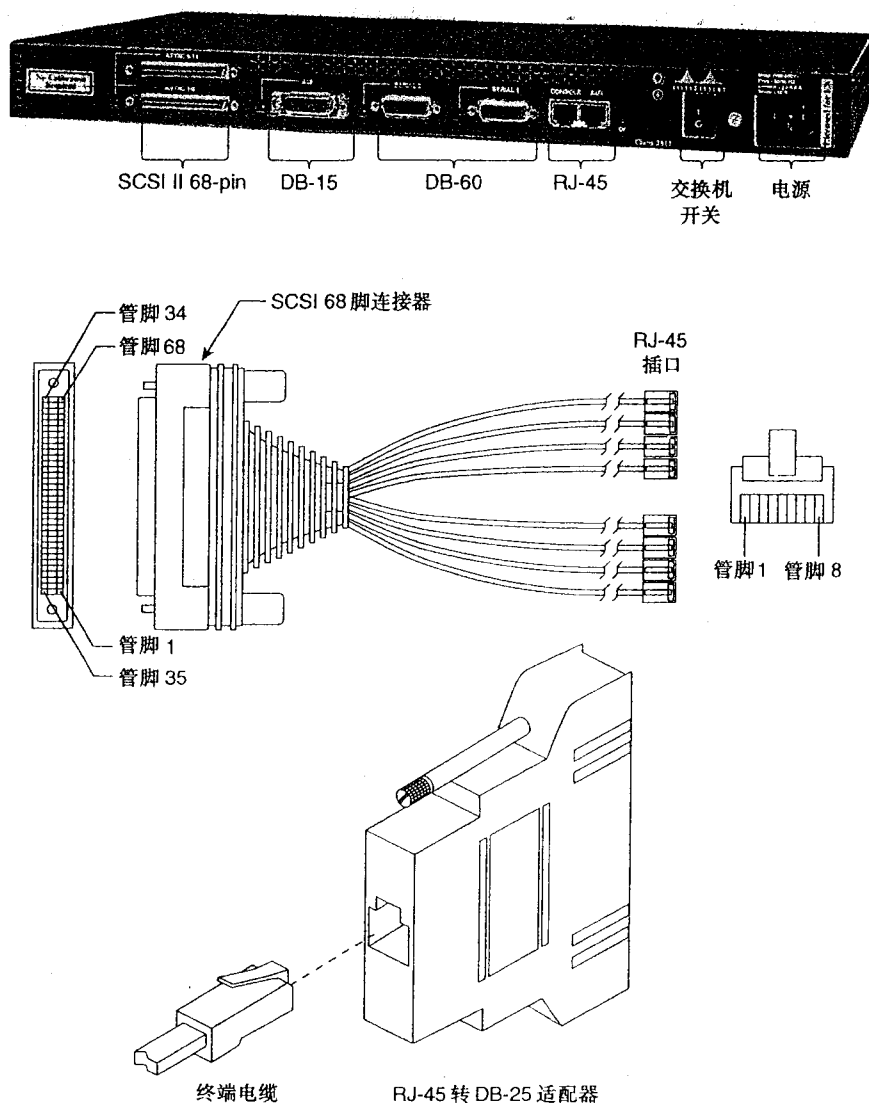


图 1-1 Cisco 2511 路由器与 1 转 8 八爪电缆

表 1-1 列出了 RJ-45 到 DB-25 转换电缆的所有引脚，表 1-2 列出了可选的异步设备的电缆连接方法。

表 1-1 RJ-45 到 DB-25 转换电缆的引脚

RJ-45 转换接头引脚	DTE M/F 引脚 (Terminal)	DCE M/F 引脚	MMOD 引脚 (MODEM)
1	4	5	5
2	20	6	8
3	2	3	3
4	7	7	7

续表

RJ-45 转换接头引脚	DTE M/F 引脚 (Terminal)	DCE M/F 引脚	MMOD 引脚 (MODEM)
5	7	7	7
6	2	2	2
7	20	20	20
8	4	4	4

表 1-2

设备电缆连接选项

访问服务器端口	RJ-45 控制台电缆类型	转换接头	端设备
Console 或 aux	全反线	DTE 引脚	串行电缆
Console 或 aux	直连线	DCE 引脚	串行电缆
Console 或 aux	全反线	MMOD/MODEM	MODEM

访问服务器的功能很多，不仅可以用来配置其他路由器，其串口以及 LAN 端口还可以运行除控制台服务器以外的其他功能。例如，一个控制台服务器可以配置成一个路由发起源或者主干路由器。访问服务器也可以通过拨号的方式进行实验远程配置。本章的后面还会对此相关的内容进行详细的阐述。在“采用路由发起源或主干路由器仿真 LAN”一节中将讨论访问服务器，而在“配置模拟远程访问”一节将讨论仿真拨号。

1.3 建立局域网（LAN）模型

建立 LAN 的模型是建立互连网络模型的一个主要部分。本书中的每一个实验都有专门一节，称为“所需设备”，该小节列出了完成某一实验所必需的基本硬件设备。某些实验可能只需要 1、2 条反接电缆用于连接 2 台路由器，而某些实验却可能需要连接主机，那么在建立该 LAN 模型的时候就应该准备一台集线器或者是交换机。建立和仿真 LAN 有 4 种方法：

- 采用集线器和介质附加设备（MAU）建立 LAN 的模型。
- 采用交换机来建立 LAN 的模型。
- 采用路由发起源或者是主干路由器来仿真 LAN。
- 采用以太网反接电缆来建立 LAN 模型。

建模与仿真

我们用建模和仿真两个术语来描述网络的特性。当使用建模这一术语时，说明网络能够用于传输数据，它是一个大型网络的缩影。当使用仿真这一术语时，说明网络只有一个主机，并且数据不能在网络上传送。网络如果要进行仿真，必须要具备一套路由选择协议。环路接口或者是不检测 keepalive 的以太网接口就是仿真的网络的例子。

1.3.1 采用集线器与 MAU 建立 LAN 的模型

和令牌环。建立第 1 层模型最简单的方法就是采用集线器和 MAU。本书中的大部分实验都需要使用不同种类的多个集线器。物理上来说，网络分段数目是受到实验中路由器上的以太接口或令牌环的接口数目限制的。所使用的集线器的种类无关紧要，重要的是集线器的功能是否齐全以及是否包含两个以上的端口。其中一些集线器是可以控制管理的，能够发送 IPX 的 SAP，这对于 IPX 过滤的测试非常有用。有些时候，最好是在网络中存在功能良好的 MAU。MAU 产生的热量很少，不带电也不产生噪声。用户可以自行决定使用哪一种 MAU。

1.3.2 采用交换机建立 LAN 的模型

使用交换机是仿真 LAN 中最简捷的方法。一个交换机可以配置多个 VLAN。可以把 VLAN 想象成为一个独立的集线器。因此，对于多个 VLAN 来说，不需要用很多的集线器来连接很多的路由器实现，而仅仅需要一台交换机即可。本书的第 2 章“LAN 协议：Catalyst 以太网和令牌环交换机的配置”详细解释了 VLAN 和交换技术。采用交换机仿真 LAN 能够节省空间和电源功耗，当然，交换机价格比集线器贵。

1.3.3 采用路由发起源或主干路由器仿真 LAN

还有两个仿真 LAN 的快捷方法，那就是采用一个环路接口和在路由器的以太接口上使用 **no keepalive** 命令。这两个方法在配置路由发起源或主干路由器的时候非常有用。路由发起源是连接到测试网络中用于收发网络中的路由更新信息的设备。一台配置有许多本地环路地址运行路由选择协议的路由器对由其下游用户而言，就是一个由许多路由器组成的网络。这一功能有利于在实验中实施对仿真网络中的路由过滤和路由视图。

例 1-1 和 1-2 使用 Cisco2501 作为路由发起源。请注意在以太网里加入了 **no keepalive** 命令对以太网进行电子欺骗，使其处于正常工作的状态。还要注意缺省的 **keepalive** 值 **10 seconds** 现在换成了 **not set**。如果在以太网段里屏蔽掉 **keepalive**，就会发现数据包输出、错误输出以及载波丢失数都会增加。重新将以太端口连接到一个真实的集线器或者是交换机的时候，请不要忘记加入 **keepalive** 命令，其缺省值 10 会被自动设置。

例 1-1 配置一个两条路由信息的发起源：只需使用环路接口地址

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int loopback 20
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback20, changed state to up
Router(config-if)#ip address 172.16.16.1 255.255.255.0
Router(config-if)#exit
Router(config)#int loopback 21
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback21, changed state to up
Router(config-if)#ip address 172.16.17.1 255.255.255.0
Router(config-if)#exit
Router(config)#router eigrp 2001
Router(config-router)#network 172.16.0.0
Router(config-router)#^Z
Router#
```

例 1-2 配置一个一条路由信息的发起源：对以太网实施电子欺骗

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int ethernet 0
Router(config-if)#no keepalive
Router(config-if)#^Z
Router#
Router#show int e0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c8d.54ac (bia 0000.0c8d.54ac)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 235/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:18, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  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 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 input packets with dribble condition detected
  21 packets output, 3030 bytes, 0 underruns
  21 output errors, 0 collisions, 2 interface resets
    0 babbles, 0 late collision, 0 deferred
  21 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
Router#
```

1.3.4 利用以太网反接电缆建立 LAN 模型

连接两台以太网主机最常见的方法就是使用以太网反接电缆。以太网反接电缆简单

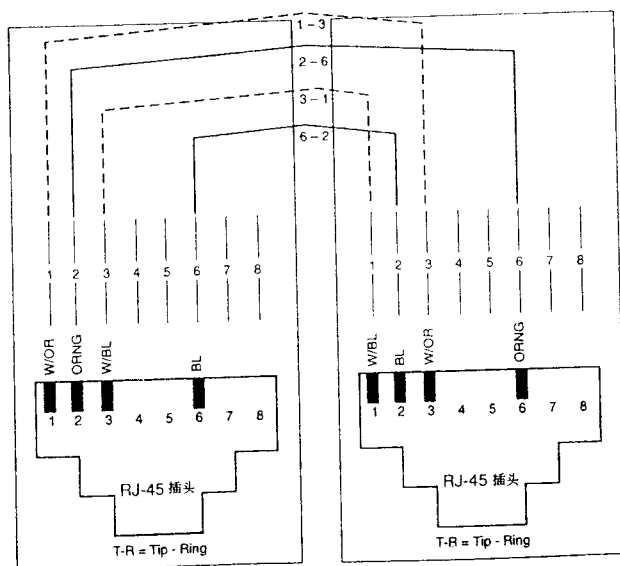


图 1-2 以太网反接电缆的引脚

来说就是一条 RJ-45 到 RJ-45 的反接电缆，从外形上看其引脚是交叉的。这种电缆最明显的缺陷就是，它只能用来连接两台设备。在实验室环境里，可以用这种电缆来连接两台路由器或者是连接一台路由器和一台主机。图 1-2 标明了以太网反接电缆的引脚（pinout）。

1.4 广域网连接方法的仿真

建立网络互联模型的另一个重要部分就是 WAN 的连接。整个设计到目前为止，所涉及的还都是 ISO 第 1 层的问题。建立网络互联模型还应该从第 1 层再往上，这是建立任何网络最符合逻辑的方法。首先，要建立所有设备之间的物理连接，然后配置所有的 LAN 接口，WAN 接口，最后是通过配置路由选择协议实现不同网络连通。这种分层构造网络的思想和方法有利于将来网络的发展和更新，因为如果网络需要更新，一次只需要升级或者是替换某一层网络而保持其他层不变。切记，这里侧重的是物理方面的问题，比如说电缆类型和引脚等。本书第 3 部分，“采用广域网互连局域网”会侧重讲述 ISO 第 2 层的协议。

建立 WAN 的连接模型有 3 种主要的方法，在 Cisco 环境下，可以选用其中一种方法来仿真 WAN。

- 采用特殊的反接电缆来建立 WAN 的模型，这种电缆是用于带有 WAN 接口卡（WIC）的路由器或者是外部信道服务单元/数据服务单元（CSU/DSU）。
- 采用 V.35 DTE—V.35 DCE 电缆或者任何一种 DTE—DCE 配置的串行电缆来建立 WAN 的模型。
- 在 CSU/DSU 上使用环路插头来仿真 WAN。
- 使用一台 Cisco 路由器作为帧中继或 X.25 交换机来建立 WAN 的模型。

注释 毫无疑问，任何一本关于网络的书籍如果没有提到 OSI 模型都是不全面和不完整的。附录 B，“简化的 OSI 参考模型”提供了 OSI 模型的概述。

1.4.1 采用特定反接电缆连接含内置或外置 CSU/DSU 的路由器实现 WAN 的建模

两台带内置或外置 CSU/DSU 的路由器可采用“背对背”的方式连接在一起。这种连接方式作为第 1 层的物理配置能够为许多 WAN 协议提供服务，包括 PPP、HDLC 等等。这种方法是通过采用一种特定的反接电缆来实现的，这种特定电缆是通过一条 4 对 5 类电缆中的部分引脚来实现的。请注意，对于 T1 和 56kbit/s 的 DSU 来说，所需要的反接电缆是有所区别的。一台带 T1 服务单元或者是 CSU/DSU 的路由器必须与另外一台带 T1 服务单元或者是 CSU/DSU 的路由器相连，56kbit/s 服务单元也是类似的。表 1-3 和表 1-4 表明了为 T1 CSU/DSU 以及 56kbit/s CSU/DSU 从 5 类电缆制作这种特殊的反接电缆的方法。

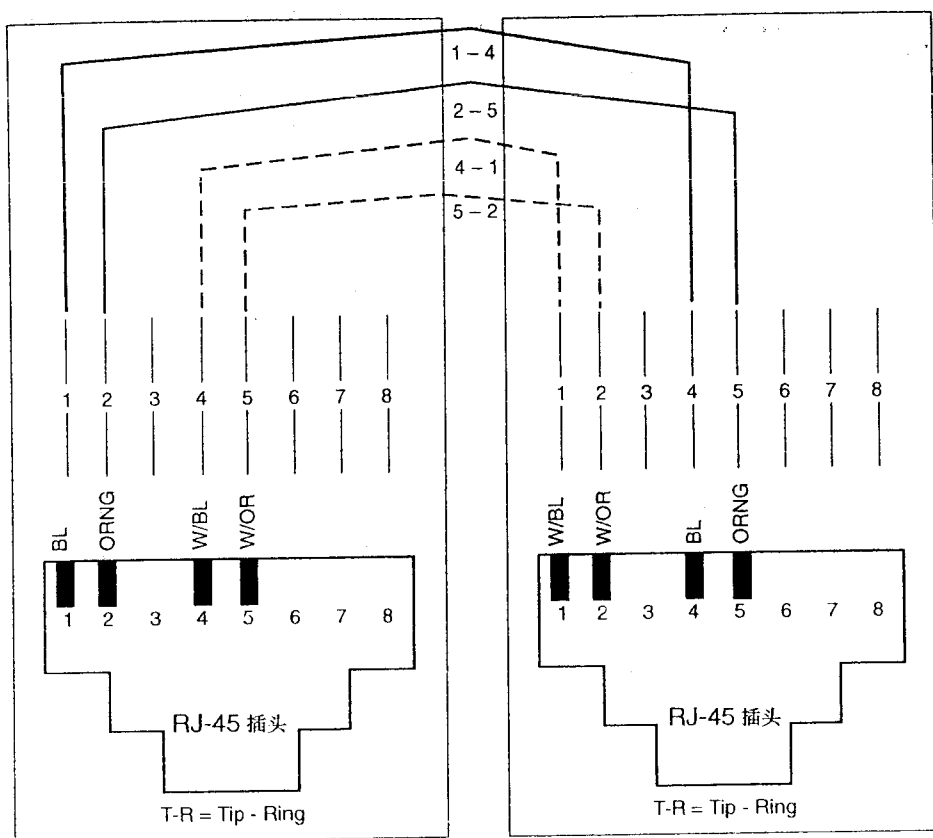


图 1-3 T1 服务模块或 CSU/DSU 所用的特殊反接电缆的引脚示意图

1.4.2 采用 V.35 DTE-DCE 电缆建立 WAN 的模型

在路由器之间建立第 1 层连接的最常见的方法就是使用母口的 V.35 DCE 和公口的 V.35DTE 串行电缆相连。任何背对背结构中最关键的因素就是一端要提供时钟，即连接 DCE 的一端。如果需要设置一个接口的时钟，只需要加入命令 **clock rate [value]** 就可以了。例 1-3 就演示了如何将一个串行接口的时钟设置为 64kbit/s。

例 1-3 在 DCE 接口上配置时钟

```
frame_relay_switch(config)#int serial 5
frame_relay_switch(config-if)#clockrate 64000
frame_relay_switch(config-if)#^Z
frame_relay_switch#
```

一定要保证电缆的确是用 V.35 DTE 电缆与 V.35 DCE 电缆相连的。DCE 或 DTE 电缆的类型公口/母口 (male/female) 无关紧要，但是确保将 DCE 的一端与 DTE 的一端直联，然后像例 1-3 中所示的那样设置时钟速率。图 1-5 是一些标准 Cisco 连接器的示意图，包括常见的 V.35、RS-232 接口，也包括 EIA613-HSSI 接口。

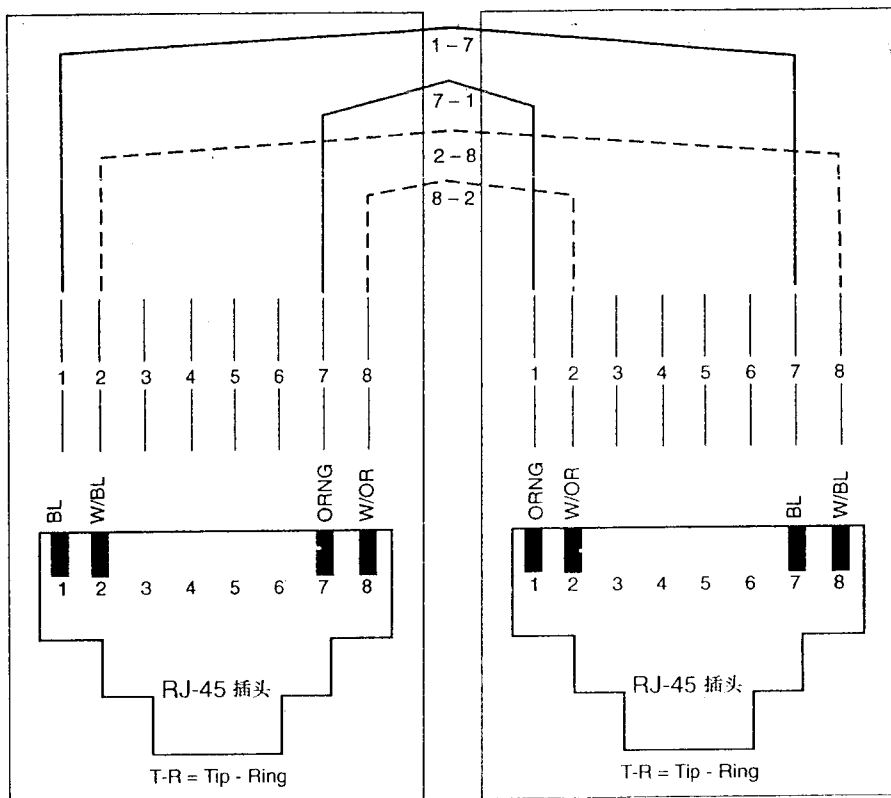


图 1-4 56kb/s 服务模块或 CSU/DSU 所用的反接电缆的引脚示意图

这些电缆都可以从 Cisco 公司购买，V.35 公口 DTE 电缆产品编号是 CAB-V35MT，母口 DCE 电缆编号为 CAB-V35FC。其他公司也提供相关的串行电缆和反接电缆。

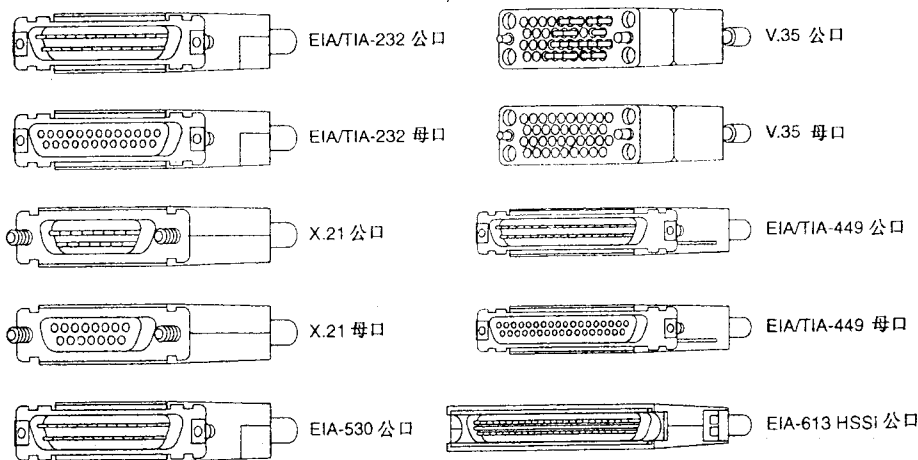


图 1-5 Cisco 路由器常见的电缆接口

当电缆背对背互连时，很难区分哪一端是 DCE 的电缆。如果在现场或者是在远程工作，无法直接到实验室里的设备，那么，怎样才能判断路由器哪一端连接了 DCE 电缆呢？命令子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责

show controller 就用于显示连接的电缆类型，用于表明电缆是 DCE 的还是 DTE 的。例 1-4 给出了在 Cisco 2501 上两个接口的例子。使用 **show controller** 命令能够判断接口类型。

例 1-4 使用 show controller 的例子

```
Router#show controller serial 0
HD unit 0, idb = 0xCED94, driver structure at 0xD3B18
buffer size 1524 HD unit 0, V.35 DTE cable
cpb = 0xE2, eda = 0x4140, cda = 0x4000
RX ring with 16 entries at 0xE24000
00 bd_ptr=0x4000 pak=0x0D6F0 ds=0xE2DDB0 status=80 pak_size=0
***text omitted***

Router#show controller serial 1
HD unit 1, idb = 0xD7788, driver structure at 0xDC508
buffer size 1524 HD unit 1, RS-232 DCE cable
cpb = 0xE3, eda = 0x2140, cda = 0x2000
RX ring with 16 entries at 0xE32000
00 bd_ptr=0x2000 pak=0x0DF0E4 ds=0xE3C468 status=00 pak_size=0
***text omitted***
```

串口 0 是使用的 V.35 DTE 电缆，串口 1 是使用的 RS-232 DCE 电缆。其他串行电缆，像 RS-232，也可以采用背对背的连接，只要把 DCE 与 DTE 连接在一起就可以了。要注意的是，不同类型的电缆都有一定的速率限制。例如在 RS-232 电缆上仿真 T1 的速率是不可能的。在实验室环境下要想实现最大的灵活性，应该尽可能地使用 V.35 电缆。

如果需要方便快速地改变实验室的环境，或是希望串行电缆连到某一个路由器上，或是又想将此串行电缆连到另外一台路由器上。在这种情况下，最好能使用接插板。在大实验室里，V.35 接插板是很常见的，也很方便。大多数的 V.35 接插板，背面的最顶端都含一个母口的 DTE 端口，而在其背面的中间部位含一个 V.35 公口 DCE 端口。当路由器与这类端口相连的时候，DTE 电缆插入 DTE 端口中而 DCE 电缆插入 DCE 端口中。接插板的正面是一些小的配线端口，每一个 DTE 和 DCE 端口的前面都有一个这样的端口。黑色的配线电缆能够将每个 DTE 端口配到相应的 DCE 端口，从而也就能将两台路由器互联。

如果需要改变这一配置，只需把配线电缆接到另外一个配线端口上。这样使用接插板能够快捷地改变许多个串口连接的物理配置。在接插板的正面应该标明哪一个是 DCE 端口，哪一个是 DTE 端口，这在处理物理层故障的时候尤为重要。图 1-6 给出了一个 V.35 接插板的实例图。

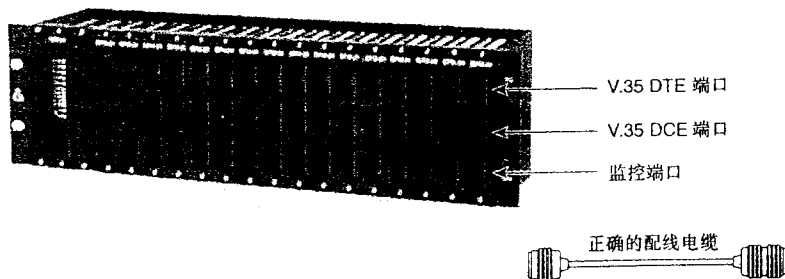


图 1-6 V.35 接插板

大多数接插板还有一个底部端口，就在 DCE 端口的下面。这个端口是用来连接在线监控器或是数据监测器的。

1.4.3 采用 HDLC 和 CSU/DSU 上的环路插头来仿真 WAN

另外一种仿真 WAN 接口的方法就是在模型中采用 HDLC 协议，并且在模型的 CSU/DSU 上加入环路插头。另外，允许在接口里配置第 3 层地址，如 IP 地址或 IPX 地址，这样，该接口就会响应 ping，而且在路由表中显示。在使用环路插头的时候，第 2 层的数据封装也就一定要设置成 HDLC 协议。内置 CSU/DSU 或 WAN 接口卡（WIC）的路由器，即使是外置 CSU/DSU 的路由器，也可以将环路插头插进它们的 RJ-45 插座里，从而使得 WAN 接口可以开始工作。例 1-5 表示的是在一台安装了一块 T1 WIC 和一个环路插头的 Cisco2524 使用 show interface 命令产生的输出。

例 1-5 在一台装了一块 T1 WIC 和一个环路插头的 Cisco2524 上使用 show interface 命令

```
router# show interfaces serial 1
Serial1 is up, line protocol is up (looped)
Hardware is HD64570 with FT1 CSU/DSU
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 00:00:02, output 00:00:02, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations 0/1/256 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  2537 packets input, 148733 bytes, 0 no buffer
    Received 2537 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  2537 packets output, 148733 bytes, 0 underruns
    0 output errors, 0 collisions, 1 interface resets
    0 output buffer failures, 0 output buffers swapped out
    1 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
router#
```

注释 在现场应用中环路插头非常有用。如果认为某台 CSU/DSU 或是 WIC 工作不正常，安装一个环路插头就能够快捷地对 CSU/DSU 的物理层进行测试。安装了环路插头之后，show interface 命令应该显示该接口是 line up, protocol up and (looped)。

环路插头很容易制作。用简单的电缆工具就可以制作，当然也可以去厂商那里定做，只要给厂商提供如图 1-7 和 1-8 所示的图纸即可。图 1-7 所示的是一个 RJ-45 的 56kbit/s 环路插头的引脚定义，而图 1-8 所示则是一个 RJ-45 的 T1 或 1.544Mbit/s 的环路插头的引脚定义图。

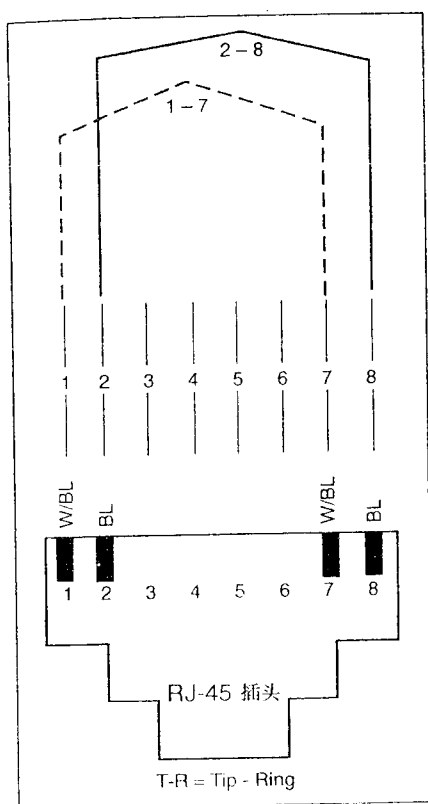


图 1-7 RJ-45 56kb/s 环路插头的引脚定义图

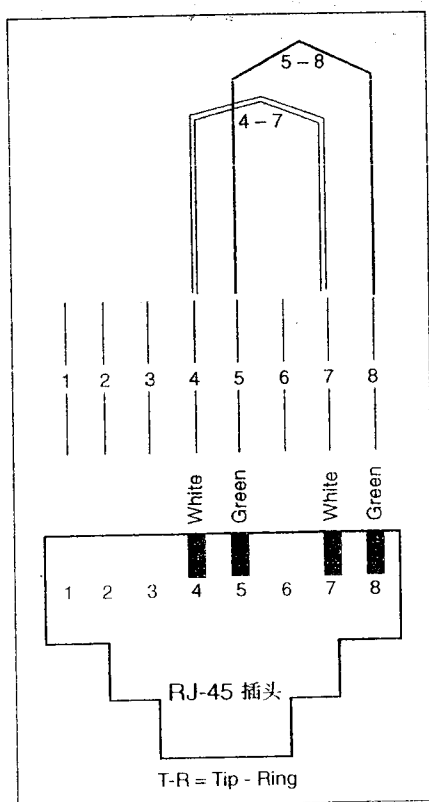


图 1-8 RJ-45 T1 环路插头的引脚定义图

1.4.4 将一台 Cisco 路由器作为帧中继或 X.25 交换机来建立 WAN 的模型

建立 WAN 模型的第 3 种方法是將 Cisco 路由器設置成帧中继或者是 X.25 交换机。任何一台配有 11.0 版本或更高的 Cisco IOS 软件并且至少含两个串口的 Cisco 路由器都可以設置成帧中继或者是 X.25 交换机。当配置成帧中继交换机时，该路由器实际上是作为收发帧中继的本地管理接口（LMI），甚至还能够將其設置成使用帧中继交换机的网络——网络接口（NNI），来建立 WAN 模型。无论何时，至少需要两个接口，因为交换机首先是一台 DCE 设备，需要两台路由器作为 DTE 设备。而由于帧中继交换机也是一台纯粹的 DCE 设备，故它也需要 DCE 串行电缆来与其他设备相连接。

Cisco 2522 和 2523 系列路由器有足够的端口与控制台设备相连接，也可以作为帧中继交换机。Cisco 2522 提供两个高速同步串口以及 8 个低速同/异步串口，并且还含一个設置成 RJ-45 或 AUI 的以太网端口以及一个 RJ-45 形式的 ISDN BRI U 接口。Cisco 2523 和 Cisco 2522 是基本一样的，除此一点，那就是 2523 不含以太网端口，取而代之的是一个 RJ-45 的令牌环接口。任何一台具备多串口的 Cisco 路由器都可以作为帧中继交换机或 X.25 交换机。帧中继交换功能与其他路由器诸如 IP 路由之类的常规路由器功能之间是相互独立的。这样，该路由

器就不仅可以作为帧中继交换机使用，同时也可作为其他功能的路由设备或是路由发起源。本章的“帧中继交换机的配置”一节将会详细讲述这方面的软件设置。

技巧 在网络学习和网络职业生涯中，通常会遇到许多的术语和缩略语。有时，很难记住所有的这些规则，就像是通信连接的 DCE 一方需要时钟这条规则一样。一个记住这些规则的方法就是单词联想法。比如说，我们知道一条电缆的两端是 DCE 和 DTE。C 是 DCE 与 DTE 不同的地方，而 C 就代表时钟。这样，DCE 一端就是需要设置时钟的一端。

可能有人会问，“ATM 是什么？那是一个 LAN 与 WAN 的协议——该在什么地方用呢？”异步传输模式（ATM）最初的时候被人们戏称为网络里的“鸭子”，鸭子既会游泳，又会飞，还会走路，尽管走得不是那么好，游得还可以，但飞得倒是很快。ATM 能够传输语音、数据以及视频信息，当然，其中一些内容的表现比另外一些更好。在决定是否将 ATM 加入到本书的介绍内容中来的时候，棘手的问题就是 ATM 的归类。ATM 是一种结合 LAN 和 WAN 的技术，并且专业化程度很高，在第 8 章“WAN 协议与技术：异步传输模式（ATM）”里将会对其进行阐述和讲解。

1.5 实验室中路由器，Cisco IOS 软件以及内存的要求

贯穿本书的每一个实验都有各自的硬件与软件要求。像前面提到的，在每个实验之前，都有一节是“所需设备”，列出了要完成该实验所需要设备的最低要求。一些实验很简单，只需要 2 至 3 个配有 IP 路由功能软件的路由器即可，而另外一些实验，像设置 SNA 的实验，就非常复杂，需要成套的功能部件以及较多的路由器、集线器等设备，就显得比较困难了。

不幸的是，一些公司没有认识到“熟能生巧”这句话的价值，他们没有从预算中拨出钱来给实验室购买设备，导致了他们的工程师们失去了提高自身的技术水平的机会。遗憾的是，这些公司竟然还选择以自己的或是其客户的网络为主。像 HP、Comdisco Inc.、IBM 这样的公司之所以能够一直保持着是行业的佼佼者，原因之一就是他们正确地权衡和分析了以高素质的工程师去仿真和部署复杂的互联网络所需的成本与所能带来的收益之间的关系。

我本人也理解搭建网络所需设备的困难，因此，在本书的实验中已经尽可能的使用最少数量的路由器以及集线器。并且书中的内容相互之间尽量独立，这样，如果无法找到令牌环/以太网路由器或交换机，仍然可以做书中大部分的实验。比如说，要完成转换桥接的实验，需要在同一路由器上有一个令牌环接口和一个以太网接口。由于很多 Cisco 路由器都能容纳不同类型的多个 LAN 段，那就不用对特定的路由器的具体型号纠缠过深。在计划购买新的路由器的时候，一条很好的原则就是：路由器越模块化，将来用于设计多种模型时候的灵活性就会越大。购买模块化的路由器另外一个好处就是：这类路由器多数都可以通过安装新的网络模块来实现升级，能节省开销。

注释 Comdisco 的实验室主要都是用 Cisco Catalyst 5500s、Cisco 2500s、4500s 和 3600 系列的路由器来建立网络的模型。Cisco 3600 系列路由器几乎能够提供所有的网络功能要求，像 ATM、吉比特以太网、IP 或帧中继语音传输，还包括 VPN 技术。

网络操作系统 (IOS) 的软件版本、DRAM 存储器和 FLASH 存储器的问题了³。这3个因素之间是密切相关的。所采用的协议和功能部件集的正常运行决定了所需要的 DRAM 的大小。Cisco IOS 的功能部件一般存储在 FLASH 中。因此，需要的功能和协议越多，需要的 DRAM 和 FLASH 存储空间就越多。本书中的实验运用了大部分的主路由选择协议和功能部件集。要想很好地包含这些内容，应该在所有的路由器上安装“增强型企业版”的功能部件集。Cisco IOS 的软件版本至少应该在 11.2.x 或 12.0.x 以上。若是特别为 CCIE 实验考试而做准备，应该在所有的路由器上安装至少 12.0 以上版本的 IOS。如果实验要求的 IOS 功能不只是 IP 路由，那会在实验要求里面说明。

注释 要想查出某一特定 Cisco IOS 和功能部件集所需要的最小存储器值，可以在 Cisco 的网站 www.cisco.com 上查询。

1.6 测试主机与数据仿真

所有网络的最终目的都是为了将数据从一端传送到另一端。如果没有数据可传，那网络也就没有了存在的价值。为确保每个模型的正常工作，需要测试数据。网络模型也不例外。没有测试数据，就难以对很多网络的特性进行测试。像远程源路由桥接 (RSRB)，进行数据的收发才能激活 RSRB 的功能。如果数据发送之前所有的电路都是没有工作的，那么数据链路交换 (DLSw) 对等关系就无法正常的连接。因此，要想正确地测试网络模型，就必须仿真各种不同的实验数据。当然在测试一些很复杂的协议 (如 SNA) 的时候，如果实验室里不具备小型机，那在仿真实验数据的时候就会有一些困难了。

幸运的是，Microsoft 的 Windows 95/98/2000 操作系统自带的 3 个主要协议对测试网络非常有用。Windows 95/98/2000 都自带 TCP/IP, IPX/SPX 和 NetBEUI 这 3 个协议。这 3 个协议能够对很多 Cisco IOS 的特性进行测试。

例如，两台在 Windows 工作站环境下运行着 NetBEUI 的主机能够用来测试一个互联网络中的 DLSW 对等体 (Peer)。浏览“网上邻居”时会强制产生一个全路由探测帧。用户在控制面板/网络/标识下的“计算机名”一栏里填入的名称就是使用 `show dlsw reachability` 命令的结果。另一个例子是，在一个 RSRB 环境里，这样的探测帧足以使远程源桥接开始工作，使之与另一个远程源桥之间建立连接。

安装了 TCP/IP 之后就能够使许多像 FTP, TFTP, DHCP 这样的共享公用程序用于测试。安装工作站之后，查看那些正在工作的过滤器的时候会发现它们就像在真实网络环境中一样工作。所有这些以 IP 为基础的公用程序都是共享软件，都可以在线找到。

注释 1993 年的时候，我第一次安装了家庭网络，同一天，发行了电脑游戏 DOOM。3D 游戏 DOOM 支持 IPX 协议下的多人网络对战模式。我们攒足了钱之后，几个小时后就用同轴电缆线将不同公寓里的电脑互联。随后，又开始设置基于 IPX 协议的 Netware-Lite 网络。在那个 DOS 的时代和讨厌的 640 K 限制之下，对这个小小的网络，我们都非常自豪。接下来的日子里，我们就通宵达旦沉浸在了 DOOM 的 3D 世界里。

直至今日，DOOM 与其后来者们仍然保留在那些网络管理者们的记忆里。一些带宽管理产品，如 Packeteer 发布的，甚至专门为 DOOM 预留了一条数据流通道 (这主要用来保证

子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责

DOOM 玩家们获得足够的带宽，而不至于因某个人的打印工作而导致其游戏数据传输的速度变慢)。无论如何，电脑游戏仍然是用来测试 IP 和 IPX 协议的传输速度和测试网络管理者对其控制情况的比较好的工具。无论是在实验室还是在家里，在新的网络上玩最新电脑游戏既能带来乐趣也能增加玩家在网络方面的知识。而这些网络专业知识能够让朋友们大吃一惊。

1.7 建立网络互联模型框架——关键组件的配置

本书中要构造的网络模型都是从一个类似的框架开始的。大部分的网络模型都包含有 1 个或多个 LAN 和 WAN，当然，这种情况下就需要路由器和集线器了。此外，还需要一台设备以便在本地或者是远程访问该模型，还需要通过一系列应用程序来运行和测试这一模型。这样，大部分网络模型的框架就包括：路由器、集线器、帧中继交换机，访问服务器以及一些工作站。从这样的框架出发，只需要对网络拓扑结构作一些轻微的改变就可以设计搭建许多不同的网络模型。在构造初始框架时，可以参照下面的步骤：

第 1 步 获取对模型设备的特权级访问

包括使用和修改 16 位的引导寄存器的值以获取对路由器的特权级访问。

第 2 步 按照模型的要求升级 Cisco IOS 软件

将新的 IOS 拷贝到 FLASH 存储器里去。

第 3 步 对本地与远程的模型访问进行配置

包括访问服务器的配置和模拟拨号访问的配置。

第 4 步 对 LAN, WAN 的设备配置

每个模型对 LAN 与 WAN 的配置要求会有所区别，不同之处只是部分电缆的重新接线。因此，主要的精力还是应该放在帧中继交换机及其永久虚拟电路 (PVC) 的初始配置问题上。

第 5 步 测试应用程序与测试网络的配置

这包括设置 Microsoft Windows 95/98/2000 网络以及配置 TCP/IP, IPX, NetBEUI 之类的网络协议。还要学会如何使用路由起源。

1.7.1 获取特权访问：16 位的引导寄存器

我个人认为 Cisco 路由器和交换机最机密之处就是 16 位的引导寄存器。几乎每一种 Cisco 平台上都存在这个寄存器，每一种又不尽相同。例如，20 世纪 90 年代早期推出的 AGS 系列路由器上就已经存在了这样一个寄存器了，不过当时采用跳线来设置的。在 2001 年 Cisco 的 Catalyst 型交换机里，同样有着这样一个寄存器。基本上，所有的 Cisco 路由器上寄存器都是一样，有时候它是可能被 CONFREG 程序给屏蔽。

另外一个使用引导寄存器的例子就是在恢复密码的时候。在恢复密码的时候，实际上是将引导寄存器的第 6 位取反，使寄存器的值从 0x2102 变为 0x2142，第 6 位置为 1 的时候，NVRAM 在引导的时候就会被忽略掉。这应该是该寄存器最常见的用法，其他的一些用法如下：

- 恢复丢失的密码。

- 使控制台 BREAK 键有效或失效。
- 允许在引导程序（ROM 监控器）提示符下使用 **B** 命令进行手工引导。
- 改变路由器引导设置，允许从内存或 ROM 引导。
- 在 ROM 监控器模式下进行维护测试。
- 将文件映像加载到内存。
- 允许路由器永久失效。

由于引导寄存器代表了路由器的“核心”，因此只了解其第 6 位的作用是不够的，应该对该寄存器所有的位的作用都应加以了解。

如果使用 **show version** 命令，该命令会将引导寄存器的值显示在命令结果的最下方。例 1-6 就是 **show version** 命令使用时的显示结果。

例 1-6 引导寄存器设为从 ROM 引导的 0x2101 的值时 show version 命令的结果

```
router(boot)#show version
Cisco Internetwork Operating System Software
IOS (tm) 3000 Bootstrap Software (IGS-RXBOOT), Version 10.2(8a), RELEASE SOFTWARE
E (fc1)
Copyright (c) 1986-1995 by cisco Systems, Inc.
Compiled Tue 24-Oct-95 15:46 by mkamson
Image text-base: 0x01020000, data-base: 0x00001000

ROM: System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE

router uptime is 34 minutes
System restarted by power-on
Running default software

cisco 2500 (68030) processor (revision L) with 14332K/2048K bytes of memory.
Processor board serial number 03071163 with hardware revision 00000000
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
ISDN software, Version 1.0.
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
1 ISDN Basic Rate interface.
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read/Write)

Configuration register is 0x2101

router(boot)#
```

这个引导寄存器的格式为：最高有效位在最右边，就像图 1-9 所示的那样。从该图也能看出 Cisco 路由器里，该寄存器的缺省值 0x2102 时如何得出来的。

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0
2				1				0				2			

图 1-9 16 位引导寄存器的缺省值

简单地分析该寄存器缺省值的设置，会发现其第 1、8 和 13 位都置为 1（打开相应的功

中存在 IOS 时就从 FLASH 引导。将第 4 到 7 位置 0 使得路由器正常情况下从 NVRAM 中引导配置，保持标志字的值不变，如果置位“全 1”表示设置为网络广播形式。第 8 位的值强制 break 键失效。剩下的寄存器位将网络广播设置为全 1，控制台波特率为 9600 bit/s，并且决定了路由器响应网络引导失败的方式。如前所述，最常见的用法是将其第 6 位的值取反，使得路由器忽略 NVRAM 中保存的引导设置信息。密码恢复时使用了这一方法。

表 1-3 详细列出了整个寄存器的结构与设置。阅读本书关于引导寄存器的详细描述时请参考该表格。

表 1-3 16 位引导寄存器的完整结构及其缺省设置

位	含 义	缺省设置
0-3	引导字段 0x0 = 从 ROM 监控模式引导 0x1 = 从主板的 ROM 引导，或者如果简化版的 IOS 存在，就进入 boot 模式 0x2-0xF 按如下操作（按操作顺序排列） 如果 FLASH 里有可用的 IOS 文件存在，则从 FLASH 引导 按配置文件 boot system 命令定制的顺序引导 根据寄存器值产生网络引导时系统映像文件的名称	0010
4	快速引导：通过配置文件中的 boot system 命令强制载入方式	0
5	高速控制台：1=控制台速度 19.2kbit/s 或 38.4kbit/s，根据第 11 和 12 位值选择	0
6	忽略引导文件：1=忽略 NVRAM	0
7	OEM 位：1=使引导时的 banner 失效	0
8	暂停：1=失效	1
9	未使用	0
10	网络引导广播格式： 置 1=处理器使用全 0 广播	0
11-12	控制台波特率 第 5 位=1 第 11 位=1 第 12 位=0 控制台波特率=38 400 第 5 位=1 第 11 位=0 第 12 位=0 控制台波特率=19 200 第 5 位=0	0

续表		
位	含 义	缺省设置
11-12	第 11 位=0 第 12 位=0 控制台波特率=9 600 ----- 第 5 位=0 第 11 位=0 第 12 位=1 控制台波特率=4 800 ----- 第 5 位=0 第 11 位=1 第 12 位=1 控制台波特率=2 400 ----- 第 5 位=0 第 11 位=1 第 12 位=0 控制台波特率=1 200	0
13	对网络引导失败的响应：1=网络引导失败后从 ROM 引导，0=继续网络引导	1
14	网络引导子网广播： 1=强制子网广播	0
15	使用诊断消息：1=忽略 NVRAM，显示诊断消息	0

1. 引导字段（第 0 到 3 位）

引导字段控制着路由器的引导。这一字段即右 4 位。如果该字段设为 0x0，路由器就会引导到 ROM 监控模式。例如，将寄存器的值设为 0x2100 会使路由器引导至 ROM 监控模式。如果设为 0x1，路由器就会从 ROM 引导，该 ROM 里可能会有一个完整的 IOS，如 7000 系列，也可能是 IOS 的一个子集，如 2500 系列。引导模式下的提示符是路由器主机名后面跟一个 (boot)。如果将该值设为 0x2 到 0xF 之间的数，并且在配置文件里有一条有效的系统引导命令，路由器就会按照这个配置来引导其系统软件。如果引导字段设置成其他值，路由器会根据该值产生一个用于网络引导的缺省引导文件名，产生这一文件名是路由器自动设置过程中的一步。产生该文件名时，路由器是以 cisco 开头，后跟一个数字，一个破折号以及处理器类型名。引导字段设为 0x1 的 Cisco 4000 试着载入文件名为 Cisco2-4000 的 TFTP 协议的文件。表 1-4 列出了设置引导字段位后处理器加载的缺省引导文件名或其他的处理方式。XXXX 代表处理器类型，比如，对 Cisco4000 来说，xxxx=4000。

表 1-4 缺省引导文件名

行为/文件名	第 3 位	第 2 位	第 1 位	第 0 位
引导至 ROM 监控模式	0	0	0	0

续表

行为/文件名	第 3 位	第 2 位	第 1 位	第 0 位
从 ROM 引导	0	0	0	1
Cisco2-xxxx	0	0	1	0
Cisco3-xxxx	0	0	1	1
Cisco4-xxxx	0	10	0	0
Cisco5-xxxx	0	1	0	1
Cisco6-xxxx	0	1	1	0
Cisco7-xxxx	0	1	1	1
Cisco10-xxxx	1	0	0	0
Cisco11-xxxx	1	0	0	1
Cisco12-xxxx	1	0	1	0
Cisco13-xxxx	1	0	1	1
Cisco14-xxxx	1	1	0	0
Cisco15-xxxx	1	1	0	1
Cisco16-xxxx	1	1	1	0
Cisco17-xxxx	1	1	1	1

2. 快速引导/强制引导位 (第 4 位)

将这个比特设为 1 将强制路由器载入由配置文件中 **boot system flash** 命令设置的 Cisco IOS。如果没有与该命令设置的文件相匹配的 Cisco IOS，路由器将会进入 boot 模式。例如，在配置文件中加入 **boot system flash c2500-js56-l.120-3.bin** 命令行，路由器将会试着在 FLASH 中寻找 c2500-js56-l.120-3.bin 文件，如果找不到该文件，路由器将会进入 boot 模式。

3. 高速控制台位 (第 5 位)

将第 5 位设置为 1，使之与第 11、12 位一起工作，使得路由器可以与速率高于 9 600 bit/s 的控制台进行通信。这一位设为 1 后，可以将控制台的端口速率设为 19 200 和 38 400 bit/s。表 1-6 列出了完整的与第 10 和 11 位配合工作的情况。

警告 第 5 位的作用没有正式公布是有原因的。控制台端口对路由器工作和故障排除的作用是非常关键的。数据传输速率越高，网络连接的灵敏度要求就越高，就越不容易在高通信速率下将路由器与控制台相连接。这种情况下，如果没有使用 Telnet 访问方式或是其他一些“后门”方式的权限，结果是非常可怕的。从以 19 200 或 38 400 bit/s 相比 9 600bit/s 速率工作的控制台端口带来的一点好处是微不足道的。请记住，该位的使用是用于路由器键入与配置，而与控制台通信没有必要使用很高的速率。使用这一位时一定要非常小心。

4. 忽略 NVRAM 位 (第 6 位)

该位设为 1 时，强制路由器忽略 NVRAM 里的引导配置文件 (*startup-config*)。忽略 NVRAM 时，从本质上来说，就是忽略了引导配置文件。用 **show** 命令仍然可以查看引导配

置文件的内容，但配置信息已经不存在于正在运行着的配置文件中了。密码恢复时，也需要把这一位置 1。

5. OEM 位（第 7 位）

这一位是用于 OEM 版本的路由器。该位设为 1 时，Cisco Systems 的标志会被忽略掉。如果 IOS 用加密软件进行过加密，则仍然显示加密警告。

6. 键位（第 8 位）

该位置为 1 将屏蔽暂停键。如果这一位设为 0，在路由器正常运行期间的任何时候（当然不包括系统的引导期间），可以通过按下单键使得操作系统暂停。这个设置影响很大，不要改动缺省值。屏蔽该功能（缺省屏蔽）在初始化起始 60 秒之内并不会影响该暂停键的功能，这段时间里，暂停键仍然可以使路由器暂停下来。

7. 保留位（第 9 位）

该位暂未使用。

8. 网络引导广播格式位（第 10 位和第 14 位）

第 10 和 14 位是设置路由器和交换机处理主机或子网广播的方式。缺省的网络广播地址是主机或子网全 1 的目的地址。改变这些位能够向后兼容许多老版本的 UNIX 主机，比如说，Berkley UNIX 4.2BSD。现在许多的 IP 协议里都用全 1 的地址作为网络广播地址，因此可能一直都不用改变这些设置。表 1-5 解释了第 10 和 14 位的用法。

表 1-5 网络广播地址控制位：第 10 和 14 位的设置

第 14 位	第 10 位	地址（<网络><主机>）
0	0	<全 1><全 1>
0	1	<全 0><全 1>
1	0	<网络号><全 1>
1	1	<网络号><全 0>

9. 系统控制控制台波特率设置位（第 5、11 和 12 位）

第 5、11 和 12 位用来设置控制台的波特率（bit/s）。路由器的出厂波特率都是设为 9600bit/s，其第 5、11 和 12 位都置为 0。表 1-6 列出了相应的波特率设置情况。比方说，为了增加控制台端口的波特率，可以将寄存器设为 0x2122，相应的波特率为 19 200 bit/s。

表 1-6 系统控制台波特率的设置情况

第 5 位	第 11 位	第 12 位	控制台波特率
1	1	0	38,400bit/s
1	0	0	19,200bit/s
0	0	0	9600bit/s
0	0	1	4800bit/s
0	1	0	1200bit/s
0	0	0	2400bit/s

10. 网络引导失败响应位 (第 13 位)

将第 13 位设为 1 将会使 cisco 路由器在 5 次网络引导失败之后自动地从缺省位置载入 Cisco IOS。该位的缺省值为 1，这就是为什么大多数路由器的寄存器的值以 2 开始的原因。如果该位设为 0，那么网络引导失败时，路由器只会不停的重新引导而不会自动从 ROM 位置去寻找 Cisco IOS 载入。

11. 显示出厂诊断位 (第 15 位)

将第 15 位设为 1 将使路由器显示出厂设置诊断信息，也将强制忽略 NVRAM。要显示这些诊断信息，可将寄存器值设为 0xA102。A 是将第 15 和 13 位设为了 1，初始化时强制显示诊断信息。

12. 对引导过程的理解

下面这一节的内容在所有新的 Cisco 路由器随机文档 CD 中都能找到。尽管如此，仍有必要再次强调这部分内容的重要性。

当一台路由器加电开机或者是重新引导时，会执行下列过程：

- ROM 监控模式的初始化。
- ROM 监控模式核对配置寄存器的引导字段（寄存器的低 4 位）。

如果引导字段为 0x0，系统不引导 IOS，而是在 ROM 监控模式的提示符下等待用户手工操作。

如果引导字段为 0x1，在 ROM 监控模式下系统会引导简版 IOS 映像文件（在某些平台上，引导简版 IOS 映像文件是用环境变量 BOOTLDR 来指定的）。

如果引导字段值在 0x2 到 0xF 之间，ROM 监控模式会去加载已经在配置文件中指定，或者是用环境变量 BOOT 指定的第一个可用的 IOS 映像文件。

- 当引导字段值为 0x2 到 0xF 之间，路由器会逐条执行命令，直到加载了可用的映像文件。如果寄存器的第 13 位置为 1，每条命令只会执行一次。如果第 13 位置为 0，**Boot system** 命令就使得最多尝试 5 次和网络服务器的连接。两条连续的执行尝试连接之间的时间间隔为 2、4、16、256 以及 300s。如果找不到可用的映像文件，则按如下顺序操作：

如果系统配置文件里所有的引导命令都指定从一个网络服务器读取映像文件来引导系统，而所有的命令执行都失败了，那么系统就会尝试用 FLASH 中的一个合法的文件来引导。

如果寄存器中的缺省由 ROM 引导选项置为 1，路由器就会加载引导映像文件（在 ROM 中的映像文件或由环境变量 BOOTLDR 来指定）。

如果寄存器中的缺省由 ROM 引导选项没有置为 1，那么系统就会在 ROM 监控模式提示符下等待用户干涉，这个时候需要用手动的方法引导系统。

如果没有找到可用的系统映像文件，路由器不会工作。必须通过直接相连的控制台端口重新进行配置。

- 在 FLASH 中寻找一份可引导的文件时。

系统会在 FLASH 中寻找文件。如果没有指定文件名，系统会在整个 FLASH 中寻找一个可以用来引导的文件而不仅仅尝试第一个文件。

该文件是否可以用于引导：

- FLASH 映像文件：系统软件会判断它是否以正确的执行地址被加载。
- RAM 映像文件：系统软件会判断系统是不是有足够的RAM空间来运行该映像文件。

对于双处理器卡或者是双 FLASH 卡的平台，如 7000 系列或 Catalyst RSM，上述的过程会有所改变。图 1-10 显示了大多数的平台（除了上面提过的特例）上的这一复杂过程。

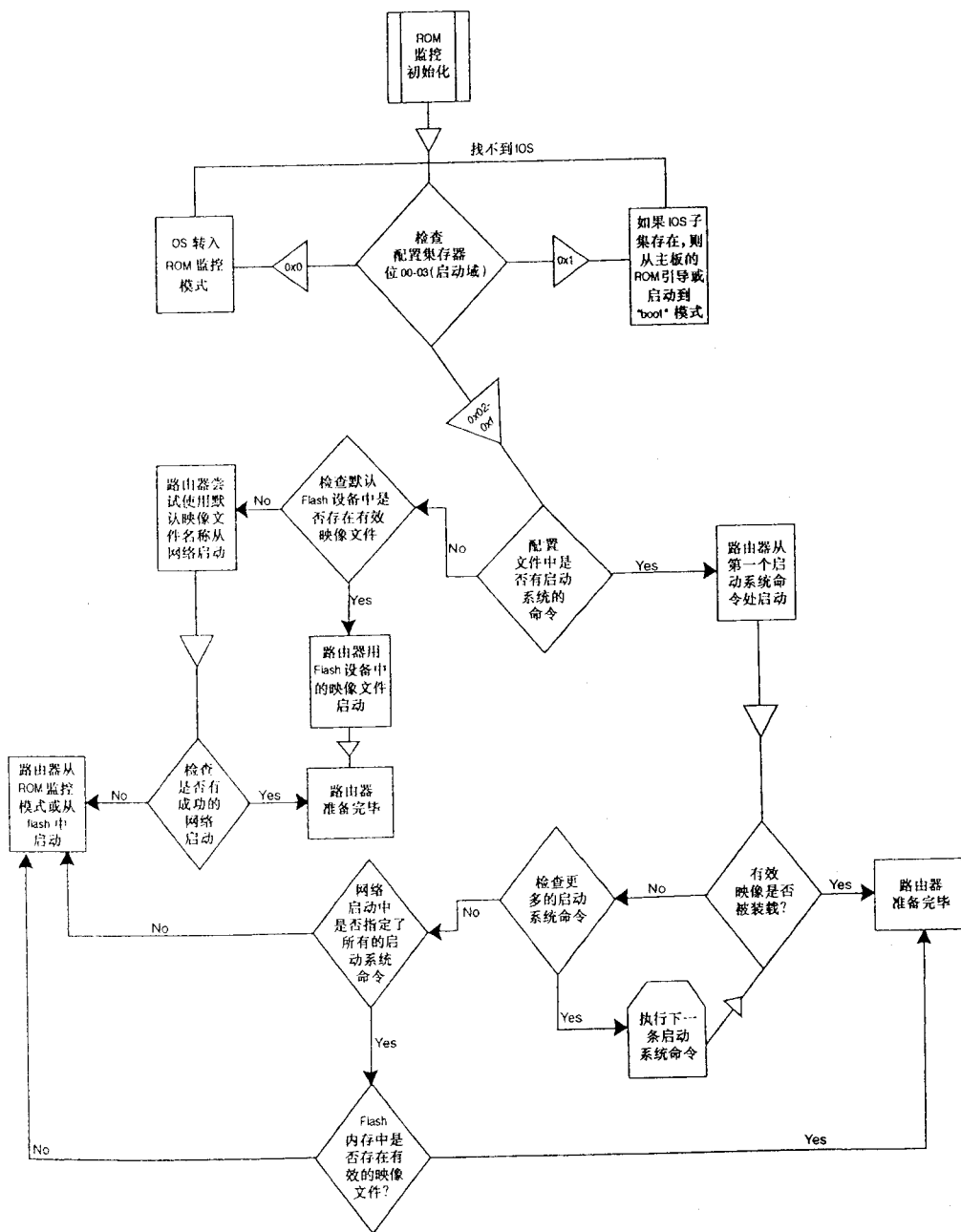


图 1-10 路由器引导过程

13. 引导寄存器的访问

引导寄存器是在路由器中以 16 进制表示的 16 位寄存器。路由器的型号和工艺决定了如何访问寄存器。如前所述，AGS 使用 16 根跳线来设置其寄存器。只要具有特权级访问权限，每一台路由器和交换机的寄存器都可以改动。交换机和路由器的工作方式大体相同。首先学习如何访问 Catalyst 交换机的寄存器，然后学习路由器寄存器的访问方式。

14. 访问和设置 Catalyst 交换机寄存器

基本说来，交换机和路由器的 16 位寄存器是很相像的，只有很少的一点差别。大部分用于网络引导和网络广播控制的位在 Catalyst 交换机上都没有用到。在 Catalyst 交换机上，第 6 位的用法和路由器上不同。将第 6 位设为 1 是清除 NVRAM 里的配置信息，如同输入了 **clear config all** 命令，即在交换机下次引导时清除 NVRAM 中存储的所有配置信息。

在 Catalyst 5000 系列的主控引擎（Supervisor Engine III）和 Catalyst 4000, 2948G 和 2926 系列交换机的初始化进程中有两个软件映像文件：ROM 监控代码和主控引擎系统代码。交换机重启时，首先执行 ROM 监控代码，然后，根据 NVRAM 中引导寄存器的设置情况，交换机或者保持在 ROM 监控模式下，或者载入主控系统映像文件。如果在加电过程中发生了致命错误，交换机就会停留在 ROM 监控模式下。图 1-11 显示了 Catalyst 系列交换机的 16 位引导寄存器的设置情况。表 1-7 是关于此寄存器的详细解释。

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
0				1				0				F			

图 1-11 缺省的 Catalyst 交换机的 16 位引导寄存器的设置情况

表 1-7 Catalyst 交换机引导寄存器各位的含义与缺省值

位	含 义	缺省设置
0-3	引导域： 0x0 = 从 ROM 监控模式引导 0x1 = 从主板的 ROM 引导，或者说如果简化版的 IOS 存在，就转入引导模式 0x2-0xF 产生下列事件（按操作顺序排列）： 在配置中找到 boot system 命令 如果没有找到环境变量 BOOT 列出的引导映像文件，则从 ROM 监控模式引导	1111
4	保留	0
5	保留	0
6	清除 NVRAM: 1=清除 NVRAM	0
7	OEM 位: 1=使引导时的不显示 cisco 标志	0
8	终止键: 1=失效	1

续表

位	含 义	缺省设置
9	不支持的波特率	0
10	IP 将使用全 0 的广播地址（未使用）	0
11-12	控制台波特率 00=9600 01=4800 10=1200 11=1200 在 catalyst4000 和 2928G 交换机上，该速率固定为 9600	00
13	对网络引导失败的响应：1=网络引导失败后从 ROM 引导，0=继续网络引导（未使用）	0
14	网络引导子网广播： 1=强制子网广播（未使用）	0
15	使用诊断消息：1=忽略 NVRAM，显示诊断消息（未使用）	0

寄存器的缺省值为 0x010f。系统将会使用环境变量 **BOOT** 指定的映像文件来引导，控制台工作速率为 9600 bit/s，NVRAM 中所有的设置信息都会在引导时加载。使用 **show boot [module_number]** 命令可以显示当前的寄存器设置情况。例 1-7 给出如何显示当前配置寄存器和 **BOOT** 环境的设置。

例 1-7 命令 show boot 的使用方法演示

```
Console>(enable) show boot
BOOT variable = slot0:cat5000-sup3.4-2-1.bin,1;bootflash:cat5000-sup3.3-2-1b.bin,1;bootflash:cat5000-sup3.4-1-2.bin,1;

Configuration register is 0x10f
Ignore-config: disabled
Console baud: 9600
Boot: image specified by the boot system commands

Console>(enable)
```

下面是 Catalyst 系列交换机专门用于寄存器操作的命令列表：

- **set boot config-register 0x value [mode_num]**

该命令可以直接按位配置引导寄存器的值，可以通过一次修改寄存器的值修改寄存器所有位。

- **set boot config-register baud {1200 | 2400 | 4800 | 9600}[module_number]**

该命令用于设置 ROM 监控模式控制台的波特率。只有当配置寄存器中指定的波特率的值与 **set system baud** 命令指定的波特率不同时才使用这个波特率，ROM 监控模式才使用该寄存器中指定的波特率。

- **set boot config-register ignore-config enable**

该命令能够使交换机重启时，清除 NVRAM 中存储的所有设置信息。这和使用 **clear config all** 命令之后再重新引导后的效果完全一样。

- **set boot config-register boot {rommon | bootflash | system}[module_number]**

该命令用于决定交换机下次引导时的引导方式：

—— **rommon** = 引导到 ROM 监控模式下

—— **bootflash** = 使用 FLASH 中存储的第一个映像文件来引导系统

—— **system** = 使用环境变量 BOOT 指定的映像文件来引导系统，这是缺省的设置。

- **set boot system flash device:[filename] [prepend] [module_number]**

该命令将在 BOOT 环境变量中指定一个映像文件，也可用于指定映像文件存放位置。

- **clear boot system flash device:[filename][module_number]**

该命令会将从 BOOT 环境变量中清除指定映像文件。

- **clear boot system all[module_number]**

该命令用于从 BOOT 环境变量中清除所有的映像文件。

15. 访问与设置 Cisco 路由器的寄存器

输入 **config-register <0x0000-0xFFFF>** 命令就可以在配置模式中修改寄存器。

例 1-8 演示了如何将寄存器的值从 2102 修改为 2142。这将强制路由器在初始化过程中忽略 NVRAM 的内容。在改变了寄存器内容之后，可以用 **show version** 命令来查看寄存器配置是否生效。

技巧 在更改寄存器设置之前，应该检查和记录当前寄存器的值。如果更改之后仍存在问題，那么这些记录对于解决问题将非常有用。

例 1-8 在配置模式中更改引导寄存器的值

```
Documenting the current setting
router#
router#show version
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS56-L), Version 12.0(3), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
*** text omitted ***
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

Configuration register is 0x2102
router#

Change the setting to 0x2142.

router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
router(config)#config-register 0x2142
router(config)#^Z
router#
router#show version
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS56-L), Version 12.0(3), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
*** text omitted ***
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

Configuration register is 0x2102 (will be 0x2142 at next reload)
```

技巧 无论何时在配置模式中更改了引导寄存器的值，系统都会提示用户在重新引导前先保存更改的配置信息。在进入配置模式之后，无论是否对系统配置进行过更改，在退出配置模式时系统都会有这样的提示。寄存器的值并不属于引导配置或是当前运行配置，因此改变寄存器设置时并不需要将配置信息保存下来。

16. 访问和设置寄存器: ROM 监控

如果无法改动路由器的配置，例如恢复密码时，可以强制性地使 Cisco IOS 暂停，进入 ROM 监控模式。要想进入监控模式，必须向路由器发送一个暂停信号。而暂停键在缺省情况下是被屏蔽的。因此，就需要重启路由器。基本上所有的 Cisco 路由器和交换机都能在初始化时的开始 60 秒之内通过向其发送暂停信号而使系统暂停下来。发送暂停信号来中断路由器、交换机的操作有很多方法，最常见方法见表 1-8。

表 1-8 标准的暂停键组合

仿真控制台	平台	操作系统	组合键
Hyperterm (version 595760)	IBM-兼容	Windows 9x	Ctrl-F6-Break
Kermit	Sun 工作站	Solaris	Ctrl-AL
Kermit	Sun 工作站	solaris	Ctrl-AB
MicroPhone Pro	IBM-兼容	Windows 9x	Ctrl-Break
Minicom	IBM-兼容	linux	Ctrl-A-F
ProComm Plus	IBM-兼容	DOS 或 Windows	Alt-B
Telrx	IBM-兼容	DOS	Ctrl-End
Telnet to Cisco	IBM-兼容	-	Ctrl-J
Teraterm	IBM-兼容	Windows 9x	Alt-B
Hyperterm	IBM-兼容	Windows 9x	Break
Hyperterm	IBM-兼容	Windows 9x	Ctrl-Break
Tip	Sun 工作站	Solaris	Ctrl-J, 然后 Break 或 Ctrl-C
VT 100 Emulation	Data general	N/A	F16
Hypterm	IBM 兼容	Windows NT	Shift-6 Shift-4 Shift-B (^SB)
Z-TERMINAL	Mac	Apple	Command-B
	Break-Out Box		Connect pin 2 (X-mit) to +V for half a second
	Cisco to aux port		Control-Shift-6, then B
	IBM 兼容		Ctrl-Break

就可以通过按下功能键和暂停键来发出暂停信号，这些键有时会在下翻页键或者是“Pause”键上。

在标准 101 键盘和有超级终端的 Windows 95/98 的组合里，暂停信号是通过按下 Ctrl-Break/Pause 组合键发出的。

在 Windows NT 操作系统环境下，需要配合一个功能键设置 NT 来发送暂停信号，要按下 ^\$B (Shift 6, Shift 4 和大写的 B) 来产生暂停信号。超级终端 5.0 个人版在 Windows NT 环境下可以不加任何附加设置来发送暂停信号。

访问 Catalyst 5000 或 2926G 系列交换机的寄存器，可以重启交换机，在初始化的前 60 秒之内按下暂停键来进入 ROM 监控模式。对于 Catalyst 4000 和 2948G 系列交换机来说，可以重启交换机，在初始化开始后的前 5 秒之内按下 Control-C 键来进入 ROM 监控模式。

使用其他终端仿真软件，请参考产品手册关于如何发送暂停信号的介绍。

成功发送暂停信号后，路由器的提示符将会改变成 a >或 rommon x >。因为有两种类型的 ROM 监控模式，因此也就有两种提示符。一种是以早期的 2000 系列主板为代表的，它对引导寄存器的设置多以手动为主。另一种是以比较新的 3600 系列和基于 RISC 平台。该 ROM 监控模式使用称为 CONFREG 的工具管理引导寄存器。表 1-9 列出了一些常用的路由器类型及其所用的 ROM 监控模式。区分所用 ROM 监控模式类别最简单的方法就是输入 ? 来获得帮助。如果有 CONFREG，可以输入 CONFREG 来使用这个工具。

表 1-9 ROM 监控模式兼容性表

CONFREG ROM 监控	基本 ROM 监控
Cisco 1003 系列	Cisco 2000 系列
Cisco 1600 系列	Cisco 2500 系列
Cisco 3600 系列	Cisco 3000 系列
Cisco 4500 系列	带 680x0 的 Cisco 4000 系列
Cisco 7200 系列	Cisco 7000 系列, 10.0ROM
Cisco 7500 系列	Cisco IGS 系列, ROM 中运行 9.1IOS
IDT Orion-based 路由器	
AS5200 and AS5300 平台	

首先，要了解基本 ROM 监控模式，然后就是 CONFREG 实用程序。当成功发出了暂停信号之后，将会看到类似于例 1-9 的屏幕信息，请注意其中的 Abort at 信息。

例 1-9 成功的通过暂停进入 ROM 监控模式的例子，然后输入 h 或 Help 命令

```
System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
Copyright (c) 1986-1995 by cisco Systems
2500 processor with 14336 Kbytes of main memory

Abort at 0x10200C2 (PC)
>
>h$           Toggle cache state
```

(待续)

```

B [filename] [TFTP Server IP address | TFTP Server Name]
    Load and execute system image from ROM or from TFTP server
C [address] Continue execution [optional address]
D /S M L V Deposit value V of size S into location L with modifier M
E /S M L Examine location L with size S with modifier M
G [address] Begin execution
H Help for commands
I Initialize
K Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
    Load system image from ROM or from TFTP server, but do not
    begin execution
O Show configuration register option settings
P Set the break point
S Single step next instruction
T function Test device (? for help)

Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
>
    
```

中止信息表明，首先，路由器操作暂停，OS 工作已经中止。其次，通过提示符可知已进入 ROM 监控模式。同样是在例 1-9 中，h 命令用于显示帮助列表，其功能和?一样。大多数的 ROM 监控模式都支持底层软硬件调试，其中一些命令是需要指出的：

- **H**——显示帮助信息，如例 1-9 所示。
- **I**——初始化路由器，与 reload 命令一样。
- **\$**——触发高速缓存，用于 TAC 调试。
- **P**——设置断点，用于 TAC 调试。
- **S**——用于 TAC 调试的单步执行命令。
- **TFuntion**——在 T 命令后面使用?命令用于对某个指定组件进行底层测试。通常是执行详细的硬件存储器的诊断。
- **B**——允许从 ROM 监控模式下进行手工引导。
 - B flash**——用 FLASH 中第一个文件引导
 - B filename [TFTP host]**——用 TFTP 通过网络进行引导
 - B flash filename**——用 FLASH 中的文件进行引导
- **L**——功能和 B 命令一样，但是路由器不会开始执行这些代码。
- **O**——查看 16 位的引导寄存器。
- **O/R 0x0000**——手动地用 16 进制值设置引导寄存器。如，**O/R 0x2102** 就将寄存器设为其缺省值。
- **D /S M L V**——将大小为 S 的值 V 存到修饰符为 M 的位置 L。
- **E /S M L**——查看大小为 S，修饰符为 M 的位置 L 处的值。如，**E/S 2000002** 就是直接从存储器中查看引导寄存器的值的。

现在可以验证路由器是否支持 CONFREG，或者是否仅仅支持基本 ROM 监控命令。后者通过 ROM 监控模式的提示符就可以判定，而通过输入?命令能够判定是否支持 CONFREG 命令。例如例 1-10 中提示符是>，表明需要通过基本 ROM 监控模式命令更改引导寄存器的

值。最后输入?命令来获得帮助，如下例所示：

例 1-10 通过暂停进入 ROM 监控模式的另一个例子，使用?或 Help 命令表明是否支持 CONFREG

```

About at 0x10200C2 (PC)
>?
$          Toggle cache state
B [filename] [TFTP Server IP address | TFTP Server Name]
           Load and execute system image from ROM or from TFTP server
C [address] Continue execution [optional address]
D /S M L V Deposit value V of size S into location L with modifier M
E /S M L   Examine location L with size S with modifier M
G [address] Begin execution
H          Help for commands
I          Initialize
K          Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
           Load system image from ROM or from TFTP server, but do not
           begin execution
O          Show configuration register option settings
P          Set the break point
S          Single step next instruction
T function Test device (? for help)

Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
>

```

例 1-11 是使用?命令后的显示情况，里面就有 CONFREG 实用工具。因此，可以用 CONFREG 来设置路由器的引导寄存器。请注意例 1-11 中的提示符 **rommon**，它说明了该系统支持 CONFREG。

例 1-11 支持 CONFREG 路由器?命令的使用

```

*** System received an abort due to Break Key ***
signal= 0x3, code= 0x0, context= 0x6033f2b8
PC = 0x6005eba4, Cause = 0x20, Status Reg = 0x34408302
rommon 1 >
rommon 1 > ?
alias          set and display aliases command
boot           boot up an external process
break          set/show/clear the breakpoint
confreg        configuration register utility
cont           continue executing a downloaded image
context        display the context of a loaded image
cookie         display contents of cookie PROM in hex
dev            list the device table
dir            list files in file system
dis            disassemble instruction stream
dnld           serial download a program module
frame          print out a selected stack frame
help           monitor builtin command help
history        monitor command history
meminfo        main memory information
repeat         repeat a monitor command

```

(待续)

```
reset          system reset
set            display the monitor variables
stack         produce a stack trace
sync          write monitor environment to NVRAM
sysret        print out info from last system return
unalias       unset an alias
unset         unset a monitor variable
rommon 2 >
```

有时，查看 CONFREG 的英文文字比按位对寄存器进行处理要更难理解。为了方便大家理解 CONFREG 的文字与寄存器对应位，请参看表 1-10。

表 1-10 CONFREG 与寄存器位的对比

CONFREG 文本	位设置	默认设置
enable "diagnostic mode"? y/n [n]:	15	关闭
enable "use net in IP bcst address"? y/n [n]:	14	关闭
disable "load rom after netboot fails"? y/n [n]:	13	打开
enable "use all zero broadcast"? y/n [n]:	10	关闭
enable "break/abort has effect"? y/n [n]:	8	关闭
enable "ignore system config info"? y/n [n]:	6	关闭
change console baud rate? y/n [n]:	11&12	关闭，关闭
change the boot characteristics? y/n [n]:	0-3	0x2

17. 密码恢复：路由器

对引导寄存器的工作原理有了充分的了解后，密码恢复的过程也就变得清晰明了。对于所有的路由器平台来说，只要改变寄存器第 6 位值，忽略 NVRAM 中的引导配置信息，然后重启路由器即可。路由器重启之后，不会再有运行配置信息存在。但配置信息仍然存在 NVRAM 中，可以在特权模式下用 **show startup-config** 命令来查看。由于运行配置文件内容为空，也就没有了进入特权模式的密码。因此，可以直接进入特权模式，用 **copy startup-config running-config** 命令拷贝一份引导配置到运行配置中。记住改回寄存器的值，设置特权密码，将关闭的接口（默认为关闭状态）打开并保存这些新的配置信息。整个过程归纳如下。

如前所述，不论寄存器第 8 位设置与否，在初始化开始后的前 60 秒内，路由器都会接受暂停信号。记住这一点，用下面的方法可以恢复大部分型号的路由器密码：

- 第 1 步 用终端仿真软件将 PC 机或 PDA 通过 Cisco 反线和路由器的控制台端口相连。
- 第 2 步 打开路由器的电源。
- 第 3 步 在路由器通电后的 60 秒内，通过按下暂停键或用前面讲述的方法发出一个暂停信号。
- 第 4 步 确定 ROM 监控模式的类型，是否支持 CONFREG?
 - 如果是基本 ROM 监控模式。
 - 设置第 6 位：>O/R 0x2142。这样，第 6 位就设为了 1，再用 **Initialize** 命令重新引导路由器。

—— 如果支持 CONFREG:

运行 CONFREG: **>CONFREG**。以缺省值回答每个问题，或直接回车，直到遇到问题：**Enable ignore system config info**。回答“yes”。这样也会将第 6 位设置为 1，用 **RESET** 命令重新引导路由器。

第 5 步 当重新引导时，路由器会试着执行 **setup**，用 **Ctrl-C** 中止 **setup** 的运行。

第 6 步 进入特权模式，拷贝引导配置文件到运行配置文件中，例如：**#copy startup-config running-config**。

第 7 步 进入配置模式，然后：

—— 将引导寄存器的值恢复为原始值。

—— 这时，所有的接口都会被关闭。将所有的接口恢复至正常状态。

—— 设置新的特权密码。

—— 保存新配置。

警告 忽略 NVRAM，重新引导路由器之后，一定要小心。路由器在 NVRAM 中仍然有一份配置文件，而且该文件很容易由于不小心敲击某个键而被覆盖。对于那些习惯用老命令的人来说，更为容易——敲入 **wr**（还不是 **wrt**）就会覆盖 NVRAM 中保存的配置信息。

技巧 进行修改或者任何可能会改变路由器的寄存器值的操作之前，一定要将当前的配置信息备份。这样花少量时间所做的事情在故障发生时的价值是无法估量的。

18. 密码恢复：交换机

交换机的密码恢复要比路由器简单。在初始化前 30 秒的时间内，密码和特权密码都是回车键。按以下步骤恢复 Catalyst 交换机的密码：

第 1 步 交换机加电。

第 2 步 交换机一旦载入，马上通过快速输入 **enable [Enter]** 进入特权模式，系统会提示输入密码。在开始的 30 秒内，按下回车键，进入特权模式。在特权模式里，用 **set password** 命令设置新的密码。当提示输入旧密码时，按回车就可以了。

第 3 步 在特权模式里，用 **set enablepass** 命令可以设置一个新的特权密码，这时的旧密码也是回车键。

1.7.2 Cisco IOS 软件的升级

有时，或者因为本书的实验要求，或者因为实际需要，有必要升级 Cisco 路由器的 IOS。如果知道了原理和方法，IOS 的升级问题是非常容易的。Cisco IOS 软件是存储在 FLASH 里的，FLASH 可能是 SIMM 形式的，也可能是卡式的。在升级 IOS 之前，要考虑 4 个问题：

- 路由器 Cisco IOS 的版本号必须高于 9.0（如果满足条件，可以升级到 IPv4）。
- FLASH 中可用的存储空间。
- 新的映像文件的大小，包括所需的 DRAM 的大小。
- 存放新映像文件的主机 IP 地址或主机名。

运行 **show flash** 命令就可以查看 SIMM 上 FLASH 的空间大小，而用 **dir [device]** 命令则

可以看到卡式 FLASH 里的内容。例如，**dir slot0:** 和/或 **dir slot1:**，这与卡式 FLASH 装在那个插槽上有关。这里有一些常用的 FLASH 命令及其 PCMCIA 的等价命令列表：

- **show flash**——显示 SIMM 上的 FLASH 内容，如例 1-8 所示。
- **dir [/all | /deleted | /long][device][filename]**。
 - **/all**——列出包括删掉和没有删掉的文件、有错的文件在内的所有文件。
 - **/deleted**——仅列出已删掉的文件。
 - **/long**——列出文件的详细信息。
 - **device**——列出某个指定的 FLASH 设备上的文件：FLASH:，BOOTFLASH:，SLOT0:，SLOT1。
 - **filename**——列出某个指定的 FLASH 文件信息。
- **cd**——在 FLASH 间切换。
- **copy source-device:filename destination-device:filename**——将文件从一个设备拷贝到另外一个设备。如果没有指定文件，后面会提示输入文件名。从 TFTP 拷贝到 FLASH 中去也是一样的。

以下是一些 FLASH 操作的例子：

- 从一个 FLASH 设备切换到另外一个，使用命令 **cd**——例如：**cd SLOT1:**。
- 查看不同设备上的文件，用 **dir [/all | deleted | long]** 命令——例如：**dir flash:** 或者 **dir**。

注释 如果使用的是卡式的 FLASH，在写入之前要确保其写保护功能已经关闭，这只要将 FLASH 卡末端的那个标卡移动一下就可以了。并不是所有的 FLASH 都一样，其产品文档或者是 FLASH 卡本身都会标明它的写保护使用方式。

例 1-12 给出了路由器上 **show flash** 和 **dir** 命令的执行结果。

例 1-12 show flash 和 dir 命令

```
router#show flash

System flash directory:
File Length Name/status
  1  10307412 c2500-js56-1.120-3.bin
[10307476 bytes used, 6469740 available, 16777216 total]
16384K bytes of processor board System flash (Read ONLY)

router#
router#dir
Directory of flash:/

  1  -rw-   10307412          <no date>  c2500-js56-1.120-3.bin

16777216 bytes total (6469740 bytes free)
router#
```

本例中，Cisco IOS 是 *c2500-js56-1.120-3.bin*，大小为 10307412 字节。这份映像文件使得 FLASH 仅剩 6.46 kB 空间。因此，在升级路由器的 IOS 时，必须删除旧的 IOS。

确认了路由器 FLASH 设备类型及剩余存储空间后，就可以准备升级 IOS。在准备过程中，第一件事情就是确定 IOS 要求的 FLASH 或者是 DRAM 的大小。每一个 IOS 有不同的 FLASH 与 DRAM 要求。确定 IOS 准确要求的惟一方法是到 Cisco 的网站上查询。

如果有权升级所用的 Cisco IOS, 那么在 Cisco 网站主页的服务与支持(Service & Support) 下的软件中心 (Software Center) 里就可以找到新版本的 IOS 软件。点击软件中心 (Software Center), 就会遇到一系列称为 Cisco IOS Planner 的问题。这些问题能够引导用户缩小所需要 IOS 软件的范围。本例中, 所选路由器 Cisco2600 的版本号是 12.0.9。请注意如图 1-12 所示, 存储器的最小要求为 4MB 的 FLASH 和 20MB 的 DRAM。必须要经过授权才能够登录 Cisco 的网站去查看和下载 Cisco IOS 软件。

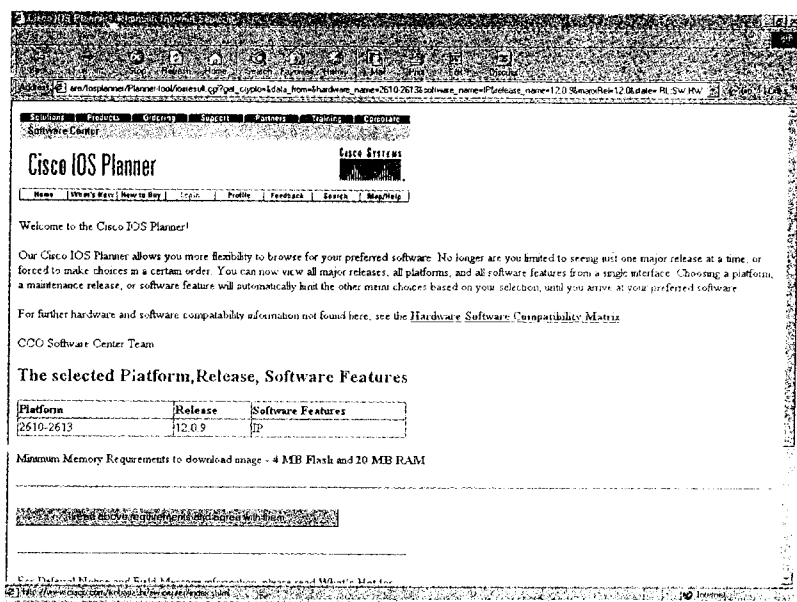


图 1-12 通过 Cisco 网站下载 IOS 的例子

值得一提的是, 查询 Cisco IOS 对存储器要求的惟一权威站点就是 Cisco 网站或者是 Cisco TAC。在几个平台上, 甚至是在同一个操作平台内, 都最好不要想当然地确定对存储器的要求。可以用图 1-12 的例子来证明这一点。同样的版本号为 12.0.9 的 Cisco IOS 软件, 在 Cisco 2500 系列的平台上要求却是 8MB 的 FLASH 和仅仅 4MB 的 DRAM。这就是为什么一定要确定所要升级的 IOS 版本对路由器要求的原因。

最后还需要确认的是, 具有可到达的存放新 FLASH 映像文件的可用 TFTP 服务器。这样, 路由器就可以进行升级操作了。用 `copy tftp flash` 命令将 Cisco IOS 软件从一台 TFTP 服务器拷贝到路由器里。在执行这条命令之前, 请检查:

- TFTP 服务器的 IP 地址。
- 服务器上 Cisco IOS 软件的名称。
- 路由器可以 ping 通 TFTP 服务器 (就是说, 应该是在一个本地连接的网络里)。

执行 `copy tftp flash` 命令时, 会遇到一系列的问题, 基本上和上面列出的一致。

注释 从版本号 11.0 的 Cisco IOS 开始, Cisco 引入了对文件处理更加接近于英语习惯的命令结构。不再使用 `configure memory` 命令, 系统使用 `copy startup-config running-config` 命令来将配置信息写入 NVRAM。表 1-11 比较了老的命令格式及其相对应的新命令。看完该表, 就会明白 Cisco 公司为什么做出这些改变。

表 1-11

Cisco IOS 软件文件命令的改变

旧命令	新命令
configure memory	copy startup-config running-config
configure network	copy {rcp tftp} running-config
configure overwrite-network	copy {rcp tftp} startup-config
copy erase flash	erase flash
copy verify or copy verify flash	verify flash, verify (cisco 7000 and Cisco 7500)
copy verify bootflash	verify boot flash
show configuration	show startup-config
Tftp-server system	tftp-server
write erase	erase startup-config
write memory	copy running-config startup-config
write network	copy running-config {rcp tftp}
write terminal	show running-config

下一个例子要做的是升级访问服务器的 IOS。在这个例子里，新的 Cisco IOS 文件是 c2500-js56-l.120-3.bin，位于 IP 地址为 206.191.241.45 的 TFTP 服务器上。从 Cisco 网站上可以查出，新的这个 IOS 需要 16MB 的 FLASH 和 8MB 的 DRAM。

如前所述，首先要确认路由器的 IOS 的版本高于 9.0。然后再确认有足够的 FLASH 和主存储器来运行新的 IOS。可以使用 **show version** 和 **show flash** 命令来查询。

例 1-13 显示了这些命令。

例 1-13 用 show version 和 show flash 命令验证 FLASH 和 DRAM

```
skynet_access_1#show version
Cisco Internetwork Operating System Software
IOS (tm) 3000 Software (IGS-INR-L), Version 10.3(7), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1995 by cisco Systems, Inc.
Compiled Wed 01-Nov-95 12:40 by vatran
Image text-base: 0x03022C14, data-base: 0x00001000

ROM: System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
ROM: 3000 Bootstrap Software (IGS-RXB00T), Version 10.2(8a), RELEASE SOFTWARE (fc1)

skynet_access_1 uptime is 1 week, 2 days, 16 hours, 19 minutes
System restarted by reload
System image file is "flash:/junky_old_ios.bin", booted via flash

cisco 2511 (68030) processor (revision L) with 14332K/2048K bytes of memory.
Processor board serial number 05309022
Bridging software.
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
16 terminal lines.
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)
```

(待续)

```

Configuration register is 0x2102

skynet_access_1#
skynet_access_1#show flash

System flash directory:
File Length Name/status
  1  4147048  /junky_old_ios.bin
[4147112 bytes used, 12630104 available, 16777216 total]
16384K bytes of processor board System flash (Read ONLY)

skynet_access_1#

```

从例子中可见，**show version** 命令的结果显示路由器的 IOS 版本号的确高于 9.0（本例中为 10.3.7）。接着，在第 16 行可以看到主存储器的大小是 14332 K/2048 K。意思是，路由器共有 16MB 的存储容量，分为 14MB 的主存储器和 2MB 的共享存储器，你只需看这两个数值的和就可以了。接下来检查可用 FLASH 存储器的大小。本例中，IOS 名称是 *junky_old_ios.bin*，大小是 4MB。由于 FLASH 共有 16MB，路由器就会提示是否删掉当前的 FLASH 映像文件。如果不想删除，请在配置文件里加入 **boot system flash IOS_filename** 命令行。

现在可以进行 Cisco IOS 的升级。已知映像文件的名称和所在服务器的 IP 地址，**ping** TFTP 服务器以确保网络连通。例 1-14 给出升级过程的其余步骤。

例 1-14 用 TFTP 服务器升级 IOS

```

skynet_access_1#ping 206.191.241.45
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 206.191.241.45, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
skynet_access_1#copy tftp flash
          **** NOTICE ****

Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the copy.
Routing functionality will not be available during that time.
If you are logged in via telnet, this connection will terminate.
Users with console access can see the results of the copy operation.
-----
[There are active users logged into the system]
Proceed? [confirm]y

System flash directory:
File Length Name/status
  1  4147048  /junky_old_ios.bin
[4147112 bytes used, 12630104 available, 16777216 total]
Address or name of remote host [255.255.255.255]? 206.191.241.45
Source file name? c2500-js56-1.120-3.bin
Destination file name [c2500-js56-1.120-3.bin]? c2500-js56-1.120-3.bin
Accessing file 'c2500-js56-1.120-3.bin' on 206.191.241.45...
Loading c2500-js56-1.120-3.bin from 206.191.241.45 (via Ethernet0): ! [OK]

Erase flash device before writing? [confirm]y
Flash contains files. Are you sure you want to erase? [confirm]y

Copy 'c2500-js56-1.120-3.bin' from server

```

（待续）

```

as 'c2500-js56-1.120-3.bin' into Flash WITH erase? [yes/no]yes

4:23:05: %SYS-5-RELOAD: Reload requested
%FLH: c2500-js56-1.120-3.bin from 206.191.241.45 to flash ...

System flash directory:
File Length Name/status
  1 4147048 /junk_ol_d_ios.bin
[4147112 bytes used, 12630104 available, 16777216 total]
Accessing file 'c2500-js56-1.120-3.bin' on 206.191.241.45...
Loading c2500-js56-1.120-3.bin .from 206.191.241.45 (via Ethernet0): ! [OK]

Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
ee ...erased
Loading c2500-js56-1.120-3.bin from 206.191.241.45 (via Ethernet0): !!!!!!!!!!!!!
<text omitted>
!!
[OK - 10307412/16777216 bytes]

Verifying checksum... OK (0xA519)
Flash copy took 0:06:04 [hh:mm:ss]
%FLH: Re-booting system after download
F3: 10070412+236968+1042784 at 0x3000060
<text omitted>
00:01:46: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS56-L), Version 12.0(3), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 08-Feb-99 22:55 by phanguye
skynet_access_1>

```

注意在例 1-14 中，删除当前 FLASH 重新引导路由器之前，路由器还要最后一次确认新的 IOS。通过查看路由器重启，会发现已经载入新的 IOS，还可以通过 **show flash** 命令来确认。

一些升级 IOS 时常见的问题如下：

- 拼错 IOS 的文件，比如容易混淆 J 和 L。
- TFTP 服务器与路由器不在同一本地网内。请确保 TFTP 服务器和路由器相邻，记住路由器会重启，而在 ROM 监控拷贝阶段路由表是不可用的。
- 没有确认新 IOS 需要的 FLASH 或者是主存储器大小。
- 如果无法实现路由，用全局命令 **IP default-gateway** 将路由器指向一个缺省的网关。

注释 最好使用 Cisco 提供的命名方式命名路由器的 IOS。Cisco 的命名规范使得软件文件名与软件功能集相互关联。

1.7.3 访问服务器的设置与使用

访问服务器能够一次为几台设备提供带外配置服务。在有多台重要的路由器和交换机的地方，访问服务器为这些设备的配置提供了最佳访问方式。后面的实验中有使用访问服务器来配置路由器和交换机的内容。

访问服务器的配置需要在其 IP 地址和 TTY 会话之间建立一种逻辑纽带。使用 Cisco 所称的 *reverse Telnet*，需要设置下面 3 个要素：

1. 一条传送指令

- 一个本地环路地址。
- 一份主机表。

设置传送指令需要了解“线路条目”的有关知识以及 Cisco 所谓的绝对线路号。在控制台模式中用 **show line** 命令可以列出配置路由器时可以使用的线路的信息。例 1-15 中 TTY 栏下最左边的数字就是绝对线路号。

例 1-15 用 show line 命令查看路由器的线路条目

Router>show line											
Tty	Type	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	1	0/0	-
1	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
2	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
3	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
4	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
5	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
6	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
7	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
8	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
9	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
10	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
11	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
12	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
13	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
14	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
15	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
16	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
17	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
18	VTY		-	-	-	-	-	0	0	0/0	-
19	VTY		-	-	-	-	-	0	0	0/0	-
20	VTY		-	-	-	-	-	0	0	0/0	-
21	VTY		-	-	-	-	-	0	0	0/0	-
22	VTY		-	-	-	-	-	0	0	0/0	-
Router>											

表 1-12 解释了例 1-15 中线路类型与线路编号的方法

表 1-12 线路类型与线路编号方法

线路类型	端口类型	描 述	编号方法
CON 或 CTY	控制台端口	用于配置	Line0
AUX	辅助端口	RS-232 DTE 端口，用于异步端口（TTY）备份	最后一个 TTY 线路编号+1
TTY	异步	与异步端口一样，通常用于远程节点使用 SLIP、PPP 和 Xremote 拨号会话	随平台而变化
VTY	虚拟终端	用于引入 Telnet、LAT、X.25、PAD 协议与同步端口的连接	最后一个 TTY 线路编号加 2，直到配置的 VTY 最大数值

例 1-15 中，1 到 16 号线都是 TTY 类型。这些线用于反向 Telnet 会话。配置一个反向 Telnet 会话，只需要在绝对线路号前面加上 20 即可。其句法有下面这两种方式。

在控制台模式下，**Telnet ip_address 20xx**，这里的 xx 就是绝对线路号（在这个例子中，从 01 到 16）。如果线路号为 1 位数，请一定要在前面的十位上加上 0。

另外一种设置反向 Telnet 会话的方式是在配置模式下利用 IP 主机表。进入配置模式后，

用 **IP host hostname 20xx ip_address** 命令。这里的 **ip_address** 应该是环路接口地址中的一个。这样，在其他物理接口关闭的情况下，就可以使用反向 Telnet 会话。要求 IP 地址必须可达，这也是为何使用本地回路地址的又一原因。

技巧 配置本地回路地址时，可以使用和网络或所建模型相关的方法。例如，每台路由器上都使用地址为 201.201.x.x 的本地回路地址。将所有路由器的环路接口 0 作为路由器的 ID，这里 x.x 是整个网络或模型中的惟一数字。选择 201.201 是希望在环路接口 0 的地址足够大，从而使其在 OSPF 中成为路由器 ID。在查看 OSPF 数据库时可以逻辑更清晰地标记路由器 ID。同时要注意不要将这类地址重分布到其他路由选择协议中。对于 DLSW 和 BGP 这类协议，可以在环路接口上使用可路由 IP 地址。我个人喜欢从环路接口 20 或更高数字开始。越具有个性化的设置就越容易进行排错和维护。另一段好的 IP 地址范围是 192.168.0.0，因为这是 RFC1918 里定义的私有地址。

在下面的例子里，假定一个本地回路地址设置为 201.201.1.1。线路命令 **transport input all** 显得更为重要。该命令允许通过 TTY 端口使用 Telnet 会话。可以用该命令每次改变一个线路属性，如例 1-16，也可以通过输入其范围一次设置多条线路，方法是键入 **line x-y**，x 是线路条目的起始号而 y 是该范围的结束项。

例 1-16 反向 Telnet 会话的设置

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip host r1 2001 201.201.1.1
Router(config)#line 1
Router(config-line)#transport input all
Router(config-line)#no exec
Router(config-line)#^Z
Router#
```

例 1-16 中在线路设置下使用了 **no exec** 命令。这条命令是配置反向 Telnet 会话的可选项。加上这条命令可以减少异步通信口上发生冲突的几率。每条线路上都有一个执行进程，或称为 **exec**。有时，这两个进程会将数据缓存到对方的区域里去，从而很难使用反向 Telnet 会话。错误信息 **% Connection refused by remote host** 表明了线路中此类冲突引起的错误。使用 **clear line line_entry** 命令清除线路中或任何用户的冲突错误。例 1-17 所示为常见的 **% Connection refused by remote host** 错误，然后显示了在清除线路冲突后，成功进入 r1 的例子。

例 1-17 清除线路中的冲突

```
Router#r1
Trying r1 (201.201.1.1, 2001)...
% Connection refused by remote host

Router#clear line 1
[confirm]y [OK]
Router#r1
Trying r1 (201.201.1.1, 2001)... Open

R1>
```

成功执行反向 Telnet 或其他 Telnet 会话之后，可能会希望回到原来的起始位置。Cisco 称之为挂起一个会话。要挂起会话，需要使用转向字符。同时按下 **Ctrl-Shift-6**，然后松开，再按下 **X** 键，使终端回到起始点。

如果想再次建立连接，先输入 **show session** 命令，找到想要恢复的连接编号，然后，键入该连接号。例 1-18 显示了 **show session** 命令的输出情况。

例 1-18 show sessions 命令

Router#show sessions				
Conn	Host	Address	Byte	Idle Conn Name
1	r2	201.201.1.1	0	3 r2
*	2	r1	201.201.1.1	0 0 r1
3	r3	201.201.1.1	0	3 r3

最左边的数字代表着相对线路号。例如，如果要回到主机 r3 中的对话中去，就键入 **3**，而要恢复 r2 上的对话，键入 **1**，再按下回车键。主机 r1 前面的*符号表明前一个会话在线。要回到该对话，只需按下回车键。

下面的过程能够跳过起始点（数字表示按下 **Ctrl-Shift-6** 键的次数）：

- 1 起始点（通过 Telnet 连接到或是作为控制台连接到的第一台 Cisco 路由器）。
- 2 从 1 过来的第一个 Telnet 或反向 Telnet 对话。
- 3 从 2 过来的第一个对话。
- 4 从 2 过来的第二个对话。
- 5 从 2 过来的第三个对话。

有时，可能从起始点 Telnet 到另外一台路由器，或可能是访问服务器。那么，从那台路由器或是访问服务器开始，可能需要做一次到模型中所有路由器去的反向 Telnet。在这种情况下，需要方便地跳回到访问服务器，避免每次费事地回到起始点。可以很快地两次按下 **Ctrl-Shift-6** 键，再按下 **X** 键就可以达到这样的目的，如从点 4 直接去到点 2，就如上面的列表中所示，不用再回到起始点了。规则体现为单独一次 **Ctrl-Shift-6** 能够回到起始点，第二次就可以进入中间站去了，以此类推。

1.7.4 帧中继交换机的配置

可以肯定，不论是在实验室里还是在其他的地方，帧中继交换机都是很有用的设备。就像引导寄存器一样，帧中继交换机的设置也是路由器设置中非常精深的内容之一。学会如何配置帧中继交换机，就能够建立很多不同的网络模型。由于我们是“服务提供者”，必须能够自主分配那些与 AT&T 或 MCI 所提供的完全一致的 DLCI。通过在实验室里建立精确的网络模型，不但能够增加安装网络的信心，还能降低配置错误或设备发生问题的概率。这一节将侧重讲述如何将一台 Cisco 路由器配置成一台帧中继交换机。第 5 章“WAN 协议与技术：帧中继”中会更为详细地讲述帧中继配置问题。

从本质上来说，帧中继交换技术是一种基于数据链路连接标识（DLCI）的帧交换技术。在路由器的帧中继 ARP 表中，DLCI 号是与接口关联的。帧中继使用其 ARP 表来检查 DLCI 与接口之间的配对情况，以决定是否把数据帧从某一特定的接口发出去。

帧中继交换机首先是一台 DCE 设备，也就是说：

- 它的任何模型都需要至少 3 台路由器：一台用于交换机功能，另外两台使用该交换机彼此通信。
- 帧交换机的串行接口处需要使用 DCE 电缆。
在这里，需要定义一些常用的帧中继用的术语：
- **Permanent virtual circuit 永久虚电路 (PVC)** ——是指用于帧传输的端到端的永久逻辑电路。PVC 的端点是用 DLCI 来寻址的。
- **Data-link connection identifier 数据链路连接标识 (DLCI)** ——是指用来识别用户端设备 (CPE) 与帧中继交换机之间的 PVC 所用的一个逻辑数字，从 16 到 1007。大多数情况下，DLCI 只在本地有意义，也就是说，只有本地设备知道 DLCI 代表含义。对于同一中心站点来说，远端可能有两个 PVC 具有同样的 DLCI 号。
- **Local Management Interface 本地管理接口 (LMI)** ——是指路由器与帧中继交换机之间的信令标准。交换机使用 LMI 来确定哪一些 DLCI 已被定义以及它们的当前状态。LMI 也支持每 10 秒一次的 keepalive 机制，用于确认 PVC 是否被激活，或是数据是否正在进行交换。Cisco 路由器支持 3 类 LMI: cisco、ansi 和 q933a。路由器能够以自动协商的方式决定使用何种类型的 LMI 进行通信：
 - **cisco** 是由网络 3 大巨头：Cisco, Digital 和 Northern Telecom 定义的 LMI 类型，是自动协商失败后的缺省类型，其状态信息是通过 DLCI 1023 发送。
 - **ansi** 是由 ANSI 通常称为附件 D 的 T1.617 标准定义的 LMI 类型。这是所有的帧中继网络里最常见的一种类型，其 LMI 状态信息是通过 DLCI 0 发送。
 - **q933a** 是由 ITU-T Q.933 (也称为附件 A) 定义的 LMI 类型，其状态信息是通过 DLCI 0 发送。
- **网络到网络接口 (NNI)** ——NNI 是用于两个交换机的通信的标准，既使用在帧中继交换机中，也用在 ATM 中。在 ATM 中，NNI 称为网络节点接口。

配置帧中继交换机时，必须完成以下操作：

第 1 步 启动帧中继交换功能。

第 2 步 设置 LMI 接口类型和帧中继接口类型。

第 3 步 用 **frame-relay route** 命令设置 PVC。

在这个例子里，要使用两台终端设备或路由器并且配置一台帧中继交换机。在开始之前做出一份 PVC 图表是很有帮助的。在图表里，需要包括 DCE、PVC 以及接口示意图。图 1-13 给出了该例子的图表，从硬件与服务供应商的角度对网络进行了强调。中间的帧中继交换机有两条 V.35 DCE 线缆用于连接另外两台路由器 R1 和 R2。这两台路由器在其串口 0 都连接 V.35 DTE 公头电缆。设置 PVC 将串口 0 上的 DLCI 101 映射到串口 5 上的 DLCI 102。

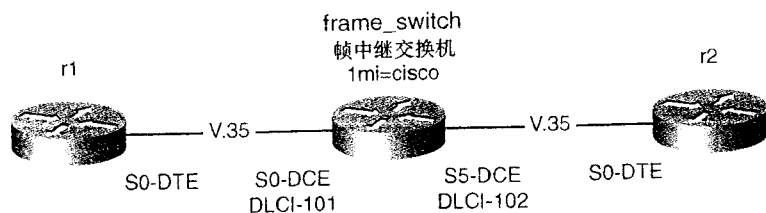


图 1-13 基本帧中继配置实例

做出这份图表之后，配置帧中继交换机的第一步是引导帧中继交换，可以用全局配置命令 **frame-relay switching** 来完成。随后，设置串口以用于帧中继交换，使用 **encapsulation frame-relay** 命令将数据格式封装成帧中继。另外，还需要在接口提示符下使用 **frame-relay lmi-type [ansi | cisco | q993a]** 命令来设置 LMI 的类型。随后，应该使用 **frame-relay intf-type dce** 命令。由于接口是 DCE 的，就要求使用 **clock rate bit/s** 命令来设置波特率，其中，*bit/s* 的值可以从 1200 到 8 000 000。最后，命令 **frame-relay route [16-1007]inbound_DLCI interface outbound_serial_ interface [16-1007]outboud_DLCI** 能够在接口上产生一个 PVC 并且将它映射到另外一个接口上去。例 1-19 演示了这些命令的用法以及帧中继交换机的基本配置方法。

例 1-19 配置基本的帧中继交换机

```
frame_switch#
frame_switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
frame_switch(config)#frame-relay switching
frame_switch(config)#interface serial 0
frame_switch(config-if)#encapsulation frame-relay
frame_switch(config-if)#frame-relay intf-type dce
frame_switch(config-if)#frame-relay lmi-type ansi
frame_switch(config-if)#clock rate 56000
frame_switch(config-if)#frame-relay route 101 interface s5 102
frame_switch(config-if)#exit
frame_switch(config)#
frame_switch(config)#interface serial 5
frame_switch(config-if)#encapsulation frame-relay
frame_switch(config-if)#frame-relay intf-type dce
frame_switch(config-if)#clock rate 56000
frame_switch(config-if)#frame-relay route 102 interface s0 101
frame_switch(config-if)#exit
frame_switch(config)#
```

例 1-20 列出了路由器的全部配置信息。

例 1-20 整个帧中继的配置

```
hostname frame_switch
!
frame-relay switching
!
interface Ethernet0
 ip address 172.16.1.2 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay
 clockrate 56000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 101 interface Serial5 102
!
<<<text omitted>>>
!
interface Serial5
 no ip address
```

(待续)

```

encapsulation frame-relay
clockrate 56000
frame-relay intf-type dce
frame-relay route 102 interface Serial0 101
!
<<<text omitted>>>
!
no ip classless

!
line con 0
line aux 0
line vty 0 4
  login
!
end

frame_switch#

```

此时，必须确定交换机正在工作。首先要注意的是检查两个不同 DLCI 数值代表的 PVC 建立了没有，是否在工作。PVC 只有当 LMI 在和两台 DTE 设备通信时才会工作。

帧中继交换的“Big show”和“Big D”

把下面的一些命令称为“Big show”和“Big D”是有原因的。诚然，还有很多别的命令，然而在进行调试时，命令越少越好。我有一个当时在 McDonnell Douglas 公司担任系统程序员的朋友，他曾经说过，“如果没有打破什么东西，那实际上什么都没有做。”在网络环境中做调试工作就会发现他的说法是对的。所有的调试命令的输出结果都是非常多的，这些调试命令需要和 **logging buffered 10000** 命令配合使用。称这些命令“big”的另一个原因是这些数目有限的调试命令能够用于解决几乎 90% 的连接和路由问题。这些“big”命令就是这里想要重点讲述的。帧中继交换所用的主要的 **show** 命令，也就是所谓的“big show”命令如下：

- **show interface xx**——显示物理连接的状态。**serial is up/down** 代表第 1 层，也就是物理层，**line protocol is up/down** 代表第 2 层协议。两个状态都应该是 **up**。**Serialx is up, line protocol is down** 就是表明有可能是 LMI 出了问题，无法匹配。
- **show frame-relay pvc**——显示 PVC 的状态。PVC 应该是激活的，而输入/输出包应该递增的。DLCI 应该是交换方式设成非本地工作状态以用于帧中继交换。
- **show frame-relay lmi**——显示收发到的 LMI 更新的状态。**Num Status Enq. Sent** 字段应该随着 **Num Status msgs Rcvd** 字段的增长而增长。**Num Status Timeouts** 字段应该是不增长的，如果超时信息往上增长，那就表明 LMI 的类型存在不匹配的问题。
- **show frame-relay route**——只在帧中继交换中有效，表示 PVC 映射到某个接口以及 DLCI 号。要确保 PVC 在相应的接口上终止的 DLCI 设置得正确，其状态应该是激活状态。
- **debug frame-relay lmi**——显示 LMI 的 keepalive 和交换信息。在一个 LMI 帧里面，类型 1 指这一帧是正常信息，而一个类型 0 则是一个 LMI 全部状态请求。其输出结果还含有 LMI 错误信息/超时信息以及连接状态。如果有一个错误的 LMI 类型被发送出去了，其代码如下：

```

invalid LMI type 1      cisco
invalid LMI type 2      Annex A 或 Q933a
invalid LMI type 3      Annex D 或 ANSI

```

还是这个例子，查看一下这些命令更为详细的信息。在帧中继交换机上，如同例 1-21 那样使用 **show interface** 命令。请注意第 1 层是在工作的，而线路协议是关闭的（有 **DCE LMI down** 信息显示），这样的信息与 **DCD=up DSR=up DTR=up RTS=up CTS=up** 一起就能够确定第 1 层正常工作的，并能够立即知道是第 2 层中一个与帧相关的问题。

例 1-21 show interface 的例子

```
frame_switch#show interface serial 0
Serial0 is up, line protocol is down
Hardware is HD64570
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
LMI enq sent 0, LMI stat recvd 0, LMI upd recvd 0
LMI enq recvd 297, LMI stat sent 297, LMI upd sent 0, DCE LMI down
LMI DLCI 0 LMI type is ANSI Annex D frame relay DCE
FR SVC disabled, LAPF state down
Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
Last input 00:00:05, output 01:24:05, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations 0/1/256 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
2229 packets input, 30711 bytes, 0 no buffer
Received 82 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
297 packets output, 4413 bytes, 0 underruns
0 output errors, 0 collisions, 645 interface resets
0 output buffer failures, 0 output buffers swapped out
1290 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
frame_switch#
```

例 1-22 中使用了 **show frame-relay pvc** 来处理与帧有关的问题，在该命令的输出结果中应该查看是否有 **PVC STATUS = ACTIVE** 的信息，**DLCI USAGE** 在帧中继交换机上应为 **SWITCHED**，而在网络的 DTE 一端应为 **LOCAL**。链路配置的所有 DLCI 都会出现在结果中。PVC 显示为 **INACTIVE** 意味着没有接收到 LMI 状态请求。

例 1-22 show frame-relay pvc 命令

```
frame_switch#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DCE)

DLCI = 101, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 0          output pkts 0          in bytes 0
out bytes 0           dropped pkts 0          in FECN pkts 0
in BECN pkts 0        out FECN pkts 0        out BECN pkts 0
in DE pkts 0          out DE pkts 0          out bcast pkts 0
out bcast pkts 0      out bcast bytes 0
pvc create time 07:01:22, last time pvc status changed 06:59:57
Num Pkts Switched 0
```

(待续)

```
PVC Statistics for interface Serial5 (Frame Relay DCE)

DLCI = 102, DLCI USAGE = SWITCHED, PVC STATUS = INACTIVE, INTERFACE = Serial5

input pkts 0          output pkts 0          in bytes 0
out bytes 0          dropped pkts 0        in FECN pkts 0
in BECN pkts 0       out FECN pkts 0       out BECN pkts 0
in DE pkts 0         out DE pkts 0
out bcast pkts 0     out bcast bytes 0
pvc create time 07:01:22, last time pvc status changed 02:12:10
Num Pkts Switched 0
frame_switch#
```

使用 **show frame-relay lmi** 命令可以缩小问题的范围。例 1-23 中关注的问题是一段时间内 LMI 超时的记录。如前所述，**Num Status Eng Rcvd** 应该随着 **Num Status msgs Sent** 的增长而增长，而 **Num St Eng. Timeouts** 则不应增长。这里还会指定 LMI 的类型。当然，无效数目不应该增长。

例 1-23 show frame-relay lmi 命令

```
frame_switch#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DCE) LMI TYPE = ANSI
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 297          Num Status msgs Sent 297
Num Update Status Sent 0          Num St Eng. Timeouts 1677

LMI Statistics for interface Serial5 (Frame Relay DCE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 2806          Num Status msgs Sent 2806
Num Update Status Sent 0          Num St Eng. Timeouts 4
frame_switch#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DCE) LMI TYPE = ANSI
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 297          Num Status msgs Sent 297
Num Update Status Sent 0          Num St Eng. Timeouts 1678

LMI Statistics for interface Serial5 (Frame Relay DCE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 2807          Num Status msgs Sent 2807
Num Update Status Sent 0          Num St Eng. Timeouts 4
frame_switch#
```

如果过一段时间再运行这条命令，那么超时的数目有可能会逐渐递增，不再收到状态信息。本例显然发生了 LMI 问题。可以用调试命令进行确认。**debug framerelay lmi** 命令提供了很多有用的信息。查看调试日志会看到例 1-24 所示条目。

例 1-24 debug frame-relay lmi 命令的输出结果

```
06:01:52: Serial5(in): StEnq, myseq 122
06:01:52: RT IE 1, length 1, type 1
06:01:52: KA IE 3, length 2, yourseq 123, myseq 122
06:01:52: Serial5(out): Status, myseq 123, yourseen 123, DCE up
06:01:53: Serial0: Invalid LMI type 1
06:01:58: Serial0(down): DCE LMI timeout
```

该日志进一步证实串口 0 上的 LMI 发生了问题。**invalid LMI type 1** 表明交换机正在从 DTE 端接收 Cisco LMI，引起超时，从而产生关闭状态。如果问题是 **invalid LMI type 2** 或者是 **invalid LMI type 3**，该 LMI 就可能分别是 Q993a 或 ANSI。不要把该类型字段与正常情况下接收到的类型字段相混淆。正常工作时，类型信息表明所接收的 LMI 帧的类型，其 **myseq** 与 **yourseen** 字段应该与串口 5 处见到的 **DCE up** 一起递增。现在可以肯定确实是 LMI 存在问题。如果在帧中继交换机上将 LMI 类型改成 cisco，就可以看到例 1-25 中日志显示的结果。

例 1-25 LMI 纠正过程中记录在日志里的 debug 输出

```
09:52:33: Serial0: Invalid LMI type 1
09:52:39: %SYS-5-CONFIG_I: Configured from console by console
09:52:42: Serial5(in): StEnq, myseq 232
09:52:42: RT IE 1, length 1, type 1
09:52:42: KA IE 3, length 2, yourseq 233, myseq 232
09:52:42: Serial5(out): Status, myseq 233, yourseen 233, DCE up
09:52:43: Serial0(down): DCE LMI timeout
09:52:43: Serial0(in): StEnq, myseq 0
09:52:43: RT IE 1, length 1, type 0
09:52:43: KA IE 3, length 2, yourseq 6, myseq 0
09:52:43: Serial0(out): Status, myseq 1, yourseen 6, DCE down
09:52:52: Serial5(in): StEnq, myseq 233
09:52:52: RT IE 1, length 1, type 1
09:52:52: KA IE 3, length 2, yourseq 234, myseq 233
09:52:52: Serial5(out): Status, myseq 234, yourseen 234, DCE up
09:52:53: Serial0(in): StEnq, myseq 1
09:52:53: RT IE 1, length 1, type 1
09:52:53: KA IE 3, length 2, yourseq 7, myseq 1
09:52:53: Serial0(out): Status, myseq 2, yourseen 7, DCE up
09:52:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
09:53:00: %FR-5-DLCICHANGE: Interface Serial5 - DLCI 102 state changed to ACTIVE

09:53:02: Serial5(in): StEnq, myseq 234
09:53:02: RT IE 1, length 1, type 1
09:53:02: KA IE 3, length 2, yourseq 235, myseq 234
09:53:02: Serial5(out): Status, myseq 235, yourseen 235, DCE up
09:53:03: Serial0(in): StEnq, myseq 2
09:53:03: RT IE 1, length 1, type 1
09:53:03: KA IE 3, length 2, yourseq 8, myseq 2
09:53:03: Serial0(out): Status, myseq 3, yourseen 8, DCE up
```

现在，对于建立帧中继交换机的配置要求已经有了基本了解。建立点到多点帧中继的连接实验中相关配置会有些许改动。在第 5 章有关于帧中继 LMI 帧及其交换中将作进一步讨论。

1.7.5 配置路由发起源或主干路由器

以下将讨论有助于建立网络互联模型的组件——路由发起源，或称为主干路由器。简而言之，路由发起源就是配置有虚拟网络或者是环路接口的路由器。这些虚拟网络的地址是第 3 层地址，并通过路由选择协议来宣告。从实用角度来看，其主要作用是通过在路由表中注入路由，使所搭建的网络看起来比真实网络更大。配置一台路由发起源，需要做以下工作：

第 1 步 加入一个或多个虚拟接口或环路接口。

第 2 步 确定所用的第 3 层协议，将其用于环路接口上。

第 3 步 使用路由选择协议宣告网络。

使用小型帧中继网络，将其中一台路由器配置成路由发起源，然后检查该路由发起源的下游路由器的状态。在配置模式下输入 **interface loopback [0-2147483647]** 命令来设置一个环路接口，然后配置第 3 层地址，并确定如何宣告这些网络。例 1-26 所示为一些配置了 IP 地址的环路接口地址。用自治域 (AS) 号为 2001 的 EIGRP 可以通过帧中继云图在路由器间宣告网络。参考图 1-14，可以将 R1 配置成路由发起源。

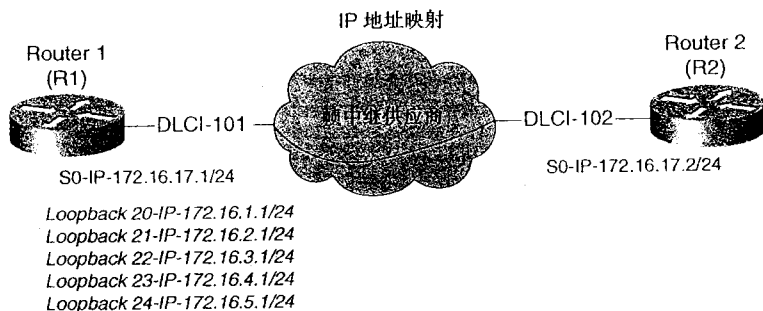


图 1-14 路由发起源的 IP 图

例 1-26 显示了 R1 的配置。

例 1-26 配置路由发起源

```
r1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
r1(config)#interface loopback 20
r1(config-if)#
02:41:51: %LINK-3-UPDOWN: Interface Loopback20, changed state to up
02:41:52: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback20, changed state to up
r1(config-if)#ip address 172.16.1.1 255.255.255.0
r1(config-if)#interface loopback 21
r1(config-if)#ip address 172.16.2.1 255.255.255.0
r1(config-if)#interface loopback 22
r1(config-if)#ip address 172.16.3.1 255.255.255.0
r1(config-if)#interface loopback 23
```

(待续)

```
r1(config-if)#ip address 172.16.4.1 255.255.255.0
r1(config-if)#interface loopback 24
r1(config-if)#ip address 172.16.5.1 255.255.255.0
r1(config-if)#exit
r1(config)#router eigrp 2001
r1(config-router)#network 172.16.0.0
r1(config-router)#exit
r1(config)#interface serial 0
r1(config-if)#ip address 172.16.128.1 255.255.255.252
r1(config-if)#^Z
r1#
```

设置完 R2 的 IP 并且加上 EIGRP 协议后，可以观察到虚拟网络在下游路由器上的状态。在后面的章节中，将利用路由发起源来练习过滤和观察不同路由选择协议对路由的处理方法。例 1-27 列出了在用串口上正确配置 IP 地址和路由选择协议之后，路由器 R2 的路由表。

例 1-27 路由发起源向下游路由器宣告网络

```
r2#
r2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
C       172.16.128.0/30 is directly connected, Serial0
D       172.16.4.0/24 [90/2297856] via 172.16.128.1, 00:11:09, Serial0
D       172.16.5.0/24 [90/2297856] via 172.16.128.1, 00:11:09, Serial0
D       172.16.1.0/24 [90/2297856] via 172.16.128.1, 00:11:09, Serial0
D       172.16.2.0/24 [90/2297856] via 172.16.128.1, 00:11:09, Serial0
D       172.16.3.0/24 [90/2297856] via 172.16.128.1, 00:11:09, Serial0
r2#
```

注意，R2 收到来自 172.16.128.1 的标志为 D 的 EIGRP 路由。该路由器可以看作是一个很大的 EIGRP 网络的一部分。

1.7.6 配置模拟远程访问

网络的远程访问在学习和实际应用中都非常有用。Cisco TAC 经常询问客户是否接有模拟调制解调器，这对用户故障的排错很有帮助。得到第一手资料对于问题的解决非常重要。通过这种带外连接方式，从远程地点发现问题并进行处理，其价值是无法估量的。本节内容就是为了使大家学会配置简单的模拟调制解调器以用于远程技术支持或 PPP 备份。Cisco Press 出版了一本很好的有 1 500 页的参考书，叫做《Cisco IOS Dial Solutions》，该书非常详尽地讲述了拨号网络的问题。本书中这方面的内容大部分改编自该书。

配置模拟远程访问有时会很困难。Cisco IOS 版本、路由器端口和调制解调器之间的联系非常紧密，这使得不同型号路由器的端口配置都不相同。通常，路由器型号或是调制解调器

类型的任何改变都将会使得路由器的设置发生改变。无论配置和重新配置多么困难，学会一些命令能够解决大部分的模拟拨号问题。

本节侧重于终端会话是如何通过访问服务器上的辅助接口或异步接口连接到路由器的。第 4 章“WAN 协议与技术：点对点协议（PPP）”讲述了 PPP 和 PPP 与辅助端口、异步端口以及串行端口是如何配合工作的。

比较路由器的异步端口与辅助端口。这两种端口都有异步功能，包括：

- 支持网络协议（例如 IP、IPX 或 AppleTalk）。
- 支持封装（例如 PPP 和 ARAP）。
- 支持认证。

辅助端口与异步端口间最显著的区别就是工作速度。异步端口的最高速度可达 115 200 bit/s，而辅助端口最高只能达到 38 400 bit/s。表 1-13 对辅助端口和异步端口的比较进行了总结。

表 1-13 AUX 与 Asynchronous 比较

特 点	异步端口	辅助端口
最大速率	115 200bit/s	38 400bit/s
是否在为无 CPU 中断的情况下直接访问内存提供 DMA 缓存	是	否
是否支持可以不增加 CPU 额外开销在接口上封装 ppp 帧，	是	否
是否支持 IP 快速交换	是	否

除了这些区别外，这两个端口的工作与配置方式几乎完全相同。

不同调制解调器的设置可能不相同。但现在大多数的调制解调器都使用标准化的 AT 命令集。AT 命令集是用来通过设置调制解调器的位寄存器来对其进行设置，包括强制性压缩，响铃应答等功能。配置路由器以支持调制解调器，需要完成下面三步配置：

第 1 步 将调制解调器连到辅助端口或异步端口。

第 2 步 设置调制解调器的线路或线路条目。

第 3 步 通过交互脚本（chat script）或自动选项来配置调制解调器。

1. 第 1 步：将调制解调器连到辅助端口或异步端口

设置模拟通信的第一步是将调制解调器与路由器连接。表 1-14 重复了表 1-2 的内容，以强调调制解调器上所用电缆、接头的类型。大多数情况是将 Cisco 黑色或蓝色的扁平电缆连到辅助端口上，MMOD 的接头用于调制解调器。

表 1-14 异步设备电缆连接选择

访问服务器端口	RJ-45 终端线类型	适配器接头	端设备
Console 或 aux 端口	全反线	DTE	串行线
Console 或 aux 端口	直连线	DCE	串行线
Console 或 aux 端口	全反线	MMOD/MODEM	调制解调器

2. 第 2 步：配置调制解调器的线路或线路条目

下一步是配置与辅助端口或异步线路对应的线路命令。用 show line 命令可以查看对应的

例中的绝对线路号是 1。要想将辅助端口设置成异步通信模式，需要在配置模式下输入 **Line 1** 命令。

例 1-28 show line 的输出结果

Router#show line											
Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	1	0/0	-
1	AUX	9600/9600	-	-	-	-	-	0	1	0/0	-
2	VTY		-	-	-	-	-	0	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	-
Router#											

如果在访问服务器上运行 **show line** 命令，其输出要稍微复杂一些。例 1-29 中，辅助端口的绝对线路号为 17。因此，要想在该端口上连接调制解调器，就要在配置模式下输入命令 **Line 17** 以便开始设置。

例 1-29 在访问服务器上运行 show line 命令的输出结果

skynet_access_1#show line											
Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	1	0	0/0	-
1	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
2	TTY	9600/9600	-	-	-	-	-	0	103	0/0	-
* 3	TTY	9600/9600	-	-	-	-	-	0	1	1400/4202	-
* 4	TTY	9600/9600	-	-	-	-	-	0	0	1401/4203	-
* 5	TTY	9600/9600	-	-	-	-	-	1	1	2/9	-
* 6	TTY	9600/9600	-	-	-	-	-	0	0	465/1704	-
7	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
8	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
9	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
10	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
11	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
12	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
13	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
14	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
15	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
* 16	TTY	38400/38400	-	inout	-	-	-	0	0	0/0	-
17	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
18	VTY		-	-	-	-	-	0	0	0/0	-
19	VTY		-	-	-	-	-	0	0	0/0	-
20	VTY		-	-	-	-	-	0	0	0/0	-
21	VTY		-	-	-	-	-	0	0	0/0	-
Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
22	VTY		-	-	-	-	-	0	0	0/0	-
skynet_access_1#											

找出了相应的线路，就可以将其设置为支持调制解调器。此时要用到第 1 层特性，即路由

器与调制解调器通信的速率，调制解调器如何处理流控制问题以及如何处理载波信号问题。

比较表 1-15 的内容，就会发现下面这些是正确的。

表 1-15 Modem 传送速率与路由器端口速率

调制解调器传送速率	端口速率
9600	38 400
14 400	57 600
28 800	115 200

表 1-13 说明辅助端口最高速率为 38 400 bit/s，而调制解调器缺省发送速率为 9600 bit/s。因此，不需要调整辅助端口的速率就已经获得其最高速率。需要调整异步线路速率时，使用 **speed [38400 | 57600 | 115200]** 命令。

如果需要调整的速率高于 38 400 bit/s，就要启动硬件数据流控制，这是通过 **flowcontrol hardware** 命令来实现。

线路还必须告诉调制解调器如何处理载波信号。如果要将线路设置成载波检测 (CD) 丢失时挂断连接，就要用命令 **modem inout**。某些情况下，要将调制解调器设置为只应答的模式，就可以使用命令 **modem dialin**。

3. 第 3 步：通过交互脚本 (chat script) 或自动的方式设置调制解调器

最后一个步骤是完成调制解调器初始设置。配置外部调制解调器最容易最直接的方法就是用 **autoconfigure** 命令。Cisco IOS 为大多数型号的调制解调器都定义了一些初始化字符串。可以用 **show modemcap** 命令查看这些预先定义好的字符串。例 1-30 列出了该命令的输出结果，12.0.3 版本的 Cisco IOS 预定义的调制解调器类型。

例 1-30 命令 show modemcap 的输出结果

```
Router#show modemcap
default
codex_3260
usr_courier
usr_sportster
hayes_optima
global_village
viva
telebit_t3000
microcom_hdms
microcom_server
nec_v34
nec_v110
nec_piafs
cisco_v110
mica

Router#
Router#show modemcap default
Modemcap values for default
Factory Defaults (FD): &F
Autoanswer (AA): S0=1
```

(待续)

```
Carrier detect (CD): &C1
Drop with DTR (DTR): &D2
Hardware Flowcontrol (HFL): [not set]
Lock DTE speed (SPD): [not set]
DTE locking speed (DTE): [not set]
Best Error Control (BER): [not set]
Best Compression (BCP): [not set]
No Error Control (NER): [not set]
No Compression (NCP): [not set]
No Echo (NEC): E0
No Result Codes (NRS): Q1
Software Flowcontrol (SFL): [not set]
Caller ID (CID): [not set]
On-hook (ONH): H0
Off-hook (OFH): H1
Miscellaneous (MSC): [not set]
Template entry (TPL): [not set]
Modem entry is built-in.
```

Router#

这个列表也列出了缺省类型调制解调器的 AT 字符串。多年来，Cisco 在提高其对模拟支持设备简易配置方法与可靠性上下了很大功夫。Cisco 脚本这种方式在以前和现在都受到普遍欢迎。交互脚本是在配置模式下通过 **chat-script EXPECT SEND EXPECT SEND** 的格式输入的。然后在线路里调用交互脚本。几乎 90% 以上的调制解调器都可以使用这种方式工作，而不必使用复杂的 AT 命令字符串。但是，如果调制解调器的类型不在 **show modemcap** 命令的输出结果中，那就应该尽量不用脚本，而用 **modem auto-configure discovery** 或 **modem auto-configure type default**。另外一个要避免使用脚本的情况就是要新建或更改现有的 modemcap 条目，这应该在配置模式下用 **modemcap edit modem-name attribute value** 来完成，该命令使得只需要最少的 AT 命令操作就可以完成配置线路接口。

找出所用的调制解调器类型或进行了类型自定义后，就可以在适当的线路项下通过加进 **modem auto-configure type modem-name** 建立逻辑连接。

4. 配置练习：把一台调制解调器连到一台路由器

例 1-31 是如何把一台调制解调器连到路由器的完整示例。

第 1 步 确定要配置的线路号。通过 **show line** 命令查看，记下绝对线路号，即例子中突出显示的数字或第一列最右边的数字。将其记为 X。

第 2 步 进入配置模式，输入适当的 **line x** 配置命令。

第 3 步 依然在该线路项下，加入下面的命令：

```
—— transport input all
—— modem inout
—— modem autoconfigure discovery
```

或是：

```
—— modem autoconfigure type [default | modem-name]
```

第 4 步 设置特权密码，允许特权模式访问。

例 1-31 配置辅助端口上的模拟拨号访问

```

Router#
Router#show line

```

Tty	Type	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	9	0/0	-
1	AUX	9600/9600	-	-	-	-	-	0	1	0/0	-
2	VTY		-	-	-	-	-	0	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	-

```

Router#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line 1
Router(config-line)#transport input all
Router(config-line)#modem inout
Router(config-line)#modem autoconfigure discovery
Router(config-line)#^Z
Router#

```

输入 **line** 命令之后，应该用一个反向 Telnet 会话来验证调制解调器的连接情况。例 1-32 中，加入了 IP 地址为 201.201.201.1 的一个环路接口来支持反向 Telnet 会话。如果会话连接上，使用 **clear line** 命令，然后再进行反向 Telnet 会话。当和调制解调器建立连接之后，用 **ATZ** 命令将调制解调器复位。如果可以进行反向 Telnet 会话，表明传输已经设置正确，线路已经打开。如果可以反向 Telnet 但不能使用 **ATZ** 命令，那可能是电缆有问题。

如果依然不能使用反向 Telnet 或 **ATZ**，先确认以上命令都已经配置，然后打开调试模式。按下 **Ctrl-Shift-6** 键再键入 **X** 可以退出或暂停反向 Telnet 会话。如果要继续查找线路故障，使用 **disconnect** 命令关闭反向 Telnet 会话。

例 1-32 反向 Telnet 会话，和 AT 命令应用

```

Router#telnet 201.201.201.1 2001
Trying 201.201.201.1, 2001 ... Open

atz
OK

Router#
Router#disconnect
Closing connection to 201.201.201.1 [confirm]y
Router#

```

5. 调制解调器的“Big show”和“Big D”

有两条功能强大的调试命令有助于查找调制解调器的连接故障：**debug modem** 命令和 **debug confmodem** 命令。他们和 **show line x** 命令一起使用能够快速缩小故障原因的范围。用 **show line x** 命令可以查看到一些表明线路工作正常的重要指标（请参考例 1-33），看到的调制解调器状态应该是 **detected** 和 **idle**。如果调制解调器状态不是 **idle**，请试着用 **clear line x** 命令将其清为闲置状态。

例 1-33 调制解调器操作和 show line 的例子

```
Router#show line 1
  Tty Typ   Tx/Rx   A Modem  Roty Acc0 AccI   Uses   Noise  Overruns  Int
    1 AUX   38400/38400 - inout   - - -      0      1      0/0      -  Ie

Line 1, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 38400/38400, no parity, 2 stopbits, 8 databits
Status: No Exit Banner, Modem Detected
Capabilities: Modem Callout, Modem RI is CD, Modem Discovery
Modem state: Idle
Group codes:      0
Modem hardware state: CTS* noDSR  DTR RTS, Modem Configured
Special Chars: Escape Hold Stop Start Disconnect Activation
                  ^x none - none
Timeouts:         Idle EXEC   Idle Session  Modem Answer  Session  Dispatch
                  00:10:00   never          none          none      not set
                  Idle Session Disconnect Warning
                  never
                  Login-sequence User Response
                  00:00:30
                  Autoselect Initial Wait
                  not set

Modem type is usr_sportster.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed transports are lat pad v120 mop telnet rlogin nasi. Preferred is lat.
No output characters are padded
No special data dispatching characters
Router#
```

例 1-34 显示的一条无效线路。请注意每一次显示的速率都不一样，这是因为路由器不停尝试和调制解调器通信；而且，状态行没有 **modem detected** 信息；最后，注意路由器不能识别清除发送（CTS）信号，这就充分说明了存在着电缆或接头的问题。

例 1-34 用 show line 显示的无效线路例子

```
Router#show line 1
  Tty Typ   Tx/Rx   A Modem  Roty Acc0 AccI   Uses   Noise  Overruns  Int
  *  1 AUX   1200/1200 - inout   - - -      3      1      0/0      -

Line 1, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 1200/1200, no parity, 2 stopbits, 8 databits
Status: Ready, Active, No Exit Banner ← notice an absence?
Capabilities: Modem Callout, Modem RI is CD, Modem Discovery ← 'modem detected'
Modem state: Ready
Group codes:      0
Modem hardware state: noCTS noDSR  DTR RTS ←no CTS
Special Chars: Escape Hold Stop Start Disconnect Activation
                  ^x none - none
Timeouts:         Idle EXEC   Idle Session  Modem Answer  Session  Dispatch
                  00:10:00   never          none          none      not set
```

(待续)

```

Idle Session Disconnect Warning
never
Login-sequence User Response
00:00:30
Autoselect Initial Wait
not set

Modem type is usr_sportster.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed transports are lat pad v120 mop telnet rlogin nasi. Preferred is lat.
No output characters are padded
No special data dispatching characters
Router#
    
```

运行本节开始提到的两条 **debug** 命令，这时会看到路由器不停地尝试和调制解调器进行通信。例 1-35 中的 TTY1 会话代表线路 1 上的 TTY 会话，这是连有调制解调器的线路。

例 1-35 调试命令 **debug confmodem** 和 **debug modem** 输出结果

```

Router#debug modem
Modem control/process activation debugging is on
Router#debug confmodem
Modem Configuration Database debugging is on
Router#
06:03:15: TTY1: autoconfigure probe started
06:03:18: TTY1: detection speed (38400) response .....
06:03:21: TTY1: detection speed (19200) response .....
06:03:24: TTY1: detection speed (9600) response .....
06:03:27: TTY1: detection speed (2400) response .....
06:03:30: TTY1: detection speed (1200) response .....
06:03:34: TTY1: detection speed (300) response .....
06:03:34: TTY1: No modem found
06:03:34: TTY1: autoconfigure probe started
06:03:37: TTY1: detection speed (38400) response .....
06:03:40: TTY1: detection speed (19200) response .....
06:03:43: TTY1: detection speed (9600) response .....
06:03:46: TTY1: detection speed (2400) response .....
06:03:49: TTY1: detection speed (1200) response .....
06:03:53: TTY1: detection speed (300) response .....
06:03:53: TTY1: No modem found
06:03:53: TTY1: autoconfigure probe started
    
```

处理电缆问题后可以看到如例 1-36 中所示的正常工作的线路。**Modem configuration succeeded** 一行和 CTS 信号的存在表明检测到了一个有效的调制解调器。

例 1-36 调试命令 **debug confmodem** 和 **debug modem** 输出结果（续）

```

06:38:21: TTY1: autoconfigure probe started
06:38:25: TTY1: detection speed (38400) response .....
06:38:28: TTY1: detection speed (19200) response .....
06:38:31: TTY1: detection speed (9600) response .....
06:38:34: TTY1: detection speed (2400) response .....
06:38:37: TTY1: detection speed (1200) response .....
06:38:40: TTY1: detection speed (300) response .....
    
```

（待续）

```
06:38:40: TTY1: No modem found
06:38:40: TTY1: CTS came up on IDLE line
06:38:40: TTY1: autoconfigure probe started
06:38:41: TTY1: detection speed (38400) response ---OK---
06:38:44: TTY1: Modem type is usr_sportster
06:38:44: TTY1: Modem command: --AT&F&C1&D2&M4&K1&B1S0=1H0--
06:38:44: TTY1: Modem configuration succeeded
06:38:46: TTY1: detection speed (38400) response ---OK---
06:38:46: TTY1: Done with modem configuration
```

最后，请参考例 1-37 中完整的配置。

例 1-37 配置带有一台连接在其辅助端口上的调制解调器的路由器

```
hostname router
!
ip subnet-zero
ip host modem 2001 201.201.201.1
!

interface Loopback0
ip address 201.201.201.1 255.255.255.0
no ip directed-broadcast
!
interface Ethernet0
no ip address
no ip directed-broadcast
shutdown
!
interface Serial0
no ip address
no ip directed-broadcast
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
shutdown

!
ip classless
!
line con 0
transport input none
line aux 0
modem InOut
modem autoconfigure discovery
transport input all
speed 38400
line vty 0 4
login
!
end

Router#
```

1.7.7 设置 Microsoft Windows 95/98 网络

所有互联网络的目的都是将数据可靠地从一个网络传输到另一个网络。因此，没有经过真实数据和应用程序检验的模型就不能称为可靠。所以，完整地建立网络模型的最后一项就是数据和应用程序的测试。

如前所述，所有 Microsoft Windows OS，像 Windows 95/98/2000 和 NT，都提供用于测试多种网络模型的所有网络协议，其中我们最关心的是 TCP/IP 和 NetBEUI 协议。可以用与 TCP/IP 相关的应用程序（像 Telnet、FTP、TFTP 等）来测试过滤、验证 IP 连通性和利用 TFTP 升级路由器。而 NetBEUI 协议则是用来测试桥接和 DLSW 的配置情况。

1. Windows 95/98 的 TCP/IP 配置概述

如果工作站或笔记本电脑没有安装 TCP/IP，可以参照下列步骤安装 TCP/IP：

- 第1步 根据生产商的说明安装网卡（NIC）。一般说来，生产商说明手册都包含有关于如何安装 TCP/IP 的内容。这在后面的附录中也有说明。
- 第2步 点击开始/设置/控制面板/网络。当“网络”对话框出现以后，点击“添加”按钮。
- 第3步 弹出一个菜单，内容包括“客户”，“适配器”，“协议”和“服务”。选择“协议”。与 Windows 配套的 TCP/IP 协议的生产商是 Microsoft，因此，点击“Microsoft”，选择“TCP/IP”。Windows 将会提示下面的安装过程，完成后会重启机器。在配置 IP 时不需要重新引导是 Windows 2000 的许多改进之一。
- 第4步 机器重启后，右键点击桌面上的“网络邻居”，选择“属性”。
- 第5步 “网络”对话框再次出现。第4步不过是该对话框的一个快捷方式而已。在“设置”选项卡中选择“TCP/IP”（对应网卡），然后点击“属性”按钮，图 1-15 就是该对话框的形式。
- 第6步 在 TCP/IP 的“属性”窗口，点击顶部的“IP 地址”选项卡。然后再点击“指定 IP 地址”，并输入该主机的 IP 地址。
- 第7步 为保障 IP 工作正常，还必须添加一个缺省的网关。该缺省网关应该是同一局域网段里的一台路由器端口的 IP 地址。该路由器将要处理工作站所有的非本地访问。点击“网关”选项卡然后输入正确的 IP 地址就完成了网关的设置。
- 第8步 最后，如果使用 DNS 服务，点击“DNS 配置”，输入要相关的 DNS 服务器，再点击“添加”按钮就完成了。

在第7或第8步完成之后，工作站就会重启。为了验证刚才所做的设置，点击开始/程序/MS-DOS 方式。这样，工作站就工作在 DOS 会话模式下。通过使用 ping 命令来测试 IP 功能，该命令还可以通过点击开始/运行，再键入 ping x.x.x.x。要测试 DNS 功能，要用 DNS 的域名而不是实际的 IP 地址来 ping 一个 IP 主机即可。

2. Windows 95/98 的 NetBEUI 配置概览

另外需要安装基于主机的协议是 NetBEUI。除了能够使 Windows 实现文件和打印机共享之外，该协议还提供了很不错的应用程序用于测试 DLSw 和桥接功能。NetBEUI 协议是不可

路由的，没有指定的网络层地址。为了将这些协议从一个网络发送到其他网络，就必须使用桥接或 DLSw。点击“开始/查找/计算机”。输入任意名字，点“查找”按钮开始查找。这将把发送全路由探测帧。探测帧将刷新 DLSw 连接，也包括源路由桥接、透明桥接以及转换桥接等的连接。很明显 NetBEUI 在实验里非常有用。

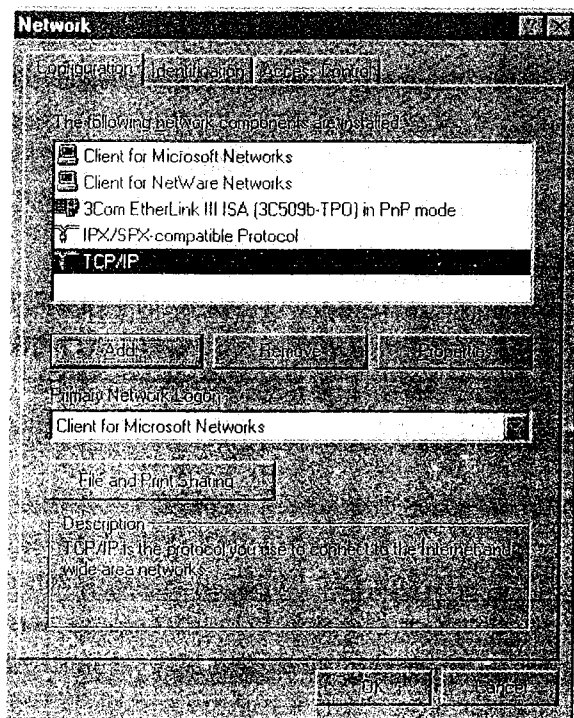


图 1-15 网络对话框

配置 NetBEUI，应遵从如下步骤：

- 第 1 步 右键点击桌面上的“网络邻居”图标，选择“属性”。
- 第 2 步 出现“网络”对话框。点击“添加”按钮。
- 第 3 步 弹出来一个菜单，内容包括“客户”，“适配器”，“协议”和“服务”。选择“协议”。点击“Microsoft”作为生产商，再选择“NetBEUI”。Windows 会提示剩下的安装过程，完成后会要求重启工作站。
- 第 4 步 工作站重启之后，右键点击“网络邻居”图标，选择“属性”。
- 第 5 步 再次出现“网络”对话框。点击顶部的“标识”选项卡，输入工作站名，该名将会出现在 DLSw 中作为标识符。
- 第 6 步 最后，必须要允许 Microsoft 文件与打印机共享才能在模型中建立端到端的会话。点击“添加”按钮，这一次是要增加一项服务，因此点击“服务”选项。
- 第 7 步 要增加的是 Microsoft 文件与打印机共享，因此在此选项上点击一下，Windows 就会安装此项服务。
- 第 8 步 当允许了文件与打印机共享之后，点击“Microsoft 文件服务”和“打印共享”

加以确认一下，这两项都应该是选中了的。然后，打开 Windows Explorer 资源管理器，右键点击某个需要共享的驱动器，选择“共享”项，如果原来是没有共享的，就点击共享项，而且最好加上密码。

警告 如果是连接到当地的电缆调制解调器 (cable modem) 的提供商，请用密码保护自己的硬盘或其他共享文件资源。电缆调制解调器的应用是以本地访问的网络广播机制来工作的。因此，电缆调制解调器网段内的每个用户都能够看见你的 PC 并访问所有共享资源。

要测试 NetBEUI 以及 Microsoft 文件与打印机共享，需要两台工作站。为测试该配置，选择“开始/查找”，输入第 5 步中指定的标识。这样就能找到该工作站，点击该工作站的名称就会建立与该工作站的一个会话，所有共享的资源都会显示出来。点击它们就能在另外的一台工作站上使用这些共享资源了。

1.8 第1章实验指南：简介

到现在为止，本章已经讨论完了建立网络互联模型所需要的关键组件。现在就要实践应用所学的知识了。没有实际应用就很难掌握上述各种概念。

本章学习的目的是建立网络互联模型。开始建立各种网络和协议的模型之前，必须要准备好这一章里讨论的网络模型的各个关键组件。随着互连网络建模的深入，实验也会变得越来越难。本书最后是 5 个完整的 CCIE 实验。这些实验提供给大家在准备 CCIE 实验考试时用于自我提高的，因此没有给出答案。

实验分成两个主要部分的：实验练习或内容描述，然后是实验步骤。首先，在不看第 2 部分实验步骤的情况下试着完成实验的第 1 部分。每一个实验都包括了该章的主题，有一些实验可能还会介绍一些新的概念。

1.9 实验 1：密码恢复——第 1 部分

1.9.1 实验说明

路由器操作中经常会遇到需要进行密码恢复的情况。在实验室中建立网络模型时，至少需要 3 台路由器。大部分设备都可能被别人使用过，对这些设备需要特权访问的权限，或者用户有时可能忘记密码。这两种情况下，都需要进行密码恢复。

1.9.2 实验内容

假设拿到很多用于建立复杂网络模型的路由器。而这些路由器又都是其他工程师已经配置过的，因而在其密码保护的 NVRAM 中存有旧的配置信息。本实验的内容就是保存设置信息之后删除当前配置信息。

1.9.3 实验目的

- 检查路由器的原始配置信息。
- 删除路由器的配置信息。路由器这时应该没有配置信息，开机时，进入设置模式中。

1.9.4 所需设备

- 一台 Cisco 路由器。
- 一套 Cisco 配置套件（一条全反电缆以及用于和 PC/笔记本电脑的串口相连的转接头）。
- 一台装有终端仿真软件的 PC 或笔记本电脑。
- 到目前为止的所有实验只需要一台 PC 或笔记本电脑，标准的配置套件以及终端仿真软件。我们只在这个实验中才会提到这些最基本的配置设备。

1.9.5 物理设计与实验准备

所用的路由器中必须要有以前所做的配置以及密码设置。

1.10 实验 1：密码恢复——第 2 部分

1.10.1 实验步骤

前面解释过，各种路由器系统平台的密码恢复都非常类似。因此，这一章先前讨论过的密码恢复过程，只要稍加改动就可以用于这个实验。

下面所用的方法适用于下面这些路由器：

- Cisco 2000 系列
- Cisco 2500 系列
- Cisco 3000 系列
- Cisco 4000 系列（CPU 为 Motorola 的 680x0）
- Cisco 7000 系列（RP 卡上 ROM 中运行的是 10.0 及更新版本的 Cisco IOS）
- IGS 系列（ROM 中运行的是 9.1 或更新版本的 Cisco IOS）

第 1 步 通过 Cisco 的全反电缆将装有终端仿真软件的 PC 或 PDA 与路由器的控制台端口相连。

第 2 步 路由器加电。

第 3 步 发送一个暂停信号给路由器。

第 4 步 确定路由器上 ROM 监控的类型，是否支持 CONREG 命令组件？

——如果是基本 ROM 监控，将第 6 位设置为 1：>O/R 0x2142。然后用 initialize

命令重启路由器。

- 如果支持 CONFREG，运行 CONFREG 功能组件：>**CONFREG.** 用默认值或者是 **Enter** 回答每一个出现的问题直到出现提示：**Enable ignore system config info**，回答“yes”，这也是把寄存器的第 6 位设为了 1。然后用 **reset** 命令重启路由器。

第 5 步 路由器重新引导后就会试图进入设置模式，按 **CTRL-C** 退出设置模式。

第 6 步 进入特权模式，用 **show startup-config** 命令检查 NVRAM 中的配置内容。

在这个实验中，希望通过一次密码恢复操作来获得访问服务器的特权操作等级。这里的访问服务器名是 *skynet_access_1*。

首先，将装有终端仿真软件的 PC 或笔记本电脑与路由器的控制台端口互连。关掉路由器，然后再次加电。在初始化的前 60s 内，从终端仿真器上发送一个暂停信号。例 1-38 为成功向路由器发送暂停信号的示例。

例 1-38 一次成功的暂停

```
System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
Copyright (c) 1986-1995 by cisco Systems
2500 processor with 14336 Kbytes of main memory

Abort at 0x10EA888 (PC)
>
```

密码恢复时如何通过终端仿真软件发送暂停信号是很常见的一个问题。下面这些技巧可能对这方面有所帮助：

- 首先要确保的是所用的 Cisco 全反电缆已经正确牢固地连接到路由器的控制台端口上。
- 如果 PDA 或笔记本电脑上用的仿真软件是 Windows 95/98/2000 的超级终端，暂停信号是通过按下 **Function** 功能键和 **Break** 键来实现，这个 **Break** 键通常是在 Page Down 键或 Pause 键上。本章前面的表 1-8 列出了所有的终端仿真软件、平台以及操作系统下标准的暂停键的组合方式。
- 在超级终端中，暂停信号是通过按下 **Ctrl-Break/Pause** 键来产生。
- 在 Windows NT 中，必须用一个功能键设置 NT 来发送暂停信号，输入字符 **^\$B**（Shift 6, Shift 4 和大写的 B）。新的 HyperTerm 5.0 用户版在 Windows NT 环境下无需任何附加设置就可以发送暂停信号。
- 使用其他终端仿真软件时，一定要查询制造商手册，弄清楚暂停信号是如何发送的。

看到中断信息之后就可以继续。如果忘记了何种类型的路由器支持 CONFREG，可在这个时候输入？查看 CONFREG 的功能组件信息。例 1-39 中就是？在访问服务器上输出的示例。

例 1-39 不支持 CONFREG 的路由器上？命令示例

```
>?
$          Toggle cache state
B [filename] [TFTP Server IP address | TFTP Server Name]
           Load and execute system image from ROM or from TFTP server
```

（待续）

```

C [address] Continue execution [optional address]
D /S M L V Deposit value V of size S into location L with modifier M
E /S M L Examine location L with size S with modifier M
G [address] Begin execution
H Help for commands
I Initialize
K Stack trace
L [filename] {TFTP Server IP address | TFTP Server Name}
Load system image from ROM or from TFTP server, but do not
begin execution
O Show configuration register option settings
P Set the break point
S Single step next instruction
T function Test device (? for help)

```

Deposit and Examine sizes may be B (byte), L (long) or S (short).

Modifiers may be R (register) or S (byte swap).

Register names are: D0-D7, A0-A6, SS, US, SR, and PC

>

例 1-40 是同样的一次暂停之后输入?命令的情况，这个路由器支持 CONFREG。

例 1-40 支持 CONFREG 的路由器上一次成功的暂停之后再输入?的结果

```

System Bootstrap, Version 5.3(16) [richardd 16], RELEASE SOFTWARE (fc1)
Copyright (c) 1996 by cisco Systems, Inc.
C4500 processor with 16384 Kbytes of main memory

```

```

monitor: command "boot" aborted due to user interrupt
rommon 1 >
rommon 1 > ?
alias                set and display aliases command
boot                 boot up an external process
break                set/show/clear the breakpoint
confreg              configuration register utility
cont                 continue executing a downloaded image
context              display the context of a loaded image
cookie               display contents of cookie PROM in hex
dev                  list the device table
dir                  list files in file system
dis                  disassemble instruction stream
dnld                 serial download a program module
frame                print out a selected stack frame
help                 monitor built in command help
history              monitor command history
meminfo              main memory information
repeat               repeat a monitor command
reset                system reset
set                  display the monitor variables
stack                produce a stack trace
sync                 write monitor environment to NVRAM
sysret               print out info from last system return
unalias              unset an alias
unset                unset a monitor variable
rommon 2 >

```

将寄存器的第 6 位置为 1，以便在引导时忽略 NVRAM 的配置信息。这是通过输入 **O/R hex-value** 然后再 **Enter** 来实现的。接着，输入 **init** 重新引导路由器。例 1-41 就是这一个过程的示例。

例 1-41 设第 6 位为 1 忽略 NVRAM 之后再执行 initialization 命令

```
System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
Copyright (c) 1986-1995 by cisco Systems
2500 processor with 14336 Kbytes of main memory

Abort at 0x10205A6 (PC)
>o/r 0x2142
>init

System Bootstrap, Version 5.2(8a), RELEASE SOFTWARE
Copyright (c) 1986-1995 by cisco Systems
```

在支持 CONFREG 的路由器上，这个过程是一目了然的。例 1-42 是这个过程在这样的平台上进行的示例，这里用的路由器是 Cisco 4700。

例 1-42 设第 6 位为 1 忽略 NVRAM 之后再执行 reset 命令

```
rommon 1 > confreg

Configuration Summary
enabled are:
load rom after netboot fails
console baud: 9600
boot: image specified by the boot system commands
or default to: cisco2-C4500

do you wish to change the configuration? y/n [n]: y
enable "diagnostic mode"? y/n [n]: n
enable "use net in IP bcast address"? y/n [n]: n
disable "load rom after netboot fails"? y/n [n]: n
enable "use all zero broadcast"? y/n [n]: n
enable "break/abort has effect"? y/n [n]: n
enable "ignore system config info"? y/n [n]: y
change console baud rate? y/n [n]: n
change the boot characteristics? y/n [n]: n

Configuration Summary
enabled are:
load rom after netboot fails
ignore system config info
console baud: 9600
boot: image specified by the boot system commands
or default to: cisco2-C4500

do you wish to change the configuration? y/n [n]: n

You must reset or power cycle for new config to take effect
rommon 2 > reset

System Bootstrap, Version 5.3(16) [richardd 16], RELEASE SOFTWARE (fc1)
Copyright (c) 1996 by cisco Systems, Inc.
```

路由器重新引导之后就没有 running-configuration 文件。当然，路由器仍然在 NVRAM 中保存着这一份配置信息。要查看这些配置的情况，首先进入特权模式，然后输入 **show**

startup-configuration 命令即可。

如果想保存现有的配置，可以按下面的步骤进行，一定要严格按照下面的顺序操作：

第 1 步 输入 **enable** 进入特权模式。

第 2 步 用 **copy startup-config running-config** 命令把引导配置信息拷贝到运行配置信息中去。

第 3 步 进入配置模式，用 **configure-register 0x2102** 命令把引导寄存器改回它的常规值。

第 4 步 由于当前所有接口默认都处在停止状态，因此需要激活所有接口。

第 5 步 配置一个新的特权密码。

第 6 步 用 **copy running-config startup-config** 命令保存现有的配置信息。

1.11 实验 2：Catalyst 5500 交换机的密码恢复——

第 1 部分

1.11.1 实验说明

同路由器一样，很多时候交换机上也需要进行密码恢复的操作。

1.11.2 实验内容

用于实验的路由器一起送来的还有一台用过的 Catalyst 5500 交换机，而且也设置了密码。要想在实验室中使用这台交换机，首先要恢复它的密码，然后为了安全访问的原因，还应该为它设置一个新的密码。

1.11.3 实验目的

- 在 Catalyst 5500 交换机上进行密码恢复的操作。
- 在 Catalyst 5500 交换机上设置一个新的密码。

1.11.4 所需设备

- 一台 Cisco Catalyst 交换机—Catalyst 5000、5500 或 4000 系列皆可。

1.11.5 物理设计与实验准备

交换机必须已使用，里面有原来的配置信息以及设置了密码。

1.12 实验 2：Catalyst 5500 交换机的密码恢复——

第 2 部分

1.12.1 实验步骤

初始化的前 30 秒时间内，密码和特权密码都是 **Enter** 键。一但出现 **Cisco Systems Console** 信息，就可以进行配置了。

首先，进入特权模式，交换机会提示输入密码，按下 **Enter** 键就行。然后用 **set password** 命令设置一个新的密码。这时，交换机要求输入旧的密码，按下 **Enter**。最后，在 30 秒时间过去之前，用 **set enablepass** 命令设置一个新的特权密码。交换机会最后一次请求输入旧的密码，再一次按下 **Enter** 键。用来设置密码的 30 秒时间很快就过去了。如果这么短时间设置密码有困难，可以先把新密码就设成 **Enter** 键。这样在提示输入旧密码和新密码时就只需两次 **Enter** 键即可。例 1-43 就是这一过程的按键输入情况。

例 1-43 Catalyst 5500 上的密码恢复

```
Console> en
Enter password:                               ←--Enter key pressed
Console> (enable) set pass
Enter old password:                           ←--Enter key pressed
Enter new password:                           ←--Enter key or new password
Retype new password:                          ←--Enter or new password
Password changed.
Console> (enable) set enablepass
Enter old password:                           ←--Enter key pressed
Enter new password:                           ←--Enter key or new password
Retype new password:                          ←--Enter or new password
Password changed.
Console> (enable)
```

1.13 实验 3：升级 IOS 以及从 TFTP 服务器恢复

配置——第 1 部分

1.13.1 实验说明

有时需要对 Cisco IOS 进行升级，但是却又无法使用本地访问路由器。这时可以将一台临近的 Cisco 路由器作为 TFTP 服务器完成升级工作。

1.13.2 实验内容

在这个实验中，要从一台路由器的 FLASH 存储器中升级另一台路由器的 IOS，此外，还要求从路由器拷贝配置文件到 TFTP 服务器，同时也需要路由器从 TFTP 服务器下载配置文件。

1.13.3 实验目的

- 以图 1-16 为参考，对所示网络进行配置——这一次不需要配置环路地址。
- 为了防止路由器 rosewell 的 Cisco IOS 将来意外“崩溃”，应将新的 Cisco IOS 从路由器 ufo 拷贝到 rosewell。
- 利用 TFTP 将引导配置文件从路由器 ufo 传输到 TFTP 服务器 172.16.16.254。用微软的 Wordpad 文本编辑器对配置文件进行编辑，把主机名改成 w-balloon。然后再利用 TFTP 把配置文件传回到路由器引导配置中去。重新引导路由器。

1.13.4 所需设备

- 两台 Cisco 路由器和一个以太集线器
- 一台装有终端仿真软件，TFTP 软件且配置了 TCP/IP 的工作站，如图 1-16 所示。

1.13.5 物理设计与实验准备

必须在工作站上如图 1-16 所示配置好 TCP/IP。

物理拓扑与 IP 地址映射

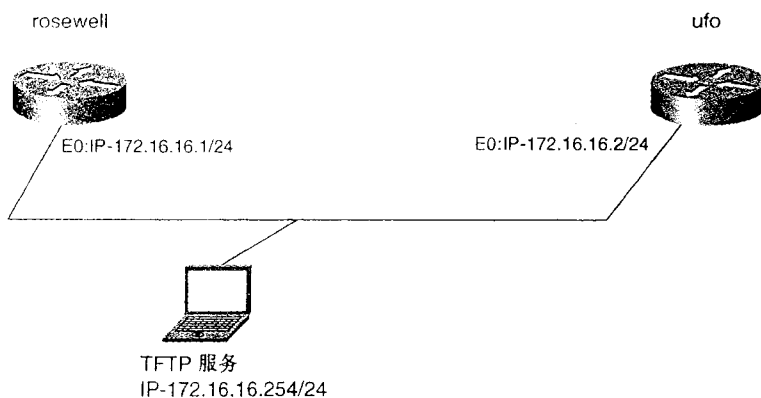


图 1-16 实验 3：物理拓扑与 IP 地址映射

1.14 实验3：升级IOS 以及从TFTP 服务器恢复配置——第2部分

1.14.1 实验步骤

实验的第1步是按照图1-16对所有的设备进行配置，也就是将一台工作站连接到同一物理网段上。当所有的设备都可以相互 ping 通之后，就可以进行下面的工作了。

本实验引入了一个新的概念——将路由器配置成一台 TFTP 服务器。要想让路由器可以作为 TFTP 服务器使用，需要在全局配置模式下输入 **tftp-server [flash | rom] filename** 命令。在例1-44中，首先是用 **show flash** 命令查看 FLASH 存储器中可用的文件（要记住文件名）。在配置模式中，输入 **tftp-server flash filename** 命令。为了避免在 FLASH 升级过程中出现键入错误，建议大家对文件名采取剪辑然后粘贴的方法。

例 1-44 将一台路由器配置成 TFTP 服务器

```
ufo#
ufo#show flash

System flash directory:
File Length Name/status
  1 8102652 c2500-js-l_112-16.bin
[8102716 bytes used, 285892 available, 8388608 total]
8192K bytes of processor board System flash (Read ONLY)

ufo#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ufo(config)#tftp-server flash c2500-js-l_112-16.bin
ufo(config)#^Z
ufo#
```

现在切换到路由器 rosewell 上，可以开始通过路由器 ufo 的 TFTP 服务器来升级 IOS 了。为此，输入 **copy tftp flash** 命令，再回答相应的提示，路由器 ufo 是 TFTP 服务器，新的 IOS 文件名是 c2500-js-l_112-16.bin。例1-45就是通过 ufo 升级 rosewell IOS 的示例。

例 1-45 通过 TFTP 升级 IOS

```
rosewell#
rosewell#copy tftp flash          ←Copying from the server to the router
**** NOTICE ****

Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the copy.
Routing functionality will not be available during that time.
If you are logged in via telnet, this connection will terminate.
Users with console access can see the results of the copy operation.
.... ***** .....
```

(待续)

```

Proceed? [confirm]y

System flash directory:
File Length Name/status
 1 8034308 c2500-js-l_112-11.bin
[8034372 bytes used, 8742844 available, 16777216 total]
Address or name of remote host [255.255.255.255]? 172.16.16.2
Source file name? c2500-js-l_112-16.bin
Destination file name [c2500-js-l_112-16.bin]? c2500-js-l_112-16.bin
Accessing file 'c2500-js-l_112-16.bin' on 172.16.16.2...
Loading c2500-js-l_112-16.bin .from 172.16.16.2 (via Ethernet0): ! [OK]

Erase flash device before writing? [confirm]y
Flash contains files. Are you sure you want to erase? [confirm]y

Copy 'c2500-js-l_112-16.bin' from server
as 'c2500-js-l_112-16.bin' into Flash WITH erase? [yes/no]yes

00:01:15: %SYS-5-RELOAD: Reload requested ←the router reloads
SERVICE_MODULE(1): self test finished: Passed
%SYS-4-CONFIG_NEWER: Configurations from version 11.2 may not be correctly under
stood.
%FLH: c2500-js-l_112-16.bin from 172.16.16.2 to flash ...

System flash directory:
File Length Name/status
 1 8034308 c2500-js-l_112-11.bin
[8034372 bytes used, 8742844 available, 16777216 total]
Accessing file 'c2500-js-l_112-16.bin' on 172.16.16.2...
Loading c2500-js-l_112-16.bin .from 172.16.16.2 (via Ethernet0): ! [OK]

Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
ee ...erased ←Flash is erased
Loading c2500-js-l_112-16.bin from 172.16.16.2 (via Ethernet0): !!!!!!!!!!!!!!!
<<<text omitted>>>
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 8102652/16777216 bytes]

Verifying checksum... OK (0x8DCB)
Flash copy took 0:04:40 [hh:mm:ss]
%FLH: Re-booting system after download
F3: 8004052+98568+315656 at 0x3000060
<<<text omitted>>>

00:00:23: %SYS-5-RESTART: System restarted ..
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 11.2(16), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1998 by cisco Systems, Inc.
Compiled Tue 06-Oct-98 11:54 by ashah
rosewell>show flash

System flash directory:
File Length Name/status
 1 8102652 c2500-js-l_112-16.bin ←New IOS
[8102716 bytes used, 8674500 available, 16777216 total]
16384K bytes of processor board System flash (Read ONLY)

rosewell>
    
```

下载到 FLASH 之后，路由器在重启之前还要先验证文件校验和的值。在上面例子的最后，用 **show flash** 命令来确认新的 IOS 已经下载完毕。

机名改为 w-balloon，然后再把这个文件拷贝回原来的地方去。为此，输入 **copy startup-config tftp** 命令，接着按照出现的提示进行相应的操作。在做这之前，最好是先 **ping** 一下 TFTP 服务器，看是否能通。例 1-46 就是 **ping** 命令和 **copy** 命令的使用示例。

例 1-46 将引导配置文件拷贝到 TFTP 服务器

```
ufo#ping 172.16.16.254

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.16.254, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
ufo#copy startup-config tftp
Remote host []? 172.16.16.254
Name of configuration file to write [ufo-conf]?          ←----- carriage return
Write file ufo-conf on host 172.16.16.254? [confirm]y
Writing ufo-conf !! [OK]
ufo#
```

有一点要注意，TFTP，顾名思义，是一种简单协议。也就是说，TFTP 无法覆盖文件或者是在文件复制开始之后再提示用户输入信息的。如果再做一次上面的操作过程，就会出现错误，因为文件由于上一次的拷贝已经存在了。这个错误有可能出现，也有可能不会出现，一些版本的 TFTP 拥有自动覆盖已有文件的功能。例 1-47 就是欲复制的文件存在，无法进行覆盖时的错误信息。

例 1-47 TFTP 复制错误，复制文件存在

```
ufo#copy startup-config tftp
Remote host []? 172.16.16.254
Name of configuration file to write [ufo-conf]?          ←----- carriage return
Write file ufo-conf on host 172.16.16.254? [confirm]y
Writing ufo-conf
TFTP: error code 0 received - File exists
[Failed]
ufo#
```

文件成功复制到 PC 之后，用 Microsoft Wordpad 文本编辑器对其进行编辑，找到 HOSTNAME 字段，把 ufo 改为 w-balloon。最后，用 **copy tftp startup-config** 命令再把文件复制回路由器。完成之后，再用 **show startup-config** 命令对其进行查看。要激活新的配置信息，可以重新引导路由器或者是执行一次 **copy startup-config running-config** 命令。做完之后，主机名就从 ufo 改为了 w-balloon。例 1-48 就是这一过程的示例。

例 1-48 从一台 TFTP 服务器复制配置文件

```
ufo#copy tftp startup-config
Address of remote host [255.255.255.255]? 172.16.16.254
Name of configuration file [ufo-conf]?
Configure using ufo-conf from 172.16.16.254? [confirm]y
Loading ufo-conf from 172.16.16.254 (via Ethernet0): !
[OK - 564/32723 bytes]
[OK]
ufo#
```

(待续)

```
%SYS-5-CONFIG_NV: Non-volatile store configured from ufo-config by console tftp from 172.16.16.254

ufo#show startup-config
Using 564 out of 32762 bytes
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname W-BALLOON
!
interface Ethernet0
 ip address 172.16.16.2 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
 no ip address
 no ip route-cache
 no ip mroute-cache
!
interface Serial1
 no ip address
ufo#
```

1.15 实验4：访问服务器的配置——第1部分

1.15.1 实验说明

访问服务器能够对很多路由器实施带外管理，这样就可以轻松地通过一台设备来访问和配置很多路由器与交换机。

1.15.2 实验内容

这个实验里，配置一台访问服务器通过反向 Telnet 访问所有的路由器和交换机。这样可以无需手动切换路由器与交换机之间的控制台电缆来完成这些设备的配置。

1.15.3 实验目的

- 在访问服务器上加一个环路地址 201.201.1.1。
- 配置 IP 主机表，包括：
 - 主机名为 r1，对应到这一组路由器中的第 1 台反向 Telnet。
 - 主机名为 r2，对应到这一组路由器中的第 2 台反向 Telnet，以此类推。
- 使异步线路上的冲突最小化。
- 避免所有远程路由器上的会话超时。

1.15.4 所需设备

- 两台 Cisco 路由器，一台 Cisco 2509/2511 或者是具有异步模块的 2600 系列路由器都可以作为访问服务器。访问路由器还需要八爪电缆。

1.15.5 物理设计与实验准备

- 通过 Cisco 黑色或浅蓝色反接电缆将工作站的串口与 Cisco 2509-2511 连接。
- 把八爪电缆中标号为 1 的线缆与 R1，也就是组中第 1 台路由器的控制台端口连接好，依此类推，连接好实验中其他路由器和交换机。

1.16 实验 4：访问服务器的配置——第 2 部分

1.16.1 实验步骤

所有控制台端口都与访问服务器的八爪电缆连好之后，就可以开始配置访问服务器支持反向 Telnet 会话了。要配置反向 Telnet 会话，首先要知道我们使用的 TTY 线路的绝对线路号或 TTY 中的相对线路号。用 **show line** 命令就可以显示线路信息。在例 1-49 中，TTY 会话的线路标号是从 1 到 16。这个实验中有 2 到 5 台路由器，因此把 2001 到 2005 的值作为反向 Telnet 会话的 Telnet 端口号。

例 1-49 show line 命令的显示信息

Router#show line											
Tty	Typ	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	1	0/0	-
1	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
2	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
3	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
4	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
5	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
6	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
* 7	TTY	9600/9600	-	-	-	-	-	0	2	0/0	-
8	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
9	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
10	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
11	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
12	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
13	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
14	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
15	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
16	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
17	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
18	VTY		-	-	-	-	-	0	0	0/0	-
19	VTY		-	-	-	-	-	0	0	0/0	-
20	VTY		-	-	-	-	-	0	0	0/0	-
21	VTY		-	-	-	-	-	0	0	0/0	-

(待续)

Tty Typ	Tx/Rx	A Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
22 VTY						0	0	0/0	
Router#									

知道绝对线路号之后，就可以配置访问服务器支持反向 Telnet 了。在配置模式中，先配置访问服务器的主机名为 **access-server**。接着是用于反向 Telnet 会话的线路号，输入 **ip host r1 200x 201.201.1.1**， x 的范围是 1 到 5。还需要做的就是加上一个 IP 地址为 201.201.1.1/24 的 Loopback 0 接口。例 1-50 就是这一过程的示例。

例 1-50 配置反向 Telnet 会话的 IP 主机名

```
Router#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname access-server
access-server(config)#ip host r1 2001 201.201.1.1
access-server(config)#ip host r2 2002 201.201.1.1
access-server(config)#ip host r3 2002 201.201.1.1
access-server(config)#ip host r3 2003 201.201.1.1
access-server(config)#ip host r4 2004 201.201.1.1
access-server(config)#ip host r5 2005 201.201.1.1
access-server(config)#interface loopback 0
access-server(config-if)#ip address 201.201.1.1 255.255.255.0
access-server(config-if)#exit
access-server(config)#
```

配置访问服务器的最后一步就是设置线路支持反向 Telnet。在相应的线路上通过 **transport input all** 命令就可以完成这一设置。在这个实验里，我们还要另外做一些工作，加上一条 **no exec** 命令以减少线路上的冲突。这只是为了方便，并不是配置反向 Telnet 会话所必需的。例 1-51 完成了访问服务器反向 Telnet 的配置工作。

例 1-51 配置反向 Telnet 的绝对线路号，并且屏蔽 Exec

```
access-server(config)#line 1 5
access-server(config-line)#transport input all
access-server(config-line)#no exec
access-server(config-line)#^Z
access-server#
```

要测试这个配置的效果，键入 **r1**，连接到 R1 的会话打开。如果会话被拒绝，记住执行一条 **clear line x** 命令，这里的 x 是被拒绝了会话的绝对线路号。例 1-52 就是整个访问服务器的配置过程。

例 1-52 将一台路由器配置为访问服务器的完整过程

```
access-server#wr t
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
```

(待续)

立和实际环境很相似的帧中继网络。

1.17.2 实验内容

在这个实验中，要配置一个全连接的帧中继网络，这里侧重的是帧中继的交换方面，而不是终端或 DTE 设备的配置问题。

1.17.3 实验目的

- 配置一台 Cisco 路由器完成帧中继交换功能，如图 1-17 所示。
- 如图所示用 ANSI LMI 配置所有的 PVC。PVC 的映射如下：
 - DLCI 112 映射到 DLCI 21.
 - DLCI 113 映射到 DLCI 31.
 - DLCI 32 映射到 DLCI 23.

1.17.4 所需设备

- 3 台具有串行接口的 Cisco 路由器，还要一台 Cisco 路由器作为帧中继交换机，它必须具有 3 个可用的串行接口。
- 总共 6 条串行电缆，或者是 3 套 DTE 到 DCE 串行电缆。

1.17.5 物理设计与实验准备

图 1-17 就是该实验的物理拓扑图。

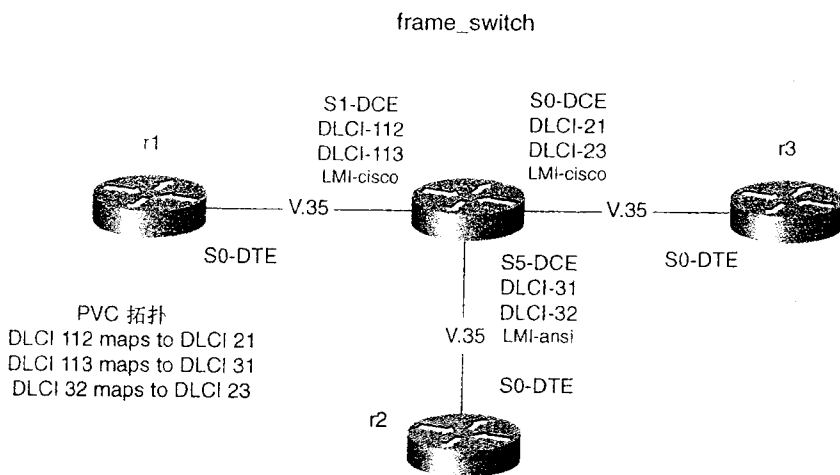


图 1-17 帧中继交换的物理拓扑

1.18 实验5：帧中继交换机的配置——第2部分

1.18.1 实验步骤

该实验中的帧中继交换机是用来提供全连接的帧中继服务功能。全连接的帧中继服务不仅仅具有通往中央站点的PVC，而且具有通往任何其他站点的PVC。这种情况在涉及到大量的站点时就会出现扩展性的问题，在多点网络中还会有一些第3层路由的问题发生，如水平分割等。这些问题会在第5章深入地解释。

配置帧中继交换的步骤如下：

第1步 引导帧中继交换。

第2步 配置接口的LMI与帧中继类型。

第3步 用 **frame-relay route** 命令对PVC进行配置。

首先，在配置模式下用全局命令 **frame-relay switching** 引导帧中继交换。然后，用 **encapsulation frame-relay** 命令如图1-17所示配置帧中继交换的串行接口类型，同时在串口5上用 **frame-relay intf-type dce** 命令和 **frame-relay lmi-type ansi** 命令对串口进行配置。例1-53就突出显示了帧中继交换的配置示例。

例 1-53 帧中继点对多点配置示例

```
hostname frame_switch
!
frame-relay switching          ← Enables Frame Relay switching
!
interface Ethernet0
  no ip address
  shutdown
!
interface Serial0
  no ip address
  encapsulation frame-relay     ← Sets Frame encapsulation
  clockrate 56000              ← Sets the clockrate, needed for DCE interface
  frame-relay intf-type dce     ← Sets Frame Relay to a DCE interface
  frame-relay route 21 interface Serial1 112 ← Creates and maps DLCI 21 to DLCI
112 on Serial 1
  frame-relay route 23 interface Serial5 32 ← Creates and maps DLCI 23 to DLCI 32
on Serial 5
!
interface Serial1
  no ip address
  encapsulation frame-relay
  clockrate 56000
  frame-relay intf-type dce
  frame-relay route 112 interface Serial0 21 ← Creates and maps DLCI 112 to DLCI
21 on Serial 0
  frame-relay route 113 interface Serial5 31 ← Creates and maps DLCI 113 to DLCI
31 on Serial 5
```

(待续)

```

!
<<<text omitted>>>
!
interface Serial5
 no ip address
 encapsulation frame-relay
 clockrate 56000
 frame-relay lmi-type ansi          ←Sets the LMI type to ANSI versus Cisco
 frame-relay intf-type dce
 frame-relay route 31 interface Serial1 113    ←Creates and maps DLCI 31 to DLCI
 113 on Serial 1
 frame-relay route 32 interface Serial0 23    ←Creates and maps DLCI 32 to DLCI 23
 on Serial 0
!
<<<text omitted>>>
end
    
```

如果想要查看帧中继交换机是否正常工作，需要对 DTE，也就是网络的路由器进行配置。完成之后，PVC 会成为“active”。例 1-54 是 R1, R2 和 R3 的帧中继配置情况。

例 1-54 R1、R2 和 R3 的有效配置部分

```

hostname r1
!
interface Serial0
 ip address 172.16.17.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.17.2 112 broadcast
 frame-relay map ip 172.16.17.3 113 broadcast
!

hostname r2
!
interface Serial0
 ip address 172.16.17.2 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 172.16.17.1 21 broadcast
 frame-relay map ip 172.16.17.3 23 broadcast
!

hostname r3
!
interface Serial0
 ip address 172.16.17.3 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 172.16.17.1 31 broadcast
 frame-relay map ip 172.16.17.2 32 broadcast
 frame-relay lmi-type ansi
!
    
```

做完这些之后，就可以用 **show frame-relay route** 命令来确认一下所有链路是否已经开始工作。同时还可以用 **show frame-relay lmi** 命令来确认 LMI 的工作情况。例 1-55 给出了这两条命令的示例。

例 1-55 show frame-relay route 和 show frame-relay lmi 命令示例

```
frame_switch#show frame-relay route
Input Intf      Input Dlci      Output Intf      Output Dlci      Status
Serial0         21              Serial1          112              active
Serial0         23              Serial5          32               active
Serial1         112             Serial0          21               active
Serial1         113             Serial5          31               active
Serial5         31              Serial1          113              active
Serial5         32              Serial0          23               active
frame_switch#
frame_switch#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DCE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 188          Num Status msgs Sent 188
Num Update Status Sent 0          Num St Enq. Timeouts 0

LMI Statistics for interface Serial1 (Frame Relay DCE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 188          Num Status msgs Sent 188
Num Update Status Sent 0          Num St Enq. Timeouts 0

LMI Statistics for interface Serial5 (Frame Relay DCE) LMI TYPE = ANSI
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Rcvd 185          Num Status msgs Sent 185
Num Update Status Sent 0          Num St Enq. Timeouts 1
frame_switch#
```

1.19 实验 6：远程访问实验室的配置——第 1 部分

1.19.1 实验说明

将调制解调器连接到辅助端口或异步线路上是对路由器、交换机进行带外管理的有效途径。可以在中央站点放置一台访问服务器做为网络的“安全配置网络”。和模拟调制解调器协同工作，访问服务器在中央站点处将所有路由器控制台端口连接起来，为网络提供了一种安全可靠的访问方式。

1.19.2 实验内容

这是第 1 章的最后一个实验，是本书中所有其他实验的出发点。今后实验中的路由器都是用这台访问服务器来配置。为了增强访问的功能，在访问服务器上的最后一个异步端口上接上一台调制解调器，该设备是建立一些目前非常复杂的网络时的关键设备。因此，Skynet 这个名字其实也道出了访问服务器和网络模型的某些特征。

1.19.3 实验目的

- 按图 1-18 所示连接访问服务器。
- 配置一个从访问服务器到所有连在八爪电缆上路由器之间的反向 Telnet 会话。利用 IP 主机名表可以简化反向 Telnet 会话的使用。
- 配置连接在异步线路上的调制解调器，使之接受呼入路由器的模拟拨号会话。

1.19.4 所需设备

- 一台 Cisco 2509/2511、Cisco 2600 或 Cisco 3600 作为访问服务器，配备有异步模块和八爪电缆。
- 1 到 5 台其他路由器（只需一台就可以完成该实验）。
- 一台模拟调制解调器及其接线头和 Cisco 反接电缆。

1.19.5 物理设计与实验准备

异步连接要按照图 1-18 进行。

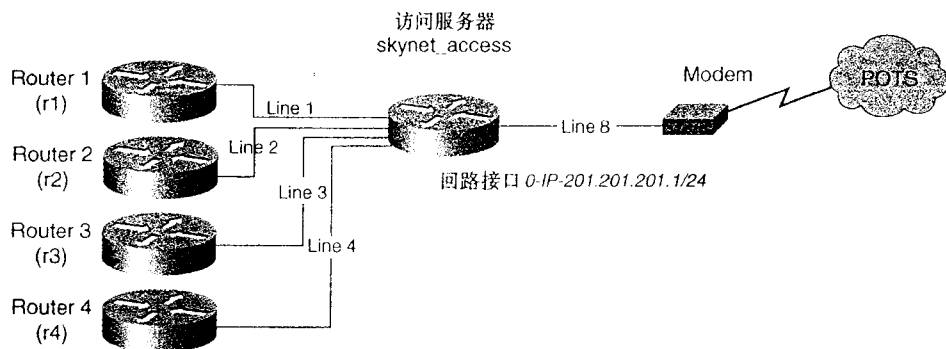


图 1-18 实验 6: 物理拓扑

1.20 实验 6：远程访问实验室的配置——第 2 部分

1.20.1 实验步骤

配置 Skynet 的第一步是如图 1-18 所示连接路由器和调制解调器。物理连接完成之后，通过 **show line** 命令检验物理设备所在绝对线路，如例 1-56 所示。图中要注意左边的一栏——异步线路都用 TTY 表示。

例 1-56 访问服务器上 **show line** 命令示例

例 1-56 访问服务器上 show line 命令示例

Router#show line											
Tty	Type	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	0	0/0	-
1	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
* 2	TTY	9600/9600	-	-	-	-	-	0	1	37/110	-
* 3	TTY	9600/9600	-	-	-	-	-	0	1	3/11	-
4	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
5	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
6	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
7	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
8	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
9	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
10	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
11	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
12	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
13	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
14	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
15	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
16	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
17	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
18	VTY		-	-	-	-	-	0	0	0/0	-
19	VTY		-	-	-	-	-	0	0	0/0	-
20	VTY		-	-	-	-	-	0	0	0/0	-
21	VTY		-	-	-	-	-	0	0	0/0	-
Tty	Type	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
22	VTY		-	-	-	-	-	0	0	0/0	-
Router#											

线路 1 到 4 是用来连接路由器的 TTY 线路。因此，需要加上下列命令：

- **transport input all**
- **no exec**

为了便于反向 Telnet 会话打开，每台路由器都用主机名表指向端口 2001 到 2004。另外为了方便反向 Telnet 会话的使用，要再加上一个环路接口。例 1-57 就是 skynet_access 的配置示例。

例 1-57 访问服务器 skynet_access 的初始配置

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname skynet_access
skynet_access(config)#interface loopback 0          ← Configuring the
skynet_access(config-if)#ip address 201.201.201.1 255.255.255.0 ← loopback
interface
skynet_access(config-if)#exit
skynet_access(config)#
skynet_access(config)#ip host r1 2001 201.201.201.1 ← Configure the IP
skynet_access(config)#ip host r2 2002 201.201.201.1 ← host table
skynet_access(config)#ip host r3 2003 201.201.201.1
skynet_access(config)#ip host r4 2004 201.201.201.1
skynet_access(config)#ip host modem 2005 201.201.201.1
skynet_access(config)#
skynet_access(config)#line 1 4                      ← Configuring the
skynet_access(config-line)#transport input all       ← the line entries to
skynet_access(config-line)#no exec                  ← support telnet
skynet_access(config-line)#^Z
skynet_access#
```

要完成该实验，还需要为调制解调器配置一条线路 Line 5，为此，在 Line 5 中键入下面的命令：

- **transport input all**
- **modem inout**
- **modem autoconfigure discovery**

在例 1-58 中，上面这些命令都加在了 Line 5 中。记住还要加上一个特权密码——否则，远程会话就没有办法进入特权访问模式中对路由器进行配置。

这些命令执行完毕之后，用 **show line** 命令再加上一条 **show line 5** 命令加以确认一下。在输出的结果中查看调制解调器使用的 **Line 5** 情况，同时要确保调制解调器设置正确，检测到 CTS 信号存在。

例 1-58 调制解调器的线路配置以及 show line 命令的显示

```
skynet_access(config)#
skynet_access(config)#enable password cisco          ← Allows privileged access
skynet_access(config)#line 5
skynet_access(config-line)#transport input all       ← Allows terminal sessions
skynet_access(config-line)#modem inout               ← Configures the modem
skynet_access(config-line)#modem autoconfigure discovery ← for autodetection
skynet_access(config-line)#^Z
skynet_access#
```

Skynet_access#show line

Tty	Typ	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	0	0/0	-
1	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
* 2	TTY	9600/9600	-	-	-	-	-	0	1	145/437	-
* 3	TTY	9600/9600	-	-	-	-	-	0	1	109/328	-
4	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
5	TTY	115200/115200	-	inout	-	-	-	0	1	0/0	-
6	TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
7	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-

```
3 TTY 9600/9600 - - - - - 0 0 0/0 -
9 TTY 9600/9600 - - - - - 0 0 0/0 -
10 TTY 9600/9600 - - - - - 0 0 0/0 -
11 TTY 9600/9600 - - - - - 0 0 0/0 -
12 TTY 9600/9600 - - - - - 0 0 0/0 -
13 TTY 9600/9600 - - - - - 0 0 0/0 -
14 TTY 9600/9600 - - - - - 0 0 0/0 -
15 TTY 9600/9600 - - - - - 0 0 0/0 -
16 TTY 9600/9600 - - - - - 0 0 0/0 -
17 AUX 9600/9600 - - - - - 0 0 0/0 -
18 VTY - - - - - 0 0 0/0 -
19 VTY - - - - - 0 0 0/0 -
20 VTY - - - - - 0 0 0/0 -
21 VTY - - - - - 0 0 0/0 -
Tty Typ Tx/Rx A Modem Roty Acc0 AccI Uses Noise Overruns Int
22 VTY - - - - - 0 0 0/0 -

skynet_access#show line 5
Tty Typ Tx/Rx A Modem Roty Acc0 AccI Uses Noise Overruns Int
5 TTY 115200/115200- inout - - - 0 1 0/0 -

Line 5, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 115200/115200, no parity, 2 stopbits, 8 databits
Status: No Exit Banner, Modem Detected
Capabilities: Modem Callout, Modem RI is CD, Modem Discovery
Modem state: Idle
Group codes: 0
Modem hardware state: CTS noDSR DTR RTS, Modem Configured
Special Chars: Escape Hold Stop Start Disconnect Activation
               ^x none none
Timeouts: Idle EXEC Idle Session Modem Answer Session Dispatch
           00:10:00 never none not set
           Idle Session Disconnect Warning
           never
           Login-sequence User Response
           00:00:30
           Autoselect Initial Wait
           not set

Modem type is usr_sportster.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed transports are lat pad v120 mop telnet rlogin nasi. Preferred is lat.
No output characters are padded
No special data dispatching characters
skynet_access#
```

配置完成之后，可以利用超级终端或其他终端仿真软件（如 ProComm）去建立模拟连接来进行测试。

这样，访问服务器的配置基本上保持不变，可以通过反向 Telnet 以及模拟连接来访问所有的路由器。

现在就有了开始 Internet 实验学习的必要网络搭建基础，可以开始实验练习了。

第2部分

LAN 模型的建立

第2章 LAN 协议：Catalyst 以太网和 令牌环交换机的配置

第 2 章

LAN 协议：Catalyst 以太网和令牌环交 换机的配置

在网络世界中，任何技术的发展速度都不会超越局域网（LAN）技术。在过去的十年之内，LAN 已经步入了许多家庭，并对很多中小企业来说也已是必不可少。很多新办公楼都利用不同类型的铜线或光纤组建了局域网。即使是在旅行中，也能在许多酒店里面见到可以连接至因特网的局域网存在。很多新建住宅区也都有了自己的社区网络（CAN），住户可以通过它连接上互联网络服务以及访问其他相关服务。

LAN 数目增加的同时，其带宽也在呈指数级别增长。过去 10 年中所设计和应用的 LAN 协议与标准的增长速度也非常可观的。例如，2002 年 3 月，万兆以太网——802.3ae 标准即将出台，而十万兆以太网看来也是指日可待了。为了形象地说明这一点，这里采用了一个不是那么科学的“Twinkie 理论”。如果一个 Twinkie 代表 10Mbit/s 以太网的带宽，那么吉比特以太网的带宽就是一个大约 333 英尺长，100 英尺高的 Twinkie——可真是一个巨大的 Twinkie。

注释 社区网络（CAN）指为一个或多个家庭用户共享同一网络结构。

LAN 协议的瞬息万变。在 LAN 的争夺战中有过很多的胜利者和失败者。像 100VG AnyLAN 这样的标准基本上没有被真正使用过，而其他像 FDDI II 这样的协议也停止所有的开发工作。以太网目前占有最大的市场份额，有些估计认为已经超过了 90%。尽管令牌环网络并不像以太网那样功能

强大，然而在大多数的 IBM 大型机上还是使用令牌环网。因此，本文主要是集中在以太网、令牌环网及其交换技术上，另外还将讨论 Catalyst 4000/5500/6500、Catalyst 2900XL/3500 和 Catalyst 2900 系列交换机的配置。

2.1 以太网：协议发展简史

以太网的发展很有意思。1972 年由 Bob Metcalfe 在 Xerox Palo Alto Research Center (PARC) 提出。1979 年，Digital Equipment Corp.、Intel 和 Xerox 公司建立了标准 DIX V1.0 的框架；两年后，提出了该框架的 2.0 版本。1981 年，IEEE 的 802 项目决定设立 802.3 小组，后者就相当于现在的以太网。表 2-1 给出了以太网标准的演变过程描述。该标准可以在 Robert Breyer 和 Sean Riley 编写的 *Switched, Fast, and Gigabit Ethernet* (第 3 版) 里找到。

Breyer 和 Riley 称以太网协议既是一项先进性的协议，也是一项革命性的协议。先进性的创新是建立在现有的基础之上，并做出了某些形式的改变。而作为一项革命性的协议，它必须要做出某些根本性的突破，而这种突破通常并不是建立在现有基础之上的。以太网已经有了 25 年的历史，并且还将是 LAN 的主要协议。在 Robert Breyer 和 Sean Riley 的 *Switched, Fast, and Gigabit Ethernet* 第三版里，有很多关于以太网历史，百兆争夺战以及吉比特以太网标准的描述。

注释 IEEE 的命名规范是这样的：10Base T 这一名称中，10 代表以 MB/s 为单位的传输速率。Base 表示是基带传输。*T* 代表非屏蔽双绞线 (UTP)，而 *F* 代表光纤 “fiber”。以前的以太网还用数字表示电缆段长度，就像 10Base 5 和 10Base 2，随着以太网能够在同一标准上有多种电缆长度之后，这种叫法就失去意义了。

表 2-1

以太网标准的演变

以太网标准 (口头)	简称 (正式)	IEEE 对应标准	速度 (Mbit/s)	LAN 拓扑	传输长度 (m)	支持介质
粗缆以太网	10Base 5	802.3	10Mbit/s	总线	500	50 Ω 铜轴 (粗)
细缆以太网	10Base 2	802.3a	10Mbit/s	总线	185	50 Ω 铜轴 (细)
宽带以太网	10Broad 36	802.3b	10Mbit/s	总线	1800	75 Ω 铜轴
10Mbit/s 中继器	中继器	802.3c	10Mbit/s	总线	—	50 Ω 铜轴 (粗/细)
光纤中继器	FOIRL	802.3d	10Mbit/s	星型	1000	光纤
星型局域网	1Base 5	802.3e	1Mbit/s	星型	250	100 Ω 2 对线 Cat 3 UTP
多点星型局域网	1Base 5	802.3f	1Mbit/s	星型	250	100 Ω 2 对线 Cat 3 UTP
层管理		802.3h	10Mbit/s	—	—	—
双绞线以太网	10Base T	802.3i	10Mbit/s	星型	100	100 Ω 2 对线 Cat 3 UTP
光纤以太网	10Base F	802.3j	10Mbit/s	星型/总线	<2000	光纤
10Mbit/s 中继器管理		802.3k	10Mbit/s	星型	—	—

续表

以太网标准 (口头)	简称 (正式)	IEEE 对应标准	速度 (Mbit/s)	LAN 拓扑	传输长度 (m)	支持介质
10BaseT 统一规范声明 (PICS)	10Base T PICS	802.3i	10Mbit/s	星型	<2000	多模或单模光纤
第2维护		802.3m	10Mbit/s	—	—	—
第3维护		802.3n	10Mbit/s	—	—	—
MAU 管理		802.3p	—	—	—	—
管理开发规范		802.3q	—	—	—	—
10Base 5 PICS	10Base 5 PICS	802.3r	10Mbit/s	—	—	—
第4维护		802.3s	10Mbit/s	—	—	—
10Base T 120 ohm 电缆		802.3t	10Mbit/s	—	100	120 Ω 2 对线 Cat3 UTP
快速以太网	100Base TX	802.3u	100Mbit/s	星型	100	100 Ω 2 对线 Cat5 UTP
3 类线快速以太网	100Base T4	802.3u	100Mbit/s	星型	100	100 Ω 4 对线 Cat3 UTP
光纤快速以太网	100Base TX	802.3u	100Mbit/s	星型	<2000	光纤
10Base T 150ohm 电缆		802.3v	10Mbit/s	—	100	150 Ω 2 对线 Cat3 UTP
增强型 MAC 或二进制 制对数算法	BLAM	802.3w	—	—	—	—
全双工流控制	FDX	802.3x	10Mbit/s	—	—	—
3 类 4 线制快速以太网	100Base T2	802.3y	100Mbit/s	星型		100 Ω 2 对线 Cat3 UTP
吉比特以太网 (短波)	1000Base SX	802.3z	1000Mbit/s	星型	300	多模光纤
吉比特以太网 (长波)	1000Base LX	802.3z	1000Mbit/s	星型	550	多模光纤
吉比特以太网	1000Base CX	802.3z	1000Mbit/s	星型	3000 25	单模光纤 150 Ω 铜线
第5维护	100Base T	802.3aa	100Mbit/s	—	—	—
5 类线吉比特以太网	1000Base T	802.3ab	1000Mbit/s	星型	100	Cat5 UTP Cat5e
虚拟以太网	VLAN	802.3ac	—	—	—	—
中继封装	连接集合	802.3ad	—	—	—	—
万兆以太网*	10000Base	802.3ae	10 000Mbit/s	星型	100 300 2k 40k	多模光纤 单模光纤
虚拟以太网标记	VLAN 标记	802.1Q	—	—	—	—
安全数据交换	安全虚拟网	802.10	—	—	—	—
流量分发	优先级交换	802.1p	—	—	—	—
MAC 桥接的生成树	MAC 桥接	802.1D				

*802.ae 还没有最后定稿，但草案已宣告了基本流程。

2.2 以太网技术概览

以太网现在非常普及，有很多关于该协议的著作和白皮书。因此，本书中假设读者已经对以太网技术有了一定了解，就不花太多的篇幅去讲述以太网数据帧、集线器以及电缆的问题。读者应该对不同的以太网数据帧类型，DIX 的版本 II 和 802.2 数据帧以及以太网中所用的各种媒介类型都有了一定的认识。这一章是重点介绍生成树、快速以太网、吉比特以太网以及以太网与令牌环网的交换技术。

2.2.1 以太网的工作原理

以太网工作在 OSI 第 2 层即数据链路层上。数据链路层实际上分成两个子层：MAC 层和逻辑链路层（LLC）。LLC 层（这里是指 802.2），就是硬件 MAC 地址与第 3 层协议之间一个标准接口。

MAC 层的功能如下：

- 产生数据帧的物理源地址和目的地址，这个 48 位的地址在所有厂商产品中是惟一的，前 3 个字节由 IEEE 分配，后 3 个字节由制造商惟一指定。
- 确保可靠传输。
- 同步数据传输。
- 检错功能。
- 数据流控制。

表 2-2 列出了 10Mbit/s、100Mbit/s 和吉比特以太网的常用物理特性。

表 2-2 常用的以太网规范

规 范	10Mbit/s	100Mbit/s	1000Mbit/s
最小帧	512bit/64 byte	512 bit/64 byte	4096 bit/512 byte
比特时间（ μ s）	0.1 μ s	0.01 μ s	0.001 μ s
最大来回延迟（ μ s）	51.2 μ s	5.12 μ s	4.096 μ s
最大网络直径（无中继器时）（m）	45710 000s	457 000s	3661 000s
冲突域中中继器最大数量	5 个左右	1 个 1 类中继器或 2 个 2 类中继器	1

1. Ethernet CSMA/CD

以太网技术通常称为带冲突检测的载波侦听多路访问（CSMA/CD）。以太网是按照下面的方式传输数据帧的：

1 Carrier sense——载波侦听：这也称为“说前先听”。要传送以数据帧的以太网工作站在送出数据之前先监听传输介质状态以确保传输介质可用。

2 Talk if quiet——空闲传输：如果一段时间（帧间间隙（IFG））内通信信道是空闲、可用的，工作站就可以开始数据的传送。如果信道忙，工作站会监听信道状态，直到信道空闲时间大于 IFG 的时候，数据传输开始进行。

3 Collision——冲突：指电缆或传输介质中检测到的冲突。冲突是在两台工作站同时传输数据时产生的。如果发生了冲突，两个数据帧都会被破坏。

4 Collision detection——冲突检测：如果一台工作站在传输数据时检测到有冲突发生，该传输将立即停止。拥塞信号也会送到传输介质上去消除所有的未传输数据，以防止产生数据碎片。

5 Backoff——退避：冲突之后，工作站会等待一段时间（称为退避时间），该时间是由一种退避算法产生的随机数。这样可以防止工作在冲突以后退后相同的时间传送数据。退避时间之后，工作站就会试着重新发送该帧。如果又发生了另外一个冲突，工作站还会再接着发送该数据 16 次。如果 16 次发送都不成功，该数据帧就会被丢弃。

2. 半双工与全双工以太网

以太网是建立在可以随时接收或发送信号的同轴电缆基础之上的。这就是为什么以太网要用 CSMA/CD 技术的原因。随着交换机的出现，以太网可以运行在 UTP 和光纤上，使得全双工以太网成为可能。全双工以太网允许一台工作站同时接收和发送数据，以太网的数据帧可以同时通过 UTP 的两对双绞线或者是光纤进行接收和发送。全双工以太网网络从本质上来说，是没有 CSMA/CD 功能的以太网。全双工模式基本上实现了以太网带宽加倍！为了应用全双工模式，工作站和交换机必须支持并且设置成全双工工作模式。连接多个工作站的集线器不能工作在全双工模式下。

注释 设置成不匹配双工模式的工作站会在其端口上产生大量的冲突，这些冲突大都可能被标记为“延时冲突”。一定要确认交换机和终端工作站的端口工作在相同的双工模式下。

3. 快速以太网

1995 年 5 月，IEEE 颁布了快速以太网标准 802.3u。几年以后，经过与 FDDI、100VG AnyLAN 以及 ATM 的争夺之后，该标准已经成为了 LAN 的主导标准。随着网卡（NIC）价格的下跌，终端成本的下降，快速以太网已经在市场份额上超过了 FDDI、100VG AnyLAN 以及 ATM，主要原因如下：

- 快速以太网可以简单而低成本地升级现有的 10Mbit/s 以太网。最初，快速以太网只能在光纤和 5 类 UTP 上运行，而如表 2-1 所示的那样，现在快速以太网已经能够运行在几乎所有的传输介质上。
- 快速以太网不需要成本很高的光纤连接方式，也不需要进行很复杂的配置。
- 快速以太网可以在提供很高的带宽时应用质量服务（QoS），而 QoS 可以由网络上层协议或网络设计来提供。
- 基本上，快速以太网已经可以从 LAN 以即插即用的方式直接升级而成。通常数据中心的百兆网络都是直接从十兆升级而成。

快速以太网的重要特性和规格如下：

- 100Mbit/s 以太网仍然采用 10Mbit/s 以太网的 MAC，只是速度提高了 10 倍。这完全是为了向后兼容 10Mbit/s 以太网网络。

- 100Base T 包含一个 MII 接口的规范。· MII 接口是 AUI 适配器的 100Mbit/s 版。
- 快速以太网支持全双工和半双工功能。
- 快速以太网可以工作在多种物理传输介质上：Cat 5 电缆、 Cat 3 电缆、 光纤等，如表 2-1 所列。

4. 吉比特以太网

快速以太网标准颁布不久，IEEE 就开始规划 802.3z，或叫做吉比特以太网的标准。短短的 3 年之后，1998 年 6 月，802.3z 标准就正式颁布了。从很大程度上说，吉比特以太网就是快速以太网乘以 10。这也就是为什么万兆以太网刚刚诞生，十万兆以太网就即将出现的原因。

吉比特以太网的一些特性和规格如下：

- 吉比特以太网使用 802.3 帧格式，与 10Mbit/s，100Mbit/s 以太网完全一样。
- 吉比特以太网包含了千兆 MII (GMII) 的规范。和 10Mbit/s、100Mbit/s 的 MII 不一样，GMII 是一份电气规范而不包括物理接口规定。Cisco 的物理千兆接口叫做 GBIC。GBIC 的类型决定了千兆网接口的物理连接。目前的 GBIC 模块类型包括多模光纤 (MMF)、单模光纤 (SMF) 和 UTP GBIC 以及 Cisco 公司私有的叫做千兆堆栈 (Gigastack) 的 GBIC。

吉比特以太网这些年成为了最通用网络协议的原因之一就是其 GMII 的概念。除了严格的 1000Base TX 交换机之外，大多数吉比特以太网交换机都带有一个 GBIC 端口。根据网络需要，可以将任何类型的 GBIC 模块连到该端口中。只需要插入 GBIC 模块，网络就可以从原来的百兆基于铜线的网络转变成千兆的基于光纤的网络！随后的内容更多的是关于常见的 GBIC、千兆标准以及距离限制的。

5. 1000Base SX 吉比特以太网

1000Base SX GBIC 使用的激光波长为 850 nm。根据线缆类型的不同，SX GBIC 的工作距离范围可以从 220m 到 550m，如表 2-3 所示。850 nm 的波长对人眼来说是可见的。

表 2-3 1000Base SX 的线缆限制

标 准	线缆类型	最远距离
1000Base SX	62.5 μ m 多模光纤	275m
1000Base SX	50 μ m 多模光纤	550m

6. 1000 Base LX 吉比特以太网

LX GBIC 使用的激光波长为 1300 nm。根据线缆类型的不同，LX GBIC 的工作距离可以从 550m 到 5000km，如表 2-4 所示。Cisco 还支持 LH 和 LX GBIC，将 IEEE 的 1000Base LX 的最远距离提高到了 5km。

表 2-4 1000Base LX 的线缆传输限制

标 准	电缆类型	最远距离
1000Base LX	62.5 μ m 多模光纤	550m
1000Base LX	50 μ m 多模光纤	550m

续表

标 准	电缆类型	最远距离
1000Base LX	9/10 μm 单模光纤	5km
1000Base LH	62.5 μm 多模光纤	550m
1000Base LH	50 μm 多模光纤	550m
1000Base LH	9/10 μm 单模光纤	10km
1000Base ZX	9/10 μm 单模光纤	70km
1000Base ZX	9/10 μm 增强光纤	100km

7. 1000Base CX 吉比特以太网

CX 标准的以太网是运行在较短距离铜线上的吉比特以太网。

1000Base CX 用的是 $150\ \Omega$ 的平衡式屏蔽铜线电缆。CX 标准的工作距离限制在 25m 之内。

8. 1000Base T 吉比特以太网

这个由 IEEE 制定的工作在 5 类 UTP 上的吉比特以太网标准叫 802.3ab。标准定义最远工作距离为 100m，铜线至少是 5 类，使用 4 对的电线，尾端是 RJ 45 的插座。图 2-1 所示即一个 GBIC。

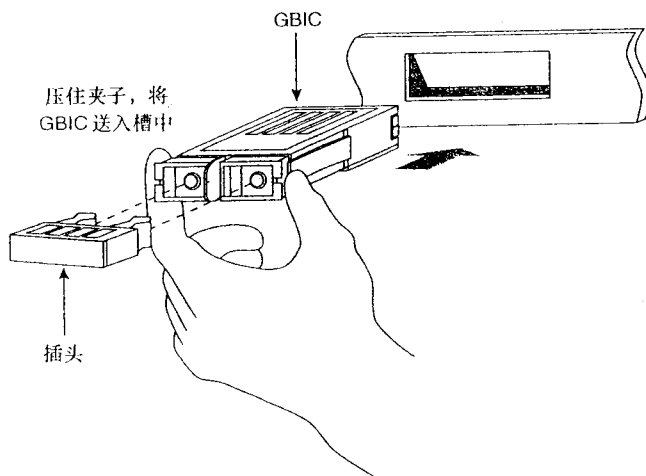


图 2-1 千兆光纤接口卡 GBIC 的安装

注释 Cisco Gigastack GBIC 是 Cisco 拥有的用于吉比特以太网交换机上连端口间互连的 GBIC。

9. 以太网的自动协商机制 (Autonegotiation)

为了简化以太网设备的配置，802.3u 委员会定义了快速连接脉冲 (FLP)。FLP 能够发送一系列的脉冲到网络以协商连接工作的双工方式以及速率。工作站以及集线器/交换机会协商最高优先级的方式并按该方式设置。表 2.5 列出了 FLP 使用的优先级情况。两个设备都需要

支持自动协商机制，这样双方才能协商。

表 2-5 以太网自动协商优先级

优先级	速率和双工设置
1*	100Base T2 全双工
2	100Base T2 半双工
3	100Base TX 全双工
4	100Base T4 全双工
5	100Base TX 全双工
6	10Base T 全双工
7	10Base T 全双工

*优先级 1 可能会被 1000Base T 全双工的工作方式取代，其他的优先级依次递减。

像路由器、服务器这样的基本设备，最好设置成全双工工作模式。多数百兆和大部分的网卡都支持全双工方式。全双工方式使得以太网的带宽基本上扩大了 1 倍，充分利用这一点是节省网络升级成本的最佳方法。

注释 双工模式是网卡 (NIC) 的一种硬件功能。软件升级可能会造成全双工模式无法使用。如果要用这种通信方式，工作站和交换机端口都必须支持全双工。

2.3 802.1d 生成树协议 (STP)

以太网从简单的共享电缆发展成带有多个桥接和集线器的网络时，需要一种环路检测和防止环路的协议。Radia Perlman 博士提出的 802.1d 协议提供了这样的环路避免功能。当大多数的网络从桥接方式过渡到路由方式时，生成树协议起到了重要的作用，现在它的重要性却渐渐被淡忘了。因此，生成树协议可能是现代互联网络中使用得最广但却是最少为人知的技术。随着以太网交换技术的巨大成功，生成树又成为一项需要读者去了解、掌握的重要协议。在随后的内容里会讨论交换型以太网网络中生成树协议为什么重要。

2.3.1 生成树 (STP) 工作原理

生成树的目的是选举一个根桥，并针对网络中所有的桥，建立无环路的指定根桥的路径。当生成树时，网络中每个桥接接口都处于其中一种状态：*转发或阻塞*。如果端口位于指定根桥的最佳路径上，该状态就是转发状态，因此也就是最短路径。所有其他端口都处在阻塞状态。STP 通过发送称为桥接协议数据单元 (BPDU) 的特殊信息来完成这一功能。BPDU 有以下两种形式：

- 用于初始 STP 配置的 BPDU。
- 用于拓扑结构改变时的拓扑变化通告 (TCN) BPDU。

BPDU 使用保留位的多播地址来通知所有的桥，发送时通过所有的 LAN 桥接端口传送，LAN 中所有的桥会收到这些信息。BPDU 不可透过路由器传送。

BPDU 包括如下相关信息：

- **Root ID**——被选为根桥的 ID，初始化时，桥认为自己就是根桥。
- **Transmitting bridge ID and port ID**——发送 BPDU 的桥 ID 和源端口 ID。
- **Cost to root**——发送 BPDU 的桥连接到根桥的最短路径。初始化时，由于桥接认为自己是根桥，它所发送 BPDU 中到根桥的开销为 0。

桥 ID (BID) 是一个 8 字节的字段，是由一个 6 字节的 MAC 和一个 2 字节的桥优先级构成的。BID 的 MAC 地址来源各不相同，与桥本身所用的硬件电路有关。路由器用的是一个物理地址，而交换机用的则是从背板或者是管理引擎产生的地址。图 2-2 就是 BID 的例子。优先级的值从 0 到 65,535，其默认值是 32 768。

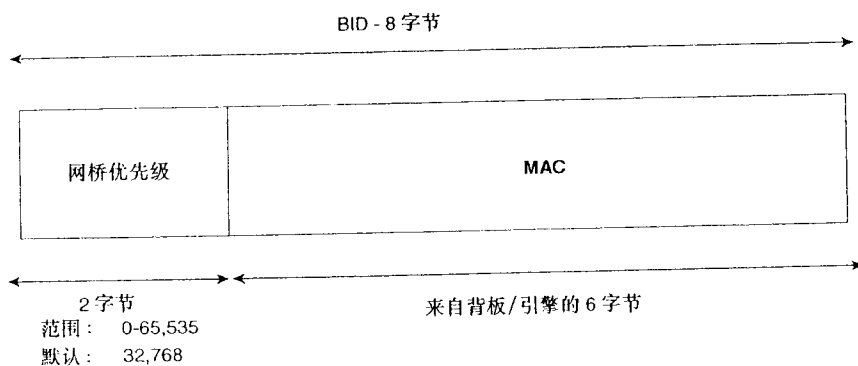


图 2-2 BID 示例

路径开销 (path cost) 用来确定到根桥的最短路径。最近 IEEE 更新了其路径开销的定义，以便将千兆网和将来更高速的网络包括进去。路径开销值越低，路径就越好。表 2-6 列出了 LAN 连接里的 STP 路径开销的值。

表 2-6 LAN 连接里的 STP 路径开销值

带 宽	STP 开销
4 Mbit/s	250
10 Mbit/s	100
16 Mbit/s	62
45 Mbit/s	39
100 Mbit/s	19
155 Mbit/s	14
622 Mbit/s	6
1 Gbit/s*	4
10Gbit/s	2

*IEEE 标准更新前，STP 的最小值可以到达 1。STP 为 1 用于高于或等于 1G 的所有链路。

STP 在工作时有 5 种状态。当 STP 收敛时，它是处于其中任一种状态，即转发状态和阻塞状态。表 2-7 列出了 STP 的状态。

表 2-7 不同的 STP 状态

STP 状态	STP 活动	是否传递用户数据
失效 (Disabled)	端口未激活，不参与 STP 的任何活动	否
中断 (Broken)	一端 802.1q 配置错误，或者默认/本地 VLAN 不匹配	否
监听 (Listening)	端口收发 BPDU	否
学习 (Learning)	建立无环转发表	否
转发 (Forwarding)	发送接收用户数据	是
阻塞 (Blocking)	不从此端口发送用户数据	否
快速端口 (Portfast) *	监听/学习状态	是

*portfast 是 Cisco 特有的用法，允许用户数据业务在 STP 收敛期间发送。

图 2-3 演示了端口如何从一种状态转换成另一种状态。

现在具体分析这些状态类型。

1. 失效状态

该状态是桥处理 BPDU 时存在问题、中继配置不正确或者是端口被关闭了的情况下出现。

2. 监听状态

在桥初始化端口时或一定时间之内没有接收到 BPDU 时，STP 进入监听状态，这时，端口实际上是阻塞的，链路上没有用户数据传送。STP 在收敛过程中有如下 3 个步骤：

(1) 选择一个根桥——初始化时，桥会在所有接口上发送 BPDU。BID 值最小的桥会被选为根桥。BID 包括优先级和 MAC 地址。如果优先级都相同的话，则 MAC 地址最小者被选为根桥。根桥所有的端口都处在转发状态。

(2) 非根桥接选择一个根端口——在选定一个根桥之后，STP 还会在每个非根桥上选择一个根端口，即该桥连接到根桥的最佳路径。选定根端口之后，就进入转发状态。要确定哪一个端口作为根端口，STP 的抉择过程如下：

- 最小的根 BID；
- 到根桥的最小代价以及到根桥代价累积起来和；
- 最低发送者的 BID；
- 最低的端口 ID。

当桥收到一个 BPDU 时，会将其存储在端口的桥接表里。端口上收到新的 BPDU 时，会与已有的 BPDU 进行比较。它按照上面所列的 4 个步骤，最佳的或者代价值最小的 BPDU 保留下来，而其他都删除。对根端口选择影响最大的值是称为到达根桥的开销 (the cost to the root bridge) 的值，这是到根桥所有的链路代价的累积和。

(3) 为每一网络分段选择指定的端口/指定网桥——对网络的每一分段，STP 都会为其选择一个用于发送和接收从该段到根桥的所有信息的端口。根端口可以看做是将数据发送到根桥的端口，而指定端口则是数据传离根桥的端口。该规则对于大多数的共享传输介质桥接或路

由器来说都是适合的。在背对背交换机中继连接中那些指定的端口并不符合该规则。

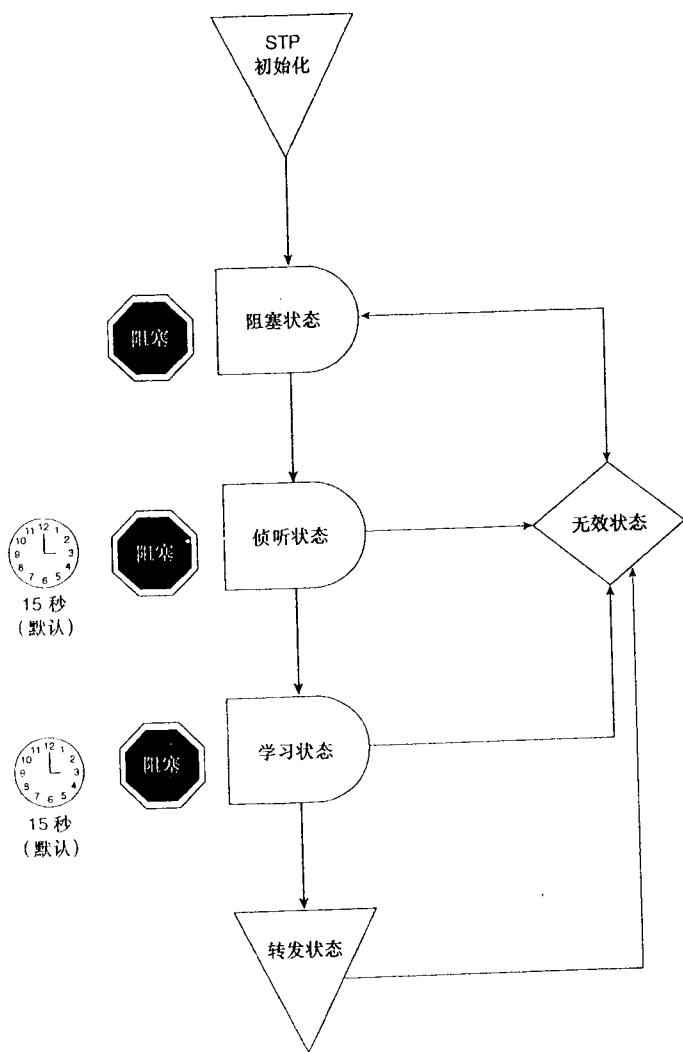


图 2-3 STP 状态转换

(4) 所有剩余端口都成为未指定的端口，进入阻塞状态。

3. 学习状态

根端口和其他的指定端口在经历了 15 秒（默认设置）的转发延时后，就将进入学习状态，这一状态其实就是又一个 15 秒用于桥建立其桥转发表的等待时间。

4. 转发状态和阻塞状态

当桥进入到这个阶段后，那些不是根桥或指定桥接端口被称做未指定端口。所有指定的端口都进入转发状态，而所有未指定的端口则进入阻塞状态。在阻塞状态中，桥不会发送配置 BPDU，但是仍然会监听这些 BPDU。阻塞的端口也不会转发任何用户数据。

图 2-4 展示了一个基本的 STP 设置情况，并标出了端口类型。

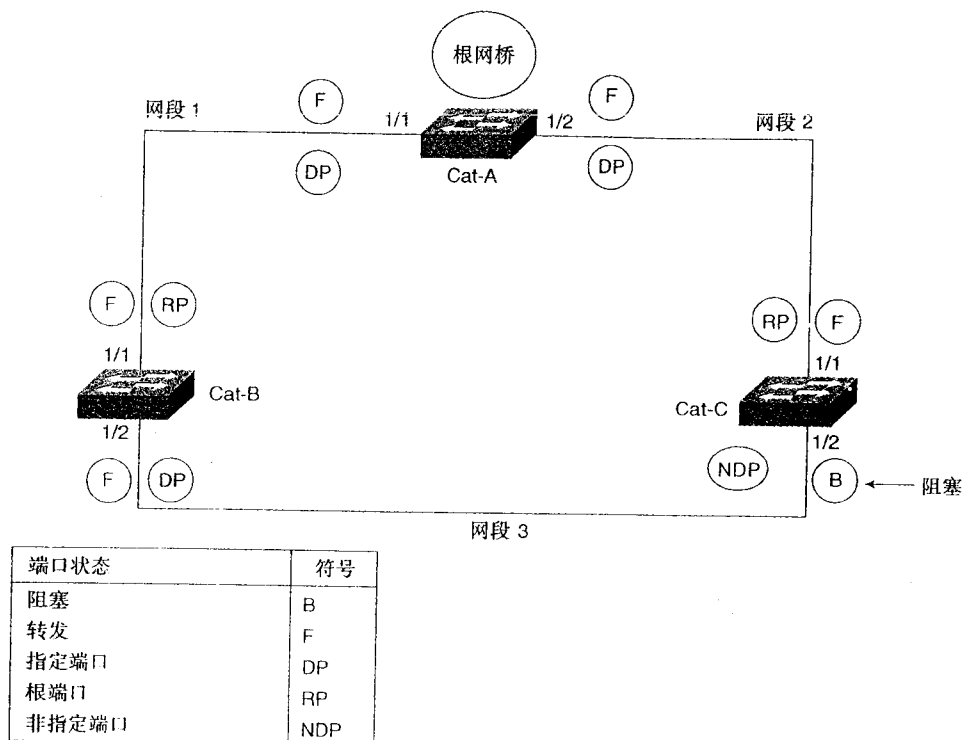


图 2-4 STP 端口与状态

2.3.2 STP 定时器

STP 有 3 个基本的定时器，用于管理和调节 BPDU：hello 定时器，转发延时定时器和最大失效定时器。它们的功能如下：

- **Hello 定时器 (Hello timer)** ——该定时器的默认值是 2 秒，是根桥发送配置 BPDU 的间隔时间。
- **转发延时定时器 (Forward delay timer)** ——该定时器默认值为 15 秒，是交换机建立转发表的等待时间。监听和学习状态都会持续这么长的时间。
- **最大失效定时器 (Max age timer)** ——该定时器的值是 BPDU 被刷新之前的存储时间。

如果接口收到新的 BPDU 之前该定时器定时溢出，接口就会转入监听状态。一般来说，这种问题都是由连接故障引起的。其默认值是 20 秒。

STP 每隔一个 Hello 时间发送一个 BPDU，并且引入 keepalive 机制。Hello 包的发送可以避免最大失效定时器的溢出。如果最大失效定时器溢出，通常表明有连接错误发生。这时，桥接会进入监听状态。STP 要从连接错误中恢复过来，一般需要 50 秒的时间。其中 BPDU 最长时失效时间 20 秒；监听状态持续 15 秒；学习状态持续 15 秒。

注释 除了 IEEE 制定的 802.1d STP，另外两种得到应用的 STP 是 DEC 和 IBM STP。所

有的 STP 类型的工作方式都相似，而 Cisco 路由器支持所有的 STP 类型。

现在需要思考这样一个问题：一个使用第 2 层网络协议，含 2 秒的 Hello 间隔，50 秒收敛时间的协议在现代网络中扮演什么角色呢？由于交换机是属于第 2 层的设备，所有的 VLAN 都使用生成树在交换机之间建立无环路径。Cisco 采用的是每个虚拟局域网一个生成树（PVST）的技术。采用 PVST 时，每一个虚拟局域网（VLAN）都有自己的生成树。以一个含有 50 个 VLAN 的网络模型为例，每条中继和每个交换机上都有 50 个生成树。很快，读者就会明白理解和掌握该协议的必要性。因为其重要性，掌握生成树协议就成为本节“Catalyst 以太网交换机的设置”的重点。

2.4 以太网交换技术

上个世纪 90 年代早期，Kalpana, Grand Junction 和 Bay Networks 等网络公司纷纷推出了各自的以太网交换机。Bay Networks 28115 是最早引入 10/100 兆自适应端口以及虚拟局域网（VLAN）等技术的交换机。更为重要的是，所有的交换机都摒弃了传统交换机的中继规则而转为增加带宽。直到现在，很多人都大胆预测说 ATM 将成为 LAN 中惟一的高速协议。如果没有以太网交换机，那或许是真的。相比传统的传输介质——共享局域网，以太网交换机有如下的主要优点：

- 通过将冲突域限制到单个端口上而大大提高了带宽利用率。
- 中继器规则被限制到单个端口。
- VLAN 能力。广播域能够预置而且不再受地域的限制。
- 增强安全性。
- 全双工能力。

交换机的工作原理与多端口的桥相似。在创建 VLAN 时，也会创建虚拟桥接以连接 VLAN 中的端口。网络广播、单播和多播的通信量都会发送到 VLAN 中的每个成员。Catalyst 5500 系列交换机会通过记录接口收到数据帧的源 MAC 地址来建立一份地址映射表。当收到的数据帧地址不在地址表映射中时，交换机会在接收到该数据帧时将其送到同一 VLAN 中所有的端口和中继中，只有该数据帧到达的接口除外。如果收到了该数据帧的回应信息，路由器会在地址映射表中记录这一新的地址。交换机会把随后的数据帧送到对应端口，而不再送到所有的端口。只有通过路由器或者是有路由功能的第 3 层交换机才能在 VLAN 间进行数据传送。

交换机转发数据的方式有 3 种：

- 存储转发（Store and forward）——端口将数据帧完整地读入存储器内，然后判定数据帧是否应该转发。数据帧校验无误后才能转发。存储转发方式减少了 LAN 中的数据错误，但却增加了转发前缓冲数据和校验数据的延时。在现代基于 ASIC 的交换机中，ASIC 速度已经快到可忽略存储转发的延时的程度。
- 直接转发（Cut through）——在这种方式中，端口收到一个数据帧开头的几个字节后就对其进行简单分析，查看它的数据头以确定该帧目的地址，然后立即将其转发出去。在转发前数据帧没有经过校验，因此这种方式可能会在网络中广播错误数据帧。

自适应的直接转发（Adaptive cut through）——这种方式将上面两种方式结合起来，

端口默认按直接转发方式工作，只有在检测到数据错误的值超过用户定义的数据帧错误阈值时，才使用存储转发方式进行转发。

2.4.1 广播域与冲突域

交换网络中的两个关键概念是：**广播域**和**冲突域**。广播域是指网络中可以将网络广播从网络的某一部分转发到其他部分中的网络区域。广播域的一个实例就是 IP 或 IPX 子网。冲突域则是一个设备物理特性的功能。同一冲突域里的设备是在同一条电缆或者是由同一台集线器/中继器连接起来的。表 2-8 说明了网络设备是如何区分冲突与广播域的。

图 2-5 则描述了不同设备上冲突域与广播域。

表 2-8 网络设备是如何分割冲突与广播域的

硬件类型	冲突域	广播域
中继器/hub	所有端口处于一个冲突域	所有端口处于一个广播域
桥	每个端口是一个单独的冲突域	所有端口处于一个广播域
路由器	每个端口是一个单独的冲突域	每个端口是一个单独的广播域*
交换机	每个端口是一个单独的冲突域	每个端口可以配置在同一个或不同个广播域

*假设桥接功能关闭

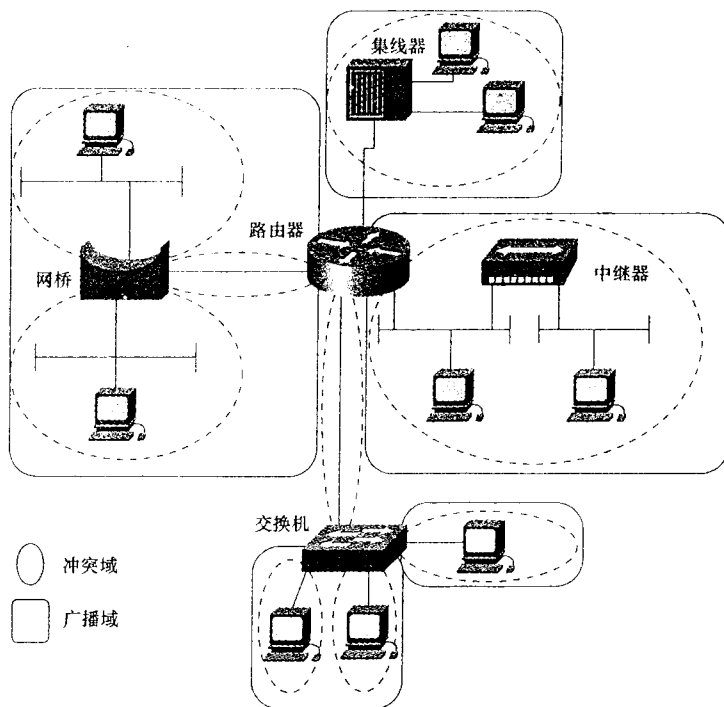


图 2-5 冲突和广播域

2.4.2 虚拟局域网 (VLAN)

VLAN 有许多定义。简单而言，VLAN 就是不受网络地域限制的广播域。在配置一台以太网交换机时，其每一个端口都要属于一个 VLAN。默认情况下的 VLAN 总是 VLAN 1。当交换机出厂时，为了方便即插即用，每个端口都设置为 VLAN 1，因此交换机的每个端口都属于一个广播域。这样可以很容易地从一个共享型集线器以太网升级为一个基本的交换式网络。一般都将 VLAN 看成是一个简单的广播域。大多数的 VLAN 最后都演变成了 IP/IPX 子网或桥接域。广播域的基本设计规则对 VLAN 同样适用：

- 每个 VLAN 都单独占用一个子网。
- 不要将不同的 VLAN 用桥接连接。
- 在 VLAN 之间进行路由需要使用路由器或 3 层交换机。
- 每个 VLAN 中都需要运行 STP 以防止环路。当然该功能可以屏蔽，但是建议最好将其保留。

再查看一些基本的交换型网络，重点是它们相互的区别。图 2-6 给出了一个基本的 VLAN 示意图。在交换机上分别配置了 VLAN1 和 VLAN2 两个 VLAN。每一个 VLAN 又都配置了一个独立的 IP 子网。如果将数据从 VLAN1 传送到 VLAN2，就需要一台路由器。这里，这台路由器在每个 VLAN 上都有一个接口。VLAN 间的数据通信首先需要一个路由器。该配置的缺陷是每个需要路由的 VLAN 都需要一个单独的接口，这样就严重限制了网络的扩展性。

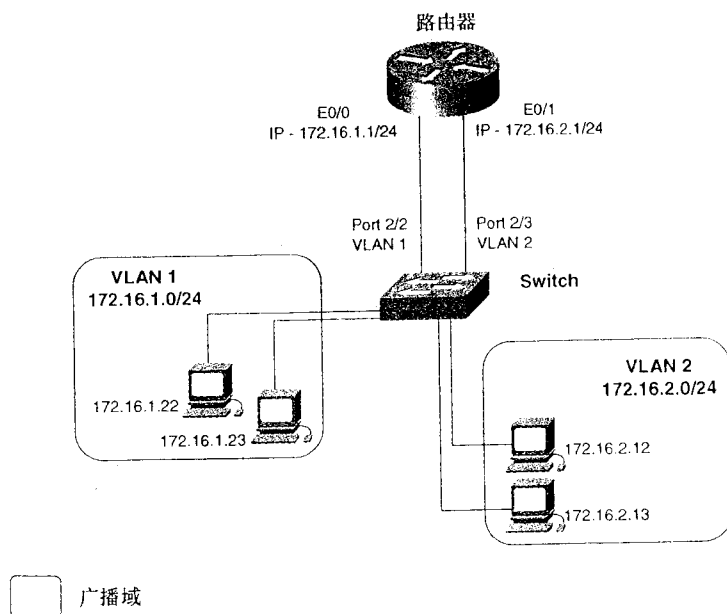


图 2-6 每个接口上连接的一个 VLAN 的路由

图 2-7 给出了另外一个基本的 VLAN 配置示意图。这里的交换机也在上面配置有两个 VLAN。路由器有一个接口（E0/0）连接到交换机，并运行 802.1q VLAN 中继协议（trunking protocol）。从一

个 VLAN 到另一个 VLAN 的通信数据必须从中继传到路由器，然后从中继再传回来。使用单一的中继在 VLAN 之间进行路由是最经济的方式。这种配置方式通常被称为“单臂路由（router on a stick）”。

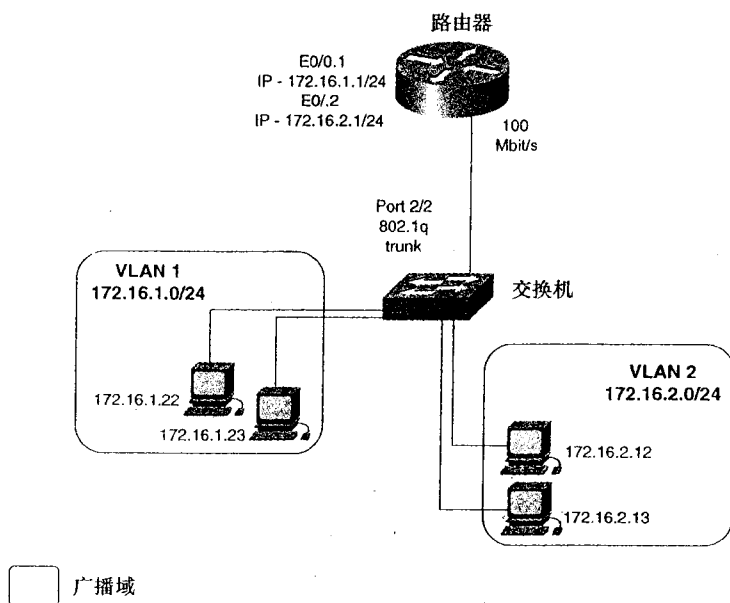


图 2-7 单臂路由（Router on a Stick）

下一步是将路由功能模块从一台独立的路由器内置到交换机中，这称作第 3 层交换。这一移动仅仅是逻辑上的移动，因为通过同一路由器接口进出的通信数据量还是加倍。首先，通过使用安装在 Catalyst 5500 系列交换机里的路由交换处理器（RSM）来完成的。现在许多交换机都提供了这一功能。图 2-8 给出了一台第 3 层交换机的示意图。

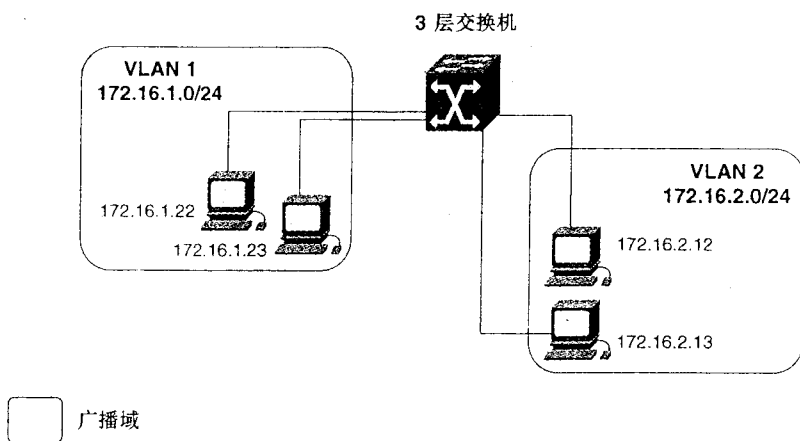


图 2-8 第 3 层交换

2.4.3 VTP 和中继协议

VLAN 有一个很强大的功能就是能够跨越地域限制。交换机与交换机之间是通过 VLAN 中继协议 (VTP) 的通信来交换 VLAN 信息的。VTP 主要是用来在交换机与交换机之间传输全局 VLAN 信息的通信协议。VLAN 管理域 (或称 VTP 域) 是指在同一管理下的一台或几台互连的交换机的集合。任何时候如果想使一个交换机上的 VLAN 获取其他交换机上的 VLAN 里的信息 (也就是这两个广播域间通信), 就需要配置一个 VTP 域以及一个中继。VTP 也能跟踪一个 VTP 域中所有的 VLAN, 并将此信息在客户/服务器模式下从一台交换机传送到另一台交换机。VTP 的目的是使网络管理更便捷并且在 VTP 域中提供一个同步的 VLAN 数据库。

VTP 的信息是通过所有的中继链路 (包括 ISL、802.1q、IEEE802.10 或 ATM LANE 进行) 传输的。VTP 的数据帧以 SNAP (LLC 代码为 AAAA) 的格式传输, 目的 MAC 地址是 0100.0ccc.ccc, SNAP 数据类型为 2003。要想使 VTP 信息成功的传送, 必须遵守以下原则:

- VTP 只接收具有相同域的信息。如果该 VTP 域设置了认证功能, 那么同时要求域内通过认证。VTP 域名区分大小写。
- VTP 只会接收版本相同的信息, 版本 I 或版本 II。该版本信息设置可更改, 第 2 版模式在连接的两端必须同时打开或禁止。交换机的 VTP 可以设置为版本 2 或者是版本 1。建议令牌环交换机最好设置成版本 2。
- Catalyst 交换机必须处在邻近位置, 在交换机之间必须配置中继连接, 对以太网网络而言, 中继协议是 dot1q (802.1q) 或者 ISL, 而 ATM 采用的是 LANE, FDDI 采用的则是 IEEE 802.10。
- 只有 VTP 客户端的 VTP 信息的更改号小于 VTP 服务器的更改号时, VTP 服务器才需要与 VTP 客户端同步。如果 VTP 客户端的更改号等于或大于 VTP 服务器的更改号, 客户端 VLAN 的数据库不进行同步。

当中继建立之后, VTP 就会周期性地发送数据到每一个中继端口中。VTP 数据包括如下内容:

- VLAN ID (ISL 和 802.1q)。
- ATMLANE 仿真的 LAN 名称。
- 802.10 的 SAID 值。
- VTP 域名以及配置更改号。更改号最高的服务器会成为主服务器, 每一次改变 VLAN 配置之后, 更改号都会递增一次。
- 每个 VLAN 的配置信息、VLAN ID、VLAN 名和 MTU 大小。
- 以太网的帧格式。

VTP 有两个版本, 即版本 I 和版本 II。VTP 域内的所有交换机都必须属于同一版本。该规则对那些工作在透明模式下的交换机来说不适用。版本 II 针对令牌环提供了如下重要的支持:

- 令牌环支持 (Token Ring support) ——VTP V2 支持令牌环 LAN 交换和 VLAN (令

牌环网桥中继功能(T-BRF)。在下面的内容里会有更多这方面的讨论。

- 未知类型（Unrecognized type）——这里包括了对长度值（TLV）的支持。交换机工作在 VTP 模式时，未知类型长度就存在 NVRAM 中。
- 独立版本的透明模式（Version dependent transparent mode）——在透明模式下，VTP 会将与 VTP 域和版本不符的 VTP 信息转发到其他的交换机。
- 一致性检查（Consistency checks）——一致性检查是检查 VLAN 名称，而且只检查交换机新收到的相关信息。

VTP 的 3 种工作模式如下：

- VTP 服务器模式（server mode）——在 VTP 服务器模式下，可以创建、修改和删除 VLAN 信息。VLAN 信息会自动送到同一个 VTP 域中邻近的 VTP 服务器和客户端。从 VTP 服务器清除一个 VLAN 时一定要注意，因为 VLAN 会在 VTP 域内所有的服务器和客户端上被删除。如果有两个设备设置成服务器模式，VTP 配置更改号较高的交换机/服务器将会成为主服务器。VLAN 的信息将会被存在交换机的 NVRAM 中。
- VTP 客户模式（client mode）——在 VTP 客户端模式中，不能够创建、修改和删除 VLAN。只有 VLAN 名称和 VTP 模式以及 VTP 修剪可修改。客户端所有的 VLAN 信息都受 VTP 服务器控制。客户端仍然需要分配端口给 VLAN，但是 VLAN 在 VTP 服务器将信息送到客户端之前，不会在交换机上被激活。此外，在收到服务器的 VLAN 信息后，Catalyst 2900XL/2500G 系列交换机将这些信息存在本地交换机上的 NVRAM 里。Catalyst 4000/5500/6500 系列交换机在配置成 VTP 客户时不会保存 VLAN 数据库。
- VTP 透明模式（transparent mode）——在 VTP 透明模式下，交换机本地产生的 VTP 信息不会被传送出去，但可转发从其他交换机接收到的信息。在透明模式下的交换机上可以创建、修改和删除 VLAN。VLAN 信息也存于 NVRAM 中。表 2-9 列出了各种工作模式和相关操作。

表 2-9 不同的 VTP 工作模式

VTP 模式	源 VTP 信息	产生本地 VTP 信息	侦听 VTP 信息	建立、修改和删除 VLAN	本地存储 VLAN 数据库
服务器模式	是	是	是	是	是
客户模式	是		是	否	是/否*
透明模式	否**	否	是**	是	是

* Catalyst 4000/5500/6500 系列交换机在 VTP 客户端交换机上不存储 VLAN 数据库。Catalyst 2900XL/3500G/系列交换机在初始化时存储并拥有 VLAN 数据库

** 透明模式下，交换机不参与 VTP。既不同步 VTP 数据库。但是 VTP 信息仍然可在中继端口上接收和发送。但本地 VLAN 信息不在中继上发送。

表 2-10 列出了 Catalyst 交换机上默认的 VTP 模式。

表 2-10 默认的 VTP 设置

VTP 特点	默认设置
VTP 域名	空

VTP 特点	默认设置
VTP 模式	服务器模式
VTP 版本 2 更新	关闭
VTP 安全/密码	关闭

VTP 要求用中继来传输 VTP 信息。中继是指以太网交换机端口与其他网络设备（如路由器或另一台交换机）之间点到点的连接。中继的功能是在一个连接上传输多个 VLAN 的信息以及将 VLAN 在互联网上加以扩展。如果没有 VTP 和中继，IP 子网就不能分散在不同的交换机上。VTP 中继可以有效地将两个广播域连接在一起。图 2-9 里就是用 802.1q 中继将 VLAN 4 和 VLAN 2 连在一起。

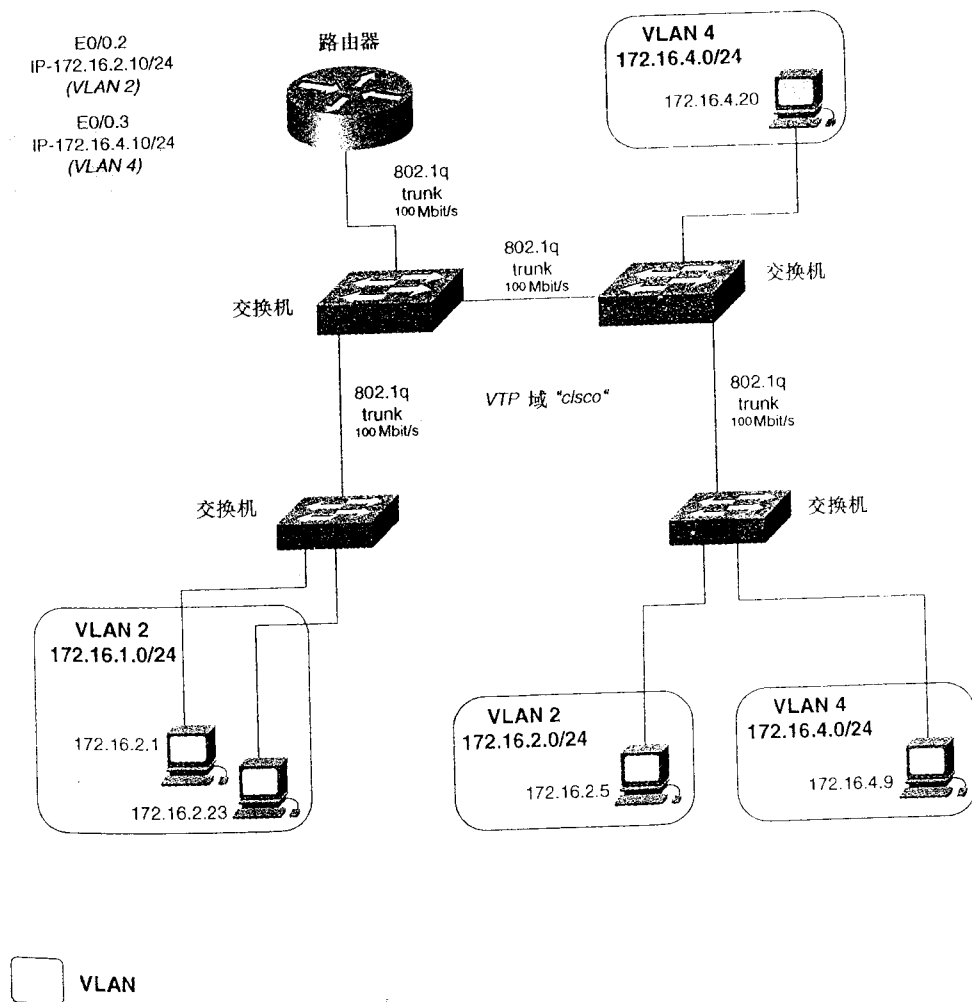


图 2-9 VLAN 中继

以太网有 3 种主要的中继封装形式：

- **交换机间链路（ISL）**——ISL 是 Cisco 的中继封装方式。它是一个帧标记协议，链路中的数据帧包括标准以太网、FDDI 或是令牌环数据帧以及与该帧有关的 VLAN 信息。百兆或者更高速率的连接都支持 ISL，它是一个非常高效的协议，是 Cisco 用于其 Catalyst 内部与路由交换机模块（RSM）或其他第 3 层交换机进行通信的协议。生成树是运行在 ISL 中继上的 PVST，这意味着每个 VLAN 都有一个根桥。在每个中继连接上，每个中继端口在每个 VLAN 上进入转发/拥塞状态。PVST 对于大型网络的控制来说是很关键的，这在下面的内容里会详细讲述到。
- **IEEE 802.1q**——802.1q 是一个工业标准中继协议，其工作方式与 ISL 稍有不同。对于 VTP 域中所有的 VLAN 来说，它用的是运行在默认 VLAN 的单生成树（MST）。在 MST 里，整个 VTP 域选出了一个根桥，叫做公用生成树（CST）。所有 VLAN 信息的传输在这种配置中只有一条路径。因为在大型网络中控制负载的同时还应该控制生成树，Cisco 采用的方式是在 802.1q 上运行 PVST。下面是 802.1q 中继上的一些其他限制因素的列表：
 - 在中继两端的原属 VLAN 应该是一样的。MST 需要运行在匹配的 VLAN 上。第三方的交换机在和 Cisco 交换机通信时，原属 VLAN 双方保持一致是非常关键的。
 - 前面曾经提到，802.1q 使用的是 MST。而 Cisco 则不同，默认使用的是 PVST。由于 BPDU 的处理问题，Cisco 和其他第三方的交换机不一样，所以在将这些由含有生成树和原属 VLAN 的交换机组成的域集成到一起时要特别小心。对于第三方交换机来说，整个 Cisco VTP 域看起来就像是一个独立的广播/生成树域。
 - 中继上原属 VLAN 的 BPDU 不加标记并按照保留的 IEEE 802.1d 生成树多播 MAC 地址（0180.c200.0000）通过中继送到接受方。所有其他中继上 VLAN 中的 BPDU 都加上了保留的 Cisco 共享生成树（SSTP）多播 MAC 地址（0100.0ccc.cccd），在中继上送出去。
- **IEEE 802.10**——802.10 实际上是工业上试用的第 1 个 VLAN 中继协议。刚开始时，其开发的目的是增强美国国家防御网络或其他大型城域网（MAN）的安全性能，由于其自身的一些局限性，现在它主要应用在 FDDI 网络中。

2. 动态 ISL（DISL）和动态中继协议（DTP）

动态 ISL 是 Cisco 的第一个中继协商协议。所有的 4.1 或更早版本的 Catalyst 5500 系列交换机都存在该协议。最初，该协议只是用于处理 ISL 协议协商的问题。后来，在 4.2 版本中，动态中继协议（DTP）逐渐代替了 DISL。DTP 实质上是自动协商 ISL 和 802.1q 中继配置的 DISL。DTP 使用保留的多播 MAC 地址 0100.0ccc.cccc 在 LAN 中进行中继协商。在默认的自动状态下，每隔 30 秒在所有的中继上传送 DTP 信息。根据不同模式，端口中继封装可能是 ISL 或者 802.1q。

DTP 的工作模式有：

- **开启（On）**——所有的端口都处于永久性的中继状态，同时也主动发出建立中继连接的协商。

- 关闭 (Off) ——屏蔽掉端口，即同时屏蔽了中继。
- 尝试方式 (Desirable) ——使某端口主动尝试建立中继连接，该端口在相邻的端口被设置成了 on, desirable 或者 auto 状态的情况下都会建立起中继连接。
- 自动协商方式 (Auto) ——端口在相邻端口处于 on 或 desirable 的状态下时，能够协商建立中继连接。
- 非协商方式 (Nonegotiate) ——将端口设为中继模式但是禁止端口发出 DTP 信息。

实际上，中继有很多的选项。网络管理员有可能会将一个端口设置成中继，也有可能不会。有人认为由于网络变得如此灵活，允许中继或是动态地建立中继连接，会造成安全隐患。这是有道理的表 2-11 列出了中继可能的组合与模式。这里显示，配置中继最可靠和最简单的方法是在连接的两端将其设为中继并且设置为“on”模式。

表 2-11 以太网 VTP 设置结果

相邻端口	中继模式和封装	关闭模式	开启模式	尝试模式	自动模式	开启模式	尝试模式	自动模式	尝试模式	自动模式
		ISL 或 DOT1Q	ISL	ISL	ISL	DOT1Q	DOT1Q	DOT1Q	自协商	自协商
关闭模式	ISL 或 DOT1Q	本: 非中继 邻: 非中继	本: ISL 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继
开启模式	ISL	本: 非中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 1Q 中继 邻: ISL 中继	本: 非中继 邻: ISL 中继	本: 非中继 邻: 非中继	本: ISL 邻: ISL	本: ISL 邻: ISL
尝试模式	ISL	本: 非中继 邻: 非中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 1Q 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: ISL 邻: ISL	本: ISL 邻: ISL
自动模式	ISL	本: 非中继 邻: 非中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: ISL 邻: ISL	本: 非中继 邻: 非中继
开启模式	DOT1Q	本: 非中继 邻: 1Q 中继	本: ISL 中继 邻: 1Q 中继	本: 非中继 邻: 1Q 中继	本: 非中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继
期望模式	DOT1Q	本: 非中继 邻: 非中继	本: ISL 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继
自动模式	DOT1Q	本: 非中继 邻: 非中继	本: ISL 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 非中继 邻: 非中继
期望模式	协商	本: 非中继 邻: 非中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: ISL 中继 邻: ISL 中继	本: ISL 邻: ISL	本: ISL 邻: ISL

续表

相邻 端口	中继模 式和 中继封装	关闭 模式	开启 模式	尝试 模式	自动 模式	开启 模式	尝试 模式	自动 模式	尝试 模式	自动 模式
		ISL 或 DOT1Q	ISL	ISL	ISL	DOT1Q	DOT1Q	DOT1Q	自协商	自协商
自动 模式	协商	本： 非中继 邻： 非中继	本： ISL 邻： ISL 中继	本： ISL 邻： ISL 中继	本： 非中继 邻： 非中继	本： 1Q 中继 邻： 1Q 中继	本： 1Q 中继 邻： 1Q 中继	本： 非中继 邻： 非中继	本： ISL 邻： ISL	本： 非中继 邻： 非中继

2.4.4 Catalyst 以太网交换机的设置

以太网交换机即现在的 Catalyst 交换机有一段不平凡的历史。Catalyst 交换机每个系列配置的命令行都略有不同。比如来源于 Grand Junction 的 1900 和 2800Catalyst 系列交换机有其独特的配置方式，而来自 Kalpana 的 Catalyst 3000 系列交换机配置方法和其他系列大相径庭。Catalyst 5500 系列交换机则是 Crescendo 的产品。Catalyst 5500 系列和 6500 系列的配置方法最初称为 XDI，现在简称为命令行接口 (CLI)，而 Catalyst 8500 系列则是采用与传统 Cisco 路由器类似的混合配置方法，不同之处在于它就像是具有上百个端口的路由器。

下面的章节里，除讲述令牌环一节以外，都侧重于 Catalyst 2900XL/3500G 系列和 Catalyst 4000/5500/6500 系列交换机的配置。这两大系列交换机代表了目前所用绝大多数的 Cisco Catalyst 系列交换机。

LAN 交换机的设计尽量做到便于安装和配置。在小型网络里，只需做很少或根本不需要配置。而在较大的具有多个 VLAN 和中继的网络中，就需要详细配置交换机了。以太网交换机的配置可以分成 4 个步骤。大多数 (不是全部) 的交换机都需要配置一个非默认的 VTP 域，才能创建 VLAN。

第 1 步 配置交换机管理。

第 2 步 配置 VTP 和 VLAN。

第 3 步 配置 VLAN 中继 (如果选用)。

第 4 步 (可选) 控制 STP 和 VLAN 的广播。

第 1 步包括交换机上的管理 VLAN、IP 地址和默认网关的设置，以便能够在 Internet 网络中对其进行访问。

第 2 步要定义 VTP 域和 VTP 服务器上的 VLAN。在这一步里，还要给 VLAN 分配端口。

第 3 步如果网络中需要 VLAN 中继，就对其进行设置。

第 4 步是可选的但是对大型网络来说非常关键。它包括通过根桥的设置来控制 STP，通过使用 VLAN 修剪减少中继上多余的 VLAN。

1. 第 1 步：配置交换机管理信息

所有的 Catalyst 交换机都可以通过 IP 地址进行管理。要达到管理的目的，应该给交换机分配一个 IP 地址，一个默认网关或是用来转发 IP 数据的默认路由。默认的管理 VLAN 是 VLAN 1。给交换机分配 IP 地址之后，除非指定成另一个 VLAN，否则该地址就属于 VLAN 1。

设置 Catalyst 4000/5500/6500 交换机的管理接口

Catalyst 4000/5500/6500 交换机的管理接口称为 SC0 接口。这是一个逻辑带内接口，即依靠其他交换机端口来为其收发数据。4000 系列还含一个 ME1 的带外管理接口。使用下面的语法结构可以设置 Catalyst 交换机的 IP 地址：

```
set interface sc0 [vlan] [ip_addr [netmask [broadcast]]]
```

还需要设置一个默认路由以便能够将数据转发。通常需要一台与交换机的接口相连且 IP 地址处于同一 VLAN 中的路由器。默认路由的设置方法有两种：输入默认网关或指向 0.0.0.0 的默认路由。

```
set ip route default ip_default_gateway
```

或

```
set ip route 0.0.0.0 IP_default_gateway
```

要确保 IP 的可达性，必须保证路由器接口地址和交换机管理地址在同一个子网/VLAN 中。交换机的默认网关 IP 地址应该是路由器的以太网接口或子接口。

例 2-1 给出了交换机 sw13 的 IP 地址和默认网关。默认网关在 VLAN2，因为 VLAN2 还在数据库里，因此在步骤 2 中要对 VLAN2 进行如下例中的配置以使其正常工作。

例 2-1 IP 地址和默认路由的配置

```
sw13 (enable) set int sc0 2 172.16.2.13 255.255.255.0
Interface sc0 vlan set, IP address and netmask set.
sw13 (enable) set ip route default 172.16.2.10
Route added.
```

注释 对于 Catalyst 4000/5500/6500 系列交换机，如果 SC0 接口不在 VLAN1 中，必须配置 VTP 域以及适当的 VLAN 以使其正常工作。

在配置 Catalyst 4000/5500/6500 系列交换机管理接口时，还有一些很有用的命令：

- **set prompt**——设置交换机的提示符，例如路由器上的 **hostname** 命令。
- **set system contact**——设置技术支持人员的姓名或电话号码。
- **set system location**——描述交换机的安装位置。
- **set ip route ip_subnet ip_next_hop**——允许在路由表中输入特定路由。输入的下一条路由必须可以通过 SC0 或 ME1 接口访问。
- **show ip route**——显示已有 IP 路由表或默认路由以及相应的下一条地址。

例 2-2 为配置静态路由，并给出 **show ip route** 命令的输出结果。

例 2-2 静态路由的设置

```
sw13 (enable) set ip route 172.18.2.0 172.16.2.10
Route added.
sw13 (enable) show ip route
Fragmentation    Redirect    Unreachable
.....
enabled          enabled    enabled

Destination      Gateway      Flags    Use      Interface
.....
```

(待续)

default	172.16.2.10	UG	165	sc0
172.18.2.0	172.16.2.10	UG	0	sc0
172.16.2.0	172.16.2.13	U	279	sc0
sw13 (enable)				

例 2-3 使用 **show system** 命令显示设定的系统值。

例 2-3 show system 命令的输出结果

sw13 (enable) show system						
PS1-Status	PS2-Status	Fan-Status	Temp-Alarm	Sys-Status	Uptime d,h:m:s	Logout
ok	ok	ok	off	ok	0,06:59:37	20 min
PS1-Type	PS2-Type	Modem	Baud	Traffic Peak	Peak-Time	
WS-C4008	WS-C4008	disable	9600	0%	0% Thu Jun 14 2001, 09:01:43	
System Name		System Location		System Contact		
switch13		CCIE Lab		Solie		
sw13 (enable)						

注释

VLAN 1 的缺点

VLAN 1 是所有 Catalyst 系列交换机的默认 VLAN。802.1q 的 MST 在其整个的生成树域中都使用该 VLAN。VLAN 1 上有对其数据量进行控制的能力。不能从任一 VLAN 中继中删除 VLAN1。任一新加入网络的交换机都被默认分在 VLAN1 中。这样会造成 VLAN1 上可能存在潜在的数据错误，使网络显得非常不稳定。因此，通常不要在 VLAN1 这样的“恒定”VLAN 上传递用户数据或管理信息。

在 Catalyst 4000/5500/6500 交换机上设置 IP 访问列表

当交换机上设置了 IP 地址后，可以通过 Telnet 和 SNMP 对其进行访问，无需加入其他设置。但有时为了限制 Telnet 或 SNMP 对交换机的访问，可以使用 **set ip permit** 命令。该命令可以设置最多 10 行。整个网络或者单个 IP 地址都可以用该命令进行过滤。例如，只允许网络 172.16.2.0/24 访问，其命令如下：

```
set ip permit 172.16.2.0 255.255.255.0
```

如要限制某个 IP 地址，可用掩码 255.255.255.255 或不写掩码。输完这些项之后，用 **set ip permit enable** 命令激活该访问列表。这样交换机就只允许 ICMP 请求和应答，而禁止 SNMP 和 Telnet 的服务。默认情况下关闭 IP 允许列表，使用时需将其激活，其命令格式如下：

```
set ip permit [ ip_address ] [ subnet_mask ]
set ip permit [enable | disable]
```

IP 访问列表可以通过 **show ip permit** 命令来查看，如例 2-4。

例 2-4 show ip permit 命令输出结果

```
sw13 (enable) show ip permit
IP permit list feature enabled.
```

(待续)

Permit List	Mask	
-----	-----	
172.16.2.0	255.255.255.0	
Denied IP Address	Last Accessed Time	Type
-----	-----	-----
172.16.3.1	06/14/01, 19:07:43	Telnet
sw13 (enable)		

Catalyst 2900XL/3500G 交换机管理接口的设置

Catalyst 系列交换机的管理接口如同一台加入特殊 *VLAN 数据库* 的路由器。分配端口，配置中继及其管理有关的命令都是在传统配置模式(或称全局配置模式[*conf t mode*])下进行的，VLAN 信息通过 **vlan database** 命令进行配置。以后，VLAN 数据库中的命令就姑且称为 *VLAN 命令*，也就是可通过输入 **vlan database** 进入 VLAN 配置模式来使用。在传统路由器模式下输入的命令称为 *配置模式命令*。

2900XL/3500G 交换机有一默认虚拟接口称为 VLAN 1，是交换机的默认 VLAN。如果要在 VLAN 1 中进行管理接口 IP 的配置，只需像在路由器里那样加入 IP 地址即可。如果在其他 VLAN 上配置管理接口，则需关闭 VLAN 1 接口，并为所需配置 VLAN 设置一个虚拟接口。不能同时激活两个 VLAN，要想使 VLAN 2 工作，需关闭 VLAN 1。例 2-5 给出了如何在另一 VLAN 上配置管理接口的过程。

例 2-5 配置一台 Catalyst 2900XL/3500G 交换机的管理接口

```
sw11#conf t
Enter configuration commands, one per line. End with CNTL/Z.
sw11(config)#interface vlan 1
sw11(config-if)#shut
sw11(config-if)#exit
sw11(config)#interface vlan 2
sw11(config-subif)#ip address 172.16.2.11 255.255.255.0
sw11(config-subif)#no shut
sw11(config-subif)#^Z
sw11#
```

在这个例子里，管理接口只有在 VLAN 2 定义之后才能够工作。这是前面所讲的配置过程的 4 个步骤中的第 2 步。

和路由器一样，可用 **ip default gateway ip_address** 命令设置默认路由。例 2-6 给出了如何配置默认网关的例子。这里的默认网关是路由器 172.16.2.10。

例 2-6 设置 Catalyst 2900XL/3500G 交换机的默认网关

```
sw15(config)#ip default-gateway 172.16.2.10
```

Catalyst 2900XL/3500G 交换机的 IP 访问控制

Catalyst 2900XL/3500G 交换机的 IP 访问控制和路由器上的 Telnet 控制完全一致。请回顾第 1 章“建立网络互联模型的主要组件”里 Telnet 访问是如何在 VTY 线路中进行控制的。在交换机里，同样可用 **show line** 命令查看这些内容。Telnet 和 SNMP 能够通过创建访问表并将其用于交换机 VTY 线路实现。由于这些内容和路由器完全一样，请参考第 1 章配置虚拟 Telnet 访问的内容。

2. 第2步：在 Catalyst 4000/5500/6500 交换机上设置 VTP 和 VLAN

这一步包括 3 个步骤，都用 **set** 命令来实现。过程如下：

第1步 设置 VTP 域和模式。

第2步 设置物理端口特性，并为 VLAN 分配端口。

第3步 如果交换机作为 VTP 服务器或在 VTP 透明模式下工作，需要配置 VLAN。

3. 在 Catalyst 4000/5500/6500 上设置 VTP 域及其工作模式

在将 VLAN 加入 VLAN 数据库前，必须对 VTP 域进行设置。

设置 VTP 域时，用下面的语法命令：

```
set vtp [domain name] [mode {client | server | transparent}] [passwd passwd]
[pruning {enable | disable}] [v2 {enable | disable}]
```

name 字段是设置 VTP 域的名称，区分大小写。VTP 的默认工作模式是服务器模式。如要改变工作模式，可以改用客户（client）、服务器（server）或是透明（transparent）模式。请记住，VLAN 服务器上的任何改动都将广播到所有的客户端 VLAN 上去。VLAN 服务器要将更新信息传到某个客户端去，该服务器的配置更改号必须大于客户端的。如果客户端的修改号大，它不会接收服务器传来的更新信息。如果 VLAN 广播过程中出问题，一定要先检查各自的更改号。要将一个 VTP 的修改号复位为 0，只需改变 VTP 域名，再改回即可。但这一方法在 2900XL/3500G 系列交换机上无效，这类交换机必须重启才能清除该 VTP 域。

允许版本 2（V2）更新与交换机类型有关。只有在令牌环交换机时才需要 V2 更新的信息。这就是为什么以太网交换机没有 V2 更新的原因。该模式下工作时，VTP 域中所有的交换机都必须具有 V2 功能。

VTP 还可使用 MD5 散列加密（hash）的密码来保护 VTP 更新信息，这一功能可在 VTP 域中用 **password** 命令实现。例 2-7 给出了用 MD5 加密的密码 CCIE 来配置 VTP 域 **ciscomd5** 的例子。

例 2-7 配置一个密码保护的 VTP 域

```
sw13 (enable) set vtp domain ciscomd5 password ccie
Generating MD5 secret for the password ....
VTP domain ciscomd5 modified
sw13 (enable)
```

可以像例 2-8 那样用 **show vtp domain** 命令来查看 VTP 域信息。

例 2-8 配置一个密码保护的 VTP 域

```
sw13 (enable) show vtp domain
Domain Name          Domain Index VTP Version Local Mode Password
-----
ciscomd5              1            2            server    configured

Vlan-count Max-vlan-storage Config Revision Notifications
-----
9           1023              0            disabled

Last Updater      V2 Mode Pruning PruneEligible on VLANs
```

（待续）

```
172.16.2.13    disabled 2-1000
sw13 (enable)
```

show vtp domain 命令列出了 VTP 域的域名、更改号和 VTP 模式以及密码保护等信息，还显示了域中的 VLAN 数目以及可更改的内容。其中最近更新 (Last Update) 列是指收到上一个 VTP 更新交换机的 IP 地址。上例中，最近更新是从 172.16.2.13 收到的。

在 Catalyst 4000/5500/6500 上设置端口物理特性以及为 VLAN 分配端口

下面的两个步骤可合并，视交换机是配置成服务器模式、透明模式还是客户端模式而定。从本质上说，这一步骤包括配置 VLAN 和端口的特性。如果交换机是工作在客户端模式，根本无需配置 VLAN。

在 Catalyst 交换机中，非中继的每个端口都会分配给默认 VLAN，即 VLAN 1。如果端口分配给了其他 VLAN，那个 VLAN 必须要在 VLAN 数据库中创建。配置中继时，VTP 服务器中创建的 VLAN 信息将会传送给其他 VTP 服务器和客户端。

这一步还要求设置端口以太网的物理特性，如双工模式、端口速率等等。下面的命令就是一些常用的端口配置命令：

- **set port disable [mod_num/port_num]**——禁用端口，与路由器的 **shutdown** 命令用法同。
- **set port enable [mod_num/port_num]**——激活端口，相当于路由器的 **no shutdown** 命令。
- **set port duplex [mod_num/port_num] [full|half]**——设置端口的工作模式为全双工还是半双工。
- **set port name [mod_num/port_num] port_name**——为某个端口分配逻辑端口名，可以用 **show port** 命令查看。
- **set port speed [mod_num/port_num] [10|100|auto]**——设置端口的传输速率为 10、100 兆或自适应 (autonegotiation)。目前吉比特以太网的端口的速率固定在 1000 mbit/s，将来可能会有改变。
- **set port level [mod_num/port_num] [normal|high]**——在 Catalyst 4000/5500/6500 系列上，如果两个端口同时访问交换机的总线，先满足优先级高的请求。

端口状态可以用 **show port** 命令来查看。该命令会列出交换机所有的端口，还包括端口的 VLAN 号、连接状态、双工设置、速率以及接口类型等信息。例 2-9 列出了使用该命令后的显示信息。这里可以看出端口逻辑名有助于识别端口的功能。该例子中，还用 **set port level** 命令使端口 2/19 获得了较高优先级。

例 2-9 show port 命令的输出结果

```
sw13 (enable) show port
Port  Name                Status    Vlan    Level Duplex Speed Type
-----
2/1   gigabit_trunk_sw11    connected trunk    normal full  1000 1000BaseSX
2/2   gigabit_trunk_sw12    connected trunk    normal full  1000 1000BaseSX
2/3                   notconnect 1        normal auto   auto  10/100BaseTX
2/4                   notconnect 1        normal auto   auto  10/100BaseTX
2/5                   notconnect 1        normal auto   auto  10/100BaseTX
```

2/6	notconnect	1	normal	auto	auto	10/100BaseTX
2/7	notconnect	1	normal	auto	auto	10/100BaseTX
2/8	notconnect	1	normal	auto	auto	10/100BaseTX
2/9	notconnect	1	normal	auto	auto	10/100BaseTX
2/10	notconnect	1	normal	auto	auto	10/100BaseTX
2/11	notconnect	1	normal	auto	auto	10/100BaseTX
2/12	notconnect	1	normal	auto	auto	10/100BaseTX
2/13	notconnect	1	normal	auto	auto	10/100BaseTX
2/14	connected	800	normal	a-full	a-100	10/100BaseTX
2/15	notconnect	200	normal	auto	auto	10/100BaseTX
2/16	notconnect	200	normal	auto	auto	10/100BaseTX
2/17	notconnect	200	normal	auto	auto	10/100BaseTX
2/18	notconnect	200	normal	auto	auto	10/100BaseTX
2/19 internet_conn	connected	100	high	a-half	a-10	10/100BaseTX
2/20 100_trunk_sw15	connected	trunk	normal	a-full	a-100	10/100BaseTX

还可使用包括端口号的 **show port** 命令来查看端口的更详细的信息，除了上面内容，还有端口物理特性方面的详细资料，如安全性、端口错误以及冲突等。例 2-10 就是这样的例子。

例 2-10 详细的端口信息

```
sw13 (enable) show port 2/1
```

Port	Name	Status	Vlan	Level	Duplex	Speed	Type
2/1	gigabit_trunk_sw11	connected	trunk	normal	full	1000	1000BaseSX

Port	Security	Secure-Src-Addr	Last-Src-Addr	Shutdown	Trap	IfIndex
2/1	disabled			No	disabled	9

Port	Send FlowControl	Receive FlowControl	RxPause	TxPause	Unsupported
	admin oper	admin oper			opcodes
2/1	desired off	off off	0	0	0

Port	Status	Channel mode	Channel status	Neighbor device	Neighbor port
2/1	connected	auto	not channel		

Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize
2/1		0	0	0	0

Port	Single-Col	Multi-Coll	Late-Coll	Excess-Col	Carri-Sen	Runts	Giants
2/1	0	0	0	0	0	0	0

Last-Time-Cleared

Sat Jun 16 2001, 13:29:17

sw13 (enable)

4. 在 Catalyst 4000/5500/6500 系列交换机上设置 VLAN

在 Catalyst 4000/5500/6500 系列交换机中，VLAN 是用 **set vlan** 命令来创建的。只需在该命令后面加上端口号就可以将该端口分配给创建的 VLAN。

set vlan [1-1001] [mod/ports]

set vlan 能够自动创建未曾定义过的 VLAN，并为其分配端口。一个 VLAN 可以分配多

个端口，各个端口之间用 “,” 隔开，或者用 “-” 指定端口范围。例如，将端口 1/1 和 1/12 分配给 VLAN 2，其命令如下：

```
set vlan 2 1/1,1/12
```

要将端口 1/10 和 1/2，2/2 以及 2/3 分配给 VLAN 3，命令为：

```
set vian 3 1/10,2/1-2/3
```

例 2-11 给出创建 VLAN 33，并且把端口 2/5，2/10，2/11，2/12 和 2/13 分配给 VLAN33 的例子。

例 2-11 VLAN 的创建

```
sw13 (enable) set vlan 33 2/5,2/10-2/13
Vlan 33 configuration successful
VLAN 33 modified.
VLAN 1 modified.
VLAN Mod/Ports
-----
33      2/1-2,2/5,2/10-13,2/20
sw13 (enable)
```

创建 VLAN 时，VLAN 存在一些默认值，如 MTU、可修剪性等。表 2-12 列出了 VLAN 这些默认值的情况。这些值大多可用 `set vlan` 命令修改，其命令句法如下：

```
set vlan 1- 1001 [name { vlan_name}] [state {active | suspend}] [said { said_value}]
[mtu mtu] [bridge { bridge_number}] [stp {ieee | ibm | auto}]
```

命令参数如下：

- **name** ——为 VLAN 取一个最大长度为 32 个字符的名字。
- **state** ——允许挂起 VLAN。挂起的 VLAN 信息仍通过 VTP 广播的，但此时该 VLAN 上没有用户数据传送。
- **Security Association ID (SAID)** ——允许改变 VLAN 的 SAID 值，该值在 802.10 中使用。
- **mtu、bridge 和 stp** ——改变默认的 MTU，网桥号以及 STP 类型值。改变 MTU，网桥号及 STP 类型的值时要非常小心。

表 2-12 默认 VLAN 设置

特 点	默 认 值
本地或默认 VLAN	VLAN1
端口分配	所有端口分配给 VLAN1；令牌环端口分配给 VLAN1003
VTP 模式	SERVER
VTP 状态	激活
普通 VLAN 范围	VLAN 2——1001
VLAN 保留范围*	VLAN1006——1009
VLAN 扩展范围*	VLAN1006——1009
MTU 大小	以太网 1500byte 令牌环 4472byte

续表

特 点	默 认 值
SAID 值	100,000 加 VLAN 号 VLAN2=SAID 100002
可修剪性	VLAN2——1000 允许修剪
MAC 地址缩减	禁止
生成树模式	PVST
默认 FDDI VLAN	VLAN1002
默认令牌环 TrCRF VLAN	VLAN1003
默认 FDDI Net VLAN	VLAN1004
默认令牌环 TrBRF VLAN	VLAN1005,网桥号 0F
TrBRF VLAN 生成树的版本	IBM
TrCRF 网桥模式	SRB

* VLAN 保留范围是用在 6500 系列交换机上映射到非保留 VLAN 的。扩展范围的 VLAN 在 6500 系列的交换机上是可用的，是普通 VLAN 的扩展。保留范围的 VLAN 到至今还不能被 VTP 传送。令牌环和 FDDI 的 VLAN 在以太网的交换机里面仅仅是作为 VTP 的全局信息列出，同样地，令牌环交换机也会列出以太网 VLAN 的信息。

交换机上的 VLAN 可以通过两种方式来查看。**show vlan** 命令列出交换机上所有 VLAN 的概要信息、分配的端口以及默认 VLAN 设置。例 2-12 列出了 **show vlan** 命令的输出结果。可见，VLAN 用指定的名称看起来非常清晰。赋予 VLAN 名称有助于网络结构自身的清晰化和规范化。

例 2-12 show vlan 命令输出结果

sw13 (enable) show vlan										
VLAN Name	Status	IfIndex	Mod/Ports	Vlans						
1 default	active	4	2/3-4,2/6-9,2/21-34							
2 management_VLAN	active	64								
3 Engineering_VLAN	active	65								
4 VLAN0004	active	70								
5 VLAN0005	active	71								
33 VLAN0033	active	72	2/5,2/10-13							
100 Internet_VLAN	active	66	2/19							
200 dummy_VLAN	active	67	2/15-18							
800 VLAN0800	active	68	2/14							
801 VLAN0801	active	69								
1002 fddi-default	active	5								
1003 token-ring-default	active	8								
1004 fddinet-default	active	6								
1004 fddinet-default	active	6								
VLAN Type	SAID	MTU	Parent	RingNo	BrdgNo	Stp	BrdgMode	Trans1	Trans2	
1 enet	100001	1500	-	-	-	-	-	0	0	
2 enet	100002	1500	-	-	-	-	-	0	0	

(待续)

```
3   enet 100003 1500 - - - - - 0 0
4   enet 100004 1500 - - - - - 0 0
5   enet 100005 1500 - - - - - 0 0
33  enet 100033 1500 - - - - - 0 0
100 enet 100100 1500 - - - - - 0 0
200 enet 100200 1500 - - - - - 0 0
800 enet 100800 1500 - - - - - 0 0
801 enet 100801 1500 - - - - - 0 0
1002 fddi 101002 1500 - - - - - 0 0
1003 trcrf 101003 1500 - - - - - 0 0
1004 fdnet 101004 1500 - - - ieee 0 0
1005 trbrf 101005 1500 - - - ibm 0 0
```

VLAN AREHops STEHops Backup CRF

1003 0 0 off
sw13 (enable)

VLAN AREHops STEHops Backup CRF

1003 0 0 off
sw13 (enable)

在 **show vlan** 命令后面加上 VLAN 号可以显示该 VLAN 的详细信息。例 2-13 为 **show vlan 2** 命令的输出结果。

例 2-13 show vlan2 命令的输出结果

sw13 (enable) show vlan 2

VLAN Name	Status	IfIndex	Mod/Ports, Vlans
-----------	--------	---------	------------------

2 management_VLAN	active	64	2/1-2,2/20
-------------------	--------	----	------------

VLAN Type	SAID	MTU	Parent	RingNo	BrdgNo	Stp	BrdgMode	Trans1	Trans2
-----------	------	-----	--------	--------	--------	-----	----------	--------	--------

2 enet	100002	1500	-	-	-	-	-	0	0
--------	--------	------	---	---	---	---	---	---	---

VLAN AREHops STEHops Backup CRF

sw13 (enable)

用 **clear vlan vlan_number** 命令可以将 VLAN 从数据库中删除。只有当交换机是处在 VTP 服务器或透明工作模式时才能删除 VLAN。VLAN 从一台 VTP 服务器上删除之后，就从整个 VTP 域中删掉。VTP 域中所有的交换机，VTP 服务器以及客户端 VTP 都从各自的数据库中删除该 VLAN。因此，删除 VLAN 时一定要非常小心。交换机会在最终删除该 VLAN 之前提示用户，如例 2-14 所示。但只有 Catalyst 4000/5500/6500 才会在删除之前发出警告。Catalyst 2900XL / 3500G 的交换机上只要执行了该命令，删除操作就会执行。

例 2-14 VLAN 的删除

sw13 (enable) clear vlan 801

This command will deactivate all ports on vlan 801

```

in the entire management domain
Do you want to continue(y/n) [n]?y
Vlan 801 deleted
sw13 (enable)

