



Cisco 职业认证培训系列
CISCO CAREER CERTIFICATIONS

ciscopress.com



CCIE 实验指南 (第2卷)

CCIE® Self-Study
CCIE Practical Studies
Volume II

Hands-on preparation for the CCIE lab exams

内附光盘

网友dada147友情制作

[美]

Karl Solie, CCIE #4599
Leah Lynch, CCIE #7220

著

姚军玲, CCIE #11470

顾彬, CCIE #5511

译

梅洪涛 胡捷

人民邮电出版社
POSTS & TELECOM PRESS



CCIE 实验指南 (第2卷)

- 通过5个综合实验练习场景来准备CCIE实验室考试；
- 通过对案例的研究及每章课程的学习提高网络知识技能；
- 深入了解思科3550智能交换平台，包括快速及多生成树协议；
- 学习通过路由映射和策略路由选择来改变、影响和控制路由选择行为；
- 掌握组播路由和交换知识；
- 检查路由器的性能管理和服务质量，包括Cisco IOS软件的服务质量技术、ATM服务质量、Intserv模式、Diffserv模式，以及监管、整形、队列等技术；
- 配置并调试BGP路由协议，检查iBGP和eBGP实施、通告BGP网络，配置复杂的BGP策略以及与内部网关协议之间的集成。

CCIE

ciscopress.com

《CCIE实验指南(第2卷)》补充了《CCIE实验指南(第1卷)》没有涉及的内容。通过建立复杂的路由和交换模型提供实践练习，指导读者掌握历时一整天的实验室考试中所有必须具备的能力。这些实验场景帮助读者掌握通过CCIE实验室考试所需要的各个领域的知识和技能。这些能力同时还可以对日常工作起到很大帮助作用。

本书可以同时作为专家级的网络参考手册和CCIE实验室考试的备考工具。本书专注于关键的路由、服务质量和交换等领域。内容包括新的强大的Catalyst 3550智能以太网交换机，并且详细讨论了边界网关协议(BGP)，另外涉及到服务质量、组播、路由映射、策略性路由及性能管理。每一章都有相应的技术概括和实验室场景来演示所介绍的技术的实际应用。本书最后提供了5个集成了之前介绍的所有概念和技术的复杂实验室场景，这些仿真的综合实验室场景将为学生顺利通过一天的实验室考试提供极大帮助。

“《CCIE实验指南(第2卷)》结构清晰，它使所有重要的实验练习准备更加高效。”

——Mike Reid, Cisco公司CCIE项目经理

内附光盘



随书光盘中提供了第10章中5个实验的完整解决方案。

ISBN 7-115-13729-3



9 787115 137296 >

ISBN 7-115-13729-3/TP·4849
定价:128.00元(附光盘)

Cisco 职业认证培训系列

CCIE 实验指南 (第2卷)

[美] Karl Solie, CCIE #4599 著
Leah Lynch, CCIE #7220

姚军玲, CCIE #11470

顾 彬, CCIE #5511

梅洪涛

胡 捷

译

网友dada147友情制作

人 民 邮 电 出 版 社

内容提要

本书是《CCIE 实验指南（第 1 卷）》的姊妹篇，本书不涉及具体协议的大量细节，而是提供实际的配置指导，以帮助读者提高网络知识技能。第 2 卷还介绍了很多基础或核心网络技术的大量信息，演示了如何能实现这些技术，并利用实际范例指导读者完成更高级的技术实现。在每一配置章节的最后，读者可以通过完成一个应用了刚刚讲过的技术的实验，来测试对这一主题知识的掌握情况。全书内容分为 6 个部分。第一部分深入地研究了新的思科 3550 智能交换平台；第二部分分析和演示了简单而功能强大的路由映射应用；第三部分深入研究了路由器和交换机平台中的组播路由和交换，演示了组播路由在真实场景中的应用；第四部分全面地研究了路由器性能管理和服务质量；第五部分详细研究并分析了 BGP 理论和配置；第六部分基于《CCIE 实验指南》两卷中的所有信息，并结合这一领域需要的各方面的技能，创建了 5 个有挑战性的实验场景。

本书适合于至少已经获得了 CCNA 或 CCDA 认证的网络工程师。无论你是计划参加 CCIE 考试，还是希望通过此书提高理论水平和工作技能，这本书都不会令你失望。

关于作者

Leah Lynch, CCIE # 7220 路由/交换，是一家大型金融机构的网络工程师。Leah 有超过 7 年的 IT 行业从业经验，有 4 年的时间专注于不同种类的互连网络环境，包括银行、零售、医药、政府、制造业、社团、销售、网络服务提供商、通信和 2.5/3G 无线网络。Leah 还拥有多个其他的思科证书，当前正在准备通信和服务的 CCIE 认证。她编写了第 6~9 章（服务质量和 BGP 章节）的内容。

Karl Solie, CCIE # 4599，是 Solie Research 咨询公司的首席工程师。Karl 在美国和其他国家诸如 McDonnell Douglas/Boeing、Unisys 等公司以及富尔顿和洛杉矶国家政府设计和实现了一些大型的基于 IP 和 SNA 的互连网络，在这方面有超过 14 年的经验。Karl 在思科专业发展方面很积极，除了本卷，他还编著了《CCIE 实验指南（第 1 卷）》（人民邮电出版社翻译出版）。Karl 是思科认证讲师，他在 Minneapolis 为 Ascolta 培训公司做培训。Karl 在 Wisconsin-Stout 大学的专业是应用数学，在位于 Irvine 的 California 大学获得法律文学学士学位。

关于合作者

Scott Morris, CCIE # 4713，是思科认证讲师。Scott 目前已拥有 4 个 CCIE 证书（路由与交换、ISP/Dial、安全和服务提供商），现在正在准备他的第 5 个认证（语音）。他到世界各地授课，并且为各种项目做咨询。为了不感到无趣或者说停滞，他还开始将兴趣扩展到 Juniper 网络世界（现在越来越具影响力的 JNCIS）的咨询和培训方面。他主要的兴趣和专长是安全、IP 技术、Cable modem 网络和高级路由。不出差的时候，他住在 Lexington, Kentucky。有时，他为各种公司做 CCIE（路由与交换）的培训，目前在 IPExpert。他参与了一些图书的编写，并且担任很多图书的技术编辑。现在他经营自己的公司，名字叫 Emanon.com，同时为 Uber-Geek.Net

关于技术审核人

Jennifer DeHaven Carroll, CCIE # 1402, 是朗讯科技公司的首席顾问。在过去的 15 年中，她规划、设计和实现了很多大型网络。她也曾开发和教授所有有关 IP 路由协议的理论以及思科实现方法的课程。Jenny 和 Jeff Doyle 合著了《TCP/IP 路由技术（第 2 卷）》（人民邮电出版社翻译出版）。

Greg Tillett, CCIE # 5231, 目前正在准备他的第二个关于安全领域的 CCIE 认证。自从取得了第一个 CCIE 认证后，他又通过了两次认证。Greg 是思科系统公司的咨询系统工程师，专注于安全、虚拟专用网和校园网技术。当前他担任思科系统工程师和客户经理，在思科研讨会上为各种听众介绍这些技术。自从 1997 年加入思科以后，他支持过很多客户，如美国州和地方政府、K-12 以及高教部门的客户和多个财富 100 客户的全球网络。这些经历给了他对于设计和支持类型完全不同的多业务网络的独特观点。

Kevin Turek, CCIE # 7284, 目前在 Research Triangle Park 的思科联邦支持项目组做网络咨询工程师。他现在为思科的美国国防部门的一些客户提供支持。Kevin 也是思科内部虚拟服务质量小组的成员，这个小组对内部的思科工程师和外部的思科客户的服务质量发展提供支持，当他们有服务质量应用需求时，向他们提供当前最优的方案。Kevin 在位于 Stony Brook 的纽约州立大学取得了商业管理工学学士学位。

献 辞

Leah Lynch: 此书献给我的丈夫 Chad Lynch，他一直支持、倾听和鼓励我。我爱你。

Karl Solie: 此书献给我的家人——我的爸爸妈妈，John 和 Linda Solie；我的两个兄弟，Mike 和 Jim。祝愿我们有一个亲密的家庭并且像国王那样富有。此书也献给我的妻子 Sandra 和两个女儿 Amanda 和 Paige，感谢你们几年来所有的付出和永恒的爱，你们三个使我的每一天都更温暖、更明亮和更快乐。

致 谢

Leah Lynch: 很多人一起工作，才完成了此书。首先，我要感谢所有来自 Cisco Press 的人——Brett Bartow、Chris Cleveland 和 Greg Balas，他们帮助提供原始资料，创作了一本真正的 Cisco Press 书籍；技术编辑 Jenny Carroll、Greg Tillett 和 Kevin Turek，他们审校我们的工作，并发现了所有不易发现的小错误。谢谢你们！

我还要感谢 Karl Solie，很多个夜晚他在电话中与我们讨论意见和内容；我不能只感谢 Karl 而不感谢他的妻子 Sandra，她与我们一起度过了这些夜晚。

我还要感谢 Jenny Carroll 和 Jeff Doyle，是他们把我介绍给 Cisco Press，帮助我迈出了成为作家的第一步。

我还要感谢 Pan Chou，我的非常有耐心的朋友，回答（至少试着回答）我模糊的 BGP 问题。还有 Scott Downing，我从他那里获得了灵感。我也要感谢 Mike Flannagan，回答我奇怪的服务质量问题；以及 Daniel Walton，感谢他在思科网络用户大会上出色的 BGP 介绍和 Q/A 对话。

当然，我必须感谢我的丈夫 Chad Lynch，他耐心地等我完成这个项目，并提供现场编辑。谢谢你忍受了我两年来的持续工作，现在我们可以休假了。

我还要感谢我的朋友 Erin Heitz，是他帮助我走进了这个领域，并激发我开始了一个真正的 IT 事业。

谢谢我的导师 George Sereno，感谢他所有的好建议和他的正直；最后，我要感谢我的家庭成员 Lynches 和 Sifuentes，感谢他们的爱和支持。

Karl Solie: 如果没有这么多 CCIE、编辑、技术人员和朋友的奉献，是不可能完成如此大的项目的。首先我要感谢 Leah 提到的 Cisco Press 的所有人，特别是主编 John Kane，他给了我非常好的机会使我成为 Cisco Press 的作者。

我还要感谢我的合著者 Leah Lynch，感谢她在这个项目中辛苦的工作，没有她的投入就没有本书。

真诚地感谢所有参与这个工作的 CCIE——Scott Morris 贡献了他在组播路由方面的专家意见，编著了第 3 章；同时，我们的技术审校人 Jennifer Carroll、Greg Tillett 和 Kevin Turek 也做了出色的工作。

我还要感谢所有《CCIE 实验指南（第 1 卷）》的读者，特别是那些来信询问如何成为 CCIE 的人。

最后，我要感谢我的父母，感谢他们对我的支持，感谢他们不断的祝福和对我的支持。

序

准备 CCIE 认证是一个具有挑战性的个性化过程，不同的考生有不同的取得成功的路径。我很高兴能和数以千计的 CCIE 笔试通过者会见和交谈，在我看来，取得认证的惟一的最主要的因素是笔试通过者在准备阶段所做的大量动手练习。Karl Solie 和 Leah Lynch 编写的《CCIE 实验指南（第 2 卷）》提供了一个清晰的框架，对最重要的动手实验部分很有益处。所有 CCIE 认证的特点就是考试内容覆盖面宽，很多笔试通过者对选择从哪里开始和如何开始他们的准备工作有困惑。本书和它的姊妹篇，《CCIE 实验指南（第 1 卷）》能够帮助笔试通过者专注于可能出现在考试中的关键点。除了通过阅读和完成实验例题获得知识外，本书可以作为自学的一个切入点，在这里笔试通过者要具有真正专家级的技巧来探究“如果……就……”这样的实验场景。

现在 CCIE 认证已有 10 年历史，在我们的行业中仍然位于认证项目的前列。衡量认证项目生命力的标准之一是考试准备资料的发展，本卷是对目前 CCIE 笔试通过者可利用的一系列资源的很有价值的补充。就像它的姊妹篇一样，我相信它对任何准备资料都会是一个极好的补充。

思科系统公司 CCIE 项目组，Mike Reid 经理。

本书采用的图标



路由器



网桥



集线器



分布式服务单元 /
集中式服务单元



Catalyst交换机



多层交换机



调制解调器



ATM
交换机



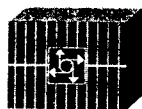
ISDN/
帧中继交换机



通信服务器



网关



访问服务器



PC 机



安装了软件
的 PC 机



Sun
工作站



Macintosh
计算机



终端



思科 Works
工作站



Web
服务器



文件
服务器



笔记本电脑



打印机



IBM
主机



前端处理器



集群控制器

线路：以太网

线路：串行线

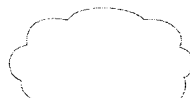
线路：交换串行线



令牌环网



FDDI



网络云

命令语法规范

本书中使用的命令语法规范与《Cisco IOS Command Reference》中使用的相同。这些规范如下：

- 竖线 (|) 分隔符，表示选择其中之一；
- 方括号 [] 表示可选的字段；
- 大括号 { } 表示必选；
- 方括号中的大括号 [{ }] 表示在可选的字段里必选；
- 粗体字表示按照显示的文字输入命令及关键字。在配置的范例及输出（不是一般的命令语法）中，粗体表明需要用户手工输入的命令（例如 **show** 命令）；
- 斜体字 表示需要你输入的实际值。

前言

CCIE 是现有的最具挑战性的认证之一。很多 CCIE 笔试通过者花几个月时间学习，甚至尝试几次实验考试后才通过。如果你正在考虑报考 CCIE，你可能意识到需要大量的自学、培训和经验来保证通过实验考试。尽管有困难，取得 CCIE 认证仍是一种非常有益的经历，要求笔试通过者更新他们熟悉的技术技能，扩展他们不熟知的领域，通常需要大量的专业技术准备。在压力和时间限制下掌握不同的技术并亲自动手增加一个人查找和排除故障的能力。

CCIE 实验考试是一个持续一天时间的考试，它测试笔试通过者在相当大的压力下，在有限的时间里处理多种协议的能力。笔试通过者必须运用他们的思科 IOS 软件知识，对一个他们不熟悉的网络进行配置、测试和查找、排除故障，从而检验其在压力环境下独立工作的能力。因为 CCIE 项目为跟上行业的需要而不断地变化，所以笔试通过者经常会遇到他们先前并不熟悉的技术。这使得 CCIE 项目对笔试通过者和雇主的适用性更广，因为笔试通过者被测试的范围不仅包括应用于他们目前的工作环境中的技术，也包括应用在很多不同的市场中的技术。路由与交换考试所包含的协议和技术适用于大量不同的网络类型：企业集团、零售、服务提供商和其他。这种全面的技能有益于笔试通过者、他们的雇主和他们的合作者。

思科公司建议 CCIE 笔试通过者在开始实验考试前至少要有两年有关思科产品的经验、正式的技术培训和相当长时间的自学。本书是第 2 卷，是帮助 CCIE 笔试通过者准备实验室考试自学用的。你可以使用书中的范例在不同的动手实验场景中测试你的技术知识。强烈建议你使用 CCIE 系列丛书中的每一本来准备考试，通读理论，做实验，复习熟悉的技术。通过 CCIE 考试之后，大多数人有很强的成就感，不再被时间限制和压力所胁迫。

我们坦诚地对你讲：成为 CCIE 之路将是漫长和艰难的，它也是对精神上的极大挑战。提到 CCIE 实验考试，它的测试

标准是严格的，监考人是严厉的。在成为 CCIE 的道路中，你不能按你的方式及意愿来争论或谈论某些技术问题。精心地准备，通向 CCIE 的道路没有捷径，所以不要浪费时间去寻找捷径。只要你的过程正确，当所有的都说了和完成了，最后你被分配你自己的 CCIE 号，这个感觉非常棒。你感觉所有的辛苦工作、付出和在实验室度过的漫长的孤独时光都得到了偿还。你将进入世界网络工程师最精英的行列——成为一名 CCIE。

《CCIE 实验指南（第 1 卷）》强调了成为 CCIE 没有捷径，没有一本万能书（包括《CCIE 实验指南》第 1 卷和第 2 卷）能帮助你成为 CCIE。除了更多地投入和大量的经验积累，没有一种所谓“买了这本书我们就能保证你通过”的捷径。本书假定大多数的 CCIE 笔试通过者对本书所覆盖的大部分技术至少有一些经验。CCIE 实验总在变化，可能的考试内容是深奥和广泛的。鉴于这些原因，没有办法提供一种 CCIE 学习的“惟一途径”。这并不是说认证指南这类书是没有价值的工具，应当把它们看作是你能利用的很多学习方法之一。

如同第 1 卷一样，第 2 卷中通常不涉及具体协议的大量细节，取而代之，它提供了实际的配置指导，可以帮助你提高网络技能，并介绍你可能还没有接触过的领域中的技术。第 2 卷同它的姊妹篇第 1 卷一起介绍了很多基础或核心网络技术，包括很多新的概念，如果将其应用到运营的网络模型，将有助于增强你的网络技能，促进你参加和通过 CCIE 实验考试的准备工作。

《CCIE 实验指南（第 2 卷）》继续了《CCIE 实验指南（第 1 卷）》的内容。《CCIE 实验指南（第 1 卷）》集中于构造 ISO 各层之间复杂的互连网络场景。它包括了物理访问、局域网和广域网数据链路协议，如帧中继、HDLC、PPP、ATM、以太网和令牌环。《CCIE 实验指南（第 1 卷）》详细介绍了思科 Catalyst 平台，包括令牌环 Catalyst 3924 和 Catalyst 35xx/5500/6500 系列产品。第 2 卷继续 Catalyst 家族交换机的介绍，集中于新的功能强大的 Catalyst 3550 智能以太网交换机。这部分学习包括三层交换和新的 802.1w 与 802.1s 生成树协议。

《CCIE 实验指南（第 1 卷）》也包括内部网关协议（IGP），如 RIP、IGRP/EIGRP 和 OSPF。《CCIE 实验指南（第 2 卷）》继续集中于介绍主要的外部网关协议（EGP）和边界网关协议（BGP）。

除路由选择协议和以太网交换建模外，本书详细地研究了服务质量（QoS）。同 BGP 一样，本书用一定的篇幅讲述高级服务质量技术，包括资源预留协议（RSVP）、区分服务编码点（DSCP）领域和加权随机早期检测（WRED）的论题。在讲述 ATM 和语音技术时也会讨论服务质量。

本书组织结构

本书包括 6 个部分，对特定技术提供了详细的技术资料。本书演示了如何实现这些技术，并利用实际范例指导你完成更高级的技术实现。在每一配置章节最后，你可以通过完成一个应用了刚刚讲过的技术的实验，来测试这一论题的知识。做完实验后，你能通过使用实验室给你的配置与我们在实验过程中生成的配置进行比较。本书讨论的主题分成下列几部分：

- 第一部分：以太网交换；
- 第二部分：控制网络传播和网络访问；
- 第三部分：组播路由；
- 第四部分：性能管理和服务质量；

- 第五部分：BGP 理论和配置；
- 第六部分：CCIE 练习实验。

《CCIE 实验指南（第 2 卷）》是可定制的学习资源。像这样按照技术细节分节使你能有效地利用你的学习时间。每一章从基本理论开始，逐步引入配置范例，你可以在自己的实验中模拟这些范例。大部分章节还包括了实际范例，这些范例应用了更复杂的配置论题，在整个实验过程中，你能够用作者在写作过程中使用的配置进行实际操作。如果你对某一技术或配置步骤有疑问，回到理论和配置部分快速查找，然后再试着做范例或实验，直到你理解了它是如何工作的。不要担心超过了实验的任何限制而不去进一步研究技术或花费时间研究一个条款的细节。你从这些网络模型的工作中得到的经验将会应用到任何其他培训和你已有的经验中，为复杂的网络实现做准备。当你对其一部分感觉轻松时，继续下一部分；如果你认为你不需要学习某一部分的资料，那么跳到最后，做实验来验证你已经熟练掌握了这个主题。本书某些章也提供“进一步阅读资料”部分，这一部分给出一些参考书目，阅读这些参考书可以进一步研究这个论题所包含的其他的详细说明。本书建立在第 1 卷所包含的信息之上，假设你已经具备配置核心技术如 IGP 路由协议、基本局域网交换概念、广域网协议配置所必备的坚实的基础技能，并且你知道如何配置 IP 服务，如网络地址翻译（NAT）。关于这些技术更多的信息，请查阅第 1 卷相应的章节。

第一部分深入地研究了新的思科 3550 智能交换平台——探究这个新平台的性能，回顾旧的交换技术，查看这些交换技术新的改进应用。在路由和交换的实际范例和实际实验场景中，你可以充分利用这个平台的所有性能。

第二部分分析和演示了简单且功能强大的路由映射应用，还包括了经常被忽略的路由映射。你将学到使用路由映射来改变或影响路由行为，基于协议特性或策略路由的流量控制方法。路由映射是很多高级路由方案中不可或缺的一部分，好的路由映射配置技巧对 BGP 路由是不可或缺的。这部分提供了路由映射及其应用的基本概况，并为你学习本书后面论及的一些技术做了准备。

第三部分深入研究了路由器和交换机平台中的组播路由和交换，在网络模型中应用实际理论，从而演示在真实场景中组播路由的应用。

第四部分首先以一个简短的与性能相关的路由器 **show** 命令分析路由器的性能，全面地研究了路由器性能管理和服务质量。利用来自这些命令的信息，你可以通过应用一些思科 IOS 软件扩展的服务质量技术来提供最高级别的服务。接下来论及 ATM 服务质量——首先回顾了 ATM 理论、ATM 和帧中继的比较，然后简单地回顾了使用新的思科 IOS 软件 ATM 配置命令对 ATM PVC 的配置，之后焦点回到 ATM 服务质量机制，根据网络服务级别的需要来应用这些流量控制技术。这部分资料也有助于企业网络专职人员理解他们的服务提供商经常使用的一些术语。这一部分还论及了三层交换方法，演示如何根据特殊的网络特性和路由器硬件及接口类型来确定正确的交换方式。

第 5 章揭开了围绕服务质量集成服务和区分服务的神秘烟云。这一章回顾了 RSVP 理论和在思科路由器上的配置，深入探究了 RSVP 的 **show** 和 **debug** 命令。你将对一个最通俗的 RSVP 网络应用即 Voice over IP 应用 RSVP 配置。之后该章研究了目前流量标记和分类能够提供的主要区分服务，这些技术利用的是保存在 IP 服务类型（ToS）字段的信息。这部分探究了 IP 优先级，新近出现的 IP 区分服务编码点（DSCP）字段和 WRED（避免拥塞算法）。探究了流量分级的方法之后，你可以在几个以 Voice over IP 作为网络应用的网络模型中应用这些技术。

第6章实际上很短小，对思科 IOS 软件目前可提供的队列、整形、分类和策略技术进行了研究并提出了主要观点。本章从研究主要的4个基本队列方法开始，通过引入更新的、更高级的队列方法，如基于类别的加权公平队列（CBWFQ）和低延迟队列（LLQ），深入研究了队列理论——这些技术涉及了本书到目前为止覆盖的许多主题。随后本章再次讲述流量整形并探究新的分类整形方法。没有寻址流量策略，服务质量章节就不完整，本章演示了新的策略方法，你可以用这些方法作为防止或包容某些病毒和不受欢迎的协议的传播的保护方法，从而可以维持一定的网络性能水平。

第五部分研究了曾经写过的最令人兴奋和容易混淆的协议：BGP。这一部分不像其他部分，整个第7章只专注于BGP理论。这章通过研究BGP有限状态机制的状态、5个BGP消息、BGP属性、路由反射器和联盟，给出了现在可用的最新最全面的BGP理论描述。这一章主要适用于思科的BGP技术实现，但资料的来源不限于思科设备。通过提供简要的BGP理论回顾，为以后的章节做实际配置提供了理论准备。

第8章从服务提供商和企业的角度对第7章的BGP理论进行了应用，探究BGP基本配置，提供一些快速BGP配置技巧，探究BGP路由对路由器的影响。本章包括了你可以应用于这个领域的实际实现的技巧。回顾了基础知识之后，这一章研究了成功的BGP实现的核心——使用BGP的show和debug命令显示配置数据和诊断问题。这一章研究了以前在BGP调试会话中显示的没有说明的条目，逐行解释BGP的debug命令的输出。通过介绍BGP发现处理故障的方法论和显示哪些命令能快速帮助你以最小的网络影响来诊断问题，为你处理几乎所有BGP的问题做了准备。

第9章研究了I-BGP和E-BGP的实现方法、BGP如何使用它的路由表、通告BGP网络、BGP和IGP的集成。这章有助于减少很多容易混淆和难懂的概念的困扰，如与两个服务提供商有BGP连接，常见的I-BGP网状网问题。本章不仅是学习指导，也是实际环境的指导，在这方面它能够使你节省几小时发现和处理故障的时间——以前两章提供的信息为基础，通过深入研究这些有益的素材资料：路由反射器、联盟、重分发、路由过滤和条件路由通告。之后，本章空前地关注一个最容易混淆和难懂的BGP论题：应用规则表达式。本章通过应用多个范例演示了规则表达式是如何工作的，使用少为人知的show命令为任务找到正确的规则表达式。研究了规则表达式之后，通过应用包含在BGP属性中的信息，使用规则表达式来过滤和修改路由。本章也包括了多路径、私有自治系统号、后门、对等体组和聚合。最后，将这个信息应用于多个实际类型的场景，建立一个牢固的BGP基础，这使你能自信地处理你所遇到的任何BGP问题。

第六部分，第10章，基于《CCIE实验指南》两卷中的所有信息，结合各方面的技能来创建5个有挑战的实验场景。根据读者对第1卷的反馈，在本卷中我们包括了实验配置，便于参考。

最后的注释

10年里全球仅有10 000多名CCIE，CCIE认证仍然是个人能够获得的最具挑战性的认证之一。它是惟一需要桌面协议、路由选择协议、以太网交换和LAN/WAN技能方面的知识，加上坚实的IP服务知识的考试。我们真诚地希望《CCIE实验指南》第1卷和第2卷成为你准备CCIE考试和在这个领域不可或缺的工具。祝你好运！祝你成功！

——Karl Solie 和 Leah Lynch

目 录

第一部分 以太网交换

第 1 章 在思科 Catalyst 3550 以太网交换机上配置

高级交换 3

1.1 进入思科 Catalyst 3550 智能以太网交换机 4

1.2 以太网交换机回顾 5

1.2.1 虚拟局域网 (VLAN) 5

1.2.2 VTP 和骨干协议 9

1.2.3 以太网物理特性：半双工和全双工以太网 19

1.3 IEEE 802.1d 生成树协议 (STP) 20

1.3.1 生成树的操作 20

1.3.2 STP 计时器 25

1.4 Catalyst 3550 的配置模式和术语 26

1.4.1 交换端口 26

1.4.2 以太通道端口组 27

1.4.3 交换虚拟接口 (SVI) 27

1.4.4 路由端口 28

1.4.5 配置 Catalyst 3550 以太网交换机 29

1.4.6 在 Catalyst 3550 以太网交换机上配置高级特性 66

1.5 实验 1：配置以太通道、三层交换、路由端口和 SVI 88

1.5.1 练习场景 88

1.5.2 实验练习 88

1.5.3 实验目的 89

1.5.4 需要的设备 90

1.5.5 物理布局和预规划 90

1.5.6 实验步骤 90

1.6 实验 2：配置 802.1w RSTP 和 802.1s MST、三层交换以及 VLAN 映射 99

1.6.1 练习场景 99

1.6.2 实验练习 99

1.6.3 实验目的 99

1.6.5 物理布局和预规划	101
1.6.6 实验步骤	101

第二部分 控制网络传播和网络访问

第 2 章 配置路由映射和策略性路由	115
2.1 路由映射介绍	115
2.1.1 配置路由映射	118
2.1.2 路由映射和策略性路由	142
2.1.3 路由映射的“Big Show”	151
2.2 实验 3: 配置复杂的路由映射和使用标记	152
2.2.1 练习场景	152
2.2.2 实验练习	152
2.2.3 实验目的	153
2.2.4 需要的设备	154
2.2.5 物理布局和预规划	154
2.2.6 实验步骤	155
2.3 实验 4: 配置策略性路由	162
2.3.1 练习场景	162
2.3.2 实验练习	162
2.3.3 实验目的	162
2.3.4 所需的设备	163
2.3.5 物理布局和预规划	164
2.3.6 实验步骤	164

第三部分 组播路由

第 3 章 配置组播路由	177
3.1 组播的基础知识	177
3.2 IP 组播地址	179
3.2.1 本地链路地址	180
3.2.2 全局分配地址	180
3.2.3 源特定的地址	180
3.2.4 GLOP 地址	181
3.2.5 管理性范围的地址	181
3.2.6 二层的组播地址	181
3.3 组播分发树	183
3.3.1 源树	183
3.3.2 共享树	184
3.3.3 组播转发	185
3.3.4 反向路径转发	185
3.4 与协议无关的组播	185
3.4.1 PIM 密集模式	186

3.4.2 PIM 稀疏模式	186
3.4.3 双向 PIM	187
3.5 实验 5: 设置基本的组播	188
3.5.1 实验 5: 解决方案	188
3.5.2 实验 5: 配置	189
3.6 组播帧中继	190
3.7 组播 TTL	190
3.8 组播边界	191
3.9 PIM 自动 RP	191
3.10 实验 6: 设置帧中继组播路由	193
3.10.1 实验 6: 解决方案	193
3.10.2 实验 6: 配置	194
3.11 组播加入	195
3.12 实验 7: 组播加入	196
3.12.1 实验 7: 解决方案	196
3.12.2 实验 7: 配置	197
3.13 控制组播	198
3.13.1 快速交换	199
3.13.2 组播末梢	199
3.13.3 负载分担或者不连续的组播网络	199
3.14 实验 8: 高级组播传输	200
3.14.1 实验 8: 解决方案	200
3.14.2 实验 8: 配置	201
3.15 DVMRP 组播路由	202
3.16 PIM 版本 2	203
3.17 实验 9: PIM	203
3.17.1 实验 9: 解决方案	204
3.17.2 实验 9: 配置	204
3.18 监控和测试	206
3.18.1 show 和 debug 命令	206
3.18.2 mtrace、mrinfo 和 mstat 命令	206
3.18.3 组播故障排查范例	207
3.18.4 组播路由管理器 (MRM)	209
3.19 CCIE 组播实验场景	211
3.20 进一步阅读资料	211

第四部分 性能管理和服务质量

第 4 章 路由器的性能管理	215
4.1 决定路由器的性能	216
4.1.1 验证思科 IOS 软件和内存的配置	216
4.1.2 决定网络应用程序的需求	217
4.1.3 验证路由器的接口性能	219
4.2 ATM: 其他的广域网技术	226

4.3 交换模式	238
4.3.1 进程交换	239
4.3.2 快速交换	239
4.3.3 最优或者分布式交换	239
4.3.4 NetFlow 交换	240
4.3.5 思科快速转发	240
4.4 压缩	244
4.4.1 Stacker 压缩算法	245
4.4.2 Predictor 压缩算法	246
4.4.3 实验 10: ATM 服务质量	247
4.5 进一步阅读资料	258
第 5 章 集成和差分服务	261
5.1 集成服务	261
5.2 范例: RSVP 和 VoIP	276
5.2.1 实验练习	276
5.2.2 实验目的	276
5.2.3 需要的设备	277
5.2.4 物理布局和预规划	277
5.3 区分服务	285
5.3.1 设置 IP Precedence	286
5.3.2 使用 DSCP 标记流量	288
5.3.3 使用 WRED 避免拥塞	291
5.4 练习场景	297
5.5 进一步阅读资料	305
第 6 章 服务质量——速率限制和对流量进行队列处理	309
6.1 最基础的: 先进先出队列	309
6.2 加权公平队列	310
6.3 优先级队列	316
6.4 定制队列	329
6.5 使用服务质量实施流量策略	338
6.6 流量整形	339
6.7 使用 CAR 分类和标记流量	342
6.8 优化实时的语音流量	346
6.9 基于类别的队列解决方案	348
6.9.1 基于类别的整形	358
6.9.2 基于类别的监管	360
6.9.3 低延迟队列 (LLQ)	368
6.10 练习场景	371
6.10.1 实验 12: 定制队列	371
6.10.2 实验目的	373
6.10.3 实验任务	373
6.10.4 实验步骤	375
6.11 实验 13: 使用 CBWFQ 和 NBAR 管理因特网的流量	382

6.12 进一步阅读资料.....	388
-------------------	-----

第五部分 BGP 理论和配置

第 7 章 BGP-4 的原理.....	393
7.1 BGP 简介.....	393
7.2 BGP 路由表.....	395
7.3 邻居关系.....	398
7.4 BGP 报文.....	408
7.4.1 OPEN 报文.....	408
7.4.2 UPDATE 报文.....	411
7.5 通知 (NOTIFICATION) 报文.....	416
7.5.1 保活 (keepalive) 报文.....	418
7.5.2 路由刷新报文.....	420
7.6 BGP 状态机的操作.....	421
7.6.1 空闲状态.....	422
7.6.2 连接状态.....	424
7.6.3 激活状态.....	426
7.6.4 OPEN 发送状态.....	426
7.6.5 OPEN 确认状态.....	427
7.6.6 已建立状态.....	428
7.7 BGP 路径属性.....	429
7.7.1 起源属性.....	429
7.7.2 AS 路径属性.....	431
7.7.3 下一跳属性.....	434
7.7.4 多出口鉴别器 (MED) 属性.....	436
7.7.5 本地优先 (LOCAL_PREF) 属性.....	438
7.7.6 权重 (WEIGHT) 属性.....	439
7.7.7 原子聚合 (ATOMIC_AGGREGATE) 属性.....	440
7.7.8 聚合者属性.....	441
7.7.9 BGP 团体属性.....	442
7.8 路由反射器.....	443
7.8.1 起源者识别符 (ORIGINATOR_ID).....	445
7.8.2 集群列表 (CLUSTER-LIST).....	446
7.9 联盟.....	446
7.10 对等体组.....	450
7.11 路由选择处理.....	450
7.12 总结.....	451
7.13 进一步阅读资料.....	451
第 8 章 BGP-4 配置介绍.....	453
8.1 BGP 配置的先决条件.....	453
8.1.1 评估路由器的 BGP 能力.....	454
8.1.2 BGP 配置提示.....	456

8.2	配置和故障排查 BGP 邻居关系	458
8.2.1	使用 BGP 报文作为征兆	477
8.2.2	BGP 空闲/活动场景	479
8.3	BGP 邻居的配置	484
8.4	E-BGP 对等关系	504
8.5	BGP 和 IGP 的相互作用	510
8.6	BGP 和 IP 路由表	512
8.7	通告本地网段	512
8.7.1	通告连接网段	513
8.7.2	通告静态路由	516
8.7.3	通告由 IGP 学到的路由	518
8.8	实验 14: BGP 路由	519
8.8.1	实验练习	519
8.8.2	实验目的	520
8.8.3	需要的设备	520
8.8.4	物理布局和预规划	520
8.8.5	实验步骤	521
8.9	进一步阅读资料	532
第 9 章	高级 BGP 配置	535
9.1	BGP 邻居认证	535
9.2	简化大型 BGP 网络	536
9.2.1	路由反射器	537
9.2.2	联盟	540
9.3	实际范例: BGP 联盟	542
9.3.1	私有自治系统	551
9.3.2	使用对等体组简化配置	553
9.4	路由聚合	555
9.4.1	聚合与路由抑制	559
9.4.2	条件路由通告	562
9.5	过滤 BGP 路由	564
9.5.1	使用分发列表来过滤网段前缀	564
9.5.2	使用前缀列表过滤 BGP 路由	566
9.5.3	采用路由映射过滤 BGP 路由	568
9.6	使用 BGP 属性来建立路由策略	571
9.6.1	修改起源属性来影响路径选择	572
9.6.2	使用 AS 路径属性来影响路径选择	576
9.6.3	使用 AS 路径属性过滤 BGP 路由	578
9.6.4	对于路径操作修改下一跳属性	584
9.6.5	使用多出口鉴别器属性来指定最佳路径	586
9.6.6	使用本地优先属性来指定网络的出口点	589
9.6.7	使用权重属性来影响路径选择	591
9.6.8	团体属性的许多用法	593
9.7	使用多路径	600
9.8	实际范例: 多归路一个 BGP 网络	605

9.9 管理距离和在 BGP 上的效果.....	615
9.10 BGP 路由衰减.....	618
9.11 调整 BGP 的性能.....	620
9.12 实验 15: 多归路一个 BGP 网络.....	625
9.12.1 实验练习	625
9.12.2 实验目的	625
9.12.3 需要的设备	625
9.12.4 物理布局和预规划	626
9.12.5 实验练习	627
9.12.6 实验步骤	630
9.13 进一步阅读资料.....	655

第六部分 CCIE 练习实验

第 10 章 CCIE 准备和练习实验.....	659
10.1 CCIE 准备	659
10.2 CCIE 练习实验	665
10.3 CCIE 练习实验: Broken Arrow	666
10.3.1 准备阶段——帧中继交换机和 ATM 配置.....	666
10.3.2 规则	669
10.3.3 第 I 部分: IP 设置	669
10.3.4 第 II 部分: Catalyst 配置.....	670
10.3.5 第 III 部分: OSPF、RIP 和帧中继	671
10.3.6 第 IV 部分: EIGRP 集成	671
10.3.7 第 V 部分: 流量控制和 ISDN.....	671
10.3.8 第 VI 部分: BGP	672
10.3.9 第 VII 部分: 服务质量和 ATM.....	672
10.3.10 第 VIII 部分: DLSW+.....	673
10.4 CCIE 练习实验: !!! Boom... ..	673
10.4.1 准备阶段——帧中继交换机、骨干路由器和 ATM 配置.....	673
10.4.2 规则	677
10.4.3 第 I 部分: IP 设置	678
10.4.4 第 II 部分: Catalyst 配置.....	679
10.4.5 第 III 部分: OSPF、三层交换和帧中继.....	679
10.4.6 第 IV 部分: RIP、EIGRP、IS-IS 集成.....	679
10.4.7 第 V 部分: 路由过滤和控制	680
10.4.8 第 VI 部分: ISDN.....	680
10.4.9 第 VII 部分: BGP	680
10.4.10 第 VIII 部分: 服务质量	681
10.4.11 第 IX 部分: DLSW+.....	681
10.5 CCIE 练习实验: The Intimidator	681
10.5.1 准备阶段——帧中继交换机和骨干路由器配置.....	681
10.5.2 规则	685
10.5.3 第 I 部分: IP 设置	685

10.5.4	第 II 部分: Catalyst 配置	686
10.5.5	第 III 部分: OSPF 和帧中继	687
10.5.6	第 IV 部分: EIGRP 集成	687
10.5.7	第 V 部分: HSRP	687
10.5.8	第 VI 部分: BGP	687
10.5.9	第 VII 部分: Voice	689
10.5.10	第 VIII 部分: 服务质量	689
10.5.11	第 IX 部分: DLSW+	689
10.6	CCIE 练习实验: Enchilada II	690
10.6.1	准备阶段——帧中继交换机、骨干路由器和 ATM 的配置	690
10.6.2	规则	695
10.6.3	第 I 部分: IP 设置	696
10.6.4	第 II 部分: Catalyst 配置	697
10.6.5	第 III 部分: EIGRP、三层交换和帧中继	697
10.6.6	第 IV 部分: RIP、OSPF 集成	697
10.6.7	第 V 部分: 路由过滤和 HSRP	698
10.6.8	第 VI 部分: ISDN	698
10.6.9	第 VII 部分: ATM	698
10.6.10	第 VIII 部分: BGP	698
10.6.11	第 IX 部分: DLSW+	699
10.6.12	第 X 部分: NAT	699
10.6.13	第 XI 部分: 组播路由	699
10.7	CCIE 练习实验: Kobayashi Maru	700
10.7.1	准备阶段——帧中继交换机、骨干路由器和 ATM 配置	700
10.7.2	规则	703
10.7.3	第 I 部分: IP 设置	703
10.7.4	第 II 部分: Catalyst 配置	704
10.7.5	第 III 部分: OSPF、EIGRP、三层交换和帧中继	704
10.7.6	第 IV 部分: IS-IS 和 RIP 集成	705
10.7.7	第 V 部分: NAT 和 DHCP	705
10.7.8	第 VI 部分: 组播路由和 NTP	705
10.7.9	第 VII 部分: ISDN	705
10.7.10	第 VIII 部分: ATM	706
10.7.11	第 IX 部分: BGP	706
10.7.12	第 X 部分: 语音	707
10.7.13	第 XI 部分: DLSW+	707

第七部分 附 录

附录 A	思科 IOS 软件的限制和约束	711
A.1	思科 IOS 软件的限制和约束	711
A.2	集群的限制和约束	716
A.3	集群管理组限制和约束	717
A.4	重要注释	717

A.4.1 思科 IOS 软件注释	717
A.4.2 集群注释	718
A.4.3 CMS 注释	719
A.4.4 CMS 中的只读模式	719
A.4.5 版本 12.1 (11) EA1 中不支持的 CLI 命令	720
附录 B RFC	727
附录 C 参考书目	729
附录 D IP 前缀列表	733

第一部分

以太网交换

第1章 在思科 Catalyst 3550 以太网交换

机上配置高级交换

第 1 章

在思科 Catalyst 3550 以太网交换机 上配置高级交换

以太网通常被认为是一种进化中的协议，而不是一种创新的协议。一些年来，以太网已经进化到以惊人的速度建立了不同的标准。进化中的协议构建在当前的标准上，并且提供了某种形式的迁移途径。而创新的协议包括某种形式的科学突破或者使用了新的技术。创新的协议，如果保留的话，保留了现有的体系结构中很少的部分。

以太网的发展一直非常引人注目。IEEE 委员会的工作人员已经频繁地批准许多新的标准，包括用 IEEE 802.1w 来修改生成树协议，而无线的以太 IEEE 802.11a 和 IEEE 802.11b 将被运行在 54Mbit/s 速率上的 802.11g 取代。10/100Mbit/s 的以太网已经入户，而 10 吉比特 IEEE 802.3ae 的产品已经提供了 OC-192 的速率。工业专家预测吉比特以太网到桌面的实现仅仅是一个时间问题，而且 40Gb 的标准正在起草。例如，苹果计算机已经在它的笔记本中携带了吉比特以太接口，并且它的 G4/G5 的桌面系统在不久的将来也会将它变为现实。有人可能会说，进化将会让位于 WAN 和 MAN 的创新。想象有一天，也许在不久的将来，互联网服务提供商（ISP）给它们的用户提供无线的以太网接入，而节点之间（POP）以 10 吉比特的链路进行连接。如此高速的带宽使得在因特网上提供下一代的杀手应用（killer application）成为可能。

随着以太网的作用不断发展，思科的产品线也在不断地发展，它是市场上最先具有许多新的以太产品的厂商。其中一个特别需要关注并且适用于企业的产品就是思科 Catalyst 3550 智能以太网交换机。就像你将要在本章末尾看到的那样，思科做了非常卓越的工作，将 Catalyst OS（CAT OS）的特性

和传统的思科 IOS 软件特性结合在一起。Catalyst 3550 许多部分的配置，从某种形式来说，你应当是相当熟悉的。

本章集中讨论 Catalyst 3550 智能以太网交换机的软件配置部分。讨论包括 Catalyst 3550 的技术层面，会有一个详细的以太交换和生成树协议的介绍。本章提供了一个完整的方法来配置 VLAN、VLAN 骨干协议（VTP）和骨干，并且介绍了其他的二层/三层功能。本章还讨论了 3550 的高级配置，包括快速生成树协议和多生成树协议。

关于常用以太交换的概念和配置 Catalyst 3900 令牌环交换机，以及思科 Catalyst 2900/3500 和 5500/6500 系列交换机的更详细信息，请参考《CCIE（实验指南第1卷）》。

1.1 进入思科 Catalyst 3550 智能以太网交换机

思科 Catalyst 3550 是一款智能以太网交换机，它可提供理想的带宽、三层交换和一定程度上的服务质量（QoS）。这款交换机之所以被称为智能交换机，是因为它为传统的企业级接入交换机带来了许多高级特性。这款交换机可以基于第三、四层的信息做出转发决定，因此称其为智能交换机。思科增强的多层软件映像（EMI）允许此交换机在小型网络中充当骨干交换机，它可提供虚拟局域网之间的路由和热备份路由协议（HSRP）。图 1-1 展示了一台思科 Catalyst 3550 交换机。



图 1-1 思科 Catalyst 3550 智能以太网交换机

思科 Catalyst 3550 智能以太网交换机的某些关键特性如下：

- 高级的冗余和容错备份——诸如上行链路加速、骨干加速和 802.1w 快速生成树协议，极大地减少了故障的恢复时间。EMI 软件特性支持 HSRP 容错这类高级路由特性。
- 集成了带宽优化的思科 IOS 特性——诸如二层或者三层的以太通道（EtherChannel）可在交换机之间提供非常大的路径带宽，最大可达到 16 Gbit/s！基于每个 VLAN 的生成树协议（PVST+）和 VTP 剪枝协议允许高级的生成树控制。
- 高级服务质量和队列——思科 3550 支持基于 802.1p 的服务质量和差分服务编码点（DSCP）字段、加权轮循队列（WRR）和加权随机早期检测（WRED）。

其他的特性包括高级安全和管理特性、灵活的流量监管特性以及 EMI 软件所提供的思科快速转发（CEF）启用的高性能路由特性。组播路由也是由 EMI 软件所提供和支持的。

注意：这里高度概括了思科 Catalyst 3550 交换机其中一部分的绝对优越特性。如想了解这些特性和其他特性的更详细信息，参看 www.cisco.com。

思科 3550 交换机也遵循了 IEEE 和其他标准实体制定的一些最新标准和规章认证。下列这些标准都在思科 3550 以太网交换机上实现了：

- IEEE 802.1x 基于端口的认证；
- IEEE 802.1w 快速生成树协议；
- IEEE 802.1s 多个生成树协议（MST）；

- IEEE 802.3 在 10BASE-T、100BASE-T 和 1000BASE-T 端口上的全双工模式；
- IEEE 802.1d 生成树协议；
- IEEE 802.1p 基于类别的服务优化 (CoS) ；
- IEEE 802.1Q 虚拟局域网的骨干；
- IEEE 802.3 10BASE-T；
- IEEE 802.3u 100BASE-TX；
- IEEE 802.3ab 1000BASE-T；
- IEEE 802.3z 1000BASE-X；
- 1000BASE-X (GBICs)：1000BASE-SX、1000BASE-LX/LH 和 1000BASE-ZX、1000BASE-T、1000BASE-CWDM 以及 GigaStack GBIC；
- 远程监控 (RMON) 类型 I 和远程监控类型 II；
- 简单网络管理协议 (SNMP) 版本 1 和简单网络管理协议版本 2c。

当前生产的 Catalyst 3550 交换机有 4 种基本型号，每种型号都有许多不同之处，型号的数量也在持续增长。Catalyst 3550-24 和 3550-48 带有标准的多层软件映像 (SMI) 或 EMI 软件映像。Catalyst 3550-12T 和 3550-12G 出厂后的默认设置只有 EMI 软件映像，而 Catalyst 3550-24 和 3550-48 两款可从现场升级到 EMI 的软件映像。EMI 提供了企业级的一些高级特性，例如基于硬件的 IP 单播和组播、VLAN 之间的路由、HSRP 和许多能在路由器上找到的高级特性。性能和容量在各种型号之间也有很大变化。表 1-1 列出了 Catalyst 3550 交换机的不同型号和容量列表。

表 1-1 不同型号的 Catalyst 3550 交换机的性能特性

交换机/特性	交换矩阵	最大转发带宽 L2/3	端口缓存	64 字节转发速率	MAC 地址	吉比特端口 /GBIC 数量	10/100M 端口数量
3550-24	8.8 Gbit/s	4.4 Gbit/s	2 MB	6.6 Mpps	8000	2	24
3550-48	13.6 Gbit/s	6.8 Gbit/s	4 MB	10.1 Mpps	8000	2	48
3550-12T	24 Gbit/s	12 Gbit/s	4 MB	17 Mpps	12 000	2	16 -10/100/1000 BASE-T 端口
3550-12G	24 Gbit/s	12 Gbit/s	4 MB	17 Mpps	12 000	10	2 -10/100/1000 BASE-T 端口

* 所有的 3550 交换机都有 64MB 的动态随机访问内存 (DRAM) 和 16 MB 的闪存 (Flash)。最大传输单元 (MTU) 和单播、组播路由的数量在各种交换机类型之间也有很大不同。如想了解更多详细的信息，参看 www.cisco.com。

1.2 以太网交换机回顾

在详细讨论 3550 交换机的配置之前，我们有必要回顾一下某些重要的技术。下面几个小节简单地回顾了 VLAN、VTP、VLAN 骨干协议、生成树 802.1d 及端口自适应技术。如果你先前已经读过《CCIE 实验指南 (第 1 卷)》这本书的话，你可能只想快速地浏览这一部分，因为这一部分的目的本来就是回顾。如想更详细、更综合地理解这些以及其他的以太网交换技术，请参考《CCIE 实验指南 (第 1 卷)》。

1.2.1 虚拟局域网 (VLAN)

对于术语 VLAN 有许多不同的定义。在我们的讨论中，对它的定义是非常简单的。虚拟局域

网（VLAN）就是一个能够跨越物理距离的广播域。在虚拟局域网内部，单播、广播和组播帧都传送给那个 VLAN 内部的成员，这被称作是 VLAN 内部的流量。不同的 VLAN 成员之间是不会把流量转发给对方的，因为这可能会引起某些潜在的安全问题。如果一个 VLAN 想和另外一个 VLAN 通信，就必须使用某种类型的路由。为了用最简单的形式来表达 VLAN，记住下面的定义：

一个 VLAN = 一个广播域 = 一个三层的网络（IP 子网）

简而言之，VLAN 提供下面的功能：

- 网络分段；
- 灵活性和管理；
- 安全。

当我们配置以太网交换机时，每一个端口默认地都会分配到一个虚拟局域网中。这个默认的虚拟局域网总是 VLAN 1。当交换机出厂以后，它们具有某种程度上的“即插即用”特性。任何一个端口都被分配到 VLAN 1 中，因此，交换机所有的端口都处在一个广播域中。这使得从一个共享的以太集线器迁移到一个基本的以太交换网络非常容易。VLAN 应当永远被认为就是一个广播域。许多 VLAN 最终成为了 IP/IPX 的子网或者桥接域。应用于广播域的设计规则也同样适用于 VLAN，如下面所述：

- 每个 VLAN 应当处于一个不同的子网中。每个 VLAN 类似于一个不同的桥接域。
- 不要将不同的虚拟局域网进行桥接。
- 虚拟局域网可以跨越多个交换机和地理区域。
- 骨干链路通过使用一种特别的封装方法可以携带多个 VLAN 的流量。
- 需要一台路由器或三层交换机实现不同虚拟局域网之间的路由。
- 在每一个 VLAN 的基础上运行生成树协议可以防止环路。这个功能可以关闭掉，但是我们不推荐这样做。

表 1-2 列出了 Catalyst 交换机的不同 VLAN 默认值。

表 1-2 默认的 VLAN 设置值

特性	默认值
本征 VLAN	VLAN 1
默认 VLAN	VLAN 1
端口的 VLAN 分配	所有的端口都分配到 VLAN 1 中，令牌环的端口都分配到了 VLAN 1003 中（TrCRF 默认）
VTP 模式	Server 模式
VTP 名字	空（Null）
VLAN 状态	激活的（Active）
保留的 VLAN 范围*	VLAN 0, VLAN 1006-VLAN 1009, VLAN 4095
正常的 VLAN 范围	VLAN 2-VLAN 1001
VLAN 的扩展范围*	VLAN 1006-VLAN 4094
MTU 的尺寸	对于以太网是 1500 字节，对于令牌环是 4472 字节
SAID 的值	100000 加上 VLAN 的号码，例如： VLAN 2=SAID 100002
剪枝的可行性	VLAN 2-1000 是可以做 VLAN 剪枝的，VLAN 1025-4094 是不可以做 VLAN 剪枝的
MAC 地址的减少	没有开启
生成树的模式	PVST+（128 个生成树实例）
默认的 FDDI VLAN	VLAN 1002

续表

特性	默认值
默认的令牌环 TrCRF VLAN	VLAN 1003
默认的 FDDI Net VLAN	VLAN 1004
默认的令牌环 TrBRF VLAN	具有桥接号码 0F 的 VLAN 1005
对于 TrBRF VLAN 的生成树协议的版本	IBM
TrCRF 的桥接模式	SRB

• VLAN 的保留范围被用于 Catalyst 6000 系列的交换机上将非保留范围的 VLAN 映射到保留范围的 VLAN。VLAN 的扩展范围在 Catalyst 6000 系列交换机和 3550 系列交换机上存在。现在扩展和保留的 VLAN 范围是不会随着 VTP 的信息传播出去的，需要交换机在 VTP 透明模式下建立这些 VLAN。令牌环和 FDDI 的 VLAN 只在以太网交换机上列出，因为它是全局的 VTP 信息。

下面考虑某些基本的交换机网络，这个讨论基于各种不同的情况。

图 1-2 显示了一个基本的局域网配置。在交换机上配置了 VLAN 1 和 VLAN 2，并将不同的端口分配到了这些 VLAN 中。每一个 VLAN 都分配了一个不同的 IP 子网。如果需要将 VLAN 1 中的信息传递到 VLAN 2 中去，那么就需要一台路由器。在这里，路由器在每一个 VLAN 里面都有一个接口。从 VLAN 1 到 VLAN 2 中的流量需要先通过路由器。这种类型的配置需要路由器在每一个 VLAN 中都放置一个接口实现 VLAN 之间的路由，因此，价格非常昂贵并且扩展性很差。

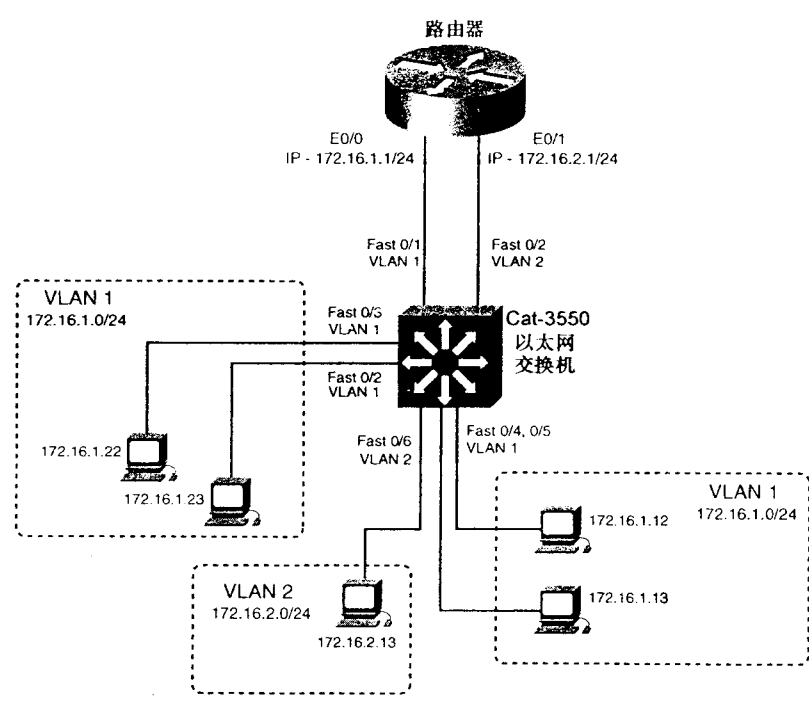


图 1-2 在每一个 VLAN 中放置一个接口的 VLAN 路由

图 1-3 显示了另外一个基本的 VLAN 配置。在交换机上一样配置了 VLAN 1 和 VLAN 2。在这里，路由器有一个单独的 100 Mbit/s 接口正在运行 VTP 协议，例如 802.1Q。从一个 VLAN 到另一个 VLAN 中的流量必须通过骨干链路到达路由器接着再返回到同一个骨干链路上。使

用一个单独的骨干链路实现 VLAN 之间的路由是一种非常经济的实现 VLAN 之间路由的方法。这种类型的配置通常被称为“单臂路由器。”

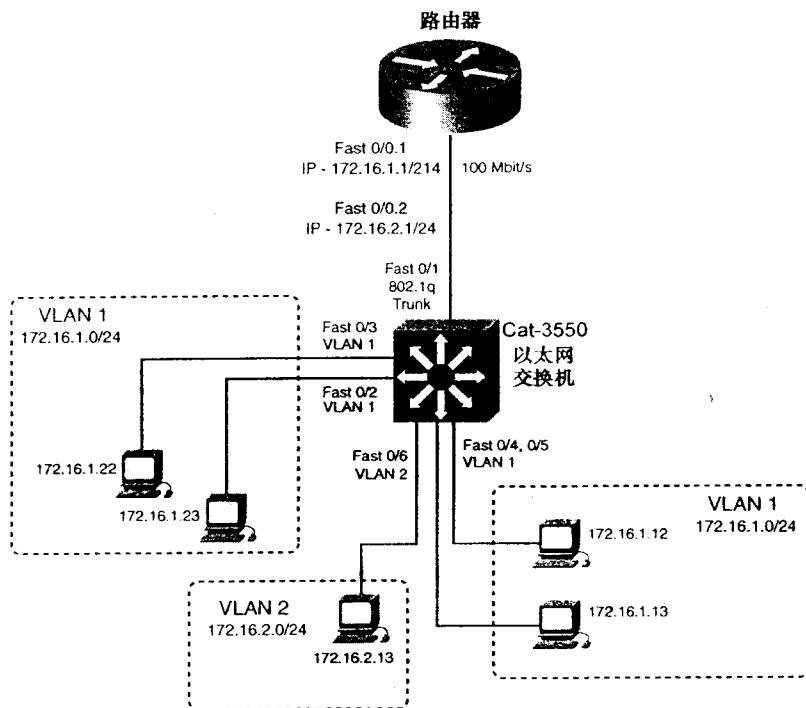


图 1-3 单臂路由器

下面的变化就是将路由功能从一个单独的路由器迁移到交换机本身。这种迁移只是逻辑上的，因为流量是在同一个接口上双倍地进入和流出。例如具有 EMI 软件的 Catalyst 3550 交换机就支持这种类型的配置。图 1-4 解释了三层交换。

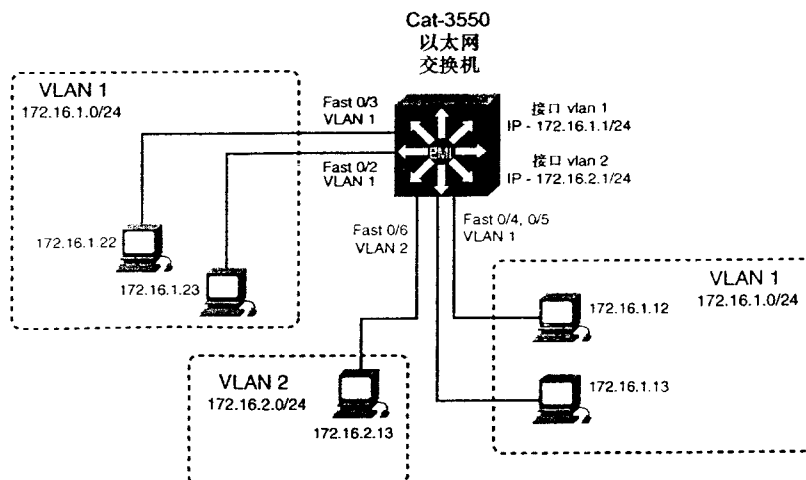


图 1-4 三层交换

1.2.2 VTP 和骨干协议

VLAN 的一个强大的功能就是它们能够跨越地理范围。一台交换机上的 VLAN 和另外一台交换机上的 VLAN 之间的通信是通过一种叫作 VLAN 骨干 (VLAN Trunking) 的协议进行通信的。VTP 维护交换机之间全局 VLAN 信息的一致性。这包括同步 VLAN 数据库，以及 VLAN 的名字在网络上进行添加、修改和删除的管理等。一个 VLAN 的管理域，或者是 VTP 域，包含一个或者多个互连的交换机，共享一个管理域的职责。任何时候，假如你想让一台交换机上的 VLAN 具有另外一台交换机上的 VLAN 信息，那么必须配置一个 VTP 域和一条骨干链路。VTP 也会跟踪一个 VTP 域里的所有 VLAN，并且以客户机/服务器的模式将 VLAN 的信息从一台交换机传播到另外一台交换机上去。VTP 的目的就是简化管理并且在 VTP 域里提供一个共同的 VLAN 数据库。VTP 的高级功能还包括 VTP 剪枝，这个功能可帮助控制在交换机之间和 VLAN 之间的广播流量。

VTP 操作有如下三种模式：

- **VTP 服务器模式**——在 VTP 服务器模式下，可以建立、修改和删除 VLAN。在同一个 VTP 域里，VLAN 信息可以自动地传送到所有相邻的 VTP 服务器和客户交换机中。一定要特别注意，当从 VTP 服务器上清除一个 VLAN 时，后果是那个 VLAN 将会从这个 VTP 域中的所有 VTP 服务器交换机和客户交换机上删除。如果有两台交换机都配置为服务器模式，那么具有最高的 VTP 配置版本号并且具有服务器模式的交换机将充当主服务器。VLAN 信息保存在交换机的非易失性随机访问内存 (NVRAM) 里。
- **VTP 客户模式**——在 VTP 客户模式下，不能进行 VLAN 的建立、修改和删除。只有 VTP 的名字和 VTP 的模式及剪枝功能可以修改。客户模式的交换机只能从 VTP 服务器模式的交换机处得到所有的 VLAN 信息。客户模式的交换机还必须把端口分配到 VLAN 中，这个 VLAN 只有当 VTP 服务器模式的交换机将那个 VLAN 的信息传送到客户模式的交换机后才能在客户模式的交换机上被激活。在 Catalyst 2900XL/3500XL/3550 系列的交换机上，当从服务器模式的交换机上收到 VLAN 信息后，这个 VLAN 信息存储在闪存的 VLAN.DAT 文件里。Catalyst 4000/5500/6500 系列的交换机在 VTP 客户模式下不保存 VLAN 数据库。
- **VTP 透明模式**——在 VTP 透明模式下，在交换机上建立的 VLAN 信息是本地的，这个 VLAN 信息交换机不会通告出去，VTP 不会在交换机之间同步 VLAN 数据库。但是如果所有的交换机在同一个 VTP 域里，那么从其他交换机收到的 VTP 信息会被转发出去。为了使得 VTP 更新数据通过一个 VTP 透明模式的交换机转发出去，这个透明模式的交换机必须和其他客户模式或服务器模式的交换机处于同一个 VTP 域里。可以在透明模式的交换机上建立、修改和删除 VLAN。透明模式的交换机同时也支持扩展范围的 VLAN。事实上，1006 到 4094 范围的 VLAN 只能在透明模式的交换机上建立，VTP 不会将这个范围内的 VLAN 传播出去。在 Catalyst 2900XL/3500XL/3550 系列的透明模式交换机上，VLAN 信息保存在闪存的 VLAN.DAT 文件里。表 1-3 高度概括了不同的 VTP 模式和操作。

表 1-3 不同 VTP 模式的操作特性

VTP 模式	源 VTP 信息	传播本地 VLAN 数据库	侦听 VTP 信息	建立、修改和删除 VLAN	被 VTP 服务器同步 VLAN 数据库	VLAN 数据库保存在 NVRAM
服务器	是	是	是	是	是	是
客户	是	N/A	是	不是	是	是/不是*
透明	不是**	不是	是**	是	不是	是

* Catalyst 4000/5500/6500 系列交换机不会在 VTP 客户模式的交换机上保存 VLAN 数据库。Catalyst 2900XL/3500G/3550 系列的交换机将 VTP 和 VLAN 信息保存在内存的 VLAN.DAT 文件中。交换机一旦初始化就会从 VLAN 数据库中得到 VLAN 信息。

** 在透明模式下，交换机不会参与 VTP 协议，也就是说，它不会被 VLAN 数据库同步。然而，收到的 VTP 信息会沿着其他骨干链路转发出去，骨干链路也不会传播本征 VLAN 信息。

图 1-5 解释了 VTP 信息是如何通过局域网传播出去的。

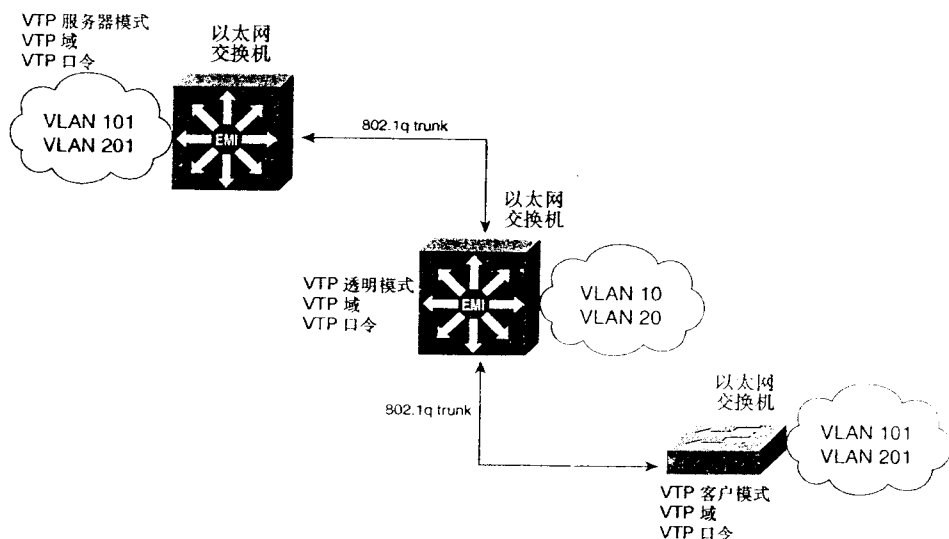


图 1-5 VTP 模式和传播

VTP 通告在所有的骨干链路上发送出去，是以 ISL(交换机之间的链路)的帧格式、802.1Q 的帧格式、IEEE 802.10 的帧格式或者 ATM 局域网仿真 (LANE) 的信元格式发送的。VTP 的帧发送到目的 MAC 地址为 0100.0ccc.cccc 的地址，逻辑链路控制代码为 SNAP 的(AAAA)。IEEE 802.1Q 的帧具有代码为 0x8100 的以太类型字段。VTP 通告每隔 5min 发送一次，或是在 VLAN 数据库有变化的时候立刻发送。为了使 VTP 的信息成功传输出去，必须满足下列条件：

- **VTP 域名**——VTP 服务器模式和客户模式的交换机只接受来自相同域名的 VTP 信息。如果那个 VTP 域配置了认证，VTP 的口令也必须相匹配。VTP 的域名和 VTP 的口令是大小写敏感的。
- **VTP 的版本模式必须匹配**——VTP 只接受具有相同版本的信息：版本 I 或者版本 II。VTP 的版本是由在骨干链路上启用或关闭 V2 模式来控制的。一个交换机可能具有 VTP 版本 II 的能力，但是 V2 模式是关闭的，这是交换机的默认设置。V2 模式只对令牌环交换机使用，因此，在安装了令牌环交换模块的 Catalyst 3924 和 Catalyst 5500/6500 系列的交换机上可以看到这种模式设置的情况。

- **VTP 客户模式的交换机只有其 VTP 信息的修订号码低于 VTP 服务器的修订号时，其 VLAN 数据库才会被同步**——如果 VTP 客户的修订号等于或大于 VTP 服务器的修订号，那么 VLAN 数据库不会被同步，因此 VTP 客户交换机不会接收来自 VTP 服务器的任何 VLAN 信息。

当建立骨干链路时，VTP 将会在每一个骨干端口上发送周期性的通告，每隔 5min 或当 VLAN 数据库有任何变化时就立刻发送这个通告。VTP 通告含有下面的信息：

- VLAN ID（ISL 和 802.1Q）。
- 对于 ATM LANE 的仿真局域网名。
- 802.10 SAID 值。
- VTP 域名和配置修订号码。具有最高修订号的服务器模式的交换机将会成为主服务器，并将它的 VLAN 数据库发送到其他交换机。这种进程被称为同步。当 VTP 同步产生后，所有 VTP 服务器和客户模式的交换机将含有相同的 VTP 修订号。当 VLAN 配置每次发生改变后，VTP 的修订号都会增加。
- VLAN 配置、VLAN ID、VLAN 名字和对于每一个 VLAN 的 MTU 大小。
- 以太的帧格式。

VTP 有两种版本：版本 I 和版本 II。在一个 VTP 域里的所有交换机必须具有相同的版本号。这个规则不适用于透明模式的交换机。VTP 版本 II 为令牌环提供了下面这些最重要的支持特性：

- **令牌环支持**——VTP 版本 II 支持令牌环局域网交换和 VLAN（令牌环桥接中继功能 [TrBRF]）。
- **不可识别的类型长度值（TLV）**——当交换机处于 VTP 服务器模式时，不可识别的 TLV 保存在 NVRAM 中。
- **与版本相关的透明模式**——当交换机操作在 VTP 透明模式版本 II 中时，VTP 将会转发和它的域名和版本号不匹配的 VTP 信息。在透明模式版本 I 里，VTP 会检查帧中的版本号，如果版本号匹配了，VTP 就会转发这个帧。这种检测进程不会发生在 VTP 版本 II 里。
- **一致性检查**——当 VLAN 信息是从命令行接口或简单网络管理协议改变时，需要在 VLAN 的名字和号码上作一致性检查。

表 1-4 列出了在 Catalyst 3550 交换机上的默认 VTP 设置。

表 1-4 在 Catalyst 3550 交换机上的默认 VTP 设置

VTP 特性	默认设置	VTP 特性	默认设置
VTP 域名	null	VTP 安全/口令	Disabled
VTP 模式	server	VTP 剪枝	Disabled
VTP 版本 2 的更新	Disabled	VLAN 骨干协议	DTP

一、VTP 剪枝

VTP 剪枝本质上是用来控制广播、组播和未知目的 MAC 地址的单播流量在不需要时通过骨干链路。常见的关于 VTP 剪枝的错误说法是 VTP 剪枝可以控制生成树协议的流量（STP），而实际情况不是这样。在 3550 交换机上的默认设置中 VTP 剪枝是关闭掉的，所有的广播、

组播和未知目的 MAC 地址的单播流量通过交换机的骨干链路传送到下游交换机上，无论这个下游交换机需要还是丢弃传送过来的流量。VTP 剪枝本质上只有在下游交换机有一个活动的端口所属的 VLAN 和始发流量的 VLAN 是同一个 VLAN 时，它才沿着骨干链路转发广播、组播和未知目的 MAC 地址的单播流量。如果目的交换机不是直接相邻，在源和目的交换机之间的交换机就会接收和转发这个流量。在图 1-6 中，在 VLAN 10 中的一个工作站发送了广播包，而由于交换机的剪枝被关闭掉了，所以局域网中的所有交换机都会接收到那个广播。

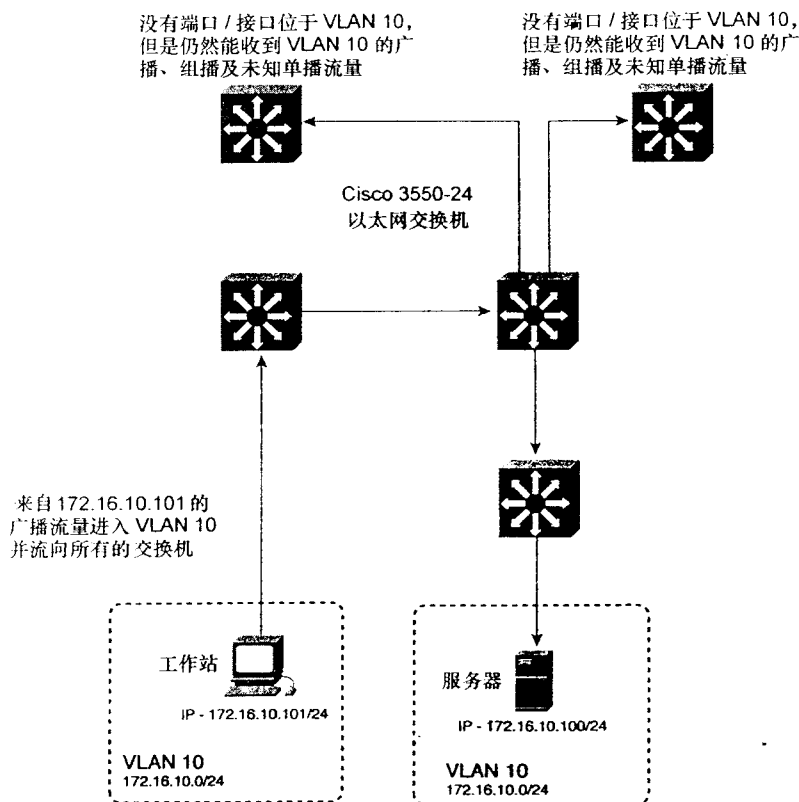


图 1-6 VTP 剪枝

在图 1-7 中，启用了 VTP 剪枝。由于启用了 VTP 剪枝，只有端口/接口在 VLAN 10 中的交换机会接收和转发 VLAN 10 的流量，沿路的交换机也是同样的行为。

二、VLAN 骨干协议

VTP 协议需要骨干链路传输 VTP 信息。骨干链路被认为是以太网交换机端口和其他网络设备之间的一条点对点链路，例如一台路由器或其他的交换机。骨干链路具有在一条链路上携带多个 VLAN 流量的能力，并且可以使 VLAN 跨越互连网络。如果不使用 VTP 和骨干协议，那么一个 IP 子网将永远不会跨过多个交换机，在物理位置上被分离。VTP 骨干提供了一种有效的方法将跨越地理区域的两个广播域连接起来。图 1-8 解释了 802.1Q 骨干是如何将 VLAN 2 和 VLAN 4 连接的。

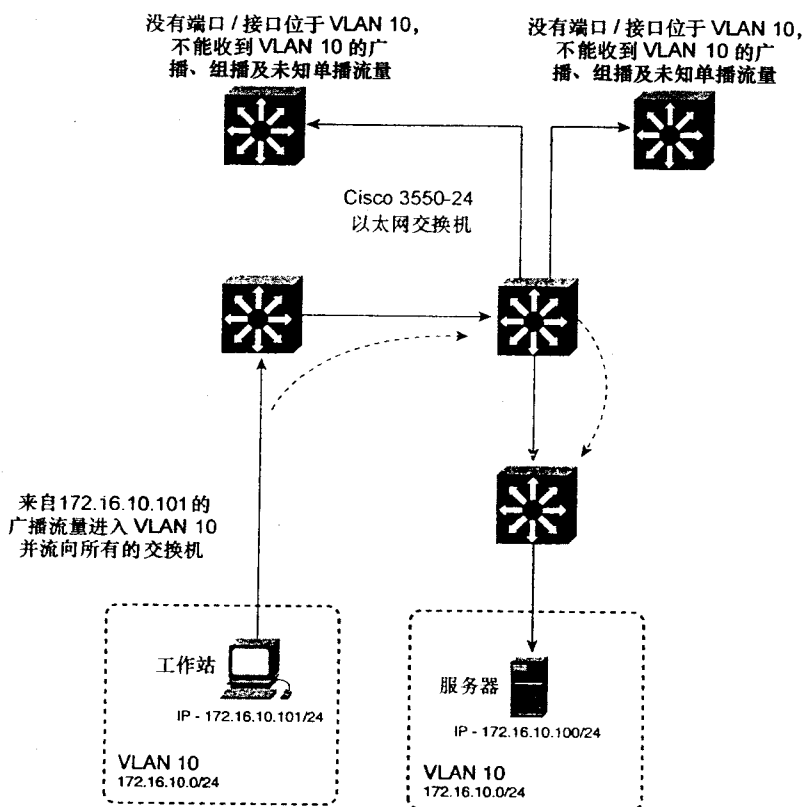


图 1-7 VTP 剪枝

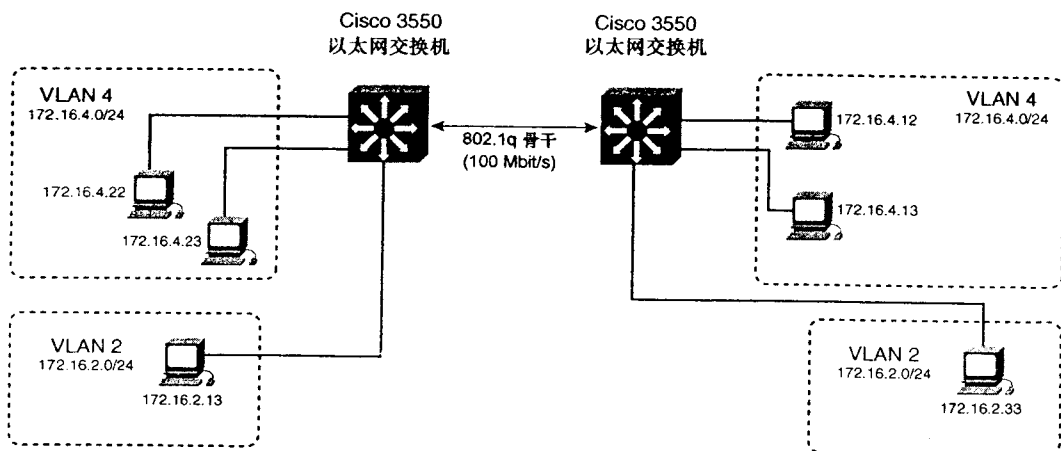


图 1-8 VLAN 骨干

在思科 Catalyst 交换机的家族系列上支持三种类型的封装形式：ISL、802.1Q 和 802.10。Catalyst 3550 以太网交换机支持 ISL 和 802.1Q，因此，下面集中讨论这些交换机。

- 交换机之间的链路（ISL）——ISL 是思科专有的骨干封装协议。ISL 是一个给帧打

为 ISL 骨干链路的端口在将每一个帧从骨干端口传送出去之前，会将每一个帧封装 26 字节的 ISL 报头，在帧尾封装一个 4 字节的循环冗余校验码（CRC）。对每一个帧的封装是一个低延时的过程。这个操作是由特定应用的集成电路芯片（ASIC）来完成的，所以速度特别快，被称为“线速”。在这条链路上的帧包括标准的以太网帧、FDDI 或令牌环的帧，以及和这个帧相关的 VLAN 信息，还有桥接数据包数据单元（BPDU）。ISL 需要在 100 Mbit/s 或更高速率的链路上支持，而且它可以支持全双工和半双工。在 ISL 骨干链路上的 STP 是基于每一个 VLAN 实现的，也称为 PVST+。这也就意味着每一个 VLAN 有自己的根桥，骨干链路决定每一个 VLAN 最终进入转发还是阻塞状态。PVST+ 在大型网络上进行控制是非常重要的，正如本章后面所讨论的。

- **IEEE 802.1Q**——802.1Q 是一个工业标准的骨干协议。802.1Q 帧使用一个以太网类型代码 0x8100，将 VLAN 信息插入到帧中，并且在帧的末尾重新计算帧的校验和。802.1Q 和 ISL 稍微有些不同。例如，它对于 VTP 域中的所有 VLAN 在本征 VLAN 上运行单一的生成树协议。802.1Q 的本征 VLAN 默认使用的是 VLAN 1。在单一生成树协议中，为整个 VTP 域选择一个根桥，这也被称为通用生成树协议（CST）。在这种类型的配置中，所有的 VLAN 流量遵循一条路径。思科明白在大型网络中控制生成树协议而同时又要控制负荷的需求，所以在所有的 802.1Q VLAN 上除了实施单一生成树协议外，还实施了 PVST+。下列限制适用于 802.1Q 骨干：

- 本征 VLAN 需要在骨干链路两端的交换机上配置为相同的。单一生成树协议就运行在这个 VLAN 里。非常关键的是本征 VLAN 需要在第三方交换机和思科交换机上配置一致。
- 正如前面所谈到的，802.1Q 使用单一生成树协议。思科使用 PVST+ 增强了这个特性。因为思科和第三方交换机对于 BPDU 的处理方式是不同的，因此将不同厂商的交换机集成在一个域里时，要特别注意生成树协议和默认 VLAN 在两台交换机上一定要是一致的。整个非思科的域对于思科的 PVST+ 的 VTP 域看起来就像一个单独的广播域/生成树域。非思科域的单一生成树将会映射到思科域的 CST，它默认使用的是思科的 VLAN 1。
- 在骨干链路上的本征 VLAN 发送的 BPDU 是不打标签的，发送到保留的 IEEE 802.1d 的生成树组播 MAC 地址（0180.c200.0000）。在骨干链路上的其他 VLAN 的 BPDU 是以打了标签的方式发送的，目的地址为保留的思科共享生成树（SSTP）组播 MAC 地址（0100.0ccc.cccc）。

动态 ISL（DISL）和动态骨干协议（DTP）

动态 ISL 是思科的第一个骨干协商协议。DISL 在新版本 CAT OS 和思科的 IOS 软件中已逐渐被动态骨干协议（DTP）所替代。DTP 本质上就是 DISL，它试图自动化 ISL 和 802.1Q 的骨干配置。对于局域网网络，DTP 使用保留的目的 MAC 地址 0100.0ccc.cccc 来协商骨干链路。在默认的“自动”状态下，DTP 信息每隔 30s 在所有的骨干链路上发送。取决于端口的模式，端口可能成为 ISL 或 802.1Q 骨干。DTP 操作模式如下所示（注意：这些模式不是在所有的交换机上都存在并且在不同的交换机上略有不同）。

- **On**——将端口设置成永久的骨干状态。它也试图将链路协商成骨干链路。
- **Off**——将端口设置成非骨干链路，因此把骨干的功能关闭掉了。
- **Desirable**——将端口试图转换成骨干链路。如果邻居端口被设置成 on、desirable 或者 auto 模式，那么这个端口就会成为骨干端口。
- **Auto**——如果邻居端口被设置成 on 或 desirable 模式，那么这个端口就会成为一个骨干端口。
- **Nonegotiate**——将这个端口设置成骨干模式，但是阻止这个端口发送 DTP 帧。

实际上，对于骨干链路来说有许多选项。网络管理员要么将端口配置成骨干链路，要么不是。现在甚至有争论说启用动态骨干协议有潜在的安全问题。表 1-5 列出了在 CAT OS 交换机上骨干的可能组合模式。正如你将要看到的，配置一个骨干的最可靠和最简单的方法就是静态地将链路的两端配置为骨干，模式为 On 模式。

表 1-5 在 CAT OS 中的以太 DTP 配置结果

邻居端口	骨干模式和骨干封装	Off	On	Desirable	Auto	On	Desirable	Auto	Desirable	Auto
Off	ISL 或 DOT1Q	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：1Q 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干
On	ISL	本地：非骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：1Q 骨干 邻居：ISL 骨干	本地：非骨干 邻居：ISL 骨干	本地：非骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干
Desirable	ISL	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：1Q 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：ISL 邻居 邻居：ISL	本地：ISL 邻居 邻居：ISL
Auto	ISL	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：非骨干 邻居：非骨干	本地：1Q 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：ISL 邻居 邻居：ISL	本地：非骨干 邻居：非骨干
On	DOT1Q	本地：非骨干 邻居：1Q 骨干	本地：ISL 骨干 邻居：1Q 骨干	本地：非骨干 邻居：1Q 骨干	本地：非骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干
Desirable	DOT1Q	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干
Auto	DOT1Q	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：非骨干 邻居：非骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：非骨干 邻居：非骨干
Desirable	Negotiate	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：1Q 骨干 邻居：1Q 骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 邻居：ISL	本地：ISL 邻居 邻居：ISL
Auto	Negotiate	本地：非骨干 邻居：非骨干	本地：ISL 骨干 邻居：ISL 骨干	本地：ISL 邻居 邻居：ISL 邻居	本地：非骨干 邻居：非骨干	本地：1Q 骨干 邻居：非骨干	本地：1Q 骨干 邻居：非骨干	本地：非骨干 邻居：非骨干	本地：ISL 邻居 邻居：ISL 邻居	本地：非骨干 邻居：非骨干

三、二层和三层的以太通道骨干

EtherChannel 将多个物理快速以太接口或者是吉比特端口/接口合并成一个逻辑上的接口，称为一个通道组。例如最多 8 个快速以太端口/接口可以被组合在一起提供全双工的 1600 Mbit/s 的逻辑链路。吉比特的 EtherChannel 可以组合最多 8 个端口提供全双工的 16 Gbit/s 的总速率。

注意：GigaStack（堆叠）吉比特以太模块不能用作吉比特 EtherChannel 骨干。

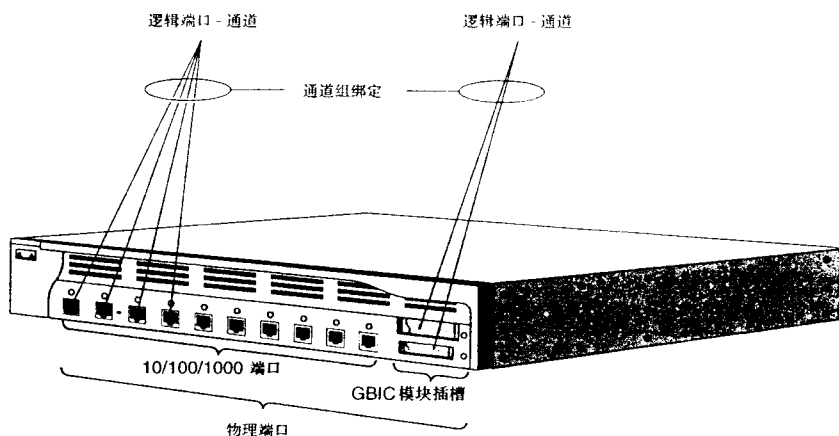


图 1-9 物理、逻辑和通道组的关系

EtherChannel 可以视为将思科的交换机作骨干的一种很好的替代方案。它优于通常的多链路骨干的一个优点就是 STP 会将多条链路看作是一条链路，而不会将任何一个端口阻塞掉而导致浪费带宽。传统上，VLAN 的流量做负载均衡很难实现，而且带宽很有限，这是因为 STP 会将冗余的端口阻塞掉。发生链路故障后，STP 将不得不等待一个默认的 50s 时间使其收敛。EtherChannel 可以在组成一个通道组的多条路径上作负载均衡，如果一条物理链路失效了，通道组只会失去那条链路的带宽。通道组在核心交换机之间特别有用。图 1-10 演示了两个思科 Catalyst 3550 交换机充当核心交换机，将吉比特以太接口组成了一个吉比特的通道组。

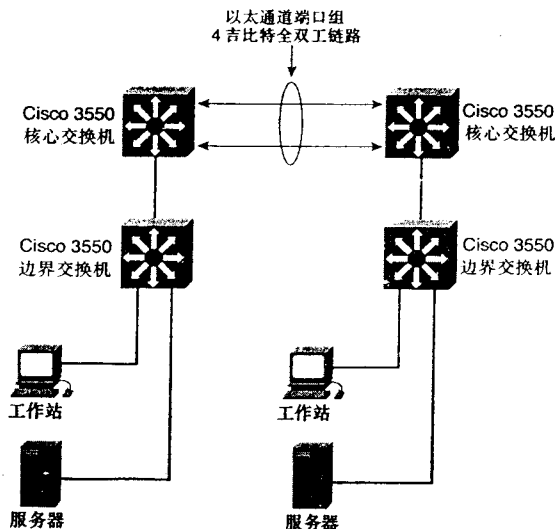


图 1-10 在 Catalyst 3550 上的吉比特以太通道组

可以放入一个以太通道组的接口的数量和类型随交换机的不同而不同。一个共知的规则就是只能将快速以太端口组成快速以太通道组，吉比特端口组成吉比特以太通道组。这是因为对特定的交换机存在特殊的规则。关于什么端口、多少个端口可以放入一个通道组，请到 www.cisco.com 去查看思科关于当前正在配置的交换机在以太通道组方面的限制。

1. 端口聚合协议 (PAgP) 和链路聚合协议 (LACP)

以太通道组使用一种叫作端口聚合的协议在相邻的交换机之间动态协商以太通道组。思科定义的 PAgP 和运行方式如下：

端口聚合协议可以帮助自动建立以太通道组。通过使用 PAgP，交换机可以获悉邻居交换机是否支持 PAgP，并且可以获悉每一个端口的能力。它接着可以动态地将具有相同配置的接口放入一个单独的逻辑链路（通道或者聚合端口），这些被组合的端口基于硬件、管理和端口参数的限制等因素被放在一起。例如，PAgP 可以根据接口的速率、双工模式、本征 VLAN、VLAN 的范围和骨干的状态及类型来决定这些接口是否被组合在一起。当把链路组合成以太通道组后，PAgP 将此组添加到生成树协议中，被当作一个交换端口看待。

出于这些原因，所以重要的是我们应当将具有相同的物理 VLAN 和 STP 参数的那些端口放入一个通道组中。

PAgP 和 LACP 共同工作协商以太通道骨干。LACP 在 IEEE 802.3AD 中定义，它允许思科交换机管理遵循 802.3AD 协议的交换机之间的以太通道组。

2. 端口聚合协议 (PAgP) 的模式

PAgP 在 CAT OS 中有 4 种模式，在思科 IOS 软件中有 6 种模式：

- **Auto**——Auto 模式将端口置为被动的协商状态，这个端口相应接收到 PAgP 的帧，但是从不主动发起 PAgP 的协商。这个设置是默认的，可以最小化 PAgP 的传输。
- **Desirable**——Desirable 模式将一个端口置为活动的协商状态，通过发送 PAgP 包主动与其他端口发起协商。
- **On**——On 强制这个端口进入通道组而无需 PAgP 或 LACP 协议。在 on 模式下，一个通道组只有在这个接口组的模式是 on 模式，它所连接的另外一个接口组的状态也是 on 模式时才是可用的。一个为 on 模式的加入通道组的端口会和这个通道组中已经存在的为 on 模式的端口具有相同的特性。
- **Off**——在这种模式下，这个端口不会进入通道组，不会和对端交换 PAgP 的帧。
- **Active (LACP) -IOS only**——将接口置为主动协商状态，在这种状态下，接口会发送 LACP 的包主动和其他接口发起协商。
- **Passive (LACP) -IOS only**——将接口置为被动的协商状态。在这种模式下，接口会相应收到 LACP 的包，但是它不会主动发起 LACP 的数据包协商。这种设置可以最小化 LACP 数据包。

交换机的接口只有在对方的接口配置为 **auto** 或者 **desirable** 模式时才和对方交换 PAgP 数据包。配置为 **on** 模式的接口不和对方交换 PAgP 帧。接口处于不同的 PAgP 模式是可以形成通道组的，但是前提是 PAgP 的模式必须是兼容的。例如，一个处于 **desirable** 模式的接口

可以和处于 **desirable** 或者 **auto** 模式的另外一个接口形成通道组。然而，一个处于 **auto** 模式的接口不会和另外一个也处于 **auto** 模式的端口形成通道组，这是因为任何一个端口都不会主动发起 PAgP 数据包的协商。

如果交换机连接的对端设备也具有 PAgP 能力，那么可以将交换机的端口配置为非安静模式。这是通过使用 **non-silent** 关键字实现的。如果在 **auto** 或者 **desirable** 模式中没有指定 **non-silent** 关键字，那么默认的就是安静模式。

3. PAgP 物理端口学习和聚合端口学习

网络设备可以划分成两大类：PAgP 物理端口学习者和聚合端口学习者。如果一个设备是通过物理端口学习地址并且根据那个学习到的地址转发流量，就被称为物理端口学习者。如果一个设备是通过聚合端口（逻辑端口）学习地址，就被称为聚合端口。

当一个设备和它的对端设备都是聚合端口的学习者时，它们都是通过逻辑端口通道学习地址。这个设备通过以太通道组中的任何一个接口传输到源端的数据帧。

PAgP 不能自动检测对端设备是物理还是聚合端口学习者。必须在本端手动设置基于源的分发学习方法，通过 **pagp learn-method src-mac** 接口配置命令来实现。当使用基于源的分发方法时，任何给定的源 MAC 地址的数据帧都会从相同的物理端口发送。

某些以太通道的特性和限制如下：

- 放入一个通道组中的接口数量是和交换机的硬件紧密相关的。确保检查思科的站点 www.cisco.com 来了解最新的软件和硬件限制。
- 动态骨干协议（DTP）、VTP 和思科发现协议（CDP）可以通过通道组中的物理接口发送和接收数据帧。骨干端口发送和接收 PAgP 协议数据单元（PDU）是通过最低号码的 VLAN 实现的。
- STP 通过以太通道组中的第一个接口发送数据帧。STP 将整个通道组看作是一条物理链路。
- 三层以太通道组的 MAC 地址用的是通道组中第一个接口的 MAC 地址。
- PAgP 只从 up 的接口，并且启用了 PAgP 功能的 **auto** 或者 **desirable** 模式的接口传送和接收 PAgP PDU 数据包，静态配置骨干端口关闭了 PAgP 功能。
- 具有不同的 GARP VLAN 注册协议（GVRP）、GARP 组播注册协议（GMRP）和服务质量配置的端口不能组成以太通道。
- 不能在以太通道的端口上启用端口安全。
- 如果端口是交换机端口分析器的目的端口（SPAN）的话，那么不能形成以太通道。可以使用以太通道组作为 SPAN 的来源来监控整个通道组的流量。
- 速率、双工模式、本征 VLAN、VLAN 范围和骨干类型（如果你正在通过以太通道建立骨干）必须在通道链路的两端匹配。

四、三层的以太通道

三层的以太通道是在交换机的路由端口上配置以太通道。会给以太通道组分配一个惟一的 IP 地址，而且端口必须用接口命令 **no switchport** 将端口的交换功能关掉。从本质来说，三层的以太通道和二层的以太通道运行的功能是完全一样的，三层的以太通道只有在交换机上安装了 EMI 软件后才可使用。

1.2.3 以太网物理特性：半双工和全双工以太网

半双工从本质上讲和以太网的载波侦听、冲突检测和多路访问（CSMA/CD）模式工作原理是一样的。以太网中的集线器就是半双工的一个典型设备。半双工以太网具有下面的特点：

- 单向数据流；
- 极高的冲突率；
- 在共享介质的设备例如集线器或工作站上工作；
- 整个链路带宽的有效利用率是 50%~60%。

全双工的以太网允许一个工作站同时发送和接收数据。以太网在双绞线的两对线缆之间或者是光纤的一对线缆之间同时发送和接收。全双工以太网本质上就是不带 CSMA/CD 功能的以太网协议。全双工模式本质上产生双倍的以太网带宽。为了运行全双工模式的以太网协议，链路两端的以太网设备必须都具有自适应的能力或者配置为全双工模式。图 1-11 演示了一个通常的以太网络和链路两端的双工设置。

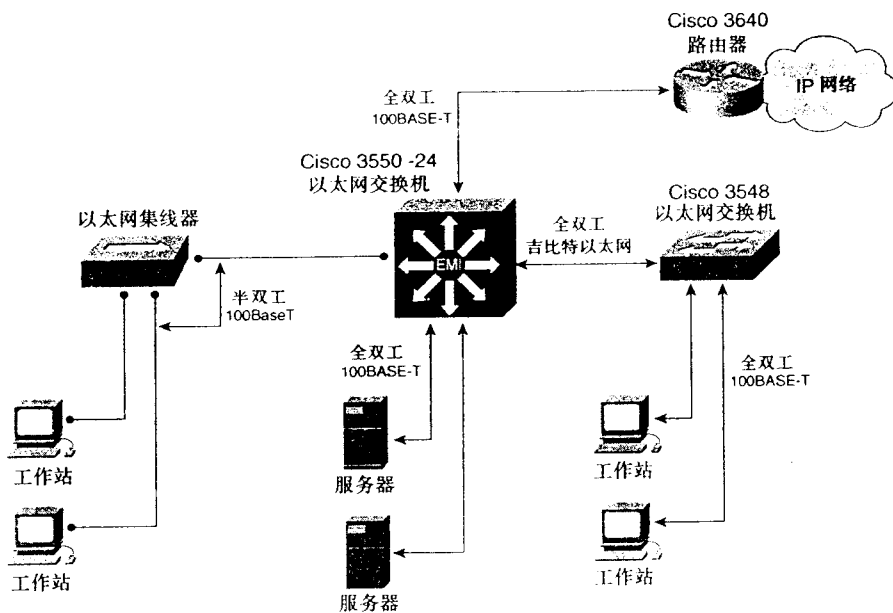


图 1-11 以太网双工设置

注意：没有在正常的双工模式下运行的工作站将会在所连接的端口上产生大量的冲突帧或者循环冗余校验码（FCS）错误。这种冲突被称为“滞后冲突（late collision）”。确保交换机的端口和终端工作站运行在相同的双工模式下。

以太网自适应

为了简化对以太网设备的配置，IEEE 委员会对 10BASE-T 网络定义了正常的链路脉冲（NLP），对 100BASE-T 和 1000BASE-T 网络定义了快速链路脉冲（FLP）。NLP 和 FLP 是网

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

络上的一系列脉冲，可以推论在当前的链路下运行的速率和双工模式。工作站和集线器/交换机协商一个最高的优先级，并以那种方式配置工作站。所有的自适应都发生在物理层。表 1-6 列出了 FLP 使用的优先级和相关的数据传输速率。为了确保自适应正常工作，两端的设备必须都支持自适应的功能。

表 1-6 以太网自适应优先级

优先级	整个数据传输速率 (Mbit/s)	速率和双工设置	优先级	整个数据传输速率 (Mbit/s)	速率和双工设置
1 (最高)	2000	1000BASE-T 全双工	6	100	100BASE-T4 半双工
2	1000	1000BASE-T 半双工	7	100	100BASE-TX 半双工
3	200	100BASE-T2 全双工	8	20	10BASE-T 全双工
4	200	100BASE-TX 全双工	9 (最低)	10	10BASE-T 半双工
5	100	100BASE-T2 半双工			

网络基础设施中的设备（例如路由器和交换机）应当永远将速率和双工模式设置为固定的。许多 100 Mbit/s 或者速率更高的网络接口卡（NIC）都支持全双工。运行在全双工情况下，本质上产生双倍的以太网带宽。充分利用这个功能就是最便宜的对网络进行升级的方式。

注意：双工模式是内置在网卡中的硬件功能。软件升级不会使你运行全双工模式。为了确保全双工模式正常工作，两台工作站都必须具备全双工功能。

1.3 IEEE 802.1d 生成树协议（STP）

随着以太从一条单独的共享线缆发展到具有多个桥和集线器的网络，需要一种环路检测和防御协议。由 Radia Perlman 开发的 802.1d 协议提供了这种环路检测功能。事实上，生成树协议是如此之好，导致当许多网络从桥接网络过渡到交换网络时，生成树协议的重要性几乎被忘记了。STP 在冗余的交换网络中可以极好地防止环路的发生。对许多网络工程师来说，这个协议运行在网络的后端而无需手工配置。因为这一点，生成树协议可能是现代交换局域网中最有用但是最不需要理解的协议了。在过去的一些年里，局域网已经从 IEEE 802.1d STP 过渡到 IEEE 802.1w 快速 STP。IEEE 802.1w 网络具有非常快的收敛速度，它使用了由思科系统最早开发的一些概念，例如端口加速、上行链路加速和骨干加速。本节集中介绍 IEEE 802.1d STP，IEEE 802.1w 和 IEEE 802.1s 在随后的章节中进行介绍。

1.3.1 生成树的操作

生成树的主要目的是选举一个根桥，对网络中所有的桥都构造一个无环的路径指向根桥。当生成树收敛完成后，网络中的每一个桥对于它的桥接端口都会是两种状态之一：转发或者是阻塞。STP 通过发送一种特殊的称为桥协议数据单元（BPDU）的消息报文来完成这个工作。802.1d 使用两种类型的 BPDU：

- 一个配置 BPDU，主要用于初始的 STP 配置；

- 一个拓扑变化通知 (TCN) BPDUs，主要用于拓扑变化。

BPDUs 传输时使用的是一个分配给“所有的桥”的保留的组播地址。BPDU 从所有桥接的局域网端口中发送出去，并且被局域网中所有的桥接收。BPDU 不会被路由器转发出局域网。

BPDU 含有下列相关的信息：

- **Root ID (根 ID)** ——这是充当根桥的桥 ID。一旦初始化，每一个桥都认为自己就是根桥。
- **Transmitting bridge ID (BID) and port ID (传输桥的 ID 和端口 ID)** ——这是传输 BPDU 数据包的桥 ID 和发送 BPDU 数据包的端口。
- **Cost to root (到达根桥的费用值)** ——这是从传输 BPDU 数据包的那个桥到达根桥的最低费用的路径。一旦初始化，因为每一个桥认为自己就是根桥，它会传输一个 0 的值代表到达根桥的费用。
- **Other STP information and timers (其他的 STP 信息和计时器)** ——完整的 802.1d 的帧会在以后的图 1-26 中进行解释。在这里你会看到 3 个 STP 的计时器及其他的 STP 信息被列出来。

一、桥 ID

BID 是一个 8 字节的字段，由 6 字节的 MAC 地址和 2 字节的桥的优先级组成。BID 中的 MAC 地址由一系列的因素生成，取决于桥所使用的硬件。路由器使用一个物理地址，而交换机使用的地址来自于背板或者主控模块。图 1-12 解释了 BID。优先级的值从 0~65535，默认的值是 32768。

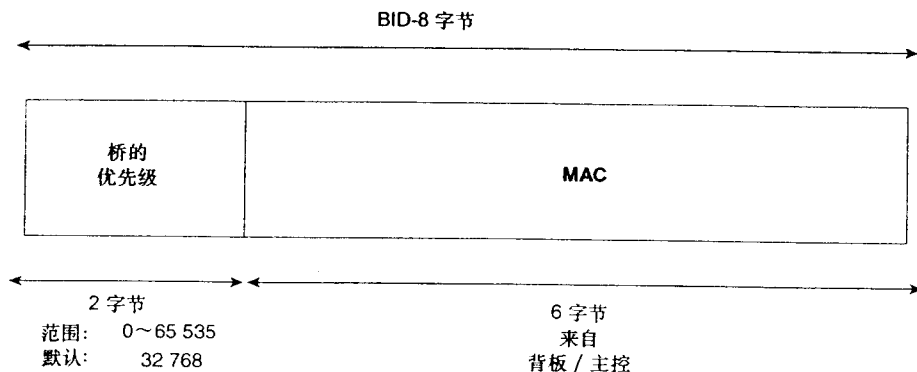


图 1-12 桥 ID (BID)

二、扩展的系统 ID 和 IEEE 802.1T

IEEE 802.1T 生成树协议的扩展注重的是优先级的值实在是太大了这个事实。802.1T 通过使用扩展的系统 ID 修正了这个问题。建立扩展的系统 ID 的部分原因就是保留 MAC 地址。IEEE 802.1d 标准要求每一个桥/交换机都有一个不同的 BID。在 PVST+ 中，每一个 VLAN 都需要一个不同的 BID，因此当在同一个交换机上面配置 VLAN 的时候就会有许多的不同的 BID。这就导致了在一个交换机上能够配置的 STP 实例的数量是有限制的。STP 使

用扩展的系统 ID、交换机的优先级和分配的 STP MAC 地址使得每一个 VLAN 都有一个不同的 BID。

在 12.1 (8) EA1 及其以后的 IOS 版本中，Catalyst 3550 交换机支持 802.1T 生成树扩展，而且先前某些用于优先级的位现在被扩展的系统 ID 所替代，它的设置就等同于 VLAN 识别符。结果就是很少的 MAC 地址保留用于交换机，可以支持很大范围的 VLAN ID，而同时维护单一的 BID 位。表 1-7 解释了交换机的优先级数值和扩展的系统 ID。

表 1-7 交换机的优先级数值和扩展的系统 ID

交换机的优先级数值				扩展的系统 ID (设置为等同于 VLAN ID)											
位															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
32 768	16 384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

从表 1-7 中可以看出，先前用于交换机优先级的 2 字节现在被重新分配变成 4 位优先级的数值和 12 位的扩展系统 ID，这个扩展 ID 等同于 VLAN ID。为了配置交换机使用扩展系统 ID，使用下面的全局配置命令：

```
3550_switch(config)#spanning-tree extend system-id
```

扩展的系统 ID 默认是在 Catalyst 3550 系列交换机上启用的。如果你的交换机使用扩展系统 ID，要特别注意当使用 **show spanning-tree summary** 命令时，这个数值将会出现在显示结果的配置列表中。

三、STP 路径费用

桥使用 STP 路径费用值来决定到达根桥的最优路径。路径费用值最近已经被 IEEE 更新，包含了吉比特甚至更高的速率。路径费用值越低，路径就越好。表 1-8 列出了 STP 对于局域网链路的费用值。

表 1-8 STP 对于局域网的费用值

带宽	*修订的 STP 费用	带宽	*修订的 STP 费用
4 Mbit/s	250	155 Mbit/s	14
10 Mbit/s	100	622 Mbit/s	6
16 Mbit/s	62	1 Gbit/s*	4
45 Mbit/s	39	10 Gbit/s	2
100 Mbit/s	19		

* 在 IEEE 的标准修改之前，STP 的最低费用值维持在 1。STP 费用值为 1 是用于所有大于或等于 1Gbit/s 的链路，费用值为 10 主要是用于 100 Mbit/s 的链路，而费用值为 100 主要是用于 10 Mbit/s 的链路。

STP 有 6 个主要的状态，有 4 个状态是它操作过程中的过渡状态，思科的交换机有两个额外的专有状态，可在操作的过程中分配。当 STP 收敛完成后，它会处于下面的两种状态之一：转发或者阻塞。表 1-9 列出了 STP 的状态。

STP 也给参与生成树协议的每一个端口分配了一个端口状态。STP 的端口状态如下所述：

表 1-9

STP 的不同状态

STP 状态	STP 激活	用户数据传送
Disabled (关闭)	端口没有激活，它不会参与任何 STP 的活动	不传送
Broken (断离)	802.1Q 骨干在一端错误配置或者默认的本征 VLAN 在两端不匹配，STP 根防护生效	不传送
Listening (侦听)	端口正在发送和接收 BPDU	不传送
Learning (学习)	构造无环的桥接表	不传送
Forwarding (转发)	发送和接收用户数据	传送
Blocking (阻塞)	不允许用户数据从端口发送	不传送
端口加速*		传送
上行链路加速*		传送

* 端口加速和上行链路加速是思科专有的状态，它允许用户的数据在 STP 的收敛过程中被转发。

- **Designated port (指定端口)**——指定端口是背向根桥的端口。在根桥上，所有的端口都是指定端口。在每一个网段上只会选举出一个指定端口。指定端口最终会置为转发状态。
- **Root port (根端口)**——根端口是面向根桥的端口。根端口是非根桥到达根桥的端口中费用最低的那个端口。在每一个非根桥上只能选举出一个根端口。根端口被置为转发状态。
- **Nondesignated port (非指定端口)**——任何选举结果中既不是根端口又不是指定端口的端口最终选举为非指定端口。非指定端口最终被置为阻塞状态。

注意：在某些交换机的文档中，你可能注意到 STP 桥是用传统的桥的图标来表示的。实际上，现实中确实没有物理的桥。桥的图标和交换机的图标是相同的。本文是用交换机的图标来表示交换机和其上的 STP 桥。

STP 端口和角色的关系在图 1-13 中表示出来。

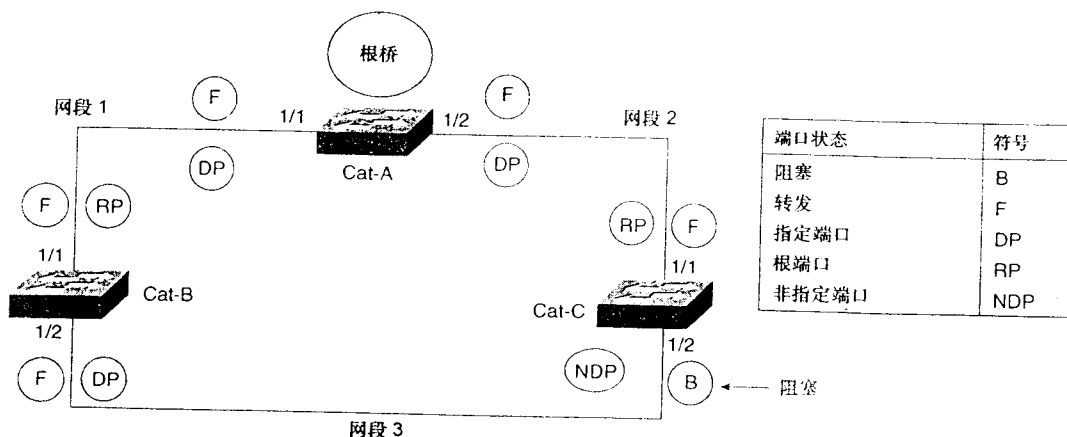


图 1-13 STP 端口和角色

在图 1-14 中，描述了端口从 STP 的一个状态切换到另外一个状态的过程。下面的内容将会更加详细地研究每一个状态。

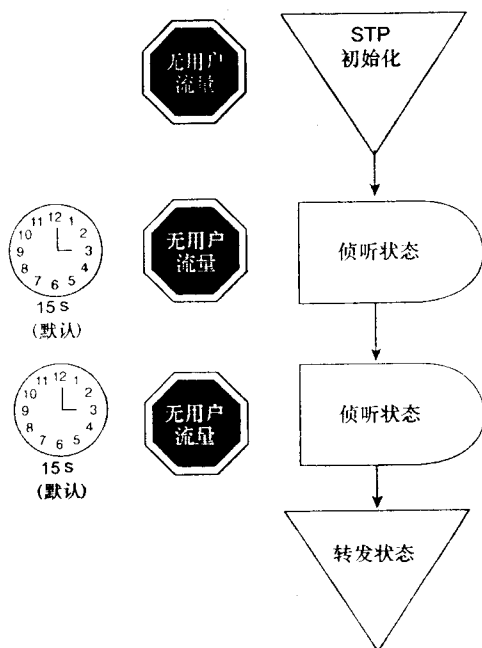


图 1-14 STP 的转换

四、关闭或者断离

断离状态发生在一个桥不能正确处理 BPDU 的数据包或者一个骨干链路没有被正确配置的情况下。断离状态发生在一个 802.1Q 的骨干链路在一端错误地配置，或者本征 VLAN 在骨干链路的两端不匹配。断离状态也可发生在骨干链路上，当 STP 的根防护生效时。从 STP 的角度来看，关闭状态发生在一个端口被管理性地关闭的情况下。

五、侦听

当一个交换机的端口被初始化或者当端口在超过最大时间计数器的时间范围（通常是 20s）接收不到 BPDU 数据包的情况下，STP 将端口过渡到侦听状态。当 STP 处于这个状态时，这个端口实际上是阻塞的，并且没有用户数据在这条链路上发送。端口在这个状态上可以保持 15s，被称为转发延迟计时器。

STP 的收敛遵循三步过程：

1. 选举出一个根桥。一旦初始化，桥开始在所有接口上发送 BPDU。根桥基于具有最低 BID 的桥被选举出来。回忆一下，BID 是优先级和 MAC 地址的组合。因为优先级在 BID 中是优先考虑的，所以具有最低优先级的桥被选举为根桥。在优先级一致的情况下，就把具有最低 MAC 地址的桥选举为根桥。根桥上的所有端口都被选举为指定端口并且置为转发状态。

2. 对每一个非根桥选举出一个根端口。当惟一的根桥被选举出来以后，STP 在每一个非根桥的交换机上选举出一个根端口。根端口是这个桥到达根桥的最佳路径。当根端口选举出来以后，它被置为转发状态。为了决定哪个端口是根端口，STP 遵循下面的决策过程：

- a. 最低的根 BID；根桥的 BID
- b. 到达根桥的最低路径值；到达根桥的所有路径的累积费用值
- c. 最低的发送者 BID；最低的端口 ID

影响根端口选举的主要可变因素就是到达根桥的费用值，这是因为许多桥和根桥没有直接相连。

当桥收到 BPDU 数据包后，它会将信息保存在那个端口的桥表里。当新的 BPDU 在那个端口收到以后，它们会和现有的 BPDU 数据包进行比较，更具有吸引力的 BPDU 或者具有更低费用值的 BPDU 会保留下来，而其他的 BPDU 数据包会被丢弃，这可能会使得交换机/桥将端口的状态转换成转发或者阻塞状态。

3. 在每一个网段上选举一个指定端口。对于每一个网段，STP 会选举出一个背离根桥的端口，称为指定端口。指定端口被置为 STP 的转发状态。

所有剩余的端口成为非指定端口，并且被置为阻塞状态。

六、学习

保持在指定或者根端口状态的端口会等待 15s 的周期，也就是默认的转发延迟时间，进入学习状态。学习状态是桥要等待的另外 15s 时间，在这个时间内它要构造桥表，这样做的目的就是确保桥的拓扑稳定。

七、转发和阻塞

当桥到达这个阶段，没有充当特定角色的端口（例如既不是根端口，也不是指定端口）被称为非指定端口。所有的非指定端口被置为 STP 的阻塞状态。在阻塞状态下，桥不发送配置的 BPDU，但是会侦听 BPDU。一个阻塞的端口不会转发用户数据。

1.3.2 STP 计时器

STP 有三种基本的计时器可以规范和老化 BPDU：hello、forward delay 和 max age。这些计时器为 STP 完成下面的任务：

- **Hello timer**——默认的 hello 计时器是 2s，这是根桥发送的配置 BPDU 的间隔时间。
- **Forward delay timer**——这个默认的 15s 是交换机用来等待并且构造桥表的计时器时间。侦听和学习阶段都使用这个 15s 的计时器。
- **Max age timer**——默认的最大老化时间是 20s。最大老化时间代表一个 BPDU 最多能够保存多长时间最终会被丢掉。如果在接口收到一个新的 BPDU 之前，这个计时器已经过期了，那么这个接口会过渡到侦听状态。一个过期的最大老化时间参数通常是由链路故障引起的。

STP 使用 hello 计时器来间隔 BPDU 的发送，有一种 keepalive 机制。hello 计时器应当防止最大老化计时器到期。如果最大老化计时器过期，通常表示链路故障。当这种情况发生时，桥会重新进入侦听状态。为了使 STP 从链路故障中恢复过来，大约需要 50s 的时间。20s 的时间用于 STP 老化，15s 用于侦听状态，15s 用于学习状态。

种形式的生成树协议。STP 所有类型的操作都是非常类似的。思科路由器支持所有类型，而思科的以太网交换机当前只支持 IEEE STP，令牌环交换机支持 IBM STP。

1.4 Catalyst 3550 的配置模式和术语

配置一个 Catalyst 3550 交换机非常类似于在先前的交换机中配置思科 IOS 软件，例如思科 Catalyst 3500XL 系列的交换机，或者类似于在传统的思科 IOS 的路由器平台上配置具有路由和服务质量特性的过程。下面的内容集中讨论安装了 EMI 软件的 Catalyst 3550 交换机如何配置。

Catalyst 3550 CLI 有不同的配置模式和不同的接口类型。例如，路由端口和交换虚拟接口 (*switched virtual interface*) 的配置不同，而交换虚拟接口的配置又和接入端口的配置不同。每种接口类型都是在不同的配置模式下配置的。因此，当讨论 Catalyst 3550 配置时，有一种通用的术语非常重要。

这些配置模式的一种类型或另外一种类型对你来说可能都是非常常见的。然而，Catalyst 3550 可能是你见过的集合了所有配置模式的第一种平台。表 1-10 列出了所有可用的配置模式和关于它们的简短描述。

表 1-10 在 Catalyst 3550 上的配置命令模式

模式的名字	提示符	起始提示符*	描述
用户模式	Switch>	Switch>	默认模式，主要用于基本的 show 命令
特权模式	Switch#	Switch>	对于 VLAN 配置模式和全局配置模式需要这个模式
全局配置模式	Switch (config) #	Switch#	用来配置适用于整个交换机的配置参数，路由选择协议也是在这儿配置的
VLAN 接口配置模式	Switch (config-vlan) #	Switch (config) #	用于在管理 VLAN 建立交换虚拟接口 (SVI) **，扩展的 VLAN 也是在这个模式下建立
VLAN 配置模式	Switch (vlan) #	Switch#vlan database	用于对 VLAN 1 到 VLAN 1005 配置 VLAN 和 VTP 参数，例如 VTP 和 VLAN 的名字，范围在 1~1001
多生成树配置模式	Switch (config-mst)	Switch (config) #	用于配置 MST 的特性，例如名字、修订号和实例
接口配置模式	Switch (config-if) #	Switch (config) #	用于配置以太接口的参数，例如 VLAN 成员的所属和双工模式
线路配置模式	Switch (config-line) #	Switch (config) #	用于配置控制台和 vty 的参数和访问权限

* 起始提示符指的是必须使用这种配置模式或必须利用它进入另外一种配置模式。

** SVI=交换虚拟接口。

Catalyst 3550 交换机也支持一系列的接口类型。每一种接口类型都配置用来支持交换机上的一个特性。下面的小节列出并简短描述了在 Catalyst 3550 交换机上支持的不同端口和接口类型。在以后的章节中我们会学习到更多配置这些接口类型方面的知识。

1.4.1 交换端口

交换端口是和物理端口关联的二层接口。Catalyst 3550 交换机有三种主要类型的交换端

口：*access port*（接入端口）、*trunk port*（骨干端口）和 *tunnel port*（隧道端口）。3550 交换机上的端口的默认模式是 *switchport*（交换端口），这一点和其他交换机例如 Catalyst 3548XL 略有不同，它们的默认模式是 *switchport access*。**switchport** 命令可以将这个端口置为路由模式或者交换模式。当一个端口置为交换模式时，它可以被配置为接入端口、骨干端口或者是隧道接口。

- **Access port（接入端口）**——接入端口是只属于一个 VLAN 并且静态分配到那个 VLAN 的端口。它们携带的流量是不带标签的，来自这个端口的流量只属于这个端口所属的 VLAN。如果接入端口收到了打标签的流量（ISL 或者 802.1Q），那么这类流量会被丢掉。
- **Trunk port（骨干端口）**——骨干端口可以配置为 802.1Q 或者 ISL 骨干。ISL 骨干端口期望在这个端口上只接收打了 ISL 标签的数据帧。802.1Q 的骨干有一个本征 VLAN，所有没有打标签的帧用的是本征 VLAN，默认的值是 1。所有打标签和没有打标签的流量如果有一个空的 VLAN ID，就认为属于本征 VLAN。一个数据帧的 VLAN ID 如果等于本征 VLAN，就不带标签发送，其他的帧带有 VLAN 标签发送。
- **802.1Q tunnel port（802.1Q 隧道端口）**——802.1Q 隧道端口将一个 VLAN 的信息和数据跨过局域网边界在另外一个 VLAN 中进行传输。边界交换机能够给数据帧加上适当的 VLAN 标签，接着通过 802.1Q 隧道将这个打了标签的帧传递给核心/分发层交换机，核心/分发层交换机给这个数据帧添加另外一层标签，然后通过局域网进行传输。交换机上配置为隧道端口的端口可以识别这些帧并且正确地处理它们。802.1Q 隧道用于非常大的企业级网络中，VLAN 的范围已经超过了 4096 这个限制。因为包含在 802.1Q 隧道中的交换机的数量和这种应用主要致力于大型企业的用户，802.1Q 隧道方面的内容超出了本章的讨论范围。

1.4.2 以太通道端口组

以太通道端口组将多个物理上的交换端口合并成一个逻辑端口。以太通道端口组将物理端口的特性绑定在新的逻辑端口上。如果这个组中的端口被配置为 802.1Q 骨干，那么这个逻辑以太通道端口就是一个 802.1Q 骨干。交换机在这个通道组中的所有物理端口上实现流量负载分担。有非常明确的规则规定什么样的交换端口和多少个交换端口可以放入以太通道组，这取决于交换机的特定体系结构。

1.4.3 交换虚拟接口（SVI）

交换虚拟接口（SVI）是逻辑接口，它含有三层的功能，例如将一个 IP 信息和 VLAN 关联在一起。相反地，SVI 也可以用来实现 VLAN 之间的路由，还可以回退到桥接非路由状态实现 VLAN 之间的桥接。对 VLAN 来说它意味着一个路由域。默认情况下，SVI 主要是出于管理 VLAN 1 的建立。如果你通过《CCIE 实验指南（第 1 卷）》非常熟悉思科 2900XL/3500XL 系列交换机的话，SVI 非常类似于用于管理的“接口 VLAN 1”。但是不像先前的交换机，你可以配置多个 SVI 和路由选择协议实现 VLAN 之间的路由。为了配置 SVI，除了默认的配置，

必须在交换机上安装 EMI 软件。

1.4.4 路由端口

路由端口的表现行为和它名字的含义非常相像。它是交换机上的物理端口，但是不属于任何 VLAN。它有三层的信息，例如 IP 地址，替代了 VLAN 信息。路由端口的功能和路由器上接口的功能是类似的。一个路由端口不能含有 VLAN 的子接口，需要在交换机上安装 EMI 软件。要成为路由端口，这个端口的交换功能必须被关闭掉（可以使用 **no switchport** 命令完成此功能）。路由端口使用一个内部的 VLAN ID。

不同的端口和接口可以通过不同的方式使用，图 1-15 演示了它们如何应用在一个通常的网络上。

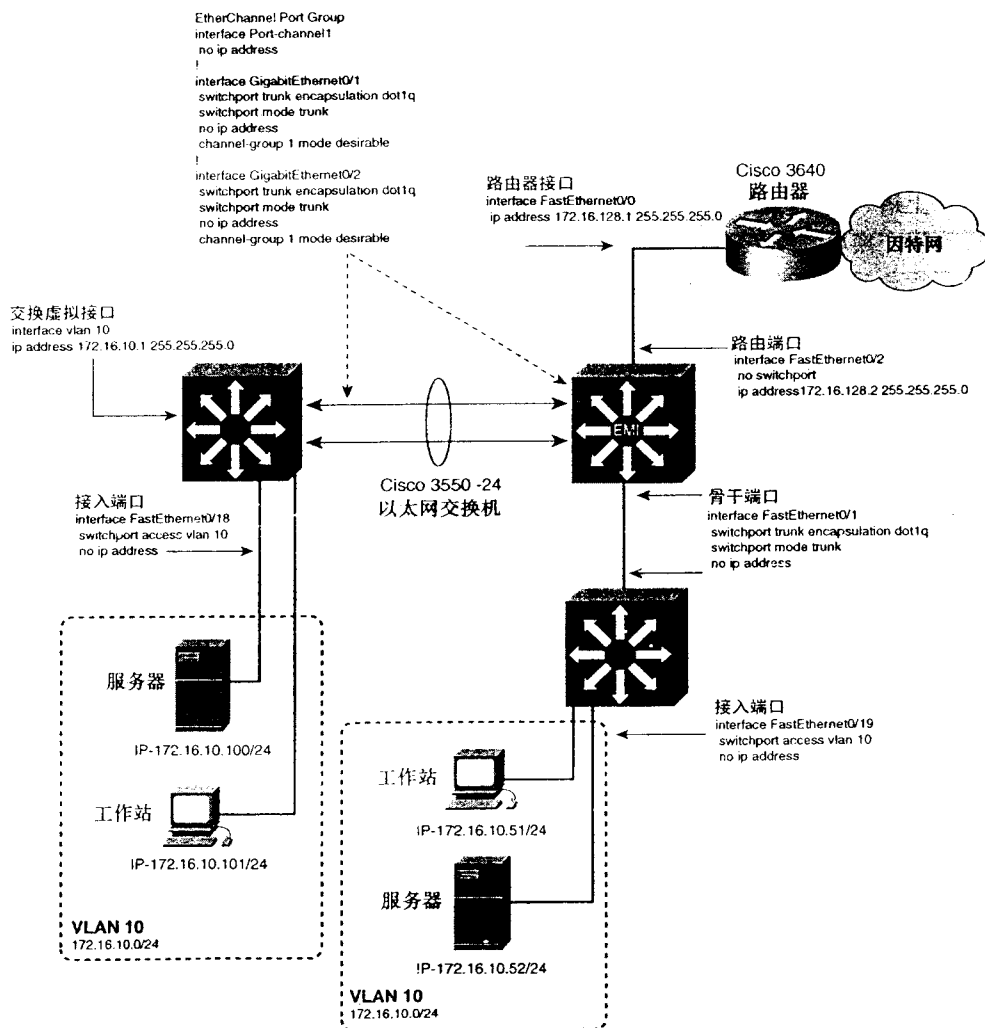


图 1-15 在 Catalyst 3550 交换机上的不同端口和接口

1.4.5 配置 Catalyst 3550 以太网交换机

Catalyst 3550 是一个非常灵活的交换机。随着 EMI 软件的安装，交换机本质上具有许多和路由器上存在的思科 IOS 软件相同的配置选项。通常的管理和安全功能就像它们在路由器上一样配置，例如，主机名、启用口令、路由选择协议和 IP 地址的配置与路由器上的配置是一样的。如果你曾经配置过 Catalyst 2900XL/35xx 系列交换机和思科的路由器，就会发现配置 Catalyst 3550 是一件多么相像的事情。本章的剩余部分将会集中讨论如何配置 Catalyst 3550 的交换特性。

局域网交换机的设计主要是易于安装和配置。在小型网络中，只需要做很少的配置或者根本不用做任何配置。在大型的冗余网络中，具有多个 VLAN 和骨干链路，交换成为一个很重的任务。在 Catalyst 3550 交换机上配置以太交换需要 7 个步骤，如下所示：

第 1 步 配置交换机的管理。

第 2 步 配置 VTP 和 VLAN，并把端口/接口分配到 VLAN 中。

第 3 步 在交换机之间使用以太通道、802.1Q 或者 ISL 封装配置连接。

第 4 步 可选：控制 STP 和 VLAN 信息的传播。

第 5 步 可选：配置 SVI。

第 6 步 可选：配置路由端口。

第 7 步 可选：配置三层交换。

第 1 步包括在交换机上配置管理 VLAN、IP 地址和默认网关，使得交换机可以通过互连网络进行访问。

第 2 步可以在 VTP 服务器或者透明模式的交换机上定义 VTP 域和 VLAN。在此步中，可以将端口分配到 VLAN 中。

第 3 步包括配置 VLAN 的骨干链路，如果网络上有的话。

第 4 步是可选的，但是对大型网络是非常重要的，它通过设置根桥控制 STP，从骨干链路上清除 VLAN，并且合理地使用 VLAN 剪枝。

第 1~4 步可以在许多 Catalyst 系列交换机上执行，而第 5~6 步只适用于 Catalyst 3550 系列交换机。

第 5 步包括配置 SVI，它主要用于 VLAN 之间的路由和连接。

第 6 步要求配置一个路由端口。路由端口用于下面这种情况：你想在一个接口上放置一个静态的三层地址，并且想让这个接口和正常的路由器接口的表现行为是一样的。也就是说，从这个接口上通过不打帧的标签，并且不发送 VLAN 的信息。路由端口当然是可以路由的。为了使用路由端口，必须安装 EMI 软件。

第 7 步也只适用于安装了 EMI 软件的交换机。不管出于什么目的，三层交换意味着在交换机上启用路由选择协议。

一、第 1 步：配置交换机的管理

所有的 Catalyst 交换机都具有通过一个 IP 地址被远程管理的功能。默认情况下，Catalyst 3550 使用动态主机配置协议（DHCP）来解析 1 号虚拟接口（SVI 1）上的默认网关。如果 DHCP 服务器不可用，IP 地址和默认网关可以手动分配。为了完成本任务，必须给交换机分

配 IP 地址、默认网关或默认路由，使得 IP 流量可以通过它转发。默认的管理 VLAN 是 VLAN 1，也可以指定另外一个 VLAN。

Catalyst 3550 交换机上的思科 IOS 软件类似于路由器，有一个特殊的 VLAN 数据库添加进去并且具有建立多个 VLAN 接口（SVI）的功能。分配端口、骨干和管理的这些命令都是从交换机的全局配置模式下执行的，从 VLAN 1 到 1001 的 VLAN 信息和 VTP 信息从全局配置模式或者是从 *VLAN 配置模式* 进行配置，有时候也称为 VLAN 数据库。从启用/特权模式下使用 **vlan database** 可以访问 VLAN 配置模式或者是 VLAN 数据库。

3550 交换机有一个默认的虚拟接口称为接口 VLAN 1。这是交换机的默认 VLAN，处于管理性关闭（Administratively Down）状态。为了分配一个管理 IP 地址，从 VLAN 接口配置模式下输入一个 IP 地址，并且使用 **no shutdown** 命令激活这个虚拟接口。如果是 VLAN 1 用于管理，接口已经激活，不需要更进一步的配置。范例 1-1 演示了如何在 VLAN 1 上配置管理接口。

范例 1-1 在 Catalyst 3550 交换机上配置管理接口

```
3550_switch(config)#interface vlan 1
3550_switch(config-if)#ip address 172.16.100.10 255.255.255.0
3550_switch(config-if)#no shut
3550_switch(config-if)#
00:07:25: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:07:26: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state up
```

如果管理地址配置在非 VLAN 1 的 VLAN 上，为了使得接口 up 和激活，必须确保满足下列条件：

1. 匹配这个接口的 VLAN 必须在 VLAN 数据库中。
2. 和那个 VLAN 相关的接口必须是 up 的，或者骨干链路必须是 up 的。

在范例 1-2 中，管理接口是 VLAN 128。为了激活这个接口，必须在交换机上建立 VLAN 128，建立的虚拟接口称为接口 VLAN 128，并且在 VLAN 128 中有一个活动的接口。如果在交换机上配置了骨干链路，这个虚拟接口也会激活。范例 1-2 演示了在 VLAN 128 上配置管理接口。注意只有在物理接口 FAST 0/10 激活后，VLAN128 才会激活。

范例 1-2 在 VLAN 128 上配置管理接口

```
3550_switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
3550_switch(config)#vlan 128
3550_switch(config-vlan)#exit
3550_switch(config)#interface vlan 128
3550_switch(config-if)#ip address 172.16.128.16 255.255.255.0
3550_switch(config-if)#exit
3550_switch(config)#interface fast 0/10
3550_switch(config-if)#switchport access vlan 128
3550_switch(config-if)#no shut
3550_switch(config-if)#
00:52:36: %LINK-3-UPDOWN: Interface FastEthernet0/10, changed state to down
00:52:37: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/10, changed state to down
00:52:40: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/10, changed state to up
00:53:10: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan128, changed state to up
```

查看管理接口可以像查看物理接口一样，使用 **show interface vlan x** 命令。

为了配置默认网关，使用 **ip default-gateway ip_address** 命令，和在路由器上一样。范例 1-3 显示了如何配置默认网关，紧随着使用 **show ip route** 命令来验证新的默认网关/路由。在这里，默认网关指向路由器 172.16.128.5。

范例 1-3 在 Catalyst 3550 上配置默认路由

```
3550_switch(config)#ip default-gateway 172.16.128.5
3550_switch(config)#exit
3550_switch#
3550_switch#show ip route
Default gateway is 172.16.128.5
Host          Gateway          Last Use      Total Uses    Interface
ICMP redirect cache is empty
3550_switch#
```

注意：VLAN 1——“就说不”

CCIE PSV 1 强调避免使用 VLAN 1 发送用户流量。我在这方面的设计规则就是尽可能避免使用 VLAN 1，有若干个理由需要这样做。VLAN 1 是所有 Catalyst 交换机的默认 VLAN 和本征 VLAN。任何添加到网络中的交换机默认情况下就是进入 VLAN 1，这使得网络易于受到潜在的 VTP、VLAN 和数据崩溃这类事件的影响。在 802.1Q 上的单生成树协议使用 VLAN 1 作为它的生成树域。根据所用的封装，交换机对 VLAN 1 数据帧的打标签方式也和其他 VLAN 不同。某些 Catalyst 交换机允许从骨干上清除 VLAN 1，而有些不允许这样做，这会导致 VLAN 1 跨越整个交换网络。出于这些原因和没有列出来的理由，我个人不在 VLAN 1 上运行生产性的流量或者管理流量。当设计局域网而 VLAN 1 也可用时，坚持说不行！

在 Catalyst 3550 上控制 IP 和 Console 的访问

在 Catalyst 3550 交换机上控制访问和在路由器上控制访问是一样的。可以设置一个启用口令，也可以设置一个 **enable secret** 口令。所有适用于路由器的启用口令和 **enable secret** 口令也适用于交换机。完成这个任务的语法如下：

```
3550_switch(config)#enable password cisco
```

启用口令没有加密并且可以在配置中看到。启用口令可以用全局命令加密：

```
3550_switch(config)#service password-encryption
```

service password-encryption 命令使用思科专有的加密方法，简称类型 5，加密交换机上的所有口令。

```
3550_switch(config)#enable secret ccie
```

enable secret 口令总是加密的，它使用思科专有的加密方法（称为类型 7）来加密数据。如果启用口令和 **enable secret** 口令同时在交换机上配置，**enable secret** 口令优先使用，此口令在配置文件中不可读。**enable secret** 口令的完整语法如下：


```
3550_switch(config)#enable secret [level level] {password | [ encryption-type] encrypted-  
password}
```

可以使用完整的语法从一个地点到另外一个地点复制和粘贴加密的口令。当使用这个命令设置级别或者加密类型时，一定要特别注意，因为很容易把口令输入错误。特别推荐的经验就是在输入完所有的口令后，使用 **service password-encryption** 命令来加密所有的口令。这会避免其他潜在的安全和窃取口令的问题。

对 3550 交换机的访问可以通过在 console (cty) 和虚拟终端 (vty) 线路上配置口令和访问控制列表来实现。回忆一下《CCIE 实验指南 (第 1 卷)》，cty 是交换机/路由器上的控制端口，而 vty 线路是虚拟的远程登录会话。可以在交换机上使用 **show line** 命令查看绝对线路值，如范例 1-4 所示。线路 0 是 vty 或者控制端口，而线路 1~16 是 vty 或者虚拟的远程登录会话。

远程登录访问可通过在交换机上建立访问控制列表，并将它们通过使用 **access-class** 线路配置命令绑定在 vty 线路上来控制。也可在 SNMP 团体字符串中调用访问控制列表来控制对 SNMP 的访问。

范例 1-4 在 Catalyst 3550 交换机上的绝对线路值

3550_switch#show line											
Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
*	0	CTY	-	-	-	-	-	0	0	0/0	-
	1	vty	-	-	-	-	-	0	0	0/0	-
	2	vty	-	-	-	-	-	0	0	0/0	-
...text omitted											
	15	vty	-	-	-	-	-	0	0	0/0	-
	16	vty	-	-	-	-	-	0	0	0/0	-

范例 1-5 演示了配置用户名和口令来控制 console 访问和远程登录访问。此范例显示了在控制端口和 16 个 vty 端口上输入 **login local** 命令来控制访问权限。这会强迫交换机使用本地用户名和口令来进行验证。一个访问控制列表 (ACL 10) 绑定在 vty 的会话上。在这个范例中，这个访问控制列表只允许用户从 172.16.0.0 范围内的网络远程登录到交换机上。关于配置 CTY 和 vty 线路以及绝对线路值的详细信息，请参考《CCIE 实验指南 (第 1 卷)》第 1 章。

范例 1-5 在 Catalyst 3550 上配置默认路由

```
3550_switch(config)#username solie password cisco  
3550_switch(config)#line 0  
3550_switch(config-line)#login local  
3550_switch(config-line)#exit  
3550_switch(config)#  
3550_switch(config)#line 1 16  
3550_switch(config-line)#login local  
3550_switch(config-line)#access-class 10 in  
3550_switch(config-line)#exit  
3550_switch(config)#  
3550_switch(config)#username ksolie password cisco  
3550_switch(config)#access-list 10 permit 172.16.0.0 0.0.255.255
```

二、第2步：在 Catalyst 3550 交换机上配置 VTP 和 VLAN

在 3550 系列的交换机上配置 VTP 和 VLAN 需要如下三步过程：

第1步 配置 VTP 域和模式。

第2步 如果交换机在 VTP 服务器模式或者透明模式下操作，配置 VLAN。

第3步 配置物理端口的属性并将端口分配到 VLAN 中。

三、在 Catalyst 3550 交换机上配置 VTP 域和模式

可以在 Catalyst 3550 交换机上从 VLAN 数据库或 VLAN 配置模式或者从类似于路由器的全局配置模式下配置 VLAN，许多部分的语法是一样的。如果你有配置 Catalyst 2900XL/35xx 交换机的大量经验，VLAN 配置模式对你来说可能会更熟悉。通过特权模式命令 **vlan database** 进入这个模式。当处在 VLAN 数据库中时，任何 VLAN 的变化都会起作用。在对 VLAN 数据库做出了修改后，可以输入下面的命令：

- **abort**——退出 VLAN 数据库并且进入 VLAN 数据库以后所做的任何 VLAN 修改全部无效，但是对 VTP 的修改不丢弃。
- **exit**——退出 VLAN 数据库并且对 VLAN 所做的所有改动起作用，同时也增加 VTP 的修订号。
- **apply**——当前的 VLAN 改动起作用并且增加 VTP 的修订号，但是不退出 VLAN 数据库。
- **reset**——清除任何当前的 VLAN 修改，并且重新读 VLAN 数据库。

出于安全原因始终应当对 VTP 域进行配置，这可以防止一台新的交换机无意中毁坏网络。默认的 VTP 域是空 (Null)，而且模式是服务器模式。为了配置 VTP 域，在 VLAN 配置模式中使用下面的语法：

```
3550_switch#vlan database
3550_switch(vlan)#vtp domain domain_name [password]
```

如果在域名后添加了口令，那么 VTP 更新将会使用一种叫做信息摘要 5 的算法 (MD5) 来哈希加密口令。使用 VTP 口令是一种非常有效的给交换域增强安全性和稳定性的方法。在当前的思科 IOS 版本中，只能在 VLAN 配置模式下配置 VTP 口令。不能在全局配置模式下输入 VTP 口令。要修改 VTP 模式，在 VLAN 配置模式下使用下面的命令：

```
3550_switch(vlan)#vtp {server | client | transparent}
```

要从全局配置模式下配置 VTP 域和模式，需要使用下面的语法：

```
3550_switch(config)#vtp domain domain_name
3550_switch(config)#vtp {server | client | transparent}
```

可以通过使用 **show vtp status** 命令来查看 VTP 域的状态。这个命令会显示关于 VTP 域的信息，例如配置修订号、域名、操作模式等诸如此类的信息。注意在显示的尾端，Catalyst 3550 交换机上会有新的信息出现。它会显示一个 IP 地址，表明 VTP 正在用哪个特定的交换机同步 VTP 信息。如果没有配置骨干链路或者配置不正确，就会出现一个全 0 的地址。如果交换机是 VTP 服务器模式并且没有通过骨干接收到更新数据，那么就会显示它自己的地址。

范例 1-6 列出了 **show vtp status** 命令的输出。

范例 1-6 查看 VTP 域的信息

```
3550_switch# show vtp status
VTP Version                : 2
Configuration Revision      : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs    : 6
VTP Operating Mode          : Server
VTP Domain Name             : psv2
VTP Pruning Mode            : Disabled
VTP V2 Mode                 : Disabled
VTP Traps Generation        : Disabled
MD5 digest                  : 0x03 0xE2 0xB2 0x25 0x2B 0xF1 0xBE 0x19
Configuration last modified by 172.16.128.16 at 3-1-93 03:16:46
Local updater ID is 172.16.128.16 on interface Vl128 (lowest numbered VLAN interface found)
Preferred interface name is 3550
3550_switch#
```

可以用下面的全局配置命令来配置接口或者 IP 地址，使 VTP 用来声明交换机，从而和 VTP 域中的其他交换机区分开。

```
3550_switch(config)#vtp interface [ VTP_updater_name | ip_address ]
```

注意：只有在 VTP 服务器模式的交换机其 VTP 修订号码大于客户模式交换机的修订号码时，VLAN 信息才会传播。如果客户模式的交换机的修订号码等于或者大于服务器模式的交换机，那么它不会接收 VLAN 信息。如想查看当前 VTP 的修订号码，在 Catalyst 4000/5500/6500 系列交换机上使用 **show vtp domain** 命令，或者在 Catalyst 2900/3500 系列的交换机上使用 **show vtp status** 命令。

1. 在 Catalyst 3550 交换机上配置正常范围和扩展范围的 VLAN

如果 VTP 模式配置为服务器模式或者透明模式时，第 2 步包括对 VLAN 的配置。如果交换机被配置为 VTP 客户模式，那么当骨干链路形成并且 VLAN 数据库被同步时，就会在交换机上出现 VLAN。可在 VLAN 的数据库中配置 VLAN，通过键入 **vlan [1-1001] options** 来实现。就像以前提到的，VLAN 1002 到 1005 和 VLAN 1009 是默认和特殊的 VLAN，它们不应当用在以太交换中。VLAN 也可以在全局配置模式下使用语法 **vlan [1-4094]** 来配置。VLAN 1006 到 4094 是扩展范围的 VLAN，可以在全局配置模式下使用。交换机必须在 VTP 透明模式下配置扩展 VLAN。

配置正常范围的 VLAN 可以在全局配置模式下或者 VLAN 数据库中配置正常范围的 VLAN，VLAN 1 到 VLAN 1001。如果 VLAN 是从 VLAN 数据库配置的话，那么 VLAN 的变化必须使用 **apply** 命令进行确认，当从 VLAN 数据库退出后，所有 VLAN 的变化都会被确认。如果发生了错误，可以用先前提到的 **abort** 或 **reset** 命令来取消 VLAN 变化。VLAN 数据库保存在闪存的 VLAN.DAT 文件中。可以将 VLAN.DAT 文件复制到 TFTP 服务器上去，就像你复制任何文件实现备份一样。范例 1-7 演示了在 Catalyst 3550 交换机上配置 VLAN 的两种方法。第一种方法使用 VLAN 数据库，而第二种方法演示了如何使用全局配置命令。在范例中，建立了两个 VLAN，VLAN 128 的名字是 **psv2_vlan128**，而 VLAN 10 的名字是 **psv2_vlan10**。

范例 1-7 配置 VLAN 128 和 VLAN 10

```
3550_switch#vlan database
3550_switch(vlan)#vlan 128 name psv2_vlan128
VLAN 128 added:
    Name: psv2_vlan128
3550_switch(vlan)#apply
APPLY completed.
3550_switch(vlan)#exit
! The preceding command automatically applies updates
APPLY completed.
Exiting....
Global Configuration mode----->
3550_switch#conf t
3550_switch(config)#vlan 10
3550_switch(config-vlan)#name psv2_vlan10
```

在 VLAN 配置模式下，有一些常见的选项可以配置在 VLAN 上，如下所示：

```
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**——允许用户给 VLAN 起一个 32 位字符的名字。
- **state**——允许用户将一个 VLAN 挂起，被挂起的 VLAN 可以通过 VTP 传播，但是用户流量不能在这个 VLAN 中传输。
- **said**——允许用户修改 VLAN 的 SAID 值，这个 SAID 值主要用于 802.10。
- **mtu, bridge 和 stp**——允许用户修改默认的 MTU 值、桥的号码和 STP 的类型。
- **No vlan [vlan_num]**——从 VLAN 数据库中删除一个 VLAN。当删除一个 VLAN 后，任何属于那个 VLAN 的端口都会变成非激活状态，包括管理接口。

如果你正在从全局配置模式中配置 VLAN 的选项，那么 VLAN 的选项是从 VLAN 接口配置模式下配置的。

对于默认的 VLAN 值，请查看本章前面的表 1-2。

为了查看 VLAN 的状态，使用 **show vlan** 命令，它可以显示交换机上的所有 VLAN，包括状态以及端口属于哪个 VLAN 等信息。为了显示关于某个 VLAN 的特定物理和逻辑信息，使用 **show vlan id [vlan_number]** 命令。范例 1-8 列出了 **show vlan** 命令的输出，随后是这个命令的特定版本的输出。注意 VLAN 的逻辑名可以帮助快速识别端口的目的。

范例 1-8 show vlan 命令的输出

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/11, Fa0/12, Fa0/13 Fa0/14, Fa0/15, Fa0/16, Fa0/17 Fa0/18, Fa0/19, Fa0/20, Fa0/21 Fa0/22, Fa0/23, Fa0/24, Gi0/1 Gi0/2
10 psv2_vlan10	active	
128 psv2_vlan128	active	Fa0/10

(待续)

```

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
10     enet    100010         1500   -      -      -      -   -        0      0
128    enet    100128         1500   -      -      -      -   -        0      0
1002   fddi    101002         1500   -      -      -      -   -        0      0
1003   tr     101003         1500   -      -      -      -   -        0      0
1004   fdnet   101004         1500   -      -      -      ieee -        0      0
1005   trnet   101005         1500   -      -      -      ibm  -        0      0
3550_switch#

-----
3550_switch#show vlan id 128
VLAN Name                Status      Ports
-----
128   psv2_vlan128         active     Fa0/10
VLAN Type  SAID          MTU    Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
-----
128    enet    100128         1500   -      -      -      -   -        0      0
3550_switch#

```

注意：Catalyst 3550 交换机支持 128 个 STP 实例。每一个 VLAN 运行一个不同的 STP 实例。如果你已经用完一个交换机上所有 128 个 STP 实例的话，那么在 STP 域中添加任何一个 VLAN 都会导致在那台交换机上这个 VLAN 的 STP 不可用。如果在那台交换机的骨干端口上有“默认的允许列表”（允许所有的 VLAN），那么这个新的 VLAN 也会在骨干链路上传输。取决于网络的拓扑，这可能会产生一个环路。这是因为新的 VLAN 被分隔了，特别是几个邻接的交换机都有超过了 128 个 STP 实例的情况。可以通过在交换机的骨干端口设置所允许的 VLAN 列表来防止这一点，使得交换机不会对所有的 VLAN 传播 STP 信息。这一点同在 Catalyst 5500/6500 系列交换机的骨干上清除 VLAN 是一样的。

2. 配置扩展范围的 VLAN

Catalyst 3550 交换机允许用户配置扩展 VLAN。扩展 VLAN 的范围在 1006~4094 之间。然而，3550 交换机对每一个路由端口使用扩展 VLAN ID。因此，实际的范围，也即安全的范围，对于扩展 VLAN 大约是 1027~4094。当配置扩展 VLAN 时必须遵循某些规则，这些规则如下：

- 在任何扩展的 VLAN 配置之前，交换机必须处于透明模式。
- 路由端口使用的扩展 VLAN 是较低的范围 1006~1026。总是选择扩展 VLAN ID 从 4094 开始往后的范围。注意可以使用命令 **show vlan internal usage** 来验证什么样的内部 VLAN 正在使用，以及什么样的接口正在使用它们。范例 1-9 演示了在配置扩展 VLAN 之前有关这个命令的使用。
- 只能从全局配置模式下建立扩展 VLAN，不能从 VLAN 配置模式下建立。
- 扩展 VLAN 不会保存在 VLAN 的数据库里，并且不会通过 VTP 协议通告出去。
- 扩展 VLAN 不被 VLAN 查询协议（VQP）或者 VLAN 成员策略服务器（VMPS）支持。
- STP 对于扩展 VLAN 默认是启用的。

- 现在，不能给扩展 VLAN 命名，只能改变 MTU 的值。

配置扩展 VLAN 和配置通常范围的 VLAN 的过程是一样的，除了你必须遵循前面列出的指导原则。范例 1-9 演示了对扩展 VLAN 4094 的配置。在配置扩展 VLAN 之前，交换机被置为 VTP 的透明模式，并且执行 **show vlan internal usage** 命令来避免 VLAN 的冲突。

范例 1-9 建立扩展 VLAN

```
3550_switch#show vlan internal usage ←Verify internal VLANs
VLAN Usage
-----
1017 -
1025 FastEthernet0/11
! VLAN 1025 in use by INT FAST 0/11
1026 GigabitEthernet0/2
! VLAN 1026 in use by INT GIG 0/2
3550_switch#
3550_switch#conf t
3550_switch(config)#vtp mode transparent
! VTP transparent mode set
Setting device to VTP TRANSPARENT mode.
3550_switch(config)#vlan 4094
! VLAN 4094 created
```

可以使用 **show vlan** 命令来查看扩展 VLAN。范例 1-10 显示了 VLAN 4094 建立后使用 **show vlan** 命令的输出。

范例 1-10 查看扩展 VLAN

```
3550_switch#show vlan
VLAN Name                Status    Ports
-----
1    default                active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                           Fa0/9, Fa0/12, Fa0/13, Fa0/14
                                           Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                           Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                           Fa0/23, Fa0/24, Gi0/1

10   psv2_vlan10             active
128   psv2_vlan128          active    Fa0/10
1002 fddi-default         active
1003 token-ring-default   active
1004 fddinet-default      active
1005 trnet-default        active
4094 VLAN4094            active

VLAN Type  SAID      MTU   Parent RingNo BridgeNo Stp   BrdgMode Trans1 Trans2
-----
1    enet    100001    1500  -     -     -     -   -       0       0
10   enet    100010    1500  -     -     -     -   -       0       0
128   enet    100128    1500  -     -     -     -   -       0       0
1002 fddi    101002    1500  -     -     -     -   -       0       0
1003 tr     101003    1500  -     -     -     -   -       0       0
1004 fdnet  101004    1500  -     -     -     -   ieee    0       0
1005 trnet  101005    1500  -     -     -     -   ibm     0       0
4094 enet    104094    1500  -     -     -     -   -       0       0
3550_switch
```

提示: Catalyst 3550 允许用户同时配置一组端口，如果需要在交换机上配置具有相同特性的许多端口，这样就可以极大地节省时间。为了配置一组端口，使用下面的全局配置命令:

```
Switch(config)#interface range interface_type staring_int - ending interface
```

为了配置范围在 0/1 到 0/10 的接口，范例中使用了下面的命令:

```
3550_switch#(config)interface range fastethernet 0/1 - 10
```

3. 在 Catalyst 3550 交换机上配置物理端口的属性并且将端口分配到 VLAN

VTP 和 VLAN 配置的下一步就是配置物理端口的属性，以及将端口分配到一个 VLAN 中去。物理端口的属性可以在接口配置模式中进行修改。表 1-11 列出了 Catalyst 3550 交换机上二层接口的设置。

表 1-11 Catalyst 3550 上默认的二层以太设置

特性	默认设置
操作模式	二层交换 (switchport)
所允许的 VLAN 的范围	VLAN 1-4094
默认 VLAN	VLAN 1
本征 VLAN	VLAN 1
VLAN 骨干协议	DTP
所有的端口启用	
速率	自适应
双工模式	自适应
流控	对于 10/100/1000Mbit/s 速率来说，接收流量是 off，发送流量是 desired (对于 10/100Mbit/s 而言发送流量总是 off)
以太通道 (PAgP)	关闭
未知组播、单播流量及风暴控制导致的端口阻塞	关闭
被保护的端口	关闭
端口安全	关闭
端口加速	关闭

范例 1-11 演示了在 3550 系列交换机上给一个以太端口配置 100Mbit/s 半双工的设置。此范例还给这个接口分配了一个逻辑名字 management_vlan_128。

范例 1-11 配置物理属性

```
3550_switch(config)#interface fast 0/10
3550_switch(config-if)#speed 100
3550_switch(config-if)#duplex half
3550_switch(config-if)#description management_vlan_128
```

注意: 为了修改一个端口的双工设置，必须首先将速率从自适应改为 10 或者 100Mbit/s。交换机不允许在端口配置为自适应的模式下修改双工设置。

可以在接口配置模式下修改下列这些常见的以太物理属性:

- **duplex [full | half | auto]**——设置端口的双工模式。
- **speed [10 | 100 | auto]**——设置端口的速率。
- **mtu [1500bytes-2018bytes]**——配置接口的 MTU。如果修改 MTU 的值，确保物理接口的 MTU 一定要匹配那个 VLAN 的 MTU。
- **description interface_description**——允许用户设置对端口的描述。
- **shutdown | no shutdown**——关闭或者启用一个接口。

接口命令 **switchport** 可以不带任何选项，它代表将一个端口置为二层交换模式。这个端口可以是一个接入端口、骨干端口、802.1Q 隧道端口、语音端口或者是保护端口。下述命令是 **switchport** 命令的子命令：

- **access**——将端口分配到一个单独的 VLAN 中。
- **trunk**——用于将这个端口配置为 802.1Q 或者是 ISL 骨干端口。下一节将更详细地讨论这个选项。
- **802.1q tunnel ports**——802.1Q 隧道端口将一个 VLAN 的信息和数据通过局域网在另外一个 VLAN 里进行传输。
- **voice vlan**——这个端口可以使用 802.1Q 和 802.1p 实现服务质量。
- **protected ports**——保护端口阻止在同一个交换机上保护端口之间的单播、组播和广播流量。

后面的内容用详细的篇幅介绍了不同的模式，现在我们的重点是将端口分配到一个 VLAN 中。为了完成这个任务，我们首先需要将这个端口配置为接入模式，接着将这个端口分配到一个 VLAN 中。用于完成这个任务的语法如下：

```
(config-if)#switchport access vlan [1-4094 | dynamic]
```

dynamic 关键字主要用于 VLAN 成员策略服务器（VMPS）的配置中。VMPS 在本书中没有涉及。关于 VMPS 的更多信息，参考《Cisco 局域网交换技术（英文版）》（人民邮电出版社）。

范例 1-12 显示了对 VLAN 2 的 Fast Ethernet 0/5 接口的配置。

范例 1-12 将接口 fast 0/5 分配到 VLAN 2 中

```
Switch(config)#int fastEthernet 0/5
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

当 VTP 模式设置为透明模式时，VLAN 可以通过 **switchport access vlan** 命令自动建立，没有必要在 VLAN 的数据库中静态配置它们。如果 VTP 的模式是客户模式，就不能在交换机上配置 VLAN。VLAN 必须在服务器模式的交换机上配置并且通过骨干，由 VTP 协议将它传播到客户模式的交换机上去。

四、第 3 步：在交换机之间使用以太通道、802.1Q 和 ISL 封装配置骨干链路

第 3 步包括在以太网交换机之间配置骨干链路，一个骨干链路可以是采用 ISL 或者 802.1Q 封装的普通骨干，也可以是以太通道骨干，同样它也可以采用 802.1Q 或者 ISL 封装。

我们这次讨论主要集中于配置一个普通的骨干，接着配置一个以太通道骨干。

在 Catalyst 3550 上配置骨干链路是一个两步的过程。根据配置之前端口的状态，你可能想关闭自适应的模式。默认情况下，端口被设置为协商封装方式并且处于 **dynamic** 和 **desirable** 模式下。

第1步 配置骨干链路的封装形式为 ISL 或者 802.1Q。

第2步 将端口配置为普通的骨干链路端口或者是以太通道骨干链路。

这些步骤是从接口配置模式下通过下述命令来实现的：

```
Switch#(config-if)#switchport trunk encapsulation {isl | dot1q | negotiate }
Switch#(config-if)#switchport mode {trunk | dynamic {auto | desirable}}
```

不同的封装类型和子命令如下所示：

- **switchport trunk encapsulation isl**——指定在骨干链路上采用 ISL 封装。
- **switchport trunk encapsulation dot1q**——指定在骨干链路上采用 802.1Q 封装。
- **switchport trunk encapsulation negotiate**——指定接口和邻居接口协商成为一个 ISL（最优）或者 802.1Q 的骨干链路，这取决于邻居接口的配置和能力。这是默认的封装类型。

作为骨干端口，可以静态配置，也可以动态配置。不同的骨干配置模式如下：

- **dynamic auto**——如果邻居设备的接口设置为骨干或者 **desirable** 模式，那么这个接口就成为骨干链路。
- **dynamic desirable**——如果邻居设备的接口设置为骨干、**desirable** 模式或者 **auto** 模式，那么这个接口就成为骨干链路，这是默认的骨干链路的模式。
- **trunk**——将这个接口设置为永久的骨干模式并和邻居设备协商将链路转变成骨干链路，即使邻居接口不是骨干接口。

你可能发现配置自适应或者 DTP 比静态地定义骨干链路要困难很多。这主要是因为对于不同的 Catalysts 交换机，默认的骨干链路的封装类型不一样。许多 Catalyst 交换机的默认类型是 ISL，然而不具有三层模块或者不具有最新思科 IOS 软件版本的 Catalyst 4000 交换机不支持 ISL 封装。另外的原因是在 CAT OS 版本 4.2 上只支持 802.1Q 自适应。这些微小的事情导致了 DTP 协议在大型且复杂的网络中不太可靠。

注意：另一个自适应的问题在 VTP 和 DISL 协议中也有所体现。当使用 DISL 协商 ISL 骨干链路时，它会在消息报文中包含 VTP 的名字。如果 VTP 的域名在交换机上不匹配，骨干链路就不会激活。而且，为了防止这一点，最好静态地配置骨干链路和封装类型。为了保证 VTP 工作，仍需匹配 VTP 的名字。

范例 1-13 显示了在吉比特以太接口 0/1 上配置 802.1Q 骨干的过程。

范例 1-13 配置 ISL 骨干链路

```
3550_switch(config)#interface gigabitEthernet 0/1
3550_switch(config-if)#switchport trunk encapsulation dot1q
3550_switch(config-if)#switchport mode trunk
```

为了验证骨干链路是否正常工作，确保链路两端的状况。**show interface interface_name**

switchport 命令的输出和 **show interface interface_name trunk** 命令的输出代表一个骨干链路的通常状态。这里出现的信息和在 Catalyst 4000/5500/6500 系列交换机上使用 **show trunk** 命令的输出结果是非常相似的。这个命令可显示骨干链路的状态和封装类型。VLAN 的信息，例如默认 VLAN、链路上的活动 VLAN 以及具有剪枝资格的 VLAN，都会在这个命令里列出来。而且，保护 VLAN 和语音 VLAN 也会在这个命令里列出来。范例 1-14 列出了 **show interface interface_name switchport** 命令的输出结果。如果骨干没有显示出来，那么就要注意下面这些配置区域：

- 模式；
- 封装类型；
- 802.1Q 骨干链路上的本征 VLAN。

范例 1-14 骨干链路的状态

```
3550_switch#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: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled

Voice VLAN: none (Inactive)
Appliance trust: none
3550_switch#
```

设置骨干链路的状态为骨干，模式为 on，或者和前面列出的 DTP 协议的一个有效设置相匹配。在骨干链路两端的封装类型必须匹配。本征 VLAN ID 是 802.1Q 的 VLAN，用于生成树协议中的一个实例（MST）。这个 VLAN 必须在整个 VTP 域中是相同的。

在 802.1Q 网络中，确保本征 VLAN 在整个 VTP 域中是相同的非常关键，这是因为 802.1Q 使用单生成树协议。单生成树协议确保整个 VTP 域对所有的第三方交换机看起来就像一个桥接域。思科通过实施 PVST+ 和 MST 确保与 MST 域的兼容性。这是基于每一个 VLAN 的生成树协议加（PVST+）的扩展版本，它可以提供对 802.1Q 网络的无缝、透明的集成。单生成树协议运行在本征 VLAN 上。出于这个原因，使得本征 VLAN 在整个互连网络中都是相同的非常重要。默认 VLAN 是 1，它同时也是默认的本征 VLAN。为了修改本征 VLAN，可在骨干链路的接口配置模式下使用下面的命令：

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

带有 **trunk** 关键字的 **show interface** 命令也会列出可剪枝的 VLAN。不要将可剪枝的 VLAN 和 VLAN 的传播混淆了。可剪枝意味着不必要的广播、组播和未知的单播流量不会随着骨干链路转发到没有任何一个活动端口在那个特定 VLAN 的交换机上。默认情况下，所有

的 VLAN 信息和每一个 VLAN 的生成树的帧都会随着骨干接口通告出去。在思科 Catalyst 5500/6500 系列的交换机上通过使用 **clear trunk** 命令将 VLAN 和 STP 从骨干链路上清除，或者是在思科 3550 系列的交换机上通过在骨干上修改可允许的 VLAN 来实现。随后我们会学习到关于这些功能的更详细的内容。

范例 1-15 列出了 **show trunk** 命令的输出。**trunk** 关键字和 **switchport** 关键字显示的信息是类似的，不过更集中于骨干链路上的 VLAN 信息。

范例 1-15 使用 trunk 关键字的骨干链路的状态

```
3550_switch#show interface gigabitEthernet 0/1 trunk
Port      Mode           Encapsulation  Status        Native vlan
Gi0/1     on              802.1q         trunking      1
Port      Vlans allowed on trunk
Gi0/1     1-4094
Port      Vlans allowed and active in management domain
Gi0/1     1,10,20,128
Port      Vlans in spanning tree forwarding state and not pruned
Gi0/1     1,10,20,128
3550_switch#
```

现在，确定一条骨干链路是否在正常工作可能会很困难。骨干可能会报告状态是骨干的，但实际上并没有真正地去交换 VTP 更新数据。应当在链路的两端都查看骨干的状态来确保它是在正常工作。

随着 VTP 在整个域内进行服务器对服务器、服务器对客户模式交换机 VLAN 数据库之间的同步，所有的交换机在它们的 VLAN 数据库中都具有相同的 VLAN 列表。只有处于 VTP 透明模式的交换机或者是从骨干上清除了某些 VLAN 的交换机具有不同的 VLAN 数据库。对通过骨干链路相连的两台交换机的 VLAN 数据库进行比较是另外一种验证骨干链路是否工作的方法。

当骨干链路激活后，就可以发送和接收 VTP 通告。下面三种类型的 VTP 通告发生在骨干链路上：

- **Subset advertisements**——当建立、删除或者修改 VLAN 时，就会发出子集通告。
- **Request advertisements**——当 Catalyst 交换机被复位或者在本地 VTP 域里发生了变化时，就会发出请求通告，例如名字的变化，或者是当交换机收听到的 VTP 汇总通告的修订版本号比它自身还要大时，都会导致这个请求的发生。
- **Summary advertisements**——汇总通告每隔 5min 就由交换机发出。这个汇总通告的主要目的是使交换机能够验证 VTP 的修订版本号，从而确保 VLAN 数据库是最新的。如果它有一个较低的修订版本号，那么它就会发出一个请求来获取最新的 VLAN 信息。

可以使用 **show vtp status** 和 **show vtp counters** 命令观察 VTP 的统计数字。这些命令告诉你交换机发送和接收的通告数量。这也可以是验证骨干链路是否在正常工作的另外一种指示方法。在验证骨干链路启用以后，仍然需要验证 VTP 的更新是否在骨干链路上交换。记住骨干的目的是传递 VLAN 信息，而它需要 VTP 协议。在骨干链路上，应当使用 **show vtp counters** 命令来检查 VTP 域的计数器。范例 1-16 列出了 **show vtp counters** 命令的输出。

范例 1-16 通过查看 VTP 的计数器来了解骨干链路的状态

```

3550_switch#show vtp counters
VTP statistics:
Summary advertisements received      : 101
Subset advertisements received       : 4
Request advertisements received      : 1
Summary advertisements transmitted  : 116
Subset advertisements transmitted    : 3
Request advertisements transmitted   : 0
Number of config revision errors     : 0
Number of config digest errors       : 0
Number of V1 summary errors          : 0

VTP pruning statistics:
Trunk      Join Transmitted Join Received  Summary advts received from
-----
Gi0/1      0                0                0
non-pruning-capable device
3550_switch#

```

show vtp status 命令的输出列出了非常有用的 VTP 信息，VTP 协议的版本号、VTP 的修订号、操作模式和域名以及 VLAN 信息都会列出来。当 VLAN 数据库被同步后，每一个交换机都具有相同数量的 VLAN。

范例 1-17 显示了 **show vtp status** 命令的输出

范例 1-17 查看 VTP 状态来了解 VTP 配置

```

3550_switch#show vtp status
VTP Version                : 3
Configuration Revision      : 3
Maximum VLANs supported locally : 1005
Number of existing VLANs    : 12
VTP Operating Mode          : Server
VTP Domain Name             : psv2
VTP Pruning Mode            : Disabled
VTP V2 Mode                 : Disabled
VTP Traps Generation        : Disabled
MD5 digest                  : 0x40 0x2B 0xD9 0xD1 0x05 0xA4 0x98 0xF8
Configuration last modified by 206.191.241.43 at 3-1-93 18:06:59
Local updater ID is 172.16.128.16 on interface Vl128 (lowest numbered VLAN interface found)
Preferred interface name is 3550
3550_switch#

```

1. 配置二层和三层的以太网通道

以太网通道是可以配置的另外一种形式的骨干链路。通常的方法是在两台交换机之间配置一个二层的以太网通道。通常同时还会配置 ISL 或者 802.1Q 的封装。如果在 Catalyst 3550 交换机上安装了 EMI 软件，那么也可以配置三层的以太网通道。

配置以太网通道时你应当意识到它有一些限制。某些限制是和硬件相关的，因此，对于你所配置的平台，查找一下关于这个平台对以太网通道的限制不失为一个很好的主意。

下面的列表适用于对 Catalyst 3550 以太网交换机的配置指导：

- 每个以太网通道最多有 8 个可配置的快速以太网接口和 8 个吉比特以太网接口。

- 不要将 GigaStack GBIC 接口配置为以太通道的一部分。
- 将以太通道中的所有接口配置为相同的速率和双工模式。
- 在一个以太通道中可以启用所有接口。可以使用 **shutdown interface** 命令关掉以太通道中的一个接口，它会被看作是一个链路故障，因此，流量就会转移到以太通道中剩余的接口上。
- 当组刚刚建立时，所有的端口都遵循第一个被添加到组中的端口的参数设置。如果你修改了这些参数中的任何一个配置，那么必须使组中所有的端口都跟随这个变化。
- 以太通道组中的一个接口作为交换机端口分析器（SPAN）的目的端口时，以太通道组不会形成。可以将以太通道组作为 SPAN 的源来监控整个通道组的流量。
- 如果一个端口属于以太通道组，那么就不能配置为安全的端口。
- 将以太通道组中的所有接口配置为属于同一个 VLAN，或者将它们配置为骨干，具有不同本征 VLAN 的接口不能形成以太通道组。
- 如果你配置的以太通道组是由骨干接口组成的，那么验证骨干链路的模式（ISL 或者 802.1Q）在骨干链路的两端都是相同的。
- 在组成一个二层的以太通道组的所有骨干接口上要保证所允许的 VLAN 的范围是一样的。如果所允许的 VLAN 的范围不一样，那么当 PAgP 设置为 auto 或者 desirable 模式时，接口不会形成以太通道。
- 在一个端口启用 802.1X 验证之前，必须将它从以太通道中移除。如果在以太通道中一个还没有激活的端口上启用了 802.1X，那么这个端口不会加入以太通道。
- 具有不同 STP 路径费用值的接口可以形成一个以太通道组，只要它们具有兼容性配置即可。设置不同的 STP 路径费用值从它自身来说，不会使得接口在以太通道组中不兼容。
- 对于三层以太通道来说，给这个逻辑接口分配一个三层地址，而不是给通道中的物理接口分配三层地址。
- 默认情况下，PAgP 不会分配或定义二层或者三层的以太通道组。对于 PAgP 的以太通道组的配置模式是 auto 和 silent；接口会响应 PAgP 数据包，但是不会主动发起 PAgP 协商。PAgP 被配置为一个聚合端口学习者，所有端口的优先级都是 128。

在 Catalyst 3550 上配置二层以太通道的 ISL/802.1Q 骨干是一个三步的过程。根据配置之前的端口状态，你可能想关闭掉自适应的状态。默认情况下，一个端口可以被设置为 dynamic 和 desirable 模式来协商封装类型。

第1步 配置骨干的封装类型为 ISL 或者 802.1Q。

第2步 将端口配置为骨干端口。

第3步 配置以太通道端口组。

这些步骤可以用下面的接口配置命令来完成：

```
Switch#(config-if)#switchport trunk encapsulation [isl | dot1q / negotiate]
Switch#(config-if)#switchport mode [trunk | dynamic {auto | desirable}]
Switch#(config-if)#channel-group [1-64] mode {auto [non-silent] | desirable [non-silent]
| on}
```

头两个命令在配置通常的 ISL 或者 802.1Q 骨干链路时是相同的。**channel-group** 命令建立一个虚拟接口称为接口端口通道 x ， x 是通道组的号码。虚拟接口列出了所有通用的属

性，它们必须和加入到端口组中的任何链路相关。这个虚拟接口也是给三层通道组分配 IP 地址的地方。通道组的号码可以从 1~64。**mode** 关键字启用或者关闭 PAgP。PAgP 工作得相当稳定，但是要确保在通道组中的所有端口上运行 PAgP 的模式是相同的。**mode** 关键字有下列参数：

- **auto**——只有另外的 PAgP 设备被检测到，才启用 PAgP。它可以将接口置为被动的协商状态，这个接口会响应收到的 PAgP 帧，但是不会主动发起 PAgP 协商。
- **desirable**——无条件地启用 PAgP。这个关键字将接口置为活动的协商状态，这样接口可以主动地向对端的接口发起 PAgP 协商数据帧。
- **on**——强制接口在没有 PAgP 协议的情况下成为通道。在 on 模式下，只有一个接口组处于 on 模式，而对端的接口组也处于 on 模式的情况下，这个通道组才会形成。
- **active (LACP)**——将接口设置为活动的协商状态，这样接口主动发送 LACP 的数据包与对方的接口协商。
- **passive (LACP)**——将接口设置为被动的协商状态。在这个模式下，接口会响应收到的 LACP 数据包，但是不主动发起 LACP 数据包的协商。这个设置可以最小化 LACP 数据包的数量。

以太网通道组也可以分配到一个特定的 VLAN 中，虽然这一点不常见。为了完成这个任务，将所有的接口配置为静态接入的端口，分配到同一个 VLAN 中。

当配置以太网通道组时，你可能注意到链路掉了并且重新初始化了一会儿——一次是因为封装类型的变化，还有至少一次是因为端口加入了通道组。为了防止这种情况发生，在配置任何骨干链路或者以太网通道组的参数之前，将链路所属的端口关掉。为了将一个接口从以太网通道组中去除，使用 **no channel-group** 接口配置命令。

图 1-16 代表了一个常见的网络。在这个局域网中，核心交换机彼此互相连接并且和边界交换机连接。核心交换机准备使用吉比特以太网通道，将两台交换机之间的链路配置成骨干。802.1Q 将会是 VLAN 的骨干协议并且最终允许 VLAN 192 实现完全连接。

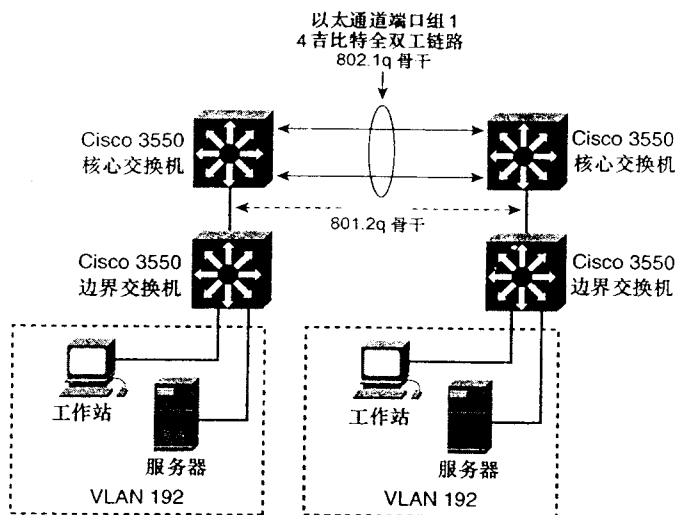


图 1-16 在 Catalyst 3550 交换机上的吉比特以太网通道

范例 1-18 显示了在图 1-16 中两台核心交换机之间配置以太通道。

范例 1-18 使用 802.1Q 封装配置吉比特以太通道

```
3550_switch(config)#interface gigabitEthernet 0/1
3550_switch(config-if)#switch trunk encapsulation dot1q
3550_switch(config-if)#switchport mode trunk
3550_switch(config-if)#channel-group 1 mode on
Creating a port-channel interface Port-channel1
3550_switch(config-if)#exit
00:23:18: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:23:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed state
to up
3550_switch(config)#interface gigabitEthernet 0/2
3550_switch(config-if)#switchport trunk encapsulation dot1q
3550_switch(config-if)#switchport mode trunk
3550_switch(config-if)#channel-group 1 mode on
00:24:29: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:24:31: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed
state to up
3550_switch(config-if)#exit
```

思科提供了一些有用的命令来验证以太通道的操作状态：

```
show etherchannel [ channel-group-number] {brief | detail | load-balance | port | portchannel
| summary}
show interface etherchannel
```

show etherchannel 命令显示了在以太通道端口组中的端口数量及其所处的模式和其他一些信息。你应当看到端口的状态是 up 的，并且属于这个端口组的所有端口都应当被列出来。这个命令也显示了负载均衡或者数据帧分发机制的一些信息、端口及端口组的状态。在组中的 L2 状态定义这个以太通道组是一个二层的以太通道组，范例 1-19 显示了 **show etherchannel** 命令的输出。

范例 1-19 show etherchannel 命令的输出

```
3550_switch#show etherchannel 1 detail
Group state = L2
Ports: 2   Maxports = 8
Port-channels: 1 Max Port-channels = 1
                Ports in the group:
                -----
Port: Gi0/1
-----
Port state      = Up Mstr In-Bndl
Channel group = 1          Mode = On/FEC      Gcchange = 0
Port-channel = Po1         GC   = 0x00010001   Pseudo port-channel = Po1
Port index     = 0         Load = 0x00
Age of the port in the current state: 00d:03h:04m:31s
Port: Gi0/2
-----
Port state      = Up Mstr In-Bndl
Channel group = 1          Mode = On/FEC      Gcchange = 0
Port-channel = Po1         GC   = 0x00010001   Pseudo port-channel = Po1
Port index     = 0         Load = 0x00
Age of the port in the current state: 00d:03h:03m:17s
Port-channels in the group:
```

(待续)

```

Port-channel: Po1
-----
Age of the Port-channel   = 00d:03h:04m:33s
Logical slot/port        = 1/0           Number of ports = 2
GC                        = 0x00010001    HotStandBy port = null
Port state                = Port-channel Ag-Inuse
Ports in the Port-channel:
Index   Load   Port      EC state
-----+-----+-----+-----
  0      00     Gi0/1     on
  0      00     Gi0/2     on
Time since last port bundled:  00d:03h:03m:19s    Gi0/2
3550_switch#

```

为了验证以太通道组的 PAgP 状态，使用下面的命令：

```
show pagp [channel-group-number] {counters | internal | neighbor}
```

这个命令显示了 PAgP 的信息，例如流量信息、内部的 PAgP 配置和邻居的信息。

2. 配置三层的以太通道

为了配置三层的以太通道，应当建立端口组的逻辑接口，然后将以太接口放入端口组中。

no switchport 命令必须用在端口组上和物理接口上。用于建立三层以太通道组的步骤和语法如下：

第 1 步 配置端口组，关闭二层交换，并且给这个端口组分配 IP 地址，如下所示。

```

3550_switch(config)#interface port-channel [1-64]
3550_switch(config-if)#no switchport
3550_switch(config-if)#ip address address subnet_mask

```

第 2 步 配置以太通道组中的物理接口，将物理接口分配到端口组中，如下所示：

```

3550_switch(config)#interface interface_name
3550_switch(config-if)#no switchport
3550_switch(config-if)#channel-group [1-64] mode {auto [non-silent] | desirable
[non-silent] | on}

```

范例 1-20 演示了使用 IP 地址 172.16.50.1/24 配置三层以太通道。

范例 1-20 配置三层的以太通道

```

3550_switch(config)#interface port-channel 2
3550_switch(config-if)#no switchport
3550_switch(config-if)#ip address 172.16.50.1 255.255.255.0
3550_switch(config-if)#exit
3550_switch(config)#interface fast 0/17
3550_switch(config-if)#channel-group 2 mode auto
3550_switch(config-if)#interface fast 0/18
3550_switch(config-if)#no switchport
3550_switch(config-if)#channel-group 2 mode auto

```

3. 配置以太通道的负载均衡

可以对以太通道配置不同类型的负载均衡。可以使用两种类型的负载均衡：基于源和基于目的的转发方法。默认的负载均衡类型是 src-mac。以太通道在一个通道组中的链路负载均衡流量的方法是通过减少帧中地址的二进制部分产生一个数值来选择通道组中的

链路。

使用源 MAC 地址转发，当数据包通过以太通道转发时，是基于进入的数据包的源 MAC 地址在通道组中的端口进行分发。因此，为了提供负载均衡，从不同的主机发出的数据包使用通道组中不同的端口，但是从同一个主机发出的数据包使用通道组中相同的端口（从交换机学习到的同一个 MAC 地址不会改变）。

当使用源 MAC 地址转发方法时，对路由的 IP 流量也基于源和目的 IP 地址进行负载均衡。所有路由的 IP 流量基于源和目的 IP 地址选择端口。两个 IP 主机之间的流量总是使用通道组中的同一个端口，而另外一对主机之间的流量使用通道组中的不同端口。

使用目的 MAC 地址转发，当数据包通过以太通道转发时，是基于进入的数据包的目的主机的 MAC 地址在通道组中的端口进行分发，因此，到达同一个目的的数据包是从同一个端口进行转发，到达不同目的的数据包是从通道组中的不同端口转发的。

为了在以太通道上配置负载均衡，使用下面的全局配置命令：

```
3550_switch(config)#port-channel load-balance {dst-mac | src-mac}
```

为了验证实际生效的负载均衡的类型，使用 **show etherchannel load-balance** 命令。这个命令会显示正在使用的是 dst-mac 还是 src-mac 的负载均衡方法。

为了使以太通道负载均衡返回到默认的配置模式下，使用 **no port-channel load-balance** 全局配置命令。

五、第4步：控制STP和VLAN传播

下一步是可选的，但是在大型网络中是非常重要的。思科实施了一组特性允许交换机在小型网络中即插即用，但是在大型网络中有相反的效果，会产生大量的流量。例如基于每个 VLAN 的生成树协议（PVST），还有默认的设置中每一个 VLAN 都可以在骨干链路上进行通信，这些特性都会导致边界交换机被生成树请求和其他的广播流量淹没。

在图 1-17 的网络中，crane 交换机只有一个 VLAN，就是 VLAN 2。因为这个交换机和其他交换机在同一个 VTP 域里，然而，它也要参与 VLAN 3 和 VLAN 4 的生成树协议。实际上确实没有必要让交换机浪费资源去处理在交换机上根本就没有 VLAN 的生成树请求。网络越大冗余性越高，问题就会变得越严重。假设，例如你有 75 个边界交换机，那么每个边界交换机在骨干链路上就有 75 个单独的生成树拓扑。而且，所有的这一切都发生在用户流量能够使用交换机之前。

有一个通常的错误概念就是认为 *VLAN 剪枝* 会解决 STP 的问题。然而，VLAN 剪枝只会影响广播、组播和未知/泛洪的单播流量。基本上，STP 构建的是数据通过的路径或者是数据流过的大路，剪枝控制的是广播数据或者通过那条路径的“流量”。

思科提供了两种非常有效的方法来处理过度的广播和 STP。

- **VLAN pruning**——VLAN 剪枝说的是，如果启用了 VTP 剪枝，那么如果一个下游的交换机没有任何一个活动的接口在那个被剪枝的 VLAN 中，交换机就会阻止将泛洪的流量转发到那些下游的可剪枝的 VLAN 中。VTP 剪枝是一种流量控制方法，它可以减少不必要的广播、组播和未知单播的流量。VTP 剪枝阻止泛洪的流量沿着骨干链路到达包含在可剪枝列表中的那些 VLAN。如果 VLAN 被配置为“不可剪枝”，泛洪就会继续。



- ## 1. 配置 VTP 剪枝

```
3550_switch(vlan)#vtp pruning
```

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

剪枝的，而这个步骤是可以忽略的。可以通过使用下面的接口命令来标注某些特定的 VLAN 可以被剪枝：

```
3550_switch(config-if)#switchport trunk pruning vlan {add | except | none | remove} vlan_range
```

可以添加多个 VLAN，通过逗号分开，或者使用分界符 (-) 添加一组 VLAN。例如，接口命令 **switchport trunk pruning vlan add 2-10** 使得 VLAN 2 到 10 可剪枝。可以利用这个命令在骨干到骨干的基础上配置 VTP 剪枝。

全局的 VTP 剪枝是否启用，可以通过使用 **show vtp status** 命令进行验证。也可以通过使用 **show interface** 命令带有 **switchport** 关键字来验证对某个 VLAN 是否启用了 VTP 剪枝。范例 1-21 演示了如何使用 **show vtp status command** 命令来验证 VTP 剪枝是否启用。

范例 1-21 验证全局 VTP 状态

```
yin#show vtp status
VTP Version           : 2
Configuration Revision : 6
Maximum VLANs supported locally : 1005

Number of existing VLANs : 14
VTP Operating Mode       : Server
VTP Domain Name         : psv2
VTP Pruning Mode         : Enabled
VTP V2 Mode             : Disabled
VTP Traps Generation    : Disabled
MD5 digest               : 0x13 0xF9 0xA7 0x89 0x56 0x56 0x8D 0x54
Configuration last modified by 172.16.192.16 at 3-1-93 02:35:01
Local updater ID is 172.16.192.16 on interface Vl192 (lowest numbered VLAN interface found)
```

范例 1-22 演示了使用 **show interface** 命令来验证 VLAN 剪枝。**show interface** 命令是在接口命令 **switch-port trunk pruning vlan 2-1001** 命令在 yin 交换机上输入后执行的。

范例 1-22 验证 VLAN 剪枝

```
yin#show interfaces fast 0/20 switchport
Name: Fa0/20
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
```

2. 通过从骨干链路上清除 STP 来控制 STP

在中等到大型网络中，控制每一台交换机上有多少个 STP 实例以及有多少个 STP 实例通过骨干非常重要。回忆一下，默认情况下，每一个 VLAN 都有 STP 实例，思科把它称为 PVST+。

交换机会在它的所有骨干链路上对它认识到的每一个 VLAN 运行一个 STP 实例。如果网络上有 5 个 VLAN，那么就有 5 个 STP 的实例，每个实例都有一个单独的根桥，以此类推。每个 Catalyst 3550 交换机支持 128 个 STP 实例，其他交换机（例如 Catalyst 3548XL 和 2900XL）可以支持 64 个 STP 实例，这个数量会随着交换机的不同而不同。要了解你的交换机支持多少个 STP 实例，请参考 www.cisco.com。如果添加了更多的 VLAN，STP 对那个交换机上的某些 VLAN 可能关闭掉了。一个更常见的问题是在边界和布线柜交换机上多个生成树协议实例给设备带来的负荷。不幸的是，VTP 剪枝不会影响生成树协议。为了从骨干链路上清除 STP 实例，使用下面的接口命令：

```
Switch(config-if)#switchport trunk allowed vlan [add | all | except | remove] vlans_2-1001
```

- **add**——将下列 VLAN 添加到骨干链路上。
- **all**——将所有的 VLAN 添加到骨干链路上。
- **except**——清除所有的 VLAN，除了某些指定的 VLAN。
- **remove**——从骨干链路上清除下面这些 VLAN。

为了清除 VLAN 3 到 VLAN 6，你会使用下面看起来有些麻烦的命令：

```
Switch(config-if)#switchport trunk allowed vlan remove 3-6
```

图 1-18 显示了和图 1-17 相同的网络，但是接口的名字有所变化。在这个范例中，在 yin 的交换机上，所有的 VLAN 都被清除了，除了 VLAN 1 和 2，都在通往 crane 交换机的骨干链路上。

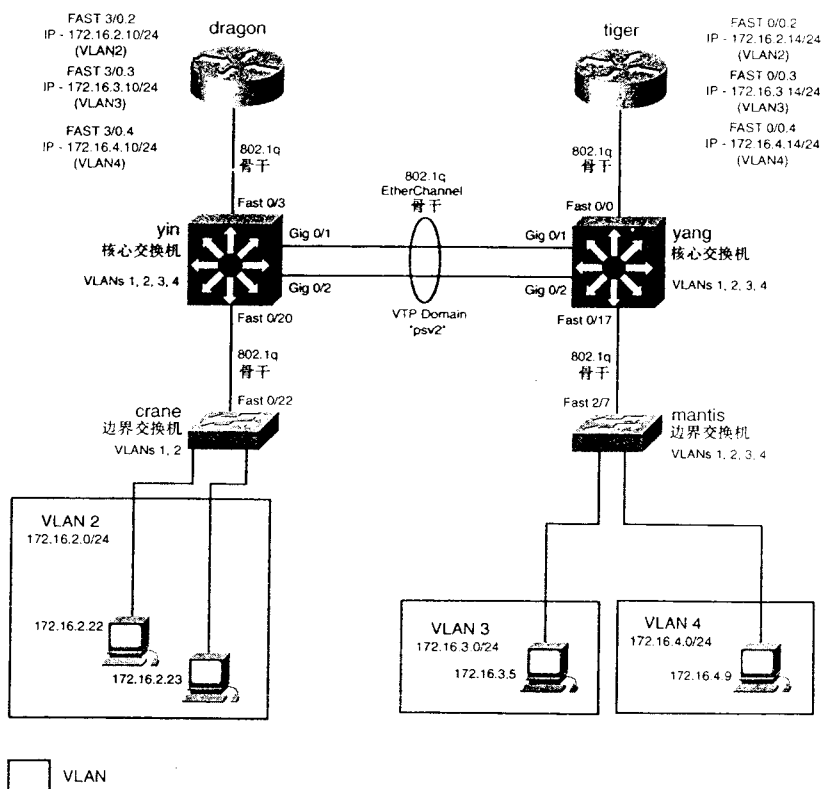


图 1-18 VLAN 骨干链路和 STP

在这个特别的范例中，为了从骨干链路上清除 STP，可以使用 **switchport** 命令。在从骨干链路上清除之前，检查 VLAN 3 的 STP 状态。范例 1-23 显示了在 yin 交换机上使用 **show spanning-tree** 命令。注意在底部 STP 正在通过 fast 0/3 接口向 dragon 路由器转发 VLAN3 的信息，fast 0/20 接口向 crane 交换机转发，po1 接口向以太通道端口转发。

范例 1-23 在 yin 交换机上使用 show spanning-tree 命令

```
yin#show spanning-tree vlan 3
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address    0004.275e.f0c8
             Cost        3
             Port        65 (Port-channel1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID   Priority    32771 (priority 32768 sys-id-ext 3)
             Address    000a.8a0e.ba80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time   300

Interface    Port ID           Designated           Port ID
Name          Prio.Nbr          Cost Sts             Cost Bridge ID        Prio.Nbr
-----
Fa0/3         128.3             19 FWD               3 32771 000a.8a0e.ba80 128.3
Fa0/20        128.16            19 FWD               3 32771 000a.8a0e.ba80 128.16
Po1           128.65            3 FWD                0 32768 0004.275e.f0c8 128.1
yin#
```

范例 1-24 显示了在 yin 交换机和 crane 交换机的骨干链路上清除 VLAN 3 到 1001。范例的第二部分显示了 VLAN 3 的生成树。注意 VLAN 3 不再从 Fa0/20 的骨干端口上转发，这是一条连接到 crane 交换机的骨干链路。

范例 1-24 从骨干链路上清除 VLAN

```
yin(config)#int fastEthernet 0/20
yin(config-if)#switchport trunk allowed vlan remove 3-1001
yin(config-if)#^Z
yin#show spanning-tree vlan 3
11:55:53: %SYS-5-CONFIG_I: Configured from console by console
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address    0004.275e.f0c8
             Cost        3
             Port        65 (Port-channel1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID   Priority    32771 (priority 32768 sys-id-ext 3)
             Address    000a.8a0e.ba80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time   15

Interface    Port ID           Designated           Port ID
Name          Prio.Nbr          Cost Sts             Cost Bridge ID        Prio.Nbr
-----
Fa0/3         128.3             19 FWD               3 32771 000a.8a0e.ba80 128.3
Po1           128.65            3 FWD                0 32768 0004.275e.f0c8 128.1
yin#
```

show interface interface_name switchport 命令也显示了哪些 VLAN 正在骨干链路上进行传输。

show interface trunk 命令是一个非常有用的命令，可以决定一条链路的骨干状态和 VLAN 状态。**show interface trunk** 命令列出了端口、它的模式、封装类型以及它是否是骨干。它还列出了在每个骨干链路上所允许的 VLAN 和这些 VLAN 的 STP 状态。范例 1-25 列出了 **show interface trunk** 命令的输出结果，表明 VLAN 3 到 VLAN 1001 不再在骨干链路 fast 0/20 上出现，而 VLAN 1002 到 VLAN 4094 是默认的 VLAN 和扩展范围的 VLAN。

范例 1-25 显示骨干链路上所允许的 VLAN

yin#show interface trunk				
Port	Mode	Encapsulation	Status	Native vlan
Fa0/3	on	802.1q	trunking	1
Fa0/20	on	802.1q	trunking	1
Po1	on	802.1q	trunking	1
Port	Vlans allowed on trunk			
Fa0/3	1-4094			
Fa0/20	1-2,1002-4094			
Po1	1-4094			
Port	Vlans allowed and active in management domain			
Fa0/3	1-4,10,20,30,40,50,192			
Fa0/20	1-2			
Po1	1-4,10,20,30,40,50,192			
Port	Vlans in spanning tree forwarding state and not pruned			
Fa0/3	1-4,10,20,30,40,50,192			
Fa0/20	1-2			
Po1	1,192			
yin#				

从骨干链路上清除 VLAN 是一种控制 STP 的方法，对于需要冗余的交换机来说，需要额外的方法来控制 STP。

注意：Catalyst 软件的新版本允许清除/移掉 VLAN 1。然而，许多交换机不允许清除/移掉 VLAN 1。从骨干链路上清除 VLAN 总是要特别注意。记住，这是 802.1Q 默认的本征 VLAN，而其他的协议可能正在 VLAN 1 上使用未打标签的帧。

3. 配置 STP 的负载均衡和根的设置

冗余的交换网络不能执行自动的负载均衡，因为 STP 的转发/阻塞决定是部分基于静态 MAC 地址的，对于所有的 VLAN，流量趋向于遵循相同的方向和同样的路径。这会导致某些链路过度使用，而另外一些链路保持空闲。图 1-19 演示了一个网络，所有的链路都汇聚在一个交换机上。yang 交换机是 VLAN 2、3、4 和 5 的 STP 根桥。

如果你想在 yin 和 yang 交换机之间负载均衡流量，或者你正在 dragon 和 tiger 路由器上使用 HSRP 协议，你可能想控制 STP 根桥的放置。例如，如果 dragon 路由器是 VLAN 2 的 HSRP 的主路由器，你可能想使流量通过 yin 交换机而不是 yang 交换机。为了在交换网络中

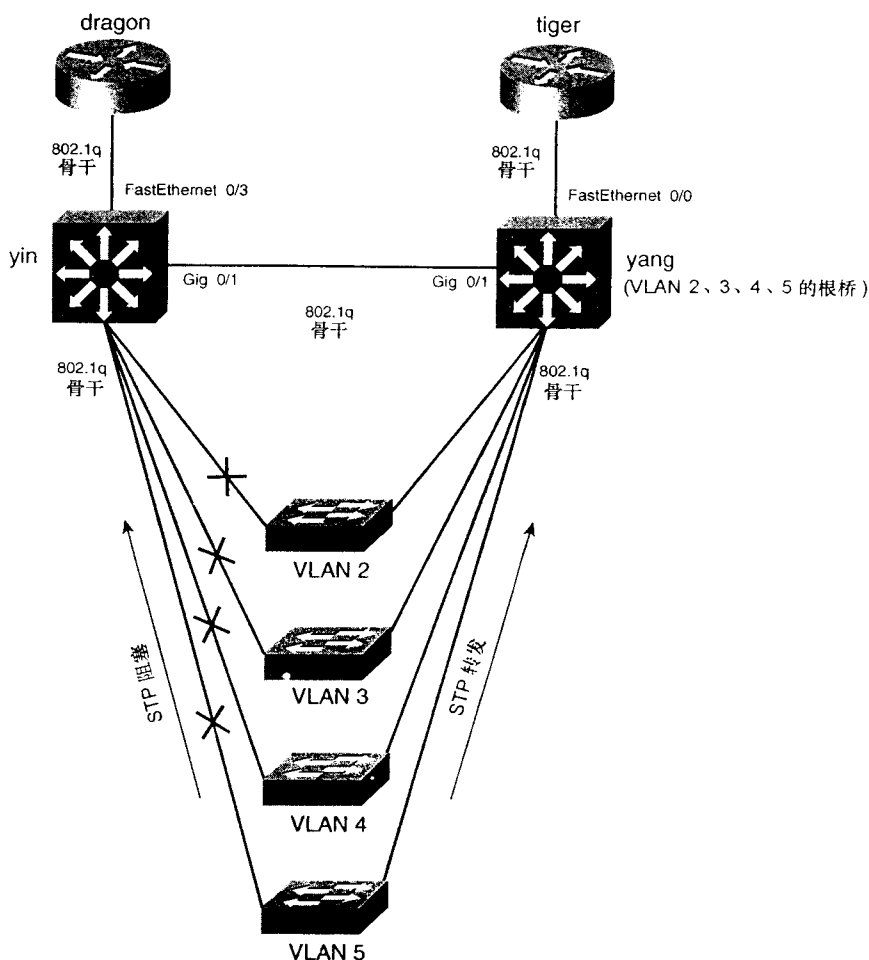


图 1-19 STP 的根桥

可以通过许多方法来配置 Catalyst 交换机的生成树的根桥。用来设置根桥的方法主要取决于你正在试图控制的环境。当设置根桥时，你实际上是告诉 STP 哪些端口置为转发状态，哪些端口置为阻塞状态。因为 STP 运行 PVST，每一个 VLAN 都有不同的根桥。在图 1-20 中，yin 交换机被设置为 VLAN 4 和 VLAN 5 的 STP 根桥，而 yang 交换机被设置为 VLAN 2 和 VLAN 3 的根桥。这使得边界交换机可以在骨干链路上更均匀地分担它们的流量。VLAN 4 和 VLAN 5 的流量转发到 yin 交换机，而 VLAN 2 和 VLAN 3 的流量转发到 yang 交换机。

在更深入地探讨设置 STP 的根桥之前，必须首先学会如何定义根桥放置的位置。**show spanning-tree root** 命令显示了每一个 VLAN 的根桥的概况。它会显示根桥的 MAC 地址、根端口、优先级、费用和对应那个 VLAN 的 STP 计时器。范例 1-26 显示了 **show span** 命令的输出。

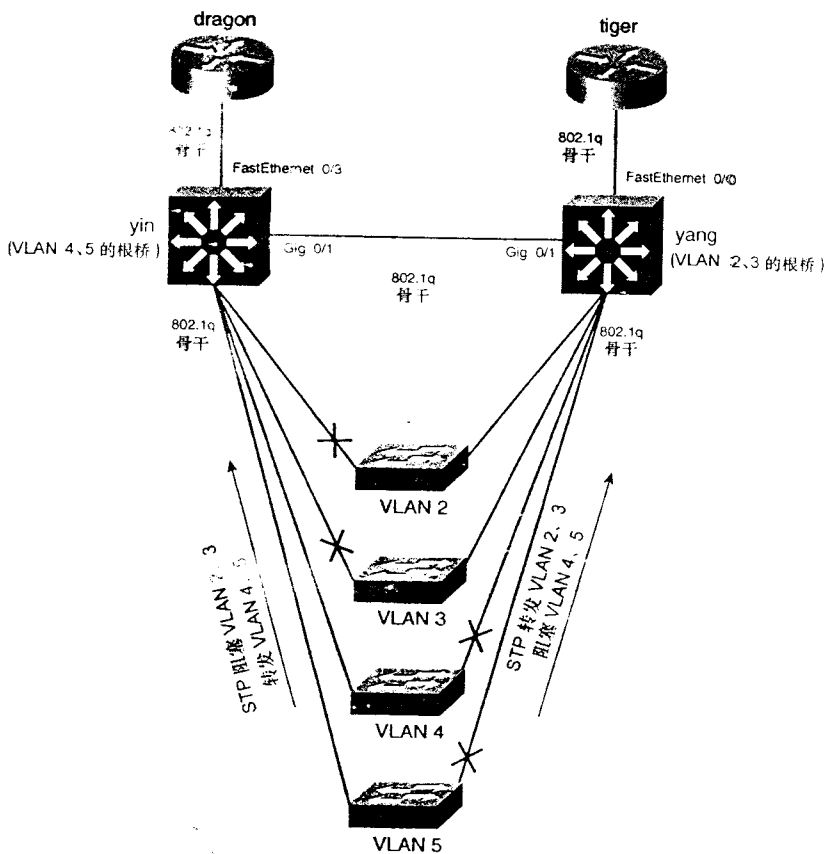


图 1-20 STP 的根桥

范例 1-26 查看 VLAN 2 的生成树

```
yin#show spanning-tree root
```

Vlan	Root ID	Root Cost	Hello Time	Max Age	Fwd Dly	Root Port
VLAN0001	32768 0004.275e.f0c0	3	2	20	15	Po1
VLAN0002	32768 0004.275e.f0c7	3	2	20	15	Po1
VLAN0003	32768 0004.275e.f0c8	3	2	20	15	Po1
VLAN0004	32768 0004.275e.f0c9	3	2	20	15	Po1
VLAN0005	32768 0004.275e.f0c1	3	2	20	15	Po1

```
yin#
```

show spanning-tree 命令和它的子命令 **show spanning-tree vlan** 显示了关于生成树协议的详细而有价值的信息。这个命令有一些变种，取决于你想查看多少信息。范例 1-27 列出了在 yin 交换机上使用 **show spanning-tree** 命令看到的 VLAN 2 的部分输出。

范例 1-27 查看 VLAN 2 的生成树协议

```
yin#show spanning-tree
VLAN0001
Spanning tree enabled protocol ieee
```

(待续)


```
Root ID Priority 32768
<<<text omitted>>>
VLAN0002
Spanning tree enabled protocol ieee
Root ID Priority 100
Address 0004.275e.f0c7
Cost 3
Port 65 (Port-channel1)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)
Address 000a.8a0e.ba80
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300

Interface Port ID Designated Port ID
Name Prio.Nbr Cost Sts Cost Bridge ID Prio.Nbr
-----
Fa0/3 128.3 19 FWD 3 32770 000a.8a0e.ba80 128.3
Fa0/20 128.16 19 FWD 3 32770 000a.8a0e.ba80 128.16
Po1 128.65 3 FWD 0 100 0004.275e.f0c7 128.1
<<<text omitted>>>
```

这个命令的每一位的信息都是非常有用的，这些字段定义如下：

- **Spanning tree type**（生成树类型）——正在使用的生成树的协议类型：IBM、DEC 或者 IEEE。
- **Root ID**（根 ID）——根桥的 MAC 地址。
- **Root ID Priority**（根桥 ID 优先级）——从根桥收到的桥的优先级。桥的优先级的范围是从 0~65 535，32 768 是一个默认的值。
- **Root ID Cost**（根 ID 的费用值）——到达根桥的累积的费用值。
- **Root ID Port**（根 ID 端口）——那个网段上的根端口。
- **Root Max Age, Hello Time, Forward Delay**——由根桥发送的 3 个 STP 计时器。
- **Bridge ID MAC ADDR**——被这个本地桥的 VLAN 使用的 MAC 地址。
- **Bridge ID Priority**——本地桥的优先级。
- **Bridge Max Age, Hello Time, Forward Delay**——在本地桥上的 3 个 STP 计时器。

最后几行显示了在这个 VLAN 内参与 STP 的每一个端口，并且列出了这个端口是转发还是阻塞状态，以及端口的费用和服务优先级。不要将这个优先级和生成树协议桥的优先级混为一谈。端口优先级的范围从 0~63，32 是端口的默认优先级。

show spantree summary 命令是一个非常有用的命令，它展示了关于生成树协议的通常的操作画面。这个命令提供了从 STP 的角度来看 VLAN 的状况和端口的状态。范例 1-28 列出了这个命令的输出。

范例 1-28 查看 VLAN 2 的生成树

```
3550_switch#show spanning-tree summary
Root Bridge for: none.
Extended system ID is enabled.
PortFast BPDU Guard is disabled
EtherChannel misconfiguration guard is enabled
UplinkFast is disabled
BackboneFast is disabled
Default pathcost method used is short
```

（待续）

Name	Blocking	Listening	Learning	Forwarding	STP Active
.....					
VLAN0001	0	0	0	5	5
VLAN0002	0	0	0	3	3
VLAN0003	0	0	0	2	2
VLAN0004	0	0	0	2	2
VLAN0005	0	0	0	2	2
.....					
5 vlans	0	0	0	14	14
yin#					

为了正确设置 STP 的根桥，我们回忆一下 STP 的四步决策过程以及生成树是如何决定根桥的。根桥是通过最低费用的 BID 来选举的。BID 由优先级和 MAC 地址组成。

1. 最低的根 BID（优先级和 MAC 地址，和根桥邻接）
2. 到达根桥的最低费用的路径值，到达根桥的所有路径的累积费用值
3. 最低的发送者 BID
4. 最低的端口 ID

从这个过程来看，可以在多个级别上影响根桥的选举。现在，你可能想让每一个端口都具有相同的 STP 优先级，而另外某个时候，你可能想让某个特定的端口具有更高的优先级，例如在负载均衡的环境下。表 1-12 显示了 4 种主要的 STP 选举状态以及 Catalyst 3550 交换机的全局配置命令。

表 1-12 以太 STP 配置结果

STP 选举状态	Catalyst 3550 配置命令
1. 最低的 BID	<code>+spanning-tree[vlan vlan_id][priority 0-65535]</code> <code>+spanning-tree vlan vlan_id root [primary/secondary] [diameter 2-7 [hello-time seconds]]</code>
2. 到达根桥的最低费用	<code>*spanning-tree[vlan vlan_id][cost 1-200000000]</code>
3. 最低的发送者 BID	<code>+spanning-tree[vlan vlan_id][priority 0-65535]</code>
4. 最低的端口 ID	<code>*spanning-tree[vlan vlan_id][port-priority 0-255]</code>

- 接口配置命令
- + 全局配置命令

可以通过很多方法影响根桥的选举过程。所选取的方法取决于你希望通过设置根桥达到什么样的目的。你用于影响根桥的选举过程越深入，那么你对没有控制权限的其他交换机上的 STP 配置和可能的折衷就有更高的安全防护。

- 全局的 `spanning-tree [vlan vlan_id] [priority 0-65535]`命令可以影响 BID 的优先级字段。优先级越低，交换机就越有可能选举为根桥。它可以在每一个 VLAN 上设置，也可以在整个交换机上全局地进行设置。VLAN ID 的有效值为 1~4094，有效的优先级的值为 4096, 8192, 12 288, 16 384, 20 480, 24 576, 28 672, 32 768, 36 864, 40 960, 45 056, 49 152, 53 248, 57 344 和 61 440。所有其他的值都被丢掉了。
- 全局命令 `spanning-tree vlan vlan_id root [primary|secondary] [diameter 2-7[hello-time seconds]]`是一个宏，类似于在 CAT OS 上用 `set root` 命令产生的宏。当此命令用 `primary` 关键字键入时，它检查在交换机上具有最高优先级的 VLAN，也就是根，并将自己的优先级设置得比它低。这个命令也可以调整 max age, hello 和 forwarding

delay 计时器。这个命令还可以使用扩展系统 ID。可选的 **diameter** 关键字指定任何两个终端站点之间交换机的最大数量。有效的范围是 2~7。可选的 **hello-time** 指定由根桥产生的配置信息的间隔时间，以秒计。范围是 1~10s，默认的为 2s。范例 1-29 演示了 **root** 宏命令的使用。

范例 1-29 使用生成树 root 宏命令

```
3550_switch(config)#spanning-tree vlan 192 root primary
vlan 192 bridge priority set to 24576
vlan 192 bridge max aging time unchanged at 20
vlan 192 bridge hello time unchanged at 2
vlan 192 bridge forward delay unchanged at 15
3550_switch(config)#
```

- 当输入这个命令时，在 VLAN 192 上的默认优先级是 32 768。因此，交换机把优先级设置得比它低（在这种情况下，是 24 576）。24 576 是一个惟一的值，它表明正在使用扩展系统 ID。如果这个优先级的值变为 8192，代表扩展系统 ID 没有被使用。
- 接口命令 **spanning-tree [vlan vlan_id] [cost 1-200000000]**影响的是接口的 STP 费用值。有效的 VLAN ID 是 1~4094，有效的费用值范围是 1~200 000 000。表 1-13 列出了默认的 STP 费用。

表 1-13 局域网链路的 STP 费用值

带宽	修订的 IEEE STP 费用值	带宽	修订的 IEEE STP 费用值
4 Mbit/s	250	155 Mbit/s	14
10 Mbit/s	100	622 Mbit/s	6
16 Mbit/s	62	1 Gbit/s	4
45 Mbit/s	39	10 Gbit/s	2
100 Mbit/s	19		

- 接口命令 **spanning-tree [vlan vlan_id] [port-priority 0-255]**配置一个接口的端口优先级。默认的端口优先级是 128，有效的范围是 0~255。这个数字越低，优先级越高。表 1-14 列出了默认的 STP 配置。

表 1-14 默认的 STP 配置

特性	默认设置	特性	默认设置
启用状态	在 VLAN 1 上启用	STP 端口费用	参看表 1-12
	128 个 STP 实例/每个交换机	Hello 计时器	2 s
交换机/桥优先级	32768	Forward delay 计时器	15 s
STP 端口优先级	128	Maximum aging 计时器	20 s

STP 的 hello、forward delay 和 max age 计时器可以通过下面的全局配置命令进行配置和调整。任何时候你配置 STP 计时器都要特别注意。PVST+在每一个 VLAN 上都运行一个 STP 实例，如果你在一台交换机上对某个 VLAN 修改了计时器，那么你就需要在所有的交换机上

对那个特定的 VLAN 修改计时器。

- **spanning-tree vlan *vlan-id* hello-time {1-100}**
- **spanning-tree vlan *vlan-id* forward-time {15-300}**
- **spanning-tree vlan *vlan-id* max-age {6-40}**

对于许多部分来说，在 Catalyst 3550 交换机上配置 STP 和在 Catalyst 3500XL/2900XL 系列的交换机上配置 STP 是一个非常类型的过程。关于详细 STP 步骤和常规的交换机配置，请返回参考《CCIE 实验指南（第 1 卷）》的第 2 章。

六、第 5 步：配置交换虚拟接口（SVI）

最后三步过程是可选的，它们是配置 SVI（路由接口和三层交换）。

1. 配置交换机管理。
2. 配置 VTP 和 VLAN，并将端口接口分配到 VLAN 中。
3. 使用以太网通道、802.1Q 和 ISL 封装配置交换机之间的连接。
4. （可选）控制 STP 和 VLAN 的传播。
5. （可选）配置 SVI。
6. （可选）配置路由端口。
7. （可选）配置三层交换。

回忆一下，SVI 是交换机上的逻辑虚拟接口，它类似于管理接口。一个 SVI 代表的是一个 VLAN，就像对路由的端口或者交换机上的桥接功能。只有一个 SVI 和一个 VLAN 相关联。SVI 可以用于实现 VLAN 之间的路由，或者作为回退桥接的入路由协议实现 VLAN 之间的桥接，或者是对交换机的管理提供一种 IP 主机的连接功能。

默认情况下，SVI 是出于对默认 VLAN（INT VLAN 1）进行管理建立的，其他的 SVI 是用下面的全局配置命令建立的：

```
3550_switch(config)#interface vlan [1-4094]
3550_switch(config-if)# ip address IP_address subnet_mask
```

当建立完 SVI 后，可以给这个接口添加 IP 地址，并且定义诸如 HSRP 或者是访问控制列表这类特性。将 SVI 看作就像路由器上的三层接口一样的接口。关于 SVI 最通常的使用就是管理和实现 VLAN 之间的路由。

注意：为了在三层模式中使用 SVI 或者通过 SVI “路由”，必须在交换机上安装 EMI 的操作系统。

在图 1-21 中，有安装了 EMI 软件的思科 3550 交换机。在交换机上有两个 VLAN：VLAN 2 和 VLAN 10。VLAN 10 有工作站在 172.16.10.0/24 的 IP 子网里，而 VLAN 2 有工作站在 IP 子网 172.16.2.0/24 里。在这个范例里，建立了两个 SVI（接口 VLAN 2 和接口 VLAN 10），并且在适当的 VLAN 范围内分配了 IP 地址。

范例 1-30 演示了如何配置两个 SVI 和分配 IP 地址。

如果有接口在 VLAN 2 或者 VLAN 10 里，或者骨干链路是激活的，就能够 ping 通这个接口。也可以使用 **show interface** 命令和 **show ip interface** 命令来查看这个接口。

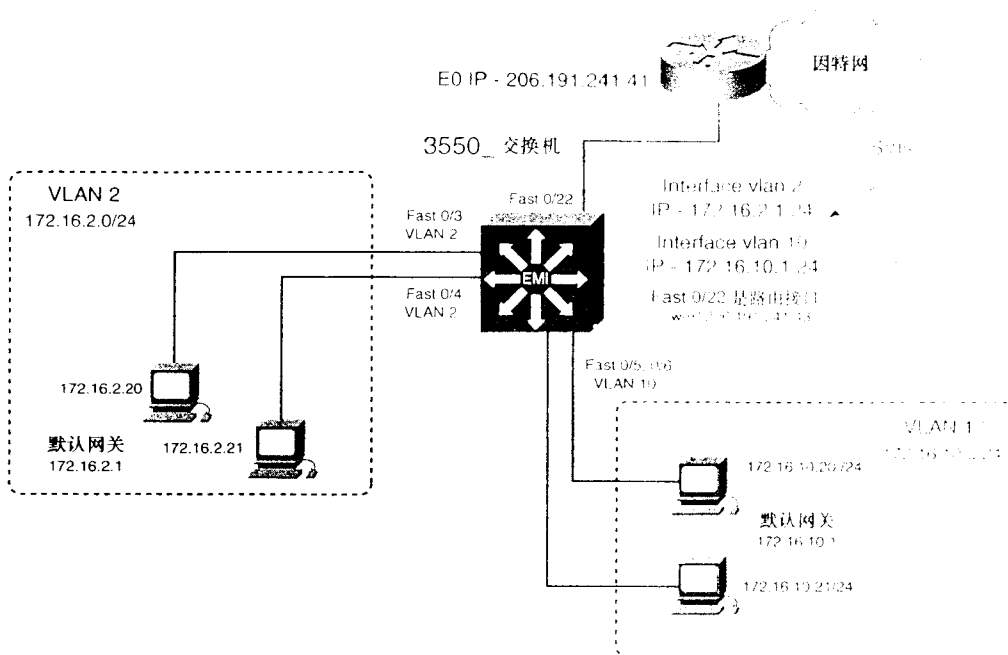


图 1-21 SVI 的配置

范例 1-30 配置 SVI

```
3550_switch(config)#interface vlan 2
02:05:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan2, changed state to up
3550_switch(config-if)#ip address 172.16.2.1 255.255.255.0
3550_switch(config-if)#exit
3550_switch(config)#interface vlan 10
02:06:17: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
3550_switch(config-if)#ip address 172.16.10.1 255.255.255.0
```

虽然 SVI 接口启用了，可以 ping 通它，但是还没有因特网和 IP 的连接。为了使这个 VLAN 中的工作站能够访问因特网，并且能够访问其他的用户，三层交换功能必须在交换机上启用。可以通过使用全局命令 **ip routing** 来启用交换机上的三层交换。一旦启用后，还必须配置一个路由选择协议实现 IP 的连接。范例 1-31 显示了用于实现 IP 完全连接的配置。

范例 1-31 启用路由/三层交换

```
3550_switch(config)#ip routing
3550_switch(config)#router eigrp 2003
3550_switch(config-router)#network 172.16.0.0
3550_switch(config-router)#network 206.191.241.0
3550_switch(config-router)#no auto-summary
```

使用 **show ip route** 命令，可以验证 SVI 的状态。SVI 的管理距离是 0，表现为一条直连的路由。范例 1-32 显示了 3550 交换机的路由/转发表。

对那个特定的 VLAN 修改计时器。

- **spanning-tree vlan *vlan-id* hello-time [1-40]**
- **spanning-tree vlan *vlan-id* forward-time [14-40]**
- **spanning-tree vlan *vlan-id* max-age [6-40]**

对于许多部分来说，在 Catalyst 3550 交换机上配置 STP 和在 Catalyst 3500XL/2900XL 系列的交换机上配置 STP 是一个非常类似的过程。关于详细 STP 生成和常规的交换机配置，请返回参考《CCIE 实验指南（第1卷）》的第2章。

六、第5步：配置交换虚拟接口（SVI）

最后三步过程是可选的，它们是配置 SVI（路由端口和三层交换）。

1. 配置交换机管理。
2. 配置 VTP 和 VLAN，并将端口接口分配到 VLAN 中。
3. 使用以太网通道、802.1Q 和 ISL 封装配置交换机之间的连接。
4. （可选）控制 STP 和 VLAN 的传播。
5. （可选）配置 SVI。
6. （可选）配置路由端口。
7. （可选）配置三层交换。

回忆一下，SVI 是交换机上的逻辑虚拟接口，它类似于物理接口。一个 SVI 代表的是一个 VLAN，就像对路由的端口或者交换机上的桥接功能。只有一个 SVI 和一个 VLAN 相关联。SVI 可以用于实现 VLAN 之间的路由，或者作为同构桥接的无路由协议实现 VLAN 之间的桥接，或者是对交换机的管理提供一种 IP 主机的连接功能。

默认情况下，SVI 是出于对默认 VLAN（INT VLAN 1）进行管理建立的，其他的 SVI 是用下面的全局配置命令建立的：

```
3550_switch(config)#interface vlan [1-4094]
3550_switch(config-if)# ip address IP_address subnet_mask
```

当建立完 SVI 后，可以给这个接口添加 IP 地址，并且定义诸如 HSRP 或者是访问控制列表这类特性。将 SVI 看作就像路由器上的三层接口一样的接口。关于 SVI 最通常的使用就是管理和实现 VLAN 之间的路由。

注意：为了在三层模式中使用 SVI 或者通过 SVI “路由”，必须在交换机上安装 EMI 的操作系统。

在图 1-21 中，有安装了 EMI 软件的思科 3550 交换机。在交换机上有两个 VLAN：VLAN 2 和 VLAN 10。VLAN 10 有工作站在 172.16.10.0/24 的 IP 子网里，而 VLAN 2 有工作站在 IP 子网 172.16.2.0/24 里。在这个范例里，建立了两个 SVI（接口 VLAN 2 和接口 VLAN 10），并且在适当的 VLAN 范围内分配了 IP 地址。

范例 1-30 演示了如何配置两个 SVI 和分配 IP 地址。

如果有接口在 VLAN 2 或者 VLAN 10 里，或者骨干链路是激活的，就能够 ping 通这个接口。也可以使用 **show interface** 命令和 **show ip interface** 命令来查看这个接口。

范例 1-32 查看路由/转发表里的 SVI

```
3550_switch#show ip route
<<<text omitted>>>
Gateway of last resort is 206.191.241.41 to network 0.0.0.0
172.16.0.0/24 is subnetted, 4 subnets
C    172.16.10.0 is directly connected, Vlan10
C    172.16.2.0 is directly connected, Vlan2
C    206.191.241.43 is directly connected, FastEthernet0/22
D*EX 0.0.0.0/0 [170/537600] via 206.191.241.41, 1d04h, FastEthernet0/22
3550_switch#
```

七、第 6 步：（可选）配置路由端口

路由端口就是 Catalyst 3550 交换机上的物理端口，在功能上类似于思科路由器上的物理接口。这是一种最简单看待它的方法。可以在它上面配置许多和路由器的物理接口相同的特性，包括 IP 地址、访问控制列表和 HSRP 组中的成员关系。路由端口不能有 VLAN 的子接口，或者是被配置成任何类型的骨干链路。配置路由端口需要在交换机上安装 EMI 软件。

图 1-22 演示了两个等同的网络。上面的网络有 3 个 Catalyst 3550 交换机，采用快速以太网路由端口和 3 个交换机相连。下面的网络有 3 个思科 2620 路由器，采用路由器的快速以太接口和交换机相连。

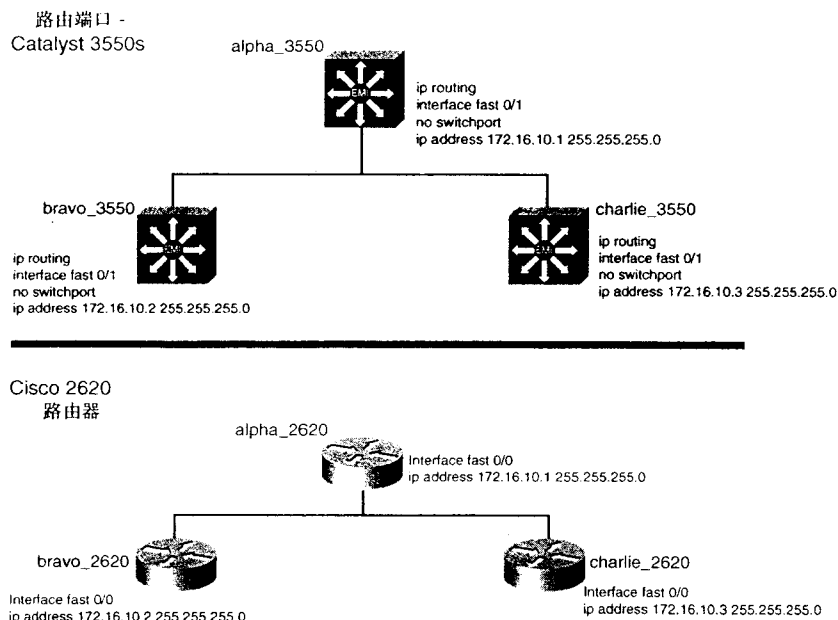


图 1-22 路由端口的比较

路由端口可以通过使用接口命令 **no switchport** 来启用。**no switchport** 命令有效地关闭了那个接口的交换功能。

交换机会使用一个内部 VLAN 来映射路由端口。这个内部 VLAN 也会用于扩展 VLAN，

但是它们不会相互矛盾。交换机所选择的内部 VLAN ID 可以通过 **show vlan internal usage**

命令查看。范例 1-33 演示了两个路由端口的配置，紧接着 **show vlan internal usage** 命令将显示交换机将路由端口分配给哪个 VLAN。

范例 1-33 配置路由端口

```
3550_switch(config)#interface fast 0/7
3550_switch(config-if)#no switchport
02:06:22: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/7, changed to
down
02:06:23: %LINK-3-UPDOWN: Interface FastEthernet0/7, changed state to down
02:06:26: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/7, changed to
up
3550_switch(config-if)#ip address 172.16.200.16 255.255.255.0
3550_switch(config-if)#interface fast 0/8
3550_switch(config-if)#no switchport
3550_switch(config-if)#
02:06:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/8, changed to
down
02:06:23: %LINK-3-UPDOWN: Interface FastEthernet0/8, changed state to down
02:06:26: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/8, changed to
up
3550_switch(config-if)#ip address 172.16.201.16 255.255.255.0
3550_switch(config-if)#^Z
3550_switch
3550_switch#show vlan internal usage
VLAN Usage
.....
1017 -
1025 FastEthernet0/7
! Internal VLANs used
1026 FastEthernet0/8
```

如果你使用 **no switchport** 接口命令将一个端口/接口从交换端口切换到路由端口，而你又想从路由端口切换回交换端口，那么必须键入接口命令 **switchport**，无需子命令。

八、第7步：(可选)配置三层交换

三层交换是一种做出三层转发决定并且将三层的数据包以二层的速度转发的能力。三层交换实际上就是路由。另外，一种更简单的定义三层交换的方法就是在同一个硬件平台上快速路由和交换的能力。当启用了 IP 路由时，Catalyst 3550 有效地成为了一个快速和高效的多端口路由器。当 IP 路由启用后，IP 路由中许多现有的 IP 特性现在一样可用。思科已经保留了所有的 IP 配置和相关命令的语法，并且将它们平滑地集成到了传统的思科 IOS 软件中。如果你知道如何配置思科路由器，那么从这点上说，配置 3550 的三层交换或者路由部分实际上就是在配置路由器。因为 3550 交换机上支持扩展的 IOS 特性，而不是所有的 IP 特性，例如数据链路交换(DLSw)。参看附录 A “思科 IOS 软件的限制和约束” 来了解 3550 不支持的一些命令的列表。

看见的并不总是可信的

我是一个使用？来获取帮助的坚定的信仰者。它总是能够在语法和显示某些新的可用特性方面给予我帮助。但是在使用 3550 的帮助时，要特别注意，许多在帮助中出现的表项实际上并不能进行配置。例如，在 IOS 12.1 (9) EA1c 中，你会看到诸如边界网关协议 (BGP) 和按需路由 (ODR) 的特性，但是你在试图配置它们时，会得到一个错误提示。

附录 A 包括一个限制的列表。关于最新一些特性的范围和限制，参考 www.cisco.com。

如想配置三层交换，遵循下面的 3 个步骤：

- 第 1 步 配置 3 种类型的三层接口中的一种，并且给它分配 IP 地址。Catalyst 3550 的路由背板可以辨别 3 种类型的三层接口：路由端口、SVI 的三层接口和三层的以太网通道接口。
- 第 2 步 使用全局配置命令 **ip routing** 启用 IP 路由。
- 第 3 步 配置内部网关协议（IGP）和其他的 IP 功能。支持的 IGP 协议有 RIP v1 和 v2、内部网关路由协议（IGRP）、增强的 IGRP 以及开放最短路径优先协议（OSPF）。交换机上的内部路由选择协议和路由器上的配置方法是相同的。出于这个原因，路由选择协议方面的内容就不在这里讨论了。关于配置 IGP 的更多信息，参考《CCIE 实验指南（第 1 卷）》。

实例 1：配置 SVI、路由端口和三层交换

在图 1-23 的网络模型中，有一个 Catalyst 3550 交换机，也就是 dragon 交换机，充当整个网络的核心路由器和交换机。dragon 交换机使用两个 SVI，一个是 VLAN 10，另外一个 VLAN 100，并实现了 VLAN 之间的路由。工作站端口例如 Fast0/7 被配置成为一个 VLAN 的接入端口，Fast0/8 接口充当一个路由端口并且和 dragon 路由器相连。路由端口的 IP 地址为 172.16.200.1/24。IP 路由已在 dragon 交换机上启用了，在自治系统 2003 里，使用 EIGRP 作为它的路由选择协议。

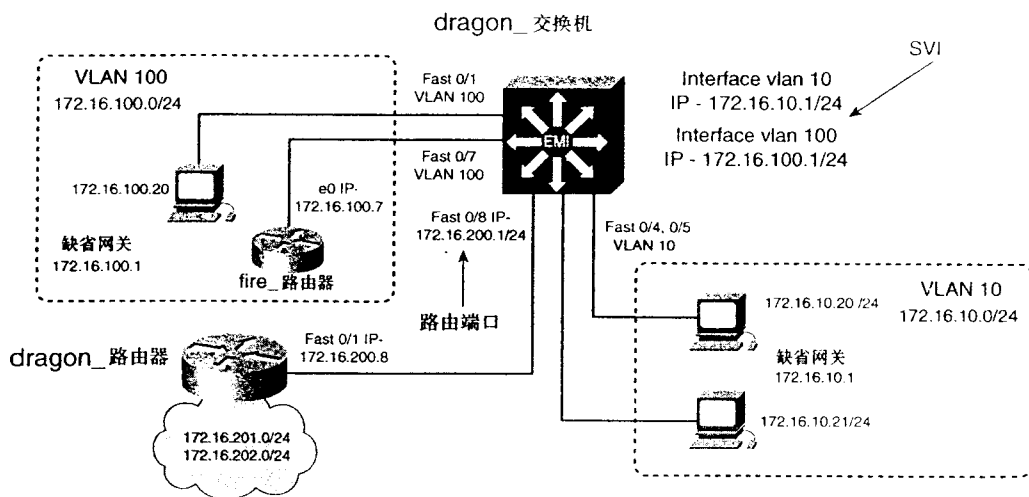


图 1-23 路由端口和 SVI 配置

范例 1-34 列出了 dragon 交换机的相关部分。

范例 1-34 dragon 交换机的配置

```
hostname dragon_switch
!
ip subnet-zero
ip routing
! Routing enabled
```

（待续）

```
!
spanning-tree extend system-id
! Extended System ID in use
!
interface FastEthernet0/1
  switchport access vlan 100
! VLAN 100
  no ip address
!
<<<text omitted>>>
!
interface FastEthernet0/4
  switchport access vlan 10
! VLAN 10
  no ip address
!
interface FastEthernet0/5
  switchport access vlan 10
! VLAN 10
  no ip address
!
interface FastEthernet0/6
  no ip address
!
interface FastEthernet0/7
  switchport access vlan 100
! VLAN 100
  no ip address
!
interface FastEthernet0/8
  no switchport
! Routed Port/interface
  ip address 172.16.200.1 255.255.255.0
! IP address
!
<<<text omitted>>>
!
interface Vlan1
! Default VLAN
  no ip address
! not used!
  shutdown
!
interface Vlan10
! SVI 10
  ip address 172.16.10.1 255.255.255.0
! IP address
!
interface Vlan100
! SVI 100
  ip address 172.16.100.1 255.255.255.0
! IP address
!
router eigrp 2003
! Routing Protocol
  network 172.16.0.0
! EIGRP on networks 172.16.0.0/16
  no auto-summary
  no eigrp log-neighbor-changes
!
```

在这个网络中,dragon 交换机通过 EIGRP 为所有的 VLAN 实现路由。VLAN 10,VLAN 100

和 IP 子网 172.16.200.0/24、172.16.201.0/24 和 172.16.202.0/24 都实现了彼此之间的 IP 可达性。dragon 交换机有两个 EIGRP 邻居，一个邻居是 fire 路由器，是使用 SVI VLAN 100 通过接入接口 Fast0/7 形成的。另外一个邻居是 dragon 路由器，是交换机通过路由端口 Fast0/8 和它形成的。

范例 1-35 列出了 dragon 交换机的路由/转发表，随后是 show ip eigrp neighbor 命令。

范例 1-35 dragon 交换机的配置

```
dragon_switch#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
C       172.16.200.0 is directly connected, FastEthernet0/8
D       172.16.201.0 [90/156160] via 172.16.200.8, 00:00:16, FastEthernet0/8
D       172.16.202.0 [90/156160] via 172.16.200.8, 00:00:09, FastEthernet0/8
C       172.16.10.0 is directly connected, Vlan10
C       172.16.100.0 is directly connected, Vlan100
dragon_switch#
dragon_switch#show ip eigrp neighbors
IP-EIGRP neighbors for process 2003
H   Address                Interface    Hold Uptime    SRTT    RTO   Q   Seq Type
  Address                (sec)       (ms)          Cnt Num
2   172.16.100.7           Vlan100     12 00:03:06      1    200   0   11
1   172.16.200.8           Fa0/8       14 00:03:40      1    200   0   9
dragon_switch#
!
```

到目前为止，你可以看到 Catalyst 3550 智能以太网交换机强大的功能和灵活的平台，以及思科为什么将它称之为智能交换机。因为 Catalyst 3550 的多样性和它能够执行的所有可能的软件配置，本章不可能完全覆盖它们。我们并不是去写一部关于 3550 的微型小说。相反，本章的目的是奠定配置 3550 交换机的某些基本和高级特性的必要基础。三层功能——例如路由选择协议、HSRP、IP 访问控制列表，等等——实际上和配置思科路由器几乎是一样的。你从其他来源所学习到的路由选择协议，例如《CCIE 实验指南（第 1 卷）》和其他参考书籍，可以很容易地迁移并应用到 3550 交换机上。

本章的剩余部分覆盖了 Catalyst 3550 交换机的某些额外和可选的特性。再次说明，因为 Catalyst 3550 的大量配置选项，所以不可能用一章来覆盖它们。为了覆盖某些主题，例如组播，它们值得这样做，也必须去覆盖它们，就需要 100 页的篇幅，这已经超出了本书的范围（然而，它们很重要，值得去研究）。下面的列表包括了 Catalyst 3550 交换机上的其他特性，使得它成为了已有的最灵活和最强大的平台：

- 二层和三层组播、IGMPv2、思科组管理协议（CGMP）和组播 VLAN 注册协议（MVR）；
- 802.1X 基于端口的认证；
- 使用 802.1Q 和 802.1p 的语音 VLAN；
- SPAN 和远程 SPAN（RSPAN）；
- SNMP 和 RMON；

- 802.1Q 隧道;
- 服务质量。

1.4.6 在 Catalyst 3550 以太网交换机上配置高级特性

生成树协议经过若干年依旧在许多网络中隐藏的主干协议，最终越来越体现了它的作用。因为 STP 所扮演的重要角色，50s 的收敛时间——20s 的最大老化时间计时器过期加上紧随的 15s 侦听和 15s 学习状态——对于许多现代网络来说实在是太长了。思科提供了许多方案，我们拿出一些来讨论，它们可以减轻收敛时间长的问题并稳定 STP。Catalyst 3550 交换机的某些高级特性如下所示：

- PortFast（端口加速）、BPDU 防护和 BPDU 过滤；
- UplinkFast（上行链路加速）；
- BackboneFast（骨干加速）；
- 根防护；
- IEEE 802.1w 快速生成树协议（RSTP）；
- IEEE 802.1s 多生成树协议（MST）；
- VLAN 映射；
- 具有单播和组播阻塞功能的 VLAN 保护端口。

我们将详细介绍这些特性。

一、配置生成树协议的 PortFast（端口加速）和 BPDU Guard（BPDU 防护）

生成树协议的端口加速功能只应该在边界交换机上配置。一旦本地发生故障，或者在初始化过程中，15s 的侦听和 15s 的学习状态就会被忽略掉。所有的端口会被置为永久的转发状态。出于这个原因，端口加速只应当在终端工作站（例如工作站和服务器）上使用。默认状态下，STP 的端口加速是关闭的，它可以使用下面的接口配置命令启用：

```
3550_switch(config-if)#spanning-tree portfast [disable]
```

关键字 **disable** 去除端口加速的配置或者关闭它。

端口加速也可以对所有的非骨干端口使用下面的全局配置命令启用：

```
3550_switch(config)#spanning-tree portfast default
```

要注意端口加速在全局级别上启用之前，要将适当的终端工作站连接到所有的端口上。可以使用 **show spanning-tree interface interface_name portfast** 命令来验证端口加速的配置。

注意：端口加速只能在终端工作站连接到交换机端口的情况下使用。如果端口加速在连接了其他网络设备（例如一台交换机）的情况下使用，那么可能会导致环路。当在 Catalyst 3550 上启用了端口加速时，你会得到下面这些信息：

```
%Warning: PortFast should only be enabled on ports connected to a single host.
Connecting hubs, concentrators, switches, bridges, etc. to this interface when
PortFast is enabled can cause temporary bridging loops.
Use with CAUTION
%Portfast has been configured on FastEthernet0/7 but will only have effect when
the interface is in a nontrunking mode.
```

端口加速启用的端口依旧能够参与 STP，并且能够发送和接收 BPDU 数据包。如果一个端口加速的端口无意中连接到另外一台交换机，就会产生 STP 的环路。思科实施了两种防止这种情况发生的特性来帮助完成端口加速：BPDU guard 和 BPDU filtering（BPDU 防护和 BPDU 过滤）。

- **BPDU guard（BPDU 防护）**——BPDU 防护强制的一个规则就是启用的端口不应当接收任何 BPDU 数据包。如果收到了 BPDU，就代表这个端口和交换机相连，可能会发生 STP 的环路。如果 BPDU 防护启用的端口收到了一个 BPDU 数据包，那么它会将这个端口置于错误的关闭状态（error-disabled）。默认情况下，BPDU 防护在所有的接口上是关闭的，如果启用了端口加速功能的话，就应该将其启用起来。它可以全局启用，也可以基于每一接口启用，使用下面的命令：

```
3550_switch(config)#spanning-tree portfast bpduguard default
```

为了在一个接口上启用或者关闭 BPDU 防护，使用下面的接口命令：

```
3550_switch(config-if)#spanning-tree bpduguard {enable | disable}
```

可以使用 **show spanning-tree summary** 命令来验证 BPDU 防护的功能。

- **BPDU filtering（BPDU 过滤）**——BPDU 过滤防止端口加速启用的端口发送或者接收 BPDU 数据包，有一个小小的例外。当链路初始化时，在被 BPDU 过滤之前，会有少量的 BPDU 数据包被发送。再次强调要特别注意这个特性：关闭发送和接收 BPDU 数据包的功能，实际上是关闭掉了那个接口的 STP 功能。因此，和先前提到的警告一样再次提出，确保没有交换机、集线器、桥等诸如此类的设备和这个接口相连。默认情况下，BPDU 过滤在所有的接口上是关闭的，如果启用了端口加速功能的话，就应该将其启用起来。它可以全局启用，也可以基于每一接口启用，使用下面的命令：

```
3550_switch(config)#spanning-tree portfast bpdufilter default
```

为了在一个接口上启用或者关闭 BPDU 过滤，使用下面的接口命令：

```
3550_switch(config-if)# spanning-tree bpdufilter {enable | disable}
```

可以通过使用 **show spanning-tree detail** 命令来验证 BPDU 过滤。在输出的尾部，你可以看到 BPDU 发送和接收的数量。接收的数量应当总是 0。发送的数量应当比较小，而且在 BPDU 过滤启用后，数量不应当增加。范例 1-36 显示了在接口 FastEthernet0/7 上启用了具有 BPDU 过滤和 BPDU 防护功能的端口加速后，**show spanning-tree detail** 命令的输出。

范例 1-36 检查生成树的细节

```
3550_switch#show spanning-tree detail
<<<text omitted>>>
VLAN0100 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, sysid 100, address 000a.8a0e.ba80
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
Number of topology changes 0 last change occurred 03:01:07 ago
Times: hold 1, topology change 35, notification 2
hello 2, max age 20, forward delay 15
```

（待续）

```
Timers: hello 0, topology change 0, notification 0, aging 300
Port 7 (FastEthernet0/7) of VLAN0100 is forwarding
Port path cost 100, Port priority 128, Port Identifier 128.7.
Designated root has priority 32868, address 000a.8a0e.ba80
Designated bridge has priority 32868, address 000a.8a0e.ba80
Designated Port id is 128.7, designated path cost 0
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
BPDUs: sent 11, received 0
! no BPDUs received
The port is in the portfast mode
! PortFast Enabled
3550_switch#
```

注意：端口加速、BPDU 防护和 BPDU 过滤可以用于 PVST+ 或者 MST 的环境。

二、配置上行链路加速

再次说到，生成树协议的 50s 收敛时间对现代网络有一定的影响。上行链路加速是思科对生成树协议的另外一种增强，主要适用于布线柜和边界交换机。它主要是用来加快边界交换机和核心交换机之间的收敛速度。图 1-24 演示了在一个常见的局域网中应当在什么地方使用端口加速、上行链路加速和骨干加速。

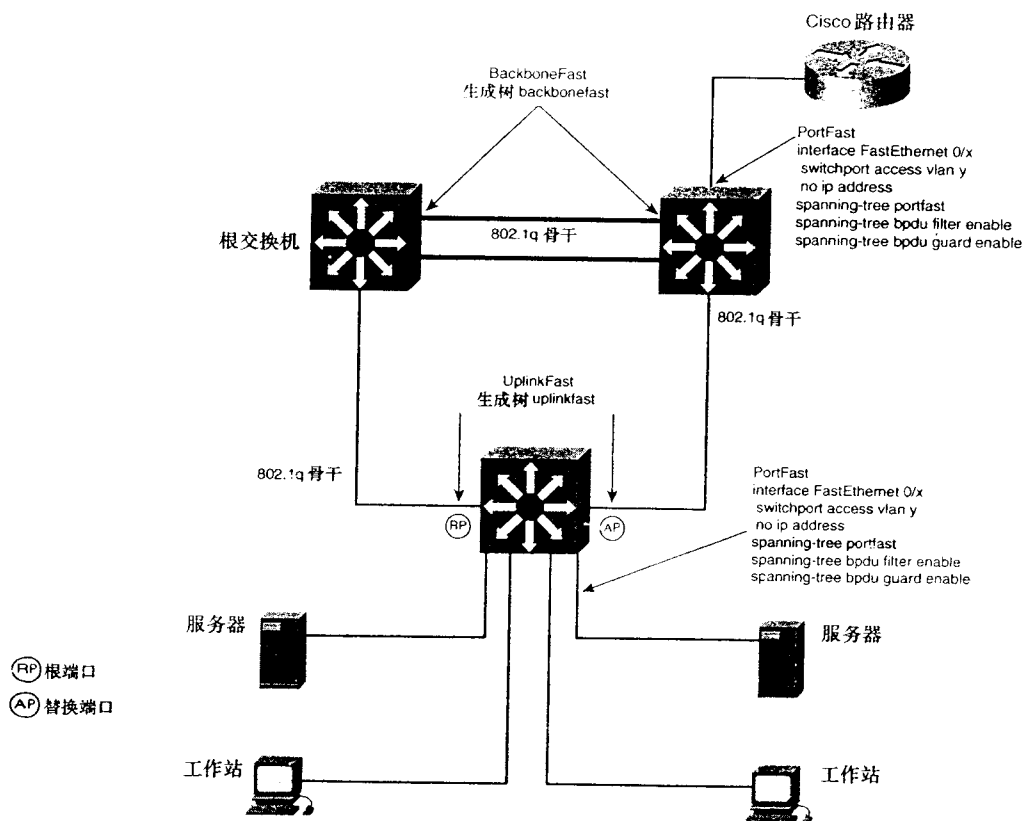


图 1-24 端口加速、上行链路加速和骨干加速的位置

上行链路加速工作在下述方式下。上行链路加速在交换机上全局启用并且影响交换机上所有的 VLAN。当这个功能启用后，交换机上所有的 VLAN 的优先级都被设为 49 152。所有端口的 VLAN 费用值在低于 3000 的情况下都会增加到 3000，以防止这个交换机成为根桥。根端口会立刻进入转发状态，而忽略 2 个 15s 的侦听和学习状态。在 VLAN 中的端口组成上行链路加速组。上行链路加速组中的一个端口处于转发状态，这个端口也就是根端口，而其余的端口都处于阻塞状态，也称为替换端口。当上行链路加速的一个端口监测到本地故障，它会将上行链路加速组中具有最低根路径费用值（仅高于最初根桥费用值）的端口置为转发状态，忽略两个 15s 的侦听和学习状态。一旦交换机将替换端口切换到转发状态，交换机就立刻在所有的转发端口上发送哑组播帧，对在本地代码地址识别逻辑（EARL）表中的每一个表项发送一次（除了和失败的根端口关联的表项）。EARL 是一个集中的处理引擎，可以通过 MAC 地址学习和转发数据包。默认情况下，大约每隔 100ms，就发送 15 个哑组播帧。每一个哑组播帧使用 EARL 表项中的工作站地址作为源 MAC 地址，而 01-00-0C-CD-CD-CD 作为这个组播帧的目的 MAC 地址。收到这些哑组播帧的交换机立刻修改 EARL 表里的表项，使每一个源 MAC 地址使用新的端口，允许交换机开始立刻使用新的路径。

如果到原始的根端口的连接恢复了，交换机会等待一个相当于 2 个转发延迟加 5s 的周期后将这个端口切换到转发状态。这个时间允许邻居端口经历侦听和学习状态。为了配置上行链路加速，使用下面的全局配置命令：

```
3550_switch(config)#spanning-tree uplinkfast [max-update-rate pkts/seconds]
```

在使用 **uplinkfast** 命令之前，将生成树协议的优先级设置为默认的值 32 768。如果 STP 的优先级已经修改了，那么将它修改回默认值。否则，**uplinkfast** 命令会失败。**uplinkfast** 命令是一个全局的命令，它会影响交换机上所有的 VLAN。不能基于每一个 VLAN 配置上行链路加速。可选的 **max-update-rate** 关键字是工作站地址更新发送的速率。默认的速率是每秒 150 个数据包。

注意：上行链路加速只能用在 PVST+。

可以使用 **show spanning-tree uplinkfast** 命令来验证上行链路加速的操作。这个命令显示了上行链路加速是否在接口上启用。它也列出了默认的计时器和统计数字。

三、配置骨干加速

骨干加速是思科的另外一种对 STP 的收敛时间进行提高的改进方案。骨干加速允许 STP 检测一个非直连的链路故障，并且在 30s 内使用它的备份路径。这个时间和默认的 STP 收敛时间 50s 相比已经极大地减少了。骨干加速通过使用内部的 BPDU 和某些智能和逻辑的推测来完成这个任务。

交换机通过从它的根端口或者被阻塞的端口收到来自指定桥的内部 BPDU 数据包来检测非直连的链路故障。先前的四步 BPDU 估计过程决定了这些 BPDU 是否是内部的。内部的 BPDU 指明，指定桥已经丢失了到根桥的连接。一个内部的 BPDU 表明一个桥既是根桥又是指定桥。在正常的生成树规则下，交换机会忽略内部的 BPDU 数据包，直到所配置的最大老化时间计时器到期。

交换机也试图检测它是否有一条备份路径可以到达根桥。如果内部的 BPDU 数据包到达

了一个阻塞的端口，根端口，交换机就可以断定它有一条到达根桥的备份路径。如果内部的 BPDU 数据包到达了根端口，所有阻塞的端口都成为到达根桥的备份路径。如果交换机有备份路径到达根桥，它使用这些备份路径传输一种新类型的 BPDU 数据包，叫做根链路查询 PDU。交换机从所有的备份路径发送根链路查询 PDU 给根桥，如果内部 BPDU 到达了根端口，但是没有阻塞的端口，交换机会认为它已经丢失了到根桥的连接。这会导致最大时间计时器过期，那么根据正常生成树的标准，交换机会成为根桥。

如果交换机有备份路径到达根桥，那么它会从所有的备份路径发送根链路查询 (RLQ) PDU 给根桥。如果交换机确定它仍然有到达根桥的备份路径，它就会使收到内部 BPDU 数据包的那些端口的最大时间计时器过期，如果所有到达根桥的备份路径表明交换机已经丢失了到根桥的连接，交换机就会使收到内部 BPDU 数据包的那些端口的最大时间计时器过期。如果一条或者多条备份路径还是可以到达根桥，交换机会使收到内部 BPDU 数据包的那些端口成为指定端口，并且从阻塞状态离开，如果处于阻塞状态，会经历侦听和学习的状态，接着进入转发状态。

注意：骨干加速只能用在 PVST+里，令牌环的 VLAN 或者第三方的交换机上并不支持。

骨干加速可以通过下面的全局配置命令启用：

```
3550_switch(config)#spanning-tree backbonefast
```

可以使用 **show spanning-tree summary** 命令来验证骨干加速的操作，正如范例 1-37 所示。

范例 1-37 验证 STP 的 UplinkFast 和 BackboneFast

```
3550_switch#show spanning-tree summary
Root Bridge for: VLAN0010, VLAN0100.
Extended system ID is enabled.
PortFast BPDU Guard is disabled
EtherChannel misconfiguration guard is enabled
UplinkFast is enabled
BackboneFast is enabled
Default pathcost method used is short
Name                Blocking Listening Learning Forwarding STP Active
-----
VLAN0001             1          0          0          4          5
VLAN0010             0          0          0          1          1
VLAN0100             0          0          0          1          1
-----
3 vlans              1          0          0          6          7
Station update rate set to 150 packets/sec.
UplinkFast statistics
-----
Number of transitions via uplinkFast (all VLANs)           : 2
Number of proxy multicast addresses transmitted (all VLANs) : 0
BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs)          : 0
Number of inferior BPDUs received (all VLANs)              : 0
Number of RLQ request PDUs received (all VLANs)            : 0
Number of RLQ response PDUs received (all VLANs)           : 0
Number of RLQ request PDUs sent (all VLANs)                : 0
Number of RLQ response PDUs sent (all VLANs)               : 0
3550_switch#
```


四、配置 STP 的根防护

根防护是 PVST+和 MST 中的一个特性，它可以防止局域网将一个不期望的交换机变为根桥。这个特性在集成两个局域网或者 VLAN 的情况下是非常有用的，它可以确保在当前局域网或者 VLAN 中的根桥，从而防止另外一个交换机成为当前网络的根桥。这个特性在服务提供商的网络里也可以提供额外的安全，防止用户网络中的设备成为服务提供商网络中的根桥。

图 1-25 演示了 STP 的根防护应当用在 VLAN 5 中的什么位置。STP 的根防护应当用于骨干链路的所有 VLAN 或接口上，但是出于本章的讨论目的，我们说的是 VLAN 5。在这个模型中，fire 交换机是 VLAN 5 期望的根桥，优先级是 32 768。外部网络可能是一个用户网络，通过 dragon 交换机相连，ranger 交换机的优先级为 8192，是那个网络中 VLAN 5 的根桥。为了防止 ranger 交换机成为 VLAN 5 的根桥，接口命令 **spanning-tree guard root** 用在了 dragon 交换机的 GigabitEthernet 0/1 接口上。

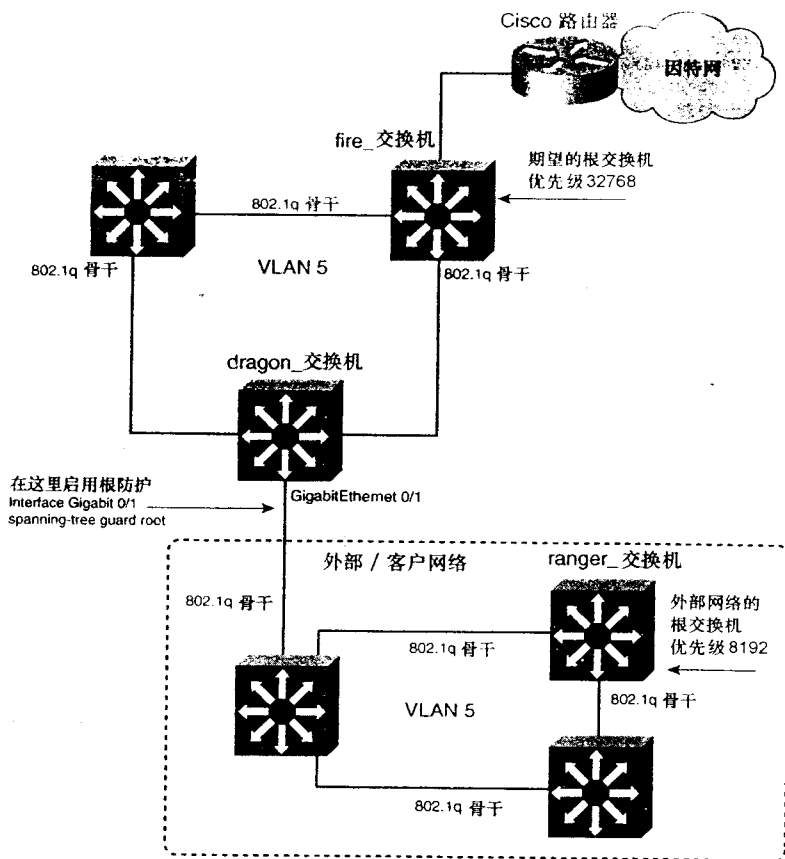


图 1-25 根防护的设置位置

一旦在 dragon 交换机的 GigabitEthernet 0/1 接口上启用了根防护，它就会执行下面的工作。当它监测到某一台交换机具有成为 VLAN 5 根的特性（在这种情况下，是 ranger 交换机），它会将这个端口置为 STP 的 broken 状态，原因是“Root Inconsistent（根的不连续）”。于是

这个端口会被置为阻塞状态，防止 ranger 交换机成为整个 VLAN 的根桥并保护现有的根桥，也就是 fire 交换机。可以用下面的接口命令完成这个功能：

```
dragon_switch(config)interface gigabitethernet 0/1
dragon_switch(config-if)spanning-tree guard root
```

默认情况下，根防护是在所有的端口上启用的。根防护不应当用在上行链路加速或者 loop guard（环路防护）这两个特性上。可以使用 **show spanning-tree detail** 命令来验证根防护的状态，如范例 1-38 所示。下面的范例演示了在 ranger 交换机试图夺取 VLAN 5 的根的情况下，dragon 交换机的 STP 详细情况。

范例 1-38 根防护启用并且激活

```
3550_switch#show spanning-tree detail
<<<text omitted>>>
Port 25 (GigabitEthernet0/1) of VLAN0005 is broken (Root Inconsistent)
  Port path cost 4, Port priority 128, Port Identifier 128.25.
  Designated root has priority 32768, address 0004.275e.f5c4
  Designated bridge has priority 32773, address 000a.8a0e.ba80
  Designated Port id is 128.25, designated path cost 19
  Timers: message age 1, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  BPDU: sent 2077, received 3078
  Root guard is enabled
<<<text omitted>>>
```

五、快速生成树协议（802.1w）和多生成树协议（802.1s）

802.1d 生成树协议在过去一些年中使用得非常好。当产生 802.1d 时，它主要的设计目的是为了桥。在 802.1d 中，BPDU 主要是从桥中继到桥，它的惟一目的就是构造一个无环的只有一个根桥的拓扑结构。在那时交换机不存在并且也没有 VLAN 存在。局域网继续蓬勃发展，交换机出现了，随之有了 VLAN 和 VLAN 骨干的概念。总的来说，STP 依旧发挥着非常好的作用。

生成树协议的收敛时间实在太长了。它需要 50s 的时间从链路的故障中恢复过来，对今天的许多快速以太网和吉比特以太网来说都实在是太长了。

IEEE 一直在非常忙碌地致力于满足不断变化的以太协议的需求。思科系统再次成为领导者，它给 IEEE 委员会提供了在 802.1w RSTP 方面的高级技术，例如端口加速和上行链路加速，等等。IEEE 开发的两个标准在大型的冗余的以太网网络中承担着非常重要的角色，一个是 IEEE 802.1w，也称为快速生成树协议（RSTP），另外一个为 IEEE 802.1s，也称为多生成树（MST）协议。

注意：RSTP 最早在 CAT OS 7.1 中作为 MST 的一部分来实施，并且在 IOS 12.1（11）EX 及以后的版本中使用。它将会作为一个单独的协议，快速的 PVST 模式，在思科 IOS 12.1（13）E 和 CAT OS 7.4 中出现。在写本书时，必须配置 MST，使得 RSTP 工作。

1. 802.1w 快速生成树协议的快速收敛

IEEE 802.1w 也被称为快速生成树协议（RSTP）。RSTP 实际上被称为 *智能的生成树协议*。

RSTP 和 STP 在根的选择、STP 的费用和 STP 的优先级方面的操作是完全一样的。它和 STP

的不同之处是它可以识别一个端口的物理状态，并且可以根据在那个端口上收到的 BPDU 数据包做出关于生成树拓扑的逻辑推测。在 RSTP 中，端口的类型或者端口的角色发挥着非常重要的作用。因为交换机的桥接功能现在是智能的，RSTP 可以在几百毫秒内收敛，而不是 802.1d 的 50s 时间，真是名副其实。RSTP 使用一些技术，例如端口加速以及有关上行链路加速和骨干加速的一些概念。它可以和 PVST+共存，并且完全向后兼容 802.1d。根桥/交换机的选举与 802.1d 是完全一样的。

拓扑变化也是用同样的拓扑变化 (TC) 标记，但是它们和 802.1d 的处理方式完全不一样。802.1w 中的拓扑变化只发生在一个端口从阻塞状态过渡到转发状态时。Edge-port 转换不会产生拓扑变化，而在 802.1d 中，TC 从产生的地方一直流向根交换机/桥，从那里，根再把 TC 传播到生成树的所有叶子上去。从某种方式来说，它有点类似于 OSPF 中的指定路由器 (DR)。在 802.1w 网络中，当变化发生时，TC 会蔓延到所有的端口上，节省了不得不首先到达根交换机的时间。这种方法对 802.1w 网络的快速收敛提供了很大的帮助，防止了不必要的端口转换和 BPDU 的泛洪。

2. 更新的和提高的 BPDU 处理

IEEE 802.1w 桥/交换机确保和传统的 802.1d 桥/交换机向后兼容，它通过使用相同的 802.1d BPDU 的格式和遵循相同的生成树规则来选取根、指定端口和非指定端口。802.1w 使用与 802.1d 相同的 BPDU，但是在如何使用 BPDU 方面两者有很大的不同。802.1w 利用 Flags 字段，使用所有的 8 位来做出智能的转发决定。

图 1-26 显示了传统的 IEEE 802.1d BPDU 的帧格式，与新的 IEEE 802.1w 快速生成树的帧格式进行了比较。802.1d BPDU 只使用了两个标记，其中一个是 TC，另外一个是做 TC 的确认。剩余的 6 位 (第 2~7.位) 没有在 802.1d 中使用。

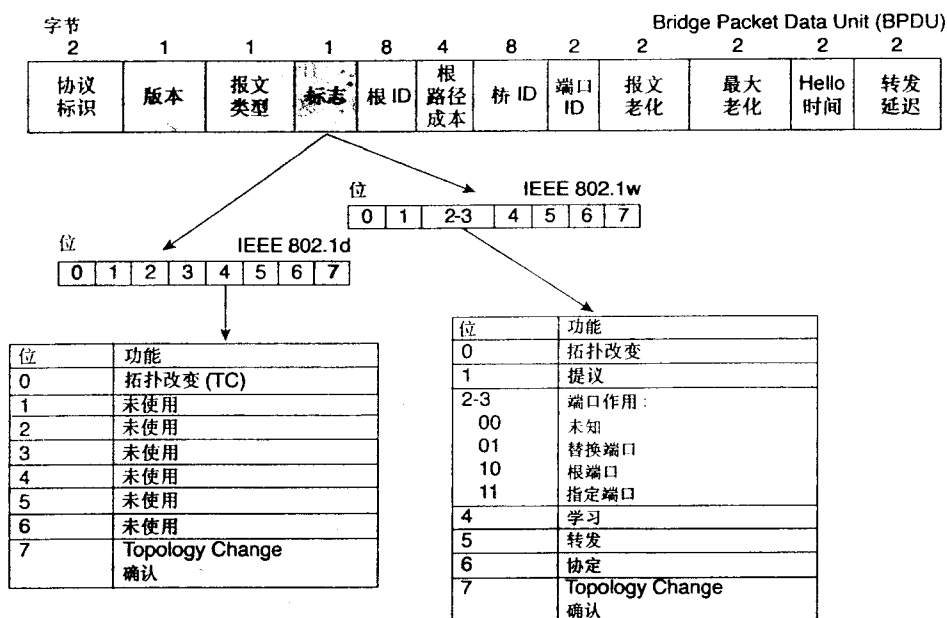


图 1-26 802.1d 和 802.1w 的帧比较

类型为 2，802.1w 可以很容易地识别出任何链路路上的传统交换机，同样，传统的 802.1d 桥不能够识别出版本 2 的 BPDU，将会丢弃它们。当一个 802.1w 的端口检测到端口上收到的 802.1d BPDU 的数据包，它会自动地为 PVST+配置那个端口并且从那个端口发送正常的 802.1d 的帧。802.1w 每隔 2s 发送 BPDU，等同于 802.1d 中的 hello 计时器，在 802.1d 中，一个非根桥只有从它的根端口收到了 BPDU 数据包时，才会产生 BPDU。一个 802.1w 的桥实际产生 BPDU 数据包，而不是像 802.1d 环境中那样中继它们。一个桥现在可以每隔 hello 计时器（默认是 2s）的时间发送关于其当前信息的 BPDU 数据包，即使它没有从根桥收到任何信息。

如果连续三次没有收到 hello 包，BPDU 的信息会立刻过时，这在最大计时器过期的情况下也会发生，BPDU 现在用作桥之间的一种检测存活与否的机制存在，一个桥认为如果连续三次丢失了 BPDU 数据包，那么它就失去了和直连根桥或者指定桥的连接。这被称为快速老化，并且允许快速失败检测。

802.1w 桥也接受内部的 BPDU 数据包，非常类似于骨干加速端口。802.1w 的桥将接受这个内部的 BPDU 信息，并会用它替换老的信息。

按照图 1-26 所示，其他的位现在也用于 802.1w 的帧。其中某些非常重要的位是 proposal（提议）位和端口类型。

- *proposal* 位只是 RSTP 使用的一种实现快速收敛的方法。proposal 机制和计时器不绑定在一起，一个 proposal 信息发送出去是为了同步交换机。当交换机检测到根桥的变化时发送 proposal，要么交换机成为根桥，要么由于收到一个更好的 BPDU 而选举一个新的根端口。当这种情况发生时，交换机会从指定的点对点的端口上给邻接的交换机发送 proposal，当下游的交换机收到 proposal 后，它会给发送这个 proposal 的交换机回送一个确认。当完成这一切后，它会将收到 proposal 的端口置为转发状态。同时所有的指定端口变成阻塞/丢弃状态，这会防止网络产生环路。指定端口接着产生 proposal 信息给下游的交换机。当 proposal 被确认后，指定端口会置为转发状态。这个同步进程会持续进行直到边界交换机，在那儿终止。如果这个端口先前的状态是阻塞状态或者它被定义为一个边界端口，这个同步过程就不会发生。在图 1-27 和图 1-28 中，一个 802.1w 网络通过所描述的同步过程。

3. RSTP/802.1w 的端口状态

802.1w 显著提高收敛时间的另外一种方法是给网络中的每一个端口分配一个特定的角色。如图 1-26 所示，你会看到 802.1w 在 BPDU 的 Flags（标志）字段中给端口的状态留出了空间。802.1w 不仅区分端口类型，也区分链路类型。图 1-29 显示了在 802.1w 网络中的端口状态和角色。

- 链路类型（点对点对共享）——802.1w 或者 RSTP 会认为在全双工状态下工作的链路是点对点链路。在点对点链路上使用 proposal/agreement 机制的收敛前面已经描述过了。如果链路工作在半双工状态下，RSTP 就认为它是一个共享链路。可以通过使用 **spanning-tree link-type** 命令来强制端口的设置。
- **Edge ports**——RSTP 使用相同的命令 **spanning-tree portfast** 来定义边界端口。这可以使 STP 平滑地从 802.1d 过渡到 802.1w。所有边界端口的操作方式和 802.1d 中是一样的，它们忽略了侦听和学习的状态并且可以立刻进入永久的转发状态。在 RSTP 的网络中，如果一个 BPDU 在边界端口上收到，它就会成为一个正常的 STP 端口，

失去它的边界和端口加速的状态。

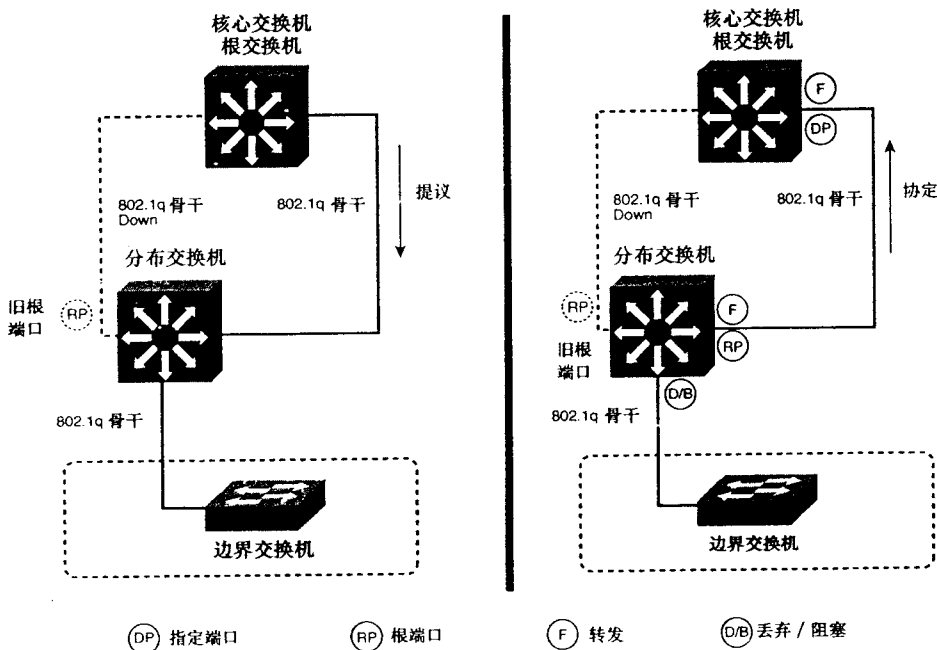


图 1-27 IEEE 802.1w 同步

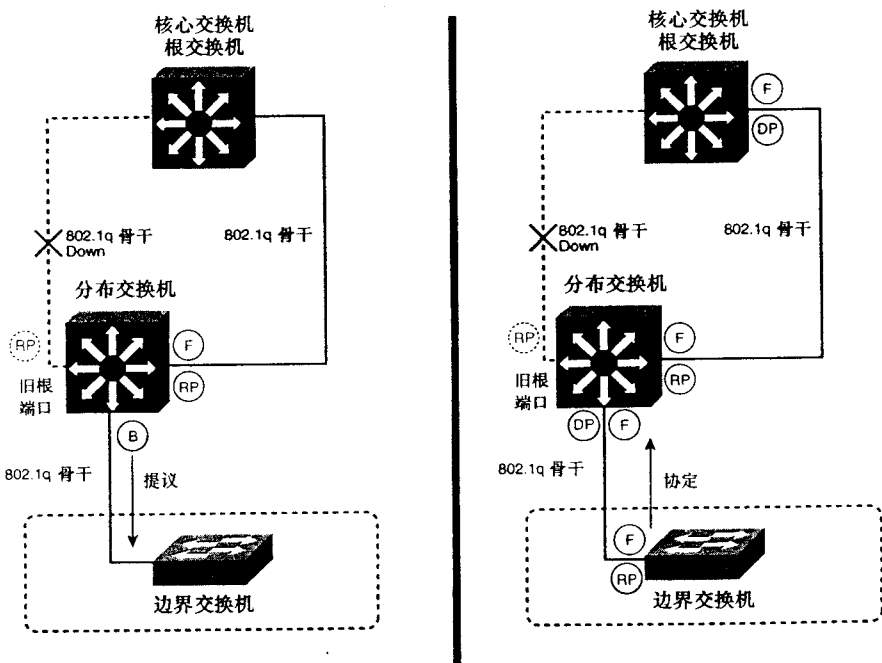
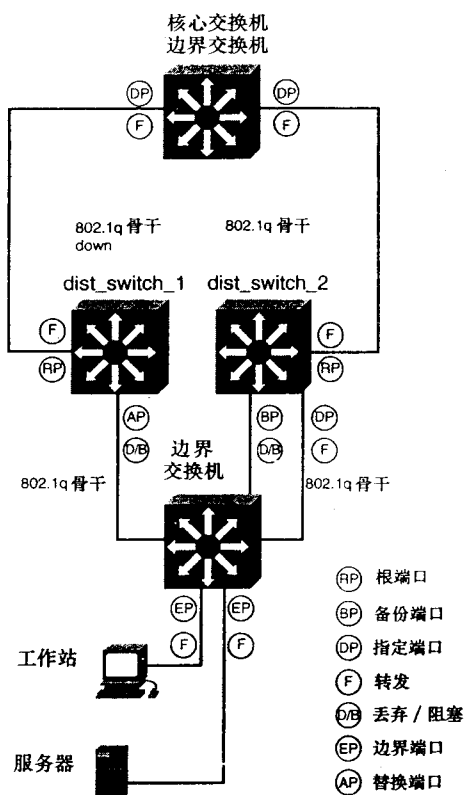


图 1-28 IEEE 802.1w 同步

- **Root ports**——根端口的操作和选举和在 802.1d STP 中是完全相同的。根端口提供一条到达根桥最好的、费用最低的路径。将根端口想象成是指向根桥的端口。如果 RSTP 选举了一个新的根端口，它会阻塞已有的旧的根端口并且立刻将新的根端口转换成转发状态。
- **Designated ports**——指定端口可以被定义成背离根桥的端口，或者是一个局域网必须通过它到达根桥的端口。在每一网段上只有一个指定端口，并且和在 802.1d 中的选举方式是完全一样的，是那个网段上发送最好的 BPDU 的那个桥。指定端口也可以使用 proposal/agreement 过程来加速 RSTP 的快速收敛并且置为转发状态。
- **Alternate ports**——替换端口是一种新的 RSTP 分类。替换端口是在同一网段上可以从另外的桥/交换机上收到更有用的 BPDU 数据包的端口。这些更有用的 BPDU 通常是来自于指定端口。替换端口被置为 RSTP 的一种新状态，叫作 discarding(丢弃)，我们将在下一节中进行讨论。discarding 基本上和阻塞状态是一样的。
- **Backup ports**——备份端口是从它们所在的同一个交换机/桥收到更有用的 BPDU 的端口。备份端口实际上是上行链路加速端口并且功能也是完全一样的。它也可以认为是同一交换机上指定端口的备份。备份端口会进入 discarding 状态。通过使用显示的替换端口和备份端口，RSTP 能够在它丢失 BPDU 或者根端口的情况下做出智能的收敛决定。这也是 RSTP 提供的另外一种快速收敛的方法。

图 1-29 演示了在一个常见的网络上新的 RSTP 端口状态。



802.1w RSTP 也使用一种和 802.1d 稍微不同的端口状态。替代了阻塞状态，RSTP 协议使用的是丢弃状态。表 1-5 比较了旧的 802.1d STP 状态和新的 802.1Q RSTP 状态。

现在，在 Catalyst 3550 交换机上配置 802.1w RSTP，需要你配置 802.1s MST。在 Catalyst 4000、6500 和其他的 CAT OS 系统上，使用 **set spantree mode** 命令，RSTP 可以单独启用而不需要 MST。

表 1-15 STP 和 RSTP 端口状态比较

802.1d STP 状态	802.1w RSTP 状态	端口包括在活动的拓扑中?
阻塞	丢弃	否
侦听	丢弃	否
学习	学习	是
转发	转发	是

六、多生成树协议 (802.1s)

多生成树 802.1s 允许用户将 VLAN 组织在一起，并把相关的 STP 树组成通用的组或者实例。同一 STP 实例的成员具有相同的 STP 拓扑，例如根和哪些端口处于转发状态等诸如此类。一个 STP 实例中的 VLAN 成员和另外一个 STP 实例中的 VLAN 成员的操作是独立的，MST 允许网络管理员快速地在网络中配置负载分担，而无需对交换机上的每一个 VLAN 设置单独的根或优先级。MST 完成这个任务，一部分是通过使用 MST 的区域来实现。

MST 的区域是互连的桥，它们具有相同的 MST 配置，配置包含下面的信息：

- MST 的实例号码和名字；
- 配置修订号；
- 4096 个元素表，用于 VLAN 关联。

实例号码、名字和配置修订号必须与相同的 MST 区域中的交换机相匹配。

本章先前已经介绍了 VLAN 的负载分担（参考图 1-30）。使用传统的 802.1d STP，必须在 yang 交换机上定义 VLAN 2 和 3 的根桥。也不得不在 yin 交换机上手动分配 VLAN 4 和 5 的根桥。这个过程在 yin 和 yang 交换机上通过链路实现负载分担是非常必要的。在大型网络中，这会导致大量的工作（需要手动设置每一个 VLAN 的根和优先级）。

如果你正在这个网络上运行 MST 802.1s，那么只需要建立两个 MST 的实例。一个实例将 VLAN 2 和 3 分配给它，而根桥是 yang 交换机。第二个 MST 的实例将 VLAN 4 和 5 分配给它，而根桥将是 yin 交换机。如果你需要给网络中添加更多的 VLAN，新的 VLAN 只需要成为这两个实例中任何一个的成员。使用 MST，仅需要对 STP 配置两个实例，而不是对每一个 VLAN 配置 STP 和它的相关参数。图 1-31 演示了用 802.1s 配置的网络。

思科实施的 MST 定义了下面的特性：

- MST 运行了生成树的一个变种，叫作内部生成树（IST）。IST 补充了公共生成树（CST）的信息，使用了关于 MST 区域的内部信息。MST 区域对邻接的 802.1d 或者单生成树（SST）以及其他的 MST 区域看起来就像一个单独的桥。参看图 1-32 和 1-33。

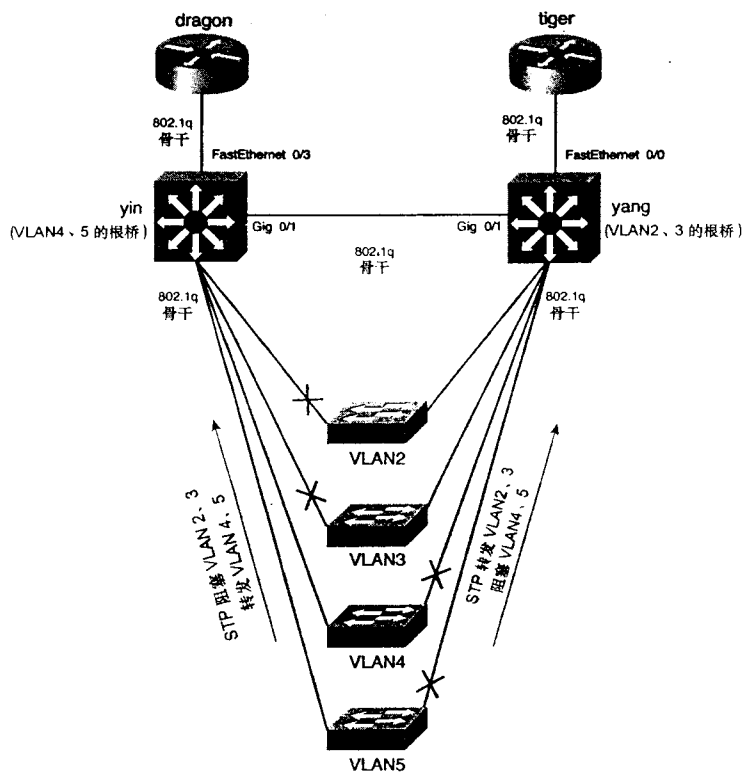


图 1-30 STP 使用 802.1d 的负载分担

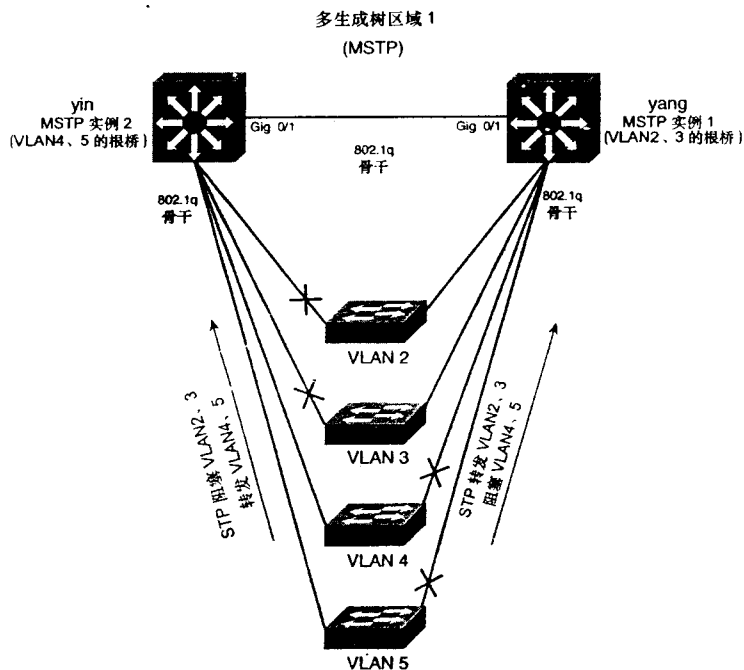


图 1-31 STP 使用 802.1s 的负载分担

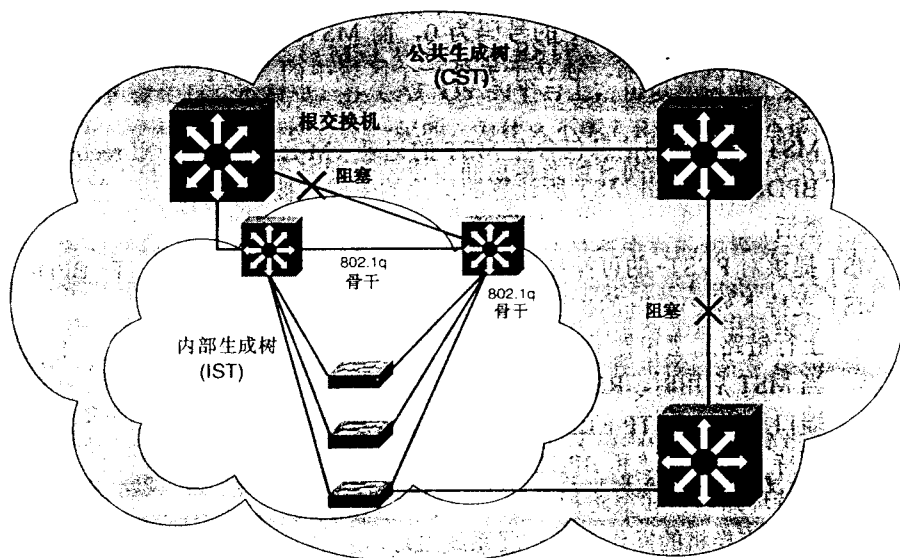


图 1-32 802.1s 中 CST IST 的关系

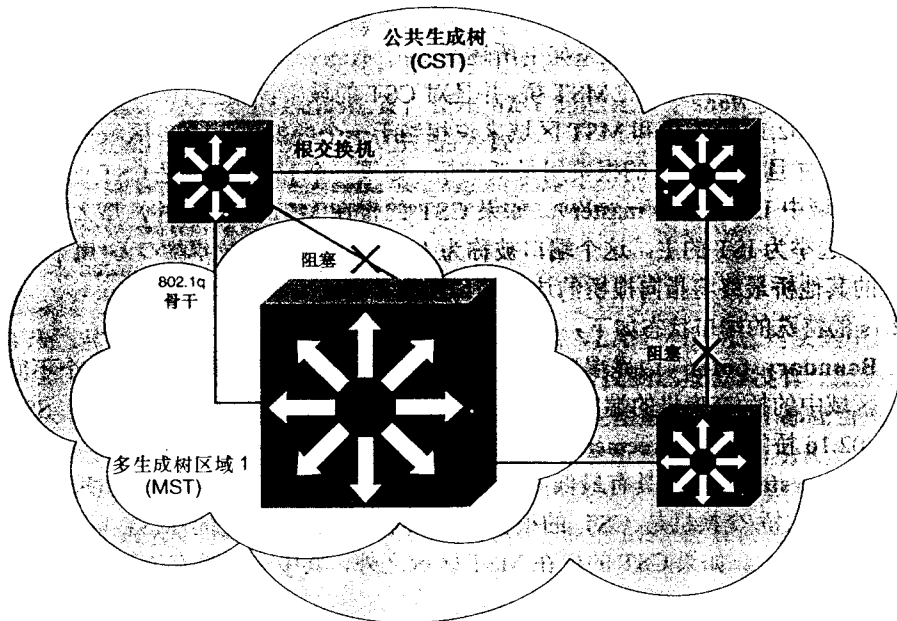


图 1-33 802.1s 中 CST MST 的关系

- 公共内部生成树 (CIST) 是下列方式的组合：在每一个 MST 区域中有 IST，CST 将 MST 的区域和传统的 802.1d 或者 SST 的桥进行互连。CIST 在一个 MST 的区域内部等同于 IST，在一个 MST 的区域外部等同于 CST。STP、RSTP 和 MST 联合选举一个交换机作为 CIST 的根桥。CIST 可以被看作是类似于 802.1Q 中的单生成树协议。

- MST 在每一个 MST 的区域内建立并维护额外的生成树。这些生成树被定义为 *MST 实例* (MSTI)。IST 的号码为 0，而 MSTI 的号码为 1、2、3 等等。MSTI 在一个 MST 区域内部，独立于另外一个区域中的 MSTI，即使这两个 MST 的区域互连。
- 关于 MSTI 生成树的信息包含在 MST 的记录 (M-record) 里。M-record 总是封装在 MST BPDU 里面。由 MST 所计算的原始的生成树被称为 M-tree，它只能在 MST 区域中激活。
- MST 提供对 PVST+ 的可操作性，通过对非 CST VLAN 产生 PVST+ BPDU 来实现。
- MST 支持下列 PVST+ 的扩展：
 - 上行链路加速和骨干加速在 MST 的模式下不可配置；它们是 RSTP 的一部分，当 MST 启用时，RSTP 自动启用。
 - 端口加速对 RSTP 的边界端口必须支持和启用。
 - 支持 BPDU 过滤和 BPDU 防护。
 - 支持环路防护和根防护。
 - MST 交换机的操作使用扩展系统 ID。

图 1-32 和 1-33 演示了 MST、IST 和 CST 这些功能的关系。相同拓扑的两张图看起来是不一样的。MST 区域对于 CST 就是一个单独的桥。CST 并不关心在一个 MST 区域中有多少个桥或者 STP 的路径。

思科定义的 IST 和 CST 的关系如下所述：

IST 连接一个区域中所有的 MST 桥，并且对 CST 的域来说相当于一个 STP 的子树。MST 区域对邻接的 802.1d SST 桥和 MST 区域来说相当于一个虚拟桥。MST 区域中的 IST 主是具有最低的 BID 并且到达 CST 的根费用最低的桥。如果一个 MST 的桥是 CST 的根桥，那么它也是 MST 区域中 IST 的主 (master)。如果 CST 的根在 MST 区域之外，那么边界中的一个 MST 桥会被选举为 IST 的主。这个端口被称为 *boundary port* (边界端口)。属于同一个区域的边界上的其他桥最终将指向根桥的边界端口阻塞掉。

802.1s 的特殊端口状态如下：

- **Boundary ports** (边界端口) 是连接到传统的 802.1d 局域网或者一个不同的 MST 区域中的桥/交换机的端口。边界端口可以通过检查另一个 MST 或者 SST 传统的 802.1d 桥发出的 agreement 信息来自动地配置自己。
- **IST master** 是一个具有最低的 BID 并且到达 CST 的根费用最低的桥/交换机。如果 MST 的桥/交换机是 CST 的根桥，那么它也是那个特定的 MST 区域中 IST 的主 (master)。如果 CST 的根在 MST 区域之外，其中一个具有边界端口的 MST 桥就会被选举为 IST 的主 (master)。

注意：IST 的 IST BPDU 数据包是以 MST 实例 0 发送的。只有 MST 的第一个实例实际发送 BPDU。思科交换机的第一个实例是 0，因此应当避免将 VLAN 映射到这个实例。将它和 VLAN 1 的特殊性同等看待。它在所到之处都在运行，对 IST 是必需的。

先前的信息是对 IEEE 802.1w 和 IEEE 802.1s 的一个综合性的概述。和许多协议一样，技术细节可能是相当复杂的。关于 802.1w 和 802.1s 的更详细信息，参看 www.ieee.org、standards.ieee.org，当然还包括 www.cisco.com。

1. 配置 IEEE 802.1w 快速生成树协议 (RSTP) 和 IEEE 802.1s 多生成树协议 (MST)

思科已经无缝地迁移到 802.1s MST 和 802.1w RSTP。事实上，当你选择 MST 作为生成树模式时，RSTP 也会自动地启用。在 CAT OS 的平台上，可以将这两个独立配置，但是在 Catalyst 3550 上，这两个是紧密结合在一起的，为什么不呢？RSTP 的好处是巨大的，而且随着网络规模的扩大这个好处会更加明显。你可能发现就像它的祖先 802.1d STP 一样，配置 802.1w 和 802.1s 是很简单的，而不像后面的概念那么复杂。

为了配置 802.1w RSTP，需要配置 802.1s MST，并且在所有的边界端口上启用生成树的端口加速。RSTP 在配置 MST 时会自动地启用。使用下面的进程在 Catalyst 3550 上配置 RSTP 和 MST。这个配置过程假设你已经将 VLAN、VTP 和 VLAN 骨干启用并运行了。

第 1 步 在所有的边界端口上配置生成树的端口加速。使用接口命令 **spanning-tree portfast**。

第 2 步 配置 MST 的名字和修订号。在一个 MST 区域中的所有交换机都必须具有相同的 MST 名字和修订号。为了配置 MST，首先进入 MST 的配置模式，使用下面的全局配置命令：

```
3550_switch(config)#spanning-tree mst configuration
```

从这个模式可以配置 MST 的实例、名字和修订号，并且显示当前的 MST 配置。这个模式的工作方式很像 VLAN 数据库，只有提交才能使得变化生效。使用关键字 **exit** 来提交变化，或者使用关键字 **abort** 来清除这个期间输入的任何配置。为了显示挂起的配置，使用 **show pending**。使用下面的 MST 配置命令来配置 MST 的参数。

```
3550_switch(config-mst)#name MST_region_name
3550_switch(config-mst)#revision revision_number_<0-65535>
3550_switch(config-mst)#exit
! Must commit changes for MST
3550_switch(config-mst)#abort
! optional Aborts MST config
```