```

5. 第2步：在 Catalyst 2900XL/3500G 交换机上设置 VTP 和 VLAN

同 Catalyst 5500 一样，在 2900XL/3500G 上设置 VTP 和 VLAN 遵从下面 3 个步骤：

第1步 设置 VTP 域及其工作模式；

第2步 设置端口物理特性以及为 VLAN 分配端口；

第3步 如果交换机工作在 VTP 服务器模式，就对 VLAN 进行设置。

在 Catalyst 2900XL/3500G 上设置 VTP 域和工作模式

切记将 VLAN 加入到 VLAN 数据库中之前，必须配置 VTP 域。VLAN 数据库是用来设置交换机的 VLAN 特性的，也就是 VLAN 配置模式。该模式的进入通过特权命令 **vlan database** 来实现。和路由器一样，配置模式用来设置物理端口特性和给 VLAN 分配端口。要进入该模式，键入 **conf t**。

配置 VTP 域的命令如下：

```

Switch#vlan database
(vlan) #vtp domain domain_name [password]

```

如果在域名后面加了密码，VTP 就会用 MD5 散列加密方式来进行加密保护。默认的 VTP 模式是服务器模式。要改变其模式，可以在 VLAN 配置模式下使用下面的命令：

```

(vlan) #vtp [server | client | transparent]

```

可以参考前面的“VTP 和中继协议”一节了解更多的关于 VTP 模式的信息。

用 **show vtp status** 命令可以查看 VTP 域的信息。该命令会显示 VTP 域相关的信息，如设置更改号、域名、工作模式等。例 2-15 列出了该命令的输出结果。

例 2-15 查看 VTP 域信息

```

Switch#show vtp status
VTP Version                : 2
Configuration Revision      : 28
Maximum VLANs supported locally : 254
Number of existing VLANs    : 13
VTP Operating Mode          : Server
VTP Domain Name             : cisco5
VTP Pruning Mode            : Disabled
VTP V2 Mode                 : Disabled
VTP Traps Generation        : Disabled
MD5 digest                  : 0xD9 0x50 0xE2 0x4F 0x09 0xDE 0x98 0x07
Configuration last modified by 172.16.2.13 at 6-17-01 18:10:24
sw11#

```

注释 只有当 VTP 服务器的更改号大于客户端 VTP 的更改号时，VLAN 的信息才会广播到客户端。如果客户端的更改号等于或大于服务器，则不接收任何 VLAN 信息。对于 Catalyst 4000/5500/6500 系列交换机，当前 VTP 的更改号可以用命令 **show vtp domain** 来查看，而对于 Catalyst 2900/3500 系列交换机，则是用 **show vtp status** 命令来查看。

在 Catalyst 2900XL/3500G 上设置端口物理特性以及为 VLAN 分配端口

2900XL/3500G 交换机上设置 VTP 和 VLAN 的下一步是设置所有端口物理特性，并给端口分配 VLAN 号。端口物理特性是在接口配置模式下设置的，就像路由器上一样。例 2-16 是在 2800 系列交换机上将一个以太网端口设置成 10 mbit/s 的全双工端口的例子。该例子中还为接口分配了逻辑名 `internet_port`。

例 2-16 端口物理特性的设置

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
04:59:58: %SYS-5-CONFIG_I: Configured from console by console
Switch(config)#interface fastEthernet 0/6
Switch(config-if)#speed 10
Switch(config-if)#duplex full
Switch(config-if)#description Internet_port
Switch(config-if)#exit
```

可设置的一些以太网端口的物理特性如下：

- **duplex [full | half | auto]**——设置端口的双工模式。
- **speed [10 | 100 | auto]**——设置端口速率。
- **mtu [1500bytes 2018bytes]**——设置接口的 MTU。改变该值时请确认物理接口的 MTU 值和 VLAN 匹配。
- **description interface_description**——设置接口的逻辑说明。
- **shutdown | no shutdown**——关闭或激活接口。

接口命令 **switchport** 用来将端口分配给 VLAN，有 3 种形式。端口可以设置成中继方式，或者是运行一个或多个 VLAN。本节的重点是把端口分配给一个 VLAN，为此，设置端口为接入模式，然后将 VLAN 分配给该端口，命令句法为：

```
(config if) #switchport mode [access | multi | trunk]
```

- **access**——接入模式：接口属于单个的 VLAN。
- **multi**——多 VLAN 模式：将接口分配给多个 VLAN，这时的 VTP 域必须是在透明模式下工作，接口必须连接到交换机或路由器。
- **trunk**——Trunk 模式：将接口类型配置成 Trunk。在后面的章节中会更多的讨论这一点。

将端口分配给一个 VLAN，命令如下：

```
(config if) #switchport access vlan [ 1-1001 | dynamic]
```

VLAN 的标准范围是 1 到 1001。关键字 **dynamic** 是用于 VMPS 的设置。本书中不讲述 VMPS 内容。要了解关于 VMPS 的详细内容，可以参考 Kennedy Clark 和 Kevin Hamilton 的 *Cisco LAN Switching* 一书。

例 2-17 是把端口 FastEthernet0/5 分配到 VLAN 2。

例 2-17 分配 VLAN 2 给端口 FastEthernet0/5

```
Switch(config)#int fastEthernet 0/5
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

当 VTP 的工作模式设为透明时，可以用 **switchport access vlan** 命令创建 VLAN。如果 VTP 模式是客户模式，在该交换机上就不能对 VLAN 进行配置，只能在服务器交换机上对该 VLAN 进行配置，然后由 VTP 通过 Trunk 传递到客户端交换机上来。

6. 在 Catalyst 2900XL/3500G 上配置 VLAN

在 Catalyst 2900XL/3500G 交换机上配置 VTP 和 VLAN 的第 3 步是当 VTP 工作模式是服务器模式的情况下，进行 VLAN 的设置。通过输入 **vlan [2 1001] options** 命令对 VLAN 进行设置并存放到 VLAN 的数据库中。例 2-18 说明了 VLAN 175（其名称为 backbone）的配置情况。VLAN 的改变必须通过 **apply** 命令来激活。退出 VLAN 数据库也会激活所有更改内容。如果出现了错误，可以用 **abort** 和 **reset** 命令取消所做的修改，**abort** 命令是从 VLAN 数据库中退出，而 **reset** 命令则是取消当前的修改，重新从数据库中读出数据。

例 2-18 VLAN 175 的设置

```
Switch#vlan database
Switch(vlan)#vlan 175 name backbone
VLAN 175 added:
    Name: backbone
Switch(vlan)#apply
APPLY completed.
Switch(vlan)#
```

在该模式里，还可以设置 VLAN 的其他选项，包括：

```
Switch (vlan) # vlan vlan_num [name vlan_name] [state {active | suspend}] [said
said_value] [mtu mtu] [bridge bridge_number] [stp type {ieee | ibm | auto}]
```

- **name**——为 VLAN 分配一个最长 32 个字符的名字。
- **state**——允许挂起 VLAN，挂起的 VLAN 信息仍可通过 VTP 广播，但该 VLAN 不传输任何用户数据。
- **said**——改变 VLAN 的 SAID 值。SAID 值主要是用于 802.10 的。
- **mtu、bridge、STP**——改变默认的 MTU、网桥号以及 STP 类型。

上一节的表 2-11 列出了 VLAN 默认设置的值。

要在 2900XL/3500G 交换机上查看 VLAN 的状态，可以使用和 4000/5500/6500 相同的命令。**show vlan vlan_number** 命令显示的是交换机上所有 VLAN 各自的状态以及分配的端口信息。要显示单个 VLAN 的物理和逻辑信息，可以用 **show vlan id [vlan_number]** 命令来完成。例 2-19 列出 **show vlan** 命令的输出结果，后面还跟着命令更明确的形式。从这里也可以看出 VLAN 逻辑名称的作用。

例 2-19 show vlan 命令输出结果

```
sw11#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,
                                           Fa0/11, Fa0/12, Fa0/13, Fa0/14,
                                           Fa0/15, Fa0/16, Fa0/17, Fa0/18,
                                           Fa0/19, Fa0/22, Fa0/23, Fa0/24,
```

（待续）

```

Fa0/25, Fa0/26, Fa0/27, Fa0/28,
Fa0/29, Fa0/30, Fa0/31, Fa0/32,
Fa0/33, Fa0/34, Fa0/35, Fa0/36,
Fa0/37, Fa0/38, Fa0/39, Fa0/40,
Fa0/41, Fa0/42, Fa0/43, Fa0/44,
Fa0/45, Fa0/46, Fa0/47, Fa0/48,
Gi0/2

2   management_VLAN      active
3   Engineering_VLAN     active   Fa0/1
4   VLAN0004              active
5   VLAN0005              active
33  VLAN0033              active
100 Internet_VLAN         active
200 dummy_VLAN           active
800 VLAN0800              active
1002 fddi-default         active
1003 token-ring-default   active
1004 fddinet-default      active
1005 trnet-default        active

```

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	0	0
2	enet	100002	1500	-	-	-	-	-	0	0
3	enet	100003	1500	-	-	-	-	-	0	0
4	enet	100004	1500	-	-	-	-	-	0	0
5	enet	100005	1500	-	-	-	-	-	0	0
33	enet	100033	1500	-	-	-	-	-	0	0
100	enet	100100	1500	-	-	-	-	-	0	0
200	enet	100200	1500	-	-	-	-	-	0	0
800	enet	100800	1500	-	-	-	-	-	0	0
1002	fddi	101002	1500	-	0	-	-	srp	0	0
1003	tr	101003	1500	-	0	-	-	-	0	0
1004	fdnet	101004	1500	-	-	1	ieee	-	0	0
1005	trnet	101005	1500	-	-	1	ibm	-	0	0

SW11#

```
SW11#show vlan id 3
```

VLAN Name	Status	Ports
3 Engineering_VLAN	active	Fa0/1

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
3	enet	100003	1500	-	-	-	-	-	0	0

SW11#

7. 第3步: 在 Catalyst 4000/5500/6500 交换机上设置 VLAN 中继

配置 Catalyst 交换机的 VLAN 中继分为两步:

第1步 将端口设为中继模式。

第2步 将中继封装形式设为自动协商, 或者使用 ISL 或 802.1q。

设置自动协商, 或 DTP 比静态地设定 Trunk 要困难得多, 主要是因为各种不同的 Catalyst 交换机的默认 Trunk 封装都各不相同。大多数 Catalyst 交换机是 ISL, 而没有第3层模块的 Catalyst 4000 不支持 ISL。另一个例子, 只有 4.2 版本的软件支持 802.1q 自动协商方式。正是由于这些细微区别使得 DTP 在大型不同种类的网络上难以实现。

然而, 这些特性有时却是需要的。表 2-12 列出了 DTP 的所有模式和结果。默认情况下,

所有端口都处于非中继状态。需要配置端口, 使其成为中继端口, 并且在将其配置成某种模

式。请回顾一下，一共有 5 个模式：

- 开启（On）——设置端口处于永久性的中继状态，同时也会主动发出建立中继连接的协商。
- 关闭（Off）——屏蔽端口，同时也屏蔽掉中继。
- 期望模式（Desirable）——使某端口尝试建立中继连接，该端口在相邻的端口被设置成 on desirable 或者 auto 状态的情况下都会建立中继连接。
- 自动协商模式（Auto）——端口在相邻端口处于 on 或 desirable 的状态下时，能够协商建立中继连接。
- 非协商模式（Nonegotiate）——将端口设为中继模式但是禁止端口发出 DTP 信息。

将端口设置成 Trunk 之后，必须设置其封装形式。封装形式共有 3 种：ISL、802.1q/DOT1Q 和自动协商模式。自动协商的工作方式是，首先尝试使用 ISL，接着是 802.1q。完成这两步的句法是：

```
Switch(enable) set trunk mod_num/port_num [on | off | desirable | auto | nonegotiate]
Switch(enable) set trunk mod_num/port_num [isl | dot1q]
```

如前所述，即使不参考表 2-13 也可知，设置 VLAN TRUNK 最可靠最快捷的方式是将中继模式设为开启模式，并且将封装形式固定为 ISL 或 802.1q/dot1q。例 2-20 给出配置端口 2/6 上的 802.1q 的例子。

例 2-20 配置端口 2/6 上的 802.1q

```
Switch(enable) set trunk 2/6 dot1q
Port(s) 2/6 trunk type set to dot1q.
Switch(enable) set trunk 2/6 on
Port(s) 2/6 trunk mode set to on.
Switch(enable) 2001 Jun 12 09:33:58 %DTP-5-TRUNKPORTON:Port 2/6 has become dot1q trunk
Switch(enable) 2001 Jun 12 09:34:11 %PAGP-5-PORTTOSTP:Port 2/6 joined bridge port 2/6
```

表 2-13

以太网 DTP 的设置结果

邻居 端口	中继模 式和 中 继封装	关闭 模式	开启 模式	尝试 模式	自动 模式	开启 模式	尝试 模式	自动 模式	尝试 模式	自动 模式
		ISL 或 DOT1Q	ISL	ISL	ISL	DOT1Q	DOT1Q	DOT1Q	自协商	自协商
关闭 模式	ISL 或 DOT1Q	本： 非中继 邻： 非中继	本：ISL 中继 邻： 非中继	本： 非中继 邻： 非中继	本： 非中继 邻： 非中继	本：1Q 中继 邻： 非中继	本： 非中继 邻： 非中继	本： 非中继 邻： 非中继	本： 非中继 邻： 非中继	本： 非中继 邻： 非中继
开启 模式	ISL	本： 非中继 邻：ISL 中继	本：ISL 中继 邻：ISL 中继	本：ISL 中继 邻：ISL 中继	本：ISL 中继 邻：ISL 中继	本：1Q 中继 邻：ISL 中继	本： 非中继 邻：ISL 中继	本： 非中继 邻： 非中继	本：ISL 邻：ISL	本：ISL 邻：ISL
尝试 模式	ISL	本： 非中继 邻： 非中继	本：ISL 中继 邻：ISL 中继	本：ISL 中继 邻：ISL 中继	本：ISL 中继 邻：ISL 中继	本：1Q 中继 邻： 非中继	本： 非中继 邻： 非中继	本： 非中继 邻： 非中继	本：ISL 邻：ISL	本：ISL 邻：ISL

续表

邻居 端口	中继模 式和 封装	关闭 模式	开启 模式	尝试 模式	自动 模式	开启 模式	尝试 模式	自动 模式	尝试 模式	自动 模式
		ISL 或 DOT1Q	ISL	ISL	ISL	DOT1Q	DOT1Q	DOT1Q	自协商	自协商
自动 模式	ISL	本: 非中继 邻: 非中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: ISL 邻: ISL	本: 非中继 邻: 非中继
开启 模式	DOT1Q	本: 非中继 邻: 1Q 中继	本: ISL 中继 邻: 1Q 中继	本: 非中继 邻: 1Q 中继	本: 非中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继
尝试 模式	DOT1Q	本: 非中继 邻: 非中继	本: ISL 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继
自动 模式	DOT1Q	本: 非中继 邻: 非中继	本: ISL 中继 邻: 非中继	本: 非中继 邻: 非中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 非中继 邻: 非中继
尝试 模式	自协商	本: 非中继 邻: 非中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: ISL 中继 邻: ISL 中继	本: ISL 邻: ISL	本: ISL 邻: ISL
自动 模式	自协商	本: 非中继 邻: 非中继	本: ISL 中继 邻: ISL 中继	本: ISL 中继 邻: ISL 中继	本: 非中继 邻: 非中继	本: 1Q 中继 邻: 1Q 中继	本: 1Q 中继 邻: 1Q 中继	本: 非中继 邻: 非中继	本: ISL 邻: ISL	本: 非中继 邻: 非中继

注释 VTP 和 DISL 二者之间还有自动协商的问题。DISL 协商 ISL Trunk 时，在协商信息中包含 VTP 域名。如果交换机上的 VTP 域名不同，中继链路就停止工作。要想解决这个问题，只需要把中继设置成开启模式，再配置中继的封装形式即可。

用下面的命令可以查看中继链路的状态：

```
show trunk [detail]
show trunk [ mod_num/port_num] [detail]
show vtp status
```

例 2-21 列出了 show trunk 命令的输出结果。如果没有显示中继连接，需要注意以下关键字段：

- 状态 (state)。
- 模式 (mode)。
- 封装 (encapsulation)。
- 管理域中许可的和已激活的 VLAN。
- 对等端口 (peer-port)。

中继链路的状态应为 trunking，模式为 on 或者是与 DTP 的一个正确设置相匹配，如表

明了中继链路传送的 VLAN 信息。如果没有列出指定 VLAN, 那么中继设置就不正确。802.1q 会将原属 VLAN 用于生成树 (MST) 中。整个 VTP 域中的相同 ID 的 VLAN 必须是同一个 VLAN。

例 2-21 show trunk 命令输出结果

Switch (enable) show trunk detail				
Port	Mode	Encapsulation	Status	Native vlan
2/1	on	dot1q	trunking	1
2/2	on	dot1q	trunking	1
Port	Peer-Port	Mode	Encapsulation	Status
2/1	GigabitEt	unknown	unknown	unknown
2/2	GigabitEt	unknown	unknown	unknown
Port	Vlans allowed on trunk			
2/1	1-1005			
2/2	1-1005			
Port	Vlans allowed and active in management domain			
2/1	1-5,33,100,200,800			
2/2	1-5,33,100,200,800			
Port	Vlans in spanning tree forwarding state and not pruned			
2/1	1-5,33,100,200,800			
2/2	1-3			
Switch (enable)				

有时很难确定中继线路是否在工作。中继线路可能显示 trunking 状态, 但却没有发送完整的 VTP 更新记录。应该在链路两端查看中继状态, 以确保其正常工作。另一个确定中继是否识别出链路另一端状态的方法是观察对等端口状态。如果对等端口的状态为未知, 那么意味着出现了封装不匹配问题, Trunk 链路没有正常工作。

域中 VTP 达到同步时, 服务器到服务器、服务器到客户端的 VLAN 数据库中的 VLAN 应该都一样。只有 VTP 透明模式下或者是中继的 VLAN 被清除的交换机才有可能具有不同的 VLAN 数据库。因此, 通过比较 VLAN 中继相连的两台交换机 VLAN 数据库, 成为验证中继链路是否正常工作的又一方法。

中继工作之后, VTP 的信号数据就会通过中继链路接收和发送。中继上的 VTP 信号共有 3 种类型:

- 部分宣告——部分宣告是创建、删除或更改 VLAN 时产生的。
- 请求宣告——请求宣告是复位后的交换机或者在本地 VTP 域发生改变时产生, 例如 VTP 域名改变、交换机监听到配置更改号大于自身的 VTP 汇总宣告时都会产生请求宣告。
- 汇总宣告——交换机每 5 秒发送的一次汇总宣告信号。发送汇总宣告的主要目的是让交换机验证 VTP 的更改号, 以确保 VLAN 数据库的实时更新。如果发现更改号低, 交换机就会产生一个请求宣告来请求更新 VLAN 信息。

链路在正常工作。

例 2-22 用 show vtp status 命令查看 VTP 宣告的情况

```
Switch (enable) show vtp status
VTP statistics:
summary advts received      66
subset advts received       4
request advts received      1
summary advts transmitted   16
subset advts transmitted    13
request advts transmitted    0
No of config revision errors 0
No of config digest errors  0

VTP pruning statistics:

Trunk      Join Transmitted  Join Received  Summary advts received from
non-pruning-capable device
-----
2/1        1047              1045          0
2/2        1041              1046          0
2/20       631               635           0
Switch (enable)
```

show trunk 命令还会列出可修剪的 VLAN 信息。不要混淆可修剪的 VLAN 与 VLAN 的广播。可修剪是指对应某个 VLAN 来说，如果交换机上没有属于这个 VLAN 的在线端口，那么对应于这个 VLAN 的网络广播和用户数据就不会通过 Trunk 线路转发到这台交换机。默认情况下，全部 VLAN 的相关信息和生成树的数据帧都会通过所有的 Trunk 接口来传输。从一个中继删除部分 VLAN 和 STP 只能通过 **clear trunk** 命令。接下来的“第 4 步：STP 和 VLAN 广播的控制”一节中会详细讨论这些功能。

注释 前面的章节中提到过，VTP 信息只能从更改号比客户端 VTP 更改号大的 VTP 服务器发送到客户端 VTP。更改 VTP 服务器或者同步网络客户端 VTP 时要特别小心。网络同步之后，所有 VTP 的更改号都要匹配。改变 VTP 或 VLAN 时，其更改号会增加，因此更改的那台交换机有可能成为更改号最大的设备。这样，又将反过来将整个网络同步到和所更改的那台交换机 VLAN 数据库相同。

8. 第 3 步：在 Catalyst 2900XL/3500G 上配置 VLAN 中继

在该系列的交换机上配置中继有两个步骤，如 Catalyst 5500 系列：

第 1 步 将端口配置成中继模式。

第 2 步 设置 Trunk 封装使用 ISL 或 802.1q。

所有端口都默认设置成非中继模式。因此，第 1 步是将端口设置成中继模式。第 2 步则设置中继的封装形式。这些步骤是在接口配置模式下用下列命令来实现的：

```
(config if) #switchport mode trunk
(config if) #switchport trunk encapsulation {isl | dot1q}
```

例 2-23 是一个 ISL Trunk 的设置示意图。

例 2-23 设置 ISL 中继

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fastEthernet 0/19
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk encapsulation isl
Switch(config-if)#^Z
```

要确保中继正常工作，要记得检查链路两端的状态。**show interface interface_name switchport** 命令的输出结果可以给出中继链路的一些状态信息。这些信息和 Catalyst 4000/5500/6500 交换机使用 **show trunk** 命令得到的信息相似，包括中继链路的状态和封装形式，还包括默认 VLAN，链路上的工作 VLAN 及任何可修剪 VLAN 等信息。例 2-24 就列出了 **show interface interface_name switchport** 命令的输出结果。

例 2-24 Trunk 链路的状态

```
sw15#show int fastEthernet 0/19 switchport
Name: Fa0/19
Switchport: Enabled
Administrative mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: isl
Operational Trunking Encapsulation: isl
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1-5,33,100,200,800
Pruning VLANs Enabled: 2-1001

Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
sw15#
```

此外，还可以用 **show vtp counters** 命令来检查 VTP 的域计数器以及中继线路的情况，请参看例 2-25。这样有助于清楚地查看中继线路是否正常工作。

例 2-25 通过查看 VTP 计数器来确定中继线路的状态

```
sw15#show vtp counters
VTP statistics:
Summary advertisements received      : 10
Subset advertisements received       : 2
Request advertisements received      : 0
Summary advertisements transmitted   : 55
Subset advertisements transmitted    : 2
Request advertisements transmitted   : 12
Number of config revision errors     : 0
Number of config digest errors       : 0
Number of V1 summary errors          : 0

VTP pruning statistics:
```

(待续)

Trunk	Join Transmitted	Join Received	Summary advts received from non-pruning-capable device
Fa0/19	801	775	0
Fa0/20	1173	1164	0
Fa0/21			

在 802.1q 网络中，确保整个 VTP 域中默认 VLAN 的一致性非常关键，因为 802.1q 使用的是 MST，而 MST 使得整个 VTP 网络对任何第三方 802.1q 的交换机而言是一个单一桥接域。Cisco 通过采用 PVST+和 MST 协同工作的方式保证了其与 MST 域的兼容性。这实际上是 PVST+的扩展版，提供了与 802.1q 网络的透明无缝集成方式。默认 VLAN 都运用 MST，因此在整个 Internet 网络中保持默认 VLAN 的一致性非常重要。下面这条中继上的接口命令可以改变默认 VLAN：

```
(config-if) #switchport trunk native VLAN vlan -id
```

在 Catalyst 4000/5500/6500 系列交换机上已激活 802.1q 中继模式的端口上创建一个 VLAN，并将其作为该端口的默认 VLAN。

9. 第 4 步：STP 和 VLAN 数据广播的控制

配置 Catalyst 以太网交换机的最后一步是可选的，但对于大型网络来说非常重要。Cisco 设置了一些特性来使交换机可以在小型网络中实现即插即用，但这样做在大型网络中会带来不良反应，产生大量的数据量。PVST 的默认设置是每个 VLAN 的通信都通过中继端口，这样会导致边缘交换机因为处理过多的生成树请求以及其他网络广播而负荷过重。

例如，图 2-10 的网络中，交换机 crane 只带有一个 VLAN：VLAN 2。但由于该交换机和其他交换机处在同一 VTP 域中，因此也会参与 VLAN 3 和 VLAN 4 的生成树活动。对于该交换机来说，没有必要浪费资源处理根本就不在它上面的 VLAN 的生成树请求等信息。网络的规模和冗余度越大，这个问题就会变得越糟糕。例如，共有 50 台边缘交换机，那么每台交换机要在中继上接受和处理 50 个相互独立的生成树拓扑结构！在完成这些之前，交换机是无法传输用户数据。

通常认为 VLAN 修剪机制可以解决该 STP 的问题。但 VLAN 修剪机制影响的只是用户数据，一般是网络广播、多播以及大量单播数据。基本上，STP 创建数据可流通的路径，修剪机制则用于控制该路径上的网络广播数据。

Cisco 提供了两种有效的方法来处理多余的网络广播和 STP 问题：

- 从中继上清除 VLAN (Clearing VLANs from trunks) —— 从中继上清除 VLAN，这样中继端口就不会出现在这些 VLAN 的生成树拓扑结构里。下游的交换机不会再接收到关于从这些 VLAN 传来的 BPDU，这些 VLAN 里的用户数据也不会再通过这一中继广播。
- VLAN 修剪机制 (VLAN pruning) —— VLAN 修剪机制规定，允许 VTP 修剪机制时，如果下游交换机在允许修剪的那个 VLAN 上没有活动端口，该交换机就会禁止将关于该 VLAN 的数据转发给已修剪 VLAN 的下游交换机。VTP 修剪机制是一种数据量控制的方法，减少了不必要的网络广播、多播以及数目庞大的单播数据的广播。如果某个 VLAN 包含在可修剪列表中，VTP 修剪就会阻止不必要的的数据经过中继传出去。如果 VLAN 配置为不可修剪，那么其数据的传输将按正常的方式进行。

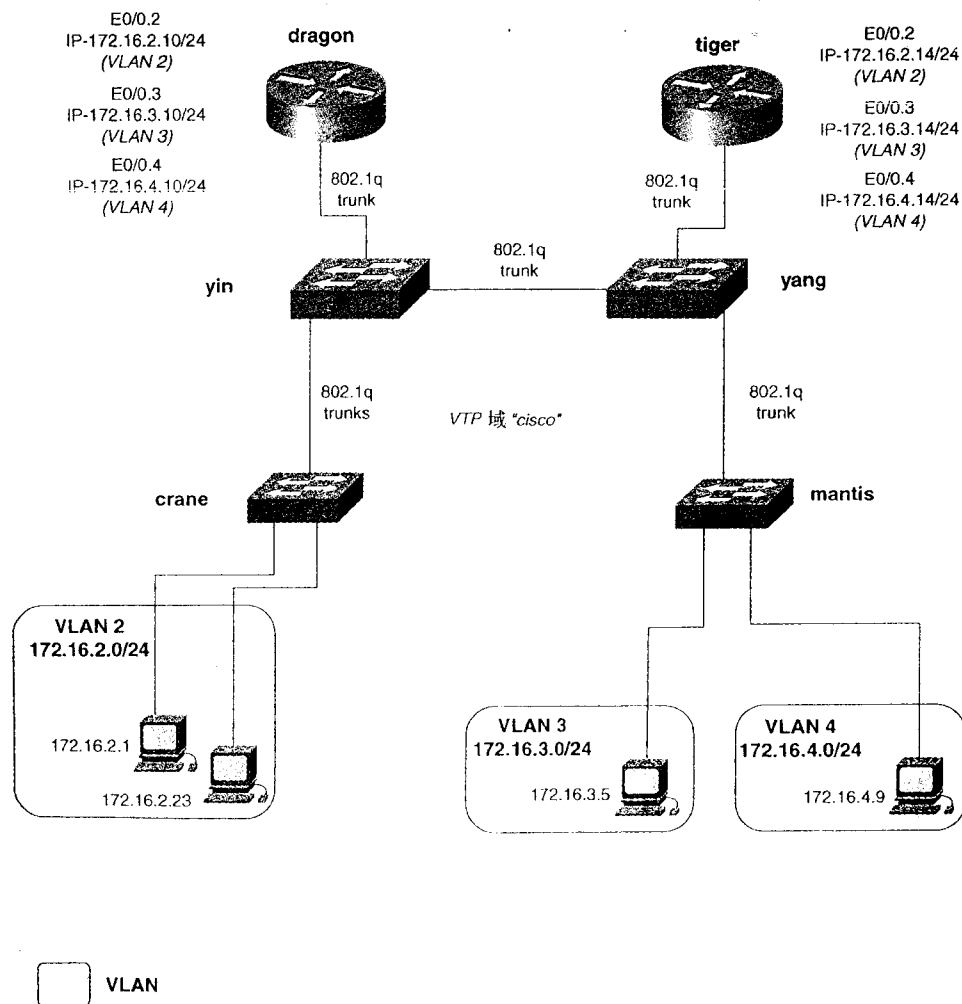


图 2-10 VLAN Trunking and STP

在图 2-10 中，VLAN 2 在交换机 crane, yin 和 yang 上，但不在交换机 mantis 上。如果 VLAN 2 允许修剪机制，交换机 mantis 就不会收到 VLAN 2 的网络广播、多播以及大量单播数据。同样，交换机 crane 也不会收到 VLAN 3 和 VLAN 4 的数据。

中继的清除是由“核心”交换机，也就是 VTP 服务器来完成的。除了下游交换机上的 VLAN 之外的所有 VLAN 都会被清除。Catalyst 新版本的软件允许清除 VLAN 1。然而，大部分交换机都禁止对 VLAN 1 的清除。在 Catalyst 4000/5500/6500 上，可以用下面的命令从中继链路上清除 VLAN：

```
Switch(enable) clear trunk [ mod_num/port_num] vlans_2 1001
```

用逗号作为分隔符或用连字符指定 VLAN 的范围，可以一次清除多个 VLAN。例如，要清除 VLAN3, VLAN 5 和 VLAN 10 到 150，可以使用下面的命令：

```
Switch(enable) clear trunk 2/1 2,5 10-150
```

要在 Catalyst 2900XL/3500G 交换机上的中继端口上清除 VLAN，可以使用下面的接口命令：

```
Switch (config-if) #switchport trunk allowed vlan [add | all | except | remove]
                        vlans_2-1001
```

这里，

- **add**——将以下 VLAN 加入中继线路。
- **all**——包括中继线路上所有的 VLAN。
- **except**——包括除了以下 VLAN 之外的所有 VLAN。
- **remove**——将以下 VLAN 从中继线路中清除。

例如，要清除 VLAN 3 到 VLAN 6，可以使用下面的命令：

```
Switch (config-if) #switchport trunk allowed vlan remove 3-6
```

图 2-11 是一个和图 2-10 同样的网络，只是接口名称发生了变化。本例中，交换机 yin 到交换机 crane 的中继线路上除了 1 和 2 以外的全部 VLAN 都会被清除。在清除中继之前，查看一下交换机 yin 上的不同 VLAN 的生成树的情况。例 2-26 列出了对 VLAN 3 使用 **show spanning tree** 命令的输出结果。该命令有助于理解和掌握交换网络中的生成树。下一节会讨论这个命令的详细用法。请注意例 2-26 中，VLAN 3 是运行在通往 dragon 路由器-yang 交换机和交换机 crane 的中继线路上。

例 2-26 生成树转发到所有中继上

```
yin#show spanning-tree vlan 3
```

```
Spanning tree 3 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0004.275e.f5c2
Configured hello time 2, max age 20, forward delay 15
Current root has priority 32768, address 0004.275e.f0c2
Root port is 67, cost of root path is 4
Topology change flag not set, detected flag not set, changes 1
Times: hold 1, topology change 35, notification 2
      hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0
```

```
Interface Fa0/10 (port 23) in Spanning tree 3 is FORWARDING
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f0c2
Designated bridge has priority 32768, address 0004.275e.f5c2
Designated port is 23, path cost 4
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 3766, received 0
```

```
Interface Fa0/19 (port 33) in Spanning tree 3 is FORWARDING ←Trunk to the crane
switch
```

```
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f0c2
Designated bridge has priority 32768, address 0004.275e.f5c2
Designated port is 33, path cost 4
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 3768, received 1
```

```
Interface Gi0/1 (port 67) in Spanning tree 3 is FORWARDING
Port path cost 4, Port priority 128
Designated root has priority 32768, address 0004.275e.f0c2
Designated bridge has priority 32768, address 0004.275e.f0c2
Designated port is 67, path cost 0
Timers: message age 2, forward delay 0, hold 0
BPDU: sent 5, received 3773
```

```
yin#
```

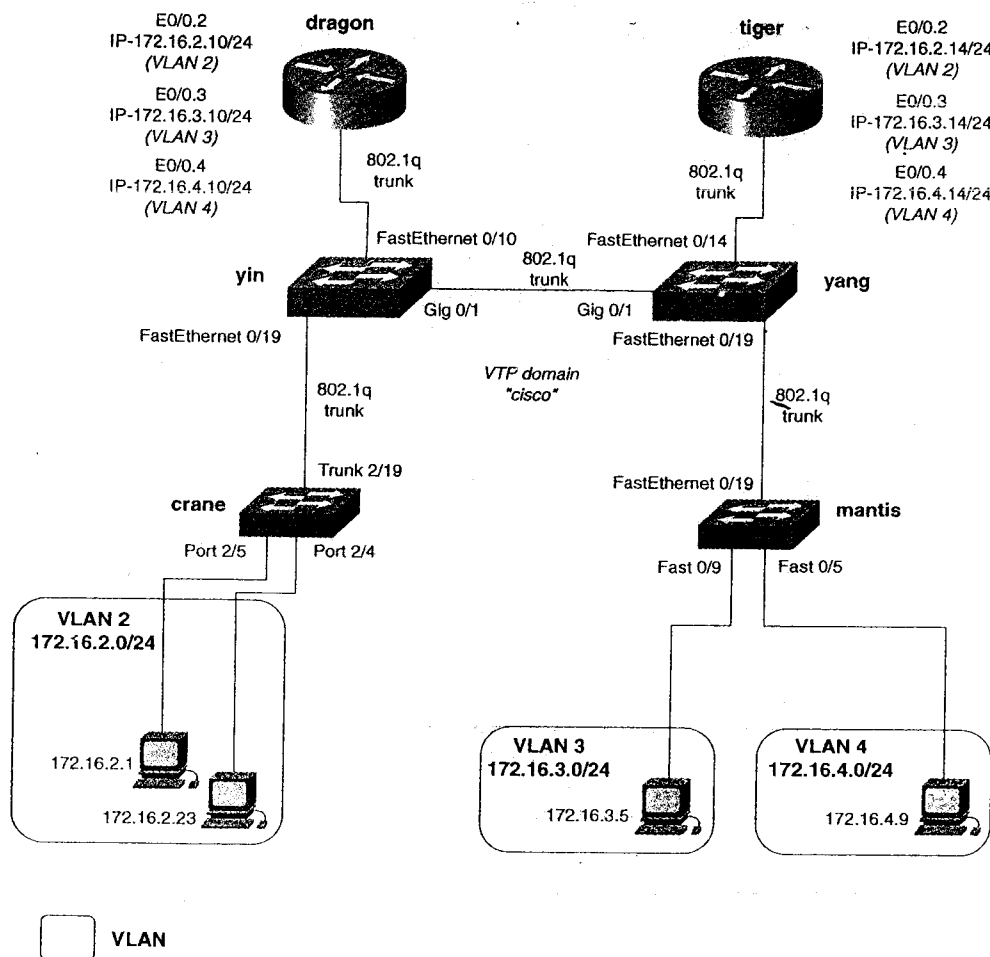


图 2-11 VLAN Trunking and STP

本例中，交换机 yin 是 Catalyst 3500G 型。因此，要使用 **switchport** 命令来清除中继。例 2-27 显示的是清除交换机 yin 和交换机 crane 之间的中继线路上的 VLAN 3 到 1001。该例表示的是 VLAN 3 的生成树的情况。请注意，图中的 VLAN 3 不再通过中继端口 Fa0/19 向交换机 crane 转发数据。

例 2-27 清除中继上的 VLAN

```
yin(config)#int fastEthernet 0/19
yin(config-if)#switchport trunk allowed vlan remove 3-1001
yin(config-if)#^Z
```

```
yin#show spanning-tree vlan 3
```

```
Spanning tree 3 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0004.275e.f5c2
```

(待续)

```

Configured hello time 2, max age 20, forward delay 15
Current root has priority 32768, address 0004.275e.f0c2
Root port is 67, cost of root path is 4
Topology change flag set, detected flag not set, changes 4
Times: hold 1, topology change 35, notification 2
      hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0

Interface Fa0/10 (port 23) in Spanning tree 3 is FORWARDING
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f0c2
Designated bridge has priority 32768, address 0004.275e.f5c2
Designated port is 23, path cost 4
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 4589, received 0

Interface Gi0/1 (port 67) in Spanning tree 3 is FORWARDING
Port path cost 4, Port priority 128
Designated root has priority 32768, address 0004.275e.f0c2
Designated bridge has priority 32768, address 0004.275e.f0c2
Designated port is 67, path cost 0
Timers: message age 3, forward delay 0, hold 0
BPDU: sent 14, received 4593

yin#

```

show interface interface_name switchport allowed vlan 命令也可以显示出中继上传输哪些 VLAN，该命令在 Catalyst 4000/5500/6500 交换机上的等价命令是 **show trunk**。例 2-28 列出 **switchport** 命令的输出结果表明在中继上，VLAN 3 到 1001 已经不存在。VLAN 1002 到 1005 不是以太 VLAN，因此不能从该中继上进行清除。

例 2-28 显示一条中继上允许通过的 VLAN

```

yin#show int fastEthernet 0/19 switchport allowed-vlan
"1,2,1002-1005"
yin#

```

清除中继线路是对 STP 进行控制的方法之一，但是对于那些需要一定冗余度的交换机来说，还得采用其他控制 STP 的方法。

注释 使用网络分析仪监视交换机端口

交换机并不是将所有数据帧都转发到 VLAN 中所有的端口。前面讲过，即使在同一 VLAN，交换机也会对其数据帧将要转发去的端口进行选择。因此，在使用网络分析仪监视某交换机的端口时，必须要输入一个特殊的命令：

```
set span { mod/src_ports } { dest_mod/dest_port_of_monitor } [rx | tx | both]
```

没有该命令，网络分析仪就不能正常捕获所要监视的 VLAN 里的信息。

配置 STP 根位置

冗余的交换网络不会进行任何形式的自动负载平衡。由于 STP 是根据静态的 MAC 地址来决定对数据转发或是拥塞的，因此所有数据对于所有的 VLAN 来说都会按照同一方向、同一路径进行传输。这就导致一些链路使用过于频繁，而另一些链路始终空闲。图 2-12 演示了会聚于单一交换机的网络。Yang 交换机是 VLAN 2、3、4 和 5 的 STP 的根。

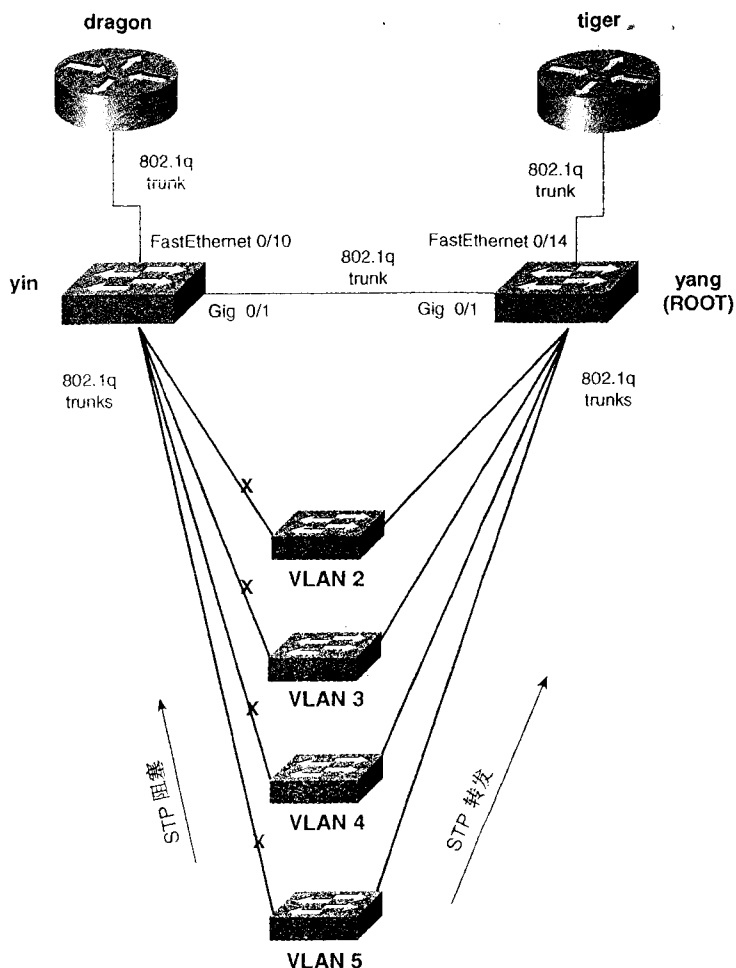


图 2-12 STP 根

如果想要平衡 yin 和 yang 交换机之间的负载,或者是在 dragon 和 tiger 路由器使用 HSRP,就可能想要控制 STP 根桥的位置。比如,如果 dragon 路由器是 VLAN 2 的主 HSRP,那就希望数据是通过交换机 yin 而不是 yang。要控制和分配交换网络中的数据流量,必须手动设置 STP 根桥的位置。

配置 Catalyst 交换机生成树的根桥位置有很多方法,使用方法取决于所要控制的网络环境。在设置根桥时,实际上是告诉 STP 哪些端口要置为阻塞模式而那些端口要置为转发模式。由于 STP 以 PVST 为基础,因此每个 VLAN 都应该有各自不同的根桥。这样就可以将数据通过那些空闲链路来传送。在图 2-13 中,交换机 yin 设置成 VLAN 4 和 5 的 STP 根,而 yang 交换机则是 VLAN 2 和 3 的 STP 根。这样,边缘交换机就可以更加平均地通过中继线路来平衡各自负载。VLAN 4 和 5 的数据会转发给 yin 而 VLAN 2 和 3 的则转发给 yang 交换机。

在进一步讨论如何设置 STP 根之前,必须知道如何确定根桥。在 Catalyst 4000/5500/6500 交换机上用于确定根桥位置的主要命令是 `show spantree vlan`。生成树的工作是以 PVST 为基础的,所以加上一个 VLAN ID 的参数。例 2-29 列出了 `show spantree` 命令的输出结果。

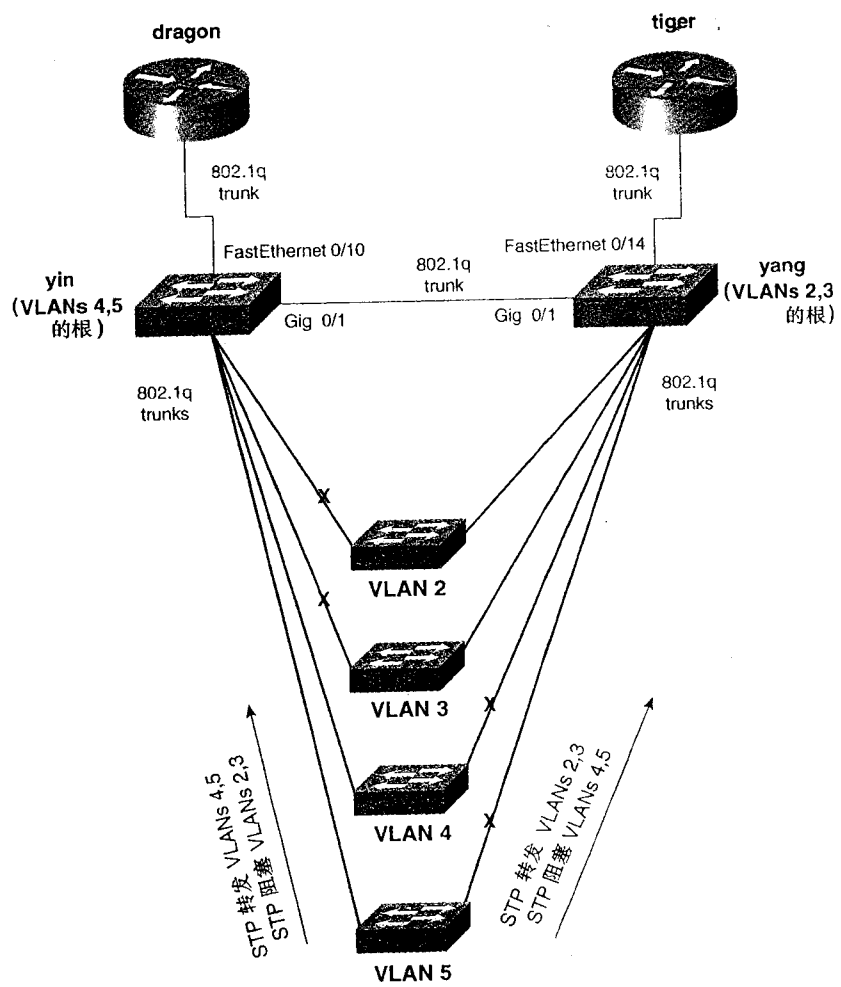


图 2-13 STP 根

例 2-29 查看 VLAN 2 的生成树

```
crane (enable) show spantree 2
VLAN 2
Spanning tree enabled
Spanning tree type          ieee

Designated Root             00-30-19-76-4d-01
Designated Root Priority     88
Designated Root Cost        0
Designated Root Port        1/0
Root Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Bridge ID MAC ADDR          00-30-19-76-4d-01
Bridge ID Priority           88
Bridge Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Port      Vlan  Port-State    Cost    Priority  Fast-Start  Group-Method
-----
```

(待续)

2/4	2	forwarding	100	32	disabled
2/19	2	forwarding	19	32	disabled
2/20	2	forwarding	19	32	disabled
crane (enable)					

该命令所提供的信息中的每一项都是有价值的，其含义如下（前面关于 STP 的一节里有更为详细的说明）：

- **Spanning Tree Type**——所用的生成树协议的类型：IBM、DEC、或 IEEE。
- **Designated Root**——根桥的 MAC 地址。
- **Designated Root Priority**——从根桥接收到的桥优先级，其取值范围是 0 到 65 535，默认值是 27 768。
- **Designated Root Cost**——到根桥的代价的累积。
- **Designated Root Port**——该段的 DR 根端口。
- **Root Max Age、Hello Time、Forward Delay**——根桥设置的 3 种 STP 定时器的值。
- **Bridge ID MAC ADDR**——本地桥在这一 VLAN 中所用的 MAC 地址标识。
- **Bridge ID Priority**——本地桥的优先级。
- **Root Max Age、Hello Time、Forward Delay**——本地桥的 3 种 STP 定时器的值。

最后一栏列出的是 VLAN 中参与 STP 活动的所有端口，还显示了端口是处于转发模式还是阻塞模式以及端口代价和服务优先级。不要混淆端口优先级和生成树桥的优先级。端口优先级取值范围是 0 到 63（从高到低），默认值是 32。

在 Catalyst 2900XL/3500G 交换机上可以用以下命令查看生成树的情况：

```
Switch#show spanning tree vlan vlan
```

例 2-26 为该命令的使用实例。

显示生成树工作状况的另一条命令是 **show spantree summary**，该命令给出 VLAN 的基本状况以及显示端口的数目和状态。例 2-30 是该命令的使用情况。

例 2-30 查看 VLAN 2 的生成树情况

```
Switch (enable) show spantree summary
Summary of connected spanning tree ports by vlan

Uplinkfast disabled for bridge.
Backbonefast disabled for bridge.

Vlan  Blocking Listening Learning Forwarding STP Active
-----
1      1          0          0          1          2
2      0          0          0          3          3
3      0          0          0          2          2
4      0          0          0          2          2
100    0          0          0          2          2
200    0          0          0          2          2
300    0          0          0          2          2

Blocking Listening Learning Forwarding STP Active
-----
Total  1          0          0          14         15
Switch (enable)
```


回想一下 STP 确定其 STP 桥以及转发端口、阻塞端口时所遵循步骤和过程对于正确设置 STP 根很有帮助。

这一过程的四步骤如下:

- 1 最小的根 BID <优先级加上 MAC 地址>。
- 2 到根桥的最小代价，到根桥的路径的累加代价为最小。
- 3 最小的数据发送者 BID。
- 4 最小的端口 ID。

在 Catalyst 4000/5500/6500 系列交换机上，有四种方法（命令）来控制 STP 根的选择:

- **set spantree root**
- **set spantree priority**
- **set spantree portvlancost**
- **set spantree portvlanpri**

下面将对这些命令进行详细讲解。

set spantree root 命令

命令的句法格式如下: **set spantree root [secondary] [vlan_list] [dia network_diameter] [hello hello_time]**

这是一条功能强大的宏命令，可以调节生成树的定时器的值，而且使本地的桥/交换机被选择成为根桥/根交换机。该命令通常一次有效。如果网络又加入新交换机，必须再执行一次。Catalyst 交换机通过检查现有根桥的 BPDU 来实现这一功能。如果发现优先级的值大于 8192，该宏命令会将本地桥的优先级设为 8192。如果发现 BPDU 中有优先级小于 8192 的桥，该宏命令就会把本地桥的优先级设为比该值小 1 的数值。例如，根桥往新加入的交换机发送配置 BPDU。执行了该宏命令的新交换机检查 BPDU 的优先级并发现其值为 89。这时，宏命令就将本地桥的优先级调整为 88，并选为新的根桥。参数 secondary 是将链桥的本地优先级设置为 16,384。由于默认桥的优先级为 32,768，因此 16,384 的优先级就可以使其桥作为备份根桥。参数 Diameter 和 Hello timers 用于调整 STP 参数 hello 和最大失效延时 (max age delay)。调节定时器值时一定要谨慎。该命令只在 Catalyst 4000/5500/6500 型的交换机上才可用。

set spantree priority 命令

该命令的句法格式如下: **set spantree priority [bridge priority] [vlans]**

该命令可以直接改变桥的优先级。由于优先级是根选择时最重要的因素，所以这条命令主要是直接用来进行根的选择。桥优先级范围是 0 到 65535，有效数值有: 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768 (默认), 36864, 40960, 45056, 49152, 53248, 57344 和 61440，其中，0 优先级最高而 61440 的优先级最低。

set spantree portvlancost 命令

该命令的句法格式为: **set spantree portvlancost mod_num/port_num [cost 1 65535] [vlans]**

这条命令可以改变生成树宣告到下游邻居的代价。STP 在考虑所有可能的路径以便确定哪一条为最佳路径时也会用到这一代价。注意，低开销的路径最佳。从表 2-5 中可以看到默认链路开销值的完整列表。

set spantree portvlanpri 命令

该命令的句法格式为: **set spantree portvlanpri mod_num/port_num [priority 0-63] [vlans]**

该命令可以设置发往下游相邻交换机的端口优先级。该命令可以按每 VLAN、每端口执行，因此非常有用，主要用于交换机网络中负载共享的情况。端口优先级的取值范围是 0 到 63，默认为 32，0 的优先级最高，63 最低。

表 2-14 列出了这些命令以及在 STP 工作过程中的作用。处在表中越高位置的命令在根选择过程中的作用就越大。

表 2-14

以太 DTP 配置结果

	Catalyst 4k/55k/65k set 命令	Catalyst 2900XL/3500G 全局配置命令
1 Lowest Root BID	set spantree priority set spantree root macro	spanning tree [vlan vlan_id] [priority 0-65535]
2 Lowest Path Cost to Root	set spantree portvncost	spanning tree [vlan vlan_id] [cost 1-65535]
3 Lowest Sender BID	set spantree priority	spanning tree [vlan vlan_id]
4 Lowest Port ID	set spantree portvlanpri	spanning tree [vlan vlan_id] [port priority 0-255]

10. 实例：配置路由交换网

现在，将这些概念用于几个实例中。图 2-14 是一个常见的包括交换机和路由器的网络。网络中有两个激活的 VLAN，分别是 VLAN 2 和 VLAN 4。VLAN 2 是管理 VLAN，里面有一些用户数据，含有一个 IP 子网 172.16.2.0/24。VLAN 4 则是一个纯用户 VLAN，含有一个 IP 子网 172.16.4.0/24。为 VLAN 之间提供路由的路由器 dragon 是交换机默认的网关。在这个例子里，可以作如下设置：

- 基本的 IP 管理，将 172.16.2.10 作为 IP 默认网关。
- 路由器 dragon 和交换机 yin 之间的 ISL 中继，路由器 dragon 用 EIGRP 为路由选择协议在 VLAN 之间进行路由。
- 交换机 yin、crane 和 mantis 之间的 802.1q 中继。
- 如图 2-14 所示的适当的 VLAN 设置。

该网络中有两种交换机，因此我们可以分别演示适用于两种系列交换机的各种命令。从交换机 yin 开始，需要为其设置 IP 地址和默认网关、VLAN 中继。VLAN 2 和 4 也是这样。回想一下配置以太网交换机的四个步骤：

第 1 步 配置交换机管理。

第 2 步 配置 VTP 和 VLAN。

第 3 步 配置 VLAN 中继（如果需要）。

第 4 步 （可选）对 STP 和 VLAN 的数据传播进行控制。

将 VLAN 2 设置为管理 VLAN，先在路由器 yin 和 mantis 上定义虚拟接口 VLAN 2，为该接口分配 IP 地址，并通过关闭虚拟接口 VLAN 1 来启动 VLAN 2 工作。例 2-31 演示了如何在交换机 yin 上进行设置。

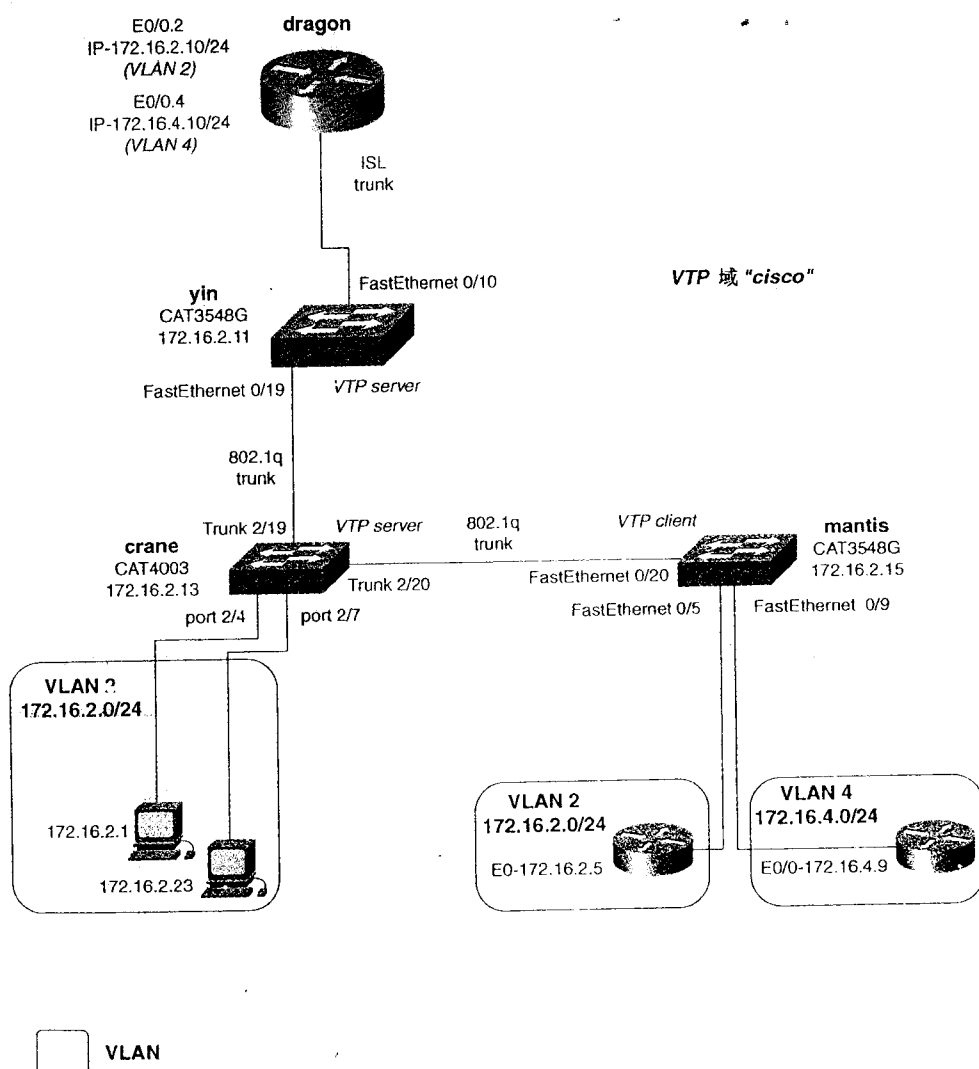


图 2-14 交换路由网络

例 2-31 基本管理设置

```

Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname yin          ←Sets hostname
yin(config)#int vlan 1               ←Disable interface VLAN 1
yin(config-if)#shut
yin(config-if)#exit

yin(config)#int vlan 2               ←Define interface VLAN 2
yin(config-subif)#ip address 172.16.2.11 255.255.255.0
yin(config-subif)#exit

yin(config)#ip default-gateway 172.16.2.10 ←IP default gateway
yin(config)#

```

交换机 mantis 上的管理部分的配置和例 2-31 中的方法类似。

例 2-32 是如何在 Catalyst 4003 交换机 crane 上配置基本管理功能的例子。该系列的交换机要求在将管理 VLAN 从 VLAN 1 改到其他 VLAN 去之前先定义 VLAN。因此要先进行第 2 个步骤的操作。第 2 步包括 VLAN 和 VTP 的定义。

例 2-32 配置基本的管理功能以及默认 VLAN

```
Console> (enable) set prompt crane          ←sets host name

crane (enable) set vtp domain cisco          ←Set VTP domain
VTP domain cisco modified
crane (enable) set vlan 2 name management    ←set VLAN 2 and name it
Vlan 2 configuration successful
crane (enable)

crane (enable) set int sc0 2 172.16.2.13 255.255.255.0 ←MNGT interface
Interface sc0 vlan set, IP address and netmask set.
crane (enable) set ip route 0.0.0.0 172.16.2.10      ←Default route to dragon
Route added.
crane (enable)
```

第 2 步需要配置 VTP 域（本例中是 cisco）以及在 VTP 服务器上定义 VLAN。交换机 mantis 只是一台 VTP 客户端，因此可以从这台交换机开始。例 2-33 为如何设置 VTP 域以及将交换机 mantis 的 VTP 模式改为客户端模式。

例 2-33 设置 VTP 域以及 VTP 客户端模式

```
mantis#vlan database          ←enter VLAN database
mantis(vlan)#vtp domain cisco ←Set VTP domain name to cisco
Changing VTP domain name from Null to cisco
mantis(vlan)#vtp client       ←Set VTP client mode
Setting device to VTP CLIENT mode.
mantis(vlan)#
```

创建 VTP 域后，可以将给交换机的端口分配给 VLAN。例 2-34 说明了交换机 mantis 的配置，将用户端口分配给了 VLAN 2 和 VLAN 4。

例 2-34 在交换机 mantis 上将端口分配给 VLAN

```
mantis#conf t
Enter configuration commands, one per line. End with CNTL/Z.
mantis(config)#interface fastEthernet 0/9
mantis(config-if)#switchport mode access          ←set port to a single VLAN
mantis(config-if)#switchport access vlan 4        ←set VLAN id
mantis(config-if)#exit
mantis(config)#interface fastEthernet 0/5
mantis(config-if)#switchport mode access
mantis(config-if)#switchport access vlan 2
mantis(config-if)#^Z
mantis#
```

类似地，交换机 crane 上的端口 2/4 到 2/7 也需要分配到 VLAN 2 中，见例 2-35。

例 2-35 在交换机 crane 上将端口分配给 VLAN

```
crane (enable) set vlan 2 2/4,2/7
VLAN 2 modified.
VLAN 1 modified.
VLAN Mod/Ports
-----
2      2/4,2/7,2/19-20

crane (enable) 2001 Jun 26 21:15:08 %PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
```

例 2-36 是交换机 crane 上的 VTP 域和 VLAN 的设置过程，其后半部分则是交换机 yin 上类似的设置过程。

例 2-36 VTP 域和 VLAN 的设置

```
crane (enable) set vlan 4

yin(vlan)#vtp domain cisco
Changing VTP domain name from Null to cisco
yin(vlan)#vlan 2 name management
VLAN 2 added:
    Name: management
yin(vlan)#
yin(vlan)#vlan 4
VLAN 4 added:
    Name: VLAN0004
yin(vlan)#
```

第3步要求对交换机之间的 VLAN 中继链路进行配置。前面讲过，配置静态的中继要比去记住庞大的自动识别表容易和快捷。例 2-37 是在交换机 yin 上配置 ISL 和 802.1q 的例子。

例 2-37 配置 ISL 的 802.1q 中继

```
yin(config)#interface fast 0/10
yin(config-if)#switchport mode trunk          ←Set port to trunk
yin(config-if)#switchport trunk encapsulation isl ←Set encapsulation to ISL
yin(config-if)#exit
yin(config)#interface fast 0/19
yin(config-if)#switchport mode trunk
yin(config-if)#switchport trunk encapsulation dot1q ←Set encapsulation to 802.1q
yin(config-if)#^Z
```

例 2-38 在交换机 crane 上进行了中继端口的设置。

例 2-38 802.1q 中继端口的设置

```
crane (enable) set trunk 2/19 on          ←Set port 2/19 to trunk
Port(s) 2/19 trunk mode set to on.
crane (enable) set trunk 2/19 dot1q      ←Set trunk type
Port(s) 2/19 trunk type set to dot1q.
2001 Jun 26 17:54:23 %DTP-5-TRUNKPORTON:Port 2/19 has
become dot1q trunk

crane (enable) set trunk 2/20 on
Port(s) 2/20 trunk mode set to on.
crane (enable) set trunk 2/20 dot1q
```

(待续)

```
Port(s) 2/20 trunk type set to dot1q.  
crane (enable)
```

设置完中继之后，可以在 VTP 客户端交换机 mantis 查看 VTP 域的状态。例 2-39 显示了通过中继传输的 VTP 信息。命令 **show vlan** 的输出显示交换机检测到新加入的 VLAN 信息。

例 2-39 交换机 mantis 上的 VTP 信息

```
mantis#show vtp status  
VTP Version           : 2  
Configuration Revision : 7  
Maximum VLANs supported locally : 254  
Number of existing VLANs : 7  
VTP Operating Mode     : Client  
VTP Domain Name        : cisco  
VTP Pruning Mode       : Disabled  
VTP V2 Mode            : Disabled  
VTP Traps Generation   : Disabled  
MD5 digest             : 0x51 0x0C 0x00 0x9A 0x0B 0x13 0xE3 0xBA  
Configuration last modified by 172.16.2.13 at 6-26-01 20:39:23 ←VTP is receiving!  
mantis#  
mantis#show vlan  
VLAN Name                Status      Ports  
-----  
1    default              active      Fa0/1, Fa0/2, Fa0/3, Fa0/4,  
                                           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, Fa0/21, Fa0/22,  
                                           Fa0/23, Fa0/24  
2    management           active      Fa0/5  
4    VLAN0004             active  
1002 fddi-default         active  
1003 token-ring-default   active  
1004 fddinet-default      active  
1005 trnet-default        active  
  
VLAN Type  SAID      MTU    Parent RingNo BridgeNo Stp   BrdgMode Trans1 Trans2  
-----  
1    enet  100001   1500   -     -     -     -     -     0     0  
2    enet  100002   1500   -     -     -     -     -     0     0  
4    enet  100004   1500   -     -     -     -     -     0     0  
1002 fddi  101002   1500   -     0     -     -     -     0     0  
1003 tr   101003   1500   -     0     -     -     srb   0     0  
1004 fdnet 101004   1500   -     -     -     ieee  -     0     0  
1005 trnet 101005   1500   -     -     -     ibm   -     0     0  
mantis#
```

11. 在路由器上设置中继链路

要想在 VLAN 之间进行路由，每个 VLAN 都需要路由器接口。由于大型网络中需要的物理接口的数目较多，可以在路由器上设置 ISL 或 802.1q 的中继端口。Cisco 支持在工作速率至少为百兆的以太网路由器接口上设置 VLAN 中继链路。

设置中继与设置帧中继子接口非常相似。设置 VLAN 的中继时，必须为每个需要路由的 VLAN 创建逻辑以太网子接口，分配封装形式，最后，要配置路由选择协议进行路由。逻辑子接口对于路由选择协议来说，类似物理接口。以上过程可以用下面的命令来完成：

```
Router (config) interface FastEthernet0.x
```

Router (config-if) encapsulation [dot1Q [native native_vlan_id | isl] [vlan_id]
继续上面的例子, 在例 2-40 里可以看到在路由器 dragon 上设置 VLAN 中继端口的过程。

例 2-40 路由器 mantis 上的 VTP 域

```
dragon(config)#int fastEthernet 0/0.2
dragon(config-subif)#encapsulation isl 2      ←Set encapsulation and VLAN
dragon(config-subif)#ip address 172.16.2.10 255.255.255.0
dragon(config-subif)#exit
dragon(config)#int fastEthernet 0/0.4
dragon(config-subif)#encapsulation isl 4
dragon(config-subif)#ip address 172.16.4.10 255.255.255.0
dragon(config-subif)#exit

dragon(config)#router eigrp 2001              ←Configuring EIGRP
dragon(config-router)#network 172.16.0.0
dragon(config-router)#no auto-summary
```

完成这些配置后, 整个域中的 IP 连接就完成了。所有的交换机、路由器以及主机都可以相互 ping 通。

12. 中继间的负载平衡

上面的模型中没有冗余的中继, 因此生成树不会存在任何问题。现在把模型做少许的改动, 在网络中加入一些生成树的问题。

图 2-15 为修改后的新网络。

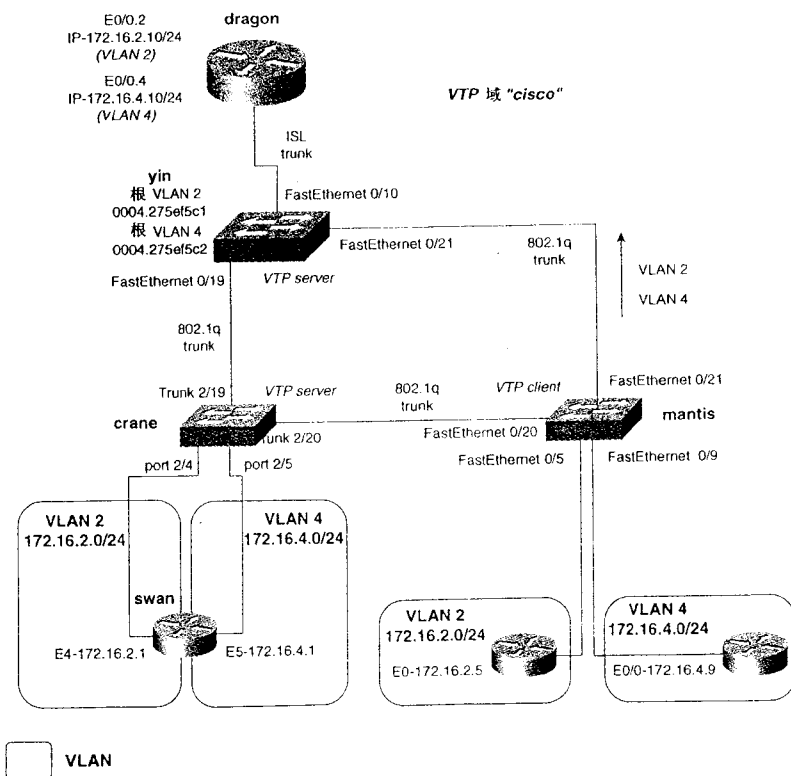


图 2-15 交换网络中的负载平衡

在交换机 mantis 和 yin 之间加入另一条中继线路，从而在网络中产生环路。这时，STP 会将一部分端口设置为阻塞模式，使得网络中没有环路。加入一台叫做 swan 的双端口路由器，它在 VLAN 2 和 4 中各有一个端口，这样就有必要在该网络中进行负载平衡。在一个实际的网络环境中，可能有 HSRP 运行于路由器 swan 和 dragon 之间。

网络如何收敛，哪条路径进入转发模式或阻塞模式，这些都是不可预料的。通常（但不是绝对），数据在整个网络中倾向于按同一路径进行传输。这样，某些链路会负担所有的数据量，而另外一些链路实际上是没有流量。

图 2-15 中，交换机 yin 和 mantis 之间配置了另一条 802.1q 中继，并且加入相应的路由器。从交换机 mantis 上的 VLAN1, 2 和 4 看 STP 会发现，所有的数据都通过这条新中继链路（Fast 0/21 到根的中继）来传输。在 VTP 域中，交换机 yin 是所有的 VLAN STP 的根。例 2-41 是 `show spanning tree vlan` 命令在交换机 mantis 上的使用情况。

例 2-41 在交换机 mantis 上使用 `show spanning tree` 命令

```
mantis#show spanning-tree vlan 2

Spanning tree 2 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 00d0.976c.b781
Configured hello time 2, max age 20, forward delay 15
Current root has priority 32768, address 0004.275e.f5c1 ←Root MAC for VLAN 2
Root port is 35, cost of root path is 19
Topology change flag not set, detected flag not set, changes 7
Times: hold 1, topology change 35, notification 2
hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0

Interface Fa0/5 (port 17) in Spanning tree 2 is FORWARDING
Port path cost 100, Port priority 128
Designated root has priority 32768, address 0004.275e.f5c1
Designated bridge has priority 32768, address 00d0.976c.b781
Designated port is 17, path cost 19
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 3066, received 0

Interface Fa0/20 (port 34) in Spanning tree 2 is BLOCKING ←Blocking
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f5c1
Designated bridge has priority 32768, address 0030.1976.4d01
Designated port is 84, path cost 19
Timers: message age 3, forward delay 0, hold 0
BPDU: sent 93, received 2972

Interface Fa0/21 (port 35) in Spanning tree 2 is FORWARDING
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f5c1
Designated bridge has priority 32768, address 0004.275e.f5c1
Designated port is 35, path cost 0
Timers: message age 3, forward delay 0, hold 0
BPDU: sent 5, received 495

mantis#show spanning-tree vlan 4

Spanning tree 4 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 00d0.976c.b782
Configured hello time 2, max age 20, forward delay 15
```

(待续)


```

Current root has priority 32768, address 0004.275e.f5c2 ← Root MAC for VLAN 4
Root port is 35, cost of root path is 19
Topology change flag not set, detected flag not set, changes 5
Times: hold 1, topology change 35, notification 2
      hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0

Interface Fa0/9 (port 22) in Spanning tree 4 is FORWARDING
Port path cost 100, Port priority 128
Designated root has priority 32768, address 0004.275e.f5c2
Designated bridge has priority 32768, address 00d0.976c.b782
Designated port is 22, path cost 19
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 1967, received 0

Interface Fa0/20 (port 34) in Spanning tree 4 is BLOCKING
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f5c2
Designated bridge has priority 32768, address 0030.1976.4d03
Designated port is 84, path cost 19
Timers: message age 2, forward delay 0, hold 0
BPDU: sent 1, received 2972

Interface Fa0/21 (port 35) in Spanning tree 4 is FORWARDING
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f5c2
Designated bridge has priority 32768, address 0004.275e.f5c2
Designated port is 35, path cost 0
Timers: message age 2, forward delay 0, hold 0
BPDU: sent 5, received 498
    
```

在该模型中，我们希望在交换机 mantis 的中继上进行负载平衡。来自 VLAN 2 的所有数据都传送到路由器 swan，而 VLAN 4 里的所有数据则默认传到路由器 dragon。要做到这一点，应将 VLAN 2 的根设置到交换机 crane，而 VLAN 4 的根则设成交换机 yin。

宏命令 **set root** 可以将 VLAN 2 的根设到交换机 crane 上，见例 2-42。用 **show spant 2** 命令观察 VLAN 2 的以前和当前的根桥。

例 2-42 为 VLAN 2 设置根桥

```

crane (enable) show spant 2
VLAN 2
Spanning tree enabled
Spanning tree type          ieee

Designated Root              00-04-27-5e-f5-c1 ←Current Root, same as in example 2-40
Designated Root Priority      32768
Designated Root Cost          19
Designated Root Port         2/19
Root Max Age 20 sec          Hello Time 2 sec   Forward Delay 15 sec

Bridge ID MAC ADDR            00-30-19-76-4d-01 ←Our MAC for VLAN 2
Bridge ID Priority             32768
Bridge Max Age 20 sec          Hello Time 2 sec   Forward Delay 15 sec

Port      Vlan  Port-State    Cost    Priority  Fast-Start  Group-Method
-----
2/4       2      forwarding    100     32       disabled
    
```

(待续)

```

2/7      2      not-connected    100      32      disabled
2/19     2      forwarding       19       32      disabled
2/20     2      forwarding       19       32      disabled
crane (enable)

crane (enable) set spant root 2      ←Set Root macro for VLAN 2
VLAN 2 bridge priority set to 8192.
VLAN 2 bridge max aging time set to 20.
VLAN 2 bridge hello time set to 2.
VLAN 2 bridge forward delay set to 15.
Switch is now the root switch for active VLAN 2.
crane (enable)
crane (enable) show spant 2
VLAN 2
Spanning tree enabled
Spanning tree type                ieee

Designated Root      00-30-19-76-4d-01    ←We are now the Root for VLAN 2
Designated Root Priority 8192
Designated Root Cost 0
Designated Root Port 1/0
Root Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Bridge ID MAC ADDR    00-30-19-76-4d-01
Bridge ID Priority 8192
Bridge Max Age 20 sec  Hello Time 2 sec  Forward Delay 15 sec

Port      Vlan  Port-State    Cost    Priority  Fast-Start  Group-Method
-----
2/4       2      forwarding    100     32      disabled
2/7       2      not-connected 100     32      disabled
2/19      2      forwarding    19      32      disabled
2/20      2      forwarding    19      32      disabled
crane (enable)

```

在交换机 mantis 上检查 VTP 的状况来确认该设置，如例 2-43 所示。交换机 mantis 现在显示 VLAN 2 的根桥是 0030.1976.4d01，即交换机 crane。现在，接口 Fast 0/20 转发 VLAN 2 的数据，而接口 Fast 0/21 则进入拥塞模式。VLAN 4 的根桥依然是交换机 jin。

例 2-43 STP 的负载平衡

```

mantis#show spanning-tree vlan 2

Spanning tree 2 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 00d0.976c.b781
Configured hello time 2, max age 20, forward delay 15
Current root has priority 8192, address 0030.1976.4d01    ←new Root bridge
Root port is 34, cost of root path is 19
Topology change flag not set, detected flag not set, changes 8
Times: hold 1, topology change 35, notification 2
      hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0

Interface Fa0/5 (port 17) in Spanning tree 2 is FORWARDING
Port path cost 100, Port priority 128
Designated root has priority 8192, address 0030.1976.4d01
Designated bridge has priority 32768, address 00d0.976c.b781
Designated port is 17, path cost 19
Timers: message age 0, forward delay 0, hold 0

```

(待续)

```

BPDU: sent 4073, received 0

Interface Fa0/20 (port 34) in Spanning tree 2 is FORWARDING
  Port path cost 19, Port priority 128
  Designated root has priority 8192, address 0030.1976.4d01
  Designated bridge has priority 8192, address 0030.1976.4d01
  Designated port is 84, path cost 0
  Timers: message age 2, forward delay 0, hold 0
  BPDU: sent 95, received 3977

Interface Fa0/21 (port 35) in Spanning tree 2 is BLOCKING
  Port path cost 19, Port priority 128
  Designated root has priority 8192, address 0030.1976.4d01
  Designated bridge has priority 32768, address 0004.275e.f5c1
  Designated port is 35, path cost 19
  Timers: message age 3, forward delay 0, hold 0
  BPDU: sent 6, received 1502
mantis#

mantis#show spanning-tree vlan 4

Spanning tree 4 is executing the IEEE compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 00d0.976c.b782
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 32768, address 0004.275e.f5c2
  Root port is 35, cost of root path is 19
  Topology change flag not set, detected flag not set, changes 5
  Times: hold 1, topology change 35, notification 2
         hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0

Interface Fa0/9 (port 22) in Spanning tree 4 is FORWARDING
  Port path cost 100, Port priority 128
  Designated root has priority 32768, address 0004.275e.f5c2
  Designated bridge has priority 32768, address 00d0.976c.b782
  Designated port is 22, path cost 19
  Timers: message age 0, forward delay 0, hold 1
  BPDU: sent 3441, received 0

Interface Fa0/20 (port 34) in Spanning tree 4 is BLOCKING
  Port path cost 19, Port priority 128
  Designated root has priority 32768, address 0004.275e.f5c2
  Designated bridge has priority 32768, address 0030.1976.4d03
  Designated port is 84, path cost 19
  Timers: message age 5, forward delay 0, hold 0
  BPDU: sent 1, received 4445

Interface Fa0/21 (port 35) in Spanning tree 4 is FORWARDING
  Port path cost 19, Port priority 128
  Designated root has priority 32768, address 0004.275e.f5c2
  Designated bridge has priority 32768, address 0004.275e.f5c2
  Designated port is 35, path cost 0
  Timers: message age 3, forward delay 0, hold 0
  BPDU: sent 5, received 1972
mantis#
    
```

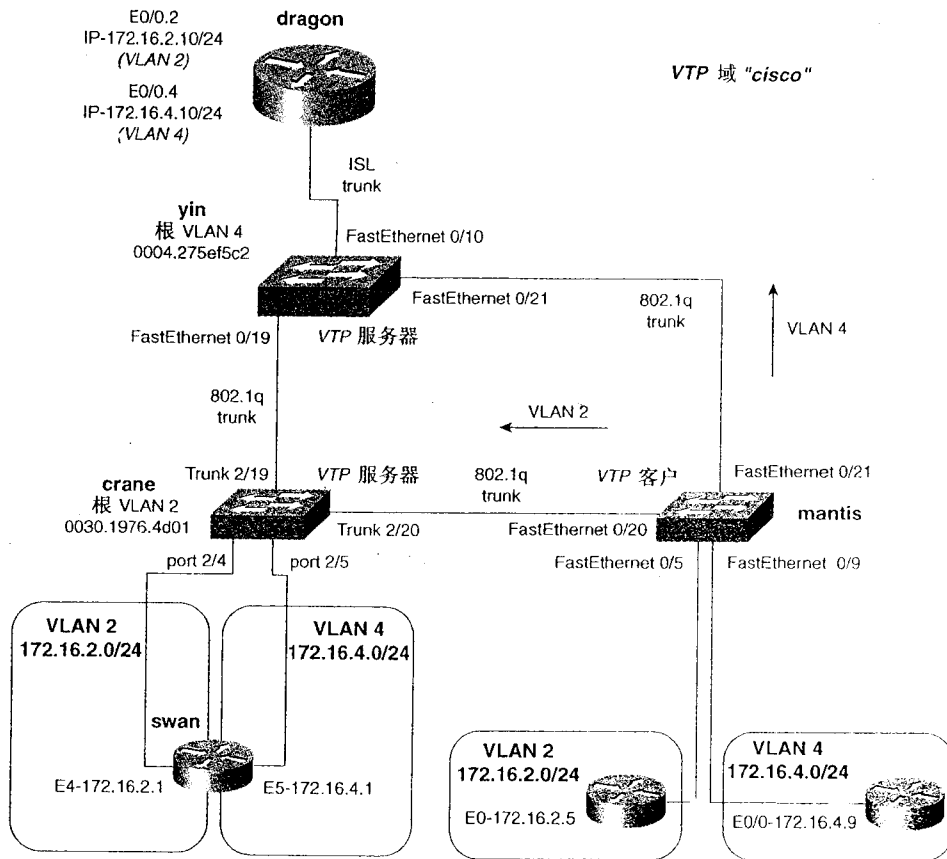
图 2-16 说明了网络如何处理和转发 VLAN 数据。

为加强整个网络生成树的一致性，还应该设置 VLAN 4 在交换机 yin 上的优先级，可以用 **priority** 命令来完成，如例 2-44 所示。

例 2-44 在 Catalyst 2900XL/3500G 交换机上设置根桥

```
yin(config)#spanning-tree vlan 4 priority 100
```

通过在交换机 mantis 上查看 VLAN 4 的情况，可以检验所作修改的效果，如例 2-45 所示。



☐ VLAN

图 2-16 中继上的负载平衡

例 2-45 验证在 VLAN 4 中的优先级为 100

```
mantis#show spanning-tree vlan 4
```

```
Spanning tree 4 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 00d0.976c.b782
Configured hello time 2, max age 20, forward delay 15
Current root has priority 100, address 0004.275e.f5c2
Root port is 35, cost of root path is 19
Topology change flag not set, detected flag not set, changes 5
Times: hold 1, topology change 35, notification 2
```

(待续)

```
hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0
```

以太网的发展非常迅速，还有许多有趣和有用的技术无法在这里一一讲解。要全面的掌握这方面的知识，强烈建议读者参考 Kennedy Clark 和 Kevin Hamilton 的著作 *Cisco LAN Switching*，该书详细讲述了 LAN 交换的技术，可以作为扩充知识的参考书籍。以下是一些读者可能感兴趣但在本书中没有涉及的内容：

- **UplinkFast、PortFast、BackboneFast**——用来帮助 STP 初始化或故障时可以处理用户数据的技术。有助于 STP 在其收敛时避免数据的丢失。其设置非常简单。
- **快速以太网通道/吉比特以太网通道 (Fast EtherChannel/Gigabit EtherChannel)**——以太网通道使得路由器可以将 4 个快速以太网端口捆绑，用来传送数据。该技术同样适用于吉比特以太网。通常把以太网通道想像成以太网的多 PPP 链路。以太网通道把捆绑后的链路当作一个大的物理链路，通过不同的方法在该捆绑链路上分发数据。在全双工模式下，以太网通道中的数据传输率可以达到 800 兆到 8 000 兆。以太网通道有助于避免 STP 问题的发生，因为它在交换机之间提供了一定的弹性。当某个链路发生错误时，链路仅损失一点带宽，数据包在发送前不用等待 STP 修复链路故障。关于端口如何捆绑有一定的规定，对不同系列的交换机，这些规定都不一样。该技术的缺陷是以太网通道只能用来直接连接两台交换机。也就是说，捆绑链路不能跨越多个交换机。
- **端口安全 (Port security)**——所有 Cisco 交换机高级别的安全功能就是端口安全。端口安全允许使用者限制某个 MAC 地址对该端口的访问。当另一个用户以不同 MAC 地址接入该端口时，端口会关闭或者向网络管理站发送报告。这一特性在实际工作中很有用，对交换机的访问提供了严格的控制。
- **多播 (Multicast [CGMP/IGMP])**——本书不讲述任何关于网络多播的内容，并不是说该内容不重要。相反，网络的多播业务在现代网络中的重要性日益提高。有关多播方面的内容将在本书的第 2 卷中讲解。

2.5 令牌环：已有 30 年历史，仍然在使用

如果有在 IBM 大型机 AS400、RS6000 或其他 SNA 设备上工作的经验，就会有使用令牌环网络的经验。上世纪 90 年代，人们认为，随着庞大的“客户/服务器”模式的网络逐渐淡化，大型机和令牌环网络最终也会退出历史舞台。然而，大型机并没有像当时的人们预期的那样完全绝迹，令牌环网络也是如此。在一些大型网络中还是能看得到大型机的身影，而多数大型机又采用了令牌环网络。令牌环是上个世纪 70 年代由 IBM 提出，很快成为 IBM 局域网的首选。不久，IBM 在其前卫的产品，如 IBM 3745 中提供了令牌环接口。当时，令牌环是一种快速局域网介质。在共享介质网络中，以太网只能够以 10 mbit/s 或更低的速率工作。然而，在一个拥有大量用户，并且充满了冲突的以太网段中真正的数据流通量究竟有多少，这个问题争议是非常大的。读者可以看到，令牌环网络是比较确定的，它使用的令牌传递技术使其能够达到它所宣称的通信速率：4 mbit/s 和 16 mbit/s。

2.6 令牌环技术概览

令牌环技术被 IEEE 正式采纳并规范为 IEEE 802.5。IBM 与 IEEE 的规范之间只有细微区别。IBM 的令牌环技术要求工作站以物理的星型拓扑结构通过双绞线连接到多站接入单元 (MSAU)，IBM 8228 就是一个典型的 MSAU。IEEE 没有指定传输介质和拓扑结构，使得令牌环网络更加灵活。

和大多数 LAN 协议一样，令牌环/IEEE 802.5 严格工作在网络的第 2 层上。和以太网一样，IEEE 将数据链路层分成两个子层：802.5 为 MAC 层而 802.2 则为 LLC 层。就功能来说，IEEE 802.5 令牌环和 IEEE 802.3 很相似。LLC 层（此处即指 802.2）是特定的硬件协议 MAC 和第 3 层协议之间的标准接口。

2.6.1 令牌环的工作原理

令牌环网络是星型物理拓扑结构，但是网络实际上是作为一个逻辑环来处理的。图 2-17 显示了逻辑与物理的令牌环结构。

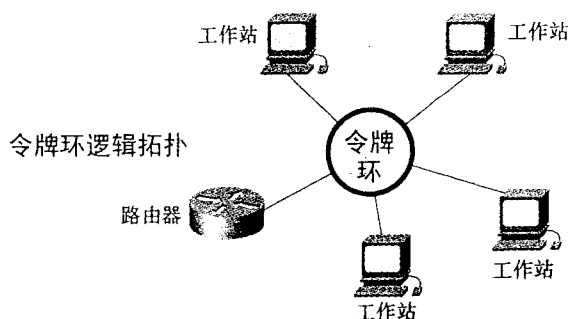
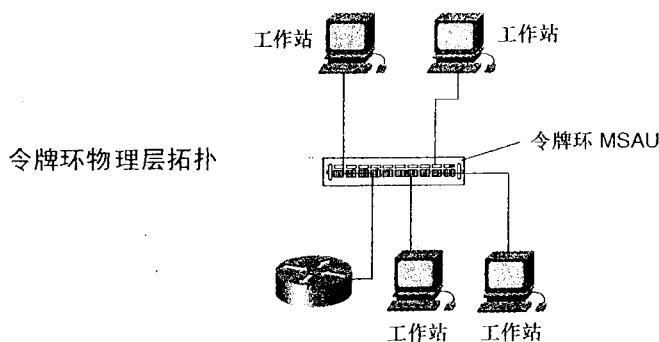


图 2-17 令牌环拓扑结构

令牌环在环中采用令牌传递技术。令牌传递网络中有一个称为令牌的小数据帧沿着环形网络进行传递。令牌环的帧有两种，一种是令牌帧、一种是数据/命令帧。令牌帧包括一个起始分隔符、一个访问控制字段以及一个结束分隔符；数据/命令帧含有同样的数据字段，另外还包括更多的用户数据，同时还含有一个源地址和一个目的地址。某个工作站想要发送数据，必须先拥有令牌。拥有了令牌就意味着工作站获得了传输数据的权利。如果工作站没有数据要传送，就将令牌传递给环形网络中的下一个工作站。

如果工作站获得了令牌并且有数据要发送，它就会改变其访问控制字段中的一个称为 T 的位，然后将它的数据信息附加到帧的后面再一起传送到环中的下一个工作站。数据帧沿着环流动一直到达目的工作站。在目的地址处，目的工作站将数据帧拷贝下来并且做上标记，表明该数据帧已经由其目的工作站接收。然后，数据帧继续沿着这个环流动，直到回到发出数据帧的源工作站。源工作站会删除标记过的数据帧。令牌提前释放的概念允许获得了令牌的工作站送出自己的数据帧之后再发一个新的令牌到环形网上去。

下面列出采用令牌环传输数据的重要特性：

- 访问控制字段 (Access control field) —— 访问控制字段是在两种令牌环帧：令牌帧和数据/命令帧中都有的一个 8 位字段。它包含如下信息：

P|P|P|T|M|R|R|R

- P 3 位长的优先级字段。只有优先级等于或高于令牌优先级的工作站才能获得该令牌。工作站获得令牌并使传输数据帧后，只有本身优先级高于发送令牌工作站优先级的工作站才能够保留令牌用于下一次传递。
- R 3 位长的保留字段，工作站用来保留令牌用于下一次环中传递。
- T 简称为 T 位。如果为 0，该帧是令牌；如果为 1，则是数据/命令帧。
- M 监视位，主动监视器用来删除环中无限传递的帧。
- 主动监视器 (Active monitor) —— 网络中的一台工作站，其职责就是主动对网络进行监控。该工作站要为网络中所有其他工作站提供集中的时钟信息以及完成环形网络维护工作。维护工作之一就是删除网络中无限循环的帧。例如，发出该帧的工作站出了问题，就不能将其发出的帧从环中删除，这导致环中其他工作站都不能传输数据。这时，主动监控者就会通过使用上面提到的 M 位来检测到这种情况，并且删除该帧。
- 可靠投递 (Reliable delivery) —— 令牌环采用 2 位长的帧状态字段确保帧的可靠投递，这些位通常叫做 A 位和 C 位。发出帧的工作站将这些位全置为 0。当这些帧通过了整个环路再返回，发出帧的这个工作站会检查这些位的状态以确认帧在环中的投递情况。接收该帧的工作站会对这些位按表 2-15 进行修改。

表 2-15 接收站对 A 和 C 位的修改情况

A 位	C 位	含 义
0	0	无法找到目的地，接收站不修改位
0	1	无效
1	0	帧被接收，但工作站无法从帧上拷贝数据
1	1	找到工作站，帧被接收并拷贝

- 令牌环帧长度 (Token Ring frame size) —— 令牌环帧的大小通常比 1518 byte 的以

太网的数据帧要大得多。在 4 mbit/s 和 16 mbit/s 的环形网中，令牌环最小帧为 21 byte，最大帧在 4 mbit/s 环中为 4511 byte，在 16 mbit/s 环中为 17 839 byte。

注释 规范与非规范的地址格式

以太网传输数据的格式称为规范地址格式。这意味着，如果有一个比特流 0110 1010，其最高位在比特流的左边而最低位在比特流的右边。以太网以规范的方式传输数据，也就是说最先发送的是数据的最低位。在发送上面的那个数据比特流时，其发送顺序是 01010110。而令牌环和 FDDI 网络发送数据的方式是非规范的。非规范格式的数据发送时先发送最高位。令牌环网发送上面数据时，其顺序为 01101010。源路由转换桥和 DLSw 将会在需要时完成这样的地址转换工作。

2.7 令牌环交换技术

令牌环交换技术很大程度上类似于以太网交换，但也有一些自己的优点。令牌环的最大优势是传输速率。就像以太网的带宽会受到网段中冲突的影响一样，令牌环网络的带宽也要受到等待令牌以发送数据的工作站影响。在一个交换环境中，同一交换机上的端口可以是属于同一环形网，然而从交换机端口上工作站的带宽来看，似乎环上就只有自己存在。令牌环交换机还能提供专用令牌（DTR）。传统的 4 mbit/s 和 16 mbit/s 令牌环接口只能工作在半双工模式下。DTR 定义了一种方法，使得交换机端口能够仿真成一个集中器端口，从而能够全双工地传输数据，这样的模式叫做立即传输（TXI）。它利用了每个端口上只有一台终端工作站从而没有必要真正去传递令牌的事实。因此，这样的令牌环接口能够同时收发数据，使得环路的带宽理论上可以达到 32 mbit/s。

令牌环交换机的端口工作模式有如下几种：

- 半双工集中器端口（Half-duplex concentrator port）——端口以半双工模式连接到工作站。对经典的令牌环来说，端口作为激活的媒体附接单元（MAU）端口来工作。
- 半双工工作站仿真（Half-duplex station emulation）——端口连接到媒体附接单元（MAU）的端口。端口工作时如同一台工作站连接到有多个工作站的传统令牌环网段。
- 全双工集中器端口（Full duplex concentrator port）——端口以全双工模式连接到工作站。
- 全双工工作站仿真（Full duplex station emulation）——端口以全双工模式连接到交换机或集线器

注释 双工模式是集成在网卡中的一种硬件功能。仅仅软件升级不能增加全双工功能。要想能工作在全双工模式，工作站和交换机端口必须同时具有全双工的能力。

2.8 令牌环网桥中继功能（TrBRF）与令牌环集中器 中继功能（TrCRF）

令牌环网桥中继功能（TrBRF）就像是一个多端口的网桥的功能，其目的是桥接多个令牌环。

通过其桥接起来的令牌环称为令牌环集中器中继功能（TrCRF）（这些令牌环或许应该叫做虚拟环，只是人们习惯了这种叫法）。多个 TrCRF 可以连接到一个 TrBRF，就像多个环可以连接到一个网桥上。TrCRF 能够通过源路由桥接（SRB）或者是源路由透明桥接（SRT）进行数据的传输。如果读者对这些桥接技术不太熟悉，可以参考第 13 章“配置桥接和 DLS+”的内容。

TrBRF 将 Catalyst 交换机进行了扩展，就像以太网的中继一样，它使得不同 Catalyst 交换机上的 TrCRF 能够属于同一个网桥号。请记住，TrCRF 实际上就是“令牌环”。在定义一个 TrCRF 之前，必须先创建一个 TrBRF。每个 TrBRF 都是通过网桥号和 VLAN ID 来识别的。创建 TrCRF 时，必须要用一个令牌环号和一个惟一的 VLAN ID 来对其进行标识。每个 TrCRF 都必须分配一个父 TrBRF。

图 2-18 显示了 Catalyst 3920 交换机上 TrBRF 和 TrCRF 的关系。这里 TrBRF 和 TrCRF 的默认值和其他所有 Catalyst 交换机相同。可以把它们想像成以太网交换机上的 VLAN 1。和以太网交换机一样，令牌环交换机也可以在小型网络上实现即插即用的功能。默认情况下，所有的端口都分配给默认的 TrCRF，默认的 TrCRF 又有自己默认的父 TrBRF。

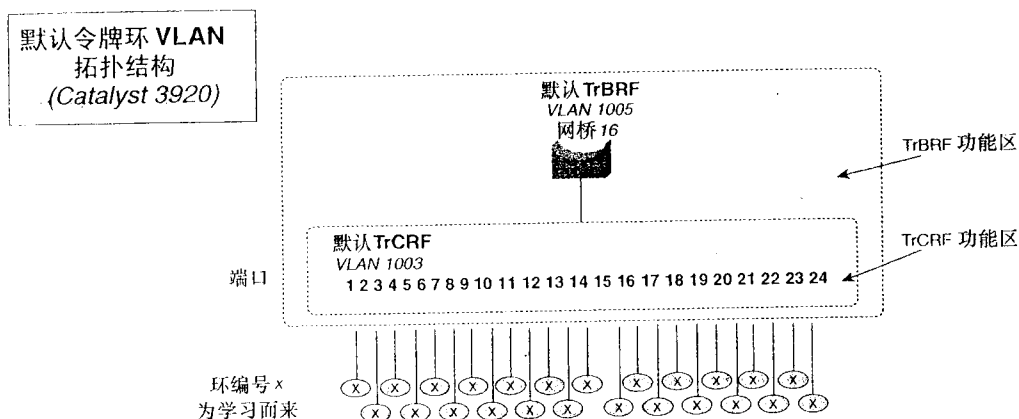


图 2-18 默认 TrBRF, TrCRF 关系

默认 TrBRF 和 TrCRF 的概念使得 LAN 从共享集线器/MSAU 环境下升级非常方便。一台令牌环交换机从产品包装中取出来后，无需设置就可以取代集线器或 MSAU。就像以太网交换机的端口位于默认的 VLAN 一样，令牌环的端口也是在默认的令牌环和网桥上。表 2-16 列出了 Cisco 交换机默认的 VLAN 设置。

表 2-16 默认的 VLAN 设置

特 点	默 认 值
本地或默认 VLAN	VLAN1
VLAN 端口分配	所有端口分配给 VLAN1；令牌环端口分配给 VLAN1003
VTP 模式	透明
VLAN 状态	激活
普通 VLAN 范围	VLAN2 至 VLAN1001

续表

特 点	默 认 值
VLAN 保留范围* VLAN 扩展范围*	VLAN1006-1009 VLAN1025-2094
MTU 大小	以太网 1500 字节 令牌环网 4472 字节
SAID 值	100 000 加 VLAN 号 VLAN2=SAID 100002
允许修剪	VLAN2-1000 允许修剪
MAC 地址缩减	无效
生成树模式	PVST
默认 FDDI VLAN	VLAN1002
默认令牌环 TrCRF VLAN	VLAN1003
默认 FDDI NET VLAN	VLAN1004
默认令牌环 TrBRF VLAN	VLAN1005, 网桥号 0F
TrBRF VLAN 生成树版本	IBM
TrCRF 桥接模式	SRB

*VLAN 保留范围用于 Catalyst6500 系列交换机和用于非保留 VLAN 的映射。VLAN 扩展范围存在于 Catalyst6500 系列交换机。该范围是正常 VLAN 范围的扩展。扩展和保留的 VLAN 信息不在 VTP 上传送。对应全局 VTP 信息，令牌环和 FDDI VLAN 在纯以太交换机上只是列出来而已。同样，令牌交换机的 VLAN 数据库也列出了以太 VLAN。

图 2-19 给出令牌环交换机的逻辑结构。交换机上创建了两个 TrBRF。TrBRF brf100 在网桥 bridge10 上。该 BRF 是 TrCRF crf-ring10 (令牌环 10) 的父 BRF。交换机上分配给该 CRF 的所有端口都在令牌环 10 中。在该图中，端口 16 到 20 是在令牌环 10 上。另一个 TrBRF 是用来将网桥 11 连到令牌环 11，端口 21 到 24 都在令牌环 11 中。

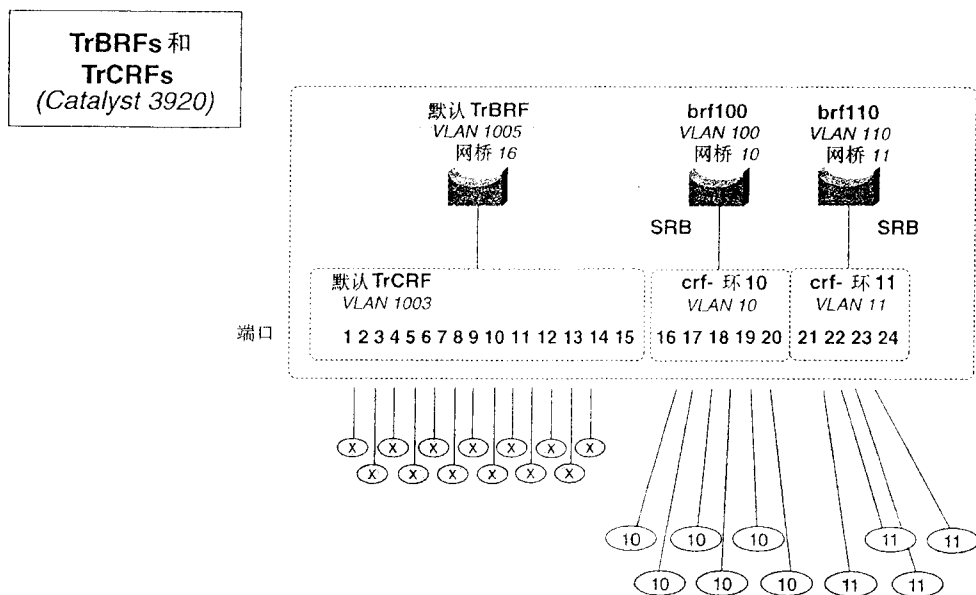


图 2-19 TrBRF 和 TrCRF 的逻辑关系

设置完该交换机，整个网络中就有了两个桥接域。图 2-20 以一种更为传统的方式表示出了这些概念。

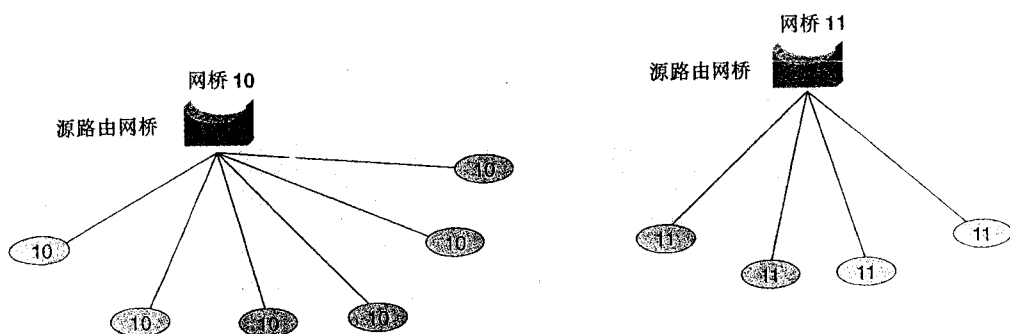


图 2-20 TrBRF 和 TrCRF 的概念视图

当桥接域之间需要进行通信时，他们需要通过另外一个源路由桥来实现，通常是通过一台路由器的方式。图 2-21 说明了如何用一台路由器来连接两个桥接域。如果网络传输的是 SNA 这样的桥接协议，路由器可以设置成源路由桥接，而这两个桥接域则可通过源路由桥接进行连接。如果网络传输的是 IP 路由选择协议，路由器可设置为在两个桥接域之间进行数据路由。

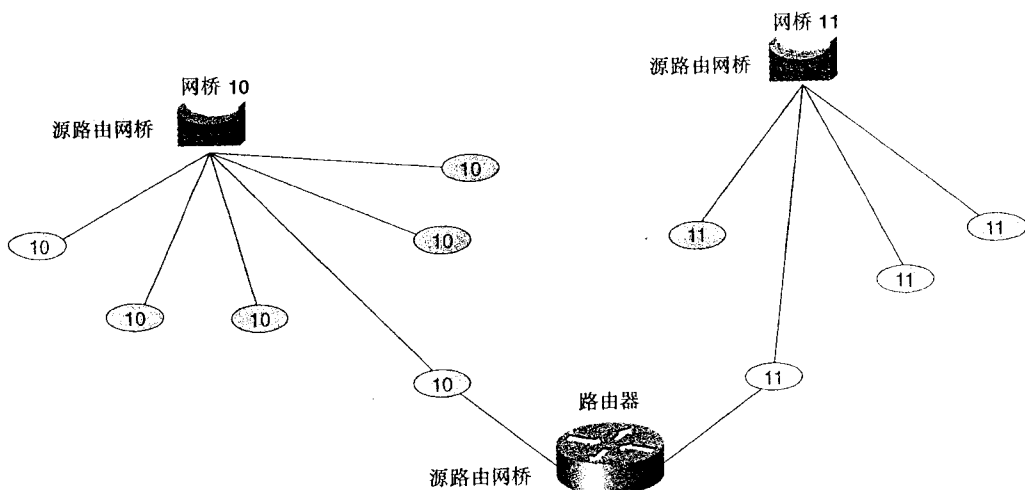


图 2-21 连接两个令牌环桥接域

2.9 在 Catalyst 3920 上配置令牌环交换

以太网交换技术中的很多概念同样适用于令牌环交换技术。例如，在令牌环网络中，同样能够见到 VLAN、VTP 域以及管理 VLAN 的概念。因此，我们不再讲解这些术语及其使用原理，而是侧重于交换机配置的讲述。

Catalyst 3920 没有提供用来设置的标准命令行接口，而是采用完全菜单化和界面式的配置界面。用户可以用光标和按键选择不同的界面选项而不必记住那些命令的句法，从而使得交换机的配置变得更加容易。

图 2-22 显示了 Catalyst 3920 令牌环交换机的主界面。

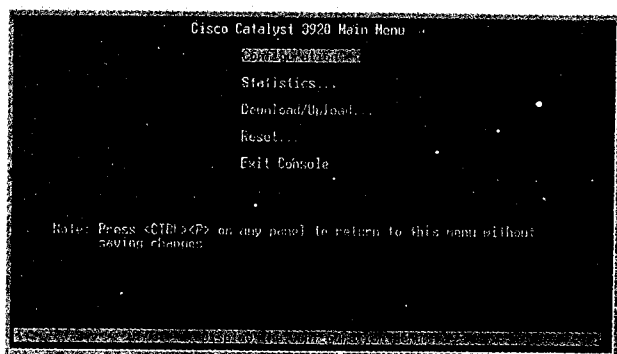


图 2-22 初始配置面板

在配置界面里可以进行交换机的 TrBRF、TrCRF、VLAN、管理 VLAN 以及其他一些软件功能的配置；在信息统计界面里，可以使用交换机的 **show** 功能，可以查看端口状态，VTP 状态以及其他一些重要的信息；在下载/上传界面里，可以升级 Catalyst 3920 的系统软件；最后，在复位界面里，可以清除 NVRAM 的内容以及重启交换机。下面对这些界面中的重要设置进行简单的说明。

2.9.1 交换机的配置界面

图 2-23 显示了交换机的配置界面。

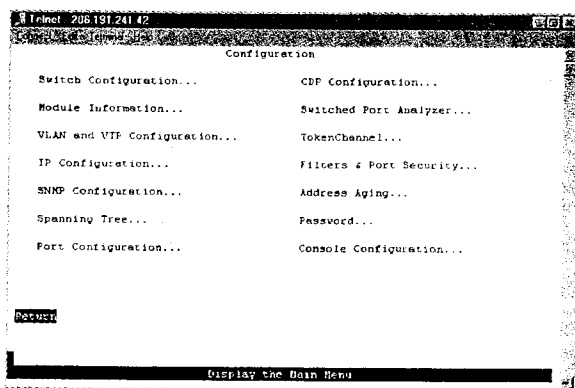


图 2-23 交换机的配置面板

交换机的配置界面里各功能界面包括：

- **交换机配置**——该界面显示系统信息以及交换机的硬件信息，如内存、软件类型。

该界面也可设置基本的交换机信息，如系统名称、日期、位置和联系方式等，还能

为交换机指定一个 MAC 地址以及指定交换机使用规范或非规范的地址格式。令牌环网的默认格式是非规范的。这里还能看到交换机的上连信息。

- 模块信息——该界面可以显示交换机模块、状态、正常运行时间以及软硬件版本等信息。
- VLAN 和 VTP 的配置——在该界面里，是需要花时间进行配置的，包括 VTP 域的设置更改、VLAN 信息、TrBRF 和 TrCRF 信息等，同时还要为 TrCRF 分配端口。
- IP 配置——在该界面里，可以对交换机的基本 IP 管理接口进行配置，包括 IP 地址、子网掩码和默认网关。只有给管理 TrBRF 分配 IP 地址后，SNMP 配置界面里的设置才能开始工作。在管理 TrBRF 和 IP 地址设置生效之后，在这里还能发送 ping 命令。
- SNMP 配置——该界面可以设置基本的 SNMP 读写、陷阱以及通信字符串等。Catalyst 3920 交换机支持的 RMON 也在此启动。管理 VLAN 或者是管理 TrBRF 必须先预设，使交换机的 SNMP 和 IP 开始工作。默认情况下，SNMP 和 IP 配置为使用默认的 TrBRF。
- 生成树——该界面可以在网桥环境下配置 STP。以前讨论过的所有同样的 STP 参数和定时器都可以用来配置令牌环网络。
- 令牌环端口配置——该界面能够显示和改变物理、逻辑端口的信息。在这个界面里可以设置双工模式，令牌提前释放、MTU 和环速率等，还能将端口交换模式从自动的直接转发更改为以前提到过的其他三种模式中的任何一种。这个界面还提供探测帧广播收听的功能。
- CDP 配置——在这个界面中，用户可以配置交换机的 Cisco CDP 信息并显示其状态。默认情况下，CDP 信息在所有的端口上都是打开的。
- 交换端口分析（SPAN）——在这个界面里，用户可以接一台网络分析仪或其他类似设备到交换机上。由于数据帧并不是自动转发到 VLAN 中的每一个端口去的，所以，这一功能在监视某个交换机端口时尤其有用。
- 令牌信道——在这个界面中，用户可以创建一个令牌通道。前 8 个端口能够合并成一个令牌通道。该技术和以太通道非常相似。
- 过滤与端口安全——在这个界面里，用户能够设置允许 MAC 地址和协议过滤以及端口安全。
- 密码——可以在这个界面中为交换机设置密码。
- 控制台设置——用户可以在这个界面中设置和显示当前的 Telnet 会话以及物理控制台端口。默认的会话超时为 5 秒，可以在这个界面中改变为最大 1440 秒。

2.9.2 信息统计界面

该界面主要是用来显示端口、VLAN 和 VTP 的各种不同状态信息以及其他重要信息，包括：

- 端口状态。
- 端口统计信息。

- 当前生成树信息。
- VLAN 统计信息。
- CDP 相邻用户显示。
- VTP 统计信息。
- 诊断测试结果。
- 数据日志信息。
- 显示信息汇总。

2.9.3 下载/上传界面

从该界面可通过 TFTP 或 RS 232 接口升级交换机的 IOS。

2.9.4 复位界面

在这个界面中，除了能进行复位操作外，用户还可在该界面里清除交换机 NVRAM 中的信息。在复位界面中，用户可以：

- 带诊断性的复位交换机。
- 不带诊断性的复位交换机。
- 复位端口地址表。
- 清除 NVRAM 内容。

2.9.5 在 Catalyst 3920 交换机上设置 VLAN

合理的配置令牌环网络的 VLAN 的步骤和以太网的相关配置一样。然而，在令牌环网络中，必须定义令牌环和网桥，这样就多出了几个步骤。在 Catalyst 3920 交换机上配置 VLAN 的步骤如下：

第 1 步 规划 TrBRF、TrCRF、令牌环号、网桥数目以及 VLAN。

第 2 步 设置 VTP。

第 3 步 配置 TrBRF VLAN 并为每个 TrBRF 分配网桥号。

第 4 步 配置 TrCRF VLAN 并为其分配一个父 TrBRF 和一个可选的令牌号。

第 5 步 分配端口给 TrCRF。

第 6 步 配置交换机管理。

令牌环交换机的配置包括 TrBRF VLAN 和 TrCRF VLAN 的配置。这里的两套 VLAN，令牌环，网桥等等的关系很容易使人混淆。为此，读者最好利用额外时间，画一张小图来表示所有实体之间的逻辑关系。

图 2-24 给出了同一个令牌环网络的两个视图。交换机 dragon 是 Cisco Catalyst 3920，它连接了两台路由器和一个用户工作站。现在，按照上面的 6 个步骤来配置这个交换型令牌环网。

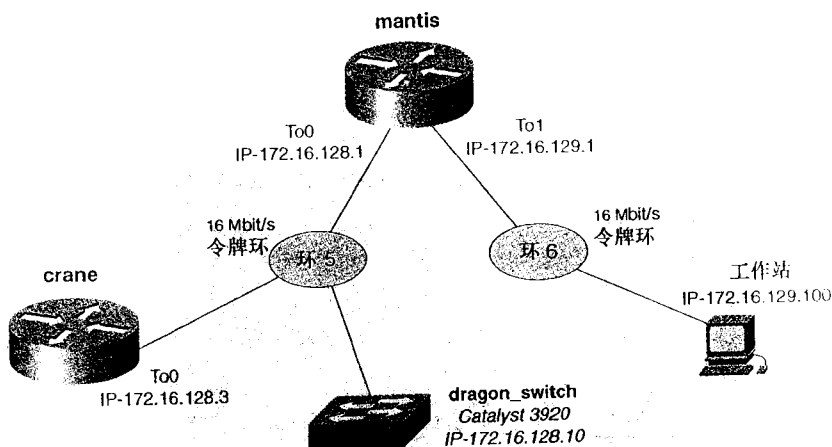
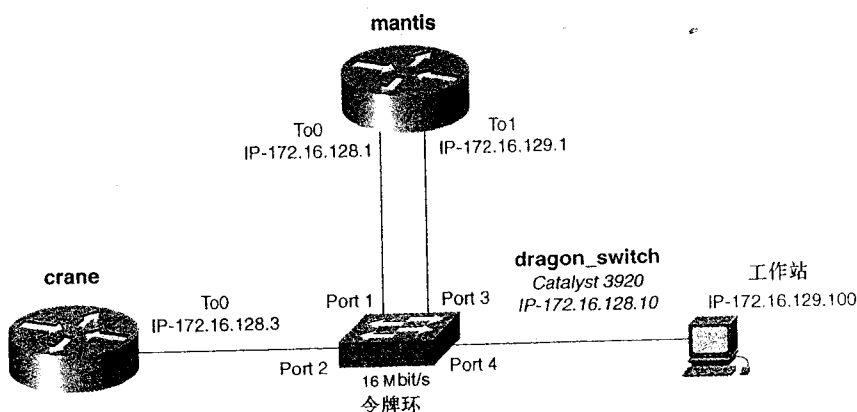


图 2-24 令牌环交换网络模型

1. 第1步: 规划 TrBRF、TrCRF、令牌环号、网桥数目以及 VLAN

在第1步里要为 VLAN、TrBRF 和 TrCRF 的配置制定计划。图 2-25 为从交换机 dragon 的角度来看的网络。需要定义两个 TrBRF 以及相关的 VLAN 和网桥号，还需要定义两个 TrCRF 及其 VLAN 和适当的令牌环号。TrCRF 是通过配置 TrBRF 是其父 TrBRF 来与其进行连接，而不是通过通常的 VLAN 来和 TrBRF 进行连接。VLAN 和网桥号之间不需要匹配。

为了便于记录网络文档，请使用合理的命名规范。本例中，TrBRF 叫做 brf5，因为它位于 VLAN 50 和网桥 5 上，叫 crf ring5 的 VLAN 5 包含 ring 5，它的父 TrBRF 就是 brf5。另外一个 TrBRF 叫做 brf6，它位于 VLAN 60 和网桥 6 上。该 TrBRF 是 TrCRF crf ring6 的父 TrBRF，相应地，叫 crf ring6 的 VLAN 6 包含 ring 6。

2. 第2步: VTP 的配置

第2步要开始配置交换机。从初始配置界面着手，首先，选择 VLAN 和 VTP 配置界面。在该界面里，进入 VTP 管理配置界面，如图 2-26 所示。在 VTP 管理配置界面中，可以进行

VTP 域名、模式、密码的设置。

TrBRF 和 TrCRF
概念
(Catalyst 3920)

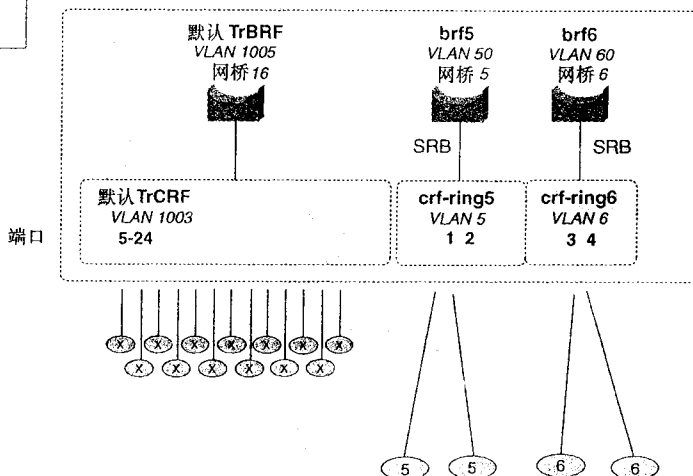


图 2-25 TrBRFs 和 TrCRFs 的定义

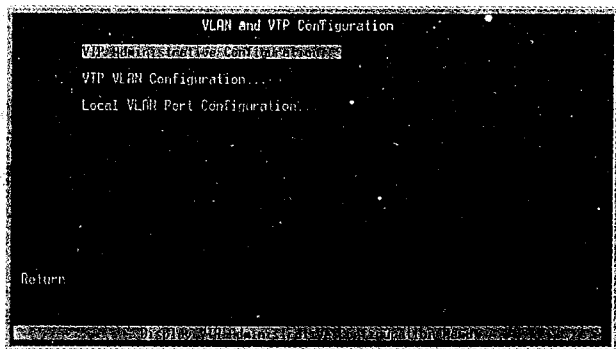


图 2-26 VLAN 和 VTP 的配置界面

图 2-27 所示为管理配置界面中将 cisco 设置为 VTP 域名的例子。



图 2-27 VTP 管理配置界面

3. 第3步: TrBRF VLAN (s) 的设置以及为每个 TrBRF 分配网桥号

第3个步骤是对 TrBRF 及其相关的 VLAN 进行设置，给每个 TrBRF 分配一个网桥号。如上所述，有两个 TrBRF 需要设置。TrBRF 的设置也是在同样的 VLAN 和 VTP 配置界面下选择 VTP VLAN 设置界面来进行的。在这一界面中，选 **Add** 来创建一个新的 TrBRF。请注意这里所说的进入 VLAN 中 VLAN 是指 TrBRF 中的 VLAN。图 2-28 和 2-29 显示了如何创建叫 brf5 的 TrBRF VLAN 50。

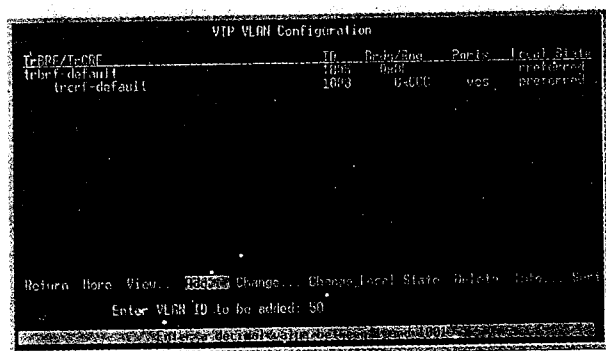


图 2-28 VTP VLAN 配置界面

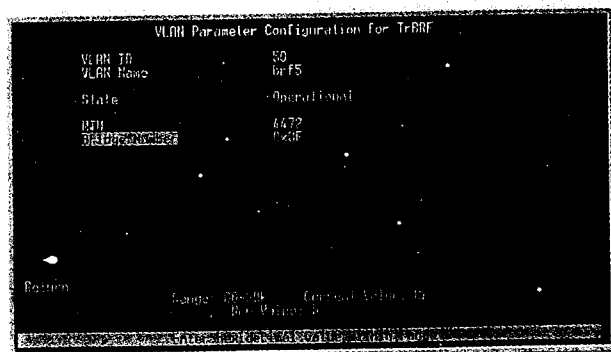


图 2-29 VLAN 参数配置界面

创建新 VLAN 时，交换机会提示用户，是将其作为 TrBRF 还是 TrCRF 的 VLAN。如果选择 TrBRF，就会进入 VLAN 参数配置菜单。在该菜单里，用户可以输入 TrBRF 的名称以及更改 VLAN 的 ID、状态和 MTU，还允许用户为 VLAN 选择其网桥号。网桥的输入和显示都是 16 进制格式的。以后在使用源路由网桥接功能时，注意输入的网桥 ID 号不要混淆 16 进制和 10 进制。图 2-29 所示为 VLAN 参数配置界面。

4. 第4步: TrCRF VLAN 的设置以及分配一个父 TrBRF 和一个可选的令牌环号

定义完两个 TrBRF VLAN 后，第4个步骤是完成 TrCRF 的定义以及为其分配一个父 TrBRF。创建 TrCRF，需要按照上一步中提到的创建一个惟一的 VLAN。该 VLAN 不适用于 TrCRF 到 TrBRF 的连接，它需要有一个惟一的 VLAN ID。在提示选择 VLAN 类型时，选 TrCRF。例 2-30 给出了 TrCRF 和 VLAN 5 创建过程的例子。

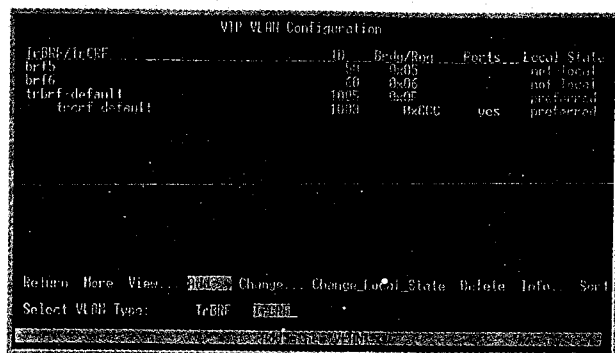


图 2-30 VTP VLAN 配置界面

图 2-31 中是用于 TrCRF 设置的 VTP VLAN 配置界面。在该界面中，可以指定 VLAN 和父 VLAN 的名称。例中 VLAN 的名称 crf ring5 是为了方便建立网络文档。重要的是父 VLAN，这里我们分配了一个称为 brf5 的父 VLAN。令牌环号默认地设为自动，表明交换机会从真实令牌环号所在的路由选择信息字段（RIF）处自动地确定令牌环号。在这个例子里，可以将令牌环号设为 ring 5，该号的显示和输入也都是 16 进制格式的。

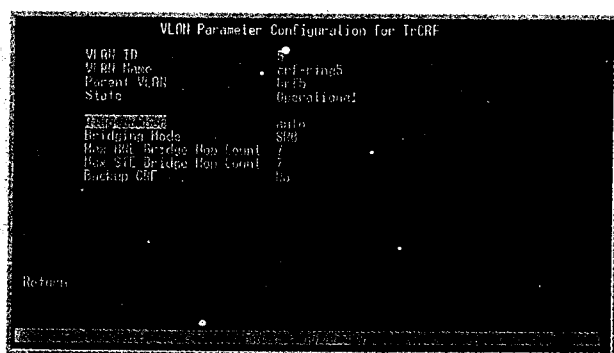


图 2-31 VLAN 参数配置界面

图 2-32 显示的 VTP VLAN 配置界面列出了新创建的 VLAN。

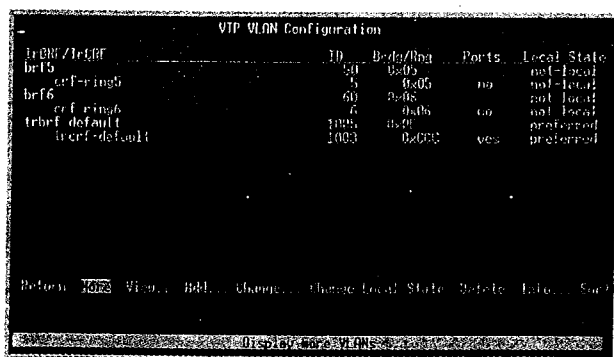


图 2-32 VTP VLAN 配置界面

5. 第5步: 为 TrCRF 分配端口

第5步包括令牌环交换机的端口的设置以及将这些端口分配给 TrCRF VLAN。VLAN VTP 配置界面的第3个选项: 本地 VLAN 端口设置, 使得用户可以改变所有的端口的默认 TrCRF。如上所述, 所有的端口默认已分配给 T 默认的 TrCRF。选择 **Change** 项之后, 交换机会提示选择所要改变的端口。选定端口之后, 交换机会列出当前已有的 TrCRF 并且允许用户对其进行更改。图 2-33 显示的就是端口 1 到 3 经过改动之后的 VLAN 端口配置界面。

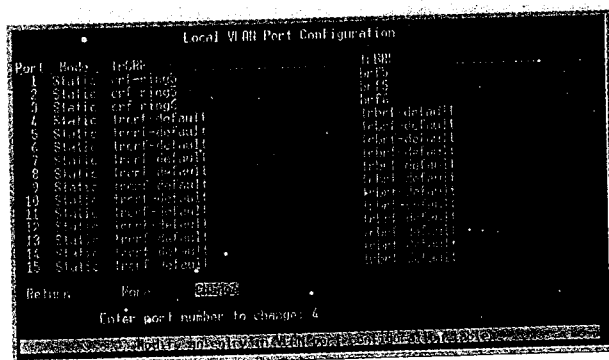


图 2-33 本地 VLAN 端口设置

设置到了配置工作的这一步, 交换机已经完全可以进行工作。如果要配置该网络中的路由器, 只需令牌环接口以及基本路由即可。例 2-46 是在路由器 crane 上对令牌环接口的设置。

例 2-46 路由器 crane 上的令牌环设置

```
crane(config)#int tokenRing 0
crane(config-if)#ring-speed 16
crane(config-if)#ip address 172.16.128.3 255.255.255.0
crane(config-if)#no shut
```

6. 第6步: 交换机管理的配置

配置过程的最后一步是设置 IP 地址和管理交换机功能。交换机管理功能, 如联络人姓名和信息, 规范的地址格式等信息能够在交换机管理界面中加以更改, 如图 2-34 所示。

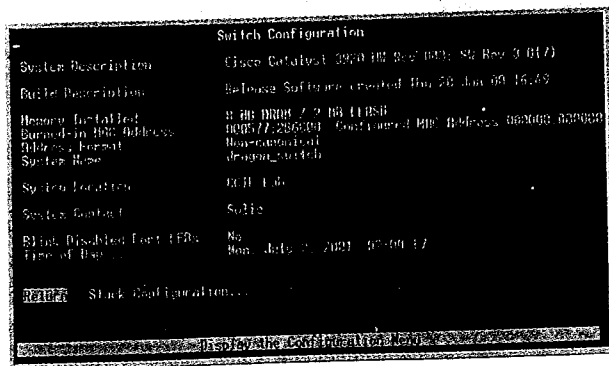


图 2-34 交换机设置

IP 设置，如交换机的 IP 地址以及默认网关，是在 IP 配置界面中进行的。选定该界面后，所有的 TrBRF 都会被列出，以供选择，交换机会提示用户选择一个 TrBRF VLAN 以设置 IP 地址。在选定一个 TrBRF 之后，IP 配置界面就会显示出来。在该界面中，可以输入的信息包括 IP 地址、子网掩码以及默认网关。也可以在这个界面中设置交换机使用自举协议 (BootP) 功能。图 2-35 为该界面示例。

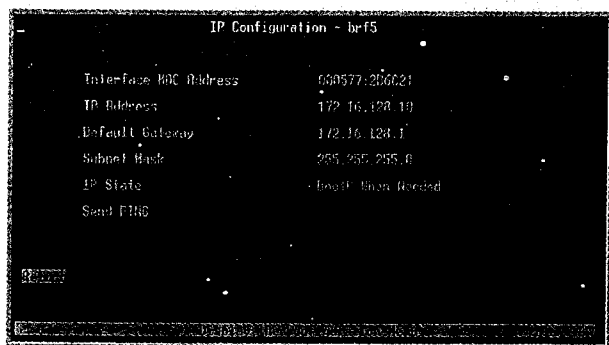


图 2-35 IP 配置界面

在该界面中还可以使用 ICMP 的 ping 命令。然而，交换机每次都只能有一个管理地址。如果没有改变管理 VLAN，交换机会使用默认的 VLAN TrBRF。Catalyst 2900 和 3500 可以有多个配有 IP 地址的虚拟接口，但同一时间内只能激活一个。要将当前起作用的默认管理 VLAN 从 TrBRF 改为另外一个 TrBRF VLAN，可以在配置界面里选择 SNMP 界面进行要求的操作。SNMP 界面还能改变默认的 VLAN 以及允许 RMON 和 SNMP 断点，定义通信字符串等。图 2-36 中就是一个 SNMP 配置界面的例子，这里新的默认 TrBRF 是 brf5。

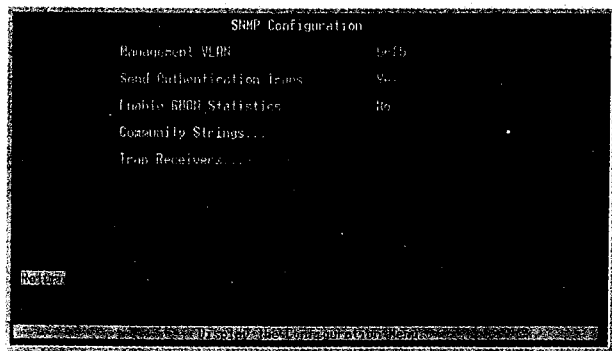


图 2-36 SNMP 配置界面

在主配置界面中可以对交换机的操作进行确认。在主配置界面的菜单中，信息统计项提供了大量的统计信息以供查看。熟悉菜单中的这些命令的最好的方法不是阅读相关的书籍，而是在交换机上选择这些选项以观察各条命令的功能。图 2-37 显示的统计界面列出了可供选择的选项中的很小一部分。图 2-38 则是一个配置菜单，是另一个查看端口状态的有效方式。

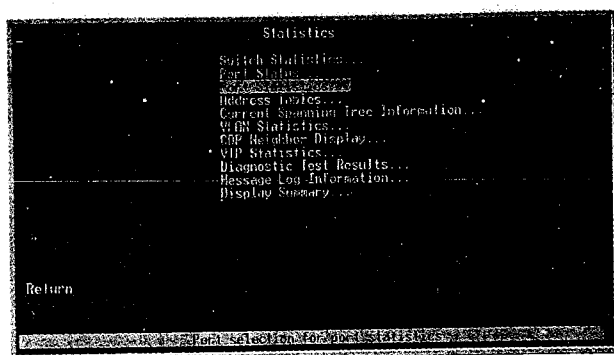


图 2-37 统计界面

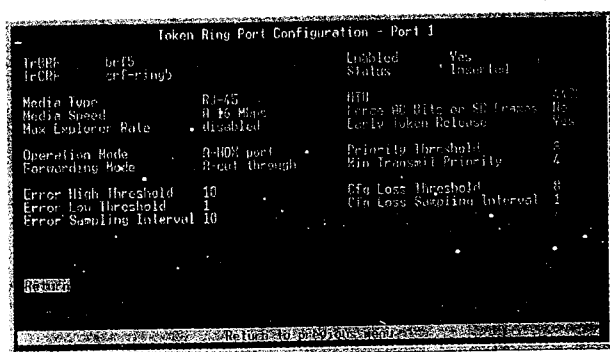


图 2-38 端口配置

2.10 更多练习：以太网/令牌环网实验

本书剩余内容中将提供大量的配置以太网和令牌环网 VLAN 的机会。本章中关于路由选择的实验以及后续章节的实验都包含有以不同方式对 LAN 进行配置的练习。如果条件允许，应好好利用这些实验章节的内容。记住，尽量不要用集线器建立网络模型，而应想办法创建含有 VLAN 和 VLAN 中继的模型。

下面的实验包括了已经提过的许多概念。在本书中，我们没有安排小型实验，像配置 VLAN 和 VLAN 中继等等。我们的实验都是比较复杂的。这些实验不仅向读者提出了挑战，同时也为读者提供了一个对将来在现场工作中可能遇到问题的实践练习机会。

2.11 实验 7：以太交换、VLAN 中继和生成树根布局

——第 1 部分

2.11.1 实验说明

这些交换机也是用户容易掌握使用的，无需过多的配置工作。只有在网络中加入了一定的冗余度时，情况才会变得比较复杂一些。该实验就是要配置一个冗余的以太网交换网络。

2.11.2 实验内容

Game LANs, Inc. 是一家在全国范围内为娱乐中心提供高速骨干网的公司，其局域网所用的交换机是思科公司的百兆和千兆以太网交换机。Game LAN 的一部分网络是冗余的。

我们的任务是以下面的参数为标准配置一个 Game LAN 网络：

- 如图 2-39 配置一个以太网交换网络。
- 网络中所有交换机使用的 VTP 域名是 funtime。
- 在网络中创建 3 个 VLAN，但是不要使用 VLAN1。VLAN1 是用作管理的，也就是 IP 子网 172.16.128.0/24。对其做上标记以便其他交换机在查看它的信息时能够很容易地认出它就是用于管理的 VLAN。另外两个 VLAN，一个（172.16.16.0/24）用于 glaccess_2 路由器，另一个（172.16.17.0/24）则用于 glaccess_1 路由器。
- 对路由器 gameserver_1 和 gameserver_2 进行配置，以便在网络所有的 VLAN 之间进行路由。所有的 VLAN 和 IP 地址都可以互相访问。以 EIGRP 为路由选择协议，自治系统（AS）为 2001。
- 对 STP 进行调整以便所有的 VLAN 中的根都与 HSRP 的配置相一致。VLAN 128 和 17 的根应该是 gl_switch1，而 VLAN 16 的根则是 gl_switch2。
- 对 gl_switch 进行配置，使得只有子网 172.16.17.0/24 上的设备可以对交换机进行 Telnet。
- （可选）端口 2/24 上是一台安全工作站，其 MAC 地址是 0000.863c.3b41。对这个端口进行配置使得仅有这台工作站才可以工作在该端口上，如果其他工作站试图接入就关闭这个端口。

2.11.3 实验目的

- 如图 2-39 所示对以太网交换网络进行配置。
- 在整个网络中使用一个 VTP 域。将 gl_switch1 和 gl_switch2 配置为 VTP 服务器，而 gl_switch3 则是 VTP 客户端设备。如图 2-39 配置 VLAN 中继和中继类型。注意要配置两种中继类型，802.1q 和 ISL。
- 确保所有 IP 接口的完全 IP 连通性——也就是能够 ping 通所有的 LAN 接口。
- 在网络中创建 3 个 VLAN，但是不要使用 VLAN1。一个 VLAN 用作管理，也就是 IP 子网 172.16.128.0/24。对其做上标记以便其他交换机在查看它的信息时能够很容易地认出它就是用于管理的 VLAN。另外两个 VLAN，一个是给子网 172.16.16.0/24 的 glaccess_2 路由器用的，另一个则是为 172.16.17.0/24 的 glaccess_1 路由器准备的。
- 对路由器 gameserver_1 和 gameserver_2 进行配置，以便在网络中所有的 VLAN 之间进行路由。该实验中用的路由选择协议是 EIGRP。在路由器之间进行 HSRP 的配置，每个子网中的第 1 个 IP 地址作为 HSRP 共享地址。举个例子，管理子网 172.16.128.0/24 应该采用 172.16.128.1 作为其 HSRP 共享地址。配置 HSRP 时，要

使得 gameserver_2 作为子网 172.16.16.0/24 的 HSRP 主服务器，而 gameserver_1 则是子网 172.16.128.0/24 和 172.16.17.0/24 的主服务器。所有的 VLAN 和 IP 地址都应该能够相互访问。

Game LAN 公司的

交换式以太网

VTP 域 = funtime

EIGRP AS 2001

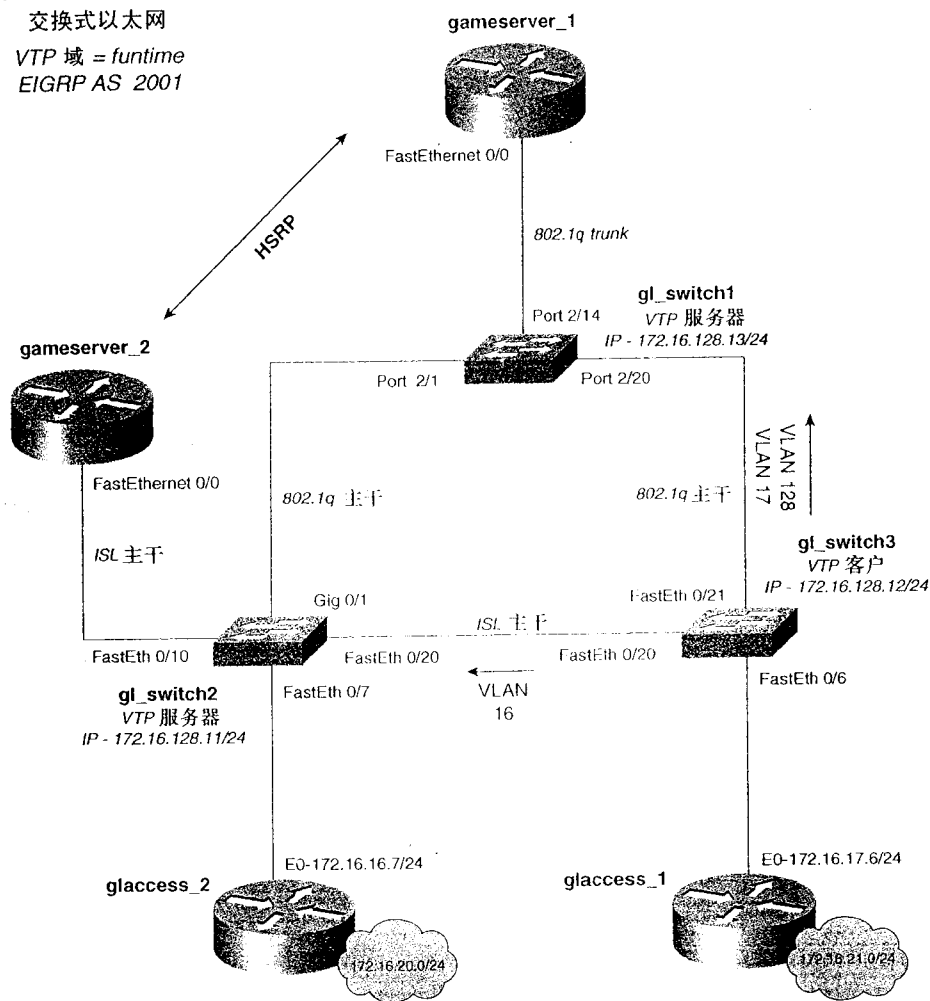


图 2-39 Game LAN 的以太网网络

- 对 STP 进行调整，以便所有的 VLAN 根都与 HSRP 的配置相一致。VLAN 128 和 17 的根应该是 gl_switch1，而 VLAN 16 的根则是 gl_switch2。
- 对 gl_switch 进行配置，使得只有子网 172.16.17.0/24 上的设备可以对交换机进行 Telnet。
- （可选）端口 2/24 上是一台安全工作站，其 MAC 地址是 0000.863c.3b41。对这个端口进行配置使得仅有这台工作站才可以工作在该端口上，如果其他工作站试图接入就关闭这个端口。

2.11.4 所需设备

- 具有以太接口的 4 台 Cisco 路由器，其中两台必须要有快速以太接口。回想一下讲过的内容，要运行 VLAN 中继协议，至少需要百兆的端口配置。如果没有百兆接口的路由器，把一台路由器的 3 个以太接口连接到交换机上也可以用来运行路由选择协议。一个 VLAN 一个接口。
- 3 台 Cisco Catalyst 以太网交换机。这个实验室的设计是采用两台 Catalyst 2900/3500 系列交换机和一台 Catalyst 4000/5500/6500 系列交换机。
- 为了增加读者的实践经验，实验中使用两种类型的 Cisco Catalyst 平台。其中 gl_switch1 是 Catalyst 4000/5500/6500 系列，而 gl_switch2 和 gl_switch3 则是 Catalyst 2900/3500 系列。具体是哪种类型对本实验的功能并无影响。

2.11.5 物理设计与实验准备

- 网络 172.16.20.0/24 和 172.16.21.0/24 分别是在 glaccess_2 和 glaccess_1 路由器上用环路接口模拟的。
- 这一章没有侧重讲解该实验中所用到的 EIGRP 和 HSRP 配置内容。我们会在实验步骤中告诉读者应该怎样做，如果想深入了解这方面的内容，可以查阅相关章节的介绍。

2.12 实验 7：以太网交换、VLAN 中继和生成树根布局 ——第 2 部分

2.12.1 实验步骤

利用 5 类 (Cat 5) 反接电缆将交换机连接在一起。背对背模式的交换机需要用 Cat 5 反接电缆连接。再用 Cat 5 正接电缆将路由器和交换机连到一起，如图 2-39 所示。

在建立这个模型时，首先配置以太网交换机，最后配置路由器。首先，定义 VLAN 及其 IP 子网。图 2-40 准确地反映了此时网络的 VLAN、HSRP 和 IP 地址的拓扑。

还需要的是定义和创建如下 VLAN：

- **VLAN 1**——在这个实验中不使用该 VLAN
- **VLAN 16**——IP 子网 172.16.16.0/24.
- **VLAN 17**——IP 子网 172.16.17.0/24.
- **VLAN 128**——IP 子网 172.16.128.0/24 (新的管理 VLAN)

VLAN ID 不需要与其子网相匹配，我们特意使 VLAN ID 与其子网相匹配是为了网络管理的方便。

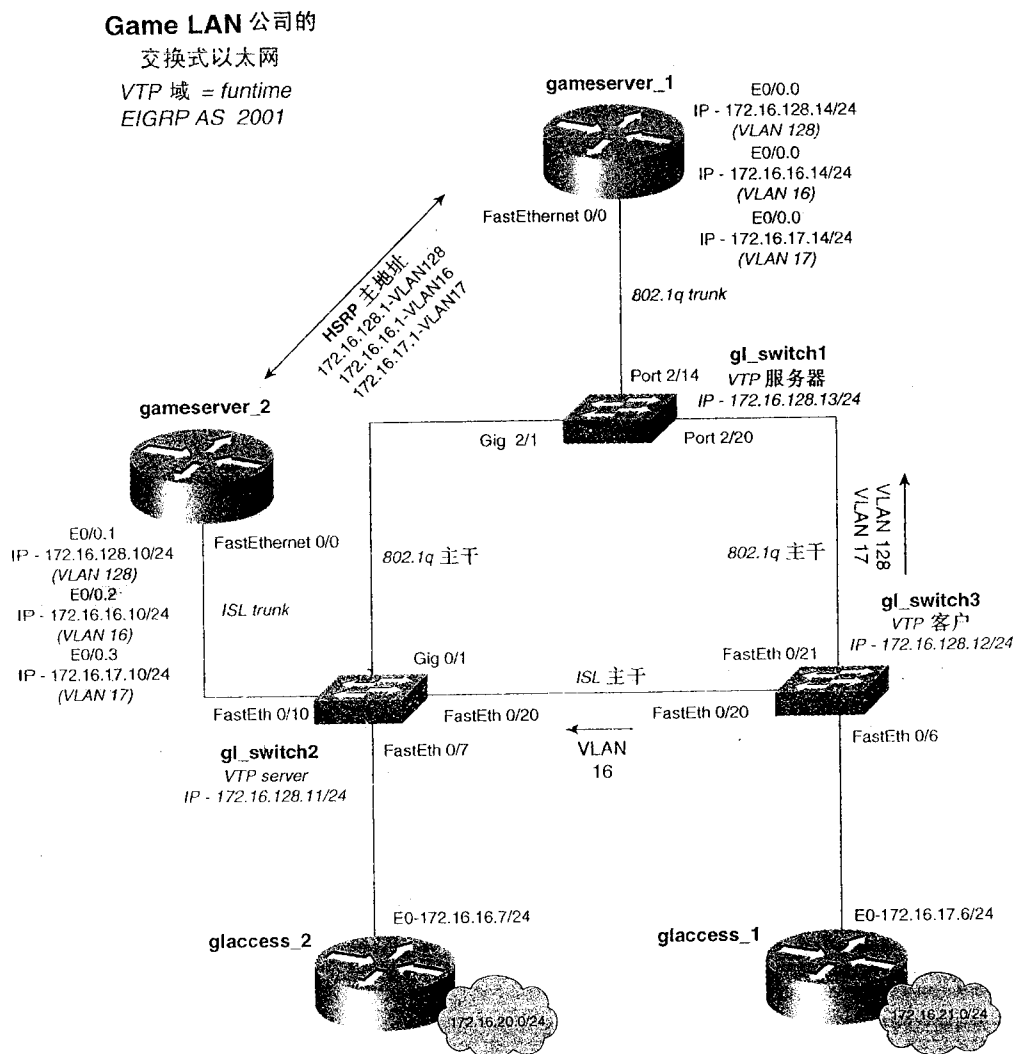


图 2-40 Game LAN 的以太网网络

首先配置 gl_switch1。这个实验中，该设备是一台 Catalyst 4000/5500/6500 系列交换机。

回想一下本章中讨论过的 4 步骤配置过程：

- 第 1 步 配置交换机管理功能。
- 第 2 步 配置 VTP 和 VLAN。
- 第 3 步 如果需要的话，配置 VLAN 中继。
- 第 4 步 （可选）对 STP 和 VLAN 的广播进行控制。

第 1 步要求配置交换机的基本管理功能。由于不采用 VLAN 1 作为默认的 VLAN，因此在配置管理接口之前要先创建一个新的默认 VLAN。要想允许对交换机的 IP 访问，还需要设置一个密码。例 2-47 就是在 gl_switch1 上创建 VTP 域，VLAN 以及新的管理接口的示例。

例 2-47 gl_switch1 的初始配置

```
Console> (enable) set prompt gl_switch1
gl_switch1 (enable) set vtp domain funtime
VTP domain funtime modified
gl_switch1 (enable) set vlan 16
Vlan 16 configuration successful
gl_switch1 (enable) set vlan 17
Vlan 17 configuration successful
gl_switch1 (enable) set vlan 128 name management
Vlan 128 configuration successful
gl_switch1 (enable)
gl_switch1 (enable) set int sc0 128 172.16.128.13 255.255.255.0
Interface sc0 vlan set, IP address and netmask set.
gl_switch1 (enable) set ip route 0.0.0.0 172.16.128.1
Route added.
gl_switch1 (enable)
```

例 2-47 给出了默认路由的添加过程，交换机用 **set ip route 0.0.0.0 172.16.128.1** 命令能够将所有 IP 数据转发到这个地址，这个地址必须和管理接口同处一个子网中，这个例子中就是子网 172.16.128.0/24。

要确认 VTP 域是否处在活动状态，可以用 **show vtp domain** 命令来查看是否配置错误。

现在已经在做第 2 步的第 1 部分工作了，通常情况下，还需要为 VLAN 分配端口。但是交换机并没有把端口分配到 VLAN 中。该交换机只有用于与其他交换机相连接的中继。因此就可以跳到第 3 步去定义 VLAN 中继。在配置中继时，设其为静态 802.1q 中继模式。例 2-48 是在交换机 gl_switch1 上配置中继线路的情况。

例 2-48 gl_switch1 上的中继配置

```
gl_switch1 (enable) set trunk 2/1 dot1q
Port(s) 2/1 trunk type set to dot1q.
gl_switch1 (enable) set trunk 2/1 on
Port(s) 2/1 trunk mode set to on.
gl_switch1 (enable) set trunk 2/14 dot1q
Port(s) 2/14 trunk type set to dot1q.
gl_switch1 (enable) set trunk 2/14 on
Port(s) 2/14 trunk mode set to on.
gl_switch1 (enable) set trunk 2/20 dot1q
Port(s) 2/20 trunk type set to dot1q.
gl_switch1 (enable) set trunk 2/20 on
Port(s) 2/20 trunk mode set to on.
gl_switch1 (enable)
```

最后一步是对生成树进行调整。只有所有的中继已经正常地处在工作状态之中之后才能进行这一步的操作。当整个网络（包括路由器在内）都收敛之后，可以调整所有交换机的 STP。现在先配置好网络中剩余的交换机。交换机 gl_switch2 和 gl_switch3 的初始配置情况类似。先对 gl_switch2 进行操作，配置它的基本管理功能。例 2-49 就是该交换机管理接口的配置示例。切记，gl_switch2 和 gl_switch3 是 Catalyst 2900/3500 系列的，因此它们的配置和 gl_switch1 不尽相同。

例 2-49 gl_switch2 的管理功能配置

```
Switch(config)#hostname gl_switch2
gl_switch2(config)#int vlan1
gl_switch2(config-if)#shut
gl_switch2(config-if)#exit
01:35:54: %LINK-5-CHANGED: Interface VLAN1, changed state to administratively do
wn
01:35:55: %LINEPROTO-5-UPDOWN: Line protocol on Interface VLAN1, changed state t
gl_switch2(config)#int vlan128
gl_switch2(config-subif)#ip address 172.16.128.11 255.255.255.0
gl_switch2(config-subif)#no shut
gl_switch2(config-if)#exit
gl_switch2(config)#ip default-gateway 172.16.128.1
```

要实现到交换机的 IP 连接，还需要配置一个默认网关，如例 2-48 所示。默认网关必须和当前激活的管理接口 VLAN 128 处在同一个子网，网关的 IP 地址是 HSRP 地址 172.16.128.1。

在第 2 步中，配置 VTP 域和 VLAN。由于已经在 gl_switch1 定义了 VLAN，因此没有必要再在其他交换机上配置 VLAN。实际上，中继开始工作之后，就可以依靠 VTP 来传输 VLAN 信息了。

例 2-50 是在 gl_switch1 上配置 VTP 域的示例。

例 2-50 gl_switch2 上的 VTP 域创建

```
gl_switch2#vlan database
gl_switch2(vlan)#vtp domain funtime
Changing VTP domain name from cisco to funtime
```

交换机 gl_switch2 在其 fastEthernet 0/7 接口上有一个用户 VLAN。用 **switchport** 命令把接口 fastEthernet 0/7 分配给 VLAN 16，如例 2-51 所示。

例 2-51 在 gl_switch2 上把一个接口分配给 VLAN

```
gl_switch2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
gl_switch2(config)#interface fastEthernet 0/7
gl_switch2(config-if)#switchport mode access
gl_switch2(config-if)#switchport access vlan 16
```

现在进入第 3 步，配置交换机到另外两台设备的中继。到 gl_switch 的中继是一条 802.1q 中继，而到 gameserver_2 和 gl_switch3 的则是 ISL 中继。在 gl_switch2 配置这些中继线路如例 2-52 所示。

例 2-52 在 gl_switch2 上地一个接口分配给 VLAN

```
Enter configuration commands, one per line. End with CNTL/Z.
gl_switch2(config)#int gig 0/1
gl_switch2(config-if)#switchport mode trunk
gl_switch2(config-if)#switchport trunk encapsulation dot1q
gl_switch2(config-if)#exit
```

(待续)

```
gl_switch2(config)#int fast 0/10
gl_switch2(config-if)#switchport mode trunk
gl_switch2(config-if)#switchport trunk encapsulation isl
gl_switch2(config-if)#exit
gl_switch2(config)#int fast 0/20
gl_switch2(config-if)#switchport mode trunk
gl_switch2(config-if)#switchport trunk encapsulation isl
```

交换机 gl_switch2 上的中继链路配置完成之后，到 gl_switch1 的中继开始工作。要确定中继的状态，可以在 gl_switch2 上用 **show vlan** 和 **show interface gigabitEthernet 0/1 switchport** 命令。显示结果中中继的状态应该是 up 和 trunking。同时也该看看 gl_switch1 上创建 VLAN 的情况，ping 一下 IP 地址 172.16.128.13 试试。例 2-53 就是 gl_switch2 上一条激活的中继状态示例。

例 2-53 在 gl_switch2 上验证 VLAN 和中继的工作状况

```
gl_switch2#show vlan
VLAN Name                Status    Ports
-----
1    default                active    Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                           Fa0/5, Fa0/6, Fa0/8, Fa0/9,
                                           Fa0/11, Fa0/12, Fa0/13, Fa0/14,
                                           Fa0/15, Fa0/16, Fa0/17, Fa0/18,
                                           Fa0/19, Fa0/21, Fa0/22, Fa0/23,
                                           Fa0/24, Fa0/25, Fa0/26, Fa0/27,
                                           Fa0/28, Fa0/29, Fa0/30, Fa0/31,
                                           Fa0/32, Fa0/33, Fa0/34, Fa0/35,
                                           Fa0/36, Fa0/37, Fa0/38, Fa0/39,
                                           Fa0/40, Fa0/41, Fa0/42, Fa0/43,
                                           Fa0/44, Fa0/45, Fa0/46, Fa0/47,
                                           Fa0/48, Gi0/2
16   VLAN0016                active    Fa0/7
17   VLAN0017                active
128  management              active
1002 fddi-default            active
1003 token-ring-default    active
1004 fddinet-default        active
1005 trnet-default          active

<<<text omitted>>>
gl_switch2#
gl_switch2#show interface gigabitEthernet 0/1 switchport
Name: Gi0/1
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))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1,16,17,128
Pruning VLANs Enabled: 2-1001

Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
gl_switch2#
```

配置完 gl_switch3 之后再回到第 4 步，即 STP 的调整上去。

交换机 gl_switch3 的基本管理功能配置和例 2-49 中的完全一样。在这个例子中，为它配置一个主机名，一个管理接口以及一个默认网关。由于 VTP 模式是透明模式，所以没有必要在这台交换机上配置 VLAN。另两台交换机的中继配置好之后 VLAN 信息就会发送过去。例 2-54 就是 gl_switch3 上 VTP 和中继的配置示例。

例 2-54 在 gl_switch3 上配置 VTP 客户端和 VLAN 中继

```
gl_switch3#vlan database
gl_switch3(vlan)#vtp domain funtime
Changing VTP domain name from Null to funtime
gl_switch3(vlan)#vtp client          ←Setting VTP client
Setting device to VTP CLIENT mode.
gl_switch3(vlan)#exit

gl_switch3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
gl_switch3(config)#interface fastEthernet 0/21    ←Trunk configuration
gl_switch3(config-if)#switchport mode trunk
gl_switch3(config-if)#switchport trunk encapsulation dot1q
gl_switch3(config-if)#exit
gl_switch3(config)#interface fastEthernet 0/20
gl_switch3(config-if)#switchport mode trunk
gl_switch3(config-if)#switchport trunk encapsulation isl
gl_switch3(config-if)#exit

gl_switch3(config)#interface fastEthernet 0/6      ←User port configuration
gl_switch3(config-if)#switchport mode access
gl_switch3(config-if)#switchport access vlan 17
```

用 **show vtp status** 命令验证这些配置，也可以查看 VLAN，确认 VLAN 信息确实是广播出去。例 2-55 列出了 gl-switch3 上状态命令的输出情况。此时，可以 **ping** 通相邻交换机的 IP 地址。

例 2-55 验证 gl_switch3 上的 VTP 状态

```
gl_switch3#show vtp status
VTP Version          : 2
Configuration Revision : 4
Maximum VLANs supported locally : 254
Number of existing VLANs : 8
VTP Operating Mode    : Client
VTP Domain Name       : funtime
VTP Pruning Mode      : Disabled
VTP V2 Mode           : Disabled
VTP Traps Generation  : Disabled
MD5 digest            : 0xC9 0xC8 0x2D 0xEE 0x8D 0xE1 0x46 0x97
Configuration last modified by 172.16.128.13 at 7-2-01 14:43:56
```

```
gl_switch3# show vlan
VLAN Name                Status    Ports
-----
1    default              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                         Fa0/5, Fa0/7, Fa0/8, Fa0/9,
                                         Fa0/10, Fa0/11, Fa0/12, Fa0/13,
                                         Fa0/14, Fa0/15, Fa0/16, Fa0/17,
```

(待续)

```

Fa0/18, Fa0/19, Fa0/22, Fa0/23,
Fa0/24, Fa0/25, Fa0/26, Fa0/27,
Fa0/28, Fa0/29, Fa0/30, Fa0/31,
Fa0/32, Fa0/33, Fa0/34, Fa0/35,
Fa0/36, Fa0/37, Fa0/38, Fa0/39,
Fa0/40, Fa0/41, Fa0/42, Fa0/43,
Fa0/44, Fa0/45, Fa0/46, Fa0/47,
Fa0/48, Gi0/1, Gi0/2

16 VLAN0016 active
17 VLAN0017 active Fa0/6
128 management active
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
-----
1 enet 100001 1500 - - - - - 0 0
16 enet 100016 1500 - - - - - 0 0
17 enet 100017 1500 - - - - - 0 0
128 enet 100128 1500 - - - - - 0 0
1002 fddi 101002 1500 - 0 - - - 0 0
1003 tr 101003 1500 - 0 - - srb 0 0
1004 fdnet 101004 1500 - - - - ieee - 0 0
1005 trnet 101005 1500 - - - - ibm - 0 0
gl_switch3#
gl_switch3#ping 172.16.128.13

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.128.13, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/7/8 ms
gl_switch3#ping 172.16.128.11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.128.11, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/3 ms
gl_switch3#
    
```

到现在为止，整个交换域就都具备了各自的功能。所有的交换机都可以 ping 通相互之间的管理接口。

现在配置两台路由器 glaccess_1 和 glaccess_2 来进行 IP 传输，这里的配置只是在以太网接口和环路接口下配置一个 IP 地址。这个网络模型中的路由选择协议是 EIGRP，其自治系统 (AS) 为 2001。EIGRP 必须在所有的路由器上配置。例 2-56 是 glaccess_1 的配置过程。路由器 glaccess_1 和 glaccess_2 的配置除了 IP 地址之外完全一样。

例 2-56 路由器 glaccess_1 的配置

```

hostname glaccess_1
!
interface Loopback20
 ip address 172.16.21.6 255.255.255.0
 no ip directed-broadcast
!
interface Ethernet0
 ip address 172.16.17.6 255.255.255.0
    
```

(待续)

```
no ip directed-broadcast
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!
```

要使 VLAN 相互之间能够通信，还必须配置一台路由器，它必须在每个 VLAN 中都有一个接口，或者具有 VLAN 中继的路由器。这个实验的模型中，路由器 gameserver_1 和 gameserver_2 不仅要配置来在 VLAN 之间进行路由，而且还要通过 HSRP 为网络提供一定的弹性。配置 VLAN 中继时，在以太接口上创建子接口，并为其分配 VLAN 和 VLAN 封装方式。每个需要进行路由的 VLAN 都需要一个子接口。例 2-57 就突出显示了两台路由器的 VLAN 中继配置部分的内容。到 gameserver_1 的中继是 802.1Q 的，而到 gameserver_2 则是 ISL。

例 2-57 路由器 gameserver_1 和 gameserver_2 的配置

```
hostname gameserver1
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.1
 encapsulation dot1Q 128
 ip address 172.16.128.14 255.255.255.0
!
interface FastEthernet0/0.2
 encapsulation dot1Q 16
 ip address 172.16.16.14 255.255.255.0
!
interface FastEthernet0/0.3
 encapsulation dot1Q 17
 ip address 172.16.17.14 255.255.255.0

hostname gameserver_2
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.1
 encapsulation isl 128
 ip address 172.16.128.10 255.255.255.0
 no ip redirects
!
interface FastEthernet0/0.2
 encapsulation isl 16
 ip address 172.16.16.10 255.255.255.0
 no ip redirects
!
interface FastEthernet0/0.3
 encapsulation isl 17
 ip address 172.16.17.10 255.255.255.0
```

(待续)

```
no ip directed-broadcast
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!
```

要使 VLAN 相互之间能够通信，还必须配置一台路由器，它必须在每个 VLAN 中都有一个接口，或者具有 VLAN 中继的路由器。这个实验的模型中，路由器 gameserver_1 和 gameserver_2 不仅要配置来在 VLAN 之间进行路由，而且还要通过 HSRP 为网络提供一定的弹性。配置 VLAN 中继时，在以太接口上创建子接口，并为其分配 VLAN 和 VLAN 封装方式。每个需要进行路由的 VLAN 都需要一个子接口。例 2-57 就突出显示了两台路由器的 VLAN 中继配置部分的内容。到 gameserver_1 的中继是 802.1Q 的，而到 gameserver_2 则是 ISL。

例 2-57 路由器 gameserver_1 和 gameserver_2 的配置

```
hostname gameserver1
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.1
 encapsulation dot1Q 128
 ip address 172.16.128.14 255.255.255.0
!
interface FastEthernet0/0.2
 encapsulation dot1Q 16
 ip address 172.16.16.14 255.255.255.0
!
interface FastEthernet0/0.3
 encapsulation dot1Q 17
 ip address 172.16.17.14 255.255.255.0
-----
hostname gameserver_2
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.1
 encapsulation isl 128
 ip address 172.16.128.10 255.255.255.0
 no ip redirects
!
interface FastEthernet0/0.2
 encapsulation isl 16
 ip address 172.16.16.10 255.255.255.0
 no ip redirects
!
interface FastEthernet0/0.3
 encapsulation isl 17
 ip address 172.16.17.10 255.255.255.0
```

(待续)


```
no ip redirects
!
```

HSRP 的配置要求 VLAN 128（子网 172.16.128.0/24）和 VLAN 17（子网 172.16.17.0/24）的主路由器都在 gameserver_1 上。VLAN 16（子网 172.16.16.0/24）的 HSRP 主路由器则是在 gameserver_2 上。为此，每个 VLAN 中都应该创建一个 HSRP 组，一共有 3 个组。给需要处在活动状态的接口分配的优先级为 101。关于 HSRP 的详细介绍，可以参考第 16 章“HSRP 的配置”。例 2-58 是路由器 gameserver1 和 gameserver2 上 HSRP 的配置。

例 2-58 gameserver1 和 gameserver2 上的 HSRP 配置

```
hostname gameserver1
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
!
interface FastEthernet0/0.1
encapsulation dot1Q 128
ip address 172.16.128.14 255.255.255.0
standby 1 priority 101 preempt
standby 1 ip 172.16.128.1
!
interface FastEthernet0/0.2
encapsulation dot1Q 16
ip address 172.16.16.14 255.255.255.0
standby 2 priority 95 preempt
standby 2 ip 172.16.16.1
!
interface FastEthernet0/0.3
encapsulation dot1Q 17
ip address 172.16.17.14 255.255.255.0
standby 3 priority 101 preempt
standby 3 ip 172.16.17.1
!

hostname gameserver_2
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
!
interface FastEthernet0/0.1
encapsulation isl 128
ip address 172.16.128.10 255.255.255.0
no ip redirects
standby 1 priority 95 preempt
standby 1 ip 172.16.128.1
!
interface FastEthernet0/0.2
encapsulation isl 16
ip address 172.16.16.10 255.255.255.0
no ip redirects
standby 2 priority 101 preempt
standby 2 ip 172.16.16.1
```

（待续）

```

!
interface FastEthernet0/0.3
 encapsulation isl 17
 ip address 172.16.17.10 255.255.255.0
 no ip redirects
 standby 3 priority 95 preempt
 standby 3 ip 172.16.17.1
!

```

现在，网络完全具备了它应有的功能，而且还具有一定的冗余度。所有的 IP 地址都是互通的。关于这一点，可以用 **ping** 命令，再断开 **gameserver_1** 或 **gameserver_2** 来测试。这样网络照样是互通的。例 2-59 就是 **glaccess_1** 的路由表的情况，可以看到冗余路由显示。

例 2-59 glaccess_1 的 IP 路由表

```

glaccess_1#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

    172.16.0.0/24 is subnetted, 5 subnets
D       172.16.128.0 [90/284160] via 172.16.17.10, 01:04:35, Ethernet0
          [90/284160] via 172.16.17.14, 01:04:35, Ethernet0
D       172.16.20.0 [90/412160] via 172.16.17.10, 01:04:35, Ethernet0
          [90/412160] via 172.16.17.14, 01:04:35, Ethernet0
C       172.16.21.0 is directly connected, Loopback20
D       172.16.16.0 [90/284160] via 172.16.17.10, 01:04:35, Ethernet0
          [90/284160] via 172.16.17.14, 01:04:35, Ethernet0
C       172.16.17.0 is directly connected, Ethernet0
glaccess_1#

```

这个实验的最后一步是生成树根的配置和 IP 访问的控制。用 **set spantree root** 命令可以在 **gl_switch1** 上设置 STP 根。通过查看 **gl_switch3** 的当前 STP 拓扑表可以看出，它就是当前的 STP 根。例 2-60 显示了 **gl_switch3** 的 STP 拓扑表。

例 2-60 gl_switch3 的 STP 拓扑

```

gl_switch3#show spanning-tree vlan 128

Spanning tree 128 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0004.275e.f0c1
Configured hello time 2, max age 20, forward delay 15
We are the root of the spanning tree
Topology change flag not set, detected flag not set, changes 2
Times: hold 1, topology change 35, notification 2
       hello 2, max age 20, forward delay 15
Timers: hello 1, topology change 0, notification 0

Interface Fa0/20 (port 34) in Spanning tree 128 is FORWARDING
      Port path cost 19, Port priority 128

```

(待续)

```

Designated root has priority 32768, address 0004.275e.f0c1
Designated bridge has priority 32768, address 0004.275e.f0c1
Designated port is 34, path cost 0
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 1376, received 0

Interface Fa0/21 (port 35) in Spanning tree 128 is FORWARDING
Port path cost 19, Port priority 128
Designated root has priority 32768, address 0004.275e.f0c1
Designated bridge has priority 32768, address 0004.275e.f0c1
Designated port is 35, path cost 0
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 1392, received 2
gl_switch3#

```

如果想设置 gl_switch1 交换机只是 VLAN 17 和 128 的 STP 根，可以参考例 2-61 的配置过程。

例 2-61 设置 gl_switch1 为 VLAN 17 和 VLAN 128 的 STP 根

```

gl_switch1(enable) set spantree root 17,128
VLANs 17,128 bridge priority set to 8192.
VLANs 17,128 bridge max aging time set to 20.
VLANs 17,128 bridge hello time set to 2.
VLANs 17,128 bridge forward delay set to 15.
Switch is now the root switch for active VLANs 17,128.
gl_switch1(enable)

```

如果像例 2-62 那样查看 gl_switch3 上的 STP 的情况，可以看到该交换机已经不再是 VLAN 128 的根了。请注意优先级已经发生了改变，现在 gl_switch1 已经是它的 STP 根了。

例 2-62 VLAN 128 的 STP

```

gl_switch3#show spanning-tree vlan 128

Spanning tree 128 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0004.275e.f0c1
Configured hello time 2, max age 20, forward delay 15
Current root has priority 8192, address 0030.1976.4d7f
Root port is 35, cost of root path is 19
Topology change flag not set, detected flag not set, changes 5
Times: hold 1, topology change 35, notification 2
hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0

Interface Fa0/20 (port 34) in Spanning tree 128 is BLOCKING
Port path cost 19, Port priority 128
Designated root has priority 8192, address 0030.1976.4d7f
Designated bridge has priority 32768, address 0004.275e.f5c3
Designated port is 34, path cost 4
Timers: message age 3, forward delay 0, hold 0
BPDU: sent 4762, received 97

Interface Fa0/21 (port 35) in Spanning tree 128 is FORWARDING
Port path cost 19, Port priority 128
Designated root has priority 8192, address 0030.1976.4d7f

```

(待续)

```

Designated bridge has priority 8192, address 0030.1976.4d7f
Designated port is 84, path cost 0
Timers: message age 3, forward delay 0, hold 0
BPDU: sent 4777, received 98
gl_switch3#

```

在 gl_switch2 上设置 VLAN 16 的 STP 根，可以用下面这条全局命令：

```
gl_switch2 (config) #spanning-tree vlan 16 priority 100
```

如果不指定 VLAN，所有 VLAN 的优先级都会改为 100。

实验的最后一部分是限制仅有子网 172.16.17.0/24 上的设备才能通过 Telnet 访问 gl_switch1，这可以通过 IP 允许列表的使用来实现。IP 允许列表输入完毕之后还必须启动它才能生效。例 2-63 就是 gl_switch1 上 IP 允许列表的配置情况。

例 2-63 IP 访问列表的使用

```

gl_switch1 (enable) set ip permit 172.16.17.0 255.255.255.0
172.16.17.0 with mask 255.255.255.0 added to IP permit list.
gl_switch1 (enable) set ip permit enable
IP permit list enabled.
gl_switch1 (enable)

```

实验的可选部分是安全性的设置。通过启动端口安全性功能，如果端口检测到的 MAC 地址是端口所不允许的，那么端口就会进入停止状态。如果在端口安全性启动了的情况下下一台工作站接入这个端口，端口会自动记录其 MAC 地址并为该地址把端口保护起来。要配置某个特定的地址，将这个 MAC 地址加到 **set port security** 命令中去即可。例 2-64 就是启动端口安全性的例子。

例 2-64 启动端口安全性

```

gl_switch1 (enable) set port security 2/24 enable 00-00-86-3c-3b-41
Port 2/24 port security enabled with 00-00-86-3c-3b-41 as the secure mac address

Trunking disabled for Port 2/24 due to Security Mode
gl_switch1 (enable)

```

如果其他工作站或其他设备接入到端口 2/24 上，这个端口会自动关闭。例 2-65 突出显示了一台未经授权的设备接入到端口 2/24 后的端口状态。

例 2-65 端口安全性

```

gl_switch1 (enable) show port 2/24

```

Port	Name	Status	Vlan	Level	Duplex	Speed	Type
2/24		shutdown	1	normal	auto	auto	10/100BaseTX

Port	Security	Secure-Addr	Last-Addr	Shutdown	Trap	IfIndex
2/24	enabled	00-00-86-3c-3b-41	00-60-5c-f3-5e-65	Yes	disabled	34

Port	Status	Channel mode	Channel status	Neighbor device	Neighbor port
2/24	shutdown	auto	not channel		

<<<text omitted>>>

2.13 实验 8：用 Catalyst 3920 配置令牌环交换网络

——第 1 部分

2.13.1 实验说明

令牌环局域网是现代数据中心应用较广泛的网络类型。拥有 SNA 大型机的多数数据中心仍然在使用令牌环网络。这个实验就是要给读者一些实际配置令牌环交换网络的经验。在整本书的剩余部分里，我们鼓励读者如果可能尽量采用令牌环交换机，以增强对实验中的路由选择协议与功能套件的理解。

2.13.2 实验内容

还是那个在全国范围内为娱乐中心提供高速以太 LAN 的公司，Game LANs, Inc.，它们也提供令牌环局域网。Game LANs, Inc. 的客户之一需要在令牌环 PS/2 PC 上玩 DOOM 的组。现在他们有两个令牌环网络，两个 IP 子网，想把这两个网络移植到交换环境中去。

我们的任务是以下面的参数为标准配置一个 Game LAN 网络：

- 如图 2-41 配置一个令牌环交换网络。
- 配置 VTP 域 “rings” 并将交换机设为一台 VTP 服务器。
- 创建两个 VLAN，一个使用子网 128.100.1.0/24，另一个为 128.100.2.0/24。
- 将端口 3-24 分配给包含子网 128.100.2.0/24 的 VLAN。
- 配置网络的 IP 路由，以实现所有工作站、路由器和交换机之间的完全 IP 连通性。用 RIP 作为整个网络的路由选择协议。

2.13.3 实验目的

- 如图 2-41 配置一个令牌环交换网络。
- 配置 VTP 域 “rings”，并把交换机设置为 VTP 服务器。创建两个 VLAN，一个给子网 128.100.1.0/24，另一个给 128.100.2.0/24。实际的配置可能需要更多的 VLAN。
- 将端口 3-24 分配给包含子网 128.100.2.0/24 的 VLAN。
- 如图所示配置交换机的 IP 地址。
- 以 RIP 为整个网络的路由选择协议。确保所有工作站、路由器和交换机之间的完全 IP 互连正常。

2.13.4 所需设备

- 两台具有令牌环接口的 Cisco 路由器，其中一台还必须具有两个令牌环接口。

Game LAN 公司的
交换式以太网
VTP 域 = rings
RIP

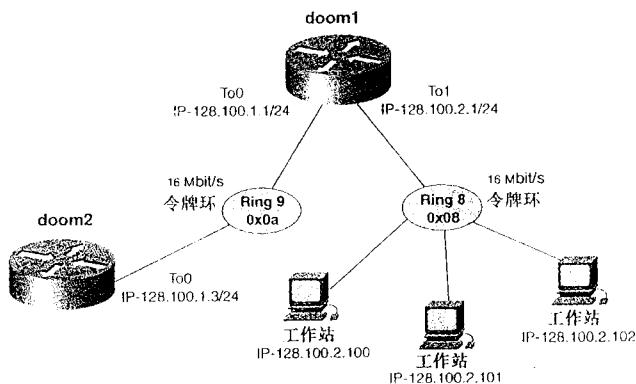
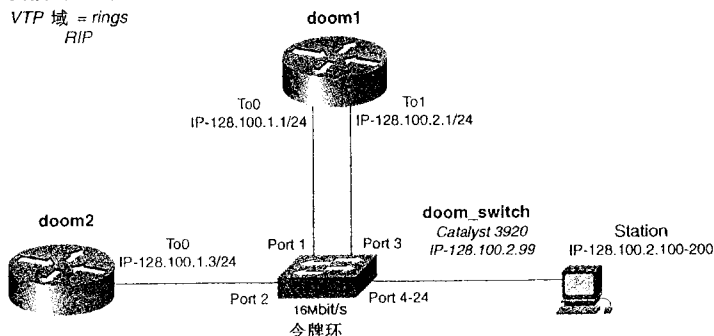


图 2-41 Game LAN 的令牌环网络

- 一台 Catalyst 3920 型令牌环交换机。
- 图中的工作站是为了做额外测试用的，该实验并不需要。

2.13.5 物理设计与实验准备

- 如图 2-41 连接好所有的交换机和路由器。
- 工作站是可选的，但是推荐使用它来进一步测试网络的功能。

2.14 实验 8：用 Catalyst 3920 配置令牌环交换网络 ——第 2 部分

2.14.1 实验步骤

就可以按照配置令牌环交换的 6 个步骤来顺序进行了：

- 第 1 步 规划好 TrBRF、TrCRF、令牌环号、网桥号以及 VLAN。
- 第 2 步 配置 VTP。
- 第 3 步 配置 TrBRF VLAN 并为每个 TrBRF 分配一个网桥号。
- 第 4 步 配置 TrCRF VLAN 并为其分配一个父 TrBRF 和一个令牌环号 (可选)。
- 第 5 步 为 TrCRFs 分配端口。
- 第 6 步 配置交换机管理功能。

图 2-42 是需要创建的 VLAN 情况，包括两个 TrBRF VLAN 和两个 TrCRF VLAN。端口 3 至 24 在 TrCRF crf-ring8 中，其父 TrBRF 是 brf8。端口 1 和 2 在 TrCRF crf-ring9 中，其父 TrBRF 则是 brf9。

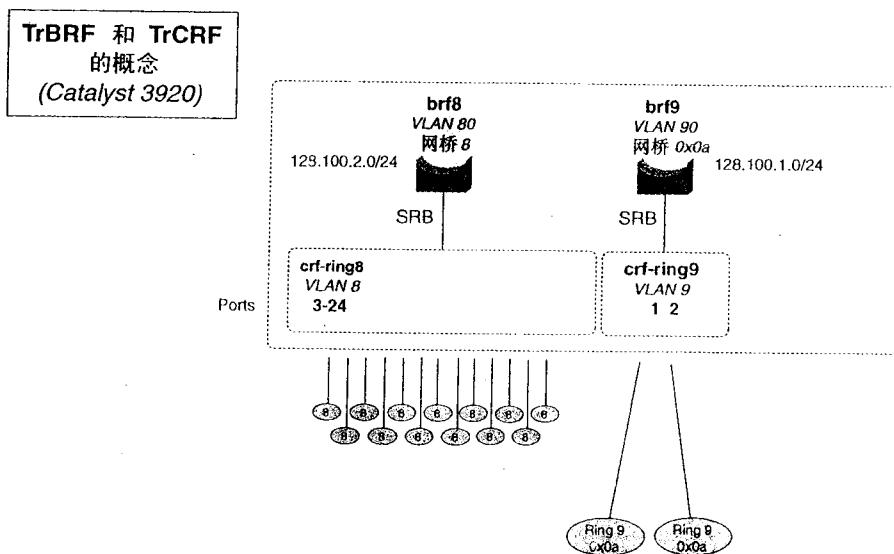
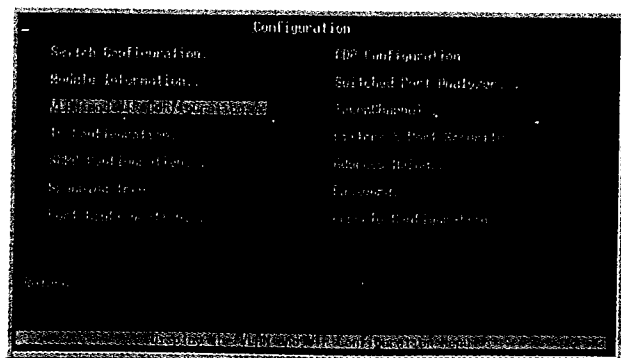


图 2-42 TrBRF 和 TrCRF VLAN 的概念性设计

交换机的配置完全是在配置菜单界面上进行的，第 1 步是配置 VTP 域，在主菜单界面中选择 VLAN 与 VTP 配置选项，这是配置 VTP 和 VLAN 的地方。图 2-43 就是配置菜单此时的示例。



在 VLAN 与 VTP 配置菜单界面中，选择 VTP 管理配置选项，将 VTP 域配置为一台服务器，为它分配一个环名称。图 2-44 是这一界面的配置情况。

第3步是配置 TrBRF。在 VLAN 与 VTP 配置界面中，进入 VTP VLAN 配置选项，选择 Add 来加入一个新的 VLAN。交换机会要求输入一个 VLAN 名。第一个要创建的 VLAN 是 brf8，VLAN 80。然后交换机会要求确定 VLAN 是一个 TrBRF 还是 TrCRF。选择 TrBRF 就会进入 VLAN 配置菜单界面中。在这一界面里，为此 VLAN 命名为 brf8 并输入一个网桥号 8。记住网桥号的输入应该是 16 进制的。网桥号 9 应该是输入 0x08。图 2-45 和 2-46 就是 TrBRF 的 VLAN 配置示例。

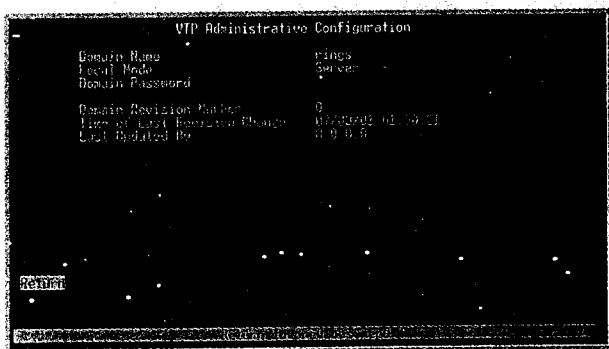


图 2-44 VTP 管理界面

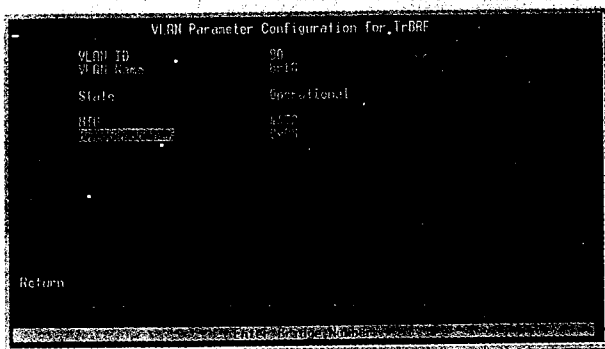
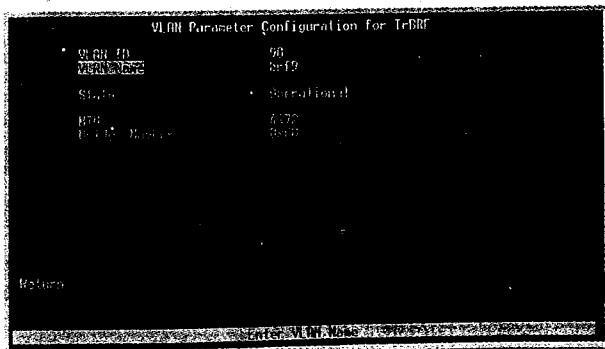


图 2-45 详细的 TrBRF VLAN 配置示例



下一步是配置 TrCRF VLAN，其创建和 TrBRF VLAN 在同一菜单下。交换机提示确定 VLAN 类型时，选择 TrCRF 而不是 TrBRF 即可。创建 TrCRF VLAN 时，不要把 VLAN ID 搞混了。TrCRF 和 TrBRF 的 VLAN ID 应该各自惟一。TrCRF 通过把 TrBRF 设置为自己的父 TrBRF 来与之建立联系，而不是通过 VLAN ID 来实现。

图 2-47 和 2-48 是 TrCRF VLAN 的配置情况。

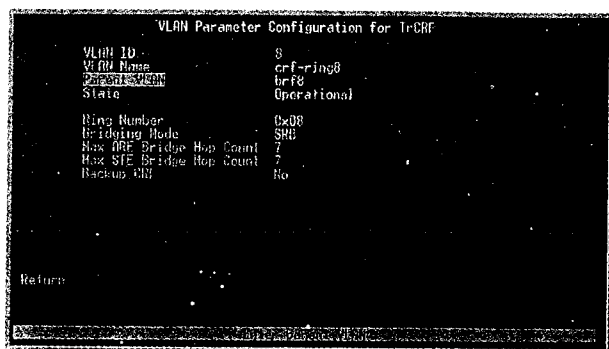


图 2-47 详细的 TrCRF VLAN 配置示例

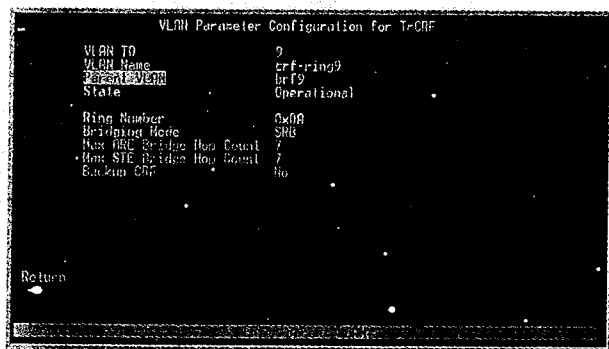


图 2-48 详细的 TrCRF VLAN 配置示例

图 2-49 则是新建 VLAN 的 VTP VLAN 配置界面的示例。

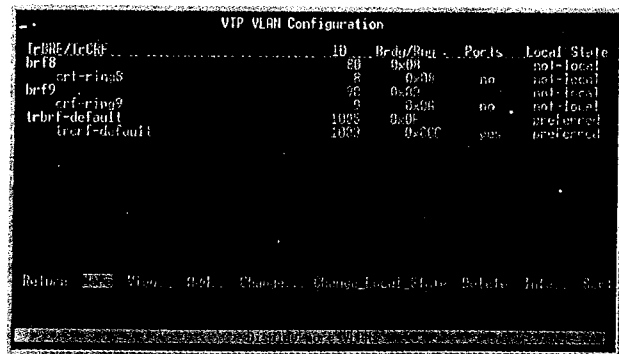


图 2-49 VLAN 列表

第5步是为新建的 TrCRF 分配端口。端口 3 到 24 分配给 TrCRF crf-ring8，而端口 1 和 2 则分配给 TrCRF crf-ring9。在 VLAN VTP 配置界面中，进入本地 VLAN 端口配置选项。在这里，选中需要进行修改的端口之后，交换机会要求确定端口要与哪个 TrCRF 联系在一起。图 2-50 是这一过程的示例。

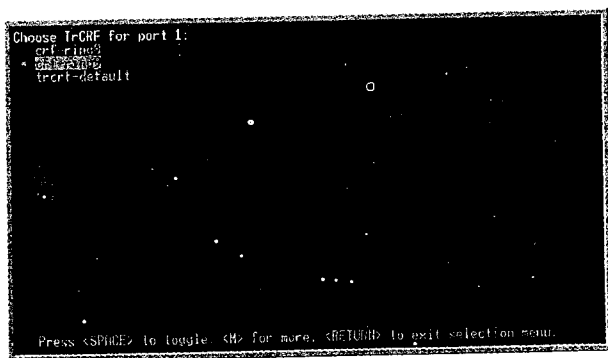


图 2-50 端口配置

最后一步是 IP 地址的配置。在主配置界面中，选择 IP Configuration 对 IP 地址进行配置。选中这一选项之后，交换机会要求给出 IP 地址要赋予的 TrBRF。在这个例子中，IP 地址是 128.100.2.99，因此将它分配给 TrBRF brf8。在这个菜单里时，输入 IP 信息，并将 128.100.2.1 作为默认网关。图 2-51 是 IP 配置界面的示例。

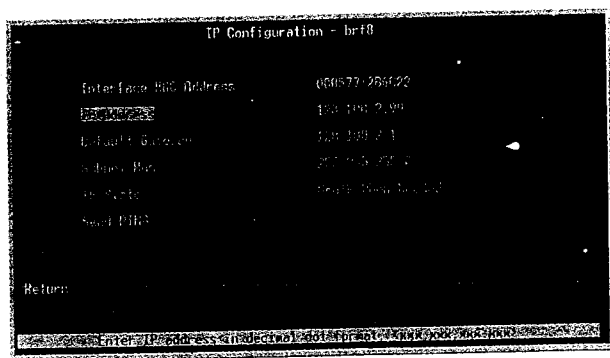


图 2-51 IP 配置示例

只有所属 TrBRF 成为默认的管理 TrBRF 接口之后 IP 接口才会开始工作，这是在 SNMP 配置界面中启动的。在 SNMP 配置界面中，把默认的管理 VLAN 改为 TrBRF brf8。

现在交换机就完全具备应有的功能了，也就可以配置路由器了。这个实验中的路由器部分是非常容易的。路由器 doom1 要有两个令牌环接口，它要运行 RIP 作为路由选择协议，而 doom2 只需要一个令牌环接口以做 IP 传输之用。例 2-66 就是 doom1 和 doom2 的相关配置部分。

路由器也配置好之后，对所有的路由器和交换机接口进行 ping 测试。路由器 doom2 的路由表也说明 RIP 已经报告子网 128.100.2.0/24。

例 2-66 路由器 doom1 和 doom2 的配置

```
hostname doom1
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 128.100.1.1 255.255.255.0
 no ip directed-broadcast
 ring-speed 16
!
interface TokenRing1
 ip address 128.100.2.1 255.255.255.0
 no ip directed-broadcast
 ring-speed 16
!
router rip
 network 128.100.0.0
!
```

```
!
! hostname doom2
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 128.100.1.3 255.255.255.0
 ring-speed 16
!
interface BRI0
 no ip address
 shutdown
!
router rip
 network 128.100.0.0
!
```

第3部分

采用广域网互连局域网

第3章 WAN协议与技术：高级数据链路控制（HDLC）

第4章 WAN协议与技术：点对点协议（PPP）

第5章 WAN协议与技术：帧中继

第6章 WAN协议与技术：通过多协议传输语音

第7章 WAN协议与技术：综合业务数字网（ISDN）

第8章 WAN协议与技术：异步传输模式（ATM）

第 3 章

WAN 协议与技术： 高级数据链路控制 (HDLC)

广域网 (WAN) 的出现，源于人们对跨地域进行数据共享的需要。WAN 的简单定义就是一个跨越巨大地理区域的网络，目的是将局域网 (LAN) 连接在一起并且在 LAN 之间进行数据传输。在本书的学习中，WAN 的作用基本如上所述。

随着时间的推移，WAN 协议在不停地进行改进。最初，WAN 协议主要是侧重于其纠错功能，运行在老式、不可靠的铜线上。现在，WAN 协议已经能够在铜线和光纤上提供高速可靠的数据传输。尽管近年来 WAN 发展迅速，但仍然是现代互联网络中速度最慢、成本最高的部分。

WAN 协议主要是工作在 OSI 模型的下面的三层上。X.25 分组协议 (PLP) 是工作在第 3 层上的 WAN 协议之一，但大多数的 WAN 协议工作在第二层。表 3-1 列出了一些常用的 WAN 协议及其对应的 OSI 参考模型层。

表 3-1 WAN 协议及其相应的 OSI 层

OSI 层	WAN 协议					
网络层	X.25 PLP					
数据链路层	Frame Relay	PPP	HDLC LAPB	X.25	SDLC	ATM-AAL
物理层	ISDN-B ISDN-D ISDN-H H11,H12	EIA/TIA-232 EIA/TIA-449 V.24,V.35,HSSI G703,EIA-530				DS-1,DS-3, SONET

在实验中，使用 Cisco 2500 和 2600 系列路由器，能够方便地建立 HDLC、PPP、帧中继和 X.25 的模型。本章的学习从这些协议开始以 ISDN 和 ATM 结束。ISDN 和 ATM 模型的建立需要使用特定的路由器和交换机。

3.1 HDLC 的兼容性和简易性

高级数据链路控制 (HDLC) 是一种高效 WAN 协议，是基于 IBM 的同步数据链路控制 (SDLC) 制定的。Cisco 的 HDLC 从 ISO 3309 协议框架发展而来。HDLC 有很多不同的协议框架形式，Cisco 版与任何其他供应商 (如 Unisys 的 HDLC 或 ISO 3309 协议框架) 的都不兼容。尽管这些版本都很相似，但是它们之间互不兼容。

Cisco 版的 HDLC 有如下特点：

- 与其他供应商不兼容。
- 快速，高效。
- 支持 keepalive 机制。
- 支持串行链路地址解析协议 (SLARP)。
- 支持 STAC 压缩。

HDLC 是默认的串口封装格式，因此没有出现在配置文件里，从 **show interface** 命令的输出结果中可以看到。

HDLC 需要的控制开销很小，包括一个起始和一个结束标志字段，一个可变的地址字段，一个控制字段和一个信息长度字段。虽然这些字段的大小都可变，但是变化的幅度也只有 1 到 4 个字节的长度。整个数据帧的控制开销仅仅从 7 个字节到 12 个字节。这就是 HDLC 的高效性。

HDLC 利用了 keepalive 机制来验证连接的完整性。连接的 DCE 一端会发送序列号到 DTE 端，而 DTE 又会按顺序将此序列号返回，这样路由器就会知道它收到从 DTE 返回来的序列号是否一致。如果该序列号连续丢失 3 次，路由器就会关闭此链路。例 3-1 中可以看到链路由于没有 keepalive 信号返回而断开。这些信息可以通过 **debug serial interface** 命令看到。

例 3-1 HDLC 帧上没有收到的 keepalive

```
06:35:59: %LINK-3-UPDOWN: Interface Serial1, changed state to up
06:36:00: Serial1: HDLC myseq 0, mineseen 0, yourseen 0, line up    ←Keepalive(KA)
06:36:00: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state
to up
06:36:10: HD(1): Deasserting DSR, CTS and DCD
06:36:10: HD(1): Reset from 0x3057C8C
06:36:10: HD(1): Asserting DSR
06:36:10: HD(1): Asserting DCD and CTS
06:36:10: HD(1): Deasserting LTST
06:36:10: HD(1): Asserting DTR and RTS
06:36:10: Serial1: HDLC myseq 1, mineseen 0, yourseen 0, line up    ←KA not received
06:36:18: HD(0): New serial state = 0x0115
```

(待续)

```
06:36:18: HD(1): got an interrupt state = 0x8055
06:36:18: HD(1): New serial state = 0x005F

06:36:18: HD(1): DTR is up.
06:36:20: HD(1): Deasserting DSR, CTS and DCD
06:36:20: HD(1): Reset from 0x3057C8C
06:36:20: HD(1): Asserting DSR
06:36:20: HD(1): Asserting DCD and CTS
06:36:20: HD(1): Deasserting LTST
06:36:20: HD(1): Asserting DTR and RTS
06:36:20: Serial1: HDLC myseq 2, mineseen 0, yourseen 0, line down ← Still no KAs
06:36:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state
to down
06:36:28: HD(0): New serial state = 0x0115

06:36:28: HD(1): got an interrupt state = 0x8055
06:36:28: HD(1): New serial state = 0x005F

06:36:28: HD(1): DTR is up.
06:36:30: Serial1: HDLC myseq 3, mineseen 0, yourseen 0, line down
06:36:40: Serial1: HDLC myseq 4, mineseen 0, yourseen 0, line down
06:36:50: Serial1: HDLC myseq 5, mineseen 0, yourseen 0, line down
06:36:51: Serial1: attempting to restart
06:36:51: HD(1): Deasserting DSR, CTS and DCD
06:36:51: HD(1): Reset from 0x3057C8C
```

例 3-1 中，其序列号或 myseq 字段没有递增，与 mineseen 字段也不匹配。在正常的连接中，keepalive 标志会递增并传递给下游路由器。下游的路由器收到这些标志后会再将其返回。连接性可以从发出该标志的路由器的 mineseen 字段中判断。如果 myseq 和 mineseen 字段的值相差大于 3，表明 keepalive 已经丢失 3 次以上，路由器会将该链路重新初始化。另一台路由器的 keepalive 在其 yourseen 字段里，功能是一样的。例 3-2 解决了此问题，请查看调试的输出结果。

例 3-2 正常工作的 HDLC 链路的调试记录

```
06:49:30: Serial1: HDLC myseq 81, mineseen 0, yourseen 0, line down
06:49:31: %SYS-5-CONFIG_I: Configured from console by console
06:49:40: Serial1: HDLC myseq 82, mineseen 82*, yourseen 82,
line up ← First KA seen
06:49:41: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state
to up
06:49:50: Serial1: HDLC myseq 83, mineseen 83*, yourseen 83, line up
06:50:00: Serial1: HDLC myseq 84, mineseen 84*, yourseen 84, line up
06:50:10: Serial1: HDLC myseq 85, mineseen 85*, yourseen 85, line up
06:50:20: Serial1: HDLC myseq 86, mineseen 86*, yourseen 86, line up
06:50:31: Serial1: HDLC myseq 87, mineseen 87*, yourseen 87, line up
06:50:41: Serial1: HDLC myseq 88, mineseen 88*, yourseen 88, line up
r3#
06:50:51: Serial1: HDLC myseq 89, mineseen 89*, yourseen 89, line up
```

只有链路的第一层正常工作时，交换 keepalive 机制才有效。如果 DCD=up、DSR=up、DTR=up、RTS=up 和 CTS=up 这些项中有一项状态未激活，keepalive 就不可能工作。

HDLC 还支持 SLARP，使得串行线路在自动安装过程中能够获取动态映射 IP 地址。Cisco 的自动安装过程利用了 SLARP 技术的这些功能。

Cisco 的 HDLC 还通过使用 STAC 压缩算法支持有效载荷压缩。STAC Electronic 开发的 STAC 压缩技术使用的是 Lempel-Ziv 压缩算法。该算法能提供良好的压缩率但占用较多 CPU

处理时间来对有效载荷数据帧进行压缩。STAC 压缩也适用于 LAPB、X.25 和帧中继。

注释 作为设计原则，在 Cisco 路由器之间进行快速可靠的简单配置，推荐使用 HDLC 封装。

3.1.1 HDLC 的设置

HDLC 是串口的默认数据封装格式。其设置可以分成 3 个步骤：

第 1 步 用 **encapsulation hdlc** 命令在接口模式下配置数据封装；

第 2 步 用 **clock rate clock_speed** 命令对连接的 DCE 端进行设置。这种方法只能用在背对背电缆连接的路由器上。使用 CSU/DSU 设备时，CSU 就是其 DCE 设备；

第 3 步 (可选) 在链路两端使用 **compression stac** 命令配置压缩。

如果想将帧中继或其他数据封装格式切换成为 HDLC，可以在接口配置模式下使用 **encapsulation hdlc** 命令来完成，也可以用 **no encapsulation PPP** 这种方式，使用 **encapsulation** 命令的 **no** 形式来进行，它将返回默认的数据封装类型，也就是 HDLC。如果使用背对背串行电缆，或者接口是用 DCE 电缆连接的，就需要在连接 DCE 线缆的那台路由器的接口模式下输入 **clock rate xxxx** 命令。如果要使用 STAC 压缩，可以在接口配置模式使用 **compression stac** 命令，注意在连接的两个路由器上都必须进行数据压缩。在运用数据压缩时要考虑 CPU 的系统开销问题。

图 3-1 所示为一个基本的 HDLC 网络结构。路由器 espn 有两个串行接口，一个连接到路由器 atlanta，另一个接到路由器 bristol_u。配置 HDLC 时可以遵循上面提到的步骤进行，然后在串行连接的两个端口上用接口命令 **encapsulation hdlc**。对于链路的 DCE 端来说，用户需要在串口上用 **clock rate** 命令设置其通信速率。

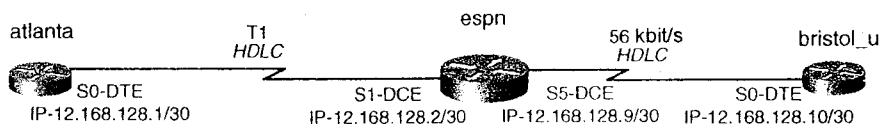


图 3-1 基本 HDLC 网络

例 3-3 给出了一个在 espn 路由器上配置 HDLC 的例子。假设该网络的路由选择协议为 EIGRP。

例 3-3 espn 路由器的 HDLC 设置

```

espn(config)#interface s1
espn(config-if)#encapsulation hdlc
espn(config-if)#clock rate 2000000
espn(config-if)#ip address 12.168.128.2 255.255.255.252
espn(config-if)#exit
espn(config)#
espn(config)#interface s5
espn(config-if)#encapsulation hdlc
espn(config-if)#clock rate 56000
espn(config-if)#ip address 12.168.128.9 255.255.255.252
    
```


例 3-4 则是路由器 bristol_u 上的 HDLC 配置情况。

例 3-4 路由器 bristol_u 上的 HDLC 配置

```
bristol_u(config)#interface s0
bristol_u(config-if)#encapsulation hdlc
bristol_u(config-if)#ip address 12.168.128.10 255.255.255.252
bristol_u(config-if)#exit
```

从上面的例子可以看出，HDLC 的设置非常简单，多数情况下甚至不用配置数据的封装格式而仅仅设置一个 IP 地址就可以了。

3.1.2 HDLC 的 “Big show” 和 “Big D”

HDLC 的 **show** 和 **debug** 命令也可以在大多数串口上使用。HDLC 是一个简单的协议，因此，其 **show** 和 **debug** 命令也很有限，但已经足以满足需要。

HDLC 的 “big show” 和 “big D” 命令包括：**show interface serial_interface** 和 **show controllers serial_interface**。下面对它们进行讨论：

3.1.3 show interface serial_interface 命令

show interface serial_interface 命令用于显示一个接口的工作状态，其主要字段包括接口线路[up/down]和线路协议[up/down]。这两个字段分别代表 OSI 网络层的第一和第二层。此外，该命令还能显示数据封装格式以及 keepalive 值。DCD、DSR、DTR、RTS 和 CTS 这些信号都应该是 “up”。接口复位字段说明了连接已经复位的次数。其他一些字段还能显示数据丢失、数据帧、操作异常中止以及 CRC 错误等信息。例 3-5 就是该命令的输出显示。

例 3-5 show interface 命令的输出显示

```
router#show interface serial 5
Serial5 is up, line protocol is up
Hardware is CD2430 in sync mode
Internet address is 12.168.128.9/30
MTU 1500 bytes, BW 115 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 00:00:02, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/1/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
3870 packets input, 206261 bytes, 0 no buffer
Received 1524 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
3907 packets output, 228500 bytes, 0 underruns
0 output errors, 0 collisions, 44 interface resets
```

(待续)

```
0 output buffer failures, 0 output buffers swapped out
24 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
router#
```

3.1.4 show controllers serial_interface 命令

用 **show controllers serial_interface** 命令能够获取串行控制器的物理层信息。该信息有助于验证电缆类型以及确定电缆是 DCE 还是 DTE。例 3-6 为该命令的输出结果。

例 3-6 show interface 命令的输出结果

```
espn#show controllers serial 1
HD unit 1, idb = 0xD7A28, driver structure at 0xDC7A8
buffer size 1524 HD unit 1, V.35 DCE cable, clockrate 2000000
cpb = 0x43, eda = 0x2140, cda = 0x2000
RX ring with 16 entries at 0x432000
00 bd_ptr=0x2000 pak=0x0DF384 ds=0x43C468 status=80 pak_size=0
01 bd_ptr=0x2014 pak=0x0DF1B4 ds=0x43BDB0 status=80 pak_size=0
02 bd_ptr=0x2028 pak=0x0DEFE4 ds=0x43B6F8 status=80 pak_size=0
03 bd_ptr=0x203C pak=0x0DEE14 ds=0x43B040 status=80 pak_size=0
04 bd_ptr=0x2050 pak=0x0DEC44 ds=0x43A988 status=80 pak_size=0
05 bd_ptr=0x2064 pak=0x0DEA74 ds=0x43A2D0 status=80 pak_size=0
06 bd_ptr=0x2078 pak=0x0DE8A4 ds=0x439C18 status=80 pak_size=0
07 bd_ptr=0x208C pak=0x0DE6D4 ds=0x439560 status=80 pak_size=0
08 bd_ptr=0x20A0 pak=0x0DE504 ds=0x438EA8 status=80 pak_size=0
09 bd_ptr=0x20B4 pak=0x0DE334 ds=0x4387F0 status=80 pak_size=0
10 bd_ptr=0x20C8 pak=0x0DE164 ds=0x438138 status=80 pak_size=0
11 bd_ptr=0x20DC pak=0x0DDF94 ds=0x437A80 status=80 pak_size=0
12 bd_ptr=0x20F0 pak=0x0DDDC4 ds=0x4373C8 status=80 pak_size=0
13 bd_ptr=0x2104 pak=0x0DDBF4 ds=0x436D10 status=80 pak_size=0
14 bd_ptr=0x2118 pak=0x0DDA24 ds=0x436658 status=80 pak_size=0
15 bd_ptr=0x212C pak=0x0DD854 ds=0x435FA0 status=80 pak_size=0
16 bd_ptr=0x2140 pak=0x0DD684 ds=0x4356E8 status=80 pak_size=0
cpb = 0x43, eda = 0x2800, cda = 0x2800
TX ring with 2 entries at 0x432800
00 bd_ptr=0x2800 pak=0x000000 ds=0x000000 status=80 pak_size=0
01 bd_ptr=0x2814 pak=0x000000 ds=0x000000 status=80 pak_size=0
02 bd_ptr=0x2828 pak=0x000000 ds=0x000000 status=80 pak_size=0
0 missed datagrams, 0 overruns
0 bad datagram encapsulations, 0 memory errors
0 transmitter underruns
0 residual bit errors

espn#
```

3.1.5 debug serial interface 命令

请记住，在进行调试之前，一定要在设置中设定 **logging buffered 10000**。这样可以避免大量的数据占用控制台，使其无法正常工作。

我们在上述的实验中加入故障内容以便于验证这些命令的作用。

每次解决问题的时候最好是集中在一个方面。先针对路由器 **espn**，运行 **show interface**

serial 5 命令，其结果如例 3-7 所示。

例 3-7 show interface 命令的输出结果

```

espn#show interface serial 5
Serial5 is up, line protocol is up
Hardware is CD2430 in sync mode
Internet address is 12.168.128.9/30
MTU 1500 bytes, BW 115 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 00:00:02, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations 0/1/256 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
3870 packets input, 206261 bytes, 0 no buffer
Received 1524 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
3907 packets output, 228500 bytes, 0 underruns
0 output errors, 0 collisions, 174 interface resets
0 output buffer failures, 0 output buffers swapped out
24 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
espn#
    
```

注意这里显示的线路和协议都是 up，这清楚地表明了第 1 层和第 2 层工作正常。所有的载波信号显示也都为 up，这是第 1 层正在工作正常的另一种标志。然而，显示的接口复位次数和载波转换次数有点令人置疑，为定位故障可以先用 **clear counters** 命令将所有的计数器清零，再用 **show interface** 命令来确定计数器是否在递增。执行该命令之后，相当的一段时间之内，计数器都在增长，表明可能有某个连接的数据封装有问题。而从该链路对路由器 bristol_u 表现出来的物理特性来看，此链路是在正常工作的。用 **show controllers** 命令能够确认第 1 层在正常工作。所以问题需要深入些。有必要进行调试，如例 3-8 所示，运行 **debug serial interface** 命令。在上面的例子中，CIRRUS（5）代表串口 5 而 HD（1）则代表串口 1。HD 和 CIRRUS 是这些端口的控制卡。

例 3-8 debug serial interface 命令的输出结果

```

CIRRUS(5): Asserting DCD                                ←Link asserts DTR
Serial5: HDLC myseq 11, mineseen 0*, yourseen 11, line up ←NO KA echoed back
CIRRUS(5): DTR is down                                  ←DTR drops and the link
Serial5, cd2430_sync_mode_init                          reinitializes
Traceback= 3078996 3078BE0 30C91DA 315F5B0 315F6E6
CIRRUS(5): Deasserting DSR
CIRRUS(5): Deasserting DCD
CIRRUS(5): Deasserting CTS
CIRRUS(5): Reset from 0x3078BD8
CIRRUS(5): Asserting DSR
CIRRUS(5): Asserting CTS
CIRRUS(5): Asserting DCD
Serial5: HDLC myseq 12, mineseen 0*, yourseen 12, line down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5, changed state to down
Serial1: attempting to restart                            ←It attempts to restart
And repeats the process
    
```

（待续）

```

HD(1): Deasserting DSR, DTR and RTS
HD(1): Reset from 0x304562A
HD(4): Asserting DSR
HD(1): Asserting DCD and CTS
HD(4): Deasserting LST
HD(4): Asserting DTR and RTS
Serial5: HDLC myseq 13, mineseen 0*, yourseen 13, line up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5, changed state to up
Serial5, cd2430_sync_mode_init
-Traceback= 3078996 3078BE0 30C9200 315F5B0 315F6E6
CIRRUS(5): Deasserting DSR
CIRRUS(5): Deasserting DCD
CIRRUS(5): Deasserting CTS
CIRRUS(5): Reset from 0x3078BD8
CIRRUS(5): Asserting DSR
CIRRUS(5): Asserting CTS
CIRRUS(5): Asserting DCD
Serial5: HDLC myseq 14, mineseen 0*, yourseen 14, line up
    
```

调试结果表明链路中确实存在 HDLC 问题，但是在哪一边呢？回忆一下前面提到的 myseq 字段和 mineseen 字段应该相等的规则。它表明收到了一个正确格式的 HDLC 数据帧，清除了其 keepalive 数之后再返回去。

在例 3-9 中在路由器 bristol_u 上使用 **how interface** 命令的显示结果。

例 3-9 路由器 bristol_u 上运行 show interface 命令的输出结果

```

bristol_u#show interface serial 0
Serial0 is up, line protocol is down
  Hardware is HD64570
  Internet address is 12.168.128.10/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:04, output 00:00:10, output hang never
  Last clearing of "show interface" counters never
  Queuing strategy: fifo
  Output queue 0/40, 44 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    4440 packets input, 258010 bytes, 0 no buffer
    Received 1954 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    4271 packets output, 227876 bytes, 0 underruns
    0 output errors, 0 collisions, 63 interface resets
    0 output buffer failures, 0 output buffers swapped out
    497 carrier transitions
    DCD=up DSR=up DTR=up RTS=up CTS=up

bristol_u#show interface serial 0
Serial0 is up, line protocol is down
  Hardware is HD64570
  Internet address is 12.168.128.10/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:00, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Queuing strategy: fifo
  Output queue 0/40, 44 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    
```

(待续)

```

4450 packets input, 258590 bytes, 0 no buffer
Received 1960 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4274 packets output, 227942 bytes, 0 underruns
0 output errors, 0 collisions, 64 interface resets    ←Note that this field is
0 output buffer failures, 0 output buffers swapped out    incrementing with
503 carrier transitions    ←this field
DCD=up DSR=up DTR=up RTS=up CTS=up
bristol_u#
    
```

这时显示线路工作正常而线路协议没有工作。等待几秒之后再执行同样的命令就会看到接口复位字段和载波转换字段都在递增。这时，再根据一端的 keepalive 没有正常发送以及线路工作而协议不工作的事实，就可以断定该路由器发生了第2层的 HDLC 问题。通过路由器 bristol_u 上的设置信息，会发现没有允许 HDLC 压缩。在路由器 bristol_u 的串口 0 上加入 **compress stac** 命令之后就能够看到路由器 espn 上的线路开始工作了。例 3-10 就显示了路由器 espn 的串口 5 的恢复情况。

例 3-10 debug serial interface 命令在路由器 espn 上的输出结果

```

Serial5: HDLC myseq 165, mineseen 0*, yourseen 67, line down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5, changed state to down
Serial1: attempting to restart
HD(1): Deasserting DSP, CTS and DCD
HD(1): Reset from 0x304562A
HD(1): Asserting DSR
HD(1): Asserting DCD and CTS
HD(1): Deasserting LTST
HD(1): Asserting DTR and RTS
Serial5: HDLC myseq 166, mineseen 166*, yourseen 68, line up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial5, changed state to up
Serial5: HDLC myseq 167, mineseen 167*, yourseen 69, line up
Serial5: HDLC myseq 168, mineseen 168*, yourseen 70, line up
Serial1: attempting to restart
HD(1): Deasserting DSR, CTS and DCD
HD(1): Reset from 0x304562A
HD(1): Asserting DSR
HD(1): Asserting DCD and CTS
HD(1): Deasserting LTST
HD(1): Asserting DTR and RTS
Serial5: HDLC myseq 169, mineseen 169*, yourseen 71, line up
Serial5: HDLC myseq 170, mineseen 170*, yourseen 72, line up
Serial5: HDLC myseq 172, mineseen 172*, yourseen 74, line up
Serial5: HDLC myseq 173, mineseen 173*, yourseen 75, line up
Serial5: HDLC myseq 174, mineseen 174*, yourseen 76, line up
    
```

在路由器 espn 上对路由器 bristol_u 用 **ping** 命令测试，可以看到该链路已经开始正常工作。

再考虑路由器 atlanta 的问题。在例 3-11 中，用 **show interface serial 1** 命令后发现线路和线路协议都没有工作，DTR 和 RTS 信号也没有，这就是第1层出了问题。但是，问题是出在链路的哪一端呢？在路由器 espn 上用 **show controller serial 1** 命令发现一切正常，控制台没有任何错误报告，同时可以看出 DCE V.35 线缆连到其端口上。再在路由器 atlanta 上执行同样的命令，会发现没有检测到串行线缆。例 3-11 显示了分别在路由器 espn 和 atlanta 上 **show controller serial x** 命令的执行情况。

例 3-11 路由器 espn 和 atlanta 上 show controller serial x 命令的执行情况

```

espn#show controllers serial 1
HD unit 1, idb = 0x07A28, driver structure at 0xDC7A8
buffer size 1524, HD unit 1, V.35 DCE cable, clockrate 1000000, DCE cable attached
cpb = 0x43, eda = 0x2140, cda = 0x2000
RX ring with 16 entries at 0x432000
00 bd_ptr=0x2000 pak=0x0DE8A4 ds=0x439C18 status=80 pak_size=0
01 bd_ptr=0x2014 pak=0x0DDC4 ds=0x4373C8 status=80 pak_size=0
<<<text omitted>>>
16 bd_ptr=0x2140 pak=0x0DD4B4 ds=0x435230 status=80 pak_size=0
cpb = 0x43, eda = 0x2800, cda = 0x2800
TX ring with 2 entries at 0x432800
00 bd_ptr=0x2800 pak=0x000000 ds=0x000000 status=80 pak_size=0
01 bd_ptr=0x2814 pak=0x000000 ds=0x000000 status=80 pak_size=0
02 bd_ptr=0x2828 pak=0x000000 ds=0x000000 status=80 pak_size=0
165 missed datagrams, 0 overruns
0 bad datagram encapsulations, 0 memory errors
0 transmitter underruns
0 residual bit errors

espn#

atlanta#
atlanta#show controller serial 0
MK5 unit 0, NIM slot 0, NIM type code 7, NIM version 1
idb = 0x60CF5DF8, driver structure at 0x60CFB100, regaddr = 0x3C000300
IB at 0x40006E64: mode=0x0108, local_addr=0, remote_addr=0
N1=1524, N2=1, scaler=100, T1=1000, T3=2000, TP=1
buffer size 1524
No serial cable attached      <-No serial cable!
RX ring with 32 entries at 0x06EC8 : RLEN=5, Rxhead 0
00 pak=0x60D0322C ds=0xA8214B44 status=80 max_size=1524 pak_size=0
01 pak=0x60D02E44 ds=0xA8214488 status=80 max_size=1524 pak_size=0
<<<text omitted>>>
30 pak=0x60D03038 ds=0xA801449C status=80 max_size=1524 pak_size=0
31 pak=0x60D02A5C ds=0xA8213DCC status=80 max_size=1524 pak_size=0
TX ring with 32 entries at 0x07108 : TLEN=5, TWD=7
tx_count = 0, tx_head = 0, tx_tail = 0
00 pak=0x000000 ds=0xA8000000 status=0x38 max_size=1524 pak_size=0
01 pak=0x000000 ds=0xA8000000 status=0x38 max_size=1524 pak_size=0
<<<text omitted>>>
30 pak=0x000000 ds=0xA8000000 status=0x38 max_size=1524 pak_size=0
31 pak=0x000000 ds=0xA8000000 status=0x38 max_size=1524 pak_size=0
XID/Test TX desc at 0xFFFFF, status=0x30, max_buffer_size=0, packet_size=0
XID/Test RX desc at 0xFFFFF, status=0x0, max_buffer_size=0, packet_size=0
Status Buffer at 0x40007340: rcv=0, tcv=0, local_state=0, remote_state=0
phase=0, tac=0, currd=0x00000, curxd=0x00000
bad_frames=0, frmr=0, T1_timeouts=0, rej_rxs=0, runs=0
0 missed datagrams, 0 overruns
0 bad datagram encapsulations, 0 memory errors
0 transmitter underruns
0 user primitive errors, 0 spurious primitive interrupts
0 provider primitives lost, 0 unexpected provider primitives
mk5025 registers: csr0 = 0x0E00, csr1 = 0x0302, csr2 = 0x0500
                  csr3 = 0x6E64, csr4 = 0x0214, csr5 = 0x0009

atlanta#
    
```

替换串行电缆后，链路工作正常。

3.2 实验 9：HDLC 的配置——第 1 部分

3.2.1 实验说明

HDLC 在实际工作中有很多的用途。前面提到过的一个例子就是在用环回插头测试 CSU/DSU 或者是内部 CSU/DSU 的时候把数据封装形式固定为 HDLC。通常是在连接第 3 方提供的 Cisco 路由器时使用 HDLC。第 3 方的路由器就是非专有自治系统内或者是不在直接控制之下的路由器。HDLC 能够很快捷、简便地进行配置，大大减少由于配置出错而导致故障的几率。

3.2.2 实验内容

这个实验里，假设我们是 ACME Finance 的网络工程师。ACME Finance 要建立一个新的信用卡授权认证中心，并且威斯康星分部也要连入网络。信用卡中心 `cc_center` 与分布路由器 `acme_dist` 之间的 56-kbit/s 连接上要运行的就是 HDLC 协议。信用卡中心想要通过有效载荷压缩的方式尽快提高传输效率。威斯康星分部的路由器 `wi_branch` 则通过一条租用的 T1 线路接入到 `acme_dist` 路由器，但是还没有配置好。WAN 是接入每栋建筑中的第一线路，里面没有用户，因此无需考虑任何 LAN 的配置情况。

3.2.3 实验目的

- 如图 3-2 配置网络。
- 只在串行连接上采用 HDLC 协议。
- 在路由器 `cc_center` 和 `acme_dist` 间的 56-kbit/s 的链路上进行有效载荷的压缩。

3.2.4 所需设备

- 3 台 Cisco 路由器，其中一台必须具有两个串行接口。
- 4 条串行线缆，最好是两条 V.35 DTE 公口线缆和两条 V.35 DCE 母口线缆。或者可以通过适当速率的反接线缆连接 DSU/CSU。在使用实际的 DSU/CSU 时，**clock rate interface** 命令是不必要的。可以参考第 1 章“建立 Internet 网络模型所需的主要组件”，回顾背对背方式连接路由器的各种方法。

3.2.5 物理设计和实验准备

- 如图 3-2 用串行线缆连接好路由器。

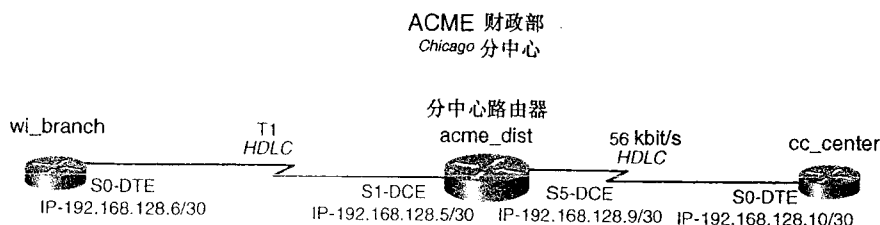


图 3-2 ACME Finance 的网络

3.3 实验 9: HDLC 的配置——第 2 部分

3.3.1 实验步骤

如图 3-2 连接好串行线缆，要确定 DTE 端口与 DCE 端口互联。如果难以确定端口类型，可以用 **show controller** 命令查看一下连在某个接口上的线缆类型（DCE-DTE）。如果路由器是用串行线缆以背对背的方式连在一起，还需要在链路的 DCE 端用 **clock rate** 命令设置波特率。如果路由器是通过 CSU/DSU 来连接，那 CSU/DSU 就是 DCE 设备，而连到 CSU/DSU 的串口和线缆就是 DTE 设备。连接了 CSU/DSU，路由器一端实际上就是 DTE，因此不需要使用 **clock rate** 命令。

分布路由器 **acme_dist** 连接的是 DCE 端。

要对该路由器进行配置，步骤如下：

- 第 1 步 （可选）设置其主机名 **acme_dist**。
- 第 2 步 用 **Encapsulation hdlc** 命令配置 Serial 5 和 Serial 1 的 HDLC 封装。
- 第 3 步 设置 Serial 1 和 Serial 5 上的波特率。
- 第 4 步 配置 Serial 1 上的 STAC 压缩。
- 第 5 步 为两个串行接口分配 IP 地址。
- 第 6 步 （可选）配置路由选择协议。

例 3-12 是这些步骤的示例。

例 3-12 配置 HDLC DCE 接口

```

Router#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname acme_dist
acme_dist(config)#interface serial 5
acme_dist(config-if)#encapsulation hdlc          ← This is optional, HDLC is default
acme_dist(config-if)#clock rate 56000           ← Tells the router to send a clock
acme_dist(config-if)#compress stac              ← Enables STAC compression
acme_dist(config-if)#ip address 192.168.128.9 255.255.255.252
acme_dist(config-if)#no shut
acme_dist(config-if)#exit
acme_dist(config)#
acme_dist(config)#interface serial 1
acme_dist(config-if)#clock rate 1000000
  
```



```
acme_dist(config-if)#ip address 192.168.128.5 255.255.255.252
acme_dist(config-if)#no shut
acme_dist(config-if)#exit
acme_dist(config)#
acme_dist(config)#router eigrp 2001          -This is optional, configures
acme_dist(config-router)#network 192.168.128.0 -EIGRP as the routing protocol
for this network
acme_dist(config-router)#^Z
acme_dist#
```

现在到信用卡中心去，需要把这台路由器配置为 HDLC。这台路由器上需要对压缩进行配置，这样它才能正确接收经过了压缩的有效载荷。要想使该实验更接近实际，还需要为它加上一个 IP 地址、一个路由选择协议以及一个主机名。例 3-13 就是信用卡中心的 cc_center 路由器的配置示例。

例 3-13 配置网络 DTE 端的 HDLC

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname cc_center
cc_center(config)#interface serial 0
cc_center(config-if)#encapsulation hdlc
cc_center(config-if)#compress stac
cc_center(config-if)#ip address 192.168.128.10 255.255.255.252
cc_center(config-if)#no shut
cc_center(config-if)#exit
cc_center(config)#
cc_center(config)#router eigrp 2001
cc_center(config-router)#network 192.168.128.0
cc_center(config-router)#^Z
cc_center#
```

最后，例 3-14 给出了威斯康星分部的配置情况。这个例子中利用了 Cisco 默认设置和缩写简化了配置。例如，co 既可能是 configure 也可能是 copy，但是‘cop’是惟一的，因此说 cop 足够了。例 3-14 给出了这种省时的命令步骤示例。

例 3-14 使用默认的简化键入方式进行 wi_branch 的 HDLC 配置示例

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname wi_branch
wi_branch(config)#int s0
wi_branch(config-if)#ip add 192.168.128.6 255.255.255.252
wi_branch(config-if)#no shut
wi_branch(config-if)#
wi_branch(config-if)#
wi_branch(config-if)#router eigrp 2001
wi_branch(config-router)#network 192.168.128.0
wi_branch(config-router)#^Z
wi_branch#
```

例 3-15 列出了整个的配置过程示例。

例 3-15 实验 9: ACME Finance 的网络配置完整示例

```

Distribution Router (acme_dist)

hostname acme_dist
!
<<<text omitted>>>
!
interface Serial1
 ip address 192.168.128.5 255.255.255.252
 clockrate 1000000
!
<<<text omitted>>>
!
interface Serial5
 ip address 192.168.128.9 255.255.255.252
 clockrate 56000
 compress stac
!
<<<text omitted>>>
!
router eigrp 2001
 network 192.169.128.0

-----

Credit Card Center (cc_center)

hostname cc_center
!
interface Serial0
 ip address 192.168.128.10 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
 compress stac
!
<<<text omitted>>>
!
router eigrp 2001
 network 192.168.128.0

-----

Wisconsin Branch (wi_branch)

hostname wi_branch
!
<<<text omitted>>>
!
interface Serial0
 ip address 192.168.128.6 255.255.255.252
 no ip mroute-cache
 no fair-queue
!
<<<text omitted>>>
!
router eigrp 2001
 network 192.168.128.0

```

要验证这一部分的配置，可以使用的命令包括：**show interface serial x**，**show ip eigrp neighbors**，当然还有 **ping**。配置全部完成之后，在路由器 **acme_dist** 上就可以看到两个 EIGRP 的邻居路由器，两台路由器互为邻居路由器。用 **show interface** 命令可以显示线路已经开始工作，协议也已经开始运行，而 DCD、DSR、DTR、RTS 和 CTS 也都进入了 up 状态。

第 4 章

WAN 协议与技术： 点对点协议（PPP）

随着 Internet 的迅速普及，Internet 访问协议——点对点协议（PPP）得到了广泛的应用。现在大多数的模拟拨号连接都采用 PPP 作为其数据链路协议，这主要是因为 PPP 在网络连接方面各种方便灵活的特性：

- 错误检测。
- 自动协商网络层地址。
- CHAP 或 PAP 认证。
- 数据压缩。
- 符合 ISO 标准。

在成为网络访问的主流协议之前，PPP 通常是和串行线路 Internet 协议（SLIP）一起使用的，因而二者之间也经常使人混淆。很多人将点对点协议称为 SLIP/PPP。但是，SLIP 仅仅支持 IP 作为其惟一的网络层协议，因此就无法满足很多采用 IP、IPX 以及 AppleTalk 等多协议的网络管理员的需要。

Cisco 的 PPP 模型是建立在 RFC 1661 上的。该 RFC 阐述了 PPP 是怎样在点对点网络连接上对网络层协议信息进行封装的。PPP 将数据链路层分为了 3 个具有各自特定功能的子层：

- 网络控制协议（NCP）——建立和协商网络层协议以及相应的地址。
- 链路控制协议（LCP）——建立链路、认证用户和

检测链路的质量。

- 高级数据链路控制 (HDLC) ——在链路上封装数据包。遵从 RFC 1662 协议规定。表 4-1 概述了 PPP 及其子层。

表 4-1 PPP 的 OSI 模型子层

OSI 层	常见协议
第 3 层	IP、IPX、AppleTalk
第 2 层	NCP、LCP、HDLC
第 1 层	EIA/TIA-232、V.24、V.23、V.35 和 ISDN 等

其他与 PPP 相关的 RFC 包括：

- RFC 1144——TCP/IP 数据包头压缩。
- RFC 1220——PPP 在网桥上的扩充。
- RFC 1334——PPP 认证协议。
- RFC 1378——PPP AppleTalk 控制协议 (ATCP)。
- RFC 1552——PPP 互联网数据包交换控制协议 (IPXCP)。
- RFC 1570——PPP LCP 协议扩充。
- RFC 1661——PPP 协议 (PPP)。
- RFC 1662——PPP 中的 HDLC 帧封装。
- RFC 1990——PPP 多链路协议 (MP)。

注释 在网站 www.isi.edu/in-notes/rfcxxxx.txt 上可以找到所有的 RFC 概要，这里的 xxxx 是 RFC 的编号。

如上所述，PPP 采用 HDLC 协议对链路上的数据包进行封装。在国际标准化组织 (ISO) 的 3309 规范中规定和讲述了 PPP 的数据帧结构和工作原理，而 1984/PDAD1 又对此进行了修正，使得 PPP 可以用于异步通信环境并可以发起和终止传输。PPP 大多数的扩展功能，包括数据纠错以及支持多种网络层协议等，都是由链接控制协议 (LCP) 和网络控制协议 (NCP) 来控制的。LCP 数据帧用于配置和测试数据链路。LCP 的工作方式如下：

第 1 步 链路建立阶段——LCP 首先打开连接，然后确定相关的通信参数，包括最大接收单元，数据压缩类型以及链路认证协议类型。链路设置完后，会和接收配置确认帧。之后是可选的链路质量确认阶段，这时 LCP 要确定链路质量是否可以运行所需的网络层协议。

第 2 步 可选 (一般来说是必要的) 的认证阶段——链路建立且认证协议确定之后就要进行认证。Cisco 提供了两种认证方式：质询应答握手认证协议 (CHAP) 和密码认证协议 (PAP)。PPP 标准本身不需要任何认证，非拨号连接和 ISDN 连接都是如此。然而，在 cisco 路由器上，异步线路上的模拟拨号连接却需要认证，可以是呼叫者身份认证或者采用 CHAP 或 PAP 的认证。这可以为什么解释 PPP 设置的适用范围不同。为了防止这种情况的发生，Cisco TAC 建议使用 CHAP 方式。RFC 1994 定义了 PPP CHAP (而不是 RFC 1334)。

第3步 网络层协议阶段——该阶段中，LCP 引导 NCP 来激活和配置网络层协议。在这一阶段结束之后，网络层的数据包就能够通过链路进行传输。

第4步 链路终止阶段——LCP 可以通过用户中断或者某个物理事件终止链路连接。LCP 指导 NCP 关闭第3层网络协议并采取相应动作。

LCP 通过3种类型的 LCP 数据帧来完成上述步骤：

- 链路建立帧（**Link establishment frames**）——建立链路。
- 链路终止帧（**Link termination frames**）——关闭链路。
- 链路维护帧（**Link maintenance frames**）——维护链路。

4.1 PPP 的多种用途

PPP 是一个多用途的协议，可以适用许多接口类型：

- 同步
- 异步
- ISDN
- 高速串口（HSSI）
- 数字用户线路（DSL）

现在可以看到，PPP 能用于许多种环境中。并且，如上所述，多数人都是以 PPP 作为访问 Internet 的数据链路层协议的。在 ISDN 和非 Cisco 设备的数据链路层上通常使用的协议也是 PPP。另外，PPP 还能用在串行备份链路中，或者通过多链路 PPP 将多个基本速率接口（BRI）绑定在一起，以达到最大带宽和负载平衡。

本书中会应用多种 PPP 使用方式在 Internet 网络中建立 PPP 模型。首先，在串行链路上配置 PPP。随后，将 PPP 的应用扩展到使用调制解调器的异步端口上。最后，本章还讨论了如何运用 PPP 的高级功能，如 PPP 数据压缩、多链路 PPP 以及其他特性。ISDN 上的 PPP 应用会在第7章“WAN 协议与技术：综合业务数字网（ISDN）”进行讲述。

4.1.1 在同步串行链路上配置 PPP

在串行接口上配置 PPP，首先就是用 **encapsulation ppp** 命令设置接口上的 PPP 数据封装。如果要配置 PPP 链路，必须在链路两端的端口上都设置 PPP 数据封装。如果配置的是两台路由器的 PPP 连接，可以采用 DTE—DCE 线缆，在 DCE 端还要用 **clock rate bit/s** 命令设置通信速率。基本的 PPP 配置就是设置端口的数据封装和网络层地址。当然，这只介绍基本的配置，在后面的章节中，我们还会用到 PPP 的一些高级特性。

第一个例子是在两台 Cisco 路由器之间的串行接口上进行 PPP 的配置。图 4-1 是网络的示意图与地址分配图。首先，配置 r1 的 Serial 1（S1）和 r2 的 Serial 0（S0）上的 PPP 数据封装。由于这是一个 V.35 背对背的连接方式，链路一端应该配置成 DCE。在 r1 的 Serial 1 接口上使用 **clock rate** 命令。

4.1.2 在模拟拨号链路的异步端口上进行 PPP 配置

在模拟拨号链路的异步端口上进行 PPP 配置比配置串行链路的 PPP 要棘手一些。本节先简单列出了其所需的步骤，然后针对每个步骤进行详细讲解。配置一个异步拨号网络连接的 PPP 步骤如下：

- 第1步 配置调制解调器和异步通信口，包括调制解调器的连接和配置，路由器异步通信端口的设置以及设定用于识别异步通信口的绝对线路号。可以参考第1章“建立网络互联模型的关键组件”，其中详细地讲述了调制解调器的连接和配置问题。
- 第2步 定义和配置异步通信口上的 PPP，设置第1步里所选择的绝对线路号所对应的异步通信口，再为该异步口设置 PPP 数据封装和 PPP 认证方式。
- 第3步 配置与异步通信口相对应的网络层地址或者是地址映射以及路由方案。
- 第4步 在异步口上设置按需拨号路由（DDR）。

1. 第1步：调制解调器和异步端口的配置

按照第一章中的说明，将调制解调器和 AUX 端口或者路由器的异步端口正确相连，这里包括用 **modem inout** 命令和 **modem autoconfigure** 命令设置会话脚本（chat script）的使用。在远端路由器拨入本地主机的情况下需要会话脚本（chat script）。在线路配置模式下使用 **script dialer script_name** 命令可以仅在拨出情况调用会话脚本 chat script。会话脚本（chat script）必须简单易懂。例如，下面这段脚本会使调制解调器复位，载入出厂默认设置，再拨 5496561 的号码并等待连接信息出现。

```
chat-script dialhost "" "ATZ&F" OK "ATDT5496561" TIMEOUT 60 CONNECT
```

脚本实际上包含了模拟拨号的号码。在拨号方映射字符串中也有一个电话号码，但这主要是用于认证和拨号识别，而不是真的要将其用于拨号中。在 ISDN 如果使用拨号映射，就是真正用于拨号了。请记住用 **show line** 命令来找出对应于连有调制解调器端口的绝对线路号。例 4-3 是将调制解调器连接到 Cisco 2500 路由器 AUX 端口的例子。

例 4-3 用 show line 命令显示 AUX 端口上的绝对线路号

例 4-3 用 SHOW line 命令显示 A 路由器上的配置

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	1	0/0	← Aux port
2	VTY		-	-	-	-	-	0	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	-
Router#											

以后将要用该绝对线路号（本例中是线路 1）来配置一个异步链路。绝对线路号会随路由器类型的不同而改变，因此一定要用 **show line** 命令来确定实际的绝对线路号的值。例如，例 4-4 中，调制解调器连在了终端服务器的端口 16 上，AUX 端口的绝对线路号是 17，而不

是 1。从这个例子也能够观察出该线路不处在工作状态，但已经设置了收发时钟。这表明，调制解调器已配置好并连接到线路上。

例 4-4 在 AUX 的端口上用 show line 命令显示绝对线路号

```
access_server#show line
```

Tty	Type	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	7	0	0/0	-
* 1	TTY	9600/9600	-	-	-	-	-	1	0	0/0	-
* 2	TTY	9600/9600	-	-	-	-	-	1	1776	0/0	-
* 3	TTY	9600/9600	-	-	-	-	-	1	1	0/0	-
* 4	TTY	9600/9600	-	-	-	-	-	1	0	0/0	-
* 5	TTY	9600/9600	-	-	-	-	-	1	1	0/0	-
* 6	TTY	9600/9600	-	-	-	-	-	1	0	0/0	-
* 7	TTY	9600/9600	-	-	-	-	-	1	0	0/0	-
8	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
9	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
10	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
11	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
12	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
13	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
14	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
15	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
16	TTY	115200/115200	-	inout	-	-	-	0	0	0/0	-
17	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
18	VTY		-	-	-	-	-	1	0	0/0	-
19	VTY		-	-	-	-	-	0	0	0/0	-
20	VTY		-	-	-	-	-	0	0	0/0	-
21	VTY		-	-	-	-	-	0	0	0/0	-
Tty	Type	Tx/Rx	A	Modem	Roty	Acc0	AccI	Uses	Noise	Overruns	Int
22	VTY		-	-	-	-	-	0	0	0/0	-

```
access_server#
```

2. 第 2 步: 在异步接口上定义和配置 PPP

紧接着就是配置路由器异步接口以运行 PPP。输入 **interface async interface-number** 命令进入异步接口。如果在接口上启动 PPP，必须先确定怎么使用 PPP 或 DDR。这主要是因为 PPP 初始化时，在呼叫建立时要使用 PPP 字符串。这样的例子就是 **/routing** 路由命令可以通过 PPP 连接传输。Cisco 提供了多条命令来支持这种灵活的配置方式：

- **async mode {dedicated | interactive}**——该命令在默认情况下关闭，也就是说，没有配置任何异步模式。因为没有允许 PPP 和 SLIP 连接，通信线路不会接受拨入的网络连接。要使用任何形式的 PPP 或 SLIP 协议，必须采用下面两种异步模式的一种：
 - **专用异步模式**：路由器不等待任何终端用户的提示，无需任何用户命令来建立远程连接，接口自动设置成 SLIP 或 PPP。远程终端不能选择任何数据封装方式、地址以及其他的模式。
 - **交互式异步模式**：路由器在建立连接之前会等待终端用户送来一个 EXEC 命令。如果远程终端用户需要建立会话参数并且通过在链路路由的话，就应该使用 **async mode interactive** 命令。
- **async {dynamic | default} routing**——在使用 **async mode interactive** 命令时，必须同时使用 **async dynamic routing** 命令，该命令会允许路由器去接受远程用户送来的

/routing 关键字。如果主机接口设成了交互式模式，只要命令中有**/route** 的关键字，**async default routing** 命令会使得 PPP 和 SLIP 的 EXEC 命令被解释。该命令还会在专用异步接口上允许运行路由选择协议。

- **autoselect {ppp | slip | during-login | arap}**——该命令用于绝对线路号上，是使用调制解调器的基本命令。**autoselect** 命令使得路由器在接收到适当的起始字符后开始运行其网络协议。例如，如果路由器收到一个回车字符，就会开始 EXEC 会话。表 4-2 列出了 SLIP、PPP 和 ARAP 用的 16 进制表示的帧标志。

表 4-2 PPP 帧标志

协 议	16 进制帧标志
回车符	0D
SLIP	C0
PPP	7E
ARAP	10

- 参数 **during-login** 用于不同协议类型的终端用户或客户端希望通过拨号连接到路由器的同一端口。例如，一客户端使用如超级终端这样的终端仿真器在执行一个 TTY 会话的同时，另外一个客户端则想通过 PPP 连接到同样的端口上，这时就需要使用这一参数。
- **transport input {all | lat | map | nasi | none | pad | rlogin | telnet | v120}**——默认情况下，Cisco 路由器不接收到其异步通信口的拨入链路。在线路能够接受一个拨入连接之前必须要指定接入的传输协议或者使用 **transport input all** 命令使路由器允许建立连接。

使用如下命令可以允许在异步通信口上运行 PPP：

- **encapsulation ppp**——如上所述，该命令能够配置接口的 PPP 数据封装。
- **ppp authentication {chap | pap}**——在模拟拨号线路上启动 PPP 认证以提供安全可靠的拨号连接。

3. 质询应答握手认证协议（CHAP）和密码认证协议（PAP）

所有使用 PPP 的接口都可以使用 CHAP 和 PAP 认证。这两种认证方式最初都出自 RFC 1334，后来 CHAP 在 RFC 1994 中做了更新和修改。CHAP 和 PAP 都是利用每个设备或每台路由器都有一个惟一的主机名这个特点。该认证过程还能防止某个路由器错误地呼叫其他路由器上已经配置好、预备和其他站点连接的端口。此时 PPP 的工作方式如下：

第 1 步 建立了一个 PPP 会话之后，路由器确认 LCP 需要的认证类型。

第 2 步 路由器确认 CHAP 或 PAP 的认证方式，然后判断下列认证方式：

- 检查本地用户数据库，寻找适当的用户名和密码匹配，这是默认设置，无需 **login local** 命令。
- 将认证请求转发给 TACACS+或 RADIUS 服务器。

第 3 步 路由器的认证过程基于从本地用户数据库或安全服务器收到的认证请求回应。

如果有肯定的返回应答，路由器就创建 PPP 过程，如果返回否定的应答，路由

器会立即拒绝用户的请求。

LCP 确定链路的参数后，开始执行 CHAP 和 PAP。PAP 通过链路发送明文密码，路由器在链路的远端对其进行确认。运用数据分析师或线路监视器能够轻易捕获返回的应答，以后就可以模拟这一过程通过认证。因此，这种认证方式不是特别安全。CHAP 不会在链路中发送明文密码，使得 CHAP 在本质上比 PAP 方式更为安全。CHAP 在质询过程中使用了 MD5 散列数发生器产生一个 128 位的随机数。链路上传送的只有这些散列数及其限定字符。

正在初始化的路由器首先发送一个质询信号到远端路由器，远端路由器会回应下面 4 部分重要信息：

- CHAP 质询信号数据包类型识别号。
- 身份标识 (ID) 的版本号，识别质询信号序列号。
- 一个随机数。
- 发出质询信号路由器的主机名。

收到应答之后，远端路由器会查找与此用户名相对应的密码。然后，身份标识 ID，随机数以及密码都会输入到 MD5 散列数发生器里，产生一个散列数。MD5 散列值就是需要在链路上返回的数字。散列数是和 CHAP 应答数据包类型识别号、身份标识以及路由器主机名一起发送的。正在初始化的那台路由器会执行相同的过程，也就是说，它会根据远端路由器送来的主机名查找到它对应的密码，然后将此密码和身份识别 ID 和随机数一起送进 MD5 散列数发生器，产生与远端路由器送来的散列数值相等的一个散列数。如果二者不相等，认证过程失败，链路关闭。

配置 PPP CHAP 认证的过程如下：

第 1 步 在适当接口上配置 PPP 数据封装格式。

第 2 步 在本地路由器上配置一个和拨入的远端路由器主机名一致的用户名。在远端路由器上也加入一个和本地路由器主机名一致的用户名。分配给两个用户名的密码必须一致。切记，密码是区分大小写的。

第 3 步 在 PPP 接口下，配置下面的与 CHAP 有关的命令：

```
ppp authentication chap
```

图 4-2 和例 4-5 是在异步接口上配置 PPP 的 CHAP 认证方式的例子。通常情况下，应该使用 `service password-encryption` 命令，这样路由器显示的密码就是经过了加密的。为了示例的可读性以及便于讲解的目的，例子中的密码都没有加密。

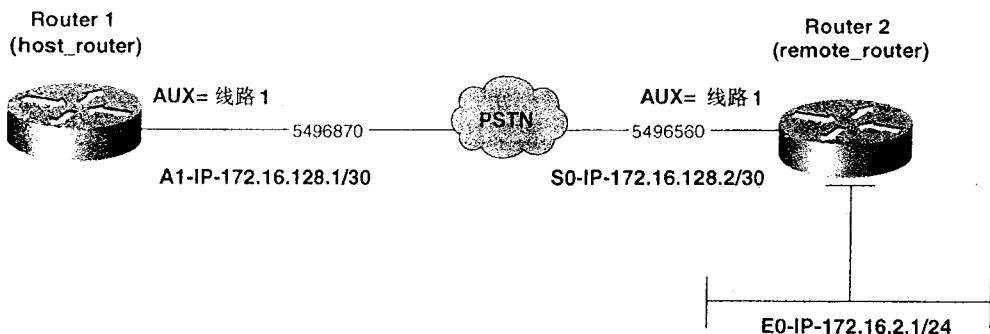


图 4-2 异步拨号配置情况下的 PPP CHAP

例 4-5 模拟拨号连接中，host_routers 的 CHAP 认证配置

```

hostname host_router
!
!
username remote_router password 0 cisco
ip subnet-zero
!
interface Async1
ip address 172.16.128.1 255.255.255.252
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer map ip 172.16.128.2 name remote_router broadcast 5496560
async mode interactive
ppp authentication chap
!
ip classless
!
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
autoselect ppp
login local
modem InOut
modem autoconfigure discovery
transport input all
speed 38400
line vty 0 4
login
!
end

```

配置 PPP PAP 认证的过程如下：

第 1 步 确保在适当的接口上配置好 PPP 的数据封装格式。

第 2 步 在本地路由器上加入和需要拨入本地网络的远端路由器主机名一致的用户名。
在远端路由器上加入本地路由器的 hostname。分配给两个用户名的密码必须一致，
注意密码区分大小写。

第 3 步 在 PPP 接口上，配置下列与 PAP 相关的命令：

```

ppp authentication pap
ppp pap sent-username local_device_name password password

```

注释 无论何时在客户端进行 PPP 拨号或者建立会话，主机都会提示用户输入密码，这表明被呼叫方在向客户端发送 PAP 质询信号。多数 Internet 运营商（ISP）都是采用这种方式。

图 4-3 是在与图 4-2 相同的网络中进行 PPP 的 PAP 配置的例子。

例 4-6 给出了图 4-3 里网络中路由器 host_router 和 remote_router 的配置情况。

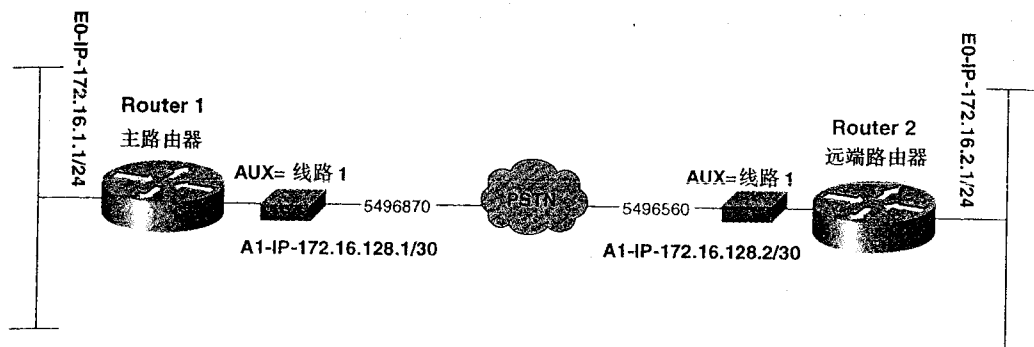


图 4-3 异步拨号配置情况下的 PPP PAP 设置

例 4-6 模拟拨号连接中路由器 host_router 和 remote_router 的 PAP 认证配置情况

```

hostname host_router
!
username remote_router password 0 cisco1
ip subnet-zero
chat-script dialremote "" "ATZ&F" OK "ATDT5496561" TIMEOUT 60 CONNECT
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
!
interface Async1
ip address 172.16.128.1 255.255.255.252
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer idle-timeout 305
dialer map ip 172.16.128.2 name remote_router broadcast 5496560
dialer-group 1
async mode interactive
ppp authentication pap
ppp pap sent-username host_router password cisco1
!
ip classless
ip route 172.16.2.0 255.255.255.0 172.16.128.2
!
dialer-list 1 protocol ip permit
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
autoselect ppp
script dialer dialremote
modem InOut
modem autoconfigure type usr_sportster
transport input all
speed 38400
line vty 0 4
login
!
end

```

(待续)

```
host_router#
hostname remote_router
!
username host_router password 0 cisco1
chat-script dialhost "" "ATZ&F" OK "ATDT5496870" TIMEOUT 60 CONNECT
!
interface Ethernet0
 ip address 172.16.2.1 255.255.255.0
!
interface Async1
 ip address 172.16.128.2 255.255.255.252
 encapsulation ppp
 async mode interactive
 dialer in-band
 dialer idle-timeout 305
 dialer map ip 172.16.128.1 name host_router broadcast 5496870
 dialer-group 1
 ppp authentication pap
 ppp pap sent-username remote_router password cisco1
!
no ip classless
ip route 172.16.1.0 255.255.255.0 172.16.128.1
!
dialer-list 1 protocol ip permit
!
line con 0
line aux 0
 autoselect ppp
 script dialer dialhost
 modem InOut
 modem autoconfigure discovery
 transport input all
 rxspeed 38400
 txspeed 38400
line vty 0 4
 login
!
end

remote_router#
```

注释 路由器命名时，最好全部用小写。如果想在用户名中加入更多的文字，可以使用下划线“_”。在整个网络或模型中要保持统一的命名规范以避免可能的打字错误或大小写的不匹配。很多时候，这种错误会在 **map** 声明命令或者是某个认证过程中出现。例如，尽管都叫做“host_router”，但“Host_router”并不等于“host_router”或“host-router”。由于一些路由器的名称是专用名词，它们在本书的每个部分都是小写的，所以这在有的时候容易造成混淆。

图 4-4 是两个路由器之间异步 PPP 拨号连接的网络示意图以及 IP 地址分配图。配置这样的网络模型时，先要将调制解调器连到每台路由器的 AUX 端口并且对其进行配置。在第一章中可以找到更多更详细的关于调制解调器配置问题的说明。

例 4-7 是对图 4-4 中的 host_router 和 remote_router 进行配置的例子。

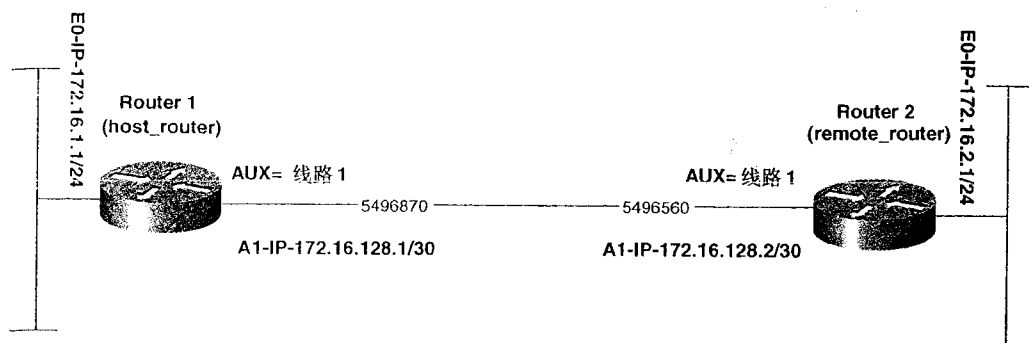


图 4-4 在异步拨号连接上的 PPP 网络示意图和 IP 配置方案

例 4-7 模拟拨号连接中路由器 host_router 和 remote_router 的 PAP 认证配置

```

hostname host_router
!
username remote_router password 0 cisco1
ip subnet-zero
chat-script dialremote "" "ATZ&F" OK "ATDT5496561" TIMEOUT 60 CONNECT
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
!
interface Async1
ip address 172.16.128.1 255.255.255.252
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer idle-timeout 305
dialer map ip 172.16.128.2 name remote_router broadcast 5496560
dialer-group 1
async mode interactive
ppp authentication pap
ppp pap sent-username host_router password cisco
!
ip classless
ip route 172.16.2.0 255.255.255.0 172.16.128.2
!
dialer-list 1 protocol ip permit
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
autoselect ppp
script dialer dialremote
modem InOut
modem autoconfigure type usr_sportster
transport input all
speed 38400
line vty 0 4
login
!
end

```

(待续)

```

host_router#
hostname remote_router
!
username host_router password 0 cisco1
chat-script dialhost "" "ATZ&F" OK "ATDT5496870" TIMEOUT 60 CONNECT
!
interface Ethernet0
 ip address 172.16.2.1 255.255.255.0
!
interface Async1
 ip address 172.16.128.2 255.255.255.252
 encapsulation ppp
 async mode interactive
 dialer in-band
 dialer idle-timeout 305
 dialer map ip 172.16.128.1 name host_router broadcast 5496870
 dialer-group 1
 ppp authentication pap
 ppp pap sent-username remote_router password cisco
!
no ip classless
ip route 172.16.1.0 255.255.255.0 172.16.128.1
!
dialer-list 1 protocol ip permit
!
line con 0
line aux 0
 autoselect ppp
 script dialer dialhost
 modem InOut
 modem autoconfigure discovery
 transport input all
 rxspeed 38400
 txspeed 38400
line vty 0 4
 login
!
end

remote_router#

```

例 4-8 演示了在路由器的异步端口及 AUX 端口上配置拨入拨出的 PPP 连接的过程，也包括对调制解调器的设置。配置调制解调器的 AUX 端口时，先用 **show line** 命令找到调制解调器将要连接的绝对线路号，然后配置调制解调器要使用的端口，接着是 PPP 的异步接口的配置工作。

例 4-8 异步端口关于调制解调器和 PPP 的配置（仅第 1 和第 2 步）

```

remote_router#show line

```

Tty	Type	Tx/Rx	A	Modem	Roty	Accu	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	1	0/0	-
1	AUX	9600/9600	-	-	-	-	-	0	1	0/0	- ←Modem port
2	VTY		-	-	-	-	-	0	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	-

（待续）

```
remote_router#
remote_router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
remote_router(config)#interface async 1
remote_router(config-if)#async mode interactive
remote_router(config-if)#encapsulation ppp
remote_router(config-if)#exit
remote_router(config)#
remote_router(config)#line 1
remote_router(config-line)#modem inout
remote_router(config-line)#modem autoconfigure discovery
remote_router(config-line)#autoselect ppp
remote_router(config-line)#transport input all
remote_router(config-line)#^Z
remote_router#
```

在第 1 章中，我们知道可以通过增加一个本地环路接口来利用反向 Telnet 会话对调制解调器进行测试。另一个快速测试调制解调器的方法就是 Telnet 连接到端口 2001 的以太网接口。例 4-9 给出了一个成功地用反向 Telnet 连接到端口 1 上调制解调器的例子。在和调制解调器进行会话通信的过程中，可以用 ATZ 命令或像 AT&V 这样更为具体的命令来显示当前调制解调器的配置情况。注意在调制解调器测试完成之后用 disconnect 命令断开与它的连接，通过键盘输入 Ctrl-Shift-6 的组合键使操作回到路由器上，然后，输入 disconnect 命令以断开连接。

例 4-9 用反向 Telnet 会话测试 Modem

```
remote_router#telnet 172.16.2.1 2001
Trying 172.16.2.1, 2001 ... Open
atz
OK
at&v
ACTIVE PROFILE:
B1 E1 L1 M1 N1 Q0 T V1 W0 X0 Y0 &C1 &D2 &G0 &J0 &K3 &Q5 &R1 &S0 &T5 &X0 &Y0
S00:001 S01:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:055 S08:002 S09:006
S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S36:007 S37:000 S38:020 S46:138
S48:007 S95:000

OK
CTRL/SHIFT/6

remote_router#disconnect
Closing connection to 172.16.2.1 [confirm]y
```

最后两个配置阶段包括网络层地址和 DDR 的设置，它们之间是紧密相关的，因为 DDR 需要使用相应的网络层寻址方式。

4. 第 3 步: 配置网络层地址或寻址方案和应用异步接口的路由模式

在异步接口上配置 PPP 的下一步骤就是设置所有接口的网络层地址，除了设置标准寻址方式外，还有一些特别的应用协议设置，像 OSPF 这样的路由选择协议所用到的按需电路 (demand circuits) 的配置。

有很多方法可以配置拨号端口的 IP 地址。可以配置路由器为客户端指定分配 IP 地址，也可以配置路由器从一个地址池中为其分配地址，或者是忽略客户端的寻址要求，而是用一个 dialer map 命令声明如何到达该地址。从地址池或路由器中为客户端分配 IP 地址时可以

使用 **async dynamic address** 和 **peer default ip address pool** 命令。表 4-3 中列出了为拨号接口配置和分配 IP 地址的组合方式。

表 4-3 推荐的 PPP 路由器地址和寻址方式的设置方法

本地路由器接口命令	远端路由器或客户端 PPP 设置
路由器静态 IP: ip address local_ip_address dialer map ip remote_ip_address	路由器静态 IP: ip address remote_ip_address dialer map ip local_ip_address
路由器静态 IP: ip address local_ip_address dialer map ip remote_ip_address	路由器动态 IP: ip address negotiated dialer map ip a.b.c.d
路由器静态 IP: ip address local_ip_address dialer map ip remote_ip_address	Windows 95/98/2000 静态 IP: 指定 IP 地址: remote_ip_address
路由器静态 IP，动态客户端: ip address local_ip_address peer default ip address remote_ip_address	Windows 95/98/2000 动态 IP: 自动获得 IP 地址: remote_ip_address
路由器静态 IP，动态客户端来源于地址池: async dynamic address ip address local_ip_address peer default ip address pool pool_name 从全局模式，配置地址池: ip local pool {default pool_name low_ip_address [high_ip_address]}	Windows 95/98/2000d 动态 IP: 自动获得 IP 地址:

在不能控制拨号接口 IP 地址分配的情况下使用 **ip address negotiated** 命令，一般来说对 ISP 就是这样。通常情况下，在用户建立一个 PPP 连接时，ISP 会为它分配一个地址。用 **ip address negotiated** 命令可以让 LCP 正确接收到路由器分配的 IP 地址，该命令用于“简易 IP 配置”，典型应用为网络地址转换（NAT）中获取动态地址作为其外部地址，并以 TCP Overload 模式工作。

注释 必须在命令 **dialer in-band** 输入之前键入 **peer default ip address local_ip_address**。如果先使用 **dialer in-band** 命令，路由器就不会接收远端路由器指定的 IP 地址。

图 4-5 为 Windows 95/98 客户端拨入访问服务器的例子。Windows 工作站的 PPP 协议堆栈配置成自动获取 IP 地址，其拨号网络使用 PPP 的 PAP 进行认证。

如果没有使用 DHCP，工作站访问 Internet 时还需要在客户端配置所需的 DNS 服务器。在例 4-10 中，访问服务器 **access_server** 为拨入 PPP 客户分配的地址是 172.16.20.2。注意访问服务器上也有一个拨号列表，定义了期望业务，使得链路在客户端发送数据时不会断开。拨号空闲超时的设置使得连接在闲置 5 分 5 秒（305 秒）之后就会自动断开链路。

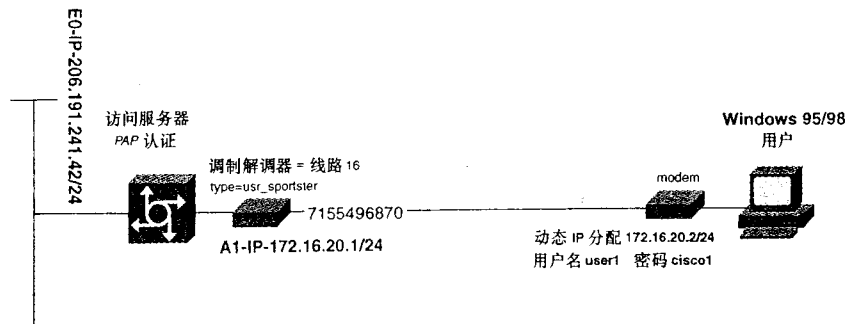


图 4-5 Windows 95 下的 PPP 拨入设置情况

例 4-10 WIN 95 下模拟拨号连接的配置情况

```

hostname access_server
!
username user1 password 0 cisco1
ip subnet-zero
!
interface Ethernet0
ip address 206.191.241.42 255.255.255.248
no ip directed-broadcast
!
interface Async16
ip address 172.16.20.1 255.255.255.0
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer idle-timeout 305
dialer-group 1
async mode interactive
peer default ip address 172.16.20.2
no cdp enable
ppp authentication pap
!
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
exec-timeout 0 0
transport input none
line 1 8
transport input all
line 9 15
line 16
autoselect ppp
login local
modem InOut
modem autoconfigure type usr_sportster
transport input all
speed 115200
line aux 0
line vty 0 4
login
!
end

access_server#

```

注释 默认情况下，所有的接口上都启动了 Cisco 发现协议（CDP）。用 **no cdp enable** 命令可以禁止该协议。CDP 应用于 DDR 路由中并可以帮助识别其他邻近的 Cisco 设备。CDP 信息可以使没有禁止 CDP 的接口拨号线路一直处于工作状态。如果没有使用 DDR，那么没有必要打开 CDP，因为有安全隐患。

5. 路由更新的控制

步骤 3 的第 2 部分是 DDR 链路的路由配置。适当控制路由更新是 DDR 配置中最为困难的部分之一。为了防止路由更新信息不停拨号以试图建立链路，必须更加全面深入了解所使用的路由选择协议。还要明确路由选择协议如何更新路由信息，并了解在不同网络上传送时，应

将有用的信息过滤掉。网络工程师的麻烦就在于如何广播路由信息，但又不能使链路一直保持激活。

有几种方法可在 DDR 链路上保持路由而又不使链路一直处于激活状态。Cisco 采用了快照路由（*snapshot routing*）技术来为 IP 协议中的距离矢量路由选择协议提供路由更新，还可用于 IPX RIP 和 AppleTalk RTMP。该技术包括为路由表瞬间快照，并让电子欺骗（*spoofing*）的方式参与快照路由的接口为激活以将路由表信息保持一段时间。这段时间称为快照稳定期（*snapshot quiet time*）。当这段时间超时后，路由器会重新呼叫快照服务器以接收路由更新信息。快照路由技术能够帮助解决动态连接中的路由问题，但是它有一个缺陷，就是不支持使用变长子网掩码（VLSM）的协议。在 OSPF 网络链路中，OSPF 必需使用按需电路（*demand circuit*）技术。

按需电路的概念很复杂，我们会在第 12 章“链路状态协议：开放式最短路径优先（OSPF）”进行详细讲述。简单地说，按需电路会抑制发往接口的 OSPF 的 hello 握手信号，从而使得链路或电路能够正常超时中断。Cisco 最近为 IP EIGRP 和 IGRP 增加了 **dialer watch** 命令，使得路由器能够查找特定路由并且根据该路由是否在其路由表中来决定是否建立连接。第 7 章将详细讲解快照路由技术和 **dialer watch** 命令。

模拟和设计任何动态链路的网络时，查看整个网络的路由方案很重要。必须考虑是否需要通过备份链路把路由宣告到主网边界。在后续的章节中会看到，RIP 和 IGRP 这样的协议只能接受相同主网的数据，而这些网络的子网掩码必须要和接收、发送路由更新信息接口的子网掩码一致。

表 4-4 列出了基于 cisco 支持建立动态链路路由方案的建议。

表 4-4 基于路由选择协议的动态和静态路由方案

路由选择协议	动态路由更新方案	静态方案
EIGRP	Dialer watch	浮动静态路由 管理距离>170
IGRP	快照路由	浮动静态路由 管理距离>100
OSPF	OSPF 按需电路	浮动静态路由 管理距离>110
RIP 版本 1	快照路由	浮动静态路由 管理距离>120
RIP 版本 2	Dialer watch	浮动静态路由 管理距离>120
IPX RIP/SAP	快照路由	浮动静态路由
IPX EIGRP	快照路由	浮动静态路由
Apple Talk	快照路由	浮动静态路由

解决路由更新难题的一个常用方法就是使用浮动静态路由和加权路由方式。浮动静态路由是一种不将路由条目一直存入路由表的静态路由方式。其路由只在特殊情况下出现。浮动静态路由配置的到达目的网络的路由条目管理距离大于所用的路由选择协议的管理距离，这样使路由更新信息首先选用路由选择协议生成的路由条目。如果通过路由选择协议学到的路由条目信息丢失，浮动静态路由就会进入路由表替代原有路由条目。配置加权或浮动静态路由的命令句法如下：

```
ip route remote_ip_subnet subnet_mask {{ ip_next_hop administrative_distance
```

(1-255)] interface }

所用的加权因子通常是管理距离为 150 到 180。表 4-5 列出了路由选择协议和静态路由所用的管理距离值。如果 **ip route** 命令中没有加入管理距离，就取默认值 1。如果静态路由指向一个接口而不是下一跳的地址，管理距离为 0，也称为直连路由。指向一个可到达的下一跳 IP 地址的静态路由条目会放入路由转发表中。直连路由总是在转发表中，除非其接口为关闭状态。

表 4-5

Cisco 路由器上默认的管理距离

路由源/类型	默认管理距离
直连接口	0
指向接口的静态路由	0
指向下一跳的接口	1
EIGRP 汇总路由	5
EBGP	20
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP-1 和 RIP-2	120
EGP	140
外部 EIGRP	170
内部 BGP	200
未知路由 不可达	255

图 4-6 列出了使用浮动静态路由技术备份串行链路所需要的相关命令。在这个例子中，还需要使用 **IP subnet-zero** 命令，因为在异步链路使用了地址 192.168.1.1/30。异步链路上的子网络号是 192.168.1.0/30，是 C 类网络上的 0 子网。将 DDR 链路设为单独的子网中就不用担心路由选择协议（本例中是 EIGRP）将路由更新信息通告到链路上。这也是另一个控制动态链路路由更新信息的方法。

6. 第 4 步：配置异步接口的按需拨号路由

按需拨号路由（DDR）可以分成以下两种：

- 传统 DDR（legacy DDR）——仅当期望的数据发往目的地时才建立临时性的连接。在物理拨号接口和目的地址之间存在着静态关系时需要配置传统 DDR。
- 带拨号属性的高级 DDR——当多个逻辑连接希望共享同一个物理接口时可以使用这种方式。这种形式的 DDR 用于一个或多个逻辑接口共享一个每次只能接受单个呼叫的物理接口的情况。例如路由器要拨号接入两个不同的远程地址并且运行各自不同的第 3 层协议的情况。有时，路由器可能想要拨入 Internet 且仅运行 IP，而有时路由器

又可能要拨入总部网络中去运行 IP 和 IPX，这也是使用该 DDR 方式的例子。

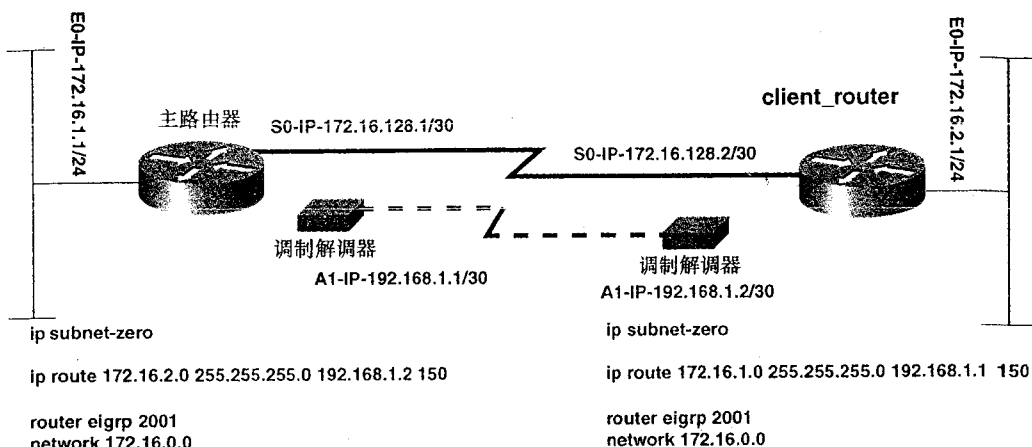


图 4-6 模拟拨号连接的浮动静态路由配置

本节讲述传统 DDR，高级 DDR 将在第 7 章中讲解。

所有的 DDR 方式的工作都是类似的：

- 第 1 步 路由器接收到数据后会检查路由表，以确定是否存在到目的地址的路由。如果存在，找到相应的出站接口。确定接口之后，路由器再次查看以确定数据是否为期望发送的数据。
- 第 2 步 路由器定位下一跳地址并使用 **dialer string** 命令进行拨号，或者在 ISDN TA 环境中使用 **dialer map** 命令建立连接。对于支持 V.25bis 的调制解调器，所拨的号是从 **chat-script** 命令中取出的。V.25bis 是一个用带内信号和 DCE 设备同步的 ITU-T 标准。
- 第 3 步 路由器查看是否有相关的接口处于启动状态且已经连接到远端目的设备。如果接口已经连接，那么数据就会被发送且拨号空闲定时器也会被复位。如果接口没有连接好，路由器就会根据 **dialer string**、**dialer map** 和 **chat-script** 参数的值发送呼叫建立信号到 DCE 设备以建立连接。
- 第 4 步 建立连接时，路由器会将所有的数据送去目的设备，包括期望发送的和限制发送的数据。在路由器期望的数据发送完毕时，路由器会重置空闲定时器。这个由 **dialer idle-timeout** 命令定义的定时器值如果溢出，路由器就会断开连接。

DDR 的配置有 4 个步骤：

- 第 1 步 定义期望业务——有两种方法。整个协议簇可以定义为期望数据，或者可以创建访问控制列表来细化期望数据的范围，这里的所谓期望数据是通过数据类型或目的网络来区分的，可以用下面的命令来完成：

```

dialer-list dialer-group_number protocol protocol_name [permit | deny | list access-list_number]

```

- 第 2 步 启动 DDR 并为接口分配拨号列表——V.25bis 设备上使用 **dialer in-band** 命令以及在接口上使用 **dialer-group** 命令来启动 DDR。**dialer in-band** 命令指定某个接口用于在路由器和外部拨号设备之间建立和断开拨号连接。ISDN 设备使用 D 信道来建立和断开拨号连接，因此对 ISDN BRI 和 PRI 来说，不需要 **dialer**

in-band 命令。**dialer-group** 命令用来为接口分配拨号列表，该列表定义了期望数据。**dialer-group** 命令必须与拨号列表一致，命令如下：

—— **dialer in-band**

—— **dialer-group 1-10**

第 3 步 定义目的参数——包括设置下一跳地址和确定路由器如何到达这些地址，还包括路由器用于认证的名称以及路由器处理 DDR 连接的方式，同时这一步也是设定 **dialer idle-timeout** 参数的好时机。**dialer map** 命令和 **dialer string** 命令都可以用来设置下一跳地址和确定路由器如何到达这些地址。**dialer string** 命令可以和 **dialer remote-name** 命令一起用于 CHAP 认证。

dialer map 命令是派生自 **frame-relay map** 命令，它能使路由器确定如何通过网络协议去访问下一跳地址，目前支持的网络协议包括 AppleTalk、bridging、ISO CLNS、DECnet、HPR、IP、IPX、LLC2、NetBIOS、快照路由选择协议、Banyan VINES 和 XNS。

```
dialer map protocol_name next_hop_address [name remote_device_name]
[class class_name] [speed 56 | 64] [broadcast] [dialer_string]
```

参数 **name** 是连接到路由器接口上的远端路由器的主机名，该参数用于认证。

参数 **speed** 和变量 **dialer_string** 专门用于 ISDN 接口。**speed** 的默认值是设为 64 kbit/s。**dialer_string** 是为 ISDN 准备的，该字符串是下一跳地址所对应的电话号码。对调制解调器这样的异步设备来说，必须要定义一个会话脚本来让路由器将拨号号码字符串传给特定设备。最后，DDR 在默认情况下是非广播协议，如果希望广播业务通过链路，可以加上 **broadcast** 关键字。

命令 **dialer map** 可以使用户用一条命令实现多个 DDR 命令的功能，其中一些命令也可以单独输入执行：

—— **dialer string dialer_string**

—— **dialer remote-name remote_device_name**

—— **dialer idle-timeout seconds**

dialer idle-timeout 是空闲定时器，用于标志上一次期望数据发送后经过的时间。如果在该定时器的计时时间内没有期望数据传输，链路就会被断开。该定时器的默认值是 120 秒。

第 4 步 配置可选的呼叫参数——此时可在接口加入附加的呼叫参数，以下是一些较为有用的选项：

—— **dialer fast-idle seconds**——该命令主要用于多个拨号属性的情况，例如具有拨号属性的 DDR。当两个逻辑拨号接口竞争同一条物理线路时，定时器会指定如果当前呼叫处于空闲状态的情况下，在断开当前呼叫而准备接入下一个用户呼叫之前应该等待的时间。在多个用户对某个链路争夺时应该使用这个命令。第 7 章中会对该命令的用法加以详述。

—— **dialer load-threshold 1-255 [outbound | inbound | either]**——这条命令指定了在某接口上拨号启用另外一个呼叫来连接目的设备的负载阈值。这个值设在 1 到 255 之间，和 **show interface** 命令一样：1 是最低负载，呼叫几乎是立即激活。通过指定数据流量的方向能进行更精确的控制。该命令和 **ppp multilink** 命令一起使用，在下面的章节中会讲到用法。

再看一个以前关于 PPP 的 PAP 认证的例子，现在侧重于整个配置的 DDR 部分。图 4-7 表
子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责

示的是用调制解调器通过 AUX 相连的两台路由器。检测到需送往其静态路由中定义的目的子网去的数据时，路由器会拨通另外一台路由器。这一拨号呼叫会一直保持，除非子网中没有 IP 数据流量的空闲时间超过了 305 秒。这一时间长度是用 **dialer idle-timeout** 命令指定的。

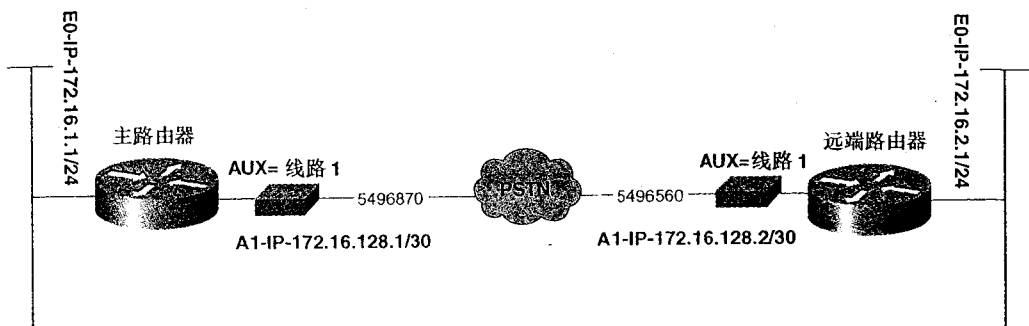


图 4-7 PPP 模拟拨号配置的网络拓扑结构

例 4-11 给出了图 4-7 中的路由器 `host_router` 和 `remote_router` 的配置情况。

例 4-11 PPP 模拟拨号的配置

```
hostname host_router
!
username remote_router password 0 cisco1
ip subnet-zero
chat-script dialremote "" "ATZ&F" OK "ATDT5496561" TIMEOUT 60 CONNECT
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
!
interface Async1
ip address 172.16.128.1 255.255.255.252
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer idle-timeout 305
dialer map ip 172.16.128.2 name remote_router broadcast 5496560
dialer-group 1
async mode interactive
ppp authentication chap
!
ip classless
ip route 172.16.2.0 255.255.255.0 172.16.128.2
!
dialer-list 1 protocol ip permit
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
autoselect ppp
script dialer dialremote
modem InOut
modem autoconfigure type usr_sportster
```

（待续）

```
transport input all
speed 38400
line vty 0 4
login
!
end

host_router#

-----

hostname remote_router
!
username host_router password 0 cisco1
chat-script dialhost "" "ATZ&F" OK "ATDT5496870" TIMEOUT 60 CONNECT
!
interface Ethernet0
ip address 172.16.2.1 255.255.255.0
!
interface Async1
ip address 172.16.128.2 255.255.255.252
encapsulation ppp
async mode interactive
dialer in-band
dialer idle-timeout 305
dialer map ip 172.16.128.1 name host_router broadcast 5496870
dialer-group 1
ppp authentication chap
!
no ip classless
ip route 172.16.1.0 255.255.255.0 172.16.128.1
!
dialer-list 1 protocol ip permit
!
line con 0
line aux 0
autoselect ppp
script dialer dialhost
modem InOut
modem autoconfigure discovery
transport input all
rxspeed 38400
txspeed 38400
line vty 0 4
login
!
end

remote_router#
```

现在回顾在异步接口上配置 PPP 的步骤。这些步骤在异步接口上进行 PPP 配置时非常有用：

- 第 1 步** 调制解调器和异步接口的配置，包括调制解调器的连接与配置，以及路由器异步接口的配置，也包括对应于该异步接口的绝对线路号的标识。在第 1 章中可以查到更多这方面的内容。
- 第 2 步** 在异步接口上定义和配置 PPP。对与第 1 步中的绝对线路号相对应的异步接口进行配置，包括 PPP 数据封装和 PPP 安全认证方式。
- 第 3 步** 配置网络层地址或寻址方式，以及与异步口对应的路由方式。
- 第 4 步** 配置用于 DDR 的异步接口。

4.1.3 PPP 数据压缩的配置

PPP 提供了有效载荷压缩以便在低带宽的链路上获得良好的性能。数据压缩的协商是 LCP 在初始化时进行的。由于要用 LCP 确定压缩的参数，因此在链路的两端都需要进行压缩配置。除了标准的 TCP 数据头压缩，Cisco 还给出了 3 种 PPP 链路上进行有效载荷压缩的方式：

- **预测压缩（Predictor）**——预测压缩是通过一种无损预测算法来实现的，它能学习数据格式并预测数据流的下一字符。该算法被称为无损，因为它可以精确地复制出原始的数据流，而没有任何数据的衰减或丢失。预测压缩算法是内存密集型的算法，需要占用较多的内存但不会占用过多的 CPU 资源。
- **栈式存储算法（Stacker）**——在第 3 章“WAN 协议与技术：高级数据链路控制（HDLC）”中提到过，栈式存储算法是一种基于 Lempel-Ziv（LZ）的数据压缩算法。该算法会建立一个压缩字典的索引，然后通过查询该索引来预测数据流中的下一个字符。路由器每次传送一个数据类型信息，加上该数据类型在数据流中出现的位置。与预测压缩算法相比，栈式存储算法占用较多的 CPU 资源和较少的内存。
- **Microsoft 的点对点压缩（MPPC）**——RFC 2118 阐述了 Microsoft 的点对点压缩（MPPC），MPPC 允许 Cisco 路由器与 Microsoft 客户机进行数据交换。和栈式存储算法 Stacker 一样，MPPC 与预测压缩算法相比占用的较多的 CPU 资源，占用的内存较少。

上面这些数据压缩技术是在接口上用 **compress** 命令进行设置的。使用以下接口命令设置有效载荷压缩：