第 3 步 将 MST 的区域划分成 MST 的实例并且将 VLAN 分配到这些实例中。记住，在一个实例中的所有 VLAN 都会遵循相同的路径到达根桥。没有分配到任何一个实例中的 VLAN 会默认地使用实例 0。所有在使用的 VLAN 都应当被分配到一个实例中。如果你只是想启用 RSTP，那么将所有的 VLAN 分配到实例 1 中。如果你想实现负载分担，那么将一半的 VLAN 分配到一个实例中，将另外一半的 VLAN 分配到另外一个实例中。使用下面的 MST 配置命令来分配 MST 的实例和关联的 VLAN。

```
3550_switch(config-mst)#instance <0-15> vlan vlan,vlan-range
```

第 4 步 启用 MST 模式。使用下面的全局配置命令在 PVST 的默认模式下启用 MST 模式。这个命令也可以启用 RSTP 802.1w。

```
3550_switch(config)#spanning-tree mode mst
```

注意：MST 实例 0 是用于 IST 的。作为一个设计规则，将 VLAN 1 和其他不用的 VLAN 分配给 MST 实例 0。这是一个设计上的选项，而不是一个功能上的需求。


```

yin_switch(config-mst)#instance 2 vlan 101-1005
! VLANs 2-100 assigned to Instance 2
yin_switch(config-mst)#show current
! view current MST changes
Current MST configuration
Name      [cisco]
Revision  1
Instance  Vlans mapped
-----
0          1,1006-4094
1          2-100
2          101-1005
yin_switch(config-mst)#exit
! commit current MST changes
yin_switch(config)#spanning-tree mode mst
! enable MST mode
yin_switch(config)#spanning-tree mst 2 root primary
! set MST instance 2 to root
% This switch is already the root bridge of the MST02 spanning tree
mst 2 bridge priority set to 24576

```

可以使用 **show spanning-tree mst 0-15 [configuration | detail | interface]** 命令来查看和验证 MST 的状态。这个命令显示了关于 MST 实例的详细信息，例如根、根的优先级、MST 的接口和接口的角色；还列出了状态和类型。范例 1-40 演示了在 yin 交换机上使用 **show spanning-tree mst** 命令。

范例 1-40 show spanning-tree mst 命令

```

yin_switch#show spanning-tree mst 2
##### MST02          vlans mapped: 101-1005
Bridge      address 000a.8a0e.ba80  priority 24578 (24576 sysid 2)
Root        this switch for MST02
Interface   role state cost      prio type
-----
Fa0/3       desg FWD  200000  128  edge P2P
Fa0/17      desg FWD  200000  128  P2P
Fa0/20      boun BLK  200000  128  P2P bound(PVST)

```

注意端口 Fast 0/17 对相同区域中的交换机来说是一个指定的点对点端口，而端口 Fast 0/20 对一个 PVST (802.1d) 的域来说是一个边界的点对点链路。接口 Fast 0/3 和路由器相连，是一个边界端口，因为它在全双工模式下，所以它也是一个点对点的链路。

为了演示 MST 和 RSTP 的收敛速度有多快，范例 1-41 从 yin 交换机发出了一个扩展 ping 到 tiger 交换机。注意在范例 1-40 中，连接到 yang 交换机的 Fast 0/17 端口正在转发，在 ping 中，端口 0/17 将会被断开，而你会看到，在 ping 中没有任何丢失。这真是比 802.1d 有不可估量的收敛速度的提高。回想一下，一个 802.1d 的网络将会经过至少 50s 的收敛时间。

范例 1-41 快速生成树在生效

```

yin_switch#ping
Protocol [ip]: ip
Target IP address: 172.16.192.13
Repeat count [5]: 5000

```

(待续)

```
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5000, 100-byte ICMP Echos to 172.16.192.13, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<<<text omitted>>>
!!
00:53:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/17, change to
down
.!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<<<text omitted>>>
Success rate is 99 percent (4999/5000), round-trip min/avg/max = 1/14/72 ms
yin_switch#
yin_switch#show spanning-tree mst 2
##### MST02          vlans mapped:    101-1005
Bridge      address 000a.8a0e.ba80  priority 24578 (24576 sysid 2)
Root        this switch for MST02
Interface   role state cost        prio type
-----
Fa0/3       desg FWD   200000    128   edge P2P
Fa0/20      boun FWD   200000    128   P2P bound(PVST)  <-Fast 0/17 is gone!!
```

同一 802.1d 生成树命令的变种在 MST 中可用来设置 STP 的主根桥、次根桥、端口优先级、端口费用值和 STP 的优先级。基本上，它们和 802.1d 的命令是一样的。修改这些可变值的语法如下：

```
3550_switch(config)#spanning-tree mst instance_id [root { primary | secondary } | cost 1-200000000 | priority 0-61440 | port-priority 0-255]
```

为了调整 MST 的计时器，使用下面的语法：

```
3550_switch(config)#spanning-tree mst instance_id [hello-time 1-10 | max-age 6-40 | forward-time 6-40 | max-hops 1-40]
```

为了将 MST 的链路类型修改为点对点，使用下面的接口命令：

```
3550_switch(config-if)#spanning-tree link-type point-to-point
```

可以使用下面的命令来验证 MST 的配置：

```
show spanning-tree mst instance_id [configuration | detail | interface]
```

show spanning-tree mst detail 命令显示了所有的 MST 实例以及相关的 STP 端口、STP 状态和计时器。范例 1-42 列出了在 yin 交换机上使用 **show spanning-tree mst detail** 命令的部分输出。关于不同的 **show** 命令的详细信息，参考思科 IOS 文档。

范例 1-42 show mst detail 命令的输出

```
yin_switch#show spanning-tree mst detail
##### MST00          vlans mapped:    1,1006-4094
Bridge      address 000a.8a0e.ba80  priority 32768 (32768 sysid 0)
```

(待续)

Root	address 0004.275e.f0c0	priority 32768 (32768 sysid 0)			
	port Fa0/17	path cost 20019			
IST master	address 0030.1976.4d00	priority 32768 (32768 sysid 0)			
		path cost 200000	rem hops 19		
Operational hello time 2, forward delay 15, max age 20, max hops 20					
Configured hello time 2, forward delay 15, max age 20, max hops 20					
FastEthernet0/3 of MST00 is designated forwarding					
Port info	port id	128.3	priority 128	cost	200000
Designated root	address 0004.275e.f0c0	priority 32768	cost	20019	
Designated ist master	address 0030.1976.4d00	priority 32768	cost	200000	
Designated	address 000a.8a0a.ba80	priority 32768	port id	128.3	
Timers: mcast	forward delay 0, transition to forwarding 0				
Bpdus sent 5214					
FastEthernet0/17 is root forwarding					
Port info	port id	128.13	priority 128	cost	200000
Designated root	address 0004.275e.f0c0	priority 32768	cost	20019	
Designated ist master	address 0030.1976.4d00	priority 32768	cost	0	
Designated bridge	address 0030.1976.4d00	priority	cost	id	32.81
<<<<					

七、使用 VLAN 映射控制流量和安全

Catalyst 3550 允许用户使用一种特殊的路由映射类型过滤，叫作 *VLAN 映射*，来控制一个 VLAN 中所有的流量。这一部分内容简单地讨论如何配置和应用 VLAN 映射。

VLAN 映射允许用户控制在交换机本地上的 VLAN 中的所有流量。VLAN 映射适用于所有从这个 VLAN 路由出去的流量或进入的流量，或者是在交换机本地上的 VLAN 中桥接的流量。VLAN 映射并没有一个方向 (in 或 out) 和它关联。

可以配置 VLAN 映射来和一个标准的、扩展的或者命名的访问控制列表一起工作。Catalyst 3550 交换机也支持 IP 标准的和 IP 扩展的访问控制列表，号码为 1~199 和 1300~2699。所有的非 IP 协议通过 MAC 地址进行控制，因此使用基于 MAC 的 VLAN 映射。应当注意不能基于 MAC 地址来过滤 IP 流量，这一点非常重要。MAC 过滤只适用于非路由的流量，例如 NetBIOS。必须配置一个 IP 标准的或者扩展的访问控制列表来转发 IP 流量。

VLAN 映射和路由映射的工作方式非常相像。如果你对路由映射不太熟悉的话，可以跳到第 2 章了解关于路由映射的详细信息。

为了利用路由映射来控制 IP 流量，首先配置 VLAN 映射，接着给这个 VLAN 映射分配一个序列号。VLAN 映射是从最低的序列号到最高的序列号执行的，使用全局配置命令 **vlan access-map map_name sequence_number** 命令。接下来，增加 **match ip** 语句，在这里你可以调用一个命名的访问控制列表来作为匹配的条件。之后给 VLAN 映射指定一个行为措施，有效的行为是 **action forward** 和 **action drop**。基于访问控制列表的结果，交换机转发或者丢弃流量。MAC 过滤也可以用来过滤非路由的流量。要应用 VLAN 映射，使用 **vlan filter map_name vlan-list vlans** 命令。

在图 1-35 中，有 3 个 IP 主机连接到交换机上，在本范例中，IP 流量需要在 VLAN 100 中进行控制，使得只有 172.16.128.7 和 172.16.128.3 主机之间能互相对话。IP 主机 172.16.128.8 不能够 ping 通 172.16.128.7 或者 172.16.128.3。

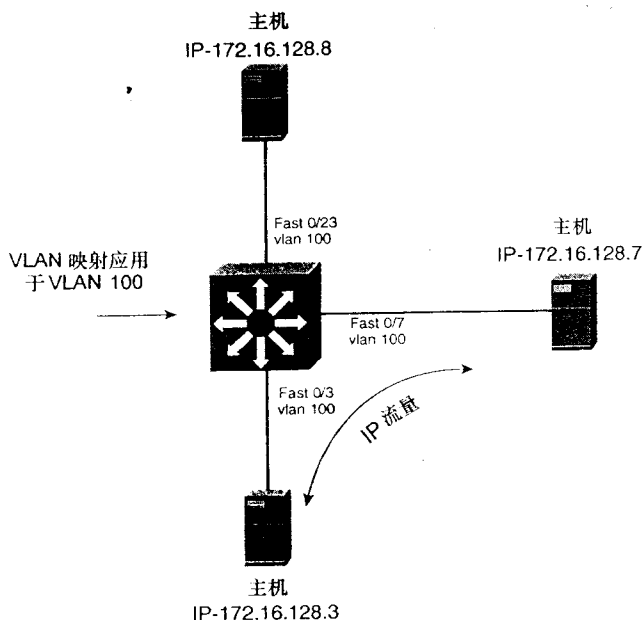


图 1-35 VLAN 映射

范例 1-43 演示了使用 VLAN 映射来控制 IP 访问的配置。

范例 1-43 配置 VLAN 映射

```
3550_switch(config)#vlan access-map allow_ip 10
! Define the VLAN map 'allowip'
3550_s(config-access-map)#action forward
! Forward ACL permitip
3550_s(config-access-map)#match ip address permitip
! Call ACL permitip
3550_s(config-access-map)#exit
3550_switch(config)#
3550_switch(config)#ip access-list extended permitip
! ACL permitip
3550_swi(config-ext-nacl)#permit ip host 172.16.100.7 host 172.16.100.3
3550_swi(config-ext-nacl)#permit ip host 172.16.100.3 host 172.16.100.7
3550_swi(config-ext-nacl)#exit
3550_switch(config)#
3550_switch(config)#vlan filter allow_ip vlan-list 100
! Apply VLAN map to VLAN 100
3550_switch(config)#
```

为了验证 VLAN 映射，使用 **show vlan access-map** 和 **show access-list** 命令来验证你的配置。

MAC 过滤可以使用 VLAN 映射来控制非路由的流量，例如 NetBIOS 或者系统网络体系结构 (SNA)。范例 1-44 列出了用来防止不安全的主机通过不可路由的协议和其他主机通信的配置。注意这只控制不可路由的流量，而对 IP 流量没有任何作用。此范例允许两个 MAC 地址 00e0.1e58.e792 和 00e0.1e58.c112 和其余的网络之间的非路由流量，但是这两个主机之间不能进行通信。

范例 1-44 MAC 地址的 VLAN 映射

```
vlan access-map allowed_macs 10
! define VLAN map 'allowed_macs'
action forward
! forward ACL valid_macs
match mac address valid_macs
! call mac ACL 'valid_macs'
!
vlan filter allowed_macs vlan-list 100
! Apply VLAN map to VLAN 100
!
mac access-list extended valid_macs
! MAC ACL 'valid_macs'
permit host 00e0.1e58.e792 any
! Allow these two MAC addresses
permit host 00e0.1e58.c112 any
```

注意: 适用于访问控制列表和路由映射的规则同样适用于 VLAN 映射。例如在访问控制列表的末尾有一个隐含的 **deny any** 规则同样也适用于 VLAN 映射。关于如何配置路由映射和访问控制列表的更多信息和配置提示, 参考《CCIE 实验指南》第 1 卷和第 2 卷中的适当章节。

八、使用被保护端口控制 VLAN 访问和安全

然而, 在 Catalyst 3550 上可以控制访问或者增强安全的另外一种方法是使用 VLAN 的被保护端口。VLAN 的被保护端口只能和非保护端口通信。来自一个 VLAN 被保护端口的流量不能到达另外一个被保护的端口。在图 1-36 中, Fast Ethernet 0/8 和 0/7 是 VLAN 的被保护端口, IP 主机 172.16.128.7 不能 ping 通 172.16.128.8, 但是它可以 ping 通 172.16.128.3。主机 172.16.128.3 可以 ping 通 172.16.128.8 和 172.16.128.7 这两台主机。

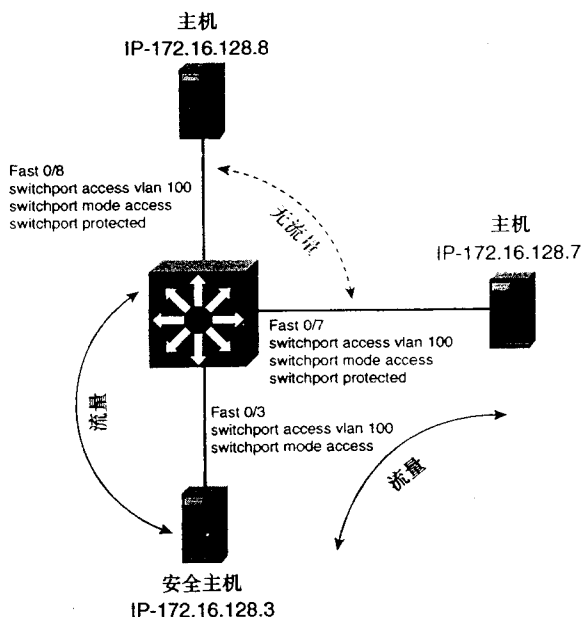


图 1-36 VLAN 被保护端口

为了将端口配置成被保护端口，使用接口命令 **switchport protected**，可以通过 **show interface fast 0/7 switchport** 命令来验证被保护端口，参看范例 1-45。

范例 1-45 验证被保护端口

```
3550_switch#show interfaces fast 0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 100 (psv2_vlan100)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: true
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled

Voice VLAN: none (Inactive)
Appliance trust: none
```

Catalyst 3550 交换机默认情况下会将所有未知目的 MAC 地址的数据包泛洪到所有的端口。如果未知的单播和组播流量转发到被保护的端口，就可能会有安全性的问题。为了防止未知的单播或者组播流量从一个端口转发到另外一个端口，可以配置一个端口（被保护的或未被保护的）来阻塞未知的单播或者组播数据包。使用下面的接口命令来阻塞未知的单播和组播流量：

```
3550_switch(config-if)#switchport block unicast
3550_switch(config-if)#switchport block multicast
```

如果启用了单播或者组播的阻塞，在 **show** 交换机端口命令中可以显示出来这个功能已经启用了，正如前面的范例中所示。

1.5 实验 1：配置以太通道、三层交换、路由 端口和 SVI

1.5.1 练习场景

以太网交换机的世界以极快的速度不断发展。在这个领域中，你会遇到许多种类型的交换机，而 Catalyst 3550 可能是其中的一款。Catalyst 3550 有许多种可配置的接口。能够配置这些不同类型的接口是非常重要的，它使得你在设计方面更加灵活。例如快速/吉比特以太通道等功能为核心交换机提供非常高的带宽和极高的冗余性。

1.5.2 实验练习

拿大和西北的广大野外地区使用。FrozenTundra.com 正在将它的骨干提升到吉比特以太网并且会使用两个吉比特以太网接口转换器 (GBIC)，它随 Catalyst 3550-24 以太网交换机自带。它还想在 3550 交换机而不是在路由器上执行三层交换。

你的任务是配置一个工作的 IP 网络，使用下面严格的设计指导配置以太 3550 交换机：

- 按照图 1-37 的描述配置 FrozenTundra.com 的 IP 网络。在所有的路由器上使用 EIGRP 作为路由选择协议，2003 为自治系统 ID。
- 按照图 1-37 配置所有的 IP 地址。所有标注的接口应当能够彼此 ping 通。
- 参看“实验目的”部分了解配置细节。

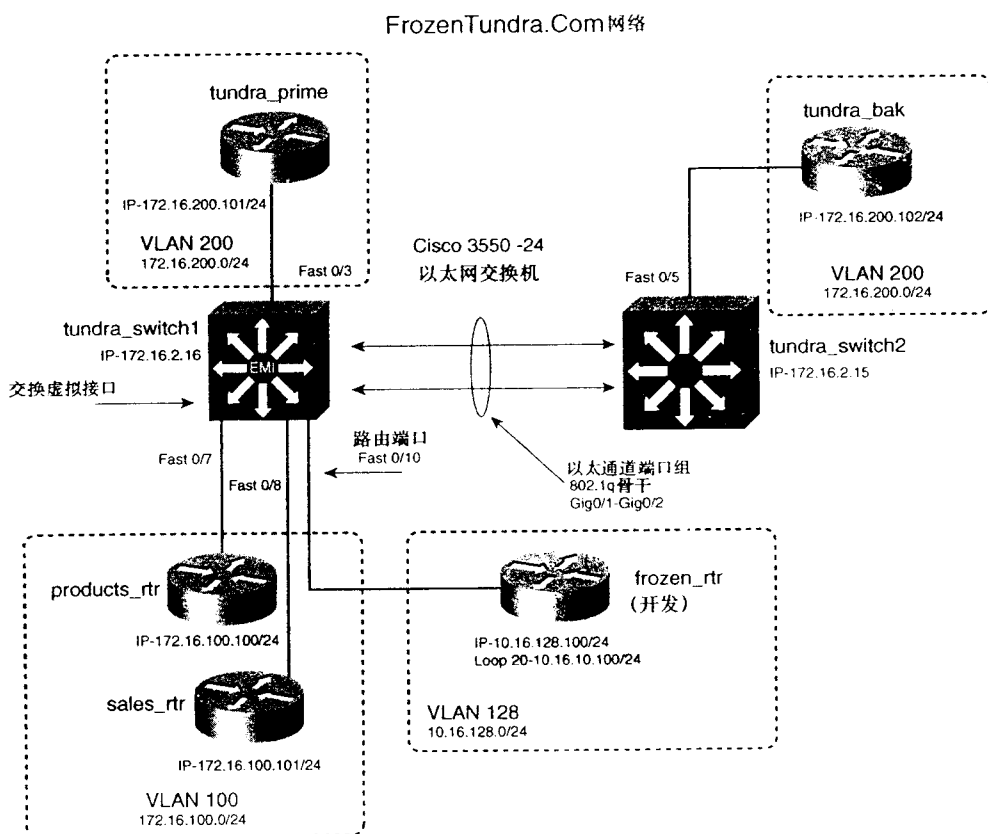


图 1-37 Tundra.net

1.5.3 实验目的

- 将 EIGRP 配置为路由选择协议，按照图 1-37 所描述的，使用 2003 作为自治系统的 ID。
- 在 tundra_switch1 上配置管理接口 172.16.2.16/24，在 tundra_switch2 上配置 172.16.2.15/24。这些地址应该互相可达，而且远程登录在两个交换机上应当支持 4

- 将两个吉比特以太网接口配置为一个吉比特以太网通道。如果你没有吉比特以太网，可以使用快速以太网。
- 将 tundra_switch1 配置为 VTP 的服务器，将 tundra_switch2 配置为 VTP 的客户端。使用 VTP 的域名 tundra 和 VTP 的口令 psv2。
- 将 tundra_switch1 的接口 Fast 0/10，一个连接到 frozen 路由器的接口，配置为路由端口，在这个接口上使用 IP 地址 10.16.128.16。
- 将其他的接口配置为接入端口，并按照图 1-37 分配到相应的 VLAN。
- 为 tundra_switch1 配置任何需要的 SVI，提供这个网络中任何 VLAN 之间的路由。
- 在 tundra_prime 路由器、tundra_switch1 和 tundra_bak 路由器上对 VLAN 200 配置 HSRP，主 IP 地址应当为 172.16.200.1/24，而 tundra_switch1 应当为 HSRP 的主。tundra_prime 路由器应当为 HSRP 的备份。
- 将 tundra_switch1 配置为 VLAN 100 和 200 的 STP 根。
- 在两台交换机的以太网通道链路上启用 VTP 剪枝。

1.5.4 需要的设备

- 5 台思科路由器，一个安装了 EMI 软件的 Catalyst 3550 交换机和另外一个 Catalyst 35xx 交换机。只需要一个 Catalyst 3550 交换机安装 EMI 软件。只要交换机能够支持 802.1Q 和以太网通道，就可以用一个交换机来仿真另外一个交换机。
- 交换机需要两个背对背的 100BASE-T 链路或者一个吉比特以太网用于以太网通道连接。按照图 1-37，其他路由器应当使用 5 类线来和适当的交换机相连。

1.5.5 物理布局和预规划

- 按照图 1-37，将交换机和路由器相连。
- 本实验关注于配置以太网交换机。

1.5.6 实验步骤

如图 1-37 所示，将所有的路由器连接到交换机上。可以在两个交换机之间使用两个吉比特以太网连接或者 100 Mbit/s 以太网连接。你的选择不会影响整个实验的运行结果。

我们回忆一下配置 3550 以太网交换机的七步过程。

第 1 步 配置交换机管理。

第 2 步 配置 VTP 和 VLAN 并且给 VLAN 分配端口/接口。

第 3 步 使用 EtherChannel、802.1Q 和 ISL 封装在交换机间配置连接。

第 4 步 （可选）控制 STP 和 VLAN 传播。

第 5 步 （可选）配置 SVI。

第 6 步 （可选）配置路由端口。

第 7 步 （可选）配置三层交换。

第 1 步是配置交换机管理。这包括设置主机名、口令和交换机上的管理地址。在本实验中，也可以在 vty 线路 0~4 上建立远程登录。范例 1-46 列出了 tundra_switch1 的管理部分。

范例 1-46 到目前为止 tundra_switch1 交换机的管理部分

```
hostname tundra_switch1
!
enable secret 5 $1$nt35$131XBSgKT6BmA1KHMqj1V1
! Enable Secret=cisco
!
<<<text omitted>>>
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
! MNGT VLAN and IP
 ip address 172.16.2.16 255.255.255.0
<<text omitted>>>
!
line con 0
line vty 0 4
 password cisco
! Telnet access allowed
 login
line vty 5 7
 login
```

第 2 步要求用户配置 VTP 和 VLAN。需要对 SVI、接入端口和管理 VLAN 配置相应的 VLAN。在这个模型中，需要配置 4 个 VLAN：VLAN 2、100、128 和 200。在 3550 交换机上，可以在全局配置模式下使用 **vlan x** 命令建立 VLAN。在输入 VLAN 的号码后，也可以给 VLAN 输入名字。tundra_switch1 交换机的 VTP 模式是 server 模式，而 tundra_switch2 交换机是 client 模式。VTP 的域名是 tundra，口令是 psv2。确保 VTP 的域名有相同的大小写，口令也是一样。域名和口令是大小写敏感的。确保 VTP 服务器的修订版本号大于 VTP 客户交换机的版本号，否则，这两者之间不会同步。VTP 的域名和模式可以从 VLAN 数据库或者 VLAN 配置模式下进行配置。范例 1-47 演示了 tundra_switch1 上的配置。

范例 1-47 在 tundra_switch1 交换机上配置 VTP

```
tundra_switch1#vlan database
tundra_switch1(vlan)#vtp domain tundra
tundra_switch1(vlan)#vtp server
tundra_switch1(vlan)#vtp password psv2
```

这一步也要求用户配置物理端口的属性并将这些端口分配到 VLAN 中去。范例 1-48 演示了到目前为止 tundra_switch1 交换机上的 VLAN 和端口配置。

可以使用 **show vlan** 命令和 **show vtp status** 命令来验证 VLAN 和 VTP，如范例 1-49 所示。

范例 1-48 配置 VLAN 的端口成员

```
hostname tundra_switch1
!
<<<text omitted>>>
!
interface FastEthernet0/3
 switchport access vlan 200
! assigned to VLAN 200
 switchport mode access
 no ip address
!
interface FastEthernet0/4
 no ip address
!
interface FastEthernet0/5
 no ip address
!
interface FastEthernet0/6
 no ip address
!
interface FastEthernet0/7
 switchport access vlan 100
! assigned to VLAN 100
 switchport mode access
 no ip address
!
interface FastEthernet0/8
 switchport access vlan 100
! assigned to VLAN 100
 switchport mode access
 no ip address
!
```

范例 1-49 验证 VTP 和 VLAN 的状态

```
tundra_switch1#show vlan
VLAN Name                Status    Ports
-----
1    default                active    Fa0/1, Fa0/2, Fa0/4, Fa0/5
                                           Fa0/6, Fa0/9, Fa0/11, Fa0/12
                                           Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                           Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                           Fa0/21, Fa0/22, Fa0/23, Fa0/24
2    psv2_vlan2             active
100  psv2_vlan100            active    Fa0/7, Fa0/8
200  psv2_vlan200            active    Fa0/3
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
100  enet    100100    1500    -      -      -      -    -      0      0
128  enet    100128    1500    -      -      -      -    -      0      0
```

(待续)

```

200 enet 100200 1500 - - - - - 0 0
1002 fddi 101002 1500 - - - - - 0 0
1003 tr 101003 1500 - - - - srb 0 0
1004 fdnet 101004 1500 - - 1 ieee - 0 0
1005 trnet 101005 1500 - - 1 ibm - 0 0
tundra_switch1#
tundra_switch1#show vtp status
VTP Version : 2
Configuration Revision : 15
Maximum VLANs supported locally : 1005
Number of existing VLANs : 8
VTP Operating Mode : Server
VTP Domain Name : tundra
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0xE6 0x6C 0xFD 0xDA 0x1B 0xCC 0x7B 0x8A
Configuration last modified by 172.16.2.16 at 3-1-93 04:03:13
Local updater ID is 172.16.2.16 on interface V12 (lowest numbered VLAN interface)
tundra_switch1#
    
```

第 3 步要求用户在交换机之间配置以太通道和 802.1Q 的骨干。在两台交换机上以太通道的配置是一样的，只要这两台都是 Catalyst 3550 交换机就可以。范例 1-50 演示了 tundra_switch1 交换机上吉比特以太通道的配置。

范例 1-50 使用 802.1Q 封装配置吉比特以太通道

```

tundra_switch(config)#interface gigabitEthernet 0/1
tundra_switch(config-if)#switchport trunk encapsulation dot1q
! 802.1q trunking
tundra_switch(config-if)#switchport mode trunk
tundra_switch(config-if)#channel-group 1 mode on
! EtherChannel Configuration
Creating a port-channel interface Port-channel1
tundra (config-if)#exit
00:23:18: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:23:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed state
to up
tundra_switch(config)#interface gigabitEthernet 0/2
tundra_switch(config-if)#switchport trunk encapsulation dot1q
tundra_switch(config-if)#switchport mode trunk
tundra_switch(config-if)#channel-group 1 mode on
    
```

对此时的配置来说，VTP 应当在交换机之间工作，应该能够 ping 通所有的本地设备。需要配置 SVI 和路由选择协议来实现 VLAN 之间的连接。在这个模型中，需要将 tundra_switch1 交换机设置为 VLAN 100 和 200 的根桥，可以使用全局配置命令 **spanning-tree vlan 100 root** 和 **spanning-tree vlan 200 root** 来完成本任务。这个宏使用扩展的系统 ID 来设置 VLAN 的优先级为 24 576，这将使得交换机成为根。在 VLAN 200 上也应当启用 VTP 的剪枝。VLAN 剪枝可以通过 VLAN 配置命令 **vtp pruning** 来启用。可以使用 **show spanning-tree root** 命令来验证 STP 的状态，如范例 1-51 所示。在这个范例的尾部是 **show interface** 命令，它可以验证在两台交换机之间的以太通道上启用了 VTP 剪枝。

范例 1-51 在 tundra_switch1 交换机上验证 STP 和 VTP 剪枝

```
tundra_switch1#show spanning-tree root
```

Vlan	Root ID	Root Cost	Hello Time	Max Age	Fwd Dly	Root Port
VLAN0001	32768 0004.275e.f0c0	3	2	20	15	Po1
VLAN0002	32768 0004.275e.f0c1	3	2	20	15	Po1
VLAN0100	24676 000a.8a0e.ba80	0	2	20	15	
VLAN0200	24776 000a.8a0e.ba80	0	2	20	15	

```
tundra_switch1#
tundra_switch1#show int port-channel 1 switchport
Name: Po1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2,100,200
<<<text omitted>>>
```

在下面两步中，我们在交换机上配置 SVI 和路由接口。在 tundra_switch1 交换机上，需要 3 个 SVI 和一个路由端口来实现完整的 IP 连接。一个 SVI（也就是 interface VLAN 2）用于管理 VLAN，而另外两个 SVI——interface VLAN 100 和 interface VLAN 200——用于其他的路由器。在你想配置为路由接口的那个端口上，首先需要启用路由，然后使用 **no switchport** 接口命令。范例 1-52 显示了 tundra_switch1 的必需配置。

范例 1-52 SVI 和路由接口的配置

```
!
ip routing
! IP routing must be enabled for routed INTs
!
interface FastEthernet0/10
 no switchport
! Disable switching
 ip address 10.16.128.16 255.255.255.0
! Assign an IP address
!
-----SVI CONFIG----->
interface Vlan2
 ip address 172.16.2.16 255.255.255.0
!
interface Vlan100
 ip address 172.16.100.16 255.255.255.0
!
interface Vlan200
 ip address 172.16.200.16 255.255.255.0
 no ip redirects
```

这个实验的最后部分就是将 EIGRP 配置为路由选择协议。IP 已在先前的步骤中启用了，所以在这步中就不需要了。为了配置实验的三层交换部分，只需要在路由器和以太网交换机上

配置 EIGRP。这实际上和路由器上的操作是一样的。这时也可以配置 HSRP 协议。再次说明，在交换机上配置 HSRP 的语法和在路由器上是一样的。《CCIE 实验指南（第 1 卷）》有对 EIGRP 和 HSRP 协议的深入配置，因此，这儿只是在配置中将它们列出来。如果你对使用的配置选项有任何问题，参考《CCIE 实验指南（第 1 卷）》。范例 1-53 列出了 tundra_switch1 交换机的完整配置，随后是交换机的路由表和 EIGRP 邻居。注意这个交换机有 5 个 EIGRP 邻居。

范例 1-53 tundra_switch1 交换机的完整配置

```
hostname tundra_switch1
!
enable secret 5 $1$nt35$131XBsGKT6BmA1KHMqj1V1
!
ip subnet-zero
ip routing
!
spanning-tree extend system-id
spanning-tree vlan 100 priority 24576
spanning-tree vlan 200 priority 24576
!
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport trunk pruning vlan 2,100,128,200
 switchport mode trunk
 no ip address
!
<<<text omitted>>>
!
interface FastEthernet0/3
 switchport access vlan 200
 switchport mode access
 no ip address
!
<<<text omitted>>>
!
interface FastEthernet0/7
 switchport access vlan 100
 switchport mode access
 no ip address
!
interface FastEthernet0/8
 switchport access vlan 100
 switchport mode access
 no ip address
!
interface FastEthernet0/9
 no ip address
!
interface FastEthernet0/10
 no switchport
 ip address 10.16.128.16 255.255.255.0
!
<<<text omitted>>>
 no ip address
!
interface GigabitEthernet0/1
 switchport trunk encapsulation dot1q
 switchport trunk pruning vlan 2,100,128,200
 switchport mode trunk
 no ip address
```

（待续）

```

channel-group 1 mode on
!
interface GigabitEthernet0/2
 switchport trunk encapsulation dot1q
 switchport trunk pruning vlan 2,100,128,200
 switchport mode trunk
 no ip address
 channel-group 1 mode on
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan2
 ip address 172.16.2.16 255.255.255.0
!
interface Vlan100
 ip address 172.16.100.16 255.255.255.0
!
interface Vlan200
 ip address 172.16.200.16 255.255.255.0
 no ip redirects
 standby 200 ip 172.16.200.1
 standby 200 priority 101
 standby 200 preempt
!
router eigrp 2003
 network 10.0.0.0
 network 172.16.0.0
 no auto-summary
 no eigrp log-neighbor-changes
!
ip classless
ip http server!
line con 0
line vty 0 4
 password cisco
 login
line vty 5 7
end
tundra_switch1#
tundra_switch1#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, 3 subnets
C       172.16.200.0 is directly connected, Vlan200
C       172.16.2.0 is directly connected, Vlan2
C       172.16.100.0 is directly connected, Vlan100
 10.0.0.0/24 is subnetted, 2 subnets
D       10.16.10.0 [90/409600] via 10.16.128.100, 03:25:34, FastEthernet0/10
C       10.16.128.0 is directly connected, FastEthernet0/10
tundra_switch1#
tundra_switch1#show ip eigrp neighbors
IP-EIGRP neighbors for process 2003
H   Address                Interface    Hold Uptime    SRTT    RTO   Q   Seq Type
                               (sec)        (ms)          Cnt Num

```

(待续)

4	172.16.100.100	V1100	13	03:22:58	1524	5000	0	6
3	172.16.100.101	V1100	11	03:23:01	1488	5000	0	7
2	10.16.128.100	Fa0/10	10	03:30:33	1080	5000	0	5
1	172.16.200.102	V1200	13	03:32:03	419	2514	0	5
0	172.16.200.101	V1200	14	03:32:06	204	1224	0	8

tundra_switch1#

范例 1-54 列出了 tundra_bak 交换机的相关配置部分。

范例 1-54 tundra_bak 交换机的配置

```
hostname tundra_switch2
!
enable secret 5 $1$nt35$131XBSgKT6BmA1KHMqj1V1
!
spanning-tree extend system-id
!
interface Port-channel1
  switchport trunk encapsulation dot1q
  switchport trunk pruning vlan 2,100,128,200
  switchport mode trunk
  no ip address
!
<<<text omitted>>>
!
interface FastEthernet0/5
  switchport access vlan 200
!
interface GigabitEthernet0/1
  switchport trunk encapsulation dot1q
  switchport trunk pruning vlan 2,100,128,200
  switchport mode trunk
  no ip address
  channel-group 1 mode on
!
interface GigabitEthernet0/2
  switchport trunk encapsulation dot1q
  switchport trunk pruning vlan 2,100,128,200
  switchport mode trunk
  no ip address
  channel-group 1 mode on
!
interface Vlan1
  no ip address
  shutdown
!
interface VLAN2
  ip address 172.16.2.15 255.255.255.0
  no ip directed-broadcast
  no ip route-cache
!
ip default-gateway 172.16.2.16
!
line con 0
line vty 0 4
  password cisco
  login
line vty 5 7
end
```

配置在其他路由器上除了 IP 地址几乎是相同的。因此，为了简化起见，并不是所有的配置都在这儿列出。

范例 1-55 tundra_prime 和 frozen_rtr 路由器的配置

```
hostname tundra_prime
!
interface FastEthernet3/0
 ip address 172.16.200.101 255.255.255.0
 duplex auto
 speed auto
 standby 200 preempt
 standby 200 ip 172.16.200.1
!
router eigrp 2003
 network 172.16.0.0
 no auto-summary
 no eigrp log-neighbor-changes
!

hostname frozen_rtr
!
interface loopback 20
 ip address 10.16.10.100 255.255.255.0
!
interface Ethernet0/0
 ip address 10.16.128.100 255.255.255.0
!
router eigrp 2003
 network 10.0.0.0
 no auto-summary
!

hostname tundra_bak
!
interface Ethernet0/1
 ip address 172.16.200.102 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 standby priority 95
 standby preempt
 standby 200 ip 172.16.200.1
!
router eigrp 2003
 network 172.16.0.0
 no auto-summary
!

hostname products
!
!
interface Ethernet0
 ip address 172.16.100.100 255.255.255.0
 no ip directed-broadcast
 media-type 10BASE-T
!
router eigrp 2003
 network 172.16.0.0
 no auto-summary
!
```

1.6 实验 2: 配置 802.1w RSTP 和 802.1s MST、 三层交换以及 VLAN 映射

1.6.1 练习场景

交换上做了极大提高的一个方面就是冗余和容错恢复，使用 IEEE 802.1w RSTP 和 IEEE 802.1s MST，生成树现在可以在几百毫秒内收敛，而不是 802.1d 所需要的 50s。当配置大型的生产性网络时，用户会花费很多金钱来实现冗余和备份。现在通过新技术实现极好的恢复时间可以帮助用户从金钱的投资中收到最多的回报。

1.6.2 实验练习

著名的 Walker 医生建立了 Walker 儿童医院，主要是治疗小孩子在腿和骨骼方面的问题。这个医院的网络已经运行了 802.1d STP 来实现冗余，但是管理员发现恢复时间太长了。在链路发生故障的情况下，区域（例如 surgery 和 recovery）之间的关键业务需要非常快的收敛速度。

你的任务就是配置一个工作的 IP 网络，并且使用下面严格的设计原则来配置以太 3550 交换机：

- 按照图 1-38 配置 Walker 儿童医院的网络。使用 EIGRP 作为路由选择协议，2003 作为所有路由器上的自治系统 ID。
- 按照图 1-38 配置所有的 IP 地址。所有标注的接口应当互相 ping 通对方。
- 参看“实验目的”部分来了解配置细节。

1.6.3 实验目的

- 按照图 1-38 所示，将 EIGRP 配置为路由选择协议。使用 2003 作为自治系统的 ID。
- 给 walker1 配置管理接口 172.16.192.16/24，给 walker2 配置 172.16.192.13/24。这些地址应当可达。
- 配置 walker1 和 walker2 交换机之间的两个接口。不要将这些接口配置为以太通道组。在这个模型中，使用一个吉比特以太网接口，一个 100 Mbit/s 的接口作为备份。
- 将 walker1 配置为 VTP 服务器，将 walker2 配置为 VTP 客户端。使用 VTP 域名 walker，VTP 口令 psv2。
- 按照图 1-38，将其他的接口配置为接入端口，并把它们分配到 VLAN 中。给 VLAN 20 分配 6 个接口，这些将是 admin VLAN 的边界端口。
- 配置 802.1w RSTP 和 802.1s MST。为了快速收敛，将所有的主机配置为边界端口。使用 walker 作为 MST 的名字。

- 将 walker1 交换机配置为 VLAN 2 到 300 范围内所有 VLAN 的根桥。

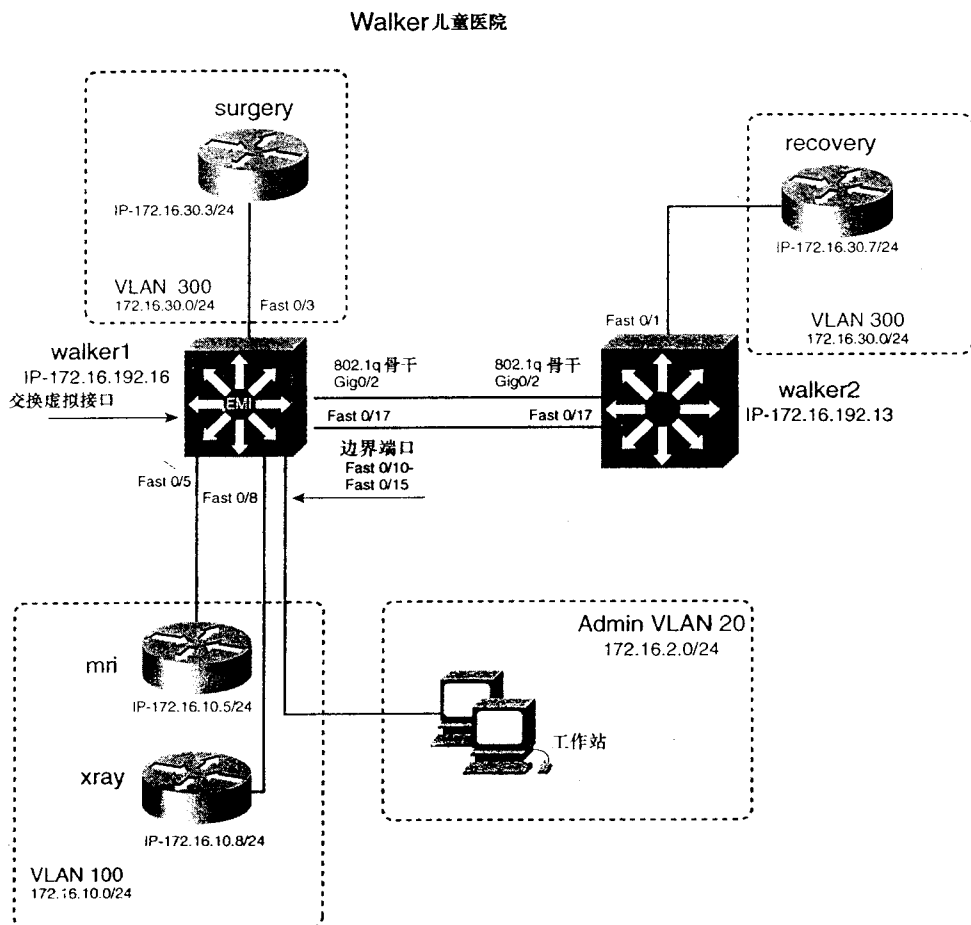


图 1-38 Walker 儿童医院

- 在 walker1 交换机上配置任意数量的 SVI，来提供网络中所有 VLAN 之间的路由。局域网应当有完全的 IP 连接。所有描述的 IP 地址都应当是可以 ping 通的。
- 在 admin VLAN 中，人们在共享文件和使用未经授权的应用程序方面有一些问题。配置这个 VLAN，使得这个 VLAN 的用户不能再共享文件或者使用网络内的应用程序。

1.6.4 需要的设备

- 一个基于 IP 的工作站，4 个思科的路由器，一个安装了 EMI 软件的 Catalyst 3550 交换机，以及一个支持 MST 和 RSTP 的 Catalyst 交换机。只需要一台是安装了 EMI 软件的 Catalyst 3550 交换机。一台路由器可以替换 VLAN 20 中的一个工作站。VLAN20 中至少应当有一个活动的 IP 设备用于测试。

- 交换机之间需要两个背对背的 100BASE-T 链路或者一个吉比特以太链路连接起来。其他的路由器应当使用 5 类线缆和适当的交换机相连，参看图 1-38。

1.6.5 物理布局和预规划

- 按照图 1-38 所示，将交换机和路由器相连。
- 本实验侧重于配置以太网交换机。

1.6.6 实验步骤

按照图 1-38 所示的那样，将所有的路由器连接到交换机上。可以在两个交换机之间使用两个吉比特以太连接或者 100 Mbit/s 以太连接。你的选择不会影响整个实验的运行结果。

我们回忆一下配置 3550 以太网交换机的七步过程。

第 1 步 配置交换机管理。

第 2 步 配置 VTP 和 VLAN，并且给 VLAN 分配端口/接口。

第 3 步 使用 EtherChannel、802.1Q 和 ISL 封装配置交换机之间的连接。

第 4 步 (可选) 控制 STP 和 VLAN 传播。

第 5 步 (可选) 配置 SVI。

第 6 步 (可选) 配置路由端口。

第 7 步 (可选) 配置三层交换。

第 1 步是配置交换机管理。这包括设置主机名、口令和交换机上的管理地址。范例 1-56 列出了 walker1 的管理部分。walker2 的配置和 walker1 几乎是一样的，除了 IP 地址用的是 172.16.192.13。

范例 1-56 到目前为止 walker1 交换机的管理部分

```
hostname walker1
! Set the hostname
!
enable secret 5 $1$nt35$131XBsgKT6BmA1KHMqj1V1
! Enable Secret=cisco
!
<<<text omitted>>>
!
interface Vlan1
no ip address
shutdown
!
interface Vlan192
! MNGT VLAN and IP
ip address 172.16.192.16 255.255.255.0
<<<text omitted>>>
!
```

第 2 步要求用户配置 VTP 和 VLAN。需要对 SVI、接入端口和管理 VLAN 配置相应的 VLAN。在这个模型中，需要配置 5 个 VLAN：VLAN 20、100、192、200 和 300。

在 3550 交换机上，可以在全局配置模式下使用 **vlan x** 命令建立 VLAN。在输入 VLAN 的号码后，也可以给 VLAN 输入名字。walker1 交换机的 VTP 模式是 server 模式，而 walker2 交换机是 client 模式。VTP 的域名是 walker，口令是 psv2。确保 VTP 的域名有相同的大小写，口令也是一样。域名和口令是大小写敏感的。确保 VTP 服务器的修订版本号大于 VTP 客户交换机的版本号，否则，这两者之间不会同步。VTP 的域名和模式可以从 VLAN 数据库或者 VLAN 配置模式下进行配置。范例 1-57 演示了 walker1 交换机上的配置。

范例 1-57 在 walker1 交换机上配置 VTP

```
walker1#vlan database
walker1(vlan)#vtp domain walker
walker1(vlan)#vtp server
walker1(vlan)#vtp password psv2
```

这一步也要求用户配置物理端口的属性并将这些端口分配到 VLAN 中去。范例 1-58 演示了到目前为止 walker1 交换机上的 VLAN 和端口配置。因为你正在配置 RSTP，所以必须使用接口命令 **spanning-tree portfast** 来配置边界端口。

范例 1-58 配置 VLAN 的端口成员

```
hostname walker1
!
<<<text omitted>>>
!
interface FastEthernet0/3
 switchport access vlan 300
! assigned to VLAN 300
 switchport mode access
 spanning-tree portfast
! Portfast used in 802.1w
 no ip address
!
interface FastEthernet0/5
 switchport access vlan 100
! assigned to VLAN 100
 switchport mode access
 spanning-tree portfast
! Portfast used in 802.1w
 no ip address
!
interface FastEthernet0/8
 switchport access vlan 100
! assigned to VLAN 100
 switchport mode access
 spanning-tree portfast
! Portfast used in 802.1w
 no ip address
!
```

当配置 VLAN 的端口范围时，使用 **range** 命令就会非常容易。范例 1-59 演示了给 VLAN 20 配置 6 个管理接口时，**range** 命令的使用。

范例 1-59 配置 VLAN 的范围

```
walker1(config)#interface range fastEthernet 0/10 - 15
walker1(config-if-range)#switchport mode access
walker1(config-if-range)#switchport access vlan 20
walker1(config-if-range)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION
%Portfast will be configured in 6 interfaces due to the range command
but will only have effect when the interfaces are in a non-trunking mode.
walker1(config-if-range)#exit
```

可以使用 **show vlan** 命令和 **show vtp status** 命令来验证 VLAN 和 VTP，如范例 1-60 所示。

范例 1-60 验证 VTP 和 VLAN 的状态

```
walker1#show vlan
VLAN Name                Status    Ports
-----
1    default                active    Fa0/1, Fa0/2, Fa0/4,
                                   Fa0/6, Fa0/9, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
20   psv2_vlan20             active    Fa0/10, Fa0/11, Fa0/12, Fa0/13
      Fa0/14, Fa0/15
100  psv2_vlan100            active    Fa0/5, Fa0/8
192  psv2_vlan192            active
300  psv2_vlan300            active    Fa0/3
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    BrgdMode Trans1 Trans2
-----
1    enet    1000001   1500    -      -      -      -      -      0      0
20   enet    1000020   1500    -      -      -      -      -      0      0
100  enet    100100    1500    -      -      -      -      -      0      0
192  enet    100192    1500    -      -      -      -      -      0      0
300  enet    100300    1500    -      -      -      -      -      0      0
1002 fddi    101002    1500    -      -      -      -      -      0      0
1003 tr     101003    1500    -      -      -      -      srb    0      0
1004 fdnet  101004    1500    -      -      1      -      ieee   0      0
1005 trnet  101005    1500    -      -      1      -      ibm    0      0
walker1#
walker1#show vtp status
VTP Version                : 2
Configuration Revision      : 3
Maximum VLANs supported locally : 1005
Number of existing VLANs    : 9
VTP Operating Mode          : Server
VTP Domain Name             : walker
VTP Pruning Mode            : Enabled
VTP V2 Mode                 : Disabled
VTP Traps Generation        : Disabled
MD5 digest                  : 0xEF 0xD8 0x4D 0x0A 0x57 0x8F 0x7E 0x14
Configuration last modified by 172.16.192.16 at 3-1-93 01:10:51
Local updater ID is 172.16.192.16 on interface Vl192 (lowest numbered VLAN interface)
walker1#
```

第 3 步要求用户在交换机之间配置 802.1Q 的骨干。在两台交换机上的配置是一样的，只要这两台都是 Catalyst 35xx 系列的交换机就可以。范例 1-61 演示了 walker1 交换机上对接口 Gig 0/2 和 Fast 0/17 的 802.1Q 骨干的配置。

范例 1-61 使用 802.1Q 封装配置吉比特以太网通道

```
walker1(config)#interface gigabit 0/2
walker1(config-if)#switchport trunk encapsulation dot1q
walker1(config-if)#switchport mode trunk
walker1(config-if)#exit
walker1(config)#interface fast 0/17
walker1(config-if)#switchport trunk encapsulation dot1q
walker1(config-if)#switchport mode trunk
walker1(config-if)#exit
```

对此时的配置来说，VTP 应当在交换机之间工作，应该能够 ping 通所有的本地设备。使用 **show vtp status** 命令来验证 VTP，确保两个交换机具有相同的 VTP 修订号和相同的 VLAN 数量。

下一部分的配置需要启用 802.1s 和 802.1w 生成树。此时通过使用 **spanning-tree portfast** 命令在所有的非骨干接口上配置，可使 RSTP 部分启用。只有在 802.1s 或者 MST 启用时，RSTP 才会完全启用。在 walker1 和 walker2 交换机上的 MST 配置是相同的，除了 walker1 交换机使用 **spanning-tree mst 1 root primary** 命令来给 VLAN 2 到 300 设置根桥。定义一个 STP 的实例，MST1，并将 VLAN 2 到 300 分配给这个实例。MST 的名字是 walker，修订号是 1。范例 1-62 演示了在 walker1 交换机上配置 MST 和 RSTP。

范例 1-62 在 walker1 交换机上配置 MST 和 RSTP

```
walker1(config)#spanning-tree mst config      ←Enter MST configuration mode
walker1(config-mst)#name walker                ←MST name
walker1(config-mst)#revision 1                ←MST revision number
walker1(config-mst)#instance 1 vlan 2-300     ←assign VLANs 2-300 to instance 1
walker1(config-mst)#exit                      ←apply changes !important!
walker1(config)#spanning-tree mst 1 root primary ←Set root for instance 1
walker1(config)#spanning-tree mode mst        ←enable MST
```

可以使用 **show spanning-tree mst 1** 和 **show spanning-tree root** 命令来验证 MST 的状态，如范例 1-63 所示。你应当可以看到 VLAN 2 到 300 在 MST 的实例 1 里，而且 MST 的实例 1 应当是 MST 的根。在这个模型中，MAC 地址 000a.8a0e.ba80 是根。

范例 1-63 验证 MST

```
walker1#show spanning-tree mst 1
##### MST01          vlans mapped: 2-300
Bridge address 000a.8a0e.ba80 priority 24577 (24576 sysid 1)
Root this switch for MST01
Interface      role state cost      prio type
-----
Fa0/3          desg FWD  200000    128 edge P2P
Fa0/5          desg FWD  2000000   128 edge SHR
Fa0/8          desg FWD  200000    128 edge P2P
Fa0/10         desg FWD  2000000   128 edge SHR
```

(待续)

```
Fa0/17      desg FWD  200000  128  P2P
Gi0/2       desg FWD  20000   128  P2P
walker1#show spanning-tree root
```

MST Instance	Root ID	Root Cost	Hello Time	Max Age	Fwd Dly	Root Port
MST00	32768 0004.275e.f0c0	200000	2	20	15	Gi0/2
MST01	24577 000a.8a0e.ba80	0	2	20	15	

```
walker1#
```

为了验证 MST 和 RSTP 的功能，执行下面的测试。从 surgery 路由器发起一个到 recovery 路由器的扩展 ping，使用高数量的 ping，例如 10 000 个。当你正在 ping 接口时，关闭活动的骨干（在这个模型中，是吉比特以太）。应当看到 RSTP 几乎是瞬间收敛，具有 99% 的成功率！范例 1-64 显示了正在做的 RSTP 测试。

范例 1-64 MST 和 RSTP 的测试

```
surgery#ping
Protocol [ip]:
Target IP address: 172.16.30.7
Repeat count [5]: 10000
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 172.16.30.7, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!..!    <-Gig 0/2 dropped
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 99 percent (9998/10000), round-trip min/avg/max = 1/2/20 ms
surgery#
```

在下面两步中，我们在 walker1 交换机上配置 SVI 并启用路由。需要 4 个 SVI——每个 VLAN 一个 SVI，一个 SVI 用于管理 VLAN。一个 SVI，也就是 interface VLAN 192，用于管理 VLAN，还需要 3 个 SVI——interface VLAN 20 用于 admin VLAN，interface VLAN 100 和 interface VLAN 300 用于路由器。范例 1-65 显示了 walker1 交换机上的必要配置。

范例 1-65 SVI 接口配置

```
interface Vlan20
 ip address 172.16.2.16 255.255.255.0
!
interface Vlan100
 ip address 172.16.10.16 255.255.255.0
!
interface Vlan192
 ip address 172.16.192.16 255.255.255.0
!
interface Vlan300
 ip address 172.16.30.16 255.255.255.0
```

这个实验的最后部分就是将 EIGRP 配置为路由选择协议。需要使用全局配置命令 **ip routing** 来启用 IP 路由。为了配置实验的三层交换部分，只需要在路由器和以太网交换机上配置 EIGRP。这和路由器上的操作是一样的。范例 1-66 列出了 walker1 交换机的完整配置，随后是交换机的 EIGRP 邻居。注意这个交换机有 4 个 EIGRP 邻居。

范例 1-66 walker1 交换机的完整配置

```
hostname walker1
!
enable secret 5 $1$0TsK$C95mG2YeDzQ4w3ecs0CkS0
!
ip subnet-zero
ip routing
!
spanning-tree mode mst
spanning-tree extend system-id
!
spanning-tree mst configuration
  name walker
  revision 1
  instance 1 vlan 2-300
!
spanning-tree mst 1 priority 24576
!
<<<text omitted>>>
!
interface FastEthernet0/3
  switchport access vlan 300
  switchport mode access
  no ip address
  spanning-tree portfast
!
<<<text omitted>>>
!
interface FastEthernet0/5
  switchport access vlan 100
  switchport mode access
  no ip address
  spanning-tree portfast
!
<<<text omitted>>>
!
!
interface FastEthernet0/8
  switchport access vlan 100
  switchport mode access
  no ip address
  spanning-tree portfast
!
<<<text omitted>>>
!
!
interface FastEthernet0/10
  switchport access vlan 20
  switchport mode access
  no ip address
  spanning-tree portfast
!
interface FastEthernet0/11
  switchport access vlan 20
```

(待续)

```
switchport mode access
no ip address
spanning-tree portfast
!
interface FastEthernet0/12
switchport access vlan 20
switchport mode access
no ip address
spanning-tree portfast
!
interface FastEthernet0/13
switchport access vlan 20
switchport mode access
no ip address
spanning-tree portfast
!
interface FastEthernet0/14
switchport access vlan 20
switchport mode access
no ip address
spanning-tree portfast
!
interface FastEthernet0/15
switchport access vlan 20
switchport mode access
no ip address
spanning-tree portfast
!
<<<text omitted>>>
!
interface FastEthernet0/17
switchport trunk encapsulation dot1q
switchport mode trunk
no ip address
!
interface GigabitEthernet0/2
switchport trunk encapsulation dot1q
switchport mode trunk
no ip address
!
interface Vlan1
no ip address
shutdown
!
interface Vlan20
ip address 172.16.2.16 255.255.255.0
!
interface Vlan100
ip address 172.16.10.16 255.255.255.0
!
interface Vlan192
ip address 172.16.192.16 255.255.255.0
!
interface Vlan300
ip address 172.16.30.16 255.255.255.0
!
router eigrp 2003
network 172.16.0.0
auto-summary
no eigrp log-neighbor-changes
!
ip classless
```

(待续)

```

ip http server
!
line con 0
line vty 5 15
!
end
walker1#
walker1#show ip eigrp neighbors
IP-EIGRP neighbors for process 2003

```

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RT0	Q Cnt	Seq Num	Type
3	172.16.10.5	Vl100	14	00:03:02	1048	5000	0	5	
2	172.16.30.3	Vl300	12	00:03:04	1	3000	0	9	
1	172.16.30.7	Vl300	13	00:03:06	1208	5000	0	10	
0	172.16.10.8	Vl100	14	00:03:06	1516	5000	0	9	

```

walker1#

```

范例 1-67 列出了 walker2 交换机的相关配置。

范例 1-67 walker2 交换机的配置

```

hostname walker2
!
enable secret 5 $1$0TsK$C95mG2YeDzQ4w3ecs0CkS0
!
spanning-tree mode mst
spanning-tree extend system-id
!
spanning-tree mst configuration
name walker
revision 1
instance 1 vlan 2-300
!
interface FastEthernet0/1
switchport access vlan 300
switchport mode access
no ip address
spanning-tree portfast
!
interface FastEthernet0/17
switchport trunk encapsulation dot1q
switchport mode trunk
no ip address
!
interface GigabitEthernet0/2
switchport trunk encapsulation dot1q
switchport mode trunk
no ip address
!
interface Vlan1
no ip address
shutdown
!
interface VLAN192
ip address 172.16.192.13 255.255.255.0
no ip directed-broadcast
no ip route-cache
!
ip default-gateway 172.16.192.16

```

实验的最后部分需要用户控制对 VLAN 20 的访问。为了防止管理工作站之间互相使用

IP 服务，可以将它们定义为被保护端口。回忆一下，被保护端口和其他被保护端口之间不能互相通信，但是被保护端口和非被保护端口在同一交换机上是可以通信的。范例 1-68 演示了使用 **range** 命令配置被保护端口。

范例 1-68 在 walker1 交换机上被保护端口的配置

```
walker1(config)#interface range fastEthernet 0/10 - 15
walker1(config-if-range)#switchport protected
walker1(config-if-range)#^z
walker1#
walker1#show interfaces fastEthernet 0/10 switchport
Name: Fa0/10
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 20 (psv2_vlan20)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: true
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled

Voice VLAN: none (Inactive)
Appliance trust: none
walker1#
```

范例 1-69 显示了 surgery、mri、xray 和 recovery 路由器的配置。

范例 1-69 surgery、mri、xray 和 recovery 路由器的配置

```
hostname surgery
!
interface FastEthernet3/0
 ip address 172.16.30.3 255.255.255.0
 duplex auto
 speed auto
!
router eigrp 2003
 network 172.16.0.0
 no auto-summary
 no eigrp log-neighbor-changes
!

hostname mri
!
interface Ethernet0/1
 ip address 172.16.10.5 255.255.255.0
!
router eigrp 2003
 network 172.16.0.0
 no auto-summary
!

hostname xray
```

(待续)

```
!  
interface Ethernet0/1  
  ip address 172.16.10.8 255.255.255.0  
!  
router eigrp 2003  
  network 172.16.0.0  
  no auto-summary  
!  
  
hostname recovery  
!  
interface Ethernet5  
  ip address 172.16.30.7 255.255.255.0  
  no ip directed-broadcast  
  media-type 10BASE-T  
!  
router eigrp 2003  
  network 172.16.0.0  
  no auto-summary  
!
```


第二部分

控制网络传播和网络访问

第2章 配置路由映射和策略性路由

第 2 章

配置路由映射和策略性路由

也许对路由映射（route map）最形象化的描述，就是它类似于网络的输送管道——并不是因为它们可以用于修正或者改进某些疏漏，而是因为它们可以应用于许多情况来解决许多问题。有时，它们可能并不是“最棒的解决方案”，但是它们是最有效的。当你学习完配置和使用路由映射后，你很快就会明白为什么有些工程师称它们为**路由管道**。在策略性路由（PBR）中，例如，当流量必须遵循某条特定的路径通过互连网络时，可以使用路由映射。这条路径可能和路由选择协议转发流量所采用的路径不一样。策略性路由和路由映射一样，允许网络工程师在实质上凌驾于路由表之上，影响流量所经过的路径。

也可以使用一些方法来应用路由映射。下面的列表含有关于路由映射的某些更通用更强大的应用：

- 在路由选择协议之间重分发时进行路由过滤；
- 在 BGP 邻居上进行路由控制和属性修改；
- 在路由选择协议之间重分发时进行路由度量的修改或者标记；
- 策略性路由（PBR）。

一旦你将路由映射放入到你的工具箱中，你将会得到思科路由器上最强大最灵活的配置工具之一。本章介绍了如何配置和使用路由映射以及如何配置策略性路由。

2.1 路由映射介绍

语句。如果某个条件为真，那么就做某件事情。路由映射允许用户定义路由策略，它在路由器检查转发表之前进行。因此，可以定义路由策略来优先于不同的路由进程。这就是为什么路由映射是可以在路由器上执行的最强大的命令之一。范例 2-1 重点强调了路由映射的逻辑。

范例 2-1 路由映射的逻辑

```
route-map route_map_name permit 10
  match criteria_1
  set perform_action_1
route-map route_map_name permit 20
  match criteria_2
  set perform_action_2
  set perform_action_3
route-map route_map_name permit 30
  match criteria_3 criteria_4 criteria_5
  set perform_action_2
  set perform_action_4
  set perform_action_5
route-map route_map_name deny 65536      ←implicit deny at the end
  match everything
```

总之，路由映射工作在下列方式下：

1. 本质上，一个进程——无论它是重分发进程、策略性路由进程还是其他一些诸如网络地址翻译（NAT）的进程——通过一个文本来调用路由映射。

2. 反之，路由映射有条件或者 **match** 语句，它们通常是（但也不一定总是）一个访问控制列表或者扩展的访问控制列表。例如，边界网关协议（BGP）可以匹配一个自治系统号码（ASN）或者是不同的属性。**match** 语句后面可以紧跟着 **set** 语句。

如果 **match** 语句返回的结果为真，那么 **set** 语句就会执行。

范例 2-2 显示了在重分发过程中路由映射的功能。

范例 2-2 在重分发过程中路由映射的功能

```
router ospf 2001
  redistribute eigrp 65001 subnets route-map route_map_name  ←Call the route-map
                                                                ←and send EIGRP routes for comparison
!
route-map route_map_name permit 10      ←Route-map with the lowest sequence number
                                         gets executed first
  match ip address access_list           ←Call access-list, the IF of the route-map
  set condition                          ←If access-list is true, THEN do something
!
route-map route_map_name permit 20      ←Next highest sequence number
                                         gets executed
  match ip address access_list           ←Call access-list, the IF of the route-map
  set condition                          ←If access-list is true, THEN do something
!
route-map route_map_name deny 65536     ←Implicit deny at the end all route-maps
  match ip address all_routes            This will not show up in the config
```

下一个范例是一个实际的路由映射的语法。范例 2-3 演示了在重分发过程中如何应用路由映射。

范例 2-3 在重分发过程中路由映射的应用

```

router ospf 65
 log-adjacency-changes
 log-adjacency-changes
 redistribute eigrp 65001 subnets route-map set_tag ←Call the route-map "set_tag"
 network 10.10.3.0 0.0.0.255 area 0
 default-metric 10
!
access-list 10 permit 172.16.32.0 0.0.0.255 ←Match the 172.16.32.0/24 subnet
access-list 11 permit 172.16.1.0 0.0.0.255 ← Match the 172.16.1.0/24 subnet
!
route-map set_tag permit 100 ←Route-map "set_tag"
 match ip address 10 ←Call access-list 10, if this is true then...
 set tag 10 ←If access-list is true set the tag of 10
!
route-map set_tag permit 200 ←If no match above, try and match the following:
 match ip address 11 ←access list 11
 set metric-type type-1 ←If the ACL is true, set the OSPF metric type to 1
 set tag 11 ←and set a tag of 11
!
route-map set_tag permit 300 ←All other routes get a tag of 300
 set tag 300
!

```

在先前的范例中，一个路由映射用于控制并且标记从 EIGRP 重分发到 OSPF 中的路由。在 OSPF 的重分发进程中，会调用一个叫 `set_tag` 的路由映射。这个路由映射含有三部分。第一部分调用访问控制列表（ACL）10，这个访问控制列表允许 172.16.32.x 的网络，并给路由打上标记 10。第二部分调用访问控制列表 11，这个访问控制列表匹配 IP 地址 172.16.1.x。如果匹配生效，那么当路由重分发时，它的度量会被设为 OSPF 类型 1，最终路由会打上标签 11。路由映射的最后一部分不会调用任何一个访问控制列表，所以所有的路由都会匹配，并且执行 `set` 的条件。在这个范例中，路由器设置标签为 300。可以通过这种方式来给网络建立路由标记的文档性帮助，也可以利用标记来区分你想过滤的路由，或者在打了标记的路由上执行某些动作。

路由映射有下列一些共性：

- 路由映射是按照从最低的序列号到最高的序列号的顺序执行的。可以通过序列号来编辑或者修改路由映射。
- 如果和路由映射中的某一序列号匹配了，那么以后序列号的路由映射语句就不再执行。
- 可以使用路由映射来允许或者拒绝和 `match` 语句匹配的信息。
- 如果在一个路由映射序列号中有多个 `match` 语句被引用，那么所有的 `match` 语句都必须匹配，才能产生为真的结果。
- 如果路由映射应用在一个策略性路由的环境中，那么和 `match` 条件不匹配的数据包会根据路由表转发出去。
- 如果在路由映射的序列号中没有 `match` 语句，那么所有的路由和数据包都会匹配。`set` 语句的行为会应用于所有的路由或者数据包。
- 如果在路由映射序列号中的 `match` 语句没有相应的访问控制列表，那么所有的路由都会匹配。`set` 语句的行为会应用在所有的路由上。

- 就像访问控制列表一样，在路由映射策略的末尾有一个隐含的 **deny** 语句。
- 可以使用路由映射基于下面的这些条件来建立策略：
 - IP 地址；
 - 终端系统 ID；
 - 应用程序；
 - 协议；
 - 数据包的大小。

2.1.1 配置路由映射

路由映射的语法粗略地讲是由三个单独的思科命令组成的，这取决于路由映射要完成什么样的任务和它要调用什么样的进程。因为本章一直围绕着路由映射的配置，我们将要详细讨论下面的这些命令：

- **route-map** 命令；
- **match** 命令；
- **set** 命令。

当配置路由映射时，可以遵循一个基本的五步配置过程，这取决于路由映射的应用，也许需要额外的配置，例如 BGP 的团体属性或者是策略性路由。

- 第1步** （可选）配置路由映射可能正在 **match** 命令中使用的访问控制列表、AS 路径列表或者任何其他匹配条件，这个应当是先做的，这样你就不会调用任何空的访问控制列表或者 AS 路径列表。
 - 第2步** 配置路由映射的序列号，这是通过 **route-map name permit | deny sequence_number** 命令来完成的。确保在序列号之间留出一些空间，使得将来更新或者修改时比较方便。具有最低序列号的路由映射语句是最先执行的。
 - 第3步** 定义匹配的条件并且配置 **match** 语句，它将在路由映射的单实例中引用。可以通过路由映射的 **match** 命令来完成。如果没有任何 **match** 命令，所有的数据包或者路由都会匹配。
 - 第4步** （可选）定义 **set** 行为并且配置在这个路由映射的单实例中将要使用的 **set** 语句。可以通过在路由映射中配置 **set** 命令来完成这个任务。
 - 第5步** （可选）配置路由映射可能正在 **match** 命令中使用的访问控制列表、AS 路径列表或者任何其他匹配条件。
 - 第6步** 应用路由映射。再次说明，取决于路由映射的应用，它可以有许多种方法的使用。部分非常常见的应用包括路由重分发、策略性路由和 BGP。
- 记住这个配置过程，我们将更加详细地讨论配置路由映射的 3 个主要的命令。

一、route-map 命令

route-map 命令的完整语法如下：

```
route-map route_map_name [permit | _deny][ sequence_number_1-65535]
```

route_map_name 也称为 *map tag*，是路由映射的基于文本的名字。这个名字是惟一的，

并且逻辑上组织并定义了整个路由映射的策略。这是在重分发或者其他的进程中用来调用路由映射的名字。

permit 和 **deny** 关键字是可选的，默认的关键字是 **permit**。如果路由映射被重分发过程所调用，而关键字设置为 **permit**，那么匹配路由映射中 **match** 条件的这些路由就会被重分发，如果关键字设置为 **deny**，那么同样的路由会被拒绝。

如果路由映射被策略性路由所调用，满足路由映射中的 **match** 条件，而且关键字设置为 **permit**，那么数据包将会被策略性路由。再次说明，**permit** 是默认的关键字。如果使用了 **deny** 关键字，那么数据包会按照正常的转发过程被转发。

sequence-number 代表路由映射语句被执行的顺序。当调用路由映射时，具有最低序列号的语句优先执行。如果在路由映射中具有最低序列号的语句没有被匹配到，那么下一个较高序列号的语句将会被执行。这个过程不断重复，直到被匹配到或者不再有路由映射的语句存在。如果被匹配到，就会停止执行对那个数据包或者路由的行为，下一个数据包或者路由会再次从路由映射中具有最低序列号的语句开始执行这个过程。默认的序列号是 10。

注意：当建立路由映射时，给序列号留出空间以便将来编辑使用。将路由映射中的第一个序列号置为 10 或者 100，这取决于你期望路由映射有多大。通过使用 10 或者 100 的增量，给路由映射留出了 65 到 650 的空间。起始于一个较高的序列号，并且在序列号之间留出空间会使编辑路由映射的工作变得很容易。路由映射的最大序列号是 65 535。

二、match 命令

match 命令允许用户定义路由映射的匹配条件。例如，可以使用 **match** 命令调用一个访问控制列表来比较路由。**match** 语句也可以匹配路由标记、路由类型或者一个数据包的长度。BGP 提供许多独有的 **match** 语句，我们将在第 4 章和第 5 章中讨论。表 2-1 列出了在思科 IOS 软件版本 12.2 中可用的 **match** 参数。

表 2-1 思科 IOS 软件 12.2 中的 match 命令

命令	它匹配的是什么	命令	它匹配的是什么
as-path	BGP AS 路径列表	ip	IP 特定的信息
clns	CLNS*信息	length	数据包的长度
community	BGP 团体属性	metric	路由度量值
extcommunity	BGP/VPN**扩展团体列表	route-type	路由类型
interface	路由中第一跳接口的信息	tag	路由标记

* CLNS=无连接的网络服务

** VPN=虚拟专用网络

到目前为止 **match ip address** 命令是最常用的 **match** 命令。**match ip address** 命令允许用户调用一个标准的、扩展的或者是扩展范围的访问控制列表。可以在重分发、BGP、NAT、策略性路由和其他的功能中使用它。**match** 命令的语法如下：

```
match ip {address [access_list | prefix-list] | next-hop [access_list] | route-source [access_list | prefix-list]}
```

在 IP 网络中，这个命令允许用户将有一个网络地址的路由和特定的访问控制列表或者前缀列表中的一项或几项进行匹配。可以使用标准的、扩展的或者是扩展范围的访问控

制列表。

next-hop 关键字允许用户将路由的下一跳和特定的访问控制列表中的一项或几项进行匹配。BGP 也主要使用它。

route-source 关键字允许用户匹配宣告路由器的 IP 地址的路由/网络。可以使用标准的、扩展的或者是扩展范围的访问控制列表。对于 BGP，也可以使用一个前缀列表。

注意：当在 BGP 中使用 **match ip address** 命令时，只能使用路由映射来过滤发送出去的更新包。**match ip address** 不支持进入方向的 BGP 路由更新。

next-hop 关键字主要用于 BGP，但是也可以用于在重分发中对基于下一跳的路由进行匹配。在这种情况下，路由器检查 NEXT_HOP 属性进行比较。

route-source 关键字允许用户匹配一台路由器宣告的 IP 地址。如果你查看 IP 路由表，路由 172.16.3.0/24 是从 IP 地址为 172.16.2.1 的路由器中宣告出来的，那么 **route-source** 关键字用于匹配宣告路由器的 IP 地址 172.16.2.1。在后面的内容中，把这些命令应用到范例中向用户演示它们是如何工作的。

三、范例：匹配路由源和 IP 地址

在这个模型中，在一个公用的局域网段上有 4 台路由器运行两种路由选择协议。earp 和 holliday 路由器正在运行 EIGRP 作为路由选择协议，ringo 和 clanton 路由器正在运行 OSPF。ringo 路由器在功能上是一个 OSPF 的自治系统边界路由器 (ASBR)，通过在 EIGRP 和 OSPF 之间重分发。ringo 路由器从 earp 和 holliday 路由器接收几条路由，如图 2-1 所示。

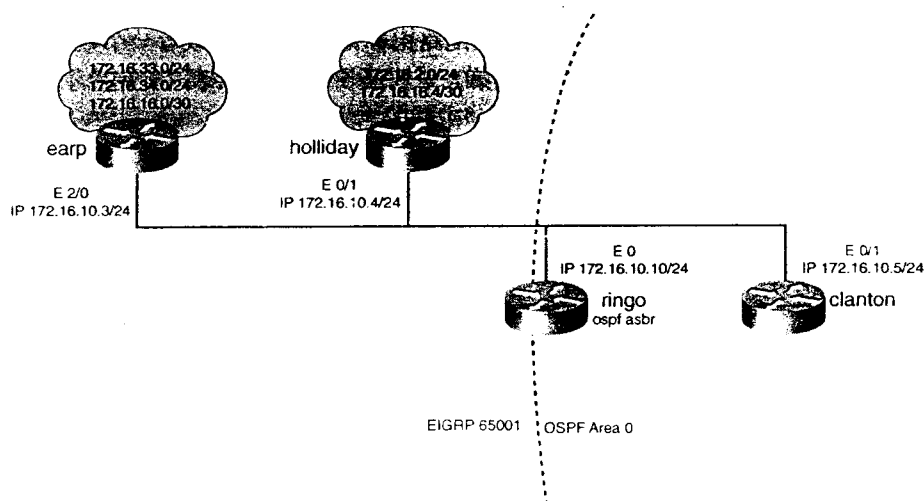


图 2-1 路由映射范例：匹配路由源和 IP 地址

在这个范例中，路由映射应用在 ringo 路由器上的 EIGRP 重分发到 OSPF 的进程中。名为 **set_tag3** 的路由映射在 ringo 路由器的 OSPF 重分发进程中调用。路由映射的第一个实例，**route-map set_tag3 permit 100**，将执行 **IP route-source** 的匹配条件。这个语句只匹配访问控制列表 5 中宣告路由器产生的路由，在这种情况下，是地址 172.16.10.3。不仅允许这些路由重分发，而且还会设置路由标记为 3。

注意: 当在 OSPF 中使用路由映射时, 宣告路由器的 OSPF router ID 成为路由源。在 OSPF 网络中使用 **route-source** 关键字时, 使用 OSPF router ID 作为路由源的 IP 地址。

范例 2-4 列出了 ringo 路由器的转发/路由表。注意路由 172.16.16.0/30、172.16.33.0/24 和 172.16.34.0/24 来自 earp 路由器, 即 172.16.10.3。172.16.2.0/24 和 172.16.16.4/30 这些路由来自 holliday 路由器, 即 172.16.10.4。

范例 2-4 ringo 路由器的转发/路由表

```
ringo# show ip route
<<<text omitted>>>
C    192.168.10.0/24 is directly connected, Loopback20
    172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D    172.16.33.0/24 [90/1812992] via 172.16.10.3, 00:07:13, Ethernet0
D    172.16.34.0/24 [90/1812992] via 172.16.10.3, 00:07:13, Ethernet0
D    172.16.16.4/30 [90/2195456] via 172.16.10.4, 00:07:13, Ethernet0
D    172.16.16.0/30 [90/1787392] via 172.16.10.3, 00:07:13, Ethernet0
C    172.16.10.0/24 is directly connected, Ethernet0
D    172.16.2.0/24 [90/307200] via 172.16.10.4, 00:07:14, Ethernet0
/ringo#
```

范例 2-5 列出了 ringo 路由器上路由映射的配置。

范例 2-5 ringo 路由器的配置

```
!
interface Loopback20
 ip address 192.168.10.10 255.255.255.0
!
interface Ethernet0
 ip address 172.16.10.10 255.255.255.0
!
<<<text omitted>>>
!
router eigrp 65001
 network 172.16.0.0
 network 192.168.10.0
 no auto-summary
 no eigrp log-neighbor-changes
!
router ospf 7
 log-adjacency-changes
 redistribute eigrp 65001 subnets route-map set_tag3 ←Route-map called
 network 172.16.10.10 0.0.0.0 area 0
 default-metric 10
!
access-list 5 permit 172.16.10.3 ←Match route 172.16.10.3 only
access-list 50 permit any ←Match all remaining routes
!
route-map set_tag3 permit 100
 match ip route-source 5 ←Match routes from 172.16.10.3 / ACL 5
 set tag 3 ←set the tag to three
!
route-map set_tag3 permit 200 ←Second Route-map instance
 match ip address 50 ←Call access-list 50 to match all routes
 set metric-type type-1 ←Set OSPF route type to External Type-1
 set tag 500 ←Set the tag to 500 for these routes
```


在先前的范例中，路由映射的第二个实例调用访问控制列表 50。访问控制列表 50 允许剩余的路由被重分发，并且设置路由的标记为 500，设置 metric-type 为 OSPF 的外部类型 1。

通过查看 OSPF 的数据库，可以清楚地看见标记以及重分发是如何工作的。范例 2-6 演示了在 ringo 路由器上使用 **show ip ospf database** 命令。

范例 2-6 show ip ospf database 命令

```
ringo# show ip ospf database
      OSPF Router with ID (192.168.10.10) (Process ID 7)
      Router Link States (Area 0)
Link ID      ADV Router      Age      Seq#          Checksum Link count
172.16.10.5  172.16.10.5      1005     0x8000000B   0x18D8   1

192.168.10.10 192.168.10.10    1027     0x8000000A   0x7017   1
      Net Link States (Area 0)
Link ID      ADV Router      Age      Seq#          Checksum
172.16.10.5  172.16.10.5      1005     0x8000000A   0x75DA

      Type-5 AS External Link States
Link ID      ADV Router      Age      Seq#          Checksum Tag
172.16.2.0   192.168.10.10    1027     0x80000009   0x10E0   500
172.16.16.0  192.168.10.10    1027     0x80000009   0xD285   3
172.16.16.4  192.168.10.10    1027     0x80000009   0x3BA6   500
172.16.33.0  192.168.10.10    1027     0x80000009   0x291B   3
172.16.34.0  192.168.10.10    1027     0x80000009   0x1E25   3
192.168.10.0 192.168.10.10    1027     0x80000009   0x8BB0   500
ringo#
```

检查下游 OSPF 路由器（例如 clanton 路由器）的路由表，可以看见 **set metric-type type-1** 命令的效果。注意在范例 2-6 中，172.16.2.0/24、192.168.10.0/24 和 172.16.16.4/30 这些路由是 OSPF 外部类型 1 的路由。通常，或者是默认情况下，路由是 OSPF 外部类型 2 的路由。关于不同的链路状态宣告（LSA）类型及其使用的更详细信息，参考《CCIE 实验指南（第 1 卷）》。在后面的内容中我们会学习到更多关于 **set** 命令的用法。范例 2-7 列出了 clanton 路由器的转发表。

范例 2-7 clanton 路由器的转发表

```
clanton# 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
O E1 192.168.10.0/24 [110/20] via 172.16.10.10, 04:47:26, Ethernet0/0
     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
O E2 172.16.33.0/24 [110/10] via 172.16.10.10, 04:47:27, Ethernet0/0
O E2 172.16.34.0/24 [110/10] via 172.16.10.10, 04:47:27, Ethernet0/0
O E1 172.16.16.4/30 [110/20] via 172.16.10.10, 04:47:27, Ethernet0/0
O E2 172.16.16.0/30 [110/10] via 172.16.10.10, 04:47:27, Ethernet0/0
C    172.16.10.0/24 is directly connected, Ethernet0/0
O E1 172.16.2.0/24 [110/20] via 172.16.10.10, 04:47:27, Ethernet0/0
clanton#
```

BGP 使用许多特定的 **match** 命令，如后两个范例所示。BGP 可以使用路由映射调用 AS 路径而不是访问控制列表来控制路由信息。表 2-2 列出了 **match as-path** 命令的语法。

表 2-2 match as-path 命令

命令	描述
match as-path /1-199/	在 BGP 中用来匹配自治系统的列表。有效的路径列表的号码是 1~199

可以在 BGP 中使用这个命令来匹配自治系统的路径（AS-PATH）属性。

BGP 中另外一个特殊的 **match** 命令是 **match community**。可以使用路由映射来匹配和设置 BGP 中的团体（COMMUNITY）属性。

match community 命令的语法如下：

match [community|extcommunity|exactmatch]

community 关键字主要用在 BGP 中调用 IP 团体列表。对于标准的团体列表，有效范围是 1~99，对于扩展的团体列表，范围是 100~199。而且，可以使用 **exact-match** 来执行关于团体的严格匹配。

也可以使用路由映射在 NAT 的应用中进行访问控制列表的匹配，并且基于输出接口选择全局地址池。**match interface** 命令用于 NAT 的应用中。也可以使用它来匹配下一跳是接口的路由，例如指向一个接口的静态路由。表 2-3 显示了 **match interface** 命令的语法。

表 2-3 match interface 命令

命令	描述
match interface <i>interface_name</i>	在 NAT 的路由映射中用来匹配输出接口，或者有一个接口作为下一跳地址而不是 IP 地址的路由

标记可以非常有效地允许用户控制和跟踪重分发的路由。思科的路由器允许网络工程师使用一个数字来标记某些路由。这个标记值是一个特殊的值，可以随着路由选择协议进行传输。这个标记值不会影响路由的转发决定，并且对路由选择协议没有特定的值。标记主要用于重分发中给路由打标记或者贴标签。当路由被打上标记后，标记值可以用在重分发进程中控制路由的重分发。RIPv2、OSPF、Integrated IS-IS、EIGRP、BGP 和 CLNS 这些协议支持标记，IGRP 和 RIPv1 不支持标记。为了查看标记，使用 **show eigrp topology ip_address subnet_mask** 和 **show ip ospf database** 命令来分别了解 EIGRP 和 OSPF。也可以使用扩展的 **show ip route** 命令 **show ip route ip_address** 来查看标记值。表 2-4 显示了 **match tag** 命令的语法。

表 2-4 match tag 命令

命令	描述
match tag [0-4294967295]	使用 match tag 命令匹配路由选择协议中的标记值，例如 RIPv2、IS-IS、OSPF、EIGRP、BGP 和 CLNS

在思科 IOS 软件 12.0 中，也可以使用路由映射来匹配特定的路由类型，例如，可以匹配

EIGRP 外部路由或者 OSPF 外部类型 1 或者类型 2 的路由。**match route-type** 关键字允许用

户匹配下面的路由类型：

- OSPF 外部类型 1 (O E1) 和类型 2 的路由 (O E2)、NSSA 外部类型 1 (O N1) 类型 2 (O N2)、域内路由 (O) 和域间路由 (O IA)；
- EIGRP 外部路由 (D EX)；
- IS-IS 一级路由 (L1) 和二级路由 (L2)；
- BGP 外部路由。

match route-type 命令的语法如下：

```
Match route-type {local|internal|external[type-1|type-2]||level-1||level2|nssa-external}
```

可以在 **match route-type** 命令中使用下面的这些关键字：

External——外部路由 (BGP、EIGRP 和 OSPF 类型 1/2)

Internal——内部路由 (包括 OSPF 域内/域间和 EIGRP 路由)

level-1——IS-IS 一级路由

level-2——IS-IS 二级路由

local——BGP 本地产生的路由

nssa-external——NSSA 外部路由

虽然在同一行中可以使用多个 **match** 语句，但你应当在每一行中只使用一个 **match** 条件，这将使故障排查和修改路由映射变得很容易。

四、范例：匹配标记

还是操作前面那个范例中的模型，下面的范例有一台路由器叫做 clanton，运行 OSPF 和 IGRP 作为路由选择协议。clanton 路由器在重分发时会调用一个路由映射，这个路由映射会将标记为 3 的路由和 OSPF 外部类型 1 (O E1) 的路由重分发到 IGRP 里。图 2-2 显示了这种新的网络模型。

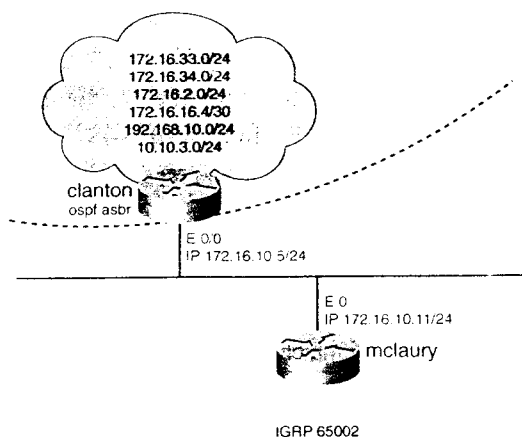


图 2-2 路由映射范例：匹配标记

范例 2-8 列出了 clanton 路由器的路由表，并将 OSPF 外部类型 1 的路由着重显示。范例 2-9 列出了 clanton 路由器上的 OSPF 数据库，着重显示了标记为 3 的路由。

范例 2-8 clanton 路由器的路由表

```
clanton# 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
O E1 192.168.10.0/24 [110/20] via 172.16.10.10, 01:59:17, Ethernet0/0
    172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
O E2   172.16.33.0/24 [110/10] via 172.16.10.10, 01:49:44, Ethernet0/0
O E2   172.16.34.0/24 [110/10] via 172.16.10.10, 01:49:44, Ethernet0/0
O E2   172.16.16.0/30 [110/10] via 172.16.10.10, 01:49:44, Ethernet0/0
C       172.16.10.0/24 is directly connected, Ethernet0/0
O E2   172.16.2.0/24 [110/10] via 172.16.10.10, 01:49:44, Ethernet0/0
    10.0.0.0/24 is subnetted, 1 subnets
O E1   10.10.3.0 [110/20] via 172.16.10.10, 01:59:18, Ethernet0/0
clanton#
```

范例 2-9 clanton 路由器上的 OSPF 数据库

```
clanton# show ip ospf database
OSPF Router with ID (172.16.10.5) (Process ID 7)

Router Link States (Area 0)
Link ID      ADV Router   Age         Seq#         Checksum Link count
172.16.10.5  172.16.10.5  557        0x80000006  0x22D3   1
192.168.10.10 192.168.10.10 1642       0x80000005  0x7A12   1

Net Link States (Area 0)
Link ID      ADV Router   Age         Seq#         Checksum
172.16.10.5  172.16.10.5  557        0x80000005  0x7FD5

Type-5 AS External Link States
Link ID      ADV Router   Age         Seq#         Checksum Tag
10.10.3.0    192.168.10.10 1642       0x80000004  0x9904   500
172.16.2.0   192.168.10.10 1133       0x80000005  0x87DF   3
172.16.16.4  192.168.10.10 1642       0x80000004  0x45A1   500
172.16.33.0  192.168.10.10 1133       0x80000005  0x3117   3
172.16.34.0  192.168.10.10 1133       0x80000005  0x2621   3
192.168.10.0 192.168.10.10 1643       0x80000004  0x95AB   500
clanton#
```

为了控制 OSPF 和 IGRP 之间的重分发，在重分发的过程中使用路由映射。这个路由映射必须有两个路由映射的实例。第一个实例将匹配所有标记值为 3 的 OSPF 的路由，第二个实例将匹配 OSPF 外部类型 1 的路由。范例 2-10 列出了 clanton 路由器上重要部分的配置。

范例 2-10 clanton 路由器上路由映射的配置

```
hostname clanton
!
router ospf 7
 network 172.16.10.5 0.0.0.0 area 0
!
router igrp 65002
 redistribute ospf 7 route-map match_me ←Redistribute OSPF and call the route-map
 network 172.16.0.0
```

(待续)

```
default-metric 10000 100 254 1 1500
!
route-map match_match_me permit 10
  match tag 3                                ←Match routes with a tag 3
!
route-map match_match_me permit 20
  match route-type external type-1          ←Match OSPF external type-1 routes
```

为了验证重分发和路由映射的正常工作，查看 mclaury 路由器的路由表。范例 2-11 列出了 mclaury 路由器的路由表。注意标记值为 3 的路由为：172.16.2.0/24、172.16.33.0/24 和 172.16.34.0/24。也要注意 OSPF 外部类型 1 的路由：192.168.10.0/24 和汇总了子网的路由 10.0.0.0/8。

范例 2-11 mclaury 路由器的路由表

```
mclaury# 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
I    192.168.10.0/24 [100/1200] via 172.16.10.5, 00:00:50, Ethernet0
     172.16.0.0/24 is subnetted, 4 subnets
I      172.16.33.0 [100/1200] via 172.16.10.5, 00:00:50, Ethernet0
I      172.16.34.0 [100/1200] via 172.16.10.5, 00:00:50, Ethernet0
C      172.16.10.0 is directly connected, Ethernet0
I      172.16.2.0 [100/1200] via 172.16.10.5, 00:00:50, Ethernet0
I     10.0.0.0/8 [100/1200] via 172.16.10.5, 00:00:50, Ethernet0
mclaury#
```

也可以使用路由映射来匹配一条路由的度量值。这是出现在路由/转发表中的路由的度量值。如果一条 OSPF 路由相关的度量值是 20，那么 **match metric 20** 用于匹配这条路由。表 2-5 列出了使用 **match metric** 命令的语法。

表 2-5 match metric 命令

命令	描述
match metric [0-4294967295]	输入度量值，它出现在路由器的路由/转发表中

使用图 2-1 作为指导，范例 2-12 列出了 clanton 路由器的路由表，随后是用于匹配具有度量值 20 的 OSPF 路由的路由映射配置。这个范例将 OSPF 具有度量值 20 的路由重分发到 EIGRP 中。

范例 2-12 演示 match metric 路由映射

```
clanton# 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
O E1 192.168.10.0/24 [110/20] via 172.16.10.10, 00:19:58, Ethernet0/0
    172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
O E2 172.16.33.0/24 [110/10] via 172.16.10.10, 00:19:59, Ethernet0/0
O E2 172.16.34.0/24 [110/10] via 172.16.10.10, 00:19:59, Ethernet0/0
O E1 172.16.16.4/30 [110/20] via 172.16.10.10, 00:19:59, Ethernet0/0
O E2 172.16.16.0/30 [110/10] via 172.16.10.10, 00:19:59, Ethernet0/0
C 172.16.10.0/24 is directly connected, Ethernet0/0
O E2 172.16.2.0/24 [110/10] via 172.16.10.10, 00:19:59, Ethernet0/0
    10.0.0.0/24 is subnetted, 1 subnets
O E1 10.10.3.0 [110/20] via 172.16.10.10, 00:19:59, Ethernet0/0

hostname clanton
!
<<<text omitted>>>
!
router ospf 7
 network 172.16.10.5 0.0.0.0 area 0
!
router eigrp 65002
 redistribute ospf 7 route-map match_metric_20
 network 172.16.0.0
 default-metric 10000 100 254 1 1500
!
ip classless
!
route-map match_metric_20 permit 10
 match metric 20
!
```

在先前的范例中，10.10.3.0/24、172.16.16.4/30 和 192.168.10.0/24 这些路由被重分发到 EIGRP 中。

match clns address 命令用于 ISO CLNS 路由中，和 IP 路由的使用方式是一样的。**match clns address** 命令调用一个 CLNS 的地址列表，并将测试的地址和它进行比较。**next-hop** 和 **route-source** 关键字在策略性路由中用于调用一个 OSI 的过滤设置。使用 CLNS 命令和 IP 的方式是一样的。**match clns** 命令的语法如下：

```
match clns {address [name]|next-hop [filter set]|route-source [filter set]}
```

使用 **match clns address** 命令来匹配网络地址在特定的 OSI 过滤设置组中的一条或者多条路由。

next-hop 关键字用于匹配下一跳地址在特定的 OSI 过滤设置组中的一条或者多条路由。

route-source 关键字用于匹配在特定的 OSI 过滤设置组中路由器所宣告的一条或者多条路由。

我们要讨论的最后一条 **match** 命令是 **match length** 命令。这条 **match** 命令主要用在策略性路由中，当访问控制列表不足以去做流量的分发时使用。**match length** 命令允许用户匹配三层数据包的长度（以字节表示），包括头和尾。可以使用路由映射以这样的方式工作：从一条路上发送小的交互式的数据包，例如远程登录的流量，而大的突发数据流量从另外一条路上发送。表 2-6 列出了 **match length** 命令的语法。

表 2-6 match length 命令

命令	描述
Match length [min_packet_length_0-2147483647] [max_packet_length_0-2147483647]	用于匹配三层数据包的长度，以字节表示，包括所有的头和尾部的信息，必须输入最小和最大的数据包的长度

关于 `match length` 命令的范例，参看 “配置策略性路由” 部分的内容。

五、set 命令

`set` 命令在路由映射的实例被成功地匹配后执行。`Set` 命令是可选的，可以被忽略。如果你在重分发的过程中使用了路由映射，或者仅仅是过滤网络，那么没有必要使用 `set` 命令，除非你想给路由打标记或者想进一步影响路由。如果在路由映射实例中没有 `match` 语句，那么所有的路由都会执行所有的 `set` 命令。在路由映射的每一个实例中，也可以使用多个 `set` 命令。这里所讨论的 `set` 命令在思科 IOS 软件版本 12.2 中支持，并且在表 2-7 中列出。`set` 命令已经被分成 3 个类别：BGP 特定的 `set` 命令、路由选择协议/重分发特定的 `set` 命令和策略性路由特定的 `set` 命令。策略性路由特定的 `set` 命令在“配置策略性路由”部分中讲述。

表 2-7 set 命令	
set 命令	描述
BGP 特定的 set 命令	
as-path	BGP AS_PATH 属性的添加字符串
community extcommunity	设置 BGP_COMMUNITY 属性
comm-list	BGP 团体列表
dampening	设置 BGP 路由波动惩罚参数
local-preference	设置 BGP_LOCAL_PREF 路径属性
origin	设置 BGP 原点代码
weight	设置 BGP 权重
路由选择协议/重分发特定的 set 命令	
metric	对目的路由选择协议设置度量值
metric-type	对目的路由选择协议设置度量类型
tag automatic-tag	对目的路由选择协议设置标记值
策略性路由特定的 set 命令	
default	设置默认的路由信息
interface	设置输出接口，用于点对点链路
ip	IP 特定的信息

六、BGP 特定的 set 命令

我们所讲述的第一个 `set` 命令是和 BGP 有关的。这一部分介绍在 BGP 中不同的 `set` 命令的语法和它们的基本应用。关于 BGP 特定的 `set` 命令在应用上的特定和详细的信息，参看第 8 章和第 9 章。

在 BGP 中 `set as-path` 命令用于给公认必遵过渡属性 AS_PATH 添加一个或者多个自治系统的号码。在 BGP 中，这可以用来影响路由决策。BGP 将当前的 AS_PATH 属性中添加了一个或者多个自治系统号码的路径看作是次优的路径，在多归路的 BGP 网络中这是非常有用的方法。

注意: **prepend** 命令的作用实质上是让 AS 路径更长——因此形成一个次优的路径——并不是完全地改变它。当在生产性的环境中使用 **set as-path prepend** 命令时，总是使用和这条路由起始的自治系统相同的 ASN。如果使用不同的 ASN，那么那个 ASN 收到这条路由后，那个自治系统/路由器不会接受这条路由。通过直接添加一个不同的自治系统来修改 AS 路径会影响 AS 路径属性所提供的内置的环路预防机制。某些思科 IOS 软件甚至不允许用户添加一个和你自己的自治系统不同的 AS 路径。出于示范的目的，我们的某些范例显示了添加一个不同的自治系统号码，这是通过着重显示添加的那个自治系统号码来做到的。

注意 **prepend** 命令在入站和出站方向的路由映射上如何工作，有很重要的不同。当 **prepend** 命令用在出站方向的路由映射上时，被添加的自治系统号码是在宣告路由器的自治系统号码之后进行添加。这是因为被添加的自治系统号码在路由更新数据包被发送之前就要归位。当发送路由更新数据包时，宣告路由器的自治系统号码是列表中的第一个。例如，如果你在一个发送方向的路由映射中添加了 AS 10 10，而你的路由器所处的自治系统号码为 5，那么接收路由器/邻居的 AS 路径为 5 10 10。

当在入站方向的路由映射上应用 **prepend** 命令时，被添加的自治系统实际上放在原始的 AS 路径之前。这是因为被添加的自治系统号码实际上是在从它的邻居收到路由之后发生的。例如，如果你在入站方向的路由映射上添加了 AS 10 10，你从邻居收到路由的 AS 路径为 5 500，那么这条路由的 AS 路径最终会为 10 10 5 500。**set as-path** 命令的语法如下：

```
set as-path {prepend [as_path1|as_path2[as_path3]] [[tag]]}
```

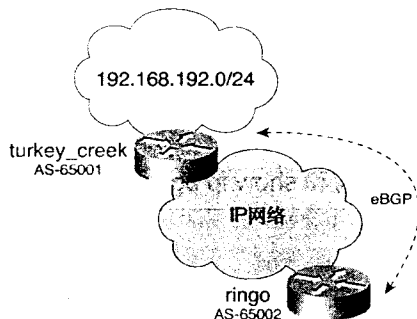
在 BGP 网络中使用 **set as-path** 命令来修改 AS 路径属性，可通过添加一个或者多个自治系统号码来实现。可以在入站和出站方向的路由映射上使用这条命令。

在 BGP 中，**tag** 关键字主要用于从重分发的内部网关协议 (IGP) 的路由的标记中恢复 AS 路径的信息。

在 BGP 中，**set as-path tag** 命令主要是用于当从 IGP 重分发时，保留一个一致和正确的 AS 路径。思科 BGP 的实施者在将 BGP 重分发到 IGP 时，自动地会将 AS 路径转换成标记的形式，而将 IGP 的路由重分发到 BGP 时，AS 路径的信息会丢失。为了从重分发的 IGP 路由的标记中恢复 AS 路径的信息，使用 **set as-path tag** 命令。

1. 范例：设置 AS 路径

图 2-3 显示的网络模型有两台路由器，它们之间运行 BGP。turkey_creek 路由器处在自治系统 65001 里，而 ringo 路由器处在自治系统 65002 里。turkey_creek 路由器通过 BGP



宣告网络 192.168.192.0/24。在这个范例里，在 turkey_creek 路由器的出站方向的路由更新数据包里，将会使用一个路由映射添加自治系统号码 65001 2001 给它的 AS 路径。

范例 2-13 列出了在 turkey_creek 路由器上操作 AS 路径属性的配置。

范例 2-13 turkey_creek 路由器的 BGP 配置

```
hostname turkey_creek
!
<<<text omitted>>>
!
router bgp 65001
 no synchronization
 network 192.168.192.0
 neighbor 172.16.100.10 remote-as 65002
 neighbor 172.16.100.10 ebgp-multihop 10
 neighbor 172.16.100.10 update-source Loopback20
 neighbor 172.16.100.10 route-map set_as out      ←Call route-map "set_as" for
                                                    outbound updates
!
route-map set_as permit 10
 set as-path prepend 65001 2001                  ←prepend AS-PATH with 65001 2001
!
```

你可能会错误地认为路由 192.168.192.0/24 的 AS 路径是 65001 2001 65001；但是，千万别忘了，这个命令的含义是“添加”。因为这是一个出站方向的路由映射，然而，被添加的自治系统号码在宣告发出去之前就会发生。因此，“被添加”的 AS 路径对于下游路由器来说，会出现在起始的自治系统之后。对于下游路由器，也就是 ringo 路由器，AS 路径将会是 65001 65001 2001。范例 2-14 在 ringo 路由器上使用 **show ip bgp** 命令通过输出结果演示了这一点。

范例 2-14 在 ringo 路由器上使用 show ip bgp 命令

```
ringo# show ip bgp 192.168.192.0
BGP routing table entry for 192.168.192.0/24, version 4
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
    65001 65001 2001
      172.16.200.10 (metric 1915392) from 172.16.200.10 (192.168.192.7)
        Origin IGP, metric 0, localpref 100, valid, external, best
ringo#
```

范例 2-15 将路由映射应用于 ringo 路由器上入站方向的路由更新数据包。这个路由映射将会给来自 turkey_creek 路由器的路由添加 AS 路径 2001 65002 65001。因为这是一个入站方向的路由映射，在 ringo 路由器上最终的 AS 路径将是 2001 65002 65001 65001。在入站方向的路由映射上，添加功能就像它的名字的含义一样。范例 2-15 列出了 ringo 路由器相关部分的配置，随后是 **show ip bgp** 命令。

范例 2-15 ringo 路由器的配置和 show ip bgp 命令

```
Hostname ringo
!
<<<text omitted>>>
!
```

(待续)

```

router bgp 65002
no synchronization
bgp log-neighbor-changes
neighbor 172.16.200.10 remote-as 55001
neighbor 172.16.200.10 ebgp-multihop 10
neighbor 172.16.200.10 update-source Loopback20
neighbor 172.16.200.10 route-map m modify_as in      ←Route-map "modify_as" is called
!
route-map modify_as permit 10
set as-path prepend 2001 65002 65001                ←Prepended AS
!

ringo# show ip bgp 192.168.192.0
BGP routing table entry for 192.168.192.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
    2001 65002 65001 65001
      172.16.200.10 (metric 1915392) from 172.16.200.10 (192.168.192.7)
        Origin IGP, metric 0, localpref 100, valid, external, best
ringo#

```

set community 命令在 BGP 中用来设置不同的团体属性。就像在后续章节中对 BGP 的讨论一样，团体是一种给一组路由应用策略的非常强大和有效的方法。团体是一个可选过渡路由属性，可在 BGP 的对等体之间通信。**set community** 命令允许用户组织团体的成员关系。当路由成为一个团体的成员之后，它们可以分配策略，例如“不要将这个路由输送给任何一个 E-BGP 的邻居或者将这条路由宣告为具有 Internet 团体属性”。为了在 BGP 中发送团体属性，必须使用 **neighbor a.b.c.d send-community** 命令。

在思科 IOS 软件版本 12.2 中，**set community** 命令的语法如下：

```

set community { community-number 1-4294967200|AA|NN|no-export|no-advertise |internet
|local-AS [additive] }|none

```

使用 **set community** 命令来指定路由的团体属性，并且将特定的策略应用在这些路由上。有效的参数和值如下：

- **Community number**——有效的数字范围为 1~4 294 967 200；这些路由将会使用这个团体数字。
- **AA: NN**——这种格式也可以用来指定团体。AA 是一个 16 位的 ASN，范围为 1~65 535，NN 是一个 1~65 440 之间的 16 位的数字。
- **Internet**——Internet 团体属性。宣告这条路由具有 Internet 属性，并且任何路由器都属于它。
- **no-export**——不要将这条路由通告给 E-BGP 的对等体。具有这种团体属性的路由可以发送给在一个联盟内部的其他子自治系统的对等体。
- **local-as**——不要将这些路由通告给本地自治系统之外的对等体。这些路由不会通告给其他的自治系统或者在一个联盟内部的子自治系统。
- **no-advertise**——不要将这些路由通告给任何对等体（内部的或者外部的），用于 I-BGP 的对等体。
- **Additive**——（可选）将团体属性添加到已有的团体属性列表。
- **None**——将通过路由映射匹配的前缀的团体属性清除。

2. 范例：设置 BGP 团体属性

考虑和先前的范例中同样的网络模型，turkey_creek 和 ringo 路由器之间运行 BGP（参看图 2-4）。turkey_creek 路由器处在 AS 65001 中，而 ringo 路由器处在 AS 65002 中。turkey_creek 路由器将会通过 BGP 宣告网络 192.168.192.0/24，并且将它的团体属性置为 7。turkey_creek 路由器也会宣告另外一条路由 128.168.192.0/24。turkey_creek 路由器会将这条路由的团体属性值置为 8，并设置团体属性为 **no-export**。no-export 团体属性告诉 ringo 路由器不要将这条路由发送给任何 E-BGP 的邻居。

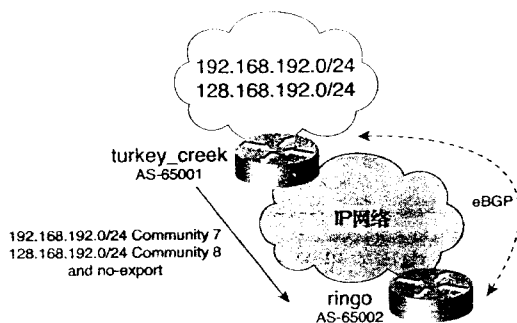


图 2-4 路由映射范例：设置团体

范例 2-16 列出了用于完成这个任务的路由映射。

范例 2-16 turkey_creek 上的路由映射针对于团体的配置

```

Hostname turkey_creek
!
<<<text omitted>>>
!
router bgp 65001
 no synchronization
 network 128.168.192.0 mask 255.255.255.0
 network 192.168.192.0
 neighbor 172.16.100.10 remote-as 65002
 neighbor 172.16.100.10 ebgp-multihop 10
 neighbor 172.16.100.10 update-source Loopback20
 neighbor 172.16.100.10 send-community      ←send-community must be enabled
 neighbor 172.16.100.10 route-map set_communities out ←route-map "set_communities"
 called
!
<<<text omitted>>>
!
access-list 10 permit 192.168.192.0 0.0.0.255      ←allow network 192.168.192.0/24 only
access-list 11 permit 128.168.192.0 0.0.0.255      ←allow network 128.168.192.0/24 only
!
route-map set_communities permit 100
 match ip address 10                                ←Match ip access-list 10 or 192.168.192.0/24
 set community 7                                    ←set the community to 7
!
route-map set_communities permit 200
 match ip address 11                                ←Match ip access-list 11 or 128.168.192.0/24
 set community 8 no-export                          ←set the community to 8 and don't export to
                                                    future E-BGP peers

```

通过观察 ringo 路由器的路由表，可以发现路由 192.168.192.0/24 具有团体属性 7。路由 128.168.192.0/24 具有团体属性 8 的 **no-export** 选项的设置。范例 2-17 列出了在 ringo 路由器上使用 **show ip bgp** 命令的输出。

范例 2-17 在 ringo 路由器上具有团体属性设置的路由

```

ringo# show ip bgp 192.168.192.0
BGP routing table entry for 192.168.192.0/24, version 3
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer
65001
  172.16.200.10 (metric 1915392) from 172.16.200.10 (192.168.192.7)
    Origin IGP, metric 0, localpref 100, valid, external, best
    Community: 7
ringo#
ringo# show ip bgp 128.168.192.0
BGP routing table entry for 128.168.192.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to
EBGP peer)
Not advertised to any peer
65001
  172.16.200.10 (metric 1915392) from 172.16.200.10 (192.168.192.7)
    Origin IGP, metric 0, localpref 100, valid, external, best
    Community: 8 no-export
ringo#

```

使用 **set comm-list delete** 命令可以清除在接收或者发送方向的路由更新数据包的团体属性值。语法如下：

```
set comm-list [{standard | extended community list}] delete
```

要了解不同的 **set communities** 命令的更多范例以及它们是如何在 BGP 中工作的，参看第 7~9 章关于配置 BGP 的部分。

和 BGP 有关的另外一个特性就是可以设置衰减，因为它需要 BGP 网络经过很长的时间才能收敛，一个不稳定的路由或者“路由波动”对大型的 BGP 网络有很显著和决定性的影响。如果一条路由失效了，就会通过 BGP 请求包给所有的对等体发送一条 **WITHDRAWN** 的信息，来通知它们从路由表中将这条路由清除。自治系统中的一条不稳定的路由会导致经常给其他的自治系统发送加入路由或者取消路由的消息。在一个自治系统中成百上千甚至成千上万的路由器环境里，这种影响会成倍增长，对 BGP 的影响是非常大的。

衰减允许路由器将路由区分为行为好或者行为不好的路由。很显然，一个行为好的路由在很长的一段时间内应该是很稳定的。从另一个角度来说，一个行为不好的路由可能是一个不稳定的路由或者是一条波动的路由。当使用 BGP 路由器命令 **bgp dampening** 在 BGP 中启用了路由衰减，路由器就会启用一个历史文件记录每一条路由波动了多少次。每次路由波动，路由衰减就会给它分配一个惩罚点。每一条路由的惩罚点都会累加，当惩罚值大于一个强制的数字（叫做 *suppress value*）时，这条路由就不再宣告出去。路由会一直处于抑制状态，直到惩罚值低于 *reuse-limit* 或者 *max_suppress* 计时器超时。对路由的惩罚可以在一段时间内减少。*half-life* 是一种计时器，以分钟表示，当这个时间过去后，惩罚值会减少一半。如果到时间了，路由依旧是稳定的，那条路由的惩罚就会减少。当惩罚值低于另一个强制的数字（叫做

reuse-limit) 路由将会解除抑制，被重新宣告出去。图 2-5 演示了在路由衰减中时间和惩罚的关系。

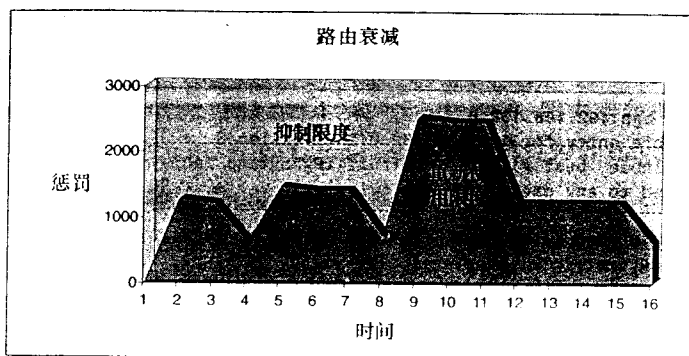


图 2-5 路由惩罚计时器的关系

注意：这种类型的路由映射是通过 BGP 的路由器命令 **bgp dampening [route-map route-map_name]**来调用的。在 **neighbor** 语句中调用的路由映射对于路由惩罚是不工作的。

在思科 IOS 软件版本 12.2 中 **set dampening** 命令如下所示：

```
set dampening {half-life_1-45 reuse_1-20000 suppress_1-20000 max_suppress_time_1-255}
```

使用 **set dampening** 命令可以影响路由器遇到不稳定的路由时如何反应。**half-life** 参数代表路由必须稳定的时间（以分钟计），在这个时间过后，惩罚值会减半。默认的时间是 15min，有效的值范围为 1~45min。

reuse 参数允许用户标记一个点，或者是一个重新使用的点，在这个点上允许路由被通告出去。当惩罚值低于重新使用的点时，路由会被解除抑制，并且重新通告出去。默认的值是 750。有效的范围是 1~20 000。

当惩罚值超过 **suppress** 参数后，路由被抑制并且不再通告出去。有效的范围是 1 到 20 000，默认的值是 2000。

max_suppress_time 是以分钟表示的一个值，它指定在衰减特性中路由最大被抑制多长时间。默认的值是 **half-life** 时间的 4 倍，或者说是 60min。有效的范围是 1~255min。

当一路由前缀被取消时，BGP 认为被取消的前缀是路由波动，于是增加 1000 个惩罚点。当 BGP 收到属性变化的前缀时，惩罚值增加 500 点。

在图 2-6 中，路由器 ringo 正在通过 BGP 给 turkey_creek 路由器宣告 129.168.192.0/24 的路由。

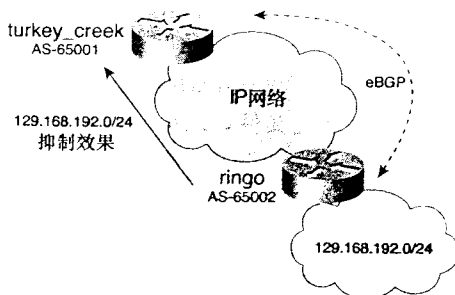


图 2-6 路由波动

turkey_creek 路由器启用了路由衰减，使用一个路由映射给 129.168.192.0/24 的路由设定了衰减。使用 **show ip bgp dampened-paths** 命令和 **show ip bgp a.b.c.d** 命令可以查看路由衰减是否发生并且查看当前的惩罚计数。注意和衰减有关的信息直到路由实际发生了波动才会出现。范例 2-18 显示了在 **turkey_creek** 路由器上的路由 129.168.192.0/24 的惩罚和衰减情况。

范例 2-18 验证波动情况

```
turkey_creek# show ip bgp dampened-paths
BGP table version is 9, local router ID is 192.168.192.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          From          Reuse      Path
*d 129.168.192.0/24 172.16.100.10    00:38:00 65002 i
turkey_creek#
turkey_creek# show ip bgp
BGP table version is 9, local router ID is 192.168.192.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
*> 128.168.192.0/24 0.0.0.0           0           32768 i
*d 129.168.192.0/24 172.16.100.10     0           0 65002 i
*> 192.168.192.0    0.0.0.0           0           32768 i
turkey_creek#
turkey_creek# show ip bgp 129.168.192.0
BGP routing table entry for 129.168.192.0/24, version 9
Paths: (1 available, no best path)
  Not advertised to any peer
  65002, (suppressed due to dampening)
    172.16.100.10 (metric 2323456) from 172.16.100.10 (172.16.100.10)
      Origin IGP, metric 0, localpref 100, valid, external, ref 2
      Dampinfo: penalty 3717, flapped 4 times in 00:04:36, reuse in 00:37:50
turkey_creek#
```

范例 2-19 列出了先前的范例中的 BGP 配置和 **turkey_creek** 路由器的相关路由映射。

范例 2-19 turkey_creek 路由器的配置

```
hostname turkey_creek
!
<<<text omitted>>>
!
router bgp 65001
 no synchronization
 bgp dampening route-map set_dampening      ←Dampening enabled with route-map
 network 128.168.192.0 mask 255.255.255.0
 network 192.168.192.0
 neighbor 172.16.100.10 remote-as 65002
 neighbor 172.16.100.10 ebgp-multihop 10
 neighbor 172.16.100.10 update-source Loopback20
!
access-list 11 permit 129.168.192.0 0.0.0.255
!
route-map set_dampening permit 100
 match ip address 11                        ←Match network 129.168.192.0/24
 set dampening 20 1000 2000 80             ←Set dampening parameters
```

关于路由波动的更多信息，参看 BGP 第 7~9 章。

也可以在 BGP 中使用路由映射来设置公认自决属性：本地优先（LOCAL_PREF）属性。本地优先属性是一个 0~4 294 967 295 的值，这个数值越高，路由就越优。默认的本地优先值是 100。表 2-8 列出了设置本地优先属性的语法。

表 2-8 在思科 IOS 软件版本 12.2 中的 set local-preference 命令

命令	描述
set local-preference {0-4294967295}	使用 set local-preference 命令来设置一条路由的本地优先值。有效的范围是 0~4 294 967 295。默认的值是 100

可以使用路由映射设置的另外一个 BGP 属性是公认必遵过渡属性：起源（ORIGIN）属性。起源属性是一个公认必遵属性。顾名思义，起源属性指定了路由的起点，它和这条路由起始的自治系统有关。BGP 支持 3 种不同类型的起点：

- **IGP (i)** ——网络层可达信息（NLRI）在起始的自治系统内部。这是一个远程 IGP 系统。路由起始于网络命令。
- **EGP (e)** ——NLRI 是通过 EGP 学习到的。这是一个本地的 EGP 系统。路由是从 EGP 进行重分发的。
- **Incomplete (?)** ——NLRI 是通过其他方法学习到的。路由是从 IGP 或者静态路由进行重分发的。

表 2-9 列出了设置原点的语法。

表 2-9 在思科 IOS 软件版本 12.2 中的 set origin 命令

命令	描述
set origin {igp egp [as_number] incomplete}	使用 set origin 命令可以设置一条或者若干路由的起源属性。有效的原点类型为：IGP、EGP 和 incomplete

我们在这里要讨论的最后一个 BGP 特定的 set 命令是 set weight 命令。权重（WEIGHT）属性是一个思科专有属性，用于衡量一条路由的优越性。权重属性是作用在路由器本地上的，不会在路由器之间交换，因此它只在入站方向的路由映射上有效。使用权重属性可以影响从多个服务提供商到一个中心站点的路由。就像本地优先，给路由分配一个更高的权重会使得它更优。权重属性也优于其他的 BGP 属性。关于 BGP 的更多信息，参看第 7~9 章。表 2-10 列出了设置权重属性的语法。

表 2-10 在思科 IOS 软件版本 12.2 中的 set weight 命令

命令	描述
set weight {0-65535}	使用 set weight 命令设置一条路由或者多条路由的权重，有效的权重范围是 0~65 535，一条路由的默认的权重值是 32 768

3. 范例：配置 BGP 属性

本范例使用和前面的范例中相同的网络模型，并且设置 BGP 的本地优先属性、权重属性和起源属性。图 2-7 是先前所示的同一网络。这个范例在 turkey_creek 路由器上调用了入站方向的路由映射。这个路由映射叫做 set_attributes，它将设置下面的这些属性：从自治系统 65002 中将权重设置为 1000，本地优先设置为 5000，起源设置为 EGP。在这个范例中，设置 local-preference 只是为了示范。通常，local-preference 不会在 E-BGP 对等体上使用或者生效。

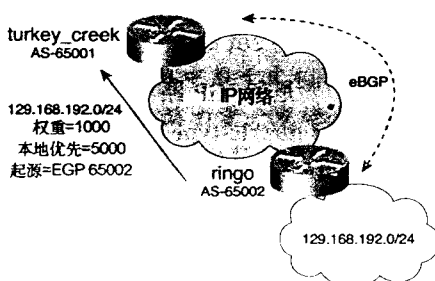


图 2-7 配置 BGP 属性

范例 2-20 列出了在 turkey_creek 路由器上完成这个任务所用的 BGP 和路由映射的配置。

范例 2-20 BGP 属性配置

```
hostname turkey_creek
!
<<<text omitted>>>
!
router bgp 65001
 no synchronization
 network 128.168.192.0 mask 255.255.255.0
 network 192.168.192.0
 neighbor 172.16.100.10 remote-as 65002
 neighbor 172.16.100.10 ebgp-multihop 10
 neighbor 172.16.100.10 update-source Loopback20
 neighbor 172.16.100.10 route-map set_attributes in ←call route-map "set_attributes"
!
route-map set_attributes permit 100
 set local-preference 5000 ←Set local-preference to 5000
 set weight 1000 ←Set weight to 1000
 set origin egp 65002 ←Set the ORIGIN to EGP in AS 65002
!
! ←*note with no match parameter all routes are
 matched from the neighbor 172.16.100.10
```

为了验证路由映射的有效性，使用 **show ip bgp** 命令，如范例 2-21 所示。

范例 2-21 验证属性

```
turkey_creek# show ip bgp
BGP table version is 4, local router ID is 192.168.192.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
*> 128.168.192.0/24 0.0.0.0              0         32768 i
*> 129.168.192.0/24 172.16.100.10        0    5000   1000 65002 e
*> 192.168.192.0   0.0.0.0              0         32768 i
turkey_creek#
turkey_creek#
turkey_creek# show ip bgp 129.168.192.0
BGP routing table entry for 129.168.192.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  65002
    172.16.100.10 (metric 2323456) from 172.16.100.10 (172.16.100.10)
      Origin EGP, metric 0, localpref 5000, weight 1000, valid, external, best,
ref 2
turkey_creek#
```


4. 配置路由选择协议/重分发特定的 set 命令

我们下面所要讨论的 **set** 命令主要是和 IGP 路由选择协议有关，并且主要用于路由重分发。在路由重分发中，**set metric**、**set metric-type** 和 **set tag** 命令都可以用来改变路由的度量值或者路由的标记。就像前面所说的那样，度量和标记也可以用来作为匹配条件，在重分发中用于进一步的路由控制。

set metric 命令的最常见用法是对目的路由协议设置度量值。例如，如果你将 EIGRP 的路由重分发到 OSPF 中，那么你可以在路由映射中使它和 **set metric** 命令相结合来设置新的 OSPF 度量值。如果你正在将路由重分发到 IGRP 或 EIGRP 中，那么你输入的度量值应当是组合度量值。这和设置默认的度量值或者在重分发中不带路由映射的度量值稍有不同，在这儿要设置所有的 5 个子度量。**set metric** 命令的另外一个用法是设置 BGP 的可选非过渡属性 MULTI_EXIT_DISC（多出口鉴别器）。在思科 IOS 软件版本 12.2 中 **set metric** 命令的语法如下：

```
set metric {[+/-<0-4294967295>]|1-4294967295}
```

+和-关键字允许用户增加或者减少当前的度量值。例如，为了给度量值增加 10，这个命令应当是 **set metric+10**。为了给 EIGRP 设置组合度量值，这个命令是 **set metric 4295**。关于 IGP 路由协议的度量值的更多信息，参考《CCIE 实验指南（第 1 卷）》。你可以在本书的第 7～9 章了解到关于 BGP 多出口鉴别器属性的更多信息。

set metric-type 命令的用法是相当有限的，它主要用于 BGP、OSPF 和 IS-IS。可以使用它来设置 IS-IS 的外部和内部度量值以及 OSPF 类型 1 和类型 2 的外部度量值。**set metric-type** 命令也可以在 BGP 中使用 IGP 的度量值来作为 BGP 的多出口鉴别器值。

在思科 IOS 软件版本 12.2 中 **set metric-type** 命令的语法如下：

```
set metric-type [internal|external|type-1|type-2]
```

- **external**——IS-IS 外部度量。
- **internal**——使用 IGP 的度量值作为 BGP 的多出口鉴别器。也可以用于设置 IS-IS 的内部度量值。
- **type-1**——用来匹配 OSPF 类型 1 的度量值。
- **type-2**——用来匹配 OSPF 外部类型 2 的度量值。

本节中我们要讨论的最后一个 **set** 命令是 **set tag** 命令。**set tag** 命令允许用户设置一条路由的管理标记。对于 IGP，标记值通常是通过路由映射和 **set tag** 命令设置的。在 BGP 中，当你将 BGP 重分发到 IGP 中时，BGP 的 ASN 会自动地置为标记值。BGP 这样做的目的是为了通过 IGP 的域时还保留 AS 路径属性。对于 IGP，标记是一个管理值，某些路由选择协议可以在路由更新数据包中携带这个值。标记值对路由的决策没有影响。相反，它主要适用于标记路由或者跟踪 BGP 的 AS 路径。标记值也可以在重分发时起作用。当在 BGP 的 **table-map** 命令中使用 **automatic-tag** 命令时，标记值包括 ASN 和原点。在思科 IOS 软件版本 12.2 中操作标记值的语法如下：

```
set {tag [ 0-4294967295]|automatic-tag}
```

使用 **set tag value** 命令设置标记值。在将 IGP 路由重分发到 BGP 中时，使用 **set**

automatic-tag 命令可以将标记值转换成 AS 路径属性。

注意：也可以使用标记值在互连网络中用于文档管理的目的。例如，你有一个 OSPF 的域，而我们需要将 RIP 路由和 EIGRP 路由重分发进来，你可能想将所有来自 EIGRP 的路由标记为 100，而将所有来自 IGRP 的路由标记为 110。当查看 OSPF 的数据库时，就会很容易地断定某条路由的起源。这对路由重分发的故障排查提供了一个非常有用的文档工具。

在 RIPv2、OSPF、集成 IS-IS、EIGRP、BGP 和 CLNS 这些协议中支持标记。IGRP 和 RIPv1 不支持标记。要想查看标记，在 EIGRP 和 OSPF 中分别使用 **show ip eigrp topology ip_address subnet_mask** 命令和 **show ip ospf database** 命令。也可以在其他路由选择协议中使用扩展的 **show ip route** 命令 **show ip route ip_address** 来查看标记。

5. 范例：设置路由标记和度量类型

在图 2-8 的互连网络中，路由器 **turkey_creek**、**earp**、**holliday** 和 **ringo** 正在运行 EIGRP。**ringo** 路由器和 **turkey_creek** 路由器还有一个 BGP 的对等体关系，它和 **clanton** 路由器之间运行的是 OSPF 协议。

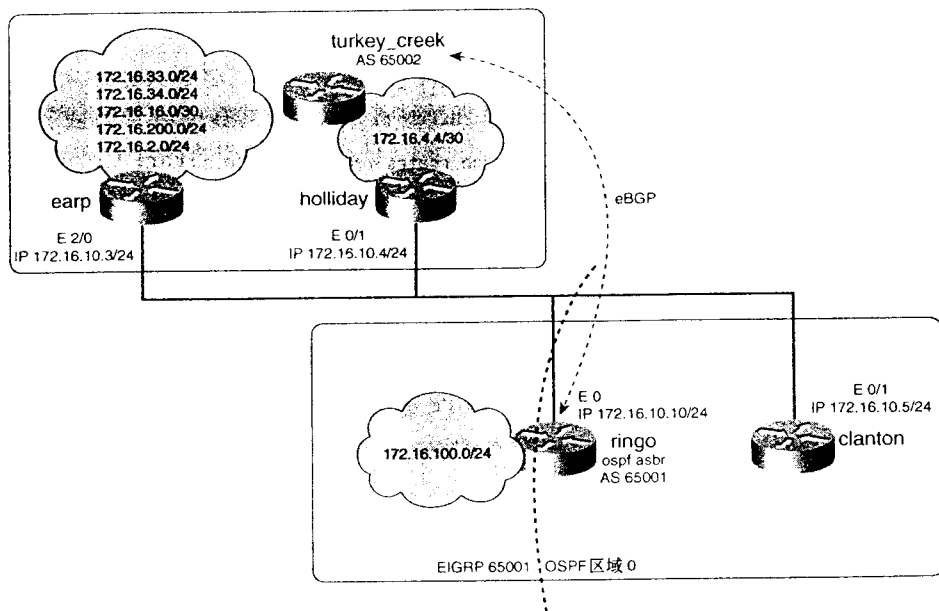


图 2-8 路由标记和度量值的设置

为了演示路由标记和度量值的设置，下面的范例在 **ringo** 路由器上写了一个路由映射。这个路由映射用在 **ringo** 路由器上主要是将 EIGRP 路由重分发到 OSPF 里。这个路由映射首先会将来自 **earp** 路由器 (172.16.10.3) 的路由打上标记 3。接下来，这个路由映射会将所有的其他路由打上标记 500，并且将这些路由标记为 OSPF 外部类型 1 的路由。范例 2-22 列出了在 **ringo** 路由器上完成这个任务的配置。

范例 2-22 ringo 路由器的配置

```

hostname ringo
!
<<<text omitted>>>
!
router eigrp 65001
 redistribute bgp 65002
 network 172.16.0.0
 network 192.168.10.0
 default-metric 10000 1000 254 1 1500
 no auto-summary
 eigrp log-neighbor-changes
!
router ospf 7
 log-adjacency-changes
 redistribute eigrp 65001 subnets route-map set_tag3 ←Redistribute and call route-map
 redistribute bgp 65002
 network 172.16.10.10 0.0.0.0 area 0
 default-metric 10
!
router bgp 65002
 no synchronization
 bgp log-neighbor-changes
 neighbor 172.16.200.10 remote-as 65001
 neighbor 172.16.200.10 ebgp-multihop 10
 neighbor 172.16.200.10 update-source Loopback20
!
access-list 5 permit 172.16.10.3 ←Match routes from 172.16.10.3
access-list 50 permit any ←Match all routes
!
route-map set_tag3 permit 100
 match ip route-source 5 ←Match routes from 172.16.10.3
 set tag 3 ←Set the TAG value to 3
!
route-map set_tag3 permit 200
 match ip address 50 ←Match all other routes
 set metric-type type-1 ←Set the OSPF metric to External Type-1
 set tag 500 ←Set the TAG value to 500
!

```

通过观察 ringo 路由器的路由表和它的 OSPF 数据库，可以看到路由映射的执行效果，如范例 2-23 所示。

范例 2-23 ringo 路由器上路由映射的效果

```

ringo# 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
B    192.168.192.0/24 [20/0] via 172.16.200.10, 01:07:04
     172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
D    172.16.200.0/24 [90/1915392] via 172.16.10.3, 01:07:08, Ethernet0

```

(待续)

```

D    172.16.33.0/24 [90/1812992] via 172.16.10.3, 01:07:08, Ethernet0
D    172.16.34.0/24 [90/1812992] via 172.16.10.3, 01:07:08, Ethernet0
D    172.16.16.4/30 [90/2195456] via 172.16.10.4, 01:07:08, Ethernet0
D    172.16.16.0/30 [90/1787392] via 172.16.10.3, 01:07:08, Ethernet0
C    172.16.10.0/24 is directly connected, Ethernet0
D    172.16.2.0/24 [90/284160] via 172.16.10.3, 01:07:09, Ethernet0
C    172.16.100.0/24 is directly connected, Loopback20
ringo#
ringo# show ip ospf database
      OSPF Router with ID (172.16.100.10) (Process ID 7)
      Router Link States (Area 0)
Link ID      ADV Router    Age      Seq#          Checksum Link count
172.16.10.5  172.16.10.5    1151     0x80000015   0x4E2    1
172.16.100.10 172.16.100.10 1875     0x80000003   0xC969   1
      Net Link States (Area 0)
Link ID      ADV Router    Age      Seq#          Checksum
172.16.10.5  172.16.10.5    1151     0x80000003   0x1693
      Type-5 AS External Link States
Link ID      ADV Router    Age      Seq#          Checksum Tag
172.16.2.0   172.16.100.10 1875     0x80000002   0x8E2E   3
172.16.15.0  172.16.100.10 1875     0x80000002   0xE1CF   3
172.16.16.4  172.16.100.10 1875     0x80000002   0x4AF0   500
172.16.33.0  172.16.100.10 1875     0x80000002   0x3865   3
172.16.34.0  172.16.100.10 1875     0x80000002   0x2D6F   3
172.16.100.0 172.16.100.10 1875     0x80000002   0xE403   500
172.16.200.0 172.16.100.10 1875     0x80000002   0x4F1    3
192.168.192.0 172.16.100.10 1876     0x80000002   0x4A22   65001
ringo#

```

注意在 OSPF 数据库的末尾是 BGP 路由 192.168.192.0/24。这条路由有一个标记为 65001，这是因为 BGP 在重分发到支持标记的 IGP 时，会试图保留 AS 路径属性。BGP 使用的标记值等同于它的自治系统的 ID 号。

也可以看到 clanton 路由器上路由映射的效果，范例 2-24 列出了 clanton 路由器上的路由表，着重显示了不同的 OSPF 路由类型。注意 172.16.16.4/30 和 172.16.100.0/24 路由没有设置成默认的 OSPF 外部类型 2 的路由，而是外部类型 1 的路由。这是因为在 ringo 路由器上，在路由映射中使用了 **set route-type type-1** 命令。

范例 2-24 clanton 路由器的路由表

```

clanton# 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
O E2 192.168.192.0/24 [110/10] via 172.16.10.10, 01:00:14, Ethernet0/0
     172.16.0.0/16 is variably subnetted, 8 subnets, 2 masks
O E2 172.16.200.0/24 [110/10] via 172.16.10.10, 01:00:14, Ethernet0/0
O E2 172.16.33.0/24 [110/10] via 172.16.10.10, 01:00:14, Ethernet0/0
O E2 172.16.34.0/24 [110/10] via 172.16.10.10, 01:00:14, Ethernet0/0
O E1 172.16.16.4/30 [110/20] via 172.16.10.10, 01:00:14, Ethernet0/0
O E2 172.16.16.0/30 [110/10] via 172.16.10.10, 01:00:14, Ethernet0/0
C    172.16.10.0/24 is directly connected, Ethernet0/0
O E2 172.16.2.0/24 [110/10] via 172.16.10.10, 01:00:15, Ethernet0/0
O E1 172.16.100.0/24 [110/20] via 172.16.10.10, 01:00:15, Ethernet0/0
clanton#

```

2.1.2 路由映射和策略性路由

在现代互连网络中，有时路由器的转发决策需要比路由选择协议和路由表所提供的转发信息更加复杂。总体来说路由器是基于数据包的目的地址来做出转发决定。策略性路由使得网络工程师可以配置策略，使数据包通过和路由表中的下一跳路径不同的路径。本小节讨论了策略性路由的优点和配置。

策略性路由提供下面这些好处：

- **转发决策不是基于目的地址**——策略性路由允许网络工程师根据数据包的属性、源/目的 IP 地址、应用端口和数据包的长度来定义一条路径，并且按照不同的策略来转发它们。策略性路由可以设置数据包的下一跳或者数据包的默认下一跳/接口。策略性路由也可以用于将数据包路由到一个空接口，实际上就是丢弃它们。
- **服务质量 (QoS)**——路由映射和策略性路由可以提供服务质量，通过允许用户设置服务类型 (ToS) 的值和 IP 报头中的 IP 优先级值来实现。服务质量配置是在边界路由器上执行的。这可以通过在核心路由器上避免额外的配置来提高性能。
- **通过使用其他路径节省费用**——IP 流量可以使用策略性路由操作，例如，大量的突发文件传输可以通过低费用、低带宽的链路发送，而对时间比较敏感、用户的交互式流量可以通过高费用和高速的链路发送。
- **基于流量特性的多条不等长路径的负载分担**——策略性路由可以基于流量特性而不是路径的费用值在多条不等长的路径上实现负载分担。

假设策略性路由已经启用并且在路由器和接口上配置，策略性路由以下列方式工作：

- 第1步** 在一个策略性路由启用的接口上收到的所有数据包都认为是策略性路由的。在那个接口上收到的每一个数据包都会通过相关的路由映射检验。
- 第2步** 路由映射会调用 **match** 命令，如果所有的 **match** 条件都匹配了，路由映射被标注为 **permit** 或者 **deny**，那么以后的路由映射实例就不再执行。如果没有配置 **match** 语句，那么路由映射和任何 **set** 命令就会对所有数据包生效。
- 第3步** 如果路由映射有一个 **permit** 语句，所有的 **set** 命令都会生效，数据包会按照新的策略进行转发。可以在一个路由映射实例中使用多个 **set** 命令。表 2-7 列出了特定于策略性路由的 **set** 命令。如果你使用多个 **set** 命令并彼此结合，那么它们的应用顺序如下所示：

```
set ip {precedence [value_0-7 | name] | tos [value_0-8 | name]}
set ip next-hop ip_address
set interface interface_name
set ip default next-hop ip_address
set default interface interface_name
```

这些命令的详细内容会在后续部分详细讨论。

- 第4步** 如果路由映射有一个 **deny** 语句，使用正常的转发行为，也就是路由/转发表中所指定的出口。**set** 语句不会应用于这个数据包。
- 第5步** 在所有路由映射实例的尾部，有一个隐含的路由映射拒绝所有的数据包。如果数据包在先前的路由映射实例中没有找到匹配，就会和这个隐含的拒绝路由映

射实例相匹配。当这种情况发生时，数据包会被路由器按照正常的路由表转发出去。

注意：策略性路由只在入站方向的数据包上工作。因此，它只能绑定在入站流量上或者接收这个流量实现策略性路由的接口上。为了策略性路由本地流量，必须启用本地策略路由。

一、范例：策略性路由

这一部分内容讨论用户如何使用策略性路由来控制互连网络中的流量。在图 2-9 所示的网络模型中，在 tombstone 路由器上配置了策略性路由来控制来自 ringo 和 curly_bill 路由器的流量。这个策略声明所有来自 ringo 路由器的 IP 流量必须转发到 holliday 路由器，而所有来自 curly_bill 路由器的流量必须转发到 earp 路由器，所有其他的 IP 流量将通过正常的路由过程进行转发。

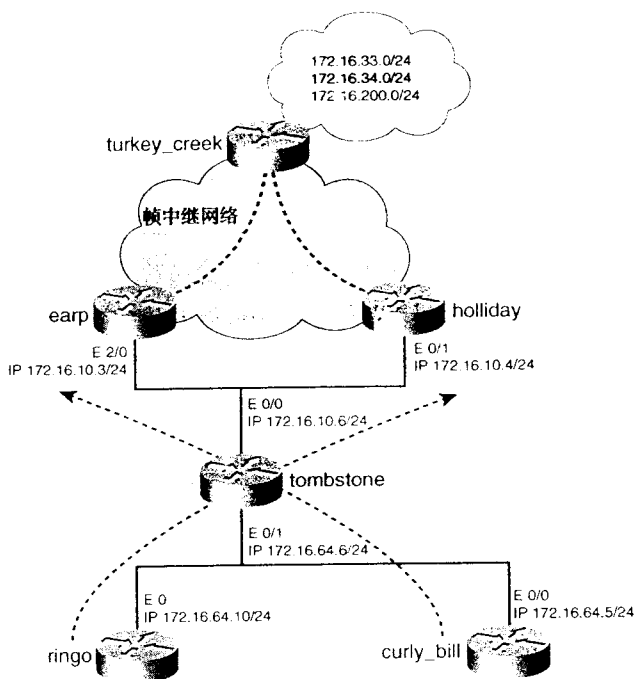


图 2-9 策略性路由

为了控制来自 ringo 和 curly_bill 路由器的流量，这个范例在 tombstone 路由器上使用了策略性路由和路由映射。策略性路由将在 tombstone 路由器的 E0/1 接口上启用。这是一个入站接口，或者说这个接口将接收来自 ringo 和 curly_bill 路由器的流量。在这个模型中使用的路由映射(policy_1)将有两个路由映射的实例。一个实例匹配来自 ringo 路由器(172.16.64.10)的数据包，并且将下一跳设置为 172.16.10.4，也就是 holliday 路由器。另外一个路由映射的实例将匹配来自 curly_bill 路由器(172.16.64.5)的数据包，并且将下一跳设置为 172.16.10.3，也就是 earp 路由器。

在 tombstone 路由器上的路由/转发表显示有两条路径可以到达 turkey_creek 路由器上的 172.16.33.0/24、172.16.34.0/24 和 172.16.200.0/24 网段。一条路径通过 earp 路由器，另外一

条通过 holliday 路由器。范例 2-25 列出了 tombstone 路由器的路由表。

范例 2-25 tombstone 路由器的路由表

```
tombstone# 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, 9 subnets, 2 masks
D    172.16.200.0/24 [90/40665600] via 172.16.10.3, 03:58:24, Ethernet0/0
      [90/40665600] via 172.16.10.4, 03:58:24, Ethernet0/0
D    172.16.33.0/24 [90/40563200] via 172.16.10.3, 03:58:24, Ethernet0/0
      [90/40563200] via 172.16.10.4, 03:58:24, Ethernet0/0
D    172.16.34.0/24 [90/40563200] via 172.16.10.3, 03:58:24, Ethernet0/0
      [90/40563200] via 172.16.10.4, 03:58:24, Ethernet0/0
D    172.16.16.4/30 [90/40537600] via 172.16.10.4, 03:59:03, Ethernet0/0
D    172.16.16.0/30 [90/40537600] via 172.16.10.3, 04:56:26, Ethernet0/0
C    172.16.10.0/24 is directly connected, Ethernet0/0
D    172.16.2.0/24 [90/284160] via 172.16.10.3, 03:59:03, Ethernet0/0
D    172.16.100.0/24 [90/409600] via 172.16.64.10, 03:49:42, Ethernet0/1
C    172.16.64.0/24 is directly connected, Ethernet0/1
tombstone#
```

通过在 tombstone 路由器上发出从地址 172.16.64.6 到 172.16.200.10 的扩展 **traceroute** 命令，可以看到 EIGRP 正在通过 earp 和 holliday 路由器使用负载分担，策略性路由将凌驾于这个过程之上，如范例 2-26 所示，将来自 ringo 路由器的 IP 流量发往 holliday 路由器，将来自 curly_bill 路由器的 IP 流量发往 earp 路由器。

范例 2-26 在 tombstone 路由器上的扩展 Trace

```
tombstone# traceroute
Protocol [ip]:
Target IP address: 172.16.200.10
Source address: 172.16.64.6
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]: 4
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 172.16.200.10
 1 172.16.10.4 0 msec
   172.16.10.3 0 msec
   172.16.10.4 0 msec
   172.16.10.3 0 msec
 2 172.16.16.5 8 msec
   172.16.16.1 12 msec
   172.16.16.5 8 msec
   172.16.16.1 12 msec
tombstone#
```

在 tombstone 路由器上策略性路由所需的配置在范例 2-27 中列出。

范例 2-27 在 tombstone 路由器上策略性路由的配置

```

hostname tombstone
!
interface Ethernet0/0
 ip address 172.16.10.6 255.255.255.0
!
interface Ethernet0/1
 ip address 172.16.64.6 255.255.255.0
 ip route-cache policy          ←Optional fast switching for policy routing
 ip policy route-map policy_1  ←Call route-map "policy_1" for policy routing
!
router eigrp 65031
 network 172.16.0.0
 no auto-summary
!
access-list 100 permit ip host 172.16.64.10 any      ←match packets from 172.16.64.10
access-list 101 permit ip host 172.16.64.5 any      ←match packets from 172.16.64.5
!
route-map policy_1 permit 100                       ←route-map "policy_1"
 match ip address 100                               ←call ACL 100 for match criteria
 set ip next-hop 172.16.10.4                         ←set IP next hop to holliday
!
route-map policy_1 permit 200                       ←next route map instance
 match ip address 101                               ←call ACL 101 for match criteria
 set ip next-hop 172.16.10.3                         ←set IP next hop to the earp router
!