```
compress [predictor | stac | mppc]
```

启动数据压缩之后，应该重新启动链路或者用 **clear interface interface_name** 命令进行对其重置。这样将强迫路由器或者 LCP 根据所采用的压缩方式重新建立链路。TCP 数据头压缩方式不能和有效载荷压缩一起使用。

RC 1144 中规定，TCP 数据头压缩采用 Van Jacobson 算法。TCP 数据头压缩在有许多小数据包时非常有用，能够有效的减小这些数据包所需要的 TCP 系统开销。用以下命令设置启用 TCP 数据头压缩：

```
ip tcp header-compression [passive]
```

参数 **passive** 只有在接口上入站 TCP 数据包压缩的情况下才对出站的 TCP 数据包进行压缩。如果没有指定 **passive** 参数，那么所有的数据都会被压缩。

如果传输文本文件或 ASCII 这类能够高度压缩的数据，压缩的效率是非常高的。而像 JPEG 或 MPEG 这种已经经过了很大压缩的文件只会减慢路由器的工作和该类数据的传输。压缩也会增加 CPU 处理时间以及内存开销。因此，在允许进行数据压缩之前，要仔细考虑当前的内存和处理器的使用情况。

判断是否能使用数据压缩的两条命令是 **show processes** 和 **show processes buffers**。例 4-12 列出了 **show processes** 命令的输出结果，从这里能够看出路由器的平均利用率为 30% 的系统资源。在这个例子中，路由器还有足够的资源来进行数据压缩的操作。如果路由器的利用率已经达到了 65%，那么一定要小心地使用数据压缩。

例 4-12 show processes 命令的运行结果

```
skynet_2#show processes
```

CPU utilization for five seconds: 29%/2%; one minute: 30%; five minutes: 32%

PID	QTY	PC	Runtime (ms)	Invoked	uSecs	Stacks	TTY	Process
1	Csp	2E68C	1132	67278	16	3760/4096	0	Load Meter
2	ME	122648	8882756	6695839	132610432/12288	0	Exec	
3	Lst	17E918	494620	45735	10814	7960/8192	0	Check heaps
4	Cwe	183AE8	0	1	0	7840/8192	0	Pool Manager
5	Mst	11A2E8	4	2	2000	7808/8192	0	Timers
6	Lwe	1D0124	219292	342622	640	7528/8192	0	ARP Input
7	Mwe	23D924	892	1668	534	7144/8192	0	DDR Timers
8	Mwe	24DA90	0	2	0	11920/12288	0	Dialer event
9	Lwe	260FC0	0	1	0	7856/8192	0	Entity MIB API
10	Mwe	429FE4	0	2	0	7816/8192	0	Serial Background
11	Mwe	42DF08	4	1	4000	7856/8192	0	SERIAL A'detect
12	Cwe	188D94	0	1	0	7848/8192	0	Critical Bkgnd
13	Mwe	1A0690	16212	49282	32810888/12288	0	Net Background	
14	Lwe	11090C	6900	19588	35211752/12288	0	Logger	
15	Msp	12C25C	274660	335655	818	7472/8192	0	TTY Background
16	Msp	19FD38	6244	335694	18	7800/8192	0	Per-Second Jobs
17	Mwe	F0DB0	8092	1186590	6	7648/8192	0	LED Timers
18	Mwe	4BB398	0	23	0	7976/8192	0	CSM timer process
19	Mwe	4BE740	212	574	369	7736/8192	0	POTS
20	Mwe	2004B8	16908	44367	381	7544/8192	0	CDP Protocol
21	Mrd	2B434C	272492	211068	129110672/12288	0	IP Input	

注释 为了节省需要占用路由器资源的数据压缩所消耗系统开销，Cisco 提供了硬件加速卡，或者叫做压缩服务适配卡（CSA）。如果路由器中配有 CSA，那么压缩操作就是在 CSA 板上完成的。

4.1.4 配置多链路捆绑 PPP

多链路捆绑 PPP 是将多个物理链路合并或者捆绑成一个大的逻辑链路的机制。这种逻辑的端到端连接称为一个 **捆绑链路**。这种捆绑链路能够增加数据传输的带宽，并且由于允许将数据进行拆分后再通过多个不同的链路同时送往同一个目的设备从而减少数据传送的延迟。

最常见的多链路 PPP 用于 ISDN，它使用多链路 PPP 将两个 64 kbit/s 的 B 信道捆绑在一起形成一个 128 kbit/s 的连接。尽管 ISDN 是多链路 PPP 最常见的应用方式，它同样也适用于以下接口：

- 异步接口。
 - 同步接口。
 - ISDN 接口。
- ISDN BRI
- ISDN PRI

图 4-8 演示了多链路 PPP 工作模型。

捆绑链路形成之后，它可以将一些其他类型的接口合并在一起使用。例如，一个异步通信接口可以和一个同步通信接口组成一个捆绑链路。

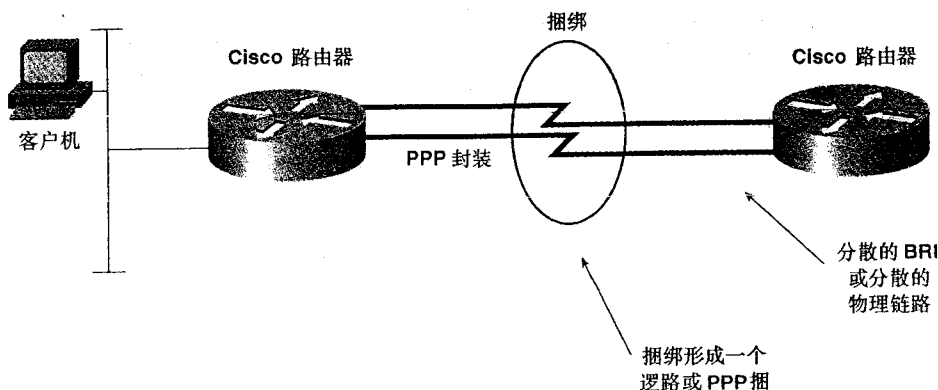


图 4-8 多链路 PPP 捆绑

注释 多链路捆绑 PPP 是在 RFC 1990 中制定的，它替代了原来的 RFC 1717。RFC 1990 实现了不同供应商设备之间相互协调工作，并且解决了早期 ISDN 的序列编号问题。

多链路 PPP 是由 LCP 在其初始化时设置的一个功能选项。选定后，多链路 PPP 会将数据包分割成小块的片段，然后将这些数据片段同时送到多个点对点连接上的同一个远端地址。这些数据片段到达连接的另一端之后，LCP 会将它们再恢复成完整的数据包。

可以在接口或拨号设备上使用下列命令对多链路 PPP 进行配置：

```
ppp multilink
dialer load-threshold load [inbound | outbound | either]
```

dialer load-threshold 命令指定了一个阈值，当负载超过这一阈值时，拨号设备开始初始化另一呼叫。阈值可以用 **show interface** 命令查看，其范围是从 1 到 255，1 代表最小阈值，使得拨号设备几乎立即开始拨号呼叫。指定数据流量方向可以进行更精确的控制。

注释 Cisco IOS 11.1 是第 1 个支持多链路 PPP 的 IOS 版本。Cisco IOS 11.3 则包含称为带宽分配控制协议（BACP）的功能部件。BACP 使两台路由器协商决定在通信中那条链路加入或退出对等连接。

可以使用 **debug ppp negotiation** 的语句结合 **ping** 命令验证多链路 PPP 的功能。通过打开 PPP 的调试模式可知道 LCP 何时打开了另外一条信道或链路。执行扩展 **ping** 命令可以产生大量的数据快速通过链路，从而超过路由器能够产生新呼叫的阈值，最终使其建立第 2 条 PPP 捆绑链路。命令 **show ppp multilink** 也可以显示多链路会话状态。例 4-13 给出 **show ppp multilink** 命令的用法，其后执行扩展 **ping** 命令，这使得路由器建立第 2 条捆绑链路。

例 4-13 show ppp multilink 命令的执行结果

```
skynet_2#show ppp multilink

Virtual-Access1, bundle name is cns_isdn_callback
Dialer interface is Dialer1
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 1 (max not set, min not set)
BRI0:1                               ←Only one active bundle
```

(待续)

```

skynet_2#ping
Protocol [ip]: ip
Target IP address: 172.16.16.2
Repeat count [5]: 1000
Datagram size [100]: 2000
Timeout in seconds [2]: 5
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 1000, 2000-byte ICMP Echos to 172.16.16.2, timeout is 5 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<<<text omitted>>>
!!!!
Success rate is 100 percent (370/370), round-trip min/avg/max = 152/161/284 ms

skynet_2#show ppp multilink

Virtual-Access1, bundle name is cns_isdn_callback
Dialer interface is Dialer1
  0 lost fragments, 0 reordered, 0 unassigned, sequence 0x496/0x2F1 rcvd/sent
  0 discarded, 0 lost received, 25/255 load
Member links: 2 (max not set, min not set)
  BRI0:1          <-Both BRIs are in the bundle
  BRI0:2
skynet_2#

```

4.1.5 PPP 的 LAPB 和 LQM 的配置

目前有两种方法可以加强 Cisco 路由器的 PPP 链路质量。一种通过 **ppp reliablelink** 命令使用平衡式链路接入过程（LAPB）数字模式协商。另一种通过 **ppp quality** 命令在链路上强制使用链路质量监控（LQM）。

LQM 能够监视链路的质量，如果其质量降到某个比率，路由器会断开链路。这里的比率是通过计算入站业务和出站业务获得的。输出 LQM 的比率是通过发送的数据包和字节的数目和对方接收到的数据包和字节的数目来计算的。入站的 LQM 的比率是比较接收到的数据包和字节的数目以及对方发送的数据包和字节的数目来计算的。

通过使用 LQM，会发送链路质量报告（LQR）来代替用于保持连接的 **keepalive**。LQR 是在 RFC 1989 “PPP Link Quality Monitoring” 中定义的。

注释 LAPB 数字模式提供了重传纠错的功能，而 LQM 只监视链路质量。

使用以下命令设置启动 LQM：

```
ppp quality percentage_of_successful_traffic
```

另一增加链路可靠性的方法是使用 **ppp reliable-link** 命令。该命令使路由器提供 LAPB 数字模式协商，它在数据链路层能够提供适合很多上层网络协议（如本例的 PPP）的错误检测机制。

链路两端都必须使用 PPP 的可靠连接以保证整个链路的可靠性。与 **ppp quality** 命令不同的是，使用 **ppp reliable-link** 的时候也可以使用数据压缩。但是不能将 **PPP multilink** 命令用于 PPP 可靠链路中。使用 **debug ppp negotiation** 命令可以查验命令是否生效，其结果里列出了 LQM 的情况。

4.1.6 PPP 和 DDR 的“Big show”和“Big D”

Cisco 提供了一些功能很强大的命令来调试 PPP 和 DDR 会话。本节讲述一些非常有用的 **show** 和 **debug** 命令。PPP 和 DDR 的“big show”和“Big D”如下：

- **show interface interface_name**——**show interface** 命令提供了接口物理状态的有用信息。配置为 DDR 的接口总是显示为 UP 状态或电子欺骗模式。
- **show line x**——**show line** 命令提供了接口物理状态的有用信息，还提供了部分逻辑状态信息。请参见第 1 章对该命令的详细讲解。
- **show ip route**——用 **show ip route** 命令可以验证 PPP 子网是否处在 up 和 active 状态。只有链路两端真正建立连接时子网状态才会显示“connected”。
- **debug ppp negotiation**——这条命令可能是 PPP 会话调试中最为有用的。它能够显示 LCP 协商过程中的每条信息。例 4-14 显示了路由器之间建立 PPP 会话的过程。IPCP 以打开 IP 协议为结束协商，可以看到每个 PPP 选项和指定的协议都有一个 LCP 协商过程。
- **debug ppp authentication**——该命令能显示 PPP 认证错误，如 CHAP 和 PAP 认证的错误。例 4-15 给出 PPP 认证失败的例子。
- **debug dialer**——**debug dialer** 命令主要是用于拨号开始时的调试。正如该命令所显示，只有 DDR 和访问列表配置正确时才能正确的进行拨号。

下面是“big Show”和“big D”命令的使用实例。由于这些调试过程能清楚地告诉用户正在发生的事情，因此这些命令应该配合在一起使用。例 4-14 中启动了异步状态下上面所列的调试命令。

例 4-14 成功的 debug ppp negotiation 命令的执行结果

```
01:01:57: Async1: Dialing cause ip (s=172.16.128.2, d=172.16.1.1) ←Dial is
started, Dialer-list is OK.
01:01:57: Async1: Attempting to dial 5496561
01:01:57: CHAT1: Attempting async line dialer script
01:01:57: CHAT1: Dialing using Modem script: dialhost & System script: none
01:01:57: TTY1: cleanup pending. Delaying DTR
01:01:57: CHAT1: process started
01:01:57: CHAT1: Asserting DTR
01:01:57: TTY1: Set DTR to 1
01:01:57: CHAT1: Chat script dialhost started.....
01:02:14: CHAT1: Chat script dialhost finished, status = Success ←Modem connected
01:02:14: TTY1: destroy timer type 1
01:02:14: TTY1: destroy timer type 0
01:02:14: As1 PPP: Async Protocol Mode started for 172.16.128.1 ←PPP started
01:02:14: As1 AAA/ACCT: Using PPP accounting list ""
01:02:14: As1 IPCP: Install route to 172.16.128.1
01:02:16: %LINK-3-UPDOWN: Interface Async1, changed state to up
01:02:16: As1 PPP: Treating connection as a callout
01:02:16: As1 PPP: Phase is ESTABLISHING, Active Open ←PPP negotiation begins
01:02:16: As1 LCP: O CONFREQ [Closed] id 21 len 20
01:02:16: As1 LCP: ACCM 0x000A0000 (0x0206000A0000)
01:02:16: As1 LCP: MagicNumber 0x0069B38A (0x05060069B38A)
01:02:16: As1 LCP: PFC (0x0702)
01:02:16: As1 LCP: ACFC (0x0802)
01:02:18: As1 LCP: TIMEOUT: State REQsent
```

(待续)

```

01:02:18: As1 LCP: O CONFREQ [REQsent] id 22 len 20
01:02:18: As1 LCP:   ACCM 0x000A0000 (0x0206000A0000)
01:02:18: As1 LCP:   MagicNumber 0x0069B38A (0x05060069B38A)
01:02:18: As1 LCP:   PFC (0x0702)
01:02:18: As1 LCP:   ACFC (0x0802)
01:02:20: As1 LCP: TIMEOUT: State REQsent
01:02:20: As1 LCP: O CONFREQ [REQsent] id 23 len 20
01:02:20: As1 LCP:   ACCM 0x000A0000 (0x0206000A0000)
01:02:20: As1 LCP:   MagicNumber 0x0069B38A (0x05060069B38A)
01:02:20: As1 LCP:   PFC (0x0702)
01:02:20: As1 LCP:   ACFC (0x0802)
01:02:20: As1 LCP: I CONFACK [REQsent] id 23 len 20
01:02:20: As1 LCP:   ACCM 0x000A0000 (0x0206000A0000)
01:02:20: As1 LCP:   MagicNumber 0x0069B38A (0x05060069B38A)
01:02:20: As1 LCP:   PFC (0x0702)
01:02:20: As1 LCP:   ACFC (0x0802)
01:02:20: As1 LCP: I CONFREQ [ACKrcvd] id 180 len 25
01:02:20: As1 LCP:   ACCM 0x000A0000 (0x0206000A0000)
01:02:20: As1 LCP:   AuthProto CHAP (0x0305C22305)
01:02:20: As1 LCP:   MagicNumber 0x0A548C93 (0x05060A548C93)
01:02:20: As1 LCP:   PFC (0x0702)
01:02:20: As1 LCP:   ACFC (0x0802)
01:02:20: As1 LCP: O CONFACK [ACKrcvd] id 180 len 25
01:02:20: As1 LCP:   ACCM 0x000A0000 (0x0206000A0000)
01:02:20: As1 LCP:   AuthProto CHAP (0x0305C22305)
01:02:20: As1 LCP:   MagicNumber 0x0A548C93 (0x05060A548C93)
01:02:20: As1 LCP:   PFC (0x0702)
01:02:20: As1 LCP:   ACFC (0x0802)
01:02:20: As1 LCP: State is Open
01:02:20: As1 PPP: Phase is AUTHENTICATING, by the peer      ←CHAP begins
01:02:20: As1 CHAP: I CHALLENGE id 39 len 32 from "host_router" ←CHAP challenge
01:02:20: As1 CHAP: O RESPONSE id 39 len 34 from "remote_router"
01:02:20: As1 CHAP: I SUCCESS id 39 len 4      ←CHAP OK
01:02:20: As1 PPP: Phase is UP
01:02:20: As1 IPCP: O CONFREQ [Closed] id 9 len 10      ←IP Parameters
01:02:20: As1 IPCP:   Address 172.16.128.2 (0x0306AC108002)
01:02:20: As1 CDP: O CONFREQ [Closed] id 9 len 4
01:02:20: As1 IPCP: I CONFREQ [REQsent] id 22 len 10
01:02:20: As1 IPCP:   Address 172.16.128.1 (0x0306AC108001)
01:02:20: As1 IPCP: O CONFACK [REQsent] id 22 len 10
01:02:20: As1 IPCP:   Address 172.16.128.1 (0x0306AC108001)
01:02:20: As1 CDP: I CONFREQ [REQsent] id 22 len 4
01:02:20: As1 CDP: O CONFACK [REQsent] id 22 len 4
01:02:20: As1 IPCP: I CONFACK [ACKsent] id 9 len 10
01:02:20: As1 IPCP:   Address 172.16.128.2 (0x0306AC108002)
01:02:20: As1 IPCP: State is Open      ←IP OK
01:02:20: dialer Protocol up for As1
01:02:20: As1 CDP: I CONFACK [ACKsent] id 9 len 4
01:02:20: As1 CDP: State is Open
01:02:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to
up
remote_router#

```

例 4-15 是使用 **debug ppp authentication** 命令显示 PPP CHAP 认证失败。

例 4-15 显示 PPP CHAP 失败

```

skynet_lab#debug ppp authentication
Se0 PPP: Phase is AUTHENTICATING, by the peer
Se0 CHAP: I CHALLENGE id 51 len 31 from "skynet_lab"

```

(待续)

```
Se0 CHAP: O RESPONSE id 51 len 31 from "isp_router"
Se0 CHAP: I FAILURE id 51 len 25 msg is "MD/DES compare failed"
Se0 PPP: Phase is AUTHENTICATING, by the peer
Se0 CHAP: I CHALLENGE id 52 len 31 from "skynet_lab"
Se0 CHAP: O RESPONSE id 52 len 31 from "isp_router"
```

debug ppp negotiation 命令是 PPP 调试中最为有用的命令之一，不需要太多数据却提供了足够的信息解决大多数与 PPP 有关的问题或缩小了问题的范围。

注释 技术，技巧，方法

这里的例子和实验使用的路由选择协议以及 VLSM 似乎过于复杂，但这是有原因的，多数教材和范例都使用标准的 24 位地址，但很少关心寻址方式和其他技术问题。这样会使得网络工程师们在 IP 地址设计时形成不良习惯。本书中，IP 地址的分配和大家在实际工作中会遇到和使用的方式一样。不断通过这样的例子进行强化能够提高技术水平。一个武术家对其每一个动作哪怕是最微小的细节也会加以注意，同样，如果不知道配置过程中所用的某个命令的目的和作用，就应该问自己“这条命令是必需的吗？”

4.1.7 PPP 回拨设置

PPP 回拨是指远程或是正在呼叫的路由器将其拨号呼叫传到回拨服务器上，然后中止链路，再等待从回拨服务器接收呼叫的过程。该方法可用于加强安全管理以及对链路一端获得连接的时间进行控制。第 7 章将详细讲述 PPP 回拨的问题。

4.2 实验 10：在异步拨号连接上配置 PPP、PAP 和数据压缩——第 1 部分

4.2.1 实验说明

这一章讲述的是占主导地位的 Internet 访问协议——PPP 的内容。PPP 不仅用于 Internet 的访问，同时还广泛用于很多的远程交换和专用网络的远程访问等领域。很多公司的应用，如 Citrix 或 WIN 以及大多数的电子邮件服务器，都要求以 IP 为其网络层协议。要访问这些应用，就有必要在远程客户工作站上运行 IP 协议，而其数据链路层的协议就是 PPP。

4.2.2 实验内容

假设我们的 Skynet 测试实验室是一个隐秘的网络测试点，在这里会对新的 Cisco IOS 的性能进行测试，会建立很多类型的 Internet 网络基础设施。有时候，一些工程师需要安全的远程访问 Skynet。目前，只有一个代号为 JP 的工程师可以访问这个实验室。现在我们的任务就是配置一个安全的远程访问连接。设计的时候要遵照下面这一些准则：

- 客户端是 Windows 95/98/2000 环境，其 IP 地址应该是设置成动态的。

- 用户 ID 是 JP，密码是 trashman。
- JP 要求运行一些需要 IP 服务的 Visual Basic 程序，他要求 IP 访问主机 172.16.1.10，该主机属于实验室本地以太网段。

4.2.3 实验目的

- 如图 4-9 配置访问服务器和网络。
- 在异步连接上使用 PPP。
- 利用指定的主机名和密码设置适当的 Windows 95/98/2000 客户端认证协议。
- 设定连接在 10 分钟内没有操作就断开。
- 允许建立到访问服务器的一个 TTY 会话，以便服务器可以通过一个标准的终端仿真软件，如超级终端来控制路由器。
- 可选：增强连接的性能。

4.2.4 所需设备

- 一台 Cisco 路由器，最好是一台访问服务器，一条 Cisco 终端电缆，标有 MODEM 的接线头以及一台调制解调器。
- 一台配有 modem 的 Windows 95/98/2000 工作站。
- 可选：要建立图 4-9 所示的网络模型，还需要一台以太集线器以及另外一台工作站。但是这些不是本实验必需的。

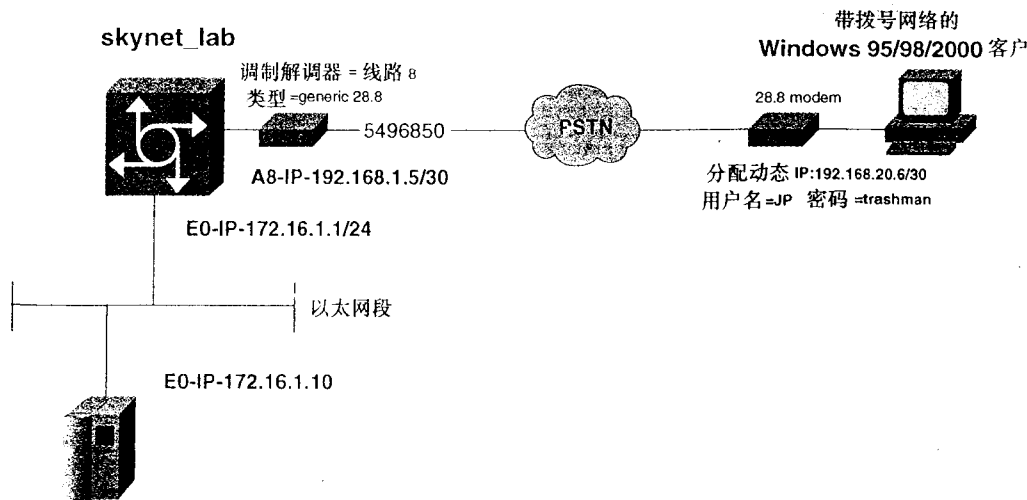


图 4-9 通过 PPP 访问 Skynet 测试实验室

4.2.5 物理设计和实验准备

- 如图 4-9，连接好路由器、调制解调器、集线器以及工作站。

证方面，使用 JP 的用户 ID 和密码 trashman。记住，这是要区分大小写的。

4.3 实验 10：在异步拨号连接上配置 PPP、PAP 和数据压缩——第 2 部分

4.3.1 实验步骤

用 Cisco 终端电缆将调制解调器与访问服务器或辅助设备端口连接在一起。用标有 Modem 的接线头来把调制解调器与线缆接起来，这个实验中用来与调制解调器连接的是访问服务器的线路 8 或称端口 8。

配置访问服务器 skynet_lab 时，可以参照本章中讨论过的异步连接上配置 PPP 的 4 步骤过程。对此过程稍加改动，在这个实验中的操作步骤如下：

第 1 步 进行访问服务器的初始设置，包括主机名 skynet_lab 的确定，enable password 的设置以及图中以太接口的配置。为了远程客户测试的方便，再在路由器上对 Telnet 和反向 Telnet 进行配置。

第 2 步 将线路 8 配置为调制解调器之用，并设置登录的时候封装方式为自动选择。

第 3 步 为异步接口设置 IP 地址和配置 PPP 的 PAP 认证方式与用户地址协商机制，包括用户名 JP 和密码 trashman。

第 4 步 对用于 DDR 的异步接口进行配置。

第 5 步 可选：设置 MPPC 压缩。

首先，对访问服务器做一些初始化设置，包括设置主机名和特权密码：

```
Router (config) #hostname skynet_lab
skynet_lab (config) #enable password cisco
```

接着，用 show line 命令确认调制解调器要使用的绝对线路号。如果启动了 Telnet 访问，还要注意一下 vty 线路号。例 4-16 就是 show line 命令的示例。注意一下端口 8 的线路号，它是 TTY 端口 8。Telnet 会话使用的线路号为 18 至 22。

例 4-16 show line 命令的显示结果

skynet_lab#show line										
Tty Typ	Tx/Rx	A	Modem	Roty	Acc0	Acc1	Uses	Noise	Overruns	Int
* 0 CTY		-	-	-	-	-	0	0	0/0	-
1 TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
* 2 TTY	9600/9600	-	-	-	-	-	0	1	9/27	-
3 TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
4 TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
5 TTY	9600/9600	-	-	-	-	-	0	1	0/0	-
6 TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
7 TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
8 TTY	9600/9600	-	-	-	-	-	0	0	0/0	← Modem port
9 TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
10 TTY	9600/9600	-	-	-	-	-	0	0	0/0	-

(待续)

11	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
12	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
13	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
14	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
15	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
16	TTY	9600/9600	-	-	-	-	-	0	0	0/0	-
17	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
18	VTY							0	0	0/0	←Telnet lines
19	VTY							0	0	0/0	
20	VTY							0	0	0/0	
21	VTY							0	0	0/0	
22	VTY							0	0	0/0	

skynet_lab#

接下来对调制解调器和 PPP 使用的线路 8 进行配置。用 **autoselect during-login** 命令可以选择线路上开始会话时的会话类型。如果客户端是用终端仿真软件拨入的，那开始的是 **TTY** 会话。如果客户端开始的是一个 PPP 会话，那就会协商确定一个 PPP 会话，为了启动 PPP 会话，还需要一条 **autoselect ppp** 命令。由于不知道调制解调器的类型，配置线路 8 上的调制解调器的时候需要一条 **autoconfigure type default** 命令。例 4-17 是调制解调器和 PPP 使用的线路 8 的配置示例。

例 4-17 配置 Modem 和 PPP 使用的线路 8

```
skynet_lab#conf t
Enter configuration commands, one per line. End with CNTL/Z.
skynet_lab(config)#line 8
skynet_lab(config-line)#modem inout
skynet_lab(config-line)#modem autoconfigure type default
skynet_lab(config-line)#login local
skynet_lab(config-line)#autoselect during-login
skynet_lab(config-line)#autoselect ppp
skynet_lab(config-line)#transport input all
skynet_lab(config-line)#^Z
skynet_lab#
```

要验证调制解调器是否已经正确配置，可以用 **show line x** 命令（这里是 **show line 8**）来确认。这条命令可以检查调制解调器是否配置好。而 **debug confmodem** 命令则可以用来确认调制解调器是否可以正常接收 **AT** 字符串命令。可以回顾第一章关于调制解调器的调试问题，例如反向 Telnet 等。

然后应该配置异步接口。异步接口必须要和调制解调器使用的线路号（这里是 8）相匹配。这个例子中还需要配置 PPP 和 PAP 认证所要使用的异步接口。Windows 95/98 的客户机是用 PAP 认证方式对 PPP 进行认证的。PPP 封装形式和 PPP 认证方式的配置如例 4-18 所示。

例 4-18 配置 PPP 封装形式和 PPP 认证方式

```
skynet_lab#conf t
Enter configuration commands, one per line. End with CNTL/Z.
skynet_lab(config)#int a8
skynet_lab(config-if)#encapsulation ppp
skynet_lab(config-if)#ppp authentication pap
skynet_lab(config-if)#exit
skynet_lab(config)#username JP password trashman
skynet_lab#
```

如例 4-19 所示，接下是异步接口 IP 地址的设置，同时还要用 **peer default ip address remote_ip_address** 命令为拨入路由器的 PPP 客户 PC 分配 IP 地址。

例 4-19 配置本地和远程 IP 地址

```
skynet_lab(config)#int a8
skynet_lab(config-if)#ip address 192.168.1.5 255.255.255.252
skynet_lab(config-if)#peer default ip address 192.168.1.6
```

现在，可以配置需要使用 DDR 的路由器。对于 DDR 来说，需要在异步接口上用下面这些命令对其进行配置：

- **dialer in-band**
- **dialer idle-timeout** *x*
- **dialer-group** *x*
- **async mode interactive**

命令 **dialer in-band** 是允许 V.24bis 的拨号方式，而 **dialer idle-timeout** 和 **dialer-group** 命令则是定义用户感兴趣的数据以及确定线路空闲的情况下过多长时间将线路断开。最后，**async mode interactive** 命令则是允许呼入的连接。

在全局配置模式中，还需要一条 **dialer-list 8 protocol ip permit** 命令。由于这个例子中的路由器不会通过连接去启动呼叫或者是路由，因此这个配置里的拨号列表非常简单。例 4-20 是 DDR 的配置示例。

例 4-20 配置 DDR

```
skynet_lab(config)#int a8
skynet_lab(config-if)#dialer in-band
skynet_lab(config-if)#dialer idle-timeout 600
skynet_lab(config-if)#dialer-group 8
skynet_lab(config-if)#async mode interactive
skynet_lab(config-if)#exit
skynet_lab(config)#dialer-list 8 protocol ip permit
skynet_lab(config)#
```

上面这些配置完成之后，就可以对其进行测试了。第 1 章提过，用 **debug confmodem** 命令和 **debug modem** 命令可以监测调制解调器状态。例 4-21 是这条命令在这个例子中的执行示例。这里可看到路由器已经成功地向调制解调器发送了 AT 命令。

例 4-21 debug modem 和 debug confmodem 命令示例

```
skynet_lab#debug modem
skynet_lab#debug confmodem
d06h: TTY8: Line reset by "Exec"
1d06h: TTY8: Modem: IDLE->HANGUP
1d06h: TTY8: destroy timer type 0
1d06h: TTY8: destroy timer type 1
1d06h: TTY8: destroy timer type 3
1d06h: TTY8: destroy timer type 4
1d06h: TTY8: destroy timer type 2
1d06h: TTY8: dropping DTR, hanging up
1d06h: tty8: Modem: HANGUP->IDLE
1d06h: TTY8: restoring DTR
1d06h: TTY8: autoconfigure probe started
```

(待续)

```

1d06h: TTY8: Modem command: AT&F&C1&D250=1H0-
1d06h: TTY8: Modem configuration succeeded
1d06h: TTY8: Detected modem speed 115200
1d06h: TTY8: Done with modem configuration

```

要验证物理层的工作，可以对网络层进行测试。首先是打开 PPP 的 “Big D”: **debug ppp Negotiation**、**debug ppp authentication** 和 **debug ppp error**。然后再通过一台 Windows 95/98/2000 工作站建立到路由器的 PPP 会话。

这个实验里，需要确认 4 个任务是否成功完成：

- PPP 的初始化——也就是说，调制解调器工作正常并接收到第一个 PPP 字符串。
- LCP 可以完成 PPP 协商确定过程。
- PAP 可以成功地完成。
- 远程终端能够通过 IPCP 获得正确的 IP 地址。

例 4-22 就是 **debug** 的输出结果，它显示所有的 4 个工作阶段都已经成功的完成，同时 IP 地址也正确的分配给 Windows 95/98 客户。

例 4-22 一次成功连接的 debug 输出结果

```

skynet_lab#debug ppp negotiation
skynet_lab#debug ppp authentication
skynet_lab#debug ppp error
1d06h: As8 IPCP: Install route to 192.168.1.6
1d06h: %LINK-3-UPDOWN: Interface Async8, changed state to up
1d06h: As8 PPP: Treating connection as a callin
1d06h: As8 PPP: Phase is ESTABLISHING, Passive Open <PPP Initializes
1d06h: As8 LCP: State is Listen <-LCP Initializes
1d06h: As8 LCP: I CONFREQ [Listen] id 3 len 23
1d06h: As8 LCP: ACCM 0x000A0000 (0x0206000A0000)
1d06h: As8 LCP: MagicNumber 0x00F1EF7A (0x050600F1EF7A)
1d06h: As8 LCP: PFC (0x0702)
1d06h: As8 LCP: ACFC (0x0802)
1d06h: As8 LCP: Callback 6 (0x0D0306)
1d06h: As8 LCP: O CONFREQ [Listen] id 15 len 24
1d06h: As8 LCP: ACCM 0x000A0000 (0x0206000A0000)
1d06h: As8 LCP: AuthProto PAP (0x0304C023)
1d06h: As8 LCP: MagicNumber 0xE7427D86 (0x0506E7427D86)
1d06h: As8 LCP: PFC (0x0702)
1d06h: As8 LCP: ACFC (0x0802)
1d06h: As8 LCP: O CONFREQ [Listen] id 3 len 7
1d06h: As8 LCP: Callback 6 (0x0D0306)
1d06h: As8 LCP: I CONFREQ [REQsent] id 4 len 20
1d06h: As8 LCP: ACCM 0x000A0000 (0x0206000A0000)
1d06h: As8 LCP: MagicNumber 0x00F1EF7A (0x050600F1EF7A)
1d06h: As8 LCP: PFC (0x0702)
1d06h: As8 LCP: ACFC (0x0802)
1d06h: As8 LCP: O CONFACK [REQsent] id 4 len 20
1d06h: As8 LCP: ACCM 0x000A0000 (0x0206000A0000)
1d06h: As8 LCP: MagicNumber 0x00F1EF7A (0x050600F1EF7A)
1d06h: As8 LCP: PFC (0x0702)
1d06h: As8 LCP: ACFC (0x0802)
1d06h: As8 LCP: TIMEOUT: State ACKsent
1d06h: As8 LCP: O CONFREQ [ACKsent] id 16 len 24
1d06h: As8 LCP: ACCM 0x000A0000 (0x0206000A0000)
1d06h: As8 LCP: AuthProto PAP (0x0304C023)

```

(待续)

```

1d06h: As8 LCP: MagicNumber 0xE7427D86 (0x0506E7427D86)
1d06h: As8 LCP: PFC (0x0702)
1d06h: As8 LCP: ACFC (0x0802)
1d06h: As8 LCP: I CONFACK [ACKsent] id 16 len 24
1d06h: As8 LCP: ACCM 0x000A0000 (0x0206000A0000)
1d06h: As8 LCP: AuthProto PAP (0x0304C023)
1d06h: As8 LCP: MagicNumber 0xE7427D86 (0x0506E7427D86)
1d06h: As8 LCP: PFC (0x0702)
1d06h: As8 LCP: ACFC (0x0802)
1d06h: As8 LCP: State is Open
1d06h: As8 LCP: LCP completes with OPEN state
1d06h: As8 PPP: Phase is AUTHENTICATING, by this end
1d06h: As8 PAP: I AUTH-REQ id 1 len 16 from "JP"
1d06h: As8 PAP: Authenticating peer JP
1d06h: As8 PAP: O AUTH-ACK id 1 len 5
1d06h: As8 PPP: Phase is UP ←PAP completes
1d06h: As8 IPCP: O CONFREQ [Closed] id 9 len 10 ←IPCP begins IP setup
1d06h: As8 IPCP: Address 192.168.1.5 (0x0306C0A80105)
1d06h: As8 CDPCP: O CONFREQ [Closed] id 4 len 4
1d06h: As8 IPCP: I CONFREQ [REQsent] id 1 len 40
1d06h: As8 IPCP: CompressType VJ 15 slots CompressSlotID (0x0206002D0F01)
1d06h: As8 IPCP: Address 0.0.0.0 (0x030600000000)
1d06h: As8 IPCP: PrimaryDNS 0.0.0.0 (0x810600000000)
1d06h: As8 IPCP: PrimaryWINS 0.0.0.0 (0x820600000000)
1d06h: As8 IPCP: SecondaryDNS 0.0.0.0 (0x830600003000)
1d06h: As8 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
1d06h: As8 IPCP: O CONFREQ [REQsent] id 1 len 34
1d06h: As8 IPCP: CompressType VJ 15 slots CompressSlotID (0x0206002D0F01)
1d06h: As8 IPCP: PrimaryDNS 0.0.0.0 (0x810600000000)
1d06h: As8 IPCP: PrimaryWINS 0.0.0.0 (0x820600000000)
1d06h: As8 IPCP: SecondaryDNS 0.0.0.0 (0x830600000000)
1d06h: As8 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
1d06h: As8 CCP: I CONFREQ [Not negotiated] id 1 len 15
1d06h: As8 CCP: MS-PPC supported bits 0x00000001 (0x120600000001)
1d06h: As8 CCP: Stacker history 1 check mode EXTENDED (0x1105000104)
1d06h: As8 LCP: O PROTREQ [Open] id 17 len 21 protocol CCP
1d06h: As8 LCP: (0x80FD0101000F12060000000111050001)
1d06h: As8 LCP: (0x04)
1d06h: As8 IPCP: I CONFACK [REQsent] id 9 len 10
1d06h: As8 IPCP: Address 192.168.1.5 (0x0306C0A80105)
1d06h: As8 LCP: I PROTREQ [Open] id 5 len 10 protocol CDPCP (0x820701040004)
1d06h: As8 CDPCP: State is Closed
1d06h: As8 IPCP: TIMEOUT: State ACKrcvd
1d06h: As8 IPCP: Address 192.168.1.5 (0x0306C0A80105)
1d06h: As8 IPCP: I CONFACK [REQsent] id 10 len 10
1d06h: As8 IPCP: Address 192.168.1.5 (0x0306C0A80105)
1d06h: As8 IPCP: I CONFREQ [ACKrcvd] id 2 len 34
1d06h: As8 IPCP: Address 0.0.0.0 (0x030600000000)
1d06h: As8 IPCP: PrimaryDNS 0.0.0.0 (0x810600000000)
1d06h: As8 IPCP: PrimaryWINS 0.0.0.0 (0x820600000000)
1d06h: As8 IPCP: SecondaryDNS 0.0.0.0 (0x830600000000)
1d06h: As8 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
1d06h: As8 IPCP: O CONFREQ [ACKrcvd] id 2 len 28
1d06h: As8 IPCP: PrimaryDNS 0.0.0.0 (0x810600000000)
1d06h: As8 IPCP: PrimaryWINS 0.0.0.0 (0x820600000000)
1d06h: As8 IPCP: SecondaryDNS 0.0.0.0 (0x830600000000)
1d06h: As8 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
1d06h: As8 IPCP: I CONFREQ [ACKrcvd] id 3 len 10
1d06h: As8 IPCP: Address 0.0.0.0 (0x030600000000)
1d06h: As8 IPCP: O CONFNAK [ACKrcvd] id 3 len 10
1d06h: As8 IPCP: Address 192.168.1.6 (0x0306C0A80106)
1d06h: As8 IPCP: I CONFREQ [ACKrcvd] id 4 len 10

```

(持续)

```

1d06h: As8 IPCP: Address 192.168.1.6 (0x0306C0A80106)
1d06h: As8 IPCP: O CONFACK [ACKrcvd] id 4 len 10
1d06h: As8 IPCP: Address 192.168.1.6 (0x0306C0A80106)
1d06h: As8 IPCP: State is Open ← IPCP completes and is Open
skynet_lab#ping 192.168.1.6

```

```

skynet_lab#ping ← Source PING
Protocol [ip]: ip
Target IP address: 192.168.1.6
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.1.1
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 192.168.1.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 136/166/196 ms
skynet_lab#

```

启动“Big D”之后，大部分的 PPP 问题都会变得很明显。如果没有 **debug**，要发现模拟拨号的错误所在是很困难的。确认所有的 4 个步骤都成功完成之后，在 172.16.1.1 做针对 192.168.1.6 的 **ping** 操作以确认 IP 的连通性。

一些常见的问题包括：

- **PPP 无法启动**（没有 **ppp debug** 信息出现）——这是调制解调器或者其他物理连接的问题。同时还要检查是否使用了 **autoselect PPP**。
- **PPP 认证失败**——这在 **debug** 显示信息中已经很清楚——密码不匹配。
- **没有到达目的网络的路由**——DDR 的作用就是确保到某个目的网络存在一条路由。出现该问题，还可能需要检查地址协商协议或发送到 PPP 客户端的 IP 地址是否正确。

要增强链路的性能，可以在接口 8 上启动 **MPPC** 并且屏蔽掉 **CDP**。**MPPC** 压缩的启动是通过在接口上加入 **compress mppc** 命令来实现的。在接口上加上这条命令之后，能够看到 **MPPC** 成功地通过协商确定下来的情况，如例 4-23 所示。而要屏蔽掉 **CDP**，只需在接口上执行 **no cdp enable** 命令即可。

例 4-23 MPPC 压缩的 debug 命令输出

```

1d07h: As8 CCP: O CONFREQ [Closed] id 2 len 10
1d07h: As8 CCP: MS-PPC supported bits 0x00000001 (0x120600000001)
<<<text omitted>>>
1d07h: As8 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
1d07h: As8 CCP: I CONFREQ [REQsent] id 1 len 15
1d07h: As8 CCP: MS-PPC supported bits 0x00000001 (0x120600000001)
1d07h: As8 CCP: Stacker history 1 check mode EXTENDED (0x1105000104)
1d07h: As8 CCP: O CONFREQ [REQsent] id 1 len 9
1d07h: As8 CCP: Stacker history 1 check mode EXTENDED (0x1105000104)
1d07h: As8 IPCP: I CONFACK [REQsent] id 13 len 10

```

（待续）

```

1d07h: As8 IPCP:      Address 192.168.1.5 (0x020600A80105)
1d07h: As8 CCP: 1 CONFACK [REQsent] id 2 len 10
1d07h: As8 CCP: 1 MS-PPC supported bits 0x00000001 (0x120600000001)
1d07h: As8 CCP: 1 CONREQ [ACKrcvd] id 2 len 10
1d07h: As8 CCP: 1 MS-PPC supported bits 0x00000001 (0x120600000001)
1d07h: As8 CCP: 0 CONFACK [ACKrcvd] id 2 len 10
1d07h: As8 CCP: 0 MS-PPC supported bits 0x00000001 (0x120600000001)
1d07h: As8 CCP: State is Open

```

注释 如果用来测试 PAP 连接的是两台路由器而不是一台路由器和一台工作站，要成功地出现 PAP 握手信号，就需要在异步接口上使用 **ppp pap sent-username** 命令。

最后，例 4-24 给出了完整的 skynet_lab 配置示例。

例 4-24 完整的 skynet_lab 配置示例

```

hostname skynet_lab
!
enable password cisco
!
username JP password 0 trashman
ip subnet-zero
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
!
<<<text omitted>>>
!
interface Async8
 ip address 192.168.1.5 255.255.255.252
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 dialer in-band
 dialer idle-timeout 600
 dialer-group 8
 async mode interactive
 peer default ip address 192.168.1.6
 compress mppc
 no cdp enable
 ppp authentication pap
!
ip classless
!
dialer-list 8 protocol ip permit
!
line con 0
 exec-timeout 0 0
 transport input none
line 1 7
 transport input all
line 8
 autoselect during-login
 autoselect ppp
 login local
 modem InOut
 modem autoconfigure type default
 transport input all

```

（待续）

```
speed 115200
line 9 16
line aux 0
line vty 0 4
  login local
!
end

skynet_lab#
```

4.4 实验 11：同步链路路上的 PPP、CHAP 和 LQM

配置——第 1 部分

4.4.1 实验说明

PPP CHAP 能够为很多类型网络媒介上的对等网会话提供安全的认证方式。前面讨论过，PPP 是用在串行、ISDN、xDSL 以及其他类型的介质上的。现在，PPP 甚至已经用在以局域网为基础的网络中。这主要因为 PPP 提供的是对等认证方式。

4.4.2 实验内容

Skynet 刚刚把它的 Internet 连接升级到了 T1 的水平。完成升级的日子正好是一些著名的 Internet 游戏发行的时间。Skynet 觉得这是对其新完成的连接进行压力测试的理想时间。现在我们的任务就是提供一条安全快捷的访问 Internet 的连接。设计的时候要以下面的要求为准绳：

- 数据链路层协议采用 PPP。ISP 路由器的主机名是 isp_router。采用 CHAP 认证方式，CHAP 密码是 2diablo2。
- 这个时候，这条链路是 Skynet 与 Internet，以及 Skynet 以外的网络进行连接的惟一通道。
- 对链路配置使其不能允许超过 40% 的数据包丢失。

4.4.3 实验目的

- 如图 4-10 配置访问服务器以及网络。
- 使用 PPP 的 CHAP 作为与 ISP 之间同步链路上数据链路层的认证协议。
- 在链路上实施链路质量管理 (LQM)。
- 可选：ISP 有一个 DNS 服务器，IP 地址是 128.200.1.2。配置 skynet_lab 路由器来使用这个 DNS 服务器。也就是说，登录到 skynet_lab 路由器之后，应该能够 ping 通 www.cisco.com。就实验室的条件，这一点可能不是太好进行测试。重要的是怎样正

确配置路由器去转发 DNS 请求数据包。

4.4.4 所需设备

- 两台 Cisco 路由器，通过 V.35 背对背线缆或类似的方式连接在一起。
- 通过集线器或交换机搭建两个局域网段。
- 一台可选的工作站作为 DNS 服务器。

4.4.5 物理设计和实验准备

- 如图 4-10 所示将集线器和串行线缆与路由器相连。
- 将两台以太集线器与两个局域网段相连，如图 4-10 所示。
- 可选：在 ISP 局域网段中连接并配置 DNS 服务器。

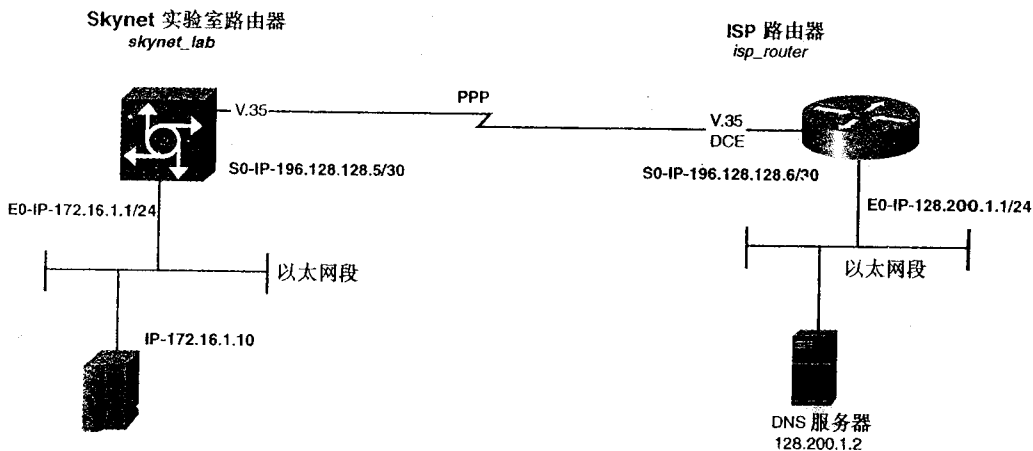


图 4-10 Skynet 访问一个 ISP 的 PPP 串行连接

4.5 实验 11：同步链路上的 PPP、CHAP 和 LQM 配置——第 2 部分

4.5.1 实验步骤

如图 4-10 所示，通过 V.35 线缆或者是带有终端线缆的 CSU/DSU 将两台路由器以背对背方式连在一起。

对于 skynet_lab 的配置，可以沿用上一个实验的大部分配置内容。这里我们要做的实际上是在接口配置 PPP，启动 CHAP 认证，然后运行 LQM。

首先，通过将封装类型改为 PPP 来把接口配置为 PPP 之用。同样，为远程路由器设置用

户名 `isp_router` 和密码 `2diablo2`:

```
skynet_lab (config) #username ips_router password 2diablo2
```

```
skynet_lab (config) #int s0
```

```
skynet_lab (config-if) #encapsulation ppp
```

```
skynet_lab (config-if) #ppp authentication chap
```

通常，上面这些就可以完成串行链路上的 PPP 配置。但是还需要 LQM 和 DNS 的服务。对于 LQM，希望设置接口上的数据丢失不允许超过 40%，为此，可以利用 `ppp quality percentage_of_successful_traffic` 命令进行配置。由于 PPP 的质量是以成功传输数据的百分比来衡量的，因此这里设置的值是 60。

现在到 ISP 的路由器上去，这台路由器的设置与 Skynet 路由器的配置方式基本一致。例 4-25 就是 ISP 路由器所需要的配置步骤。

例 4-25 ISP PPP 链路的配置示例

```
isp_router(config)#username skynet_lab password 2diablo2
isp_router(config)#int s0
isp_router(config-if)#ip address 196.128.128.6 255.255.255.0
isp_router(config-if)#encapsulation ppp
isp_router(config-if)#ppp authentication chap
isp_router(config-if)#clock-rate 2000000
isp_router(config-if)#ppp quality 60
isp_router(config-if)#^Z
```

现在可以通过对两台路由器用 `ping` 命令来测试串行链路的连通性。如果启动了 `debug ppp negotiation` 和 `debug ppp authentication`，还能够看到链路上的 LQM 信息帧在交换，以及 PPP CHAP 认证过程。

最后一步是 ISP 和 Skynet 路由器的配置。这时 Skynet 只有一条通向它以外的网络去的路由。因此可以将所有的数据量都通过这一条路由指向 ISP 的串行接口 IP 地址。所需要的命令如下：

- `ip classless`
- `ip default-network 0.0.0.0`
- `ip route 0.0.0.0 0.0.0.0 196.128.128.6`

命令 `ip classless` 使得路由器将转发所有没有在路由表中找到目的地址网络条目的数据。没有这条命令，一旦路由器没有通往目的地址的路由，那它就不会对任何数据进行转发。命令 `ip default-network` 是把默认路由设为 0.0.0.0，而 `ip route` 则是指向网络 0.0.0.0 的一条静态路由，所有的数据都会发送到这里指定的下一跳地址处。

对 ISP 的路由来说，需要更具体一些。在这个例子中，只需要指定一条通往远程网络的静态路由即可：

```
ip route 172.16.1.0 255.255.255.0 196.128.128.5
```

这些都完成之后，从每台路由器的以太网端口发送一些扩展的 `ping` 命令来测试是否已经建立起了双向连接。如果在 Skynet 上执行 `show ip route` 命令，可以看出默认路由已经设置好了，如例 4-26 所示。

例 4-26 Skynet 和 ISP 路由器上的 show ip route 命令示例—请注意所设置的默认路由

```
skynet_lab#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, * - candidate default
        U - per-user static route, o - ODR
        T - traffic engineered route

Gateway of last resort is 196.128.128.6 to network 0.0.0.0    <-Default route set

C    201.201.201.0/24 is directly connected, Loopback0
    196.128.128.0/24 is variably subnetted, 2 subnets, 2 masks
C    196.128.128.4/30 is directly connected, Serial0
C    196.128.128.6/32 is directly connected, Serial0
    172.16.0.0/24 is subnetted, 1 subnets
C    172.16.1.0 is directly connected, Ethernet0
    192.168.1.0/30 is subnetted, 1 subnets
C    192.168.1.4 is directly connected, Async8
S*  0.0.0.0/0 [1/0] via 196.128.128.6    <-IP next hop of default route
skynet_lab#

isp_router#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, * - candidate default
        U - per-user static route, o - ODR

Gateway of last resort is not set

    128.200.0.0/24 is subnetted, 1 subnets
C    128.200.1.0 is directly connected, Ethernet0
    172.16.0.0/24 is subnetted, 1 subnets
S    172.16.1.0 [1/0] via 196.128.128.5    <-route to Skynet
    196.128.128.0/24 is variably subnetted, 2 subnets, 2 masks
C    196.128.128.5/32 is directly connected, Serial0
C    196.128.128.0/24 is directly connected, Serial0
isp_router#
```

接下来这一步可选，允许向 DNS 服务器转发 DNS 请求。为了进行 DNS 请求的转发，需要用下面这两条命令进行配置：

- **ip name-server** *DNS_server_IP_address*
- **ip domain-lookup**

命令 **ip name-server** 向路由器指定了 DNS 服务器的 IP 地址，而 **ip domain-lookup** 则是对 UDP DNS 数据包进行转发。例 4-27 就是利用 ISP 处的 DNS 服务器所需要的配置情况。这个例子是从一台真正的 ISP 路由器处摘出来的，请注意其中并没有与 Internet 的“真正”连接，因此这里的命令也只是作参考之用，因而也就没有列在例 4-28 中。

例 4-27 使用路由器上 DNS 服务的配置示例

```
skynet_lab(config)#ip name-server 204.221.151.248
skynet_lab(config)#ip domain-lookup
```

（待续）

```

skynet_lab(config)#^Z
skynet_lab#ping www.cisco.com
Translating "www.cisco.com"...domain server (204.221.151.248) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.133.219.25, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/116 ms
skynet_lab#

```

如果 DNS 出现问题，首先检查与 DNS 服务器的连接，然后再确定路由器的 UDP 数据包是否正确地发出来了。

例 4-28 给出了整个的配置过程。

例 4-28 Skynet 和 ISP 路由器的配置示例

```

skynet_lab#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname skynet_lab
!
enable password cisco
!
username JP password 0 trashman
username isp_router 0 2diablo2
ip subnet-zero
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
ip address 196.128.128.5 255.255.255.252
no ip directed-broadcast
encapsulation ppp
no ip mroute-cache
no fair-queue
ppp quality 60
!
interface Serial1
no ip address
no ip directed-broadcast
shutdown
!
interface Async8
ip address 192.168.1.5 255.255.255.252
no ip directed-broadcast
encapsulation ppp
no ip mroute-cache
dialer in-band
dialer idle-timeout 600

```

(待续)

```
dialer-group 8
async mode interactive
peer default ip address 192.168.1.6
compress mppc
no cdp enable
ppp authentication pap
!
ip classless
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 196.128.128.6
!
dialer-list 8 protocol ip permit
!
line con 0
exec-timeout 0 0
transport input none
line 1 7
transport input all
line 8
autoselect during-login
autoselect ppp
login local
modem InOut
modem autoconfigure type default
transport input all
speed 115200
line 9 16
line aux 0
line vty 0 4
login local
!
end

skynet_lab#
```

```
isp_router#show running-config
```

```
Building configuration...
```

```
Current configuration:
```

```
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname isp_router
!
username skynet_lab password 2diablo2
!
interface Ethernet0
ip address 128.200.1.1 255.255.255.0
!
interface Serial0
ip address 196.128.128.6 255.255.255.0
encapsulation ppp
clockrate 2000000
ppp quality 60
!
interface Serial1
no ip address
shutdown
```

（待续）

```
!  
interface BRI0  
  no ip address  
  shutdown  
!  
no ip classless  
ip route 172.16.1.0 255.255.255.0 196.128.128.5  
!  
line con 0  
  exec-timeout 0 0  
line aux 0  
line vty 0 4  
  login  
!  
end  
  
isp_router#
```

4.6 实验 12：同步链路的 PPP 模拟拨号备份——

第 1 部分

4.6.1 实验说明

大多数的节点都通过应用数据链路协议 PPP 的 ISDN 或模拟线路作为其串行链路的备份。DDR 一个常见的问题就是如何保证在一定条件下进行拨号呼叫。而 ISDN 或其他异步线路则可能由于一个配置不当的访问列表而进行拨号呼叫，从而导致线路成本的高昂费用。再加上其他一些网络设计上的限制，使我们不得不考虑以其他方式来对网络进行配置。例如，如果在 OSPF 的 Area 0 或其他常规区域中配置一个拨号接口，由于链路状态信息的大量传输，拨号连接就会一直不断地存在。

4.6.2 实验内容

SuperGreat Food Corp.在其公司总部基于 IBM 3090x 的平台上运行着公司以 IP 为基础的一个库存盘点系统。每个分公司都是通过 64-kbit/s 的 PPP 连接与 SuperGreat Food Corp.总部的系统相连的，分公司处采用的是 RS6000。RS6000 要求与大型机以及 SuperGreat 的 IP 网络之间建立 IP 连接。SuperGreat 的 IP 网络是在自治系统 (AS) 2001 中的。为了实现完全的 IP 互连，在所有的链路上必须运行 EIGRP 协议。做网络设计的时候必须遵守 SuperGreat Food Corp.提出来的这些要求：

- 所有串行链路上采用 PPP 作为数据链路协议。所有的 PPP 链路上都通过 CHAP 进行认证。CHAP 密码是 cub9biggs。
- 路由选择协议采用 EIGRP，所有链路上都要进行路由交换，自治系统 ID 是 2001。
- 设置串行链路并必须保证在有线路失效发生的时候，AUX 端口可以提供拨号备份。

在拨号备份上也应该可以收发路由更新信息。

4.6.3 实验目的

- 如图 4-11 配置 SuperGreat 的网络。
- 在分部和公司总部路由器之间的串行连接上采用 PPP。
- 两个站点之间的模拟拨号备份要配置成只有 PPP 线路失效时才能触发备份的呼叫。
- 路由选择协议采用 EIGRP。不要使用任何形式的静态路由。

4.6.4 所需设备

- 两台 Cisco 路由器，通过 V.35 背对背线缆或类似方式连接在一起。LAN 的类型并不重要。
- 两台模拟调制解调器，Cisco 终端电缆以及 MODEM 接线头。
- 通过集线器或交换机实现的两个 LAN 网段，LAN 的类型对实验的配置并不重要。
- 两台 Windows 95/98/2000 工作站，其 IP 配置成模拟 IBM 大型机和 RS6000 小型机，如图 4-11 所示。

4.6.5 物理设计和实验准备

- 如图 4-11 所示连接路由器、集线器、串行线缆以及调制解调器。
- 把令牌环集线器连接到 sub_branch 路由器，而以太集线器则连到 sub_corp 路由器。
- 连接和配置两个基于 IP 的工作站以做测试之用。

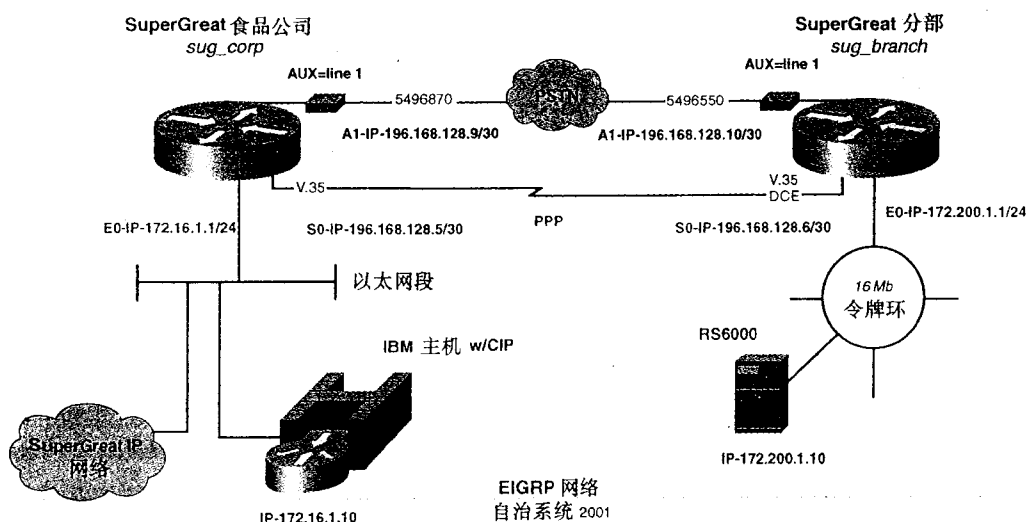


图 4-11 SuperGreat Foods 具有拨号备份的 PPP 网络

4.7 实验 12: 同步连接的 PPP 模拟拨号备份——

第 2 部分

4.7.1 实验步骤

如图 4-11 所示, 用 V.35 线缆或者是带反接线缆的 CSU/DSU 将两台路由器以背对背方式连接起来。把调制解调器和“真正的”模拟线路或者是一个 MODEM 消除器连接起来, 并与路由器的 AUX 端口相连, 并确认使用的是具有 MODEM 接线头的 Cisco 终端电缆进行连接的。调制解调器与路由器 AUX 端口的连接与配置可以参考第 1 章的内容。

这个实验设计要求我们把 EIGRP 作为路由选择协议, 并且只有在 PPP 线路失效的情况下才需要进行拨号呼叫。Cisco 提供了一条 **backup interface** 命令来将接口配置成平时处于待机状态, 只在某些特定情况下才进行拨号呼叫。

命令 **backup interface** 就是指定接口保持在待机状态, 只在下面情况下才进入工作状态:

- 主链路由于数据链路层的问题停止工作。
- 主链路上的负载到达了特定的阈值。
- 主链路上的负载超过了特定的阈值。

命令 **backup interface** 用于主串行接口上。如果有 2503 路由器, 两个串行连接, 一个 ISDN 端口, 而串行连接是通过 ISDN 接口来备份, 串行连接称为主接口, 而 ISDN 则称为备份接口, **backup interface** 命令是在作为主接口的串行链路上使用, 而不是在 ISDN 接口上使用的。

命令 **backup** 是在接口模式下输入的, 其句法结构如下:

```
backup interface interface_name_or_type
```

主链路上状态发生改变之后到备份链路变化之间的延时, 这个延迟时间也是可以设置的, 有两个时间值可以设置:

- 设置主链路停止工作之后备份链路开始工作之前的延迟时间。
- 设置主链路重新开始工作之后备份链路断开连接之前的延迟时间。设置延迟时间的命令是:

```
backup delay { enable-delay | never } { disable-delay | never }
```

- 也可以指定一个负载备份, 即激活备份链路是根据主链路的负载量来决定的, 这可以通过下面这条命令来实现:

```
backup load { enable-threshold | never } { disable-load | never }
```

在配置完 **backup** 命令之后, 备份接口马上就会进入待机状态。例 4-29 是一个处于备份模式下的链路的情况。这种状态下的接口不会对 ping 做出回应。

例 4-29 执行了 backup 命令的链路状态

```
Async1 is standby mode, line protocol is down
Hardware is Async Serial
Internet address is 192.168.128.10/30
```

```
MTU 1500 bytes, BW 38 Kbit, DLY 1000000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set, keepalive not set
DTR is pulsed for 5 seconds on reset
LCP Closed
Closed: IPCP, CDPCP
```

这种情况下，**backup** 命令的使用很方便。在用 **backup** 命令配置备份的时候要注意在网络的一端对拨号呼叫进行限制。可能会遇到多个站点呼入同一个 BRI 端口或模拟端口的情况。如果在链路两端都使用了 **backup** 命令，那就有可能导致链路某一端总是处于等待状态之中。比如在一个帧中继网络中，链路每一端的状态都只依赖于本地帧中继交换机，这种情况下就有可能出现链路的一端是“Line UP, Protocol UP”，**backup** 无法激活备份链路，而同时帧中继网络的那一端却是处在 down 状态中。因此，建议只在链路的远程端使用 **backup** 命令。

考虑到备份的问题，现在首先来重点进行分部路由器的配置，为此，需要做的是：

- 进行路由器的初始设置，包括主机名以及用户名 **sug_corp**，用户相关密码。
- 配置令牌环接口的 IP 地址，并且用 **ping** 命令测试一下 RS6000。把 172.200.1.1/24 作为路由器 To0 端口的 IP 地址。
- 配置 Serial 0 接口的 PPP 封装和 IP 地址，其 IP 地址是 196.168.128.6/30。
- 在路由器上配置 EIGRP，使用的 AS ID 是 2001，而路由器的 IP 网络为 196.168.128.0 和 172.200.0.0。

这些完成之后，我们的配置就开始生效，如例 4-30 所示。

例 4-30 sug_branch 的初始设置

```
hostname sug_branch
!
enable password cisco
!
username sug_corp password 0 cub9biggs      ← Used for CHAP
ip subnet-zero
!
interface Serial0
ip address 192.168.128.6 255.255.255.252
no ip directed-broadcast
encapsulation ppp
no ip mroute-cache
no fair-queue
ppp authentication chap
!
interface Serial1
no ip address
no ip directed-broadcast
shutdown
!
interface TokenRing0
ip address 172.200.1.1 255.255.255.0
no ip directed-broadcast
ring-speed 16
!
router eigrp 2001
network 192.168.128.0
network 172.200.0.0
no auto-summary
```

（待续）


```
!  
no ip classless  
!  
line con 0  
line aux 0  
line vty 0 4  
  login  
!  
end
```

在配置调制解调器和拨号备份之前，还应该对主机站点进行配置。对 sug_corp 路由器的配置工作和 sug_branch 差不多，需要：

- 路由器的初始设置、主机名、用户名 sug_branch 以及密码。
- 配置以太接口的 IP 地址，并对大型机进行 ping 测试。路由器 E0 端口的 IP 地址是 172.16.1.1/24。
- 配置用于 Serial 0 接口的 PPP 封装和 IP 地址，其 IP 地址是 196.168.128.5/30。
- 在路由器上配置 EIGRP，使用的 AS ID 是 2001，而路由器的 IP 网络为 196.168.128.0 和 172.16.0.0。

这一系列的配置如例 4-31 所示。

例 4-31 sug_corp 的完整配置示例

```
hostname sug_corp  
!  
enable password cisco  
!  
username sug_branch password 0 cub9biggs ←Used for CHAP  
!  
interface Ethernet0  
  ip address 172.16.1.1 255.255.255.0  
!  
interface Serial0  
  ip address 192.168.128.5 255.255.255.252  
  encapsulation ppp  
  no fair-queue  
  clockrate 64000  
  ppp authentication chap  
!  
interface Serial1  
  no ip address  
  shutdown  
!  
router eigrp 2001  
  network 192.168.128.0  
  network 172.16.0.0  
  no auto-summary  
!  
no ip classless  
!  
line con 0  
line aux 0  
line vty 0 4  
  login  
!  
end
```

现在 RS6000 和大型机之间就建立起了完整的 IP 互连，可以用 **ping** 命令在二者之间进行测试。同时，还要检查一下网络链路各端上的路由表以确保 172.16.1.0 和 172.200.1.0 网络相互之间已在发送路由信息。在串行接口上还需要建立一个 EIGRP 相邻关系。例 4-32 是路由器 sug_branch 上的路由表，后面是 **show eigrp neighbors** 命令的输出结果。

例 4-32 路由器 sub_branch 上的路由表以及 show eigrp neighbor 命令示例

```
sug_branch#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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

    192.168.128.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.128.8/30 [90/70440192] via 192.168.128.5, 00:09:19, Serial0
C       192.168.128.4/30 is directly connected, Serial0
C       192.168.128.5/32 is directly connected, Serial0
    172.200.0.0/24 is subnetted, 1 subnets
C       172.200.1.0 is directly connected, TokenRing0
    172.16.0.0/24 is subnetted, 1 subnets
D       172.16.1.0 [90/2195456] via 192.168.128.5, 00:09:19, Serial0  *-Corp Subnet
sug_branch#
sug_branch#show ip eigrp neighbors
IP-EIGRP neighbors for process 2001
H   Address                Interface    Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)                Cnt Num
0   192.168.128.5           Se0         12 00:09:38    32     200  0  5
sug_branch#
```

确认了路由的相互交换以及 EIGRP 相邻关系的建立之后，开始这个实验的拨号备份部分的配置。

首先是 sub_branch 路由器，按照异步接口上配置 PPP 的 4 个步骤来进行：

- 第 1 步 对调制解调器以及路由器的异步端口进行配置。记得用 **show line** 命令来确认绝对线路号。
- 第 2 步 在异步接口上对 PPP 进行配置。
- 第 3 步 对异步接口上的 IP 进行配置。
- 第 4 步 对异步接口上的 DDR 进行配置。

两台路由器的异步端口上的配置和前面的实验非常类似，主要的区别在路由选择协议上。在异步端口下，要配置 EIGRP，需要使用 **async default routing** 命令。由于要使用 **backup interface** 命令，拨号列表可以允许 sug_branch 上的所有 IP 数据。在加上 **backup interface** 命令之后，如果配置了 EIGRP，就能够看到链路马上就开始拨号呼入。如果再配置一条 **debug dialer** 命令，就会看到是一条 EIGRP 多播信息激活拨号呼叫，如例 4-33 所示。

例 4-33 debug dialer 命令示例

```

11:25:13: Async1: Dialing cause ip (S=192.168.128.10, d=224.0.0.10)
multicast
11:25:13: Async1: Attempting to dial 5496870
11:25:13: CHAT1: Attempting async line dialer script
11:25:13: CHAT1: Dialing using Modem script: dialsug & System script: none
11:25:13: CHAT1: process started
11:25:13: CHAT1: Asserting DTR
11:25:13: CHAT1: Chat script dialsug started

```

在 sug_corp 路由器上没有必要使用拨号列表, 因为网络这一端不会需要建立和维护链路。

例 4-34 是整个 SuperGreat 网络的配置示例。

例 4-34 SuperGreat 网络备份的完整配置

```

hostname sug_corp
!
enable password cisco
!
username sug_branch password 0 cub9biggs
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0
 ip address 192.168.128.5 255.255.255.252
 encapsulation ppp
 no fair-queue
 clockrate 64000
 ppp authentication chap
!
interface Async1
 ip address 192.168.128.9 255.255.255.252
 encapsulation ppp
 async default routing
 async mode interactive
 dialer in-band
 dialer idle-timeout 300
 dialer map ip 192.168.128.10 name sug_branch broadcast 5496550
 ppp authentication chap
!
router eigrp 2001
 network 192.168.128.0
 network 172.16.0.0
 no auto-summary
!
ip classless
!
line con 0
line aux 0
 autoselect ppp
 modem InOut
 modem autoconfigure discovery
 transport input all
 rxspeed 38400
 txspeed 38400
line vty 0 4
 login
!

```

(待续)

```

end

hostname sug_branch
!
enable password cisco
!
username sug_corp password 0 cub9biggs
ip subnet-zero
chat-script dialaug "" "ATZ&F" OK "ATDT5496870" TIMEOUT 60 CONNECT
!
interface Serial0
ip address 192.168.128.6 255.255.255.252
no ip directed-broadcast
encapsulation ppp
no ip mroute-cache
backup interface Async1
no fair-queue
ppp authentication chap
!
interface TokenRing0
ip address 172.200.1.1 255.255.255.0
no ip directed-broadcast
ring-speed 16
!
interface Async1
ip address 192.168.128.10 255.255.255.252
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer map ip 192.168.128.9 name sug_corp broadcast 5496870
dialer-group 1
async default routing
async mode interactive
ppp authentication chap
!
router eigrp 2001
network 172.200.0.0
network 192.168.128.0
no auto-summary
!
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
transport input none
line aux 0
autoselect ppp
script dialer dialaug
modem InOut
modem autoconfigure discovery
transport input all
speed 38400
line vty 0 4
login
!
end

```

要对 SuperGreat 的网络备份功能进行测试，可以在公司总部处拔掉或者是关掉串行接口。在连接丢失的那一刻，会发现分公司的路由器开始呼叫远程路由器。路由表也会在异步接口上收敛，例 4-35 突出显示这一过程的效果。

例 4-35 sug_branch 路由器上 debug dialer 命令的输出以及路由表的收敛

```
sug_branch#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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

    192.168.128.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.128.8/30 [90/70440192] via 192.168.128.5, 00:04:19, Serial0
C       192.168.128.4/30 is directly connected, Serial0
C       192.168.128.5/32 is directly connected, Serial0
       172.200.0.0/24 is subnetted, 1 subnets
C       172.200.1.0 is directly connected, TokenRing0
       172.16.0.0/24 is subnetted, 1 subnets
D       172.16.1.0 [90/2195456] via 192.168.128.5, 00:04:43, Serial0    ← Corp
Ethernet
sug_branch#
sug_branch#
11:34:49: %LINK-3-UPDOWN: Interface Serial0, changed state to down    ← Serial
Drops
11:34:50: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state
to down
11:35:06: Async1: re-enable timeout
11:35:06: Async1: Dialing cause ip (s=192.168.128.10, d=224.0.0.10)    ← EIGRP forces
dial
11:35:06: Async1: Attempting to dial 5496870
11:35:06: CHAT1: Attempting async line dialer script
11:35:06: CHAT1: Dialing using Modem script: dialaug & System script: none
11:35:06: CHAT1: process started
11:35:06: CHAT1: Asserting DTR
11:35:06: CHAT1: Chat script dialaug started
11:35:25: CHAT1: Chat script dialaug finished, status = Success
11:35:25: CHAT1: Chat script dialaug finished, status = Success
11:35:32: dialer Protocol up for As1
11:35:32: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state
to up    ← Async UP
sug_branch#
sug_branch#
sug_branch#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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

    192.168.128.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.128.8/30 is directly connected, Async1
C       192.168.128.9/32 is directly connected, Async1
       172.200.0.0/24 is subnetted, 1 subnets
C       172.200.1.0 is directly connected, TokenRing0
       172.16.0.0/24 is subnetted, 1 subnets
```

(待续)

```
D 172.16.1.0/24 (192/89863792) via 192.168.128.0/24, Async: Comp Ethernet
Reported over the Async int
sug_branch#
sug_branch#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:
[111] The BANGS are our friends
Success rate is 100 percent (5/5), round-trip min/avg/max = 200/238/324 ms
sug_branch#
```

第 5 章

WAN 协议与技术： 帧中继

过去的 5 到 7 年里，帧中继几乎成为“WAN 之王”。许多私有网络都将原有点对点串行链路过渡到了基于帧中继的网络。尽管目前仍很流行，但是帧中继已经开始衰落。帧中继电路最终会被更为低廉、更为高速的电路，如 DSL 或者电缆调制解调技术代替。如果家庭用户能够以 T3 的速度访问 Internet 服务商（ISP），那么 DSL 或其他使用铜轴电缆的技术代替低带宽的帧中继服务只是时间问题。当然，新协议完全替代目前帧中继技术的王者地位还需要好几年的时间。

本章讲述帧中继的一些术语，大致分析帧中继的原理和 LMI 工作方式。另外，本章还讲解了基本和较复杂的帧中继配置问题，包括帧中继的流量整形问题。

5.1 帧中继的相关术语

在讲解帧中继之前，有必要讨论一些常用的术语。以下部分术语已经在第一章中提到：

- 永久虚电路（PVC）——用于帧传输的逻辑端到端电路。PVC 的终点是用 DLCI 来寻址的。
- 数据链路连接标识符（DLCI）——是一个 16 到 1007 的逻辑数字，用来标识用户端设备（CPE）和帧中继交换机之间的 PVC。DLCI 只在本地有效，这表明只有本地设备和帧中继交换机才关心 DLCI 值。
- 本地管理接口（LMI）——路由器和帧中继交换机之间使用的信令标准。交换机使用 LMI 来确定已定

义的 DLCI 及其状态。LMI 还支持 10 秒间隔的 keepalive 机制，这能确保 PVC 处于工作状态且正在交换数据。Cisco 路由器支持 3 种类型的 LMI: CISCO、ANSI 和 Q933A。如果没有设置 LMI 类型，路由器会通过自动协商机制在 3 种类型中选择使用：

- **CISCO**——由 3 大巨头 Cisco、Digital 和 Northern Telecom 定义的 LMI 类型。是自动协商机制失败后的 LMI 默认类型。其状态信息通过 DLCI 0 传送。
- **ANSI**——ANSI 标准 T1.617（通常称为 Annex D）定义的 LMI 类型。这是所有的帧中继网络中最常用的 LMI 类型。其状态信息通过 DLCI 1023 传送。
- **Q933a**——定义为 ITU-T Q.933 的 LMI 类型，或简称为 Annex A。其状态信息通过 DLCI 0 传送。
- **网络到网络接口 (NNI)**——两台交换机间通信的标准。帧中继和 ATM 都使用 NNI。ATM 中称为网络节点接口 (Network Node Interface)。
- **本地访问速率**——与帧中继服务提供者相连链路的时钟速率或称接口速率。通常，该电路为 56 kbit/s、64 kbit/s 或 T1 速率，但也能工作在 T3 或 **高速串口 (High Speed Serial Interface, HSSI)** 下。

图 5-1 为常用的帧中继网络示意图，上述术语都在图中标出。

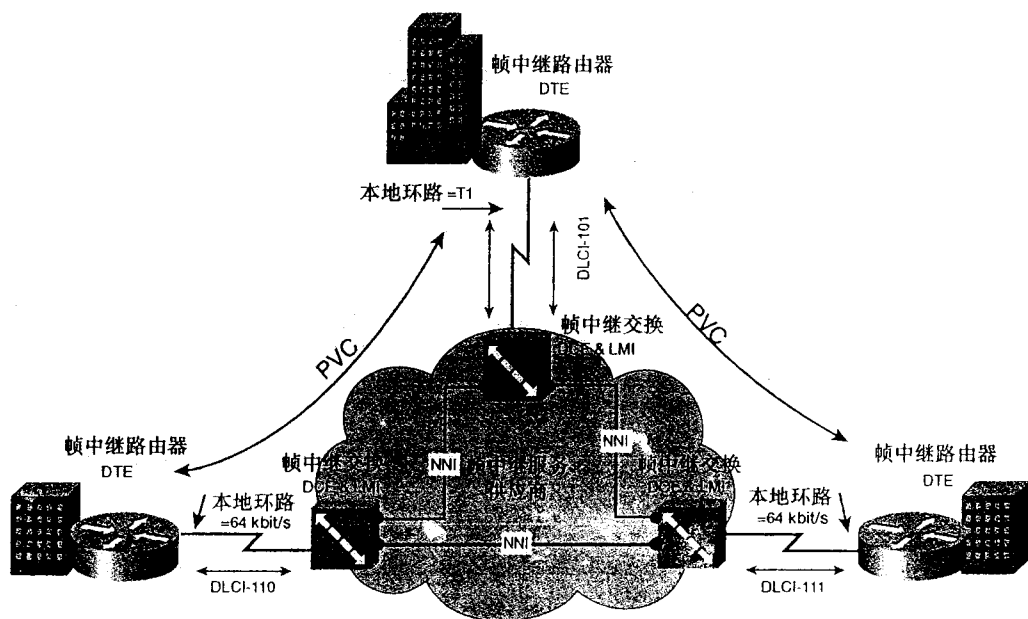


图 5-1 部分连接的帧中继网络

下面这些术语称为 **数据速率度量值**。帧中继服务提供者使用这些 **数据速率度量值** 来规定服务级别。下面的这些术语还用于配置帧中继的流量整形：

- **承诺突发量 (Bc)**——以 CIR 为基础的允许接收和发送的比特数。
- **承诺信息速率 (CIR)**——PVC 允许的最大数据速率。超过时数据就会被设置为可丢弃位 (DE)，该位表示，如果已经达到帧中继交换机的最大承载能力，数据帧可被丢弃。单位 bit/s。

- 过量突发量 (Be) —— 达到了承诺突发值之后可发送的超出比特数。
- 最大速率 (MaxR) —— 该值的单位是 bit/s，其计算公式如下：

$$\text{MaxR} = \text{CIR} \times ((\text{Bc} + \text{Be}) / \text{Bc})$$

图 5-2 显示了各速率间的关系。

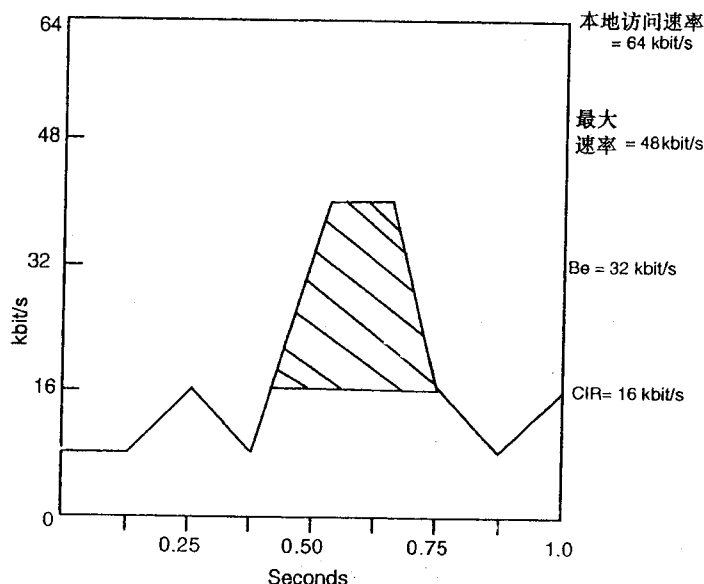


图 5-2 帧中继数据速率度量

5.2 帧中继技术概览

需要是发明之母。这句话也适用于网络协议的发展。帧中继刚出现时，X.25、SDLC 以及当时其他一些 WAN 协议都有很多的缺陷。帧中继具有很多有价值的功能，有助于设计更完善的网络。帧中继的特点如下：

- 帧中继通过统计多路复用技术使多个逻辑电路的功能在一个物理电路上实现。
- 帧中继不需要链路具备专门的端到端电路，有助于降低线路成本。
- 统计多路复用技术通过减少对路由器串口和 CSU/DSU 的使用需求，增加了网络的可扩展性。
- 帧中继的可扩展性网络设计：
 - 始终坚持使用三层网络模型——核心层、分布层和访问层。
 - 允许全连接，部分连接和混合连接的方案。
 - 增加了协议广播和性能控制。

帧中继是一个 CCITT 和 ANSI 的标准，是 X.25 协议的换代协议。通常把 X.25 协议称为设计过度协议 (*overengineered protocol*)，因为它在数据链路上和网络层中进行了大量的检错和纠错工作。这是因为 X.25 要运行在很多低质量的链路上。帧中继使用的是面向连接的数据流，依靠上层网络协议完成错误检测和纠正的工作。下列标准描述了帧中继：

- ANSI T1.606: “帧中继传输服务的框架结构和服务描述” (1991)。
- ANSI T1.617: “帧中继传输服务的信令定义” (1991)。
- ANSI T1.618: “帧中继传输服务的核心部分” (1991)。
- ITU Q.933 and Q.922: 用户控制。
- RFC 1490: 帧中继封装定义。

可以从 www.frforum.com 找到更多帧中继的应用实例和其他相关资料。

5.2.1 帧中继 LMI 的操作

LMI 对于帧中继非常重要。当一个帧中继链路在 Cisco DTE 设备上激活并开始工作时，会连续地向路由器传送 3 个 LMI 消息，这 3 个消息的顺序是 ANSI、ITU 和 Cisco。路由器在 DLCL1023 上监听 CISCO 的消息，在 DLCL 0 上监听 ANSI LMI 和 ITU 消息。帧中继会对其所配置的 LMI 类型做出响应，然后路由器设定接口的 LMI 类型以与所接收到的 LMI 类型相匹配。如果接收到的是多种类型的 LMI，路由器设定其 LMI 类型时会使用最后接收到的 LMI 类型。Cisco 称这种方式为 *LMI 自动识别 (LMI autosense)*。然后，路由器每 10 秒发送一次 LMI 的状态信息。该状态信息称为用于保持 LMI 连接信息 (*LMI keepalive*)。路由器工作在如图 5-3 所示的方式中。

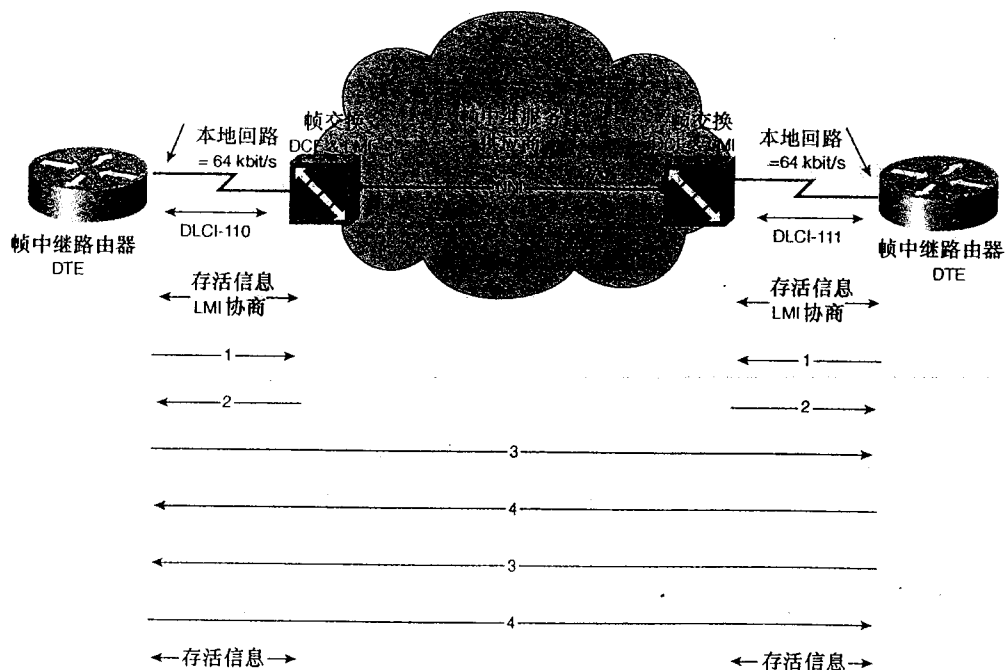


图 5-3 帧中继 LMI 的工作模式

1 在每经过第六个 LMI 状态请求时，DTE 设备会发出一个完整的状态请求。该请求也是一种用于保持连接的信号，帧中继交换机会以链路定义的所有 DLCI 列表作为响应。例 5-1 显示了初始交换时发生的情况。

例 5-1 debug frame lmi 和 debug frame packet 获得的初始 LMI 设置和数据交换的结果

```

00:19:19: Serial0(out): StEnq, myseq 1, yourseen 0, DTE up
00:19:19: datagramstart = 0x400002DC, datagramsize = 14
00:19:19: FR encap = 0x00010308
00:19:19: 00 75 95 01 01 00 03 02 01 00
00:19:19:
00:19:19: Serial0(out): StEnq, myseq 1, yourseen 0, DTE up
00:19:19: datagramstart = 0x4000053C, datagramsize = 13
00:19:19: FR encap = 0x00010308
00:19:19: 00 75 51 01 00 53 02 01 00
00:19:19:
00:19:19: Serial0(out): StEnq, myseq 1, yourseen 0, DTE up
00:19:19: datagramstart = 0x400002DC, datagramsize = 13
00:19:19: FR encap = 0xFCF10309
00:19:19: 00 75 01 01 00 03 02 01 00
00:19:19:
00:19:19: Serial0(in): Status, myseq 1
00:19:19: RT IE 1, length 1, type 0
00:19:19: KA IE 3, length 2, yourseq 1, myseq 1
00:19:19: PVC IE 0x7, length 0x6, dlci 110, status 0x0, bw 0
00:19:29: Serial0(out): StEnq, myseq 2, yourseen 1, DTE up
00:19:29: datagramstart = 0x400002DC, datagramsize = 13
00:19:29: FR encap = 0xFCF10309
00:19:29: 00 75 01 01 01 03 02 02 01
00:19:29:
00:19:29: Serial0(in): Status, myseq 2
00:19:29: RT IE 1, length 1, type 0
00:19:29: KA IE 3, length 2, yourseq 2, myseq 2
00:19:29: PVC IE 0x7, length 0x6, dlci 110, status 0x0, bw 0

```

2 帧中继交换机接收状态请求信号后，发送完整的状态响应消息，该消息包含该接口上所有处于工作状态的 DLCI 列表。

3 对于每个工作中的 DLCI，路由器都会根据接口配置的第 3 层网络协议的情况发送一个逆向 ARP 请求。例如，接口上配置了 IP 和 IPX，路由器就会发送两个逆向 ARP 请求，请求含有相应网络层地址的路由器做出应答。如果不支持逆向 ARP 请求，数据传输就要通过 **frame-relay map** 命令来进行（在“帧中继的配置”一节中详细讲述）。

4 路由器会根据收到的逆向 ARP 信息里所包含的每条 DLCI 在其帧中继映射表中建立一个映射项。这个映射表包括本地 DLCI 和发出请求的远端路由器的网络层地址信息，还有 PVC 的状态信息，该状态信息可以用 **show frame-relay pvc** 命令显示，各种状态如下：

- **ACTIVE**——表明 PVC 处于工作状态，信息可以进行交换。
- **INACTIVE**——表明帧中继的本地连接正常，但远端路由器到帧中继交换机的连接没有工作。
- **DELETED**——表明帧中继没有收到 LMI 或者没有建立物理层连接。

5 路由器继续每 10 秒交换一次 keepalive 数据。每 60 秒或者说每到第 6 次交换时，发送一个完整的 LMI 状态请求，如此循环。如果连续 3 次完整状态请求没有收到 LMI 信号，就表明链路断开了。

5.3 帧中继的配置

在 Cisco 路由器上只需两个步骤的工作就可以完成帧中继配置。但这里我们列出了一个 4 步骤的过程，这些步骤的部分内容实际上不需要进行配置。无论怎样，读者要了解配置一台完整的帧中继设备所需要的全部过程和步骤。其中一些命令是默认设置，无需输入。在这些基本步骤的基础上也还可以加入一些额外的配置。但就使帧中继链路在路由器上开始工作而言，下面的步骤已经足够：

第 1 步 在接口或子接口上进行帧中继封装

可以通过以下命令实现：

```
router (config-if) #encapsulation frame-relay [cisco | ietf]
```

cisco 是默认的封装类型，在 Cisco 设备或符合 RFC 1490 的设备相连使用该类型。在第三方设备相连时使用 **ietf**。

第 2 步 设置 LMI 类型

所有使用 Cisco IOS11.2 或更高版本 IOS 的 Cisco 路由器都支持 LMI 自动识别，不需要对 LMI 做额外的设置。可以用下面的接口命令手工配置 LMI：

```
Router (config-if) #frame-relay lmi-type [ansi | cisco | q933i]
```

可以参考“帧中继的相关术语”一节中关于不同 LMI 类型的讲述。Cisco 是默认的 LMI 类型。

第 3 步 配置静态或动态地址映射

下面需要确定帧中继接口使用何种类型的地址映射。根据帧中继接口的配置以及远端设备是否支持帧中继逆向 ARP，可以选用 **frame-relay map** 命令或是 **frame-relay interface-dlci** 命令，甚至本手册任何命令来配置地址映射的类型。子接口是物理接口的逻辑划分。如前所述，动态地址映射使用帧中继逆向 ARP。由于物理接口分成了多个子接口，必须做一些配置工作将子接口与 DLCI 映射。帧中继网络中有两种类型的子接口：点对点 (point-to-point) 和点对多点 (multipoint)。如果是点对点子接口，使用如下命令

```
Router (config-if) #frame-relay interface-dlci dlci_number
```

创建点对多点子接口需要使用静态寻址方式，这并不是帧中继的要求，主要是考虑路由方面的原因。逆向 ARP 仍然在工作，但没有静态寻址，路由选择协议就不知道将数据包转发到适当的下一跳地址。下面这条命令可以用来设置点对多点子接口：

```
Router (config-if) #frame-relay map protocol [ip | dec | appletalk | xns | ipx |  
vines | clns | bridge | llc2 | dlsiw] next_hop_address dlci [broadcast]  
[ietf | cisco]
```

命令 **frame-relay map** 在本地 DLCI 和下一跳地址之间建立静态的地址映射。关键字 **broadcast** 用来转发特定的网络协议广播，例如 OSPF。建议该关键字无论何时都应采用。关键字 **ietf** 和 **cisco** 用于在 PVC 上不同的帧封装类型。**frame-relay map** 命令也可以用于帧中继网络中分担数据。例如，IPX 数据可以映射到一个 DLCI，而 IP 数据则可映射到另一个 DLCI 上。根据 RFC 1490 定义，该命令也可以用于传输链路协议（如生成树帧和数据链接交换帧）。

等)中。这条命令的应用很多,关于 **frame-relay map** 所有的用法和选项,读者可以参考《*Cisco IOS Software Configuration Guide*》一书。

表 5-1 列出了不同接口逆向 ARP 地址映射的推荐用法。

表 5-1 推荐的逆向 ARP 和对应的地址映射

标准接口	点对多点子接口	点对点接口	连接到不支持逆向 ARP 的设备
给每个协议加入网络层地址	给每个协议加入网络层地址。使用 frame-relay map 命令	给每个协议加入网络层地址。使用 frame-relay interface-dlci 命令	给每个协议加入网络层地址。使用 frame-relay map 命令
静态或动态寻址	静态寻址	动态寻址	静态寻址

第 4 步 相关协议问题

在帧中继上配置路由选择协议时有一些需要注意的问题。比如,只有网络类型改变或者使用了 **neighbor** 命令声明的情况下,OSPF 才能够正常工作。所有使用距离矢量协议或是 EIGRP 的多点网络都会遇到水平分割问题。表 5-2 列出了帧中继网络中的常见问题。

表 5-2 帧中继网络常见的问题

协议	点对多点子接口	点对点接口
OSPF	必须使用 neighbor 声明,或者在接口上使用 ip ospf network type 命令。使用优先级 1 设置 DR 路由器,该路由器应有到所有邻居的 PVC	必须使用 neighbor 声明,或者在接口上使用 ip ospf network type broadcast 命令
EIGRP	关闭 IP 或 IPX 水平分割功能。加入 bandwidth 命令	加入 bandwidth 命令
RIP	关闭 IP 或 IPX 水平分割功能	无
IGRP	关闭 IP 或 IPX 水平分割功能。加入 bandwidth 命令	加入 bandwidth 命令
BGP	无	无
桥接	设置到路由器的根桥,该根桥具有到所有子桥的 PVC	设置到路由器的根桥,该根桥具有到所有子桥的 PVC

注释 水平分割 (*Split horizon*) 是指某条路由信息不会通过接收到该信息的同一接口或子接口再发送出去。这一问题在多点网络中很常见。路由更新信息从某个子接口接收进来后,还必须从同一子接口发送到多点网络中其他路由器上去。默认情况下,都存在水平分割问题,它会妨碍 EIGRP、IGRP 和 RIP 等协议的路由更新信息在多点网络中的正确发送。在 RIP 或 IGRP 网络中可以使用 **no ip split-horizon** 命令避免这一问题的发生,而在 EIGRP 网络中则可以用 **no ip split-horizon eigrp autonomous_system** 命令。在 IPX 和 AppleTalk 网络中也有类似的命令。如果配置了两个点对点子接口,路由更新信息是从一个子接口接收进来而从另一个发送出去。这主要是因为每个子接口都属于不同的网络。因此,在点对点子接口配置中,没有必要去关闭水平分割功能。

5.3.1 实例：配置混合型帧中继网络

下面的例子提供了应用不同类型接口来配置一个完整的帧中继网络的过程。图 5-4 是该混合型帧中继网络的示意图。

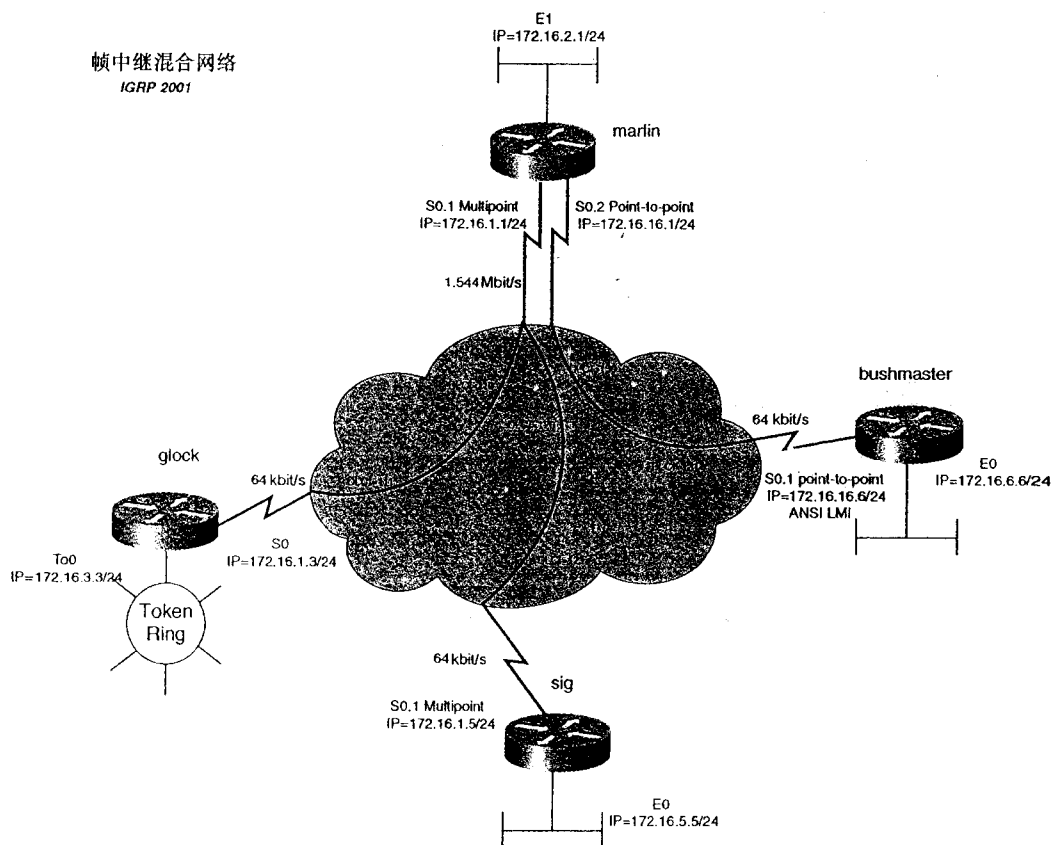


图 5-4 帧中继混合型网络

这个例子要配置路由器 marlin、glock 和 sig 之间的帧中继点到多点网络。在路由器 marlin 和 bushmaster 之间还要配置一个帧中继点对点网络。路由选择协议是 IGRP。

首先，从 marlin 路由器开始。按照帧中继配置过程的四个步骤，先在串行接口上设置帧中继的数据封装形式。然后，定义两种类型的子接口。子网 172.16.1.0/24 需要一个点到多点设置以便将路由器 glock 和 sig 连接在一起。要将子网 172.16.16.0/24 和路由器 bushmaster 连接，可以用一个点对点或者点到多点设置。

该例中使用的是一个点对点网络。路由器配置如例 5-2 所示。

例 5-2 数据封装的设置和子接口的定义

```
marlin#conf t
Enter configuration commands, one per line. End with CNTL/Z.
marlin(config)#int s0
marlin(config-if)#encapsulation frame-relay
marlin(config-if)#int s0.1 multipoint
marlin(config-subif)#ip address 172.16.1.1 255.255.255.0
marlin(config-subif)#exit
marlin(config)#int s0.2 point-to-point
marlin(config-subif)#ip address 172.16.16.1 255.255.255.0
marlin(config-subif)#^Z
```

定义路由器 glock 上的点对多点子接口和路由器 bushmaster 上的点对点子接口时的步骤与上面例子相似。路由器 glock 上不需要使用任何子接口，可以将它看成是个连接多点的路由器。要做的只是在路由器的 s0 接口上定义帧中继数据封装形式。

下一步是配置 LMI。如上所述，帧中继的自动识别功能能够自动识别和配置 LMI，无需另外进行配置。作为练习之用，读者可以在路由器 bushmaster 上将 LMI 手动配制成 ANSI 类型。这可以在其 s0 接口下使用 **frame-relay lmi-type ansi** 命令来完成。

第三步是配置静态或动态寻址。在路由器 marlin 上的多点接口 s0.1 接口上使用静态寻址方式。S0.2 接口是一个点对点接口，因此应该使用动态寻址方式。子网 172.16.1.0/24 上的每一个远程路由器都需要对应一条 **frame-relay map** 映射命令。例 5-3 是静态地址映射的示意图。

例 5-3 配置静态地址映射

```
marlin(config)#int s0.1 multipoint
marlin(config-subif)# frame-relay map ip 172.16.1.3 110 broadcast
marlin(config-subif)# frame-relay map ip 172.16.1.5 120 broadcast
marlin(config-subif)#exit
```

例 5-4 是路由器 marlin 所需的动态寻址配置。

例 5-4 配置动态寻址方式

```
marlin(config)#int s0.2 point-to-point
marlin(config-subif)#frame-relay interface-dlci 130
marlin(config-fr-dlci)#^Z
marlin#
```

路由器 glock 的串口位于多点网络中，因此使用静态寻址方式。可以用 **frame-relay map** 命令来设置。用 **frame-relay map** 命令设置 IP 地址 172.16.1.1 指向 DLCI 111，另一条命令设置位于同样 DLCI 111 上的 IP 地址 172.16.1.5。路由器 glock 的串口配置如例 5-5 所示。

例 5-5 路由器 glock 的串口配置

```
Interface serial0
ip address 172.16.1.3 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
no fair-queue
frame-relay map ip 172.16.1.5 111 broadcast
frame-relay map ip 172.16.1.1 111 broadcast
```

路由器 sig 的 s0 接口为多点子接口，因此，该路由器需要两条 **frame-relay map** 命令，一条对应于路由器 glock，一条对应于路由器 marlin。路由器 sig 的串口配置如例 5-6 所示。

例 5-6 路由器 sig 的串口配置

```
interface serial0.1 multipoint
ip address 172.16.1.5 255.255.255.0
no ip directed-broadcast
no ip mroute-cache
frame-relay map ip 172.16.1.3 121 broadcast
frame-relay map ip 172.16.1.1 121 broadcast
!
```

回到路由器 marlin 完成第 3 步点对点链路设置。子接口 s0.2 是对应于路由器 bushmaster 的点对点接口，因此，在该接口上可以使用动态寻址方式，可通过在接口 s0.2 下用 **frame-relay interface-dlci** *dlci_number* 命令来完成，如例 5-7 所示：

例 5-7 路由器 marlin 接口 s0.2 配置

```
interface serial0.2 point-to-point
ip address 172.16.16.1 255.255.255.0
frame-relay interface-dlci 130
!
```

重复以上步骤可以完成对路由器 bushmaster 点对点子接口的配置，但这次指向 DLCI 131。

现在进入配置过程的第 4 步：解决与协议相关的问题。如前所述，在本例中采用 IGRP 协议的多点网络上会出现水平分割的问题。由于水平分割在默认情况下打开，因此路由器 marlin 不会通过接口 s0.1 向路由器 glock 转发以太网 172.16.5.0/24 的路由更新信息，同时也不会通过接口 s0.1 向路由器 sig 转发令牌环网络 172.16.3.0/24 的路由更新信息。要解决这一问题，应在路由器 marlin 的接口 s0.1 上使用 **no ip split-horizon** 命令。现在就建立了整个帧中继网络完整的 IP 连接。要想进一步调整网络，可以在每个串行接口上使用 **bandwidth** 命令进行配置以使路由选择更加准确。例 5-8 列出了所有路由器配置中的相关内容。

例 5-8 图 5-4 中路由器的相关配置列表

```
hostname marlin
!

interface Ethernet1
 ip address 172.16.2.1 255.255.255.0
 media-type 10BaseT
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
 bandwidth 1544
 no fair-queue
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip split-horizon
 frame-relay map ip 172.16.1.3 110 broadcast
 frame-relay map ip 172.16.1.5 120 broadcast
!
interface Serial0.2 point-to-point
 ip address 172.16.16.1 255.255.255.0
 frame-relay interface-dlci 130
!
router igrp 2001
 network 172.16.0.0
!

hostname glock
!
<<<text omitted>>>
!
interface Serial0
 bandwidth 64
 ip address 172.16.1.3 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 no fair-queue
 frame-relay map ip 172.16.1.5 111 broadcast
 frame-relay map ip 172.16.1.1 111 broadcast
!
interface TokenRing0
 ip address 172.16.3.3 255.255.255.0
 no ip directed-broadcast
 ring-speed 16
!
router igrp 2001
 network 172.16.0.0
!

hostname sig
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.5.5 255.255.255.0
```

(待续)

```

no ip directed-broadcast
!
interface Serial0
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
no fair-queue
!
interface Serial0.1 multipoint
bandwidth 64
ip address 172.16.1.5 255.255.255.0
no ip directed-broadcast
no ip mroute-cache
frame-relay map ip 172.16.1.3 121 broadcast
frame-relay map ip 172.16.1.1 121 broadcast
!
router igrp 2001
network 172.16.0.0
!

hostname bushmaster
!

interface Ethernet0
ip address 172.16.6.6 255.255.255.0
!
interface Serial0
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
!
interface Serial0.1 point-to-point
ip address 172.16.16.6 255.255.255.0
bandwidth 64
frame-relay interface-dlci 131
!
router igrp 2001
network 172.16.0.0

```

现在可以使用标准的 **ping** 命令和 **trace** 命令来测试帧中继网络的功能。然而，有时候读者需要得到帧中继网络更多的工作状态信息。帧中继的“**Big show**”和“**Big D**”命令能够提供很多有用信息，下面讨论这方面内容。

5.4 帧中继的“Big show”和“Big D”命令

帧中继的 **show** 命令和 **debug** 命令非常有用，有助于迅速发现问题的根源而又不会带来无用的多余信息。帧中继的 **show** 命令和 **debug** 命令包括：

```

show frame-relay pvc [ dlci | interface]
show frame-relay lmi
show frame-relay map
debug frame-relay lmi

```

注释 在《IOS WAN Configuration Guide》一书中可以找到非常完整的帧中继 **show** 命令

5.4.1 show frame-relay pvc 命令

命令 **show frame-relay pvc** 可以显示路由器上所有的 PVC，如果加上可选的关键字，则可以显示某个特定的 DLCI 或接口上的信息。该命令显示的大部分信息可读性都很强，它以数据包和字节为单位显示数据传输速率，也能显示前向显式拥塞通知/后向显式拥塞通知和数据包的 DE 信息。“帧中继流量整形的配置”一节会详细讨论 FECN/BECN 的问题，并且还有一些 **show frame-relay pvc** 命令的显示信息介绍。

显示信息中的一个重要的字段就是 PVC 状态。如上所述，PVC 状态可以是下列状态之一：

- **ACTIVE**——表明 PVC 处于工作状态，信息可以进行交换。
- **INACTIVE**——表明到帧中继的本地连接工作正常，但是远端路由器到帧中继的连接不正常。
- **DELETED**——表明帧中继没有收到 LMI 信息或者物理层连接没有建立起来。

例 5-9 列出了在上面那个例子中路由器 marlin 上使用 **show frame pvc** 命令的显示信息。链路上数据正常传输，PVC 的状态是 **ACTIVE**。

例 5-9 show frame pvc 命令在路由器 marlin 上的执行结果

```
marlin#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 110, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 500          output pkts 250          in bytes 62900
  out bytes 29762         dropped pkts 2           in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
  in DE pkts 0           out DE pkts 0
  out bcast pkts 250     out bcast bytes 29762
  pvc create time 05:31:58, last time pvc status changed 05:29:46

DLCI = 120, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 498          output pkts 249          in bytes 27132
  out bytes 29670         dropped pkts 0           in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
  in DE pkts 0           out DE pkts 0
  out bcast pkts 249     out bcast bytes 29670
  pvc create time 05:31:59, last time pvc status changed 05:29:47

DLCI = 130, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

  input pkts 585          output pkts 590          in bytes 107506
  out bytes 118208        dropped pkts 0           in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
  in DE pkts 0           out DE pkts 0
  out bcast pkts 590     out bcast bytes 118208
  pvc create time 05:32:00, last time pvc status changed 05:31:07

marlin#
```

字段“PVC Create Time”和“Last Time PVC Status Changed”也很重要。在接口开始工作之后，如果 PVC 也处于工作状态，这两个时间量应该很接近。例 5-10 列出了路由器 marlin 存在 PVC 问题时的命令执行结果。现在能否通过执行该命令来找出问题的原因？

例 5-10 在路由器 marlin 上存在 PVC 问题时的 show frame pvc 命令执行结果

```
marlin#show frame pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 110, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE = Serial0.1

input pkts 508          output pkts 255          in bytes 63860
out bytes 30362         dropped pkts 2           in FECN pkts 0
in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
in DE pkts 0           out DE pkts 0
out bcast pkts 255     out bcast bytes 30362
pvc create time 05:38:00, last time pvc status changed 00:00:19

DLCI = 120, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

input pkts 508          output pkts 254          in bytes 27632
out bytes 30270         dropped pkts 0           in FECN pkts 0
in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
in DE pkts 0           out DE pkts 0
out bcast pkts 254     out bcast bytes 30270
pvc create time 05:38:01, last time pvc status changed 05:35:49

DLCI = 130, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

input pkts 595          output pkts 601          in bytes 109422
out bytes 120372        dropped pkts 0           in FECN pkts 0
in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
in DE pkts 0           out DE pkts 0
out bcast pkts 601     out bcast bytes 120372
pvc create time 05:38:01, last time pvc status changed 05:37:09
marlin#
```

请注意，PVC 110 的状态已经改变，当前状态为 INACTIVE。接口 S0.1 上其他 PVC，DLCI 120 都处在正常状态，而 DLCI 110 的状态却是 inactive，这表明远程终端出了问题。这个例子里，路由器 glock 的接口关闭，这证明即使是在实验设置过程中也会存在“异常线路”。

5.4.2 show frame-relay lmi 命令

如果 LMI 工作不正常，帧中继就一定会出问题。使用 show frame-relay lmi 命令时，看看“Num Status Enq. Sent xx”是不是随着字段“Num Status msgs Rcvd xx”的增长而增长。如果只发送 LMI 而没有收到 LMI，问题可能就是帧交换出错或 LMI 匹配错误。要确保接口的 LMI 和交换机设置一致。例 5-11 是在路由器 glock 上演示 LMI 匹配错误的例子。仔细观察 LMI 如何发送，注意超时的增长。

例 5-11 用 show frame lmi 命令显示路由器上 LMI 不匹配的问题

```
glock#show frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Sent 82          Num Status msgs Rcvd 18
Num Update Status Rcvd 0          Num Status Timeouts 63

glock#show frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Sent 117          Num Status msgs Rcvd 18
Num Update Status Rcvd 0          Num Status Timeouts 98
glock#
```

将 LMI 改为 Cisco 后再使用 **clear counters** 命令，就会看到线路工作正常。此时，如图 5-12 所示，Num Status Enq. Sent 字段确实是随着 Num Status Msgs Rcvd 字段的增长而增长了。

例 5-12 show frame lmi 命令的执行结果

```
glock#show frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Sent 1            Num Status msgs Rcvd 1
Num Update Status Rcvd 0          Num Status Timeouts 0
glock#

glock#show frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0          Invalid Report IE Len 0
Invalid Report Request 0          Invalid Keep IE Len 0
Num Status Enq. Sent 8            Num Status msgs Rcvd 8
Num Update Status Rcvd 0          Num Status Timeouts 0
glock#
```

5.4.3 show frame-relay map 命令

命令 **show frame-relay map** 能够显示和与本地路由器相连的远端目的网络层地址对应的 DLCI，也能显示映射关系是通过静态还是动态获得的。该命令可以用来验证 **frame-relay map**

命令的执行情况和检查逆向 ARP 的工作情况。例 5-13 是该命令的输出示例。

例 5-13 show frame-relay map 命令的输出结果

```
marlin#show frame-relay map
Serial0.1 (up): ip 172.16.1.3 dlci 110(0x6E,0x18E0), static,
                broadcast,
                CISCO, status defined, active
Serial0.1 (up): ip 172.16.1.5 dlci 120(0x78,0x1C80), static,
                broadcast,
                CISCO, status defined, active
Serial0.2 (up): point-to-point dlci, dlci 130(0x82,0x2020), broadcast
                status defined, active
marlin#
```

5.4.4 debug frame-relay lmi 命令

尽管帧中继有很多 **debug** 命令可用，但大多数是用于某种特定配置类型，不适用于通用帧中继配置。然而，LMI 的调试为大多数帧中继的调试提供帮助。

debug frame-relay lmi 命令在显示 LMI 信息方面非常有用，能够使用户快速确定路由器是否在正常交换 LMI 信息。使用这条命令时，可以查看 **yourseq** 和 **myseq** 是否在递增。路由器接收到一个序列号时，它会将其加 1 并在下一次数据交换时以这个序列号返回。如果连续 3 次丢失这些 LMI 信息或 **keepalive**，就是链路被复位了。

如果该命令的执行结果表明只有一个序列号改变，很有可能是由 LMI 不匹配引起的。如果没有出现任何信息，那么路由器和帧交换机之间就有不良的链路连接。例 5-14 给出了正常工作的帧中继电路的例子。

例 5-15 显示了“异常线路”引起了 LMI 匹配错误时的链路情况。

例 5-14 debug frame-relay lmi 命令在正常链路上的输出

```
sig#debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
sig#
18:48:30: Serial0(out): StEnq, myseq 38, yourseen 37, DTE up
18:48:30: datagramstart = 0xE23E94, datagramsize = 13
18:48:30: FR encap = 0xFCF10309
18:48:30: 00 75 01 01 01 03 02 26 25
18:48:30:
18:48:30: Serial0(in): Status, myseq 38
18:48:30: RT IE 1, length 1, type 1
18:48:30: KA IE 3, length 2, yourseq 38, myseq 38
18:48:40: Serial0(out): StEnq, myseq 39, yourseen 38, DTE up
18:48:40: datagramstart = 0xE23E94, datagramsize = 13
18:48:40: FR encap = 0xFCF10309
18:48:40: 00 75 01 01 01 03 02 27 26
18:48:40:
18:48:40: Serial0(in): Status, myseq 39
18:48:40: RT IE 1, length 1, type 1
18:48:40: KA IE 3, length 2, yourseq 39, myseq 39
```

例 5-15 debug frame-relay lmi 命令在 LMI 匹配错误下的输出结果

```

sig#debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
sig#
18:59:26: Serial0(out): StEnq, myseq 7, yourseen 5, DTE up    ←missed one LMI
18:59:26: datagramstart = 0xE23E94, datagramsize = 13
18:59:26: FR encap = 0xFCF10309
18:59:26: 00 75 01 01 01 03 02 07 05
18:59:26:
18:59:36: Serial0(out): StEnq, myseq 8, yourseen 5, DTE up    ←missed two LMIs
18:59:36: datagramstart = 0xE23E94, datagramsize = 13
18:59:36: FR encap = 0xFCF10309
18:59:36: 00 75 01 01 01 03 02 08 05
18:59:36:
18:59:46: Serial0(out): StEnq, myseq 9, yourseen 5, DTE up    ←Strike three, link
down
18:59:46: datagramstart = 0xE23E94, datagramsize = 13
18:59:46: FR encap = 0xFCF10309
18:59:46: 00 75 01 01 01 03 02 09 05
18:59:46:
18:59:56: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 121 state changed to INACTIVE
18:59:56: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed state
to down
18:59:56: %FR-5-DLCICHANGE: Interface Serial0 - DLCI 121 state changed to DELETED
18:59:56: Serial0(out): StEnq, myseq 1, yourseen 5, DTE down
18:59:56: datagramstart = 0xE23E94, datagramsize = 13
18:59:56: FR encap = 0xFCF10309
18:59:56: 00 75 01 01 01 03 02 01 05
18:59:56:
18:59:57: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state
to down

```

5.5 其他帧中继配置命令

还有一些命令可以用来控制或修改帧中继网络的配置。下面列出了一些常用的调整帧中继网络特性的命令。在《IOS WAN Configuration Guide》一书中可以找到所有命令。

- **Router (config-if) #no frame-relay inverse-arp**——禁止发送逆向 ARP 信息。将这条命令和 **no arp frame-relay** 命令一起使用可禁止 PVC 动态映射。
- **Router (config-if) #no arp frame-relay**——禁止 ARP 响应，和 **no frame-relay inverse-arp** 命令结合使用。
- **Router (config-if) #keepalive keepalive_interval_in_seconds**——将默认的用于保持连接的 keepalive 间隔值从 10 秒改成另外一个数值。用这条命令时，请记住链路两端必须一致。
- **clear frame-relay-inarp**——清除动态创建的帧中继地址映射。
- **frame-relay priority-dlci-group group-number high-dlci medium-dlci normal-dlci low-dlci**——为不同类型的帧中继数据使用多个并行的 DLCI。

5.6 帧中继流量整形的设置

流量整形的工作原理是，路由器对输出的数据进行控制，使数据流与远端设备的通信速率相匹配。某些特定类型的数据信号可以经过整形满足下游用户的要求，避免下游用户出现瓶颈现象。如图 5-5 是一个部分连接的帧中继网络的情况。

在该网络模型中，远端站点需要对授权中心进行 IP 访问。任何时间都有 300 个端对端的 TCP 连接从远端站点连到授权中心。如果与中心的链路出了问题，但马上可以恢复了，在主站点到授权中心的连接上就会出现许多 TCP 连接请求，伴随着大量应用数据，这会使得 64kbit/s 速率的链路迅速饱和。远端站点以 T1 速率工作，因此传送的数据也是同样的速率，而且不知道授权中心接受数据的速率只能为 64kbit/s。FRTS 能够控制这类情况下的数据突发。

回顾前面讨论过的几个术语，其定义与 FRTS 相关：

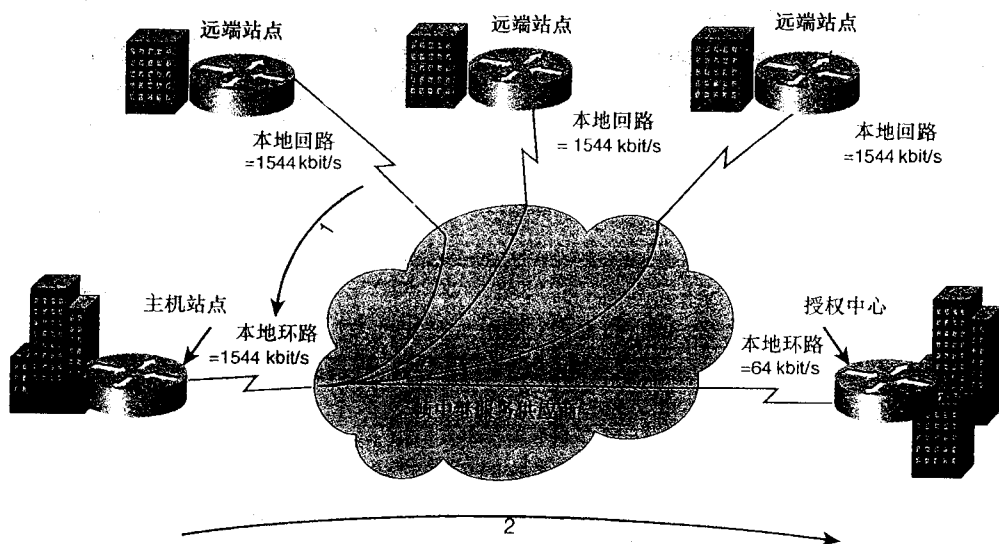


图 5-5 一个部分连接的帧中继网络

- 承诺突发量 (Bc) —— 每个 Tc 时段里传送的数据量，以位 (bits) 来计算，通常是设为 CIR 的 1/8。
- 过量突发量 (Be) —— 第一个时段里传送的超过 Bc 的那一部分多余的数据量 (bits)。Be 的值可以设置成帧中继交换机中允许突发的任意值。Be 的位数是任意的，在实际工作环境中，这些参数是由 WAN 供应商指定的。
- 承诺信息速率 (CIR) —— 期望数据发送的平均速率。在这里，CIR 不由供应商提供，而是和物理接口的速率一致。
- 时段 (Tc) —— 时段，不能超过 125ms。Tc = Bc/CIR。

- **最小承诺信息速率 (MinCIR)** ——信道拥挤时能够传送的最小数据量。该值应该设成供应商提供的实际 CIR 值。
- **传输字节 (Byte increment)** ——其值等于 $Bc/8$ ，是每个时段里传送的数据量。
- **前向显式拥塞通知/后向显式拥塞通知 (FECN/BECN)** ——每个数据帧的地址字段里有两个用于显式信号的比特位：FECN 和 BECN，它们是由服务提供商检测到信道拥挤时设置的。服务提供商的其他设备在收到的数据帧含有设置过的这两位或两个位之一，就不能将其清除，这样能为终端用户提供真实的信号。
 - **后向显式拥塞通知 (BECN)** ——这个比特指示终端用户，对于那些从接收到数据帧的设备发送过来的数据量，终端用户应该在其反方向上启动其拥挤避免程序，也意味着用户接下来在该 VC 的这个方向上传输的数据帧有可能会遇到信道拥挤的情况。
 - **前向显式拥塞通知 (FECN)** ——这个比特指示终端用户，对于那些从接收到数据帧的设备发送过来的数据量，终端用户应该在这一方向上启动其拥挤避免程序了，也意味着用户接下来在该 VC 的这个方向上传输的数据帧有可能会遇到信道拥挤的情况。

启动流量整形后，路由器在发送数据包之前会检查是否有可用的令牌桶或者标志。令牌桶里存有以一定速率放进去的令牌，它事先定义了一定的容量。如果桶中已经装满了令牌，就不能再装入新的令牌以用于后面的数据包了。因此，任何时候路由器发送的数据速率大小都和令牌桶一致。如果路由器没有足够的令牌来传送数据包，该数据包会等待令牌桶有了足够的令牌再传出去，或者直接丢弃。在数据送出接口之前都要通过为 VC 而建立的队列。默认的排队是一个先进先出队列，也可以使用定制排队或者优先级排队。可以参考 Cisco Press 的《Cisco IOS 12.0 Quality of Service》了解更多排队的知识。

BECN 响应模式是流量整形的方式之一。在 BECN 响应模式下，流量整形可以运作。如果路由器在当前时段里收到一个 BECN，它会将其传输速率降低 25%。该速率在接下来的每个 T_c 时间间隔过后都会再降低 25%，直到速率降到 MinCIR 为止。速率降低之后，如果有 16 个时段没有再收到 BECN，将会开始增加，每次增加的量是 $(Bc+Bc)/16$ 。了解了这一点，我们就可以开始具体配置 FRTS 了。配置帧中继流量整形的步骤如下：

第 1 步 串行接口上使用 **framerelay traffic-shaping** 命令允许帧中继流量整形，。

第 2 步 为需要流量整形的 VC 创建帧中继映射类。多个 VC 可以使用同一个帧中继映射，可以通过在 VC 下使用 **frame-relay class class_name_1 [in | out]** 命令来完成关联，而全局命令 **map-class frame-relay class_name_1** 则可以定义映射类。在设置映射类时，in/out 选项都是可选的，如果忽略，对进出的数据都适用。

第 3 步 在映射类配置模式中，设置下面的选项：

- **frame-relay adaptive-shaping [becn | foresight]** ——使用 BECN 流量整形，如果连接的是 Cisco IGX 或 BPX 交换机，还具有预测的功能。
- **frame-relay cir [in | out] bit/s** ——将该速率设为物理接口的速率。
- **frame-relay bc [in | out] bit/s** ——每个时段发送的数据量，比较好的设置是远端设备的 $1/8$ CIR。
- **frame-relay be out bit/s** ——要在第 1 时段发送的超过 Bc 的比特数，其值

—— **frame-relay mincir [in | out] bit/s**——VC 在收到 BECN 后可以降到的最低值。该值应该设为供应商提供的 CIR 的值。

—— (可选) **frame-relay traffic-rate cir peak_speed**——用作第二步到第五步的快捷方式。对于 CIR 来说，应该使用供应商在该线路上已经预先定好了的值。它的最大值不能超过远端路由器的物理连接速率。

第 4 步 (可选) 将任意的定制排队或优先级排队应用于映射类而不是接口。FIFO 是默认排队机制。在需要的情况下可以用定制排队或者优先级排队，可以用下面的命令进行配置：

```
frame-relay custom-queue-list list_number
frame-relay priority-group list_number
```

5.6.1 实例：帧中继流量整形的配置

图 5-6 是一个帧中继点对点网络。

该实验的目的是防止路由器 marlin 发送大量的数据拥塞路由器 glock 64 kbit/s 的 PVC，同时还要对传送到路由器 sig 的流量进行整形。路由器 marlin 和 sig 的接口速率是 1 544 Mbit/s，路由器 glock 则是 64 kbit/s。路由器 marlin 和 glock 之间的 PVC 的 CIR 由供应商设定为 32 kbit/s，而路由器 marlin 和 sig 之间的 PVC 的 CIR 为 512 kbit/s。

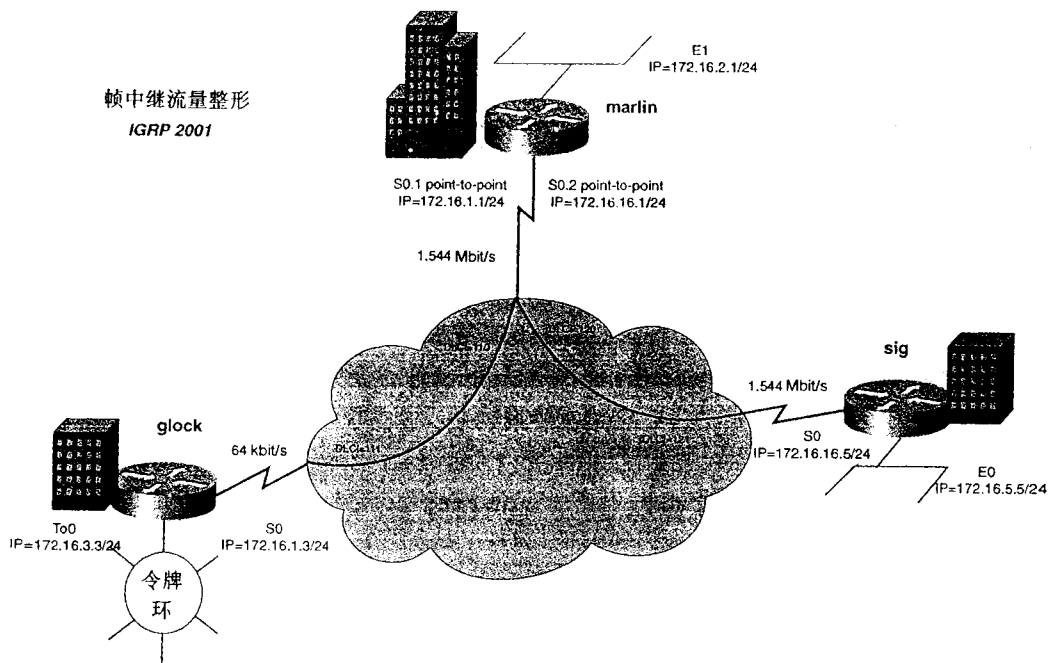


图 5-6 帧中继的流量整形

例 5-16 给出了路由器 marlin 配置的第 1 部分。

例 5-16 FRTS: marlin 路由器的配置

```
marlin(config)#int serial 0
marlin(config-if)#frame-relay traffic-shaping      ←Enable FRTS
marlin(config-if)#int s0.1
marlin(config-subif)#frame-relay class 64kb      ←Set map class
marlin(config-subif)#exit
marlin(config)#int s0.2
marlin(config-subif)#frame-relay class t1      ←Set other map class
marlin(config-subif)#^Z
```

现在可以定义两个映射类，一个叫 64 kb，另一个叫 t1。例 5-17 为映射类的配置过程。

例 5-17 映射类的配置

```
marlin(config)#map-class frame-relay 64kb
marlin(config-map-class)#frame-relay adaptive-shaping becn      ←Enable BECN response
mode
marlin(config-map-class)#frame-relay cir 1544000      ←Set to physical port speed
marlin(config-map-class)#frame-relay bc 8000      ←set to remote port speed/B
marlin(config-map-class)#frame-relay be 64000      ←Initial burst
marlin(config-map-class)#frame-relay mincir 32000      ←Carrier enforced CIR
marlin(config-map-class)#exit
marlin(config)#map-class frame-relay t1
marlin(config-map-class)#frame-relay adaptive-shaping becn      ←Enable BECN response
mode
marlin(config-map-class)#frame-relay cir 1544000      ←Set to physical port speed
marlin(config-map-class)#frame-relay bc 8000      ←set to remote port speed/B
marlin(config-map-class)#frame-relay be 64000      ←Initial burst
marlin(config-map-class)#frame-relay mincir 512000      ←Carrier enforced CIR
marlin(config-map-class)#
```

使用 **show traffic-shape** 命令和 **show frame-relay pvc dlc_i_number** 命令来验证上面所做的配置。例 5-18 分别列出了这两条命令的输出结果。

例 5-18 show traffic-shape 和 show frame-relay pvc 命令的执行结果

```
marlin#show traffic-shape
Access Target      Byte      Sustain      Excess      Interval      Increment Adap
t
I/F      List      Rate      Limit bits/int  bits/int  (ms)      (bytes) Acti
ve
Se0      56000      7875      56000      56000      125      875      BECN
Se0.1      1544000      10412      8000      64000      12      2412      BECN
Se0.2      1544000      10412      8000      64000      12      2412      BECN
marlin#
marlin#
marlin#show frame pvc 120

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 120, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

input pkts 904      output pkts 3229      in bytes 94596
out bytes 477394      dropped pkts 0      in FECN pkts 0
in BECN pkts 0      out FECN pkts 0      out BECN pkts 0
in DE pkts 0      out DE pkts 0
out bcast pkts 3204      out bcast bytes 474980
```

(待续)

```
Shaping adapts to BECN
pvc create time 19:25:28, last time pvc status changed 19:16:38
cir 1544000 bc 19300 be 64000 limit 10412 interval 12
mincir 512000 byte increment 2412 BECN response yes
pkts 3160 bytes 468932 pkts delayed 0 bytes delayed 0
shaping inactive
Serial0.2 dlc1 120 is first come first serve default queueing

Output queue 0/40, 0 drop, 0 dequeued
marlin#
```

通过这些命令可以验证上面的配置是否正确。本例中，由于没有从帧交换机接收到 BECN，因此流量整形的状态显示为 inactive。

5.7 实验 13：配置帧中继网络与控制帧中继 ARP

——第 1 部分

5.7.1 实验说明

本章开始时描述了帧中继网络在这些年的普及情况。如果还没有接触过帧中继，现在是一个很好的机会。应用如此普遍的帧中继有一个不普遍的方面，就是每个帧中继网络都有不同的设计方式。一些网络严格地以点对点的方式配置子接口，而另外一些则可能采用部分网状连接的网络形式。这个实验中读者将有机会配置多种类型的帧中继网络。

5.7.2 实验内容

假定 Dr. Evil 正在完成它称之为“Evil 信息高速公路”（EIF）的一个大工程。EIF 是用于连接 Dr. Evil 的邪恶王国中各个不同部分的一个帧中继网络。我们现在是 Dr. Evil 的追随者，要按照下面的规定设计这个网络：

- 把 IP 子网 10.10.1.8/29 作为 Secret Volcano Lair（sv_lair），Scott's house（scott's_house）和 mini-me（mini_me）之间的帧中继网络地址。
- 将 IP 子网 192.168.1.4/30 作为 sv_lair 和 starbucks_90210 之间的帧中继网络链路地址。
- IP 路由选择协议采用 EIGRP。自治系统 ID 是 666。
- 帧中继交换机在 scott's_house 和 mini_me 之间建立一个 PVC。不要允许任何数据从 scott's_house 到 mini_me 之间的 PVC 直接传输。所有的数据必须首先通过 sv_lair。
- 对 Dr. Evil 来说，只能使用 sv_lair 路由器上的子接口——否则，读者想想后果吧。

5.7.3 实验目的

- 如图 5-7 对 EIF 网络进行配置，按照图中所示对 IP 进行设置。LAN 的类型对这个

实验没有什么影响。

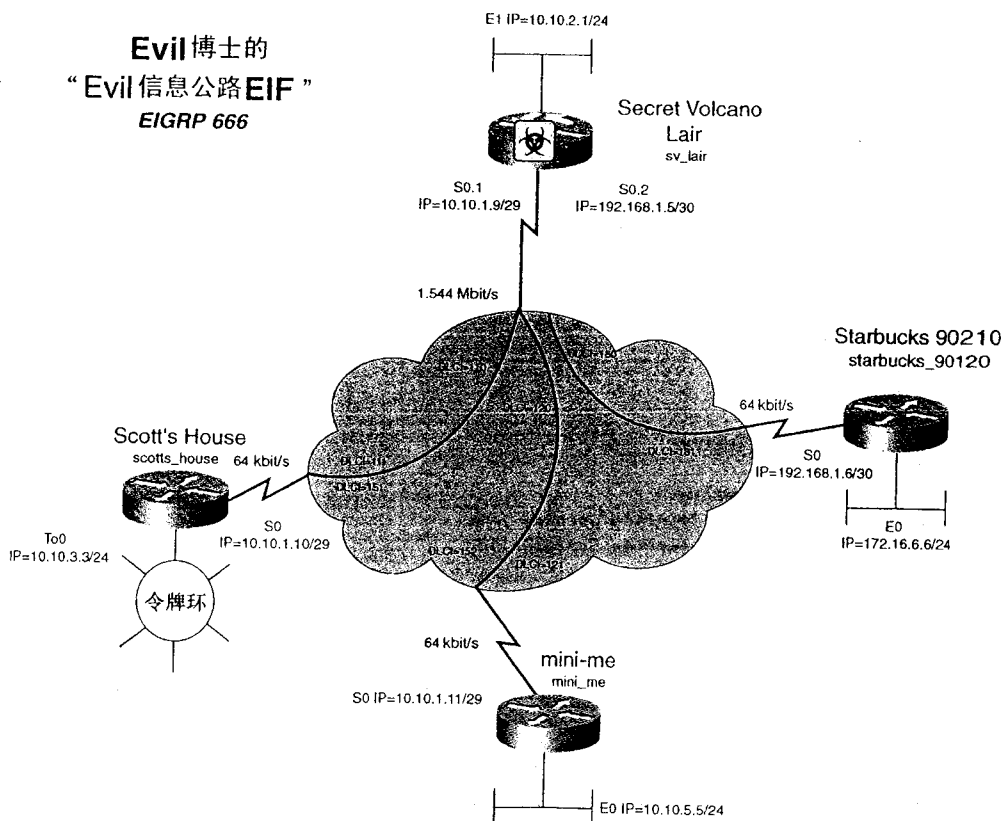
- 将帧中继作为 WAN 上的数据链路层协议。
- 禁止 scotts_house 和 mini_me 之间 PVC 的动态映射。所有从 mini_me 到 scotts_house 以及相反方向的路由都必须首先经过 sv_lair 路由器。

5.7.4 所需设备

- 5 台 Cisco 路由器，通过 V.35 背对背线缆或类似方式连接在一起。其中一台路由器用作帧中继交换机，因此需要有 4 个串行口。
- 通过集线器或交换机划分 4 个 LAN 网段，mini_me 和 starbucks_90210 的 LAN 网段可由环路接口模拟。

5.7.5 物理设计和实验准备

- 如图中所示配置帧交换机以提供 PVC。如果对帧交换机的配置有疑问，可以参考第 1 章的内容。例 5-19 是一个帧交换机配置的例子。
- 如图 5-7 用集线器和串行线缆与将路由器连接好。
- 连接 3 个以太集线器构成 3 个 LAN 网段，如图 5-7 所示。在这个实验中，作为练习之用，scotts_house 可以采用令牌环也可以采用以太网。



例 5-19 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
interface Ethernet0
  no ip address
  shutdown
!
interface Serial0
  no ip address
  encapsulation frame-relay
  no fair-queue
  clockrate 148000
  frame-relay intf-type dce
  frame-relay route 121 interface Serial1 120
  frame-relay route 152 interface Serial5 151
!
interface Serial1
  no ip address
  encapsulation frame-relay
  clockrate 148000
  frame-relay intf-type dce
  frame-relay route 110 interface Serial5 111
  frame-relay route 120 interface Serial0 121
  frame-relay route 130 interface Serial3 131
!
interface Serial2
  no ip address
  shutdown
!
interface Serial3
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 131 interface Serial1 130
!
interface Serial4
  no ip address
  shutdown
!
interface Serial5
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 111 interface Serial1 110
  frame-relay route 151 interface Serial0 152
!
```

5.8 实验 13：配置帧中继网络与控制帧中继 ARP

——第 2 部分

5.8.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将 4 台路由器以背对背的方式与帧中继交换机相互连在一起构成帧中继网络。再利用交换机或者是集线器/MAU 划分 4 个 LAN 网段。

物理搭建完成之后，为所有的 LAN 接口分配 IP 地址，如图 5-7 所示。在继续下一步实验之前，先用 **ping** 命令测试路由器的本地 LAN 接口。

接着，首先做路由器 sv_lair 的配置部分，在这上面可以使用子接口。由于 scotts_house, mini_me 和 sv_lair 这 3 台路由器的接口都在同一个 IP 子网中，必须在路由器上创建一个多点接口与之相对应。在 sv_lair 和 starbucks_90210 之间还有一个 IP 子网。对于这个子网，还需要在 Serial 0 接口上创建一个点对点接口。意识到以上的问题以后就可以开始帧中继配置的 4 个步骤了。

首先，利用接口命令 **encapsulation frame-relay** 在 Serial 0 接口上定义帧中继的封装格式。然后，定义子接口。刚才提过，需要定义的包括一个多点子接口和一个点对点子接口。例 5-20 是在路由器 sv_lair 上进行这一步的配置示例。

例 5-20 配置帧中继封装和子接口

```
sv_lair(config)#int s0
sv_lair(config-if)#encapsulation frame-relay
sv_lair(config-if)#exit
sv_lair(config)#int s0.1 multipoint
sv_lair(config-subif)#exit
sv_lair(config)#int s0.2 point-to-point
sv_lair(config-subif)#^Z
sv_lair#
```

下一步是 LMI 类型的设置。在这个例子中，LMI 的类型是 Cisco，而且我们采用的是 LMI 自动识别，因而没有必要再在第 2 步操作中进行配置。这一点对网络中所有的路由器都适用。

第 3 步指定动态或者是静态的地址映射方式。EIF 要求采用两种方式。路由器 sv_lair 利用的是接口 0.1 上的多点网络，需要静态映射方式，因而每个协议，每个站点都需要用一条 **frame-relay map** 命令进行配置。一条 **frame-relay map** 命令指向 DLCI 110 上的 IP 地址 10.10.1.10，而另一条则指向 DLCI 120 上的 IP 地址 10.10.1.11。例 5-21 是完成这些任务的配置示例。

例 5-21 多点接口的配置

```
sv_lair(config)#int s0.1
sv_lair(config-subif)#ip address 10.10.1.9 255.255.255.248
sv_lair(config-subif)#frame-relay map ip 10.10.1.10 110 broadcast
sv_lair(config-subif)#frame-relay map ip 10.10.1.11 120 broadcast
sv_lair(config-subif)#exit
sv_lair(config)#int s0.2
sv_lair(config-subif)#ip address 192.168.1.5 255.255.255.252
sv_lair(config-subif)#frame-relay interface-dlci 130
sv_lair(config-fr-dlci)#^Z
```

在点对点接口 s0.2 进行配置时，利用了逆向 ARP 解析网络地址的功能。因此，惟一需要做的是加上一条 **interface-dlci** 命令。路由器 starbucks_90210 只需要一条 **encapsulation frame-relay** 命令和一条 **interface-dlci 131** 命令就可以完成配置。

在完成所有与协议相关的问题之前，必须使各个地方的帧中继网络都可以正常工作。配置路由选择协议之前，必须能够 **ping** 通所有的本地 WAN 接口以及所有的本地 LAN 接口。

路由器 scotts_house 不允许使用与 mini_me 之间的子接口或者是 PVC。一旦 scotts_house 和 mini_me 之间链路两端都进入活动状态，逆向 ARP 就会在二者之间创建一个动态的 PVC。为了防止这种情况的发生，需要禁止反向 ARP。另外，还需要为 IP 加上两条 **frame-relay map** 命令，这两条命令使用的 DLCI 都相同，但指向不同的 IP 地址，它们使得路由器将所有的数据都通过 DLCI 111 来发送。例 5-22 是 scotts_house 路由器的配置范例。

例 5-22 scotts_house 的配置示例

```
scotts_house(config)#interface s0
scotts_house(config-if)#encapsulation frame-relay
scotts_house(config-if)#no frame-relay inverse-arp      ←Disables Inverse-ARP
scotts_house(config-if)#no arp frame-relay              ←Disables ARP
scotts_house(config-if)#ip address 10.10.1.10 255.255.255.248
scotts_house(config-if)#frame-relay map ip 10.10.1.9 111 broadcast
scotts_house(config-if)#frame-relay map ip 10.10.1.11 111 broadcast
```

路由器 mini_me 的配置与此类似，也需要两条 **frame-relay map** 命令来把 IP 地址 10.10.1.9 和 10.10.1.10 都指向 DLCI 121。例 5-23 是 mini_me 的配置示例。

例 5-23 mini_me 的配置示例

```
mini_me(config)#int s0
mini_me(config-if)#encapsulation frame-relay
mini_me(config-if)#no frame-relay inverse-arp
mini_me(config-if)#no arp frame-relay
mini_me(config-if)#ip address 10.10.1.11 255.255.255.248
mini_me(config-if)#frame-relay map ip 10.10.1.10 121 broadcast
mini_me(config-if)#frame-relay map ip 10.10.1.9 121 broadcast
```

在 scotts_house 和 mini_me 上执行 **show frame-relay map** 命令，确定 DLCI 151 和 150 上

没有动态 PVC 的存在，如例 5-24 所示。

例 5-24 show frame-relay map 显示正常的示例，没有动态 ARP

```
scotts_house#show frame-relay map
Serial0 (up): ip 10.10.1.9 dlci 111(0x6F,0x18F0), static,
              broadcast,
              CISCO, status defined, active
Serial0 (up): ip 10.10.1.11 dlci 111(0x6F,0x18F0), static,
              broadcast,
              CISCO, status defined, active
scotts_house#
```

如果 **frame-relay map** 的显示结果如例 5-25 所示，即采用的是动态逆向 ARP。

例 5-25 show frame-relay map 显示不正常的情况，使用的是逆向 ARP

```
scotts_house#show frame-relay map
Serial0 (up): ip 10.10.1.9 dlci 111(0x6F,0x18F0), dynamic,
              broadcast,, status defined, active
Serial0 (up): ip 10.10.1.11 dlci 151(0x97,0x2470), dynamic,
              broadcast,, status defined, active
```

要测试 **frame-relay map** 命令的效果，用 **ping** 命令测试本地路由器 10.10.1.9、10.10.1.10 和 10.10.1.11。

帧中继配置过程的第 4 步，也是最后一步，是解决所有与协议相关的问题。首先，配置所有路由器上的 EIGRP 协议，并把 666 做为自治系统（AS）的 ID。路由器 **sv_lair** 和 **starbucks_90210** 上，需要在 EIGRP 下加上两条 **network** 命令。网络中其他路由器则只需要配置 **network 10.0.0.0** 和 **no auto-summary** 命令。对 **sv_lair** 的配置请参考例 5-26。关于 EIGRP 的详细配置情况，可以参考第 11 章“混合协议：增强型内部网关路由选择协议（EIGRP）”。

例 5-26 sv_lair 和 starbucks_90210 上的 EIGRP 配置

```
router eigrp 666
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

这时配置过程似乎已经完成，EIGRP 相邻关系也形成了，路由信息也在交换了。但是，再进一步检查一下 **scotts_house** 和 **mini_me** 的路由表，如例 5-27 所示，就会发现 **sv_lair** 的路由传播情况不正常。路由器 **sv_lair** 并没有把 **scotts_house** 的路由信息转发到 **mini_me** 去，同时也没有把 **mini_me** 的路由信息转发到 **scotts_house** 去。然而，**sv_lair** 确实是把 **starbucks_90120** 的路由信息和它自己的本地路由信息转发到 **scotts_house** 和 **mini_me** 去了的。

例 5-27 检查路由表

```
scotts_house#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
```

（待续）

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, * - candidate default
 U - per-user static route, o - ODR
 T - traffic engineered route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks

D 10.10.2.0/24 [90/2195456] via 10.10.1.9, 00:53:29, Serial0

C 10.10.3.0/24 is directly connected, TokenRing0

D 10.10.6.0/24 [90/2707456] via 10.10.1.9, 00:52:27, Serial0

C 10.10.1.8/29 is directly connected, Serial0

192.168.1.0/30 is subnetted, 1 subnets

D 192.168.1.4 [90/2681856] via 10.10.1.9, 00:53:29, Serial0

scotts_house#

<<<<no 10.10.5.0 subnet>>>>

mini_me#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, * - candidate default

U - per-user static route, o - ODR

T - traffic engineered route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks

D 10.10.2.0/24 [90/2195456] via 10.10.1.9, 00:53:38, Serial0

C 10.10.5.0/24 is directly connected, Ethernet0

D 10.10.6.0/24 [90/2707456] via 10.10.1.9, 00:52:37, Serial0

C 10.10.1.8/29 is directly connected, Serial0

192.168.1.0/30 is subnetted, 1 subnets

D 192.168.1.4 [90/2681856] via 10.10.1.9, 00:53:38, Serial0

mini_me#

<<<<no 10.10.3.0 route>>>>

sv_lair#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, * - candidate default

U - per-user static route, o - ODR

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

C 10.10.2.0/24 is directly connected, Ethernet0

D 10.10.3.0/24 [90/2185984] via 10.10.1.10, 00:53:10, Serial0.1

D 10.10.5.0/24 [90/2195456] via 10.10.1.11, 00:53:10, Serial0.1

D 10.10.6.0/24 [90/2195456] via 192.168.1.6, 00:52:08, Serial0.2

C 10.10.1.8/29 is directly connected, Serial0.1

192.168.1.0/30 is subnetted, 1 subnets

C 192.168.1.4 is directly connected, Serial0.2

sv_lair#

<<<<all routes present>>>>

读者可能没法猜到原因，这是由水平分割引起的问题。记住一条规则：一条路由不会通过接收到该路由的接口或子接口再发送出去。要解决这个问题，应在路由器 `sv_lair` 上屏蔽 EIGRP 的水平分割功能，这是通过在 `sv_lair` 的 0.1 接口上加入下面这条命令来实现的：

```
sv_lair (config-subif) #no ip split-horizon eigrp 666
```

例 5-28 列出了做了如上更改之后，`scotts_house` 路由表的情况。请注意，现在显示了路由 10.10.5.0。

例 5-28 屏蔽掉水平分割之后，`scotts_house` 上 `show ip route` 命令的显示结果

```
scotts_house#show ip route
<<<text omitted>>>
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D    10.10.2.0/24 [90/2195456] via 10.10.1.9, 00:00:09, Serial0
C    10.10.3.0/24 is directly connected, TokenRing0
D    10.10.5.0/24 [90/2707456] via 10.10.1.9, 00:00:09, Serial0
D    10.10.6.0/24 [90/2707456] via 10.10.1.9, 00:00:09, Serial0
C    10.10.1.8/29 is directly connected, Serial0
    192.168.1.0/30 is subnetted, 1 subnets
D    192.168.1.4 [90/2681856] via 10.10.1.9, 00:00:09, Serial0
scotts_house#
```

还有一条命令可以帮助确认 EIGRP 是否工作正常，就是 `show ip eigrp neighbor`。在路由器 `sv_lair` 上执行这条命令，可以看到 3 台 EIGRP 邻居路由器，每个远程节点一个。例 5-29 是这条命令的输出示例。第 11 章还会再详细地讨论这条命令的用法。

例 5-29 `show ip eigrp neighbor` 命令的执行结果

```
sv_lair#show ip eigrp neighbors
IP-EIGRP neighbors for process 666
H   Address                Interface    Hold Uptime    SRTT   RTO  Q  Seq
                               (sec)          (ms)
0   192.168.1.6              Se0.2        128 00:04:34   32    200  0  8
2   10.10.1.10                Se0.1        179 00:08:19   33    200  0  7
1   10.10.1.11                Se0.1        161 00:08:54   24    200  0  7
sv_lair#
```

例 5-30 则是该实验中所有路由器配置情况的范例。

例 5-30 完整的配置列表

```
hostname sv_lair
!
interface Ethernet0
 ip address 10.10.2.1 255.255.255.0
 media-type 10BaseT
!
<<<text omitted>>>
!
interface Serial0
 no ip address
```

(待续)

```

encapsulation frame-relay
no ip mroute-cache
!
interface Serial0.1 multipoint
ip address 10.10.1.9 255.255.255.248
no ip split-horizon eigrp 666
frame-relay map ip 10.10.1.10 110 broadcast
frame-relay map ip 10.10.1.11 120 broadcast
!
interface Serial0.2 point-to-point
ip address 192.168.1.5 255.255.255.252
frame-relay interface-dlci 130
!
<<<text omitted>>>
!
router eigrp 666
network 10.0.0.0
network 192.168.1.0
no auto-summary
:

hostname scotts_house
!
ip subnet-zero
!
interface Serial0
ip address 10.10.1.10 255.255.255.248
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
no arp frame-relay
frame-relay map ip 10.10.1.9 111 broadcast
frame-relay map ip 10.10.1.11 111 broadcast
no frame-relay inverse-arp
!
interface TokenRing0
ip address 10.10.3.3 255.255.255.0
no ip directed-broadcast
ring-speed 16
!
router eigrp 666
network 10.0.0.0

hostname mini_me
!
ip subnet-zero
!
interface Ethernet0
ip address 10.10.5.5 255.255.255.0
no ip directed-broadcast
!
interface Serial0
ip address 10.10.1.11 255.255.255.248
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
no arp frame-relay
frame-relay map ip 10.10.1.9 121 broadcast
frame-relay map ip 10.10.1.10 121 broadcast
no frame-relay inverse-arp
!
interface Serial1
no ip address

```

```

no ip directed-broadcast
shutdown
!
interface BRI0
no ip address
no ip directed-broadcast
shutdown
!
router eigrp 666
network 10.0.0.0

hostname mini_me
!
ip subnet-zero
!
interface Ethernet0
ip address 10.10.5.5 255.255.255.0
no ip directed-broadcast
!
interface Serial0
ip address 10.10.1.11 255.255.255.248
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
no arp frame-relay
frame-relay map ip 10.10.1.9 121 broadcast
frame-relay map ip 10.10.1.10 121 broadcast
no frame-relay inverse-arp
!
router eigrp 666
network 10.0.0.0

hostname starbucks_90210
!
interface Ethernet0
ip address 10.10.6.6 255.255.255.0
!
interface Serial0
ip address 192.168.1.6 255.255.255.252
encapsulation frame-relay
frame-relay interface-dlci 131
!
<<text omitted>>>
!
router eigrp 666
network 10.0.0.0
network 192.168.1.0
no auto-summary
    
```

5.9 实验 14：帧中继网络、数据整形、OSPF 及 DLSw/LLC2 配置——第 1 部分

5.9.1 实验说明

大部分网络的增长都不是水平递增的。网络的某些部分可能具有更快更新的链路，而其

据整形技术有助于解决这个问题，它能够对数据突发实施控制，也能对某个时间内接口上发送的数据量加以控制。这样就能解决高速链路将大量的数据包发送到低速链路带来的问题。这个实验中，要在一个帧中继多点网络中配置数据整形。

5.9.2 实验内容

喷气推进实验所 Jet Propulsion Laboratory (JPL) 要从一些望远镜那里收到很多很大的图片。JPL 接收到这些图片之后，立即就会把它们上载到位于佛罗里达和休斯顿，德克萨斯的 Cape Canaveral 那里去。这些图片有时候很大，通常很快就会使得这两个国家航空和宇宙航行局 (NASA) 之间的链路达到饱和状态。为了对这种突发性的大数据流进行控制，JPL 决定采用帧中继数据整形 (FRTS) 技术。下面是配置 FRTS 时要遵守的一些规则：

- 所有的地址都属于 IP 网络 128.10.0.0。
- 在 JPL 局域网上只允许 14 个主机地址。
- 在 WAN 上采用一个 29 位的子网掩码。
- 在剩余的 LAN 上使用 24 位掩码的地址方式。
- IP 路由选择协议采用 OSPF，如图 5-8 所示分配网络区域。
- 配置帧中继数据整形时的要求是：JPL 路由器的本地端口速率是 1.544 Mbit/s。通往 nasa_houston 的 PVC 的 CIR 是 32 kbit/s，而 nasa_houston 的本地端口速率是 64 kbit/s。利用 FRTS 来配置最慢的链路，也就是对 nasa_houston 的 PVC 进行优化。
- (可选)：在两台 NASA 路由器的以太网段之间配置 DLSw，使用的封装形式是 LLC2。

5.9.3 实验目的

- 如图 5-8 对网络进行配置，按照上面的规则对 IP 进行设置。
- WAN 上的数据链路层协议采用帧中继。
- 按照前面提出的要求在 JPL 路由器上配置帧中继数据整形。
- 可选：在两台 NASA 路由器之间配置 DLSw，对等体类型为帧中继。

5.9.4 所需设备

- 4 台 Cisco 路由器，通过 V.35 背对背线缆或类似的线缆连接在一起。其中，一台路由器用作帧交换机，因而需要具有 3 个串行端口。
- 用集线器或交换机划分 3 个 LAN 网段。

5.9.5 物理设计与实验准备

- 如图 5-8，配置帧交换机以提供 PVC。如果帧中继交换机的配置方面有疑问，可以再回顾一下第一章的相关内容。例 5-31 是一个帧中继交换机的配置实例。
- 按照图 5-8 所示将集线器以及串行线缆与路由器相连。

例 5-31 帧中继交换机的配置

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
 frame-relay route 121 interface Serial1 120
 frame-relay route 165 interface Serial3 166
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 120 interface Serial0 121
 frame-relay route 130 interface Serial3 131
!
<<<text omitted>>>
!
interface Serial3
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 131 interface Serial1 130
 frame-relay route 166 interface Serial0 165
```

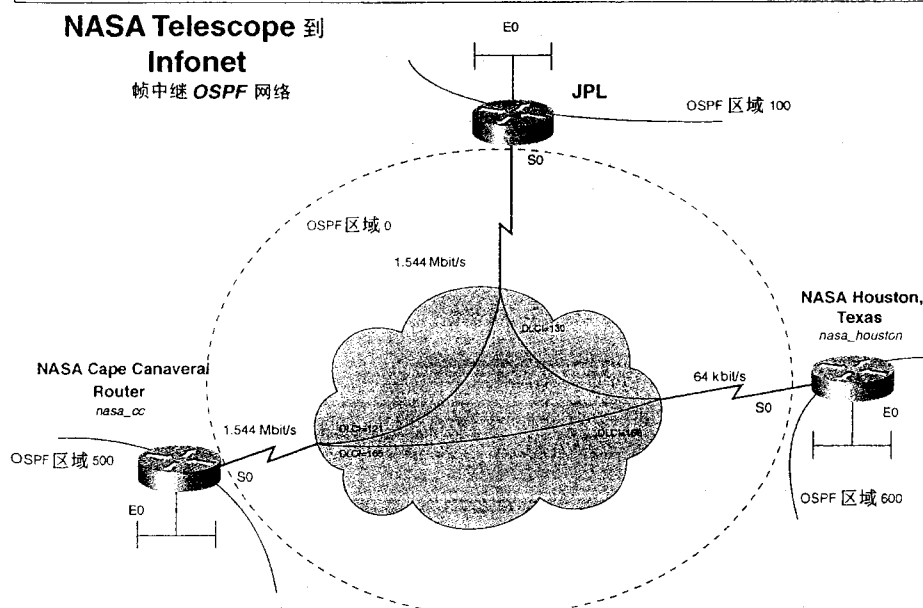


图 5.8 NASA 的帧中继网络

5.10 实验 14：帧中继网络、数据整形、OSPF 及 DLSw/LLC2 的配置——第 2 部分

5.10.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将 4 台路由器以背对背的方式与帧中继交换机相互连在一起形成帧中继网络。再利用交换机或者是集线器/MAU 划分 4 个 LAN 网段。

该实验要求对 IP 和 OSPF 进行配置。网络配置的顺序应该是，首先配置 LAN 的 IP，然后是 WAN 的 IP，最后是 OSPF。

物理连接完成之后，为所有的 LAN 接口分配 IP 地址，如图 5-9 所示。记得用 **ping** 命令对路由器的本地 LAN 接口测试连通性通过之后再继续以后操作。

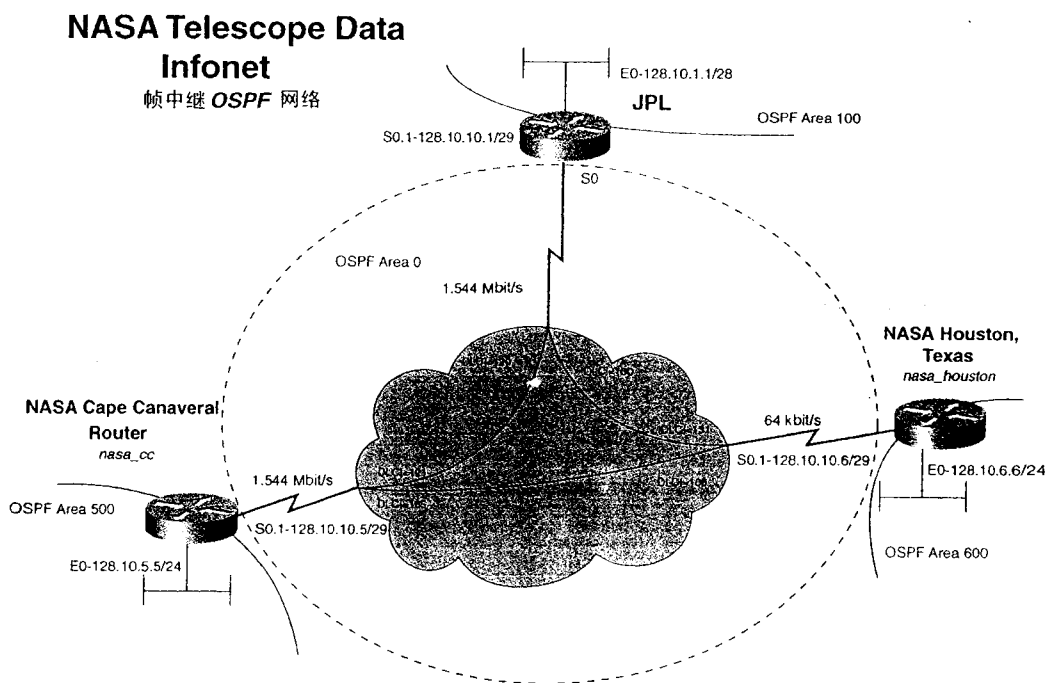


图 5-9 IP 地址分配方案

首先，来对 JPL 路由器进行配置。按照帧中配置的 4 个步骤，第一步，将接口 s0 上的封装类型设置为 **frame-relay**。接着第 2 步是设置 LMI 类型。这个例子中的 LMI 类型是 ANSI，因此，用 **frame-relay lmi-type ansi** 命令将其静态地设为 ANSI。第 3 步是将 IP 地址静态映射到 DLCI 去。JPL 路由器需要两条 **frame-relay map** 命令，一条将 IP 地址 128.10.10.5 指向 DLCI 120，另一条将 IP 地址 128.10.10.6 指向 DLCI 130。例 5-32 是 JPL 路由器到现在为止的配置示例。

例 5-32 JPL 帧中继的配置

```
interface Serial0
  no ip address
  encapsulation frame-relay
  no ip mroute-cache
  frame-relay lmi-type ansi
!
interface Serial0.1 multipoint
  ip address 128.10.10.1 255.255.255.248
  frame-relay map ip 128.10.10.5 120 broadcast
  frame-relay map ip 128.10.10.6 130 broadcast
```

在进行第 4 步配置路由选择协议之前，还要把帧中继网络的另外两个成员配置完成。

路由器 nasa_cc 和 nasa_Houston 的配置类似。每个站点都需要一条 **frame-relay map** 命令来指向 JPL 路由器，还要一条来指向另一个 nasa 节点。

例 5-33 中分别列出了 nasa_cc 和 nasa_houston 的帧中继配置范例。

例 5-33 NASA 路由器的帧中继配置

```
hostname nasa_cc
interface Serial0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay
  no ip mroute-cache
  frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
  ip address 128.10.10.5 255.255.255.248
  no ip directed-broadcast
  frame-relay map ip 128.10.10.1 121 broadcast
  frame-relay map ip 128.10.10.6 165 broadcast

hostname nasa_houston
!
interface Serial0
  no ip address
  encapsulation frame-relay
!
interface Serial0.1 multipoint
  ip address 128.10.10.6 255.255.255.248
  frame-relay map ip 128.10.10.1 131 broadcast
  frame-relay map ip 128.10.10.5 166 broadcast
!
```

要验证 **frame-relay map** 命令的效果，可以在 JPL 路由器上用 **ping** 命令对 NASA 路由器的远程串行接口进行测试。如果能到达所有的设备，LAN 和 WAN 的 IP 连通性测试都通过的话，就可以开始路由选择协议的配置了。

关于帧中继上 OSPF 的配置，可以参考第 12 章“链路状态协议：开放式最短路径优先 (OSPF)”。这个实验里并没有太多的介绍。

首先来看看 jpl 路由器，配置它的 OSPF 时需要两条 **network** 命令，一条是用于 Area 100 中的 LAN，另外一条则是用于 Area 0 中 LAN 的。配置时要当心，通配符掩码的使用

要准确，这个实验中，LAN 的掩码是 0.0.0.7，而 WAN 的则是 0.0.0.15。要确保相邻关系

的建立，应在每个 NASA 站点上都加上一条 **neighbor** 命令。例 5-34 是路由器的 OSPF 配置示例。

例 5-34 OSPF 的配置

```
hostname jpl
!
router ospf 2001
 network 128.10.1.0 0.0.0.15 area 100
 network 128.10.10.0 0.0.0.7 area 0
 neighbor 128.10.10.5 priority 1
 neighbor 128.10.10.6 priority 1
!

hostname nasa_cc
!
router ospf 2001
 network 128.10.5.0 0.0.0.255 area 500
 network 128.10.10.0 0.0.0.15 area 0
 neighbor 128.10.10.6 priority 1
 neighbor 128.10.10.1 priority 1
!

hostname nasa_houston
!
router ospf 2001
 network 128.10.6.0 0.0.0.255 area 600
 network 128.10.10.0 0.0.0.7 area 0
 neighbor 128.10.10.5 priority 1
 neighbor 128.10.10.1 priority 1
```

到现在为止，就建立了完整的 IP 连接。用 **ping** 命令对此进行测试，用 **show ip ospf neighbors** 命令查看 OSPF 邻居路由器的情况，确保每台路由器都有两个邻居路由器。如果没有形成相邻关系，检查一下 **network** 命令的情况，串行接口的 IP 地址设置以及用 **frame-relay map** 命令看看有没有配置错误。

最后一步是在 JPL 路由器的串行接口上应用 FRTS。为此，首先要启动 FRTS，其次是为命令 **frame-relay map** 配置一个帧中继映射类，FRTS 的启动是在 Serial 0 接口上执行 **frame-relay traffic-shaping** 命令来实现的。

然后是根据所给参数的情况为每条 PVC 配置一个映射类。每个映射类中需要设置的主要参数包括：

```
adaptive shaping becn
frame-relay cir
frame-relay bc
frame-relay be
frame-relay mincir
```

需要配置一个名叫 64K 的映射类，主要是定义通往 nasa_houston 的 PVC。在这个例子的配置中，映射类允许自适应数据整形的 BECN 响应，CIR 是 1544000，BC 是远程端口速率的 1/8，也就是 8000。BE 字段的设置不能超过另外一个 64000 的端口速率，在实际应用中，这个值是和 WAN 供应商提供的 QoS 参数相匹配的。这个端口上的 mincir 是设置成链路实际的 CIR 值，即这个网络中的 32 kbit/s。例 5-35 是 64K 映射类所需的配置示例。

例 5-35 64k 帧中继映射

```

1
map-class frame-relay 64k
frame-relay cir 1544000
frame-relay bc 8000
frame-relay be 64000
frame-relay mincir 32000
frame-relay adaptive-shaping becn
1

```

映射类定义好之后，用 **frame-relay class class_name** 命令把它应用到 PVC 上去，具体在这里是 **frame-relay class 64k** 命令。然后，可以用 **show frame-relay pvc** 命令验证映射类的应用情况，而 **show frame-relay pvc 130** 命令则能提供更多的信息，如例 5-36 所示。要确认图中突出显示的部分与所做的设置吻合。

例 5-36 show frame-relay pvc 130 命令的执行情况

```

jpl#show frame-relay pvc 130

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 130, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 396          output pkts 391          in bytes 30732
  out bytes 30800        dropped pkts 0           in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
  in DE pkts 0           out DE pkts 0           out DE pkts 0
  out bcast pkts 0       out bcast bytes 0
  Shaping adapts to BECN
  pvc create time 03:55:53, last time pvc status changed 02:07:28
  cir 1544000 bc 8000 be 64000 limit 9000 interval 5
  mincir 32000 byte increment 1000 BECN response yes
  pkts 225 bytes 17320 pkts delayed 0 bytes delayed 0
  shaping inactive
  Serial0.1 dlci 130 is first come first serve default queueing

  Output queue 0/40, 0 drop, 0 dequeued
jpl#

```

该实验的可选部分是要利用帧中继封装格式在两台 NASA 路由器之间创建一个 LLC2 封装的 DLSw。对于本地对等体，可以使用 LAN 的地址。同时，还需要用帧中继封装类型创建一个远程对等体，在 Ethernet 0 上定义一个网桥组并将其与 DLSw 网桥组关联，接着再利用帧中继封装类型创建一个 LLC2 对等体。要把这种类型的数据直接以帧中继方式封装，需要使用 **frame-relay map llc dlci_number broadcast** 命令。

例 5-37 分别给出了 nasa_cc 和 nasa_houston 路由器的配置示例，突出显示了 DLSw 配置部分。

例 5-37 NASA 路由器上帧中继的 DLSw 配置

```

hostname nasa_cc
!
ip subnet-zero

```

(待续)

```

!
dlsw local-peer peer-id 128.10.5.5
dlsw remote-peer 0 frame-relay interface Serial0.1 165
dlsw bridge-group 1
!
interface Ethernet0
ip address 128.10.5.5 255.255.255.0
no ip directed-broadcast
bridge-group 1
!
interface Serial0
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
ip address 128.10.10.5 255.255.255.248
no ip directed-broadcast
frame-relay map llc2 165 broadcast
frame-relay map ip 128.10.10.1 121 broadcast
frame-relay map ip 128.10.10.6 165 broadcast
!
router ospf 2001
network 128.10.5.0 0.0.0.255 area 500
network 128.10.10.0 0.0.0.15 area 0
neighbor 128.10.10.6 priority 1
neighbor 128.10.10.1 priority 1
!
ip classless
!
bridge 1 protocol ieee
!

hostname nasa_houston
!
!
dlsw local-peer peer-id 128.10.6.6
dlsw remote-peer 0 frame-relay interface Serial0.1 166
dlsw bridge-group 1
!
interface Ethernet0
ip address 128.10.6.6 255.255.255.0
bridge-group 1
!
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.1 multipoint
ip address 128.10.10.6 255.255.255.248
frame-relay map llc2 166 broadcast
frame-relay map ip 128.10.10.1 131 broadcast
frame-relay map ip 128.10.10.5 166 broadcast
!
router ospf 2001
network 128.10.6.0 0.0.0.255 area 600
network 128.10.10.0 0.0.0.7 area 0
neighbor 128.10.10.5 priority 1

```

(待续)

```
neighbor 128.10.10.1 priority 1
!
no ip classless
!
bridge 1 protocol ieee
```

利用 **show dlsw peer** 可以确认对等体是否已经处在了“连接”状态。关于 DLSw 及其验证与检测，可以参考第 13 章“配置桥接和增强数据链路交换（DLSw+）”。

作为参考，例 5-38 列出了 JPL 路由器的配置示例。

例 5-38 JPL 路由器的配置列表

```
hostname jpl
!
interface Ethernet0
 ip address 128.10.1.1 255.255.255.240
 media-type 10BaseT
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay traffic-shaping
 frame-relay lmi-type ansi
!
interface Serial0.1 multipoint
 ip address 128.10.10.1 255.255.255.248
 frame-relay class 64k
 frame-relay map ip 128.10.10.5 120 broadcast
 frame-relay map ip 128.10.10.6 130 broadcast
!
router ospf 2001
 network 128.10.1.0 0.0.0.15 area 100
 network 128.10.10.0 0.0.0.7 area 0
 neighbor 128.10.10.5 priority 1
 neighbor 128.10.10.6 priority 1
!
ip classless
!
map-class frame-relay 64k
 frame-relay cir 1544000
 frame-relay bc 8000
 frame-relay be 64000
 frame-relay mincir 32000
 frame-relay adaptive-shaping becn
!
```

第 6 章

WAN 协议与技术： 通过多协议传输 语音

Eric Sandberg 供稿

CCIE 的实验可能包括使用 Cisco 具有语音传输功能的路由器来完成语音传输的任务（也就是说，通过帧中继、IP 或者 ATM 传输语音）。本章的目的就是通过详细的讲解帮助读者完成配置任务，但本书并不是关于电话技术和语音传输技术的惟一权威书籍，有很多优秀的读物以及 Cisco 系列的书籍都能够满足广大读者这方面的需求。我们向读者推荐 Cisco Press 出版的《Cisco Voice over Frame Relay, ATM, and IP》和《Voice over IP Fundamental》，还有其他一些由 Cisco Systems 认证培训伙伴提供的与电话和语音技术相关的课程。

本章提供了 Cisco 语音传输技术的解决方案，分析这些技术在网络中的应用，讨论这些技术的配置和运用问题，并列举这些语音技术的优点，包括降低长途成本，在占用较少带宽的情况下支持多呼叫，更多更好的服务以及对 IP 协议更有效的利用。

本章开始简单介绍模拟电话技术的基础知识。目的是使网络工程师们学习如何使用 Cisco 的各种多服务访问设备，如 Cisco 1750、2600、3600 和 3810 等在企业网络或管理网络环境下设计、组合以及配置通过帧中继、ATM 和 IP 传输语音。

6.1 模拟电话技术简介

本节讲解电话网络的问题，重点关注模拟电话技术，包括下面这些主题：

- 电话呼叫的组件。
- 电话设置的过程。
- 电话信令。
- 本地环路。
- 语音交换机。
- 中继。
- 中继/线路抢占信号类型。
- 电话呼叫过程。

6.1.1 电话呼叫的组件

如图 6-1 所示，基本的电话呼叫组件包括电话，本地环路，语音交换机（CO/PBX）以及

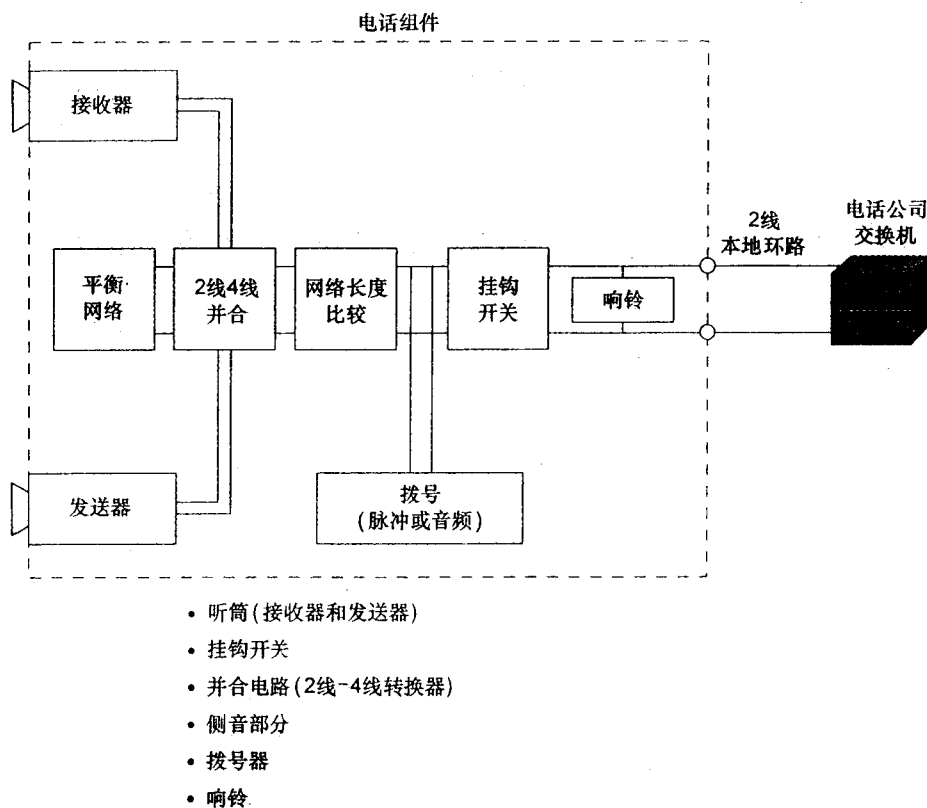


图 6-1 电话的基本组成部分

相应的通信中继。每个人都对电话很熟悉，但未必熟悉电话机的组成部分。电话听筒是读者握在手里讲话（发送）和听话（接收）的部分。挂钩开关是一个控制杆，听筒放在支架上时就会将它压下去（on-hook，断开状态）。如果拿起听筒准备打一个电话，挂钩开关就会弹起来，此时的状态（off-hook，接通状态）能够允许电流流过电话机开始工作。

听筒分成两对 4 条导线，一对用于发送一对用于接收。每台电话机里都有一个混合的 2 线到 4 线的转换器，可当作一个通信网桥，提供 4 线听筒和 2 线本地环路之间的转换功能。侧音部分也是一个混合电路，使得话音的一部分传递到听筒使用户能够知道自己说话的音量如何。拨号器，无论是按键式的还是转动式的，用于向电话局发出用户正拨打电话的信号。在按键式电话上按下按键或者在转动式电话上摇动拨号器都会送出信号到电话局，告诉公司所拨的电话号码。

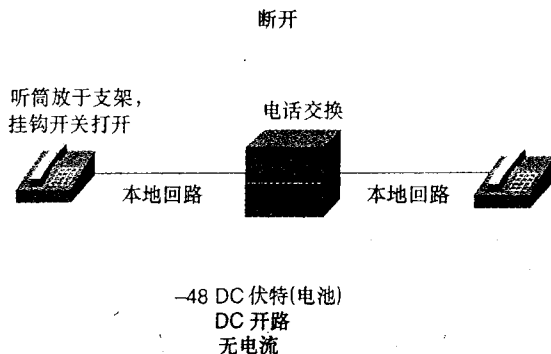
最后一个组成部分就是振铃。当有电话时，电话局会通过电线传送电压信号到电话机里，这个信号触发振铃电路发出铃声通知用户。

6.1.2 电话信令

拨打电话和接听电话时，必须将自己的意图告知电话局。信令完成了这项工作。本章要讨论的两种电话信令就是管理信令和地址信令。管理信令用于用户和电话局相互通知各自的呼叫状态。管理信令的 3 种类型是断开（on-hook），接通（off-hook）和振铃（ringing）。

如前所述，将听筒放在支架上（on-hook）使挂钩开关断开，没有电流在话机中流动，此时只有振铃电路工作。断开状态的信令方式请参见图 6-2。将听筒提离支架使得电流在话机中开始流动，并向电话局发出通话请求。反过来，电话局向话机返回一个拨号音表明它的线路已经准备好了。图 6-3 表示了接通状态的信令方式。当有人拨打，电话会发送电压信号到振铃电路，而电话局也会向拨号者发送拨号音，告诉拨号者正在向受话者的电话发送振铃电压信号。振铃时如图 6-4 所示。

地址信令方式有两种，脉冲和双音多频（DTMF）。轮盘式电话尽管有些过时，但仍然还有人在使用。这种电话使用脉冲地址信令。每个脉冲包括一个闭合信号和一个断开信号。闭合信号是电路接通期间的信号，而断开信号则是电路断开期间的信号。一个信号周期应该由 60% 的断开信号和 40% 的闭合信号组成。拨号盘里的控制器控制着信号按照这样的比率来形成脉冲。脉冲地址信令过程如图 6-5 所示。



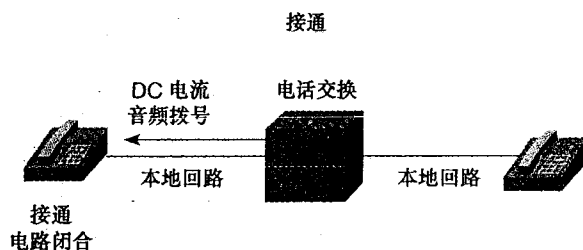


图 6-3 管理信号——接通状态 (Off-Hook)

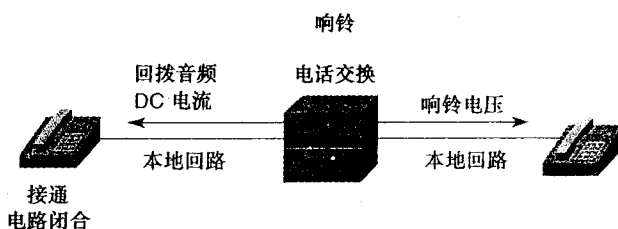


图 6-4 管理信号——振铃状态 (Ringing)

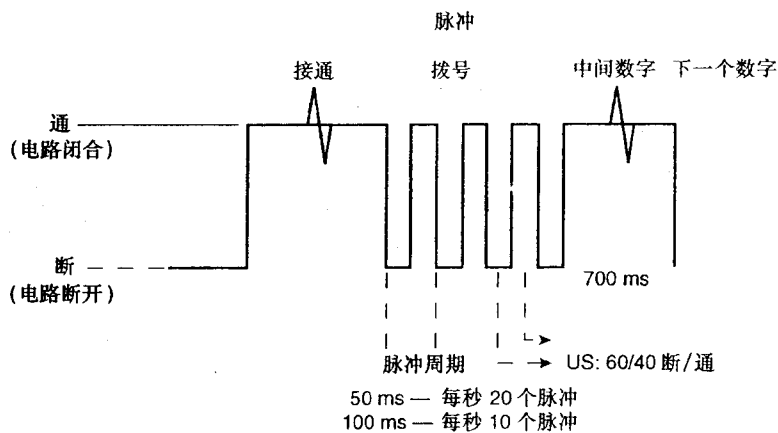


图 6-5 地址信令——脉冲

按键式电话使用双音多频 (DTMF) 信令。拨号盘上每一行按键通过低音频来识别，而每一列的按键则通过高音频来识别。将两个音频组合就能够通知电话局按下的是什么键（这就是 DTMF 这个名称的来历）。

图 6-6 是每个按键的音频组合情况。

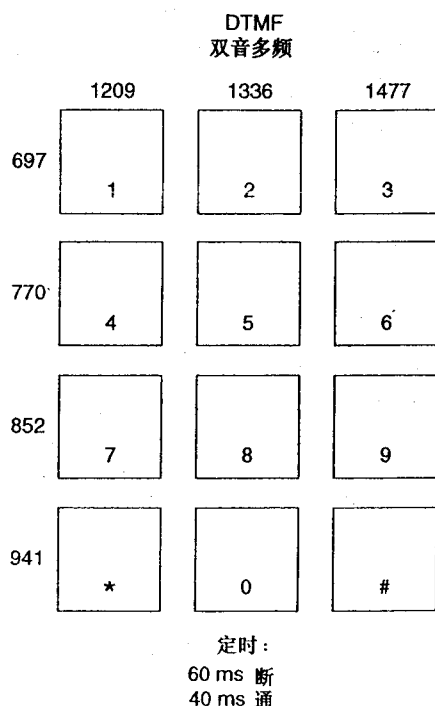
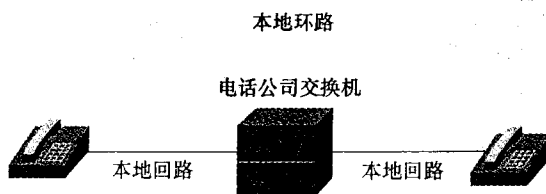


图 6-6 地址信令——双音多频 (DTMF)

6.1.3 本地环路

电话通过本地环路与电话局相连，该环路通常被人们称为“最后一里”。这一环路含有两条线路的电气通信路径，一条用于发送一条用于接收语音信号（请参考图 6-7）。这个双线电路通常叫做套线和芯线。芯线连接到电话局的电池负极而套线接地。当听筒离开支架，线路接通，电流会通过环路流动，电话局就能够为该话机提供服务。



环路是从用户到电话公司交换机的物理线对

图 6-7 本地环路

6.1.4 语音交换机

中间媒介交换机和专用分组交换机（PBX）。语音交换机是机械或电气设备，能够将语音请求传递到适当的目的地。语音交换机通常安放在控制室或者电话局。语音交换机选择性地建立和释放通信双方之间的连接，以提供通话双方进行语音信息交换的专用信道。信道在信息交换之前建立，在通话双方中止会话之前，交换机都会维持着专用信道。

私人电话直接连到中心局（CO）交换机。当电话拨入中心局（CO）交换机时，会将此呼叫转发到：

- 另外一个中心局（CO）交换机。
- 终端用户的电话（如果是连接在同一个 CO 上的）。
- 中间媒介交换机。

CO 交换机提供了电话工作的全部要素——如电池、电流检波器、拨号音发生器和振铃发生器。电池是电路和电话的电源。电流检波器通过查看电路通断来监控电路的状态。拨号音发生器用于产生确认服务请求的拨号音。当 PBX 检测到其接口上有电流流动时，拨号寄存器就会接收所拨号码。振铃发生器则通过向被拨方的电话振铃电路发送振铃信号到来通知电话的到来。

中间媒介交换机主要用作交换机之间呼叫转发的中间媒介，连接中继线路。

专用分组交换机（PBX）用于私人领域。例如商业领域，如果将办公室的每台电话都直接与 CO 交换机连在一起，线路远远是不够的。可以在办公室安装一台 PBX 交换机用于通过中继线路连接办公室的电话和 CO 交换机。

图 6-8 比较了不同的语音交换机。

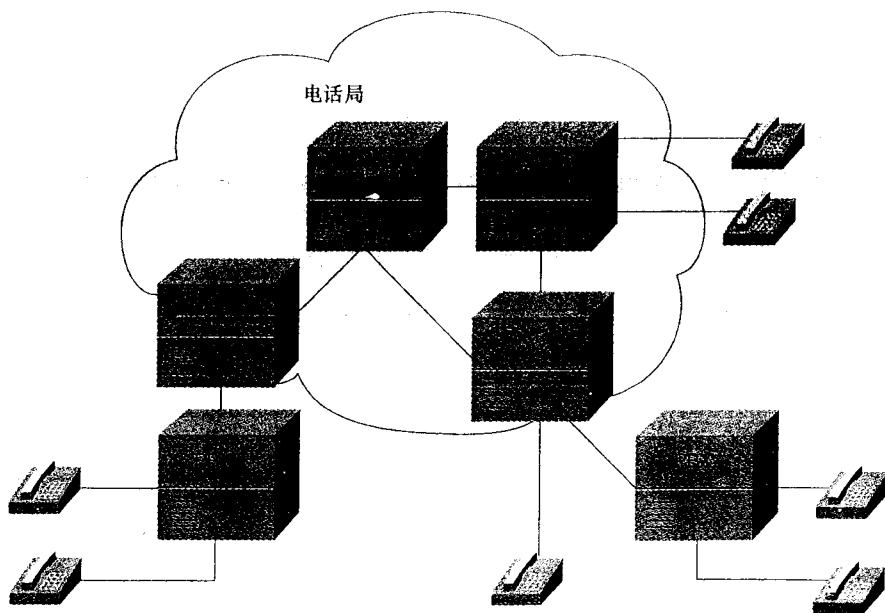


图 6-8 语音交换机——中心局交换机（CO）、中间媒介交换机和专用分组（PBX）交换机

6.1.5 中继

中继的主要作用就是在交换机之间提供通信的路径。很多不同的用户共享一条中继。但

是每次一条中继只能有一个用户使用。所以往往两台交换机之间有很多中继链路。一些常见的中继类型包括私用中继线路，中心局中继，外部交换中继和直接进出的拨号中继，如图 6-9 所示。

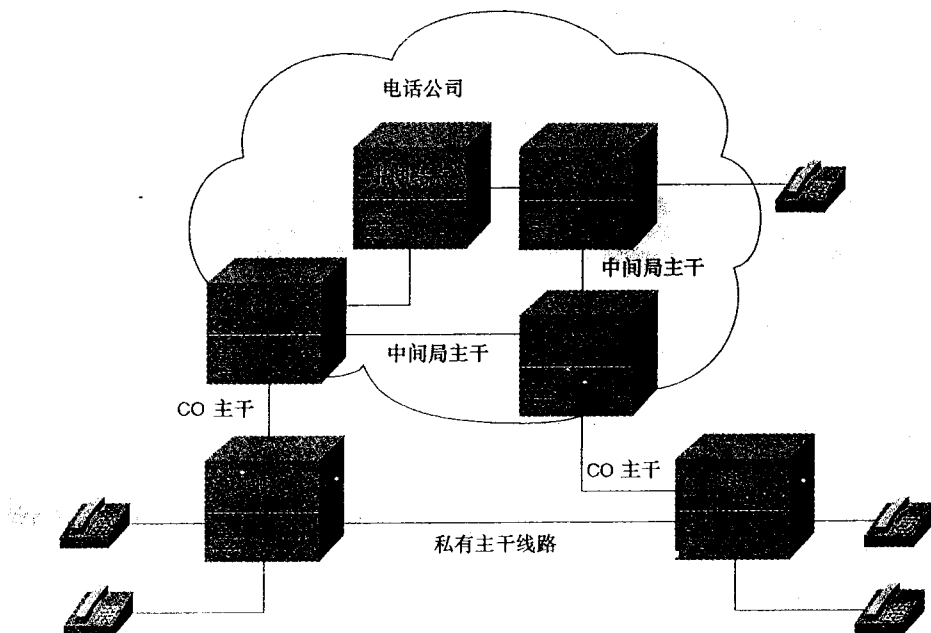


图 6-9 中继——私用、中心局（CO）和外部交换中继

私用中继通常是连接专用分组（PBX）交换机，而中心局中继则连接本地中心局（CO）交换机和 PBX。外部交换中继能使交换机认为远程电话和该交换机是直连的，要做到这一点，还需要设立外部交换局（FXO）和外部交换站（FXS）。

FXO 位于交换机连接的终端位置。它直接插入交换机的线路端，使得交换机认为 FXO 接口就是一部电话。交换机通过向 FXO 发送一个振铃电压信号来通知 FXO 有电话打进来，同样，FXO 也是通过闭合环路让电流流动来答复拨入的电话的。电流流动时，FXO 接口会使用某种电流技术将信号传送到 FXS。

FXS 位于远端，对电话来说，FXS 就像交换机，能够为电话提供拨号音和电池支持。电话会认为它就是交换机。

直接拨入（DID） 中继是一条单向的中继，用户可以在没有操作员干预的情况直接拨入到 PBX。中心局（CO）知道哪一个电话呼叫应该传递到 DID 中继上，因为它为每条 DID 中继都分配了一组号码。

直接拨出（DOD） 中继也是一条单向中继，允许用户直接连接到 CO。例如，用户想拨打电话到其公司网络以外的地方，就只需要先拨一个访问号码，如 9，PBX 就将该拨号转到 CO，然后 CO 再提供一次拨号音，并使用剩余的拨号号码将呼叫转到最终的目的地址。

图 6-10 所示为 DID 和 DOD 中继的例子。

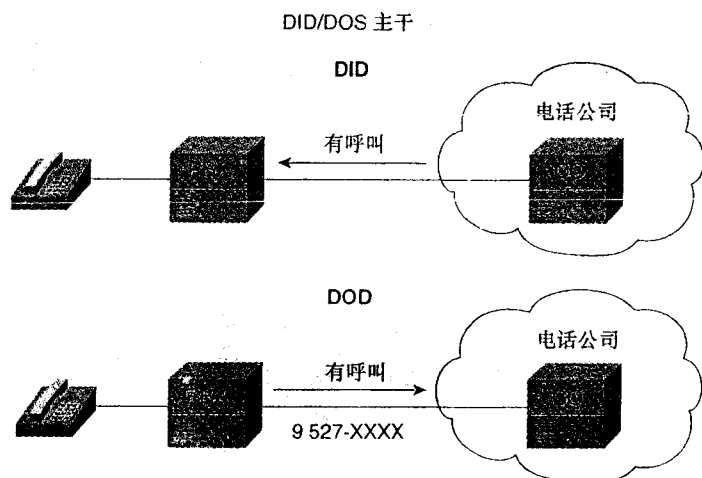


图 6-10 中继——直接拨入/拨出中继

6.1.6 中继抢占信令的类型

本节讲述中继/线路抢占的信令类型，这是电话网络中线路和中继之间的信令标准。重点讲解的是环路开始信令方式，接地开始信令方式和E&M信令。这些对于成功配置和应用Cisco语音传输系统非常关键。

注释 E&M信号通常称为耳与口，或者接收和发送，但是其起源却是地与磁的说法。地代表电气的信号地，而磁则代表用于产生语音的电磁体。

当呼叫发生时，环路开始信令使用户或电话局能够占用一条中继或线路，如图6-11所示。该信号主要是用在本地环路而不是中继上。前面说过，如果一条线路处于空闲状态，该线路所连接的电话是处在断开状态（on-hook）的。将听筒拿离支架之后，挂钩开关就进入了接通状态（off-hook），环路闭合。电流开始在电路中流动，中心局（CO）检测到电流的流动就会返回一个拨号音。如果电话振铃通知用户有电话拨入，中心局（CO）将交流振铃电压叠加到电压为-48 VDC的电池上，使振铃发生器发出振铃通知。当PBX或电话用户接电话时，中心局（CO）会将该振铃电压去掉。

对于用户很多的中继网络来说，环路开始信令是一种容易出问题的解决方法，这主要是因为电话用户经常可能在两端同时抢占中继的占用权。这种情况被称为相互瞪视（glare）。读者都有这样的经历，拿起听筒准备向某个人打一个电话，却发现对方在电话线的那一端正在通话。这就是双方同时占用了一条中继环路，是一次相互瞪视（glare）。这在家里可能不是什么问题，但是在办公地点就不一样了。能够在连接的两端检测环路或中继的抢占占用情况的信令可以解决这个问题。

接地开始信令是环路信令做过修改之后的一种信令，能够通过连接的两端进行电流检测而避免相互瞪视的情况的发生，如图6-12所示。接地开始信令在用户多、中继密度大的环境下非常有用。

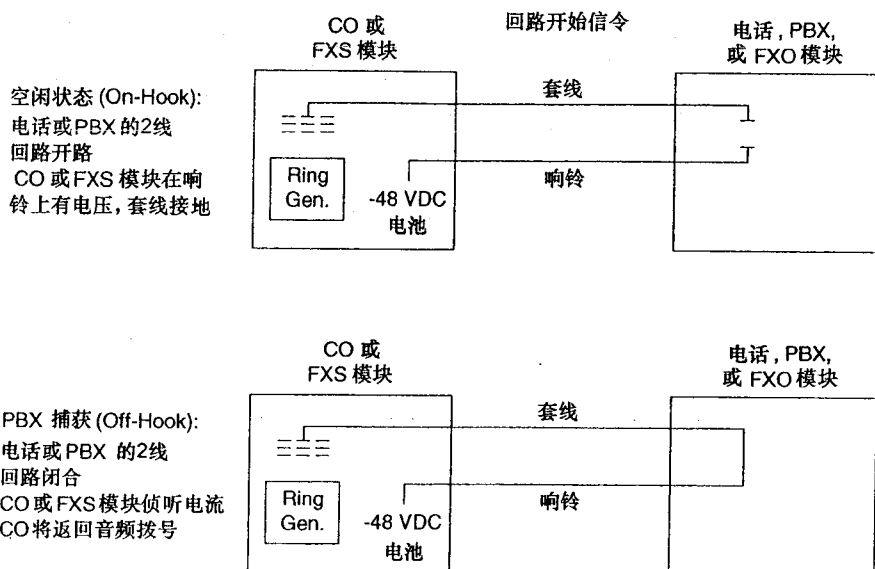


图 6-11 环路开始信令方式

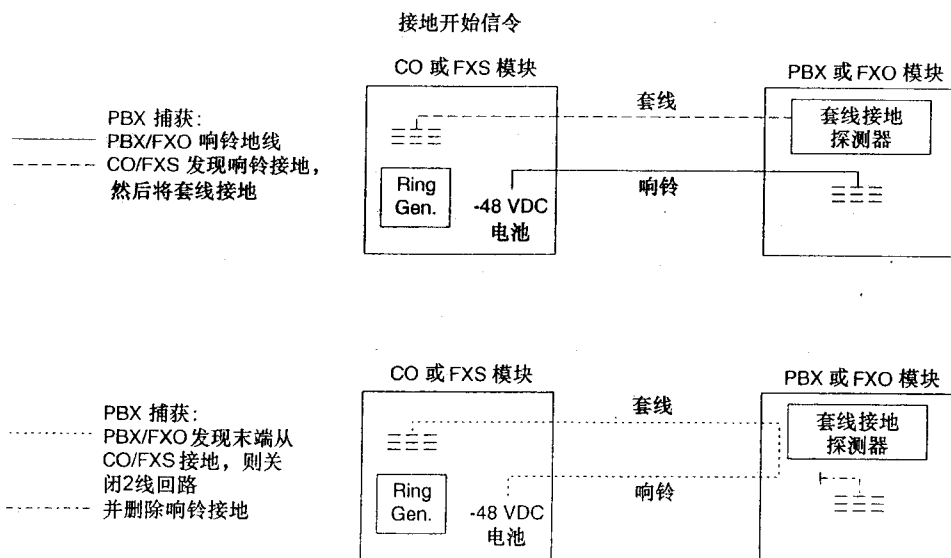


图 6-12 接地开始信令

下面是接地信令方式的工作过程总结:

- 1 线路处于空闲或断开 (on-hook) 状态时, PBX 会监测套线是否接地。
- 2 振铃时, CO 的电池开始供电。
- 3 听筒拿起之后, PBX 将振铃接地。
- 4 CO 检测到振铃接地, 将套线接地。

5 PBX 检测到套线接地，闭合电路开始一次通话。

E&M 信令支持集线设备功能或在语音交换机之间的信号。E&M 信令不是将语音和信号叠加在同一条线路上，相反地它是将二者分开用不同的导线传输。读者可能了解到，M 导线发送信号而 E 导线则接收信号。例如，如果要打电话给远端的一个用户，本地的 PBX 会为该请求在两个用户之间分配用作中继的信号导线。PBX 通过其 M 导线向对方发出请求，对方在其 E 导线上发现有电流流动而检测到该请求。远端的 PBX 将一个拨号登记发送到中继上和本地的 PBX。本地 PBX 再发送出去所拨的号码，远端 PBX 用其 M 导线通知本地拨号者通话可以开始了。如表 6-1 所示，一共有 5 种类型的 E&M 信令。

表 6-1

E&M 信号

PBX 至中间媒介设备			
类 型	导 线	断开状态	接通状态
I	M	Ground	Battery (-48VDC)
II	M	Open	Battery (-48VDC)
III	M	Ground	Battery (-48VDC)
IV	M	Open	
V	M	Open	
中间媒介设备至 PBX			
类 型	导 线	断开状态	接通状态
I	E	Open	Ground
II	E	Open	Ground
III	E	Open	Ground
IV	E	Open	Ground
V	E	Open	Ground

E&M 类型 I 信号方式（图 6-13）常用于北美的双线信号类型，一条线是 E 导线，另一条是 M 导线。北美大约 75% 的 PBX 都使用该类 E&M 信令方式。使用 E&M 类型 I 信令，集线设备通过将 E 导线接地（ground）来产生送给 PBX 的 E 信号，PBX 通过阻抗负载发现有电流增加从而检测到送来的 E 信号。同样的 PBX 是通过将电流连接到集线设备来产生 M 信号的，集线设备也用了一个阻抗负载来检测送来的 M 信号的。E&M 类型 I 接口 PBX 和集线设备使用同一个信号地。

E&M 类型 V（图 6-14）也是一种双线信号方式，通常用在北美以外的地区。和类型 I 一样，E&M 类型 V 的一条线作为 E 导线，一条作为 M 导线。E&M 类型 V 也要求 PBX 和集线设备共用同一信号地，这由信号地导线提供。

E&M 类型 II 是一个 4 线的接口（图 6-15）。像 E&M 类型 I 和 V 一样，E&M 类型 II 用两条线作为 E 和 M 导线。然而，E&M 类型 II 还把剩下的两条线作为信号地和信号电源，这两条线也用作 E 和 M 导线的环路。E&M 类型 II 接口不需要公共地，每个信号都有自己的

E&M 类型 I

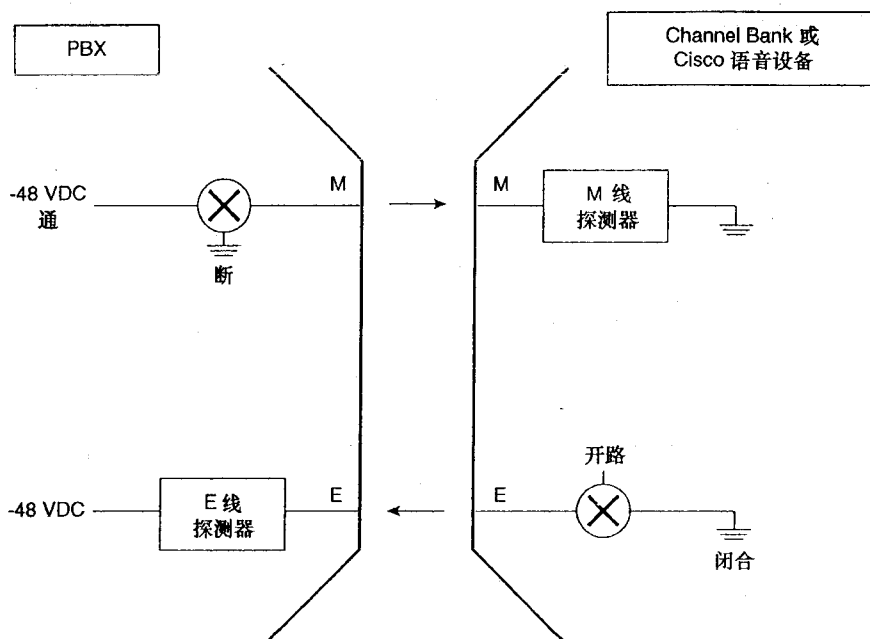


图 6-13 E&M 类型 I

E&M 类型 5

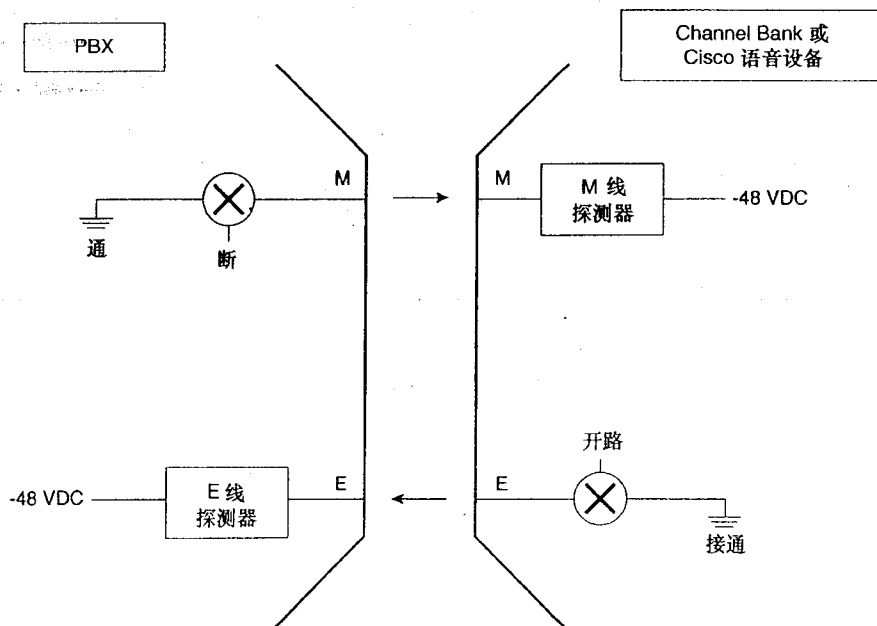


图 6-14 E&M 类型 V

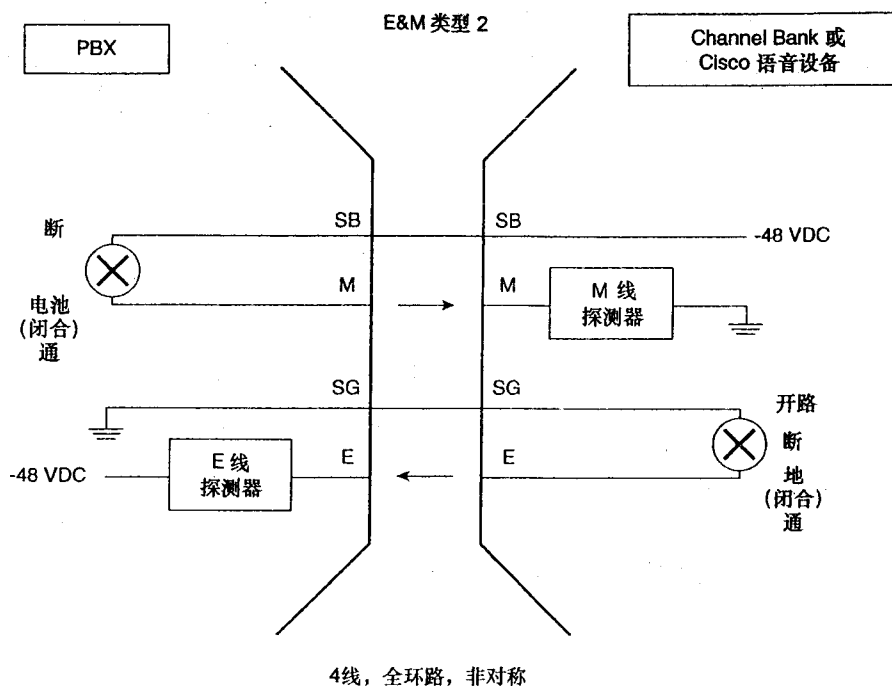
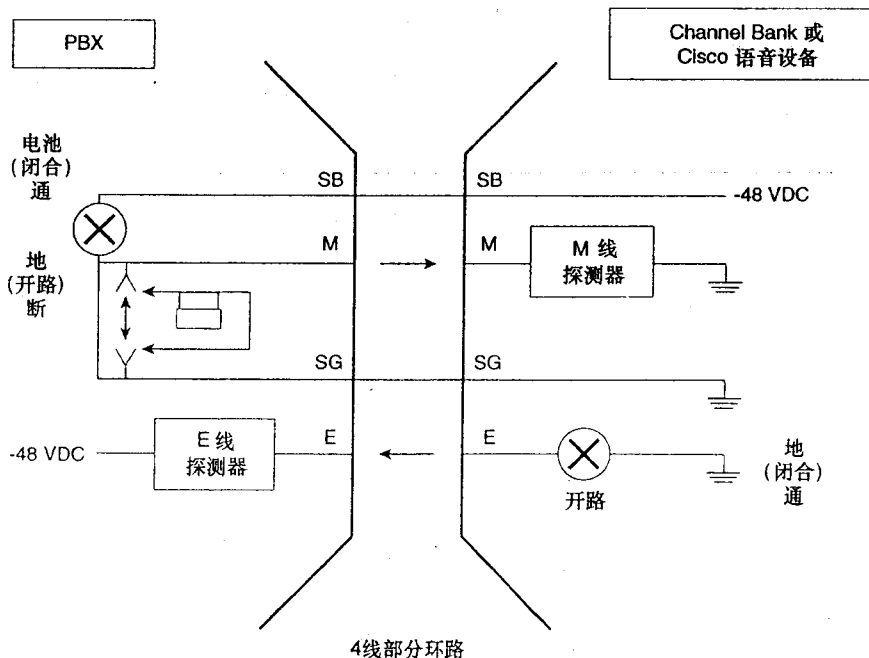


图 6-15 E&M 类型 II

E&M 类型 III 和类型 II 很相似，但它把信号地作为公共地使用（图 6-16）。E 导线和类型 I 的工作方式一样。在该配置中，PBX 发出 M 信号是通过将该导线接地而不是为它打开一个电流环路来实现的。这不是一个公共信令类型。

E&M 类型 3



E&M 类型 IV 是对称的，不需要公共地（图 6-17）。发送信号时，链路每端都会闭合一个电流环路，通过一个阻抗负载检测电流的流动以证实信号的存在。Cisco 不支持 E&M 类型 IV。

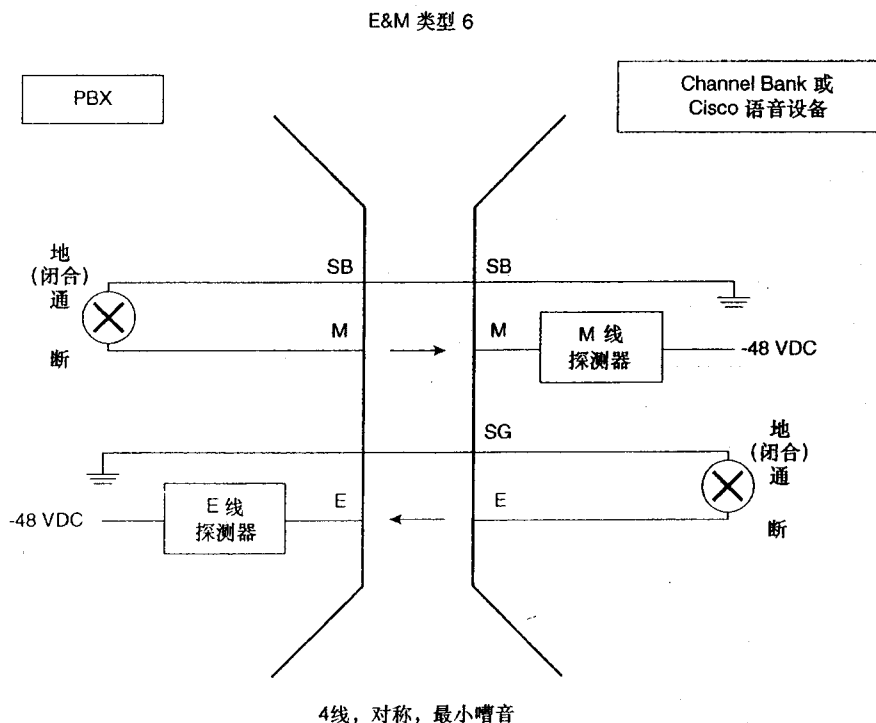


图 6-17 E&M 类型 IV

6.1.7 中继监控

本节讲述用于中继监控的开始协议，包括闪烁开始、延时开始和立即开始几种方式。链路中继线具有双向监控信号，它允许链路的任一端来占用中继。PBX 占用了中继并希望从链路的另一端得到确认应答。本地终端则需要能够区分返回的确认信号和远端请求信号。

最常用的 E&M 中继抢占信号类型是闪烁开始 (wink start) 信令。对于这种类型的信令，拨出电话的一方通过拿起听筒占用线路。远端的一方在检测到呼入方占用了线路之后并不马上返回接通确认。相反，远端电话维持其断开状态 (on-hook)，直到接收号码登记完成。接着，被叫方触发其接通状态 (off-hook) 并维持一段时间（这就是闪烁一词的来历）。拨叫方接收到这一闪烁信号之后，会将所拨的号码发送到远端用户去。被叫方接电话，PBX 则发出 M 导线的信号。图 6-18 为这种闪烁开始信号方式的示意图。

干线监控信号闪烁开始

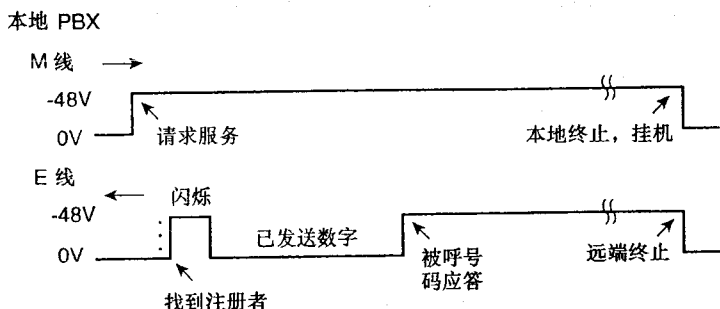


图 6-18 中继监控信令: 闪烁开始

对于延时开始信令，拨打电话的一方拿起听筒并且延时大约 200ms，然后再检查远端的一方是否还处在断开状态 (on-hook)。如果远端处于断开状态，本地电话就会将拨叫号码发送过去。如果远端处于接通状态 (off-hook)，本地电话则会继续等待直到对方进入断开状态再将拨叫号码发送过去。延时信号实际上就是说“等一下，我还没有准备好接收号码”。图 6-19 就是延时信令的过程示意图。

干线监控信号延时开始

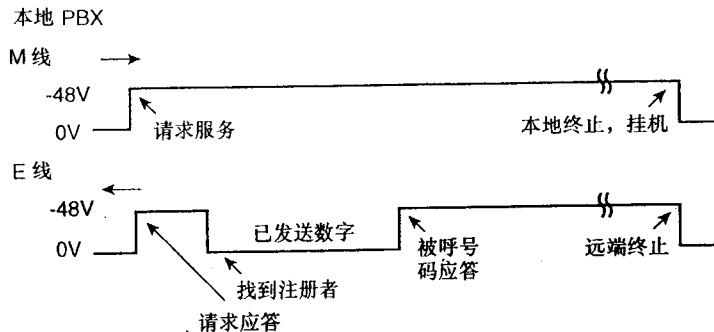


图 6-19 中继监控信令: 延时开始

干线监控信号立即开始

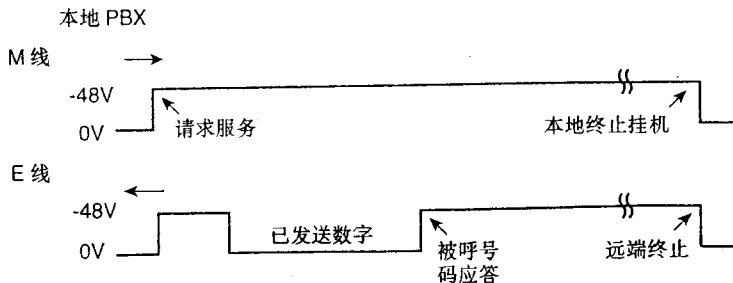


图 6-20 中继监控信令: 立即开始

现在再看一下立即开始信令的过程。立即开始的信令中呼叫方的 PBX 通过进入接通状态 (off-hook) 占用线路 (图 6-20)。本地 PBX 不是去等待远端发送过来的双重确认信号,而是等待一个预先设定好的时间 (比如说 150ms) 然后无论何种状态都发送拨叫号码。远端 PBX 只有在被叫方接电话以后才会为该次呼叫发送一个确认信号。

6.1.8 2 线到 4 线转换和回音

本地环路由两条线构成。本地环路到达中心局 (CO) 交换机时,交换机会用 2 线-4 线转换器将其转换成 4 条线。这对于通过中继在网络中传输的信号来说很有必要。如果线路之间的阻抗匹配很好,该混合电路就是平衡的,没有或很少反射能量。然而,如果线路之间的阻抗匹配不好或者平衡不够,一部分发送出去的语音就会反射回听筒,产生回音。

某种程度的回音总是存在的。如果回音的幅度或者音量太高,那就成问题了。

有两种常见的回音形式,一种是说话者回音,说话者的声音反射了回来 (幅度大于我们前面讨论过的侧音),说话者会两次听见自己的声音。另外一种受话者回音,使得受话者能够两次听到讲话者的声音。回音是由延时引起的。

解决这一问题的方法有两个,回音抑制和回音抵消。回音抑制主要抑制返回路径中讲话者的声音,回音抑制器确定返回路径中哪些信号与讲话者的声音匹配而哪一些与对方的声音相匹配。回音抑制器确定了在返回路径中存在着回音时,或者对其进行衰减或者干脆切断传输路径。但是,如果回音抑制器发现路径中双方的声音都存在时,在不影响语音质量的情况下衰减回音是不可能的。一个更好的方法就是回音抵消。在处理回音时,回音抵消不是去对回音进行衰减或者切断其路径,而是使用回音抵消器建立声音的声谱数学模型,将回音从传输路径中减去。

注释 回音抵消器只能去除电路一端的回音。如果电路的两端都存在回音,则另一端也需要回音抵消器。

总结一下打电话时的一些基本要素。电话听筒放在支架上时 (on-hook), 线路处于空闲状态,电话机或者 PBX 断开双线环路。将听筒拿离支架使得挂钩开关接通 (off-hook), 环路闭合,电流开始在电路中流动。交换机检测到电流流动后返回一个拨号音。接收到拨号音之后,拨号者通过拨某个电话号码请求建立连接。交换机通过向被叫方发送振铃电压信号通知其有电话进来。同时,交换机回送振铃信号给拨号者,通知呼叫正在进行。被叫方拿起听筒,闭合了环路之后,此时模拟链路建立完成。

6.2 数字语音技术

数字环路载波技术是在上个世纪 70 年代早期为了通过数字技术提高传输性能而发展起来的。相比于模拟技术,数字技术除了增强了传输的性能之外,还提高了传输的可靠性以及其可维护性,主要原因是数字信号的可再生性,不像模拟信号那样易于产生噪声的累积。模

（没有电脉冲）的序列。

6.2.1 模拟信号的数字化

模数变换是通过编码解码器（codec）来实现的。编码解码器用于将语音频率信号通道转换成 64 kbit/s 的数字信号（DS0）通道。编码解码器是通过对信号进行抽样、量化和编码来实现转换的。在深入讲述模数变换的 3 个步骤之前，读者要先了解一下奈奎斯特（Nyquist）定理（图 6-21）。奈奎斯特（Nyquist）定理如下：

在发送端对一个信号以信号最高频率的 2 倍或者更高的速率对其进行瞬间抽样，得出的抽样结果含有足够的信息能在接受端将该信号准确恢复出来。

语音信号的最高频率是 4000 Hz，因此，要在接受端恢复语音信号，应该每秒进行 8000 次抽样，或者说每 125ms 进行一次抽样。下面这个公式可以用来计算数字语音信号的比特率：

$$2 \times 4 \text{ kHz} \times 8 \text{ bits} = 64\,000 \text{ bit/s} \quad (64 \text{ kbit/s}) \text{ 或 } 1 \times \text{DS-0}$$

语音信号数字化：奈奎斯特定理

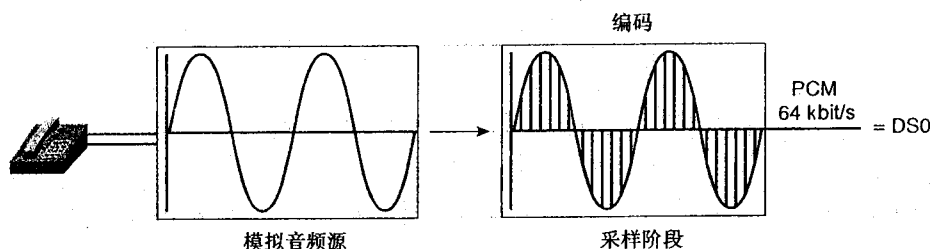


图 6-21 语音信号的数字化：奈奎斯特（Nyquist）定理

6.2.2 模拟信号到数字信号的转换过程

了解了奈奎斯特（Nyquist）定理后再简要介绍模数转换的 3 个步骤（抽样、量化和编码）以及一个可选的步骤，即压缩：

- 1 抽样——周期性地对模拟信号进行抽样，抽样的结果是脉冲幅度调制（PAM）信号。
- 2 量化——PAM 信号与一个分段的标尺进行比较。目的是测量 PAM 信号的幅度，然后分配一个整数数字来表示这一幅度。
- 3 编码——10 进制的整数值转换成一个 8 位的二进制数，其结果是一个每位为 0（无脉冲）或 1（脉冲）的 8 位二进制数字。
- 4 压缩（可选）——节省带宽。用户可以通过压缩来在一条信道传输更多的语音信息。后面将详细讲述这 4 个步骤的相关内容。

1. 抽样和量化

量化（图 6-22）将模拟抽样信号的幅度值的范围划分成一组阶跃值，使得该值与原始的模拟信号的幅度值最为接近。电压范围划分成 16 个段（正的 0 到 7 和负的 0 到 7）。从第 0

段开始，每个段都比上一段的阶跃要少，这样能够减小信噪比。如果发生了信噪比问题，可以使用对数标度将 PAM 换成 PCM 来解决。模拟信号的线性抽样会导致小信号的噪声大。 μ -定律和 A-定律是解决这一问题的两种量化方法，它们在量化时，遇到小信号会采用小的阶跃值。二者在发送信号时都对其进行了压缩，在另一端需要将信号扩展成其原来的值。这样量化的结果是使得信号幅度越小，其精确度就越高，而且在整个输入信号的范围内信噪比一致。

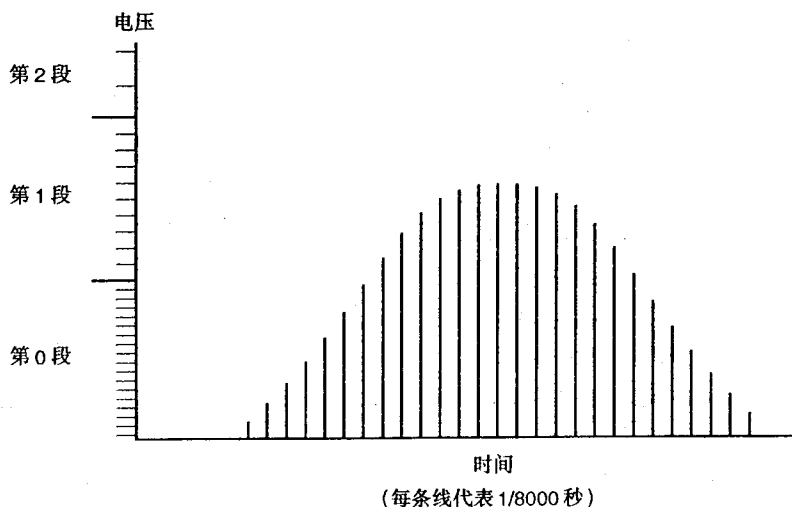


图 6-22 量化

2. 语音编码模式

这里我们要讨论 3 种语音编码模式，即波形编码器、声音合成器和混合编码器。波形编码器是对模拟信号的波形每秒进行 8000 次采样，然后确定一种最为高效的方式对其进行编码，然后发送。脉冲编码调制 (PCM)、自适应差分脉冲编码调制 (ADPCM)、 μ -定律和 A-定律都是波形编码的例子。声音合成器采用比特率低的声音合成技术来进行编码，主要是应用在军事领域。LPC、信道和相位都是声音合成器的例子。混合编码器是合成分析编码 (Abs) 方式的一部分。Abs 连续分析声音波形在下一个 5ms 的数值，因此混合编码方式所得到的结果要比简单的分析和合成要精确得多，这方面的例子包括 APC、SELP 和 CELP。

PCM (对电话信道中的语音信号每秒采样 8000 次) 是模数转换中最为常用的方法。PCM 信号送到接收器之后，接受器要将其转换恢复成模拟信号。恢复包括解调和滤波两个阶段。在解调过程中，接收到的 8 位的数字要解调成调制时定义那次抽样幅度的数字。该数字要用来重建原始幅度的 PAM 信号值。然后，PAM 信号通过一个滤波器就能够恢复原来的模拟波形信号。

3. 语音压缩技术

压缩的好处之一当然就是能够降低所需的带宽，也就能减少语音传输所需的时间和成本。尽管在高带宽的 LAN 中不需要对语音进行压缩，但是在广域网 WAN 上，对传输的语音进行压缩的好处是显而易见的。然而，压缩有可能产生失真以及被称为回音的延时。

制 (ADPCM) 就是波形压缩的一个例子。ADPCM 在将模拟语音信号编码成数字信号时，会根据前面的编码值对紧跟在后面的信号进行预测。自适应的这一部分减少了对语音信号进行编码所需的位数。波形压缩的 ITU 标准为：

G.721 等级—— $32 \text{ kbit/s} = (2 \times 4 \text{ kHz}) \times 4 \text{ bits/抽样}$

G.723 等级—— $24 \text{ kbit/s} = (2 \times 4 \text{ kHz}) \times 3 \text{ bits/抽样}$

G.726 等级—— $16 \text{ kbit/s} = (2 \times 4 \text{ kHz}) \times 2 \text{ bits/抽样}$

注释 请牢记，标准的脉冲编码调制 (PCM/G.711) 需要 64 kbit/s 。

源压缩的两个例子是低延时编码激励线性预测 (LDCELP) 方式和共扼结构代数编码激励线性预测 (CS-ACELP) 方式。CELP 是一种混合编码方式，能够通过很低的比特率传输高质量的语音信号，CELP 使用 DSP 来完成，是密集型处理器的方式。

CELP 转换模拟语音信号的过程如下：

- 1 编码器的输入信号从 8 位线性 PCM 码转换成 16 位线性 PCM 抽样码。
- 2 代码利用反馈来连续学习和预测语音的波形。
- 3 白噪声产生器触发编码器工作。
- 4 编码的数学结果送至远端解码器用于合成，恢复语音的波形。

CELP 的 ITU 标准如下：

- G.728 等级 = 16 kbit/s
- G.729 等级 = 8 kbit/s

G.729a 是一个衍生方式，它使用 8 kbit/s ，需要的处理器资源不多，允许每个 DSP 对两路语音进行编码压缩。

- G.729 就是 CS-ACELP，它是 Cisco 在其所有的语音路由器中使用的标准，适用于高质量的 8 kbit/s 环境。缺陷是每个 DSP 只能支持一路语音的编码压缩。

G.729a 虽然压缩质量不是特别高，但需要的处理器资源要少一些，而且每个 DSP 支持两路语音的编码压缩。

对上面讨论过的语音压缩技术总结如下：

- **PCM**——每秒对语音信号的幅度进行 8000 次的抽样和量化。每个抽样值用一个 8 位二进制数表示然后传出去。使用 μ -定律或 A-定律来减少噪声。技术上说，PCM 是一种编码解码技术，而不是一种压缩技术。
- **ADPCM**——这种方法中采用了抽样电流和基于上一次抽样值对下一次所作的预测值之间的差分数值。该方法以降低信号质量为代价换取了它对带宽要求的降低。抽样值可以编码成 2、3、4 或 5。
- **CELP**——在所有的该类算法中传输了一个触发值和一组线性的预测滤波结果。滤波结果值发送得没有触发值的次数多，只有在需要时才会发送。

图 6-23 对这些技术做了一个比较，它们建立和维持语音压缩所需要的带宽以及各自不同的语音质量。

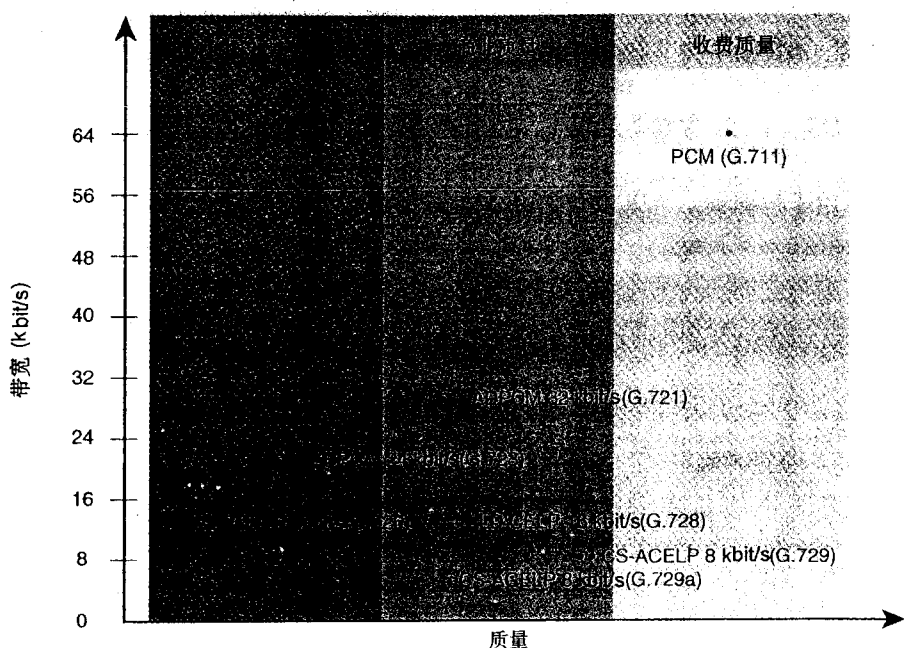


图 6-23 Cisco 的语音压缩技术

6.2.3 数字语音插入

和统计多路复用技术一样，数字语音嵌入 (DSI) 在用户的数量超过线路数量时对带宽进行多路复用。DSI 使用语音活动检测和静音抑制技术来对人语音中的无声部分进行分配，并将这些部分加以利用。请记住，人的语音中 50% 的部分都是静音。平均意见分 (MOS) 是一种评定电话语音质量的主观方法。MOS 是通过征集大量用户对于语音链路质量的意见来用统计的方法测定语音质量优劣的方法。评分等级是 1 到 5 分，5 是优秀而 1 是不满意。

6.2.4 信道信令类型和帧格式

数字服务级别 0 (DS-0) 是整个语音传输体系中最小的传输单位，它是一个 64 kbit/s 的电路。一个 DS-0 信道能够传输一个数字 PCM 语音通话。一共 24 个 DS-0 通过多路复用在一起形成一次群，叫做 DS-1。一个数字服务水平 1 (DS-1) 是一个 1.544 mbit/s 的电路，能够一次传输 24 个 8 位的 DS-0。一个 DS-1 帧有 193 位长，包括每个 DS-0 的 8 位再加上 1 个帧位。T1 的两种主要帧格式是 D4 和扩展超帧格式 (ESF)。D4 指定 12 个顺序的帧形成一个超帧。一个超帧将 A 和 B 位信令或者第 6 和 12 帧借位来作为控制信令。这些借位是 8 位中的最低位。

私用和公用网络中更为常用的是 EFS 格式。一个 EFS 由连续的 24 帧组成，包括起始位和循环冗余校验 (CRC) 位。ESF 也在帧中借用第 6、12、18 和 24 位，称为 ABCD 信令。

和 D4 一样，这些借位也是帧中的最低位。两种格式都含有基本的 192 个数据位，后面跟着

一个帧位。每个 DS1 帧的第 193 位用于同步。T1/DS-1 的欧洲格式是 E1。E1 由 32 个 64 kbit/s 的信道构成，其传输率为 2.048 Mbit/s。帧中的 30 个信道用于语音和数据。

注释 DS-1 和 T1 常容易混淆。T1 实际上指一种 1.544 mbit/s 的传输介质的物理属性。

信道 0 用于数据帧，信道 16 则用于信道信令。在 E1 帧格式中，32 个时隙组成一个帧。16 个 E1 帧组成一个多帧。

数字信道信令的两种类型是信道关联信令(CAS)和公共信道信令(CCS)。CAS(图 6-24)中，切换某个给定电路所需的信号是通过该电路自身或者通过一个关联的信道信令传送的。

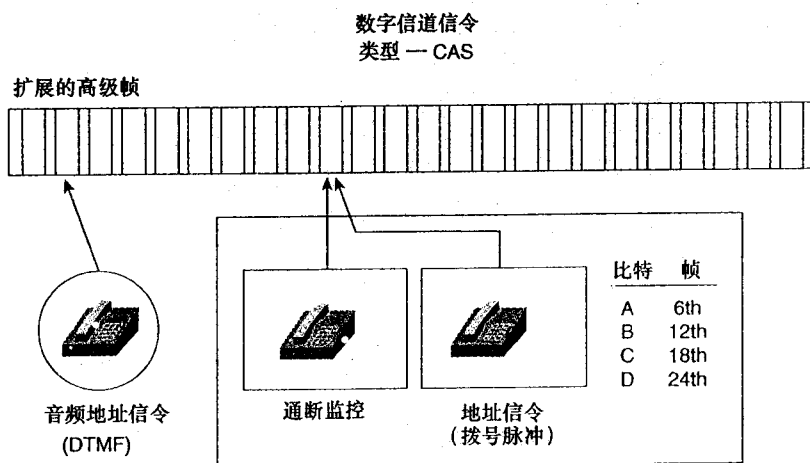


图 6-24 数字信道信令的类型——CAS

CCS(图 6-25)是链路中的一条信令信道，用于该链路中所有信道的数据控制、解释和管理。用于 CCS 的信道不会传输任何用户数据。

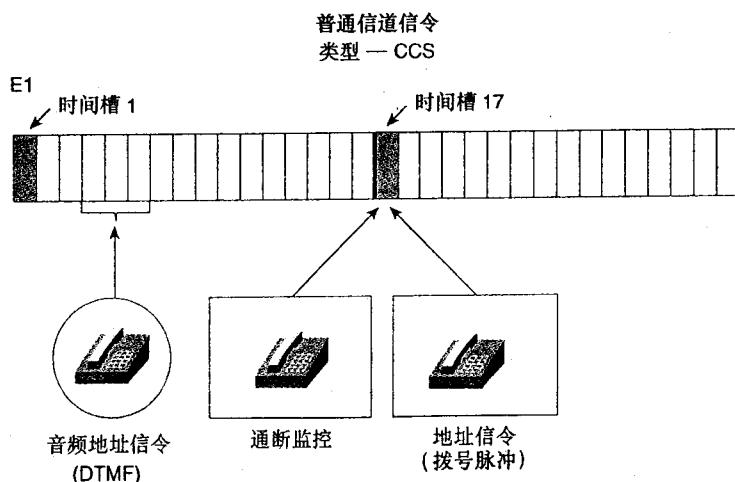


图 6-25 数字信道信令的类型——CCS

6.3 Cisco 语音产品

本节列出了 Cisco 的语音路由器，并简要介绍了其语音/数据集成功能。这里要讲述的语音路由器包括 Cisco 1750、Cisco MC3810、Cisco 2600、Cisco 3600 和 Cisco 7200 系列路由器。

Cisco 2600 和 3600 的语音网络模块支持有 3 种不同类型的 2 块语音接口卡。每一个接口都提供了一个有稍许不同的接口用于与下面这些设备互连：

- 外部交换局 (FXO) ——FXO 接口能够通过模拟连接方式与公共电话网 (PSTN) 的中心局相连。该接口在扩展用户应用中非常有价值，只需要一个语音接口连接到电话线路就可以了。该接口也可以为 PSTN 或者中央交换机的操作提供备份。
- 外部交换站 (FXS) ——FXS 接口能够为通常的电话服务机、传真机、按键设备以及 PBX 提供连接和振铃电压，拨号音等等。主要在电话直接与路由器相连接的情况下使用。
- 接收和发送 (E&M) ——E&M 接口主要连接 PBX 中继线路 (专线)。是双线和 4 线电话和中继接口的信令技术，已经成为 PBX 应用中常见的接口。

6.3.1 Cisco 1750

Cisco 1750 是一个多服务访问路由器，采用具有 QoS 的 Cisco IOS 来提供语音/传真和数据的集成功能。Cisco 1750 支持的广域网 WAN 接口与 Cisco 1600、1720、2600 和 3600 路由器相同。Cisco 1750 还和 Cisco 2600 和 3600 路由器一样支持模拟语音接口卡和 IP 语音传输技术，它支持的语音接口卡包括 FXO、FXS 和 E&M。

6.3.2 Cisco 2600

IP 传输语音的功能使得 Cisco 2600 和 3600 能够通过 IP 网络将语音信息和数据同时进行传输。IP 传输语音技术主要是一种支持语音和数据传输的软件功能。Cisco 2600 和 3600 对 ISDN BRI 信号的支持使得二者能够为 ISDN 电话网络或者 PBX/按键系统的数字接口提供语音访问的连接功能。语音或者数据也能通过 IP 网络到达与网络相连的路由器。

6.3.3 Cisco 3600

Cisco 3600 支持 IP 传输语音，帧中继传输语音以及 ATM 传输语音 (3640 和 3660)，它能让语音在现有的 WAN 基础网络上传输，包括 ISDN、租用线路、ATM 和帧中继等。另外它还有一些功能，包括对传真中继的支持以及和其他 H.323 应用的互操作性等。

6.3.4 Cisco MC3810

MC3810 主要的优点就是可以集成语音和数据。MC3810 可以通过公用或私有的帧中继，

子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责

ATM、TDM 和 VoIP 网络将 LAN、同步数据、语音、视频和传真数据集成在一起进行传输。MC3810 还通过将语音和数据在同一物理接口上进行多路复用优化了网络的带宽。MC3810 的模拟模块提供了 6 个语音接口，而数字模块则有一个数字语音访问接口（T1/E1），该接口含有 24 个压缩的语音信道。

6.3.5 Cisco 7200

Cisco 7200 系列为企业和运营商提供，有 4 插槽和 6 插槽两种主板。7200 为运营商提供了路由器管理和网络管理功能，由于具有将数据、语音和视频集成在一起的多服务功能。7200 将其多服务功能扩展到支持 VoIP 服务，包含语音处理扩展功能，如语音压缩以及 PBX 信号方式，可以为低端用户和分支机构 CPE 的集合多服务应用提供廉价的端到端解决方案。

6.3.6 Cisco 语音路由器的比较

总结一下，Cisco 能够提供高质量的多服务语音/数据集成功能的语音路由器产品包括：

- Cisco 1750 提供语音/传真/数据的集成。
- Cisco MC3810 提供端到端的多服务解决方案，包括 VoFR、VoATM 和 VoIP。
- Cisco 2600、3600 和 7200 支持 VoFR、VoATM 和 VoIP。

6.4 实验 15：通过帧中继、IP 和 ATM 传输语音

6.4.1 实验说明

语音传输技术是一门正在成长中的技术。建议读者在将任何语音网络付诸实施之前都先在实验室中建立模型进行测试，这能使读者在实际应用之前对所需要做的配置做到心中有数。实际上在这里要配置的是一种*收费旁路技术*。*收费旁路技术*也是在 IP 网络中进行一次呼叫时不会产生收费的技术。这一个实验单元里的所有实验配置的都是各种形式的*收费旁路技术*。这是引领通向多服务汇聚网络的第一步。

6.4.2 实验内容

该实验是单元形式的，包括：

- 通过帧中继传输语音。
- 通过 IP 传输语音。
- 通过 ATM 传输语音。

6.4.3 实验目的

- 在这个实验里，要通过下面这些方法来对收费旁路进行配置：
- 通过帧中继传输语音。按图 6-26 配置帧中继网络以使语音呼叫可以从一台路由器传递到下一台。
- 通过帧中继网络上的 IP 协议传输语音。如图 6-27 对网络进行配置，使得语音可以从一台路由器传输到另一台。
- 通过 ATM 应用传输语音，作为可选内容，还安排了通过专线自动振铃（PLAR）应用来传输语音的内容。按照图 6-28 配置网络使得语音可以从一台路由器传递到下一台。

这个实验单元包括 3 个部分，帧中继、IP 和 ATM。通过帧中继和通过 IP 来语音传输方式是相互依靠的。我们下面将循序介绍各部分内容。

6.4.4 所需设备

- 两台具有语音功能的 Cisco 路由器，每台都具有一个外部交换站（FXS）接口用于连接模拟电话机。实验中采用的是 Cisco 3600，当然 Cisco 2600 也完全能够胜任。
- 一台 Cisco 2600 这样的 Cisco 路由器，具有两个串行接口，用作帧中继交换机。
- 两部模拟电话机。
- 用于 ATM 部分的两台 Cisco 语音路由器，每台都有用于连接模拟电话机的 FXS 接口，而且每台路由器至少还需要一个用于点对点连接的 ATM 接口。

6.5 实验 15a：VoFR 的配置——第 1 部分

6.5.1 物理设计与实验准备

- 按照图 6-26 将一台 Cisco 语音路由器与 Cisco 2600 路由器（帧交换机）的 Serial 1/2 接口相连。
- 如图 6-26 所示，将另一台 Cisco 语音路由器与 Cisco 2600 路由器（帧交换机）的 Serial 1/3 接口相连。
- 按照图 6-26，在每台 Cisco 语音路由器的 FXS 接口上接上一部模拟电话机。

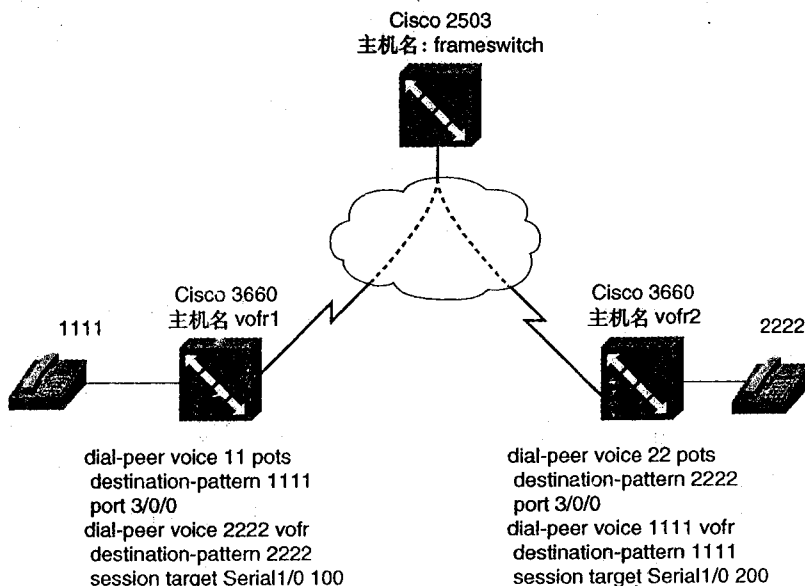


图 6-26 通过帧中继传输语音的实验

6.5.2 语音端口的配置与验证

对路由器上与电话机相连的端口进行配置可以在实验中或占用部分准备时间来完成。例 6-1 是 VoIP 和 VoFR 实验中用到的语音端口和帧中继交换机的配置范例。

例 6-1 语音端口和帧中继交换机的配置示例

```

hostname frame-switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial1/0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 64000
frame-relay intf-type dce
!
interface Serial1/1
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
!
interface Serial1/2
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
    
```

(待续)

```
!
frame-relay route 200 interface Serial1/0/200
!
interface Serial1/1/3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 100 interface Serial1/2/200
!
```

```
<<<text omitted>>>
end
```

```
frame-switch#
```

```
hostname vofr1
!
<<<text omitted>>>
!
voice-port 3/0/0
timeouts call-disconnect 0
!
voice-port 3/0/1
timeouts call-disconnect 0
!
voice-port 3/1/0
timeouts call-disconnect 0
!
voice-port 3/1/1
timeouts call-disconnect 0
!
```

```
dial-peer voice 11 pots
destination-pattern 9191
port 3/0/0
!
dial-peer voice 2222 vofr
destination-pattern 2222
session target Serial1/0/100
!
```

```
<<<text omitted>>>
```

```
!
interface Serial1/0/1
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip route-cache
frame-relay traffic-shaping
!
interface Serial1/0/1 point-to-point
ip address 150.150.10.1 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 100
class voice
vofr cisco
!
```

```
<<<text omitted>>>
```

```
!
!
map-class frame-relay vofr
frame-relay voice bandwidth 64000
!
```

(待续)

```
!
!
class frame-relay voice
frame-relay cir 768000
frame-relay bc 38400
frame-relay mincir 128000
no frame-relay adaptive-shaping
frame-relay rain-queue
frame-relay voice bandwidth 76800
frame-relay fragment 1500
!
<<<text omitted>>>
!
end

vofr1#
```

6.6 实验 15a：VoFR 的配置——第 2 部分

6.6.1 实验步骤

POTS 对等体能够使呼入业务由某个特定的电话设备来接收。配置 POTS 时，需要为其分配一个惟一的标记号以便识别，即定义它的电话号码，并将它与建立呼叫的语音端口之间建立一种连接关系。这个例子中，采用的是 4 位的拨号方式。进行 POTS 配置时，首先进入全局配置模式中，然后按照表 6-2 里的命令做相应的配置工作。

表 6-2

配置 POTS 拨号对等体的步骤

步 骤	命 令	用 途
第一步	Router#configure terminal	进入全局配置模式
第二步	Router (config) #dial-peer voice number pots	进入拨号对等体配置模式并定义用于连接 POTS 网络的本地拨号对等体 number 参数是一个或多个数字组成的对等体标识，允许的范围为 1 到 2147483647 pots 关键字用于标志对等体是用于本地电话服务
第三步	Router (config-dial-peer) # destination-pattern string[T]	配置用于呼叫远方目的对等体的电话号码串 string 参数是一串用于标识 E.164 协议或私有电话服务号码的数字。每一位允许的范围是 0 到 9 和字母 A 到 D。不可以使用加号 (+)。可以使用以下特殊字符： <ul style="list-style-type: none">• 星号 (*) 可以代表任意一个标准的拨号按键，但不可用在字符串的第一位（例如*650）。• 点号 (.) 代表一个任意合法字符• 逗号 (,) 用于前缀并插入一秒的延迟脉冲 当时间标识 (T) 字符加在目的字符串的最后时，系统将直接接受拨号数字直到时间超时（默认时间为 10 秒）或者直到用户使用中止键（默认是 # 号键）

配置 VoFR 拨号对等体时，也是为它分配一个惟一的标识，然后再定义呼出串行端口号和虚拟线路号。

根据拨号规划的不同，可能需要设置一些其他的对等体参数，如可变量拨号方式，号码扩展，超长号码的消除，转发号码以及默认语音路由，或者是搜索组的使用等等。

如果打算通过帧中继网络发送交换式呼叫，就还要配置 VoFR 来支持交换式呼叫。

要配置 VoFR 拨号对等体支持交换式呼叫的使用，表 6-3 提供了一些在全局配置模式下的命令，可作参考。

表 6-3 配置 VoFR 拨号对等体支持交换式呼叫的配置步骤

步 骤	命 令	用 途
第一步	Router (config) #dial-peer voice number vofr	定义 VoFR 拨号对等体并进入拨号对等体配置模式。你所进行的设置在退出配置模式的时候就会应用生效。 Number 参数用于标识拨号对等体，在路由器上必须是惟一的，不可重复。
第二步	Router (config-dial-peer) # destination-pattern string[T]	配置拨号对等体的目的号码。其应用限制和 VoFR 中的 POTS 拨号对等体相同。
第三步	Router (config-dial-peer) # Session target interface dlci[cid]	配置拨号对等体的帧中继转发目标方向

物理串行接口 Serial 1/0 是配置来支持帧中继封装形式的。读者可能还注意到上面过程中还对帧中继数据整形进行了配置。接口上 FRTS 功能使用的同时还启动了接口上数据整形和所有 PVC 和 SVC 的虚拟线路 (VC) 队列功能。数据整形使得路由器可以对线路上的数据输出速率加以控制，如果配置了拥塞通知功能，数据整形还能对线路拥塞通知信息做出反应。

逻辑接口 Serial 1/0.1 配置成点对点接口。也请注意这个接口上另外两个配置参数：*class voice* 和 *vofr cisco*。映射类与一个特定的数据链路连接标识 (DLCI) 之间关联关系的确立用 *class* 虚拟线路 (VC) 配置命令来实现。二者之间关系的取消则利用该命令的 *no* 形式来完成。在 Cisco 2600、3600 和 7200 路由器上，输入 *vofr cisco* 命令是配置 Cisco 专用语音封装形式的惟一方法。然后，必须配置一个映射类来启动 PVC 上的语音数据传输。

命令 *map-class frame-relay vofr* 的作用是定义一个名为 *vofr* 的映射类。要指定某个特定的 DLCI 上为语音数据留出多少带宽，可以采用 *framerelay voice bandwidth* 命令。而要释放预留出来的带宽，也可以利用该命令的 *no* 形式。

例 6-2 定义了一个名为 *voice* 的帧中继映射类。

例 6-2 帧中继映射类的定义

```
map-class frame-relay voice
frame-relay cir 768000
frame-relay bc 1000
frame-relay mincir 120000
no frame-relay adaptive-shaping
```

(待续)


```

frame-relay fair-queue
frame-relay voice bandwidth 78000
frame-relay fragment 1500

```

要指定帧中继虚拟线路上呼入、呼出业务的承诺信息速率 (CIR)，可采用 **frame-relay cir** 这个映射类配置命令。而要将 CIR 复位为默认值，可使用该命令的 **no** 形式。

要指定帧中继虚拟线路上呼入、呼出业务的承诺突发数据 (BC) 的大小，可采用 **frame-relay bc** 这个映射类配置命令。这个例子中用的是 1000 bits。要将其复位为默认值，可使用该命令的 **no** 形式。

要指定帧中继虚拟线路上能够接受的每秒呼入、呼出最少 CIR 比特，可采用 **frame-relay mincir** 这个映射类配置命令。要将其复位为默认值，可使用该命令的 **no** 形式。

要选择所使用的向后通知的类型，可采用 **frame-relay adaptive-shaping** 这个映射类配置命令。要禁止向后通知功能，则可选用该命令的 **no** 形式。

要启动一个或多个帧中继 PVC 的加权公平队列功能，可使用 **frame-relay fair-queue** 这个映射类配置命令再加上一条 **map-class framereelay** 命令。要禁止帧中继映射类的加权公平的队列功能，可使用该命令的 **no** 形式。

要指定某个特定 DLCI 上预留给语音数据的带宽，可使用 **frame-relay voice bandwidth** 命令，而要释放预留给语音数据的带宽，可使用该命令的 **no** 形式。

要对帧中继映射类的帧中继数据帧进行分解，可以使用 **frame-relay fragment** 这个映射类配置命令。而要禁止这一功能，则可以采用该命令的 **no** 形式。参数 *fragment_size* 是指定原始的数据帧分解到每个碎片中有效载荷的字节数目。这个数目不包括原始数据帧的帧中继报头在内。除了最后一个之外，帧中继数据帧所有分解后的碎片大小都要等于 *fragment_size*，最后一个碎片则应该是小于或等于 *fragment_size*。该参数的有效范围是 16 到 1600 字节，默认是 53。

在例 6-3 中，vofr2 的配置参数除了拨号对等体之外和 vofr1 几乎完全一样。目的地址当然为保留地址，而 DLCI 信息也进行了改动，以指向相应的 DLCI。

例 6-3 vofr2 上的语音端口配置

```

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname vofr2
!
enable password cisco
!
ip subnet-zero
no ip domain-lookup
!
!
!
!

```

```

voice-port 3/0/0
  timeouts call-disconnect 0
!
voice-port 3/0/1
  timeouts call-disconnect 0
!
voice-port 3/1/0
  timeouts call-disconnect 0
!
voice-port 3/1/1
  timeouts call-disconnect 0
!
dial-peer voice 22 pots
  destination-pattern 222
  port 3/0/0
!
dial-peer voice 111 voip
  destination-pattern 111
  session target Serial1/0/200
!
!
interface Ethernet0/0
  no ip address
  no ip directed-broadcast
  shutdown
!
interface TokenRing0/0
  no ip address
  no ip directed-broadcast
  shutdown
  ring-speed 16
!
interface Serial1/0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay
  frame-relay traffic-shaping
!
interface Serial1/0:1 point-to-point
  ip address 150.150.10.2 255.255.255.0
  no ip directed-broadcast
  frame-relay interface-dlci 200
  class voice
  vrf cisco
!
interface Serial1/1
  no ip address
  no ip directed-broadcast
  shutdown
!
interface Serial1/2
  no ip address
  no ip directed-broadcast
  shutdown
!
interface Serial1/3
  no ip address
  no ip directed-broadcast
  shutdown
!
interface FastEthernet2/0

```

(待续)

```
no ip address
no ip directed-broadcast
shutdown
!
router igrp 1
 network 150.150.0.0
!
ip classless
no ip http server
!
!
map-class frame-relay vofr
 frame-relay voice bandwidth 64000
!
map-class frame-relay voice
 frame-relay cir 768000
 frame-relay bc 1200
 frame-relay mincir 12000
 no frame-relay adaptive-shaping
 frame-relay fair-queue
 frame-relay voice bandwidth 76800
 frame-relay fragment 1500
!
!
line con 0
 password cisco
 transport input none
line aux 0
 password cisco
line vty 0 4
 password cisco
 login
!
end

vofr2#
```

现在，两台路由器的配置都完成了，拿起 vofr1 上的电话，拨打号码 2222。如果配置正确，而且路由器也如前所述的完成连接，这个时候就可以成功地进行通话了。挂断 vofr1 上的电话，再到 vofr2 上拨打 1111 试一下。

6.7 实验 15b：VoIP 的配置——第 1 部分

6.7.1 所需设备

VoIP 的实验需要如下设备：

- 两台具有语音功能的 Cisco 路由器，每台都具有一个外部交换工作站（FXS）接口用于连接模拟电话机。实验中采用的是 Cisco 3600，当然 Cisco 2600 也完全能够胜任。
- 一台低端 Cisco 路由器，像 Cisco 2500（具有两个串行接口的），这是用作帧中继交换机的。

6.7.2 物理设计与实验准备

- 如图 6-27 所示，将一台 Cisco 语音路由器与 Cisco 2600（帧中继交换机）上的 Serial 1/2 接口相互连接。
- 按照图 6-27，把另一台 Cisco 语音路由器与 Cisco 2600（帧中继交换机）上的 Serial 1/3 接口连在一起。
- 如图 6-27，在每台 Cisco 语音路由器的 FXS 接口上接上一部模拟电话机。

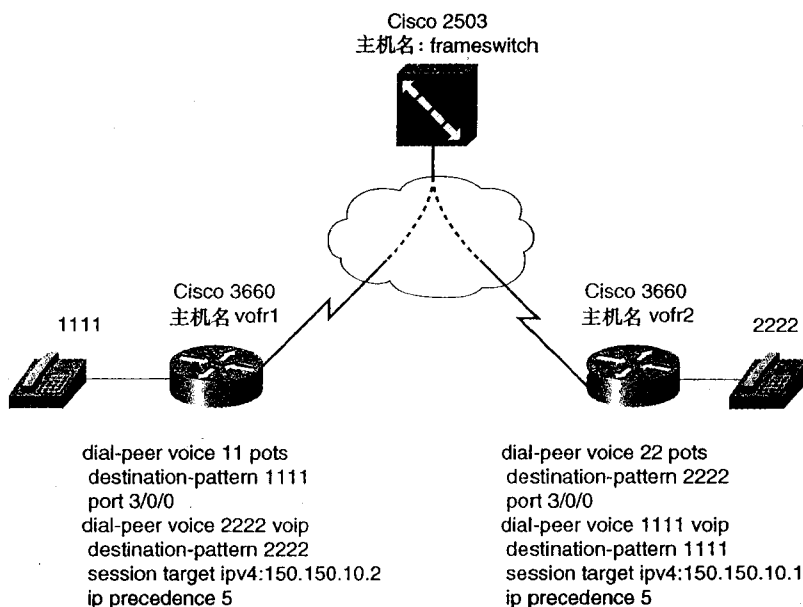


图 6-27 通过帧中继传输语音的实验

6.8 实验 15b：VoIP 的配置——第 2 部分

6.8.1 实验步骤

例 6-4 为在这个 VoIP 实验中，需要在每台路由器上所做的配置。我们将讨论关于语音端口、拨号对等体以及 QOS 机制等的配置。用于试验标识目的，分别命名这两台 Cisco 语音路由器为 voip1 和 voip2。

例 6-4 voip1 上的 VoIP 配置

```

Current configuration:
!
version 12.1
    
```

（待续）

```

service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname VoIP1
!
enable password cisco
!
!
!
!
!
!
ces 1/0
!
ip subnet-zero
no ip domain-lookup
!
lane client flush
!
!
!
!
!
controller T1 1/0
!
!
!
interface FastEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface ATM1/0
no ip address
shutdown
no atm scrambling cell-payload
no atm ilmi-keepalive
!
interface Serial2/0
no ip address
encapsulation frame-relay
no ip mroute-cache
no fair-queue
!
interface Serial2/0.1 point-to-point
ip address 150.150.10.1 255.255.255.0
no ip mroute-cache
frame-relay class voice
frame-relay interface-dlci 100
frame-relay ip rtp header-compression
!
interface Serial2/1
no ip address
shutdown

```

子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责。

址。最好是利用一个环路接口的 IP 地址来做这个地址，因为环路接口是最稳定的，也就是说，只要路由器还在通电工作，该接口就会一直处于工作状态。在这个例子中，是把会话指向远端串行子接口的 IP 地址。

IP RTP 报头压缩能够把 40 字节的 IP+UDP+RTP 报头压缩到 2 至 4 个字节，这样就大大减少了语音呼叫在点对点连接中的带宽要求。报头是在连接的一端进行压缩，而在另一端进行解压缩。这种方式的另一个标准名称是 CRTP，意思是压缩的 RTP。配置 IP RTP 报头压缩时，需要在串行接口上执行 **ip rtp header-compression** 命令，或者是在帧中继子接口上运行 **frame-relay ip rtp header-compression** 命令。

帧中继 IP RTP 优先级是指定一组属于不同 UDP 目的端口的 RTP 数据包，并以此在帧中继 PVC 上严格划分优先级队列。IP RTP 优先级建立的依据是，VoIP 数据包通过它使用的 UDP 端口范围（16384-32767）来识别。尽管实际使用的端口是终端设备或者网关之间动态地协商而定的，但是所有的 Cisco VoIP 产品采用的都是同样的端口范围。路由器检测到 VoIP 数据包之后，它会把该数据包放到一个严格的优先级队列中去。可以放到这样一个队列中去的数据包的数量是通过一个可由用户设置的系统比率来确定的。对这样一个优先级队列的服务总是会优先于任何其他用户数据，这样也就使得 VoIP 数据的抖动和延时达到了最小化。

IP 优先级机制使得语音数据包具有比其他任何 IP 数据都要高的优先级。Cisco 3600 路由器用 **ip precedence** 命令来区分语音数据和其他数据。因此，需要确定非语音数据包的优先级都要低于语音数据包的优先级。在 IP 优先性的配置中，1 到 5 的数字是区分 IP 数据流，而 6 到 7 则是用于网络与骨干路由的路由更新信息。建议把 IP 优先级 5 用于传输语音数据包。

例 6-5 是 voip2 路由器的配置示例。

例 6-5 voip2 上的 VoIP 配置

```
Current configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip2
!
enable password cisco
!
!
!
!
!
!
ces 1/0
!
ip subnet-zero
no ip domain-lookup
!
lane client flush
!
!
!
!
```

```
controller T1 1/0
!
!
!
interface FastEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface ATM1/0
no ip address
shutdown
no atm scrambling cell-payload
no atm ilmi-keepalive
!
interface Serial2/0
no ip address
encapsulation frame-relay
no ip route-cache
no fair-queue
!
interface Serial2/0:1 point-to-point
ip address 150.150.10.2 255.255.255.0
no ip route-cache
frame-relay class voice
frame-relay interface-dlci 200
frame-relay ip rtp header-compression
!
interface Serial2/1
no ip address
shutdown
!
interface Serial2/2
no ip address
shutdown
!
interface Serial2/3
no ip address
shutdown
!
router rip
network 150.150.0.0
!
ip classless
no ip http server
!
!
map-class frame-relay voice
frame-relay voice bandwidth 78000
frame-relay fragment 1500
frame-relay ip rtp priority 16384 16383 312
```

(待续)


```
no frame-relay adaptive-shaping
frame-relay cir 768000
frame-relay bc 1000
frame-relay mincir 120000
frame-relay fair-queue

!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
!

dial-peer voice 22 pots
destination-pattern 2222
port 3/1/0

dial-peer voice 1111 voip
destination-pattern 1111
session target ipv4:150.150.10.1
ip precedence 5

!
!
line con 0
password cisco
transport input none
line aux 0
password cisco
line vty 0 4
password cisco
login
!
end
```

现在，两台路由器的配置都已经完成，拿起 voip1 上的电话，拨打号码 2222。如果配置正确，而且路由器也是如前所述完成连接，这个时候就可以成功地进行一次通话了。挂断 voip1 上的电话，再到 voip2 上拨打 1111 试一下。

6.9 实验 15c：VoATM 的配置——第 1 部分

6.9.1 所需设备

通过 ATM 传输语音的实验需要如下设备：

- 两台 Cisco 语音路由器，每一台都要具有一个 FXS 接口用于连接模拟电话机，而且每台路由器还要至少有一个 ATM 接口用于点对点连接。
- 两部模拟电话机。

```
ces 1/0
!
ip subnet-zero
!
lane client flush
!
!
!
!
!
controller T1 1/0
!
!
!
interface FastEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface ATM1/0
no ip address
no atm scrambling cell-payload
no atm lmi keepalive
pvc 1/150
vbr-rt 128 64 32
encapsulation aal5mux-voice
!
!
interface Serial2/0
no ip address
shutdown
!
interface Serial2/1
no ip address
shutdown
!
interface Serial2/2
no ip address
shutdown
!
interface Serial2/3
no ip address
shutdown
!
ip classless
no ip http server
!
!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
```

```

1
dial-peer voice 1 pots
 destination-pattern 4444
 port 3/0/0
!
dial-peer voice 2 voatm
 destination-pattern 1000
 session target ATM1/0 pvc 1/150
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
!
end

```

下面是 ATM 接口上一些配置参数的介绍。

命令 **no atm scrambling cell-payload** 可以使 ATM 信元载荷的数据帧随机化，这样做可以避免出现连续不变的数据位模式，从而提高 ATM 的信元描述算法的效率。通常情况下，这条命令的默认设置就足以满足需要，无需再做设置。对 T1 或 E1 连接来说，这种数据的不规则性默认是关闭的。

如果想要启用或禁止临时本地管理接口 (ILMI) 的连接或者是改变 ILMI 的 **keepalive** 轮询的时间间隔，可以使用接口配置命令 **atm ilmi-keepalive**。禁止 ILMI 连接是使用该命令的 **no** 形式，这是默认的设置。

如果要设置 VoATM 连接的实时可变比特率 (VBR)，可以采用 ATM 虚拟线路配置命令 **vbr-rt**。它的句法格式是：**vbr-rt peak-rate average-rate burst**。

这条命令的参数包括：

- **peak-rate**——语音连接的峰值信息速率 (PIR)，单位是 kbit/s，它的取值范围是 56 到 10000。峰值 = $(2 \times \text{最大呼叫次数}) \times 16 \text{ kbit}$ 。
- **average-rate**——语音连接的平均信息速率 (AIR)，单位是 kbit/s，它的范围是 1 到 56。

平均值 = $(1 \times \text{最大呼叫次数}) \times 16 \text{ kbit}$ 。

- **burst**——信元突发比特数目，范围是 0 到 65536。

突发值 = $4 \times \text{最大呼叫次数}$ 。

命令 **encapsulation aal5mux voice** 设置 PVC 的封装形式用于支持语音数据。

命令 **pvc 1/150** (**pvc vpi/vci**) 是为语音数据创建一个 ATM PVC 并进入虚拟线路配置模式中。

POTS 拨号对等体的配置和 VoFR 以及 VoIP 实验中一样，与之相关联的是语音端口 3/0/0。

VoATM 拨号对等体不是像 VoIP 那样指向 IP 地址，它指向的是 ATM 接口以及语音 PVC。

例 6-7 是 VoATM 实验中另一台路由器的配置情况。它和前一台路由器的配置基本上一样，只是在 ATM 接口上使用了配置参数 **atm clock internal**，它指定传输的时钟是内部产生的。

例 6-7 voatm2 上的 VoATM 配置

```
Current configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voatm2
!
enable password cisco
!
!
!
!
!
!
ces 1/0
!
ip subnet-zero
!
lane client flush
!
!
!
!
!
controller T1 1/0
!
!
!
interface FastEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface ATM1/0
no ip address
atm clock INTERNAL
no atm scrambling cell-payload
no atm ilmi-keepalive
pvc 1/150
vbr-rt 128 64 32
encapsulation aal5mux voice
!
!
interface Serial2/0
no ip address
encapsulation frame-relay
shutdown
frame-relay lmi-type cisco
!
interface Serial2/1
```

(待续)

```

no ip address
shutdown
!
interface Serial2/2
no ip address
shutdown
!
interface Serial2/3
no ip address
shutdown
!
ip classless
no ip http server
!
!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
!
dial-peer voice 1 pots
destination-pattern 1000
port 3/1/0
!
dial-peer voice 2 voatm
destination-pattern 4444
session target ATM1/0 pvc 1/150
!
!
line con 0
transport input none
line aux 0
line vty 0 4
!
end

```

现在，两台路由器的配置都已完成，拿起 voatm1 上的电话，拨打号码 1000。如果配置正确，而且路由器也是像前面所说的那样连接的，这个时候就应该可以成功进行通话了。挂断 voatm1 上的电话，再到 voatm2 上拨打 1000 试一下。

6.11 实验 15d：可选实验，私有专线自动振铃

（PLAR）连接

在这个可选实验中，要建立一个私有专线自动振铃（PLAR）的连接。有时，这种方式也称为 *Bat Phone*。以前面的 VoFR 实验为参考就可以完成这个实验的配置。建立一个 PLAR 连接要做的是在语音端口上加上一条配置命令。例 6-8 是路由器 vofr2 的配置示例。

例 6-8 一个 PLAR 连接的配置示例

```
hostname v6fr2
!
enable password cisco
!
ip subnet-zero
no ip domain-lookup
!
!
!
!
voice-port 3/0/0
  timeouts call-disconnect 0
  connection plar 1111
!
voice-port 3/0/1
  timeouts call-disconnect 0
!
voice-port 3/1/0
  timeouts call-disconnect 0
!
voice-port 3/1/1
  timeouts call-disconnect 0
!
dial-peer voice 22 pots
  destination-pattern 2222
  port 3/0/0
!
dial-peer voice 1111 vofr
  destination-pattern 1111
  session target Serial1/0 200
!
!
interface Ethernet0/0
  no ip address
  no ip directed-broadcast
  shutdown
!
interface TokenRing0/0
  no ip address
  no ip directed-broadcast
  shutdown
  ring-speed 16
!
interface Serial1/0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay
  frame-relay traffic-shaping
!
interface Serial1/0.1 point-to-point
  ip address 150.150.10.2 255.255.255.0
  no ip directed-broadcast
  frame-relay interface-dlci 200
  class voice
  vofr cisco
!
interface Serial1/1
```

(待续)

```
no ip address
no ip directed-broadcast
shutdown

interface al1/2
no ip address
no ip directed-broadcast
shutdown

interface al1/3
no ip address
no ip directed-broadcast
shutdown

interface Ethernet0
no ip address
no ip directed-broadcast
shutdown

router interface
network 150.0.0

ip class
no ip header

ap-class
frame-relay
frame-relay
frame-relay
no frame-relay
frame-relay
frame-relay
frame-relay

line console
password
transport
line aux
password
line vty
password
login

end
```

与 器 vof 之后，以拿起路由器 vofr2 上的电话，它就会自动拨 1111 这个号码，的电话听起来。

第 7 章

WAN 协议与技术： 综合业务数字网 (ISDN)

Daniel keller 供稿

综合业务数字网 (ISDN) 基本上是数字化的公共交换电话网 (PSTN)，为语音、视频和数据服务的集成公共平台。在讨论 ISDN 时，一般指的是窄带综合业务数字网 (N-ISDN)，除此之外，还有一种宽带 ISDN (B-ISDN)，能够提供高速的传输功能，是 ATM 的基础。ATM 将在第 8 章 “WAN 协议与技术：异步传输模式 (ATM)” 中讨论。

7.1 ISDN 的发展、组成和结构

在公共电话网络早期，电话用户通过一系列模拟电路相连。上个世纪 60 年代早期，电话局开始用基于数字信号电路来代替网络核心部分的模拟电路。在本地环路及其他地方，用户还在使用模拟电路。ISDN 技术直接带给了用户，使他们能够通过网络传输各种类型的数据。以前，用户只能通过 ISDN 在现有的电话线上传输语音、视频及其他信息。ISDN 刚出现时，人们都认为这种技术会成为数字网络技术的主宰。现在，随着 xDSL 和电缆调制解调技术的出现，ISDN 在家庭网络市场中不再受到青睐。在商务市场上还是主力，ISDN 可以用来对主链进行备份。而且 PRI/E1 还应用在数据访问和 V.90 远程拨号方面。相比 DSL 和电缆调制解调技术来说，ISDN 的安装也

使得它还能拥有一部分家庭用户。

1984 年成立了一个标准委员会来协调 ISDN 的发展。该委员会就是国际电话电报咨询委员会 (CCITT)，现在称为国际电信联盟 (ITU)。ITU 根据 3 个应用领域来规范 ISDN 协议：

- 以字母 *E* 开头的协议是 ISDN 的电话网络标准。例如，E.164 协议描述了国际编址技术的内容。
- 以字母 *I* 开头的协议是指一些概念、术语和通用的方法。I.100 系列协议讲述了通用 ISDN 的概念和其他推荐 I 协议的结构。I.200 协议是 ISDN 服务方面的内容，而 I.300 则是网络方面的问题，I.400 是用户网络接口 (UNI) 的内容。
- 以字母 *Q* 开头的协议内容是交换和信令的工作问题。Q.921 讲述 ISDN 是如何在数据链路层处理链路访问协议 (LAPD) 的，该协议工作时近似于 OSI 模型的第 2 层，可用于 ISDN 的 D 信道的数据封装。Q.931 指定了第 3 层的功能。

对 Q.921 和 Q.931 进行 Debug 有助于解决 ISDN 的链路故障，这将在后面的内容中详细讲述。

ISDN 的主要优势在于它允许同一电路上多个数字信道同时工作。目前共有 3 种 ISDN 信道：

- **B (承载) 信道 (B [Bearer] channel)** —— 64 kbit/s，用于用户数据。一些 ISDN 交换机限制 B 信道的容量为 56 kbit/s。
- **D (数据) 信道 (D [Data] channel)** —— 16 kbit/s 或 64 kbit/s，取决于 ISDN 电路类型。该信道用于 ISDN 信令和通话建立及断开。
- **H 信道 (H channel)** —— 用于结合多个 B 信道。在北美通常不使用该信道。H 信道按如下方式使用：
 - **H0** —— 384 kbit/s (6 个 B 信道) 用于高质量的音频信号/高速的数字信息。
 - **H10** —— 1.472 Mbit/s (23 个 B 信道) 用于远程会议/数字信息。
 - **H11** —— 1.536 Mbit/s (24 个 B 信道) 用于远程会议/数字信息。
 - **H12** —— 1.92 Mbit/s (30 个 B 信道) 用于远程会议/数字信息。
 - **H4** —— 150 Mbit/s (大致带宽) 用于高清晰度电视 (HDTV)。

Cisco 支持下面 2 种 ISDN 接口，它们结合了上面的信道类型：

- **基本速率接口 (BRI)** —— 1 个 16 kbit/s 的 D 信道 + 2 个 B 信道。
- **基群速率接口 (PRI)** —— 1 个 64 kbit/s 的 D 信道 + 23 个 B 信道，用在北美和日本，总容量为 1 536 kbit/s。1 个 64 kbit/s 的 D 信道 + 30 个 B 信道，用于欧洲和其他地方，总容量为 1 984 kbit/s。

7.1.1 ISDN 组件和参考点

访问 ISDN 网络要使用本地终端设备：用户端设备 (CPE)。该设备的功能是和 ISDN 网络进行正确连接。下面是 ISDN CPE 设备类型：

- **TE1 (终端设备类型 1)** —— 具有兼容 ISDN 接口的设备。
- **TE2 (终端设备类型 2)** —— 没有兼容 ISDN 接口的设备，与 ISDN 网络连接需要使用终端适配器 (TA)。

- **NT1 (网络终端 1)** ——将 BRI 信号转换成 ISDN 的数字线路使用的信号的设备，是 ISDN 网络和 CPE 之间的界线。
- **NT2 (网络终端 2)** ——在用户处集中和切换所有的 ISDN 线路的设备。通常和专用分组交换机 (PBX) 协作。
- **TA (终端适配器)** ——用于将 EIA/TIA-232, V.35 和其他信号转换成 BRI 信号。

由于 CPE 可能包括以上一个或多个功能，因此 CPE 与其他 ISDN 设备相连的方式也有多种。所以，ISDN 标准将这些不同的接口称为参考点。参考点定义了不同 CPE 之间的逻辑点。图 7-1 列出了这些参考点，下面进行介绍：

- **R** ——非 ISDN 设备 (TE2) 与 TA 之间的连接。
- **S** ——终端用户 CPE 和 NT2 之间的连接。
- **T** ——NT2 和 NT1 之间的连接。
- **U** ——NT1 和 ISDN 网络之间的连接。

多数 BRI 用户都不需要 NT2，NT2 通常是 PBX 才使用的设备。这样，CPE 到载波之间的接口就称为 S/T 接口。通常路由器都有集成的 S/T ISDN 接口，因此参考点大都看起来像图 7-2 所示。

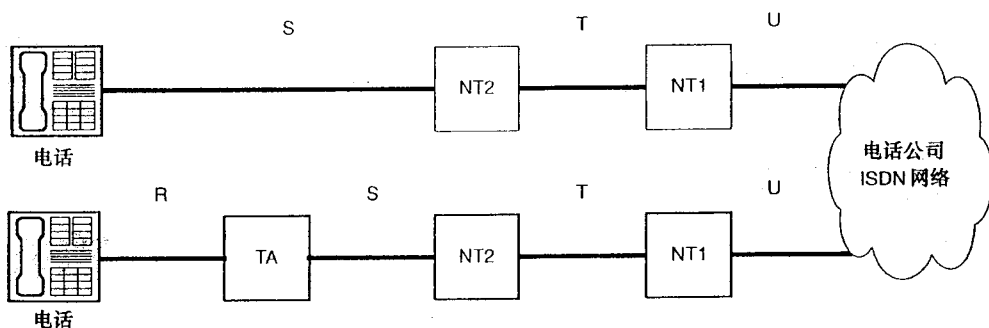


图 7-1 ISDN 参考点

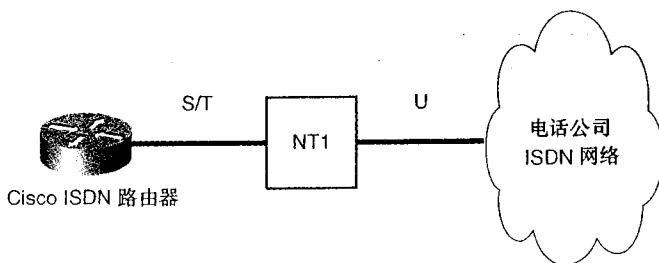


图 7-2 具有 S/T ISDN 接口的路由器的 ISDN 参考点

另外，Cisco 的路由器配有 S/T 接口和集成的 NT1，因此参考点 S、T 和 U 可被集成在 Cisco 路由器的 BRI/PRI 接口中。

在北美，所有的 ISDN 线路都通过 U 接口在本地中心局 (CO) 和 ISDN 交换机相互连接。U 接口需要网络终端将其转换成 S/T 接口以连接终端设备。NT1 能够完成该要求，并且还能作为 ISDN 线路提供电源。在欧洲，NT1 包括在电话局提供的 ISDN 线路中，因此只需 S/T 接

口即可。

使用 Cisco 路由器时，确认路由器是否具有 S/T 接口很重要。S/T 接口需要外部 NT 1，或内置 NT 1 的 U 接口。可以查看硬件手册来确定接口类型问题。

7.1.2 ISDN 分层

ISDN 工作在 OSI 模型的第 1、2 和 3 层上。只有这 3 层都正常工作，ISDN 连接才能工作。第 1 层是物理层，处理路由器与 ISDN 电路的物理连接问题。第 2 层是数据链路层，处理 D 信道上的 Q.921 协议以及每个 B 信道（HDLC 和 PPP）的数据封装问题。Q.921 协议是路由器的 ISDN 接口和 ISDN 交换机之间的信令标准。第 3 层的网络层包括 D 信道上的 Q.931 协议和 B 信道上的网络层协议（IP、IPX、AppleTalk 等）。Q.931 协议处理呼叫方和被叫方之间的呼叫建立问题。全面了解工作在 D 信道和 B 信道上的各种协议有助于配置 ISDN 以及排除 ISDN 故障。

7.1.3 ISDN 数据封装格式

ISDN 路由器支持 PPP 和 HDLC 封装。默认封装为 HDLC，但多数 ISDN 路由器都使用 PPP 封装，主要原因是 HDLC 不能同时使用两个 B 信道。在 HDLC 封装下，两条 B 信道可以同时工作，但是一条用于发送数据而另一条用于接收数据。其实每个 B 信道都具有全双工功能（能够同时收发），因此用 HDLC 将可能的数据速率减少了一半。

第 4 章“WAN 协议与技术：点对点协议（PPP）”里讲述过 PPP，该章的大部分内容对 ISDN 都适用。使用 PPP 数据封装的好处包括能够同时使用两条 BRI B 信道，能够使用 **PPP multilink** 命令以及可以进行第 4 章介绍的认证操作（PAP 或 CHAP）。

7.2 ISDN 配置基础

讲完 ISDN 枯燥的细节问题之后，现在将要开始较为有趣的话题，即如何实际配置 Cisco 路由器来使用 ISDN。但是不幸的是，在目前没有实际 ISDN 电路或是昂贵的 ISDN 仿真器的情况下，在实验室环境里实际配置 ISDN 是不可能的。然而，ISDN 是 CCIE 的路由与交换实验考试的核心部分，学习这方面知识的惟一途径就是做大量的实际练习。

配置基本的 ISDN，必须配置两个内容：

- 设定 ISDN 交换机类型。
- 服务配置文件标识符（SPID）。

服务配置文件标识符（SPID）是分配给初始化的 ISDN 终端的数字，使得存储程序控制交换系统（SPCS）能够在 D 信道的信号协议中惟一识别该 ISDN 终端。

每个 B 信道都会分配一个服务配置文件标识符（SPID），服务配置文件标识符（SPID）就像是一个电话号码，后面还跟着几个额外的位。例如，如果路由器连接到一台西门子的 ISDN 交换机，服务配置文件标识符（SPID）通常就是 aaabbbbbbbccdd 的格式，其中：

—aaa 是 3 位区号

—bbbbbbb 是 7 位的电话号码

对一些 ISDN 交换机来说，最常用的服务配置文件标识符值是 NPANXXXXXX0101，这是以现在的配置为基础的。NPA/NXX 是运营商对本地区号/访问号的称呼。

并不是所有的 ISDN 交换机都要求配置服务配置文件标识符 (SPID)，但是多数要求配置。ISDN 运营商为用户提供这方面的信息。目前，Cisco 路由器支持多种 ISDN 交换机类型，如表 7-1 所示。

表 7-1 Cisco 支持的 ISDN 交换机类型

标 识	描 述
Basic-ltr6	德国 1TR6 ISDN 交换机
Basic-5ess	AT&T 基本速率交换机
Basic-dms100	NT DMS-100 基本速率交换机
Basic-net3	NET3 ISDN 和 Euro-ISDN 交换机 (英国和其他国家)，也称为 E-DSS1 DSS1
Basic-ni11	国际 ISDN-1 交换机
Basic-nwnet3	挪威 NET3 交换机
Basic-nznet3	新西兰 NET3 交换机
Basic-ts013	澳大利亚 TS013 交换机
None	没有指定交换机
Ntt	日本 NTT ISDN 交换机 (只用于 ISDN BRI)
Primary-4ess	美国 AT&T 4ESS 交换机类型 (只用于 ISDN PRI)
Primary-5ess	美国 AT&T 5ESS 交换机类型 (只用于 ISDN PRI)
Primary-dms100	美国 NT DMS-100 交换机类型 (只用于 ISDN PRI)
Primary-net5	NET5 ISDN PRI 交换机 (欧洲)
Primary-ntt	日本 INS-NET 1500 (只用于 ISDN PRI)
Primary-ts014	澳大利亚 TS014 交换机 (只用于 ISDN PRI)
Vn2	法国 VN2 ISDN 交换机 (只用于 ISDN BRI)
Vn3	法国 VN3 ISDN 交换机 (只用于 ISDN BRI)
Vn4	法国 VN4 ISDN 交换机 (只用于 ISDN BRI)

用户可以咨询其 ISDN 运营商来确定所用的交换机类型。交换机类型可以在全局模式和接口模式下配置。如果在全局模式下配置，交换机类型适用于路由器中所有 ISDN 接口，如果是在接口模式下配置，则只适用于该接口。如果二者都进行了配置，那么，接口模式下的配置将覆盖全局模式下的配置。例如，如果用户有一台具有 6 个 ISDN 接口的路由器，其中的 5 个接口和 DMS-100 交换机相连，剩下的一个和 NI 交换机相连，可以在全局模式下使用 `isdn switch-type basic-dms100` 命令，然后在和 NI 交换机 BRI 接口相连的路由器接口的接口

模式中使用 **isdn switch-type basic-ni** 命令。

7.3 按需拨号路由 (DDR) 的配置

必须在接口上创建 **按需拨号路由 (DDR)** 功能以保证 ISDN 的正常工作。通常，ISDN 链路用于 PPP 或帧中继线路的备份。如果配置不当，ISDN 链路可能一直保持工作状态或是不停地连接，挂断，再连接。ISDN 运营商通常是按分钟收费，配置不当可能使得用户每个月 ISDN 账单超过 \$1000。DDR 能够由用户自己定义希望传输的数据。只有在用户允许或是有用户期望的数据到达 ISDN 接口的情况下 ISDN 才会连接工作。使用 **dialer idle-timeout** 命令可以实现：用户所期望的数据通过 ISDN 接口一段时间后，ISDN 停止工作。默认的超时时间是 120 秒，可以通过下面的方式设置时间值：

```
ISDN_Router (config-if) #dialer idle-timeout ?  
<1-2147483> Idle timeout before disconnecting a call
```

该时间以秒为单位。需要注意的是，当 ISDN 链路工作时，所有的数据都通过 ISDN 链路，而不仅仅是那些用户所期望的数据。

但是，只有用户所期望的数据才能使 ISDN 链路开始工作。可以把拨号空闲超时值想像为一个倒计时的计时器。当用户期望的数据使链路开始工作时，计时器也开始工作。当计时器计时到 0 时，链路断开。也只有传输用户所期望的数据（在拨号列表中定义）才能将该计时器的值复位。可见，DDR 的实现有多种方法。关键在于配置路由器，使 ISDN 链路只在需要时才会建立，不需要时断开。本章讨论的 DDR 问题是传统 DDR (legacy ddr) 和 dialer profile。传统 DDR 的功能是当有数据要发送时建立连接。这里，拨号参数在 ISDN 物理接口上指定。dialer profile 包括如何使用逻辑拨号接口来进行 DDR 通话，逻辑拨号接口和 ISDN 物理接口之间是分离的。在这一设置中，大量的配置命令都放在逻辑拨号接口上，只有一些必需的命令才放在 ISDN 物理接口上。传统 DDR 和 dialer profile 的配置实例将会在本章中进行讨论。

配置 DDR 时，要记住 ISDN 通话包含了通话双方：**主叫路由器和被叫路由器**。主叫路由器是发起建立 ISDN 通话的路由器，而被叫路由器则是 ISDN 连接对端的目的路由器。每台路由器都可以作为主叫方和被叫方，但是，在设置 DDR 时，在同一时刻只能作为拨号双方中的一方。路由器对发起呼叫可以有很多不同的要求。图 7-3 为典型的简单 ISDN 的配置简表。

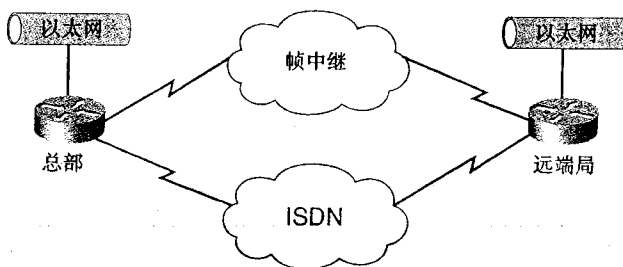


图 7-3 ISDN 作为一个主链路备份

通常，远端局初始化 ISDN 连接，成为主叫路由器，而总部路由器则成为被叫路由器。

大型的网络通常都有很多的远端局，这些远端局使用具有 BRI 接口的路由器，而总部路由器用 PRI 接口来及时处理多个 ISDN 呼叫。

该设置中，帧中继链路作为主链路，ISDN 链路作为帧中继的备份，只有在帧中继的失效时才激活 ISDN。通常远端路由器是主叫路由器，而总部路由器为被叫路由器。后面可以看到有很多方法来实现这一功能。

配置 DDR 的步骤如下：

第 1 步 如果需要，配置 ISDN 交换机类型和服务配置文件标识符信息。

第 2 步 指定启动 ISDN 呼叫的用户所期望的数据类型。

第 3 步 配置拨号信息，包括呼叫被叫路由器的号码以及任何和呼叫有关的信息。

第 4 步 配置其他与 ISDN 接口有关的可选参数，包括接口封装形式、拨号超时设置、认证选项、回拨等。本章的后面部分会讲述这方面的内容。

第 5 步 在 ISDN 线路上对数据进行路由，这里含多种选择，如指定浮动静态路由、使用备份接口命令或者使用带有各自选择参数的动态路由选择协议等。这些选择参数如下：

—— 对 OSPF：OSPF 按需电路

—— 对 EIGRP、IGRP、OSPF：拨号监控（dialer watch）

—— 对 IGRP、RIP：快照路由

—— 所有的：备份接口

本章后面部分将讲述这些路由选择协议，并通过不同的实验演示这些可选参数。本章实验和第 4 章实验有很多类似之处，但考虑 ISDN 的重要性，为求完整，还增加了一些第 4 章中没有的实验内容。

CCIE 的实验范围提供了交换机类型和服务配置文件标识符 SPID，因此不用去猜这些内容。而在实际中，由于运营商的缘故，ISDN 是一个令人头痛的问题。多数典型的非载波问题都是由不正确的输入信息（SPID 或者是交换机类型等）引起的。另外，使 ISDN 线路工作的方法就是保存路由器配置信息后再重新启动路由器。在实验考试中，如果确信路由器已经正确配置，工作线缆已经正确插到相应的接口，在遇到问题时应该尽快告诉考官线路出了问题，并向他请教该线路以前是否有过问题。向考官提问时应该将他当作客户！

7.3.1 第 1 步：ISDN 交换机类型和 SPID 信息的设置

再参考图 7-1，ISDN 交换机类型可以在全局模式和接口模式下指定。图中的路由器只有一个 BRI 接口，因此在何处指定交换机类型没有关系。为了演示需要，这里在接口模式下指定交换机类型，并且设置 SPID 信息，如例 7-1 所示：

例 7-1 在 BRI 接口设置 SPID 信息

```
ISDN_Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
ISDN_Router(config)#interface BRI0
ISDN_Router(config-if)#isdn switch-type basic-dms100
ISDN_Router(config-if)#isdn spid1 61293198331111 ldn
ISDN_Router(config-if)#isdn spid2 61293198461111 ldn
```

LDN 是本地目录号，是一个由运营商分配的用于呼叫路由的 7 位数字。建立基于 ISDN 的连接时并不需要 LDN，但是如果接收第 2 个 B 信道上的呼入业务，就需要对 LDN 进行设置。只有配置了两个 SPID（如和 DMS 或 NI1 交换机相连时）时才需要设置 LDN。每个 SPID 与一个 LDN 相关联，LDN 的设置使得第 2 个 B 信道的呼入业务能够得到正确回应。如果没有设置 LDN，第 2 个 B 信道的呼入业务就可能失败。

警告 输入 SPID 时要十分小心，因为很可能由于输入错误而出错。我们自己也常犯这种错误，浪费了大量时间查找故障原因，本以为是 ISDN 的线路故障，到最后发现是输入的 SPID 有误。

7.3.2 第 2 步：指定用户所期望的数据

用户所期望的数据定义为可以初始化 ISDN 连接的数据。连接建立后，只要定义为用户所期望的数据还在 ISDN 链路中传输，链路就会一直保持工作状态。

该步骤通过在使用接口命令 **dialer-group**，以及一个特定的定义了用户所期望数据的访问列表（称为拨号列表）来实现。**dialer-list** 的句法格式如下：

```
dialer-list dialer-group protocol protocol-name [permit|deny|list] access-list number
```

参数 **dialer-group** 用于定义接口的用户所期望的数据组号，和 **dialer-group** 命令中输入的数字一样。要确保 **dialer-group** 号和 **dialer-list** 号相匹配。例如，如果要将所有的 IP 数据指定为用户所期望的数据，可以使用 **dialer-list 1 protocol ip permit** 命令。

还可以使用扩展拨号列表来定义用户所希望的细粒度数据条件。比如，如果想指定只有源于 10.1.1.0/24 发送到目的为 10.1.2.0/24 的数据才能够初始化链路，可以通过下面的命令来完成：

```
ISDN_Router (config) #dialer-list 1 protocol ip list 101
```

```
ISDN_Router (config) #access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
```

第 1 条命令将拨号列表和扩展访问控制列表相结合。本例中使用访问控制列表 101。通过使用扩展访问控制列表可以更具体地定义用户所期望的数据类型。

7.3.3 第 3 步：拨号信息的设置

在配置拨号信息时应考虑一些选项的设置。最基本的做法是使用接口命令 **dialer-string**。这种方式配置拨号信息的缺陷是，无论实际目的地址是什么，拨号字符串都会用于所有的呼出业务。如果呼入路由器只是和另外一台路由器互连，并没有什么问题，但是如果 ISDN 路由器需要拨入多个 ISDN 路由器就会有问题。另外一个功能更强大的完成这一步任务的办法是使用 **dialer-map** 命令。这样，路由器能够根据第 3 层地址更精确地拨入指定 ISDN 路由器。这两个用法通常称为传统 DDR。最后一个选项使用一组逻辑拨号接口。使用逻辑拨号接口的主要好处是 ISDN 路由器能够同时拨入多个 ISDN 路由器。图 7-4 中用简化的网络模型讲解了这 3 种选项。



图 7-1 基本的 ISDN 网络

1. 配置拨号信息，方法 1：使用拨号字符串

这是配置 ISDN 最简单（最不健全）的方法。例 7-2 中，定义为用户所期望的数据建立连接，开始呼叫业务，所有的数据将发送到拨号字符串指定的号码上。这种方式在只有一个目的路由器的情况下能够很好地工作，但在有多个需要相互直接访问的 ISDN 站点的情况下则不是很理想。本例中的 BRI 接口封装已经设置成 PPP。这种情况下，不需要做更多的配置工作，和默认封装形式 HDLC 的情况类似。

例 7-2 使用拨号字符串时，路由器 Cheech 和 Chong 的配置

```

Cheech#show running-config

version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Cheech
!
enable password cisco
!
isdn switch-type basic-dms100
!
interface Ethernet0
no ip address
shutdown
!
interface Serial0
no ip address
shutdown
!
interface Serial1
no ip address
shutdown
!
interface BRI0
ip address 175.10.23.1 255.255.255.252
encapsulation ppp
isdn spid1 61293199371111 <SPID associated with the first B channel
isdn spid2 61293199381111 <SPID associated with the second B channel
dialer string 6129319833 <number to dial for all outgoing connections
dialer-group 1
!
ip classless
!
dialer-list 1 protocol ip permit <all ip traffic is interesting
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
  
```

（待续）


```
line aux 0
line vty 0 4
  password cisco
  login
!
end

Chong#show running-config
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Chong
!
enable password cisco
!
isdn switch-type basic-dms100
!
interface Ethernet0
no ip address
shutdown
!
interface Serial0
no ip address
shutdown

!
interface Serial1
no ip address
shutdown
!
interface BRI0
ip address 175.10.23.2 255.255.255.252
encapsulation ppp
isdn spid1 61293198331111 <SPID for the first B channel
isdn spid2 61293198461111 <SPID for the second B channel
dialer string 6129319937 <number to call for all connections
dialer-group 1
!
no ip classless
!
dialer-list 1 protocol ip permit <all ip traffic is interesting
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
line aux 0
line vty 0 4
no login
privilege level 15
!
end
```

本例中，两台路由器都可以触发 ISDN 工作。如果只想要 Cheech 呼入 Chong，但是 Chong 不能呼入 Cheech，可以在 Chong 的配置中去掉 **dialer string** 命令。

注释 尽管例 7-2 所示的配置过程简单明了，但大家可能对所用的命令并不十分熟悉。两台路由器都有一些故意设置的严重的安全漏洞。**line con 0** 下 **privilege level 15** 命令，使得与路由器相连的用户可以不需要密码直接进入 enable 模式。**exec-timeout 0 0** 命令，使得所有的控制台连接永远不会超时。这样可以减少进入工作模式的时间和输入密码的麻烦。使用 Telnet 时，在 VTY 下作类似的设置可以直接进入工作模式而不必输入密码。尽管在实验环境中有用，但是尽量不要在实际的路由器中使用这些命令。还有一个值得注意的是控制台配置下面的 **logging synchronous** 命令。该命令能够防止当用户正在输入命令时路由器打扰用户的工作。默认情况下，路由器会将所有的事件记录报告给控制台，这样使得用户在键入命令时很容易被打扰。解决该问题的一个可选方法是用 **no logging console** 命令，但是我们更倾向于使用 **logging synchronous** 命令，因为这条命令可以使用户看到有用的控制台记录项，而同时避免默认情况下出现干扰。

2. 配置拨号信息，方法 2：使用拨号映射

多个 ISDN 连接时拨号字符串会产生问题。图 7-5 所示的网络中，总部路由器需要两组拨号字符串来与远端路由器 Cheech 和 Chong 相连接。但是，总部路由器没有办法将 IP 地址和拨号字符串相关联，如果和路由器 Chong 相连，有可能拨入到路由器 Cheech。

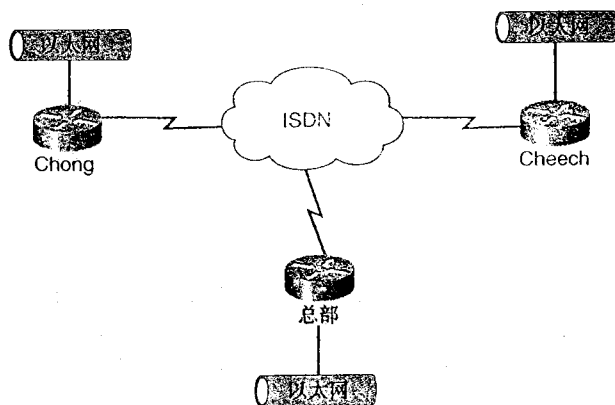


图 7-5 拨号映射配置

在 IP 地址和所拨号码之间建立映射的方法之一是使用拨号映射。拨号映射在路由器连接某个 IP 地址时，告诉路由器应该呼叫的拨号字符串。例 7-3 为使用拨号映射的路由配置示意图。

例 7-3 总部路由器的拨号字符串的配置实例

```
interface BRI0
ip address 175.10.23.3 255.255.255.248
encapsulation ppp
isdn spid1 61293193601111
isdn spid2 61293197761111
dialer map ip 175.10.23.1 name Cheech 6129319937
dialer map ip 175.10.23.2 name Chong 6129319833
dialer-group 1
```

一子网中。

使用拨号映射时还需注意一些问题。一个是通过关键字 **speed** 手动设置每个 B 信道的连接速率为 56 kbit/s 或 64 kbit/s。另一有用的命令是关键字 **name**，它可以放在拨号映射声明的结尾。在命令 **dialer map** 中加入 **name** 关键字和 CHAP 结合，电话号码和给定下一跳地址的路由器名称都会和已经连接的路由器进行比较。这样可以避免再次呼叫已经建立连接的远端路由器。

关键字 **broadcast** 非常重要。如果用户想通过 ISDN 链路传输路由更新信息（路由选择协议使用广播数据包宣告其路由信息），必须在拨号映射语句中加入 **broadcast** 关键字，如例 7-4 所示。这和帧中继很相似，在帧中继中，如果想要传输路由更新信息，就必须在所有的 **frame-relay map** 命令中加入 **broadcast** 参数。

例 7-4 广播应用：总部路由器的配置

```
interface BRI0
 ip address 175.10.23.3 255.255.255.248
 encapsulation ppp
 isdn spid1 61293193601111
 isdn spid2 61293197761111
 dialer map ip 175.10.23.1 name Cheech broadcast 6129319937
 dialer map ip 175.10.23.2 name Chong broadcast 6129319833
 dialer-group 1
```

例 7-4 中的配置使得路由更新信息和其他广播数据能够在网络中传输。但是，如果不想让这些数据真正触发呼叫，可以指定限制性的拨号列表或者是使 ISDN 接口在路由配置中处于 **passive** 状态。

3. 配置拨号信息，方法 3：使用逻辑拨号接口

Cisco IOS 11.2 中引入了 **dialer profile**，提供了将传统 DDR 中使用的配置参数和物理接口相互分离的功能。这可能是最复杂的配置 DDR 的方法，但是提供了最大的设计灵活性，通过创建用于存放配置选项的逻辑拨号接口工作。只有建立连接之后，拨号接口才能通过拨号池和物理接口相关联。

要配置 **dialer profile**，首先必须删除 BRI 0 接口下所有传统 DDR 命令。然后创建逻辑接口，**dialer 1**，将所有的配置选项和第 3 层信息放到该接口。配置 **dialer profile** 的步骤如下：

- 第 1 步 删除 BRI 0 接口下所有的传统 DDR 命令。这样才能使 IOS 允许将拨号接口映射到物理接口。
- 第 2 步 用 **interface dialer X** 命令对逻辑接口进行配置，这里的 **X** 值可以是 0 到 255。
- 第 3 步 在拨号接口上配置 **dialer remote-name**。只能指定一个远端设备的名称，即希望与其建立 ISDN 链路的路由器名称。
- 第 4 步 建立逻辑接口到物理接口的映射，这通过将逻辑接口上的拨号池与物理接口上的拨号池成员相关联来实现。所用的命令包括物理接口上的 **dialer pool-member x** 命令和逻辑拨号接口上的 **dialer pool x** 命令，**x** 是 1 到 255 之间的一个整数。两个接口上的数应该一致，而且每个拨号接口只能有一个拨号池。
- 第 5 步 定义用户所期望的数据。这一步仅在需要启动呼叫的路由器（呼叫路由器）口才需要，方法和传统 DDR 类似，但命令是在逻辑接口上设置的。

第6步 在逻辑接口上为呼叫提供拨号字符串，该号码用于呼叫，并与被叫 ISDN 路由器建立连接。注意，拨号映射不能在拨号接口上配置。和另一台路由器建立连接后，拨号映射动态产生。

第7步 在拨号接口上设置可选参数，包括拨号空闲超时时间，网络寻址方式等。

例 7-5 给出了包含这 7 个步骤的配置实例，注意该例中引入了逻辑拨号接口。

例 7-5 dialer profile 的配置

```
Remote_Site#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Remote_Site
!
!
username Headquarters password 0 cisco
ip subnet-zero
isdn switch-type basic-dms100
!
interface Loopback0
ip address 175.10.101.1 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0
ip address 175.10.2.1 255.255.255.0
no ip directed-broadcast
no keepalive
!
interface Serial0
no ip address
no ip directed-broadcast
shutdown
no fair-queue
clockrate 125000
!
interface Serial1
description PRIMARY LINK TO HQ
bandwidth 64
ip address 175.10.200.1 255.255.255.252
no ip directed-broadcast
!
interface BRI0
no ip address
no ip directed-broadcast
encapsulation ppp
dialer pool-member 1
isdn switch-type basic-dms100
isdn spid1 61293199371111
isdn spid2 61293199381111
!
interface Dialer1
ip address 175.10.23.1 255.255.255.252
```

（待续）

```

no ip directed-broadcast
encapsulation ppp
dialer remote-name Headquarters
dialer string 6129319833
dialer pool 1
dialer-group 1
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
  privilege level 15
  logging synchronous
  transport input none
line aux 0
line vty 0 4
  login
!
end

! We can verify our configuration by using the show dialer command. Notice here
that the BRI 0 interface is now bound to the logical dialer interface, dialer 1.
Remote_Site#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=175.10.23.1, d=175.10.23.2)
Interface bound to profile Di1
Time until disconnect 111 secs
Current call connected 00:00:10
Connected to 6129319833 (6129319833)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

Di1 - dialer type = DIALER PROFILE
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up

Dial String      Successes  Failures  Last called  Last status
6129319833       2          0        00:00:11    successful  Default

```

7.3.4 第4步：配置高级可选参数

ISDN 能够提供很多高级特性，如 ISDN 回拨、呼叫筛选和 PPP 认证等，这些选项大部分只适用于 PPP 封装的接口。可以参考例 7-6 中 BRI 接口的 PPP 选项的例子。

例 7-6 可用的 PPP 选项

R2(config-if)#ppp ?	
authentication	Set PPP link authentication method
bap	Set BAP bandwidth allocation parameters
bridge	Enable PPP bridge translation
callbac k	Set PPP link callback option
chap	Set CHAP authentication parameters
compression	Enable PPP Compression control negotiation
ipcp	Set IPCP negotiation options
lcp	PPP LCP configuration
max-bad-auth	Allow multiple authentication failures
multilink	Make interface multilink capable
pap	Set PAP authentication parameters
quality	Set minimum Link Quality before link is down
reliable-link	Use LAPB with PPP to provide a reliable link
timeout	Set PPP timeout parameters
use-tacacs	Use TACACS to verify PPP authentications

例如，如果通过认证来确认一个 ISDN 呼入业务的合法性，必须使用 PPP 封装。PPP 也可以用来设置拨号回拨和多链路连接。由于这些功能都必须使用 PPP，而且几乎所有的实际 ISDN 接口配置都使用 PPP 封装，因此所有实验中也使用 PPP 封装的 BRI 接口。

最常用的配置选项是认证，包括下面将要讨论的 PAP 或 CHAP。

1. 例 1：使用密码认证协议（PAP）进行认证

PAP 是远端路由器用来验证远程节点合法性的简单办法。主要缺点是安全性不高，这是因为 PAP 中用户名和密码在 ISDN 网络中以明文方式传输，可以用一台协议分析仪看到相关信息。由于 PAP 在链路传输过程中没有加密，因此任何人都可以用协议分析仪轻易地得到这些信息。PAP 对那些回放和反复尝试法（词典轮询或者是暴力破解法）的攻击方式无能为力。因此，CHAP 是首选的认证方法。Cisco 支持 CHAP 认证。

配置 PAP 的 4 个步骤如下：

- 第 1 步** 全局配置模式下用 **username name password password** 命令设置远端路由器的用户名和密码。用户名必须和远端路由器的主机名完全匹配，用户名区分大小写。
- 第 2 步** 接口配置模式下，用 **encapsulation ppp** 命令设置 ISDN 接口的 PPP 封装形式。
- 第 3 步** 用 **ppp authentication pap** 命令指定使用 PAP 认证方式。
- 第 4 步** 用 **ppp pap sent-username username password password** 命令指定要发送到远端路由器的本地用户名和密码。这些信息要和第 1 步中远端路由器为本地路由器指定的信息相匹配。

例 7-7 是呼叫路由器的 PAP 认证配置，而例 7-8 则是被叫方的 PAP 认证配置情况。

例 7-7 配置 PAP 认证：路由器 1（呼叫路由器）

```
ISDN-1#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
```

（待续）

```

service timestamps log uptime
no service password-encryption
!
hostname ISDN-1
!
!
username ISDN-2 password 0 CISCO
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Loopback0
ip address 175.10.101.1 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0
ip address 175.10.2.1 255.255.255.0
no ip directed-broadcast
no keepalive
!
interface Serial0
no ip address
no ip directed-broadcast
encapsulation frame-relay
logging event subif-link-status
logging event dlci-status-change
no fair-queue
clockrate 125000
!
interface Serial1
description PRIMARY LINK TO HQ
bandwidth 64
ip address 175.10.200.1 255.255.255.252
no ip directed-broadcast
!
interface BRI0
ip address 175.10.23.1 255.255.255.252
no ip directed-broadcast
encapsulation ppp
dialer map ip 175.10.23.2 name ISDN-2 broadcast 6129319033
dialer-group 1
isdn switch-type basic-dms100
isdn spid1 61293199371111
isdn spid2 61293199381111
ppp authentication pap
ppp pap sent-username ISDN-1 password 7 0802657D2A36
!
router eigrp 1
passive-interface BRI0
network 175.10.0.0
no auto-summary
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
privilege level 15
    
```

(待续)

```
logging synchronous
transport input none
line aux 0
line vty 0 4
  login
!
end
```

例 7-8 配置 PAP 认证：路由器 2（被叫路由器）

```
ISDN-2#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ISDN-2
!
logging buffered 9096 debugging
!
username ISDN-1 password 0 CISCO
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Loopback0
 ip address 175.10.102.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 ip address 175.10.1.1 255.255.255.0
 no ip directed-broadcast
 no keepalive
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 logging event subif-link-status
 logging event dlci-status-change
!
interface Serial1
 description PRIMARY LINK TO REMOTE SITE
 bandwidth 64
 ip address 175.10.200.2 255.255.255.252
 no ip directed-broadcast
 clockrate 125000
!
interface BRI0
 ip address 175.10.23.2 255.255.255.252
 no ip directed-broadcast
 encapsulation ppp
 dialer-group 1
 isdn switch-type basic-dms100
```

（待续）


```

isdn spid1 61293198331111
isdn spid2 61293198461111
ppp authentication pap
ppp pap sent-username ISDN-2 password 7 01302F377824
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
  privilege level 15
  logging synchronous
  transport input none
line aux 0
line vty 0 4
  login
!
end

```

在该配置中，BRI0 接口下的密码经过加密。这由 Cisco IOS 实现，目的为了防止简单查看屏幕上的配置信息就可获取密码。该密码只在配置文件中加密，而实际的密码（CISCO）在链路中则以明文形式传输。

2. 例 2: 用质询应答认证协议 (CHAP) 进行认证

几乎所有实际的 PPP 认证方案都采用 CHAP 方式，这是因为 CHAP 的安全性比 PAP 更高。CHAP 能周期性地验证远端节点的身份，使用 MD5 散列算法对 CHAP 过程进行加密。因此，即使在线路中放置了协议分析仪 (sniffer) 的黑客也不能破解加密的密码。CHAP 采用了下面的 3 次握手过程进行认证：

- 1 中心路由器向远端路由器发送一个质询信号。
- 2 远端路由器响应该响应信号。
- 3 中心路由器接受或拒绝连接。

PPP 的 CHAP 的配置只需要下面的 3 个步骤：

第 1 步 在 ISDN 接口上将封装类型设置为 PPP。

第 2 步 在接口配置模式下使用 **ppp authentication chap** 命令对 CHAP 进行配置。

第 3 步 在全局配置模式下用命令 **username name password secret** 设置远端 ISDN 路由器的用户名和密码，这里 *name* 指远端路由器的主机名而 *secret* 则是特权密码。*name* 区分大小写，必须与远端路由器的主机名完全匹配。两台路由器上的密码也必须完全匹配。如果 CHAP 失败，首先要做的就是检查用户名是否正确以及密码是否匹配。

例 7-9 显示了路由器 Router 1 的 PPP CHAP 配置，例 7-10 为 Router 2 的 PPP CHAP 配置。

例 7-9 PPP CHAP 的配置实例: Router 1

```

ISDN-1#show running-config
!

```

```

version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ISDN-1
!
!
username ISDN-2 password cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
 ip address 175.10.2.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 no ip directed-broadcast
 shutdown
!
interface BRI0
 ip address 175.10.23.1 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer map ip 175.10.23.2 name ISDN-2 broadcast 6129319833
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 ppp authentication chap
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

```

例 7-10 PPP CHAP 的配置实例: Router 2

```

ISDN-2#show running-config
!
version 12.0
service timestamps debug uptime

```

(待续)

```

service timestamps log uptime
no service password-encryption
!
hostname ISDN-2
!
!
username ISDN-1 password cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Serial1
 no ip address
 no ip directed-broadcast
 shutdown
!
interface BRI0
 ip address 175.10.23.2 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer map ip 175.10.23.1 name ISDN-1 broadcast 6129319937
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293198331111
 isdn spid2 61293198461111
 ppp authentication chap
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

```

例子中突出显示的部分表示 CHAP 认证的重要配置命令。

3. 例 3: 使用可替换的主机名进行认证

Cisco IOS 允许使用除呼叫路由器的主机名之外的其他所有主机名称配置 CHAP。

当与远端 cisco 路由器相连的其他 Cisco 路由器或非 cisco 中心路由器从属于不同的管理控制，或不同的 ISP，或者需要连接不同路由器的时候，有必要为其设置一个不同于本身的真实主机名的认证主机名。这种情况下，路由器的主机名在不同时段是变化的。而且，ISP

分配的主机名和密码也可能不是远端路由器的主机名。这时，可以用 **ppp chap hostname** 命令来指定一个用于认证的替换主机名。

以有多个远端设备拨入中心站点的情况为例。通常 CHAP 情况下，每台远端设备的用户名（也就是主机名）以及共享的密码都必须在中心路由器上进行设置。这样的情况下，中心路由器的配置管理会变得非常烦琐。但是，如果远端设备使用与它们的主机名不同的用户名，就可以避免这样的情况。中心站点可以只设置一个用户名和一个共享的密码就能够对多个拨入的用户进行认证。

上述的过程可以用命令 **ppp chap hostname “alternate-host-name”** 来实现：

ISDN-1 (config-if) #**ppp chap hostname ?**

WORD Alternate CHAP hostname

在被叫路由器上，可以用呼叫路由器的替换主机名而不是真实主机名来对其用户名/密码进行设置。例 7-11 给出了呼叫路由器的配置实例，例 7-12 为被叫路由器的配置实例。

例 7-11 CHAP 替换主机名配置：呼叫路由器

```
ISDN-1#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ISDN-1
!
username ISDN-2 password cisco
ip subnet-zero
isdn switch-type basic-dms100
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 no ip directed-broadcast
 shutdown
!
interface BRI0
 ip address 175.10.23.1 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer map ip 175.10.23.2 name ISDN-2 broadcast 6129319833
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
```

(待续)

```

ppp authentication chap
ppp chap hostname FAKE-NAME
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
  privilege level 15
  logging synchronous
  transport input none
line aux 0
line vty 0 4
  login
!
end

```

例 7-12 CHAP 替换主机名配置：被叫路由器

```

ISDN-2#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ISDN-2
!
!
username FAKE-NAME password cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
  ip address 172.16.1.1 255.255.255.0
  no ip directed-broadcast
!
interface Serial0
  no ip address
  no ip directed-broadcast
  shutdown
!
interface Serial1
  no ip address
  no ip directed-broadcast
  shutdown
!
interface BRI0
  ip address 175.10.23.2 255.255.255.248
  no ip directed-broadcast
  encapsulation ppp
  dialer map ip 175.10.23.1 name ISDN-1 broadcast 6129319937

```

(待续)

```

dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 6129319831111
 isdn spid2 61293198461111
 ppp authentication chap
 !
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

```

上例中，被叫路由器用主机名 FAKENAME 来对呼叫路由器进行认证，而呼叫路由器的实际主机名是 ISDN-1。用 **debug ppp authentication** 命令可以看到这一过程，如例 7-13 所示。

例 7-13 呼叫路由器的认证过程验证

```

ISDN-1#debug ppp authentication
PPP authentication debugging is on
ISDN-1#ping 175.10.23.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.23.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 40/41/44 ms
ISDN-1#
08:38:29: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
08:38:29: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833
08:38:29: BR0:1 PPP: Treating connection as a callout
08:38:30: BR0:1 PPP: Phase is AUTHENTICATING, by both
08:38:30: BR0:1 CHAP: Using alternate hostname FAKE_NAME
08:38:30: BR0:1 CHAP: O CHALLENGE id 8 len 30 from "FAKE_NAME"
08:38:30: BR0:1 CHAP: I CHALLENGE id 8 len 27 from "ISDN-2"
08:38:30: BR0:1 CHAP: Using alternate hostname FAKE_NAME
08:38:30: BR0:1 CHAP: O RESPONSE id 8 len 30 from "FAKE_NAME"
08:38:30: BR0:1 CHAP: I SUCCESS id 8 len 4
08:38:30: BR0:1 CHAP: I RESPONSE id 8 len 27 from "ISDN-2"
08:38:30: BR0:1 CHAP: O SUCCESS id 8 len 4
08:38:31: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state
to up
ISDN-1#

```

请注意例中第 1 个 ICMP 的 ping 数据包超时，这是由 ISDN 呼叫认证和连接建立过程所花的时间引起的。

4. 例 4：只对呼入业务进行 PPP 认证——单向 CHAP 认证的配置

两台设备都使用普通 CHAP 认证方式时，每一方都会发出质询信号，然后等待对方作出

应答，对方同时也发出质询信号。双方都会独立地对对方信息进行认证。如果需要呼叫不支持认证功能的非 Cisco 路由器或设备，就必须用 **ppp authentication chap callin** 命令来解决。在使用带有 **callin** 关键字的 **ppp authentication** 命令时，访问服务器只有在远端设备呼叫时（例如，远端设备“呼入”）才会对远端设备进行认证。这种情况下，认证只针对呼入（接收到的）的业务。

以表 7-2 为例，如果 Router 1 呼叫 Router 2，Router 2 会对 Router 1 进行认证，但是 Router 1 不会对 Router 2 进行认证。这是因为 Router 1 上设置了 **ppp authentication chap callin**。这就是单向认证的例子。

表 7-2

单向认证

Router 1	Router 2
Router 1 calls Router 2 --->	
<--- Router 2 challenges Router 1	
Router 1 does not challenge Router 2 because it is configured for one-way authentication.	

5. 例 5: 使用 PPP 链路质量

第 4 章中已讲过 PPP 链路质量。有一点很重要，就是只要接口是配置成 PPP 封装，该特性也适用于 ISDN 接口。

6. 补充的可选参数：链路质量监控 (LQM)

链路质量监控 (LQM) 适用于所有运行 PPP 的串行接口，它监控链路质量。如果链路质量低于某个设置的百分比标准，路由器会关闭链路。这个百分比标准的计算既包括呼入方向，也包括呼出方向。输出质量是将所有发送总的数据包和字节比目的节点接收到的总的数据包和字节来计算的。输入质量是将所有接收到的数据包和字节比目的节点发送出来的所有的数据包和字节来计算的。

启动 LQM 后，以前的 **keepalive** 信号就由**链路质量报告 (LQR)** 取代，以 **keepalive** 为周期发送 LQR。所有呼入的 **keepalive** 都能得到正常的应答。如果没有设置 LQM，以 **keepalive** 为周期发送 **keepalive** 信号，所有的呼入 LQR 都会以一个 LQR 来应答。LQR 是 1989 年由 Computer Systems Consulting Service 的 William A. Simpson 在 RFC 的 1989 “PPP 链路质量监控”中制定的。可以在接口配置模式下使用 **ppp quality percentage** 命令在接口上使用 LQM。

变量 **percentage** 指定了链路质量的阈值。必须维持该链路质量标准的值，否则链路质量就会下降而导致链路关断。该值既针对呼入业务也针对呼出业务。输出质量是将所有发送出去的总的数据包和字节比上链路对端所有接收到的总的数据包和字节来计算的。输入质量是将所有接收到的数据包和字节比上链路对端发送出来的所有的数据包和字节来计算的。

如果链路质量的百分比标准无法维持，表明链路质量很差，会使链路关闭。规范中加上了一个滞后的时间，以避免链路在连通和断开之间来回动荡。

7. 附加参数：ISDN 呼叫筛选

除了 CHAP 和 PAP 认证外，还有其他办法对 ISDN 呼入进行认证。基于呼叫者身份 ID

的认证能比前面讨论的方法提供更高的安全性，它对远端用户的认证不仅基于其用户名和密码，而且还有包括呼叫的地点。呼叫者身份认证方式通过 ISDN 的设置信息中的呼叫者身份信息来判断接受还是拒绝用户到服务器的呼叫。但并不是任何地方都可以获得呼叫者身份 ID，因此，如果想在网络中加入这项功能，一定要咨询 ISDN 运营商。

配置 ISDN 呼叫筛选只需要一条命令，对于传统 DDR，可以用：

```
isdn caller remote-number [callback]
```

对于 dialer profile 来说，则可以用：

```
dialer-caller remote-number [callback]
```

选项 **callback** 使路由器中止当前呼入，而去回拨呼叫的路由器。下一节将会讲述回拨的内容。例 7-14 是如何配置呼叫筛选的例子。

例 7-14 呼叫筛选

```
interface BRI 0
isdn caller 6129319937
```

对于每个呼入的号码该命令都可以重复使用。Cisco IOS 还允许在远端路由器上用字母 *x* 设置“任意值”位。例如，允许从区号 952 处呼入所有业务，可以使用命令 **isdn caller 952xxxxxxx** 来实现。每一个可以设为任意值的位都必须用一个 *x* 来表示。命令 **isdn caller 952x** 和命令 **isdn caller 952xxxxx** 不一样。第 1 条命令只允许 952 开头的 4 位数，而第 2 条命令则允许 952 开头的 7 位数。

debug ISDN 可以用于排除 ISDN 呼叫筛选的故障。

8. 附加参数：ISDN 回拨

回拨对于在中心位置控制所有的呼出 ISDN 业务费用，或其他商业目的来说非常有用。回拨功能（Cisco IOS 11.0 引入的功能）允许远端路由器呼叫中央站点，请求中央站点回拨远端路由器。然后，中央站点断开连接，再呼叫远端路由器用户。配置回拨功能可以减少 ISDN 的远端用户的费用，这是因为真正的数据传输是在中央站点回拨远端用户之后发生的。例 7-15 和 7-16 分别例示了远端路由器和中央路由器 ISDN 的回拨配置。

例 7-15 配置 ISDN 回拨：远端路由器

```
Remote_Site#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Remote_Site
!
!
username Headquarters password 0 cisco
ip subnet-zero
isdn switch-type basic-dms100
```

（待续）


```

!
!
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 ip address 175.10.50.1 255.255.255.252
 no ip directed-broadcast
 no fair-queue
 clockrate 125000
!
interface Serial1
 ip address 175.10.1.1 255.255.255.252
 no ip directed-broadcast
!
interface BRI0
 ip address 175.10.23.1 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer wait-for-carrier time 10
 dialer map ip 175.10.23.2 name Headquarters broadcast 6129319833
 dialer hold-queue 100 timeout 10
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 ppp callback request
 ppp authentication chap
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

```

例 7-16 配置 ISDN 回拨：中央路由器

```

Headquarters#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!

```

(待续)

```
hostname Headquarters
!
!
username Remote_Site password @ cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 ip address 175.10.50.2 255.255.255.252
 no ip directed-broadcast
!
interface Serial1
 ip address 175.10.1.2 255.255.255.252
 no ip directed-broadcast
 clockrate 125000
!
interface BRI0
 ip address 175.10.23.2 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer callback secure
 dialer enable timeout 5
 dialer map ip 175.10.23.1 name Remote_Site class callback 6129319937
 dialer hold-queue 100
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293198331111
 isdn spid2 61293198461111
 ppp callback accept
 ppp authentication chap
!
no ip classless
!
!
map-class dialer callback
 dialer callback-server username
 dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

注释 回拨功能依赖于 PPP 认证，因此，必须正确配置 PAP 或 CHAP，使回拨功能正常工作。

enable-timeout 命令的设置来确定开始回拨之前要等待的时延。

该定时器的最小值是 1 秒，默认值是 15 秒。在路由器上设置的时间长度应该小于呼叫路由器（远端）用 **dialer wait-for-carrier** 命令为 DDR 的 ISDN 快速呼叫重新路由模式而设置的时间间隔。Cisco 建议呼叫路由器上用 **dialer wait-for-carrier** 命令设置的定时间隔为回拨路由器上设置的定时间隔的 2 倍。本例中，中央路由器上将拨号允许超时设置为 5 秒，而远端路由器 Remote_Site 上的拨号载波等待定时设置为 10 秒。

配置命令 **dialer callback-server username** 使接口在回拨成功协商之后进行呼叫。关键字 **username** 指定了接口将要通过命令 **dialer map** 寻找已通过认证的主机名，以便确定用来进行回拨的拨号字符串的位置。

这里修改了接口配置命令 **dialer map** 以包含关键字 **class**，以及用命令 **map-class** 指定的 **class** 的名称。关键字 **name** 是为了在远端路由器 Remote_Site 拨入时，接口能找到 **dialer map** 声明，从而取得回拨远端路由器 Remote_Site 要用的拨号字符串。例 7-17 是如何利用拨号调试工具来验证回拨设置的例子。

例 7-17 验证回拨的设置

```
Remote_Site#debug dialer
Dial on demand events debugging is on
Remote_Site#ping 175.10.23.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.23.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Remote_Site#
00:30:08: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=175.10.23.2)
00:30:08: BR0 DDR: Attempting to dial 6129319833
00:30:08: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
00:30:08: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833
00:30:08: BR0:1 DDR: Callback negotiated, waiting for server disconnect
00:30:09: %LINK-3-UPDOWN: Interface BRI0:1, changed state to down
00:30:09: DDR: Callback client for Headquarters 9529319833 created
00:30:09: BR0:1 DDR: disconnecting call
00:30:09: BR0:1 DDR: disconnecting call
00:30:14: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
00:30:14: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833
00:30:16: BR0:1 DDR: No callback negotiated
00:30:16: BR0:1 DDR: dialer protocol up
00:30:16: BR0:1 DDR: Callback received from Headquarters 6129319833
00:30:16: DDR: Freeing callback to Headquarters 6129319833
00:30:16: BR0:1 DDR: Call connected, 4 packets unqueued, 4 transmitted, 0 discarded
00:30:17: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
00:30:20: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833
Headquarters
Remote_Site#ping 175.10.23.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.23.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/44 ms
```

```

Headquarters#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319937       4          0        00:00:10    successful
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (5 secs)
Dialer state is data link layer up
Dial reason: Callback return call
Time until disconnect 111 secs
Connected to 6129319937 (Remote_Site)

```

这里看到开始的 5 个 **ping** 命令都不成功，这是由于中央路由断开呼入的业务然后再回拨路由器 **Remote_Site** 所花的时间引起的。这一过程完成之后，**ping** 命令可以正常使用。在运行 **debug dialer** 命令时可以看到回拨的过程。中央路由器上通过 **show dialer** 命令可以清楚地看对它进行呼叫的原因是“回拨呼入的业务”。命令 **debug dialer** 和命令 **show dialer** 可以有效验证路由器是否正确配置了 ISDN 回拨服务。

9. 附加参数：使用多链路 PPP

多链路 PPP 可以使网络数据在多个 WAN 链路上实现负载平衡。启动多链路功能后，多个物理链路可以捆绑在一起，作为一个逻辑链路。在 ISDN 中，可捆绑多个 B 信道捆绑，作为一个大的数据通道使用。多链路的启动是对通过 **dialer load-threshold** 命令定义的阈值做出的响应。该阈值可基于呼入，也可基于呼出数据，或者同时基于呼出和呼入数据，其范围为 1 到 255，其中 1 表示第 2 个 BRI 信道在 ISDN 连接建立的瞬间就要开始工作，而 255 则意味着第 2 个 BRI 信道仅在第 1 个信道完全饱和之后才开始工作。例如，如果要将第 2 条 BRI 信道在第 1 条信道的利用率达到了 50% 的情况下与之捆绑在一起，就可以用 **dialer load-threshold 127** ($255 \times 0.50 = 127$ ，捆绑之后) 命令来进行设置。配置多链路 PPP 需要在接口模式下使用以下命令：

```

encapsulation ppp
dialer load-threshold load
ppp multilink

```

例 7-18 给出了一个多链路 PPP 的配置的实例。

例 7-18 多链路 PPP 的配置

```

Remote_Site#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!

```

（待续）

```
hostname Remote_Site
!
!
username Headquarters password 0 cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Loopback0
 ip address 175.10.101.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 ip address 175.10.2.1 255.255.255.0
 no ip directed-broadcast
 no keepalive
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
 no fair-queue
 clockrate 125000
!
interface Serial1
 description PRIMARY LINK TO HQ
 ip address 175.10.200.1 255.255.255.252
 no ip directed-broadcast
!
interface BRI0
 ip address 175.10.23.1 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer map ip 175.10.23.2 name Headquarters broadcast 6129319833
 dialer load-threshold 2541688
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 ppp multilink
!
router eigrp 1
 network 175.10.0.0
 no auto-summary
!
no ip classless
!
access-list 101 deny eigrp any any
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

这里，当第1条B信道的利用率超过了其呼入或者是呼出业务约10% ($255 \times 0.10 \approx 25$)，而不是25%的阈值，启用两条B信道。

show dialer 命令可以验证多链路PPP的配置情况。例7-19中，路由器通过简单的 **ping** 与链路另一端建立ISDN连接。

例 7-19 验证多链路PPP的配置

```
Remote_Site#ping 175.10.23.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.23.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 44/45/48 ms

Remote_Site#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319833       29         62      00:00:06     successful
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is multilink member
Dial reason: ip (s=175.10.23.1, d=175.10.23.2)
Connected to 6129319833 (Headquarters)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
```

在例子中可以看到已经成功地建立了ISDN连接，第1个B信道实际上是一个多链路的成员。但是，第2条B信道仍然显示它处于空闲状态，并没有和第1条B信道捆绑在一起。原因就是简单的 **ping** 命令产生的链路传输还不足以超过第1条B信道10%利用率的阈值。

如例7-20所示，用 **show interface bri0 1** 命令可以查看第1条B信道的利用情况。

例 7-20 显示第1条B信道的利用率

```
Remote_Site#show interface bri 0 1
BRI0:1 is up, line protocol is up
  Hardware is BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 3/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  LCP Open, multilink Open
  Last input: 00:00:05, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
```

(待续)

```
4183 packets input, 2365973 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4342 packets output, 2396289 bytes, 0 underruns
0 output errors, 0 collisions, 33 interface resets
0 output buffer failures, 0 output buffers swapped out
210 carrier transitions
```

显然，这条 B 信道的利用率仅仅是 3/255。而在配置中，第 2 条 B 信道仅有当第 1 条 B 信道的利用率超过了 25/255（大约 10%）以后才会开始工作。

要在这一配置中让第 2 条 B 信道开始工作，需要增加传输的数据量。一个方法就是如例 7-21 所示使用扩展 ping 命令：

例 7-21 增加数据量以使两条 B 信道都开始工作

```
Remote_Site#ping
Protocol [ip]:
Target IP address: 175.10.23.2
Repeat count [5]: 100
Datagram size [100]: 1500
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 175.10.23.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 212/359/428 ms
```

现在，再用 **show int bri0 1** 命令查看接口的利用率，如例 7-22 所示：

例 7-22 显示两条 B 信道的利用率

```
Remote_Site#show int bri0 1
BRI0:1 is up, line protocol is up
Hardware is BRI
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 27/255
Encapsulation PPP, loopback not set, keepalive set (10 sec)
LCP Open, multilink Open
Last input 00:00:04, output 00:00:04, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 8000 bits/sec, 5 packets/sec
5 minute output rate 7000 bits/sec, 4 packets/sec
4452 packets input, 2579061 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4611 packets output, 2611177 bytes, 0 underruns
0 output errors, 0 collisions, 33 interface resets
0 output buffer failures, 0 output buffers swapped out
212 carrier transitions
```

现在利用率达到了 27/255，因此，第 2 条 B 信道也就已经和第 1 条 B 信道捆绑在一起，开始工作。用 **show dialer** 命令和 **show ppp multilink** 命令都可以对此加以验证，如例 7-23 所示。

例 7-23 用 show dialer 命令验证第 2 条 B 信道的工作情况

```

Remote_Site#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319833       32         62       00:00:55    successful
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is multilink member
Dial reason: ip (s=175.10.23.1, d=175.10.23.2)
Connected to 6129319833 (Headquarters)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is multilink member
Dial reason: Multilink bundle overloaded
Connected to 6129319833 (Headquarters)

```

两条信道都在工作，而第 2 条信道还显示拨号的原因是一个多链路 PPP 过载。

例 7-24 为如何验证多链路 PPP 的配置。

例 7-24 用 show ppp multilink 命令验证第 2 条 B 信道的工作情况

```

ISDN-1#show ppp multilink

Bundle ISDN-2, 2 members, Master link is Virtual-Access1
Dialer Interface is BRI0
  0 lost fragments, 0 reordered, 0 unassigned, sequence 0xF8/0xF8 rcvd/sent
  0 discarded, 0 lost received, 1/255 load

Member Links: 2 (max not set, min not set)
BRI0:1
BRI0:2

```

例子中，两条信道都成为 PPP 多链路成员。

通过链路发送 100 个 1500 字节的 ICMP 数据包足以使第 2 条 B 信道开始工作。两条信道捆绑后形成了一条大的虚拟电路。

10. 第 5 步：ISDN 上的路由数据

如前所述，确保在 ISDN 接口上为数据路由的方法有很多。通常，我们希望数据能在 ISDN 链路建立时有正常的路由传输，但是又不希望路由数据使得链路无限期地保持工作状态。注意一条重要的路由命令 **passive-interface**。被动接口会监听进入的路由更新信息，但不会从接口发送任何路由更新信息。这一特性在按需呼叫接口中非常有用，因为用户可以抑制路由更新信息以免启动不必要的连接。该命令在路由进程中运行，几乎适用于所有的路由选择协议。

例 7-25 是在 EIGRP 协议下的例子。

例 7-25 定义被动接口

```
router eigrp 1
  passive-interface BRI0
  network 175.10.0.0
  no auto-summary
```

如果没有使用 **passive-interface BRI0** 命令，那么 ISDN 链路在有路由更新信息传输的情况下会一直保持连接工作状态。在命令 **dialer map** 中加入了关键字 **broadcast**，该关键字允许多数路由选择协议使用的多播数据通过接口传输。

多数实际应用在 ISDN 中都是使用浮动静态路由选择协议来进行路由的，这样，如果主链路出了故障，ISDN 接口会作为备份开始工作。但是，有很多同样出色的其他协议可以使用户在 ISDN 网络中控制路由信息，这些协议包括：

- 浮动静态路由选择协议。
- OSPF 按需电路协议。
- 拨号监控协议。
- 备份接口协议。
- 备份负载协议。
- 快照路由选择协议。

11. 例 1：浮动静态路由

浮动静态路由是指管理距离值（AD）高于动态路由选择协议的静态路由，管理距离是在 **ip route** 命令末尾的一个参数。每个路由选择协议都有与之相关联的不同管理距离。管理距离简单地讲就是某路由的可靠性程度，或者开销。管理距离越小，与其相关的路由信息就更可靠。如果路由器有多个到达目的网络的路由条目（也就是说有多种路由选择协议同时运行），应选用管理距离最小的路由选择协议条目。表 7-3 列出了直连接口、静态路由和不同路由选择协议的默认管理距离。

表 7-3

默认的管理距离值

协 议	默认管理距离
直连接口	0
静态路由	1
EIGRP 汇总路由	5
BGP	20
内部 EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120

续表

协 议	默认管理距离
EGP	140
外部 EIGRP	170
内部 BGP	200
未知	255

静态路由在分配了比路由选择协议的 AD 更高的 AD 之后变得不固定，称之为“浮动”是因为只有在其他路由选择协议条目消失时，浮动静态路由才会在路由表上“浮”出来为路由器所用。例如，用 `ip route 10.1.1.0 255.255.255.0 BRI 0 200` 命令就能将值为 200 的 AD 手动分配给该路由条目。路由器确认仅在没有其他路由选择协议条目可以到达子网 10.1.1.0/24 时，才会通过 BRI 0 接口将数据包路由到该子网。这在用户希望只有主链路出现故障的情况下，才通过 ISDN 网络对数据进行路由时非常有用，是实际应用中将 ISDN 用作主链路的备份链路的常见方法。图 7-6 为路由器 Cheech 和 Chong 之间的简单网络。路由器的主通信链路是帧中继，而 ISDN 作为备份使用。

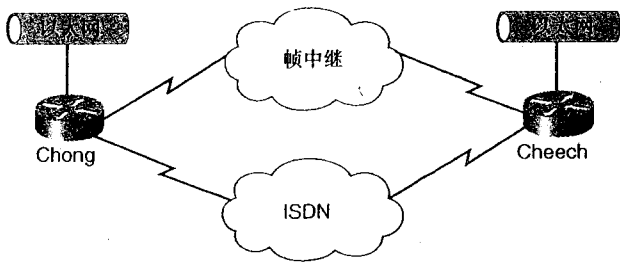


图 7-6 浮动静态路由的例子

下面的例子中，我们希望在两个站点之间的以太网段上维持用户之间的连接。例 7-26 显示了本例中浮动静态路由的配置方法。

例 7-26 浮动静态路由的配置

```
Cheech#show running-config
Building configuration...

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Cheech
!
enable password cisco
!
username Chong password 0 cisco
```

(待续)

```

isdn switch-type basic-dms100
!
interface Loopback0
 ip address 175.10.2.2 255.255.255.0
!
interface Ethernet0
 ip address 175.10.22.1 255.255.255.0
 no keepalive
 no mop enabled
!
interface Serial0
 ip address 175.10.123.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 175.10.123.2 300 broadcast
!
interface Serial1
 no ip address
 shutdown
!
interface BRI0
 ip address 175.10.23.1 255.255.255.0
 encapsulation ppp
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 dialer map ip 175.10.23.2 broadcast 6129319833
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
router eigrp 1
 passive-interface BRI0
 network 175.10.0.0
 no auto-summary
!
ip classless
ip route 175.10.35.0 255.255.255.0 175.10.23.2 200
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 password cisco
 login
!
end

```

```

Chong#show running-config
Building configuration...

```

```

Current configuration:

```

```

!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Chong
!

```

(待续)

```

enable password cisco
!
username Cheech password 0 cisco
isdn switch-type basic-dms100
!
interface Loopback0
 ip address 175.10.3.3 255.255.255.0
!
interface Ethernet0
 ip address 175.10.35.3 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
!
interface Serial0.1 point-to-point
 ip address 175.10.123.2 255.255.255.0
 frame-relay interface-dlci 200
!
interface Serial0.2 point-to-point
 ip address 175.10.134.2 255.255.255.252
 shutdown
 frame-relay interface-dlci 400
!
interface Serial1
 no ip address
 shutdown
!
interface BRI0
 ip address 175.10.23.2 255.255.255.0
 encapsulation ppp
 isdn spid1 61293198331111
 isdn spid2 61293198461111
 dialer map ip 175.10.23.1 broadcast 6129319937
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
router eigrp 1
 passive-interface BRI0
 network 175.10.0.0
 no auto-summary
!
no ip classless
ip route 175.10.22.0 255.255.255.224 175.10.23.1 200
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 password cisco
 login
!
end

```

EIGRP 过程中已经将 BRI 0 接口设置为被动接口，以防止路由更新信息启动呼叫，避免链路无限期维持激活状态。由于拨号列表中所有的 IP 数据包都指定为用户所期望的数据，EIGRP 多播更新信息将启动 ISDN 链路工作。解决这种问题的另一种方法是制定更严格的限制条件来拒绝 EIGRP 数据包。

12. 例 2: OSPF 按需电路

顾名思义，OSPF 按需电路支持在 ISDN 链路上运行 OSPF 协议。第 4 章介绍过该协议，第 12 章“链路状态协议：开放式最短路径优先 (OSPF)”中还会更详细地讲述该协议的内容。在第 12 章会讲到，按需电路可以抑制 OSPF hello 握手数据包在 ISDN 链路上的传输，而 ISDN 呼叫只在发生了逻辑拓扑变化之后才会启动。这一功能是通过下面的接口命令实现的：

```
ISDN_Router#config term
ISDN_Router (config) #int bri0
ISDN_Router (config-if) #ip ospf demand-circuit
```

看上去很简单。但是请记住，OSPF 链路状态的数据库发生变化后，会产生 ISDN 呼叫。因此，某些情况下，比如要将其他协议重分布到原有的 OSPF 协议时，这时一定要小心避免产生路由环路而导致链路状态不停地变化。很快就可以在一个高级实验中看到这样的例子，该例中不是简单使用 **ip ospf demand-circuit** 命令使网络正常工作。图 7-7 是将要用来演示 OSPF 按需电路配置的参考网络。

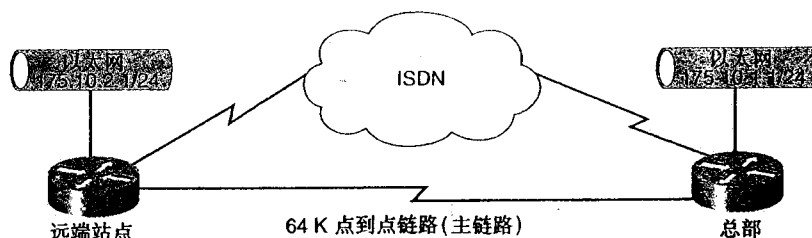


图 7-7 OSPF 按需电路配置实例

例 7-27 是中心路由器和远端路由器的配置情况。

例 7-27 OSPF 按需电路的配置

```
Remote_Site#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Remote_Site
!
!
username Headquarters password 0 cisco
ip subnet-zero
```

(待续)

```

isdn switch-type basic-dms100
!
!
!
interface Loopback0
 ip address 175.10.101.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 ip address 175.10.2.1 255.255.255.0
 no ip directed-broadcast
 no keepalive
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
 no fair-queue

!
interface Serial1
 description PRIMARY LINK TO HQ
 ip address 175.10.200.1 255.255.255.252
 no ip directed-broadcast
 bandwidth 64
!
interface BRI0
 ip address 175.10.23.1 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 ip ospf demand-circuit
 ip ospf cost 9999
 dialer map ip 175.10.23.2 name Headquarters broadcast 6129319833
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
!
router ospf 1
 network 175.10.0.0 0.0.255.255 area 0
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

Headquarters#show running-config
Building configuration...

Current configuration:
!

```

（待续）

```
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Headquarters
!
logging buffered 9096 debugging
!
username Remote_Site password 0 cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Loopback0
 ip address 175.10.102.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 ip address 175.10.1.1 255.255.255.0
 no ip directed-broadcast
 no keepalive
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Serial1
 description PRIMARY LINK TO REMOTE SITE
 ip address 175.10.200.2 255.255.255.252
 no ip directed-broadcast
 clockrate 64000
 bandwidth 64
!
interface BRI0
 ip address 175.10.23.2 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293198331111
 isdn spid2 61293198461111
!
router ospf 1
 network 175.10.0.0 0.0.255.255 area 0
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

接口模式下的配置命令 **ip ospf demand circuit** 使路由器不通过接口发送 hello 握手数据包。如果没有该命令，OSPF 的 hello 数据包会使链路一直保持工作状态。

例 7-28 是 **show ip ospf interface bri0** 命令的示例。

例 7-28 **show ip ospf interface** 命令的使用情况

```
Remote_Site#show ip ospf interface bri0
BRI0 is up, line protocol is up (spoofing)
Internet Address 175.10.23.1/29, Area 0
Process ID 1, Router ID 175.10.101.1, Network Type POINT_TO_POINT, Cost: 9999
Configured as demand circuit.
Run as demand circuit.
DoNotAge LSA allowed.
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:01
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 175.10.102.1 (Hello suppressed)
Suppress hello for 1 neighbor(s)
```

可见，OSPF 在中心路由器产生一个邻接关系，并抑制 hello 握手数据包不通过 BRI0 接口发送。

在该网络中，主链路是一个 64 kbit/s 的点对点线路。在本书的 OSPF 部分可以了解到，清楚标明使用 OSPF 协议的每个接口的带宽是个好习惯，可以通过在路由器的 Serial 1 接口上使用 **bandwidth 64** 命令来完成。OSPF 中使用的尺度是开销 (Cost)，计算方式是 100 000 000/带宽。

OSPF 会将串口的带宽假定为一个完全的 T1 接口，并且会将接口的开销值错误地设定为 64，除非在串行接口中特别说明其实际的带宽值。对 ISDN 接口来说，OSPF 假定链路是 64 kbit/s，并为其分配 1562 的开销值。这些信息可以通过 **show ip ospf interface bri0** 命令显示，如例 7-29 所示。

例 7-29 在 OSPF 环境中为 ISDN 接口配置 cost 值

```
Remote_Site#show ip ospf int bri0
BRI0 is up, line protocol is up (spoofing)
Internet Address 175.10.23.1/29, Area 0
Process ID 1, Router ID 175.10.101.1, Network Type POINT_TO_POINT, Cost: 1562
Configured as demand circuit.
Run as demand circuit.
DoNotAge LSA allowed.
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:03
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 175.10.102.1 (Hello suppressed)
Suppress hello for 1 neighbor(s)
```

请注意例子中 OSPF 的 hello 握手数据包已经被抑制，不会通过 ISDN 网络传输。

这里的问题是，OSPF 将 1562 的开销值既分配给了 Serial 1 接口也分配给了 BRI 接口，这些链路将被看作等价开销的路径，数据在两个链路上进行负载平衡。这会使得 ISDN 链路

无限期地处于工作状态。要解决这一问题，通常的做法是分配一个自定义的高开销值给 BRI 接口，以确保数据只有在其他可选路径都不可用的情况下才会通过该链路路由。

这个例子中，用 `ip ospf cost 9999` 命令为 ISDN 连接分配值为 9999 的开销。这和实际采用的开销值大小关系不大，但是这个值应该很大，大到 OSPF 不会选择它出现在路由表中。

通常情况下，路由表应该和例 7-30 中所示类似。

例 7-30 OSPF 按需电路的设置测试

```
Remote_Site#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    175.10.0.0/16 is variably subnetted, 6 subnets, 4 masks
C       175.10.200.0/30 is directly connected, Serial1
O       175.10.1.0/24 [110/1572] via 175.10.200.2, 00:00:39, Serial1
C       175.10.2.0/24 is directly connected, Ethernet0
C       175.10.23.0/29 is directly connected, BRI0
C       175.10.101.1/32 is directly connected, Loopback0
O       175.10.102.1/32 [110/1563] via 175.10.200.2, 00:00:39, Serial1
Remote_Site#ping 175.10.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/18/20 ms
```

现在手动关闭 Serial 1 接口，并且证实和中心路由器的连接依然存在，如例 7-31 所示。

例 7-31 关闭掉主链路测试 ISDN 的备份功能

```
Remote_Site#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Remote_Site(config)#int s1
Remote_Site(config-if)#shut
Remote_Site(config-if)#
3d03h: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
Remote_Site(config-if)#end
Remote_Site#
3d03h: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833
3d03h: %LINK-5-CHANGED: Interface Serial1, changed state to administratively down
n
3d03h: %SYS-5-CONFIG_I: Configured from console by console
3d03h: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to
up
3d03h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to
down
```

(待续)

```

Remote_Site#
3d03h: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833 Headquarters
Remote_Site#sho 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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    175.10.0.0/16 is variably subnetted, 6 subnets, 3 masks
O       175.10.1.0/24 [110/10009] via 175.10.23.2, 00:00:06, BRI0
C       175.10.2.0/24 is directly connected, Ethernet0
C       175.10.23.2/32 is directly connected, BRI0
C       175.10.23.0/29 is directly connected, BRI0
C       175.10.101.1/32 is directly connected, Loopback0
O       175.10.102.1/32 [110/10000] via 175.10.23.2, 00:00:06, BRI0
Remote_Site#ping 175.10.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/43/44 ms
    
```

由于 OSPF 检测到网络中有拓扑变化，于是建立连接。如例 7-32 所示执行 **show dialer** 命令可以看到 ISDN 呼叫建立的原因是有数据包传输到目的地址 224.0.0.5，这就是 OSPF 所用的多播地址。

例 7-32 show dialer 命令显示 ISDN 呼叫的原因

```

Remote_Site#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319833       44         62       00:01:55     successful
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=175.10.23.1, d=224.0.0.5)
Time until disconnect 49 secs
Connected to 6129319833 (Headquarters)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
    
```

主链路恢复后，经过拨号空闲超时间隔，ISDN 连接被断开。这在网络中很容易产生路由环路，尤其是在对多个路由选择协议进行重分布时。一定要采取过滤措施避免这些环路产生不必要的 ISDN 连接。

13. 例 3: 拨号监控 (dialer watch)

Cisco IOS 11.3 (2) T 具有 dialer watch 的功能。dialer watch 让 IOS 对路由表进行监控以跟踪用户自定义的路由条目。删除一个或多个 dialer watch 表中定义的路由条目时会触发 ISDN 连接的建立。运用 dialer watch 的速度和效果是由所用路由选择协议的收敛时间和性质决定。EIGRP 的效果最好，因而通常和 dialer watch 功能一起使用，OSPF 和 IGRP 也支持 dialer watch。

dialer watch 将 ISDN 网络作为备份方式在远程站点之间提供可靠的连接，它监听路由表，当主链路出故障时，在 dialer watch 表中定义的路由条目在路由表中消失后，自动启动呼叫过程。其工作过程如下：

1 如果监控的路由条目被删除，dialer watch 会查找定义的被监控 IP 地址的合法路由。

2 如果没有合法路由，认为主链路已不可用。

3 如果找到了至少一个被监控 IP 地址的合法路由，而且该路由指向某个接口，而不是作为 dialer watch 配置的备份接口，就认为主链路还在工作。例如，远程站点有两个通到中心路由器的帧中继 PVC，一个 PVC 已经停止工作，但数据仍然可以通过另一个 PVC 而不是 ISDN 链路进行传输。

4 如果主链路断开，路由选择协议会立即通知 dialer watch 进程，第二链路（这里是 ISDN 线路）会马上开始工作。

5 第二链路开始工作后，会在每个空闲超时时段快结束时检查主链路。

6 如果主链路仍然是断开状态，空闲定时器重置。

如果发现主链路已经恢复，断开第二链路。可设置关闭定时器来延长第二链路的断开时间。

对图 7-8 中的网络来说，备份链路是通过查看 175.10.1.0/24 子网和中心路由器的环路地址来实现的，这样即使帧中继网络停止工作，还能保证整个连接的畅通。

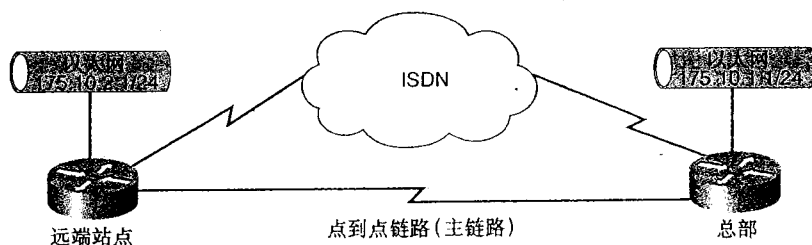


图 7-8 dialer watch 配置实例的网络拓扑结构

例 7-33 是 dialer watch 的配置实例。

例 7-33 配置 dialer watch

```
Remote_Site#show running-config
Building configuration...

Current configuration:
!
```

(待续)

```

version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Remote_Site
!
!
username Headquarters password 0 cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Loopback0
 ip address 175.10.101.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 ip address 175.10.2.1 255.255.255.0
 no ip directed-broadcast
 no keepalive
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
 no fair-queue
 clockrate 125000
!
interface Serial1
 description PRIMARY LINK TO HQ
 ip address 175.10.200.1 255.255.255.252
 no ip directed-broadcast
!
interface BRI0
 ip address 175.10.23.1 255.255.255.248
 no ip directed-broadcast
 encapsulation ppp
 dialer map ip 175.10.23.2 name Headquarters broadcast 6129319833
 dialer map ip 175.10.102.1 name Headquarters broadcast 6129319833
 dialer map ip 175.10.1.0 name Headquarters broadcast 6129319833
 dialer watch-group 1
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
!
router eigrp 1
 network 175.10.0.0
 no auto-summary
!
no ip classless
!
access-list 101 deny eigrp any any
access-list 101 permit ip any any
dialer watch-list 1 ip 175.10.102.1 255.255.255.255
dialer watch-list 1 ip 175.10.1.0 255.255.255.0
dialer-list 1 protocol ip list 101
!
!

```

(待续)

```

line con 0
  privilege level 15
  logging synchronous
  transport input none
line aux 0
line vty 0 4
  login
!
end

Headquarters#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Headquarters
!
!
username Remote_Site password 0 cisco
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Loopback0
  ip address 175.10.102.1 255.255.255.255
  no ip directed-broadcast
!
interface Ethernet0
  ip address 175.10.1.1 255.255.255.0
  no ip directed-broadcast
  no keepalive
!
interface Serial0
  no ip address
  no ip directed-broadcast
  shutdown
!
interface Serial1
  description PRIMARY LINK TO REMOTE SITE
  ip address 175.10.200.2 255.255.255.252
  no ip directed-broadcast
  clockrate 125000
!
interface BRI0
  ip address 175.10.23.2 255.255.255.248
  no ip directed-broadcast
  encapsulation ppp
  dialer-group 1
  isdn switch-type basic-dms100
  isdn spid1 61293198331111
  isdn spid2 61293198461111
!
router eigrp 1
  network 175.10.0.0

```

（待续）

```

no auto-summary
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
  privilege level 15
  logging synchronous
  transport input none
line aux 0
line vty 0 4
  login
!
end

```

如果 dialer watch 表中定义的任意路由条目仍在路由表中，就认为主链路接口处在正常工作状态，ISDN 呼叫不会开始。这个例子中，dialer watch 要让路由器去触发一次呼叫，必须到中心路由器以太网段和环路地址的路由都从路由表中消失。环路地址是利用 dialer watch 的有效途径，这是因为该接口总是处于工作状态。如果在 dialer watch 表中只指定以太网段，那么只要以太接口断开一次，就要产生一次 ISDN 呼叫。这样的结果不是我们所期望的，因为这时主链路仍然在工作，如果以太接口断开，就不能通过备份链路或主链路访问以太网段。这里所举的例子并不是简单地把中心路由器环路网段放到 dialer watch 表中，相反，以太接口和环路网络都放到了 dialer watch 表里。这样，要想建立呼叫，所有被监控路由条目必须都消失。

警告 dialer watch 表中定义的网络必须和路由表中的网络和子网掩码严格匹配。例如，如果路由表显示 175.10.0.0/16，而配置显示的是 **dialer watch-list 1 ip 175.10.0.0 255.255.255.0**，那么 dialer watch 过程就不能检测到 175.10.0.0/16 已经不在路由表中的事实。

14. 例 4：接口备份

用 ISDN 备份主链路的另一有效方法是使用命令 **backup interface**。这条命令能够跟踪主链路接口的链路状态，仅当该状态从工作状态转为断开状态时触发 ISDN 连接的建立。

注释 一些书籍中将这一点解释成只有主链路接口的线路协议不工作时，备份接口才开始工作，这种说法是不正确的。实际物理接口状态必须是 down，而其协议也是 down，这样备份接口才会工作。后面会给出一个例子，该例中，接口被手动关闭，线路协议也停止工作，备份接口也没有工作。

备份接口不像浮动静态路由那样使用广泛，因为验证 ISDN 链路是否正常工作的唯一途径就是要将整个主链路网络用物理方法使它停止工作。

配置备份接口只需要一条命令：

Router (config-if) **#backup interface bri 0**

该命令是在主链路的接口下而**不是**在 ISDN 的接口下设置。一个值得注意的可选项是

backup delay 命令。它能让用户决定 ISDN 在检测到主链路故障，开始进行呼叫之前要等待

的时间，以及当主链路恢复工作后，备份网络断开连接的时间延迟。这在主链路网络遇到了频繁而短暂的故障，用户不想 ISDN 路由器每次都触发呼叫的情况下很有用。可以通过例 7-34 中的方法来实现，这一配置过程也是在主链路的接口配置模式下完成的。

例 7-34 配置 Backup interface

```
interface Serial0
 backup delay 5 60
 backup interface BRI0
 ip address 175.10.123.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 175.10.123.2 300 broadcast
```

这里的主链路是 Serial 0 上的帧中继链路，备份链路是 BRI 0 上的 ISDN 链路。ISDN 连接在 Serial 0 接口停止工作后触发一次呼叫前会等待 5 秒，而在 Serial 0 重新开始工作之后，中断 ISDN 的链路之前要等待 60 秒。备份延时配置的单位是秒，其值从 0 到 4 294 967 294。备份延时的配置是可选的，如果没有设置，ISDN 的开始呼叫和中断都是瞬时的，没有任何延时。

设置延迟时间可避免快速抖动线路启动 ISDN。抖动线路会对链路状态路由选择协议，如 OSPF 的工作造成破坏。

将 ISDN 链路配置为备份接口时，BRI 0 接口状态和线路协议的状态都会从工作状态分别转为待机状态和停止工作状态，如例 7-35 所示。

例 7-35 使用备份接口配置时的 ISDN 接口状态

```
Cheech#show int bri0
BRI0 is standby mode, line protocol is down
Hardware is BRI
Internet address is 175.10.23.1/24
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set
Last input 00:19:42, output 00:19:42, output hang never
Last clearing of "show interface" counters 1d04h
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 21175 packets input, 88385 bytes, 0 no buffer
  Received 4 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 21175 packets output, 89187 bytes, 0 underruns
  0 output errors, 0 collisions, 2 interface resets
  0 output buffer failures, 0 output buffers swapped out
 3 carrier transitions
```

备份接口保持在待机状态，在主链路停止工作之前不可用。

图 7-9 给出了备份接口配置的参考网络。

例 7-36 是备份接口的配置实例。

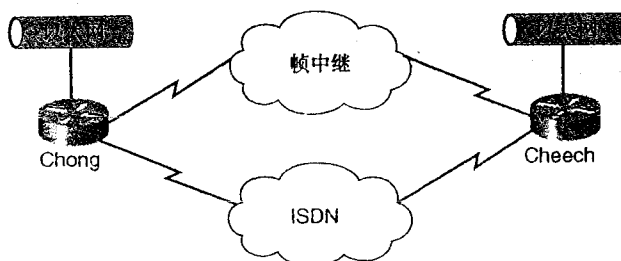


图 7-9 备份接口配置的网络拓扑结构

例 7-36 配置备份接口

```

Cheech#show running-config
Building configuration...

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Cheech
!
enable password cisco
!
username Chong password 0 cisco
isdn switch-type basic-dms100
!
interface Loopback0
 ip address 175.10.2.2 255.255.255.0
!
interface Ethernet0
 ip address 175.10.22.1 255.255.255.0
 no keepalive
 no mop enabled
!
interface Serial0
 backup delay 5 60
 backup interface BRI0
 ip address 175.10.123.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 175.10.123.2 300 broadcast
!
interface Serial1
 no ip address
 shutdown
!
interface BRI0
 ip address 175.10.23.1 255.255.255.0
 encapsulation ppp
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 dialer map ip 175.10.23.2 broadcast 6129319833
 dialer-group 1
 no fair-queue

```

(待续)


```
ppp authentication chap
!
router eigrp 1
 network 175.10.0.0
 no auto-summary
!
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 password cisco
 login
!
end
```

```
Chong#show running-config
Building configuration...
```

```
Current configuration:
```

```
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Chong
!
enable password cisco
!
username Cheech password 0 cisco
isdn switch-type basic-dms100
!
interface Loopback0
 ip address 175.10.3.3 255.255.255.0
!
interface Ethernet0
 ip address 175.10.35.3 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
!
interface Serial0.1 point-to-point
 ip address 175.10.123.2 255.255.255.0
 frame-relay interface-dlci 200
!
interface Serial1
 no ip address
 shutdown
!
interface BRI0
 ip address 175.10.23.2 255.255.255.0
 encapsulation ppp
 isdn spid1 61293198331111
```

(待续)

```

isdn spid2 61293198461111
dialer idle-timeout 9999
dialer-group 1
no fair-queue
ppp authentication chap
!
router eigrp 1
 network 175.10.0.0
 no auto-summary
!
no ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 password cisco
 login
!
end

```

上面的例子中，不用在路由器 Chong 做其他特别设置，BRI 0 接口在 EIGRP 路由设置中也没有设置成被动接口。通常，如果不这样设置，EIGRP 会使 ISDN 链路一直处于工作状态，但由于该接口设置成备份接口，除非 Serial 0 接口停止工作，它是不会开始工作的。

在测试备份接口时，认识到一点很重要，这就是并不是简单地手动关掉 Cheech 路由器上的串口就可以使备份接口开始工作。备份接口的连接只有在主链路接口真正停止工作后（而不是手动关闭）才会建立。但是，如果在路由器 Cheech 上采用了逻辑子接口，也就是用 Serial 0.1 通过帧中继和 Chong 相连，那么在 Serial 0.1 上可能也作了备份接口命令的配置。这样的话，如果关掉了物理接口 Serial 0，那么 Serial 0.1 接口也应该停止工作（不是手动关闭），ISDN 备份链路应该开始工作。但是，在上面的例子中，我们用备份命令在物理串行接口中作了配置，要启动 ISDN 开始工作，需要将帧中继线路和路由器断开（或者是将帧中继交换机断电）。

例 7-37 是 Cheech 路由器在帧中继断开前的路由表，例 7-38 是帧中继断开后的路由表。

例 7-37 Cheech 路由器在帧中继断开前的路由表

```

Cheech#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

175.10.0.0/24 is subnetted, 7 subnets

```

（待续）

```

D      175.10.35.0 [90/2195456] via 175.10.123.2, 00:05:07, Serial0
D      175.10.5.0 [90/2323456] via 175.10.123.2, 00:05:07, Serial0
D      175.10.3.0 [90/2297856] via 175.10.123.2, 00:05:07, Serial0
C      175.10.2.0 is directly connected, Loopback0
D      175.10.23.0 [90/41024000] via 175.10.123.2, 00:04:41, Serial0
C      175.10.22.0 is directly connected, Ethernet0
C      175.10.123.0 is directly connected, Serial0
Cheech# ping 175.10.35.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.35.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms

```

例 7-38 Cheech 路由器在帧中继断开后的路由表

```

Cheech#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

175.10.0.0/16 is variably subnetted, 7 subnets, 2 masks
D      175.10.35.0/24 [90/40537600] via 175.10.23.2, 00:00:12, BRI0
D      175.10.5.0/24 [90/40665600] via 175.10.23.2, 00:00:12, BRI0
D      175.10.3.0/24 [90/40640000] via 175.10.23.2, 00:00:12, BRI0
C      175.10.2.0/24 is directly connected, Loopback0
C      175.10.23.2/32 is directly connected, BRI0
C      175.10.23.0/24 is directly connected, BRI0
C      175.10.22.0/24 is directly connected, Ethernet0
Cheech#ping 175.10.35.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.35.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/41/44 ms
Cheech#

```

帧中继线路和路由器 Cheech 的连接拔除之后，Cheech 会在 5 秒之后拨入 Chong 路由器。线路恢复连接后，ISDN 在 60 秒之后断开呼叫。当然，这些时间间隔也可以由用户自行定义。

15. 例 5: backup load 命令的使用

Backup load 命令在主链路过载时也很有用，通过在主链路接口上使用 **backup load** 命令可以提供更多的带宽，同时还提供了配置路由器使得 ISDN 能够用于故障链路的备份的方法，用这种方法启动 ISDN 链路的主要优点在于它能够结合在任何路由选择协议和可路由选择协议上。这是按需分配带宽，而不是备份。然而，**backup load** 需要和 **backup interface** 命令配合工作，如下所示。

可以通过 **backup interface** 命令把 Cisco IOS 接口置于备份模式：

- 接口配置命令 **backup interface** 指定备份的接口。
- **backup load** 命令指定备份接口开始工作和停止工作的流量阈值。
- **backup delay** 命令指定主接口状态发生改变之后，备份接口开始工作和停止工作之前的等待时间。

备份接口通常将用做备份的接口锁定到备份状态，以使它不用于其他用途。dialer profile 可以解除锁定状态，使物理接口有多种用途。浮动静态路由 DDR 也能消除拨号接口的锁定状态。

以例 7-39 的配置为例，BRI 0 只有在串口 1/0（主链路）停止工作后才启动。**backup delay** 命令的设置使备份链路在串口 0 停止工作之后，启动备份之前的等待时间是 30 秒，而串口 0/1 恢复之后，备份断开之前的等待时间是 60 秒。

例 7-39 配置 Backup Delay 命令选项

```
interface serial 1/0
  ip address 172.20.1.4 255.255.255.0
  backup interface bri 2/0
  backup delay 30 60
```

以例 7-40 的配置为例，只有在 Serial 0（主链路）的带宽利用率超过了 75%的情况下，BRI 2/0 才会开始工作。主链路和备份链路所承受的负载之和不足主链路带宽 5%的情况下，备份链路会停止工作。

例 7-40 使用 Backup Load 进行配置的情况

```
interface serial 1/0
  ip address 172.20.1.4 255.255.255.0
  backup interface bri 2/0
  backup load 75 5
```

以例 7-41 的配置为例，只有串口 1/0 停止工作或利用率超过了 25%这两种情况下，BRI 2/0 才会启动工作。串口 1/0 停止工作后 BRI 0 开始工作前等待 10 秒。串口 1/0 恢复后 BRI 2/0 还会再保持工作状态 60 秒。如果 BRI 2/0 的启动是由串口 1/0 上的利用率过高引起的，当串口 1/0 和 BRI 2/0 所承受的负载之和不到串口 1/0 的带宽的 5%时，BRI 2/0 就会停止工作。

例 7-41 配置备份负载应用示例

```
interface serial 1/0
  ip address 172.20.1.4 255.255.255.0
  backup interface bri 2/0
  backup load 25 5
  backup delay 10 60
```

要注意多链路 PPP 和备份负载之间的区别。在多链路 PPP 中，拨号负载阈值是在 1 到 255 之间，而在备份负载中，启动备份链路的阈值是一个纯百分数（1 到 100 之间）。

16. 例6: 快照路由

快照路由基于客户/服务器原理，在快照路由中，路由器（通常是中心路由器）是指定为快照服务器，而一台或多台路由器（远程站点）则是快照客户端。客户端在某些指定的时间（称为活动期间）与服务器建立连接，以获取服务器里的路由信息。快照是指客户端的活动时间结束之后，ISDN 断开连接，但是客户端仍然会保留路由表的“快照”。ISDN 线路空闲时，这些路由条目会存入客户端的路由表中。空闲时间之后，客户端拨入快照服务器以获取最新的路由信息。空闲时间和活动时间的长度是可以设置的（活动时间可以从5分钟到1000分钟，而空闲时间可以从8分钟到1000000分钟）。

由于链路状态协议依靠定期性的 hello 信号保持邻接设备的工作，快照路由只能和非链路状态协议，如和 IP 的 IGRP 和 RIP，IPX 的 RIP 以及 AppleTalk 的 RTMP 等一起使用。

17. 快照客户端的配置

快照路由的配置相对来说要简单一些。对客户端路由器来说，只需要在配置模式下使用两个命令。第1个命令：

```
router (config-if) #snapshot client active-time quiet-time [suppress-statechange-updates] [dialer]
```

这条命令设置活动时间和空闲时间。选项 **suppress-statechange-updates** 禁止每次由额外数据传输引起链路工作的路由更新信息进行传输。默认情况下，无论什么原因，只要 ISDN 连接处于工作状态之中，路由信息就会进行传输。选项 **dialer** 则是用来让客户端路由器在即使没有周期性的数据时也继续拨入快照服务器，并且指向第2步中指定的适当的拨号映射。

第2个命令是：

```
router (config-if) # dialer map snapshot sequence-number name name dial-string
```

这条命令定义了向快照服务器获取路由更新信息的拨号映射。

在该命令中使用 **help** (或?) 时，有一点容易产生歧义，即系统提示输入协议地址，而实际上需要的只是序列号而已。可以参考例7-42。

例 7-42 dialer map snapshot 命令的歧义之处示例

```
Cheech#conf t
Cheech(config)#int bri0
Cheech(config-if)#dialer map snapshot ?
N Protocol specific address
```

这里 IOS 似乎需要一个具体的协议地址，而实际上只需要一个序列号。序列号用来在客户呼叫服务路由器时识别拨号映射和指定优先级序号（从1到254）。如果只有一个服务器，可以使用任何数字。要注意只需输入1到254之间的任何数字，而不是某个第3层的地址值。

18. 快照服务器的配置

这一配置更简单，只需要一条命令：

```
router (config-if) #snapshot server active-time [dialer]
```

参数 *active-timer* 值必须和客户端上设置的值匹配。

19. 快照路由的配置

在图 7-10 的简单网络中，两个站点之间的主通信链路是帧中继，快照路由则用来交换二者之间的路由信息，这样即使帧中继出现问题，ISDN 连接仍然是畅通的。BRI 接口设置为被动接口以避免路由更新信息触发呼叫建立。

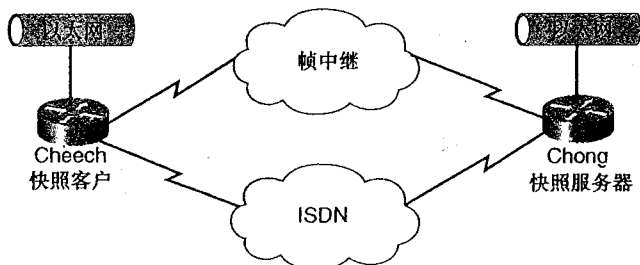


图 7-10 快照网络参考

例 7-43 是快照路由的正确配置。

例 7-43 配置快照路由器

```
Cheech#show running-configuration
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Cheech
!
enable password cisco
!
username Chong password 0 cisco
isdn switch-type basic-dms100
!
interface Loopback0
 ip address 175.10.2.2 255.255.255.0
!
interface Ethernet0
 ip address 175.10.22.1 255.255.255.0
 no keepalive
 no mop enabled
!
interface Serial0
 ip address 175.10.123.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 175.10.123.2 300 broadcast
!
interface Serial1
 no ip address
 shutdown
!
interface BRI0
 ip address 175.10.23.1 255.255.255.0
 encapsulation ppp
```

（待续）

```

isdn spid1 61293199371111
isdn spid2 61293199381111
dialer map snapshot name Chong 6129319833
dialer map ip 175.10.23.2 broadcast 6129319833
dialer-group 1
snapshot client 10 20 dialer
no fair-queue
ppp authentication chap
!
router igrp 1
network 175.10.0.0
!
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
line aux 0
line vty 0 4
password cisco
login
!
end

```

```

Chong#show running-configuration
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Chong
!
enable password cisco
!
username Cheech password 0 cisco
isdn switch-type basic-dms100
!
interface Loopback0
ip address 175.10.3.3 255.255.255.0
!
interface Ethernet0
ip address 175.10.35.3 255.255.255.0
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
!
interface Serial0.1 point-to-point
ip address 175.10.123.2 255.255.255.0
frame-relay interface-dlci 200
!
interface Serial1
no ip address
shutdown
!
interface BRI0
ip address 175.10.23.2 255.255.255.0
encapsulation ppp

```

```

isdn spid1 61293198331111
isdn spid2 61293198461111
dialer-group 1
snapshot server 10
no fair-queue
ppp authentication chap
!
router igrp 1
network 175.10.0.0
!
no ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
line aux 0
line vty 0 4
password cisco
login
!
end

```

这里有一点很重要，BRI 接口没有在 IGRP 中设置为被动接口。由于路由数据会通过 ISDN 链路进行传输，快照路由会阻止该链路一直处于工作状态。为了确定路由器在活动状态期间确实建立了连接，路由数据确实在传输，可以使用 **show dialer** 命令和 **show snapshot** 命令，如例 7-44 所示。

例 7-44 验证活动状态时期路由器的连接状况和路由数据的传输情况

```

Cheech#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319833       85         0         00:10:22    successful
0 incoming call(s) have been screened.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: snapshot
Time until disconnect 6 secs
Connected to 6129319833 (Chong)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
Cheech#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319833       85         0         00:10:26    successful

```

(待续)


```

0 incoming call(s) have been screened.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: Snapshot
Time until disconnect

```

该 ISDN 连接是为了交换快照路由信息。上一例中，这些信息的交换是每 20 分钟进行一次，每次持续 10 分钟。在实际应用中，由于链路每 20 分钟工作一次可能过于频繁，开销太高，因此，通常都会加长空闲时间。

Show snapshot 命令可以验证快照过程是否正常工作，还可以显示所使用的快照路由选项，如例 7-45 所示。

例 7-45 验证快照过程和所用选项

```

Cheech#show snapshot
BRI0 is up, line protocol is up Snapshot client
Options: dialer support
Length of active period:          10 minutes
Length of quiet period:           20 minutes
Length of retry period:           13 minutes
For dialer address 1
Current state: active, remaining/exchange time: 8/2 minutes
Connected dialer interface:
BRI0:1
Updates received this cycle: ip

```

只有 IP 路由允许通过，并可以证明该路由器配置成使用拨号信息的客户端，而且设置了活动时间，空闲时间以及失败重试时间。默认的失败重试时间是所设置的活动时间再加上 3 分钟。

如果要测试帧中继链路断开后 ISDN 链路是否会继续工作，从路由器 Cheech 断开帧中继线路连接。例 7-46 和 7-47 给出了帧中继链路断开前后的路由表情况。

例 7-46 帧中继链路断开前的路由表

```

Cheech#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, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

175.10.0.0/24 is subnetted, 6 subnets
I    175.10.35.0 [100/8576] via 175.10.123.2, 00:00:04, Serial0
I    175.10.3.0 [100/8976] via 175.10.123.2, 00:00:04, Serial0

```

(待续)

		第	WAN 协	术：综	数字网	1)	3
	1	.2.0 is	ctly co	ed, Loop			
	1	.23.0 i	ectly c	ted, BR			
	1	.22.0 i	ectly c	ted, Et	to		
	1	.123.0	rectly	sted, Sc			
eech# p		75.10.3					
pe esca	quence	ort.					
ending 5	-byte I	chos to	10.35.3	out is	onds:		
!!!!							
ccess r	s 100 p	t (5/5)	nd-trip	avg/max	7/8 ms		

7-47 继链路 后的路

eech#sh	route						
odes: C	ected,	static,	IGRP, R	, M - m	, B - B		
D	RP, EX	RP exte	0 - OSPF	- OSPF	area		
N1	PF NSSA	onal typ	N2 - OS	SA exte	type 2		
E1	PF exte	type 1,	OSPF ex	l type	- EGP		
i	IS, L1	IS leve	.2 - IS-	level-2,	andidat	ault	
U	-user s	route,	DDR				
ateway c	t resor	not set					
175.	0/16 is	ably sut	ed, 6 su	, 2 mas			
1	.35.0/2	0/158350	a 175.10	, 00:00	3R10		
1	.3.0/24	/158750]	175.10.	00:00:	10		
1	.2.0/24	irectly	ected, L	ok0			
1	.23.2/3	irectly	ected,				
1	.23.0/2	irectly	ected,				
1	.22.0/2	irectly	ected,	net0			
eech#pi	5.10.35						
ype esca	quence	ort.					
ending: 5	-byte I	chos to	0.35.3,	out is	onds:		
!!!!							
ccess r	s 100 p	t (5/5),	nd-trip	vg/max	43/44 m		

立 ISDN 只需在 分钟。在 继链路 正常后 链路 数据传输 很 快

IS N 调 的 ‘ g sh ’ 和 ‘Big ’ 命

面讲述 了 SDN 古 的技巧， 有助于 并缩小 N 故障 的范围

4.1 ON 的 ig sh

下是在 问题和 故障范围 方面非常 用的命令 是在 确认 SDN 链 否 正 作时应 先考虑的 法。

show i status

命令能 从路由 否和 IS 交换机正 通信，还 了为每 口分配 的 DN

交换机的类型以及 SPID 的状态和各层的信息。例 7-48 是 BRI 接口配置正常工作后与 ISDN 交换机相连的例子。

例 7-48 show isdn status 命令的示例

```
Cheech#show isdn status
The current ISDN Switchtype = basic-dms100
ISDN BRI0 interface
  Layer 1 Status:      -Shows that the interface is up and the ISDN
  ACTIVE              -circuit has been plugged into the router
  Layer 2 Status:
    TEI = 104, State = MULTIPLE_FRAME_ESTABLISHED
    TEI = 113, State = MULTIPLE_FRAME_ESTABLISHED
  Spid Status:
    TEI 104, ces = 1, state = 5(init)
    spid1 configured, no LDN, spid1 sent, spid1 valid
    Endpoint ID Info: epsf = 0, usid = 0, tid = B
    TEI 113, ces = 2, state = 5(init)
    spid2 configured, no LDN, spid2 sent, spid2 valid
    Endpoint ID Info: epsf = 0, usid = 1, tid = B
  Layer 3 Status:
    1 Active Layer 3 Call(s) -Shows that a connection has been made to another
    router
    Activated dsl 0 CCBs = 2
    CCB: callid=0x0, sapi=0, ces=1, B-chan=0
    CCB: callid=0x802A, sapi=0, ces=1, B-chan=1
    Total Allocated ISDN CCBs = 2
```

要使路由器接口和 ISDN 交换机正确连接，根据路由器型号以及 IOS 版本的不同，可能需要重新启动 BRI 接口。为了证实这一点，参考例 7-49，在装有 Cisco IOS 11.2 (20) 的 Cisco 2503 上输入交换机类型和 SPID 信息之后，查看结果。

例 7-49 验证 ISDN 第 2 层的状态

```
ISDN_Router#show isdn status
The current ISDN Switchtype = basic-dms100
ISDN BRI0 interface
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
    TEI = 88, State = MULTIPLE_FRAME_ESTABLISHED
  Spid Status:
    TEI 88, ces = 1, state = 5(init)
    spid1 configured, no LDN, spid1 sent, spid1 valid
    Endpoint ID Info: epsf = 0, usid = 0, tid = B
    TEI Not Assigned, ces = 2, state = 1(terminal down)
    spid2 configured, no LDN, spid2 NOT sent, spid2 NOT valid
  Layer 3 Status:
    0 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 0
    Total Allocated ISDN CCBs = 0
```

可见，系统接受 SPID 1，但拒绝 SPID 2。证实 SPID 信息确实正确之后，用 **shutdown/no shutdown** 命令重新启动 BRI 接口，结果如例 7-50 所示：

例 7-50 重启 BRI 接口

```

ISDN_Router(config)#int bri0
ISDN_Router(config-if)#shut
ISDN_Router(config-if)#no shut
ISDN_Router(config-if)#
%LINK-5-CHANGED: Interface BRI0, changed state to administratively down
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:2, changed state to down
%LINK-3-UPDOWN: Interface BRI0, changed state to up
ISDN_Router(config-if)#end
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 88 changed to up
%ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 97 changed to up
%SYS-5-CONFIG_I: Configured from console by console
Blue-R8#show isdn status
The current ISDN Switchtype = basic-dms100
ISDN BRI0 interface
  Layer 1 Status:
    ACTIVE
  Layer 2 Status:
    TEI = 88, State = MULTIPLE_FRAME_ESTABLISHED
    TEI = 97, State = MULTIPLE_FRAME_ESTABLISHED
  Spid Status:
    TEI 88, ces = 1, state = 5(init)
      spid1 configured, no LDN, spid1 sent, spid1 valid
      Endpoint ID Info: epsf = 0, usid = 0, tid = B
    TEI 97, ces = 2, state = 5(init)
      spid2 configured, no LDN, spid2 sent, spid2 valid
      Endpoint ID Info: epsf = 0, usid = 1, tid = B
  Layer 3 Status:
    0 Active Layer 3 Call(s)
  Activated dsl 0 CCBs = 1
    CCB: callid=0x0, sapi=0, ces=1, B-chan=0
  Total Allocated ISDN CCBs = 1

```

接口复位后，两个 SPID 都可以和 ISDN 交换机进行正常通信。验证与 ISDN 供应商交换机之间的连接时首先考虑的是 **show isdn status** 命令。如果复位之后 SPID 仍然无效，断电重启路由器。实践证明，在其他方法失败的情况下，这样做能够从交换机获得正确的信号。

2. show interface bri 0 命令

这条命令可以显示 ISDN 接口的一般状态。通常情况下，ISDN 线路和接口相连并且启动之后，会进入电子欺骗状态。在电子欺骗状态下，IOS 使接口伪装成工作状态，这样路由表可以指向该接口进行数据传输。例 7-51 给出了该命令的示例。

例 7-51 检查 ISDN 接口的状态

```

Cheech#show int bri0
BRI0 is up, line protocol is up (spoofing)
Hardware is BRI
Internet address is 175.10.23.1/30
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set
Last input 00:00:01, output 00:00:01, output hang never
Last clearing of "show interface" counters 00:00:02

```

(待续)

```
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  1 packets input, 4 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  1 packets output, 4 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
```

可以使用 **show interface bri0 1 2** 命令查看每个 B 信道的状态。见例 7-52。

例 7-52 检查 ISDN 接口上每个 B 信道的状态

```
Cheech#show interface bri0 1 2
BRI0:1 is down, line protocol is down
Hardware is BRI
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set, keepalive set (10 sec)
LCP Closed, multilink Closed
Closed: IPCP, CDPCP
Last input 00:00:07, output 00:00:05, output hang never
Last clearing of "show interface" counters 00:00:49
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  9 packets input, 144 bytes, 0 no buffer
Received 9 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  10 packets output, 152 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  1 carrier transitions
BRI0:2 is down, line protocol is down
Hardware is BRI
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set, keepalive set (10 sec)
LCP Closed, multilink Closed
Closed: IPCP, CDPCP
Last input 00:10:44, output 00:10:44, output hang never
Last clearing of "show interface" counters 00:00:53
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
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 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
```

该例子中，两个 B 信道都不在工作状态中（ISDN 线路处于空闲状态）。

3. show dialer 命令

该命令能够确定所连接的 B 信道、目的电话号码以及到呼叫撤除的剩余时间，还能显示每个呼叫建立的原因（只在呼叫路由器中）。例 7-53 是该命令的示例。

例 7-53 show dialer 命令的示例

```
Cheech#show dialer

BRI0 - dialer type = ISDN

Dial String      Successes  Failures  Last called  Last status
6129319833       31         0        00:00:15    successful
0 incoming call(s) have been screened.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is physical layer up
Dial reason: ip (s=175.10.23.1, d=175.10.23.2)
Time until disconnect 104 secs
Connected to 6129319833 (Chong)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is physical layer up
Dial reason: Multilink bundle overloaded
Time until disconnect 104 secs
Connected to 6129319833 (Chong)
```

这个例子中，两个 B 信道都处在工作状态之中，都和一个叫 Chong 的路由器连在一起。第一个呼叫是由一个从 175.10.23.1 到 175.10.23.2（在拨号列表中）的 IP 数据包引起的。第 2 个 B 信道是用多链路捆绑造成的（此 BRI 0 接口上配置了 PPP 多链路）。

4. show isdn active 命令

如例 7-54 所示，该命令用来显示当前处于工作状态的呼叫数目，以及拨叫的号码和断开连接前的剩余空闲时间。

例 7-54 show isdn active 命令的示例

```
ISDN-1#show isdn active

-----
ISDN ACTIVE CALLS
-----

History table has a maximum of 100 entries.
History table data is retained for a maximum of 15 Minutes.
-----

Call    Calling    Called    Remote    Seconds  Seconds  Seconds  Charges
Type    Number     Number    Name      Used     Left     Idle     Units/Currency
-----
```

（待续）

Out	6129319833	ISDN-2	6	0	0

7.4.2 ISDN 的 “Big D”

这里是一些常用的确定 ISDN 的故障原因的调试命令。最常见的故障包括无法建立呼叫连接，呼叫无法挂断以及拨号接口反复的连接、挂断、再连接等。

1. debug isdn q.921 命令

我个人认为这条 **debug** 命令除了调试一些 SPID 问题外并不是特别有用。**debug isdn q.921** 命令可以显示第 2 层的工作情况。例 7-55 是 SPID 配置不正确时的情况。

例 7-55 debug q.921 命令的示例

```
debug isdn q921
19:27:31: TX -> IDREQ ri = 19354 ai = 127 dsl = 0
19:27:33: TX -> IDREQ ri = 1339 ai = 127 dsl = 0
19:27:35: TX -> IDREQ ri = 22764 ai = 127 dsl = 0
19:27:37: TX -> IDREQ ri = 59309 ai = 127 dsl = 0
19:27:39: TX -> IDREQ ri = 25214 ai = 127 dsl = 0
19:27:41: TX -> IDREQ ri = 35423 ai = 127 dsl = 0
19:27:43: TX -> IDREQ ri = 12368 ai = 127 dsl = 0
19:27:45: TX -> IDREQ ri = 13649 ai = 127 dsl = 0
19:27:47: TX -> IDREQ ri = 35426 ai = 127 dsl = 0
19:27:49: TX -> IDREQ ri = 12419 ai = 127 dsl = 0
19:27:51: TX -> IDREQ ri = 14516 ai = 127 dsl = 0
19:28:04: TX -> IDREQ ri = 50165 ai = 127 dsl = 0
19:28:06: TX -> IDREQ ri = 838 ai = 127 dsl = 0
19:28:08: TX -> IDREQ ri = 14247 ai = 127 dsl = 0
19:28:34: TX -> IDREQ ri = 45592 ai = 127 dsl = 0
19:28:36: TX -> IDREQ ri = 54169 ai = 127 dsl = 0
19:28:38: TX -> IDREQ ri = 3370 ai = 127 dsl = 0
19:29:09: TX -> IDREQ ri = 57291 ai = 127 dsl = 0
19:29:11: TX -> IDREQ ri = 56444 ai = 127 dsl = 0
19:29:13: TX -> IDREQ ri = 42045 ai = 127 dsl = 0
19:29:44: TX -> IDREQ ri = 59406 ai = 127 dsl = 0
19:29:46: TX -> IDREQ ri = 26863 ai = 127 dsl = 0
19:29:48: TX -> IDREQ ri = 63456 ai = 127 dsl = 0
19:30:19: TX -> IDREQ ri = 30177 ai = 127 dsl = 0
19:30:21: TX -> IDREQ ri = 54258 ai = 127 dsl = 0
19:30:23: TX -> IDREQ ri = 4883 ai = 127 dsl = 0
19:30:54: TX -> IDREQ ri = 17476 ai = 127 dsl = 0
19:30:56: TX -> IDREQ ri = 34949 ai = 127 dsl = 0
19:30:58: TX -> IDREQ ri = 4310 ai = 127 dsl = 0
19:31:24: TX -> IDREQ ri = 7735 ai = 127 dsl = 0
19:31:26: TX -> IDREQ ri = 424 ai = 127 dsl = 0
```

路由器向 ISDN 交换机送出了身份识别请求 (IDREQ)，但是没有收到应答。如果路由器中的 SPID 配置不正确，该 **debug** 命令会显示 SPID 请求被拒绝。

2. debug isdn events 命令

检查呼入呼出的 ISDN 业务状态时调试 Q.931 非常有用。这条 **debug** 命令能够显示呼叫

例 7-56 debug isdn events 命令的示例

```

BRI0: Dialing cause: BRI0: ip PERMIT
BRI0: Attempting to dial 6968900 TX -> SETUP dsl = 0 pd = 8 callref = 0x01 Bearer
Capability i = 0x8890218F Channel ID i = 0x83
Called Party Number i = 0x80, '6968900' RX RELEASE dsl = 0 pd = 8 callref = 0x01 RX
Router#

```

由此可见，用户所期望的数据试图建立一个 ISDN 连接，但是远程路由器没有响应（调试信息 RX RELEASE 表明了这一点）。该问题通常由配置不正确的拨叫号码或 SPID 引起。上个例子中是拨叫号码有误引起的。

3. debug dialer 命令

这可能是现有命令中最简单也是最重要的 **debug** 命令。它能让用户深入地查询很多信息，包括呼叫建立的原因，远端路由器有否响应，以及呼叫失败的原因等。例 7-57 是路由器配置中没有 **dialer map** 命令和 **dialer string** 命令时 **debug dialer** 命令的示例。

例 7-57 debug dialer 命令的示例

```

ISDN-1#debug dialer
1w1d: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=175.10.23.2)
1w1d: BR0 DDR: No dialer string, dialing cannot occur
1w1d: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=175.10.23.2)
1w1d: BR0 DDR: No dialer string, dialing cannot occur
1w1d: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=175.10.23.2)
1w1d: BR0 DDR: No dialer string, dialing cannot occur
1w1d: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=175.10.23.2)
1w1d: BR0 DDR: No dialer string, dialing cannot occur
1w1d: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=175.10.23.2)
1w1d: BR0 DDR: No dialer string, dialing cannot occur

```

与其他 **debug** 命令的执行结果不同，**debug dialer** 命令给出了与呼叫失败有关的一些直观信息。

其中最有用的信息是呼叫建立的原因，它显示了哪些数据包通过拨号列表产生呼出业务。例如，尽管有限制性的拨号列表在起作用，但 ISDN 线路仍然始终处于连接状态。例 7-58 就是这种情况下，**debug dialer** 命令的示例。

例 7-58 debug dialer 命令的示例

```

ISDN-1#debug dialer
1w1d: BR0 DDR: Dialing cause ip (s=175.10.23.1, d=224.0.0.10)
1w1d: BR0 DDR: Attempting to dial 6129319833
1w1d: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
1w1d: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 6129319833
1w1d: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
1w1d: Vi1 DDR: dialer protocol up

1w1d: %SYS-5-CONFIG_I: Configured from console by console
1w1d: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
1w1d: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed
state to up

```


示例中的突出显示部分显示了产生该 ISDN 呼叫的原因，是源地址 175.10.23.1 的一个 IP 数据包传输到了目的地址 224.0.0.10 处。如果看过相关内容就会知道 224.0.0.10 是 EIGRP 用来发布路由信息的一个多播地址。因此，EIGRP 更新信息触发了此次呼叫。要杜绝这种情况的发生，可以将 BRI 接口在 EIGRP 中设置为被动接口，或者是在拨号列表中拒绝所有的 EIGRP 数据包。debug dialer 命令有一些扩充命令，包括 debug dialer packets 命令和 debug dialer events 命令。命令 debug dialer packets 能够提供有关用户期望或不期望的数据的较为深入信息，而命令 debug dialer events 则能够提供与呼叫建立和撤除过程相关的附加信息。

4. debug ppp authentication 命令

例 7-59 反映的情况是 BRI 接口反复地建立和断开连接，但没有数据在链路中传输。

例 7-59 认证失败的症状

```
Cheech#ping 175.10.23.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 175.10.23.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Cheech#
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
Cheech#
```

产生这种症状最可能的原因是没有通过 PPP 认证。例 7-60 表明了这种问题情况下 debug ppp authentication 命令的输出结果。

例 7-60 PPP 认证的调试过程

```
Cheech#debug ppp authentication
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
BR0:1 PPP: Treating connection as a callout
BR0:1 PPP: Phase is AUTHENTICATING, by both
BR0:1 CHAP: O CHALLENGE id 17 len 27 from "Cheech"
BR0:1 CHAP: I CHALLENGE id 17 len 26 from "Chong"
BR0:1 CHAP: O RESPONSE id 17 len 27 from "Cheech"
BR0:1 CHAP: I FAILURE id 17 len 21 msg is "MD compare failed"
%LINK-3-UPDOWN: Interface BRI0:1, changed state to down
```

可以看出，MD 的比较失败，这表明其中一个路由器上设置的用户名和密码不匹配。结

果发现，Cheech 的配置中有如下的配置：

```
username Chong password cisco
```

而在 Chong 上则是：

```
username Chong password Cisco
```

由此可见，密码必须完全匹配，而且要区分大小写。

7.5 技巧和窍门

本节讲述有助于正确配置 ISDN 网络的方法。尽管这些方法并不能适用于所有的 ISDN 网络，但是它们确实能够解决很多常见的配置问题。

- 如果只需一台路由器启动 ISDN 呼叫，可以删除被叫路由器上所有的拨号映射或拨号字符串。被叫路由器会动态地获取下一跳地址以及拨号信息。
- 对于被叫路由器来说，可以将拨号空闲超时时间增加到一个很大的值（如 9999），使得只有呼叫路由器才能终止连接着的呼叫业务。例如，如果希望 ISDN 链路能够在呼叫业务开始之后起码保持 5 分钟，那么在呼叫路由器上将拨号空闲超时时间设置为 300 秒。但是，如果没有调整被叫路由器的空闲超时时间的值，那么每次呼叫都会在 120 秒（默认值）后终止，因为被叫路由器会认为没有用户所期望的数据通过了链路，于是断开链路。要避免这种情况的发生，可在被叫路由器上将其拨号空闲超时时间的值设得非常大。
- 不要忘记在声明 **dialer map** 中加入关键字 **broadcast**。如果路由信息通过 ISDN 链路传输时有问题，很可能是 **dialer map** 声明中没有加入 **broadcast** 关键字。
- 如果发现链路无限期地保持工作状态，或者是发现刚断开之后立即又开始建立连接，就需要增加拨号列表的限制条件，要确保 ISDN 的接口在路由进程中设置为被动。
- 确保没有产生由环路引起路由器不停拨号的情况，使用 OSPF 按需电路特性时尤其应该注意。可以通过在路由进程中加入分配列表或路由映射（Route Map）来实现。
- 配置时要灵活处理。通常有很多途径来实现同一目标，配置 ISDN 也可以有很多种方法。
- 对 PPP 认证来说，注意要正确设置用户名和密码（记住它们区分大小写），确保每台路由器密码匹配。
- 对 OSPF 按需电路的配置，记得增加 OSPF 的 ISDN 链路开销值，以确保首选的路径是主链路，还要避免网络中出现环路而导致不必要的拨号出现。
- 配置可选参数，如认证模式，呼叫者身份等。在配置高级选项之前，要确保 ISDN 已经正常工作。完成 ISDN 的每一步配置任务之后，用相应的 **show** 命令和 **debug** 命令验证网络的工作情况。配置路由器时应该逐步进行，不要一次配置所有内容，否则出现问题时很难找到问题所在。

通过实验找到最合适的方法，这或许会成为每个人的默认方法，但是最好多尝试几种方法，了解问题的关键所在，这样才能处理各种不同的情况。

7.6 ISDN 实验

我们将上面所学的所有内容都放到下面一系列的实验中，这些实验与 CCIE 中路由与交换技术实验考试非常相似。这里给出了一套实验要求，大家可以根据要求自己确定对路由器的配置。这里也提供了实验的解决方案。如果大家愿意在看完本书后面部分，了解更多的路由选择协议和可路由选择协议内容之后再来完成这些实验，效果会更好。

7.7 实验 16：配置 ISDN 上的 PPP 认证、回拨和多链路连接

网络 ABC 的拓扑结构如图 7-11 所示，主通信链路是帧中继，其 ISDN 电路只是在特定的要求下才使用。该网络的 CEO 要求网络设计要满足一些下列条件：

1 路由选择协议采用 OSPF，如图 7-11 所示。

2 即使帧中继链路出现故障也要保证各个节点之间的连接。只有在帧中继链路出现物理故障或者主链路的利用率超过了 50% 的时候，ISDN 链路才开始工作。ISDN 呼叫链路在帧中继链路由于出现问题而终止工作之后等待 10 秒进入工作状态，而在帧中继链路恢复之后，或者是帧中继链路的负荷降到 25% 以后，ISDN 链路再等待 2 分钟之后才断开连接。在帧中继接口上只可以使用一个 PVC。

3 连接一旦建立，两个 ISDN B 信道都要同时开始工作。

4 采用认证方式，但是 ISDN 链路上的密码不需要加密封装。

5 链路两端的路由器要都能够发起 ISDN 呼叫。

6 不使用静态路由。

图 7-11 是这个实验的网络拓扑结构图。

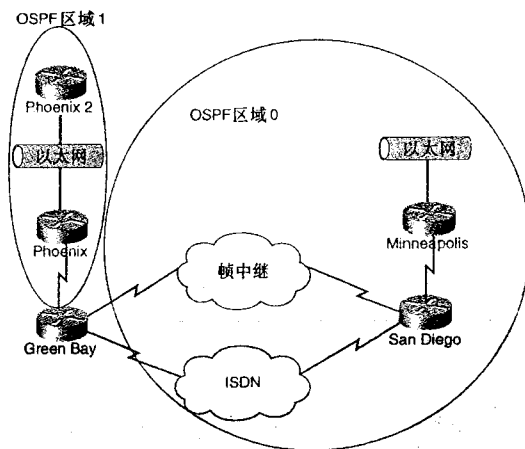


图 7-11 本实验的网络拓扑结构

7.7.1 实验 16 的解决方案

例 7-61 是该实验的完整解决方案，后面还将对其进行讨论。XJFXJF

例 7-61 路由器 Phoenix2, Phoenix, Green Bay, San Diego 和 Minneapolis 的配置

```
Phoenix2#show running-config
Building configuration...

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Phoenix2
!
enable password cisco
!
!
interface Ethernet0
 ip address 170.10.35.2 255.255.255.0
!
interface Ethernet1
 no ip address
 shutdown
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
!
router ospf 1
 network 170.10.0.0 0.0.255.255 area 1
!
no ip classless
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 password cisco
 login
!
end
```

```
Phoenix#show running-config
Building configuration...
```

Current configuration:

(待续)

```

!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Phoenix
!
logging buffered 9096 debugging
!
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
 ip address 170.10.35.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Serial1
 description POINT TO POINT LINK TO GREEN BAY
 bandwidth 64
 ip address 170.10.23.2 255.255.255.252
 no ip directed-broadcast
 clockrate 125000
!
router ospf 1
 network 170.10.0.0 0.0.255.255 area 1
!
no ip classless
!
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

```

```

Green_Bay#show running-config
Building configuration...

```

```

Current configuration:

```

```

!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Green_Bay
!
!
username San_Diego password 0 isdnlab

```

(待续)

```

ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
 ip address 170.10.22.1 255.255.255.0
 no ip directed-broadcast
 no keepalive
!
interface Serial0
 backup delay 10 120
 backup interface BRI0
 backup load 50 25
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 logging event subif-link-status
 logging event dlci-status-change
 no fair-queue
 clockrate 125000
!
interface Serial0.1 point-to-point
 ip address 170.10.29.1 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 300
!
interface Serial1
 description POINT TO POINT LINK TO PHOENIX
 bandwidth 64
 ip address 170.10.23.1 255.255.255.252
 no ip directed-broadcast
!
interface BRI0
 ip address 170.10.129.1 255.255.255.252
 no ip directed-broadcast
 encapsulation ppp
 dialer map ip 170.10.129.2 name San_Diego broadcast 6129319360
 dialer load-threshold 1 either
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 ppp authentication pap
 ppp pap sent-username Green_Bay password 7 141E010F02082B29
 ppp multilink
!
router ospf 1
 network 170.10.22.1 0.0.0.0 area 1
 network 170.10.23.1 0.0.0.0 area 1
 network 170.10.29.1 0.0.0.0 area 0
 network 170.10.129.1 0.0.0.0 area 0
!
no ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none

```

（待续）

```
line aux 0
line vty 0 4
  login
!
end
```

```
San_Diego#show running-config
Building configuration...
```

```
Current configuration:
```

```
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname San_Diego
!
enable password cisco
!
username Green_Bay password 0 isdnlab
ip subnet-zero
no ip domain-lookup
isdn switch-type basic-dms100
!
interface Ethernet0
  no ip address
  no keepalive
  media-type 10BaseT
!
interface Serial0
  backup delay 10 120
  backup interface BRI0
  backup load 50 25
  no ip address
  encapsulation frame-relay
!
interface Serial0.1 point-to-point
  ip address 170.10.29.2 255.255.255.252
  frame-relay interface-dlci 200
!
interface Serial1
  ip address 170.10.49.2 255.255.255.252
  clockrate 125000
!
!
interface BRI0
  ip address 170.10.129.2 255.255.255.252
  encapsulation ppp
  isdn spid1 61293193601111
  isdn spid2 61293197761111
  dialer map ip 170.10.129.1 name Green_Bay broadcast 6129319937
  dialer load-threshold 1 either
  dialer-group 1
  no fair-queue
  ppp authentication pap
  ppp pap sent-username San_Diego password 7 09455D0D17091610
  ppp multilink
!
interface BRI1
  no ip address
  shutdown
```

(待续)

```
!  
interface BRI2  
  no ip address  
  shutdown  
!  
interface BRI3  
  no ip address  
  shutdown  
!  
router ospf 1  
  network 170.10.0.0 0.0.255.255 area 0  
!  
ip classless  
!  
dialer-list 1 protocol ip permit  
!  
line con 0  
  exec-timeout 0 0  
  privilege level 15  
  logging synchronous  
line aux 0  
line vty 0 4  
  exec-timeout 0 0  
  privilege level 15  
  logging synchronous  
  no login  
!  
end
```

```
Minneapolis#show running-config  
Building configuration...
```

```
Current configuration:  
!  
version 11.2  
no service password-encryption  
no service udp-small-servers  
no service tcp-small-servers  
!  
hostname Minneapolis  
!  
enable password cisco  
!  
!  
interface Ethernet0  
  ip address 170.10.44.1 255.255.255.0  
  no keepalive  
!  
interface Serial0  
  no ip address  
  shutdown  
  no fair-queue  
!  
interface Serial1  
  ip address 170.10.49.1 255.255.255.252  
!  
router ospf 1  
  network 170.10.0.0 0.0.255.255 area 0  
!  
no ip classless  
!
```

（待续）


```
!  
line con 0  
  privilege level 15  
  logging synchronous  
line aux 0  
line vty 0 4  
  privilege level 15  
  password cisco  
  no login  
!  
end
```

7.7.2 实验 16 解决方案的讨论

这里侧重讨论 Green Bay 和 San Diego 路由器的配置，其他 3 台的配置非常直接明了。

由于该实验中要求 ISDN 链路在帧中继链路出现物理故障之后才进入工作状态，因此 **backup interface** 命令是最为适当的配置方法。另外，实验要求帧中继链路出现拥塞时 ISDN 也要投入使用，因而还使用了 **backup load** 命令。PPP 的使用和拨号呼叫负载阈值设为 1 满足了第 3 个条件的要求。使用 PAP 认证方式符合第 4 个条件的要求。最后，两台路由器在配置中都使用了拨号映射，这是对第 5 个条件的满足。这个实验并没有特别要求使用拨号映射，因此拨号字符串或者是 dialer profile 的设置都是越简单越好。

7.8 实验 17：配置 ISDN 上 OSPF 按需电路

这个实验是假设大家对各种路由选择协议都已经有了一个比较深入的认识。因此大家可以选择先阅读一下第 11 章“混合协议：增强型内部网关路由选择协议（EIGRP）”和第 12 章的内容，然后再来完成这个实验。在做这个实验的时候，最好是独立完成，尽量多去查一查相关的命令，自己想出一个方法来，然后才去看看我们提供的解决方案。

先请参考一下图 7-12 的网络图，作为参考，我们姑且把这个网络称为 ABC 公司。

对这个 ABC 网络的设计，有下面这样一些要求：

1 按照上面图中的结构配置 OSPF 和 EIGRP，记得要在协议之间进行再分布，以保证完整的 IP 连接。

2 56-kbit/s 的帧中继链路是 Green Bay 和 San Diego 之间通信的主链路。不希望 OSPF 数据能启动 ISDN 呼叫，但是希望保证整个网络的 IP 互通，即使是出现帧中继故障也是这样。ISDN 呼叫的建立应该是由路由的丢失触发。

3 两台路由器都可以发起呼叫。从 Green Bay 发起的呼叫的费用要便宜一些，因此要确保绝大多数的 ISDN 呼叫都是从 Green Bay 处发起。

4 每一次 ISDN 链路都要求通过认证，而且要用 MD5 加密。

5 Green Bay 路由器只能接受来自 San Diego 路由器的呼入业务。

6 ISDN 呼叫在帧中继链路恢复 5 分钟之后必须断开。

7 帧中继链路只是 56 k 的，确保配置中能够反映这一点。

8 这个实验中不需要使用任何静态路由。

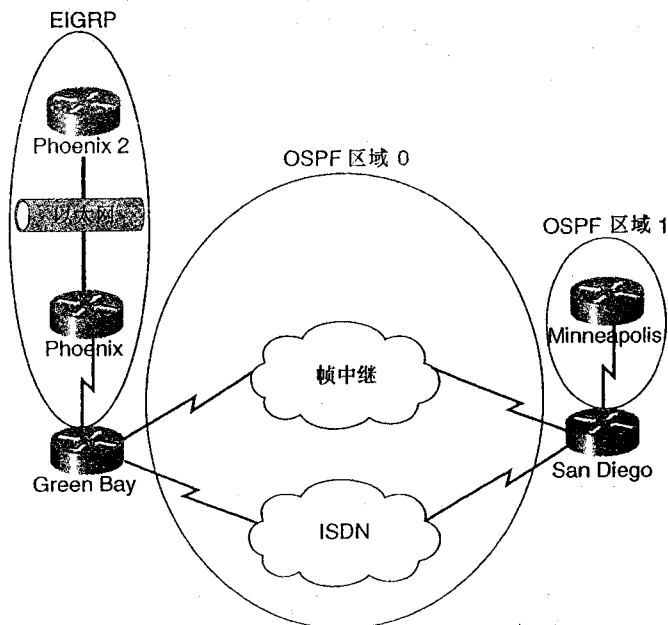


图 7-12 本实验的网络拓扑图

7.8.1 实验 17 的解决方案

例 7-62 就是能够满足上述要求的配置过程。

例 7-62 Phoenix 2、Phoenix、Green Bay、San Diego 和 Minneapolis 路由器的配置

```
Phoenix2#show running-config
Building configuration...

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Phoenix2
!
enable password cisco
!
!
interface Ethernet0
 ip address 170.10.35.2 255.255.255.0
!
interface Ethernet1
 no ip address
 shutdown
!
interface Serial0
 no ip address
 shutdown
```

(待续)

```

no fair-queue
!
interface Serial1
no ip address
shutdown
!
router eigrp 1
network 170.10.0.0
no auto-summary
!
no ip classless
!
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
line aux 0
line vty 0 4
password cisco
login
!
end

```

```

Phoenix#show running-config
Building configuration...

```

```

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Phoenix
!
logging buffered 9096 debugging
!
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
ip address 170.10.35.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
no ip address
no ip directed-broadcast
shutdown
!
interface Serial1
description POINT TO POINT LINK TO GREEN BAY
bandwidth 64
ip address 170.10.23.2 255.255.255.252
no ip directed-broadcast
clockrate 125000
!
router eigrp 1
network 170.10.0.0
no auto-summary

```

```

!
no ip classless
!
!
!
line con 0
  privilege level 15
  logging synchronous
  transport input none
line aux 0
line vty 0 4
  login
!
end

Green_Bay#show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Green_Bay
!
!
username San_Diego password 0 isdnlab
ip subnet-zero
isdn switch-type basic-dms100
!
!
!
interface Ethernet0
  ip address 170.10.22.1 255.255.255.0
  no ip directed-broadcast
  no keepalive
!
interface Serial0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay
  logging event subif-link-status
  logging event dlci-status-change
  no fair-queue
  clockrate 125000
!
interface Serial0.1 point-to-point
  description 56K FRAME RELAY CONNECTION
  bandwidth 56
  ip address 170.10.29.1 255.255.255.252
  no ip directed-broadcast
  frame-relay interface-dlci 300
!
interface Serial1
  description POINT TO POINT LINK TO PHOENIX
  bandwidth 64
  ip address 170.10.23.1 255.255.255.252
  no ip directed-broadcast
!

```

```

interface BRI0
 ip address 170.10.129.1 255.255.255.252
 no ip directed-broadcast
 encapsulation ppp
 ip ospf cost 9999
 ip ospf demand-circuit
 dialer callback-secure
 dialer idle-timeout 300
 dialer enable-timeout 5
 dialer map ip 170.10.129.2 name San_Diego class isdnlab broadcast 6129319360
 dialer-group 1
 isdn switch-type basic-dms100
 isdn spid1 61293199371111
 isdn spid2 61293199381111
 isdn caller 6129319360 callback
 ppp callback accept
 ppp authentication chap
!
router eigrp 1
 redistribute ospf 1 metric 64 100 200 10 1500
 passive-interface BRI0
 passive-interface Serial0.1
 network 170.10.0.0
 no auto-summary
!
router ospf 1
 redistribute eigrp 1 metric 100 subnets route-map DENY_BRI_ROUTE
 network 170.10.29.1 0.0.0.0 area 0
 network 170.10.129.1 0.0.0.0 area 0
 passive-interface Serial1
!
no ip classless
!
!
map-class dialer isdnlab
 dialer callback-server username
access-list 1 permit 170.10.129.0 0.0.0.3
dialer-list 1 protocol ip permit
route-map DENY_BRI_ROUTE deny 10
 match ip address 1
!
route-map DENY_BRI_ROUTE permit 20
!
!
!
line con 0
 privilege level 15
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 login
!
end

```

```

San_Diego#show running-config
Building configuration...

```

```

Current configuration:
!
version 11.2

```

```
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname San_Diego
!
enable password cisco
!
username Green_Bay password 0 isdnlab
ip subnet-zero
no ip domain-lookup
isdn switch-type basic-dms100
!
interface Ethernet0
no ip address
no keepalive
media-type 10BaseT
!
interface Ethernet1
no ip address
media-type 10BaseT
!
interface Ethernet2
no ip address
shutdown
media-type 10BaseT
!
interface Ethernet3
no ip address
shutdown
media-type 10BaseT
!
interface Ethernet4
no ip address
shutdown
media-type 10BaseT
!
interface Ethernet5
no ip address
shutdown
media-type 10BaseT
!
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.1 point-to-point
description 56K FRAME RELAY CONNECTION
ip address 170.10.29.2 255.255.255.252
bandwidth 56
frame-relay interface-dlci 200
!
interface Serial1
ip address 170.10.49.2 255.255.255.252
clockrate 125000
!
interface Serial2
no ip address
shutdown
!
interface Serial3
no ip address
```

(待续)

```

shutdown
!
interface BRI0
 ip address 170.10.129.2 255.255.255.252
 encapsulation ppp
 ip ospf cost 9999
 ip ospf demand-circuit
 isdn spid1 61293193601111
 isdn spid2 61293197761111
 dialer idle-timeout 300
 dialer wait-for-carrier-time 10
 dialer map ip 170.10.129.1 name Green_Bay broadcast 6129319937
 dialer-group 1
 no fair-queue
 ppp callback request
 ppp authentication chap
!
interface BRI1
 no ip address
 shutdown
!
interface BRI2
 no ip address
 shutdown
!
interface BRI3
 no ip address
 shutdown
!
router ospf 1
 network 170.10.29.2 0.0.0.0 area 0
 network 170.10.129.2 0.0.0.0 area 0
 network 170.10.49.2 0.0.0.0 area 1
!
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 no login
!
end

```

```

Minneapolis#show running-config
Building configuration...

```

```

Current configuration:

```

```

!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers

```

(续表)

```

!
hostname Minneapolis
!
enable password cisco
!
!
interface Ethernet0
 ip address 170.10.44.1 255.255.255.0
 no keepalive
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 ip address 170.10.49.1 255.255.255.252
!
router ospf 1
 network 170.10.0.0 0.0.255.255 area 1
!
no ip classless
!
!
line con 0
 privilege level 15
 logging synchronous
line aux 0
line vty 0 4
 privilege level 15
 password cisco
 no login
!
end

```

7.8.2 实验 17 解决方案的讨论

同样，Phoenix, Phoenix 2 和 Minneapolis 的配置也没有什么特别之处，它们在这里的作用只是产生路由而已。配置命令中大部分都是 Green Bay 路由器用到的。

这个实验中使用了 OSPF 按需电路，因为按照要求，在帧中继链路发生了路由丢失的情况的时候，ISDN 链路才会启用。如果要求帧中继链路出现连接的物理丢失的情况才启用 ISDN 链路，那用 **backup interface** 命令进行配置就会更恰当一些了。

为了满足第 3 个要求，我们把 Green Bay 路由器设置成了一台回拨路由器。第 4 个要求启用 MD5 加密的认证方式，因此还需要配置 CHAP。此外，我们还利用了 ISDN 呼叫者 ID 的设置来满足第 5 个要求。Green Bay 和 San Diego 都使用了 **dialer map** 命令，因此二者都能够发起 ISDN 呼叫。第 6 个条件的满足是通过在 ISDN 的连接两端把拨号空闲超时设为 300 秒来实现的。

这个实验最大的问题就是如何在网络稳定的情况下使 ISDN 链路保持空闲状态。如果能够在不看我们提供解决方案的情况下自己想出来，就非常了不起了。

配置 Green Bay 路由器的时候很重要的一点就是在利用 **route map** 命令把 EIGRP 再分布到 OSPF 的时候。OSPF 按需电路那一节讨论过，这个时候我们需要做的有时不仅仅是简单的执行一条 **ip ospf demand-circuit** 命令就可以的。这个实验就是一个很好的例子。在路由器

Green Bay 上把 EIGRP 再分布进 OSPF 的时候, 由于 BRI 接口(170.10.129.1)也包含在 EIGRP 网络中, 这样就产生了一个路由环路。因而, 这个接口会不停地从 OSPF 再分布进 EIGRP, 同时也不停地从 EIGRP 再分布进 OSPF。为了解决这一问题, 需要创建路由图来拒绝把 170.10.129.0/30 网络再分布进 OSPF。另外一个办法是采用分配列表来完成同样的工作。

另外比较重要的一点就是在 Green Bay 和 San Diego 路由器的 BRI 接口上使用 **ip ospf cost 9999** 命令。第 7 个要求基本上是需要要在两台路由器的接口 s0.1 上配置一条 **bandwidth 56** 命令。因此, OSPF 会由于 BRI 连接的路由成本更低而倾向于使用这条链路。这一点可以通过将 BRI 接口的 OSPF 路由成本增加到一个很高的数值来解决。当然, 我们也可以在接口 s0.1 上利用 **ip ospf cost** 命令来降低其成本值。

7.9 总 结

ISDN 是一个广泛而且综合性很强的题目, 而通常又是 CCIE 路由于交换实验考试的主要内容。如果对本章中介绍过的内容没有一个非常深入全面的理解, 想在 CCIE 考试中获取好成绩是很困难的。准备这个苛刻考试的 ISDN 部分的最好办法就是将这些内容付诸实践。如果没有相应的设备可以再三地练习, 那么想要牢固地掌握这些知识也是不大可能的。对于这一点怎么强调也不算过分。如果没有 ISDN 设备, 就需要去购买一台 ISDN 模拟器以及一些具有 BRI 接口的路由器。阅读书中的内容是一回事, 而亲手去练习、去实践从而获得知识又是完全不同的一种体验。

在做 ISDN 的实验的时候, 要尽量使内容变得复杂一些。采用不同的路由选择协议并注意相应的配置变化。往配置中加入 IPX 以及其他路由选择协议并注意会产生什么效果。然后, 试着在 ISDN 链路上配置 BGP、DLSw 以及 NAT 等。在路由器上配置某一个方面的内容相对来说是比较简单的。但是, 当多个内容一起配置, 相互结合的时候, 就有可能出现一些意想不到的情况。知道并解决这些问题的惟一途径就是尽可能多地亲自动手去做, 尽量实施不同种类的实验, 并且把它们融合到一起。

第 8 章

WAN 协议与技术： 异步传输模式 (ATM)

Galina Diker Pildush 供稿

异步传输模式(ATM)技术被认为可以任何事情——可以传输语音、数据和视频信息，意味着语音和数据载荷，批量数据和实时数据都可以从世界的一端传送到另一端。这种方式下，传输质量，包括数据的完整性和吞吐量等，都能够得到有效保证，而所提供的服务也多种多样。《Cisco ATM Solutions》一书详细阐述了与 ATM 有关的问题。ATM 给人的感觉就是其内在的简单和美。

ATM 的原理非常简单。ATM 传输大小相等的有效载荷(ATM 信元包括一个 48 字节的有效载荷和一个 5 字节的包头)，该有效载荷包括所有应用形式。ATM 不对载荷进行错误校验，不浪费任何系统开销进行多余的处理，除了序列号(因为 ATM 是面向连接的)，所做的只是对 ATM 的数据包头进行错误校验(在物理层完成)，所有载荷的传输在 OSI 模型低层中里以尽可能快的速度完成。实际上，ATM 的工作是在 OSI 参考模型的 1.25 个层中完成的——没错，1.25 层！图 8-1 给出了 ATM 信元的结构信息。

表 8-1 则列出了信元包头各个字段的含义。

ATM 功能强大的专用网络——网络接口(PNNI)有助于发展诸有流量工程的 MPLS 的协议，PNNI 将面向连接的概念引入非面向连接的 IP。ATM 信令协议采用 PNNI 路由选择协议创建 SVC。

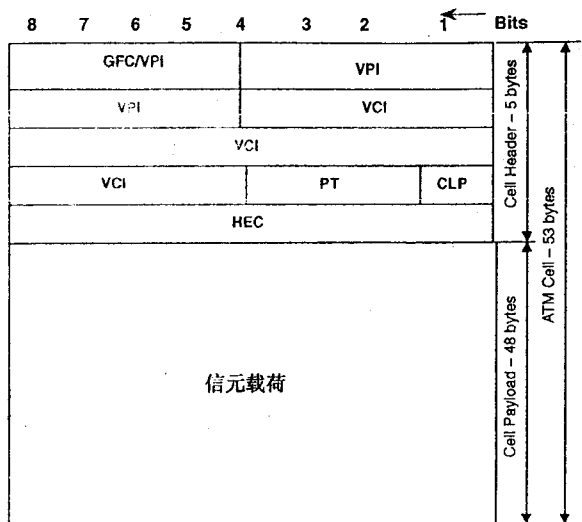


图 8-1 ATM 信元格式

表 8-1

ATM 信元包头字段

字 段	大 小	意 义
GFC	4 比特	一般流控：在 UNI 级别使用
VPI	8 或 12 比特	虚拟路径标识符：ATM 具有本地标识性的地址部分，UNI 为 8 比特，NNI 为 12 比特
VCI	16 比特	虚拟信道标识符：ATM 具有本地标识性的地址部分
PT	3 比特	载荷类型——3 比特使用如下：第一有效比特位，用于标识数据和 OAM 信元，下一比特位用于标识网络拥塞（称为显式前向拥塞标识[EFCI]），最后 1 比特位用于标识高级包/帧的最后信元，称为消息末梢。在 ATM 云的拥塞控制算法中十分重要
CLP	1 比特	信元丢弃优先级：出现拥塞时，标识网络可以丢弃某一信元
HEC	8 比特	包头差错校验：ATM 只进行包头校验而不进行载荷差错校验

认识到 ATM 在客户企业级网络中的地位，Cisco 在路由与交换技术 CCIE 认证考试中加入了 ATM 各种互联协议的内容。本章将讨论 ATM 的两种协议互连方法，其讲述过程与最新设备的使用相互联系：

- RFC 2684，“ATM 适配层 5 的多协议封装”。
- RFC 2225，“ATM 上的经典 IP 和 ARP”。

ATM 的实际应用目前还没有出现在 CCIE 实验考试中。《Cisco ATM Solutions》一书详细讲述了 ATM 的应用问题。

ATM 网络的最终目的是通过 ATM 传输上层网络信息。目前应用的方式有 4 种：

- 采用 RFC 2684，针对多个上层协议的手工方式。该方式称为普通方式，没有超出 OSI 模型上层的内容。
- 采用 RFC 2225（经典的 IP 协议），针对 IP 应用的动态方式。该方式称为能通过 ATM 传输 IP 的有魔力的方式。
- 采用局域网仿真（LANE）用于第 2 层的动态方式。

本章的重点是讲述前两种方式——RFC 2684 和 RFC 2225。

本章的两个实验提供了实践练习的机会。如果还想了解其他连接方式（例如 LANE、MPOA 等）或更多的例子和实验，可以参考《Cisco ATM Solutions》。

8.1 ATM 实验学习所需的特定组件

推荐以下设备用于 ATM 实验：

- 一台 LightStream 1010（LS1010）ATM 交换机。
- 两台或三台具有 ATM 接口的路由器。

如果没有 LS1010，建议进行实验之前可以参加 Cisco 的“Campus ATM Solution”培训课程。该课程有几台 LS1010 以及路由器，可以满足要求。同时，《Cisco ATM Solutions》一书也提供了很多配置实例和实验练习以提高对 ATM 的认识和掌握。

本章侧重路由器的配置。典型的 LS1010 不需要特殊配置，除非有一个多 LS1010 的网络。《Cisco ATM Solutions》一书给出了关于配置 LS1010 的详细例子和解释，如有必要可参考该书。

路由器可为 4XXX、7XXX 或 36XX 系列。Cisco 对具有 ATM 接口的路由器和交换机产品有清楚的分类。表 8-2 给出了分类情况。注意 Cisco 支持的 ATM 适配层（AAL），用户网络接口（UNI）类型和 ATM 服务种类情况。

表 8-2

Cisco ATM 边缘设备小结^①

设备	交换类型	支持的 业务类型	支持的 ATM 接口速率	支持的 AAL	支持的 UNI	支持的 ATM 服务类别
26xx	第 3 层	语音和数据	1×25mbit/s 带 IMA 的 4xDS-1/E1 带 IMA 的 4xDS-1/E1 DS-3/E3	AAL1, AAL2, AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, CBR, nrt-VBR, rt-VBR
36xx	第 3 层	语音和数据	1×25mbit/s; 1×(oc-3)/(SMT-1) 带 IMA 的 4xDS-1/E1 带 IMA 的 4xDS-1/E1 DS-3/E3	AAL1, AAL2, AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, nrt-VBR
3810	第 3 层	语音和数据	1×DS-1/E1	AAL1, AAL2, AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, nrt-VBR
4500	第 3 层	数据	1×OC-3/STM-1	AAL5, AAL3/4	UNI3.0 UNI3.1 UNI4.0	UBR, CBR, nrt-VBR, ABR
4700	第 3 层	数据	1×OC-3/STM-1	AAL5, AAL3/4	UNI3.0 UNI3.1 UNI4.0	UBR, nrt-VBR, ABR

续表

设备	交换类型	支持的 业务类型	支持的 ATM 接口速率	支持的 AAL	支持的 UNI	支持的 ATM 服务类别
6400	第 3 层	语音和数据	1×OC-3/STM-1 1×OC-12/STM-4	AAL1, AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, GFR, nrt-VBR
7100	第 3 层	数据	T3/E3 2×OC-3/STM-1	AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, GFR, nrt-VBR
72xx	第 3 层	语音和数据	带 IMA 的 8×DS-1/E1 1×OC-3/STM-1 1×OC-3/STM-4 DS-3/E3	AAL5, AAL1	UNI3.0 UNI3.1 UNI4.0	UBR, nrt-VBR, ABR
7400	第 3 层	数据	带 IMA 的 8×DS-1/E1 1×DS-3/E3 1×OC-3/STM-1	AAL5, AAL1	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, CBR, nrt-VBR
75XX	第 3 层	数据	1×OC-3/STM-1 1×OC-12/STM-4 1×DS-3/E3 带 IMA 的 8×DS-1/E1	AAL5, AAL3/4	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, CBR, nrt-VBR
76XX	第 3 层	数据	2×OC-12/STM-4	AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, nrt-VBR, ABR
12000	第 3 层	数据	1×OC-3/STM-1 4×OC-3/STM-1	AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, nrt-VBR, ABR,
Catalyst 2900	第 2 层	数据	1×OC-3/STM-1	AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, nrt-VBR, ABR,
Catalyst 3900	第 2 层	数据	1×OC-3/STM-1	AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, nrt-VBR, ABR,
Catalyst 5xxx	第 2 层	语音和数据	25mbit/s DS-1/E1 DS-3/E3 1×OC-3/STM-1 1×OC-12/STM-4	AAL5	UNI3.0 UNI3.1 UNI4.0	UBR, ABR, rt-VBR, nrt-VBR, CBR, GFR
Catalyst 6000	第 2 层	数据	1×OC-12/STM-4	AAL5	UNI3.0 UNI3.1 UNI4.0	

续表

设备	交换类型	支持的 业务类型	支持的 ATM 接口速率	支持的 AAL	支持的 UNI	支持的 ATM 服务类别
Catalyst 85xx	第 2 层 第 3 层 第 1.25 层 ^②	数据和语音	最多 64×DS-1/E1 最多 64×DS3/E3 96×25mbit/s 128×OC-3/STM-1 32×OC-12/STM-4 8×OC-48/STM-16	AAL1, AAL2, AAL5	UNI3.0 UNI3.1 UNI4.0	

表 8-2 摘自《Cisco ATM Solutions》，Galina Diker Pildush, Cisco Press.

① 表 8-2 的信息基于 2001 年 8 月的 Cisco 产品目录。

② Catalyst 8500 系列中的 Catalyst 8540MSR 也可以作为 ATM 交换机，不仅支持新的功能模块，而且也支持现有的 LightStream 1010 模块。

Cisco 提供多个平台所支持的 ATM 路由器接口，如表 8-3 所示。

表 8-3 NPM、AIP、ATM PA-A1 和 ATM PA-A3 的比较

物理接口/特性	NPM	AIP	ATM PA-A1	ATM PA-A3
平台	4500,4700	7000,7500	7200, 基于 VIP2 的 7500	7200,基于 VIP2 的 7500
OC-3MMF 支持	是	是	是	是
OC-3SMF 支持	是	是	是	是
DS-3/E3 支持	是	是	否	否
TAXI 支持	否	是	否	否
UNI 支持	3.0/3.1/4.1	3.0/3.1	3.0/3.1	3.0/3.1/4.0
LANE 支持	是	是	是	是
RFC2684, 2225 支持	是	是	是	是
最大 packet/s (64 字节, 双向)	-	110 000packet/s	150 000packet/s	170 000packet/s
同时 SARs (#包数)	192	256 (最大可达 512)	512	1024
最大 VC 数	1023	2048	2048	4096
AAL 支持	AAL5	AAL3/4, AAL5	AAL5	AAL5
ATM 服务类型支持	UBR, ABR	UBR	UBR	Nrt-VBR,UBR,ABR
业务整形支持	是	是	无	是
OAM 支持	F4, F5	F4, F5 (cisco IOS 11.3 (2) T)	F4, F5 (cisco IOS 11.3 (2)T 和 cisco IOS 11.3 (22) CC)	F4, F5 (cisco IOS 特别版本 11.1 (22) CC 和 12.0)

表 8-3 摘自《Cisco ATM Solutions》，Galina Diker Pildush, Cisco Press.

8.2 RFC 2684 的配置

《Cisco ATM Solutions》一书中讲述了RFC 2684的完整理论。这里需要说明的是，RFC 2684（前身是RFC 1483）是ATM上所有路由或桥接协议的封装方式。对RFC2684的描述如下：

RFC2684是多协议封装方式。在单VC上对多协议（第3层或桥接）的封装通过LLC/SNAP实现，在单VC上对单独协议的封装通过mux实现。ATM支持PVC或SVC，并能同时支持PVC和SVC。

永久虚电路（PVC）是静态定义的路由，而交换式虚电路（SVC）则通过信令动态定义路由。通过本节可以了解PVC和SVC应用背后的原理。

RFC 2684是一个基本的封装方式，没有任何“魔力”。协议地址必须手动地把ATM地址和PVC应用中的VPI/VCI对应或者SVC应用中的NSAP/E.164地址来与ATM地址映射。

下面来看一下PVC和SVC应用的细节问题。

8.2.1 PVC 的配置

ATM网络中，PVC的实施配置是一个静态而乏味的过程。如果配置基于PVC的ATM网络，要确保VPI/VCI号的信息准确。VPI/VCI号的任何一个错误都有可能无法预测的严重后果，很可能使得本来到纽约的连接连到莫斯科。请记住，VPI/VCI号是ATM PVC配置过程中使用的具有本地意义的地址。

为了本章讨论的方便，假设ATM网络已经正确配置完毕。

PVC网络的配置可以采用下面两种方式之一：

- 静态VPI/VCI分配，VPI/VCI号要根据供应商提供的信息手动输入。
- 动态VPI/VCI分配，具备该功能的边缘路由器能够从临近交换机动态获取VPI/VCI的值。

不论哪种方式，VPI/VCI的号都只在本地有效。

注释 VPI/VCI号与帧中继网络中的DLCI很相似，DLCI也是本地有效。其意义适用于接口级别。也就是说，如果不同路由器的ATM物理接口具有完全相同的VPI/VCI号，ATM网络也会将它们视为不同的连接。

静态配置包括全部手动配置。如果ATM的VPI/VCI分配值发生了改变，就需要手动调整路由器的设置。而动态查询确定VPI/VCI数值则显得更为吸引人，因为这样的情况下，所连接的路由器可以从各自临近的交换机动态地获取这些信息。路由器动态发现的PVC以及PVC的数据参数可以在ATM的主接口或者是指定的子接口上进行配置。路由器可以通过使用临时本地管理接口（ILMI）获取PVC的参数信息。

看一下基于PVC的网络配置的两种方式的句法结构和配置实例。

与ATM相连的3台路由器（A、B和C）要处理两个协议——IP和IPX。从OSI模型第3层（网络层）的角度来看，所有的路由器都是通过一个IP网络131.108.168.0/24和IPX网

络 100 互连的。每台路由器都和另外一个 IP 和 IPX 网络相连。

1. 静态 PVC 配置

例 8-1 给出了路由器 A、B 和 C 的配置情况。注意图中的路由器通过其子接口与 ATM 相连。也可以用其主接口进行连接。但是，使用子接口能够使网络更具有可扩展性。

该例子也给出了 LLC/SNAP 封装实例（与多路复用相反）。LLC/SNAP 封装使 VC 能够承载多个协议，而多路复用封装只能承载一个协议。

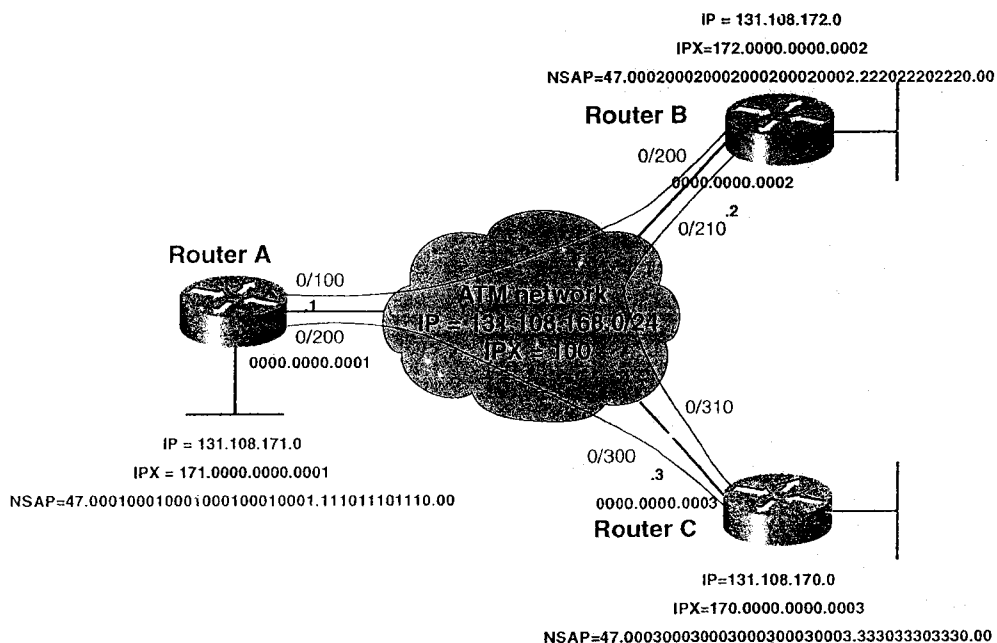


图 8-2 基于 PVC 的配置实例

注释 要在一个 VC 上只是承载一个协议，我只能想到一个例子：那就是需要进行单个协议的流量整形情况。假设需要在 ATM 网络上传输 IP 和 IPX，需要为 IP 和 IPX 网络定义不同的流量整形参数，大家可以考虑采用多路复用的封装方式。

例 8-1 多协议封装的 RFC 2684 静态 PVC 配置

```
Router A (config)# interface atm 0.1 multipoint
ip address 131.108.168.1 255.255.255.0
ipx network 100
atm pvc 10 0 100 aal5snap
atm pvc 11 0 200 aal5snap
map-group pvc-static-routerA-ip
map-group pvc-static-routerA-ipx

map-list pvc-static-routerA-ip
ip 131.108.168.2 atm-vc 10 broadcast
ip 131.108.168.3 atm-vc 11 broadcast
map-list pvc-static-routerA-ipx
ipx 100.0000.0000.0002 atm-vc 10 broadcast
```

(待续)


```
ipx 100.0000.0000.0003 atm-vc 11 broadcast

Router B (config)# interface atm 0.1 multipoint
ip address 131.108.168.2 255.255.255.0
ipx network 100
atm pvc 10 0 200 aal5snap
atm pvc 11 0 210 aal5snap
map-group pvc-static-routerB-ip
map-group pvc-static-routerB-ipx

map-list pvc-static-routerB-ip
ip 131.108.168.1 atm-vc 10 broadcast
ip 131.108.168.3 atm-vc 11 broadcast
map-list pvc-static-routerB-ipx
ipx 100.0000.0000.0001 atm-vc 10 broadcast
ipx 100.0000.0000.0003 atm-vc 11 broadcast

Router C (config)# interface atm 0.1 multipoint
ipx network 100
ip address 131.108.168.3 255.255.255.0
atm pvc 10 0 300 aal5snap
atm pvc 11 0 310 aal5snap
map-group pvc-static-routerC-ip

map-list pvc-static-routerC-ip
ip 131.108.168.1 atm-vc 10 broadcast
ip 131.108.168.2 atm-vc 11 broadcast
map-list pvc-static-routerC-ipx
ipx 100.0000.0000.0001 atm-vc 10 broadcast
ipx 100.0000.0000.0002 atm-vc 11 broadcast
```

配置过程非常简单：采用某种封装方法（本例中是 aal5snap 封装）手动分配 PVC，然后手动地在相应下一跳 IP 及 IPX 地址和对应协议目的地址的 PVC 的 VPI/VCI 之间建立正确的映射关系。VPI/VCI 的数值必须从管理 ATM 的组织获得。Cisco IOS 的映射是通过在全局配置模式中定义的 *map-lists* 建立，然后通过 *map-groups* 来引用的。请注意例子中给出的这两个映射列表的用法，每个列表都专门用于某个协议（本例中是 IP 和 IPX）。尽管可以将一份映射列表为两个协议公用，但建议为每个协议分配一份独立的映射列表，这样能使配置的网络更模块化。

图 8-3 是配置命令的完整句法结构，表 8-4 作了详细的解释。如果了解静态 PVC 通过 RFC 2684 封装形式进行配置的更多更深入的信息和更多的例子，可以参考《Cisco ATM Solutions》一书。

表 8-4

atm pvc 命令参数描述

字 段	描 述
Vcd	虚电路描述符。在路由器上惟一标识 PVC，vcd 之所以必须在整个路由器上惟一，是因为全局模式配置中将引用该编号。Vcd 编号位于路由器
Vpi	虚拟路径标识符，是 ATM VC 地址的一部分，必须与载波为特定地址提供的 VPI 相匹配，vpi 在路由器物理接口上必须惟一 ^①
Vci	虚拟信道标识符，是 ATM VC 地址的一部分，必须与匹配载波为特定地址提供的 VCI 相匹配，vci 在路由器物理接口上必须惟一 ^①

字 段	描 述
<i>midlow</i>	可选参数。只用于设置 aal34smds 封装。是 PVC 的起始消息标识符（MID）号码。默认为 0。如果设置了 aal34smds 封装的峰值和平均值（突发值为可选），则必须设置 midlow 和 midhigh 的值。这个可选项在 ATM 端口适配器上不可用
<i>midhigh</i>	可选参数。只用于设置 aal34smds 封装。是 PVC 的结束消息标识符（MID）号码。默认为 0。如果设置了 aal34smds 封装的峰值和平均值（突发值为可选），则必须设置 midlow 和 midhigh 的值
<i>peak</i>	可选参数，定义了虚电路传输的最大速率，单位：千比特每秒，默认值：peak=155 000kbit/s ^②
<i>average</i>	可选参数，定义了虚电路传输的平均速率，单位：千比特每秒，默认值：average=155 000kbit/s ^②
<i>burst</i>	可选参数。与 VC 在 PVC 上以 peak 速率传输的 ATM 最大信元数相关。默认值随不同的路由器类型变化。例如 7xxx 系列默认值为 94 ^③
<i>inarp x</i>	在 PVC 上启动逆向 ARP 的可选参数（只用于 IP，见 RFC2225:经典 IP）。逆向 ARP 数据包每 x 分钟在 PVC 上发送一次。默认值为 15 分钟
<i>oam x</i>	可选参数，用于配置 OAM F5 查询信元每 X 秒的速率。OAM F5 信元用于确认虚电路的连通性。远端主机必须以同样的信元作为应答

^① 引用物理接口不要和子接口的符号相混淆。Cisco 路由器把子接口看作真实的物理接口，每个子接口有单独的广播域，可以进行数据传输控制、定义策略和使用访问控制列表，例如，VPI/VCI 在实际物理接口范围内必须惟一，因为 ATM 交换机不能区分出“subinterface”。

^② Peak, Average 和 Burst 编号是 PCR, SCR 和 BT 值，路由器用该值按漏桶算法进行流量整形，该部分内容在《cisco ATM solution》一书的第 4 章“ATM 流量和网络管理”中讨论。

^③ 7xxx 系列的 burst 默认值 94 指定默认的突发极限为 94 个令牌，7xxx 系列的每个令牌处理 32 个信元，4xxx 系列中，每个令牌只处理 1 个信元。因此，在 cisco 的表述中，可以看到在 4xxx 命令中的“信元”和 7xxx 命令中的“令牌”。

以上信息摘自《cisco ATM solution》一书, Galina Diker Pildush, Cisco Press。

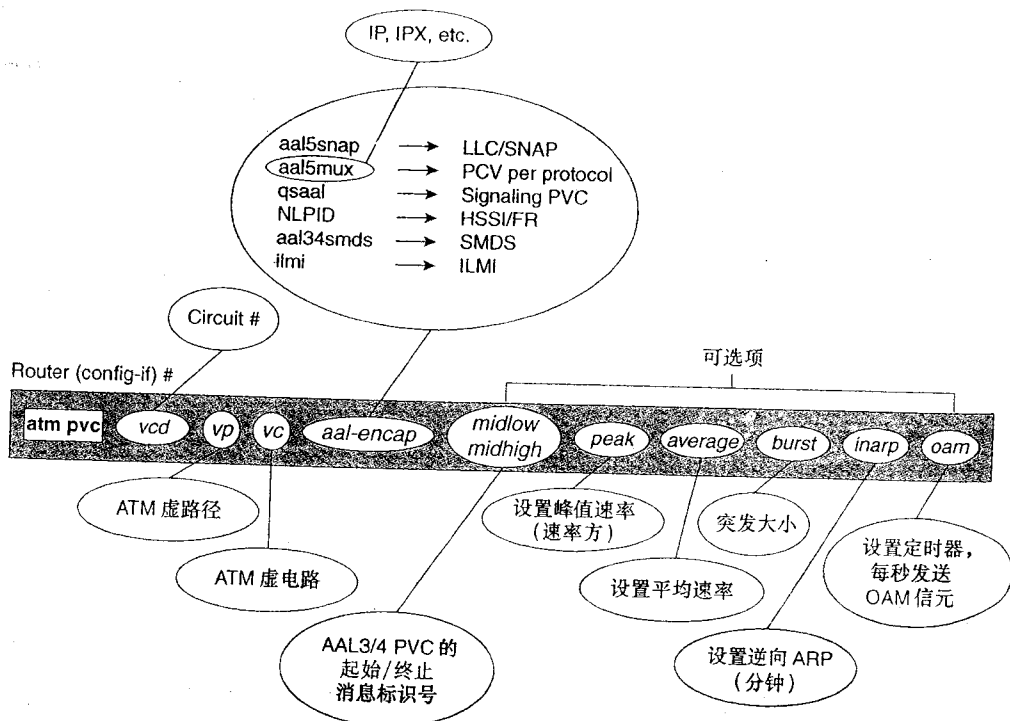


图 8-3 静态 PVC 配置命令的句法结构

2. 动态 PVC 的配置

动态 PVC 的配置不像静态方法那样需要用户干预。PVC 是通过 ATM 云创建的永久路径。这里的动态是指通过临时本地管理接口 (ILMI) 自动获得 VPI/VCI 值。使用 ILMI，需要通过 VCI = 16，这是 ATM 论坛规定的 ILMI PVC。输入 **atm ilmi-pvc-discovery subinterface** 命令后，ILMI PVC 携带传输 PVC 标识。例 8-2 详细解释了动态 PVC 的配置。

例 8-2 在图 8-2 的基础上对动态 PVC 的配置进行描述。

例 8-2 配置多协议封装的动态 PVC

```
Router A (config)# interface atm 0
atm pvc 1 0 16 ilmi
atm ilmi-pvc-discovery subinterface
interface atm 0.1 multipoint
ip address 131.108.168.1 255.255.255.0
ipx network 100
```

```
Router B (config)# interface atm 0
atm pvc 1 0 16 ilmi
atm ilmi-pvc-discovery subinterface
interface atm 0.1 multipoint
ip address 131.108.168.2 255.255.255.0
ipx network 100
```

```
Router C (config)# interface atm 0
atm pvc 1 0 16 ilmi
atm ilmi-pvc-discovery subinterface
interface atm 0.1 multipoint
ip address 131.108.168.3 255.255.255.0
ipx network 100
```

注意 **atm ilmi-pvc-discovery** 命令中的关键字 **subinterface**。该关键字使动态识别的 PVC 驻留在该关键字指定的 ATM 子接口上，子接口号应该和该 PVC 的 VPI 号匹配。该例中，动态识别的 PVC 必须具有一个值为 1 的 VPI 并将其分配给子接口 0.1。

注意这里没有定义静态映射。激活的 PVC 上允许 PVC 自动发现功能，并且路由器终止该 PVC 时，PVC 会产生 ATM 逆向 ARP 请求。这样，PVC 可以在没有静态映射的情况下解析其网络地址。

为防止通过逆向 ARP 获取的地址映射超时失效。地址映射还会周期性地刷新。周期间隔可以用 **inarp** 命令设置，默认值为 15 分钟。逆向 ARP 起初只适用于 IP（根据 RFC 2225 中经典 IP 的规定）。现在，Cisco IOS 已经将逆向 ARP 的适用范围扩展到了 IPX 协议。本章的“RFC 2225（经典 IP）的配置”一节会更详细地讲述这方面内容。

注释 使用 ATM 逆向 ARP 请求动态获得的 PVC 和经典 IP 的情况很相似。尽管经典的 IP 结构实际上只是基于 IP 协议，但是 Cisco 把这种功能扩展到了 IPX 协议领域。

8.2.2 SVC 的配置

SVC 的配置比 PVC 的配置显得更加具有动态性和灵活性。为什么呢？因为 SVC 是按需

建立的。不需要手动干预。如果要将从 A 点传送到 Z 点，信令会通过静态或是动态 ATM 路由选择协议动态地建立 VC。VC 建立完毕后，所有数据能通过预设好的路径传输。SVC 的好处在于用户无需担心 ATM 的可用性（当然，整个网络要处于工作状态）。用于某个预设好的 VC 链路如果出了问题，通过 Q.2931 信令可以动态地为数据建立新 VC。建立 SVC 有很多方法，可以用静态路由选择协议也可以采用动态路由选择协议。但这些已经不属于本书的范畴，可以参考《Cisco ATM Solutions》一书。

由于 SVC 的建立是动态的，其寻址方式也和 PVC 不同。PVC 地址是一个 VPI/VCI 的组合，具有本地标识。本地标识说明了一切——ATM 网络入口处的协议不会将 ATM 的目的地址在整个网络中进行传递。相反，SVC 则具有全局意义，经由从 ATM 网络接收信息并进行传输的 ATM 边缘设备来建立 VC。这些 ATM 边缘设备通过 Q.2931 信令协议完成建立 VC 的工作，Q.2931 信令携带 ATM 目的信息和 QoS 参数。ATM 网络中的路径选择和建立该 VC 所需的跳数（ATM 交换机）对于 IP 层来说都不重要。

基于 SVC 的 ATM 地址格式包括 160 位（40 个 16 进制数）的二进制数字，如图 8-4 所示。如果使用专用 ATM 网络，可能需要用到一个基于 NSAP 的 ATM 地址。如果使用公用 ATM 网络，则可能需要一个基于 E.164 的 ATM 地址。《Cisco ATM Solutions》一书详细地讨论了不同的 ATM 地址格式的问题，可以作为参考。

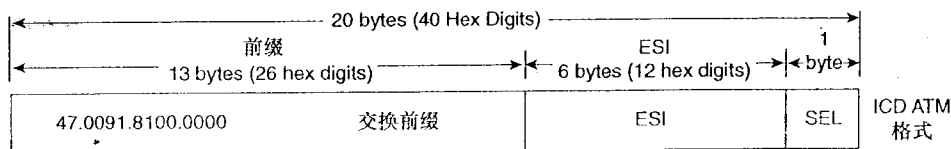


图 8-4 SVC ATM 地址格式

尽管这里的地址看上去很长，输入时很麻烦（以 16 进制表示），但不用担心，ATM 论坛已经估计到这可能会是一个问题，因此允许这些地址以 ILMI 的方式获取。这里要做的是给它分配一个终端系统标识符（ESI）。ESI 包括 12 个 16 进制数和 2 个 16 进制的自主号。地址的前缀是用 ILMI 从所连接的 ATM 交换机上自动获取的。ESI 分配的方法很多——可以用路由器 LAN 接口的 MAC 地址，或是分配一个与接口的 IP 地址相符合的任意地址。例如，如果 ATM 接口（或子接口）的 IP 地址是 177.10.168.1，可以为 ESI 分配地址 0177.1016.8100。使用 ILMI 则需要建立 ATM 论坛定义的 VCI=16 的 ILMI PVC。

表 8-5 列出了 SVC 和 PVC 的配置过程的相同点和不同点。

表 8-5 SVC 与 PVC 的配置

相 同 点	不 同 点
为接口或子接口分配 ATM 地址	ATM 地址格式不同（全局地址）
将第 3 层的下一跳地址静态映射到相应的 ATM 地址（使用全局 ATM 地址）	需要为信令建立 PVC，如 ILMI（可选）

RFC 2684 的配置包括手动为 ATM 边缘设备分配完整的 ATM 地址，或者只是手动分配 NSAP 地址的 ESI 部分并设置 ILMI。例 8-3 在图 8-1 的基础上对边缘路由器 A、B 和 C 进行了 RFC 2684 的配置，配置中还包括了对 IP 和 IPX 协议的 RFC 2684 处理，而且还为路由器

A, B 和 C 分配了完整的 NSAP 地址。

例 8-3 配置多协议封装的 RFC 2684 SVC

```
Router A (config)# interface atm 0
atm pvc 5 0 5 qsaal

interface atm 0.1
ip address 138.108.168.1 255.255.255.0
atm nsap-address 47.000100010001000100010001.111011101110.00
map-group ip-routerA
map-group ipx-routerA

map-list ip-routerA
ip 131.108.168.2 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast
ip 131.108.168.3 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

map-list ipx-routerA
ipx 100.0000.0000.0002 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast
ipx 100.0000.0000.0003 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

Router B (config)# interface atm 0
atm pvc 5 0 5 qsaal

interface atm 0.1
ip address 138.108.168.2 255.255.255.0
atm nsap-address 47.000200020002000200020002.222022202220.00
map-group ip-routerB
map-group ipx-routerB

map-list ip-routerB
ip 131.108.168.1 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ip 131.108.168.3 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

map-list ipx-routerB
ipx 100.0000.0000.0001 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ipx 100.0000.0000.0003 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

Router C (config)# interface atm 0
atm pvc 5 0 5 qsaal

interface atm 0.1
ip address 138.108.168.3 255.255.255.0
atm nsap-address 47.000300030003000300030003.333033303330.00
map-group ip-routerC
map-group ipx-routerC

map-list ip-routerC
ip 131.108.168.1 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ip 131.108.168.2 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast

map-list ipx-routerC
ipx 100.0000.0000.0001 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ipx 100.0000.0000.0002 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast
```

例 8-4 以图 8-1 为基础，对与例 8-3 中同样的边缘设备进行了另一次 RFC 2684 SVC 配置。在这个例子中，ATM 地址的 ESI 部分分配给路由器的接口，而 ILMI 则是用来从连接的 ATM 交换机中获取 NSAP 前缀部分。请注意，例 8-3 只有信令 PVC，而例 8-4 则含有两种 PVC 的应用情况——信令 PVC 和 ILMI 的 PVC。这两种 PVC 都必须分配给主接口。

例 8-4 多协议封装的 RFC 2684 SVC 配置

```
Router A (config)# interface atm 0
atm pvc 5 0 5 qsaal
atm pvc 2 0 16 ilmi

interface atm 0.1
ip address 138.108.168.1 255.255.255.0
atm esi-address 111011101110.00
map-group ip-routerA
map-group ipx-routerA

map-list ip-routerA
ip 131.108.168.2 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast
ip 131.108.168.3 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

map-list ipx-routerA
ipx 100.0000.0000.0002 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast
ipx 100.0000.0000.0003 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

Router B (config)# interface atm 0
atm pvc 5 0 5 qsaal
atm pvc 2 0 16 ilmi

interface atm 0.1
ip address 138.108.168.2 255.255.255.0
atm esi-address 222022202220.00
map-group ip-routerB
map-group ipx-routerB

map-list ip-routerB
ip 131.108.168.1 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ip 131.108.168.3 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

map-list ipx-routerB
ipx 100.0000.0000.0001 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ipx 100.0000.0000.0003 atm-nsap 47.000300030003000300030003.333033303330.00 broadcast

Router C (config)# interface atm 0
atm pvc 5 0 5 qsaal
atm pvc 2 0 16 ilmi

interface atm 0.1
ip address 138.108.168.3 255.255.255.0
atm esi-address 333033303330.00
map-group ip-routerC
map-group ipx-routerC

map-list ip-routerC
```

(待续)

```
ip 131.108.168.1 atm-nsap 47.000100010001000100010001.111011101110.00 broadcast
ip 131.108.168.2 atm-nsap 47.000200020002000200020002.222022202220.00 broadcast

map-list ipx-routerC
ipx 100.0000.0000.0001 atm-nsap 47.000100010001000100010001.111011101110.00
broadcast
ipx 100.0000.0000.0002 atm-nsap 47.000200020002000200020002.222022202220.00
broadcast
```

RFC 2684 的 SVC 命令句法总结如下：

```
Router (config-if) # atm nsap-address nsap-address
```

或

```
Router (config-if) # atm esi-address esi
```

```
Router (config-if) # map-group name
```

```
Router (config) # map-list name
```

```
Router (config-map-list) # protocol protocol-address atm-nsap atm-nsap-address
[class class-name][broadcast]
```

表 8-6 在基于 SVC 的 ATM 网络中对路由器互连需要的所有参数进行了总结。

表 8-6

SVC 相关命令的参数描述

参 数	描 述
nsap-address	源地址，定义为 40 个 16 进制数字
esi	终端工作站系统标识 (esi)，定义为 12 个 16 进制数字。要形成完整的 NSAP 地址，可借助 ILMI 从入口交换机处动态学习 26 个 16 进制数字长的前缀
name	在全局模式下配置的映射表名称。必须在接口或子接口模式下关联调用组引用映射表名称以应用映射列表
protocol	根据所使用的第 3 层协议选择下列关键字之一：ip、ipx、appletalk、decnet、vines、apollo 等等
Protocol-address	映射到当前 SVC 的目标地址
atm-nsap-address	目标 ATM NSAP 地址
class-name	流量参数映射类列表的引用名称。改变流量参数的默认值时可选。借助 map-class 声明可以自定义多种类型流量的流量整形参数
broadcast	如果 ATM 接口要使用协议广播数据，如路由更新，则必须使用该关键字

表 8-6 摘自《cisco ATM solution》一书,Galina Diker Pildush,Cisco Press。

Cisco IOS 通过 map-class 命令扩展了 SVC 的流量整形功能：

```
Router (config-map-class) # atm forward-peak-cell-rate-clp0 rate
atm backward-peak-cell-rate-clp0 rate
atm forward-peak-cell-rate-clp1 rate
atm backward-peak-cell-rate-clp1 rate
atm forward-sustainable-cell-rate-clp0 rate
atm backward-sustainable-cell-rate-clp0 rate
atm forward-sustainable-cell-rate-clp1 rate
atm backward-sustainable-cell-rate-clp1 rate
atm forward-max-burst-size-clp0 cell-count
atm backward-max-burst-size-clp0 cell-count
atm forward-max-burst-size-clp1 cell-count
atm backward-max-burst-size-clp1 cell-count
```

映射类通过映射列表中的 **class** 命令调用，如例 8-5 所示。

例 8-5 配置 SVC 数据管理的例子

```
Router(config)# map-class atm contract-svcs
Router(config-map-class)# atm forward-peak-cell-rate-clp0 56000
Router(config)# map-list test ip 200.0.0.1 atm-nsap
44.44440000000000000000000000.000000000000.00 class contract-svcs
```

8.3 RFC 2225（经典 IP）的配置

RFC 2225 中定义的经典 IP 是通过 ATM 网络进行动态方式的 IP 互联，使用 RFC 2684 封装，为 ATM 网络的 IP 互联提供了一种动态方式，使用户无需手工进行大量映射声明的配置。这里是对经典 IP 的定义：

它只对 IP 进行互联。使得 IP 在 ATM 网络中具有原本的特性，即 IP 地址到 ATM PVC 映射的 IP ARP 功能或 SVC 都是动态映射。这里所说的动态，对于 SVC 来说是指通过 ARP 服务器来实现，而对于 PVC 来说，必须为所定义的每个 PVC 都进行 ATM 逆向 ARP 的配置。SVC 不会为每个规范都提供一定的冗余度，Cisco 用自己的一套方法来提供冗余度。RFC 1577 和 2225 都指定了单个网络/子网（或者可以称之为一个广播域）内的 IP 互操作性。

经典 IP 采用的是客户/服务器结构，这有助于 ATM 将自己伪装成一个网络广播域。回想一下，IP ARP 是一个本地广播协议，但 ATM 却是一个非广播域，这就需要采用这种客户/服务器结构来解决这个问题。所有的客户端都是和服务器连接在一起的，而服务器则会提供 ARP 的功能。

这里要注意的是只有基于 SVC 的 ATM 网络才采用客户/服务器结构，而基于 PVC 的则不需要，这是由于基于 PVC 的 ATM 网络预先创建了 VC，采用的是具有本地意义的 ATM 寻址方式，这类 ATM 网络需要的是采用 ATM 逆向 ARP 来把 VC 标识符与相应的 IP 地址对应起来，而不需要使用 ARP 服务器。这也就是逆向 ARP 帧中继工作的方式。最早的经典 IP 规范（RFC 1577）并没有指定 ARP 服务器冗余度的问题，Cisco 在提供了 ARP 服务器冗余度的 RFC 2225 发布之前就给出了关于服务器冗余度问题的解决方案。可以参考《Cisco ATM Solutions》一书，以了解更多与经典 IP 理论有关的内容。

现在来看一些基于 PVC 和 SVC 的 ATM 网络的配置实例。

8.3.1 PVC 的配置

PVC 网络的经典 IP 配置采用 ATM 逆向 ARP，ATM 逆向 ARP 的作用就是动态地将与预定义的 PVC 相关联的 IP 地址向边缘设备发布，这样边缘设备可以建立设备 IP 地址与相应的 PVC 虚拟电路描述符（VCD）之间的动态 ARP 表。

每台路由器（A，B，C 和 D）都会采用 ATM 逆向 ARP 动态地建立一份表，这份表将 IP 地址与本地有意义的 VCD 号联系在一起，如表 8-7 所示。

表 8-7

动态 IP 和 VCD 号的分配

路 由 器	IP 地址	VCD 号
A	138.108.168.2	12
	138.108.168.3	13
	138.108.168.4	14
B	138.108.168.1	21
	138.108.168.3	23
	138.108.168.4	24
C	138.108.168.1	31
	138.108.168.2	32
	138.108.168.4	34
D	138.108.168.1	41
	138.108.168.2	42
	138.108.168.3	43

配置经典 IP 的 PVC 命令句法结构与一般 PVC 命令完全一样。创建 PVC 和应用、配置 RFC 2684 一样。区别只有两点。在 **atm pvc** 命令的结尾需要指定 **inarp**，并且还要删去引用相应映射表的 **map-group** 声明。表 8-4 详细列出了 PVC 命令的所有细节。

现在看一下图 8-5 所示的网络情况。4 台路由器通过 ATM 网络互连，标识的 VCD 值允许使用全连接的拓扑结构，这样任何一台路由器都可以直接使用虚拟电路 (VC) 与任何一台路由器相连接。

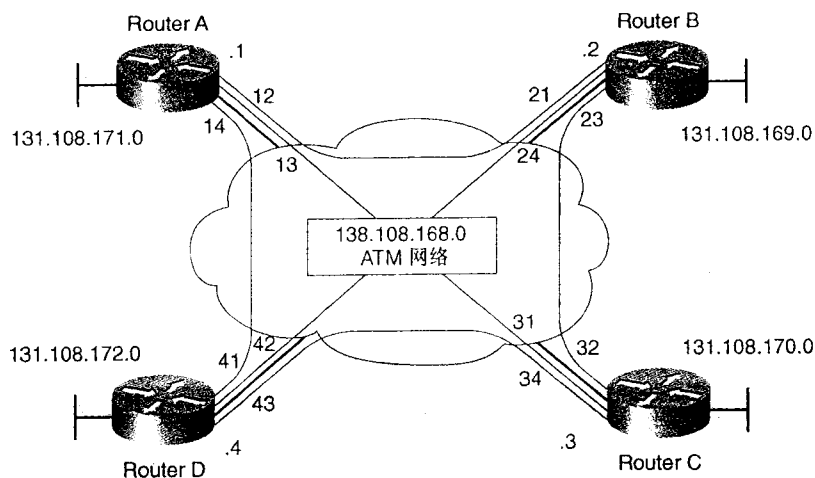


图 8-5 基于 PVC 的 ATM 网络的经典 IP

例 8-6 给出了所有路由器的配置情况。

例 8-6 基于 PVC 的 ATM 网络的经典 IP 配置情况

```
Router A (config)# interface atm 0
no shutdown
interface atm 0.1
ip address 138.108.168.1 255.255.255.0
atm pvc 12 0 77 aal5snap inarp 5
atm pvc 13 0 78 aal5snap inarp 5
atm pvc 14 0 79 aal5snap inarp 5
```

```
Router B (config)# interface atm 0
no shutdown
interface atm 0.1
ip address 138.108.168.2 255.255.255.0
atm pvc 21 0 87 aal5snap inarp 5
atm pvc 23 0 88 aal5snap inarp 5
atm pvc 24 0 89 aal5snap inarp 5
```

```
Router C (config)# interface atm 0
no shutdown
int atm 0.1
ip address 138.108.168.3 255.255.255.0
atm pvc 31 0 97 aal5snap inarp 5
atm pvc 32 0 98 aal5snap inarp 5
atm pvc 34 0 99 aal5snap inarp 5
```

```
Router D (config)# interface atm 0
no shutdown
int atm 0.1
ip address 138.108.168.4 255.255.255.0
atm pvc 41 0 107 aal5snap inarp 5
atm pvc 42 0 108 aal5snap inarp 5
atm pvc 43 0 109 aal5snap inarp 5
```

每台路由器都设置成每 5 分钟发送一次逆向 ARP 数据包。

8.3.2 SVC 的配置

基于 SVC 的 ATM 网络的经典 IP 的配置以客户/服务器结构为基础。这意味着必须有一台服务器为多个客户端设备提供服务。经典 IP 提供 ARP 服务器（称为 ATM ARP 服务器）为广播域内的众多客户提供服务。用户不用再去配置全局 ATM 地址与 IP 地址之间的静态映射关系，ATM ARP 服务器会动态地完成这些工作。

结构的简单使得 ATM ARP 服务器和客户端上需要做的配置工作都很少，要做的只是在服务器和客户路由器上分配完整的 ATM 地址，或者仅分配 ESI 前缀（此时，每台路由器会动态从入口 ATM 交换机处获取这些前缀），然后客户端上还需指定 ATM ARP 服务器的完整 ATM 地址。所有的客户端和这台服务器都必须具有信令和 ILMI PVC。接着，神奇之处出现。获得 ATM ARP 服务器地址的客户端会自动将自己的情况发布给服务器。反过来，客户/服务器的 VC 设置之后，服务器会建立其动态 ARP 表，表中包含有客户端的 IP 地址和 ATM 地址

的前后对照情况。

图 8-6 给出了一个经典 IP 的设置实例，4 台路由器通过 ATM 共享 IP 网络 138.108.168.0。路由器 B 是 ATM ARP 服务器，A，B 和 C 都是客户端。

定义了客户端之后，每一台客户路由器都会建立一个到 ATM ARP 服务器的 VC，然后，将各自的 ATM ARP 请求数据包发送到 ATM ARP 服务器。服务器 B 会对每个 ATM ARP 请求进行检查，用获得的信息建立其 ATM ARP 缓存。这些信息就是客户的 ATM 地址与 IP 地址的前后参照情况，如表 8-8 所示。客户端利用 ARP 表中的各项可以进行相互通信而无需静态映射列表。

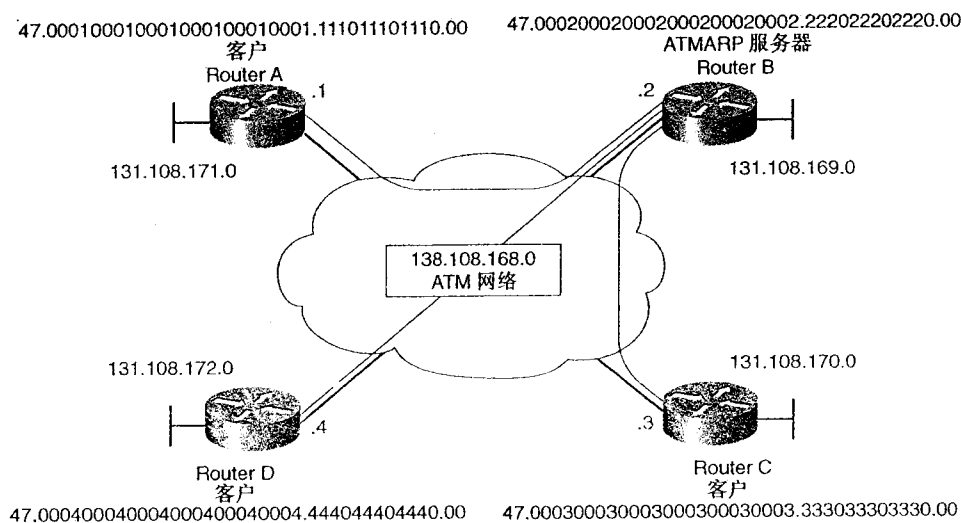


图 8-6 基于 SVC 的 ATM 网络的经典 IP

表 8-8 动态产生的 ATM ARP 服务器缓存项

IP 地址	ATM (NSAP) Address
138.108.168.1	47.000100010001000100010001.111011101110.00
138.108.168.2	47.000200020002000200020002.222022202220.00
138.108.168.3	47.000300030003000300030003.333033303330.00
138.108.168.4	47.000400040004000400040004.444044404440.00

例 8-7 是 4 台路由器（客户路由器 A，C 和 D 以及 ATM ARP 服务器 B）的配置情况。注意，不再需要静态映射。

例 8-7 应用 ATM SVC 网络的经典 IP 配置

```
Router A (config)# interface atm 0
atm pvc 1 0 5 qsaal
no shutdown
interface atm 0.1
```

（待续）

ip address	138.108.168.1 255.255.255.0
atm interface address	000100010001000100010001.111011101110.00
atm interface server	47.000200020002000200020002.222022202220.00
Router A (config)	interface atm 0
atm interface	1 0 5 qs
no shutdown	
interface atm 0.	
ip address	138.108.168.2 255.255.255.0
atm interface address	000200020002000200020002.222022202220.00
atm interface server	
Router B (config)	interface atm 0
atm interface	1 0 5 qs
no shutdown	
interface atm 0.	
ip address	138.108.168.3 255.255.255.0
atm interface address	000300030003000300030003.333033303330.00
atm interface server	47.000200020002000200020002.222022202220.00
Router D (config)	interface atm 0
atm interface	1 0 5 qs
no shutdown	
interface atm 0.	
ip address	138.108.168.4 255.255.255.0
atm interface address	000400040004000400040004.444044404440.00
atm interface server	47.000200020002000200020002.222022202220.00

例 8-7 在一个广域网中使用 ATM ARP 服务器的例子。如果需要 ATM ARP 服务器冗余，那就有可能需要像图 8-7 那样在客户端中再定义一个 ATM ARP 服务器，如例 8-8 所示。第二个 ATM ARP 服务器由路由器 D。

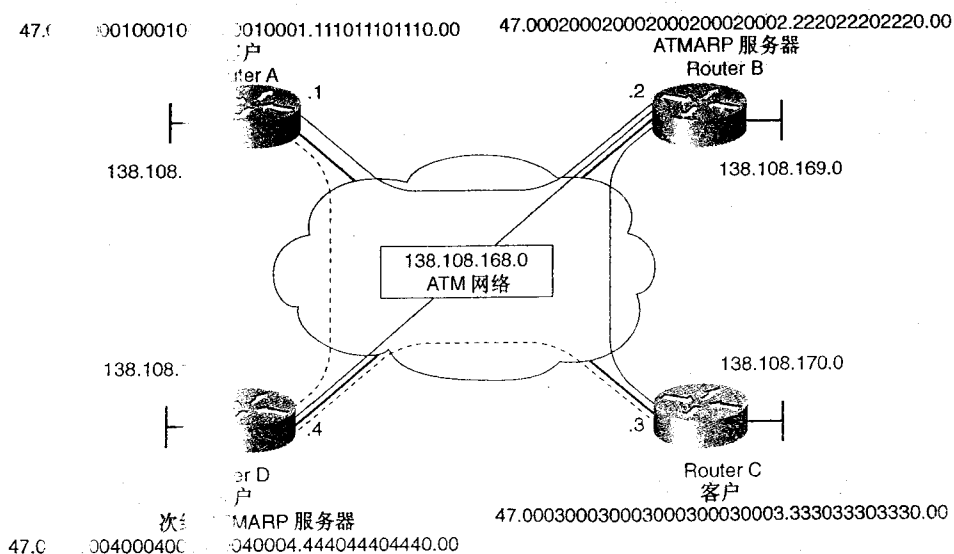


图 8-7 基于 SVC 的 ATM 网络冗余 ATM ARP 服务器

例 8-8 基于 SVC 的 ATM 网络的冗余 ATM ARP 服务器配置

```
Router A (config)# interface atm 0
atm pvc 1 0 5 qsaal
no shutdown
interface atm 0.1
ip address 138.108.168.1 255.255.255.0
atm classic-ip-extensions bfi
atm nsap-address 47.00010001000100010001.111011101110.00
atm arp-server nsap 47.00020002000200020002.222022202220.00
atm arp-server nsap 47.00040004000400040004.444044404440.00
```

```
Router B (config)# interface atm 0
atm pvc 1 0 5 qsaal
no shutdown
interface atm 0.1
ip address 138.108.168.2 255.255.255.0
atm classic-ip-extensions bfi
atm nsap-address 47.00020002000200020002.222022202220.00
atm arp-server nsap 47.00040004000400040004.444044404440.00
atm arp-server self
Router C - Client
```

```
Router C (config)# interface atm 0
atm pvc 1 0 5 qsaal
no shutdown
interface atm 0.1
ip address 138.108.168.3 255.255.255.0
atm classic-ip-extensions bfi
atm nsap-address 47.00030003000300030003.333033303330.00
atm arp-server nsap 47.00020002000200020002.222022202220.00
atm arp-server nsap 47.00040004000400040004.444044404440.00
```

```
Router D (config)# interface atm 0
atm pvc 1 0 5 qsaal
no shutdown
interface atm 0.1
ip address 138.108.168.4 255.255.255.0
atm classic-ip-extensions bfi
atm nsap-address 47.00040004000400040004.444044404440.00
atm arp-server nsap 47.00020002000200020002.222022202220.00
atm arp-server self
```

新命令 **atm classic-ip-extensions bfi** 的使用值得注意，它允许 ATM ARP 服务器的冗余度。起初，客户 A 和 B 将路由器 B 作为 ATM ARP 服务器。如果路由器 B 变为不可用，客户端就会使用备份的路由器 D 作为 ATM ARP 服务器。

经典 IP 的配置只需在客户端和服务服务器上定义 ATM 地址的 ESI 部分，然后利用 ILMI 从相应的入口 ATM 交换机处获取客户和服务器的 NSAP 地址的前缀部分。

总结一下，基于 SVC 的 ATM 网络上客户端的经典 IP 配置命令的句法结构如下：

```
Router (config) # int atm int#
Router (config-if) # atm pvc vcd# vpi# 5 qsaal
如果使用了地址的 ESI 部分，那么：
Router (config-if) # atm pvc vcd# vpi# 16 ilmi
Router (config) # int atm subint#
```

```
Router (config-if) # atm nsap nsap-address
```

或：

```
Router (config-if) # atm esi esi-portion-address  
Router (config-if) # atm arp-server nsap nsap-address
```

ATM ARP 服务器的配置语法格式如下：

```
Router (config) # int atm int#  
Router (config-if) # atm pvc vcd# vpi# 5 qsaal
```

如果使用了地址的 ESI 部分，那么：

```
Router (config-if) # atm pvc vcd# vpi# 16 ilmi  
Router (config) # int atm subint#  
Router (config-if) # atm nsap nsap-address
```

或：

```
Router (config-if) # atm esi esi-portion-address  
Router (config-if) # atm arp-server self [time-out minutes]
```

命令 **atm arp-server self** 告诉路由器，它就是 ATM ARP 服务器。参数 **time-out** 是可选的，单位是分钟，表明在服务器开始验证移出某个映射之前，ARP 缓存中的某个目的项已经存储的时间。RFC 2225 中规定 ATM ARP 缓存超时的默认值是 20 分钟。

8.4 实验 18：Cisco 7XXX 路由器上的 PVC， RFC 2684 的配置——第 1 部分

8.4.1 实验说明

对于 ATM 来说，很不幸的一点是大部分的 ATM 应用都以 PVC 为基础。之所以称其为“不幸”，是因为以 SVC 为基础的网络设置起来要容易得多，而且动态 ATM QoS 功能也只有基于 SVC 的网络才能提供。但理解基于 PVC 的 ATM 网络配置过程是 CCIE 实验考试的一个前提条件。

8.4.2 实验内容

假定我们现在运营着一个基于 IP 的网络，网络本身通过了一个 ATM 网络区域。我们需要把各个节点互连起来。ATM 网络是以 PVC 为基础。而我们的服务提供商答应向我们提供连接 5 个节点所需的所有地址。采用的网络路由选择协议是 EIGRP。在设计网络的时候，下面这些要求是必须实现的：

- 路由选择协议是 EIGRP，自治系统（AS）ID 是 100。
- 所有的 PVC 都需要分配适当的 VPI/VCI 号。
- 维持一个全连接的网络拓扑体系。

8.4.3 实验目的

实验结束的时候，我们希望完成：

- 按照要求对所有的 Cisco 路由器进行配置。
- 配置一个全连接的 PVC 网络环境。
- 对 RFC 2684 进行配置。

8.4.4 所需设备

这个实验需要 5 台路由器和一台 ATM 交换机。如果是为了准备 CCIE，则只需要两台路由器通过 ATM 网络连接。总之，这个实验中，我们需要：

- 至少 5 台具有 ATM 接口的路由器。
- 一台 LS1010 ATM 交换机。

8.4.5 物理设计与实验准备

- 按照图 8-8 将路由器与 ATM 交换机相连。
- 请注意，尽管图 8-8 中给出了 5 台路由器，但是实际上我们把两台路由器通过 ATM 网络连接起来也能完成本实验的内容。

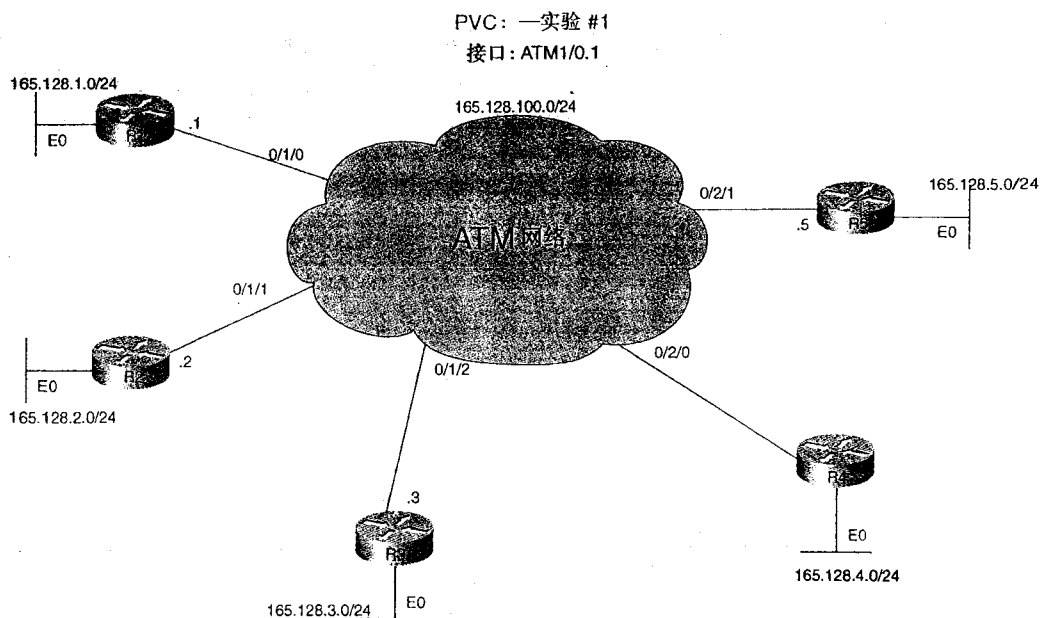


图 8-8 IP 的 PVC

8.5 实验 18：Cisco 7XXX 路由器上的 PVC，RFC 2684 的配置——第 2 部分

8.5.1 实验步骤

我们需要对路由选择协议以及 IP 地址方案进行配置，这在两个实验中都会用到。初始设置完成之后，我们需要建立 8 条通往这些路由器的 PVC，从而组成了一个全连接的网络拓扑结构。IP 的互连必须通过 ATM 网络延伸。

表 8-9 是用于完成该实验所需的配置命令。

表 8-9 实验所需命令总结

命 令	描 述
router eigrp	在路由器上启动 EIGRP 路由选择协议
atm pvc	设置双向虚电路的参数
map-list	将第 3 层地址映射到 PVC
show atm vc	显示路由器上的虚电路配置
show atm interface atm	显示特定端口的详细配置
show atm map	显示用户创建的映射表内容
debug atm event	显示路由器和 AM 交换机之间的 PVC 的信息
show ip route	显示 IP 路由表
Ping	测试第 3 层的连通性

参考图 8-8 按照下面的步骤完成本实验的配置工作。

1. 第 1 步：配置路由器的参数

配置路由器下面的一些参数：

- 主机名：Rn（n 是路由器编号）。
- 密码：atmlab。
- 虚拟终端/控制台密码：cisco。
- IP 路由发送协议：EIGRP。
- EIGRP 自治系统号：100。
- IP 是惟一的被路由协议。

例 8-9 给出了 R1 的配置范例。

例 8-9 路由器 R1 的配置

```
R1(config)# hostname R1
!
enable password atmlab
!
router eigrp 100
 network 165.128.0.0
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password cisco
 login
!
```

2. 第 2 步. 在以太接口 0 和 ATM1/0.1 接口配置 IP 地址

根据表 8-10 为以太接口 Ethernet 0 和 ATM 1/0.1 配置 IP 地址。

表 8-10

IP 地址分配

路 由 器	E0	ATM1/0.1
R1	165.128.1.1/24	165.128.100.1/24
R2	165.128.2.2/24	165.128.100.2/24
R3	165.128.3.3/24	165.128.100.3/24
R4	165.128.4.4/24	165.128.100.4/24
R5	165.128.5.5/24	165.128.100.5/24

例 8-10 是 R1 这一步的配置情况。

例 8-10 路由器 R1 的 IP 地址配置

```
R1(config)# interface e0
 ip address 165.128.1.1 255.255.255.0
interface atm1/0.1 multipoint
 ip address 165.128.100.1 255.255.255.0
```

3. 第 3 步: 为到达其他每台路由器配置一条 PVC

要创建一个全连接的 ATM PVC 网络，需要为每一个路由器配置一条 PVC，采用 AAL5SNAP。VPI/VCI 号的分配请参考表 8-11。

表 8-11

VPI/VCI 的分配

来 自	去 往	VCD	VPI	VC1
R1	R2		0	112
	R3		0	113

续表

来 自	去 往	VCD	VP1	VC1
	R4		0	114
	R5		0	115
R2	R1		0	121
	R3		0	123
	R4		0	124
	R5		0	125
R3	R1		0	131
	R2		0	132
	R4		0	134
	R5		0	135
R4	R1		0	141
	R2		0	142
	R3		0	143
	R5		0	145
R5	R1		0	151
	R2		0	152
	R3		0	153
	R4		0	154

例 8-11 是在 R1 上的配置的例子。

例 8-11 路由器 R1 的 PVC 配置

```
R1(config)# interface atm1/0.1 multipoint
ip address 165.128.100.1 255.255.255.0
atm pvc 112 0 112 aal5snap
atm pvc 113 0 113 aal5snap
atm pvc 114 0 114 aal5snap
atm pvc 115 0 115 aal5snap
```

4. 第 4 步：配置 LS1010 以管理 PVC

在 CCIE 实验考试中，ATM 交换机是预先配置好的。但是，对于我们准备 CCIE 试验来说，还需要自己来对交换机进行配置，如例 8-12 所示。

例 8-12 配置 ATM 交换机来管理 PVC

```
ATM-Switch# version 11.3
no service pad
no service udp-small-servers
```

(续表)

```
no service tcp-small-servers
!
hostname ATM-Switch
!
enable password atmlab
!
no ip domain-lookup
!
atm address 47.0091.8100.0000.0010.0739.a101.0010.0739.a101.00
atm router pnni
  node i level 56 lowest
  redistribute atm-static
!
interface ATM0/1/0
  no ip address
  no atm auto-configuration
  atm uni version 3.1
  atm maxvpi-bits 3
  atm maxvci-bits 10
  atm pvc 0 112 interface ATM0/1/1 0 121
  atm pvc 0 113 interface ATM0/1/2 0 131
  atm pvc 0 114 interface ATM0/2/0 0 141
  atm pvc 0 115 interface ATM0/2/1 0 151
!
interface ATM0/1/1
  no ip address
  no atm auto-configuration
  atm uni version 3.1
  atm maxvpi-bits 3
  atm maxvci-bits 10
  atm pvc 0 121 interface ATM0/1/0 0 112
  atm pvc 0 123 interface ATM0/1/2 0 132
  atm pvc 0 124 interface ATM0/2/0 0 142
  atm pvc 0 125 interface ATM0/2/1 0 152
!
interface ATM0/1/2
  no ip address
  no atm auto-configuration
  atm uni version 3.1
  atm maxvpi-bits 3
  atm maxvci-bits 10
  atm pvc 0 131 interface ATM0/1/0 0 113
  atm pvc 0 132 interface ATM0/1/1 0 123
  atm pvc 0 134 interface ATM0/2/0 0 143
  atm pvc 0 135 interface ATM0/2/1 0 153
!
interface ATM0/2/0
  no ip address
  no atm auto-configuration
  atm uni version 3.1
  atm maxvpi-bits 3
  atm maxvci-bits 10
  atm pvc 0 141 interface ATM0/1/0 0 114
  atm pvc 0 142 interface ATM0/1/1 0 124
  atm pvc 0 143 interface ATM0/1/2 0 134
  atm pvc 0 145 interface ATM0/2/1 0 154
!
interface ATM0/2/1
  no ip address
  no atm auto-configuration
  atm uni version 3.1
```

(续表)

```

atm maxvpi-bits 3
atm maxvci-bits 10
atm pvc 0 151 interface ATM0/1/0 0 115
atm pvc 0 152 interface ATM0/1/1 0 125
atm pvc 0 153 interface ATM0/1/2 0 135
atm pvc 0 154 interface ATM0/2/0 0 145
!
interface Ethernet2/0/0
ip address 10.0.0.12 255.0.0.0
!
ip classless
!
line con 0
exec-timeout 0 0
password cisco
login
line aux 0
line vty 0 4
exec-timeout 0 0
password cisco
login
!
end

```

5. 第5步：建立邻居路由器的IP网络号与适当的PVC之间的映射关系

例8-13 利用R1演示了这一步的配置情况。

例8-13 建立IP网络与相应PVC之间的映射关系

```

R1(config)# interface atm1/0.1 multipoint
ip address 165.128.100.1 255.255.255.0
atm pvc 112 0 112 aal5snap
atm pvc 113 0 113 aal5snap
atm pvc 114 0 114 aal5snap
atm pvc 115 0 115 aal5snap
map-group ip-Pvc
!
map-list ip-Pvc
ip 165.128.100.2 atm-vc 112 broadcast
ip 165.128.100.3 atm-vc 113 broadcast
ip 165.128.100.4 atm-vc 114 broadcast
ip 165.128.100.5 atm-vc 115 broadcast

```

6. 第6步：对所做的配置进行测试

8.6 实验19：在Cisco 7XXX路由器上利用SVC对经典IP，RFC 2225进行配置——第1部分

8.6.1 实验说明

射关系等等，这个实验里还演示了 SVC 的用法。

8.6.2 实验内容

假定我们现在运营着一个基于 IP 的网络，网络本身通过了一个 ATM 网络区域。我们需要把各个节点互连起来。ATM 网络是以 SVC 为基础的。服务提供商并没有提供 SVC 地址的详情。因此，我们就需要利用 ILMI 来从服务提供商的入口 ATM 交换机处获取地址的前缀。此外，我们还需要使用经典 IP，路由选择协议采用 EIGRP。设计网络的时候一定要满足下面要求：

- 路由选择协议采用 EIGRP，自治系统 ID 为 100。
- 分配 NSAP 地址的 ESI 部分。
- 建立信令和 ILMI PVC。
- 将 R2 做为 ATM ARP 服务器，其他的都作为客户机。
- 维持全连接的网络拓扑体系。

8.6.3 实验目的

在实验结束的时候，我们需要完成：

- 启动 SVC 网络的经典 IP (RFC 2225)。
- 完成 ARP 服务器与客户机的配置 (RFC 2225)。
- 完成一个逻辑 IP 子网 (LIS)。

8.6.4 所需设备

这个实验需要 5 台路由器和一台 ATM 交换机。如果是为了准备 CCIE，则只需要两台路由器通过 ATM 网络连接。总之，这个实验中，我们需要：

- 至少 5 台具有 ATM 接口的路由器。
- 一台 LS1010 ATM 交换机。

8.6.5 物理设计与实验准备

- 按照图 8-9 将路由器与 ATM 交换机相连。
- 请注意，尽管图 8-9 中给出了 5 台路由器，但是实际上如果只有两台路由器通过 ATM 网络连接起来也能完成本实验的内容。

图 8-9 给出了这个实验中需要用到网络与地址的情况。

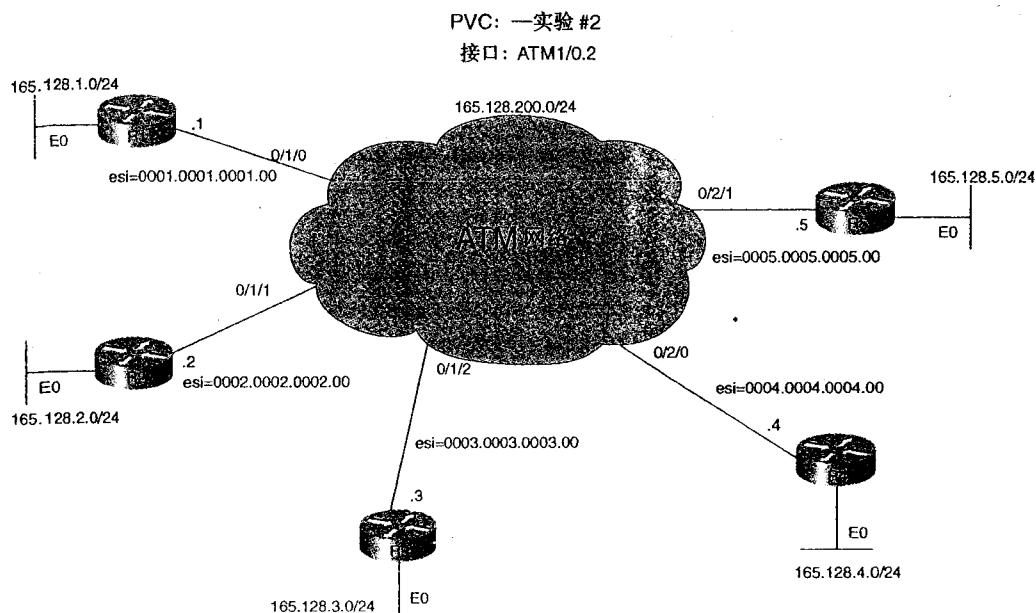


图 8-9 经典 IP

8.7 实验 19：在 Cisco 7XXX 路由器上利用 SVC 对经典 IP、RFC 2225 进行配置——第 2 部分

8.7.1 实验步骤

这里所需要使用的 IP 地址方式和路由选择协议与上一个实验一样，所需要的配置步骤我们也在这里给出了参考。ATM 网络是基于 SVC，网络的 IP 连接必须跨过 ATM 网络部分，建立一个全连接的网络环境。

这个实验演示了利用经典 IP 扩展 IP 连接的情况。除此之外，还采用了 ILMI 动态分配 ATM 地址前缀。

表 8-12 列出了这个实验中需要使用的命令。

表 8-12

实验所需命令汇总

命 令	描 述
Atm pvc	用于信令和 ILMI PVC 的建立
Atm arp-server self	为 LIS 配置 ARP 服务器
Atm arp-server nsap	配置 ARP 客户
Atm esi-address	配置 NSAP（AESA）地址的 ATM ESI 部分

Debug atm event

显示交换机和路由器之间的 PVC 建立情况

续表

命 令	描 述
Debug atm arp	显示 ATM ARP 事件
Show ip route	显示 IP 路由表
Ping	测试第 3 层连接

参考图 8-9 的网络图表，按照下面的步骤配置网络。

1. 第 1 步：配置路由器的参数

需要配置的路由器参数包括：

- 主机名：Rn（n 是路由器编号）。
- 密码：atmlab。
- 虚拟终端/控制台密码：cisco。
- 路由选择协议：EIGRP。
- EIGRP 的自治系统号：100。
- IP 是惟一的被路由协议。

例 8-14 就是路由器 R1 这一个步骤的配置情况。

例 8-14 路由器参数的配置

```
R1(config)# hostname R1
!
enable password atmlab
!
router eigrp 100
 network 165.128.0.0
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password cisco
 login
!
```

2. 第 2 步：为 ATM 1/0.2 接口配置 IP 地址

按照表 8-13 对 ATM 1/0.2 的 IP 地址进行配置。

表 8-13

IP 地址分配

路 由 器	ATM1/0.1
R1	165.128.200.1/24
R2	165.128.200.2/24
R3	165.128.200.3/24
R4	165.128.200.4/24
R5	165.128.200.5/24

例 8-15 就是 R1 上这一步的配置过程。

例 8-15 路由器 R1 的 IP 地址设置

```
R1(config)# interface atm1/0.2 multipoint
ip address 165.128.200.1 255.255.255.0
```

3. 第 3 步：两条 PVC 的配置——每台路由器的主接口上信令和 ILMI

例 8-16 是在 R1 上配置这一步的过程。

例 8-16 PVC 上信令和 ILMI 的配置

```
R1(config)# interface atm1
atm pvc 1 0 5 qsaal
atm pvc 2 0 16 ilmi
interface atm 1/0.2 multipoint
ip address 165.128.200.1 255.255.255.0
```

4. 第 4 步：配置 LS1010 以管理 PVC

在 CCIE 实验考试中，ATM 交换机是预先配置好的。但是，对于准备 CCIE 来说，还是需要自己来对交换机进行配置，如例 8-17 所示。

例 8-17 配置一台管理 PVC 的 ATM 交换机

```
ATM-Switch# version 11.3
no service pad
no service udp-small-servers
no service tcp-small-servers
!
hostname ATM-Switch
!
enable password atmlab
!
no ip domain-lookup
!
atm address 47.0091.8100.0000.0010.0739.a101.0010.0739.a101.00
atm router pnni
node 1 level 56 lowest
redistribute atm-static
!
interface ATM0/1/0
no ip address
no atm auto-configuration
atm uni version 3.1
!
interface ATM0/1/1
no ip address
no atm auto-configuration
atm uni version 3.1
!
interface ATM0/1/2
no ip address
no atm auto-configuration
atm uni version 3.1
```



```
1
interface ATM0/2/0
  no ip address
  no atm auto-configuration
  atm uni version 3.1
!
interface ATM0/2/1
  no ip address
  no atm auto-configuration
  atm uni version 3.1
```

5. 第 5 步: 为路由器接口分配 NSAP 地址的 ESI 部分

这一步是为所有路由器的 atm1/0.2 接口分配其 NSAP 地址的 ESI 部分，如表 8-14 所示。

表 8-14

IP 地址分配

路 由 器	地址的 ESI 部分
R1	0001.0001.0001.00
R2	0002.0002.0002.00
R3	0003.0003.0003.00
R4	0004.0004.0004.00
R5	0005.0005.0005.00

例 8-18 就是这一步操作在 R1 上的体现。

例 8-18 将 NSAP 地址的 ESI 部分映射到 atm1/0.2 接口

```
R1(config)# interface atm1
  atm pvc 1 0 5 qsaal
  atm pvc 2 0 16 ilmi
interface atm 1/0.2 multipoint
  ip address 165.128.200.1 255.255.255.0
  atm esi-address 0001.0001.0001.00
```

6. 第 6 步: 定义 ATM ARP 服务器的地址

这一步包括在 R1, R2, R3, R4 和 R5 上定义 ATM ARP 服务器的地址，并且把 R2 定义为 ATM ARP 服务器。

R2 是 ATM ARP 服务器，R1, R3, R4 和 R5 都要依靠它来对 IP 和 ATM 地址做出解析。

例 8-19 显示了如何把 R2 配置为一台 ATM ARP 服务器。

例 8-19 将 R2 配置为一台 ATM ARP 服务器

```
R2(config)# interface atm1
  atm pvc 1 0 5 qsaal
  atm pvc 2 0 16 ilmi
interface atm 1/0.2 multipoint
  ip address 165.128.200.2 255.255.255.0
  atm esi-address 0002.0002.0002.00
  atm arp-server self
```

R1, R3, R4 和 R5 都是客户机。客户机的配置中必须含有 ATM ARP 服务器的 NSAP 地址。例 8-20 就是如何用 R1 来配置客户机的例子。

例 8-20 ATM ARP 服务器的客户端的配置

```
R1(config)# interface atm1
  atm pvc 1 0 5 qsaal
  atm pvc 2 0 16 ilmi
interface atm 1/0.2 multipoint
  ip address 165.128.200.1 255.255.255.0
  atm esi-address 0001.0001.0001.00
  atm arp-server nsap 47.0091.8100.0000.0010.0739.a101.0002.0002.0002.00
```

注意上面这个例子中 R2 创建 NSAP 地址时利用了来自 LS1010 的前缀和分配给 R2 的 ESI 部分。

7. 第 7 步：对所做的配置进行测试

8.8 总 结

Cisco 通过不同的产品对 ATM 网络上各种各样的路由器连接都提供了支持，这一章曾经对此进行了总结。此外，这一章的内容还包括了在以 PVC 和 SVC 为基础的 ATM 网络上对多协议封装（RFC 2684）和经典 IP（RFC 2225）的实施，给出了相关的 Cisco IOS 命令以及相应的实例。

多协议封装形式能够在 ATM 网络的第 3 层和第 2 层之间建立起互连关系。这一章中的例子侧重第 3 层的支持。对于第 2 层的支持，也就是 LANE，不在本章的范围之内，而且也不再是 CCIE 实验考试的内容。

经典 IP（RFC 2225）的讲述重点只是在 IP 上，它允许在 IP 和 ATM 之间建立动态的地址映射关。

第4部分

路由选择协议

路由选择协议是 Internet 网络的胶合剂，是路由器的基础，也是 Internet 发展到现在如此庞大规模的基石。对路由选择协议的掌握程度在设计和实施 IP 网络时具有关键性作用。下面的章节会讲述什么是路由选择协议，路由选择协议的各种类型和以及它们使用的度量值 (metric)，并且将对两种主要路由选择协议——距离矢量协议和链路状态协议进行比较。

本书第 4 部分包括：

第 9 章：“距离矢量协议：路由信息协议版本 1 和版本 2 (RIP-1 和 RIP-2)”

第 10 章：“距离矢量协议：内部网关路由选择协议 (IGRP)”

第 11 章：“混合协议：增强型内部网关路由选择协议 (EIGRP)”

第 12 章：“链路状态协议：开放式最短路径优先协议 (OSPF)”

什么是路由选择协议

如果协议中包含明确的网络地址，并且网络层地址中有足够的信息让路由器做出智能的转发决定，那该协议就是一个被路由协议。路由就是数据包从一个网络传输到另一个网络的过程。路由选择协议通过提供传输路由途径的信息来支持被路由协议。路由信息包括可用的路由、路由开销和下一跳地址。路由选择协议通过在路由器之间交换消息来更新和维持路由表。有一点很重要，路由选择协议并不会将终端用户的数据从一个网络传输到另一个，它只是建立终端用户的数据进行传输的路径。

路由表

目前的路由选择协议有很多，它们之间互不相同，但是目的都一样，即进行路由操作，维护路由表。

路由表包含下列信息：

- 网络/路由或主机路由列表。
- 学习路由的途径：从路由选择协议动态获取，或者通过手动设置静态路由。
- 路由的管理距离。
- 路由的度量或开销。
- 路由的下一跳地址。
- 路由的当前状态，包括上次路由更新后的时间，路由是否保持等等。
- 与实现该路由相关的接口。即数据包转发至下一跳地址的接口。

例 IV-1 描述了一份复杂的路由表，包括了 OSPF、EIGRP 以及默认路由等。

例 IV-1 路由表

```
r2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.128.1 to network 0.0.0.0

  10.0.0.0/24 is subnetted, 1 subnets
O      10.10.10.0 is a summary, 03:05:49, Null0
  129.201.0.0/24 is subnetted, 1 subnets
O E1   129.201.1.0 [110/90] via 172.16.2.66, 03:05:45, TokenRing1
  128.200.0.0/24 is subnetted, 1 subnets
D EX   128.200.1.0 [170/679936] via 172.16.192.3, 05:42:57, Serial1
  129.200.0.0/24 is subnetted, 1 subnets
O E1   129.200.1.0 [110/90] via 172.16.2.66, 03:05:45, TokenRing1
  128.201.0.0/24 is subnetted, 1 subnets
D EX   128.201.1.0 [170/679936] via 172.16.192.3, 05:42:57, Serial1
C      201.201.101.0/24 is directly connected, Loopback0
```

(待续)

```

O E2 132.31.0.0/16 [110/2] via 172.16.2.66, 00:58:04, TokenRing1
O E2 131.31.0.0/16 [110/2] via 172.16.2.66, 00:58:04, TokenRing1
    172.16.0.0/16 is variably subnetted, 27 subnets, 4 masks
O IA 172.16.152.0/24 [110/71] via 172.16.2.66, 03:05:45, TokenRing1
O IA 172.16.150.0/24 [110/80] via 172.16.2.66, 03:05:45, TokenRing1
O IA 172.16.151.0/24 [110/71] via 172.16.2.66, 03:05:45, TokenRing1
C 172.16.144.0/21 is directly connected, Loopback20
C 172.16.136.0/21 is directly connected, Ethernet1
C 172.16.128.0/21 is directly connected, Ethernet0
C 172.16.192.0/24 is directly connected, Serial1
C 172.16.192.3/32 is directly connected, Serial1
O IA 172.16.42.2/32 [110/70] via 172.16.2.66, 03:05:46, TokenRing1
O IA 172.16.42.3/32 [110/70] via 172.16.2.66, 03:05:46, TokenRing1
O E2 172.16.42.0/24 [110/2] via 172.16.2.66, 03:05:46, TokenRing1
O IA 172.16.42.1/32 [110/6] via 172.16.2.66, 03:05:46, TokenRing1
O IA 172.16.21.0/24 [110/76] via 172.16.2.66, 03:05:46, TokenRing1
O IA 172.16.22.0/24 [110/71] via 172.16.2.66, 03:05:46, TokenRing1
O E2 172.16.1.0/24 [110/2] via 172.16.2.66, 03:05:46, TokenRing1
O E2 172.16.2.0/24 [110/2] via 172.16.2.66, 03:05:46, TokenRing1
D 172.16.102.0/24 [90/679936] via 172.16.192.3, 05:42:59, Serial1
D 172.16.103.0/24 [90/409600] via 172.16.128.1, 05:42:59, Ethernet0
O E2 172.16.84.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2 172.16.85.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2 172.16.81.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2 172.16.82.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2 172.16.83.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2 172.16.64.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
C 172.16.1.64/26 is directly connected, TokenRing0
C 172.16.2.64/26 is directly connected, TokenRing1
D*EX 0.0.0.0/0 [170/20028160] via 172.16.128.1, 05:43:00, Ethernet0

```

在突出显示的行上可以看到许多上面提到过的路由表组件。路由 172.16.102.0/24 由 EIGRP 报告，管理距离是 90，EIGRP 的度量是 679936。该路由是 5 小时 42 分钟前报告的，通过接口 Serial 1 到达其下一跳路由器，IP 地址是 172.16.192.3。

路由选择协议算法

所有的动态路由选择协议都是围绕一些通用算法建立的。路由算法包括以下内容：

- 用于发布来自或去往其他某一路由器的网络可达性信息的过程。
- 基于来自其他路由器的可达性信息，用于确定和记录最佳路由的过程。
- 用于发布和适应网络拓扑结构变化的过程。

路由器何时处理路由？

默认情况下，所有的 Cisco 路由器首先对所有可路由的协议如 IP 或 IPX 进行路由。如果没有允许对可路由的协议进行路由，但允许了对该协议的桥接，路由器就会对该协议进行桥接操作。路由器如何处理协议，不仅取决于该协议是否可路由，而且还取决于该协议的路由是如何设置的。例如，如果路由器收到一个 IPX 数据包，并允许 IPX 路由，那么就会试着将该数据包路由或转发到下一跳地址。这是通常的 IPX 操作。但是，如果路由器接收到了一个

DLSw 端口。这种情况下，路由器不再把 IPX 看成是可路由的协议。

要成功的对数据包进行路由，路由器必须掌握下面这些信息：

- 协议必须是可路由的，而且对该协议的路由设置为允许。
- 路由器必须知道目的网络或者设置了默认路由。
- 必须存在合法的下一跳地址或者指向目的网络的接口。

当确定要对某个数据包进行路由时，路由器会检查数据包，看其目的地址是否在本地连接的网络中。如果目的地址在本地网络中，路由器会直接通过适当的端口转发数据包。如果目的地址不在本地网中，路由器会查询路由表。路由表包含该路由器已知的网络、与这些网络相关联的路由开销以及到下一跳路由器的路径。路由器通过最长匹配查找方式比较数据包与其路由表条目。与目的地址存在着最长匹配关系的项被用来确定转发数据包的路径。最长匹配路由也称为最准确路由，最长匹配方式只有在设置了 **ip classless** 时才完全正确，其适用于绝大多数无类路由选择协议。

注释 当路由器需要确定究竟使用其路由表中的哪一项来进行数据包的路由时，需要使用最长匹配查询。例如，假设路由器接收到了一个目的地址是 172.16.1.0/24 的数据包，要完成将该数据包转发到其目的地址的任务，共有两条路由途径，一条是通过 172.0.0.0/8 从 S1 转发，另一条是通过 172.16.0.0/16 从 S2 转发。路由器会采用哪个接口来转发该数据包呢？这时候，路由器要进行最长匹配查询。路由器会将目的地址值的每一位从左到右与路由表中各路由的位进行比较。路由器按顺序对每一位进行比较，到第一次发现与所比较的路由表项的位不相同停止比较。此时，路由器选择连续匹配位数最大的路径/路由进行数据包的路由。在这个例中，路由器会采用 172.16.0.0 的路由而不是 172.0.0.0，因为 172.16.0.0 路由是最长匹配的，或者换句话说，更明确。

路由度量

路由选择协议的另一个重要属性是在确定每个目的网络的最佳路径时提供给网络一个无环路拓扑结构。路由器以度量为单位来宣告到达某个网络的路径。度量值或者度量类型视路由选择协议而定。例如，RIP 将跳数作为其度量，而 OSPF 则用路由开销来作为度量。路由器在对通往同一网络去的多个路径进行评估时使用度量值。所有路由选择协议的度量值都是可调的，从而影响路由器对转发数据包路径的选择。

下面对常用的路由度量做简单描述：

- **跳数**——经过路由器跳数的度量。跳数越大，路径越不理想。
- **带宽**——测量带宽的度量。带宽值越高，路径越理想。
- **负载**——反映去往某条路径的链路上数据量的度量。负载越低，路径越理想。
- **延时**——数据包经过某条路由所花费时间的度量。延时越低，路径越理想。
- **可靠性**——反映某条路径出现问题几率的度量。可靠性越高，路径越理想。
- **路由开销**——一个可配置的度量。开销越低，路径越理想。

每个路由选择协议都有各自的度量方法，后面的章节中将讨论与路由选择协议相关的度量问题。

管理距离

任何时候，每台路由器上可以有多个路由选择协议同时工作，路由器需要区分从不同路由选择协议接收到的路由。Cisco 引入了**管理距离**的概念来衡量 IP 路由信息源的可信度。管理距离的值越低，该路径就越可靠。使用 **distance** 命令可以更改路由选择协议的管理距离。表 IV-1 列出了路由源的默认管理距离值。

表 IV-1 Cisco 路由器上的默认管理距离

路由源/类型	默认管理距离
直连接口	0
指向接口的静态路由	0
指向下一跳接口的静态路由	1
EIGRP 汇总路由	5
外部 BGP	20
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP-1, RIP-2	120
EGP	140
外部 EIGRP	170
内部 BGP	200
未知路由 不可达	255

距离矢量和链路状态协议

大多数的路由选择协议都可以归入下面两类：

- 距离矢量协议。
- 链路状态协议。

Cisco 的 EIGRP 是一个例外，通常称为**混合协议**，因为它综合了链路状态和距离矢量协议的一些特点，但它更接近距离矢量协议，与链路状态协议差别要大一些，因为 EIGRP 路由宣告采用距离作为度量而不是链路状态。

距离矢量协议

基于距离矢量的算法通常被称为 Bellman-Ford 算法，周期性地在路由器之间传送路由表的拷贝。通常的路由更新是在网络拓扑结构发生改变时产生的。距离矢量协议以**矢量**对路由进行衡量。每个矢量都有与之相关联的距离和方向。例如，在 RIP 中，子网 172.10.0.0 是一个**矢量**，其距离是 5 跳，其方向是下一跳路由器。

距离矢量协议的简单工作方式如下：

1 每隔一定的时间间隔，路由器通过每个接口广播整个路由表，这里不包括被过滤器和水平分割抑制掉的路由或者是与发送广播的接口处在不同的主网边界的路由。广播的内容包括网络或路由以及与该路由相关联的度量。

2 邻居路由器接收到路由更新信息，并且将更新信息中的路由与路由表中的路由进行比较。具有最佳度量——最低度量值的路由被存到转发表中。

3 邻居路由器将其新的路由表发布到它的所有邻居路由器。

4 路由器继续周期性地向邻居路由器广播路由表。以 RIP 为例，这个周期是 30 秒。

所有的距离矢量协议都具有以下共同特性：

- 周期性全路由更新——某个特定时限的终点，通常是 10 到 90 秒，通过网络广播发送完整的路由表到邻居路由器。
- 邻居路由器——邻居路由器可以定义为共享同一数据链路的路由器。距离矢量路由器会将路由更新信息发送到所有的邻居路由器，而这些邻居路由器又会将路由表发送到它们的邻居路由器。
- 广播更新——路由器使用广播地址查找其邻居路由器并发送路由表。
- 路由失效计时器——这些计时器提供了路由器开始对其路由信息价值进行降级处理，并最终将其从路由表中剔除的方式。收到新的路由更新时，计时器复位。
- 水平分割——将某条接收到的路由回送到发送这条路由的路由器上，这种路由现象称为反向路由。水平分割避免了路由器之间出现反向路由。简单水平分割的规则是在将一个路由更新通过某个接口或者是子接口发送出去时，不包括从该接口处获取的网络信息。所有的串行接口或子接口上水平分割都是默认允许的。在本书整个过程中都会讲述水平分割的问题，尤其是水平分割与帧中继中点到点和点到多点接口的配合问题。
- 最大跳数或无限计数——距离矢量网络的跳数上限值，达到该跳数时，路由被认为不可到达。RIP 的最大跳数是 16。
- 逆向毒化——逆向毒化的规则是将路由通过接收到该路由的接口发送出去，发送时给此路由的度量属性设置为不可到达。不同的路由选择协议在不同的时间应用这条规则来避免路由环路。

下面是距离矢量路由选择协议列表：

- IP 路由信息协议 (RIP)。
- Xerox 网络系统 XNS RIP。
- IPX RIP。
- Cisco 的内部网关路由选择协议 (IGRP)。
- DEC 的 DNA Phase IV。
- AppleTalk 的路由表维护协议 (RTMP)。
- 网关到网关协议 (GGP)。
- 外部网关协议 (EGP)。

链路状态协议

另一类路由选择协议称为链路状态协议。距离矢量协议基于 R.E. Bellman 和 R. Ford 和 D.R. Fulkerson 算法，链路状态协议基于 E.W. Dijkstra 算法。

链路状态协议的工作方式和距离矢量协议有很大的区别，主要区别如下：

- 支持可变长子网掩码（VLSM）——所有链路状态协议都支持 VLSM。主要因为路由更新中包括子网掩码。
- 邻居路由器——所有的链路状态协议通过 Hello 协议来建立邻居。
- 非存根路由器——Nonstub 路由器中保存网络中所有路径的完整映射。
- 事件触发的路由宣告——路由更新通过从一个区域到下一个区域的路由状态更新发送来传输。LSA 传输的方向由 SPF/Dijkstra 算法确定。
- 链路状态数据库——链路状态数据库将链路状态发布（LSA）信息以列表的方式存储，数据库中的信息包括路由器 ID 的记录，直连网络，邻居路由器及与之相应的路由开销。
- 分层拓扑结构——外部区域需要与骨干区域建立连接。链路状态网络必须围绕这些要求来设计。OSPF 虚链路可以打破这一规则，但应该尽量避免使用这种做法。

以下是链路状态协议工作的简单过程：

- 1 路由器与每个邻居路由器建立邻居关系。
- 2 路由器将其链路状态宣告（LSA）或链路状态数据包（LSP）发送到邻居路由器。表中的每个路由都会产生一条 LSA 信息。LSA 提供了路由标识、状态、路由度量以及与此路由相连的所有邻居路由器。接收到该信息的邻居路由器又会将信息转发给其邻居路由器。
- 3 路由器存储从数据库中接收到的所有的 LSA。
- 4 该数据库称作链路状态数据库，含有整个网络的拓扑图。路由器采用 Dijkstra 算法来计算到每个网络的最短路径，然后将此信息输入到路由表中。

最常见的链路状态协议如下：

- IP 的开放式最短路径优先（OSPF）协议。
- ISO IS-IS 的中间系统-中间系统。
- DEC 的 DNA Phase V。
- Novell 的 NetWare 链路服务协议(NLSP)。

距离矢量与链路状态路由选择协议的比较

表 IV-2 给出了距离矢量与链路状态这两种路由选择协议的主要区别。

表 IV-2

IP 路由选择协议的比较

	RIP-1	RIP-2	IGRP	EIGRP	OSPF	IS-IS
水平分割	X	X	X			
周期性更新	X	X	X			

续表

	RIP-1	RIP-2	IGRP	EIGRP	OSPF	IS-IS
触发更新	X	X	X	X	X	X
支持 VLSM		X		X	X	X
负载平衡			X	X		
自动有类路由汇总	X*	X	X	X		
手动无类路由汇总				X	X	X
路由度量	跳数	跳数	延时, MTU, 负载, 带宽, 可靠性	延时, MTU, 负载, 带宽, 可靠性	开销	默认, 延时, 开销, 差错
认证类型 类型 1-明文 类型 2-MD5	无	类型 1 类型 1	无	类型 2	类型 1 类型 2	类型 1
最大跳数	15	15	255	255	无限	1024
分层设计要求					X	X
可扩展性	小	小	中	大	大	很大
路由算法	Bellman-Ford	Bellman-Ford	Bellman-Ford	DUAL	Dijkstra	IS-IS
Cisco 管理距离	120	120	100	90/5**	110	115

* 无法关闭路由汇总。

** 5 是 EIGRP 汇总路由的管理距离。

下面各章讲述了 RIP、IGRP、EIGRP 和 OSPF 的配置。可以参考 Jeff Doyle 的《Routing TCP/IP 卷 1、2》来全面深入了解这些协议的内容。

第 9 章

距离矢量协议： 路由信息协议版本 1 和版本 2（RIP-1 和 RIP-2）

进入 21 世纪后，RIP 这个最早的路由选择协议之一还在很多现代网络中使用。RIP 成功地从它诞生之日起一直适用到现在。这证明了一件事情，尽管有其缺陷，RIP 仍然能够很好地完成工作，满足人们的需要。经过这些年的发展，RIP 已经从有类路由选择协议（RIP-1）演化成无类路由选择协议（RIP-2）。本章讲述 RIP-1 和 RIP-2 的工作方式、配置过程以及调优方法，还有关于路由重分布的内容。

9.1 RIP 技术概览

RFC 1058 制订了 RIP-1，RFC 1721、1722 和 1723 则是 RIP 2 的扩充内容。

注释 <http://www.isi.edu/in-notes/rfcxxxx.txt> 上可以找到所有的 RFC 规范的内容，这里的 xxxx 是指 RFC 的编号。

RIP 工作在 UDP 的端口 520 上——也就是说，所有的 RIP 数据包的源端口和目的端口都是 520。RIP 的工作过程如下：

1 初始化——RIP 初始化时，会从每个参与工作的接口上发送 **请求数据包**。该请求数据包会向所有的 RIP 路由器请

求一份完整的路由表。该请求通过 LAN 上的广播形式发送 LAN 或者在点对点链路发送到下一跳地址来完成。这是一个特殊的请求，向相邻设备请求完整的路由更新。

2 接收请求——RIP 有两种类型的消息：*响应*和*接收消息*。请求数据包中的每个路由条目都会被处理，从而为路由建立度量以及路径。RIP 采用跳数度量，值为 1 的跳数意味着一个直连的网络，如果值为 16，网络不可到达。路由器会把整个路由表作为接收消息的应答返回。

3 接收到响应——路由器接收并处理响应，它会通过对路由表项进行添加，删除或者修改作出更新。

4 常规路由更新和定时——路由器第 30 秒一次地将整个路由表以应答消息的形式发送到邻居路由器。路由器收到新路由或者现有路由的更新信息时，会设置一个 180 秒的*超时时间*（或称*失效时间*）。如果 180 秒内该路由没有任何更新信息，路由的跳数设为 16（不可到达）。路由器以度量值 16 宣告该路由，直到刷新计时器从路由表中删除该路由。*刷新计时器*的时间设为 240 秒，或者比过期计时器时间多 60 秒。Cisco 的 RIP 配置没有在 RFC 1058 中定义，Cisco 还使用了第 3 个计时器，称为*抑制计时器*。接收到一个度量更高的路由之后的 180 秒时间就是抑制计时器的时间，在此期间，路由器不会用它接收到的新信息对路由表进行更新，这样能够为网络的收敛提供一段额外的时间。

5 触发路由更新——当某个路由度量发生改变时，路由器只发送与改变有关的路由，并不发送完整的路由表。

所有的计时器都可以通过 *timers basic update invalid holddown flush* 命令来设置。

警告 在调整 RIP 的计时器时，整个路由域中的所有路由器计时器都必须作相应的修改，否则就有可能产生无法预料的后果。

9.1.1 有类路由（仅 RIP-1）

RIP-1 是一个有类路由选择协议，因此路由宣告中不携带子网掩码。RIP-1 采用接收路由的接口的子网掩码来确定目的网络的子网掩码。这种做法仅对接收到的路由和直连网络处于同一主网的情况有效。如果接收到的路由不是同一个主网，路由器就会试着去匹配该路由的主网掩码，可能是 A、B 或 C 类。因此，在整个 RIP 路由域中保持每个主网掩码长度的一致性很重要。

图 9-1 表示 RIP-1 网络上有类路由的情况。看一下路由器 wolverine 和 rogue 处理 RIP 的过程。

路由器 wolverine 在主网 128.200.0.0 中有两个接口，每个接口上的 24 位掩码一致。因此，路由器只会接收属于主网 128.200.0.0 的 24 位掩码的路由更新信息。当路由器接收到属于另一个主网（如 192.16.1.4/30）的路由更新时，会将主网掩码位或者是地址的类边界掩码的一个汇总路由加入到其路由表中去，该地址汇总为 192.16.1.0/24。例 9-1 给出了 *debug ip rip* 命令的输出信息，以解释这一概念。

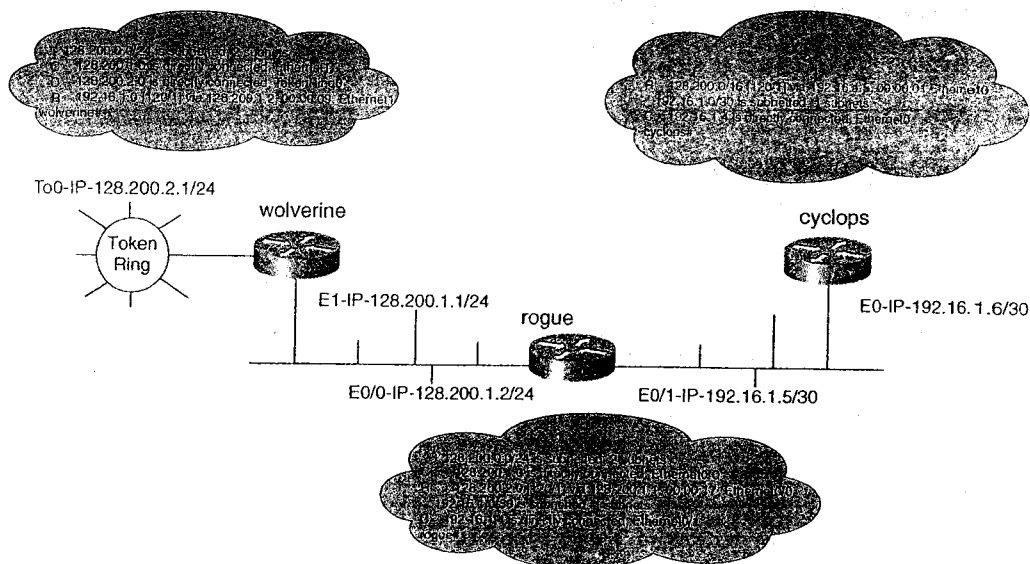


图 9-1 有类路由的例子

例 9-1 路由器 wolverine 上 debug ip rip 命令的输出信息

```
01:24:39: RIP: sending v1 update to 255.255.255.255 via Ethernet1 (128.200.1.1)
01:24:39: subnet 128.200.2.0, metric 1
01:24:39: RIP: Update contains 1 routes
01:24:39: RIP: Update queued
01:24:39: RIP: Update sent via Ethernet1
01:24:39: RIP: sending v1 update to 255.255.255.255 via TokenRing0 (128.200.2.1)
01:24:39: subnet 128.200.1.0, metric 1
01:24:39: network 192.16.1.0, metric 2
01:24:39: RIP: Update contains 2 routes
01:24:39: RIP: Update queued
01:24:39: RIP: Update sent via TokenRing0
```

路由器 rogue 也有两个接口。一个在主网 128.200.0.0/16 中，另一个在网络 192.16.1.0/24 中。当 rogue 从 E0/0 接口接收到 128.200.2.0/24 和 128.200.1.0/24 子网时，它会试着从接口 E0/1 把它们发送出去。由于该接口的掩码（30 位）不同，只有汇总路由 128.200.0.0/16 被发送给路由器 cyclops。例 9-2 给出了通过 **debug ip rip** 命令显示这一过程的例子。

例 9-2 rogue 路由器上 debug ip rip 命令的输出信息

```
RIP: received v1 update from 128.200.1.1 on Ethernet0/0
128.200.2.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (128.200.1.2)
network 192.16.1.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Ethernet0/1 (192.16.1.5)
network 128.200.0.0, metric 1
```

9.1.2 无类路由（仅 RIP-2）

充。这些扩展提供以下对 RIP 功能的增强：

- 支持 VLSM。路由器在路由更新中包含了子网掩码，使得路由器能够处理 VLSM 寻址。
- 每个路由条目中都携带下一跳地址。
- 支持外部路由标签。
- 多播路由更新。
- 支持 MD5 认证。

这些增强功能中最重要的是对 VLSM 的支持，使 RIP-2 成为无类路由选择协议，从而无需在整个路由域中保持掩码的一致性。

RIP-2 的多数工作过程和计时器和 RIP-1 完全相同。RIP-2 用来发送路由更新的多播地址是 224.0.0.9，而 RIP-1 使用包含全部主机在内需要接受的广播地址。

RIP-2 对 RIP-1 是完全向后兼容，这通过兼容交换机制和接收控制交换机制来实现，RFC 1723 中制订了这方面的规则。根本上说，这些交换机制能让用户控制路由器接收和发送的 RIP 更新信息的类型。路由器可以配置为只接收版本 1 的更新信息，只接收版本 2 的更新信息，两者都可以接收，或者是两者都不接收；仅发送版本 1 更新信息，以广播的形式发送版本 2 更新信息，发送版本 2 更新信息作为多播信息，或者是不发送任何更新信息。可以使用下面的接口命令来手动配置交换机制：

```
ip rip [send | receive] version [1 | 2 | 1 2]
```

9.2 RIP-1 和 RIP-2 的配置

RIP-1 和 RIP-2 的配置十分简单：

第 1 步 在路由器上启动 RIP 并指定 RIP 的版本号。启动 RIP 协议的是全局命令 **router rip**。除此之外，还要定义 RIP 的版本号。如果要使用 RIP-1，不需做额外设置。

如果采用 RIP-2，则要在 **config-router#** 提示符下使用 **version 2** 命令。

第 2 步 在 **config-router#** 提示符下用 **network a.b.c.d** 命令添加运行 RIP 的网络。

9.2.1 RIP-1 的配置

例 9-3 是对图 9-1 的网络进行 RIP-1 配置的例子。

例 9-3 RIP-1 的配置

```
hostname wolverine
!
router rip
 network 128.200.0.0
!
-----
hostname rogue
!
```

(待续)

```

router rip
 network 128.200.0.0
 network 192.16.1.0
!

hostname cyclops
!
router rip
 network 192.16.1.0
!

```

9.2.2 RIP-2 的配置

现在把图 9-1 中的网络升级为 RIP-2 网络。在 wolverine 路由器上，配置在令牌环网络上发送和接收 RIP-1 和 RIP-2 的路由更新信息。但是，wolverine 以外的以太网段只能收发版本 2 的路由更新信息。路由器 rogue 和 cyclops 都配置成只能收发 RIP-2 更新信息。例 9-4 中的路由器配置就可以满足这些要求。

例 9-4 RIP-2 的配置

```

hostname wolverine
!
interface Ethernet1
 ip address 128.200.1.1 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
 media-type 10BaseT
!
interface TokenRing0
 ip address 128.200.2.1 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
 ring-speed 16
!
router rip
 version 2
 network 128.200.0.0
 no auto-summary
!

hostname rogue
!
interface Ethernet0/0
 ip address 128.200.1.2 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
!
interface Ethernet0/1
 ip address 192.16.1.5 255.255.255.252
 ip rip send version 2
 ip rip receive version 2
!
router rip
 version 2
 network 128.200.0.0
 network 192.16.1.0

```

(待续)

```

no auto-summary

hostname cyclops
!
interface Ethernet0
 ip address 192.16.1.6 255.255.255.252
 ip rip send version 2
 ip rip receive version 2
!
router rip
 version 2
 network 192.16.1.0
 no auto-summary
    
```

图 9-2 给出了网络升级为 RIP-2 之后路由表的变化。要注意在路由器 wolverine 和 cyclops 上反映的各个子网的情况。即使要发送 RIP-2 更新信息，也必须用 **no auto-summary** 命令来禁止在主网边界上自动汇总。

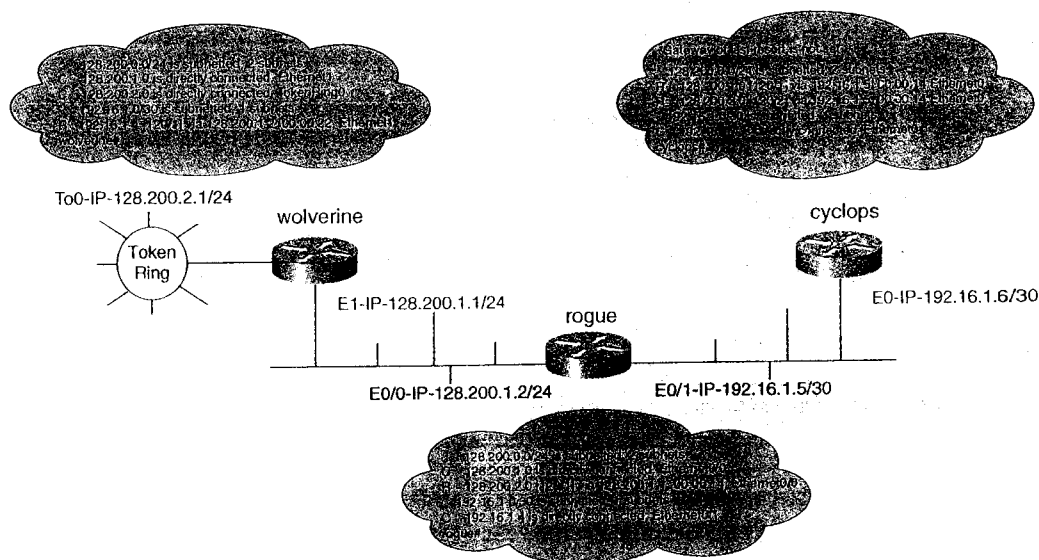


图 9-2 无类路由的例子

9.2.3 RIP 的 “Big show” 和 “Big D”

RIP 的调试是件很容易的事情。大多数的 RIP 配置错误可以归纳为 **network** 声明不正确，子网不连续或者是主网分离。由于多数这些错误都与设计有关，因此 RIP 的 **debug** 和 **show** 命令很有限，下面的这个命令列表也不完全，这是一些很有用的命令。RIP 的 **Big show** 和 **Big D** 命令如下：

- **show ip protocols {summary}**
- **show ip route**
- **debug ip rip {events}**

9.2.4 show ip protocols {summary}命令

该命令能够显示所有的路由选择协议、详细的计时器和度量信息以及路由更新信息。例9-5列出了该命令的执行结果。

例 9-5 show ip protocols 命令的执行结果

```

rogue#show ip protocols
Routing Protocol is "rip"    <-Routing Protocol Type
  Sending updates every 30 seconds, next due in 29 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240    <-Timer information
  Outgoing update filter list for all interfaces is    <-Distribute list (if any)
  Incoming update filter list for all interfaces is
  Default redistribution metric is 2    <-Default metric
  Redistributing: rip, eigrp 2001    <-Redistribution is on
  Default version control: send version 1, receive any version
    Interface      Send Recv  Key-chain
  Ethernet0/0      1        1.2    <-RIP Versions running
  Routing for Networks:    <-Networks participating in RIP
    128.200.0.0
  Passive Interface(s):
    Ethernet0/1    <-Network listening to RIP
  Routing Information Sources:
    Gateway      Distance    Last Update
    128.200.1.1    120        00:00:07 <-RIP Neighbors
  Distance: (default is 120)    <-Administrative Distance
    
```

9.2.5 show ip route 命令

该命令能够列出路由器当前的路由表以及路由器用来转发数据包的路由信息。路由器中可能存在关于某个路由的多个条目，但列出的只能是管理距离最小的路由。该命令的执行结果列出了路由来自哪个路由选择协议，如例9-6所示，例子中的R代表RIP。路由后边的数字是该路由的管理距离，接着是跳数。Via 字段解释了路由源自何处、路由更新信息的接收时间和接收接口。例9-6列出了路由器 rogue 上 show ip route 命令的结果。

例 9-6 show ip route 命令的执行结果

```

rogue#show ip route
Gateway of last resort is not set

  128.200.0.0/16 is variably subnetted, 4 subnets, 2 masks
R    128.200.10.0/24 [120/1] via 128.200.1.1, 00:00:17, Ethernet0/0
C    128.200.1.0/24 is directly connected, Ethernet0/0
R    128.200.2.0/24 [120/1] via 128.200.1.1, 00:00:17, Ethernet0/0
C    128.200.3.16/29 is directly connected, Ethernet0/1
rogue#
    
```

该例中，路由 128.200.10.0/24 的度量是 120，跳数是 1。发送该路由的 RIP 邻居路由器是

128.200.1.1，在 12 秒之前发送了上一个更新信息，rogue 路由器通过以太口 E0/0 接收该路由。

9.2.6 debug ip rip {events}命令

该命令能够显示路由器中所有的 RIP 活动，还可以显示发送和接收路由所用的接口，更新信息的 RIP 版本以及更新信息中每条路由的度量也通过该命令显示。例 9-7 列出了 **debug ip rip** 命令的执行结果。请注意 RIP 正在接收和发送路由信息。

例 9-7 debug ip rip 命令的执行结果

```
wolverine#debug ip rip
1d02h: RIP: received v1 update from 128.200.10.2 on TokenRing1
1d02h:      subnet 128.200.10.0, metric 1
1d02h: RIP: sending v1 update to 255.255.255.255 via Ethernet1 (128.200.1.1)
1d02h:      subnet 128.200.10.0, metric 1
1d02h:      subnet 128.200.2.0, metric 1
1d02h: RIP: sending v1 update to 255.255.255.255 via TokenRing0 (128.200.2.1)
1d02h:      subnet 128.200.10.0, metric 1
1d02h:      subnet 128.200.1.0, metric 1
1d02h: RIP: sending v1 update to 128.200.10.2 via TokenRing1 (128.200.10.1)
1d02h:      subnet 128.200.10.0, metric 1
1d02h:      subnet 128.200.1.0, metric 1
1d02h:      subnet 128.200.2.0, metric 1
```

9.3 RIP 更新信息的调整、重分布和控制

RIP 提供了调整计时器和控制广播及路由的参数。下面是 RIP 调整时的一些常用参数：

- Router (config-router) **#timers basic update invalid holddown flush**——该参数使得用户可以设置 RIP 的更新、失效、抑制和刷新计时器的值。
- Router (config-router) **#passive-interface interface_name**——该命令能够禁止在某个接口上发送路由更新信息，但路由器仍然会在该接口上监听并接收更新信息。
- Router (config-router) **#neighbor ip-address**——该命令能够定义一个 RIP 邻居路由器来与之进行单播更新信息的交换，它要和 **passive-interface** 命令配合使用。
- Router (config-router) **#offset-list [access_list_0-99 {in|out} offset [metric_offset_1-16]**
这条命令可以用来增加路由度量的值。度量的增加值不能超过 16。

下面这些命令不是 RIP 专用，其他路由选择协议也可以使用这些设置：

- Router (config-router) **#distribute-list [1-199] [in | out] [interface]**——该命令可以调用标准或扩展访问控制列表以对输入或是输出的路由更新信息进行过滤。
- Router (config-router) **#distance [1-255] adjacent_neighbors_ip_address wildcard_mask [access_list_0-99]**——这条命令可以用来改变从某个邻居路由器接收到路由条目的管理距离。如果 IP 地址和通配符这两个参数都没有指定，那么，该协议所有路由的管理距离都会被设置成这条命令指定的值。
- Router (config-router) **#redistribute [connected, static, bgp, igrp, eigrp, ospf, isis] [metric] [route-map]**——用这条命令能够将别的路由选择协议重分布到 RIP 中，可以添加路由图来作为路由的附加控制。如果协议使用了某个度量，这条命令能够设置度量为这个特定的协议和特定的自治系统所重分布的度量。

布的选项是使用 **default-metric** 命令。对路由进行重分布时，记住 IP 需要一条能够通往某个目的地址并能从该目的地址返回的路由，一般来说，需要对路由进行双向重分布。

- Router (config-router) **#default-metric** [1-16]——这条命令可以用来设置分布到 RIP 中的所有路由的默认度量值。只要进行重分布，就需提供一个默认的度量值。

为了查看这些命令如何工作，我们将这些概念运用到现有的实验中。图 9-3 对路由器 **rogue** 和 **cyclops** 到 128.200.3.16/29 之间的子网进行了修改，该网段上运行的不再是 RIP 协议，而是 EIGRP。在路由器 **rogue** 的 EIGRP 和 RIP 协议之间进行重分布。如图 9-3 所示，另一台路由器 **storm** 加入到网络。在 **storm** 和 **wolverine** 之间的令牌环网段上使用的不再是广播式更新信息，而是单播式的更新信息。

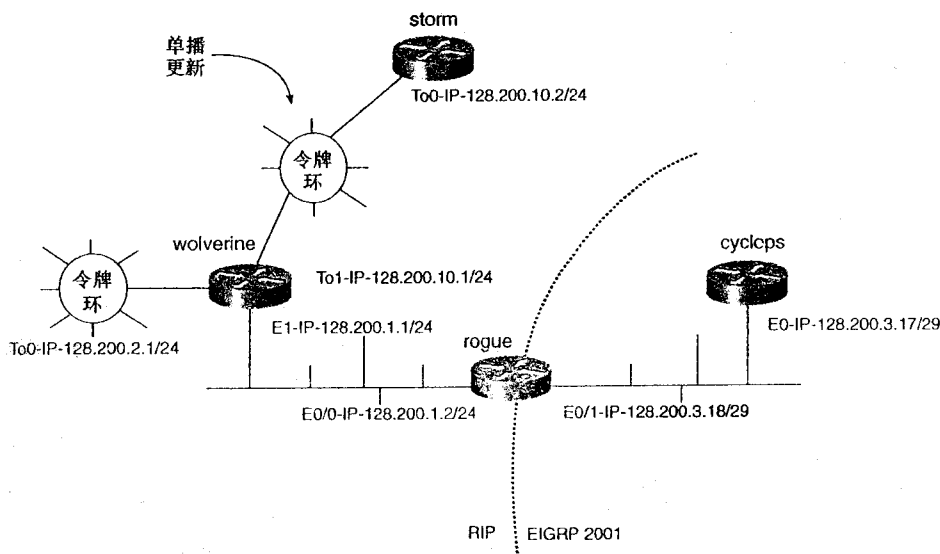


图 9-3 路由重分布和单播传送的例子

要在 **storm** 和 **wolverine** 之间配置 RIP 的单播，首先要用 **passive-interface** 命令来防止 RIP 广播信息进入令牌环网络。接着，加上一条 **neighbor** 声明来指向 RIP 更新信息要发送的路由器。例 9-8 列出了 **storm** 和 **wolverine** 之间的 RIP 配置。

例 9-8 RIP 的单播配置

```
hostname wolverine
!
router rip
  passive-interface TokenRing1
  network 128.200.0.0
  neighbor 128.200.10.2

hostname storm
!
router rip
  passive-interface TokenRing0
  network 128.200.0.0
  neighbor 128.200.10.1
```

下一步是在路由器 **rogue** 上的 EIGRP 和 RIP 之间进行双向重分布。网络中只有一个重分布点，因此进行双向重分布时就没有必要做路由过滤。路由器 **rogue** 上的 EIGRP 和 RIP 之间进行双向重分布要使用命令 **redistribution**，再加上默认的度量。为 RIP 选的默认度量值是 3，而 EIGRP 的是 10000 1000 254 1 1500。路由域（主网）是相互重叠的（也就意味着 RIP 广播也能在 **cyclops** 上接收到），因此可以用 **passive-interface** 命令防止出现过多不必要的广播。例 9-9 给出了路由器 **rogue** 上的 EIGRP 和 RIP 的配置情况。

例 9-9 路由器 **rogue** 上的 EIGRP 和 RIP 配置

```
router eigrp 2001
 redistribute rip
 passive-interface Ethernet0/0
 network 128.200.0.0
 default-metric 10000 1000 254 1 1500
 no auto-summary
!
router rip
 redistribute eigrp 2001
 passive-interface Ethernet0/1
 network 128.200.0.0
 default-metric 2
```

到目前为止，网络接近完整。但是，如果看一下路由器 **storm** 上的路由表，就会发现表中没有到 128.200.3.16/29 去的路由，如图 9-4 所示。

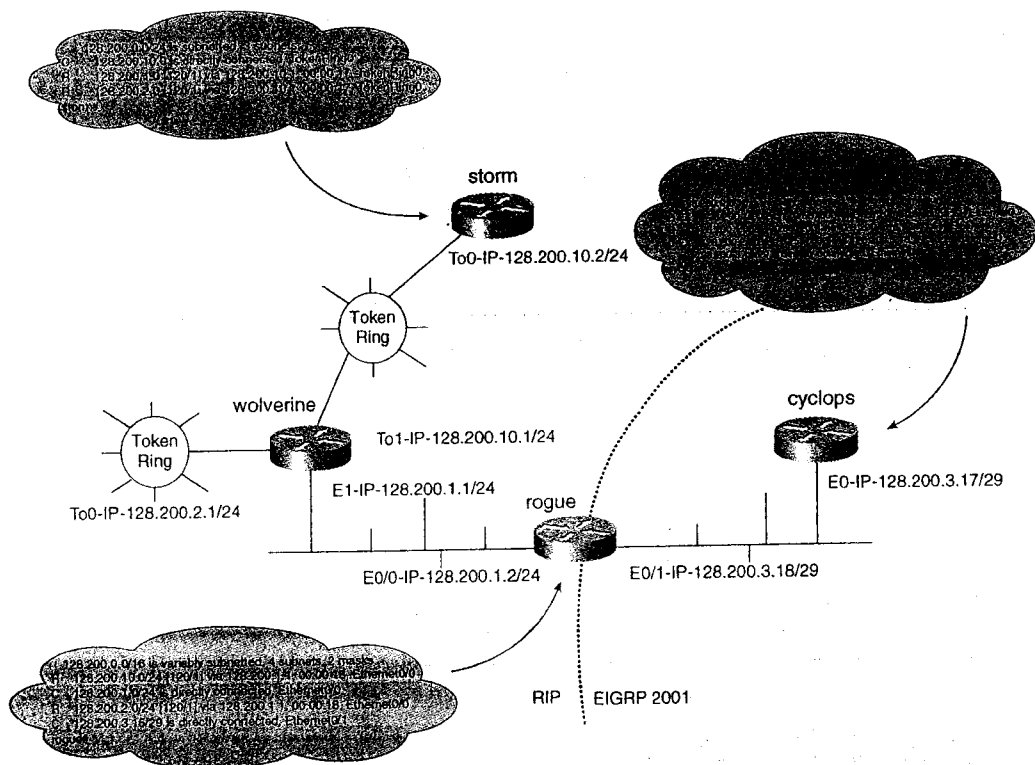


图 9-4 路由表差异：路由汇总之前的 RIP/EIGRP 网络

路由器 **rogue** 和 **cyclops** 之间的以太网段的子网掩码是 29 位的，而 **rogue** 和 **wolverine** 之间的网段则是 24 位的，因此 **rogue** 路由器不会从它的 E0/0 端口发送任何 29 位的网络更新信息。在 **rogue** 路由器上执行 **debug ip rip** 命令能够证实这一点，如例 9-10 所示。

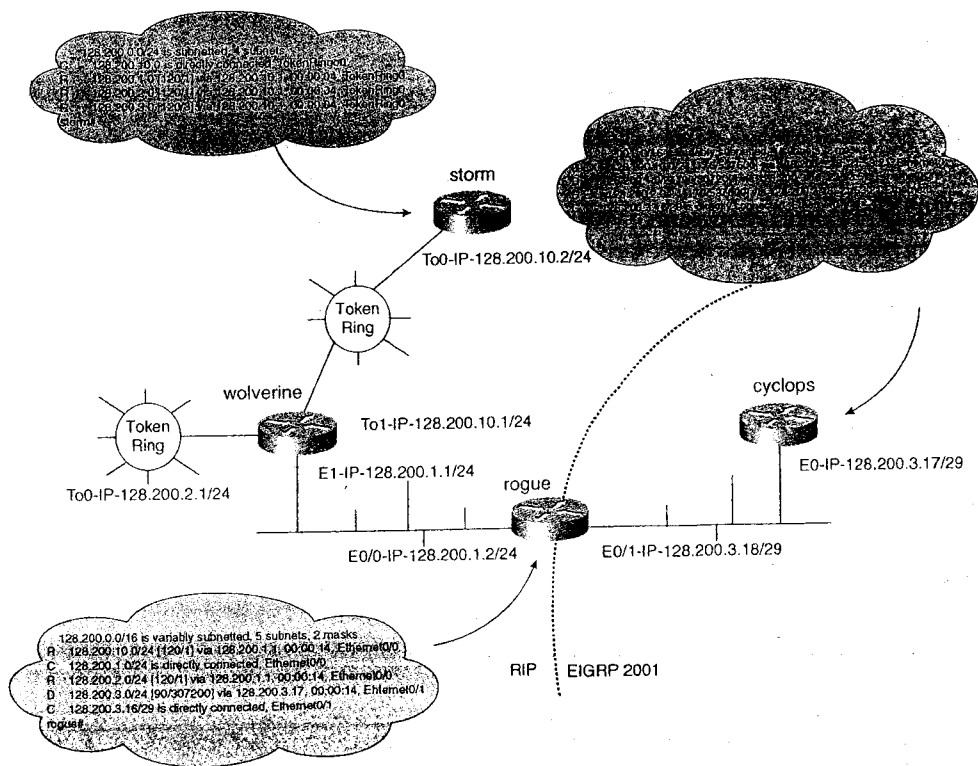
例 9-10 debug ip rip 命令的执行结果

```
rogue#debug ip rip
RIP protocol debugging is on
RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (128.200.1.2) -
suppressing null update      ←no 128.200.3.x route
RIP: received v1 update from 128.200.1.1 on Ethernet0/0
    128.200.10.0 in 1 hops
    128.200.2.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (128.200.1.2) -
suppressing null update
RIP: received v1 update from 128.200.1.1 on Ethernet0/0
    128.200.10.0 in 1 hops
    128.200.2.0 in 1 hops
rogue#
```

要在路由器 **cyclops** 和 **storm** 之间实现完全的相互可达性，之间的路由必须一致为 24 位掩码，可以通过在 **cyclops** 路由器上执行下面这条命令来实现：

```
cyclops (config-if) #ip summary-address eigrp 2001 128.200.3.0 255.255.255.0
```

观察一下 **rogue** 上路由表，现在有了一条 128.200.3.0/24 的 EIGRP 路由，如图 9-5 所示。由于该路由是 24 位掩码，**rogue** 可以通过其 E0/0 端口转发，**storm** 最后收到该路由信息。



警告 将路由选择协议重分布进另一个路由选择协议时要小心。如果网络中存在两个以上的重分布点，就可能出现路由环路。如果只有一个重分布点，路由选择协议固有的环路预防机制就足以避免环路的出现。在网络中有多个重分布点的情况下，距离矢量协议很容易出现环路，可以使用路由图和仔细斟酌 IP 地址分配来控制环路的出现。

技巧 使用子网掩码与路由更新信息不同时传输的路由选择协议(如 RIP-1 和 IGRP)时，一定要注意保持整个 Internet 网络中掩码位的一致性。例如，RIP 域可能工作在 24 位的网络上，而 OSPF 域则可能部分是 24 位的 LAN 网络，而整个 WAN 部分却使用 30 位掩码。RIP 域与 LAN 之间的数据传输没有问题，因为二者的掩码位匹配，但是，它不能与 WAN 接口进行数据的传输。要在两个域之间实现完全的互连，存在于 30 位网络掩码上的 OSPF WAN 网络在重分布进 RIP 之前必须要先汇总为 24 位的网络。

9.4 RIP 默认路由

连接 Internet 时，无论何时都需要默认路由。没有默认路由，路由器就需要在其路由表中有到每个网络的路径。默认路由配置成指向缺省网关的路由。路由器在其路由表中找不到适合某个数据包的路由时，就会将数据包转发到该网关。Cisco 路由器通常会进行有类路由查找，即除非使用全局命令 **ip classless** 设置，否则路由器不会将数据包转发到网关。在 Cisco IOS 11.3 以及更新版本的路由器上，默认允许 **ip classless** 命令。

默认路由的概念随路由选择协议的不同而不同，每个路由选择协议使用特定的方法来定义和宣告默认路由。

配置 RIP 默认路由时有两个步骤的工作：

第 1 步 定义或标记一个默认网络。要做到这一点，也有两个步骤：

首先，RIP 会将地址 0.0.0.0 视为一个默认路由。这样的默认路由是通过加入全 0 的静态路由实现的，可以用命令 **ip route 0.0.0.0 0.0.0.0 a.b.c.d** 来完成。建立了这个静态路由之后，就不必将其重分布进 RIP。RIP 会自动宣告这一路由。把网络标记为默认路由的另外一个办法就是用 **ip default-network a.b.c.d** 命令来实现。

第 2 步 确保路由器上已经设置 **ip classless**。没有 **ip classless**，路由器不会将数据转发到网关。

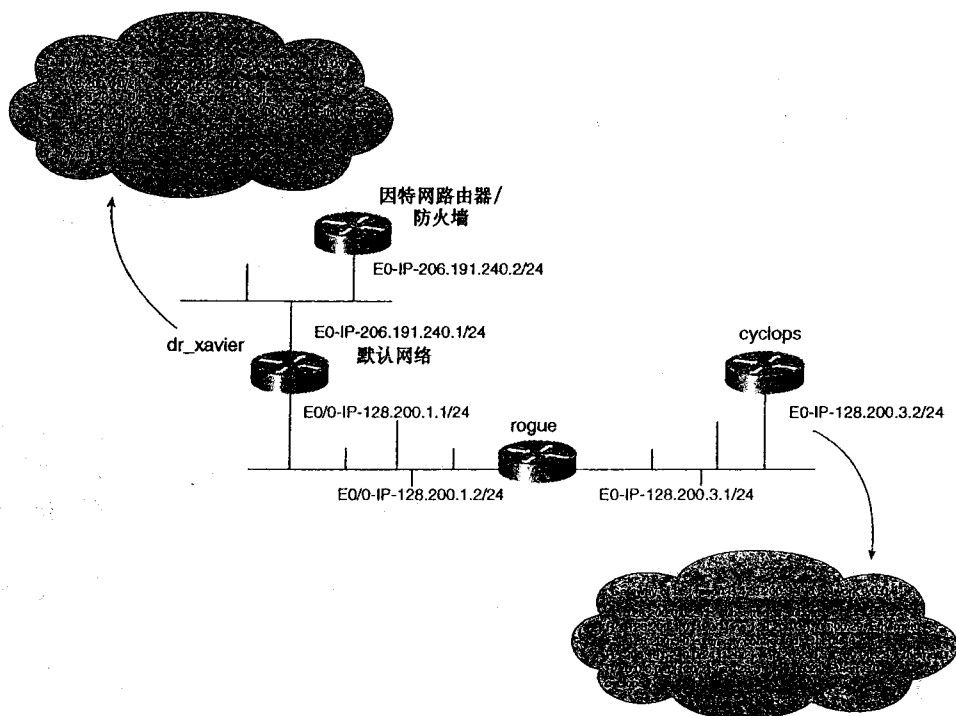
图 9-6 中加入了路由器 dr_xavier，它有一个通过 Internet_router/Firewall 到达 Internet 的默认路由。该例使用静态默认路由。在路由器 dr_xavier 上使用 **ip route 0.0.0.0 0.0.0.0 206.191.240.2** 命令，默认路由会被传播到 rogue 和 cyclops。路由器 dr_xavier 的路由表中将此路由作上“*”标记，即该路由为默认路由。这一路由传到下游路由器后，下游路由器就会成为新的缺省网关，如图 9-6 所示。

例 9-11 列出了 dr_xavier 路由器的配置情况。

例 9-11 dr_xavier 路由器配置的相关部分

```
!
router rip
```

```
network 128.200.0.0
network 206.191.240.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 206.191.240.2
!
```



9.5 实验 20：集成 RIP 网络：重分布、路由的过滤和控制——第 1 部分

9.5.1 实验说明

RIP 仍然是今天使用最广泛的路由选择协议之一。尽管 RIP 在一个大型网络中可能不占统治地位，但 RIP 还是散布于网络之中。如何集成散布的 RIP 以及如何弥补因不支持 VLSM 带来的缺陷就成为一种挑战。本实验主要练习如何在网络中集成 RIP 以及如何控制路由更新。

9.5.2 实验内容

络上的信息源自雪茄生产商，如 montercriso 和 romeo_y_julieta。Hanano 连接了世界范围的进口商，如 churchill_imports。我们的任务是将雪茄生产商处的 RIP 网络集成到出口商处的 EIGRP 网络。网络设计应参考下面要求：

- 路由器 habanos、montecristo 和 romeo_y_julieta 之间的帧中继网络使用 IP 子网 150.100.100.0/24。路由选择协议为 RIP。确保 montecristo 和 romeo_y_julieta 之间的 IP 连通性。
- 路由器 habanos 和 churchill_imports 之间的帧中继网络使用 IP 子网 150.100.200.0/30。路由选择协议为 EIGRP，自治系统号 2001。
- 在路由器 habanos 上将 RIP 和 EIGRP 相互重分布。确保整个网络的 IP 全连接性。
- 阻止路由器 habanos 上的子网 150.100.10.0/24 访问 montecristo 和 romeo_y_julieta。
- （可选）配置路由器 habanos，使来自 romeo_y_julieta 的路由管理距离为 5。

9.5.3 实验目的

- 按图 9-7 配置 Habano 网络及相关的 IP 地址，实验中的 LAN 拓扑类型对实验结果没有影响。
- WAN 的数据链路层协议为帧中继。
- 按图 9-7 配置 RIP 和 EIGRP。在 RIP 和 EIGRP 间进行重分布以提供整个网络的 IP 全连接。注意防止不必要的路由选择协议广播进入不运行该协议的网段。
- 阻止路由器 habanos 上的子网 150.100.10.0/24 访问 montecristo 和 romeo_y_julieta。
- （可选）调整来自 remeo_y_julieta 的路由管理距离。

9.5.4 所需设备

- 5 台路由器，通过 V.35 背对背线缆或类似方式连接。一台作为帧交换机的路由器，要求有 4 个串行接口。
- 利用集线器或交换机构成的 4 个 LAN 网段。

9.5.5 物理设计与实验准备

- 配置帧中继交换机，提供如图 9-7 所列的 PVC。如果需要关于配置帧中继交换机的帮助，请参考第 1 章：“建立网络互联模型的关键组件”。例 9-12 给出了帧中继交换机的配置示例。
- 按 9-7 所示，将集线器和串行线缆与路由器相连。

例 9-12 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
```

（待续）


```

interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148000
frame-relay intf-type dce
frame-relay route 121 interface Serial1 120
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay intf-type dce
frame-relay route 110 interface Serial5 111
frame-relay route 120 interface Serial0 121
frame-relay route 130 interface Serial3 131
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 131 interface Serial1 130
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 111 interface Serial1 110
!
    
```

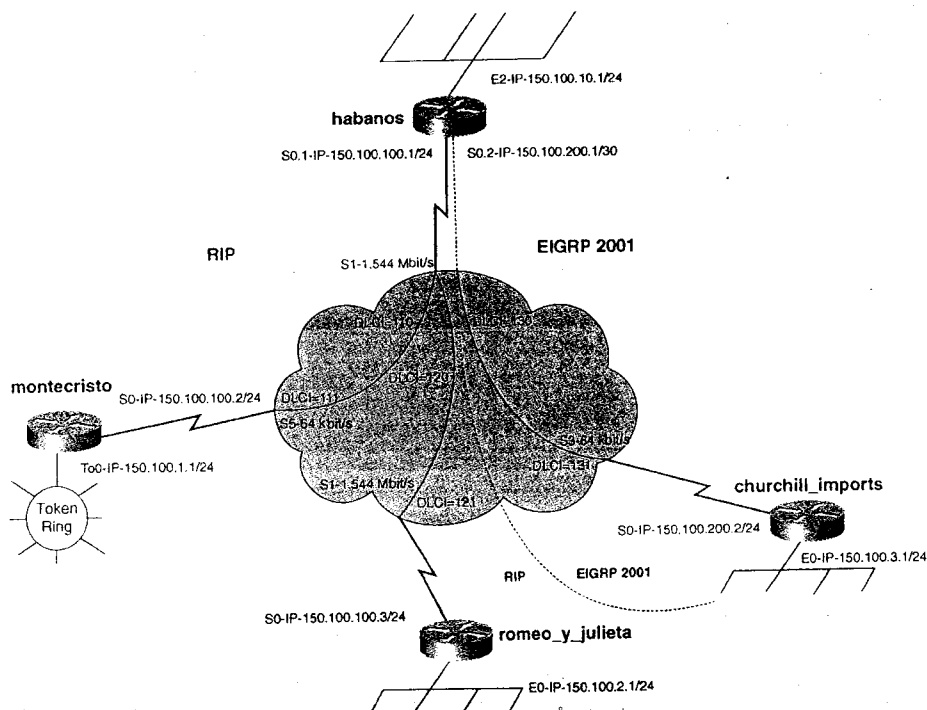


图 9-7 habano 网络

9.6 实验 20：集成 RIP 网络：重分布、路由的过滤和控制——第 2 部分

9.6.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将 4 台路由器以背对背的方式与帧中继交换机连接起来，再利用交换机或者是集线器/MAU 构成 4 个 LAN。

物理连接完成之后，按照图 9-7 为所有的 LAN 接口分配 IP 地址。在进行下一步的操作之前一定要用 **ping** 命令对路由器的本地 LAN 接口进行测试。

首先，对 RIP 域中的路由器进行配置。从 habanos 路由器上的多点子接口开始，habanos, montecristo 和 romeo_y_julieta 之间的网络是一个多点网络，因此可以利用 **frame-relay map** 命令进行配置。例 9-13 是这部分的帧中继配置示例。

例 9-13 帧中继配置的第 1 部分——RIP

```
! hostname habanos
!
interface Serial0.1 multipoint
 ip address 150.100.100.1 255.255.255.0
 no arp frame-relay
 frame-relay map ip 150.100.100.2 110 broadcast
 frame-relay map ip 150.100.100.3 120 broadcast
 no frame-relay inverse-arp

-----

hostname montecristo
!
 interface Serial0
 ip address 150.100.100.2 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 150.100.100.1 111 broadcast
 frame-relay map ip 150.100.100.3 111 broadcast
 frame-relay lmi-type cisco

-----

hostname romeo_y_julieta
!
 interface Serial0
 ip address 150.100.100.3 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 150.100.100.1 121 broadcast
 frame-relay map ip 150.100.100.2 121 broadcast
 frame-relay lmi-type cisco
```

现在应该可以 **ping** 通配置完毕的 WAN 接口。

配置 RIP 的时候，RIP 域中每台路由器上都需要用 **router rip** 命令和 **network 150.100.0.0** 命令进行设置。这里的帧中继网络是一个多点网络，因此，需要在路由器 habanos 上配置

平分割功能。如果没有屏蔽水平分割，montecristo的令牌环网络就不会向romeo_y_julieta路由器传送路由信息，同时也不会逆向传送路由信息。水平分割的屏蔽是在habanos路由器的Serial0.1接口上利用no ip split-horizon命令来实现的。

现在完成了RIP域的配置，开始对EIGRP域进行配置。第1步是habanos和churchill_imports之间的帧中继点对点网络的配置。例9-14是这一配置示例，这里使用了帧中继逆向ARP。

例 9-14 帧中继配置的第2部分——EIGRP

```
! hostname habanos
!
interface Serial0.2 point-to-point
 ip address 150.100.200.1 255.255.255.252
 frame-relay interface-dlci 130
!

!
hostname churchill_imports
!
interface Serial0
 ip address 150.100.200.2 255.255.255.252
 encapsulation frame-relay
 no fair-queue
 frame-relay interface-dlci 131
!
```

EIGRP的配置与RIP基本上一样，在habanos和churchill_imports上使用router eigrp 2001命令和network 150.100.0.0命令。如果这方面有什么问题，可以参考第11章“混合协议：增强型内部网关路由选择协议（EIGRP）”。

要获得完整的IP连接，还需要到habanos路由器上在RIP和EIGRP之间进行重分布，这是通过redistribute命令加上一个默认度量指标来实现的。RIP的默认度量指标是2，而EIGRP的默认度量指标则是10000 1000 254 1 1500。同时，这里还应用passive-interface命令来避免把RIP和EIGRP路由更新信息通过同一个接口发送。例9-15是habanos的配置示例。

例 9-15 配置 habanos 上的路由选择协议

```
! hostname habanos
!
router eigrp 2001
 redistribute rip          ←Redistributing RIP
 passive-interface Ethernet2 ←Prevents EIGRP broadcasts
 passive-interface Serial0.1
 network 150.100.0.0
 default-metric 10000 1000 254 1 1500
!

router rip
 redistribute eigrp 2001   ←Redistributing EIGRP
 passive-interface Serial0.2 ←Prevents RIP broadcasts
 network 150.100.0.0
 default-metric 2
!
```

现在 churchill_imports 路由器上已经有了一份完整的路由表，在里面看到两条 EIGRP 路由，150.100.100.0/24 和 150.100.10.0/24，这是由于 EIGRP 的路由距离比 RIP 的要低。此外还可以看到两条外部路由，150.100.2.0/24 和 150.100.1.0/24，它们是从 RIP 重分布到 EIGRP 中的。例 9-16 给出了更新之后 churchill_imports 路由器上的路由表的情况。

例 9-16 churchill_imports 上的路由表

```
churchill_imports#show ip route
<<<text omitted>>>
  150.100.0.0/16 is variably subnetted, 6 subnets, 2 masks
C       150.100.200.0/30 is directly connected, Serial0
D       150.100.100.0/24 [90/2681856] via 150.100.200.1, 00:29:52, Serial0
D EX    150.100.2.0/24 [170/2425856] via 150.100.200.1, 00:18:01, Serial0
C       150.100.3.0/24 is directly connected, Ethernet0
D EX    150.100.1.0/24 [170/2425856] via 150.100.200.1, 00:18:01, Serial0
D       150.100.10.0/24 [90/2195456] via 150.100.200.1, 00:29:52, Serial0
churchill_imports#
```

现在配置过程基本完成，但是如果观察一下 montecristo 上的路由表，如例 9-17 所示，会发现表中缺少了路由 150.100.200.0/30。

例 9-17 montecristo 的路由表

```
montecristo#show ip route
<<<text omitted>>>
  150.100.0.0/24 is subnetted, 5 subnets
C       150.100.100.0 is directly connected, Serial0
R       150.100.2.0 [120/2] via 150.100.100.1, 00:00:05, Serial0
R       150.100.3.0 [120/2] via 150.100.100.1, 00:00:05, Serial0
C       150.100.1.0 is directly connected, TokenRing0
R       150.100.10.0 [120/1] via 150.100.100.1, 00:00:05, Serial0
montecristo#
```

路由 150.100.200.0/30 丢失的原因是该路由的网络掩码位是 30 位，RIP 没有办法通过 habanos 路由器的 Serial0.1 端口发送路由更新信息。要解决这一问题，EIGRP 必须把关于 habanos 和 churchill_imports 之间的帧中继链路上的路由汇总到 24 位长的掩码。该路由的汇总总是通过在 churchill_imports 路由器的 Serial 0 接口上配置 **ip summary-address eigrp 2001 150.100.200.0 255.255.255.0** 命令来实现。这样修改之后，再查看 montecristo 的路由表，就能看到这条汇总路由 150.100.200.0/24，如例 9-18 所示。

例 9-18 montecristo 上的完整路由表

```
montecristo#show ip route
<<<text omitted>>>
  150.100.0.0/24 is subnetted, 6 subnets
R       150.100.200.0 [120/2] via 150.100.100.1, 00:00:02, Serial0
C       150.100.100.0 is directly connected, Serial0
R       150.100.2.0 [120/2] via 150.100.100.1, 00:00:02, Serial0
R       150.100.3.0 [120/2] via 150.100.100.1, 00:00:02, Serial0
C       150.100.1.0 is directly connected, TokenRing0
R       150.100.10.0 [120/1] via 150.100.100.1, 00:00:02, Serial0
montecristo#
```

技巧 在处理较为复杂的网络时，网络的收敛以及路由的传播都会花费一定的时间。可以利用 **clear ip route *** 命令强制加快路由更新的过程，这条命令可以清除所有的路由，从而强制性地地进行路由更新。

完全建立起了 IP 连接后，就可以进行路由过滤了。在这个实验模型中，要阻止路由 150.100.10.0/24 传送到路由器 montecristo 和 romeo_y_julieta。可以通过在 habanos 路由器的 RIP 配置部分加上一条 **distribute-list out** 命令来实现。当然，也需要定义访问控制列表，以帮助过滤掉网络 150.100.10.0。例 9-19 是最后的配置部分。

例 9-19 完整的路由配置

```
hostname habanos
!
interface Ethernet2
 ip address 150.100.10.1 255.255.255.0
 media-type 10BaseT
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 multipoint
 ip address 150.100.100.1 255.255.255.0
 no ip split-horizon
 no arp frame-relay
 frame-relay map ip 150.100.100.2 110 broadcast
 frame-relay map ip 150.100.100.3 120 broadcast
 no frame-relay inverse-arp
!
interface Serial0.2 point-to-point
 ip address 150.100.200.1 255.255.255.252
 frame-relay interface-dlci 130
!
router eigrp 2001
 redistribute rip
 passive-interface Ethernet2
 passive-interface Serial0.1
 network 150.100.0.0
 default-metric 10000 1000 254 1 1500
!
router rip
 redistribute eigrp 2001
 passive-interface Serial0.2
 network 150.100.0.0
 default-metric 2
 distribute-list 10 out Serial0.1
!
ip classless
!
access-list 10 deny 150.100.10.0 0.0.0.255
access-list 10 permit any

hostname montecristo
!
ip subnet-zero
!
interface Serial0
```

```

ip address 150.100.100.2 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay map ip 150.100.100.1 111 broadcast
frame-relay map ip 150.100.100.3 111 broadcast
frame-relay lmi-type cisco
!
interface TokenRing0
ip address 150.100.1.1 255.255.255.0
no ip directed-broadcast
ring-speed 16
!
router rip
network 150.100.0.0
!
ip classless

```

```

hostname romeo_y_julieta

```

```

!
ip subnet-zero
!
interface Ethernet0
ip address 150.100.2.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0
ip address 150.100.100.3 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay map ip 150.100.100.1 121 broadcast
frame-relay map ip 150.100.100.2 121 broadcast
frame-relay lmi-type cisco
!
router rip
network 150.100.0.0

```

```

hostname churchill imports

```

```

!
interface Ethernet0
ip address 150.100.3.1 255.255.255.0
!
interface Serial0
ip address 150.100.200.2 255.255.255.252
ip summary-address eigrp 2001 150.100.200.0 255.255.255.0
encapsulation frame-relay
no fair-queue
frame-relay interface-dlci 131
!
router eigrp 2001
network 150.100.0.0
!