```

为了测试新的策略，在 ringo 和 curly_bill 路由器上发出到驻留在 turkey_creek 路由器上的 IP 地址为 172.16.200.10 的 **traceroute** 命令。从 ringo 路由器发出的 **traceroute** 将显示数据包通过了 tombstone 路由器，接着是 holliday 路由器，最终是 turkey_creek 路由器。范例 2-28 演示了在 ringo 路由器上启用了策略性路由后的 **traceroute** 命令。

范例 2-28 在 ringo 路由器上执行 traceroute

```

ringo# traceroute 172.16.200.10
Type escape sequence to abort.
Tracing the route to 172.16.200.10
 1 172.16.64.6 4 msec 4 msec 4 msec
 2 172.16.10.4 8 msec 4 msec 4 msec
 3 172.16.16.5 20 msec 8 msec *
ringo#

```

为了测试对 curly_bill 路由器的新策略，在 curly_bill 路由器上发出到 IP 地址为 172.16.200.10 的 **traceroute** 命令。数据包将通过 tombstone 路由器，接着是 earp 路由器，最后是 turkey_creek 路由器。范例 2-29 演示了在 curly_bill 路由器上执行 **traceroute** 命令。

范例 2-29 在 curly_bill 路由器上执行 traceroute

```

curly_bill# traceroute 172.16.200.10
Type escape sequence to abort.
Tracing the route to 172.16.200.10
 1 172.16.64.6 4 msec 4 msec 4 msec
 2 172.16.10.3 4 msec 4 msec 0 msec
 3 172.16.16.1 12 msec 9 msec *
curly_bill#

```


注意：无论什么时候执行策略性路由，注意考虑网络上运行的应用程序和网络流量的转发路径和返回路径。在先前范例的模型中，可以在 `turkey_creek` 路由器上实施策略性路由来避免非对称路由。非对称路由指的是 IP 数据包沿一条路径转发到一个目的地，但是却遵循一条不同的路径返回，这对某些应用程序会产生问题，例如组播。

二、配置策略性路由（PBR）

可以通过下面的这些步骤配置策略性路由。取决于使用策略性路由的应用程序，某些步骤可以省略。

第1步 定义并配置策略所需要路由映射。通过 `route-map` 命令来实现，正如前面所述。

第2步 定义并配置路由映射将使用的 `match` 语句。最常用的 `match` 语句如下：

```
match ip address [ access-list number]
```

`match ip address` 用来调用一个标准的、扩展的或者扩展范围的访问控制列表。

```
match length [min_packet_length_0-2147483647] [max_packet_length_0-2147483647]
```

`match length` 用于匹配三层数据包的长度，以字节计，包括所有相关的头和尾部信息。可以输入最小和最大的数据包的长度。使用 `match length` 命令基于数据包的尺寸来策略性路由流量。可以利用这一点将大数据包或者小数据包路由到网络中的特定区域。

第3步 使用 `set` 命令配置和定义新的路由策略。可以使用多个 `set` 命令，如果使用了多个命令，它们按照下面的顺序执行：

```
set ip {precedence [value_0-7 | name] | tos [value_0-8 | name]}
set ip next-hop ip_address
set interface interface_name
set ip default next-hop ip_address
set default interface interface_name
Set ip precedence {[1-7]}[routine|critical|flash|flash-
override|immediate|internet|network|priority]}
```

通过设置优先级，可以操作在 IP 报头中 ToS 字段的 8 位中的头 3 位，也就是 0～2 位。早期版本的 TCP/IP 说明这个字段是不可用的，是被路由器忽略的，除了某些路由选择协议之外。这在过去可能是事实，但是，随着 Voice over IP 的出现和更新的服务质量特性，Precedence 字段具有了新的活力和意义，IP 优先级成为了在接口出现拥塞时进行调节的一个重要因素。默认情况下，思科路由器在数据包到达路由器时并不操作 IP 报头中的优先级值，而保持它原有的设置。当加权公平队列（WFQ）启用和优先级位设置后，数据包就会按照优先级值的顺序进行传输。优先级的值越高，它在队列中传输的可能性就越大。如果要使路由器对优先级的作用生效的话，这个链路必须是拥塞的，而且必须启用队列。否则，数据包会按照先进先出（FIFO）的顺序进行传输。当设置优先级时，可以对优先级使用数值或者名字来代表。优先级的设置应当使下游的 IP 设备能够利用你对它的设置。表 2-11 列出了对于 `set precedence` 命令的有效名字值。

关于 **set precedence** 命令的详细信息，参看第 5 章和第 6 章。

表 2-11 在思科 IOS 软件版本 12.2 中的 **set precedence** 命令

命令	功能
routine	设置 routine precedence (值= 0)
priority	设置 priority precedence (值= 1)
immediate	设置 immediate precedence (值= 2)
flash	设置 Flash precedence (值= 3)
flash-override	设置 Flash override precedence (值= 4)
critical	设置 critical precedence (值= 5)
internet	设置 internetwork control precedence (值= 6)
network	设置 network control precedence (值= 7)

注意：为了使得路由器的队列机制对优先级位生效，下列两个条件必须满足：

- 发送的链路必须是拥塞的。
- 发送的链路必须配置加权公平队列或者是加权随机早期检测 (WRED)。

```
Set ip tos {[1-15]||[normal|min-delay|max-throughput|max-reliability|min-monetarycost|priority]}
```

set ip tos 命令允许用户设置 IP 报头中 8 位的 ToS 字段中的第 3~6 位。ToS 字段是由 4 位组成的。这些位如下所述：

- **D bit (bit 3)** ——正常= off，低延迟= on
- **T bit (bit 4)** ——正常= off，高吞吐= on
- **R bit (bit 5)** ——正常= off，高可靠性= on
- **C bit (bit 6)** ——在思科的路由器上没有使用。RFC 1349 称它为*最小的费用值*。某些 TCP/IP 实现忽略这一位或者对它的处理方法不一样。第 7 位在 ToS 字段中当前没有使用并且设置为 0。如果所有的 4 位都为 0，那么代表的是正常的服务。

表 2-12 列出了根据协议类型推荐的 ToS 设置。

表 2-12 依据协议推荐的 ToS 值

协议	最小延迟	最大吞吐量	最大可靠性	最低花费值
Telnet/Rlogin	1	0	0	0
HTTP	1	0	0	0
FTP 控制	1	0	0	0
FTP 数据	0	1	0	0
任何批量数据	0	1	0	0
TFTP	1	0	0	0
SMTP 命令	1	0	0	0
SMTP 数据阶段	0	1	0	0
DNS UDP 质询	1	0	0	0
DNS TCP 质询	0	0	0	0
DNS 域 xfer	0	1	0	0
ICMP	0	0	0	0

续表

协议	最小延迟	最大吞吐量	最大可靠性	最低花费值
IGP	0	0	1	0
SNMP	0	0	1	0
BOOTP	0	0	0	0
NNTP	0	0	0	1

注意：只有流量处在加权公平队列、WRED 或者加权轮循队列（WRR）中时思科 IOS 软件才考虑 ToS 字段中的优先级位。优先级位在策略性路由、优先级队列（PQ）、定制队列（CQ）或者基于类别的加权公平队列（CBWFQ）中是不考虑的。

```
set ip next-hop {ip_address}
```

使用这个命令来设置数据包要被转发的下一跳路由器的 IP 地址。IP 地址必须是邻接的路由器。

```
set interface {interface_name}
```

使用这个命令来设置被匹配的数据包的输出接口。

```
set ip default next-hop {ip_address}
```

这个命令的使用类似于 **ip next-hop** 命令。如果在路由表中没有显式的路由到达目的网段，它指定数据包应该被转发到哪一个 IP 地址。把这条命令视为由于策略性路由的默认路由。下一跳的地址必须是邻接的路由器。

```
set default interface {interface_name}
```

这个命令的功能非常类似于 **ip default next-hop** 命令。如果在一个点对点的链路上没有显式的路由到达目的网段，它指定被匹配的数据包从哪个接口转发出去。

注意：**set ip next-hop** 和 **set ip default next-hop** 命令很类似，但功能是不一样的。**set ip next-hop** 让路由器先使用策略性路由，然后使用路由表。**set ip default next-hop** 命令让路由器先使用路由表，然后再使用策略性路由到达默认的下一跳。

第 4 步 （可选）定义和配置新路由策略所使用的任何访问控制列表。例如，使用扩展访问控制列表，你可以基于流量的类型（例如，其他流量一条路，FTP 的流量另外一条路）来定制策略转发流量。也可以使用访问控制列表控制来自特定地址的流量的转发。当使用标准的访问控制列表时，策略性路由会将数据包中的源 IP 地址和访问控制列表进行对比。

第 5 步 在入站的接口上配置策略性路由。为了对一个接口配置策略性路由，使用下面的接口命令：

```
router(config-if)# ip policy route-map route-map_name
```

第 6 步 （可选）对策略性路由启用快速交换。在思科 IOS 软件版本 12.0 中，策略性路由可以实现快速交换。在思科 IOS 软件版本 12.0 之前，策略性路由只能实现进程交换。在进程交换的环境中，转发速率大约是每秒 1000~10 000 个数据包。对许多应用程序来说，这个速度不够快。可以使用下面的接口命令来对策略性

路由启用快速交换：

```
router(config-if)# ip route-cache policy
```

在配置快速交换策略性路由之前，必须先配置策略性路由。快速交换的策略性路由不支持 **set ip default next-hop** 和 **set default interface** 命令。**set interface** 命令在点对点的链路上支持，或者在有一条静态路由的缓存表项指向 **set interface** 命令所指定的接口时支持。

第 7 步 （可选）配置本地策略性路由。通过路由器产生的数据包是不会进行策略性路由的。如果你想对路由器本地产生的流量进行策略性路由的话，你必须启用它。为了实现本地的策略性路由，使用下面的全局配置命令：

```
router(config)# ip local policy route-map route-map_name
```

三、范例：配置策略性路由和设置 ToS

在这一部分中，可以在范例的策略性路由中应用这些概念。对于图 2-10 所示的网络，建立一个策略路由，它将远程登录的流量转发到 **earp** 路由器，也就是 172.16.10.3，而将 ToS 位设置为最小的延迟。所有其他的 IP 流量将被转发到 **holliday** 路由器，也就是 172.16.10.4。

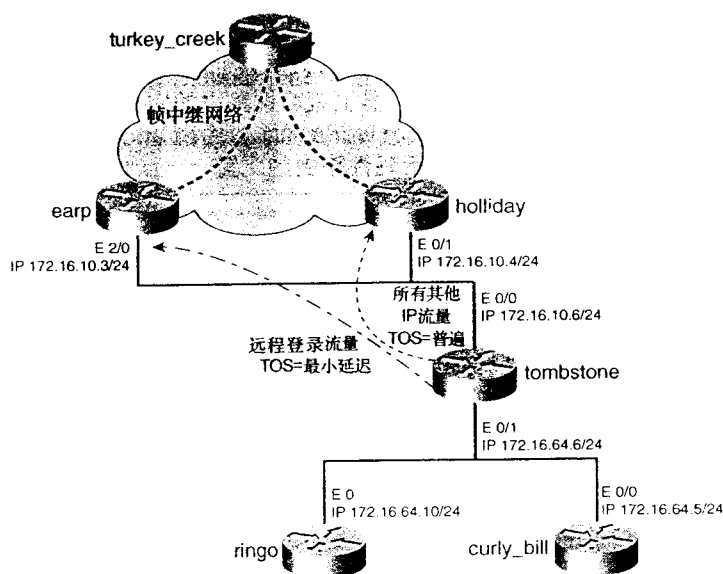


图 2-10 策略性路由

我们看一下配置策略性路由的多个步骤，第 1~3 步要求用户首先使用必需的 **match** 和 **set** 命令来配置路由映射。这个路由映射将调用一个访问控制列表来匹配远程登录的流量，而 **set** 命令将把 IP 的下一跳设置为 **earp** 路由器的 IP 地址。表 2-12 指定了远程登录的流量应当将 ToS 设置为 **min-delay**，因此，路由映射将把远程登录流量中的 ToS 值设置为 **min-delay**。另外一个路由映射的实例将用来匹配所有其他的流量，并将流量转发到 **holliday** 路由器。因为这个路由映射的实例将匹配所有其他的流量，所以就没有必要包括 **match** 命令。范例 2-30 列出了在 **tombstone** 路由器上完成这个任务的路由映射配置。

范例 2-30 tombstone 路由器上路由映射的配置

```

route-map policy_2 permit 100
  match ip address 101          ←Call access-list 101
  set ip next-hop 172.16.10.3    ←Set the next hop to 172.16.10.3/earp
  set ip tos min-delay          ←Set the TOS to min-delay
!
route-map policy_2 permit 200
  set ip next-hop 172.16.10.4    ←Match all routes and set the next hop
                                ←to 172.16.10.4/holliday

```

现在必须配置路由映射所需的任何访问控制列表。在这个范例里，配置一个简单的访问控制列表匹配来自任何 IP 地址的远程登录流量。所使用的访问控制列表类似于下面所示：

```
access-list 101 permit tcp any any eq telnet
```

没有必要配置一个访问控制列表来匹配所有的常规流量。正如前面所说的那样，在路由映射中缺少 **match** 语句，例如在路由映射的第二个实例中，它将匹配所有的数据包或者路由。

最后两步要求用户将策略性路由绑定到接口上，并且启用快速交换的策略性路由。这可以通过接口命令 **ip policy route-map** 和 **ip route-cache policy** 来完成。在这个模型中，可以在 tombstone 路由器的 E0/1 接口上启用策略性路由。随着策略性路由在 E0/1 接口上的启用，所有远程登录的流量将会被转发到 earp 路由器，而所有其他的 IP 流量将会被转发到 holliday 路由器。范例 2-31 列出了在 tombstone 路由器上策略性路由的完整配置。

范例 2-31 在 tombstone 路由器上策略性路由的配置

```

hostname tombstone
!
interface Ethernet0/0
  ip address 172.16.10.6 255.255.255.0
!
interface Ethernet0/1
  ip address 172.16.64.6 255.255.255.0
  ip route-cache policy          ←enable PBR fast-switching
  ip policy route-map policy_2   ←Call route-map "policy_2" for PBR
!
router eigrp 65001
  network 172.16.0.0
  no auto-summary
  no eigrp log-neighbor-changes
!
access-list 101 permit tcp any any eq telnet ←Match Telnet traffic
!
priority-list 1 protocol ip high          ←Priority queuing for TOS enforcement
priority-list 1 default low
!
route-map policy_2 permit 100
  match ip address 101          ←call access-list 101 and match Telnet
  set ip next-hop 172.16.10.3    ←Set the next hop to earp/172.16.10.3
  set ip tos min-delay          ←Set TOS min-delay bit
!
route-map policy_2 permit 200          ←Match all other traffic
  set ip next-hop 172.16.10.4    ←Set the next hop to holliday/172.16.10.4
!

```

在这个模型中，因为我们设置了 ToS 的值，所以需要在输出接口上配置 WRED 或者加权公平队列。加权公平队列不是以太接口上的默认队列机制，它是在 2.048 Mbit/s 或者更小带宽的串行接口上的默认队列机制。这部分的配置没有在本范例中出现。关于配置 WRED 和加权公平队列的更多信息，参看第 5 章和第 6 章。

2.1.3 路由映射的“Big Show”

《CCIE 实验指南（第 1 卷）》介绍了什么叫做 *Big Show* 和 *Big D*。这些术语之所以这么叫，是因为我们选择性地介绍了一些我们认为非常有用的 **show** 和 **debug** 命令。

在路由映射上 Big Show 和 Big D 命令的用法是非常有限的。测试策略性路由和路由映射功能的最好方法就是通过查看路由表和使用 **traceroute** 命令来实际了解它们是如何执行的。思科所提供的 **show** 命令在显示路由映射绑定在哪里，它所运行的逻辑顺序方面是非常有帮助的。我们这里所要讨论的 Big Show 命令如下：

- **show route-map**
- **show ip policy**
- **show ip cache policy**

show route-map 命令允许用户定义路由映射的逻辑顺序和执行过程。如果启用了策略性路由，那么这个命令也会显示匹配的次數和策略性路由所通过的字节数。我们还是使用先前的网络模型，范例 2-32 演示了在 tombstone 路由器上的 **show route-map** 命令

范例 2-32 在 tombstone 路由器上的 show route-map 命令

```
tombstone# show route-map
route-map policy_2, permit, sequence 100
  Match clauses:
    ip address (access-lists): 101
  Set clauses:
    ip next-hop 172.16.10.3
    ip tos min-delay
  Policy routing matches: 264 packets, 15852 bytes
route-map policy_2, permit, sequence 200
  Match clauses:
  Set clauses:
    ip next-hop 172.16.10.4
  Policy routing matches: 60 packets, 4478 bytes
route-map policy_1, permit, sequence 100
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 172.16.10.4
    ip tos max-throughput
  Policy routing matches: 85 packets, 6880 bytes
route-map policy_1, permit, sequence 200
  Match clauses:
    ip address (access-lists): 101
  Set clauses:
    ip next-hop 172.16.10.3
  Policy routing matches: 43 packets, 3318 bytes
tombstone#
```

使用 **show ip policy** 命令来验证在哪些接口上启用了策略性路由，以及它们当前正在用

哪个路由映射来实现策略性路由。范例 2-33 演示了在 tombstone 路由器上的 **show ip policy** 命令。

范例 2-33 在 tombstone 路由器上的 show ip policy 命令

```
tombstone# show ip policy
Interface      Route map
Ethernet0/1    policy_2
```

可以使用 **show ip cache policy** 命令来验证对于策略性路由快速交换是否已经启用。这个命令显示策略的类型、使用的路由映射和缓存表项的生存时间。如果策略是下一跳的策略，那么也会显示下一跳。范例 2-34 列出了在 tombstone 路由器上使用 **show ip cache policy** 命令的输出。

范例 2-34 在 tombstone 路由器上的 show ip cache policy 命令

```
tombstone# show ip cache policy
Total adds 4, total deletes 2
Type Route map/sequence      Age          Interface      Next Hop
NH policy_2/100              00:38:27    Ethernet0/0    172.16.10.3
NH policy_2/200              00:43:56    Ethernet0/0    172.16.10.4
tombstone#
```

2.2 实验 3：配置复杂的路由映射和使用标记

2.2.1 练习场景

路由映射是可以在路由器上配置的最强大的特性之一。我们可以在重分发、策略性路由、BGP 和许多其他的场景下使用它们。这个实验给你一个练习配置复杂路由映射的机会，把它们用于重分发，接下来练习设置和使用路由标记。

2.2.2 实验练习

GameNetworks.com 是一个致力于给 console 游戏提供广域网和局域网连接的快速发展的公司。GameNetworks.com 允许用户通过它们的专有网络在线玩一些最新和最好的 console 游戏。GameNetworks.com 在 Wisconsin 和 California 有两个新的地点。你的任务就是使用下面严格的设计指导配置一个 IP 网络。

- 按照图 2-11 所示配置 GameNetworks.com 的 IP 网络。使用 EIGRP 作为路由选择协议，2002 作为 wisconsin_x、unreal 和 halo 路由器的自治系统 ID。在 california_x 路由器和 gamenet 路由器上使用 EIGRP 作为路由选择协议，这些路由器的自治系统 ID 为 65001。
- 将 gamenet 和 wisconsin_x 路由器加入 EIGRP 的路由域。
- 按照图 2-11 所示配置帧中继的网络。

- 按照图 2-11 所示配置所有的 IP 地址。
- 使用“实验目的”部分查看配置细节。

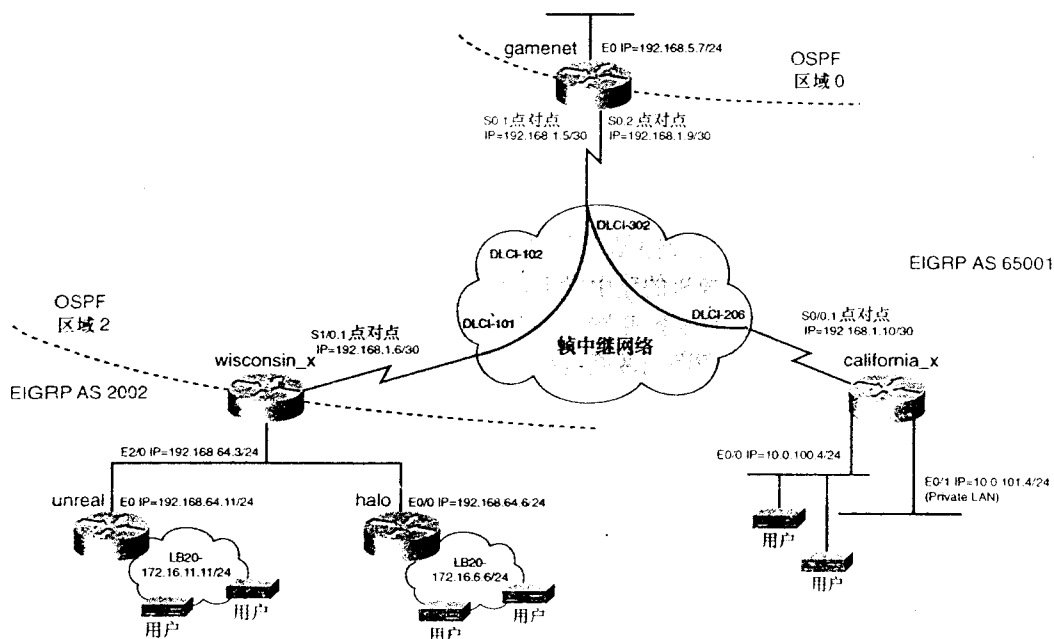


图 2-11 GameNetworks.com

2.2.3 实验目的

- 按照图 2-11 所示，配置路由选择协议。在 wisconsin_x 路由器上广播 EIGRP 路由更新数据包的惟一接口应当是局域网的接口。
- 在 wisconsin_x 和 gamenet 路由器上配置 OSPF 的协议。wisconsin_x 路由器的串行接口将放在 OSPF 区域 2 里。gamenet 路由器的串行接口 s0.1 将放在 OSPF 区域 2 里，局域网接口放在 OSPF 区域 0 里。
- gamenet 路由器的 s0.2 接口将配置为 EIGRP，使用的是 AS 65001。california_x 路由器的所有接口都处于 EIGRP 域里。
- 在 unreal 路由器上配置一个环回接口，IP 地址为 172.16.11.11/24，在 halo 路由器上配置一个 IP 地址为 172.16.6.6/24。将这些网络以 EIGRP 的路由通告。
- 将来给所有的路由分配一个标记值 100，并且由 halo 路由器通告出去。这包括所有的本地直连的路由和局域网的网段。将来，其他的网络也许要添加到 halo 路由器上，确保这些网络也得到一个标记值 100。
- 当一条路由标记为值 100 后，确保那个标记值对于路由域 EIGRP 65001 是保留的。
- 路由只被 halo 路由器通告出去而不是其他的路由器，并且这些路由到达 gamenet 路由器时应当表现为 OSPF 类型 1 的路由。

- 给来自 unreal 路由器的网段 172.16.11.0/24 分配一个 OSPF 的标记值 10。
- 当在 gamenet 路由器上将 OSPF 路由重分发到 EIGRP 65001 中时，只重分发具有标记值 100 的那些路由。california_x 路由器应当能够 ping 通 halo 路由器的网络 172.16.6.0/24，但是不能 ping 通 unreal 路由器的 172.16.11.0/24 网络。
- 不允许 california_x 路由器上的专有局域网络 10.0.101.0/24 被重分发到 OSPF 的路由域里。

2.2.4 需要的设备

- 6 台思科的路由器，3 台路由器通过背对背的 V.35 线缆互连，或以同样的方式连接到一个充当帧中继交换机的路由器上。
- 4 个局域网段，通过集线器或者交换机实现。这个图显示 california_x 路由器有两个局域网接口，其中的一个局域网接口可以被环回接口替代。

2.2.5 物理布局和预规划

- 按照图 2-11 所示，给路由器连接集线器和串行线缆。
- 需要一台配置了两条 PVC 的帧中继交换机。范例 2-35 列出了在这个实验中使用的帧中继的配置。

范例 2-35 帧中继交换机的配置

```
hostname frame_switch
!
frame-relay switching
!
interface Serial0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
 frame-relay intf-type dce
 frame-relay route 102 interface Serial1 101
 frame-relay route 302 interface Serial4 206
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 2000000
 frame-relay intf-type dce
 frame-relay route 101 interface Serial0 102
!
<<<text omitted>>>
!
interface Serial4
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 206 interface Serial0 302
```

2.2.6 实验步骤

配置帧中继交换机，并且将 3 台路由器以背对背的方式连接到帧中继交换机上。使用 V.35 线缆来连接路由器。按照图 2-11 所示，通过使用交换机或者集线器，建立 4 个以太网局域网段。

当物理连接完成后，按照图 2-11 所示给所有的局域网和广域网的接口分配 IP 地址。在 gamenet 和 wisconsin_x 路由器之间以及 gamenet 和 california_x 路由器之间配置一个帧中继的点对点网络。按照图所示使用数据链路连接识别符（DLCI）。范例 2-36 列出了 gamenet、wisconsin_x 和 california_x 路由器的帧中继配置。

范例 2-36 gamenet、wisconsin_x 和 california_x 的帧中继配置

```
hostname gamenet
!
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 192.168.1.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 102
!
interface Serial0.2 point-to-point
 ip address 192.168.1.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 302

-----

hostname wisconsin_x
!
interface Serial1/0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type cisco
!
interface Serial1/0.1 point-to-point
 ip address 192.168.1.6 255.255.255.252
 frame-relay interface-dlci 101

-----

hostname california_x
!
interface Serial0/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
 ip address 192.168.1.10 255.255.255.252
 frame-relay interface-dlci 206
```

当配置完所有局域网和广域网的接口后，分配 IP 地址并且验证本地连接。所有的路由器应当能够 ping 通它们的邻接路由器。例如，unreal、wisconsin_x 和 halo 应当能够 ping 通其他路由器的以太网地址。当本地连接验证后，可以开始配置路由选择协议。

在试图控制路由更新数据包和写路由映射之前，确保所有的网络具有 IP 连接性，能够自由地重分发所有的路由，而无需过滤。通过这样的确认，当问题是由路由重分发或者其他和路由选择协议有关的问题引起时，可以避免对路由映射进行故障排查。

开始在 wisconsin_x、unreal 和 halo 路由器之间配置 EIGRP 的域。在这 3 台路由器上配置 EIGRP 是相当直观的。在 wisconsin_x 路由器上，你需要一个 **network** 语句和一个 **default-metric**，因为你需要将 OSPF 的路由重分发到 EIGRP 里。范例 2-37 列出了 wisconsin_x 路由器上的 EIGRP 配置。

范例 2-37 对于 wisconsin_x 的 EIGRP 配置

```
hostname wisconsin_x
!
router eigrp 2002
 redistribute ospf 2002
 network 192.168.64.0
 default-metric 1000 100 254 1 1500
 no auto-summary
```

unreal 和 halo 路由器的 EIGRP 配置是完全一样的。在范例 2-38 中，EIGRP 配置演示了两种方法来配置 EIGRP 网络。在思科 IOS 软件版本 12.1 中，EIGRP 支持带通配符掩码 (wildcard) 的 **network** 语句。网络 172.16.11.0 正在使用这种方法的配置，我们的这个范例正在遵循标准的方法对 192 网络配置 EIGRP。这样做的目的纯粹是为了演示。

范例 2-38 unreal 和 halo 路由器的配置

```
!
hostname unreal
!
router eigrp 2002
 network 172.16.11.0 0.0.0.255
 network 192.168.64.0
 no auto-summary
 eigrp log-neighbor-changes
!

-----

hostname halo
!
router eigrp 2002
 network 172.16.6.0 0.0.0.255
 network 192.168.64.0
 no auto-summary
 eigrp log-neighbor-changes
```

接下来可以在 gamenet 路由器上配置 OSPF 和 EIGRP。对于 EIGRP 所配置的自治系统 ID 为 65001。发送 EIGRP 更新数据包的惟一接口是 s0.2，也就是 192.168.1.9。接口 S0.1 处于 OSPF 区域 2 中，而接口 E0 处于 OSPF 区域 0 中。范例 2-39 列出了在 gamenet 路由器上的 OSPF 和 EIGRP 配置。这时，在任何路由器上都没有配置路由映射。

范例 2-39 gamenet 路由器上的 OSPF 和 EIGRP 配置

```
hostname gamenet
!
router eigrp 65001
 redistribute ospf 2002
 passive-interface Ethernet0
 passive-interface Serial0.1
 network 192.168.1.0
 default-metric 1000 100 254 1 1500
 no auto-summary
!
router ospf 2002
 redistribute eigrp 65001 subnets
 network 192.168.1.5 0.0.0.0 area 2
 network 192.168.5.0 0.0.0.255 area 0
 default-metric 100
!
```

california_x 路由器将会配置 EIGRP 协议，自治系统 ID 为 65001。范例 2-40 列出了在 california_x 路由器上的 EIGRP 配置。

范例 2-40 在 california_x 路由器上的 EIGRP 配置

```
hostname california_x
!
router eigrp 65001
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

在所有的路由器上配置路由选择协议后，使用标准的 ping 测试来验证 IP 的连接性。确保 california_x 路由器可以 ping 通 gamenet 的局域网以及 halo 和 unreal 路由器。确保环回网络被通告出去并且能够被 unreal 和 halo 路由器连通。不要试图写路由映射来实现过滤，我们这样做的目的首先是为了实现 IP 的可达性。

这个实验指导要求用户写一个路由映射，将来自 halo 路由器的路由打上标记 100，并且把来自 unreal 路由器的路由打上标记 10。也可以将 192.168.64.0/24 的路由打上标记 100。因此，在 wisconsin_x 路由器上，可以在重分发时写一个路由映射来完成这个任务。

遵循配置路由映射的五步过程，首先使用相关的 **match** 和 **set** 命令来配置路由映射。这个路由映射（即 **set_tag**）将匹配和 **match ip route-source** 命令相关的路由。来自源 IP 地址 192.168.64.11 的路由，也就是 unreal 路由器的路由，会将路由标记设置为 10。来自源 IP 地址 192.168.64.6 的路由，也就是 halo 路由器的路由，会将路由标记设置为 100。来自这些源的路由会将度量值设置为 OSPF 类型 1 的度量。范例 2-41 列出了在 wisconsin_x 路由器上的路由映射的语法。

此时我们已经完成了配置路由映射的第 1~3 步，现在应用路由映射。在这个模型中，在 wisconsin_x 路由器上进行 EIGRP 到 OSPF 的重分发时，可以应用这个路由映射。范例 2-42 列出了 wisconsin_x 路由器的完整配置，包括访问控制列表。

范例 2-41 在 wisconsin_x 路由器上的 route-map set_tag 配置

```

hostname wisconsin_x
!
route-map set_tag permit 10    ←First route-map instance
 match ip route-source 1      ←Match ACL 1, 192.168.64.11
 set tag 10                   ←Set tag to 10
!
route-map set_tag permit 20    ←Second route-map instance
 match ip route-source 2      ←Match ACL 2, 192.168.64.6
 set metric-type type-1       ←Set route type to Ext OSPF type-1
 set tag 100                  ←Set tag to 100
!
route-map set_tag permit 30    ←Third route-map instance
 match ip address 10          ←Match ACL 10, all other routes
 set tag 100                  ←Set tag to 100
!

```

范例 2-42 wisconsin_x 路由器的配置

```

hostname wisconsin_x
!
<<<text omitted>>>
!
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 frame-relay lmi-type cisco
!
interface Serial1/0.1 point-to-point
 ip address 192.168.1.6 255.255.255.252
 frame-relay interface-dlci 101
!
<<<text omitted>>>
!
interface Ethernet2/0
 ip address 192.168.64.3 255.255.255.0
!
router eigrp 2002
 redistribute ospf 2002          ←redistribute OSPF
 network 192.168.64.0
 default-metric 1000 100 254 1 1500 ←default metric
 no auto-summary
!
router ospf 2002
 redistribute eigrp 2002 subnets route-map set_tag ←Redistribute and call route-map
 network 192.168.1.6 0.0.0.0 area 2
 default-metric 10                ←default metric
!
access-list 1 permit 192.168.64.11    ←match routes from 192.168.64.11
access-list 2 permit 192.168.64.6    ←match routes from 192.168.64.6
access-list 10 permit any            ←match all other routes/192.168.64.0
!
route-map set_tag permit 10          ←route-map "set_tag" begins
 match ip route-source 1
 set tag 10
!
route-map set_tag permit 20

```

(待续)

```

match ip route-source 2
set metric-type type-1
set tag 100
!
route-map set_tag permit 30
match ip address 10
set tag 100

```

这个模型的另外一个需求就是在 gamenet 路由器上只将标记值为 100 的 OSPF 路由重分发到 EIGRP 65001 中，并且保留这个标记。可以建立一个路由映射，并且在重分发的过程中应用它，只匹配标记为 100 的路由以达到这个目的。可以使用 **match tag** 命令来完成。范例 2-43 列出了所需的路由映射。

范例 2-43 gamenet 路由器上的路由映射 match_tag100

```

hostname gamenet
!
route-map match_tag100 permit 10      ←begin route-map "match_tag100"
match tag 100                        ←match the tag value of 100
set tag 100                          ←set the tag for EIGRP.
!

```

这个路由映射将会应用于从 OSPF 到 EIGRP 的重分发过程中。然而，在应用这个路由映射之前，配置这个模型中所需的最后一个路由映射。

最后一个需求也是防止专有局域网络 10.0.101.0/24，来自 california_x 路由器，在 gamenet 路由器上被从 EIGRP 重分发到 OSPF 的域里。可以在重分发的过程中使用路由映射来防止这一点。用于过滤这个子网的路由映射将调用一个访问控制列表来只匹配网络 10.0.101.0/24。范例 2-44 列出了这个路由映射，叫做 filter_net，用于过滤网络 10.0.101.0/24 和相关的访问控制列表。

范例 2-44 在 gamenet 路由器上的路由映射 filter_net

```

hostname gamenet
!
access-list 10 deny 10.0.101.0 0.0.0.255 ←deny network 10.0.101.0/24
access-list 10 permit any                ←Allow other networks to be redistributed
route-map filter_net permit 10          ←begin route-map "filter_net"
match ip address 10                     ←Match ACL 10

```

此时，在重分发的过程中可以应用两个路由映射。范例 2-45 列出了 gamenet 路由器的最终配置。

范例 2-45 gamenet 路由器的最终配置

```

hostname gamenet
!
interface Ethernet0
ip address 192.168.5.7 255.255.255.0
no ip directed-broadcast
media-type 10BaseT
!
<<<text omitted>>>

```

(待续)

```

!
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 192.168.1.5 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 102
!
interface Serial0.2 point-to-point
 ip address 192.168.1.9 255.255.255.252
 no ip directed-broadcast
 frame-relay interface-dlci 302
!
router eigrp 65001
 redistribute ospf 2002 route-map match_tag100      ←call route-map "match_tag100"
 passive-interface Ethernet0
 passive-interface Serial0.1
 network 192.168.1.0
 default-metric 1000 100 254 1 1500                ←set default metric
 no auto-summary
!
router ospf 2002
 redistribute eigrp 65001 subnets route-map filter_net ←call route-map "filter_net"
 network 192.168.1.5 0.0.0.0 area 2
 network 192.168.5.0 0.0.0.255 area 0
 default-metric 100                                ←set default metric
!
access-list 10 deny 10.0.101.0 0.0.0.255
access-list 10 permit any
route-map filter_net permit 10
 match ip address 10
!
route-map match_tag100 permit 10
 match tag 100
 set tag 100

```

如果路由器 california_x 只能够看到具备标记值为 100 的路由，可以 Ping 通 172.16.6.0/24 子网而不能 Ping 通 172.16.11.0/24 子网，这就验证了配置。范例 2-46 是一个路由表显示以及在路由器 california_x 上的 Ping 测试示例。

范例 2-46 验证 california_x 路由器的配置

```

california_x# 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, 1 subnets
D EX   172.16.6.0 [170/3097600] via 192.168.1.9, 02:47:46, Serial0/0.1
D EX 192.168.64.0/24 [170/3097600] via 192.168.1.9, 02:48:50, Serial0/0.1
 10.0.0.0/24 is subnetted, 2 subnets

```

(待续)

```
C 10.0.100.0 is directly connected, Ethernet0/0
C 10.0.101.0 is directly connected, Ethernet0/1
  192.168.1.0/30 is subnetted, 2 subnets
C 192.168.1.8 is directly connected, Serial0/0.1
D 192.168.1.4 [90/2681856] via 192.168.1.9, 02:58:26, Serial0/0.1
california_x#
california_x# show ip route 172.16.6.0
Routing entry for 172.16.6.0/24
  Known via "eigrp 65001", distance 170, metric 3097600
  Tag 100, type external
  Redistributing via eigrp 65001
  Last update from 192.168.1.9 on Serial0/0.1, 02:48:18 ago
  Routing Descriptor Blocks:
    * 192.168.1.9, from 192.168.1.9, 02:48:18 ago, via Serial0/0.1
      Route metric is 3097600, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1000 Kbit
      Reliability 254/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
california_x#
california_x# ping 172.16.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.6.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
california_x# ping 172.16.11.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.11, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
california_x#
```

为了验证专有子网（10.0.101.0/24）已从 OSPF 中过滤，可以查看 wisconsin_x 路由器的路由表，如范例 2-47 所示。

范例 2-47 wisconsin_x 路由器的最终路由表

```
wisconsin_x# 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, 2 subnets
    D 172.16.11.0 [90/409600] via 192.168.64.11, 03:00:27, Ethernet2/0
    D 172.16.6.0 [90/409600] via 192.168.64.6, 03:00:27, Ethernet2/0
  C 192.168.64.0/24 is directly connected, Ethernet2/0
  O IA 192.168.5.0/24 [110/58] via 192.168.1.5, 03:01:39, Serial1/0.1
    10.0.0.0/24 is subnetted, 1 subnets
  O E2 10.0.100.0 [110/100] via 192.168.1.5, 03:01:03, Serial1/0.1
    192.168.1.0/30 is subnetted, 2 subnets
  O E2 192.168.1.8 [110/100] via 192.168.1.5, 03:01:44, Serial1/0.1
  C 192.168.1.4 is directly connected, Serial1/0.1
wisconsin_x#
```


2.3 实验4：配置策略性路由

2.3.1 练习场景

路由映射也可以用于策略性路由。可以使用策略性路由来强制流量走和正常的转发/路由表不同的路径。可以使用策略性路由基于 ToS、数据包的大小和类型以及源地址等因素来控制流量。这个实验给用户提供了一个练习的机会，基于数据包的大小来配置复杂的策略性路由映射，并且控制默认的路由。

2.3.2 实验练习

Wizards of the Woods 是一个先进的奇异卡游戏生产厂商，奇异的角色扮演游戏公司和计算机游戏公司。Wizards of the Woods 公司按照地理区域将公司划分成几部分，在每一个区域中，都有两条来自总部路由器（也就是 wow 路由器）的帧中继 PVC。一条 PVC 的速率是 T1，这条 PVC 运行在 wow 和 plains 路由器之间。另外一条是低速的 PVC，速率是 64 kbit/s，运行在 wow 和 swamp 路由器之间。wow 路由器也提供对每一个区域的因特网服务。Wizards 公司想要控制和规划通过广域网链路和到达 wow 服务器的流量，通过策略性路由实现。你的任务就是使用下面严格的设计指导来配置 IP 网络和使用策略性路由。

- 按照图 2-12 所示配置 Wizards of the Woods 公司的 IP 网络。
- 按照图 2-12 所示配置帧中继的网络。
- 按照图 2-12 所示配置所有的 IP 地址。
- 使用“实验目的”部分了解配置的要求。

2.3.3 实验目的

- 按照图 2-12 所示，配置 EIGRP 作为路由选择协议，使用 65002 作为自治系统的 ID。
- 配置 EIGRP，使得路由选择协议对于来自 forest 路由器的流量将更愿意使用在 plains 和 wow 路由器之间的更高带宽的链路，而不是 swamp 和 wow 路由器之间的链路。EIGRP 对于 forest、mountain 和 island 路由器的流量将更愿意使用这条路径实现路由（提示：在串行接口上适当地设置带宽）。
- 这个实验的测试和功能在有一条到因特网的可用连接的情况下会极大地增强。wow 路由器将被配置为对于因特网的流量通告一条默认路由。如果一条因特网的连接不可用，可以用一个环回地址或者另外一台路由器来仿真它。
- 按照下面的指导配置策略性路由：
 - 来自 mountain 和 island 路由器的 IP 流量，如果是小的数据包（0~1199 字节），它们到 wow 服务器的流量应当使用 plains 和 wow 路由器之间的高速链路。
 - 来自 mountain 和 island 路由器的 IP 流量，如果是大的数据包（1200~1544 字节），它们到 wow 服务器的流量应当使用 swamp 和 wow 路由器之间的低速链路。
 - mountain 路由器的因特网流量应当使用通过 plains 路由器的高速链路。

- 图 2-12 wizards of the Woods

- 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

- 4 个局域网段，通过集线器或者交换机提供。图 2-12 显示 wow 路由器具有两个局域网接口，其中的一个接口可以通过环回接口或者另外一台路由器来仿真因特网，在一个真正的因特网连接不可用的情况下可以这样做。
- 可以使用 IP 工作站或者服务器来仿真 wow 服务器。

2.3.5 物理布局和预规划

- 按照图 2-12 所示将集线器和串行线缆连接到路由器上。
- 需要一台连接了两条 PVC 的帧中继交换机。范例 2-28 列出了本实验中帧中继的配置。

范例 2-48 帧中继交换机的配置

```
hostname frame_switch
!
frame-relay switching
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 2000000
frame-relay intf-type dce
frame-relay route 102 interface Serial1 101
frame-relay route 302 interface Serial4 206
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 2000000
frame-relay intf-type dce
frame-relay route 101 interface Serial0 102
!
<<<text omitted>>>
!
interface Serial4
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 206 interface Serial0 302
```

2.3.6 实验步骤

配置帧中继交换机并且将 3 台路由器以背对背的形式连接到帧中继交换机，使用 V.35 线缆连接路由器。通过使用交换机或者集线器，建立 4 个局域网段，如图 2-12 所示。

当物理连接完成后，按照图 2-12 所示给所有的局域网接口和广域网接口分配 IP 地址。将帧中继网络配置为所有的路由器之间在广域网段上的一个多点网络。使用图中的 DLCI 值。因为帧中继网络是一个多点网络，记住你需要在某些点上关闭 EIGRP 的水平分割。这时，你可能想设置带宽的语句使得 EIGRP 通过网络可以选择最佳的路径。范例 2-49 列出了所有路由器的帧中继配置。

范例 2-49 wow、plains 和 swamp 路由器的帧中继配置

```

hostname wow
!
interface Serial0
  bandwidth 1544                                ←BW for EIGRP
  ip address 192.168.1.7 255.255.255.0
  encapsulation frame-relay
  no ip split-horizon eigrp 65002                ←used to disable split-horizons
  no ip mroute-cache
  frame-relay map ip 192.168.1.3 102 broadcast    ←Map statement to plains
  frame-relay map ip 192.168.1.4 302 broadcast    ←Map statement to swamp
  frame-relay lmi-type cisco
!

hostname plains
!
interface Serial1/0
  bandwidth 1544                                ←BW for EIGRP
  ip address 192.168.1.3 255.255.255.0
  encapsulation frame-relay
  frame-relay map ip 192.168.1.4 101 broadcast    ←Map statement to swamp
  frame-relay map ip 192.168.1.7 101 broadcast    ←Map statement to wow
  frame-relay lmi-type cisco
!

hostname swamp
!
interface Serial0/0
  bandwidth 64                                  ←BW for EIGRP
  ip address 192.168.1.4 255.255.255.0
  encapsulation frame-relay
  no ip mroute-cache
  frame-relay map ip 192.168.1.3 206 broadcast    ←Map statement to plains
  frame-relay map ip 192.168.1.7 206 broadcast    ←Map statement to wow
  frame-relay lmi-type cisco
!

```

当配置完所有的局域网和广域网的接口后，分配 IP 地址并且验证本地连接。所有的路由器应当能够 ping 通它们的邻接路由器。例如，plains、swamp 和 forest 路由器应当能够 ping 通其他路由器的以太接口地址。当本地连通性的验证完成后，可以开始配置路由选择协议。

开始在所有的路由器上配置 EIGRP 域，从 wow 路由器开始。在 wow 路由器上，需要两个 **network** 语句，一个是对于网络 172.16.0.0，另一个是对于网络 192.168.1.0。这台路由器也需要为因特网流量产生一条默认路由。为了产生一条默认路由，使用命令 **ip route 0.0.0.0 0.0.0.0 206.191.241.41** 配置一条到地址 206.191.241.41 的默认静态路由。为了使 wow 路由器通告这条路由，它需要被重分发到 EIGRP 里。范例 2-50 列出了 wow 路由器上的 EIGRP 配置。

范例 2-50 在 wisconsin_x 上的 EIGRP 配置

```

hostname wow
!
router eigrp 65002
  redistribute static                            ←redistribute the default route
  network 172.16.0.0

```

(待续)

```
network 192.168.1.0
default-metric 10000 100 254 1 1500      ←default metric
no auto-summary
!
ip classless
ip route 0.0.0.0 0.0.0.0 206.191.241.41  ←default route
```

注意: **ip classless** 启用后, 数据包才会遵循一条默认路由。

因为帧中继网络是多点的网络, 应当使用命令 **no ip split-horizon eigrp 65002** 将串行接口上的 EIGRP 水平分割关掉。如果没有关掉 EIGRP 的水平分割, 如果 plains 和 swamp 路由器之间的以太网链路失效了, 那么 plains 路由器将收不到来自 swamp 路由器的路由, 并且路由将会中断。EIGRP 配置的另外一个重要的部分在先前的范例中已经列出, 就是在串行接口上配置 **bandwidth** 语句。**bandwidth** 语句的配置允许 EIGRP 选择最佳的可能路径实现路由。

plains 和 swamp 路由器的 EIGRP 配置类似于 wow 路由器。范例 2-51 列出了配置。

范例 2-51 plains 和 swamp 路由器的 EIGRP 配置

```
hostname plains
!
router eigrp 65002
 network 172.16.0.0
 network 192.168.1.0
no auto-summary
!

-----

hostname swamp
!
router eigrp 65002
 network 172.16.0.0
 network 192.168.1.0
no auto-summary
```

在 forest、mountain 和 island 路由器上的 EIGRP 配置是非常直接的, 如范例 2-52 所示。

范例 2-52 forest、mountain 和 island 路由器的 EIGRP 配置

```
hostname forest
!
router eigrp 65002
 network 172.16.0.0
no auto-summary
!

-----

hostname mountain
!
router eigrp 65002
 network 172.16.0.0
no auto-summary
!

-----

hostname island
!
router eigrp 65002
 network 172.16.0.0
no auto-summary
```

当所有的路由器配置完 EIGRP 之后,应当具有 IP 的端对端的连接性。island 和 mountain 路由器应当能够连通 wow 服务器。EIGRP 应当能够通告一条默认路由。范例 2-53 列出了 island 路由器的路由表。

范例 2-53 island 路由器的路由表

```
island# 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 172.16.2.6 to network 0.0.0.0
 172.16.0.0/24 is subnetted, 3 subnets
D    172.16.7.0 [90/2246656] via 172.16.2.6, 01:07:24, Ethernet0/0
D    172.16.1.0 [90/307200] via 172.16.2.6, 02:10:57, Ethernet0/0
C    172.16.2.0 is directly connected, Ethernet0/0
D    192.168.1.0/24 [90/2221056] via 172.16.2.6, 02:10:57, Ethernet0/0
D*EX 0.0.0.0/0 [170/2246656] via 172.16.2.6, 01:07:24, Ethernet0/0
island#
```

这个实验的可选部分要求用户在 wow 路由器上配置 NAT 来实现对因特网的可达性。用实际的 IP 主机来做测试将帮助用户验证路由映射和策略性路由是否正常工作。实际的 IP 主机可以用环回接口来替换，并且启用本地策略路由。当配置 NAT 时，将 wow 路由器的 serial 0 和 E4 接口作为 NAT 的内口，E3 接口作为 NAT 的外口。因为你只有一个 IP 地址，可以使用端口地址翻译 (PAT)，有时候称为过载特性。在这个模型中使用的 NAT/PAT 配置在范例 2-54 中列出来了。关于配置 NAT 的更多详细的信息，参考《CCIE 实验指南 (第 1 卷)》。

范例 2-54 wow 路由器上的 NAT/PAT 配置

```
hostname wow
!
interface Ethernet3
 ip address 206.191.241.43 255.255.255.248
 no ip directed-broadcast
 ip nat outside                                     ←NAT outside interface/Internet
 media-type 10BaseT
!
interface Ethernet4
 ip address 172.16.7.7 255.255.255.0
 no ip directed-broadcast
 ip nat inside                                     ←NAT inside interface
 media-type 10BaseT
!
interface Serial0
 bandwidth 1544
 ip address 192.168.1.7 255.255.255.0
 no ip directed-broadcast
 ip nat inside                                     ←NAT inside interface
 encapsulation frame-relay
 no ip split-horizon eigrp 65002
 no ip mroute-cache
 frame-relay map ip 192.168.1.3 102 broadcast
 frame-relay map ip 192.168.1.4 302 broadcast
```

(待续)

```
frame-relay lmi-type cisco
!
ip nat inside source list 101 interface Ethernet3 overload ←PAT enabled for E3
!
access-list 101 permit ip any any ←translate all traffic
```

为了配置本实验的策略性路由，需要在 forest 路由器上配置策略性路由。这个实验并不要求流量的返回路径必须和出去的路径是同一个路径，然而，作为额外的练习，你可能想在 wow 路由器上配置策略性路由，使得流量遵循同一个返回路径。

这个实验的目的就是要求用户按照下面的指导配置策略性路由。

- 来自 mountain 和 island 路由器的 IP 流量，如果是小的数据包（0~1199 字节），它们到 wow 服务器的流量应当使用 plains 和 wow 路由器之间的高速链路。
- 来自 mountain 和 island 路由器的 IP 流量，如果是大的数据包（1200~1544 字节），它们到 wow 服务器的流量应当使用 swamp 和 wow 路由器之间的低速链路。
- mountain 路由器的因特网流量应当使用通过 plains 路由器的高速链路。
- island 路由器的因特网流量应当使用通过 swamp 路由器的低速链路。
- 对策略性路由配置快速交换。

在 forest 路由器上用于策略性路由的路由映射将会有 4 个路由映射的实例。第一个实例将匹配来自路由器 mountain（172.16.2.10）的流量，和 island 路由器（172.16.2.5）的流量。当来自这些源的流量被验证后，将匹配小的数据包长度，范围是 0~1199 字节。通过这两项条件的流量会将下一跳设置为 172.16.1.3，使用到 plains 路由器的高速链路。第二个路由映射的实例将匹配来自相同地址的流量，但是这个实例将匹配大的数据包的长度，范围是 1200~1544 字节。通过这两个条件的流量将被转发到下一跳 172.16.1.4，使用到 swamp 路由器的低速链路。

最后两个路由映射的实例用于因特网的流量。一个实例将匹配来自 mountain 路由器的流量，也就是 172.16.2.10，并将 IP 默认的下一跳设置为 plains 路由器，也就是 172.16.1.3。另外一个实例将匹配来自 island 路由器的流量，也就是 172.16.2.5，并将 IP 默认的下一跳设置为 swamp 路由器，也就是 172.16.1.4。回忆一下，当路由器在它的转发/路由表中没有数据包的目的地址时，它就会使用 IP 默认的下一跳地址。

回忆一下配置策略性路由的步骤，如下：

- 第 1 步 配置访问控制列表。
- 第 2 步 配置路由映射的实例。
- 第 3 步 配置 match 命令。
- 第 4 步 配置 set 命令。
- 第 5 步 在接口上配置策略性路由。
- 第 6 步 配置快速交换。
- 第 7 步 （可选）配置本地的策略性路由。

范例 2-55 包括了在 forest 路由器上配置策略性路由的第 1~4 步的配置。

最后一部分的配置，也就是第 5~6 步，要求用户应用策略性路由，并且启用策略性路由的快速交换。这是通过接口命令 **ip policy route-map** 和 **ip route-cache policy** 来完成的。范例 2-56 列出了 forest 路由器的完整配置。

范例 2-55 forest 路由器上的路由映射和访问控制列表的配置

```

Hostname forest
!
access-list 110 permit ip host 172.16.2.10 172.16.7.0 0.0.0.255
access-list 110 permit ip host 172.16.2.5 172.16.7.0 0.0.0.255
!
access-list 130 deny ip any 172.16.0.0 0.0.255.255
access-list 130 deny ip any 192.168.1.0 0.0.0.255
access-list 130 permit ip host 172.16.2.10 any
!
access-list 140 deny ip any 172.16.0.0 0.0.255.255
access-list 140 deny ip any 192.168.1.0 0.0.0.255
access-list 140 permit ip host 172.16.2.5 any
!
route-map policy_1 permit 10          ←PBR small packets
  match ip address 110
  match length 0 1199
  set ip next-hop 172.16.1.3
!
route-map policy_1 permit 20          ←PBR large packets
  match ip address 110
  match length 1200 1544
  set ip next-hop 172.16.1.4
!
route-map policy_1 permit 30          ←PBR for default routing
  match ip address 130
  set ip default next-hop 172.16.1.3
!
route-map policy_1 permit 40          ←PBR for default routing
  match ip address 140
  set ip default next-hop 172.16.1.4
!

```

范例 2-56 forest 路由器的配置

```

hostname forest
!
<<<text omitted>>>
!
interface Ethernet0/0
  ip address 172.16.1.6 255.255.255.0
!
interface Ethernet0/1
  ip address 172.16.2.6 255.255.255.0
  ip route-cache policy
  ip policy route-map policy_1
!
router eigrp 65002
  network 172.16.0.0
  no auto-summary
  no eigrp log-neighbor-changes
!
ip classless
no ip http server
!
access-list 110 permit ip host 172.16.2.10 172.16.7.0 0.0.0.255
access-list 110 permit ip host 172.16.2.5 172.16.7.0 0.0.0.255
access-list 130 deny ip any 172.16.0.0 0.0.255.255

```

(待续)


```

access-list 130 deny ip any 192.168.1.0 0.0.0.255
access-list 130 permit ip host 172.16.2.10 any
access-list 140 deny ip any 172.16.0.0 0.0.255.255
access-list 140 deny ip any 192.168.1.0 0.0.0.255
access-list 140 permit ip host 172.16.2.5 any
route-map policy_1 permit 10
 match ip address 110
 match length 0 1199
 set ip next-hop 172.16.1.3
!
route-map policy_1 permit 20
 match ip address 110
 match length 1200 1544
 set ip next-hop 172.16.1.4
!
route-map policy_1 permit 30
 match ip address 130
 set ip default next-hop 172.16.1.3
!
route-map policy_1 permit 40
 match ip address 140
 set ip default next-hop 172.16.1.4

```

范例 2-57 列出了 wow 路由器上的策略性路由的配置。

范例 2-57 wow 路由器上的策略性路由的配置

```

hostname wow
!
ip subnet-zero
ip name-server 206.191.193.1
!
<<<text omitted>>>
!
interface Ethernet3
 ip address 206.191.241.43 255.255.255.248
 no ip directed-broadcast
 ip nat outside
 media-type 10BaseT
!
interface Ethernet4
 ip address 172.16.7.7 255.255.255.0
 no ip directed-broadcast
 ip nat inside
 media-type 10BaseT
!
interface Serial0
 bandwidth 1544
 ip address 192.168.1.7 255.255.255.0
 no ip directed-broadcast
 ip nat inside
 encapsulation frame-relay
 no ip split-horizon eigrp 65002
 no ip mroute-cache
 frame-relay map ip 192.168.1.3 102 broadcast
 frame-relay map ip 192.168.1.4 302 broadcast
 frame-relay lmi-type cisco
!
router eigrp 65002
 redistribute static

```

(待续)

```

network 172.16.0.0
network 192.168.1.0
default-metric 10000 100 254 1 1500
no auto-summary
!
ip nat inside source list 101 interface Ethernet3 overload
ip classless
ip route 0.0.0.0 0.0.0.0 206.191.241.41
no ip http server
!
access-list 101 permit ip any any

```

为了测试这个策略，从 mountain 和 island 路由器上发出几个扩展 ping。通过在 forest 路由器上使用 **show route-map** 命令，就能够确定数据包是否进行了策略性路由。范例 2-58 演示了在 mountain 路由器上的两个 ping——一个 ping 到 wow 服务器，另外一个到 www.cisco.com（在因特网上）。

范例 2-58 测试和验证策略性路由

```

mountain# ping
Protocol [ip]:
Target IP address: 172.16.7.11
Repeat count [5]: 50
Datagram size [100]: 100
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 172.16.7.11, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (50/50), round-trip min/avg/max = 8/8/12 ms
mountain#
mountain# ping www.cisco.com
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 = 136/700/1116 ms
mountain#

forest# show route-map
route-map policy_1, permit, sequence 10                               ←small packets matched
  Match clauses:
    ip address (access-lists): 110
    length 0 1199
  Set clauses:
    ip next-hop 172.16.1.3
  Policy routing matches: 51 packets, 5814 bytes
route-map policy_1, permit, sequence 20
  Match clauses:
    ip address (access-lists): 110
    length 1200 1544
  Set clauses:
    ip next-hop 172.16.1.4
  Policy routing matches: 0 packets, 0 bytes
route-map policy_1, permit, sequence 30                               ←Internet traffic
  Match clauses:
    ip address (access-lists): 130
  Set clauses:

```

（待续）

```

ip default next-hop 172.16.1.3
Policy routing matches: 10 packets, 1140 bytes
route-map policy_1, permit, sequence 40
Match clauses:
  ip address (access-lists): 140
Set clauses:
  ip default next-hop 172.16.1.4
Policy routing matches: 0 packets, 0 bytes
forest#

```

为了在 island 路由器上执行相同的测试，除了将 ping 数据包的大小置为 1500 字节，你可以观察到策略性路由正在 forest 路由器上工作。范例 2-59 列出了在 island 路由器运行测试后，在 forest 路由器上执行 **show route-map** 命令的输出。

范例 2-59 在 wow 和 forest 路由器上的 show route-map 命令

```

forest# show route-map
route-map policy_1, permit, sequence 10
Match clauses:
  ip address (access-lists): 110
  length 0 1199
Set clauses:
  ip next-hop 172.16.1.3
Policy routing matches: 51 packets, 5814 bytes
route-map policy_1, permit, sequence 20      ←Large packets matched
Match clauses:
  ip address (access-lists): 110
  length 1200 1544
Set clauses:
  ip next-hop 172.16.1.4
Policy routing matches: 101 packets, 152914 bytes
route-map policy_1, permit, sequence 30
Match clauses:
  ip address (access-lists): 130
Set clauses:
  ip default next-hop 172.16.1.3
Policy routing matches: 10 packets, 1140 bytes
route-map policy_1, permit, sequence 40      ←Internet traffic
Match clauses:
  ip address (access-lists): 140
Set clauses:
  ip default next-hop 172.16.1.4
Policy routing matches: 12 packets, 1286 bytes
forest#

```



第三部分

组播路由

第 3 章

配置组播路由

组播已经用于不同的目的很多年了。现在我们所说的“组播”通常是指从一个特定的源来的视频和音频流。然而从一个更基本的角度来说，组播是指一种技术，它允许一台主机发送一股数据流，而到达多台主机。

没有组播时，用户可用的选项只有：

- 单播流——流量的拷贝数量等同于目的主机的数量
- 广播流——虽然只有来自源的一股数据流，但是它会为所有的工作站复制，而无论对方是否愿意接收。

在媒体流的早些日子，单播实际上就是通过因特网接收数据流的一种方法。这对发送者、接收者和它们之间的任何网络部分来说，都会导致大量的带宽浪费。

对现实世界来说，变化、修订和 RFC 的快速出现都呈现出对多媒体在线需求不断增长的趋势。组播骨干（MBONE）是最早的一种方法，它可以跨越因特网和服务提供商的网络传输组播流量。

本章的目的不是要求读者学会关于组播网络的设计和保护的细节，它只是充当一个复习的资料——有一系列的范例教会大家如何配置它们，特别是和 CCIE 实验考试有关！

3.1 组播的基础知识

考虑将一股流量发送给多个目的工作站，但不是所有的目的工作站，这导致了组播组这个概念的出现。目的工作站必须维护它在一个特定的组播组中的成员关系来接收组播的信息。如果不属于组播组的成员，组播流就不能发送给网络上的该工作站。

为了理解组播的有效性，考虑给用户提供一个频道内容的视频服务器，如图 3-1 所示。为了实现动作连贯和全屏的视觉效果，一股服务器到客户端的视频流大概需要 1.5 Mbps 的带

宽。在一个单播的环境中，服务器需要给网络中的每一个客户发送一股单独的视频流。（这将会消耗 $1.5 \times n$ Mbit/s 的链路带宽，其中 n = 客户端用户的数量）。如果服务器上有一个 10 Mbit/s 的以太网接口，那么 6~7 股服务器到客户端的视频流就会完全将网络接口的带宽消耗掉。即使是在一个高性能、高智能的吉比特以太网接口的服务器上，对于 1.5 Mbit/s 的视频流来说，实际的数额限制也就是 250~300 股视频流。因此，服务器接口的容量会成为一个很大的瓶颈，限制了每一个视频服务器的单播视频流的发送数量。重复的单播传输消耗了网络中的大量带宽，这也是另外一个较大的限制。如果服务器和客户之间的传输路径经过了 h_3 个路由器跳和 h_2 个交换机跳，那么“多个—单播”视频流要消耗掉 $1.5 \times n \times h_3$ Mbit/s 的路由器带宽，加上 $1.5 \times n \times h_2$ Mbit/s 的交换机带宽。如果有 100 个客户和服务器相距 2 个路由跳和 2 个交换机跳（如图 3-2 所示），那么一个多重单播视频流需要消耗 300 Mbit/s 的路由器带宽和 300 Mbit/s 的交换机带宽。即使将视频流的带宽在服务器上调整到 100 kbit/s（屏幕上提供一个小的窗口，视觉效果可接受），一个多重单播视频流也需要消耗 20 Mbit/s 的路由器带宽和 20 Mbit/s 的交换机带宽。

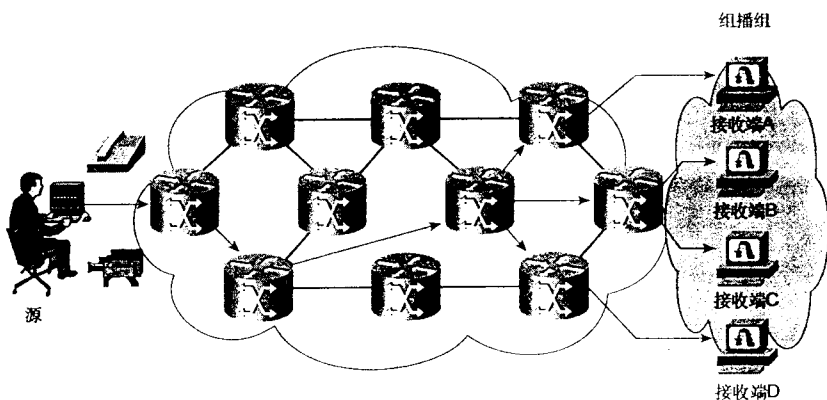


图 3-1 组播的目的：一对多路由

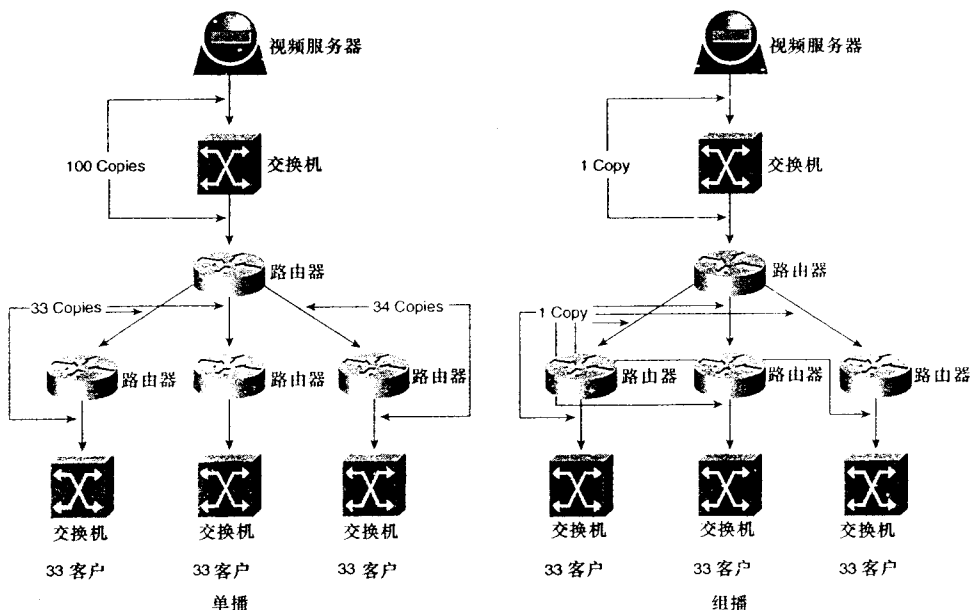


图 3-2 组播针对单播的有效性

组播数据包在网络中的适当路径上进行复制，利用协议无关组播（PIM）协议、Internet 组管理协议（IGMP）和其他相关的协议来建立最有效的路由传输机制。

组播提供了极大的优点。潜在的好处是节省整体的带宽和保留处理能力。然而，组播也有缺点。组播流在本质上大部分是 UDP。UDP 流量从定义上讲就是“尽力传递”，当然，这意味着“如果你能够，你可以得到它；如果你不能，也无所谓”。

UDP 从本质上讲通过传输可能丢弃更多的数据包。UDP 没有提供拥塞控制机制，例如窗口流控或者重传机制。在基于 UDP 的传输中，序列号是另外一个可能出现的问题。你可能看到由于数据包丢弃造成视频或者语音质量的下降。重放导致的无序数据包的情况是没有任何意义的。对于重复的数据包情况也是一样的。

3.2 IP 组播地址

IP 组播地址是 IPv4 的一个特定的地址空间，称为 *D 类地址*。用一种特定的二进制方法，所有类别的地址都被排列出来。表 3-1 显示了 IPv4 地址类别的列表。

表 3-1

IPv4 类别地址

地址类别	二进制表示	十进制表示	地址类别	二进制表示	十进制表示
A	0xxxxxxx	第一个八比特组 1~126	D	1110xxxx	第一个八比特组 224~239
B	10xxxxxx	第一个八比特组 128~191	E	1111xxxx	第一个八比特组 240~255
C	110xxxxx	第一个八比特组 192~223			

注意在表 3-1 中列出的地址值是有某些限制的。例如，127.0.0.0/8 的地址是保留的，用于不同类别的环回测试。而且，E 类地址是保留给将来使用的，或者用作研究的目的。D 类地址空间和组播有关，也是我们在这里所关注的。

D 类地址和许多先前的类别是有区别的。通常，一个 IP 地址被认为是代表网络上一个单独的、特定的主机的地址值（源地址）。在 D 类地址中，这个地址代表的是一个接收组。在许多情况下，组播组没有地理或者区域的边界。组播数据包的源永远被认为是单播的源地址（类别 A、B 或者 C）。

D 类地址又被进一步划分成一些可管理的段。因特网分配地址授权机构（IANA）控制 IPv4 地址空间的分配，包括组播地址。IANA 将 D 类地址的空间又划分成某些特定的组，以便于分配。

注意这些地址都是全局分配的并且一次只分配一个，而不是像其他的 IPv4 地址那样分配一个范围。表 3-2 列出了 D 类地址空间的分段。

表 3-2

D 类组播地址分配

描述	IPv4 地址范围	描述	IPv4 地址范围
本地链路地址（保留）	224.0.0.0/24	（子集）GLOP 地址	233.0.0.0/8
全局范围地址（已分配）	224.0.1.0 到 238.255.255.255	管理性范围的地址（本地）	239.0.0.0/8
（子集）源特定的组播地址	232.0.0.0/8		

3.2.1 本地链路地址

224.0.0.0 到 224.0.0.255 范围内的地址已经被 IANA 保留了，由本地网段（子网）上的网络协议使用。本地链路地址组播数据包有一个生存周期值（TTL）为 1，所以它们不会被其他路由器转发到不同的网段上。

许多路由选择协议使用组播地址来最大化它们的效率。表 3-3 列出了本地地址的某些范例。

表 3-3 共知的组播组

IP 组播地址	协议的使用	IP 组播地址	协议的使用
224.0.0.1	所有的系统	224.0.0.9	所有的 RIPv2 路由器
224.0.0.2	所有的路由器	224.0.0.10	所有的思科增强的 IGRP 路由器
224.0.0.5	所有的 OSPF 路由器	224.0.0.12	DHCP 服务器和代理服务
224.0.0.6	所有的 OSPF 指定路由器	224.0.0.13	所有的组播 PIM 路由器

这并不是已经分配的本地组播地址的完整列表，而是通用的大部分的表示方法。

3.2.2 全局分配地址

D 类组播地址空间的大部分被称为全局分配地址。IANA 控制和分配这些地址用于特定的组播应用和用法。这些地址再次代表了侦听特定数据流的组地址，并不代表信息的数据源。

这些地址也可以独立分配，而没有范围或者子网的概念。表 3-4 显示了某些范例。

表 3-4 常见应用的全局范围地址

IP 组播地址	协议的使用	IP 组播地址	协议的使用
224.0.1.1	所有的系统（网络时间协议）	224.0.1.40	思科 RP 发现（自动—RP）
224.0.1.39	思科 RP 宣告（自动—RP）		

这个范围的组播地址分配在 RFC 1112（*IP 组播的主机扩展*）中进一步定义。另外，可以在 [http://www.iana.org/ assignments/multicast-addresses](http://www.iana.org/assignments/multicast-addresses) 中进一步研究所有的当前分配。某些更进一步的地址在 RFC 1112 中保留使用。

3.2.3 源特定的地址

落在 232.0.0.0/8 范围内的地址主要保留用作源特定的组播地址。这种类型的组播允许组播网络的某些特性，例如集合点（RP）——以后讨论——通过目录服务学到特定的源信息后可以忽略。

源特定的组播也可以清除组播源发现协议（MSDP）或者其他自治系统之间的组播共享树的需求。作为 PIM 协议的扩展，其他不同于 RP 的机制可以提供“带外管理”的组播服务。

通常，接收者必须发出一个 join 命令到一个组播组的地址。如果多个接收者加入的是同一个组播组，即使信息是从不同的源服务器发送的，两种应用程序从两个发送源接收流量。这种解决方案对网络产生额外的流量。

在一个源特定的组播实施中，路由器看见特定于某个特殊的组播源的加入消息。这是在

IGMP 版本 3 中通过“包括”模式来完成的。接下来路由器将请求直接发送到源，而不是将它发送给通常所用的 RP。

在处理源特定的组播时，没有共享分发树。所有事件都是通过源路由树处理的。

3.2.4 GLOP 地址

落在 233.0.0.0/8 范围内的组播地址被 RFC 2770 保留用于 GLOP。作为一个有意义的注解，GLOP 并不代表任何首字母缩略语，然而，这是一个有意义的单词！一个自治系统的号码在通过因特网时会自动地转换成组播地址。

自治系统的号码是一个 16bit 的数字（1~65 535），代表因特网上独立的边界网关协议（BGP）系统的用户。在这里代入公式，取 16bit 并且将它们放入中间的两个八比特组，这样就会产生 256 个组播地址。

例如，AS 22222 用二进制表示是 01010110 11001110——或者是 86.206 变成两个八比特组并且转换成十进制。所以 AS 22222 在通过因特网时，会自动地转换成组播地址 233.86.206.0/24。

3.2.5 管理性范围的地址

也被称为*有限范围的地址*，管理性范围的地址落在 239.0.0.0/8 的范围内。RFC 2365 将这些地址在一个公司或者组织内部使用。私有的公司、校园或者其他网络可以使用这些地址来运行组播应用程序，而它们不能在自治系统之外转发。

服务提供商路由器通常会配置用来过滤这种类型的组播流量，以确保应用程序不要将流量发送到组播域之外。大的公司也可能想将它们分成小范围（组播的子网理论），从而将它们分离成小的组播域。

3.2.6 二层的组播地址

通常，一个系统上的网络接口卡（NIC）仅能够识别目的为它们的烧入的 MAC 地址（BIA）或者是广播的 MAC 地址的帧（所有的位都为 F）。在使用 IP 组播的网络中，多个主机需要能够接收具有相同目的地址的一股数据流。802.3 标准实际上通过使用最重要的字节（最左边的字节）的最不重要的位（位 0）允许这种情况发生。当这个位设置为 0 时，它代表的是 NIC 所注册的 BIA 的独立地址，当这个位设置为 1 时，它代表的是包含广播和组播地址的组地址。

对那些还记得 CCIE 笔试考试的读者来说，这就是在以太网 MAC 地址中被称为 I/G 的位。

作为一个二层组播地址的范例，考虑下面的中间系统到中间系统（IS-IS）的路由选择协议。起源于 OSI 无连接的网络服务（CLNS）协议组，IP IS-IS 在和邻居会话时，使用的是一个二层的组播地址：

Level 1 IS-IS 路由器和 01-80-C2-00-00-14 会话，Level 2 IS-IS 路由器和 01-80-C2-00-00-15 会话。

作为一个角注，相同字节的下一个次重要的位（位 1）代表一个本地分配的 MAC 地址

(LAA)，它允许多个独立的地址被接收。这也是在令牌环网络中常见的用法，在那里，“功能性地址”主要用于在网络操作中承担必要角色的设备。在图 3-3 中显示了一个 MAC 地址位的设计。

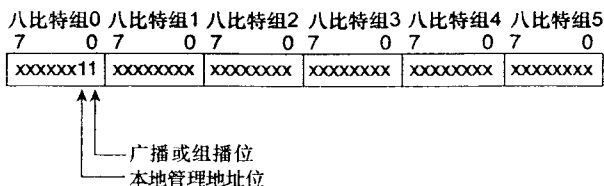


图 3-3 MAC 地址位的设计

你也许会再次想起来那些和 CCIE 笔试有关的日子以及其他的与以太网有关的琐事，MAC 地址的头 3 个八比特组代表的是 OUI 代码。IANA 已经为以太的组播 MAC 地址分配了一个组织独立识别符 (OUI) 的代码。这个 OUI 的代码是 01:00:5E，还分配了一个额外的位并且强制它为 0。这在 48bit 预分配的地址中产生了 25bit 的地址，还剩下 23bit 可变的位，如图 3-4 所示。

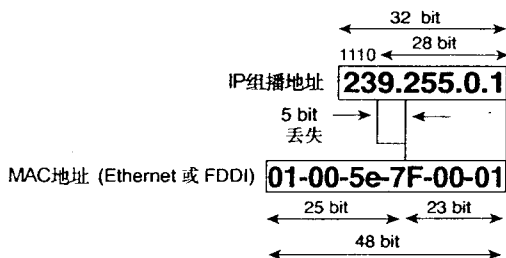


图 3-4 IP 组播到 MAC 地址

现在，我们来看一下和映射的值有关的一些小事。看一下在这里的 OUI 分配代码中的二进制值——特别是这个 E 值。E 是一个十六进制，用二进制表示是 1110。所有的 D 类 IP 地址都从二进制值 11100000 (224) 开始到 11101111 (239)。所有组播地址的头半个字节是 1110 (在十六进制中为 E)。

关于 IP 组播地址映射成 MAC 地址，然而，你可以发现还是给你留下了 23bit 需要重新映射。32bit 组播 IP 地址的低 23bit 在这里映射。因为先前的 4bit 已经用 E 代表了，这就留下了 5bit 没有映射，如图 3-5 所示。

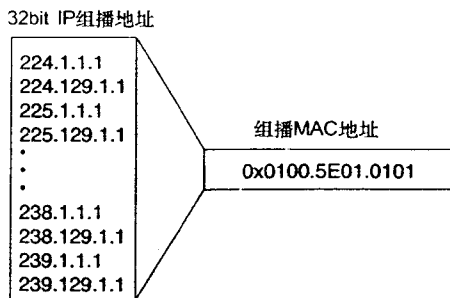


图 3-5 重叠的组播 MAC 地址

以太网组播 MAC 地址有某种重叠——同一个 MAC 地址被分配给 32 个不同的组播组。如果在一个以太网段上的用户要加入组播组 225.1.1.1，另外一个用户要加入组播组 225.129.1.1，那么两个用户都会收到两股组播流。在工程性组播网络的局域网环境下，这种重叠需要特别注意，并有意识地避免它们。

在令牌环网络中，重叠会更加严重。就像前面所说，令牌环网络使用的是功能性地址。也要记住令牌环使用的是逆序地址，所以比特是在字节的级别上交换的。三层的 IP 组播地址被映射到一个功能性地址，只留下了少数比特重叠。从 IP 组播地址中减去 4bit 共同的地址会留下 28bit 的重叠地址，或者说 268 435 200 个组播地址会映射到一个 MAC 地址。

不用说，在二层上组织组播的最好方法就是不要使用令牌环。在思科的配置中，默认的机制就是将组播数据包映射到广播帧 (FFFF.FFFF.FFFF)。

如果你想使用令牌环的功能性地址，在令牌环的接口上使用 `ip multicast use-functional` 命令。这将使用 C000.0004.0000 来映射组播 IP 数据包。

3.3 组播分发树

组播路由器建立分发树来控制组播流量在网络体系结构中传输流量所经过的路径。分发树由两种基本类型组成：源树和共享树。

3.3.1 源树

源分发树也称为最短路径树，顾名思义，它是一棵从树根（源）到树叶（接收者）具有最短路径的生成树。图 3-6 显示了组播源树的范例。

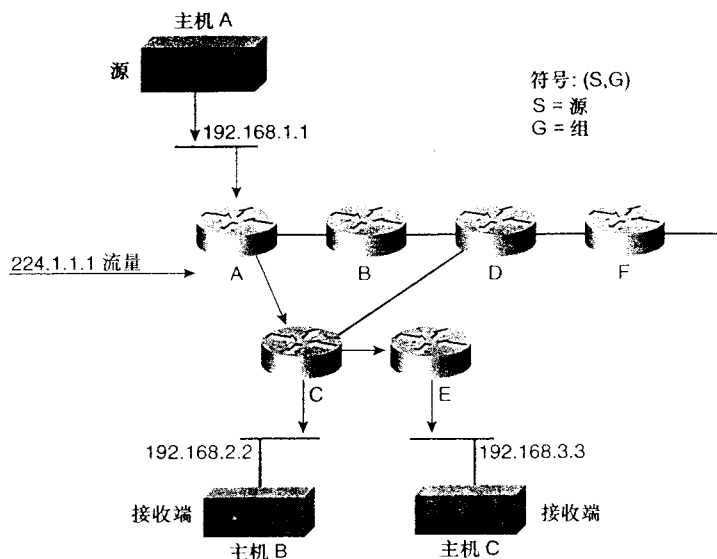


图 3-6 组播源树

S, G 的符号代表一对源（单播）地址和组（组播）地址：这一对地址发现最短路径树。在图 3-6 中，S, G 代表的是 (192.168.1.1, 224.1.1.1)。

每一棵源树都会使用 S, G 来表示。每一个独立的源向每一个特定的组发送组播数据导致一棵独立的 S, G 树生成。在大型网络中，这会导致在网络中生成不同寻常数量的 S, G 树。这种低效率需要共享生成树，并且鼓励对共享树的使用。

3.3.2 共享树

不像源树，共享树中所有的组播组无论源是什么，都有一个共同的根。所有这些树的共享根被称为集合点 (RP)。不像源树中你看到的 S, G 映射那样，在共享树中，你会看见*, G 的映射，因为并不是特别关注源，因此，星号 (*) 代表任何源。

共享树在本质上是单向的。所有的流量从源发送到 RP，流量接着从共享树 RP 向下传送直到到达每一个接收者。然而，这个规则也有例外。例如，如果接收者位于源和 RP 之间，那么接收者就会直接通过源树接收流量。

当通过共享树工作并和 RP 通信后，任何中间的组播路由器可能决定一个到达组播源的最短路径而不是通过 RP 共享树的路径。在这种情况下，一个组播路由器加入源树 (S, G) 并且从共享树中剪枝。最短路径是由路由表决定的。

图 3-7 显示了一个具有 RP 的组播网络。因为组播组中所有的源使用的是同一棵树，组播*, G 树映射为 (*, 224.2.2.2)。共享树概念中的一个难点是所有的组播路由器并不自动地学习新的组播组。在 PIM 稀疏模式下，所有的源使用注册信息到 RP 注册来代表一个新的组播源。所有其他的组播路由器通过查询 RP 来作为客户加入不同的组播组。

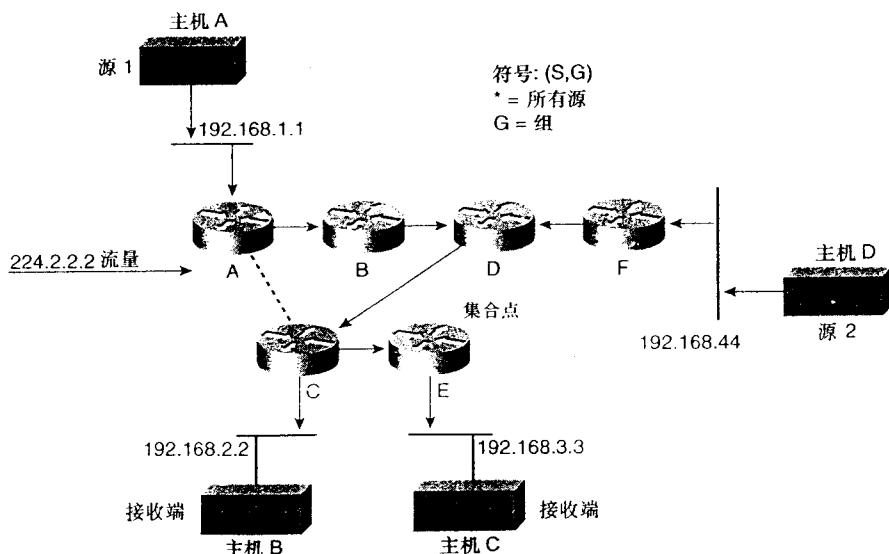


图 3-7 在组播网络中的集合点

共享树和源树都是无环的。在整个拓扑中，随着客户系统加入或者离开组播组，组播数据包会沿着活动的树的枝杈传送。当一个枝杈上所有的接收者全都离开了这个组播组，路由

器就会剪掉这一枝。当更多的客户加入，路由器动态地修改这棵树。

路由器对每一个源保持路径信息。在大型的网络中，随着成百上千的组被监控，需要考虑路由器上内存的消耗和组播设计中组播路由表的大小。共享树本质上需要的内存较小是因为到达 RP 的共同路径。同样，在网络设计中，考虑 RP 的放置位置和组播源的位置以及共享树的尺寸。

3.3.3 组播转发

在正常的单播网络中，所有的转发决定都是基于数据包的目的地址。在组播网络中，路径的决定方式更随意，它随着哪些枝上有活动的客户和哪些枝上没有活动的客户而变化。

在源树中，流量的转发是基于源地址和其他的因素决定的。通常，流量被认为是离开源而不是朝向接收者。

3.3.4 反向路径转发

单播路由数据库建立一棵组播分发树。PIM 选择从接收者到源的反向路径。PIM 使用路由表来决定上游和下游的接口。取决于你使用的是哪种 PIM 模式（稀疏或者密集模式），反向路径转发（RPF）检查可以基于朝向 RP 的分发树或者朝向组播源的分发树。下一节更详细地讨论 PIM 树。RPF 检查帮助确保组播分发树是无环的。

当一个组播数据包被路由器接收后，如图 3-8 所示，路由器对这个数据包执行 RPF 检查。如果 RPF 检查成功，数据包被转发。如果 RPF 检查失败，数据包被丢掉。

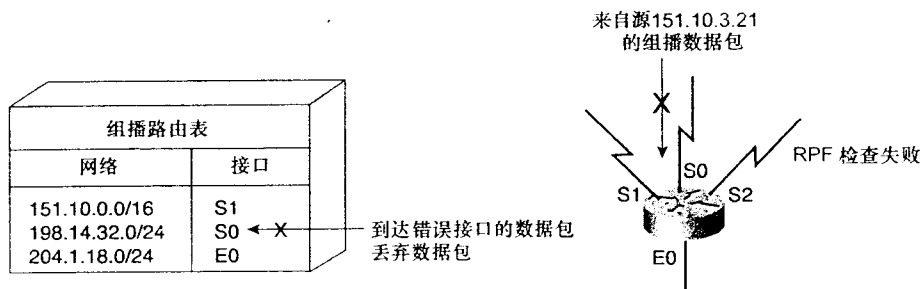


图 3-8 反向路径转发

路由器首先在单播路由表中检查源地址，查看这个数据包是否在反向返回到源的接口上接收到的。如果数据包是在反向返回到源的接口上接收到的，那么检查成功并且数据包被转发。如图 3-8 所示，如果它到达的是其他的接口，那么 RPF 检查失败并且数据包会被丢掉。

3.4 与协议无关的组播

与协议无关的组播 (PIM) 是一种和路由选择协议无关的方法，在整个互连网络中传递

组播数据包。无论你使用的是哪种路由选择协议，包括从静态路由到 OSPF 和 BGP，PIM 使用来自路由信息数据库（RIB）的信息执行组播路由。虽然 PIM 使用单播路由表进行 RPF 检查，但它并不像其他的路由选择协议那样发送和接收路由更新数据包。所有的 PIM 模式在每一个接口的基础上配置。

对于 CCIE 考试，需要了解下面 3 种 PIM 的转发模式：

- PIM 密集模式；
- PIM 稀疏模式；
- 双向 PIM。

3.4.1 PIM 密集模式

PIM 密集模式使用一种推的方式通过网络传输组播数据包。用简单的术语来说，组播路由器通过所有的接口发送组播数据，直到其他的设备告诉它停止传送（剪枝）。

密集模式是一个持续的行为。然而，它会每隔 3min 就向全网泛洪并且必须被剪枝。PIM 密集模式只支持源树，并且不能用于构造一个共享的组播树（注意我们这里的重点是关注树的类型）。

为了配置 PIM 的密集模式，在接口配置模式中使用下面的命令：

```
Router(config-if)# ip pim dense-mode
```

3.4.2 PIM 稀疏模式

PIM 稀疏模式使用一种拉的方式来通过网络传递组播数据包，有活动接收者的网络分支是惟一能够接收组播流量的网段。不同类型的组播路由器都关注于加入或者离开一个组播组，或者需要的时候对流量进行剪枝。

PIM 稀疏模式需要一个 RP。当接收者注册后，数据会沿着共享树向下传递到接收者。每一个组播路由比较到达 RP 地址的度量值和到达组播组的源地址的度量值，如果到达源的度量值更好（着重显示网络中 RP 的位置），那么就会构造 S，G 树。这个树在短时间内可能会走相同的路径，因此也被称为全等路径，如图 3-9 所示。

为了配置 PIM 的稀疏模式，在接口配置模式中使用下面的命令：

```
Router(config-if)# ip pim sparse-mode
```

这是代表 PIM 稀疏模式的一种典型的方法，允许它对于某些操作或者兼容性工作也可以工作在密集模式下。如果你的设计要求你不允许使用密集模式，那么可以发出下面的命令：

```
Router(config-if)# ip pim sparse-mode
```

可以在每一个组播路由器上手动配置 RP，告诉它们数据流如何进行通信，如下面的范例所示：

```
Router(config)# ip pim rp-address (ip#) [(acl#)]
```

可选的访问控制列表限制了所列出的 RP 服务哪些特定的组播组。

---> PIM 源注册报文
-----> 组播数据流

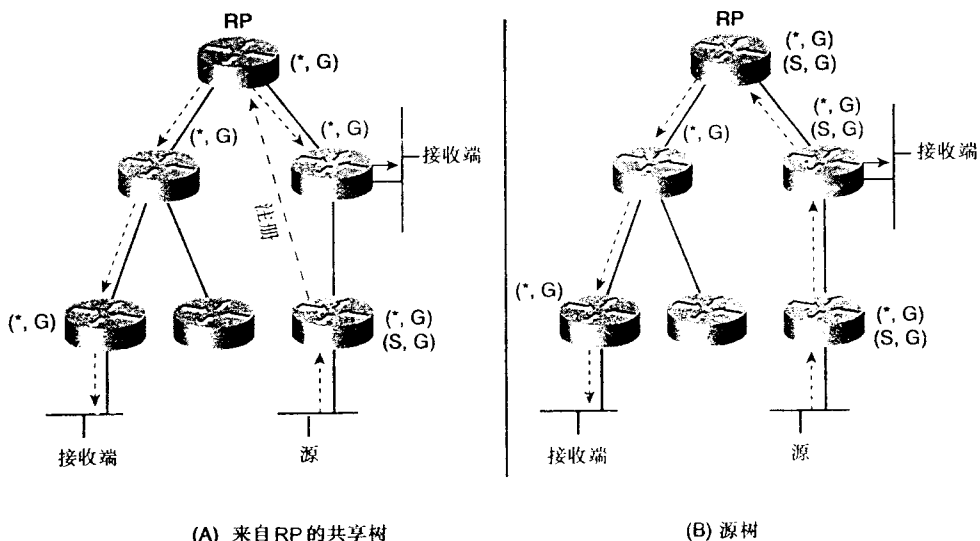


图 3-9 共享树和源树的不同点

如果你的设计场景不允许在稀疏模式下使用源树，你就不能使用 **pim sparse-dense** 命令，相反，使用下面的命令：

```
Router(config-if)# ip pim spt-threshold infinity
```

SPT 是最短路径树算法，它比较源特定的组播树和共享组播树到 RP 的度量值，这个命令关闭在源和 RP 之间的最短路径算法的比较。

3.4.3 双向 PIM

双向 PIM 在先前树建立的学习方法上作了扩展。在向下转发数据包时（RP 到接收者），稀疏模式和双向 PIM 模式没有太大的差别。但是在上游方向上转发数据包时，却有极大的差别。

PIM 稀疏模式不能朝上游方向传输数据包。这将和所有的组播路由器所执行的 RPF 检查相冲突。只有当所有的其他流量在共享树中通过 RP 向下流动时，所有的加入信息才会包含在注册信息中发给 RP。

双向 PIM 选举一个指定的转发路由器（DF）来保持组播拓扑的无环。每一个网段和点对点链路选举一个 DF。这个 DF 负责将适当的组播流量向上游转发。在这个网段中对于 RP 具有最佳路由的路由器成为 DF。

在一个网络中对每一个 RP 都会选举一个 DF。因为选举是基于朝向 RP 的路由的度量值发生的，当处理每一个网段上有多个 RP 时，就有可能发生每一个网段上有多个 DF 的情况。

理方面，有非常多的技术进展，但是对于本章的范围，记住 DF 的选举是非常重要的。

3.5 实验 5：设置基本的组播

考虑图 3-10 所示的拓扑，将所有路由器配置在 VLAN B 和 VLAN 60 内，使得它们可以加入组播组 239.42.42.42。在路由器之间应当没有不必要的组播流量的交换。

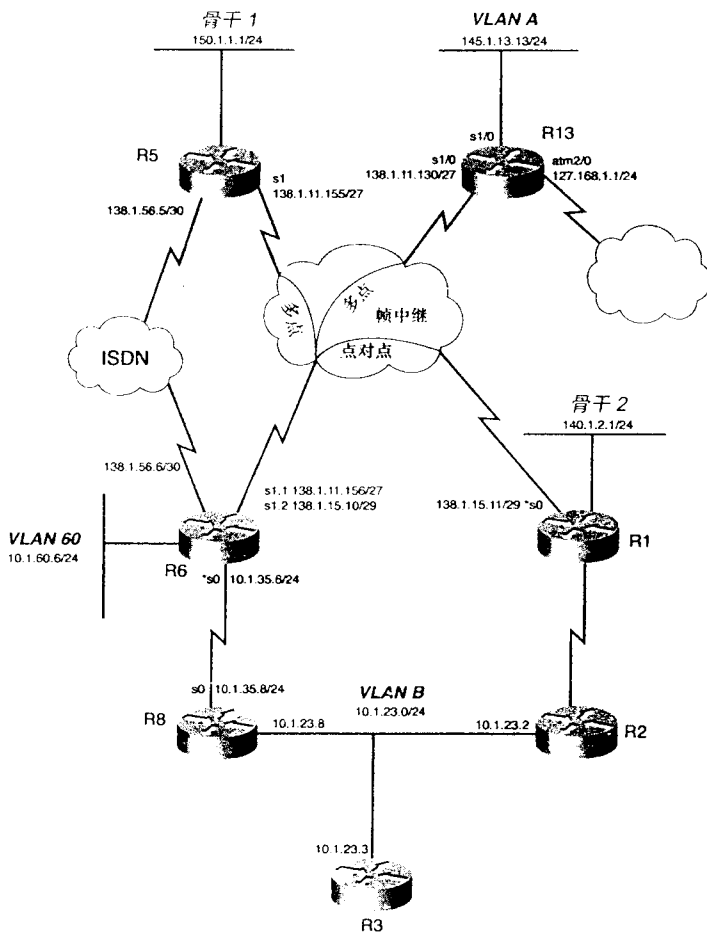


图 3-10 组播实验网络图

3.5.1 实验 5：解决方案

当考虑这种类型的场景时，用的话语是非常关键的。这个场景的事实是它不需要交换不必要的流量，这就意味着不应当使用 PIM 密集模式。

配置中的下一个问题就是应当将 **pim** 语句放在哪里，谁应当成为 RP。很显然，所有的路由器都需要在全局模式下配置 **ip multicast-routing** 命令。而且，在远程终点之间的所有接

口都必须配置 **ip pim sparse-mode interface** 命令。

在我们这个范例中选择哪台路由器作为 RP? 在这个场景的问题中，没有什么优先顺序，这个拓扑不是足够大来区分哪台路由器适合做 RP。在一个活动的组播网络中，当查看功能性选择，例如 SPT 算法和设计整体的流量流动时，RP 的位置是非常关键的。

下一步是使用 **ip pim rp-address** 命令配置所有其他的路由器。和它有关的一个常见的问题就是 **rp-address** 命令是否必须放置在充当 RP 的实际的路由器上。答案是无关紧要。如果显式地配置了，路由器知道了。如果没有显式地配置，当其他的路由器发送 PIM 加入和剪枝信息时，路由器自动获知可承担 RP 的角色。

注意：关于这个概念检查思科 IOS 软件版本的注解。新版本的思科 IOS 软件实际上需要用户在 RP 上配置 **rp-address** 命令。

为了测试这个场景并完成加入，需要在两个 VLAN 中选择某些接口并且发出 **ip igmp join-group 239.42.42.42 interface** 命令。当发出这些命令后，可以 ping 这个组播组并且接收来自每一个加入的路由器的响应。

3.5.2 实验 5: 配置

范例 3-1 在路由器上使用 show running-configuration 编辑的命令表项

```
R2
ip multicast-routing
!
interface ethernet 0
ip pim sparse-mode
ip igmp join-group 239.42.42.42
!
ip pim rp-address 10.1.23.3

R3
ip multicast-routing
!
interface ethernet 0
ip pim sparse-mode

R6
ip multicast-routing
!
interface ethernet 0
ip pim sparse-mode
ip igmp join-group 239.42.42.42
!
interface serial 0
ip pim sparse-mode
!
ip pim rp-address 10.1.23.3

R8
ip multicast-routing
!
```

(待续)

```
interface ethernet 0
 ip pim sparse-mode
 ip igmp join-group 239.42.42.42
!
interface serial 0
 ip pim sparse-mode
!
ip pim rp-address 10.1.23.3
```

3.6 组播帧中继

在帧中继网络上运行组播类似于在任何其他的网络上运行组播，除了在实际的网络中你可能注意到某些显著的不同点。在点对点的帧中继接口上，例如路由选择协议和其他的选项趋向于工作“正常”。

在多点接口上，考虑不同之处。帧中继是一个非广播的多点访问网络。单词“广播”非常类似于以太的 MAC，也代表组播数据包。为了使得路由选择协议工作，需要使用带 **broadcast** 参数的 **frame-relay map** 命令。

还需要考虑帧中继接口是如何处理组播流量的。在一个物理接口上，有两种接口队列——一种处理正常的流量，另外一种处理广播的流量。广播队列是一个严格优先级的队列，通常处理重要的流量类型，例如路由选择协议更新数据包。帧中继接口没有一种方法能够区分出组播流量流（例如视频流或者音频流）和其他的组播流（例如 OSPF 路由选择协议）。

进入广播队列的流量也默认是进程交换的，而不是快速交换方式。

在一个实验性的网络中，没有人关心这一点。在现实世界中，对带宽敏感的视频流独占严格优先级队列，而将其他的“正常”流量饿死，这是一种很严重的事件。为了解决这个问题，必须指引路由器处理非路由的组播流量，就像它和接口上任何其他“正常的”流量一样。

帧中继处理组播也可能引起其他问题。通常，PIM 工作在接口的基础上。在一个正常的多点帧中继环境下，从同一个物理接口出去可能有多条路径。当它接收到加入或者剪枝的消息时，这可能导致问题，即一台路由器的剪枝消息可能切断到其他路由器的流量。

许多技术上的不同点都超出了本书的范围（你可以参考在本章末尾列出来的一些参考资料，更进一步地了解）。

对于基于实验的场景和许多实际生活的场景，需要注意帧中继接口关于组播流量的不同处理方式。

ip pim nbma-mode 接口命令允许用户完成这个任务。这个命令只对 PIM 的稀疏模式工作，因为它依赖于 PIM 的加入消息来指明流量的类型。除了 **ip pim sparse-mode** 命令，这个命令还要额外发出。这个命令在其他的功能中允许组播流量通过帧中继的接口快速交换。观察你的 CCIE 实验的场景和拓扑。

3.7 组播 TTL

当组播数据包通过路由器时，TTL 会减少。如果 TTL 少于或等于 0，这个数据包就会丢

掉。如果 TTL 大于 0，它可能会和在路由器上手动配置的 TTL 阈值进行比较，如果数据包的 TTL 值大于阈值，它就会转发。

通常，TTL 的阈值只在组播或者自治系统的边界路由器上设置，以确保流量不会穿过不应通过的边界。

为了设置 TTL 的阈值，使用 **ip multicast ttl-threshold ttl-value** 接口命令。

3.8 组播边界

作为一个更严格的控制，如果不想让组播流量通过路由器的某些边界，可以设置一个组播边界。可以通过一个标准的 IP 访问控制列表来对某些组播组进行组播流量边界的限制。

ip multicast boundary (acl#) 接口命令允许用户建立组播边界。组播边界在本质上是双向的。也可以在命令中增加一个参数 **filter-autorp** 来过滤自动 RP 信息中的组播范围宣告。自动 RP 在下面讨论。

```
Router(config-if)# ip multicast boundary 1
Router(config)# access-list 1 deny 239.0.0.0 0.255.255.255
Router(config)# access-list 1 permit 224.0.0.0 15.255.255.255
```

3.9 PIM 自动 RP

在每一个组播路由器上我们不需要手动配置 RP，相反，RP 可以自动地宣告它自己。这在大型的网络中尤其有用。

自动 RP 使用 224.0.1.39 和 224.0.1.40 组播组来发送信息。自动 RP 通过 PIM 的密集模式将这个信息泛洪出去。为了使得自动 RP 正常工作，路由器必须使用 **ip pim sparse-dense-mode** 接口命令。如果没有密集模式的能力，RP 将永远不会被学到。

自动 RP 功能也包括映射代理。映射代理侦听 RP（通过 224.0.1.39 组播组），并且通过 224.0.1.40 组播组以发现信息的格式发送 RP 到组的映射。

映射代理通过网络从候选的 RP 接收信息。映射代理负责建立一致的组播组到 RP 的映射，并且通过密集模式泛洪把这些宣告发送给所有的组播路由器。

在帧中继的环境下使用自动 RP，必须注意一些问题。所有候选的 RP 必须有一个 **map** 语句或者其他的语句连接到映射代理。所有的映射代理必须连接到所有的组播路由器。

为了将路由器配置为 RP 并且宣告出去使得其他的组播路由器能够自动地学习到它，在全局配置模式下使用 **ip pim send-rp-announce source intf scope ttl-value** 命令。

为了作为一个映射代理起作用，使用 **ip pim send-rp-discovery scope ttl-value** 全局命令。

通常，环回地址用于 RP 的地址（源接口）。环回接口必须能够通过内部网关协议（IGP）可达，并且在接口上启用了 PIM。选择环回接口是因为它们永远是 up 的接口，因此，可以通过任何其他“up”的接口可达。

anycast RP

一种新的通过互连网络控制组播 RP 稳定性的方法叫做 anycast RP。关于这种方法有新的概念和协议。anycast RP 的要点是一个单独的 IP 地址通过网络静态地配置为 RP。

这个 IP 地址可以同时多台路由器上配置（这个概念让许多人阐述起来都很模糊）。是的，你可以在多台路由器上配置同一个 IP 地址。关于 IP 地址的一件有趣的事，特别是对于 /32 路由，就是网络中所有的路由表并不关心 IP 在哪里存在。所有的路由表从 RIB 中取信息，而 RIB 是通过路由选择协议交换所得。路由选择协议基于度量值来决定 IP 的可达性。如果多台路由器宣告了同一个 IP 网络，最佳的路径是基于度量值进行选择。没有路由器真正知道路由在哪里或者比较路由更进一步的信息。组播发送者和接收者基于路由度量值加入离它们最近的 RP。

运用这个通常的概念，需要理解基本的路由功能是如何运作的以及多个 IP 地址的存在是有帮助的。你需要考虑一个额外的协议。

通常是服务提供商的域间组播类型设计，组播源发现协议（MSDP）在这个场景中确保所有配置的 RP 含有组播源和组播组的相同的基本信息。

MSDP 会话运行在所有的 RP 路由器之间。正如图 3-11 所示，一个 IP 网络可以有多个 RP 存在。整个网络中的每一个组播路由器都静态地配置 RP。

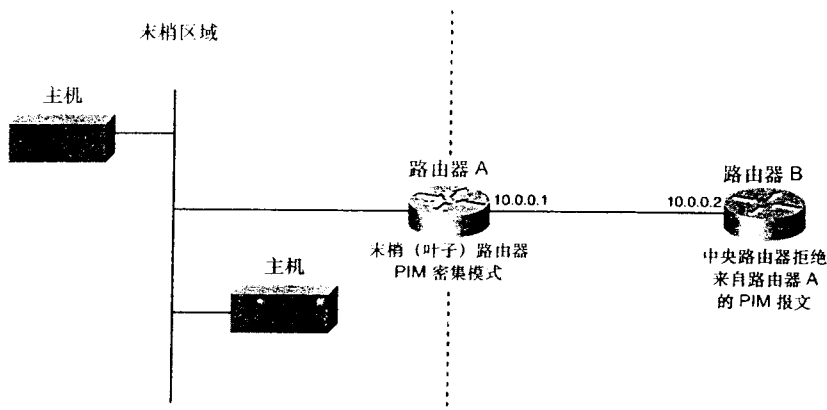


图 3-11 Anycast RP 图

每一个 RP 的路由器都有一个环回地址，这个 IP 地址被认为是 RP 的地址。而且，每一台路由器有其他 IP 地址来唯一地识别它。这第二个 IP 地址和 MSDP 成为对等体。在大型的环境中，可以在对等体之间配置 MSDP 的全冗余。

MSDP 对等体之间采用的是 TCP 会话，并且和所有其他的对等体交换任何新的源活动信息（SA）。列出的命令是使 MSDP 运作起来的最小配置。这个最小的配置列在这儿不是将技术缩小化，而是因为理论通常面向的是服务提供商，而不是面向典型的企业。这个内容对于准备 CCIE 路由和交换实验的考生来说，还需要进一步的细心考察。

范例 3-2 中列出的命令在每一个 RP 上建立一个共享的（10.1.1.1）和独立的（10.0.0.101 和 10.0.0.102）IP 地址。ip msdp 命令指定对等体的 RP，以及哪个接口是所有信息的源和起

始者 ID。这可以在每一台路由器的路由表中避免混淆以及避免路由表的明显不同。

范例 3-2 在 anycast RP 路由器上配置 MSDP

```
RP1
interface loopback 0
 ip address 10.1.1.1 255.255.255.255
!
interface loopback 1
 ip address 10.0.0.101 255.255.255.255
!
ip msdp peer 10.0.0.102 connect-source loopback 1
ip msdp originator-id loopback 1

RP2
interface loopback 0
 ip address 10.1.1.1 255.255.255.255
!
interface loopback 1
 ip address 10.0.0.102 255.255.255.255
!
ip msdp peer 10.0.0.101 connect-source loopback 1
ip msdp originator-id loopback 1
```

3.10 实验 6: 设置帧中继组播路由

使用和图 3-10 相同的网络，将 VLAN A 和 Backbone 1 配置为加入组播组 225.3.3.3。路由器 13 需要成为所有组播组的 RP，除了管理性范围的地址，但是不应当在任何其他的路由器上显式地配置。路由器 5 需要成为一个管理性范围的地址的 RP。

确保组播数据包不要进入 VLAN B 或者其他的网络。这些其他的网络可能运行不同的组播组。

3.10.1 实验 6: 解决方案

对于所有的 CCIE 场景，每一句措辞都要注意。在 VLAN A 和 Backbone 1 上运行组播等于告诉你路由器 5、路由器 6 和路由器 13 必须参与组播网络，当然，也就是帧中继的多点网络的接口。

因此，除了在这 3 台路由器上运行 **ip multicast-routing** 命令，帧中继的接口还必须运行 **ip pim nbma-mode** 命令。

其次，你意识到必须使用 PIM 的稀疏模式，这是因为有 RP。然而，因为你必须使用自动的 RP 命令，也需要 PIM 的密集模式。每一个帧中继接口（组播路由器之间的链路）将被配置 IP PIM 的稀疏密集模式。

必须建立访问控制列表来过滤某些组播网络。在路由器 13 上，管理性范围的地址不能作为 RP 使用。

```
access-list 13 deny 239.0.0.0 0.255.255.255
access-list 13 permit 224.0.0.0 15.255.255.255
```

那个访问控制列表拒绝管理性范围的地址但是允许所有其他的地址。在路由器 5 上，正好相反。

```
access-list 5 permit 239.0.0.0 0.255.255.255
```

这个访问控制列表隐式地拒绝所有其他的组播地址。

接着每一台路由器使用 **ip pim send-rp-announce src-intf scope 16 group-list acl#**全局命令，开始宣告它自己成为访问控制列表中组播组的 RP。

路由器 6 是多点的帧中继接口，也是路由器 5 和路由器 13 之间的流量必须加入的接口。记住，这是使用映射代理（组播 RP 中继）的最好方法。使用 **ip pim send-rp-discovery scope 16** 全局命令来使得它成为映射代理。

最终，这个场景需要在帧中继云组播网络和下面的其他网络之间配置边界。

```
access-list 6 deny any
ip multicast boundary 6
```

问题是在哪里配置 **multicast boundary**？答案是无论你放在哪里，确保不会过线。在路由器 6 上的 3 个其他接口服务于其他网络：以太接口、到路由器 8 的串行链路和另一个到路由器 1 的帧中继子接口。需要在这里面的每一个接口上放置 **multicast boundary** 命令。

3.10.2 实验 6：配置

为了配置路由器的命令表项，参考范例 3-3 来了解 **show running-configuration** 是如何在路由器上工作的。

范例 3-3 在路由器上使用 show running-configuration 编辑的命令表项

```
R5
ip multicast-routing
!
access-list 5 permit 239.0.0.0 0.255.255.255
!
ip pim send-rp-announce ethernet 0 scope 16 group-list 5
!
interface ethernet 0
 ip pim sparse-dense-mode
 ip igmp join-group 225.3.3.3
!
interface serial 1
 ip pim sparse-dense-mode
 ip pim nbma-mode
!

R6
ip multicast-routing
!
ip pim send-rp-discovery scope 16
!
access-list 6 deny any
!
interface serial 1.1 multipoint
```

（待续）

```
ip pim sparse-dense-mode
ip pim nbma-mode
!
interface serial 1.2 point-to-point
ip multicast boundary 6
!
interface ethernet 0
ip multicast boundary 6
!
interface serial 0
ip multicast boundary 6

R13
ip multicast-routing
!
access-list 13 deny 239.0.0.0 0.255.255.255
access-list 13 permit 224.0.0.0 15.255.255.255
!
ip pim send-rp-announce ethernet 1/0 scope 16 group-list 13
!
interface ethernet 1/0
ip pim sparse-dense-mode
ip igmp join-group 225.3.3.3
!
interface serial 1/0
ip pim sparse-dense-mode
ip pim nbma-mode
```

3.11 组播加入

从实验 5 的解决方案（和实验 6 的隐含含义），我们了解了 `ip igmp join-group mcast#` 接口命令。为了确保路由器（实验）确实响应组播和加入组播组，必须键入像 `igmp` 这样的命令。然而，它还有什么用处？

在现实生活中，可以使用它来给一个局域网段提供组播组，使得客户能够解析组播但是不能通过 IGMP 发起组成员的信息。当接口参加了组播组，组播流量会转发到那个局域网段。在 CCIE 的实验场景下，要注意这个功能的应用。

在实际生活中，这一点会有问题。因为数据包是被路由器处理后再发送出去，它们只能被进程交换。总体上来说这会降低路由器的性能，并不是一个很好的做法。然而，在本实验中，你不必有这样的顾虑。

但是等一下——有一个更好的方法来完成这个任务！考虑 CCIE 实验场景中用的措辞，寻找这样的话，如转发组播流量到一个局域网段，但是不从客户端接收 IGMP 信息。另外，查看参考资料确保路由器并不处理组播数据包。在做这件事时，优化路由器的处理速度。

这意味着什么？

如果使用 `ip igmp static-group mcast#` 接口命令，它正好完成那个任务。因此，观察实验场景中的措辞来推断你需要知道什么。

使用 `ip igmp static-group` 命令，组播数据包被自动地快速转发而无需和 RP 交互。

另外一种要观察的情况是将进入的组播流量转换成某种类型的数据包，例如广播。这对于不能有效地接收组播的客户是有用的。缺点是当转换成广播数据包后，更多的工作站可能会接收（并且处理）组播流量，而实际上可能并不想接收。

这个转换过程使用组播的 `helper address` 和一个 UDP 端口实现。整个操作类似于动态主机配置协议（DHCP）中继的工作过程。首先，会选择一个惟一的、高的 UDP 端口号，并且建立一个过滤的访问控制列表，使用下面的命令：

```
Router(config)# ip forward-protocol udp 4400
Router(config)# access-list 101 permit udp any any eq 4400
Router(config)# access-list 101 deny udp any any
```

接着，它们绑定在一个局域网接口的转换过程中。为了泛洪这个信息，必须在下面的命令中使用 PIM 密集模式：

```
Router(config-if)# ip pim dense-mode
Router(config-if)# ip directed-broadcast
Router(config-if)# ip multicast helper-map broadcast 225.4.4.4 101
```

这些命令将组播组 225.4.4.4 绑定到访问控制列表 101 中指定的 UDP 端口，并且对那个接口执行将组播组转换成广播数据包的转换过程。在这里注意 `ip directed-broadcast` 命令。在思科 IOS 软件 12.0 或以后的版本中默认地不允许子网级别的广播。

如果这样做，可能潜在地会开启一个安全的隐患。但是再次说明，在 CCIE 的实验中，你不总是需要顾虑这些问题。

3.12 实验 7：组播加入

再次考虑图 3-10 所示的网络拓扑。Backbone 2 的客户不能发送 IGMP 加入信息，但是需从源 10.1.60.6 侦听组播流 225.9.13.5。在帧中继的云团上不需要通过组播流量。路由器 1 应当优化来处理组播流量因为它已经负担过重。

3.12.1 实验 7：解决方案

我们在这里讨论新的一些事情是很重要的。我们将提供的信息分开来看。源是 10.1.60.66（路由器 6 上的 VLAN 60），运行在组播组 225.9.13.5 上。目的是 Backbone 2 上的客户（远离路由器 1）。

记住组播树在穿越网络时遵循单播的最佳路由的逻辑。这也就告诉用户在路由器 6 和路由器 1 之间的帧中继线路是最好的路径。然而，不允许以那种方式工作。谁说过 CCIE 的实验很容易？

所有的路由器都运行 `ip multicast-routing` 全局命令。

一次只处理这个场景中的一个步骤。可以将路由器 6 设置为 RP，这时 E0/0 可以加入 IGMP 的组播组 225.9.13.5。在类似这样的场景的情况下，对只采用稀疏模式还是采用稀疏密集模式的 PIM 来说，没有好坏之分，通常采用后者（参看范例 3-4）。

范例 3-4 在路由器上使用 show running-configuration 编辑的命令表项

```
R6(config-if)# ip pim sparse-dense-mode
R6(config-if)# ip igmp join-group 225.9.13.5
```

通常，需要在每一台路由器上都定义 RP。

```
R6(config)# ip pim rp-address 10.1.60.6
```

在路由器 6 和路由器 1 之间的所有路由器（不通过帧中继云）都需要启用组播和 PIM 来转发流量。

路由器 1 需要将它的以太网段加入组播组，但是也需要优化。一个没有经过优化的路由器会花费大量的时间处理数据包并且使用它不应该使用的内存。这使我们更趋向于使用 **static-group** 命令，而不是 **join-group** 命令。客户不能使用 IGMP 的加入消息这一事实也告诉我们需要这样做（参看范例 3-5）。

范例 3-5 在路由器上使用 show running-configuration 编辑的命令表项

```
R1(config-if)# ip pim dense-mode
R1(config-if)# ip igmp static-group 225.9.13.5
```

最后一件必须要考虑的事情就是组播数据包的路由。本章还没有重点强调组播路由的重要性，然而，在你还没有在 CCIE 的实验中看到它们，面对它们之前，需要思考这些事情并且考虑如何解决这些问题。

组播数据包会自动地进行 RPF 检查，这是基于从所期望的接口可以返回到组播发送者所在的 IP 源的接口。如果一个组播数据包到达的接口不是可以返回到发送者 IP 的那个接口的话，这个数据包会被丢掉。因为你是四处发送组播数据包，所以在这个场景下，所有的接口都需要启用组播功能。

毫无疑问，需要“调整”路由器 1 来实现路由。是否需要对其他的路由器这样做完全取决于 IP 路由表需要的下一跳是什么。在路由器 1 上，可以使用静态组播路由来手动调整选择，如下

```
R6(config)# ip mroute 10.1.60.6 255.255.255.255 [protocol as-number] {rpf IP# | intf}
[(admin. Distance)]
```

在路由器 1 上，RPF IP 是路由器 2 上的地址；或者类似于 IP 静态路由，可以路由到一个接口上。**ip mroute** 命令允许用户在这个命令的地址部分设置组播的源 IP 地址。

3.12.2 实验 7：配置

本小节展示了在这个实验的解决方案中路由器的配置（参看范例 3-6）。

范例 3-6 路由器上使用 show running-configuration 编辑的命令表项

```
R6
ip multicast-routing
!
```

（待续）

```
interface ethernet 0
 ip pim sparse-dense-mode
!
interface serial 0
 ip pim sparse-dense-mode
!
!

R8
ip multicast-routing
ip pim rp-address 10.1.60.6
!
interface serial 0
 ip pim sparse-dense-mode
!
interface ethernet 0
 ip pim sparse-dense-mode
!

R2
ip multicast-routing
ip pim rp-address 10.1.60.6
!
interface ethernet 0
 ip pim sparse-dense-mode
!
interface serial 1
 ip pim sparse-dense-mode
!

R1
ip multicast-routing
ip pim rp-address 10.1.60.6
!
interface serial 1
 ip pim sparse-dense-mode
!
interface ethernet 0
 ip pim dense-mode
 ip igmp static-group 225.9.13.5
!
ip mroute 225.9.13.5 255.255.255.255 serial 1
!
```

3.13 控制组播

当我们进行到控制组播网络时，我们面临几个问题并且在几个点上可以控制它。当和速率限制有关时，这些控制额外重要。如何对组播的流量进行限速？简短的答案是：根据执行限速的设备的性能，有几种方法。

在 Catalyst 3550 上，可以基于每一个端口进行控制，使用一种称为风暴控制的方法。为了确保组播的流量不占用一个特定接口（或者以太网通道组）10%以上的带宽，发出下面的命令：

```
Cat3550(config-if)# storm-control multicast level 10
```

在路由器上,包括广域网的链路,可以通过发出 **ip multicast rate-limit(in | out)[group-list (acl#)] [source-list (acl#)] interface kbit/s** 命令来实现限速。如果没有配置这个命令,那么就没有执行限速的功能。如果配置了这个命令,但是没有设置带宽,那么默认的值就是 0,意味着不允许组播流量。

3.13.1 快速交换

回忆一下,当我们使用 **ip igmp static-group** 命令时要特别注意,确保组播数据包可以通过路由器快速交换。现在,考虑一下如果场景要求你将快速交换关闭掉,你需要做什么。

这个过程类似于使用单播路由的过程。快速转发包括使用一个路由缓存来保存最近使用的路由选项并且加速随后的路径选择。为了关闭单播,使用 **no ip route-cache interface** 命令。在组播的环境下,逻辑是相同的。**no ip mroute-cache interface** 命令完成这个功能。

3.13.2 组播末梢

当构造 PIM 树时,在每一个方向上只有一个路径的枝杈(段)。类似于在单播世界中的末梢网络,可以在一个末梢区域中控制进入或者流出这个区域的流量,这是因为没有其他的路径选项。图 3-12 展示了一个组播的末梢网络。

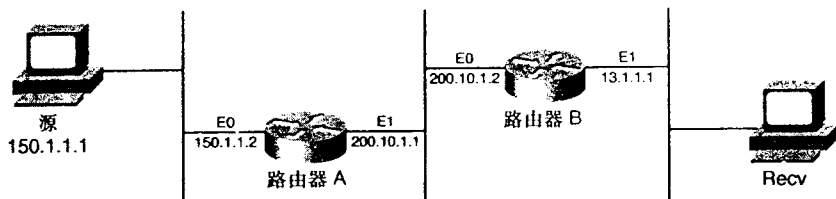


图 3-12 组播末梢网络

在末梢路由器(路由器 A)上,朝外的链路使用 **ip igmp helper-address 10.0.0.2 interface** 命令。这将转发所有的 IGMP 信息给中心的路由器,而无需它自身处理典型的 IGMP 报告和查询信息,就可使得 PIM 工作。

在中心的路由器(路由器 B)上,面向末梢的链路将会有有一个 **filter** 命令调用一个访问控制列表,来停止路由器 A 和路由器 B 之间的 PIM 机制。

```
RouterB(config)# access-list 11 deny host 10.0.0.1
RouterB(config-if)# ip pim neighbor-filter 11
```

经过这些配置步骤后,路由器 A 后的任何主机的 IGMP 信息都会转发到路由器 B。路由器 A 确实参与任何 PIM 树,但是基于 **filter** 命令也可以不参与。

3.13.3 负载分担或者不连续的组播网络

组播并不提供负载分担或者负载均衡的功能(从某种角度上讲)。它执行 RPF 检查,答
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

案要么是 yes 要么是 no，没有这种灰色的区域。如何在等长路径上负载分担流量？如何在两台路由器之间发送组播流量，而两者中间的网络并不支持组播？就像在 CCIE 世界中的任何事情一样，你需要超乎常规去考虑一些问题

一个简单的术语：隧道。不要忘记隧道可以作为整个解决方案的一部分。隧道可以提供一种简单的方法将其他的非路由的流量封装后，接着从点 A 传输到点 B。通过将组播数据包（或者其他流量）封装到一个 IP 单播的通用路由封装（GRE）数据包里，被封装的数据包就会遵循单播的属性。使用了这个功能，中间的路由器只会看到一个目的到那里的单播的 IP 包。单播的数据包可以实现负载均衡，这是因为路由器现在只关心目的，而对组播世界中的源或者组信息都不再关心。

3.14 实验 8：高级组播传输

按照图 3-10 所示的网络，在 VLAN A、VLAN B 和 VLAN 60 上的源之间对于组 226.7.6.5 启用组播流量。广域网不能够直接携带组播流量。确保 VLAN B 不使用超过 2 Mbit/s 的带宽来携带组播流量。

3.14.1 实验 8：解决方案

和其他的配置一样，必须在所有适当的路由器上发出 `ip multicast-routing global` 命令。所有的以太网接口也需要合适的 `ip pim sparse-dense-mode` 命令。

为了允许组播流量间通过帧中继云，在路由器 13 和路由器 6 上建立隧道，使用下面的命令：

```
R13(config)# interface Tunnel 0
R13(config-if)# ip unnumbered Serial 1/0
R13(config-if)# ip pim sparse-dense-mode
R13(config-if)# tunnel source Serial 1/0
R13(config-if)# tunnel destination 138.1.11.156
```

这将建立隧道并且允许封装组播数据包。

然而，还必须完成其他的步骤，以忽略典型的 RPF 检查（它把 Serial 1/0 视为路径），配置如下：

```
R13(config)# ip mroute 10.1.60.0 255.255.255.0 Tunnel 0
```

因为源在 VLAN 60 上，没有必要在路由器 6 上忽略组播路由（RPF）。

对于 VLAN B 的限制，必须在 Catalyst 3550 上进行配置的修改。当然，所配置的数量取决于这个接口是 10 Mbit/s 还是 100 Mbit/s！记住广播风暴是一个基于百分比的算法。

```
Cat3550(config)# interface intf
! Note: 10 megabit Ethernet interface
Cat3550(config-if)# storm-control multicast level 2

Cat3550(config)# interface intf
! Note: 100 megabit Ethernet interface
Cat3550(config-if)# storm-control multicast level 20
```

如果你的实验场景指定一个组播源，那么可以在路由器上配置速率限制，指定一个特定的源 IP 作为限速的条件。你的实验场景决定了用哪种方法来配置。

3.14.2 实验 8: 配置

本实验还演示了如何在路由器上使用 `show running-configuration` 来编辑命令表项。

范例 3-7 在路由器上使用 `show running-configuration` 编辑的命令表项

```
R8
ip multicast-routing
!
interface ethernet 0
 ip pim sparse-dense-mode
 ip igmp join-group 226.7.6.5
!
interface serial 0
 ip pim sparse-dense-mode

-----

R6
ip multicast-routing
!
interface ethernet 0
 ip pim sparse-dense-mode
 ip igmp join-group 226.7.6.5
!
interface serial 0
 ip pim sparse-dense-mode
!
interface tunnel 0
 ip unnumbered serial 1.1
 ip pim sparse-dense-mode
 tunnel source serial 1
 tunnel destination 138.1.11.130
!

-----

R13
ip multicast-routing
!
interface ethernet 1/0
 ip pim sparse-dense-mode
 ip igmp join-group 226.7.6.5
!
interface tunnel 0
 ip unnumbered serial 1/0
 ip pim sparse-dense-mode
 tunnel source serial 1/0
 tunnel destination 138.1.11.156
!
ip mroute 10.1.60.0 255.255.255.0 tunnel 0

Cat3550
interface 0/8
 description Link to R8.VLAN B
 storm-control multicast level 20
```

注意：不需要添加其他的 VLAN B 的路由器，因为按照图它们将不会路由组播流量。

3.15 DVMRP 组播路由

因为组播路由和单播路由处理不同的拓扑，所以策略要求 PIM 遵循组播的拓扑来构造一棵无环的分发树。PIM 可以使用任何单播路由协议来作为 RPF 检查的参考，但是组播特定的协议可以更好地构造组播分发树。

使用距离向量型的组播路由协议（DVMRP），思科的路由器可以和其他的路由器交换 DVMRP 的单播路由或者基于组播路由的机制。PIM 也可以对 RPF 的信息使用这一点。这里很重要的声明是 DVMRP 是一种单播路由的路由协议，用在组播路由拓扑中。它不是一种直接通过网络路由组播的方法，也不是对通常的单播路由实现一种更好的路径的方法。

思科的路由器可以交换 DVMRP 路由，但是实际上不会通过由 DVMRP 产生的决策来路由组播数据。然而，运行 DVMRP 允许 PIM 使用组播的拓扑，这允许稀疏模式的 PIM 在整个因特网的拓扑中使用。MBONE 是这种类型的另外一种应用，参与者使用组播路由协议在不连续的网络上构造有效的组播拓扑。

一旦 DVMRP 单播路由完成后，对于 DVMRP，学习到的路由会保存在单独的 RIB 里。PIM 偏向于通过 DVMRP RIB 学习到的路由，而不是通过其他单播路由协议学习到的 RIB 中的路由。

DVMRP 单播路由可以运行在任何接口类型上。使用 GRE 隧道，存在一种特殊的操作模式来指明 PIM 拓扑构造所使用的隧道。在隧道接口下，发出下面的这些命令：

```
Router(config)# interface tunnel 0
Router(config-if)# tunnel mode dvmrp
```

就像以前所提到的，这并不会启用真正的组播路由的决策，但是允许 PIM 根据更多的组播拓扑信息来建立一个树型的构造决策。总之，路由器需要知道哪些接口用来缓存 DVMRP 的信息以构造组播的拓扑。可以通过发出下面的这些命令来完成这个任务：

```
Router(config)# interface intf
! Any interface
Router(config-if)# ip dvmrp unicast-routing
```

默认情况下，在任何一个接口上只能交换 7000 个 DVMRP 路由。这些接口是特别启用了 DVMRP 的接口或者启用了 DVMRP 的隧道，通过接口可以发现 DVMRP 的邻居。可以通过使用 **ip dvmrp route-limit limit-value** 全局命令来改变这个默认值。而且，可以通过汇总地址来增强路由拓扑。这是一个接口特定的命令：

```
Router(config)# interface intf
! Any interface
Router(config-if)# ip dvmrp summary-address net-addr net-mask [metric value]
```

DVMRP 自动汇总成有类的边界。然而，**ip dvmrp summary-address (mcast-net#) (mask) interface** 命令允许用户超越它的默认行为。而且，**no ip dvmrp auto-summary interface** 命令允许用户将它关掉。

记住在某些潜在的情况下需要使用 **multicast static route** 命令来凌驾和进一步操作路由表。就像其他的路由协议一样，也可以在更复杂的拓扑下给度量值应用偏移量，使用 **ip dvmrp**

metric-offset [in | out] increment 命令。

在 CCIE 的实验中，你可能会碰到更复杂的一些实验场景。只要记住单播路由和处理的观念。毕竟，组播路由和处理是在单播的基础上作了镜像，但是延深度很大。其他要记住的一些重要的点包括可以在 PIM 的单播拓扑中使用 **ip dvmrp default-information originate** 命令建立一条默认路由，并且可以使用 **ip dvmrp accept-filter access-list [distance | ip neighbor-list access-list]** 命令来特别过滤或者修改路由。

3.16 PIM 版本 2

到目前为止我们已经讨论了组播的一些基本操作，特别是 RP，它和 PIM 的版本 1 一起工作。PIM 的版本 1 有一种有趣的方法通过单播的路由结构构造树和路由组播。PIM 版本 2 对这一点作了增强。记得我们先前讨论的自动 RP，这是一个思科专有的特性。它是一个很好的特性，每一个人都很喜欢它，但是只有思科的设备能够理解它。PIM 版本 2 有一个自主的路由器（BSR），它可以提供相同类型的功能和宣告特性。PIM 版本 2 和 PIM 版本 1 彼此之间不会自动兼容。

PIM 版本 1 和 RP 工作在活动模式下。在这种拓扑中，可以有一个或者多个 RP，但是它们都处在活动状态下，处理信息和组播树并且路由信息。使用 PIM 版本 2，现在有一个备份的自主 BSR（RP）的概念。随着备份的出现，需要保持拓扑运行的信息的数量比以前少。很多的细节知道以后会非常好，但是已经超出了本章的范畴。

如果你有 PIM 版本 1 路由器，不要使用 BSR。相反，使用自动-RP（如果都是思科的路由器）的特性或者手动的 RP 分配。使用 BSR，可以在一个组播域内使用多个 BSR 的候选者。具有最高优先级的路由器优先，但是这种设计允许在整个网络内实现容错的概念。

BSR 可以和自动 RP 一样处理宣告特性。也有类似发现信息的特性，但是在整个网络内 BSR 并不需要一定是 RP。

为了在路由器上配置 PIM 的版本，使用 **ip pim version (1 | 2)** 全局命令。

当选择 BSR 时，使用 **ip pim bsr-candidate src intf hash-length# priority#** 全局命令。hash-length 值和长度有关，主要用于消息交换过程中的哈希运算。虽然不需要，我们还是推荐在所有的 BSR 的候选者上将这个值配置为相同的数。高优先级值的路由器成为 BSR。

为了确保 PIM 版本 1 和 PIM 版本 2 彼此不互相干扰，或者建立两个不同的 PIM 版本 2 的域，设置组播的边界。可以有两种方法。对于 PIM 版本 2，使用 **ip pim border interface** 命令，使得 BSR 的信息不会穿过。对于 PIM 版本 1，使用 **ip multicast boundary interface** 命令，使它与匹配 224.0.1.39 和 224.0.1.40 的访问控制列表相关联来防止自动 RP 组播组穿过那个接口。

除了这些，设置路由器使得它成为某些或者所有的组播组的 RP 的候选者，使用 **ip pim rp-candidate (src intf) (ttl#) [group-list (acl#)]** 全局命令来将路由器设置为一个 RP 的候选者。

3.17 实验 9: PIM

再次使用图 3-10 所示的网络，Backbone 1 和 VLAN A 使用 PIM 版本 1。路由器 5 需要

自动地宣告它自己成为 RP。VLAN 60 和 VLAN B 使用 PIM 版本 2。所有的 PIM 版本 2 的路由器需要成为 BSR 的候选者，虽然路由器 3 将赢得这个选举。路由器 2 应当是前半组播组范围的 RP，而路由器 8 应当成为后半组播组范围的 RP。

3.17.1 实验 9：解决方案

越复杂的实验需要越多的时间来进行设置。这个实验需要稍微想一想并且在过程中多加思索。

VLAN A 和 Backbone 1 由路由器 5 和路由器 13 代表。然而，为了彼此会话，路由器 6 的 Serial 1.1 也必须参与这个版本的组播路由。所有的路由器需要启用 IP 组播路由。在组播网络中的这半部分，和帧中继云协同工作，路由器 5 需要成为 RP，并且宣告它自己。

这就提醒用户一些不同的需求。首先，PIM 的稀疏密集模式是必需的。其次，因为路由器 5 和路由器 13 之间的帧中继在路由器 6 上是一个多点接口，所以也需要在路由器 13 上建立一个映射代理来转发 RP 的宣告。

作为一个帧中继云团，应当在串行接口上有 **ip pim nbma-mode** 命令。

虽然不需要，也应当在路由器 5 和路由器 13 上指定 IP PIM 版本 1。路由器 6 不应配置这条命令，因为这个场景的第二部分特别需要版本 2。

查看这个网络的第二部分和场景，可以看到 VLAN 60 和 VLAN B 需要组播。那里有很多路由器。很显然，每一台路由器都需要启用组播。

这个场景声明这里所有的路由器都应当是 BSR 的候选者，因此，每一台路由器都需要 **ip pim bsr-candidate** 命令；虽然路由器 3 应当比其他路由器有更高的权重，实际上应当被选举为主路由器。

当你关注完 BSR 后，现在是查看这部分网络中的 RP 的时候了。路由器 2 和路由器 8 都应当是 RP，虽然对于不同的组播组。因此，在每一台路由器上使用 **ip pim rp-candidate** 命令并带有一个组列表来调用一个访问控制列表。

这个练习要求用户有一点关于二进制的知识，来建立包括半个组播范围的访问控制列表。记住组播的整体范围是 224.0.0.0/4。因此，224.0.0.0/5 是一部分的组播范围，而剩下的 232.0.0.0/5 是另外一部分范围。二进制——它使得生活更加富有激情。

3.17.2 实验 9：配置

这个实验演示了命令表项的另一种用法以及如何如何在路由器上使用 **show running-configuration** 来编辑命令表项。（参看范例 3-8）

范例 3-8 在路由器上使用 show running-configuration 编辑的命令表项

```
R13
ip multicast-routing
ip pim version 1
!
interface ethernet 1/0
ip pim sparse-dense-mode
```

（待续）


```
!
interface serial 1/0
 ip pim sparse-dense-mode
 ip pim nbma-mode
!

R5
ip multicast-routing
ip pim version 1
!
interface ethernet 0
 ip pim sparse-dense-mode
!
interface serial 1
 ip pim sparse-dense-mode
 ip pim nbma-mode
!
ip pim send-rp-announce ethernet 0 scope 16

R6
ip multicast-routing
ip pim bsr-candidate ethernet 0 30 10
ip pim send-rp-discovery scope 16
!
interface serial 1.1
 ip pim sparse-dense-mode
 ip pim nbma-mode
!
interface ethernet 0
 ip pim sparse-dense-mode
!
interface serial 0
 ip pim sparse-dense-mode
!

R8
ip multicast-routing
ip pim bsr-candidate ethernet 0 30 10
!
interface serial 0
 ip pim sparse-dense-mode
!
interface ethernet 0
 ip pim sparse-dense-mode
!
access-list 8 permit 232.0.0.0 7.255.255.255
ip pim rp-candidate ethernet 0 group-list 2

R3
ip multicast-routing
ip pim bsr-candidate ethernet 0 30 20
!
interface ethernet 0
 ip pim sparse-dense-mode
!
```

(待续)

```
R2
ip multicast-routing
ip pim bsr-candidate ethernet 0 30 10
!
interface ethernet 0
 ip pim sparse-dense-mode
!
access-list 2 permit 224.0.0.0 7.255.255.255
ip pim rp-candidate ethernet 0 group-list 2
```

3.18 监控和测试

当在整个网络拓扑中的路由器上配置组播路由后，测试完整的功能是一个很好的主意。在 CCIE 的实验中，任何场景的目的都是考虑路由器。

也就是说，很多的命令允许用户去“看”路由器看到的东西，并且试图像一台路由器那样去思考问题。

3.18.1 show 和 debug 命令

一系列的 **show** 和 **debug** 命令允许用户对组播网络进行故障排查和监控。对组播网络进行故障排查的本质类似于对单播网络进行故障排查，因为组播依赖于单播路由表进行决策。

当对组播进行故障排查时，需要考虑下面两个主要的区域：

- 数据包自身的流动（例如，将单播路由表和所使用的配置命令作比较）；
- 组播的信令、RP 的选举和使用以及相关的配置。

下面有一些命令可以使用：

```
show ip pim neighbor
show ip pim interface
show ip pim rp
show ip mroute
show ip mroute summary
show ip igmp groups
show ip igmp interface
show ip rpf (ip#)
debug ip pim (multicast#)
debug ip igmp
debug ip mroute (multicast#)
debug ip mpacket
```

3.18.2 mtrace、mrinfo 和 mstat 命令

mtrace、**mrinfo** 和 **mstat** 命令内置在思科 IOS 软件中并且提供一些有用的特性。

mtrace 命令允许用户执行 RPF 的检查，并且跟踪从组播源通过组播树到达一个特定目的的路径或者说一个组播组看见的是什。这个命令的基本命令语法如下：

```
mtrace source-addr [destination-addr] [group-addr]
```

范例 3-9 显示了这个命令的某些输出范例。

范例 3-9 mtrace 命令输出

```
Router> mtrace 172.16.0.0 172.16.0.10 239.254.254.254
Type escape sequence to abort.
Mtrace from 172.16.0.0 to 172.16.0.10 via group 239.254.254.254
From source (?) to destination (?)
Querying full reverse path...
0 172.16.0.10
-1 172.16.0.8 PIM thresh^ 0 0 ms
-2 172.16.0.6 PIM thresh^ 0 2 ms
-3 172.16.0.5 PIM thresh^ 0 894 ms
-4 172.16.0.3 PIM thresh^ 0 893 ms
-5 172.16.0.2 PIM thresh^ 0 894 ms
-6 172.16.0.1 PIM thresh^ 0 893 ms
```

mrinfo 命令允许用户决定哪些路由器和当前正在做测试的路由器交换 PIM 信息。使用标记来讨论组播路由器的某些特殊的能力。这个命令的基本命令语法如下：

```
mrinfo [ mcast-neighbor#] [ interface]
```

范例 3-10 显示了这个命令的输出范例。

范例 3-10 mrinfo 命令输出

```
Router# mrinfo
172.31.7.37 (r8.lab.emanon.com) {version cisco 12.1} {flags: PMSA}:
172.31.7.37 -> 172.31.7.34 (r4.lab.emanon.com) [1/0/pim]
172.31.7.37 -> 172.31.7.47 (r7.lab.emanon.com) [1/0/pim]
172.31.7.37 -> 172.31.7.44 (r14.lab.emanon.com) [1/0/pim]
10.11.26.10 -> 10.11.26.9 (routera.lab.emanon.com) [1/32/pim]
```

在这个输出中的标记包括下面这些：

- **P**——可剪枝；
- **M**——可做 mtrace；
- **S**——具有 SNMP 能力；
- **A**——可做自动 RP。

mstat EXEC 命令允许用户查看对于某一个源、目的或者组播组地址的 IP 组播数据包的速度和丢失信息。这个命令的基本命令语法如下：

```
mstat source-addr [destination-addr] [group-addr]
```

3.18.3 组播故障排查范例

正如图 3-13 所示的组播网络，组播数据包从组播源 150.1.1.1 进入了路由器 A 的 E0 接口，

并且发送到组播组 225.3.3.3。这将产生一个 S，G (150.1.1.1, 225.3.3.3)。

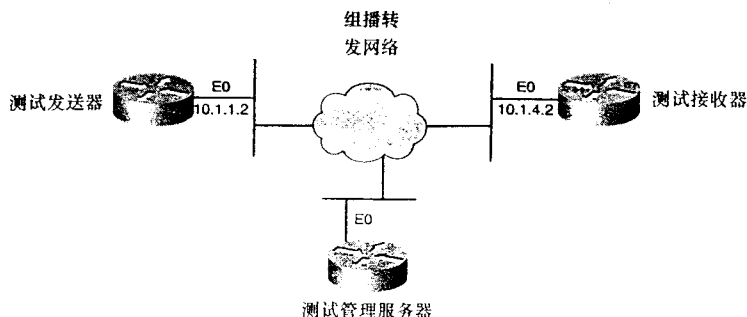


图 3-13 故障排查组播网络

连接到路由器 A 的主机正确地接收到了组播流量，但是连接到路由器 B 的主机却不能够接收到。第一步应当是在两台路由器上查看组播路由表。范例 3-11 显示了路由器 A 的配置。

范例 3-11 路由器 A 的配置

```
RouterA# show ip mroute 225.3.3.3
IP Multicast Routing table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 225.3.3.3), 00:01:23/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:23/00:00:00
(150.1.1.1, 225.3.3.3), 00:01:23/00:03:00, flags: TA
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1, Forward/Sparse-Dense, 00:01:23/00:00:00
```

因为路由器运行在 PIM 的密集模式下，*，G 的路由是不重要的。标志 D 代表是密集模式。S，G 路由代表接收或者发送的接口应当被检查。路由器 A 表现为正在正常工作。范例 3-12 显示了路由器 B 的配置。

范例 3-12 用于验证的组播 show 命令

```
RouterB# show ip mroute 225.3.3.3
IP Multicast Routing table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 225.3.3.3), 00:05:36/00:02:19, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:05:36/00:00:00
    Ethernet1, Forward/Sparse-Dense, 00:05:37/00:00:00
```

在范例 3-12 中的组播路由表没有显示 S, G 组, 这意味着路由器 B 没有转发组播数据包。参考范例 3-13 了解在网络验证时所使用的 **show ip pim neighbor** 命令。

范例 3-13 用于验证的组播 show 命令

```
RouterB# show ip pim neighbor
PIM Neighbor Table
Neighbor Address Interface Uptime Expires Ver Mode
200.10.1.1 Ethernet0 2d00h 00:01:15 v2
```

路由器 A 被证明是一个 PIM 的邻居, 正如我们所期望的。范例 3-14 显示了用于提供验证的 **show ip rpf 150.1.1.1** 命令。

范例 3-14 用于验证的组播 show 命令

```
RouterB# show ip rpf 150.1.1.1
RPF information for ? (150.1.1.1)
RPF interface: Ethernet2
RPF neighbor: ? (4.1.1.2)
RPF route/mask: 150.1.1.1/32
RPF type: unicast (static)
RPF recursion count: 1
Doing distance-preferred lookups across tables
```

这条命令显示出到达 150.1.1.1 的 IP 路由出现了, 是我们所期望的路由器 B 的 Ethernet2 接口。基于这个图, E0 应当是所期望的, 但是你永远都不知道在某个场景下是什么影响路由表的。范例 3-15 显示了用于验证的组播 **debug** 的输出。

范例 3-15 用于验证的组播 debug 输出

```
RouterB# debug ip mpacket
*Jan 14 09:45:32.972: IP: s=150.1.1.1 (Ethernet0)
d=225.3.3.3 len 60, not RPF interface
*Jan 14 09:45:33.020: IP: s=150.1.1.1 (Ethernet0)
d=225.3.3.3 len 60, not RPF interface
*Jan 14 09:45:33.072: IP: s=150.1.1.1 (Ethernet0)
d=225.3.3.3 len 60, not RPF interface
*Jan 14 09:45:33.120: IP: s=150.1.1.1 (Ethernet0)
d=225.3.3.3 len 60, not RPF interface
```

基于 **debug**, 你可以看到什么决定了 RPF 的检查。组播数据包所到达的接口不是与 RPF 检查相关的接口, 因此, 这个接口会将组播数据包丢掉。

假设单播路由表是基于其他的问题或理由建立的, 解决这个问题的最简单的方法就是对组播组 225.3.3.3 的源建立一条静态的组播路由, 将期望的接口设置为 Ethernet0。下面的 **ip mroute** 命令也许可以帮助解决这个问题:

```
Router(config)# ip mroute 150.1.1.1 255.255.255.255 ethernet0
```

3.18.4 组播路由管理器 (MRM)

执行一个“活动”的组播测试：发送者测试（组播源）、接收者测试（组播接收者）和测试管理器。

图 3-14 所示的网络显示了 MRM 测试是如何设计的。

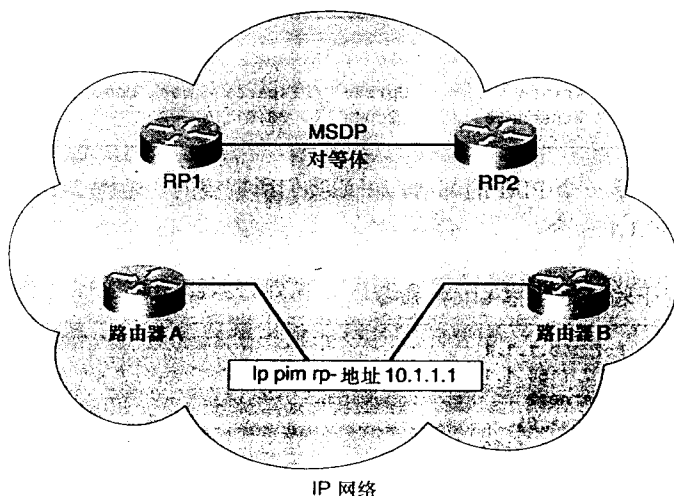


图 3-14 组播测试

在组播转发网络中组播路由器的数量是无关的。将发送者测试和接收者测试的位置适当地放置可以测试并且对网络的不同部分进行故障排查。

发送者测试在它的 Ethernet0 接口上将使用 **ip mrm test-sender interface** 命令。同样，接收者测试在它的 Ethernet0 接口上将使用 **ip mrm test-receiver interface** 命令。

测试管理器将需要更多的配置步骤。首先，必须配置访问控制列表来界定网络中的发送者和接收者。一个标准的访问控制列表可以界定特定的主机。访问控制列表 1 是发送者列表，而访问控制列表 2 是接收者列表。

```
Manager(config)# access-list 1 permit 10.1.1.2
Manager(config)# access-list 2 permit 10.1.4.2
```

下一步，配置一个 MRM 测试并且注明在那个测试中的发送者和接收者。注意发送者和接收者需要调用访问控制列表来指定发送者和接收者。可以在管理器中设置不止一个当前的测试。

```
Manager(config)# ip mrm manager mynettest
Manager(config-mrm)# manager ethernet0 group 239.2.3.4
Manager(config-mrm)# senders 1
Manager(config-mrm)# receivers 2 sender-list 1
```

当完成配置后，可以从 EXEC 模式启用测试，使用 **mrm test-name start** 命令。

MRM 是对组播网络的一个完整的测试。发送者和接收者必须加入一个特定的组播组（224.0.1.111）来和管理器会话。控制信息通过这个组播组传送。而且，有一系列的 UDP 信息和 RTP 信息的测试过程（除了所期望的组）。

当测试开始后，MRM 发送单播的控制信息给发送者和接收者。接着，管理器开始发送测试的信号。发送者和接收者对这个信号发送确认并且发起对所配置的组播组的测试。报告信息发送给管理器来决定测试结果是成功还是失败。

当测试还在进行中，发送者每隔 200ms（默认）就发送 RTP 数据包给所配置的组播组地址。接收者期望在同一个窗口中收到数据包，因此给管理器一个统计报告。如果接收者监测到数据包在 5s 的窗口内丢失，就会发送一个报告给管理器。

```
Manager# mrm mynettest start
*Mar 20 10:29:51.798: IP MRM test mynettest starts .....
Manager#
```

在屏幕上不会出现自动更新。为了在管理器的路由器上显示状态报告，输入下面的命令：

```
Manager# show ip mrm status
IP MRM status report cache:
Timestamp      Manager      Test Receiver  Pkt Loss/Dup (%)  Ehsr
*Mar 20 14:12:46 10.1.2.2      10.1.4.2       1                  (4%)              29
*Mar 20 18:29:54 10.1.2.2      10.1.4.2       1                  (4%)              15
Manager#
```

这个报告显示接收者（10.1.4.2）发送了两个单独的状态报告（每行一个）。每一个报告含有在窗口间隔内一个数据包的丢失（默认 1s）。Ehsr 值显示了对于 MRM 发送者的下一个估测的序列号。如果 MRM 接收者看到相同的数据包，在 Pkt Loss/Dup 这一列显示的是一个负数。

为了停止测试，输入下面的命令：

```
Manager# mrm mynettest stop
*Mar 20 10:31:32.018: IP MRM test mynettest stops
Manager#
```

3.19 CCIE 组播实验场景

当在 CCIE 实验中学习组播时，考虑措辞的重要性。本章提供了一些范例，从中你可以了解找到一些关键词的重要性。你必须确定 CCIE 实验要求你完成什么样的工作。

当比较组播路由表和单播路由表时，你可能遇到先前我们在“组播故障排查范例”一节中遇到的问题。你可能会遇到其他一些不可预知的困难。

3.20 进一步阅读资料

RFC 2362, *Protocol Independent Multicast-Sparse Mode*

RFC 1075, *Distance Vector Multicast Routing Protocol*

Developing IP Multicast Networks: The Definitive Guide to Designing and Deploying Cisco IP Multicast Networks, Volume I, by Beau Williamson (Cisco Press, 2000)

Cisco Connection Online—Documentation CD—Configuring IP Multicast Guides

第四部分

性能管理和服务质量

第4章 路由器的性能管理

第5章 集成和差分服务

第6章 服务质量——速率限制 和对流量进行队列处理

第 4 章

路由器的性能管理

从某种意义来说，每一个网络的生命周期中，都必须实施某种程度的质量机制来提供某种级别的服务。对某些网络来说，可能只需要隔若干年对硬件或者软件做一些简单的升级。对另外一些网络来说，可能需要使用*服务级别约定 (SLA)*，要么是用户，要么是服务提供商，必须确保提供一定程度的服务级别。有一系列的方法可以提供一定程度的服务质量 (QoS)；所选择的方法是由方案的可用性、费用和它给公司带来的价值决定的。当决定所需的服务级别时，必须决定是需要一个“尽力传递”的服务级别，还是需要某种程度的确保的服务质量。例如，你的网络在峰值工作时刻，可能只需要一个确定的带宽量，在传输介质上具有一定的传输速率，或者你在网络上有应用程序，它可能有特定的需求。在每种情况下，都可以建立一个服务质量方法来确保网络运行在一个建立的极限内。也必须考虑获取一定的服务级别，还可能需要冗余的链路和硬件，花费更多的费用来实施和维护。

在许多情况下，网络质量问题由某些问题导致，而这不能用服务质量的问题解决。在设计或者实施思科 IOS 服务质量技术之前，验证网络已经运行在最佳的状态下。例如，我确定每一个人都已经看到至少一次这样的情况，就是某台路由器正在不断地产生网络延迟。这个网络中的用户通常会抱怨他们的网络运行得太慢了，但是没有人知道这是为什么，直到有一天某个人查看了路由器的以太接口并且注意到在这个接口上有大量的错误。在这个接口上看到的错误表明有一个坏的以太线缆，当这个线缆被替换后，所有的一切都工作正常了。下面列举的简单质量控制问题不是思科 IOS 服务质量能够解决的：

- 路由器的资源限制——路由器丢弃数据包是因为它们的资源被消耗光了。
- 路由器硬件的问题——坏的接口导致的性能问题。
- 第一层网络的问题——坏的线缆或者不符合标准的线缆。

路由器的资源限制通常发生在路由器不再能够支持流量的性质或者是当前网络中正在使用的一些特性的情况下。这些问题通常可以通过增加费用或者替换过时的设备来解决。路由器的硬件问题趋向于越来越难发现，但是易于解决，当你仔细查看你的网络时，可能这种问题就消失了。第一层的线缆问题可以导致一些奇怪和复杂的问题，通常较难追踪。

本章的开始涵盖了质量控制的问题，并且包括了几个快速的故障排查练习，可以帮助用户加快诊断和解决问题的速度。整本书都专注于质量控制的问题，本章只是辨别在思科 IOS 软件中已经存在的一些工具，并且显示了这些工具的输出如何提供有用的故障排查数据。这部分中一些命令的输出附在本书的最后作为参考。

在这部分内容讨论之后，本章会在这儿告一段落，接着探讨 ATM 服务质量技术。ATM 一节以 ATM 的回顾开始，直接过渡到 ATM 的服务质量。本章会接着探讨不同版本思科 IOS 的交换方法，即如何应用它们来提高网络的接口性能。本章以对接口压缩的深入探讨作为结束，告诉用户如何使用这个技术来提高网络性能，通过现有的接口发送更多的数据包而无需增加费用去升级网络带宽。

4.1 决定路由器的性能

在试图决定网络所需的服务质量类型之前，首先完成下面的一些任务：

- 验证网络硬件被正确地配置并且工作正常。
- 执行网络基线的检查，确定硬件是否足够支持你的需求，以及是否有足够的带宽来支持你的网络应用程序。基线也可以显示网络中的任何应用程序是否有一些链路速率或传输质量的要求。
- 考虑关键的网络参与者。决定谁需要参与网络的规划活动并且确保你知道当前和将来的网络需求。

4.1.1 验证思科 IOS 软件和内存的配置

有一些关键的命令可以帮助用户确定一个运行思科 IOS 软件的路由器是否正常工作。在一段时间内收集和记录信息，并且考虑在峰值和低谷期时网络的运行状态。花费多长时间收集和记录信息完全取决于你的网络覆盖和规模。从头开始，我们来看一下路由器当前正在运行的思科 IOS 软件的版本，并且检查路由器上 Flash 内存的大小和 DRAM 的大小。验证软件版本和内存大小是否足够支持你现在和将来所需要的一些特性。在思科的路由器上，为了找到思科 IOS 软件的版本和所安装的内存大小，使用 **show version** 命令，如范例 4-1 所示。

范例 4-1 show version 命令

```
Router# show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-JS-M), Version 12.0(3)T3, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 15-Apr-99 17:05 by kpma
Image text-base: 0x80008088, data-base: 0x80C2D514
ROM: System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
2610 uptime is 2 hours, 21 minutes
System restarted by reload
System image file is "flash:c2600-js-mz.120-3.T3.bin"
cisco 2610 (MPC860) processor (revision 0x203) with 24576K/8192K bytes of memory.
Processor board ID JAD04180ETY (2670216847)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
 1 Ethernet/IEEE 802.3 interface(s)
 2 Serial network interface(s)
16 terminal line(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read/Write)
Configuration register is 0x2102
Router#
```

在这个范例中，路由器正在运行思科 IOS 版本 12.0 (3)，映像的名字是 c2600-js-mz.120-3.T3.bin，8 MB 的闪存；这台路由器还有 32 MB 的 DRAM、25 MB 的系统内存和 8 MB 的共享数据包内存。思科 IOS 软件的版本和闪存的大小及随机访问内存的大小应当被跟踪并且记录下来以备日后参考。可以使用这个信息来跟踪软件的故障，跟踪特性，并且准备升级。在此时，了解路由器如何启用的信息是非常有帮助的，在这个范例中，路由器是由于 **reload** 而重启的。当路由器有错误时作记录总是一个非常好的主意，这样做能够记住错误，并且可观察将来是否再次发生。

System restarted by error - a SegV exception, PC 0x808da564

记录未知事件的系统重启可以节省故障排查的时间并且提供有用的信息，利用它可以发现路由器重启的原因。可以通过 www.cisco.com 搜索这个错误来找到相关的信息，或者在思科技术支持中心 (TAC) 开一个案例 (Case)。当诊断路由器重启的原因时你会发现下面的这些工具非常有用：

- 缺陷跟踪器；
- 搜索 TAC 站点；
- 错误信息解码器。

如果你发现你的路由器经常遇到真正的硬件和软件问题，首先集中精力解决这个问题，当解决完故障后，你可以开始考虑网络应用程序的需求并且找到增强应用程序性能的解决方案。

4.1.2 决定网络应用程序的需求

使用的需求是什么，有多少计算机将要使用这些新的应用程序，它们位于哪里，是否有任何带宽或者链路质量的需求。如果你不能在网络中增加传输的带宽，还可以通过使用思科 IOS 软件的服务质量特性来增强网络性能，包括下面这些特性：

- 简单队列和流量的优化；
- 高级交换方法；
- 压缩；
- 拥塞避免；
- 高级队列和拥塞管理；
- 流量整形；
- 流量监管；
- 采用 ATM 服务质量；
- 低延迟队列；
- 将流量进行分类，以在不同的网络点上提供服务质量。

学习和理解新的应用程序和技术的需求是驱动网络提高质量的动力。例如，你可能发现具有很低的广域网链路带宽的分枝路由器需要压缩来支持已经或者即将实施的网络应用程序。当决定路由器将对应用程序实施压缩技术后，你可能发现压缩技术非常依赖于路由器的处理器或者内存。当你决定推动这个计划来实施压缩技术后，你可能想增加内存的大小，或者在某些情况下，替换旧的设备来支持相关的技术。

为了检查处理器的使用和 CPU 各个进程利用率的分配，使用 `show processes cpu` 命令，如范例 4-2 所示。

范例 4-2 `show processes cpu` 命令

Router# show processes cpu								
CPU utilization for five seconds: 1%/0%; one minute: 0%; five minutes: 0%								
PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	1650	2	0.00%	0.00%	0.00%	0	Load Meter
2	1573	2653	592	1.31%	0.49%	0.34%	0	Exec
3	5701	990	5758	0.00%	0.04%	0.05%	0	Check heaps
4	0	1	0	0.00%	0.00%	0.00%	0	Pool Manager
5	0	2	0	0.00%	0.00%	0.00%	0	Timers
6	4	61	65	0.00%	0.00%	0.00%	0	Serial Backgroun
7	0	276	0	0.00%	0.00%	0.00%	0	Environmental mo
8	0	143	0	0.00%	0.00%	0.00%	0	ARP Input
9	5	6	833	0.00%	0.00%	0.00%	0	DDR Timers
10	0	2	0	0.00%	0.00%	0.00%	0	Dialer event
11	8	2	4000	0.00%	0.00%	0.00%	0	Entity MIB API
12	0	1	0	0.00%	0.00%	0.00%	0	SERIAL A'detect
13	0	1	0	0.00%	0.00%	0.00%	0	Critical Bkgnd
14	52	992	52	0.00%	0.00%	0.00%	0	Net Background
15	4	59	67	0.00%	0.00%	0.00%	0	Logger
16	48	8228	5	0.00%	0.00%	0.00%	0	TTY Background
17	8	8380	0	0.00%	0.00%	0.00%	0	Per-Second Jobs
18	16	8312	1	0.00%	0.00%	0.00%	0	Partition Check
19	88	725	121	0.00%	0.00%	0.00%	0	Net Input
20	12	1651	7	0.00%	0.00%	0.00%	0	Compute load avg
21	3915	141	27765	0.00%	0.05%	0.00%	0	Per-minute Jobs

`show processes cpu` 命令的第一行通常是非常重要的：CPU utilization for five seconds:

1%/0%; one minute: 0%; five minutes: 0%。这一行显示了 CPU 在 5s、1min 和 5min 间

隔的利用率。这个数据可以在路由器本地通过重复地发起这个命令来即时查看，或者可以使用数据收集软件在一段时间内收集数据并且使用它来发现网络的趋势，从而决定将来网络的需求。在先前的范例中路由器运行 0% 的利用率。如果你注意到路由器经常运行在超过 75% 的利用率上，你可能需要考虑路由器升级，或者在先前的压缩方式下，你可以考虑升级广域网的链路并且关闭压缩。

为了从路由器收集性能趋势信息，**show processes cpu** 命令的输出在一段时间内是非常有价值的信息，包括峰值和低流量时间。如果处理器的利用率很高，从 PID 这列记录最消耗时间的进程 ID。可以关掉某些进程来节省资源。

当收集处理器利用率时，也可以收集内存的利用率。虽然某些时候很难读懂或者理解，**show memory** 命令显示一些关于系统利用率的信息。有许多 **show memory** 命令的变种，其中最有用的一个就是 **show memory dead** 命令。

正如范例 4-3 所示，**show memory dead** 命令显示关于内存使用的汇总信息、总体量、已使用的和空闲的内存数据，并接着显示所有的死亡进程，但仍然占据着分配给它们的内存。如果这个数字很大，可能需要找到已死亡的进程并且和思科 TAC 协同工作以解决问题。

范例 4-3 show memory dead 命令

Router# show memory dead								
	Head	Total(b)	Used(b)	Free(b)	Lowest(b)	Largest(b)		
Processor	811E15FC	6416900	3884876	2532024	2495784	2508960		
I/O	1800000	8388608	1566808	6821800	6819308	6821756		
Processor memory								
Address	Bytes	Prev.	Next	Ref	PrevF	NextF	Alloc PC	What
8120E740	64	8120E6E8	8120E7AC	1			808AF3AC	CEF process
812A3F44	92	812A3EB0	812A3FCC	1			801D4870	TTY timer block
812A8C00	24	812A8BB8	812A8C44	1			808AF3AC	CEF process
812A8DDC	24	812A8D98	812A8E20	1			808AF3A0	CEF process

除了显示内存的汇总和对死亡进程的内存分配之外，**show memory** 命令对检查内存分配故障方面的问题也非常有帮助，可以使用 **show memory failures alloc** 命令。这个命令显示了任何内存分配的失败问题，当通过一段时间收集后，可以指明增加内存的需要。在正常的情况下，这个命令不应当有任何输出。

作为一个规则，路由器永远不应当运行在一个持续的高 CPU 利用率或高内存利用率的情况下。有一系列的原因导致一个人应当判断路由器的处理器和内存的利用率。通常，作为一个提请注意的方法，在添加任何服务质量功能之前，确保路由器能够处理由新的服务质量特性所增加的负荷。如果路由器的内存利用率已经非常高，增加新的特性，即便是诸如思科快速转发（CEF）交换这类交换模式的变化，都可能导致路由器处理能力走向极限。当验证路由器的基本能力能够执行你需求的功能之后，使用刚提到的 **processor** 和 **memory** 命令，或者你意识到需要做路由器的升级或者替换。接着验证路由器有足够的接口能力来处理流量的负荷。下一小节讨论路由器接口的性能评估，显示了如何识别接口的硬件和电缆故障、流量的瓶颈以及路由交换模式选择的有效性。

4.1.3 验证路由器的接口性能

配置、利用率、错误和队列的信息。范例 4-4 显示了 **show interface serial** 命令的输出，表 4-1 显示了 **show interface serial** 命令的输出描述。

范例 4-4 show interface 的输出

```
Router# show interface serial s 0/1
Serial0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 175.25.33.98/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:02, output 00:00:03, output hang never
  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 0/2/256 (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
    179 packets input, 12647 bytes, 0 no buffer
    Received 70 broadcasts, 0 runts, 0 giants, 0 throttles
    1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored, 0 abort
    173 packets output, 17321 bytes, 0 underruns
    0 output errors, 0 collisions, 78 interface resets
    0 output buffer failures, 0 output buffers swapped out
    106 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

表 4-1 show interface serial 的输出描述

表项	描述
Hardware is PowerQUICC Serial	在这个范例里，描述了接口的硬件名字，硬件是一个 PowerQUICC WIC-1T 串行模块 更详细的硬件类型描述和接口的特定故障排查计数器可以使用 show controllers 命令找到
Internet address is 175.25.33.98/24	分配给这个接口的 IP 地址。这个信息只出现在 IP 接口上
MTU 1500 bytes	这个接口的 MTU 尺寸 可以在接口配置模式下使用 mtu 命令改变一个接口的 MTU 尺寸。 no mtu 命令可以将 MTU 的尺寸置为默认值
BW 1544Kbit	显示这个接口的带宽。带宽值并不实际改变这个接口的可用带宽。这个命令只是对 EIGRP 或者 IGRP 的路由协议提供一个度量值来限制 Hello 流量 默认的带宽值将是接口的值，或者是可以通过在接口配置模式下使用 bandwidth 命令输入的一个更准确的值
DLY 20000 usec	以毫秒表示的平均接口延迟。而且，注意这里显示的延迟值只是用于 EIGRP 或者 IGRP 路由协议的度量值 可以在接口配置模式下使用 delay 命令来修改一个接口的延迟值
reliability 255/255	5min 间隔内的链路平均可靠性 255/255 指的是 100% 的可靠性 127/255 指的是 50% 的可靠性 1/255 将是 0% 的可靠性
txload 1/255	接口在 5min 内的传输负荷。255/255 指的是 100% 的接口利用率
rxload 1/255	接口在 5min 内的接收负荷。255/255 指的是 100% 的接口利用率
Encapsulation HDLC	接口的封装类型
loopback not set	显示是否配置了一个环回。接口的环回可以用于测试物理连接的问题，通过传输一个信号给远端的站点，有时称为“接口打环”给服务提供商。为了配置一个打环的接口，在接口配置模式下使用 loopback 命令

续表

表项	描述
Keepalive set (10 sec)	先是接口的 keepalive 。对一个串行接口的标准的 keepalive 值是 10s 为了改变接口的 keepalive ，在接口的配置模式下使用 keepalive 命令
Last input 00:00:02	显示最后一次在这个接口上的输入时间
output 00:00:03	显示最后一次在这个接口上的发送时间
output hang never	显示因为传输花费时间太长而导致接口被复位的最后一次时间
Last clearing of show interface counters never	显示这个接口的计数器最后一次被清零的时间 可以在启用模式下使用 clear interface 命令来清空接口的计数器
Input queue:0/75/0 (size/max/drop)	显示接口的输入队列的尺寸 size 显示当前的输入队列的尺寸 max 显示队列的最大尺寸 drops 显示当最大队列的尺寸达到时，数据包被丢弃的数量
Total output drops: 0	显示输出丢弃的总体数量。输出丢弃发生在当路由器试图传输数据，而没有可用的缓冲区，故而将包丢弃的情况下
Queuing strategy: weighted fair	显示接口的队列策略 默认的情况下低于 2Mb (E1) 的串行接口的默认队列类型是加权公平队列 如果没有配置队列的类型，或者加权公平队列已经被关掉了，那么默认的队列类型是先进先出
Output queue: 0/1000/6410 (size/max total threshold/drops)	显示接口的输出队列尺寸 size 显示当前队列尺寸 max total 显示队列的最大尺寸 threshold 显示了在新的数据包被丢弃之前，可以存储在队列中的数据包个数 drops 显示了丢弃的数据包数量
Conversations 0/2/256 (active/max active/max total)	显示了对接口的加权公平队列的设置。加权公平队列会在下一章中详细讨论 active 显示了加权公平队列的当前的会话数量 max active 显示了可以最大化的加权公平队列的会话数量 max total 显示了动态加权公平队列的会话数量
Reserved Conversations 0/0 (allocated/max allocated)	当启用 RSVP 后，当前的 RSVP 资源分配的数量和最大的 RSVP 资源分配的数量
5 minute input rate 0bits/sec, 0 packets/sec	显示对于接口的 5min 的平均输入速率
5minute output rate 0bits/sec, 0 packets/sec	显示对于接口的 5min 的平均输出速率
235 packets input 15967 bytes 0 no buffer	这些计数器显示了下面的信息： 接收的数据包数量 在接口上收到的字节数 路由器缓存空间匮乏次数
Received 126 broadcasts 0 runts 0 giants 0 throttles	这些计数器显示了下面的信息： 收到的广播数量 收到的侏儒帧的数量。侏儒帧是指数据包的尺寸小于接口的最小数据包的尺寸 收到的巨人帧的数量。巨人帧是指数据包的尺寸超过接口的 MTU 的尺寸 收到的切换抑制的次数。切换抑制发生在当路由器的缓存空间不够用或者当处理器的资源匮乏时，接口的接收器不可用
2 input errors 0 CRC 2 frame 0 overrun 0 ignored 0 abort	这些计数器显示下面的情况： 所有的输入错误的累计情况。输入错误是指当任何数据包到达一个接口时，具有错误类型。不止一次的错误类型只被计算一次 收到的 CRC 的数量。这个数量应当低于接口收到的总体字节的 0.0001 的百分比，按照公式 (CRC 错误/总体字节) * 100 = CRC 错误的百分比。高错误率代表第一层的问题 对于接收的数据包，缓存区匮乏的次数。当接口接收数据包的速度大于系统缓存处理数据的速度时，就会发生这种情况 被忽略的数据包的数量。当接口用完缓存区的空间时，不得不忽略新的数据包，直到资源可用 abort 计数器显示了接口收到不合法的数据的次数。接口 abort 通常代表一个时钟错误

续表

表项	描述
256 packets output 22838 bytes 0 underruns	这些计数器显示下面的这些信息： 传输的数据包的数量 传输的字节的数量 路由器检测到数据的发送者发送的速度大于路由器可以接收的速度的次数
0 output errors 0 collisions 80 interface resets	这些计数器显示下面的这些信息： 输出的错误的数量 因为冲突，数据包重传的数量——串行接口不应当有冲突 接口重新复位的次数
0 output buffer failures 0 output buffers swapped out	这些计数器显示下面的这些信息： 在输出时，路由器收到的没有资源错误的次数 路由器将数据包交换到 DRAM 中的次数
106 carrier transitions	接口上感应到的载波变化的次数。当载波监测到信号状态变化时，就发生了载波状态的变化
DCD=up	DCD（数据载波检测）——DCE 发送的信号代表已经从 DTE 收到了载波检测的信号
DSR=up	DSR（数据设置就绪）——DCE 发送的信号代表去通知 DTE，即 DCE 已经就绪了
DTR=up	DTR（数据终端就绪）——DTE 发送给 DCE 的信号，用于新的连接或者维持一个现有的连接
RTS=up	RTS（请求发送）——DTE 发送的信号通知 DCE，即 DTE 已经准备就绪了
CTS=up	CTS（清除发送）——DCE 发送的信号代表 DCE 已经准备好从 DTE 接收数据了

范例 4-5 显示了 **show interface fastethernet** 命令的输出，而表 4-2 显示了命令输出的描述。

范例 4-5 show interface fastethernet 命令

```
1750a>show interface fastethernet 0
FastEthernet0 is administratively down, line protocol is down
Hardware is PQ1ICC_FEC, address is 0004.2722.81d8 (bia 0004.2722.81d8)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Auto-duplex, 10Mb/s, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output 01:03:50, 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
    Received 0 broadcasts, 0 runs, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog
    0 input packets with dribble condition detected
177 packets output, 35436 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
```


表 4-2 和以太有关的 show interface 输出

表项	描述
FastEthernet0 is administratively down, Line protocol is down.	显示当前接口和线路协议的状态：快速以太接口的可能状态是管理性的 up 或者 down up down administratively down 为了使得一个接口持续地处于 up 状态，它必须在配置的时间间隔内收到 keepalive 包
Hardware is PQICC_FEC	显示安装的硬件的类型
Address is 0004.2722.81d8 (bia 0004.2722.81d8)	显示了当前的 MAC 地址和烧入的地址（BIA） 可以在接口配置模式下使用 mac-address 命令改变 MAC 地址
MTU 1500 bytes BW 100000 Kbit DLY 100usec	MTU 带宽 以毫秒表示的接口延迟 这些值通常最好设置在期望的部分：改变带宽或者延迟并不改变实际的数值，然而，MTU 的值有时可以改变来实现不同厂商的硬件设备之间的互操作性。这些值不会动态改变
Auto-duplex	接口的双工模式 接口的双工模式可以在接口配置模式下使用 full-duplex 或者 half-duplex 命令改变
10Mbit/s	显示接口的速率 对于快速以太接口或者更快的速率，可以在接口配置模式下使用 speed 命令改变接口的速率 速率可以强行设置到某个速率，或者如果速率已经改变了，可以通过指定 auto ，将它设置为自动
100BaseTX/FX	显示以太的介质类型
ARP type: ARPA	显示 ARP 的类型 可以在接口配置模式下使用 arp type 命令来改变 ARP 的类型默认的 ARP 类型是 ARPA
ARP timeout 04:00:00	显示 ARP 超时[更多] 你可以在接口配置模式下通过 arp timeout 命令来改变 ARP 超时参数
Queuing strategy: fifo	显示接口的队列策略的类型，在以太接口上，默认的队列类型是先进先出
0 watchdog	显示看门狗计时器超时的次数。当数据包的大小大于 2048 字节时，通常发生看门狗计时器超时的情况
0 input packets with dribble condition detected	显示帧的长度超过了尺寸，但是仍旧被转发的数量
0 interface resets	显示了接口复位自己的次数
0 collisions	显示了在接口上收到的冲突的数量 在快速以太接口上通常不发生冲突
0 babbles 0 late collision 0 deferred	这些计数器显示下面的这些信息： 传输的含糊的计时器过期的次数 最新的冲突的次数，当数据帧的前导位已经被传输后，发生冲突 因为载波故障而发生的数据包被推迟的数量
0 lost carrier 0 no carrier	显示在传输期间接口丢失载波的次数 显示在传输期间接口没有载波的次数

当在一段时间内验证接口的状态后，就能够决定出问题的路由器的故障类型。在这点上，你应当看见一个明显的趋势，告诉你可能是 3 个原因。也许路由器资源消耗过度丢弃了数据包，或者路由器有物理层质量的问题。所有的这些问题都不能使用服务质量来解决。而且，也许路由器需要额外的调整才能更好地负载流量，并且可以通过 QoS 技术来提高网络的服务质量。

- 路由器资源问题——由大量的缓存故障指明。这可以通过调整缓存来实现，但是大

部分的情况下，取决于条件，最终需要路由器内存的升级。

- 物理层的问题——可看到大量的错误，可通过好的故障排查习惯来解决。
- 路由器上的大流量负荷——可看到大量的 txload、rxload 以及大量丢弃的数据包和缓存错误。

为了进一步隔离有关接口性能的质量问题，可以进一步采取一些措施。可以更详细地查看接口控制器，或者如果接口有集成的信道服务单元/数据服务单元（CSU/DSU），可以监控任何报警条件。当对链路质量进行故障排查时，首先要看的就是 **show controllers** 命令。**show controllers** 命令显示关于接口硬件的信息和电缆类型及时钟信息。**show controllers** 命令的最后一些行也可以显示特定于硬件的错误。范例 4-6 显示了 **show controllers serial** 命令。

范例 4-6 show controllers serial 命令的输出

```
Router# show controller s 0/1
Interface Serial0/1
Hardware is PowerQUICC MPC860
DTE V.35 TX and RX clocks detected.
idb at 0x8129D3E8, driver data structure at 0x812A2958
SCC Registers:
General [GSMR]=0x2:0x00000030, Protocol-specific [PSMR]=0x8
Events [SCCE]=0x0000, Mask [SCCM]=0x001F, Status [SCCS]=0x06
Transmit on Demand [TODR]=0x0, Data Sync [DSR]=0x7E7E
Interrupt Registers:
Config [CICR]=0x00367F80, Pending [CIPR]=0x00000800
Mask [CIMR]=0x20200400, In-srv [CISR]=0x00000000
Command register [CR]=0x640
Port A [PADIR]=0x0000, [PAPAR]=0xFFFF
      [PAODR]=0x0000, [PADAT]=0xF0F7
Port B [PBDIR]=0x03A0F, [PBPAR]=0x0C00E
      [PBODR]=0x0000E, [PSDAT]=0x31DDD
Port C [PCDIR]=0x00C, [PCPAR]=0x000
      [PCSO]=0x0A0, [PCDAT]=0xF30, [PCINT]=0x00F
Receive Ring
rmd(68012330): status 9000 length 18 address 1935788
rmd(68012338): status 9000 length 11D address 1932388
rmd(68012340): status 9000 length 18 address 1938508
rmd(68012348): status 9000 length 18 address 1937E88
rmd(68012350): status 9000 length 18 address 1933D88

rmd(68012358): status 9000 length 18 address 1937808
rmd(68012360): status 9000 length 18 address 1937188
rmd(68012368): status 9000 length 18 address 1934A88
rmd(68012370): status 9000 length 11D address 1936488
rmd(68012378): status 9000 length 18 address 1935E08
rmd(68012380): status 9000 length 11D address 1934408
rmd(68012388): status 9000 length 18 address 1933088
rmd(68012390): status 9000 length 18 address 1936B08
rmd(68012398): status 9000 length 18 address 1933708
rmd(680123A0): status 9000 length 18 address 1932A08
rmd(680123A8): status 8000 length 18 address 1938B88
Transmit Ring
tmd(680123B0): status 5C00 length 18 address 193A158
tmd(680123B8): status 5C00 length 18 address 193A158
tmd(680123C0): status 5C00 length 18 address 193A158
tmd(680123C8): status 5C00 length 18 address 193A158
tmd(680123D0): status 5C00 length 18 address 193A158
tmd(680123D8): status 5C00 length 123 address 1950098
```

(待续)

```
tmd(680123E0): status 5C00 length 123 address 194DE38
tmd(680123E8): status 5C00 length 18 address 193A158
tmd(680123F0): status 5C00 length 18 address 193A158
tmd(680123F8): status 5C00 length 18 address 193A158
tmd(68012400): status 5C00 length 18 address 193A158
tmd(68012408): status 5C00 length 18 address 193A158
tmd(68012410): status 5C00 length 18 address 193A158
tmd(68012418): status 5C00 length 123 address 194F2D8
tmd(68012420): status 5C00 length 123 address 1950098
tmd(68012428): status 7C00 length 18 address 193A158
SCC GENERAL PARAMETER RAM (at 0x68013D00)
Rx BD Base [RBASE]=0x2330, Fn Code [RFCR]=0x18
Tx BD Base [TBASE]=0x2380, Fn Code [TFCR]=0x18
Max Rx Buff Len [MRBLR]=1548
Rx State [RSTATE]=0x18008440, BD Ptr [RBPTR]=0x2380
Tx State [TSTATE]=0x18000548, BD Ptr [TBPTR]=0x23B8
SCC HDLC PARAMETER RAM (at 0x68013D38)
CRC Preset [C_PRES]=0xFFFF, Mask [C_MASK]=0xF0B8
Errors: CRC [CRCEC]=0, Aborts [ABTSC]=0, Discards [DISFC]=0
Nonmatch Addr Cntr [NMARC]=0
Retry Count [RETRC]=0
Max Frame Length [MFLR]=1608
Rx Int Threshold [RFTHR]=0, Frame Cnt [RFCNT]=65046
User-defined Address 0000/0000/0000/0000
User-defined Address Mask 0x0000

buffer size 1524
PowerQUICC SCC specific errors:
0 input aborts on receiving flag sequence
0 throttles, 0 enables
0 overruns
0 transmitter underruns
0 transmitter CTS losts
```

对带有集成的 CSU/DSU 控制器的广域网接口卡 (WIC) 模块的链路质量进行故障排查的另外一个命令就是 **show service-module serial** 命令。正如范例 4-7 所示，这个命令可以显示关于内部的 CSU/DSU 的信息，例如报警状态和自测试信息。应当在一段时间内跟踪 CSU/DSU 的报警。范例 4-7 显示了 **show service-module serial** 命令的输出，表 4-3 描述了输出。

范例 4-7 show service module serial 命令的输出

```
Router# show service-module serial 0/0
Module type is 4-wire Switched 56
  Hardware revision is B, Software revision is 1.00,
  Image checksum is 0x42364436, Protocol revision is 1.0
Receiver has no alarms.
CSU/DSU Alarm mask is 0
Current line rate is 56 Kbits/sec
Last module self-test (done at startup): Passed
Last clearing of alarm counters 02:13:56
  oos/oof           : 0,
  loss of signal    : 0,
  loss of frame     : 0,
  rate adaptation attemp: 0,
```

表 4-3 show service-module serial 命令的输出

表项	描述
模块类型是 4 线的 Switched 56 硬件的版本是 B，软件修订是 1.00 映像的修订号是 0x42364436，协议修订号是 1.0	CSU/DSU 模块类型
接收者没有报警，CSU/DSU 报警掩码是 0	这个区域可以显示任何由 CSU/DSU 监测到的报警
当前线路速率是 56 kbit/s	显示当前线路的速率
最后一个模块的自测试（在启用时进行）：通过	显示最后一个模块的自测试的结果
最后一次清除报警计数器的时间是 02: 13: 56	显示最后一次清除报警计数器的时间
oos/oof: 0,	out-of-synchronization (OOS) 报警显示了一个时钟同步的问题 out-of-frame (OOF) 报警显示了 1/4 的帧位丢失的问题
信号丢失: 0,	loss-of-signal (LOS) 报警显示了没有监测到物理信号
帧的丢失: 0,	loss-of-frame (LOF) 报警显示丢失了帧位
速率适配企图: 0,	显示接收者试图进行速率调整

当你修复了路由器的接口问题后，或者是验证了路由器并没有任何硬件或者软件故障而导致链路质量的问题后，在进行服务质量配置之前，你可以进一步专注于两个更有意义的方面。首先，可以验证路由器使用的是最有效的交换模式；其次，如果这个接口仍然拥塞的话，可能需要考虑采用压缩或者服务质量技术。

4.2 ATM：其他的广域网技术

网络工程师通常遇到的一个障碍就是引进了新的技术。虽然异步传输模式（ATM）不是一个新的技术——第一个 ATM 的标准是在 20 世纪 90 年代初期开发的，随后 ATM 的硬件快速出现了——并且虽然许多网络工程师对其他广域网协议非常有经验，例如高级数据链路控制协议（HDLC）、PPP、帧中继和 X.25，但是对较新的 ATM 技术还是不熟悉。本节的目的不是重复这个系列第 1 卷中的 ATM 技术，而主要是提供对 ATM 路由器性能和服务质量技术的一个基本理解。本节探讨了下面的 ATM 特定的主题：

- 理解基本的 ATM 的概念；
- 比较 ATM 和帧中继的技术；
- ATM 的性能管理（显示接口数据和基本的 ATM 故障排查技术）；
- 思科路由器上的基本的 ATM 服务质量（概念、应用和故障排查）。

一、ATM 和帧中继的相同点和不同点

本节先前提到的所有的第二层广域网技术都具有相同的共同点。例如，按照表 4-4 所示，HDLC、PPP、ISDN、X.25 和帧中继都具有相似的二层的帧格式，基于相似的二层成帧的标准。Link Access Procedure Balanced (LAPB)、Link Access Procedure on the D channel (LAPD)、Link Access Procedure for Frame Relay (LAPF) 和 Synchronous Data Link Control (SDLC) ——所有的这些技术都使用相似的帧格式，含有标志、地址、控制、信息、FCS 和标志字段。然而，这些技术最初都是为低带宽的接口设计的，例如 T1、ISDN BRI、PRI 或者 DS3。这些技术所使

用的帧主要设计用来处理可变长度数据包，因为这些协议主要设计为可变长度的三层数据单元服务的。

表 4-4

二层的广域网协议

二层协议	接口封装类型	二层协议	接口封装类型
LAPB	X.25	LAPD	ISDN
LAPF, LAPD	Frame Relay	B-ISDN*	ATM
SDLC	HDLC		

B-ISDN = 宽带 ISDN

ATM 的建立是为了使用较高带宽的接口，以恒定的速率工作。从一开始，ATM 协议主要是为了能够支持语音、数据和视频的流量，因此也被称为多业务的流量。这是通过使用固定长度的 ATM 信元完成的。ATM 交换机构成了网络核心，这和帧中继构成的网络核心类似，为诸如路由器这类 CPE 设备提供基于虚路径或虚通道的虚电路。实际上，当你从服务提供商租用帧中继电路，他们实际上很可能给你提供的是 ATM 交换机上的帧中继电路，例如思科的 MGX 交换机。当你对网络记录文档时，应当像图 4-1 所示的那样标明连接到帧中继云的路由器，这是因为服务提供商很可能不会给你提供关于他们的网络的详细信息。帧中继的流量封装在 ATM 的信元里，以 ATM 流量的形式通过 ATM 的骨干网传送，按照图 4-2 所示，在 ATM 的边界交换机上再重新翻译回帧中继的流量。因为本书的主要目的是集中于介绍路由和交换的技术，ATM 交换不会详细地介绍。



图 4-1 用户对帧中继网络的洞察力

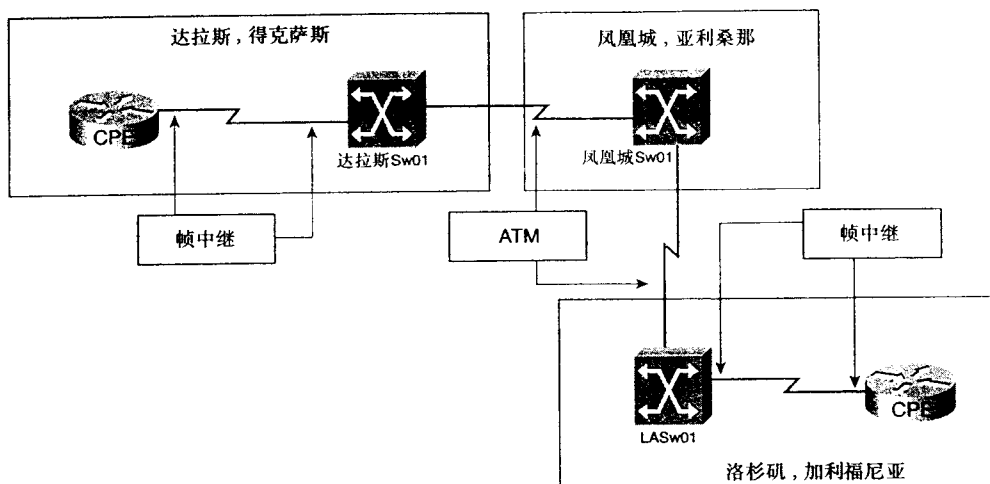


图 4-2

使得两种技术如此相似的事实就在于它们都使用虚电路来提供某种级别的服务。帧中继虚电路是通过本地有效的数据链路连接识别符 (DLCI) 来标识的。通常来说，帧中继的电路可以提供某种级别的服务；承诺信息速率 (CIR)，它可以确保访问速率。非常有可能获取低费用、尽力而为的帧中继服务。使用零 CIR 服务，交换机只在不拥塞的期间转发帧中继的流量。这个术语并不是在路由器之间的连接上必须使用的。从某种角度上来说，服务提供商网络内的拥塞可能影响流量，而你从路由器上看不到。帧中继也支持突发，或者在低利用率期间传输额外的帧。帧中继的流量可以被控制或者整形，使用保持突发速率 (Bc) 和额外突发速率 (Be) 在边界路由器上执行流量整形。

使用帧中继，低优先级的流量通过使用可丢弃位 (DE) 来注明这种流量可以被丢弃。当帧中继交换机在拥塞期间收到含有 DE=1 位设置的帧后，它会认为这种 DE 帧是低优先级的流量，可以被丢弃。不幸的是，在许多情况下，当 DE 位没有从 0 (默认值) 改变过来并且网络拥塞时，所有离开路由器的帧都会认为是可以丢弃的。在拥塞网络路径中的任何帧中继交换机都可能丢弃任何这样的帧。因为帧中继是一个无连接的协议，它依赖于上层的协议 (例如 TCP) 来重传那些丢弃的帧。

帧中继网络也有一种服务质量拥塞通知系统。这个系统使用前向显式拥塞通知 (FECN) 和后向显式拥塞通知 (BECN) 帧来通知拥塞网络路径中上游或者下游的邻居。因为 FECN 和 BECN 帧的使用必须显式地在整个网络中配置，包括在用户和服务提供商设备上配置。然而，如果没有配置拥塞通知，它就不会有任何意义。当设备没有配置响应拥塞通知帧时，它们能够提供的惟一好处就是通过帧中继的计数器来提供关于网络可靠性的历史参考。所以在拥塞期间，没有配置使用帧中继流量整形和拥塞通知的帧中继网络被证明是非常不可靠的。

ATM 被设计用来支持最初为帧中继网络设计的许多相同的技术。当帧中继最初设计时，许多服务质量特性留给厂商实施，所以这些特性的使用依赖于帧中继的硬件/软件厂商的帧中继的实施和服务提供商的帧中继网络的设计和配置。因为 ATM 是一个较新的技术，是在技术团体经历了较老的 X.25 和帧中继技术以后设计的，然而，ATM 网络本质上通过使用 ATM 的适配层 (AAL) 类型和 ATM 的服务分类来支持服务质量，如表 4-5 所示。

表 4-5

AAL 类型和它们的使用意图

AAL 类型	AAL 描述	使用意图
AAL-1	恒定的比特速率 (CBR) ——设计用来支持运行一个低信元丢失需求和最小信元延迟变量 (CDV) 的应用程序。CBR 电路被设计为仿真传统的电路，就像一个真正的 TDM 电路那样对信元速率提供并且强制一个硬限制	语音和视频的流量，不适用于突发流量，例如数据
AAL-2	这种 AAL 类型主要设计用来支持具有可变比特速率的、延迟比较敏感的面向连接的应用程序	语音和视频的流量
AAL-3/4	AAL-3/4 最初设计用来支持 Switched Multimegabit Data Service (SMDS) 的流量	过时的 SMDS 数据流量
AAL-5	AAL-5 特别设计用来支持突发的可变速率的数据流量。AAL-5 不使用对延迟比较敏感的应用程序	数据流量

不像帧中继，它最初就是为窄带技术设计的，ATM 是为宽带技术设计的，并且运行在高速率的网络上。许多 ATM 接口有内置的 ATM 逻辑，特别是为 ATM 网络设计的，不可以和其他的串行接口互换。因此，认真规划 ATM 网络是非常重要的。因为 ATM 的标准是为宽带网络设计的，所以 ATM 接口通常是在 DS3 或者更高速率的接口上可用。出于这个原因，ATM

接口的位置和使用应当提前规划。

注意：有一些类型的接口（ATM-数据交换接口[ATM-DXI]、数字用户线[DSL]和 ATM 的反向多路复用[IMA]），这些接口都支持 ATM 运行在低于 DS3 的速率下。这些类型的网络不在本书中介绍。

当配置 ATM 的子接口时，也可以有不同的 AAL-5 封装类型来选取。AAL-5 子网接入协议（SNAP）封装是默认的 ATM 接口的封装类型，因此适用于许多数据流量。表 4-6 显示了 AAL 的封装类型、它们的描述和推荐的流量类型。

表 4-6 AAL-5 封装类型

AAL-5 封装类型	描述	推荐的流量类型
aal5ciscoppp	Cisco PPP over AAL-5 封装	PPP over ATM 流量
aal5mux	AAL-5 MUX 封装可以在一个物理电路上多个不同的永久虚电路（PVC）上多路复用不同的 AAL 类型	IP 或者语音的流量
aal5nlpid	AAL-5 网络层协议识别（NLPID）封装	RFC 1483 多协议数据流量
aal5snap	AAL-5 逻辑链路控制（LLC）/SNAP 封装	默认的，RFC 1490 数据多协议流量

思科 IOS 软件的 ATM 命令在过去的一些主要版本里已经变得非常成熟。当前，主要会碰到三种不同的 ATM 配置类型。在以后的思科 IOS 软件版本里，ATM AAL 类型在思科的路由器上被称为封装类型，可以在 VC 配置模式下配置。

快速浏览一下在思科路由器上配置一个标准的 ATM PVC 的一些步骤，并且比较 ATM 和帧中继的配置。在这个范例中，我们将使用最新的思科 IOS 软件配置命令。

第 1 步 启用物理接口并且配置全局接口属性。

帧中继	ATM
启用串行接口 interface Serial0/0 no shutdown 配置帧中继封装类型 encapsulation frame-relay IETF 可选地，配置本地管理接口（LMI）类型 frame-relay lmi-type ansi 可选地，配置接口时钟或者 CSU/DSU clockrate 1300000	启用物理的 ATM 接口 interface ATM0 no shutdown

第 2 步 建立一个多点子接口。

帧中继	ATM
建立一个多点子接口，作为一个好的经验，建议你将子接口的号码和 PVC 的 DLCI 号码对应起来 interface serial0/0.651 multipoint	建立一个多点子接口，作为一个好的经验，建议你将子接口的号码和 PVC 的虚路径识别符/虚通道识别符（VPI/VCI）号码对应起来 interface ATM0.4 multipoint

第 3 步 给予接口分配 IP 地址。

```
interface Serial0/0.651 multipoint
ip address 192.168.26.1 255.255.255.252
```

或者

```
interface ATM0.4 multipoint
```

ip address 192.168.25.2 255.255.255.252

第 4 步 给子接口分配二层地址。

帧中继	ATM
给帧中继的子接口分配 DLCI 的值 interface Serial0/0.651 multipoint ip address 192.168.26.1 255.255.255.252 frame-relay map ip 192.168.26.2 651 broadcast 或者在一个物理接口上 interface Serial0/0 ip address 192.168.26.1 255.255.255.252 frame-relay interface-dlci 651	分配一个 VPI/VCI 的对和一个可选的虚电路描述符 (VCD) 的名字或者号码给予接口，使用 pvc [vcd-name] vpi/vci 命令 interface ATM0.4 multipoint ip address 192.168.25.2 255.255.255.252 pvc 4/482

注意：在帧中继网络中，可以使用 frame relay map 命令或者 frame-relay interface dlci 命令，但不能同时使用。

第 5 步 在 ATM 中，根据服务提供商提供的 ATM AAL 来选择 ATM 封装类型。

```
interface ATM0.4 multipoint
ip address 192.168.25.2 255.255.255.252
pvc 4/482
encapsulation aal5snap
```

将远端的非广播多点 (NBMA) 邻居的三层 IP 地址映射成二层的识别。可选地，启用伪-广播复制。

帧中继	ATM
如果你还没有完成，那么开始将 DLCI 映射到 IP 地址并且启用广播复制 interface Serial0/0.651 multipoint ip address 192.168.26.1 255.255.255.252 frame-relay map ip 192.168.26.2 651 broadcast	将 VCD 和 VPI/VCI 对映射到 IP 地址并且启用广播复制 interface ATM0.4 multipoint ip address 192.168.25.2 255.255.255.252 pvc 4/482 protocol ip 192.168.25.1 broadcast encapsulation aal5snap

第 6 步 可选地，配置 ATM 服务质量参数。这在本小节的后面讨论。

我们看一下在先前的配置步骤中，范例所使用的完整的 ATM 和帧中继网络。图 4-3 显示了完整的 ATM/帧中继网络，包括所有的二层和三层地址。

在这个范例中，Fred 和 Wilma 路由器属于 ATM 网络，而 Betty 和 Barney 路由器属于帧中继网络。一个令牌环的局域网连接了这两个网络。这个范例解释了帧中继网络和 ATM 网络的相同点和不同点。范例 4-8 显示了 Fred 路由器的配置。

范例 4-8 Fred ATM 路由器的配置

```
hostname Fred
!
interface Loopback100
ip address 192.168.25.9 255.255.255.248
!
interface ATM0
no ip address
no atm ilmi-keepalive
!
interface ATM0.4 multipoint
```

(待续)


```
ip address 192.168.25.2 255.255.255.252
pvc 4/482
  protocol ip 192.168.25.1 broadcast
  encapsulation aal5snap
!
router eigrp 1911
  network 192.168.25.0 0.0.0.3
  network 192.168.25.8 0.0.0.7
  no auto-summary
  no eigrp log-neighbor-changes
```

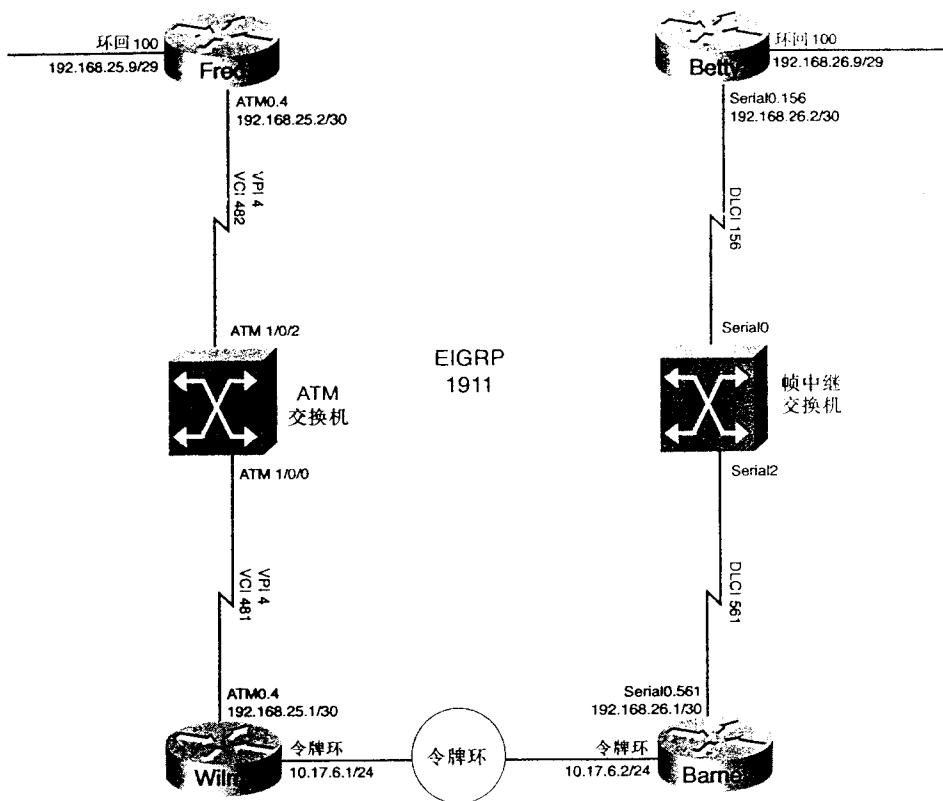


图 4-3 完整的 ATM/帧中继网络

3 个命令用于配置和启用 ATM 接口，并且在先前的范例中的 ATM 接口上配置增强的内部网关路由协议(EIGRP)。首先，**pvc 4/482** 命令用于在 ATM0.4 的多点 ATM 接口上建立 PVC。使用 **protocol ip 192.168.25.1 broadcast** 命令在子接口上将 IP 地址映射到 PVC 上。**broadcast** 选项的增加允许 EIGRP 在 NBMA 网络 ATM 网络上运行。**encapsulation aal5snap command** 在子接口上启用 AAL-5 SNAP 适配。范例 4-9 显示了对端的 Wilma 路由器上的 ATM 电路配置。

正如你所看到的，Fred 和 Wilma 路由器具有非常类似的 ATM 配置。这些配置也可以通过几种 ATM 的 show 命令进行测试。例如，**show atm interface atm0.4** 命令显示关于已经通过一个 ATM 接口所传输的数据包的数量和类型，如范例 4-10 所示。

范例 4-9 Wilma 路由器的配置

```
hostname Wilma
!
interface TokenRing0
 ip address 10.17.6.1 255.255.255.0
 ring-speed 16
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
!
interface ATM0.4 multipoint
 ip address 192.168.25.1 255.255.255.252
 pvc 4/481
  protocol ip 192.168.25.2 broadcast
  encapsulation aal5snap
!
router eigrp 1911
 network 10.17.6.0 0.0.0.255
 network 192.168.25.0 0.0.0.3
 no auto-summary
```

范例 4-10 Fred 路由器上的 show atm interface atm0.4 命令

```
Fred# show atm interface atm 0.4
Interface ATM0.4:
AAL enabled: AAL5 AAL3/4, Maximum VCs: 1023, Current VCCs: 1
Maximum Transmit Channels: 0
Max. Datagram Size: 4528, MIDs/VC: 1024
PLIM Type: SONET - 155000Kbps, TX clocking: LINE
1981 input, 1986 output, 0 IN fast, 0 OUT fastUBR+ : 4
Avail bw = 154996
Rate-Queue 0 set to 56Kbps, reg=0x0 DYNAMIC, 1 VCC
Config. is ACTIVE
```

当对一个 ATM 的接口进行故障排查时，能够看见协议的映射是非常有帮助的。**show atm map** 命令显示对于一台路由器上所有的 VC 的第二层到第三层的协议映射信息，非常类似于帧中继网络上的 **show frame-relay map** 命令。范例 4-11 显示了在 Fred 路由器上使用 **show atm map** 命令的输出。

范例 4-11 Fred 路由器上的 show atm map 命令

```
Fred# show atm map
Map list ATM0.4pvc1 : PERMANENT
ip 192.168.25.1 maps to VC 1, VPI 4, VCI 482, ATM0.4
, broadcast
```

先前的范例显示了 ATM 接口 0.4 有一个永久的对于 IP 地址 192.168.25.1 到 VPI 4 和 VCI 482 的 PVC 映射。VPI/VCI 4/482 属于 VC 1，并且这个 VC 支持伪广播，可通过在 ATM 的子接口下使用 **protocol ip 192.168.25.2 broadcast** 命令进行配置。**show atm vc** 命令显示了一个 ATM 路由器上的 VC 的配置，类似于帧中继网络上的 **show frame-relay pvc** 命令，如范例 4-12 所示。

范例 4-12 Fred 路由器上的 show atm vc 命令

```

Fred# show atm vc
          VCD /
Interface Name      VPI   VCI   Type   Encaps   SC      Peak  Avg/Min  Burst
0.4         1         4    482   PVC    SNAP    UBR    155000  Kbps    Cells  Sts
                                     UP

```

show atm vc 命令显示了 VC 所存在的接口、VCD 的名字、VPI 和 VCI 的号码、VC 的类型、封装、ATM 的服务类型、峰值速率、平均信元速率（都是 kbit/s）、以信元表示的突发速率以及 VC 的状态。所有的这些参数都必须匹配由服务提供商提供的信息或者说 ATM 交换机的配置。因此，在这个范例中，Fred 路由器的 ATM0.4 接口是未指定比特速率（UBR）电路，峰值速率为 155 000 kbit/s，全线速，电路起来了。范例 4-13 显示了 ATM 交换机的配置。

范例 4-13 ATM 交换机的配置

```

interface ATM1/0/0
 no ip address
 logging event subif-link status
 no atm ilmi-keepalive
!
interface ATM1/0/2
 no ip address
 logging event subif-link status
 no atm ilmi-keepalive
 atm pvc 4 482 interface ATM1/0/0 4 481

```

范例 4-14 显示了 Barney 和 Betty 帧中继路由器的配置，着重标注了帧中继的配置。可以将这个信息和先前在范例 4-10 到 4-13 中的 ATM 配置进行比较，来决定 ATM 和帧中继配置方法的相同点和不同点。

范例 4-14 Barney 和 Betty 路由器的配置

```

hostname Barney
!
interface Serial0/0
 no ip address
 encapsulation frame-relay IETF
 clockrate 1300000
 frame-relay lmi-type ansi
!
interface Serial0/0.651 multipoint
 ip address 192.168.26.1 255.255.255.252
 frame-relay map ip 192.168.26.2 651 broadcast
!
interface TokenRing0/0
 ip address 10.17.6.2 255.255.255.0
 ring-speed 16
 ip rsvp bandwidth 822 24
!
router eigrp 1911
 network 10.17.6.0 0.0.0.255
 network 192.168.26.0 0.0.0.3
 no auto-summary
 no eigrp log-neighbor-changes

```

（待续）

```
hostname Betty
!
interface Loopback100
 ip address 192.168.26.9 255.255.255.252
 no ip directed-broadcast
!
interface Serial0/1
 no ip address
 encapsulation frame-relay IETF
 clockrate 1300000
 frame-relay lmi-type ansi
!
interface Serial0/1.156 multipoint
 ip address 192.168.26.2 255.255.255.252
 frame-relay map ip 192.168.26.1 156 broadcast
!
router eigrp 1911
 network 192.168.26.8 0.0.0.3
 network 192.168.26.0 0.0.0.3
 no auto-summary
```

表 4-7 列出了 ATM 和帧中继的相同点和不同点。

表 4-7 ATM 和帧中继的比较

帧中继技术	技术描述	ATM 技术	技术描述
DLCI	识别帧中继 VC	VPI/VCI	识别 ATM VC
LMI	从帧中继交换机到 FRAD 进行第二层信令信息的通信	ILMI	ATM 交换机和路由器上的 ATM CPE 接口之间进行 ATM 信令信息的通信
串行子接口	用于建立逻辑的点对点或点对多点的帧中继电路接口	ATM 子接口	用于建立逻辑的点对点或点对多点的 ATM 电路接口
帧中继映射语句	将帧中继的 DLCI 映射成第三层的 IP 地址并且可选地启用 NBMA 虚假广播	映射列表或者协议映射（依赖于思科 IOS 软件版本或者配置喜好）	映射 ATM VCD 和 VPI/VCI 到第三层的 IP 地址并且可选地启用 NBMA 虚假广播
在串行接口上的帧中继封装	在串行接口上有 11 种可用的封装类型	ATM 接口封装	在 ATM 接口上惟一可用的封装类型。ATM 接口必须有硬件支持它。其他的 ATM 封装类型可以在 VC 的基础上选择和应用
CIR 和可选的 Be、Bc	帧中继对于虚拟电路的 QoS SLA	CBR, UBR, ABR, VBR	ATM 电路服务质量类型。在 ATM 交换机上，VC 是基于下面的服务类型构建的： CBR UBR ABR VBR-rt VBR-nrt
DE	帧中继 DE 位——用于将帧标记为可以丢弃的或者低优先级。DE=0 高优先级，DE=1 低优先级	CLP	ATM 信元丢失优先级（CLP）位——给信元标记一个优先级，用于在拥塞接口的情况下，可以将其先丢掉 CLP=0 高优先级，CLP=1 低优先级
FECN/BECN	前向和后向拥塞通知帧，由帧中继交换机发送，以标识拥塞	EFCI 和 ER	由 ATM ABR 等级采用的拥塞通知模式。显式前向拥塞标识（EFCI）是一种前向通知模式，后向拥塞通知采用显式速率（ER）模式
FRTS	帧中继流量整形（FRTS）使用帧中继的 CIR、Be 和 Bc 在帧中继的出口整形流量，以在拥塞的时候控制帧中继的帧丢失的情况	内置的 ATM 服务质量	ATM 电路本质上支持 ATM QoS 的某些模式。ATM 电路类型决定了所支持的服务质量类型，ATM 路由器的接口所使用的服务质量类型部分是由配置决定的

现在你已经看到我们比较了 ATM 和帧中继技术的不同，下一部分内容将介绍用户了解 ATM 服务质量机制，以及它们是如何利用思科 IOS 软件进行实施的。

二、ATM 服务质量

不像帧中继网络是基于可丢弃位来转发或者丢弃数据。ATM 网络有 4 种主要类型的服务，可在 ATM 交换机上配置：恒定比特率 (CBR)、可变比特率 (VBR)、未指定比特率 (UBR) 和可用比特率 (ABR)。其中两个主类还有子类；有两种方式的 VBR 电路：VBR 实时 (VBR-rt) 和 VBR 非实时 (VBR-nrt)，以及 UBR 和 UBR+。所有的这些选择通常都是在 ATM 交换机和 CPE 设备上配置。路由器有匹配的服务质量参数，允许路由器遵循交换机的配置。配置的电路类型取决于 SLA，并且电路的价格取决于要求的服务级别。每种类型的服务在拥塞期间有不同的行为，并且提供不同级别的服务，所以对网络所支持的流量类型在 ATM 网络中要做很好的规划。表 4-8 列出了 ATM 的服务类别和它们所支持的流量类型。

表 4-8 ATM 服务类别

ATM 服务分类	服务级别流量特性
CBR	提供恒定比特率，类似于物理电路。就像物理电路一样，CBR 电路不允许流量突发，当比特率超过时，任何超过的流量都会被丢掉 CBR 最适用于恒定比特率，不能容忍延迟的流量——例如服务提供商网络上恒定使用的语音或者视频流量。出于这个原因，CBR 电路通常不是为数据网络设计的
VBR-rt	VBR 实时推荐具有实时数据需求并且不能容忍延迟或者抖动的流量 VBR-rt 电路通常是语音或者视频网络设计的，它们不会时刻要求全线速率，特别适用于 VoIP 网络或者视频会议系统，它们通常不会要求恒定比特率
VBR-nrt	VBR 非实时电路提供可变速率的服务，它们支持流量突发，就像我们在数据网络中看到的那样 VBR-nrt 通常适用于企业网络中突发数据的应用程序，它们可以容忍可变速率的延迟，或者通过协议重传或者网络应用程序支持重传
UBR	UBR 通常是为只需要“尽力传递”的网络类型设计的 UBR 电路可以被认为类似于帧中继的 O-CIR 电路；取决于网络拥塞的情况，它可以提供尽量好的服务。UBR 电路适用于 WAN 电路中数据网络的应用程序支持延迟和重传或者是因特网流量
UBR+	UBR+电路并不是在 ATM 交换机上配置的，它特指在思科路由器上配置的 UBR 服务被称为 UBR+ UBR+配置允许用户在路由器上配置最小信元速率 (MCR)，它和 ATM 交换机通信。ATM 网络并不需要确保 UBR+的服务级别；他们还是需要和 ATM 服务提供商协商并且协定一个 SLA
ABR	ABR 电路在 ATM 网络中允许一个协商级别的服务，可提供 MCR，并且允许网络不拥塞时提供突发流量。在 ABR 电路中，ATM 网络提供一种基本服务，通过在资源管理 (RM) 信元中设置信息和和路由器进行网络状态信息的通信，使得路由器的 ATM 接口在低流量期间使用额外的网络资源

为了获得 ATM 服务类型的最大好处，某些 ATM 服务质量参数必须在路由器的 ATM 接口上配置。每一种 ATM 服务类型都有它自己的参数；这些参数在 PVC 配置模式下使用 ATM 服务类型命令配置。准确的配置值和可靠性将取决于 ATM 接口类型和 ATM 交换机的配置。在申请一个 ATM 电路之前，确保你有适当的 ATM 硬件来使用这个电路；某些平台只支持某些 ATM 电路类型。这一小节的剩余部分主要集中于思科 4500 和 4700 系列的 NP-1A-OC3 接口配置。许多适用于 4500 系列的命令也适用于其他较新的路由器。

三、配置 VBR-nrt 电路

就像名字所暗示的，VBR-nrt 电路设计用来支持不需要实时特性和能够容忍抖动和延迟

的流量类型。虽然不需要 ATM 服务级别的配置，路由器仍需配置以支持适当的 ATM 流量整形值，来提供 ATM 服务提供商所提供的服务级别。VBR-nrt VC 需要 3 个参数来适当地整形流量。这些参数如下：

- 保持信元速率（SCR）；
- 峰值信元速率（PCR）；
- 最大突发速率（MBS）。

每一个参数都是在 PVC 配置模式下使用下面的命令来配置的：

```
vbr-nrt pcr scr [ mbs]
```

PCR（以 kbit/s 描述）是 ATM 网络能够接受的绝对的峰值速率。接口使用这个数值来限制流量的峰值并且平滑流量，使得流量的突发不会在 ATM 网络中被丢掉。SCR 是 ATM 网络允许流量传输的持续速率。MBS（以信元衡量）是网络能够接受的最大突发量。

注意：当计算网络需求时，总是申请有空间剩余的电路以适应将来的增长，并且 ATM VC 总是应当配置为持续速率。永远不要设计网络使用峰值速率，否则可能导致网络不可用或者不稳定。

范例 4-15 和 4-16 显示了 ATM VBR-nrt 电路是如何在 Wilma 和 Fred 路由器上建立的。

范例 4-15 在 Wilma 路由器上使用 VBR-nrt

```
interface ATM0
  no ip address
  no atm ilmi-keepalive
  !
interface ATM0.4 multipoint
  ip address 192.168.25.1 255.255.255.252
  pvc 4/481
    protocol ip 192.168.25.2 broadcast
    vbr-nrt 44209 9000
    encapsulation aal5snap
```

范例 4-16 在 Fred 路由器上使用 VBR-nrt

```
interface ATM0
  no ip address
  no atm ilmi-keepalive
  !
interface ATM0.4 multipoint
  ip address 192.168.25.2 255.255.255.252
  pvc 4/482
    protocol ip 192.168.25.1 broadcast
    vbr-nrt 44209 9000
    encapsulation aal5snap
```

注意：如果 MBS 在配置中没有指定，正如在先前的范例中所示，路由器使用默认值。

可以使用扩展 ping 来测试这个配置，使用 **show atm pvc**、**show atm vc detail** 和 **show controller atm 0.4 | begin Packet switching** 命令，正如在范例 4-17 中的 Fred 路由器所示。

范例 4-17 在 Fred 路由器上使用 atm show 命令来验证 ATM 的配置

```
Fred# show atm pvc
VCD /
Interface Name      VPI  VCI  Type  Encaps  SC   Peak  Avg/Min  Burst
0.4      1      4    482  PVC   SNAP   VBR   44209   9000    95   UP
Fred# show atm vc detail
ATM0.4: VCD: 1, VPI: 4, VCI: 482
VBR-NRT, PeakRate: 44209, Average Rate: 9000, Burst Cells: 95
AAL5-LLC/SNAP, etype:0x0, Flags: 0x20, VMode: 0x401
OAM frequency: 0 second(s)
InARP frequency: 15 minutes(s)
InPkts: 329444, OutPkts: 329722, InBytes: 1169546091, OutBytes: 1169566161
InPRoc: 329444, OutPRoc: 328129, Broadcasts: 1593
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
Fred# show controllers atm 0.4 | begin Packet switching
Packet switching
  Fastswitched  0
  To-process    329564
  Bridged       0
Transmit errors
  Restarts      0
  Pktid misses  0
  Bad pktid     0
  Wrong queue   0
  No pkt        0
  Tx errors     0
  Bad VC        0
Receive errors
  Bad pktid     0
  Wrong queue   0
  No pkt        0
  CRC           0
  Length        0
  Giant         0
  Reas tout     0
  AAL5 format   0
```

在先前的范例中，**show atm pvc** 命令显示了 Fred 和 Wilma 路由器之间的 ATM PVC 的 VC 配置，并且 **show atm vc detail** 和 **show controller atm 0.4 | begin Packet switching** 命令验证了数据包没有任何错误成功地传输。

四、配置 UBR 和 UBR+电路

UBR 电路不确保从一个接口传输的所有流量一定会通过 ATM 网络传输。这些电路通常会用在两种情况下：通过网络传输的流量能够容忍延迟和抖动并且只需要尽力传递的服务，或者有费用上的限制不能使用一种更好的服务。标准的 UBR 电路只需要一个配置参数，**PCR**，并且在 PVC 配置模式下使用 **ubr pcr** 命令（pcr 以 kbit/s 衡量）配置。

UBR+电路也允许一个 MCR 值，以 kbit/s 衡量，允许支持峰值和最小信元速率。UBR+ 是在 PVC 配置模式下使用 **ubr+ pcr mcr** 命令配置。范例 4-18 显示了 ATM UBR+服务级别在 Fred 和 Wilma 路由器之间配置了一条额外的 100Mbit/s PVC 后是如何使用的。这个范例显示了 Fred 路由器的 PVC 配置。

范例 4-18 给 Mix 增加一个 UBR+ PVC

```
interface ATM0.5 multipoint
ip address 192.168.25.5 255.255.255.252
pvc 5/582
protocol ip 192.168.25.6 broadcast
ubr+ 106000 100000
encapsulation aal5snap
```

可以通过使用 **atm show** 命令验证这个配置。范例 4-19 显示了 Fred 路由器上使用 **show atm pvc** 和 **show atm vc vcd** 命令的输出。

范例 4-19 在 Fred 路由器上验证配置

```
Fred# show atm pvc
VCD /
Interface Name      VPI  VCI  Type  Encaps  SC   Peak Kbps  Avg/Min Kbps  Burst Cells  Sts
0.4        1          4   482  PVC   SNAP    VBR   44209   9000      95  UP
0.5        4          5   582  PVC   SNAP    UBR+ 106000 100000      UP

Fred# show atm vc 4
ATM0.5: VCD: 4, VPI: 5, VCI: 582
UBR+, PeakRate: 106000, Minimum Guaranteed Rate: 100000
AAL5-LLC/SNAP, etype:0x0, Flags: 0x20, VCmode: 0x1
OAM frequency: 0 second(s)
InARP frequency: 15 minutes(s)
InPkts: 9877, OutPkts: 9969, InBytes: 25996105, OutBytes: 26002689
InPRoc: 9877, OutPRoc: 9878, Broadcasts: 91
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
```

4.3 交换模式

路由器使用两种模式来决定路径和转发流量，即路由和交换。每一个协议使用一种路由方法来决定数据单元数据包、帧和信元的目的位置。三层和二层地址彼此映射，接着，如果配置了路由缓存，这个信息会保存在路由缓存里。若一个数据包的目的已知，而又启用了路由缓存，那么相关的信息会保存在路由缓存里。任何将来属于同一个流的数据包，含有相同的地址信息，都会使用路由缓存中的信息转发到它们的目的接口。然而，目的映射是基于每一个数据包的。将二层映射到三层的地址并且转发到目的接口的过程被称为交换。每一个接口有一个默认的交换方式；即使你没有显式地配置某个类型的交换，路由器也会使用它的默认方法来交换数据包。交换方法的有效性取决于你启用的特性和正在使用的交换模式。在讨论如何配置服务质量来提高现有网络的性能之前，验证路由器的接口正在使用最有效的交换方法是非常重要的。

注意：某些服务质量和安全技术有某些交换方法的需求。当选择一个服务质量方法时，总是记住规划所需的交换方法。

4.3.1 进程交换

取决于所安装的硬件和软件类型，不同的路由模型使用不同的交换模式。最基本的交换模式是**进程交换**。进程交换将一个流中的第一个数据包复制到系统缓冲区中。目的地址在路由表中查找。循环冗余校验码（CRC）使用路由处理器计算。接着数据包的二层信息被重新改写并且发送到目的接口。属于同一个流的后续数据包使用同样的三层-到-二层的路径被交换。进程交换在所有的交换类型中有较高的延迟，这是因为它使用系统缓冲区并且使用处理器来处理 and 存储它所接受的每一个数据包。进程交换可以通过关掉默认的快速或者最优交换来启用，使用命令 **no ip route-cache** 来关掉快速交换，增加 **optimum** 参数来关掉最优交换。进程交换在某些时候对于某些需要处理器参与的数据包处理进程是必需的，例如 **DEBUG IP** 数据包。

4.3.2 快速交换

快速交换使用路由缓存来存储数据流的信息。但快速交换启用后，数据流中第一个数据包的信息存储在数据包内存中，系统缓存中一个独立的区域。系统处理器用它来执行三层-到-二层的映射，接着路径信息存储在路由缓冲区中，使得同一股流中的后续数据包可以被快速交换。下一个数据包和同一股流中未来的数据包都会被快速交换。因为数据流的目的是已知的，在快速交换中，我们会查询路由缓存来找到目的接口。当目的找到并且在缓存中存储时，数据包会被进行适当的二层头改写，使用接口的处理器来计算 CRC。数据包永远不会中断系统处理器，这是因为目的接口的信息是已知的，而且系统缓存区不会用来存储数据包的信息。快速交换是许多思科路由器默认的交换方式，包括 1600、1700、2500 和 2600 的以太网、快速以太网和串行接口。如果快速交换被关掉了，可以通过 **ip route-cache** 命令在接口上启用。可以通过使用 **show ip cache** 命令监控快速交换。

4.3.3 最优或者分布式交换

两种其他的交换模式——不在 1600、1700、2500 或者 2600 系列的平台上提供——就是最优交换和分布式交换。最优交换和快速交换所遵循的过程是一样的，区别在于当第一个数据包被处理后，同一股流的所有后续数据包的路径信息就存储在最优交换缓存中，它更快。分布式交换需要使用一种灵活接口处理器（VIP）卡来处理进程交换信息。最优交换的方法也使用一种更有效的搜索算法来减少必须由 VIP 卡执行的查找时间。VIP 卡保持了一个路由缓存的拷贝，并且在本地执行所有的交换功能，使得接口不需要等待使用系统缓存中共享数据包的内存或者等待路由器。也可以安装多个 VIP 卡来进一步增加交换的性能。这使得分布式交换比快速交换或者最优交换更快。最优交换模式只在思科的高端路由器上可用，例如 7200。为了启用最优交换，必要时在每一个接口上使用 **ip route-cache optimum** 命令。为了监控或者故障排查最优交换的问题，使用 **show ip cache optimum** 命令。

4.3.4 NetFlow 交换

NetFlow 交换允许用户收集并且存储计费的 IP 数据，可以针对网络的利用率进行计费。*NetFlow* 交换使用默认的快速或者最优交换模式来转发 IP 流量；除了路由缓存之外，*NetFlow* 交换跟踪关于 IP 网络流量流的信息。可以基于用户、协议、端口和服务类型来跟踪流量。这些信息可以接着被导出到网管工作站上。*NetFlow* 交换可以执行我们先前所提到的快速、最优或者 CEF 交换模式。然而，当一个流已经建立后，属于同一个流的所有新的数据包对于 *NetFlow* 的接口，将绕过访问控制列表并且收集关于那个流的统计数字。

因为 *NetFlow* 计费数据存储在路由缓存里，*NetFlow* 交换数据采集进程对所有其他的网络设备都是透明的。然而，*NetFlow* 交换确实增加了路由器的处理器和内存的负荷，所以在执行这种交换方式之前，最好弄清楚需要多少内存。默认情况下，*NetFlow* 缓存对每股流使用 64 字节的缓存。如果使用了默认的 65 536 流，就需要 4 MB 的 DRAM 来支持每一个接口上的 *NetFlow* 进程。

注意：如果没有配置一个路由缓存方法并且 *NetFlow* 交换被启用了，默认的交换方式（CEF、快速或者最优）就会被启用。

NetFlow 交换可以使用 `ip route-cache flow` 命令在接口配置模式下启用，并且可以使用 `show ip cache flow` 命令监控。这个命令显示了接收到的不同尺寸的数据包的百分比，以字节表示的 *NetFlow* 缓存的大小，活动的流、不活动的流的数量，流分配的问题，以及详细的流信息，这也包括源和目的接口。为了将 *NetFlow* 的缓存表项导出到一个网管工作站，使用 `ip flow-export` 命令来指定工作站的地址和发送数据所使用的 UDP 端口号。

4.3.5 思科快速转发

思科快速转发是最有效地交换三层数据的方法。CEF 交换比快速或者最优交换更先进的原因是 CEF 交换对 CPU 消耗不大，它通过使用转发信息表（FIB）和邻接表来实现。FIB 查找表用于存储路由表中所有的已知路由，使用了一种更先进的搜索算法和数据结构，而不是像进程交换那样。不像其他的路由缓存交换方法，CEF 使用 FIB 表，它们可以随着网络拓扑的变化而调整。邻接表用来存储关于 CEF 邻居的信息。CEF 节点在彼此只有一跳时被认为是邻居。邻接表对每一个 FIB 的表项存储二层的下一跳的地址信息。每一个表项可能有不止一条路径，使得 CEF 可以在多条路径之间负载均衡来交换数据。每一次数据包在 CEF 启用的端口上收到后，就会查询 FIB 表来查找路由，封装二层数据，并且交换数据包。

CEF 交换可以在全局方式下使用 `ip cef` 命令启用。当 `ip cef` 命令在全局配置模式下启用后，CEF 就默认可以在所有的 CEF 启用的端口上启用。如果 CEF 在接口上关掉了，它可以通过在接口上使用 `ip route-cache cef` 命令重新启用，并且使用 `no version` 命令关掉。在高端路由器上也有一种分布式的 CEF，它在 `ip cef` 命令发出后可以默认就启用。可以使用 `show ip cef` 命令来监控 CEF，并且可以通过使用 `show ip cef detail` 命令来查看详细的 CEF 信息，如范例 4-20 所示。

范例 4-20 show ip cef detail 命令的输出

```
Router# show ip cef detail
IP CEF with switching (Table Version 10), flags=0x0
 10 routes, 0 reresolve, 0 unresolved (0 old, 0 new)
 13 leaves, 17 nodes, 19240 bytes, 13 inserts, 0 invalidations
 0 load sharing elements, 0 bytes, 0 references
 2 CEF resets, 0 revisions of existing leaves
 refcounts: 1061 leaf, 1058 node
Adjacency Table has 2 adjacencies
0.0.0.0/32, version 0, receive
1.1.1.1/32, version 6, connected, receive
35.132.253.0/24, version 7, attached, connected, cached adjacency to Serial0/2
0 packets, 0 bytes
  via Serial0/2, 0 dependencies
  valid cached adjacency
35.132.253.0/32, version 4, receive
35.132.253.1/32, version 3, receive
35.132.253.255/32, version 5, receive
167.56.24.0/24, version 8, attached, connected, cached adjacency to Serial0/1
0 packets, 0 bytes
  via Serial0/1, 0 dependencies
  valid cached adjacency
167.56.24.0/32, version 1, receive
167.56.24.31/32, version 0, receive
167.56.24.255/32, version 2, receive
224.0.0.0/4, version 9
0 packets, 0 bytes, Precedence routine (0)
  via 0.0.0.0, 0 dependencies
  next hop 0.0.0.0
  valid drop adjacency
224.0.0.0/24, version 2, receive
255.255.255.255/32, version 1, receive
```

一、CEF 负载均衡

就像先前所提到的，可以使用 CEF 在多条路径间负载均衡流量来交换数据包。CEF 通过配置可以实现基于每一个目的或者每一个数据包的负载均衡，这主要取决于网络的需求。基于每一个目的发送具有相同的源和目的的数据将使用相同的路径，在同一条路径上分发具有相同的源和目的流量。如果你使用基于每一个目的的负载均衡，具有相同的源和目的的数据在每个方向上将使用相同的路径。取决于反向路由器的配置，并不总是使用相同的返回路径。因为使用基于目的的负载均衡，可以确保数据包总是使用相同的路径，数据包总是按照它们发送的顺序到达目的。这种方式的负载均衡最适用于需要数据包按照某种顺序到达的应用，并且当 CEF 启用后是默认的负载均衡方式。如果你需要流量在多条路径上负载均衡，考虑使用基于每个数据包的负载均衡方式。然而记住这一点非常重要，基于每一个数据包的负载均衡方式并不能确保每一个数据包遵循相同的路径，这将导致数据包无序到达。基于每一个数据包的负载均衡方式在不均衡的流量需要在多条路径上进行负载均衡时是非常重要的。为了将基于每一个目的的负载均衡方式改为基于每一个数据包的负载均衡方式，需要关掉基于目的的方式，关掉在每一个需要的接口上使用的 **ip load-sharing per-destination** 命令，使用 **ip load-sharing per-packet** 命令来启用基于每一个数据包的负载均衡方式，如范例 4-21 所示。

范例 4-21 更改到基于每一个数据包的负载均衡方式

```
Router(config)# int serial 0/1
Router(config-if)# no ip load-sharing per-destination
Router(config-if)# ip load-sharing per-packet
Router(config)# int serial 0/2
Router(config-if)# no ip load-sharing per-destination
Router(config-if)# ip load-sharing per-packet
```

二、验证 CEF 的配置

为了决定当前接口所配置的交换模式，使用 **show ip interface** 命令。范例 4-22 显示了如何使用这个命令来显示接口 serial 0/1 的当前交换模式。按照 **show ip interface** 命令的输出，serial 0/1 当前正在使用默认的快速交换方式。在同一个接口上使用了快速交换和组播快速交换方式。流交换和分布式交换当前没有启用。

范例 4-22 查看当前的路由交换配置

```
Router# show ip interface serial 0/1
Serial0/1 is up, line protocol is up
  Internet address is 167.56.24.31/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is enabled
  IP Flow switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

为了启用 NetFlow 交换并且关闭组播路由缓存，在接口上使用 **ip route-cache flow** 命令，如范例 4-23 所示。

范例 4-23 改变路由交换配置

```
Router(config-if)#ip route-cache ?
cef          Enable Cisco Express Forwarding
flow         Enable Flow fast-switching cache
policy       Enable fast-switching policy cache for outgoing packets
same-interface Enable fast-switching on the same interface
<cr>
Router(config-if)#ip route-cache flow
Router(config-if)#^Z
Router# show ip int s 0/1
Serial0/1 is up, line protocol is up
  Internet address is 167.56.24.31/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is enabled
  IP Flow switching is enabled
  IP Flow switching turbo vector
  IP multicast fast switching is disabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

表 4-9 简短地描述了思科 IOS 软件中的每一种交换模式，并且列出了用于激活它们的命令。

表 4-9

交换模式

交换模式	描述	IP 交换命令
进程交换	每一个数据包都会被系统处理器和缓存处理，地址信息也是同样处理	no ip route-cache
快速交换	流中的第一个包是进程处理，流中以后的包通过路由缓存进行快速交换	ip route-cache
最优交换	流中的第一个包是进程处理，流中以后的包通过最优路由缓存进行快速交换	ip route-cache optimum
分布式交换	数据包在本地通过 VIP 卡处理，防止每一个数据包使用系统处理器、路由缓存或者缓冲区	ip route-cache distributed
NetFlow 交换	存储计费数据用于网络利用率采集和计费	ip route-cache flow

续表

交换模式	描述	IP 交换命令
CEF 交换	将三层路由信息存储在 FIB 表里，将二层的邻居信息存储在邻接表里。FIB 表里的拓扑信息会随着路由表而动态地变化。这使得 CEF 成为最有效的交换方式，因为在交换过程中没有使用进程交换	为了全局启用 CEF，使用： <code>ip cef</code> 在每一个接口上： <code>ip route-cache cef</code>

4.4 压 缩

另外一种可以增加传输数据包的数量方法就是通过压缩减少数据帧的大小。因为压缩的帧尺寸较小，可以通过介质传输更多的压缩帧，提高了传输时间。压缩可以基于硬件，也可以基于软件来做，这主要是取决于所安装的思科 IOS 的软件版本，接口的类型和正在使用的封装类型以及它所安装的硬件平台。本章只讨论软件压缩的技术，特别讨论了 STAC 和 Predictor 的压缩算法。

在任何路由器上启用压缩之前，检查处理器和内存的利用率非常重要。如果一台路由器的内存利用率超过了 40%，压缩并不是一种有用的方法。我们也需要关注 STAC 和 Predictor 都支持不同的封装协议，并且对内存和 CPU 具有不同的需求。表 4-10 总结了这些问题。

表 4-10 压缩问题

压缩方法	协议	系统需求
STAC	HDLC, PPP, LAPB, X.25	高 CPU 需求
Predictor	PPP, LAPB	高内存需求

被传输的流量总量、发送的数据包类型以及可用带宽的总量这些因素都会影响在路由器上执行压缩的效果。如果在一个主要用于下载数据的接口上压缩已经压缩过的数据，那么这种压缩是有害无益的，因为数据不能被压缩两次。如果一个接口的带宽很大，而传输的数据量很大，内存中用于压缩算法的字典空间的消耗就会非常大。为了检查内存的利用率，使用 `show memory summary` 命令，并且比较总体内存和空闲内存。如果没有足够的空闲内存，路由器就有可能不能处理压缩。为了验证 CPU 利用率，使用 `show process cpu` 命令；注意在一段时间内的处理器平均利用率。如果利用率持续超过 40%，那么压缩不是一种很好的性能解决方案。范例 4-24 显示了在 STAC 压缩算法启用前后路由器的处理器利用率。

范例 4-24 压缩如何影响利用率

```
Before STAC Compression
Lilo# show proc cpu
CPU utilization for five seconds: 2%/0%; one minute: 0%; five minutes: 4%
After STAC Compression
Lilo# show proc cpu
CPU utilization for five seconds: 44%/36%; one minute: 47%; five minutes: 25%
```

4.4.1 Stacker 压缩算法

Stacker 压缩算法也称为 STAC LZS，是一种基于 Lemple-Ziv 标准算法的压缩算法，它可以使用代码替换数据流中的字符。这些代码存储在字典中，含有符号代码的定义，用于将数据压缩成实际的数据字符。这个字典可以随着正在做压缩的流量的类型而不断地变化。

思科 IOS 软件支持对 PPP、LAPB、HDLC，帧中继和 X.25 接口的封装类型进行 STAC 的压缩算法。因为存储在内存中的动态 STAC 压缩字典会不断变化，所以监控运行了 STAC 算法的路由器的内存利用率是非常重要的。由于不断地检查数据包，使用 STAC 压缩算法的接口需要大量的可用处理器的时间。

为了在 PPP 或者 HDLC 封装的点对点接口上配置 STAC 压缩算法，可以在连接链路的两端使用 `compress stac` 命令。范例 4-25 显示了 STAC 是如何在 Lilo 和 Stitch 路由器之间的 HDLC 连接上使用 STAC 压缩算法的。

范例 4-25 STAC 压缩

```
hostname Lilo
!
interface Serial0/2
 ip address 175.25.25.1 255.255.255.0
 no ip directed-broadcast
 clockrate 1300000
 compress stac

-----

hostname Stitch
!
interface Serial0
 ip address 175.25.25.2 255.255.255.0
 no ip mroute-cache
 compress stac
```

为了验证 STAC 压缩算法的操作，使用 `show compress` 命令，如范例 4-26 所示。这个命令显示了关于压缩启用的接口的信息：在 1min、5min 和 10min 间隔内压缩的字节数；未压缩的统计情况以及发送和接收的压缩字节数。

范例 4-26 使用 show compress 命令

```
Router# show compress
Serial0/2
  Software compression enabled
  uncompressed bytes xmt/rcv 7313/6614
  1 min avg ratio xmt/rcv 0.000/0.992
  5 min avg ratio xmt/rcv 0.000/0.993
  10 min avg ratio xmt/rcv 0.000/0.926
  no bufs xmt 0 no bufs rcv 0
  resyncs 0
  Additional Stacker Stats:
    Transmit bytes:  Uncompressed = 18653960 Compressed = 6053
    Received bytes:  Compressed = 5604 Uncompressed = 0
```

4.4.2 Predictor 压缩算法

*Predictor 压缩算法*也是一个基于字典的压缩算法。然而，当处理数据时，Predictor 试图预测数据流中的下一段字符，使用的是压缩字典中的索引号，它存储这些序列号。如果数据流的下一段匹配第一段，那么存储在字典中的数据序列号将替代数据流中的数据序列号。这种预测方法使得 Predictor 在 CPU 使用方面更加有效，但是它要比 STAC 使用更多的内存。

为了在 PPP 或者 LAPB 封装的接口上启用 Predictor 压缩算法，使用 **compress predictor** 命令。范例 4-27 显示了 Predictor 压缩算法是如何在 Lilo 和 Stitch 路由器上使用的。

范例 4-27 使用 Predictor 压缩算法

```
hostname Lilo
!
interface Serial0/2
 ip address 175.25.25.1 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 clockrate 1300000
 compress predictor
!

hostname Stitch
!
interface Serial0
 ip address 175.25.25.2 255.255.255.0
 encapsulation ppp
 no ip mroute-cache
 compress predictor
```

为了检查 Predictor 压缩算法在思科 IOS 软件中的状态，使用 **show compress** 命令。范例 4-28 显示了 **show compress** 命令如何在 Predictor 启用的接口上显示相关的信息。**show compress** 命令显示了在 1min、5min 和 10min 的间隔内关于压缩/未压缩的字节数，并且显示了在 **no bufs** 区域中关于内存问题的故障排查信息。当连接的两端在字典中丢失了同步时，需要花费时间重新同步，这就增加了连接的延迟。关于字典重新同步的信息显示在字典的 **resyncs** 区域中。

范例 4-28 在 Predictor 中使用 show compress 命令

```
Lilo# show compress
Serial0/2
  Software compression enabled
  uncompressed bytes xmt/rcv 681/544
  1 min avg ratio xmt/rcv 0.414/0.328
  5 min avg ratio xmt/rcv 0.211/0.118
  10 min avg ratio xmt/rcv 0.211/0.118
  no bufs xmt 0 no bufs rcv 0
  resyncs 0
```

当运行两种压缩算法中的任何一个时，对每一台路由器监控处理器和内存的利用率不失为一个很好的主意。范例 4-29 显示了在 Lilo 和 Stitch 路由器上处理器和内存的利

用率的不同。注意两台路由器都经历了内存利用率的增加，但是在处理器的利用率方面增加不多。

范例 4-29 Predictor 的内存和 CPU 的利用率

```
Lilo Before Predictor
Lilo# show process cpu
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
Lilo# show mem sum
      Head      Total(b)    Used(b)    Free(b)    Lowest(b) Largest(b)
Processor 8148D770    5712016    3997864    1714152    1504420    1637856
I/O      1A00000    6291456    1909112    4382344    4382344    4382300
Lilo After Predictor
Lilo# show proc cpu
CPU utilization for five seconds: 1%/0%; one minute: 2%; five minutes: 0%
Lilo# show memory sum
      Head      Total(b)    Used(b)    Free(b)    Lowest(b) Largest(b)
Processor 8148D770    5712016    4132576    1579440    1504420    1506656
I/O      1A00000    6291456    1909112    4382344    4382344    4382300
Stitch Before Predictor
Stitch# show process cpu
CPU utilization for five seconds: 11%/11%; one minute: 2%; five minutes: 2%
Stitch# show memory sum
      Head      Total(b)    Used(b)    Free(b)    Lowest(b) Largest(b)
Processor 81257BA0    5932128    3578052    2354076    2149660    2228244
I/O      1800000    8388616    1746452    6642164    6642164    6642108
Stitch After Predictor
Stitch# show process cpu
CPU utilization for five seconds: 1%/0%; one minute: 2%; five minutes: 1%
Stitch# show memory sum
      Head      Total(b)    Used(b)    Free(b)    Lowest(b) Largest(b)
Processor 81257BA0    5932128    3711024    2221104    2149660    2097044
I/O      1800000    8388616    1746452    6642164    6642164    6642108
```

当验证路由器可以运行期望的软件之后，没有第一层的问题，路由器使用的是最有效的交换模式，你可能考虑使用压缩，你已经解决了所有可能影响路由器性能的基本问题。另外，影响路由器性能的更高级的做法是使用服务质量机制。

下两章将解释服务质量的不同类型是如何不同，它们如何配置和监控，以及每种服务质量机制在什么情况下工作是最好的。

4.4.3 实验 10：ATM 服务质量

本章集中于一些和质量有关的故障排查问题并且建立了一些实例，可以用于现场或者在实验性的环境中对网络应用程序提供更好级别的质量。本章的实验集中于 ATM 服务质量技术和它们的应用。

一、实验目的

这个实验的主要目的就是 ATM 技术和服务质量技术，然而，这个实验也可以提供下面这些技术的实践：

- 在 NBMA 网络中的 EIGRP 路由；
- 策略性路由；
- Voice over IP。

二、需要的设备

- 一个思科 LightStream 1010 ATM 交换机，带有两个 ATM OC-3 接口。
- 两个思科路由器，带有 ATM OC-3 接口和一个具有令牌环接口，一个具有串行接口。
- 一个思科路由器，具有一个令牌环接口和一个以太网接口。
- 一个思科路由器，具有一个以太网接口和一个 FXS 语音的模块。
- 一个思科路由器，具有一个串行接口和一个 FXS 语音的模块。
- 一个多工作站的访问单元（MSAU）和以太网交换机或者集线器。

三、物理布局和预规划

对于这个实验，可以使用图 4-4 所示的网络设计。City VetNet 路由器将通过 City 路由

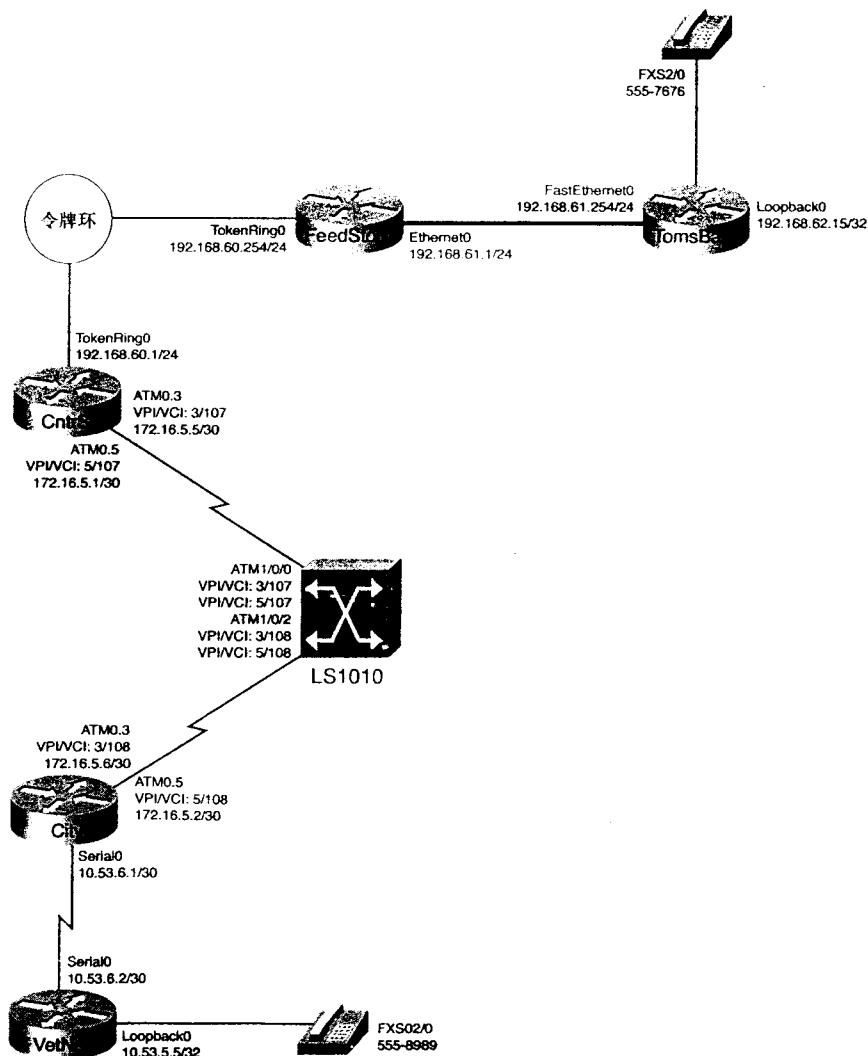


图 4-4 紧急兽医网络 (VetNet)

器和 Country Store (CntryStr) 路由器之间的 OC-3 连接到达 Tom's Barn (TomsBarn) 路由器。CntryStr 路由器通过令牌环网络连接到 Feed Store (FeedStore) 路由器。在 FeedStore 路由器和 TomsBarn 路由器之间使用的是 VoIP, TomsBarn 路由器和其余的网络是通过以太网连接使用 VoIP。

第 1 步 按照图 4-4 将路由器进行线缆连接。在进行其余的实验之前，验证所有的第一层连接都正常工作。

第 2 步 按照表 4-11 所示的接口和 VPI/VCI 对来配置 ATM 交换机。

表 4-11

ATM 交换机配置

交换机 ATM 接口	VPI	VCI	路由器的名字和接口
ATM 1/0/0	3	107	CntryStr—ATM0.3
ATM 1/0/0	5	107	CntryStr—ATM0.5
ATM 1/0/2	3	108	City—ATM0.3
ATM 1/0/2	5	108	City—ATM0.5

范例 4-30 显示了 ATM 的配置和从 ATM 交换机上执行 **show atm vc** 命令的输出。

范例 4-30 ATM VC 配置

```
ATM-Switch# show atm vc interface atm 1/0/0
hostname ATM-Switch
!
interface ATM1/0/0
no ip address
!
interface ATM1/0/1
no ip address
!
interface ATM1/0/2
no ip address
atm pvc 3 108 interface ATM1/0/0 3 107
atm pvc 5 108 interface ATM1/0/0 5 107
!
CRLF
ATM-Switch# show atm vc interface atm 1/0/0
Interface      VPI  VCI  Type  X-Interface      X-VPI X-VCI Encap  Status
ATM1/0/0       0    5    PVC   ATM2/0/0         0     39   QSAAL  UP
ATM1/0/0       0    16   PVC   ATM2/0/0         0     35   ILMI   UP
ATM1/0/0       3    107  PVC   ATM1/0/2         3     108   UP
ATM1/0/0       5    107  PVC   ATM1/0/2         5     108   UP
ATM-Switch# show atm vc interface atm 1/0/2
Interface      VPI  VCI  Type  X-Interface      X-VPI X-VCI Encap  Status
ATM1/0/2       0    5    PVC   ATM2/0/0         0     41   QSAAL  UP
ATM1/0/2       0    16   PVC   ATM2/0/0         0     37   ILMI   UP
ATM1/0/2       3    108  PVC   ATM1/0/0         3     107   UP
ATM1/0/2       5    108  PVC   ATM1/0/0         5     107   UP
```

四、实验练习

第 1 步 按照表 4-12 所示配置所有的 IP 地址。在进行第 2 步之前，确保所有的路由器能够 ping 通它们所直连的邻居的接口。配置 ATM 的接口，使用最适合的 ATM 封装类型来突发数据流量。

表 4-12 这个网络模型的 IP 地址

路由器的名字	路由器的接口	IP 地址	路由器的名字	路由器的接口	IP 地址
TomsBarn	FastEthernet0	192.168.61.254/24	City	ATM0.3	172.16.5.6/30
	Loopback0	192.168.62.15/32		ATM0.5	172.16.5.2/30
FeedStore	Ethernet0/0	192.168.61.1/24		Serial2	10.53.6.1/30
	TokenRing0/0	192.168.60.254/24	VetNet	Serial1	10.53.6.2/30
CntryStr	ATM0.3	172.16.5.5/30		Loopback0	10.53.5.5/32
	ATM0.5	172.16.5.1/30			
	TokenRing0	192.168.60.1/24			

第 2 步 对所有的路由器配置 EIGRP 路由，将所有的路由器接口放入 EIGRP AS 62。确保 EIGRP 路由器只宣告最特定的路由，不要允许自动汇总，在进行第 3 步之前，验证所有的路由器能够 ping 通 TomsBarn 和 VetNet 路由器上的环回接口。

第 3 步 在 City 和 CntryStr 路由器上配置 ATM PVC，使得在接口 ATM0.3 上的 PVC 将使用未指定比特速率服务级别，具有最大突发速率 149 344 Mbit/s，和最小的信元速率 44 209 Mbit/s，而接口 0.5 将具有非实时的可变比特速率服务级别，具有最大的突发速率 6.176 Mbit/s 和最小的确保速率 1.544 Mbit/s。

第 4 步 在 TomsBarn 和 VetNet 路由器之间配置 Voice over IP。使用 loopback0 IP 地址作为会话的目的，并且使用 FXS voice port 2/0 来作为电话接口。在连接两台路由器的电话上发起测试的呼叫来验证配置。

第 5 步 在 CntryStr 和 City 路由器上配置策略性路由，使得语音，而且只有语音流量（包括呼叫建立）通过 1.5 Mbit/s ATM 的接口传送。验证语音流量通过适当的接口传送。

第 6 步 在 VetNet 和 City 路由器之间的串行线路上新的 OC-3 导致了瓶颈。在这些路由器上启用压缩，采用最节省 CPU 利用率的压缩方法。当你能够成功地从 TomsBarn 和 VetNet 电话之间进行呼叫时，这个实验就已成功地完成了。

五、实验步骤

第 1 步 按照表 4-12 所示，配置所有的 IP 地址。在进行第 3 步之前，确保所有的路由器能够 ping 通它们直连邻居的接口。配置 ATM 接口，使用最适合于突发数据流的 ATM 封装类型。

AAL5Snap 是专门为突发数据流量需求而建立的 ATM 封装类型。AAL5Snap 封装是使用 **encapsulation aal5snap** 命令在 PVC 配置模式下配置的，如 CntryStr 路由器在范例 4-31 中所示。

第 2 步 对所有的路由器配置 EIGRP 路由，并且将所有路由器的接口放入 EIGRP AS 62 中。确保 EIGRP 路由器只通告最特定的路由，不要允许自动汇总。不能在这个实验中使用 EIGRP neighbor 语句。在进行第 3 步之前，要确保所有的路由器能够 ping 通 TomsBarn 和 VetNet 路由器的环回接口。

范例 4-31 使用 AAL5Snap Encapsulation

```
interface ATM0
  no ip address
  no atm ilmi-keepalive
!
interface ATM0.3 multipoint
  ip address 172.16.5.5 255.255.255.252
  pvc 3/107
    protocol ip 172.16.5.6 broadcast
    encapsulation aal5snap
  !
!
interface ATM0.5 multipoint
  ip address 172.16.5.1 255.255.255.252
  pvc 5/107
    protocol ip 172.16.5.2 broadcast
    encapsulation aal5snap
```

有两种方法使得 EIGRP 的邻居在 NMBA 的网络上聚合。第一种方法是在对等体之间使用 EIGRP 邻居语句来配置静态的邻居关系，但这种方法在我们这个实验中是不允许的。第二种方法是使用虚假广播方式来建立二层-到-三层的协议映射，以允许 ATM 接口使用虚假广播，允许 EIGRP 在 NBMA 的网络上聚合。这一步需要准确的 ATM 配置才能工作。回忆一下，在本章的 ATM 的回顾部分，在新版本的思科 IOS 软件中，在 ATM 网络上，二层-到-三层的协议映射是使用 **protocol ip ip address broadcast** 语句在 PVC 配置模式中对 ATM 子接口下建立的。范例 4-32 显示了对 City 路由器的 ATM 配置。

范例 4-32 City 路由器的 ATM 配置

```
interface ATM0
  no ip address
  no atm ilmi-keepalive
!
interface ATM0.3 multipoint
  ip address 172.16.5.6 255.255.255.252
  pvc 3/108
    protocol ip 172.16.5.5 broadcast
    encapsulation aal5snap
  !
!
interface ATM0.5 multipoint
  ip address 172.16.5.2 255.255.255.252
  pvc 5/108
    protocol ip 172.16.5.1 broadcast
    encapsulation aal5snap
```

当验证完 ATM 的配置后，可以通过发出 **show atm map** 命令并且验证每一条 PVC 都有相关的广播语句来检查 NMBA 的广播支持，正如在范例 4-33 中 City 路由器的配置所示。

范例 4-33 在 City 路由器上的 show atm map

```
City# show atm map
Map list ATM0.3pvc1 : PERMANENT
ip 172.16.5.5 maps to VC 1, VPI 3, VCI 108, ATM0.3
, broadcast
Map list ATM0.5pvc2 : PERMANENT
ip 172.16.5.1 maps to VC 2, VPI 5, VCI 108, ATM0.5
, broadcast
```

可以从 TomsBarn 和 VetNet 路由器上 ping 环回接口来验证 EIGRP 的配置在正常工作。范例 4-34 显示了 CntryStr 路由器的 EIGRP 配置，以及从 TomsBarn 和 VetNet 路由器发出的 ping。

范例 4-34 CntryStr 路由器的 EIGRP 配置

```
router eigrp 62
network 172.16.5.0 0.0.0.3
network 172.16.5.4 0.0.0.3
network 192.168.60.0
no auto-summary
TomsBarn# ping 10.53.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.53.5.5, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
VetNet# ping 192.168.62.15

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.62.15, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
```

注意：早期的思科 IOS 软件版本需要不依赖于 PVC 的 map 列表来将二层映射成三层协议。如果你想使用它们，这些命令在新的软件版本中依然存在。

第3步 在 City 和 CntryStr 路由器上配置 ATM PVC，使得在接口 ATM0.3 上的 PVC 将使用未指定比特速率服务级别，具有最大突发速率 149 344 Mbit/s 和最小的信元速率 44 209 Mbit/s，而接口 0.5 将具有非实时的可变比特速率服务级别，具有最大的突发速率 6.176 Mbit/s 和最小的确保速率 1.544 Mbit/s。这个实验的 ATM 流量整形练习需要用户在 City 和 CntryStr 路由器上使用不同级别的 ATM 服务来配置每一条 PVC。第一条，较大的 PVC 被设置为使用持续速率 45 Mbit/s (DS3)，它也能够突发到 150 Mbit/s；这是通过在 ATM 3/107 和 3/108 PVC 上使用 UBR+ ATM 的服务级别实现的。这个配置可以通过使用 show atm vc 命令来验证。范例 4-35 显示了 City 路由器上的 UBR+ 配置。

范例 4-35 City 路由器的 UBR+配置

```
interface ATM0.3 multipoint
ip address 172.16.5.6 255.255.255.252
pvc 3/108
protocol ip 172.16.5.5 broadcast
ubr+ 149344 44209
encapsulation aal5snap
```

```
City# show atm vc
          VCD /
Interface Name      VPI  VCI  Type  Encaps  SC   Peak  Avg/Min Burst  Cells  Sts
0.3        1         3   108  PVC   SNAP   UBR+ 149344 44209          UP
```

第二条，带宽较小的、T1 尺寸的 PVC 应当配置使用 VBR-nrt 的服务级别，PCR 为 6 176 kbit/s，SCR 为 1 544 kbit/s，如范例 4-36 中的 CntryStr 路由器所示。

范例 4-36 CntryStr 路由器的 VBR-nrt 配置

```
interface ATM0.5 multipoint
ip address 172.16.5.1 255.255.255.252
pvc 5/107
protocol ip 172.16.5.2 broadcast
vbr-nrt 6176 1544
encapsulation aal5snap
```

注意：即使改变了 ATM 接口的 ATM 服务级别，使用 **show interface** 命令时看到的带宽参数也不会改变。记住，使用 **show interface** 命令看到的带宽参数只是对于这个接口的 EIGRP 的带宽度量。

第 4 步 在 TomsBarn 和 VetNet 路由器之间配置 Voice over IP。使用 loopback0 IP 地址作为会话的目的，并且使用 FXS voice port 2/0 作为电话接口。在连接两台路由器的电话上发起测试的呼叫来验证配置。

这一步是非常直观的，假设到目前为止所有的配置都正常工作。现在要做的是在每一台路由器上建立两个 dial peer，并且将会话目的和端口设置为本地和远端的 dial peer。范例 4-37 显示了对 TomsBarn 路由器的语音配置。

范例 4-37 TomsBarn Voice over IP 配置

```
dial-peer voice 5557676 pots
destination-pattern 5557676
port 2/0
!
dial-peer voice 5558989 voip
destination-pattern 5558989
session target ipv4:10.53.5.5
```

第 5 步 在 CntryStr 和 City 路由器上配置策略性路由，使得语音，而且只有语音流量（包括呼叫建立）通过 1.5Mbit/s ATM 的接口传送。验证语音流量通过适当的接口传送。

这一步需要一些任务才能工作正常。首先，在其中的一台路由器上，在我们的这个范例中，可以使用 CntryStr 路由器，建立一个访问控制列表匹配来自 Toms Barn 路由器的语音流量。接着，建立一个路由映射来匹配上述的流量并且将它发送到接口 ATM 0.5。然后，进行测试，如果需要，使用 **debug ip policy** 命令来调整配置并且测试从 TomsBarn 发出的电话呼叫。最后，当配置正确后，重复相同的步骤在 City 路由器上进行配置。范例 4-38 显示了 CntryStr 路由器上的策略性路由配置。

范例 4-38 CntryStr 路由器的策略性路由配置

```
interface TokenRing0
 ip address 192.168.60.1 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 ip policy route-map voice-traffic
 ring-speed 16
!
access-list 150 permit tcp host 192.168.61.254 host 10.53.6.2 eq 1720
access-list 150 permit tcp host 192.168.61.254 eq 1720 host 10.53.6.2
access-list 150 permit tcp host 192.168.61.254 host 10.53.5.5 eq 1720
access-list 150 permit udp host 192.168.61.254 host 10.53.6.2 range 16384 32767
route-map voice-traffic permit 10
 match ip address 150
 set interface ATM0.5
```

在先前的范例中，头三行指定在两台路由器之间的 H.323 呼叫建立流量，最后一行指定 RTP 的语音流量。路由映射中匹配访问控制列表 150 的语音流量会被发送到接口 ATM 0.5。可以通过从 TomsBarn 路由器发起到 VetNet 路由器的测试呼叫来验证这一切，使用 **show route-map** 和 **debug ip policy** 来显示匹配的策略，如范例 4-39 所示。

范例 4-39 show route-map 和 debug ip policy

```
CntryStr#show route-map voice-traffic
route-map voice-traffic, permit, sequence 10
 Match clauses:
  ip address (access-lists): 150
 Set clauses:
  interface ATM0.5
 Policy routing matches: 3942 packets, 328996 bytes
02:24:57: IP: s=192.168.61.254 (TokenRing0), d=10.53.5.5, len 346, policy match
02:24:57: IP: route map voice-traffic, item 10, permit
02:24:57: IP: s=192.168.61.254 (TokenRing0), d=10.53.5.5 (ATM0.5), len 346, policy
routed
02:24:57: IP: TokenRing0 to ATM0.5 172.16.5.2
02:24:58: IP: s=192.168.61.254 (TokenRing0), d=10.53.5.5, len 40, policy match
02:24:58: IP: route map voice-traffic, item 10, permit
02:24:58: IP: s=192.168.61.254 (TokenRing0), d=10.53.5.5 (ATM0.5), len 40, policy
routed
02:24:58: IP: TokenRing0 to ATM0.5 172.16.5.2
02:24:58: IP: s=192.168.61.254 (TokenRing0), d=10.53.6.2, len 60, policy match
02:24:58: IP: route map voice-traffic, item 10, permit
02:24:58: IP: s=192.168.61.254 (TokenRing0), d=10.53.6.2 (ATM0.5), len 60, policy
routed
```

第6步 在 VetNet 和 City 路由器之间的串行线路上新的 OC-3 导致了瓶颈。在这些

路由器上启用压缩，采用最节省 CPU 利用率的压缩方法。当你能够成功地从 TomsBarn 和 VetNet 电话之间进行呼叫时，这个实验就已成功地完成了。Predictor 压缩算法是最有效的使用路由器的 CPU 资源的方法。然而，在使用 Predictor 压缩算法之前，必须使用 PPP 封装。当你在 City 和 VetNet 路由器上配置 PPP 和 Predictor 压缩算法之后，应当能够在 TomsBarn 和 VetNet 路由器之间建立成功的测试呼叫。范例 4-40 显示了对 VetNet 路由器的压缩配置。

范例 4-40 VetNet 压缩配置

```
interface Serial1
 ip address 10.53.6.2 255.255.255.252
 encapsulation ppp
 clockrate 13000000
 compress predictor
```

当你完成测试呼叫后，这个实验就完成了。将你的路由器配置和范例 4-41 所显示的配置进行对比。

范例 4-41 这个实验的完整的路由器配置

```
hostname TomsBarn
!
ip cef
!
interface Loopback0
 ip address 192.168.62.15 255.255.255.255
!
interface FastEthernet0
 ip address 192.168.61.254 255.255.255.0
!
router eigrp 62
 network 192.168.61.0
 network 192.168.62.15 0.0.0.0
 no auto-summary
!
ip classless
!
dial-peer voice 5557676 pots
 destination-pattern 5557676
 port 2/0
!
dial-peer voice 5558989 voip
 destination-pattern 5558989
 session target ipv4:10.53.5.5

hostname FeedStore
!
ip cef
!
interface Ethernet0/0
 ip address 192.168.61.1 255.255.255.0
!
interface TokenRing0/0
 ip address 192.168.60.254 255.255.255.0
 ring-speed 16
```

(待续)

```
!  
router eigrp 62  
network 192.168.60.0  
network 192.168.61.0  
no auto-summary  
  
hostname CntryStr  
!  
ip cef  
!  
interface TokenRing0  
ip address 192.168.60.1 255.255.255.0  
ip route-cache policy  
no ip route-cache cef  
ip policy route-map voice-traffic  
ring-speed 16  
!  
interface ATM0  
no ip address  
no atm ilmi-keepalive  
!  
interface ATM0.3 multipoint  
ip address 172.16.5.5 255.255.255.252  
pvc 3/107  
protocol ip 172.16.5.6 broadcast  
ubr+ 149344 44209  
encapsulation aal5snap  
!  
interface ATM0.5 multipoint  
ip address 172.16.5.1 255.255.255.252  
pvc 5/107  
protocol ip 172.16.5.2 broadcast  
vbr-nrt 6176 1544  
encapsulation aal5snap  
!  
router eigrp 62  
network 172.16.5.0 0.0.0.3  
network 172.16.5.4 0.0.0.3  
network 192.168.60.0  
no auto-summary  
!  
ip classless  
!  
access-list 150 permit tcp host 192.168.61.254 host 10.53.6.2 eq 1720  
access-list 150 permit tcp host 192.168.61.254 eq 1720 host 10.53.6.2  
access-list 150 permit tcp host 192.168.61.254 host 10.53.5.5 eq 1720  
access-list 150 permit udp host 192.168.61.254 host 10.53.6.2 range 16384 32767  
route-map voice-traffic permit 10  
match ip address 150  
set interface ATM0.5  
  
hostname City  
!  
ip cef  
!  
interface Serial0  
ip address 10.53.6.1 255.255.255.252  
encapsulation ppp  
no ip route-cache cef  
ip policy route-map voice-traffic  
compress predictor
```

(待续)

```
!  
interface ATM0  
  no ip address  
  no atm ilmi-keepalive  
!  
interface ATM0.3 multipoint  
  ip address 172.16.5.6 255.255.255.252  
  pvc 3/108  
    protocol ip 172.16.5.5 broadcast  
    vbr+ 149344 44209  
    encapsulation aal5snap  
!  
interface ATM0.5 multipoint  
  ip address 172.16.5.2 255.255.255.252  
  pvc 5/108  
    protocol ip 172.16.5.1 broadcast  
    vbr-nrt 6176 1544  
    encapsulation aal5snap  
!  
router eigrp 62  
  network 10.53.6.0 0.0.0.3  
  network 172.16.5.0 0.0.0.3  
  network 172.16.5.4 0.0.0.3  
  no auto-summary  
!  
ip classless  
!  
access-list 1 deny 172.16.5.4 0.0.0.3  
access-list 1 permit any  
access-list 150 permit tcp host 10.53.6.2 host 192.168.61.254 eq 1720  
access-list 150 permit tcp host 10.53.6.2 eq 1720 host 192.168.61.254  
access-list 150 permit tcp host 10.53.6.2 host 192.168.62.15 eq 1720  
access-list 150 permit udp host 10.53.6.2 host 192.168.61.254 range 16384 32767  
route-map voice-traffic permit 10  
  match ip address 150  
  set interface ATM0.5  
  
hostname VetNet  
!  
interface Loopback0  
  ip address 10.53.5.5 255.255.255.255  
!  
interface Serial0  
  ip address 10.53.6.2 255.255.255.252  
  encapsulation ppp  
  clockrate 1300000  
  compress predictor  
!  
router eigrp 62  
  network 10.53.5.5 0.0.0.0  
  network 10.53.6.0 0.0.0.3  
  no auto-summary  
!  
ip classless  
!  
dial-peer voice 5558989 pots  
  destination-pattern 5558989  
  port 2/0  
!  
dial-peer voice 5557676 voip  
  destination-pattern 5557676  
  session target ipv4:192.168.62.15
```

4.5 进一步阅读资料

RFC 2330, *Framework for IP Performance Metrics*, by Paul L. Della Maggiora, Christopher E. Elliott, Robert L. Pavone, Jr., Kent J. Phelps, and James M. Thompson.

Network Consultants Handbook, by Matthew J. Castelli.

Internetworking Troubleshooting Handbook, Second Edition, by Cisco Systems.