```

最后，本实验的可选部分还要求把所有来自 romeo_y_julieta 路由器的路由的管理距离设置为 5。在 RIP 中利用 **distance** 命令就可以达到这一目的。实验中并不是要把所有的管理距离都设为 5，而只是要对来自某个邻居路由器的路由管理距离进行设置。因此，在 **distance** 命令中需要调用访问控制列表以实现该目的，如例 9-20 所示。

例 9-20 设置某路由条目的管理距离

```

hostname habanos
!
router rip
 redistribute eigrp 2001
 passive-interface Serial0.2
 network 150.100.0.0
 default-metric 2
 distribute-list 10 out Serial0.1
 distance 5 150.100.100.3 0.0.0.11  -- Set distance for routes in list 11 only
!
ip classless
!
access-list 10 deny 150.100.10.0 0.0.0.255
access-list 10 permit any
access-list 11 permit 150.100.2.0 0.0.0.255  -- Allow only 150.200.2.0 through
!

```

现在看一下 habanos 路由器上路由表的情况，如例 9-21 所示，尽管 RIP 的默认路由管理距离是 120，路由 150.100.2.0 的管理距离却是 5。

例 9-21 设置某路由条目的管理距离

```

habanos#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    150.100.0.0/16 is variably subnetted, 7 subnets, 2 masks
D       150.100.200.0/24 [90/2681856] via 150.100.200.2, 00:04:42, Serial0.2
C       150.100.200.0/30 is directly connected, Serial0.2
C       150.100.100.0/24 is directly connected, Serial0.1
R       150.100.2.0/24 [5/1] via 150.100.100.3, 00:00:20, Serial0.1
D       150.100.3.0/24 [90/2195456] via 150.100.200.2, 00:04:42, Serial0.2
R       150.100.1.0/24 [120/1] via 150.100.100.2, 00:00:07, Serial0.1
C       150.100.10.0/24 is directly connected, Ethernet2
habanos#

```

第 10 章

距离矢量协议： 内部网关路由选择 协议（IGRP）

思科在 1986 年左右推出内部网关路由选择协议(IGRP)，那时，网络管理员们对 RIP 协议的缺陷还没有什么解决办法。RIP 的 15 跳限制和过于简单的度量使网络不易扩展，无法在非等价路由开销的路径间分担数据传输。OSPF 还无法在未来的两年内定稿，这样就需要新的路由选择协议来代替现有的路由选择协议。Cisco 以网络先驱的身份推出了 IGRP，以专门解决 RIP 的缺陷。

IGRP 与其他距离矢量协议不同的特性：

- 可扩展性——最大限制值为 255 的跳计数比 RIP 的 15 上限为网络提供了更为广阔的范围。IGRP 的跳数限制默认值为 100。
- 更快收敛性——IGRP 采用 *立即更新* (Flash updates) 将网络拓扑结构改变时的更新信息发送到邻居路由器。
- 完善的度量——IGRP 使用以 5 个单独的度量（带宽、延时、可靠性、负载和最大传输单元 MTU）为基础的复合度量来决定路由的选择。
- 非等价路由开销的负载平衡——IGRP 复合路由度量使得它可以在非等价的路径进行负载平衡。

这些特性大大增强了上个世纪 80 年代中期的路由选择协议的性能。但是，同 RIP 一样，IGRP 最终也让位给了它的同胞——增强型 IGRP (EIGRP)。本章要讲述 IGRP 的性能和工作方式以及相关的配置和重分布设置。

10.1 IGRP 技术概览

IGRP 是一个有类距离矢量协议，IP 协议类型是 9，它通过直连接口交换路由信息。IGRP 采用自治域的概念，周期性地把整个路由表以广播的形式传送到所有的邻居路由器。而且还使用了一些计时器和度量来控制路由的有效性。IGRP 工作机制如下：

- **路由更新**——初始化之后每 90 秒，IGRP 将路由更新信息通过广播的形式从所有的 IGRP 接口发送出去。这些更新信息包括所有的路由以及路由的类型和度量，包括除了被水平分隔和过滤器抑制的路由之外路由表中全部的路由。IGRP 还使用立即更新 (Flash update) 方式。只要网络拓扑结构发生改变，IGRP 就立即发送更新路由信息到所有的邻居路由器，这一更新信息包括了全部的路由表。
- **计时器**——90 秒的路由更新时间过去之后，**更新计时器**会触发另一次的路由更新信息的发送。当路由为邻居路由器接收后，本地路由器设置**失效计时器**，是更新计时器的 3 倍，也就是 270 秒，同时还要设置一个**刷新计时器**，其默认值是更新计时器的 7 倍，即 630 秒。这些计时器的定义如下：
 - **更新计时器**——该计时器是指定路由器每次发送路由信息间隔的等待时间。同时，在所有的路由更新信息之间都存在一个随机的延时抖动时间偏差，这是为了防止路由信息之间的同步问题。
 - **失效计时器**——这个计时器是指定路由器在接收到另一次更新信息之前继续宣告路由信息的时间间隔。如果在接收到另一次路由更新信息之前，该计时器就计数溢出，那么这个路由就会被标记为不可到达。
 - **刷新计时器**——这个计时器指定了路由器将一个路由标记为不可到达之后直到将该路由从路由表中删去之前的时间间隔。
 - **抑制计时器**——如果路由成为不可到达，或者是下一跳路由器增加了某路由的度量，该路由就置于抑制状态。抑制计时器的值是更新计时器的 3 倍再加上 10 秒，默认情况下是 280 秒。该计时器还能使路由器在收敛阶段接收不到新的信息。

这些计时器的默认值可以用 `timers basic update invalid holddown flush [sleeptime]` 命令加以修改。参数 `sleeptime` 让路由器在接收了一个触发的更新信息，等待 `sleeptime` 的时间再去发送路由更新信息。

IGRP 采用了水平分隔和逆向毒化的概念来防止路由环路的出现。二者的描述如下：

- **水平分隔**——IGRP 用水平分隔来避免出现路由环路。前面章节讲过，水平分隔就是某路由的信息不通过接收到该信息的接口或子接口发送出去。默认情况下，所有的接口上都允许水平分隔。要禁止该功能，可以使用 `no ip split-horizon interface` 命令。
- **逆向毒化**——路由器发送逆向毒化更新信息来删除路由并将其置于抑制状态，这也是为了防止路由环路的出现。路由器检测到 1.1 倍或更大的度量增加时就发送逆向毒化信息。IGRP 通过将复合度量设置为 4,294,967,295 并将路由回送到该路由的源端来实现这一功能。

10.1.1 IGRP 的路由类型

可以看到 IGRP 的计时器工作方式和 RIP 很相似。这两个协议的不同之处是路由的发送方式。有如下 3 种主要的 IGRP 路由：

- 外部路由——外部路由用 `ip default-network a.b.c.d` 命令来设置。如果路由器上执行了 `ip classless` 命令，而 IGRP 也配置了默认网络，路由器会将没有路由指向的数据包转发到默认网络。
- 内部或子网路由——如果某路由属于发送该路由的接口所在主类子网的一部分，则该路由就是做为内部路由来发送的。
- 系统或网络路由——如果某路由不属于发送该路由的接口所在主类子网的一部分，则该路由就是一个系统路由。系统路由总是在主网边界上作为汇总路由发送。

图 10-1 解释了路由器发送这 3 种类型的路由的情况。这个例子中，突出显示的路由器在其配置中使用了 `ip default-network 206.191.241.0` 命令，将 206.191.241.0 标记为外部路由发送出去。

例子中的 `igrp_rtr` 在同一自治域系统中有 4 个接口。Ethernet 5 (E5) 接口是在子网 172.16.1.0/24 中。由于发送路由的接口和路由本身是处在同一主网边界之中的，因此，IGRP 将路由 172.16.2.0 作为一个子网路由从 E5 接口发送出去。同时 IGRP 把路由 172.18.0.0 作为 172.18.1.0/24 的汇总路由从 E5 接口发送出去，这是因为发送路由的接口和路由本身不在同一主网边界之中，最后，对于路由 206.191.241.0/29，IGRP 将它作为 206.191.241.0 的外部路由来发送。接收 IGRP 的路由器视此路由为其默认路由，或者是叫做最后手段的网关。

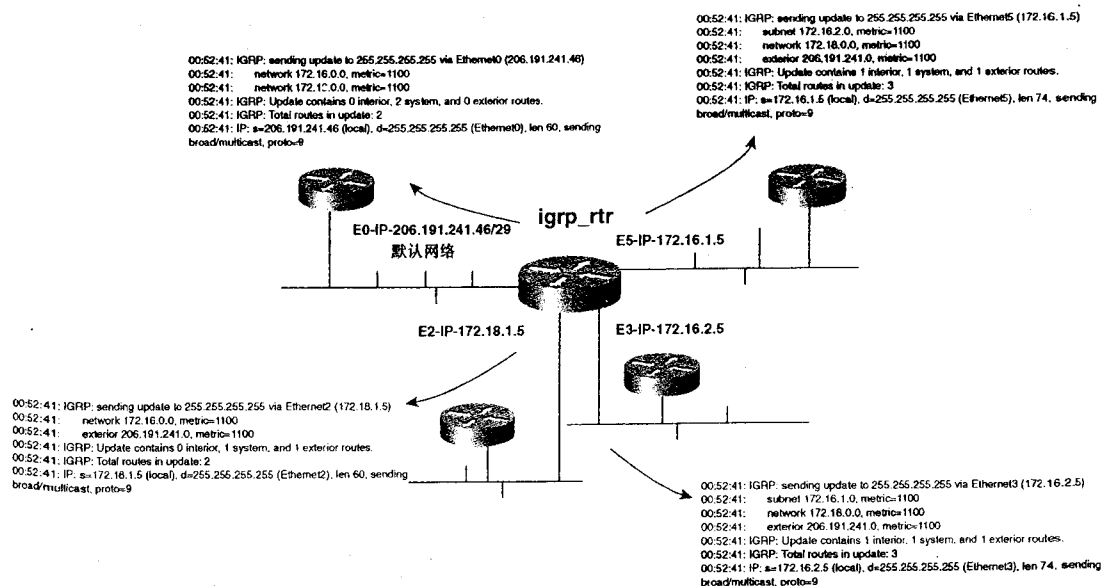


图 10-1 IGRP 的路由更新情况

10.1.2 IGRP 的度量

与 RIP 相比，IGRP 的又一增强之处是采用了复杂的组合度量。带宽、延时、负载、可靠性和 MTU 都是其度量的因素。默认情况下，IGRP 主要根据带宽和延时来选择路由。第 11 章“混合协议：增强型内部网关路由选择协议（EIGRP）”中，将介绍怎样通过设置 *k* 值来产生和调整组合度量。下面详细讨论这 5 个 IGRP 的度量：

- **带宽**——带宽用 kbit/s 表示。对于带宽，最好手工设置以准确表达运行 IGRP 的接口属性。例如，带宽 56kbit/s 的接口和 T1 接口的默认带宽都是 1544kbit/s。可以用接口命令 **bandwidth kbit/s** 准确地更改带宽值。表 10-1 是一些常见带宽的默认设置情况。
- **延时**——延时的单位是毫秒（ms），也必须手工设置延时的值，以准确地表示某个运行着 IGRP 的接口属性。某接口的延时可以用接口命令 **delay time_in_microseconds** 来进行调整。
- **可靠性**——可靠性是一个动态的数字，其值从 1 到 255。255 表示完全可靠的链路，而 1 则表示不可靠链路。
- **负载**——负载是一个 1 到 255 的数，代表接口的输出负载。负载值的是动态的，可以通过 **show interfaces** 命令来查看。这里的 1 代表负载最小的链路，而 255 则代表满负载链路。
- **MTU**——路径中最小的最大传输单元值。

注释 无论是在 IGRP 还是 EIGRP 中，如果想要改变对路由的取向，最好是使用延时而不要使用带宽度量。改变带宽会对其他路由选择协议（如 OSPF）造成影响。延时的改变只会影响 IGRP 和 EIGRP。

表 10-1 是一些常用度量的情况。

表 10-1 常用的 IGRP 和 EIGRP 度量

介 质	带 宽	延 时
100M ATM	100000 kbit/s	100 微秒
快速以太网	100000 kbit/s	100 微秒
FDDI	100000 kbit/s	100 微秒
HSSI	45045 kbit/s	20000 微秒
16M Token Ring	16000 kbit/s	630 微秒
10M 以太网	10000 kbit/s	1000 微秒
T1	1544 kbit/s	20000 微秒
DS-0	64 kbit/s	20000 微秒
56K	56 kbit/s	20000 微秒

* 这些指标不是 IGRP 更新信息中实际传输的度量值，而是用来产生实际传输的度量值的参数。

10.2 IGRP 的配置

基本的 IGRP 配置包括两个步骤。配置 IGRP 需要定义一个自治域系统。自治域系统是在同一授权技术管理之下的一组路由器的集合。IGRP、EIGRP 和 BGP 都使用自治域系统的概念，但真正在路由选择过程中使用了自治域系统的路由协议只有 IGRP。配置 IGRP 或 EIGRP 时，不需注册自治域系统。

AS)。按定义，和 BGP 采用自治域系统的概念，但真正在路由选择过程中使用了自治域系统的路由协议只有 IGRP。

下面这两个步骤再加上可选的第 3 步能够针对特定的环境对 IGRP 进行细致的调

第 1 步 在路由器上激活 IGRP 运行并定义自治域系统。通过全局配置模式下的 `router igrp autonomous_system_id` 来完成。

细致的调
router igrp

第 2 步 加入运行 IGRP 协议的网络。在 `config-router#` 提示符模式下使用 `network` 命令完成。输入 `network` 声明之后接着需要再输入的只是网络地址和掩码。

network 172.16.0.0
network 172.18.0.0
network 206.191.241.0

第 3 步 可选：使用 `bandwidth` 命令对 IGRP 的度量进行微调，或者使用 `ip classless` 命令来配置无类路由。其他一些可选项。

边界的路由器。配置 IGRP 时

例 10-1 是对图 10-1 的 IGRP 网络进行配置的情况。

例 10-1 配置 IGRP

```
1 hostname igrp_rtr
1
router igrp 2001          ← IGRP routing process
network 172.16.0.0       ← Networks running IGRP
network 172.18.0.0
network 206.191.241.0
!
ip classless
ip default-network 206.191.241.0 ← Default Network
!
```

配置其他 IGRP 选项之前，请先看一下 IGRP 的“Big D”和“Big S”命令。

10.2.1 IGRP 的“Big show”和“Big D”

IGRP 的排错过程和 RIP 很相似。IGRP 的配置错误大多是 `network` 命令使用不恰当造成的。子网不连续或者 IGRP 域的位掩码不连续。下面是一些非常有用的 IGRP 命令：

使用不恰当造成
和 debug 命令：

- `show ip protocols [summary]`
- `Router#show ip route`
- `Router (config-router) #debug ip igrp [transactions | events]`

下面详细讲述这些命令的用法。

1. show ip protocols [summary] 命令

这条命令可以显示所有路由选择协议的信息，详细的计时器和度量值，以及更新

信息。例 10-2 是这条命令的执行结果的情况。

例 10-2 show ip protocols 命令的执行结果

```
igrp_rtr#show ip protocols
Routing Protocol is "igrp 2001"
  Sending updates every 90 seconds, next update in 19 seconds, hold time 180
  Invalid after 270 seconds, hold down 280, flushed after 630
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0, K6=0
  IGRP maximum hopcount 100
  IGRP maximum metric variance 1
  Redistributing: igrp 2001
  Routing for Networks:
    172.16.0.0
    172.18.0.0
    206.191.241.0
  Routing Information Sources:
    Gateway      Distance      Last Update
    172.18.1.55   100           00:00:29
    206.191.241.42 100           00:00:06
    172.18.1.7    100           00:01:06
    172.16.2.4    100           00:38:38
    172.16.1.1    100           00:50:01
  Distance: (default is 100)
igrp_rtr#
```

2. show ip route 命令

该命令可以给出路由器当前的路由表信息以及用于做出数据转发决定的路由表情况。执行的结果还会显示某个路由来自什么路由选择协议（对本章而言，I 代表 IGRP 协议）。路由后面的数字是该路由的管理距离，后面跟着 IGRP 的混合度量值。字段 via 的含义是路由来自何处，接收到最近路由更新的时间以及是通过哪一个接口接收到的。例 10-3 给出了 **show ip route** 命令的执行结果。例子中的 206.191.241.0/24 是默认路由，用*标记。发送此路由的邻居路由器是 172.18.1.5。该路由上一次更新是在 52 秒前通过以太网0进行的。

例 10-3 show ip route 命令的输出结果

```
r7#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.18.1.5 to network 206.191.241.0

I*  206.191.241.0/24 [100/1200] via 172.18.1.5, 00:00:52, Ethernet0
I   172.16.0.0/16 [100/1200] via 172.18.1.5, 00:00:53, Ethernet0
```

(待续)

```

172.18.0.0/24 is subnetted, 3 subnets
I    172.18.18.0 [100/1600] via 172.18.1.55, 00:00:29, Ethernet0
C    172.18.19.0 is directly connected, Loopback20
C    172.18.1.0 is directly connected, Ethernet0
r7#

```

3. Router (config-router) #debug ip igrp [transactions | events] 命令

命令 **debug ip igrp transactions** 能够显示正在通过不同接口接收和发送的路由更新的详细信息。命令 **debug ip igrp transactions** 能提供综合的路由细节信息，列出每个网络的路由情况和组合度量。这两条命令都能显示哪个接口正在接收和发送路由。而 **debug ip igrp events** 命令只能给出正在接收和发送的路由类型。例 10-4 是 **debug ip igrp transactions** 命令的执行结果，例 10-5 则是 **debug ip igrp events** 命令的例子。

这些命令都是在图 10-1 中的路由器 **igrp_router** 上执行的。

例 10-4 debug ip igrp transactions 命令的结果

```

01:40:07: IGRP: received update from 206.191.241.42 on Ethernet0
01:40:07:     network 172.16.0.0, metric 1121211 (neighbor 1121111)
01:40:27: IGRP: sending update to 255.255.255.255 via Ethernet0 (206.191.241.46)
01:40:27:     network 172.16.0.0, metric=1100
01:40:27:     network 172.18.0.0, metric=1100
01:40:27: IGRP: sending update to 255.255.255.255 via Ethernet1 (172.18.1.5)
01:40:27:     network 172.16.0.0, metric=1100
01:40:27:     exterior 206.191.241.0, metric=1100
01:40:27: IGRP: sending update to 255.255.255.255 via Ethernet3 (172.16.2.5)
01:40:27:     subnet 172.16.1.0, metric=1100
01:40:27:     network 172.18.0.0, metric=1100
01:40:27:     exterior 206.191.241.0, metric=1100
01:40:27: IGRP: sending update to 255.255.255.255 via Ethernet5 (172.16.1.5)
01:40:27:     subnet 172.16.2.0, metric=1100
01:40:27:     network 172.18.0.0, metric=1100
01:40:27:     exterior 206.191.241.0, metric=1100
igrp_rtr#

```

例 10-5 debug ip igrp events 命令的结果

```

02:52:53: IGRP: sending update to 255.255.255.255 via Ethernet0 (206.191.241.46)
02:52:53: IGRP: Update contains 0 interior, 2 system, and 0 exterior routes.
02:52:53: IGRP: Total routes in update: 2
02:52:53: IGRP: sending update to 255.255.255.255 via Ethernet1 (172.18.1.5)
02:52:53: IGRP: Update contains 0 interior, 1 system, and 1 exterior routes.
02:52:53: IGRP: Total routes in update: 2
02:52:53: IGRP: sending update to 255.255.255.255 via Ethernet3 (172.16.2.5)
02:52:53: IGRP: Update contains 1 interior, 1 system, and 1 exterior routes.
02:52:53: IGRP: Total routes in update: 3
02:52:53: IGRP: sending update to 255.255.255.255 via Ethernet5 (172.16.1.5)
02:52:53: IGRP: Update contains 1 interior, 1 system, and 1 exterior routes.
02:52:53: IGRP: Total routes in update: 3
02:52:55: IGRP: received update from 172.18.1.7 on Ethernet1
02:52:55: IGRP: Update contains 1 interior, 0 system, and 0 exterior routes.
02:52:55: IGRP: Total routes in update: 1

```

10.3 IGRP 更新信息的调整、重分布和控制

和 RIP 一样，IGRP 可以调整计时器、广播控制、配置负载共享以及路由控制。下面详细介绍一下这些命令：

- **Router (config-router) #timers basic update invalid holddown flush [sleep-time]**——这条命令使得用户可以对 IGRP 的更新、失效、抑制、刷新以及可选的睡眠计时器进行设置。

要设置 IGRP 的单播更新，可以用 **neighbor** 命令定义下一跳的邻居路由器。该命令的功能和 RIP 完全一样。可以再参考第 9 章“距离矢量协议：路由信息协议版本 1 和版本 2 (RIP-1 和 RIP-2)”，回顾一下关于配置单播路由更新的内容。

- **Router (config-router) #passive-interface interface_name**——这条命令能够禁止路由更新信息通过某个接口发送。但路由器还是监听并接收该接口上的广播或者是单播更新信息。
- **Router (config-router) #neighbor a.b.c.d**——这条命令能够定义 IGRP 邻居路由器与本路由器进行单播更新信息的交换。该命令应该和 **passive-interface** 命令配合使用。
- **Router (config-router) #offset-list {access_list 0-99 [in | out] offset {metric_offset_1-214748364} [interface]}**——这条命令可以用来增加路由度量的值，增加的值不能超过 214748364。

可以使用分布列表 (distribute list) 对 IGRP 的路由更新信息进行过滤。回想一下，分布列表可以调用标准或者扩展的访问控制列表对路由更新信息作相应的过滤处理。将一个协议重分布进另一协议时，需要使用 **redistribute** 命令和默认度量值。重分布过程中要用路由图 (route map) 取代分布列表来实现对指定路由的控制。在 IGRP 的集成一节中，将看到 **redistribute** 命令和 **default-metric** 命令的例子。

- **Router (config-router) #distribute-list {1-199} [in | out] [interface]**——这条命令可以用来调用标准的或者扩展的访问控制列表，对输入或输出的路由更新信息进行过滤。
- **Router (config-router) #redistribute {connected | static | bgp | rip | eigrp | ospf | isis} [metric metric-value] [route-map map-tag]**——这条命令用来将其他路由选择协议重分布进 IGRP。还可以加上路由图对路由进行额外的控制，也可以为被重分布协议的路由提供一个不同于默认度量的路由度量值。重分布路由时要记住 IP 需要一个可以在源地址和目的地址之间往返通信的路由，多数时候这是通过相互重分布来实现的。
- **Router (config-router) #default-metric bandwidth_kbit/s 1-4214748364 delay_microseconds 1-4214748364 reliability 1-255 load 1-244 mtu 1-4214748364**——这条命令

用来设置重分布进 IGRP 的所有路由的默认度量。要记住的是，前面的那条 **redistribute** 命令中如果定义了某些度量，那么这条命令定义的参数就会被覆盖。通过使用 **default-metric** 命令或者通过在 **redistribute** 命令中指定度量来在协议之间进行重分布时，必须提供一个默认的度量，通常是使用 **default-metric 10000 1000 254 1 1500** 命令来指定，它让路由器采用一个

10000 的带宽值，一个 1000 的延时值，一个 1（没有负载）的负载值和一个 1500 的 MTU 值来产生组合的度量值。这条的链路可靠性是 254（255 为完全可靠）。

下面的命令集用于改变路由选择。单独的度量就像 IGRP 的管理距离一样可以修改。在改变某链路的度量时，最好使用 **delay** 命令改变延时值而不要用 **bandwidth** 命令改变带宽值。虽然二者都可以使用，但是 **bandwidth** 还会对 OSPF 造成影响，而 **delay** 只与 IGRP 和 EIGRP 有关。

- **Router (config-router) #metric weights 0 k1 k2 k3 k4 k5**——该命令让用户可以通过带宽、负载、延时可靠性等因素来设置 IGRP 度量的权重值。
- **Router (config-router) #distance weight_1-255 [adjacent_neighbors_ip_address wildcard_mask [access_list_0-99]]**——这条命令可以改变从某个邻居路由器接收路由的管理距离值。如果没有使用命令中的 IP 地址和掩码参数，那么协议下所有的路由都会设置成这里指定的距离值。参考第 9 章和该章的实验可以再次看到这条 **distance** 命令的应用和练习实例。

- **Router (config-if) #delay microseconds_1-4214748364**——这条命令用来指定某个接口的延时（单位是 ms）。该命令只用于路由选择协议，不影响链路上的数据传输。
- **Router (config-if) #bandwidth bandwidth_kbit/s_1-4214748364**——这条命令指定某个链路带宽（单位是 kbit/s），只用于路由选择协议，不影响链路上数据的传输。

10.3.1 非等价路由开销的负载平衡

IGRP 具有非等价路由开销的负载平衡功能。路由器在选择最大度量值的可选择路径时采用了一个 **variance** 值做为乘数。

配置非等价路由开销的负载平衡可以分为下面 3 步：

第 1 步 在与负载共享有关的接口上配置链路两端的带宽值，可以通过 **bandwidth kbit/s** 命令完成。

第 2 步 确定最低路由开销的度量和最高路由开销的度量。根据这些值，计算出这个 **variance** 乘数并将它加入 IGRP 的路由进程。

第 3 步 可选：设置最多路径或者是数据共享变量。

下面的例子计算假想的 **variance** 乘数。假设 IGRP 路由的度量值是 100。路由器有其他指向同一目的地址的路由，度量值是 200 和 300。要让这 3 个路径共享 IGRP 的数据，这个 **variance** 乘数就应该设为 3。（ 3×100 ）=300，或者是（最佳度量） \times （**variance** 乘数）=共享数据的路径最大度量。用下面的公式可以正确的设置这个 **variance** 乘数的值：

variance 乘数 = $1 + \{[(\text{最高路由开销的度量值}) / (\text{最低路由开销的路由度量值})]$
舍去小数加 1 进位}

最低路由开销的路由度量值可以用 **debug ip igrp transactions** 命令来查询得到。记住改变这个 **variance** 乘数和其他变量的值时要在链路两端同时进行。带宽也应该在所有的串行链路上设置。应用下面的命令来设置负载平衡：

- **Router (config-router) #variance metric_multiplier 1-128**——定义要用在非等价路由开销负载平衡过程中的度量乘数值。默认的 **variance** 值是 1，实际上也就是等价负载平衡。

- **Router (config-router) #maximum-paths 1-6**——默认情况下，路由器使用 4 条等价路由开销的路径来进行负载共享，用这条命令可以设置 1 到 6 条路径用于负载共享。一组在路由中显示为一跳的指向同一目的地址的多条路径称为一个**负载共享组**。
- **Router (config-router) #traffic-share {balanced | min}**——如果存在多条路由开销最少的路径而且执行了 **traffic-share min** 命令，IGRP 就会使用等价负载平衡。该命令默认设置成 **balanced**，这时负载数据按照度量的比例来分配。例如，如果 **variance** 为 3，并且 **traffic-share** 是设为 **balanced** 的，最佳的那条路径传输的数据量就会是最差路径的 3 倍。
- **Router (config-router) #bandwidth xx-kbit/s**——这条命令设置 IGRP 用于路由选择参数之一的带宽值。

如果某个路由要包含到非等价路由开销的负载平衡过程中去，必须满足 3 个条件：

- 将此路由加入到负载共享组中去不会超过路径数目的最大限制。
- 下游路由器到目的地址的度量必须更优。
- 最低路由开销的度量值在乘上乘数 (**variance**) 以后，必须比要加入到负载共享组中去的路由的度量值大。

10.3.2 IGRP 的非等价路由开销的负载平衡的配置

图 10-2 的网络模型中有两台路由器，klipsch 和 carver；二者通过串行链路相连，一条链路是 56 kbit/s，另一条是 T1 的速率。两台路由器都在 65001 的私有自治域系统中运行 IGRP。在该模型中配置通过这两条链路的非等价路由开销负载平衡。

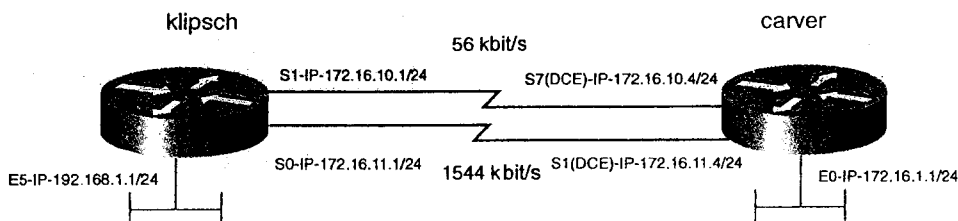


图 10-2 IGRP 的非等价度量计算

基本的 IGRP 配置后，第一步是设置串行接口上的带宽。如果没有调整好带宽就开始运行 IGRP，两台路由器的网络度量就会不一致。例 10-6 中，路由器 klipsch 计算的串口 Serial 1 上子网 172.16.1.0 的组合度量为 180671，而串口 Serial 0 上同样路由的度量则为 8576。

用 **debug ip igmp transactions** 命令可以看到 IGRP 是如何计算每条路由的度量的，结果如例 10-6 所示。

例 10-6 路由器 klipsch 上路由 172.16.1.0 的度量值

```
klipsch# debug ip igmp transactions
IGMP protocol debugging is on
klipsch#
```

(待续)

```

00:50:05: IGRP: received update from 172.16.11.4 on Serial0
00:50:05:      subnet 172.16.10.0, metric 182571 (neighbor 180571)
00:50:05:      subnet 172.16.1.0, metric 8576 (neighbor 1100)
00:50:05: IGRP: received update from 172.16.10.4 on Serial1
00:50:05:      subnet 172.16.11.0, metric 182571 (neighbor 8476)
00:50:05:      subnet 172.16.1.0, metric 180671 (neighbor 1100)
00:50:05:      subnet 172.16.100.0, metric 182671 (neighbor 8576)
klipsch#

```

例 10-7 给出了没有在串行链路两端设置带宽时链路的度量值在路由器上的不一致。

例 10-7 路由器 carver 的 192.168.1.0 路由的度量

```

carver#debug ip igrp transactions
IGRP protocol debugging is on
03:12:26: IGRP: received update from 172.16.11.1 on Serial1
03:12:26:      subnet 172.16.10.0, metric 10476 (neighbor 8476)
03:12:26:      network 192.168.1.0, metric 8576 (neighbor 1100)
03:12:26: IGRP: received update from 172.16.10.1 on Serial7
03:12:26:      subnet 172.16.11.0, metric 10956 (neighbor 8476)
03:12:26:      network 192.168.1.0, metric 89056 (neighbor 1100)
carver#

```

串行接口上使用 **bandwidth** 命令有助于同步双方的度量。使用 **bandwidth kbit/s** 命令后，整个网络中的度量达到一致，如例 10-8 所示。

例 10-8 路由器 klipsch 和 carver 上 debug ip igrp transactions 命令的输出结果

```

klipsch#
03:54:18: IGRP: received update from 172.16.11.4 on Serial0
03:54:18:      subnet 172.16.10.0, metric 182571 (neighbor 180571)
03:54:18:      subnet 172.16.1.0, metric 8576 (neighbor 1100)
03:54:19: IGRP: received update from 172.16.10.4 on Serial1
03:54:19:      subnet 172.16.11.0, metric 182571 (neighbor 8476)
03:54:19:      subnet 172.16.1.0, metric 180671 (neighbor 1100)
03:54:19:      network 192.168.1.0, metric 182671 (neighbor 8576)

carver#
03:49:31: IGRP: received update from 172.16.11.1 on Serial1
03:49:31:      subnet 172.16.10.0, metric 182571 (neighbor 180571)
03:49:31:      network 192.168.1.0, metric 8576 (neighbor 1100)
03:49:31: IGRP: received update from 172.16.10.1 on Serial7
03:49:31:      subnet 172.16.11.0, metric 182571 (neighbor 8476)
03:49:31:      subnet 172.16.1.0, metric 182671 (neighbor 8576)
03:49:31:      network 192.168.1.0, metric 180671 (neighbor 1100)
carver#
carver#

```

路由表一致后，所有的数据都通过路由开销最低的路径来传输。在图 10-3 中可以看到路由器 klipsch 只显示有到网络 172.16.1.0 的最低路由开销路径。

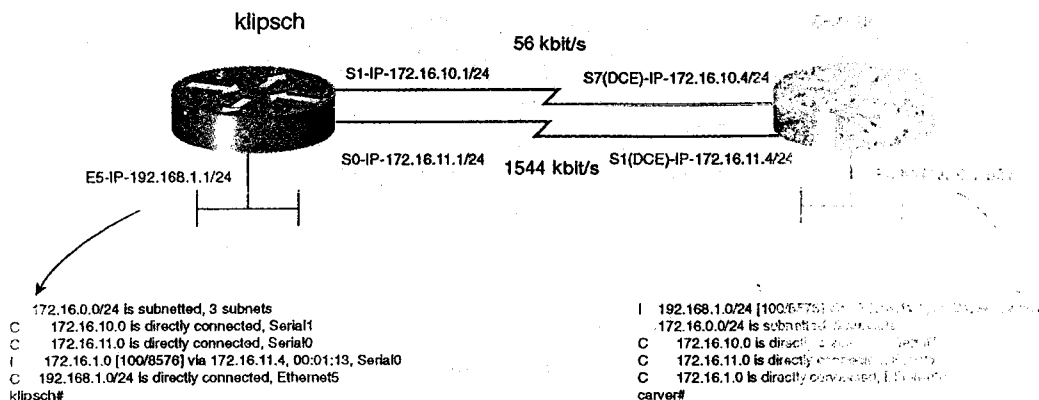


图 10-3 相等度量标的 IGRP 网络

最后一步设置 variance 值。回想一下，计算 variance 值的公式如下：

1+ ([最高路径开销/最低路径开销] 向上舍入)

因此这个例子中，variance 是 $1 + (180671/8576) = 22$ 。variance 设置完后，路由器显示两条路径都通向目的网络并且共享负载。分担的负载量和设置的 variance 值成正比。在该模型中，每 22 个数据包就有一个通过路由开销最高的路径传输。

最后，例 10-9 显示了非等价路由开销负载均衡发生作用时，路由器 klipsch 和 carver 上的路由表同时列出了所有通往目的网络的可选路径。

例 10-9 show ip route 列出了通往远端网络的所有可能的路径

```

carver#
192.168.1.0/24 [100/8576] via 172.16.11.4, 00:01:09, Serial1
I    172.16.0.0/24 [100/180671] via 172.16.10.4, 00:01:10, Serial7
C    172.16.10.0 is directly connected, Serial7
C    172.16.11.0 is directly connected, Serial1
C    172.16.1.0 is directly connected, Ethernet0
carver#

klipsch#
172.16.0.0/24 is subnetted, 3 subnets
C    172.16.10.0 is directly connected, Serial1
C    172.16.11.0 is directly connected, Serial0
I    172.16.1.0 [100/8576] via 172.16.11.4, 00:00:20, Serial0
I    172.16.1.0 [100/180671] via 172.16.10.4, 00:00:20, Serial1
C    192.168.1.0/24 is directly connected, Ethernet5
klipsch#
    
```

例 10-10 则列出了路由器 carver 和 klipsch 的配置。

例 10-10 路由器 carver 和 klipsch 配置中的相关部分

```

hostname carver
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
!
    
```

```

interface Serial1
 ip address 172.16.11.4 255.255.255.0
 bandwidth 1544
 clockrate 2000000
!
interface Serial7
 ip address 172.16.10.4 255.255.255.0
 bandwidth 56
 clockrate 56000
!
router igrp 65001
 variance 22
 network 172.16.0.0
!
 ip classless

hostname klipsch
!
interface Ethernet5
 ip address 192.168.1.1 255.255.255.0
 media-type 10BaseT
!
interface Serial0
 ip address 172.16.11.1 255.255.255.0
 no ip mroute-cache
 bandwidth 1544
!
interface Serial1
 ip address 172.16.10.1 255.255.255.0
 bandwidth 56
!
router igrp 65001
 variance 22
 network 172.16.0.0
 network 192.168.1.0
!
 ip classless

```

10.3.3 IGRP 和 EIGRP 的集成和移植

IGRP 和 EIGRP 的移植毫不费力。基本上，这种说法是正确的，如果 IGRP 和 EIGRP 使用相同的自治域系统标识（ID），可以自动完成协议之间的重分布。这样使得 IGRP 到 EIGRP 的升级成为简单的过程。但是如果二者的路由进程在不同的自治域系统中，就必须手动配置重分布。

IGRP 也是一个有类路由选择协议，也就是说，与接收到路由更新信息的接口处在同一主网或者是位边界内的路由更新信息，必须具有和接收更新信息的接口相匹配的子网掩码。如果更新信息不在同一主网中，更新信息就会在该类的主网边界上自动汇总为 8 位、16 位或者 24 位。因此，使用汇总功能时要非常小心，注意保证支持有类路由的所有网络部分都使用相同长度的子网。

要扩展上例中的网络模型，还需为网络添加新的路由器并更改子网，如图 10-4 所示。图中的模型现在有了一个 EIGRP 域。新的路由器 *dtc* 通过 T1 串行链路路与路由器 *carver* 相连。

要将 dts 路由器集成到网络中，需要手动进行重分布工作，这是因为这个网络处在不同的自治域系统中。例 10-11 给出了在路由器 carver 上进行 IGRP 和 EIGRP 重分布的例子。命令 **passive-interface** 是可选的，这条命令用来防止不必要的 EIGRP hello 信号和 IGRP 广播信息在不必要的区域里传播。

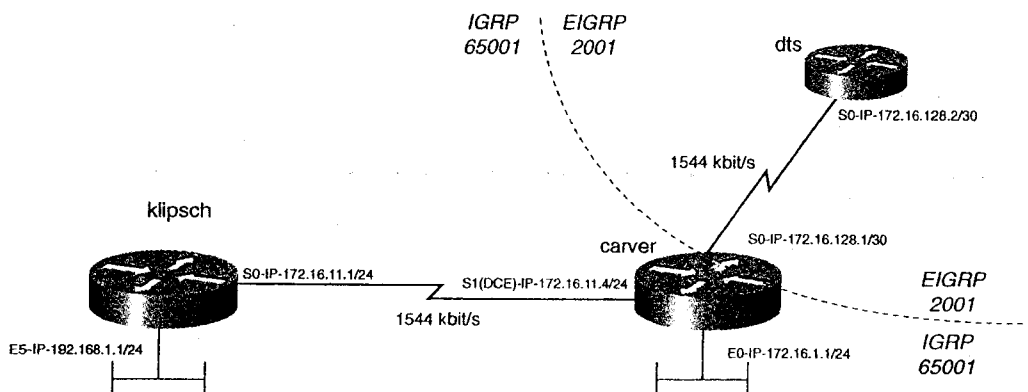


图 10-4 IGRP 和 EIGRP 集成

例 10-11 路由器 carver 上 IGRP 和 EIGRP 的重分布

```
!
router eigrp 2001
 redistribute igrp 65001      <-Redistribute IGRP into EIGRP
 passive-interface Ethernet0  <-do not send EIGRP hellos on these interfaces
 passive-interface Serial1
 network 172.16.0.0
 default-metric 1544 100 255 1 1500 <-Use this metric for redistributed routes
 no auto-summary
!
router igrp 65001
 redistribute eigrp 2001     <-Redistribute EIGRP into IGRP
 passive-interface Serial0
 network 172.16.0.0
 default-metric 1544 100 255 1 1500 <-Default metric for redistributed routes
!
```

路由器 carver 和 dts 之间的链路处在 30 位的边界上。由于该链路的子网掩码与发送接口的子网掩码不匹配，因此，IGRP 不能发送这一网络的更新信息。要解决这个问题，在路由器 carver 的 Serial 0 接口下用命令 **ip summary-address eigrp 2001 172.16.128.0 255.255.255.0** 将网络 172.16.128.0/30 汇总为 172.16.128.0/24 即可。

现在，下游路由器 klipsch 可以到达子网 172.16.128.0/24。例 10-12 列出了路由器 klipsch 上的路由表的情况。

例 10-12 路由器 klipsch 的路由表

```
klipsch#show ip route
<<<text omitted>>>
Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 3 subnets
I       172.16.128.0 [100/10476] via 172.16.11.4, 00:00:31, Serial0
C       172.16.11.0 is directly connected, Serial0
I       172.16.1.0 [100/8576] via 172.16.11.4, 00:00:31, Serial0
klipsch#
```

10.3.4 IGRP 和默认路由

在与 Internet 相连时总是需要默认路由，如果没有默认路由，路由器需要在其路由表中有通往所有网络的路由。基本上，默认路由是指最后手段的网关。路由器在不能为某个数据包找到匹配的路由时，就会把该数据包转发到一个最后手段的网关。Cisco 的路由器默认是有类路由查询，这就意味着除非设置了全局命令 **ip classless**，路由器就不会将数据包正确转发到最后手段的网关。在 Cisco IOS 11.3 或更新的版本上，都默认设置了 **ip classless** 命令。

每个路由选择协议的默认路由概念都不尽相同，不同的路由选择协议使用不同的方法设置使用默认路由。

配置 IGRP 默认路由的两个步骤如下：

第 1 步 定义默认网络。IGRP 不会将 0.0.0.0 的地址视为默认路由。回想一下，IGRP 实际上是将默认路由作为外部网络来发送的。要将一个网络“标记”或定义为外部的，必须要做两件事情。首先必须用 **ip default-network a.b.c.d** 命令对该路由进行标记。然后对于那些想要标记为外部的路由，发送路由的接口和默认网络不能是处在同一主网边界之内的。

第 2 步 确保路由器上配置了 **IP classless**。

图 10-5 中，**igrp_rtr** 的默认网络为 206.191.241.40/29，配置中已经加入了 **ip default-network 206.191.241.0** 命令。这样使得要标记的路由成为默认外部网络。路由器 **igrp_rtr** 的路由表也将该路由用*标记，说明这条路由已经是可用的默认路由了。这条路由会传播到下游路由器去，成为最后手段的网关，如图 10-5 所示。

例 10-13 显示了路由器 **igrp_rtr** 的配置示例。

例 10-13 路由器 igrp_rt 的相关配置

```
router igrp 2001
 network 172.16.0.0
 network 206.191.241.0
!
ip classless
ip default-network 206.191.241.0
!
```

Gateway of last resort is not set

```
172.16.0.0/24 is subnetted, 3 subnets
C 172.16.1.0 is directly connected, Ethernet4
C 172.16.2.0 is directly connected, Ethernet1
I 172.16.3.0 [100/1600] via 172.16.1.9, 00:00:34, Ethernet4
* 206.191.241.0/29 is subnetted, 1 subnets
C 206.191.241.40 is directly connected, Ethernet0
igrp_rtr#
```

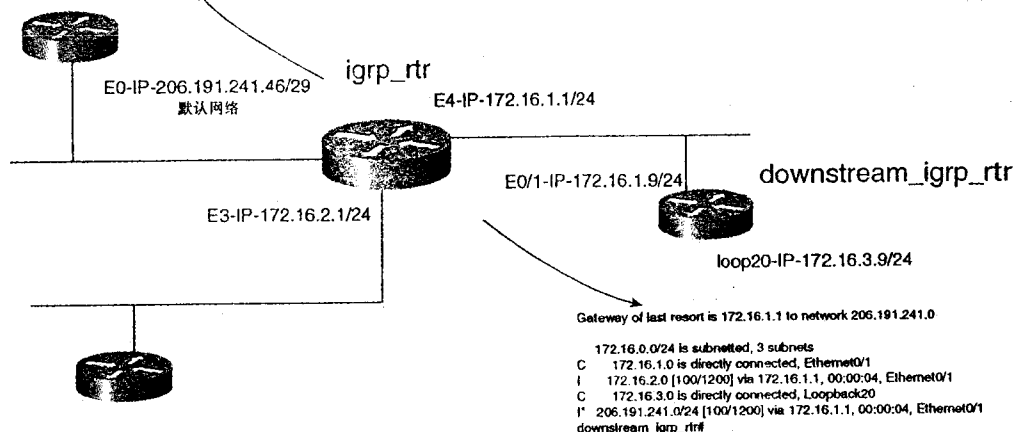


图 10-5 IGRP 的默认网络

10.4 实验 21：配置 IGRP：默认路由、路由过滤和非等价负载平衡——第 1 部分

10.4.1 实验说明

大部分的 IGRP 网络都已经升级成了 EIGRP 网络，但还是有一些保持原状。设计 IGRP 网络的关键就是能够对路由更新进行控制，能正确传播默认路由信息。

这个实验能够让大家有机会练习控制路由更新，过滤以及发送默认路由等内容。

10.4.2 实验内容

Sea Shepherd Conservation Society (SSCS)，也称 Sea Shepherd International，是一个环保主义者的强有力组织，他们致力于保护这个地球上毫无发言权的那些种群。SSCS 拥有一支有组织的海军，称为 Neptune。Neptune 拥有一些军舰，他们工作在这个星球的不同地方，他们在一些生态热点地区巡逻，实施联合国 (UN) 的各种决定以及制裁措施。Neptune Navel Intelligence Network 将 Neptune 各个不同的部门连接在一起，使得 SSCS 可以快捷方便地访问一些重要的数据。我们现在的任务就是以下列要求为准绳设计一个 IGRP 网络：

- 按照图 10-6 配置一个 IP 网络，路由选择协议采用 IGRP，自治系统 (AS) ID 为 65001。
- 配置一个从路由器 sea_shepherd 到 204.30.121.0/24 的默认网络 (路由)，并将其传播到整个 IGRP 区域中。

- 路由器 ocean_warrior 具有一个专用子网 172.16.128.0/24。要禁止 mirage 路由器看到这个子网。
- 对 ocean_warrior 路由器进行配置，使它能够使用通往默认网络的所有可能路径。
- 可选：对 IGRP 进行配置，以免使路由器 ocean_warrior、sirenian 和 mirage 之间的 LAN 网段上网络广播的频率太大。

10.4.3 实验目的

- 按照图 10-6 配置 Neptune 海军情报网络（Neptune Naval Intelligence Network），按图中所示对 IP 进行配置，LAN 的拓扑类型在这个实验中对结果没有影响。
- WAN 上数据链路协议采用 HDLC。
- 对默认网络进行配置，并将其传播到整个 IGRP 域中去。
- 在路由器 ocean_warrior 上配置默认网络的不等路由成本负载平衡。
- 对路由器 ocean_warrior 的子网 172.16.128.0/24 进行过滤，使 mirage 路由器不能访问这个子网。为此，可能会需要配置多个访问控制列表。
- 可选：在路由器 ocean_warrior、sirenian 和 mirage 之间的 LAN 网段上配置网络单播更新。

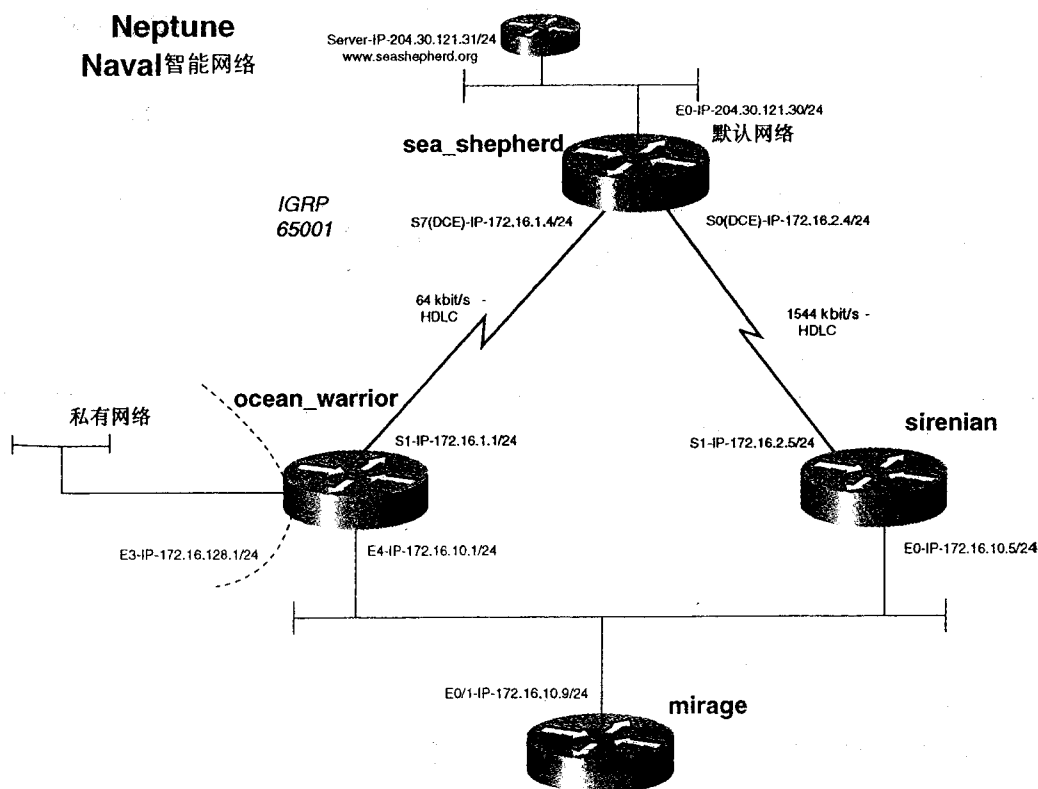


图 10-6 Neptune 海军情报网


```
router igrp 65001
 network 110.0.0.0
 network 172.16.0.0
```

```
hostname sirenian
!
router igrp 65001
 network 172.16.0.0
!
```

```
hostname mirage
!
router igrp 65001
 network 172.16.0.0
!
```

到现在，在所有路由器之间的路由已经建立。利用 **show ip route** 命令以及通过 **ping** 不同的接口对此加以检验。路由器 **sea_shepherd** 上的路由表如例 10-15 所示。

例 10-15 路由器 **sea_shepherd** 上的路由表

```
sea_shepherd#show ip route
<<<text omitted>>>
Gateway of last resort is not set

C    204.30.121.0/24 is directly connected, Ethernet0
     172.16.0.0/24 is subnetted, 4 subnets
I       172.16.128.0 [100/8676] via 172.16.2.5, 00:00:32, Serial0
I       172.16.10.0 [100/8576] via 172.16.2.5, 00:00:56, Serial0
C       172.16.1.0 is directly connected, Serial7
C       172.16.2.0 is directly connected, Serial0
sea_shepherd#
```

观察路由器 **mirage** 上的路由表会发现有两条路由通往网络 204.30.121.0/24，一条通过路由器 **ocean_warrior**，172.16.10.1，另一条则通过路由器 **sirenian**，如例 10-16 所示。这是由于串行接口上的默认带宽没有加以改动，因而 IGRP 得出的路由度量值不正确。要获得正确的路由度量指标，需要在 EAN 连接的串行接口上加上 **bandwidth** 命令。如果没有 **bandwidth** 命令，则路由器 **sirenian** 和 **ocean_warrior** 会把它们的链路视作 T1 链路。而这个例子中的路由器 **sea_shepherd** 是一台 Cisco 2522，其低速的异步端口会得出一个完全不正确的结论，它会认为默认度量指标的带宽是 115。这再一次说明了为什么正确配置路由时需要认真配置和检查 **bandwidth** 命令。

例 10-16 **mirage** 路由器的路由表

```
mirage#show ip route
<<<text omitted>>>
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
I       172.16.128.0 [100/1200] via 172.16.10.1, 00:01:10, Ethernet0/1
C       172.16.10.0 is directly connected, Ethernet0/1
```

（待续）

```

I      172.16.1.0 [100/8576] via 172.16.10.1, 00:00:41, Ethernet0/1
I      172.16.2.0 [100/8576] via 172.16.10.5, 00:00:44, Ethernet0/1
I*    204.30.121.0/24 [100/8676] via 172.16.10.5, 00:00:44, Ethernet0/1
mirage#

```

在串行接口上加入 **bandwidth** 命令之后，mirage 路由器就会优选通过 sirenian 路由器的线路，也就是 172.16.10.5，前往 204.30.121.0/24 网络的路由。分别在路由器 ocean_warrior 和 sirenian 的串行接口上加入了 **bandwidth 64** 命令和 **bandwidth 1544** 命令之后，mirage 路由器上的 **show ip route** 命令执行结果如例 10-17 所示。这些命令同样也应加在 sea_shepherd 路由器上。

例 10-17 增加 bandwidth 命令之后路由器 mirage 的路由表情况

```

mirage#show ip route
<<<text omitted>>>
Gateway of last resort is not set

172.16.0.0/24 is subnetted, 4 subnets
I      172.16.128.0 [100/1200] via 172.16.10.1, 00:01:10, Ethernet0/1
C      172.16.10.0 is directly connected, Ethernet0/1
I      172.16.1.0 [100/158350] via 172.16.10.1, 00:01:10, Ethernet0/1
I      172.16.2.0 [100/8576] via 172.16.10.5, 00:00:19, Ethernet0/1
I*    204.30.121.0/24 [100/8676] via 172.16.10.5, 00:00:19, Ethernet0/1
mirage#

```

实验的下一步就是在路由器 sea_shepherd 上配置一个默认网络（默认路由），这是通过在 sea_shepherd 路由器上加上一条全局命令 **ip default-network 204.30.121.0** 来实现的。要记住一点，某台路由器要使用默认路由的时候，还必须加上一条 **ip classless** 命令。现在来看看 mirage 上的路由表，默认路由是用*来表示的，最后路由手段网关设置成了 204.30.121.0。例 10-18 就是在路由器 sea_shepherd 上设置了默认路由之后 mirage 路由表的示例。

例 10-18 在 sea_shepherd 上设置了默认路由之后 mirage 上的路由表

```

mirage#show ip route
<<<text omitted>>>
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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.10.5 to network 204.30.121.0

172.16.0.0/24 is subnetted, 4 subnets
I      172.16.128.0 [100/1200] via 172.16.10.1, 00:03:06, Ethernet0/1
C      172.16.10.0 is directly connected, Ethernet0/1
I      172.16.1.0 [100/158350] via 172.16.10.1, 00:01:18, Ethernet0/1
I      172.16.2.0 [100/8576] via 172.16.10.5, 00:00:27, Ethernet0/1
I      110.0.0.0/8 [100/1200] via 172.16.10.1, 00:01:18, Ethernet0/1
I*    204.30.121.0/24 [100/8676] via 172.16.10.5, 00:00:27, Ethernet0/1
mirage#

```

该实验的另一个任务是在始于 mirage 路由器的默认路由上配置非等路由开销的负载均衡。这一章的前面部分提过，需要对带宽进行设置，然后算出度量差值并对其进行配置。前面用 **bandwidth** 命令对 IGRP 进行了配置，因而完成了前一个步骤。然后就要求出差值。度量差分值公式是：

$1 + ([\text{最高路径代价}/\text{最低路径代价}] \text{ 相除结果向上舍入})$

利用 **debug ip igrp transactions** 找到通往 204.30.121.0/24 的最低路由度量指标。例 10-19 就是该命令在路由器 ocean_warrior 上的执行结果。

例 10-19 路由器 ocean_warrior 上 debug ip igrp transactions 命令的输出结果

```
ocean_warrior#debug ip igrp transactions
IGRP protocol debugging is on
ocean_warrior#
04:35:06: IGRP: received update from 172.16.1.4 on Serial1
04:35:06: subnet 172.16.10.0, metric 160350 (neighbor 8576)
04:35:06: subnet 172.16.2.0, metric 160250 (neighbor 8476)
04:35:06: exterior network 204.30.121.0, metric 158350 (neighbor 1100) ←High
metric
04:35:06: IGRP: received update from 172.16.10.9 on Ethernet4
04:35:06: subnet 172.16.10.0, metric 1200 (neighbor 1100)
04:35:06: subnet 172.16.2.0, metric 8676 (neighbor 8576)
04:35:06: exterior network 204.30.121.0, metric 8776 (neighbor 8676) ←low
metric
```

在这个网络模型中，差值 = $1 + (158350/8776) = 20$ 。设置好差值之后，路由器会报告说到目的网络有两条路径，负载是在二者上共享的。在 IGRP 过程中加入 **variance 20** 命令之后，路由器 ocean_warrior 上的路由表如例 10-20 所示。

例 10-20 路由器 ocean_warrior 上 show ip route 命令的输出结果

```
ocean_warrior:#show ip route
<<<text omitted>>>
Gateway of last resort is 172.16.1.4 to network 204.30.121.0

172.16.0.0/24 is subnetted, 4 subnets
C    172.16.128.0 is directly connected, Ethernet3
C    172.16.10.0 is directly connected, Ethernet4
C    172.16.1.0 is directly connected, Serial1
I    172.16.2.0 [100/160250] via 172.16.1.4, 00:00:11, Serial1
      [100/8576] via 172.16.10.5, 00:00:34, Ethernet4
I*   204.30.121.0/24 [100/158350] via 172.16.1.4, 00:00:11, Serial1
      [100/8676] via 172.16.10.5, 00:00:34, Ethernet4
      Two paths to the default network!
ocean_warrior#
```

接下来要做的是把关于 ocean_warrior 的子网 110.16.20.0/24 的路由从 mirage 路由器中过滤出去。为此就需要在路由器 ocean_warrior 和 sirenian 上配置相应的分布列表。例 10-21 给出了 ocean_warrior 上分布列表的配置情况，而 sirenian 上的分配列表除了以太网端口之外完全一样。

例 10-21 路由器 ocean_warrior 上分布列表的配置

```
ocean_warrior(config)#router igrp 65001
ocean_warrior(config-router)#distribute-list 10 out e4
ocean_warrior(config-router)#exit
ocean_warrior(config)#access-list 10 deny 172.16.128.0 0.0.0.255
ocean_warrior(config)#access-list 10 permit any
```

从 ocean_warrior 路由器的以太网端口上发送路由的时候，路由 172.16.128.0/24 会被过滤掉，sirenian 会试图通过路由器 sea_shepherd 来到达这个网络。要避免出现这种情况，可以在 sirenian 上加入一条静态路由。例 10-22 列出了不带子网 172.16.128.0/24 路由器 mirage 上的路由表。

例 10-22 过滤之后 mirage 路由器上的路由表示例

```
mirage#show ip route
<<<text omitted>>>
Gateway of last resort is 172.16.10.5 to network 204.30.121.0

    172.16.0.0/24 is subnetted, 3 subnets
C       172.16.10.0 is directly connected, Ethernet0/1
I       172.16.1.0 [100/158350] via 172.16.10.1, 00:00:08, Ethernet0/1
I       172.16.2.0 [100/8576] via 172.16.10.5, 00:00:27, Ethernet0/1
I*    204.30.121.0/24 [100/8676] via 172.16.10.5, 00:00:27, Ethernet0/1
mirage#
```

实验最后的可选部分是对路由器之间以太网段上的网络广播加以限制。网络广播的限制是通过启动单播路由更新来实现的。命令 **neighbor a.b.c.d** 加上 **passive interface** 就可以实现只更新单播路由。例 10-23 是最后的路由器配置情况，单播的配置部分也包含在内。

例 10-23 Neptune 海军情报网络的路由器最终配置清单

```
hostname sea_shepherd
!
interface Ethernet0
ip address 204.30.121.30 255.255.255.0
!
interface Serial0
ip address 172.16.2.4 255.255.255.0
bandwidth 1544
no fair-queue
clockrate 2000000
!
interface Serial7
ip address 172.16.1.4 255.255.255.0
bandwidth 64
clockrate 64000
!
router igrp 65001
network 172.16.0.0
```

(待续)

```

network 204.30.121.0
!
ip classless
ip default-network 204.30.121.0
!

hostname ocean_warrior
!
interface Ethernet3
 ip address 172.16.128.1 255.255.255.0
 media-type 10BaseT
!
interface Serial1
 ip address 172.16.1.1 255.255.255.0
 bandwidth 64
!
router igrp 65001
 variance 20
 passive-interface Ethernet4
 network 110.0.0.0
 network 172.16.0.0
 neighbor 172.16.10.5
 neighbor 172.16.10.9
 distribute-list 10 out Ethernet4
!
ip classless
!
access-list 10 deny 172.16.128.0 0.0.0.255
access-list 10 permit any

hostname sirenian
!
interface Ethernet0
 ip address 172.16.10.5 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 ip address 172.16.2.5 255.255.255.0
 bandwidth 1544
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
router igrp 65001
 passive-interface Ethernet0
 network 172.16.0.0
 neighbor 172.16.10.1
 neighbor 172.16.10.9
 distribute-list 10 out Ethernet0
!
ip classless
ip route 172.16.128.0 255.255.255.0 172.16.10.1

hostname mirage
!
interface Ethernet0/1
 ip address 172.16.10.9 255.255.255.0
!
router igrp 65001
 passive-interface Ethernet0/1
 network 172.16.0.0
 neighbor 172.16.10.5

```

（待续）

```
neighbor 172.16.10.1
!
ip classless
```

如果想要对单播更新加以确认，可以在路由器 *sirenian* 或 *ocean_warrior* 上使用 **debug ip igrp transactions** 命令。例 10-24 是这条命令在 *sirenian* 上的执行情况，可见，端口 Ethernet 0 发送出的路由更新信息已不再发往广播地址 255.255.255.255，而发往一个用 **neighbor** 命令指定的特定地址。

例 10-24 验证单播更新配置

```
sirenian#debug ip igrp transactions
IGRP protocol debugging is on
sirenian#
01:01:40: IGRP: sending update to 255.255.255.255 via Serial0 (172.16.2.5)
01:01:40:      subnet 172.16.10.0, metric=1100
01:01:40:      subnet 172.16.1.0, metric=158350
01:01:41: IGRP: sending update to 172.16.10.9 via Ethernet0 (172.16.10.5)
01:01:41:      subnet 172.16.10.0, metric=1100
01:01:41:      subnet 172.16.1.0, metric=158350
01:01:41:      subnet 172.16.2.0, metric=8476
01:01:41:      exterior 204.30.121.0, metric=8576
01:01:41: IGRP: sending update to 172.16.10.1 via Ethernet0 (172.16.10.5)
01:01:41:      subnet 172.16.10.0, metric=1100
01:01:41:      subnet 172.16.2.0, metric=8476
01:01:41:      exterior 204.30.121.0, metric=8576
```

第 11 章

混合协议：增强型 内部网关路由选择 协议（EIGRP）

随着 Internet 网络在上个世纪 90 年代早期在规模和种类方面的快速增长，需要新的路由选择协议满足发展的需要。为了顺应这一需要，解决当时 IGRP 和 RIP 等协议的缺陷，Cisco 推出了增强型内部网关路由选择协议（EIGRP）。WAN 的发展需要产生一种能够高效地利用 WAN 连接和 LAN 网络地址空间的路由选择协议。OSPF 可以达到这样的要求，但是它需要的 CPU 资源量却不适合当时大多数只有小型处理器的边缘或远程路由器，其配置也比 RIP 或 IGRP 复杂得多。当时需要一种既能够支持 VLSM，也能够扩充 Internet 网络，但又不像 OSPF 那样占用 CPU 资源的路由选择协议。1994 年，Cisco 在 IOS 9.21 中推出了增强型 IGRP，满足了这一系列的需求。现在，EIGRP 已经用在很多的大型政府和商业网络中。实践证明，EIGRP 是稳定、灵活而高速的路由选择协议。除了这些特点外，EIGRP 配置的便捷性也使它成为最受网络工程师青睐的一种路由选择协议。

EIGRP 可以看成是混合协议，它综合了传统的距离矢量协议和链路状态协议的特点。

具体一点，EIGRP 的“增强”体现在下面这 4 种路由技术的使用：

- 邻居路由器的发现/恢复。
- 可靠传输协议（RTP）。
- DUAL 有限状态机制。
- 协议相关模块。

本章将讲述这些技术以及 RIGRP 的工作原理和配置方法。

11.1 EIGRP 技术概览

和其他路由选择协议相比，EIGRP 有如下优势：

- 支持 VLSM——EIGRP 是无类路由选择协议，路由的子网掩码和路由更新信息一起传输。
- 快速收敛——通过采用扩散刷新算法 (DUAL) 定义的可行备选 (feasible successors) 的概念，预先选出通往目的地址的下一条最佳路径，这使得链路出故障后能够快速收敛。
- 低 CPU 占用率——通常的工作情况下，只有 hello 信号和部分更新信息会通过链路传输。路由更新信息不会扩散 (flooding)，而对这些更新信息的处理也只是周期性的。
- 增量更新信息——EIGRP 不发送完整的路由表，只发送有变化的路由信息。
- 可扩展性——由于采用 VLSM 和复杂的组合度量，EIGRP 网络规模的可扩展性增强。
- 配置简单——EIGRP 支持分层网络设计方式，但不像 OSPF 需要非常严格的配置规则。
- 自动路由汇总——EIGRP 能够在主网边界上自动进行路由汇总。
- MD5 路由认证——从 Cisco IOS 11.3 开始，EIGRP 支持对路由更新信息进行 MD5 密码加密认证。

看到上面列出的这些特点就会理解为什么 EIGRP 能够成为如此受欢迎的路由选择协议。EIGRP 能提供 OSPF 具备的多项增强功能，却没有 OSPF 那样严格的配置规则。有人说 EIGRP 的弱点就是它是 Cisco 的私有协议，但是通过路由选择协议重分布，这个弱点不再成其为问题。

EIGRP 是无类路由选择协议，基于 IP 协议传输，IP 协议号为 88。EIGRP 采用多播地址 224.0.0.10 来进行 hello 信号和路由更新信息的发送，而不是像 RIP 那样使用广播地址。EIGRP 还采用了 hello 机制和抑制计时器来维持邻居路由器。除了初始的路由更新信息之外，只有当网络拓扑结构发生改变时才会有路由更新信息发送。路由更新信息的发送还会受到限制，即更新信息只能发送到相关的路由器。像 IGRP 一样，EIGRP 采用组合度量来计算到某目的地址的最佳路径。下面各节会介绍 EIGRP 如何在工作过程中使用这些度量、邻居路由器、可靠传输以及扩散刷新算法 (DUAL)。

注释 EIGRP 早期版本存在着低速串行链路的稳定性问题和难以维持多邻居路由器的问題。Cisco 在 Cisco IOS 10.3 (11)，11.0 (8) 和 11.1 (3) 中解决了这些问题。现在通常把早期的 EIGRP 版本统称为 EIGRP V.1。目前 Cisco 提供的都是装有 IOS 12.0 以上版本的路由器。

11.1.1 EIGRP 的度量

EIGRP 度量方法和 IGRP 一样。路由表中的每条路由由都有一个与之相关联的度量值。和

IGRP 一样，EIGRP 也采用组合度量，不同的是这里的度量被乘上了乘数 256。回想一下第 10 章“距离矢量协议：内部网关路由选择协议（IGRP）”的内容，度量因素包括带宽、延时、负载、可靠性以及最大传输单元（MTU）。还是和 IGRP 一样，EIGRP 选择路由主要还是根据其带宽和延时，或者是说是最小的组合度量值。在为某个路由计算度量时，EIGRP 称此指标为该路由的可行距离。EIGRP 会计算网络中所有的路由的可行距离。下面是这些因素的细述：

- 带宽——带宽的单位是 kbit/s，必须静态设置带宽，以准确反应运行着 EIGRP 的接口。例如速度为 56 kbit/s 的接口和 T1 接口的默认带宽为 1544kbit/s。要准确地设置带宽值，可以使用接口命令 **bandwidth kbit/s**。表 11-1 列出了一些常用的接口带宽值。
- 延时——延时的单位是微秒，也必须静态设置，以便准确地表达运行 EIGRP 的接口。调整延时的接口命令是 **delay time_in_micoseconds**。表 11-1 也列出了一些常用的延时值。
- 可靠性——可靠性是一个范围为 1 到 255 的动态值，255 表示绝对可靠的链路而 1 代表不可靠链路。
- 负载——负载是一个 1 到 255 之间的数，表示接口的输出负载值，其值是动态的，可以用 **show interfaces** 命令查看。该值为 1 表示链路负载最小，而 255 则是表示链路已经满负载。
- 最大传输单元（MTU）——最大传输单元（MTU）是路径中记录下来的最小 MTU 值，通常是 1500。

注释 配置 IGRP 或 EIGRP 中影响路由选择因素时，应该优先考虑使用延时，最好不用带宽。这是因为带宽变化可能对其他路由选择协议（如 OSPF）造成影响，而延时的变化只影响 IGRP 和 EIGRP。

表 11-1 列出了常用的一些度量。

表 11-1 常用的 IGRP 和 EIGRP 度量

介 质	带 宽	延 时
100M ATM	100000 kbit/s	100 微秒
吉比特以太网	100000 kbit/s	100 微秒
快速以太网	100000 kbit/s	100 微秒
FDDI	100000 kbit/s	100 微秒
HSSI	45045 kbit/s	20000 微秒
16M 令牌环	16000 kbit/s	630 微秒
10M 以太网	10000 kbit/s	1000 微秒
T1	1544 kbit/s	20000 微秒
DS-0	64 kbit/s	20000 微秒

EIGRP 使用的组合度量 (CM) 就是从这 5 个度量因素计算出来的。计算时，EIGRP 采用了包含 5 个常量或者是“k”值的公式。这些常数有以下默认值：

$k1 = k3 = 1$ 和 $k2 = k4 = k5 = 0$

这里的 $k2$ 、 $k4$ 和 $k5$ 设为 0，本质上来说就是忽略了负载、可靠性和 MTU 这 3 个子度量的作用。这也正是为什么在改变 EIGRP 对路由的选择时要首选改变延时和带宽值的原因。

EIGRP 用来计算组合度量的公式如下：

组合度量 $CM = 256 \{ [k1 \times \text{最小带宽} + (k2 \times \text{最小带宽}) / (256 - \text{负载}) + k3 \times \text{延迟的和}] \times X \}$

这里，

最小带宽 = $10^7 /$ 路径中最慢链路的带宽

延迟的和 = Σ (路径中的所有延迟)

$X = k5 / (\text{可靠性} + k4)$ 当且仅当 $k1 > 1$ ；如果 $k1 = 1$ ，那么 $X = 1$

k 值设为默认值时，

$k1 = k3 = 1$

$k2 = k4 = k5 = 0$

$CM = 256 (\text{最小带宽} + \text{延迟的和})$

注释 路由器这种方法计算出来的组合度量的值和我们计算出来的结果总是有一些轻微的差别，这是由于路由器在进行浮点数学运算过程中的舍入偏差造成的。

采用常数的默认值， $k1 = k3 = 1$ 和 $k2 = k4 = k5 = 0$ ，公式可以简化为：

$(256 \times [\text{最小带宽} + \text{延迟的和}])$

将常数值代入公式，我们可以得到：

$CM = 256 \{ [1 \times \text{最小带宽} + (0 \times \text{最小带宽}) / (256 - \text{负载}) + 1 \times \text{延迟的和}] \times 1 \}$

$CM = 256 \{ [\text{最小带宽} + (0) / (256 - \text{负载}) + \text{延迟的和}] \times 1 \}$

$CM = 256 (\text{最小带宽} + \text{延迟的和})$

注释 作为参考，IGRP 的度量计算方法和上面计算类似，只是带宽和延时的结果没有乘 256，而延迟的和变量用 10 除了一次。

$CM = \{ k1 \times \text{最小带宽} + [k2 \times \text{最小带宽}] / [256 - \text{负载}] + [k3 \times \text{延迟的和}] \times X \}$

这里，最小带宽 = $10^7 /$ 路径中最慢链路的带宽

延迟的和 = Σ (路径中的所有延迟) / 10

$X = k5 / (\text{可靠性} + k4)$ 当且仅当如果 $k1 > 1$ ；如果 $k1 = 1$ 那么 $X = 1$

$k1 = k3 = 1$

$k2 = k4 = k5 = 0$

k 取默认值，

$CM = \text{最小带宽} + \text{延迟的和}$

可以参考图 11-1 查看组合度量的计算情况。在该例中，EIGRP 在 alpha 路由器上计算了从 alpha 到 charlie 路由器 172.16.1.0/24 的路由组合度量。

假设工程师已经设置好了 **bandwidth** 声明，alpha 和 charlie 之间路径上的最低带宽是

56 kbit/s。这样

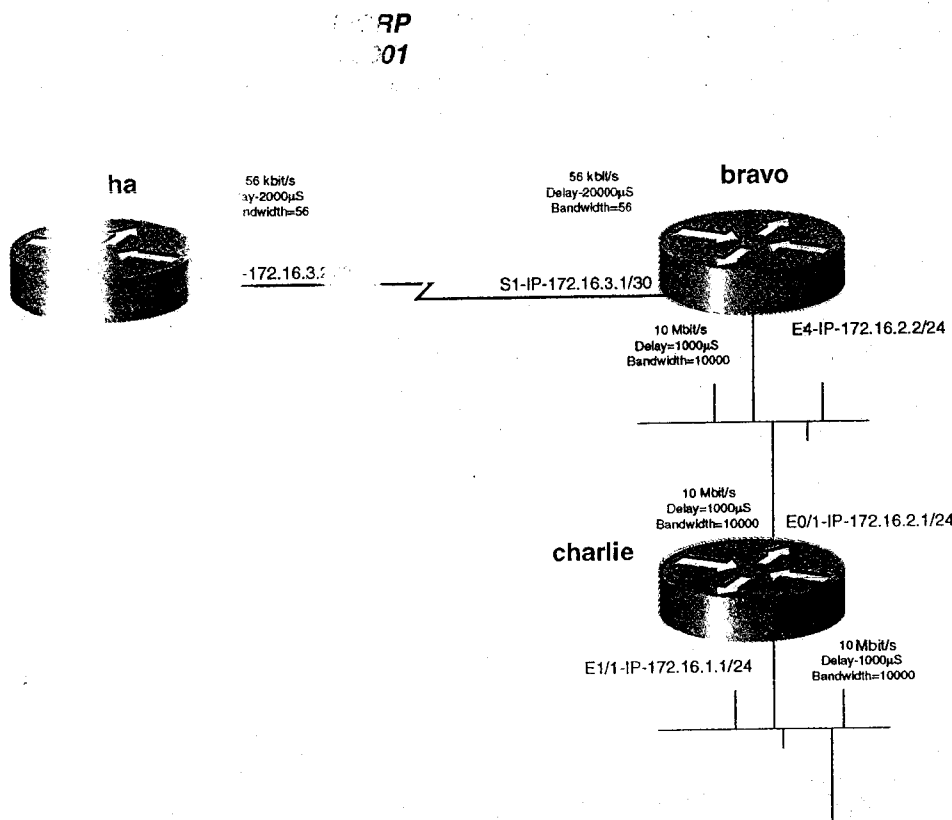


图 11-1 EIGRP 路由更新

最小 $= 10^7 / 178571$
这里 是输入 口上的 的总和，这个总和一直加到最后的子网所在的接口上的
延时。从 a 到 br 的延时是 1000，从 bravo 到 charlie 是 1000，还包括 charlie 上最后
这个接口 时 1000 一样，
延迟 $= 20000 + 1000 + 1000 = 22000$
现在 合度量 出来是
CM $\times (178571) + 2 \times (22000) = 46277485$
如例 所示， alpha 路 上用 `show ip route 172.16.1.0` 命令可以查看度量因子和
且合度量 记住， 处理器 运算舍入操作的缘故，计算的度量和显示结果并不是完
全吻合的

例 用 `show ip route` 显示 EIGRP 的度量值

```
al show ip route 172.16.1.0
Ro entry for 172.16.1.0/24
via "Ethernet0/1", distance 90, metric 46277376, type internal
Distributing a eigrp update
update for 172.16.3.1 on Serial17, 00:50:53 ago
```

(待续)

Routing Descriptor Blocks:

```
* 172.16.3.1, from 172.16.3.1, 00:50:53 ago, via Serial7
  Route metric is 46277376, traffic share count is 1
  Total delay is 22000 microseconds, minimum bandwidth is 56 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 2
```

alpha#

采用接口命令 **delay xx** 可以运用度量来改变路由的选择。记住，如果希望对称路由，一定要在接口链路两端都配置延时。对称路由指数据包会沿着相同的路径返回源端。默认情况下，EIGRP 会在路由之间进行等价路由开销的负载平衡。例如，执行 **show ip route** 命令时发现到同一目的地址有两条路由，EIGRP 会在这两条路由上进行负载平衡。

为了让大家了解延时的用法，在 bravo 和 charlie 之间加入另一以太网段，在 charlie 路由器上增加了一个回路接口 172.16.128.1/24，如图 11-2 所示。

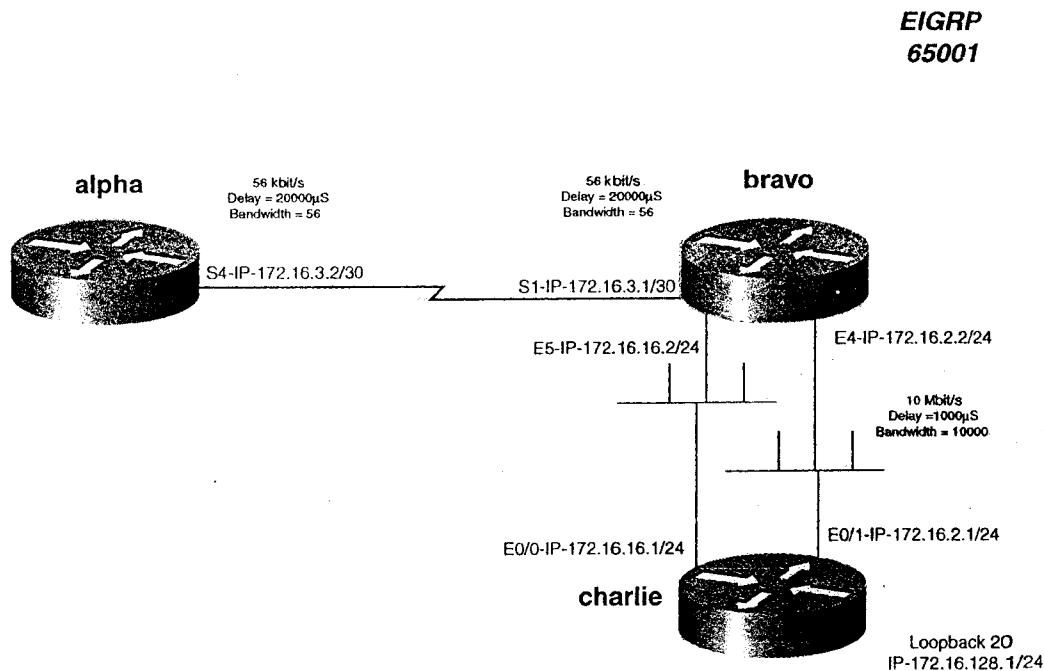


图 11-2 EIGRP 的负载共享

在 bravo 路由器上执行 **show ip route** 命令会发现该路由器有两条到网络 172.16.128.0/24 的路由，如例 11-2 所示。命令 **show ip eigrp topology** 也能列出这些路由以及它们的组合度量。

例 11-2 到 172.16.128.0/24 去的两条路由

```
bravo#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
```

(待续)

```

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, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D    172.16.128.0/24 [90/409600] via 172.16.2.1, 00:23:50, Ethernet4
      [90/409600] via 172.16.16.1, 00:23:50, Ethernet5
C    172.16.16.0/24 is directly connected, Ethernet5
C    172.16.2.0/24 is directly connected, Ethernet4
C    172.16.3.0/30 is directly connected, Serial1
bravo#

```

如果希望 EIGRP 优先选择某一条路径，可以在接口的两端加入 **delay** 命令。要明改变链路延时仅仅对路由选择协议造成影响，但与链路本身的实际数据传输没有任何关系。

接着来看这个例子，现在设置延时的值使得通往 172.16.128.0 的主要链路设定为 172.16.16.1 的链路。可以通过在 bravo 的 E4 接口和 charlie 的 E0/1 接口上加上 1000 的延迟。

例 11-3 给出了 bravo 路由器上设置延时的情况。

例 11-3 加入 delay 命令

```

bravo#conf t
Enter configuration commands, one per line. End with CNTL/Z.
bravo(config)#int e4
bravo(config-if)#delay 1000
bravo(config-if)#^Z

```

例 11-4 则是显示的在 bravo 和 charlie 上加了延时之后 bravo 路由器的路由表的情况

例 11-4 通往 172.16.128.0/24 的路由

```

bravo#show ip route
172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D    172.16.128.0/24 [90/409600] via 172.16.16.1, 00:00:11, Ethernet5
C    172.16.16.0/24 is directly connected, Ethernet5
C    172.16.2.0/24 is directly connected, Ethernet4
C    172.16.3.0/30 is directly connected, Serial1
bravo#

```

要记住，尽管路由表中删除了第 2 条路由，但是 EIGRP 仍然知道这条路由的存在而将它视为一个可行备选。

也可以通过改变 *k* 值来改变路由的选择，这可以用 **metric weights tos k1 k2 k3 k4 k5** 来完成。这些值的改变会直接影响 EIGRP 对于所有路由组合度量的计算，因此只有在 Cisco 路由器来解决某些问题时才改变这些加权值。

11.1.2 EIGRP 的邻居路由器

EIGRP 并不周期性地发送全部路由信息。因此，它需要用某种方法找到邻近设备，然后再进行路由信息的交换。EIGRP 通过邻居路由器的使用来做到这一点。初始化时，EIGRP 会将多播 hello 信号发送到地址 224.0.0.10 (在广播介质上)。在 NBMA 介质、X.25、帧中继以及 ATM 上，每 60 秒以单播形式发送 hello 信号。EIGRP 周期性发送 hello 信号的间隔时间与介质类型有关。

在下列介质上，EIGRP 每 5 秒发送一次 hello 信号：

- LAN 的广播介质，如以太网，令牌环和 FDDI。
- 高速串行连接 (速率高于 T1)，如帧中继高速串行接口 (HSSI)。
- 点对点串行链路，如 PPP 或 HDLC。
- ATM 和帧中继点对点接口。

在下列介质接口中，EIGRP 每 60 秒发送一次 hello 信号：

- 低速串行接口 (低于 T1 速率)，包括帧中继和多点 X.25。
- ATM 和帧中继的点对多点接口以及 ATM 的 SVC。
- ISDN 的 BRI。

处于同一网络中的路由器接收多播 hello 信号，然后做出应答，形成邻接关系。图 11-3 以及下面列举的就是初始化进程中形成这种邻接关系的过程：

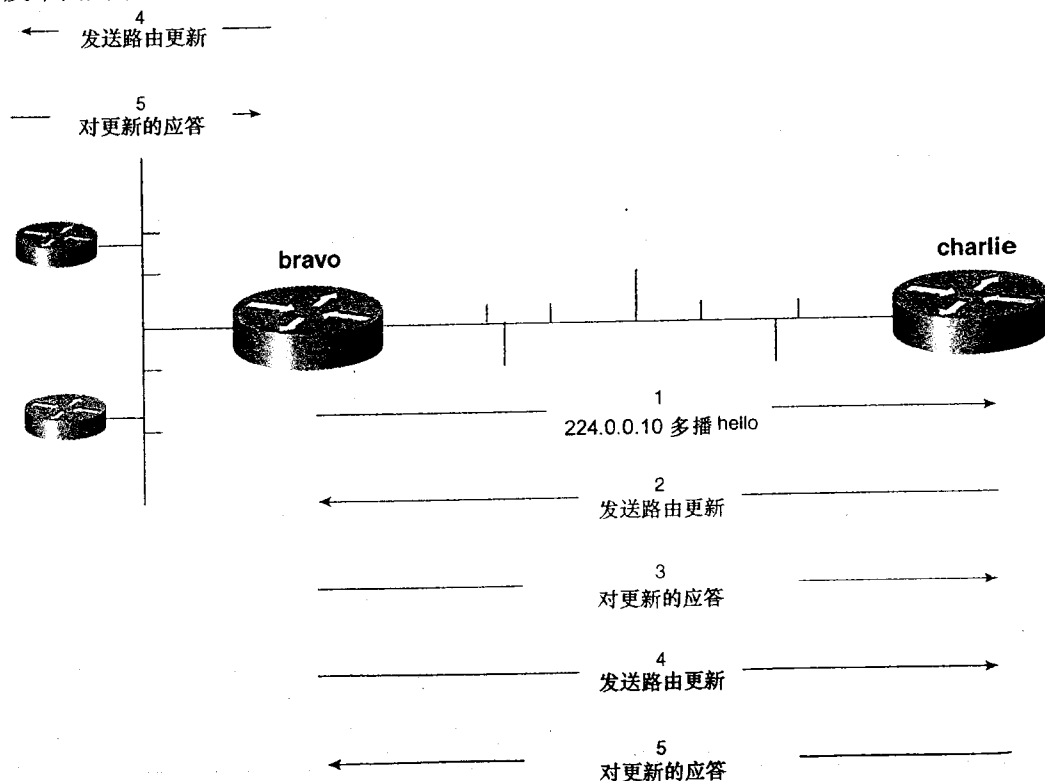


图 11-3 EIGRP 邻居路由器的建立

1 除了设定为被动模式（passive）的接口，所有参与 EIGRP 工作的接口都发出 hello 信号。所有的 hello 信号和路由更新信息发送都采用多播地址 224.0.0.10。

2 同一 IP 子网中的路由器接收到多播信息，用一个完整的路由更新作为应答，这是通过设置 EIGRP 报头中的 INITIALIZATION 位来实现。路由更新信息包括 EIGRP 识别到的所有网络以及这些路由的度量（除了被水平分隔抑制了的路由）。这个更新信息数据包在 hello 信号的收发双方之间建立起邻接关系（adjacency）。hello 数据包中还含有一个等待计时器，设定了在路由器接收到 hello 信号之前和将其判定为不可到达之前应该等待的时间，并将此信息报告给扩散刷新算法（DUAL）进程。等待计时器的值设定为 hello 计时器的 3 倍，通常是 15 或 180 秒，这与介质类型有关。

3 路由器 bravo 通过发送设置了 ACK 位的 hello 信号来响应初始化数据包。EIGRP 用设置 ACK 位的方法对接收到的含有数据在内的所有信息进行确认。这也是 EIGRP 能进行可靠的传输的原因之一（具体细节将会在后面章节进行讲解）。

4 现在，路由器 bravo 往路由表中插入新的更新信息。由于有了新的更新信息，bravo 会发送更新信息到所有的邻居路由器。

5 接收到 bravo 发送更新信息的邻居路由器用确认数据包作为应答。

6 路由器通过交换 hello 信号保持邻接关系。如果在等待计时器的时间里没有接收到 hello 信号，路由器将此路由标记为不可达。

建立邻接关系之后，路由器将此关系看作传输路由信息的虚链路。

路由器用下列信息创建邻接表：

- 接收的 hello 信号的源 IP 地址。
- 保持计时器的值。
- SRTT 或者是往返时间。
- 邻居路由器的正常运行时间。

用 **show ip eigrp neighbors** 命令可以查看邻居路由器的状态，如例 11-5 所示。邻居路由器的正常运行时间也就是邻接关系建立的时间。

例 11-5 bravo 上 show ip eigrp neighbors 命令的输出

```
bravo#show ip eigrp neighbors
IP-EIGRP neighbors for process 65001
H   Address                Interface    Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)          Cnt Num
1   172.16.2.1              Et4          12 01:10:36    8       200  0  29
2   172.16.16.1             Et5          13 02:14:15    3       200  0  28
0   172.16.3.2              Se1          11 07:07:44    23      2604  0  23
bravo#
```

稳定的 EIGRP 邻接关系是 EIGRP 网络中最为重要的因素。没有稳定的邻接关系，EIGRP 网络就无法正常工作。检查任何 EIGRP 网络工作情况的第 1 步就应该查看邻接关系的状态。

11.1.3 EIGRP 的可靠传输协议（RTP）

保正常的发送，EIGRP 采用了 Cisco 私有的可靠多播报文。邻居路由器接收到可靠多播数据包时，要求以一个单播确认数据包来作为应答。更新信息也有序列号，路由器通过序列号来确保更新信息按照正常的顺序发送。为了实施 RTP 以及其他 EIGRP 功能，Cisco 采用了 4 种主要的数据包类型（实际上使用 5 种数据包）。如前所述，所有的 EIGRP 数据包都直接利用 IP 层 88 号协议接口通信，多播更新使用 IP 地址 224.0.0.10。所用的 5 种数据包类型如下：

- **Hello**——用来发现和维持邻居路由器。该数据包类型使用不可靠发送方式。
- **确认应答 (ACK)**——用于对数据包作出确认。本质上来说，确认数据包是没有数据的 hello 数据包。该数据包类型使用不可靠发送方式。
- **路由更新**——包含了路由更新信息。根据产生的方式不同，路由更新数据包可以是单播的也可以是多播。这类数据包采用可靠传输方式。
- **查询**——DUAL 进程用来查找路由的可行备选的数据包，可以是单播的，也可以是多播的，采用可靠的传输方式。
- **应答**——DUAL 进程用来帮助查找路由的可行备选的数据包。应答数据包总是以可靠方式进行单播传输。

注释 一些文献中称查询数据包和应答数据包为第 4 种和第 5 种数据包。实际上，第 5 种数据包是请求数据包。EIGRP 从没有采用请求数据包，这是为路由服务器准备的。IPX SAP 在 EIGRP 包头中另外采用了其他类型的数据包。

11.1.4 扩散刷新算法 (DUAL)

DUAL 算法是 EIGRP 的“大脑”，负责跟踪所有邻居路由器的所有路由情况，确保网络拓扑中没有任何环路。DUAL 以 E.W. Dijkstra 和 C.S. Scholten 最初提出的算法为基础，后来又经过了 J.J. Garcia-Luna-Aceves 的补充和完善。

通过 DUAL、EIGRP 以及上面提到的进程，EIGRP 中维护以下表：

- **邻接表**——EIGRP 将创建的邻接关系记录在邻接表中。和邻居路由器的邻接关系会一直保持，除非连续 16 次都没有从该路由器接收到确认单播数据 ACK，它就从邻接表中删去该邻居路由器。`show ip eigrp neighbors` 命令可以查看邻居路由器的情况。
- **拓扑表**——所有从邻居路由器学习到的路由信息都存储在拓扑表中。拓扑表还记录与路由相关联的度量和可行距离。拓扑表的情况可以用 `show ip eigrp topology as_number` 命令来查看。
- **路由表/转发表**——只有具有最低组合度量值的路由才会成为最终的路由表或称转发表中。这是路由器转发数据的路由。

DUAL 维持无环路网络拓扑的过程非常严谨。EIGRP 路由表中的每条路由都有一个**首选路由**和**可行备选路由**。**首选路由**是路由的主路径，或者是说路由器用来转发数据包的路径。而**可行备选路由**则会在主路由变为不可达之后成为下一跳地址。**可行备选路由**总是在其下游位置，因此它的可行距离必须小于当前路由的可行距离。这样，由于下游路由器要成为可行备选者，所以必须具有低于当前路由开销值的可行路由开销，因此可以避免路由

环路。

DUAL 管理着可行距离、可行备选路由以及 EIGRP 拓扑表中各路由的备选路由的确定问题。由于已经在拓扑表中定义了备份路径，因此当主路径出现问题时，路由器可以很快地收敛到新的备份路径上。

11.1.5 协议相关模块

EIGRP 是少数几个能够和多个被路由协议协同工作的路由选择协议。Cisco 在代码中采用 *协议相关模块* 来处理各个协议不同之处。例如，IPX EIGRP 需要接收和发送 SAP 更新信息。IP 和 IPX 利用不同的报文格式来形成邻接关系。

EIGRP 的工作和所有的路由选择协议一样。也就是说，它利用 DUAL 来找到转发数据的最短路径。另一些相关协议模块则将数据送入 DUAL 进程处理，以便形成适当的拓扑表以及最终的路由表。

和 IGRP 一样，EIGRP 采用了水平分隔和逆向抑制来避免路由环路的出现。

11.2 水平分隔

回想一下，以前讲的 *水平分隔* 路由技术，它不允许路由信息从接收到该路由的接口或子接口中再发送出去。在多点网络中，水平分隔的使用非常普遍。路由更新信息进入某个子接口以后，不可以再从同一个子接口发送到多点网络中其他路由器。默认情况是激活水平分隔的，用来避免 EIGRP、IGRP 和 RIP 的路由更新信息在多点配置环境中不正常的传播。可以用 `no ip split-horizon eigrp autonomous system` 命令关闭此功能。IPX 和 AppleTalk 中，该命令的也有相似的形式。

图 11-4 中，路由器 grinch 从路由器 whos 和 whoville 接收到路由更新信息，但由于水平分隔，grinch 不会通过其串行 s0.1 多点接口宣告 172.16.5.0 和 172.16.6.0 信息。由于 grinch 不是从串口 s0.1 学到的 172.16.2.0 网络信息，它会将该网络发送给路由器 whos 和 whoville。

要让路由器 whos 和 whoville 正常接收信息，需要做的是在路由器 grinch 的子接口上用 `no ip split-horizon eigrp` 命令关闭水平分隔，如例 11-6 所示。

例 11-6 关闭 grinch 路由器上的水平分隔

```
grinch(config)#int s0.1
grinch(config-subif)#no ip split-horizon eigrp 2001
```

图 11-5 则是关闭 grinch 上的水平分隔后该路由器上的路由表的情况。所有的路由都在正常传输。

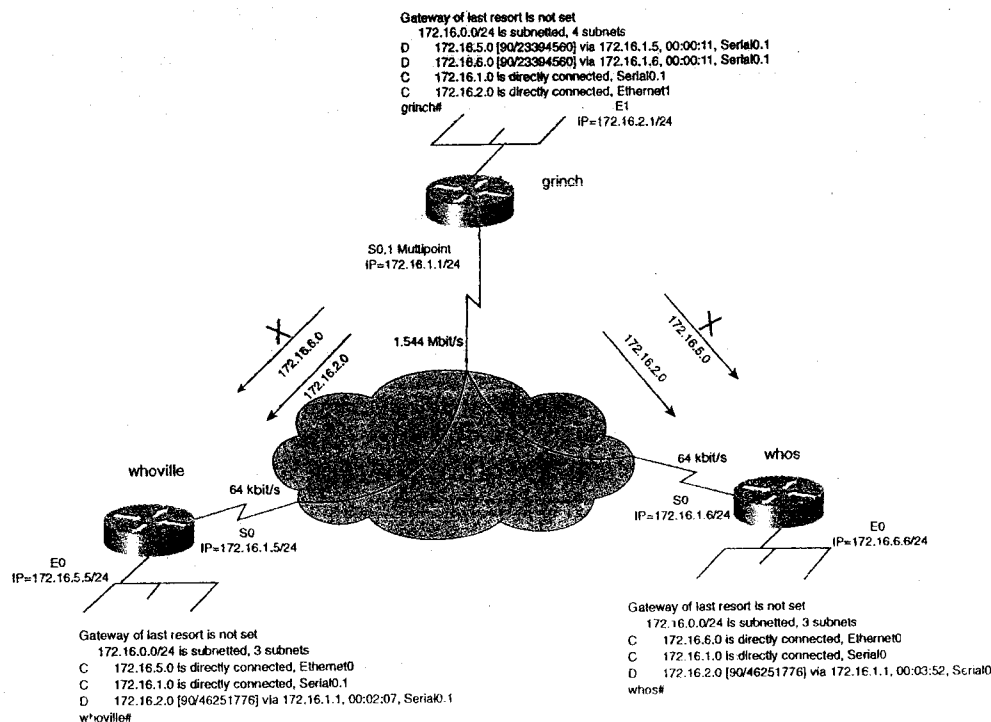
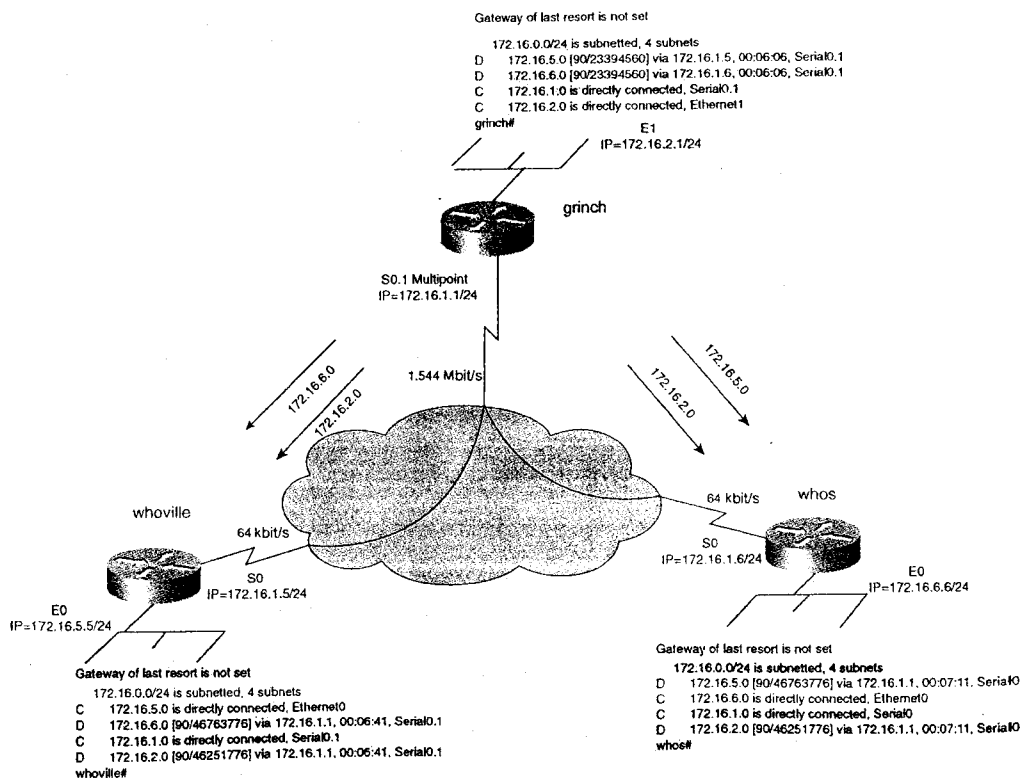


图 11-4 EIGRP 水平分隔的路由抑制作用



11.3 EIGRP 的配置

在很大程度上，基本 EIGRP 的配置和 IGRP 完全一样。EIGRP 的配置需要定义自治域系统（AS）。自治域系统（AS）是定义为一组在同一管理机构控制下的路由器的集合。和 IGRP 一样，EIGRP 利用 AS 的概念分离路由进程。配置 EIGRP 时并不需要拥有注册过的 AS。

用以下 3 个步骤来配置 EIGRP。第 3 个步骤是针对某些特定环境的可选项。

第 1 步 允许 EIGRP 运行并在路由器上定义 AS。可以通过全局命令 **router eigrp autonomous_system_id** 来完成。

第 2 步 加入想要运行 EIGRP 的网络。可以在 config-router#提示符下执行 **network a.b.c.d** 命令。输入 **network** 声明之后，还需要提供主网边界。在 Cisco IOS 12.0 以及更新的版本中，**network** 命令加入了 **wildcard mask** 选项，就像 OSPF 一样。这其实是反向掩码位。例如，只想在网络 172.16.1.0 上运行 EIGRP，命令句法就应该是 **network 172.16.1.0 0.0.0.255**，但是请注意，即使你不小心输入了子网掩码，EIGRP 也能够智能地将子网掩码转换为反向掩码。这是个不错的功能。

第 3 步 （可选）用 **bandwidth** 命令对 EIGRP 的度量进行微调，或者是配置 EIGRP 的汇总和其他可选项。花一些时间对带宽作一下设置能够更准确地了解网络情况，还有助于避免网络广播引起链路拥塞。帧中继网络上一般应该设置带宽值，接口命令 **bandwidth kbit/s** 可以完成这一功能。本章的后面各节还会更详细地讲述带宽和 EIGRP 汇总方面的问题。

例 11-7 是图 11-5 中路由器 grinch 的配置。

例 11-7 EIGRP 的配置实例

```
! hostname grinch
!
interface Ethernet1
 ip address 172.16.2.1 255.255.255.0
 media-type 10BaseT
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip split-horizon eigrp 2001      ←Split Horizons disabled
 bandwidth 112                      ←Bandwidth set to the sum of the remote PVCs
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 130 broadcast
!
router eigrp 2001                    ←EIGRP routing process
 network 172.16.0.0                  ←Networks running EIGRP
!
```

在进一步详细讨论这些和其他 EIGRP 选项之前，先来看一看 EIGRP 的 **show** 命令。

11.4 EIGRP 的 “Big show” 和 “Big D” 命令

Cisco 提供了一些很好用的工具来判断 EIGRP 的工作状况，其中最好却又最容易被忽略的就是 **show ip eigrp neighbors** 命令。EIGRP 的邻居路由器让我想起了 Robert Frost 的诗句，“有好的栅栏就有好的邻居（Good fences make good neighbors）”，在 EIGRP 中则是“有好的网络就有好的邻居路由器（Good networks make good neighbors）”，邻居路由器的状态对于 EIGRP 的工作至关重要。Cisco 提供的调试工具除了能够查看邻居路由器的状态之外。还能查看 EIGRP 的拓扑表情况以及 EIGRP 事件的详细记录。

下面列出一些最为有用的 EIGRP 的 **show**，**logging** 和 **debug** 命令：

```
show ip eigrp neighbors [ as_number | interface_name]
show ip eigrp topology [ as_number | active | pending | summary] [ as_number subnet
    subnet_mask]
show ip protocols [summary]
show ip route
debug eigrp packets
eigrp log-neighbor-changes
```

11.4.1 show ip eigrp neighbors 命令

这是检验 EIGRP 工作状态时最为有用的命令，能显示所有 EIGRP 邻居路由器的状态。EIGRP 要在链路上运行，邻居路由器应该处在“up”状态。EIGRP 形成的邻接关系包含了同一自治域系统同一子网中所有的路由器。EIGRP 在 k 值不匹配的情况下不会形成邻接关系，但是，路由器可以在 hello 周期和“致命”计时器不匹配的情况下形成邻接。正常运行时间短的邻居路由器很可能是有问题。还有很重要的一点是队列数。队列数是指正在等待传送到某个邻居路由器去的数据包的数目，它的值（Q）应该是 0 或者是小于 20 的数。Q 值如果始终在 60 附近或者更大，那就是过高。高 SRTT 值表明数据包正处于链路某种类型的延时。例 11-8 是 **show ip eigrp neighbor** 命令的输出结果，下面讲解例子中出现的字段的内容：

例 11-8 路由器 grinch 上 show ip eigrp neighbor 命令的执行结果

```
grinch#show ip eigrp neighbors
IP-EIGRP neighbors for process 2001
H   Address                Interface    Hold Uptime    SRTT    RTO  Q  Seq
   (sec)                  (ms)          Cnt Num
1   172.16.1.5              Se0.1       136 05:48:23   36   1302  0  15
0   172.16.1.6              Se0.1       131 05:48:24   40   1302  0  17
grinch#
```

- **句柄 Handle (H)** ——Cisco IOS 用来对邻居路由器进行标识的内部号码。不要把此句柄和跳计数相混淆。

- 邻居路由器地址——邻居路由器的 IP 地址。邻接关系形成于自治域系统中运行 EIGRP 的同一子网内所有路由器之间。
- 接口——邻居路由器记录的接口。
- 等待时间（HoldTime）——倒计数的时间，是 EIGRP 等待某个邻居路由器的 hello 信号，直到关闭这个邻居路由器为止的那段时间。
- 正常运行时间（Uptime）——邻居路由器持续正常工作的时间，也等于和邻居路由器之间链路正常工作的时间间隔。
- 顺利往返时间（SRRT）——从一个 EIGRP 数据包发送到邻居路由器再到本地路由器接收到邻居路由器返回的确认信号之间的间隔时间（ms）。如果这个时间的值为 0，那么数据包根本就没有成功地进行本地路由器和邻居路由器之间的往返。
- 超时重发时间（RTO）——EIGRP 等待重发队列中数据重新发送到邻居路由器之前的总时长，通常用微秒表示。
- 队列数（Q）——在队列中等待发送到邻居路由器的数据包数目。它的值应该是 0 或者是很小的数。高队列数说明数据传输出现问题。
- 序列号（Seq-Num）——上一次接收到的从该邻居路由器发送来的更新信息，查询或者是应答数据包的序列号。如果该数为 0，说明该邻居路由器没有发送可靠的数据包到本地路由器，表明邻居路由器出了问题。

注释 仅仅根据某个网络出现在路由表中并不能说明路由正常工作。某些情况下，如计时器不匹配等，网络也可能会进入或者移出路由表。判断网络的工作情况，还要参考其他因素，如邻居路由器和数据库等，这样才能正确判断网络路由是否正常工作。

11.4.2 show ip eigrp topology 命令

这条命令能够列出前面讨论过的 EIGRP 的拓扑表的情况，包括 EIGRP 发现了的所有路由的情况以及 EIGRP 是否正在处理某条路由上的信息等。多数情况下，路由应该是处于被动状态，并且没有相关这条路由的 EIGRP 算法进程。如果某路由处在活动状态中，表明处于~~陷入~~活动状态，或称 SIA，这将在后面进行详细的讲解。这条命令也可以扩展显示单个路由或子网的信息。包括与路由有关的所有信息，全部的度量以及路由的备选条目，还包括路由是如何学到的。例 11-9 给出了 show ip eigrp topology 命令的用法，后面是扩展命令的使用范例。

例 11-9 路由器 grinch 上的 EIGRP 拓扑表

```
grinch#show ip eigrp topology
IP-EIGRP Topology Table for process 2001

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
```

（待续）

```

P 172.16.5.0/24, 1 successors, FD is 23394560
    via 172.16.1.5 (23394560/281600), Serial0.1
P 172.16.6.0/24, 1 successors, FD is 23394560
    via 172.16.1.6 (23394560/281600), Serial0.1
P 172.16.1.0/24, 1 successors, FD is 23368960
    via Connected, Serial0.1
P 172.16.2.0/24, 1 successors, FD is 281600
    via Connected, Ethernet1
grinch#

grinch#show ip eigrp topology 2001 172.16.5.0 255.255.255.0
IP-EIGRP topology entry for 172.16.5.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 23394560
Routing Descriptor Blocks:
  172.16.1.5 (Serial0.1), from 172.16.1.5, Send flag is 0x0
    Composite metric is (23394560/281600), Route is Internal
    Vector metric:
      Minimum bandwidth is 112 Kbit
      Total delay is 21000 microseconds
      Reliability is 254/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 1
grinch#

```

这条命令的执行结果中值得注意的字段有：

- **P**——被动状态，没有 EIGRP 算法进程，这是理想的状态。
- **A**——活动状态，EIGRP 计算被激活，正在处理该目的地址。路由不停地出现在活动状态表明邻居路由器或者是查询数据包出现问题，这二者都属于 SIA 问题。
- **U**——更新状态，送往目的地址的更新数据包。
- **Q**——查询状态，送往目的地址的查询数据包。
- **R**——应答状态，送往目的地址的应答数据包。
- **Route information**——路由信息，包括某个路由或网络的 IP 地址、相应的子网掩码、首选路由、或者是该网络的下一跳地址以及可行备选者。
- **Send Flag**——发送标志，该目的地址需要发送数据包的类型。
 - 0x1: 路由器接收到该网络的应答信号，需要发送单播应答。
 - 0x2: 路由处于活动状态，应该发送多播查询信号。
 - 0x3: 路由发生了改变，应该发送多播更新信号。

11.4.3 show ip protocols 命令

这条命令可以显示所有的路由选择协议信息，详细的计时器和度量的信息以及路由更新的信息。例 11-10 就是该命令的执行示例。

例 11-10 show ip protocols 命令的示例

```

grinch#show ip protocols
Routing Protocol is "eigrp 2001"          ←AS system ID
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is

```

(待续)

```

Default networks flagged in outgoing updates
Default networks accepted from incoming updates
EIGRP metric weights K1=1, K2=0, K3=1, K4=0, K5=0, K6=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 2001
Automatic network summarization is in effect (1 is auto-summary in effect)
Routing for Networks:
  172.16.0.0
  ----- Networks running EIGRP
Routing Information Sources:
  Gateway         Distance      Last Update
  172.16.1.6      90            00:08:48
  172.16.1.6      90            00:08:52
Distance: internal 90 external 170
          ----- Default admin distance
grinch#
    
```

11.4.4 show ip route 命令

这条命令能够列出路由器当前的路由表，或称转发表，包括路由来自什么协议（这里的 D 代表 EIGRP 内部路由，D EX 代表重分布进入 EIGRP 的路由）。路由后面括号中的数值是该路由的管理距离，再后面是其组合度量的值。字段 **via** 是说明了路源，最近从该接口接收的更新信息时长。

例 11-11 列出了这条命令的示例。

例 11-11 show ip route 命令的示例

```

grinch#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

  172.16.0.0/24 is subnetted, 4 subnets
D       172.16.5.0 [90/23394560] via 172.16.1.5, 00:17:51, Serial0.1
D       172.16.6.0 [90/23394560] via 172.16.1.6, 00:29:06, Serial0.1
C       172.16.1.0 is directly connected, Serial0.1
C       172.16.2.0 is directly connected, Ethernet1
grinch#
    
```

11.4.5 debug eigrp packets 命令

EIGRP 的“Big D”命令就是：大。前面讲过，debug 命令总是和日志记录结合使用。但是，EIGRP 的“Big D”命令输出的结果太多了，因此需要用一些附加的命令来控制原有 debug 命令的输出。命令 **debug eigrp packets** 就是这些需要附加命令来控制的命令中的一个。

用 **debug eigrp packets** 命令可以确认 EIGRP 的 hello 信号是否正常交换，邻接关系是否

已建立等。每一个接收到的和发送出去的 EIGRP 数据包也都会罗列在这条命令的输出结果中。这条命令的输出显示结果可以用另一些调试选项来加以控制，如 **debug ip eigrp [neighbor as_number IP_address_of_neighbor]**。使用 **debug ip eigrp** 这条命令时要小心，调试时该命令只是进一步了解问题。不要根据这条命令的结果匆忙开始进行 EIGRP 故障的排除工作。例 11-12 列出了该命令的示例。

例 11-12 debug eigrp packets 命令的示例

```
grinch#debug eigrp packets
06:22:29: EIGRP: Received HELLO on Serial0.1 nbr 172.16.1.5
06:22:29: AS 2001, Flags 0x0, Seq 0/0 idbQ 0/0
06:22:29: EIGRP: Enqueueing UPDATE on Serial0.1 nbr 172.16.1.5 iidbQ un/rely 0/1
peerQ un/rely 0/0 serno 2-10
06:22:29: EIGRP: Requested unicast on Serial0.1
06:22:29: EIGRP: Sending UPDATE on Serial0.1 nbr 172.16.1.5
06:22:29: AS 2001, Flags 0x1, Seq 7/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely
0/1 serno 2-10
```

11.4.6 eigrp log-neighbor-changes 命令

EIGRP 提供了一条独特的日志命令，在缩小网络故障的原因范围时非常有用。这条路由器命令 **eigrp log-neighbor-changes** 可以检测 EIGRP 邻居路由器的变化情况。例 11-3 列出了一个 EIGRP 等待时间过期之后的记录结果。

例 11-13 EIGRP 邻居路由器变化之后的记录

```
grinch(config-router)#eigrp log-neighbor-changes
06:42:12: %DUAL-5-NBRCHANGE: IP-EIGRP 2001: Neighbor 172.16.1.6 (Serial0.1) is down: holding time expired
```

11.5 调整 EIGRP 的更新信息

和 IGRP 一样，EIGRP 也有参数用来调整计时器、控制网络广播、负载分担以及管理网络路由等。下面是对这些参数的说明：

- Router (config-if) **#ip hello-interval eigrp as_number interval_in_seconds**——这条接口命令可以改变 EIGRP 的 hello 计时器值。默认情况下，该命令和接口有关。按照默认值，hello 数据包每 5 秒发送一次，只有在低速和非广播多路访问介质 (NBMA) 的情况下例外，是 60 秒。所谓的低速，是定义在 T1 (1.544 Mbit/s) 或更低的速率。处于同一网络中的所有邻居路由器应该具有相同的 hello 计时器。
- Router (config-if) **#ip hold-time eigrp as_number holdown_timer_in_seconds**——用这条命令可以改变 EIGRP 对该接口上接收路由的等待计时器值。这个值默认情况下，对那些低速和 NBMA 网络是 180 秒，而对其他网络则为 15 秒。处于一个网络中的所有邻居路由器应该具有相同的等待计时器值。

11.6 EIGRP 的重分布和路由控制

可以采用路由过滤列表对 EIGRP 中的路由更新信息进行过滤。过滤列表调用标准或者扩展的访问控制列表对路由更新信息进行相应的过滤处理。在将一个路由选择协议重分布进另一个时，可以使用 **redistribute** 命令以及默认的度量值。在重分布过程中对某些路由进行控制时，则应该用路由图（route map）来代替分布列表。IGRP 和 EIGRP 在同一个自治域系统中时，两个协议之间的重分布自动进行。

- Router (config-router) **#distribute-list [1-199] [in | out] [interface]**——这条命令能够调用标准或扩展的访问控制列表来对输入的或者是输出的路由更新进行过滤。选项 **in** 和 **out** 都是对接口而言的，换句话说，要防止更新信息从接口发送，就使用 **out** 选项；要禁止路由更新信息从接口进入路由器，就采用 **in** 选项。
- Router (config-router) **#redistribute [connected | static | bgp | rip | igrp | ospf | isis] {metric} {route-map}**——这条命令用于将其他路由选择协议重分布进 EIGRP。要对路由做额外控制，可以加上路由图。还可以为要进行重分布的路由选择协议的相关路由条目提供不同于当前默认度量值的度量（可选）。在重分布路由时，请记住，IP 需要有在本地路由器和目的地址之间往返的路由，这通常是通过双向重分布来实现的。
- Router (config-router) **#default-metric [bandwidth_kbit/s 1-4214748364] [delay_ms 1-4214748364] [reliability 1-255] [load 1-244] [mtu 1-4214748364]**——用这条命令可以设置重分布进 EIGRP 的所有路由的默认度量值。重分布时，必须提供一个默认度量。常用的度量是 **default-metric 1544 100 254 1 1500**。该命令让路由器通过 1544 的带宽值和 100 的延时值，链路可靠性为 254（255 为 100%可靠），链路负载为 1（没有负载）以及 MTU 为 1500 这样的度量来计算组合度量。比实际的度量值可能更重要的就是养成在整个 EIGRP 域中使用同样的度量值的习惯，这样所有重分布后的路由就具有同样的加权值。

注释 路由选择协议重分布时，必须使用默认度量或者在 **redistribute** 命令中提供一个度量值。

下面的子命令集可以用来改变 EIGRP 对路由的选择，除了可以改变 EIGRP 的管理距离之外，还可以改变单个的度量值。不管什么时候，如果要改变某个链路的度量，最好使用 **delay** 命令，而不要用 **bandwidth**。尽管两个命令都可用，但是 **bandwidth** 命令还会影响其他路由选择协议，如 OSPF，而 **delay** 命令只是影响 IGRP 和 EIGRP。

- Router (config-router) **#metric weights 0 k1 k2 k3 k4 k5**——这条命令可以用来更改 EIGRP 的度量的各个因素：带、负载、延时和可靠性的加权值。改变这些值时要小心，EIGRP 不会和那些 K 值不匹配的路由器建立邻接关系。
- Router (config-router) **#distance [1-255] adjacent_neighbors_ip_address wildcard_mask [access_list_0-99]**——这条命令可以用来改变从某个邻居路由器接收到的路由管理

距离。如果省略了 IP 地址和反向掩码，协议所有的路由管理距离都会设成这个值。大家可以在第 10 章“距离矢量协议：内部网关路由选择协议 (IGRP)”中看到这方面的详细内容。

- Router (config-if) **#delay** [1-4214748364]——该命令用来指定某个接口的延时，单位是毫秒。这条命令用于路由选择协议，且不会影响链路上的数据量。
- Router (config-if) **#bandwidth** [bandwidth_kbit/s 1-4214748364]——指定某个接口的带宽，单位是 kbit/s。这条命令只用于路由选择协议，不会影响链路上的数据量。
- Router (config-router) **#passive-interface** interface_name——禁止在该链路上发送 EIGRP 的 hello 信号。这条命令的工作和 IGRP 的不一样。由于抑制了 hello，不能与其他路由器建立邻接关系，因此也不会发送和接收到任何路由更新信息。
- Router (config-router) **#offset-list** [access_list_0-99 {in | out} offset [metric_offset_1-214748364] [interface]——用于增加路由度量的值，增加的偏移量不能超过 214748364。偏移量列表的用法和 RIP 中一样。可以参考第 9 章“距离矢量协议：路由信息协议版本 1 和版本 2 (RIP-1 和 RIP-2)”中这方面的例子和应用。

11.6.1 实例：EIGRP 重分布的应用

将这些概念应用到实际的网络模型中，进行路由的重分布和管理。图 11-6 的网络模型含有 3 个路由域。路由器 canada 和帧中继网络处在 EIGRP 域中。跨过帧中继网络中还有两个其他路由域：路由器 mexico 在 IGRP 域中，而路由器 usa 则在一个 OSPF 域中。

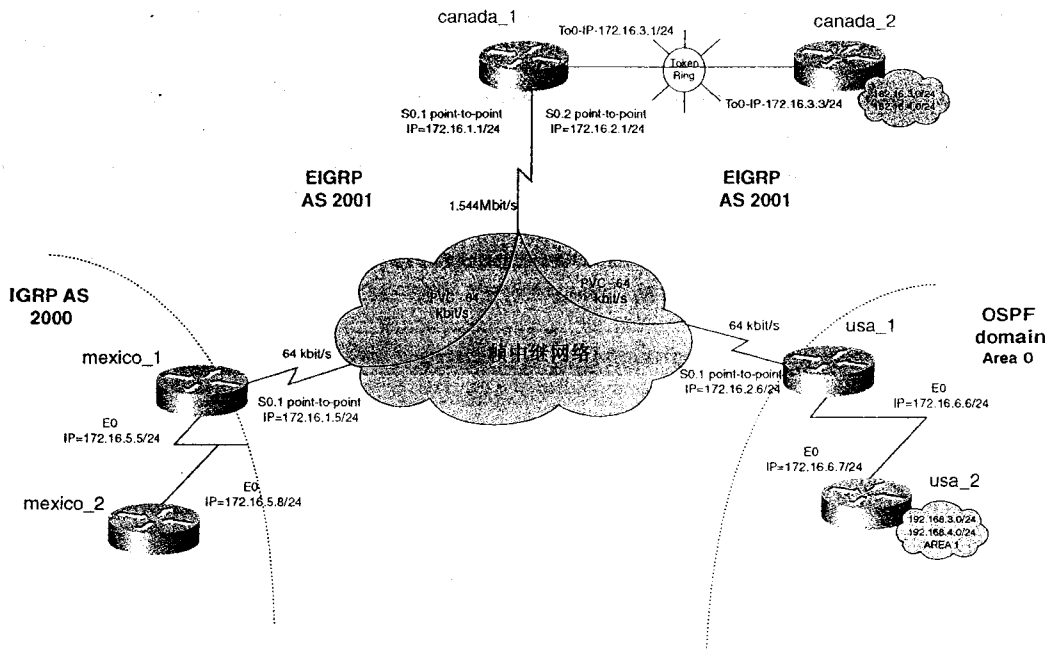


图 11-6 用于 EIGRP 重分布和路由管理的 EIGRP 网络实例

要实现 IP 的端到端连接，必须首先在路由域中确认两件事：

- IGRP 域是一个 24 位界的域，也就是说，IGRP 域收到一个路由时，该路由要被接

受，就必须是处在一个主网边界上或者是该接口的一个 24 位边界。

- EIGRP 和 IGRP 之间，EIGRP 和 OSPF 之间必须进行相互重分布。

首先配置路由器 `canada_1`，按照本章前面讲的配置 EIGRP 的 3 步骤过程进行。第 1 步，由于所有的 EIGRP 路由器都在定义好的自治域系统 2001 中，所以将此作为 AS 的 ID。接着，要运行 EIGRP 的网络现在都在主网 172.16.0.0 中，可以作为 `network` 命令的参数。第 3 步是可选的，但在这里，由于在帧中继上配置 EIGRP，因此在串行子接口下使用 `bandwidth` 命令是个不错的选择。这个网络模型中，将带宽值设为远程路由器帧中继接口的端口速率。例 11-14 给出了路由器 `canada_1` 的配置示例。

例 11-14 路由器 `canada_1` 的配置

```
hostname canada_1
!
interface Serial0
  no ip address
  encapsulation frame-relay
  no ip mroute-cache
!
interface Serial0.1 point-to-point
  ip address 172.16.1.1 255.255.255.0
  bandwidth 64          ←EIGRP bandwidth set
  frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
  ip address 172.16.2.1 255.255.255.0
  bandwidth 64          ←EIGRP bandwidth set
  frame-relay interface-dlci 130
!
interface TokenRing0
  ip address 172.16.3.1 255.255.255.0
  ring-speed 16
!
router eigrp 2001          ←EIGRP routing enabled
  network 172.16.0.0      ←Networks running EIGRP
```

路由器 `mexico_1` 和 `usa_1` 的 EIGRP 配置可以用同样的方法进行，只有稍许不同。这两台路由器不希望在路由器的以太网络上建立任何 EIGRP 邻接关系。因此，在路由器 `mexico_1` 和 `usa_1` 的 EIGRP 下加入 `passive-interface ethernet 0` 命令。例 11-15 列出了到现在为止路由器 `mexico_1` 和 `usa_1` 的配置示例。关于配置中的 IGRP 和 OSPF 部分，可以参考第 10 章以及第 12 章“链路状态协议：开放式最短路径优先（OSPF）”。

例 11-15 路由器 `mexico_1` 和 `usa_1` 的 EIGRP 配置

```
hostname mexico_1
!
interface Ethernet0
  ip address 172.16.5.5 255.255.255.0
  no ip directed-broadcast
!
interface Serial0
  no ip address
  no ip directed-broadcast
```

（待续）

```

C    172.16.1.0 is directly connected, Serial0.1
C    172.16.2.0 is directly connected, Serial0.2
C    172.16.3.0 is directly connected, TokenRing0
D    182.16.0.0/16 [90/304128] via 172.16.3.3, 00:43:27, TokenRing0
canada_1#
canada_1#show ip eigrp neighbors
IP-EIGRP neighbors for process 2001
H    Address                Interface    Hold Uptime    SRTT    RTO    Q    Seq
      (sec)                  (ms)          Cnt Num
2    172.16.3.3              To0          11 00:43:36    685    4110    0    3
1    172.16.2.6              Se0.2        14 1d06h      48     2280    0    28
0    172.16.1.5              Se0.1        12 1d06h      29     2280    0    23
canada_1#

```

结果中比较重要的信息是经过 172.16.1.5 到 172.16.5.0/24 的路由，经过 172.16.2.6 到 172.16.6.0/24 的路由以及经过 172.16.3.3 到 182.16.3.0/24 和 182.16.4.0/24 的路由。由于 EIGRP 的自动汇总，182.16.3.0/24 和 182.16.4.0 在通过 canada_2 的令牌环接口发送时，会自动汇总为 16 位边界。命令 **show ip eigrp neighbors** 证实路由器 canada_1 和另外两台路由器之间已经建立了 EIGRP 邻接关系。

要允许 EIGRP 到 OSPF 域的连接，必须在路由器 usa_1 上对 EIGRP 和 OSPF 进行相互重分布，这是 EIGRP 和 OSPF 惟一的重新分布点，因此，无需考虑“路由反馈”或重分布环路的问题。例 11-17 给出了路由器 usa_1 的配置示例。

例 11-17 usa_1 的重分布配置示例

```

!
router eigrp 2001
 redistribute ospf 69
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
router ospf 69
 redistribute eigrp 2001 subnets
 network 172.16.6.6 0.0.0.0 area 0
 default-metric 100
!

```

现在，OSPF 路由 192.168.3.0/24 和 192.168.4.0/24 对于路由器 canada_1 来说成为外部 EIGRP 路由。同样，所有的 EIGRP 路由对于路由器 usa_2 来说成为 OSPF 外部类型 2 路由。

EIGRP 和 IGRP 之间也必须在路由器 mexico_1 上进行手动路由双向重分布。如果 IGRP 路由域与 EIGRP 是在同一自治域系统中，重分布就没有必要，因为这一过程会自动完成。例 11-18 是路由器 mexico_1 的配置示例。

例 11-18 mexico_1 上的重分布配置示例

```

!
router eigrp 2001
 redistribute igrp 2000
 passive-interface Ethernet0

```

（待续）

```

network 172.16.0.0
default-metric 1544 100 254 1 1500
!
router igrp 2000
redistribute eigrp 2001
passive-interface Serial0.1
network 172.16.0.0
default-metric 1544 100 254 1 1500
!

```

现在路由器 `mexico_2` 的路由表显示出该模型中所有网络的路由。例 11-19 给出了路由器 `mexico_2` 的路由表的情况。

例 11-19 重分布后 `mexico_2` 的路由表

```

mexico_2#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 5 subnets
C       172.16.5.0 is directly connected, Ethernet0/0
I       172.16.6.0 [100/10676] via 172.16.5.5, 00:00:16, Ethernet0/0
I       172.16.1.0 [100/8576] via 172.16.5.5, 00:00:16, Ethernet0/0
I       172.16.2.0 [100/10576] via 172.16.5.5, 00:00:16, Ethernet0/0
I       172.16.3.0 [100/8639] via 172.16.5.5, 00:00:16, Ethernet0/0
I       192.168.4.0/24 [100/10676] via 172.16.5.5, 00:00:16, Ethernet0/0
I       182.16.0.0/16 [100/9139] via 172.16.5.5, 00:00:16, Ethernet0/0
I       192.168.3.0/24 [100/10676] via 172.16.5.5, 00:00:16, Ethernet0/0
mexico_2#

```

这个模型中的路由重分布相对比较简单，因为模型中所有的网络或者是在 24 位界上或者是以 24 位界进行自动汇总。EIGRP 在发送数据包或重分布时自动汇总到主网边界上。在重分布进 IGRP 的过程中，由于和 192.168.3.0/24 同处于 24 位界，EIGRP 对网络 192.168.4.0/24 进行了自动汇总。182.16.0.0 网络经过汇总通过 s0.1 和 s0.2 接口发送。

有一点很重要，如果宣告接口的 IP 地址和宣告路由处在同一主网边界之内，那么不会进行自动汇总。例如，如果路由器通过 IP 地址为 172.16.10.1/24 的接口宣告路由 172.16.100.0/30，EIGRP 就不会将该路由汇总到它的主网边界去。而如果同样的网络 172.16.100.0/24，要通过地址为 172.17.10.1/24 的接口宣告，EIGRP 汇总路由为 172.16.0.0/16，上面的模型中可以看到这一点。

在下面的 EIGRP 的汇总部分可以看到，对 IP 地址的结构作小小改动，就一定要在重分布正常工作之前进行手动汇总。

11.6.2 实例：EIGRP 路由控制的应用

的应用看看路由管理的问题。在路由器 `usa_1` 上运用过滤列表阻止 EIGRP 将路由 192.168.3.0/24 传播到整个 EIGRP 域中。要达到这一目的，使用命令 `distribute-list` 结合访问列表拒绝 192.168.3.0/24 同时允许其他路由条目的发送。例 11-20 给出了路由器 `usa_1` 的相关配置示例，这里允许发送除了路由 192.168.3.0/24 外的所有路由条目。

例 11-20 分布列表的应用

```
router eigrp 2001
 redistribute ospf 69
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
 distribute-list 10 out Serial0/1 ← Apply access list 10 to interface S0/1
!
router ospf 69
 redistribute eigrp 2001 subnets
 network 172.16.6.0 0.0.0.0 area 0
 default-metric 100
!
ip classless
 access-list 10 deny 192.168.3.0 0.0.0.255 ← deny route 192.168.3.0/24
 access-list 10 permit any ← allow all other routes to pass
!
```

要对从一个路由选择协议到另一个的路由更新信息进行控制、管理，就要用到路由映射。这个模型里用了路由图来禁止 OSPF 路由 172.16.6.0/24 从 EIGRP 到 IGRP 的重分布。路由图在 IGRP 中是用 `redistribution` 命令来调用是，然后路由映射又调用访问列表，并禁止与访问列表相匹配路由的发送。例 11-21 给出了路由器 `mexico_1` 的配置示例，它运用了路由映射来过滤掉路由 172.16.6.0/24。

例 11-21 在 `mexico_1` 上的重分布过程中调用路由映射

```
router eigrp 2001
 redistribute igrp 2000
 passive-interface Ethernet0
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
router igrp 2000
 redistribute eigrp 2001 route-map noospf ← call route map named noospf
 passive-interface Serial0/1
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
ip classless
!
 access-list 11 deny 172.16.6.0 0.0.0.255 ← deny 172.16.6.0/24
 access-list 11 permit any
 route-map noospf permit 10
 match ip address 11 ← allow routes that pass access list 11
!
```

现在，路由器 `mexico_2` 上的路由表显示只有一条路由 192.168.4.0 来自 OSPF 域。比较例 11-22 和例 11-19 的结果，查看应用了路由映射和分布列表的效果。

例 11-22 经过路由过滤之后 mexico_2 的路由表

```
mexico_2#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 4 subnets
C       172.16.5.0 is directly connected, Ethernet0/0
I       172.16.1.0 [100/8576] via 172.16.5.5, 00:00:51, Ethernet0/0
I       172.16.2.0 [100/10576] via 172.16.5.5, 00:00:51, Ethernet0/0
I       172.16.3.0 [100/8639] via 172.16.5.5, 00:00:51, Ethernet0/0
I       192.168.4.0/24 [100/10676] via 172.16.5.5, 00:00:51, Ethernet0/0
I       182.16.0.0/16 [100/9139] via 172.16.5.5, 00:00:51, Ethernet0/0
mexico_2#
```

注释 也可以在 `route-map` 命令中使用 `set tag xx`，以利用路由图来设置 EIGRP 标签。标签的设置对于观察路由如何进入路由表很有帮助。标签的查询，在 EIGRP 中可用 `show ip eigrp topology` 命令，在 OSPF 中可用 `show ip ospf database` 命令。OSPF 的标签可以直接在 `redistribution` 命令中输入。

11.7 EIGRP 的汇总

理解 EIGRP 的汇总原理，知道如何有效地运用这一技术对于大型 EIGRP 网络的设计是至关重要的。EIGRP 的网络能很好地扩展，但是当路由数目上升到上百个时，对有关路由的传播和查询数据包的范围的问题就要格外当心。尽管在小型网络中 EIGRP 可以做到即插即用，但是大型网络上却不行。网络规模越大，对于路由的传播方式就越要小心谨慎。

汇总机制为 EIGRP 提供了两个方面的增强和提高。首先，通过减少路由表中的路由数量，使得 EIGRP 的数据包发送数目和大小都大为减少。其次（也是更重要的一点）是能够限制 EIGRP 的查询范围。

11.7.1 通过汇总控制查询范围以及 SIA 路由的问题

大型 EIGRP 网络面临的最常见而又最复杂的问题之一就是陷入活动状态 (SIA)。当 EIGRP “活跃地”不停地计算某个路由时，该路由就陷入了 SIA 状态。EIGRP 会记录很多如下信息：

```
%DUAL-3-SIA: Route 192.168.1.16 Stuck-in-Active
```

多数时候，路由是因为等待从邻居路由器返回的查询信息而处于活动状态的，可归纳以下几个原因：

- 路由器负载过度——边缘路由器可能被查询请求淹没，缺乏足够的 CPU 资源来处理请求。查询得到不规则的应答（如果有的话），使得路由一直处于活动状态。
- 路由器的内存原因——这种情况可能被低速处理器和路由器的大量等待队列所恶化。
- 电路使用过度——EIGRP hello 信号可能无法正常通过链路，导致丢失邻居路由器的邻接关系。

有两种配置方式很可能导致 SIA 状况的出现：

- 多数 EIGRP 网络都关闭了自动汇总功能，这主要是因为 IP 寻址方案中含有不连续的子网段，从而无法限制查询范围。
- 在大型帧中继网络中，很多远程节点接入同一台路由器，使得本地路由器产生大量的邻居路由器。

例如，在帧中继网络中，如果一个 PVC 进休眠状态，或者路由来回抖动，就会产生一次小的 EIGRP 查询风暴（query storm）。图 11-7 和 11-8 给出了常见的帧中继网络及其查询过程。

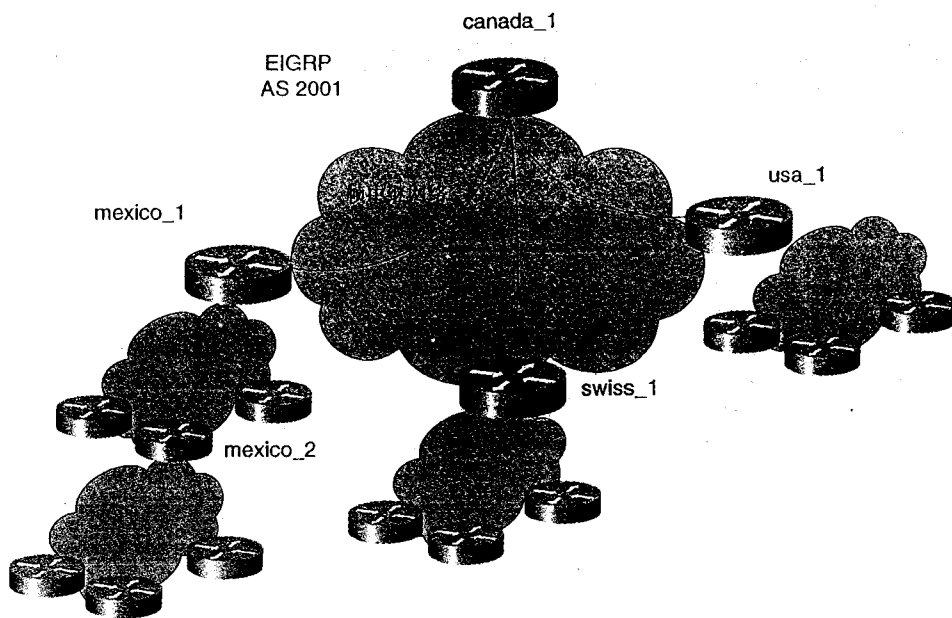


图 11-7 EIGRP 网络

如果 canada_1 到 mexico_1 的 PVC 丢失，canada_1 路由器会向所有的邻居路由器发出关于丢失从 mexico_1 来的路由的 EIGRP 查询信息。它试图为此路由寻找一个新的可行备选者。在这个例子里，这条信息发送到 swiss_1 和 usa_1 路由器。而 mexico_1 路由器同样也要向它所有的邻居路由器发出一条信息，为它丢失的路由寻找一个新的可行备选者。EIGRP 域中的所有的路由器继续向他们的邻居路由器发出查询信息。路由在 EIGRP 收到对它发出的查询的“应答”之前就保持在“活动”状态中。如要扩展该网络，增加一台具有 HSSI 或 T3 接口的路由器，就有可能造成在一个接口上存在着上百个 PVC 的情况，而一个

PVC 的丢失就可能产生成百上千的查询信息。幸好，EIGRP 的汇总机制对查询处理作了限制，这是控制查询风暴问题最有效的方法之一。一个大型 EIGRP 网络如果不具备汇总功能，就会导致 SIA 问题。

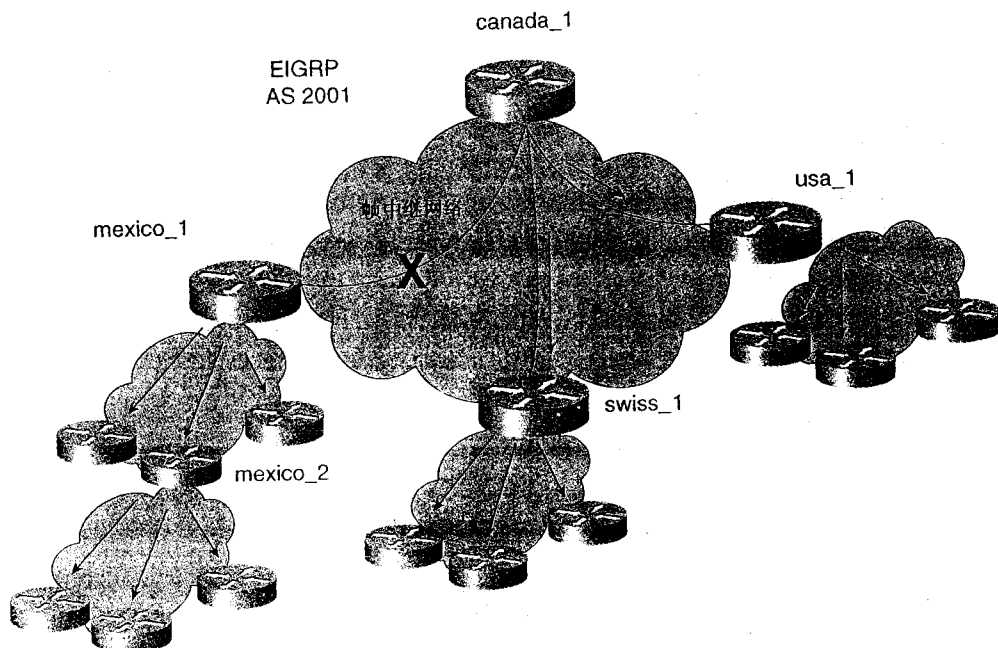


图 11-8 EIGRP 的查询风暴

11.7.2 EIGRP 的自动汇总功能

EIGRP 会默认在两种情况下进行自动汇总：

- 将 EIGRP 重分布进某个有类路由选择协议，如 IGRP 或 RIP 时，在主网边界上会进行自动汇总处理。这种类型的汇总是无法关闭的。
- 当路由正在通过向与路由本身不处于同一主网边界之内的某个接口进行发送时，在主网边界上会进行自动汇总处理。这类汇总可通过 `router (config-router)` 提示符下的 `no auto-summary` 命令来关闭。

EIGRP 不会对 EIGRP 外部路由自动汇总。

汇总后的 EIGRP 路由的管理距离是 90。图 11-9 修改了上面的网络模型，在路由器 `canada_2` 中加入两个网络。

允许了自动汇总功能后，路由 `canada_2` 上的 EIGRP 会向 `canada_1` 发送两份汇总路由。路由 `182.16.3.0/24` 和 `182.16.4.0/24` 汇总为 `182.16.0.0/16`。路由 `10.1.1.0/24` 和 `10.1.2.0/30` 则是汇总到它们的主网边界上的路由 `10.0.0.0/8`。有一点很重要，EIGRP 只有在通过处在不同类边界中的接口发送路由时才对路由进行汇总。例如，如果 `canada_1` 和 `canada_2` 之间的网络是 `10.1.3.0/24`，这个 10 开头的网络不会进行汇总，只有 `182.16.x.x` 的网络才进行汇总。例 11-23 列出了 `canada_1` 的路由表，突出了经过汇总的路由。

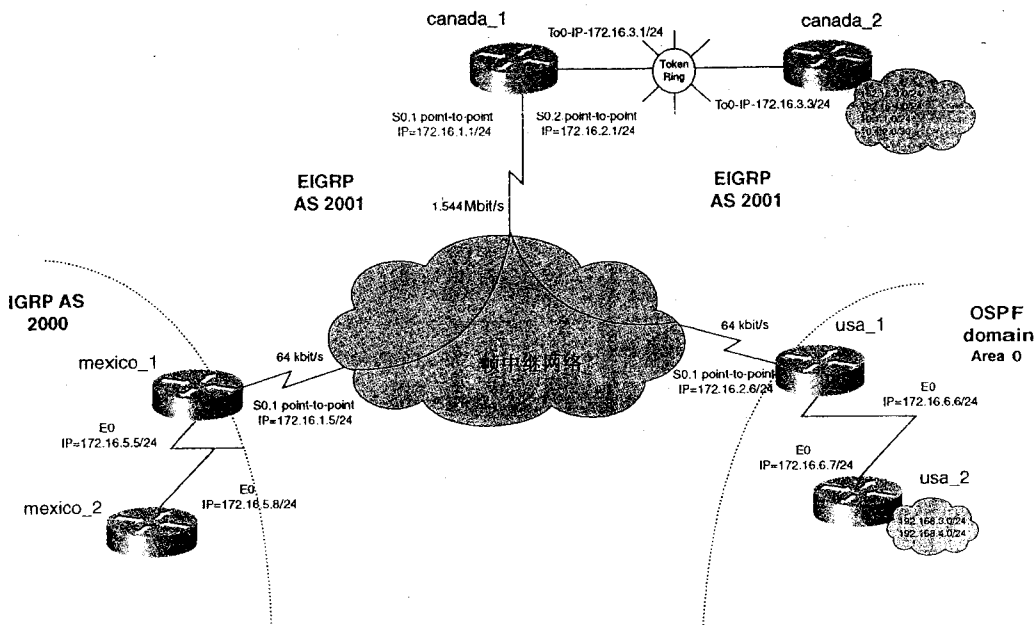


图 11-9 EIGRP 的自动汇总

例 11-23 canada_1 路由表中的汇总路由

```
canada_1#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 5 subnets
D      172.16.5.0 [90/2195456] via 172.16.1.5, 01:06:41, Serial0.1
D      172.16.6.0 [90/2195456] via 172.16.2.6, 01:08:01, Serial0.2
C      172.16.1.0 is directly connected, Serial0.1
C      172.16.2.0 is directly connected, Serial0.2
C      172.16.3.0 is directly connected, TokenRing0
192.168.4.0/32 is subnetted, 1 subnets
D EX   192.168.4.1 [170/2195456] via 172.16.2.6, 01:07:46, Serial0.2
D      10.0.0.0/8 [90/304128] via 172.16.3.3, 00:51:27, TokenRing0
D      182.16.0.0/16 [90/304128] via 172.16.3.3, 00:51:27, TokenRing0
192.168.3.0/32 is subnetted, 1 subnets
D EX   192.168.3.1 [170/2195456] via 172.16.2.6, 01:07:46, Serial0.2
canada_1#
```

尽管 EIGRP 的自动汇总功能表面上看来非常有用，但是在多数现代网络中，它也有不足之处。它强制性地使得无类路由选择协议在主网边界上具有不连续的子网。EIGRP 形成用于发送的汇总路由时，在该汇总中也含有了对所有不存在网络的空路由。比如，**canada_2** 会为每个与该路由器接口存在连接的主类网络产生 3 个空路由。该路由使得路由器丢掉任何没有

子。

例 11-24 EIGRP 自动汇总的空路由

```

canada_2#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, * - candidate default
        U - per-user static route, o - ODR

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.5.0/24 [90/2211584] via 172.16.3.1, 00:12:02, TokenRing0
D       172.16.6.0/24 [90/2211584] via 172.16.3.1, 00:12:02, TokenRing0
D       172.16.0.0/16 is a summary, 00:12:06, Null0
D       172.16.1.0/24 [90/2185984] via 172.16.3.1, 00:12:02, TokenRing0
D       172.16.2.0/24 [90/2185984] via 172.16.3.1, 00:12:02, TokenRing0
C       172.16.3.0/24 is directly connected, TokenRing0
    192.168.4.0/32 is subnetted, 1 subnets
D EX    192.168.4.1 [170/2211584] via 172.16.3.1, 00:12:02, TokenRing0
    10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       10.1.2.0/30 is directly connected, Loopback32
D       10.0.0.0/8 is a summary, 00:12:06, Null0
C       10.1.1.0/24 is directly connected, Loopback31
    182.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       182.16.4.0/24 is directly connected, Loopback21
C       182.16.3.0/24 is directly connected, Loopback20
D       182.16.0.0/16 is a summary, 00:12:07, Null0
    192.168.3.0/32 is subnetted, 1 subnets
D EX    192.168.3.1 [170/2211584] via 172.16.3.1, 00:12:03, TokenRing0
canada_2#

```

要使自动汇总功能正常工作，一定要避免主网中子网的不连续性。不幸的是在现代网络中，一些原因使得子网部署在了非正确的位置，EIGRP 也存在着数据转发的问题。用 **no auto-summary** 命令关闭自动汇总功能，就不会产生空路由，也不会再转发自动汇总路由。可以采用手动汇总代替自动汇总。大部分网络工程师在使用 EIGRP 时都关闭自动汇总功能，以防止的空路由发送到空接口（null interface）上。

11.7.3 EIGRP 的手动汇总或路由聚合

EIGRP 的手工汇总对大型网络来说很关键，能够限制 EIGRP 的查询，明显减小路由表条目。手动汇总有两种方法：

- 用下面这条接口命令宣告汇总地址或聚合地址：
`ip summary-address eigrp as_number summary_address address_mask`
- 用接口命令宣告默认路由：

```
ip summary-address eigrp as_number 0.0.0.0 0.0.0.0.
```

这条命令使得只宣告默认路由，禁止其他路由更新。

EIGRP 的一个强大的功能就是可以在不同的接口上发送多个汇总路由和默认路由。图

11-10 中的网络处在同一个自治域系统中，所有的路由器的自动汇总功能都被禁止。这个模

型中，EIGRP 配置成通过 canada_1 的 s0.2 接口发送默认路由到 usa_1。路由器 canada_1 还通过其 s0.1 接口发送两个汇总路由 182.0.0.0/8 和 10.0.0.0/8 到 mexico_1。

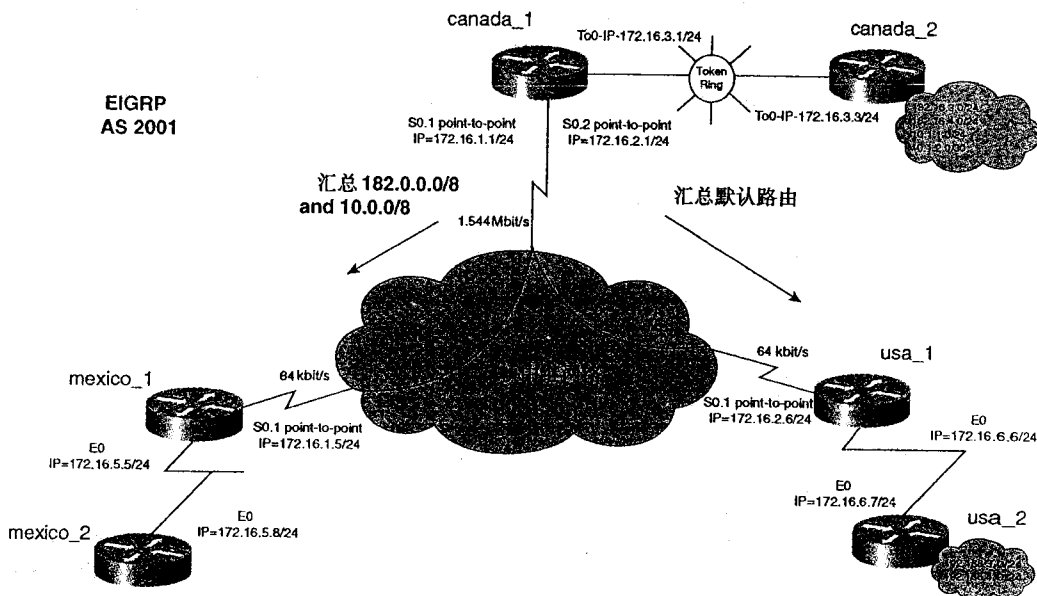


图 11-10 EIGRP 的手动汇总

例 11-25 列出了进行手动汇总时 canada_1 路由器上的配置示例。

例 11-25 在 canada_1 的串行接口上进行手动汇总

```
interface Serial0
no ip address
encapsulation frame-relay
no ip mroute-cache
!
interface Serial0.1 point-to-point
ip address 172.16.1.1 255.255.255.0
ip summary-address eigrp 2001 182.0.0.0 255.0.0.0 ← Manual summarization
ip summary-address eigrp 2001 10.0.0.0 255.0.0.0
frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
ip address 172.16.2.1 255.255.255.0
ip summary-address eigrp 2001 0.0.0.0 0.0.0.0 ← Advertise a default route only
frame-relay interface-dlci 130
!
interface TokenRing0
ip address 172.16.3.1 255.255.255.0
ring-speed 16
!
router eigrp 2001
network 172.16.0.0
no auto-summary
!
```

例 11-26 列出了路由器 mexico_1 和 usa_1 的路由表的情况。注意这里的路由如何汇总。

路由器 `usa_1` 只接收默认路由 `0.0.0.0`，并将其设置为最后手段的网关。

例 11-26 进行了汇总之后 `mexico_1` 和 `usa_1` 的路由表

```
mexico_1#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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

      172.16.0.0/24 is subnetted, 5 subnets
C       172.16.5.0 is directly connected, Ethernet0
D       172.16.6.0 [90/2707456] via 172.16.1.1, 00:34:11, Serial0.1
C       172.16.1.0 is directly connected, Serial0.1
D       172.16.2.0 [90/2681856] via 172.16.1.1, 00:34:11, Serial0.1
D       172.16.3.0 [90/2185984] via 172.16.1.1, 00:34:11, Serial0.1
D      192.168.4.0/24 [90/2835456] via 172.16.1.1, 00:34:11, Serial0.1
D      10.0.0.0/8 [90/2313984] via 172.16.1.1, 00:34:11, Serial0.1 ←Summary Route
D      192.168.3.0/24 [90/2835456] via 172.16.1.1, 00:34:11, Serial0.1
D      182.0.0.0/8 [90/2313984] via 172.16.1.1, 00:34:11, Serial0.1 ←Summary Route

mexico_1#

usa_1#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.2.1 to network 0.0.0.0

D      192.168.3.0/24 [90/409600] via 172.16.6.7, 00:45:52, Ethernet0
D      192.168.4.0/24 [90/409600] via 172.16.6.7, 00:45:52, Ethernet0
      172.16.0.0/24 is subnetted, 2 subnets
C       172.16.6.0 is directly connected, Ethernet0
C       172.16.2.0 is directly connected, Serial0.1
D*    0.0.0.0/0 [90/2185984] via 172.16.2.1, 00:44:54, Serial0.1

usa_1#
```

注释 从 Cisco IOS 12.0 (4) T 开始，可以为汇总地址添加管理距离，以改变默认值为 90 的管理距离。

11.8 EIGRP 的默认路由

有两种方法可以向 EIGRP 注入默认路由：

- 将默认的静态路由重分布进 EIGRP，以使 EIGRP 将 `0.0.0.0` 看成是默认路由。用全局命令 `ip route 0.0.0.0 0.0.0.0 next_hop_IP_address` 创建默认静态路由，然后用

redistribute static 命令将该默认路由重分布到 EIGRP。如果不使用网络 0.0.0.0，还可以用 **ip default-network a.b.c.d** 命令将此路由标记为默认的路由。

- 用接口命令 **ip summaryaddress eigrp as_number 0.0.0.0 0.0.0.0** 对默认路由由 0.0.0.0 进行汇总。上节中的例子已经解释了如何用这条命令来转发默认路由。

这两种方法的使用都需要首先执行全局命令 **ip classless**，这样，对于那些没有明确路由的数据包，路由器会将其转发到默认路由。Cisco 12.0 以及以上版本的 IOS 中都是默认激活 **ip classless**。

图 11-11 中，canada_1 发送默认路由到 usa_1 和 mexico_1，canada_1 通过创建指向下一跳地址 172.16.3.3 或路由器 canada_2 的静态路由来完成这一过程。

例 11-27 列出了 canada_1 宣告默认静态路由的配置示例。

例 11-27 在 canada_1 上宣告 EIGRP 的默认静态路由

```

router eigrp 2001
  redistribute static      !-redistribute the static routes
  network 172.16.0.0
  default-metric 16000 630 254 1 1500    !-Don't forget the default metric
  no auto-summary
!
ip classless              !-IP classless must be enabled for default routing
ip route 0.0.0.0 0.0.0.0 172.16.3.3    !-the default route points at Canada_1 router
!
    
```

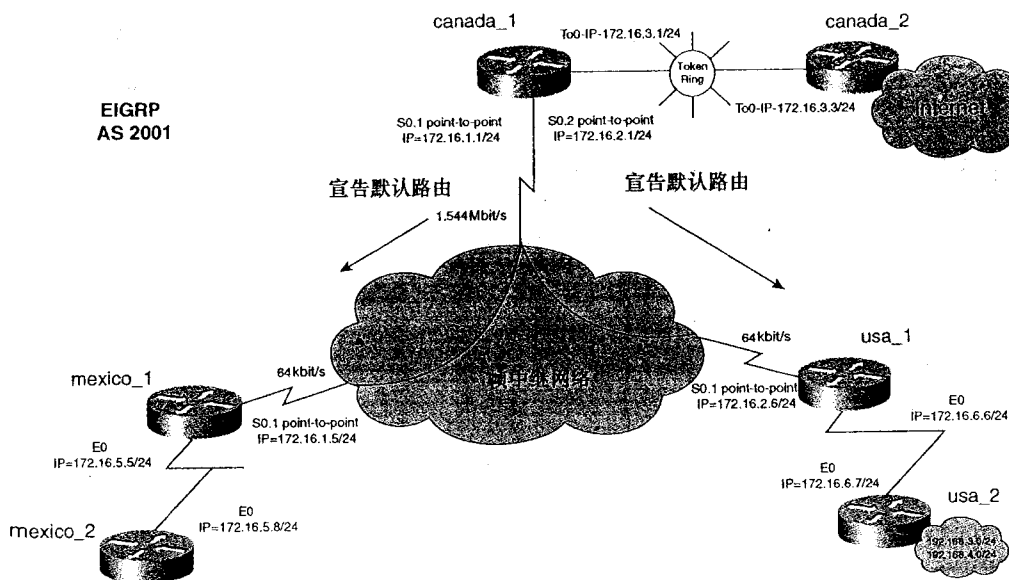


图 11-11 EIGRP 的默认路由

例 11-28 列出了 mexico_1 的路由表，演示了接收默认路由的过程。从例子中可以看出，接收到路由时，由于该路由经过了重分布，成为外部路由，“*” 标记表明是默认路由。网络中还设置了最后手段的网关。

例 11-28 mexico_1 的路由表

```
mexico_1#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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is 172.16.1.1 to network 0.0.0.0

172.16.0.0/24 is subnetted, 5 subnets
C      172.16.5.0 is directly connected, Ethernet0
D      172.16.6.0 [90/2707456] via 172.16.1.1, 00:04:15, Serial0.1
C      172.16.1.0 is directly connected, Serial0.1
D      172.16.2.0 [90/2681856] via 172.16.1.1, 00:04:51, Serial0.1
D      172.16.3.0 [90/2185984] via 172.16.1.1, 00:04:51, Serial0.1
D      192.168.4.0/24 [90/2835456] via 172.16.1.1, 00:04:15, Serial0.1
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D      10.1.2.0/30 [90/2313984] via 172.16.1.1, 00:04:51, Serial0.1
D      10.1.1.0/24 [90/2313984] via 172.16.1.1, 00:04:52, Serial0.1
182.16.0.0/24 is subnetted, 2 subnets
D      182.16.4.0 [90/2313984] via 172.16.1.1, 00:04:52, Serial0.1
D      182.16.3.0 [90/2313984] via 172.16.1.1, 00:04:52, Serial0.1
D      192.168.3.0/24 [90/2835456] via 172.16.1.1, 00:04:15, Serial0.1
D*EX 0.0.0.0/0 [170/2331136] via 172.16.1.1, 00:00:53, Serial0.1
mexico_1#
```

11.9 EIGRP 的存根路由

Cisco IOS 12.0 (7) T 中, Cisco 引入 EIGRP 存根路由, 以进一步控制网络的稳定性, 以及提高资源的利用率, 这一功能也完全集成到了 Cisco IOS 12.0 (15) S 中。EIGRP 存根路由的工作形式和 OSPF 存根区域非常相似。存根路由器只有一条和路由域连接的路径, 所有的数据都转发到中央路由器或分布路由器上去。换一种说法, 就是存根网络不能作为 EIGRP 的传输路由器, 它只能有一个 EIGRP 邻居路由器。

对 EIGRP 存根路由进行配置时, 只有一台远程或宣告 (spoke) 路由器需要配置成存根路由器。该路由器用“无法到达”的信息对汇总查询、直连路由、重分布的静态路由、外部路由以及内部路由做出响应。这一过程大大降低了响应远端路由器的查询所需的系统开销。存根路由器还能发送一条特殊的对等信息到它的邻居路由器, 以通知这些邻居路由器, 它是一台 Stub 路由器。

EIGRP 下用下面这条命令可以配置存根路由:

```
Router (config-router) #eigrp stub [receive-only | connected | static | summary]
```

命令中的选项为:

- **receive-only**——使路由器不发送任何路由。
- **connected**——路由器向邻居路由器宣告所有直连路由, 不需要进行重分布。
- **static**——路由器向邻居路由器宣告所有静态路由。静态路由还需要重分布进要宣告的 EIGRP 中。

- **summary**——路由器宣告汇总路由。

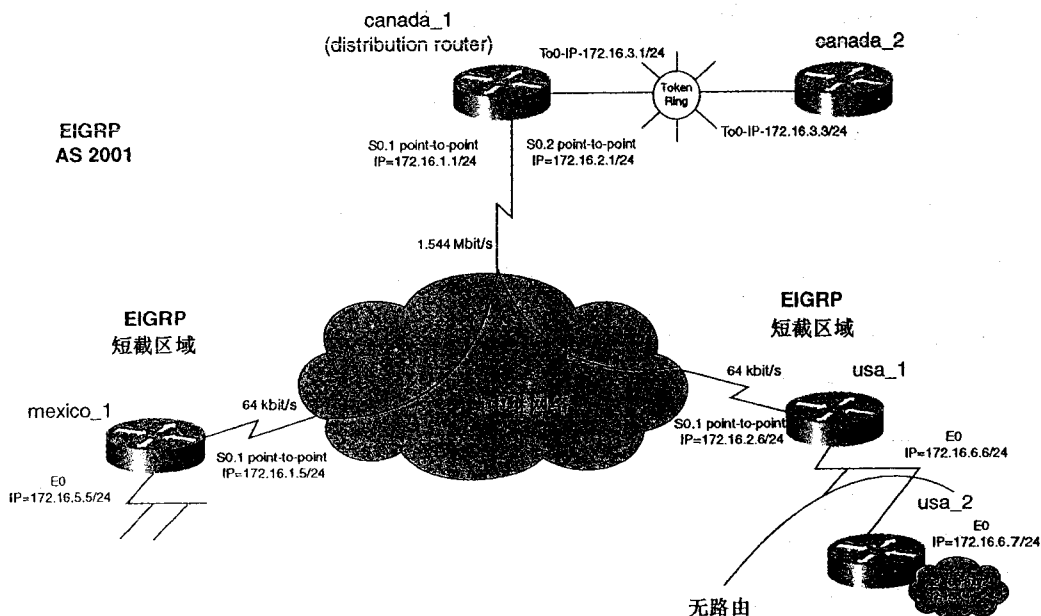


图 11-12 EIGRP 的 stub 路由

Stub 路由器可以配置成同时发送直连路由和静态路由，这是多数存根域采用的方式。图 11-12 给出了两个配置好的 EIGRP 存根网络的情况。路由器 `mexico_1` 配置成存根路由器，仅仅宣告本地直连的以太网信息。而 `usa_1` 则在宣告本地以太网信息的同时，还会宣告两条静态路由，网络 `192.168.3.0/24` 和 `192.168.4.0/24`。路由器 `usa_2` 有一个指向 `172.16.6.6` 的默认网关，但是没有启动任何路由。分布路由器是 `canada_1`，不需要作任何额外的 EIGRP 配置。

例 11-29 给出了路由器 `mexico_1` 和 `usa_1` 的 EIGRP 配置。

例 11-29 EIGRP 的存根配置示例

```
!
hostname mexico_1
!
router eigrp 2001
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
 no auto-summary
 eigrp stub connected      ←Set EIGRP stub, and advertise connected routes
!

!
hostname usa_1
!
router eigrp 2001
 redistribute static      ←Redistribute static
```

(待续)


```

network 172.16.0.0
default-metric 1544 100 254 1 1500
no auto-summary
eigrp stub connected static  ←Set EIGRP stub, and advertise connected and static
                             routes
!
ip classless
ip route 192.168.3.0 255.255.255.0 172.16.6.7
ip route 192.168.4.0 255.255.255.0 172.16.6.7

```

最后，通过观察 canada_1 的路由表，所有的路由都显示正常，如例 11-30 所示。

例 11-30 具有两个 EIGRP 存根域的路由器 canada_1 路由表

```

canada_1#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 5 subnets
D       172.16.5.0 [90/2195456] via 172.16.1.5, 01:03:47, Serial0.1
D       172.16.6.0 [90/2195456] via 172.16.2.6, 00:48:40, Serial0.2
C       172.16.1.0 is directly connected, Serial0.1
C       172.16.2.0 is directly connected, Serial0.2
C       172.16.3.0 is directly connected, TokenRing0
D EX 192.168.4.0/24 [170/2195456] via 172.16.2.6, 00:43:27, Serial0.2
       10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.2.0/30 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
D       10.1.1.0/24 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
       182.16.0.0/24 is subnetted, 2 subnets
D       182.16.4.0 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
D       182.16.3.0 [90/304128] via 172.16.3.3, 07:31:49, TokenRing0
D EX 192.168.3.0/24 [176/2195456] via 172.16.2.6, 00:43:28, Serial0.2
canada_1#

```

用 **show ip eigrp neighbor detail** 命令可以验证路由器是否已经配置成 EIGRP 存根路由器。该命令执行结果的最后一行显示存根路由是否已经打开，以及存根路由器发送的内容。

例 11-41 中可以看到这条命令的执行示例。命令 **show eigrp packet stub** 则显示对等路由器的存根状态调试信息。

11.10 EIGRP 的等价路由开销和非等价

路由开销的负载平衡

默认情况下，EIGRP 支持在 4 条等价路由开销的路径上进行负载共享。要进行负载共享，用于共享负载的路由必须出现在 IP 转发表或 **show ip route** 命令的输出结果中。仅当转发表中含有具备多条路径的路由时才有可能进行负载的共享。在串行接口上用 **bandwidth** 命令来

确保网络的 EIGRP 度量的一致性。这也有助于使路由出现在 IP 转发表中。

EIGRP 也可以在非等价路由开销的路由上进行负载平衡，和 IGRP 一样。在选择具有最大度量值的路径的上限时也采用 **variance** 作为乘数。

配置 EIGRP 的非等价路由开销负载平衡可以分为下面 3 个步骤：

第 1 步 在要进行负载共享的接口两端配置带宽，用 **bandwidth xx kbit/s** 命令完成。

第 2 步 定义最低路由开销和最高路由开销的度量。通过这些值计算 **variance** 乘数并将它加入到 EIGRP 的路由过程。和前面讨论的一样，用 **show ip eigrp topology** 命令查看 EIGRP 使用的组合度量的情况。

第 3 步 （可选）设置最大路径或数据共享等变量。

下面是一个假设的 **variance** 的计算过程。EIGRP 有一个度量为 100 的路由，路由器还有通向同一目的地址的两条路由，度量分别为 200 和 300。要在这 3 条路径上进行数据的共享，我们可以把 **variance** 设为 3：

$$3 \times 100 = 300$$

换个说法， $\text{variance} \times \text{最低度量} = \text{共享负载路径中的最高度量}$ ，这里是 300。用下面的公式可以在实际网络中正确设置 **variance** 的值：

$$\text{variance} = 1 + \lceil [\text{metric of highest cost route} / \text{metric of the lowest cost route}] \rceil$$
，向上舍入到整数位)

用 **show ip eigrp topology** 命令查询最低路由开销和最高路由开销的路由度量的值。记住一定要在链路两端同时更改 **variance** 以及其他变量，如带宽等。所有的串行链路上都必须对带宽进行设置。下面是一些用于负载平衡的命令：

```
Router (config-router) #variance { metric_multiplier 1-128}
Router (config-router) #maximum-paths [ 1-6]
Router (config-router) #traffic-share {balanced | min across-interface}
Router (config-if) #bandwidth xx kbit/s
```

命令 **variance** 用于定义在非等价路由开销的负载平衡中路由度量的乘数值。默认的 **variance** 值是 1，实际上就是等价路由开销的负载平衡。

路由器利用 **maximum-paths** 命令可以定义最多 6 条路径来进行数据共享。用这条命令可以限制参与负载共享的路由数量。一组在路由中显示为一跳的指向同一目的地址的多条路径通道称为一个**负载共享组**。参与负载共享的路径数目默认是 4 条。

如果有多条最低路由开销的路径，EIGRP 通过 **traffic-share- min** 命令的配置进行等价路由开销的负载平衡。默认情况下，这条命令设置成 **balanced**，这样数据在路由间的分配会与路由的度量值成比例。

例如，如果 **variance** 设为 3，数据共享设置成 **balanced**，最佳路由传输的数据将是最差路由的 3 倍。

一条路由要包括在非等价路由开销的负载平衡中，必须还要满足 3 个条件：

- 将这条路由由加入到负载共享组中不会超出最大路径数的限制。
- 下游路由器必须在它的度量上更加接近目的地址一些。
- 最低路由开销的路由度量在乘 **variance** 后必须比要加入的路由度量更大。

第 10 章曾给出一个在 IGRP 上进行负载共享的详细实例，EIGRP 的做法和它完全一样。

11.11 实验 22：配置 EIGRP：路由重分布、 汇总以及存根路由——第 1 部分

11.11.1 实验说明

随着 EIGRP 网络的逐渐普及和推广，对 EIGRP 协议的应用控制和与其他路由选择协议，如 RIP 和 OSPF 的集成与融合也就显得越来越重要，同时，理解 EIGRP 的默认设置，如水平分隔、自动汇总等的重要性也就更迫切。

这个实验的内容包括了应用控制，EIGRP 与其他路由选择协议的集成以及自动汇总的使用等。

11.11.2 实验内容

Cisco 培训伙伴（Cisco Training Partners）在全美范围内提供客户化的 Cisco 培训教程，现在正致力于将其所有的培训点集成到一个公众网络中去。你的任务是按照下面要求配置一个 EIGRP 网络：

- 按照图 11-13 配置一个 IP 网络，路由选择协议采用 EIGRP，自治系统（AS）ID 为 65001。
- 将路由器 wisconsin, georgia 和 ohio 之间的帧中继网络配置为多点网络，而路由器 wisconsin 和 minnesota 之间的帧中继网络则配置为点对点网络。
- 确保 RIP 域的完全 IP 互通性，并且不使用静态路由或者发布默认路由。
- （可选）将路由器 georgia 和 ohio 配置为 EIGRP 存根路由器，它们要能够宣告本地 LAN 信息到 EIGRP 中。

11.11.3 实验目的

- 按照图 11-13 配置 Cisco 培训伙伴的网络以及相应的 IP 地址。LAN 的拓扑类型在这个实验中没有影响。
- 在 WAN 上使用帧中继数据链路协议。
- 在 RIP 和 EIGRP 之间进行重分布。
- 确保所有接口的完全 IP 连通，也就是说，要确保在 RIP 域中可以 ping 通所有的帧中继以及 LAN 接口。同时，还要保证路由器 georgia 和 ohio 能够相互 ping 通对方的帧中继和 LAN 接口。此外，在网络上不能配置任何静态路由或者是默认路由。
- （可选）将 georgia 和 ohio 配置为 EIGRP 存根路由器，此时，需要 Cisco IOS 12.0 (7) T 或者是 Cisco IOS 12.0 (15) S 或 12.1 以及更新版本的 Cisco IOS，而且必须是 T 和 S 系列的。

11.11.4 所需设备

- 6 台 Cisco 路由器，其中 4 台要通过 35 背对背电缆或者是类似的方式与帧中继交换机连接在一起。
- 通过集线器或交换机构建的 4 个 LAN 网段。这个实验中对 LAN 的拓扑类型没有要求。

11.11.5 物理设计与实验准备

- 按照图 11-13 将集线器以及串行线路由器连接起来。
- 路由器 stillwater 只在网络 172.16.0.0 上运行 RIPv2。因此在配置工作的时候把这台路由器的配置参考第 11.11.1 节的内容。
- 此外，还需要具有 3 条 PVC 的一台帧中继交换机。例 11-31 是该实验所需要的帧中继配置的示例。

例 11-31 配置帧中继交换机

```

hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148000
frame-relay intf-type dce
frame-relay route 111 interface Serial1
frame-relay route 121 interface Serial3
frame-relay route 150 interface Serial5
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay intf-type dce
frame-relay route 110 interface Serial0
!
<<<text omitted>>>
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 102 interface Serial0
!
<<<text omitted>>>
!

```

```

interface Serial15
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 151 interface Serial0 150
!

```

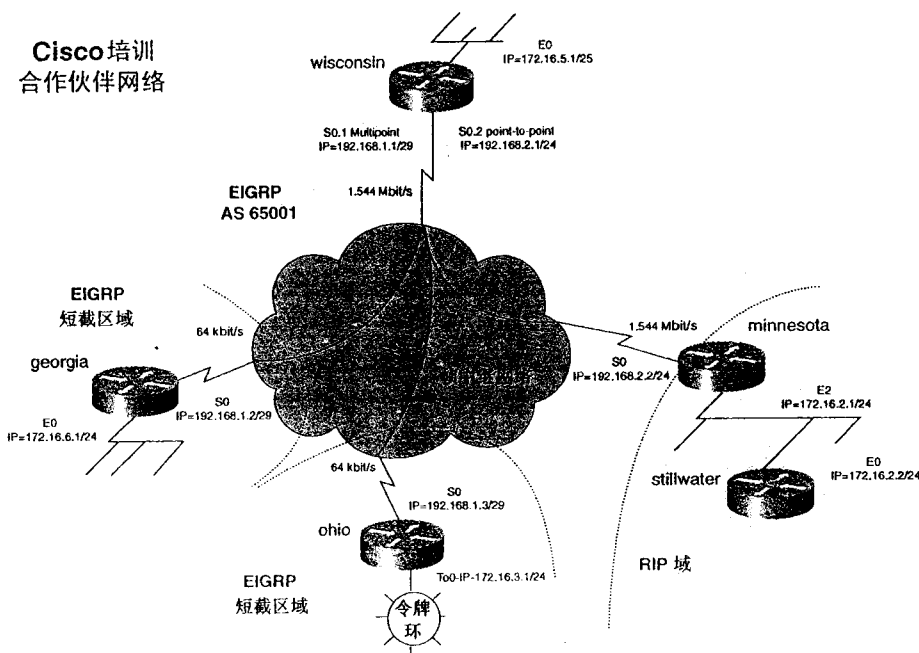


图 11-13 Cisco 培训伙伴网络

11.12 实验 22：配置 EIGRP：路由重分布、 汇总以及存根路由——第 2 部分

11.12.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 4 台路由器以背对背方式连接在一起。然后按照图 11-13 利用交换机或集线器/MAU 创建 4 个 LAN 网段。

物理连接完成之后，再按照图 11-13 为所有的 LAN 和 WAN 接口分配 IP 地址。在继续下一步操作之前，一定要对每一台路由器的本地 LAN 和 WAN 接口都用 **ping** 命令进行测试。路由器 wisconsin 需要使用子接口，一个是多点接口，另一个是点对点接口。多点子接口上需要用 **framerelay map** 命令进行配置，而 wisconsin 和 minnesota 之间的点对点接口上则需要一条 **frame-relay interface-dlci** 命令。

为了建立完整的 IP 连接，ohio 和 georgia 路由器上还需要用 **frame-relay map** 来相互指向对方，建立映射。例 11-32 是到目前为止涉及到的所有相关路由器的配置清单。

例 11-32 路由器 wisconsin、georgia、ohio 和 minnesota 的帧中继配置

```

!
hostname wisconsin
!
<<<text omitted>>>
!
interface Serial0
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
ip address 192.168.1.1 255.255.255.248
no ip directed-broadcast
frame-relay map ip 192.168.1.2 121 broadcast
frame-relay map ip 192.168.1.3 150 broadcast
!
interface Serial0.2 point-to-point
ip address 192.168.2.1 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 111
!
-----
hostname georgia
!
<<<text omitted>>>
!
interface Serial0
ip address 192.168.1.2 255.255.255.248
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay map ip 192.168.1.1 102 broadcast
frame-relay map ip 192.168.1.3 102 broadcast
frame-relay lmi-type cisco
!
-----
hostname ohio
!
enable password cisco
!
<<<text omitted>>>
!
interface Serial0
ip address 192.168.1.3 255.255.255.248
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay map ip 192.168.1.1 151 broadcast
frame-relay map ip 192.168.1.2 151 broadcast
frame-relay lmi-type cisco
!
-----
hostname minnesota
!
<<<text omitted>>>
!
interface Serial0

```

```
ip address 192.168.2.2 255.255.255.0
encapsulation frame-relay
no ip mroute-cache
frame-relay interface-dlci 110
```

本地 WAN 和 LAN 连接建立起来之后的网络配置工作分为两个部分。首先是 EIGRP 域的配置，然后是与 RIP 的集成。

所有路由器上的 EIGRP 基本配置都类似。按照本章中讨论过的配置 EIGRP 的 3 个步骤，首先是用 **router eigrp 65001** 命令在所有的路由器上启动 EIGRP。然后定义要运行 EIGRP 的网络。路由器 wisconsin、georgia 和 ohio 要在主网 172.16.0.0 和 192.168.1.0 上运行 EIGRP。因此，将这些网络作为 **network** 命令的参数加以定义。此外，路由器 wisconsin 和 minnesota 则是要在 192.168.2.0 和 172.16.0.0 上运行 EIGRP。由于这是帧中继网络，因此最好设置带宽参数，在 wisconsin 的接口 s0.1 上将 **bandwidth** 设置为 128 kbit/s (包含两条 64-kbit/s 的 PVC)。路由器 georgia 和 ohio 则是在它们的帧中继接口上将 **bandwidth** 设为了 64 kbit/s。默认的带宽是 1.544 Mbit/s (T1 速率)，因而在 wisconsin 路由器的 S0.2 接口上没有必要再加以改动。例 11-33 是到目前为止 wisconsin 路由器的配置示例。

例 11-33 路由器 wisconsin 的配置示例

```
hostname wisconsin
!
interface Ethernet0
 ip address 172.16.5.1 255.255.255.128
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 bandwidth 128
 ip address 192.168.1.1 255.255.255.248
 no ip directed-broadcast
 frame-relay map ip 192.168.1.2 121 broadcast
 frame-relay map ip 192.168.1.3 150 broadcast
!
interface Serial0.2 point-to-point
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 111
!
router eigrp 65001
 network 172.16.0.0
 network 192.168.1.0
 network 192.168.2.0
!
```

粗略地看，路由功能似乎已正常工作。毕竟路由器 wisconsin 已经有了 1 条路由和 3 台

EIGRP 邻居路由器了。但是只有关闭 EIGRP 的一些默认设置才能使这个网络正常工作。例

11-34 中，EIGRP 在路由表中是已经有 3 台邻居路由器了，但不幸的是路由器同时也在它的转发表中将某些路由置为无效。

例 11-34 路由器 wisconsin 上 show ip route 和 show ip eigrp neighbors 命令的执行结果

```
wisconsin#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

    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.5.0/25 is directly connected, Ethernet0
D       172.16.0.0/16 is a summary, 00:10:58, Null0
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
D       192.168.1.0/24 is a summary, 00:11:33, Null0
C       192.168.1.0/29 is directly connected, Serial0.1
C       192.168.2.0/24 is directly connected, Serial0.2
wisconsin#
wisconsin#show ip eigrp neighbors
IP-EIGRP neighbors for process 65001
H   Address                Interface    Hold Uptime    SRTT    RTO  Q  Seq Type
                               (sec)          (ms)          Cnt Num
2   192.168.1.2             Se0.1        171 00:14:00    768    4608  0  4
1   192.168.1.3             Se0.1        152 00:14:11   1544    5000  0  4
0   192.168.2.2             Se0.2        157 00:14:22     0    3000  0  11
wisconsin#
```

如果 ping 一下 172.16.0.0 域中的任何一台路由器，结果都是失败的。这是因为路由器把这些数据包转发到它定义的空接口去了。

这里有两个问题需要解决：

- 网络的主边界网络掩码位范围内含有不连续的子网。主网 172.16.0.0/16 分成了网络 172.16.2.0/24 和 172.16.3.0/24 等子网。要解决这一个问题，在网络中所有的路由器上用 EIGRP 路由器命令 **no auto-summary** 关闭 EIGRP 自动汇总功即可。
- 尽管关闭了自动汇总功能，水平分隔的问题仍对路由有影响，要解决这个问题，需要关闭水平分隔功能。

整个网络中的自动汇总功能都关闭以后，路由器 wisconsin 的路由表就成了例 11-35 中所示的那样。可以看到路由 172.16.2.0/24、172.16.3.0/24 和 172.16.5.0/24 现在都出现在了转发表中。

例 11-35 路由器 wisconsin 上 show ip route 命令的执行结果

```
wisconsin#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
```

（待续）


```

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

```

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C    172.16.5.0/25 is directly connected, Ethernet0
D    172.16.6.0/24 [90/20537600] via 192.168.1.2, 00:01:47, Serial0.1
D    172.16.2.0/24 [90/40537600] via 192.168.2.2, 00:54:38, Serial0.2
D    172.16.3.0/24 [90/20528128] via 192.168.1.3, 00:08:32, Serial0.1
192.168.1.0/29 is subnetted, 1 subnets
C    192.168.1.0 is directly connected, Serial0.1
C    192.168.2.0/24 is directly connected, Serial0.2
wisconsin#

```

刚才提到的，另外一个需要解决的问题是水平分隔的问题。如果严格地从路由器 wisconsin 来测试 IP 连接的情况，一切看上去都是正常的。但是，在检查路由器 georgia 和 ohio 的转发表的时候，发现 georgia 路由器上找不到子网 172.16.3.0/24，而 ohio 上也没有子网 172.16.6.0/24，如例 11-36 所示。

例 11-36 路由器 ohio 和 georgia 上 show ip route 命令的输出结果

```

ohio#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

```

172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D    172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:00:52, Serial0
D    172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:00:52, Serial0
C    172.16.3.0/24 is directly connected, TokenRing0
192.168.1.0/29 is subnetted, 1 subnets
C    192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:00:53, Serial0
ohio#

```

```

georgia#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

```

172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D    172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:01:21, Serial0
C    172.16.6.0/24 is directly connected, Ethernet0

```

(待续)

```

D    172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:01:21, Serial0
    192.168.1.0/29 is subnetted, 1 subnets
C    192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:01:21, Serial0
georgia#

```

这些路由没有进行传播是由 EIGRP 的水平分隔问题引起的，用 **debug ip eigrp packets** 命令就可以验证这一点。要想使路由更新信息能够在多点网络中正常地传播，需要在路由器 wisconsin 的接口 s0.1 上用 **no ip split-horizon eigrp** 命令关闭该接口上的水平分隔功能：

```
wisconsin (config) #int s0.1
```

```
wisconsin (config-subif) #no ip split-horizon eigrp 65001
```

在路由器 wisconsin 上关闭了水平分隔功能之后，路由器 ohio 和 georgia 上的转发表如例 11-37 所示。

例 11-37 路由器 ohio 和 georgia 上 show ip route 命令示例

```

ohio#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

    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D    172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:00:04, Serial0
D    172.16.6.0/24 [90/41049600] via 192.168.1.1, 00:00:04, Serial0
D    172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:00:04, Serial0
C    172.16.3.0/24 is directly connected, TokenRing0
    192.168.1.0/29 is subnetted, 1 subnets
C    192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:00:04, Serial0
ohio#

georgia#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

    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D    172.16.5.0/25 [90/40537600] via 192.168.1.1, 00:01:41, Serial0
C    172.16.6.0/24 is directly connected, Ethernet0
D    172.16.2.0/24 [90/41049600] via 192.168.1.1, 00:01:41, Serial0
D    172.16.3.0/24 [90/41040128] via 192.168.1.1, 00:00:49, Serial0
    192.168.1.0/29 is subnetted, 1 subnets
C    192.168.1.0 is directly connected, Serial0
D    192.168.2.0/24 [90/41024000] via 192.168.1.1, 00:01:41, Serial0
georgia#

```

现在已经在除了 stillwater 以外所有的路由器上建立了完整的 IP 连接。剩下的这个路由器 stillwater 处在 RIP 域。

要将 RIP 域完整地集成到 EIGRP 中，必须完成两项工作：

- 路由器 minnesota 上 RIP 与 EIGRP 之间的相互重分布。
- 所有 EIGRP 路由都必须汇总到一个 24 位长的网络掩码位范围之内，这是 RIP 网络所在的网络掩码位范围。

用 **redistribution** 命令和 **default-metric** 命令就可以在路由器 minnesota 上启动相互重分布的进行。例 11-38 是在 minnesota 上所做的配置情况。

例 11-38 路由器 minnesota 上 EIGRP 和 RIP 的配置

```
!  
router eigrp 65001  
 redistribute rip  
 network 172.16.0.0  
 network 192.168.2.0  
 default-metric 1544 100 254 1 1500  
 no auto-summary  
!  
router rip  
 redistribute eigrp 65001  
 network 172.16.0.0  
 default-metric 4  
!
```

这样一来，路由器 stillwater 可以接收来自 minnesota 的路由，但它能接收的只是 24 位掩码的路由，不能够接收 wisconsin 路由器上的帧中继多点网络 192.168.1.0/29，或者是以太网网络 182.16.5.0/25 的路由。要想使路由器 stillwater 可以接收这些路由，必须在路由器 wisconsin 和 minnesota 之间的点对点子上配置两个 24 位边界的汇总路由。例 11-39 显示了路由器 wisconsin 上这一配置的情况。

例 11-39 路由器 wisconsin 上的 EIGRP 汇总功能

```
!  
interface Serial0.2 point-to-point  
 bandwidth 64  
 ip address 192.168.2.1 255.255.255.0  
 no ip directed-broadcast  
 ip summary-address eigrp 65001 192.168.1.0 255.255.255.0 5  
 ip summary-address eigrp 65001 172.16.5.0 255.255.255.0 5  
 frame-relay interface-dlci 111  
!
```

例 11-40 给出了路由器 stillwater 的 IP 转发表示例，后面还使用了 **ping** 命令。为了测试整个的 IP 连通情况，在路由器 stillwater 上对所有子网掩码不是 24 位的网络都用 **ping** 进行了测试。

例 11-40 路由器 stillwater 上 show ip route 命令以及紧跟的 ping 命令的执行情况

```

stillwater#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

R    192.168.1.0/24 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
R    192.168.2.0/24 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.5.0 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
R       172.16.6.0 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
C       172.16.2.0 is directly connected, Ethernet0
R       172.16.3.0 [120/4] via 172.16.2.1, 00:00:01, Ethernet0
stillwater#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 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
stillwater#ping 192.168.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
stillwater#ping 172.16.5.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.5.1, timeout is 2 seconds:
!!!!

```

本实验的最后可选部分是把路由器 georgia 和 ohio 配置为 EIGRP 存根路由器。这两台路由器仍必须宣告与之相连的网络路由，因此，配置的时候要使用带有关键字 **connected** 的 **eigrp stub** 命令。需要配置成存根路由器的只是 georgia 和 ohio，wisconsin 上无需任何配置工作。这两台路由器需要的配置命令如下：

georgia (config-router) #eigrp stub connected

如果想要验证存根路由器的工作，可以像例 11-41 那样使用 **show ip eigrp neighbors detail** 命令。输出结果的最后一行显示有没有允许存根路由宣告以及存根路由器宣告的内容。同样也需要从 RIP 域中用 **ping** 对新建立的存根区域进行测试以验证 IP 的连通性。

例 11-41 配置存根路由转发

```

wisconsin#show ip eigrp neighbors detail 65001
IP-EIGRP neighbors for process 65001
H   Address                Interface    Hold Uptime    SRTT   RTO  Q  Seq Type
                               (sec)         (ms)          Cnt  Num
2   192.168.1.3             Se0.1       178 00:00:53   52  1140  0  25

```

(待续)

```

Version 12.0/1.1, Retrans: 1, Retries: 0
Stub Peer Advertising ( CONNECTED ) Routes
1  192.168.1.2          Se0.1          156 00:03:11  209 1254  0  28
Version 12.0/1.1, Retrans: 0, Retries: 0
Stub Peer Advertising ( CONNECTED ) Routes
0  192.168.2.2          Se0.2          130 01:01:01  26  2280  0  33
Version 11.3/1.0, Retrans: 1, Retries: 0
wisconsin#

```

本实验的最后一个例子 11-42 是路由器 georgia、wisconsin 和 minnesota 的完整配置清单。

例 11-42 路由器 georgia, wisconsin 和 minnesota 的配置清单

```

hostname georgia
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.6.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 bandwidth 64
 ip address 192.168.1.2 255.255.255.248
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 fair-queue 64 256 0
 frame-relay map ip 192.168.1.1 102 broadcast
 frame-relay map ip 192.168.1.3 102 broadcast
 frame-relay lmi-type cisco
!
router eigrp 65001
 network 172.16.0.0
 network 192.168.1.0
 no auto-summary
 eigrp stub connected
!

hostname wisconsin
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.5.1 255.255.255.128
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 bandwidth 128
 ip address 192.168.1.1 255.255.255.248
 no ip directed-broadcast

```

(待续)

```
no ip split-horizon eigrp 65001
frame-relay map ip 192.168.1.2 121 broadcast
frame-relay map ip 192.168.1.3 150 broadcast
!
interface Serial0.2 point-to-point
ip address 192.168.2.1 255.255.255.0
no ip directed-broadcast
ip summary-address eigrp 65001 192.168.1.0 255.255.255.0 5
ip summary-address eigrp 65001 172.16.5.0 255.255.255.0 5
frame-relay interface-dlci 111
!
interface Serial1
no ip address
no ip directed-broadcast
shutdown
!
interface BRI0
no ip address
no ip directed-broadcast
shutdown
isdn guard-timer 0 on-expiry accept
!
router eigrp 65001
network 172.16.0.0
network 192.168.1.0
network 192.168.2.0
no auto-summary

hostname minnesota
!
<<<text omitted>>>
!
interface Ethernet2
ip address 172.16.2.1 255.255.255.0
media-type 10BaseT
!
<<<text omitted>>>
!
interface Serial0
ip address 192.168.2.2 255.255.255.0
encapsulation frame-relay
no ip mroute-cache
frame-relay interface-dlci 110
!
router eigrp 65001
redistribute rip
network 172.16.0.0
network 192.168.2.0
default-metric 1544 100 254 1 1500
no auto-summary
!
router rip
redistribute eigrp 65001
network 172.16.0.0
default-metric 4
!
```

11.13 实验 23：配置 EIGRP 网络：默认路由、 路由的管理与过滤——第 1 部分

11.13.1 实验说明

现在大多数的网络都以某种形式与 Internet 互连。与 Internet 的互连通常需要一条默认路由，而且需要传播到整个网络中去。这个实验是给大家一个机会，可以练习对路由的控制以及在 EIGRP 网络中生成默认路由。

11.13.2 实验内容

假定几家网络咖啡店及其供应商打算合伙租用到 Internet 的连接。Solar Bucks Inc., 瑞典的 G & S INC, 还有 Barneys 决定在向各自客户提供新服务的同时相互共享它们的网络。而一些网络咖啡店也拥有自己的私有网络，他们不愿把自己的网络传播到其他网络咖啡店去。现在的任务是按照下面的要求配置一个 EIGRP 网络：

- 按照图 11-14 配置一个 IP 网络，使用 EIGRP 作为路由选择协议，自治系统 ID 为 2001。
- 将帧中继网络配置为一个所有路由器之间的点对点网络，不要创建多点网络。
- 不允许任何其他网络咖啡店知道路由器 barneys 上的子网 172.16.3.0/24。
- 在路由器 solar_bucks 上加入一条默认路由，指向路由器 internet_router。
- 路由器 solar_bucks 和 g_and_s 之间的直接帧中继连接是非常贵的。对 EIGRP 进行配置以使得来自路由器 g_and_s 的数据先去到 barneys，然后再转到 solar_bucks 去。如果 barneys 和 g_and_s 之间的 PVC 出现故障，数据就会直接从 g_and_s 发送到 solar_bucks。

11.13.3 实验目的

- 按照图 11-14 对网络咖啡店的网络以及 IP 进行配置。这个实验中 LAN 的拓扑类型对实验没有影响。
- 在 WAN 上使用帧中继数据链路协议，在帧中继网络上则只使用点对点网络。
- 确保所有 IP 接口的完全 IP 连接——也是说除了那些被过滤掉的接口以外的所有帧中继与 LAN 接口都可以 ping 通。
- 对来自路由器 g_and_s 和 solar_bucks 的网络 172.16.3.0/24 进行过滤。
- 向路由器 solar_bucks 中加入一条默认路由，把所有流量都指向路由器 internet_router。

器 barneys，去往 172.16.50.0/0 的数据同样要先经过 barneys。此外，不要使用路由策略。

11.13.4 所需设备

- 5 台 Cisco 路由器，其中 4 台要通过 V.35 背对背线缆或者是类似的方式与帧中继交换机连接在一起。
- 利用集线器或交换机构建 4 个 LAN 网段。LAN 的拓扑类型对实验没用影响。Internet 连接可以是真实的也可以模拟，这不会对路由器的配置产生影响。

11.13.5 物理设计与实验准备

- 按照图 11-14 将集线器以及串行线缆与路由器相连。
- 将剩下的那台路由器配置来作为与 Internet 的连接，其路由选择协议采用 EIGRP。
- 此外，还需要一台具有 3 条 PVC 的帧中继交换机。例 4-13 是整个实验中的帧中继交换机的配置示例。

例 11-43 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
 frame-relay route 121 interface Serial3 102
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 110 interface Serial0 111
 frame-relay route 130 interface Serial3 131
!
interface Serial2
 no ip address
 shutdown
!
interface Serial3
 no ip address
 encapsulation frame-relay
 clockrate 64000
```

（待续）


```

frame-relay intf-type dce
frame-relay route 102 interface Serial0 121
frame-relay route 131 interface Serial1 130
!

```

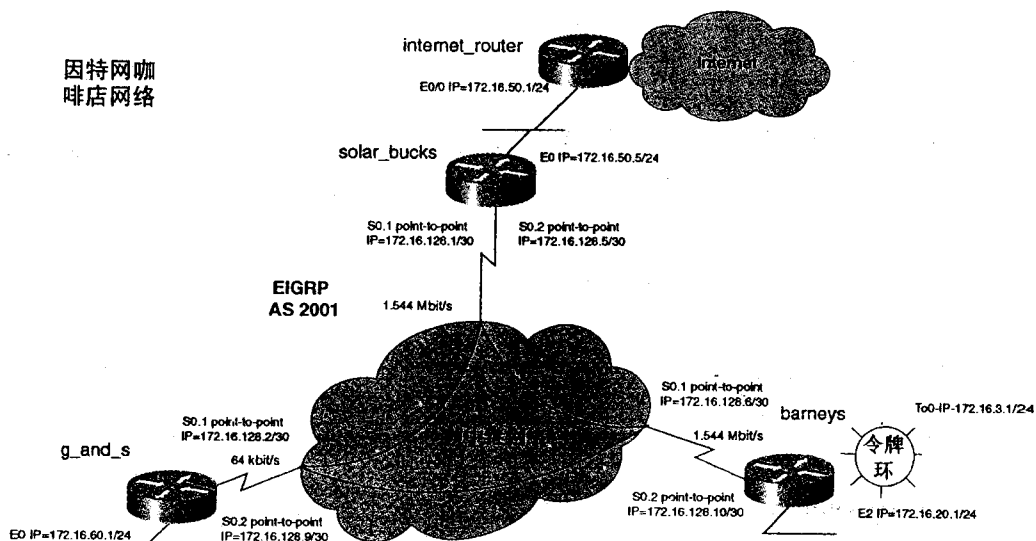


图 11-14 网络咖啡店网络拓扑

11.14 实验 23：配置 EIGRP 网络：默认路由、路由的管理与过滤——第 2 部分

11.14.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 3 台路由器以背对背的方式连接在一起。然后按照图 11-14 利用交换机或集线器/MAU 创建 4 个 LAN 网段。

物理连接完成之后，按照图 11-14 所示为所有的 LAN 和 WAN 接口配置 IP 地址。在继续下面的操作之前，必须用 **ping** 命令对每台路由器的本地 LAN 和 WAN 接口进行测试。然后，所有路由器之间的点对点接口上还需要用 **frame-relay interface-dlci** 命令进行配置。例 11-44 是目前为止所有相关路由器上的配置示例。

例 11-44 帧中继配置

```

hostname solar_bucks
!
<<<text omitted>>>
!
interface Serial0

```

(待续)

	章 混	增强型	网关路由	协议 (EI	573
address directe sulation mroute- -relay 1	adcast e-relay				
1 i face Seri dress 17 directe -relay 1	point- 128.1 2 adcast ace-dlc	int 5.255.25			
1 i face Seri dress 17 directe -relay 1	point- 128.5 2 adcast ace-dlc	int 5.255.25			
1 i face Seri dress 17 directe -relay 1	point- 128.2 2 adcast ace-dlc	int 5.255.25			
1 i face Seri dress 17 directe -relay 1	point- 128.9 2 adcast ace-dlc	int 5.255.25			
1 i face Seri dress 17 directe -relay 1	point- 128.6 2 adcast ace-dlc	int 5.255.25			
1 i face Seri dress 17 directe -relay 1	point- 128.10 2 adcast ace-dlc	int 5.255.25			

这 验的 EI 基本配 比上一 验还要 网络中 不连续的 网，因此，
没有必 关闭 RIG 自动汇 能。帧中 网络是一 因而水 隔的问题
也就不 问题了。 配置 EI 的 3 个 只需求 动 EIGR 由，AS 为 2001。
每个 禁止用于 需 参 后 24 小时内 删除， 如您喜 欢 书， 请 购 买 正 版。 若 因 自 私 散 布 造 成 法 律 问 题， 本 人 概 不 负 责。

EIGRP 路由配置工作。通往 g_and_s 路由器的 PVC 只是 64 kbit/s 的，因此把去往 g_and_s 路由器的所有帧中继连接的带宽设置为 64。路由器 solar_bucks 的 EIGRP 配置部分如例 11-45 所示，它和所有其他路由器的 EIGRP 配置都一样。

例 11-45 到目前为止所有路由器的 EIGRP 配置

```
!
router eigrp 2001
 network 172.16.0.0
!
```

现在就可以利用 ping 命令以及路由表的查询来对路由转发进行检查了。如果基本路由转发功能工作正常，就可以继续下面的操作，即要求路由器不在 EIGRP 域中传播子网 172.16.3.0。要做到这一点，方法很多，但是这个实验中，采用的是分布列表 (distribution list)。这份列表是要适用于从路由器 barneys 的 s0.1 和 s0.2 接口发送出去的 EIGRP 路由更新信息。例 11-46 是配置访问列表来拒绝网络 172.16.3.0/24 的过程，然后在 EIGRP 中又使用了分布列表来调用访问列表 10。为了防止路由返回网络，把这份分布列表应用在了接口 s0.1 和 s0.2 上。

例 11-46 分布列表的配置

```
barneys(config)#access-list 10 deny 172.16.3.0 0.0.0.255
barneys(config)#access-list 10 permit any
barneys(config)#router eigrp 2001
barneys(config-router)#distribute-list 10 out serial 0.1
barneys(config-router)#distribute-list 10 out serial 0.2
barneys(config-router)#^z
```

观察路由器 g_and_s 上的转发表，会发现找不到路由 172.16.3.0/24，如例 11-47 所示。但是子网 172.16.20.0/24 仍然可以 ping 通，因此过滤是成功的。

例 11-47 路由过滤效果的测试

```
g_and_s#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

172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C       172.16.128.8/30 is directly connected, Serial0.2
D       172.16.128.4/30 [90/41024000] via 172.16.128.1, 00:05:14, Serial0.1
          [90/41024000] via 172.16.128.10, 00:05:14, Serial0.2
C       172.16.128.0/30 is directly connected, Serial0.1
C       172.16.60.0/24 is directly connected, Ethernet0
D       172.16.50.0/24 [90/40537600] via 172.16.128.1, 00:05:14, Serial0.1
D       172.16.20.0/24 [90/40537600] via 172.16.128.10, 00:05:13, Serial0.2
```

```
D      172.16.0.0/16 is a summary, 01:10:38, Null0
g_and_s#ping 172.16.20.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.20.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/41/44 ms
g_and_s#
```

实验的下一个步骤是要求路由器 solar_bucks 将一条默认路由加入到 EIGRP 域中去。为此，配置一条默认静态路由，把所有流量都指向 internet_routers 路由器的以太端口上，下一跳地址为 172.16.50.1。而要路由器可以使用这个默认网络（路由），还需要设置 IP classless。静态路由由重分布进 EIGRP 中如例 11-48 所示，在路由器 solar_bucks 上配置默认路由。

例 11-48 配置 EIGRP 默认路由

```
solar_bucks(config)#ip route 0.0.0.0 0.0.0.0 172.16.50.1
solar_bucks(config)#router eigrp 2001
solar_bucks(config-router)#redistribute static
solar_bucks(config-router)#default-metric 1544 100 254 1 1500
solar_bucks(config-router)#^Z
solar_bucks#
```

现在来看看 g_and_s 或 barneys 路由器上的转发表，会发现配置好的默认路由已经在网络中传播了，而且系统还把该路由标记为一条外部的默认备选路由，如例 11-49 所示。

例 11-49 路由器 Barneys 上的默认路由

```
g_and_s#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, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is 172.16.128.1 to network 0.0.0.0

172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C      172.16.128.8/30 is directly connected, Serial0.2
D      172.16.128.4/30 [90/41024000] via 172.16.128.1, 00:20:43, Serial0.1
      [90/41024000] via 172.16.128.10, 00:20:43, Serial0.2
C      172.16.128.0/30 is directly connected, Serial0.1
C      172.16.60.0/24 is directly connected, Ethernet0
D      172.16.50.0/24 [90/40537600] via 172.16.128.1, 00:20:43, Serial0.1
D      172.16.20.0/24 [90/40537600] via 172.16.128.10, 00:20:42, Serial0.2
D      172.16.0.0/16 is a summary, 01:26:07, Null0
D*EX 0.0.0.0/0 [170/40537600] via 172.16.128.1, 00:09:12, Serial0.1
g_and_s#
```

实验的最后一步是设法改变 EIGRP 路由转发的选择。在前面的设置过程中，路由器 g_and_s 是以 solar_bucks 作为其访问 Internet 的路由。通过改变这条链路上的延时，可以对路由的选择施加影响，从而使得路由器 barneys 成为 g_and_s 访问 Internet 的首选路由。要达到这一目的，要在路由器 g_and_s 和 solar_bucks 之间 PVC 的两端都加上一条 delay 1000 命令。

例 11-50 是这个时候 g_and_s 上路由表的情况，它表明所有的路由现在都会首先通过 barneys 连接到 Internet。如果想要进一步测试这一配置，可以使用 trace 的方法。

例 11-50 加上延时之后路由器 g_and_s 的路由表

```
g_and_s#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 172.16.128.10 to network 0.0.0.0

    172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C       172.16.128.8/30 is directly connected, Serial0.2
D       172.16.128.8/30 [90/41024000] via 172.16.128.10, 00:00:01, Serial0.2
C       172.16.128.0/30 is directly connected, Serial0.1
C       172.16.60.0/24 is directly connected, Ethernet0
D       172.16.50.0/24 [90/41049600] via 172.16.128.10, 00:00:01, Serial0.2
D       172.16.20.0/24 [90/40537600] via 172.16.128.10, 00:00:11, Serial0.2
D       172.16.0.0/16 is a summary, 01:28:54, Null0
D*EX 0.0.0.0/0 [70/41049600] via 172.16.128.10, 00:00:02, Serial0.2
g_and_s#
```

例 11-51 是最后的配置列表。

例 11-51 网络咖啡店路由器的最终配置

```
hostname solar_bucks
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.50.5 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 bandwidth 64
 ip address 172.16.128.1 255.255.255.252
 no ip directed-broadcast
 delay 1000
 frame-relay interface-dlci 121
!
interface Serial0.2 point-to-point
 ip address 172.16.128.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 111
```

(待续)

```

!
<<<text omitted>>>
!
router eigrp 2001
 redistribute static
 network 172.16.50.0 0.0.0.255  + Optional 12.0 way, listed for example only
 network 172.16.0.0
 default-metric 1544 100 254 1 1500
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.50.1

```

```

hostname g_and_s
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 172.16.60.1 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 bandwidth 64
 ip address 172.16.128.2 255.255.255.252
 no ip directed-broadcast
 delay 1000
 frame-relay interface-dlci 102
!
interface Serial0.2 point-to-point
 bandwidth 64
 ip address 172.16.128.9 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 131
!
router eigrp 2001
 network 172.16.0.0
!
ip classless

```

```

hostname barneys
!
<<<text omitted>>>
!
interface Ethernet2
 ip address 172.16.20.1 255.255.255.0
 media-type 10BaseT
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point

```

```
ip address 172.16.128.6 255.255.255.252
frame-relay interface-dlci 110
!
interface Serial0.2 point-to-point
ip address 172.16.128.10 255.255.255.252
bandwidth 64
frame-relay interface-dlci 130
!
<<<text omitted>>>
!
router eigrp 2001
network 172.16.0.0
distribute-list 10 out Serial0.1
distribute-list 10 out Serial0.2
!
ip classless
!
access-list 10 deny 172.16.3.0 0.0.0.255
access-list 10 permit any
```

第 1 章

链路状态协议： 开放最短路径 优先（OSPF）

自从 1989 年诞生以来，OSPF 已经历了多次版本更新。现在，OSPF 已经成为网络工程任务组（IETF）正式发布的标准。OSPF 协议已经广泛地应用于各种网络中，它采用了一种分布式的路由算法，可以动态地适应网络的变化。OSPF 协议的主要特点如下：

- 快速收敛（Fast Convergence）
- 路由发布（Route Advertisement）
- 路由计算（Route Calculation）
- 路由更新（Route Update）
- 路由维护（Route Maintenance）
- 路由备份（Route Backup）
- 路由冗余（Route Redundancy）
- 路由负载均衡（Route Load Balancing）
- 路由故障检测（Route Fault Detection）
- 路由故障恢复（Route Fault Recovery）
- 路由故障隔离（Route Fault Isolation）
- 路由故障清除（Route Fault Clearing）
- 路由故障预防（Route Fault Prevention）
- 路由故障处理（Route Fault Handling）
- 路由故障报告（Route Fault Reporting）
- 路由故障记录（Route Fault Logging）
- 路由故障分析（Route Fault Analysis）
- 路由故障诊断（Route Fault Diagnosis）
- 路由故障修复（Route Fault Repair）
- 路由故障预防（Route Fault Prevention）
- 路由故障处理（Route Fault Handling）
- 路由故障报告（Route Fault Reporting）
- 路由故障记录（Route Fault Logging）
- 路由故障分析（Route Fault Analysis）
- 路由故障诊断（Route Fault Diagnosis）
- 路由故障修复（Route Fault Repair）

OSPF 一直随着现代网络的进步而不断进步。OSPF 已经成为开放式系统内部路由选择协议。对它的不断升级，OSPF 协议已经扩展到整个节点。

OSPF 协议是由互联网工程任务组（IETF）正式发布的，它弥补了距离矢量协议存在的缺陷。1989 年，OSPF 协议的补充升级，最初是 RFC 1131，然后是 RFC 1247 中修订后的版本 2，然后又是 RFC 2328 的版本 II。在现代大型网络中，OSPF 已经成为标准的开放式路由选择协议。所谓开放式路由选择协议，是指路由选择算法，Dijkstra 的最短路径优先级算法，由网络中的每个节点（路由器）来计算，而不属于某个组织。这使得其他厂商的产品，如 Unisys、DEC 的产品一样可以采用 OSPF 协议。

OSPF 协议对传统的距离矢量协议进行了很多卓有成效的改进。

OSPF 采用了一种可靠的扩散算法，根据网络拓扑变化来更新邻居路由。当网络拓扑发生变化时，只发送产生的路由更新信息。整个网络在 OSPF 域中所有路由器上运行，具有完全一样的特点，使得 OSPF 在某个网络中能够迅速地收敛。

- 支持 VLSM、超网以及汇总功能——OSPF 采用汇总功能和 VLSM 来节省地址空间，高效地进行路由。
- 支持大型网络结构——通过采用 VLSM 和以区域为基础的分层网络设计思想，OSPF 网络的规模实际上可以做到无限大。
- 存根区域路由——OSPF 采用存根区域路由大大减小了路由表的规模，这也是 OSPF 支持大型网络的一个原因。
- 路由更新的高效、可靠传输——通过采用两个保留多播地址来传输路由更新信息，运行 OSPF 协议的路由器对未运行 OSPF 的路由器和设备不会造成影响。OSPF 通过对数据包显式或隐式的确认应答保证了路由更新传输的可靠性。
- 介质利用的高效性——多播报文只出现在广播介质，而在 NBMA 和点对点网络使用单播。
- 基于路由代价 (Cost) 的可变度量——OSPF 采用可变路由代价度量标准来决定路由选择。
- 等价路由的负载平衡——OSPF 在等价路由的路径上进行负载平衡，以对带宽和多条路径进行优化。
- 支持类型 I 和类型 II (MD5) 的认证方式——OSPF 通过类型 1 的明文密码或者是类型 2 的 MD5 加密认证方式来确保安全可靠的路由传输。
- 支持 OSPF 外部路由的路由标记功能——可以为重分布进入 OSPF 的外部路由添加标记，作为在自治系统或内部结构中对路由进行控制管理的又一途径。
- 完全的无类路由选择协议支持——OSPF 支持无类路由表的查找，不会遇到有类协议的问题，如不连续子网等。
- 没有水平分隔的问题。

尽管具有这么多优点，但很多人却批评 OSPF 的配置过于复杂，而且需要占用大量的处理器资源。这是真的，即使是最小的 OSPF 网络，在实际使用之前也需要作少量的设计工作，而且还比其他路由选择协议要多的占用更多处理器资源。但是，随着现代高速 CPU 的出现，OSPF 所需要的资源对于这些高速 CPU 来说已经不再成为问题。本章中在讨论 OSPF 技术问题时将更多地侧重 OSPF 配置方面。

12.1 OSPF 技术概览

OSPF 是一个无类路由选择协议，利用 IP 协议号 89 直接交换路由信息。OSPF 采用多播 hello 信号和路由限时器的概念来发现和维持邻居路由器。OSPF 路由更新称为链路状态通告 (LSA)，OSPF 拓扑表通常称为链路状态数据库。OSPF 在区域内扩散发送 LSA，直到路由器对网络映像 (称为链路状态数据库) 完全一致。当路由器对网络映像达到一致时，SPF 算法，或称 Dijkstra 算法在数据库中运行，而且创建了用来描述通往每一目的路径路由代价最短的无环路径图表，称为 SPF 树。路由表或转发表中的 OSPF 路由就是从这个 SPF 树中产生的。每台路由器中都有整个 SPF 树的完全拷贝，因此可以进行快速收敛。在确定到某个目的地址的最短路径时，OSPF 采用基于路由代价的可变度量标准。

从一个网络传播到另一个网络的关键。1. 当路由器 (称为 PFRout) 通过特定网络信息 (hello 数据包) 2. 当数据包经过邻居。3. 当路由器建立邻居。4. 当邻居路由器记录新信息。5. 当路由器的路由数据库 (完并之路径，路由器) 6. 当 SPF 算法并符合其中。1. OSPF 的邻居发现协议

如所述，OSPF 的 hello 数据包每 30s。hello 数据包包括：源 IP 地址、源接口 ID、源接口地址掩码、源接口地址类型和 hello 数据包的间隔时间、源接口地址的存活时间、路由器的优先级、指定路由器 (DR) 和备份指定路由器 (BDR)、5 个标志、源路由器的邻居 ID。接收标准：OSPF 在邻居路由器之间建立关系的条件包括 hello 数据包发送间隔、helloInterval、路由器的存活时间 (RouterDeadInterval)、区域号、认证类型和密码必须匹配。邻居路由器之间的存活时间 (keepalive) 通常是 40s，链路类型有：如果在邻居路由器消亡时间段内没

- **广播和 NBMA 网络上 DR 和 BDR 的选择**——路由器 ID、DR 和 BDR 字段以及路由器优先级等因素有助于确定 DR 和 BDR 的状态。后面的内容会有更多关于 DR 和 BDR 的讲述。

注释 有时，邻居路由器 (neighbors) 和邻接关系 (adjacencies) 这两个名词作为同义词使用。在 OSPF 中，这两个名词相互之间有关系，但指不同的东西。RFC 2328 将邻居路由器定义为路由器的接口在共同的网络里。邻居路由器通过 OSPF 的 hello 协议来动态发现和维持。而邻接关系的定义是为实现交换路由信息的目的，在选出的邻居路由器之间建立的关系。并不是所有的邻居路由器都具有邻接关系。

12.1.2 OSPF 的邻居路由器和网络类型

正如从 Frost 的诗句所类推出来的，“有好的邻居路由器就有好的网络 (Good neighbors make good networks)”这不仅适用于 EIGRP，同样也适用于 OSPF。和 EIGRP 一样，只有邻居路由器之间建立起了邻接关系，链路状态才能够相互交换。稳定的 OSPF 邻居路由器对于 OSPF 网络非常重要。OSPF 对邻居路由器的处理和链路状态的传播与本地路由器和邻居路由器所在网络类型有关。OSPF 网络有 5 种类型：

- **点对点网络**——点对点网络的例子包括 HDLC 网络、PPP 和具有点对点接口的帧中继网络。链路状态和 hello 数据包使用多播地址 224.0.0.5。这种网络中没有 DR 或 BDR 的选择问题。这是 Cisco 专有的网络类型，没有 RFC 的定义。
- **广播网络**——该类网络包括以太网、令牌环网和 FDDI 网络。hello 数据包使用多播地址 224.0.0.5 选择一个 DR 和一个 BDR，网络中其他的路由器发送链路状态的多播地址是 224.0.0.6。只有 DR 和 BDR 会监听该地址的更新信息，而这两台路由器又会反过来将链路状态通过地址 224.0.0.5 发送到其他路由器。下一节会讨论 DR 和 BDR 的选择和功能问题。
- **NBMA 网络**——NBMA 网络包括普通的或者是具有多点接口的帧中继以及 X.25 网络。在这类网络中，多播数据包并不像通常那样转发到所有的邻居路由器，因为这类网络不具备广播功能。因此，必须静态定义 OSPF 的邻居路由器。网络中会选出一个 DR 和一个 BDR，所有的 OSPF 数据包都以单播形式传输。NBMA 网络中，DR/BDR 应该是具有 PVC、SVC 或者是能通往所有其他路由器去的线路，或者是称为中心路由器的路由器。
- **点对多点网络**——点对多点网络必须静态定义。路由器把帧中继多点网络看成多个点对点网络链路。网络中不选举 DR 和 BDR，OSPF 数据包以多播形式发送。
- **虚链路**——虚链路是一种特殊类型的网络，作用是对区域 Area 0 进行扩展。虚链路会在“OSPF 虚链路”一节中进行讲述。

12.1.3 指定路由器 (DR) 和备份指定路由器 (BDR)

在以太网、令牌环网，FDDI 网这样的多路访问网络上，如果每台邻居路由器都向其所有邻居路由器宣告链路状态，网络的效率会急剧降低。而每台路由器之间如果都具有邻接关

系会使得这一问题更加恶化。事实上，OSPF 会选出一台路由器，称为指定路由器 (DR)。这台路由器会监听多播地址 224.0.0.6 上的链路状态并将状态地址 224.0.0.5 转发到其他路由器上。这是除了备份指定路由器 (BDR) 之外惟一监听 224.0.0.6 处的链路状态更新信息的路由器。BDR 会同步跟踪 DR 的工作并对其进行备份，仅当 DR 出现问题之后才会接替 DR 工作。从本质上看，DR/BDR 方案提供了如下优点：

- 通过对链路状态的收发管理达到简化路由更新信息的目的。
- 对于 OSPF 域中其他路由器来说，DR 和 BDR 就代表整个多路访问网络。作为惟一的控制点，DR 还确保了在这个多路访问网络中的路由器具有一致的链路状态信息。
- BDR 的概念能加速网络的同步过程。所有的路由器对 BDR 来说都是邻接的，如果 DR 发生问题，BDR 可以在很短的时间内接替 DR 工作。

一旦选举了 DR 和 BDR，新加入的路由器都只和 DR 和 BDR 建立邻接关系。DR 和 BDR 之间也是相互邻接的关系。

路由器选举 DR 和 BDR 的过程如下：

1 参与选举的邻居路由器必须首先是处在双向 (2-way) 状态。也就是说，每台路由器都和参与选举的其他路由器之间进行 hello 数据包的接收和发送。可以参考“OSPF 的基本邻接关系”一节。

2 选举过程中要查看优先级的值。优先级值为 0 的路由器被排除在选举范围之外。优先级最高的邻居路由器成为 BDR。如果有优先级相等的路由器，RID 最高的路由器当选。OSPF 的优先级默认值为 1，通过接口命令 `ip ospf priority [0-255]` 可以更改优先级的值。

3 如果链路中没有发现有 DR，BDR 就会升级为 DR，然后再选举一个新的 BDR。如果 BDR 的选举过程中出现相同条件，则通过路由器 ID 来解决。路由器 ID 最高的成为 BDR。

4 如果有一台优先级更高的路由器加入了网络，不会立即进行新的 DR 和 BDR 选举。只有 DR 或 BDR 发生问题时，DR/BDR 的选举才会进行。

5 路由器会继续每 10s (广播网络中默认的) 交换一次 hello 数据包。如果在消亡计时器的时间段 (hello 计时器的 4 倍) 内某路由器没有收到邻居路由器发送的 hello 数据包，路由器就会摒弃该邻居路由器。

从本质上看，DR/BDR 的选举使得 OSPF 简化了网络中路由信息的更新。在图 12-1 的以太网网络中能够看到大型网络中路由过程的效率是如何迅速降低的。没有 DR/BDR，路由器需要和网络中所有其他路由器进行 LS 信息的交换。

具有适当的 DR 和 BDR (图 12-2) 后，LS 信息或路由信息就处在 DR 的控制管理之下。

12.1.4 OSPF 的路由器标识 (RID)

OSPF 的路由器标识 (RID) 是分配给每台运行 OSPF 的路由器的惟一 32 位号码。这个号码在自治系统中对路由器进行了惟一标识。通过为 AS 中每台路由器分配惟一的 OSPF 路由器标识 (RID)，使 OSPF 能够实现下列目的：

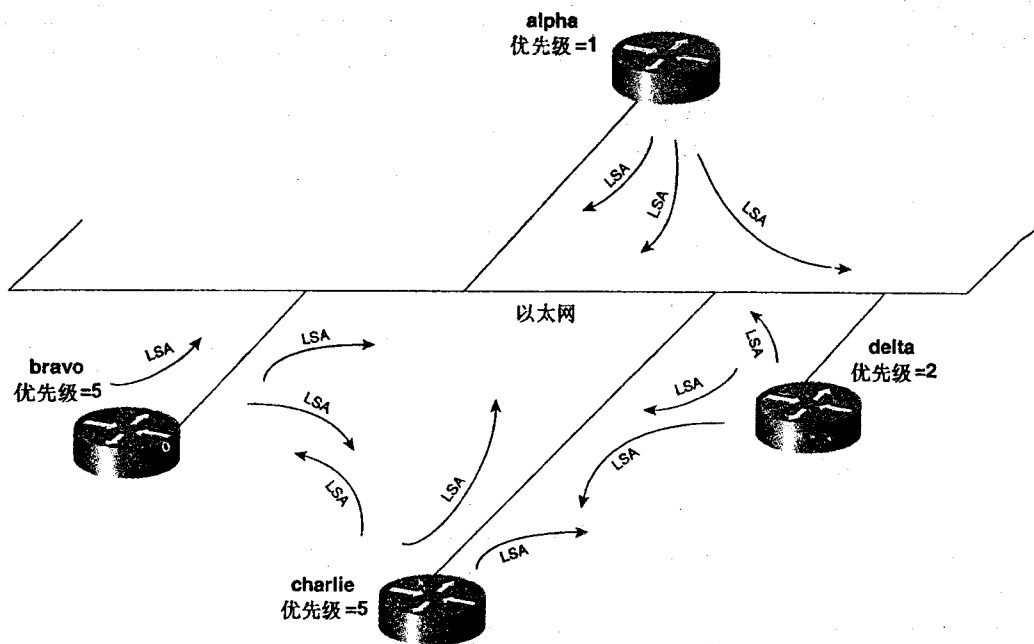


图 12-1 没有 DR 和 BDR 的 OSPF 以太网 LS 信息的传播 (假定)

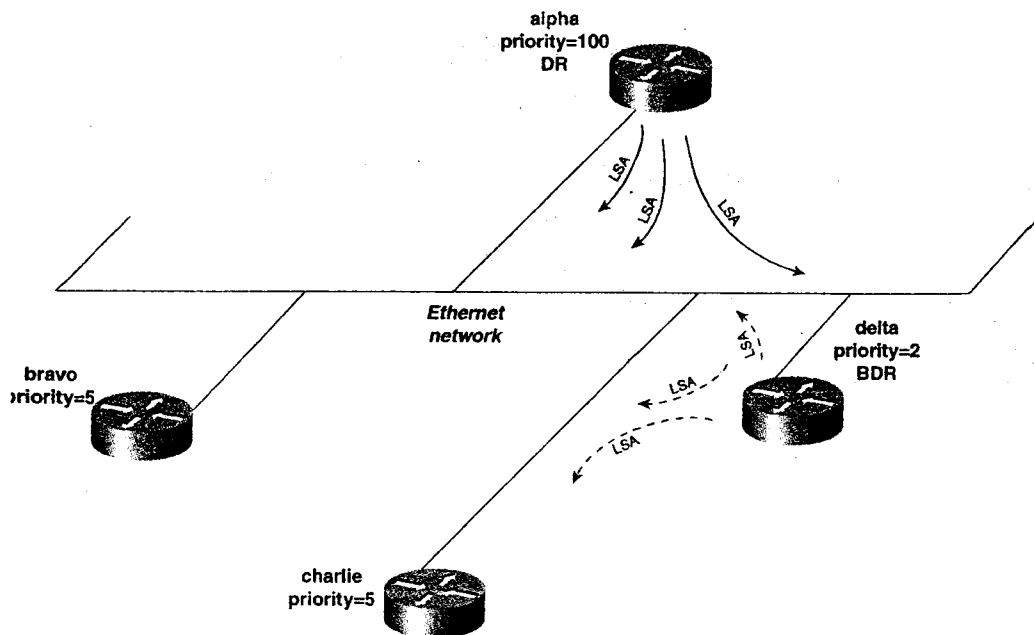


图 12-2 具有 DR 和 BDR 的 LS 传播

- 轻易识别重复的 LSA。
- 识别惟一的虚链路终端。
- 打破 DR/BDR 选举过程中的平等条件。

路由器 ID 是从为 Cisco 路由器配置的 IP 接口中选出来的。路由器从所有工作的 IP 接口选出值最高的 IP 地址，被选接口的线路和线路协议都必须处在工作状态之中。如果在路由器上定义环路地址，路由器就会选取这个环路地址。如果定义了多个环路地址，路由器会选取接口 IP 地址最高的环路地址。

可以使用具有高 IP 地址（如 192.168.200.X）的环路接口强制指定路由器 ID。没有必要将此网络传播到路由选择协议中。这类网络（更确切地说是用于路由器 ID 的 IP 主机地址）是不需要别人访问的地址，或者说不必是可以 ping 通的地址。在 12.0 及以上版本的 Cisco IOS 中，OSPF 的路由器 ID 可以通过 OSPF 路由命令进行设定：

```
Router (config-router) #router-id ip_address
```

技巧 建议用 **router-id** 命令或者利用环路接口来设定路由器 ID，这样能够大大提高 OSPF 网络的稳定性。例如，OSPF 的虚链路以路由器 ID 为基础。如果路由器 ID 不固定，网络中加入新网络或环路接口时，路由器 ID 会因为路由器的的问题而重新进行计算，导致路由器 ID 的改变，使虚链路出现问题。

12.1.5 OSPF 的基本邻接关系

OSPF 的邻居路由器在进行 LSA 交换前会对一些状态进行检查，如图 12-3 所示。这些状态称为邻居路由器的状态机。**show ip ospf neighbor** 命令可以查看 OSPF 邻居路由器的状态。

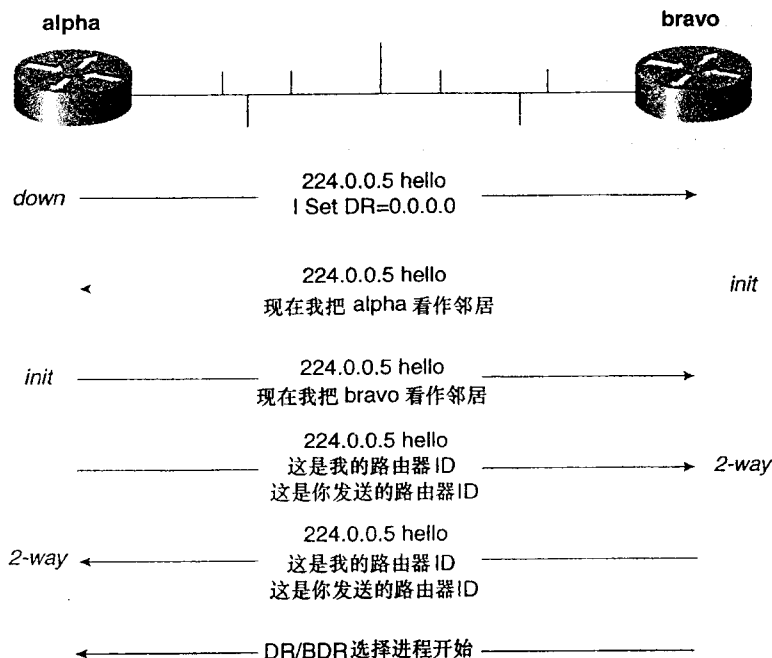


图 12-3 基本的 OSPF 邻居关系图

下面简要描述了 OSPF 邻居路由器的状态及其工作方式：

- **停止 (Down)** —— 邻居路由器的初始状态，表明上一个消亡时间段里没有收到来自该邻居路由器的 hello 数据包。
- **尝试 (Attempt)** —— 这种状态只适用于 NBMA 网络中的邻居路由器，表明邻居路由器已经用 **neighbor** 命令静态设置。接口进入工作状态后，或者当路由器为 **DR** 或 **BDR** 时，路由都会进入 **ATTEMPT** 状态。
- **初始 (Init)** —— 表明已经从邻居路由器接收到 hello 数据包，但还没有开始双向通信。
- **双向 (2-way)** —— 该状态表明路由器在从邻居路由器接收到的 hello 数据包的 Neighbor 字段中发现了自己的路由器 ID 号，也说明双向通信已经建立，可以进行 **DR** 和 **BDR** 选举。

当 OSPF 的接口进入工作状态之后，会发送出 hello 数据包。路由器收到相互的 hello 数据包后，会将邻居路由器置于 *init* 状态。邻居路由器处在 *init* 状态时，会将自己的路由器 ID 放到 hello 数据包中。路由器收到含有它的邻居路由器的 ID 号的 hello 数据包时，会将此邻居路由器置为 *2-way* 状态。*2-way* 状态保证在路由器之间建立双向通信途径。路由器要进行 **DR/BDR** 选举以及交换 LSA 就必须处在这一状态。

路由器完成 *2-way* 状态的工作之后，OSPF 就会进入最后状态：

- **开始交换 (ExStart)** —— 路由器之间形成一种主从关系，准备传输数据库描述数据包。接口地址最大的邻居路由器会成为主路由器。
- **交换 (Exchange)** —— 路由器会在 *exchange* 状态下向邻居路由器发送数据库描述数据包。数据库描述数据包描述整个链路状态数据库。链路状态数据库可以在这个阶段完成相互同步的工作。同步之后，路由器为下面两个最终状态之一：
 - **装载 (Loading)** —— 路由器会向所有处在 *loading* 状态中的路由器发送链路状态请求数据包。该状态要求发送最新的 LSA。
 - **完全邻接 (Full)** —— 该状态中的路由器具有完整的邻接关系。

OSPF 邻接关系的建立可以总结为下面 4 个阶段：

- 1 发现邻居路由器。
- 2 在邻居路由器之间建立双向通信。
- 3 对 SPF 数据库进行同步。
- 4 建立完整的邻接关系。

用 **show ip ospf neighbor** 命令可以查看 OSPF 邻接关系的状态，而 **debug ip ospf adj** 命令则能提供邻接关系实际建立过程的信息。后面会详细讲述 OSPF 状态查询命令。

图 12-4 中，现有的 OSPF 网络中加入了路由器 *charlie*。通过 **debug** 命令可以观察邻接关系的建立过程，如例 12-1 所示。

例 12-1 用 **debug ip ospf adj** 命令和 **show ip ospf neighbor** 命令查看邻接关系的形成

```
charlie#debug ip ospf adj
OSPF adjacency events debugging is on

OSPF: Interface Ethernet0 going Up
```

```

OSPF: Build router LSA for area 0, router ID 172.16.1.1
OSPF: Build router LSA for area 0, router ID 172.16.1.1
OSPF: Build router LSA for area 0, router ID 172.16.1.1
%SYS-5-CONFIG I: Configured from console by console
OSPF: 2 Way Communication to 172.16.1.10 on Ethernet0, state 2WAY --Router enters
two-way state
OSPF: Build router LSA for area 0, router ID 172.16.1.1
OSPF: 2 Way Communication to 172.16.1.5 on Ethernet0, state 2WAY
OSPF: Backup seen Event before WAIT timer on Ethernet0
OSPF: DR/BDR election on Ethernet0 --DR/BDR election begins
OSPF: Elect BDR 172.16.1.5
OSPF: Elect DR 172.16.1.10
      DR: 172.16.1.10 (Id)   BDR: 172.16.1.5 (Id)
OSPF: Send DBD to 172.16.1.5 on Ethernet0 seq 0x1370 opt 0x2 flag 0x7 len 32
OSPF: Send DBD to 172.16.1.10 on Ethernet0 seq 0x218C opt 0x2 flag 0x7 len 32
OSPF: Build router LSA for area 0, router ID 172.16.1.1
OSPF: Rcv DBD from 172.16.1.10 on Ethernet0 seq 0x1137 opt 0x2 flag 0x7 len 32
      state EXSTART --EXSTART state begins slave/master will be selected
OSPF: NBR Negotiation Done. We are the SLAVE
OSPF: Send DBD to 172.16.1.10 on Ethernet0 seq 0x1137 opt 0x2 flag 0x2 len 52
OSPF: Rcv DBD from 172.16.1.5 on Ethernet0 seq 0x16D9 opt 0x42 flag 0x7 len 32
      state EXSTART --EXSTART state begins for the other neighbor
OSPF: NBR Negotiation Done. We are the SLAVE
OSPF: Send DBD to 172.16.1.5 on Ethernet0 seq 0x16D9 opt 0x2 flag 0x2 len 52
OSPF: Rcv DBD from 172.16.1.10 on Ethernet0 seq 0x1138 opt 0x2 flag 0x3 len 92
      state EXCHANGE --Exchange state begins for one neighbor
OSPF: Send DBD to 172.16.1.10 on Ethernet0 seq 0x1138 opt 0x2 flag 0x0 len 32
OSPF: Database request to 172.16.1.10
OSPF: sent LS REQ packet to 172.16.1.10, length 36
OSPF: Rcv DBD from 172.16.1.5 on Ethernet0 seq 0x16DA opt 0x42 flag 0x3 len 92
      state EXCHANGE --Exchange state begins for the other neighbor
OSPF: Send DBD to 172.16.1.5 on Ethernet0 seq 0x16DA opt 0x2 flag 0x0 len 32
OSPF: Database request to 172.16.1.5
OSPF: sent LS REQ packet to 172.16.1.5, length 36
OSPF: Rcv DBD from 172.16.1.10 on Ethernet0 seq 0x1139 opt 0x2 flag 0x1 len 32 s
tate EXCHANGE
OSPF: Exchange Done with 172.16.1.10 on Ethernet0
OSPF: Send DBD to 172.16.1.10 on Ethernet0 seq 0x1139 opt 0x2 flag 0x0 len 32
OSPF: Synchronized with 172.16.1.10 on Ethernet0, state FULL --LS database is
synced and the adjacency is in FULL status for this neighbor
OSPF: Build router LSA for area 0, router ID 172.16.1.1
OSPF: Rcv DBD from 172.16.1.5 on Ethernet0 seq 0x16DB opt 0x42 flag 0x1 len 32 s
tate EXCHANGE
OSPF: Exchange Done with 172.16.1.5 on Ethernet0
OSPF: Synchronized with 172.16.1.5 on Ethernet0, state FULL --The same "FULL"
state is achieved with this neighbor
OSPF: Build router LSA for area 0, router ID 172.16.1.1
OSPF: Send DBD to 172.16.1.5 on Ethernet0 seq 0x16DB opt 0x2 flag 0x0 len 32
OSPF: Build router LSA for area 0, router ID 172.16.1.1
charlie#

charlie#show ip ospf neighbor

Neighbor ID      Pri   State           Dead Time   Address      Interface
172.16.1.5        1    FULL/BDR        00:00:35   172.16.1.5   Ethernet0
172.16.1.10       1    FULL/DR         00:00:30   172.16.1.10  Ethernet0
charlie#
    
```

这个例子中，172.16.1.10（也就是 alpha 路由器）由于其 IP 地址是整个链路中最大的 IP，

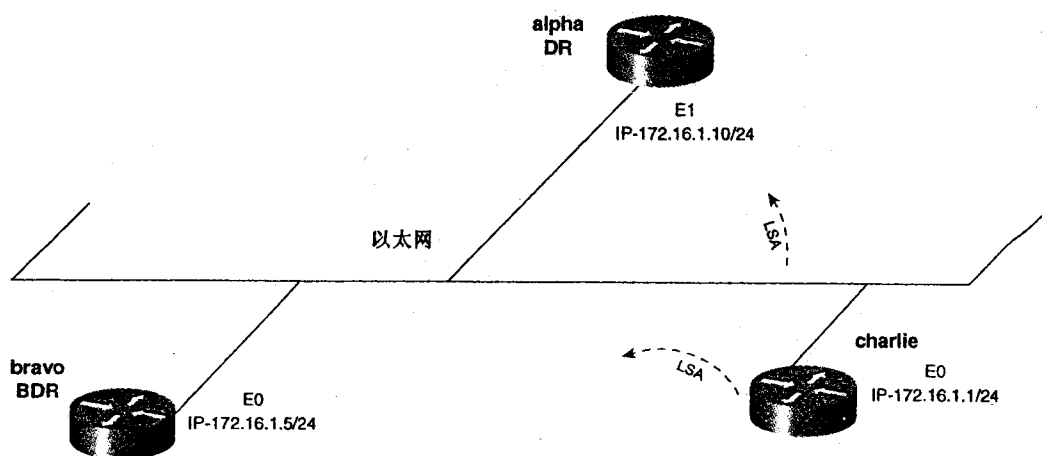


图 12-4 OSPF 基本邻接关系

12.1.6 最短路径树 (SPF) 和 OSPF 的度量代价

区域内的 LS 数据库同步之后, Dijkstra 算法会经过两个步骤遍历数据库形成最短路径树 (SPF)。第一次遍历 SPF 数据库形成树的分支, 也就是域中的路由器邻接关系。第二次遍历 SPF 数据库则是为树增加枝叶, 即存根网络。OSPF 建立 SPF 树时, 会以到目的地址路由代价的和为基础来确定每个目的地址的最短路径。路由代价越低, 路由越优先。路由代价是到达该目的地址经历所有输出接口代价的和。奇怪的是, RFC 2328 没有为路由代价指定特定的值。Nortel Networks 运用 RFC 2328 的 OSPF 时, 用来计算路由代价的公式和 Cisco 一样。如果网络中含有多个供应商提供的设备, 一定要多花一点时间注意路由代价的计算方法, 以保证整个 OSPF 网络的一致性。

Cisco 路由器用 $(100000000 / BW)$ 公式来计算 OSPF 代价, 计算的结果向下舍入, BW 是设置或者默认的带宽值。表 12-1 列出了常见的默认 OSPF 代价的设置情况。

表 12-1

默认的 OSPF 接口代价

接口类型	默认代价 (100000000 / BW)
FDDI, ATM, Fast Ethernet, Gigabit Ethernet (> 100 Mbit/s)	1
HSSI (45M)	2
16-Mbit/s Token Ring	6
10-Mbit/s Ethernet	10
4-Mbit/s Token Ring	25
T1 (1.544 Mbit/s)	64

续表

接口类型	默认代价 (100000000/BW)
DS-0 (64 kbit/s)	162
56 kbit/s	1785
Tunnel (9 kbit/s)	1111

默认代价值可以用 `ip ospf cost 1-65535` 命令加以修改，查看路由代价可以用 `show ip route` 命令。回顾以前的内容，命令结果中管理距离后面的变量就是路由器的度量代价值。图 12-5 给出计算代价的例子。路由器 echo 的路由表列出 172.16.2.0 网络的代价是 70。T1 加上 16 MB 令牌环网的带宽等于 70 ($64+6=70$)。因此，到以太网的路由的代价就是 80 ($64+6+10$)。

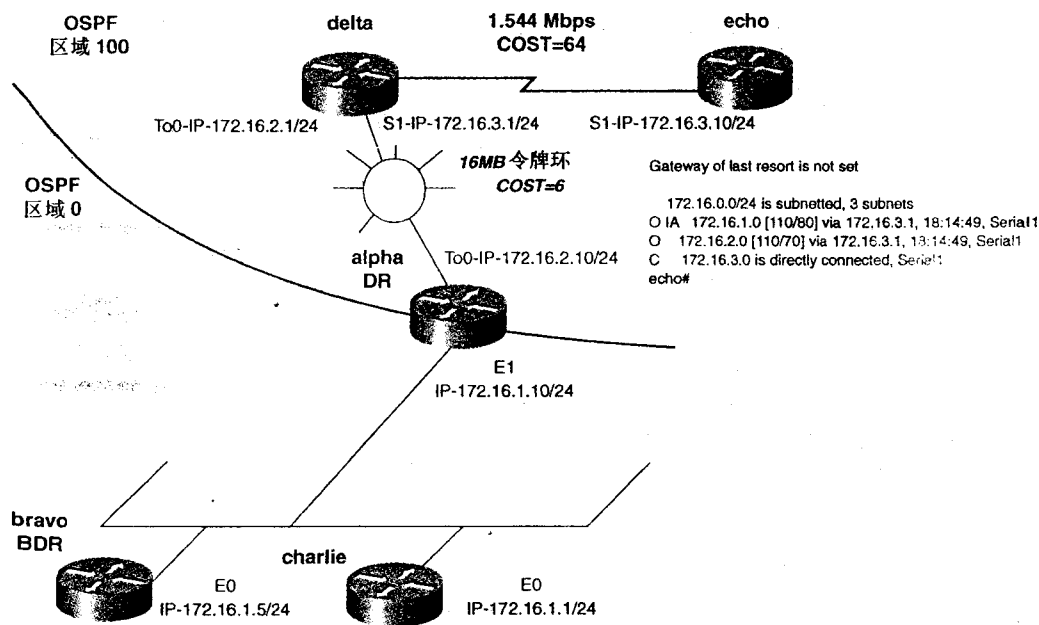


图 12-5 计算 OSPF 的代价

12.1.7 OSPF 的路由器类型、区域以及 LSA

本章开始曾提到 OSPF 可以看成 CPU 密集型的协议，配置较复杂。现在已经明白多个 SPF 数据库、复杂的算法以及大量发送的链路状态不断引起 CPU 中断都会大大增加对 CPU 资源的需求。

OSPF 采用区域概念控制链路状态的扩散（flooding）发送和数据库的同步。OSPF 区域可以定义为一组将 OSPF 域划分成很多子域的路由器和链路的集合。区域通过一个 32 位的区域 ID 来标识，这个 ID 可以用点分十进制计数法或者十进制数来表示。

OSPF 共有 5 种类型的区域

- **主干区域 (Backbone area)**，或称 **Area 0** (或 **0.0.0.0**) ——所有的数据必须通过主干区域，非主干区域不能直接交换数据。所有区域必须和 Area 0 邻接。主干区域必须连续，不能进行分区。但 Area 0 可以通过虚链路扩展。
- **非主干区域 (Nonbackbone)**，非存根区域 (**nonstub area**) ——除 Area 0 之外的标准 OSPF 区域。除了类型 7 以外的 LSA，都通过该区域进行传输。
- **存根区域 (Stub area)** ——存根区域中没有通告的外部路由，也不产生外部类型 5 的 LSA。骨干区域发送默认路由，或者目的地址为 0.0.0.0 的汇总 LSA 到存根区域。存根区域还有其他一些限制：
 - 不能在存根区域中配置虚链路。
 - 除了和区域边界路由器建立邻接关系之外，不能和其他非存根路由器建立邻接关系。
 - 存根区域中的路由器不能作为自治系统边界路由器 (ASBR)，因为外部路由或类型 5 的外部 LSA 都不会发送到存根区域。总之，存根区域不能进行重分布。
 - 类型 4 和 5 的 LSA 不能进入 Stub 区域，只有类型 1、2 和 3 的 LSA 才可以进入存根区域。
- **完全存根区域 (Totally stubby area)** ——外部路由和内部路由都不能进入该区域。类型 3、4 和 5 的 LSA 也禁止入内。只能宣告 LSA 类型 3 的默认路由。路由器利用该默认路由到达区域以外的目的地址。
- **非完全存根区域 (Not-so-stubby area) (NSSA)** ——有时需要将另一种路由选择协议 (如 RIP) 重分布到存根区域中。由于这样违反 Stub 区域的定义，因此需要一种新的区域类型。RFC 1587 就定义了非完全 Stub 区域 (NSSA)。该区域在保留 Stub 区域其他特性的基础上允许外部路由通过重分布进入区域内。外部路由重分布进入 NSSA 区域的路由器时，路由器产生类型 7 的 LSA，将外部目的地址发送到 NSSA 区域中的路由器。类型 7 的 LSA 进入 Area 0 时会由区域边界路由器 (ABR) 转换成类型 5 的 LSA。NSSA 区域中不存在类型 5 的 LSA。该区域中的所有路由器也必须配置成 NSSA。

OSPF 要求通过采用特定路由器类型控制之下的区域来实现分层网络设计目标。路由器可能同时属于多种类型，例如，区域边界路由器 (ABR) 同时也是主干路由器。OSPF 的路由器类型包括：

- **内部路由器 (Internal routers)** ——接口处于相同 OSPF 区域的路由器。所有内部路由器的 SPF 数据库都一致。
- **区域边界路由器 (Area Border Routers)** ——连接一个或多个区域与骨干区域的路由器。ABR 转发骨干区域的全部 LSA。ABR 至少有一个接口在 Area 0 中，并为每一个与之相连的区域保存了一份独立的 SPF 数据库。因此，ABR 应该是一些高端的路由器。
- **主干路由器 (Backbone routers)** ——在 Area 0 中至少具有一个接口。路由器所有的接口有可能都处在 Area 0 中，这种情况下，路由器是内部主干路由器。
- **自治系统边界路由器 (Autonomous System Boundary Routers)** ——对其他路由选择协议进行重分布或者发送外部路由的路由器称为自治系统边界路由器

链路状态通告 (LSA) 这个词已经提到多次了。OSPF 利用 LSA 建立 OSPF 数据库。OSPF 会根据上面提到的区域和路由器类型的定义和规则，将特定的 LSA 发送到 OSPF 域的特定部分。LSA 也有多种类型，每种类型的 LSA 都有特定的作用，下面是这些 LSA 的类型说明：

- **路由器 LSA (类型 1)** —— 该类 LSA 包含了一个区域中的路由器和及其链路的信息。类型 1 的 LSA 只在一个区域中发送。LSA 还能区分路由器是存根类型的还是 ASBR，或者路由器是否含有虚链路的一端等。OSPF 转发表中以 O 来代表这个类型。
- **网络 LSA (类型 2)** —— 该类 LSA 用于在区域中传输网络信息，描述了与网络相连的路由器集合。类型 2 的 LSA 不会被宣告到区域外。OSPF 转发表中也用 O 来代表这类 LSA。
- **ABR 汇总 LSA (类型 3)** —— 这类 LSA 用于将内部网络信息发送到区域以外的路由器上去，这些路由就称为域间路由。这类 LSA 可能含有一个汇总路由或一个单独的路由。ABR 是惟一能产生这类 LSA 的路由器。OSPF 在转发表中用 OIA 来标记该类 LSA。
- **ASBR 汇总 LSA (类型 4)** —— 该类 LSA 用于宣告 ASBR 的位置。寻找外部路由路径的路由器利用类型 4 LSA 来确定下一跳地址。OSPF 转发表中用 OIA 来标记该类 LSA。这个 LSA 类型很不好记，大家可以把它想像成“我怎样才能离开这里的 LSA”。
- **自治系统外部 LSA (类型 5)** —— 该类型 LSA 用于将路由重分布进入 OSPF，这样的路由称为 OSPF 外部路由。这些路由会在整个 OSPF 自治系统中除存根区域、完全存根区域以及 NSSA 区域之外的所有部分进行传输。OSPF 转发表中以 O E1 或 O E2 来标记该类 LSA，究竟使用哪一个视路由的类型而定。
- **NSSA 外部 LSA (类型 7)** —— 该类 LSA 是为了将外部路由重分布到非完全存根区域中去而产生的。这类 LSA 会在整个 NSSA 区域传输，到达 ABR 时，ABR 会将其转换成类型 5 的 LSA，再转发到 Area 0。类型 7 的 LSA 不会离开 NSSA 区域。OSPF 转发表中用 O N1 或 O N2 来标记这类 LSA，具体是哪一种视路由如何重分布而定。

表 12-2 总结了每个区域许可的 LSA。

表 12-2

每个区域中许可的 LSA 类型

区域类型	LSA 1 和 2	LSA 3 和 4	LSA 5	LSA 7
主干区域 (Area 0)	是	是	是	否
存根区域	是	是	否	否
完全存根区域	是	否*	否	否
非完全存根区域	是	是	否	是
非主干区域，非存根区域	是	是	是	否

* 有一种类型 3 的 LSA 用于宣告默认路由

注释 RFC 2370 定义了非透明链路状态。非透明 LSA 包括类型 9、10 和 11 的链路状态通告。这类 LSA 可以直接为 OSPF 所用，也可以间接地用于要在整个 OSPF 域发布消息的应用（如 RSVP）。非透明 LSA 的功能为 OSPF 的将来提供可扩展性。下面一小节内容直接摘自 RFC 2370:

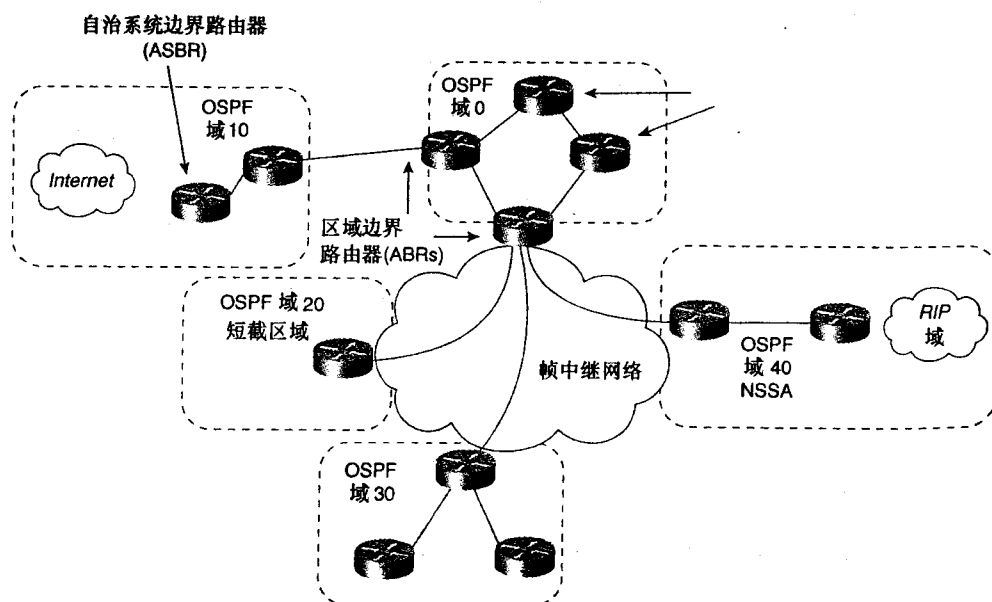
3.0 The Opaque LSA

Opaque LSAs are types 9, 10 and 11 link-state advertisements. Opaque LSAs consist of a standard LSA header followed by a 32-bit aligned application-specific information field. Standard link-state database flooding mechanisms are used for distribution of Opaque LSAs. The range of topological distribution (i.e., the flooding scope) of an Opaque LSA is identified by its link-state type. This section documents the flooding of Opaque LSAs.

The flooding scope associated with each Opaque link-state type is defined as follows.

Link-state type 9 denotes a link-local scope. Type-9 Opaque LSAs are not flooded beyond the local (sub)network. Link-state type 10 denotes an area-local scope. Type-10 Opaque LSAs are not flooded beyond the borders of their associated area. Link-state type 11 denotes that the LSA is flooded throughout the Autonomous System (AS). The flooding scope of type-11 LSAs are equivalent to the flooding scope of AS-external (type-5) LSAs. Specifically type-11 Opaque LSAs are 1) flooded throughout all transit areas, 2) not flooded into stub areas from the backbone and 3) not originated by routers into their connected stub areas. As with type-5 LSAs, if a type-11 Opaque LSA is received in a stub area from a neighboring router within the stub area the LSA is rejected.