第 5 章

集成和差分服务

前面的章节探讨了路由器的性能并且检查了几种路由交换机制，可以使用它们来减少由于错误和设备的资源利用率所带来的延迟和抖动问题，从而提供一定级别的服务质量（QoS）。本章集中介绍由集成和区分服务所带来的更灵活的服务质量技术。本章介绍了下面的这些主题：

- 如何使用资源预留协议（RSVP）提供一个确保级别的服务。
- 如何使用内置的 IP 服务类型位（TOS）的优先级级别标记流量。
- 如何使用 IP precedence 位来优化流量。
- 如何使用新的差分服务编码点位来实现高级的流量分类和标记。

当分析这些主题时，本章还使用了实际的范例来应用这些技术，并且使用了实际的实验来获取对真实应用的经验。

5.1 集成服务

集成服务通常也被称为 IntServ，是一种提供端对端服务质量的体系结构。IntServ 的解决方案允许终端工作站向网络发出质量请求；参与这种服务质量机制的网络对请求的工作站要么预留要么不预留网络资源。集成服务的体系结构提供了一种方法来确保网络的质量级别，通过指定预留的服务并且控制设备上流量的负荷来提供确保的服务需求。在集成服务体系结构中最常用的实施方案就是 RSVP 信令协议。

使用 RSVP 的带宽预留协议

RSVP 也称为资源预留建立协议，在 RFC 2205 中定义，它作为一种信令协议主要用于资源预留，可以提供一个端到端的服务质量预留，由请求的主机或者应用程序发起。RSVP 支持组播或者单播的 IP 流。流基本上被定义为从一个特定的 IP 地址、协议类型和端口号到一个特定的 IP 地址或者组播组的特定端口号或特定协议类型的流量。因为流是基于源和目的的协议信息来定义的，每一个流可以提供主机之间的会话的单向描述。使用 RSVP，允许实时应用程序指定这个应用程序可以运作的网络质量的参数。在 RSVP 中，主机通常请求特定的服务质量特性，并且主机之间的路由器可以提供这些服务。非常关键的一点是，我们要注意 RSVP 请求是从发起主机到目的的单向流，路径中的每一个设备都参与 RSVP 会话。RSVP 使用路由表中的信息来发现到达目的的路由。使用路由表中提供的信息和不同的消息类型，RSVP 动态地随着网络条件而调整。

RSVP 也发送周期性的刷新消息用于维护 RSVP 状态。如果消息在一个特定的时间内没有收到，这个时间在 RSVP 请求消息中定义，RSVP 状态超时并且预留被删除。

RSVP 请求使用流标准也称为 *flowspec* 和 *filter spec* 来形成一个流描述。流描述用于描述流的特性。*flowspec* 定义了请求主机的质量需求。数据包的分发器使用 *flowspec* 所提供的信息来决定分发的需求。对于流，*filter spec* 用于定义主机数据包分类的需求。数据包的分发器决定了什么时候数据包被转发，而数据包的分类器决定了在流中的数据包的服务质量特性。

在 RSVP 中有两种流预留的分类：独立预留和共享预留。独立预留是被流定义的，由每一个发送者发起并且对每一个发送者预留。共享预留可能由一个发送者或者多个发送者发起。独立预留为每一个请求独立预留的发送者建立。而在共享预留中，对于所有的发送者来说只有一个预留被建立并共享。共享预留类型通常是用于应用程序。表 5-1 汇总了 RSVP 预留类型并简单地描述了它们的应用程序。

表 5-1 RSVP 预留类型

预留类型	描述
独立预留	一个发送者发起流量流
共享预留	至少一个发送者发起流量流。通常这些流不会同时操作，因此，它们可以共享相同的预留

RSVP 预留使用两种类型的列表来定义发送者组。显式发送者选择列表使用 *filter spec* 来指定发送者，它定义单个的发送者，而通配符列表指定了使用相同的 *filter spec*，使用了相同的服务质量特性的发送者。显式发送者对独立预留使用固定过滤器（FF）风格，或者使用对共享预留共享显式（SE）风格。通配符发送者使用通配符过滤器（WF）来做共享预留并且对独立预留没有定义。表 5-2 显示了过滤类型是如何与发送者列表匹配的，以及每一种风格的特色。

表 5-2 RSVP 预留风格

过滤风格	描述	预留类型	发送者选择
Wildcard-filter style (WF)	使用一个单独的预留，被多个流共享	共享	通配符
Fixed-filter style (FF)	使用一个单独的预留，被来自一个流的所有数据包共享	独立	显式
Shared-explicit style (SE)	被来自多个源的组播应用程序的流使用	共享	显式

就像先前所提到的，RSVP 是一个端对端的服务质量模型，这也就是说在 RSVP 路径中的每一个设备都必须从另外一个设备请求资源。在路径中每一个 RSVP 启用 (RSVP-enabled) 的路由器在批准这个请求时必须考虑两个条件：路由器自身是否有足够的资源来提供所请求的资源；所请求的主机是否有权限来做出预留。这些决定是基于准许控制模块和策略控制模块做出的。*准许控制模块*让路由器决定是否有足够的资源给请求者，而*策略控制模块*决定请求的主机是否有权限来请求这个服务。如果两个条件都满足，就会产生资源预留。如果两个条件中有一个不满足，路由器就拒绝预留请求，但是流量还是按照常规的服务进行传送。RSVP 使用几种消息类型来传送预留请求和预留请求的参数。这些消息类型会简短地介绍，在本章描述建立 RSVP 路径的步骤之后。为了建立 RSVP 路径，遵循下面的这些步骤：

- 第 1 步 RSVP 发送者（请求服务的主机）发送一个 RSVP PATH 消息来描述试图发送的数据。
- 第 2 步 在到达目的的路径中每一台 RSVP 路由器读取 RSVP 消息，存储关于前一跳的 IP 地址的信息，并将自己的地址添加到消息中作为前一跳，接着将这个信息传送到下一台路由器。
- 第 3 步 接收主机收到 PATH 信息。
- 第 4 步 当读完 PATH 消息后，RSVP 接收者反向发送者主机请求资源预留，使用准确的反向路径并且使用 RSVP RESV 消息。
- 第 5 步 RSVP 启用的路由器如果没有足够的资源，要么拒绝 RSVP 请求，要么同意请求并且从下一台路由器请求预留（反向路径）。
- 第 6 步 原始的发送主机从最近的下一跳路由器（预留资源的路由器）接收请求并且使用预留的路径。

图 5-1 显示了 RSVP 会话是如何使用 RSVP PATH 和 RESV 消息来建立的。

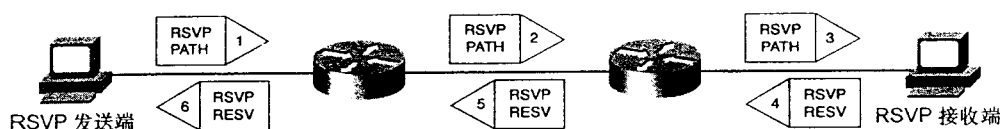


图 5-1 RSVP 会话建立图

当使用 RSVP 协议时，记住一些关键的术语。*RSVP 发送者*指的是发起 RSVP 预留的主机。*RSVP 接收者*是资源被预留的主机。任何在发送者和接收者之间配置运行 RSVP 协议的路由器被称为 *RSVP 启用 (RSVP-enabled) 的路由器*。

资源预留路径 (RSVP PATH) 消息，这个消息最初是由 RSVP 发送者发送的，请求一个预留，列出了用于到达 RSVP 接收者的 RSVP 路径中所有的跳数。RSVP PATH 消息使得每一

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

个 RSVP 启用的路由器为每一个预留请求存储 RSVP 状态成为可能。资源预留请求 (RSVP RESV) 消息由接收主机发送并且被路径中每一个 RSVP 启用的路由器处理，直到它到达目的主机，也就是 RSVP 发送者。发送者接收和回应 RESV 消息，使用准确的反向路径，也就是最初发送请求的路径，将它们发送回接收主机。

注意：由 RSVP 接收者发送的原始的 RSVP RESV 消息并不是在所有的情况下总是被发送回发送主机。如果多个接收者发送 RSVP RESV 消息，在某一点上这些预留的 flowspec 合并，只有最大的 flowspec 会通过所有的路径转发到发送者。RSVP RESV 消息也可能包括对资源预留的确认的请求。在这种情况下，要么发送者要么 RSVP 路由器 (在 flowspec 合并点) 给接收者发送一个确认。

RSVP 使用 IP 协议来实现它所有的通信。因为 IP 不是一个可靠的协议，有时 RSVP 消息将不会被所有需要的设备接收到。为了解决这个问题，RSVP 使用原始的 PATH 消息中指定的 hello 间隔来发送周期性的刷新消息。这些消息是使用 RSVP PATH 和 RESV 消息发送的。当一个发送者主机应用程序使用资源完成后，它应当发出 RSVP TEARDOWN 消息。下一跳的路由器接收到 TEARDOWN 消息后，清除预留，并且沿路径给下一跳的路由器发送 TEARDOWN 消息。使用 RSVP TEARDOWN 消息并不局限于 RSVP 发送主机，在任何时候，只要它决定结束 RSVP 会话，RSVP 接收者也可以发送一个 TEARDOWN 消息。在 TEARDOWN 消息丢失的情况下，没有必要担心，因为 RSVP 会话会自动地在一个称为 *cleanup timeout interval* 的计时器到期后超时。

RSVP 使用一些消息类型来建立、维护和拆除 RSVP 会话。在有问题的情况下，这些消息可以用于对 RSVP 会话进行故障排查。表 5-3 详细地描述了这些消息。

表 5-3

RSVP 消息类型

消息类型	消息细节	消息描述
PATH	建立 RSVP 会话的必要消息	PATH 消息存储关于 RSVP 路径状态的信息，包括前一跳的 IP 地址，这个信息被接收主机作为反向路径来到达发送者 这个 PATH 消息含有下列字段： SESSION——描述接收者的目的 IP 地址（单播或者组播）、协议类型和端口号码 RSVP_HOP——路径中每一个 RSVP 启用的设备的 IP 地址和逻辑出口 (LIH)，指定路径中每一个前一跳 (PHOP) 和下一跳 (NHOP) TIME_VALUES——RSVP 会话的刷新周期 这个 PATH 消息也含有一个发送者描述，它用于描述发送主机的特性，使用了 SENDER_TEMPLATE 和 SENDER_TSPEC SENDER_TEMPLATE 含有发送工作站的 IP 地址，协议类型和端口号码。SENDER_TSPEC 定义了流所需的特性，例如源和目的的 IP 地址、协议类型和端口号码。PATH 消息也作为周期性的 hello 消息来保持 RSVP 会话活动。默认的 hello 间隔是 30s
PATH ERROR	当在 PATH 消息中有错误发现时，会发送一个可选的错误消息	当在 PATH 消息中有错误发现时，这个消息会作为错误通知发送给发送者
PATH TEARDOWN	一个可选的错误消息通知下一跳路由器这个 PATH 不再有效，应当被删除	当路径被立刻清除后，PATH TEARDOWN 消息会发送出去。它们可以被 RSVP 发送者或者 RSVP 启用的路由器发送。这个消息会沿着预留的路径发送给所有 RSVP 启用的路由器，并被转发到所有的 RSVP 接收者

续表

消息类型	消息细节	消息描述
RESV	RSVP 接收者发送必需的消息来共享流标准	RESV 消息用于传输 RSVP 预留的请求，并且在 RSVP 启用的路由器之间请求数据。RESV 含有下列数据： SESSION、RSVP_HOP、TIME_VALUES 这些字段在前面的 PATH 消息中提到 RESV_CONFIRM——含有请求 RSVP 会话确认的接收者的 IP 地址 SCOPE——含有这个消息所适用的发送者的显式列表 STYLE——消息的预留风格 Flow Descriptor List——含有流描述的列表 Flow Descriptor——这个消息的流描述，包括 flowspec、filter spec 和预留风格（EF、WF、SE）
RSV ERROR	当 RESV 消息错误被发现时，发送一个可选的错误消息	当错误在 RESV 消息中发现时，这个消息作为错误通知发送给接收者
RESV CONFIRM	发送者发送一个可选的消息给接收者作为通知，这个消息适用于端对端	这个消息发送给接收者，通知接收者一个端对端的 RSVP 会话
RESV TEARDOWN	RSVP 接收者和中间的 RSVP 启用的路由器发送一个可选的消息来指明 RSVP 的资源应当被删除	当一条路径立刻被删除时，TEARDOWN 消息会发出。这些消息可以被 RSVP 接收者或 RSVP 启用的路由器发送，并且应当向上转发到所有的 RSVP 启用的路由器和 RSVP 的发送者

RSVP 可以使用两种类型的预留：受控的负荷服务和确保比特速率服务。*受控的负荷服务*允许一个 RSVP 的会话通过网络流动，而被其他流量流中断的可能性最小，从某种程度上说就像仿真电路一样。*确保比特速率服务*试图确保一个流通过网络时最坏情况的延迟。确保比特速率服务计算 RSVP 路径中 PATH 消息的延迟，并且在资源预留中将这个信息提供给接收者。我们已经看到了 RSVP 如何使用不同的消息类型来建立、维护和拆除 RSVP 预留，现在来看看 RSVP 是如何在思科的路由器上配置的。

一、RSVP 配置

RSVP 的配置需要两步。首先，RSVP 路径中所有路由器的接口必须配置使用加权公平队列（WFQ）。加权公平队列需要基于每个接口给 RSVP 提供流支持和队列。RSVP 带宽必须在每一个接口上预留。默认情况下，RSVP 可以预留一个接口 75% 的带宽。

注意：低延迟队列（LLQ）也可以用于提供 RSVP 支持。LLQ 在第 6 章中讨论。

第 1 步 在 RSVP 路径中启用加权公平队列。对每一个参与 RSVP 预留进程中的路由器的接口，必须启用加权公平队列。加权公平队列在小于 E1 速率的接口上启用。为了启用加权公平队列，在每一个接口上使用 **fair-queue** 命令。

```
fair-queue [discard-threshold][dynamic-queues][reservable-queues]
```

除了 *reservable-queues* 的值，默认的加权公平队列设置通常对任何低带宽的接口都适用。加权公平队列 *discard-threshold* 是一个从 1~4096 范围内的值。这个值指定在一个拥塞的接口上最多缓存多少个数据包而最终导致新的数据包被丢掉，默认的值是 64。*dynamic-queues* 参数允许用户指定在一个拥塞的接口上动态流的数量。*dynamic-queues* 参数的范围是 16~4096 个队列。默认情况下，加权公平队列使用 256 个动态队列。*reservable-queues* 参数允许用户配置一定数量的加权公平队列所支持的 RSVP 预留队列，可以配置的队列范围在 0~1000

之间。默认情况下，加权公平队列并不支持预留队列。加权公平队列会在下章中详细介绍。

第 2 步 使用 **ip rsvp bandwidth** 命令来配置 RSVP 的带宽预留限制。表 5-4 显示了可选的 RSVP 带宽参数和它们的描述。

```
ip rsvp bandwidth [reservable-bandwidth][largest-flow]
```

表 5-4 RSVP 带宽参数

命令	描述
<i>reservable-bandwidth</i>	(可选) 这个命令参数允许用户配置一个接口上以 kbit/s 表示的带宽总量。默认的设置是以 kbit/s 表示的可用带宽的 75%
<i>largest-flow</i>	(可选) 这个命令参数允许用户指定以 kbit/s 表示的最大流的尺寸。默认情况下，最大流限制为每秒可用带宽的 75%

除了 RSVP 带宽分配的配置之外，可以使用一些其他可选的 RSVP 命令来定制 RSVP 的性能和安全。为了使用静态的邻居分配语句配置 RSVP，使用 **ip rsvp neighbor** 命令，并且指定一个标准或者扩展的访问控制列表（列表号码为 1~199）。

```
ip rsvp neighbor {access-list}
```

如果在 RSVP 中使用 NetFlow 交换，**ip rsvp flow-assist** 命令允许 RSVP 使用 NetFlow 交换来支持 RSVP。

```
ip rsvp flow-assist
```

也可以通过使用 **ip rsvp precedence** 或者 **ip rsvp signalling** 命令来配置 RSVP，修改 IP 的优先级或者数据包的 DSCP 值。使用 **ip rsvp precedence** 命令，可以将 IP 的优先级修改为 0~7 的值。使用 **ip rsvp signalling dscp** 命令，可以将 DSCP 的值修改为 0~63 的值。也可以使用 **ip rsvp tos** 命令来修改类型服务（ToS）的值为 0~31。对于每一个命令，要么可以改变数据包遵循流的尺寸，要么数据包超过流的尺寸，或者两者都可以。IP 优先级、ToS 和 DSCP 数据包的标记在本章的后面讨论。

```
ip rsvp precedence [conform | exceed] precedence-value [conform | exceed]
precedence-value
ip rsvp signalling dscp dscp-value
ip rsvp tos [conform | exceed] tos-value [conform | exceed] tos-value
```

二、仿真 RSVP 消息

在实验的环境下，可以使用 **ip rsvp sender-host** 和 **ip rsvp reservation-host** 命令仿真静态的 RSVP 发送者和接收者。**ip rsvp reservation-host** 命令仿真一个 RSVP RESV 消息，而 **ip rsvp sender-host** 命令仿真一个 RSVP PATH 消息。表 5-5 显示了 RSVP 发送者和预留的命令参数以及它们的描述。

```
ip rsvp reservation-host destination-address source-address [IP-protocol-number
| tcp | udp] destination-port source-port next-hop-address interface-name
interface-number [ff | se | wf] [load | rate] average-bit-rate maximum-burst
ip rsvp sender-host destination-address source-address [IP-protocol-number | tcp
| udp] destination-port source-port next-hop-address interface-name interface-
number [ff | se | wf] [load | rate] average-bit-rate maximum-burst
```

注意: **ip-rsvp reservation-host** 和 **ip rsvp sender-host** 命令在不同的路由器平台上有不同

的可用选项。也可以发出这些命令的测试来发现哪些选项对你的路由器的平台可用，以找到你可以使用的选项。

表 5-5 rsvp simulation 命令参数	
命令参数	参数描述
destination-address	RSVP 会话的目的 IP 地址
source-address	RSVP 会话的源 IP 地址
[IP-protocol-number tcp udp]	与 RSVP 流相关的端口号码。这个端口号码可以是 TCP、UDP 或者是一个 0~255 的特定的 IP 协议号码
destination-port	目的端口号码，范围为 0~65 535。对于未指定的端口号码，使用 0 作为源和目的端口号码
source-port	源端口号码，范围为 0~65 535。对于未指定的端口号码，使用 0 作为源和目的端口号码
[ff se wf]	(仅 ip rsvp reservation-host 命令) 指定预留风格： FF (固定过滤器风格) —— 多个流共享一个预留 SE (共享显式风格) —— 每一个流一个预留 WF (通配符过滤器风格) —— 对于多个流的组播应用支持
[load rate]	带宽预留类型，负荷或者速率。负荷参数用于指定可控的负荷服务，而速率参数指定确保的位速率
average-bit-rate	保留的平均位速率，以 kbit/s 表示。这个值可以为 0~10 000 000
maximum-burst	这个最大突发值以千字节表示。这个值的范围为 0~65 535

范例 5-1 显示了图 5-2 中的两台主机(即发送者和接收者)是如何通过 ip rsvp sender-host 和 reservation-host 命令建立的。为了仿真 RSVP 的发送者或者接收者，必须在接口上启用 RSVP。RSVP 发送者/接收者的源地址必须在路由器上本地存在。

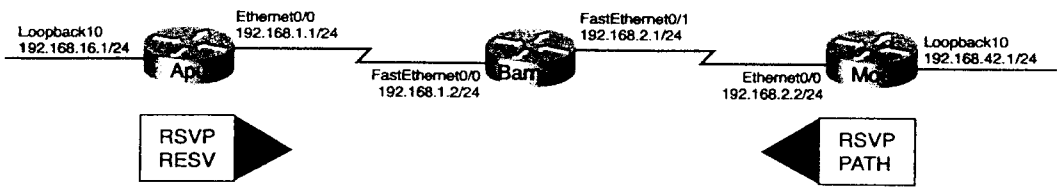


图 5-2 仿真 RSVP RESV 和 PATH 消息

范例 5-1 在 Apu 路由器上的 RSVP 仿真

```
Apu# show run | begin Loopback
interface Loopback10
 ip address 192.168.16.1 255.255.255.0
 ip rsvp bandwidth 7000 7000
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip rsvp bandwidth 7000 7000
!
router eigrp 170
 network 192.168.1.0
```

(待续)

```

network 192.168.16.0
no auto-summary
eigrp log-neighbor-changes
!
ip rsvp reservation-host 192.168.16.1 192.168.42.1 TCP 0 0 FF RATE 128 8

```

在这个范例中，Apu 路由器正在和 Moe 路由器仿真 RSVP 会话。**ip rsvp reservation-host** 命令用于仿真从 Apu 路由器的 Loopback10 接口到 Moe 路由器的 Loopback10 接口的 RSVP RESV 消息。范例 5-2 显示了来自 Apu 路由器的详细的 RSVP 预留信息。

范例 5-2 来自 Apu 路由器的 show ip rsvp reservation 信息

```

Apu# show ip rsvp reservation detail
RSVP Reservation. Destination is 192.168.16.1, Source is 192.168.42.1,
  Protocol is TCP, Destination port is 0, Source port is 0
  Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
  Average Bitrate is 128K bits/sec, Maximum Burst is 8K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 65535 bytes
  Resv ID handle: 00001301.
  Policy: Forwarding. Policy source(s): Default

```

正如你所看到的，Apu 路由器建立了来自 Moe 路由器的 192.168.42.1 的 IP 地址的预留。这个预留将适用于任何 IP 流量。Apu 路由器预留了一个确保的 128kbit/s 的平均速率，允许一个最大的突发速率 8 kilobytes，因此，在突发时可以支持的最大数据容量是 192 kbit/s。为了继续验证这个 RSVP 会话是端对端的，可以在 Barney 路由器上使用 **show ip rsvp senders** 命令，来验证它已经接收了来自 Moe 路由器的 RSVP PATH 消息，如范例 5-3 所示。

注意：用于找到 RSVP 突发速率的公式如下：

```

R = 位速 (kbit/s)
T = 时间间隔 (总是 1s)
B = 突发 (从 kilobytes 转换成 kilobits)
BR = 突发速率
R(T) + B = BR
所以，如果你使用先前范例中的信息，就会找到突发速率。
128kbit/s (1s) + 64kbit = 192 kilobit

```

为了将 kilobytes 转换成 kilobits，使用下面的公式（B 等同于以 kilobytes 表示的突发值）：

$B * 8 =$ 以 kilobits 表示的突发速率

范例 5-3 使用 show ip rsvp sender 命令来验证 Barney 路由器上的端对端预留

```

Barney# show ip rsvp sender

```

To	From	Pro	DPort	Sport	Prev Hop	I/F	BPS	Bytes
192.168.16.1	192.168.42.1	TCP	0	0	192.168.2.2	Fa0/1	128K	8K

正如你所看到的，Barney 路由器收到了来自 Moe 路由器的 RSVP PATH 消息，显示了前面的一跳，也就是 Moe 路由器的 Ethernet0/0 接口，以及目的主机的 IP 地址 192.168.16.1，也就是 Apu 路由器的 Loopback10 接口。作为一个最终的端对端的验证，范例 5-4 显示了在 Moe 路由器上的配置和 **show ip rsvp sender detail** 信息。

范例 5-4 在 Moe 路由器上的 RSVP 仿真

```
Moe# show run | begin Loopback10
interface Loopback10
 ip address 192.168.42.1 255.255.255.0
 ip rsvp bandwidth 7000 7000
!
interface Ethernet0/0
 ip address 192.168.2.2 255.255.255.0
 ip rsvp bandwidth 7000 7000
!
router eigrp 170
 network 192.168.2.0
 network 192.168.42.0
 no auto-summary
 no eigrp log-neighbor-changes
 ip rsvp sender-host 192.168.16.1 192.168.42.1 tcp 0 0 128 8
Moe# show ip rsvp sender detail
PATH Session address: 192.168.16.1, port: 0. Protocol: TCP
  Sender address: 192.168.42.1, port: 0
  Traffic params - Rate: 128K bits/sec, Max. burst: 8K bytes
                   Min Policed Unit: 0 bytes, Max Pkt Size 65535 bytes
  Path ID handle: 00000601.
  Incoming policy: Accepted. Policy source(s): Default
  Output on Ethernet0/0. Policy status: Forwarding. Handle: 00000601
```

在先前的范例中，一个 RSVP 发送者，即主机 192.168.42.1，loopback10 接口的 IP 地址，被设置为在 TCP 端口 0 上建立一个到主机 192.168.16.1 的 RSVP 预留。**ip rsvp sender-host 192.168.16.1 192.168.42.1 tcp 0 0 128 64** 命令用于建立来自 Moe 路由器的 RSVP PATH 仿真消息，这个命令并不会出现在运行的配置文件中。

三、建立静态的 RSVP 预留

RSVP 预留也可以通过使用 **ip rsvp reservation** 和 **ip rsvp sender** 命令来静态地配置。**ip rsvp reservation** 命令，在后面显示，对于一个 RSVP 接收者建立一个静态的预留，而 **ip rsvp sender** 命令对于一个 RSVP 发送者建立一个静态的预留。这两个命令允许在两端配置 RSVP 会话预留：

```
ip rsvp reservation destination-address source-address [IP-protocol-number { tcp
| udp } destination-port source-port next-hop-address interface-name interface-
number [ff | se | wf] [load | rate] average-bit-rate maximum-burst
ip rsvp sender destination-address source-address [IP-protocol-number { tcp |
udp } destination-port source-port previous-hop-address interface-name
interface-number average-bit-rate maximum-burst
```

ip rsvp reservation 和 **ip rsvp sender** 命令有几个必需的参数。表 5-6 列出了这些 **rsvp** 命令的参数和它们的描述。

表 5-6 static rsvp 命令和描述

RSVP 命令参数	描述
<i>destination-address</i>	RSVP 接收者的 IP 地址或主机名
<i>source-address</i>	RSVP 发送者的 IP 地址或主机名
[<i>IP-protocol-number</i> {tcp udp}]	<i>IP-protocol-number</i> 是一个 0~255 的 IP 协议，或者是 TCP 或 UDP 协议
<i>destination-port</i>	目的端口号，范围为 0~65 535

续表

RSVP 命令参数	描述
<i>source-port</i>	源端口号，范围为 0~65 535
<i>next-hop-address</i> 或 <i>previous-hop-address</i>	ip rsvp reservation 命令需要下一跳的 IP 地址或者主机名 ip rsvp sender 命令需要前一跳的地址或者主机名
[ff se wf]	FF——固定过滤器预留对每一个流提供一个单独的预留 SE——共享显式预留对特定的流提供预留 WF——通配符过滤器预留对所有的流提供共享预留
[load rate]	Load——代表可控负荷服务。通过指定平均位速率和最大的突发速率来将一个流和其他流之间的干扰降为最低 Rate——代表确保位速率。通过指定平均位速率和最大的突发速率来提供一个确保的位速率
<i>average-bit-rate</i>	这个值范围为 1~10 000 000，指定了以 kbit/s 表示的平均位速率
<i>maximum-burst</i>	这个值范围为 1~65 535，指定了以 kilobits 表示的最大的突发数据量

范例 5-5 显示了如何使用 **ip rsvp reservation** 命令建立从发送者 152.148.89.91 到接收者 10.1.1.11 的 TFTP 流量的预留，它使用 FF 预留，使用 64 kbit/s 的平均位速率和 4 Kb 的最大突发来将流量发送到下一跳 10.2.2.2。show ip rsvp host receivers 命令的输出如下所示。

范例 5-5 使用静态预留

```
RSVP-Example# show run | begin Serial0
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip rsvp bandwidth 75000 75000
!
interface Serial0/0
 ip address 10.2.2.1 255.255.255.0
 ip rsvp bandwidth 1158 1158
!
ip rsvp reservation 10.1.1.11 152.148.89.91 UDP 69 0 10.2.2.2 Serial0/0 FF
RATE 64 4
RSVP-Example# show ip rsvp host receivers
To          From          Pro DPort Sport Next Hop      I/F   Fi Serv BPS Bytes
10.1.1.11   152.148.89.91 UDP 69      0    10.2.2.2   Se0/0 FF RATE 64K 4K
```

在先前的范例中，一个静态的 RSVP 会话建立在主机 10.1.1.11 和 152.148.89.91 之间。主机 152.148.89.91 将使用一个 RSVP PATH 消息请求 RSVP 会话，而主机 10.1.1.11 将使用 RSVP RESV 消息来响应这个信息。RSVP 会话为任何通过主机 152.148.89.91 发送的 TFTP 流量预留。

现在你已经看到 RSVP 如何用于仿真 RSVP 会话、动态的 RSVP 会话或者静态的 RSVP 预留。现在来看看可以使用 RSVP 实现多业务的语音应用的其他方法。

四、对语音保留适量的带宽

当配置在 IP 上传输语音并且与 RSVP 一起使用时，记住非常重要是有许多不同的语音编码方法可以选择，每一种编码方法对网络都有不同的服务质量需求。用于采样和编码数据包的编码方法将影响性能和呼叫的质量。复杂的采样数据的编码方法通常会有更长的数据包传输延迟，然而，它们通常需要较低的传输带宽，只是因为它们压缩数据并且发送较少的数据包。为了选择一种语音编码，在远端 voice over IP dial-peer 上使用 **codec codec-name** 命令

选择一种编码速率，默认的编码方法是 g729r8。

注意：数据包串行化延迟是指采集原始的语音样本并且将那个采样数据编码成数据包进行传输所需的时间。

表 5-7 显示了在思科 1750 系列路由器上不同的语音编码速率、编码的名字、以位表示的每秒钟的编码速率、数据包的串行化延迟和编码所请求的实际的 RSVP 速率。确保你所选择的编码方法将从网络得到合理的带宽。如果你不能通过配置 RSVP 得到足够的带宽来建立预留，并且你通过配置 dial-peer 来请求或者接受 RSVP 的设置，呼叫将不能被接受。

表 5-7 Voice Over IP 编解码方法

编解码速率	编解码名字	以 bit/s 表示的编解码速率	打包延迟	实际请求的 RSVP 速率
g711alaw	G.711 A Law	64 000bit/s	10~30ms	80k
g711ulaw	G.711 u Law	64 000bit/s	10~30ms	80k
g723ar53	G.723.1 ANNEX-A	53 00bit/s	30ms	18k
g723ar63	G.723.1 ANNEX-A	63 00bit/s	30ms	18k
g723r53	G.723.1	53 00bit/s	30ms	18k
g723r63	G.723.1	63 00bit/s	30ms	18k
g726r16	G.726	16 000 bit/s	10~30ms	32k
g726r24	G.726	24 000 bit/s	10~30ms	40k
g726r32	G.726	32 000 bit/s	10~30ms	48k
g728	G.728	16 000 bit/s	10~30ms	32k
g729br8	G.729 ANNEX-B	8 000 bit/s	10~30ms	24k
g729r8	G.729	8 000 bit/s	10~30ms	24k

show dialer-peer voice | include codec 命令显示了当前的编码配置，可以使用这些信息来计算对范例 5-6 所示的语音流量的 RSVP 预留信息。**show dial-peer voice** 命令的完整版本显示了每一个 dial-peer 的详细信息。

范例 5-6 使用 Show 命令来发现 Codec

```
Show-me-the-codec# show dial-peer voice | include codec
codec = g729r8, payload size = 20 bytes,
```

五、对语音流量使用 RSVP

为了配置 Voice over IP (VoIP) 来请求 RSVP 服务，在 VoIP 会话的 dial peer 下使用 **req-qos** 命令。**req-qos** 命令是请求服务质量的省略写法，用于从网络请求某种程度的服务质量级别，可以请求三种不同类型的服务：**best-effort**、**controlled-load** 或者 **guaranteed-delay**。**acc-qos** 命令定义了通过网络可接受的最低程度的可接受服务类型。**controlled-load** 命令用于请求或者接受流量。**best-effort** 命令用于从 dial-peer 连接中清除已经存在的 RSVP 预留。表 5-8 汇总了 dial peer qos 命令参数并且对它们的使用给出了简短的描述。

表 5-8

rsvp voice qos 命令汇总

RSVP 语音命令	命令描述
best-effort	best-effort 服务就像它的名字所暗示的，是一种尽力传递的服务。在尽力传递路径中所有的设备试图以最大的可能性来传输数据包，但是没有任何特殊的方法来优化尽力传递的流量。这个命令用于从 dial peer 中清除 controlled-load 或者 guaranteed-delay 命令
controlled-load	对实时的延迟敏感的应用程序提供一种预留，可以提供有限的延迟和数据包的丢弃。类似于 ATM VBR-nPVC。受控的负荷预留可以提供带宽预留，它可以限制实时的网络应用程序在通过一个有负荷的网络时所经历的延迟和数据包的丢失率
guaranteed-delay	提供一种确保的速率，类似于 ATM CBR PVC，通过从 RSVP PATH 消息中收集数据

下面的 4 步需要配置来启用 VoIP 的预留请求，使用 **req-qos** 或者 **acc-qos** 命令：

第 1 步 使用 **dial-peer** 命令来配置本地和远端的 VoIP dial peer。

第 2 步 在 dial-peer 配置模式中，在 VoIP dial peer 中添加 **acc-qos** 或者 **req-qos** 命令。

第 3 步 转发语音流量的每一个接口都需要启用加权公平队列来启用 RSVP 的配置。在启用 RSVP 配置之前，使用 **fair-queue** 命令在接口的级别上启用加权公平队列。

第 4 步 对每一个接口配置 RSVP。在配置 RSVP 之前，确保你知道每一种语音的编码方法所需要的带宽和延迟。当你知道需要预留多少带宽之后，对每一个启用 RSVP 的接口，使用 **ip rsvp bandwidth bandwidth** 命令来配置 RSVP。

注意：如果你没有显式地配置语音编码，而你需要发现你所使用的编码方法，可以使用本章先前所提到的 **show voice | include codec** 命令来查看。

下面的范例显示了如何在 Bender 路由器上使用上述步骤来启用对于 VoIP 呼叫的 RSVP。图 5-3 显示了 Bender 路由器和它的 VoIP dial peer 即 Frye 路由器之间的语音连接。

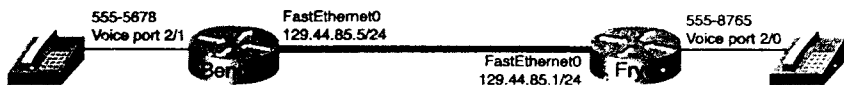


图 5-3 Bender 和 Frye 网络

第 1 步 使用 **dial-peer** 命令来配置本端和远端的 VoIP dial peer。下面的范例显示了对 Bender 路由器的 dial-peer 配置。在这个范例中，dial-peer 5555678 指定了本地的目的号码形式（即 5555678）和本端的 FXS 端口（port 2/1）。dial-peer 5558765 指定了位于 129.44.85.1 上的远端的对等体，并且给分配了 5558765 的目的号码。g726r16 语音编码用于这个 dial peer 的语音呼叫。

```
Bender(config)# dial-peer voice 5555678 pots
Bender (config-dial-peer)# destination-pattern 5555678
Bender (config-dial-peer)# port 2/1
Bender (config)# dial-peer voice 5558765 voip
Bender (config-dial-peer)# destination-pattern 5558765
Bender (config-dial-peer)# session target ipv4:129.44.85.1
Bender (config-dial-peer)# codec g726r16
```

第 2 步 在 dial-peer 的配置模式中，在 VoIP dial peer 中添加 **acc-qos** 或者 **req-qos** 命令。在这个范例中，Bender 路由器配置了从网络中请求和接受受控的负荷服务。

```
Bender(config-dial-peer)# dial-peer voice 5558765 voip
Bender(config-dial-peer)# req-qos controlled-load
```



```
Bender(config-dial-peer)# acc-qos controlled-load
```

第3步 每一个转发语音流量的接口在配置 RSVP 之前都将配置加权公平队列。因此，在启用 RSVP 之前，使用 **fair-queue** 命令启用接口上的加权公平队列。发现当前正在接口上使用的队列机制的快速方法是使用 **show queueing interface interface-name interface-number | include strategy** 命令。你会看到如果使用的是先进先出队列，队列的策略将是空。在这种情况下，在配置 RSVP 之前，应当启用加权公平队列。

```
Bender# show queueing interface fastEthernet 0 | include strategy
Interface FastEthernet0 queueing strategy: none
```

在下面的范例中，Bender 路由器使用它的 FastEthernet0 接口和 Frye 路由器连接，在这里加权公平队列必须启用：

```
Bender(config)# interface FastEthernet0
Bender(config-if)# ip address 129.44.85.5 255.255.255.0
Bender(config-if)# fair-queue
```

第4步 对每一个接口配置 RSVP，在配置 RSVP 之前，确保你知道语音编码需要多少带宽和延迟。因为 Bender 路由器已经配置使用了 g726r16 的编码，所以你知道 RSVP 至少需要 32 kbit/s 的带宽预留。为了确保 RSVP 的配置允许对这个带宽的预留，按照如下所示，使用 **ip rsvp bandwidth 32** 命令。

```
Bender(config)# interface FastEthernet0
Bender(config-if)# ip rsvp bandwidth 32
```

为了验证连接，从其中的一台路由器发起测试的呼叫，在呼叫的进行中，使用 **show ip rsvp installed** 和 **show ip rsvp reservation detail** 命令来显示当前的 RSVP 预留。**show ip rsvp installed** 命令显示了对当前的 RSVP 会话的快速汇总。**show ip rsvp reservation detail** 命令显示了每一种 RSVP 预留的所有特性，如范例 5-7 所示。

范例 5-7 使用 show ip rsvp reservation detail 命令来验证 VoIP

```
Bender# show ip rsvp installed
RSVP: FastEthernet0
BPS      To          From          Protoc DPort  Sport  Weight Conversation
32K      129.44.85.1      129.44.85.5   UDP     17176  18930  0           264
Bender# show ip rsvp reservation detail
RSVP Reservation. Destination is 129.44.85.1, Source is 129.44.85.5,
Protocol is UDP, Destination port is 17176, Source port is 18930
Next Hop is 129.44.85.1, Interface is FastEthernet0
Reservation Style is Fixed-Filter, QoS Service is Controlled-Load
Average Bitrate is 32K bits/sec, Maximum Burst is 160 bytes
Min Policed Unit: 80 bytes, Max Pkt Size: 80 bytes
Resv ID handle: 00000E01.
Policy: Forwarding. Policy source(s): Default
RSVP Reservation. Destination is 129.44.85.5, Source is 129.44.85.1,
Protocol is UDP, Destination port is 18930, Source port is 17176
Reservation Style is Fixed-Filter, QoS Service is Controlled-Load
Average Bitrate is 32K bits/sec, Maximum Burst is 160 bytes
Min Policed Unit: 80 bytes, Max Pkt Size: 80 bytes
Resv ID handle: 00000C01.
Policy: Forwarding. Policy source(s): Default
```

在这个范例中，每次在 555-5678 和 555-8765 两部电话之间进行呼叫时，会产生两个 RSVP 的预留。一个是从 129.44.85.1 到 129.44.85.5，另外一个是从 129.44.85.5 到 129.44.85.1。每一个预留对每一路电话使用受控的负荷服务来提供 32 kbit/s 的平均比特速率。一旦呼叫结束，预留会被清除，带宽会被释放用于其他目的。

范例 5-8 显示了在先前的范例中 Bender 和 Frye 路由器之间的 VoIP 会话的 RSVP 受控速率服务的完整配置。

范例 5-8 使用 VoIP 和 RSVP

```
Bender# show run | begin FastEthernet
interface FastEthernet0
 ip address 129.44.85.5 255.255.255.0
 fair-queue 64 256 1
 ip rsvp bandwidth 32 32
!
interface Serial1
 dial-peer voice 5555678 pots
 destination-pattern 5555678
 port 2/1
!
 dial-peer voice 5558765 voip
 destination-pattern 5558765
 session target ipv4:129.44.85.1
 req-qos controlled-load
 acc-qos controlled-load
 codec g726r16
```

```
Frye# show run | begin FastEthernet
interface FastEthernet0
 ip address 129.44.85.1 255.255.255.0
 fair-queue 64 256 1
 ip rsvp bandwidth 32 32
!
 dial-peer voice 5558765 pots
 destination-pattern 5558765
 port 2/0
!
 dial-peer voice 5555678 voip
 destination-pattern 5555678
 session target ipv4:129.44.85.5
 req-qos controlled-load
 acc-qos controlled-load
 codec g726r16
```

注意：如果语音 RSVP 服务质量参数只在连接的一端配置，呼叫将永远不会成功完成。为了成功地允许语音呼叫使用 RSVP，连接的一端必须请求服务的级别，另外一端必须愿意接受那个服务级别。

六、RSVP 故障排查

可以使用一些命令进行 RSVP 的故障排查。然而，在配置 RSVP 的故障排查之前，还需要检查一些事情。首先，验证加权公平队列已经在 RSVP 的接口上启用了。如果还没有，使用 **fair-queue** 命令启用它。其次，当在不连续的电路上启用 RSVP 时，例如在 Frame Relay DS0 上时，记住要配置接口的带宽，这是因为串行接口默认的带宽是 1158 kbit/s，或者是默认的

接口带宽的 75%。

show ip rsvp neighbor 命令显示连接了 RSVP 邻居的接口并且显示关于邻居的 IP 地址，如范例 5-9 所示。

范例 5-9 显示 RSVP 邻居

```
Silly# show ip rsvp neighbor
Interface Neighbor      Encapsulation
Se1.1      192.168.1.2      RSVP
Se1.2      192.168.2.2      RSVP
```

show ip rsvp sender 和 **show ip rsvp request** 命令提供了关于 RSVP 发送者和 RSVP 请求的汇总信息。为了查看 RSVP 请求的详细信息，在 **show ip rsvp request** 命令中使用 **detail** 参数。这些命令的范例如范例 5-10 所示。

范例 5-10 RSVP show rsvp sender 和 request 命令

```
Smiley# show ip rsvp sender
To          From          Pro DPort Sport Prev Hop      I/F BPS Bytes
192.168.1.2 192.168.2.2      UDP 18182 18050 192.168.2.2 Se1.2 24K 1K
192.168.2.2 192.168.1.2      UDP 18050 18182 192.168.1.2 Se1.1 24K 1K
Smiley# show ip rsvp request
To          From          Pro DPort Sport Next Hop      I/F Fi Serv BPS Bytes
192.168.1.2 192.168.2.2      UDP 18182 18050 192.168.2.2 Se1.2 FF RATE 24K 1K
192.168.2.2 192.168.1.2      UDP 18050 18182 192.168.1.2 Se1.1 FF RATE 24K 1K
Grumpy# show ip rsvp request detail
RSVP Reservation. Destination is 192.168.2.2, Source is 192.168.1.2,
  Protocol is UDP, Destination port is 18634, Source port is 18540
Next Hop is 192.168.2.1, Interface is Serial0
Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
Average Bitrate is 24K bits/sec, Maximum Burst is 1K bytes
```

如范例 5-11 中的 **show ip rsvp installed** 命令所示，它给出了关于当前的 RSVP 预留的信息，例如 RSVP 接口、预留的以 bit/s 表示的大小、源和目的 IP 地址、协议、源和目的端口号、RSVP 流的权重以及会话的数量。**show ip rsvp interfaces** 命令，如范例 5-11 所示，显示了关于路由器的每一个 RSVP 接口的信息。

范例 5-11 show ip rsvp installed 命令

```
Grumpy# show ip rsvp installed
RSVP: Serial1
BPS To          From          Protoc DPort Sport Weight Conversation
RSVP: Serial1.1
BPS To          From          Protoc DPort Sport Weight Conversation
24K 192.168.1.2 192.168.2.2      UDP 18182 18050 6 265
RSVP: Serial1.2
BPS To          From          Protoc DPort Sport Weight Conversation
24K 192.168.2.2 192.168.1.2      UDP 18050 18182 6 266
Grumpy# show ip rsvp interfaces
interface allocated i/f max flow max pct UDP IP UDP_IP UDP M/C
Se1      48K      1158K 1158K 4 0 0 0 0
Se1.1    24K      128K 128K 18 0 1 0 0
Se1.2    24K      128K 128K 18 0 1 0 0
```

为了查看关于当前的所有 RSVP 预留的信息，使用 `show ip rsvp reservation` 命令。这个命令显示了每一个预留的源和目的 IP 地址、协议号码、源和目的端口号码、下一跳的 IP 地址及用于到达每一个发送者的接口、预留的过滤类型（FF、SE 或者 WF）、预留类型（RATE 或者 LOAD）、以 bit/s 表示的预留大小和以字节表示的突发大小，如范例 5-12 所示。

范例 5-12 show ip rsvp reservation 命令

Grumpy# show ip rsvp reservation										
To	From	Pro	DPort	Sport	Next Hop	I/F	Fi	Serv	BPS	Bytes
192.168.1.2	192.168.2.2	UDP	18182	18050	192.168.1.2	Se1.1	FF	RATE	24K	1K
192.168.2.2	192.168.1.2	UDP	18050	18182	192.168.2.2	Se1.2	FF	RATE	24K	1K
192.168.2.3	192.168.1.2	UDP	18502	16808	192.168.2.2		FF	LOAD	24K	1K

5.2 范例：RSVP 和 VoIP

VoIP 需要一定级别的服务质量来正常工作。当在广域网环境下带宽很小且拥塞的链路上使用 VoIP 时，在许多情况下，需要实施某种程度的服务质量。幸运的是，VoIP 有内置的 RSVP 的支持，使得它配置起来很简单。在下面的实验中，你可以练习 VoIP 的配置并且使用 RSVP 来支持语音。

5.2.1 实验练习

Dan's Pizza 在世界各地有四千多个站点。每一个商店都通过帧中继的连接和它的中心站点连接。这个中心站点对每一个区域提供对总部网络的访问，支持所有的网络应用程序，对内部呼叫提供一路电话线。在过去几个月中，已经实施了几个新的应用程序，这使得语音流量变得有些抖动并且会话难以听懂。将配置 RSVP 来预留足够的带宽从而平稳语音的呼叫。在这个实验中，将配置子网 area 140 的部分配置，来提供语音流量的服务质量。

5.2.2 实验目的

本实验的目的是使用 RSVP 来对两个商店之间的语音流量预留带宽。对于这个网络模型，使用图 5-4 所示的 Dan's Pizza 的网络部分。这个练习演示了 RSVP 是如何配置的，使用 VoIP 作为测试的应用程序。RSVP 配置是通过使用 RSVP `show` 和 `debug` 命令来验证的。

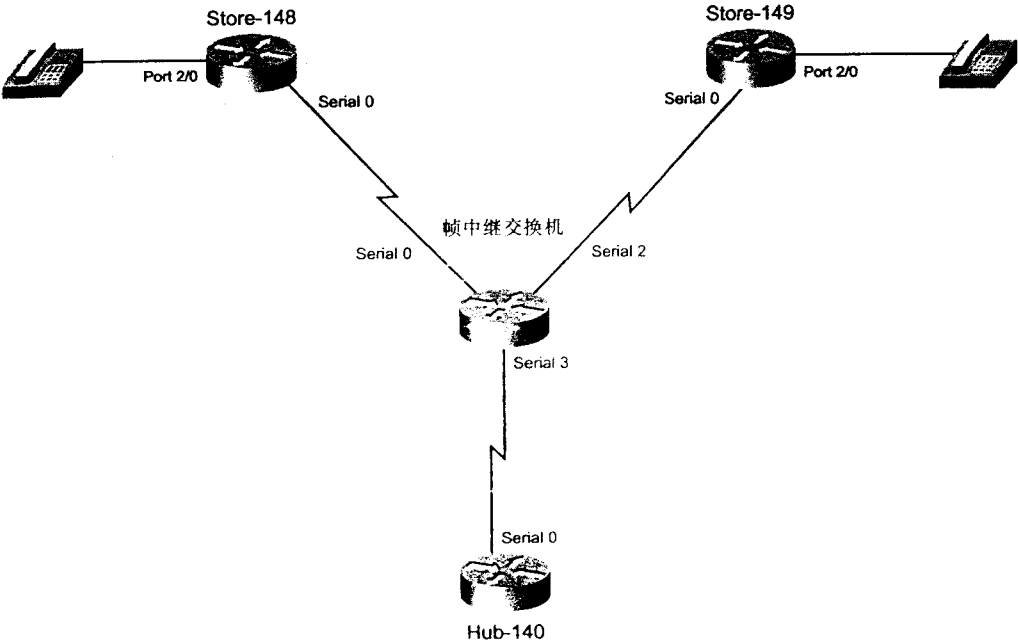


图 5-4 Dan's Pizza 子网 140

5.2.3 需要的设备

为了使用 RSVP，并且对两个商店之间的语音流量预留带宽，需要下面的设备：

- 两台思科路由器，具有至少一个语音端口和一个串行端口；
- 一台思科路由器，具有一个串行端口；
- 一台思科路由器，具有 3 个串行端口来充当帧中继交换机；
- 两部电话用于测试的目的。

5.2.4 物理布局和预规划

为了完成物理布局和预规划，执行下面的这些任务：

- 按照图 5-4 所示对路由器进行布线。
- 在每一个具有语音的路由器的语音端口上连接一部电话。
- 按照表 5-9 所示使用 PVC 的信息来配置帧中继交换机。
- 验证所有的接口都是 up 的状态。

表 5-9 范例中帧中继交换机的配置

本地接口	本地 DLCI	远端接口	远端 DLCI
Serial0	148	Serial3	841
Serial2	149	Serial3	941
Serial3	841	Serial0	148
Serial3	941	Serial2	149

一、实验任务

为了完成这个实验的练习，需要完成下面的这些任务：

- 按照图 5-5 所示配置 IP 网络，在 Hub 140 路由器上配置 IP 地址并使用子接口配置帧中继，在 Store 路由器上配置物理接口。
- 在所有的路由器上配置 OSPF，所有的串行接口都属于 OSPF 区域 0，验证所有的路由器都是可达的。
- 配置 Store 148 和 Store 149 彼此能够使用图 5-5 所示的电话号码来呼叫对方。测试电话连接。
- 配置所需的 RSVP 支持来允许 Store 148 呼叫 Store 149，反之亦然，采用 RSVP 确保的位速率服务。每一股流应当接收至少 24kbit/s 的平均位速率，应当允许突发到整个接口带宽的 75%。测试并且验证 RSVP 会话正常工作。

为了完成这个任务，执行随后的步骤。

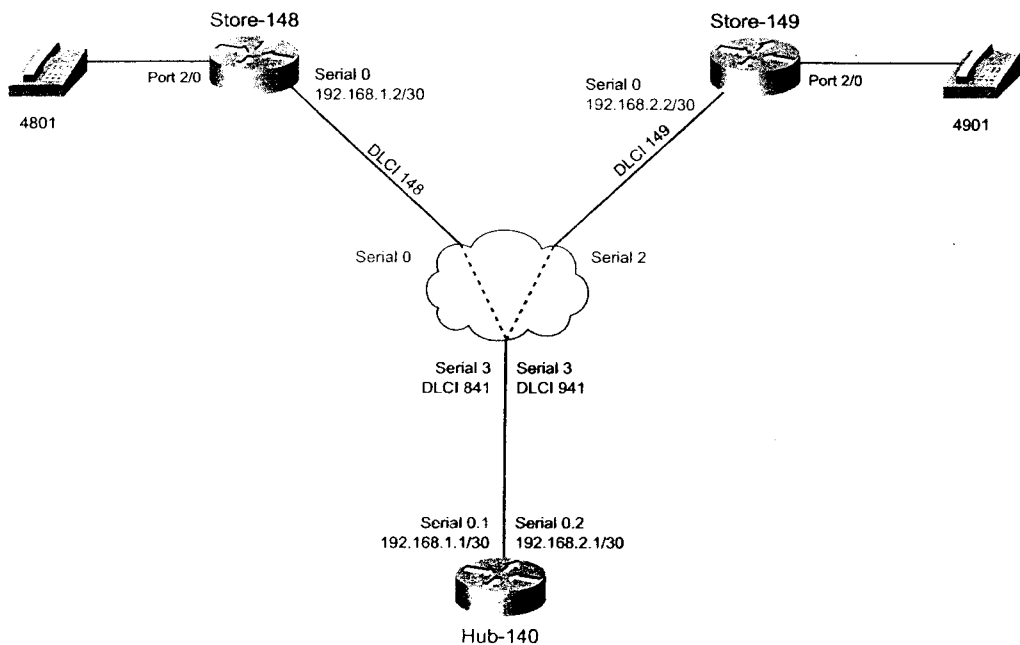


图 5-5 本实验的物理布局

二、实验步骤

第 1 步 将所有的路由器连接到帧中继的交换机上。Hub 140、Store 148 和 Store 149 都应当使用它们的串行接口连接到帧中继的交换机上，注意每一个串行连接并且使用这个信息来配置帧中继的交换机。Store 148 在它的串行接口上使用的是 DLCI 148，Store 149 应当配置使用 DLCI 149 的号码，而 Hub 140 应当分配 DLCI 841 和 DLCI 941。配置每一台路由器都支持帧中继的连接。因为 Store 148 和 Store 149 使用物理接口作为它们的帧中继的连接，它们应当被配置使用帧中继

的映射来指向 Hub 140。Hub 140 使用多点子接口，所以它也可以在每一个子接口上使用帧中继映射的命令。范例 5-13 显示了帧中继交换机的配置和帧中继的路由。

范例 5-13 帧中继交换机的配置

```
frame-relay-switch# show run | begin frame
frame-relay switching
!
interface Serial0
  no ip address
  encapsulation frame-relay IETF
  clockrate 1300000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 148 interface Serial3 841
!
interface Serial2
  no ip address
  encapsulation frame-relay IETF
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 149 interface Serial3 941
!
interface Serial3
  no ip address
  encapsulation frame-relay IETF
  clockrate 1300000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 841 interface Serial0 148
  frame-relay route 941 interface Serial2 149
frame-relay-switch# show frame-relay route
Input Intf      Input Dlci      Output Intf      Output Dlci      Status
Serial0         148             Serial3          841              active
Serial2         149             Serial3          941              active
Serial3         841             Serial0          148              active
Serial3         941             Serial2          149              active
```

第 2 步 当验证路由器之间帧中继连接的配置后，应当分配 IP 地址。Store 148 的接口 Serial 0 应当使用 IP 地址 192.168.1.2/30。Store 149 的串口应当使用 IP 地址 192.168.2.2/30，Hub 140 的接口 Serial 0.1 应当使用 192.168.1.1/30，0.2 应当使用 192.168.2.1/30。在继续下一步之前，应当验证所有的 Store 路由器能够 ping 通它们直连的 Hub 的子接口。

第 3 步 在每一台路由器上配置 OSPF，使得每一台路由器能够 ping 通它的邻居，并且每一个 store 路由器能够 ping 通彼此。在这个范例中使用非广播的开放最短路径优先（OSPF）配置。

为了在 hub 路由器和两个 store 路由器之间配置非广播连接的 OSPF，必须对非广播网络配置 OSPF。在这种情况下，需要使用 **ip ospf network non-broadcast** 命令并且使用静态的邻居配置。范例 5-14 显示了 **ip ospf network** 命令是如何在 Hub 140 路由器上使用的。在进行第 4 步之前，从一个 store 路由器 ping 另外一台路由器的串口的 IP 地址来验证 OSPF 配置。

范例 5-14 Hub 140 路由器的 OSPF 配置

```
Hub-140#show run | begin Serial0/0.1
interface Serial0/0.1 multipoint
 ip address 192.168.1.1 255.255.255.252
 ip ospf network non-broadcast
 frame-relay map ip 192.168.1.2 841 broadcast
!
interface Serial0/0.2 multipoint
 ip address 192.168.2.1 255.255.255.252
 ip ospf network non-broadcast
 frame-relay map ip 192.168.2.2 941 broadcast
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.3 area 0
 network 192.168.2.0 0.0.0.3 area 0
 neighbor 192.168.2.2
 neighbor 192.168.1.2
```

第 4 步 当所有的路由器能够 ping 通彼此后，按照如下所示在 Store 148 路由器上配置 VoIP：建立一个 dial-peer（在这个范例中，我使用对方的号码 4801 使事情变得简单）。将 dial-peer 4801 分配为目的的拨号 4801。dial-peer 4801 必须分配到一个端口中（在这个情况下，我使用的是端口 2/0）。为了从 Store 149 路由器拨打 4901 扩展号，必须建立一个 VoIP dial-peer，指定 Store 149 的串行 IP 地址，并且给它分配目的号码 4901。Store 149 的配置和 Store 148 的配置应当是类似的。当每一台路由器的语音配置完成后，应当能够从 Store 149 的电话呼叫 4801 的号码，也能够从 Store 148 的电话呼叫 4901。范例 5-15 显示了对 Store 148 路由器的 VoIP 配置。

范例 5-15 Store-148 路由器的 Voice Over IP 配置

```
Store-148# sho run | begin dial-peer
dial-peer voice 4801 pots
 destination-pattern 4801
 port 2/0
!
dial-peer voice 4901 voip
 destination-pattern 4901
 session target ipv4:192.168.2.2
```

第 5 步 当完成语音连接的测试后，现在应该配置 RSVP 了。RSVP 配置的第一步是启用加权公平队列，如果还没有配置的话。加权公平队列是使用 **fair-queue** 命令来启用的。因为这是一个简单的加权公平队列的配置，可以只是输入 **fair-queue** 命令并且接受默认的配置。当配置完加权公平队列后，在接口上启用 RSVP，使用 **ip rsvp bandwidth** 命令。确保至少保留 24 kbit/s 的带宽，并且允许突发到接口带宽的 75%，也就是 1158 kbit/s。下一步，在 **ip rsvp bandwidth** 请求中使用 **req-qos** 命令启用语音，来请求一个确保的位速率。范例 5-16 显示了对 Store 149 路由器的 VoIP 和 RSVP 配置，范例 5-17 显示了 Store 148 路由器上使用 **show ip rsvp reservation detail** 命令的输出。

范例 5-16 对 Store 149 的 RSVP 和语音的配置

```
Store-149# show run | begin Serial
interface Serial0
 ip address 192.168.2.2 255.255.255.252
 ip ospf network non-broadcast
 ip ospf priority 0
 frame-relay map ip 192.168.2.1 149 broadcast
 ip rsvp bandwidth 1158 24
!
dial-peer voice 4901 pots
 destination-pattern 4901
 port 2/0
!
dial-peer voice 4801 voip
 destination-pattern 4801
 session target ipv4:192.168.1.2
 req-qos guaranteed-delay
```

范例 5-17 来自 Store 148 路由器的预留信息

```
Store-148# show ip rsvp reservation detail
RSVP Reservation. Destination is 192.168.1.2, Source is 192.168.2.2,
Protocol is UDP, Destination port is 17188, Source port is 19346
Next Hop is 192.168.2.1, Interface is Serial0
Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
Average Bitrate is 24K bits/sec, Maximum Burst is 120 bytes
Min Policed Unit: 60 bytes, Max Pkt Size: 60 bytes
Resv ID handle: 00007601.
Policy: Forwarding. Policy source(s): Default
RSVP Reservation. Destination is 192.168.2.2, Source is 192.168.1.2,
Protocol is UDP, Destination port is 19346, Source port is 17188
Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
Average Bitrate is 24K bits/sec, Maximum Burst is 120 bytes
Min Policed Unit: 60 bytes, Max Pkt Size: 60 bytes
Resv ID handle: 00007201.
Policy: Forwarding. Policy source(s): Default
```

正如你所看到的，在先前的范例中，Store 148 路由器预留了 24 kbit/s 的平均速率，而最大突发速率是 120 字节；这个 RSVP 预留是确保速率。

测试此配置的另外一种方法是通过使用 **debug ip rsvp detail** 命令来启用详细的 RSVP 调试信息，然后从 Store 148 路由器拨打 4901。当你拿起 4901 电话后，应当收到类似于范例 5-18 所示的输出。

范例 5-18 debug ip rsvp detail 输出

```
*Mar 1 05:28:57.294: RSVP 192.168.1.2_17598-192.168.2.2_18180: Static
reservation is new
Comment: New reservation requested
*Mar 1 05:28:57.294: RSVP-RESV: Locally created reservation. No admission/traffic
control needed
*Mar 1 05:28:57.298: RSVP session 192.168.1.2_17598: Sending PATH message for
192.168.1.2 on interface Serial0
Comment: RSVP PATH information from 192.168.1.2
Comment: Reservation information - IP addresses and port numbers
*Mar 1 05:28:57.298: RSVP: version:1 flags:0000 type:PATH cksum:31D8 ttl:255
```

(待续)

```

reserved:0 length:136
*Mar 1 05:28:57.298: SESSION type 1 length 12:
*Mar 1 05:28:57.298: Destination 192.168.1.2, Protocol_Id 17, Don't Police ,
DstPort 17598
Comment: RSVP Destination Information
*Mar 1 05:28:57.298: HOP type 1 length 12: C0A80202
*Mar 1 05:28:57.298: : 00000000
*Mar 1 05:28:57.302: TIME_VALUES type 1 length 8 : 00007530
*Mar 1 05:28:57.302: SENDER_TEMPLATE type 1 length 12:
*Mar 1 05:28:57.302: Source 192.168.2.2, udp_source_port 18180
Comment: RSVP Sender information
*Mar 1 05:28:57.302: SENDER_TSPEC type 2 length 36:
*Mar 1 05:28:57.302: version=0, length in words=7
*Mar 1 05:28:57.302: Token bucket fragment (service_id=1, length=6 words
*Mar 1 05:28:57.302: parameter id=127, flags=0, parameter length=5
*Mar 1 05:28:57.302: average rate=3000 bytes/sec, burst depth=120 bytes
*Mar 1 05:28:57.302: peak rate =3000 bytes/sec
*Mar 1 05:28:57.306: min unit=60 bytes, max pkt size=60 bytes
Comment: Reservation parameters contained in TSPEC
*Mar 1 05:28:57.306: ADSPEC type 2 length 48:
*Mar 1 05:28:57.306: version=0 length in words=10
*Mar 1 05:28:57.306: General Parameters break bit=0 service length=8
*Mar 1 05:28:57.306: IS Hops:1
*Mar 1 05:28:57.306: Minimum Path Bandwidth (bytes/sec):193000
*Mar 1 05:28:57.306: Path Latency (microseconds):0
*Mar 1 05:28:57.306: Path MTU:1500
*Mar 1 05:28:57.306: Controlled Load Service break bit=0 service length=0
Comment: Minimum bandwidth, latency, and MTU requirements
*Mar 1 05:28:57.306:
*Mar 1 05:28:57.346: RSVP: version:1 flags:0000 type:PATH cksum:0000 ttl:254
reserved:0 length:136
*Mar 1 05:28:57.346: SESSION type 1 length 12:
*Mar 1 05:28:57.350: Destination 192.168.2.2, Protocol_Id 17, Don't Police ,
DstPort 18180
Comment: RSVP PATH information from 192.168.2.2
Comment: Reservation information - IP addresses and port numbers
*Mar 1 05:28:57.350: HOP type 1 length 12: C0A80201
*Mar 1 05:28:57.350: : 00000000
*Mar 1 05:28:57.350: TIME_VALUES type 1 length 8 : 00007530
*Mar 1 05:28:57.350: SENDER_TEMPLATE type 1 length 12:
*Mar 1 05:28:57.350: Source 192.168.1.2, udp_source_port 17598
*Mar 1 05:28:57.350: SENDER_TSPEC type 2 length 36:
*Mar 1 05:28:57.354: version=0, length in words=7
*Mar 1 05:28:57.354: Token bucket fragment (service_id=1, length=6 words
*Mar 1 05:28:57.354: parameter id=127, flags=0, parameter length=5
*Mar 1 05:28:57.354: average rate=3000 bytes/sec, burst depth=120 bytes
*Mar 1 05:28:57.354: peak rate =3000 bytes/sec
*Mar 1 05:28:57.354: min unit=60 bytes, max pkt size=60 bytes
*Mar 1 05:28:57.354: ADSPEC type 2 length 48:
*Mar 1 05:28:57.354: version=0 length in words=10
*Mar 1 05:28:57.354: General Parameters break bit=0 service length=8
*Mar 1 05:28:57.354: IS Hops:2
*Mar 1 05:28:57.354: Minimum Path Bandwidth (bytes/sec):193000
*Mar 1 05:28:57.358: Path Latency (microseconds):0
*Mar 1 05:28:57.358: Path MTU:1500
*Mar 1 05:28:57.358: Controlled Load Service break bit=0 service length=0
*Mar 1 05:28:57.358:
*Mar 1 05:28:57.358: RSVP 192.168.1.2_17598-192.168.2.2_18180: Received PATH
Message for 192.168.2.2(Serial0) from 192.168.2.1, rcv IP ttl=253
*Mar 1 05:28:57.358: RSVP 192.168.1.2_17598-192.168.2.2_18180: start requesting
24 kbps FF reservation for 192.168.1.2(17598) UDP-> 192.168.2.2(18180) on

```

(待续)

```

Serial0 neighbor 192.168.2.1
*Mar 1 05:28:57.366: RSVP 192.168.1.2_17598-192.168.2.2_18180: Sending RESV
message 192.168.2.2(18180) <- 192.168.1.2(17:17598)
*Mar 1 05:28:57.366: RSVP session 192.168.2.2_18180: send reservation to
192.168.2.1 about 192.168.2.2
<text omitted>
Comment: Exchanging RSVP PATH and RSVP messages to create reservations
*Mar 1 05:28:57.450: RSVP 192.168.1.2_17598-192.168.2.2_18180: RESV CONFIRM
message for 192.168.2.2 (Serial0) from 192.168.2.1
Comment: RSVP CONFIRM message
*Mar 1 05:29:08.662: RSVP 192.168.2.2_18180-192.168.1.2_17598: remove sender
host PATH 192.168.1.2(17598) <- 192.168.2.2(17:18180)
*Mar 1 05:29:08.662: RSVP 192.168.2.2_18180-192.168.1.2_17598: remove Serial0
RESV 192.168.1.2(17598) <- 192.168.2.2(17:18180)
*Mar 1 05:29:08.662: RSVP 192.168.2.2_18180-192.168.1.2_17598: remove sender
host PATH 192.168.1.2(17598) <- 192.168.2.2(17:18180)
*Mar 1 05:29:08.666: RSVP session 192.168.1.2_17598: send path teardown
multicast about 192.168.1.2 on Serial0
Comment: Teardown session, remove sender 192.168.1.2
<packet data omitted>
*Mar 1 05:29:08.678: RSVP 192.168.1.2_17598-192.168.2.2_18180: remove receiver
host RESV 192.168.2.2(18180) <- 192.168.1.2(17:17598)
*Mar 1 05:29:08.678: RSVP 192.168.1.2_17598-192.168.2.2_18180: remove Serial0
RESV request 192.168.2.2(18180) <- 192.168.1.2(17:17598)
*Mar 1 05:29:08.678: RSVP session 192.168.2.2_18180: send reservation teardown
to 192.168.2.1 about 192.168.2.2
Comment: Teardown session, remove receiver 192.168.2.2
*Mar 1 05:29:08.682: RSVP:      version:1 flags:0000 type:RTEAR cksum:572F ttl:255
reserved:0 length:100
<packet data omitted>
*Mar 1 05:29:08.702: RSVP 192.168.1.2_17598-192.168.2.2_18180: PATH TEAR message
for 192.168.2.2 (Serial0) from 192.168.1.2
Comment: RSVP TEAR message from 192.168.1.2
*Mar 1 05:29:08.706: RSVP 192.168.1.2_17598-192.168.2.2_18180: remove Serial0
PATH 192.168.2.2(18180) <- 192.168.1.2(17:17598)
*Mar 1 05:29:08.714: RSVP:      version:1 flags:0000 type:RTEAR cksum:0000
ttl:255
<packet data omitted>
*Mar 1 05:29:08.726: RSVP 192.168.2.2_18180-192.168.1.2_17598: RESV TEAR message
for 192.168.1.2 (Serial0) from 192.168.2.1
Comment: RSVP TEAR message from 192.168.2.2

```

当呼叫初始发起后，应当可以看到建立 RSVP 会话的 RSVP PATH 和 RESV 信息。在呼叫的进行中，应当可以看到更进一步的 RSVP PATH 和 RESV 信息作为 hello 信息发送，以维护整个呼叫中的 RSVP 会话。RSVP PATH 信息应当含有整个呼叫的 RSVP 预留参数，包括平均速率、每秒的字节数、突发深度、峰值速率和包的尺寸。当你挂断电话后，应当可以看到 RSVP TEARDOWN 信息。除了 RSVP **debug** 的输出外，可以使用本章早先列出的 **show** 命令来显示 RSVP 的配置。

范例 5-19 显示了本实验中所有路由器的完整配置。

范例 5-19 实验中所有路由器的完整配置

```

The Hub-140 Router
interface Serial0
 encapsulation frame-relay
 fair-queue 64 256 48

```

(待续)

```

frame-relay lmi-type ansi
ip rsvp bandwidth 1536 1536
!
interface Serial0.1 multipoint
ip address 192.168.1.1 255.255.255.252
ip ospf network non-broadcast
frame-relay map ip 192.168.1.2 841 broadcast
ip rsvp bandwidth 1158 24
!
interface Serial0/0.2 multipoint
ip address 192.168.2.1 255.255.255.252
ip ospf network non-broadcast
frame-relay map ip 192.168.2.2 941 broadcast
ip rsvp bandwidth 1158 24
!
router ospf 1
network 192.168.1.0 0.0.0.3 area 0
network 192.168.2.0 0.0.0.3 area 0
neighbor 192.168.2.2
neighbor 192.168.1.2 priority 1

```

The Store-148 Router

```

!
interface Serial0
ip address 192.168.1.2 255.255.255.252
encapsulation frame-relay
fair-queue 64 256 37
frame-relay lmi-type ansi
ip ospf network non-broadcast
ip ospf priority 0
frame-relay map ip 192.168.1.1 148 broadcast
ip rsvp bandwidth 1158 24
!
router ospf 1
log-adjacency-changes
network 192.168.1.0 0.0.0.3 area 0
neighbor 192.168.1.1 priority 1
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer voice 4801 pots
destination-pattern 4801
port 2/0
!
dial-peer voice 4901 voip
destination-pattern 4901
session target ipv4:192.168.2.2
req-qos guaranteed-delay

```

The Store-149 Router

```

interface Serial0
ip address 192.168.2.2 255.255.255.252
encapsulation frame-relay IETF
fair-queue 64 256 37
frame-relay lmi-type ansi
ip ospf network non-broadcast
ip ospf priority 0
clockrate 1300000
frame-relay map ip 192.168.2.1 149 broadcast

```

(待续)

```
ip rsvp bandwidth 1158 24
!
!
router ospf 1
 network 192.168.2.0 0.0.0.3 area 0
 neighbor 192.168.2.1 priority 1
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer voice 4901 pots
 destination-pattern 4901
 port 2/0
!
dial-peer voice 4801 voip
 destination-pattern 4801
 session target ipv4:192.168.1.2
 req-qos guaranteed-delay
```

既然你已经看到可以使用集成服务来提供端对端的服务质量，现在来看看区分服务是如何对特定的服务质量数据包进行分类的。

5.3 区分服务

区分服务通常也称为 DiffServ，提供了将数据包分成类别或者服务类别（COS）的方法。类别服务是通过 IP 报头的服务类型（TOS）字段中的值来定义的。这个字段的内容最初是在 RFC 1122 和 1349 中定义的，作为 Precedence 和服务类型字段来定义。几个工作组对数据包的分类方法做出了许多变种，但是许多努力直到最近多业务应用程序开始需要通过网络得到更多的质量控制和调整才最终变为现实。RFC 1349 定义了 ToS 字节中的第 3 到第 6 位作为表 5-10 所示的服务类型的定义。ToS 字段最初是作为一种机制将数据包分成不同的服务类型来满足应用程序对延迟、吞吐、可靠性和费用的网络需求。

注意：DiffServ 服务类别不要和二层的 service class 混淆，例如本地局域网上的 Inter-Switch Link（ISL）或者 802.1Q 的帧标记服务。本章只使用术语 *服务类别* 来参考三层数据包的标记。

表 5-10

服务类型值

十六进制制位	十进制值	服务类型	思科 IOS 软件 ToS 值
0000	0	正常	normal
1000	8	最小延迟	min-delay
0100	4	最大吞吐	max-throughput
0010	2	最大可靠性	max-reliability
0001	1	最小费用	min-monetary-cost

使用 ToS 值，可以标记来自某些应用程序的数据包，并且当拥塞发生时，可以在网络中以后使用那个分类信息来提供对这些应用程序的更高级别的服务。默认的情况下，所有的 IP 数据包的 ToS 值都为 0000，指定它们都应当采用“尽力传递”的服务。在思

科 IOS 软件中，我们可以使用访问控制列表来改变应用程序的 ToS 的值，如范例 5-20 所示。使用访问控制列表，可以对数据包使用 ToS 值的名字或者 0~15 的十进制值来分类数据包。

范例 5-20 在访问控制列表中使用 ToS 值

```
interface Serial1
ip address 192.168.1.2 255.255.255.252
ip ospf network non-broadcast
ip ospf priority 0
ip policy route-map throughput
frame-relay map ip 192.168.1.1 148 broadcast
!
ip local policy route-map throughput
!
access-list 150 permit udp host 192.168.1.2 range 16384 32767 host 192.168.2.2
range 16384 32767
access-list 150 permit udp host 192.168.2.2 range 16384 32767 host 192.168.1.2
range 16384 32767
access-list 150 permit tcp host 192.168.1.2 eq 1720 host 192.168.2.2
access-list 150 permit tcp host 192.168.1.2 host 192.168.2.2 eq 1720
!
route-map throughput permit 10
match ip address 150
set ip tos max-throughput
!
dial-peer voice 4801 pots
destination-pattern 4801
port 2/0
!
dial-peer voice 4901 voip
destination-pattern 4901
session target ipv4:192.168.2.2
```

在先前的范例中，**route-map throughput** 用于标记访问控制列表 150（从端口 16 384 到 32 767 的 UDP 流量和 TCP 端口号为 1720 的流量）中所有的语音和信令流为最大吞吐的 ToS 值。这些信息可以在网络中以后使用，来使用区分服务的一些应用（例如数据包的分类、标记和流量整形及限速）对语音提供最好级别的服务。

本章的剩余部分集中于区分服务的技术，探讨使用 IP Precedence、差分服务编码点（DSCP）的数据包标记，以及使用加权随机早期检测（WRED）实现拥塞控制的一些技术。下一章将介绍一些高级流量整形和限速特性，例如使用通用流量整形和基于类别的整形、流量监管和使用承诺的访问速率（CAR）。

5.3.1 设置 IP Precedence

IP Precedence 是 IP 报头的 ToS 区域中的一个字段。有 8 种级别的优先级，从 0~7，如表 5-11 所示。就像 TOS 值一样，IP 优先级的值可以通过对流量进行分类来设置。

表 5-11 IP 优先级的值

值	描述
Routine (0)	IP 数据包的默认设置
Priority (1)	设置 priority 的优先级

续表

值	描述
Immediate (2)	设置 immediate 优先级
Flash (3)	设置 Flash 优先级
Flash-Override (4)	设置 Flash-override 优先级
Critical (5)	非路由器流量的最高设置
Internet (6)	设置 Internet 控制的优先级，对路由流量保留，例如路由更新数据包
Network Control (7)	设置网络控制优先级，对路由流量和网络控制流量保留

当改变 IP 数据包的优先级时，要非常重视两件事情。首先，默认情况下，所有的 IP 流量，除了路由器产生的控制和路由流量，都使用 routine 优先级的值。如果你没有做出任何变化，所有的 IP 数据包都使用这个设置。其次，当改变 IP 优先级的值时，虽然也可以使用 Internet 和 Network Control 的值，但是这些值通常是路由器和网络控制流量保留的。将它们用于其他类型的流量将中断路由器的操作，中断网络服务。

对于思科路由器，设置 IP 优先级的最简单的方法之一就是使用路由映射。关于配置路由映射的更多信息，请参考第 2 章。需要两个基本步骤来使用路由映射设置 IP 优先级：定义需要做出改变的数据包，建立路由映射来指定这种变化。

第 1 步 使用标准或者扩展访问控制列表来定义需要做设置变化的数据包，将这种流量的优先级值进行改变。下列访问控制列表指定了来自主机 10.1.1.4 的所有流量。

```
Router(config)# access-list 15 permit host 10.1.1.4
```

第 2 步 建立一个路由映射来指定需要做改变的数据包和需要做的变化。

```
Router(config)# route-map precedence
Router(config-route-map)# match ip address 15
Router(config-route-map)# set ip precedence ?
<0-7>          Precedence value
critical       Set critical precedence (5)
flash         Set flash precedence (3)
flash-override Set flash override precedence (4)
immediate     Set immediate precedence (2)
internet      Set internetwork control precedence (6)
network       Set network control precedence (7)
priority      Set priority precedence (1)
routine       Set routine precedence (0)
<cr>
Router(config-route-map)# set ip precedence 5
Router(config-route-map)# exit
```

第 3 步 将路由映射绑定到一个接口上，使用 ip policy route-map 命令实现。

```
Router(config)# interface ethernet 0/0
Router(config-if)# ip policy route-map precedence
```

为了监控策略的状态，可以使用 show route-map 命令或者 debug ip policy。show route-map 命令显示关于路由映射的配置和统计数字。debug ip policy 显示与策略匹配和不匹配的数据包。要特别注意在生产性路由器上使用 debug ip policy 命令，如果这个策略正常工作并且你有太多匹配的数据包，要么会对路由器产生太大的负荷，要么不能看见调试信息。范例 5-21 显示了使用 show route-map 命令的输出。

范例 5-21 show route-map 命令

```
Router# show route-map precedence
route-map precedence, permit, sequence 10
  Match clauses:
    ip address (access-lists): 15
  Set clauses:
    ip precedence critical
  Policy routing matches: 5 packets, 766 bytes
Router# debug ip policy
00:38:09: IP: s=10.1.1.1 (local), d=10.1.1.4, len 100, policy match
00:38:09: IP: route map precedence, item 15, permit
00:38:09: IP: s=10.1.1.1 (local), d=10.1.1.4, len 100, policy rejected -- normal forwarding
00:38:09: IP: s=10.1.1.1 (local), d=10.1.1.4, len 100, policy match
00:38:09: IP: route map precedence, item 15, permit
```

注意：**ip policy route-map route-map-name** 命令用于在每一个接口的基础上绑定策略性路由。这并不包括路由器本地产生的数据包。为了将策略性路由绑定到路由器本地产生的流量，在全局配置模式下使用 **ip local policy route-map route-map-name** 命令。

互连网络的标准在不断更新，新的数据包分类方法不断地加入到思科 IOS 软件中。在写本书时，已经有几种新的方法使用 IP 优先级值来分类数据包和对打标记的数据包操作。这些方法如下：

- 使用访问控制列表来标记数据包。
- 使用路由映射或者策略性路由来标记数据包。
- 使用 RSVP 及数据包分类。
- 分类数据包，做队列优化，使用加权公平队列、优先级队列（PQ）、定制队列（CQ）和基于类别的加权公平队列（CBWFQ）。
- 使用 CAR 和流量监管的高级数据包分类。
- 对分类的流量进行整形，使用通用流量整形（GTS）、基于类别的整形和帧中继的流量整形（FRTS）。
- 通过设置 IP RTP 优先级来优化实时数据协议（RTP）流量。
- 使用低延迟队列（LLQ）来优化实时流量。
- 使用 WRED 的拥塞控制。
- 使用 DiffServ 值来标记语音流量。

FRTS 是一个例外，它在《CCIE 实验指南（第 1 卷）》中论述过，这些技术在第 6 章中进行讨论。不幸的是，因为本书必须限定在一定的篇幅中，我们必须将它们限定在一定的页数并且在本书出版时，最终停止写这部分。因此，我们不会对每一种数据包进行分类进行详细地论述。

5.3.2 使用 DSCP 标记流量

在过去一些年中，IP 报头中的 ToS 字段已经被重新定义来支持新的 DiffServ 特性。新的区分服务（DS）字段含有两个子字段，它们被划分成所谓的编码点。编码点基本上是在 IP 报头中 DS 字段的子分类，它含有和 DSCP 字段相同的值。DS 字段含有两个编码点：*类选择编码点*，

先前被称为 IP 优先级字段；**确保转发 (AF) 编码点**。为了保持和 IP 优先级的兼容，类选择编码点是位 0、1 和 2 (DS 字段中 XXX000 的头 3 位)。DS 字段的头 6 位属于 DSCP 字段，它可以产生 64 个类别用于数据包的标记。AF 编码点在本节的后面介绍。

RFC 2474 和 RFC 2475 描述了对于通过使用 DSCP 字段作为数据包的标记的区分服务应用程序的定义和体系结构。数据包的标记基本上是一个读、使用或改变 DSCP 字段的值的过程，提供对于流量的调控、整形或者限速的**每一跳行为 (PHB)**。PHB 被定义为在一个遵循区分服务的设备上应用于**行为汇聚 (BA)**的行为或者转发对待。BA 是向同一个方向的具有相同编码点的数据包。

注意：DSCP 字段的使用在 RFC 2474、2475、2597、2598 和 2697 中定义，并且以后也在 RFC 3168 和 3260 中进行了更新。

在这些 64 个 DSCP 类别中，IETF 指定了 3 个类池，如表 5-12 所示。第一个地址池使用了 DSCP 字段的头 5 位，以 0 位为终止位，是为标准类分配保留的，它由 IANA 管理。例如，前缀 000、001、010、011、100、101、110 和 111 是保留和 IP 优先级兼容的。000000 保留用于尽力传递服务，并且任何流量如不匹配任何类将被发送到 000000 编码点。

表 5-12 DSCP 池

地址池的号码	编码点的值	保留
1	位 0, 1, 2, 3, 4 xxxxx0	为 IANA 所管理的标准保留
2	位 0, 1, 2, 3 xxxxx11	为实验性或本地使用保留
3	位 0, 1, 2, 3 xxxxx01	为实验性或本地使用和将来的扩展保留

使用 DSCP 字段作为数据包的标记允许对流量优化建立许多分类，当你使用的是需要确保带宽、具有低抖动和延迟的流量例如语音或者视频流量时，这是非常有用的。出于这个原因，RFC 2598 描述了**加速转发 (EF) PHB**。EF PHB 提供了区分服务所定义的最高级别的服务质量。EF PHB 为高优先级的流量提供了 AF 类别，这个类别具有 101110 的值，作为最高的优先级提供最好的质量。

也可以将 DSCP 的值和 WRED 使用来控制对 TCP 数据包的预先丢弃，通过指定 AF 的类别来实现。RFC 2597 定义了 AF 的类别来作为数据包丢弃优先级的标准。为了解释在一个网络环境中的 AF 类别，假设你定义了 3 种类别的流量作为高优先级，然而，当网络拥塞达到一定点，数据包开始丢弃时，使用 AF 类别，可以指定数据包的丢弃顺序。表 5-13 显示了 AF 类别和它们的丢弃优先级。类别 1 中所有的位起始于标准的 IP 优先级值 001，它是 priority 优先级。类别 2 开始于值 010，它是 immediate 优先级，类别 3 开始于值 011，它是 Flash 优先级，类别 4 开始于值 100，它是 Flash-override 优先级。

注意：WRED 的使用在本章的后面详细讨论。

表 5-13 AF 类别和丢弃优先级

丢弃顺序	类 1	类 2	类 3	类 4
低丢弃	AF11 DSCP 10 001010	AF21 DSCP 18 010010	AF31 DSCP 26 011010	AF41 DSCP 34 100010
中等丢弃	AF12 DSCP 12 001100	AF22 DSCP 20 010100	AF32 DSCP 28 011100	AF42 DSCP 36 100100
高丢弃	AF13 DSCP 14 001110	AF23 DSCP 22 010110	AF33 DSCP 30 011110	AF43 DSCP 38 100110

DSCP 的值在思科 IOS 软件中有多种用法。它可以和访问控制列表使用来指定 IP 数据包中的 DSCP 值。它也可以和分级映射（class map）和策略映射（policy map）使用来标记数据包。DSCP 位也可以和 CAR 使用来基于一个数据包的 DSCP 值指定对这个数据包的转发行为。DSCP 还可以和 WRED 使用来指定在数据包的预丢弃情况下，哪些流量预先被丢弃。表 5-14 显示了可以设置的 DSCP 值，要么通过名字，要么通过十进制的数字，以及它们的描述。

表 5-14 思科 IOS 软件的 DSCP 值

DSCP 值	DSCP 十进制和十六进制值	描述
af11	10 001010	AF11—确保转发，低丢弃可能性，类 1 DSCP 和 priority 优先级
af12	12 001100	AF12—确保转发，中等丢弃可能性，类 1 DSCP 和 priority 优先级
af13	14 001110	AF13—确保转发，高丢弃可能性，类 1 DSCP 和 priority 优先级
af21	18 010010	AF21—确保转发，低丢弃可能性，类 2 DSCP 和 immediate 优先级
af22	20 010100	AF22—确保转发，中等丢弃可能性，类 2 DSCP 和 immediate 优先级
af23	22 010110	AF23—确保转发，高丢弃可能性，类 2 DSCP 和 immediate 优先级
af31	26 011010	AF31—确保转发，低丢弃可能性，类 3 DSCP 和 Flash 优先级
af32	28 011100	AF32—确保转发，中等丢弃可能性，类 3 DSCP 和 Flash 优先级
af33	30 011110	AF33—确保转发，高丢弃可能性，类 3 DSCP 和 Flash 优先级
af41	34 100010	AF41—确保转发，低丢弃可能性，类 4 DSCP 和 Flash-override 优先级
af42	36 100100	AF42—确保转发，中等丢弃可能性，类 4 DSCP 和 Flash-override 优先级
af43	38 100110	AF43—确保转发，高丢弃可能性，类 4 DSCP 和 Flash-override 优先级
cs1	1 001000	CS1 或者 priority IP 优先级 1
cs2	2 010000	CS2 或者 immediate IP 优先级 2
cs3	3 011000	CS3 或者 Flash IP 优先级 3
cs4	4 100000	CS4 或者 Flash-override IP 优先级 4
cs5	5 101000	CS5 或者 Critical IP 优先级 5
cs6	6 110000	CS6 或者 Internet IP 优先级 6
cs7	7 111000	CS7 或者 Network Control IP 优先级 7
default	0 000000	所有流量默认的“尽力传递”的值
ef	46 101110	EF-PHB——加速转发，最高的服务级别

对 DSCP 进行分类的最标准应用程序就是访问控制列表。范例 5-22 演示了两种方法来使

用 AF DSCP 的值标记所有的 UDP 语音流量，使其具有最低的丢弃可能性和最高的优先级。对语音信令流量推荐的 DSCP 值是 DSCP 26 或者 AF31。这在本质上和将流量用 Flash IP 优先级值来标注是一样的。通过将数据包使用 AF31 DSCP 的值来标注，可以确保队列或者拥塞控制机制，例如加权公平队列或者 WRED，将会给这些数据包较高的优先级，推荐为语音信令流量使用，使其具有最低的丢弃可能性，并且还可以使用其他更先进的方法来控制给这些应用程序所提供的质量级别。

范例 5-22 使用 DSCP 分类来优化语音流量

```
interface Serial1
 ip address 192.168.1.2 255.255.255.252
 frame-relay map ip 192.168.1.1 148 broadcast
 ip rsvp bandwidth 1158 24
 ip rsvp signalling dscp 26
!
dial-peer voice 4801 pots
 destination-pattern 4801
 port 2/0
!
dial-peer voice 4901 voip
 destination-pattern 4901
 session target ipv4:192.168.2.2
 req-qos guaranteed-delay
 ip qos dscp af31 signalling
```

在先前的范例中，**ip rsvp signalling dscp 26** 命令用于给 RSVP 的信令流量分配 AF DSCP 的值 af31（低丢弃/Flash）。第二个要关注的命令是 **ip qos dscp af31 signalling**，它对语音信令流量提供优化分类，它也可以在网络中的其他部分进行流量的优化。每一个命令都可以在拥塞发生时，使这两种不同的协议在加权公平队列或者拥塞策略中收到更高的优先级。

注意：用 DSCP 的值标记流量本身来说并不确保流量将会在网络中受到更好地对待。数据包的标记只是识别流量，使得你可以在网络中的其他地方对那种流量实施服务质量的技术。

既然我们已经学习了使用 DSCP 的值分类流量来进行拥塞控制，现在我们来研究拥塞控制本身是如何工作的以及它是如何配置的。

5.3.3 使用 WRED 避免拥塞

当没有采用任何拥塞避免机制的时候，接口会基于尾丢弃的方法来丢弃数据包。尾丢弃基本上意味着当一个接口的队列满了以后，任何新的到达那个接口需要传输的数据包会被丢弃，直到接口有足够的队列空间来服务新的数据包。另外一种管理网络拥塞的方法是避免它。加权随机早期检测（WRED）就是用于做这件事情。基于随机早期检测（RED）的算法，由 Sally Floyd 和其他的开发者开发，WRED 可以基于估测的平均队列大小预先丢弃数据包，最小的队列尺寸，这时没有任何数据包被丢弃，最大的队列尺寸，这时所有的数据包都被丢掉。在队列上发生拥塞时，WRED 会丢弃数据包来防止一种称为 *全局同步* 的现象发生。

注意：关于 RED 的更多信息，参见 RFC 2309 和 <http://ftp.ee.lbl.gov/floyd/red.html> 或者 Sally Floyd 的站点 <http://www.icir.org/floyd/>。

全局同步发生在当网络发生拥塞并且数据包被丢弃时，导致所有的 TCP 终端工作站同时降低发送速率并重传丢弃的数据包，这将浪费网络资源。在全局同步发生时，网络的流量将会持续到峰值然后又降到谷底，这是因为所有运行 TCP 应用程序的工作站已经被同步了。WRED 可以防止全局同步，通过从大的流量预先丢弃数据包，导致占优势的网络工作站减少它们的 TCP 窗口尺寸，发送较少的数据包，减少它们对网络的利用率，给一些小的流量留出更多的空间，并且防止更多的数据包被丢弃。

注意：WRED 和 RED 的主要区别是 WRED 会基于流量的 IP 报头中的 IP 优先级来对流量实施权重，而 RED 不会。使用 WRED，高优先级的流量会有较高的权重，在网络发生拥塞的时候被丢弃的可能性就越小。

我们要特别注意，WRED 只对 TCP 的流量工作，因为 TCP 是面向连接的，需要使用窗口和确认去实施流量控制。而 UDP、IP 和其他非 IP 的协议，例如互连网络数据包交换协议 (IPX) 和 AppleTalk 是非面向连接的协议，并且不像 TCP 那样提供窗口机制，它们可能由于 WRED 的作用受到不好影响。如果接口的拥塞中含有许多非面向连接的或者非 IP 的流量，那么 WRED 的拥塞避免方法不会提供任何好处。

在思科 IOS 软件中，可以采用两种方法在接口中配置 WRED。最简单的方法就是在一个接口上使用 **random-detect** 命令来启用接口上的 WRED 功能。当启用 WRED 后，也可以使用 **random-detect exponential-weighting-constant** 命令来配置平均队列深度计算的权重。这个命令指定了当计算平均队列长度时 WRED 所使用的权重，默认的权重是 9。

random-detect exponential-weighting-constant exponent

可接受的指数值范围从 1~16，格式为 2^n 。为了配置 IP 的优先级，使用 **random-detect precedence** 命令来对数据包的丢弃使用权重，这个命令可以指定最小和最大的 WRED 数据包的极限，并且指定丢弃的比率。缺省情况下，IP 优先级为 0 的流量的最小极限是这个接口最大极限流量的一半。表 5-15 解释了最小和最大极限以及各种标记示意的更详细内容。

random-detect precedence precedence-value minimum-threshold maximum threshold
[*mark-probability-denominator*]

表 5-15 WRED 和 IP Precedence 的值

命令参数	描述
<i>precedence-value</i>	特定的要匹配的 IP 优先级的值，范围从 0~7
<i>minimum-threshold</i>	在队列中最小数据包的数量，之后，具有特定优先级的数据包将会被随机丢弃
<i>maximum-threshold</i>	在队列中最大数据包的数量，之后，具有特定优先级的数据包将会被尾部丢弃
[<i>mark-probability-denominator</i>]	(可选) 这个数值代表在拥塞发生时，平均队列长度等于最大的容量时，被丢弃的部分流量。换句话说，在达到最大极限值之前，每 10 个数据包中就会有一个被丢弃

可以通过对不同的优先级设置极限来对不同的网络应用定制 WRED 配置。**mark-probability-denominator** 参数可以改变数据包被丢弃的速率。例如，在默认的情况下，对于 WRED 的接口，**mark-probability-denominator** 的值为 10，所以当数据包处于最小和最大的极限范围内时，每 10 个包中就会有一个被丢掉。当达到最大极限值后，具

有这个优先级值的数据包将会被执行尾部丢弃。范例 5-23 演示了如何配置 WRED 来限制低优先级的队列尺寸，即队列 0~4，并且将关键流量（precedence 5）的最小平均队列深度增加到 35 个数据包。

范例 5-23 采用 WRED 和 IP Precedence

```
Sally-1# show run | begin Serial0
interface Serial0
ip address 289.22.78.1 255.255.255.0
ip ospf network point-to-point
no ip mroute-cache
random-detect
random-detect precedence 0 17 40
random-detect precedence 1 19 40
random-detect precedence 2 21 40
random-detect precedence 3 23 40
random-detect precedence 4 25 40
random-detect precedence 5 35 40 20
```

范例 5-24 显示了配置 WRED 参数前后，在串行接口上使用 `show queueing random-detect` 命令的输出。

范例 5-24 WRED 配置变化前后

```
Sally-1# show queueing random-detect
Current random-detect configuration:
Serial0
  Queueing strategy: random early detection (WRED)
  Exp-weight-constant: 9 (1/512)
  Mean queue depth: 0
  Class Random Tail Minimum Maximum Mark
        drop drop threshold threshold probability
    0      0      0      20      40      1/10
    1      0      0      22      40      1/10
    2      0      0      24      40      1/10
    3      0      0      26      40      1/10
    4      0      0      28      40      1/10
    5      0      0      31      40      1/10
    6      0      0      33      40      1/10
    7      0      0      35      40      1/10
  rsvp    0      0      37      40      1/10

Sally-1# show queueing random-detect
Current random-detect configuration:
Serial0
  Queueing strategy: random early detection (WRED)
  Exp-weight-constant: 9 (1/512)
  Mean queue depth: 0
  Class Random Tail Minimum Maximum Mark
        drop drop threshold threshold probability
    0      0      0      17      40      1/10
    1      0      0      19      40      1/10
    2      0      0      21      40      1/10
    3      0      0      23      40      1/10
    4      0      0      25      40      1/10
    5      0      0      35      40      1/20
    6      0      0      33      40      1/10
    7      0      0      35      40      1/10
  rsvp    0      0      37      40      1/10
```

就像在先前的范例中所看到的，**show queueing random-detect** 命令显示了对每一个 WRED 启用接口的 WRED 配置，包括指数权重常数，对每一个优先级所丢弃的数据包的数量，对 8 个 IP 优先级值中的每一个和 RSVP 的最小和最大的极限值。

就像以前所提到的，默认情况下，WRED 和 IP 优先级工作来防止高优先级的数据包在发生拥塞的情况下被丢掉。如果流量持续较高增长，接口依旧处于拥塞状态，而数据包处于最小和最大的极限之间，处于某个特定优先级值的数据包就会按照配置的极限被丢掉。WRED 也可以和 DSCP 值配合一起工作，可以使用 **random-detect dscp-based** 命令，如范例 5-25 所示。

注意：在接口收到高流量的非 TCP 的流量时，标记了 IP 优先级或者 DSCP 值的高优先级的流量可能会超过最大的极限值，导致这种优先级的数据包被尾部丢弃。

范例 5-25 在 WRED 中使用 DSCP 的值

```
Store-148#sho run | begin Serial1
interface Serial1
  no ip address
  encapsulation frame-relay
  random-detect dscp-based
  frame-relay lmi-type ansi
```

在先前的范例中，WRED 被配置为使用 DSCP 的值作为权重，而不是使用 IP 优先级。WRED 在和 DSCP 的分类结合使用的时候能力会变得非常强。不是支持 8 个基于 IP 优先级的队列的 WRED，基于 DSCP 的 WRED 支持所有的基于 AF 和 CS 的 DSCP 值，而且每一种队列可以使用 **random-detect dscp-based dscp-value minimum-threshold, maximum-threshold mark-probability-denominator** 命令进行修改。范例 5-26 显示了当配置完基于 DSCP 的 WRED 后 **show queueing** 命令的输出。

范例 5-26 show queueing 和基于 DSCP 的 WRED

```
Sally-1# show queueing
Current fair queue configuration:
  Interface      Discard      Dynamic   Reserved   Link    Priority
                threshold   queues    queues     queues  queues
  Serial0        64          256       37          8        1
Current DLCI priority queue configuration:
Current priority queue configuration:
Current custom queue configuration:
Current random-detect configuration:
  Serial1
    Queueing strategy: random early detection (WRED)
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0
  dscp          Random drop      Tail drop      Minimum Maximum   Mark
                pkts/bytes    pkts/bytes    thresh  thresh  prob
  af11          0/0            0/0            33      40     1/10
  af12          0/0            0/0            28      40     1/10
  af13          0/0            0/0            24      40     1/10
  af21          0/0            0/0            33      40     1/10
  af22          0/0            0/0            28      40     1/10
  af23          0/0            0/0            24      40     1/10
  af31          0/0            0/0            33      40     1/10
```

(待续)

af32	0/0	0/0	28	40	1/10
af33	0/0	0/0	24	40	1/10
af41	0/0	0/0	33	40	1/10
af42	0/0	0/0	28	40	1/10
af43	0/0	0/0	24	40	1/10
cs1	0/0	0/0	22	40	1/10
cs2	0/0	0/0	24	40	1/10
cs3	0/0	0/0	26	40	1/10
cs4	0/0	0/0	28	40	1/10
cs5	0/0	0/0	31	40	1/10
cs6	0/0	0/0	33	40	1/10
cs7	0/0	0/0	35	40	1/10
ef	0/0	0/0	37	40	1/10
rsvp	0/0	0/0	37	40	1/10
default	0/0	0/0	20	40	1/10
Current per-SID queue configuration:					

WRED 信息也可以使用 **show interface** 命令汇总，如范例 5-27 所示，它可以显示丢弃数据包的数量。

范例 5-27 show interface 命令和 WRED

```
Sally-1# show interface serial 0
Serial0 is up, line protocol is up
Hardware is PQQUICC with 56k 4-wire CSU/DSU
Internet address is 2.2.2.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:17, output 00:00:02, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: random early detection(RED)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 2826 packets input, 201606 bytes, 0 no buffer
Received 2821 broadcasts, 0 runts, 0 giants, 0 throttles
1427 input errors, 99 CRC, 479 frame, 0 overrun, 0 ignored, 841 abort
3934 packets output, 274630 bytes, 0 underruns
0 output errors, 0 collisions, 243 interface resets
0 output buffer failures, 0 output buffers swapped out
175 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

WRED 也包括对 RSVP 的支持。默认的情况下，WRED 对 RSVP 的流量有一个 37 字节的最小平均队列尺寸，是所有的平均队列尺寸中最大的。可以使用 **random-detect precedence rsvp** 或者 **random-detect dscp rsvp** 命令来定制 RSVP WRED 的配置，配置最小和最大的平均队列尺寸。

注意：如果你正在运行 WRED 的接口上规划使用先进先出队列，而且你正在考虑其他的队列方法，例如加权公平队列、定制队列或者优先级队列，将来你会意识到 WRED 和加权公平队列、定制队列和优先级队列都是互相排斥的技术。当 WRED 被配置完后，在启用任何其他队列技术之前，WRED 必须被清除掉。

WRED 也可以配置来支持单独的流量流。基于流的 RED 通常被称为 FRED。每一股流含有源和目的 IP 地址和端口号码。FRED 监控每一股流的状态信息并且防止任何资源消耗的流独占资源，这通过给每一股流分配缓冲区来实现。

为了启用 FRED，必须首先使用 **random-detect** 命令来启用 WRED，接着使用 **random-detect flow** 命令启用 FRED，如果需要，配置平均队列深度和所允许的动态队列的数量。默认情况下，FRED 被限制为 256 个流，平均队列的深度因子为 4。平均队列深度用于对每一个流扩展缓存的数量。它决定了在每一个队列中所允许的数据包的数量，并且可以通过 **random-detect flow average-depth-factor** 命令进行配置。深度因子可以是 1、2、4、8 或者 16，默认的平均队列深度因子是 4。

```
random-detect flow average-depth-factor depth-factor
```

可以通过使用 **random-detect flow count** 命令来设置活动的流的最大数量。这个流的数量可以从 16~32 768，默认值为 256 个流。

```
random-detect flow count flow-count
```

这些 FRED 流的配置工具允许用户建立更加灵活的拥塞控制配置，使得用户可以基于 DSCP 的值来对流量实施不同的拥塞控制行为，限制流的数量，并且定义队列的尺寸，如范例 5-28 所示。

范例 5-28 建立定制的 WRED 配置

```
Store-148#sho run | begin Serial1
interface Serial1
no ip address
encapsulation frame-relay
random-detect dscp-based
random-detect flow
random-detect flow average-depth-factor 2
frame-relay lmi-type ansi
```

先前的范例建立了 3 个新的字段，这可以在 **show queueing random-detect** 命令的输出中看到。平均队列深度 (*mean queue depth*) 在 WRED 启用后也可以显示，显示了每一个队列的最小和最大队列深度的平均值。最大流 (*Max flow*) 显示了在当前的配置中所允许的最大的流的数量。平均深度因子 (*Average depth factor*) 显示了当前的平均深度因子的配置，而流 (*flow*) 字段显示了活动的流的数量，活动的流的最大数量，根据当前配置的活动的流的最大可能数量。范例 5-29 显示了 **show queueing random-detect** 命令的输出，正好在范例 5-28 的配置显示之后。

范例 5-29 在流配置之后 show queueing 命令的输出

```
Sally-1# show queueing random-detect interface Serial 1
Current random-detect configuration:
Serial1
Queueing strategy: random early detection (WRED)
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0
Max flow count: 256      Average depth factor: 2
```

(待续)

Flows (active/max active/max): 0/0/256					
dscp	Random drop	Tail drop	Minimum	Maximum	Mark
	pkts/bytes	pkts/bytes	thresh	thresh	prob
af11	0/0	0/0	33	40	1/10
af12	0/0	0/0	28	40	1/10
af13	0/0	0/0	24	40	1/10
af21	0/0	0/0	33	40	1/10
af22	0/0	0/0	28	40	1/10
af23	0/0	0/0	24	40	1/10
af31	0/0	0/0	33	40	1/10
af32	0/0	0/0	28	40	1/10
af33	0/0	0/0	24	40	1/10
af41	0/0	0/0	33	40	1/10
af42	0/0	0/0	28	40	1/10
af43	0/0	0/0	24	40	1/10
cs1	0/0	0/0	22	40	1/10
cs2	0/0	0/0	24	40	1/10
cs3	0/0	0/0	26	40	1/10
cs4	0/0	0/0	28	40	1/10
cs5	0/0	0/0	31	40	1/10
cs6	0/0	0/0	33	40	1/10
cs7	0/0	0/0	35	40	1/10
ef	0/0	0/0	37	40	1/10
rsvp	0/0	0/0	37	40	1/10
default	0/0	0/0	20	40	1/10

本章介绍了使用集成服务和区分服务来给应用程序提供服务质量的几种方法。在对接收到的已经打了标记的数据包实施队列、整形或者限速机制前，许多技术是难于理解的。好的区分服务设计的最大好处只有在高级队列、整形和限速技术被应用时才可以看到。下一章探讨区分服务如何通过应用更高级队列、整形、限速和分类技术来扩展和增值。

5.4 练习场景

下面的练习场景可以帮助用户牢固掌握我们在本章中强调的某些概念。

实验 11: Jetsons Meet IntServ and DiffServ

集成和区分服务对如今的拥塞网络提供了几种弥补措施。在这个练习的场景中，我们探索这些技术可以使用的一些不同方法来提供更高的网络性能。

一、实验练习

在这个实验的场景中，配置集成和区分服务来对 Jetsons 网络中的用户提供更好的 VoIP 质量。用于这个实验场景的网络将利用本章中谈到的许多技术，包括具有 DSCP 分类的 RSVP 和通过 ATM WAN 的 WRED 的拥塞控制技术。

二、实验目的

在这个实验中，必须达到下面这些目的：

- 使用 RSVP 来对 VoIP 流量预留资源。

- 对某种类型的 RSVP 和语音信令流量实施 DSCP 标记。
- 使用 WRED 对通过广域网的流量进行拥塞控制。
- 采用语音编码来提供最好的压缩、质量和可靠性。
- 对 ATM WAN 的接口采用 WRED 和 RSVP 技术来提高 ATM 的技巧。
- 配置一个 LightStream 1010 ATM 交换机来实现在 ATM 路由器之间的 PVC 连接。

三、所需的设备

需要下面的设备：

- 一台 LightStream ATM 交换机，具有两个 OC-3 模块。
- 两台具有 ATM OC-3 接口的思科路由器，一台路由器至少有一个串行接口，另外一台路由器有一个令牌环的接口。
- 具有一个以太网接口和一个令牌环接口的路由器。
- 具有一个串行接口和一个 FXS 语音接口的路由器，以及一个用于测试的电话。
- 具有一个快速以太网接口和一个 FXS 语音接口的路由器，以及一个用于测试的电话。
- 一个集线器或者交换机来实现以太网连接，一个多工作站访问单元（MSAU）来实现令牌环连接。

注意：这个实验利用了 ATM 设备作为广域网的核心网络。如果你没有 ATM 设备，使用帧中继来仿真这些连接。这个实验也使用了令牌环的接口，然而，因为令牌环不是这个实验中的关键部件，你可以使用以太网来替换它。

四、物理布局和预规划

需要完成下面的物理布局和预规划：

- 按照图 5-6 所示给路由器布线，并且给 ATM 交换机连接 ATM OC-3 接口。

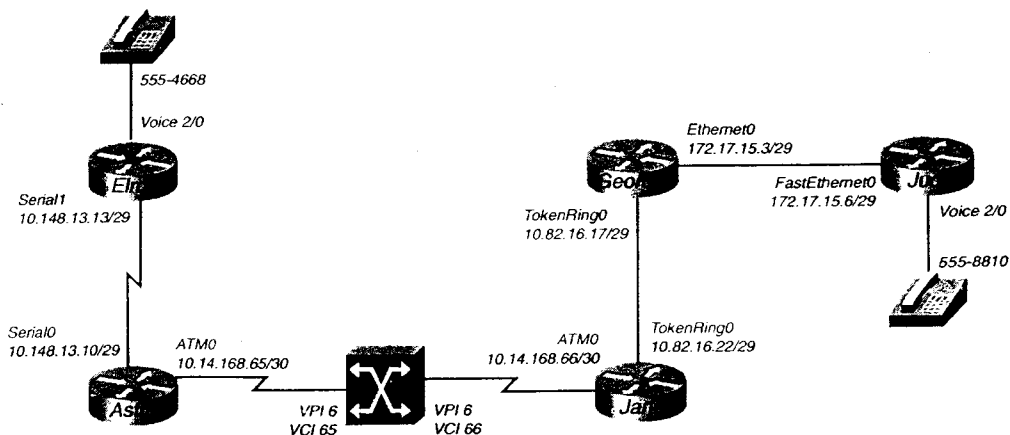


图 5-6 Jetsons 网络

- 使用背对背的线缆连接 Elroy 和 Astro 的串行接口。
- 将 Jane 和 George 路由器连接到 MSAU。
- 将 Judy 和 George 路由器连接到以太网交换机或者集线器上。

- 将电话连接到 Elroy 和 Judy 路由器的 FXS 端口。
- 使用表 5-16 所示的信息配置 ATM 交换机。

表 5-16 ATM PVC 配置

路由器接口	VPI	VCI	交换机接口	VPI	VCI
Astro ATM0	6	65	ATM1/0/2	6	65
Jane ATM0	6	66	ATM1/0/0	6	66

在 ATM 交换机上配置两条 PVC 是一个非常简单的过程。只需要在其中的一个接口上建立一个 ATM PVC 语句，指明在那个 PVC 上的 ATM 流量应当通过连接远端网络的另外一个 ATM 接口。范例 5-30 显示了 ATM 交换机的配置。

范例 5-30 ATM 交换机的配置

```
interface ATM1/0/2
no ip address
atm pvc 6 65 interface ATM1/0/0 6 66
```

- 按照先前的图所示，配置串口、ATM、令牌环和以太接口的地址，验证所有的路由器能够使用 ping 到达它们所直连的邻居。

五、实验的任务

遵循下面的这些步骤来完成本实验：

- 第 1 步** 在所有的路由器上启用增强内部网关路由协议（EIGRP）进程 32074，并且确保它们不会有类汇总网络。在进行第 2 步之前，验证 IP 的连通性。
- 第 2 步** 使用图 5-5 所示的电话号码来为连接到 Elroy 和 Judy 路由器的电话配置 VoIP。这些电话当摘机时能够自动呼叫对方。使用一种最不占用带宽的语音编码的方法。通过从两个方向测试语音呼叫来验证配置。
- 第 3 步** 对所有的 VoIP 流量为了实现确保的延迟服务配置 RSVP 的请求和接受。确保所有的 RSVP 和语音信令流量使用 DSCP 被标记为具有最高优先级的分类。不要允许一个接口使用的带宽超过 Jetsons 网络中最小接口带宽的 50%。最大的流尺寸不应当超过语音编码所需要的流尺寸。在进行第 4 步之前，在两部电话上测试配置，这一步需要一些任务才能工作正常。
- 第 4 步** 在 Astro 和 Elroy 路由器的串行接口上启用 WRED 来控制拥塞。每一台路由器应当基于数据包的 DSCP 值来衡量包的优先级，标记为 000000 DSCP 值的数据包在收到 20 个字节后应当被丢弃。在其他非默认的 DSCP 数据包被 WRED 丢弃之前，不应当有超过默认的 DSCP 数据包。

当对所有的路由器都进行布线后，使用 **show cdp neighbors** 和 **show ip interface brief** 命令验证连接性。它对电缆问题和时钟问题可以节省大量的时间。当验证完二层的连接后，使用图 5-6 所示的信息给每一台路由器分配 IP 地址。当分配完 IP 地址后，使用 **ping** 命令验证所有直连网络之间的三层连通性。接着，当验证完所有直连的路由器的接口都可达后，你就可以进行这个实验的剩余部分了。

六、实验步骤

下面的过程显示了成功地完成本实验的步骤。

第 1 步 在所有的路由器上启用 EIGRP 进程 32074，确保它们没有有类汇总网络。在进行第 2 步之前，验证 IP 的连通性。

这一步看起来相当容易。一开始在你启用 EIGRP 路由协议时，你可能就会注意到 Astro 和 Jane 路由器并没有自动成为邻居，这是因为它们正在通过一个非广播的多点访问（NBMA）ATM 网络互连。需要完成两个任务，才能使得这两个对等体成为 EIGRP 的邻居。

- 建立一个 ATM 的映射列表，将三层的地址映射到二层并且启用广播，这就像帧中继的 **map** 语句，并且使用 **map-group map-list-name** 命令将这个映射列表绑定到 ATM 的子接口，如范例 5-31 所示。

范例 5-31 Astro 路由器的 ATM 配置

```
Astro# show run | begin ATM
interface ATM0
  no ip address
  no atm ilmi-keepalive
!
interface ATM0.20 multipoint
  ip address 10.14.168.65 255.255.255.252
  map-group atm
  atm pvc 20 6 65 aal5snap
!
map-list atm
ip 10.14.168.66 atm-vc 20 broadcast
```

atm map-group 将 IP 地址映射到接口的 ATM 地址。当你将映射组绑定到 ATM 的子接口后，应当使用 **show atm map** 和 **show atm vc** 命令验证 ATM 的配置，如范例 5-32 所示。

范例 5-32 验证 Astro 路由器的 ATM 配置

```
Astro# show atm map
Map list atm : PERMANENT
ip 10.14.168.66 maps to VC 20
, broadcast
Astro# show atm vc
```

Interface	Name	VPI	VC1	Type	Encaps	SC	Peak Kbps	Avg/Min Kbps	Burst Cells	Sts
0.20	20	6	65	PVC	SNAP	UBR	155000			UP

- 可选地，使用 EIGRP 的 **neighbor IP-address interface-name interface-number** 命令建立一个静态的邻居分配。范例 5-33 显示了 Astro 路由器的 EIGRP 配置和 **show ip eigrp neighbors** 命令的输出。

第 2 步 使用图 5-6 所示的电话号码对连接到 Elroy 和 Judy 路由器的 FXS 接口的电话配置 VoIP。当拿起电话时，应当自动呼叫对方。使用一种最不占用带宽的语音编解码方法。在两个方向上通过电话测试来验证这个配置。

范例 5-33 Astro 路由器的 EIGRP 配置

```
Astro# show run | begin eigrp
router eigrp 32074
  network 10.14.168.64 0.0.0.3
  network 10.148.13.8 0.0.0.7
  neighbor 10.14.168.66 ATM0.20
  no auto-summary
Astro# show ip eigrp neighbors
IP-EIGRP neighbors for process 32074
H   Address                Interface    Hold Uptime    SRTT    RTO  Q  Seq Type
   (sec)              (ms)          Cnt Num
1   10.14.168.66           AT0.20      13 00:18:05 1264   5000  0  7  S
0   10.148.13.13           Se0         13 00:19:28   1    200  0  8
```

到目前为止，这一步需要适用于所有 VoIP 的相同的配置原则。配置两个 dial peer：设置目的的模式、会话的目的、端口和编解码方法。最不耗费资源的语音编解码方法是 g.723 的编解码方法。这个配置的惟一不同的地方就是配置了自动拨号。在语音端口下可以很容易地使用 connection plar dial-string 命令完成这个功能。范例 5-34 显示了对 Judy 路由器的 VoIP 配置。这个范例也显示了两路呼叫可以成功地完成。可以在两台路由器上使用 show call active voice 命令显示活动的呼叫汇总信息。

范例 5-34 Judy 路由器的 VoIP 配置和测试数据

```
Judy# show run | begin voice-port
voice-port 2/0
  connection plar 5554668
!
voice-port 2/1
!
dial-peer voice 5558810 pots
  destination-pattern 5558810
  port 2/0
!
dial-peer voice 5554668 voip
  destination-pattern 5554668
  session target ipv4:10.148.13.13
  codec g723ar63
Astro# show call active voice
Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 1
Judy# show call active voice
Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 1
```

第 3 步 对所有的 VoIP 流量为了实现确保的延迟服务配置 RSVP 的请求和接受。确保所有的 RSVP 和语音信令流量使用 DSCP 被标记为具有最高优先级的分类。不要允许一个接口使用的带宽超过 Jetsons 网络中最小接口带宽的 50%。最大的流尺寸不应当超过语音编码所需要的流尺寸。在进行第 4 步之前，在两部电话上测试配置，这一步需要一些任务才能工作正常。

一 在所有的接口上启用 RSVP，使用预留带宽 772 kbps，它是一个串行接口中最

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

小接口带宽的 50%。最大的预留流不应当大于 18 bit/s，这是语音编解码的方法。并且所有 RSVP 的信令流量应当被标记为 EF DSCP 的值。可以使用两个命令来完成这个任务：**ip rsvp bandwidth 772 18** 和 **ip rsvp signalling dscp 46**。

- 其次，需要配置所有的语音流量从网络中请求并且接受确保的延迟服务。这只需要两个配置任务：在 dial-peer 配置模式下，对远端的对等体 Elroy 和 Judy 路由器输入 **req-qos guaranteed-delay**、**acc-qos guaranteed-delay** 和 **ip qos dscp ef signalling** 命令。范例 5-35 显示了 Elroy 路由器的 RSVP 配置。

范例 5-35 Elroy VoIP RSVP 配置

```
Elroy# show run | begin Serial1
interface Serial1
 ip address 10.148.13.13 255.255.255.248
 fair-queue 64 256 26
 ip rsvp bandwidth 772 18
 ip rsvp signalling dscp 46
!
voice-port 2/0
 connection plar 5558810
!
voice-port 2/1
!
dial-peer voice 5554668 pots
 destination-pattern 5554668
 port 2/0
!
dial-peer voice 5558810 voip
 destination-pattern 5558810
 session target ipv4:172.17.15.6
 req-qos guaranteed-delay
 acc-qos controlled-load
 codec g723ar63
 ip qos dscp ef signalling
```

可以使用 **show ip rsvp reservation detail** 命令在 Elroy 路由器上验证这一步。这个命令应当显示和范例 5-36 类似的数据。

范例 5-36 Elroy 路由器上 show ip rsvp reservation detail 命令的输出

```
Elroy# show ip rsvp reservation detail
RSVP Reservation. Destination is 10.148.13.13, Source is 172.17.15.6,
 Protocol is UDP, Destination port is 16394, Source port is 19344
 Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
 Average Bitrate is 18K bits/sec, Maximum Burst is 80 bytes
 Min Policed Unit: 40 bytes, Max Pkt Size: 40 bytes
 Resv ID handle: 0000B801.
 Policy: Forwarding. Policy source(s): Default
RSVP Reservation. Destination is 172.17.15.6, Source is 10.148.13.13,
 Protocol is UDP, Destination port is 19344, Source port is 16394
 Next Hop is 10.148.13.10, Interface is Serial1
 Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
 Average Bitrate is 18K bits/sec, Maximum Burst is 80 bytes
 Min Policed Unit: 40 bytes, Max Pkt Size: 40 bytes
 Resv ID handle: 0000BA01.
 Policy: Forwarding. Policy source(s): Default
```

第4步 接下来，在 Astro 和 Elroy 路由器的串行接口上启用 WRED 来控制拥塞。每一台路由器应当基于数据包的 DSCP 值来衡量包的优先级，标记为 000000 DSCP 值的数据包在收到 20 个字节后应当被丢弃。在其他非默认的 DSCP 数据包被 WRED 丢弃之前，不应当有超过默认的 DSCP 数据包。

这个命令只需要两个任务：启用基于 DSCP 的 WRED 并且对具有默认 DSCP 值的数据包建立一个限制。对 Elroy 路由器的 WRED 配置如范例 5-37 所示。

范例 5-37 对 Elroy 路由器的 WRED 配置

```
Elroy# show run | begin Serial1
interface Serial1
 ip address 10.148.13.13 255.255.255.248
 random-detect dscp-based
 random-detect dscp 0 20 30
 ip rsvp bandwidth 772 18
 ip rsvp signalling dscp 46
```

作为最后一个 WRED 的配置步骤，可以使用 `show queueing random-detect | begin default` 命令来验证 WRED 的默认的 DSCP 值，如范例 5-38 所示。

范例 5-38 在 Elroy 路由器上验证 WRED 配置

```
Elroy# show queueing random-detect | begin default
default      0/0      0/0      20      30  1/10
```

范例 5-39 显示了这个实验的完整配置。

范例 5-39 实验 5 的完整配置

```
Elroy Router Configuration
interface Serial1
 ip address 10.148.13.13 255.255.255.248
 random-detect dscp-based
 random-detect dscp 0 20 30
 ip rsvp bandwidth 772 18
 ip rsvp signalling dscp 46
!
router eigrp 32074
 network 10.148.13.8 0.0.0.7
 no auto-summary
 no eigrp log-neighbor-changes
!
voice-port 2/0
 connection plar 5558810
!
voice-port 2/1
!
dial-peer voice 5554668 pots
 destination-pattern 5554668
 port 2/0
!
dial-peer voice 5558810 voip
 destination-pattern 5558810
```

(待续)

```
session target ipv4:172.17.15.6
req-qos guaranteed-delay
acc-qos controlled-load
codec g723ar63
ip qos dscp ef signalling
```

Astro Router Configuration

```
interface Serial0
 ip address 10.148.13.10 255.255.255.248
 random-detect dscp-based
 random-detect dscp 0 20 30
 clockrate 1300000
 ip rsvp bandwidth 772 18
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 ip rsvp bandwidth 772 18
!
interface ATM0.20 multipoint
 ip address 10.14.168.65 255.255.255.252
 map-group atm
 atm pvc 20 6 65 aal5snap
 ip rsvp bandwidth 772 18
!
router eigrp 32074
 network 10.14.168.64 0.0.0.3
 network 10.148.13.8 0.0.0.7
 neighbor 10.14.168.66 ATM0.20
 no auto-summary
!
map-list atm
 ip 10.14.168.66 atm-vc 20 broadcast
```

Jane Router Configuration

```
interface TokenRing0
 ip address 10.82.16.22 255.255.255.248
 ring-speed 16
 ip rsvp bandwidth 772 18
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 ip rsvp bandwidth 772 18
!
interface ATM0.20 multipoint
 ip address 10.14.168.66 255.255.255.252
 map-group atm
 atm pvc 20 6 66 aal5snap
 ip rsvp bandwidth 772 18
!
router eigrp 32074
 network 10.14.168.64 0.0.0.3
 network 10.82.16.16 0.0.0.7
 neighbor 10.14.168.65 ATM0.20
 no auto-summary
!
map-list atm
 ip 10.14.168.65 atm-vc 20 broadcast
```

George Router Configuration

(待续)


```

interface Ethernet0/0
 ip address 172.17.15.3 255.255.255.248
 ip rsvp bandwidth 772 18
!
interface TokenRing0/0
 ip address 10.82.16.17 255.255.255.248
 ring-speed 16
 ip rsvp bandwidth 772 18
!
router eigrp 32074
 network 10.82.16.16 0.0.0.7
 network 172.17.15.0 0.0.0.7
 no auto-summary

```

Judy Router Configuration

```

interface FastEthernet0
 ip address 172.17.15.6 255.255.255.248
 ip rsvp bandwidth 772 18
 ip rsvp signalling dscp 46
!
router eigrp 32074
 network 172.17.15.0 0.0.0.7
 no auto-summary
!
voice-port 2/0
 connection plar 5554668
!
voice-port 2/1
!
dial-peer voice 5558810 pots
 destination-pattern 5558810
 port 2/0
!
dial-peer voice 5554668 voip
 destination-pattern 5554668
 session target ipv4:10.148.13.13
 req-qos guaranteed-delay
 acc-qos controlled-load
 codec g723ar63
 ip qos dscp ef signalling

```

5.5 进一步阅读资料

RFC 1122, *Requirements for Internet Hosts—Communication Layers*, by Robert Braden.

RFC 1349, *Type of Service in the Internet Protocol Suite*, by Philip Almquist.

RFC 2205, *Resource ReSerVation Protocol (RSVP) —Version 1 Functional Specification*, by Bob Braden, Lixia Zhang, Steve Berson, Shai Herzog, and Sugih Jamin.

RFC 2309, *Recommendations on Queue Management and Congestion Avoidance in the Internet*, by Craig Partridge, Larry Peterson, K. K. Ramakrishna, Scott Shaker, John Wroclawski, and Lixia Zhang.

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, by Kathleen Nichols, Steven Blake, Fred Baker, and David L. Black.

RFC 2475, *An Architecture for Differentiated Services*, by Steven Blake, David L. Black,

Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss.

RFC 2597, *Assured Forwarding PHB Group*, by Juha Heinanen, Fred Baker, Walter Weiss, and John Wroclawski.

RFC 2598, *An Expedited Forwarding PHB*, by Van Jacobson, Kathleen Nichols, and Kedarnath Poduri.

RFC 2697, *A Single Rate Three Color Marker*, by Juha Heinanen and Roch Guerin.

Douskalis, Bill. *Putting VoIP to Work, Softswitch Network Design and Testing*.

Douskalis, Bill. *IP Telephony*.

Huston, Geoff. *Internet Performance Survival Guide*.

Ibe, Oliver C. *Converged Network Architectures*.

第 6 章

服务质量——速率限制和对流量进行队列处理

前面的两章介绍了路由器的性能管理、设备的质量管理、ATM 服务质量 (QoS)、三层交换、压缩、使用集成服务的端对端的服务质量以及使用区分服务的标记流量优先级的方法。当你应用了这些服务质量方法后，接着需要考虑对每种特定的流量类型采用最有效的队列机制。每一个接口使用某种类型的队列；你决定使用的类型将取决于你的服务策略需要对流量的控制程度、链路的带宽和流量的传输质量需求。本章探讨了不同的队列方法和它们的应用程序，包括下面的：

- 先进先出队列；
- 基于权重的队列；
- 优先级队列。

当我们介绍完“基本的 4 种”队列类型后，本章探讨更高级流量整形、队列、限速和标记技术，例如下面的：

- 通用流量整形；
- 基于类别的加权公平队列；
- 基于类别的整形；
- 流量的限速；
- 低延迟队列；
- 设置 IP RTP 优先级；
- 使用承诺速率来强制流量策略。

6.1 最基础的：先进先出队列

宽大于 2Mbit/s 的所有接口上。或者换句话说，E1 大小或者大于它的接口。使用先进先出机制，数据包通过接口转发的顺序和它们通过接口接收的顺序是一样的。例如，图 6-1 显示了 3 个流量的会话，或者说流。会话 A 由 Telnet 的数据包组成，大概是 64 字节；会话 B 中的数据包来自网络应用程序，包的尺寸范围是 750~1020 字节，而来自会话 C 的数据包是 HTTP 的 web 流量的数据包，它大概是 1500 字节，当这 3 台工作站在网络利用率较低的时间段内发送数据包时，所有的 3 个会话都应当是成功的。但是，如果这 3 台工作站的会话发生在高网络利用率期间，那么会话 C 的流量将会散布在从 A 到 B 的小数据包之间，这对远程登录的流量将会产生较大的抖动。

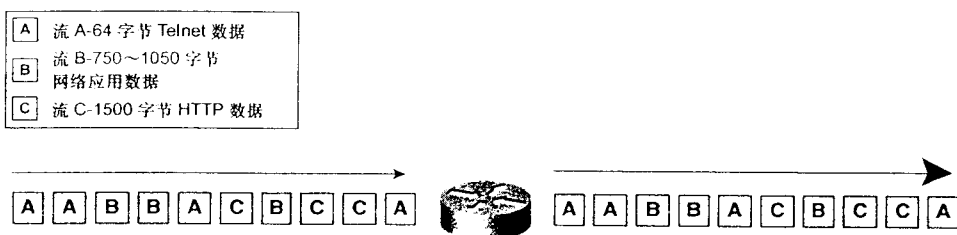


图 6-1 先进先出队列

在许多情况下，当网络应用程序的流量在线路的接口速率范围内时，通常运行先进先出队列是没有任何问题的。当接口开始遇到高拥塞时，或者遭遇到大量的大尺寸的数据包时，在这种情况下先进先出队列对于使用小数据包的协议或者对网络延迟比较敏感的应用程序就会产生问题。实时应用程序（例如语音和视频应用程序）对串行化延迟非常敏感，它指的是一个接口串行化数据包所需要的时间；这些应用通常在低速链路上和其他流量穿插在一起，运行的状态不是很好。

因此，当一个接口连续遇到或者超过它的带宽范围的流量时，或者在一个网络环境里经常遇到流量突发的情况时，就需要一个更高级的队列技术。

6.2 加权公平队列

基于 min-max 的公平共享算法，加权公平队列 (WFQ) 对于带宽低于 E1 速率的接口是默认的队列方法。

min-max 公平共享算法是基于轮循队列系统地按需要分配资源的。使用 min-max 公平共享算法，小的数据包在大的数据包之前优先进行传输。等待传输的数据包会在队列中排队，这是基于一个公式，即可用资源的带宽除以在队列中等待的数据包的个数。

$$\text{公平分配} = \frac{(\text{资源能力} - \text{已被分配的资源})}{\text{数据包的数量}}$$

思科加权公平队列算法和 min-max 的公平共享算法的不同之处是加权公平队列是基于 IP 报头中的 IP 优先级位来衡量数据的权重。加权公平队列算法试图使用这个信息，通过衡量数据包的尺寸和数据包的优先级在大包和小包之间公平地分担网络的负荷。对于一个 IP 优先级值为 0 的数据包，也就是默认的 routine precedence，权重是根据下面的公式

来计算的：

权重 = $\frac{32768}{\text{IP 优先级} + 1}$

表 6-1 显示了基于 IP 优先级值而产生的权重值。

表 6-1 权重表

IP 优先级的值	权重	IP 优先级的值	权重
0	32 768	4	6554
1	16 384	5	5461
2	10 923	6	4681
3	8192	7	4096

注意：在思科 IOS 的早期版本中，如 IOS 版本 12.0（5）T 之前，权重实际上是以不同的方法计算的。为了兼容旧版的思科 IOS 软件，将 32768 的值替换为 4096，如下所示：

权重 = $4096 \times (\text{IP 优先级} + 1)$

当工作站使用源地址和目的地址、IP 协议和 TCP 或者 UDP 端口号码，这被称为一个流。加权公平队列使用两种流类型：活动流，它是活动的会话，数据包等待被传输；非活动的流，这是以前从未看到过的新的会话或者是完整会话的空闲流。在加权公平队列的过程中，当新的数据包到达时，要特别注意数据包的尺寸。如果它们所属的 IP 流是新的，那么一个 rounded 的数据包尺寸也被使用。总之，数据包的尺寸、rounded 的数据包尺寸和 IP 优先级字段的值都用于产生序列号。序列号较低的数据包会被优先传输。一旦权重找到后，就会对队列中的每一个数据包产生序列号。注意一个流的 IP 优先级值只对一个流的第一个数据包起作用，随后的数据包使用第一个数据包的权重。

非活动流的序列号	$SN = (P \times W) + R$
活动流的序列号	$SN = W + RN$

SN = 序列号
P = 数据包尺寸（字节）
W = 权重
R = Rounded 数据包尺寸
RN = 活动流中最后一个数据包的序列号

图 6-2 显示了从不同流来的数据包是如何放在队列中并且使用加权公平队列转发的。在这个范例中，有来自 4 个会话的数据流：会话 A，有两个 1024 字节的数据包，它们的 IP 优先级值为 1，标记为 A1 和 A2；会话 B，有 3 个 64 字节的数据包，它们具有默认的 IP 优先级值 0；会话 C，有 4 个 64 字节的数据包，它们的 IP 优先级值为 5；会话 D，有一个 768 字节的数据包，IP 优先级的值为 0。数据包按照图右边的顺序到达加权公平队列的路由器：C-1，A-1，B-1，B-2，C-2，C-3，C-4，A-2，B-3 和 D-1。因为数据包 C-1 首先到达加权公平队列路由器，它是第一个产生序列号的数据包。数据包 C-1 被分配的序列号为 35 010，按照范例 6-1 所示，采用了非活动流的公式。

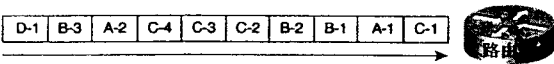


图 6-2 加权公平队列图

范例 6-1 加权公平队列和 C 数据包的数学

```
Packets C-1 is 64 bytes with IP Precedence = 5
```

```
Weight = 32768/5+1
```

```
Weight = 5461
```

```
SN = (64 x 5461) + 60
```

```
SN = 349504 + 60
```

```
SN = 349564
```

```
Packet C-2 is 64 bytes
```

```
SN = 5461 + 349564
```

```
SN = 355025
```

```
Packet C-3 is 64 bytes:
```

```
SN = 5461 + 355025
```

```
SN = 360486
```

```
Packet C-4 is 64 bytes:
```

```
SN = 5461 + 360486
```

```
SN = 365947
```

在这个范例中，数据包 C-1 是一个 IP 优先级值为 5 的 64 字节的数据包，被分配的权重为 5461。在这个范例中的权重是通过公式 $\text{权重} = 32768 / (\text{优先级} + 1)$ 来计算的，而序列号是使用 $\text{SN} = (\text{P} \times \text{W}) + \text{R}$ 公式对本章早期提到的非活动流来计算的。C 会话的任何新到的数据包将使用 $\text{SN} = \text{W} + \text{RN}$ 公式来计算活动流的序列号。数据包 C-2、C-3 和 C-4 的序列号使用刚才所提到的活动流的公式计算。下一个数据包，也就是 C-2，使用的是来自数据包 C-1 的权重和序列号，即 $\text{W} = 5461$ 和 $\text{RN} = 349564$ ，对数据包 C-2 产生一个新的序列号 355025。范例 6-2 显示了对于数据包 A-1 和 A-2 序列号是如何产生的。

范例 6-2 对于数据包 A-1 和 A-2 计算序列号

```
Packet A-1 is 1024 bytes with IP Precedence = 0
```

```
Weight = 32768/0+1
```

```
Weight = 32768
```

```
SN = (1024 x 32768) + 1000
```

```
SN = 33554432 + 1000
```

```
SN = 33555432
```

```
Packet A-2 is 1024:
```

```
SN = 32768 + 33555432
```

```
SN = 33588200
```

因为 A 会话是一个新的流，加权公平队列路由器使用非活动流的公式来计算数据包 A-1 的序列号，它产生的权重为 32768，序列号为 33555432。来自数据包 A-1 的权重和序列号用于帮助发现数据包 A-2 的序列号，使用活动流的公式，即 $\text{SN} = \text{W} + \text{RN}$ ，或者 $32768 + 33555432 = 33588200$ 。数据包 B-1 是一个新的流，使用的是非活动流的公式，数据包 B-2 和 B-3 使用的是范例 6-3 所示的活动流的计算公式。

范例 6-3 数据包 B-1、B-2 和 B-3 的序列号

```
Packets B-1 is 64 bytes with IP Precedence = 0

    Weight = 32768
    SN = (64 x 32768) + 60
    SN = 2097152 + 60
    SN = 2097212

Packet B-2 is 64 bytes

    SN = 32768 + 2097212
    SN = 2129980
Packet B-3 is 64 bytes:

    SN = 32768 + 2129980
    SN = 2162748
```

数据包 D 的序列号如范例 6-4 所示。

范例 6-4 数据包 D-1 的序列号

```
Packet D-1 is 768 bytes with IP Precedence = 0

    Weight = 32768
    SN = (768 x 32768) + 700
    SN = 25165824 + 700
    SN = 25166524
```

当最后一些数据包的信息被收集起来后，就得到了如表 6-2 所示的结果。

表 6-2 数据包传送的顺序

数据包的名字	序列号	数据包的名字	序列号
C-1	349 564	B-2	2 129 980
C-2	355 025	B-3	2 162 748
C-3	360 486	D-1	25 166 524
C-4	365 947	A-1	33 555 432
B-1	2 097 212	A-2	33 588 200

前面表中的序列号适用于在加权公平队列接口中每一个等待传输的数据包，并且数据包会按照最小到最大的序列号进行传输，如图 6-3 所示。具有较高优先级的较小的数据包和小序列号的数据包会优先进行传输，而具有 routine 优先级的较大的数据包和大序列号的数据包必须等待小数据包传输之后才能进行传输。加权公平队列在下面的环境中非常有用，即会话是由小数据包组成或者数据包具有较高的 IP 优先级，它们需要实时的传输速度（例如 Telnet 的数据包）。



图 6-3 加权公平队列数据包的传输顺序

就像以前所提到的，加权公平队列是速率为 E1 或者更小速率的接口的默认的队列方式。如果加权公平队列被关闭了，可以很容易地通过使用 **fair-queue** 命令来启用它。表 6-3 显示了 **fair-queue** 命令的参数和它们的描述。

fair-queue [congestive-discard-threshold] [dynamic-queues] [reservable-queues]

表 6-3 fair-queue 命令参数

参数	描述
<i>congestive-discard-threshold</i>	(可选) 在每一个队列中允许的数据包的数量 范围为 1~4096 默认的拥塞丢弃极限为 64
<i>dynamic-queues</i>	(可选) 可以建立的动态队列的数量。范围为 0~4096。从 16 开始，以 2 的倍数增长，即 (16, 32, 64, 128, 256, 512, 1024, 2048 和 4096) 默认的动态队列的数量为 256
<i>reservable-queues</i>	(可选) 当启用 RSVP 后，可以配置保留队列的数量 范围为 0~1000。默认的情况下，没有保留队列

为了使用默认的队列大小启用加权公平队列，可以输入 **fair-queue** 命令而不带任何参数，并且加权公平队列可以使用默认的队列大小和 256 个动态的队列启用。为了清除加权公平队列，将队列的方法修改为先进先出，输入 **no fair-queue**。为了查看在当前的接口上使用的队列方法，使用 **show interface** 命令。单独的队列值在早期的 6-3 表中显示，并且在范例 6-5 中着重注明。

范例 6-5 队列配置

```
Vacation# show interface serial 0/1
Serial0/1 is up, line protocol is up
Hardware is PowerQUICC Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:09, output 00:00:03, 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)
```

为了限制队列信息的显示，也可以使用 **show queueing interface** 命令，它显示了一个特定接口的队列信息。如范例 6-6 所示，这个命令显示了和 **show interface** 命令相同的队列信息。

范例 6-6 show queueing interface 命令

```
Vacation# show queueing interface serial 0/1
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)
```



```
3 input errors, 0 CRC, 3 frame, 0 overrun, 0 ignored, 0 abort
457 packets output, 31892 bytes, 0 underruns
0 output errors, 0 collisions, 7 interface resets
0 output buffer failures, 0 output buffers swapped out
2 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

注意：在改变队列尺寸之前，总是执行一个详细的流量分析并且测试配置，以避免导致生产性网络出现问题。

正如你在前一章中学习到的，加权公平队列需要运行其他的服务质量特性，例如 WRED 和资源预留协议（RSVP）。加权公平队列也是低延迟队列（LLQ）和基于类别的加权公平队列（CBWFQ）的基础，所以理解加权公平队列和流量分类和标记的技术是如何工作的非常重要。

6.3 优先级队列

当工作站调用一种队列机制来允许某些应用程序比其他应用程序具有更高的优先级，应当使用优先级队列（PQ）。优先级队列有 4 个队列，每一个队列都有不同的优先级。只有具有较高优先级的队列中的全部数据被传输完以后，才会转发较低优先级队列中的数据。使用优先级队列，有 4 个优先级的队列：高、中等、正常和低优先级队列。在每一个队列内，数据包是按照先进先出的顺序转发的。当使用优先级队列时，记住一些事情：

- 队列的尺寸并不一定影响从那个队列收到的数据包的转发时间。优先级队列的尺寸的限制可以配置。每一个队列按照优先级的顺序被服务。高优先级的队列总是优先服务，接着，如果高优先级队列中的数据被传完了，就会传输中等优先级队列中的数据。任何时候，只要高优先级队列中收到了新的数据包，就会服务这个队列，之后去处理其他队列中的数据。一旦中等优先级队列中的数据被传完了，如果这时候没有任何数据在高优先级的队列中，就会服务正常优先级队列中的数据。最终，如果高、中等和正常队列中的数据全部传输完了，就会服务低优先级队列中的数据。于是，就有这样的可能性，当使用优先级队列时，低优先级队列中的数据不能及时转发，导致网络应用程序超时。
- 如果数据包没有匹配任何配置的队列，那个数据包就会进入默认的队列，也就是正常的队列。可以修改默认的队列，在这章后面显示。
- 优先级队列不是动态的；它不会随着网络的形式调整。当使用优先级队列时，一个好的方法是周期性地执行网络基线校对并且分析流量，确保队列的尺寸和协议分发被正确地配置来处理峰值时刻的流量。

表 6-4 显示了 4 个优先级队列是如何服务的。

表 6-4 优先级队列

队列	描述
高	到达高优先级队列的数据包会被立刻服务。当高优先级队列没有数据包时，中等、正常和低优先级队列的数据包才会被服务。如果任何时候数据包又到达了高优先级队列，它们会在任何其他优先级队列被服务之前被转发，直到高优先级队列中的数据被服务完。默认的高优先级队列的尺寸是 20 个数据包

续表

队列	描述
中等	当高优先级的队列中没有数据包后，才会服务中等优先级队列的数据。如果中等优先级的队列正在转发数据时，又有任何数据包到达了高优先级队列，那么会优先转发高优先级队列的数据，直到整个队列中没有数据包后，才会继续服务中等优先级队列的数据。中等优先级队列的默认尺寸为 40 个数据包
正常	如果在高或者中等优先级队列中没有数据包，才会服务正常优先级队列。如果有数据包到达高或者中等优先级队列，它们就会按照从高到中等优先级队列的顺序进行转发，直到它们的队列中没有任何数据包时，才会服务正常优先级队列中的数据 正常优先级队列的默认尺寸是 60 个数据包 默认的情况下，所有未指定的流量会被分配到正常优先级的队列中，然而，可以使用 <code>default</code> 参数改变这个默认的行为
低	只有在其他优先级的队列中没有数据时，才会转发低优先级队列中的数据。如果其他队列中有数据的话，就会按照优先级的顺序进行转发，直到队列中没有数据时，才会服务低优先级队列中的数据。默认的低优先级队列的尺寸是 80 个数据包

图 6-4 显示了当优先级队列生效时，数据包是如何在队列中被服务的。

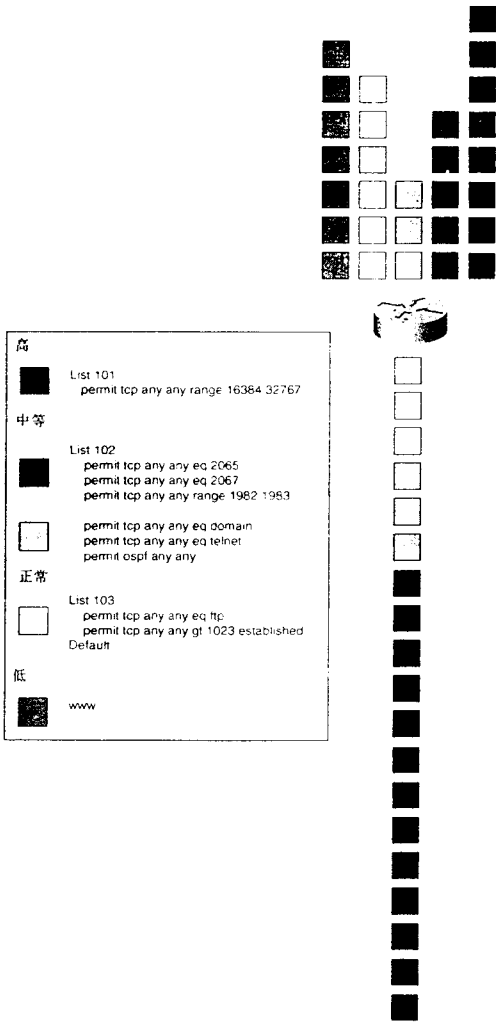


图 6-4 优先级队列图

为了配置优先级队列，可以使用 **priority-list** 命令来建立一个优先级列表。可以配置总共 16 个不同的优先级列表。每个优先级列表含有 4 个队列：高、中等、正常和低优先级队列。数据包会基于它们的特性：协议、进入的接口、数据包的尺寸和其他的因素被分配到 4 个队列中。不在 4 个队列中的任何流量会被分配到默认的队列，如果没有特别指定的话，就会分配到正常优先级的队列。表 6-5 显示了 **priority-list** 命令，它的参数、关键字和它们的描述。

表 6-5 priority-list 命令和描述

命令	参数	描述
Priority-list list-number default {high medium normal low}	None	定义了对于特定的优先级列表号码的默认队列。默认的队列指的是不匹配任何语句的数据包被发送的队列。如果没有特别指定，默认的队列是正常优先级队列
Priority-list list-number interface interface-number {high medium normal low}	None	指定了来自某个特定接口的流量被放到某个队列中，并进行优化传输
Priority-list list-number protocol argument	arp {high medium normal low}[gt frame-size lt frame-size] brigde {high medium normal low} [gt frame-size list access-list-number lt frame-size] bstun {high medium normal low} [address BSTUN-group-number hex-address gt frame-size lt frame-size] cdp {high medium normal low}[gt frame-size lt frame-size] compressedtcp {high medium normal low}[gt frame-size lt frame-size] dlsww {high medium normal low}[gt frame-size lt frame-size]	指定了 ARP 协议 指定了高、中等、正常或者低优先级队列 (可选)gt 指的是帧尺寸大于特定的 ARP 帧尺寸，范围从 0~65 535 (可选)lt 指的是帧尺寸小于特定的 ARP 帧尺寸，范围从 0~65 535 指定的是透明的桥协议 指定了高、中等、正常或者低优先级队列 (可选)gt 指定了大于特定帧尺寸的帧尺寸，范围为 0~65 535 (可选)list 指定了一个相关的访问控制列表 (200-299) 米用于流量分配 (可选)lt 指定了小于特定帧尺寸的帧尺寸，范围为 0~65 535 指定 Block Serial Tunnel (BSTUN) 协议 指定高、中等、正常或者低队列 (可选)address 指定一个特定的 BSTUN 的组号，范围为 1~255，并且地址以十六进制表示 (可选)gt 指定一个帧尺寸大于特定的 BSTUN 的帧尺寸，范围从 0~655 35 (可选)lt 指定一个帧尺寸小于特定的 BSTUN 的帧尺寸，范围从 0~655 35 指定一个思科发现协议 (CDP) 指定高、中等、正常或者低队列 (可选)gt 指定一个帧尺寸大于特定的 CDP 的帧尺寸，范围从 0~655 35 (可选)lt 指定一个帧尺寸小于特定的 CDP 的帧尺寸，范围从 0~655 35 指定一个压缩的 TCP 流量作为协议 指定高、中等、正常或者低队列 (可选)gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~655 35 (可选)lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~655 35 指定 DLSww 作为协议 指定高、中等、正常或者低队列 (可选)gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~655 35 (可选)lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~655 35

续表

命令	参数	描述
Priority-list list-number protocol argument (继续)	ip {high medium normal low} [fragments gt frame-size list access-list-number lt frame-size tcp port-number udp port-number]	指定 tcp/ip 作为协议 指定高、中等、正常或者低队列 (可选) fragment 指定对于 IP 包碎片的优化，也就是说，IP 包的 Fragment Offset 字段设为 1 (可选) gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~65535 (可选) list 指定一个相关的访问控制列表(1-199)应当用于流量的划分 (可选) lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~65535 (可选) tcp 指定来自或者去往某个 tcp 端口的流量作为指定的流量 端口的范围为 0~65535 或者下面列表中的关键字： bgp, chargen, cmd, daytime, discard, domain, echo, exec, finger, ftp, ftp-data, gopher, hostname, ident, irc, klogin, kshell, login, lpd, nntp, pin-auto-rp, pop2, pop3, smtp, sunrpc, syslog, tacacs, talk, telnet, time, uucp, whois 和 www (可选) udp 指定来自或者去往某个 udp 端口的流量作为指定的流量 端口的范围为 0~65535 或者下面列表中的关键字： biff, bootpc, bootps, discard, dnsix, domain, echo, isakmp, mobile-ip, nameserver, netbios-dgm, netbios-ns, netbios-ss, ntp, pim-auto-rp, rip, snmp, snmptrap, sunrpc, syslog, tacacs, talk, tftp, time, who 或者 xmcp
	ipx {high medium normal low}[gt frame-size][list list-number][lt frame-size]	指定 IPX 作为协议 (可选) gt 指定一个帧尺寸大于特定的 IPX 帧尺寸，范围从 0~65535 (可选) list 指定一个 IPX 的标准或者扩展的访问控制列表(800-899) (可选) lt 指定一个帧尺寸小于特定的 IPX 帧尺寸，范围从 0~65535
	llc2 {high medium normal low}[gt frame-size]][lt frame-size]	指定逻辑链路控制，类型-2 (LLC2) 协议 指定高、中等、正常或者低队列 (可选) gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~65535 (可选) lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~65535
	pad {high medium normal low}[gt frame-size]][lt frame-size]	指定 X.25 包组装/拆分 (PAD) 协议 指定高、中等、正常或者低队列 (可选) gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~65535 (可选) lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~65535
	qllc {high medium normal low}[gt frame-size]][lt frame-size]	指定合格逻辑链路控制 (QLLC) 协议 指定高、中等、正常或者低队列 (可选) gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~65535 (可选) lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~65535
	rsrb {high medium normal low}[gt frame-size]][lt frame-size]	指定远程源路由桥接 (RSTB) 协议 指定高、中等、正常或者低队列 (可选) gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~65535 (可选) lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~65535

续表

命令	参数	描述
Priority-list list-number protocol argument (继续)	snapshot {high medium normal low} [gt frame-size][lt frame-size] stun {high medium normal low} [address STUN-group STUN-address] gt frame-size lt frame-size}	指定了快照路由协议 指定了高、中等、正常或者低队列 (可选) gt 指定了大于特定帧尺寸的帧尺寸，范围为 0~65 535 (可选) lt 指定了小于特定帧尺寸的帧尺寸，范围为 0~65 535 指定了串行隧道 (STUN) 协议 指定了高、中等、正常或者低队列 (可选) address 指定了 STUN 的组号，范围为 0~255，以及一个十六进制的 STUN 地址，它必须以十六进制的格式书写 (例如，0x01) (可选) gt 指定了大于特定帧尺寸的帧尺寸，范围为 0~65 535 (可选) lt 指定了小于特定帧尺寸的帧尺寸，范围为 0~65 535
Priority-list list-number queue-limit	high-queue-limit medium-queue-limit normal-queue-limit low-queue-limit	对于优先级队列列表号码，改变每一种优先级队列的单独的队列限制 (高、中等、正常和低优先级的队列)

就像刚才所显示的，优先级队列允许用户以不同的方法来分类流量。

- **Protocol type** (协议类型) ——这包括主要的协议类型，例如 IP 或者 IPX，以及任何子协议信息，例如 TCP 或者 UDP 的端口号码。
- **Interface** (接口) ——流量所来自的接口。
- **Packet size** (数据包的尺寸) ——数据包的尺寸，或者大于或者小于一个特定的值，包括 MAC 封装，以字节表示。
- **Fragments** (碎片) ——被分成碎片的数据包。
- **Multiple criteria** (多个条件) ——使用访问控制列表来定义多个流量属性。

优先级队列配置需要 3 个步骤：定义队列分配、定制队列的配置并且将配置绑定到一个接口上。

第 1 步 定义队列。使用 **priority-list** 命令，对 4 个队列中的每个队列指定协议属性或者接口。在这个范例中，**access-list 188** 定义了 GRE 和 NTP 数据包，这些数据包被分配到高优先级的队列中，Telnet 数据包被分配到中等优先级的队列中，SMTP 数据包被分配到正常优先级的队列中，而 HTTP 的 web 数据包被认为具有低优先级并被发送到低优先级的队列中，可以使用 **priority-list** 命令实现。

```
access-list 188 permit gre any any
access-list 188 permit udp any any eq ntp
priority-list 1 protocol ip high list 188
priority-list 1 protocol ip medium tcp telnet
priority-list 1 protocol ip normal tcp smtp
priority-list 1 protocol ip low tcp www
```

第 2 步 定制队列的配置。对未分配的数据包配置默认的队列。如果默认队列没有显式地定义，所有未分配的数据包都会分配到正常优先级的队列中去。

```
Bart(config)# priority-list 7 default medium
```

可选地，可以对 4 个队列修改尺寸。**queue-limit** 命令允许用户对每一个队列定

义数据包的尺寸，可以使用 **priority-list list-number queue-limit high-limit medium-limit normal-limit low-limit** 命令。

```
Bart(config)# priority-list 7 queue-limit 40 20 30 20
```

第3步 给接口分配优先级列表。对于优先级队列可以不配置隧道和子接口。

```
interface Serial0/1
 ip address 10.2.1.1 255.255.255.0
 priority-group 7
```

为了查看优先级队列的配置，使用 **show interface** 命令。

```
Queueing strategy: priority-list 7
Output queue (queue priority: size/max/drops):
 high: 34/40/54, medium: 0/20/0, normal: 0/30/0, low: 0/20/0
```

范例 6-9 显示了如何使用优先级队列来使语音流量得到最高的优先级。数据链路层交换 (DLSw)、域名系统 (DNS)、Telnet (远程登录) 和开放最短路径优先 (OSPF) 流量被送到中等优先级的队列中去，而 FTP 和其他未指定的流量将被送到正常优先级的队列中去。在这个范例中，World Wide Web 流量具有低优先级。范例 6-10 显示了使用 **show queueing priority** 命令所得到的配置数据。

范例 6-9 优先级队列的行为

```
interface Serial0/1
 ip address 158.42.18.12 255.255.255.0
 priority-group 1
!
access-list 101 remark High Priority Queue - voice traffic
access-list 101 permit udp any any range 16384 32767
access-list 101 permit tcp any any eq 1720
access-list 102 remark Medium Priority Queue - DLSw, DNS, Telnet, OSPF
access-list 102 permit tcp any any eq 2065
access-list 102 permit tcp any any eq 2067
access-list 102 permit tcp any any range 1981 1983
access-list 102 permit tcp any any eq domain
access-list 102 permit tcp any any eq telnet
access-list 102 permit ospf any any
access-list 103 remark Normal Priority Queue - FTP and established
access-list 103 permit tcp any any eq ftp
access-list 103 permit tcp any any gt 1023 established
priority-list 1 protocol ip high list 101
priority-list 1 protocol ip medium list 102
priority-list 1 protocol ip normal list 103
priority-list 1 protocol ip low tcp www
```

范例 6-10 显示优先级队列的配置数据

```
Bart# show queueing priority
Current DLCI priority queue configuration:
Current priority queue configuration:
List Queue Args
1 high protocol ip list 101
1 medium protocol ip list 102
1 normal protocol ip list 103
1 low protocol ip tcp port www
```

当应用这个配置并等待数据传输以后，范例 6-11 显示了高优先级的队列当前有 34 个数据包在队列中，最大的队列尺寸是 20 个数据包，并且高优先级队列已经丢弃了 54 个数据包。然而，中等、正常和低优先级队列是空的，还没有丢弃任何数据包。在这种情况下，你可能想获取更进一步的数据分析来重新调整队列的尺寸，从而对数据分发进行更平均的分配。

范例 6-11 使用测试流量来显示优先级队列

```
Bart# show interfaces serial 0/1
Serial0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 158.42.18.12/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: priority-list 1
  Output queue (queue priority: size/max/drops):
    high: 34/20/54, medium: 0/40/0, normal: 0/60/0, low: 0/80/0
  5 minute input rate 139000 bits/sec, 7 packets/sec
  5 minute output rate 308000 bits/sec, 33 packets/sec
    4 packets input, 240 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
  228 packets output, 341544 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

范例：应用优先级队列

这个实验在真实的网络模型中使用 Windows PC 和其他的 Windows 服务器来测试优先级队列。为了测试优先级队列的配置，可以通过在工作站上建立动态主机配置协议（DHCP）、Microsoft Windows Internet Naming Service（WINS）和 DNS 服务，使得 PC 和服务器能够发送和接收传统的 TCP/IP 网络消息，在客户工作站和服务器之间的路由器能够对流量实施队列管理。

一、实验练习

这个实验需要两台路由器，每一台路由器具有一个以太网接口或者令牌环接口，以及一个串行接口。这两台路由器应当按照图 6-5 所示进行配置和布线。这个实验也含有两个终端工作站：一个运行 FTP 服务器的 Microsoft Window 服务器，提供 WINS 和 DNS 服务；一个被配置使用 DNS 和 WINS 服务的 Windows 客户 PC 机。为了验证队列的配置，需要 Windows PC 和服务器。如果你没有一个 PC 或者服务器的话，还是可以完成这个实验的部分配置。然而，没有流量产生的软件，队列的尺寸不可能增加。

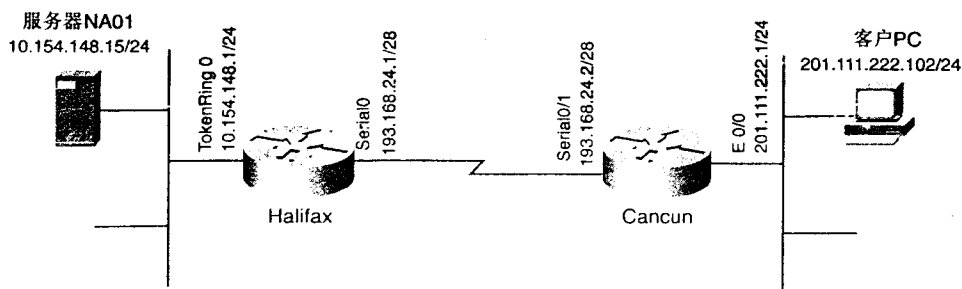


图 6-5 北美网络

二、实验目的

在本实验中，可以学到下面这些内容：

- 如何配置优先级队列。
- 如何测试优先级队列的配置。

三、需要的设备

- 对于本实验，需要两台思科路由器，每一台路由器具有一个以太接口或者令牌环接口和一个串行接口。
- 一个运行 TCP/IP 协议并且安装了 Windows 软件的 PC。
- 一个运行 TCP/IP 并配置了 DNS、FTP 和 WINS 服务功能的安装了 Windows 服务器软件的 PC。

四、物理布局和预规划

- 按照图 6-5 所示，对路由器进行布线。
- 按照图 6-5 所示，将每一台 PC 连接到网络中（路由器的接口）。

五、实验的任务

第 1 步 按照图 6-5 所示，配置路由器。Halifax 路由器应当有一个以太接口或者令牌环接口连接到服务器上，并且应当通过它的串行接口连接到 Cancun 路由器上。IP 地址应当按照前面的图 6-5 所示进行分配。

第 2 步 配置一个 Windows 服务器提供 DNS、WINS 和 FTP 服务。这个服务器应当被配置使用静态 IP 地址 10.154.148.15/24。FTP 的客户应当通过消极 FTP 会话(Passive FTP Session) 连接到 FTP 的服务器上。以后，一个 Windows 客户 PC 机将配置使用来自服务器的 WINS 和 DNS 服务。可以在 MS-DOS 提示符下使用 `ipconfig/all` 命令来验证客户机和服务器上的 TCP/IP 服务。

第 3 步 不是在 Windows 客户 PC 机上配置静态的 IP 地址、静态的 DNS 和 WINS 服务器，相反，配置 Cancun 路由器使用 DHCP 协议来提供信息。使用下面的这些值来配置 DHCP：

DHCP 范围：201.111.222.0/24

默认网关: 201.111.222.1

DHS 服务器: 10.154.148.15

域名: cciepsv2.net

WINS 服务器: 10.154.148.15

第 4 步 配置优先级队列和任何访问控制列表来支持表 6-6 所示的协议。

表 6-6 优先级队列的配置

队列	协议
High	DNS WINS
Medium	Windows NetBIOS 支持 NetBIOS 会话、数据包和名字服务，以及 DNS 和 WINS 管理 SNMP
Normal	消极模式的 FTP
Low	World Wide Web HTTP 流量 所有未指定的流量

第 5 步 将优先级队列进程分配到接口上，它对所有跨过 Cancun 和 Halifax 路由器之间的 WAN 连接的客户机流量进行队列管理（记住对于不同的接口类型的有效的流量队列的规则）。

第 6 步 验证客户和服务计算机彼此能够 ping 通对方。使用一种消极的 FTP 会话，从客户 PC 上复制文件到 ServerNA01。尽量使用 FTP 从服务器上获取其他的文件。当复制这些文件时，使用 **show interface** 输出来查看队列的信息。

六、实验的步骤

第 1 步 按照图 6-5 所示配置路由器。Halifax 路由器应当有一个以太接口连接到服务器上，并且有一个串行接口连接到 Cancun 路由器上。IP 地址应当按照前面的图 6-5 所示进行分配。

第 2 步 配置一个 Windows 服务器提供 DNS、WINS 和 FTP 服务。这个服务器应当被配置使用静态 IP 地址 10.154.148.15/24。FTP 的客户应当通过消极 FTP 会话连接到 FTP 的服务器上。以后，一个 Windows 客户 PC 机将配置使用来自服务器的 WINS 和 DNS 服务。可以在 MS-DOS 提示符下使用 **ipconfig /all** 命令来验证客户机和服务器上的 TCP/IP 服务。

范例 6-12 显示了在服务器和客户 PC 上使用 **ipconfig /all** 命令的输出。如果在服务器和客户机之间有任何连通性的问题，记住要验证每一个计算机的默认网关都配置为路由器的以太接口。还要验证每一个计算机能够 ping 通它的默认网关、沿途的每一跳路由器，最终到服务器。

注意：在 Windows 95 中，不存在 **ipconfig** 命令。为了验证 Windows 95 上的 TCP/IP 配置，在开始菜单的运行下使用 **winipcfg.exe** 命令。如图 6-6 所示，**winipcfg.exe** 是一个图形程序，显示和 **ipconfig** 在命令提示符下相同的信息。

第 3 步 不是在 Windows 客户 PC 机上配置静态的 IP 地址、静态的 DNS 和 WINS 服务器，相反，配置 Cancun 路由器使用 DHCP 协议来提供信息。使用下面的这些值来配置 DHCP：

范例 6-12 Windows 服务器和客户机的 TCP/IP 配置

```
The Server
C:\>ipconfig /all
Windows 2000 IP Configuration
    Host Name . . . . . : ServerNA01
    Primary DNS Suffix . . . . . : cciepsv2.net
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix . :
    Description . . . . . : FEM656C-3Com Global 8-100+56K CardB
us PC Card-(Fast Ethernet) #2
    Physical Address. . . . . : 00-50-DA-AC-5D-4C
    DHCP Enabled. . . . . : No
    IP Address. . . . . : 10.154.148.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.154.148.1
    DNS Servers . . . . . : 10.154.148.15
    Primary WINS Server . . . . . : 10.154.148.15

The Client
C:\>ipconfig /all
Windows 98 IP Configuration
    Host Name . . . . . : clientpc.cciepsv2.net
    DNS Servers . . . . . : 10.154.148.15
    Node Type . . . . . : Hybrid
    NetBIOS Scope ID. . . . . :
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    NetBIOS Resolution Uses DNS : Yes
0 Ethernet adapter :
    Description . . . . . : Xircom Ethernet 10/100 + Modem 56 PC Card
    Physical Address. . . . . : 00-80-C7-1D-12-A7
    DHCP Enabled. . . . . : Yes
    IP Address. . . . . : 201.111.222.102
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 201.111.222.1
    DHCP Server . . . . . : 201.111.222.1
    Primary WINS Server . . . . : 10.154.148.15
    Secondary WINS Server . . . :
    Lease Obtained. . . . . : 01 07 02 7:23:30 PM
    Lease Expires . . . . . : 01 08 02 7:23:30 PM
```

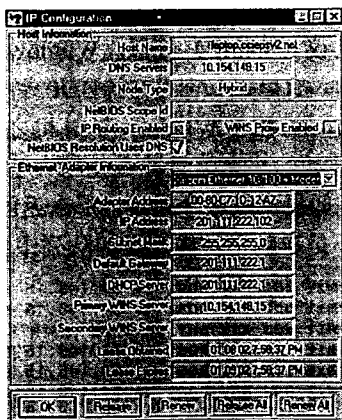


图 6-6 winipcfg.exe 程序

DHCP 范围: 201.111.222.0/24

默认网关: 201.111.222.1

DNS 服务器: 10.154.148.15

域名: cciepsv2.net

WINS 服务器: 10.154.148.15

为了对客户机配置 DHCP，在 Cancun 路由器上需要配置下面这些步骤：

(a) 建立一个 DHCP 地址池。在这个范例中，建立 client-pcs 池。

```
ip dhcp pool client-pcs
```

(b) 给这个 DHCP 地址池分配网段、默认的路由器、DNS 服务器、WINS 服务器和域名。

```
network 201.111.222.0 255.255.255.0
default-router 201.111.222.1
dns-server 10.154.148.15
domain-name cciepsv2.net
netbios-name-server 10.154.148.15
```

(c) 使用 **exclude-address** 命令来节省你不想用在 DHCP 中的地址。在这个范例中，范围为 201.111.222.1 到 100 的地址从 DHCP 中排除。

范例 6-13 显示了对 Cancun 路由器的 DHCP 配置。

```
ip dhcp excluded address 201.111.222.1 201.111.222.100
```

范例 6-13 Cancun 路由器的 DHCP 配置

```
ip dhcp excluded-address 201.111.222.1 201.111.222.100
!
ip dhcp pool laptops
network 201.111.222.0 255.255.255.0
default-router 201.111.222.1
dns-server 10.154.148.15
domain-name cciepsv2.net
netbios-name-server 10.154.148.15
```

第 4 步 配置优先级队列和任何访问控制列表来支持表 6-6 所示的协议。

为了配置优先级队列，正如表 6-6 所示，需要使用 3 个访问控制列表。access list 101 用于指定 DNS 和 WINS 流量。access list 102 用于指定 Windows NetBIOS 和简单网络管理协议 (SNMP) 流量。Windows 把 TCP 端口 135 用于 DNS 和 WINS 的管理流量，TCP 端口号 139，UDP 端口号 137 和 138，或者是关键字 **netbios-ns** 和 **netbios-ss** 用于 Windows 计算机之间的 NetBIOS 流量。最终，access list 103 用于指定被动的 FTP 流量并且使用大于 1023 的随机的 TCP 端口号作为 FTP 的文件传输。如果不做这个指定，那么返回的 FTP 流量就会被送到低优先级的队列中，而不是送到正常优先级的队列中去。

```
access-list 101 permit tcp any host 10.54.148.15 eq domain
access-list 101 permit udp any host 10.54.148.15 netbios-ns
access-list 101 permit udp any any eq snmp
access-list 102 permit tcp any host 10.54.148.15 eq 135
access-list 102 permit udp any host 10.54.148.15 eq netbios-ns
access-list 102 permit udp any host 10.54.148.15 eq netbios-ss
access-list 102 permit tcp any host 10.54.148.15 eq 139
access-list 103 permit tcp any host 10.54.148.15 eq ftp
access-list 103 permit tcp any host 10.54.148.15 gt 1023 established
```

访问控制列表的号码用于 **priority-list** 命令中来建立 4 个优先级的队列，并且 **default** 关键字用于将未指定的流量发送到低优先级的队列中去。

```
priority-list 10 protocol ip high list 101
priority-list 10 protocol ip medium list 102
priority-list 10 protocol ip normal list 103
priority-list 10 protocol ip low
priority-list 10 default low
```

第 5 步 将优先级队列进程分配到接口上，它对所有跨过 Cancun 和 Halifax 路由器之间的 WAN 连接的客户机流量进行队列管理（记住对于不同的接口类型的有效的流量队列的规则）。

优先级队列进程通过使用 **priority-group** 命令分配到 Cancun 路由器的串行接口上。

```
interface Serial0/1
priority-group 10
```

第 6 步 验证客户和服务计算机彼此能够 ping 通对方。使用一种消极的 FTP 会话，从客户 PC 上复制文件到 ServerNA01。尽量使用 FTP 从服务器上获取其他的文件。当复制这些文件时，使用 **show interface** 输出来查看队列的信息。

在这个实验中，不同的流量类型、TFTP 文件拷贝、扩展 ping、数据包产生文件、Windows Explorer 中的文件拷贝和 web surfing 都尝试过了，这也是在范例 6-14 中所示的结果。

范例 6-14 查看 FTP 会话的队列信息

```
Cancun# show interfaces serial 0/1
Serial0/1 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 193.168.24.2/29
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 28/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:01, output 00:00:05, output hang never
Last clearing of "show interface" counters 00:03:56
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: priority-list 10
Output queue (queue priority: size/max/drops):
    high: 0/20/0, medium: 0/40/0, normal: 3/60/0, low: 0/80/0
5 minute input rate 7000 bits/sec, 10 packets/sec
5 minute output rate 174000 bits/sec, 18 packets/sec
2726 packets input, 156448 bytes, 0 no buffer
Received 28 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4983 packets output, 6970545 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

当你已经发送了某些测试流量并且验证了优先级队列的配置后，你已经完成了这个练习。范例 6-15 显示了关于 Halifax 和 Cancun 路由器的完整配置。

范例 6-15 练习实验中的完整配置

```
hostname Cancun
!
ip dhcp excluded-address 201.111.222.1 201.111.222.100
!
ip dhcp pool laptops
    network 201.111.222.0 255.255.255.0
    default-router 201.111.222.1
    dns-server 10.154.148.15
    domain-name cciepsv2.net
    netbios-name-server 10.154.148.15
!
interface Ethernet0/0
    ip address 201.111.222.1 255.255.255.0
!
interface Serial0/1
    ip address 193.168.24.2 255.255.255.248
    priority-group 10
    clockrate 1300000
!
router rip
    version 2
    network 193.168.24.0
    network 201.111.222.0
!
access-list 101 permit tcp any any host 10.54.148.15 eq domain
access-list 101 permit udp any any host 10.54.148.15 eq netbios-ns
access-list 101 permit udp any any eq snmp
access-list 102 permit tcp any host 10.54.148.15 any eq 135
access-list 102 permit udp any host 10.54.148.15 any eq netbios-ns
access-list 102 permit udp any host 10.54.148.15 any eq netbios-ss
access-list 102 permit tcp any host 10.54.148.15 any eq 139
access-list 103 permit tcp any host 10.54.148.15 any eq ftp
access-list 103 permit tcp any host 10.54.148.15 any gt 1023 established
priority-list 10 protocol ip high list 101
priority-list 10 protocol ip medium list 102
priority-list 10 protocol ip normal list 103
priority-list 10 protocol ip low
priority-list 10 default low

-----
hostname Halifax
!
interface Ethernet0
    ip address 10.154.148.1 255.255.255.0
!
interface Serial0
    ip address 193.168.24.1 255.255.255.248
!
router rip
    version 2
    network 10.0.0.0
    network 193.168.24.0
```

既然你已经看到优先级队列是如何工作的，你可能注意到你不想在网络中启用优先级队列的一个原因：在你必须将流量做队列处理时，低优先级队列中的数据可能会被饿死，而你并没有严格优先级队列的需求，有几种队列的机制你可以考虑作为严格优先级队列的替代方法。

6.4 定制队列

目前所讨论的每一种队列的方法都是对某一种优先级的流量实施了最优的数据转发。这些队列方法也不仅仅是静态配置的能力。加权公平队列仅允许用户控制队列的尺寸和数量，并不允许更多的定制，如果你想对多种流量进行排序，这可能会产生问题。优先级队列允许用户仅配置 4 个队列和每一个队列中所允许的数据包数量。优先级队列也有一个显著的缺点：低优先级的队列可能得不到足够的关注，因此，在某些情况下，取决于高优先级流量的大小，它们可能永远也得不到关注。定制队列（CQ）利用它高度可定制的配置属性解决了这些问题。

定制队列得名于它总共有 17 个队列，而其中 16 个可以由用户定义的流量类型来配置。第一个队列（队列 0）是系统队列，主要被思科 IOS 软件用于系统流量，这个队列用户不可配置。另外 16 个其他的队列中每一个都有队列尺寸的限制，要么以字节表示，要么以这个队列中可以包含多少个数据包来限制。任何一个队列都会被执行，直到字节的数量或者数据包的限制达到。如果其中的任何一种情况发生，当前数据包的转发就会完成，接着下一个队列会依次类推（达到字节数或者包的个数），以轮循的方式来确保每一个队列得到同等程度的关注，并且没有任何一个队列可以阻止其他的队列得不到这种关注。如果一个队列满了，任何新到达这个队列的数据包就会被丢掉。如果一个队列空了，它就会被忽略，而去服务下一个队列。定制队列的内容是由下面的因素决定的：

- 进入的接口（接收流量的接口）。
- 访问控制列表，定制队列支持所有的主要协议，包括 IP、IPX、AppleTalk 和 SNA 协议以及它们的访问控制列表。
- 数据包的尺寸，要么大于要么小于一个特定的尺寸。
- 或者由地址、端口号码或者由思科 IOS 软件参数所定义的特定于协议的特性。

例如，在图 6-7 中，你可以看到有 6 个队列。队列 1 被分配使用接口带宽的 50%，队列 2 分配使用接口带宽的 20%；队列 3 被分配使用接口带宽的 12%；队列 4 使用接口带宽的 5%；队列 5 使用接口带宽的 3%，剩下 10% 的带宽由队列 6 使用。带有箭头的线路代表队列被服务的顺序。每个队列按照它的字节数量或者数据包数量被转发后，就会轮到下一个队列。使用这种队列的机制，当队列 1 传输完它的所有的数据包后，就会轮到队列 2，3，4，5，6。当其他队列正在被服务时，又有新的数据包到达了许多队列，如图 6-8 所示。当一个队列的服务字节数达到后，就会服务下一个队列，也服务相应的字节数。当其中一个队列不含有任何数据包时，这个队列就会被忽略，如队列 4。当一个队列的数据包或者字节数量的限制达到后，任何到达这个队列的新数据包都会被丢弃。

在图 6-8 中，队列 2 是 100% 满了。当一个队列中数据包的总量达到了队列的限制或者队列以字节表示的尺寸时，我们就说这个队列已经满了。使用定制队列，当一个队列满了，队列中的最后一个数据包被传输后，才轮到下一个队列被服务。如果一个队列在等待服务的过程中满了，那么新到达这个队列的数据包就会被丢掉。

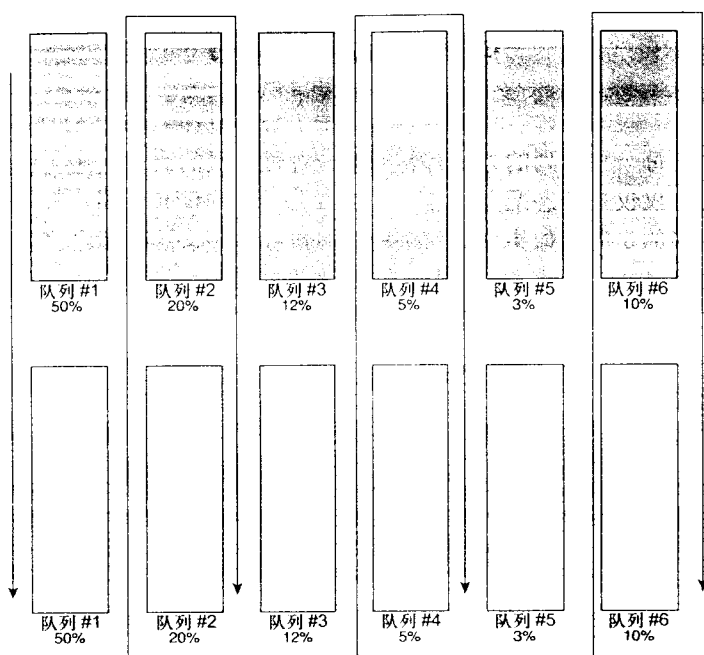


图 6-7 定制队列的图

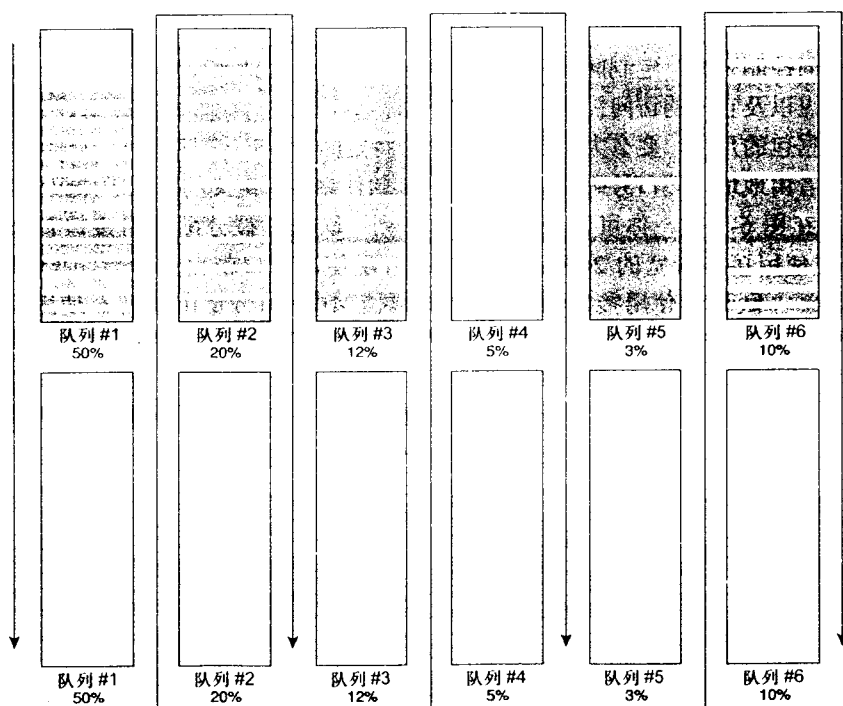


图 6-8 定制队列图表重显

注意：虽然在定制队列中有 17 个队列，只有 16 个是用户可以配置的。队列 0 是被操作系统使用来转发网络的控制流量的。当使用 **queue-list** 命令时，0 队列表现为可以配置，然而，除非是路由器产生流量，它不可以用于其他流量。

queue-list 命令定义了 16 个定制队列，在每台路由器上可以定义多达 16 个这样的定制队列的访问控制列表。表 6-7 显示了 **queue-list** 命令，它的参数和它们的描述。

表 6-7 定制队列的语法

命令	参数	描述
queue-list list-number default queue-number	没有	default 命令对没有特别指定某一个队列的流量定义默认队列 list-number 指定配置应用于哪个队列的列表。这个数字的范围为 1~16 queue-number 指定命令应用于 17 个队列中的哪一个。这个数字的范围为 0~16
queue-list list-number interface interface-name interface-number queue-number	没有	interface 命令用于指定来自某一个接收接口的所有的流量，即下面的接口名字和接口号码所定义的流量，被分配到 queue-number 参数所指定的队列中去
queue-list list-number lowest-custom queue-number	没有	lowest-custom 命令用于指定如果所有的 16 个队列都没有用于定制队列时，队列列表所使用的最低队列号码
queue-list list-number protocol protocol queue-number	协议的参数： arp gt frame-size lt frame-size brige[gt frame-size list list-number lt frame-size] bstun[address group-number hex-number gt frame-size lt frame-size] cdp[gt frame-size lt frame-size] Compressedtcp[gt frame-size lt frame-size]	protocol 命令指定所有来自下列协议的流量将被发送到指定的队列中去 arp 关键字用于指定 ARP 协议 (可选) gt 指定来自 ARP 协议的包尺寸大于特定的帧尺寸，范围从 0~655 35 (可选) lt 指定来自 ARP 协议的包尺寸小于特定的帧尺寸，范围从 0~655 35 brige 关键字用于指定那些透明桥接的流量。 (可选) gt 指定 dls+流量的数据包尺寸大于特定的帧尺寸，范围从 0~655 35 (可选) list 指定一个相关的访问控制列表 (200-299) 应当用于流量的划分 (可选) lt 指定一个 dls+流量的数据包尺寸小于特定的帧尺寸，范围从 0~655 35 bstun 关键字指定 BSTUN 作为协议 (可选) address 指定那些来自特定 BSTUN 组的流量，地址格式为 16 进制 BSTUN 组范围从 1~255 (可选) gt 指定一个帧尺寸大于特定帧尺寸的 BSTUN 流量，范围从 0~655 35 (可选) lt 指定一个帧尺寸小于特定帧尺寸的 BSTUN 流量，范围从 0~655 35 cdp 关键字用于指定 CDP 协议 (可选) gt 指定一个帧尺寸大于特定帧尺寸的 CDP 流量，范围从 0~655 35 (可选) lt 指定一个帧尺寸小于特定帧尺寸的 CDP 流量，范围从 0~655 35 compressedtcp 关键字用于指定压缩的 TCP 流量 (可选) gt 指定一个帧尺寸大于特定帧尺寸的 TCP 流量，范围从 0~655 35 (可选) lt 指定一个帧尺寸小于特定帧尺寸的 TCP 流量，范围从 0~655 35

续表

命令	参数	描述
queue-list list-number protocol queue-number (继续)	dlsw [gt frame-size lt frame-size] ip [fragments gt frame-size list list-number lt frame-size tcp tcp-protocol udp udp-protocol] llc2 [gt frame-size][lt frame-size] pad [gt frame-size][lt frame-size] qllc [gt frame-size][lt frame-size] rsrbr [gt frame-size][lt frame-size] snapshot [gt frame-size lt frame-size] stun [address group-number hex-number gt frame-size lt frame-size]	<p>dlsw 关键字用于指定 DLSw+的协议</p> <p>(可选) gt 指定 DLSw+流量的数据包尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定 DLSw+流量的数据包尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>ip 关键字指定 TCP/IP 作为协议</p> <p>指定高、中等、正常或者低队列</p> <p>(可选) fragment 指定只有被分成碎片的数据包的碎片才被匹配，不是第一个碎片</p> <p>(可选) gt 指定一个帧尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) list 指定一个相关的访问控制列表 (1-199 或者 1300-1399) 应当用于流量的划分</p> <p>(可选) lt 指定一个帧尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) tcp 指定来自或者去往某个 TCP 端口的流量作为指定的流量</p> <p>端口的范围为 0~655 35 或者下面列表中的关键字：</p> <p>bgp, chargen, cmd, daytime, discard, domain, echo, exec, finger, ftp, ftp-data, gopher, hostname, ident, irc, klogin, kshell, login, lpd, nntp, pin-auto-rp, pop2, pop3, smtp, sunrpc, syslog, tacacs, talk, telnet, time, uucp, whois 和 www</p> <p>(可选) udp 指定来自或者去往某个 UDP 端口的流量作为指定的流量</p> <p>端口的范围为 0~655 35 或者下面列表中的关键字：</p> <p>biff, bootpc, bootps, discard, dnssix, domain, echo, isakmp, mobile-ip, nameserver, netbios-dgm, netbios-ns, netbios-ss, ntp, pim-auto-rp, rip, snmp, snmptrap, sunrpc, syslog, tacacs, talk, tftp, time, who 或者 xdmcp</p> <p>llc2 关键字用于指定 LLC-2 协议</p> <p>(可选) gt 指定 LLC-2 的流量尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定 LLC-2 的流量尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>pad 关键字用于指定 PAD 协议</p> <p>(可选) gt 指定 PAD 流量的包尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定 PAD 流量的包尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>qllc 关键字用于指定 QLLC 协议</p> <p>(可选) gt 指定 QLLC 流量的包尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定 QLLC 流量的包尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>rsrb 关键字用于指定 RSRB 协议</p> <p>(可选) gt 指定 RSRB 流量的包尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定 RSRB 流量的包尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>snapshot 关键字用于指定快照路由的流量</p> <p>(可选) gt 指定快照路由流量的包尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定快照路由流量的包尺寸小于特定的帧尺寸，范围从 0~655 35</p> <p>stun 关键字用于指定 STUN 协议</p> <p>(可选) address 指定来自特定的 STUN 组和地址的流量是以十六进制的形式表示的。STUN 组的范围是从 1~255</p> <p>(可选) gt 指定 STUN 流量的包尺寸大于特定的帧尺寸，范围从 0~655 35</p> <p>(可选) lt 指定 STUN 流量包尺寸小于特定的帧尺寸，范围从 0~655 35</p>
queue-list list-number queue-number [byte-count byte-size] [limit queue-entries]	没有	<p>byte-count 选项用于为特定的队列指定字节数目限制，字节数量范围从 1~16 777 215</p> <p>limit 选项用于指定可能退出特定队列的条目数据限制，范围从 0~32 767</p> <p>这些选项的用法在本节后面讲述</p>
queue-list list-number stun queue-number address group-number hex-number	没有	<p>stun 选项用于指定属于 STUN 组、具备特定 16 进制地址的 STUN 流量</p> <p>STUN 组的范围从 1~255，16 进制地址必须以 0x 前缀作为开始</p>

定制队列配置需要 4 个步骤：使用访问控制列表来定义放到队列中的流量，通过建立队列并且将流量类型分配到队列建立队列列表，定制队列，并且将队列绑定到接口上。在下一个范例中，这些步骤用于配置一个样例网络。

第 1 步 通过将流量类型分配到队列中建立队列列表。

在这个范例中，如表 6-8 所示分发流量。

表 6-8 定制队列实验的协议分发

队列号	流量类型	队列号	流量类型
1	OSPF, SNMP	6	NFS
2	GRE	7	到 192.16.12.8 的被动模式 FTP, TFTP
3	DLSw-	8	WWW
4	DNS, SMTP 和 DHCP	9	其他
5	Windows NetBIOS 支持		

为了配置这 9 个队列，如范例 6-16 所示，使用 7 个 IP 访问控制列表。

范例 6-16 定制队列的访问控制列表

```
access-list 101 permit ospf any any
access-list 101 permit udp any any eq snmp
access-list 102 permit gre any any
access-list 103 remark DLSw 2065, 2067, 1981, 1982, and 1983
access-list 103 permit tcp any any eq 2065
access-list 103 permit tcp any any eq 2067
access-list 103 permit tcp any any eq 1981
access-list 103 permit tcp any any eq 1982
access-list 103 permit tcp any any eq 1983
access-list 104 permit tcp any any eq domain
access-list 104 permit tcp any any eq smtp
access-list 104 permit udp any any eq bootpc
access-list 105 permit tcp any any eq 139
access-list 105 permit udp any any eq netbios-dgm
access-list 105 permit udp any any eq netbios-ns
access-list 105 permit udp any any eq netbios-ss
access-list 106 permit tcp any any eq 2049
access-list 106 permit udp any any eq 2049
access-list 107 permit tcp any 192.16.12.8 eq ftp
access-list 107 permit tcp any 192.16.12.8 gt 1023 established
access-list 107 permit udp any any eq tftp
```

第 2 步 当建立访问控制列表以后，现在来配置队列分配。这是通过 **queue-list** 命令完成的，并且在 **queue-list** 命令中参考访问控制列表，如范例 6-17 所示。

范例 6-17 队列列表的配置

```
queue-list 3 protocol ip 1 list 101
queue-list 3 protocol ip 2 list 102
queue-list 3 protocol ip 3 list 103
queue-list 3 protocol ip 4 list 104
queue-list 3 protocol ip 5 list 105
queue-list 3 protocol ip 6 list 106
queue-list 3 protocol ip 7 list 107
queue-list 3 protocol ip 8 tcp www
```

第 3 步 接下来，定制队列的配置。在这个范例中，默认的 IP 流量应当被发送到队列 9。

```
queue-list 3 default 9
```

第 4 步 将队列机制分配到一个接口上。这是在接口配置模式中通过使用 **custom-queue-list** 命令完成的。当定制队列在一个接口上启用后，可以通过两种方法来验证它的配置：**show queueing** 命令和 **show interface** 命令。**show queueing** 命令显示了对路由器的当前队列的配置。如果超过一种队列类型正在使用，可以在 **show queueing** 命令中添加 **custom** 关键字来指定只显示定制队列的配置，如范例 6-18 所示。

```
interface Serial0/2
ip address 165.11.2.1 255.255.255.0
custom-queue-list 3
```

范例 6-18 验证定制队列的配置

```
FS_HQ# show queueing custom
Current custom queue configuration:
List  Queue  Args
3      9      default
3      1      protocol ip      list 101
3      2      protocol ip      list 102
3      3      protocol ip      list 103
3      4      protocol ip      list 104
3      5      protocol ip      list 105
3      6      protocol ip      list 106
3      7      protocol ip      list 107
3      8      protocol ip      tcp port www
3      9      protocol ip
```

为了看到队列的数据包的尺寸限制，使用 **show interface** 命令。范例 6-19 显示了定制队列 3 正在使用，16 个队列中每一个队列都被限制到 20 个数据包的大小，这也是默认的设置。

范例 6-19 对定制队列使用 show interface 命令

```
FS_HQ#sh int s0/2
Serial0/2 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 165.11.2.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 6/255, rxload 6/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:00, output 00:00:02, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: custom-list 3
Output queues: (queue #: size/max/drops)
    0: 0/20/0 1: 0/20/0 2: 0/20/0 3: 0/20/0 4: 0/20/0
    5: 0/20/0 6: 0/20/0 7: 0/20/0 8: 0/20/0 9: 0/20/0
    10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0
    15: 0/20/0 16: 0/20/0
5 minute input rate 41000 bits/sec, 4 packets/sec
5 minute output rate 41000 bits/sec, 4 packets/sec
```

(待续)

```
1087 packets input, 1437808 bytes, 0 no buffer
Received 53 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
1079 packets output, 1435130 bytes, 0 underruns
0 output errors, 0 collisions, 6 interface resets
0 output buffer failures, 0 output buffers swapped out
18 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

注意每一个队列都是使用当前队列的尺寸来显示的，每一个队列中最大的数据包的数量和每一个队列中被丢弃的数据包的数量。在先前的范例中，每一个队列当前都是空的，这是因为定制队列只有在接口上发生拥塞时，才会使用它，而在本例中，这个接口每秒钟传输的数据包还不到一个。

使用定制队列可以控制每一个队列的尺寸。在改变对每一个队列的带宽分配之前，首先要考虑一些事情。首先，当调整队列的尺寸时，考虑平均数据包的尺寸，以字节为单位限制队列的尺寸。如果你设置队列的尺寸为 2000 个字节，而你的平均数据包的大小为 1024 个字节，在这个范例中，每次这个队列被服务时只有两个数据包会从这个队列中发送出去。其次，如果你设置的数据包的尺寸太大了，带宽可能没有适当地分配，会导致对队列空间的浪费。因此，在给队列分配带宽之前，最好首先对平均数据包的尺寸进行分析，这是因为设置队列的尺寸太小了将导致非常规数据包的传送问题，而设置队列的尺寸太大了将使队列的尺寸空间不能充分利用或者导致某个协议会过度使用接口的带宽。

需要 9 个基本的步骤来定义分配到每一个队列的带宽的大小。**byte-count** 命令允许用户控制单独的队列的尺寸。**byte-count** 命令基本上用于基于流量的百分比来给某种流量类型分配带宽。在分配流量之前，定义每一种协议和队列的平均数据包的尺寸以及接口带宽的总量非常重要。接着定义每一种队列所需要的接口带宽的百分比。

例如，下面的步骤高度概括了一个简单的队列机制是如何建立的，通用路由协议 (GRE)、WWW 和被动模式的 FTP 都使用了在先前的范例中定义的相同的协议。

第 1 步 对每一个协议找到平均数据包的尺寸。表 6-9 显示了本范例中协议的平均数据包的尺寸。在这个范例中，也显示了队列所用到的带宽的分配：

平均数据包尺寸 (A)

以字节表示的总体流量 (B)

总体数据包的数量 (P)

$$A = B/P$$

表 6-9 协议的数据包尺寸

协议	带宽分配	平均数据包的尺寸	协议	带宽分配	平均数据包的尺寸
GRE	55	794	FTP	25	678
WWW	20	746			

第 2 步 发现数据包的比例来计算分配给队列的百分比。在开始带宽的分配进程之前首先要发现带宽的百分比，这个比例是通过将数据包的尺寸除以带宽的百分比得到的。表 6-10 显示了这个公式的结果：

流量比例 (R)

带宽的百分比 (B)

数据包的尺寸 (P)

$R = B/P$

$55/794 = 0.06926$
 $20/746 = 0.02680$
 $25/678 = 0.03687$

表 6-10 定制队列流量分配

协议	带宽分配	平均数据包的尺寸	比例
GRE	55	794	0.06926
WWW	20	746	0.02680
FTP	25	678	0.03687

第 3 步 对第 2 步中发现的比例进行正常化；这是通过将每一个比例和第 2 步中的最低比例进行除法运算得到的。表 6-11 显示了对这个范例的正常化的比例。

最低比例 (L)

比例 (R)

正常化号码 (N)

$N = R/L$

0.02680 is the lowest ratio
 $0.06926/0.02680 = 2.58$ rounded to 2.6
 $0.02680/0.02680 = 1$
 $0.03687/0.02680 = 1.38$ rounded to 1.4

表 6-11 定制队列正常化比例

协议	带宽分配	平均数据包的尺寸	比例	正常化比例
GRE	55	794	0.06926	2.6
WWW	20	746	0.02680	1
FTP	25	678	0.03687	1.4

第 4 步 将每一个比例换算成下一个最靠近的十进制整数。数据包的比例应当被换算成一个整数，这是因为定制队列传输队列中最后一个完整的数据包后才会移动到下一个队列中去。表 6-12 显示了这个范例中以整数表示的比例。

表 6-12 定制队列整体的比例

协议	带宽分配	平均数据包的尺寸	比例	正常化比例	整体比例
GRE	55	794	0.06926	2.6	3
WWW	20	746	0.02680	1	1
FTP	25	678	0.03687	1.4	2

第 5 步 为了将数据包的比例转换成字节的计数，这个比例必须和平均数据包的大小相乘。表 6-13 显示了字节的计数：

数据包的比例 (R)

平均数据包的大小 (P)

字节计数 (B)

$$B = R * P$$

$$\begin{aligned} 3 \times 794 &= 2382 \\ 1 \times 746 &= 746 \\ 2 \times 678 &= 1356 \end{aligned}$$

表 6-13 定制队列字节计数

协议	带宽分配	平均数据包的尺寸	比例	正常化的比例	整体的比例	字节的计数
GRE	55	794	0.06926	2.6	3	2382
WWW	20	746	0.02680	1	1	746
FTP	25	678	0.03687	1.4	2	1356

第 6 步 为了发现这个比例所代表的带宽分配，合并所有的队列所使用的整体带宽。

带宽分配 (D)

字节计数 (B)

$$D = B + B + B \text{ (each B)}$$

$$2382 + 746 + 1356 = 4484$$

第 7 步 为了找到每一个队列以字节表示的整体带宽的百分比，将每一个队列的字节计数除以总体带宽的分配。表 6-14 显示了这个范例的带宽的百分比。

带宽的百分比 (P)

带宽的分发 (D)

字节的计数 (B)

$$P = B/D$$

$$\begin{aligned} 4484 \\ 2382/4484 &= 53 \\ 746/4484 &= 17 \\ 1356/4484 &= 30 \end{aligned}$$

表 6-14 定制队列带宽的百分比

协议	带宽分配	平均数据包的尺寸	比例	正常化比例	整体比例	字节计数	带宽的百分比
GRE	55	794	0.06926	2.6	3	2382	53
WWW	20	746	0.02680	1	1	746	17
FTP	25	678	0.03687	1.4	2	1356	30

第 8 步 如果这个比例和原始的带宽分配的百分比不是足够接近的话，返回到第 3 步并且将这个比例和另外一个值相乘。在这个范例中，使用数字 2 和 3。注意数字 2 和所需的字节数最接近，3 超过了字节数。在这种情况下，我决定试一下 2.5，它和原始的带宽分配的百分比最接近。表 6-15 显示了最终的带宽百分比分配和字节的计数。

$$\begin{aligned} 2.6 \times 2 &= 5.2 \text{ rounded to } 6 \\ 1 \times 2 &= 2 \\ 1.4 \times 2 &= 2.8 \text{ rounded to } 3 \\ 6 \times 794 &= 4764/8288 = 58 \\ 2 \times 746 &= 1492/8288 = 18 \\ 3 \times 678 &= 2032/8288 = 25 \\ &\dots\dots\dots \\ &8288 \\ 2.6 \times 3 &= 7.8 \text{ rounded to } 8 \end{aligned}$$

```
1 x 3 = 3
1.4 x 3 = 4.2 rounded to 5
8 x 794 = 6352/11980 = 53
3 x 746 = 2238/11980 = 19
5 x 678 = 3390/11980 = 28
-----
11980
2.6 x 2.5 = 6.5 rounded to 7
1 x 2.5 = 2.5 rounded to 3
1.4 x 2.5 = 3.5 rounded to 4
7 x 794 = 5558/10508 = 53%
3 x 746 = 2238/10508 = 21%
4 x 678 = 2712/10508 = 26%
-----
10508
```

表 6-15

每个队列最终的带宽分配

协议	带宽的分配	平均数据包的尺寸	比例	正常化比例	整体比例	字节计数	带宽的百分比
GRE	55	794	0.06926	2.6	7	5558	53
WWW	20	746	0.02680	1	3	2238	21
FTP	25	678	0.03687	1.4	4	2712	26

第 9 步 一旦找到了字节的计数，将它们绑定到队列上，使用 **queue-list byte-count** 命令，如范例 6-20 所示。

范例 6-20 完成定制队列定制的字节计数的配置

```
interface Serial0/2
 ip address 165.11.2.1 255.255.255.0
 custom-queue-list 5
!
access-list 110 permit gre any any
access-list 120 permit tcp any any eq ftp
access-list 120 permit tcp any any gt 1023 established
queue-list 5 protocol ip 1 list 110
queue-list 5 protocol ip 2 list 120
queue-list 5 protocol ip 3 tcp www
queue-list 5 queue 1 byte-count 5558
queue-list 5 queue 2 byte-count 2238
queue-list 5 queue 3 byte-count 2712
FS_HQ# show queueing custom
Current custom queue configuration:
List Queue Args
5 1 protocol ip list 110
5 2 protocol ip list 120
5 3 protocol ip tcp port www
5 1 byte-count 5558
5 2 byte-count 2238
5 3 byte-count 2712
```

本章从讨论思科 IOS 软件中 4 种基本的队列类型开始，剩余部分将不只探讨基本的队列技术，还将讨论如何应用前面两章学到的技术，使用队列技术并且对比它们。

从这一章中，建立定制的服务质量解决方案。下面几节集中于更先进的队列、整形、限速、优化和分类的技术，从下一节开始，我们将讨论更先进的流量策略的实施技术。

6.5 使用服务质量实施流量策略

网络通常都有必须实施的基本流量策略需求。例如，服务提供商给用户提供 WAN 的电

路，例如 ATM 或者帧中继。这些电路具有某种程度的服务级别约定，这是服务提供商承诺给用户提供的某种级别的服务。用户负责确保它们的网络流量和约定一致，可以通过整形、速率限制和优化等思科 IOS 软件中提供的服务质量工具来实现这个功能。本节探讨这些技术并显示它们是如何用来对网络应用程序提供服务质量的。

6.6 流量整形

流量整形通过减少外出流量的速率来强迫流量遵循某种带宽的分配限制。不像流量监管会丢弃超过突发尺寸的流量，它是将突发的流量放入到流量整形的缓冲区中，当带宽可用时，再将它们发送出去，或者是当缓冲的数据包的数量低于配置的限制时，再发送出去，因此平滑流量的输出。

注意：流量整形并不替代正常的电路配置。它设计的主要目的是平滑流量的突发。流量整形不给一个接口提供额外的带宽，接口在持续拥塞的情况下还是会丢弃数据包。

流量整形使用一种令牌桶的系统来决定是否传输、延迟或者丢弃新的数据包。使用这种令牌桶系统，每一个接口都有*承诺的信息速率 (CIR)*，它是在一个时间段内接口能够传输数据包的速率。*持续突发速率 (Bc)* 定义了在一个时间间隔内令牌桶可以含有的最大令牌数。当数据包到达一个接口后，它就会从令牌桶中取出一个令牌。当数据包被发送以后，令牌就会释放。当过了*时间间隔 (Tc)* 后，这个令牌就会返回到令牌桶中。如果令牌桶空了，任何新到达那个接口的数据包都会被放到队列中，直到时间间隔过去，令牌又重新填入。如果 CIR 持续超过，令牌就会以大于它们添加的速度从令牌桶中挪走，而去填充队列并且导致数据包被丢弃掉。好的流量整形设计的关键是建立的令牌桶能够持续地有足够的令牌来排队或者转发每一个数据包，当数据包从缓冲区移走并且发送后可以替换令牌。

通用流量整形

流量整形可以应用到一些不同的二层技术中去，例如以太、ATM（可变比特率[VBR]和可用比特率[ABR]）、高级数据链路控制（HDLC）、PPP（ISDN 和拨号接口不支持）和帧中继。除了帧中继以外，所有的这些技术都支持通用流量整形（GTS），这个功能是在思科 IOS 软件版本 11.2 中介绍的，GTS 有能力在每一个接口的基础上平滑输出的流量。GTS 也可以整形在访问控制列表中定义的某种类型的流量，通过在流量整形中指定组来实现。

注意：关于帧中继流量整形（FRTS）的更多信息，参考《CCIE 实验指南（第 1 卷）》第 5 章。

在启用 GTS 之前，你必须知道一些事情。首先，就像帧中继的流量整形一样，为了配置 GTS，必须知道对于接口的目的比特速率，通常被称为*承诺信息速率 (CIR)*。这个速率指的是流量在正常情况发送的速率。知道对于流量突发的持续和过量的突发速率也是很有帮助

的，但不是必需的。*持续突发速率* (Bc) 指的是在每个时间间隔内流量被允许突发超出正常流量速率的速率，以比特表示。*过量突发速率* (Be) 是指在第一个时间间隔内，流量被允许突发超出持续突发速率的速率。每隔一个时间间隔 (Tc)，流量会被填充到流量整形的令牌桶中。为了正确配置流量整形，首先必须知道流量整形用于填充令牌桶的时间间隔，通过使用下面的公式： $Tc = Bc/CIR$ 。

注意：流量整形的时间间隔不能小于 10ms 或者大于 125ms。路由器基于 $Tc = Bc/CIR$ 的公式发现最好的时间间隔。默认的时间间隔是 125ms。这个时间间隔是 CIR 和 Bc 配置的结果，用户不可配置。思科建议 Bc 应当是 CIR 的 1/8，它将会在每秒钟内产生 8 个 125ms 的时间间隔。

为了对所有的接口流量配置 GTS，在每一个需要流量整形的接口上使用 **traffic-shaping rate** 命令。为了定义特定的需要做整形的流量，使用 **traffic-shaping group** 命令和一个访问控制列表。表 6-16 显示了在思科 IOS 软件版本 12.12 (T) 中可用的 GTS 命令、命令的参数和参数的描述。

traffic-shape {group | rate access-list} target-bit-rate [sustained] [excess] [buffer-limit]

表 6-16 通用流量整形的命令参数

命令参数	描述
group access-list	指定匹配访问控制列表 (1-2699) 的所有流量都被整形
rate	指定在这个接口上的所有流量都被整形
target-bit-rate	这个流量将被传输的正常速率 (CIR)，范围为 8000 到接口的以每秒比特位表示的完整 CIR。例如，一个 100Mbit/s 接口的完整的 CIR 范围为 8000~100 000 000 某些思科 IOS 软件的版本对于这个命令有不同的数值。必须使用这台路由器正在运行的软件版本所在范围内的一个数值
sustained	(可选) 持续比特率 (Bc) 指的是流量被允许突发的数值，范围从 0~100 000 000，以每个时间间隔内的比特位表示
excess	(可选) 过量比特率 (Be) 指的是在第一个时间间隔内突发的超出持续比特速率的流量，范围从 0~100 000 000，以每个时间间隔内的比特位表示 Be 是一个可选的参数，它会假设令牌桶已经完全满了： $Be=Bc*2$
buffer	(可选) 用于指定一个缓存的限制，范围从 1~4096

GTS 配置需要两个步骤：发现流量整形的数值，并在接口上配置流量整形。

第 1 步 找到正确的流量整形的数值。为了找到对于你的特定的流量整形配置的数值，需要下列信息：

- CIR
- Bc
- Be

如果你将配置流量整形的数值为接口的 CIR 限制，只需要知道某个特定接口的 CIR。对于更高的灵活性，也需要配置 Bc。Bc 指定了在某个时间间隔内接口可以传输的比特位的数量。如果你不知道你的 Bc，可以使用下面的公式来发现它：

$$Bc = CIR * Tc$$

最后，也是可选的表项，就是在配置 GTS 之前配置 Be。Be 指定当接口填充了足够的令牌时可以支持的突发的流量，这通常是考虑在第一个时间间隔内。Be 可以使用下面的公式计算：

$$Be = Bc * 2$$

如果这个接口不能支持突发，可以使用下面的公式：

$$Be = Bc$$

第 2 步 在接口配置模式下，使用 **traffic-shaping** 命令启用流量整形。在下面的范例中，流量整形正在限制接口 serial0/0 上所有流量的速率为 256 kbit/s。这个限制是在每个时间间隔内将流量限制为 32 Kb；用于整形流量的时间间隔为 125ms。所以，在这种情况下，在每一个 125ms 的时间间隔内，接口 serial0/0 可以传输到 32 Kb。在这个时间间隔内任何超过 32 Kb 的流量将被放在队列中直到下一个时间间隔才进行传输。

```
interface Serial0/0
ip address 10.1.1.5 255.255.255.0
traffic-shape rate 256000 32000 32000 1000
```

Router# show traffic-shape

Interface	Se0/0							
VC	Access List	Target Rate	Byte Limit	Sustain bits/int	Excess bits/int	Interval (ms)	Increment (bytes)	Adapt Active
		256000	8000	32000	32000	125	4000	

在范例 6-21 中，来自网络 136.78.65.0/28 的数据包会通过 WAN 接口进行传输，如图 6-9 所示。流量整形用于将离开 Ethernet 0 的源地址开始为 136.78.65.0/28 的数据包的流量限制为 512 kbit/s，而具有 64Kb/interval 的持续比特速率。在这种情况下，没有过量的突发速率。这意味着来自 136.78.65.0/28 网络的流量将在 8 个 125ms 的时间间隔内被整形为 64Kb，过量的流量将被放在队列中，直到下一个时间间隔才被发送出去，防止接口在 125ms 时间段内发送超过 512 kbit/s 或者 64 Kb 的流量。

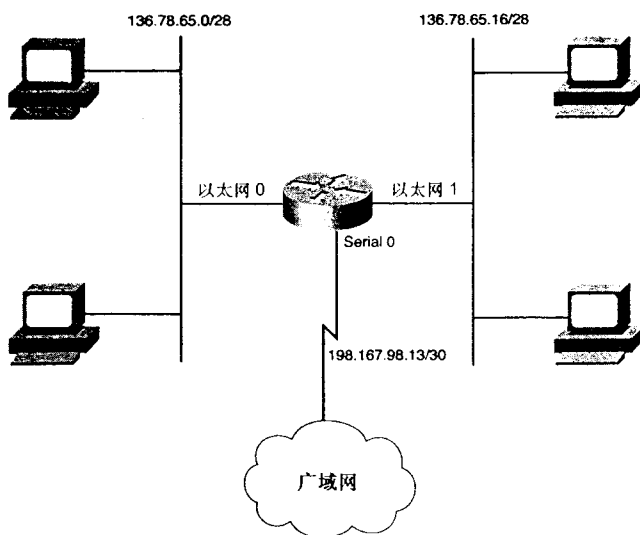


图 6-9 使用 GTS 来限制 LAN 的流量适应 WAN 的流量

范例 6-21 使用 GTS 来整形流量

```
interface Ethernet0
 ip address 136.78.65.1 255.255.255.240
 traffic-shape rate 512000 64000 64000
 !
interface Ethernet1
 ip address 136.78.65.17 255.255.255.240
 !
interface Serial0
 ip address 198.167.98.14 255.255.255.252
 !
access-list 136 permit ip 136.78.65.0 0.0.0.15 any
 !
LAN-Router# show traffic-shape
Interface  Et0
VC          Access Target   Byte   Sustain   Excess   Interval   Increment   Adapt
-          List    Rate   Limit  bits/int bits/int  (ms)        (bytes)    Active
-          136    512000 8000   64000    0        125         8000      -
LAN-Router# show traffic-shape statistics
I/F          Access Queue   Packets   Bytes   Packets   Bytes   Shaping
Et0          List  Depth           Delayed   Delayed   Active
Et0          136    0        39        2886     0        0        no
```

为了查看 GTS 的配置，使用 **show traffic-shaping** 命令。**show traffic-shaping statistics** 命令允许用户监控 GTS 的活动。这个命令可以显示每一个启用了 GTS 的接口的当前队列深度的信息、有流量整形队列延迟的发送的数据包的数量、没有流量整形队列延迟的发送的数据包的数量，以及流量整形当前是否正在生效。只要流量的速率低于流量整形的速率，流量就不会被整形。当流量速率超过配置的流量整形参数时——或者，换句话说，如果数据包到达接口的速率大于令牌被填充的速率——过度的流量就会被整形。只有当接口超过它的 CIR、Bc 和 Be 时，流量整形才会激活。

6.7 使用 CAR 分类和标记流量

承诺访问速率 (CAR) 是一种流量策略的分类和标记的方法，它基于 IP 优先级、DSCP 值、MAC 地址或者访问控制列表来限制 IP 流量的速率。

流量策略的分类包括定义一个流量策略并且使用 CAR 来实施速率限制。遵循配置的速率限制的流量可以被转发，或者被标记在整个网络的路径中提供服务质量。

标记可以改变在 IP 报头的 ToS 字节中的 IP 优先级或者 DSCP。流量标记的行为在数据包遵循一个数值或者超过一个数值的情况下发生。通过标记流量，CAR 可以影响流量在网络中以后是如何对待的，这是因为加权公平队列和 WRED 可以作用于 CAR 分配的 ToS 值，给高优先级的流量更大的权重。

CAR 使用令牌桶的机制，类似于流量整形使用的方法决定一个接口是否有可用的资源来传输一个数据包，通过检查来查看在令牌桶中是否有足够的令牌。如果一个接口有可用资源来转发数据包（有可用的令牌），令牌可以从令牌桶中挪走，数据包被转发，当这个时间间隔过去后，令牌会重新添加到令牌桶中。如果接口没有资源可用，没有可用的令牌，那么 CAR

可以定义对数据包采取的行为。CAR 匹配的流量行为是遵循的行为，即数据包可以遵循一个特定的流量行为，或者是过量的值，即流量超出一个特定的流量的数值。CAR 使用 3 种速率定义来定义流量的速率：

- **Normal rate**（正常的速率）——就像流量整形中的 CIR，在 CAR 中正常的速率可以被描述为流量的平均速率，或者是令牌被添加到令牌桶中的平均速率。
- **Normal burst**（正常的突发）——就像流量整形中的持续比特速率（Bc），正常的突发是在时间间隔内允许超过正常流量速率的流量。
- **Excess burst**（过量突发）——超过正常突发的流量。当配置过量突发时，会借令牌并且将它添加到令牌桶中来允许某种程度的流量突发。当被借的令牌已经使用后，在这个接口上收到的任何超出的流量会被扔掉。流量突发只会发生在短时间内，直到令牌桶中没有令牌存在才停止传输。
 - 思科建议正常的流量速率等于在一段时间内的平均流量速率。正常的突发速率应当等于正常速率的 1.5 倍（对于位，乘以 8）。如果你计划使用扩展速率，扩展速率必须大于正常的突发速率。如果扩展的突发速率没有大于正常的突发速率（ $Bc = Be$ ），接口就不允许扩展的突发，所以扩展速率应当是正常突发速率的 2 倍。如果你有一个速率为 1.544 Mbit/s，正常的突发速率为 2.316 Mbit/s，过量的突发为 4.632 Mbit/s。

注意：当决定在网络中使用哪种整形或者限速的方法时，总是遵循一个规则：流量整形使用缓存区来整形流量，所以整形总是应当在输出接口上进行，输出接口可以缓存过度的流量。流量监管或者 CAR 当应用于输入流量时会更有效，这是因为限速和速率限制并不会缓存流量。

为了配置 CAR，在接口配置模式下使用 **rate-limit** 命令。表 6-17 显示了 **rate-limit** 命令的参数和它们在思科 IOS 软件版本 12.2（12）T 中的描述。

```
rate-limit {input | output} {rate | access-group {access-list-number | rate-limit access-list-number} | dscp dscp-value | qos-group qos-group-index } normal-burst maximum-burst conform-action {continue | drop | set-dscp-continue dscp-value | set-dscp-transmit dscp-value | set-mpls-exp-continue mpls-exp-value | set-mpls-exp-transmit mpls-exp-value | set-prec-continue precedence-value | set-prec-transmit precedence-value | set-qos-continue qos-group-index | set-qos-transmit qos-group-index | transmit} exceed-action {continue | drop | set-dscp-continue dscp-value | set-dscp-transmit dscp-value | set-mpls-exp-continue mpls-exp-value | set-mpls-exp-transmit mpls-exp-value | set-prec-continue precedence-value | set-prec-transmit precedence-value | set-qos-continue qos-group-index | set-qos-transmit qos-group-index | transmit}
```

表 6-17 CAR 命令的参数和它们的描述

命令参数	描述
input output	指定流量的方向
normal-rate	平均流量的速率，在正常的情况下，对于一段时间，以位/秒表示，范围从 8000～2 000 000 000
access-group {access-list-number rate-limit access-list-number} dscp dscp-value qos-group qos-group-index	指定一个标准或者扩展的访问控制列表，范围从 1～2699，或者是一个速率列表 速率列表 0-99 用于指定 IP 优先级的值，速率列表 100～199 用于指定 MAC 地址 指定一个 DSCP 的值，范围从 0～63 指定一个服务质量组，范围从 0～99
normal-burst	指定正常的突发尺寸，以字节表示，范围从 1000 到 512 000 000 正常突发可以使用下面的公式： 正常突发（Bc）= 正常的速率（CIR 以字节表示）* 1.5s

续表

命令参数	描述
<i>maximum-burst</i>	指定以字节表示的过量突发大小，范围从 2000~1 024 000 000 如果使用，可以使用下面的公式发现过量突发速率： 过量突发 (Be) = 正常的突发 (Bc) * 2 否则，过量突发等于正常突发，正如以下所示： 过量突发 (Be) = 正常的突发 (Bc)
<i>conform-action</i>	任何遵循正常速率的数据包将会执行这个命令的下一个数值所规定的行为
<i>continue</i> <i>drop</i>	继续执行列表的剩余部分 立刻丢弃数据包并且退出列表
<i>set-dscp-continue dscp-value</i> <i>set-dscp-transmit dscp-value</i>	将 DSCP 的值设置为特定的值，范围从 0~63，并且继续执行剩余的列表 设置 DSCP 的值，范围从 0~63，传输这个数据包，退出这个列表，而不做进一步的处理
<i>set-mpls-exp-continue mpls-exp-value</i> <i>set-mpls-exp-transmit mpls-exp-value</i>	设置 MPLS 的实验值，范围从 0~7，并且继续处理剩余的列表 设置 MPLS 的实验值，范围从 0~7，立刻传输这个数据包，退出这个列表，而不做进一步的处理
<i>set-prec-continue precedence-value</i>	设置 IP 优先级的值，范围从 0~7，并且继续处理剩余的列表
<i>set-prec-transmit precedence-value</i>	设置 IP 优先级的值，范围从 0~7，传输这个数据包，退出列表而无需做进一步的处理
<i>set-qos-continue qos-group-index</i> <i>set-qos-transmit qos-group-index</i> <i>transmit</i>	对于数据包，设置服务质量组号，范围从 0~99，继续处理剩余的列表 对于数据包，设置服务质量组号，范围从 0~99，传输数据包，退出列表而无需做进一步的处理 传输数据包并且停止对剩余的列表处理
<i>exceed-action</i>	指定当超过正常的速率时应该采取的行为
<i>continue</i> <i>drop</i> <i>set-dscp-continue dscp-value</i>	对超过的流量的处理行为 继续处理剩余的列表 立刻丢弃数据包并且退出列表
<i>set-dscp-transmit dscp-value</i> <i>set-mpls-exp-continue mpls-exp-value</i> <i>set-mpls-exp-transmit mpls-exp-value</i>	设置 DSCP 的值内特定的值，范围从 0~63，继续处理剩余的列表 设置 DSCP 的值，范围从 0~63，传输这个数据包，退出这个列表，而不做进一步的处理 设置 MPLS 的实验值，范围从 0~7，并且继续处理剩余的列表 设置 MPLS 的实验值，范围从 0~7，立刻传输这个数据包，退出这个列表，而不做进一步的处理
<i>set-prec-continue precedence-value</i> <i>set-prec-transmit precedence-value</i>	设置 IP 优先级的值，范围从 0~7，并且继续处理剩余的列表 设置 IP 优先级的值，范围从 0~7，立刻传输这个数据包，退出这个列表，而不做进一步的处理
<i>set-qos-continue qos-group-index</i> <i>set-qos-transmit qos-group-index</i> <i>transmit</i>	设置服务质量组号码，范围从 0~99，并且继续处理剩余的列表 设置服务质量组号码，范围从 0~99，立刻传输这个数据包，退出这个列表，而不做进一步的处理 传输这个数据包，退出这个列表

在范例 6-22 中，**rate-limit** 命令用于和访问控制列表 101 一起工作，限制来自主机 195.42.48.155 的进入流量的速率为 2 Mbit/s，具有 375 000 字节的正常突发和 750 000 字节的过量突发。任何遵循正常流量速率的流量都将它的 IP 优先级设置为 Flash-override (4)，并且立刻进行传输。超过正常突发速率的流量将被列表继续处理。

接下来，CAR 用于对流量进行限速和标记。首先，**rate-limit** 命令和访问控制列表 102 一起使用，把到达主机 195.42.48.7 的所有被动的 FTP 流量限制为 4 Mbit/s。接着，它也用于设置正常的突发速率为 75 000 字节和扩展的突发速率为 1 500 000 字节。任何遵循速率限制的流量都应当被传输，路由器应当接着查找 CAR 列表继续处理。任何超过这个规则的 FTP 流量应当被丢弃。

范例 6-22 使用 CAR 来限速和标记流量

```
interface Ethernet0
  ip address 195.42.48.1 255.255.255.0
  rate-limit input access-group 101 2000000 375000 750000 conform-action set-prec-
continue 4 exceed-action continue
  rate-limit input 2000000 3000 6000 conform-action
  transmit exceed-action drop
  rate-limit input access-group 102 4000000 750000 1500000 conform-action continue
exceed-action drop
  rate-limit output 2000000 3000 6000 conform-action
transmit exceed-action drop
!
access-list 101 permit ip any host 195.42.48.155
access-list 102 permit tcp any host 195.42.48.7 eq ftp
access-list 102 permit tcp any host 195.42.48.7 gt 1023 established
```

注意：下面的公式用于找到这个 FTP 范例的 CAR 参数。

1. 以字节表示的正常速率=以 bit/s 表示的正常速率*（1 个字节/8 位=125）
4 000 000 位*125 = 500 000 000 位= 500 000 字节
2. 正常的突发=以字节表示的正常速率*1.5s
500 000 字节* 1.5 = 750 000 字节
3. 过量的突发=正常的突发*2

另外一种使用 CAR 来指定流量的方法是使用 **access-list rate-limit** 命令，和速率列表联合使用来对基于 IP 优先级或者 MAC 地址的流量进行限速。**access-list rate-limit** 命令类似于 **access-list** 命令，列表 0~99 是 IP 优先级的列表，要么用于指定一个准确的 IP 优先级的值（0~7），要么使用掩码指定某个优先级的值。列表 100~199 用于指定 MAC 地址。

```
access-list rate-limit list-number { precedence-value | precedence-mask }
access-list rate-limit list-number MAC-address
```

优先级的掩码可以将 IP 优先级的值转换成 8 位数字来建立。routine value（0）被转换成 8 位数字 00000001，例如，priority bit（1）被转换成 00000010，如表 6-18 所示。

表 6-18 IP 优先级的值和掩码值

优先级的值	8 位的数字值	优先级的值	8 位的数字值
Routine（0）	00000001	Flash-override（4）	00010000
Priority（1）	00000010	Critical（5）	00100000
Immediate（2）	00000100	Internet（6）	01000000
Flash（3）	00001000	Network（7）	10000000

为了找到对于 IP 优先级掩码的位掩码的值，每一匹配的优先级的值增加一个 8 位的数字。这个数字接着被转换成十六进制，也就是命令所需的格式。例如，为了匹配所有的高优先级的流量——Network、Internet 和 Critical——一个二进制位掩码 11100000 被转换成十六进制，这等同于 E0。

Network（7）	10000000
Internet（6）	01000000
Critical（5）	00100000
Bitmask =	11100000

所以，为了建立一个访问控制列表来匹配 IP 优先级的值 1、3、5、7，需要建立掩码 10101010，而这个掩码需要转换成十六进制 AA。

Network (7)	10000000
Critical (5)	00100000
Flash (3)	00001000
Priority (1)	00000010
Bitmask =	10101010

范例 6-23 显示了如何使用 rate-limit 访问控制列表来指定偶数的 IP 优先级的流量并且被限速为 256 kbit/s，具有 48 000 字节的正常突发和 96 000 字节的过量突发。

范例 6-23 使用 Rate-Limit 访问控制列表

```
interface Serial0/0
 ip address 36.128.42.11 255.255.255.0
 rate-limit output access-group 1 256000 48000 96000 conform-action continue
 exceed-action drop
 !
 access-list rate-limit 1 mask AA
```

为了验证并且监控 CAR 的行为，使用 **show interface rate-limit** 命令。这个命令显示了关于在每一个接口的基础上配置速率限制的信息。范例 6-24 显示了在范例 6-23 中 Serial 0/0 接口的 **show interface rate-limit** 命令的 CAR 配置的情况。

范例 6-24 show interface rate-limit 命令

```
Simpson# show int e 0 rate-limit
Simpson#show interfaces serial 0/0 rate-limit
Serial0/0
Output
 matches: access-group 1
  params: 256000 bps, 48000 limit, 96000 extended limit
  conformed 2050 packets, 1534364 bytes; action: continue
  exceeded 629 packets, 514122 bytes; action: drop
  last packet: 160ms ago, current burst: 122 bytes
  last cleared 00:21:28 ago, conformed 9000 bps, exceeded 3000 bps
```

既然我们已经学习了控制流量策略的基本方法，例如使用速率限制和流量整形，现在我们来查看如何利用 IP RTP 优先级来优化实时的语音流量。

6.8 优化实时的语音流量

*IP RTP 优先级*允许所有发送方向的实时协议（RTP）流量在接口的级别上严格地优于其他的流量发送，其他的流量使用加权公平队列。IP RTP 优先级在低于链路速率 1.544 Mbit/s（T1）的链路上非常有用，而语音流量由于碎片、拥塞或者串行化会产生很大的延迟。因为语音流量是一个实时的流量，它对延时非常敏感。可以在接口配置模式下使用 **ip rtp priority** 命令来启用 IP RTP 的优先级。通过 **ip rtp priority** 命令建立的优先级

队列是一个严格优先级的队列，当在 `ip rtp priority` 命令中配置的带宽被超过后，所有后来到达那个队列的数据包都会被丢掉，直到队列空间可以存储新的数据包。在任何接口上配置 RTP 优先级之前，应当收集一些重要的信息：必须被建立的语音呼叫的数量、语音编解码的方法和呼叫的频率。也必须考虑是只需要优化语音流量，还是需要优化其他的控制信息。出于这个原因，IP RTP 优先级的带宽必须要正确地分配。就像 LLQ 一样，总是应当配置比所需的带宽稍微多一点的带宽，以避免数据包被丢弃掉，因为有数据包的头、网络的抖动或者控制流量这些因素存在。和 CBWFQ 和 LLQ 一样，给 IP RTP 优先级配置的带宽总和不能超过接口可用带宽的 75%，剩下的 25% 的带宽保留用于网络控制信息和路由流量。

为了从接口配置模式下启用 IP RTP 优先级，使用 `ip rtp priority` 命令。表 6-19 列出了 `ip rtp priority` 命令的参数和它们的描述。

`ip rtp priority starting-port-number port-range bandwidth`

表 6-19 ip rtp priority 命令参数

命令参数	描述
<code>starting-port-number</code>	分配给优先级队列的起始的 RTP 端口号码。RTP 端口号是 UDP 端口号，范围从 2000～65 535
<code>port-range</code>	RTP 端口的范围，和起始的 RTP 端口范围相加，得到的是整个进行优化的 RTP 端口范围，范围从 0～16 383
<code>bandwidth</code>	指定用于 RTP 优先级队列的最大的可使用带宽，范围从 0～2000，以 kbit/s 表示

范例 6-25 显示了如何使用 RTP 优先级来严格地优化所有的 RTP 流量，它们的范围从 UDP 端口号 16 384～32 767（RTP 端口号的完整范围），并且将优先级队列的带宽限制为 64kbit/s。在这个接口上所有的其他流量使用加权公平队列公平分配。

范例 6-25 使用 ip rtp priority 来优化语音流量

```
interface Serial0
bandwidth 256
ip address 85.114.95.1 255.255.255.0
encapsulation frame-relay
fair-queue 64 256 0
frame-relay interface-dlci 110
ip rtp priority 16384 16383 64
```

为了验证 RTP 的配置，可以使用 `show interface` 或者 `show queue` 命令。每个命令都显示相同类型的 RTP 优先级的数据、预留的带宽。范例 6-26 显示了在 RTP 优先级应用之前 `show interface` 命令的输出，而范例 6-27 显示了在 RTP 优先级应用之后 `show interface` 和 `show queueing` 命令的输出。

范例 6-26 配置 RTP 优先级之前

```
Simpson#show interfaces serial 0 | begin Queue
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/2/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 1158 Kilobits/sec
```


范例 6-27 显示 RTP 优先级的带宽

```
Simpson#show queueing interface serial 0
Interface Serial0 queueing strategy: fair
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/2/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1094 kilobits/sec

Simpson#show interfaces serial 0 | begin Queue
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/2/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 1094 kilobits/sec
```

第一个范例显示了在应用 RTP 优先级之前的接口。在这个范例中，接口对所有的接口流量都有 1158 kbit/s 的可用带宽（1158 kbit/s 正好是串行接口可用带宽的 75%，而另外 25% 的带宽保留用于路由协议和信令信息）。第二个范例显示了当应用 RTP 优先级之后同一路由器的接口。在这个范例中，RTP 优先级配置了对 RTP 严格优先级队列保留 64 kbit/s 的带宽，所以只有 1094 kbit/s 的带宽可以用于其他未指定的流量。**debug priority** 命令显示了对严格优先级 RTP 队列的加权公平队列输出的丢弃。

就像你所看到的，对接口设置 RTP 优先级来保留小量的带宽对于实时的、关键的、对延迟敏感的 RTP 流量来说是非常节省资源的。本节也显示了如何应用流量整形、速率限制，以及在一个接口的基础上如何优化语音流量来实施服务质量技术。下一节将探讨更强大和灵活的服务质量工具，即在思科 IOS 软件内的基于类别的队列解决方案。

6.9 基于类别的队列解决方案

基于类别的加权公平队列 (CBWFQ) 合并了高级定制队列和加权公平队列的优点，建立了一个高级队列方法，它对 64 个用户可定义类别进行公平的队列分配。CBWFQ 类别可以通过协议类型、访问控制列表或者输入接口等信息定义，每一个类别有它自己的队列。类别可以使用带宽、权重和队列的尺寸等信息进行定制。当一个队列超过它的最大尺寸之后，数据包就会使用尾部丢弃的方法被丢弃掉，这是默认的丢弃方法，或者 WRED 如果配置了 WRED 的话。不匹配任何类别特性的流量会被发送到默认的队列，在那里，每个流会公平地使用加权公平队列来共享带宽（流量共享同一个源和目的地址和端口号码）。

在配置 CBWFQ 之前，你需要意识到一些规则，包括下面这些规则：

- 在应用 CBWFQ 之前，接口必须运行它们的默认的队列方法。CBWFQ 会覆盖默认的队列方法。
- 除非特别指定，当丢弃数据包时，CBWFQ 使用的是尾部丢弃的方法，而不是采用 WRED。
- 如果你计划在 CBWFQ 中使用 WRED，确保这个接口没有配置运行 WRED。
- CBWFQ 不支持子接口，它必须运行在物理接口上。

- CBWFQ 只支持 ATM 可变比特率 (VBR) 和可用比特率 (ABR) 的电路。
- 策略映射可以用于不止一个接口，节省配置的空间。
- CBWFQ 可以配置的带宽不能超过接口带宽的 75%。其余的 25% 带宽用于负荷控制和路由流量。如果一个策略映射所使用的带宽超过了接口的可用带宽，策略映射会被拒绝，并且从所有的接口上清除。
- CBWFQ、定制队列、优先级队列、加权公平队列和 WRED 彼此都是互相排斥的，在其他的队列方法应用之前必须清除相应的服务策略。
- CBWFQ 支持队列尺寸的限制和 WRED，但是在同一个类别的策略里不能两个同时存在。

就像下一小节所讨论的，CBWFQ 是一个强大的服务质量工具。使用 CBWFQ，可以配置非常灵活的服务质量策略，在同一个接口上，以不同的方法来管理不同类型的流量。

CBWFQ 也可以使用基于网络的应用程序识别 (NBAR) 协议来区分网络流量。虽然本书没有详细地介绍 NBAR，但是 NBAR CBWFQ 的配置在本章的后面会进行介绍。

注意：NBAR 协议可以帮助识别一些协议和应用程序，而先前可能需要一些较长的、复杂的访问控制列表来实现。NBAR 使用数据包描述语言 (PDL) 来定义协议的特性。PDL 可以在思科网站的思科 IOS 软件的软件下载区域中找到，或者是在其他的思科 IOS 扩展区域、数据包描述语言模块中找到。可以在全局配置模式下使用 `ip nbar path: filename` 命令来指定它的位置。

注意：CBWFQ NBAR 支持需要在启用了服务策略的接口上启用思科快速转发 (CEF) 交换。

CBWFQ 类别可以使用分级映射来定义。分级映射含有匹配的条件，它用来指定属于每一个类别的协议。分级映射利用新的思科 IOS 软件模块的命令行接口 (CLI)，并且使用 `class-map` 命令建立。`class-map` 命令在思科 IOS 版本 12.1 和 12.2 中略有区分。在 12.2 中，可以添加可选的 `match-any` 或者 `match-all` 语句。

在思科 IOS 软件版本 12.2 和更高的版本中，也可以通过使用可选的 `match-all` 或者 `match-any` 语句指定分级映射的类型。`match-all class map` 匹配所有的条件 (逻辑与)，而 `match-any class map` 匹配分级映射中所指定的任何一个条件 (逻辑或)。

思科 IOS 软件版本 12.1:

```
class-map class-name
```

思科 IOS 软件版本 12.2:

```
class-map [match-any | match-all] class-name
```

注意：在思科 IOS 软件版本 12.2 中做出了一系列的服务质量变化。在本章中，思科 IOS 软件版本 12.2 用在所有的范例中。为了保持和思科 IOS 软件版本 12.1 的兼容性，我试图使用在 12.1 和 12.2 中都存在的命令。

当建立分级映射后，可以进入分级映射的全局配置模式下，在那里可以指定匹配的条件。

在分级映射的配置模式下使用 `match` 命令，可以定义分级映射来使用访问控制列表、输入接

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

口、协议类型和许多其他的条件作为定义的条件。表 6-20 显示了在思科 IOS 软件版本 12.2 (7) 中分级映射的配置命令和它们的定义。

表 6-20 分级映射 match 命令值

匹配的命令	IOS 版本	描述
access-group {access-list-number name access-list-name}	12.1	匹配一个访问控制列表。范围从 1~2699，或者是一个命名的访问控制列表
any	12.2	匹配任何数据包
class-map class-map-name	12.2	匹配另外一个嵌入的分级映射
cos cos-value	12.2	类型服务 (CoS) 匹配任何一个 IEEE 802.1Q/ISL 类别服务/用户优先级的值，范围从 0~7。可以使用空格将总共 4 个 CoS 的值输入
destination-address mac hex-address	12.2	匹配一个目的 MAC 地址为十六进制格式的 xxxx.xxxx.xxxx
input-interface interface-name interface-number	12.1	匹配一个输入接口
ip {dscp dscp-value precedence precedence-value rtp lower- port-range range}	12.2	ip dscp 匹配总共 8 个 DSCP 的值，范围从 0~63，表 7-14 所提到的 12 个 AF 类别之一，7 个类别选择编码点 (CS) 对应于一个 IP 优先级的值，默认的 DSCP 值，或者是加速转发 (EF) PHB 的值 ip precedence 匹配 (总共 4 个) IP precedence 的值，要么使用一个整数 (0~7)，要么使用表 6-14 所示的 IP 优先级的值 ip rtp 匹配一个从 2000~65535 的 RTP UDP 端口号范围，和一个从 0~16383 的 RTP UDP 端口号范围
mpls experimental value	12.2	多协议标签交换 (MPLS) 匹配总共 8 个 MPLS 的值，范围从 0~7
not {access-group access- list-number any class-map class-map-name destination- address mac hex-address input-interface interface-name interface- number ip {dscp dscp-value precedence precedence-value rtp lower-port-range range} mpls value qos-group qos- group-index source-address mac hex-address}	12.2	不要匹配任何 access-group、任何 class-map、目的地址、输入接口、ip、mpls、qos-group 或者指定的源地址
protocol protocol-name	12.1*	匹配 NBAR 所识别的特定协议： arp ——IP ARP bgp ——BGP 协议 bridge ——桥接协议 bstun ——Block Serial Tunnel cdp ——思科发现协议 citrix ——Citrix 流量 clns ——ISO CLNS clns_es ——ISO CLNS 终端系统 clns_is ——ISO CLNS 中间系统 cmns ——ISO CMNS compressedtcp ——被压缩的 TCP cuseeme ——CU-SeeMe Desktop 视屏 custom-01 ——Custom protocol Custom-01 custom-02 ——Custom protocol Custom-02 custom-03 ——Custom protocol Custom-03 custom-04 ——Custom protocol Custom-04 custom-05 ——Custom protocol Custom-05 custom-06 ——Custom protocol Custom-06 custom-07 ——Custom protocol Custom-07 custom-08 ——Custom protocol Custom-08 custom-09 ——Custom protocol Custom-09

续表

匹配的命令	IOS 版本	描述
protocol protocol-name (继续)		custom-10 ——定制协议 custom-10 dhcp ——DHCP 协议 dls ——数据链路交换 dns ——DNS 查找 egp ——EGP 路由协议 eigrp ——EIGRP 路由协议 exchange ——MS-RPC for Exchange fasttrack ——FastTrack 流量 (KaZaA、Morpheus、Grokster 等等) finger ——Finger ftp ——FTP 协议 gnutella ——Gnutella 流量 (BearShare、LimeWire、Gnutella 等等) gopher ——Gopher gre ——GRE 隧道协议 http ——HTTP web 流量 icmp ——ICMP 协议 imap ——IMAP 协议 ip ——IPv4 协议 ipinip ——IP in IP 隧道协议 ipsec ——P 安全协议 (ESP/AH) ipv6 ——IPv6 ipx ——Novell IPX irc ——Internet Relay 协议 kerberos ——Kerberos 验证 l2tp ——L2F/L2TP 隧道 ldap ——LDAP 目录协议 llc2 ——LLC-2 napster ——Napster 流量 netbios ——NetBIOS netshow ——Microsoft NetShow nfs ——UNIX 网络文件系统 nntp ——网络新闻传输协议 notes ——Lotus Notes novadigm ——Novadigm EDM ntp ——网络时钟协议 pad ——X.25 PAD pcanywhere ——Symantec pcANYWHERE pop3 ——Post Office 协议 pptp ——Microsoft PPTP 隧道协议 printer ——LPD print spooler qlc ——QLLC 协议 rcmd ——BSD r 命令 (rsh, rlogin, rexec) realaudio ——实时音频流协议 rip ——RIP 路由协议 rsrb ——RSRB 桥接 rsvp ——RSVP 协议 rtp ——实时时间协议 secure-ftp ——FTP over TLS/SSL secure-http ——安全的 http secure-imap ——IMAP over TLS/SSL secure-irc ——IRC over TLS/SSL secure-ldap ——LDAP over TLS/SSL secure-nntp ——NNTP over TLS/SSL secure-pop3 ——POP3 over TLS/SSL secure-telnet ——Telnet over TLS/SSL smtp ——SMTP 协议 snapshot ——快照路由协议 snmp ——SNMP 协议 socks ——SOCKS sqlnet ——SQL*NET for Oracle

续表

匹配的命令	IOS 版本	描述
protocol protocol-name (继续)		sqlserver ——MS SQL 服务器 ssh ——Secured Shell streamwork ——Xing Technology StreamWorks player stun ——串行隧道协议 stunrpc ——Sun RPC syslog ——系统日志工具 telnet ——Telnet tftp ——TFTP 协议 vdolive ——VDOLive streaming video vofr ——语音 over 帧中继 xwindows ——X Windows 远程访问 xns ——Xerox 网络服务
qos-group qos-group-index	12.2	匹配一个特定的服务质量组，范围从 0~99
source-address mac hex-address	12.2	匹配一个源 MAC 地址为十六进制格式的 xxxx.xxxx.xxxx

* 不是在所有的思科 IOS 版本中都提供所有的协议。

当你已经输入分级映射配置模式后，你可以输入我们已经介绍过的 **match** 命令。为了对分级映射配置一个描述，使用 **description** 命令。为了重新命名分级映射而无需清除它，使用 **rename** 命令。

当定义分级映射之后，必须定义一个策略映射来使得这个策略应用于分级映射。策略映射是使用 **policy-map policy-name** 命令定义的，它使你进入到策略映射配置模式下，可以看到 (**config-pmap**) #提示符。策略映射使用 **service policy** 绑定到接口，使用的是 **policy-map** 命令。在这个模式下，也可以给策略映射添加一个描述，修改它的配置或者重新命名策略映射。

注意：使用思科模块化的服务质量命令行接口，也可以在一个类别和策略下嵌入其他的策略和类别，通过这种方法可以建立非常灵活的服务质量配置，而无需重新输入每一个类别或者策略的定义。

在策略映射配置模式下，必须定义策略映射将使用的类别，使用 **class class-name** 命令，它使你进入到策略映射类别配置模式下，看到 (**config-pmap-c**) #提示符。

当你处于策略映射类别配置模式下，这个模式用于配置你先前指定的类别的策略，可以对这个服务策略定义参数。表 6-21 显示了服务策略的参数。

表 6-21

服务策略的参数

策略命令	IOS 版本	描述
bandwidth {bandwidth-list percent percentage remaining percent remaining-percentage}	12.1	对于这个类别分配一个带宽的限制。这个限制可以用 kbit/s 来指定，或者用一个百分比来指定，（不要超过接口带宽的 75%） 为了使用一个带宽的特定量，输入这个量，范围从 8 到 2,000,000，以 kbit/s 表示 为了指定接口带宽的百分比，使用 percent 或者 remaining percent 关键字，跟随一个范围从 1 到 100 的值
police {rate-bps {[normal-burst-][excess-burst]} [bc normal-burst][bc excess-burst] cir rate-bps [normal-burst][excess-burst][bcnormal-burst][be excess-burst] pir[peak-rate][excess-burst]}[conform-action action][exceed-action action][violate-action action]}	12.2	对这个类别的流量启用流量监管 基于类别的限速会在本章后面讨论

续表

策略命令	IOS 版本	描述
priority {bandwidth burst percent percentage burst}	12.1	在一个服务策略内定义一个严格优先级的队列，称为 低延迟队列 (LLQ) ，它会在本章的后面讨论 bandwidth 指定了对一个严格优先级队列的带宽的限制，范围从 8~2 000 000，以 kbit/s 表示 burst 为 32~2 000 000，以字节表示 percent 定义了带宽的百分比，从 1~100 的百分比 burst 范围从 32~200000，以字节表示
queue-list number-of-packets	12.1	定义队列的最大尺寸。当超过队列尺寸后，所有的数据包采用尾部丢弃的方法被丢掉 范围是从 1~512 个数据包。在所有的非 vip 的平台上的默认值是 64 个数据包
random-detect [dscp dscp-value minimum-threshold max-threshold mark-probability-denominator dscp-based ecn exponential-weighting-constant weighed-average prec-based precedence [precedence-value minimum-threshold max-threshold mark-probability-denominator rsvp minimum-threshold max-threshold mark-probability-denominator]	12.1*	对超过最大队列尺寸的数据包启用 WRED dscp 值匹配一个从 0~63 的 DSCP 值（总共 4 个值），12 个 AF 类别中的一个，7 个类别选择编码点相应于一个 IP 优先级的值（0~7），默认的 DSCP 值，加速转发（EF）PHB 的值，以数据包表示的指定了最小和最大极限值的 RSVP 流量，以及可选性的 RSVP 的丢弃比率。可以在表 7-14 中找到 AF、CS 和 EF 的值 dscp-based 启用基于 DSCP 的 WRED，而不是基于 precedence 的 WRED ecn ——加速拥塞通知 exponential-weighting-constant 指定了当计算平均队列长度时 WRED 所使用的权重：默认的权重因子是 9，范围从 1~16，格式为 $2^{[number]}$ prec-based 允许实施基于 precedence 的 WRED，这是 WRED 的默认行为 precedence 配置 IP 优先级的值——每一个 IP 优先级的值，范围从 0~7，一个数据包被包丢弃的最小和最大极限值，以及当达到极限后，一个数据包被丢弃的比率
service-policy	12.2	指定另外一个策略映射的名字
shape	12.2	配置基于类别的整形，在本章的后面讨论 average CIR[Bc][Be] max-buffers 配置了最大的缓冲区的限制 peak CIR[Bc][Be]

* DSCP 命令直到 12.2 的版本才出现。

在默认的情况下，所有未定义的流量属于一个类别，将采用尽力传递的服务，然而，也可以定义一个默认的队列。默认的队列允许对任何未分类的流量进行配置，在这种方式下，默认类别里的未分类的流量要么在一个加权公平队列启用的接口上获得相同级别的服务，每个未分类的流量都将平均分配剩余带宽，要么实现先进先出的队列方式，有带宽限制。

默认的分类是通过建立一个 class-default 类别来实现的，在策略映射配置模式下使用 **class class-default** 命令，它允许你在这个默认的分类进入到策略映射类别配置模式下。

```
Router(config-pmap)#class class-default
```

当定义 class-default 类别时，**fair-queue** 命令可用，允许所有先前未分类的流量使用加权公平队列放到队列中去。这个命令只对默认的分类有用。

```
fair-queue dynamic-queue-limit
```

使用 **fair-queue** 命令，可以对默认的分类中所有的加权公平队列流量定义一个动态的队列限制。**dynamic-queue-limit** 的范围从 16~4096，并且可以以 2 的倍数增长（ $2^{[number]}$ ）。

另外，不是对未分类的流量配置加权公平队列，我们可以使用先进先出尽力传递的队列，设置一个带宽的限制，可以使用 **bandwidth** 命令。

注意：当配置默认的分类时，注意这一点是非常重要的：加权公平队列或者带宽的限制都可以配置，但是两个命令不能同时配置。

默认的分类也可以有类别的参数，例如流量监管、IP RTP 优先级、队列限制导致的尾部丢弃、WRED 和基于类别的整形，这正如在表 6-21 中所提到的。

当配置了分级映射而策略映射也定义完成后，现在可以建立一个服务策略。为了将这个服务策略绑定到接口上，在接口配置模式下使用 **service-policy** 命令。也可以在接口的进入或者发送方向上使用 **service policy** 命令，后面跟着 **input** 或者 **output** 命令参数。

```
Interface serial0
  service-policy {input | output} policy-name
```

使用 CBWFQ，可以将流量类型分类成服务组，并且应用适当的策略来实施适当的流量限制或者优化。在下面的范例中，定义了两个类别。ClassIP 给 IP 流量提供了接口带宽的 25%，并且使用 WRED 来作为一种拥塞避免的机制。ClassIPX 给 IPX 流量提供了另外 25% 的接口带宽；然而，因为 IPX 不被 WRED 支持，在拥塞发生期间，使用尾部丢弃来丢弃数据包。任何其他的未分类的流量使用 16 个 WFQ 的队列放在队列中。

第 1 步 CBWFQ 配置所需要的第一步就是定义类别。在这个范例中，ClassIP 被定义为匹配所有的 IP 流量。

```
Simpson(config)#class-map ClassIP
```

第 2 步 当类别被定义完成后，在分级映射的配置模式下，定义类别的特性。ClassIP 类别必须匹配所有的 IP 数据包，所以需要使用 **match protocol ip** 语句。当匹配的条件定义完成以后，可以退出分级映射的配置模式。

```
Simpson(config-cmap)# match protocol ip
Simpson(config-cmap)# exit
```

第 3 步 （可选）建立另外一个所需的分类，总共 64 个类别。这一步对每一个类别定义都是需要的，它将用于服务策略。在这个范例中，ClassIPX 匹配所有的 IPX 流量。

```
Simpson(config)# class-map ClassIPX
Simpson(config-cmap)# match protocol ipx
Simpson(config-cmap)# exit
```

第 4 步 建立一个策略映射。这个策略映射用于定义类别的策略。一个策略映射可以含有多个类别和它们的策略。在这个范例中，myPolicy 策略用于 ClassIP 和 ClassIPX 的分类策略定义。

```
Simpson(config)# policy myPolicy
```

第 5 步 在策略映射下指定服务策略中所应用的分级映射。为了建立对 IP 流量的服务策略，在 myPolicy 下指定 ClassIP。

```
Simpson(config-pmap)# class ClassIP
```

第6步 在策略映射分类配置模式下，指定策略参数。就像先前所提到的，ClassIP 被分配了接口带宽的 50%，这是通过使用 **bandwidth percent 50** 命令完成的。为了配置这个策略使用 WRED 来做 IP 的拥塞避免，可以使用 **random-detect** 命令，而无需任何参数。

```
Simpson(config-pmap-c)# bandwidth percent 50
Simpson(config-pmap-c)# random-detect
Simpson(config-pmap-c)# exit
```

第7步 （可选）如果需要，对每一个类别定义重复第 5 和第 6 步。接下来，ClassIPX 被分配了接口带宽的 25%。

```
Simpson(config-pmap)# class ClassIPX
Simpson(config-pmap-c)# bandwidth percent 25
Simpson(config-pmap-c)# exit
```

第8步 （可选）对所有未分类的流量建立一个默认类别。在这个范例中，建立了一个默认类别来缓存所有未分类的流量，使用了 16 个动态的加权公平队列。

```
Simpson(config-pmap)#class class-default
Simpson(config-pmap-c)# fair-queue 16
Simpson(config-pmap-c)# exit
Simpson(config-pmap)# exit
```

第9步 当完成建立分级映射和策略后，使用 **service-policy** 命令将策略绑定到接口上。为了激活这个服务策略，它应当绑定到接口上。在这个范例中，它应用于接口 serial 0/1 上的输出流量。

```
Simpson(config)# int s 0/1
Simpson(config-if)# service-policy output myPolicy
```

第10步 范例 6-28 显示了从先前的步骤得到的完整配置。

范例 6-28 CBWFQ 的完整配置

```
class-map match-all ClassIPX
  match protocol ipx
class-map match-all ClassIP
  match protocol ip
!
policy-map myPolicy
  class ClassIP
    bandwidth percent 50
    random-detect
  class ClassIPX
    bandwidth percent 25
  class class-default
    fair-queue 16
!
interface Serial0/1
  ip address 192.168.3.1 255.255.255.252
  ipx network 10AB
  service-policy output myPolicy
```


第 11 步 监控并且验证这个策略的配置，使用 **show policy-map** 或者 **show policy-map interface** 命令。**show policy-map myPolicy** 命令显示了 myPolicy 是如何配置的。在这个范例中，ClassIP 被配置使用加权公平队列，所有的 IP 流量占用了接口带宽的 50%，使用默认的 WRED 的 IP 优先级设置作为 WRED 的配置。ClassIPX 使用接口带宽的 25% 传输 IPX 流量，在发生拥塞的情况下采用尾部丢弃的方法。所有未分类的流量都被分配到 class-default 队列中去，而 class-default 队列使用的是加权公平队列。

```
Simpson# show policy-map myPolicy
Policy Map myPolicy
Class ClassIP
  Bandwidth 50 (%)
    exponential weight 9
  class      min-threshold      max-threshold      mark-probability
  .....
      0      -                  -                  1/10
      1      -                  -                  1/10
      2      -                  -                  1/10
      3      -                  -                  1/10
      4      -                  -                  1/10
      5      -                  -                  1/10
      6      -                  -                  1/10
      7      -                  -                  1/10
      rsvp   -                  -                  1/10

Class ClassIPX
  Bandwidth 25 (%) Max Threshold 64 (packets)
Class class-default
  Flow based Fair Queueing
Bandwidth 0 (kbps) Max Threshold 64 (packets)
```

show policy-map interface serial 0/1 命令显示了关于接口 serial 0/1 的详细信息，包括发送的数据包的数量、数据包的传输速率、丢弃的数据包的数量、放在队列中的数据包的数量和详细的队列信息。

```
Simpson#sh policy-map interface serial 0/1
Serial0/1

Service-policy output: myPolicy

Class-map: ClassIPX (match-all)
  5 packets, 520 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol ip
Queueing
  Output Queue: Conversation 25
  Bandwidth 50 (%)
  Bandwidth 772 (kbps)
  (pkts matched/bytes matched) 5/520
  (depth/total drops/no-buffer drops) 0/0/0
  exponential weight: 9
  mean queue depth: 0
```

class	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
0	5/520	0/0	0/0	20	40	1/10
1	0/0	0/0	0/0	22	40	1/10
2	0/0	0/0	0/0	24	40	1/10
3	0/0	0/0	0/0	26	40	1/10
4	0/0	0/0	0/0	28	40	1/10
5	0/0	0/0	0/0	30	40	1/10
6	0/0	0/0	0/0	32	40	1/10
7	0/0	0/0	0/0	34	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10

```

Class-map: ClassIPX (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol ipx

Queueing
  Output Queue: Conversation 26
  Bandwidth 25 (%)
  Bandwidth 386 (kbps) Max Threshold 64 (packets)
  (pkts matched/bytes matched) 0/0
  (depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)
  140 packets, 9840 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
  Queueing
    Flow Based Fair Queueing
    Maximum Number of Hashed Queues 16
    (total queued/total drops/no-buffer drops) 0/0/0
    
```

当监控启用了 CBWFQ 的接口时，可以使用 **show interface** 的输出来显示默认的分类的配置，包括队列的策略、队列的计数以及加权公平队列是否在接口上启用了，还会显示关于加权公平队列和 RSVP 会话的信息。在使用任何 CBWFQ 的命令之前，**show interfaces** 命令显示的带宽将是接口带宽的 75%。这是 CBWFQ 可以使用的最大带宽。另外 25% 的带宽保留用于路由器的控制流量和路由协议的流量。在这个范例中，在配置 CBWFQ 之前的可用带宽是 1158 Kb，串行接口 1544 Kb 带宽的 75%。当实施了 CBWFQ 之后，接口的可用带宽应当是 0。如果在一个服务策略中所配置的 **bandwidth** 命令超过了可用的带宽，这个策略将会从接口上和任何绑定了这个策略的接口上清除。可以在接口配置模式下使用 **max-reserved-bandwidth percent** 命令改变 CBWFQ 可以使用的带宽，虽然使用这个命令可能严重影响路由器的性能。范例 6-29 显示了在应用 CBWFQ 之前和之后从 **show interfaces** 命令看到的 CBWFQ 是如何影响输出的。

范例 6-29 CBWFQ 和 show interfaces 命令

```

Simpson# show interfaces serial 0/1
Serial0/1 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 192.168.3.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
  reliability 252/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:09, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total: output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations 0/1/16 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)
  Available Bandwidth 1158 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
74999 packets input, 4663284 bytes, 0 no buffer
Received 60312 broadcasts, 0 runts, 0 giants, 0 throttles
7 input errors, 0 CRC, 7 frame, 0 overrun, 0 ignored, 0 abort
60335 packets output, 4175959 bytes, 0 underruns
    
```

(待续)

```
0 output errors, 0 collisions, 15 interface resets
0 output buffer failures, 0 output buffers swapped out
13 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

Simpson# show interfaces serial 0/1
Serial0/1 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 192.168.3.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:06, output 00:00:06, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/16 (active/max active/max total)
    Reserved Conversations 2/2 (allocated/max allocated)
    Available Bandwidth 0 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 74950 packets input, 4660302 bytes, 0 no buffer
Received 60263 broadcasts, 0 runts, 0 giants, 0 throttles
 6 input errors, 0 CRC, 6 frame, 0 overrun, 0 ignored, 0 abort
60284 packets output, 4172143 bytes, 0 underruns
 0 output errors, 0 collisions, 14 interface resets
 0 output buffer failures, 0 output buffers swapped out
13 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

本小节介绍了 CBWFQ 并且描述了使用这个技术进行标记、队列调度或者基于类别来丢弃流量的一些方法。下一小节将讨论 CBWFQ 自己的流量整形机制：基于类别的整形。

6.9.1 基于类别的整形

就像在前面的小节所提到的，在思科 IOS 软件版本 12.2 中，可以启用 CBWFQ 的整形来实现基于类别的整形。基于类别的整形允许用户在整个服务策略内基于分类来配置整形的参数，而不是像 GTS 那样基于接口进行整形。基于类别的整形可以在策略映射类别配置模式下在 CBWFQ 内使用 **shape** 命令来实现。表 6-22 显示了基于类别的整形命令和它们的参数。

```
shape {average target-bit-rate [sustained-bit-rate] [ excess-per-interval] | peak target-
bit-rate [sustained-bit-rate] [ excess-per-interval] | max-buffers buffers }
```

表 6-22 基于类别的整形命令和它们的描述

命令	描述
average target-bit-rate {sustained-bit-rate} {excess-per-interval}	CBS average 命令配置使得路由器将流量整形到一个平均的速率；使用平均速率整形，整形器将所有的流量在每一个时间间隔内整形到一个正常的突发速率。就像 GTS CIR 一样， target-bit-rate 是流量将被传输的正常的速率（CIR），范围从 8000 到接口的完整速率，以比特每秒表示。例如，一个 1.544Mbit/s 的接口的完整 CIR 范围将从 8000~154 400 000 (可选) 就像 GTS 的持续比特速率（Bc）一样，CBS 的 sustained-bit-rate 指的是在每个时间间隔内流量被允许突发到多个 128 的倍数，范围从 256~1 544 000（在串行接口上），以比特每秒表示。思科建议不要手动配置这个值，让算法去配置这个数值 可以使用下面的这个公式找到流量在每个时间间隔内可以突发的速率：

续表

	$Bc = Tc \cdot CIR$ (可选) excess bit /每个时间间隔 (Be) 指的是流量可以允许被突发到超过 sustained-bit-rate ，是 128 的倍数，范围从 0~1 544 000 (在串行接口上)，以比特/每个时间间隔表示。思科建议不要手动配置这个数值，让算法去计算这个数值。如果没有输入 Be ，软件就会假设 $Be = Bc$ 过量突发总是大于正常的突发，所以建议找到 Be 的公式如下： $Be = Bc \cdot 2$
peak target-bit-rate [sustained-bit-rate][excess-per-interval]	CBS peak 命令配置路由器在每个时间间隔内将流量整形到峰值速率 ($Bc + Be$)。使用峰值速率整形，如果有可用的令牌，流量就会被整形到正常的突发速率。就像 GTS 的 CIR 一样， target-bit-rate 流量是可以传输的正常的速率 (CIR)，范围从 8000 到接口的完整速率，以比特每秒表示。例如，一个 1.544Mbit/s 的接口的完整 CIR 的范围将从 8000~1 544 000 (可选) 就像 GTS 的持续比特速率一样 (Bc)，CBS 的 sustained-bit-rate 指的是允许流量被突发到多个 128 的倍数，范围从 0~1 544 000 (在串行接口上)，以比特/每个时间间隔表示。思科建议不要手动配置这个数值，让算法去计算这个数值
peak target-bit-rate [sustained-bit-rate][excess-per-interval]	可以使用下面的这个公式找到流量在每个时间间隔内可以突发的速率： $Bc = Tc \cdot CIR$ (可选) excess bit /每个时间间隔 (Be) 指的是流量可以允许被突发到超过 sustained-bit-rate ，是 128 的倍数，范围从 0~1 544 000 (在串行接口上)，以比特/每个时间间隔表示。思科建议不要手动配置这个数值，让算法去计算这个数值。如果没有输入 Be ，软件就会假设 $Be = Bc$ 过量突发总是大于正常的突发，所以建议找到 Be 的公式如下： $Be = Bc \cdot 2$
max-buffers buffers	(可选 1) 用于指定一个缓存的限制，范围从 1~4096

shape 命令非常类似于 GTS 所使用的 **traffic-shape** 命令。对于整形的类型有两种选择：**average** 和 **peak**。如果使用 **average**，正在整形的流量被整形到目的比特速率所指定的速率 (CIR)，可选地，可以配置一个持续比特速率 (**Bc**) 和过量比特速率 (**Be**)。**peak** 整形类型允许在带宽可用的情况下，流量突发超出 CIR 来达到峰值速率，可以使用范例 6-30 所示的 CIR、**Be** 和 **Bc** 等参数。然而，思科并不建议当使用 CBS 时，手动配置正常和过量突发参数。

范例 6-30 在加权公平队列中使用基于类别的整形

```
class-map match-all Internet-traffic
  match protocol ip
  match access-group 101
!
!
policy-map Internet
  class Internet-traffic
    bandwidth percent 20
    shape peak 768000 19200 38400
!
interface Serial0/1
  ip address 36.128.42.11 255.255.255.0
  service-policy output Internet
!
access-list 101 permit tcp any any eq www
access-list 101 permit tcp any host 192.168.1.1 eq ftp
access-list 101 permit tcp any host 192.168.1.1 gt 1023 established
```

在这个范例中，所有离开接口 serial 0/1 的 web 和被动 FTP 流量都被整形到 768Kbit 的峰值速率，并且被限制使用接口带宽的 20%。在有可用带宽的情况下，如果有可用的令牌的话，在每一个时间间隔内，流量可以突发到 38 400 位，这可以使用 **peak** 命令来实现。范例 6-31

使用 **show policy-map** 命令来验证这个配置。

范例 6-31 验证基于类别的整形的配置

```
Internet-Router# show policy-map Internet
Policy Map Internet
  Class Internet-traffic
    Bandwidth 20 (%) Max Threshold 64 (packets)
    Traffic Shaping
      Peak Rate Traffic Shaping
        CIR 768000 (bps) Max. Buffers Limit 1000 (Packets)
  Bc 19200                                Be 38400

Internet-Router# show policy-map interface serial 0/1
Serial0/1

Service-policy output: Internet

Class-map: Internet-traffic (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol ip
  Match: access-group 101
  Queueing
    Output Queue: Conversation 265
    Bandwidth 20 (%)
    Bandwidth 308 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
  Traffic Shaping
    Target/Average      Byte      Sustain      Excess      Interval      Increment
    Rate                Limit    bits/int    bits/int    (ms)         (bytes)
    2304000/768000      7200    19200      38400      25           7200

    Adapt Queue      Packets  Bytes      Packets  Bytes      Shaping
    Active Depth      0        0          Delayed  Delayed    Active
    -              0        0          0        0         no

Class-map: class-default (match-any)
  3 packets, 404 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

既然你已经看到使用 CBWFQ 给分类的流量增加流量整形的策略是多么容易，下面考虑如何在 CBWFQ 中实施流量监管。

6.9.2 基于类别的监管

当必须实施流量的策略时，当流量遵循、超过或者和某种速率冲突时，必须采取某种措施。你可能会考虑采用流量监管。*流量监管*允许用户基于用户定义的条件对进入或者输出的流量限制速率来实施流量的策略。可以通过分级映射和策略映射来定义流量的条件，并且将最终的流量服务策略绑定到接口上。也可以使用流量策略来实施一个最大的流量的速率来传输、丢弃或者标记数据包。

在本章较早的内容中，我们学习到关于流量整形和使用 CAR 的速率限制。在本小节中我们将学习如何使用流量监管来强制流量的速率，同样的原则也适用于流量整形和 CAR。使

用流量整形，例如，当发送的流量被整形时，它被缓存到接口的缓存区中。流量整形和流量监管都使用令牌桶算法。令牌以流量的速率填充到令牌桶中。为了传输一个数据包，在令牌桶中必须有足够的令牌。流量监管可以应用于接收和发送的流量，但是并不使用缓存来实施策略。使用流量整形，令牌只有在每个时间间隔内才会添加到令牌桶中，而使用流量监管，令牌总是会添加到令牌桶中。如果在令牌桶中没有足够的令牌，数据包就会被丢掉或者分类。流量监管不会将数据包放在队列中。流量监管不会在过量或者冲突的行为发生时，将令牌从令牌桶中清除。

当流量突发时，流量要么被丢掉，要么被标记。因为流量监管不会像流量整形那样支持缓存，流量监管丢弃超过接口带宽限制的数据包。这就是为什么流量监管支持对流量的分类的行为。也可以使用流量监管来标记数据包，通过修改服务质量值以备以后使用，例如 ATM CLP 位、Frame Relay DE 位、IP 优先级或者 DSCP 的值。当流量被标记后，通常在边界设备上，其他的服务质量方法，例如加权公平队列、WRED 或者流量整形，可以应用到下游的设备上。所以，如果一个接口有带宽可以转发一个突发数据包，并且流量策略允许它，这个数据包就会按照流量的策略进行转发。传输的突发数据包应当包括某种类型的行为，这个行为应当对突发的数据包设置丢弃位或者标记 TOS 来标记这个数据包。如果正常和过量的突发参数被正确配置的话，流量监管应当鼓励终端工作站在意识到数据包被丢弃的情况下，降低它们的 TCP 窗口的尺寸，就像 WRED 那样来防止全局的同步。

流量整形、CAR、流量监管三者之间的另一个不同点就是双令牌桶策略。在流量整形中，当你定义一个冲突策略时，你实际上是定义了第二个令牌桶，它将用于已经超出了正常和突发速率的那部分流量。

流量监管可以在一个策略映射中的策略映射类别配置模式下使用 **police** 语句。在思科 IOS 软件中有几种方法可以使用 **police** 命令来配置流量的限速。第一种方法，也就是这里所说的，同时输入所有的流量监管的参数，这是非常麻烦的。

```
police { rate-bps {[ normal-burst] [ excess-burst] } [bc normal-burst] [bc excess-burst] |
  cir rate-bps [ normal-burst] [ excess-burst] [bc normal-burst] [be excess-burst] | pir
  [ peak-rate] excess-burst} [conform- action { action | exceed-action} [exceed-action
  action [violate-action action]
```

另外一种方法，就是在策略映射策略配置模式下配置流量的策略，通过使用 **police** 命令来实现，正如这里所示：

```
police { rate-bps {[ normal-burst] [ excess-burst] } [bc normal-burst] [bc excess-burst] |
  cir rate-bps [ normal-burst] [ excess-burst] [bc normal-burst] [be excess-burst] | pir
  [peak-rate] excess-burst}}
```

当发出 **police** 命令后，就进入到策略映射策略配置模式下，可以看到 **Router (config-pmap-c-police) #**提示符。在这个模式下，可以发出或者挪去任何遵循、超出或者冲突的行为，一次一个，而无需输入较长的命令。遵循、超出或者冲突的行为如下所示：

```
conform-action {drop | set-clp-transmit | set-dscp-transmit dscp-value | set-frde-transmit
  | set-mls-exp-transmit mpls-experimental-value | set-prec-transmit precedence-value |
  set-qos-group qos-group-index | transmit}
exceed-action {drop | set-clp-transmit | set-dscp-transmit dscp-value | set-frde-transmit
  | set-mls-exp-transmit mpls-experimental-value | set-prec-transmit precedence-value |
  set-qos-group qos-group-index | transmit}
violate-action {drop | set-clp-transmit | set-dscp-transmit dscp-value | set-frde-transmit
  | set-mls-exp-transmit mpls-experimental-value | set-prec-transmit precedence-value |
  set-qos-group qos-group-index | transmit}
```

表 6-23 显示了 **police** 命令和策略映射策略配置模式命令的参数和它们的描述。

表 6-23 流量监管命令和描述

命令参数	描述
traffic-rate	平均的流量速率，在正常的情况下，在一段时间内，范围从 8000~2 000 000 000: $CIR = Tc/Bc$ in bit/s
normal-burst	(可选) 以字节表示的突发的速率。 范围从 1000~512 000 000: $Bc \text{ (in bytes)} = CIR \text{ (in bit/s)} * (1\text{byte}) / (8\text{bits}) * 1.5\text{seconds}$ 注意: 1.5 s 是一个平均往返的时间，如果平均往返的时间少于 1.5s，必须修改这个时间来准确地反映路由的路途时间
excess-burst	特别注明超额突发大小，范围从 1000 到 512 000 000: $Ber(\text{in byte})=Bc*2$
conform-action	(可选) 任何遵循正常速率的数据包将按照它所指定的行为工作
drop exceed-action set-clp-transmit set-frde-transmit set-dscp-transmit dscp-value set-mpls-exp-transmit mpls-experimental-value set prec-transmit precedence-value set qos-group qos-group-index transmit	指定遵循的行为 立刻丢弃数据包并且退出列表 忽略冗余的行为配置并且允许用户直接过度到超过的行为。当遵循和超过的行为相同的时候这样做 设置 ATM 的信元丢失的优先级 (CLP) 位和传输信元 设置帧中继的可丢弃位 (DE) 并且传输这个数据包 设置 DSCP 的值 (范围从 0~63)，并且传输这个数据包 设置 MPLS 的试验位 (范围从 0~7) 并且传输这个数据包 设置 IP 优先级的值 (范围从 0~7) 并且传输这个数据包 设置服务质量组的号码 (范围从 0~99) 并且传输这个数据包 传输这个数据包
[exceed-action {drop set-clp-transmit set-frde-transmit set-dscp-transmit dscp-value set-mpls-exp-transmit mpls-experimental-value set-prec-transmit precedence-value set-qos-group qos-group-index transmit}]	(可选) exceed-action 命令指定了当流量处于正常到过量突发范围 (Bc 到 Be) 内时对其所采取的行为。 exceed-action 命令可以通过执行的行为来完成
[violate-action { drop set-clp-transmit set-frde-transmit set-dscp-transmit dscp-value set-mpls-exp-transmit mpls-experimental-value set-prec-transmit precedence-value set-qos-group qos-group-index transmit}]	(可选) violate-action 命令指定了当流量超过最大的突发范围 (Be) 时所应采取的行为。 violate-action 命令可以通过执行的行为来完成

在流量的策略配置模式中需要四或者五个步骤 (取决于你决定使用命令的长格式还是较短的策略映射策略模式): 定义服务分类来指定流量的特性; 定义对流量分类所采取的行为方式; 将服务策略分配到一个接口中; 验证和监控配置。

第 1 步 使用 **class-map** 命令定义流量的分类。流量的分类用于定义策略所匹配的流量。在这个范例中，类别 **IP-traffic** 用于匹配所有的 IP 流量，而类别 **IPX-traffic** 用于匹配所有的 IPX 流量:

```
Simpson(config)# class-map IP-traffic
Simpson(config-cmap)# match protocol ip
Simpson(config-cmap)# exit
Simpson(config)# class-map IPX-traffic
Simpson(config-cmap)# match protocol ipx
Simpson(config-cmap)# exit
```

第 2 步 对服务策略配置定义一个策略，并且将类别分配到流量策略中。在这个范例中，策略 **WAN-traffic** 用于限制所有 IP 流量的速率为 512 kbit/s，具有 96 000 字节的突发大

小，使用思科推荐的 $Bc = CIR * (1\text{byte}) / (8\text{ bits}) * 1.5s$ 的公式。遵循这个策略的数据包被传输，超过这个策略的流量会被丢弃掉。同一类型的策略也用于配置类别 IPX-traffic 的所有 IPX 流量。

```
Simpson(config)# policy-map WAN-traffic
Simpson(config-pmap)# class IP-traffic
Simpson(config-pmap-c)# police 512000 96000 conform-action transmit exceed-
  action drop
Simpson(config-pmap-c)# exit
Simpson(config)# policy-map WAN-traffic
Simpson(config-pmap)# class IPX
Simpson(config-pmap-c)# police 512000 96000 conform-action transmit exceed-
  action drop
Simpson(config-pmap-c)# exit
Simpson(config-pmap)# exit
```

第 3 步 或者如果你使用的是模块化的策略映射策略配置模式的方法，你将使用 **police 512000 96000** 命令进入到策略映射策略配置模式中。接着输入在那个模式下遵循或者超出的行为方式，正如这里所示：

```
Simpson(config-pmap-c)#police 512000 96000
Simpson(config-pmap-c-police)#
Simpson(config-pmap-c-police)# conform-action transmit
Simpson(config-pmap-c-police)#exceed-action drop
Simpson(config-pmap-c-police)#exit
Simpson(config-pmap-c)#class IPX-traffic
Simpson(config-pmap-c)# police 512000 96000
Simpson(config-pmap-c-police)#
Simpson(config-pmap-c-police)# conform-action transmit
Simpson(config-pmap-c-police)#exceed-action drop
Simpson(config-pmap-c-police)#exit
Simpson(config-pmap-c)#exit
```

第 4 步 将这个策略映射作为一个服务策略分配到一个接口中去。

```
Simpson(config)#interface serial 0/1
Simpson(config-if)#service-policy output WAN-traffic
```

第 5 步 验证这个配置。为了验证并且监控流量的限速配置，使用 **show policy-map** 或者 **show policy-map interface** 命令。**show policy-map** 命令显示关于当前的流量策略配置的信息，而 **show policy-map interface** 命令显示的是关于当前流量策略状态的详细信息。

```
Simpson# show policy-map WAN-traffic
Policy Map WAN-traffic
  Class IP-traffic
    police cir 512000 bc 96000
      conform-action transmit
      exceed-action drop
  Class IPX-traffic
    police cir 512000 bc 96000
      conform-action transmit
      exceed-action drop
Simpson# show policy-map interface serial 0/1
Serial0/1

Service-policy output: WAN-traffic

Class-map: IP-traffic (match-all)
  6887 packets, 5241646 bytes
  5 minute offered rate 121000 bps, drop rate 75000 bps
    Match: protocol ip
  police:
    cir 512000 bps, bc 96000 bytes
    conformed 4351 packets, 1857386 bytes; actions:
      transmit
```



```

exceeded 2536 packets, 3384260 bytes; actions:
  drop
conformed 46000 bps, exceed 75000 bps

Class-map: IPX-traffic (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol ipx
police:
  cir 512000 bps, bc 96000 bytes
  conformed 0 packets, 0 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    drop
  conformed 0 bps, exceed 0 bps

Class-map: class-default (match-any)
  19 packets, 1428 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any

```

范例 6-32 显示了如何使用流量监管给不同类型的流量分配流量的策略。类别 management 使用访问控制列表 101 来指定 SNMP、DNS、DHCP、syslog 和 TFTP 的流量。类别 user-traffic 使用访问控制列表 102 来指定 NetBIOS 和 Telnet 的流量作为用户的流量。并且类别 internet 使用访问控制列表 103 来定义 HTTP 的 Web 流量和到达主机 10.1.1.141 的被动 FTP 流量作为 Internet 的流量。这些类别在策略 traffic-policy 下，对每一个类别都采用 **police** 命令来指定这个类别的流量策略。类别 management 被分配了 2Mbit/s 的速率限制，375 000 字节的正常突发和 750 000 字节的扩展突发。遵循正常流量速率的数据包被设置为 IP 优先级的值 Flash-override (4) 并且传输出去。当来自类别 management 的流量超出了过量突发速率时，它还是可以传输，但是这个数据包的 IP 优先级的值不再改变。来自类别 user-traffic 的流量遵循正常的流量速率 3 Mbit/s，具有正常的突发速率 562 500 字节和扩展突发速率 1 125 000 字节，将设置它的 IP 优先级的值为 Flash (3) 并且在超过正常速率的情况下依旧进行传输。来自类别 internet 的流量遵循速率限制为 5 Mbit/s，具有正常的突发速率 937 500 字节和过量突发速率 1 875 000 字节，将被传输，但是超出的流量部分被丢掉。

范例 6-32 使用流量监管来控制流量

```

class-map match-all management
  match access-group 101
class-map match-all internet
  match access-group 103
class-map match-all user-traffic
  match access-group 102
!
policy-map traffic-policy
  class management
    police cir 2000000 bc 375000 be 750000
      conform-action set-prec-transmit 4
      exceed-action transmit
  class user-traffic
    police cir 3000000 bc 562500 be 1125000
      conform-action set-prec-transmit 3
      exceed-action transmit
  class internet
    police cir 5000000 bc 937500 be 1875000

```

(待续)

```

conform-action transmit
exceed-action drop

!
interface Ethernet0/0
 ip address 10.1.1.101 255.255.255.0
 service-policy output traffic-policy
!
access-list 101 permit udp any any eq snmp
access-list 101 permit udp any any eq domain
access-list 101 permit tcp any any eq domain
access-list 101 permit udp any any eq bootps
access-list 101 permit udp any any eq bootpc
access-list 101 permit udp any any eq syslog
access-list 101 permit udp any any eq tftp
access-list 102 permit udp any any eq netbios-dgm
access-list 102 permit udp any any eq netbios-ns
access-list 102 permit udp any any eq netbios-ss
access-list 102 permit tcp any any eq telnet
access-list 103 permit tcp any any eq www
access-list 103 permit tcp any host 10.1.1.141 eq ftp
access-list 103 permit tcp any host 10.1.1.141 gt 1023 established

```

范例 6-33 显示了 **show policy-map** 命令和 **show policy-map interface** 命令是如何显示关于策略 **traffic-policy** 的信息的。

范例 6-33 使用 show policy-map 命令

```

Simpson# show policy-map traffic-policy
Policy Map traffic-policy
Class management
  police cir 2000000 bc 375000 be 750000
    conform-action set-prec-transmit 4
    exceed-action transmit
Class user-traffic
  police cir 3000000 bc 562500 be 1125000
    conform-action set-prec-transmit 3
    exceed-action transmit
Class internet
  police cir 5000000 bc 937500 be 1875000
    conform-action transmit
    exceed-action drop

Simpson# show policy-map interface ethernet 0/0
Ethernet0/0

Service-policy output: traffic-policy

Class-map: management (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 101
  police:
    cir 2000000 bps, bc 375000 bytes
    conformed 0 packets, 0 bytes; actions:
      set-prec-transmit 4
    exceeded 0 packets, 0 bytes; actions:
      transmit
    conformed 0 bps, exceed 0 bps

Class-map: user-traffic (match-all)

```

(待续)

```

0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group 102
police:
  cir 3000000 bps, bc 562500 bytes
  conformed 0 packets, 0 bytes; actions:
    set-prec-transmit 3
  exceeded 0 packets, 0 bytes; actions:
    transmit
  conformed 0 bps, exceed 0 bps

Class-map: internet (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group 103
police:
  cir 5000000 bps, bc 937500 bytes
  conformed 0 packets, 0 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    drop
  conformed 0 bps, exceed 0 bps

Class-map: class-default (match-any)
794 packets, 54247 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any
Simpson#

```

下一个范例，也就是范例 6-34，显示了本令牌桶的流量策略是如何基于流量类型和突发尺寸对不同的数据包设置 ToS 位的。范例 6-34 显示了类别 Servers 如何对所有到达网络 209.145.63.0/27 的流量设置流量策略。类别 apps 指定了所有使用 Telnet、SMTP 协议或者到达服务器 209.145.63.8 的被动 FTP 流量，而类别 Web 指定 HTTP 的 Web 流量。在这个范例中，属于类别 Servers 的流量遵循平均位速率 4 Mbit/s，具有 750 000 字节的正常突发和 1 500 000 字节的扩展突发，扩展突发将会把它的 DSCP 值改为 cs2，超过正常突发的流量将会把它的 DSCP 值改为 cs4，并且任何和过量突发速率冲突的 Servers 流量将会不改变 DSCP 的值进行传输。类别 apps 指定了到达 209.145.63.0/27 网段的服务器的流量将会有 3 Mbit/s 的平均速率、562 500 字节的正常突发和 1 125 000 字节的扩展突发。遵循于 apps 策略的流量会将它的 DSCP 值设置为 cs3，超过正常突发的流量会把它的 DSCP 值改为 cs4，和这个策略相冲突的流量将会传输，但是不会改变 DSCP 的数值。并且最终，属于类别 Web 的 Web 流量将会和 apps 类别具有相同的流量策略配置参数。但是，遵循、超出和冲突的行为是不一样的。在这种情况下，遵循 Web 策略的流量将会传输，但不改变其 DSCP 的值，超出过量和正常突发速率的流量将会被丢弃。使用这种类型的配置，在网络边界的设备可以指定 ToS 的值来对下游运行加权公平队列或者 WRED 的设备改变其服务质量的对待。通过改变 DSCP 的值，数据包的丢弃优先级被修改成一个更高的数值，降低了这些数据包被丢弃的可能性。

范例 6-35 显示了 **show policy-map policy1** 和 **show policy-map interface** 命令的输出。

有时候，整形和限速都不是解决问题的最好方案。在某些情况下，某些流量需要一个严格优先级的队列。下一小节显示如何使用低延迟队列来提供严格优先级的队列，就像用优先级队列建立的一样，在基于类别的队列设计中使用它。

范例 6-34 使用两个令牌桶流量策略

```

class-map match-all apps
  match access-group 102
class-map match-all Servers
  match access-group 101
class-map match-all web
  match access-group 103
!
policy-map policy1
  class Servers
    police cir 4000000 bc 750000 be 1500000
      conform-action set-dscp-transmit cs2
      exceed-action set-dscp-transmit cs4
      violate-action transmit
  class apps
    police cir 3000000 bc 562500 be 1125000
      conform-action set-dscp-transmit cs3
      exceed-action set-dscp-transmit cs4
      violate-action transmit
  class web
    police cir 3000000 bc 562500 be 1125000
      conform-action transmit
      exceed-action drop
!
interface Ethernet0/0
  ip address 10.1.1.111 255.255.255.0
  service-policy output policy1
!
access-list 101 permit ip any 209.145.63.0 0.0.0.31
access-list 102 permit tcp any any eq telnet
access-list 102 permit tcp any any eq smtp
access-list 102 permit tcp any host 209.145.63.8 eq ftp
access-list 102 permit tcp any host 209.145.63.8 gt 1023 established
access-list 103 permit tcp any any eq www

```

范例 6-35 双令牌桶的 show 命令

```

Simpson# show policy-map policy1
Policy Map policy1
  Class Servers
    police cir 4000000 bc 750000 be 1500000
      conform-action set-dscp-transmit cs2
      exceed-action set-dscp-transmit cs4
      violate-action transmit
  Class apps
    police cir 3000000 bc 562500 be 1125000
      conform-action set-dscp-transmit cs3
      exceed-action set-dscp-transmit cs4
      violate-action transmit
  Class web
    police cir 3000000 bc 562500 be 1125000
      conform-action transmit
      exceed-action drop
Simpson# show policy-map interface ethernet 0/0
Ethernet0/0

Service-policy output: policy1

```

(待续)

```

Class-map: Servers (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 101
  police:
    cir 4000000 bps, bc 750000 bytes, be 1500000 bytes
    conformed 0 packets, 0 bytes; actions:
      set-dscp-transmit cs2
    exceeded 0 packets, 0 bytes; actions:
      set-dscp-transmit cs4
    violated 0 packets, 0 bytes; actions:
      transmit
    conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map: apps (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 102
  police:
    cir 3000000 bps, bc 562500 bytes, be 1125000 bytes
    conformed 0 packets, 0 bytes; actions:
      set-dscp-transmit cs3
    exceeded 0 packets, 0 bytes; actions:
      set-dscp-transmit cs4
    violated 0 packets, 0 bytes; actions:
      transmit
    conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map: web (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 103
  police:
    cir 3000000 bps, bc 562500 bytes
    conformed 0 packets, 0 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps

Class-map: class-default (match-any)
  714 packets, 48821 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
    
```

6.9.3 低延迟队列 (LLQ)

低延迟队列 (LLQ) 也称为基于优先级的加权公平队列，使得在一个基于类别的策略下使用 CBWFQ 和模块化的服务质量 CLI 来严格优化流量的类别成为可能。

LLQ 允许来自至少一个类别策略中的流量被发送到严格优先级的队列中去，被称为优先级类别。使用 LLQ 比使用优先级队列或者 CBWFQ 有两个明显的好处。使用优先级队列，只要高优先级的队列中是满的，它就可能独占所有的带宽，导致其他的低优先级的流量永远没有被传输的机会。然而使用 LLQ，优先级队列的带宽被限制为一个用户可以定义的带宽。当这个限制达到后，后面的数据包就会被丢弃掉，直到有足够的资源可用。CBWFQ 公平地在它的类别之间分配流量，这可能潜在地对某些需要确保资源和低延迟低抖动的应

用程序带来很大问题。LLQ 通过建立一个高优先级的队列来解决这个问题，当配置合适时，可以防止抖动。

为了启用 LLQ，在策略类别的配置模式下使用 **priority** 命令。**policy** 命令有两个参数：**bandwidth** 和 **burst**。**bandwidth** 参数用于指定优先级队列的带宽限制。可选的 **burst** 参数指定流量的大小，以字节表示，允许突发到带宽的极限之上。

```
Simpson(config-pmap-c)# priority bandwidth [burst]
```

在正常的条件下，当没有拥塞时，严格优先级的流量不会受到带宽的限制，然而在发生拥塞的期间，当带宽的限制达到后，任何到达优先级队列的新数据包都会被丢弃。因为 LLQ 主要是为语音设计的，优先级的类别不支持使用 **random-detect** 命令，这是因为 WRED 并不对 UDP 的流量提供拥塞避免。当和 **priority** 命令使用时，也不支持 **bandwidth** 命令，这是因为 **priority** 命令有它自己的带宽的参数，并且这个优先级的类别也不使用对于流量监管的队列限制。**queue-limit** 命令在优先级的类别里也不支持。如果在优先级的类别里发出了不支持的命令，就会出现一个错误，警告用户在这个命令执行之前，必须清除严格优先级的队列。

在配置 LLQ 之前，非常重要的是要了解在严格优先级队列中所指定的流量需要多少带宽。LLQ 有一种流量计量的算法，当分配流量时会考虑二层的报头开销。然而，它并不补偿来自上游路由器的网络抖动、ATM 信元头，或者路由器产生的控制或者路由流量。如果带宽分配不是足够大以允许这种流量，数据包就会在不寻常的高流量期间或者突发时被丢弃掉。下面的列表显示了当和 CBWFQ 一起使用 LLQ 时要考虑的规则：

- 因为 LLQ 有它自己的带宽参数，当对流量进行限速时，不要在优先级的分类中使用 **bandwidth** 命令。
- 为了适当地支持无连接的语音流量，在优先级的类别中不支持 WRED。
- 因为 LLQ 使用带宽作为它的策略限制，在优先级的类别中不支持使用队列的限制。
- LLQ 在 VoIP on Frame Relay 上不支持。

为了演示在具有 FXS 端口的路由器上如何在 CBWFQ 中利用 LLQ 来实现 Voice over IP (VoIP)，图 6-10 显示了路由器 Albuquerque 和路由器 Santa Fe 如何通过一个串行的 HDLC 点对点链路连接起来。连接到路由器 Albuquerque 的电话号码是 4567，而连接到路由器 Santa Fe 的电话号码是 7879。以后，在发生网络拥塞的时候，来自 Albuquerque 路由器的语音呼叫的质量非常差。为了解决这个问题，在 Albuquerque 路由器上实施了 LLQ。因为路由器 Albuquerque 正在使用语音的编解码 g729r8，因此可以知道在这个接口上使用的优先级队列只需要最大 30 kb 的带宽。

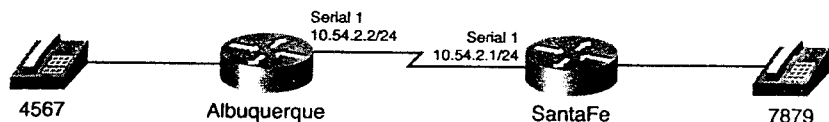


图 6-10 对语音流量使用 LLQ

范例 6-36 显示了如何建立一个服务策略来优化所有的语音流量，同时还提供可接受的数据流量的吞吐。

范例 6-36 对 Voice over IP 使用 LLQ

```

hostname Albuquerque
!
class-map data
  match protocol ip
class-map voice
  match access-group 101
!
policy-map voice-data
  class voice
    priority 30
  class data
    bandwidth 1125
    random-detect
!
dial-peer voice 4567 pots
  destination-pattern 4567
  port 2/0
!
dial-peer voice 7879 voip
  destination-pattern 7879
  session target ipv4:10.54.2.1
!
interface Serial1
  ip address 10.54.2.2 255.255.255.0
  service-policy output voice-data
!
access-list 101 permit udp any any range 16384 32767
access-list 101 permit tcp any any eq 1720

```

类别 voice 匹配所有的 VoIP 端口，由访问控制列表 101 定义，而类别 data 匹配所有的 IP 协议。策略映射 voice-data 将类别 voice 分配到一个严格优先级的分类中，使用的是 30 kb 的带宽，包括了路由器抖动和控制流量的空间，而类别 data 限制到 1125 kb 的带宽并且将使用 WRED 在拥塞发生期间来预先丢弃数据包。范例 6-37 显示了 **show policy-map** 命令的输出，它显示了策略配置的汇总信息，并且解释它如何和通过这个网络发送的流量工作。

范例 6-37 show policy-map 命令的输出

```

Albuquerque# show policy-map voice-data
Policy Map voice-data
  Weighted Fair Queueing
    Class voice
      Strict Priority
      Bandwidth 30 (kbps)
    Class data
      Bandwidth 1125 (kbps)
      exponential weight 9
      class      min-threshold      max-threshold      mark-probability
      .....
      0          -                  -                  1/10
      1          -                  -                  1/10
      2          -                  -                  1/10
      3          -                  -                  1/10
      4          -                  -                  1/10
      5          -                  -                  1/10
      6          -                  -                  1/10
      7          -                  -                  1/10

```

(待续)

```

rsvp - - - - - 1/10
Albuquerque# show policy-map interface serial 1
Serial1 output : voice-data
  Weighted Fair Queueing
    Class voice
      Strict Priority
      Output Queue: Conversation 264
        Bandwidth 30 (kbps) Packets Matched 152
        (total drops/bytes drops) 0/0
    Class data
      Output Queue: Conversation 265
        Bandwidth 1125 (kbps) Packets Matched 48
        (depth/total drops/no-buffer drops) 0/0/0
        exponential weight: 9
        mean queue depth: 0
    drops: class random tail min-th max-th mark prob
           0      0      0    20    40    1/10
           1      0      0    22    40    1/10
           2      0      0    24    40    1/10
           3      0      0    26    40    1/10
           4      0      0    28    40    1/10
           5      0      0    30    40    1/10
           6      0      0    32    40    1/10
           7      0      0    34    40    1/10
           rsvp   0      0    36    40    1/10
```

就像你所看到的，CBWFQ 可以执行一系列的服务质量技术。当你看到 CBWFQ 可以使用的一系列方法后，你可能想象到在网络中使用这种技术的一系列方法，例如下面的：

- 对于策略实施，标记流量。
- 将流量分类，放入到策略组中。
- 使用加权公平队列或者优先级队列的技术，把某种类型的流量放入到队列中。
- 执行尾部丢弃或者 WRED，这取决于流量的类型。
- 优化流量，预留带宽。
- 整形流量。
- 通过对流量监管，强制流量的策略。

就像你可以想象到的那样，这 3 章可以很容易地扩充成一本 1000 页以上的书。最好的测试和应用这些服务质量技术的方法就是在实验的环境下使用测试流量，然后在完成测试后，在生产性的环境下应用这些服务质量解决方案。掌握了这些服务质量章节中某些创造性和技巧性的指示，你就可以建立非常灵活的服务质量方案。

6.10 练习场景

6.10.1 实验 12：定制队列

律师行 Blackerby、Smith 和 Heitz 通常被称为 BSH，它有一个网络，由总部在 Orlando 的所有服务器和 PBX 组成。它们当前有两个分支站点：Columbia 和 Atlanta。然而，在下两个月中，它们计划增加两个新的站点：一个在 Birmingham，另一个在 Greensboro，如图 6-11 所示。

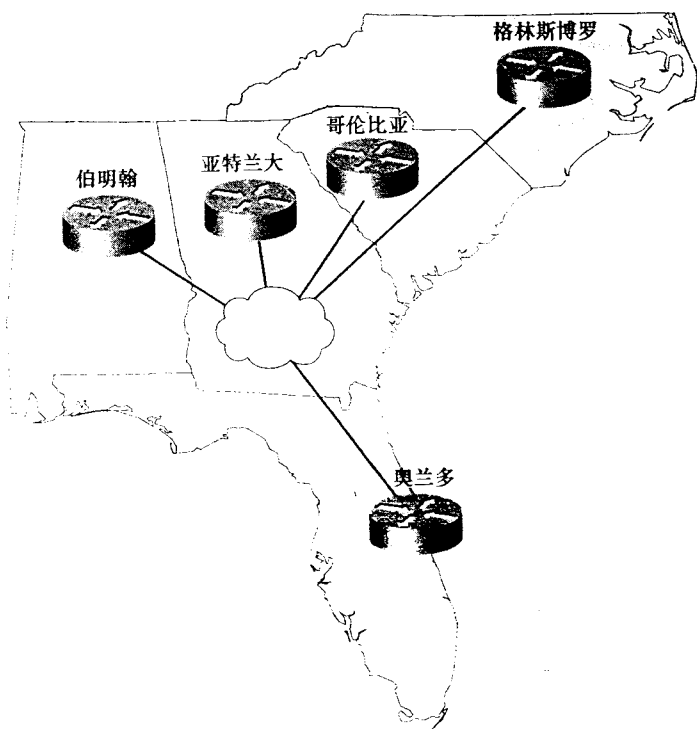


图 6-11 Blackerby、Smith 和 Heitz 网络图

一、需要的设备

这个实验需要下面的设备：

- 具有一个串口、一个以太接口和两个 FXS 语音端口的 3 台路由器；
- 具有 4 个串口充当帧中继交换机的路由器；
- （可选）具有以太接口的两台计算机；
- （可选）具有一个以太接口的额外的路由器。

这个实验的核心需要 4 台路由器，其中 3 台路由器需要一个串口，一台充当帧中继交换机的路由器需要 4 个串口。路由器应当按照图 6-12 所示连接它们的串口。

二、物理布局和预规划

- 按照图 6-13 所示的那样配置帧中继，使用表 6-24 所示的 IP 地址和 DLCI 分配。

表 6-24 IP 地址和帧中继的 DLCI 分配

路由器的接口	DLCI	IP 地址	路由器的接口	DLCI	IP 地址
Atlanta Serial 0/2	201	192.168.2.2/30	Orlando Serial 1.102	102	192.168.2.1/30
Columbia Serial 0	301	192.168.3.2/30	Orlando Serial 1.103	103	192.168.3.1/30

- 按照图 6-13 所示，配置帧中继交换机，使用表 6-25 所示的 DLCI 分配。

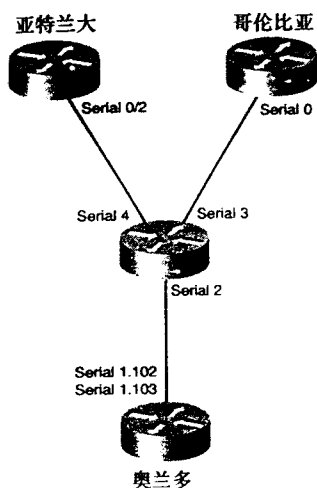


图 6-12 物理实验配置

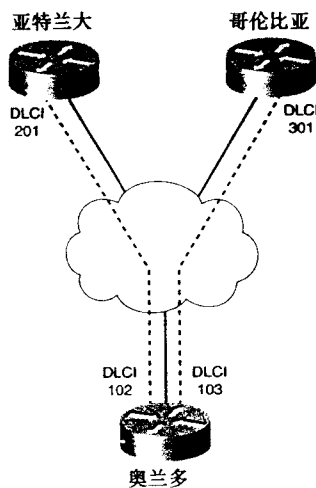


图 6-13 帧中继的 DLCI 配置

表 6-25

帧中继交换机 DLCI 分配

本地接口	本地 DLCI	远程接口	远程 DLCI	本地接口	本地 DLCI	远程接口	远程 DLCI
Serial 4	201	Serial 2	102	Serial 2	102	Serial 4	201
Serial 3	301	Serial 2	103	Serial 2	103	Serial 3	301

6.10.2 实验目的

每一个站点都有几台计算机，本地办公室的员工通过它们来访问总部在 Orlando 的文件服务器和应用程序。每一个站点也有两部电话，它们是用于拨打总部的电话。从分支办公室到总部 Orlando 站点的电话通常是在一天的不同时刻不经常呼叫，但是两部电话很少同时使用。在 Orlando 和 Atlanta 站点之间的所有的语音和数据流量是通过 CIR 为 256kbit/s 的帧中继电路传输的。总部当前有 T1 的帧中继，具有 768kbit/s CIR。当前的流量类型最近作了分析，发现在峰值时刻，上午 9: 00~10: 30，下午 12: 00~1: 00，3: 30~5: 00，流量最繁忙，而有些应用程序对繁忙时刻所导致的延迟不能够容忍。当对两个新的站点实施网络升级时，已经决定把位于 Orlando 的帧中继电路的 CIR 升级到 1.544 Mbit/s。这在流量繁忙的时刻可以缓解问题。为了防止新的问题，已经决定在升级之前实施定制队列，只在 Orlando 的站点上去做。这个实验的目的包括下面这些：

- 在 Orlando 和 Columbia 路由器的 FXS 卡之间配置 VoIP。
- 配置定制队列来基于字节的计数支持流量的限制。

6.10.3 实验任务

第 1 步 配置 Orlando 路由器连接到 Atlanta 和 Columbus 路由器，而不要使用 **frame-relay map** 语句。同样，配置 Atlanta 和 Columbus 路由器连通 Orlando 路由器。此时，

所有的路由器应当在线路和协议上都是 up 状态。

第 2 步 按照图 6-14 所示,对每一台路由器配置 IP 地址。配置所有的网络属于 OSPF area 0, 并且验证 IP 的连通性。

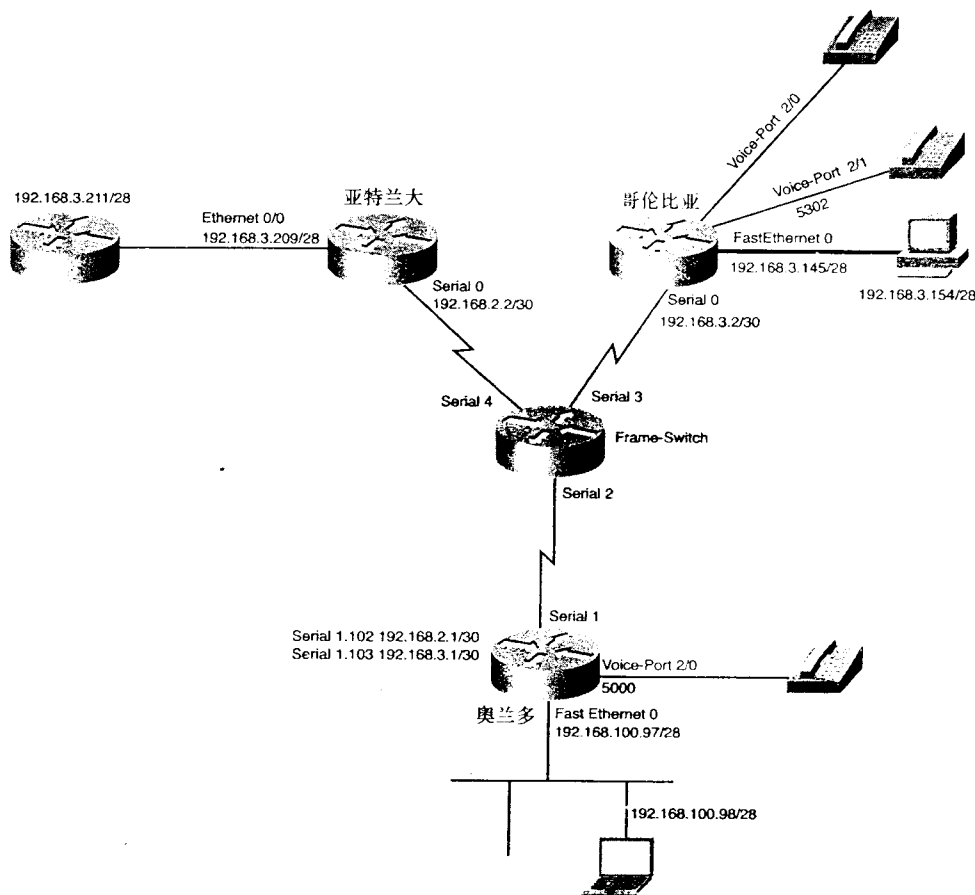


图 6-14 完整的网络图

第 3 步 如果可能,在 Columbia 和 Orlando 路由器之间配置 VoIP,如图 6-14 所示。在 Columbia 路由器上配置其中的一部电话的号码是 5301,另外一部电话的号码是 5302,配置 Orlando 站点的一部电话使用的号码是 5000。验证所有的电话彼此可以呼叫对方。

第 4 步 配置两台计算机,一台在 192.168.3.144 网络上,另外一台在 192.168.100.96 网络上。

第 5 步 将另外一台路由器放在 192.168.3.208 网络上。配置这台路由器使用一条默认路由到达 Atlanta 路由器的 192.168.3.209 接口。配置这台路由器允许远程登录的访问。

第 6 步 验证在网络 192.168.3.144 和 192.168.100.96 上的主机可以远程登录到路由器 192.168.3.211。

第 7 步 使用表 6-26 中的信息和本章前面的“定制队列”部分的公式,在表 6-27 中输

入 byte-count 的数据。来自 Byte Count 字段的信息将被用于配置 byte-count 的队列限制。

表 6-26

这个实验的带宽百分比

队列号码	协议	带宽的百分比	平均数据包尺寸
1	语音	25	64
2	DHCP, DNS, SNMP	5	79
3	Telnet (远程登录)	10	64
4	SMTP	10	625
5	到达 192.168.3.211 的被动 FTP	10	315
6	WWW	5	1024
7	其他	25	1042

表 6-27

字节计数限制的数据

协议	带宽分配	平均数据包尺寸	比率	正常化的比率	整个比率	字节计数
语音	25%	64				
DHCP, DNS, SNMP	5%	79				
Telnet	10%	64				
SMTP	10%	625				
到 192.168.3.211 的被动的 FTP	10%	315				
WWW	5%	1024				
其他的	25%	1042				

第 8 步 对 Orlando 路由器的串口配置定制队列，对于每一个队列，使用表 6-27 所示的字节计数的限制。配置任何所需的访问控制列表来将流量排序放入到队列中。

第 9 步 确保队列 7 对于所有的未指定流量是默认的队列。

6.10.4 实验步骤

配置帧中继交换机使得它有 DLCI 201。它应当和 DLCI 102 和 DLCI 301 匹配，还应当匹配 103。范例 6-38 显示了帧中继交换机的完整配置。

范例 6-38 帧中继交换机的配置

```
hostname Frame-Relay
!
frame-relay switching
!
interface Serial2
description Connection to Orlando
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 102 interface Serial4 201
frame-relay route 103 interface Serial3 301
!
interface Serial3
```

(待续)

```

description Connection to Columbia
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 301 interface Serial2 103
!
interface Serial4
description Connection to Atlanta
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 201 interface Serial2 102
!
end

```

在这个范例中，注意接口 serial 2 上的 **frame-relay route** 语句都有本地 DLCI 的号码，DLCI 102 和 103。这些号码用于匹配分配给 Orlando 路由器的 DLCI 号码。另外两个 DLCI 号码，即 201 和 203，分配给接口 3 和 4，它们连接到 Atlanta 和 Columbus 路由器。范例 6-39 显示了来自帧中继交换机的帧中继路由表。

范例 6-39 来自帧中继交换机的帧中继路由表

```

Frame-Switch# show frame route
Input Intf Input DlcI Output Intf Output DlcI Status
Serial2      102 Serial4      201 active
Serial2      103 Serial3      301 active
Serial3      301 Serial2      103 active
Serial4      201 Serial2      102 active

```

第 1 步 配置 Orlando 路由器连接到 Atlanta 和 Columbus 路由器，而不要使用 **frame-relay map** 语句。同样，配置 Atlanta 和 Columbia 路由器连通 Orlando 路由器。此时，所有的路由器应当在线路和协议上都是 up 状态。

为了配置从 Orlando 路由器到达 Atlanta 和 Columbus 路由器的二层连接，而不使用 **frame-relay map** 语句，在 Orlando 路由器的串行接口上使用子接口。