图 12-6 给出了一个现代 OSPF 网络的模型，图中标出了路由器的不同类型。



12.1.8 OSPF 的确认信号

为确保 LSA 的正确传输，OSPF 要求对每一个 LSA 都进行确认。LSA 的确认类型包括：

- 非显式确认 (implicit acknowledgment) —— 发送路由器从邻居路由器接收到 LSA 副本时会出现这种情况。路由器检查邻居路由器的 LSA 报告，可以“含蓄地”知道它接收到了 LSA。
- 显式确认 (Explicit acknowledgment) —— 要求接收路由器发送特定的链路状态确认数据包对 LSA 做出响应。

要确保 LSA 是最新的和合法的，每个 LSA 都需要有一个序列号，一个校验位以及一个 MaxAge 值。序列号和校验位确保了 LSA 的合法性，而期限参数则能够保证 LSA 是最新的当前 LSA。MaxAge 是用来检查 LSA 已经生成时间，通常最大值为 3600 秒，即 1 个小时。路由器生成 LSA 时，MaxAge 设置为 0。每次发送 LSA 到另一台路由器时，MaxAge 会加上一个称为 InfTransDelay 的时间参数，这个参数的默认值为 1。当 LSA 达到 MaxAge 值时，这些 LSA 就会在整个网络中重新进行传输。路由器还会再比较两个 LSA 谁最新，这个时候会用到这个 MaxAge 参数。这样的 LSA 数据扩散传输在大型网络上来说有可能有点过度了，不太必要。从 Cisco IOS 12.1 开始，Cisco 引入称为减少 LSA 发送的概念。后面会更详细地讨论 LSA 发送控制方面的问题。

12.1.9 OSPF 的路径类型

在 OSPF 上执行 show ip route 命令时，结果中的所有路由都是按照 OSPF 的 6 种路径类型分类的。路径前面是表明路由类型的标记。这些路由类型及其对应的标记如下：

- (O) 区域内路径/路由——处在同一 OSPF 区域中的路由。
- (O IA) 区域间路径/路由——处在不同的 OSPF 区域，但在同一自治系统中的路由。
- (O E1) 外部类型 1 路径/路由——外部路由重分布进 OSPF 时，必须为它分配一个度量或代价。类型 1 的路径代价是该外部路径/度量的代价再加上 ASBR 报告此路由的内部路径代价之和。
- (O E2) 外部类型 2 路径/路由——和类型 1 一样，只是这类路由的默认代价不会加上到 ASBR 的内部路由代价。默认情况下，所有分配到 OSPF 的路由都成为外部类型 2 路由。可以在重分布时进行更改。
- (O N1) OSPF NSSA 类型 1——外部路由重分布进 OSPF 的 NSSA 区域时，会成为这一类型。类型 1 路径的代价是该外部路径代价加上报告此路由的路由器的内部路径代价。
- (O N2) OSPF NSSA 类型 2——和类型 1 一样，只是不加上内部路由的代价。默认情况下，所有分配到 OSPF 的 NSSA 区域的路由都会成为 OSPF NSSA 类型 2 路由，可以在重分布时进行更改。

例 12-2 列出了较复杂的含有这 4 种类型 OSPF 路由的路由表。

例 12-2 复杂的 OSPF 路由表

```
skynet#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.128.1 to network 0.0.0.0

 10.0.0.0/24 is subnetted, 1 subnets
O      10.10.10.0 is a summary, 03:05:49, Null0
 129.201.0.0/24 is subnetted, 1 subnets
O E1   129.201.1.0 [110/90] via 172.16.2.66, 03:05:45, TokenRing1
 128.200.0.0/24 is subnetted, 1 subnets
D EX   128.200.1.0 [170/679936] via 172.16.192.3, 05:42:57, Serial1
 129.200.0.0/24 is subnetted, 1 subnets
O E1   129.200.1.0 [110/90] via 172.16.2.66, 03:05:45, TokenRing1
 128.201.0.0/24 is subnetted, 1 subnets
D EX   128.201.1.0 [170/679936] via 172.16.192.3, 05:42:57, Serial1
C      201.201.101.0/24 is directly connected, Loopback0
O E2   132.31.0.0/16 [110/2] via 172.16.2.66, 00:58:04, TokenRing1
O E2   131.31.0.0/16 [110/2] via 172.16.2.66, 00:58:04, TokenRing1
 172.16.0.0/16 is variably subnetted, 27 subnets, 4 masks
O IA   172.16.152.0/24 [110/71] via 172.16.2.66, 03:05:45, TokenRing1
O IA   172.16.150.0/24 [110/80] via 172.16.2.66, 03:05:45, TokenRing1
O IA   172.16.151.0/24 [110/71] via 172.16.2.66, 03:05:45, TokenRing1
C      172.16.144.0/21 is directly connected, Loopback20
C      172.16.136.0/21 is directly connected, Ethernet1
C      172.16.128.0/21 is directly connected, Ethernet0
C      172.16.220.0/24 is directly connected, Loopback69
C      172.16.192.0/24 is directly connected, Serial1
C      172.16.192.3/32 is directly connected, Serial1
O IA   172.16.42.2/32 [110/70] via 172.16.2.66, 03:05:46, TokenRing1
O IA   172.16.42.3/32 [110/70] via 172.16.2.66, 03:05:46, TokenRing1
O E2   172.16.42.0/24 [110/2] via 172.16.2.66, 03:05:46, TokenRing1
O IA   172.16.42.1/32 [110/6] via 172.16.2.66, 03:05:46, TokenRing1
O IA   172.16.21.0/24 [110/76] via 172.16.2.66, 03:05:46, TokenRing1
O IA   172.16.22.0/24 [110/71] via 172.16.2.66, 03:05:46, TokenRing1
O E2   172.16.1.0/24 [110/2] via 172.16.2.66, 03:05:46, TokenRing1
O E2   172.16.2.0/24 [110/2] via 172.16.2.66, 03:05:46, TokenRing1
D      172.16.102.0/24 [90/679936] via 172.16.192.3, 05:42:59, Serial1
D      172.16.103.0/24 [90/409600] via 172.16.128.1, 05:42:59, Ethernet0
O E2   172.16.84.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2   172.16.85.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2   172.16.81.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2   172.16.82.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2   172.16.83.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
O E2   172.16.84.0/24 [110/2] via 172.16.2.66, 03:05:47, TokenRing1
<<<text omitted>>>
```

本节关于 OSPF 技术方面的内容是为了向大家灌输 OSPF 的基础知识，以便更深入地理解后面的 OSPF 配置命令。还有一些很好的书籍更详细深入地讲述了 OSPF 的数据包结构以及其他 OSPF 的内容。如果想更好的学习这一协议，推荐参考 Jeff Doyle 的《*Routing TCP/IP, Volume I*》，Tom Thomas 的《*OSPF Network Design Solutions*》以及 John Moy 的《*Anatomy of an Internet Routing Protocol*》。Cisco 在其网站 www.cisco.com 上发布的 OSPF 设计指南以及 RFC

2328 也都是很好的参考资料。

12.2 配置 OSPF

和其他路由选择协议不同，OSPF 在实施之前需要进行一定的预设计，将 OSPF 网络作为一个整体而不仅是一个区域来仔细地加以考虑。下面是部署设计 OSPF 时应该加以考虑的要点：

- **区域设置**——Area 0 必须连续，应该位于整个网络中最为稳定的地方，通常包含核心路由器。
- **路由器 ID**——静态设置路由器 ID (RID)。可以用私有子网 192.168.0.0 来做到这一点。请记住，地址最高的路由器会成为 DR 和 BDR。12.0 以及以上版本的 Cisco IOS 可以静态地设置 RID，而不再需要配置环路接口。
- **用 RID 和优先级来“硬性”选举 DR、BDR**——在 OSPF 需要选举 DR 和 BDR 的地方，运用 OSPF 优先级或路由器 ID 来强制指定 DR 和 BDR。帧中继网络的 DR 应该是对其所在区域内的所有邻居路由器都有 PVC，可以与之直连的路由器。在 LAN 中，DR 和 BDR 应该是最高端的路由器。
- **区域内连续的 IP 寻址**——只要可能，区域中的所有地址应该连续，这样能够确保路由汇总和合理的分层设计。
- **不同形式的存根区域**——很多边缘路由器和帧中继网络都能很好地应用存根区域。尽量使用不同形式的存根区域。
- **尽量避免虚链路的出现**——尽管还将要讨论虚链路的内容，但是这些虚链路的配置和存在却表明网络的设计很差。有些情况下备份链路可能不是直接和 Area 0 相连接，这就需要虚链路。总的来说，实际网络中应该尽量避免虚链路。

记住这些设计要点，按照下面这 7 个步骤对 OSPF 进行设置：

第 1 步 根据上面提及的设计标准，将 OSPF 网络划分成不同的区域

绘出网络示意图，标出 Area 0 和其他的区域以及区域各自的类型。如果网络要使用 DR/BDR，标出 DR/BDR 的路由器。

第 2 步 (可选) 为网络分配固定的路由器 ID (RID)

如果没有 12.0 以上版本的 Cisco IOS，配置环路接口以设置静态 RID。环路接口应该位于很高的专用地址空间中，而且不需要 OSPF 的访问。建议在环路接口 0 上使用 192.168.x.x 范围的地址。如果 OSPF 网络还要选举 DR/BDR，就像在以太网网络中那样，为要选为 DR/BDR 的路由器分配更高的地址，如 192.168.250.251 和 192.168.250.250。在 12.0 及以上版本的 Cisco IOS 中，可以用 OSPF 路由器命令 `router-id ip_address` 来进行 RID 的分配。注意：在分配 RID 之前必须先启动 OSPF 进程。

第 3 步 在路由器上启动 OSPF 并配置 RID

通过全局命令 `router ospf process_id` 来实现的。我个人通常将进程 ID (process_id) 看成自治系统 ID。自治系统中所有路由器上的 ID 应该一致 (译者注：作为一个良好的设计原则，在整个自治系统网络中应该尽量保持 process_id 的一致，但 process_id 的作用不真正等同 IGRP/EIGRP 中的自治系统 ID，路由器间不同的 process_id 也可以相互传输 OSPF 信息，

但一个路由器上不同的 OSPF 的 process_id 之间不直接通信，需要重分布)。启动 OSPF 后就可以用 `router-id ip_address` 命令配置第 2 步分配的 RID 了。

第 4 步 配置要参与 OSPF 网络活动的接口

OSPF 使用下面的网络命令配置 OSPF 接口，以反向掩码和区域 ID 作为命令参数：

```
network a.b.c.d wildcard_mask area X
```

反向掩码可以看做对掩码位取反，“0”是“检查”位，而“1”是“忽略”位。例如，如果只想在 Area 0 的 128.10.1.0/24 到 128.10.255.0/24 的网络上运行 OSPF，命令句法就是：

```
network 128.10.0.0 0.0.255.255 area 0
```

再举一个例子，要在 Area 0 的网络 172.16.128.4/30 上运行 OSPF，命令就应该是：

```
network 172.16.128.4 0.0.0.3 area 100
```

用 0.0.0.0 的通配符可以允许单个接口上 OSPF 的运行。配置 OSPF 时，应尽量使用紧凑的通配符，这样能够避免 OSPF 发送设计者未意识到的网络，也能防止不必要的 OSPF hello 数据包进入这些网段。

第 5 步 OSPF 邻居路由器的配置

在某些情况下，OSPF 网络可能需要额外配置以建立 OSPF 邻接关系。在 NBMA 网络（例如帧中继网络）中，OSPF 的配置与接口相关。表 12-3 给出了常见的网络类型以及所需的附加配置。

表 12-3

OSPF 网络的配置表

物理接口类型	默认 OSPF 网络类型	静态邻接关系	DR/BDR 选举	期望邻接状态	推荐优先级
广播介质以太网、令牌环等等	广播*	无	是	FULL/DR FULL/BDR	是
帧中继点到点	点到点	否	否	FULL/-	否
普通帧中继或点到多点	NBMA	是	是	FULL/DROTHER	否
帧中继点到多点	点到多点**	否	否	FULL/-	否

* 如果帧中继网络上的 OSPF 网络类型改变为 BROADCAST，就应该进行 DR 和 BDR 选举并设置优先级。

** OSPF 的点到多点网络类型不是任何接口的默认网络类型，该网络类型应静态配置。

在帧中继多点接口上配置 OSPF 时，有必要配置静态邻接路由器，否则路由器的邻接路由器会陷入持续不断的等待状态中，无法形成邻接关系。配置静态邻接路由器的 OSPF 路由器命令如下：

```
Router (config-router) #neighbor ip_address_of_neighbor
```

多点网络的中心路由器，或者具有通往每个节点的 PVC 的路由器都应静态配置为 DR。为此，可以将远端路由器的优先级设为 0。优先级为 0 说明该接口或邻居路由器不参与 DR/BDR 选举。DR 选举还可以通过将路由器所在的链路的优先级设为一个大数（如 255）来实现。下面的接口命令可以设置优先级：

```
Router (config-if) #ip ospf priority 0-255
```

OSPF 的默认优先级为 1。

另一个强制建立邻接关系的方法是将 OSPF 网络类型改成更为理想的类型。将帧中继多点网络改为点对点网络会使 OSPF 将多点网络当成多个点对点网络。将网络类型改为广播

类型也能强制建立邻接关系和 DR/BDR 的选举。下面这条命令可以改变 OSPF 的网络类型：

```
Router (config-if) #ip ospf network [broadcast | non-broadcast |
point-to-multipoint | point-to-point]
```

第 6 步 (可选) 配置 OSPF 的特殊区域类型

配置存根、NSSA 以及完全存根的 OSPF 区域类型，可以采用下面这条命令：

```
Router (config-route) #area x [nssa | stub | virtual-link ] [no-summary]
```

在 area 命令后面加上相应的参数就可以将区域配置成 Stub 或 NSSA 类型。例如，将 Area 10 配置成存根区域，命令为：

```
Router (config-route) #area 10 stub
```

完全存根区域的配置是普通存根区域声明后面加上 no-summary 声明，以阻止类型 3 和 4 的 LSA 进入本区域，命令是：

```
Router (config-route) #area 10 stub no-summary
```

第 7 步 (可选) 可选的 OSPF 参数配置

这些参数包括 hello 计时器、路由汇总、认证等。后面还将讨论这些参数。

12.2.1 实例：在帧中继中配置多 OSPF 区域的类型

下面通过实际应用来更好地理解这个较长的配置过程。

图 12-7 给出一个处在配置过程第 1 步的 OSPF 网络。该 OSPF 网络模型中含有帧中继多

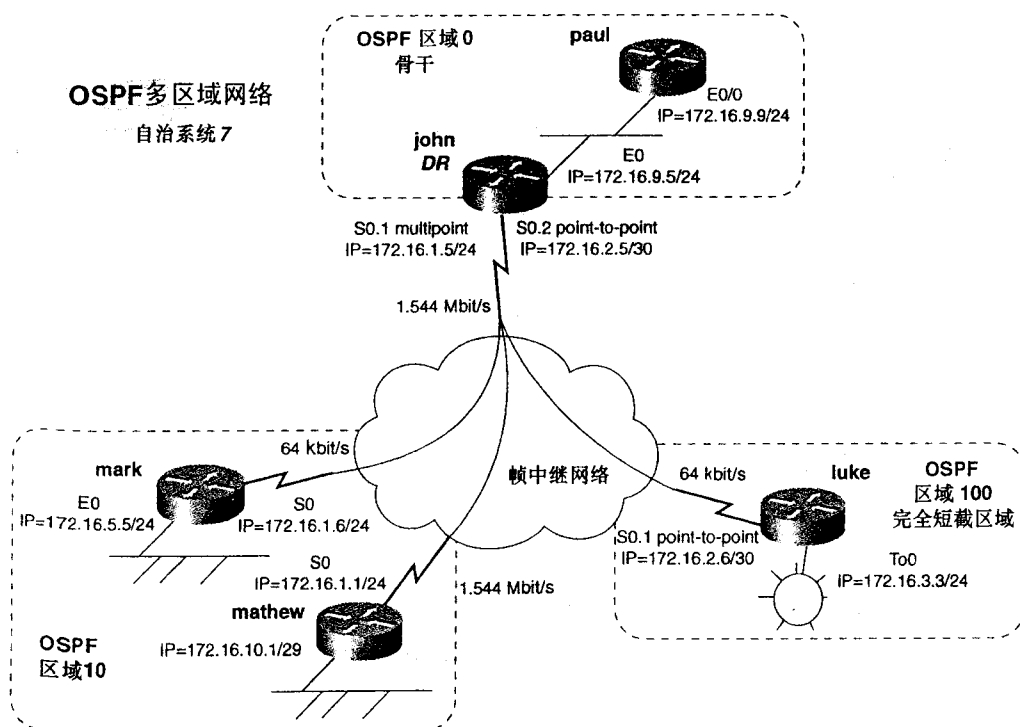


图 12-7 OSPF 的多区域网络

点和点对点网络。要为路由器 luke 配置完全存根区域 Area 100。路由器 paul 要和路由器 john 的以太网接口一起位于骨干区域 Area 0。而路由器 mark 和 mathew 则要放置在 Area 10 中。路由器 john 是 WAN 和 LAN 网络中选出的 DR。

第 2 个步骤是进行 RID 的设置。在这个网络中，只有 mathew 路由器使用 12.0 以前版本的 Cisco IOS，因此，对此路由器要采用一个环路地址来配置其 RID。OSPF 会将最高的环路地址作为 RID，这就是为什么把这一步骤放在启动 OSPF 进程之前的原因。mathew 路由器上配置静态 ID 的命令为：

```
mathew (config) #int loop 0
mathew (config-if) #ip address 172.16.250.1 255.255.255.0
```

图 12-8 列出了该模型使用 RID 的情况。要注意的是，这个步骤是可选的，但是为了网络的稳定性，还是建议大家进行这一步的操作和配置。

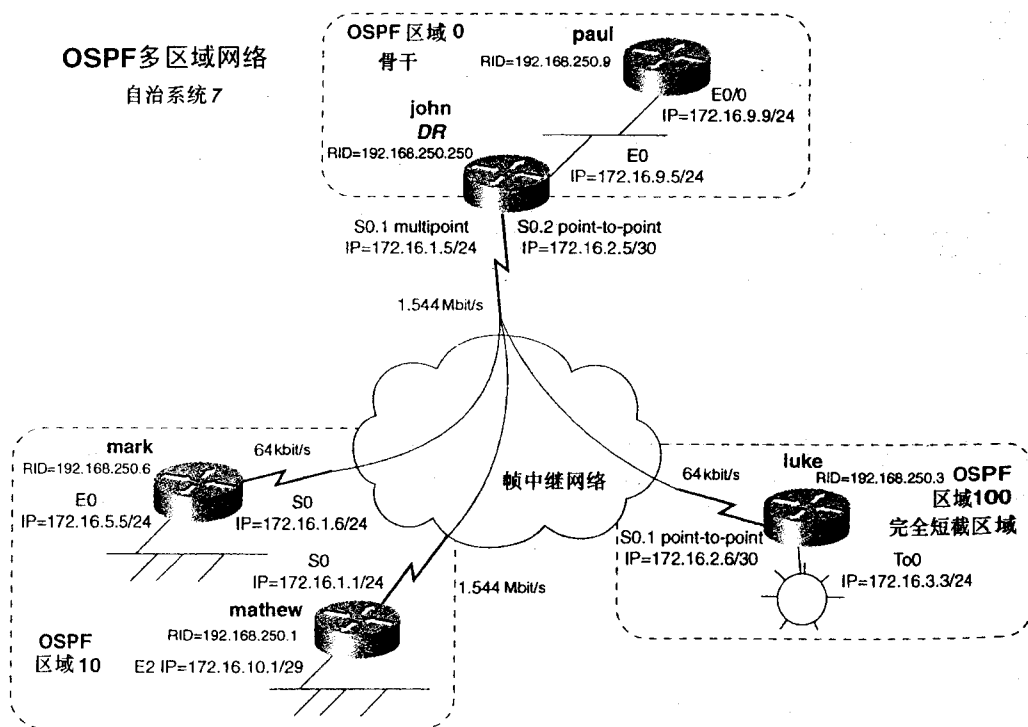


图 12-8 OSPF 的静态 RID

技巧 实验室环境下配置 OSPF 时，分配的 RID 后 8 位比特就是路由器的编号。例如，有 R1、R2 和 R3 三个路由器，分配的 RID 就是 192.168.250.1、192.168.250.2 和 192.168.250.3。实验中的 DR 具有很高的 RID (250) 以便它从其他路由器中选举出来。当查看 OSPF 数据库时，如果 RID 是“自我标识”，会非常有利于 OSPF 的网络结构清晰易懂。

第 3 个步骤包括启动 OSPF 进程以及设置 RID。OSPF 的自治域是 7，因此建议将 7 作为 OSPF 进程 ID。这个步骤中，要用路由器命令 `router-id ip_address` 设置 RID。例 12-3 给出了 mark 和 john 路由器上该步骤的配置示例。

例 12-3 在 mark 和 john 上启动 OSPF

```
mark(config)#router ospf 7
mark(config-router)#router-id 192.168.250.6

john(config)#router ospf 7
john(config-router)#router-id 192.168.250.250
```

在所有的路由器上启动 OSPF 后，第 4 个步骤要求对参与 OSPF 的接口或网络进行配置。用路由器命令 **network ip_address wildcard_mask area x** 可以定义要运行 OSPF 的网络并指定这些网络所在的区域。例 12-4 是在路由器 john 上使用该 **network** 命令的示例。

例 12-4 在路由器 john 上配置 OSPF

```
john(config)#router ospf 7
john(config-router)#network 172.16.9.0 0.0.0.255 area 0
john(config-router)#network 172.16.1.0 0.0.0.255 area 10
john(config-router)#network 172.16.2.4 0.0.0.3 area 100
```

在其他路由器上有很多不同的配置 **network** 命令的方法。建议用带有反向掩码的 **network** 命令作一些限制，将每条语句限制为只针对一个网络，或者用 0.0.0.0 作为反向掩码将声明限制到只用于一个接口。如果希望在大型网络中用一条简单的 **network** 语句将多个接口分配到同一个 OSPF 区域，这样的配置方式会得到较低的运行效率。但是，如果需要将新接口加入到不同的区域内，就必须先删去 **network** 语句，然后再输入 **network** 语句才会生效。

现在，要在 john 和 luke 之间的帧中继的点对点网络上建立邻接关系。另一个邻接关系应该建立在 john 和 paul 之间的以太网上。但是如果没有额外配置，OSPF 不能在 john、mark 和 marthew 之间的帧中继多点网络上建立邻接关系。可以在路由器 john 上用 **show ip ospf neighbor** 命令查看邻接关系的信息，如例 12-5 所示。后面将讨论 **show** 命令和 **debug** 命令。

例 12-5 在路由器 john 上执行 show ip ospf neighbor 命令

```
john#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.250.9	1	FULL/BDR	00:00:34	172.16.9.9	Ethernet0
192.168.250.3	1	FULL/	00:00:38	172.16.2.6	Serial0.2

```
john#
```

从上面命令的结果中可以看到路由 ID 的显示非常有条理。这对任何大规模网络的 OSPF 故障排除工作都非常有帮助。

第 5 个步骤配置额外的邻居路由器信息，以解决 john、mark 和 mathew 之间的邻接关系。这里花了一些时间来静态设定所有路由器的 RID，路由器 john 选为所连接的 WAN 和 LAN 的 DR。在定义 **neighbor** 声明时可以再进一步。要使路由器 john 和 mark 与 mathew 建立起邻接关系，需要为所有的路由器加上 **neighbor** 声明。通过将其关联接口的 OSPF 优先级设为 0，迫使路由器不选为 DR/BDR。选举。多点网络中，只有具有能通往所有远程地址的路由器才能

参加 DR/BDR 选举。指向 john 的路由器 mark 和 mathew 的 **neighbor** 声明中默认优先级 (1) 就足够了。例 12-6 分别给出了 john, mark 和 mathew 上 OSPF 配置的相关部分。

例 12-6 路由器 john, mark 和 mathew 的 OSPF 配置

```
hostname john
!
interface Serial0.1 multipoint
 ip address 172.16.1.5 255.255.255.0
 no ip directed-broadcast
 ip ospf priority 255      --Set this router's priority to 255, forcing the DR
 frame-relay map ip 172.16.1.6 121 broadcast
 frame-relay map ip 172.16.1.1 111 broadcast
!
interface Serial0.2 point-to-point
 ip address 172.16.2.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 150
!
router ospf 7
 router-id 192.168.250.250
 network 172.16.1.0 0.0.0.255 area 10
 network 172.16.2.4 0.0.0.3 area 100
 network 172.16.9.0 0.0.0.255 area 0
 neighbor 172.16.1.1      --A neighbor priority of 0 will not
 neighbor 172.16.1.6      --be listed in the configuration
!

hostname mark
!
interface Serial0
 ip address 172.16.1.6 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 ip ospf priority 0      --This router will not participate in DR/BDR election
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 102 broadcast
 frame-relay map ip 172.16.1.1 102 broadcast
!
router ospf 7
 router-id 192.168.250.6
 network 172.16.1.0 0.0.0.255 area 10
 network 172.16.5.0 0.0.0.255 area 10
 neighbor 172.16.1.5
!

hostname mathew
!
interface Serial0
 ip address 172.16.1.1 255.255.255.0
 encapsulation frame-relay
 ip ospf priority 0      --This router will not participate in DR/BDR election
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 110 broadcast
!
router ospf 7
 network 172.16.1.0 0.0.0.255 area 10
 network 172.16.10.0 0.0.0.255 area 10
 neighbor 172.16.1.5
!
```

通过加入 **neighbor** 语句，OSPF 已经在帧中继多点网络上建立起了邻接关系，通过在路由器 john 上执行 **show ip ospf neighbor** 命令可以证实这一点，如例 12-7 所示。

例 12-7 路由器 john 上的 show ip ospf neighbor 命令示例

```
john#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.250.9	1	FULL/BDR	00:00:32	172.16.9.9	Ethernet0
172.16.250.1	0	FULL/DROTHER	00:01:55	172.16.1.1	Serial0.1
192.168.250.6	0	FULL/DROTHER	00:01:46	172.16.1.6	Serial0.1
192.168.250.3	1	FULL/-	00:00:37	172.16.2.6	Serial0.2

```
john#
```

由于 mathew 和 mark 之间没有链路，因此二者不会形成邻接关系。例 12-8 列出了 mathew 上活动的邻居路由器。邻接关系处在 FULL 状态，john 是链路的 DR。

例 12-8 路由器 mathew 上 show ip ospf neighbor 命令示例

```
mathew#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.250.250	255	FULL/DR	00:01:48	172.16.1.5	Serial0

```
mathew#
```

上面这些也可以不用 **neighbor** 命令来完成。另外一个办法是将 OSPF 网络类型更改成广播类型或者是点对多点类型。就技术而言，点对多点更精确。配置同样的网络模型，现在用点对多点的网络类型来创建 mathew、mark 和 john 之间的邻接关系。例 12-9 列出了 john 和 mathew 上相关的配置信息，用 **ip ospf network type** 来代替 **neighbor** 命令，路由器 mark 的配置和 mathew 完全一样。

例 12-9 配置 john 和 mark 的网络类型

```
hostname john
!
interface Serial0
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
!
interface Serial0.1 multipoint
ip address 172.16.1.5 255.255.255.0
no ip directed-broadcast
ip ospf network point-to-multipoint    --Change the default OSPF network type
to PTM                                --Priority is not needed, no DR/BDR on PTM
frame-relay map ip 172.16.1.6 121 broadcast
frame-relay map ip 172.16.1.1 111 broadcast
!
router ospf 7
router-id 192.168.250.250
```

(待续)

```

network 172.16.1.0 0.0.0.255 area 10
network 172.16.2.4 0.0.0.3 area 100
network 172.16.9.0 0.0.0.255 area 0          --No neighbors
!

hostname mathew
!
interface Serial0
 ip address 172.16.1.1 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 110 broadcast
!
router ospf 7
 network 172.16.1.0 0.0.0.255 area 10
 network 172.16.10.0 0.0.0.255 area 10
!

```

例 12-10 列出了 mathew 和 john 上邻接关系的情况。

例 12-10 路由器 mathew 和 john 上的 OSPF 邻居路由器

```
mathew#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.250.250	1	FULL/-	00:01:35	172.16.1.5	Serial0

```
mathew#
```

```
john#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.250.9	1	FULL/BDR	00:00:36	172.16.9.9	Ethernet0
172.16.250.1	1	FULL/-	00:01:58	172.16.1.1	Serial0.1
192.168.250.6	1	FULL/-	00:01:58	172.16.1.6	Serial0.1
192.168.250.3	1	FULL/-	00:00:38	172.16.2.6	Serial0.2

```
john#
```

OSPF 配置过程的第 6, 7 步都是可选项, 包括 OSPF 区域类型的配置以及其他增强 OSPF 的配置。对于这个模型, 只需再将路由器 luke 设置到完全存根区域中。设置完全存根区域, 首先要将该区域配置成 Stub 区域, 然后再应用参数 **no-summary**。所用的命令格式为:

area 100 stub no-summary

这些命令应该在路由器 john 上运行。例 12-11 列出了 luke 上 **show ip ospf** 命令和 **show ip route** 命令的执行结果。

例 12-11 确认路由器 luke 上的完全存根区域

```
luke# show ip ospf 7
```

```

Routing Process "ospf 7" with ID 192.168.250.3
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs

```

```

Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 0 normal 1 stub 0 nssa
External flood list length 0
  Area 100
    Number of interfaces in this area is 2
    It is a stub area, no summary LSA in this area
    Area has no authentication
    SPF algorithm executed 46 times
    Area ranges are
    Number of LSA 3. Checksum Sum 0x16F14
    Number of opaque link LSA 0. Checksum Sum 0x0
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

luke#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 172.16.2.5 to network 0.0.0.0

    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.2.4/30 is directly connected, Serial0.1
C       172.16.3.0/24 is directly connected, TokenRing0
O*IA 0.0.0.0/0 [110/65] via 172.16.2.5, 00:02:23, Serial0.1
luke#
    
```

路由器 luke 只从 john 接收一个默认路由，而不是完整的路由表，如例 12-12 所示。
例 12-12 给出了该模型中所用的配置过程和路由表的相关部分。

例 12-12 路由器 john、mark、mathew、luke 和 paul 上的配置和路由表

```

!
hostname john
!
interface Ethernet0
 ip address 172.16.9.5 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 multipoint
    
```

(待续)


```

ip address 172.16.1.5 255.255.255.0
no ip directed-broadcast
ip ospf network point-to-multipoint
frame-relay map ip 172.16.1.6 121 broadcast
frame-relay map ip 172.16.1.1 111 broadcast
!
interface Serial0.2 point-to-point
ip address 172.16.2.5 255.255.255.252
no ip directed-broadcast
frame-relay interface-dlci 150
!
router ospf 7
router-id 192.168.250.250
area 100 stub no-summary
network 172.16.1.0 0.0.0.255 area 10
network 172.16.2.4 0.0.0.3 area 100
network 172.16.9.0 0.0.0.255 area 0

john#
john#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

    172.16.0.0/16 is variably subnetted, 8 subnets, 4 masks
C       172.16.9.0/24 is directly connected, Ethernet0
O       172.16.10.0/29 [110/74] via 172.16.1.1, 00:16:06, Serial0.1
O       172.16.5.0/24 [110/74] via 172.16.1.6, 00:16:06, Serial0.1
C       172.16.2.4/30 is directly connected, Serial0.2
O       172.16.1.6/32 [110/64] via 172.16.1.6, 00:16:06, Serial0.1
O       172.16.1.1/32 [110/64] via 172.16.1.1, 00:16:06, Serial0.1
C       172.16.1.0/24 is directly connected, Serial0.1
O       172.16.3.0/24 [110/70] via 172.16.2.6, 00:13:47, Serial0.2

john#
!
hostname mark
!
interface Ethernet0
ip address 172.16.5.5 255.255.255.0
no ip directed-broadcast
!
interface Serial0
ip address 172.16.1.6 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
ip ospf network point-to-multipoint
no ip route-cache
frame-relay map ip 172.16.1.5 102 broadcast
frame-relay map ip 172.16.1.1 102 broadcast
!
router ospf 7
router-id 192.168.250.6
network 172.16.1.0 0.0.0.255 area 10
network 172.16.5.0 0.0.0.255 area 10
!

```

```
mark#
mark#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

```
172.16.0.0/16 is variably subnetted, 8 subnets, 4 masks
O IA 172.16.9.0/24 [110/74] via 172.16.1.5, 00:29:30, Serial0
O     172.16.10.0/29 [110/138] via 172.16.1.5, 00:29:30, Serial0
O     172.16.1.5/32 [110/64] via 172.16.1.5, 00:29:30, Serial0
C     172.16.5.0/24 is directly connected, Ethernet0
O IA 172.16.2.4/30 [110/128] via 172.16.1.5, 00:29:30, Serial0
O     172.16.1.1/32 [110/128] via 172.16.1.5, 00:29:30, Serial0
C     172.16.1.0/24 is directly connected, Serial0
O IA 172.16.3.0/24 [110/134] via 172.16.1.5, 00:15:36, Serial0
```

```
mark#
mark#
```

```
!
hostname mathew
!
interface Loopback0
 ip address 172.16.250.1 255.255.255.0
!
interface Ethernet2
 ip address 172.16.10.1 255.255.255.248
 media-type 10BaseT
!
interface Serial0
 ip address 172.16.1.1 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 110 broadcast
!
```

```
router ospf 7
 network 172.16.1.0 0.0.0.255 area 10
 network 172.16.10.0 0.0.0.255 area 10
!
```

```
mathew#
mathew#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, * - candidate default
        U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
172.16.0.0/16 is variably subnetted, 9 subnets, 4 masks
C     172.16.250.0/24 is directly connected, Loopback0
O IA 172.16.9.0/24 [110/74] via 172.16.1.5, 00:29:44, Serial0
C     172.16.10.0/29 is directly connected, Ethernet2
```

(待续)

```

O      172.16.1.5/32 [110/64] via 172.16.1.5, 00:29:44, Serial0
O      172.16.5.0/24 [110/138] via 172.16.1.5, 00:29:44, Serial0
O IA   172.16.2.4/30 [110/128] via 172.16.1.5, 00:29:44, Serial0
O      172.16.1.6/32 [110/128] via 172.16.1.5, 00:29:44, Serial0
C      172.16.1.0/24 is directly connected, Serial0
O IA   172.16.3.0/24 [110/134] via 172.16.1.5, 00:15:54, Serial0
mathew#
mathew#

!
hostname luke
!
interface Serial0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay
  no ip mroute-cache
  frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
  ip address 172.16.2.6 255.255.255.252
  no ip directed-broadcast
  frame-relay interface-dlci 151
!
router ospf 7
  router-id 192.168.250.3
  area 100 stub no-summary
  network 172.16.2.4 0.0.0.3 area 100
  network 172.16.3.0 0.0.0.255 area 100
!
luke#
luke#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 172.16.2.5 to network 0.0.0.0

    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C      172.16.2.4/30 is directly connected, Serial0.1
C      172.16.3.0/24 is directly connected, TokenRing0
O*IA 0.0.0.0/0 [110/65] via 172.16.2.5, 00:14:56, Serial0.1
luke#
luke#

!
hostname paul
!
interface Ethernet0/0
  ip address 172.16.9.9 255.255.255.0
  no ip directed-broadcast
!
router ospf 7
  router-id 192.168.250.9
  network 172.16.9.0 0.0.0.255 area 0
!
paul#

```

(待续)

```
paul#
paul#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 8 subnets, 4 masks
C       172.16.9.0/24 is directly connected, Ethernet0/0
O IA    172.16.10.0/29 [110/84] via 172.16.9.5, 00:30:32, Ethernet0/0
O IA    172.16.1.5/32 [110/10] via 172.16.9.5, 00:31:27, Ethernet0/0
O IA    172.16.5.0/24 [110/84] via 172.16.9.5, 00:30:42, Ethernet0/0
O IA    172.16.2.4/30 [110/74] via 172.16.9.5, 00:31:27, Ethernet0/0
O IA    172.16.1.6/32 [110/74] via 172.16.9.5, 00:30:42, Ethernet0/0
O IA    172.16.1.1/32 [110/74] via 172.16.9.5, 00:30:32, Ethernet0/0
O IA    172.16.3.0/24 [110/80] via 172.16.9.5, 00:16:41, Ethernet0/0
paul#
```

在进一步讨论 OSPF 的配置问题之前，请先看一下 OSPF 的“Big show”和“Big D”命令。

12.3 OSPF 的“Big show”和“Big D”命令

同多数路由选择协议一样，Cisco 也为 OSPF 提供了很多 **show** 命令和 **debug** 命令。在实际应用中，大部分与 OSPF 有关的问题和信息都可以通过下面这 3 个主要命令来获得：

```
show ip ospf database
show ip ospf neighbors
debug ip ospf adj
```

和 EIGRP 一样，其中最有用却又最容易为人们忽略的是 **show ip ospf neighbors**。

Cisco 还提供了一些命令用于检查 OSPF 数据库以及提供 OSPF 邻接关系的详细信息。

下面是一些非常有用但比较复杂的 **show** 和 **debug** 以及日志命令，后面将进行详细讲述：

```
show ip ospf neighbors [detail | interface_name]
show ip ospf [ process-id area-id] database
show ip ospf interface { interface_type }
show ip route
show ip ospf [ process_id]
debug ip ospf adj
debug ip ospf events
Router (config-router) #log-adjacency-changes
clear ip ospf process
```

12.3.1 show ip ospf neighbors 命令

这条命令在确认 OSPF 的工作状态时最为有用。命令 **show ip ospf neighbor** 能够显示所有邻居路由器的状态以及邻居路由器是否为 DR、BDR 或 DROTHER。要建立邻居路由器，Hello 间隔时间、路由器消亡时间、区域 ID、认证类型和密码都必须匹配。以太网、令牌环和 FDDI 这样的广播型网络可以自动建立邻居路由器，而在 OSPF 的 NBMA 网络中还需要做额外配置建立路由器邻接关系。

参数 **detail** 可以添加在 **show ip ospf neighbor** 命令后面以提供邻居路由器的更详细的信息。这些详细信息包括 OSPF 计时器和选项信息，状态转变计数器以及链路中哪台路由器是 DR、BDR 等。如果执行之后没有信息显示，则意味着没有收到 hello 数据包。如果没有形成邻居路由器，**debug ip ospf adj** 命令通常会显示原因。例 12-13 是该命令在路由器 john 上的输出。

例 12-13 路由器 john 上 show ip ospf neighbor 命令的输出示例

```
john#show ip ospf neighbors
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.250.9	1	FULL/BDR	00:00:37	172.16.9.9	Ethernet0
172.16.250.1	1	FULL/ -	00:01:57	172.16.1.1	Serial0.1
192.168.250.6	1	FULL/ -	00:01:52	172.16.1.6	Serial0.1
192.168.250.3	1	FULL/ -	00:00:35	172.16.2.6	Serial0.2

john#

输出示例中值得注意的字段是：

- **Neighbor ID**——邻居路由器的 RID。
- **Pri**——从本地路由器上看到的邻居路由器的优先级。
- **State**——该字段是邻居路由器的状态以及该邻居路由器是否是 DR，BDR 或 DROTHER。两个正常的状态是 2-way 和 FULL。如果路由器的状态不是这两个，就表明路由器出了问题。状态字段有可能是：
 - **Down**——OSPF 邻居路由器的初始状态，表明从该邻居路由器没有接收到任何信息，但是可以向这一状态下的邻居路由器发送 hello 数据包。如果本地路由器在路由器消亡时间间隔段内（默认情况下，路由器消亡时间 = $4 \times \text{Hello 间隔时间}$ ）没有从某邻居路由器接收到 hello 数据包，则该邻居路由器的状态会从 FULL 变为 DOWN。
 - **Attempt**——该状态只对由邻居路由器命令定义的 NBMA 环境下的路由器有效。Attempt 的意思是路由器在向邻居路由器发送 hello 数据包，但是没有接收到任何返回信息。
 - **Init**——该状态说明路由器接收到邻居路由器发送的 hello 数据包，但是在接收的 hello 数据包中并没有包含自己的 RID。
 - **2-Way**——该状态说明两台路由器之间已经建立了双向通信。双向的意思就是双方路由器都接收到对方的 hello 数据包。邻居路由器是否建立邻接关系取决于这个状态。在广播介质上，各路由器只有与 DR 和 BDR 路由器之间的链路

才能进入 FULL 状态；剩下的所有邻居路由器之间都处在 2-way 状态中。

—— **Exstart**——建立邻接关系的第一个状态，用来选择链路上的主、从路由器。

—— **Exchange 和 loading**——在这些状态下，OSPF 发送链路状态请求数据包以及链路状态更新数据包。

—— **Full**——该状态下，路由器相互之间建立了完全邻接关系。所有的路由器和网络 LSA 都在进行交换，数据库也已经完全同步。

- **Dead time**——路由器没有接收到 hello 数据包时，将此路由器宣布为消亡之前应该等待的时间。
- **[IP] address 和 interface**——这里的 address 是指邻居路由器的物理接口地址，而 interface 是指从邻居路由器接收到 hello 数据包的接口。

12.3.2 show ip ospf database 命令

命令 **show ip ospf database** 用来显示整个 OSPF 数据库，数据库中的每个链路状态及其所处的区域。数据库中使用链路状态的字母名称而不以类型来表示链路状态，像类型 1、类型 2 等。如果 OSPF 检查到某个网络，那么该网络就已经在数据库中。数据库中采用路由器 ID 来识别发送链路状态的各个路由器。例 12-14 显示了该命令在路由器 john 上的执行情况。

例 12-14 路由器 john 上 show ip ospf database 命令的执行示例

```
john#show ip ospf database

OSPF Router with ID (192.168.250.250) (Process ID 7)

Router Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum Link count
192.168.250.9  192.168.250.9 450         0x80000033  0x1370  1
192.168.250.250 192.168.250.250 334        0x8000002A  0xD0DA  1

Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum
172.16.9.5     192.168.250.250 334        0x80000024  0xC14

Summary Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum
172.16.1.1     192.168.250.250 1592       0x8000000B  0x9242
172.16.1.5     192.168.250.250 1850       0x8000000B  0xE729
172.16.1.6     192.168.250.250 1592       0x8000000B  0x606F
172.16.2.4     192.168.250.250 1850       0x8000000B  0x577C
172.16.3.0     192.168.250.250 845        0x8000000B  0xC20B
172.16.5.0     192.168.250.250 1592       0x8000000B  0xD4F2
172.16.10.0    192.168.250.250 1594       0x8000000B  0x7356

Router Link States (Area 10)

Link ID        ADV Router    Age          Seq#          Checksum Link count
172.16.250.1   172.16.250.1 1740       0x80000052  0x6209  3
```

```

192.168.250.6 192.168.250.6 1812 0x80000025 0xE048 3
192.168.250.250 192.168.250.250 1594 0x80000053 0x72A6 3

```

Summary Net Link States (Area 10)

Link ID	ADV Router	Age	Seq#	Checksum
172.16.2.4	192.168.250.250	1595	0x80000030	0xDA1
172.16.3.0	192.168.250.250	848	0x8000000B	0xC20B
172.16.9.0	192.168.250.250	92	0x8000002F	0xDD02

Router Link States (Area 100)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
192.168.250.3	192.168.250.3	694	0x80000051	0x2DA0	3
192.168.250.250	192.168.250.250	848	0x80000039	0x3291	2

Summary Net Link States (Area 100)

Link ID	ADV Router	Age	Seq#	Checksum
0.0.0.0	192.168.250.250	848	0x8000000B	0xD202

john#

输出中值得注意的字段包括：

- **ADV Router**——发送数据包的路由器 ID。
- **Age**——链路状态的存活时间。
- **Seq# and Checksum**——验证链路状态的完整性。
- **Tag**——如果重分布过程中加入了 OSPF 标签，该标签会在输出结果的右面一列显示出来。

12.3.3 show ip ospf interface 命令

OSPF 的一个常见问题就是使用了不正确的网络语句和反向掩码位。最好的验证生效 OSPF 接口配置参数的命令是 **show ip ospf interface**。该命令输出的重要字段有网络类型、区域、进程 ID、计时器以及邻居路由器和邻接关系数，另外还有 DR/BDR 路由器和优先级。例 12-15 是 mark 路由器上该命令的情况。

例 12-15 路由器 mark 上 show ip ospf interface 命令的输出示例

```

mark#show ip ospf interface
Ethernet0 is up, line protocol is up
Internet Address 172.16.5.5/24, Area 10
Process ID 7, Router ID 192.168.250.6, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 172.16.5.5, Interface address 172.16.5.5
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:06
Index 2/2, 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)
Serial0 is up, line protocol is up
Internet Address 172.16.1.6/24, Area 10
Process ID 7, Router ID 192.168.250.6, 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
Hello due in 00:00:05
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 192.168.250.250
Suppress hello for 0 neighbor(s)
mark#
```

12.3.4 show ip route 命令

这条命令显示 IP 转发表或路由表的内容，可以列出 OSPF 路由的 6 种类型：

- (O) ——OSPF 区域内路由，或者是说来自同一区域内的路由。
- (O IA) ——OSPF 区域间路由，或者是说来自另一区域的路由。
- (O N1) ——OSPF NSSA 类型 1。
- (O N2) ——OSPF NSSA 类型 2。
- (O E1) ——OSPF 外部类型 1。
- (O E2) ——OSPF 外部类型 2。

可以参考前面“OSPF 的路径类型”一节中路由类型的详细内容。转发表中路由后面是该路由的管理距离和路由代价。转发表还能列出报告该网络路由的路由器接口和接收时间。

12.3.5 show ip ospf 命令

该命令能给出 OSPF 区域的综合信息，列出区域的类型、认证方式、SPF 计数器、重分布以及详细的 LSA 计时器信息。例 12-16 是该命令的情况。

例 12-16 路由器 john 上 show ip ospf 命令的执行示例

```
john#show ip ospf
Routing Process "ospf 7" with ID 192.168.250.250
Supports only single TOS(TOS0) routes
Supports opaque LSA
It is an area border.router
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
```

(待续)


```
Number of areas in this router is 3. 2 normal 1 stub 0 nssa
External flood list length 0
```

```
Area BACKBONE(0)
```

```
Number of interfaces in this area is 1
Area has no authentication
SPF algorithm executed 11 times
Area ranges are
Number of LSA 10. Checksum Sum 0x5A54A
Number of opaque link LSA 0. Checksum Sum 0x0
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

```
Area 10
```

```
Number of interfaces in this area is 1
Area has no authentication
SPF algorithm executed 35 times
Area ranges are
Number of LSA 6. Checksum Sum 0x40CCF
Number of opaque link LSA 0. Checksum Sum 0x0
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

```
Area 100
```

```
Number of interfaces in this area is 1
It is a stub area, no summary LSA in this area
generates stub default route with cost 1
Area has no authentication
SPF algorithm executed 32 times
Area ranges are
Number of LSA 3. Checksum Sum 0x10A47
Number of opaque link LSA 0. Checksum Sum 0x0
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

```
john#
```

```
Gateway of last resort is not set
```

```
172.16.0.0/16 is variably subnetted, 8 subnets, 4 masks
O IA 172.16.9.0/24 [110/74] via 172.16.1.5, 09:11:03, Serial0
O 172.16.10.0/29 [110/138] via 172.16.1.5, 09:11:03, Serial0
O 172.16.1.5/32 [110/64] via 172.16.1.5, 09:11:03, Serial0
C 172.16.5.0/24 is directly connected, Ethernet0
O IA 172.16.2.4/30 [110/128] via 172.16.1.5, 09:11:03, Serial0
O 172.16.1.1/32 [110/128] via 172.16.1.5, 09:11:03, Serial0
C 172.16.1.0/24 is directly connected, Serial0
O IA 172.16.3.0/24 [110/134] via 172.16.1.5, 08:57:09, Serial0
mark#
```

12.3.6 debug ip ospf adj 和 debug ip ospf events 命令

这是两个 OSPF 的“Big D”命令。两条命令在很大程度上是相同的。命令的输出内容很多，因此实际使用时，最好打开记录功能。该命令的结果非常全面，可以找到大部分 OSPF 常见问题的信息，例如：

- 子网掩码不匹配。
- hello/消亡时间段不匹配。
- 认证密钥不匹配。
- 区域 ID 和类型不匹配。

12.3.7 log-adjacency-changes/show log 命令

和 EIGRP 一样，OSPF 提供了一条特殊的命令来记录邻接关系的变化。使用户无需通过浏览调试结果而了解邻接关系的问题。记录邻接关系的命令句法如下：

```
john (config) #router ospf 7
john (config-router) #log-adjacency-changes
```

例 12-17 列出了丢失邻接关系后的记录情况。

例 12-17 show log 命令记录下一个进入 Down 状态的邻居路由器

```
john#show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level debugging, 1228 messages logged
  Monitor logging: level debugging, 0 messages logged
  Buffer logging: level debugging, 2 messages logged
  Trap logging: level informational, 68 message lines logged

Log Buffer (10000 bytes):

1d00h: %SYS-5-CONFIG_I: Configured from console by console
1d00h: %OSPF-5-ADJCHG: Process 7, Nbr 172.16.250.1 on Serial0.1 from FULL to DOWN, Neighbor Down
john#
```

12.3.8 clear ip ospf process

12.0 以上版本的 Cisco IOS 可以使用该命令来清除所有的 OSPF 邻居路由器、RID 以及 SPF 数据库。这条命令将整个 OSPF 进程初始化，如同路由器复位。

12.4 OSPF 的存根区域配置

3 类 OSPF 存根区域是需要配置的：

- Stub 区域
- NSSA
- 完全 Stub 区域

配置 Stub 区域可以使用 **area** 命令加上要配置的存根区域类型。同一 IP 网络上的所有路由器必须处在同一区域，建立邻居路由器及邻接关系时区域参数必须完全匹配。

例如，可以用下面这条命令将 Area X 配置为存根区域：

```
Router (config-router) #area X stub
```

下面这条命令可以将 Area X 配置成非完全存根区域:

```
Router (config-route) #area X nssa {default-information originate}.
```

在上例中的 area 命令中加入关键字 **default-information originate** 可以将本路由器产生的默认路由发送到 NSSA 区域中。

要把 Area X 配置成完全 Stub 区域，可以用这条命令:

```
Router (config-route) #area X stub no-summary
```

12.5 OSPF 的调整

OSPF 提供一些参数可以用来调整计时器，控制网络广播以及管理路由和链路状态的传播。下面是常见的用来调整 OSPF 的参数语法:

```
Router (config-if) #ip ospf hello-interval interval_in_seconds
```

```
Router (config-if) #ip ospf dead-interval dead_interval_in_seconds
```

```
Router (config-if) #ip ospf retransmit-interval
```

接口命令 **ip ospf hello-interval** 可以改变 OSPF hello 计时器的时间段长度，这个时间段的默认值与接口有关。默认情况下，hello 数据包的计时器长度在广播和点对点网络中是 10 秒，在 NBMA 网络中是 30 秒。网络和区域中所有的邻居路由器应该具有相同的 hello 计时器值。当 hello 时间段改变时，消亡时间段和等待计时器会自动改变。

可以用 **ip ospf dead-interval** 命令来更改某接口上接收到的消亡时间值。该计时器的默认值是 hello 计时器的 4 倍，因此，广播和点对点网络上该计时器是 40 秒，而 NBMA 网络上则是 120 秒。再回想一下，消亡计时器的含义就是如果某邻居路由器在该段时间里一直没有接收到 hello 数据包，即视该路由器处于“消亡”状态。

用 **ip ospf retransmit-interval** 命令可以改变 OSPF 的重传时间间隔。区域中所有的路由器的重传时间间隔长度也应该一致。

12.6 减少 OSPF 的扩散

OSPF 减少扩散的概念是在 Cisco IOS 12.1 (2) T 中引入的。OSPF 的 LSA 周期性地每 3600 秒刷新一次。总的来说，尽管 OSPF 网络相当稳定，但却导致大量不必要的从一个区域涌向另一个区域。就技术角度来看，如果 LSA 没有发生改变，那为什么还要每过 3600 秒又重新发送呢？OSPF 数据发送控制是一项新的技术，将通常的 LSA 转变成了 DoNotAge LSA。这样，LSA 就不会每 3600 秒就在整个区域内重新发送一次。要实现这一特性的功能，区域中与要进行数据发送控制的路由器相连接的所有路由器都必须装有 12.1 (2) T 及以上版本的 Cisco IOS。用下面这条接口命令就可以启动在接口上进行数据发送的控制:

```
Router (config-if) #ip ospf flood-reduction
```

另外，也可以选择禁止所有的 LSA 从某个接口发送。要在广播、非广播和点对点接口上禁止发送 OSPF 的 LSA，可以选用下面这条接口命令:

```
Router (config-if) #ospf database-filter all out
```

```
Router (config-router) #neighbor ip_address database-filter all out
```

12.7 OSPF 重分布和路由控制

为了对 OSPF 中特定的路由更新信息进行过滤，必须使用向内的分布列表。OSPF 路由不用采用传统的方式发送路由更新信息，因此，只有在接收更新信息的路由器上使用向内的分布列表时才有助于路由控制。在将一个协议重分布到另一个去时，可以使用 **redistribute** 命令加上默认度量标准值。在重分布过程中，如果要对特定的路由进行控制，就应该用路由图取代过滤列表进行操作。路由图能够提供很多重分布过程的控制选项，是一个非常有用的路由工具。

12.7.1 用于控制路由过滤和重分布的命令

可以使用下面的命令调用标准访问控制列表来过滤接收到的路由更新信息：

```
Router (config-router) #distribute-list [ 1-99 ] [in] [ interface_name ]
```

可选参数 **in** 是从接口来看的方向。换句话说，要阻止路由更新信息进入某个接口，可以使用 **in** 选项。这条命令只能过滤路由，而不能过滤 LSA。

要将其他路由选择协议重分布到 OSPF 中去，可以采用下面这条命令：

```
Router (config-router) #redistribute [connected | static | bgp | rip | igrp | eigrp | isis]
[ subnets ] [ tag tag_number ] [ metric cost ] [ metric-type { OE1 | OE2 } ] [ route-map ]
```

OSPF 中，如果要将多个网络重分布进 OSPF，一定要使用关键字 **subnets**。如果没有使用这个关键字，路由重分布进 OSPF 时，只有非子网化的路由才会重分布进入 OSPF。在重分布过程中可以添加一个附加的标签。这个标签出现在 OSPF 数据库中，用来迅速判定网络中何处进行了路由重分布。而路由图也可作为附加的路由控制。如果进行重分布的路由选择协议的路由度量与默认度量不同，还可以为这些路由提供可选的度量或是路由代价。OSPF 重分布在默认情况下新生成的路由类型是 OSPF 外部类型 2 路由（O E2）。要将此类型改成 OSPF 外部类型 1 路由（O E1），需在 **redistribution** 命令中加入 **metric-type** 参数。

用下面这条命令可以配置所有重分布进 OSPF 的默认路由代价：

```
Router (config-router) default-metric [cost 1-4294967295]
```

进行重分布时，必须提供默认的度量。常用的度量用 **default-metric 10** 命令来指定。前面讲过，比实际的度量的值更为重要的是保持在整个路由区域中度量的一致性，因此，所有重分布后的路由应该具有相同的路由代价。

12.7.2 用于改变 OSPF 的路由选择的命令

有很多方法可以控制 OSPF 的路由更新信息。前面讲过，OSPF 以一个包含带宽在内的公式为基础来计算到某个目的地址的路由代价。要想改变路由的选择，一种方法是改变接口的带宽值从而影响链路代价，另一种方法是直接改变接口的代价值。

和 EIGRP 一样，OSPF 也支持直接用 **distance** 命令改变管理距离的值。此外，OSPF 还

能够使用被动接口来阻止 hello 数据包发送到链路上。

下面这条命令可以用来设定 OSPF 的接口代价值:

```
Router (config-if) #cost_{cost 1-4294967295}
```

这条命令只用于 OSPF，不会对链路的实际数据流通造成影响。

下面这条命令指定接口的带宽值 (kbit/s):

```
Router (config-if) #bandwidth [ bandwidth_kbit/s 1-4214748364]
```

命令 **bandwidth** 只被路由选择协议用来计算接口代价值，也不会影响链路实际数据的流通。

下面这条命令则可以用来改变 OSPF 路由的管理距离:

```
Router (config-router) #distance ospf {[intra-area [ 1-255] [inter-area [ 1-255] [external [ 1-255]]]
```

OSPF 有 3 种不同的管理距离: 区域内、区域间和外部。区域内部的路由就称为区域内，来自其他区域的路由就称为区域间，而通过重分布进入到区域中的路由则称为外部。每种类型的路由的默认管理距离都是 110。

要阻止 OSPF 的 hello 数据包在链路上发送，可以使用这条命令:

```
Router (config-router) passive-interface interface_name
```

由于抑制了 hello 数据包，因此不会建立邻居路由器，导致不进行路由更新信息的接收和发送。

12.7.3 实例：路由的过滤和重分布

现在把上面的概念应用到实际模型中去以加深理解，要做的是对路由进行过滤和重分布。图 12-9 将例 12-8 中的网络模型作了一些修改。路由器 mark 现在是 OSPF 域和 RIP 域的 ASBR，还将在 paul 路由器上对环路接口进行重分布。路由器 mathew 上要采用向内过滤列表对外部网络或者来自 paul 的环路网络进行过滤。

配置从路由器 paul 开始。这台路由器上添加了两个环路接口: 128.100.1.1/24 和 128.100.2.1/24。要对这些网络进行重分布，可以使用带有 **subnets** 参数的 **redistribute connected** 命令。重分布后的路由的默认度量标准或路由代价值是 10。这个例子的重分布过程中设置了一个标签。例 12-20 表明该标签在 OSPF 的数据库中如何显示。例 12-18 列出了路由器 paul 上的相关配置情况。

例 12-18 路由器 paul 的配置

```
interface Loopback20
 ip address 128.100.1.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback21
 ip address 128.100.2.1 255.255.255.0
 no ip directed-broadcast
!
interface Ethernet0/0
 ip address 172.16.9.9 255.255.255.0
 no ip directed-broadcast
!
<<<text omitted>>>
!
```

(待续)

```

router ospf 7
router-id 192.168.250.9
redistribute connected subnets tag 0 --Redistribute the loopback interfaces
network 172.16.9.0 0.0.0.255 area 0
default-metric 10 --Use a cost of 10 on redistributed networks
!

```

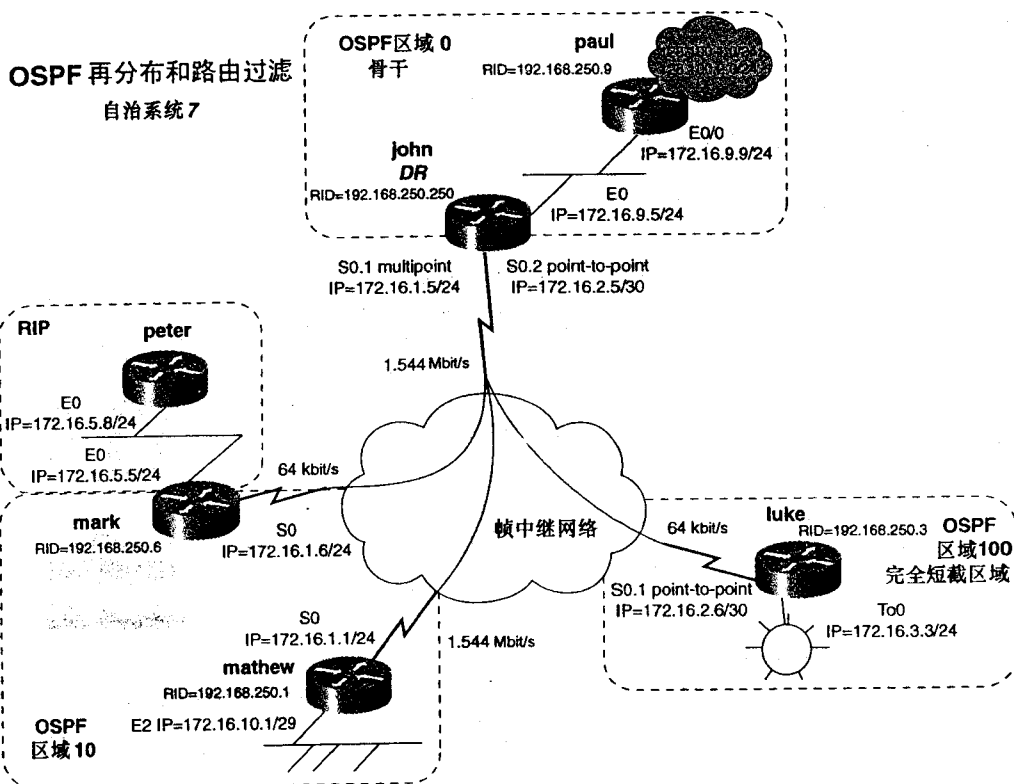


图 12-9 用于路由过滤和重分布的 OSPF 网络

在 mark 路由器上用 **show ip route** 命令能够确认重分布的结果，如例 12-19 所示。

例 12-19 路由器 mark 上的 show ip route

```

mark#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

172.16.0.0/16 is variably subnetted, 8 subnets, 4 masks

(待续)

```

O IA 172.16.9.0/24 [110/74] via 172.16.1.5, 08:58:07, Serial0
O 172.16.10.0/24 [110/138] via 172.16.1.5, 18:33:11, Serial0
O 172.16.1.5/32 [110/64] via 172.16.1.5, 18:33:11, Serial0
C 172.16.5.0/24 is directly connected, Ethernet0
O IA 172.16.2.4/30 [110/128] via 172.16.1.5, 08:58:08, Serial0
O 172.16.1.1/32 [110/128] via 172.16.1.5, 18:33:11, Serial0
C 172.16.1.0/24 is directly connected, Serial0
O IA 172.16.3.0/24 [110/134] via 172.16.1.5, 08:58:09, Serial0
128.100.0.0/24 is subnetted, 2 subnets
O E2 128.100.1.0 [110/20] via 172.16.1.5, 08:58:09, Serial0 →redistributed routes
O E2 128.100.2.0 [110/20] via 172.16.1.5, 08:58:09, Serial0

```

重分布过程中也在路由器上放置了标签 9。例 12-20 列出了 mark 的 OSPF 数据库内容，显示了标签在整个 OSPF 中的传播情况。

例 12-20 路由器 mark 上的 OSPF 数据库

```

mark#show ip ospf database

OSPF Router with ID (192.168.250.6) (Process ID 7)

<<text omitted>>>

Type-5 AS External Link States

Link ID      ADV Router   Age         Seq#         Checksum Tag
128.100.1.0   192.168.250.9 1094       0x80000024 0xDE42 9 →Tag added during
128.100.2.0   192.168.250.9 1095       0x80000024 0xD34C 9 →redistribution
172.16.9.0    192.168.250.9 844        0x80000026 0x3807 9
mark#

```

请注意，进行重分布时，路由 172.16.9.0 也被重分布，这是由于以太网被认为是一个本地网络。为了避免这种情况的发生，给 **redistribution** 命令加上路由图就可以过滤任何不需要的网络。

接下来在路由器 mark 上将 RIP 域集成到 OSPF 中。要做到这一点，首先在 mark 上启动 RIP 的运行。这个时候，对于 OSPF，将 mark 的 E0 接口置为被动状态，对 RIP，则将 mark 的 S0 接口也置为被动状态。用一个更为明确的 **network** 命令即可做到，这也是为什么要在 **network** 命令中将反向掩码位限制到单一网络或接口的原因。在 mark 上用 **redistribute rip subnets** 命令可以进行 RIP 的重分布。同样，还需要将 OSPF 重分布进 RIP。用于 OSPF 的默认度量是 10。例 12-21 是 mark 上 OSPF 的配置。

例 12-21 路由器 mark 上的路由选择协议配置

```

router ospf 7
router-id 192.168.250.6
redistribute rip subnets →redistribute RIP into OSPF
passive-interface Ethernet0 →No OSPF hellos are to enter E0 (optional)
network 172.16.1.0 0.0.0.255 area 10
default-metric 10 →Use 10 as the cost to the RIP domain
!
router rip

```

```

redistribute ospf 7          -redistribute OSPF into RIP
passive-interface Serial0    -No RIP broadcasts out S0
network 172.16.0.0
default-metric 3             -Use a hop count of 3 for OSPF routes

```

由于网络中仅有一个重分布点，因此不必担心会出现重分布环路或“路由反馈”的问题。在完成重分布之前，必须对网络 172.16.2.4/30 进行汇总，这是由于 RIP 是通过一个 24 位界的接口来接收路由的。这个模型中，汇总通过 **area range** 命令来完成。下一节还将专门讨论这方面的问题。在这个模型里验证重分布的最佳办法就是查看路由器 peter 上的路由表，可以用 ping 确认上面所有的路由。例 12-22 为 peter 的路由表。

例 12-22 重分布之后路由器 peter 上的路由表

```

peter#show ip route
<<<text omitted>>>

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
R       172.16.9.0/24 [120/3] via 172.16.5.5, 00:00:01, Ethernet0
R       172.16.10.0/24 [120/3] via 172.16.5.5, 00:00:01, Ethernet0
R       172.16.1.5/32 [120/3] via 172.16.5.5, 00:00:01, Ethernet0
C       172.16.5.0/24 is directly connected, Ethernet0
R       172.16.1.1/32 [120/3] via 172.16.5.5, 00:00:02, Ethernet0
R       172.16.1.0/24 [120/1] via 172.16.5.5, 00:00:02, Ethernet0
R       172.16.2.0/24 [120/3] via 172.16.5.5, 00:00:02, Ethernet0
R       172.16.3.0/24 [120/3] via 172.16.5.5, 00:00:02, Ethernet0
R       128.100.0.0/16 [120/3] via 172.16.5.5, 00:00:02, Ethernet0
peter#

```

在 mathew 路由器上利用输入过滤表对 OSPF 的路由进行过滤。路由器 mathew 的配置和下面的这个例子很相似。

例 12-23 路由器 mathew 上的过滤列表

```

router ospf 7
network 172.16.1.0 0.0.0.255 area 10
network 172.16.10.0 0.0.0.255 area 10
distribute-list 10 in Serial0      ←distribute list applied to s0
!
ip classless
!
access-list 10 deny 128.100.0.0 0.0.255.255 ←access list deny all 128.100.x.x
                                              networks
access-list 10 permit any
!

mathew#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, * - candidate default
        U - per-user static route, o - ODR

```

(待续)


```
Gateway of last resort is not set
```

```

172.16.0.0/16 is variably subnetted, 9 subnets, 2 masks
C    172.16.255.0/24 is directly connected, Loopback0
O IA 172.16.9.0/24 [110/74] via 172.16.1.5, 01:01:06, Serial0
C    172.16.10.0/24 is directly connected, Ethernet2
O    172.16.1.5/32 [110/64] via 172.16.1.5, 01:01:06, Serial0
O    172.16.5.0/24 [110/129] via 172.16.1.5, 01:01:06, Serial0
O    172.16.1.6/32 [110/128] via 172.16.1.5, 01:01:06, Serial0
C    172.16.1.0/24 is directly connected, Serial0
O IA 172.16.2.0/24 [110/128] via 172.16.1.5, 01:01:06, Serial0
O IA 172.16.3.0/24 [110/134] via 172.16.1.5, 01:01:06, Serial0
mathew#

```

例 12-23 的最后部分是 mathew 新的路由表，已经没有了关于网络 128.100.1.0/24 和 128.100.2.0/24 的路由。

12.8 OSPF 的汇总功能

OSPF 有两种汇总形式。一种是将其他路由选择协议重分布进 OSPF 时对路由进行的汇总。另一种是对一个区域的汇总。这两种汇总方式都创建汇总 LSA 并发送到 Area 0，骨干区域又会将链路状态发送到其他区域。进行 OSPF 汇总配置时要注意以下事项：

- OSPF 区域中的地址空间应该连续。这样能够使 ABR 上的汇总容易进行。将很多网络汇总到一条宣告中可以减小路由表的规模，改善 OSPF 的整体性能和可扩展性。
- 尽量在主网边界或简单的 8 位边界上进行汇总。如果网络中含有有类路由选择协议，如 RIP 或 IGRP，必须在有类路由选择协议可以接收到的位边界上进行汇总。
- 不能在主干区域进行汇总。所有的汇总结果都发送到 Area 0，而后又从这一点发送出来。因此，Area 0 不能进行汇总。

对外部路由或者是重分布进 OSPF 的路由进行汇总，在 ASBR 上使用下面这条命令：

```
Router(config-router)#summary-address network_address network_mask [tag tag_number]
```

参数 tag 的用法与在重分布过程中一样，即将路由用数字作标记。

要汇总从一个 OSPF 区域进入到 Area 0 的路由，使用下面这条路由器命令：

```
Router(config-router)#area area_id range network_address network_mask
```

图 12-10 中再次修改了网络模型，改变了一些接口，增加了 RIP 域中的网络数量。OSPF 的 Area 20 现在包括子网 100.10.1.0/24 到 100.10.3.0/24。首先，把这些网络汇总到一个路由 100.10.0.0/16 中，可以通过在 ABR（路由器 john）上执行路由器命令 **area 20 range 100.10.0.0 255.255.0.0** 来完成。路由器 john 会将汇总路由 100.10.0.0/16 转发给 mark。这称为区域间汇总。

为了实现这个网络与 RIP 域的完全 IP 互连，还需要进行另一种形式的区域间汇总。RIP 以太接口的 IP 地址在一个 24 位界上。第 9 章“距离矢量协议：路由信息协议版本 1 和版本 2 (RIP-1 和 RIP-2)”中讲过，如果要接收路由，RIP 必须处在自然的位边界 8、16 或 24 上，该例中就是一个 24 位边界。这对 IGRP 也一样，OSPF 域中所有的网络都通过路由器 john 和

luke 之间的帧中继点对点网络) 都处在 24 位边界上。为了与 RIP 相适应, 帧中继点对点网络也必须汇总到 24 位边界上。要将 172.16.2.4/30 网络汇总到 172.16.2.0/24, 应在 john 上使用 **area 100 range 172.16.2.0 255.255.255.0** 命令。路由器 john 现在会将路由 172.16.2.0/24 转发给 mark 并最终到达 peter。例 12-24 为是 mark 的路由表。

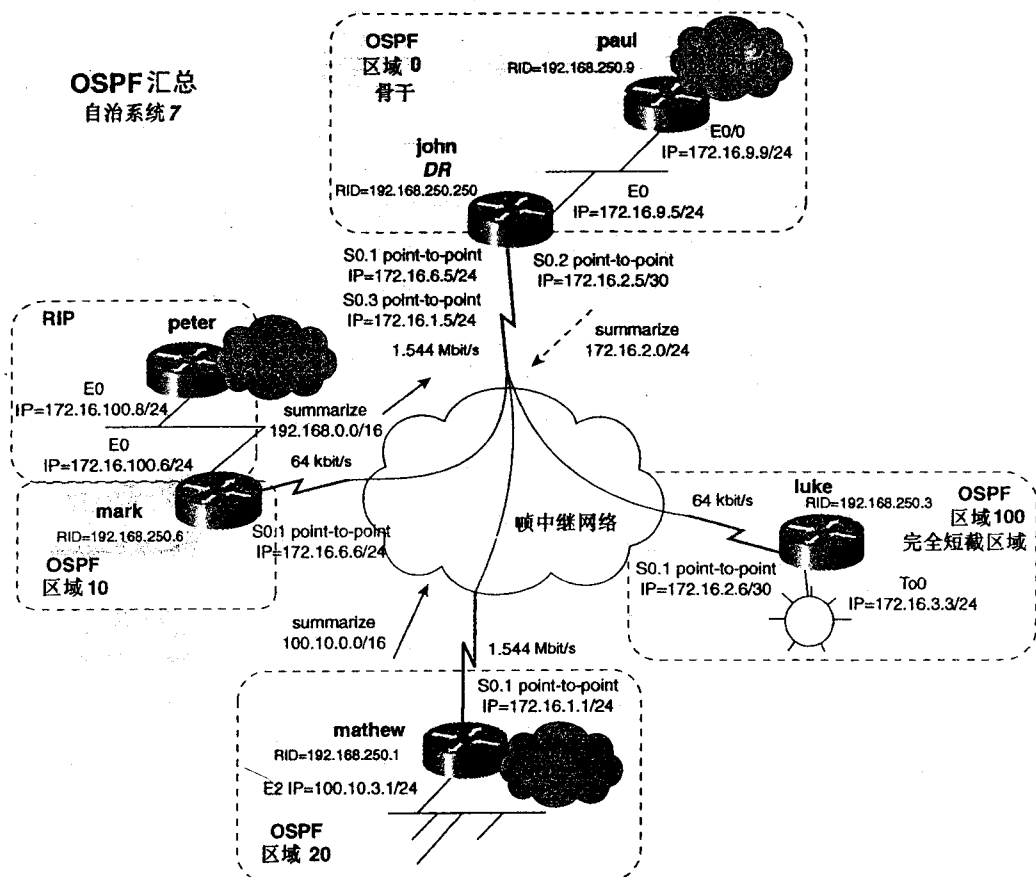


图 12-10 OSPF 的汇总

例 12-24 路由器 mark 的路由表

```
mark#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

100.0.0.0/16 is subnetted, 1 subnets
O IA 100.10.0.0 [110/129] via 172.16.6.5, 00:26:45, Serial0.1
172.16.0.0/24 is subnetted, 6 subnets
```

(待续)

```

O IA 172.16.9.0 [110/74] via 172.16.6.5, 00:26:45, Serial0.1
C    172.16.6.0 is directly connected, Serial0.1
O IA 172.16.1.0 [110/128] via 172.16.6.5, 00:26:45, Serial0.1
O IA 172.16.2.0 [110/128] via 172.16.6.5, 00:26:45, Serial0.1
O IA 172.16.3.0 [110/134] via 172.16.6.5, 00:26:45, Serial0.1
C    172.16.100.0 is directly connected, Ethernet0
      128.100.0.0/24 is subnetted, 2 subnets
O E2 128.100.1.0 [110/20] via 172.16.6.5, 00:26:45, Serial0.1
O E2 128.100.2.0 [110/20] via 172.16.6.5, 00:26:45, Serial0.1
R    192.168.1.0/24 [120/1] via 172.16.100.8, 00:00:06, Ethernet0
R    192.168.2.0/24 [120/1] via 172.16.100.8, 00:00:07, Ethernet0
mark#

```

最后进行外部汇总。在路由器 mark 上，把两个 RIP 路由 192.168.1.0/24 和 192.168.2.0/24 汇总为一个 OSPF 路由 192.168.0.0/16。这可以通过 OSPF 的路由器命令 **summary-address 192.168.0.0 255.255.0.0** 来完成。例 12-25 分别列出了 john 和 mark 上的路由选择协议配置示例。

例 12-25 john 和 mark 的配置示例

```

hostname mark
!
router ospf 7
router-id 192.168.250.6
summary-address 192.168.0.0 255.255.255.0
redistribute rip subnets
passive-interface Ethernet0
network 172.16.6.0 0.0.0.255 area 10
default-metric 10
!
router rip
redistribute ospf 7
passive-interface Serial0
network 172.16.0.0
default-metric 3
!

hostname john
!
router ospf 7
router-id 192.168.250.250
area 7 stub
area 20 range 100.10.0.0 255.255.0.0
area 100 stub no-summary
area 100 range 172.16.2.0 255.255.255.0
network 172.16.1.0 0.0.0.255 area 20
network 172.16.2.4 0.0.0.3 area 100
network 172.16.6.0 0.0.0.255 area 10
network 172.16.9.0 0.0.0.255 area 0
!

```

例 12-26 则是 paul 的路由表，列出了所有汇总路由。

例 12-26 paul 的路由表

```
paul#show ip route
<<<text omitted>>>

Gateway of last resort is not set

  100.0.0.0/16 is subnetted, 1 subnets
O IA   100.10.0.0 [110/75] via 172.16.9.5, 1d22h, Ethernet0/0
  172.16.0.0/24 is subnetted, 6 subnets
C       172.16.9.0 is directly connected, Ethernet0/0
O IA   172.16.6.0 [110/74] via 172.16.9.5, 1d22h, Ethernet0/0
O IA   172.16.1.0 [110/74] via 172.16.9.5, 1d22h, Ethernet0/0
O IA   172.16.2.0 [110/74] via 172.16.9.5, 1d22h, Ethernet0/0
O IA   172.16.3.0 [110/80] via 172.16.9.5, 1d22h, Ethernet0/0
O E2   172.16.100.0 [110/10] via 172.16.9.5, 01:19:17, Ethernet0/0
  128.100.0.0/24 is subnetted, 2 subnets
C       128.100.1.0 is directly connected, Loopback20
C       128.100.2.0 is directly connected, Loopback21
O E2   192.168.0.0/16 [110/10] via 172.16.9.5, 00:01:17, Ethernet0/0
paul#
```

12.9 OSPF 的默认路由

现在，大部分（有可能成为全部）的网络都是与 Internet 相连的。一些路由器具有登记的地址空间，而另一些则带有 Internet 路由器，或是防火墙。能够产生和传播默认路由是很重要的。前面讲过，默认路由的配置过程包括 3 个步骤：

第 1 步 将网络标记为默认。OSPF 将某个路由视为默认路由之前，该路由必须事先标记为默认路由。这可以通过下面两条全局命令中的一条来完成：

```
Router (config) #ip default-network network_address
Router (config) #ip route 0.0.0.0 0.0.0.0 ip_address
```

这条特殊的静态路由不需要重分布进 OSPF 来进行传播。OSPF 会将 0.0.0.0 0.0.0.0 视为一个默认路由，并对它进行相应的处理。如果使用上面的 **default-network** 命令，就还需要用到下面第 2 步中的关键字 **always**。

第 2 步 发送传播该默认路由，用下面这条 OSPF 的路由器命令来完成：

```
default-information originate [always] [metric cost] [metric-type OE1 |
OE2] [route-map map-name]
```

第 3 步 启动无类 IP。前面讲过，任何路由选择协议向不在其路由表中的目的地址发送数据包时，路由器必须启动 **ip classless**。在 12.0 及更新版本的 Cisco IOS 中，**ip classless** 是默认启动的。

图 12-11 中又向网络中添加了链路 206.191.200.1，这是网络通往 Internet 的网关。下面的两条全局配置命令都可以将网络 206.191.200.0 标记为默认网络：

```
ip default-network 206.191.200.0
```

或者直接将其指向一个地址：

```
ip route 0.0.0.0 0.0.0.0 206.191.200.1
```

命令 **default-information originate always** 可以发送该默认网络路由。如果路由器用

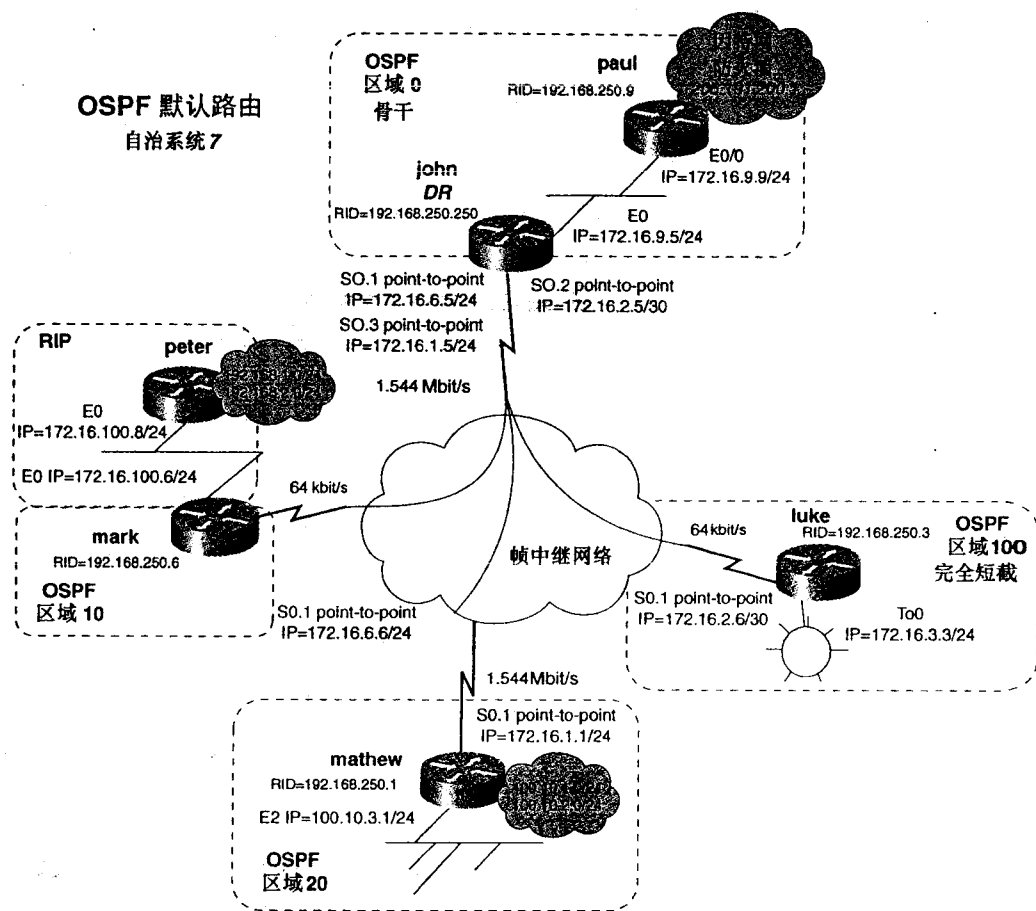


图 12-11 OSPF 的默认路由

例 12-27 列出了 paul 的配置信息，显示了默认路由的配置情况。

例 12-27 paul 上的默认路由配置

```
router ospf 7
router-id 192.168.250.9
redistribute connected subnets tag 9
network 172.16.9.0 0.0.0.255 area 0
default-information originate always ←Propagate the default route
default-metric 10
!
ip classless ←IP classless is always needed
ip route 0.0.0.0 0.0.0.0 206.191.200.1 ←"Flag" the route as default
!
```

例 12-28 解释了路由如何传播到 NSSA 区域的 mathew 去。注意，例子中设置了最后手段的网关。

例 12-28 mathew 的路由表

```
mathew#show ip route
<<<text omitted>>>

Gateway of last resort is 172.16.1.5 to network 0.0.0.0

100.0.0.0/24 is subnetted, 3 subnets
C    100.10.2.0 is directly connected, Loopback21
C    100.10.3.0 is directly connected, Ethernet2
C    100.10.1.0 is directly connected, Loopback20
172.16.0.0/24 is subnetted, 8 subnets
C    172.16.250.0 is directly connected, Loopback0
O IA  172.16.9.0 [110/74] via 172.16.1.5, 1d23h, Serial0.1
O     172.16.10.0 is a summary, 2d00h, Null0
O IA  172.16.6.0 [110/128] via 172.16.1.5, 1d23h, Serial0.1
C    172.16.1.0 is directly connected, Serial0.1
O     172.16.2.0 is a summary, 2d00h, Null0
O IA  172.16.3.0 [110/134] via 172.16.1.5, 1d23h, Serial0.1
O E2  172.16.100.0 [110/10] via 172.16.1.5, 02:54:58, Serial0.1
O E2  206.191.200.0/24 [110/20] via 172.16.1.5, 00:09:56, Serial0.1
12.0.0.0/16 is subnetted, 1 subnets
O E2  12.16.0.0 [110/10] via 172.16.1.5, 1d23h, Serial0.1
O*E2  0.0.0.0/0 [110/1] via 172.16.1.5, 01:04:36, Serial0.1
O E2  192.168.0.0/16 [110/10] via 172.16.1.5, 01:36:57, Serial0.1
mathew#
```

12.10 OSPF 的认证

OSPF 采用了两种认证方式，类型 I 和类型 II，两种方式的配置都非常容易。设置密码时，不需要输入接口密码的加密类型，只要在所有的配置完成之后用全局命令 **service password-encryption** 启动所有的密码保护功能即可。

12.10.1 类型 I 认证方式

类型 I 认证为明文形式的认证。如果在网络中放置了网络分析仪就可以截获密码，因此类型 I 的安全性没有类型 II 高。类型 I 认证方式的配置分为两个步骤：

第 1 步 在区域中所有的路由器上用下面的命令启动区域认证：

```
Router (config-route) #area area_id authentication
```

第 2 步 在接口上输入明文形式的密码，所用命令为：

```
Router (config-if) #ip ospf authentication-key password
```

整个区域中所有接口的密码和认证方式都必须匹配，否则会丢失邻接关系。

12.10.2 类型 II 认证方式

类型 II 是信息摘要 5（MD5）加密校验和的认证方式。OSPF 进程从 OSPF 的密钥和密码可以算出一个散列值。该散列值是链路中发送的惟一数值，链路不传输任何密码，这样就提高了认证的安全性。配置类型 II（MD5）认证方式的步骤为：

第 1 步 在区域中所有路由器上使用下列命令启动 MD5 区域认证:

Router (config-route) #area area_id authentication message-digest

第 2 步 为每个接口设置密钥，所用命令为:

Router (config-if) #ip ospf message-digest-key key_value md5 password

网络中所有路由器的 *key_value* 和 *password* 参数都必须完全匹配。

不同的密钥值可以迅速改变密码，实现单个区域具有多个密码。

12.10.3 类型 I 和类型 II 认证实例

图 12-12 是 Area 10 中网络的一部分。例 12-29 和 12-30 分别是该网络的类型 I 和类型 II 认证的 OSPF 配置示例。

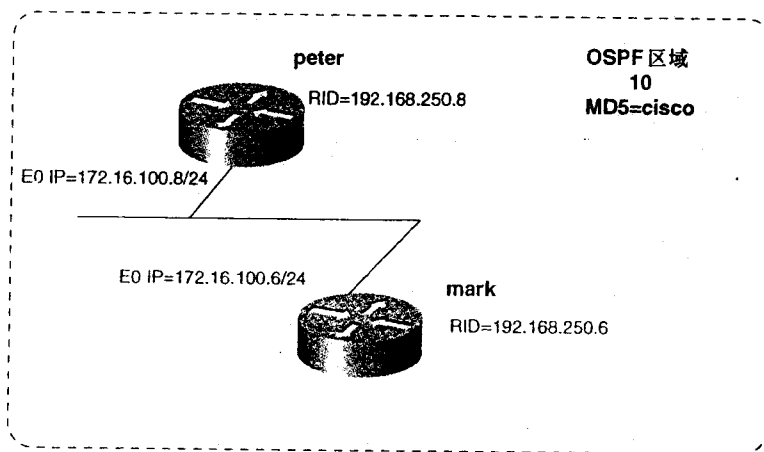


图 12-12 OSPF 的认证

例 12-29 网络的 OSPF 类型 1 认证

```
!
hostname peter
!
interface Ethernet0
 ip address 172.16.100.8 255.255.255.0
 ip ospf authentication-key cisco    →Cisco is the password
!
router ospf 7
 network 172.16.100.8 0.0.0.0 area 10
 area 10 authentication              →Type 1 authentication enabled in area 10
!
-----
hostname mark
!
interface Ethernet0
 ip address 172.16.100.6 255.255.255.0
 no ip directed-broadcast
 ip ospf authentication-key cisco
!
router ospf 7
 router-id 192.168.250.6
 area 10 authentication
```

```

network 172.16.6.0 0.0.0.255 area 10
network 172.16.100.6 0.0.0.0 area 10
!

```

例 12-30 是对图 12-12 中的网络进行 MD5 认证的情况。

例 12-30 Area 10 的 OSPF 类型 2 认证

```

!
hostname peter
!
interface Ethernet0
ip address 172.16.100.8 255.255.255.0
ip ospf message-digest-key 1 md5 cisco ~Cisco is the password, key=1
!
router ospf 7
network 172.16.100.8 0.0.0.0 area 10
area 10 authentication message-digest ~Type 2 authentication enabled in area 10

hostname mark
!
interface Ethernet0
ip address 172.16.100.6 255.255.255.0
no ip directed-broadcast
ip ospf message-digest-key 1 md5 cisco
!
router ospf 7
router-id 192.168.250.6
area 10 authentication message-digest
network 172.16.6.0 0.0.0.255 area 10
network 172.16.100.6 0.0.0.0 area 10
!

```

12.11 OSPF 按需电路和备份

在 OSPF 中进行备份很困难，尤其是用 ISDN 做按需拨号备份就更难，这种困难主要是由于 OSPF 的区域连接性造成的。对备份接口进行控制的关键在于备份接口所处的 OSPF 区域。不同的区域有不同的特性，就像不同类型的 LSA 能够进入不同的区域等，备份区域选择非常重要。另外还会影响备份接口进入不同的区域的因素，包括路由器在网络中位置，区域类型以及这些区域如何与 Area 0 相连接等。这里要讲述的不是众多的 OSPF 备份和按需电路的实例，而是一些配置标准以及常见问题。可以参考前面第 5 章“WAN 协议与技术：帧中继”和第 7 章“WAN 协议与技术：综合业务数字网 (ISDN)”的相关内容来更多的了解备份接口和 OSPF 按需电路方面的实例和相关讲解。下面将涉及到 OSPF 按需电路和备份的配置标准问题。

12.11.1 坚持 OSPF 的设计规则

备份线路开始接替工作后，整个网络仍然要遵守所有 OSPF 的设计规则。即 Area 0 不能进行分区，备份线路不能部署在备份区域中，所有区域必须与 Area 0 相连等。在备份模式下，本人概不负责

网络拓扑结构可能会发生改变，但是仍然得严格遵守同样的 OSPF 的设计规则。

12.11.2 OSPF 按需电路

RFC 1793 中概要说明了 OSPF 按需电路最初的标准。就其实质来看，OSPF 按需电路会伪装 hello 数据包（在多播地址 224.0.0.5 上）以启动按需电路。它通过只在电路第 1 次激活时交换 LSA 信息以及设置 LSA 的 DoNotAge 位的方法来试图对 LSA 的发送加以控制。如果拨号链路在 Area 0 中，或者是 OSPF 网络具有外部 LSA 或类型 5 的 LSA，或者是 OSPF 网络在其 NSSA 区域里含有类型 7 的 LSA，那么按需电路就无法正常工作。类型 5 和 7 的 LSA 或者 NSSA 的 LSA 会强制使一个 DDR 链路（如 ISDN）不停地进行呼叫。多数网络都会有某种形式的重分布操作，并且一定会在网络中有类型 5 的 LSA 不断振荡。只有和按需电路相连的 Stub 区域可以避免由类型 5 的 LSA 导致的 DDR 链路不停呼叫。配置按需电路，需要进行 3 个步骤的工作：

- 第 1 步 在网络链路的两端将接口配置成为 OSPF 点对点网络的接口。前面讲过，“广播”类型的网络中不能抑制 hello 数据包，而且虚链路也不能建立在存根区域中。
- 第 2 步 用 `Area x stub` 命令将所有的接口配置到同一个 Stub 区域中。
- 第 3 步 用接口命令 `ip ospf demand-circuit` 将网络链路的呼叫方配置成按需电路。

12.11.3 Area 0 的设计准则

只有主接口位于 Area 0 中时其备份接口才会进入 Area 0 中。用 `backup interface` 命令或 `dialer watch` 命令可以启动动态的路由方式。切记所有的 LSA 都会进入 Area 0 中。LSA 的不停流入 Area 0 会导致接口的不停呼叫。在这种情况下，必须运用一些配置技术强制性的使接口不进行没有必要的呼叫。

12.12 OSPF 的虚链路

我们在最后才讨论 OSPF 的虚链路问题，这主要是因为虚链路应该是配置时的最后一个选项。Cisco 的设计指导中警告说，虚链路的使用是表明设计质量不高。很大程度上来说，这样说是对的。虚链路是用来将 Area 0 通过另一个区域进行扩展的方式。也可以把虚链路看成 LSA 的通道。需要使用虚链路的情况是划分区域或者是某个区域没有和 Area 0 相连。下面这条路由器命令可用来配置虚链路：

```
Router (config-route) #area transit_area_id virtual-link router_id_of_remote
```

参数 `area_id` 是 OSPF 要通过的“传输区域”通道。“传输区域”不能是任何类型的存根区域。通道的另一端应该是另一台路由器作为虚链路的终点，`router_id` 字段指定的就是该路由器。虚链路用 RID 来标识路由器，这也是配置 OSPF 时采用固定 RID 的原因之一。有一点很重要，那就是在 Area 0 中改变任何计时器的值或者是 OSPF 的认证方式时，一定要记住在虚链路的另一端作相应的修改。虚链路实际上就是 Area 0 的特殊扩展方式。应把虚链路看作路由器接入 Area 0 的新接口，在虚链路上完成所有的接口配置项。

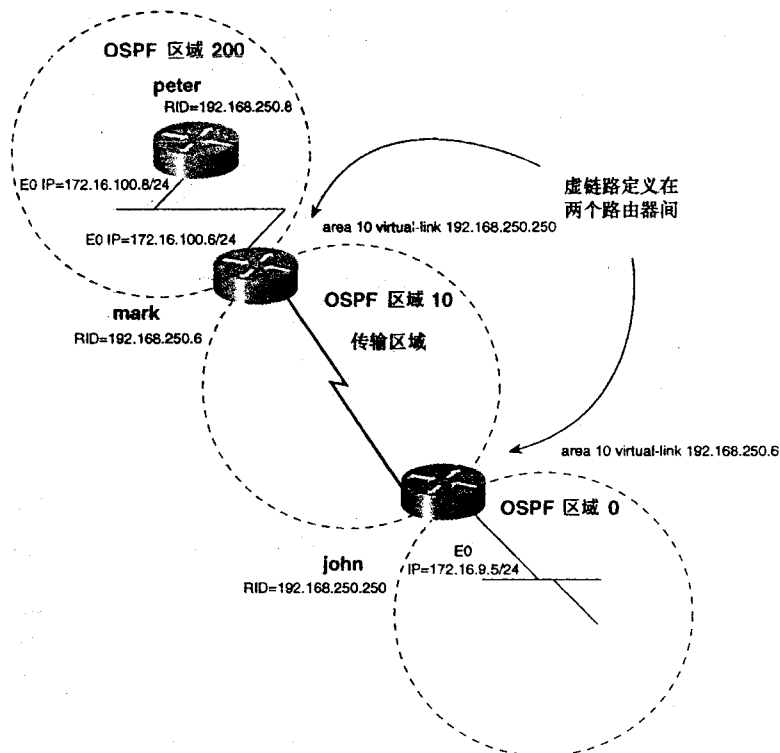


图 12-13 OSPF 的虚链路

图 12-13 中，路由器 peter 处在 Area 200 中。注意，Area 200 和 Area 0 之间没有任何直连的链路。要使该网络正常工作，需要通过 Area 10 定义一条虚链路。在路由器 mark 上定义一条将 Area 10 作为传输区域的虚链路，该链路的终端 RID 是 192.168.250.250，也就是路由器 john。在 john 上配置一条以 Area 10 作为传输区域的虚链路，并以 RID 为 192.168.250.6 的路由器作为链路的终端。各个虚链路的配置过程请参考图 12-13。

利用 `show ip ospf virtual-links` 命令和标准的 `show ip route` 以及 `ping` 可以验证虚链路的功能。如果工作正常，虚链路的状态应为“up”，邻接关系的状态则为“FULL”。如果虚链路没有如期开始工作，检查 RID 是否为 OSPF 所用的 RID。如果不清楚当前路由器的 RID，可以查看一下 OSPF 数据库的内容。例 12-31 列出了 john 上 `show ip ospf virtual-links` 命令的输出示例。

例 12-31 验证 OSPF 虚链路

```
john#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 192.168.250.6 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 10, via interface Serial0.1, Cost of using 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:09
  Adjacency State FULL (Hello suppressed)
  Index 2/5, retransmission queue length 0, number of retransmission 1
```

(待续)

```
First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
Last retransmission scan length is 1, maximum is 1
Last retransmission scan time is 0 msec, maximum is 0 msec
john#
```

12.13 实验 24：配置 OSPF：多域路由、认证、 路径管理和默认路由——第 1 部分

12.13.1 实验说明

本书中多次提到，现在的网络大都以某种形式与 Internet 连接在一起。与 Internet 的互连通常需要在整个网络中传播一条默认路由。这个实验就为大家提供了在 OSPF 网络中配置多个不同类型的 OSPF 区域，对某个区域进行认证，操控路由路径以及生成并传播默认路由的练习机会。

12.13.2 实验内容

假定著名医生 Dr. Stai 拥有一些地方性的事务所，每个事务所都专门研究牙科的某个特定领域，如牙根管填充手术，补牙等等。Dr. Stai 希望通过一个帧中继网络将他的这些事务所连接起来。这些事务所同时还希望通过共享一个公用连接实现与 Internet 的互连，这样他们才能及时获取减痛新技术的最新发展动态。要求按照下列标准配置一个 OSPF 网络：

- 按照图 12-14 配置一个 IP 网络，路由选择协议采用 OSPF，进程 ID 为 2002。
- 将路由器 dental_ho 和 fillings 之间的帧中继网络配置为点对点网络，而 dental_ho, crowns 与 root_canals 之间的网络则配置为帧中继多点网络。
- 按照图中所示配置所有的 OSPF 区域，Area 10 要配置成 NSSA 区域。
- 在路由器 dental_ho 中加入一条默认路由，把所有访问 Internet 的流量都指向网络 128.10.1.0/24，同时把这条路由在整个 OSPF 区域中进行传播。
- 在 Area 200 中配置类型 2 的认证方式。
- 路由器 root_canals 与 dental_ho 之间有一条 T1 连接。对这一网络（路由）进行配置，使得路由器 pain_center 在访问 WAN 上任何网络的时候会优先采用这条与 root_canals 的 T1 连接，而不是去用与 crowns 路由器的 64-kbit/s 连接。

12.13.3 实验目的

- 按照图 12-14 对 Dr. Stai 的牙科网络以及相应的 IP 地址进行配置。这个实验中的 LAN 拓扑类型对实验结果没用影响。
- 在 WAN 上使用帧中继数据链路协议，而且按照图中所示的那样只使用多点网络和点对点网络。

- 确保所有 IP 接口的 IP 连接正常——也就是说，可以 ping 通所有的帧中继和 LAN 接口。
- 不要改变默认的 OSPF 网络类型，不要使用任何静态路由。
- 在路由器 dental_ho 中加入一条默认路由，把所有的 IP 流量都指向子网 128.10.1.0/24。利用一个默认网络来做到这一点，并将此默认路由在整个 OSPF 网络中传播。
- 在 Area 200 中使用 Type 2 认证方式，密码是 cisco。
- 对 OSPF 进行调整，使得路由器 pain_center 会优先使用通过 root_canals 的路径而不是通过 crowns 的那条路径。也就是说，所有 pain_center 发出的数据都要求优先考虑通过 root_canals。
- (可选) 当且仅当路由器 dental_ho 的路由表中出现了子网 128.10.1.0/24 之后才会对默认路由进行传播。如果该子网不在路由表中，说明 dental_ho 路由器无法访问这个子网，因而默认路由也就不应该进行传播。在这个实验中只需要对通往普通区域的默认路由进行控制，因此大家不用担心如何控制 NSSA 区域中的默认路由。

12.13.4 所需设备

- 6 台 Cisco 路由器，其中 4 台要通过 V.35 背对背线缆或者是类似的方式与帧中继交换机连接在一起。
- 利用集线器或交换机构建 4 个 LAN 网段。LAN 的拓扑结构没有关系。Internet 连接可以是真实的，也可以模拟，这不会对路由器的配置有影响。

12.13.5 物理设计与实验准备

- 按照图 11-14 将集线器以及串行线缆与路由器相连。
- 将剩下的那台路由器配置来作为与 Internet 的连接，这一步是可选的。
- 此外还需要一台具有 3 条 PVC 的帧中继交换机，例 12-32 是实验中所需的帧中继交换机的配置。

例 12-32 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148800
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
 frame-relay route 121 interface Serial3 102
 frame-relay route 150 interface Serial5 151
!
interface Serial1
```

(待续)

```

no ip address
encapsulation frame-relay
clockrate 148000
frame-relay intf-type dce
frame-relay route 110 interface Serial0 111
!
interface Serial2
no ip address
shutdown
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 102 interface Serial0 121
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 151 interface Serial0 150

```

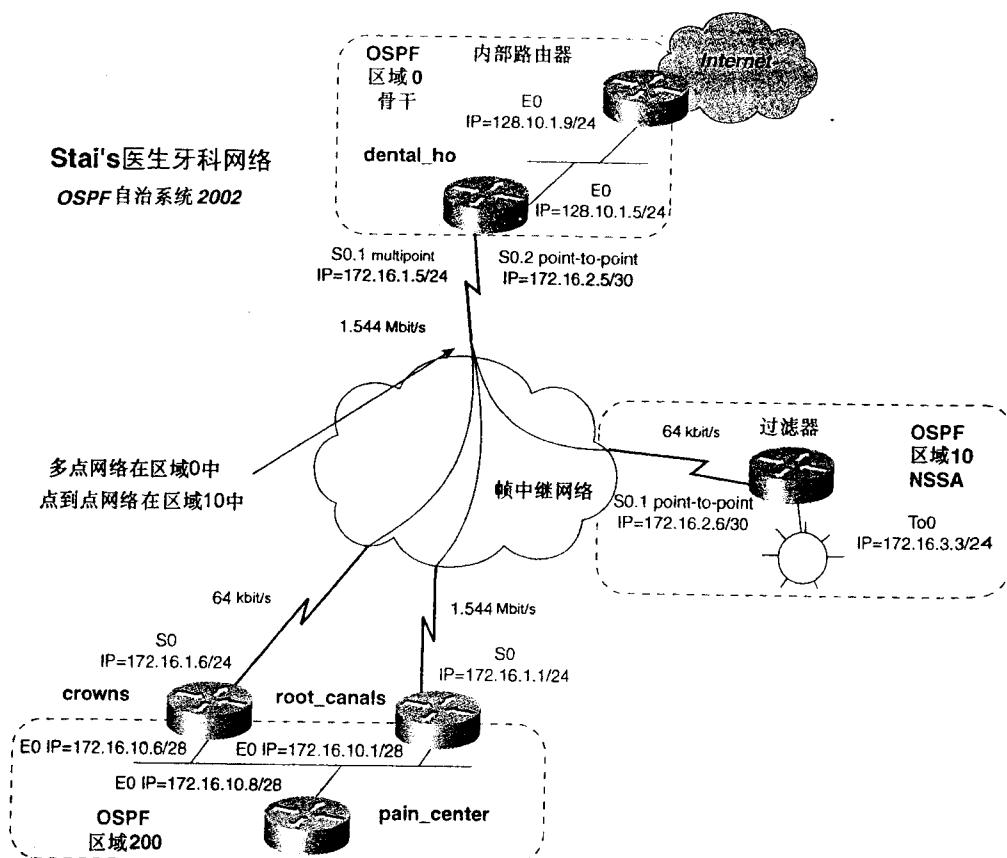


图 12-14 Dr. Stai 的牙科网络

12.14 实验24：配置 OSPF：域间路由、认证、 路径管理和默认路由——第2部分

12.14.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 3 台路由器以背对背方式连接在一起。然后按照图 11-14 利用交换机或集线器/MAU 创建 4 个 LAN 网段。

物理连接完成之后，按照图 11-14 所示为所有的 LAN 和 WAN 接口分配 IP 地址。在继续下面的操作之前，必须用 ping 命令对每台路由器的本地 LAN 和 WAN 接口进行测试。然后，在点对点接口上还需要用 **frame-relay interface-dlci** 命令进行配置，而多点接口上则需要 **frame-relay map** 命令。关于帧中继的配置，可以回顾一下第 5 章的相关内容。例 12-33 是到现此为止所有相关路由器上的帧中继配置。

例 12-33 帧中继配置

```
hostname dental_ho
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 172.16.1.5 255.255.255.0
 frame-relay map ip 172.16.1.6 121 broadcast
 frame-relay map ip 172.16.1.1 111 broadcast
!
interface Serial0.2 point-to-point
 ip address 172.16.2.5 255.255.255.252
 frame-relay interface-dlci 150
!

-----

hostname crowns
!
<<<text omitted>>>
!
interface Serial0
 ip address 172.16.1.6 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 102 broadcast
 frame-relay map ip 172.16.1.1 102 broadcast
 frame-relay lmi-type cisco
!

-----

hostname root_canals
```

```

!
interface Serial0
ip address 172.16.1.1 255.255.255.0
encapsulation frame-relay
no ip mroute-cache
frame-relay map ip 172.16.1.5 110 broadcast
frame-relay map ip 172.16.1.6 110 broadcast
!

hostname fillings
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
ip address 172.16.2.6 255.255.255.252
frame-relay interface-dlci 151
!

```

LAN 和 WAN 接口配置完成，基本的 IP 连接建立起来之后，就可以开始对 OSPF 进行配置。回想一下 OSPF 配置的步骤：

- 第 1 步 配置 OPSF 区域和选定 DR/BDR。
- 第 2 步 在运行着 Cisco IOS 12.0 及更新版本的路由器上，利用环路接口指定路由器 ID (RID)。
- 第 3 步 启动 OSPF，并在路由器上分配 RID，这些路由器上应运行 12.0 或更新版本的 Cisoc IOS。
- 第 4 步 对 OSPF 接口进行配置。
- 第 5 步 如果需要，为额外的邻居路由器进行配置。
- 第 6 步 对 OSPF 的区域类型进行配置。
- 第 7 步 对包括认证方式在内的其他 OSPF 参数进行配置。

第 1 步是 OSPF 区域的配置。在这个网络模型的 Area 200 中含有 crowns、root_canal 和 pain_center 路由器之间的 LAN。在 Area 200 中还要进行 Type 2 认证。路由器 fillings 是处在 NSSA 区域中的，多点帧中继网络以及路由器 dental_ho 的 LAN 局域网则是处在 Area 0 中。由于只有路由器 dental_ho 才具有与 crowns 和 root_canal 的直接 PVC 连接，因此该路由器应是帧中继多点网络的 DR。

第 2 步是在那些含 12.0 以上版本的 Cisco IOS 的路由器上设置 RID。这通过利用这些路由器上的环路接口来实现。图 12-15 是设置了 RID，分配了 OSPF 区域之后的网络情况。

第 3 步开始真正的配置过程。在所有的路由器上用 **router ospf 2002** 命令启动 OSPF 的运行。然后，在安装了 12.0 及以上版本的 Cisco IOS 的路由器上用 **router-id ip_address** 命令为路由器静态地指定 RID。

在第 4 步中，需要定义那些接口将要参与 OSPF 的路由以及这些路由器所在的区域。在路由器 dental_ho 上，接口 E0 是和多点接口 s0.1 同处在 Area 0 中的，s0.2 接口则是在 Area 10 中。例 12-34 是路由器 dental_ho 到目前为止所涉及到的 OSPF 配置。

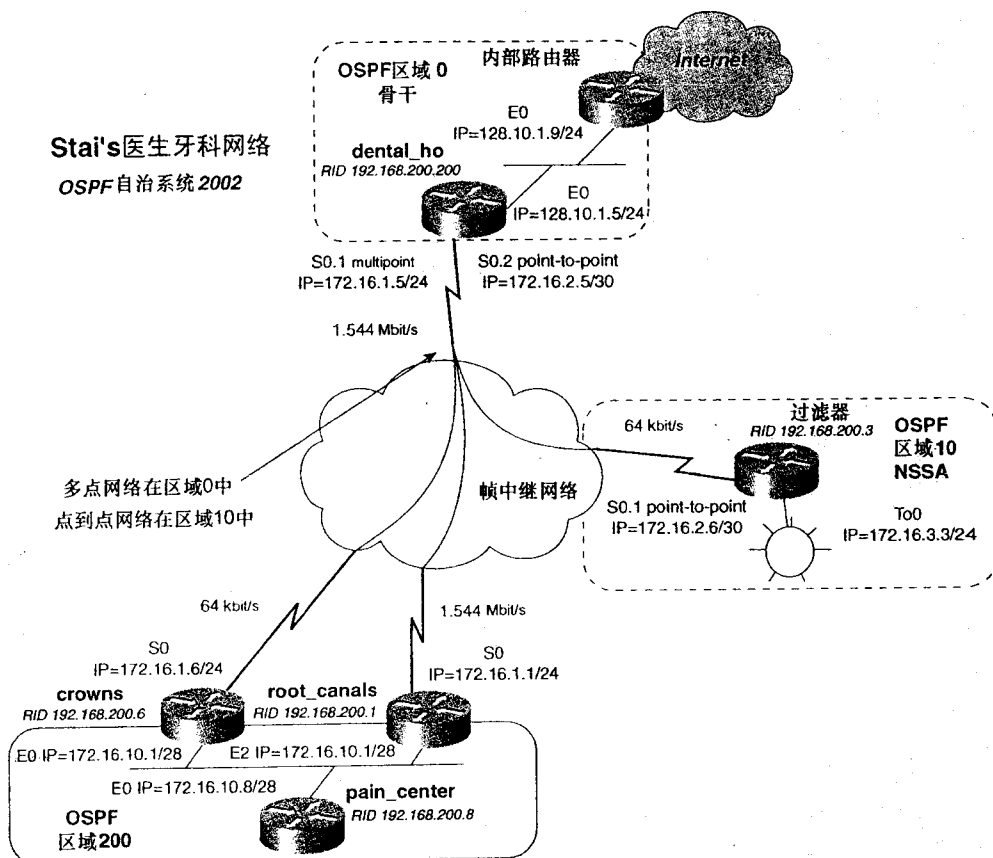


图 12-15 Dr. Stai 牙科网络的路由器 ID 分配

例 12-34 路由器 dental_ho 的初步 OSPF 配置示例

```
router ospf 2002
router-id 192.168.200.200
network 128.10.1.5 0.0.0.0 area 0
network 172.16.1.5 0.0.0.0 area 0
network 172.16.2.5 0.0.0.0 area 10
!
```

本章中提到过，通配符掩码 0.0.0.0 说明每个位元都必须匹配。在这个 OSPF 配置中，只把一个接口放入到一个区域中去。

路由器 **crowns**、**root_canals** 和 **pain_center** 上的初步 OSPF 配置相互之间都非常类似。路由器的 S0 接口是处在 Area 0 中的，而 LAN 接口则是处在 Area 200 中。路由器 **crowns** 的初始 OSPF 配置如例 12-35 所示。

例 12-35 路由器 crowns 的初步 OSPF 配置

```

router ospf 2002
router-id 192.168.200.6
network 172.16.1.6 0.0.0.0 area 0
network 172.16.10.6 0.0.0.0 area 200
!
```

路由器 fillings 在 Area 10 中有两个接口, 由于该路由器无需任何额外的邻居路由器支持, 因而配置这台路由器时可以跳过第 5 步, 直接把这个区域配置为一个 NSSA 区域。例 12-36 是路由器 fillings 的 OSPF 配置情况。

例 12-36 路由器 fillings 的初始配置

```

router ospf 2002
router-id 192.168.200.3
area 10 nssa
network 172.16.2.6 0.0.0.0 area 10
network 172.16.3.3 0.0.0.0 area 10
!
```

第 5 步是对 OSPF 所需要的额外邻居路由器的支持进行配置, 以便在帧中继多点网络上形成相应的邻接关系。要想正确地形成邻接关系, 应把路由器 dental_ho 的优先级设置为 255, 把 crowns 和 root_canals 的优先级设为 0。此外, 路由器上还需要用 **neighbor** 命令进行配置。例 12-37 是路由器 crowns 和 dental_ho 的相关配置。

例 12-37 路由器 crowns 和 dental_ho 的 OSPF 配置

```

!
interface Serial0
no ip address
encapsulation frame-relay
frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
ip address 172.16.1.5 255.255.255.0
ip ospf priority 255
frame-relay map ip 172.16.1.6 121 broadcast
frame-relay map ip 172.16.1.1 111 broadcast
!
interface Serial0.2 point-to-point
ip address 172.16.2.5 255.255.255.252
frame-relay interface-dlci 150
!
interface Serial1
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
```

(待续)

```

router ospf 2002
router-id 192.168.200.200
area 10
network 128.10.1.5 0.0.0.0 area 0
network 172.16.1.5 0.0.0.0 area 0
network 172.16.2.5 0.0.0.0 area 10
neighbor 172.16.1.1
neighbor 172.16.1.6
!

```

第 6 步只是路由器 dental_ho 和 fillings 的配置，要求是将 Area 10 配置为 NSSA 区域。要把一个区域配置成 NSSA 区域，只需要把 NSSA 参数添加到 area 命令中去即可。路由器 dental_ho 和 fillings 都需要对 NSSA 区域进行配置。例 12-38 是 fillings 路由器的相关 OSPF 配置。

例 12-38 路由器 fillings 上的 OSPF NSSA 配置

```

!
router ospf 2002
router-id 192.168.200.3
area 10 nssa
network 172.16.2.6 0.0.0.0 area 10
network 172.16.3.3 0.0.0.0 area 10
!

```

到现在为止，OSPF 的所有功能就已经完全具备，网络中所有路由器之间的 IP 连通都已经建立。可以通过检查邻居路由器，路由表以及使用标准 ping 命令进行测试。而对 NSSA 区域的验证则可以通过 show ip ospf 命令来进行。例 12-39 是路由器 dental_ho 上 show ip ospf neighbor 命令和 show ip ospf 命令的执行情况。

例 12-39 验证 OSPF 功能与 NSSA 配置

```

dental_ho#show ip ospf neighbor

Neighbor ID    Pri   State           Dead Time   Address        Interface
192.168.200.1    0    FULL/DROTHER    00:01:42   172.16.1.1    Serial0.1
192.168.200.6    0    FULL/DROTHER    00:01:44   172.16.1.6    Serial0.1
192.168.200.3    1    FULL/-          00:00:38   172.16.2.6    Serial0.2
dental_ho#
dental_ho#show ip ospf
Routing Process "ospf 2002" with ID 192.168.200.200
Supports only single TOS(TOS0) routes
<<<text omitted>>>
Area 10
  Number of interfaces in this area is 1
  It is a NSSA area
  Perform type-7/type-5 LSA translation
  generates NSSA default route with cost 1
  Area has no authentication
  SPF algorithm executed 11 times
  Area ranges are
  Number of LSA 6. Checksum Sum 0x30908
  Number of opaque link LSA 0. Checksum Sum 0x0

```

```

Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0

```

```
dental_ho#
```

最后一步需要配置 3 个参数：一条默认路由，认证方式以及路径选择。首先，配置默认路由的时候，将一条路由标记为默认的，然后在整个 OSPF 区域中传播。要将一条路由标记为默认路由而不使用静态路由，采用的是全局命令 **default-network 128.10.1.0**。而路由的传播则是通过 OSPF 命令 **default-information originate always** 来实现的。这里要记住的一点是，一台路由器要把数据包转发到一条默认路由去，所有的路由器都需要启用全局命令 **ip classless**。而默认路由不会自动进入 NSSA 区域，因此 NSSA 区域必须将参数 **default-information-originate** 添加到路由器命令 **area 10 nssa** 中去，以正常接收默认路由。例 12-40 是路由器 dental_ho 的配置情况，这里突出显示了相关默认路由转发的命令。

例 12-40 路由器 dental_ho 的配置

```

router ospf 2002
router-id 192.168.200.200
area 10 nssa default-information-originate
network 128.10.1.5 0.0.0.0 area 0
network 172.16.1.5 0.0.0.0 area 0
network 172.16.2.5 0.0.0.0 area 10
neighbor 172.16.1.1
neighbor 172.16.1.6
default-information originate always
!
ip classless
ip default-network 128.10.0.0

```

在验证默认路由传播情况的时候，可以查看除了 dental_ho 以外任何一台路由器的路由表，找到所使用的最后手段的网关，以及以*标记，表示是默认路由的路由。例 12-41 是 fillings 路由器的路由表情况，从这里可以看到默认路由被作为 OSPF NSSA 的一条外部类型 2 路由发送到了 NSSA 区域中。

例 12-41 路由器 fillings 的路由表

```

fillings#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 172.16.2.5 to network 0.0.0.0
```

```
172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
```

```
O IA 172.16.10.0/28 [110/138] via 172.16.2.5, 00:09:18, Serial0.1
```

```

C      172.16.2.4/30 is directly connected, Serial0.1
O IA   172.16.1.0/24 [110/128] via 172.16.2.5, 00:09:18, Serial0.1
C      172.16.3.0/24 is directly connected, TokenRing0
      128.10.0.0/24 is subnetted, 1 subnets
O IA   128.10.1.0 [110/74] via 172.16.2.5, 00:09:18, Serial0.1
O*N2 0.0.0.0/0 [110/1] via 172.16.2.5, 00:09:18, Serial0.1
fillings#

```

接下来到 Area 200 中配置 Type 2，或者说是 MD5 认证。在 OSPF 的 `area` 命令和接口上都需要允许认证的使用。MD5 密码设为 `cisco`。例 12-42 是路由器 `pain_center` 上为认证而作的配置。Area 200 中所有路由器也必须进行完全相同的配置。

例 12-42 路由器 `pain_center` 上 MD5 认证的配置

```

interface Ethernet0
 ip address 172.16.10.8 255.255.255.240
 ip ospf message-digest-key 1 md5 cisco
!
<<<text omitted>>>
!
router ospf 2002
 network 172.16.10.8 0.0.0.0 area 200
 area 200 authentication message-digest

```

在整个 Area 200 中启用了认证之后，这里的路由和邻居路由器就会开始过期并且消失，这是所配置的认证开始工作的表现。通过 `show ip ospf` 命令也可以看出来区域加上了认证。所有的路由器都启用了认证之后，邻居路由器和相应的路由就会重新出现。

除了可选部分之外，本实验的最后一步是对来自路由器 `pain_center` 的数据包的传输路径施加某些影响。这台路由器发出的数据总是优先选用通过路由器 `root_canals` 的主路径。在 `pain_center` 的路由表中可以看到到网络其他部分存在着两条路径，一条是通过 `root_canals`，另一条则经过 `crowns`，如例 12-43 所示。

例 12-43 路由器 `pain_center` 的路由表

```

pain_center#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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.10.1 to network 0.0.0.0

    128.10.0.0/24 is subnetted, 1 subnets
O IA   128.10.1.0 [110/84] via 172.16.10.1, 00:14:03, Ethernet0
        [110/84] via 172.16.10.6, 00:14:03, Ethernet0
C      192.168.200.0/24 is directly connected, Loopback0
    172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
C      172.16.10.0/28 is directly connected, Ethernet0
O IA   172.16.2.4/30 [110/138] via 172.16.10.1, 00:14:03, Ethernet0
        [110/138] via 172.16.10.6, 00:14:03, Ethernet0
O IA   172.16.1.0/24 [110/74] via 172.16.10.1, 00:14:03, Ethernet0

```

```

[110/74] via 172.16.10.6, 00:14:03, Ethernet0
O IA 172.16.3.0/24 [110/144] via 172.16.10.1, 00:14:04, Ethernet0
[110/144] via 172.16.10.6, 00:14:04, Ethernet0
O*E2 0.0.0.0/0 [110/1] via 172.16.10.1, 00:14:04, Ethernet0
[110/1] via 172.16.10.6, 00:14:04, Ethernet0
pain_center#

```

OSPF 会在这些路由上进行负载共享，但是实际上只想使用一条路由。如果主路由不可用，OSPF 会启用通过路由器 crowns 的备份路由。要改变路由器转发路由的路径选择，可以利用 **bandwidth** 命令改变链路的路由成本，或者是利用接口命令 **ip ospf cost** 直接改变路由成本。在这个网络模型中改变转发路由的方法很多，这里采用的是将路由器 root_canals 接口 s0 的 OSPF 路由成本设为 15 的方法。在路由器 root_canals 上完成这样的改动之后，路由器 pain_center 的路由表如例 12-44 所示，后面还执行了一次 trace 操作。路由表中就只有通过 172.16.10.1 的主路径。

例 12-44 路由器 pain_center 的路由表

```

pain_center#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, * - candidate default
U - per-user static route, o - ODR

Gateway of last resort is 172.16.10.1 to network 0.0.0.0

128.10.0.0/24 is subnetted, 1 subnets
O IA 128.10.1.0 [110/35] via 172.16.10.1, 00:00:17, Ethernet0
C 192.168.200.0/24 is directly connected, Loopback0
172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
C 172.16.10.0/28 is directly connected, Ethernet0
O IA 172.16.2.4/30 [110/89] via 172.16.10.1, 00:00:17, Ethernet0
O IA 172.16.1.0/24 [110/25] via 172.16.10.1, 00:00:17, Ethernet0
O IA 172.16.3.0/24 [110/95] via 172.16.10.1, 00:00:17, Ethernet0
O*E2 0.0.0.0/0 [110/1] via 172.16.10.1, 00:00:17, Ethernet0
pain_center#trace 128.10.1.5

Type escape sequence to abort.
Tracing the route to 128.10.1.5

 1 172.16.10.1 0 msec 0 msec 0 msec
 2 172.16.1.5 16 msec 24 msec *
pain_center#

```

实验的可选部分要求，仅当网络 128.10.1.0/24 出现在了路由表中的情况下才会对默认路由进行传播。如果网络 128.10.1.0 不可用，则路由器 dental_ho 不会将默认路由传播到 OSPF 区域其他部分。为了达到这个目的，就需要配置一个所谓的条件默认路由，这是通过在 **default-information originate** 命令中调用一份路由图来实现的，如例 12-45 所示。这份路由图的匹配条件是网络 128.10.1.0 的地址前缀列表。

例 12-45 配置条件默认路由

```

1
router ospf 2002
router-id 192.168.200.200
area 10 nssa default-information-originate
network 128.10.1.5 0.0.0.0 area 0
network 172.16.1.5 0.0.0.0 area 0
network 172.16.2.5 0.0.0.0 area 10
neighbor 172.16.1.6
neighbor 172.16.1.1
default-information originate always route-map condition --calls route-map
condition
1
ip classless
ip default-network 128.10.0.0
no ip http server
1
1
ip prefix-list cond seq 5 permit 128.10.1.0/24 --match route 128.10.1.0/24
route-map condition permit 10
match ip address prefix-list cond --call prefix-list called cond
1

```

以太接口关闭之后，路由器 dental_ho 不会再转发默认路由。由于 NSSA 的默认路由是以不同的方法加以控制的，因而 NSSA 区域的默认路由不能通过调用路由图来进行控制。未来的 Cisco IOS 可能会提供这一功能。

例 12-46 是该实验相关路由器的配置清单。

例 12-46 本实验中用到的路由器配置

```

hostname dental_ho
1
interface Ethernet0
ip address 128.10.1.5 255.255.255.0
1
interface Serial0.1 multipoint
ip address 172.16.1.5 255.255.255.0
ip ospf priority 255
frame-relay map ip 172.16.1.6 121 broadcast
frame-relay map ip 172.16.1.1 111 broadcast
1
interface Serial0.2 point-to-point
ip address 172.16.2.5 255.255.255.252
frame-relay interface-dlci 150
1
<<<text omitted>>>
1
router ospf 2002
router-id 192.168.200.200
area 10 nssa default-information-originate
network 128.10.1.5 0.0.0.0 area 0
network 172.16.1.5 0.0.0.0 area 0
network 172.16.2.5 0.0.0.0 area 10
neighbor 172.16.1.6
neighbor 172.16.1.1
default-information originate always route-map condition

```

（待续）

```

!
ip classless
ip default-network 128.10.0.0
no ip http server
!
ip prefix-list cond seq 5 permit 128.10.1.0/24
route-map condition permit 10
  match ip address prefix-list cond
!
!

```

```

hostname crowns
!
interface Ethernet0
  ip address 172.16.10.6 255.255.255.240
  no ip directed-broadcast
  ip ospf message-digest-key 1 md5 cisco
!
interface Serial0
  ip address 172.16.1.6 255.255.255.0
  no ip directed-broadcast
  encapsulation frame-relay
  ip ospf priority 0
  no ip mroute-cache
  frame-relay map ip 172.16.1.5 102 broadcast
  frame-relay map ip 172.16.1.1 102 broadcast
  frame-relay lmi-type cisco
!
router ospf 2002
  router-id 192.168.200.6
  area 200 authentication message-digest
  network 172.16.1.6 0.0.0.0 area 0
  network 172.16.10.6 0.0.0.0 area 200
!
ip classless
!
!

```

```

hostname root_canals
!
interface Loopback0
  ip address 192.168.200.1 255.255.255.0
!
<<<text omitted>>>
!
interface Ethernet2
  ip address 172.16.10.1 255.255.255.240
  ip ospf message-digest-key 1 md5 cisco
  media-type 10BaseT
!
!
interface Serial0
  ip address 172.16.1.1 255.255.255.0
  encapsulation frame-relay
  ip ospf cost 15
  ip ospf priority 0
  no ip mroute-cache
  frame-relay map ip 172.16.1.5 110 broadcast
  frame-relay map ip 172.16.1.6 110 broadcast
!
<<<text omitted>>>
!
router ospf 2002
  network 172.16.1.1 0.0.0.0 area 0

```

```

network 172.16.10.1 0.0.0.0 area 200
area 200 authentication message-digest
!
ip classless
!
!

hostname pain center
!
interface Loopback0
ip address 192.168.200.8 255.255.255.0
!
interface Ethernet0
ip address 172.16.10.8 255.255.255.240
ip ospf message-digest-key 1 md5 cisco
!
<<<text omitted>>>
!
router ospf 2002
network 172.16.10.8 0.0.0.0 area 200
area 200 authentication message-digest
!
ip classless
!
!

hostname fillings
!
interface Serial0
no ip address
encapsulation frame-relay
frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
ip address 172.16.2.6 255.255.255.252
frame-relay interface-dlci 151
!
interface TokenRing0
ip address 172.16.3.3 255.255.255.0
ring-speed 16
!
router ospf 2002
router-id 192.168.200.3
area 10 nssa
network 172.16.2.6 0.0.0.0 area 10
network 172.16.3.3 0.0.0.0 area 10
!
!
ip classless

```

12.15 实验 25：配置 OSPF：多域路由、路由的重分布与汇总功能——第 1 部分

12.15.1 实验说明

随着 OSPF 网络的持续增长和发展，不可避免地需要将网络划分成各种不同的区域。路

成时，汇总功能就成为必须理解的一个内容。这个实验里，大家有机会配置多 OSPF 区域、虚拟链路、重分布以及汇总。

12.15.2 实验内容

假定爬虫信息网络（简称 HIN）是爬虫学者们就一些爬虫类和两栖类物种交换相互的生物与生态信息的一个网络。HIN 拥有一个帧中继网络以用于在全国范围内发布相关的信息。网络一直处在不停的改变之中，并深受其苦。现在的任务是解决当前 OSPF 网络的问题并将其与一个 IGRP 网络集成到一起。网络设计的时候必须遵照下面这些要求：

- 按照图 12-26 配置一个 IP 网络，路由选择协议采用 OSPF，进程 ID 为 2001。
- 按照图中所示将整个帧中继网络配置成 3 个点对点网络。
- 配置图中所有的 OSPF 区域。路由器 hin_hq、gecko 与 tree_frog 之间的 2 个帧中继点对点网络都处在 OSPF Area 0 中，gecko 的 LAN 接口是在 OSPF Area 10 中，而 tree_frog 和 python 的 LAN 接口则是在 OSPF Area 20 中。路由器 python 的串行接口和 boa 的全部接口都在 OSPF Area 75 中。
- 将路由器 hin_hq 和 chameleon 之间的帧中继点对点网络配置到 IGRP 的路由域中去。确保从 chameleon 的 LAN 接口到所有 OSPF 网络与接口的 IP 连接畅通。
- 在路由器 chameleon 上配置 3 个环路接口，并将此作为一条单独的 OSPF 路由进行转发。
- 路由器 gecko 路由表中没有 boa 上 LAN 的 IP 网络。

12.15.3 实验目的

- 按照图 12-16 配置 HIN 网络以及相应的 IP，LAN 的拓扑类型对实验结果没用影响。
- 在 WAN 上使用帧中继数据链路协议。按照图中所示，只使用点对点网络。
- 确保所有 IP 接口的 IP 连通性——即从所有路由器上对帧中继和 LAN 接口进行 ping 测试，但不能使用任何形式的静态路由。
- 不要改变默认的 OSPF 网络类型。
- 在路由器 chameleon 上对 IGRP 域进行配置，确保这个网络区域与整个 OSPF 网络具有完全的 IP 连接性。
- 对路由器 chameleon 上的 3 个环路网络进行配置并转发，这 3 个网络是 10.1.16.0/24，10.1.17.0/24 和 10.1.18.0/24。将这 3 个网络聚合到一条 OSPF 路由中去。
- 禁止路由器 gecko 访问网络 10.1.70.1/24。
- （可选）对 OSPF 进行配置，路由器 tree_frog 每 30 秒发送一次 OSPF 的 hello 数据包。

12.15.4 所需设备

- 7 台 Cisco 路由器，其中 4 台要通过 V.35 背对背线缆或者是类似的方式与帧中继交换机连接在一起。

- 两台路由器，需要通过 V.35 背对背线缆或类似方式直接相连。
- 利用集线器或交换机构建 5 个 LAN 网段。这个实验中 LAN 的拓扑结构没有影响。

12.15.5 物理设计与实验准备

- 按照图 12-16 将集线器以及串行线缆与路由器连接起来。
- 此外，还需要一台具有 3 条 PVC 的帧中继交换机。例 12-47 是该交换机的配置范例，和先前实验一样。

例 12-47 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
 frame-relay route 121 interface Serial3 102
 frame-relay route 150 interface Serial5 151
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 110 interface Serial0 111
!
interface Serial2
 no ip address
 shutdown
!
interface Serial3
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 102 interface Serial0 121
!
interface Serial5
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 151 interface Serial0 150
```

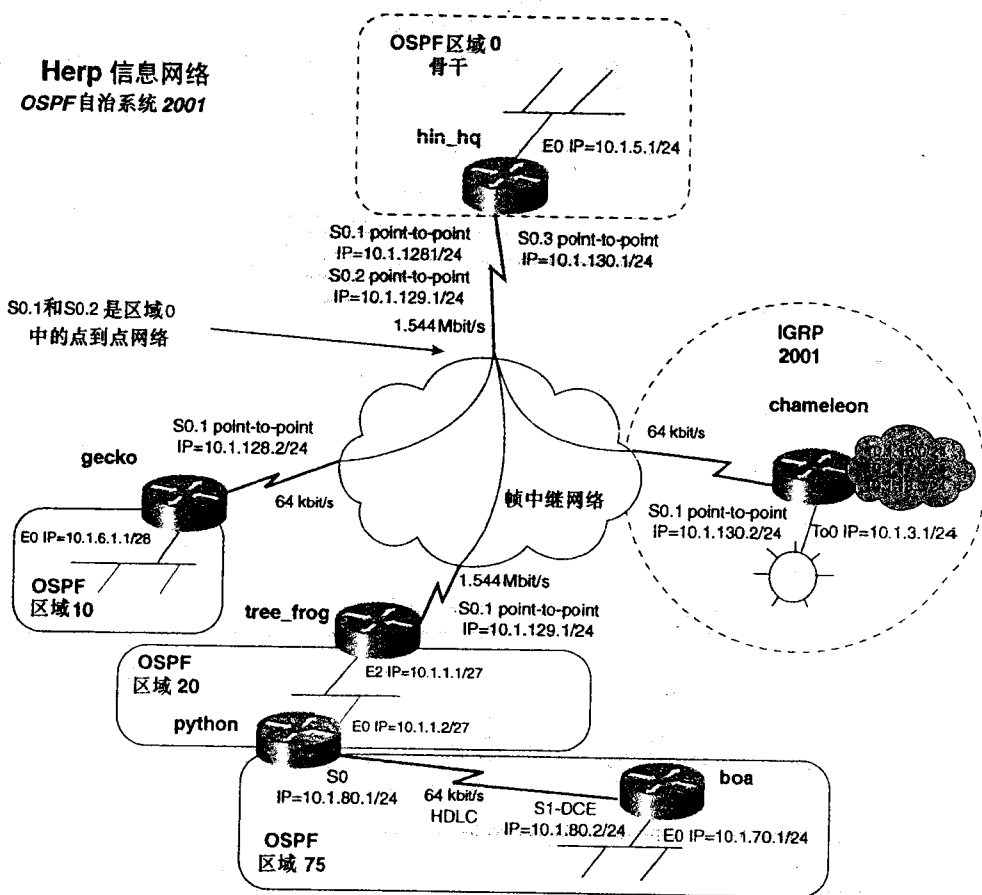


图 12-16 HIN 网络

12.16 实验 25：配置 OSPF：多域路由、路由的重分布与汇总功能——第 2 部分

12.16.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 4 台路由器以背对背方式连接在一起。然后按照图 11-16 利用交换机或集线器/MAU 创建 5 个 LAN 网段。物理连接完成之后，来按照图 11-16 所示为所有的 LAN 和 WAN 接口分配 IP 地址。在继续下面的操作之前，必须用 **ping** 命令对每台路由器的本地 LAN 和 WAN 接口进行测试。接着，在路由器 chameleon 上配置 3 个环路地址，分别是 10.1.16.0/24，10.1.17.0/24 和 10.1.18.0/24。在继续往下做之前，再用 **ping** 命令对每台路由器的本地 LAN 和 WAN 接口进行测试。再下来是在所有的点对点接口上用 **framerelay interface-dlci** 命令进行配置。例 12-48 是所有路由器

到目前为止所有相关的帧中继的配置情况。

例 12-48 帧中继的配置

```
hostname hin_hq
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
!
interface Serial0.1 point-to-point
 ip address 10.1.128.1 255.255.255.0
 frame-relay interface-dlci 121
!
interface Serial0.2 point-to-point
 ip address 10.1.129.1 255.255.255.0
 frame-relay interface-dlci 111
!
interface Serial0.3 point-to-point
 ip address 10.1.130.1 255.255.255.0
 frame-relay interface-dlci 150
!
```

```
hostname gecko
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 ip address 10.1.128.2 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 102
!
```

```
hostname tree_frog
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
!
interface Serial0.1 point-to-point
 ip address 10.1.129.2 255.255.255.0
 frame-relay interface-dlci 110
!
```

```
hostname chameleon
!
<<<text omitted>>>
!
interface Serial0
 no ip address
```

(续)

```

encapsulation frame-relay
frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
ip address 10.1.130.2 255.255.255.0
frame-relay interface-dlci 151
!

```

配置路由器 python 和 boa 之间 WAN 网络的时候，链路的一端需要对时钟进行设置，即 DCE 端的设置。例 12-49 是路由器 boa 上的串行配置情况，这是链路的 DCE 端。如果使用带有反接线缆的 CSU/DSU，则不需要这一设置。

例 12-49 路由器 boa 的串行设置

```

!
interface Serial1
ip address 10.1.80.2 255.255.255.0
clockrate 56000

```

LAN 和 WAN 接口配置完毕，基本 IP 连接建立之后，进行 OSPF 和 IGRP 的配置。首先来看看 OSPF 的配置，然后再把它集成到 IGRP 中去。回想一下 OSPF 配置的详细过程：

- 第 1 步 对网络区域以及 DR/BDR 进行配置。
- 第 2 步 在装有 12.0 及更新版本的 Cisco IOS 的路由器上利用环路接口设置 RID。
- 第 3 步 在装有 12.0 及更新版本的 Cisco IOS 的路由器上启动 OSPF 并指定 RID。
- 第 4 步 对 OSPF 接口进行配置。
- 第 5 步 如果需要，配置额外的邻居支持。
- 第 6 步 配置 OSPF 区域类型以及虚链路。
- 第 7 步 对其他一些 OSPF 功能，如汇总和重分布进行配置。

第 1 步是对网络区域的配置。在这个实验中，帧中继点对点网络处在 OSPF Area 0 中，路由器 gecko 的 LAN 接口在 OSPF Area 10 中，路由器 tree_frog 和 python 的 LAN 接口是在 OSPF Area 20 中，路由器 python 的串行接口和 boa 的所有接口则是在 OSPF Area 75 中。此时可能大家都可以看到，还需要一条虚链路连接 Area 75 与剩下的 OSPF 网络。

第 2 步是在装有 12.0 及更新版本的 Cisco IOS 路由器上对 RID 进行指定，这是通过使用这些路由器上的环路接口来实现的。图 12-17 是分配了 RID，标记了虚链路后的网络图。

第 3 步是真正配置的开始。在除 chameleon 路由器以外的所有路由器上使用 **router ospf 2001** 命令，把 OSPF 设置在 AS 2001 中。在装有 12.0 版本的 Cisco IOS 的路由器上采用路由器命令 **router-id ip_address**，为路由器静态指定 RID。

第 4 步是定义哪些接口将要参与 OSPF 路由转发以及这些路由接口所处的区域。在路由器 hin_hq 上，E0 接口会和点对点接口 s0.1 和 s0.2 一起处在 Area 0 中。例 12-50 是路由器 hin_hq 上到目前为止的 OSPF 配置。

```
encapsulation frame-relay
frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
ip address 10.1.130.2 255.255.255.0
frame-relay interface-dlci 151
!
```

配置路由器 python 和 boa 之间 WAN 网络的时候，链路的一端需要对时钟进行设置，即 DCE 端的设置。例 12-49 是路由器 boa 上的串行配置情况，这是链路的 DCE 端。如果使用带有反接线缆的 CSU/DSU，则不需要这一设置。

例 12-49 路由器 boa 的串行设置

```
!
interface Serial1
ip address 10.1.80.2 255.255.255.0
clockrate 56000
```

LAN 和 WAN 接口配置完毕，基本 IP 连接建立之后，进行 OSPF 和 IGRP 的配置。首先来看看 OSPF 的配置，然后再把它集成到 IGRP 中去。回想一下 OSPF 配置的详细过程：

- 第 1 步 对网络区域以及 DR/BDR 进行配置。
- 第 2 步 在装有 12.0 及更新版本的 Cisco IOS 的路由器上利用环路接口设置 RID。
- 第 3 步 在装有 12.0 及更新版本的 Cisco IOS 的路由器上启动 OSPF 并指定 RID。
- 第 4 步 对 OSPF 接口进行配置。
- 第 5 步 如果需要，配置额外的邻居支持。
- 第 6 步 配置 OSPF 区域类型以及虚链路。
- 第 7 步 对其他一些 OSPF 功能，如汇总和重分布进行配置。

第 1 步是对网络区域的配置。在这个实验中，帧中继点对点网络处在 OSPF Area 0 中，路由器 gecko 的 LAN 接口在 OSPF Area 10 中，路由器 tree_frog 和 python 的 LAN 接口是在 OSPF Area 20 中，路由器 python 的串行接口和 boa 的所有接口则是在 OSPF Area 75 中。此时可能大家都可以看到，还需要一条虚链路连接 Area 75 与剩下的 OSPF 网络。

第 2 步是在装有 12.0 及更新版本的 Cisco IOS 路由器上对 RID 进行指定，这是通过使用这些路由器上的环路接口来实现的。图 12-17 是分配了 RID，标记了虚链路后的网络图。

第 3 步是真正配置的开始。在除 chameleon 路由器以外的所有路由器上使用 **router ospf 2001** 命令，把 OSPF 设置在 AS 2001 中。在装有 12.0 版本的 Cisco IOS 的路由器上采用路由器命令 **router-id ip_address**，为路由器静态指定 RID。

第 4 步是定义哪些接口将要参与 OSPF 路由转发以及这些路由接口所处的区域。在路由器 hin_hq 上，E0 接口会和点对点接口 s0.1 和 s0.2 一起处在 Area 0 中。例 12-50 是路由器 hin_hq 上到目前为止的 OSPF 配置。

Herp 信息网络
OSPF 自治系统 2001

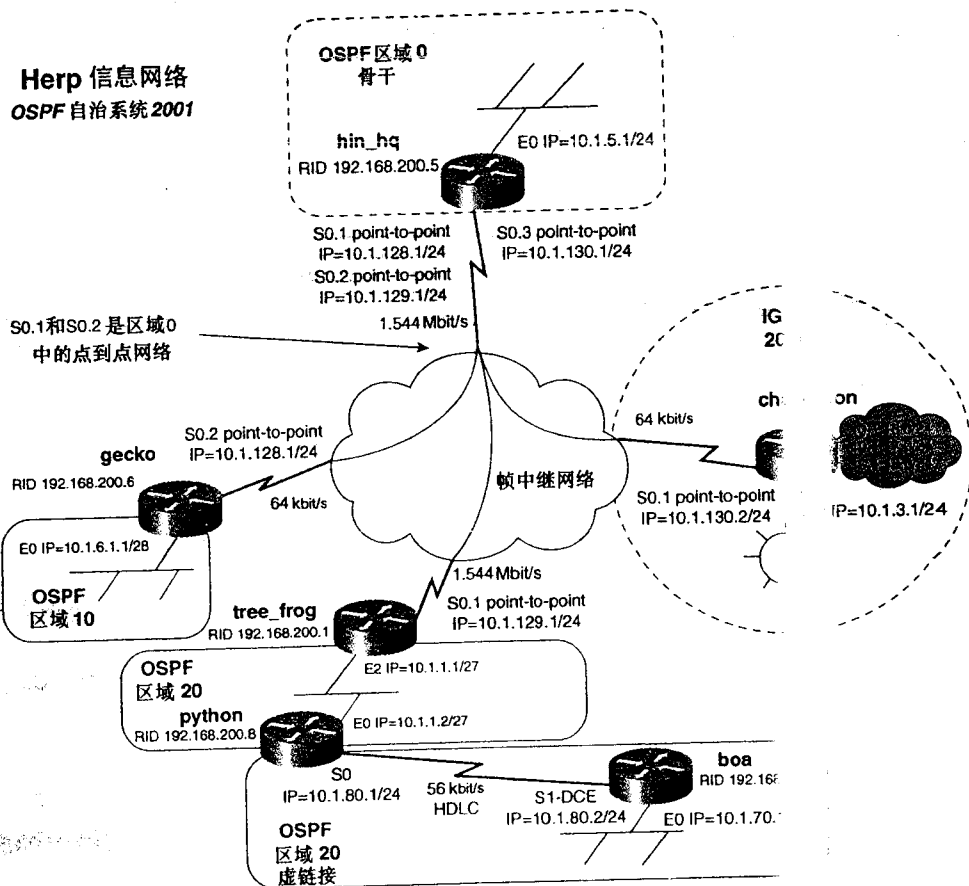


图 12-17 分配了 RID 之后的 HIN 网络拓扑

例 12-50 路由器 hin_hq 的初始 OSPF 配置

```
router ospf 2001
router-id 192.168.200.5
redistribute igrp 2001 subnets tag 5
network 10.1.5.1 0.0.0.0 area 0
network 10.1.128.1 0.0.0.0 area 0
network 10.1.129.1 0.0.0.0 area 0
default-metric 64
!
```

上个例子中的 **network** 命令可简化为一条命令——**network 10.1.0.0**。在简化网络中，将掩码设置得具体一些是值得的。网络变化太快，不同随时会添加进来。如果掩码太过泛泛，就需要配置新的、更加具体的，OSPF 就会暂时中断工作。

路由器 **gecko** 的 LAN 接口是处在 OSPF Area 10 中的，而串行接口中。路由器 **tree_frog** 的 LAN 接口是在 OSPF Area 20 中，而串行接口则

55.255 area 0。
区域的新的接口
ork 命令，这样

在 OSPF Area
OSPF Area 0 中

例 12-51 路由器 gecko 和 tree_frog 上的初始 OSPF 配置

```
hostname gecko
!
router ospf 2001
  router-id 192.168.200.6
  network 10.1.6.1 0.0.0.0 area 10
  network 10.1.128.2 0.0.0.0 area 0
!

hostname tree_frog
!
router ospf 2001
  network 10.1.1.1 0.0.0.0 area 20
  network 10.1.129.2 0.0.0.0 area 0
!
```

注释 路由器没有显示指定其 RID，这是因为该路由器将一个环路接口做为它的 RID。路由器 tree_frog 上的 Cisco IOS 是 12.0 以前版本。

路由器 python 有两个分别处在 Area 20 和 Area 75 中的接口，而 boa 的两个接口都在 Area 75。例 12-52 是这些路由器的 OSPF 配置情况。

例 12-52 路由器 python 和 boa 的初始 OSPF 配置

```
hostname python
!
router ospf 2001
  network 10.1.80.1 0.0.0.0 area 75
  network 10.1.1.2 0.0.0.0 area 20
!

hostname boa
!
router ospf 2001
  network 10.1.0.0 0.0.255.255 area 75
!
```

第 5 步是要提供额外的一些邻居路由器的支持。在这个网络模型中，帧中继点对点网络上的邻接关系会自动形成。无需做额外配置。

到现在，除了 Area 75 之外，OSPF 就已经完全可以工作了，每台路由器的 IP 连接也已经建立起来。可以检查邻居路由器和路由表的情况加以验证，还可以通过标准的 ping 命令来进行。例 12-53 是路由器 hin_hq、gecko、tree_frog 和 python 上 show ip ospf neighbor 命令的结果。

例 12-53 OSPF 邻居路由器的验证

```
hin_hq#show ip ospf neighbor

Neighbor ID    Pri   State           Dead Time   Address        Interface
192.168.200.6  1     FULL/-          00:00:35   10.1.128.2     Serial0/1
```



```
192.168.200.1 1 FULL/ - 00:00:32 10.1.129.2 Serial0.2
hin_hq#

tree_frog#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
192.168.200.8 1 FULL/DR 00:00:35 10.1.1.2 Ethernet2
192.168.200.5 1 FULL/ - 00:00:34 10.1.129.1 Serial0.1
tree_frog#

python#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
192.168.200.1 1 FULL/BDR 00:00:39 10.1.1.1 Ethernet0
192.168.200.7 1 FULL/ - 00:00:35 10.1.80.2 Serial0
python#
```

第 6 步是 OSPF 特殊区域（如存根区域）的配置——这个例子中是配置虚链路。前面提过，Area 75 和其他 OSPF 区域的连接需要采用一条虚链路。配置虚链路的时候，首先要找到用作链路两个端点的路由器。在这个模型中，Area 20 是传输区域，因此虚链路应该定义在路由器 tree_frog 和 python 上。路由器 tree_frog 上的命令句法是 **area 20 virtual-link 192.168.200.8**，这是路由器 python 的 RID。而路由器 python 又指向 tree_frog 的 RID，所采用的命令是 **area 20 virtual-link 192.168.200.1**。

静态指定 RID 对 OSPF 虚链路的工作至关重要。如果虚链路无法正常进入工作状态，而且又可以确认配置没有错误的，可以试一试重新启动路由器。虚链路常见的一个问题是 RID，在 12.0 以前版本的 Cisco IOS 中，重新启动将重新分配 RID。要检查虚链路是否在工作状态，可以采用 **show ip ospf virtual-link** 命令。来自路由器 boa 的路由也应该重新出现在整个 OSPF 区域中。例 12-54 是路由器 tree_frog 上命令 **show ip ospf virtual-link** 的示例。

例 12-54 虚链路的验证

```
tree_frog#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 192.168.200.8 is up
Run as demand circuit
DoNotAge LSA allowed.
Transit area 20, via interface Ethernet2, Cost of using 10
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:00
Adjacency State FULL (Hello suppressed)
tree_frog#
```

现在整个 OSPF 区域就完全建立起了应有的功能。通过检查路由器 boa 的路由表，就可以确认网络中每个目的地址都有了相应的路由，如例 12-55 所示。

例 12-55 OSPF 区域的验证

```
boa#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, * - candidate default
U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
O IA 10.1.1.0/27 [110/74] via 10.1.80.1, 00:28:02, Serial1
O IA 10.1.6.0/28 [110/212] via 10.1.80.1, 00:28:02, Serial1
O IA 10.1.5.0/24 [110/148] via 10.1.80.1, 00:28:02, Serial1
C 10.1.70.0/24 is directly connected, Ethernet0
C 10.1.80.0/24 is directly connected, Serial1
O IA 10.1.129.0/24 [110/138] via 10.1.80.1, 00:28:02, Serial1
O IA 10.1.128.0/30 [110/266] via 10.1.80.1, 00:28:02, Serial1
O IA 10.1.128.0/24 [110/202] via 10.1.80.1, 00:28:02, Serial1
C 192.168.200.0/24 is directly connected, Loopback0
boa#
```

要想与 IGRP 域相互集成，还需要在路由器 chameleon 和 hin_hq 上对 OSPF 进行配置。例 12-56 是路由器 hin_hq 上的 IGRP 配置情况。记住要使用 **passive-interface** 以防止以太接口以及其他串行接口上的一些不必要的广播。路由器 hin_hq 和 chameleon 的 IGRP 配置完全相同。

例 12-56 路由器 hin_hq 上的 IGRP 配置

```
!
router igrp 2001
passive-interface Ethernet0
passive-interface Serial0.1
passive-interface Serial0.2
network 10.0.0.0
!
```

接下来配置 IGRP 和 OSPF 之间的重分布。网络中只有一个重分布点，因而不必担心路由反馈或重分布环路的情况发生。例 12-57 是路由器 hin_hq 上的相关配置部分，突出显示的是重分布命令。这个例子使用的 OSPF 默认路由度量是 64，这是一个 T1 接口的路由代价。在将多个子网重分布进 OSPF 的时候还需要使用 **subnets** 关键字。

例 12-57 路由器 hin_hq 的重分布

```
!
router ospf 2001
router-id 192.168.200.5
redistribute igrp 2001 subnets tag 5      ←Redistribute IGRP into OSPF
network 10.1.5.1 0.0.0.0 area 0
network 10.1.128.1 0.0.0.0 area 0
network 10.1.129.1 0.0.0.0 area 0
default-metric 64      ←Default metric or cost
!
router igrp 2001
redistribute ospf 2001      ←Redistribute OSPF into IGRP
passive-interface Ethernet0
passive-interface Serial0.1
```

(待续)

子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责。

```

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 12 subnets
C       10.1.3.0 is directly connected, TokenRing0
I       10.1.1.0 [100/8486] via 10.1.130.1, 00:00:28, Serial0.1
I       10.1.6.0 [100/8486] via 10.1.130.1, 00:00:28, Serial0.1
I       10.1.5.0 [100/8576] via 10.1.130.1, 00:00:28, Serial0.1
C       10.1.18.0 is directly connected, Loopback22
C       10.1.17.0 is directly connected, Loopback21
C       10.1.16.0 is directly connected, Loopback20
I       10.1.70.0 [100/8486] via 10.1.130.1, 00:00:28, Serial0.1
I       10.1.80.0 [100/8486] via 10.1.130.1, 00:00:29, Serial0.1
C       10.1.130.0 is directly connected, Serial0.1
I       10.1.129.0 [100/10476] via 10.1.130.1, 00:00:29, Serial0.1
I       10.1.128.0 [100/10476] via 10.1.130.1, 00:00:29, Serial0.1
chameleon#

```

如果还没有在路由器 chameleon router 配置环路接口，那现在就可以完成这一工作了。希望路由器 hin_hq 把路由 10.1.16.0/24, 10.1.17.0/24 和 10.1.18.0/24 作为一条单一的 OSPF 路由进行发送。要把来自其他网络的路由进行汇总，可以采用 OSPF 命令 **summary-address**。这里的汇总地址是 10.1.16.0，掩码则是 255.255.252.0。例 12-60 是现在路由器 hin_hq 的 OSPF 配置。

例 12-60 路由器 hin_hq 的 OSPF 配置

```

router ospf 2001
router-id 192.168.200.5
summary-address 10.1.16.0 255.255.252.0
redistribute igrp 2001 subnets tag 5
network 10.1.5.1 0.0.0.0 area 0
network 10.1.128.1 0.0.0.0 area 0
network 10.1.129.1 0.0.0.0 area 0
default-metric 64
!
router igrp 2001
redistribute ospf 2001
passive-interface Ethernet0
passive-interface Serial0.1
passive-interface Serial0.2
network 10.0.0.0
default-metric 1544 10 254 1 1500
!

```

在路由器 boa 上列出路由表可以查看汇总路由的情况，如例 12-61 所示。

例 12-61 路由器 boa 的路由表，突出显示部分是汇总路由

```

boa#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, * - candidate default
       U - per-user static route, o - ODR

```

```

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 12 subnets, 4 masks
O E2   10.1.3.0/24 [110/64] via 10.1.80.1, 00:37:49, Serial1
O IA   10.1.1.0/27 [110/74] via 10.1.80.1, 00:27:57, Serial1
O IA   10.1.1.0/24 [110/84] via 10.1.80.1, 00:27:57, Serial1
O IA   10.1.6.0/24 [110/212] via 10.1.80.1, 00:28:57, Serial1
O IA   10.1.5.0/24 [110/148] via 10.1.80.1, 01:13:37, Serial1
O E2   10.1.16.0/22 [110/64] via 10.1.80.1, 00:17:23, Serial1
C      10.1.70.0/24 is directly connected, Ethernet0
C      10.1.80.0/24 is directly connected, Serial1
O E2   10.1.130.0/24 [110/64] via 10.1.80.1, 00:37:50, Serial1
O IA   10.1.129.0/24 [110/138] via 10.1.80.1, 01:13:37, Serial1
O IA   10.1.128.0/30 [110/266] via 10.1.80.1, 01:13:37, Serial1
O IA   10.1.128.0/24 [110/202] via 10.1.80.1, 01:13:37, Serial1
C      192.168.200.0/24 is directly connected, Loopback0
boa#

```

实验的最后一个部分是禁止将路由器 boa 的 LAN 网络 10.1.3.0/24 发送到 gecko，可以通过在路由器 gecko 上配置分布列表来实现。例 12-62 是在路由器 gecko 上的配置情况。

例 12-62 在路由器 gecko 上配置分布列表

```

gecko(config)#router ospf 2001
gecko(config-router)#distribute-list 10 in s0.1 Applied to s0.1
gecko(config-router)#exit
gecko(config)#access-list 10 deny 10.1.70.0 0.0.0.255 deny 10.1.70.0/24
gecko(config)#access-list 10 permit any
gecko(config)#

```

为了验证访问列表的功能，可以清除路由器 gecko 的路由表，然后再查看路由表。例 12-63 是应用了分布列表之后路由器 gecko 的路由表。

例 12-63 路由器 gecko 上的路由表

```

gecko#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

    10.0.0.0/8 is variably subnetted, 11 subnets, 5 masks
O E2   10.1.3.0/24 [110/64] via 10.1.128.1, 00:00:03, Serial0.1
O IA   10.1.1.0/27 [110/148] via 10.1.128.1, 00:00:03, Serial0.1
O IA   10.1.1.0/24 [110/138] via 10.1.128.1, 00:00:03, Serial0.1
C      10.1.6.0/28 is directly connected, Ethernet0
O      10.1.5.0/24 [110/74] via 10.1.128.1, 00:00:04, Serial0.1
O E2   10.1.16.0/22 [110/64] via 10.1.128.1, 00:00:03, Serial0.1
O IA   10.1.80.0/24 [110/202] via 10.1.128.1, 00:00:04, Serial0.1

```

```
O E2 10.1.130.0/24 [110/64] via 10.1.128.1, 00:00:04, Serial0.1
O 10.1.129.0/24 [110/128] via 10.1.128.1, 00:00:04, Serial0.1
O 10.1.128.0/24 [110/128] via 10.1.128.1, 00:00:04, Serial0.1
C 10.1.128.0/24 is directly connected, Serial0.1
gecko#
```

本实验的可选部分是改变路由器 `tree_frog` 上发送 hello 数据包的时间间隔。关于这一点，需要在路由器 `tree_frog` 的 E2 接口上利用 `ip ospf hello-interval` 命令来配置。如果只在一个接口更改了这个时间值，路由器邻居关系会失效，而路由的转发也会出问题。在改变 OSPF 的计时间隔值的时候，记住一定要在同一 IP 网络中所有的路由器上进行。在这个网络模型中，需要更改路由器 `tree_frog` 和 `python` 上计时间隔的值。例 12-64 是路由器 `tree_frog` 上计时间隔的配置。

例 12-64 改变某个接口上 OSPF Hello 数据包的发送时间间隔

```
tree_frog(config)#int e2
tree_frog(config-if)#ip ospf hello-interval 30
tree_frog(config-if)#
```

为了确定计时时间的配置效果，可以检查 OSPF 邻居路由器的状态，并在该接口上执行一条 `show ip ospf interface` 命令。例 12-65 列出了计时间隔改变之后，路由器 `tree_frog` 上 OSPF 接口的情况。OSPF 的消亡计时器和等待计时器的值则会自动地调整为 hello 计时器的 4 倍。

例 12-65 观察路由器 `tree_frog` 上计时间隔改变

```
tree_frog#show ip ospf interface e2
Ethernet2 is up, line protocol is up
Internet Address 10.1.1.1/27, Area 20
Process ID 2001, Router ID 192.168.200.1, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 192.168.200.8, Interface address 10.1.1.2
Backup Designated router (ID) 192.168.200.1, Interface address 10.1.1.1
Timer intervals configured: Hello 30, Dead 120, Wait 120, Retransmit 6
Hello due in 00:00:11
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 192.168.200.8 (Designated Router)
Suppress hello for 0 neighbor(s)
tree_frog#
```

例 12-66 总结了整个实验步骤，给出了所有路由器的配置清单。

例 12-66 HIN 网络的最终配置

```
hostname hin_hq
!
<<<text omitted>>>
!
interface Ethernet0
ip address 10.1.5.1 255.255.255.0
!
```

```

interface Serial0
  no ip address
  encapsulation frame-relay
!
interface Serial0.1 point-to-point
  ip address 10.1.128.1 255.255.255.0
  frame-relay interface-dlci 121
!
interface Serial0.2 point-to-point
  ip address 10.1.129.1 255.255.255.0
  frame-relay interface-dlci 111
!
interface Serial0.3 point-to-point
  ip address 10.1.130.1 255.255.255.0
  frame-relay interface-dlci 150
!
router ospf 2001
  router-id 192.168.200.5
  summary-address 10.1.16.0 255.255.252.0
  redistribute igmp 2001 subnets tag 5
  network 10.1.5.1 0.0.0.0 area 0
  network 10.1.128.1 0.0.0.0 area 0
  network 10.1.129.1 0.0.0.0 area 0
  default-metric 64
!
router igmp 2001
  redistribute ospf 2001
  passive-interface Ethernet0
  passive-interface Serial0.1
  passive-interface Serial0.2
  network 10.0.0.0
  default-metric 1544 10 254 1 1500
!
ip classless
!

```

```

hostname gecko
!
<<<text omitted>>>
!
interface Ethernet0
  ip address 10.1.6.1 255.255.255.240
  no ip directed-broadcast
!
interface Serial0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay
  no ip mroute-cache
  frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
  ip address 10.1.128.2 255.255.255.0

  frame-relay interface-dlci 102
!
<<<text omitted>>>
!
router ospf 2001
  router-id 192.168.200.6
  area 10 range 10.1.6.0 255.255.255.0
  network 10.1.6.1 0.0.0.0 area 10

```

(待续)

```

network 10.1.128.2 0.0.0.0 area 0
distribute-list 10 in Serial0.1
!
ip classless
!

hostname tree_frog
!
interface Loopback0
ip address 192.168.200.1 255.255.255.0
!
interface Ethernet0
no ip address
shutdown
media-type 10BaseT
!
interface Ethernet1
no ip address
shutdown
media-type 10BaseT
!
interface Ethernet2
ip address 10.1.1.1 255.255.255.224
ip ospf hello-interval 30
media-type 10BaseT
!
interface Serial0
no ip address
encapsulation frame-relay
no ip mroute-cache
!
interface Serial0.1 point-to-point
ip address 10.1.129.2 255.255.255.0
frame-relay interface-dlci 110
!
<<<text omitted>>>
!
router ospf 2001
network 10.1.1.1 0.0.0.0 area 20
network 10.1.129.2 0.0.0.0 area 0
area 20 range 10.1.1.0 255.255.255.0
area 20 virtual-link 192.168.200.8
!
ip classless
!

hostname python
!
<<<text omitted>>>
!
interface Loopback0
ip address 192.168.200.8 255.255.255.0
!
interface Ethernet0
ip address 10.1.1.2 255.255.255.224
ip ospf hello-interval 30
!
interface Serial1
ip address 10.1.80.1 255.255.255.0
!
<<<text omitted>>>
!

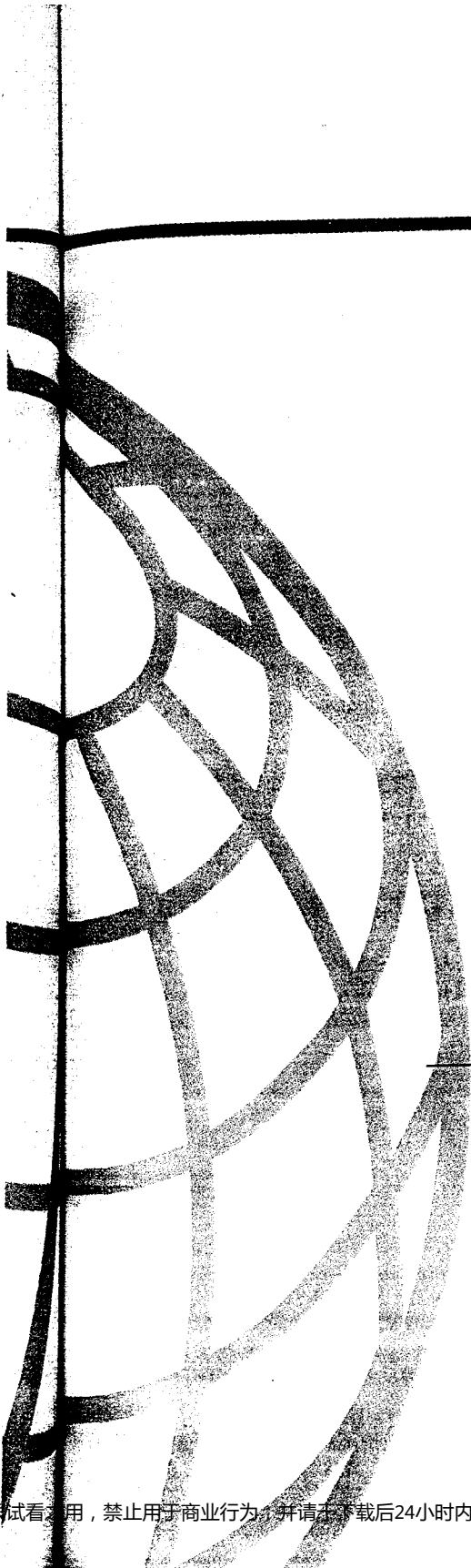
```

(待续)


```
router ospf 2001
 network 10.1.80.1 0.0.0.0 area 75
 network 10.1.1.2 0.0.0.0 area 20
 area 20 virtual-link 192.168.200.1
!
ip classless
!

hostname boa
!
<<<text omitted>>>
!
interface Loopback0
 ip address 192.168.200.7 255.255.255.0
!
interface Ethernet0
 ip address 10.1.70.1 255.255.255.0
!
interface Serial1
 ip address 10.1.80.2 255.255.255.0
 clockrate 56000
!
router ospf 2001
 network 10.1.0.0 0.0.255.255 area 75
!
ip classless
!

hostname chameleon
!
<<<text omitted>>>
!
interface Loopback20
 ip address 10.1.16.1 255.255.255.0
!
interface Loopback21
 ip address 10.1.17.1 255.255.255.0
!
interface Loopback22
 ip address 10.1.18.1 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type cisco
!
interface Serial0.1 point-to-point
 ip address 10.1.130.2 255.255.255.0
 frame-relay interface-dlci 151
!
interface TokenRing0
 ip address 10.1.3.1 255.255.255.0
 ring-speed 16
!
router igrp 2001
 network 10.0.0.0
!
ip classless
```



第5部分

不可路由协议的 传输

第13章 配置桥接和增强数据链路交换 (DLSW+)

第 13 章

配置桥接和增强数据链路交换 (DLSW+)

多数早期协议在设计时都没有明确的网络地址。因此，这些协议并没有现在所熟知的传统的第 3 层网络的概念。根据定义，没有明确的网络层地址的协议称为不可路由协议或桥接协议。常见的桥接协议包括 IBM 的系统网络体系结构 (SNA)、NetBEUI、NetBIOS 和 DEC LAT。

现在常用的桥接协议有 SNA 和 NetBEUI。IBM 的 SNA 现在还用于很多大型数据中心，也可能是现有的实际应用中最常见的桥接协议。随着 Windows 9x 和 Microsoft Networking 的出现，NetBEUI 也开始出现在很多网络应用中。

伴随着 SNA、NetBEUI 和其他不可路由协议的应用，出现了一个必然的灾难——即将这些协议通过很多 LAN 网段和 WAN 网段进行传输。称此为必然的灾难是因为网桥数据传输形式是极端密集广播方式。在网络中启动网桥会对网络性能造成非常严重的影响。为此，在设置任何形式的网桥时，一定要对网桥加以控制或对使用网桥的网段加以限制。

本章用于传输不可路由协议的方法包括：

- 透明桥接 (TB)。
- 综合路由桥接 (IRB)。
- 源路由桥接 (SRB)。
- 远程源路由桥接 (RSRB)。
- 数据链路交换 (DLSw)。

13.1 透明桥接（Transparent Bridging）

透明桥接用于在以网络中传输不可路由协议。透明桥接最早由 DEC 在 1980 年代早期提出，并将其提交给 IEEE，IEEE 把该技术整理成 IEEE 802.1 标准。

网桥的一个基本功能是在网络中转发数据。网桥接收到数据帧，经过简单检查后，根据数据帧中的信息做出转发决定。网桥通过建立桥接表或工作站表完成转发任务。图 13-1 给出了桥接网络中桥接表的例子。

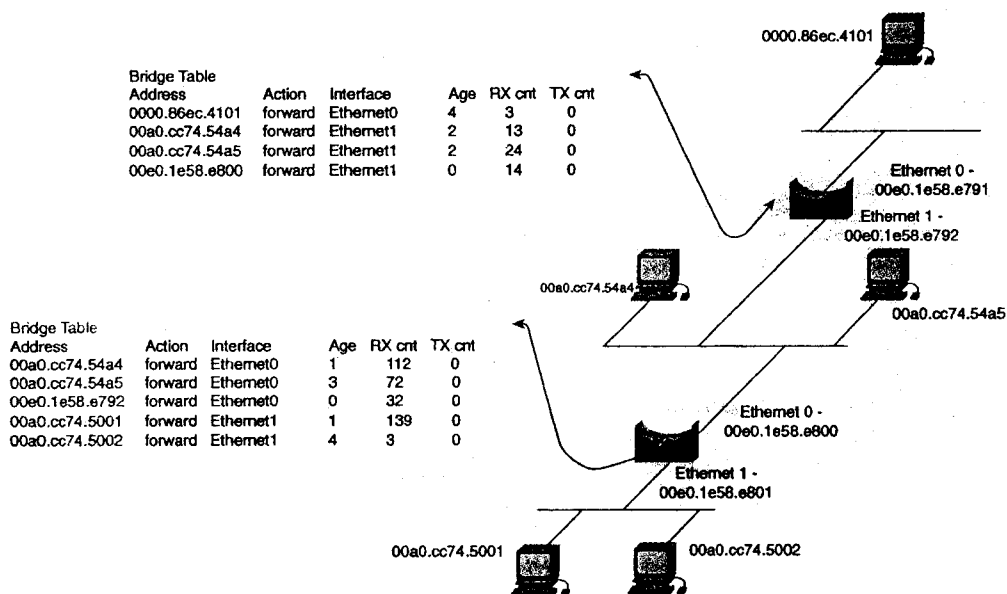


图 13-1 桥接网络

网桥工作在 OSI 模型的下面两层中。第 2 章“LAN 协议：Catalyst 以太和令牌环交换机的配置”中讲过，第 2 层（数据链路层）又分为两个子层：MAC 层和 LLC 层。网桥主要运行在 MAC 层，通过源 MAC 地址和目的 MAC 地址工作。

13.1.1 透明桥接的工作

网桥基本上按下列方式工作：

- 1 初始化时，透明桥接以混杂模式侦听网络中的数据帧。
- 2 接口收到数据帧时，数据帧中的源 MAC 地址以及接收该数据帧的接口/端口信息被记录到工作站缓存或桥接表中。桥接表跟踪网桥检测到的所有 MAC 地址和地址所在端口信息。
- 3 接口再接收到后续数据帧时，网桥检查数据帧中的目的 MAC 地址，将此地址与桥接表或工作站缓存中的地址进行比较，然后据此选择进行下列操作之一：

- 如果 MAC 地址位于接收到该数据帧的网络或接口，则网桥不转发该数据帧，并将其丢弃。
- 如果 MAC 地址在桥接表中，网桥会将此数据帧转发到表中指定接口/端口上。
- 如果 MAC 地址不在桥接表，网桥将该数据帧转发到除接收到该数据帧端口以外的所有端口。

4 网桥工作站缓存表中的每一项在经过一段时间后都会失效并被从表中删除，这一段时间称为最长存活 (MAX Age) 计时器。MAX Age 计时器在计时结束时如果没有接收到和桥接表中 MAC 相匹配的源 MAC 地址时，会刷新桥接表中的相关条目。

1. 生成树 (STP) 回顾

由于网桥是将数据帧从一个网段转发到另一个网段，因此需要用某种方式对环路加以控制。Cisco 路由器提供了 3 种环路预防机制：

- IEEE 802.1D Spanning-Tree Protocol (STP)。即第 2 章中曾详细讨论过的生成树协议。
- 以 IEEE 标准为基础的数字协议。
- 用于令牌环网络上继承透明桥接的 IBM 的 STP。

所有形式的 STP 都很近似，这里将侧重讲述最主要的 802.1d (Cisco 交换机默认方式)。

以下内容摘自第 2 章，可以再回顾一下该章以便更深入地了解 STP 的相关内容。另外建议大家参考 Radia Perlman 的《Interconnections: Bridges and Routers》一书。现在看一下第 2 章中讲过的 STP 通过图 13-2 所示的各个阶段进行转换的过程，接下来再讲一下 STP 状态方面的内容。

2. 失效状态 (Disabled state)

当 Trunk 配置不当或是端口被手动关闭时，网桥由于处理 BPDU 出现问题就导致 STP 进入该状态。

3. 侦听状态 (Listening state)

网桥端口初始化或一定时间内没有接收到 BPDU 时，STP 进入侦听状态。STP 处于该状态时，端口实际上是“阻塞”的，该链路无用户数据发送。STP 按下面 4 个步骤收敛：

1 选举一个根桥——初始化时，网桥在所有的桥接接口上发送 BPDU，此时，网桥 ID (BID) 最小者成为根桥。前面讲过，BID 是一个优先级和 MAC 地址的组合。如果 BID 相等，MAC 地址最低者成为根桥。根桥的所有端口都会进入转发状态。

2 为每个非根桥选根端口——选定根桥后，STP 还会在每个非根的网桥/交换机上选出一个根端口。根端口选出后会进入转发状态。STP 选择根端口时依据的标准依次（按优先级次序）是：

- 根 BID 最低。
- 到根桥接的路径和到根的所有路由代价之和最小。
- 发送方 BID 最小。
- 端口 ID 最小。

接收到 BPDU 时，网桥会将其保存在该端口的桥接表中。该端口上再接收到新的 BPDU 时，网桥会将这些 BPDU 和已有的 BPDU 进行比较。按照上面所提的 4 步骤过程，网桥保留路径代价最低的 BPDU，丢弃其他 BPDU。影响根端口选择的主要因素是通往根网桥的路径

代价，这是指所有连接到根网桥的路径代价之和。

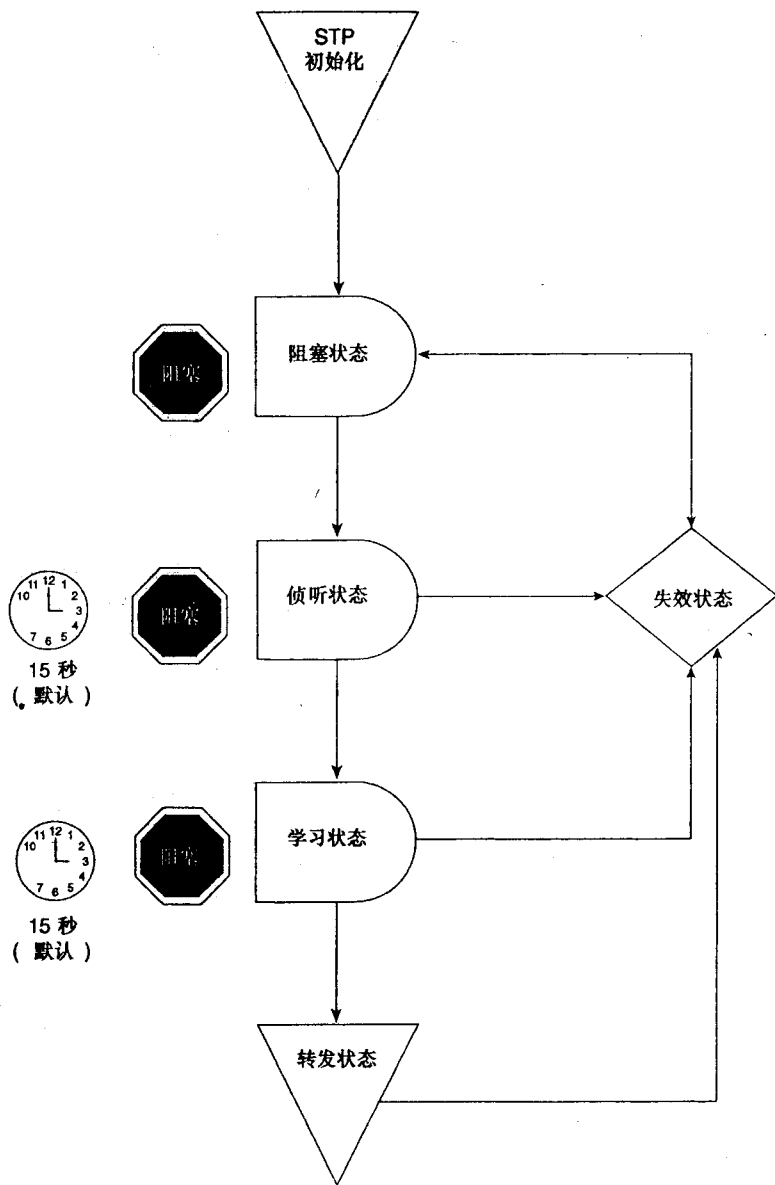


图 13-2 STP 的状态转换

3 为每个网段选出一个指定端口或指定网桥——STP 会为每个网段选出一个端口，该网段与根网桥之间所有的信息收发都是通过这个指定端口来进行的。可以把根端口看作是往根端口转发信息的端口，而指定端口则是将数据从根部传送出去的端口。这样的规则主要适用于共享媒体网桥或者是路由器，但是不适用于交换 Trunk 线路上的指定端口。

4 所有剩下的端口都会成为非指定端口从而进入阻塞状态。

4. 学习状态

指定端口或根端口保持连续 15 秒（默认的转发延时值）后会进入学习状态。在该状态中，网桥会再等待 15 秒以便建立自己的桥接表。

5. 转发和堵塞状态

网桥进入该阶段之后，那些既不是根端口，也不是指定端口，因而没有特殊用途的端口称为非指定端口。所有的指定端口都会进入转发状态，而所有的非指定端口则会进入阻塞状态。在阻塞状态中，网桥不会发送任何配置性 BPDU，但仍然会对这些 BPDU 进行侦听。阻塞的端口也不会转发任何用户数据。

6. STP 的计时器

STP 有 3 种基本计时器用于 BPDU 的管理：

- **Hello 计时器 (Hello timer)** —— Hello 计时器的默认时间长度是 2 秒，是根网桥发送出来的配置 BPDU 之间的时间间隔。
- **转发延时 (Forward delay)** —— 转发延时的默认值是 15 秒，即路由器或网桥在创建桥接表时的等待时间。侦听和学习状态都要用到这个计时器。
- **最大生存时间 (MAX Age)** —— 该计时器的时间长度是 BPDU 在刷新前保存的时间。如果某个接口在接收到新的 BPDU 之前该计时器的值溢出，那么接口进入侦听状态。

MAX Age 的值溢出通常是由连接失败造成的。该计时器的默认值是 20 秒。

STP 利用 hello 计时器来隔离不同的 BPDU，并且还使用了存活 (keepalive) 机制。Hello 计时器总是试图避免 MAX Age 计时器的值溢出，如果溢出，通常是发生了链路故障。此时，网桥进入侦听状态。STP 从一次链路故障中恢复大约需要 50 秒，BPDU 的 MAX Age 溢出要 20 秒，侦听状态停留 15 秒，而学习状态也是 15 秒。

13.1.2 透明桥接的配置

配置透明桥接遵循简单的 3 步骤过程：

第 1 步 通过下面这条全局命令分配一个桥组号码，定义生成树协议 (STP)：

```
Router (config) # bridge-group [ 1-255] protocol [ieee | ibm | dec]
```

第 2 步 使用下面的接口命令将需要桥接的每个网络接口分配至指定桥组中：

```
Router (config-if) # bridge-group [ 1-255]
```

如果接口是帧中继的多点接口，需要用接口 **frame-relay map** 命令将网桥映射到一个 DLCI 上，该命令的句法结构是：

```
Router (config-if) # frame-relay map bridge [ DLCI Number_16-1007] broadcast
```

如果接口是 DDR 接口（如 ISDN 接口），需要用 **dialer-map** 命令在 DDR 链路上传输网桥数据，命令格式为：

```
Router (config-if) # dialer-map bridge {name { remote_host_name}} broadcast  
dialer_string
```

第 3 步 （可选）配置生成树的根，选出要作为根的网桥或接口。前面讲过，影响根的选择的因素有很多，最好最直接的就是设置 STP 的优先级。可以根据自己想要的改变根的方式，可以选择在接口上或在全局范围内设置 STP 的优先级。

优先级越低，网桥就越有可能成为根网桥。下面这条命令可用来在接口或全局

配置模式下改变网桥优先级，从而影响 STP 根网桥的选择：

```
Router (config) # bridge-group [ 1-255] priority [ 0-65535]
```

设置网桥的端口优先级，可以采用下面这条命令：

```
Router (config-if) # bridge-group [ 1-255] priority [ 1-255]
```

下面这条命令则用于设置网桥路径代价：

```
Router (config-if) # bridge-group [ 1-255] path-cost [ 0-65535]
```

建立透明桥接的第 1 步是定义生成树协议 (STP) 并分配桥组号码。可以选用的生成树协议 (STP) 包括 IEEE 802.1D STP、Digital，或者是 IBM 版本的 STP。IEEE 802.1D STP 是用于运行网桥的首选 STP。Digital STP 或 IBM 版的 STP 都只是为了向后兼容而采用的。

下一步是将每个网络接口分配给一个桥组。Cisco 关于桥组的定义的原文如下：

An internal organization of network interfaces on a router. Bridge groups within the same router function as distinct bridges; that is, bridged traffic and bridge protocol data units (BPDUs) cannot be exchanged between different bridge groups on a router. Furthermore, bridge groups cannot be used to multiplex or demultiplex different streams of bridged traffic on a LAN. An interface can be a member of only one bridge group. (译文：作为路由器上网络接口的一个内部组织。同一路由器内的桥组是作为不同的网桥来工作的，也就是说，桥接数据和桥接协议数据单元 (BPDU) 不能在同一路由器上的桥组之间进行交换。此外，桥组不能用于局域网中不同网桥数据流的多路复用或多路分用。一个接口只能是一个桥组的成员。)

如果在帧中继多点网络或 DDR 网络上进行网桥的配置，则还需要附加的 **map** 命令，以便能够在网络中进行网桥数据的传输。

把接口放置到桥组中，有一定的原因：

- 将所有不可路由的数据通过网桥与组成桥组的网络接口进行桥接。
- 在同一桥组中的局域网中接收和发送 BPDU，从而参与到正常的生成树算法活动中。每个配置好的桥组都会运行一个独立的生成树进程。每个桥组只参与自己的生成树运作。

图 13-3 中，接口 e0 和 e1 位于桥组 1 中。这两个接口之间转发桥接数据。而接口 e3 不属于这个桥组，因而不会接收到来自该桥组的任何数据。

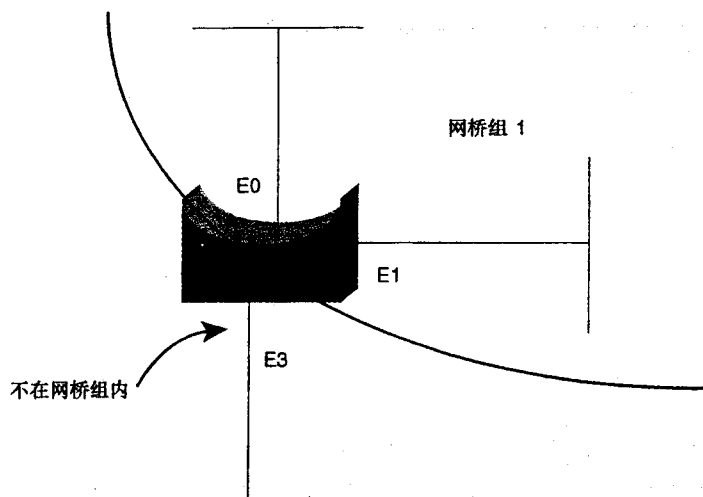


图 13-3 透明桥接组

技巧 交换网络中隔离网桥数据的一条有效途径是创建一个针对网桥数据的 VLAN。需要网桥数据的所有设备都位于这个 VLAN 中。而数据链路交换技术可以不用把数据传送到中间的所有网段中，而将 VLAN 数据（或者叫做网桥数据）越过 LAN 或 WAN 传输到目的地去。

13.1.3 透明桥接模型

图 13-4 给出了透明桥接的实例。在这个模型中，工作站是运行着 NetBEUI（不可路由协议）的 MS Windows 9x。要在这些工作站之间建立通信，必须在路由器 shuttle_5 和 shuttle_6 之间的帧中继网络和以太桥接口上启动透明桥接。

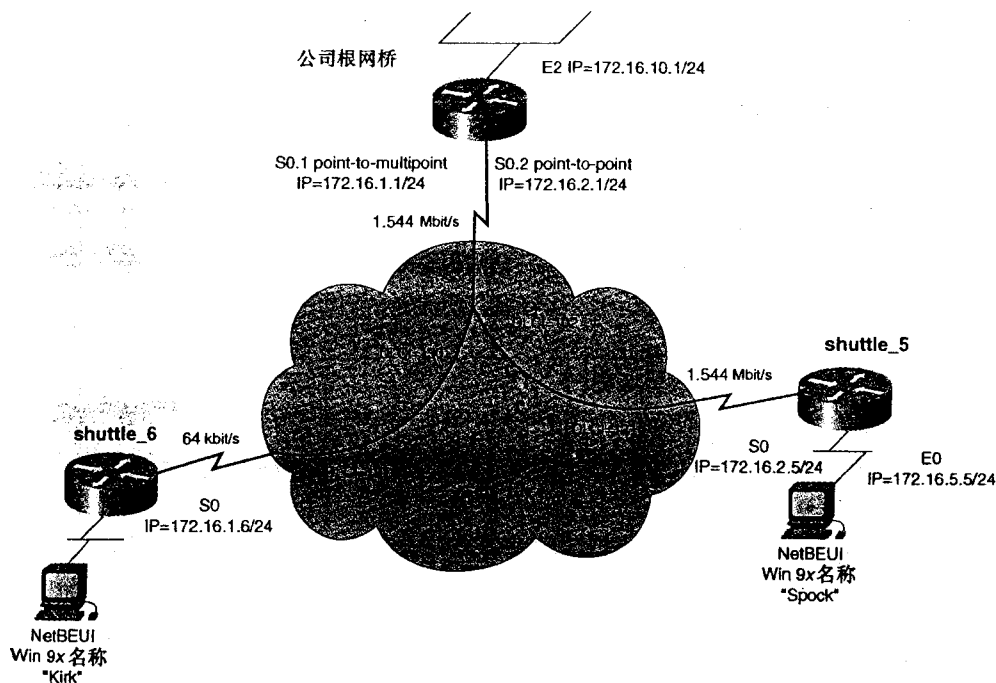


图 13-4 透明桥接

按照前面的 3 步骤过程的方法可以在路由器 enterprise 上启动透明桥接。首先，为网桥区域分配一个桥组和 STP，这可以通过全局配置命令 **bridge group 1 protocol ieee** 来完成。这个模型中采用 802.1d 作为 STP。第 2 步，通过接口命令 **bridge-group 1** 来分配桥组的接口，在路由器 enterprise 上这条命令必须在 E0 以太桥接口以及 s0.1 和 s0.2 这两个帧中继接口上输入。由于 S0.1 是帧中继多点网络接口，因此还需要用 **frame-relay map bridge** 命令来把一个特定的 DLCI 映射到该网桥。最后就是根网桥的设置。这个模型中选路由器 enterprise 作为 STP 的根。全局配置命令 **bridge-group 1 priority 100** 可以用来通过将 enterprise 路由器/网桥的优先级设置为 100 来强制进行根的选择。例 13-1 给出了路由器 enterprise 的配置示例。

例 13-1 路由器 enterprise 上的透明桥接配置示例

```

hostname enterprise
!
<<<text omitted>>>
!
interface Ethernet2
 ip address 172.16.10.1 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 1          ← Assign E2 to bridge 1
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 logging event subif-link-status
 logging event dlci-status-change
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 frame-relay map bridge 130 broadcast      ← Map statement needed for bridging
 frame-relay map ip 172.16.1.6 130 broadcast
 bridge-group 1          ← Assign S0.1 to bridge 1
!
interface Serial0.2 point-to-point
 ip address 172.16.2.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 102
 bridge-group 1          ← Assign S0.2 to bridge 1
!
<<<text omitted>>>
!
bridge 1 protocol isis          ← Define bridge 1 with 602 id as the SFP
 bridge 1 priority 100          ← Set Bridge Priority to 100, forcing ROOT
!

```

路由器 shuttle_5 和 shuttle_6 的配置和 enterprise 很相似。例 13-2 分别列出这两台路由器的桥接部分的配置。注意，帧中继映射命令只在帧中继多点网络中才需要。

例 13-2 路由器 shuttle_5 和 shuttle_6 的透明桥接配置示例

```

hostname shuttle_5
!
interface Ethernet0
 ip address 172.16.5.5 255.255.255.0
 bridge-group 1          ← Assign E0 to bridge 1
!
interface Serial0
 ip address 172.16.2.5 255.255.255.0
 encapsulation frame-relay
 frame-relay interface-dlci 121
 frame-relay lmi-type cisco

```

```

!
!
<<<text omitted>>>
!
!
bridge 1 protocol ieee          - Define bridge 1 with 802.1d as the STP
!

hostname shuttle_6
!!
interface Ethernet0
 ip address 172.16.6.6 255.255.255.0
 no ip directed-broadcast
!
bridge-group 1                  - Assign E0 to bridge 1
!
interface Serial0.1             - Remember this is a multi point!
 ip address 172.16.1.6 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 no fair-queue
!
frame-relay map bridge 131 broadcast - Map bridge 1 to DLCI 131
frame-relay map ip 172.16.1.1 131 broadcast
!
bridge-group 1                  - Assign S0 to bridge 1
!
<<<text omitted>>>
!
bridge 1 protocol ieee          - Define bridge 1 with 802.1d as the STP
!

```

13.1.4 透明桥接的检验，透明桥接和 STP 的“Big show”命令

Cisco 为网桥工作状态的检验提供了一些非常有用的命令。我个人不推荐使用 **debug** 命令对透明桥接进行调试。这类 **debug** 命令都没有公开，要不就是提供的信息没有什么意义，比如下面这一个：

```

11:23:34: ST: Serial0.1
00000000008000000605CF35DA400000000800000605CF35DA480060000140002000F00

```

建议大家采用其他更有用也更容易理解的命令，而不要去试着理解像上面这些由 **debug spantree tree** 命令产生的数据位流。如不是网桥调试时的一些有帮助的 **show** 命令：

```

show bridge [ bridge_number]
show spanning-tree [ bridge_number]

```

1. show bridge 命令

命令 **show bridge** 可以显示网桥的当前状态，网桥获取到的 MAC 地址以及网桥是否正通过某个接口在转发数据等信息，还能列出时长、发送数和接收数等。如果该命令以网桥号码为参数，就可以列出已知的网桥端口以及端口所处的 STP 状态：学习、侦听、转发还是阻塞。例 13-3 列出了在前面例子中的路由器 **shuttle_5** 上执行不同形式的 **show bridge** 命令的结果。可以参考第 2 章再回顾一下关于 STP 状态的内容。

例 13-3 路由器 **shuttle_5** 上的 **show bridge** 命令输出

```

shuttle_5#show bridge

```

```
Total of 300 station blocks, 296 free
Codes: P - permanent, S - self
```

```
Bridge Group 1:
```

Address	Action	Interface	Age	RX count	TX count
0000.8139.6c45	forward	Ethernet0	0	248	0
0000.863c.3b41	forward	Serial0	0	126	107
00e0.b055.5789	forward	Serial0	0	506	0
00a0.cc74.54a4	forward	Ethernet0	0	449	157

```
shuttle_5#show bridge group
```

```
Bridge Group 1 is running the IEEE compatible Spanning Tree protocol
```

```
Port 2 (Ethernet0) of bridge group 1 is forwarding
Port 6 (Serial0 Frame Relay) of bridge group 1 is forwarding
```

2. show spanning-tree 命令

网桥的 `show spanning-tree` 命令可提供的信息和 Catalyst 交换机 `show spanning-tree` 命令的结果基本一致，包括生成树的当前根的位置、到根的路径代价、优先级以及详细的 STP 计时器信息。可以参考第 2 章“802.1d 生成树协议 (STP)”一节回顾该命令输出结果中各个字段的具体含义。例 13-4 是上例中 enterprise 路由器上该命令的执行结果。注意，这是根网桥，其优先级是 100，这和前面在模型中配置一致。

例 13-4 路由器 enterprise 上 show spanning-tree 命令的执行示例

```
enterprise#show spanning-tree
```

```
Bridge group 1 is executing the IEEE compatible Spanning Tree protocol
```

```
Bridge identifier has 0, priority 100, address 00a0.1e58.e798
```

```
Configured hello time 2, max age 20, forward delay 15
```

```
We are the root of the spanning tree
```

```
Topology change flag not set, detected flag not set
```

```
Times: hold 1, topology change 35, notification 2
```

```
hello 2, max age 20, forward delay 15
```

```
Timers: hello 0, topology change 0, notification 0
```

```
bridge aging time 300
```

```
Port 8 (Ethernet2) of Bridge group 1 is forwarding
```

```
Port path cost 100, Port priority 128
```

```
Designated root has priority 100, address 00e0.1e58.e798
```

```
Designated bridge has priority 100, address 00e0.1e58.e798
```

```
Designated port is 8, path cost 0
```

```
Timers: message age 0, forward delay 0, hold 0
```

```
BPDU: sent 876, received 0
```

```
Port 13 (Serial0.1 Frame Relay) of Bridge group 1 is forwarding
```

```
Port path cost 647, Port priority 128
```

```
Designated root has priority 100, address 00e0.1e58.e798
```

```
Designated bridge has priority 100, address 00e0.1e58.e798
```

```
Designated port is 13, path cost 0
```

```
Timers: message age 0, forward delay 0, hold 0
```

```
BPDU: sent 632, received 2
```

```
Port 14 (Serial0.2 Frame Relay) of Bridge group 1 is forwarding
Port path cost 647, Port priority 128
Designated root has priority 100, address 00e0.1e58.e798
Designated bridge has priority 100, address 00e0.1e58.e798
Designated port is 14, path cost 0
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 347, received 0

enterprise#
```

注释 Cisco IOS 12.0 在默认情况下会关闭生成树功能，用 **no bridge-group bridge_number spanning-disabled** 命令可以启动该项功能。

3. Windows 9x 或 2000 上透明桥接的测试

启动了 Microsoft networking 功能（更确切地说是 NetBEUI 协议）的 Windows 9x 或 2000 是所有网桥和 DLSw 网络的一个很好的测试平台。两台启动了 Microsoft networking 和 NetBEUI 的 Windows 工作站可以测试任何形式的网桥环境。如果还启动了“Microsoft 文件和打印共享”功能，那么可测试通过网桥或 DLSw 网络的文件传输功能。可以运用网络浏览器或者是 Windows 下的“查找计算机”应用程序在网络中传输广播数据。关于 Microsoft 网络配置的问题，可以参考第 1 章“建立网络互联所需的关键组件”或查阅 Microsoft 的相关文档。

13.2 综合路由和桥接

综合路由和桥接 (IRB) 能够实现在多个网段中通过网桥传输数据，同时还能使桥接网络中的主机访问路由网络中的主机或路由器。从本质上说，IRB 能够实现路由网络区域到桥接网络区域的互连互访。

利用 IRB 的特性，可以在同一台路由器内部的路由接口和桥组之间对某一指定的协议进行路由。确切地说，本地数据或不可路由的数据在同一桥组内的网桥接口之间进行桥接传输的同时，路由数据则可以通过路由传输到其他路由接口或桥组去。

综合路由和桥接 (IRB) 采用了桥组虚拟接口 (BVI) 的概念，利用这些接口来对给定协议的数据包进行交换。一个 BVI 就是路由器内部的一个虚拟接口，其工作过程和普通路由接口一样。BVI 不支持桥接，但却能在路由器内部代表实际的桥组和路由接口进行数据交换。BVI 和桥组之间通过接口号码关联。所有的第 3 层信息（如 IP 地址、过滤等）都在 BVI 之间进行交换，与实际的物理接口没有直接的联系。

13.2.1 IRB 的注意点

在启动 IRB 前，应该注意以下事项：

- 路由器上的路由/网桥的默认是首先对所有的数据包进行路由，然后再通过网桥进行传输。这正是透明桥接的配置不影响路由区域的原因。但是，IRB 功能启动后，这

个情况就变成了先将所有的数据包通过网桥进行传输。如果还需启动 IP 路由，就必须用 **bridge bridge_number route ip** 命令启动桥组进行路由。

- 不可路由协议的数据包，如本地区域传输 (LAT) 或者 SNA，必须通过网桥传输。对于这类协议，不能关闭其网桥功能。
- BVI 接口上不能设置网桥属性。
- IRB 已经取代了并行路由与网接 (CRB)，CRB 技术已经不再被使用。

图 13-5 给出了一个常见的 IRB 环境。注意该图中，网桥区域中的以太网接口上没有第 3 层网络地址。这些接口由于都属于同一桥组，因此都是 BVI 的成员。BVI 号码 (这里是 10) 必须是网桥号，该号码说明两个区域如何互连。桥接和路由协议的所有网络层信息都通过该 BVI 接口传输。图中只需一个 IP 地址。给 BVI 接口配置一个 IP 地址就可以使路由器在属于桥组 10 的所有接口上对 IP 进行桥接和传输。

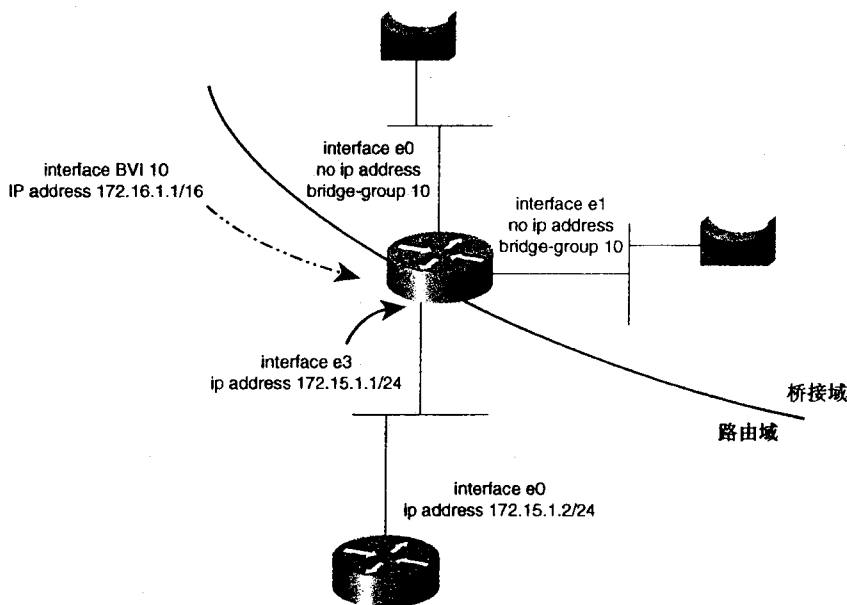


图 13-5 综合路由与网接

13.2.2 配置 IRB

IRB 的配置包括 3 个步骤:

第 1 步 按照前面讲述的方法在要进行桥接和路由传输的接口上配置透明桥接，包括创建一个桥组以及为该组分配接口。

第 2 步 配置 IRB 和 BVI。给 BVI 接口分配一个和网桥号相同的号码。例如，如果网桥号为 2，BVI 的设定就应该是 **interface bvi 2**。以下路由器全局配置下的命令可启动 IRB 工作:

```
Router (config) # bridge irb
```

第 3 步 配置桥组相关协议的路由和桥接信息。这个步骤非常重要。用 **bridge irb** 命令启动 IRB 开始工作后，该桥组中所有的接口，或启动了透明桥接的所有接口会先进行协议的桥接工作。如果链路上还有另外的协议需要路由，就会产生灾难性后果。为此，应该明确通知所有第 3 层协议是否要进行路由、桥接或者是两者兼支持。要做到这一点，需要完成以下两个步骤：

(a) 将要进行桥接和路由的协议的所有第 3 层地址分配到 BVI 接口，其第 3 层地址不能在任何物理接口上设置。

(b) 对每个协议，确定对其启动或关闭路由和桥接。

启动某个协议的路由或桥接，可以在全局配置提示符下执行这条命令：

```
Router (config) # bridge bridge_number [route | bridge] [ip | ipx | appletalk | decnet]
```

关闭某个协议的路由或桥接，可以在全局配置提示符下执行这条命令：

```
Router (config) # no bridge bridge_number [route | bridge] [ip | ipx | appletalk | decnet]
```

show interface irb 命令可以查看路由器在路由、或桥接某个给定协议，还是二者同时进行，如例 13-5 所示。

例 13-5 show interface irb 命令的执行示例

```

irb_router#show int irb

Ethernet2

Routed protocols on Ethernet2:
  ip          ipx

Bridged protocols on Ethernet2:
  appletalk  cns          decnet      vines
  apollo    xns

Software MAC address filter on Ethernet2
Hash Len  Address          Matches Act    Type
0x00:  0  ffff.ffff.ffff          0 RCV Physical broadcast
0x2A:  0  0900.2b01.0001          0 RCV DEC spanning tree
0x86:  0  00e0.1e58.e798          0 RCV Interface MAC address
0xC0:  0  0100.0ccc.cccc          0 RCV CDP
0xC2:  0  0180.c200.0000          0 RCV IEEE spanning tree
0xC2:  1  0180.c200.0000          0 RCV IBM spanning tree
    
```

13.2.3 实例：IRB 的配置

在图 13-6 的网络中对接口 e2 和 e3 的 IP 进行了桥接和路由。配置该路由器上的 IRB，首先定义一个桥组，再将需要进行桥接和路由的接口放置到桥组中。本例中定义了一个桥组，其号码为 5，然后将接口 e2 和 e3 放到这个桥组中。

如果此时检查 IRB，会发现它看上去和普通路由器一样。例 13-6 列出了路由器 irb_router 上用 **show interface irb** 命令查看 IRB 的输出结果情况。注意，路由器中的 IP 协议或者是桥接，或者是路由，但不能是二者兼有。

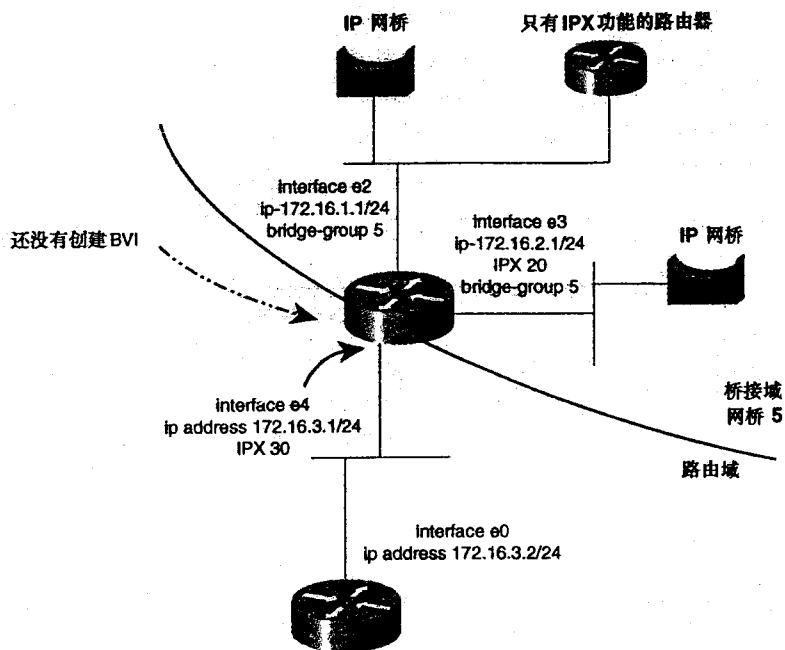


图 13-6 综合路由与网接的例子

例 13-6 show interface irb 命令的执行结果

```
irb_router#show int irb
```

Ethernet2

Routed protocols on Ethernet2:

ip ipx

Bridged protocols on Ethernet2:

appletalk cns decnet vines
apollo xns

Software MAC address filter on Ethernet2

Hash	Len	Address	Matches	Act	Type
0x00:	0	ffff.ffff.ffff	0	RCV	Physical broadcast
0x2A:	0	0900.2b01.0001	0	RCV	DEC spanning tree
0x86:	0	00e0.1e58.e798	0	RCV	Interface MAC address
0xC0:	0	0100.0ccc.cccc	0	RCV	CDP
0xC2:	0	0180.c200.0000	0	RCV	IEEE spanning tree
0xC2:	1	0180.c200.0000	0	RCV	IBM spanning tree

Ethernet3

Routed protocols on Ethernet3:

ip ipx

Bridged protocols on Ethernet3:

appletalk cns decnet vines
apollo xns


```
Software MAC address filter on Ethernet3
Hash Len  Address      Matches  Act      Type
0x00:  0  ffff.ffff.ffff      0  RCV  Physical broadcast
0x2A:  0  0900.2b01.0001      0  RCV  DEC spanning tree
0x85:  0  00e0.1e58.e79b      0  RCV  Interface MAC address
0xC0:  0  0100.0ccc.cccc      0  RCV  CDP
0xC2:  0  0180.c200.0000      0  RCV  IEEE spanning tree
0xC2:  1  0180.c200.0000      0  RCV  IBM spanning tree

Ethernet4

Routed protocols on Ethernet4
ip ipx
irb_router#
```

配置的第 2 步是启动 IRB 以及创建 BVI。由于网桥号是 5，因此 BVI 接口号也应该是 5。BVI 创建后，路由器会生成一些命令，这是路由器用来从一个透明桥接接口获取所有第 3 层协议信息进行路由所必需的命令。例 13-7 演示了这个网络模型中启动 IRB 后路由器产生的一些结果。该模型中，透明桥接接口上启动并运行了 IPX 和 IP 协议。因此，这些命令都由路由器自动生成。

例 13-7 激活 IRB

```
irb_router(config)#bridge irb
IRB: generating 'bridge 5 route ip' configuration command
IRB: generating 'bridge 5 route novell' configuration command
irb_router(config)#int bvi 5
04:17:56: %LINEPROTO-5-UPDOWN: Line protocol on Interface BVI5, changed state to
up
irb_router(config-if)#
```

现在应该从物理接口中删除路由和桥接协议的第 3 层网络地址。但是，其他不需在桥组里进行路由和桥接的第 3 层地址仍然应该保留在物理接口上。接着，把路由和桥接协议的第 3 层地址分配到 BVI 接口。图 13-7 提供了网络所需作的改变。

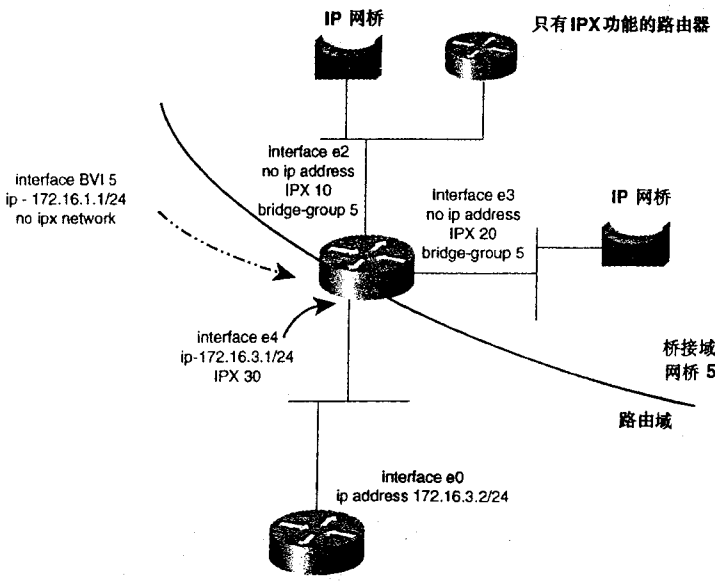


图 13-7 设置地址和创建 BVI

完成了 IP 地址的变更后，模型也就基本完成了。这时，IP 通过桥接和路由传输，可以用 `show int irb` 命令加以确认，如例 13-8 所示。

例 13-8 show interface irb 命令示例

```
irb_router#show int irb

BVI5

Routed protocols on BVI5:
ip

Ethernet2

Routed protocols on Ethernet2:
ip      ipx

Bridged protocols on Ethernet2:
appletalk cline      decnet 1  ip
vines      apollo     ipx      xns

Software MAC address filter on Ethernet2
Hash Len  Address      Matches Act    Type
0x00: 0  ffff.ffff.ffff      0 RCV Physical broadcast
0x2A: 0  0900.2b01.0001      0 RCV DEC spanning tree
0x86: 0  00e0.1e58.e798      0 RCV Interface MAC address
0x86: 1  00e0.1e58.e798      0 RCV Bridge-group Virtual Interface
0xC0: 0  0100.0ccc.cccc      0 RCV CDP
0xC2: 0  0180.c200.0000      0 RCV IEEE spanning tree
0xC2: 1  0180.c200.0000      0 RCV IBM spanning tree

Ethernet3

Routed protocols on Ethernet3:
ip      ipx

Bridged protocols on Ethernet3:
appletalk cline      decnet 1  ip
vines      apollo     ipx      xns

Software MAC address filter on Ethernet3
Hash Len  Address      Matches Act    Type
0x00: 0  ffff.ffff.ffff      0 RCV Physical broadcast
0x2A: 0  0900.2b01.0001      0 RCV DEC spanning tree
0x85: 0  00e0.1e58.e79b      0 RCV Interface MAC address
0x86: 0  00e0.1e58.e798      0 RCV Bridge-group Virtual Interface
0xC0: 0  0100.0ccc.cccc      0 RCV CDP
0xC2: 0  0180.c200.0000      0 RCV IEEE spanning tree
0xC2: 1  0180.c200.0000      0 RCV IBM spanning tree

Ethernet4

Routed protocols on Ethernet4:
ip      ipx

irb_router#
```

通过观察例 13-8 的显示结果，可以发现 BVI 看上去就和 E4 这样的普通接口一样。注意例子中只有 IP 运行于 BVI 上，该协议是惟一要在桥组中进行桥接和路由的协议。

台下游 IPX 路由器上产生了问题。为此，第 3 个步骤的第 2 部分要求用 **no bridge 5 bridge ipx** 命令关闭 IPX 的桥接功能。完成该命令后再看一下 IRB 接口，注意它的变化。IPX 不再在接口 E2 和 E3 上同时进行桥接和路由，而只是进行路由。例 13-9 列出了在该路由器上 **show irb** 命令的执行结果，显示出 IPX 不再同时进行桥接和路由。

例 13-9 show interface irb 命令示例

```
irb_router#show int irb
BVI5

Routed protocols on BVI5:
ip

Ethernet2

Routed protocols on Ethernet2:
ip      ipx

Bridged protocols on Ethernet2:
  appletalk  cns      decnet      ip
  vines      apollo    xns

Software MAC address filter on Ethernet2
Hash Len  Address      Matches  Act      Type
0x00:  0  ffff.ffff.ffff      0  RCV  Physical broadcast
0x2A:  0  0900.2b01.0001      0  RCV  DEC spanning tree
0x86:  0  00e0.1e58.e798      0  RCV  Interface MAC address
0x86:  1  00e0.1e58.e798      0  RCV  Bridge-group Virtual Interface
0xC0:  0  0100.0ccc.cccc      0  RCV  CDP
0xC2:  0  0180.c200.0000      0  RCV  IEEE spanning tree
0xC2:  1  0180.c200.0000      0  RCV  IBM spanning tree

Ethernet3

Routed protocols on Ethernet3:
ip      ipx

Bridged protocols on Ethernet3:
  appletalk  cns      decnet      ip
  vines      apollo    xns

Software MAC address filter on Ethernet3
Hash Len  Address      Matches  Act      Type
0x00:  0  ffff.ffff.ffff      0  RCV  Physical broadcast
0x2A:  0  0900.2b01.0001      0  RCV  DEC spanning tree
0x85:  0  00e0.1e58.e79b      0  RCV  Interface MAC address
0x86:  0  00e0.1e58.e798      0  RCV  Bridge-group Virtual Interface
0xC0:  0  0100.0ccc.cccc      0  RCV  CDP
0xC2:  0  0180.c200.0000      0  RCV  IEEE spanning tree
0xC2:  1  0180.c200.0000      0  RCV  IBM spanning tree

Ethernet4

Routed protocols on Ethernet4:
ip      ipx
irb_router#
```

例 13-10 配置 IRB 的网络模型

```
hostname irb_router
!
ip subnet-zero
ipx routing 00e0.1e58.e792
!
bridge irb
!
<<<text omitted>>>
!
interface Ethernet2
 no ip address
 no ip directed-broadcast
 media-type 10BaseT
 ipx network 10
 bridge-group 5
!
interface Ethernet3
 no ip address
 no ip directed-broadcast
 media-type 10BaseT
 ipx network 20
 bridge-group 5
!
interface Ethernet4
 ip address 172.16.3.1 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
 ipx network 30
!
<<<text omitted>>>
!
interface BVI5
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
!
router eigrp 2001
 network 172.16.0.0
!
ip classless
!
bridge 5 protocol ieee
 bridge 5 route ip
 bridge 5 route ipx
 no bridge 5 bridge ipx
!
```

13.3 源路由桥接（SRB）

在 IEEE 802.1 委员会考虑将透明桥接加以改编作为局域网互连标准的同时，也把源路由桥接作为另一选择。历史告诉我们，IEEE 802.1 委员会最后选择了透明桥接。与此同时，IEEE 802.5 采纳了源路由桥接，将其作为连接 IBM 的令牌环局域网和 IEEE 802.5 局域网的协议。

图 13-8 为 802.5 令牌环的数据帧结构。

令牌环 802.5	SD	AC	FC	目标地址	源地址	路由信息域	信息域	FCS	ED	FS
--------------	----	----	----	------	-----	-------	-----	-----	----	----

图 13-8 IEEE 802.5 令牌环数据帧格式

13.3.1 源路由桥接 (SRB) 概览

源路由桥接通过结合探测数据包和 RIF 字段来确定通过桥接网络的最佳路径。源路由桥接采用了 IEEE 802.5 数据包中 MAC 报头里的路由信息字段 (RIF) (如图 13-9 所示)，决定数据包必须传输到哪一个令牌环或者令牌环的网段。这就是源路由名称中“路由”的来历。

目标地址	RII	源地址	RIF	DATA	FCS
------	-----	-----	-----	------	-----

图 13-9 IEEE 802.5 MAC 数据帧格式

在源工作站发送的每一个数据帧中，MAC 报头里都有一个 RIF 字段紧跟在源地址字段之后。目的工作站则将路由字段反转过来以访问源工作站。透明生成树网桥需要 50 秒的时间从链路故障中恢复，与此不同的是，源路由桥接中有多条活动路径通过网络，这样，出现问题时就能以很快的速度切换到另一条路径。很重要的一点是，源路由桥接允许终端工作站自行决定数据帧通过网络的最佳路由，这样就把传输数据帧的负担放到了终端工作站。IBM 令牌环在 RIF 中规定最大令牌环数为 8，最大网桥数为 7。而 802.5 规定的最大令牌环数为 14，最大网桥数 13。图 13-9 给出了一个 802.5 MAC 数据帧的结构图。

令牌环网络中的环是在路由信息标识符 (RII) 中用一个惟一的 12 位令牌环号 (1-4095) 来指定。而两个令牌环之间的网桥则是用 RIF 中一个惟一的 4 位网桥号来指定的。网桥号的有效范围是 1 到 15，连接同样两个令牌环的网桥中保持其惟一性即可。如果 RII 设置为 0，数据帧中就没有 RIF，如果 RII 设置为 1，数据帧中就有 RIF。

1. RIF 字段

RIF 由 16 位的路由控制字段和路由描述字段组成。图 13-10 给出了 RIF 的格式图。

RC	RD	RD	...
----	----	----	-----

RC = 路由控制域
RD = 路由描述符
每个字段为 16 个比特长

图 13-10 基本 RIF 格式

图 13-11 是 RIF 路由控制字段的格式以及每个字段的含义。

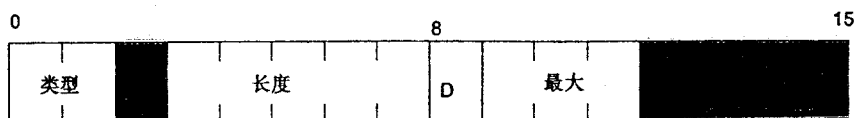


图 13-11 RIF 的路由控制字段格式

- 图中阴影部分是保留字段。
- **type**——所用的探测帧类型：
 - 00: 特定路由探测帧——
 - 10: 所有环，全路由探测帧
 - 11: 所有环，跨越路由（受限广播）类型
- **length**——RIF 的字节总长度。
- **D**——方向，含义如下：
 - 0: 从左到右读取路由（正向）
 - 1: 从右到左读取路由（反向）
- **largest**——路由器通过该路由能够处理的最大数据帧长度：
 - 000: 516 bytes (DDN 1822)
 - 001: 1500 bytes (以太网)
 - 010: 2052 bytes
 - 011: 4472 bytes (令牌环和 Cisco 的最大长度)
 - 100: 8144 bytes (令牌环总线)
 - 101: 11,407 bytes
 - 110: 17,800 bytes
 - 111: 65,535 (初始值)

图 13-12 是 RIF 的路由描述字段的格式。在配置静态 RIF 或表示 RIF 时，该字段用点分的 16 进制数来表示。

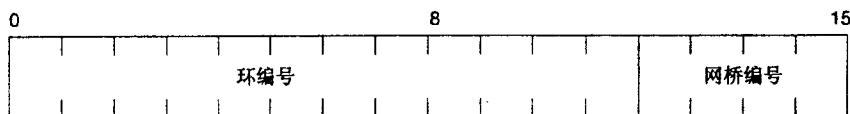


图 13-12 路由描述字段格式

- **Ring number**——桥接网络中惟一的十进制令牌环号。
- **Bridge number**——连接相同两个令牌环的任意网桥之间的惟一的十进制网桥号。
网桥号为 0 表示 RIF 的终止。

图 13-13 是一个 SRB 网络的例子。工作站 Alpha 到 Bravo 的 RIF 读成是：
0830.0012.002a.00b0

其中，0830 是 16 位的 RC 字段，0012，002a 和 00b0 则是 3 个 16 位的 RD 字段。从左向右的前 4 个位说明浏览器类型为 0，即特定路由探测帧。8 是指整个 RIF 长度为 8。D 位设置成 0，说明 RIF 是从左往右读，即正向。接下来的 3 个位是 011，把数据帧的最大传输单元设置成了 4472，即 Cisco 的最大帧尺寸。RD 字段的分解非常简单：

RING1-BRIDGE2 = 0012

RING2-BRIDGE10 = 002a

RING11-BRIDGE0 = 00b0

号码为 0 的网桥终止 RIF，后面不会再有桥连接在令牌环上。

“静态 RIF 的配置”一节中还将详细讲述静态 RIF 的配置问题。

RIF = 0830.0012.002a.00b0

or

RING1-BRIDGE2-RING2-
BRIDGE10-RING11

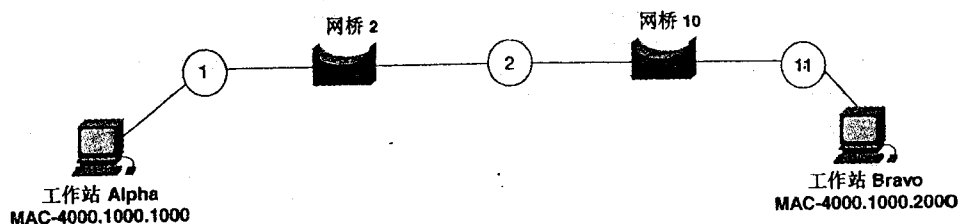


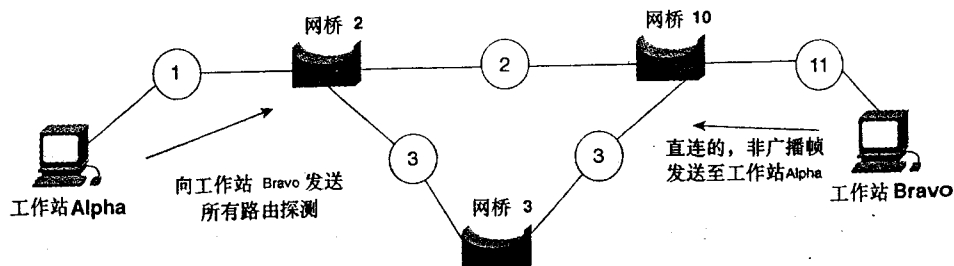
图 13-13 源路由桥接——RIF

源路由桥接可以通过查看 RIF 字段确定是否要将探测数据包转发到令牌环。如果 RIF 中有重复的环——网桥——环模式存在，探测数据包不会转发到该令牌环。

RIF 中的信息从源节点产生的探测数据帧中获取。这些探测数据包能够穿越整个源路由桥接网络，在源节点所有可能用来转发数据的路径上收集可用信息。SRB 采用了 3 种类型的探测数据帧：

- 全路由探测或全令牌环探测帧——这种类型的探测数据包从一个接一个 SRB 的令牌环到传播到相应的目的地址。目的工作站接收到全路由探测数据包后转发正向的非广播数据帧返回到产生探测数据包的源工作站。
- 特定路由探测或本地探测帧——这种类型的探测数据包是终端工作站用来确定本地令牌环中某个特定工作站的位置的。NetBIOS 和 SNA 会产生这类探测数据包。
- 生成树探测或受限路由探测帧——生成树探测数据包只有在令牌环接口上启动了源网桥生成树功能的情况下才能进行传输。NetBIOS 这样的协议需要这种类型的探测数据帧。终端工作站接收到一个生成树探测数据包时，会向产生这个数据包的源工作站发送全路由探测数据包作为响应。

图 13-14 和 13-15 为全路由和生成树类型的探测数据包如何工作的例子。



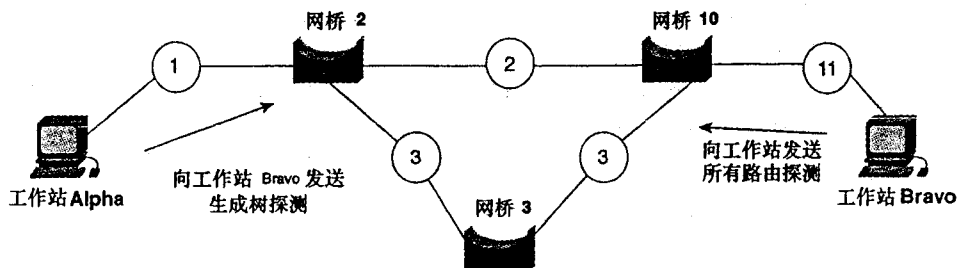


图 13-15 SRB 的生成树探测数据包

13.3.2 源路由桥接 (SRB) 的配置

源路由桥接可以通过 3 种主要方式进行配置：

- 基本的本地 SRB。
- 多端口的本地 SRB。
- 远程 SRB (RSRB)。

1. 基本的本地源路由桥接 (SRB) 的配置

本地 SRB 以其最简单的形式存在于路由器的两个令牌环之间。图 13-16 给出了该类 SRB 的配置情况。

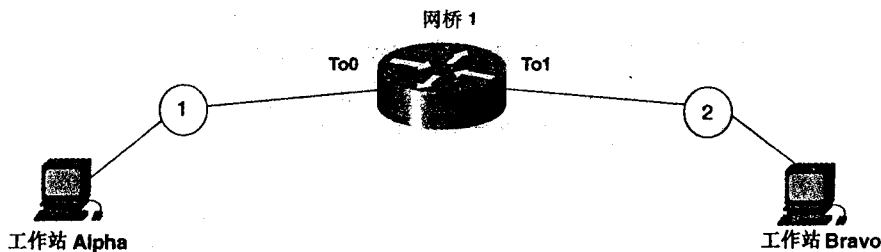


图 13-16 基本的本地 SRB 配置

这种类型的 SRB 的配置可按照下面的两个步骤进行：

第 1 步 如果需要，用接口命令 **multiring all** 启动 RIF，该命令的句法如下：

```
Router (config-if) #multiring { protocol-keyword | all | other}
no multiring { protocol-keyword | all | other}
```

Cisco IOS 还允许通过参数 *protocol-keyword* 指定协议。利用该关键字使得在接口上按协议将 RIF 附加到被路由协议中去。采用了针对某个协议的该关键字之后，路由器会发出包含源路由桥接所用信息的数据包。所支持的协议以及关键字有：

- **apollo**——Apollo 域
- **appletalk**——AppleTalk Phases 1 和 2
- **clns**——ISO CLNS
- **decnet**——DECnet Phases IV

- novell——Novell IPX
- vines——Banyan VINES
- xns——XNS

还有另外两个关键字可以和 **multiring** 命令一起使用。关键字 **all** 启动所有数据帧的 RIF，推荐使用这种方法。而关键字 **other** 则启动了除上面列出的能够支持的协议之外的所有路由数据帧的 RIF。和适当关键字一起使用的 **no multiring** 命令则可以禁止指定协议使用 RIF 信息。

第 2 步 令牌环接口的 SRB 配置，通过下面这条命令来完成：

Router (config-if) #source-bridge local_ring bridge_number destination_ring
SRB 的配置情况请参见例 13-11。

例 13-11 本地 SRB 的配置示例

```
interface TokenRing0
no ip address
no ip directed-broadcast
ring-speed 16
multiring all          ←RIF enabled
source-bridge 1 1 2    ←From ring 1 thru bridge 1 to ring 2
!
interface TokenRing1
no ip address
no ip directed-broadcast
ring-speed 16
multiring all          ←RIF enabled
source-bridge 2 1 1    ←From ring 2 thru bridge 1 to ring 1
!
```

2. 多端口本地源路由桥接的配置

如若有两个以上的令牌环接口需要进行桥接，就要采用另一种类型的 SRB。这类 SRB 的配置需要在路由器上定义一个虚拟令牌环。顾名思义，虚拟令牌环就是连接两个或两个以上的本地或远程物理令牌环的一个虚拟实体。虚拟令牌环也称为令牌环组。下一节“远程源路由网桥的配置”会看到，虚拟令牌环可以对整个 IP 区域进行扩展。在这种类型的 SRB 配置中，虚拟令牌环是限制在本地路由器上的。图 13-17 中含有一个 3 端口 SRB 的例子。要配置令牌环 1、2 和 10 之间的 SRB，首先需定义一个虚拟令牌环。然后通过源网桥将每个真实令牌环与虚拟令牌环互连。图 13-18 概念性地给出加入了虚拟令牌环后网络的情况。

这类 SRB 的配置包含下列的 4 个步骤：

第 1 步 在路由器中定义一个虚拟令牌环，所需要的全局配置命令如下：

Router (config) #source-bridge ring-group virtual_ring_number
虚拟令牌环号 virtual_ring_number 的范围为 1 到 4095。

第 2 步 如果需要，用路由器接口命令 **multiring all** 启动 RIF，该命令的句法结构为：

Router (config-if) #multiring { protocol-keyword | all | other}
no multiring { protocol-keyword | all | other}

第 3 步 配置令牌环接口的 SRB，使用以下接口命令：

Router (config-if) #source-bridge local_ring bridge_number virtual_ring

第 4 步 (可选) 启动生成树探测数据包。这样可以减少通过网络传输的探测数据包的数量。NetBIOS 和 NetBEUI 都需要生成树探测数据包在网络传输才能正常工作。

Cisco 建议在复杂的多协议网络环境下最好启动生成树探测数据包的传输。下面这条接口命令可完成这一工作:

Router (config-if) #source-bridge spanning

例 13-12 给出了图 13-17 的网络上多端口 SRB 配置的情况。

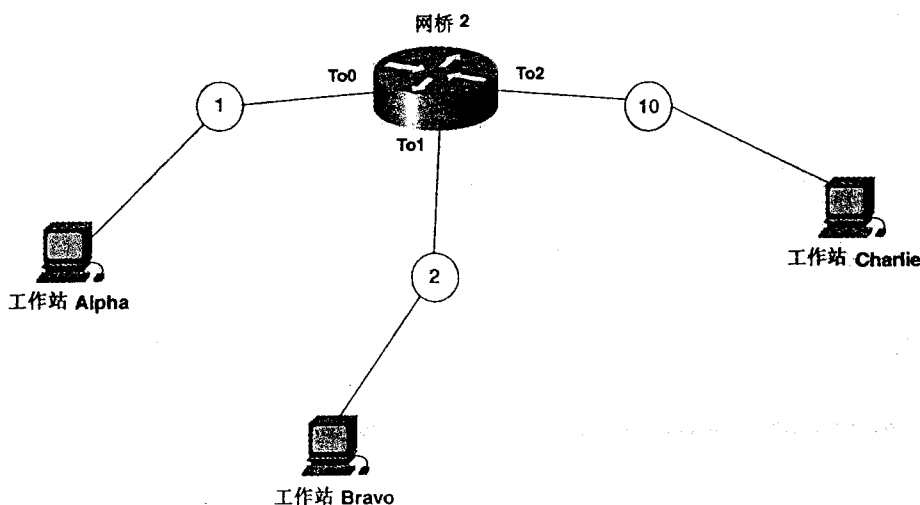


图 13-17 SRB 的多端口网接

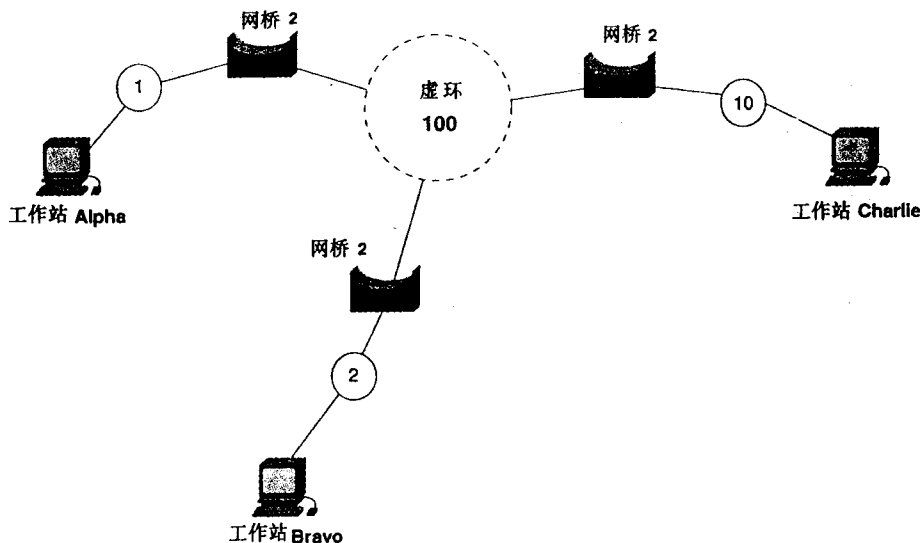


图 13-18 SRB 多端口网接的概念性视图

例 13-12 多端口 SRB 的配置示例

```
1
source-bridge ring-group 100 ←Configure a virtual ring of 100
1
interface TokenRing0
```

```
no ip address
no ip directed-broadcast
ring-speed 16
multiring all          RIF enabled
source-bridge 1 2 100  From ring 1 thru bridge 2 to V-ring 100
!
interface TokenRing1
no ip address
no ip directed-broadcast
ring-speed 16
multiring all          RIF enabled
source-bridge 2 2 100  From ring 2 thru bridge 2 to V-ring 100
!
interface TokenRing2
no ip address
no ip directed-broadcast
ring-speed 16
multiring all          RIF enabled
source-bridge 10 2 100 From ring 10 thru bridge 2 to V-ring 100
```

3. 配置远程源路由桥接 (RSRB)

SRB 还可以配置对某个 WAN 串行接口或整个 IP 区域进行扩展，这就是 **远程源路由桥接 (RSRB)**，即定义了一个虚拟令牌环来连接远程网桥。例 13-19 为与另一个常用的帧中继网络互连的令牌环 SRB。

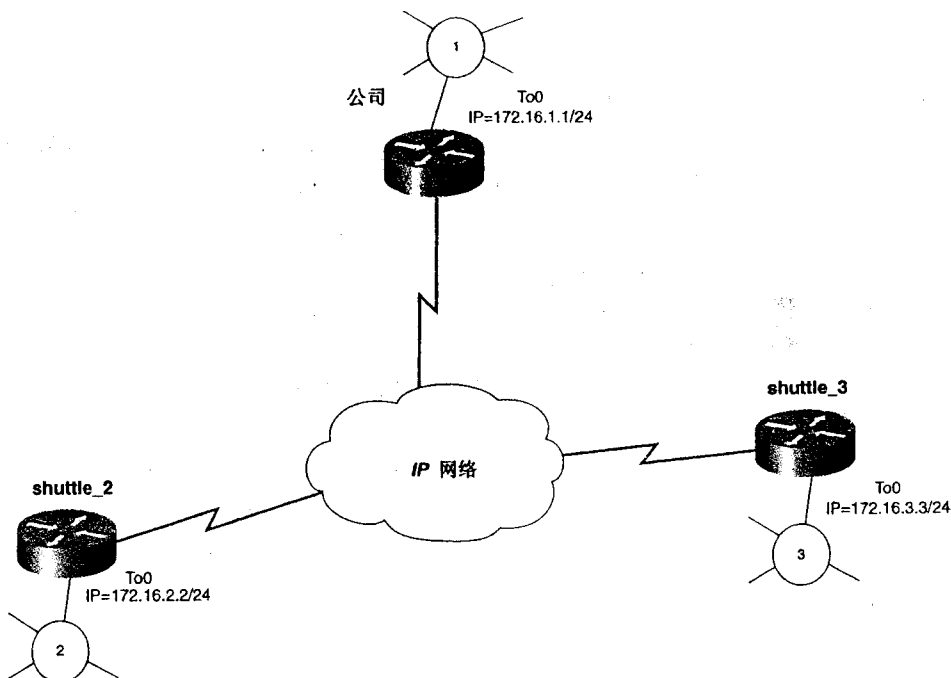


图 13-19 与一个帧中继网络互连的 RSRB 网络

RSRB 的配置需要定义一个用于连接所有 SRB 的公用虚拟令牌环。本例中，虚拟令牌环最合理的位置是在 IP 网络或者 WAN 中。图 13-20 显示了定义虚拟令牌环后的 RSRB 网络情况。

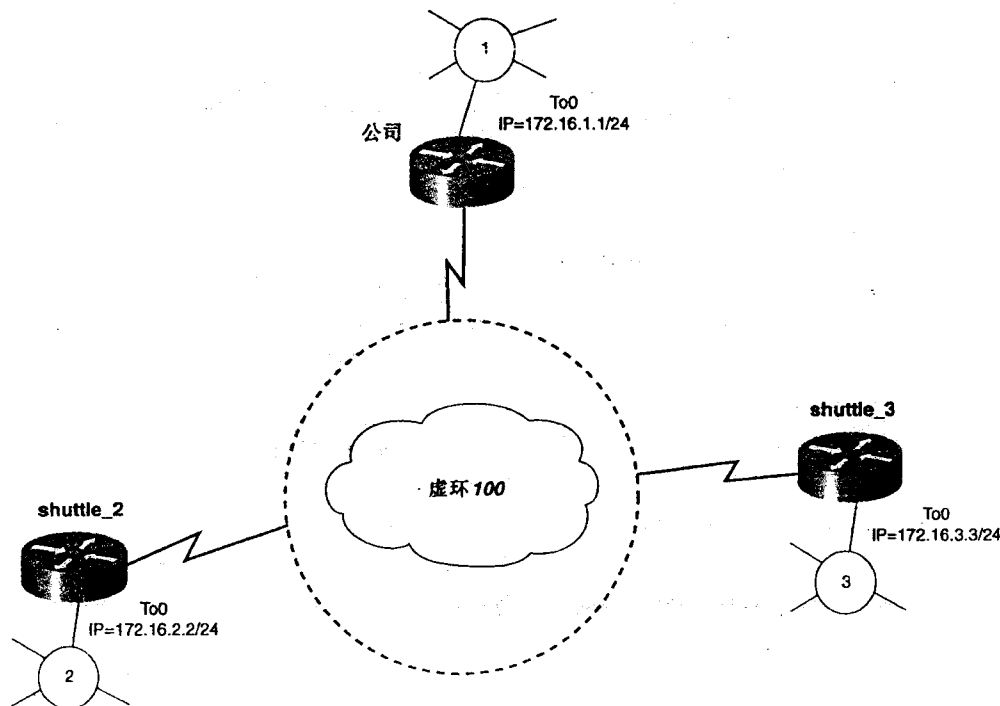


图 13-20 定义了虚拟令牌环的 RSRB 网络

RSRB 提供了多种在 IP 网络中封装 SRB 信息的方式,这主要根据 WAN 接口而定。RSRB 能够用 4 种方式对 RSB 信息进行封装:直接方式、FST、TCP 和帧中继方式。这几种方式很相似,只是可能会需要增加额外的配置。表 13-1 是对这 4 种封装类型的归纳,下面进行详细的讨论。

表 13-1

RSRB 封装的类型与需求

封装类型	需求的链路类型	语 法
直接	LAN——只用于单跳 WAN——必须使用 HDLC	<code>source-bridge remote-peer V_ring interface interface_name</code>
帧中继	WAN——只用于帧中继	<code>source-bridge remote-peer V_ring framerelay interface interface_name</code> <code>frame-relay map rsrb dci_number</code>
TCP	WAN 或 LAN	<code>source-bridge remote-peer V_ring tcpip_address</code>
FST	WAN 或 LAN	<code>FST WAN or LAN source-bridge fst-peername ip_address</code> <code>source-bridge remote-peer V_ring fst ip_address</code>

* V_ring = the virtual ring

- 直接封装——直接封装方式可在两台路由器之间的单个物理网络上快速进行 SRB 数据帧的封装。这种方式不提供 RSRB 的一些高级特性,如本地确认等,但其效率很高。在 WAN 接口上使用直接封装方式时,该接口必须以 HDLC 作为其数据链路的协议。记住,直接封装方式只能用于相互邻接的两台路由器。

- **帧中继封装**——帧中继封装方式使用 RFC 1490 直接把 RSRB 封装在帧中继数据帧中。这种方式能够以 PVC 为基础对 RSRB 进行控制。除了 RSRB 命令之外，还需要一条 **frame-relay map rsr** 命令用来把 RSRB 映射到多点接口上的一个 DLCI 号。帧中继封装方式需要的系统开销少于 TCP 方式，但没有 TCP 方式所有的一些高级特性。
- **TCP 封装**——TCP 封装方式能够以更少的工作提供最多的有益特性。Cisco 推荐在通过异类网连接令牌环网桥时采用该封装方式。TCP 方式还提供负载平衡和本地确认功能。
- **FST 封装**——FST 即快速序列传输。尽管这种方式消耗的系统开销低于 TCP，但速度却没有直接封装方式快。FST 封装方式可以用来连接一个以上的 SRB，但是不提供本地确认功能。还可为 FST RSRB 分配一个 FST 对等名。

配置 RSRB 可以按照下面这 4 个步骤来进行：

第 1 步 如果需要，使用路由器接口命令 **multiring all** 启动 RIF。

第 2 步 使用 **source-bridge ring-group virtual_ring** 命令启动虚拟令牌环。

第 3 步 从物理令牌环到虚拟令牌环，对 SRB 进行配置。

第 4 步 确定要使用的封装方式，对 RSRB 进行配置。

—— **直接封装方式**：为每台对等路由器以及本地路由器创建一个相应的远程对等体，所用的全局配置命令是：

```
Router (config) #source-bridge remote-peer virtual_ring interface interface_name
```

—— **帧中继封装方式**：用下面这条全局配置命令为每个对等路由器和本地路由器创建一个相应的远程对等体：

```
Router (config) #source-bridge remote-peer virtual_ring frame-relay interface  
interface_name dlci_number [ if_largest_frame_size]
```

如果使用多点接口，还必须加上一条 **frame-relay map** 命令，其命令句法为：

```
Router (config-if) #frame-relay map rsr dlci_number
```

—— **TCP 封装方式**：用下面这条全局配置命令为每个对等路由器和本地路由器创建一个相应的远程对等体：

```
Router (config) #source-bridge remote-peer virtual_ring tcp ip_address  
[ if_largest_frame_size] [ local_ack]
```

这里的 **ip_address** 指要访问的远程路由器的 IP 地址，前提是要保证与该 IP 地址的连通性。

—— **FST 封装方式**：首先创建一个源网桥 FST 对等名，它应该是一个环路地址或本地令牌环接口。通过下面这条命令来完成：

```
Router (config) #source-bridge fst-peername local_ip_address
```

然后，用下面这条全局配置命令为每个对等路由器和本地路由器创建一个相应的远程对等体：

```
Router (config) #source-bridge remote-peer virtual_ring fst ip_address  
[ if_largest_frame_size]
```

这里的 **ip_address** 指要访问的远程路由器的 IP 地址，必须保证该地址与本地路由器的连通性。

注释 建议将环路地址作为 RSRB 对等体和 DLSw 对等体。通过将等体指向一个环路接口，就为该对等体提供了一个只有本地路由器停止工作或 IP 路由出现故障时才会停止工作的接口。通常情况下，物理接口的关闭将会导致对等体工作的停止，而这又会对需要 RSRB 或 DLSw 数据的远程路由器的接口造成影响。通过将等体指向一个环路地址能够使对等体一直保持在工作状态，从而实现路由器上所有端口之间的相互独立。

4. 源路由桥接状态的确认

可以用下面的命令查看 SRB 状态：

```
show source-bridge
show source-bridge interface
```

命令 **show source-bridge** 可以显示路由器上所有 SRB 的相关信息。可通过这条命令来检验对 SRB 所作的配置。要确保令牌环和网桥号与所设置的相同。另外，该命令还能显示网络中接收到的探测数据包的数量和类型，SRB 的错误与丢失情况。可以参考 Cisco Press 出版的《Cisco IOS Bridging and IBM Networks Solutions》，书中有这些命令以及所有与源路由桥接有关的命令的详细讲解和例证。例 13-13 是 **show source-bridge** 命令的执行示例。

例 13-13 SRB 的状态

```
srb_router#show source-bridge

Local Interfaces:

```

	srn	bn	trn	r	p	s	n	max hops	receive cnt	transmit cnt	drops
To0	1	1	100	*	b			7 7 7	7297	2	154
To1	2	1	1		b			7 7 7	2	390	0

```

Global RSRB Parameters:
  TCP Queue Length maximum: 100

Ring Group 100 Virtual Ring Number
No TCP peername set, TCP transport disabled
Maximum output TCP queue length, per peer: 100
Rings:
  bn: 1 rn: 1 local ma: 4007.781a.e789 TokenRing0 fwd: 0

Explorers: ..... input .....

```

	spanning	all-rings	total	spanning	all-rings	total
To0	0	6856	6856	0	1	1
To1	0	1	1	0	390	390

```

Explorer fastswitching enabled
Local switched: 6300 flushed 0 max Bps 38400

rings inputs bursts throttles output drops
To0 6300 0 0 0 0
To1 0 0 0 0 0

srb_router#
```

命令 **show source-bridge interface** 能够快速浏览网络上 SRB 的情况，包括 SRB 状态、令牌环、网桥号以及输入输出的数据包，还能迅速判定网桥是否在正常工作以及数据传输。例 13-14 为该命令的执行示例。

例 13-14 SRB 接口的状态

```
srb srb_router#show source-bridge interfaces
```

Interface	St	MAC-Address	srn	bn	v p s n r						IP-Address	Packets	
					trn	r	x	p	b	c		In	Out
To0	up	0007.781a.e789	1	1	100	*		b		F		35373	7393
To1	up	0007.781a.e729	2	1	1			b		F		30158	7228

```
srb_router#
```

13.3.3 实例：远程源路由桥接的配置

图 13-21 是 3 台令牌环路由器通过一个帧中继网络互连的模型。可以按照前面讲的 4 步骤的过程来配置网络的 RSRB。第 1 步是用接口命令 **multiring all** 启动 RIF 字段的工作。这个模型中，所有的路由器上都需要执行这条命令。然后就是定义一个虚拟令牌环。这个模型中虚拟令牌环的合理位置是在帧中继网络中。用全局配置命令 **source-bring ring-group 100** 设置一个虚拟令牌环 100。

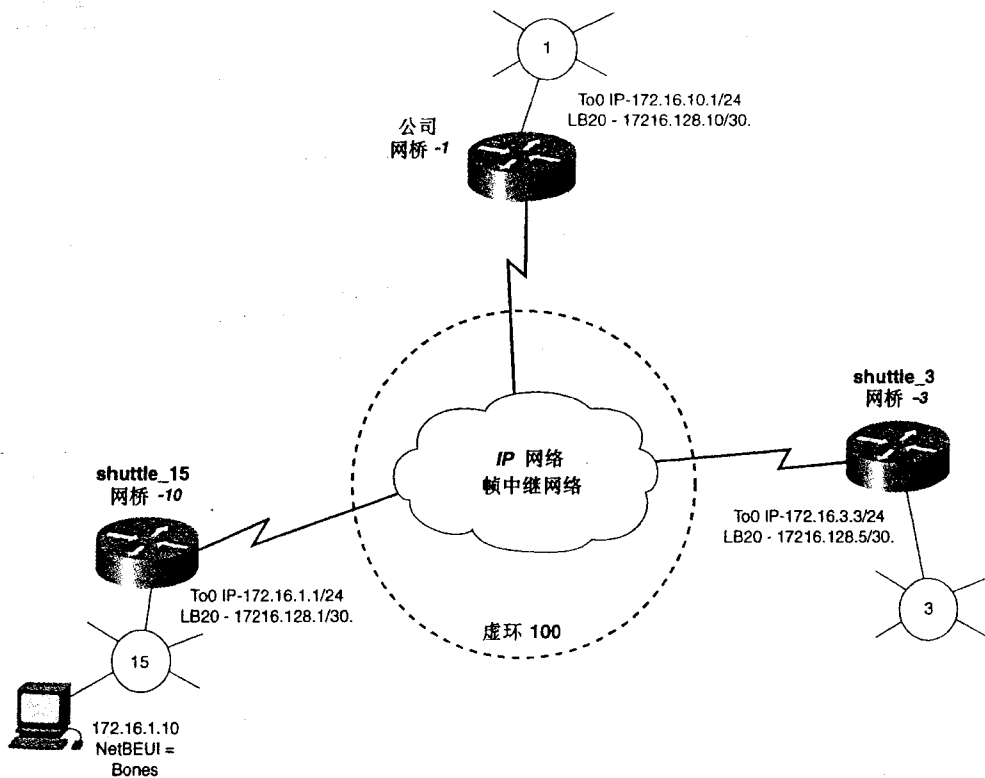


图 13-21 TCP 的 RSRB

第 3 步是在路由器上对 RSB 进行配置，这些路由器用来连接 RSRB。例 13-15 是路由器 shuttle_15 和 shuttle_3 上的 SRB 的配置情况。

例 13-15 路由器 shuttle_15 和 shuttle_3 上的 SRB 配置示例

```

hostname shuttle_15
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 ring-speed 16
 multi-ring all
 source-bridge 15 10.100
 source-bridge spanning
!

```

```

hostname shuttle_3
!
interface TokenRing0
 ip address 172.16.3.3 255.255.255.0
 ring-speed 16
 multi-ring all
 source-bridge 3 3.100
 source-bridge spanning
!

```

注释

利用 Windows 9x NetBEUI 对 RSRB 进行测试

在这个模型里可以用 Windows 9x 网络平台（更确切地说是 NetBEUI）对网桥进行测试。在没有实际数据的情况下，RSRB 会保持在关闭状态。RSRB 检测到数据之后，网桥会从关闭状态进入开启状态。在实验室环境下，可以通过将 Windows 9x 的 NetBEUI 配置为开启“文件和打印机共享”就可以产生 RSRB 所需的数据。在 Windows 9x 环境中，浏览“网上邻居”时（或者依次选择“开始”、“查找”和“搜索计算机”时）实际上就是强制使数据通过 RSRB。SRB 一检测到数据其状态就会从关闭（closed）转为工作（up）状态。这是一条简单而有效的测试 RSRB 的途径。同样的方法也可以用来对 DLSw 进行测试。在两台配置好的工作站上进行打印和文件的共享实际上就是通过透明、源路由，或 DLSw 网络完成的。Windows 2000、Me 和 NT 也可以用来进行这样的测试。

用 Windows 9x 网络平台测试该模型中的 RSRB，还需要在源路由桥接上配置生成树探测，如例 13-15 所示。

4 步骤配置过程的最后一步是选定 RSRB 上要采用的封装方式并对其进行配置。该模型采用 RSRB 和 TCP 封装方式。RSRB 组中的每台路由器都需要用 **source-bridge remote-peer** 命令进行配置。每个对等体以及本地路由器都需要用这条命令进行配置。这里利用了环路接口来连接 RSRB。例 13-16 是路由器 enterprise 的完整配置过程。例 13-17 则是路由器 shuttle_15 配置中的 RSRB 部分。

例 13-16 路由器 enterprise 上的 RSRB 配置示例

```

hostname enterprise
!
source-bridge ring-group 100
source-bridge remote-peer 100 tcp 172.16.128.10 ←Peer for the local router
source-bridge remote-peer 100 tcp 172.16.128.6 ←Peer for the shuttle_5 router
source-bridge remote-peer 100 tcp 172.16.128.1 ←Peer for the shuttle_15 router
!
!
interface Loopback20
 ip address 172.16.128.10 255.255.255.252
 no ip directed-broadcast
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 logging event subif-link-status
 logging event dlci-status-change
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 172.16.2.5 255.255.255.252
 no ip directed-broadcast
 frame-relay map ip 172.16.2.6 170 broadcast
!
interface Serial0.2 point-to-point
 ip address 172.16.2.1 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 180
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.10.1 255.255.255.0
 no ip directed-broadcast
 ring-speed 16
 multiring all
 source-bridge 1 1 100
 source-bridge spanning
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary

```

例 13-17 路由器 shuttle_15 上的 RSRB 配置示例

```

hostname shuttle_15
!
source-bridge ring-group 100
source-bridge remote-peer 100 tcp 172.16.128.1 ←Peer for the local router
source-bridge remote-peer 100 tcp 172.16.128.5 ←Peer for the shuttle_5 router
source-bridge remote-peer 100 tcp 172.16.128.10 ←Peer for the enterprise router
!
interface Loopback20
 ip address 172.16.128.1 255.255.255.252
 no ip directed-broadcast
!
interface Serial0

```

```

ip address 172.16.2.6 255.255.255.252
no ip directed-broadcast
encapsulation frame-relay
no ip route-cache
no ip mroute-cache
logging event subif-link-status
logging event dlci-status-change
frame-relay map ip 172.16.2.5 171 broadcast
!
<<<text omitted>>>
!
interface TokenRing0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
ring-speed 16
multiring all
source-bridge 15 10 100
source-bridge spanning
!
router eigrp 2001
network 172.16.0.0
no auto-summary

```

RSRB 的验证和普通源路由桥接 (RSB) 一样。使用 **show source-bridge** 命令，确认 TCP 对等体是处在关闭还是开启状态。没有某种类型的数据通过网桥，它不会从关闭状态转为开启状态。这里，工作站 Bones 浏览“网络邻居”，启动了 RSRB 工作。例 13-18 显示了 **show source-bridge** 命令在路由器 enterprise 上的执行示例。

例 13-18 show source-bridge 命令的执行示例

```

enterprise#show source-bridge

Local Interfaces:

```

	srn	bn	trn	r	p	s	n	max hops	receive cnt	transmit cnt	drops
To0	1	1	100	*	f			7 7 7	1019	0	0

```

Global RSRB Parameters:
TCP Queue Length maximum: 100

Ring Group 100:
This TCP peer: 172.16.128.10
Maximum output TCP queue length, per peer: 100
Peers:

```

	state	bg lv	pkts_rx	pkts_tx	expl_gn	drops	TCP
TCP 172.16.128.10	-	3	0	0	0	0	0
TCP 172.16.128.5	open	3	0	1258	1019	0	0
TCP 172.16.128.1	open	3	0	708	1019	346	0

```

Rings:
Rings:
bn: 3 rn: 3 remote ma: 4000.30b1.270a TCP 172.16.128.5 fwd: 0
bn: 10 rn: 15 remote ma: 4000.309a.68bb TCP 172.16.128.1 fwd: 0

Explorers: ----- input -----
spanning all-rings total
To0 284 735 1019
----- output -----
spanning all-rings total
To0 0 0 0

Explorer fastswitching enabled

```

Local switched: 1019	flushed 0	max Bps 38400		
rings	inputs	bursts	throttles	output drops
To0	1019	0	0	0
enterprise#				

现在，把上例中的 RSRB 封装类型改为帧中继方式。图 13-22 中突出显示了相关的部分，列出了正在使用中的 DLCI。

要完成这一改动，按照步骤 1 到 3 的过程，和上一节的内容完全一样。帧中继封装方式只需要在连接 RSRB 的每一台远程路由器上配置 **source-bridge remote-peer** 即可。这种类型的 RSRB 不需要为本地路由器配置远程对等体的命令，但需要在其多点子接口上为 RSRB 加上命令 **Frame-relay map**。例 13-19 列出了路由器 enterprise 和 shuttle_15 的配置情况，突出显示了帧中继 RSRB 的部分。这是该模型和上面的 TCP 模型之间惟一的差别。

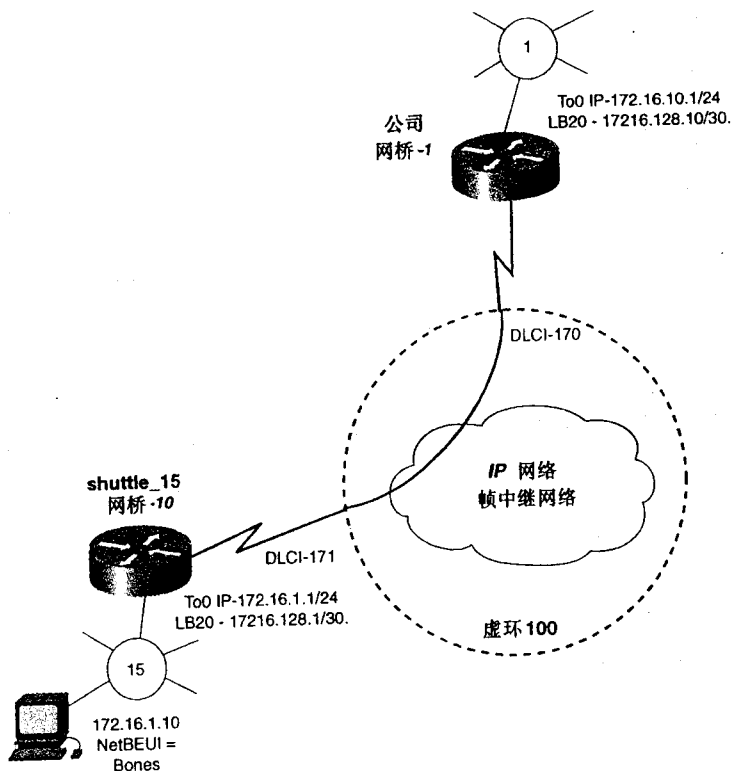


图 13-22 帧中继的 RSRB

例 13-19 RSRB 的帧中继封装形式

```
hostname enterprise
!
ip subnet-zero
!
```

(待续)

```

source-bridge ring-group 100
source-bridge remote-peer 100 frame-relay interface Serial0/1 170
!
<<<text omitted>>>
!
interface Serial0
mtu 4096
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
logging event subif-link-status
logging event dlci-status-change
frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
ip address 172.16.2.5 255.255.255.252
no ip directed-broadcast
frame-relay map rarb 170 broadcast
frame-relay map ip 172.16.2.6 170 broadcast
!
<<<text omitted>>>
!
interface TokenRing0
ip address 172.16.10.1 255.255.255.0
no ip directed-broadcast
ring-speed 16
multiring all
source-bridge 1 1 100
source-bridge spanning
!

hostname shuttle_15
!
ip subnet-zero
!
source-bridge ring-group 100
source-bridge remote-peer 100 frame-relay interface Serial0/1 171
!
interface Serial0
mtu 4096
ip address 172.16.2.6 255.255.255.252
no ip directed-broadcast
encapsulation frame-relay
no ip route-cache
no ip mroute-cache
logging event subif-link-status
logging event dlci-status-change
frame-relay map rarb 171 broadcast
frame-relay map ip 172.16.2.5 171 broadcast
!
<<<text omitted>>>
!
interface TokenRing0
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
ring-speed 16
multiring all
source-bridge 15 10 100
source-bridge spanning

```

在路由器 enterprise 上的执行示例。

例 13-20 路由器 enterprise 上的 RSRB 状态

```
enterprise#show source-bridge

Local Interfaces:
      srn bn  trn r p s n  max hops  receive  transmit  drops
To0      1  1  100 *   f   7  7  7    4223      0      0

Global RSRB Parameters:
  TCP Queue Length maximum: 100

Ring Group 100:
  No TCP peername set, TCP transport disabled
  Maximum output TCP queue length, per peer: 100
  Peers:
    FR Serial0.1      state  bg lv  pkts_rx  pkts_tx  expl_gn  drops TCP
    n/a
  Rings:
    bn: 1 rn: 1      local ma: 4007.781a.e789 TokenRing0      fwd: 0
    bn: 10 rn: 15    remote ma: 4000.309a.68bb FR Serial0.1    170 fwd: 0

Explorers: ----- input -----
      spanning all-rings  total  spanning all-rings  total
To0      1886      2337    4223      0      0      0

Explorer fastswitching enabled
Local switched: 4223      flushed 0      max Bps 38400

      rings  inputs  bursts  throttles  output drops
To0      4223      0      0      0      0

enterprise#
```

13.3.4 其他 SRB 功能与特性的配置

Cisco 提供了很多很有用的功能来对数据进行控制，以及对源路由桥接进行微调，包括：

- RSRB TCP 和 LLC2 的本地确认功能。
- 最大数据帧的设置。
- 生成树探测帧的设置。
- 静态 RIF。
- LSAP 和 MAC 过滤。

下面各节是对这些功能的概念与用法的讲述。

1. RSRB TCP LLC2 的本地确认功能

SNA 会话是完全的端到端的会话。前台处理器发出的每个数据帧都必须得到接收到该数据帧的工作站或控制器的确认。如果 SNA 会话需要通过低速链路（如 64 kbps）跨越较大的地理距离之间进行，那么 T1 计时器溢出超时的几率非常高。T1 计时器是一个预定义的时长，发送数据帧的主机希望在这段时间里能够收到接收该数据帧的主机的肯定或否定响应。所有的 LLC2 数据帧（包括管理帧 PR，RNR 和 REJ）都必须以这种端到端的方式进行确认。图

13-23 是 RSRB 环境下的一个典型的 LLC2 会话。

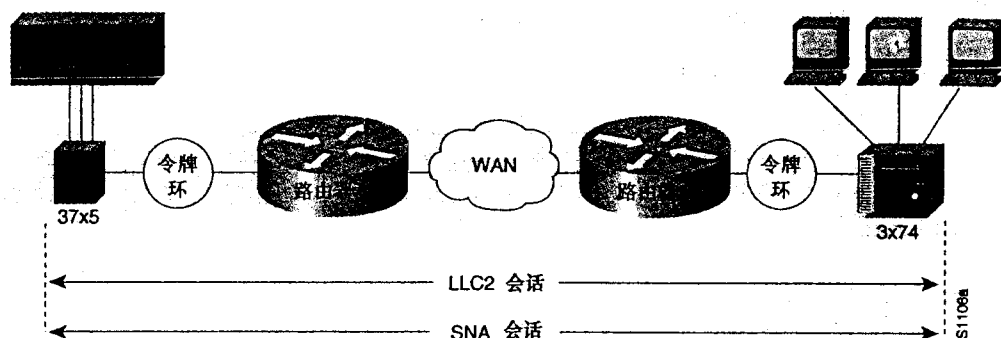


图 13-23 没有本地确认功能的 LLC2 会话

Cisco 为基于 TCP 的 RSRB 提供了本地确认功能，这样就无需改变终端节点的配置就能达到解决 T1 计时器问题的目的。启动本地确认功能之后，所有的 LLC2 数据帧都会得到路由器的确认。惟一通过网络传输的 LLC2 数据帧是 I 帧，即信息数据帧。图 13-24 就是 LLC2 本地确认的工作过程。

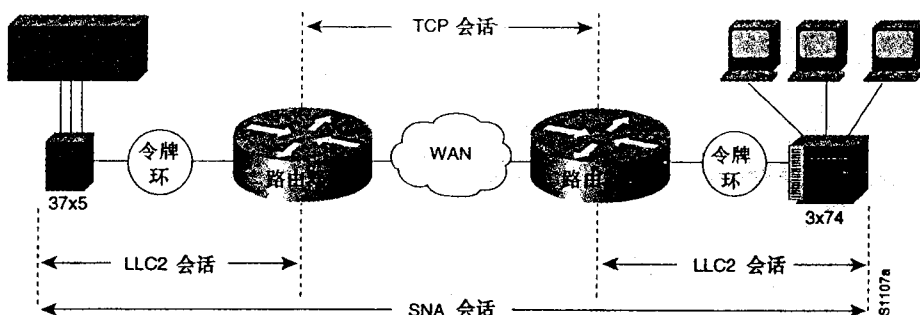


图 13-24 具有本地确认功能的 LLC2 会话

配置两个 RSRB 之间的本地确认功能，可在 **source-bridge remote-peer** 命令中使用 **local-ack** 参数，详细格式如下：

```
Router (config) #source-bridge remote-peer virtual_ring tcp ip_address local-ack
```

路由器必须与每个主机都维持一个完整的 LLC2 会话，因此，路由器能够同时支持会话的数目也就成为了应该考虑的因素。Cisco 建议只有在遇到了 T1 计时器问题或者是 LLC2 故障时才采用本地确认。本地确认功能不会影响 NetBIOS 的超时设置。

2. 最大数据帧大小的设置

在混合环境下，例如令牌环和以太网的混合，可以通过将最大数据帧的大小固定为 1500 或其他数值来避免整个网络中出现过多分段。将数据帧大小固定设为 1500 能使数据帧通过网络中的以太和令牌环网段时出现的数据分段数量大为减少。这一设置是通过帧中继、TCP 和 FST 的 **remote-peer** 命令中的 **if** 参数来实现的：

```
Router (config) #source-bridge remote-peer virtual_ring frame-relay interface
```

```
interface_name dlc_i_number [if largest frame size]
```

```
Router (config) #source-bridge remote-peer virtual_ring tcp ip_address
```

```
[ !f largest_frame_size] [ local-ack]
Router (config) #source-bridge remote-peer virtual_ring fst ip_address
[ !f largest_frame_size]
```

3. 生成树探测帧数据的配置

默认情况下，Cisco 路由器会利用全路由探测数据帧来产生 RIF。在大型冗余网络中，探测数据包在整个网络中复制和转发的过程中，其数量会按指数关系增加。前面讲过，生成树探测帧能够减少网络中的探测数据包的数量。生成树节点只把生成树探测数据发送到配置为生成树的节点去。用下面这条令牌环接口命令就可以启动 **spanning tree** 探测帧的工作：

```
Router (config-if) #source-bridge spanning
```

Microsoft NetBIOS 也使用生成树，因此，作为一个准则，使用 Microsoft Windows networking 时也应该配置生成树探测帧。

4. 静态 RIF 的控制

Cisco 提供静态配置路由器上 RIF 的方法。配置静态 RIF 必须非常熟悉路由控制字段和路由描述字段。回顾一下前面讲述源路由桥接时的字段格式图。

图 13-25 是 RIF 的路由控制字段的格式，后面是每个字段的解释。

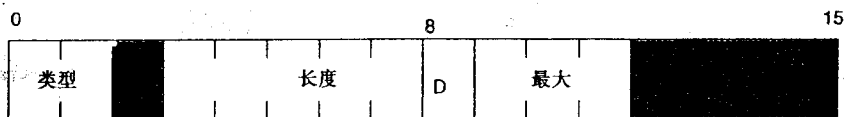


图 13-25 RIF 的路由控制字段格式

- 阴影部分是保留字段。
- **type**——探测帧所使用的类型：
 - **00**: 特定路由探测帧
 - **10**: 全令牌环，全路由探测帧
 - **11**: 全令牌环，跨越路由（受限广播）生成树探测帧
- **length**——RIF 中字节的总长度
- **D**——方向，含义如下：
 - **0**: 从左往右读取路由（正向）
 - **1**: 从右往左读取路由（后向）
- **largest**——该路由能够处理的最大数据帧长度：
 - **000**: 516 bytes (DDN 1822)
 - **001**: 1500 bytes (以太)
 - **010**: 2052 bytes
 - **011**: 4472 bytes (令牌环和 Cisco 的最大数据帧)
 - **100**: 8144 bytes (令牌总线)
 - **101**: 11407 bytes
 - **110**: 17800 bytes
 - **111**: 65535 (初始值)

图 13-26 是 RIF 的路由描述字段的格式。在配置静态 RIF 时，路由描述字段用点分十进

制数来表示。

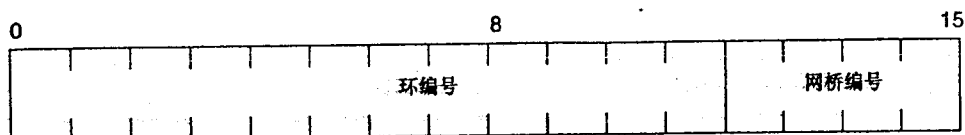


图 13-26 路由描述字段的格式

- **Ring number**——桥接网络中惟一的十进制令牌环号。
- **Bridge number**——连接相同两个令牌环的任意网桥之间的惟一网桥号。值为 0 的网桥号表示 RIF 终结点。

图 13-27 给出了一个 SRB 网络的例子。

RIF = 0830.0072.064a.00b0
or
RING7-BRIDGE2-RING100-
BRIDGE10-RING11

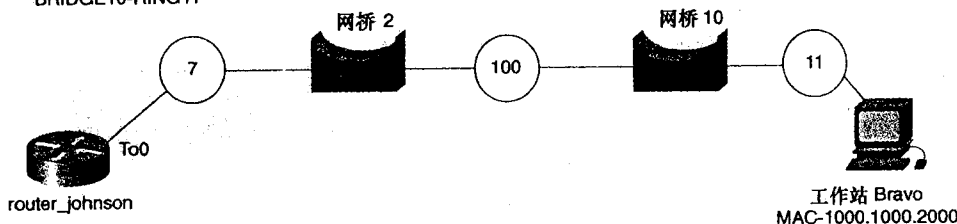


图 13-27 源路由桥接: RIF

例子中路由器 router_johnson 到工作站 Bravo 的静态 RIF 读作: 0830.0072.064a.00b0

用前面的例图将 RIF 分解成含义不同的几部分。0803 是 16 位的 RC 字段。从左往右读，位的含义是:

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0

头两位 (00) 将探测帧类型设置为特定路由探测帧，由于这是一个静态 RIF，因此采用这种浏览器类型。第 3 位设为 0，是保留位。下面的 5 位是设置 RIF 的字节长度。这个例子中，RIF (不仅仅是令牌环—网桥—令牌环部分，而是整个的 RIF) 是 8 字节。下一位 (D 位) 也是 0，说明 RIF 是从左往右读的，即正向。接下来的 3 位是 011，表明数据帧大小为 4472，即 Cisco 的最大帧尺寸。最后 4 位也是保留位。

后 3 个字节 (RD 字段) 也很容易分解。

这 3 个字节 (0072, 064a 和 00b0) 是 3 个 16 位的 RD 字段。每个字节的头 3 位是 16 进制的令牌环号。最后一位也是 16 进制的令牌环号。这个例子的 RIF 可以得出:

RING7-BRIDGE2 = 0072

RING100-BRIDGE10 = 064a

RING11-BRIDGE0 = 00b0

号为 0 的网桥让 SRB 终结 RIF，后面的环中不会再有网桥了。下面这条全局配置命令可用于配置路由器上的静态 RIF:


```
Router (config) #rif mac-address rif-string { interface-name | ring-group ring}
```

例 13-21 配置路由器 router_johnson 到工作站 bravo 的静态 RIF。

例 13-21 配置静态 RIF

```
router_johnson(config)#rif 1000.1000.2000 0830.0072.064a.00b0 too
```

5. LSAP、MAC 和 NetBIOS 的过滤

后面将再讨论 LSAP 和 MAC 过滤的问题，现在只说明源路由桥接环境中过滤器的应用方法。

下面命令用于配置 IEEE 802 封装数据帧的 LSAP 过滤：

```
Router (config-if) #source-bridge input-lsap-list access_list_number
Router (config-if) #source-bridge output-lsap-list access_list_number
Router (config) #rsrb remote-peer ring-group group {tcp ip_address | fst ip_address |
interface interface_name} lsap-output-list access_list_number
```

LSAP 访问列表的范围为 200 到 299，其过滤以 LSAP 类型代码为基础。

要以 IEEE 802 源地址为基础进行过滤，可以使用下面的命令：

```
Router (config-if) #source-bridge input-address-list access_list_number
Router (config-if) #source-bridge output-address-list access_list_number
```

其访问表列范围是 700 到 799。

要以 NetBIOS 名为基础进行过滤，可以采用下面的命令：

```
Router (config-if) #netbios input-access-filter host station_name
Router (config-if) #netbios output-access-filter host station_name
Router (config) #rsrb remote-peer ring-group {tcp ip_address | fst ip_address | interface
interface_name} netbios-output-list access_list_number
```

13.4 增强数据链路交换 (DLSw+)

众所周知，数据链路交换(DLSw)是 1995 年由高级对等网执行工作组 APPN AIW 在 IBM 的赞助下率先推出的。这并不是 DLSw 的第一个 RFC。IBM 曾经试图制定一种通过 TCP/IP 网络传输 LLC2 数据帧的方法，并于 1993 年起草了 RFC 1434。IBM 提出的方法原理非常健全，但是缺乏对不同厂商产品互操作性的支持（令人不解）。

APPN AIW 的目的是在最初的 DLSw RFC (RFC 1434) 的基础上开发出一些新的特性和功能。起初的一些成就汇编成了 DLSw 的第一份标准，即 RFC 1795。RFC 1795 对通常所称的数据链路交换 V1 进行了定义。

DLSw V2 在 1997 年完成，并编入到了 RFC 2166 中，它在版本 1 的基础上作了一些增强和改进，使得 DLSw 网络更容易进行扩展，而实用性也比 RSRB 或其他标准应用要好很多。思科把自己的 DLSw 应用称之为 DLSw+。DLSw+最显著的一个特点就是边界对等体与对等组的概念。

注释 阅读本书时，请把 DLSw 与 DLSw 版本 1、2 以及 Cisco 的 DLSw+看成同义。在讲到与版本或相关应用的部分时，本书会对其加以说明。

13.4.1 DLSw+的特性

1991 年时, RSRB 是很多网络工程师在 IP 网络中对令牌环或 LLC2 网络进行桥接时的惟一选择。很短的时间里, 数以千计的 RSRB 网络如雨后春笋般地出现。但是很快这些 RSRB 网络就让位给一种新型的通过 IP 网络传输 LLC2 的方式 DLSw 了。到 1995 年为止, 所有原来准备采用的 RSRB 网络都暂停建设, 因为整个行业很明显地都倾向于选择 DLSw 这个宠儿。从此以后, DLSw 就开始崭露头角, 并成为现代网络中用于传输传统协议内容的最可靠、最高效的方式之一。

DLSw 提供了一种通过 IP 运行 SNA 与 NetBIOS 协议的方式, 也提供了优于 RSRB 的可扩展性、多功能性、可管理性以及可控性。DLSw 通过在以太网设备引入本地确认功能, 为物理单元 (PU) 2.1 设备加入同步数据链路控制协议 (SDLC), 以及其他一些主要特性和功能解决了 RSRB 的一些缺陷, 还提供了更高的可用性, 包括负载平衡、备份性能等。

与 SRB 相比, DLSw+的一些优越之处包括:

- 支持多厂商产品的互操作。
- 数据链路控制 (DLC) 的超时设置。
- 广域网 WAN 上的数据链路控制 (DLC) 确认功能。
- 电路级数据流和拥塞控制。
- 对等体优先级设置和端口负载共享。
- 以太网上的本地确认功能。
- 备份、动态和高容错率的对等体。
- 增强型广播减少机制。
- UI 数据帧的用户数据协议 (UDP)。
- RIF 的终止性使网络范围扩展。
- 对等体组和边界对等体缓存的广播优化。
- 一些增强提高性的性能, 包括:
 - 内置媒体转换功能 (PU 2.0, 2.1 和 4)
 - 支持 Token Ring LANE, Token Ring ISL 和 SRB FDDI 上的终端系统
 - 详细的功能交换
 - SNA DDR

注释

DLSw, 对传统协议进行集成的惟一途径

在实际应用中, 可以说 DLSw+是最可靠而配置最简单的协议之一。SNA 对时间非常敏感, 而 DLSw+在网络中传输 SNA 的简单与方便是极为杰出的。在学会 DLSw+的配置之后, 很快我就把 DLSw+作为了我自己传输不可路由协议 (如 SNA, NetBEUI, NetBIOS) 的惟一选择。相信大家在完成这一节的学习之后也会和我一样。

13.4.2 DLSw+技术概览

IEEE 802.2 LLC2 在设计时，假设网络传输延迟时间很小而且可预测。令牌环和以太网协议都是针对局域网。如果跨越很大的物理距离配置远程网桥，网络延时会随着链路的改变而发生剧烈的变化。延时太大时会出现 LLC2 超时以及数据重新发送的情况。由于数据帧只是延迟了一段时间，LLC2 发现复制的数据帧时会产生混乱，就有可能断开 LLC2 会话。图 13-28 是 LLC2 如何在一个传统网桥环境中进行端对端确认的例子。

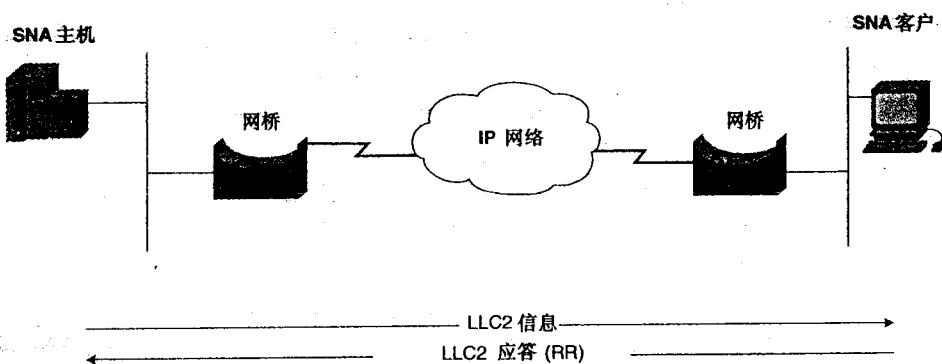


图 13-28 LLC2 的端对端确认

DLSw+会在 DLSw+设备/路由器上终止 LLC2 连接，这样，LLC2 链路就不会再穿过 WAN 传输，而 WAN 正是产生较长延迟时间的地方。现在的 LLC 层延时只出现在局域网中。这样一来，SDLC 链路、查询以及查询响应等数据就只出现在本地而不是在整个 WAN 中，从而大大减少 WAN 上的数据量。查找数据帧的广播也是处于 DLSw+路由器的控制之下，而不是路由器定位目的工作站之后发出去的。图 13-29 是 DLSw+环境中确认过程的处理情况。

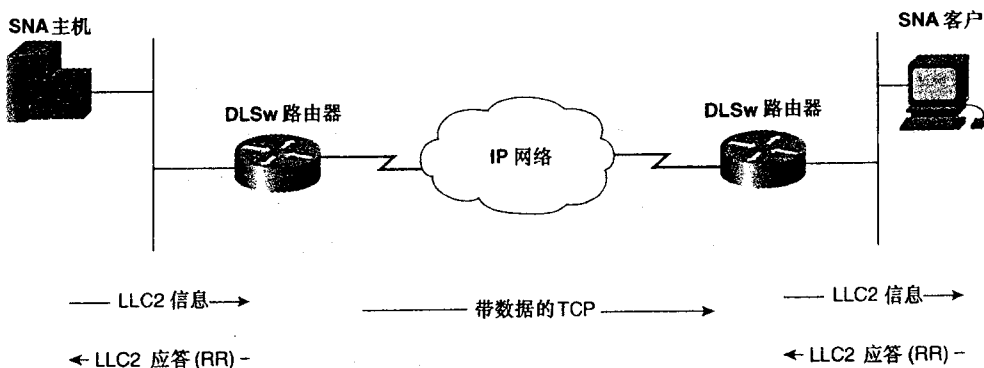


图 13-29 LLC2 的本地确认

前面提过，DLSw+提供了一种通过 TCP/IP 传输 SNA、NetBIOS 和 NetBEUI 的方法。DLSw+将路由器称为对等体。通过 TCP 或其他传输协议与另一台路由器建立连接时，称为

这两台工作站的数据就是通过这一电路进行传输的。一个对等体可以支持多个电路，这些电路的类型包括：

- SNA PU 类型 2.0/2.1 和 SNA PU 类型 4。
- 从 SDLC PU1 类型 1 到 PU 类型 4。
- NetBIOS。
- Windows 9.x NetBEUI。

默认情况下，DLSw+对等体每过 30 秒向所有的对等体发送一次存活（keepalive）信号。存活（keepalive）是用来确保与对等体的连接正常运转的。如果连续 3 次存活（keepalive）信号丢失，则与该对等体的连接就会断开。

数据链接交换机（对我们来说，即 Cisco 路由器）采用交换机到交换机协议（SSP）来建立 DLSw+的连接。一对路由器可以利用 SSP 协议在一个对等体连接上进行可靠传输的数据链路多路复用。两台路由器之间可以使用 DLSw+之前，必须相互建立对等体的关系。DLSw 对等体用于连接的传输协议是 TCP。Cisco 的 DLSw+提供了 4 种类型的传输方式：

- **TCP 封装方式**——TCP 用于需要本地确认功能的情况，有助于避免数据链路控制超时的出现。TCP 封装方式还提供了备份对等体以及其他一些特性，以实现链路失败时不受干扰的重路由功能。

TCP 是所有传输协议中最为灵活，也是最为可靠的。但是，由于有 20 字节的 TCP 报头，20 字节的 IP 报头以及 16 字节的 DLSw 报头，TCP 需要的系统开销最多。但在现代网络中，这样的系统开销微不足道。默认情况下，DLSw TCP 封装方式会在端口 2067 上对 TCP 进行侦听，在端口 2065 上发送 TCP 数据包。也可以通过 TCP 端口号为端口分配优先级。表 13-3 列出了所用的端口以及相应的优先级情况。

表 13-2

TCP 的端口优先级

优先级	端口
高	2065
中	1981
普通	1982
低	1983

TCP 封装方式还允许通过 LSAP、DMAC 以及 NetBIOS 名称过滤的办法对能力信息交换和探测数据帧进行综合控制。TCP 对等体可以在任何支持 TCP/IP 的 LAN 或 WAN 上存在。

- **快速序列传输（FST）封装方式**——FST 封装方式是通过 IP 传输 DLSw 的一种低系统开销方式。这种方式不具有数据帧可靠传输和本地确认的功能。所有的存活（keepalive）数据帧都是通过 FST 传输方式从一个端点传输到另一个端点的。FST 利用 IP 作为其传输途径，连接失败时还会进行重路由。只有终端系统处在令牌环网中时才能使用 FST 封装方式，而 FST 对等体只能通过 HDLC、以太网、令牌环网、FDDI、ATM 和帧中继网建立。
- **直接封装方式**——直接封装方式是通过 HDLC 或帧中继连接传输 LLC 的一种低开销方式。它只包含一个 16 字节的 DLSw 报头以及用来传输的数据帧报头。这种直

接方式不支持数据帧可靠传输和本地确认功能。和 FST 一样, 存活 (keepalive) 数据帧也会从一个端点传输到另一个端点, 直接封装方式只能用于终端系统处在令牌环网中的情况。

- **LLC2 封装方式 (DLSw Lite)** ——这种封装方式也是一种低系统开销方式, 采用 RFC 1490 (具有 16 字节的报头), 允许在帧中继数据帧中使用直接封装方式。当然, 该方式只能用于帧中继网中。DLSw+ Lite 支持本地确认以及数据帧的可靠传输。链路失效也不会对 DLSw + Lite 对等体产生影响。

1. DLSw 的电路建立

电路建立发生在两个终端系统之间。某个终端工作站产生具有某个特定 MAC 地址的 SNA TEST 或 XID 探测数据帧时, SNA 电路就会建立。DLSw 路由器向每个工作中的对等体发送 CANUREACH 数据帧。正常情况下, 对等体会以一个 ICANREACH 数据帧作为响应。在交换了一系列的 XID 和其他信息之后, 电路就建立了。每条电路都具有一个惟一的 ID 号, 一个 TCP 对等体能够支持多条电路。这个 ID 号包括源和目的端的 MAC 地址、源和目的 LSAP 以及一个数据链路端口 ID。只有电路进入连接状态之后, 数据才能在主机之间进行交换。每个 DLSw 都会将 MAC 地址和 NetBIOS 名称缓存下来, 以避免后来的探测数据帧再在网络进行传输。图 13-30 和 13-31 摘自 RFC 1795, 演示了 SNA 和 NetBIOS 的 DLSw 电路建立的完整过程。

NetBIOS 工作站的电路建立过程也是相似的, 只是发送的不是 CANUREACH 数据帧, 而是一个有指定 NetBIOS 名字的 NetBIOS 名字查询数据帧。图 13-31 是 NetBIOS 的电路建立全过程。

2. DLSw 的能力交换

DLSw 电路建立的一个关键过程是能力交换。能力交换过程有别于 DLSw 于其他桥接技术, 交换内容为一个特殊的 DLSw SSP 控制消息, 该消息用来描述 DLSw 发送方路由器的性能。两台 DLSw 设备建立新的连接以后, 首先发送出去的 SSP 消息就是初始的能力交换信息, 用来描述 DLSw 版本号以及 DLSw 设备其他性能的信息。

SSP 采用称为控制向量的概念在 DLSw 设备之间进行信息交换。首先应该发送的必须是这个控制向量, 顺序是:

- 1 厂商 ID。
- 2 DLSw 版本号。
- 3 初始步调窗口。
- 4 所支持的 SAP 列表。

控制向量中剩余部分不需要严格按照顺序发送, 包括:

- 厂商 ID 控制向量。
- DLSw 版本号控制向量, 表明所用的 DLSw 标准。
- 初始步调窗口控制向量, 用于数据流的控制。
- 版本号字符串向量。
- MAC 地址专用性控制向量, 包括的只是该交换机可以访问的 MAC 地址。
- 所支持的 SAP 列表控制向量, 包括所有的 SAP 的列表。

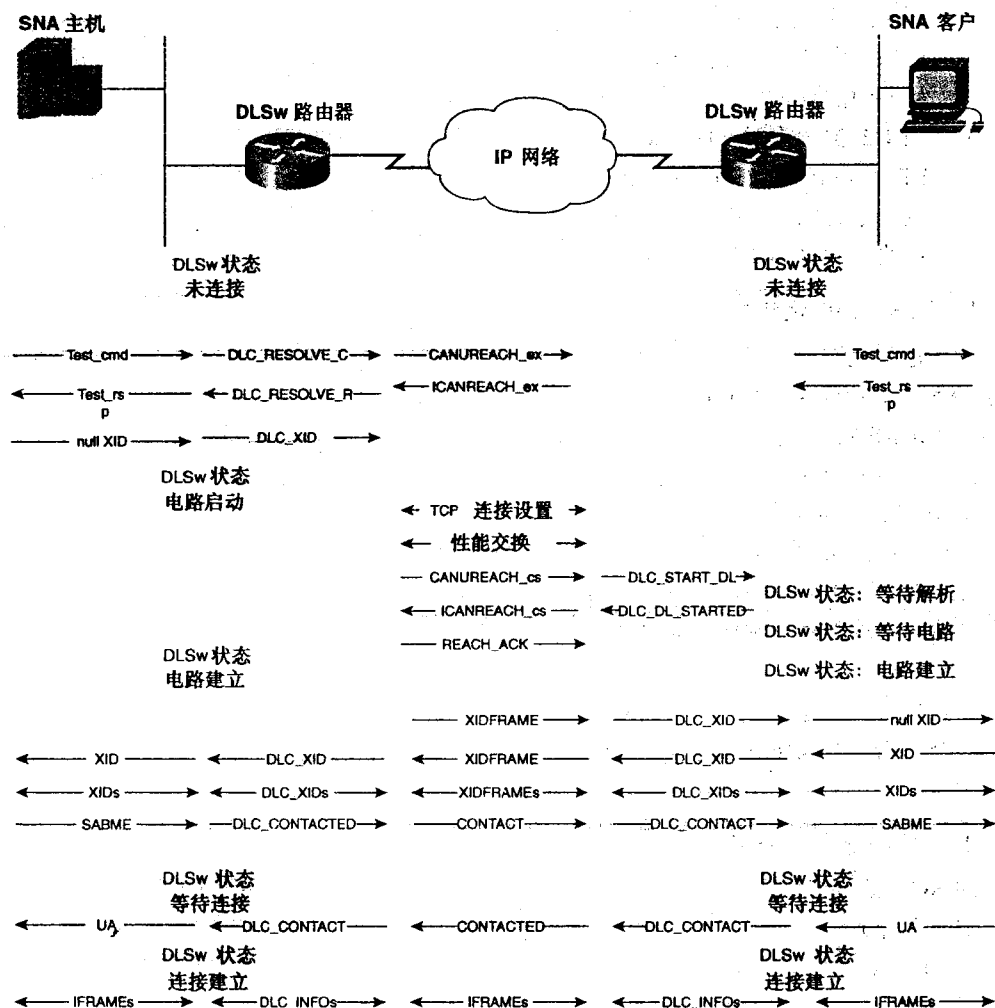


图 13-30 SNA 的电路建立过程

- TCP 连接控制向量，用于指定要支持的 TCP 连接的数目。
- NetBIOS 名称专用控制向量。
- MAC 地址列表控制向量。
- NetBIOS 名称列表控制向量。
- 厂商内容控制向量。
- 保留备用。
- 厂商专用。

Cisco 的 TCP 连接控制向量应用实施规定只能有一个 TCP 连接用于传输数据。DLSw+ 电路建立之初，两个 TCP 连接都处于活动状态之中。DLSw+ 指定将最高的 IP 地址连接断开，以保证只有一个 TCP 连接用于数据传输。

能力交换结束之后，DLSw 路由器会接着完成整个电路建立的过程。只有有一个电路进入了连接状态之后，数据才能够通过该电路进行交换。

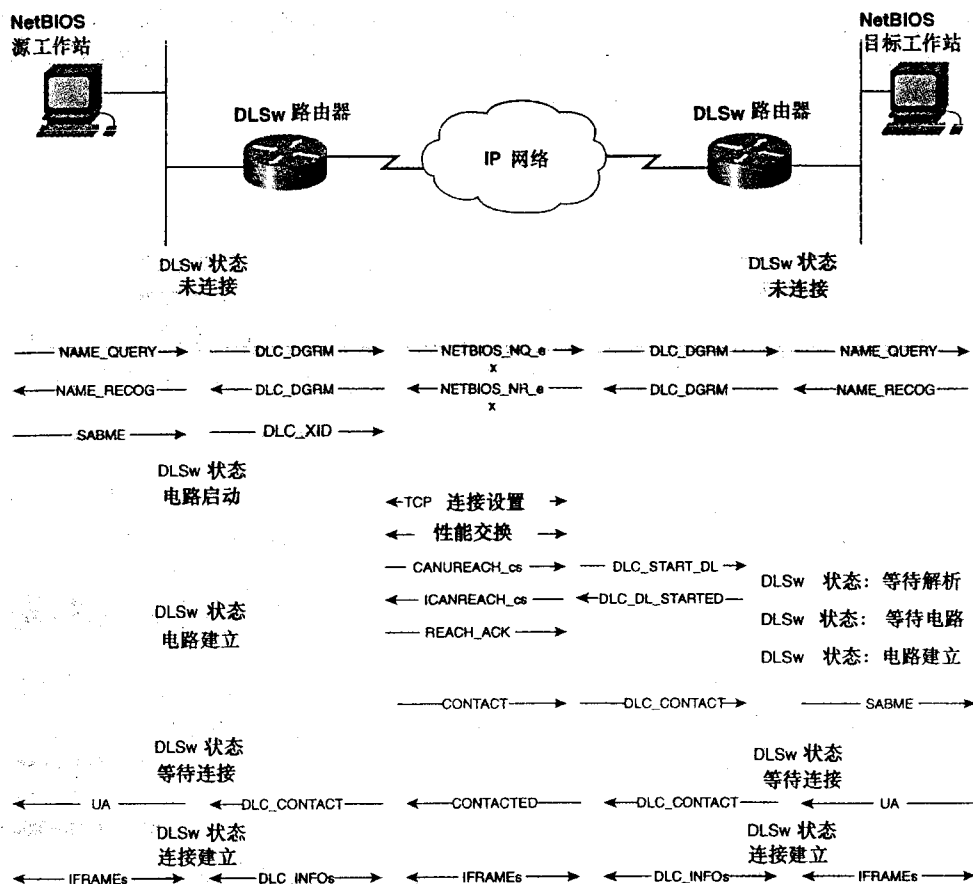


图 13-31 NetBIOS 的电路建立过程

3. DLSw 的数据流控制

DLSw 采用自适应的步调方式控制数据流，指定了两条独立的单向电路来控制每条电路中的数据流。DLSw 采用基于缓冲器大小的动态窗口、TCP 传输队列以及终端工作站数据流控制机制实现数据流的控制。

4. DLSw 的 RIF 终止功能和通过功能

在使用源路由桥接时，DLSw 支持 RIF 终止功能。所有的远程设备/网桥看上去都和虚拟令牌环相连。默认情况下，RIF 由 DLSw 路由器终止。此外，还可以为 DLSw 对等体配置 RIF 通过功能。在配置这一功能时，记住一定要保证两端对等体的虚拟令牌环的匹配，以及双方的 RIF 通过功能都同时启动。

5. 规范与非规范的地址格式

回顾一下第 2 章以太网使用的规范位格式和令牌环网使用的非规范格式。从以太网桥接收到一个数据帧时，DLSw 会把它转换成非规范的格式。在将数据从一个对等体传输到另一个时，DLSw 严格使用非规范格式。数据帧到达目的接口时，DLSw 会对该接口进行检查，确定是以太网接口还是令牌环接口。如果是以太网接口，数据帧又会转换成规范格式，然后再从

该接口发送出去；如果是令牌环接口，则数据帧无需转换，直接就会从这个接口发送出去。在将 SNA 与以太网接口连接使用时要注意它们应该工作在规范格式下。大多数的 SNA 设备（如 IBM 3174）在以太网中都是在规范格式下工作的。

13.4.3 DLSw+的配置

本书要讨论的是令牌环和以太网上的 DLSw 配置。可以参考 Cisco Press 出版的《Cisco IOS Bridging and IBM Network Solutions》一书中关于 FDDI、SDLC 和 QLLC 上的配置讲解。

DLSw 的配置包括 4 个步骤，只需要为局域网接口配置桥接以及定义本地和远程的 DLSw 对等体。当然配置过程还需要其他操作，但基本 DLSw 配置这 4 个步骤即可。这也能说明为什么 DLSw 能够如此快地得到普及。

下面是配置 DLSw 必须的 4 个步骤：

第 1 步 配置 IP 环路接口并将其添加到要运行 DLSw 的路由器所在路由区域中。和 RSRB 和 OSPF 一样，使用环路地址可以使 DLSw 对等体的工作更加稳定。使用逻辑接口之后，DLSw 对等体的连接不再需要由物理接口的状态来决定，这在多接口路由器上非常重要。要确保环路 IP 地址必须是远程对等体路由器可以访问到的，通常用一个路由选择协议来传播环路地址。

第 2 步 为 DLSw 定义一个本地对等体。本地对等体的创建可以激活路由器中的 DLSw 代码的运行。本地对等体的 IP 地址应该是第 1 步中配置的环路接口的地址。下面是所需要的一些命令的句法格式，后面还会再讨论一些相关的命令参数：

```
Router (config) #dlsw local-peer [peer-id ip_address]
Router (config) #dlsw local-peer [peer-id ip_address] [group peer_group_1-255]
[border] [cost 1-5] [lf largest_frame 516-11407] [keepalive seconds]
[passive] [promiscuous] [init-pacing-window size_1-2000]
[max-pacing-window size] [biu-segment]
```

第 3 步 启动以太网接口上的透明桥接，或者令牌环接口上的源路桥接。源路由桥接会通过虚拟令牌环自动连接到 DLSw。以太网透明桥接到 DLSw 还需要一些附加的命令，这一点后面还会进行描述。可以把网桥看成一个广播或 LLC2 捕捉实体。数据帧被捕捉以后会传输到远程对等体。而远程对等体传来的数据则只会转发到虚拟令牌环或者是 dlsw bridge-group 命令定义的透明桥接组。可以参考前面讲过的透明桥接和源路由桥接部分的内容。

—— 以太网：在 SNA、NetBIOS 或其他 LLC2 协议的以太网接口上启动透明桥接可以参考下面的例子。这里使用的网桥号是 10。

```
Router (config) #bridge 10 protocol ieee
Router (config) #dlsw bridge-group 10
Router (config-if) #bridge-group 10
```

命令 dlsw bridge-group 的完整格式是：

```
Router (config) # dlsw bridge-group group-number [llc2 [N2 number]
[ack-delay-time milliseconds] [ack-max number][idle-time milliseconds]
[local-window number][t1-time milliseconds] [tbusy-time milliseconds][tpf-time
milliseconds] [trej-time milliseconds][txq- maxnumber] [xid-neg-val-time
milliseconds] [xid-neg-val-time milliseconds] [local-address priority list
list number] [sap-priority priority list number]
```


—— 令牌环网络：下面是在 SNA、NetBIOS 或其他 LLC2 协议的令牌环接口上配置源路由桥接的例子：这个过程中必须定义一个 SRB 的虚拟令牌环，本例中使用的虚拟令牌环号是 100，令牌环接口的令牌环号是 1。

```
Router (config) #source-bridge ring-group 100
```

```
Router (config-if) #source-bridge 1 2 100
```

第 4 步 确定对等体要采用的封装类型。前面提过，可供选择的封装类型共有 4 个，表 13-3 列出了各种不同类型的封装类型以及所支持的一些功能。

表 13-3

DLSw 封装类型与功能列表

封装类型	可靠传输	本地应答	连接失败时无干扰重路由	DLSw+ 的系统开销**	支持端系统拓扑	支持备份对等体
TCP	是	是	是	56 字节	全部	是
FST	否	否	否*	36 字节	只有令牌环	是
直接	否	否	否	16 字节	只有令牌环	否
DLSw Lite	是	是	否*	20 字节	令牌环、以太网、SDLC、QLLC	否

* FST 和 DLSw 这两种类型可以通过配置支持连接失败时的无干扰重路由功能。注意，重路由期间，会话是断开的。

** 这里指的系统开销没有包括 DLC 或者是与 DLSw 相关联的数据帧报头在内。

下面的这些全局配置命令可用于远程对等体上对各种不同的封装类型进行配置。远程对等体的 IP 地址是远程对等体上环路接口的 IP 地址。每个列表中的第 1 条命令都只需在远程对等体上作最少的配置工作，随后是各个封装类型的远程对等体命令的完整命令句法：

—— TCP 封装方式：

```
Router (config) #dlsw remote-peer 0 tcp ip_address
Router (config) # dlsw remote-peer list-number tcp ip-address [backup-peer
[ ip-address | frame-relay interface serial number dlci-number |
interface name]] [bytes-netbios-out bytes-list-name] [circuit-weight weight]
[cluster cluster-id] [cost cost] [dest-mac mac-address] [dmac-output-list
access-list-number]
[host-netbios-out host-list-name] [inactivity] [dynamic] [keepalive seconds] [lf size]
[lingerminutes] [lsap-output-list list] [no-llc minutes] [passive] [priority]
[rif-passthru
virtual-ring-number] [tcp-queue-max size] [timeout seconds]
```

—— FST 封装方式：

```
Router (config) #dlsw remote-peer 0 fst ip_address
Router (config) # dlsw remote-peer list-number fst ip-address [backup-peer [ ip-address
[bytes-netbios-out bytes-list-name] [circuit-weight weight] [cost cost] [dest-ma
mac-address] [dmac-output-list access-list-number] [host-netbios-out host-list-nan
[keepalive seconds] [lf size] [linger minutes] [lsap-output-list list]
```

—— 帧中继的直接封装方式：

如果接口是多点接口，还需要添加一条 frame-relay map dlsw 命令：

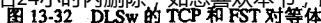
```
Router (config) #dlsw remote-peer 0 frame-relay interface serial number DLCI_number
Router (config-if) #frame-relay map dlsw DLCI_number
```

```
Router (config) #dlsw remote-peer 0 interface serial number
Router (config) # dlsw remote-peer list-number interface
serial number [bytes-netbios-out bytes-list-name]
[circuit-weight weight] [cost cost] [dest-mac mac-address]
[dmac-output-list access-list-number]
[host-netbios-out host-list-name] [keepalive seconds] [lf size]
[linger minutes] [lsap-output-list list] pass-thru
```

如果接口是多点接口，还必须添加一条 **frame-relay map llc2** 命令：

```
Router (config) #dlsw remote-peer 0 frame-relay interface serial number DLCI_number
Router (config-if) #frame-relay map llc2 DLCI_number
dlsw remote-peer list-number frame-relay interface serial number
dlci-number [bytes-netbios-out bytes-list-name] [circuit-weight weight]
[cost cost] [dest-mac mac-address] [dmac-output-list access-list-number]
[host-netbios-out host-list-name] [keepalive seconds] [lf size]
[linger minutes] [lsap-output-list list] pass-thru
```

图 13-32 为连接 4 台路由器的一个帧中继网络模型。要在模型中创建两种类型的 DLSw



连接。从路由器 skywalker 创建一个到路由器 solo 的 TCP 对等体，从路由器 vader 创建一个到路由器 chewbacca 的 FST 对等体。

首先创建 skywalker 和 solo 之间的 TCP 对等体。按照上面所示的 4 步骤的配置过程，开始在 skywalker 和 solo 上创建环路接口，并为它们分配 IP 地址，这时要确保这些地址可以通过路由选择协议进行发送。环路地址会在本地和远程的对等体中使用，因此必须保证二者之间连接正常。这个模型中使用 EIGRP 作为路由选择协议，因此 EIGRP 一定要通告所有的环路地址。所以在进行第 2 步的操作之前，先要确认模型中所有相关的 IP 地址都可以 ping 通。这样不必以后再花时间去检查 IP 连接引起的问题。

第 2 步是配置每台路由器的 DLSw 本地对等体。记住，这里要使用的 IP 地址是环路地址。全局配置命令 `dls local-peer peer-id ip_address` 可以用来配置本地对等体。例如，配置路由器 skywalker 的本地对等体，可以使用 `dls local-peer peer-id 172.16.128.5`。

配置完本地对等体后要定义一个透明桥接组。在以太网中，必须在终端工作站所在的以太网段上配置透明桥接。然后，用全局命令 `dls bridge-group bridge-group` 将透明桥接组连接到 DLSw。例 13-22 给出了到目前为止路由器 solo 上相关配置部分的内容。

例 13-22 路由器 solo 上的 DLSw TCP 配置

```
hostname solo
!
<<<text omitted>>>
!
dls local-peer peer-id 172.16.128.0      -- IP address of Loopback 20
dls remote-peer 0 tcp 172.16.128.5      -- Configured in step 4. IP address of
                                         Skywalker
dls bridge-group 1                      -- Must match the bridge group on E0
!
interface Loopback20
 ip address 172.16.128.9 255.255.255.252
 no ip directed-broadcast
!
interface Ethernet0
 ip address 172.16.6.1 255.255.255.0
 no ip directed-broadcast
 bridge-group 1                         -- Transparent Bridging enabled for E0
!
interface Serial0
 ip address 172.16.1.6 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 131 broadcast
 frame-relay map ip 172.16.1.1 131 broadcast
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
router eigrp 65001
 network 172.16.0.0
 no auto-summary
!
ip classless
 no ip http server
!
```

(待续)

```
bridge 1 protocol ieee
<<<transparent bridging enabled>>>
```

要配置 TCP 的远程对等体，可以使用 **dls w remote-peer 0 tcp ip_address** 命令。除非采用了 DLSw 端口列表，否则命令中都是使用参数 0，说明没有使用任何列表。上例中有这条命令在路由器 solo 上的使用方法。例 13-23 是路由器 skywalker 上用这条命令进行配置的例子。

例 13-23 路由器 skywalker 上的 DLSw TCP 配置

```
hostname skywalker
!
<<<text omitted>>>
!
dls w local-peer peer-id 172.16.128.5
dls w remote-peer 0 tcp 172.16.128.5
dls w bridge-group 1
!
interface Loopback20
 ip address 172.16.128.5 255.255.255.252
!
interface Ethernet0
 ip address 172.16.5.1 255.255.255.0
!
bridge-group 1
!
interface Serial0
 ip address 172.16.1.5 255.255.255.0
 encapsulation frame-relay
 no arp frame-relay
 frame-relay map ip 172.16.1.6 111 broadcast
 frame-relay map ip 172.16.1.1 111 broadcast
 no frame-relay inverse-arp
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
router eigrp 65001
 network 172.16.0.0
 no auto-summary
!
<<<text omitted>>>
!
bridge 1 protocol ieee
```

可以用 **show dls w peers** 命令查看对等体的状态。例 13-24 是该命令在 solo 上的执行结果。只有对等体处于连接状态时才能创建电路传输数据。请注意，处在连接状态中的对等体并不意味着 DLSw 完全可用。DLSw 电路中的 **ckts** 标识是端对端会话在 TCP 对等体中是否存在的惟一标志，后面还会讨论这些内容。

例 13-24 show dks w peers 命令的执行示例

```
solo#show dls w peers
Peers:                state      pkts_rx  pkts_tx  type  drops  ckts TCP  uptime
TCP 172.16.128.5      CONNECT      255      444  conf      0      1  0 00:39:33

Total number of connected peers: 1
Total number of connections: 1
```

```
solo#
```

例 13-25 是另外一种获取更全面的 DLSw 电路信息的方法，如下例的 **show dlsw circuits** 命令使用示例。

例 13-25 show dlsw circuits 命令的执行示例

```
solo#show dlsw circuits
Index          local addr(lsap)  remote addr(dsap)  state      uptime
1778384900     0000.613c.dc82(F0) 0005.332e.2a25(F0) CONNECTED  00:12:27
Total number of circuits connected: 1
```

注释 该模型中是使用 NetBEUI 协议和 Windows 网络功能的 Windows 98 工作站来启动 DLSw 电路的。Windows 网络是测试 DLSw 的一个很不错的应用程序。要真正地创建一条电路，需要做的是浏览“网络邻居”，登录工作站，访问某个网络资源，如查看一个文件夹。用尽量多的应用程序对网络进行测试很重要，这是用于确定所作的配置是否成功的惟一依据。例如，在这个模型里，如果没有工作站，就看不到电路是否真正创建。因此，大家可能会想“好吧，我可以查看一下对等体和 DLSw 性能的状态，然后就会明白我做的配置是否正确了”。但是，假设我们忘记了在以太网接口上启动桥接，要这时对等体依然会处于活动状态中，从 DLSw 的角度来说，一切配置都正常。但是，由于忽略了这个关键因素，DLSw 不会转发任何数据到以太网段。图 13-33 显示了工作站 R2-D2 查找并查看 luke 上的文件的过程。

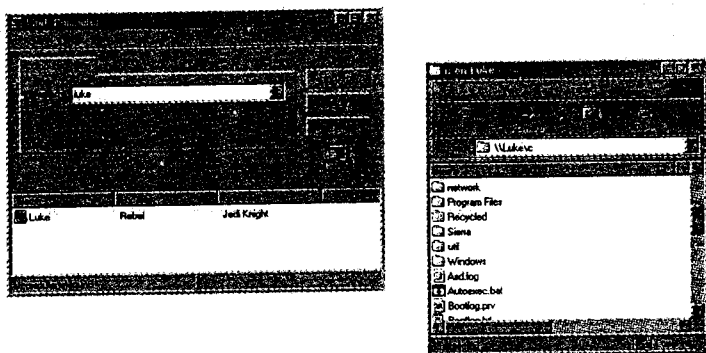


图 13-33 使用 Windows 9x 测试 DLSw

要配置这个模型中的 FST 对等体，首先要给路由器 vader 和 chewbacca 加上环路接口及地址。

第 2 个步骤为每台路由器加上其本地对等体，并使这些本地对等体指向刚创建的环路接口。

本地 FST 对等体所用的命令和 TCP 对等体完全一样。例如，在 vader 路由器上使用 **dlsw local-peer peer-id 172.16.128.1** 命令，在 chewbacca 上使用 **dlsw local-peer peer-id 172.16.128.13** 命令。

下一步是配置源路由桥接，为所有含有终端工作站的接口分配虚拟令牌环。虚拟令牌环是子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责

DLSw 与源路由桥接的连接纽带。这里没有必要使用 **dlsw bridge-group** 这样的命令进行配置。

第 4 步是要求对 DLSw 远程 FST 对等体进行配置。在路由器 **chewbacca** 上完成这一工作的命令是 **dlsw remote-peer 0 fst 172.16.128.1**。例 13-26 分别列出了 **chewbacca** 和 **vader** 上的配置过程。

例 13-26 路由器 **chewbacca** 和 **vader** 上的 FST 对等体配置

```

hostname chewbacca
!
<<<text omitted>>>
!
source-bridge ring-group 111          - virtual ring
dlsw local-peer peer-id 172.16.128.13  - IP address of Loopback 20
dlsw remote-peer 0 fst 172.16.128.1    - FST peer to Loopback address on Vader
!
interface Loopback20
 ip address 172.16.128.13 255.255.255.252
!
interface Serial0
 ip address 172.16.2.2 255.255.255.252
 encapsulation frame-relay
 frame-relay interface-dlci 181
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.3.1 255.255.255.0
 ring-speed 16
source-bridge 11 2 111                - SHB enabled
source-bridge spanning
!
router eigrp 65001
 network 172.16.0.0
 no auto-summary
!

hostname vader
!
<<<text omitted>>>
!
source-bridge ring-group 110
dlsw local-peer peer-id 172.16.128.1
dlsw remote-peer 0 fst 172.16.128.13
!
!
interface Loopback20
 ip address 172.16.128.1 255.255.255.252
 no ip directed-broadcast
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 logging event subif-link-status
 logging event dlci-status-change

```

```

frame-relay lmi-type cisco
!
interface Serial0/1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 no ip split-horizon eigrp 65001
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 130 broadcast
!
interface Serial0/2 point-to-point
 ip address 172.16.2.1 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 180
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.30.1 255.255.255.0
 no ip directed-broadcast
 ring-speed 16
 source-bridge 10.1.110
 source-bridge spanning
!
<<<text omitted>>>
!
router eigrp 65001
 network 172.16.0.0
 no auto-summary
!

```

现在再用 **show dls w peer** 命令来查看以下对等体的状态信息，如例 13-27 所示。

例 13-27 路由器 chewbacca 上的 FST 对等体状态

```

chewbacca#show dls w peers
Peers:                state      pkts_rx  pkts_tx  type  drops  ckts TCP  uptime
-----
FST 172.16.128.1      CONNECT    1635      1371   conf      0      1  - 02:23:08

Expected: 230 Next Send: 194 Seq errors: 0
Total number of connected peers: 1
Total number of connections: 1

chewbacca#

```

13.4.5 DLSw+的“Big show”和“Big D”命令

DLSw+的 **show** 命令和 **debug** 命令非常全面，包括很多用来调试以及显示 DLSw+信息的命令和子命令。这里不列出全部的 DLSw+相关命令，只列出被称之为“Big show”和“Big D”的一些命令。重要的 **show** 命令和 **debug** 命令的参考实例都是以上面的模型为基础进行操作的。DLSw+的“Big D”命令包括 **debug dls w peers**、**debug dls w core**、**debug dls w reachability** 及其子命令。这些命令的执行输出结果很多，因此最好是用全局配置命令 **logging buffered 10000** 启动日志记录。

“Big show”和“Big D”的命令句法如下：

```

show dls w peer [interface interface_name | ip-address ip_address_of_peer]

```

```
show dlsw reachability [mac-address mac_address] [netbios-name name]
show dlsw circuits [detail] [-circuit_number] [-mac-address address] [-sap-value value]
| circuit_id]
show dlsw capabilities [interface type number | ip_address ip_address | local]
debug dlsw peers [interface type number | ip_address ip_address]
debug dlsw reachability [error | verbose] [netbios | sna]
debug dlsw core [flow-control | messages] [state | xis]
```

1. show dlsw peer 命令

该命令能够显示静态以及处于直连状态的当前对等体信息。例 13-28 中该命令的执行结果包括对等体类型、TCP、FST 或是直接封装类型接口号。类型字段描述对等体是否已配置，是混杂形式还是按需分配的对等体 (POD)。

例 13-28 show dlsw peer 命令的输出结果

```
skywalker#show dlsw peer
Peers:                state      pkts_rx  pkts_tx  type  drops  ckts  TCP  uptime

TCP 172.16.128.9      CONNECT      1863      847  conf      0      1      0 04:19:26

Total number of connected peers: 1
Total number of connections: 1
```

对等体的可能状态包括：

- **Connect**——DLSw 处于工作状态，且具有一个激活的传输连接，这是对等体的正常状态。
- **DISCONNECT**——本地对等体不具有通往远程对等体的有效或激活地传输连接。
- **CAP_EXG**——本地对等体处于和远程对等体的能力交换阶段。本地对等体正在等待远程的能力交换响应。
- **WAIT_RD**——对等体建立过程的最后一步，本地对等体的 TCP 写管道（即 TCP 端口 2065）正在等待远程对等体打开它的读端口（即 TCP 端口 2067）。
- **WAN_BUSY**——TCP 输出队列已满，数据包不能再进行发送。

命令 **show dlsw peer** 还能显示发送以及接收的数据包编号以及丢失次数。TCP 一列是对等体的 TCP 队列，这个数字应该保持很小的值，大多数时候是 0，至少应该小于 10。较高的 TCP 数字就表明与远程对等体的连接出现了阻塞或数据传输的问题。另外，ckts 那一列列出的数字是对等体上活动电路的数目。

2. show dlsw reachability 命令

这条命令在确认哪台工作站 DLSw 处在当前的缓存中十分有用。可达性缓存是一张表，即 DLSw 接收到一个建立会话的请求时检查该表以查找所请求的资源。如果目的地址不在可达性缓存中，则 DLSw 会向它的对等体进行查询。例 13-29 是该命令的输出结果。

例 13-29 路由器 solo 上 show dlsw reachability 命令的执行示例

```
solo#show dlsw reachability
DLSw Local MAC address reachability cache list
Mac Addr      status      Loc.      port      rif
```



```
0000.613c.dc82  FOUND      LOCAL  TBridge-001  --no rif--
0006.3acf.7aa6  FOUND      LOCAL  TBridge-001  --no rif--

DLSw Remote MAC address reachability cache list
Mac Addr      status    Loc.    peer
0005.332e.2a25  FOUND      REMOTE  172.16.128.5(2065) max-1f(1500)

DLSw Local NetBIOS Name reachability cache list
NetBIOS Name  status    Loc.    port          rif
R2-D2         FOUND      LOCAL  TBridge-001  --no rif--

DLSw Remote NetBIOS Name reachability cache list
NetBIOS Name  status    Loc.    peer
LUKE          FOUND      REMOTE  172.16.128.5(2065) max-1f(1500)

solo#
```

字段 Status 描述本地对等体与该项的关系。字段 Location 则表明终端工作站是位于本地路由器还是远程路由器。字段 Status 的可能值包括：

- **FOUND**——路由器可以定位终端工作站。
- **NOT_FOUND**——终端工作站没有对本地路由器及其对等体的查询做出响应。
- **SEARCHING**——路由器正在发出查询信号以找到终端工作站。
- **UNCONFIRMED**——终端工作站是一个静态项，例如一个 DLSW 的 **ICANREACH** 项。
- **VERIFY**——缓存就要过期，路由器正在确认。

结果中其他一些信息还包括替换后的对等体/端口或工作站可到达的对等体的信息，以及 RIF 和最大数据帧大小。RIF 字段的 **no rif** 值说明工作站不支持 RIF，例如以太网工作站。

知道这些信息保存在缓存中很重要，可以对缓存中的项进行过期替换以及刷新缓存等。如果某个工作站正在或将要发送数据，那么这条命令的输出结果中应该列出这一工作站。

3. show dlsw circuits 命令

该命令可以显示使用 TCP 或帧中继直接封装方式，具有本地确认功能的对等体端对端会话情况，它显示的是本地 MAC 地址、远程地址以及所使用的服务接入点 (SAP)。例 13-30 列出了工作站 luke 和 R2-D2 之间的电路的情况。SAP 值是 (F0)，说明服务接入点是 NetBIOS。例子中的后半部分是电路的详细信息。

例 13-30 路由器 solo 上 show dlsw circuits 命令的执行示例

```
solo#show dlsw circuits
Index          local addr(lsap)  remote addr(dsap)  state          uptime
2919235595     0000.613c.dc82(F0) 0005.332e.2a25(F0) CONNECTED      00:01:17
Total number of circuits connected: 1

solo#show dlsw circuits detail
Index          local addr(lsap)  remote addr(dsap)  state          uptime
2919235595     0000.613c.dc82(F0) 0005.332e.2a25(F0) CONNECTED      00:01:28
PCEP: 49AC68   UCEP: 142AFC
Port:TB1       peer 172.16.128.5(2065)
Flow-Control-Tx CW:20, Permitted:39; Rx CW:20, Granted:29; Op: Repeat
Congestion: Low(02), Flow Op: Half: 0/0 Reset 0/0
```

```

RIF = --no rif--
Bytes:          2762/6467      Info-frames:      41/31
XID-frames:      0/0          UInfo-frames:      0/0
Total number of circuits connected: 1

solo#

```

电路存在两种状态，CONNECTED 或 CKT_ESTABLISHED。电路处于 CKT_ESTABLISHED 状态时，DLSw 已经正确建立了电路，但是终端工作站还没有，或不能够通过电路建立会话。重新启动终端工作站可能会解决这一问题。

4. show dlsw capabilities 命令

这条命令可以列出能力交换阶段本地对等体与其他对等体相互交换的控制向量的信息。其输出结果在 DLSw 对等体使用了 SAP 和 NetBIOS 过滤功能时非常有用。例 13-31 为该命令的输出情况。

例 13-31 路由器 solo 上 show dlsw capabilities 命令的输出结果

```

solo#show dlsw capabilities
DLSw: Capabilities for peer 172.16.128.5(2065)
 vendor id (OUI)       : '00C' (cisco)
 version number        : 2
 release number        : 0
 init pacing window    : 20
 unsupported saps       : none
 num of tcp sessions   : 1
 loop prevent support   : no
 icanreach mac-exclusive : no
 icanreach netbios-excl. : no
 reachable mac addresses : none
 reachable netbios names : none
 V2 multicast capable   : yes
 DLSw multicast address : none
 cisco version number   : 1
 peer group number      : 0
 peer cluster support   : no
 border peer capable    : no
 peer cost              : 3
 biu-segment configured : no
 UDP Unicast support    : yes
 Fast-switched HPR supp. : no
 NetBIOS Namecache length : 15
 local-ack configured   : yes
 priority configured    : no
 cisco RSVP support     : no
 configured ip address  : 172.16.128.5
 peer type              : conf
 version string         :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 12.1(2)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Tue 16-May-00 15:28 by ccai

```

5. debug dlsw peers 命令

上一节的图 13-30 和 13-31 与该 debug 命令结合使用非常有用，有助于定位数据流中的

这条命令能够提供对等体状态以及当前操作的综合信息。在某个对等体不能正常连接或无法保持在活动状态中时，这条 **debug** 命令会显得很有作用。例 13-32 列出了 NetBIOS 会话建立阶段 **debug dlsw peers** 命令的执行结果。

例 13-32 NetBIOS 会话建立阶段 debug dlsw peer 命令的执行结果

```
solo#debug dlsw peers
DLSw peer debugging is on
solo#
01:55:59: %SYS-5-CONFIG_I: Configured from console by console
01:55:59: %LINEPROTO-5-UPDOWN: Line protocol on Interface DLSw Port0, changed state
to up
01:56:00: DLSw: passive open 172.16.128.5(11000) -> 2065
01:56:00: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:TCP-RD PIPE OPENED
state:DISCONN
01:56:00: DLSw: dtp_action_c() opening write pipe for peer 172.16.128.5(2065)
01:56:00: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:DISCONN->WWR_RDOP

01:56:00: DLSw: Async Open Callback 172.16.128.5(2065) -> 11010
01:56:00: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:TCP-WR PIPE OPENED
01:56:00: DLSw: dtp_action_i() write pipe opened for peer 172.16.128.5(2065)
01:56:00: DLSw: CapExId Msg sent to peer 172.16.128.5(2065)
01:56:00: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:WWR_RDOP->WAIT_CAP

01:56:00: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:SSP-CAP MSG RCVD st
ate:WAIT_CAP
01:56:00: DLSw: dtp_action_j() cap msg rcvd from peer 172.16.128.5(2065)
01:56:00: DLSw: Recv CapExId Msg from peer 172.16.128.5(2065)
01:56:00: DLSw: received fhpr capex from peer 172.16.128.5(2065): support: false
, fst-prio: false
01:56:00: DLSw: Pos CapExResp sent to peer 172.16.128.5(2065)
01:56:00: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:WAIT_CAP->WAIT_CAP

01:56:00: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:SSP-CAP MSG RCVD st
ate:WAIT_CAP
01:56:00: DLSw: dtp_action_j() cap msg rcvd from peer 172.16.128.5(2065)
01:56:00: DLSw: Recv CapExPosRsp Msg from peer 172.16.128.5(2065)
01:56:00: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:WAIT_CAP->WAIT_CAP

01:56:00: DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP

01:56:00: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:SSP-CAP EXCHANGED s
tate:WAIT_CAP
01:56:00: DLSw: dtp_action_k() cap xchged for peer 172.16.128.5(2065)
01:56:00: DLSw: closing read pipe tcp connection for peer 172.16.128.5(2065)
01:56:00: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:WAIT_CAP->PCONN_WT

01:56:00: DLSw: Processing delayed event:TCP-PEER CONNECTED - prev state:PCONN_W
T
01:56:00: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:TCP-PEER CONNECTED
state:PCONN_WT
01:56:00: DLSw: dtp_action_m() peer connected for peer 172.16.128.5(2065)
01:56:00: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:PCONN_WT->CONNECT
01:56:31: DLSw: START-TPFSM (peer 172.16.128.5(2065)): event:DLX-KEEPAIVE REQ s
tate:CONNECT
01:56:31: DLSw: dtp_action_q() keepalive request from peer 172.16.128.5(2065)
01:56:31: DLSw: Keepalive Response sent to peer 172.16.128.5(2065)
01:56:31: DLSw: END-TPFSM (peer 172.16.128.5(2065)): state:CONNECT->CONNECT
```

6. debug dlswh reachability 命令

这条命令列出终端工作站的 MAC 地址以及相关的 SSAP 和 DSAP 的详细信息。这条 debug 命令还能提供该终端工作站正在发送的消息的类型。如果终端工作站不能出现在 DLSw 的可达性缓存中，这条 debug 命令就会发挥作用。例 13-33 为该命令的输出情况。

例 13-33 NetBIOS NAME_QUERY 阶段 debug dlswh reachability 命令的执行结果

```
solo#debug dlswh reachability
DLSw reachability debugging is on at event level for all protocol traffic
09:51:40: CSM: Received CLSI Msg : UDATA_STN.Ind dlen: 79 from DLSw Port0
09:51:40: CSM: smac 0000.613c.dc82, dmac c000.0000.0000, ssap F0, dsap F0
09:51:40: CSM: Received frame type NETBIOS NAME_QUERY from 0000.613c.dc82, DL0
09:51:40: CSM: Received CLSI Msg : CONECT_STN.Ind dlen: 47 from DLSw Port0
09:51:40: CSM: smac 0000.613c.dc82, dmac 0005.332e.2a25, ssap F0, dsap F0
09:51:43: CSM: Received CLSI Msg : UDATA_STN.Ind dlen: 86 from DLSw Port0
09:51:43: CSM: smac 0006.3acf.7aa6, dmac ffff.ffff.ffff, ssap AA, dsap AA
```

7. debug dlswh core 命令

这条命令能够显示 DLSw 代码中发生的所有情况。如果没有添加子命令，所有的重要记录都会启动。显然，该命令会产生大量的信息以帮助缩小某个问题或故障的起因的范围。在实际使用中，通常添加某些子命令来减少该命令输出结果的数量。

13.4.6 DLSw+的高级配置

DLSw 提供了很多特性以方便地配置对等体，控制探测帧以及备份和过滤能力信息。这一节提出了一些高级的 DLSw+特性配置，以下是此类特性：

- DLSw+混杂对等体的配置。
- DLSw+备份的配置。
- DLSw+的边界对等体，对等体组以及按需对等体。
- 利用令牌环列表，桥组列表以及端口列表对 DLSw 进行控制。
- DLSw+的动态对等体。
- 利用 icanreach 命令设置 DLSw+的可达性。

1. DLSw+混杂对等体的配置

DLSw+可以对不同类型的对等体进行配置。到目前为止，所配置的对等体类型都是静态的。即每一个需要连接的 DLSw 对等体都需要定义一个远程对等体与之相对应。在大型网络中，对等体很多，这种配置方式会变得非常麻烦。如果将本地对等体配置成混杂模式，那这个本地对等体就无需配置某个特定的远程对等体，它会自动地接受远程对等体的请求而建立远程连接。将某个本地对等体配置为混杂对等体的命令是：

```
Router (config) #dlswh local-peer peer-id ip_address promiscuous
```

图 13-34 的 DLSw+网络中，路由器 vader 将其本地对等体用作混杂对等体。这样就没有必要为到 skywalker、solo 和 chewbacca 连接进行 remote-peer 命令配置。但是，远程的路由器上仍然需要使用该 remote-peer 命令来指向该混杂对等体。

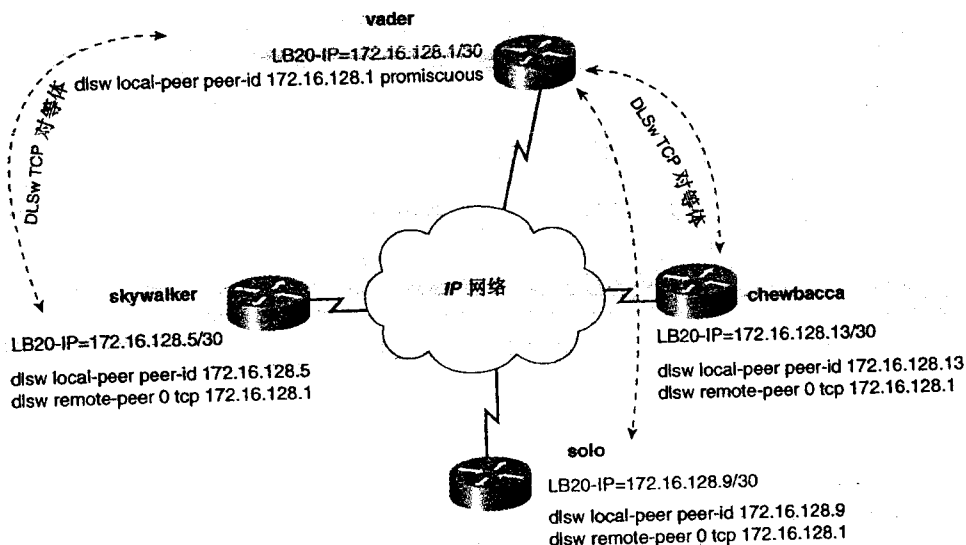


图 13-34 混杂对等体

混杂对等体对那些与之相连的所有远程对等体都有标准的默认值。例如，如果想要改变与混杂对等体相连的远程对等体的 `keepalive` 值或其他参数的话，就可以使用这条命令。而需要改变这些默认值或是使用访问列表时可以使用下面的这些命令：

```
Router (config) #dls prom-peer-defaults [bytes-netbios-out bytes-list-name] [cost
1-5]
[dest-mac destination_mac_address] [dmac-output-list access-list-number]
[host-netbios-out host-list-name] [keepalive seconds] [if largest_frame_516-11407]
[LSAP-output-list list] [tcp-queue-max size]
```

2. DLSw+的备份配置

DLSw 提供了两种方法进行冗余配置，区别在于是否保持备份 DLSw 对等体的活动状态。一种方法是将某对等体配置为备份对等体。对等体配置成为备份对等体之后，只有在路由器与主对等体或 DLSw 路由器连接出现问题时才会进入工作状态。另外一种方法主要用于链路失效时保持对等体的稳定性。当路由选择协议正在收敛，或是正在激活一个 DDR 或备份连接时，可利用这种方法调整 DLSw 的超时设置，以保持对等体处于工作状态之中，而不会由于超时而退出工作状态。

3. DLSw+的备份对等体

备份对等体的创建很简单，只需在新建的远程对等体上添加一个 `backup-peer` 参数即可。在创建备份对等体之前，必须先定义主对等体。备份对等体将指向另一个 DLSw 路由器而非主对等体路由器。关键字 `linger` 使得路由器在 X 秒的时间段内不断开与备份对等体的连接，以等待主对等体恢复工作。如果没有加上 `linger` 关键字，连接一恢复，主对等体立即就会进入工作状态。

`linger` 时间段过期之后，LLC2 会话会自动终止，这对备份对等体来说是不可取的。如果

通过备份连接建立新的电路。备份连接上的现有 LLC2 电路仍保持在活动状态。因此，如果不想终止备份对等体上的活动电路，就不要使用 **linger** 选项。如果 **linger** 的值为 0，备份对等体会保持工作，直到自己出现问题而停止工作，与主对等体的状态无关。选项 **linger** 的有用之处就是可以在连接出现抖动情况时，稳定 DLSw 对等体的工作。配置备份对等体的步骤以及所用的命令如下：

第 1 步 用下面这条命令配置主 DLSw 路由器的主对等体：

```
dlsw remote-peer 0 tcp primary_peers_ip_address
```

第 2 步 配置新加入的 DLSw 路由器的备份对等体：

```
dlsw remote-peer 0 tcp backup_peers_ip_address backup-peer  
primary_peers_ip_address linger timeout_in_minutes
```

图 13-35 中，路由器 solo 具有到 falcon 路由器的主对等体和到 skywalker 的备份对等体。路由器 falcon 和 vader 上的对等体配置成混杂对等体，因此，DLSw 路由器与这些路由器建立对等体连接无需 **remote-peer** 命令。路由器 solo 上还配置到 skywalker 的备份对等体。通过使用命令 **dlsw remote-peer 0 tcp 172.16.128.5 backup-peer 172.16.128.1 linger 5**，即使通往 falcon 的对等体发生故障，路由器 solo 也可以建立一个新的通往路由器 skywalker 的对等体。例 13-34 就是上图中路由器 solo 中 **show dlsw peer** 命令的输出结果。请注意，如果主对等体处于工作状态，则备份对等体处在断开状态。

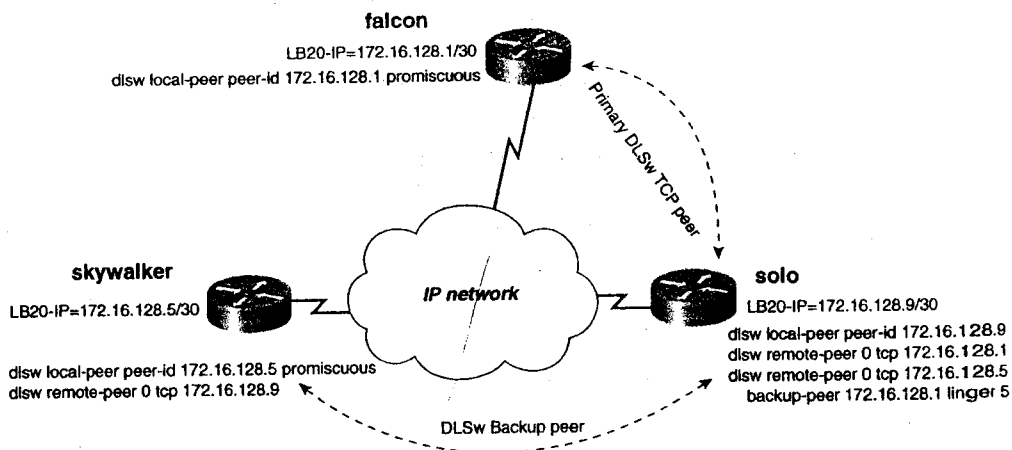


图 13-35 备份和混杂对等体

例 13-34 路由器 solo 上 **show dlsw peer** 命令显示备份对等体状态

```
solo#show dlsw peer
```

Peers:	state	pkts_rx	pkts_tx	type	drops	ckts	TCP	uptime
TCP 172.16.128.1	CONNECT	1268	190	conf	1	0	0	00:21:56
TCP 172.16.128.5	DISCONN	0	0	conf	0	0	.	.

Total number of connected peers: 1
Total number of connections: 1

主对等体出现故障之后，到 skywalker 的备份对等体进入工作状态，如例 13-35 所示。

例 13-35 路由器 solo 上 show dlsw peer 命令显示备份对等体状态

```
solo#show dlsw peer
Peers:
state      pkts_rx  pkts_tx  type  drops  ckts  TCP  uptime
TCP 172.16.128.5  CONNECT      2      4  conf      0      0  0 00:00:29
TCP 172.16.128.1  DISCONN      0      0  conf      0      0  -  -
Total number of connected peers: 1
Total number of connections: 1
solo#
```

路由器 solo 和 falcon 之间的主对等体恢复工作状态后，solo 等待 linger 计时器的时长超时（该例子中是 5 分钟），然后断开备份对等体，重建主对等体的连接。

4. DLSw+在 DDR 上的备份

另一种 DLSw 的备份方法是在链路失效时保持对等体连接。例如，如果采用 ISDN 连接作为备份，在 ISDN 电路进行呼叫建立连接时有可能需要使对等体保持在活动状态。这类连接失败后收敛所需的时间有可能超过 DLSw+ 的 keepalive 计时器的时间长度，从而使得对等体停止工作。DLSw+ 的 keepalive 工作在 TCP 端口 2065 上，这种情况下，数据和 keepalive 信息使用同一个端口，从而大大增加了利用访问控制列表 (ACL) 控制有效数据的难度。参数 **no keepalive** 也能避免这类数据引起的拨号呼叫。将 keepalive 设置为 0 后，DLSw+ 就无法以 keepalive 为基础来判断对等体是否还处于在工作状态之中，也不会因为丢失了 keepalive 而断开对等体。通过增加超时设置的值，如果在这一超时时间段里没有从某个对等体接收到数据，对等体就会自动断开连接。

要把 DLSw+ 配置成这样的工作模式，可以对 DLSw+ 的超时设置和 keepalive 进行控制，步骤如下：

第 1 步 在双方路由器的本地对等体上使用关键字 **keepalive 0**。

第 2 步 在两台路由器中为 **remote-peer** 命令加上一个 **timeout** 值。

图 13-36 中，路由器 solo 具有一个 DLSw+ TCP 端口以及一个通往路由器 falcon 的 ISDN 备份连接。

要想在 ISDN 收敛时保持对等体处于活动状态中，可以将 keepalive 设为 0 而超时值设为 5 分钟，即 300 秒。例 13-36 是 IP 连接暂时不存在而处于收敛过程中，而路由器 solo 到 falcon 之间的对等体依然保持活动状态的例子。

例 13-36 IP 连接不存在时，DLSw+ 对等体保持在工作状态中

```
solo#show dlsw peer
Peers:
state      pkts_rx  pkts_tx  type  drops  ckts  TCP  uptime
TCP 172.16.128.1  CONNECT      14      2  conf      0      0  0 00:01:37
Total number of connected peers: 1
Total number of connections: 1
```

```
solo#ping 172.16.128.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.128.1, timeout is 2 seconds:
```

```
.....
```

```
Success rate is 0 percent (0/5)
```

```
solo#
```

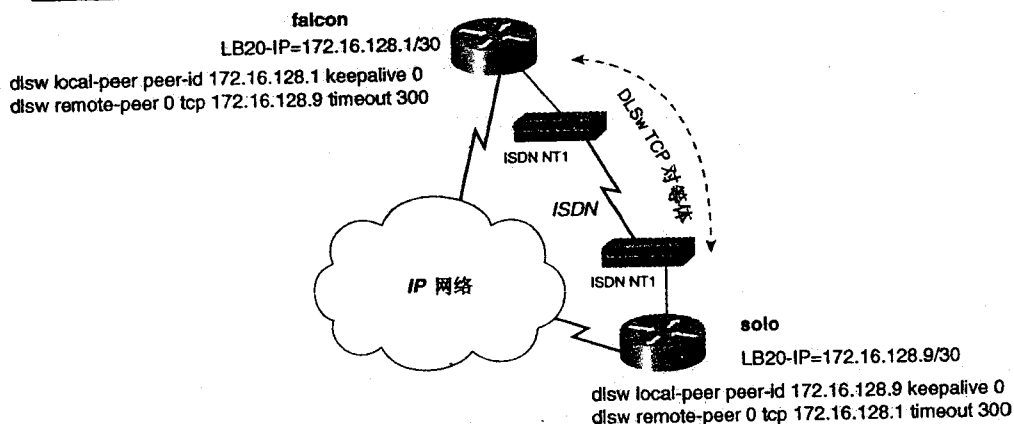


图 13-36 通过 DDR 的备份对等体

5. DLSw+的边界对等体，对等体组和按需对等体

对于那些需要能够任意连接，并且实现控制探测帧的 DLSw 网络而言，边界对等体和对等体组为之提供了一种行之有效的扩展方法。需要任意可达性的 DLSw 路由器要求与之相连的每一台路由器都使用 **remote-peer** 命令。图 13-37 就这类网络中一个常见的例子。

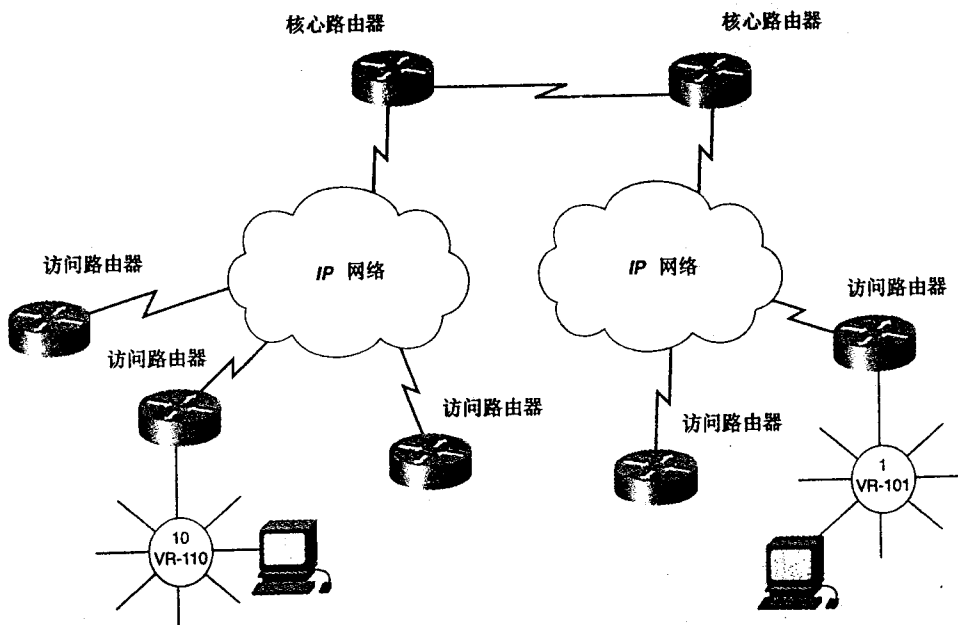


图 13-37 DLSw 的完全可达性

图中只给出了两台工作站，但是它们代表了所有的访问路由器所在 LAN 网段上工作站的情况。访问路由器上的工作站要到达任意其他访问路由器上的任意工作站，需要配置多个 **remote-peer** 命令。

图 13-38 列出了实现这种任意可达性的网络所需要的所有 **remote-peer** 命令的情况。

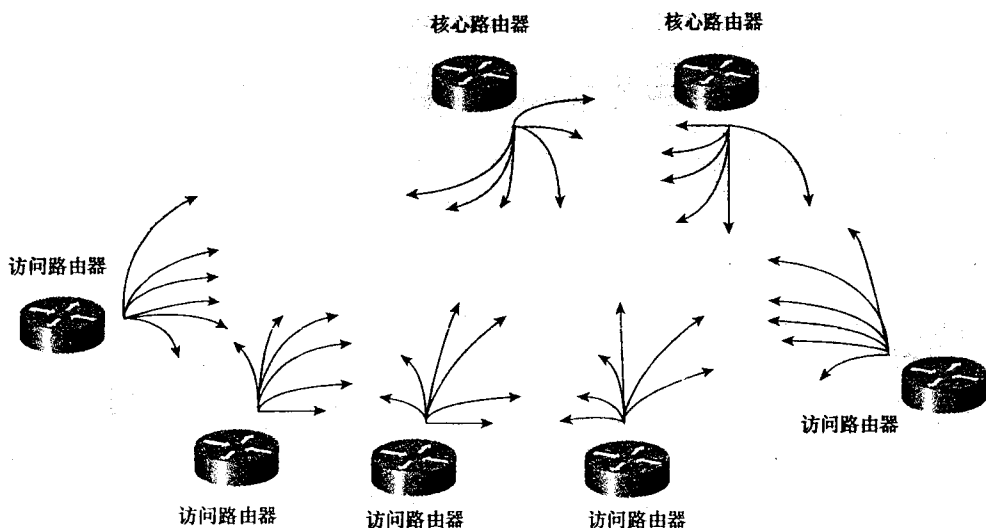


图 13-38 一个全连接网络所需的 TCP 对等体

为此，在这个小小的网络中就需要 42 条 **remote-peer** 命令的配置。所以，这种类型的网络很难进行扩展。此外，伴随着不同网络类型相关的配置信息是使从一个对等体传输到另一个对等体的探测数据帧的数量急剧增加，这要比某些错误配置的问题要严重的多。

Cisco DLSw+ 支持对等体组、边界对等体和按需对等体的概念。对等体组是一组路由器，其中一个或几个成员是指定为边界对等体。边界对等体的作用是为对等体组中的各成员路由器转发探测数据帧。接收到一个探测数据包时，边界对等体在将它转发到其他路由器之前，会先检查该数据包的本地缓存、远程缓存以及组缓存的内容。边界对等体对接收到的数据包在缓存中进行检查，如果在本地缓存中匹配成功，就不再将该探测数据包转发到其他路由器。远程缓存中保存本对等体组中的可达性信息，如果边界对等体的检查在该类型的缓存中匹配成功，就只把这一数据包转发到同一组中的路由器即可。组缓存中包含的则是该边界对等体所在组以外的对等体组的信息，如果边界对等体的检查在该类型的缓存中匹配成功，它就只会把探测数据包转发到边界对等体去。图 13-39 是划分成两个对等体组以后网络的情况。每个对等体组都必须含有一个边界路由器来正确处理该组探测数据包的转发工作。边界对等体之间也必须要有对等体连接。

下面是配置边界对等体和对等体组的步骤：

第 1 步 将网络划分成对等体组。

第 2 步 配置对等体组，建立该组中所有路由器到同一个对等体之间的对等体，这一个对等体路由器就是配置成边界对等体的那台路由器。配置对等体组可在

local-peer 命令中加上 **group x** 关键字，而要将指定路由器配置成边界对等体，可以在 **local-peer** 命令中加入 **border** 关键字。

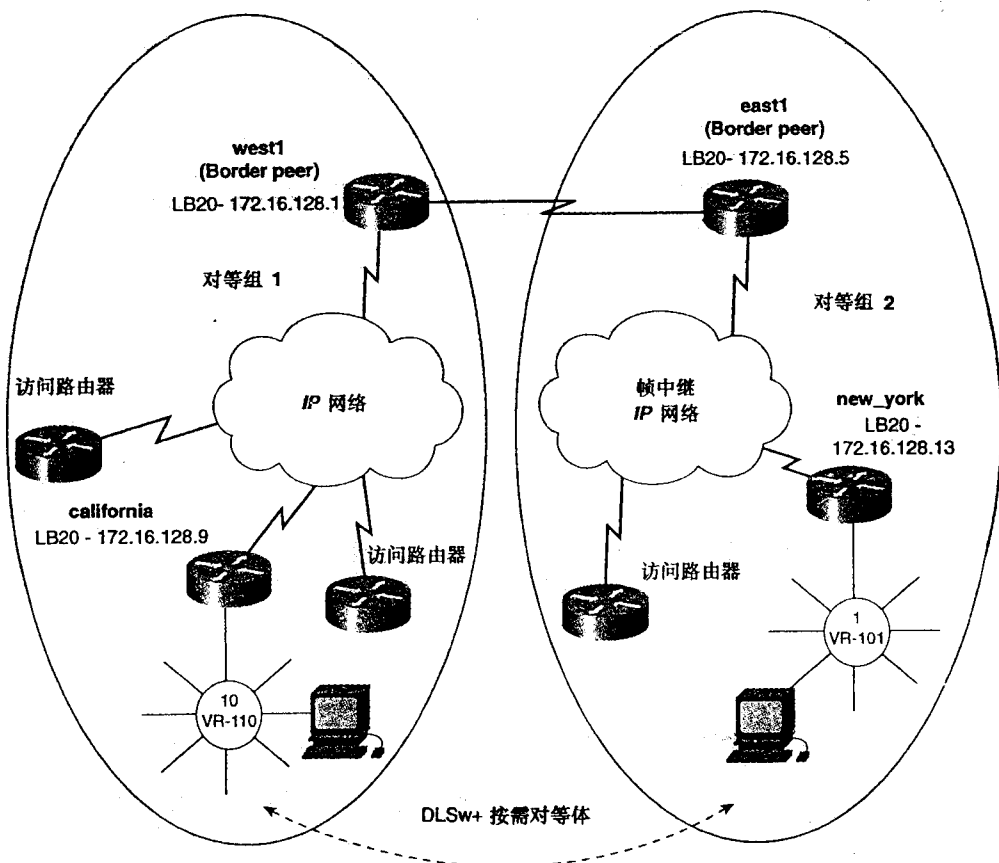


图 13-39 DLSw 的边界对等体和对等体组

第 3 步 在边界对等体之间配置一个 DLSw 对等体。

第 4 步 (可选) 配置 DLSw 的按需对等体。对任意可达性网络 (如 NetBIOS 或 APPN 网络)，必须配置按需对等体，它能在没有静态对等体的终端工作站之间创建按需对等体。按需对等体是在某个对等体组中的路由器向本组中或其他组里的路由器请求服务时建立的。配置需求对等体可以用这条命令：

```
dls w peer-on-demand-defaults [tcp 25-2000]
```

命令 **dls w peer-on-demand** 能够实现“按需”创建对等体，或者称之为按需对等体。不要把按需对等体和动态对等体相混淆，这是两种类型的对等体。下面的全局配置命令可以用来改变按需对等体的默认值：

```
Router (config) #dls w peer-on-demand-defaults [fst] [bytes-netbios-out
bytes-list-name]
[cost 1-5] [dest-mac destination_mac_address] [dmac-output-list access-list-number]
[host-netbios-out host-list-name] [keepalive seconds] [lf largest_frame_516-11407]
[lsap-output-list list] [port-list port-list-number] [priority] [tcp-queue-max size]
配置边界对等体和对等体组时，注意下面这些规则：
```

- 在一个组中，每个成员对等体必须和组内每个边界对等体建立对等关系。
- 组中所有的边界对等体之间也必须建立对等关系。
- 一个组中所有的边界对等体必须和其他组中的每一个边界对等体建立对等关系。
- 边界对等体会将探测帧数据包转发到它所在组中的所有成员对等体和所有其他边界对等体以及每个其他对等体组中的一个边界对等体。

图 13-39 为路由器分配了 IP 地址，把网络划分成两个对等体组。路由器 west 作为组 group 1 的边界对等体，而 east 则是 group 2 的边界对等体。例 13-37 是图 13-39 中路由器的配置示例。

例 13-37 边界对等体和对等体组的配置

```
Configuration of the west router:

dls w local-peer peer-id 172.16.128.1 group 1 border promiscuous
dls w remote-peer 0 tcp 172.16.128.5
!
!
Configurations of the routers in group 1, such as the california router:

dls w local-peer peer-id 172.16.128.9 group 1 promiscuous
dls w remote-peer 0 tcp 172.16.128.1
dls w peer-on-demand-defaults tcp-queue-max 50
!
Configuration of the east router:

dls w local-peer peer-id 172.16.128.5 group 2 border promiscuous
dls w remote-peer 0 tcp 172.16.128.1
!
!
Configurations of the routers in group 2, such as the new_york router:

dls w local-peer peer-id 172.16.128.13 group 2 promiscuous
dls w remote-peer 0 tcp 172.16.128.5
dls w peer-on-demand-defaults tcp-queue-max 50
!
```

6. 利用令牌环列表，桥组列表以及端口列表对 DLSw+探测帧进行控制

一台路由器上只能配置一个本地对等体，因此针对对等体之间探测数据包的控制就显得非常困难。通过令牌环列表可以指定哪些虚拟令牌环或哪些桥组可以接收某个特定远程对等体的探测数据包。

例如，图 13-40 中，路由器 yoda 有两个以太网接口，接口 E0 在桥组 1 中，接口 E2 在桥组 2 中。

对这个模型中的 DLSw+进行控制以便来自路由器 ben 的探测数据包只能转发到 E0 接口，或这是说桥组 1 中，而来自路由器 luke 的探测数据包则只能转发到路由器 yoda 上去。为了完成这样的配置，DLSw+必须要支持令牌环列表和端口列表。创建令牌环或端口列表，可以按下面的步骤进行：

- 第 1 步** 将令牌环和以太网桥区域分开。对以太网来说，把网桥区域分开是通过把以太网段放置到不同的桥组实现的。假设把 E0 划分到桥组 1 中，E1 划分到桥组 2 中。该例中，需要加入 **bridge 1 protocol ieee** 命令和 **bridge 2 protocol ieee** 命

令。在每一个桥组中，还要把该桥组加到 DLSw+ 中。因此，在上面的例子中还要用到这两条命令：**dls w bridge-group 1** 和 **dls w bridge-group 2**。

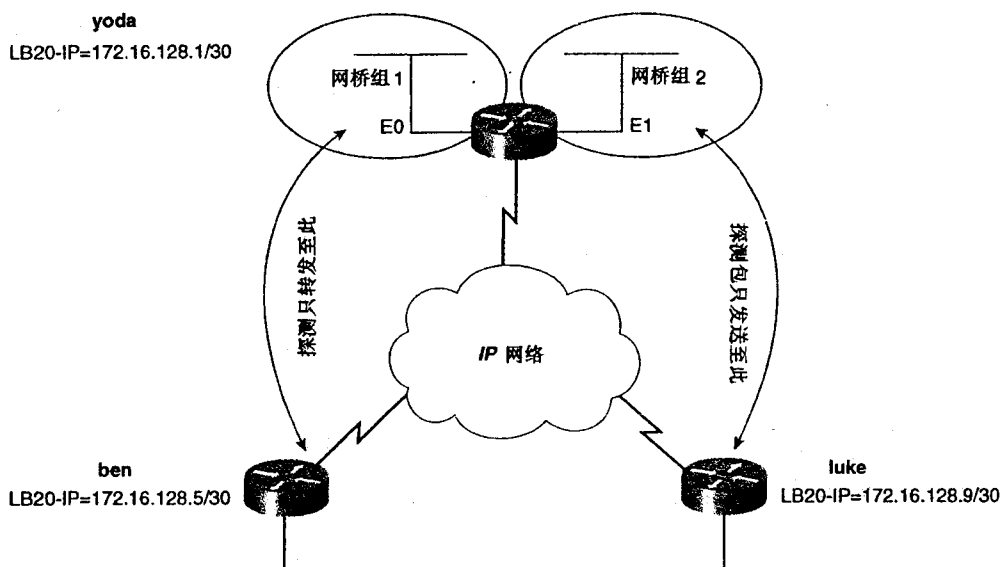


图 13-40 DLSw 桥组列表

对令牌环网络来说，需要先创建相互独立的虚拟令牌环，然后再把每个实际的令牌环接口通过源路由桥接指向这些虚拟的令牌环。比如，对于实际的令牌环 10，需要使用 **source-bridge 10 1 100** 命令通过源路由桥接到虚拟令牌环 100。另一个接口也要利用源路由桥接指向另一个虚拟令牌环。

第 2 步 创建令牌环列表。对令牌环网络来说，用下面的命令来创建令牌环列表：

Router (config) #dls w ring-list list_number_1-255 rings virtual-ring (s)

而对以太网则是使用这条命令来创建令牌环列表：

Router (config) #dls w bgroup-list list_number_1-255 bgrou ps bridge_group_number (s)

第 3 步 DLSw+ 的 remote-peer 命令中调用令牌环列表。例如，如果采用 **dls w bgroup-list 1 bgrou ps 1** 命令为以太网创建了令牌环列表，就可以用 **dls w remote-peer 1 tcp ip_address** 命令来调用令牌环表。

图 13-41 在路由器 yoda 上创建了两个桥组，每个桥组又加到了 DLSw bgroup 列表中。到路由器 ben 的对等体称为 bgroup list 1，而到 luke 的对等体则称为 bgroup list 2。

来自 luke 的探测数据包只会转发到路由器 yoda 上的 E1 接口。而来自 ben 的探测数据包却只会转发到 yoda 上的 E0 接口。例 13-38 是 yoda 上的配置情况。

例 13-38 路由器 yoda 的配置

```
hostname yoda
!
<<<text omitted>>>
!
```

(待续)

```

dlsw local-peer peer-id 172.16.128.1
dlsw bgroup-list 1 bgroups 1
dlsw bgroup-list 2 bgroups 2
dlsw remote-peer 1 tcp 172.16.128.5
dlsw remote-peer 2 tcp 172.16.128.9
dlsw bridge-group 1
dlsw bridge-group 2
!
interface Loopback20
 ip address 172.16.128.1 255.255.255.252
 no ip directed-broadcast
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 1
!
interface Ethernet1
 no ip address
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 2
!
<<<text omitted>>>
!
bridge 1 protocol ieee
bridge 2 protocol ieee

```

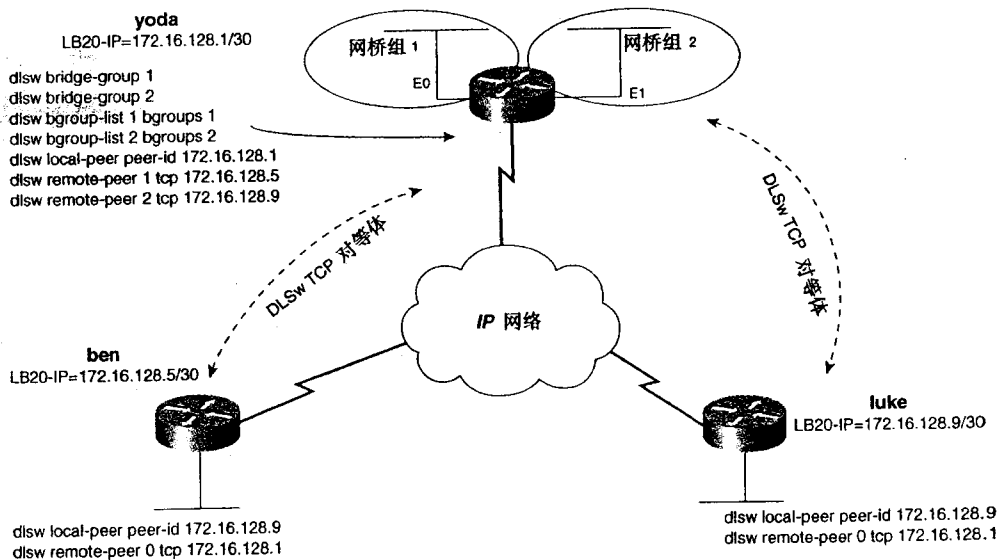


图 13-41 DLSw+桥组列表实例

端口列表可以用来将一个本地接口的流量（令牌环的或者是串行的）映射到远程对等体，命令如下：

```

Router (config) #dlsw port-list list_number_1-255 [token-ring | serial]
interface_number

```

端口列表的调用和令牌环或网桥列表一样，也使用 **remote-peer** 命令。

7. DLSw+的动态对等体

动态对等体又是一种类型的 DLSw 对等体，只在满足特定条件的情况下才能进入工作状态。这里的特定条件可以是 MAC 地址或 SAP 类型。动态对等体的配置可以使用 **remote-peer** 命令的 **dynamic** 关键字来进行，同时还应该对 **inactivity** 计时器加以设置。最近一次的电路与对等体断开连接之后，动态对等体还会再在工作状态中保持 10 分钟。创建好一个动态对等体之后，路由器会自动加入一个超时设置值，并且禁止 **keepalive** 的使用，原因和上面配置 DDR 的备份时的解释一样。

如果同时使用动态对等体和混杂对等体，记住一定要更改混杂对等体的默认值以禁止 **keepalive** 的使用，如前所述，这是用 **dls w prom-peer- defaults** 命令来完成的。

图 13-42 中，路由器 solo 配置了一个到 skywalker 的动态对等体。命令 **remote-peer** 指定对等体为动态的，只有当输出的 SAP 过滤表 201 的条件得到满足，它才会进入到工作状态。上一次的电路断开后的 5 分钟内该对等体还会保持在工作状态中。路由器会自动地在 **remote-peer** 命令中加入 **keepalive 0** 和 **timeout 90** 的选项。这个例子中的 SAP 过滤表是访问控制列表 201，只允许 SAP 0xF0F0（即 NetBIOS SAP）的使用。下一节将更详细地讨论 SAP 过滤表的问题。

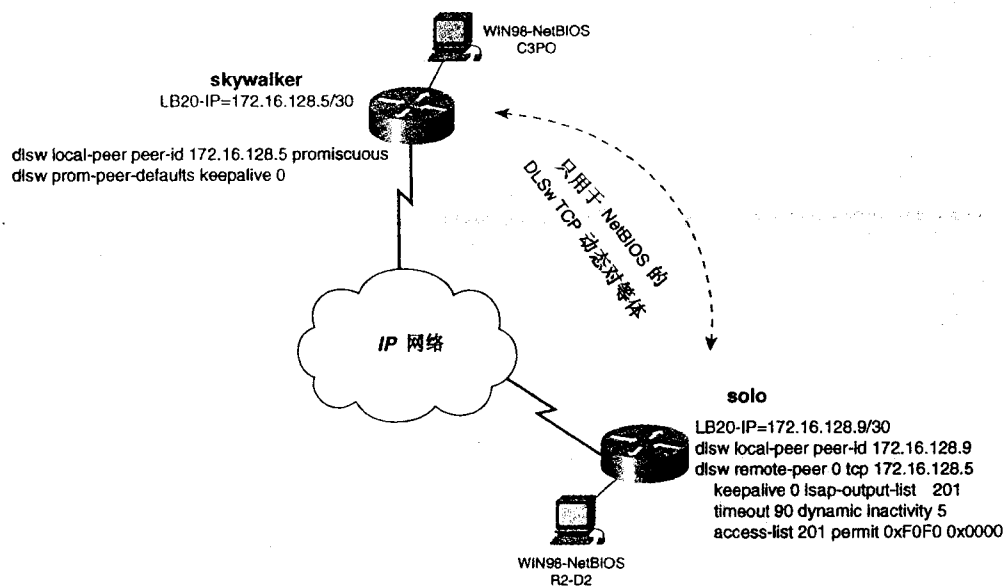


图 13-42 DLSw 的动态对等体

例 13-39 列出了路由器 solo 的配置情况，该路由器上已经配置了到 skywalker 的动态对等体。

例 13-39 路由器 solo 的配置

```

hostname solo
!
<<<text omitted>>>
!

```

```

dlsw local-peer peer-id 172.16.128.9
dlsw remote-peer 0 tcp 172.16.128.5 keepalive 0 lsap-output-list 201 timeout 90
dynamic inactivity 5
dlsw bridge-group 1
!
interface Loopback20
ip address 172.16.128.9 255.255.255.252
no ip directed-broadcast
!
interface Ethernet0
ip address 172.16.6.1 255.255.255.0
no ip directed-broadcast
bridge-group 1
!
<<<text omitted>>>
!
access-list 201 permit 0xF0F0 0x0000 - NETBIOS SAP
bridge 1 protocol ieee

```

注释 不要把动态对等体和混杂对等体相混淆。这一点比较困难，因为二者在某种程度上看都是动态的。

8. 用 icanreach 命令配置 DLSw+的可达性

在 DLSw+ 进行能力交换时，路由器之间还交换控制向量中可达的资源。这些都可以在路由器上静态配置。通过设置路由器可以到达的 SAP、MAC 地址以及 NetBIOS 名称，可以大大减少发送到远程对等体去的探测数据包数量。除了定义路由器可以到达的资源之外，还可以配置 SAP 值定义路由器不能到达的资源。如果使用 **icannotreach** 命令定义的录入项，路由器也会将此通知它的对等体。对等体会对它不能到达的资源加以跟踪，以避免向这些对等体发送探测数据包。下面的命令就可以用来配置 DLSw+ 的可达性：

```

Router (config) #dlsw icanreach [mac-address | saps | netbios-name]
Router (config) #dlsw icannotreach [saps] <0-FE> Even SAP Value (hex)

```

使用 **icanreach saps** 命令时要小心，因为它会拒绝所有其他服务接入点 (SAP)。换句话说，这条命令隐含“拒绝所有 SAP”的意思。所以，最好使用 **icannotreach** 命令来明确拒绝需要拒绝的 SAP。

用 **mac-exclusive** 参数和 **netbios-exclusive** 参数可以把可达性设置发送到某个地址或主机。这些参数必须和其他命令 (**dlsw icanreach mac address** 或 **netbios name**) 结合使用：

```

Router (config) #dlsw icanreach [mac-exclusive | netbios-exclusive]

```

图 13-43 中配置了路由器 falcon 上的 DLSw+ 可达性。falcon 宣告只能到达一个 MAC 地址：3745.1000.1010。例 13-40 是 falcon 上所需要的配置情况。

例 13-40 路由器 falcon 上的 DLSw+ 可达性配置示例

```

!
dlsw local-peer peer-id 172.16.128.5 promiscuous
dlsw icanreach mac-exclusive
dlsw icanreach mac-address 3745.1000.1010 mask ffff.ffff.ffff
dlsw bridge-group 1
!

```

通过查看路由器 solo 上 DLSw 的可达性，可以看到 falcon 惟一宣告的 MAC 地址，如例

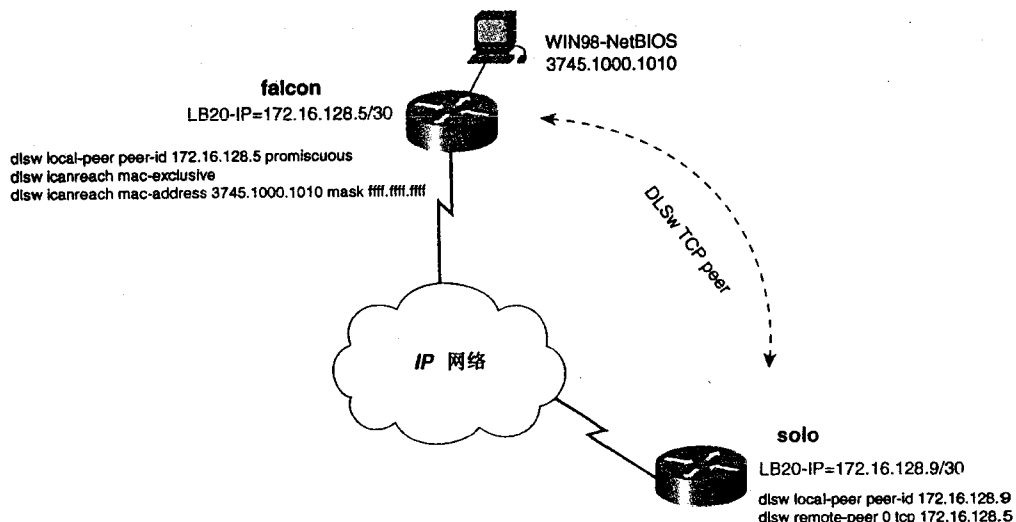


图 13-43 DLSw 的可达性

例 13-41 路由器 solo 上的 DLSw 可达性

```
solo#show dlsw reach
DLSw Local MAC address reachability cache list
Mac Addr      status    Loc.    port      rif
0000.613c.dc82 FOUND     LOCAL  TBridge-001 --no rif--
0006.3acf.7aa6 FOUND     LOCAL  TBridge-001 --no rif--
0007.781a.e7a9 FOUND     LOCAL  TBridge-001 --no rif--

DLSw Remote MAC address reachability cache list
Mac Addr      status    Loc.    peer
3745.1000.1010 UNCONFIRM REMOTE 172.16.128.5(2065)

DLSw Local NetBIOS Name reachability cache list
NetBIOS Name  status    Loc.    port      rif
R2-D2         FOUND     LOCAL  TBridge-001 --no rif--

DLSw Remote NetBIOS Name reachability cache list
NetBIOS Name  status    Loc.    peer
```

用 **show dlsw capabilities** 命令可以查看路由器现在能够到达的地址和 SAP 的详细情况。

例 13-42 列出了路由器 solo 的各种性能的情况。请注意，这里报告了一个 MAC 地址，而 max-exclusive 一栏设为了 yes。

例 13-42 路由器 solo 的 DLSw+性能

```
solo#show dlsw capabilities
DLSw: Capabilities for peer 172.16.128.5(2065)
vendor id (OUI)      : '00C' (cisco)
version number       : 2
release number       : 0
init pacing window   : 20
unsupported saps      : none
```



```

num of tcp sessions      : 1
loop prevent support     : no
icanreach mac exclusive : yes
icanreach netbios-excl. : no
reachable mac addresses  : 1745.1000.1010 mask 0fff.ffff.ffff
reachable netbios names  : none
V2 multicast capable    : yes
DLSw multicast address   : none
cisco version number     : 1
peer group number        : 0
peer cluster support     : no
border peer capable      : no
peer cost                : 3
biu-segment configured   : no
UDP Unicast support      : yes
Fast-switched HPR supp.  : no
NetBIOS Namecache length : 15
local-ack configured     : yes
priority configured      : no
cisco RSVP support       : no
configured ip address    : 172.16.128.5
peer type                 : prom
version string            :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 12.1(2)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Tue 16-May-00 15:28 by ccai
solo#

```

还可以利用 **dlsw reachability** 命令来对网络进行测试。用 **dlsw icanreach dummyname** 命令配置路由器，然后通过检查其对等路由器的性能状态判断所作的 DLSw 配置是否正确。

13.5 网桥环境下的数据过滤

本节讨论在网桥环境中进行数据过滤的方法。如果对访问列表不太熟悉，可以参考第 14 章“理解 IP 访问控制列表”。尽管该章只是关于 IP 访问列表，但是一些概念、规则和技巧适用于所有访问列表。

网桥和数据链路交换机的数据过滤都在数据链路层进行。Cisco 为数据链路层提供了 3 种主要过滤方式：

- 服务接入点 (SAP) 过滤。
- MAC 过滤。
- NetBIOS 名称过滤。

13.5.1 对服务接入点 (SAP) 的过滤

对以子网访问协议 (SNAP) 方式封装的数据帧，访问列表可以以数据帧的 DSAP/SAP/OUI 字段之后的 2 字节类型 (TYPE) 字段为基础进行过滤处理。对 IEEE 802.2 数据帧，访问列表可以以 DSAP/SSAP 字段为基础进行过滤操作。以 SAP 为基础的访问列表过滤所用到的命

令是:

```
Router (config) #access-list [ 200-299] [deny | permit] [ 0x0-0xFFFF] < 0x0-0xFFFF>
```

命令中的第 1 个值是协议类型代码，第 2 个值是协议类型代码的掩码。以协议类型或服务访问协议 (SAP) 为基础进行过滤的访问列表的参数 (协议类型) 取值范围是 200 到 299。访问列表的输入格式是 16 进制，而 16 进制的地址后面还要跟一个反掩码。反掩码用于地址，1 表示“忽略”，而 0 则表示“相关”。全 0 的掩码说明地址的所有位必须完全匹配，以便在访问列表进行比较时结果是“真” (TRUE)。可以参考第 14 章，了解访问列表以及通配符方面更多的内容。

1. SNA 的 SAP

SNA 使用多个 SAP。所幸，多个 SAP 可以通过 SAP 0x0D0D 进行过滤。

SNA 使用的 SAP 包括:

0x04 = IBM SNA 路径控制 (单个)

0x05 = IBM SNA 路径控制 (分组)

0x08 = IBM SNA 3270 终端

0x09 = IBM SNA

0x0c = IBM SNA 3270 终端

所有这 5 种 SAP 都可以通过“通配”SAP (0x0D0D) 进行过滤，它包括了所有的 SAP 类型。只允许 SNA SAP 的访问列表形式如下:

```
Router (config) #access-list 200 permit 0x0d0d 0x0000
```

2. NetBIOS 的 SAP

NetBIOS 数据使用的 SAP 值如下:

0xf0 = IBM NetBIOS 命令

0xf1 = IBM NetBIOS 响应

这两个 SAP 对应的 16 进制的掩码是 0xf0f0 和 0x0101。只允许 NetBIOS SAP 的访问列表:

```
Router (config) #access-list 200 permit 0xf0f0 0x0101
```

仅过滤 NetBIOS 命令就足以控制所有的 NetBIOS 数据:

```
Router (config) #access-list 200 permit 0xf0f0 0x0d0d
```

3. IPX 的 SAP

使用 802.2 封装形式的 IPX 所用的 SAP 值是:

0xe0 = Novell NetWare

只允许 IPX SAP 的访问列表如下所示:

```
Router (config) #access-list 200 permit 0xe0e0 0x0000
```

或者简单地:

```
Router (config) #access-list 200 permit 0xe0e0
```

4. 过滤及阻塞所有 SAP

包括 200 系列在内的所有访问列表在所列的清单上最后都有一条隐含的 **deny any**。即使在显示列表内容时，该内容也不会显示出来。

3 章 配

和增强数

路交换 ()

Router (config) #
相反，拒绝所有 SA

access-list
访问列

deny 0

0x0fff:

Router (config) #

access-list

deny 0x0

0xffff

警告 将 IPX 网络数据进行桥接。即 DLSw 令。但是，一台路由器“SNA/DLSw 对等体”不会承载任何 IPX 数据。器具有可以接收 IPX 数所有的 DLSw 对等体。它们需要对整个 IPX 域议在 SNA/DLSw 对等体的过滤措施。

DLSw 网
传输 IP
仅配置了
”。如果
为该路由
接口，也
， IPX 网
桥接。幸
器上运

时，要千
器包，除
路由，以
也在该
的骨干
没有配
对 IPX
会停止
一问题
者是在

心。默
DLSw
为 SNA
的骨干
X 路由
路由，1
串行
很容易
适当的

下，C
器上配
中的主
行，“
果 SNA
把这些
会中
解决，
IOS 软

由器会
px rou
w 对等
对等体
w 对等
通过网
一切都
是允许
使用 IP

命令
或
器”
由
到
为
协
AP

13.5.2 MAC 址的过

以 700 开始的访问列
Router (config) #
MAC_address_mas

access-list
799] [

MAC 址

过滤。7

列的过

下面

：

一般来说，该访问列
含性，除非自己指定。t
么所用的命令可以是：

输入时只
如果要
一条访

一个 M
地址，主
页只允

因为 I
C 地址

地址不
5cf3.5c

包
那

：

access-list 701 p

0060.5

da4

13.5.3 NetBI 名称的 滤

要进行基于名称的
Router (config) #
例如，过滤名为 HA

IOS 过
os acce
的 Net

可以采
st host
，可用

的命令：
ss_list

[deny

mit] ;

rn

netbios access-li

ost deny

deny

为：

下面是使用 NetBIO

访问列表

意的问题

：

- 访问列表区分
- 访问列表中的
- 和 ADD_ NAM
- 0a 和 0e (DAT
- 名称字段进行

号。多数
占名称在
(QUERY)
AM, N

BIOS 名
IOS 命
与源名
_QUER

是大写。
和 01
段进行
NAME

是 GRO
而在
OGNIZ

AME_C
OS 命
则是

Y
的

13.5.4 实例：桥环 的过

访问列表的应用方法。多。上述 滤方式 用于物理 1、DLS 对等体或 源

路由桥接。上面的内容中已经列出了这些过滤方式应用于源路由桥接或 DLSw 对等体所需的命令句法。

图 13-44 给出了一个含有路由器 solo 和 chewbacca 的 DLSw 网络。现在要采用 LSAP 过滤使两台路由器之间的 DLSw 连接上只有 SNA SAP 能通过。

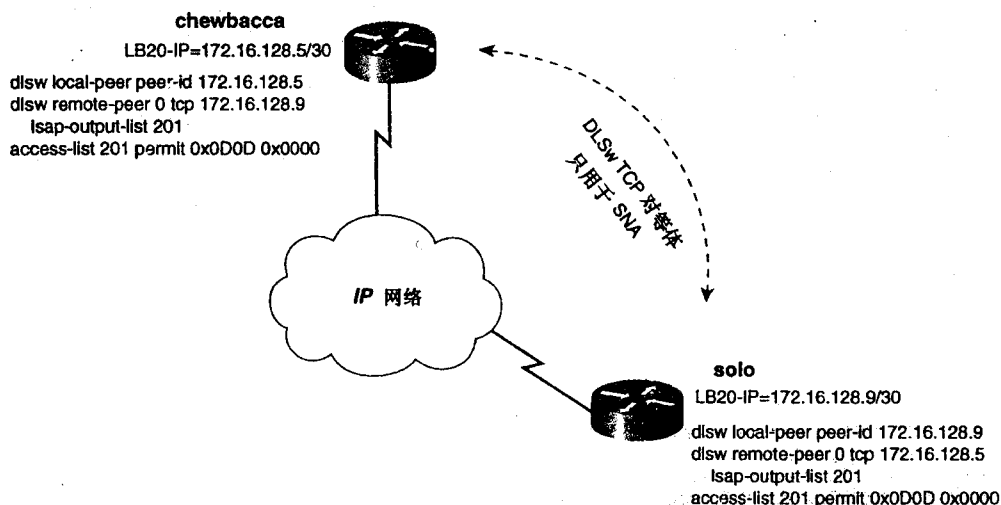


图 13-44 DLSw 的 LSAP 过滤

要做到这一点，需要配置一个访问列表，许可 SAP 0x0d0d 通过，如下所示：

```
access-list 201 permit 0x0d0d
```

再用 lsap-output-list 参数将过滤关联到每台路由器的 remote-peer 命令中。

13.6 实验 26：透明桥接、远程源路由桥接 (RSRB) 和 LSAP 过滤——第 1 部分

13.6.1 实验说明

不可路由的协议（如 NetBIOS, SNA 等）仍然活跃在许多现代网络中。在网络中传输这类协议的方法也有很多。这个实验就是让大家可以练习透明桥接、RSRB、设置根以及 LSAP 过滤。

13.6.2 实验内容

假定在全美国范围内（从加利福尼亚的峡谷到北威斯康星的山脉）活跃着一个犯罪团伙的秘密网络。参与者的具体地点是保密的，路由器相互都是通过代号来识别的，像 trashman, bumbelly 等。这些犯罪团伙者们试图将他们现有的网络连接起来以利用超级计算机，本人概不负责。

H.O.O.V.E.R.。他们拥有的超级计算机 (H.O.O.V.E.R.和 H.O.O.V.E.R.2) 都是 SNA 大型机，因此远程站点必须通过桥接才能与总部建立连接。我们的任务就是按照下面这些要求配置一个桥接和路由的网络：

- 按照图 13-45 配置一个 IP 网络，路由选择协议采用 EIGRP，自治系统 ID 为 2001。
- 按照图 13-45 配置帧中继网络，不要在这个实验中使用 DLSw。
- 配置网络使得路由器 lone_rhino 和 trashman 可以将 SNA 传输到 HQ 站点或路由器 wolf。
- 对路由器 beerbelly 进行配置，使它可以通过 SNA 协议访问大型机 H.O.O.V.E.R.2。大型机 H.O.O.V.E.R.2 需要 RIF 信息，确保我们配置的网络对此提供支持。
- 禁止 NetBIOS 数据从路由器 beerbelly 上的 Ring 2 传输到路由器 wolf 上的 Ring 1 去。
- 将路由器 wolf 配置为透明桥接域的根。
- 在路由器 beerbelly 上的令牌环接口上存在着一台“双重机密”的工作站。由于其绝对机密性，我们需要专门为它配置一个静态 RIF，该 RIF 配置路径为：秘密工作站的 MAC 地址 = 0101.0027.0081；RING2-BRIDGE9-RING50-BRIDGE5-RING52-BRIDGE13-RING7。

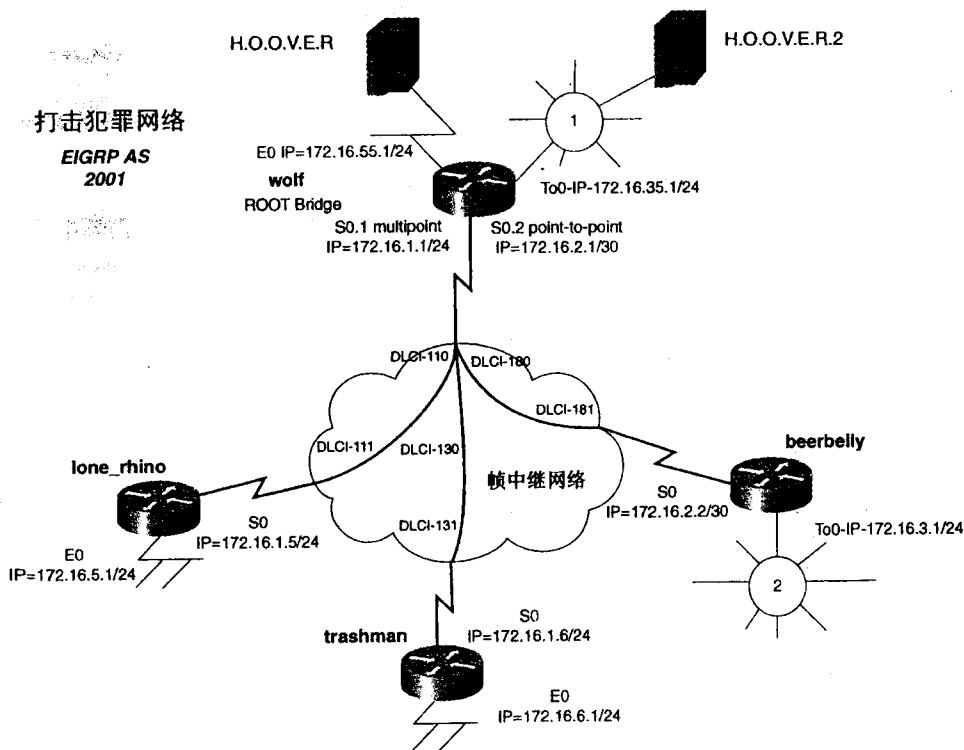


图 13-45 犯罪团伙网拓扑

13.6.3 实验目的

- 按照图 13-45 对这个犯罪团伙网络及其相应的 IP 进行配置，路由选择协议采用 EIGRP，自治系统 ID 为 2001。

- 在 WAN 上将帧中继作为数据链路协议，在路由器 wolf, lone_rhino 和 trashman 之间配置帧中继多点网络，而 wolf 和 beerbelly 之间则是帧中继点对点网络。
- 在路由器 lone_rhino、trashman 和 wolf 的以太网接口上配置透明桥接，在帧中继网络上将这些路由器通过网桥进行连接。
- 在路由器 beerbelly 和 wolf 的令牌环接口之间配置一个远程源路由桥接。
- 对源路由桥接进行配置，使得远程源路由网桥不会转发 NetBIOS 数据。
- 在路由器 beerbelly 上配置一个静态 RIF，其信息为：MAC =0101.0027.0081；RING2-BRIDGE9-RING50-BRIDGE5-RING52-BRIDGE13-RING7。

13.6.4 所需设备

- 5 台 Cisco 路由器，其中 4 台通过 V.35 背对背线缆或类似方式与一台帧中继交换机相连。
- 用集线器或交换机构成 5 个 LAN 网段。一台路由器需要具有一个令牌环接口和一个以太网接口，还要有一台路由器也需要具有令牌环接口。

13.6.5 物理设计与实验准备

- 按照图 13-45 将集线器以及串行线缆与路由器相连。
- 还需要有一台帧中继交换机具有 3 条 PVC。例 13-43 就是该实验所需帧中继交换机的配置示例。

例 13-43 配置帧中继交换机

```
hostname frame_switch
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay intf-type dce
 frame-relay route 110 interface Serial0 111
 frame-relay route 130 interface Serial3 131
 frame-relay route 180 interface Serial5 181
!
<<<text omitted>>>
!
interface Serial3
 no ip address
 encapsulation frame-relay
```

```

clockrate 64000
frame-relay intf-type dce
frame-relay route 131 interface Serial1 130
!
<<<text omitted>>>
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 181 interface Serial1 180

```

13.7 实验 26：透明桥接、远程源路由桥接 (RSRB) 和 LSAP 过滤——第 2 部分

13.7.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 4 台路由器以背对背的方式连接在一起。然后按照图 13-45，利用交换机或集线器/MAU 创建 3 个以太网 LAN 网段和 2 个令牌环 LAN 网段。

物理连接完成之后，再按照图 13-45 为所有的 LAN 和 WAN 接口分配 IP 地址。在继续下一步操作之前，一定要对每一台路由器的本地 LAN 和 WAN 接口都用 ping 命令进行测试。在路由器 wolf 与 lone_rhino 和 trashman 之间配置一个帧中继多点网络，而在路由器 wolf 和 beerbelly 之间则配置帧中继点对点网络。网络中的路由选择协议采用 EIGRP。要把路由器 trashman 的子网转发到 lone_rhino 去，我们需要在 wolf 路由器上屏蔽掉 EIGRP 的水平分割功能。路由器 wolf、lone_rhino、trashman 和 beerbelly 的 EIGRP 和帧中继的相关配置部分如例 13-44 所示。关于配置的细节问题，可以参考第 5 章“WAN 协议与技术：帧中继”以及第 11 章“混合协议：增强型内部网关路由选择协议 (EIGRP)”。

例 13-44 配置路由器 wolf、lone_rhino 和 beerbelly 的帧中继与 EIGRP

```

hostname wolf
!
<<<text omitted>>>
!
interface Serial0
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
logging event subif-link-status
logging event dlci-status-change
frame-relay lmi-type cisco
!

```

(待续)

```

interface Serial0.1 multipoint
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
no ip split-horizon eigrp 2001
frame-relay map ip 172.16.1.5 131 broadcast
frame-relay map ip 172.16.1.6 131 broadcast

interface Serial0.2 point-to-point
ip address 172.16.2.1 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 300

<<<text omitted>>>

router eigrp 2001
passive-interface Ethernet0
network 172.16.0.0
no auto-summary

hostname lone_rhino

interface Serial0
ip address 172.16.1.5 255.255.255.0
encapsulation frame-relay
frame-relay map ip 172.16.1.6 131 broadcast
frame-relay map ip 172.16.1.1 131 broadcast

<<<text omitted>>>

router eigrp 2001
network 172.16.0.0
no auto-summary

hostname trashman

interface Serial0
ip address 172.16.1.6 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay map ip 172.16.1.5 131 broadcast
frame-relay map ip 172.16.1.1 131 broadcast
frame-relay lmi-type cisco

<<<text omitted>>>

router eigrp 2001
network 172.16.0.0
no auto-summary

hostname beerbelly

interface Serial0
ip address 172.16.2.2 255.255.255.0

```



```

encapsulation frame-relay
frame-relay interface-dlci 181
frame-relay lmi-type cisco
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!

```

帧中继网络配置完毕和 IP 连接建立之后，就可以开始配置桥接的网络环境了。

首先是配置路由器 wolf、lone_rhino 和 trashman 组成的以太网段之间的透明桥接，同时将生成树的根设置成路由器 wolf。要达到这些目标，应按照下列 3 个步骤进行：

第 1 步 设置网桥号以及相应的生成树。

第 2 步 将相应接口配置为该网桥组的成员。

第 3 步 设置根网桥。

第 1 步的工作可以利用路由器命令 **bridge-group 1 protocol ieee** 来完成，这条命令能够在那些需要配置透明桥接的路由器上创建所需的网桥组。第 2 步为创建的网桥组分配物理或逻辑接口，这通过接口命令 **bridge-group 1** 实现。在帧中继多点接口，如路由器 wolf 的 S0.1 接口以及 lone_rhino 和 trashman 的 s0 接口，还需要利用 **frame-relay map bridge** 命令进行配置。例 13-45 就是在路由器 wolf 上进行第 1 和第 2 步操作的过程。

例 13-45 配置路由器 wolf 上的透明桥接

```

wolf(config)#bridge 1 protocol ieee
wolf(config)#interface ethernet 0
wolf(config-if)#bridge-group 1
wolf(config)#interface serial 0.1
wolf(config-subif)#bridge-group 1
wolf(config-subif)#frame-relay map bridge 110 broadcast
wolf(config-subif)#frame-relay map bridge 130 broadcast
wolf(config-subif)#

```

例 13-46 则是路由器 lone_rhino 上的透明桥接的配置过程。

例 13-46 配置路由器 lone_rhino 上的透明桥接

```

lone_rhino(config)#bridge 1 protocol ieee
lone_rhino(config)#interface e0
lone_rhino(config-if)#bridge-group 1
lone_rhino(config-if)#exit
lone_rhino(config)#interface s0
lone_rhino(config-if)#bridge-group 1
lone_rhino(config-if)#frame-relay map bridge 111 broadcast

```

路由器 trashman 上透明桥接的配置和 lone_rhino 基本一样，trashman 上完整的 **frame-relay map** 命令是 **frame-relay map bridge 131 broadcast**。此时，透明桥接应开始工作。通过 **show bridge** 命令可以确定网桥的工作状态，如例 13-47 所示。

例 13-47 查看透明网桥的工作状态

```
trashman#show bridge

Total of 300 station blocks, 295 free
Codes: P - permanent, S - self

Bridge Group 1:

      Address      Action  Interface  Age  RX count  TX count
0060.5cf3.5e65    forward Ethernet0    0     44        0
0050.5475.e1ad    forward Serial0      0     10        0
0000.8108.caae    forward Serial0      0     20        0
0000.863c.3b41    forward Serial0      3      2        0
00e0.b05a.66e4    forward Serial0      3      1        0

trashman#
```

这时网桥应该显示 MAC 地址，并通过串行接口和以太网接口进行转发。如果没看到这些信息，先检查一下帧中继和以太网接口是否是在同一个网桥组中。同时还应该检查帧中继中配置过 **frame-relay map** 命令。

第 3 步需要将路由器 wolf 配置成生成树的根。可能与我们所期望的不同，生成树的根可能是 trashman。可以使用 **show spanning-tree** 命令来查看当前根路由器。例 13-48 是在路由器 trashman 上执行这一条命令的结果。从中可见，trashman 是 STP 的当前根。

例 13-48 查看路由器 trashman 上的 STP 信息

```
trashman#show spanning-tree

Bridge group 1 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0060.5cf3.5da4
Configured hello time 2, max age 20, forward delay 15
We are the root of the spanning tree
Port Number size is 9
Topology change flag not set, detected flag not set
Times: hold 1, topology change 35, notification 2
      hello 2, max age 20, forward delay 15
Timers: hello 1, topology change 0, notification 0
bridge aging time 300

Port 2 (Ethernet0) of Bridge group 1 is forwarding
Port path cost 100, Port priority 128
Designated root has priority 32768, address 0060.5cf3.5da4
Designated bridge has priority 32768, address 0060.5cf3.5da4
Designated port is 2, path cost 0
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 0, received 0

Port 6 (Serial0 Frame Relay) of Bridge group 1 is forwarding
Port path cost 647, Port priority 128
Designated root has priority 32768, address 0060.5cf3.5da4
Designated bridge has priority 32768, address 0060.5cf3.5da4
Designated port is 6, path cost 0
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 0, received 0

trashman#
```

要将网桥的根改到路由器 wolf，可以在 wolf 上执行下面这条全局命令：

```
wolf (config) #bridge 1 priority 100
```

用这条命令配置之后，再看一下路由器 trashman 上的 STP 情况，如例 13-49 所示，可见，当前网桥的根已经改为了 wolf，而它的优先级也改为了 100。

例 13-49 查看路由器 trashman 上的 STP 信息

```
trashman#show spanning-tree
Bridge group 1 is executing the IEEE compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0060.5cf3.5da4
Configured hello time 2, max age 20, forward delay 15
Current root has priority 100, address 00e0.1e58.e792
Root port is 6 (Serial0), cost of root path is 647
Port Number size is 9
Topology change flag not set, detected flag not set
Times: hold 1, topology change 35, notification 2
      hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0
bridge aging time 300

Port 2 (Ethernet0) of Bridge group 1 is forwarding
Port path cost 100, Port priority 128
Designated root has priority 100, address 00e0.1e58.e792
Designated bridge has priority 32768, address 0060.5cf3.5da4
Designated port is 2, path cost 647
Timers: message age 0, forward delay 0, hold 0
BPDU: sent 0, received 0

Port 6 (Serial0 Frame Relay) of Bridge group 1 is forwarding
Port path cost 647, Port priority 128
Designated root has priority 100, address 00e0.1e58.e792
Designated bridge has priority 100, address 00e0.1e58.e792
Designated port is 12, path cost 0
Timers: message age 2, forward delay 0, hold 0
BPDU: sent 0, received 57

trashman#
```

实验的下一步就是对路由器 beerbelly 进行配置，以使得它的令牌环网络可以通过 SNA 协议访问路由器 wolf 令牌环网络上的大型机 H.O.O.V.E.R.2。SNA 的应用需要 RIF，因此也必须对此加以配置。要想在 WAN 上传输 SNA，需要在这里采用 RSRB，这主要还是由于该实验中不能使用 DLSw+。

RSRB 的配置包括下面这 4 个步骤：

第 1 步 如果需要，先通过路由器接口命令 **multiring all** 启动 RIF。

第 2 步 利用 **source-bridge ring-group virtual_ring** 命令启动虚拟令牌环。

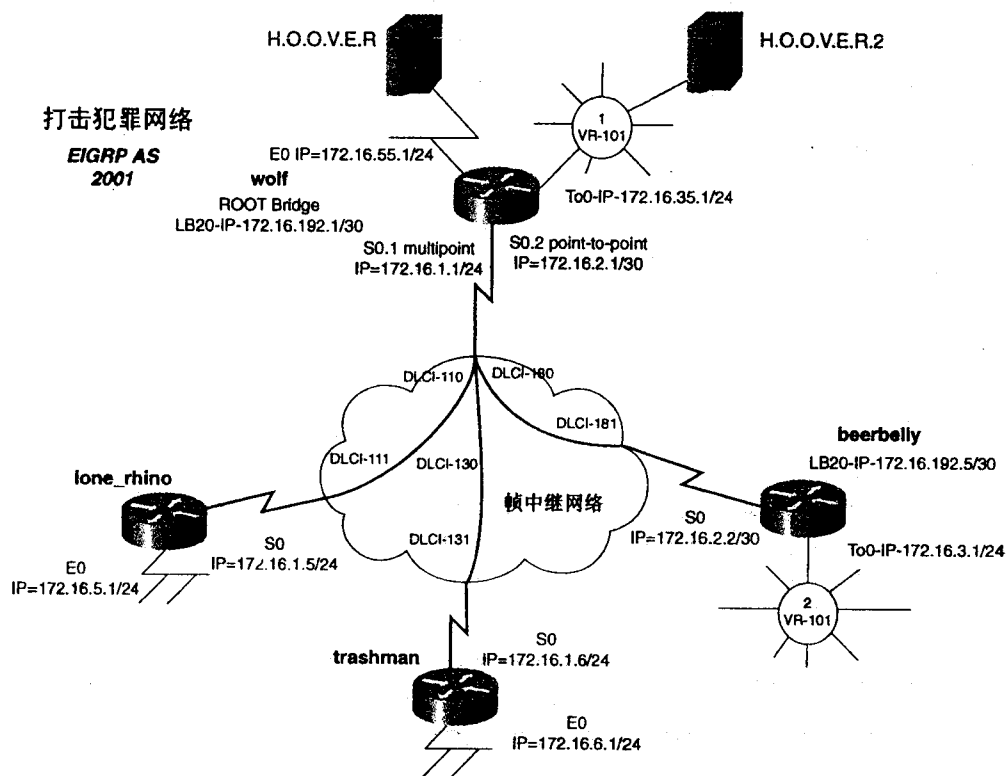
第 3 步 配置从物理令牌环到虚拟令牌环的 SRB。

第 4 步 确定所要使用的封装类型，并对 RSRB 进行配置。本实验中采用的封装类型是 TCP。

以 TCP 为封装类型为每个对等体路由器以及本地路由器创建远程对等体时需要的命令是：

```
source-bridge remote-peer virtual_ring tcp ip_address [lf largest_frame_size]
[ local_ack]
```

RSRB 对等体配置好环路地址。标出了环路地址/接口以及虚拟令牌环 101 的网络图请参见图 13-46。



在这个实验中，大型机 H.O.O.V.E.R.2 需要 RIF 字段。因此首先要利用接口命令 **multiring all** 在所有的令牌环接口上启动 RIF。第 2 步是虚拟令牌环的配置。这里打算使用的是虚拟令牌环 101，这是在路由器 wolf 和 beerbelly 上用全局命令 **source-bridge ring-group 101** 来实现的。第 3 步是在需要加入到 RSRB 组中去的令牌环接口上配置 SRB。在路由器 beerbelly 上配置 SRB 的命令如下：

```
beerbelly (config-if) #source-bridge 2 1 101
```

而路由器 wolf 上的 **source-bridge** 命令为：

```
wolf (config-if) #source-bridge 1 1 101
```

例 13-50 是路由器 beerbelly 上源路由网桥（SRB）的配置情况。

例 13-50 到目前为止路由器 beerbelly 的配置示例

```
hostname beerbelly
!
<<<text omitted>>>
!
source-bridge ring-group 101 -virtual ring
```

(待续)

```
!
interface Loopback0
  ip address 172.16.192.5 255.255.255.252
!
<<<text omitted>>>
!
interface TokenRing0
  ip address 172.16.3.1 255.255.255.0
  ring-speed 16
  multi-ring all
  source-bridge 2 1 101
  source-bridge 2 1 101
```

第 4 步是对 RSRB 对等体以及传输类型进行配置。这个模型中用于传输 RSRB 的是 TCP。这样需要在每本地路由器上配置一个 RSRB TCP 对等体指向它们自己，此外还需要一个 RSRB TCP 对等体指向远程路由器——确切地说，是另一台路由器的环路地址。例 13-51 就是路由器 beerbelly 和 wolf 上所需的 RSRB 配置示例。路由器 beerbelly 上的 **remote-peer** 命令应该和 wolf 上是完全一样的。这里需要加以注意的就是 RSRB 配置中还需要为本地路由器创建一个远程对等体。

例 13-51 路由器 beerbelly 的相关配置

```
!
source-bridge ring-group 101
source-bridge remote-peer 101 tcp 172.16.192.5
source-bridge remote-peer 101 tcp 172.16.192.1
!
```

到现在为止，RSRB 已经开始工作了，可以用 **show source-bridge** 命令查看 RSRB 的状态。例 13-52 是 wolf 上 RSRB 状态的情况。如果 RSRB 检测到了数据，就会进入“open”状态。

例 13-52 在路由器 wolf 上查看 RSRB 的状态

```
wolf#show source-bridge

Local Interfaces:
      srn bn  trn r p s n  max hops  cnt      cnt      drops
To0      1  1  101 *  b   7  7  7      40       0       0

Global RSRB Parameters:
  TCP Queue Length maximum: 100

Ring Group 101:
  This TCP peer: 172.16.192.1
  Maximum output TCP queue length, per peer: 100
Peers:
  state  bg lv  pkts_rx  pkts_tx  expl_gn  drops TCP
TCP 172.16.192.1  -      3      0      0      0      0  0
TCP 172.16.192.5  open    3      0      4      2      0  0
Rings:
  bn: 1  rn: 1  local  ma: 4007.781a.e789 TokenRing0      fwd: 0
  bn: 1  rn: 2  remote ma: 4000.30b1.270a TCP 172.16.192.5  fwd: 0

Explorers: ----- input ----- ----- output -----
```

	spanning	all-rings	total	spanning	all-rings	total
To0	0	40	40	0	0	0
Explorer fastswitching enabled						
Local switched: 40		flushed 0		max Bps 38400		
	rings	inputs	bursts	throttles	output drops	
To0	40		0	0	0	
wolf#						

RSRB 进入工作状态之后，还打算对其进行过滤。在这个实验中，希望禁止 RSRB 传输 NetBIOS。要在 RSRB 上过滤掉 NetBIOS，需要配置 SAP 过滤，过滤掉 SAP 0xf0。由于所有的 SAP 都有一条隐含的 deny，因而还必须在 SAP 中加入一行覆盖这条 deny 的作用。然后是用 **rsrb remote-peer lsap-output-list** 命令将 SAP 过滤器应用到 RSRB 中去。例 13-53 就是在路由器 wolf 上的相关配置部分。

例 13-53 在 RSRB 上进行 SAP 过滤

```
rsrb remote-peer 101 tcp 172.16.192.5 lsap-output-list 201 filter to peer
172.16.192.5
1
access-list 201 deny 0xf0f0 0x0000          ← Deny NETBIOS
access-list 201 permit 0x0000 0xffff         ← Permit all SAPs
1
```

实验的最后部分是在路由器 beerbelly 上配置一个静态 RIF。需要配置的这个静态 RIF 是：

MAC = 0101.0027.0081; RING2-BRIDGE9-RING50-BRIDGE5-RING52-BRIDGE13-RING7

本章中讨论过，静态 RIF 的创建是从左到右进行的。这个静态 RIF 中的第一个字节是 0a30。

从左到右的头两个位是 00，它将探测帧类型设置成了一个特定路由探测帧，使用这种类型的探测帧主要是因为这里的 RIF 是一个静态 RIF。第 3 位是 0，是保留位。接下来的 5 位是设置 RIF 的字节长度，这里是设置成 10 个字节的，也就是 0x0a。下一位 (D 位，也就是方向位) 是 0，它表明 RIF 是从左往右读，也就是正向的。下面的 3 位是 011，它把帧尺寸设成了 4472，也就是 Cisco 的最大尺寸。最后 4 位都是保留位。

RD 字段，也就是后面的 4 个字节，也很容易分解。这 4 个字节 (0029, 0325, 0034d 和 0070) 就是 16 位的 RD 字段。每个字节的前 3 位就是十六进制的令牌环号。最后一位也是十六进制的令牌环号。以这个例子中的 RIF 为例：

```
RING2 to BRIDGE9 = 0029
RING50 to BRIDGE5 = 0325
RING52 to BRIDGE13 = 034d
RING7 to BRIDGE0 = 0070
```

网桥号为 0 说明 SRB 到此就终止了 RIF，不需要经过其他的网桥。

例 13-54 就是路由器 beerbelly 上的静态 RIF 的配置情况，后面还运行了一条 **show rif** 命令。

例 13-54 静态 RIF 的配置与查询

```
beerbelly#conf t
Enter configuration commands, one per line. End with CNTL/Z.
beerbelly(config)#rif 0101.0027.0081 0a30.0029.0325.034d.0070 to0
beerbelly(config)#exit
```

```
beerbelly#show rif
Codes: * interface, - static, + remote

Dst HW Addr   Src HW Addr   How   Idle(min) Vlan Routing Information Field
0101.0027.0081 N/A      To0    0A30.0020.0325.034D.0070
0000.30b1.270a N/A      To0    *
beerbelly#
```

例 13-55 示这个实验的完整配置清单。

例 13-55 最终配置清单

```
hostname wolf
!
source-bridge ring-group 101
source-bridge remote-peer 101 tcp 172.16.192.1
source-bridge remote-peer 101 tcp 172.16.192.5
rsrb remote-peer 101 tcp 172.16.192.5 lsap-output-list 201
!
interface Loopback20
 ip address 172.16.192.1 255.255.255.252
 no ip directed-broadcast
!
interface Ethernet0
 ip address 172.16.55.1 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 1
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 logging event subif-link-status
 logging event dlci-status-change
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 no ip split-horizon eigrp 2001
 frame-relay map bridge 130 broadcast
 frame-relay map bridge 110 broadcast
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 130 broadcast
 bridge-group 1
!
interface Serial0.2 point-to-point
 ip address 172.16.2.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 180
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.35.1 255.255.255.0
 no ip directed-broadcast
```

```

ring-speed 16
multiring all
source-bridge 1 1 101
!
router eigrp 2001
  passive-interface Ethernet0
  network 172.16.0.0
  no auto-summary
!
<<<text omitted>>>
!
access-list 201 deny   0xF0F0 0x0000
access-list 201 permit 0x0000 0xFFFF
!
bridge 1 protocol ieee
bridge 1 priority 100

```

```

hostname lone_rhino
!
<<<text omitted>>>
!
interface Ethernet0
  ip address 172.16.5.1 255.255.255.0
  bridge-group 1
!
interface Serial0
  ip address 172.16.1.5 255.255.255.0
  encapsulation frame-relay
  frame-relay map bridge 111 broadcast
  frame-relay map ip 172.16.1.6 111 broadcast
  frame-relay map ip 172.16.1.1 111 broadcast
  bridge-group 1
!
<<<text omitted>>>
!
router eigrp 2001
  network 172.16.0.0
  no auto-summary
!
<<<text omitted>>>
!
bridge 1 protocol ieee

```

```

hostname trashman
!
<<<text omitted>>>
!
interface Ethernet0
  ip address 172.16.6.1 255.255.255.0
  no ip directed-broadcast
  bridge-group 1
!
interface Serial0
  ip address 172.16.1.6 255.255.255.0
  no ip directed-broadcast
  encapsulation frame-relay
  no ip mroute-cache
  frame-relay map bridge 131 broadcast
  frame-relay map ip 172.16.1.5 131 broadcast
  frame-relay map ip 172.16.1.1 131 broadcast
  frame-relay lmi-type cisco
  bridge-group 1

```



```

!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!
<<<text omitted>>>
!
bridge 1 protocol isee

hostname beerbelly
!
!
rif 0101.0027.0081 0A30.0029.0325.034D.0070 TokenRing0
!
<<<text omitted>>>
!
source-bridge ring-group 101
source-bridge remote-peer 101 tcp 172.16.192.5
source-bridge remote-peer 101 tcp 172.16.192.1
rsrb remote-peer 101 tcp 172.16.192.1 lsap-output-list 201
!
interface Loopback20
 ip address 172.16.192.5 255.255.255.252
!
interface Serial0
 ip address 172.16.2.2 255.255.255.0
 encapsulation frame-relay
 frame-relay interface-dlci 101
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.3.1 255.255.255.0
 ring-speed 16
 multiring all
 source-bridge 2 1 101
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!
<<<text omitted>>>
!
access-list 201 deny 0xF0F0 0x0000
access-list 201 permit 0x0000 0xFFFF

```

13.8 实验 27：DLSw+的 TCP、LLC2、混杂、动态

以及备份对等体的配置——第 1 部分

13.8.1 实验说明

置参数可以用来对数据加以控制并提供先进的备份功能。在这个实验中，大家就可以练习多种类型的 DLSw+ 对等体的配置，包括 DLSw+ 的 TCP 对等体、LLC2 对等体、混杂对等体、备份对等体以及动态对等体，此外大家还有机会利用 SAP 过滤和端口列表对数据加以控制。

13.8.2 实验内容

我们这里处理的还是那个犯罪团伙的网络。现在他们利用透明桥接和远程源路由桥接 (RSRB) 在帧中继网络上传输 SNA 和 NetBIOS。他们仍然拥有两台超级计算机，H.O.O.V.E.R. 和 H.O.O.V.E.R.2，各地的犯罪团伙希望通过 DLSw 访问这些超级计算机。我们现在的任务就是严格按照下面这些要求对 DLSw+ 进行配置：

- 按照图 13-47 对 IP 网络进行配置，以 EIGRP 作为路由选择协议，自治系统 ID 是 2001。
- 按照图 13-47 配置帧中继网络。

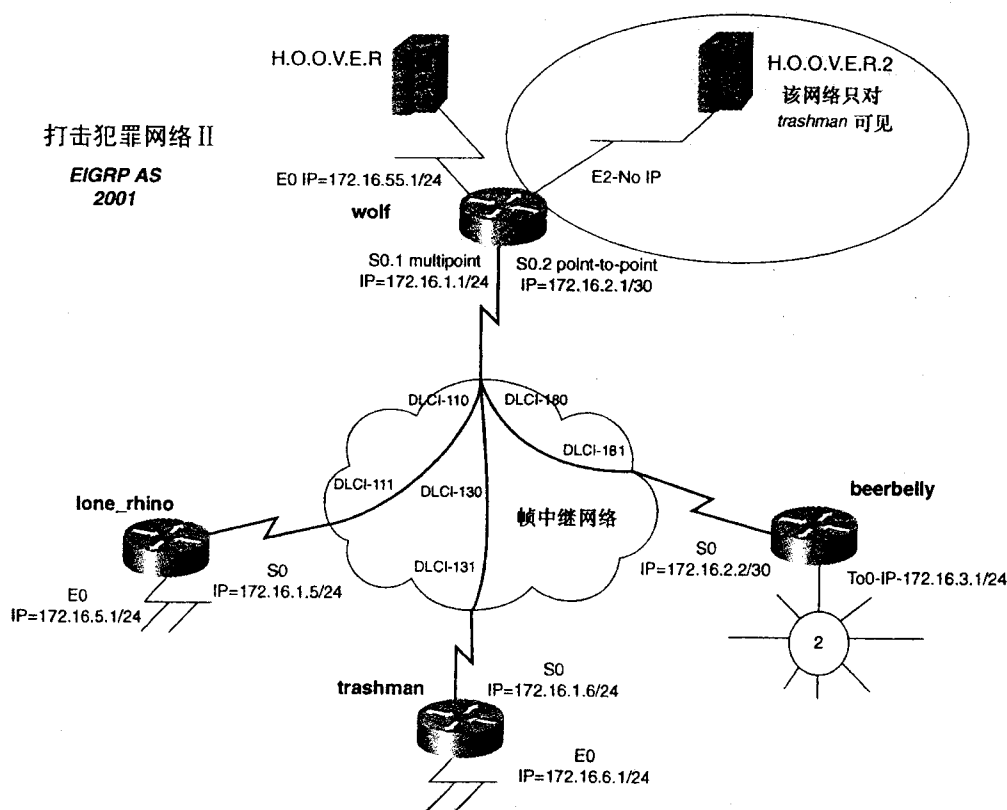
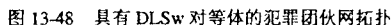


图 13-47 犯罪团伙网络拓扑，第 2 部分

- 按照图 13-48 和下面这些要求对网络中的 DLSw+ 对等体进行配置：

在路由器 wolf 的 Ethernet 2 网段和 trashman 的 Ethernet 网段之间配置 DLSw+。这

打击犯罪网络 II
EIGRP AS
2001



- 子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责。

13.8.3 实验目的

- 按照图 13-47 和 13-48 配置这个犯罪团伙网络及其相应的 IP，以 EIGRP 为路由选择协议，自治系统号为 2001。由于该实验是从上一个实验演变而来的，因此可以在这里沿用上个实验开始的配置。
- 在 WAN 上采用帧中继作为数据链路协议。在路由器 wolf、lone_rhino 和 trashman 之间配置帧中继多点网络，而路由器 wolf 和 beerbelly 之间则是一个帧中继点对点网络。
- 按照图 13-48 在网络上配置 DLSw+ 对等体。具体要求如下：
 - 在路由器 wolf 的 Ethernet 2 网段和 trashman 的 Ethernet 网段之间配置 DLSw+。这是在路由器 wolf 上惟一需要用 **remote-peer** 进行配置的地方。
 - 对 DLSw+ 进行配置，使得只有路由器 trashman 才对大型机 H.O.O.V.E.R.2 所在的以太网段具有可见性和可访问性。其他任何 DLSw+ 对等体都不能看到和访问 H.O.O.V.E.R.2 所在的 Ethernet 网段。
 - 配置一个从路由器 lone_rhino 到 wolf 的动态 SNA TCP 对等体。若 7 分钟之内没有数据，则该对等体就会超时撤销，只有 SNA 才能激活此动态对等体进入工作状态。该对等体只在路由器 lone_rhino 的以太网段和 wolf 的 E0 网段之间传输 SNA，也就是大型机 H.O.O.V.E.R 所在网段。
 - 配置一个 TCP 对等体来将数据从路由器 beerbelly 的令牌环网段传输到大型机 H.O.O.V.E.R 所在的 wolf 路由器的以太网段上去。
 - 在路由器 trashman 和 lone_rhino 之间配置一个备份对等体，用其对 trashman 和 wolf 之间的对等体进行备份。只要有一条电路连接还在工作之中，备份对等体就应该保持在工作状态。主对等体进入工作状态的时候，要确保备份对等体上的活动电路不会断开。

13.8.4 所需设备

- 5 台 Cisco 路由器，其中 4 台通过 V.35 背对背线缆或类似方式与一台帧中继交换机相连。
- 用集线器或交换机创建 5 个 LAN 网段。一台路由器还需要具有 2 个以太网接口，另一台路由器需要具有令牌环接口。

13.8.5 物理设计与实验准备

注意：前面一个实验的帧中继交换机的配置在这里完全适用。

- 按照图 13-47 将集线器以及串行线缆与路由器相连。
- 还需要有一台帧中继交换机具有 3 条 PVC。例 13-56 就是该实验中需要的帧中继交换机的配置情况。

例 13-56 配置帧中继交换机

```

hostname frame_switch
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148000
frame-relay intf-type dce
frame-relay route 111 interface Serial1 110
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay intf-type dce
frame-relay route 110 interface Serial0 111
frame-relay route 130 interface Serial3 131
frame-relay route 180 interface Serial5 181
!
<<<text omitted>>>
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 131 interface Serial1 130
!
<<<text omitted>>>
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 181 interface Serial1 180

```

13.9 实验 27: DLSW+ 的 TCP、LLC2、混杂、动态以及备份对等体的配置——第 2 部分

13.9.1 实验步骤

该实验是上一个实验的继续，物理设计上惟一的区别就是在路由器 wolf 上，这里的路由器 wolf 具有两个以太接口而没有令牌环接口。如果沿用上一实验的同样配置，记得要在 WAN 和 RSRB 上关闭透明桥接。

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 4 台路由器以背对背的方式连接在一起。然后按照图 13-47 利用交换机或集线器/MAU 创建 4 个以太 LAN 网段和令牌环 LAN 网段。

物理连接完成之后，按照图 13-47 为所有的 LAN 和 WAN 接口分配 IP 地址。在路由器 wolf, lone_rhino 和 trashman 上配置

置一个帧中继点对点网络，路由选择协议采用 EIGRP。要想将路由器 trashman 的子网转发到路由器 lone_rhino，必须在路由器 wolf 上屏蔽掉 EIGRP 的水平分割功能。例 13-57 是路由器 wolf, lone_rhino, trashman 和 beerbelly 的 EIGRP 和帧中继配置情况。

例 13-57 配置路由器 wolf, lone_rhino 和 beerbelly 的帧中继与 EIGRP

```
hostname wolf
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 logging event subif-link-status
 logging event dli-status-change
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 no ip split-horizon eigrp 2001          ← Split horizon disabled
 frame-relay map ip 172.16.1.5 110 broadcast ← Map statement to lone_rhino
 frame-relay map ip 172.16.1.6 130 broadcast ← Map statement to trashman
!
interface Serial0.2 point-to-point
 ip address 172.16.2.1 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 160          ← Inverse ARP
!
<<<text omitted>>>
!
router eigrp 2001                        ← Routing EIGRP
 passive-interface Ethernet0
 network 172.16.0.0
 no auto-summary
!

-----

hostname lone_rhino
!
<<<text omitted>>>
!
interface Serial0
 ip address 172.16.1.5 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.1.6 111 broadcast ← Map statement to trashman
 frame-relay map ip 172.16.1.1 111 broadcast ← Map statement to wolf
!
<<<text omitted>>>
!
router eigrp 2001                        ← Routing EIGRP
 network 172.16.0.0
 no auto-summary
!

-----

hostname trashman
!
<<<text omitted>>>
!
```

```

interface Serial0
 ip address 172.16.1.6 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 172.16.1.5 131 broadcast <-Map statement to lone_rhino
 frame-relay map ip 172.16.1.1 131 broadcast <-Map statement to wolf
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
router eigrp 2001 <-Routing EIGRP
 network 172.16.0.0
 no auto-summary
!

hostname beerbelly
!
<<<text omitted>>>
!
interface Serial0
 ip address 172.16.2.2 255.255.255.0
 encapsulation frame-relay
 frame-relay interface-dlci 181
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!

```

DLSw+的配置过程简单总结如下：

- 第1步 为对等体配置环路地址。
- 第2步 对本地对等体进行配置。
- 第3步 对 SRB 或透明桥接进行配置。
- 第4步 对远程对等体进行配置。

图 13-49 给出了网络的配置情况，突出显示了一些更为具体的配置细节问题，如网桥组、虚拟令牌环以及本地、远程对等体所需的环路地址。

第1步就是按照图 13-49 分配环路接口。由于 EIGRP 处在与前面配置相同的主掩码位范围之内，所以它能够宣告这些环路地址。如果所有的环路接口都可以通过 ping 测试，就可以进行下一步的工作。

第2步为路由器分配本地对等体。本实验中通过全局路由器命令 **dlsw local-peer peer-id loopback_IP_address** 来实现。按照要求，在路由器 wolf 上只允许配置一个远程对等体。因此，wolf 的本地对等体必须配置成混杂对等体。如果还想节省一点配置远程对等体的时间，我们还可以把路由器 lone_rhino 的本地对等体配置成混杂对等体。在 wolf 上配置本地对等体的命令是：

```
wolf (config) #dlsw local-peer peer-id 172.16.192.1 promiscuous
```

第3步在需要加入到 DLSw 域的路由器和接口上配置透明或源路由桥接。在路由器 lone_rhino, trashman 和 beerbelly 上，这个过程是和上一个实验完全相同。因此，可以不必再花时间去配置这部分。但是，路由器 wolf 需要配置两个网桥组。Ethernet 0 需要在网桥

组 1 中，而 Ethernet 2 则是在网桥组 2 中。这样需要设置一份 DLSw 网桥列表用于远程对等体的配置。例 13-58 是路由器 wolf 上透明桥接和 DLSw 部分的配置。

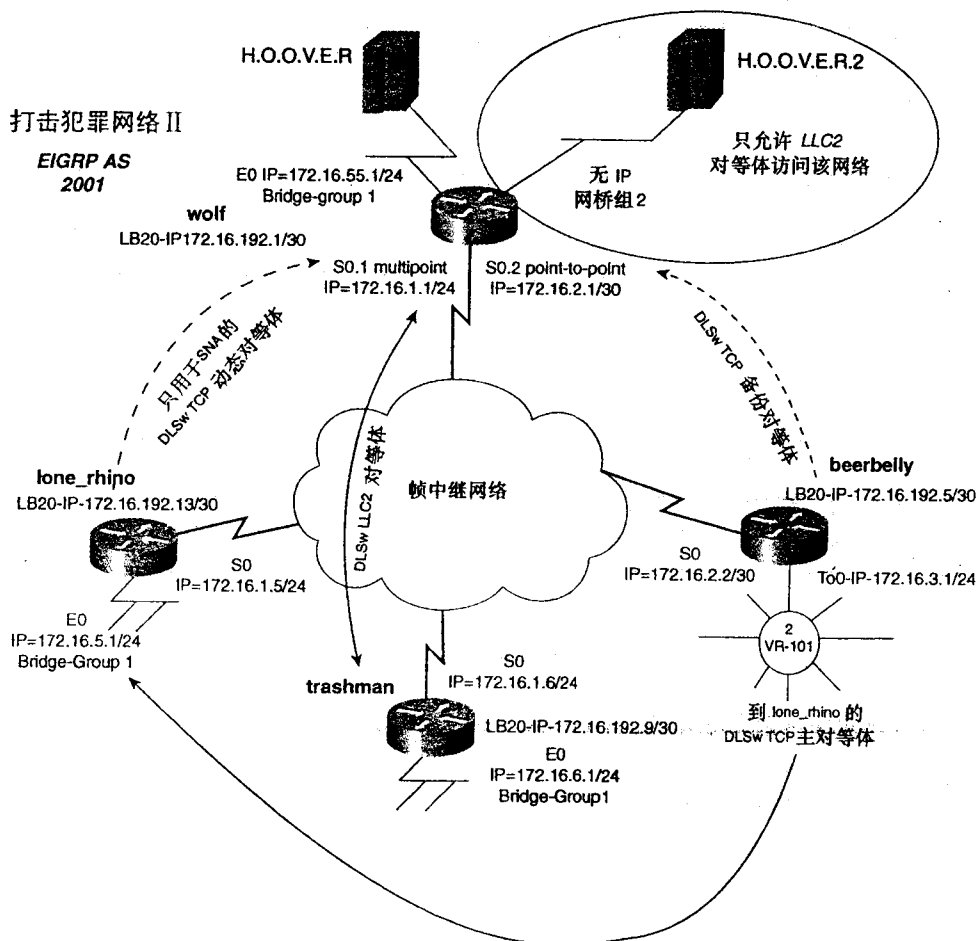


图 13-49 配置了 DLSw 对等体的犯罪团伙网络

例 13-58 路由器 wolf 上的透明桥接配置

```
hostname wolf
!
dlsw local-peer peer-id 172.16.192.1 promiscuous ←Local Peer, Loopback 20
dlsw bridge-group 1 ←Link to bridge 1
dlsw bridge-group 2 ←Link to bridge 2
!
interface Loopback20
ip address 172.16.192.1 255.255.255.252
no ip directed-broadcast
!
interface Ethernet0
ip address 172.16.55.1 255.255.255.0
no ip directed-broadcast
```



```
media-type 10BaseT
bridge-group 1          ← in bridge 1
!
<<<text omitted>>>
!
interface Ethernet2
no ip address
no ip directed-broadcast
media-type 10BaseT
bridge-group 2          ← in bridge 2
!
<<<text omitted>>>
!
bridge 1 protocol ieee   ← STP for bridge 1 and bridge 2
bridge 2 protocol ieee
```

配置完透明网桥组配置后，需要使用 **dls w bridge-group X** 命令将这些组与相应的 DLSw 域连接。这条命令的用法在上一实验中已经在路由器 **wolf** 上演示过。对于路由器 **beerbelly** 的源路由桥接 (SRB) 来说，是虚拟令牌环将 SRB 和 DLSw 域相连接。例 13-59 列出了路由器 **beerbelly** 这一部分的配置示例。

例 13-59 配置路由器 **beerbelly** 上的 SRB

```
hostname beerbelly
!
<<<text omitted>>>
!
source-bridge ring-group 101          ←virtual ring
dls local-peer peer-id 172.16.192.5   ←Local Peer Loopback address
!
interface Loopback20
ip address 172.16.192.5 255.255.255.252
!
interface Serial0
ip address 172.16.2.2 255.255.255.0
encapsulation frame-relay
frame-relay interface-dlci 181
frame-relay lmi-type cisco
!
<<<text omitted>>>
!
interface TokenRing0
ip address 172.16.3.1 255.255.255.0
ring-speed 16
multiring all
source-bridge 2 1 101                ←SRB enabled
!
<<<text omitted>>>
!
router eigrp 2001
network 172.16.0.0
no auto-summary
!
<<<text omitted>>>
beerbelly#
```

第 4 步为所有的路由器配置远程对等体。这个例子中所有的远程对等体都是不同的，现

在逐个进行配置。首先是路由器 **wolf**。

路由器 wolf 上只允许有一个远程对等体，这就是为什么将它的本地对等体配置成混杂对等体的原因。这个需要定义的远程对等体是它与 trashman 之间的 DLSw+或 LLC2 对等体。同时还要对路由器 trashman 可以访问的以太网段加以限制，这是通过只定义网桥组 2 的 DLSw 网桥列表来实现的。然后，该网桥列表会作为参数用于路由器 trashman 的 **remote peer** 命令中。在配置 LLC2 对等体的时候，还需要往 S0.1 接口加上一条 **frame relay map llc2** 命令。例 13-60 就是路由器 wolf 的 DLSw 配置的情况。

例 13-60 配置路由器 wolf 上的 DLSw

```
hostname wolf
!
<<<text omitted>>>
!
dlsw local-peer peer-id 172.16.192.1 promiscuous
dlsw bgroup-list 2 bgroups 2          ←allows only bridge 2
dlsw remote-peer 2 frame-relay interface Serial0.1 130 ←LLC2 remote peer w/bridge
list
dlsw bridge-group 1                  ←DLSw link to bridge groups
dlsw bridge-group 2
!
interface Loopback20
 ip address 172.16.192.1 255.255.255.252
 no ip directed-broadcast
!
interface Ethernet0
 ip address 172.16.55.1 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 1                      ←Bridge group 1
!
<<<text omitted>>>
!
interface Ethernet2
 no ip address
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 2                      ←Bridge group 2
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 logging event subif-link-status
 logging event dlci-status-change
 frame-relay lmi-type cisco
!
interface Serial0.1 multipoint
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 no ip split-horizon eigrp 2001
 frame-relay map llc2 130 broadcast ←LLC2 MAP statement for DLSW
 frame-relay map ip 172.16.1.5 110 broadcast
 frame-relay map ip 172.16.1.6 130 broadcast
!
interface Serial0.2 point-to-point
```

```

ip address 172.16.2.1 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 180
!
<<<text omitted>>>
!
router eigrp 2001
network 172.16.0.0
no auto-summary
!
bridge 1 protocol ieee
bridge 2 protocol ieee
!

```

例 13-61 则是路由器 *trashman* 一端的配置情况。

例 13-61 配置路由器 *trashman* 的 DLSw

```

hostname trashman
!
<<<text omitted>>>
!
dls local-peer peer-id 172.16.192.9
dls remote-peer 0 frame-relay interface Serial0 131 <- LLC2 peer
!
interface Loopback20
ip address 172.16.192.9 255.255.255.0
no ip directed-broadcast
!
interface Ethernet0
ip address 172.16.6.1 255.255.255.0
no ip directed-broadcast
bridge-group 1
!
interface Serial0
ip address 172.16.1.6 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
frame-relay map llc2 131 broadcast <- LLC2 map statement
frame-relay map ip 172.16.1.5 131 broadcast
frame-relay map ip 172.16.1.1 131 broadcast
frame-relay lmi-type cisco
!
<<<text omitted>>>
!
router eigrp 2001
network 172.16.0.0
no auto-summary
!
<<<text omitted>>>
!
bridge 1 protocol ieee

```

路由器 *beerbelly* 的远程对等体配置是到 *lone_rhino* 的主对等体和到 *wolf* 的备份对等体。按照要求，在主对等体重新进入工作状态之后，该路由器的备份对等体不能够将 LLC2 会话断开，因此，不能在该路由器上加入 **linger** 参数。例 13-62 是路由器 *beerbelly* 的相关配

例 13-62 配置路由器 beerbelly 的 DLSw

```

hostname beerbelly
!
<<<text omitted>>>
!
source-bridge ring-group 101
dlsw local-peer peer-id 172.16.192.5
dlsw remote-peer @ tcp 172.16.192.13 <-Primary Peer
dlsw remote-peer @ tcp 172.16.192.1 backup-peer 172.16.192.13 <-Backup Peer
!
interface Loopback20
 ip address 172.16.192.5 255.255.255.252
!
interface Serial0
 ip address 172.16.2.2 255.255.255.0
 encapsulation frame-relay
 frame-relay interface-dlci 181
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!
interface TokenRing0
 ip address 172.16.3.1 255.255.255.0
 ring-speed 16
 multiring all
 source-bridge 2 1 101
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary

```

最后需要配置的远程对等体就是从路由器 lone_rhino 到 wolf 的一个动态 TCP 对等体。配置这一对等体时，为其加入 LSAP 过滤列表以便只允许 SNA 通过，SNA 的 SAP 值是 0x0d0d。在 **remote peer** 命令中加入 **dynamic** 和 **inactivity** 关键字就可以使得该对等体成为动态的。计时器 **inactivity** 的值是指定为 7 分钟。例 13-63 就是路由器 lone_rhino 这一部分的配置情况。动态对等体配置完毕之后，**keepalive** 和超时值会自动分配。

例 13-63 路由器 lone_rhino 的相关配置

```

hostname lone_rhino
!
<<<text omitted>>>
!
dlsw local-peer peer-id 172.16.192.13 promiscuous
dlsw remote-peer @ tcp 172.16.192.1 keepalive 0 lsap-output-list 201 timeout 90
 dynamic inactivity 7 <-dynamic peer
dlsw bridge-group 1
!
interface Loopback20
 ip address 172.16.192.13 255.255.255.252
!
interface Ethernet0
 ip address 172.16.5.1 255.255.255.0
 bridge-group 1

```

```

interface Serial0
 ip address 172.16.1.5 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.1.6 111 broadcast
 frame-relay map ip 172.16.1.1 111 broadcast
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.16.0.0
 no auto-summary
!
ip classless
no ip http server
!
access-list 201 permit 0x0D0D 0x0000
!
bridge 1 protocol ieee

```

配置的效果可以通过检查各个路由器上的对等体的状况来加以验证。把路由器 lone_rhino 的串行链路断开之后，路由器 beerbelly 上的备份对等体开始工作。路由器之间的可达性则可以按照这一章里讨论过的那样，用“Windows 网络”来加以检验。而对于动态对等体的测试，可以把 SAP 切换为 NetBIOS（用 WIN 9x）来进行。模拟 SAP 为 0x0d 的情况不大容易。

例 13-64 是路由器 wolf 上包括它与 beerbelly 之间的备份对等体在内所有对等体的情况。从中可以看到，其中的一个对等体是 LLC2 对等体，而另外两个对等体都是 TCP 混杂对等体。

例 13-64 验证路由器 wolf 的 DLSW 对等体

```

wolf#show dlsw peer
Peers:

```

	state	pkts_rx	pkts_tx	type	drops	ckts	TCP	uptime
LLC2 Se0.1	CONNECT	50	50	conf	0	0	0	00:23:38
TCP 172.16.192.5	CONNECT	14	53	prom	0	0	0	00:06:19
TCP 172.16.192.13	CONNECT	12	9	prom	0	0	0	00:01:02

```

Total number of connected peers: 3
Total number of connections: 3
wolf#

```

例 13-65 是路由器 lone_rhino 所有可能的对等体的情况，包括它与 wolf 之间的动态对等体。其中，一个对等体是 LLC2 对等体，另外两个都是 TCP 混杂对等体。

例 13-65 验证路由器 lone_rhino 上的 DLSW 对等体

```

wolf#show dlsw peer
Peers:

```

	state	pkts_rx	pkts_tx	type	drops	ckts	TCP	uptime
TCP 172.16.192.5	CONNECT	26	32	prom	0	0	0	00:12:34
TCP 172.16.192.1	CONNECT	2	5	dynam	0	0	0	00:00:06

```

Total number of connected peers: 2

```

```
Total number of connections:      2
lone_rhino#
```

例 13-66 则是路由器 beerbelly 上所有可能的对等体的情况。此时备份对等体处在停止工作的状态之中。

例 13-66 验证路由器 beerbelly 上的 DLSw 对等体

```
wolf#show dlsw peer
Peers:
state      pkts_rx  pkts_tx  type  drops  ckts  TCP  uptime
TCP 172.16.192.13  CONNECT    43      32  conf    0    0  0 00:15:23
TCP 172.16.192.1  DISCONN     0       0  conf    0    0  .  .
Total number of connected peers: 1
Total number of connections:    1
beerbelly#
```

13.10 实验 28：DLSw+ 的可达性，边界对等体，按需对等体和弹性对等体的配置——第 1 部分

13.10.1 实验说明

DLSw+ 提出了边界对等体的概念以对探测帧进行控制，对 DLSw 网络进行扩展。边界对等体可以大大减少链路上探测帧数据包的数量，简化远程对等体的配置。这个实验里，大家就有机会配置 DLSw+ 的边界对等体、对等体组和按需对等体了，另外还能对 DLSw 的弹性功能进行配置。

13.10.2 实验内容

假定美国与加拿大两国共同建立了一个旅游网络，让游客们可以访问两国旅游热点地区的最新信息。现在我们就根据下面的要求对这个网络进行配置：

- 按照图 13-50 配置一个 IP 网络，路由选择协议采用 EIGRP，自治系统 ID 为 2001。
- WAN 连接使用 HDLC 协议。在路由器 us_tour 和 us_border 之间配置两条 WAN 连接，其中一条作为另一条的备份。两条连接不能同时处在工作状态之中。
- 按照图 13-50，以下面的要求为准对网络中的 DLSw 对等体进行配置：
 - 在网络所有的以太网段中，配置 SNA 和 NetBIOS 的任意可达性。
 - 不要在路由器 us_tour 和 canada_tour 之间配置对等体。
 - 不要在任何 WAN 接口上配置桥接。
 - 对 DLSw 的可达性进行配置，使得路由器 us_tour 会把它的可达性信息发送到

名为 US_STATIONS 的工作站去，而路由器 Canada_tour 则会把它的可达性信息发送到名为 CANADA_STATIONS 的工作站去。

- 对路由器 us_tour 和 us_border 之间的对等体进行配置，使得主 HDLC 连接出现故障的情况下该对等体可以保持在工作状态，而在网络对新的连接进行收敛的过程中，该对等体也不会退出工作状态。

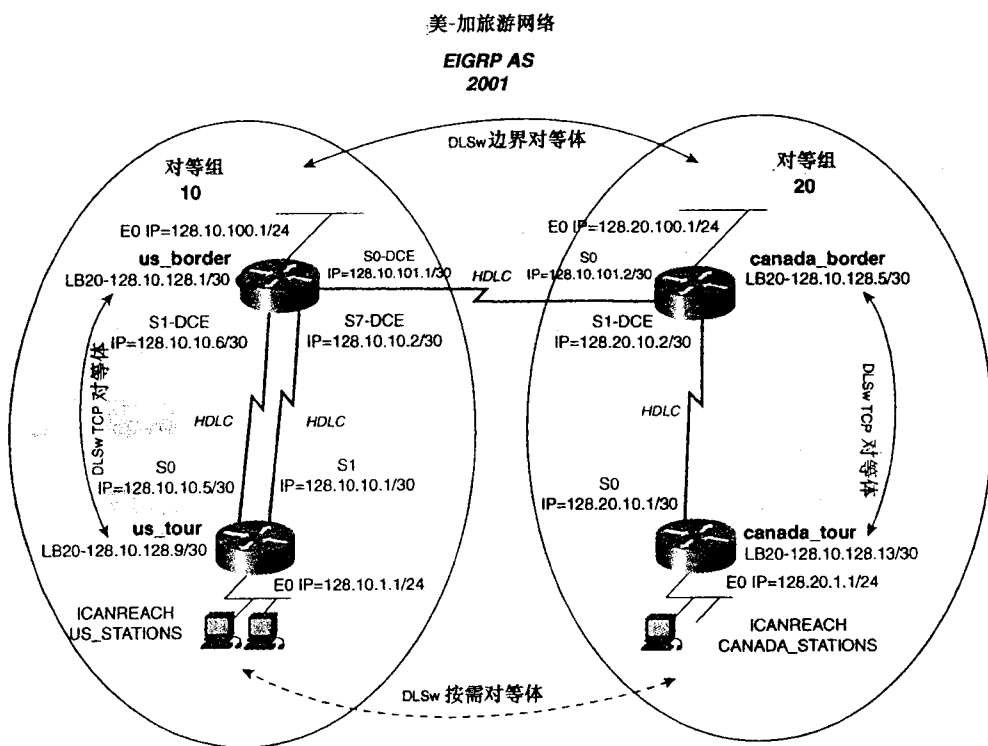


图 13-50 美加旅游网络

13.10.3 实验目的

- 按照图 13-50 配置一个 IP 网络，路由协议采用 EIGRP，自治系统 ID 为 2001。
- WAN 连接使用 HDLC 协议。在路由器 us_tour 和 us_border 之间配置两条 WAN 连接，其中一条作为另一条的备份。两条连接不能同时处在工作状态之中。
- 按照图 13-50，以下面的要求为准对 DLSw 对等体组和边界对等体进行配置：
 - 配置两个 DLSw 对等体组，对等体组 10 含有 U.S.的路由器，而对等体组 20 则含有 Canada 的路由器，同时对这两个组之间的可达性进行配置。
 - 对 DLSw 进行配置，使得路由器 us_tour 和 canada_tour 之间的即时对等体是按需创建的，二者之间不要配置成一个对等体。
 - 对 DLSw 的可达性进行配置，使得路由器 us_tour 宣告到名为 US_STATIONS

CANADA_STATIONS 的工作站的可达信息。

- 对路由器 `us_tour` 和 `us_border` 之间的对等体进行配置，使得主 HDLC 连接出现故障的情况下该对等体可以保持工作状态，而在网络对新的连接进行收敛的过程中，该对等体也不会退出工作状态。

13.10.4 所需设备

- 4 台 Cisco 路由器，通过 V.35 背对背或类似的方式连接在一起。
- 通过集线器或交换机创建的 4 个 LAN 网段。LAN 的拓扑结构对该实验结果没有影响。

13.10.5 物理设计与实验准备

- 按照图 13-50 将集线器以及串行线缆与路由器相互连接好。
- 按需分配的按需对等体的测试可能要用到 Windows 9x 工作站。

13.11 实验 28：DLSw+ 的可达性，边界对等体，按需对等体和弹性对等体的配置——第 2 部分

13.11.1 实验步骤

利用 V.35 线缆或者是带有反接线缆的 CSU/DSU 将帧中继交换机与 4 台路由器以背对背方式连接在一起。然后按照图 13-50 利用交换机或集线器/MAU 创建 4 个以太 LAN 网段和令牌环 LAN 网段。

物理连接完成之后，按照图 13-50 为所有的 LAN 和 WAN 接口分配 IP 地址。在配置 WAN 接口的时候，如果采用的是背对背线缆，链路的某一端必须配置成数据通信设备（DCE），这通过在该端的接口上用 `clock rate speed` 命令设置通信速率来完成。WAN 协议是 HDLC，因此无须再做进一步的配置来激活链路。按照要求，路由器 `us_border` 和 `us_tour` 之间一次只能有一条串行链路处在活动状态。因此需要用 `backup interface` 命令将其中的一条链路设置为另一条的备份链路。

例 13-67 就是路由器 `us_border` 上串行链路的配置。

例 13-67 路由器 `us_border` 串行链路的配置

```
hostname us_border
!
<<<text omitted>>>
!
interface Serial0
```


		第 13 章	置桥接和	数据链路	DLSW+)		767
		address 1 fair-queu ckrate 64	.101.1 2 55.255.25				
		rface Ser kup dela kup inte address ckrate 6	Serial7 .10.6 2 5.255.252				
		text omit	>				
		rface Se address ckrate 6	3.10.1 2 5.255.252				
		text omit	>				
		ter eigrp twork 128 auto-sum	.0				
这个 的 E	ping 命	所有的路	器本地接	行测试	配置 1	P 为路	译协议。
	模型中会	个主网	128.20.0	128.10	例 13-6	路由器	a_bord
	配置示						
	3-68	路由器 c	ia_bord	EIGRP			
		ter eigr twork 12 twork 12 auto-su	.0 .0				
下面	开始对	v 进行配	首先是	路由器	对等体的	。DLS	配置分
	步骤:						
	步 配	等体的环	接口地址				
	步 对	对等体透	配置。				
	步 对	或透明	进行配置				
	步 对	对等体透	配置。				
	是在所	路由器上	量相应的	接口地	这是为	远程对	准备的
	步是本	等体的配	U.S.的	器位于	组 10	把一个	对等体
入至	一个对等	中去，	通过在	l-peer	中加入	参数	。路由
us_1	r 实际上	等体组	的边界对	。只要	al-peer	中加入	border
键与	可以实现	对等体	置。而 C	的路由	是在对	组 20 中	对等体
的文	各路由器是	da_bord					
	第 3 步对	桥接进行	置时，需	接口命	idge-gr	将以	加入到
桥结	同时还	全局命	idge 1 pr	ol IEEE	动 STP	作，这	“1”是
透明	桥的号码	有路由器	透明桥接	置都完	洋。例 1	是路由	_tour 的
明	的配置中	同时还	了将网桥	加到 D	域中去	的命令。	

例 13-69 配置透明桥接

```

dlsw bridge-group 1          ← Attach Bridge 1 to DLSw
!
<<<text omitted>>>
!
interface Ethernet0
 ip address 128.10.1.1 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 1
!
<<<text omitted>>>
!
bridge 1 protocol ieee
!

```

第 4 步是配置路由器 us_tour 和 us_border 之间的 TCP 对等体。在串行线路进行收敛的过程中，按要求，这个对等体的连接不能丢失。为了防止对等体在链路发生故障时退出工作状态，为链路两端的远程对等体设置 **timeout 500** 和 **keepalive 0** 的参数。路由器 us_tour 和 canada_tour 之间也需要配置按需对等体。配置按需对等体的全局命令如下：

```
us_tour (config) #dlsw peer-on-demand-defaults tcp-queue-max 50
```

路由器 us_tour 上最后需要配置的就是 DLSw 的可达性。为了将该路由器的可达性信息转发到 US_STATION 工作站，采用下面这条 DLSw 命令：

```
us_tour (config) #dlsw icanreach netbios-name US_STATIONS
```

例 13-70 是路由器 us_tour 的完整配置情况。

例 13-70 路由器 us_tour 的完整配置清单

```

hostname us_tour
!
<<<text omitted>>>
!
dlsw local-peer peer-id 128.10.128.9 group 10
dlsw remote-peer 0 tcp 128.10.128.1 keepalive 0 timeout 500
dlsw icanreach netbios-name US_STATIONS
dlsw peer-on-demand-defaults tcp-queue-max 50
dlsw bridge-group 1
!
!
interface Loopback20
 ip address 128.10.128.9 255.255.255.252
 no ip directed-broadcast
!
interface Ethernet0
 ip address 128.10.1.1 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
 bridge-group 1
!
<<<text omitted>>>
!
interface Serial0
 ip address 128.10.10.5 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache

```

```

interface Serial1
 ip address 128.10.10.1 255.255.255.252
 no ip directed-broadcast
 !
<<<text omitted>>>
 !
router eigrp 2001
 network 128.10.0.0
 no auto-summary
 !
<<<text omitted>>>
 !
bridge 1 protocol ieee
 !

```

路由器 us_border 还与 canada_border 之间建立了一个额外的对等体。例 13-71 是路由器 us_border 的完整配置。

例 13-71 路由器 us_border 的完整配置清单

```

hostname us_border
 !
dls local-peer peer-id 128.10.128.1 group 10 border
dls remote-peer 0 tcp 128.10.128.9 keepalive 0 timeout 500
dls remote-peer 0 tcp 128.10.128.5
dls bridge-group 1
 !
interface Loopback20
 ip address 128.10.128.1 255.255.255.252
 !
interface Ethernet0
 ip address 128.10.100.1 255.255.255.0
 bridge-group 1
 !
interface Serial0
 ip address 128.10.101.1 255.255.255.252
 no fair-queue
 clockrate 64000
 !
interface Serial1
 backup delay 0 0
 backup interface Serial7
 ip address 128.10.10.6 255.255.255.252
 clockrate 64000
 !
<<<text omitted>>>
 !
interface Serial7
 ip address 128.10.10.1 255.255.255.252
 clockrate 64000
 !
<<<text omitted>>>
 !
router eigrp 2001
 network 128.10.0.0
 no auto-summary
 !
<<<text omitted>>>
 !

```

(待续)

```
bridge 1 protocol ieee
```

如果要查看 us_border 对等体的 NetBIOS 的可达性，可以如例 13-72 输入 **show dlsw reachability** 命令。可以看到边界路由器上有名为 US_STATIONS 的 NetBIOS 信息。

例 13-72 在路由器 us_border 上查看可达性

```
us_border#show dlsw reachability
DLsw Remote MAC address reachability cache list
Mac Addr      status    Loc.    port          rif

DLsw Local MAC address reachability cache list
Mac Addr      status    Loc.    peer
0000.613c.dc82 FOUND      REMOTE  128.10.128.5(2065)

DLsw Local NetBIOS Name reachability cache list
NetBIOS Name  status    Loc.    port          rif

DLsw Remote NetBIOS Name reachability cache list
NetBIOS Name  status    Loc.    peer
US_STATIONS   UNCONFIRM REMOTE  128.10.128.9(2065)

us_border#
```

路由器 canada_border 和 canada_tour 的配置过程和这两台 U.S. 路由器完全一样。例 13-73 是这些路由器的完整配置清单。

例 13-73 Canada 的路由器配置清单

```
hostname canada_border
!
<<<text omitted>>>
!
dlsw local-peer peer-id 128.10.128.5 group 20 border
dlsw remote-peer 0 tcp 128.10.128.1
dlsw remote-peer 0 tcp 128.10.128.13
dlsw bridge-group 1
!
!
interface Loopback20
ip address 128.10.128.5 255.255.255.252
!
interface Ethernet0
ip address 128.20.100.1 255.255.255.0
bridge-group 1
!
interface Serial0
ip address 128.10.101.2 255.255.255.252
no fair-queue
!
interface Serial1
ip address 128.20.10.2 255.255.255.252
clockrate 64000
!
<<<text omitted>>>
!
```

```

router eigrp 2001
  network 128.10.0.0
  network 128.20.0.0
  no auto-summary
!
<<<text omitted>>>
!
bridge 1 protocol ieee
!

hostname canada_tour
!
!
dlsw local-peer peer-id 128.10.128.13 group 20
dlsw remote-peer 0 tcp 128.10.128.5
dlsw icanreach netbios-name CANADA_STATIONS
dlsw peer-on-demand-defaults tcp-queue-max 50
dlsw bridge-group 1
!
<<<text omitted>>>
!
interface Loopback20
  ip address 128.10.128.13 255.255.255.252
!
interface Ethernet0
  ip address 128.20.1.1 255.255.255.0
  bridge-group 1
!
interface Serial0
  ip address 128.20.10.1 255.255.255.252
  no fair-queue
!
<<<text omitted>>>
!
router eigrp 2001
  network 128.20.0.0
  network 128.10.0.0
  no auto-summary
!
<<<text omitted>>>
!
bridge 1 protocol ieee

```

可以通过查看边界对等体的 DLSw 可达性来确认这些配置。正常情况下，可以看到与边界对等体同处一个组中对等体发送过来的静态 ICANREACH 信息。前面提到，Windows 工作站是用于测试 DLSw 的理想平台。通过将该工作站放置到不同 LAN 网段中，可以产生数据、创建电路以及控制探测帧数据包。例 13-74 中将工作站连接到了 DLSw 区域上，并向 NetBIOS 站点 BORDER-PATROL 发送了一个探测帧数据包。

例 13-74 配置的确认

```

us_border#show dlsw reachability
DLSw Remote MAC address reachability cache list
Mac Addr      status      Loc.      port      rif

DLSw Local MAC address reachability cache list
Mac Addr      status      Loc.      peer
0000.613c.dc82 FOUND      REMOTE    128.10.128.5(2065)

```

```

DLSw Local NetBIOS Name reachability cache list
NetBIOS Name   status   Loc.   port   rif
BORDER-PATROL  SEARCHING LOCAL

DLSw Remote NetBIOS Name reachability cache list
NetBIOS Name   status   Loc.   peer
TOURIST        FOUND    REMOTE 128.10.128.5(2065)
US_STATIONS    UNCONFIRM REMOTE 128.10.128.9(2065)

us_border#

```

检验 DLSw 对等体的连接状况与 DLSw 性能是确保 DLSw 网络正常工作的另一种办法。

例 13-75 是在路由器 canada_border 上执行 **show dlsw peer** 命令进行检验的结果。

例 13-75 在路由器 canada_border 上对对等体进行检验

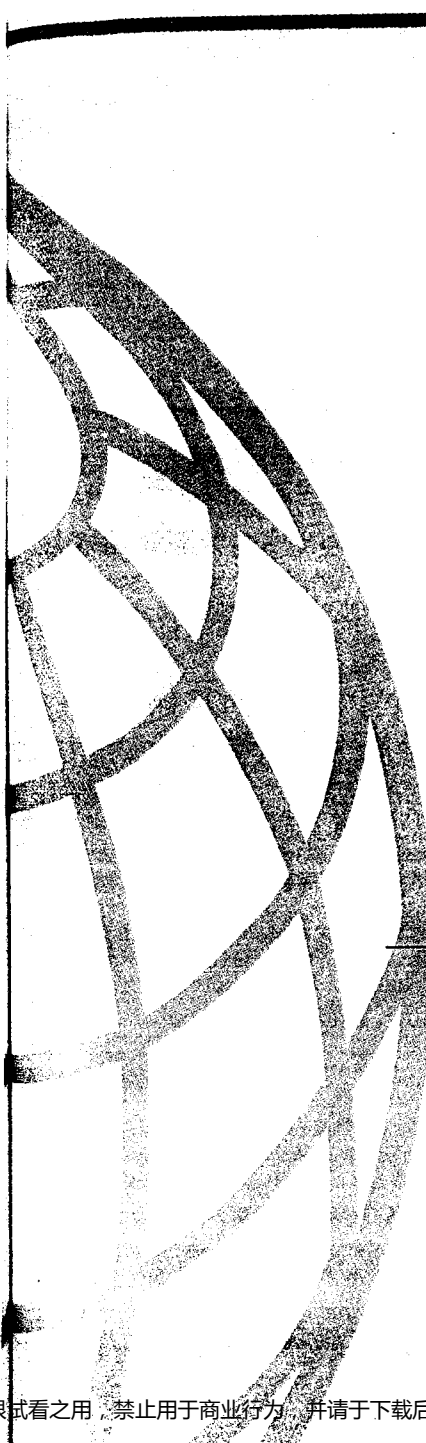
```

canada_border#show dlsw peer
Peers:                state   pkts_rx  pkts_tx  type  drops  ckts TCP  uptime
TCP 128.10.128.1      CONNECT      65       85  conf      0      0  0 00:32:20
TCP 128.10.128.13     CONNECT      67      103  conf      0      0  0 00:33:00

Total number of connected peers: 2
Total number of connections:    2

canada_border#

```



第6 部分

网络控制与网络 访问

第14章 理解 IP 访问控制列表

第 14 章

理解 IP 访问控制列表

现代网络中，全面而无限制的 IP 访问已不可取。这既有安全方面的考虑也有管理上的因素，例如，具有同样 IP 地址空间的两家公司合并到一起。很快就会有网络访问限制的要求，这种情况下，就需要使用访问控制列表。

对路由更新、数据传输路径以及协议的控制是路由配置过程中更为复杂的问题之一。理解二进制算术原理及其与访问控制列表的关系对访问控制列表的学习非常关键。而对于数据过滤非常重要且熟悉并理解所要过滤的协议的特性。

这一章将简单讨论访问控制列表的相关内容，并解释二进制计算的重要性，还讲述了 IP 访问控制列表的不同类型：标准、扩展、动态以及命名。

注释 理解协议的工作原理对于过滤该协议非常关键。制定 IP 协议的过滤规则时，必须知道客户端采用哪个端口来建立到服务器的连接，还要知道服务器向客户端发送数据所用的端口号，这两个端口可能不一样。在这一节的后续部分可以看到，FTP 就是一个很好的例子，该协议发送数据的端口和用来建立会话的端口是两个不同的端口。在制定特定的 IP 数据过滤规则时，可以参考这方面的著作，如 Richard Stevens 的《TCP/IP Illustrated》和 Douglas Comer 的《Internetworking with TCP/IP》，首先弄清楚协议工作原理，然后再设计过滤规则。

14.1 理解访问控制列表的工作方式

从本质上说，访问控制列表是一系列自上而下按序逐条执行的控制条件。满足某个条件时，就会跳出执行，不再进行下一步比较，并且向调用该访问控制列表的进程返回一个“真”或“假”的结果。访问控制列表多年来新增了很多不同种类。Cisco IOS12.0 增加了 IP 扩展访问控制列表的范围，如例 14-1 所示。

例 14-1 Cisco IOS 12.0 中访问控制列表的范围

```
router(config)#access-list ?
<1-99>      IP standard access list
<100-199>    IP extended access list
<1000-1099>  IPX SAP access list
<1100-1199>  Extended 48-bit MAC address access list
<1200-1299>  IPX summary 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)
<300-399>    DECnet access list
<400-499>    XNS standard access list
<500-599>    XNS extended access list
<600-699>    Appletalk access list
<700-799>    48-bit MAC address access list
<800-899>    IPX standard access list
<900-999>    IPX extended access list
```

标准访问控制列表的过滤基于一个条件，即地址的匹配。在考虑访问控制列表时，将其看成“真”(true)或“假”(false)的条件，访问控制列表会把结果返回给调用的进程。这样看待访问控制列表很重要，因为访问控制列表不只是用来过滤数据包，还包括路由图、再分布信息以及其他特性，如网络地址转换(NAT)等。所以，不要把访问控制列表仅仅限定在“网络”或“数据包”的范围，而应考虑什么进程在调用访问控制列表，访问控制列表返回给该进程什么结果。访问控制列表返回的仅仅是条件满足与否的结果，或者是“真”，或者是“假”。这样，调用了该列表的进程就会根据这一返回结果做出允许或拒绝的决定。

在配置访问控制列表时，应遵守以下规则和建议：

- 所有访问控制列表的最后都存在一个隐含的 **deny**，这条在配置文件是看不到的。
- 访问控制列表是自上而下，按顺序执行的。某个条件满足时，执行停止，不再做进一步的比较。
- 访问控制列表中的各项应该按照从特殊到一般的顺序配置。先拒绝特定的主机，然后才是组或一般条件的过滤操作。
- 表中新的列表项总是加在访问控制列表的最后。命令 **no access-list x** 可以删除整个列表，不能有选择地添加或删除某些条目。
- 没有任何条目定义的访问控制列表能允许所有的数据。
- 在配置访问控制列表时，先配置好列表再将它应用于某个进程，无论是标准数据包过滤、路由图或是 **redistribute** 命令都应该如此。这样能很容易测试或删除新建的列表。

- IP 访问控制列表会发送一个 ICMP 主机不可到达的消息到数据包的发送者，然后将数据包丢弃。
- 采用的过滤规则与所要过滤的数据源要尽量地接近。安全过滤通常是阻止入站访问，而数据过滤则通常是禁止数据通过某链路，使用出站过滤方式。
- 删除某访问控制列表时要非常小心。如果访问控制列表用于某个实际接口，删除列表之后，该接口会因为一条默认的 **deny any** 规则而终止所有的数据传输。
- 出站过滤不会影响路由器自身产生的数据。

14.2 访问控制列表、反向掩码和二进制算术

任何数字世界使用的都是 0 和 1 的语言，访问控制列表也不例外。标准型和扩展型的访问控制列表都采用反向掩码的概念，这是以点分十进制的方式表示的一个二进制数。访问控制列表里的地址位，或者说第一个地址项，会和一个相应的反向掩码进行比较。如果反向掩码的相应位是 0，这说明访问控制列表中地址的相应位必须和所比较数据包中的地址位相匹配，因此这一位称为**相关位**。如果反向掩码的位是 1，说明这一位不会进行任何比较，通常把这一位叫做**无关位**。相关位和无关位的概念只有在分解成为二进制形式的情况下才有意义。

图 14-1 是配置在路由器的以太 0/0 端口上的一个简单访问控制列表的例子。接下来查看访问控制列表的工作过程。

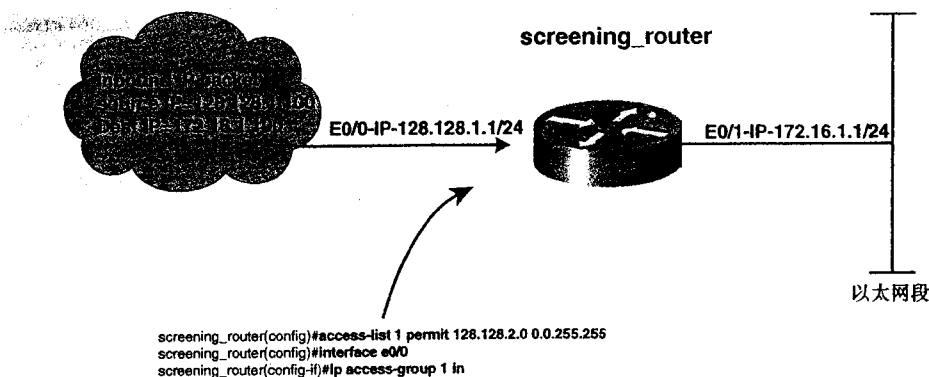


图 14-1 简单的访问控制列表实例

理解访问控制列表的第一步是将访问控制列表的内容用二进制表示出来。图 14-1 中有下面这样一条列表项：

```
access-list 1 permit 128.128.2.0 0.0.255.255
```

将其第一部分写成二进制为：

128	.128	.2	.0
1000 0000	1000 0000	0000 0010	0000 0000

说比较位中的相应位必须要匹配的。将第二部分写成二进制就是：

0	.0	.255	.255
0000 0000	0000 0000	1111 1111	1111 1111

现在把这两个部分放到一起，就能清楚地看到哪些位有意义。在该例中，由于掩码中相应的位是 0，因此地址位部分的前两个字节都是相关的位。

1000 0000	1000 0000	0000 0010	0000 0000
0000 0000	0000 0000	1111 1111	1111 1111

根本上说，任何要与这个访问控制列表进行匹配的地址的第 1 位必须是 1，接下来的 7 个位则必须是 0。第 2 个 8 位字节也是如此。在第 3、4 字节中，反向掩码全是 1，因此无需考虑这两个字节与列表如何匹配，是 1 是 0 都没有关系。

这个例子使用的是标准访问控制列表，因此路由器会将入数据包的源 IP 地址用于与访问控制列表的比较。将源地址分解：

128	.128	.1	.100
1000 0000	1000 0000	0000 0001	0100 0100

检查第 1 位，是一个 1。从上面可知这一位是有关的，必须是 1。第 2 位是 0，同样，反向掩码的对应值说明这一位也是有关的，必须是 0。做个完整的比较就会发现，这个地址产生 true，即肯定的结果。

路由器使用的方法是逻辑“或”，或者是叫做“布尔或”。访问控制列表中逻辑“或”的使用方式和路由器用逻辑“与”来把目的数据包的地址和路由器接口上的掩码进行比较以找到子网中某个特定的地址的方式一样。

在用逻辑“与”比较两个二进制数时的规则就是，当且仅当两个被比较的位都是 1 时，结果才会是 1。例如，用逻辑“与”比较两个地址：128.128.1.1 和 255.255.255.0，结果是 128.128.1.0。

1000 0000	1000 0000	0000 0001	0000 0001
1111 1111	1111 1111	1111 1111	0000 0000
1000 0000	1000 0000	0000 0001	0000 0000
结果：128.128.1.0			

而逻辑“或”与此相反，它在比较两个二进制数时的规则是，当且仅当被比较的两个位都是 0 时结果才是 0。例如，用逻辑或比较两个地址：128.128.1.1 和 255.255.255.0，结果是 255.255.255.1。

1000 0000	1000 0000	0000 0001	0000 0001
1111 1111	1111 1111	1111 1111	0000 0000
1111 1111	1111 1111	1111 1111	0000 0001

现在把逻辑“或”的概念用于访问控制列表。所有的访问控制列表，无论是标准的还是扩展的，在对反向掩码和所测试的地址之间做了逻辑“或”之后会产生一个结果，如果这个结果与访问控制列表中地址和掩码经过逻辑或得出的结果相等，那整个比较的结果就是“真”。例如，假设采用了下面这条标准访问控制列表项：

```
access-list 1 permit 128.128.0.0 0.0.255.255
```

要进行过滤的源地址有两个，128.128.1.1 和 128.192.1.1。哪一个地址可以通过访问控制列表的过滤呢？

将地址和反向掩码进行逻辑“或”，128.128.1.1 和 0.0.255.255 逻辑“或”得 128.128.255.255。然后，再把地址与访问控制列表的掩码进行逻辑“或”，128.128.0.0 与 0.0.255.255 逻辑“或”产生 128.128.255.255。由于这两个结果是相同的，所以访问控制列表对地址 128.128.1.1 产生一个“真”（true）的结果。对第 2 个地址，128.192.1.1 和 0.0.255.255 逻辑“或”得到的是 128.192.255.255。而访问控制列表中的地址和掩码逻辑“或”的结果还是 128.128.255.255，两个结果不一样，因此访问控制列表对地址 128.192.1.1 产生的结果是“假”。

在这个例子中，位边界清楚地说明了一切，进行逻辑“或”看上去需要很多的工作。但是，在用 `access-list 1 permit 64.35.100.150 0.4.10.254` 这样的访问控制列表开始练习时，将这些数字分解成二进制的 0 和 1 进行逻辑“或”是弄清楚哪些位有意义、哪些没有意义的惟一途径。

14.3 标准访问控制列表

到现在为止，本章已经从总体上讨论了访问控制列表。现在进一步讨论特定类型的 IP 访问控制列表及其配置方式。

在 Cisco IOS 12.0 中，标准访问控制列表是处在 1 到 99 和 1300 到 1999 的范围之内的。任何访问控制列表项的最后都可以加上一个 `log` 关键字，这样，与该项匹配的信息就会发送到控制台，其句法为：

```
access-list x {deny | permit} a.b.c.d wildcard_mask {log}
```

这里的 `a.b.c.d` 参数是掩码要与之逻辑“或”以产生“真”或“假”的结果的 IP 地址。

标准访问控制列表的应用方式包括：

- 数据包过滤。
- 路由过滤。
- 为某些功能（如 NAT 或 DDR 链路）定义期望数据。

当然，还有很多应用，这就是为什么不能把访问控制列表的作用仅仅限定在数据包过滤的原因。

本节的例子利用了标准访问控制列表来过滤路由，拒绝网络访问以及拒绝虚拟终端的访问等。图 14-2 是将 EIGRP 作为路由选择协议的简单网络。该例中，通过配置多个访问控制列表可以实现：

- 禁止 jefferson 具有到 172.16.2.0/24 的路由。
- 允许用户 172.16.1.129 通过 Telnet 访问 henry，但是拒绝其他任何访问方式。

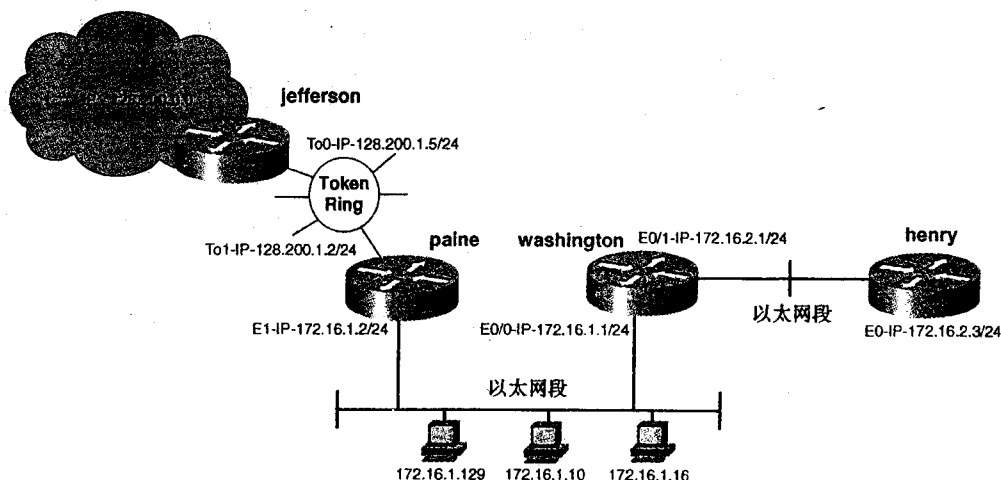


图 14-2 标准 IP 访问控制列表实例

通常情况下，每台路由器上加上了 **eigrp** 的 **no auto-summary** 命令之后，路由器 **jefferson** 上的路由表如例 14-2 所示。

例 14-2 路由器 jefferson 上的路由表

```
jefferson#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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 2 subnets
D    172.16.1.0 [90/297728] via 128.200.1.2, 00:00:31, TokenRing0
D    172.16.2.0 [90/323328] via 128.200.1.2, 00:00:18, TokenRing0
128.200.0.0/24 is subnetted, 1 subnets
C    128.200.1.0 is directly connected, TokenRing0
jefferson#
```

禁止 **jefferson** 访问 172.16.2.0/24 的一个方法就是在路由器 **paine** 上利用分布列表来调用访问控制列表。例 14-3 是在 **paine** 上加入访问控制列表的情况。前面关于路由选择协议的那一章讲过，过滤表用来过滤路由更新。

例 14-3 加入过滤列表来调用标准访问控制列表

```
paine(config)#router eigrp 2001
paine(config-router)#distribute-list 1 out to1
paine(config-router)#exit
paine(config)#access-list 1 deny 172.16.2.0 0.0.0.255
paine(config)#access-list 1 permit any
paine(config)#exit
paine#
```

例 14-4 是应用访问控制列表之后路由器 jefferson 上的路由表。

例 14-4 路由器 jefferson 的路由表

```
jefferson#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, * - candidate default
        U - per-user static route, o - ODR
        T - traffic engineered route

Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 2 subnets
D       172.16.1.0 [90/297728] via 128.200.1.2, 00:00:31, TokenRing0
C       128.200.1.0 is directly connected, TokenRing0
jefferson#
```

按照本章开始列出的规则，先过滤特殊地址，然后再允许一般地址。列表最后有一个隐含的 **deny any**，因此需要在这个隐含的 **deny any** 之前允许不希望滤掉的路由更新。表 14-1 列出了访问控制列表的快捷表示。

表 14-1 访问控制列表的快捷表示

地址	掩码	返回 true 值	快捷表示
0.0.0.0	255.255.255.255	任何地址都返回 true	Any
a.b.c.d	0.0.0.0	精确匹配地址 a.b.c.d	host

要限制路由器 henry 上的 Telnet 访问，可以在路由器的 vty 端口上运用标准访问控制列表。第 1 章“建立 Internet 网络模型所需的主要组件”讨论过，**show line** 命令可以用来显示 vty，也就是 Telnet 访问端口。下面这条命令可以在路由器的某个端口上运用访问控制列表：

```
access-class access-list_number {in | out}
```

这里要允许地址 172.16.1.129 通过 Telnet 访问 henry，但是拒绝所有其他访问方式。要做到这一点，首先要确定 vty 会话的绝对线路号，这可以用 **show line** 命令来完成。接着，应用一个访问组，调用针对这些线路号的访问控制列表。例 14-5 列出了限制某地址通过 Telnet 访问 henry 所需的命令。

例 14-5 用标准 IP 访问控制列表控制 Telnet 的访问

```
henry#show line
Tty Typ Tx/Rx A Modem Roty Acc0 AccI Uses Noise Overruns
* 0 CTY - - - - - 0 0 0/0
  1 AUX 9600/9600 - - - - - 0 0 0/0
  2 VTY - - - - - 2 0 0/0
    ←Telnet sessions
  3 VTY - - - - - 0 0 0/0
  4 VTY - - - - - 0 0 0/0
  5 VTY - - - - - 0 0 0/0
```

(待续)

子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责。

表 14-2 Cisco IOS 12.0 中扩展 IP 访问控制列表的 protocol_type 值

值	意 义
<0-255>	IP 协议号
ahp	认证报头协议
eigrp	CISCO EIGRP 路由选择协议
esp	封装安全负载
gre	CISCO GRE 隧道
icmp	因特网控制信息协议
igmp	因特网网关信息协议
ip	所有因特网协议
ipinip	IP 隧道中的 IP
nos	IP 隧道之中的 KA9Q NOS-兼容 IP
ospf	OSPF 路由选择协议
pcp	载荷压缩协议
pim	独立于路由选择协议的多播
tcp	传输控制协议
udp	用户数据报协议

上面表中显示字段协议类型 (protocol_type) 的值随着 Cisco IOS 发展而发展。指定协议类型是避免复杂过滤方法的一个简单途径。例如，在过滤 IGRP、EIGRP、OSPF 等路由选择协议时，通过协议类型 (protocol_type) 关键字指定协议类型的方法很简单，但是要是想通过这些路由选择协议所使用的某些多播信息来进行过滤就很麻烦。表 14-3 列出了当前 Cisco IOS 12.0 支持的 TCP 端口号。

表 14-3 Cisco IOS 12.0 支持的扩展 IP 访问控制列表可用 TCP 端口号

值	含 义
<0-65535>	端口号
bgp	边界网关协议 (179)
chargen	符号发生器 (19)
cmd	远程命令 (rcmd, 514)
daytime	时间 (13)
discard	丢弃 (9)
domain	域名服务 (53)
echo	应答 (7)
exec	操作 (rsh, 512)

续表

值	含 义
finger	Finger 协议 (79)
ftp	文件传输协议
ftp-data	FTP 数据连接 (不常使用, 20)
gopher	gopher 协议 (70)
hostname	NIC 主机名服务器 (101)
ident	IDENT 协议 (113)
irc	因特网延时对话 (194)
klogin	Kerberos 登录 (543)
kshell	Kerberos shell (544)
login	远程登陆 (rlogin, 513)
lpd	打印服务 (515)
nntp	网络新闻传输协议 (119)
pim-auto-rp	PIM auto-RP (496)
pop2	邮局协议 V2 (109)
pop3	邮局协议 V3 (110)
smtp	简单邮件传输协议 (25)
sunrpc	SUN 远程进程调用 (111)
syslog	系统日志服务 (514)
Tacacs	TAC 访问控制系统 (49)
Talk	对话 (517)
telnet	TELNET 协议 (23)
Time	实时时间同步 (37)
Uucp	UNIX 到 UNIX 拷贝程序 (540)
whois	别名 (43)
www	WWW 服务 (HTTP, 80)

表 14-4 列出的是当前 Cisco IOS 12.0 支持的 UDP 端口号。

表 14-4 Cisco IOS 12.0 支持的扩展 IP 访问控制列表可用 UDP 端口号

值	含 义
<0-65535>	端口号
biff	BIFF (邮件通知, comsat, 512)
bootpc	BOOTSTRAP 协议客户 (68)

续表

值	含 义
bootps	BOOTSTRAP 协议服务器 (67)
discard	丢弃 (9)
dnsix	DNSIX 安全协议审查 (195)
domain	域名服务 (DNS, 53)
echo	应答 (7)
isakmp	因特网安全助理和关键管理协议 (500)
mobile-ip	移动 IP 注册 (434)
nameserver	IEN 16 命名服务 (42)
netbios-dgm	NETBIOS 数据报服务 (138)
netbios-ns	NETBIOS 命名服务 (137)
netbios-ss	NETBIOS 会话服务 (139)
ntp	网络时间协议 (123)
pim-auto-rp	PIM 自动 RP (496)
rip	路由信息协议 (router, in.routed, 520)
snmp	简单网管协议 (161)
snmptrap	SNMP 陷阱 (162)
sunrpc	SUN 远程进程调用 (111)
syslog	系统日志 (514)
Tacacs	TACACS 访问控制系统 (49)
talk	对话 (517)
tftp	简单文件传输协议 (69)
time	时间 (37)
who	WHO 服务 (RWHO, 513)
xdmcp	X 显示管理控制协议 (177)

扩展访问控制列表的另一个增强功能是可以按数据源和数据目的指定匹配条件。访问控制列表的末尾可以设定优先级与服务类型 (TOS) 值。路由器在 IP 数据包的报头中检查该值。优先级的范围是 0 到 7，而 TOS 的范围则是 0 到 15。关键字 **established** 用来在 TCP 报头中查找确认 (ACK) 或重置 (RESET) 标志。如果设置了这些标志，那么就匹配。这一命令用来使已建立的数据流通过访问控制列表。

下面看一下用在 Internet 连接上的扩展访问控制列表的情况。这个例子中，和 Internet 相连的 BRI 接口上应用了进入方向的扩展访问控制列表。图 14-3 是这个例子的网络拓扑示例。

现在，把例 14-4 的命令语句加到路由器 access_router 上。

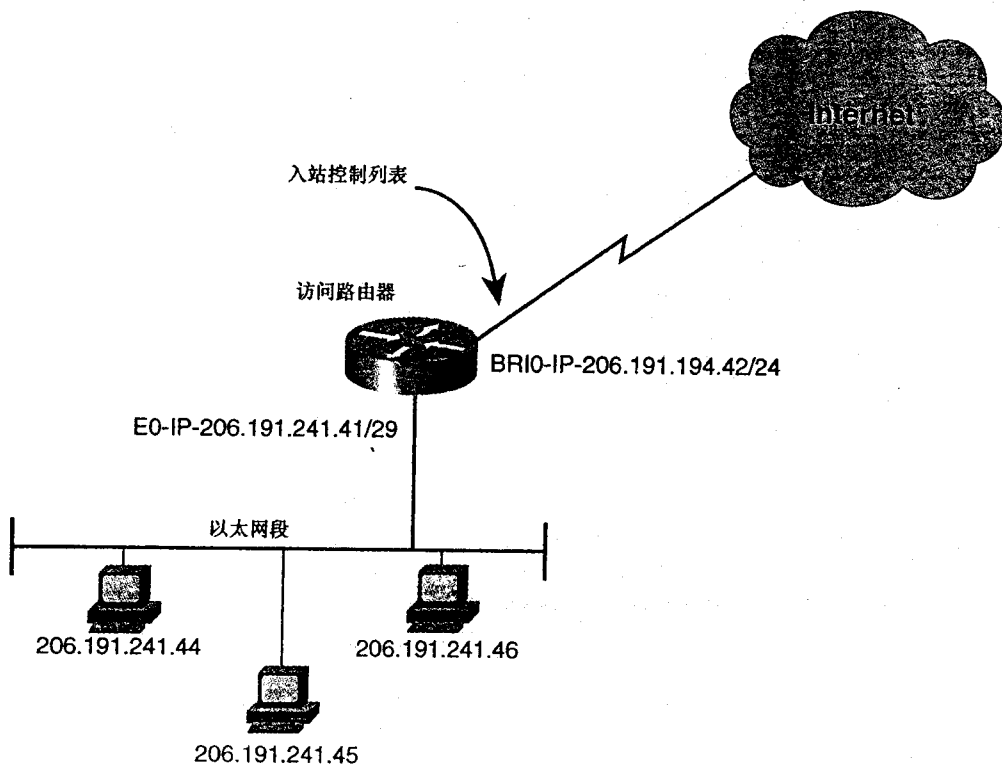


图 14-3 扩展访问控制列表的示例

例 14-7 在路由器 *access_router* 上运用扩展访问控制列表

```

access_router(config)#access-list 199 permit tcp any any established
access_router(config)#access-list 199 deny ip 206.191.241.40 0.0.0.7 any
access_router(config)#access-list 199 deny ip host 206.191.194.42 host
206.191.194.42
access_router(config)#access-list 199 permit icmp any any echo
access_router(config)#access-list 199 permit icmp any any echo-reply
access_router(config)#access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq www
access_router(config)#access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq smtp
access_router(config)#access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq
domain
access_router(config)#access-list 199 permit udp any 206.191.241.40 0.0.0.7 eq
domain
access_router(config)#access-list 199 deny tcp any 206.191.241.40 0.0.0.7 lt 1024
access_router(config)#access-list 199 deny tcp any 206.191.241.40 0.0.0.7 gt 1023
access_router(config)#access-list 199 permit udp any 206.191.241.40 0.0.0.7 gt 1023
access_router(config)#access-list 199 deny udp any 206.191.241.40 0.0.0.7 gt 50000
access_router(config)#access-list 199 deny udp any 206.191.241.40 0.0.0.7 lt 1024

```

用下面的命令将访问控制列表应用到 BRI 接口：

```

access_router (config) #int bri 0
access_router (config-if) #ip access-group 199 in

```

例 14-7 中，访问控制列表的第 1 行调用了关键字 **established**，路由器开始在 TCP 报头中找寻确认 (ACK) 或重置 (RESET) 位。这样做就使得已建立的数据流通过访问控制列表

时产生匹配。这个关键字可以在用于任何交互式 TCP 数据流中，如 WWW。

例 14-7 中的 2、3 行用来对付电子欺骗攻击。第 2 行拒绝源于子网 206.191.241.40/29 的 IP 数据，这个子网是分配给以太网端口，不接受来自与以太网段以外同处一个源地址范围的外部网络的任何数据包。第 3 行防止 BRI 端口上的电子欺骗攻击。扩展访问控制列表的下面两行允许 ICMP 探测信号和对探测信号的应答，即允许对该网络的 ping 的进出操作。如果不能确定哪些位是有效的位，最好将地址展开为二进制的形式。例 14-6 接下去的几行都非常直观。接下去的几行是：

```
access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq www
access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq smtp
access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq pop3
access-list 199 permit tcp any 206.191.241.40 0.0.0.7 eq domain
access-list 199 permit udp any 206.191.241.40 0.0.0.7 eq domain
```

第 1 项允许任何源地址或网络的 TCP 可以去往子网 206.191.241.40，包括 41，42，43，44，45 和 46，前提是所访问的 TCP 端口是 HTTP 或 WWW 所用的端口 80。第 2，3 行允许的是简单邮件传输协议 (SMTP) 从端口 25 以及 POP3 邮件从端口 110 对同一子网区域的入站访问。最后还有两个 DNS 的列表项。一个是允许 TCP 端口 53 上的 DNS 操作，另一个则是允许 UDP 端口 53 上的 DNS 访问，这是 DNS 服务更为常用的一种传输方式。

例 14-6 中列表项的最后几行如下：

```
access-list 199 deny tcp any 206.191.241.40 0.0.0.7 lt 1024
access-list 199 deny tcp any 206.191.241.40 0.0.0.7 gt 1023
access-list 199 permit udp any 206.191.241.40 0.0.0.7 gt 1023
access-list 199 deny udp any 206.191.241.40 0.0.0.7 gt 50000
access-list 199 deny udp any 206.191.241.40 0.0.0.7 lt 1024
```

这里的第 1 行拒绝了从任何源网络到特定子网 206.191.241.40/29 的访问小于 TCP 数据端口 1024 的数据包。第 2 行除了端口是大于 1023 之外其他与第 1 行一样。下一行允许大于 1023 的 UDP 端口上的数据通过，而下两行则是分别拒绝了访问大于 50000 和小于 1024 的 UDP 端口的数据。基本上来说，这一个列表项子集过滤众所周知的 UDP TCP 端口数据。其实隐含的 deny any 就可以过滤所有的端口数据，这些列表项都是多余的。但是，有时候能够亲眼在配置中看到某些端口被过滤，又能在有人访问时看到记录内容，比一个看不见的 deny any 要可取得多。在这个例子中，还可以用 range 参数包含过滤的端口。例如，下面两行可以合并为一行：

```
access-list 199 deny tcp any 206.191.241.40 0.0.0.7 lt 1024
access-list 199 deny tcp any 206.191.241.40 0.0.0.7 gt 1023
```

合并为：

```
access-list 199 deny tcp any 206.191.241.40 0.0.0.7 range 1 65535
```

警告 命令 access-list 的增强功能出于向后兼容的考虑，从早期版本升级到 11.1 会自动地对访问控制列表进行转换。11.1 以前版本不向上兼容，没有这些增强功能。如果在 11.1 以后的 Cisco IOS 环境中保存了访问控制列表，然后又改用了 11.1 以前的 IOS，就会导致访问控制列表无法正确得到解释执行，产生严重的安全问题。

换句话说，11.1 以及更新版本的 Cisco IOS 的访问控制列表向前、向后都兼容。11.1 以前的则既不向前兼容也不向后兼容。因此，在不同的 Cisco IOS 版本环境中，或者是由于某种原因导致 IOS 降级或升级的情况下，访问控制列表有可能失效。

14.5 访问控制列表的显示

现在，大家可能想知道如何显示访问控制列表的信息，或者是怎样查找和排除关于访问控制列表的故障。要查看访问控制列表，可以在 **enable** 提示符下使用下面这些命令：

- **show access-list**——显示所有协议的访问控制列表情况，并包含匹配某访问控制列表的每一行的数据包的数量。用 **clear access-list counter** 命令可以清除这些计数器的内容。
- **show ip access-list [access-list number]**——显示定义的所有 IP 访问控制列表。如果指定某个特定的访问控制列表，那就只会显示该列表的信息，并显示匹配该访问控制列表每一行的数据包的数量。用 **clear access-list counter** 命令可以清除这些计数器的值。
- **show log**——这条命令是和 **log** 关键字配合使用来跟踪任意访问控制列表的。记住要在配置中用一条 **logging buffered** 命令来捕捉所有的控制台消息，所记录的信息包括访问控制列表号，允许还是拒绝数据包的记录，所用的协议以及源地址和目的地址。为了避免出现庞大的记录文件，路由器只为第一个匹配的数据包产生这样的消息，以后每隔 5 分钟才产生一次，并包含前一个 5 分钟时内允许或拒绝的数据包的数量。

例 14-8 是 **show ip access-list** 命令的输出示例。

例 14-8 **show ip access-list** 命令的执行示例

```
access_router# show ip access-lists
Standard IP access list 69
  permit 206.191.241.0, wildcard bits 0.0.0.255 log
Extended IP access list 101
  deny udp host 172.16.16.2 host 204.221.151.211 eq domain
  permit tcp any any established (15992 matches)
  permit ip any 192.168.5.0 0.0.0.255 (43 matches)
  permit ip any 204.221.151.0 0.0.0.255 (169 matches)
  permit icmp any any echo (78 matches)
  permit icmp any any echo-reply (9 matches)
  permit tcp any any eq www (216 matches)
  permit udp any any
Extended IP access list 110
  permit ip any any (37779 matches)
  permit tcp any any established
Extended IP access list 199
  permit tcp any any established (175 matches)
  deny ip 206.191.241.40 0.0.0.7 any
  deny ip host 206.191.194.42 host 206.191.194.42
  permit icmp any any echo
  permit icmp any any echo-reply
  permit tcp any 206.191.241.40 0.0.0.7 eq www
  permit tcp any 206.191.241.40 0.0.0.7 eq smtp
  permit tcp any 206.191.241.40 0.0.0.7 eq domain
  permit udp any 206.191.241.40 0.0.0.7 eq domain
  deny tcp any 206.191.241.40 0.0.0.7 lt 1024
```

```
deny tcp any 206.191.241.40 0.0.0.7 gt 1023
permit udp any 206.191.241.40 0.0.0.7 gt 1023 (13 matches)
deny udp any 206.191.241.40 0.0.0.7 gt 50000
deny udp any 206.191.241.40 0.0.0.7 lt 1024
access_router#
```

14.6 动态访问控制列表

动态访问控制列表能够允许用户在通过路由器的认证之后进行临时性的访问。例如，实际应用中可能需要 Cisco 的 TAC 工程师登录到路由器中帮助你解决网络出现的问题。动态访问控制列表可以给这些工程师提供预定时间长度的特权访问。经过设置的时间之后，该特权就会过期，会话也会关闭，数据访问同时被拒绝。这种形式的访问控制列表也称为 **锁定与解锁安全 (lock-and-key security)**。

配置动态访问控制列表，需要经过以下这些步骤：

第 1 步 定义一个用户名和密码。

第 2 步 定义用户名时加上 **autocommand** 参数和 **timeout** 参数；这些参数必须与动态访问控制列表中指定的超时值匹配。

第 3 步 定义一个仅有一行的动态访问控制列表，指定需要在用户通过认证之后才能传输数据。这一行也必须包含一个超时设置值，而且应该和上面设置的超时相匹配。

第 4 步 定义一个扩展访问控制列表，其范围应该和动态访问控制列表一样，用这个列表对接口进行常规数据包的过滤。其中必须允许该接口的 Telnet 访问，因为这是用于 Telnet 认证的。最后，把这份访问控制列表应用于某个接口。

第 5 步 将 **login local** 加到 vty 线路号中去，这些线路信息可以通过 **show line** 命令来显示。

回想一下前面例子里的那个网络，现在你已经知道了怎样清除所有的访问控制列表，允许任意可达性的路由方式。在图 14-4 中，每台路由器的路由表中都含有子网 128.200.0.0，这里有完全可达性。

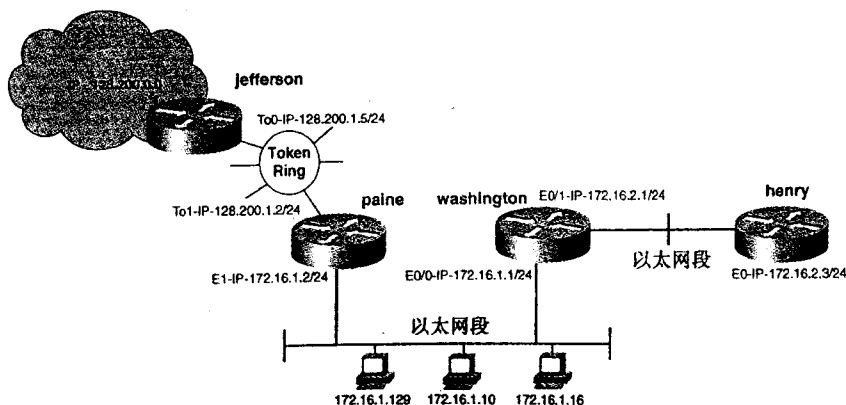


图 14-4 动态访问控制列表的实例

这个例子里，要在 paine 路由器的以太端口 0 上定义一份动态访问控制列表，只允许子网 172.16.1.0/24 中的用户进行认证，认证通过之后允许访问子网 128.200.0.0。访问控制列表要禁止任何未经认证的用户数据进入该接口。通过认证之后，允许该用户访问子网 172.16.1.0/24 中的所有设备，访问时间长度是 5 分钟，然后动态访问控制列表就要关闭。

要完成上面这些任务，首先，设置认证用的用户名和密码：

```
username franklin password ben
username franklin autocommand access-enable timeout 5
```

上面的第 2 行说明用户 franklin 登录进来之后特殊的 autocommand 命令会被执行。而 access-enable 则也是一条特殊的命令，用 ? 符号（参数帮助）也看不到这条命令。要记住这条命令。这里的 timeout 值是一个空闲超时值，把它设为 5 分钟，这样，如果访问控制列表 5 分钟没有检测到数据，就会自动关闭。

接下来，定义动态访问控制列表：

```
access-list 101 dynamic allowben timeout 5 permit ip 172.16.1.0 0.0.0.255 any
access-list 101 permit tcp 172.16.1.0 0.0.0.255 host 172.16.1.2 eq telnet
```

访问控制列表的名称必须惟一，可以设定为任意名。重要的是 timeout 的值，这是一个绝对的超时值。如果使用了两个计时器，空闲超时的值要等于或者低于这个绝对超时的值。访问控制列表中其他项允许子网 172.16.1.0/24 的 IP 数据在通过认证之后可以访问任意网络。

下面是 Cisco 关于配置动态访问控制列表的一些规则和建议：

- 对于超时的设置，既可以通过 autocommand 命令的子命令 access-enable 中的关键字 timeout 来定义，也可以在后面用 access-list 命令来定义一个绝对超时值。空闲超时或绝对超时是必须定义的，否则，临时性的访问控制列表就会无限期地常驻在接口上（即使用户终止会话），除非系统管理员手动地将其删除掉。
- 如果要配置一个空闲超时，其值应该等于拨号的空闲超时值。
- 如果既要配置空闲超时，又要配置绝对超时，空闲超时的值应该小于或等于绝对超时的值。

下一行是常规的访问控制列表项，这一项一直在发挥作用，直到有用户通过了认证。这里的访问控制列表必须以一条 Telnet 的允许项开始，也就是必须在使用访问控制列表的接口上允许 Telnet 的访问，否则用户就没有办法进行认证。这个例子中，只允许子网 172.16.1.0/24 中的用户可以进行认证，拒绝所有其他数据。可以把这份访问控制列表应用到路由器 paine 的以太端口 0 上，这是通过在这个接口上执行 ip access-group 101 in 命令来实现的。

最后要作的是在 vty 端口上允许 Telnet 的访问以及设置适当的密码。可以参考第 1 章回顾绝对线路号的内容。

例 14-9 是路由器 paine 上的配置情况。

例 14-9 路由器 paine 的配置

```
hostname paine
!
enable password 7 02050D480809
!
username franklin password 7 02040155
```

```

username franklin autocommand access-enable timeout 5
!
!
interface Ethernet0
no ip address
shutdown
media-type 10BaseT
!
interface Ethernet1
ip address 172.16.1.2 255.255.255.0
ip access-group 101 in
media-type 10BaseT
!
<<<text omitted>>>
!
interface TokenRing1
ip address 128.200.1.2 255.255.255.0
ring-speed 16
!
router eigrp 2001
network 128.200.0.0
network 172.16.0.0
no auto-summary
!
ip classless
!
access-list 101 dynamic allowben timeout 5 permit ip 172.16.1.0 0.0.0.255 any
access-list 101 permit tcp 172.16.1.0 0.0.0.255 host 172.16.1.2 eq telnet
!
!
line con 0
line aux 0
line vty 0 4
login local
!
end

```

要对这些配置进行测试，可以将一台工作站连接到 172.16.1.0/24 的以太网段，或者是利用路由器 *washington* 来完成。该路由器总是把距离目的地址最近的地址作为 IP 数据包的源地址。如果要用其他端口作为 Telnet 会话的源，可以使用这条命令：

```
ip telnet source-interface interface_name
```

例 14-10 先给出了从 *washington* 到 *jefferson* 的一次不成功的 ping 操作的例子。接着，用户在 *paine* 上进行认证，然后就有了到 *jefferson* 的成功 ping 操作。5 分钟以后，路由器 *paine* 关闭临时通道，拒绝后面对以太端口的入站访问。注意 Telnet 在认证之后立即关闭的情况，这是常规的操作过程。

例 14-10 测试动态访问控制列表

```

washington#
washington#ping 128.200.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 128.200.1.5, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

```

(待续)


```

washington#
washington#
washington#telnet 172.16.1.2
Trying 172.16.1.2 ... Open

User Access Verification

Username: franklin
Password:
[Connection to 172.16.1.2 closed by foreign host]
washington#
washington#
washington#ping 128.200.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 128.200.1.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
washington#
<<<After 5 minutes expires>>>
washington#ping 128.200.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 128.200.1.5, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
washington#

```

14.7 命名访问控制列表

随着 Cisco IOS 11.2 的出现，Cisco 引入了用惟一的名字标示访问控制列表的方法。命名访问控制列表允许对访问控制列表进行标识时使用一些描述性的名称，而不是像以前那样仅有一个无描述作用的号码。这一改进对需要使用大量访问控制列表的网络管理员来说真是受益匪浅的。

配置命名访问控制列表，首先用下面的命令将访问控制列表定义为标准或扩展型：

```
ip access-list {standard | extended} access_list_name
```

输入这一行之后，路由器会提示输入控制列表的条目。标准访问控制列表的提示需要输入的句法如下：

```
{permit | deny} a.b.c.d [ wildcard_mask]
```

扩展访问控制列表的句法为：

```
{permit | deny} protocol_type source_address source_address_wildcard
destination_address destination_address_wildcard [ protocol specific options] {log}
```

除了常规访问控制列表中每一行前的列表号之外，所有的规则、命令句法对这两种情况仍然适用。图 14-5 是应用于以太接口的一个简单的命名访问控制列表的例子。

图 14-5 的例子中，一个名为 **allow_net_172** 的命名访问控制列表应用于以太 0/0 端口上。把命名访问控制列表应用到某个接口的命令是 **access-group**，使用的参数是访问控制列表的名称 **name** 而不是号码。

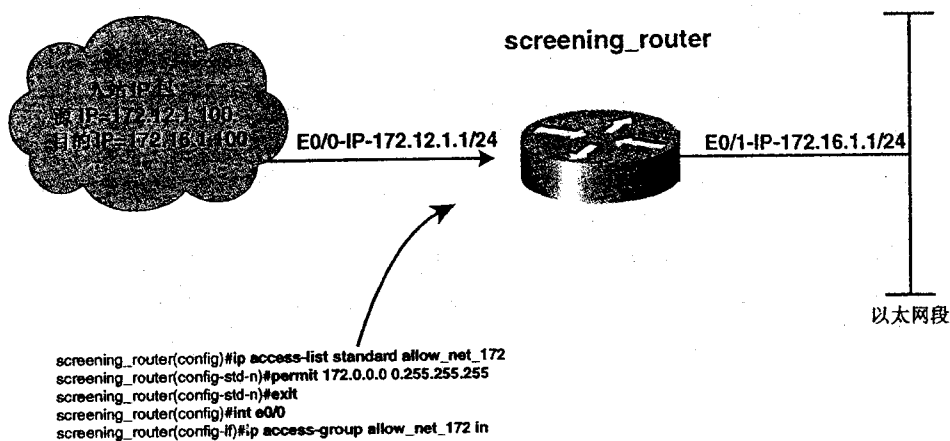


图 14-5 命名访问控制列表的例子

14.8 实验 29：配置访问控制列表、命名访问控制列表以及 EIGRP 路由过滤——第 1 部分

14.8.1 实验说明

本章再三强调理解二进制访问控制列表的重要性。现在对此加以练习。在实际应用中，访问控制列表要做得尽量的短而高效。

14.8.2 实验内容

假设州巡警与郡司法部门（State Patrol and the County Sheriff）都需要利用 FBI 总部的全国指纹查询系统。由于该系统使用需求的巨大增长，FBI 希望降低可以使用该网络的用户的数量。FBI 现在规定它们的网络允许那些偶数子网的州巡警站点和奇数子网的郡司法站点访问。而州巡警站点和郡司法站点之间还有相互重叠的子网部分，因此执行这一规定时要非常小心。设计该网络时要遵循下面的要求：

- 整个网络的路由选择协议采用 EIGRP，自治系统 ID 是 2001。
- 对路由更新进行控制，使得路由器 FBI_hq 只接收来自路由器 State Patrol 的偶数子网和路由器 County Sheriff 的奇数子网。
- 图中的两个云图代表与路由器相连的 IP 网络，这些网络是通过我们所创建的路由源来模拟。
- 使用命名访问控制列表。

14.8.3 实验目的

- 按照图 14-6 对网络进行配置，路由选择协议采用 EIGRP。
- 只允许子网号为偶数的路由更新可以从路由器 `state_patrol` 上宣告，同时，只允许子网号为奇数的路由更新可以从路由器 `county_sheriff` 宣告。
- 以尽量少的命令行创建访问控制列表。
- 通过从路由器 `fbi_hq` 向地址 150.100.2.1 执行 `trace` 命令来对网络进行检查，这条命令的信息应该能够到达路由器 `state_patrol`。另外再通过观察路由器报告了那些路由条目来检查路由是否异常。

14.8.4 所需设备

- 3 台 Cisco 路由器通过 V.35 背对背线缆或类似方式连接在一起。
- 通过集线器或交换机构建一个 LAN 网段。

14.8.5 物理设计与实验准备

- 按照图 14-6 将集线器以及串行线缆与路由器连接起来。
- 按照图 14-6 模拟一个 LAN 网段。

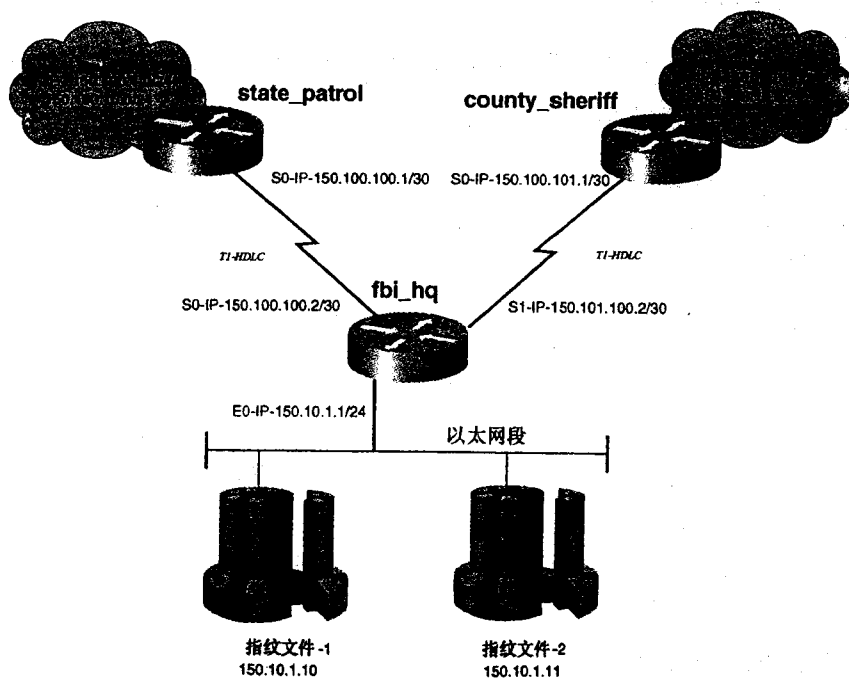


图 14-6 全国指纹系统——广域网的访问

- 在路由器 `state_patrol` 和 `county_sheriff` 上配置路由源，这通过在路由器上配置 10 个环路接口地址来实现，两台路由器采用同一个地址范围——从 150.100.1.0/24 到 150.100.10.0/24。
- 路由选择协议采用 EIGRP。

14.9 实验 29：配置访问控制列表、命名访问控制列表以及 EIGRP 路由过滤——第 2 部分

14.9.1 实验步骤

物理连接完成之后，在所有的路由器之间建立 IP 连接。这时不必再使用 `ping` 测试路由器的环路地址。在进行过滤之前，网络中存在着路由环路。

首先来看看路由器 `fbi_hq`，为以太接口和两个串行接口配置 IP 地址。这是在配置链路的 DCE 端，因此需要在串行接口上用 `clock rate` 命令设置通信速率。先配置路由器 `state_patrol`，从路由器 `state_patrol` 上可以 `ping` 通 `fbi_hq` 的串行接口之后，开始配置 EIGRP。这里应该能看到各个子网的详细信息，因此应该在 EIGRP 配置中加入 `no autosummary` 命令。将路由器 `state_patrol` 配置成路由源时，先利用 Windows 中的记事本制作与类似下面所示内容的文件：

```
int loop 20
ip add 150.100.1.1 255.255.255.0
int loop 21
ip add 150.100.2.1 255.255.255.0
int loop 22
ip add 150.100.3.1 255.255.255.0
int loop 23
ip add 150.100.4.1 255.255.255.0
int loop 24
ip add 150.100.5.1 255.255.255.0
int loop 25
ip add 150.100.6.1 255.255.255.0
int loop 26
ip add 150.100.7.1 255.255.255.0
int loop 27
ip add 150.100.8.1 255.255.255.0
int loop 28
ip add 150.100.9.1 255.255.255.0
int loop 29
ip add 150.100.10.1 255.255.255.0
```

将这些内容剪切粘贴到文件中去比手工键入要快捷得多。配置完成且路由器 `fbi_hq` 显示路由之后，就可以以同样的方式继续配置路由器 `local_sheriff`。这一配置完成后，路由器 `fbi_hq` 的路由表如例 14-11 所示。

例 14-11 路由器 fbi_hq 上 show ip route 命令示例

```
fbi_hq# 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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

150.10.0.0/24 is subnetted, 1 subnets
C      150.10.1.0 is directly connected, Ethernet0
150.100.0.0/16 is variably subnetted, 12 subnets, 2 masks
C      150.100.100.0/30 is directly connected, Serial0
C      150.100.101.0/30 is directly connected, Serial1
D      150.100.2.0/24 [90/2297856] via 150.100.100.1, 00:00:07, Serial0
          [90/2297856] via 150.100.101.1, 00:00:07, Serial1
D      150.100.3.0/24 [90/2297856] via 150.100.100.1, 00:00:07, Serial0
          [90/2297856] via 150.100.101.1, 00:00:07, Serial1
D      150.100.1.0/24 [90/2297856] via 150.100.100.1, 00:00:07, Serial0
          [90/2297856] via 150.100.101.1, 00:00:07, Serial1
D      150.100.6.0/24 [90/2297856] via 150.100.100.1, 00:00:07, Serial0
          [90/2297856] via 150.100.101.1, 00:00:07, Serial1
D      150.100.7.0/24 [90/2297856] via 150.100.100.1, 00:00:07, Serial0
          [90/2297856] via 150.100.101.1, 00:00:07, Serial1
D      150.100.4.0/24 [90/2297856] via 150.100.100.1, 00:00:08, Serial0
          [90/2297856] via 150.100.101.1, 00:00:08, Serial1
D      150.100.5.0/24 [90/2297856] via 150.100.100.1, 00:00:08, Serial0
          [90/2297856] via 150.100.101.1, 00:00:08, Serial1
D      150.100.10.0/24 [90/2297856] via 150.100.100.1, 00:00:08, Serial0
          [90/2297856] via 150.100.101.1, 00:00:08, Serial1
D      150.100.8.0/24 [90/2297856] via 150.100.100.1, 00:00:09, Serial0
          [90/2297856] via 150.100.101.1, 00:00:09, Serial1
D      150.100.9.0/24 [90/2297856] via 150.100.100.1, 00:00:09, Serial0
          [90/2297856] via 150.100.101.1, 00:00:09, Serial1
```

请注意，现在有两个路由源向路由器 fbi_hq 发送同样的路由。如果只是用了 ping 命令来测试，可能会得出一切正常的结论。但是如果从路由器 fbi_hq 的以太接口进行一次源地址 trace，会发现路由存在问题。例 14-12 是执行源 trace 和 ping 的结果。

例 14-12 在路由器 fbi_hq 上执行 trace 和 ping 命令的结果

```
fbi_hq# ping 150.100.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.100.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
fbi_hq# trace
Protocol [ip]:
Target IP address: 150.100.1.1
Source address: 150.10.1.1
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 150.100.1.1
```

```
  1 150.100.100.1 4 msec
    150.100.101.1 4 msec
    150.100.100.1 8 msec
```

现在再来看看路由器 `state_patrol` 的配置，按照要求，创建一份命名访问控制列表，只允许偶数子网被宣告到路由器 `fbi_hq`。这些子网包括 150.100.0.0 网络的 0、2、4、6、8 和 10 子网。将 1 到 10 的数写成二进制，结果如下：

```
0000 0001 = 1
0000 0010 = 2
0000 0011 = 3
0000 0100 = 4
0000 0101 = 5
0000 0110 = 6
0000 0111 = 7
0000 1000 = 8
0000 1001 = 9
0000 1010 = 10
```

可见，所有的偶数子网的右边第一位都是 0。因此在访问控制列表中指定第 3 个 8 位字节的第 1 位必须为 0。例 14-13 是如何利用这些参数配置访问控制列表的例子。由于要求访问控制列表第 1 部分中的 3 个 8 位字节的第 1 位必须是 0，因此通配符掩码就是 0.0.254.255。

例 14-13 只允许偶数子网的命名访问控制列表

```
state_patrol(config)#ip access-list standard alloweven
state_patrol(config-std-nacl)#permit 150.100.0.0 0.0.254.255
state_patrol(config-std-nacl)#exit
state_patrol(config)#router eigrp 2001
state_patrol(config-router)#distribute-list alloweven out s0
state_patrol(config-router)#^Z
```

现在开始对路由器 `local_sheriff` 进行配置，情况和上一台路由器类似，按要求，该路由器转发的子网只有奇数部分才能宣告到路由器 `fbi_hq` 去。按照与例 14-13 中访问控制列表同样的道理，将访问控制列表中源地址的第 3 个 8 位字节的右边第 1 位置为 1。掩码仍然是 0.0.254.255，说明该位必须是 1。例 14-14 是路由器 `local_sheriff` 的配置示例。

例 14-14 只允许奇数子网的命名访问控制列表

```
county_sheriff(config)#ip access-list standard allowodd
county_sheriff(config-std-na)#permit 150.100.1.0 0.0.254.255
county_sheriff(config-std-na)#exit
county_sheriff(config)#router eigrp 2001
county_sheriff(config-router)#distribute-list allowodd out s0
county_sheriff(config-router)#^Z
county_sheriff#
```

为了测试最后的配置情况，在路由器 `fbi_hq` 上执行 `show ip route` 命令和 `trace`。例 14-15

达了接 1 口 Serial 0，而来自 150.100.101.1 的子网则只有奇数部分才到达了接口 Serial 1。

例 14-15 路由器 fbi_hq 上 show ip route 和 trace 命令执行示例

```
fbi_hq# 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, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

150.10.0.0/24 is subnetted, 1 subnets
C      150.10.1.0 is directly connected, Ethernet0
150.100.0.0/16 is variably subnetted, 12 subnets, 2 masks
C      150.100.100.0/30 is directly connected, Serial0
C      150.100.101.0/30 is directly connected, Serial1
D      150.100.2.0/24 [90/2297856] via 150.100.100.1, 00:01:35, Serial0
D      150.100.3.0/24 [90/2297856] via 150.100.101.1, 00:01:30, Serial1
D      150.100.1.0/24 [90/2297856] via 150.100.101.1, 00:01:30, Serial1
D      150.100.6.0/24 [90/2297856] via 150.100.100.1, 00:01:35, Serial0
D      150.100.7.0/24 [90/2297856] via 150.100.101.1, 00:01:30, Serial1
D      150.100.4.0/24 [90/2297856] via 150.100.100.1, 00:01:35, Serial0
D      150.100.5.0/24 [90/2297856] via 150.100.101.1, 00:01:30, Serial1
D      150.100.10.0/24 [90/2297856] via 150.100.100.1, 00:01:35, Serial0
D      150.100.8.0/24 [90/2297856] via 150.100.100.1, 00:01:35, Serial0
D      150.100.9.0/24 [90/2297856] via 150.100.101.1, 00:01:31, Serial1
fbi_hq#
fbi_hq#trace
Protocol [ip]:
Target IP address: 150.100.1.1
Source address: 150.10.1.1
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 150.100.1.1

  1 150.100.101.1 0 msec 0 msec *
fbi_hq#
```

例 14-16 列出了路由器 state_patrol, county_sheriff 和 fbi_hq 的完整配置清单。

例 14-16 路由器 state_patrol、county_sheriff 和 fbi_hq 的完整配置清单

```
hostname state_patrol
!
ip subnet-zero
!
interface Loopback20
ip address 150.100.1.1 255.255.255.0
no ip directed-broadcast
!
```

```
interface Loopback21
 ip address 150.100.2.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback22
 ip address 150.100.3.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback23
 ip address 150.100.4.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback24
 ip address 150.100.5.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback25
 ip address 150.100.6.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback26
 ip address 150.100.7.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback27
 ip address 150.100.8.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback28
 ip address 150.100.9.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback29
 ip address 150.100.10.1 255.255.255.0
 no ip directed-broadcast
!
<<<text omitted>>>
!
interface Serial0
 ip address 150.100.100.1 255.255.255.252
 no ip directed-broadcast
!
<<<text omitted>>>
!
router eigrp 2001
 network 150.100.0.0
 distribute-list alloweven out Serial0
 no auto-summary
!
ip access-list standard alloweven
 permit 150.100.0.0 0.0.254.255

hostname county_sheriff
!
ip subnet-zero
!
interface Loopback20
 ip address 150.100.1.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback21
 ip address 150.100.2.1 255.255.255.0
```

(待续)


```

no ip directed-broadcast
!
interface Loopback22
ip address 150.100.3.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback23
ip address 150.100.4.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback24
ip address 150.100.5.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback25
ip address 150.100.6.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback26
ip address 150.100.7.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback27
ip address 150.100.8.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback28
ip address 150.100.9.1 255.255.255.0
no ip directed-broadcast
!
interface Loopback29
ip address 150.100.10.1 255.255.255.0
no ip directed-broadcast
!
<<<text omitted>>>
!
interface Serial0
ip address 150.100.101.1 255.255.255.252
no ip directed-broadcast
!
<<<text omitted>>>
!
router eigrp 2001
network 150.100.0.0
distribute-list allowodd out Serial0
no auto-summary
!
ip access-list standard allowodd
permit 150.100.1.0 0.0.254.255

hostname fbi_hq
!
interface Ethernet0
ip address 150.10.1.1 255.255.255.0
!
interface Serial0
ip address 150.100.100.2 255.255.255.252
no fair-queue
clockrate 2000000
!
interface Serial1
ip address 150.100.101.2 255.255.255.252
clockrate 2000000

```

```

!
<<<text omitted>>>
!
router eigrp 2001
 network 150.10.0.0
 network 150.100.0.0
 no auto-summary

fbi_hq#
    
```

14.10 实验 30：利用命名访问控制列表配置动态访问控制列表和数据过滤——第 1 部分

14.10.1 实验说明

随着 Internet 和 Intranet 的增长，各种应用要求对网络访问进行控制。控制访问的最佳办法是在路由选择协议内不通告私有子网。但有时又常常需要访问 IP，因而不得不将网络通过路由选择协议转发。在这种情况下，可以采用访问控制列表在数据包级别上对访问加以控制。

14.10.2 实验内容

假设新近出现的一家公司 Wavester.com 向用户提供可以通过安全 FTP 和 TFTP 来访问的庞大 MP3 库。很多大学生都深受 Internet 访问费用之苦。让这些大学生们高兴的是，Wavester.com 现在为用户提供了直接的 T1 访问方式。在这个实验中，是要定义一条与 Wavester.com 站点相连的 T1 链路。可以在链路上通过的协议包括 FTP、TFTP、ping 和路由选择协议。按照下面的要求将这个过滤功能安排到链路最为有效的位置上：

- 路由选择协议采用 OSPF，所有新建的站点都要配置成存根区域。
- 对数据进行控制，保证串行链路上只能通过 FTP，TFTP 和 ping 数据包。只允许子网 132.31.5.16/27 通过 FTP 访问服务器 150.10.1.10。
- 使用命名访问控制列表。
- 配置访问控制列表，使用户在通过 wavester 路由器的认证之前无法对路由器 graceland 进行 Telnet 访问，如果通过了认证，也只允许子网 132.31.5.16/27 对其进行访问。

14.10.3 实验目的

- 按照图 14-7 对网络进行配置，路由选择协议采用 OSPF，路由器 jo_college 配置在存根区域中。
- 串行链路上只允许 Telnet、FTP、TFTP、ping 和路由选择协议通过，而 FTP 的访问则只允许访问服务器 150.10.1.10。

- 在 wavester 上再配置一份访问控制列表，禁止子网 132.31.5.16 对路由器 graceland 的 Telnet 访问。当用户 theking 以密码 elvis 在路由器 wavester 上进行认证时，允许子网 132.31.5.16/27 对路由器 graceland 的 Telnet 访问。并在用户登录 10 分钟之后中止连接。

14.10.4 所需设备

- 3 台 Cisco 路由器，其中 2 台选过 V.35 背对背线缆或类似方式连接在一起。
- 通过集线器或交换机创建两个 LAN 网段。
- 2 台用于测试 FTP 和 TFTP 的文件传输功能的工作站。FTP 和 TFTP 客户机与服务器上的软件可以在 <http://download.cnet.com/> 网站下载。FTP 发送数据的端口和进行初始连接使用的端口不同。将配置的过滤应用到实际的 FTP 客户服务器环境中，能够发现非实际环境中无法发现的各种问题。请记住，路由器发送数据包的源是离目的地址最近的接口，因此需要用工作站来对访问控制列表加以测试。

14.10.5 物理设计与实验准备

- 按照图 14-7 将集线器以及串行线缆与路由器连接在一起。
- 按照图 14-7 创建 LAN 网段。
- 将一台工作站与路由器 wavester 的以太网段相连。该工作站将作为 FTP 和 TFTP 的服务器。另一台工作站则与路由器 jo_college 的以太网段相连作为 FTP 和 TFTP 的客户机。二者所需的软件可以在 <http://download.cnet.com/> 网站下载。

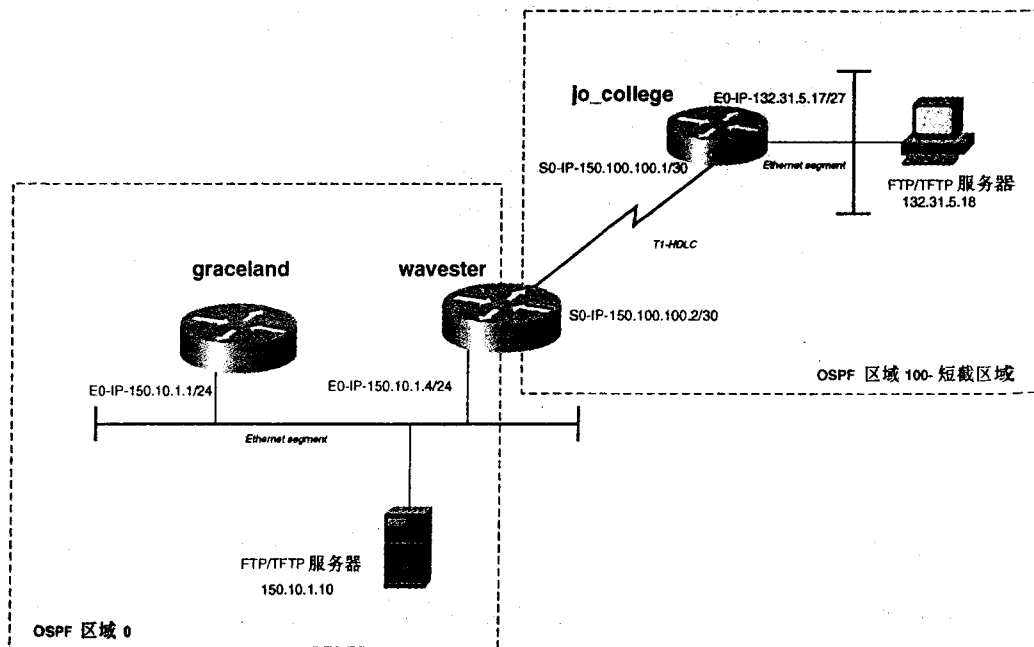


图 14-7 Wavester.com 网络

- 路由器也可以用来测试 TFTP，但是要注意路由器发送数据包的源处，因为这关系过滤的位置。
- 路由选择协议采用 OSPF。将路由器 graceland 和 wavester 放置在 Area 0 中，而 jo_college 则放在存根区域 Area 100 中。在进行过滤之前一定要先确保 IP 连接的畅通。

14.11 实验 30：利用命名访问控制列表配置动态访问控制列表和数据过滤——第 2 部分

14.11.1 实验步骤

物理连接完成之后，先为所有的路由器建立 IP 连接。在路由器 graceland 和 wavester 的以太网接口上配置 OSPF Area 0，路由器 wavester 的串行接口放入 OSPF Area 100 中。利用 **area 100 stub** 命令将 jo_college 配置成存根路由器。在子网 132.31.5.18 和子网 150.10.1.1 之间建立 IP 的全连接后，开始测试文件传输。

在路由器 wavester 的以太网段上将一台工作站配置为 FTP 和 TFTP 服务器，用 FTP 客户机或者是路由器测试其文件传输功能，在继续进行过滤器的配置之前一定要确保文件传输的通畅。FTP 和 TFTP 的服务器与客户机软件可以从 <http://download.cnet.com/> 下载。

首先要配置的访问控制列表是在路由器 jo_college 上。该访问控制列表必须允许 Telnet，FTP，OSPF，TFTP 和 ICMP 数据包通过。将访问控制列表放置在离要过滤的数据最近的地方，也就是 FTP/TFTP 客户机所在处。该命名访问控制列表的情况请参见例 14-17。

例 14-17 IP 命名访问控制列表

```
ip access-list extended allow_filetrans
permit tcp any any established
permit tcp 132.31.5.16 0.0.0.15 any eq telnet
permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.10 eq ftp
permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.10 gt 1023
permit ospf any any
permit udp any any eq tftp
permit icmp any any echo
permit icmp any any echo-reply
```

访问控制列表的第 1 行允许已经建立了的连接的数据包以及 TCP 报头中设置了确认 (ACK) 或重置 (RST) 位的数据包通过。第 2 行是允许来自子网 132.31.5.16 的 Telnet 访问数据包通往任何目的地址。第 2 行的掩码是指定有效位匹配 16 个地址。如果只允许 16 子网的主机数据包通过，就需要对第 4 个字节的前 4 位进行匹配比较：

0001 0000 = 子网 16

0000 1111 = 掩码 = 15

因此，掩码 0.0.0.15 只允许子网 132.31.5.16/27 的数据包通过。

下面 3 行是与 FTP 访问相关的。

```

permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.10 eq ftp
permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.10 gt 1023
permit tcp host 150.10.1.10 132.31.5.16 0.0.0.15 gt 1023

```

FTP 通过 TCP 端口 21 从客户机到服务器建立起会话，但是发送数据则是通过大于 1023 的一个随机端口来进行的。很多人都会产生误解，认为 FTP 是利用 TCP 端口 20 来发送数据的。在配置 FTP 的访问控制列表时，需要指定使用大于 1023 的 TCP 端口来发送数据。

访问控制列表的下一行允许 OSPF 数据包，再下一行则是允许端口等于 69 的 UDP 数据包，即 TFTP 数据包的通过。编写访问控制列表时的另一问题是路由选择协议的过滤。当然，出现这方面问题时，路由表中的所有路由都会消失，因此很容易发现问题的存在。

最后 2 行允许 ICMP 用于存活性探测应答的数据包通过。这两行使得我们依然可以 ping 远端路由器。用下面这条命令将这一访问控制列表应用到以太接口上：

```
ip access-group allow_filetrans in
```

访问控制列表应用之后，可以来测试一下 FTP/TFTP 服务器与客户的文件传输。实际的应用环境是测试任何类型数据过滤的惟一途径。

实验的下一步是在路由器 wavester 上配置动态访问控制列表，用于允许对该路由器的 Telnet 访问和拒绝对路由器 graceland 的 Telnet 访问。当用户 theking 通过认证之后，为路由器 graceland 分配 10 分钟的 Telnet 访问授权。完成这一任务的访问控制列表如例 14-18 所示。

例 14-18 路由器 wavester 上的动态访问控制列表

```

ip access-list extended allowtelnet
dynamic allowking timeout 10 permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.1 eq
telnet
permit tcp 132.31.5.16 0.0.0.15 host 150.101.100.2 eq telnet
deny tcp any host 150.10.1.1 eq telnet
permit ip any any

```

这一份访问控制列表的第 1 行是一条 **dynamic** 命令，允许从子网 132.31.5.16/27 到一个特定主机地址 150.10.1.1 的 Telnet 访问。某个用户认证通过之后，该访问控制列表会将链路为其开放 10 分钟。下一行是允许对路由器 wavester 串行接口的 Telnet 访问，这主要用作认证。再下一行是禁止任何对路由器 graceland 以太端口的 Telnet 访问。最后一行则是允许所有的 IP 数据通过。该访问控制列表通过 **ip access-group allowtelnet in** 命令应用到串行接口上。

动态访问控制列表的第 2 部分是配置用户名、密码以及 **autocommand** 参数。如果使用了两个超时设置值，参数 **autocommand** 超时值必须和访问控制列表中的 **dynamic** 行中的值相匹配。例 14-19 是路由器 wavester 上用户名的组合应用情况。

例 14-19 动态访问控制列表的用户名和密码

```

username theking password elvis
username theking autocommand access-enable timeout 10

```

对 Telnet 访问的配置进行测试之前，记得还要配置路由器的 vty 会话以对 Telnet 提供支持。测试动态访问控制列表时，首先试一下从路由器 jo_college 到 graceland 的 Telnet 访问情况。这个会话应该会遭到拒绝。然后，Telnet 访问路由器 wavester 的串行接口，以 theking 的用户名和 elvis 的密码登录。这一会话应该立即关闭而跳到路由器 jo_college 上。接着，再 Telnet 访问路由器 graceland，应该能够成功登录。

例 14-20 是路由器 wavester 和 jo_college 的完整配置清单。

例 14-20 jo_college 和 wavester 的完整配置清单

```
hostname jo_college
!
enable password cisco
!
username cisco password 0 cisco
ip subnet-zero
!
<<<text omitted>>>
!
interface Ethernet0
ip address 132.31.5.17 255.255.255.240
ip access-group allow_filetrans in
no ip directed-broadcast
!
interface Serial0
ip address 150.100.100.1 255.255.255.252
no ip directed-broadcast
!
<<<text omitted>>>
!
router ospf 69
network 132.31.5.17 0.0.0.0 area 100
network 150.100.100.1 0.0.0.0 area 100
area 100 stub
!
ip classless
!
ip access-list extended allow_filetrans
permit tcp any any established
permit tcp 132.31.5.16 0.0.0.15 any eq telnet
permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.10 eq ftp
permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.10 gt 1023
permit tcp host 150.10.1.10 132.31.5.16 0.0.0.15 gt 1023
permit ospf any any
permit udp any any eq tftp
permit icmp any any echo
permit icmp any any echo-reply
!
line con 0
transport input none
line aux 0
line vty 0 4
end

hostname wavester
!
username theking password 0 elvis
username theking autocommand access-enable timeout 10
clock timezone PAC -8
!
interface Ethernet0
ip address 150.10.1.4 255.255.255.0
!
interface Serial0
ip address 150.100.100.2 255.255.255.252
ip access-group allowtelnet in
no fair-queue
```

(待续)

```
clockrate 2000000
!
<<<text omitted>>>
!
router ospf 69
 network 150.10.1.4 0.0.0.0 area 0
 network 150.100.100.2 0.0.0.0 area 100
 area 100 stub
!
ip classless
!
ip access-list extended allowtelnet
 dynamic allowing timeout 10 permit tcp 132.31.5.16 0.0.0.15 host 150.10.1.1 eq
 telnet
 permit tcp 132.31.5.16 0.0.0.15 host 150.101.100.2 eq telnet
 deny tcp any host 150.10.1.1 eq telnet
 permit ip any any
!
line con 0
line aux 0
line vty 0 4
 login local
!
end
```

第7部分

增强型网络协议

第15章 配置网络地址转换（NAT）

第16章 热备份路由选择协议
（HSRP）的使用

第17章 网络时间协议（NTP）与简单网络时
间协议（SNTP）的配置

第 15 章

配置网络地址 转换 (NAT)

Internet 的快速增长使得可用的 IP 地址空间快速缩小。为了解决这一日益严重的压力，先后产生了一些功能较为强大的技术与方法，像无类域间路由 (CIDR) 和 IPv6 等。CIDR 是 IPv6 成为 IP 的主导版本之前的一个短期解决方案。但是很多专用网络以及 ISP 都还没有将其网络升级到 IPv6，导致这一延迟的原因之一可能是当前应用成功的过渡方案仍然还能解决目前所有问题，这个方案就是网络地址转换 (NAT)。

根据 RFC 1918, NAT 允许私人公司以及个人在自己的网络中采用私有地址空间，这样可以节省大量珍贵的公共地址空间。NAT 还使得同一路由域中的地址能够相互重叠而依然可以访问共同的主机。NAT 提供地址转换的同时也保证了 Internet 上传输的数据所需的一定级别的安全性。本章着重讨论 NAT 的工作原理以及 NAT 的技术术语，同时还包括 NAT 的 3 种应用方式——NAT 池、静态 NAT 和 NAT 过载 (overload)。

15.1 NAT 技术概览

RFC 1631 即“网络地址转换 (NAT)”对 NAT 做了概要描述。NAT 通常安装在存根区域的路由器上，在只有一个出口点的网络中。准确地说，NAT 所做的转换有两种：

- 外部地址转换。
- 内部地址转换。

要理解这些转换的工作方式，首先来看一下 NAT 的技术术语。

15.1.1 NAT 的术语

前面提到的术语对所有的 NAT 配置都适用。所说的内部是指处于路由区域内的专用网络。内部是要转换的地址。而外部通常指为外界路由“可见”的网络。全局这一术语也常和内部、外部一起使用，是由 NIC 分配的地址。表 15-1 是这些术语的一些使用情况。

表 15-1 NAT 的技术术语

NAT 术语	定 义
内部本地地址	分配给主机进行地址转换的 IP 地址。由 RFC1918 提供
内部全局地址	可路由的合法地址，由 NIC 或 ISP 分配。其范围必须在可路由至因特网或目的网络之内
外部本地地址	从内部来看，外部网络主机的 IP 地址，该地址空间从内部可达，但不一定是注册地址空间。主要用于静态转换
外部全局地址	外部网络的 IP 地址，该地址必须在因特网上可路由且可见，属于注册地址，主要用于静态转换

NAT 按下列方法处理内部网络产生的数据包：

- 1 来自内部网络接口的数据包到达 NAT 并且符合 NAT 的转换条件，在 NAT 表中查找某个外部地址转换项，这个匹配项的外部本地地址应该等于数据包目的 IP 地址。
- 2 如果无法找到匹配的外部地址项，将该数据包丢弃。
- 3 如果发现匹配项，NAT 用表中找到的外部全局地址替代数据包中的目的地址。
- 4 NAT 继续在 NAT 表中查找，以确定是否有内部本地地址与该数据包中的源 IP 地址相匹配。
- 5 如果找到这样的匹配项，NAT 用内部全局地址代替数据包中的源地址。
- 6 如果没有找到这样的匹配项，NAT 会自行创建一个新的内部地址项，然后将此地址插入到数据包中去。

NAT 对外部网络产生的数据包的处理：

- 1 来自外部网络接口的数据包到达 NAT 并且通过了转换条件测试之后，NAT 查询地址转换表以找到一个和数据包中的目的地址相同的内部全局地址。
- 2 如果无法找到匹配项，将此数据包丢弃。
- 3 如果找到匹配项，NAT 用转换表中的内部本地地址值代替数据包中的目的地址。
- 4 路由器继续在 NAT 表中查询，以找到一个与数据包中的源 IP 地址相等的外部全局地址。
- 5 如果找到这样的匹配项，NAT 将数据包中的源地址用表中的外部全局地址代替。
- 6 如果没有找到这样的匹配项，NAT 会自行创建一个外部全局地址，然后进行同样的操作。
- 7 每次 NAT 修改报头时，路由器还需要重新计算 IP 和 TCP 的校验和的值，并用新的值代替原来的校验和。

图 15-1 中，私有网络 172.16.1.0/24 上的工作站运行的 TCP 应用程序需要访问公众网络

128.100.1.0/24。

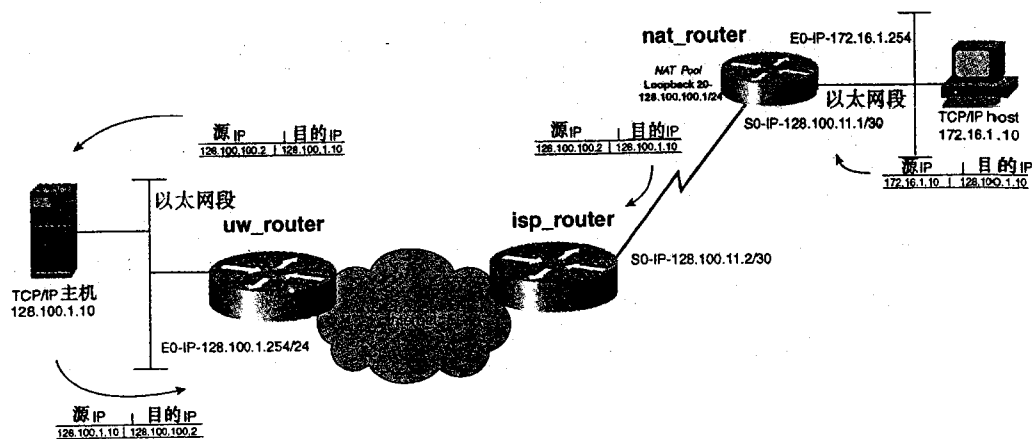


图 15-1 NAT 转换实例

专用网络 172.16.1.0 看不到公共网络 128.100.1.0/24 (UW)。但是，172.16.1.0 有一条路由通到子网 128.100.1.0/24。从主机 172.16.1.10 到 128.100.1.10 的数据包的源 IP 地址是 172.16.1.10，目的地址是 128.100.1.10。数据包进入 NAT 路由器的内部接口 (E0) 后，路由器需要把它路由到一个 NAT 外部接口 (S0)。

这时路由器会根据用户定义的一系列标准来确定是否进行地址的转换，通常情况下是对访问表的检查来实现的。如果要进行地址转换，路由器会根据所配置的 NAT 进行转换。这里的例子采用的网络是一个存有 254 条地址的 NAT 池 128.100.100.0/24。这些地址是内部全局地址。

现在 NAT 在 172.16.1.10 和 128.100.100.2 之间建立关联表，使用地址 128.100.100.2 取代数据包中的源 IP 地址 172.16.1.10，然后通过 S0 端口将数据包转发出去。这个例子中为 NAT 池提供了一个环路接口，其 IP 地址是 128.100.100.1。NAT 会把该子网中下一个可用的地址用于与其他地址建立关联，例子使用 128.100.100.2。UW 路由器接收到数据包时会发现数据好像是来自子网 128.100.100.0/24 的。这个网络是该路由器可以访问的，因此它会对此请求进行服务操作。

NAT 路由器上至少也可以需要一个可以全局访问的 IP 地址，即这个例子中称为内部全局地址的地址。这个地址也可以是地址池里的地址被 NAT 用来替代初始的源 IP 地址。该数据包到达目的地址后，链路的另一端（即目的主机）会认为该数据包来自这个内部全局地址，或地址池。例 15-1 是从 172.16.1.10 ping 128.100.1.10 的结果。使用 `debug ip nat` 命令和 `show ip nat translations` 命令可以看到转换的过程。

例 15-1 NAT 转换的实例

```
nat_router#debug ip nat
00:17:30: NAT*: s=172.16.1.10->128.100.100.2, d=128.100.1.10 [4097]
00:17:30: NAT*: s=128.100.1.10, d=128.100.100.2->172.16.1.10 [4097]
00:17:31: NAT*: s=172.16.1.10->128.100.100.2, d=128.100.1.10 [4353]
00:17:31: NAT*: s=128.100.1.10, d=128.100.100.2->172.16.1.10 [4353]
00:17:32: NAT*: s=172.16.1.10->128.100.100.2, d=128.100.1.10 [4609]
00:17:32: NAT*: s=128.100.1.10, d=128.100.100.2->172.16.1.10 [4609]
```

(待续)

```

00:17:33: NAT*: s=172.16.1.10->128.100.100.2, d=128.100.1.10 [4865]
00:17:33: NAT*: s=128.100.1.10, d=128.100.100.2->172.16.1.10 [4865]
nat_router#
nat_router#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 128.100.100.2      172.16.1.10      ---              ---

```

为方便参考，把这一类的 NAT 称为动态转换，后面有一节会更详细地讨论这一问题。例 15-2 给出的是图 15-1 中路由器 nat_router 的配置情况。

例 15-2 路由器 nat_router 的 NAT 动态池配置

```

hostname nat_router
!
ip subnet-zero
!
interface Loopback20
ip address 128.100.100.1 255.255.255.0
no ip directed-broadcast
!
interface Ethernet0
ip address 172.16.1.254 255.255.255.0
no ip directed-broadcast
ip nat inside
!
interface Serial0
ip address 128.100.11.1 255.255.255.252
no ip directed-broadcast
ip nat outside
!
<<<text omitted>>>
!
router eigrp 2001
network 128.100.0.0
!
ip nat pool publicpool 128.100.100.2 128.100.100.254 netmask 255.255.255.0
!
!nat 128.100.11.1 is not part of the pool since it is the address of the loopback
interface
ip nat inside source list 69 pool publicpool
ip classless
!
access-list 69 permit 172.16.1.0 0.0.0.255
!

```

15.2 NAT 和 RFC 1918

NAT 的大多数特性都是以节省公用 IP 地址空间为目的的。从上例可见，NAT 允许用户使用私有地址空间而同时也可以访问自身路由域以外的网络。RFC1918 (“私有互联网的地址分配”) 预留了一些 ISP 不会使用的地址空间。保证用户在设计网络时不用担心所需的地址是否已经注册过。表 15-2 列出了 RFC 1918 预留的 IP 地址空间的情况。

表 15-2 RFC 1918 分配的 IP 地址空间

IP 地址类	IP 地址范围
A 类	10.0.0.0-10.255.255.255
B 类	172.16.0.0-172.31.255.255
C 类	192.168.0.0-192.168.255.255

注释 应该灵活使用私有地址空间。但是，很多网络的设计都采用 10 开头的 IP 地址空间，点对点链路上使用 24 位掩码，而以太网段则使用 16 位掩码，等等。不要因为 IP 地址空间非常庞大就随意分配 IP 地址。良好的子网划分有助于路由汇总和路由传播，而这两个特点对大型网络有非常重要的影响。很多互连网络也建立在 10 开头的地址空间上。RFC 颁布的那几年里，很多网络都使用 10 开头的子网，这样网络就出现了地址重用。172.16.0.0 这一网段上的上千个网络地址却一直无人使用。这里的意思是建议大家明智地使用地址空间，使用时把这些地址当成已注册地址。设计时要有创意。设计时多花一点时间正确处理 IP 地址的问题，有助于网络扩展或者与其他网络集成。

15.3 NAT 的配置

NAT 转换的配置有 3 种主要方式：

- 动态转换——NAT 将内部地址转换到全局地址池。经过一段时，转换地址超时作废，全局地址又回收至地址池中再次使用。所有 NAT 转换的超时值与配置有关。在“清除和改变 NAT 转换”一节中还会再讨论超时的问。
- 静态转换——NAT 采用一对一的地址映射方式。这种情况下，内部网络会根据 NAT 地址向内部网络发出会话请求。
- 单 IP 地址过载——这种方式通过复用来实现，多个本地 IP 地址利用端口地址转换 (PAT) 共享一个全局 IP 地址。

这 3 种方式的配置过程都具有相同的 4 个步骤：

第 1 步 定义 NAT 的内部网络和外部网络。首先需要定义那些网络要转换。要进行转换的网络不一定是整个的内部网络。还要注意从本地路上查看这一网络的接口。然后，确定路由区域中出口点的位置，即数据包离开路由区域后通过的接口，通常情况下数据包离开后去往 Internet。将这个接口配置为外部 (outside) 接口。这些操作在接口或子接口提示符下运行 `ip nat outside` 命令来完成。

第 2 步 保护目的网络/Internet 与地址池中的地址之间的 IP 连通性。如果配置的是动态或静态转换方式，必须确保外部网络可以访问内部全局网络的子网。内部全局网络是转换成的公共地址。把内部全局网络的子网放到环境接口上，并确保子网通过路由选择协议或静态路由来进行传播，这是为了确保内部网络可以访问该

第3步 需要进行转换的网络进行配置。如果配置动态 NAT 池，可以用这条命令：

```
ip nat inside source [ list {1-99} | route-map] pool pool_name overload
```

然后用访问控制列表或路由图来对所要转换的网络进行匹配过滤。这里要考虑到所有需要通过这个接口的网络，而不单单是考虑本地网络的要求。参数 **pool** 对用于转换的地址池进行定义，而 **overload** 则允许路由器将一个全局地址用于多个本地地址。

如果配置的是静态转换方式，可以使用下面这条命令：

```
ip nat inside source static local_ip_addr global_ip_addr
```

第4步 置地址池。如果使用静态转换方式，则第3步应进行这一步的操作。如果采用动态转换方式，先把全局子网配置到一个环路地址上。例如，如果要把地址转换到子网 150.100.100.0/24，把这个子网放到一个环路地址上去而不是使用实际接口的辅助 IP 地址（secondary address）。这样即使某个接口出现问题，NAT 还可以工作在多个其他的接口上。这种做法也能避免辅助 IP 地址可能产生的某些问题。定义地址池可以采用下面这条命令：

```
ip nat pool pool_name starting_ip_addr ending_ip_addr {netmask netmask | prefix-length prefix-length}
```

15.3.1 NAT 动态转换方式的配置

按照上面的 4 个步骤配置本章前面提到的 NAT 实例。图 15-2 给出了一个私有 IP 网络 172.16.1.0/24。

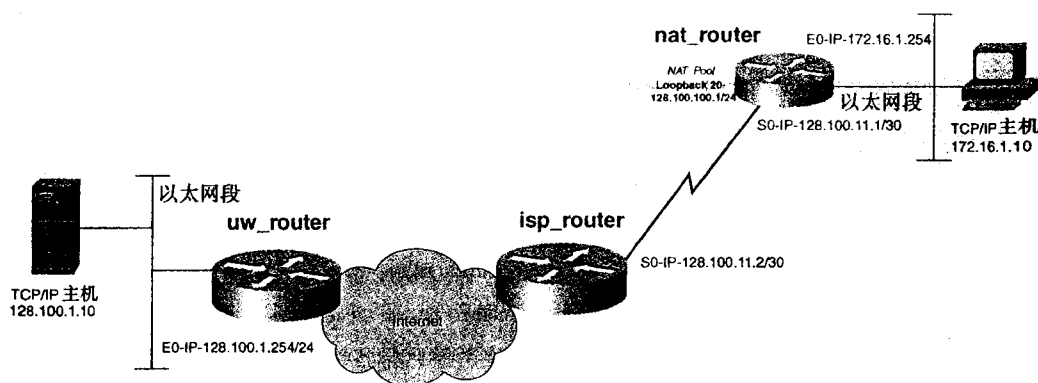


图 15-2 动态 NAT 实例

这个网络需要访问 Internet，确切地说，要访问 UW 以太网段的主机 128.100.1.10。路由器 **nat_router** 通过路由器 **isp_router** 与 Internet 之间有一个 T1 的链路。ISP 将网络 128.100.100.0/24 分配给 **nat_router** 路由器以便它用于访问 Internet。维护路由器 **nat_router** 的工程师不打算去把 172.16.1.x 的主机地址改变成 128.100.100.x 之间的主机地址，因此选用动态 NAT 的方式。

首先定义 NAT 的内部和外部网络。内部网络是需要进行转换的网络，而外部网络是目的网络。这个例子中把端口 E0 作为 NAT 内部接口，而 S0 则作为 NAT 外部接口。图 15-3 突出了显示了内部网络和外部网络。

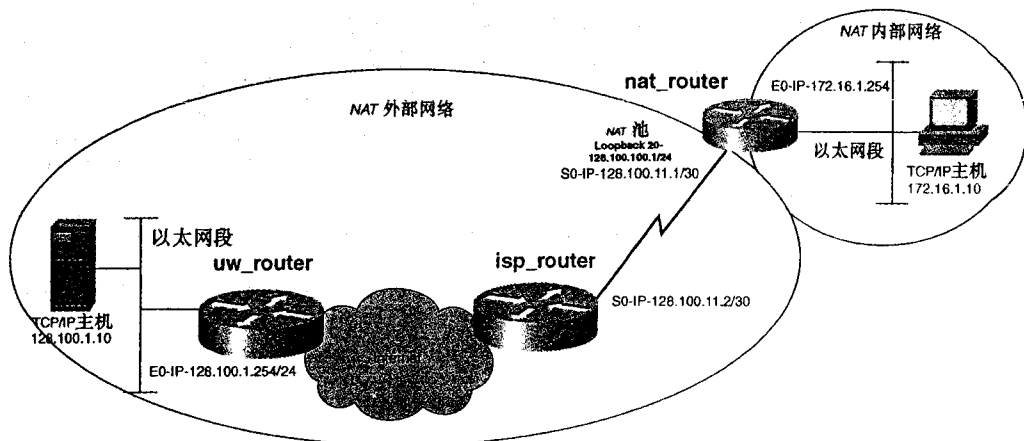


图 15-3 NAT 的内部和外部网络

配置内部接口和外部接口可以用 **ip nat [inside | outside]** 命令。例 15-3 是该命令的用法。

例 15-3 内部接口和外部接口的配置

```
nat_router(config)#interface e0
nat_router(config-if)#ip nat inside
nat_router(config-if)#exit
nat_router(config)#interface s0
nat_router(config-if)#ip nat outside
```

下一步是在路由器上设置转换用的子网，具体做法是把 ISP 分配的网络地址（本例中是 128.100.100.0/24）配置到一个本地接口上。在这里，采用环路接口的第一个主机地址作为这个本地接口的地址，而从 128.100.100.2 到 128.100.100.254 的地址都作为地址池。这一步还要求建立一条通往外部网络的路由和一条外部网络通往子网 128.100.100.0/24 的路由。这个例子中，在路由器 nat_router 上加上一条静态默认路由，还需要配置 **ip classless** 命令，如下：

```
interface Loopback20
ip address 128.100.100.1 255.255.255.0
ip classless
ip route 0.0.0.0 0.0.0.0 128.100.11.2
```

与此例有关但没有显示出来的是路由器 isp_router 到子网 128.100.100.0/24 的静态路由，它必须由 isp_router 路由器宣告到所有的目的网络成员，如 uw_router。这时，在进行下一步的配置之前，要确保所有的路由器都能够访问子网 128.100.100.0/24。没有正常的 IP 互连，NAT 无法进行工作。

第三步是用 **ip nat inside source** 命令定义需要转换的网络。这个例子中需要用下面的命令进行配置：

```
ip nat inside source list 69 pool publicpool
access-list 69 permit 172.16.1.0 0.0.0.255
```

这条命令会调用编号为 69 的访问控制列表，将进入内部接口数据包的源网络地址与此列表进行比较。如果源 IP 地址是在子网 172.16.1.x 中，网络地址会转换到名为 publicpool 的 IP 地址池去。

最后一步是用 `ip nat pool` 命令对地址池 `publicpool` 进行定义：

```
ip nat pool publicpool 128.100.100.2 128.100.100.254 netmask 255.255.255.0
```

这条命令将 128.100.100.2 到 128.100.100.254 的地址放入转换的地址池，子网掩码是 255.255.255.0。由于 128.100.100.1 是环路接口的地址，不要把这个地址包括在地址池的范围内。例 15-4 是路由器 `nat_router` 的相关配置部分的内容。

例 15-4 NAT 动态转换方式的配置示例

```
hostname nat_router
!
!
ip subnet-zero
!
interface loopback20
ip address 128.100.100.1 255.255.255.0
no ip directed-broadcast
!
interface Ethernet0
ip address 172.16.1.254 255.255.255.0
no ip directed-broadcast
ip nat inside
!
interface Serial0
ip address 128.100.11.1 255.255.255.252
no ip directed-broadcast
ip nat outside
!
<<<text omitted>>>
!
ip nat pool publicpool 128.100.100.2 128.100.100.254 netmask 255.255.255.0
ip nat inside source list 69 pool publicpool
ip classless
ip route 0.0.0.0 0.0.0.0 128.100.11.2
!
access-list 69 permit 172.16.1.0 0.0.0.255
```

15.3.2 NAT 静态转换方式的配置

静态转换方式的配置和动态方式很相似，区别是静态方式下不用配置 IP 地址池，而是配置一对一的地址映射，这是要转换成的主机地址与指定地址之间的映射。静态转换可以用在内部静态转换或者是外部静态转换之中。大多数的 NAT 应用都是仅用了内部静态转换，但是当有地址重叠时，可能需要使用外部源静态转换。

以前面图 15-3 中的例子为基础，做一些修改实现仅有一个地址 172.16.1.10 会转换成 128.100.100.10。要配置静态 NAT，也是按照上面的步骤进行，先定义内部网络和外部网络，再定义环路地址以设置全局网络，然后还要确保子网 128.100.100.0/24 和外部网络的路由畅通。与动态 NAT 的配置惟一不同的是网络进行转换的方式。静态转换方式是采用 `ip nat inside static` 命令来进行转换的，而不是命令 `ip nat inside source list x`。在这个例子中使用下面命令：

```
ip nat inside source static 172.16.1.10 128.100.100.10
```

这条命令将地址 172.16.1.10 映射到了地址 128.100.100.10。路由器上不会再进行别的转

换操作，例 15-5 是静态转换方式的配置情况。

例 15-5 NAT 静态转换的实例

```
hostname nat_rout
!
ip subnet-zero
!
interface Loopba
ip address 128.1.1.1 255.255.255.0
no ip directed-b
!
interface Etherne
ip address 172.16.1.254 255.255.255.0
no ip directed-b
ip nat inside
!
interface Serial0
ip address 128.1.1.1 255.255.255.252
no ip directed-b
ip nat outside
!
<<<text omitted>>>
!
ip nat inside sou static 172.16.1.10 128.100.100.10
ip classless
ip route 0.0.0.0 0.0.0 128.100.11.2
```

15.3.3 简单 IP 和端口地址转换 (PAT) 的配置

“简单 IP”可能是 NAT 简单 IP 地址过载方式的最佳实例，它融合了 NAT 过载/PAT 与 PPP/PPPoE 的特性。但 NAT TCP 并不局限于 PPP 的应用。

为了讲述方便，本例将 NAT TCP 过载和 PAT 看作一样的概念。PAT 提供了多对一的 IP 转换，即多个内部地址共享或者转换成一个 IP 地址。PAT 在内部全局地址上采用了一次性的转换。

简单 IP (阶段 1) 允许本地主机通过这个 IP 地址访问全局网络或互联网。很多 ISP 都利用 IPCP 来把 IP 地址动态地分配给远程串行接口。在分配之前是未知的，因此 NAT 的静态和动态转换方式都没有办法进行配置。所以 Cisco 采用简单 IP 方式来完成这种类型的配置。

下面是这种方式的配置过程：

- 第 1 步 路由器向 ISP 或中央站点路由器请求建立一个 PPP 连接时，简单 IP 利用 PPP/IPCP 从 ISP 或中央站点路由器里的动态主机配置协议 (DHCP) 服务器获取一个地址。
- 第 2 步 简单 IP 接收新的“动态”地址并将它分配给 WAN 接口。
- 第 3 步 后，简单 IP 利用端口地址转换 (PAT) 建立一个多对一的地址/端口联系，把多个内部地址和这个新的“动态”地址联系在一起。

注释 配置简单 IP 必须安装 11.3 以上 (含 11.3) 版本的 Cisco IOS。

配置简单 IP 时也按照前面的 4 步骤进行，主要区别是第 3 步。简单 IP 是典型的通过 ISDN 连接到 ISP 的家庭用户或小型办公室用户。家庭用户没有自己的 IP 地址空间，而是在拨号呼入 ISP 时获得一个 IP 地址。该用户通常使用 Internet 浏览器访问 Internet，这是简单 IP 的一个理想应用对象。

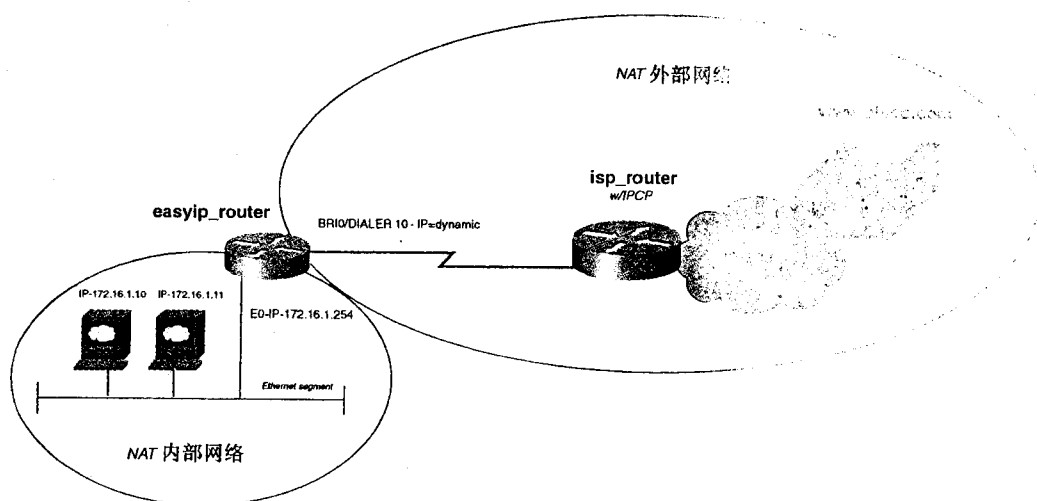


图 15-4 简单 IP 的实例

第 1 步是定义内部和外部网络。这里的内部网络是以太网，而外部网络是 Internet。用同样的 **ip nat inside** 命令和 **ip nat outside** 命令在适当的接口上配置 NAT。

下一步是建立路由器和 Internet 之间的路由。由于只有一个出口点，因此这里把默认的静态路由来指向拨号接口。使用默认路由时记得要加上 **ip classless** 命令。

第 3 步是定义网络需要转换成的地址以及转换的方式。由于 IP 地址未知，因此这里把转换指向 Dialer 10 接口。用 **overload** 命令参数通知路由器使用 PAT，这样允许多个连接通过一个 IP 地址与 Internet 实现互连。这条 **overload** 命令的形式为：

```
ip nat inside source list 10 interface Dialer10 overload
```

由于没有地址池来定义静态的转换方式，因此第 4 步用来配置 IPCP。配置 IPCP 必须采用 PPP 对第 2 网络层进行封装，而且还必须安装 11.3 或更新版本的 IOS。在串行接口或拨号接口上使用 **ip address negotiated** 命令启动 IPCP 的工作。图 15-4 是简单 IP 所需的呼叫配置以及 IPCP 的配置示例。

例 15-6 简单 IP 的 IPCP 和呼叫配置示例

```
interface BRI0
no ip address
no ip directed-broadcast
encapsulation ppp
dialer pool-member 10
isdn switch-type basic-ni
isdn spid1 71538154750101 3815475
isdn spid2 71538154760101 3815476
```

```

ppp multilink
!
interface Dialer10
ip address negotiated          ← IPCP configuration
no ip directed-broadcast
ip nat outside
encapsulation ppp
no ip mroute-cache
dialer remote-name isp_router
dialer idle-timeout 300
dialer string 4262200
dialer hold-queue 80
dialer load-threshold 10 either
dialer pool 10
dialer-group 10
compress stac
no cdp enable
ppp authentication pap
ppp pap sent-username ksolie password 7 1304474B5B5D577E
ppp multilink
!

```

注意，多数的 ISP 还使用 PAP 认证，这也成为访问 ISP 需要做的配置内容之一。可以回顾第 4 章“WAN 协议与技术：点对点协议 (PPP)”和第 7 章“WAN 协议与技术：综合业务数字网 (ISDN)”了解拨号的配置或 ISDN 的设置。

例 15-7 是简单 IP 的整个配置过程。

例 15-7 简单 IP 的配置

```

hostname easyip_router
!
ip subnet-zero
!
isdn switch-type basic-ni
!
interface Ethernet0
ip address 172.16.1.254 255.255.255.0
no ip directed-broadcast
ip nat inside
!
interface BRI0
no ip address
no ip directed-broadcast
encapsulation ppp
dialer pool-member 10
isdn switch-type basic-ni
isdn spid1 71538154750101 3815475
isdn spid2 71538154760101 3815476
ppp multilink
!
interface Dialer10
ip address negotiated
no ip directed-broadcast
ip nat outside
encapsulation ppp
no ip mroute-cache
dialer remote-name isp_router
dialer idle-timeout 300

```

(待续)

```
dialer string 4262200
dialer hold-queue 80
dialer load-threshold 10 either
dialer pool 10
dialer-group 10
no cdp enable
ppp authentication pap
ppp pap sent-username ksolie password 7 1304474B5B5D577E
ppp multilink
!
ip nat inside source list 10 interface Dialer10 overload
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer10
!
access-list 10 permit 172.16.1.0 0.0.0.0.255
access-list 110 permit ip any any
dialer-list 10 protocol ip list 110
```

15.4 NAT 的 “Big show” 和 “Big D” 命令

NAT 的 “big show” 命令包括用于显示详细 NAT 表信息的 `show ip nat translations` 命令和用于显示路由器上发生的 NAT 转换的统计信息命令 `show ip nat statistics`。

命令 `show ip nat translations` 能够显示路由器上所有 NAT 转换的信息，包括所用的协议以及内部和外部的全局与本地转换信息。例 15-8 是该命令在前面简单 IP 的配置例子中的使用情况，显示了两台工作站（172.16.1.10 和 172.16.1.11）使用同一个内部全局地址（206.191.194.42）访问 Internet 上的两个主机的情况。地址 206.191.194.42 是连接建立时 ISP 动态分配的一个转换地址。

例 15-8 show ip nat translations 命令的执行结果

```
easyip_router#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
tcp 206.191.194.42:1169 172.16.1.10:1169 198.133.219.25:80 198.133.219.25:80
tcp 206.191.194.42:1168 172.16.1.10:1168 198.133.219.25:80 198.133.219.25:80
tcp 206.191.194.42:1171 172.16.1.10:1171 198.133.219.25:80 198.133.219.25:80
tcp 206.191.194.42:1170 172.16.1.10:1170 198.133.219.25:80 198.133.219.25:80
tcp 206.191.194.42:1173 172.16.1.10:1173 198.133.219.25:80 198.133.219.25:80
tcp 206.191.194.42:1172 172.16.1.10:1172 198.133.219.25:80 198.133.219.25:80
tcp 206.191.194.42:1167 172.16.1.10:1167 198.133.219.25:80 198.133.219.25:80
udp 206.191.194.42:1050 172.16.1.11:1050 206.191.193.1:53 206.191.193.1:53
udp 206.191.194.42:1048 172.16.1.11:1048 206.191.193.1:53 206.191.193.1:53
udp 206.191.194.42:1049 172.16.1.11:1049 206.191.193.1:53 206.191.193.1:53
udp 206.191.194.42:1046 172.16.1.11:1046 206.191.193.1:53 206.191.193.1:53
udp 206.191.194.42:1044 172.16.1.11:1044 206.191.193.1:53 206.191.193.1:53
tcp 206.191.194.42:1045 172.16.1.11:1045 63.251.8.23:80 63.251.8.23:80
udp 206.191.194.42:1057 172.16.1.11:1057 206.191.193.1:53 206.191.193.1:53
easyip_router#
```

注释 测试三种中任何一种 NAT 配置方式的最为简单的方法是在内部网络中测试与外部

命令 **show ip nat statistics** 则是对路由器上的 NAT 操作进行总结，列出了当前活动的转换以及方式：静态、动态或扩展的，此外还能显示 NAT 的内部和外部接口的情况。例 15-9 是该命令在路由器 **easy_ip** 上的输出示例。

例 15-9 show ip nat statistics 的执行示例

```
easyip_router#show ip nat statistics
Total active translations: 12 (0 static, 12 dynamic; 12 extended)
Outside interfaces:
  BRI0:1, BRI0:2, Dialer10, Virtual-Access1
Inside interfaces:
  Ethernet0
Hits: 2304 Misses: 190
Expired translations: 134
Dynamic mappings:
-- Inside Source
access-list 10 interface Dialer10 refcount 12
```

本例的转换全是动态扩展。代码中的 **Hits** 是指 Cisco IOS 在转换表中查找时找到匹配项的次数，而 **Misses** 则是 IOS 无法找到匹配项而创建新地址的次数。**Expired translations** 列出路由器启动之后过期了的转换数目。

命令 **show ip nat translations verbose** 显示的转换信息要比 **show ip nat translations** 的更为详细，包括转换创建的时间，转换所用的时间以及过期的时间，同时还标注了所有用到的标志，如扩展端口转换标志等。例 15-10 是该命令在前面那个简单 IP 实例中的执行结果。

例 15-10 show ip nat translation verbose 命令的执行示例

```
easyip_router#show ip nat translations verbose
Pro Inside global      Inside local      Outside local      Outside global
tcp 206.191.194.42:1066 172.16.1.11:1066 128.11.25.241:80   128.11.25.241:80
  create 00:00:23, use 00:00:22, left 23:59:37, flags:extended
tcp 206.191.194.42:1063 172.16.1.11:1063 128.11.25.252:80   128.11.25.252:80
  create 00:00:23, use 00:00:23, left 23:59:36, flags:extended
tcp 206.191.194.42:1065 172.16.1.11:1065 128.11.25.241:80   128.11.25.241:80
  create 00:00:23, use 00:00:23, left 23:59:36, flags:extended
easyip_router#
```

NAT 提供的 **debug** 命令比较有限，都是从 **debug ip nat** 命令派生出来的，其格式为：

```
debug ip nat [detailed]
```

命令 **debug ip nat** 能够显示所有当前转换的每个端口与地址联系情况，其派生形式可以提供与接口相关的附加信息，同时还包括端口协商信息等。实际使用时要千万小心。一个设备通过这条命令的输出结果数量非常大。例 15-11 中可以看到每毫秒 (ms) 产生的信息数量。最好只在查找某个特定的 NAT 问题时才使用这条命令。

例 15-11 路由器 easy_ip 上 debug ip nat detailed 命令的输出结果

```
easyip_router#debug ip nat detailed
IP NAT detailed debugging is on
00:24:07: NAT: i: udp (172.16.1.10, 137) -> (206.191.193.1, 53) [25601]
00:24:07: NAT: ipnat_allocate_port: wanted 137 got 137
```

(待续)

```
00:24:07: NAT: s=172.16.1.10->206.191.194.42, d=206.191.193.1 [25601]
00:24:07: NAT: o: udp (206.191.193.1, 53) -> (206.191.194.42, 137) [44225]
00:24:07: NAT: s=206.191.193.1, d=206.191.194.42->172.16.1.10 [44225]
00:24:51: NAT: i: udp (172.16.1.10, 1046) -> (206.191.193.1, 53) [25857]
00:24:51: NAT: ipnat_allocate_port: wanted 1046 got 1046
00:24:51: NAT: s=172.16.1.10->206.191.194.42, d=206.191.193.1 [25857]
00:24:51: NAT: o: udp (206.191.193.1, 53) -> (206.191.194.42, 1046) [22909]
00:24:51: NAT: s=206.191.193.1, d=206.191.194.42->172.16.1.10 [22909]
00:24:51: NAT: i: udp (172.16.1.10, 1047) -> (206.191.193.1, 53) [26113]
00:24:51: NAT: ipnat_allocate_port: wanted 1047 got 1047
```

警告 使用 `debug ip nat` 命令时一定小心。在 1 毫秒 (ms) 的时间里能产生许多项输出结果。最好是和全局配置命令 `logging buffered` 一起使用这条 `debug` 命令。

15.5 NAT 转换的清除和改变

默认情况下，NAT TCP 转换在 24 小时之后会过期，根据不同的协议类型，可以采用下面的命令来改变这个超时的设置值：

- `ip nat translation timeout seconds`——指定动态转换方式（不包括过载转换方式）使用的超时，默认是 86 400 秒，即 24 小时。
- `ip nat translation udp-timeout seconds`——指定 UDP 协议转换的超时，默认是 300 秒，即 5 分钟。
- `ip nat translation dns-timeout seconds`——指定 DNS 的转换超时，默认 60 秒。
- `ip nat translation tcp-timeout seconds`——指定 TCP 的转换超时，默认 86 400 秒，即 24 小时。
- `ip nat translation finrst-timeout seconds`——指定那些设置了报头中的 FIN 或 RST 位的 NAT TCP 数据流的转换超时，默认是 60 秒。
- `ip nat translation icmp-timeout seconds`——指定 NAT ICMP 数据包的超时，默认 60 秒。
- `ip nat translation port-timeout [tcp | udp] port_number seconds`——指定那些特定 TCP 或 UDP 端口上的转换超时。
- `ip nat translation syn-timeout`——指定那些报头中设置了 SYN 位的 NAT TCP 数据流的转换超时。

要清除 NAT 的转换统计信息，可以采用下面的命令：

- `clear ip nat translations [*|inside inside_address | outside outside_address | tcp port_number | udp port_number]`
- `clear ip nat statistics`

15.6 NAT 的局限性以及使用

部网络向外呼叫避免了外部网络中的主机向内建立会话，从而赋予了网络内在的安全保证。但是也有其局限性。很多协议，如 SNMP 和 BOOTP，都在其数据部分中嵌入了 IP 地址。不少应用程序不用 IP 报头中的源地址，相反地却去采用数据部分中的嵌入地址以便路由回到接收信息的主机。这类情况下，NAT 将会失效。NAT 可以识别这些类型的数据，针对这种数据类型（如 FTP）需要用特殊用法来处理 NAT。表 15-3 列出了 NAT 支持与不支持的数据类型。

表 15-3 NAT 对数据类型支持情况列表

NAT 支持的数据类型/应用	NAT 不支持的数据类型/应用
应用数据中不带源或目的 IP 地址的所有 TCP/UDP 数据	IP 多播 CISCO IOS 12.0 (1) T 支持下列类型： 数据包源地址转换 PIM, Auto-RP, PIM V2 和 BSR mstat, mrimfio 和 mtrace SDR 宣告或应用载荷
HTTP	路由更新
TFTP	DNS 域传送
TELNET	BOOTP
Archie	Talk,ntalkj
Finger	SNMP
NTP	netshow
NFS	
Rlogin,RSH,RCP	

NAT 支持以下在数据部分携带 IP 地址的应用：

ICMP.

FTP. (参照以下章节：非标准端口的操作。)

基于 TCP/IP 上的 NETBIOS (仅仅支持数据报和名字服务，会话服务将在以后版本的 IOS 中支持。)

不支持网络实时音频、RTSP

White Pines 的 CuSeeMe.

Xing 公司的 SteamWorks.

DNS 的“A”和“PTR”查询。

IOS 12.0 (1) /12.0 (1) T 或者更新的版本支持 H.323.

IOS 12.0 (1) /12.0 (1) T 或者更新的版本支持 NetMeeting 2.1, 2.11, 3.01

12.1 (5) T 版本的 IOS 支持 NetMeeting Directory (ILS Servers)

IOS 11.3 (4) /11.3 (4) T 或者更新的版本支持 VDOLive

IOS 11.3 (4) /11.3 (4) T 或者更新版本支持 Vxtreme

注释 NAT 地址池和转换都遵守 0 子网原则。如果 NAT 地址池在 IP 的 0 子网中，NAT 不能进行。12.0 及更新版本的 Cisco IOS 默认设置 **ip subnet zero** 命令。如果想要在 12.0 以前版本的 Cisco IOS 上将 0 子网作为 NAT 地址池，可以使用 **ip subnet zero** 命令。

15.7 NAT 与非标准 FTP 端口号

路由器利用了端口 21 的作用，相应地把数据中的地址用一个新转换得来的地址代替，然后重新计算所需的校验值。FTP 在使用非标准端口号时出现了一个问题。NAT 无法将数据流识别为 FTP 请求，因而传输时未做任何更改。自然，数据到达目的地址之后，由于有效载荷中的地址与 IP 报头中的地址不符，FTP 请求肯定会以失败告终。

在 Cisco IOS 11.3 (3) 和 Cisco IOS 11.2. (13) 中，Cisco 增加了使用非标准 TCP 端口号进行 FTP 的功能。命令 `ip nat service list [1-100] ftp tcp port xxxx` 调用需要进行转换的网络访问控制列表，然后寻找工作在端口 xxxx 上的 FTP 数据包。如果找到匹配项，路由器会对相应的 FTP 数据包做相应的修改。

15.8 实验 31：配置动态 NAT 与非标准 FTP 端口号的应用——第 1 部分

15.8.1 实验说明

随着各种网络的相互融合和 Internet 访问的激增，使用 NAT 的需求也与日俱增。NAT 也可用来提高网络的安全性。可以通过使用某个路由选择协议不传播特定子网来避免外部网络向内部网络发起建立会话请求。

15.8.2 实验内容

假设杜兰德（Durand）学区决定将两所比较小的高中合并以形成一个较大的学区。JP Memorial School 可以获得一条 T1 的 HDLC 链路与 Durand 高中相连。dhs_router 注册了无类域间路由（CIDR）地址，包括 200.100.1.16/29 和 200.100.1.32/29。jpmc 则是采用没有注册的 IP 地址范围 9.3.3.0/24。路由器 jpmc_router 要求访问 Internet，但同时应限制对服务器 200.100.1.18 的 FTP 和 ping 访问。在设计该网络时需要遵循下面要求：

- 对网络进行配置，使得子网 9.3.3.0/24 中的工作站可以访问 Internet，并应限制对子网 200.100.1.16/29 的访问。
- 对访问进行控制，使 NAT 支持对 Internet 访问，但是当数据是发送前往主机 200.100.1.18 时，只允许 NAT 转换 ICMP 和 FTP。
- 将 200.100.1.32/29 的 CIDR 地址作为我们的 NAT 地址池。
- 可选：配置 FTP 工作在端口 2021 而不是 21 上。

15.8.3 实验目的

- 按照图 15-5 配置网络，利用环路接口地址模拟 Internet，使 198.133.219.25/24 地址在实验中模拟 Internet。

只转换 FTP 应用，允许 ping 主机 200.100.1.18，同时禁止从子网 9.3.3.0/24 到 200.100.1.17 的 ping 访问。

- 不要建立从路由器 dhs_router 到 jpms_router 的以太网端口的路由，不要使用路由选择协议传播子网 9.3.3.0/24 的路由信息。

15.8.4 所需设备

- 2 台 Cisco 路由器，相互之间通过 V.35 背对背线缆或类似方式相连。本实验的可选部分还需要 11.2 及更新版本的 Cisco IOS 和 11.2 (13) 或 11.3 (3) 及更新版本的 Cisco IOS。
- 通过集线器或交换机创建的 2 个 LAN 网段。
- 2 台 IP 工作站，一台作为 FTP 服务器，另一台作为客户工作站。FTP 服务器和客户端软件可以从 download.cnet.com 网站下载。

15.8.5 物理设计与实验准备

- 按照图 15-5 将集线器以及串行线缆与路由器相互连接。
- 按照图 15-5 将两台以太集线器与路由器相连形成两个 LAN 网段。
- 按照图 15-5 对两个 IP 工作站进行连接和配置。将工作站 200.100.1.18 作为 FTP 服务器，而 9.3.3.10 则作为 FTP 客户端。通过将一份测试文件拷贝到 FTP 服务器的公共文件夹中去，来测试 FTP 功能。
- 可选：配置 FTP 工作端口为 2021 而不是端口 21。这是通过把服务器软件改为运行在端口 2021 上来实现的。同时还要确保客户端也相应设置成与服务器端口 2021 相连。

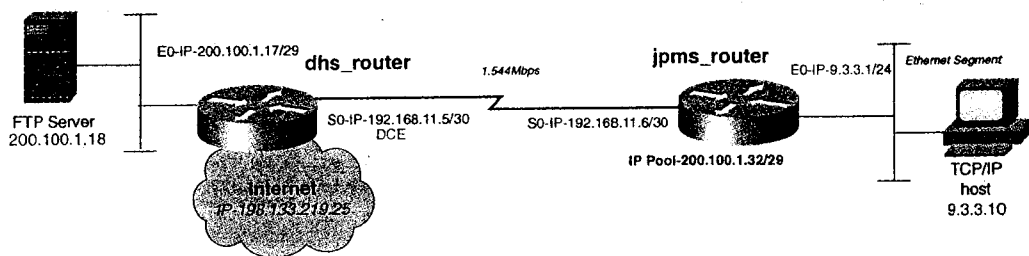


图 15-5 DHS 学区网络——动态 NAT

15.9 实验 31：配置动态 NAT 与非标准 FTP 端口号的应用——第 2 部分

15.9.1 实验步骤

首先来看路由器 `dhs_router`，为它的以太网接口以及串行接口配置 IP 地址。Serial 0 端口是链路的 DCE 端，因此要用 `clockrate` 命令设置通信速率。这台路由器上不应“看到”子网 9.3.3.0/24。按要求，还要为路由器分配一个用于 NAT 的 CIDR 地址范围 200.100.1.32/29，而这需要有一条路由来指明网络的位置。为此，在路由器 `dhs_router` 上创建一条相应到达该路由器的静态路由。

就实验而言，与真正 Internet 的连接其实可有可无。如果与 Internet 没有连接，可以通过加上一个地址为 198.133.219.25 的环路接口来模拟一台 IP 主机。NAT 正常工作时，从子网 9.3.3.0/24 可以 ping 通这个地址。例 15-12 是到目前为止路由器 `dhs_router` 的相关配置示例。

例 15-12 `dhs_router` 路由器的配置

```
hostname dhs_router
!
<<<text omitted>>>
!
interface Loopback20
 ip address 198.133.219.25 255.255.255.0
!
interface Ethernet0
 ip address 200.100.1.17 255.255.255.248
!
interface Serial0
 ip address 192.168.11.5 255.255.255.252
 no fair-queue
 clockrate 2000000
!
<<<text omitted>>>
!
no ip classless
ip route 200.100.1.32 255.255.255.248 192.168.11.6
```

路由器 `jpmis_router` 的配置稍微复杂一些。首先为其以太网接口和串行接口分配适当的 IP 地址。然后是配置指向 192.168.11.5 的默认路由，使用默认路由时别忘了加上 `ip classless` 命令：

```
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.11.5
```

到现在，从这台路由器可以 ping 通“Internet”地址 198.133.219.25。但如果从该路由器的 Ethernet 0 接口或者是从工作站 9.3.3.10 上向“Internet”或者是子网 200.100.1.16/29 执行源地址 ping，应该 ping 不通。

现在应该在路由器 `jpmis_router` 上配置 NAT。首先定义内部网络和外部网络。图 15-6 分别给出了路由器 `jpmis_router` 的内部网络和外部网络拓扑。然后，通过在串行接口上应用 `ip nat outside` 命令，在以太网接口上设置 `ip nat inside` 命令配置内部和外部网络。

NAT 配置的下一步是确保地址池与路由器 `dhs_router` 之间存在 IP 路由关系。DHS 分配了一个 IP 子网 200.100.1.32/29，必须在路由器 `jpmis_router` 上配置该子网。因此，在路由器上创建环路接口并分配 IP 地址 200.100.1.33。请注意，到达该子网的路由也是路由器 `dhs_router` 上惟一的子网静态路由。

第 3 步是定义要转换的地址范围与协议。本例中，希望利用 NAT 进行转换的包括去往子网 200.100.1.16/29 中的惟一主机 200.100.1.18 的 FTP 和 ICMP 地址。同时还有通往 Internet

的所有地址。这一切是通过在 `ip nat inside source` 命令中使用路由图来实现的。该路由图可以调用扩展访问列表，访问列表中对数据类型做了明确的规定。此外，对该命令还需要使用 `pool` 关键字。例 15-13 是路由器 `jpms_router` 所需的路由图和访问列表示例。

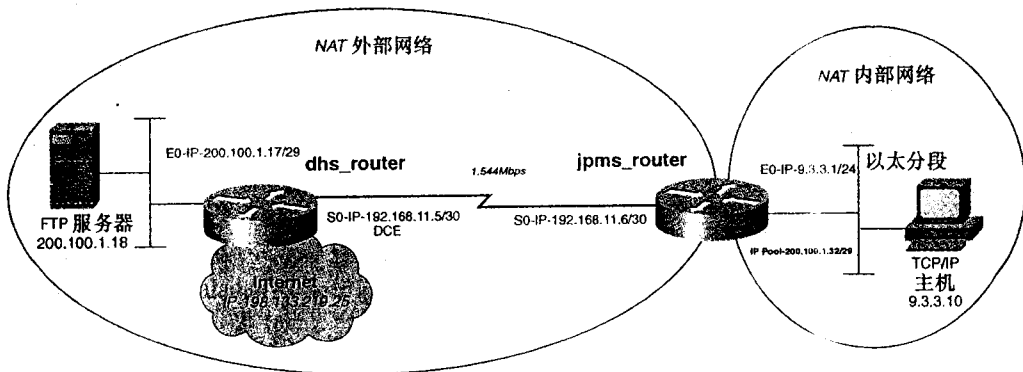


图 15-6 DHS 学区网络的内部网络与外部网络

例 15-13 配置所要转换的地址和协议

```
jpms_router(config)#ip nat inside source route-map trans_nat pool legalpool
jpms_router(config)#route-map trans_nat permit 10
jpms_router(config-route-map)# match ip address 101
jpms_router(config-route-map)#exit
jpms_router(config)# access-list 101 permit icmp 9.3.3.0 0.0.0.255
host 200.100.1.18 echo
jpms_router(config)# access-list 101 permit icmp 9.3.3.0 0.0.0.255
host 200.100.1.18 echo-reply
jpms_router(config)# access-list 101 permit tcp 9.3.3.0 0.0.0.255
host 200.100.1.18 eq ftp
jpms_router(config)# access-list 101 deny ip 9.3.3.0 0.0.0.255
200.100.1.16 0.0.0.7
jpms_router(config)#access-list 101 permit ip 9.3.3.0 0.0.0.255 any
```

NAT 配置的最后一步是配置 NAT 地址池。由于不希望转换环路接口的主机地址，因而地址池的起始地址是 200.100.1.34，而终点地址是 200.100.1.38，其中忽略了广播地址 200.100.1.39。

利用下面这条命令可以对名为 `legalpool` 的地址池进行配置：

```
jpms_router (config) # ip nat pool legalpool 200.100.1.34 200.100.1.38
netmask 255.255.255.248
```

例 15-14 是路由器 `jpms_router` 的完整配置情况。

例 15-14 路由器 `jpms_router` 的完整配置

```
hostname jpms_router
!
<<<text omitted>>>
!
interface Loopback20
ip address 200.100.1.33 255.255.255.248
```

(待续)

```

no ip directed-broadcast
!
interface Ethernet0
 ip address 9.3.3.1 255.255.255.0
 no ip directed-broadcast
 ip nat inside
!
interface Serial0
 ip address 192.168.11.6 255.255.255.252
 no ip directed-broadcast
 ip nat outside
 no ip mroute-cache
!
<<<text omitted>>>
!
ip nat pool legalpool 200.100.1.34 200.100.1.38 netmask 255.255.255.248
ip nat inside source route-map trans_nat pool legalpool
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.11.5
!
access-list 101 permit icmp 9.3.3.0 0.0.0.255 host 200.100.1.18 echo
access-list 101 permit icmp 9.3.3.0 0.0.0.255 host 200.100.1.18 echo-reply
access-list 101 permit tcp 9.3.3.0 0.0.0.255 host 200.100.1.18 eq ftp
access-list 101 deny ip 9.3.3.0 0.0.0.255 200.100.1.16 0.0.0.7
access-list 101 permit ip 9.3.3.0 0.0.0.255 any
route-map trans_nat permit 10
 match ip address 101

jpms_router#
    
```

现在可以在路由器 jpms_router 上的工作站 9.3.3.10 上对所做的配置进行测试。需要两台工作站正常测试 NAT 配置。路由器 dhs_router 上的工作站运行 FTP 服务器软件，而 jpms_router 上的工作站配置为 FTP 客户端。从客户端上 ping FTP 服务器时，应该是可以 ping 通服务器，但是 ping 不通路由器 dhs_router 的以太网端口。此外，还可以 ping 通地址 198.133.219.25。FTP 的测试通过从客户端向服务器发起 FTP 会话请求来完成。

接着，通过测试保证可以通过网络传输文件。如果这方面有问题，首先检查路由器 dhs_router 是否在路由器 jpms_router 上的地址池中。又由于没有采用任何路由选择协议，所以这里用于测试的路由器、服务器、工作站都需要含有默认路由或者静态路由，以确保相互之间的连通。再者，确认所用的路由图调用了正确的访问列表。用 show access-list 命令检查访问表的配置是否正确。另外，确保采用的 NAT 地址池主机地址是和环路接口在同一个子网。只有这样，子网才算是该路由器上的地址池中。例 15-15 是进行测试时，NAT 转换的 show 命令执行示例。

例 15-15 配置 NAT 的进出

```

jpms_router#show ip nat trans
Pro Inside global      Inside local      Outside local      Outside global
icmp 200.100.1.33:512  9.3.3.10:512      200.100.1.18:512  200.100.1.18:512
tcp 200.100.1.33:1076  9.3.3.10:1076     200.100.1.18:21   200.100.1.18:21
tcp 200.100.1.33:1077  9.3.3.10:1077     200.100.1.18:20   200.100.1.18:20
tcp 200.100.1.33:1072  9.3.3.10:1072     200.100.1.18:21   200.100.1.18:21
jpms_router#
jpms_router#show ip nat stat
    
```

(待续)

```

Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Outside interfaces:
  Serial0
Inside interfaces:
  Ethernet0
Hits: 3727 Misses: 87
Expired translations: 89
Dynamic mappings:
-- Inside Source
route-map trans_nat pool legalpool refcount 1
pool legalpool: netmask 255.255.255.248
  start 200.100.1.33 end 200.100.1.38
  type generic, total addresses 6, allocated 1 (16%), misses 0
jpms_router#
jpms_router# show access-lists
Extended IP access list 101
  permit icmp 9.3.3.0 0.0.0.255 host 200.100.1.18 echo (1 match)
  permit icmp 9.3.3.0 0.0.0.255 host 200.100.1.18 echo-reply
  permit tcp 9.3.3.0 0.0.0.255 host 200.100.1.18 eq ftp (2 matches)
  deny ip 9.3.3.0 0.0.0.255 200.100.1.16 0.0.0.7 (4 matches)
  permit ip 9.3.3.0 0.0.0.255 any (1 match)
jpms_router#

```

本实验的可选部分是配置新的 NAT 特性，Cisco 引入这些特性是为了处理在数据流中传输 IP 地址的常用应用，FTP 就是其中之一。FTP 的应用率非常高，进行 NAT 转换时，Cisco 路由器如果识别到端口号 21，就会将数据包的数据部分以及 IP 报头和校验和加以改动。只要 FTP 工作在端口 21 上，这一方法就极为有效。但如果 FTP 连接不发生在端口 21 上，NAT 就无法正确处理这些数据包。例 15-16 是 `debug ip nat detailed` 命令的结果显示部分数据包没有正确转换的情况。

例 15-16 debug ip nat detailed 结果显示 FTP 端口失败

```

jpms_router#debug ip nat detailed
IP NAT detailed debugging is on
jpms_router#
11:36:27: NAT: i: udp (9.3.3.10, 1154) -> (206.191.193.1, 53) [36138]
11:36:27: NAT: i: udp (9.3.3.10, 1154) -> (204.221.151.213, 53) [36394]
11:36:27: NAT: o: icmp (192.168.11.5, 53) -> (200.100.1.33, 1154) [524]
11:36:31: NAT: i: udp (9.3.3.10, 1154) -> (206.191.193.1, 53) [36650]
11:36:31: NAT: i: udp (9.3.3.10, 1154) -> (204.221.151.213, 53) [36906]
11:36:31: NAT: o: icmp (192.168.11.5, 53) -> (200.100.1.33, 1154) [525]
11:36:38: NAT: i: tcp (9.3.3.10, 1155) -> (200.100.1.18, 2021) [37162]
11:36:41: NAT: i: tcp (9.3.3.10, 1155) -> (200.100.1.18, 2021) [37418]
11:36:47: NAT: i: tcp (9.3.3.10, 1155) -> (200.100.1.18, 2021) [37674]
11:36:59: NAT: i: tcp (9.3.3.10, 1155) -> (200.100.1.18, 2021) [37930]
11:37:24: NAT: i: tcp (9.3.3.10, 1156) -> (200.100.1.18, 2021) [38442]
11:37:27: NAT: i: tcp (9.3.3.10, 1156) -> (200.100.1.18, 2021) [38698]
11:37:31: NAT: deleting alias for 200.100.1.33
11:37:33: NAT: i: tcp (9.3.3.10, 1156) -> (200.100.1.18, 2021) [38954]
11:37:45: NAT: i: tcp (9.3.3.10, 1156) -> (200.100.1.18, 2021) [39210]
11:38:11: NAT: i: tcp (9.3.3.10, 1157) -> (200.100.1.18, 2021) [39466]
11:38:14: NAT: i: tcp (9.3.3.10, 1157) -> (200.100.1.18, 2021) [39722]
11:38:20: NAT: i: tcp (9.3.3.10, 1157) -> (200.100.1.18, 2021) [39978]
11:38:32: NAT: i: tcp (9.3.3.10, 1157) -> (200.100.1.18, 2021) [40234]
11:40:09: NAT: i: udp (9.3.3.10, 1158) -> (206.191.193.1, 53) [40490]
11:40:09: NAT: map match trans_nat
11:40:09: NAT: installing alias for address 200.100.1.33
11:40:09: NAT: alias insert failed for 200.100.1.33

```

入站接口从未收到来自 200.100.1.18 到数据包。例 15-17 成功地在端口 21 上建立了 FTP 连接，可以把两个例子的结果比较一下。

例 15-17 debug ip nat detailed 结果显示成功的 FTP NAT 转换

```
jpms_router#debug ip nat detailed
IP NAT detailed debugging is on
jpms_router#
11:33:03: NAT: created edit_context (9.3.3.10, 1145) -> (200.100.1.18, 21)
11:33:03: NAT: o: tcp (200.100.1.18, 21) -> (200.100.1.33, 1145) [40457]
11:33:03: NAT: i: tcp (9.3.3.10, 1145) -> (200.100.1.18, 21) [11791]
11:33:03: NAT: o: tcp (200.100.1.18, 21) -> (200.100.1.33, 1145) [40713]
11:33:03: NAT: i: tcp (9.3.3.10, 1145) -> (200.100.1.18, 21) [12047]
11:33:03: NAT: o: tcp (200.100.1.18, 21) -> (200.100.1.33, 1145) [41225]
11:33:03: NAT: i: tcp (9.3.3.10, 1145) -> (200.100.1.18, 21) [12303]
11:33:03: NAT: o: tcp (200.100.1.18, 21) -> (200.100.1.33, 1145) [41481]
11:33:03: NAT: i: tcp (9.3.3.10, 1145) -> (200.100.1.18, 21) [12559]
```

这个例子中，在端口 21 上接收到了来自 200.100.1.18 的入站和出站请求，表明转换成功。

要允许 NAT 在非 21 的端口上进行，可以采用 **ip nat service** 命令，再加上用于识别 FTP 主机的访问列表。在这个实验中，还需改动访问列表，将 TCP 端口 2021 包括进去。例 15-18 是这一配置的改动情况。

例 15-18 非标准 FTP 端口号与 FTP 的使用

```
jpms_router(config)#ip nat service list 1 ftp tcp port 2021
jpms_router(config)# access-list 1 permit 200.100.1.18
jpms_router(config)# no access-list 101
jpms_router(config)# access-list 101 permit icmp 9.3.3.0 0.0.0.255
host 200.100.1.18 echo
jpms_router(config)# access-list 101 permit icmp 9.3.3.0 0.0.0.255
host 200.100.1.18 echo-reply
jpms_router(config)# access-list 101 permit tcp 9.3.3.0 0.0.0.255
host 200.100.1.18 eq 2021
jpms_router(config)# access-list 101 deny ip 9.3.3.0 0.0.0.255
200.100.1.16 0.0.0.7
jpms_router(config)#access-list 101 permit ip 9.3.3.0 0.0.0.255 any
```

现在 NAT 认为到主机 200.100.1.18 的端口 2021 也是 FTP 数据流，对此做相应改动。例 15-19 是现在 **debug ip nat detailed** 命令的输出情况，表明端口 2021 得到了正确的使用。

例 15-19 debug ip nat detailed 显示端口 2021 上 FTP 的工作情况

```
11:48:17: NAT: i: tcp (9.3.3.10, 1164) -> (200.100.1.18, 2021) [52266]
11:48:17: NAT: o: tcp (200.100.1.18, 2021) -> (200.100.1.33, 1164) [4645]
11:48:17: NAT: i: tcp (9.3.3.10, 1164) -> (200.100.1.18, 2021) [52522]
11:48:17: NAT: o: tcp (200.100.1.18, 2021) -> (200.100.1.33, 1164) [5157]
11:48:17: NAT: i: tcp (9.3.3.10, 1164) -> (200.100.1.18, 2021) [52778]
11:48:17: NAT: o: tcp (200.100.1.18, 2021) -> (200.100.1.33, 1164) [5413]
```

15.10 实验 32：配置静态 NAT 和 DLSw——第 1 部分

15.10.1 实验说明

随着网络自动化的增强，很多用户在使用包含特定 IP 地址的应用程序。当 IP 地址发生改变时，需要在很多工作站上花费时间和精力修改代码和主机文件。这种情况下，如果使用 NAT 地址的静态映射方式会带来理想的效果，可以避免每次需要服务时都要使用不同的转换地址。静态映射方式同时还允许外部网络向内部网络请求建立会话。记住，这只在用户应用程序不需要在数据流中传输 IP 地址的情况下才有效。

15.10.2 实验内容

假设 Harms 公司是北威斯康星的一家主要的咨询公司，该公司打算对其网络进行 IP 地址的改动。为了避免一次对所有的地址和主机表进行更改，Harms Co.打算采用 NAT 用于变动时期。子网 190.10.1.0/24 以前是在路由器 green_bay 的以太网段上。很多主机（没有在这个实验的图 15-7 中表示出来）都包含有到 190.10.1.0/24 这个子网中主机的静态条目。新的子网 210.168.1.0/24 安排在这台路由器 green_bay 的以太网段上。这种情况下，需要利用 NAT 保存路由器 green_bay 以太网段上工作站的当前主机表。需要按照下面的要求设计网络：

- 按照图 15-7 对网络进行配置，路由选择协议采用 EIGRP，自治系统 ID 为 7。
- 对路由器 green_bay 进行配置，使其不会以 EIGRP 来传播新的子网 210.168.1.0/24。
- 将静态转换配置如下：
 - 210.168.1.254 转换为 190.10.1.1
 - 210.168.1.250 转换为 190.10.1.2
- 可选：在路由器 harms_co 和 green_bay 之间配置 DLSw 对等体，harms_co 的本地对等体是 198.100.1.10，而 green_bay 的本地对等体则是 210.168.1.254。

15.10.3 实验目的

- 按照图 15-7 配置网络，路由选择协议采用 EIGRP，而子网 210.168.1.0/24 的传播则不采用 EIGRP。
- 将静态转换配置如下：
 - 210.168.1.254 转换为 190.10.1.1
 - 210.168.1.250 转换为 190.10.1.2

15.10.4 所需设备

以太网段与另一台路由器相连。所需的 Cisco IOS 为 11.2 或更新版本。

- 通过集线器或交换机创建的 2 个 LAN 网段。
- 两台 IP 工作站，以用于 NAT 配置的测试。

15.10.5 物理设计与实验准备

- 按照图 15-7 将集线器以及串行线缆与路由器相连。
- 按照图 15-7 将两台以太集线器与路由器相连，以形成两个 LAN 网段。
- 按照图 15-7 对两台 IP 工作站进行连接和配置。这是可选内容，但是可以对所做的 NAT 配置进行测试。

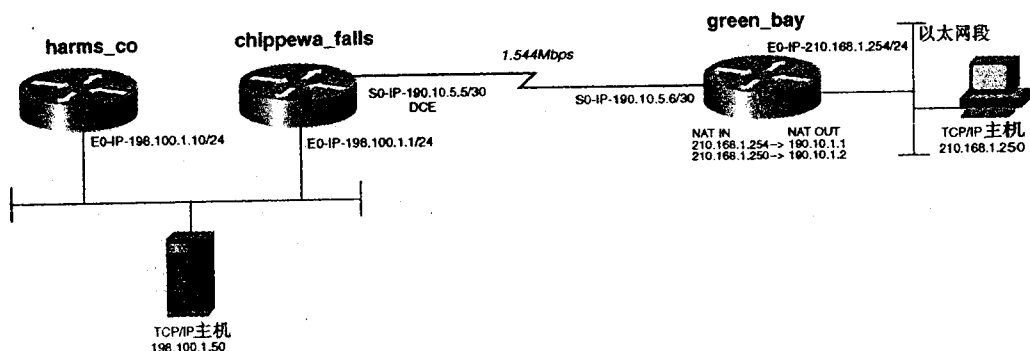


图 15-7 Harms Co. 网络——含有 DLSw 的静态 NAT

15.11 实验 32：配置静态 NAT 和 DLSw——第 2 部分

15.11.1 实验步骤

完成了串行链路与两个以太网段的物理连接后，需要在适当的子网间建立 IP 连接。所有路由器上都采用 EIGRP 作为路由选择协议。记住子网 210.168.1.0/24 不能用 EIGRP 来传播。

首先对路由器 green_bay 进行配置，对内部网络和外部网络进行定义。在这个实验中，外部网络是 198.100.1.0/24 和 190.10.5.4/30，而内部网络则是 210.168.1.0/24 和 190.10.1.0/24。为此，在 E0 端口配置 ip nat inside 命令，而在 S0 端口则配置 ip nat outside 命令。

接下来，在 IP 子网 190.10.1.0/24 和网络剩余部分之间建立完整的 IP 连接，通过创建一个环路接口以及使用子网 190.10.1.0/24 中的一个 IP 主机地址来实现。路由建立之后，可以从路由器 harms_co ping 通这些地址。例 15-20 是 NAT 配置的前两个步骤所需的命令。

例 15-20 路由器 green_bay 上的 NAT 配置

```
green_bay(config)#int e0
green_bay(config-if)#ip nat inside
```

（待续）


```
green_bay(config-if)#int s0
green_bay(config-if)#ip nat outside
green_bay(config-if)#exit
green_bay(config)#router eigrp 7
green_bay(config-router)#network 190.10.0.0
green_bay(config-router)#^Z
green_bay#
```

现在可以定义需要用 NAT 进行转换的地址。路由器 green_bay 上需要把 210.168.1.250 转换为 190.10.1.2 和把 210.168.1.254 转换为 190.10.1.1。这通过下面的全局 NAT 命令来实现：

```
ip nat inside source static 210.168.1.250 190.10.1.2
ip nat inside source static 210.168.1.254 190.10.1.1
```

到现在为止，工作站 210.168.1.250 可以访问网络的剩余部分。外部网络的主机可以分别通过 190.10.1.2 和 190.10.1.1 访问 210.168.1.250 和 210.168.1.254。例 15-21 是路由器 green_bay 的 NAT 表的情况。

例 15-21 路由器 green_bay 上 show ip nat translations 命令的执行情况

```
green_bay#show ip nat trans
Pro Inside global      Inside local      Outside local      Outside global
tcp 190.10.1.2:1084    210.168.1.250:1084 198.100.1.50:21    198.100.1.50:21
--- 190.10.1.2         210.168.1.250     ---                ---
--- 190.10.1.1         210.168.1.254     ---                ---
green_bay#
```

例 15-22 分别给出了路由器 chippewa_falls 和 green_bay 的配置示例。

例 15-22 chippewa_falls 和 green_bay 路由器的配置示例

```
hostname chippewa_falls
!
<<<text omitted>>>
!
interface Ethernet0
ip address 198.100.1.1 255.255.255.0
!
interface Serial0
ip address 190.10.5.5 255.255.255.252
no fair-queue
clockrate 2000000
!
<<<text omitted>>>
!
router eigrp 7
network 198.100.1.0
network 190.10.0.0

hostname green_bay
!
<<<text omitted>>>
!
interface Loopback20
ip address 190.10.1.3 255.255.255.0
```

(待续)

```

no ip directed-broadcast
!
interface Ethernet0
ip address 210.168.1.254 255.255.255.0
no ip directed-broadcast
ip nat inside
!
interface Serial0
ip address 190.10.5.6 255.255.255.252
no ip directed-broadcast
ip nat outside
no ip mroute-cache
no fair-queue
!
<<<text omitted>>>
!
router eigrp 7
network 190.10.0.0
!
ip nat inside source static 210.168.1.250 190.10.1.2
ip nat inside source static 210.168.1.254 190.10.1.1
<<<text omitted>>>
    
```

该实验的可选部分解决了与 NAT 相关的 DSLw 的问题。首先，按照要求对 DSLw 进行配置，在 210.168.1.254 上进行 DSLw 的配置，以路由器 green_bay 为本地对等体，而远程对等体则是指向 198.100.1.10。例 15-23 就是相关的 DSLw 配置的情况。

例 15-23 配置路由器 green_bay 上的 DSLW，设置 210.168.1.254 为本地对等体，198.100.1.10 为远程对等体

```

dslw local-peer peer-id 210.168.1.254
dslw remote-peer 0 tcp 198.100.1.10
dslw bridge-group 1
!
interface Ethernet0
ip address 210.168.1.254 255.255.255.0
no ip directed-broadcast
ip nat inside
bridge-group 1
    
```

例 15-24 是路由器 harms_co 上的 DSLw 配置示例。

例 15-24 路由器 harms_co 上的 DSLw 配置

```

dslw local-peer peer-id 198.100.1.10 promiscuous
dslw bridge-group 1
!
interface Ethernet1
ip address 198.100.1.10 255.255.255.0
media-type 10BaseT
bridge-group 1
    
```

这里的配置过程没有错误，但是 DSLw 会由于 NAT 的原因而发生错误。观察对等体时会发现尽管对等体之间存在 IP 连通性，但是二者始终没有建立连接。RFC 1795 定义了如何通过控制向量处理 TCP 连接。性能交换过程中通过协商确定控制向量。DSLw 利用两个 TCP

会话来进行数据的交换。而 Cisco 路由器只使用一个 TCP 会话，断开了另一个 TCP 连接。在确定断开哪一个 TCP 会话时，路由器会在对等体声明中寻找一个最高的 IP 地址，然后断开相应的 TCP 会话。

这个实验中，路由器 harms_co 的自身 IP 地址是 198.100.1.10，远程对等体地址则是 190.10.1.2。因此，它会断开自身始发的 TCP 连接；路由器 green_bay 的 IP 地址是 210.168.1.254，远程对等体地址是 198.100.1.10，因而它也会断开自身始发的 TCP 连接。由于两端都断开了它们认为的最高 IP 地址，这样就终止了连接。例 15-25 是此时 **debug dls w peer** 和 **debug dls w core** 命令的执行结果。

例 15-25 debug 命令显示 TCP 会话的断开

```

harms_co#
02:21:02: DLSw: passive open 190.10.1.1(11005) -> 2065
02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:TCP-RD PIPE OPENED st
ate:DISCONN
02:21:02: DLSw: dtp_action_c() opening write pipe for peer 190.10.1.1(2065)
02:21:02: DLSw: END-TPFSM (peer 190.10.1.1(2065)): state:DISCONN->WWR_RDOP

02:21:02: DLSw: Async Open Callback 190.10.1.1(2065) -> 11006
02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:TCP-WR PIPE OPENED st
ate:WWR_RDOP
02:21:02: DLSw: dtp_action_i() write pipe opened for peer 190.10.1.1(2065)
02:21:02: DLSw: END-TPFSM (peer 190.10.1.1(2065)): state:WWR_RDOP->WAIT_CAP

02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:SSP-CAP MSG RCVD stat
e:WAIT_CAP
02:21:02: DLSw: dtp_action_j() cap msg rcvd from peer 190.10.1.1(2065)
02:21:02: DLSw: Recv CapExId Msg from peer 190.10.1.1(2065)
02:21:02: DLSw: Unknown CV D9 with length 3 from peer 190.10.1.1(2065)
02:21:02: DLSw: Pos CapExResp sent to peer 190.10.1.1(2065)
02:21:02: DLSw: CapExId Msg sent to peer 190.10.1.1(2065)
02:21:02: DLSw: END-TPFSM (peer 190.10.1.1(2065)): state:WAIT_CAP->WAIT_CAP

02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:SSP-CAP MSG RCVD stat
e:WAIT_CAP
02:21:02: DLSw: dtp_action_j() cap msg rcvd from peer 190.10.1.1(2065)
02:21:02: DLSw: Recv CapExPosRsp Msg from peer 190.10.1.1(2065)
02:21:02: DLSw: END-TPFSM (peer 190.10.1.1(2065)): state:WAIT_CAP->WAIT_CAP

02:21:02: DLSw: Processing delayed event:SSP-CAP EXCHANGED - prev state:WAIT_CAP
02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:SSP-CAP EXCHANGED sta
te:WAIT_CAP
02:21:02: DLSw: dtp_action_k() cap xchged for peer 190.10.1.1(2065)
02:21:02: DLSw: closing read pipe tcp connection for peer 190.10.1.1(2065)
02:21:02: DLSw: END-TPFSM (peer 190.10.1.1(2065)): state:WAIT_CAP->PCONN_WT

02:21:02: DLSw: Processing delayed event:TCP-PEER CONNECTED - prev state:PCONN_W
T
02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:TCP-PEER CONNECTED st
ate:PCONN_WT
02:21:02: DLSw: dtp_action_m() peer connected for peer 190.10.1.1(2065)
02:21:02: DLSw: END-TPFSM (peer 190.10.1.1(2065)): state:PCONN_WT->CONNECT

02:21:02: DLSw: dls w tcpd_fini() for peer 190.10.1.1(2065)
02:21:02: DLSw: START-TPFSM (peer 190.10.1.1(2065)): event:ADMIN-CLOSE CONNECTIO
N state:CONNECT
  
```

(待续)

```
02:21:02: DLSw: dtg_action(1) close connection for peer 190.10.1.1(2065)
02:21:02: DLSw: END_IPFSW (peer: 190.10.1.1(2065)); state:CONNECT->DISCONN
02:21:03: DLSw: freeing 190.10.1.1
```

这个问题的解决方法是使 DLSw 链路两端对地址的大小看法一致。例如本例中，不转换成 190.10.1.0/24，而是一个高于 198.100.1.10 的 IP 地址，如 199.100.1.0/24。这样，路由器 green_bay 上的本地对等体会高于另外两个对等体，即使是经过 NAT 转换之后也是如此。更快捷的方法是为路由器 harms_co 添加一个低于 190.10.1.1 的环路接口。例如，添加一个 IP 地址为 100.100.1.1 的环路接口并把它作为新的本地对等体，如例 15-26 所示。

例 15-26 将一个环路接口配置为本地对等体

```
dls local-peer peer-id 100.100.1.1 promiscuous
dls bridge-group 1
!
interface Loopback20
 ip address 100.100.1.1 255.255.255.0
```

现在，可以为路由器 green_bay 加上指向 100.100.1.1 的一个新远程对等体，这样 DLSw 对等体可以通过 NAT 转换相互连接。

第 16 章

热备份路由选择协议 (HSRP) 的使用

有人认为 Cisco 的热备份路由选择协议 (HSRP) 叫做 *热备份默认网关 (HSDG)* 可能更为贴切，因为这正是 HSRP 提供给本地局域网中主机的服务。HSRP 给人的印象似乎更多是“备份”，而实际上 HSRP 并不提供备份功能。但是很多时候准确的解释 HSRP 到底能为客户网络做些什么要比赋予它一个好的名字更有必要。

HSRP 能够为 IP 和 IPX (有限制) 提供一个始终可达的网络层地址。这样即使出现故障，也能够由某种形式的冗余度给与保障。HSRP 最常见的应用是在 LAN 环境中两台路由器“共享”一个公共的地址的情况。称为 *热备份地址* 的主机地址用作某局域网段所有本地主机的默认网关。主路由器用于接收发往热备份地址的数据。备份路由器则忽略这些数据，除非某些设置好的条件满足之后使得路由器的 *HSRP 优先级* (或称 *备用优先级*) 发生了改变。备份路由器的优先级超过主路由器优先级之后，会向主路由器发送成为热备份路由器的请求。工作站具有一个指向热备份地址的默认网关。图 16-1 是一个常见的 HSRP 的配置情况。

在图 16-1 中，所有 TCP/IP 客户端都把网关设置为 172.16.1.1。这个默认网关使得 IP 客户端把无法在本地子网中到达的数据转发到指定的 IP 地址。地址 172.16.1.1 作为 *HSRP 虚拟地址* 使用。在这个例子中，路由器 caladan 用于跟踪路由器 arakas 的串行接口。如果这个接口进入关闭 (down) 状态，它会把相应路由器的优先级减去 10 或其他相应设置值。串行接口开始工作之后，caladan 会接收和转发工作站来的 IP 数据包。如果 caladan 上的串行接口停止工作，giedi_prime 会成

为主 HSRP 路由器，立即开始对工作站来的 IP 数据包进行服务，从而保持了所有活动会话的继续进行。路由器 caladan 的工作停止对局域网段的工作站来说是完全透明的。

在主机要把数据默认发送到某个网络地址去的情况下，HSRP 的作用非常有效，如图 16-1 所示。

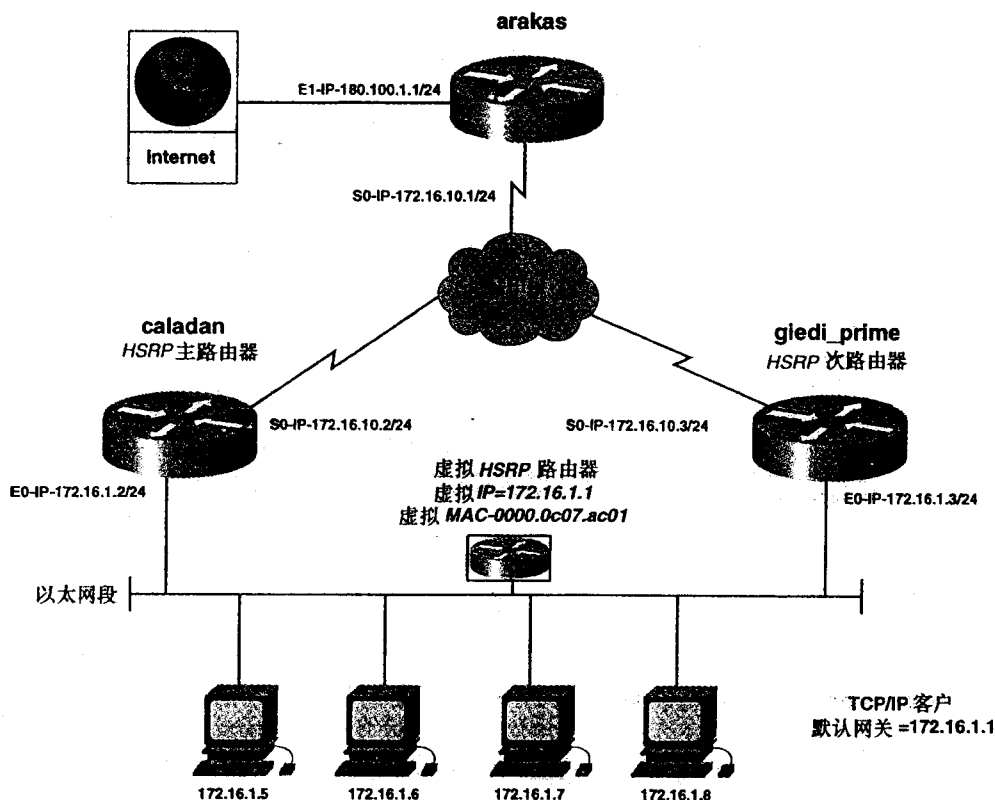


图 16-1 典型网络环境中的 HSRP

16.1 HSRP 的概览与配置

HSRP 利用组播在同一备份组中的路由器之间交流配置好的优先级。优先级定义了组中的主路由器和备份路由器。默认优先级是 100，优先级最高的路由器会成为组中的主路由器。如果优先级相同，首先进入工作状态的成为主路由器。然后，所有的优先级都以 IP 地址为基础。如果一台具有同样优先级的路由器加入备份组，不会影响原来的主路由器，即使 IP 地址更高也没有关系，但是会影响备份路由器。

在优先级不相同的情况下，如果优先级更高的路由器加入备份组，即使没有设置优先级取代功能，也会立即开始工作。但是，如果路由器已经处在工作状态中，或优先级发生了改变（由于跟踪重配置的缘故），不会影响没有配置优先级取代功能的主路由器。

HSRP 采用了 3 种组播消息来交换备份组信息：

- **Hello**——hello 消息包含发送路由器的优先级与状态信息。每 3 秒进行一次 hello 消息的交换。如果发送路由器无法在指定的时间段里发送 hello 消息，接收路由器（如果优先级允许）会成为组中的主路由器。
- **Coup**——备份路由器成为主路由器之后，往组中的各路由器发送一个 coup 消息。
- **Resign**——主路由器要停止工作时或者接收到优先级高于自己的 hello 消息时，通过发送一个 resign 消息让出自己的主路由器之位。

HSRP 的配置包括下面这些步骤：

第 1 步 选一个虚拟地址作为 HSRP 地址。这个地址必须位于要运行 HSRP 的局域网所获得的地址空间之中。通常把这个地址称为**备用 IP 地址**。备份组中的每台路由器都必须用 `standby group_number ip a.b.c.d` 命令定义同一个虚拟 IP 地址。

备份组号是一个惟一的号码，在以太网或 FDDI 上可以识别 1 到 255 个备份组，在令牌环网中则可以识别 0 到 2 个不同的组。如果不指定组号，默认采用第 0 组。

如果在 VLAN 中继上配置 HSRP，每个 VLAN 或以太网接口都必须在不同的备份组中。

第 2 步 确定主路由器，将该路由器的优先级至少设置为 101。另外再用 `preempt` 命令使这台路由器参与主路由器的选择。这一步的操作可以通过这条命令来完成：

`standby group_number preempt` 和 `standby group_number priority 1-255`

这里：

- 参数 `preempt` 使得组中具有最高优先级的路由器成为主路由器。
- 参数 `priority` 为路由器分配一个优先级，默认情况下是 100，优先级最高的路由器会成为主路由器，或称活动路由器。

第 3 步 用下面的命令配置主路由器的跟踪，认证以及计时器进行配置：

—— `standby group_number track interface_name [cost]`

- 在对某接口进行跟踪时，如果接口该停止工作，HSRP 会把该接口的优先级减去 10。运用这条命令，HSRP 组可以根据接口的工作状态使路由器在主路由器和备份路由器之间进行切换。默认的值 10 可以通过这条命令加以更改。

—— `standby group_number authentication character_string`

这条命令可以创建认证信息以进行 HSRP 的组播操作，这样可以确保只有授权了的路由器才能进入 HSRP 组中。这里的“`character_string`”字符串必须和 HSRP 组中所有的路由器项匹配。

—— `standby group_number timers hello_interval_seconds holddown_timer_seconds`

参数 `timers` 可以设置 hello 消息和抑制计时器的时间长度。抑制计时器是路由器在宣告在转入活动状态之前应该等待的时间。默认值分别是 3 秒和 10 秒。这些计时器设置必须和组中所有路由器相匹配。

—— `standby group_number mac-address H.H.H`

这条命令指定一个 MAC 地址的静态条目，有利于对来自下游设备的 HSRP 地址进行管理 and 过滤。

第 4 步 通过为路由器分配 99 以下的优先级，把备份组中别的路由器配置为备份路由器。

第 5 步 配置备份路由器的优先级取代功能、跟踪、认证以及计时器。

EIGRP 或 OSPF。HSRP 的作用就是在无重传和丢包的情况下，重路由数据的传输。为此，路由器必须具有快速收敛的能力。

16.1.1 在路由器之间配置 HSRP

以图 16-1 为例，在路由器 caladan 和 giedi prime 之间配置 HSRP。这个例子中，以太网段 172.16.1.0/24 上有多个 IP 客户端设备。所有的 TCP/IP 客户需要通过 arakas 来访问 Internet。路由器 caladan 和 giedi prime 可以通过帧中继网络访问 arakas。所有路由器都必须以 EIGRP 为路由选择协议，以便相互交换路由信息。HSRP 使得 IP 客户可以无间断地访问 arakas。

要实现上面这些目的，把路由器 caladan 选作为主路由器，而 giedi prime 则是作为备份路由器，二者之间的虚拟 IP 地址使用 172.16.1.1。客户的最终目的设备是 arakas，因此需要在串行接口上进行跟踪。如果通往 arakas 的连接失败，通过跟踪串行接口可以使 giedi prime 成为主路由器。

注释 跟踪激活时，有两种情况可以使得 giedi prime 成为主路由器。一种情况是到 caladan 的连接失败，如 caladan 上的以太网端口发生了物理失效。另一种情况是 arakas 和 caladan 之间的物理连接丢失导致 caladan 上的串行口停止工作。

首先配置 caladan。往 E0 接口加上一个备份组。由于 caladan 将要作为主路由器，因此应为它设置一个高于 100 的优先级，这里采用的是 105。对串口进行跟踪，默认跟踪成本是 10，因此如果 caladan 路由器的串行连接失败，其 HSRP 优先级是 95。注意这一数值，后面需要把 giedi prime 的优先级设为大于 95 而小于 105 的一个数值。例 16-1 是 caladan 的配置情况。

例 16-1 HSRP 主路由器的配置

```
caladan(config)#interface ethernet 0
caladan(config-if)#standby 1 ip 172.16.1.1
caladan(config-if)#standby 1 priority 105
caladan(config-if)#standby 1 preempt
caladan(config-if)#standby 1 track s0
```

HSRP 组的某个成员从备用状态进入到活动状态时会产生下面这些信息：

```
01:10:14: %STANDBY-6-STATECHANGE: Standby: 1: Ethernet0 state Speak
-> Standby
01:10:14: %STANDBY-6-STATECHANGE: Standby: 1: Ethernet0 state Standby
-> Active
```

配置路由器 giedi_prime 时，备份组的设置应该和 caladan 的备份组 1 相同，虚拟 IP 地址也和 caladan 一样。参数 **preempt** 使 giedi prime 路由器在本身的优先级超过 caladan 的之后成为主路由器。这里最重要的可能是 **priority** 参数。在主路由器上，设其优先级为 105 并且对串口进行了跟踪。如果 caladan 丧失主路由器地位，优先级会成为减去 10 成为 95。因此，备份路由器 giedi_prime 的优先级应该比 95 大。这个例子中用的是 101。这个 101 也使得备份路由器的优先级比任何新加入的路由器的优先级要高，因为新加进来的路由器的优先级默认是 100。

例 16-2 是路由器 giedi prime 的配置示例。

例 16-2 HSRP 备份路由器的配置示例

```
giedi_prime(config)#interface ethernet 0
giedi_prime(config-if)#standby 1 ip 172.16.1.1
giedi_prime(config-if)#standby 1 priority 101
giedi_prime(config-if)#standby 1 preempt
giedi_prime(config-if)#standby 1 track s0
```

用 **show standby** 命令可以验证 HSRP 的功能。这条命令能够显示主路由器以及主路由器是否设置了优先级取代功能、该组中虚拟 IP 地址和 MAC 地址的情况。例 16-3 是 **show standby** 命令在主路由器 caladan 和备份路由器 giedi prime 上的执行示例。

例 16-3 主路由器和备份路由器上 show standby 命令的执行示例

```
caladan#show standby
Ethernet0 - Group 1
  Local state is Active, priority 105, may preempt ← Active=Primary router
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:02.496
  Hot standby IP address is 172.16.1.1 configured ←Virtual IP address
  Active router is local
  Standby router is 172.16.1.3 expired
  Standby virtual mac address is 0000.0c07.ac01
  Tracking interface states for 1 interface, 1 up:
    Up Serial0
caladan#

giedi_prime#show standby
Ethernet0 - Group 1
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:02.540
  Hot standby IP address is 172.16.1.1 configured
  Active router is 172.16.1.2 expires in 00:00:09
  Standby router is local
  Tracking interface states for 1 interface, 1 up:
    Up Serial0
giedi_prime#
```

要加入认证功能，只需在以太接口上加上 **standby 1 authentication password** 命令。这时要确保组中所有的路由器也都设置了认证。例 16-4 是整个配置的过程。

例 16-4 caladan

```
hostname caladan
!
<<<text omitted>>>
!
interface Ethernet0
ip address 172.16.1.2 255.255.255.0
```

(待续)

```

no ip redirects      -this is added by the router when standby is enabled
no ip directed-broadcast
standby 1 priority 105
standby 1 preempt
standby authentication cisco - cisco is the password and is case sensitive
standby 1 ip 172.16.1.1
standby 1 track Serial0
!
interface Serial0
ip address 172.16.10.2 255.255.255.0
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
no fair-queue
frame-relay map ip 172.16.10.1 21 broadcast
frame-relay map ip 172.16.10.3 21 broadcast
!
router eigrp 2001
network 172.16.0.0
!

hostname giedi_prime
!
<<<text omitted>>>
!
interface Ethernet0
ip address 172.16.1.3 255.255.255.0
no ip redirects
delay 0000000 - influence EIGRP to not load-share
standby 1 priority 101
standby 1 preempt
standby authentication cisco - cisco is the password and is case sensitive
standby 1 ip 172.16.1.1
standby 1 track Serial0
!
interface Serial0
ip address 172.16.10.3 255.255.255.0
encapsulation frame-relay
no fair-queue
frame-relay map ip 172.16.10.1 31 broadcast
frame-relay map ip 172.16.10.2 31 broadcast
!
router eigrp 2001
network 172.16.0.0
!

```

16.2 HSRP 的 “Big show” 和 “Big D” 命令

仍然采用第一个例子，本节讨论 HSRP 的 **show** 命令和 **debug** 命令的用法：

```
show standby {brief | interface}
debug standby
```

命令 **show standby** 显示某个接口是在备用状态还是在活动状态。活动状态说明接口或路由器是主路由器，备用则说明路由器是次级或称备份的。命令 **show standby** 还能给出所用的 hello 计时器，虚拟 MAC 地址以及与跟踪相关的信息。例 16-5 是该命令在路由器 caladan 上的执行示例。

例 16-5 show standby 命令的执行示例

```
caladan#show standby
Ethernet0 - Group 1
  Local state is Active, priority 105, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:02.496
  Hot standby IP address is 172.16.1.1 configured
  Active router is local
  Standby router is 172.16.1.3 expired
  Standby virtual mac address is 0000.0c07.ac01
  Tracking interface states for 1 interface, 1 up:
    Up Serial0
caladan#
```

命令 **debug standby** 可以显示 hello 计时器和抑制计时器的设置情况和备份组的信息，如哪台路由器在活动中，活动与备份路由器的优先级等。例 16-6 是 **debug standby** 命令在路由器 giedi prime 上的执行示例。

例 16-6 路由器 giedi prime 上 debug standby 命令的执行示例

```
giedi_prime#debug standby
SB1:Ethernet0 Hello in 172.16.1.2 Active pri 105 hel 3 hol 10 ip 172.16.1.1
SB1:Ethernet0 Hello out 172.16.1.3 Standby pri 95 hel 3 hol 10 ip 172.16.1.1
SB1:Ethernet0 Hello in 172.16.1.2 Active pri 105 hel 3 hol 10 ip 172.16.1.1
SB1:Ethernet0 Hello out 172.16.1.3 Standby pri 95 hel 3 hol 10 ip 172.16.1.1
SB1:Ethernet0 Hello in 172.16.1.2 Active pri 105 hel 3 hol 10 ip 172.16.1.1
SB1:Ethernet0 Hello out 172.16.1.3 Standby pri 95 hel 3 hol 10 ip 172.16.1.1
```

16.3 实验 33：配置 HSRP、跟踪与非对称路由——

第 1 部分

16.3.1 实验说明

HSRP 是为 TCP/IP 提供具有良好容错性默认网关的一种有效途径。HSRP 主要用于 IP 主机设置静态默认网关的 IP 网络环境中。如果在网络中具有冗余路由器，但 IP 客户端仍然会将 IP 数据包转发到默认网关地址，即使该默认网关关闭也照样转发。即使有一条可行的离开这个网络的路径，但是由于客户端只知道怎样转发数据包去默认网关，因此客户端会传送失败。在这种情况下，HSRP 提供了一个可由很多路由器共享的默认网关地址，从而解决了冗余网络的这个用户漏洞问题。

16.3.2 实验内容

假设 4th Army Com Net 公司在它的公司总部和 Charlie 分公司之间运行着一个帧中继网

络。Charlie 分公司处的工作站基于 IP，并且它们需要对公司总部的主服务器和备用服务器进行不间断访问。公司之间传输的数据都非常重要，如果一台路由器出现故障，另一台需要取代那台故障路由器的功能。在为这个网络进行设计时要遵循下面这些要求：

- Charlie 分公司的工作站都基于 IP，具有一个指向 10.25.61.3 的静态默认网关。对这个网络进行配置，即使路由器 charlie_1 或 charlie_2 出现故障，这些工作站依然可以不间断地访问总部路由器 headquarters_co。这里的 charlie_1 应为主路由器。
- 如果 charlie_1 或 charlie_2 的串行接口出现故障，要确保出现故障的路由器不成为主路由器。

16.3.3 实验目的

- 按照图 16-2 对网络进行配置，路由选择协议采用 RIP V2。对 RIP 进行配置以进行具有 HSRP 主路由器的非对称路由。即如果 charlie_1 是主路由器，数据的流动是从各个工作站到 charlie_1，再到 headquarters_co，然后再沿着 charlie_1 返回各个工作站。除非 charlie_2 成为 HSRP 的主路由器，否则数据不从 headquarters_co 转发到 charlie_2。
- 在 charlie_1 和 charlie_2 之间配置 HSRP。将 charlie_1 配置为 HSRP 的主路由器。
- 在串行接口上实施跟踪。
- 可选：对 Charlie 分公司的网络加以改善，通过采用 EIGRP 作为路由选择协议提高 HSRP 环境的工作效率。

16.3.4 所需设备

- 4 台 Cisco 路由器，3 台作为网络中使用的路由器，1 台作为帧中继交换机。帧中继交换机需要具有 3 个串行端口。路由器通过 V.35 背对背线缆或类似方式连接在一起。
- 通过集线器或交换机创建的 2 个 LAN 网段。
- 可选：两台 IP 工作站，一台作为 Charlie 分公司的工作站，另一台作为总部服务器。

16.3.5 物理设计与实验准备

- 按照图 16-2 将集线器以及串行线缆与路由器相连。将帧中继交换机配置在多点环境中以便提供一条从 headquarters_co 到 charlie_1 和 charlie_2 的 PVC。这里的图中没有表示出帧中继交换机的配置情况。
- 按照图 16-2 将两台以太网集线器与路由器相连以形成两个 LAN 网段，其中一个网段连接 charlie_1 和 charlie_2，另一个则是 headquarters_co 以外的所有网络部分。
- 按照图 16-2 对两台 IP 工作站进行连接与配置，默认网关设为 25.100.61.3。
- 按照图中所示对路由器 headquarters_co 进行配置，路由选择协议采用 RIP V2。
- 可选：改善 Charlie 分公司的网络，通过采用 EIGRP 作为路由选择协议提高其在 HSRP 环境中的工作效率。

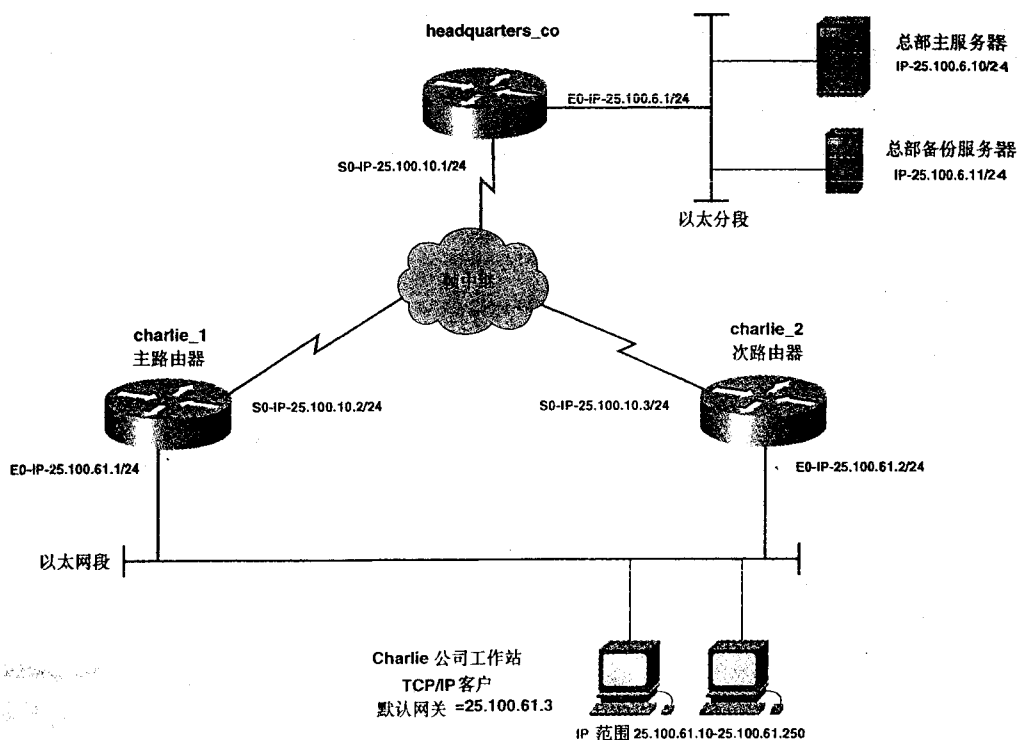


图 16-2 4th Army Com Net 网络

16.4 实验 33：配置 HSRP、跟踪与非对称路由—— 第 2 部分

16.4.1 实验步骤

物理连接完成，帧中继交换机也配置妥当后，可以着手在所有路由器之间建立 IP 连接。
例 16-7 是实验中所需的帧中继的配置情况。

例 16-7 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
```

```

no fair-queue
clockrate 148000
frame-relay intf-type dce
frame-relay route 131 interface Serial1 31
frame-relay route 121 interface Serial3 21
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay intf-type dce
frame-relay route 31 interface Serial0 131
!
interface Serial2
no ip address
shutdown
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 21 interface Serial0 121
!

```

首先是路由器 `headquarters_co`，需要配置以太接口和串行接口的 IP 地址。这里要配置多点帧中继网络，因此需要 `frame-relay map` 命令，利用这个 `map` 命令来指向路由器 `charlie_1` 和 `charlie_2` 的 IP 地址，如下所示：

```

frame-relay map ip 25.100.10.2 121 broadcast
frame-relay map ip 25.100.10.3 131 broadcast

```

接下来，配置 RIP V2 路由选择协议，如例 16-8 所示。RIP-2 的配置通过在路由选择协议下加上 `version 2` 参数来实现。这个例子中，还需加上 `distance` 参数，这是为了把主路由指向路由器 `charlie_1`，`charlie_2` 的管理距离是 125，比通常的 RIP 距离（120）大 5。这样，所有的出站数据会先通过路由器 `charlie_1`，使网络形成了不对称的路由方式。

例 16-8 配置 RIP-2

```

router rip
version 2
network 25.0.0.0
distance 125 0.0.0.3 255.255.255.0

```

4th Army Com Net 网络中没有运行 Cisco IOS 12.0 的所有路由器上也需配置同样的命令。这些配置完成之后，配置串行接口和以太接口收发 RIP-2 更新信息，所需的命令包括：

```

ip rip send version 2
ip rip receive version 2

```

和前面一样 4th Army Com Net 网络中所有的路由器上也必须配置这组命令。路由器 `headquarters_co` 配置完成之后，其结果如例 16-9 所示。

例 16-9 路由器 headquarters_co 的配置

```

hostname headquarters_co
!
interface Ethernet0
 ip address 25.100.6.1 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
 media-type 10BaseT
!
interface Serial0
 ip address 25.100.10.1 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 25.100.10.2 121 broadcast
 frame-relay map ip 25.100.10.3 131 broadcast
!
<<<text omitted>>>
!
router rip
 version 2
 network 25.0.0.0
 distance 125 0.0.0.3 255.255.255.0
!
headquarters_co#

```

现在对 Charlie 分公司的路由器进行配置，首先为所有路由器的以太和串行接口分配 IP 地址。如前所述，每台路由器还需要配置 RIP 2，具体步骤如前面所讨论的内容。由于该网络是一个多点帧中继网络，还需要加上 **frame-relay map** 命令。完成了每台路由器的 IP 配置之后，其配置情况如例 16-10 所示。

例 16-10 charlie_1 和 charlie_2 的 IP 配置

```

hostname charlie_1
!
<<<text omitted...
!
interface Ethernet0
 ip address 25.100.61.1 255.255.255.0
 no ip directed-broadcast
 ip rip send version 2
 ip rip receive version 2
!
interface Serial0
 ip address 25.100.10.2 255.255.255.0
 no ip directed-broadcast
 ip rip send version 2
 ip rip receive version 2
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay map ip 25.100.10.1 21 broadcast
 frame-relay map ip 25.100.10.3 21 broadcast
 frame-relay lmi-type cisco
!
<<<text omitted>>>
!

```

```

router rip
  version 2
  network 25.0.0.0
!
<<<text omitted>>>

charlie_1#

hostname charlie_2
!
interface Ethernet0
  ip address 25.100.61.2 255.255.255.0
  ip rip send version 2
  ip rip receive version 2
!
interface Serial0
  ip address 25.100.10.3 255.255.255.0
  ip rip send version 2
  ip rip receive version 2
  encapsulation frame-relay
  frame-relay map ip 25.100.10.1 31 broadcast
  frame-relay map ip 25.100.10.2 31 broadcast
!
<<<text omitted>>>
router rip
  version 2
  network 25.0.0.0
!
    
```

建立了端对端的 IP 连接之后，开始 HSRP 的配置。按照 HSRP 的配置步骤，首先定义 HSRP 或虚拟路由器的 IP 地址。Charlie 分公司的所有路由器都指向默认网关地址 25.100.61.3。因而这个地址就是 HSRP 地址。此外，每台路由器上还需要用 **preempt** 命令和 **priority** 命令进行配置。

由于 charlie_1 是作为主路由器的，因而优先级应该设置为大于默认优先级 100。在设置其优先级之前，还要考虑该路由器不担当主路由器的条件。在这个例子中，除非连接失效，否则 charlie_1 就是主路由器。如果 charlie_1 的优先级设置为 105，而 charlie_1 和 charlie_2 的接口默认跟踪成本设为 10，这样就会导致串行连接失效的优先级就是 95。因此，charlie_2 的优先级应该设置在 95 和 105 之间。例 16-11 是路由器 charlie_1 的相关配置情况。

例 16-11 charlie_1 路由器的 HSRP 配置

```

charlie_1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
charlie_1(config)#interface ethernet 0
charlie_1(config-if)#standby 1 ip 25.100.61.3
charlie_1(config-if)#standby 1 preempt
charlie_1(config-if)#standby 1 priority 105
04:08:11: %STANDBY-6-STATECHANGE: Standby: 1: Ethernet0 state Speak -> Standby
04:08:11: %STANDBY-6-STATECHANGE: Standby: 1: Ethernet0 state Standby -> Active
charlie_1(config-if)#standby 1 track serial 0
    
```

配置 HSRP 时，如果 HSRP 进入工作状态，会发送一个状态改变消息。

“active”状态的路由器，要确保处在“active”状态的路由器都为“local”，同时也注意一下显示的主机备份 IP 地址是否为配置的那一个。

例 16-12 主路由器的状态

```
charlie_1#show standby ethernet 0
Ethernet0 - Group 1
  Local state is Active, priority 105, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:00.678
  Hot standby IP address is 25.100.61.3 configured
  Active router is local
  Standby router is 25.100.61.2 expired
  Standby virtual mac address is 0000.0c07.ac01
  Tracking interface states for 1 interface, 1 up:
    Up Serial0
charlie_1#
```

除了优先级之外，charlie_2 的配置和 charlie_1 完全一样。路由器 charlie_2 的优先级应该大于默认值 100。将其设为 101 比较理想，表明该路由器仅次于主路由器。这样也有助于打破新路由器加入到 HSRP 组中可能产生的优先级平等条件。完成这一配置之后，用 **show standby Ethernet 0** 命令验证 charlie_2 的 HSRP，如例 16-13 所示。

例 16-13 备用路由器的状态

```
charlie_2#show standby ethernet 0
Ethernet0 - Group 1
  Local state is Standby, priority 101, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:01.336
  Hot standby IP address is 25.100.61.3 configured
  Active router is 25.100.61.1 expires in 00:00:09
  Standby router is local
  Tracking interface states for 1 interface, 1 up:
    Up Serial0
charlie_2#
```

本例中，希望验证本地路由器的状态为备用，并且在优先级允许的情况下，该备用路由器可以实施优先级占用，即接替活动路由器的工作。例 16-14 是 charlie_1 和 charlie_2 的以太网接口上 HSRP 的配置。

例 16-14 charlie_1 和 charlie_2 的 HSRP 配置

```
charlie_1#
interface Ethernet0
ip address 25.100.61.1 255.255.255.0
no ip redirects
no ip directed-broadcast
ip rip send version 2
ip rip receive version 2
standby 1 priority 105
standby 1 preempt
standby 1 ip 25.100.61.3
standby 1 track Serial0
```

(待续)

```

charlie_2#
interface Ethernet0
 ip address 25.100.61.2 255.255.255.0
 no ip redirects
 ip rip send version 2
 ip rip receive version 2
 standby 1 priority 101
 standby 1 preempt
 standby 1 ip 25.100.61.3
 standby 1 track Serial0
|

```

为了测试 HSRP 的功能，按图 16-2 所示连接一台 IP 工作站。此时要保证工作站的默认网关指向 25.100.61.3。如果做了相应的配置，这台工作站上可以 ping 通路由器 headquarters_coE0 端口的 IP 地址，即总部服务器。现在再测试一下连接失效时备用路由器的工作过程。在帧中继交换机上关闭与路由器 charlie_1 相连的串行接口。HSRP 检测到该接口不在工作状态，将这个接口的优先级减去一个相应的数值——这里是默认的 10。这样会使 charlie_1 的优先级变为 95，小于 charlie_2 的优先级，因而 charlie_2 会接替 charlie_1 的工作进入工作状态。要验证该过程，可以在工作站上跟踪路由，并执行 **show standby interface** 命令。工作站上关闭了帧中继交换机与路由器 charlie_1 相连的串行接口之后，在 charlie_2 上执行 **show standby interface** 命令的结果如例 16-15 所示，该路由器的状态从“standby”变为“active”。

例 16-15 连接失效并进行备份之后，show standby 命令的执行情况

```

charlie_2#show standby ethernet 0
Ethernet0 - Group 1
  Local state is Active, priority 101, may preempt
  Hellosent 3 holdtime 10
  Next hello sent in 00:00:00.952
  Hot standby IP address is 25.100.61.3 configured
  Active router is local
  Standby router is 25.100.61.1 expires in 00:00:07
  Tracking interface states for 1 interface, 1 up:
    Up Serial0
charlie_2#

```

此外，还要确保 IP 路由的正常，这是通过从工作站发送 ping 测试数据包和查看路由器 headquarters_co 的路由表来实现的。例 16-16 就是连接失效备份前后路由器 headquarters_co 的路由表的情况。这里先是显示 RIP 更新信息来自主路由器 25.100.10.2，在完成备份之后，路由更新信息就是来自 25.100.10.3 的了，此时的管理距离是 125。

例 16-16 连接失效备份前后的 IP 路由表

```

headquarters_co#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
<<<text omitted>>>

25.0.0.0/24 is subnetted, 3 subnets

```

(待续)

```

R 25.100.61.0/24 via 25.100.10.2, 00:00:01, Serial0
C 25.100.10.0 is directly connected, Serial0
C 25.100.6.0 is directly connected, Ethernet0
headquarters_co#

I AFTER WE DOWN THE FRAME INTERFACE WE HAVE THE FOLLOWING:
headquarters_co#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
<<<text omitted>>>

Gateway of last resort is not set

25.0.0.0/24 is subnetted, 3 subnets
R 25.100.61.0/24 is possibly down,
  routing via 25.100.10.2, Serial0
C 25.100.10.0 is directly connected, Serial0
C 25.100.6.0 is directly connected, Ethernet0
headquarters_co#

headquarters_co#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
<<<text omitted>>>

Gateway of last resort is not set

25.0.0.0/24 is subnetted, 3 subnets
R 25.100.61.0/24 via 25.100.10.3, 00:00:17, Serial0
C 25.100.10.0 is directly connected, Serial0
C 25.100.6.0 is directly connected, Ethernet0
headquarters_co#

```

RIP-2 从主路由器到备用路由器的收敛时间可能是几分钟。这是本实验可选部分要解决的问题。为了改善 The 4th Army Com Net 网络的设计，现在把路由选择协议从 RIP-2 改为 EIGRP 或 OSPF。路由选择协议的改变加快了 IP 收敛速度，反过来又更好地支持了 HSRP 功能。

实验中选用 EIGRP 是因为从 RIP 过渡到 EIGRP 非常容易。过渡到 EIGRP 的最佳途径是加入一条含有自治系统 ID 和网络地址的 **router eigrp** 命令。由于 EIGRP 的管理距离小于 RIP，因而路由表会自动从 RIP 收敛到 EIGRP，如例 16-17 所示。

例 16-17 路由器 headquarters_co 上 EIGRP 的 IP 路由表

```

headquarters_co#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
<<<text omitted>>>

Gateway of last resort is not set

25.0.0.0/24 is subnetted, 3 subnets
D 25.100.61.0 [90/2195456] via 25.100.10.2, 00:00:02, Serial0
C 25.100.10.0 is directly connected, Serial0
C 25.100.6.0 is directly connected, Ethernet0
headquarters_co#

```

为了影响 EIGRP 宣告的路由，可以在 charlie_2 的以太网段上加入 DELAY。这样，EIGRP

向路由器 `headquarters_co` 宣告的路由被加权。这又使得 EIGRP 选用通过路由器 `charlie_1` 去往子网 `25.100.61.0/24` 的路由。DELAY 一定要加到以太接口而不是串行接口上，因为从路由器 `charlie_2` 发往总部主服务器的数据还需要通过串行接口发送。如果延时（DELAY）放置在这个接口上，EIGRP 会把这类数据通过以太端口首先发送到 `charlie_1`。配置完 EIGRP 之后，用 `no router rip` 命令删去配置中的 RIP 部分，该命令将删去所有与 RIP 相关的内容。

第 17 章

网络时间协议 (NTP) 与简单网络时间协议 (SNTP) 的配置

Bob 最近都快为他的网络连接慢的问题发疯了。这个问题和他一星期前遇到的问题很相似。Bob 并不是他的真名，但是由于是新来的，所有的人都这么叫他。Bob 知道电信领域几乎所有的首字母缩略词，他还知道 ATM 比以太网快很多。在他那个乱得像一锅粥似的网络中，Bob 最后找到了他认为的“确凿证据”。他要彻底解决这个网络连接慢的问题，他要向所有的人证明自己的价值。小心翼翼地跟踪仅有的那一点线索，Bob 发现了两处连接失败的地方，时戳分别是 12:01:21 OCT 18, 1999 和 11:23:40 OCT 18, 1999。迅速地查看远程站点的日志之后，Bob 发现了 105 处失效的地方，时间都是在 11:57:42.079 UTC Mar 1 1993 和 03:32:12.022 UTC Mar 17 之间。

很不幸的是，Bob 不得不再等一天，等另一次出现失败时来进行他的调查取证。下一次的情况就会不一样了，因为 Bob 在他的缩略词清单里加上了网络时间协议 (NTP) 的概念。

17.1 NTP 技术概览

NTP 在无管理的全球 Internet 环境中为远程设备提供了一个精确稳定的时钟。在 NTP 之前，其他协议也提供了类似的服务，如日间协议、时间协议以及 ICMP 时间戳等。数字

时间服务（DTS）也实现了不少 NTP 的功能。但是，NTP 提供了层（stratum）的概念用于时钟的选择和精确的补偿措施用于自身时钟频率误差的校正。相比之下，DTS 就没有使用 stratum 和时钟频率补偿方法。

Stratum 的概念是直接来自 BELL 86 的电话技术中引申而来。每一台 NTP 服务器的准确度用一个 stratum 号码来定义，最精确的服务器从 stratum 为 1 开始，依此基础叠加。Stratum 使 NTP 可以从多个时钟源中进行选择，以决定与哪一个进行同步。

Stratum 为 1 的精确度需要由原子钟来提供。显然，我们无法配置或提供 stratum 为 1 的时钟精度的 Cisco 路由器。

注释 原子钟使用的晶体振荡器是能够与自然现象相互协调，从而可以维持非常精确的频率。最古老、最根本的晶体振荡器是天体。但是由于天体的巨大以及对天体知识的缺乏，科学家们选择原子钟的基准时没选择行星和太阳系的轨道状态，而是电子的轨道状态。原子提供了一个科学家们可以准确测量的稳定而精确的时钟模型。原子振荡器是以氢、铯和铷原子的状态转换为基础的。

注释 共享软件 TARDIS2000 V1.2 可对 NTP 进行测试，可从网站 <http://download.cnet.com> 下载该软件。在模型中可使用的两个公共 NTP/SNTP 时钟源的地址是 zeus.tamu.edu（IP 地址 128.194.103.14）和 tmc.edu（IP 地址 128.249.1.1）。

在和原子钟同步时钟时，多数的原子钟能够为客户提供 stratum 为 3 或更高的时钟精度。在使用 NTP 时，把 stratum 等于 3 的精度作为相当可靠的 stratum 级别。

NTP 主要用来提供下面这 3 种类型的参数：

- **时钟偏移量**——时钟补偿量是要把本地时钟与参考时钟相互同步时，本地时钟需要做的调整量。
- **往返延时**——往返延时使得参考时钟有可能在某个指定时间发送一个最后会回到参考时钟处的消息。
- **差量**——差量是本地时钟相对于参考时钟或 NTP 服务器的最大误差值。

这些参数都与本地时钟有关，采用格林尼治标准时间（GMT），世界调整时间（UTC）或协调通用时间（CUT）作为公共的参考时间。不可靠的本地时钟会对 NTP 同步造成影响，进而影响 NTP 提供的数据。NTP 的这些工作都是利用 UDP 的端口 123 来完成的。数据的完整性是由 UDP 校验来保证的，没有必要再提供数据流的控制以及任何重发措施。

NTP 应用在很多方面，最常见的是使一个 NTP 客户端通过 IP 从外部时间源获取有效的时钟。如果多个客户都通过一个时间源进行同步，就可以实现整个网络的同步。网络的同步有这样一些有用的功能：

- 就像 Bob 的悲惨故事，NTP 在设法解决一些网络不正常现象（如连接或相邻关系失常等）时非常有用。
- 网络时钟同步有助于获取多个路由器时间精确的日志记录以及调试信息。
- 网络管理平台（如 CiscoWorks 和 HP OpenView）能够更为有效，更为准确地记录

网络统计信息。

NTP 首先出现在 RFC 958 中，自从问世以来已经经历了很多次演变。NTP V.3 是目前 NTP 的主导版本。RFC 1305 中定义的 NTP 使得 RFC 1119、1059 和 958 都成为过时的事物。RFC 2030 提出了简单网络时间协议 (SNTP)，它是在 NTP V.3 的基础上改编而成的。SNTP V.4 的惟一显著变化是能够正确解译 IPv6 的报头和 OSI 寻址方式。

注释 网站 www.isi.edu/in-notes/rfcxxxx.txt 上可以查到所有 RFC 内容，这里的 xxxx 是 RFC 号码。

17.2 NTP 的配置

NTP 通过配置可以支持不同的网络环境。下面是最常见的 NTP 应用，本章会对其进行配置：

- **NTP 广播客户端模式**——路由器可以配置成被动监听 NTP 广播，这样可以不用静态地指定某个特定的时间服务器。
- **NTP 静态客户模式**——路由器可以配置成监听静态指定的 NTP 服务器的信息和与之交换信息。
- **NTP 主模式**——路由器配置成用于转发 NTP 广播的一台 NTP 服务器。
- **NTP 对等体关系**——路由器可以配置为与另一台路由器建立 NTP 对等体关系。路由器既能以其他系统为准进行同步，也可以作为其他系统进行同步的基准。
- **NTP 的可选项以及与时间相关的配置**——NTP 的可选项包括认证以及日历设置。而与时间相关的设置则包括夏令时和当前时区的设置。

17.2.1 NTP 广播客户端模式的设置

Cisco 路由器可以配置成在接口上接收 NTP 网络广播。这种类型的设置应该用于局域网环境中避免设置多个 NTP 服务器。在距离 NTP 服务器最近的接口或者是接收 NTP 广播的接口上使用 `ntp broadcast client` 命令就可以将路由器配置成接收 NTP 广播的状态。图 17-1 是配置了一台 NTP 服务器的局域网示例。NTP 服务器的 stratum 为 5，IP 地址是 206.191.241.44。

例 17-1 给出了在路由器的以太 0 接口上接收 NTP 广播所需的配置。

例 17-1 NTP 广播客户模式的配置

```
ntp_client(config)#int ethernet 0
ntp_client(config-if)#ntp broadcast client
ntp_client(config-if)#exit
ntp_
```

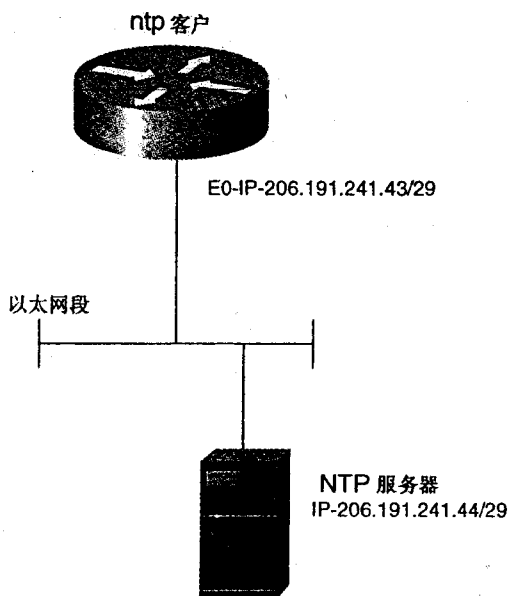


图 17-1 NTP 广播客户模式

通过 **show ntp associations** 命令和 **show ntp status** 命令可以验证时钟是否已经同步。例 17-2 和 17-3 是这两条命令在路由器 ntp_client 上的执行示例。

例 17-2 show ntp associations 命令示例

```
ntp_client#show ntp associations

address      ref clock    st when poll reach delay offset disp
* 206.191.241.44 128.194.103.14 5 7 8192 76 4.14 0.05 0.05
* master (syncd), # master (unsyncd), + selected, - candidate, - configured
ntp_client#
```

例 17-2 突出显示了一些很关键的参数值。NTP 同步之后，有一个星号 (*) 表示路由器已经从紧跟星号之后的地址处接收到了 UDP 数据包，说明该路由器已经和 NTP 服务器同步完毕。字段 st 说明该时钟源是 stratum 为 5 的标准。字段 ref clock (参考时钟) 是 NTP 源与之同步的时钟。如果该数为 127.127.7.1 且设备是一台 Cisco 路由器，那这个时钟是和自身进行同步。

例 17-3 show ntp status 命令的执行情况

```
ntp_client#show ntp status
Clock is synchronized, stratum 6, reference is 206.191.241.44
nominal freq is 250.0000 Hz, actual freq is 250.0093 Hz, precision is 2**19
reference time is BCF2162C.BDF0EE87 (09:33:16.741 CSTDST Wed Jun 14 2000)
clock offset is -91.9576 msec, root delay is 4.14 msec
root dispersion is 135.64 msec, peer dispersion is 43.67 msec
ntp_client#
```


里的 `stratum` 经过调整后比它主时钟的原始 `stratum` 值要大 1。这个例子里还配置了一些与时间相关的选项，包括夏令时的设置以及与 UTC 6 小时的偏离值。这些选项后面会再讨论。

17.2.2 NTP 静态客户模式的配置

配置 NTP 客户的另一种方式是将 NTP 客户端静态地映射到某个指定的时间服务器上。如果想要从某个特定主机接收 NTP 广播，应该静态指定 NTP 服务器。这种应用的一个例子是连接 Internet 使路由器与其上的原子钟进行时钟同步。

在上一个模型的基础上，现在要在同一台路由器上配置另一个 NTP 客户以连接 Internet 上的原子钟。例 17-4 中的新 NTP 配置环境如图 17-2 所示。在 TMC.EDU 或者是 128.249.1.1 有一个可以用作 NTP 服务器的原子钟。配置例子 17-4 是按照图 17-2 的拓扑建立的。

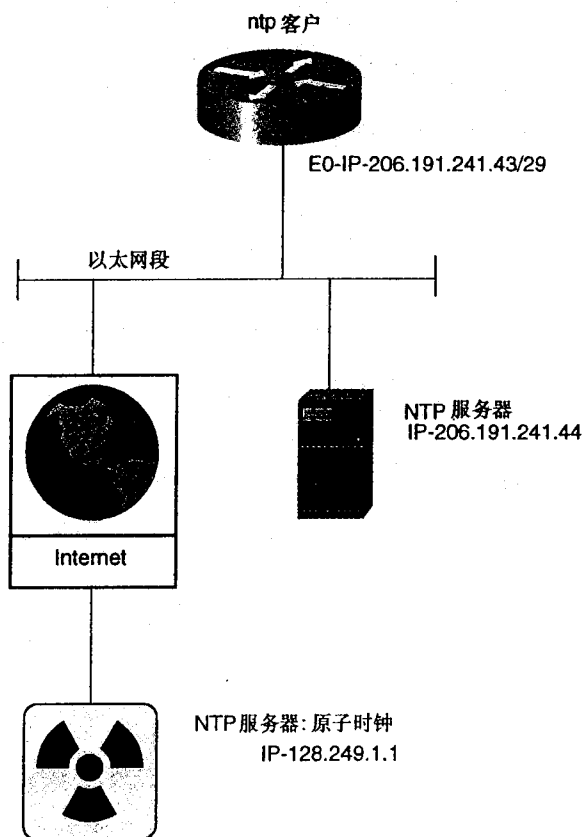


图 17-2 NTP 静态客户模式

例 17-4 路由器 ntp_client 的配置

```

hostname ntp_client
!
clock timezone CST -6
!
    
```

(待续)

```

<<<text omitted>>>
!
interface Ethernet0
ip address 206.191.241.43 255.255.255.248
no ip directed-broadcast
ntp broadcast client
!
<<<text omitted>>>
!
ntp clock-period 17179279      ←This is added by the router

```

用 `ntp server a.b.c.d` 命令可以配置路由器接收某个特定主机的 NTP 广播，这是一个全局命令，路由器可以同时配置多个 NTP 服务器。

现在可以利用 `show` 命令（例如 `show ntp assoc` 和 `show ntp status`）来验证时钟同步的情况。

例 17-5 显示了 NTP 缓慢收敛于新的 NTP 服务器的情况。RFC 1305 指出，时钟的同步需要很长的时间以及多次的比较，以保证时间的准确性。同步进行的时间与很多因素有关系。在时钟进行同步时，大家要作好多等一段时间的准备，但是，如果一个小时左右同步还没有完成，就得检查一下 NTP 的设计和配置。这里的试验和实例中，NTP 的时钟同步基本上都在 5 分钟之内完成。

例 17-5 show ntp assoc 和 show ntp status 命令示例

```

ntp_client#show ntp stat
Clock is synchronized, stratum 6, reference is 206.191.241.44
nominal freq is 250.0000 Hz, actual freq is 250.0096 Hz, precision is 2**19
reference time is BCF258FA.69DAF6F5 (14:18:18.413 CSTDST Wed Jun 14 2000)
clock offset is -16.8153 msec, root delay is 4.06 msec
root dispersion is 409.61 msec, peer dispersion is 392.78 msec
ntp_client#
ntp_client#show ntp ass

      address          ref clock      st when poll reach delay offset  disp
-128.249.1.1          0.0.0.0        16   -   64   0    0.0  0.00 16000 ←NTP
configured but no synced
* 206.191.241.44      128.194.103.14  5    46  8192  77    4.1 -16.82 392.8
* master (synced), # master (unsynced), + selected, - candidate, - configured
ntp_client#

ntp_client#show ntp ass

      address          ref clock      st when poll reach delay offset  disp
*-128.249.1.1         139.78.160.41   3    11  512 377 114.5 30.71 26.0 ←NTP Sync
 206.191.241.44      128.194.103.14  5    26  8192  77    4.1 15.23 20.0
* master (synced), # master (unsynced), + selected, - candidate, - configured
ntp_client#show ntp stat
Clock is synchronized, stratum 4, reference is 128.249.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0093 Hz, precision is 2**19
reference time is BCF26309.F694401F (15:01:13.963 CSTDST Wed Jun 14 2000)
clock offset is 30.7128 msec, root delay is 151.06 msec
root dispersion is 97.17 msec, peer dispersion is 26.05 msec
ntp_client#

```

17.2.3 NTP 主模式的配置

Cisco 路由器也可以利用 NTP 作为授权中心 NTP 服务器。把路由器配置为 NTP 服务器时要注意 stratum 级别，其范围应该是在 6 到 15 之间，默认情况下是 8。中高档的路由器，如 36xx, 47xx 和 7k 系列，能够提供更高的可靠性以及日历功能。NTP 服务器应该选用这类路由器。

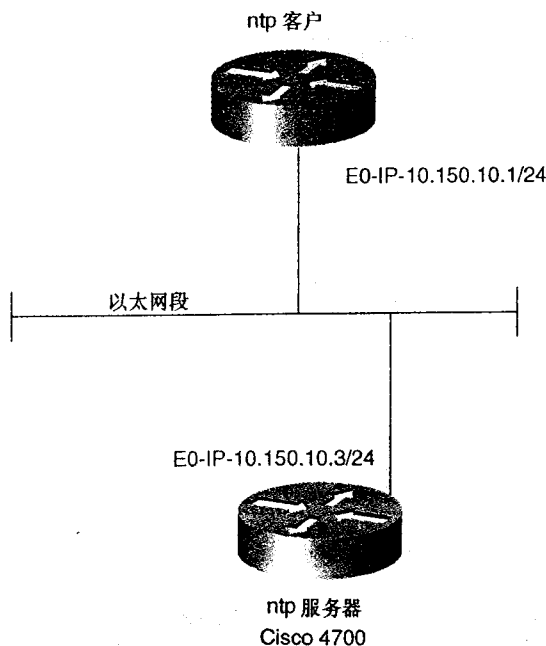
在全局配置模式下使用 **ntp master [stratum_number]** 命令可以把 Cisco 路由器配置为授权中心 NTP 服务器。图 17-3 是把 Cisco 4700 路由器配置为 NTP 主服务器，而 Cisco 2500 作为 NTP 客户端的例子。Cisco 4700 以其优于 2500 的高可靠性成为 NTP 服务器的首选。在服务器上需要执行 **ntp master** 命令，而客户端则可以配置成静态客户或者是广播客户。

例 17-6 则是 NTP 主服务器和 NTP 静态客户端的配置情况。

例 17-6 一台 NTP 主服务器和一台 NTP 静态客户端的配置

```
ntp_server#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ntp_server(config)#ntp master 7
ntp_server(config)#exit
ntp_server#

ntp_client#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ntp_client(config)#ntp server 10.150.10.3
ntp_client(config)#exit
ntp_client#
```



用 **show ntp status** 命令可以验证时钟是否已经同步完毕，而 **show ntp associations** 命令则能够提供本地路由器与之同步的时钟服务器的详细信息。例 17-7 是这些命令的输出情况。NTP 主服务器的地址是 10.150.10.3。

例 17-7 show ntp status 命令和 show ntp associations 命令示例

```
ntp_client#show ntp status
clock is synchronized, stratum 8, reference is 10.150.10.3
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**19
reference time is BD13BB90.AE4D80D0 (03:03:44.680 UTC Mon Jul 10 2000)
clock offset is 1.2757 msec, root delay is 3.78 msec
root dispersion is 1.97 msec, peer dispersion is 0.67 msec
ntp_client#
ntp_client#show ntp associations

      address          ref clock      st when poll reach delay offset  disp
* 10.150.10.3         127.127.7.1      7   60   64  377    3.0   1.28   0.7
* master (syncd), # master (unsyncd), + selected, - candidate, - configured
ntp_client#
```

17.2.4 NTP 对等体关系的配置

NTP 的对等体关系和静态对等体的情况非常相似。NTP 的对等体关系指该系统可以和其他系统进行同步，也可以是其他系统来和本系统进行同步。这种类型的 NTP 配置可以用在标

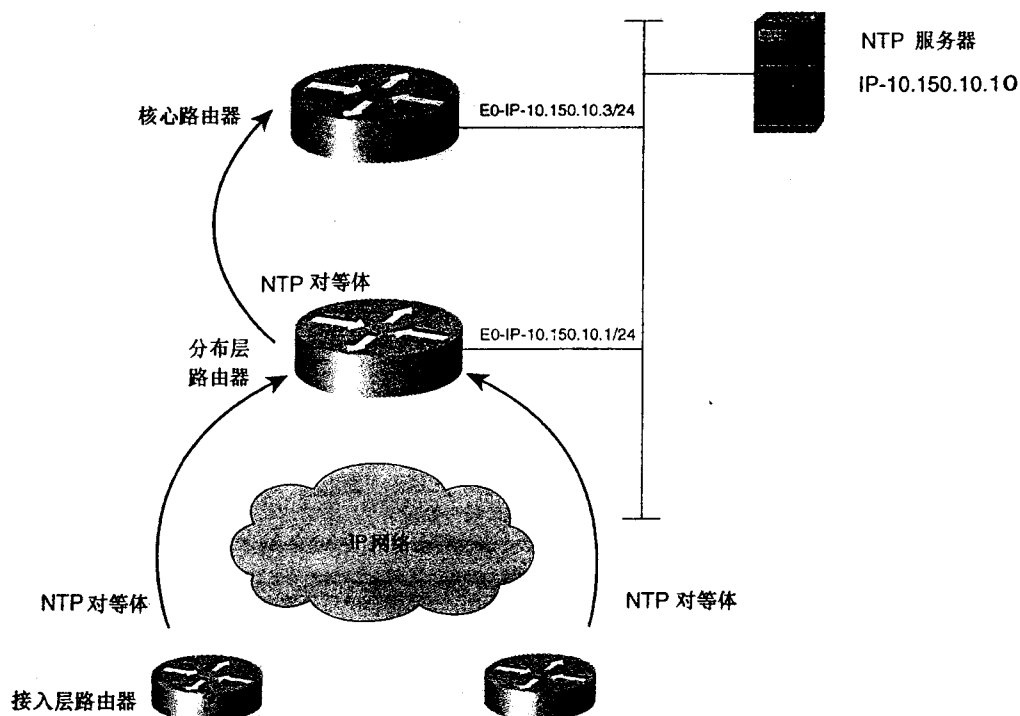


图 17-4 NTP 的对等体关系

准的 Cisco 三层分布网络设计中 (核心层, 分布层和访问层)。核心层是 NTP 主服务器, 可以是路由器或真正的 NTP 服务器。分布层路由器与核心层路由器之间有单一的对等体关系。而访问层路由器又和分布层路由器之间有对等体关系。这样就防止了所有路由器到同一主机的 NTP 广播, 无论这种广播是多么微不足道, 而且这种设计还提供了整个网络完整的时钟同步功能。如果某对等体失去了同步, 按照 NTP 的规则, 就不会与任何其他外部时钟源进行同步。因此, 要确定好中央对等体的位置, 因为这里发生的物理连接问题有可能会造成时钟无法同步。

图 17-4 中, 访问层路由器与分布层路由器的以太网段之间有单一的 NTP 对等体关系, 而分布层路由器又和核心层路由器之间有单一的 NTP 对等体关系。核心层路由器可以产生时钟源或者与一个更为稳定可靠的外部时钟源进行同步。配置 NTP 对等体, 可以使用 `ntp peer ip_address` 命令, `ip_address` 是与之同步的时钟源地址。这里没有必要在链路两端都使用 `ntp peer` 命令。图 17-4 中的访问层路由器需要一条 `ntp peer` 命令, 分布层和核心层路由器也一样, 每条命令都指向获得时钟源的地址。例 17-8 是在图 17-4 所示的网络中实现 NTP 对等体所需的命令应用示例。

例 17-8 图 17-4 所示网络的 NTP 对等体配置示例

```
Access routers:
ntp peer 10.150.10.1

Distribution router:
ntp peer 10.150.10.3

Core router:
ntp server 10.150.10.10    or
ntp master 6
```

17.2.5 NTP 认证以及与时间相关的选项的配置

NTP 也能为需要安全时钟源的应用提供 NTP 数据包的消息摘要 (MD5) 认证。对于需要按分钟或秒进行收费的应用来说, 在整个网络中保持一个安全可靠的时钟非常关键。MD5 认证的步骤如下:

- 第 1 步 用全局命令 `ntp authenticate` 允许进行 NTP 认证。
- 第 2 步 定义认证用的密钥, 设置一个 MD5 密码和一个认证密钥, 所用的全局命令是 `ntp authentication-key key_number md5 md5_password`。
- 第 3 步 应用一个可信密钥。使用第 2 步中的 `key_number`, 定义一个可信密钥, 该密钥是路由器之间进行 NTP 认证时用的, 这可以用 `ntp trusted-key key_number` 命令来完成。

图 17-5 是两台要进行 NTP 认证的路由器的例子。路由器 `ntp_t_server` 是 NTP 主路由器, 而 `ntp_t_client` 则是 NTP 客户路由器。

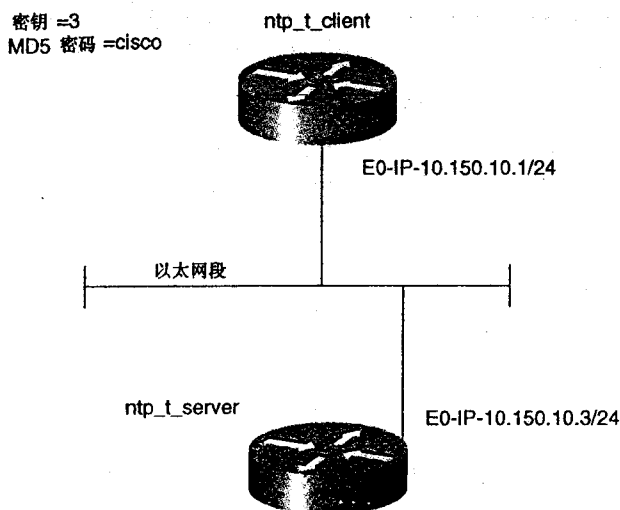


图 17-5 NTP 主路由器和客户路由其进行 MD5 认证

两台路由器的 **ntp authentication** 命令应该完全一样。每台路由器上都应首先启动认证功能，然后定义认证密钥和可信密钥。例 17-9 是配置 NTP 主路由器所需的命令示例，例 17-10 则是客户端路由器的配置示例。

例 17-9 在主路由器上配置 NTP 的认证

```
ntp_t_server(config)#ntp master 6
ntp_t_server (config)#ntp authenticate
ntp_t_server (config)#ntp authentication-key 3 md5 cisco
ntp_t_server (config)#ntp trusted-key 3
```

例 17-10 在客户端路由器上配置 NTP 的认证

```
ntp_t_client(config)#ntp server 10.150.10.3
ntp_t_client(config)#ntp authenticate
ntp_t_client(config)#ntp authentication-key 3 md5 cisco
ntp_t_client(config)#ntp trusted-key 3
```

利用上面提到的 **show** 命令 (**show ntp status** 和 **show ntp associations**) 可以验证 NTP 是否已经同步，如例 17-11 所示。

例 17-11 show ntp stat 和 show ntp assoc 命令示例

```
ntp_t_client#show ntp stat
Clock is synchronized, stratum 7, reference is 10.150.10.3
nominal freq is 250.0000 Hz, actual freq is 249.9990 Hz, precision is 2**19
reference time is BD15D137.C52F08AF (17:00:39.770 UTC Tue Jul 11 2000)
clock offset is 1.4114 msec, root delay is 3.77 msec
root dispersion is 1.46 msec, peer dispersion is 0.03 msec

ntp_t_client#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
---------	-----------	----	------	------	-------	-------	--------	------

```
*~10.150.10.3      127.127.7.1      6      37      64      377      3.8      1.41      0.0
* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured
ntp_t_client#
```

时钟与时区的配置

在同一网络中保持共同精确与稳定的时钟非常重要，这也包括了时区和夏令时的设置。默认的时区是 UTC 或 GMT。如果使用 Cisco 3600、7000 以及其他高端路由器，由于它们都具有日历功能，NTP 也可以周期性的对日历进行更新。

配置与时区相关的参数值，可以在全局配置模式下使用下列命令：

- **clock timezone** *timezone_name* [*hours_plus_or_minus_from_UTC*] [*minutes_offset_from_UTC*]
- **clock summertime** *summer_timezone_name* [*recurringdate*]
- **ntp update-calendar**

在配置时区时，可以根据所属是太平洋时区还是中央时区分别输入时区的名称，如 PAC 或 CST 等，也可以自己虚构一个名称。此外，还可以设置本地时间与 UTC 标准时间相比所要做的补偿小时数 (+23 到 -23) 和分钟数。用 **clock summertime** 命令配置夏令时，可以输入一个时区名，这个名称在夏令时模式运行时显示，也可以设置任意的夏令时的补偿值或者是当地政府采用作息时间的补偿值。夏令时的默认规则是 4 月的第一个周日的 2:00 a.m.，路由器或交换机把时钟前拨一个小时，而 10 月的最后一个周日的 2:00 a.m. 则把时钟往后拨一个小时。

现在在图 17-5 的路由器上执行这些命令。将 **ntp_t_server** 的时区设置为美国中央时区 (CST) 并且启动夏令时。CST 与 GMT 之间的偏移值是 -6 小时，据此在路由器上进行相应的调整。例 17-12 是设置图 17-5 中的主路由器时钟为 CST 的例子。

例 17-12 时区和夏令时的设置

```
ntp_t_server (config)#clock timezone CST -6
ntp_t_server (config)#clock summer-time CDT recurring
```

在路由器 **ntp_t_master** 上执行 **show clock** 命令可以得到例 17-13 的结果。

例 17-13 show clock 命令显示夏令时正在运行

```
ntp_t_server#show clock
13:44:49.063 CDT Tue Jul 11 2000
ntp_t_server#
```

请注意上面的 CDT 是时区名，表明正在运行夏令时。

技巧 NTP 在更新时并不传输时区信息。因此，在正确配置时区时，必须对网络中的所有路由器使用 **clock timezone** 命令。如果网络规模很大，或者是范围很广，建议大家把时钟设为默认的 UTC。如果网络规模较小，路由器可以使用核心层路由器的时区。没有公共的时区，问题和事件的处理将会变得非常困难。

注释 出现在路由器的配置清单中的 **ntp clock-period** 是路由器在启动 NTP 之后自动添加的，用来在路由器重新启动时起动 NTP 的频率补偿功能。

17.3 简单网络时间协议（SNTP）的配置

小型路由器（如 Cisco 100x, 80x 以及其他低端路由器）不支持 NTP 协议，这时可以在这类路由器上配置 SNTP。然而，SNTP 不具有 NTP 的一些增强功能，不能用作 NTP 服务器，也没有认证和统计机制。和 NTP 类似，SNTP 可以以两种形式进行配置：

- 路由器配置成被动的监听线路上 SNTP 广播。
- SNTP 静态地映射到某个指定服务器。

如果配置了两种方式，路由器从双方都可以接收广播信息，但是路由器更倾向于使用来自静态指定的服务器的时钟信息，这里是假设二者 stratum 值都一样的情况。

可以使用下面这些全局命令在小型路由器系统平台上配置 SNTP：

- **sntp server server_IP_address**——这条命令静态地把一台 SNTP 服务器映射到路由器上用于 SNTP 更新，和 NTP 服务器命令很相似。
- **sntp broadcast client**——该命令使路由器被动地监听路由器接口上的 SNTP 广播。
- **show sntp**——这条命令用于显示 SNTP 状态。

本节的例子都采用名为 skynet_2 的 Cisco 804 路由器，在其 LAN 接口上连有一台 NTP 服务器，如图 17-6 所示。

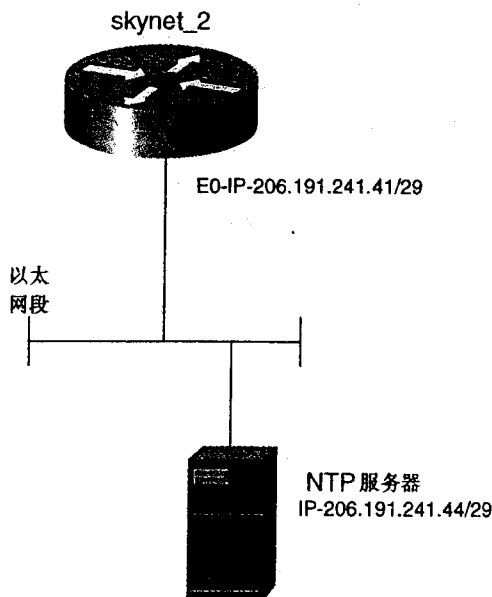


图 17-6 SNTP 配置的拓扑结构图

要将这台路由器配置来接收某个未知时钟源的更新信息，只要加上一条配置命令 **sntp broadcast client** 即可。当然，还要将时区设置为 CST 以启动夏令时。例 17.14 是配置示例。

例 17-14 在一台 Cisco 804 路由器上配置 SNTP

```

skynet_2(config)#ntp broadcast client
skynet_2(config)#clock timezone CST -6
skynet_2(config)#clock summer-time CDT recurring
skynet_2(config)#exit

skynet_2#show ntp
SNTP server      Stratum    Version    Last Receive
206.191.241.44   14         3          00:00:39   Synced Bcast

Broadcast client mode is enabled.

skynet_2#show clock
14:12:45.116 CDT Tue Jul 11 2000

```

例子中的 SNTP 是和 stratum 为 14 的服务器 206.191.241.44 进行了同步。命令 **show clock** 的结果显示时钟采用的是夏令时。

17.4 NTP 和 SNTP 的 “Big show” 和 “Big D” 命令

已经介绍和应用了 3 个主要的 NTP 和 SNTP 的 **show** 命令，这些命令包括：

- **show ntp status**——这条命令可以提供 NTP 同步的详细信息以及 NTP 的 3 个要素：延时、偏移量和差量。
- **show sntp**——这个命令可以用来确认 SNTP 是否已经配置以及是否找到了有效的时钟源，如果 SNTP 已经同步完毕，那么可以显示服务器的 IP 地址及其 stratum 值。
- **show ntp associations [detailed]**——这条命令可以显示所有静态配置的 NTP 对等体信息以及时钟的选择和对等体是否同步的信息。

例 17-15 给出 **show ntp status** 命令的执行结果。结果说明这里的时钟已经同步完毕，具有一个有效的 stratum 值（通常是小于 16）。此外，输出信息还包括参考时钟、时钟偏移量、根延时、根与对等体差量，在时钟同步之后，这些值应该很小，通常是毫秒（ms）量级。

例 17-15 show ntp status 命令示例

```

ntp_t_client#show ntp status
Clock is synchronized, stratum 7, reference is 10.150.10.3
nominal freq is 250.0000 Hz, actual freq is 249.9984 Hz, precision is 2**19
reference time is BD1604F7.3618FF07 (20:41:27.211 UTC Tue Jul 11 2000) ←Correct
time!
clock offset is 0.3951 msec, root delay is 3.78 msec
root dispersion is 0.44 msec, peer dispersion is 0.03 msec
ntp_t_client#

```

例 17-16 列出的也是 **show ntp status** 命令的执行结果，这里时钟还没有同步。可以看到 stratum 没有设置，而参考时钟也不存在。

例 17-16 show ntp status 命令在时钟没有同步时的输出示例

```
timex#show ntp stat
Clock is unsynchronized, stratum 16, no reference clock
nominal freq is 250.0000 Hz, actual freq is 250.0003 Hz, precision is 2**19
reference time is BD15D99C.4AC66EE0 (17:36:28.292 UTC Tue Jul 11 2000)
clock offset is -0.1224 msec, root delay is 31.14 msec
root dispersion is 1.45 msec, peer dispersion is 0.14 msec
timex#
```

例 17-17 列出 **show sntp** 命令的执行结果。这里的信息包括 SNTP 服务器的 IP 地址、有效的 stratum 值以及时钟有否同步的确认。如果没有找到有效的 SNTP 广播，这条命令通常只是显示 SNTP 客户模式已经启动的信息。

例 17-17 show sntp 命令示例

```
skynet_2#show sntp
SNTP server      Stratum   Version   Last Receive
206.191.241.44   14          3        00:00:44   Synced Bcast

Broadcast client mode is enabled.
```

例 17-18 给出 **show ntp associations** 命令的执行情况。这里的关键信息是*符号，表明主路由器时钟已经完成同步，*后面是主路由器的 IP 地址以及与之同步的时钟的 stratum 值。这个例子的底部列出该命令的扩展形式，包括了延时、偏移量和差量的信息。输出的第 1 行是主 NTP 的简要信息。

例 17-18 show ntp associations 命令示例

```
timex#show ntp associations

address          ref clock      st when poll reach delay offset disp
*10.150.10.3     10.150.10.3    7 38 64 377 6.1 -0.51 0.1
* master (synced), # master (unsynced), + selected, - candidate, - configured
timex#
timex#show ntp associations detail
10.150.10.3 configured, our master, sane, valid, stratum 7
ref ID 10.150.10.3, time BD160B37.3911241E (21:08:07.222 UTC Tue Jul 11 2000)
our mode active, peer mode passive, our poll intvl 64, peer poll intvl 64
root delay 3.78 msec, root disp 0.41, reach 377, sync dist 5.432
delay 6.07 msec, offset -0.5087 msec, dispersion 0.09
precision 2**19, version 3
org time BD160B48.A2712FA3 (21:08:24.634 UTC Tue Jul 11 2000)
rcv time BD160B48.A3598FCA (21:08:24.638 UTC Tue Jul 11 2000)
xmt time BD160B48.A19B718B (21:08:24.631 UTC Tue Jul 11 2000)
filtdelay =      6.07  6.04  6.03  6.13  6.03  6.09  6.01  6.06
filtoffset =    -0.51 -0.41 -0.40 -0.43 -0.48 -0.43 -0.41 -0.39
filtererror =     0.02  0.99  1.97  2.94  3.92  4.90  5.87  6.85
timex#
```

Cisco 也提供了很多 NTP 用的 **debug** 命令，最有用的可能就是 **debug ntp select** 命令，它能显示 NTP 对等体的状态以及交换的数据包信息。其他 **debug** 命令大部分都只有在事件触发时才能显示相应的信息，像 **debug ntp events** 命令和 **debug ntp sync** 命令。这些显示信息只

有状态发生改变时才会出现，而且这些也不包括任何内部处理和操作的信息。

例 17-19 是 `debug ntp select` 命令的输出结果，它表明输出的数据包是传向 NTP 服务器 10.150.10.3，即配置的时钟源。此外，结果中还有在残存时钟上测得的偏移量，表明时钟已经被校正过。

例 17-19 debug ntp select 命令示例

```
03:42:37: NTP: nlist 1, allow 0, found 0, low -0.001984, high 0.002411
03:42:37: NTP: candidate 10.150.10.3 cdist 96.002197 error 0.000305
03:42:37: NTP: survivor 10.150.10.3 offset 0.000226, cdist 96.00220

03:43:23: NTP: nlist 1, allow 0, found 0, low -0.001785, high 0.002151
03:43:23: NTP: candidate 10.150.10.3 cdist 96.001968 error 0.000076
03:43:23: NTP: survivor 10.150.10.3 offset 0.000185, cdist 96.00197

03:43:41: NTP: nlist 1, allow 0, found 0, low -0.002045, high 0.002411
03:43:41: NTP: candidate 10.150.10.3 cdist 96.002228 error 0.000336
03:43:41: NTP: survivor 10.150.10.3 offset 0.000185, cdist 96.00223
```

17.5 实验 34：配置 NTP 服务器、客户端和认证——

第 1 部分

17.5.1 实验说明

NTP 是网络管理的关键部分。如果没有公共的时钟，对任何网络异常问题的跟踪、纠正都非常困难。配置 NTP 后，所有的路由器会同步到同一个时钟，从而简化了网络的故障排除工作。

17.5.2 实验内容

我们现在扮演的角色是 management.com 网络的工程师，负责的任务是将网络中所有的路由器与公共时钟源同步。在将 NTP 应用到整个网络之前，必须检查是否满足下面的要求：

- 如果无法访问 NTP 服务器，必须配置路由器让网络能访问 NTP 服务。
- 所有的路由器都处在美国-太平洋时区，与格林尼治标准时间 (GMT) 或 UTC 之间的偏移量是 8 个小时。确定整个网络的时区，并将其标记为 PAC。
- 对 NTP 进行配置，使得 NTP 只传输安全的更新信息。
- 远程路由器 client_router 的 stratum 值应该是 6，它的时钟要同步于 mngt_router 的时钟。
- (可选) 设置所有路由器在夏令时状态，并以 S-PAC 作为标记。

17.5.3 实验目的

- 按照图 17-7 对网络进行配置，路由选择协议采用 OSPF，客户端路由器放置在存根

- 将管理路由器配置为 NTP 服务器，而客户路由器作为 NTP 客户端。
- 对 NTP 实施 MD5 认证，密码是 cns，密钥是 2。
- 设置适当的 stratum 级别。

17.5.4 所需设备

- 2 台 Cisco 路由器通过 V.35 背对背线缆或类似方式连接。路由器必须支持 NTP——Cisco 2500 以及更高级的路由器。建立该网络的模型时，使用 Cisco 4700 和 Cisco 2500 系列的路由器。
- 通过集线器或交换机创建 2 个 LAN 网段。

17.5.5 物理设计与实验准备

- 按照图 17-7 将集线器以及串行线缆与路由器互连，WAN 协议采用 HDLC。
- 按照图 17-7 将 2 台以太网集线器与路由器相连形成 2 个 LAN 网段。

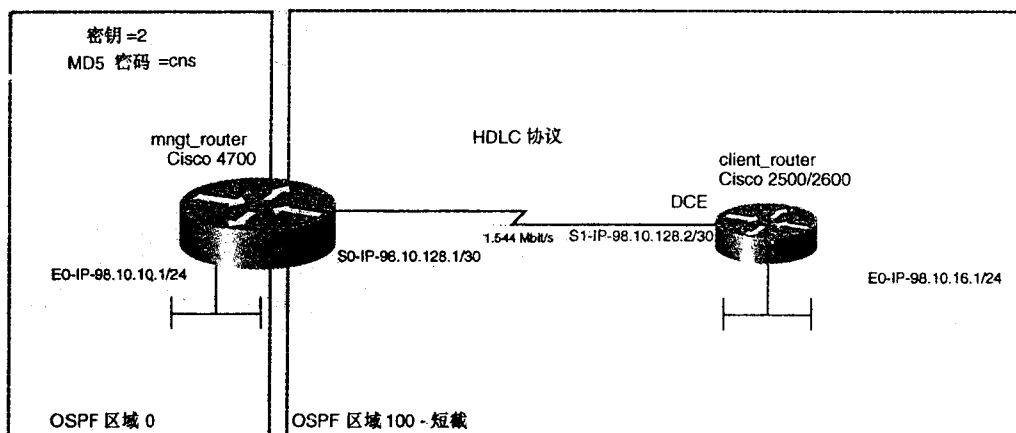


图 17-7 Management.Com 网络：NTP 概念的应用

17.6 实验 34：配置 NTP 服务器、客户端和认证——

第 2 部分

17.6.1 实验步骤

完成了 LAN 和 WAN 网段的物理连接之后，建立所有路由器之间的 IP 连接。在进行 NTP 的配置之前，一定要从每台路由器的以太端口通过源地址 ping 命令来验证端到端连接。

首先对路由器 mngt_router 进行配置，在以太 E0 端口和串行端口，这是串行链路的 DTE 端，因而无需 clock rate 命令进行配置。WAN 协议采用 HDLC，所以也没有必要在这个链路

上配置封装的类型。

路由器 `client_router` 需要在 E0 端口和 S1 端口上对 IP 地址进行配置。这是链路的 DCE 端，因而需要用 `clock rate` 命令设置通信速率。

配置 OSPF 前，一定要保证所有的路由器可以 ping 通相互之间的串行接口。这里的 WAN 是作为本地网络，远程终端应该是可以被访问的。

在路由器 `mngt_router` 上配置 OSPF 的时候，需要加上两条 `network` 命令加上 `area` 子命令。例 17-20 是路由器 `mngt_router` 到现在为止的相关 IP 配置。

例 17-20 路由器 `mngt_router` 的相关 IP 配置

```
! hostname mngt_router
!
interface Ethernet0
 ip address 98.10.10.1 255.255.255.0
 media-type 10BaseT
!
interface Serial0
 ip address 98.10.128.1 255.255.255.252
 no ip mroute-cache
!
router ospf 100
 network 98.10.10.1 0.0.0.0 area 0
 network 98.10.128.1 0.0.0.0 area 100
 area 100 stub
```

路由器 `client_router` 上的 OSPF 配置也类似。该路由器全部位于存根区域，可以通过在 `network` 声明中使用反向掩码对配置过程加以简化。另外，Area 100 还需要用 `area stub` 命令进行配置。例 17-21 是路由器 `client_router` 的相关 IP 配置示例。

例 17-21 路由器 `client_router` 的相关 IP 配置

```
hostname client_router
!
interface Ethernet0
 ip address 98.10.16.1 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 ip address 98.10.128.2 255.255.255.252
 clockrate 2000000
!
router ospf 100
 network 98.10.0.0 0.0.255.255 area 100
 area 100 stub
!
```

现在来对 NTP 部分进行配置，需要配置的包括：

- NTP 主时钟源是 stratum 级别为 5 的 `mngt_router`。

- NTP 的客户时钟源在 `client_router` 上，因此同步之后，`stratum` 级别为 6。
- 时区是 PAC，与 UTC 的偏移量为 -8 (小时)。
- 作为可选部分，在网络中实施夏令时，名称是 S-PAC。

路由器 `mngt_router` 上的 NTP 配置需要加上 `ntp master 5` 命令，这里的 5 会在与路由器 `client_router` 同步时将客户端的 `stratum` 级别设为 6。认证的配置则需要下面这 3 个步骤：

第 1 步 允许认证。

第 2 步 定义认证密钥。

第 3 步 定义置信密钥。

例 17-22 是路由器 `master_router` 上的命令示例。路由器 `client_router` 上也需要配置相同的 `authentication` 命令。

例 17-22 路由器 `mngt_router` 上的 NTP 和 MD5 认证

```
mngt_router(config)#ntp master 5
mngt_router(config)#ntp authenticate
mngt_router(config)#ntp authentication-key 3 md5 cns
mngt_router(config)#ntp trusted-key 3
mngt_router(config)#exit
```

客户路由器的配置和主路由器类似，区别在于用 `ntp server 98.10.128.1` 命令取代了 `ntp master` 命令。认证部分的配置则完全一样。

检查 NTP 的状态可以使用 `show ntp status` 和 `show ntp assoc` 命令。在路由器 `client_router` 上执行这些命令的结果如例 17-23 所示。

例 17-23 `show ntp status` 和 `show ntp assoc` 命令的执行结果

```
client_router#show ntp status
Clock is synchronized, stratum 6, reference is 98.10.128.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**19
reference time is BD19C766.31F0D0F5 (17:07:50.195 UTC Fri Jul 14 2000)
clock offset is -0.3234 msec, root delay is 4.46 msec
root dispersion is 0.52 msec, peer dispersion is 0.15 msec
client_router#
client_router#show ntp assoc
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*-98.10.128.1	127.127.7.1	5	59	64	377	4.5	-0.32	0.2

* master (syncd), # master (unsyncd), + selected, - candidate, - configured
client_router#

例 17-23 中确认时钟已经同步完毕，`stratum` 的值设为 6。同时还请注意时钟应该有一个合适的参考时间，这应该和所配置一致。

最后要做的是根据实验中指定的时区和名称设置时间与时钟，通过在客户端与主机路由器上执行 `clock timezone PAC -8` 命令来实现。现在查看当前时间时，就会发现 UTC 已经为 PAC 所代替，并以 8 小时的偏移量进行了弥补。

S-PAC recurring 命令来完成的。在夏令时中的路由器命令执行结果如例 17-24 所示。

例 17-24 夏令时情况下 show ntp status 命令的执行情况

```
client_router#show ntp status
Clock is synchronized, stratum 6, reference is 98.10.128.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**19
reference time is BD19CA66.316D2935 (10:20:38.193 S-PAC Fri Jul 14 2000)
clock offset is -0.5572 msec, root delay is 4.49 msec
root dispersion is 0.66 msec, peer dispersion is 0.06 msec
client_router#
```

例 17-25 则是两台路由器的配置情况。

例 17-25 路由器 mngt_router 和 client_router 的配置

```
hostname client_router
!
enable password cisco
!
clock timezone PAC -8
clock summer-time S-PAC recurring
!
interface Ethernet0
 ip address 98.10.16.1 255.255.255.0
!
<<<text omitted>>>
!
interface Serial1
 ip address 98.10.128.2 255.255.255.252
 clockrate 2000000
!
<<<text omitted>>>
!
router ospf 100
 network 98.10.0.0 0.0.255.255 area 100
 area 100 stub
!
ip classless
!
<<<text omitted>>>
!
ntp authentication-key 2 md5 070C2F5F 7
ntp authenticate
ntp trusted-key 2
ntp clock-period 17179866
ntp server 98.10.128.1
```

```
hostname mngt_router
!
enable password cisco
!
clock timezone PAC -8
clock summer-time S-PAC recurring
!
!
interface Ethernet0
 ip address 98.10.10.1 255.255.255.0
 media-type 10BaseT
!
```

```
<<<text omitted>>>
|
interface Serial0
  ip address 98.10.128.1 255.255.255.252
  no ip mroute-cache
|
<<<text omitted>>>
|
router ospf 100
  network 98.10.10.1 0.0.0.0 area 0
  network 98.10.128.1 0.0.0.0 area 100
  area 100 stub
|
ip classless
|
<<<text omitted>>>
|
ntp authentication-key 2 md5 104D070A 7
ntp authenticate
ntp trusted-key 2
ntp master 5
```

17.7 实验 35：配置 NTP 服务器、客户端和对等体

——第 1 部分

17.7.1 实验说明

如前所述，NTP 是网络管理以及需要精确时钟源的网络应用关键组成部分。Cisco 路由器可以同步于外部时钟源，也可以让其他路由器与之同步。

17.7.2 实验内容

假设 Independent Ticket, Inc. 开发了一个票务销售系统，该系统需要精确的时钟源，因为必须保证票务的销售开始、结束于正确的时间。一些乐队采用了这个系统，它们希望自己来协调、管理自己的表演门票的销售，而不是依靠现有的票务垄断者来做。他们拥有一个小型的中央站点为其提供公共的表演、集会信息，同时也为它们提供了一个安全的 NTP 服务器，所有的乐队都通过 56-kbps 链路与这个中央站点连接。乐队 Metallica 和 Pearl Jam 最近也与这个站点签了合同，准备接收站点提供的 NTP 服务。我们的现在的任务就是按照下面的要求对路由器 metallica 和 pearl_jam 进行配置：

- 网络中有一个核心路由器 ticket_central，通过两条 56K 的 HDLC 链路连接路由器 metallica 和 pearl_jam。
- 在路由器 ticket_central 的主干以太网段中有一个 NTP 服务器，NTP 服务器的 IP 地址是 206.191.241.45，对路由器 ticket_central 进行配置，使其向这台服务器同步。
- 对路由器 metallica 和 pearl_jam 进行配置，使其与路由器 ticket_central 之间建立起

- 所有的路由器都处在美国-太平洋时区，与 GMT 或 UTC 之间有 8 小时的偏移量。设计的时候要确定网络的时区，同时以 PAC 作为标记。
- （可选）禁止任何 NTP 广播信息传到路由器 *metallica* 和 *pearl_jam* 的以太网段。

17.7.3 实验目的

- 按照图 17-8 对网络进行配置，路由选择协议采用，自治系统 ID 是 2001。
- 将路由器 *ticket_central* 配置为 NTP 主路由器，时钟源来自 NTP 服务器 206.191.241.45。
- 对路由器 *metallica* 和 *pearl_jam* 进行配置，使其与路由器 *ticket_central* 之间建立起对等体。
- 远程站点的 stratum 级别为 5。

17.7.4 所需设备

- 3 台 Cisco 路由器，其中 2 台通过 V.35 背对背线缆与 *ticket_central* 路由器的串行端口相连。
- 通过集线器或交换机形成的 3 个 LAN 网段。
- 一台 NTP 服务器位于路由器 *ticket_central* 的以太网段。到共享软件站点 *download.com* 下载一个 NTP 服务器软件，在这个网站能够找到实验所需的 NTP 服务器软件 TARDIS 2000。任何服务器（或其他路由器）都可以完成本实验中的服务器功能。

17.7.5 物理设计与实验准备

- 按照图 17-8 将集线器以及串行线缆与路由器相连，WAN 协议采用 HDLC。
- 按照图 17-8 将 3 台以太网集线器与路由器相连以形成 3 个 LAN 网段。
- 在网络中配置 EIGRP 作为路由选择协议，自治系统 ID 采用 2001。

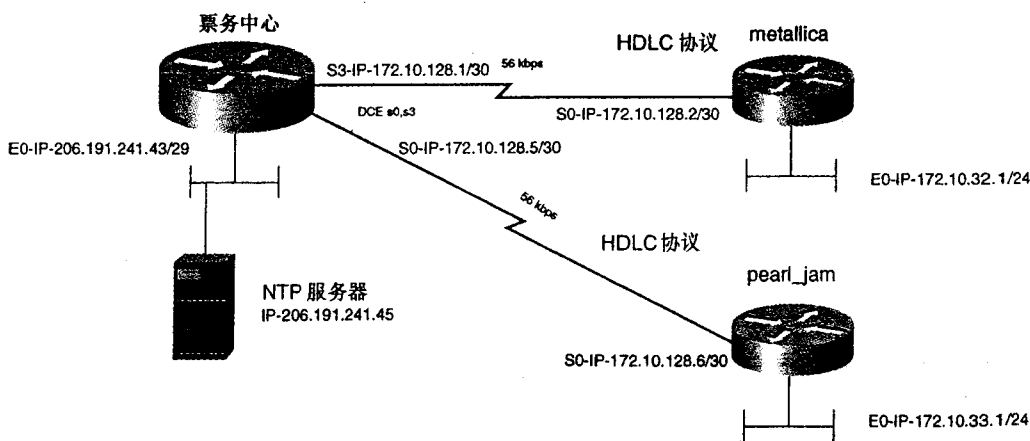


图 17-8 配置 NTP 对等体关联关系

- 在路由器 ticket_central 的以太网段上添加并配置 NTP 服务器，启动 NTP 广播，在服务器软件中设置 stratum 级别为 3。服务器的 IP 地址是 206.191.241.45。

17.8 实验 35：配置 NTP 服务器、客户端和对等体

——第 2 部分

17.8.1 实验步骤

LAN 和 WAN 网段的物理连接完成之后，在所有的路由器之间建立起 IP 连接。在开始 NTP 的配置之前，一定要记得在每台路由器以太接口上用源 **ping** 命令验证网络的端对端连接的情况。

以上配置全都完成之后，对路由器 ticket_central 进行操作，为 E0 端口和 2 个串行端口分配 IP 地址。这是串行链路的 DCE 端，因而需要用 **clock rate** 命令在其 S0 和 S3 接口上设置通信速率。由于 WAN 协议是 HDLC，因而无需在该链路上配置封装类型。

在路由器 metallica 和 pearl_jam 上，需要为其 E0 和 S0 端口分配 IP 地址。配置完 IP 地址，链路进入工作状态后，可以 **ping** 通每台路由器的串行端口。

接口进入工作状态之后，可以在所有的路由器上配置 EIGRP 的路由选择协议，自治系统 ID 2001。例 17-26 是实验中所有路由器的相关 IP 配置的情况。

例 17-26 实验中路由器的相关 IP 配置情况

```
hostname ticket_central
!
interface Ethernet0
 ip address 206.191.241.43 255.255.255.248
!
interface Serial0
 ip address 172.10.128.1 255.255.255.252
 no fair-queue
 clockrate 56000
!
<<<text omitted>>>
!
interface Serial3
 ip address 172.10.128.5 255.255.255.252
 clockrate 56000
!
<<<text omitted>>>
!
router eigrp 2001
 network 206.191.241.0
 network 172.10.0.0

hostname pearl_jam
!
interface Ethernet0
 ip address 172.10.33.1 255.255.255.0
```

```

!
interface Serial0
 ip address 172.10.128.6 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.10.0.0

```

```

hostname metallica
!
interface Ethernet0
 ip address 172.10.32.1 255.255.255.0
!
interface Serial0
 ip address 172.10.128.2 255.255.255.252
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.10.0.0
!

```

在对路由器 ticket_central 的 NTP 进行配置的时候，该路由器必须既是客户端又是主路由器。首先，在路由器上用 **ntp server 206.191.241.45** 命令来通过 E0 端口指向 NTP 服务器，而 NTP 主路由器的设置是通过 **ntp master** 命令来实现的。因为我们希望保持 NTP 服务器广播信息中包含的 stratum 级别，所以这里没有必要加上 stratum 数值。最后将时区设置为 PAC，与 GMT 或 UTC 之间有 8 小时的偏移量，这通过 **clock timezone PAC -8** 命令的配置来实现。例 17-27 是路由器 ticket_central 上的 NTP 配置情况。

例 17-27 路由器 ticket_central 的 NTP 配置

```

ticket_central(config)#clock timezone PAC -8
ticket_central(config)#ntp server 206.191.241.45
ticket_central(config)#ntp master

```

现在用 **show ntp status** 命令检查 NTP 的状态，时钟应该已同步，同步时钟的参考时钟应该指向 206.191.241.45 上。

最后配置路由器 metallica 和 pearl_jam 的 NTP 对等体关系，这通过分别在两台路由器上执行 **ntp peer 172.10.128.1** 和 **ntp peer 172.10.128.5** 命令来完成。同样也用 **clock timezone PAC -8** 命令设置时区为 PAC。

在远程路由器上利用 **show ntp status** 和 **show ntp associations** 命令验证 NTP 同步的情况。例 17-28 是路由器 metallica 上的命令执行情况。

例 17-28 验证 NTP 同步

```

metallica#show ntp status
Clock is synchronized, stratum 5, reference is 172.10.128.1

```

```
nominal freq is 250.0000 Hz, actual freq is 250.0010 Hz, precision is 2**19
reference time is BD1D1551.469209A4 (21:17:05.275 PAC Sun Jul 16 2000)
clock offset is -10.3164 msec, root delay is 34.47 msec
root dispersion is 70.31 msec, peer dispersion is 3.92 msec
metallica#
metallica#show ntp associations

      address      ref clock      st when poll reach delay offset  disp
* 172.10.128.1    206.191.241.45  4   39   64 377   30.3  10.32  3.9
* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured
metallica#
```

从中可见, `show ntp associations` 显示的参考时钟就是 NTP 服务器 206.191.241.45.stratum 为 5 说明该 stratum 来自 NTP 服务器, 因而在原来的基础上增加了 1。

实验的可选部分要求禁止 NTP 广播信息传送到远程路由器的以太网端。要禁止 NTP 广播信息进入在某个接口, 只需在该接口下用 `ntp disable` 命令进行配置即可。例 17-29 是路由器 `metallica` 和 `pearl_jam` 的配置情况。

例 17-29 路由器 `metallica` 和 `pearl_jam` 上禁止 NTP 广播的配置

```
hostname metallica
!
clock timezone PAC -8
!
interface Ethernet0
ip address 172.10.32.1 255.255.255.0

    ntp disable
!
interface Serial0
ip address 172.10.128.2 255.255.255.252
no ip directed-broadcast
no ip mroute-cache
no fair-queue
!
<<<text omitted>>>
!
router eigrp 2001
network 172.10.0.0
!
<<<text omitted>>>
!
ntp clock-period 17179749
ntp peer 172.10.128.1

-----

hostname pearl_jam
!
!
clock timezone PAC -8
!
interface Ethernet0
ip address 172.10.33.1 255.255.255.0
ntp disable
!
interface Serial0
ip address 172.10.128.6 255.255.255.252
```

```
no fair-queue
!
<<<text omitted>>>
!
router eigrp 2001
 network 172.10.0.0
!
<<<text omitted>>>
!
ntp clock-period 17179636
ntp peer 172.10.128.5
```

第 8 部分

CCIE 准备与自我评估

第 18 章 CCIE 实验考试: 考试准备与 CCIE 实验室练习

第 18 章

CCIE 实验考试： 考试准备与 CCIE 实验室练习

“成功没有捷径，不要费时去寻觅。”

这话出自一位我个人很崇拜的人物，已退休的 Colin Powell 将军之口。他的话非常适用于渴望成为 CCIE 的人，就像它适用于想当将军的士兵一样。CCIE 所需的知识不能完全包括在任何一个地方、任何一本书（包括本书）中。正如 Colin Powell 将军所说，“不要费时去寻觅”。本书完成时，2001 年 9 月，Cisco 宣布世界一共有 6678 名 CCIE。看看这个数字，再看看 Cisco 在市场上的占有情况，可以想到 CCIE 的行列不是那么容易进入。

为了反映当前的市场状况，CCIE 也处在不停的变化之中。1997 年，所有的考试标准化。1999 年，Cisco 在核心的“路由与交换”考试的基础上推出了其他专项考试，如 WAN 交换、SNA 或其他专业领域。这些专业认证考试近年来也发生了改变。2000 年，语音和 ATM 也加入到考试内容之中，而在当时的前几年，这类交换机还只在实验室中。2001 年，考试从原来的两天 16 小时改成了一天 8 个半小时。尽管发生了这么多变化，CCIE 实验考试有一点还是没有变化的。随着时间的推移，CCIE 在不断地增加难度。举个例子，我成为 CCIE 时，还不知道语音传输、令牌环交换或者 ATM 为何物。这正是没有任何东西能够包容所有 CCIE 内容的原因：考试本身在不停地变化着。

查找 CCIE 考试最新消息的最好地方是
www.cisco.com/go/ccie。

18.1 新的一天制 CCIE 试验考试

我有幸和 CCIE 考试的管理人 Lorne Braddock Sr. 谈过考试从两天制改为一天制的情况。和大多数的 CCIE 一样，我对这个变化也有一种下意识的反应。我意识到 Cisco 遇到的巨大负担，大量的参试者等着进入实验室，有的甚至一年前就已经预约。对很多 Cisco 合作伙伴来说，金牌合作者与银牌合作者之间的收入差别上百万。Cisco 与其客户遇到的另一个难题是怎样安排两天考试的时间表，因为参试者大都要很早就从家里出来。这不仅为 Cisco 的客户带来了时间、成本和金钱上的负担，而且也使得很多公共实验室处于窘境。

改成 8 个半小时的一天制考试可以解决很多问题，但是问题是——削减哪一部分的考试时间呢？考试委员会决定改进第 1 层（物理层）的书面考试。实验考试的物理层内容就技术而言是很小的一个部分，但是却非常费时间。另一个可以削减的部分是故障排除部分。事实上，如果成功地完成了前面的 8 个半小时非常困难的实验考试，那在实践中是不会被诸如有人改了路由器密码或 IP 地址这样的问题给难倒的。我认识很多 CCIE 和不少没有通过这一考试的朋友，没有人过不了故障排除这一关。因此，4 个小时故障排除部分也从实验考试中删除了。

简而言之，CCIE 考试的书面部分由于加入了更多的物理层问题提高了难度。而实验考试缩短成了一天，8 个半小时，考试时时间的合理利用就显得非常关键了。

18.2 怎样成为一名 CCIE

做任何重要的事情的第一步是对自身目前阶段的能力做一个严格真实的评估，知道自己的长处，认识自己的不足。Cisco 的实验考试是为了淘汰没有或者只有很少实际经验的参试者。在进行自我评估时，对自己以前的经验一定要真实对待，这样才能认识到需要投入更多精力的地方。例如，很多人（除非那些成天工作在 SNA 上的）都缺乏 DLSw 方面的经验。在实验室里模拟 DLSw 就能提供实验考试需要的宝贵经验。

第一步是全身心地投入考试准备。最好准备 1 到 3 年专心致志的高强度训练以取得这一证书，包括阅读大量相关的书籍和进行数月的实验工作。必须使自己目前对网络拓扑和各种协议的理解更上一层楼。不仅仅要明白生成树如何工作，还要理解需要怎样工作，为什么需要这样工作。任何准备期间的欠缺都会造成考试临近之日的仓促以至于失败。

可能整个准备过程中最为关键的是最后的两三个月。到这个时候，所有的“正式”准备或者是“教室”学习已经完成，现在该是实验室应用。这时已经阅读了大量的书籍，应该将精力聚焦在各种技术的高级部分。在这最后的两个月里，尽自己所能想的，花尽量多的时间在实验室里配置尽量多类型的网络实例。阅读所有的配置指导书，浏览文档 CD 的内容，尽量熟悉每种协议或功能组件的设置方法。这也有助于尽快熟悉配置指导以及文档 CD 的内容。这些将是考试期间惟一获取信息的来源。虽然有这些信息可查，但是新的考试时间非常紧。

间。

最后，如果在考试的那天真的发现有很多欠缺的地方，千万不要放弃。诚然有人一次就通过了考试，但是大部分人都考了不止一次。一定要坚持；记住，“任何有价值的东西都不可能毫不费力的得到”。

祝大家好运。

18.3 CCIE：推荐读物以及知识点提纲

本书只是学习准备阶段需要阅读的众多书籍中的一本。下面列出对考试准备非常有价值的书籍，在前面提到过的 CCIE 网页上也有参考书籍的清单：

《TCP/IP Illustrated》—— Stevens 著

《Internetworking with TCP/IP》—— Comer 著

《Interconnections, Second Edition: Bridges, Routers, Switches, and Internetworkin Protocols》—— Perlman 著

《Routing TCP/IP, Volume I and II》—— Doyle 著

《Internetwork Routing Architectures》—— Halabi 著

《Cisco LAN Switching》—— Hamilton/Clark 著

《Bridges, Routers, and Switches》—— Caslow, Bruce: 著

《CCIE Network Design and Case Studies》—— Cisco Press 出版

《Cisco ATM Solutions》—— Diker-Pildush 著

表 18-1 是 CCIE 知识点的一个简单提纲（但绝对不是完整的提纲）。CCIE 参考者应该对这些知识点的内容非常熟悉以作为学习的理想起点。

表 18-1

CCIE 的知识点提纲

主 题	副 主 题
帧中继	帧中继交换 帧中继子接口 点对点 and 点对多点链路 帧中继映射：桥、LLC、DLSw 和其他关键字 RFC 1490 封装 帧中继上的桥接应用 帧中继上的语音应用 帧中继上的 PPP 应用 帧中继的 ARP 协议和逆向 ARP 解析协议 帧中继数据整形
HDLC	压缩方式
PPP	PPP 认证：PAP/CHAP PPP 回拨 PPP 多链路捆绑

续表

主 题	副 主 题
PPP	DDR 技术 压缩方式 IPCP
ISDN	拨号映射/DDR IPX、IP 和其他协议在 ISDN 上的应用 如何处理在 ISDN 上运行的路由选择协议，例如 EIGRP、OSPF、IGRP 等 快照路由 拨号监控 (dialer watch) 按需电路 复杂的 IPX 和 IP 的访问控制列表在拨号时的应用
BGP	路由反射器 环路地址的应用 路由同步规则 IBGP 和 EBGp 的区别 路由图和路由重分布 AS 路径过滤 BGP 路径选择机制 路径控制机制: MED、本地优先级、权重等 BGP 联盟 BGP 团体属性 超网的宣告、汇总 BGP 映射
OSPF	和其他路由选择协议之间的重分布 两种地址汇总命令的区别 (ip summary 和 area range) 帧中继和 X.25 上的 OSPF OSPF 按需电路 OSPF 的路由图和路由过滤 OSPF 的路径代价和管理距离 存根区域、NSSA 区域、主干区域 认证类型 I 和 II DR 和 BDR 的选择: 优先级命令 默认路由的宣告
EIGRP	IP 和 IPX 上的 EIGRP 和其他路由选择协议间的重分布 地址汇总 EIGRP 的路由图和路由过滤 MD5 加密认证 运行在 ISDN 上的 EIGRP 在点对多点网络上的水平分割 管理距离

续表

主 题	副 主 题
IGRP	和其他路由选择协议间的重分布 ISDN 上的 IGRP 快照路由 点对多点网络上的水平分割问题 默认网络 管理距离 不支持 VLSM 的问题
RIP	和其他路由选择协议间的重分布 ISDN 上的 RIP 快照路由 不支持 VLSM 的 RIPv1 的应用 RIPv2
IPX	IPX 路由选择协议：NLSP/RIP/EIGRP 静态 SAP，SAP 应用和过滤 IPX 网络过滤 NLSP、RIP 和 EIGRP 之间的重分布 ACL 在 ISDN 上的 IPX 拨号控制的应用 ISDN 上的 IPX 快照路由 IPX 通道 在点对多点网络上的水平分割 SPX 和看门狗欺骗 IPX 帧类型，例如帧类型 20
DLSW	TCP、FST、直接和 Frame Relay 对等体 备份对等体 混杂模式的对等体 边界对等体和对等体组 对等体的代价 探测帧的控制和 DLSw 中的 LLC 控制 LSAP 过滤
桥接	透明桥接 生成树算法 帧中继上的桥接 源路由桥接 远程源路由桥接 转换桥接 探测帧控制和洪泛 LSAP 过滤 综合路由和桥接 默认网关
路由和流量控制	标准访问控制列表 扩展访问控制列表 命名访问控制列表 动态路由控制列表

续表

主 题	副 主 题
路由和流量控制	路由图和策略路由 默认路由的传播
排队	加权公平排队 优先级排队 定制排队 普通和帧中继数据整形 RSVP, WRED 基本配置
一般 Cisco IOS 考点	访问服务器配置 设置寄存器值 路由和交换机的密码恢复 通过 TFTP 和自动安装配置路由器 命令控制: 超时设置、命令级别等 安全 日志
Cisco IOS 的特性	NAT: 动态、静态、地址池 NTP: NTP 认证和层 (stratum) 设置 DNS HSRP: 跟踪和优先级 IRDP 快照路由 拨号监控 移动 IP ARP 控制机制 SNMP: 只读/可写通信字, 设置和读取陷阱信息 UDP 中的 IP Forward 命令机制 GRE 通道
Catalyst 交换机	Catalyst 55xx 系列 VLAN 设置 Catalyst 39xx 系列 VLAN 设置 Catalyst 29xx 系列 VLAN 设置 VTP 域 生成树协议 端口安全设置和 IP 控制 ISL、802.1Q Trunk 中继 (Trunk) 上的 VLAN 传播和控制 VLAN 间路由 组播路由
组播路由	组加入机制 稀疏模式和密集模式的操作机制
ATM	传统 IP、路由在 ATM 上的传送 VPL, VCD 和 VCI 的定义

续表

主 题	副 主 题
ATM	ARP 控制 PVC 映射
语音	IP 上的语音传输 帧中继上的语音传输 ATM 上的语音传输 FXO、FXS 和 E&M 电路 H.323
VPN	加密类型 基于 IPSec 加密的 BGP 隧道 IPSec 传输和隧道模式 转换规则，加密图 “Key” 鉴定
已经取消的考点（在第二次修改考 点时取消）	ATM LANE AppleTalk LAT DECnet Apollo Banyan VINES ISO CLNS XNS X.25

本书的实验安排是想让大家真实感受一下现实中 CCIE 实验考试的感觉。实验考试中的很多题目这里都没有涉及。正如前面所说，没有任何一本书，至少是没有任何一本这么厚的书，可以包括所有 CCIE 实验考试的内容，尤其是内容到了一定深度。本书第 2 卷中会讲述有关 BGP、IPX、网络多播、IPSec 等方面的内容。

这里的实验分成两部分，都是计时的实验。每个实验有不同的硬件要求，可能还需要预先做一些准备工作才能够顺利完成。和通常的实验一样，这里没有答案。实验的解决方案都能在 Cisco 出版社的网站 www.ciscopress.com/1587200023 上找到。这样做的目的是希望大家在查看答案之前能够自己完成实验，尽一切可能自己提出解决的提案。而且，每一个实验练习只能给自己 8 个半小时的时间。

18.4 CCIE 实验考试模拟练习：“Skynet”的配置

18.4.1 设备清单

- 一台帧中继交换机：具有 4 个串行接口。
- 一台访问服务器/主干路由器：8 个异步接口，1 个以太网接口。

- 二台实验路由器: 1 个以太接口, 2 个串行接口。
- 三台实验路由器: 1 个令牌环接口, 2 个串行接口。
- 一台实验路由器: 2 个以太接口, 1 个令牌环接口。
- 三台以太集线器, 4 台令牌环集线器/MAU。
- 集线器和 MAU 可以用带有适当接口的 Catalyst 5000 交换机来代替。这个实验考试不会使用任何 Catalyst 交换机。这是模拟考试中惟一不需要使用交换机的实验。

18.4.2 实验准备工作: 帧中继交换机的配置

按照图 18-1 为帧中继交换机配置 PVC, 这一部分的实验考试是不需要计时的。例 18-1 就是帧中继交换机的配置示例。

例 18-1 配置帧中继交换机

```

hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 121 interface Serial1 120
frame-relay route 152 interface Serial5 151
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 110 interface Serial5 111
frame-relay route 120 interface Serial0 121
frame-relay route 130 interface Serial3 131
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 131 interface Serial1 130
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 111 interface Serial1 110

```

```
frame-relay route 151 interface Serial0 152
```

帧中继配置

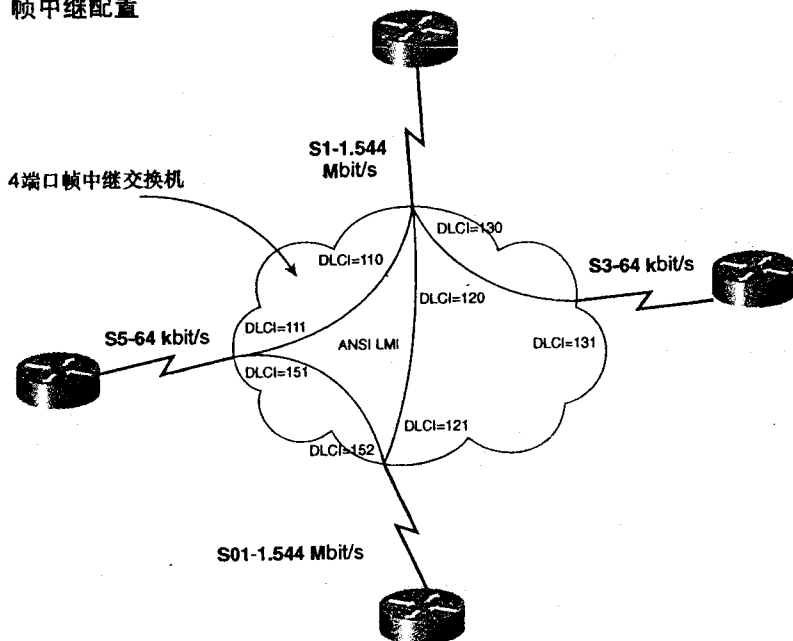


图 18-1 配置帧中继交换机

18.4.3 实验准备工作：主干路由器的配置

将路由器 R7 配置为主干路由器，并用作 IP 地址为 192.128.128.2/24 的路由器 R2 的外部 BGP 对等体，该路由器的自治系统 (AS) ID 定为 2010。按照下面的地址为该路由器配置回环接口：

128.200.1.1/24

128.201.1.1/24

128.202.1.1/24

用 **network** 命令宣告网络的路由。例 18-2 是主干路由器 R7 的配置范例。

例 18-2 主干路由器的配置

```
hostname r7_backbone_router
!
interface Loopback20
 ip address 128.200.1.1 255.255.255.0
!
interface Loopback21
 ip address 128.201.1.1 255.255.255.0
!
interface Loopback22
```

(待续)

```
ip address 128.202.1.1 255.255.255.0
!
interface Ethernet1
description place in vlan 3 - backbone 1
ip address 192.128.128.1 255.255.255.0
media-type 10BaseT
ip rip send version 2
ip rip receive version 2
!
router rip
version 2
no auto-summary
network 192.128.128.0
network 128.200.0.0
network 128.201.0.0
network 128.202.0.0
!
router bgp 2010
no synchronization
network 192.128.128.0
network 128.200.0.0
network 128.201.0.0
network 128.202.0.0
neighbor 192.128.128.2 remote-as 2001
neighbor 192.128.128.2 ebgp-multihop 10
!
```

下面这些实验考试内容都属于计时范围，在帧中继交换机和主干路由器的配置完成之后，可以开始进行下面的实验考试内容。

18.4.4 计时实验考试部分

实验考试规则

- 除非专门指定，不要使用静态路由或浮动静态路由。
- 严格按照要求进行设计和配置，何时何地转发路由，PVC 的使用都要按要求进行。
- 前一部分配置内容在第 2 部分可能需要改动，进行第 2 部分的配置之前一定要先完成前面部分的配置工作。
- 实验考试时惟一的参考资料只能是所提供的配置指南和 Cisco 文档 CD 盘。
- 实验考试时间一共 8 个半小时。考试进行的过程中不要相互交谈。
- 在开始操作之前，建议先将整个的实验内容通读一遍。
- 要求大家提供正确、清楚的网络说明。
- 实验物理拓扑参考如图 18-2 所示。

第 1 节：基本 IP 配置

1 访问服务器：配置访问服务器/路由器，以使所有的路由器和交换机都可以通过反向 Telnet 对其进行访问。对所有的路由器和交换机实施密码保护，密码为 cisco。

2 IP 地址的分配：按照图 18-2 为所有的物理接口分配 IP 地址，所有接口上都以 140.100.

$x.x$ 为主网。除下列接口以外的所有接口都使用 24 位掩码：

CCIE 实验考试模拟练习“Skynet”

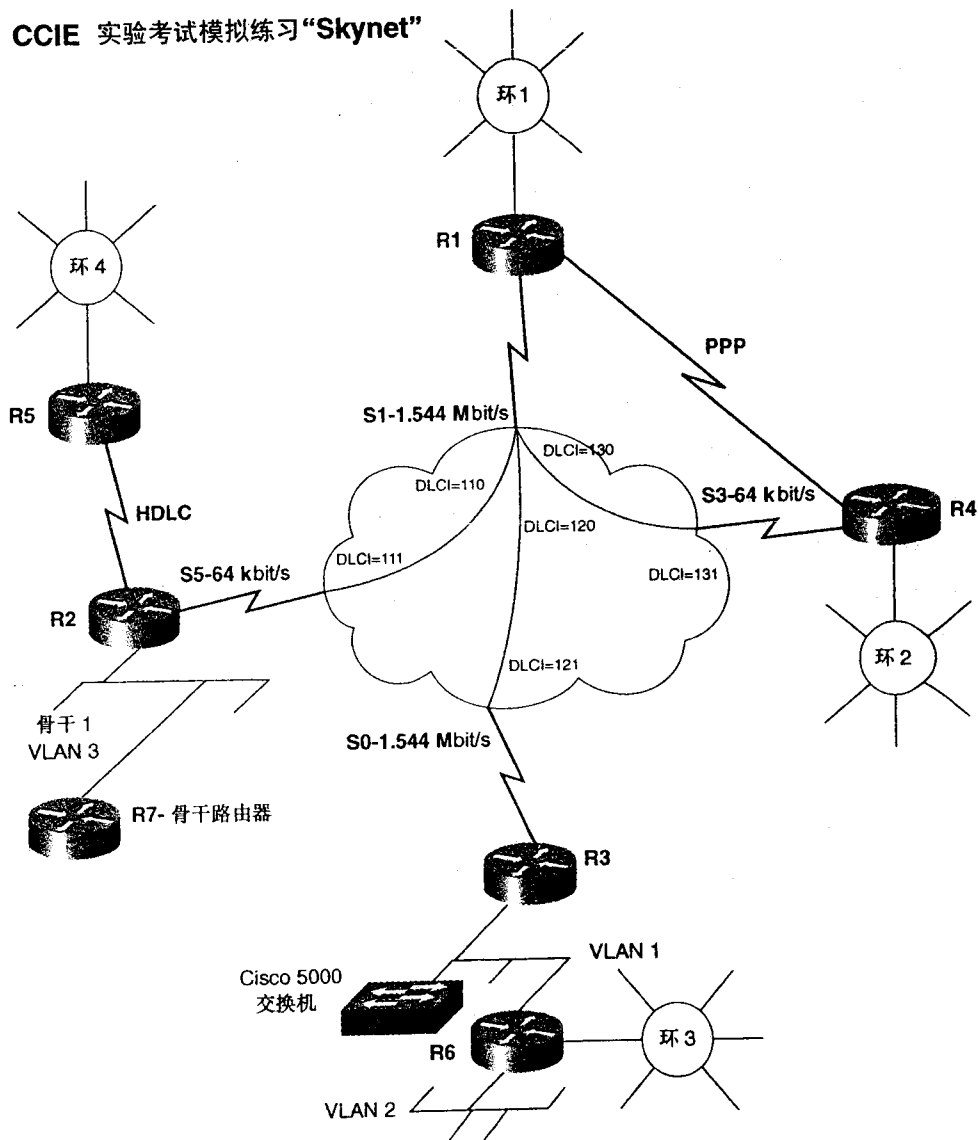


图 18-2 “Skynet”网络拓扑

R1: 令牌环接口上允许包含 30 个主机地址。在 R1 与 R4 之间的 PPP 和帧中继链路上采用 30 位的地址。

R2: 以太接口的 IP 地址为 192.128.128.2/24。在 R2 与 R5 之间的 HDLC 链路上分配一个 30 位的子网。

R3: 一个回环接口上配置 IPX 网络 DEAD。

R4: 令牌环接口采用 27 位掩码，并用从 199.199.1.1/24 到 199.199.10.1/24 的 IP 地址配置 10 个回环接口。

R5: 令牌环接口需要支持 14 个主机，并用 151.100.1.1/24, 151.101.1.1/24 和 172.16.1.1/24 这 3 个 IP 地址配置 3 个回环接口。在 R2 与 R5 之间的 HDLC 链路上配置一个 30 位的子网。

R6: 为该路由器在 VLAN 2 中的以太接口分配 IP 地址 10.10.10.1/24。

除非特别指出，所有以太、令牌环和回环接口都需要建立完整的 IP 和 IPX 连接。

3 建立网络的完整文档资料，包括所有的 OSPF 区域、IP/IPX 地址、IPX 网络等等。

第 2 节: Catalyst/LAN 的配置

1 按照图 18-2 对 VLAN 进行配置，同时还包括配置令牌环网段和 R2 与主干路由器网段的连接。

第 3 节: OSPF 与帧中继的配置

1 按照图 18-2 对帧中继网络进行配置，只有路由器 R1 上才可以使用子接口，只能按照图中所示使用 DLCI 来对数据进行路由，路由器 R2 到 R3 的数据必须都要通过 R1。

2 路由器 R1, R2 和 R3 应该共享同一个 IP 子网。这些路由器之间配置 OSPF 的 Area 0 区域，但是配置 OSPF 时不能使用 `ip ospf network` 命令。同时将路由器 R1 和 R4 之间的帧中继链路放入到 OSPF Area 0 中。

3 配置路由器 R1 和 R4 之间的 PPP 链路为 OSPF Area 10。令牌环 Ring 2 则应该在 OSPF Area 20 中，这条链路只有在帧中继服务中断时才进入工作状态。

4 对路由器 R3 上串行链路 OSPF 的 hello 定时间隔进行配置，使得其 hello 数据包每 60 秒通过网络广播发送一次。

5 在路由器 R4 上配置 OSPF 的时候，需要宣告 10 个环路地址，从 199.199.1.1 到 199.199.10.1，但不要将这些回环接口分配到 OSPF 区域中。

第 4 节: 路由转发协议及其重分布的配置

1 网络中，在且仅在路由器 R3/R6 之间的 VLAN 1 和 R6 的令牌环 Ring 3 上配置 IGRP 路由选择协议，此外还要禁止 IGRP 网络广播信息传到 VLAN 2 上。

2 在路由器 R2 到主干路由器的链路上配置 RIP 2 路由选择协议，将 R2 配置成只能接收和发送主干路由器的 RIP 2 更新信息。

3 禁止路由器 R4 上的令牌环子网转发到 R7。

4 在路由器 R2/R5 之间的 HDLC 链路上运行 EIGRP 路由选择协议，在路由器 R4 的令牌环 Ring 4 上也运行 EIGRP 路由选择协议，自治系统 (AS) ID 定为 2020。

5 将环路网络 151.100.1.1 和 151.101.1.1 聚合为一条路由进行发布。确保路由器 R6 在没有使用静态路由的情况下可以 ping 通所有的 EIGRP 网络。

第 5 节: IPX 的配置

1 在路由器 R5 的令牌环 Ring 4 和路由器 R2 上配置 IPX NLSP。

- 2 在且仅在所有的 LAN 接口上配置 IPX RIP/SAP。
- 3 在帧中继 WAN 链路上配置 IPX EIGRP。
- 4 在路由器 R6 上设置一个距离该路由器的令牌环接口有 3 跳距离的静态 SAP。这个 SAP 提供打印服务，其配置也因依此而行。
- 5 禁止路由器 R3 将这个 SAP 传播到 WAN 中去。
- 6 路由器 R3 配置一个用于与 IPX 网络 DEAD 相连的回环接口，而且仅仅禁止路由器 R4 接收到该路由信息。

第6节：桥接配置

- 1 在 VLAN 1 和 VLAN 3/Backbone 1 之间通过帧中继网络的透明桥接传输 SNA。
- 2 强制指定路由器 R1 为生成树根桥接。

第7节：各种 IOS 相关的配置

- 1 在路由器 R1 和 R4 之间的 PPP 链路上配置 CHAP 认证方式，密码采用 ccie。
- 2 在路由器 R5 上应用入过滤列表，过滤来自路由器 R4 上从 199.199.1.1 到 199.199.10.1 环路地址的偶数子网。
- 3 在路由器 R6 上应用入过滤列表，使得只有路由选择协议、ping 数据包和 WWW 数据包才能进入本路由器去访问 Ring 3 和 VLAN 2 上的 WWW 主机。
- 4 用户 skynet 是位于路由器 R1 上，允许该用户访问 Ring 3 和 VLAN 2 上的所有 IP 服务，该用户的有效访问时间是 10 分钟。

第2部分

第8节：BGP 的配置

- 1 路由器 R1、R2 和 R3 都处在自治系统 (AS) 2001 中，R5 则是在自治系统 (AS) 65001 中。配置从路由器 R2 到在自治系统 (AS) 2010 中 192.128.128.1 的 EBGP 对等体。此外，再配置一个从路由器 R5 到 R2 的 EBGP 对等体。
- 2 配置路由器 R2 到 R1 和 R1 到 R3 的 IBGP 对等体，R2 到 R3 则不要配置对等体。通过配置保证所有的 BGP 路由器都可以从主干路由器接收到路由 128.20x.1.0。
- 3 配置路由器 R5 使其将超网 128.200.0.0 发送到所有其他网络部分中去，包括 R1、R2、R3、R4、R6、R7，发送时屏蔽 128.200.0.0 中所有的子网路由。
- 4 配置路由器 R5 使其通过 BGP 发送子网 172.16.1.0/24 的 metric 值为 75。
- 5 主干路由器接收到路由 172.16.1.0 的时候，通过配置让主干路由器认为该路由来自 AS 2011。
- 6 将路由器 R5 上的 BGP hello 数据包发送间隔改为 5 分钟。

第9节: DLSw 的配置

- 1 在 Ring 2 和 Ring 3 之间配置一个 DLSw TCP 类型对等体。
- 2 在 Ring 1 和 Ring 4 之间配置一个 DLSw FST 类型对等体。
- 3 在路由器 R1 和 R4 之间配置另一个对等体，使得 Ring 3 到 Ring 4 可以实现 NetBIOS 的互访，同时还要使得探测帧数据量最少。
- 4 配置一个 Ring 2 到 Ring 1 的 TCP 类型对等体，并保持该对等体一直处于工作活动状态，即使帧中继服务中断也是如此。
- 5 配置路由器 R5 的对等体，仅仅转发去往 MAC 地址 3745.0001.0101 的数据帧。

第10节: 各种 Cisco IOS 特性的设置

- 1 路由器 R6 的 VLAN 2 中 IP 地址属于 10.0.0.0/8 的网络不能够通过路由选择协议进行转发。对这个 VLAN 中的用户进行配置，使得它们无需更改其 IP 地址或者是转发子网路由 10.10.10.0 就可以对整个网络进行访问。
- 2 在路由器 R1, R3 和 R6 上配置组播路由方式。R6 应该属于组播组 224.10.10.10，并且应该对 R3 和 R1 发出的 ping 做出响应。

18.5 CCIE 实验考试的模拟练习: “Darth Reid”

18.5.1 设备清单

- 一台帧中继交换机: 具有 4 个串行接口。
- 一台访问服务器/主干路由器: 8 个异步接口, 1 个以太网接口。
- 一台主干路由器: 2 个以太网口。
- 二台实验路由器: 1 个以太网接口, 1 个 ISDN 接口, 1 个串行接口。
- 二台实验路由器: 1 个以太网接口, 1 个令牌环接口, 1 个串行接口。
- Catalyst 5000 型交换机。
- Catalyst 3900 型交换机。

18.5.2 实验准备工作: 帧中继交换机的配置

按照图 18-3 为帧中继交换机配置 PVC, 这一部分的实验是不需要计时的。例 18-3 就是帧中继交换机的配置示例。

例 18-3 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 148000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
 frame-relay route 121 interface Serial1 120
 frame-relay route 152 interface Serial5 151
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 148000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
 frame-relay route 110 interface Serial5 111
 frame-relay route 120 interface Serial0 121
 frame-relay route 130 interface Serial3 131
!
interface Serial3
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 131 interface Serial1 130
!
interface Serial5
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 111 interface Serial1 110
 frame-relay route 151 interface Serial0 152
!
```

18.5.3 实验准备工作：主干路由器的配置

将路由器 R6 配置为一台主干路由器，并作为路由器 R1 地址 160.100.2.1 和 R7 地址 160.100.1.1 的外部 BGP 对等体。该路由器的自治系统 (AS) ID 定为 2001，按照下面的地址为该路由器配置回环接口：

160.100.100.1/24

197.192.100.1/24

197.192.101.1/24

197.192.102.1/24

帧中继配置

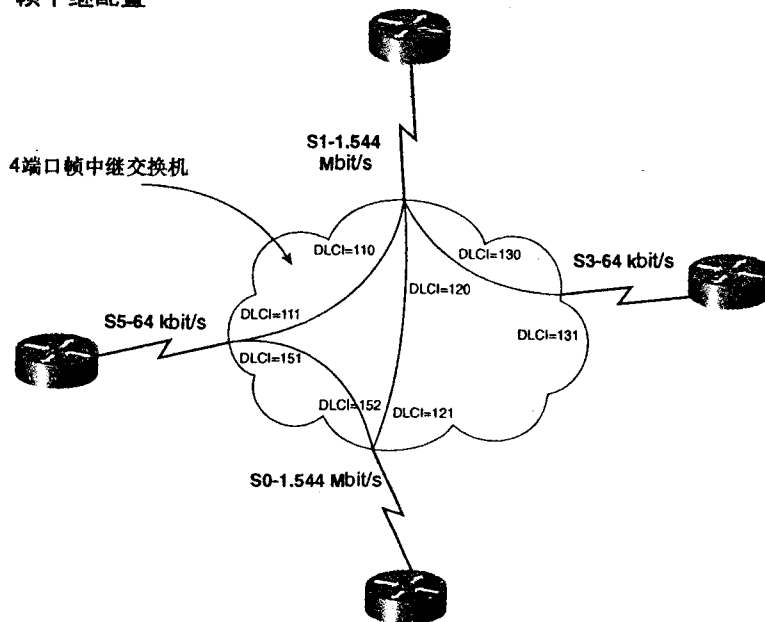


图 18-3 帧中继交换机的配置

用 **network** 命令宣告这些网络的路由。主干路由器 R6 的配置如例 18-4 所示。

例 18-4 主干路由器的配置

```
hostname r6_backbone_router
!
interface Loopback20
 ip address 192.190.100.1 255.255.255.0
!
interface Loopback21
 ip address 192.190.101.1 255.255.255.0
!
interface Loopback22
 ip address 192.190.102.1 255.255.255.0
!
interface Loopback23
 ip address 160.100.100.1 255.255.255.0
!
interface Loopback24
 ip address 160.100.128.1 255.255.255.0
!
interface Loopback25
 ip address 160.100.129.1 255.255.255.0
!
interface Loopback26
 ip address 160.100.130.1 255.255.255.0
!
interface Ethernet1
 description place on vlan 20 - Backbone 1
 ip address 160.100.2.254 255.255.255.0
```

```

media-type 10BaseT
!
interface Ethernet2
description place on vlan 10 - Backbone 2
ip address 160.100.1.254 255.255.255.0
media-type 10BaseT
!
router rip
passive-interface Ethernet1
network 160.100.0.0
network 192.190.100.0
network 192.190.101.0
network 192.190.102.0
!
router bgp 2001
no synchronization
network 160.100.100.0 mask 255.255.255.0
network 160.100.128.0 mask 255.255.255.0
network 160.100.129.0 mask 255.255.255.0
network 160.100.130.0 mask 255.255.255.0
neighbor 160.100.1.1 remote-as 2010
neighbor 160.100.1.1 ebgp-multihop 10
neighbor 160.100.2.1 remote-as 2010
neighbor 160.100.2.1 ebgp-multihop 10
!
ip route 133.10.0.0 255.255.0.0 160.100.2.1

```

下面这些实验考试的内容都是计时的，在帧中继交换机和主干路由器的配置完成之后，开始进行下列实验考试内容。

18.5.4 计时实验考试部分

实验考试规则

- 除非专门指定，不要使用静态路由或浮动静态路由。
- 严格按照要求进行设计和配置，何时何地转发路由，PVC 的使用都要按要求进行。
- 前一部分配置内容在第 2 部分可能需要改动，进行第 2 部分的配置之前一定要先完成前面部分的配置工作。
- 实验考试时惟一的参考资料只能是所提供的配置指南和 Cisco 文档 CD 盘。
- 实验考试时间一共 8 个半小时。考试进行的过程中不要相互交谈。
- 在开始操作之前，建议先将整个的实验内容通读一遍。
- 要求大家提供正确、清楚的网络说明。
- 参考的实验布局如图 18-4 所示。

第 1 节：基本 IP 的配置

1 访问服务器：配置访问服务器/路由器，要求所有的路由器和交换机都可以通过反向 Telnet 对其进行访问。对所有的路由器和交换机实施密码保护，密码为 cisco。

2 IP 地址的分配：按照图 18-6 为所有的物理接口分配 IP 地址。所有接口上都以 133.10.0.0/24 为网络地址。

CCIE 实验考试模拟练习 “Darth Reid”

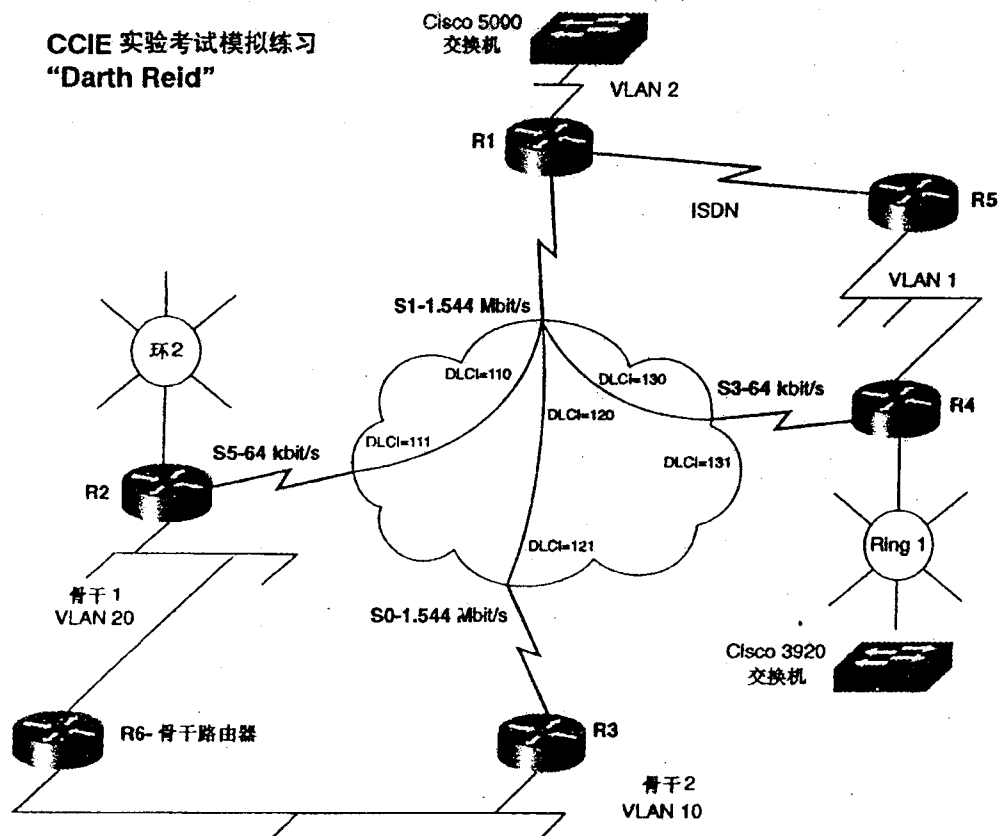


图 18-4 “Darth Reid” 的网络拓扑

4 建立网络的完整文档资料,包括所有的 OSPF 区域、IP/IPX 地址、IPX 网络等等。

第 2 节: Catalyst 交换机的配置

子书仅限试看之用，禁止用于商业行为，并请于下载后24小时内删除，如您喜欢本书，请购买正版。若因私自散布造成法律问题，本人概不负责。

- 2 将 Backbone 1 和 Backbone 2 这两个网段配置到两个任选的单独 VLAN 中去。
- 3 路由器 R1 的以太网接口与 Catalyst 5K 交换机的 VLAN 2 相连，而 R4/R5 的以太网段则与 Catalyst 5K 交换机 VLAN 1 相连，在 VLAN 2 的 IP 范围内分配一个 IP 地址。
- 4 对 Catalyst 交换机进行配置，使得仅有 VLAN 1 上的用户可以通过 Telnet 进行访问，这里所有的路由器对该交换机的访问都不受访问列表的限制。
- 5 对交换机进行配置，使得仅有 VLAN 2 的所有数据在其接口 2/10 上可以通过嗅探器或协议分析仪进行监测。
- 6 将 VLAN 2 的生成树参数 MAXAGE 设置为 25 秒。
- 7 对 VLAN 2 进行配置，使得它的接口上惟一可以接入设备是路由器 R1。如果 R1 从这个接口拆除而接上另外一台设备，catalyst 交换机会将该接口置为非激活状态。
- 8 对令牌环交换机进行配置，使得可以通过 Telnet 对它进行管理和配置。

第 3 节：OSPF 和帧中继的配置

- 1 按照图 18-4 对帧中继网络进行配置，只有路由器 R1 上才可以使用子接口，只能按照图中所示使用指定 PVC 来对数据进行路由，从路由器 R2 到 R3 的数据必须先通过 R1。
- 2 路由器 R1、R3 和 R4 共享同一 IP 子网。将这些路由器之间的区域作为 OSPF Area 0，在配置 OSPF 的时候，不要改变 `ip ospf network` 的类型。
- 3 配置路由器 R4/R5 的以太网段为 OSPF Area 30，配置 R4 的令牌环网段在 Area 30 中。在路由器 R5 中加上一个 IP 地址为 192.168.1.1/24 的回环接口，并将此地址配置在 OSPF Area 50 中，同时应保证 R3 以外所有路由器都可以访问该地址。
- 4 在路由器 R1 和 R2 之间的 PVC 上配置帧中继的数据整形，使之对 BECN 进行响应。供应商提供的 CIR 是 32K。路由器 R2 的本地接口速率为 62K，而 R1 的本地接口速率为 1.54 Mbit/s。

第 4 节：ISDN 的配置

- 1 将路由器 R1 和 R5 之间的 ISDN 接口链路配置为 OSPF 主干线路的一部分，配置路由器使只有 R5 可以发起呼叫，在继续下一步之前一定要保证本地的 ISDN 接口可以通过 `ping` 命令的测试。
- 2 仅有在检测到 OSPF 的拓扑发生变化而且数据是向其他路由器传送时，路由器 R5 才会进行一次呼叫，这个时候的路由传输应该是在 R1 和 R5 之间进行的。

第 5 节：路由选择协议以及重分布的配置

- 1 在路由器 R1 和 R2 之间的帧中继链路上配置 IGRP 路由选择协议。R2 的令牌环接口和 R1 的以太网接口也应该在 IGRP 域中。同时还要保证所有的 OSPF 路由信息都被 R2 学到，而且 R2 可以通过源地址 `ping` 访问所有配置的地址。
- 2 在路由器 R1、VLAN 2 和 R4 Ring 1 之间配置 EIGRP 路由协议。
- 3 在路由器 R3 与 Backbone 1 相连的以太网接口上配置 RIP 路由协议，并将该接口的 IP 地址

址定为 160.100.1.1/24。同时确保在该接口上不运行 OSPF，而是把 RIP 重分布进 OSPF。此外，配置路由器 R3 可以用 ping 命令访问 R2 的令牌环接口。

4 在 RIP 配置完毕后，在路由器 R3 上应该看到 RIP 的路由信息。对这些路由进行过滤使得只有路由 192.190.102.0/24 会重分布到实验网络中去。此外，禁止网络中任何路由条目发布到 Backbone 1 上去。

5 在路由器 R4 上添加一个 IP 地址为 161.100.1.1/24 的回环接口，将此回环接口放置到 EIGRP 域中去。对 EIGRP 进行聚合使得这个接口上只出现 160.100.1.0 和 161.100.1.0 这两个路由。

6 将路由器 R2 与 Backbone 1 相连的以太网接口的 IP 地址设为 160.100.2.1/24。对于这个接口，不允许用 network 命令把它放置到 IGRP 域中去，但是要保证它与整个网络的完全 IP 互连。

第 6 节: BGP 的配置

1 在路由器 R2 和 R3 上配置 BGP，自治系统 (AS) ID 采用 2010。在自治系统 (AS) 2001 中分别配置到 160.100.1.254 和 160.100.2.254 这两个地址的 EBGP 对等体，从而在此自治系统 (AS) 中产生两个出口点。同时在路由器 R1 上配置 BGP，并且在该路由器与 R2 和 R3 之间分别建立 IBGP 邻居路由器的关系。R1 应该包含来自自治系统 (AS) 2001 的多条路由。

2 在路由器 R1 上通过 BGP 路由选择协议发送路由 128.200.1.0/24。

3 对路由器 R2 和 R3 进行配置，使得所有接收到 BGP 路由的 weight 值为 700。

4 通过配置使得所有去往 AS 2001 的路由不会再从 AS 2001 转发到其他自治系统 (AS) 中去。

5 将来自 AS 2001 的路由聚合为一条路由，然后重分布到 OSPF 中去。

6 确保你可以通过 ping 命令访问包括 160.100.100.1 在内的所有 BGP 路由，即使路由器 R2 或 R3 的以太网接口停止工作也不例外。

第 7 节: 各种 Cisco IOS 特性的配置

1 在路由器 R1 的以太网段上，配置一份用于数据过滤的访问控制列表，禁止来自下列应用的数据：(列表中命令行要尽量少的)

拒绝来自 131.24.194.x 的 FTP 和 HTTP 数据

拒绝来自 131.25.194.x 的 FTP 和 HTTP 数据

拒绝来自 135.152.1.1 的 FTP 和 HTTP 数据

拒绝来自 227.24.194.x 的 FTP 和 HTTP 数据

拒绝来自 131.24.195.x 的 FTP 和 HTTP 数据

拒绝来自 131.24.196.x 的 FTP 和 HTTP 数据

2 将路由器 R1、R4 和 R5 配置为组播 224.10.10.1 的成员，对于 R1 和 R4，则要求它们必须可以在二者所在的两个 VLAN 之间直接通过组播传输数据，至于组播传输的模式则可以自行选择。

3 在路由器 R4 和 R5 之间配置 HSRP，R4 作为主默认网关。如果 R4 的帧中继接口出现故障，默认网关将由 R5 的以太网接口来代替。

第 2 部分

第 8 节：IPX 的配置

1 除了回环接口和主干路由器接口之外的所有接口上都需要配置 IPX 协议，在帧中继接口上采用 IPX EIGRP，而所有的 LAN 接口则采用 IPX RIP。

2 在路由器 R5 上配置一个静态的 SAP，该 SAP 用于提供文件服务，名为 FILESERV，它离网络 0xBB00 的距离是两个跳计数的距离。如果需要，可以考虑使用静态路由。

3 禁止路由器 R3 访问这个 SAP FILESERV。

4 在路由器 R1 和 R5 的 ISDN 链路上配置 IPX 协议，使得路由可以通过 ISDN 链路从 R5 传送到 R1。

第 9 节：DLSw 的配置

1 在 VLAN 2 和 Ring 2 之间配置一个 DLSw 的 TCP 类型对等体，在路由器 R2 上则不允许配置远程对等体。

2 从 VLAN 2 到 Ring 2 的链路上只允许 SNA 数据传输，在访问控制列表（ACL）中使用的命令行要尽量少。

3 在 VLAN 2 上出现了大量的 IP 数据包碎片，对这里的 DLSw 进行调整，使得 IP 数据包的碎片不要如此频繁地大量出现。

第 10 节：桥接的配置

1 在从 Ring 1 到 Ring 2 的链路上配置远程源路由桥接。每个令牌环中都有一台 IBM 大型机。在路由器 R2 和 R4 之间的串行链路上存在着大量的重发数据，这样就导致了很多的数数据帧副本在数据帧本身到达远程主机的同时也到达了这些远程主机。这样的问题就中断了 LLC2 的正常运行，从而导致两台 IBM 大型机之间的话路出现丢失的现象。想办法解决这个问题，同时还要把将来的网络扩展考虑进去。

18.6 CCIE 实验考试的模拟练习：“The Lab, the Bad, the Ugly”

18.6.1 设备清单

- 一台帧中继交换机：4 个串行接口。
- 一台访问服务器：6 个异步接口，1 个以太网接口。

- 二台实验路由器：1个以太网接口，2个串行接口，1个 ISDN BRI 接口。
- 二台实验路由器：1个令牌环接口，2个串行接口。
- 一台实验路由器：1个以太网接口，2个串行接口。
- 两台令牌环集线器/MAU。
- 一台 Catalyst 5000 交换机。
- 如果还有一台 Catalyst 交换机，那就可以用于 vlan 中继的模拟，当然，这并不是必须的。

18.6.2 实验准备工作：帧中继交换机的配置

按照图 18-5 为帧中继交换机配置 PVC，这部分的实验考试不需计时。例 18-5 是帧中继交换机的配置示例。

例 18-5 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 121 interface Serial1 120
frame-relay route 152 interface Serial5 151
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 110 interface Serial5 111
frame-relay route 120 interface Serial0 121
frame-relay route 130 interface Serial3 131
frame-relay route 140 interface Serial3 141
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 131 interface Serial1 130
frame-relay route 141 interface Serial1 140
!
interface Serial5
no ip address
encapsulation frame-relay
```

(待续)

```

clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 111 interface Serial11 110

frame-relay route 151 interface Serial0 152
!

```

帧中继的配置

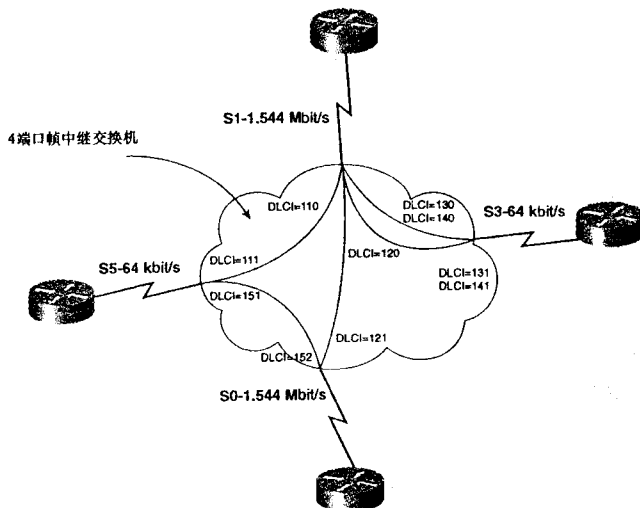


图 18-5 帧中继交换机的配置

下面这些实验考试的内容计时，在帧中继交换机和主干路由器的配置完成之后，可以开始进行下面的实验考试的内容。现在则不要进行任何路由器的配置。

18.6.3 计时实验考试部分

实验考试规则

- 除非专门指定，不要使用静态路由或浮动静态路由。
- 严格按照要求进行设计和配置，何时何地转发路由，PVC 的使用都要按要求进行。
- 前一部分配置内容在第 2 部分可能需要改动，进行第 2 部分的配置之前一定要先完成前面部分的配置工作。
- 实验考试时惟一的参考资料只能是所提供的配置指南和 Cisco 文档 CD 盘。
- 实验考试时间一共 8 个半小时。考试进行的过程中不要相互交谈。
- 在开始操作之前，建议先将整个的实验内容通读一遍。
- 要求大家提供正确、清楚的网络说明。
- 参考的实验布局如图 18-6 所示。

第 1 节：基本 IP 的配置

Telnet 对其进行访问。对所有的路由器和交换机实施密码保护，密码为 cisco。

2 IP 地址的分配:按照图 18-6 为所有的物理接口分配 IP 地址,所有接口上都以 165.10.x.x 为主网。除下列接口以外所有接口都使用 24 位掩码:

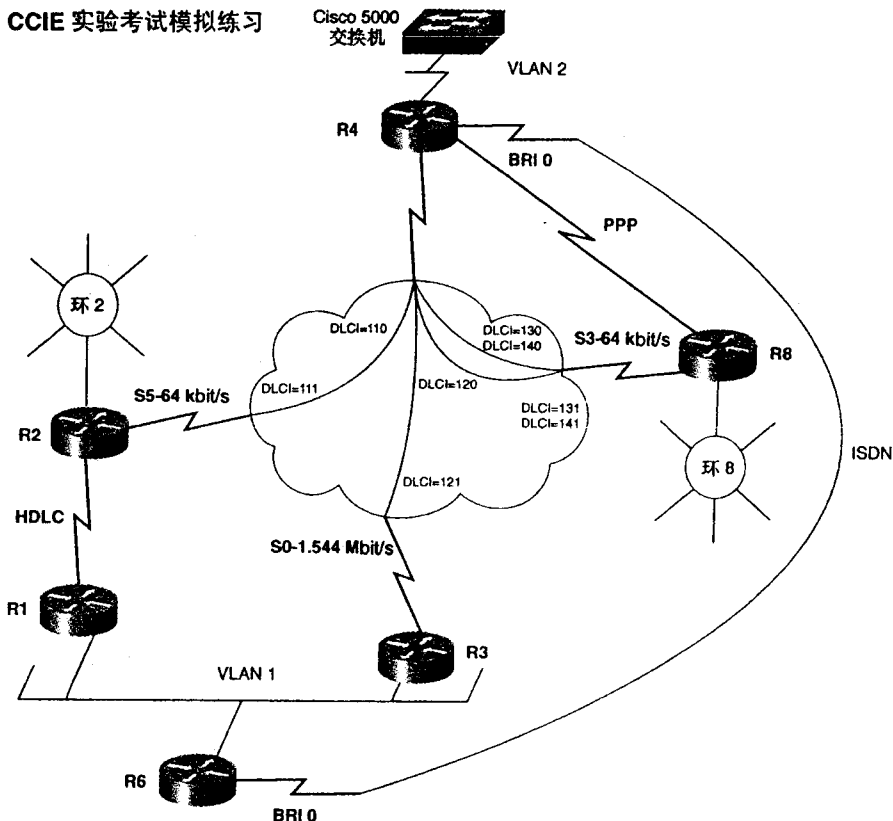


图 18-6 “The Lab, the Bad, the Ugly” 网络拓扑图

R1: 在路由器 R1 和 R2 之间的 HDLC 链路上采用 30 位的地址。

R2: 该路由器的回环接口 IP 地址为 172.16.1.1/24。

R4: 该路由器的以太网接口 IP 地址为 165.10.10.1，回环接口地址为 200.128.1.1/24。

R8: 该路由器的令牌环接口 IP 地址为 10.10.10.1/24。

3 除非特别指出，所有以太、令牌环和回环接口都需要建立完整的 IP 和 IPX 连接。

4 建立网络的完整文档资料，包括所有的 OSPF 区域、IP/IPX 地址、IPX 网络等等。

第 2 节: Catalyst/LAN 的配置

1 按照图 18-6 对 VLAN 进行配置，路由器 R1，R3 和 R6 都位于 VLAN 1 中，而 R4 则处在 VLAN 2 中。同时也对令牌环网段进行配置。

2 在接口 1/1 上配置为全双工，并启动 ISL Trunk。对生成树进行配置使得生成树的根总是本交换机。如果以后还有一台交换机连接到网络中，对整个网络的配置没有影响。

3 对 VTP 进行配置，实现对 ETP 更新信息的监听和转发，但是该信息不用于 VLAN 的数据库更新。

第3节：OSPF 和帧中继的配置

- 1 按照图 18-6 对帧中继网络进行配置。
- 2 路由器 R1、R3 和 R4 共享同一 IP 子网，将这些路由器之间的网段配置为 OSPF 的 Area 10。
- 3 将 VLAN 1 配置在 OSPF 的 Area 0 中，路由器 R2 的 Ring 2 在 Area 100 中，R1/R2 之间的 HDLC 链路则在 Area 30 中。
- 4 在路由器 R1 和 R2 之间配置 Type 2 认证方式，密码定为 lbu。
- 5 在主干区域路由器日志中记录 OSPF 相邻关系的任何变化。

第4节：路由选择协议以及路由重分布的配置

- 1 在 R4 与 R8 之间的帧中继线路以及 VLAN 2 上配置 IGRP 路由选择协议，AS ID 设为 2010，这个 AS 对于 OSPF 网络来说，应该是完全可达的。
- 2 路由器 R4/R8 之间配置的 PPP 备用链路仅当帧中继服务出现中断的时候才会进入工作状态。工作的时候，PPP 链路应该可以传输所有路由，但是不能在这里使用静态路由。
- 3 禁止路由器 R8 的令牌环网络路由传输到 R4。

第5节：ISDN 的配置

- 1 对路由器 R4 和 R6 的 ISDN 接口进行配置，在继续工作之前，一定要确保可以 ping 通 R4/R6 的 ISDN 接口。
- 2 在这个 ISDN 链路上配置 CHAP 认证方式，密码采用 ccie。
- 3 将这些 ISDN 接口配置在 OSPF 的 Area 20 中。

第6节：IPX 的配置

- 1 在 VLAN 1，VLAN 2，Ring 2 和 Ring 8 上配置 IPX 协议。
- 2 在路由器 R1，R2，R3 和 R6 上配置 IPX EIGRP。从路由器 R3 到 R4 通过帧中继网络创建一条隧道。同时确保所有 IPX 网络对所有路由器都具有可见性。
- 3 在路由器 R4 和 R8 之间通过帧中继网络配置 IPX RIP/SAP，同时通过配置使得从路由器 R4 到 R8 传输 IPX 数据的 PVC 与传输 IP 数据的 PVC 不同。
- 4 在 Ring 2 上配置一个提供打印服务的 IPX SAP，该 SAP 取名为 fakeprint，它使用的是 socket 451。此外，还要保证路由器 R4 可以使用这个 SAP。
- 5 在路由器 R6 上设置一个静态 SAP，它距离该路由器的以太网接口是 3 跳距离。这个 SAP 也是为了提供打印服务，其配置要与此相符。

第 7 节: 桥接的配置

1 利用透明转发桥接将 SNA 数据在从 VLAN 1 到 VLAN 2 的帧中继网络上传输, R4 则作为生成树的根。

2 路由器 R4 的 VLAN 2 上 IPX 数据是通过桥接和路由进行传输的。

第 8 节: 各种 Cisco IOS 特性的配置

1 按照后面这些规则在路由器 R4 上配置队列: 数据帧字节大小为 500 的 EIGRP, OSPF 和 IPX 数据需要 25% 的链路, 数据帧字节大小为 1412 的 WWW 数据需要占用 10% 的链路, 而默认队列的数据帧大小为 700, 它需要使用 65% 的链路。

2 路由器 R8 上的子网 10.10.10.0/24 不能用任何路由选择协议进行传播。在这个子网络上两个主机 10.10.10.5 和 10.10.10.10, 要求可以访问整个网络, 两台主机经过转换之后的 IP 地址允许采用相同的地址。

3 VLAN 2 上有一台大型机, 具有 3 个 IP 地址: 165.10.10.100, 165.10.10.101 和 165.10.10.102。这 3 个 IP 地址的 MAC 地址相同, 为 2200.0001.0001。配置路由器 R4 来将数据转发到 3 个 IP 地址共用的这个 MAC 地址。

4 在路由器 R6 的以太网接口应用一个出过滤列表, 使得仅有路由选择协议, ping 和 WWW 数据可以从这个接口发送出去。

第 2 部分

第 9 节: BGP 的配置

1 路由器 R1, R3 和 R6 都处在 AS 2001 中, R2 是在 AS 5 中, 而 R4 则是在 AS 2010 中的。配置一个从 R2 到 R1 的 EBGP 对等体和一个从 R3 到 R4 的 EBGP 对等体, 这样到 AS 2001 中有两个出口点。

2 在路由器 R4 上分配一个回环接口 200.128.1.0/24, 并通过 BGP 进行转发。在路由器 R2 上则分配一个回环接口 172.16.1.0/24, 也通过 BGP 进行转发。

3 分别从路由器 R1 到 R3, R1 到 R6 各配置一个 IBGP 对等体, 从 R3 到 R6 则不要配置对等体。同时确保这些路由信息可以为所有 BGP 路由器接收到。

4 在整个 BGP 网络中对 AS 的路径进行管理, 使得从 AS 5 到 AS 2010 的首选路径为从 AS 5 到 R2 到 R1 到 R3 再到 R4 的这样一条路径, 而从 AS 2010 到 AS 5 的首选路径则是从 AS 2010 到 R4 到 R3 到 R1 再到 R2。

第 10 节: DLSw 的配置

1 在 R2 的 Ring 2 和 R8 的 Ring 8 之间配置一个 DLSw 的 FST 类型对等体。在 Ring 2 和 Ring 8 之间过滤所有以字母 “lab” 开头的 NetBIOS。

2 在 R2 的 Ring 2 和 R6 的 VLAN 1 之间配置一个 DLSw 的 TCP 类型对等体。

3 通过配置使得在 R2 丢失了它与 R6 之间的对等体的情况下会有另一个指向 R1 的对等体进入激活状态。R2 与 R1 之间的对等体及其链接只有在 R2 与 R6 的对等体丢失的情况才能作为备用进入工作状态，在 R2 与 R6 之间对等体的 TCP 链接恢复 5 分钟之后，这个 R2 与 R1 的对等体应该终止其 LLC2 会话。

第 11 节：各种 Cisco IOS 特性的配置

1 将路由器 R8 的静态 RIF 配置成这样：

Ring8-Bridge7-Ring9-Bridge11-Ring10-主机 (MAC 地址 2200.600E.900E)

2 将路由器 R2 配置为一台 NTP 服务器。把 R6 的时钟和 R2 同步。对 NTP 进行 MD5 加密，密码是 cisco。

3 在路由器 R3 的帧中继接口上配置优先级队列，Telnet，EIGRP 和 IPX 数据为优先级高的队列，而 WWW 数据则为优先级低的队列。

所有没有提及的都应该按照默认值进行设置。

18.7 CCIE 实验考试的模拟练习：“The Enchilada”

18.7.1 设备清单

- 一台帧中继交换机：5 个串行接口。
- 一台访问服务器：10 个异步接口，1 个以太接口。
- 两台 Cisco 3810 或 3600：1 个以太接口，1 个串行接口，1 个带有电话机的 FXS 接口。
- 一台实验路由器：1 个令牌环接口，1 个串行接口，1 个 ISDN BRI 接口。
- 一台实验路由器：1 个以太接口，1 个令牌环接口，1 个串行接口，1 个 ISDN BRI 接口。
- 两台实验路由器：1 个以太接口，1 个 ATM 接口。
- 一台实验路由器：1 个 100-MB 以太接口，1 个串行接口。
- 一台 Catalyst 3900 交换机。
- 一台 Catalyst 5000 交换机。
- 一台 ATM 交换机。

18.7.2 实验准备工作：帧中继交换机的配置

按图 18-7 为帧中继交换机配置 PVC，这一部分的实验考试不需计时。例 18-6 是帧中继交换机的配置过程。

例 18-6 配置帧中继交换机

```

hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 121 interface Serial1 120
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 148000
frame-relay intf-type dce
frame-relay route 110 interface Serial5 111
frame-relay route 120 interface Serial0 121
frame-relay route 130 interface Serial3 131
frame-relay route 140 interface Serial2 141
!
interface Serial2
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 141 interface Serial1 140
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 131 interface Serial1 130
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 111 interface Serial1 110
!

```

下面这些实验考试的内容计时，在帧中继交换机和主干路由器配置完成后，可以开始下列实验考试内容。现在则不要进行任何路由器的配置。

18.7.3 计时实验考试部分

实验考试规则

- 严格按照要求进行设计和配置，何时何地转发路由，PVC 的使用都要按要求进行。
- 前一部分配置内容在第 2 部分可能需要改动，进行第 2 部分的配置之前一定要先完成前面部分的配置工作。

帧中继配置

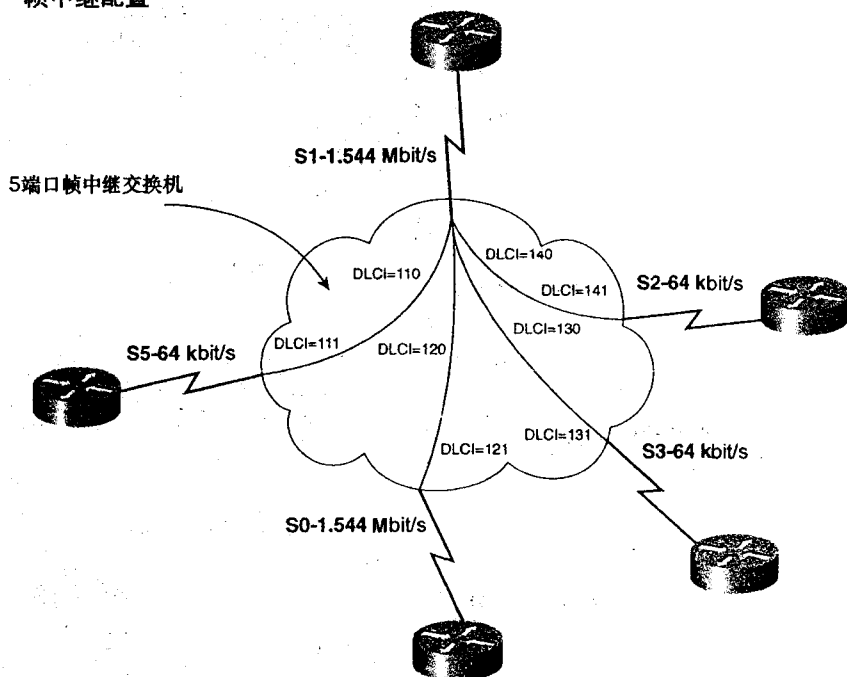


图 18-7 帧中继交换机的配置

- 实验考试时惟一的参考资料只能是所提供的配置指南和 Cisco 文档 CD 盘。
- 实验考试时间一共 8 个半小时。考试进行的过程中不要相互交谈。
- 在开始操作之前，建议先将整个的实验内容通读一遍。
- 要求大家提供正确、清楚的网络说明。
- 参考的实验布局如图 18-8 所示。

第 1 节：基本的 IP 配置

1 访问服务器：配置访问服务器/路由器，要求所有的路由器和交换机都可以通过反向 Telnet 对其进行访问。对所有的路由器和交换机实施密码保护，密码为 cisco。

2 IP 地址的分配：按照图 18-8 为所有的物理接口分配 IP 地址，所有接口上都以 155.100.x.x 为主网。除下列接口以外所有接口都使用 24 位掩码：

R1：路由器 R1 与 R5 之间的 VLAN 1 上采用 25 位的地址。

R2：在 VLAN 20 上采用 27 位的地址。

R3：在 VLAN 30 上采用 28 位的地址。

R4: 该路由器 Ring 2 分配的 IP 地址是 10.11.10.0/24。

3 除非特别指出，所有以太、令牌环和回环接口都需要建立完整的 IP 和 IPX 连接。

4 建立网络的完整文档资料，包括所有的 OSPF 区域、IP/IPX 地址、IPX 网络等等。

CCIE 实验考试模拟练习

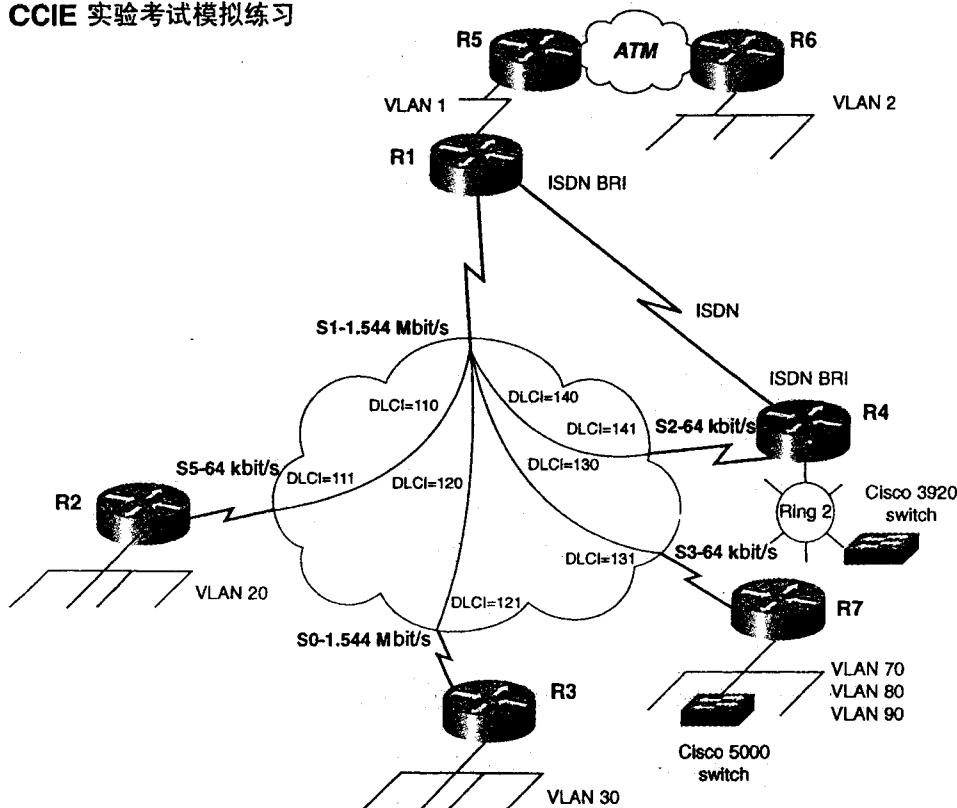


图 18-8 “The Enchilada” 网络拓扑图

第 2 节: Catalyst/LAN 的配置

1 按照图 18-8 对 VLAN 进行配置，VLAN 70，VLAN 80 和 VLAN 90 都包含路由器 R7 的 100-Mbit/s 接口内。配置路由器来对这些 VLAN 进行路由转发。

2 配置 Catalyst 以太交换机和令牌环交换机，使得网络中所有的路由器都可以通过 Telnet 对这些交换机进行管理。

3 这里配置令牌环网段时不要使用默认的令牌环 VLAN。

4 在 Catalyst 交换机的接口 1/1 上配置一条全双工的 802.1Q Trunk，同时禁止通过这条 Trunk 来传输 VLAN 70、80 和 90 的 STP。这条链路是否连接并不重要，交换机的配置都是一样的。

第 3 节: OSPF 和帧中继的配置

1 按照图 18-8 对帧中继网络进行配置。

2 路由器 R1, R2 和 R3 三者共享同一 IP 子网，只有 R1 上才能使用子接口。将这个子网配置在 OSPF 的 Area 0 中，配置 OSPF 的时候不要使用 `ip ospf network` 命令。路由器 R1 和 R7 之间的帧中继线路则处于 OSPF Area 10 中。

3 VLAN 20 配置到 OSPF Area 20 中，VLAN 30 配置到 OSPF Area 30 中，而 VLAN 70、80 和 90 则是在 OSPF Area 70 中。

4 配置 Area 30 向那些 VLAN 30 中所有新加入的路由器发送类型 7 的链路状态信息。

5 在 Area 0 实施 Type 2 的认证方式，密码为 cisco。

第 4 节: 路由选择协议与路由重分布的配置

1 在 R1/R4 之间的帧中继线路以及 Ring 2 上配置 IGRP 路由选择协议，同时确保 OSPF 网络对这个网络可以进行完全的访问。

2 禁止路由器 R4 上令牌环网络的子网转发到 R7 去。

第 5 节: ATM 和 EIGRP 的配置

1 将路由器 R5 和 R6 都与 ATM 交换机相连，在这两台路由器之间的 ATM 连接上配置经典 IP 方式，以便二者可以相互 ping 通对方。

2 在路由器 R5 所在的 VLAN 1 和 R6 所在的 ATM 网络中配置 EIGRP，再把 EIGRP 重分布进 OSPF。这时要确保 IGRP 和 OSPF 区域都可以对此 EIGRP 网络进行访问。

第 6 节: ISDN 的配置

1 对路由器 R1 和 R4 的 ISDN 接口进行配置，仅有 R4 可以向 R1 发起呼叫。在进行下一步之前，一定要保证 R1/R4 的 ISDN 接口都可以 ping 通。

2 将这些 ISDN 接口配置为 IGRP 主干网络的一部分，仅当来自 R1 的 IGRP 路由消失时，ISDN 链路状态才会激活。

第 7 节: VoIP 的配置

1 在路由器 R2 和 R3 之间配置 VoIP，利用 FXS 接口在两台路由器之间创建一个振铃线路。

第 8 节: IPX 的配置

2 在路由器 R1, R2, R3 和 R5 上配置 IPX EIGRP, R4 的 Ring 2 上配置 IPX, R4 和 R1 之间的帧中继链路上则配置 IPX RIP/SAP, 并且保证所有路由器都能接收到所有 IPX 网络信息。

3 在 R5 上配置 IPX SAP 以支持文件服务, 名为 fakefservr, 它采用的是 socket 452, 并且保证所有路由器都可以接收到该 SAP 的信息。

4 在 R2 上应用一个 SAP 过滤列表, 阻止所有以字母 “fake” 开头的 SAP。

5 对 VLAN 30 进行配置, 使得仅当新的服务器上线之后, 才会转发 SAP。

第 9 节: 各种 Cisco IOS 特性的配置

1 VLAN 20 上工作站的 IP 地址是通过 VLAN 1 上的动态主机配置协议 (DHCP) 服务器来获得的。配置路由器 R2 对此进行支持。

2 对路由器 R3 进行配置, 使得 VLAN 30 中的工作站可以动态确定它们默认网关的位置, 但是却不能使用 DHCP。

3 在路由器 R1, R2 和 R5 上配置组播路由方式, 作为组播 224.0.0.7 的一个成员, R1 能够对 R2 和 R5 的 ping 命令进行响应。

4 通过配置路由器 R1, R2 和 R5 实现 Cat5k 交换机用于组播的动态配置。

第 2 部分

第 10 节: BGP 的配置

1 路由器 R2 和 R7 位于 AS 2001 中, R5 位于 AS 5 中, 而 R4 位于 AS 4 中。从 R2 到 R7 配置一个 IBGP 对等体, 而从 R2 到 R5 和 R7 到 R4 则各配置一个 EBGP 对等体。

2 在路由器 R4 上, 设置一个地址为 220.128.1.0/24 的回环接口, 并通过 BGP 进行转发。在路由器 R5 设置地址分别为 24.128.1.0/24, 24.128.2.0/24 的回环接口, 并通过 BGP 进行转发。

3 当且仅当 R5 中含有 220.128.1.0/24 路由时, 通过路由器 R5 向 ATM 网络转发一个默认路由。

4 对路由器 R4 进行配置, 使得所有来自 AS 2001 的路由加权值为 350。

第 11 节: DLSw 的配置

1 从路由器 R1 所在的 VLAN 1 到 R2 所在的 VLAN 20 配置一个 DLSw 的 TCP 类型对等体。在 VLAN 20 中有一台 MAC 地址为 2200.900e.0001 的 SNA 地址, 这个 DLSw 对等体只允许去往这个 SNA 地址的探测数据帧通过。

2 在路由器 R4 所在的 Ring 2 和 R1 所在的 VLAN 1 之间配置一个 DLSw 的 TCP 类型对等体。

3 即使帧中继服务中断, 也要保证路由器 R4 上的对等体处在 “连接” 状态之中。

ISDN 链路上的收敛需要大约 3 分钟，在这段时间里，对等体不可以断开连接或退出工作状态。

第12节：各种 Cisco IOS 特性的配置

1 在从 VLAN 20 到 VLAN 1 的帧中继网络区域上通过透明桥接进行连接。

2 将路由器 R1 配置为一台 NTP 服务器，建立一个对等体使得 R4 的时钟向 R1 进行同步。在 R4 完成同步之后，R2 和 R3 再以 R4 为标准进行时钟的同步。如果 R4 没有和 R1 完成同步，R2 和 R3 不可以进行时钟同步。

3 IPX 数据在 VLAN 30 上可以桥接和路由。

18.8 CCIE 实验考试的模拟练习：“The Unnamed Lab”

18.8.1 设备清单

- 一台帧中继交换机：3 个串行接口。
- 一台访问服务器：8 个异步接口，1 个以太网接口。
- 两台实验路由器：1 个以太网接口，2 个串行接口，1 个 ISDN BRI 接口。
- 两台实验路由器：1 个令牌环接口，1 个以太网接口，1 个串行接口。
- 两台实验路由器：1 个以太网接口，1 个串行接口。
- 两台实验路由器：1 个以太网接口，1 个 ATM 接口。
- 一台 Catalyst 3900 交换机。
- 一台 Catalyst 5000 交换机。

18.8.2 实验准备工作：帧中继交换机的配置

按照图 18-9 为帧中继交换机配置 PVC，这一部分的实验考试不需计时。例 18-1 是帧中继交换机的配置示例。

例 18-7 配置帧中继交换机

```
hostname frame_switch
!
frame-relay switching
!
<<<text omitted>>>
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 148800
frame-relay lmi-type ansi
frame-relay intf-type dce
```


例 18-8 主干路由器的配置

```
hostname r6_backbone_router
!
interface Loopback20
 ip address 192.190.100.1 255.255.255.0
!
interface Loopback21
 ip address 192.190.101.1 255.255.255.0
!
interface Loopback22
 ip address 192.190.102.1 255.255.255.0
!
interface Loopback23
 ip address 160.100.100.1 255.255.255.0
!
interface Loopback24
 ip address 160.100.128.1 255.255.255.0
!
interface Loopback25
 ip address 160.100.129.1 255.255.255.0
!
interface Loopback26
 ip address 160.100.130.1 255.255.255.0
!
interface Ethernet0
 description place on vlan 2 - Backbone 2
 ip address 133.7.77.254 255.255.255.0
 media-type 10BaseT
!
interface Ethernet1
 description place on vlan 10 - backbone 1
 ip address 160.100.2.254 255.255.255.0
 media-type 10BaseT
!
router rip
 no auto-summary
 network 160.100.0.0
 network 192.190.100.0
 network 192.190.101.0
 network 192.190.102.0
!
router bgp 2001
 no synchronization
 network 160.100.100.0 mask 255.255.255.0
 network 160.100.128.0 mask 255.255.255.0
 network 160.100.129.0 mask 255.255.255.0
 network 160.100.130.0 mask 255.255.255.0
 neighbor 160.100.2.1 remote-as 2010
 neighbor 160.100.2.1 ebgp-multihop 10
!
```

下面这些实验考试内容计时，在帧中继交换机和主干路由器的配置完成之后，就可以开始进行下列实验考试内容了。

18.8.4 计时实验考试部分

实验考试规则

- 除非专门指定，不要使用静态路由或浮动静态路由。
- 严格按照要求进行设计和配置，何时何地转发路由，PVC 的使用都要按要求进行。
- 前一部分配置内容在第2部分可能需要改动，进行第2部分的配置之前一定要先完成前面部分的配置工作。
- 实验考试时惟一的参考资料只能是所提供的配置指南和 Cisco 文档 CD 盘。

CCIE 实验考试模拟练习

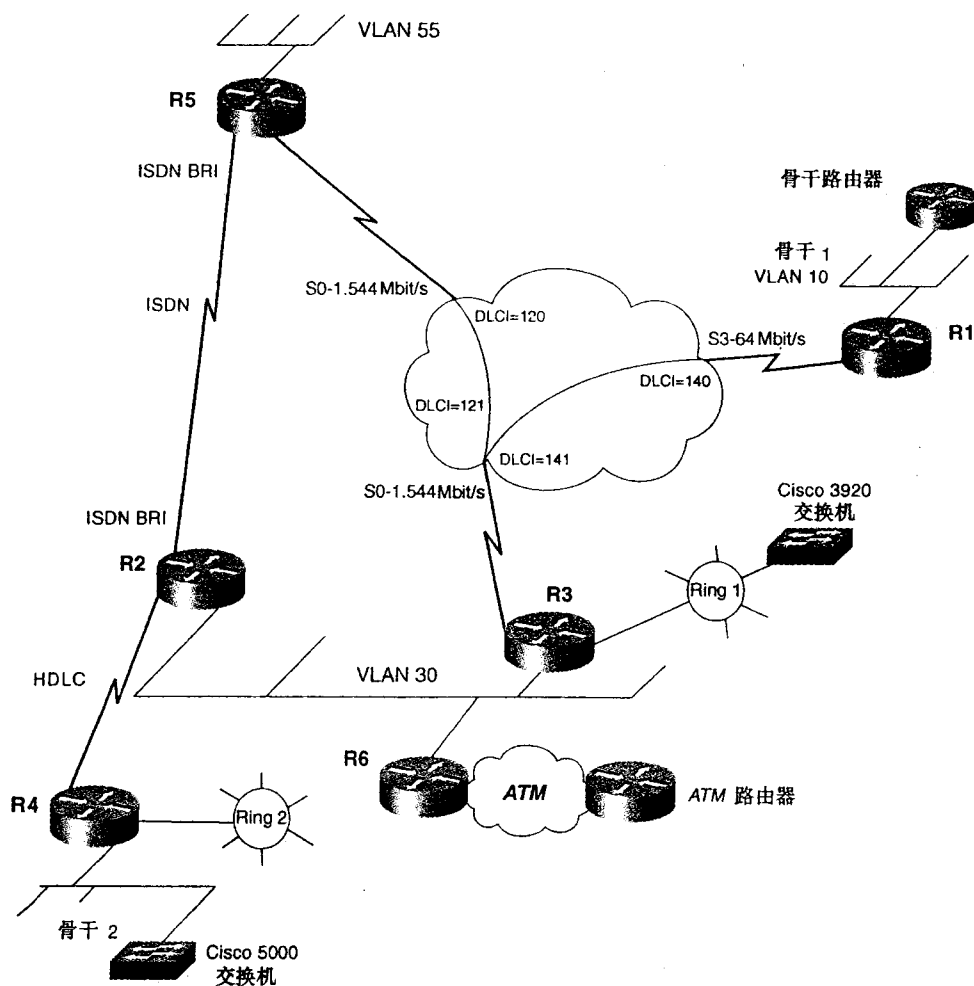


图 18-10 “The Unnamed lab” 网络拓扑图

- 实验考试时间一共 8 个半小时。考试进行的过程中不要相互交谈。
- 在开始操作之前，建议先将整个的实验内容通读一遍。
- 要求大家提供正确、清楚的网络说明。
- 参考的实验布局如图 18-10 所示。

第 1 节：基本 IP 的配置

1 访问服务器：配置对访问服务器/路由器，要求所有的路由器和交换机都可以通过反向 Telnet 对其进行访问。对所有的路由器和交换机实施密码保护，密码为 cisco。

2 IP 地址的分配：按照图 18-10 为所有的物理接口分配 IP 地址，所有接口上都以 133.7.x.x 为其主网。在每台路由器上都配置一个回环接口，地址都使用 133.7.x.x，x.x 就是路由器的编号。除下列接口以外所有接口都使用 24 位掩码：

R1：该路由器在以太网段（VLAN 10）上接口的 IP 地址为 160.100.2.1/24。

R2、R3、R6：这些路由器在 VLAN 30 上的接口采用 22 位的掩码。

R3：该路由器在 Ring 1 上的接口地址掩码为 26 位。

R4：该路由器在 Ring 2 上的接口地址为 10.10.10.1，这个 Ring 2 上的子网可以含有 14 个主机。该路由器在 VLAN 2 中的 IP 地址为 133.7.77.1/24。

R5：该路由器在 VLAN 55 中接口的 IP 地址为 26 位。

3 除非特别指出，所有以太、令牌环和回环接口都需要建立完整的 IP 和 IPX 连接。

4 建立网络的完整文档资料，包括所有的 OSPF 区域、IP/IPX 地址、IPX 网络等等。

第 2 节：Catalyst/LAN 的配置

1 按照图 18-10 对 VLAN 进行配置，Catalyst 交换机的 IP 地址为 133.7.10.254。对 Catalyst 交换机进行配置，使得网络中所有的路由器都可以对此交换机进行管理或者是 Telnet 访问。

2 对 VLAN 30 中的 Catalyst 接口进行配置，使得这些接口只能连接路由器 R2、R3 和 R6。如果其他设备接入了这些接口，Catalyst 交换机会使这些设备接口进入非激活状态。

3 配置 Catalyst 交换机，如果交换机的背板链路负载过重，VLAN 30 上接口的应用优先级高于 VLAN 55 中的接口。

4 对 VTP 实施密码保护，密码为 Cisco_CCIE。

第 3 节：OSPF 和帧中继的配置

1 按照图 18-10 对帧中继网络进行配置，PVC 的应用应严格按照图中定义使用。从路由器 R1 到 R5 不能使用任何动态 PVC 映射。

2 路由器 R1、R3 和 R5 应该是共享同一个处于 Area 10 中的 IP 子网。

3 将 VLAN 30 配置在 OSPF Area 0 中，VLAN 55 则是在 OSPF Area 55 中。

4 将路由器 R3 的 Ring 1 配置在 Area 20 中，R5 与 R2 之间的 ISDN 网络处在 Area 100 中。

第 4 节: 路由选择协议与路由重分布的配置

1 在 R2/R4 之间 HDLC 线路上和 Backbone 2 上配置 IGRP 路由选择协议，将 IGRP 重分布进 OSPF 中去。在 Ring 2 中不要使用任何路由选择协议。

2 在不使用路由聚合配置和静态路由的情况下，将一条默认路由发送到 R4 连接的 Backbone 2 上去。R4 路由器上除了 Ring 2 之外的所有接口都需要与整个网络建立完整的 IP 连接。

3 在路由器 R1 的以太网段，也就是 Backbone 1 上配置 RIP 协议。来自主干网络的路由 192.190.100.0/24，192.190.101.0/24 和 192.190.102.0/24 需要以一条网络路由的形式转发到网络的其余部分去。

4 禁止任何实验网络的路由发送进 Backbone 1 网段。

5 除了 Ring 2 网络之外，所有的 OSPF，IGRP 和 RIP 网络区域都需要保持完整的 IP 可互访性。

第 5 节: ATM 和 EIGRP 的配置

1 将路由器 R6 的 ATM 接口配置为 EIGRP 的 AS 2010 的一部分。

2 路由器 R6 可以无需反向 ARP 就能 ping 通 ATM 的实验路由器。

3 在 EIGRP AS 2010 中配置 MD5 认证方式。

第 6 节: ISDN 的配置

1 配置路由器 R2 和 R5 的 ISDN 接口，仅有 R5 可以发起 ISDN 呼叫。在 ISDN 链路上使用 CHAP 认证方式，密码为 cisco11。在进行下一步的操作之前，一定要保证本地 ISDN 接口可以通过 ping 命令进行访问。

2 将这些 ISDN 接口配置为 OSPF 区域的一部分。除非特别指定，该 ISDN 链路只有在路由表发生变化时才会进入激活状态。而且，仅有 TCP 数据才能激活该链路，路由的更新信息不能激活该链路。

第 7 节: IPX 的配置

1 在除了回环接口、ISDN 网络接口、Backbone 2 的接口和 ATM 网络接口之外的所有接口上配置 IPX 协议。

2 在 Ring 2 和 R2/R4 之间的 HDLC 网络上配置 IPX NLSP。在 VLAN 30 上使用 IPX RIP。在帧中继网络、VLAN 55 和 R1 所在的 Backbone 1 上使用 IPX EIGRP。所有路由器都可以通过 IPX 协议访问所有的 IPX 网络。

3 在路由器 R5 上配置 IPX SAP 以支持打印服务，该 SAP 名为 fakepserver，使用的是 socket

451. 同时这个 SAP 的信息可以转发到所有路由器。

4 配置路由器 R2，使得 VLAN 55 的 IPX 网络不会转发到 R4。

第 8 节：各种 Cisco IOS 特性的配置

1 分配给 Ring 2 的 IP 网络不能通过任何路由选择协议进行转发。这个令牌环中的用户可以不使用重分布、静态路由或默认路由而对整个网络进行完全的 IP 访问。

2 拒绝路由器 R1 对 R2 的 ping 命令，但是允许 R2 对 R1 的 ping。

第 2 部分

第 9 节：BGP 的配置

1 路由器 R1, R2 和 R6 是在 AS 2010 中。R4 是在 AS 2020 中。Backbone 1 路由器则是在 AS 2001 中。同时分别从 R1 到 R2 和 R2 到 R6 各配置一个 IBGP 对等体。

2 从 R2 到 R4 配置一个 EBGP 对等体，配置一个从 R1 到 Backbone 1 路由器 AS 2001 中的 160.100.2.254 的 EBGP 对等体。

3 同步 BGP 和 OSPF。

4 在路由器 R4 上通过 BGP 转发属于 10.10.10.0 的网络接口。

5 从主干路由器需要往外转发的子网路由共有 3 条：从 192.190.100.0 到 192.190.102.0。将这些子网路由聚合为一个超网 192.0.0.0/8，转发的时候就只转发这个超网路由，而抑制具体的路由。

第 10 节：DLSw 的配置

1 在路由器 R3 所在的 Ring 1 和 R5 所在的 VLAN 55 之间配置一个 DLSw 的 TCP 类型对等体。再在 R3 和 R1 的 Backbone 1 之间配置一个 TCP 类型的对等体。

2 从 Ring 1 到 Backbone 1 之间的链路上只能传输 NetBIOS 数据。

3 配置路由器 R5 宣告可以本地访问一台名为 unnamed 的服务器。

第 11 节：各种 Cisco IOS 特性的配置

1 配置路由器 R5，使得从 VLAN 55 到 VLAN 30 的 IP ping 数据包的传输会通过 ISDN 链路进行，而其他数据则是通过帧中继线路来传输。

2 配置路由器 R4，使 Ring 2 只允许通过发送到 MAC 地址为 3745.0001.0001 的 SNA 数据包。

3 配置路由器 R1，使得用户 unnamed 登录到该路由器之后立即进入特权模式中。

4 配置访问服务器，使得它与网络中所有路由器的反向 Telnet 会话永远不会超时。

第 9 部分

附 录

附录 A ISDN 交换机类型、原因代码以及原因代码值

附录 B 简化的 OSI 参考模型

附录 C RFC 清单

附录 D 常见的电缆类型以及引脚定义

附录 E 参考书目

附录 A

ISDN 交换机类型、 原因代码以及 原因代码值

这里给出了 ISDN 支持的交换机类型清单，同时还有 ISDN 的原因代码 (cause code)、原因代码值 (cause value)、承载性能值以及过程描述域的值，这些都可以在 ISDN 的 debug 命令输出中显示。

注释 ITU-T 执行以前由国际电报电话组织顾问委员会 (CCITT) 执行的职能。

A.1 交换机类型

表 A-1 列出了 ISDN 接口支持的交换机类型。

表 A-1 支持的 ISDN 交换机类型

标识符	描述
Basic-1tr6	德国 1 TR 6 ISDN 交换机
Basic-5ess	AT&T 基本速率交换机
Basic-dms 100	NT DMS-100 基本速率交换机
Basic-net3	NET3 ISDN 和欧洲 ISDN 交换机 (英国及其他)，也称为 E-DSS1 或 DSS1
Basic-n11	国际 ISDN-1 交换机
Basic-nwnet3	挪威 net3 交换机
Basic-nznet3	新西兰 net3 交换机
Basic-ts013	澳大利亚 TS013 交换机

续表

标 识 符	描 述
None	未定义的交换机
Ntt	日本 NTT ISDN 交换机（仅 ISDN BRI）
Primary-4ess	用于北美的 AT&T 4ESS 交换机类型（仅 ISDN PRI）
Primary-5ess	用于北美的 AT&T 5ESS 交换机类型（仅 ISDN PRI）
Primary-dms100	用于北美的 nt dms-100 交换机类型（仅 ISDN PRI）
Primary-net5	NET5 ISDN PRI 交换机（欧洲）
Primary-ntt	用于日本的 INS-Net1500 交换机（仅 ISDN PRI）
Primary-ts014	澳大利亚 TS014 交换机（仅 ISDN PRI）
Vn2	法国 VN2 ISDN 交换机（仅 ISDN BRI）
Vn3	法国 VN3 ISDN 交换机（仅 ISDN BRI）
Vn4	法国 VN3 ISDN 交换机（仅 ISDN BRI）

A.2 原因代码字段

表 A-2 是 ISDN 的 debug 命令输出显示的原因代码字段的情况，格式如下：

i=0x y1 y2 z1 z2 [a1 a2]

表 A-2 ISDN 原因代码字段

字 段	值 描 述
0x	以下值都为 16 进制
Y1	8-ITU-T 标准编码
Y2	0-用户 1-用于本地用户的专用网络 2-用于本地用户的公有网络 3-传输网络 4-用于远端用户的公用网络 5-用于远端用户的专用网络 7-国际互联网 A-国际互联点之下的网络
Z1	原因值的分类（高位 16 进制数）。见表 A-3 以获得对该值的更详细信息。
Z2	原因值的分类（低位的 16 进制数）。见表 A-3 以获得对该值的更详细信息。
a1	（可选）总是为 8 的诊断字段
a2	（可选）为下列值的诊断字段 0-未知 1-永远 2-暂时

下面是 **debug isdn q931** 命令的一个输出结果的例子：

Cause i = 0x8790

A.3 原因代码值

表 A-3 是原因代码值的含义。后面的注释是针对表中的诊断 (Diagnostics) 一栏的内容。对于 **debug isdn q931** 命令的输出结果在对照这个表格之前，先要把结果的最高位给去掉。例如，如果其值为 0x90，那就取其为 0x10。

表 A-3 ISDN 原因代码值

十进制值	十六进制值	原 因	诊 断	解 释
1	01	未分配数字	Note10	ISDN 号码以正确的格式发送给交换机，但该号码不会被分配给任何目的设备
2	02	没有到指定的传输网络的路由	传输网络标识 (note 9)	ISDN 交换被用于通过一个未知的中间网络进行呼叫路由
3	03	没有到达目的地的路由	Note10	呼叫通过不用于目的地址的中间网络被路由
6	06	通道不可接受		指定通道的服务质量不足以接受连接
7	07	呼叫被检测到并通过已建立的通道传递		用户被分配一个进入的呼叫，该呼叫连接到已建立连接的呼叫通道
16	10	正常呼叫清除	Note 10	正常呼叫清除已发生
17	11	用户忙		被叫系统已对连接请求作出应答，但是因为 B 信道被占用，所以不能接收该呼叫
18	12	无用户应答		因为目的对呼叫无应答而无法完成连接
19	13	没有来自用户的回答 (用户忽略)		目的地应答连接请求，但不能在预定时间内完成连接。问题出在连接的远端用户
21	15	呼叫被拒绝	Note10: 用户应用诊断 (note4)	目的地可以接受呼叫，但因为某个未知原因而拒绝呼叫
22	16	号码改变		ISDN 号码用于建立未分配给任何系统的呼叫
26	1A	未选择的用户清除		目的地可接受呼叫，但因为未分配给用户而拒绝呼叫
27	1B	目的地出错		因为接口工作不正常和信令消息无法传输而导致目的地不可达。这种情况可能是暂时的，但也可能会持续一段时间。例如，远端设备被关闭。
28	1C	无效的数字格式		因为目的地址格式无法识别或目的地址不完整而使得连接无法建立
29	1D	设备拒绝	设备标识 (note1)	网络无法提供用户要求的设备
30	1E	对 STATUS ENQUIRY 的响应		状态信息是对收到的前一个状态查询的响应
31	1F	普通，未指定		报告无标准原因时的正常事件的发生。无行动要求
34	22	无有效电路、通道		因为没有正确的通道接受呼叫而使得连接无法建立
38	26	网络出错		因为网络工作不正常而导致目的地无法到达，该情况可能会延续一段时间。快速尝试的重连接可能会不成功
41	29	暂时失败		因为网络工作不正常而出现的错误。该问题可以很快解决
42	2A	交换设备拥塞		因为网络交换设备暂时过载而导致目的地不可达
43	2B	访问信息丢失	丢失视信息因素标识 (note5)	网络无法提供所需的访问信息

续表

十进制值	十六进制值	原因	诊断	解释
44	2C	要求的电路/通道不存在		远端设备因为未知原因不能提供所需的通道。这可能是暂时性的问题
47	2F	资源不存在，未指定		要求的通道或者服务因为未知原因而不存在。这可能是暂时性的问题
49	31	服务质量不存在	表 A-2	网络不能提供要求的服务质量。这可能是预定的问题
50	32	没有预定要求的工具	工具标识 (note1)	远端设备只通过预定支持要求的设备
57	39	承载性能未授权	Note3	用户要求网络提供某一承载性能，但用户未被授权使用。这可能是预定的问题
58	3A	承载性能当前不存在	Note3	网络正常提供要求的承载性能，但是当前不存在。这可能是暂时的网络问题或预定问题
63	3F	服务或选项不存在，未指定		网络或远端设备因为未指定的原因不能提供要求的服务选项。这可能是一个预定问题
65	41	承载性能未实施	Note3	网络不能提供用户要求的承载性能。
66	42	通道信号未实施	通道型号 (note 6)	网络或目的设备不支持要求的通道型号
69	45	要求的工具未实施	工具标识 (note 1)	远端设备不支持要求的设备
70	46	仅受限的数字信息承载性能存在		网络不能提高未受限的数字信息承载性能
79	4F	服务或者选项未实施，未指定		网络或者远端设备因为某个未指明的原因不能提供要求的服务选项。这可能是一个预定问题
81	51	无效的呼叫参考值		远端设备收到带有呼叫参考的呼叫，但该呼叫在用户网络接口上未使用
82	52	带标识的通道不存在	通道标识	接收设备被要求使用接口上未激活的通道
83	53	暂停的呼叫仍然存在，但该呼叫身份不存在		网络收到一个呼叫继续的请求。该呼叫继续请求包含一个呼叫身份信息标识，用来标识一个暂停的呼叫
84	54	呼叫身份使用中		网络收到一个呼叫继续的请求。该呼叫继续请求包含一个呼叫身份信息标识，用来标识一个暂停的呼叫
85	55	无暂停的呼叫		无暂停呼叫时网络收到一个呼叫继续请求。这可能是一个暂时的错误，当后续呼叫再次尝试时将得到纠正
86	56	呼叫拥有的要求的呼叫身份已经被清除	清除原因	网络收到一个呼叫继续的请求。该呼叫继续请求包含一个呼叫身份信息标识，该标识曾用来标识一个暂停的呼叫。但是暂停的呼叫通过超时或者远端用户被清除
88	58	不兼容的目的地	不兼容参数 (note 2)	这意味着在进行连接非 ISDN 设备的尝试，例如，连接一个模拟线路
91	5B	无效传输网络选择		ISDN 交换被用于通过一个无法识别的中间网络进行呼叫路由
95	5F	无效消息，未指定		收到无效消息，未应用标准原因。这主要是由 D 信道错误造成的。如果该错误系统性出现，应向 ISDN 服务提供商报告
96	60	必需信息组件丢失	信息组件标识 (note5)	接收设备收到未包括必需信息组件的消息。这一般是 D 信道出错的结果。如果该错误系统性出现，应向 ISDN 服务提供商报告
97	61	消息类型不存在或未应用	消息类型	接收设备收到不可识别的消息，或者因为消息类型无效，或者因为消息类型不支持。原因可能是远端配置问题或者本地 D 信道问题
98	62	消息和呼叫状态不兼容，或者消息类型不存在或未实施	消息类型	远端设备收到一个无效的消息，且未应用标准原因。原因是 D 信道错误。如果该错误系统性出现，应向 ISDN 服务提供商报告

续表

十进制值	十六进制值	原 因	诊 断	解 释
99	63	信息组件不存在或者未实施	信息组件标识 (note5 和 note7)	远端设备收到包括不可识别的信息组件的消息。这通常是 D 信道错误的结果。如果该错误系统性出现，应向 ISDN 服务提供商报告
100	64	无效信息组件内容	信息组件标识 (note 5)	远端设备收到包括无效信息组件的消息。这通常是 D 信道错误的结果
101	65	消息和呼叫状态不兼容	消息类型	远端设备收到一个和当前连接状态不对应的非预期的消息。这通常是 D 信道错误的结果
111	6F	协议错误，未指定		当未用其他标准原因时，通常是未指定的 D 信道错误的结果
127	7F	互连网络，未指定		事件发生，但网络不提供采取该行为的原因。确切问题未知

注释 1——facility identification 的代码值与网络有关。

注释 2——参数 incompatible parameter 由 incompatible information element identifier 构成。

注释 3——39, 3A 和 41 的 diagnostic 字段可以参考 ITU-T Q.850 规范的表 3b/Q.850。

注释 4——用户提供的 diagnostic 字段是根据用户的规范进行编码的，可能会有很长的原因信息。这类信息必须保证不与表 A-2 中的内容相矛盾。

注释 5——ITU-T Q.931 规范中的锁定与非锁定更替过程在这里仍然适用。大体上，information element identifiers 和接收到的 information elements 的顺序是一样。

注释 6——编码方式是这样的：

- Bit 8——扩展位。
- Bits 7 到 5——备用预留。
- Bits 4 到 1——根据表 4-15/Q.931 octet 3.2，代表 ITU-T Q.931 规范中规定的通道类型。

注释 7——如果只有锁定更替信息而没有长度可变的 information element identifier，说明没有采用锁定更替本身的代码集。

注释 8——timer number 是编码成 IA5 字符。每个字节都是采用下面的编码方式的：

- Bit 8——保留“0”。
- Bits 7 到 1——IA5 字符

注释 9——diagnostic 字段包括整个的传输网络选择或者是有效的特定网络设施的 information element。

注释 10——所用编码方式请参考表 A-2。

A.4 承载能力值

表 A-4 是 ISDN debug 命令可以显示的承载能力值，其格式如下：

0x8890 for 64 kbps or 0x8890218F for 56 kbps

表 A-4

ISDN 的承载能力值

字 段 值	描 述
0x	表示以下为 16 进制值
88	ITU-T 编码标准：未受限的数字信息

续表

字 段 值	描 述
90	电路模式，64kbit/s
21	第一层，V.110/X.30
8F	同步，无带内协商，56kbit/s

A.5 “处理”（Progress）字段的值

表 A-5 是 ISDN 的 Progress 指标信息元中包含的“Progress description”字段的情况。

表 A-5 ISDN 的 Progress Description 字段的值

比 特	十进制数字	描 述
0000001	1	非端对端 ISDN 呼叫；后续的呼叫进程信息可能在带内
000010	2	目的地址非 ISDN
0000011	3	起始地址非 ISDN
0000100	4	呼叫返回到 ISDN
0001000	8	当前存在带内信息或者正确的模式
0001010	10	应答延迟发生在目的接口上

该字段任何其他值都保留未用

附录 B

简化的 OSI 参考模型

几乎所有与网络相关的书籍都会提及 OSI 参考模型，本书也不例外。但是，我不是想在这里重复大家可能已经不只 50 次读到过的东西，表 B-1 是关于该模型的一种新的说法。作为“简化”模型，每一层的描述语句都不超过 10 个字。如果大家想要参考一下“非简化”的 OSI 模型，可以参阅 Radia Perlman 著的《*Interconnections, Second Edition: Bridges, Routers, Switches, and Internetworking Protocols*》和 William Stallings 著的《*Networking Standards a Guide to OSI ISDN LAN, and MAN*》。

表 B-1 简化的 OSI 参考模型

OSI 层	简单描述
应用层	控制端用户应用和服务：FTP，WWW 浏览器，Telnet，SMTP
表达层	处理数据格式，编码和传输：GIF，JPEG，ASCII，EBCDIC，HTML
会话层	处理应用会话控制，数据成组和恢复，NFS，SQL，NetBIOS
传输层	管理网络连接，可靠（TCP 和 SPX）和不可靠（UDP）传输
网络层	编址和路由数据包，包分段：IP，IPX，AppleTalk
数据链路层	控制物理层数据流，帧中继，ATM，PPP，IEEE802.x
物理层	定义电器和物理规范，EIA/TIA-232，V.35，10BASEx，B8ZS，NRZ

附录 C

RFC 清单

表 C-1 是本书中遇到的一些常见的 RFC 清单。完整 RFC 可以在网站 www.isi.edu 上查到。

表 C-1

RFC 清单

RFC 编号	注 释	RFC 标题
RFC 2125		PPP 带宽分配协议 (BAP) \PPP 带宽分配控制协议 (BACP)
RFC 2037		使用 SMTv2 的实体 MIB
RFC 2030		简单网络时间协议 (SNTP) 用于 IPv4, Ipv6 和 OSI 的版本 4
RFC 2018		TCP 选择性应答选项
RFC 1997		BGP 通信属性
RFC 1994	取代 RFC 1334	PPP 质询握手认证协议 (CHAP)
RFC 1990	取代 RFC 1717	PPP 多链路协议 (MP)
RFC 1989	取代 RFC 1333	PPP 链路质量监测
RFC 1918		专用网络的地址分配
RFC 1907		用于简单网管协议版本 2 的管理信息库
RFC 1906		用于简单网管协议版本 2 的传输映射
RFC 1905		用于简单网管协议版本 2 的协议操作
RFC 1904		用于简单网管协议版本 2 的一致性声明
RFC 1903		用于简单网管协议版本 2 的文本习惯
RFC 1902		用于简单网管协议版本 2 的管理信息结构
RFC 1901	取代 RFC1441-1445	基于通信的 SNMPv2 介绍

续表

RFC 编号	注 释	RFC 标题
RFC 1889		RTP:用于实时应用的传输协议
RFC 1850		OSPF 版本 2 的管理信息库
RFC 1812		IP 版本 4 路由要求
RFC 1795		DLSw: 交换机到交换机协议
RFC 1793		支持按需电路的扩展 OSPF
RFC 1771		边界网关协议 4
RFC 1745		用于 IP 的 BGP4/IDRP-OSPF 互操作
RFC 1724		RIP 版本 2MIB 扩展
RFC 1723		带附加信息的 RIP 版本 2
RFC 1722		RIP 版本 2 协议应用申明
RFC 1717	被 RFC1990 取代	PPP 多链路 (MP)
RFC 1695		使用 SMIV2 的 ATM 管理版本 8.0 的定义
RFC 1661	取代 RFC1548	PPP (点到点协议)
RFC 1647		TN3270 增强
RFC 1646		Cisco 只支持 Luname 选择方式
RFC 1638		PPP 桥接控制协议 (BCP)
RFC 1634	取代 1362 和 1551	在变化的 WAN 介质 (IPXWAN) 上的 Novell IPX
RFC 1633		Internet 构架中的集成服务: 概述
RFC 1631		IP 网络地址转换 (NAT)
RFC 1618		ISDN 上的 PPP
RFC 1604		帧中继服务的管理目标的定义
RFC 1587		OSPF 不完全短截区域 (NSSA) 选项
RFC 1583	取代 RFC1247	OSPF 版本 2
RFC 1577		ATM 上的经典 IP 和 ARP
RFC 1576		TN3270 当前实践
RFC 1559		DECnet PhaseIV MIB 扩展
RFC 1552		PPP Internetwork 包交换控制协议 (IPXCP)
RFC 1549		HDLc 帧中的 PPP
RFC 1548	被 RFC 1661 取代	点到点协议 (PPP)
RFC 1541	被 RFC 1531 取代	动态主机配置协议
RFC 1531	被 RFC 1541 取代	动态主机配置协议
RFC 1519		无类域间路由 (CIDR): 一种地址分配和汇总策略

续表

RFC 编号	注 释	RFC 标题
RFC 1510		Kerberos 网络认证服务 (V5)
RFC 1492		访问控制协议，有时也称为 TACACS
RFC 1483		ATM 适配层 5 上的多协议封装
RFC 1469		令牌环本地区域网络的 IP 多播
RFC 1450	被 RFC 1907 取代	用于 SNMP 版本 2 的 MIB
RFC 1403		BGP OSPF 互操作
RFC 1397		BGP2 中的默认路由宣告和边界网关协议的 BGP3 版本
RFC 1395		BootP 厂商信息扩展
RFC 1393		使用 IP 选项的 traceroute
RFC 1390		FDDI 网络上的 IP 和 ARP 传输
RFC 1382		用于 X.25 数据包层的 SNMP MIB 扩展
RFC 1381		用于 X.25 LAPB 的 SNMP MIB 扩展
RFC 1378		PPP AppleTalk 控制协议 (ATCP)
RFC 1377		PPP OSI 网络层控制协议 (OSINLSP)
RFC 1376		PPP DECnet Phase IV 控制协议 (DNCP)
RFC 1370		OSPF 的应用声明
RFC 1362		多种 WAN 介质上的 Novell IPX (IPXWAN)
RFC 1356		X.25 和 ISDN 的包模式上的多协议互联
RFC 1350		TFTP 协议 (修改号 2)
RFC 1349		Internet 协议簇的服务类型
RFC 1348		DNS NSAP RR
RFC 1334	被 RFC 1994 取代	PPP 认证协议
RFC 1333	被 RFC 1989 取代	PPP 链路质量监控
RFC 1332		PPP Internet 协议控制协议 (IPCP)
RFC 1331	被 RFC 1548 取代	点到点链路中用于多协议数据包传输的点到点协议 (PPP)
RFC 1323		用于高端操作的 TCP 扩展
RFC 1315		用于帧中继 DTE 的管理信息库
RFC 1305		网络时间协议 (版本 3)，规范、应用和分析。
RFC 1294	被 RFC 1940 取代	帧中继上的多协议互操作
RFC 1293		反向地址解析协议
RFC 1286		用于桥接的管理对象的定义
RFC 1285		FDDI 管理信息库

续表

RFC 编号	注 释	RFC 标题
RFC 1269		用于边界网关协议（版本 3）的管理对象的定义
RFC 1268		Internet 上的 BGP 应用
RFC 1267		边界网关协议（BGP3）
RFC 1256		ICMP 路由发现信息
RFC 1253		OSPF 版本 2 管理信息库
RFC 1247	被 RFC 1583 取代	OSPF 版本 2
RFC 1236		用于 DDN 的 IP 到 X.121 地址映射
RFC 1234		IP 网络上通过通道传输 IPX 数据
RFC 1231		IEEE 802.5 令牌 MIB
RFC 1220		用于桥接的点到点协议扩展
RFC 1219		分配子网号码
RFC 1215		定义和 SNMP 一起使用的事件的方式
RFC 1213		用于基于 TCP/IP 网络的网络管理的管理信息库：MIB II
RFC 1212		简明 MIB 定义
RFC 1209		SDMS 服务上的 IP 数据报传输
RFC 1196		Finger 用户信息协议
RFC 1195		TCP/IP 和双重环境中的 OSI IS-IS 路由
RFC 1191		路径 MTU 发现
RFC 1188	被 RFC 1390 取代	用于 FDDI 网络的 IP 数据报传输标准
RFC 1172		点到点（PPP）初始化配置选项
RFC 1171	被 RFC 1331 取代	用于点到点链路上的多协议数据报传输的 PPP
RFC 1166		INTRENET 数字
RFC 1164		Internet 的 BGP 应用
RFC 1163		边界网关协议（BGP）
RFC 1157		简单网络管理协议（SNMP）
RFC 1156	被 RFC 1213 取代	用于 TCP/IP 的 MIB
RFC 1155	被 RFC 1212 取代	用于基于 TCP/IP 的 Internet 的管理信息的结构和定义
RFC 1144		低速串行网络上的 TCP/IP 包头压缩
RFC 1141		Internet 增量更新校验
RFC 1139		ISO8472 的应答功能（ping）
RFC 1136		管理域和路由域：Internet 上的路由模型
RFC 1122		Internet 主机要求—通信层

续表

RFC 编号	注 释	RFC 标题
RFC 1119	废除 RFC119,1059,958	网络时间协议 (NTP) 版本 3
RFC 1112		用于 IP 多播的主机扩展
RFC 1108	DCA 草稿	IP 安全选项
RFC 1101		网络名称和其他类型的 DNS 编码
RFC 1091		Telnet 终端类型选项
RFC 1084		BootP 扩展
RFC 1080		Telnet 远端流量控制选项
RFC 1079		Telnet 终端速率选项
RFC 1069		ISO 无连接模式的网络协议中 Internet IP 地址的使用指南
RFC 1060		分配号码
RFC 1058		路由信息协议 (RIP)
RFC 1055		串行链路上 IP 数据报传输标准
RFC 1042		在 IEEE802 网络上的 IP 数据报传输标准
RFC 1035		域名—应用和规范
RFC 1034		域名—概念和工具
RFC 1027		将 ARP 用于应用子网网关 (ARP 代理)
RFC 1009		Internet 网关要求
RFC 995	被 ISO9542 代替	与 ISO8473 共同使用时的端系统到中间系统的路由交换协议
RFC 994	被 ISO8473 代替	提供无连接模式网络服务的协议
RFC 982		
RFC 951		Bootstrap 协议 (BOOTP)
RFC 950		Internet 标准子网划分过程
RFC 925		多 LAN 地址解析 (代理 ARP)
RFC 922		子网划分过程中的广播数据报
RFC 919		广播 Internet 数据报
RFC 906		使用 TFTP 的 Bootstrap 载入
RFC 904		外部网关协议 (EGP) 正式规范
RFC 903		保留地址解析协议
RFC 896		TCP/IP 互联网中的拥塞控制
RFC 895		试验性以太网上的 IP 数据报的传输标准
RFC 894		以太网上的 IP 数据报的传输标准
RFC 879		Hello 协议员

续表

RFC 编号	注 释	RFC 标题
RFC 877		TCP 最大分段尺寸和相关主题
RFC 874		Telnet 协议规范
RFC 863		丢弃协议
RFC 862		应答协议
RFC 860		Telnet 定时标记选项
RFC 858		Telnet 压制前进选项
RFC 857		Telnet 应答选项
RFC 856		Telnet 二进制传输
RFC 855		Telnet 选项规范
RFC 854	MIL STD1782	Telnet 协议规范
RFC 827		外部网关协议（EGP）
RFC 826		以太地址解析协议（ARP）
RFC 815		IP 数据报重组算法
RFC 813		TCP 中的窗口和应答策略
RFC 793	MIL STD1778	传输控制协议（TCP）
RFC 792		Internet 控制报文协议（ICMP）
RFC 791	MIL STD1777	Internetwork 协议（IP）
RFC 783		简单文件传输协议（TFTP 版本 2）
RFC 779		Telnet 发送定位选项
RFC 768		用户数据报协议（UDP）

附录 D

常见的电缆类型 以及引脚定义

这里给出了引脚定义的电缆包括：

- 控制台端口引脚 (RJ-45)。
- 辅助端口引脚 (RJ-45)。
- EIA-530 DTE 电缆引脚 (DB-60 到 DB-25)。
- EIA/TIA-232 DTE 电缆引脚 (DB-60 到 DB-25)。
- EIA/TIA-232 DCE 电缆引脚 (DB-60 到 DB-25)。
- EIA/TIA-449 DTE 电缆引脚 (DB-60 到 DB-37)。
- EIA/TIA-449 DCE 电缆引脚 (DB-60 到 DB-37)。
- V.35 DTE 电缆引脚 (DB-60 到 34-针)。
- V.35 DCE 电缆引脚 (DB-60 到 34-针)。
- X.21 DTE 电缆引脚 (DB-60 到 DB-15)。
- X.21 DCE 电缆引脚 (DB-60 到 DB-15)。
- 以太 (AUI) 电缆引脚 (DB-15)。
- 令牌环端口引脚 (DB-9)。
- 异步分拆电缆引脚 (8-针 RJ-45)。
- 异步电缆引脚 (68-针 SCSI)。
- RJ-45 到 DB-25 适配器引脚。
- 异步设备电缆。

D.1 控制台端口与辅助端口的信 号与引脚定义

控制台端口用作数据通讯设备 (DCE)，而辅助端口则是

配置用作数据终端设备 (DTE)。控制
器或其他外部通讯设备进行连接，可
置成异步串行端口。

表 D-1 给出了控制台端口的引脚

注释 表 D-16 和表 D-17 是适配

表 D-1

控制

引脚 ¹	
1	
2	
3	
4	
5	
6	
7	
8	

¹ 未提到的引脚无连接。

表 D-2

辅助

引脚 ¹	
1	
2	
3	
4	
5	
6	
7	
8	

¹ 未提到的引脚无连接

辅助端

RJ-45

用的老

3-25 的

RJ-45 适

器来实

器。要与

这两种

解调

都配

，而表

则是系

端口的号

义。

脚定义

引脚定

RJ-45)

控制台端口 (

信号

—

DTR

TxD

GND

GND

RxD

DSR

—

√输出

—

输出

输出

—

—

输入

输入

—

引脚定义

1-45)

端口引脚

信号

RTS

DTR

TXD

GND

GND

RXD

DSR

CTS

√输出

输出

输出

输出

—

—

输入

输入

输入

D.2 串行电 缆的音 卡和 绑定

下面是一些串行电缆部件和引脚的图表，包括 E 30-DCI 和 EIA/T 32。

EIA/TIA-449, V.35, X.21 DTE 和 DCE 等串行电缆。

D.2.1 EIA-530

图 D-1 是 EIA-530 串行电缆部件图，表 D-3 是其引脚定义。

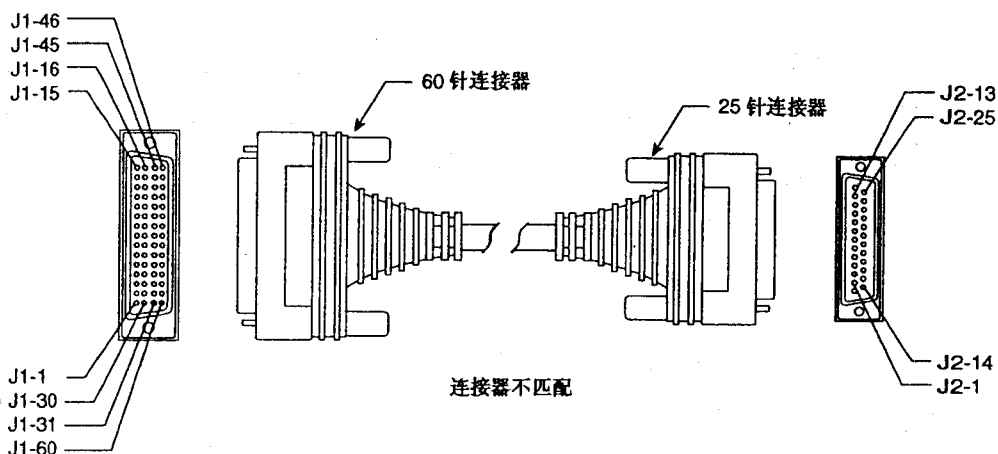


图 D-1 EIA-530 串行电缆的部件装配图

图中的箭头表示信号的方向。→表明是 DTE 到 DCE，而←则是 DCE 到 DTE。

表 D-3 EIA-530 DTE 电缆引脚定义 (DB-60 到 DB-25)

60 引脚 ¹	信 号	25 引脚	信 号	DTE DCE 方向 ²
J1-11	TxD/ RxD+	J2-2	BA (A) ,TxD+	→
J1-12	TxD/ RxD-	J2-14	BA (B) ,TxD-	→
J1-28	RxD/TxD+	J2-3	BB (A) ,RxD+	←
J1-27	RxD/TxD-	J2-16	BB (B) ,RxD-	←
J1-9	RTS/CTS+	J2-4	CA (A) ,RTS+	→
J1-10	RTS/CTS-	J2-19	CA (B) ,RTS-	→
J1-1	CTS/RTS+	J2-5	CB (A) ,CTS+	←
J1-2	CTS/RTS-	J2-13	CB (B) ,CTS-	←
J1-3	DSR/DTR+	J2-6	CC (A) ,DSR+	←
J1-4	DSR/DTR-	J2-22	CC (B) ,DSR-	←
J1-46	Shield_GND	J2-1	屏蔽	短路
J1-47	MODE_2	-	-	
J1-48	GND	-	-	短路

续表

60 引脚 ¹	信 号	25 引脚	信 号	DTE DCE 方向 ²
J1-49	MODE-1	-	-	
J1-5	DCD/DCD+	J2-8	CF (A) ,DCD+	←
J1-6	DCD/DCD-	J2-10	CF (B) ,DCD-	←
J1-24	TxC/RxC+	J2-15	DB (A) ,TxC+	←
J1-23	TxC/RxC-	J2-12	DB (B) ,TxC-	←
J1-26	RxC/TxCE+	J2-17	DD (A) ,RxC+	←
J1-25	RxC/TxCE-	J2-9	DD (B) ,RxC-	←
J1-44	LL/DCD	J2-18	LL	→
J1-45	Circuit_GND	J2-7	Circuit_GND	-
J1-7	DTR/DSR+	J2-20	CD (A) ,DTR+	→
J1-8	DTR/DSR-	J2-23	CD (B) ,DTR-	→
J1-13	TxCE/TxC+	J2-24	DA (A) ,TxCE+	→
J1-14	TxCE/TxC-	J2-11	DA (B) ,TxCE-	→

¹ 未提到的引脚无连接。

² EIA-530 接口在 DCE 模式中不能操作。DCE 电缆不适用于 EIA-530 接口。

D.2.2 EIA/TIA-232

图 D-2 是 EIA/TIA-232 电缆部件装配图，而表 D-4 是其 DTE 引脚定义，表 D-5 则是其 DCE 引脚定义。箭头指明信号的方向。→表明是 DTE 到 DCE，而←则是 DCE 到 DTE。

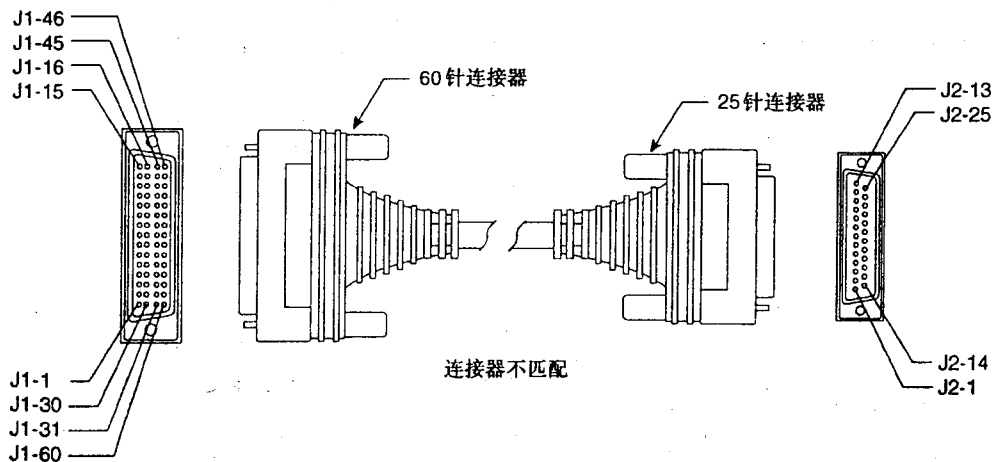


图 D-2 EIA/TIA-232 电缆部件装配图

		D 常		电缆类型	引脚	953	
		EIA	32 DTE	引脚定	3-60 到	j)	
		信 号		送	向	25 引脚	信 号
1		MODB					
		GND					
		MODB D					
2		屏蔽 GND			—	J2-1	屏蔽 GND
		RxD/RxT	第 1 号	绞线		J2-2	TXD
		RxD/TxT	第 2 号	绞线	←	J2-3	RxD
3		—			—	屏蔽	—
		RxD/CTS	第 3 号	绞线		J2-4	RxTS
		CTS/RT	第 4 号	绞线	←	J2-5	CTS
4		—			—	屏蔽	—
		DSR/DTR	第 5 号	绞线		J2-6	DSR
		电路 GND	第 6 号	绞线	—	J2-7	电路 GND
5		—			—	屏蔽	—
		LCR/LC	第 7 号	绞线		J2-8	LCR
		TxC/Nt	第 8 号	绞线	←	J2-15	TxC
6		—			—	屏蔽	—
		RxC/TxC	第 9 号	绞线	←	J2-17	RxC
		—			—	屏蔽	—
7		LL/DC	第 10 号	绞线	→	J2-18	LTST
		—			—	屏蔽	—
8		DTR/D ^s	第 11 号	绞线	—	J2-20	DTR
		—			—	屏蔽	—
9		TxCE/T	第 12 号	绞线	→	J2-24	TxCE
		—			—	屏蔽	—

表 D-5

EIA/TIA-232 DCE 电缆引脚定义 (DB-60 到 DB-25)

60 引脚 ¹	信 号	描 述	方 向	25 引脚	信 号
J1-50	NC/GND	屏蔽	—		
J1-51	GND	屏蔽	—		
J1-46	屏蔽 GND	单线	—	J2-1	屏蔽 GND
J1-35	RxD/TxD	第 9 对双绞线	—	J2-2	TxD
屏蔽	—		—	屏蔽	—
J1-41	TxD/RxD	第 5 对双绞线	→	J2-3	RxD
屏蔽	—		—	屏蔽	—
J1-3	GND/RTS	第 10 对双绞线	—	J2-4	CTS
屏蔽	—		—	屏蔽	—
J1-42	RTS/CTS	第 4 对双绞线	→	J2-5	CTS
屏蔽	—		—	屏蔽	—
J1-33	DTR/DSR	第 3 对双绞线	→	J2-6	DSR
屏蔽	—		—	屏蔽	—
J1-45	电路 GND	第 1 对双绞线	—	J2-7	电路 GND
屏蔽	—		—	屏蔽	—
J1-34	RTS/CTS	第 2 对双绞线	→	J2-8	CTS
屏蔽	—		—	屏蔽	—
J1-39	TxCe/TxC	第 7 对双绞线	→	J2-15	TxC
屏蔽	—		—	屏蔽	—
J1-40	NIL/RxC	第 6 对双绞线	→	J2-17	RxC
屏蔽	—		—	屏蔽	—
J1-33	DCD/LL	第 12 对双绞线	←	J2-18	LTST
屏蔽	—		—	屏蔽	—
J1-34	DSR/DTR	第 11 对双绞线	←	J2-20	DTR
屏蔽	—		—	屏蔽	—
J1-38	RxC/TxCe	第 8 对双绞线	←	J2-24	TxCe
屏蔽	—		—	屏蔽	—

¹ 未提到的引脚无连接。

D.2.3 EIA/TIA-449

图 D-3 给出 EIA/TIA-449 电缆部件装配图，表 D-6l 是其 DTE 的引脚定义，表 D-7 则是其 DCE 引脚定义。箭头指明信号的方向。→表明是 DTE 到 DCE，而←则是 DCE 到 DTE。

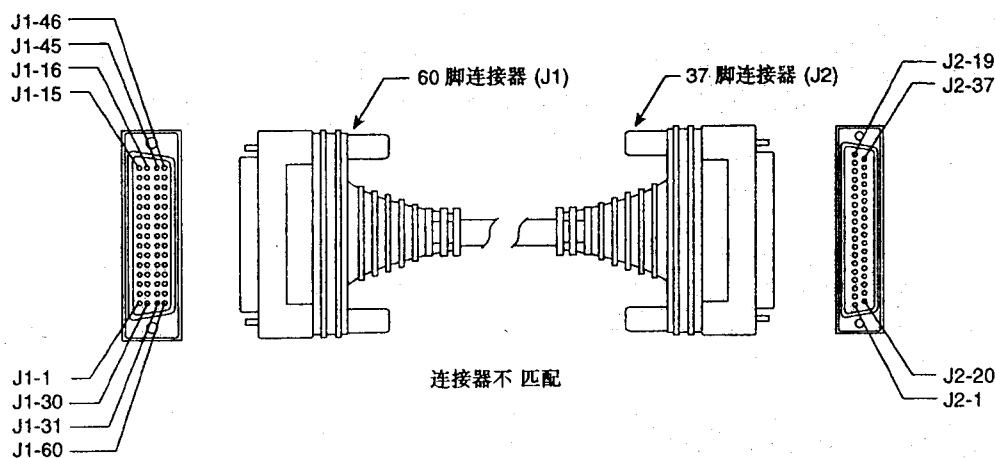


图 D-3 EIA/TIA-449 电缆部件装配图

表 D-6 EIA/TIA-449 的 DTE 电缆引脚定义 (DB-60 到 DB-37)

60 引脚 ¹	信 号	描 述	方 向	37 引脚	信 号
J1-61	MODE_C	简化组	—	—	—
J1-51	GND	简化组	—	—	—
J1-52	MODE_DCE	简化组	—	—	—
J1-53	屏 GND	屏 GND	—	J2-1	屏蔽 GND
J1-11	TxD/RxD+	第 6 对双绞线	→	J2-4	SD+
J1-12	TxD/RxD-		→	J2-22	SD-
J1-24	TxC/RxC+	第 9 对双绞线	→	J2-5	ST+
J1-25	TxC/RxC-		→	J2-23	ST-
J1-28	RxD/TxD+	第 11 对双绞线	←	J2-6	RD+
J1-27	RxD/TxD-		←	J2-24	RD-
J1-9	RTS/CTS+	第 5 对双绞线	→	J2-7	RS+
J1-10	RTS/CTS-		→	J2-25	RS-
J1-26	RxC/TxC+	第 10 对双绞线	←	J2-8	RT+
J1-25	RxC/TxC-		←	J2-26	RT-

续表

60 引脚 ¹	信 号	描 述	方 向	37 引脚	信 号
J1-1	CTS/RTS+	第 1 对双绞线	→	J2-9	RS+
J1-2	CTS/RTS-		→	J2-25	RS-
J1-44	LL/DCD	第 12 对双绞线	→	J2-10	LL
J1-45	Circuit_GND		—	J2-37	SC
J1-3	DSR/DTR+	第 2 对双绞线	→	J2-11	DM+
J1-4	DSR/DTR-		→	J2-29	DM-
J1-7	DTR/DSR+	第 4 对双绞线	→	J2-12	TR+
J1-8	DTR/DSR-		→	J2-30	TR-
J1-5	DCD/DCD+	第 3 对双绞线	→	J2-13	RT+
J1-6				J2-14	RT-
J1-13	TxCE/TxC+		→	J2-17	TT+
J1-14	TxCE/TxC-	第 7 对双绞线	→	J2-35	TT-
J1-15	Circuit_GND	第 9 对双绞线	→	J2-19	SG
J1-16	Circuit_GND		→	J2-20	RG

¹ 未提到的引脚无连接。

表 D-7

EIA/TIA-449 的 DCE 电缆引脚定义 (DB-60 到 DB-37)

60 引脚 ¹	信 号	描 述	方 向	37 引脚	信 号
J1-49	MODEM	简化组			
J1-48	GND	简化组			
J1-46	屏蔽 GND	单线	—	J2-1	屏蔽 GND
J1-28	RxD/TxD+	第 11 对双绞线	→	J2-9	SD+
J1-27	RxD/TxD-		→	J2-23	SD-
J1-13	TxCE/TxC+	第 7 对双绞线	→	J2-5	ST+
J1-14	TxCE/TxC-		→	J2-23	ST-
J1-11	TxD/RxD+	第 6 对双绞线	→	J2-6	RD+
J1-12	TxD/RxD-		→	J2-24	RD-
J1-1	CTS/RTS+	第 1 对双绞线	→	J2-7	RS+
J1-2	CTS/RTS-		→	J2-25	RS-
J1-24	TxC/RxC+	第 9 对双绞线	→	J2-8	RT+
J1-23	TxC/RxC-		→	J2-26	RT-
J1-9	RTS/CTS+	第 5 对双绞线	→	J2-9	CS+

续表

60 引脚 ¹	信 号	描 述	方 向	37 引脚	信 号
J1-10	RTS/CTS-		→	J2-27	CS-
J1-29	NIU/L	第 1 对双绞线	→	J2-10	TT
J1-30	Circuit_GND		→	J2-37	SG
J1-7	DTR/DSR+	第 4 对双绞线	→	J2-11	DM+
J1-8	DTR/DSR-		→	J2-29	DM-
J1-4	DSR/DTR+	第 2 对双绞线	→	J2-12	TR+
J1-5	DSR/DTR-		→	J2-30	TR-
J1-5	DCD/DCD+	第 3 对双绞线	→	J2-13	RR+
J1-6	DCD/DCD-		→	J2-31	RR-
J1-28	R/C/T/GCE+	第 10 对双绞线	→	J2-17	TT+
J1-29	R/C/T/GCE-		→	J2-35	TT-
J1-15	Circuit_GND	第 8 对双绞线	-	J2-19	SG
J1-16	Circuit_GND		-	J2-20	RC

¹ 未提到的引脚无连接。

D.2.4 V.35

图 D-4 是 V.35 的部件装配图，表 D-8 是其 DTE 引脚定义，而表 D-9 则是其 DCE 引脚定义。箭头指明信号的方向。→表明是 DTE 到 DCE，而←则是 DCE 到 DTE。

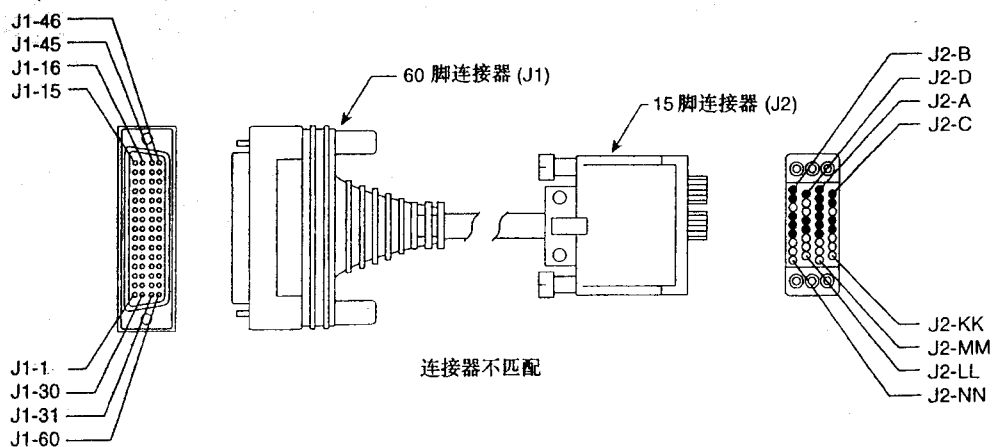


图 D-4 V.35 电缆部件装配图

表 D-8

V.35 的 DTE 电缆引脚定义 (DB-60 到 34 针)

60 引脚 ¹	信 号	描 述	方 向	34 引脚	信 号
J1-49	MODEM	简化组	—	—	—
J1-48	DSR	简化组	—	—	—
J1-50	MODE_0	简化组	—	—	—
J1-51	GND	简化组	—	—	—
J1-52	MODE_DCE	简化组	—	—	—
J1-53	RTS/CTS	第 9 对双绞线	→	J2-C	RTS
J1-54	RTS/CTS	第 9 对双绞线	←	J2-D	CTS
J1-55	RTS/CTS	第 9 对双绞线	→	J2-E	RTS
J1-56	RTS/CTS	第 9 对双绞线	←	J2-F	CTS
J1-57	RTS/CTS	第 9 对双绞线	→	J2-G	RTS
J1-58	RTS/CTS	第 9 对双绞线	←	J2-H	CTS
J1-46	Shield_GND	单线	—	J2-A	Frame GND
J1-47	Shield_GND	单线	—	J2-B	Frame GND
J1-42	RTS/CTS	第 9 对双绞线	→	J2-C	RTS
屏蔽	-	—	—	屏蔽	—
J1-43	RTS/CTS	第 9 对双绞线	←	J2-D	CTS
屏蔽	-	—	—	屏蔽	—
J1-34	DSR/DTR	第 7 对双绞线	→	J2-E	DSR
J1-35	DSR/DTR	第 7 对双绞线	←	J2-F	DTR
屏蔽	-	—	—	屏蔽	—
J1-36	DSR/DTR	第 7 对双绞线	→	J2-G	DSR
J1-37	DSR/DTR	第 7 对双绞线	←	J2-H	DTR
屏蔽	-	—	—	屏蔽	—
J1-38	DSR/DTR	第 7 对双绞线	→	J2-I	DSR
J1-39	DSR/DTR	第 7 对双绞线	←	J2-J	DTR
屏蔽	-	—	—	屏蔽	—
J1-40	DSR/DTR	第 7 对双绞线	→	J2-K	DSR
J1-41	DSR/DTR	第 7 对双绞线	←	J2-L	DTR
屏蔽	-	—	—	屏蔽	—
J18	TxD/RxD+	第 1 对双绞线	→	J2-P	SD+
J17	TxD/RxD-	第 1 对双绞线	→	J2-S	SD-
J1-28	RxD/TxD+	第 5 对双绞线	→	J2-V	SD+
J1-27	RxD/TxD-	第 5 对双绞线	→	J2-W	SD-
J1-20	TxCE/TxC+	第 2 对双绞线	→	J2-U	SCTE+
J1-19	TxCE/TxC-	第 2 对双绞线	→	J2-W	SCTE-
J1-26	RxC/TxC+	第 4 对双绞线	→	J2-Y	SCTE+
J1-25	RxC/TxC-	第 4 对双绞线	→	J2-Z	SCTE-

续表

60 引脚 ¹	信 号	描 述	方 向	34 引脚	信 号
J1-24	TxC/RxC+	第 3 对双绞线	←	J2-Y	SCT+
J1-23	TxC/RxC-		←	J2-AA	SCT-

¹ 未提到的引脚无连接。

表 D-9 V.35 的 DCE 电缆引脚定义 (DB-60 到 34 针)

60 引脚 ¹	信 号	描 述	方 向	34 引脚	信 号
J1-49	MODE_1	简化组			
J1-48	GND	简化组			
J1-50	MODE_0	简化组	—	—	—
J1-51	GND	简化组	—	—	—
J1-52	TC/NHL	简化组			
J1-53	RxC/DCE	简化组			
J1-54	RxD/TxD	简化组			
J1-55	GND	简化组			
J1-46	Shield_GND	单线	—	J2-A	Frame GND
J1-45	Current_GND	第 12 对双绞线	←	J2-B	Frame GND
屏蔽					
J1-35	CTS/RTS	第 8 对双绞线	←	J2-C	RTS
屏蔽	-		—	屏蔽	—
J1-36	RTS/CTS	第 9 对双绞线			
屏蔽					
J1-43	DTR/DSR	第 10 对双绞线	→	J2-E	DSR
屏蔽	-		—	屏蔽	—
J1-44	LL/DCD	第 11 对双绞线	←	J2-F	RLSD
屏蔽			—	屏蔽	—
J1-34	DSR/DTR	第 7 对双绞线	←	J2-H	DTR
屏蔽			—	屏蔽	—
J1-33	DCD/LL	第 6 对双绞线	←	J2-K	RT
屏蔽			—	屏蔽	—
J1-28	RxD/TxD+	第 5 对双绞线	←	J2-P	SD+
J1-27	RxD/TxD-		←	J2-S	SD-
J1-13	TxD/RxD+	第 1 对双绞线	→	J2-R	RD+

续表

60 引脚 ¹	信 号	描 述	方 向	34 引脚	信 号
J1-17	TxD/RxD-			J2-1	RD
J1-26	RxC/TxCe+	第 4 对双绞线	←	J2-U	SCTE+
J1-25	RxC/TxCe-		←	J2-W	SCTE-
J1-22	NIL/RxC+	第 3 对双绞线	←	J2-V	SCR+
J1-21	NIL/RxC-		←	J2-X	SCR-
J1-20	TxCe/TxC+	第 2 对双绞线	→	J2-Y	SCT+
J1-19	TxCe/TxC-		→	J2-AA	SCT-

¹ 未提到的引脚无连接。

D.2.5 X.21

图 D-5 是 X.21 电缆部件装配图，表 D-10 是其 DTE 引脚定义，而表 D-11 则是其 DCE 引脚定义。箭头指明信号的方向。→表明是 DTE 到 DCE，而←则是 DCE 到 DTE。

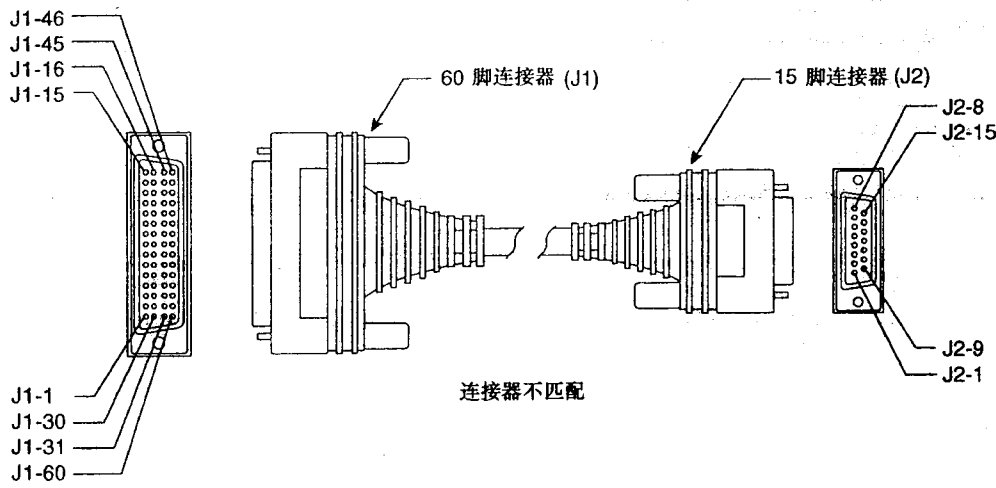


图 D-5 X.21 电缆的部件装配图

表 D-10 X.21 的 DTE 电缆引脚定义 (DB-60 到 DB-15)

60 引脚 ¹	信 号	描 述	方 向	34 引脚	信 号
J1-48	GND	简化组	—	—	—
J1-47	MODE_2	简化组	—	—	—
J1-51	GND	简化组	—	—	—
J1-52	MODE_DCE	简化组	—	—	—
J1-46	Shield_GND	单线	—	J2-1	屏蔽 GND

续表

60 引脚 ¹	信 号	描 述	方 向	34 引脚	信 号
J1-12	TxD/RxD-		→	J2-9	Transmit-
J1-9	CTS/RTS+	第 2 对双绞线	→	J2-10	Control+
J1-10	RTS/CTS-		→	J2-11	Control-
J1-28	RxD/TxD+	第 6 对双绞线	←	J2-4	Receive+
J1-27	RxD/TxD-		←	J2-11	Receive-
J1-11	CTS/RTS-	第 1 对双绞线	←	J2-12	Indication+
J1-12	RTS/CTS+		←	J2-13	Indication-
J1-26	RxC/TxC+	第 5 对双绞线	←	J2-6	Timing+
J1-25	RxC/TxC-		←	J2-13	Timing-
J1-15	Control_GND	第 5 对双绞线	—	J2-8	Control GND

¹ 未提到的引脚无连接。

表 D-11 X.21 的 DCE 电缆引脚定义 (DB-60 到 DB-15)

60 引脚 ¹	信 号	描 述	方 向	15 引脚	信 号
J1-15	Control_GND	第 5 对双绞线	—	J2-8	Control GND
J1-27	RxD/TxD-	第 6 对双绞线	←	J2-4	Receive-
J1-46	Shield_GND	单个	—	J2-1	屏蔽 GND
J1-11	CTS/RTS-	第 1 对双绞线	←	J2-3	Control+
J1-2	CTS/RTS+		←	J2-10	Control-
J1-11	TxD/RxD-	第 1 对双绞线	←	J2-11	Receive-
J1-12	TxD/RxD+		←	J2-12	Receive+
J1-9	RTS/CTS+	第 2 对双绞线	→	J2-5	Indication+
J1-10	RTS/CTS-		→	J2-12	Indication-
J1-24	TxC/RxC+	第 4 对双绞线	←	J2-6	Timing+
J1-23	TxC/RxC-		←	J2-13	Timing-
J1-15	Control_GND	第 5 对双绞线	—	J2-8	Control GND
屏蔽	—		—	屏蔽	—

D.3 以太电缆的部件与引脚定义

图 D-6 是以太（AUI）电缆部件图，而表 D-12 则是 AUI 电缆引脚定义。

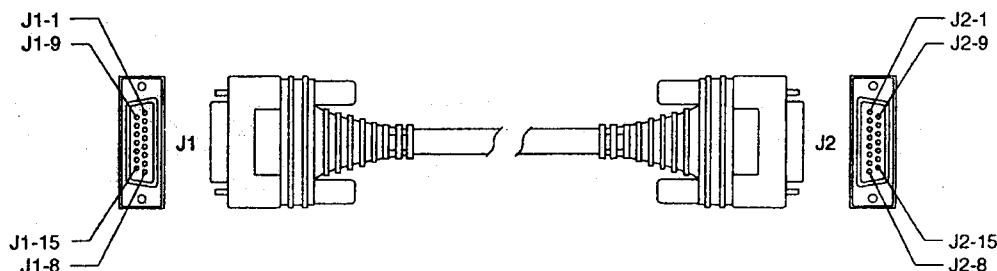


图 D-6 以太（AUI）电缆部件图

表 D-12

以太（AUI）电缆引脚定义（DB-15）

引 脚 ¹	以太电路	信 号
3	DO-A	数据输出电路 A
10	DO-B	数据输出电路 B
11	DO-S	数据输出电路屏蔽
5	DI-A	数据输入电路 A
12	DI-B	数据输入电路 B
4	DI-S	数据输入电路屏蔽
2	CI-A	控制输入电路 A
9	CI-B	控制输入电路 B
1	CI-S	控制输入电路屏蔽
6	VC	一般电压
13	VP	电压输入
14	VS	电压屏蔽（L25 和 M25）
Shell	PG	接地保护

¹ 未提到的引脚无连接。

D.4 令牌环端口引脚定义

表 D-13 是令牌环接口引脚定义情况。

表 D-13

令牌环端口引脚定义 (DB-9)

9个引脚 ¹	信 号
1	接收
3	+5V ²
5	传输
6	接收
9	传输

¹ 2, 4, 7 和 8 脚接地。

² 最大 600 毫安。

D.5 异步串行端口

图 D-7 是 RJ-45 分拆电缆示意图，它有一个 68 针的 SCSI 接口和一个 RJ-45 串行接口。
表 D-14 是 RJ-45 端引脚定义，而表 D-15 则是 68 针的 SCSI 连接器引脚定义。

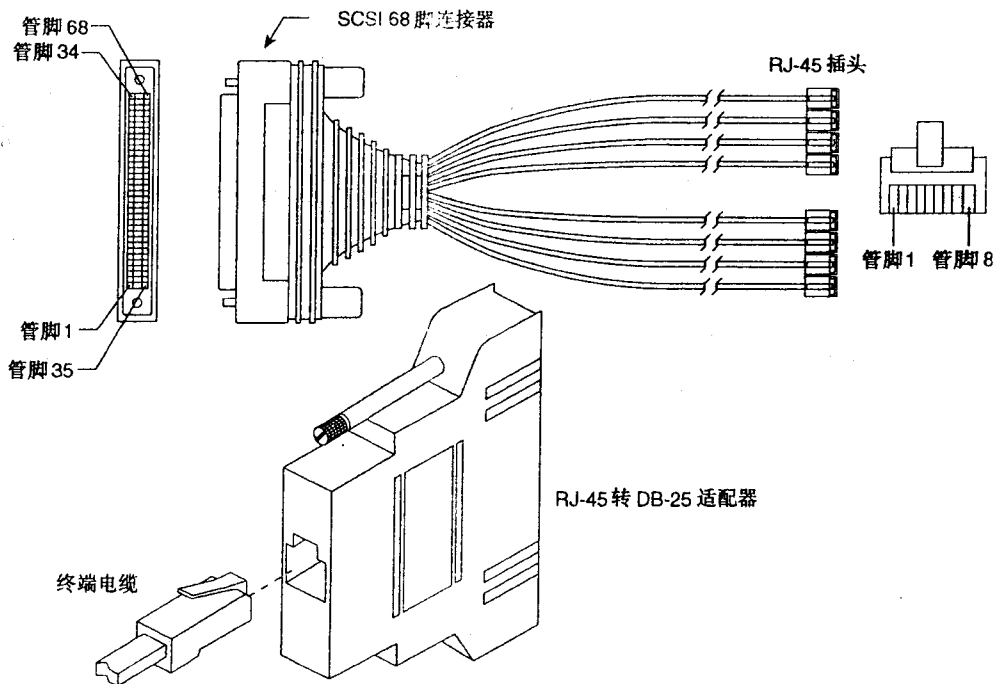


图 D-7 异步串行接口的分拆电缆部件示意图

表 D-14

异步分拆电缆的引脚定义（8 针 RJ-45）

RJ-45 的 8 个引脚	信 号	方 向
1	CTS	←
2	DSR/DCD	←
3	RXD	←
4	RXD/GND	-
5	TXD/GND	-
6	TXD	→
7	DTR	→
8	RTS	→

注释 异步分拆电缆就等于控制台电缆或者是辅助设备电缆再加上反接 RJ-45 电缆。表 D-17 里可以查到异步设备连接方式。

表 D-15

异步电缆引脚定义（68 针 SCSI）

RJ-45 插座	引 脚	信 号	68 引脚的 SCSI (J1)
1	1	CTS	39
	2	DSR	5
	3	RXD	38
	4	RXD GND	4
	5	TXD GND	37
	6	TXD	3
	7	DTR	36
	8	RTS	2
2	1	CTS	43
	2	DSR	9
	3	RXD	42
	4	RXD GND	8
	5	TXD GND	41
	6	TXD	7
	7	DTR	40
	8	RTS	6
3	1	CTS	47
	2	DSR	13
	3	RXD	46
	4	RXD GND	12

续表

RJ-45 插座	引 脚	信 号	68 引脚的 SCSI (J1)
3	5	TXD GND	45
	6	TXD	11
	7	DTR	44
	8	RTS	10
4	1	CTS	51
	2	DSR	17
	3	RXD	50
	4	RXD GND	16
	5	TXD GND	49
	6	TXD	15
	7	DTR	48
	8	RTS	14
5	1	CTS	55
	2	DSR	21
	3	RXD	54
	4	RXD GND	20
	5	TXD GND	53
	6	TXD	19
	7	DTR	52
	8	RTS	18
6	1	CTS	59
	2	DSR	25
	3	RXD	58
	4	RXD GND	24
	5	TXD GND	57
	6	TXD	23
	7	DTR	56
	8	RTS	22
7	1	CTS	63
	2	DSR	29
	3	RXD	62
	4	RXD GND	28
	5	TXD GND	61
	6	TXD	27
	7	DTR	60
	8	RTS	26

续表

RJ-45 插座	引 脚	信 号	68 引脚的 SCSI (J1)
8	1	CTS	67
	2	DSR	33
	3	RXD	66
	4	RXD GND	32
	5	TXD GND	65
	6	TXD	31
	7	DTR	64
	8	RTS	30

D.6 RJ-45 适配器的引脚定义

参考表 D-6 中 RJ-45 到 DB-25 适配器的引脚情况，配一条 RJ-45 电缆，就可以将终端设备和调制解调器与 Cisco 2500 系列的访问服务器相互连接。

可以采用的电缆包括反接电缆或者是直通电缆。

通过比较电缆两端的部件就可以识别全反电缆。把电缆两端并排着拿在手中，让二者的突出部分都朝下，左右两端部件上与最外面一边的引脚相连的导线的颜色应该一样。如果是从 Cisco 购买的电缆，一端上的 pin 1 应该是白色的，而另外一端上的 pin 8 也应该是白色的（全反电缆将 1 和 8，2 和 7，3 和 6，4 和 5 的位置翻转过来了）。（请参考图 D-8）

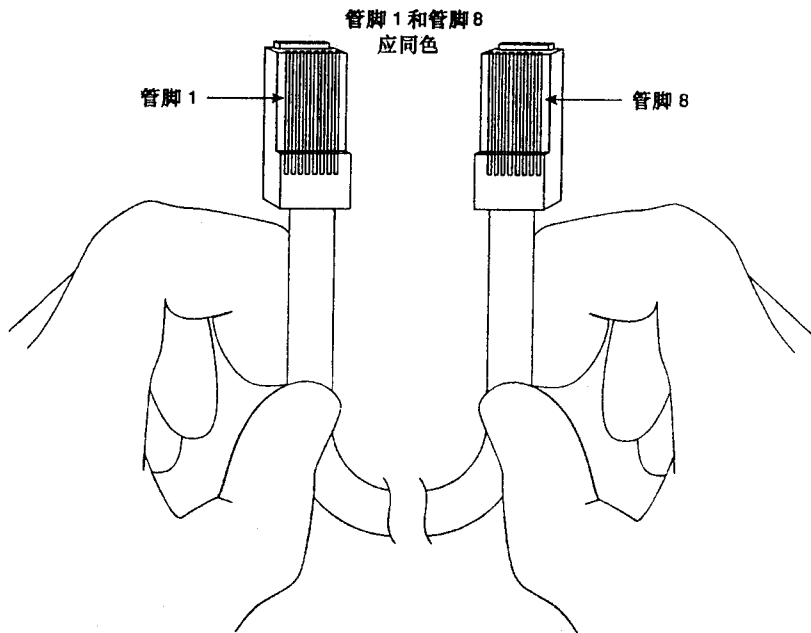


图 D-8 全反电缆的识别

Cisco 2500 系列的访问服务器出厂时配备的都是全反电缆。要与终端设备或调制解调器相互连接，还需要 RJ-45 到 DB-25 适配器，也可以借助于 DB-25 到 DB9 适配器的帮助，可以参考表 D-17 中电缆和适配器的配置来连接终端或调制解调器到 Cisco2500 系列访问服务器。

表 D-16 RJ-45 到 DB-25 适配器的引脚定义

适 配 器	DTE M/F 引脚 ¹	DCE M/F 引脚	MMOD 引脚 ²
RJ-45 引脚	DB-25 引脚		
1	4	5	5
2	20	6	8
3	2	3	3
4	7	7	7
5	7	7	7
6	3	2	2
7	6	20	20
8	5	4	4

¹ Cisco 设备上存在的母数据终端设备 (FDTE) 适配器标识为“终端”。

² Cisco 设备上存在的 MMOD 适配器标识为“modem”。

表 D-17 异步设备的电缆连接搭配

访问服务器端口	RJ-45 线缆类型	DB-25 适配器	端 设 备
控制台或辅助端口	全反线	FDTE ¹	终端
控制台或辅助端口	直连线	FDCE	终端
辅助端口控制台端口	全反线	MMOD ²	Modem ³

¹ FDTE RJ-45-TO-DB-25 适配器标为“终端”。

² MMOD RJ-45-TO-DB-25 适配器标为“modem”。

³ 异步断路线缆（见表 D-14 和表 D-15）在功能上等同于全反线（全反线）。

附录 E

参考书目

以下是在本书的成书过程中用到的参考资料清单。没有这些著者、工程师和专家们的辛勤工作，深入的钻研和无穷的创造，这本书的问世是不可能。

Resource	Title/ Resource Type	Web Page	Chapter	Author
	<i>Configuring IP Routing Protocol- Independent Features</i>		Pages P1C- 189 to 215	
Access Services Configuration Guide, Release 11.1				Cisco
Advanced IP Network Design				Alvaro Retanna, Don Slice, Russ White
Bridging and IBM Networking Command Reference, Cisco IOS Software Release 12.0				Cisco
Bridging and IBM Networking Configuration Guide	<i>Configuring Source-Route Bridging</i>			

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Bridging and IBM Networking Configuration Guide, Cisco IOS Software Release 12.0				
Bridging and IBM Networking Configuration Guide, Release 11.1				Cisco
Catalyst 2900 Series XL and Catalyst 350 Series XL Software Configuration Guide	<i>Configuring VLANs</i>		Chapter 8	
Catalyst 3920 Token Ring Switch User Guide, Release 1.0				
Catalyst 5000 Series Software Configuration Guide, Release 4.2				
Catalyst 6000 Family Software Configuration Guide			Chapters 5, 6, and 11	
Catalyst Token Ring Switching Implementation Guide	<i>Port Operation Modes</i>		Chapter 1	
Cisco: Understanding Service Access Point Access Control Lists	<i>Understanding Service Access Point Access Control Lists</i>			

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Cisco Document 78-2414-02 Rev A0	<i>Update for Catalyst 5000 Series Configuration Guide and Command Reference</i>			
Cisco IOS Desktop Switching Software Configuration Guide	<i>Creating and Maintaining VLANs</i>		Chapter 5	
Cisco IOS Quality of Service Solutions Configuration Guide	<i>Configuring Frame Relay and Frame Relay Traffic Shaping</i>			
Cisco IOS Switching Services Configuration Guide, Cisco IOS Release 12.0				Cisco
Cisco IOS Wide- Area Networking Configuration Guide	<i>Configuring Frame Relay</i>			
Debug Command Reference, Cisco IOS Software Release 12.0				Cisco
Dial Solutions Command Reference, Cisco IOS Software Release 11.03				Cisco
DLSW+ Design and Implementation Guide			Chapters 1, 2, 3, 4, 5, 7, and 9	

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Enabling Enterprise Multihoming with Cisco IOS Network Address Translation (NAT)	Whitepaper, 1997			Akkiraju, Delgadillo, Rekhter
Installation and Maintenance of Cisco Router, Volume 1, version 11.3a	Student Guide			
Installation and Maintenance of Cisco Router, Volume 2, version 11.3a	Student Guide			
Installation and Maintenance of Cisco Router, Volume 3, version 11.3a	Student Guide			
Internetwork Design Guide				Cisco
Internetworking Case Studies				Cisco
Internetworking Technology Overview, June 1999	<i>Enhanced IGRP</i>		Chapter 36	
Internetworking Technology Overview, June 1999	<i>Mixed-Media Bridging</i>		Chapter 24	
Internetworking Technology Overview, June 1999	<i>Token Ring/ IEEE 802.5</i>		Chapter 9	

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Introduction to Cisco Router Configuration: Student Guide, Release 11.0				Cisco
IP Routing Primer				Robert Wright
Network Time Protocol (Version 3) Specification, Implementation, and Analysis				D. Mills
Quality of Service Solutions Configuration Guide	<i>Configuring Frame Relay and Frame Relay Traffic Shaping</i>			
Router Products Configuration and Reference			Chapters 1 and 19	
Router Products Configuration and Reference	<i>Configuring Source-Route Bridging</i>		Chapter 1	
Router Products Configuration and Reference	<i>Configuring Transparent Bridging</i>		Chapter 1	
Router Products Configuration Guide	<i>Configuring DLSw+</i>		Chapter 30	
Simple Network Time Protocol (SNTP), version 4 for IPv4, IPv6 and OSI				D. Mills

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Software Configuration Guide, Release 5.4	<i>Configuring Faster EtherChannel and Gigabit EtherChannel</i>		Chapter 7	
Software Configuration Guide, Release 6.1			Chapters 9 and 12	
Software Configuration Guide, Release 5.2	<i>Configuring Spanning Tree</i>		Chapter 8	
Statement of Direction	<i>10 Gigabit Ethernet Position Statement</i>			
Web Site	"APPN Implementer's Workshop Closed Pages Document"	Info.internet.isi.edu/in-notes/rfc/files/rfc2166.txt		
Web Site	"Avoiding Routing Loops When Using Dynamic NAT"	Cisco.com/warp/public/556/4.html		
Web Site	"Cisco IOS Network Address Translation (NAT)"	Cisco.com/warp/public/701/60.html		
Web Site	"Configuration Notes for the Enhanced Implementation of EIGRP"	www.cisco.com/warp/public/103/12.html		
Web Site	"Configuring IP Enhanced IGRP"	www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgr/np1_c/1cprt1/1ceigrp.htm		
Web Site	"Data Link Switching"	www.cisco.com/warp/public/100/49.html		

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Web Site	"Data Link Switching: Switch-to-Switch Protocol AIW DLSw RIG: DLSw Closed Pages, DLSw Standard Version 1.0"	Info.internet.isi.edu/in-notes/rfc/files/rfc1795.txt		
Web Site	"DLSw and Network Address Translation (NAT)"	Cisco.com/warp/public/697/6.html		
Web Site	"DLSw+ SAP/ MAC Filtering Techniques"	Cisco.com/warp/public/697/dlswfilter.shtml		
Web Site	"Enhanced IGRP Stub Routing"	www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s15/eigrpstb.htm		
Web Site	"Enhanced Interior Gateway Routing Protocol"	www.cisco.com/warp/customer/103/eigrp1.html		
Web Site	"Enhanced Interior Gateway Routing Protocol"	www.cisco.com/warp/customer/103/eigrp5.html		
Web Site	"Enhanced Interior Gateway Routing Protocol"	www.cisco.com/warp/public/customer/103/eigrp6.html		
Web Site	"Frame Relay Traffic Shaping"	www.cisco.com/warp/public/125/21.html		
Web Site	"Introduction to Enhanced IGRP (EIGRP)"	www.cisco.com/warp/public/459/7.html		

(待续)

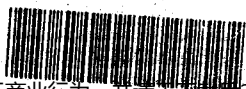
Resource	Title/ Resource Type	Web Page	Chapter	Author
Web Site	"ISDN Switch Types, Codes, and Values"	www.cisco.com/ univercd/dd/td/ doc/product/ software/ios112/ dbook/disdn.html		
Web Site	"NAT Pools and Subnet Zero"	www.cisco.com/ warp/public/556/ 7.html		
Web Site	"NAT: Local and Global Definitions"	www.cisco.com/ warp/public/556/ 8.html		
Web Site	"Password Recovery Techniques"	www.cisco.com/ warp/public/701/ 22.html		
Web Site	"RIF Passthrough in DLSw+ Training Supplement"	www.cisco.com/ warp/public/779/ largeent/sna/trng/ rif_pt/rif_pt.html		
Web Site	"Troubleshooting Token Ring"	www.cisco.com/ univercd/cc/td/doc/ cisintwk/itg_v1/ tr1906.html		
Web Site	"Understanding and Configuring FastEtherChannel on Cisco Switching and Routing Devices"	www.cisco.com/ warp/public/473/ 4.html		
Web Site	"Understanding and Configuring Spanning Tree Protocol (STP) on Catalyst Switches"	www.cisco.com/ warp/public/473/ 5.html		
Web Site	"Using Nonstandard FTP Port Numbers with NAT"	www.cisco.com/ warp/public/556/ 6.html		

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Web Site	"Using the Border Gateway Protocol for Internet Routing"	www.cisco.com/univ-src/3.6/data/doc/cintnet/ics/icsbgp4.html		
Web Site	"Configuring a Gateway Last Resort Using IP"	www.cisco.com/warp/public/105/default.html		
Cisco IOS IP and IP Routing Configuration Guide			PC1-C149 to 180	
Cisco IOS Software Release 12.1(2)T	OSPF Fast Reduction			
Web Site	RFC 2328 Version 2	OSPF Faqs.org/rfcs/rfcs/rfc2328.html		
Advanced Cisco Router Configuration: Student Guide, Release 11.0				Cisco
Introduction to Cisco Router Configuration: Student Guide, Revision 11.3				Cisco
Advanced Cisco Router Configuration: Student Guide, Release 11.2				Cisco
Cisco 1000BaseT GBIC	Data Sheet			Cisco
Router Products Configuration and Reference	Configuration and Reference		Chapter 1	

(待续)

Resource	Title/ Resource Type	Web Page	Chapter	Author
Web Site	"Connectors and Cables"	Cisco.com/univercd/ cc/td/doc/product/ lan/c2900x1/ gbic/fig_gbic/ mamopins.html		
Layer 3 Switching Software Feature and Configuration Guide	<i>Configuring Bridging</i>			



北航

C0698884