```

Orlando(config)# interface Serial1
Orlando(config-if)# encapsulation frame-relay
Orlando(config-if)# clockrate 1300000
Orlando(config-if)# interface Serial0.102 point-to-point
Orlando(config-if)# frame-relay interface-dlci 102
Orlando(config-if)# interface Serial0.103 point-to-point
Orlando(config-if)# frame-relay interface-dlci 103

```

另外两台路由器只需要 **encapsulation frame-relay** 命令，如果它们在连接的数据电路终止点（DCE）上，还需要配置一个时钟速率。

```

Atlanta(config-if)# int s 0/2
Atlanta(config-if)# encapsulation frame-relay
Atlanta(config-if)# clockrate 1300000
Columbia(config-if)# int s 0
Columbia(config-if)# encapsulation frame-relay
Columbia(config-if)# clockrate 1300000

```

第 2 步 按照图 6-14 所示，对每一台路由器配置 IP 地址。配置所有的网络都属于 OSPF area 0，并且验证 IP 的连通性。

为了建立从 Atlanta 和 Columbus 路由器到 Orlando 路由器的 IP 连通性，下面的地址被分配了。确保对 OSPF 的连接使用 **ip ospf network point-to-point** 命令。下面的范例显示了对于 Orlando 子接口的帧中继接口配置。

```
Orlando(config)# interface Serial1.102 point-to-point
Orlando(config-if)# ip address 192.168.2.1 255.255.255.252
Orlando(config-if)# ip ospf network point-to-point

Orlando(config)# interface Serial1.103 point-to-point
Orlando(config-if)# ip address 192.168.3.1 255.255.255.252
Orlando(config-if)# ip ospf network point-to-point
```

Atlanta 和 Columbia 路由器允许使用 **frame-relay map** 语句。当所有的 IP 地址被分配后，每一台路由器将需要 OSPF 的配置。下面的范例显示了对于 Atlanta 和 Columbia 路由器的帧中继接口配置。

```
Atlanta(config)# int s 0/2
Atlanta(config-if)# ip address 192.168.2.2 255.255.255.252
Atlanta(config-if)# frame-relay map ip 192.168.2.1 201 broadcast
Atlanta(config-if)# ip ospf network point-to-point

Columbia(config)# int s 0
Columbia(config-if)# ip address 192.168.3.2 255.255.255.252
Columbia(config-if)# frame-relay map ip 192.168.3.1 301 broadcast
Columbia(config-if)# ip ospf network point-to-point
```

第 3 步 如果可能，在 Columbia 和 Orlando 路由器之间配置 VoIP，如图 6-14 所示。在 Columbia 路由器上配置其中的一部电话的号码是 5301，另外一部电话的号码是 5302，配置 Orlando 站点的一部电话使用的号码是 5000。验证所有的电话彼此可以呼叫对方。

如果你有两台具有语音能力的带有 FXS 卡的路由器，需要完成这个步骤，你需要在每一台路由器上建立两个 dial-peer。一个 **dial-peer** 语句将用于端口。这个语句应当指定目的的形式，这是电话的本地号码，而这个端口也是本地连接到语音端口的端口。另外一个 **dial-peer** 语句是 **voip** 语句，它指定了 VoIP 使用的远端的电话号码和 IP 地址。

```
Orlando(config)#dial-peer voice 5000 pots
Orlando (config-dial-peer)# destination-pattern 5000
Orlando (config-dial-peer)# port 2/0
Orlando (config-dial-peer)#dial-peer voice 5301 voip
Orlando (config-dial-peer)# destination-pattern 5301
Orlando (config-dial-peer)# session target ipv4:192.168.3.2
Orlando (config-dial-peer)#dial-peer voice 5302 voip
Orlando (config-dial-peer)# destination-pattern 5302
Orlando (config-dial-peer)# session target ipv4:192.168.3.2
Columbia(config)#dial-peer voice 5301 pots
Columbia (config-dial-peer)# destination-pattern 5301
Columbia (config-dial-peer)# port 2/0
Columbia (config-dial-peer)#dial-peer voice 5302 pots
Columbia (config-dial-peer)# destination-pattern 5302
Columbia (config-dial-peer)# port 2/1
Columbia (config-dial-peer)#dial-peer voice 5000 voip
Columbia (config-dial-peer)# destination-pattern 5000
Columbia (config-dial-peer)# session target ipv4:192.168.3.1
```

第 4 步 配置两台计算机，一个在 192.168.3.144 网络上，另外一个在 192.168.100.96 网络上。

如果你有两台额外的计算机，一台放在 192.168.3.144 网络上，另外一台放在 192.168.100.96 网络上。

第 5 步 将另外一台路由器放在 192.168.3.208 网络上。配置这台路由器使用一条默认路

由到达 Atlanta 路由器的 192.168.3.209 接口。配置这台路由器允许远程登录的访问。

第4台路由器应当只需要在它的以太网接口上有一个 IP 地址和一条到达 192.168.3.209 的默认路由。

```
Router(config)# interface Ethernet0
Router(config-if)# ip address 192.168.3.211 255.255.255.240
Router(config)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.209
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#pass cisco
```

第6步 验证在网络 192.168.3.144 和 192.168.100.96 上的主机能够远程登录到路由器 192.168.3.211。

如果你能够完成第4步，那么应当能够验证在网络 192.168.3.144 和 192.168.100.96 上的主机能够 ping 彼此。如果你能够成功地完成第5步，这两台主机也都能够远程登录到 192.168.3.208 网络上的路由器。

第7步 使用表 6-26 的信息和本章前面的公式，输入表 6-27 所示的字节计数的数据。来自 Byte Count 字段的信息将用于配置字节计数队列的限制。表 6-28 显示了这个实验的字节计数的尺寸。

表 6-28

定制队列的字节计数的尺寸

协议	带宽分配	平均数据包尺寸	比率	正常化的比率	整个比率	字节计数	实际带宽
语音	25%	64	0.3906	79.7	80	5120	26.8%
DHCP, DNS, SNMP	5%	79	0.0633	12.9	13	1027	5.3%
Telnet	10%	64	0.1563	31.9	32	2048	10.7%
SMTP	10%	625	0.016	3.3	4	2500	13%
到 192.168.3.211 的被动 FTP	10%	315	0.0317	6.5	7	2205	11.5%
WWW	5%	1024	0.0049	1	1	1024	5.3%
其他	25%	1042	0.0240	4.9	5	5210	27.2%
						19 314	

第8步 对 Orlando 路由器的串行接口配置定制队列。对每一个队列使用表 6-27 中字节计数的限制。配置任何所需的访问控制列表来将流量排序放入到队列中。

对于这个试验，访问控制列表 101 用于指定语音流量；访问控制列表 102 用于指定 DHCP、DNS 和 SNMP 的流量。访问控制列表 103 用于指定 FTP 的流量。这些访问控制列表都和 queue list 1 使用来指定流量的类型和对于每一个队列的字节中计数。这个 queue list 是使用 custom-queue-list 命令绑定到接口 serial 1 上的。

```
Orlando(config)#access-list 101 permit tcp any any eq 1720
Orlando(config)#access-list 101 permit udp any any range 16384 32767
Orlando(config)#access-list 101 remark Voice traffic
Orlando(config)#access-list 102 remark DHCP, DNS and SNMP traffic
Orlando(config)#access-list 102 permit udp any any eq bootpc
Orlando(config)#access-list 102 permit udp any any eq domain
Orlando(config)#access-list 102 permit tcp any any eq domain
Orlando(config)#access-list 102 permit udp any any eq snmp
Orlando(config)#access-list 103 remark FTP and random port for data
```

```

Orlando(config)#access-list 103 permit tcp any host 192.168.3.211 eq ftp
Orlando(config)#access-list 103 permit tcp any host 192.168.3.211 gt 1023
established
Orlando(config)#queue-list 1 protocol ip 1 list 101
Orlando(config)#queue-list 1 protocol ip 2 list 102
Orlando(config)#queue-list 1 protocol ip 3 tcp telnet
Orlando(config)#queue-list 1 protocol ip 4 tcp smtp
Orlando(config)#queue-list 1 protocol ip 5 list 103
Orlando(config)#queue-list 1 protocol ip 6 tcp www
Orlando(config)#queue-list 1 protocol ip 7
Orlando(config)#queue-list 1 queue 1 byte-count 5120
Orlando(config)#queue-list 1 queue 2 byte-count 1027
Orlando(config)#queue-list 1 queue 3 byte-count 2048
Orlando(config)#queue-list 1 queue 4 byte-count 2500
Orlando(config)#queue-list 1 queue 5 byte-count 2205
Orlando(config)#queue-list 1 queue 6 byte-count 1024
Orlando(config)#queue-list 1 queue 7 byte-count 5210
Orlando(config)#interface Serial1
Orlando(config-if)#custom-queue-list 1

```

第9步 使得 Queue 1 成为所有未知流量的默认的队列。为了使得 Queue 7 成为默认的队列，只需要使用 **queue-list** 命令来指定 Queue 7 作为它的默认队列。

```
queue-list 1 default 7
```

范例 6-40 Orlando 路由器的配置

```

hostname Orlando
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer voice 5000 pots
destination-pattern 5000
port 2/0
!
dial-peer voice 5301 voip
destination-pattern 5301
session target ipv4:192.168.3.2
!
dial-peer voice 5302 voip
destination-pattern 5302
session target ipv4:192.168.3.2
!
interface Serial1
no ip address
encapsulation frame-relay
custom-queue-list 1
clockrate 1300000
!
interface Serial1.102 point-to-point
ip address 192.168.2.1 255.255.255.252
ip ospf network point-to-point
frame-relay interface-dlci 102
!
interface Serial1.103 point-to-point
ip address 192.168.3.1 255.255.255.252
ip ospf network point-to-point
frame-relay interface-dlci 103
!

```

(待续)


```

interface FastEthernet0
 ip address 192.168.100.97 255.255.255.240
!
router ospf 101
 network 192.168.2.0 0.0.0.3 area 0
 network 192.168.3.0 0.0.0.3 area 0
 network 192.168.100.96 0.0.0.15 area 0
!
access-list 101 permit tcp any any eq 1720
access-list 101 permit udp any any range 16384 32767
access-list 101 remark Voice traffic
access-list 102 remark DHCP, DNS and SNMP traffic
access-list 102 permit udp any any eq bootpc
access-list 102 permit udp any any eq domain
access-list 102 permit tcp any any eq domain
access-list 102 permit udp any any eq snmp
access-list 103 remark FTP and random port for data
access-list 103 permit tcp any host 192.168.3.211 eq ftp
access-list 103 permit tcp any host 192.168.3.211 gt 1023 established
queue-list 1 protocol ip 1 list 101
queue-list 1 protocol ip 2 list 102
queue-list 1 protocol ip 3 tcp telnet
queue-list 1 protocol ip 4 tcp smtp
queue-list 1 protocol ip 5 list 103
queue-list 1 protocol ip 6 tcp www
queue-list 1 protocol ip 7
queue-list 1 default 7
queue-list 1 queue 1 byte-count 5120
queue-list 1 queue 2 byte-count 1027
queue-list 1 queue 3 byte-count 2048
queue-list 1 queue 4 byte-count 2500
queue-list 1 queue 5 byte-count 2205
queue-list 1 queue 6 byte-count 1024
queue-list 1 queue 7 byte-count 5210
!

```

范例 6-41 显示了 **show interface** 和 **show queueing** 命令的输出。注意 **show interface** 命令显示定制队列已经启用了，当前在队列中没有数据包。**show queueing** 命令的输出用于显示在这个实验中定制队列的信息。

范例 6-41 在 Orlando 路由器上 **show interface** 和 **show queueing** 命令的输出

```

Orlando# show interface serial 1
Serial0 is up, line protocol is up
Hardware is PowerQUICC Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 42/255, rxload i/255
Encapsulation FRAME-RELAY, loopback not set
Keepalive set (10 sec)
LMI enq sent 604, LMI stat recvd 597, LMI upd recvd 0, DTE LMI up
LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
LMI DLCI 0 LMI type is ANSI Annex D frame relay DTE
FR SVC disabled, LAPF state down
Broadcast queue 0/64, broadcasts sent/dropped 1431/3, interface broadcasts 1224
Last input 00:00:05, output 00:00:05, output hang never
Last clearing of "show interface" counters 01:47:08
Input queue: 0/75/2/0 (size/max/drops/flushes); Total output drops: 33540
Queueing strategy: custom-list 1
Output queues: (queue #: size/max/drops)

```

(待续)

```

0: 0/20/0 1: 0/20/0 2: 0/20/0 3: 0/20/0 4: 0/20/0
5: 0/20/0 6: 0/20/0 7: 0/20/33540 8: 0/20/0 9: 0/20/0
10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0
15: 0/20/0 16: 0/20/0
5 minute input rate 4000 bits/sec, 25 packets/sec
5 minute output rate 259000 bits/sec, 27 packets/sec
  14023 packets input, 884229 bytes, 0 no buffer
Received 0 broadcasts, 0 runs, 0 giants, 0 throttles
  1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored, 0 abort
 14672 packets output, 16220918 bytes, 0 underruns
   0 output errors, 0 collisions, 4 interface resets
   0 output buffer failures, 0 output buffers swapped out
   15 carrier transitions
 DCD=up DSR=up DTR=up RTS=up CTS=up
Orlando# show queueing
Current fair queue configuration:
Current priority queue configuration:
Current custom queue configuration:
List Queue Args
1      7      default
1      1      protocol ip          list 101
1      2      protocol ip          list 102
1      3      protocol ip          tcp port telnet
1      4      protocol ip          tcp port smtp
1      5      protocol ip          list 103
1      6      protocol ip          tcp port www
1      7      protocol ip
1      1      byte-count 5120
1      2      byte-count 1027
1      3      byte-count 2048
1      4      byte-count 2500
1      5      byte-count 2205
1      6      byte-count 1024
1      7      byte-count 5210
Current random-detect configuration:

```

范例 6-42 显示了对 Atlanta 路由器的完整配置，而范例 6-43 显示了对 Columbia 路由器的完整配置。

范例 6-42 Atlanta 路由器的配置

```

hostname Atlanta
!
interface Ethernet0/0
 ip address 192.168.2.209 255.255.255.240
!
interface Serial0/2
 ip address 192.168.2.2 255.255.255.252
 encapsulation frame-relay
 ip ospf network point-to-point
 clockrate 1300000
 frame-relay map ip 192.168.2.1 201 broadcast
!
router ospf 101
 network 192.168.2.0 0.0.0.3 area 0
 network 192.168.2.208 0.0.0.15 area 0
!

```

范例 6-43 Columbia 路由器的配置

```
hostname Columbia
!
voice-port 2/0
!
voice-port 2/1
!

dial-peer voice 5301 pots
 destination-pattern 5301
 port 2/0
!
dial-peer voice 5302 pots
 destination-pattern 5302
 port 2/1
!
dial-peer voice 5000 voip
 destination-pattern 5000
 session target ipv4:192.168.3.1
!
interface Serial0
 ip address 192.168.3.2 255.255.255.252
 encapsulation frame-relay
 ip ospf network point-to-point
 clockrate 1300000
 frame-relay map ip 192.168.3.1 301 broadcast
!
interface FastEthernet0
 ip address 192.168.3.145 255.255.255.240
!
router ospf 101
 network 192.168.3.0 0.0.0.3 area 0
 network 192.168.3.144 0.0.0.15 area 0
```

6.11 实验 13：使用 CBWFQ 和 NBAR 管理

因特网的流量

在这个实验中，你可以将学习到的知识应用于实际网络的服务质量环境下。这个实验应用 CBWFQ 来解决常见的因特网问题：用户使用企业级的网络来做个人娱乐目的。

一、实验练习

在这个实验中，NBAR 可以指定某种类型的数据，并且将类别和服务策略应用到一起来实施适当的因特网利用。这个场景包括下面的这些技术：

- 使用 NBAR 来分类流量。
- 使用 DSCP 位来标记流量。
- 配置 ATM 服务质量。
- 使用带宽预留来优化流量。
- 选择性地对某种类型的流量应用尾部丢弃、WRED 和加权公平队列。
- 选择适当的队列和交换类型。

二、实验的目的

本实验的目的就是应用到目前为止所有的服务质量技术，对图 6-15 所示的网络模型，应用一个因特网的服务策略。

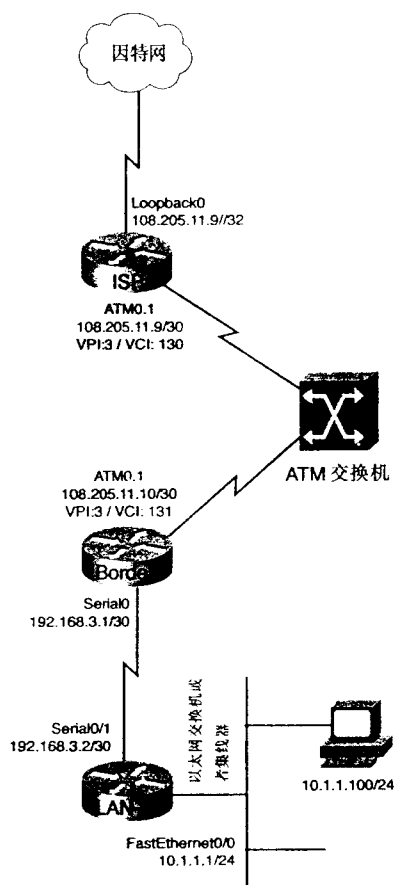


图 6-15 因特网边界的模型

三、需要的设备

- 具有一个 OC-3 ATM 接口的思科路由器。
- 具有一个 OC-3 ATM 接口和一个串行接口的思科路由器。
- 具有两个 OC-3 ATM 接口的 LightStream ATM 交换机。
- 具有一个以太网端口和一个串行端口的思科路由器。
- 具有运行 TCP/IP 协议的以太网 NIC 的 PC 机。
- 一个以太网交换机或者集线器。

四、物理布局和预规划

- 按照图 6-15 所示，对路由器进行布线。

- 将 PC 连接到以太网交换机或者集线器上，并且将它配置在 10.1.1.0/24 的网络上。
- 配置 ATM 交换机使用图 6-15 所示的 PVC 信息。
- 除了在 Border 和 ISP 路由器上的 ATM 接口，给路由器的每一个接口配置 IP 地址，并且验证路由器可以 ping 通它们直接相连的邻居。
- 验证所有的接口都处于 up/up 状态。

五、实验的任务

- 第 1 步** 在 ISP 和 Border 路由器上配置 ATM PVC。给 ISP 路由器的 ATM 0.1 接口分配 VPI: 3 和 VCI 130，给 Border 路由器的 ATM 0.1 接口分配 VPI: 3 VCI: 131。配置这些路由器使用 VBR-nrt，即 45Mbit/s 的持续信元速率和 50Mbit/s 的峰值信元速率。验证每一台路由器彼此可以 ping 通对方。
- 第 2 步** 除了 ISP 路由器，将所有的其他路由器分配到 EIGRP AS 148。在它们的真正的网络边界上汇总这些路由；不要使用有类的汇总。在 Border 路由器上重分发指到 ISP 路由器的默认路由。在进行第 3 步之前，验证所有的路由器，能够 ping 通所有其他的路由器。
- 第 3 步** 从 Border 路由器上，配置网络地址翻译（NAT），使得所有的内部网络 10.1.1.0/24 和 192.168.0.0/16 能够连到超出 ISP 路由器的因特网上，而无需任何额外的路由。验证主机 PC 能够连通接口地址为 108.205.11.9/32 的 ISP 路由器。
- 第 4 步** 对 LAN-rtr 上发送方向的接口配置一个策略。这个策略应当匹配表 6-29 所示的可变量。
- 第 5 步** 在 Border 路由器的发送方向的 ATM 接口上启用 DSCP WRED。这是第 4 步所建立的策略的实施。

表 6-29

策略的配置

类的名字	流量的类型	策略
High-Pri_Internet	到达 cisco.com 的 HTTP 流量	保留 15% 的带宽 给每一个包标记上 EF DSCP 的值
Med-Pri-Internet	其他的 HTTP 和 SSH 流量	保留 55% 的带宽 给每一个包标记上 CS3 DSCP 的值
Low-Pri-Internet	FTP、Telnet、SFTP、HTTPS 和安全的 POP3	保留 5% 的带宽，应用 WRED，而不是尾部丢弃 给每一个包标记上 CS1 DSCP 的值
No-Pri_Internet	Gnutella、MS NetShow、Napster、NNTP、Real Audio、Streamwork streaming 协议	遵循于这个策略的任何包限制为 8bit/s，将它们的 DSCP 值设为默认的 DSCP 值 超过这个值的任何数据包都会被丢掉
Default	未分类的	使用加权公平队列的队列方法和 WRED 的丢弃数据包的方法

六、实验的步骤

- 第 1 步** 在 ISP 和 Border 路由器上配置 ATM PVC。给 ISP 路由器的 ATM 0.1 接口分配 VPI: 3 和 VCI 130，给 Border 路由器的 ATM 0.1 接口分配 VPI: 3 VCI: 131。配置这些路由器使用 VBR-nrt，即 45Mbit/s 的持续信元速率和 50Mbit/s 的峰值信元速率。验证每一台路由器彼此可以 ping 通对方。

这一步是相当直接的并且只需要一些表项。在 ISP 和 Border 路由器上配置 ATM PVC，使用 VBR-nrt 的整形，并且验证这些路由器彼此可以连通对方。

```
ISP Router
interface ATM0.1 multipoint
 ip address 108.205.11.9 255.255.255.252
 pvc 3/130
  protocol ip 108.205.11.10 broadcast
  vbr-nrt 50000 45000
  encapsulation aal5snap
```

```
Border Router
interface ATM0.1 multipoint
 ip address 108.205.11.10 255.255.255.252
 pvc 3/131
  protocol ip 108.205.11.9 broadcast
  vbr-nrt 50000 45000
  encapsulation aal5snap
```

第 2 步 除了 ISP 路由器，将所有的其他路由器分配到 EIGRP AS 148。在它们的真正的网络边界上汇总这些路由；不要使用有类的汇总。在 Border 路由器上重分发指到 ISP 路由器的默认路由。在进行第 3 步之前，验证所有的路由器能够 ping 通所有其他的路由器。

这一步只需要一些表项才能正常地工作。首先，必须建立正确的网络语句，使得整个 108.205.0.0/16 的网络不会宣告到 Border 路由器所在的内部网络里。其次，必须关掉自动汇总来防止有类的汇总，最后，需要使用 **redistribute static** 命令将 Border 路由器上的默认路由进行重分发。

```
router eigrp 148
 redistribute static
 network 108.205.11.8 0.0.0.3
 network 192.168.3.0
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 108.205.11.9
```

第 3 步 从 Border 路由器上，配置网络地址翻译 (NAT)，使得所有的内部网络 10.1.1.0/24 和 192.168.0.0/16 能够连到超出 ISP 路由器的因特网上，而无需任何额外的路由。验证主机 PC 能够连接接口地址为 108.205.11.9/32 的 ISP 路由器。

只需要 3 个任务来配置第 3 步：建立一个访问控制列表来指定两个内部网络，建立一个 NAT 的语句来将访问控制列表中指定的地址 NAT 到 ATM 0.1 接口的 IP 地址，并且将 NAT 的配置绑定到 Border 路由器的 ATM0.1 和 Serial0 接口上，正如这里所示：

```
interface Serial0
 ip address 192.168.3.1 255.255.255.252
 ip nat inside
!
interface ATM0.1 multipoint
 ip address 108.205.11.10 255.255.255.252
 ip nat outside
 pvc 3/131
  protocol ip 108.205.11.9 broadcast
  vbr-nrt 50000 45000
  encapsulation aal5snap
!
ip nat inside source list 1 interface ATM0.1 overload
!
access-list 1 permit 192.168.0.0 0.0.255.255
access-list 1 permit 10.1.1.0 0.0.0.255
```

第4步 对 LAN-rtr 上发送方向的接口配置一个策略。这个策略应当匹配表 6-29 所示的可变量。

这一步需要多个表项来正常地工作。首先，必须对表中所定义的每一种流量类型定义一种分级映射，将每一种协议的类型分配到它所属的类中。其次，建立一个策略映射，它可以参考每一种类别的定义并且将这个类别应用到策略中去。接着，建立一个 class-default 的类别来匹配所有未定义的流量并应用默认的策略。这个策略用于绑定到接口 Serial0/1 上，使用 **outbound service-policy** 命令，就像这里所示，绑定到 LAN-rtr 路由器上。

```
class-map match-all No-Pri-Internet
  match protocol gnutella
  match protocol netshow
  match protocol napster
  match protocol nntp
  match protocol realaudio
  match protocol streamwork
class-map match-all Low-Pri-Internet
  match protocol ftp
  match protocol telnet
  match protocol secure-ftp
  match protocol secure-http
  match protocol secure-pop3
class-map match-all High-Pri-Internet
  match protocol http host "cisco.com"
class-map match-all Med-Pri-Internet
  match protocol http
  match protocol ssh
!
policy-map Internet-Policy
  class High-Pri-Internet
    bandwidth percent 15
    set ip dscp ef
  class Med-Pri-Internet
    bandwidth percent 55
    set ip dscp cs3
  class Low-Pri-Internet
    bandwidth percent 5
    random-detect
    set ip dscp cs1
  class No-Pri-Internet
    police cir 8000
    conform-action set-dscp-transmit default
    exceed-action drop
  class class-default
    fair-queue
    random-detect
!
interface Serial0/1
  ip address 192.168.3.2 255.255.255.252
  service-policy output Internet-Policy
  clockrate 1300000
```

第5步 在 Border 路由器的发送方向的 ATM 接口上启用 DSCP WRED。这是第4步所建立的策略的实施。

最后一步只需要一行配置，如这里所示。当你完成这部分的配置后，任何在 LAN-rtr 路由器上标记了 DSCP 值的流量一旦从 Border 路由器流出，将在发送方向的 ATM 接口上应用基于 DSCP 的 WRED 策略。记住，**random-detect** 命令只在物理接口上支持。

```
interface ATM0
  no ip address
  no atm ilmi-keepalive
  random-detect dscp-based
```

范例 6-44 显示了对于这个实验的完整的路由器配置。

范例 6-44 完整的路由器配置

```
hostname ISP
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
!
interface ATM0.1 multipoint
 ip address 108.205.11.9 255.255.255.252
 pvc 3/130
  protocol ip 108.205.11.10 broadcast
  vbr-nrt 50000 45000
  encapsulation aal5snap

hostname Border
!
ip cef
!
interface Serial0
 ip address 192.168.3.1 255.255.255.252
 ip nat inside
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 random-detect dscp-based
!
interface ATM0.1 multipoint
 ip address 108.205.11.10 255.255.255.252
 ip nat outside
 pvc 3/131
  protocol ip 108.205.11.9 broadcast
  vbr-nrt 50000 45000
  encapsulation aal5snap
!
router eigrp 148
 redistribute static
 network 108.205.11.8 0.0.0.3
 network 192.168.3.0
 no auto-summary
!
ip nat inside source list 1 interface ATM0.1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 108.205.11.9
!
access-list 1 permit 192.168.0.0 0.0.255.255
access-list 1 permit 10.1.1.0 0.0.0.255

hostname LAN-rtr
!
ip cef
!
class-map match-all No-Pri-Internet
 match protocol gnutella
 match protocol netshow
 match protocol napster
```

(待续)


```

match protocol nntp
match protocol realaudio
match protocol streamwork
class-map match-all Low-Pri-Internet
match protocol ftp
match protocol telnet
match protocol secure-ftp
match protocol secure-http
match protocol secure-pop3
class-map match-all High-Pri-Internet
match protocol http host "cisco.com"
class-map match-all Med-Pri-Internet
match protocol http
match protocol ssh
!
policy-map Internet-Policy
class High-Pri-Internet
bandwidth percent 15
set ip dscp ef
class Med-Pri-Internet
bandwidth percent 55
set ip dscp cs3
class Low-Pri-Internet
bandwidth percent 5
random-detect
set ip dscp cs1
class No-Pri-Internet
police cir 8000
conform-action set-dscp-transmit default
exceed-action drop
class class-default
fair-queue
random-detect
!
!
interface Ethernet0/0
ip address 10.1.1.1 255.255.255.0
!
interface Serial0/2
ip address 192.168.3.2 255.255.255.252
service-policy output Internet-Policy
clockrate 1300000
!
router eigrp 148
network 10.1.1.0 0.0.0.255
network 192.168.3.0 0.0.0.3
no auto-summary

```

6.12 进一步阅读资料

IP Quality of Service, by Srinivas Vegesna.

Cisco IOS 12.0 Quality of Service, by Cisco Systems.

Cisco Voice over Frame Relay, ATM, and IP, by Scott McQuerry, Kelly McGrew, and Stephen

Foy.

Integrating Voice and Data Networks, by Scott Keagy.

Deploying Cisco Voice over IP Solutions, by Phil Bailey.

RFC 1122, *Requirements for Internet Hosts—Communication Layers*, by Robert Braden.

RFC 1349, *Type of Service in the Internet Protocol Suite*, by Philip Almquist.

RFC 2205, *Resource ReSerVation Protocol (RSVP) —Version 1 Functional Specification*, by Bob Braden, Lixia Zhang, Steve Berson, Shai Herzog, and Sugih Jamin.

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, by Kathleen Nichols, Steven Blake, Fred Baker, and David L. Black.

RFC 2475, *An Architecture for Differentiated Services*, by Steven Blake, David L. Black, Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss.

RFC 2597, *Assured Forwarding PHB Group*, by Juha Heinanen, Fred Baker, Walter Weiss, and John Wroclawski.

RFC 2598, *An Expedited Forwarding PHB*, by Van Jacobson, Kathleen Nichols, and Kedarnath Poduri.

RFC 2697, *A Single Rate Three Color Marker*, by Juha Heinanen and Roch Guerin.

第五部分

BGP 理论和配置

第7章 BGP-4 的原理

第8章 BGP-4 配置介绍

第9章 高级BGP配置

第 7 章

BGP-4 的原理

边界网关协议第四版 (BGP-4) 是 BGP 目前的最新版本，它建立在对 BGP 第二版和第三版的扩展上，是现在对基于 IPv4 的因特网进行路由管理的主要路由协议。最初由 RFC1105/1163/1267 提出的 BGP 于 20 世纪 90 年代初替代了外部网关协议 (EGP) 成为主要的因特网路由协议。本章主要介绍了 BGP 协议，解释了相关的术语，并且涵盖了对 BGP 基本操作的描述，下一章将侧重于对 BGP 配置的介绍。

7.1 BGP 简介

BGP-4 (在本书的后续部分简称为 BGP) 是主要用来在自治系统之间路由 IPv4 流量的域间路由协议，自治系统是遵循相同的策略，在统一的管理控制下的一些路由域，如图 7-1 所示，AS1 和 AS2 是两个相连的自治域，每个里面各包含了一些遵循相同策略，被统一管理控制的路由器。

与 IP 地址类似，公共自治系统号码 (AS 号码) 必须是每个网络惟一的，一般由地区性因特网注册机构 (RIR) 分配，如美国的 American Registry for Internet Numbers (ARIN)。内部网关协议 (IGP) 通常用来管理路由域 (自治系统) 内部的路由，而包括 BGP 在内的外部网关协议 (EGP) 通常用来处理路由域 (自治系统) 之间的路由。

BGP 会话有两种类型：内部 BGP (I-BGP) 和外部 BGP (E-BGP)。内部 BGP 主要用于在自治系统内部路由流量，所有自治系统内部的流量必须遵循相同的路由策略，对外部 BGP 网络表现出本自治系统的一致性。外部 BGP 在自治系统的边界路由流量，每个自治系统维持自己的路由策略，同时利用边界路由器来增强路由策略的控制。每个参与公共因特网路由的自治系统都需要惟一的自治系统号码，自治系统号

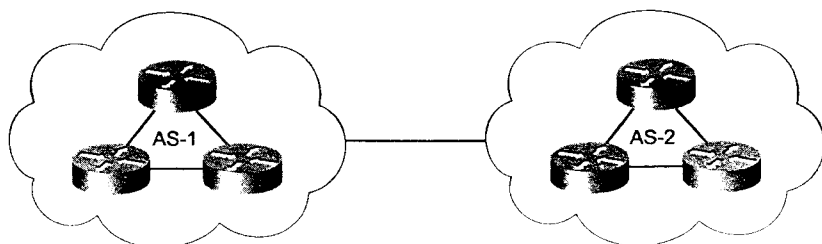


图 7-1 BGP 自治系统

码的范围是从 1~65535，其中 64512~65535 保留给私有自治系统使用。参与 BGP 会话的路由器称为 BGP “发言人 (Speaker)”，它们通过基于 TCP 端口 179 的可靠的 TCP 连接建立 BGP 端到端的会话，BGP 利用 TCP 协议来实现会话的建立、流量控制、重传和会话拆除。

注意：两个 BGP 对等体要想建立 BGP 会话，首先必须建立 TCP 会话。当进行 BGP 连接问题的故障排查时，很关键的一点是要验证每个 BGP 对等体是否能够通过 TCP 端口 179 访问另外一个对等体。

BGP 会话通过不同的消息类型来初始化、维持和结束，这些消息类型帮助 BGP 对等体通过不同的连接状态。当每个对等体都进入 Established 状态后，它们将相互交换路由更新。在初始的路由表被交换之后，BGP 路由更新只包含路由的变动（增加、修改和删除）。如果由于任何原因导致 BGP 对等体之间的 TCP 会话出错，BGP 的处理会立刻停止，所有通过 BGP 会话学到的路由信息也会从路由表中删除。

注意：BGP 消息类型将在本章“BGP 报文”一节中详细介绍。

当交换路由信息时，每个 BGP “发言人”可能会收到多个 BGP 路由，但是它只会使用和转发到每个目的网段的最佳路由。如果一个 BGP “发言人”无法使用 IP 主路由表的信息去验证某个路径的可达性，BGP 将不会使用这个路径。尽管如此，BGP 还是会将所有的路由信息（包括当前没有选为最佳的路径）存放在一个内部的 BGP 表中。

注意：BGP 的路由选择处理将在本章的“路由选择处理”一节中详细介绍。

与距离向量协议或者链路状态协议不同的是，BGP 是基于到达某个目的网段的 AS 路径来进行路由选择的。AS 路径是为了到达目的所需要经过的自治系统的列表。因为 BGP 是设计用来支持整个因特网路由表的，它不知道单独每一跳如何走，但是，BGP 包含了到达目的所需要经过的每一个 AS 号码。由于 BGP 记录了网络路径，而不是距离向量协议或链路状态协议的路由信息，所以 BGP 被称为路径向量协议。为了减少通告的网段数目和增加路由的可信度，一般网络管理人员都会在边界路由器上聚合或是汇总网段，这样可以维持 BGP 路由表的大小，减少发送给 BGP 邻居的网段数，同时也可以实现对网络策略的更好控制。

注意：在最新的思科 IOS 软件版本中，思科的 BGP 实现可以同时支持 IPv4 和 IPv6 的单播以及组播网络。本书只包含 BGP 的 IPv4 单播，本章中的“IP”是指“IPv4”。

路由策略是通过配置 BGP 属性来实现的，这些属性通常被指定给单独的网络路径，或是在 AS 边界路由器上指定给整个自治系统。BGP “发言人”通过路径属性来选择到达某个目的网段的最佳路径，对内部 BGP 和外部 BGP 来说它们的路径选择标准也不同。因为外部 BGP “发言人”必须选择从其他自治系统发出的路由，它们一般根据最短的自治系统路径和一些其他的 BGP 属性来选路。内部 BGP “发言人”只是在同一个自治系统内部接收和转发路由信息，所以这些路由的自治系统路径是空的，因此，它们就需要通过其他的 BGP 属性来选择最佳路径。为了防止路由循环，在同一个自治系统内部的所有内部 BGP “发言人”都不能接受在自治系统路径属性中包含自己的自治系统号码的路由。

注意：BGP 路径属性将会在本章的“BGP 路径属性”一节中进行详细介绍。

7.2 BGP 路由表

运行 BGP 协议的路由器为了不同的目的使用不同的路由表，主 IP 路由表包含了通过 RIP 或是开放最短路径优先 (OSPF) 等内部网关协议获得的路由、静态路由和直连网段。除此之外还有 3 个概念性的 BGP 表，通常被称为路由信息表 (RIB)，它们一般只包含和 BGP 相关的路由信息。BGP 表被用来保存 BGP 路径的信息，这些信息包括到每一个目的网段的最佳路径（用于本地路由），发送给其他 BGP 对等体的信息以及从其他 BGP 对等体收到的信息。当 BGP 选定了到某个网段的最佳路径后，该路径会被加入到主 IP 路由表中。

BGP 使用两个不同的路由表来分别存放接收到的和发出的网段通告：Adj-RIB-In 和 Adj-RIB-Out。这些路由表存放从其他 BGP “发言人”那里接收到的信息和发送给其他 BGP 对等体的信息。每个 BGP “发言人”为每个 BGP 对等体邻居维护一个“Adj-RIB-In”和“Adj-RIB-Out”。“Adj-RIB-In”表存放从其他 BGP 对等体那里收到的尚未处理的 BGP 信息，BGP 路由处理进程根据这些表中存放的 BGP 属性决定到某个网段的最佳路径。BGP 路由状态机（本地的 BGP 路由处理进程）处理表中的信息并发送到本地的 BGP 表中。“Adj-RIB-Out”表中的信息被发送给其他的 BGP 对等体。

当本地的 BGP 决策进程选好了到每个目的网段的最佳路径后，相关的信息被存放到本地的被称为 Loc-RIB 的 BGP 表中。Loc-RIB 存放着符合本机配置的 BGP 策略的路由信息，BGP “发言人”通过本地的 BGP 配置或是通过 BGP 会话从其他 BGP “发言人”学到这些信息。和其他两个 BGP 路由表不同的是，每台路由器（对 IPv4 BGP 路由而言）只有一个 Loc-RIB，在 Loc-RIB 中的每一条路径都附有以下路由信息：用来到达目的网段的下一跳 IP 地址，目的网段的度量，本地优先属性，权重属性，用来到达目的网段的 AS 路径，该路径是从内部或外部 BGP 进程还是从未知的起源学到。如果本地路由器能够验证下一跳是可以通过本地路由表中的某条内部网关协议、静态路由或是直连网段到达的，BGP 进程将选择路由并放入主 IP 路由表中。图 7-2 举例说明了 BGP 路由表在两个 BGP 对等体 (Apples 和 Oranges) 之间的 BGP 路由交换过程中是如何被使用的。

注意：术语“RIB”表示路由信息数据库，也代表路由表。

第 1 步 BGP “发言人” Apples 和 Oranges 建立 BGP 对等会话。

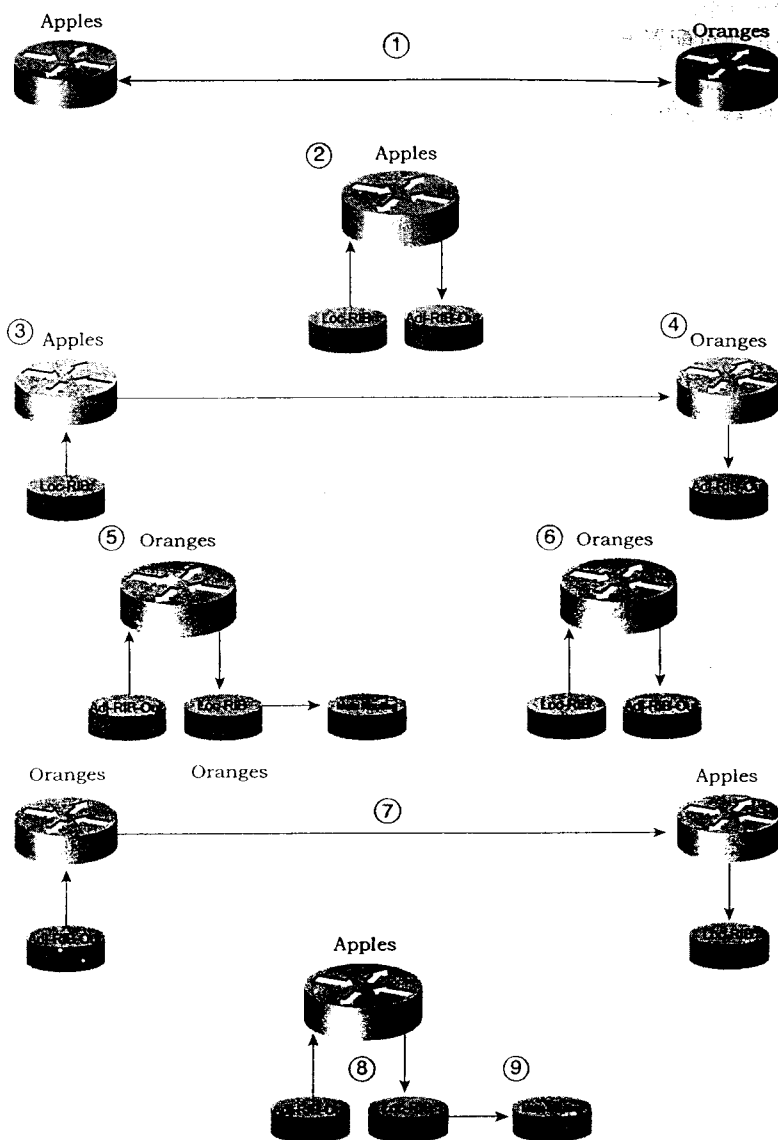


图 7-2 BGP 路由表

- 第 2 步** 路由器 Apples 从它的本地 BGP 表 (Loc-RIB) 中取出路由，根据针对 Oranges 路由器的 BGP 输出路由策略进行处理，将符合策略的路由放入 BGP 输出表 (Adj-RIB-Out) 中。
- 第 3 步** 路由器 Apples 将存放在 Apples/Oranges 对等体会话的 Adj-RIB-Out 表中的路由发送给 Oranges 路由器，这些 BGP 路由不仅符合本地的 BGP 路由策略，同时也遵循针对 Oranges 路由器的策略。
- 第 4 步** Oranges 路由器将从 Apples 路由器收到的路由信息存放在 Apples/Oranges 对等会话的 Adj-RIB-In 表中，以供 BGP 决策进程进行处理。
- 第 5 步** Oranges 路由器逐条地处理存放在 Adj-RIB-In 表中的每一条新路由，把和 Apples

对等体的 BGP 引入策略相匹配的每一个目的网段的最佳路径存储在 Loc-RIB 表中。尽管对每一个 BGP 会话都会有一对 Adj-RIB-In 和 Adj-RIB-Out，但是每台路由器都只有一个主 BGP Loc-RIB 表。当 Oranges 路由器检索本地的主 IP 路由表验证下一跳路径是可达的，如果本地的主路由表中没有通过内部网关协议学到到达同一目的网段的具有更低管理距离的路由，该路由就会被放入主路由表中参加将来的路由决策。

第 6 步 Oranges 路由器从它的本地 BGP 路由表(Loc-RIB)中取出路由，根据针对 Apples 路由器的输出策略处理这些路由，然后这些将要被通告的路由放入 BGP 的输出表 Adj-RIB-Out 中。

第 7 步 Oranges 路由器将它的 BGP 输出表 (Adj-RIB-Out) 中符合针对 Apples 路由器的输出策略的路由发送给 Apples 路由器，Apples 路由器将收到的路由放入 BGP 进入表 (Adj-RIB-In) 中。

第 8 步 Apples 路由器根据针对 Oranges 路由器的引入策略处理在它的 Adj-RIB-In 表中的路由信息，并且将到达每个目的网段的最佳路径（符合针对 Oranges 路由器的入站策略）放入本地的 BGP 表 (Loc-RIB) 中。

第 9 步 Apples 路由器验证在 Loc-RIB 表中到每个目的网段的下一跳是否可达，如果是可达的并且本地的主 IP 路由表中没有到达同一目的网段的更低管理距离的另外一个路由，Apples 路由器会把这些最佳路由放入它的主 IP 路由表中。

当路由器结束了更新处理，这些被处理的路由中只有路由的增加、变化和删除会被发送。只要 BGP 对等体之间的 TCP 会话保持在连接状态，对等的路由器就只会发送路由的变化。如果 TCP 会话有中断，那么所有通过该会话学到的路由都将被删除，当会话重新恢复时，整个路由交换过程会再次发生。

除非被显式地配置，BGP “发言人”不会主动通告任何网段，所以在通告任意网段之前，该网段需要显式地配置为 BGP 网段。BGP 网段可以通过以下这些方式配置：通过 **network** 命令；作为聚合网段的一部分；通过路由重分发；或是通过配置 BGP 的条件通告来触发。BGP 网段的配置会生成应用于每个 BGP 对等体的 BGP 输出策略。当生成 BGP 输出策略时，可以对配置的每一个 BGP 网段指定 BGP 属性，这些 BGP 属性可以使得路由更被优选或是更不被选中，从而影响其他路由器对某个特定路由的处理。

在 BGP “发言人”将一个网段的路由放入主 IP 路由表之前，它必须知道如何到达该网段的下一跳，这种路由的可达性是通过在主 IP 路由表中查找关于下一跳的路由来验证的。一般的内部网关协议，比如 EIGRP 和 OSPF，如果通过一个有效的邻居学到了相关的路由就会认为这个路由是有效的，BGP 与这些内部网关协议不一样，BGP 如果无法确认学到的某个路由的可达性就不会安装该路由到路由表中。只有当下一跳的路由在主 IP 路由表中找到以后，这条 BGP 路由才有可能被放入主 IP 路由表中。如果下一跳的路由在主 IP 路由表中无法找到，那么这条 BGP 路由还是会被放入 BGP 输入表 adj-RIB-In 中，而且通过 **show ip bgp** 命令可能还可以看到这条路由，但是这条路由不会被放入主 IP 路由表中。如果一条已经被放入主 IP 路由表中的 BGP 路由变为不可达（下一跳的路由在主 IP 路由表中被删除），那么这条 BGP 路由也会从主 IP 路由表中删除。如果还有到达同一网段的

其他的可达 BGP 路由存在，新的路由会被加入到主路由表中，当原来的路由重新可达后，

如果原来的路由是到达目的网段的最佳路径，那么原来的路由会替代主路由表中到达目的网段的其他路由。

7.3 邻居关系

两个 BGP“发言人”为了交换路由信息就必须建立对等体的关系，这种关系我们称为“邻居关系”。每个 BGP“发言人”都必须和任意一个需要交换路由信息的路由器建立对等关系，BGP 的邻居关系分为两种：内部和外部。本节将介绍 BGP 对等体之间如何建立 TCP 会话以及内部和外部 BGP 对等体是怎样建立邻居关系的。

内部和外部 BGP

如前面所描述，有两种类型的 BGP 会话：外部 BGP 会话用于自治系统（AS）之间的互连，内部 BGP 会话用在同一自治系统的 BGP“发言人”之间。外部和内部 BGP“发言人”都依赖内部网关协议（IGP）来维护路由表以及传送 BGP 路径信息。

一、外部 BGP 的运行

外部 BGP 被用来在属于不同的自治系统（AS）的路由器之间交换路由信息，每个自治系统都有自己的路由策略并且被不同的部门或是组织的人员独立地管理。由于 E-BGP 对等体分别属于不同的网络，每个 E-BGP 对等体必须配置相关的策略来控制内部路由向外部网络的广播，过滤掉不应该对外通告的内部网段，在需要的时候聚合路由和提供稳定的会话。除非特别指定，E-BGP 对等体路由器必须互相直连。图 7-3 显示了 E-BGP 是如何在 AS1、AS2 和 AS3 之间建立外部 BGP 会话的。必须注意的是只有自治系统边界路由器参加 E-BGP，E-BGP 对等体在自治系统的边界互相直连。

当 BGP 配置好后，每个对等体将协商建立 BGP 会话并且交换路由。当你把一个 BGP 路由器连向服务提供商时，很大的可能是你会使用串行、ATM 或是帧中继（Frame Relay）的连接，从你的因特网边界路由器直接连到你的服务提供商针对客户的边界路由器。在大多数的情况下，这些连接不会通过其他的非 BGP 路由器。

注意：出于对当前网络中永远存在的安全威胁考虑，E-BGP 连接可能需要在离开网络的时候通过防火墙或是其他的网络安全设备。为了规避 E-BGP 对等体必须直连的规则，可以使用 `ebgp-multihop` 命令指定 BGP 会话通过超过一跳的连接来建立，我们会在第 8 章中详细描述 `ebgp-multihop` 命令的使用。

当设计 BGP 网络的时候，应该使用稳定可靠的接口以避免路由的衰减。当某个接口的状态持续发生从接通到中断的变化时对等的 BGP 路由器就会衰减这条路由，临时挂起从振荡的路由器得到的路由通告一段时间直到重新稳定。很多服务提供商给客户id提供它们的路由衰减策略和违背策略的处罚措施。当为一个多归路的路由器配置 E-BGP 连接时，实践经验告诉我们z将环回（loopback）接口设为 BGP 的路由 ID，这样就会减少网络不稳定带来的影响，避免路由的衰减。

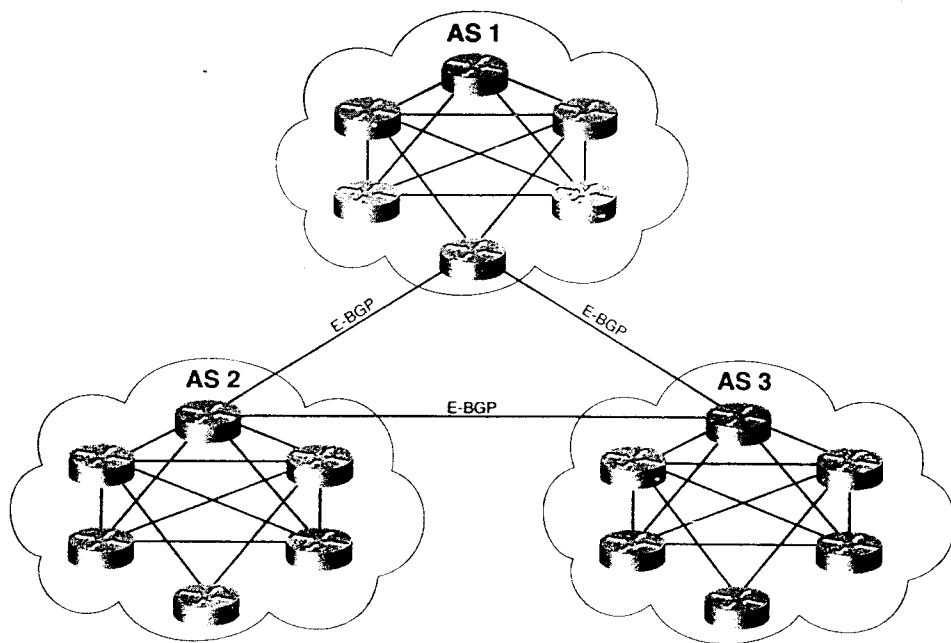


图 7-3 E-BGP 逻辑图

注意: 大多数的服务提供商要求客户的非多归路的路由器必须使用直连接口的 IP 地址来建立 BGP 会话。为了规避这个问题, 可以将路由器的 BGP ID 配置为环回接口的 IP 地址, 同时将 BGP 更新的源地址配置为直连接口的 IP 地址。关于多归路的配置会在第 8 章中详细描述。

注意: 通常来说使用环回接口作为 BGP 的路由 ID 是个好主意, 但是当路由器上同时运行 OSPF 和 BGP 的时候, 需要更仔细地规划 BGP 和 OSPF 的路由 ID。RFC 1745 中描述到“当路由器运行时 BGP / IDRP 的识别符必须要和 OSPF 的路由 ID 始终保持一致”。如果 OSPF 的路由 ID 和 BGP 的路由 ID 不一致, BGP 将无法与 OSPF 同步, 导致 BGP 不会向任何对等体通告任何未同步的路由。

在大多数的企业环境中, I-BGP 通常用来连接两个或是更多的企业边界路由器, 将网络多归路到两个以上的服务提供商。然而, 有些大型企业可能在核心路由器之间使用 I-BGP, 在不同核心站点的核心路由器之间使用 E-BGP 来提供路由策略。在很多的企业网络上, E-BGP 会话比 I-BGP 连接更为广泛地使用, 这是因为 E-BGP 会话被用来连接本地的自治系统到使用 I-BGP 的服务提供商。有很多方法可以连接专有网络到公共因特网, 最常用的方法是使用静态路由来为任何未知网段提供默认路由。当使用这种配置时, 服务提供商提供自己内部网络的 BGP 路由并且通告提供给客户的网段地址, 在这种情况下, 客户的网络不需要使用 BGP。如图 7-4 所示, 因特网路由器提供了通过服务提供商网络访问因特网的惟一路由。客户的网络运行自己的内部网关协议来路由本地网络不同楼层之间的内部流量, 因特网路由器提供了通过服务提供商的网络访问因特网的默认路由。

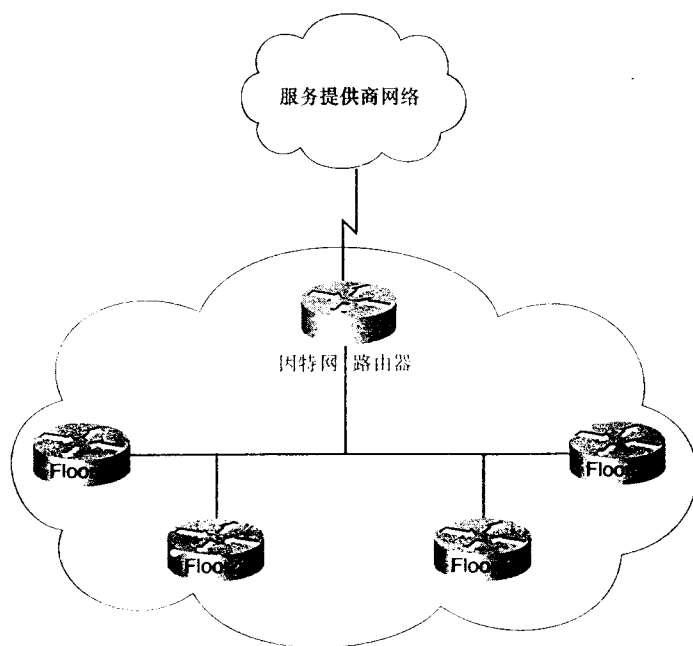


图 7-4 单归路网络

如果你的网络有通过公共地址注册机构（例如可以注册公有的地址和自治域号码的 ARIN）得到的公有 IP 地址，你就必须拥有你自己的惟一 BGP 自治域号码来向因特网通告你的公有网段。

注意：可以访问 ARIN 的站点 www.arin.net 获取关于美国的因特网注册机构的更多信息，访问 RIPE NCC（Réseaux IP Européens Network Coordination Centre）的站点 www.ripe.net 获取关于欧洲地址注册的更多信息，访问 APNIC（Asia Pacific Network Information Centre）的站点 www.apnic.net 获取关于亚太地区地址注册的信息。以上每个站点都包含了关于因特网地址分配和指定的信息以及相关的政策，同时还有统计信息。

在你被分配了公有 IP 地址范围并且注册了自治系统号码后，你就需要安排根据服务提供商的策略通告这些信息给你的上一级服务提供商。有很多种方法来连接和通告网段给上一级服务提供商，常用的方法有两种：单归路连接，一般不需要单独的自治系统号码和 RIR 分配的公有 IP 地址；多归路连接到一个以上的服务提供商，需要单独的自治系统号码和公有 IP 地址。图 7-5 显示了一个园区网是如何通过 BGP 多归路连接到两个不同的服务提供商的。在这个范例中，Notebook.com 连接到服务提供商 1（AS 890）和服务提供商 2（AS 123），Notebook.com 使用 AS 567 通告自己的网段。其中的因特网连接的冗余性通过将不同的服务提供商连接到同一台路由器上来实现，在某些情况下你的投资预算只允许你购买一台路由器的时候可以考虑这种解决方案，但是，你应该意识到只有一台路由器会造成单点故障。

在，下一个范例中，如图 7-6 所示，Quicky Web Title Registration 使用因特网路由器 1 和 2 向它的上一级服务提供商通告自己 NW、SW、NE 和 SE 地区的网段，Quicky 的网络使用自治系统号码 456 来通告自己的公共地址，服务提供商使用自治系统号码 876 和自治系统

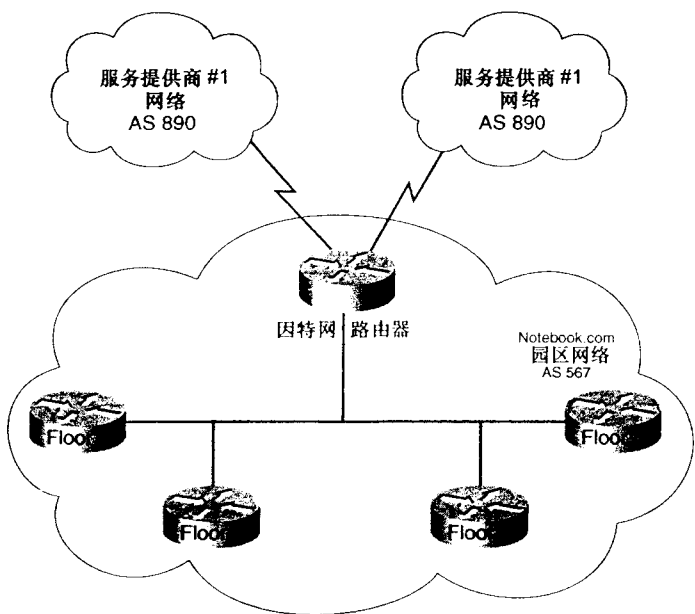


图 7-5 单归路园区到多个服务提供商

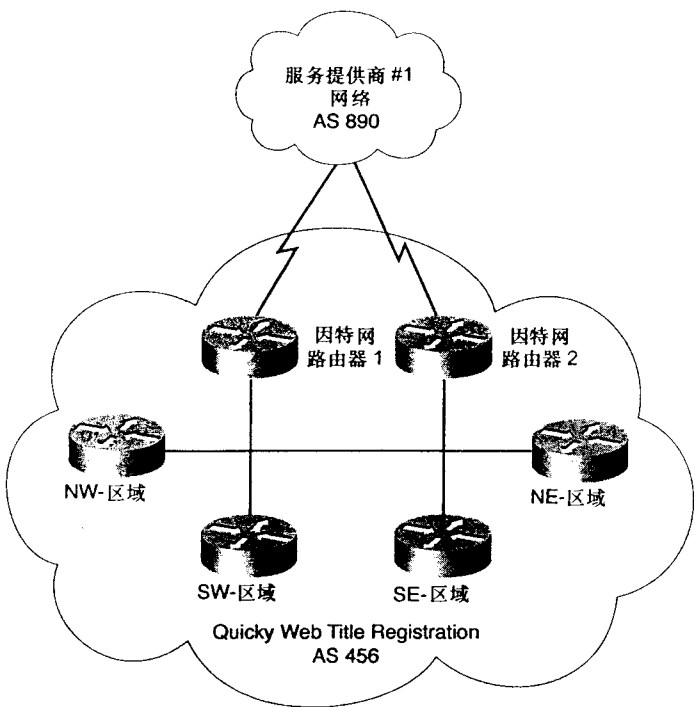


图 7-6 多归路园区到单个服务提供商

456 连接。在这个范例中，因特网连接的冗余性是通过将两台因特网边界路由器连接到一个服务提供商来实现的，这样就需要两台路由器、两个广域网接口和两条电路来保证硬件的冗

余性，尽管如此，由于只有一个服务提供商，还是有单点故障导致失败的可能，比如说，当服务提供商的网络出了故障，Quikcy 就将无法接入因特网。

在图 7-7 中，ServiceBank 公司使用自治系统 345 与服务提供商 1（自治系统 923）以及服务提供商 2（自治系统 159）连接。在这个范例中，ServiceBank 公司使用了两台路由器，每个连接到不同的服务提供商，也就是说多归路到多个服务提供商。在这种情况下，I-BGP 被用来在两个 E-BGP 因特网路由器之间交换路由信息。本范例使用了两台路由器、两条电路以及两个服务提供商、这种配置消除了所有的单点故障可能导致的失败，如果 ServiceBank 公司的任意一台路由器、电路或是服务提供商出了问题，ServiceBank 公司仍然可以连接到因特网，继续收发流量。

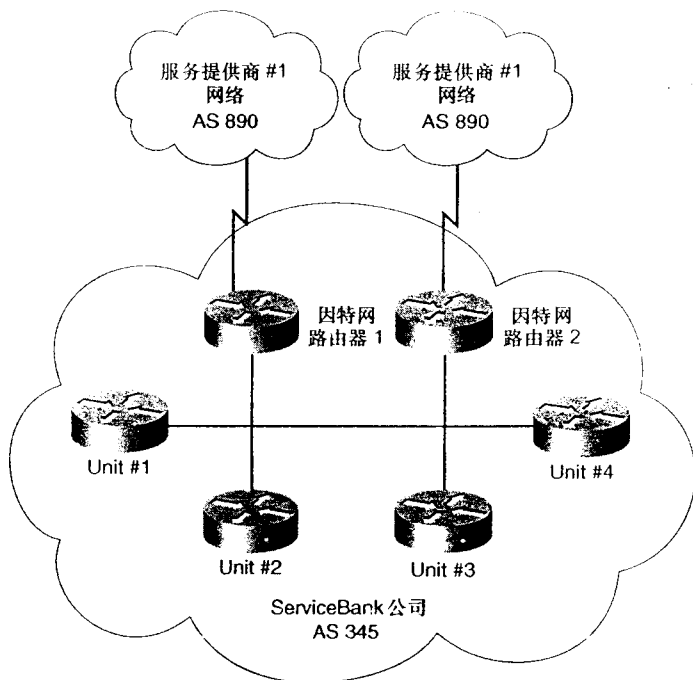


图 7-7 多归路园区到多个服务提供商

图 7-8 描述了 Mighty 软件公司是如何使用一个 BGP 自治系统 5655 将它的欧洲和美洲的网络连接到因特网的。在这个范例中，Mighty 软件公司在欧洲的路由器使用自治系统号码 5655 和它的服务提供商的自治系统 888 建立 E-BGP 的连接，这台路由器同时也通过串行 E1 广域网连接部分网状地接入 Paris、Vienna、London 和 Rome 的路由器。Paris、Vienna、London 和 Rome 的路由器也通过 E1 电路连接，同时使用内部网关协议来路由由内部网络。这台欧洲路由器负责与服务提供商处理和欧洲的网络流量相关的所有 BGP 路由，欧洲其他地区的路由器都通过这台路由器访问因特网。同样，在美洲的网络上，美国的路由器通过和服务提供商的 E-BGP 连接来处理所有的因特网流量，所有的美国路由器建立部分网状的拓扑来路由流量到因特网，到欧洲的网络或是互相访问。在这个范例中，几乎没有什么影响到因特网连接性的单点故障，惟一的失败可能在于单一的因特网服务提供商。

在图 7-9 中，Supernet 有两个部门，每个都有自己的自治系统号码，每个自治系统都是

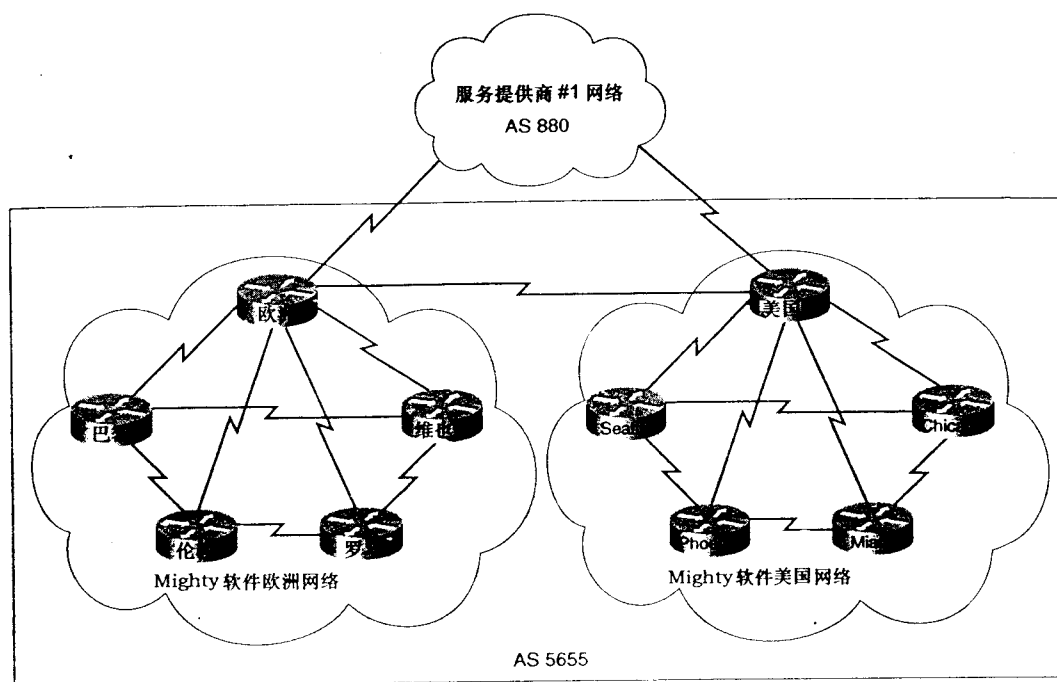


图 7-8 国际性的多归路到单个服务提供商

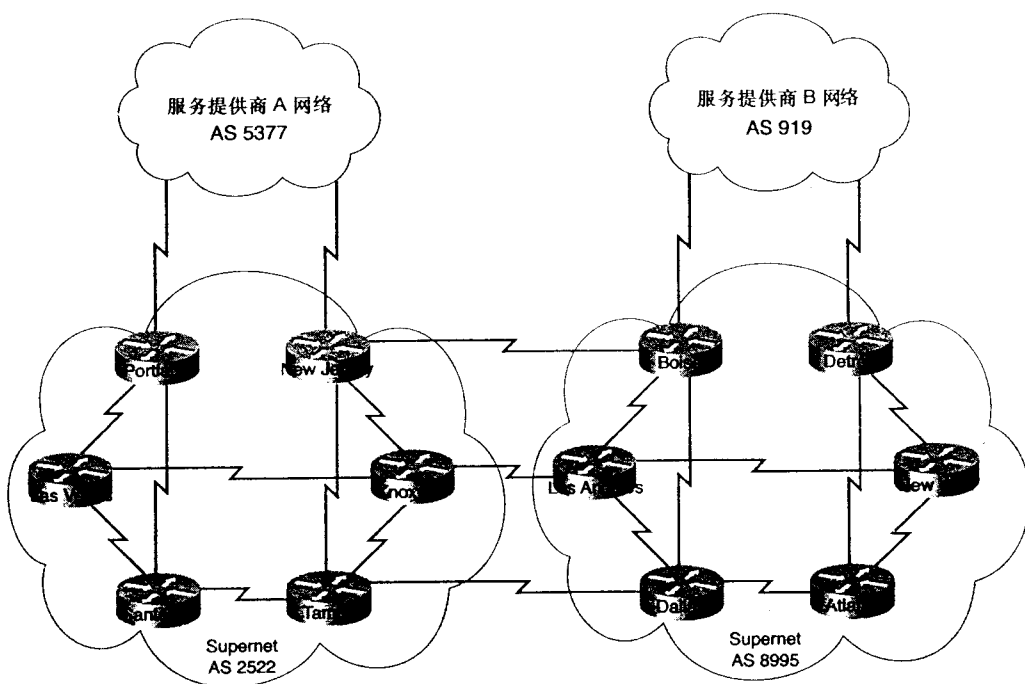


图 7-9 国内的多归路到多个服务提供商

和 New Jersey 与服务提供商 A（自治系统 5377）建立 E-BGP 连接，自治系统 8995 使用路由器 Boise 和 Detroit 与服务提供商 B（自治系统 919）建立 E-BGP 连接。每个网络都是部分网状的，也都使用内部网关协议来进行内部路由，路由器 New Jersey 和 Boise 同时还被用来在内部的两个自治系统之间建立 E-BGP 连接。为了互相通信的需要，I-BGP 连接也分别在路由器 Portland 和 New Jersey 之间和路由器 Boise 和 Detroit 之间建立。本范例是目前所介绍的最冗余的配置，多个站点有多个连接到多个服务提供商，减少了失败点的个数，当有足够的资源的时候，最好尽可能地建立最冗余的体系结构，减少潜在的失败点。

当边界路由器和上游的服务提供商建立了 E-BGP 的对等体关系后，这个边界路由器必须同时运行内部 BGP 进程和本自治系统内部的其他 BGP“发言人”之间通信，随后会讨论 I-BGP 的操作和 I-BGP 的规则。

二、I-BGP 的操作

I-BGP 被用在同一自治系统内部的 BGP 对等体之间。和 E-BGP 一样，每个 I-BGP“发言人”都必须被配置以便和相邻的 BGP 路由器建立对等的关系。BGP 不支持邻居关系的自动识别，为了给通过 I-BGP 连接的路由器提供对网络的一致理解，这些路由器必须配置为全网状连接的体系结构，如图 7-10 所示。每个有 I-BGP 对等关系的路由器都必须通过本地的 BGP 配置和其他所有的 I-BGP 对等体建立连接。每个 I-BGP 也必须在不同的 BGP 状态之间转换，给每一个相邻的对等体发送同样的 BGP 消息和建立 BGP 连接以交换路由信息。

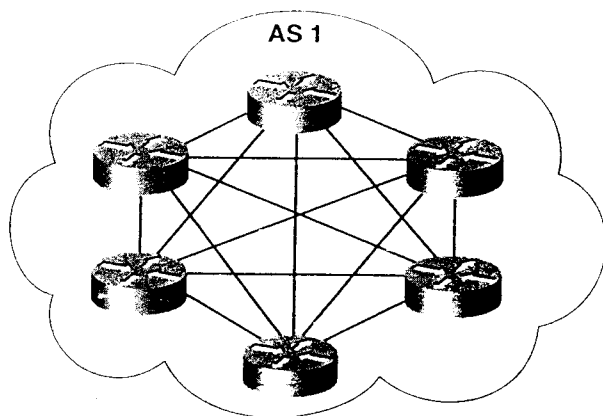


图 7-10 I-BGP 逻辑图

与 E-BGP 对等路由器不同的是，I-BGP 路由器不需要直接相连。如图 7-11 所示，在这个范例中，自治系统 4589 包含 5 个 I-BGP 对等路由器：Las Vegas、Cleveland、Omaha、D.C. 和 Tulsa。

尽管没有直接相连，每个 I-BGP“发言人”路由器都和自治系统 4589 中其他的路由器建立 I-BGP 对等连接。E-BGP“发言人”路由器——Cleveland 和 Vancouver 以及 Tulsa 和 Juarez 都通过直接的串行连接建立 E-BGP 的会话。需要注意的是，其他 I-BGP“发言人”路由器没有和它们自治系统外面的 E-BGP 路由器建立对等体关系，这是因为无论是 I-BGP 还是 E-BGP，每个 BGP 会话都必须显式地配置在每个对等路由器上。表 7-1 列出了 BGP 连接类型和 BGP 对等邻居。

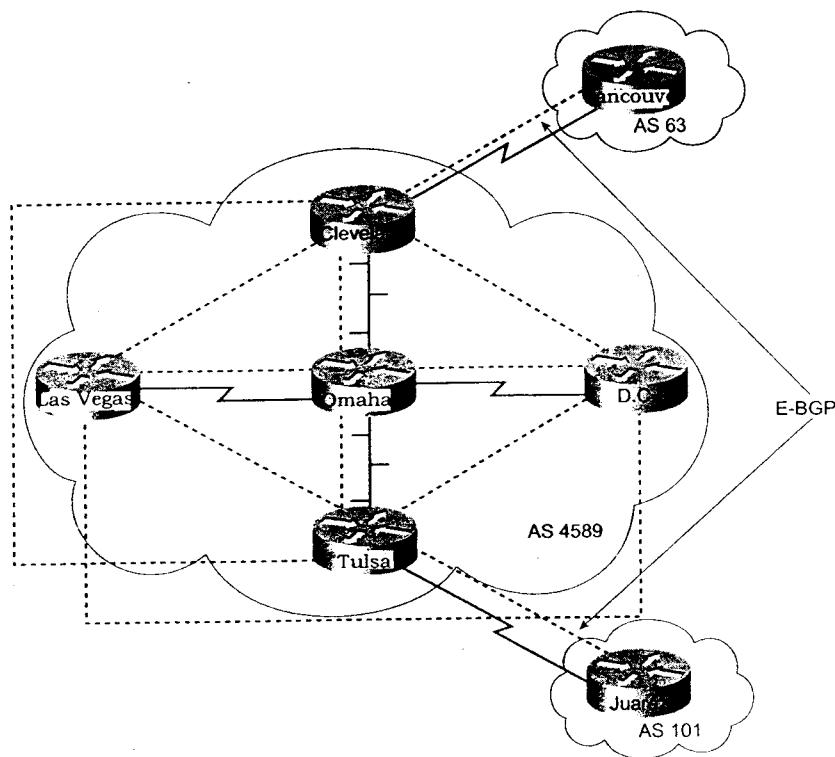


图 7-11 I-BGP 全网状连接及 E-BGP

表 7-1

BGP 对等连接

BGP 路由器	I-BGP 对等体	E-BGP 对等体	BGP 路由器	I-BGP 对等体	E-BGP 对等体
Las Vegas	Cleveland Omaha Tulsa D.C.	None	Omaha	Cleveland Las Vegas D.C. Tulsa	None
Cleveland	Las Vegas Omaha D.C. Tulsa	Vancouver	Tulsa	Las Vegas Omaha D.C. Cleveland	Juarez
D.C.	Cleveland Omaha Tulsa Las Vegas	None			

在本章的前面提到过，BGP 是个路径-向量路由协议，也就是说参与 BGP 路由进程的路由器根据 AS 路径来路由流量，而不是根据距离-向量算法的单个路由跳数或是链路状态协议中的开销值，BGP 为了建立无环路的路径，使用了一个称为“AS 路径”的属性，这个属性包括了到达目的地需要穿过的所有路径，每个 E-BGP “发言人”路由器在其学到的路由中将自己的自治系统号加入到“AS 路径”中，然后将这些信息转发给下游的 BGP 路由器，下游的 BGP 路由器利用这些信息来决定返回路径。I-BGP 邻居不会转发或是重新通告从它们自己的自治系统学到的路由（包含在 AS 路径中）给其他的 I-BGP 邻居，这样就可以防止在自治系统内部形成路由环路，当同一自治系统内部的两台路由器分别连到其他自治系统的两个 E-BGP 路由器时，它们不会将自己的自治系统号码加入“AS 路径”中发给内部的路由器。

注意：AS 路径属性将会在本章的“AS 路径属性”小节中详细描述。

如图 7-12 所示，路由器 A 通过 E-BGP 和路由器 C 相连，路由器 B 通过 E-BGP 和路由器 D 相连，路由器 A 和 B 同时还有 I-BGP 连接。当路由器 A 通过它的 E-BGP 会话从路由器 C 学到路由时，每条路由的 AS 路径属性中会包含自治系统 209。当路由器 A 将这些路由转发给路由器 B 的时候，它不会把自己的自治系统号码 400 放入 AS 路径中，这是因为路由器 A 和路由器 B 之间是 I-BGP 的对等体关系。但是当路由器 B 将路由转发给路由器 D 的时候，它会在 AS 路径中加入自己的自治系统号码 400，这是因为路由器 D 是个 E-BGP 对等体。因此，路由器 D 将会在通往路由器 C 的路由中看到含有 400 和 209 的 AS 路径，但是它不会知道在自治系统 400 中对同一路径其实有多条路由。

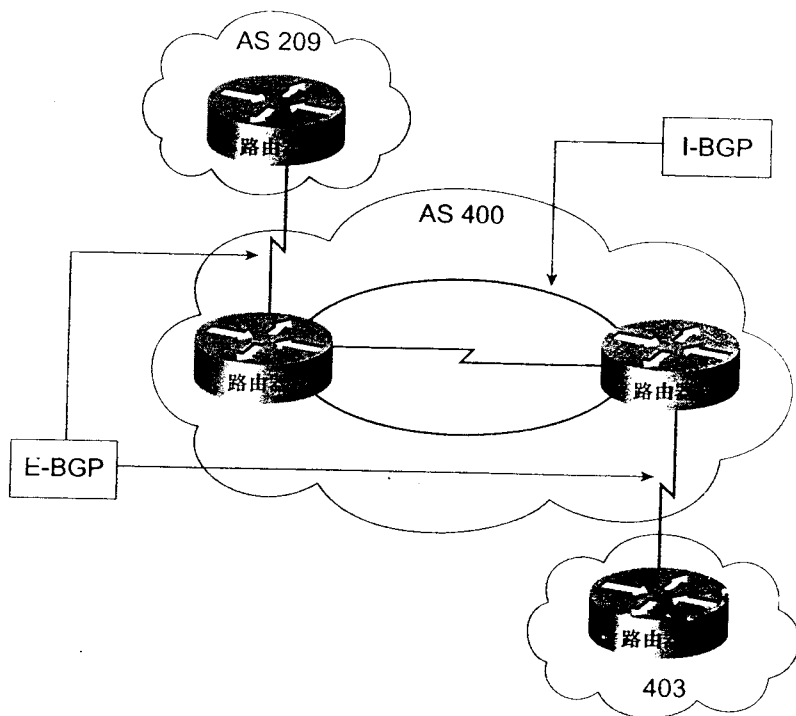


图 7-12 自治系统路径和 I-BGP

在前面这个范例中，当路由器 A 从路由器 C 处收到路由更新的时候，这些更新的 AS 路径中含有 209。当路由器 A 向路由器 B 转发路由器 C 通告给自己的网段时，因为路由器 A 和路由器 B 属于同一个自治系统，所以这些更新的 AS 路径属性还是只含有 209。但是当路由器 A 和 B 转发从路由器 C 学到的路由给路由器 D 的时候，会在 AS 路径属性中加上自己的自治系统号码 400，这样路由器 D 会看到从路由器 C 学到的路由的 AS 路径属性中同时包含 400 和 209。同样的道理，路由器 C 会看到从路由器 D 学到的路由的 AS 路径属性中同时包含 400 和 403。

如果改变一下拓扑结构，使路由器 D 同时和路由器 A 以及路由器 B 建立 E-BGP 会话，路由器 D 仍然有一条路径可以到达自治系统 209 的路由器 C，这样自治系统 400 中的路由器 A 和路由器 B 到达自治系统 209 的路由不会形成环路。图 7-13 描述了这些。路由器 C 通过路由器 A 可以到达自治系统 403 里的路由器 D，如果路由器 A 和路由器 D 之间的链路发生中断，

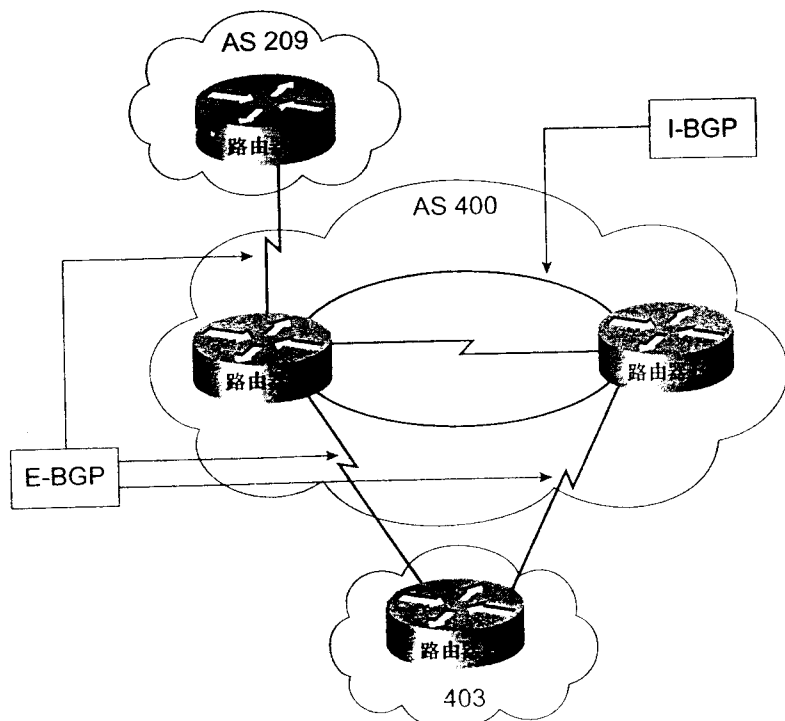


图 7-13 在自治系统 400 中加入一个新的 E-BGP 连接

路由器 A 和 D 还是可以通过路由器 B 相互访问。同样，如果路由器 D 和路由器 A 或是 B 之间的任何一个链路发生故障，路由器 D 还是可以访问自治系统 209 中的路由器 C。

I-BGP “发言人”向 E-BGP “发言人”发送的自治系统内部的 BGP 路由首先在自己的 IP 路由表中必须存在。在 IP 路由表中不存在或是没有同步的路由将不会通告给任何 BGP 对等体，这是因为 I-BGP 发言人需要验证在主 IP 路由表中不存在的路由的可达性。如果路由器在自己的主 IP 路由表中有通过内部网关协议、静态路由或是直连网络学到的完全一致的路由，这些路由将会被通告给其他的 BGP 对等体。这就是 BGP 的“同步规则”，BGP 表中的路由在向远程对等体通告之前必须和主路由表中的路由同步（这也就是说一个完全一致的、有效可达的匹配路由必须存在）。

注意：关于 BGP 一个需要被牢记的关键概念就是“同步”，本地 BGP 表（Loc-RIB）中的路径必须要 IGP 路由表中有效可达的路由同步，否则本地的 BGP 进程就不会将这些路径通告给远程的 BGP 邻居或是将这些 BGP 路由放入主 IP 路由表。换句话说，当启用“同步”功能后，通过 I-BGP 学到的路由是根据 IGP 协议学到的路由来验证有效性的。通常在同时运行 I-BGP 和 E-BGP 的路由器上是关闭同步特性的，如果没有关闭，并且 IGP 协议没有提供相应的路由信息，BGP “发言人”就不会使用或是传播这些它认为不可达的路由信息。关于 BGP 同步法则的使用将会在第 8 章中详细描述。

为了给上游的 BGP 对等体提供对自治系统一致的理解，默认情况下，自治系统边界路由器不会将从 I-BGP 会话学到的非同步的路由通告给 E-BGP 对等体，这是由同步法则决定的。

BGP 的同步法则允许 I-BGP 对等体给上游的对等体提供对自己网络的一致理解，如果 I-BGP 路由器的 BGP 和 IGP 路由表是同步的，这就表明所有的内部对等体都有相同的路由表，也就没有任何非同步的路由。只要 I-BGP 网络中所有的 BGP 路由器都是全网状连接的并且对网络有相同的理解，IGP 和 BGP 的同步特性就可以被关闭。以图 7-13 为例，只有关闭 BGP 同步，或者 IGP 路由协议使得 IGP 和 BGP 路由表同步，路由器 A 才会把从路由器 B 学到的路由通告给路由器 C 或是 D。同样，只有关闭 BGP 同步或者 IGP 和 BGP 路由是同步的，路由器 B 才会把从路由器 A 学到的路由通告给路由器 D。

到目前为止我们已经介绍了基本的 BGP 工作原理和术语，现在我们将介绍更深入的 BGP 的工作过程，下面的内容将会详细介绍以下主题：

- BGP 的消息；
- BGP 的状态机；
- BGP 属性；
- 路由反射器和联盟；
- BGP 决策过程。

7.4 BGP 报文

BGP 使用一系列的报文来与对等路由器初始化 BGP 会话，验证会话是活动的，发送路由更新，在错误发生时提醒对等路由器。每个报文都用来实现一个特定的功能，表 7-2 列出了所有 BGP 对等会话用到的报文的总结。

表 7-2

BGP 报文总结

报文号	报文类型	相关描述
1	OPEN 报文	用来建立 BGP 会话
2	UPDATE 报文	在已经建立的 BGP 会话上传送路由更新
3	通知报文	通知对端路由器错误的发生
4	保活报文	在 BGP 对等体之间互相发送来验证 BGP 会话是否处于连通状态
5	路由刷新报文	可选报文(在通告支持功能的过程中协商), 可用来动态请求对端 BGP 路由器发送 Adj-RIB-Out 表中的路由更新

注意：BGP-4 协议的工作原理最初在 RFC 1171 中定义，目前 IETF Inter-Domain Routing (IDR) 工作组正在起草更新版本，该版本应当于 2003 年底被接受。关于 IETF Inter-Domain Routing (IDR) 工作组的相关信息，请访问 <http://www.ietf.org/html.charters/idr-charter.html>。

7.4.1 OPEN 报文

为了建立一个 BGP 会话，每个 BGP 对等体必须给自己的每一个邻居发送一条 OPEN 报文。当 TCP 会话建立后，OPEN 报文将被发送，其中包括了关于本地 BGP “发言人”的相关信息。在一个 BGP 会话可以用来交换路由信息之前，OPEN 报文中的所有字段都需要被协商和接受。表 7-3 列出了构成 OPEN 报文的相关信息。

表 7-3 BGP OPEN 报文参数

报文参数	相关描述
版本 (version)	本地 BGP 路由器使用的 BGP 版本号 本地路由器的 BGP 版本号一般都是当前的版本，当对等路由器运行旧的 BGP 版本时也可以配置修改自己的版本号以保持兼容 如果 BGP 版本号不一致就无法建立 BGP 会话，每个对等路由器在建立 BGP 会话之前会相互协商兼容的 BGP 版本号
我的自治系统 (My AS)	本地 BGP 路由器使用的自治系统号码 如果“我的自治系统”和对端路由器配置的不相符将会导致 BGP 会话无法建立 “我的自治系统”的值也会决定 BGP 对等体之间是建立内部还是外部的 BGP 会话
保持计时器 (Hold Timer)	BGP 路由器从对端收到 UPDATE 或是保活报文之前期望等待的时间 在建立 BGP 会话之前 BGP 对等体路由器必须就保持时间协商并达成一致。在思科的路由器上 BGP 会话的默认保持时间是 180s，保持时间是可以配置的，取值范围是从 0~4 294 967 295，如果保持时间设为 0，将不会使用保活报文来验证 BGP 会话的有效性，如果保持时间没有被设为 0，那么保持时间必须被设为大于 3s 的值。在第 8 章会介绍通过命令 <code>default timers bgp</code> 来配置保持时间 特别需要强调的是，在建立 BGP 会话之前 BGP 对等体路由器必须就可接受的保持时间达成一致，除非对等体路由器同时修改保持时间，单台路由器不能单独修改已经被双方同意的保持时间
BGP 识别符 (BGP ID)	本地 BGP 路由器的识别符 BGP 识别符通常是路由器的本地识别符，和 OSPF 类似，是环回接口的最大 IP 地址，环回接口可以为路由器识别符提供最为可靠稳定的接口。在第 8 章会介绍通过命令 <code>bgp router-id</code> 来修改 BGP 识别符 BGP 识别符的值必须和本地及远端关于 BGP 对等关系的配置相一致，同时远端对等体必须从本地对等体可达，否则无法建立会话
可选参数 (optional)	包含了可选的 BGP 参数，例如 Marker 字段包含了用于认证的信息，如果没有配置认证，Marker 字段的值会全是 1 可选的 Capabilities 字段包含了 BGP 特性协商所需要的信息，BGP 对等体或者支持或者不支持这个字段，如果不支持，对端将忽略这个参数并重新协商会话

图 7-14 演示了路由器 A 和路由器 B 是如何使用 BGP 的 OPEN 报文来建立 BGP 会话的。在本例中，路由器 A 发送了一条 OPEN 报文给路由器 B，其中包含了自己的 BGP 版本号是 4，自己的自治系统号是 402，保持时间是 180s 以及 BGP 识别符是 204.168.75.1。路由器 B 返回自己的 OPEN 报文，包含了自己的 BGP 版本号是 4，自己的自治系统号是 917，保持时间是 180s 以及 BGP 识别符是 204.168.75.25。需要特别指出的是，在本例中每个 BGP 路由器分别位于不同的自治系统，这从它们的自治系统号码可以看出，因此它们之间将会建立 E-BGP 会话。

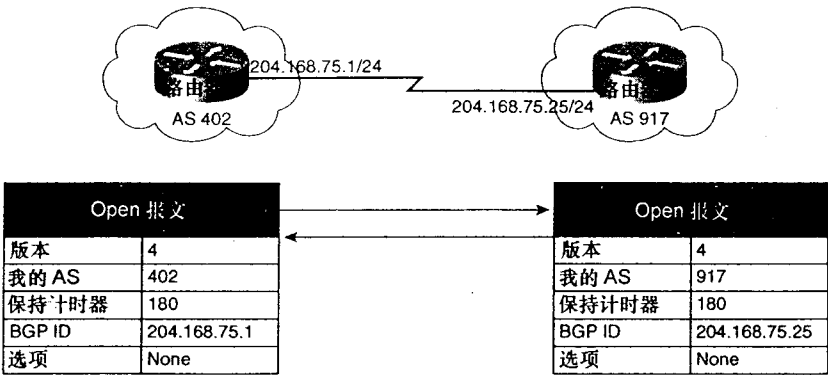


图 7-14 建立一个 BGP 会话

范例 7-1 给出了截获的一个包含 BGP OPEN 报文的数据包。BGP 使用的 IP 优先值

范例 7-1 BGP OPEN 报文

```

Frame Status Source Address Dest. Address Size Rel. Time Delta Time Abs. Time
Summary
8 [10.50.4.1] [10.50.4.2] 99 0:00:37.326 0.003.216 04/28/2002 03:14:50 PM
BGP: type = Open
DLC: -----
DLC Header -----
DLC:
DLC: Frame 8 arrived at 15:14:50.2341; frame size is 99 (0063 hex) bytes.
DLC: Destination = Station 000427228197
DLC: Source = Station 0004272281D8
DLC: Ethertype = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = C0
IP: 110. .... = internetwork control
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 85 bytes
IP: Identification = 2
IP: Flags = 0X
IP: .0.. .... = might fragment
IP: ..0. .... = last fragment IP: Fragment offset = 0 bytes
IP: Time to live = 1 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = 9C7B (correct)
IP: Source address = [10.50.4.1]
IP: Destination address = [10.50.4.2]
IP: No options
IP:
TCP: ----- TCP header -----
TCP:
TCP: Source port = 11002
TCP: Destination port = 179 (BGP)
TCP: Sequence number = 3817488861
TCP: Next expected Seq number= 3817488906
TCP: Acknowledgment number = 3816595146
TCP: Data offset = 20 bytes
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16384
TCP: Checksum = 97C3 (correct)
TCP: No TCP options
TCP: [45 Bytes of data]
TCP:
BGP: ----- BGP Message -----
BGP: BGP: 16 byte Marker (all 1's)
BGP: Length = 45
BGP: BGP type = 1 (Open)
BGP:

```

(待续)

```
BGP: Version = 4
BGP: AS number = 1
BGP: Hold Time = 180 Second(s)
BGP:
BGP Identifier = C0A80501, [192.168.5.1]
BGP:
BGP: Optional Parameters Length = 16
BGP: Unknown Option Data
BGP:
ADDR HEX                                ASCII 0000:
00 04 27 22 81 97 00 04 27 22 81 d8 08 00 45 c0 | ..'".....E.
0010: 00 55 00 02 00 00 01 06 9c 7b 0a 32 04 01 0a 32 | .U.....{.2...2
0020: 04 02 2a fa 00 b3 e3 8a 41 dd e3 7c 9e ca 50 18 | ..*....A..I..P.
0030: 40 00 97 c3 00 00 ff ff ff ff ff ff ff ff ff | @.....
0040: ff ff ff ff ff ff 00 2d 01 04 00 01 00 b4 c0 a8 | .....
0050: 05 01 10 02 06 01 04 00 01 00 01 02 02 80 00 02 | .....
0060: 02 02 00 | ...
```

是“网络控制”，如显示的“110000”，“网络控制”一般用来标记高优先级的路由流量，请参考第 5 章。注意这个报文的 TCP 会话使用了目的端口号 179，也就是 BGP 的目的端口号。这个 OPEN 报文（BGP 报文类型 1）的 BGP 报头包括了一个全为 1 的 Marker 字段，这表明没有使用 MD-5 认证，在 45 个字节的报头中，“版本”字段指明了发送主机使用的是 BGP 第四版，这台主机属于自治系统 1，保持时间是 180s，它的 BGP 识别符是 192.168.5.1。

BGP 能力通告

从 BGP 第四版开始，在初始 BGP 会话的过程中 BGP 对等体可以进行能力协商，这是通过 OPEN 报文中可选择的“能力”参数来实现的。RFC 2842 描述了 BGP 的能力协商，通过增加这个参数可以使得 BGP 的规范不需要进行协议的升级就支持新的特性。

BGP 对等体可以利用通告来交换各自对不同能力的支持，并且在会话中协商和尽可能地使用双方都同意的特性。如果有任何一方不支持某个可选参数，它可以发送带有错误信息“Unsupported Optional Parameter.”的通知报文，当收到通知报文后，原来的发送方会重新发送不带未支持参数的 OPEN 报文，直到双方都同意参数的设置为止。表 7-4 列出了 IANA 定义的 BGP 能力代码。

表 7-4

BGP 能力代码

能力代码	相关描述	能力代码	相关描述
0	保留	5~63	未分配
1	BGP 第四版的多协议扩展	64	平滑重启能力
2	BGP 第四版的路由刷新能力	65	支持 4 个八位组的自治系统号码的能力
3	合作路由过滤能力	66	支持动态能力
4	多路由能力	128~255	设备制造商定义的能力

7.4.2 UPDATE 报文

文包含了需要通告给对等路由器的每一条路由。在 BGP 路由中，网络前缀也被称为网络层可达性信息 (*Network Layer Reachability Information, NLRI*)。表 7-5 列出了 BGP UPDATE 报文包含的信息以及 BGP UPDATE 报文字段的描述。

表 7-5

BGP UPDATE 报文信息

报文参数	相关描述
不可用的路由长度	本字段包含了将要被从 BGP 路由表中撤销的路由总条数，如果值为 0 表示这个报文中没有路由被撤销
撤销路由	本字段包含了需要从 BGP 表中删除的前缀，相关信息以 [长度, 前缀] 的格式保存，每个要被删除的路由由同时通过这种格式发送给对端的邻居
路径属性总长度	本字段确定了路径属性的总长度 (以八位组表示)
路径属性	<p>BGP 路径属性 (属性类型代码) 是 BGP 决策进程使用的基本衡量手段，IANA 定义了 19 个 BGP 路径属性，最重要的 10 个如下：</p> <ol style="list-style-type: none">1. 起源 (ORIGIN)2. AS 路径 (AS_PATH)3. 下一跳 (NEXT_HOP)4. 多出口鉴别器 (MULTI-EXIT-DISC)5. 本地优先 (LOCAL-PREF)6. 原子聚合 (ATOMIC-AGGREGATE)7. 聚合者 (AGGREGATOR)8. 团体 (COMMUNITY)9. 起源者识别符 (ORIGINATOR_ID)10. 集群列表 (CLUSTER_LIST) <p>路径属性字段包含 3 个值：</p> <ul style="list-style-type: none">• 属性类型—包括两个子字段，一个是上面所列的属性类型代码，另一个是这些属性的标志位• 属性长度—定义了属性的长度• 属性值—包含了每个属性类型代码的值
属性类型 (是路径属性字段的一个子部分)	<p>属性类型字段包括两项：属性标志和属性类型代码，每个路径属性字段的属性类型代码部分都和属性类型目录相关，属性类型目录定义了属性是如何转发给其他 BGP 路由器的，以下为 4 种属性类型：</p> <ol style="list-style-type: none">1. 公认必遵2. 公认自决3. 可选过渡4. 可选非过渡 <p>下面很快就会介绍属性标志</p>
网络层可达性信息 (NLRI)	<p>NLRI 字段作为 UPDATE 报文的一部分包含了被通告为可达的路径 (网络层可达性信息)</p> <p>NLRI 字段包含了用 [长度, 前缀] 格式通告的每条路径的前缀，这些信息从本地路由器的 Adj-RIB-Out 数据库中取得并将被加入相邻路由器的 Adj-RIB-In 数据库中</p>

两个 BGP 对等体成功地建立 BGP 会话后，它们就可以通过 UPDATE 报文交换路由信息，UPDATE 报文中包含的信息有：将要被加入 BGP 表中的新路由、不再可达的路由 (将要被从 BGP 表中删除) 以及路由的路径属性。

如前面的表所示，字段“不可用的路由长度”包含了将要被从 BGP 表中删除的路由条数，字段“撤销路由”以 [长度, 前缀] 格式包含了被删除的实际路由，路径属性字段包含了 UPDATE 报文中路径的属性类型代码，属性标志字段指明了路由进程应该如何处理这些属性，最后，字段“网络层可达性信息 (NLRI)”包含了需要被通告的新增和修改的路由。

在 BGP 中，每个路由更新包含了报文中所有网络层可达性信息路径的属性，以下列出了 10 种在 IP 环境中处理 BGP 时最常用到的基本属性类型代码和属性值：

1. 起源 (ORIGIN) ——指明路由的起源：内部边界网关协议、外部边界网关协议或者不完整。

2. AS 路径 (AS_PATH) ——包含了路由经过的自治系统列表。

3. 下一跳 (NEXT_HOP) ——到达目的网段需要经过的下一跳。
4. 多出口鉴别器 (MULTI-EXIT-DISC) ——如有多个离开自治域的出口，就以多出口鉴别器作为标准来判定选用哪条路径。
5. 本地优先 (LOCAL-PREF) ——在自治系统内部表明某个路径比其他路径更优先。
6. 原子聚合 (ATOMIC-AGGREGATE) ——表明本地进程在到某个目的网段的路由中没有选取最精确的路径。
7. 聚合者 (AGGREGATOR) ——本属性用来指明会聚了路由的路由器的 IP 地址。
8. 团体 (COMMUNITY) ——指明了本地 BGP 团体值，默认的情况下所有支持团体属性的路由器都属于 “Internet” 团体。
9. 起源者识别符 (ORIGINATOR_ID) ——指明了路由反射集群中的一个路由反射器。
10. 集群列表 (CLUSTER_LIST) ——包含了一个被反射的路由通过的一个反射路径。

每个属性代码类型都伴随着一个属性标志，属性标志指明了对等路由器应该如何处理这些属性。表 7-6 列出了 4 种属性标志以及它们的关联标志，在本章的后面会详细介绍。

表 7-6 BGP 属性标志

属性标志	标志名称	描述
最高位	可选位	定义属性是公认 (0) 或是可选 (1)
第二位	过渡位	定义一个可选属性是非过渡 (0) 或是过渡的 (1)
第三位	部分位	定义一个可选过渡属性是完整 (0) 还是部分的 (1)
第四位	扩展长度位	定义一个属性的长度是单字节 (0) 还是双字节 (1)，本标志只有在属性长度大于 255 个八位组的时候才使用 (设置为 1)

范例 7-2 展示了对 UPDATE 报文的协议分析。注意，范例中的报文是 68 字节 BGP 类型 2 的 UPDATE 报文，其中字段 “Marker” 的值为全 “1”，表明没有使用认证。这个更新中不可用的路由长度为 0，表明不包含任何被撤销路由。报文中的第一个属性是公认过渡类型 1 的起源属性，值为 0-IGP，表明路由来自内部网关协议。下一个公认过渡属性是类型为 2 的 AS 路径，其中列出了路由经过的所有自治系统，它的路径类型段落的值为 2 (ASSEQUENCE)，表示更新报文中包含了有序的自治系统列表。路径长度部分的值为 1，表明路径中只包含一个自治系统，自治系统识别符表明数据包始发于自治系统 2。再下一个公认过渡属性是下一跳，它的值为 10.50.4.2。最后一个可选非过渡的类型为 4 的多出口鉴别器属性，主要用来判断当自治系统有多个出口时应该使用哪条路由，本例中的多出口鉴别器为 0。

范例 7-2 BGP UPDATE 报文

```
Frame Status Source Address Dest. Address Size Rel. Time Delta Time Abs. Time
Summary
13 [10.50.4.2] [10.50.4.1] 141 0:00:37.537 0.001.028 04/28/2002 03:14:50 PM
BGP: type = Update
DLC: ..... DLC Header .....
DLC:
DLC: Frame 13 arrived at 15:14:50.4449; frame size is 141 (008D hex) bytes.
DLC: Destination = Station 0004272281D8
DLC: Source = Station 000427228197
DLC: Ethertype = 0800 (IP)
```

(待续)


```

DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = C0
IP: 110. .... = internetwork control
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 127 bytes
IP: Identification = 3
IP: Flags = 0X
IP: .0.. .... = might fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 1 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = 9C50 (correct)
IP: Source address = [10.50.4.2]
IP: Destination address = [10.50.4.1]
IP: No options
IP:
TCP: ----- TCP header -----
TCP:
TCP: Source port = 179 (BGP)
TCP: Destination port = 11002
TCP: Sequence number = 3816595210
TCP: Next expected Seq number = 3816595297
TCP: Acknowledgment number = 3817488925
TCP: Data offset = 20 bytes
TCP: Flags = 18
TCP: ..0. .... = (No urgent pointer)
TCP: ...1 .... = Acknowledgment
TCP: .... 1... = Push
TCP: .... .0.. = (No reset)
TCP: .... ..0. = (No SYN)
TCP: .... ...0 = (No FIN)
TCP: Window = 16320
TCP: Checksum = 19F9 (correct)
TCP: No TCP options
TCP: [87 Bytes of data]
TCP:
BGP: ----- BGP Message -----
BGP:

BGP: 16 byte Marker (all 1's)
BGP: Length = 68
BGP:
BGP: type = 2 (Update)
BGP:
BGP: Unfeasible Routes Length = 0
BGP: No Withdrawn Routes in this Update
BGP: Path Attribute Length = 25 bytes
BGP: Attribute Flags = 4X
BGP: 0... .... = Well-known
BGP: .1... .... = Transitive
BGP: ..0. .... = Complete
BGP: ...0 .... = 1 byte Length
BGP: Attribute type code = 1 (Origin)
    
```

(待续)

```

BGP: Attribute Data Length = 1
BGP: Origin type = 0 (IGP)
BGP: Attribute Flags = 4X
BGP: 0... .. = Well-known
BGP: .1... .. = Transitive
BGP: ..0... .. = Complete
BGP: ...0... .. = 1 byte Length
BGP: Attribute type code = 2 (AS Path)
BGP: Attribute Data Length = 4
BGP: Path segment type = 2 (AS_SEQUENCE)
BGP: Path segment length = 1
BGP: AS Identifier = 2
BGP: Attribute Flags = 4X
BGP: 0... .. = Well-known
BGP: .1... .. = Transitive
BGP: ..0... .. = Complete
BGP: ...0... .. = 1 byte Length
BGP: Attribute type code = 3 (Next Hop)
BGP: Attribute Data Length = 4
BGP: Next Hop = [10.50.4.2]
BGP: Attribute Flags = 8X
BGP: 1... .. = Optional
BGP: ..0... .. = Non-transitive
BGP: ...0... .. = Complete
BGP: ...0... .. = 1 byte Length
BGP: Attribute type code = 4 (Multi Exit Disc)
BGP: Attribute Data Length = 4
BGP: Multi Exit Disc Attribute = 0
BGP:
BGP: Network Layer Reachability Information:
BGP: IP Prefix Length = 24 bits, IP subnet mask [255.255.255.0]
BGP: IP address [192.168.11.0]
BGP: IP Prefix Length = 24 bits, IP subnet mask [255.255.255.0]
BGP: IP address [192.168.12.0]
BGP: IP Prefix Length = 24 bits, IP subnet mask [255.255.255.0]
BGP: IP address [192.168.13.0]
BGP: IP Prefix Length = 24 bits, IP subnet mask [255.255.255.0]
BGP: IP address [192.168.14.0]
BGP: IP Prefix Length = 24 bits, IP subnet mask [255.255.255.0]
BGP: IP address [192.168.15.0]
BGP:
BGP: 16 byte Marker (all 1's)
BGP: Length = 19
BGP:
BGP type = 4 (KEEPALIVE)
BGP:
DLC: --- Frame too short
ADDR HEX
                                ASCII
0000: 00 04 27 22 81 d8 00 04 27 22 81 97 08 00 45 c0 | ...E.
0010: 00 7f 00 03 00 00 01 06 9c 50 0a 32 04 02 0a 32 | ...P.2...
0020: 04 01 00 b3 2a fa e3 7c 9f 0a e3 8a 42 1d 50 18 | ...*.J...B.P.
0030: 3f c0 19 f9 00 00 ff ff ff ff ff ff ff ff ff | ?..
0040: ff ff ff ff ff ff 00 44 02 00 00 00 19 40 01 01 | .....D....
0050: 00 40 02 04 02 01 00 02 40 03 04 0a 32 04 02 80 | .e.....e...2...
0060: 04 04 00 00 00 00 18 c0 a8 0b 18 c0 a8 0c 18 c0 | .....
0070: a8 0d 18 c0 a8 0e 18 c0 a8 0f ff ff ff ff ff ff | .....
0080: ff ff ff ff ff ff ff ff ff ff 00 13 04 | .....

```

报文中的下一个字段是网络层可达性信息 (NLRI)，包括了通告的新增和需要被修改的路由。本报文中包含了到 192.168.11.0/24、192.168.12.0/24、192.168.13.0/24、192.168.14.0/24

在图 7-15 所示的范例中，路由器 A 和 B 建立了 BGP 会话并且正在通过 UPDATE 报文交换路由信息。路由器 A 发送一个 UPDATE 报文要求删除两条路由：一个是 50.1.1.0/24，另一个是 50.2.2.0/24。这个更新报文中同时要求增加 4 条新的路由：51.3.3.0/24、51.4.4.0/24、51.5.5.0/24 和 60.1.1.0/24。这些路由通过 E-BGP 的方式学习到并且发送出去，但是起源设为 I-BGP 会话（通过类型 1 的 IGP 路径属性得出），AS 路径包括了自治系统 402、10 和 30，下一跳为 51.5.2.4。路由器 B 收到 UPDATE 报文后，从 Adj-RIB-In 表中删除到 50.1.1.0/24 和 50.2.2.0/24 的路由，同时增加到 51.3.3.0/24、51.4.4.0/24、51.5.5.0/24 和 60.1.1.0/24 的路由以供 BGP 决策进程处理。

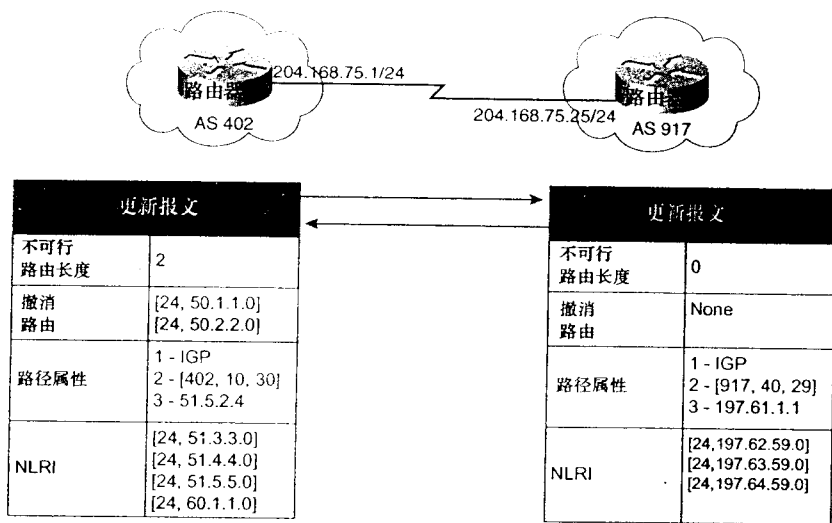


图 7-15 路由器交换 BGP 更新

路由器 B 从本地的 Adj-RIB-Out 表中取出路由，然后发送更新给路由器 A，其中包括新增到网段 197.62.59.0/24、197.63.59.0/24 和 197.64.59.0/24 的路由。这些新路由都来自 E-BGP 会话，但是起源于 I-BGP 会话，AS 路径包括了自治系统 917、40 和 29，下一跳为 197.61.1.1。路由器 A 接收了新路由后放入本地的 Adj-RIB-In 表中供 BGP 决策进程处理，然后将最佳路由加入到本地的 BGP 表 Loc-RIB 中。如果没有新的路由变化，路由器 A 和 B 就不会再发送新的 UPDATE 报文，它们只会来回发送保活报文通知对方 BGP 会话保持在激活状态。

7.5 通知（NOTIFICATION）报文

BGP 通知报文用于表明一个错误，这个错误可能会导致 BGP 会话中断。通知报文通常在会话刚刚中断时就立即产生。当 BGP 连接中断后，在两个 BGP 对等体之间的 TCP 会话将关闭，所有资源被释放。“路由撤销”报文会向其他 BGP 对等体发送，所有 BGP 路由会从路由表中删除。BGP 会话可能会出于多种原因被中断。表 7-7 描述了 6 个主要的错误通知报文。

表 7-7

BGP 通知报文

报文号	报文类型	描述
1	报文报头错误	表明处理 BGP 报文的报头时发现错误，通过子代码可以表明出错的原因
2	OPEN 报文错误	表明一个 OPEN 报文中有错误，通过子代码可以表明出错的原因
3	UPDATE 报文错误	表明一个 UPDATE 报文中有错误，通过子代码可以表明出错的原因
4	保持计时器超时	表明在协商好的时间间隔内没有收到 KEEPALIVE 或是 UPDATE 报文
5	有限状态机错误	当发生不可预见的错误时发送给对端路由器来终结 BGP 会话
6	停止	表示立刻终止 BGP 会话

每个通知报文包含了 3 个字段：错误代码、错误子代码和数据。错误代码字段指明了 NOTIFICATION 错误的类型，如果有的话错误子代码给出出错的详细解释，数据字段包含了和错误相关的诊断信息，不是所有的通知报文都有数据字段。

当处理 BGP 报头的时候如果发现错误，将会产生一个“报文报头错误”的通知报文。这个报文一般在以下几种情况下产生：收到一个有非法的 Marker 字段的 BGP 报头的时候，报文报头的长度值小于或是大于所需值的时候，或者是报文报头的类型不可知的时候。表 7-8 列出了报文报头出错提醒子代码以及相关描述。

表 7-8

报文报头出错提醒子代码

报文号	报文子代码类型	描述
0	无出错子代码	空值
1	连接不同步	表明 BGP 报文中 Marker 字段的值不是期望的值 OPEN 报文—全 1，除非使用了 TCP MD-5 其他在 OPEN 报文中协商
2	报文长度无效	报文报头的长度值小于或是大于所需值，错误值被放入报文的数据字段。 OPEN—最小 29 个八位组，最大 4096 个八位组 UPDATE—最小 23 个八位组，最大 4096 个八位组 KEEPALIVE—不大于或小于 19 个八位组（全空的 BGP 保活报文的大小）
3	报文类型无效	表明收到了未知类型的报文，类型代码值被放入数据字段

BGP 的 OPEN 报文出错可能有以下几种原因：失败的或是配置错误的 TCP MD-5 认证请求，被破坏的 TCP 数据包，或是 BGP 的配置问题。OPEN 报文出错包括的报文子代码描述了报错的原因。

表 7-9

OPEN 报文出错提醒子代码

报文号	报文子代码类型	描述
1	不支持的版本号码	表明 BGP 对等体使用的是不支持的 BGP 版本，数据字段包含了本地支持的最大 BGP 版本号
2	无效对等体 AS	对等路由器的自治系统号不是期望的值，这个错误一般是由某个对端路由器的配置错误导致的
3	无效 BGP 识别符	对等路由器的 BGP 识别符不是期望的值，这个错误一般是由某端路由器的配置错误导致的，这个值必须是个有效的 IP 地址
4	不支持的可选参数	本地路由器收到了不支持的可选参数
5	认证失败	当 BGP 认证失败时产生
6	无法接受的保持时间	本地无法接受保持时间值，任何保持时间都有可能被拒绝，保持时间必须由 BGP 对等体之间协商

报文。在处理 UPDATE 报文的时候可能有各种不同的错误发生，这些错误通常都是由路由器上的错误配置导致的。表 7-10 中列出了各种不同的“UPDATE 报文错误”通知报文以及相关描述。

表 7-10 “UPDATE 报文错误”提醒子代码

报文号	报文字代码类型	描述
1	属性列表格式错误	表明不可用的路由长度和 / 或全部属性长度加上固定的 UPDATE 报头的大小 [19] 加上全部路径属性长度字段的大小 [2] 加上不可用的路由长度字段 [2] 超长 另外一种可能是同样的属性在同一个 UPDATE 报文中出现多次
2	无法识别的公认属性	提示有一个未知的公认必遵属性，这个属性的值会放入报文的数据字段
3	公认属性短缺	提示缺少一个公认必遵属性，这个属性会放入报文的数据字段
4	属性标志错误	属性标志字段和属性代码字段不符，可能是由于错误的属性、错误的标志、错误的代码或是错误的值，相关信息也在报文的数据字段
5	属性长度出错	实际的属性长度和属性长度字段指定的长度不符，属性数据（属性类型、长度和值）会被放入报文的数据字段
6	无效的起源属性	起源属性的值没有定义或是无法识别，该值会被放入出错报文中
7	AS 路由环路	在 UPDATE 报文中含有本地自治系统号码，由此可以假设路由环路发生
8	无效的下一跳	下一跳地址是一个无效的 IP 地址，很可能是格式错误，出错报文中会包含该值
9	可选参数错误	表明一个已识别的可选属性的值出错，错误值将会放入报文的数据字段
10	无效的网络字段	说明在报文的 NLRI 字段有语法错误
11	AS 路径格式错误	AS 路径语法不正确

如果 BGP 会话没有任何错误，除非接口断掉或是 BGP 配置被修改，你不会看到任何通知报文。两个 BGP 对等体建立 BGP 会话后，它们会交换保活报文来验证 BGP 会话的完整性，下一小节会讨论 BGP 保活报文。

7.5.1 保活（keepalive）报文

当 BGP 会话成功地建立并且完成了 BGP 更新的发送和接收后，BGP 对等体会周期性地互相发送保活报文。默认的情况下 BGP 路由器每隔 60s 发送保活报文，通知相邻的对等体 BGP 连接仍然处于激活状态，保活报文的间隔可以从默认值改为 3~4 294 967 295 之间的任意数值，或是设为 0 表示不需要交换保活报文。将保活的值设为 1 或是 2s 是无效的，如果使用了非法的 KEEPALIVE 值，BGP 会话会失败，同时发出“Open failed: Connection refused by remote host”的通知报文，KEEPALIVE 的计时器值也可能被设置为默认是 180s 的协商得到的保持时间值的 1/3。图 7-16 显示了包括 KEEPALIVE 在内的 3 种 BGP 报文在成功建立 BGP 会话中的交互过程。

保活报文不包含任何数据，它只是一个 19 字节的 BGP 报头，这在范例 7-3 的协议分析中可以看出。

范例 7-3 BGP 保活报文

```
Frame Status Source Address Dest. Address Size Rel. Time Delta Time Abs. Time
Summary
10 [10.50.4.1] [10.50.4.2] 73 0:00:37.336 0.008.155 04/28/2002 03:14:50 PM
BGP: type =
```

(待续)

KEEPALIVE

DLC: ----- DLC Header -----

DLC:

DLC: Frame 10 arrived at 15:14:50.2443; frame size is 73 (0049 hex) bytes.

DLC: Destination = Station 000427228197

DLC: Source = Station 0004272281D8

DLC: Ethertype = 0800 (IP)

DLC:

IP: ----- IP Header -----

IP: IP: Version = 4, header length = 20 bytes

IP: Type of service = C0

IP: 110. = internetwork control

IP: ...0 = normal delay

IP: 0... = normal throughput

IP:0.. = normal reliability

IP:0. = ECT bit - transport protocol will ignore the CE bit

IP:0 = CE bit - no congestion

IP: Total length = 59 bytes

IP: Identification = 3 IP: Flags = 0X

IP: .0.. = might fragment

IP: ..0. = last fragment

IP: Fragment offset = 0 bytes

IP: Time to live = 1 seconds/hops IP: Protocol = 6 (TCP)

IP: Header checksum = 9C94 (correct)

IP: Source address = [10.50.4.1]

IP: Destination address = [10.50.4.2]

IP: No options

IP:

TCP: ----- TCP header -----

TCP:

TCP: Source port = 11002

TCP: Destination port = 179 (BGP)

TCP: Sequence number = 3817488906

TCP: Next expected Seq number= 3817488925

TCP: Acknowledgment number = 3816595191

TCP: Data offset = 20 bytes

TCP: Flags = 18 TCP: ..0. = (No urgent pointer)

TCP: ...1 = Acknowledgment

TCP: 1... = Push

TCP:0.. = (No reset)

TCP:0. = (No SYN)

TCP:0 = (No FIN)

TCP: Window = 16339

TCP: Checksum = 7BB6 (correct)

TCP: No TCP options

TCP: [19 Bytes of data]

TCP: BGP: ----- BGP Message -----

BGP:

BGP: 16 byte Marker (all 1's)

BGP: Length = 19 BGP: BGP type = 4 (KEEPALIVE)

BGP:

BGP:

ADDR HEX

ASCII

0000: 00 04 27 22 81 97 00 04 27 22 81 d8 08 00 45 c0 | ..'".....'.....E.

0010: 00 3b 00 03 00 00 01 06 9c 94 0a 32 04 01 0a 32 | .;.....2...2

0020: 04 02 2a fa 00 b3 e3 8a 42 0a e3 7c 9e f7 50 18 | ..*.....B...P.

0030: 3f d3 7b b6 00 00 ff ff ff ff ff ff ff ff ff | ?.{.....

0040: ff ff ff ff ff ff 00 13 04 |

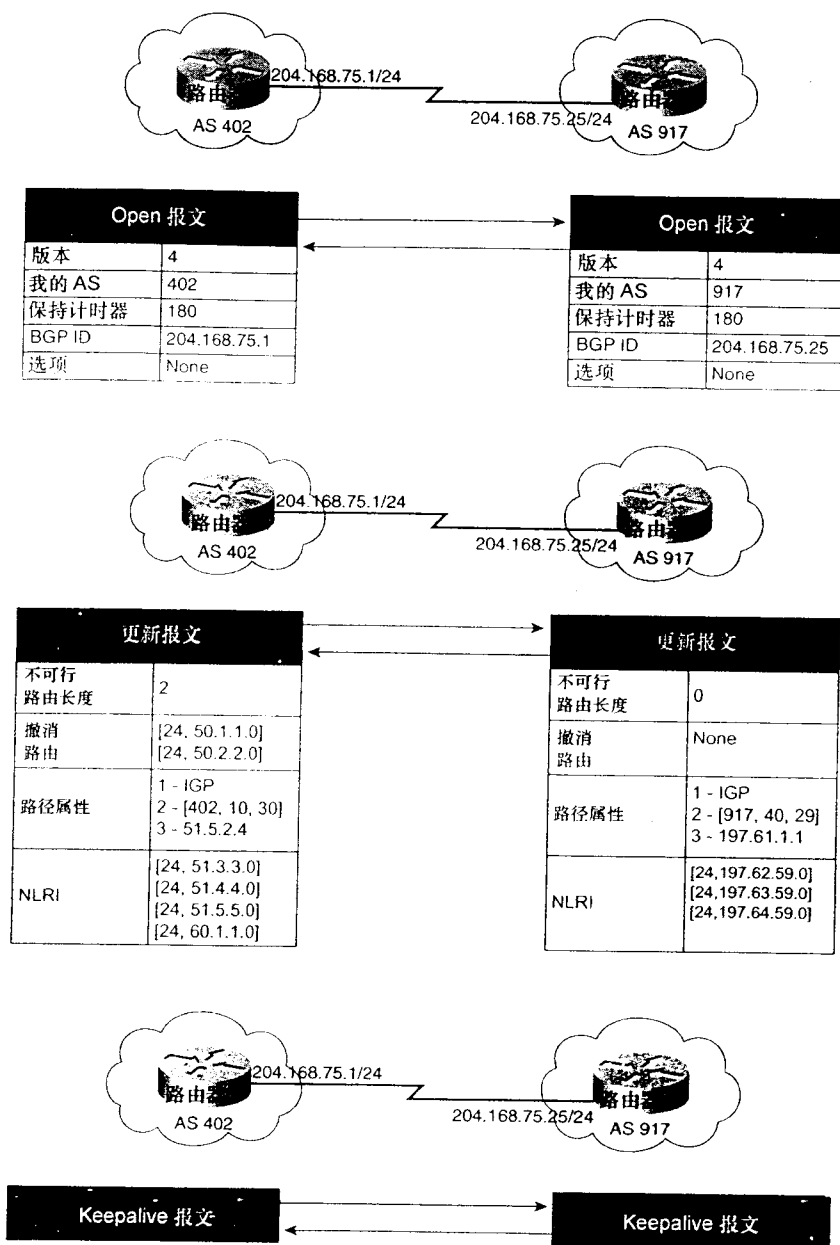


图 7-16 BGP 报文

7.5.2 路由刷新报文

在思科 IOS 软件版本 12.0 (6) T 之前，所有的 BGP 路由器每次本地的路由策略改变后都需要手工重启 BGP 会话。会话的重启可以使路由器接收处理远端对等体的路由更新时应用新的策略，在早期版本的思科 IOS 软件中，这个问题通过在每个对等体之间使用 BGP “*soft reconfiguration*” 来解决。当 BGP 软重配特性被加入到传统的配置上后，路由器会在内存中

为从每一个远端对等体收到的 Adj-RIB-In 表建立完全未加修改的备份，尽管这个特性通过防止 BGP 会话的中断而提高了网络的可靠性，但是同时也消耗大量的系统资源。每次使用命令 **clear ip bgp** *{*|ip-address [peer-group]}* **soft[in|out]** 都会触发软重配的发生，关于这个命令的具体使用办法在第 9 章中介绍。当使用这个命令后，本地的 BGP 路由器会假装自己从远端重新收到了完整的路由更新，实际上是利用保存在内存中的 Adj-RIB-In 的备份信息重新刷新 Loc-RIB 表中的路由。

RFC2918 中定义的 BGP 路由刷新能力在思科的 IOS 软件中一般称为 BGP 的 “*soft reset enhancement*”，该特性在较新的思科 IOS 软件中自动启用，BGP 路由器会在 BGP 会话初始化的能力交换部分协商是否启用，它能够不需要软重配就允许 BGP 路由器动态地向对端路由器请求或是发送路由更新。在 BGP OPEN 报文的可选能力字段中定义了 IANA 制定的 ROUTE-REFRESH 能力 (2)。在路由刷新报文被发送和理解之前，每个参加 BGP 会话协商的对等体都必须支持这个能力，如果某个不支持该能力的 BGP 路由器从对端收到了路由刷新报文，它会忽略这个报文并且记录下 “Unsupported OPEN Parameter” 后继续工作。当参加一个 BGP 会话的两台路由器都不支持 ROUTE-REFRESH 能力时，这两台路由器都将无法使用这个能力，不得不使用软重配或是手工重启会话来更新 Adj-RIB-In 表。当然，如果有时 ROUTE-REFRESH 请求由于某种原因不起作用，还是可以通过手工来重启会话。

7.6 BGP 状态机的操作

BGP 对等体在建立相邻关系和交换路由信息之前会经历多个不同的状态，在每个状态中，路由器必须发送和接收报文，处理报文数据并且为进入下一阶段初始化资源，这个过程就是 BGP 的 “*Finite-State Machine (FSM)*” 状态机。如果过程在任意一点失败，会话会被结束，对等体会返回空闲状态然后重新开始处理过程。每次会话终结的时候从相应的对等体学到的路由都会从路由表中删除，这样就会导致停工。如果是由于某个 BGP 对等体的配置错误引起的，对等体会持续地在未建立状态之间变化直到错误被排除。BGP 对等体在成功建立 BGP 会话之前会经历以下状态：

- 空闲 (Idle)；
- 连接 (Connect)；
- 激活 (Active)；
- Open 发送 (OpenSent)；
- Open 确认 (OpenConfirm)；
- 已建立 (Established)。

每个状态都有相关联的 *input events* (IE，输入事件)，输入事件是在 BGP 会话过程中发生的会触发动作的一些事件。

表 7-11

BGP 输入事件

事件号	事件名	描述
1	BGP 启用	在空闲阶段发生，BGP 启用事件表明 BGP 会话的开始，同时也为 BGP 进程初始化资源，BGP 启用事件只有在空闲状态才会被侦听，本地发言人在非空闲状态收到的 BGP 启用事件将会被忽略

续表

事件号	事件名	描述
2	BGP 终止	BGP 终止事件通知终结一个 BGP 会话
3	BGP 传输连接打开	本事件通知本地的发言人 TCP 连接已经打开，BGP 的资源初始化已经完成
4	BGP 传输连接关闭	本事件通知本地发言人远端的 BGP 发言人已经关闭了 TCP 会话，同时触发释放 BGP 资源，使得本地发言人返回到空闲状态
5	BGP 传输连接失败	本事件通知本地发言人与远端 BGP 发言人的 TCP 会话无法建立，同时触发释放 BGP 资源，使得本地发言人返回到空闲状态
6	BGP 传输严重错误	本事件通知本地发言人与远端 BGP 发言人的 TCP 会话发生严重错误，同时触发释放 BGP 资源，使得本地发言人返回到空闲状态
7	重连接超时	当重连接计时器超时的时候触发本事件，同时重新开始计数
8	保持时间超时	当保持时间超时的时候触发本事件，表示远端对等体没有应答本地对等体发送的报文
9	KEEPALIVE 超时	本事件表明 KEEPALIVE 计数器超时，也就是说在一定的时间间隔内没有从远端对等体收到保活报文
10	收到 Open 报文	本事件通知本地系统从远端对等体收到了 BGP OPEN 报文，BGP 会话可以进入 Open 确认状态
11	收到保活报文	本事件通知本地系统从远端对等体收到了一个 BGP 的保活报文，BGP 会话可以进入已建立状态
12	收到 Update 报文	本事件通知本地系统从远端对等体收到了一个 BGP 的 UPDATE 报文
13	收到通知报文	本事件通知本地系统从远端对等体收到了一个 BGP 的通知报文，BGP 会话应该被立刻终止

7.6.1 空闲状态

根据 RFC 1771 的定义，在每个 BGP 会话开始的时候，每个对等体路由器都必须经历不同的 BGP 状态。当配置了 BGP 后路由器进入的第一个状态就是空闲状态，在空闲状态，BGP 发言人路由器拒绝收到的 BGP 会话请求。在这个时候，路由器的 BGP 进程不拥有任何资源，路由器只有在收到 BGP 启用事件后才会给 BGP 进程分配资源，BGP 启用事件可以由 BGP 进程初始化或是由用户人工干涉产生。表 7-12 总结了空闲状态的行为表现和原因。

表 7-12 空闲状态行为表现

空闲状态行为表现	原因	
拒绝呼入会话请求	路由器刚刚配置完并且以前没有与该对等体建立过会话或者是 BGP 会话刚刚重启，直到一个 BGP 启用事件产生后呼入会话请求才不会被拒绝	
未分配 BGP 资源	新配置的 BGP 对等体会话或是会话重启，只有在收到一个 BGP 启用事件后资源才会被分配	
发送或是收到 BGP 启用事件	发送 BGP 启用事件后，BGP 对等体初始化它的资源，启用重连接计时器，试图与其他对等体建立 TCP 连接，同时侦听呼入的 TCP 连接尝试	
出错时	TCP 会话会被结束，路由器会保持在空闲状态，重启事件会重新发生，每当一个重启事件产生后，当前和最后一个重启事件之间的时间间隔会指数性增长	
从其他状态转变到空闲状态	激活状态	当其他未定义的错误发生时返回到空闲状态
	Open 发送状态	当以下错误发生时返回到空闲状态： OPEN 报文出错 BGP 停止事件（发送或是接收） 保持时间超时 其他未定义的错误

续表

空闲状态行为表现	原因	
	Open 确认状态	当以下错误发生时返回到空闲状态： 收到 TCP 连接中断通知 保持时间超时 收到通知报文 BGP 终止事件 其他未定义的错误
	已建立状态	当以下错误发生时返回到空闲状态： UPDATE 报文错误 收到 TCP 连接中断通知 收到通知报文 BGP 终止事件 保持时间超时 其他未定义的错误

BGP 启用事件最初发生在初始 BGP 配置之后，如果状态机从其他状态转变到空闲状态，下一个启用事件将会在 60s 后发生。为了防止路由器不停地连接和断掉 BGP 会话，每个启用事件之间的间隔是指数增长的。

当启用事件发生后，路由器初始化它的 BGP 资源，并且启用控制 TCP 连接尝试的频率的重连接计时器。在这个时候，路由器尝试和它已配置的 BGP 对等体建立 TCP 会话，同时也会侦听从 BGP 对等体接收到的 TCP 会话请求。如果 TCP 连接被关闭或是由于其他别的原因失败了，状态机将保持在空闲状态，BGP 启用事件之间的间隔将指数化增长，这就会大大加大 BGP 启用事件间的间隔。如果一切正常，状态机将会过渡到连接状态。图 7-17 显示了在 BGP 空闲状态过程中状态机经历步骤的逻辑流程。在这个图中，黑色文字框显示了进行的行动，灰色文字框显示了和进行的行动相关的 BGP 事件，白色文字框显示了发生的行动的具体细节。

注意：当两个 BGP 对等体路由器同时尝试建立 TCP 连接，或是当 BGP 会话启用后远端对等体试图重新建立一个新的连接时会发生 *Connection Collisions*（连接冲突）。在这种情况下，两个对等体将比较 BGP ID，由最高 BGP ID 的对等体建立的连接将会保持，其他的连接会被关闭。如果冲突发生在一个 BGP 会话已经启用的情况下，那么新的 BGP 会话请求将会被关闭，连接冲突只有在 Open 发送、Open 确认或是已建立的状态下才会被检测。

如果你发现一台路由器始终保持在空闲状态，那么可以检查以下方面：

- 验证在远端对等体上为本地对等体配置了正确的 IP 地址和自治系统号码。你可能需要修改 BGP 的更新来源或是 BGP 的路由器识别符，使得对等体看到的 BGP 请求来自正确的 IP 地址。需要记住的是 BGP 不接受来自未知的 BGP 对等体的连接。
- 验证在本地为远程对等体配置了正确的 IP 地址和自治系统号码。记住，BGP 验证 OPEN 报文的内容，如果从远端对等体收到的 OPEN 报文的内容和本地 BGP 配置不符，路由器将不会建立 BGP 的对等关系。
- 确认路由器可以互相访问配置的 IP 地址和 TCP 端口号 179。你可能需要添加路由，修改访问列表或是防火墙的规则配置来允许 BGP 对等体互相通信。

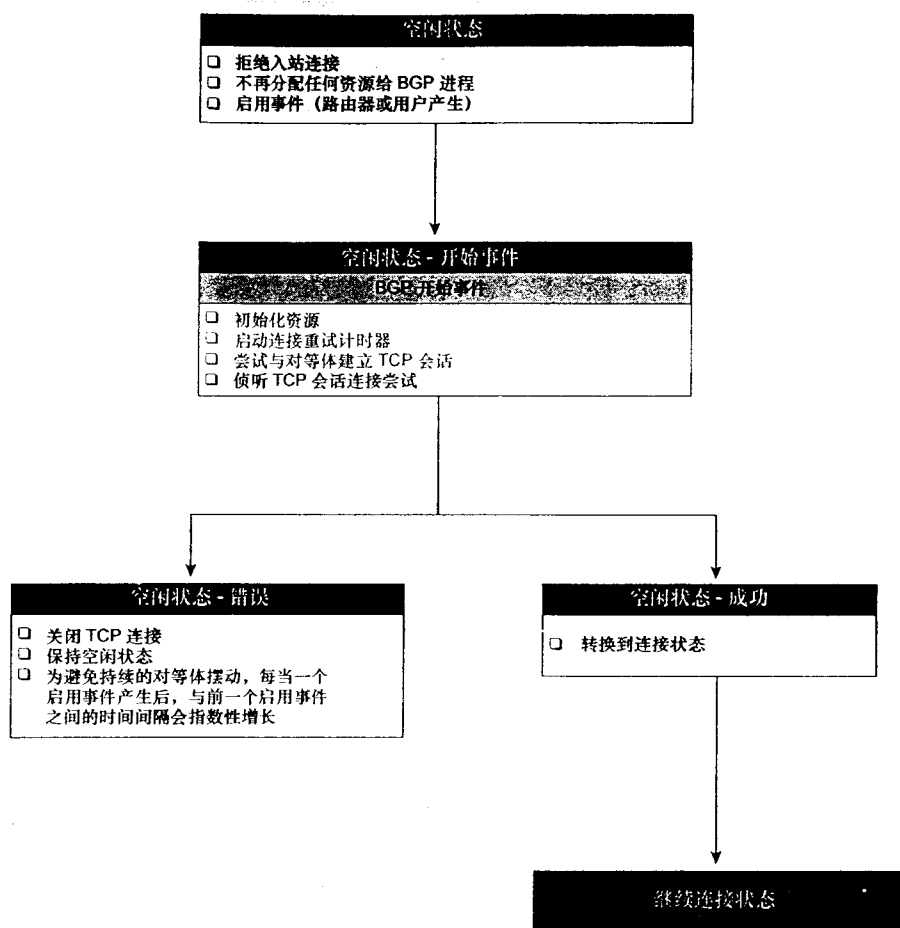


图 7-17 BGP 空闲状态

7.6.2 连接状态

在连接状态时，路由器等待和它的 BGP 对等体之间的 TCP 连接成功建立。在连接建立后，状态机会将重连接计时器清零，结束对 BGP 资源的初始化，然后发送 OPEN 报文给它的对端。表 7-13 列出了不同的连接状态行为和相关事件以及相关的状态过渡。

表 7-13

连接状态行为

连接状态行为	原因
忽略收到的启用事件	启用事件只有在空闲状态才会被接收和响应，在连接状态收到的任何启用事件都会被忽略
BGP 资源分配结束	路由器上的 BGP 进程开始工作，然而，只有在状态机进入已建立状态后才会进行路由
OPEN 报文被发送给对端	当 OPEN 报文被发送给对端后，路由器将进入 OPEN 发送状态
TCP 连接错误发生	重连接计时器被重置，路由器仍然继续侦听从它的对等体发来的 TCP 会话请求，但是它的状态会从连接状态变为激活状态

续表

连接状态行为	原因	
重连接计时器超时	重连接计时器被重置，路由器试图重新初始化与对等体的 TCP 会话，侦听从它的对等体发来的 TCP 会话请求，仍然保持在连接状态	
未定义错误发生	如果有任何其他的事件发生，路由器会释放 BGP 资源并且转变回空闲状态	
从其他状态转变到连接状态	激活状态	当路由器处于激活状态时重连接计时器超时，这个对等体将会： 重置重连接计时器 试图与远端对等体建立 TCP 会话 侦听远端对等体的 TCP 连接

在一个成功的 BGP 对等体会话过程中，对等路由器通常不会在转变到 OPEN 发送状态之前在连接状态花费过多的时间。图 7-18 显示了 BGP 连接状态行为和相关的原因，其中黑色文字框显示了进行的行动，灰色文字框显示了和进行的行动相关的 BGP 事件，白色文字框显示了发生的行动的具体细节。

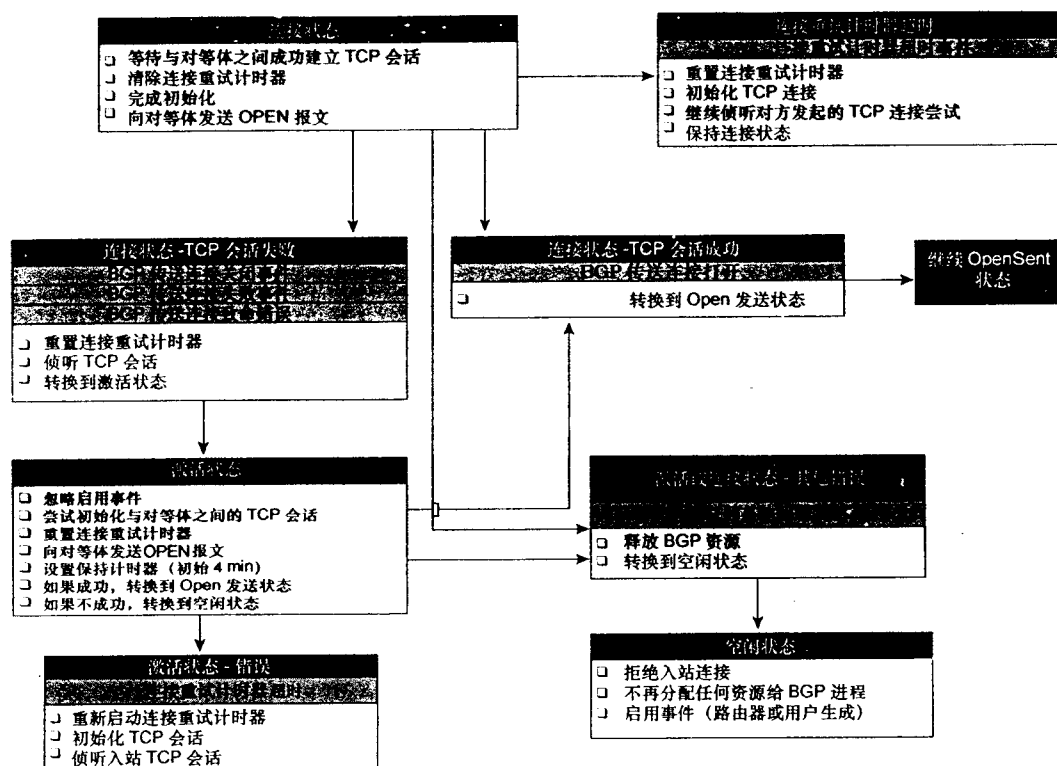


图 7-18 BGP 连接状态

如果两个 BGP 对等体之间的 TCP 会话由于任何原因被结束或是失败，状态机会重置重连接计时器，继续侦听它的对等体发送的 TCP 会话请求，同时进入激活状态。

当 BGP 对等体由于配置错误陷入连接状态无法自拔时：

- 始终确认在端口 179 上有进入和离开的 TCP 连接（同时在源端是比 1023 大的随机产生的 TCP 端口号），这样 BGP 会话才能在每个方向建立。BGP 的 TCP 会话使用随机的源端口号，TCP 目的端口号是 179。

- 验证本地和远端的 BGP 配置，检查 IP 地址和自治系统号有无输入错误，确认 BGP 路由进程编码正确。

7.6.3 激活状态

如果路由器无法和它的某个 BGP 对等体建立成功的 TCP 连接，路由器就会进入激活状态，这时 BGP 路由器会忽略启用事件（注意只有在空闲状态才侦听启用事件），试图和其他路由器建立 TCP 会话，同时重置重连接计时器。

如果当 BGP 路由器处在激活状态的时候成功地建立了 TCP 会话，它将发送一个 OPEN 报文给它的对等体，设置保持时间，以决定对等体需要等待返回报文的时间，然后进入 OPEN 发送状态。保持时间的初始值设为 4 min，当 BGP 会话成功地建立以后，保持时间的值会变为在 OPEN 报文处理时协商得到的值。

如果在重连接计时器超时之前还没有成功地建立 TCP 会话，状态机将重启重连接计时器，试图初始化一个 TCP 连接，同时在转换回连接状态的过程中继续侦听从对等体发来的 TCP 会话请求。

你可能会注意到在下面的情况下路由器在空闲和激活状态之间循环：

- BGP 对等体识别符配置错误；
- BGP 对等体无法通过 TCP 端口 179 访问；
- 网络拥塞导致重连接计时器超时；
- 网络接口抖动。

7.6.4 OPEN 发送状态

在 OPEN 发送状态，BGP 对等体等待从它的对等体发来的 OPEN 报文。当收到一个 OPEN 报文后，会检查报文的有效性，在这个时候，BGP 对等体会检查本地的配置和 OPEN 报文的所有字段是否匹配，任何不一致都会导致 OPEN 报文错误的发生。同时 BGP 对等体还会检查验证没有发生连接冲突。如果报文是有效的，对等体会发送一个保活报文给它的对等体，设置 KEEPALIVE 计时器，设置保持时间计时器，然后进入 OPEN 确认状态。表 7-14 列出了 OPEN 发送状态行为和相关的描述。

表 7-14 OPEN 发送状态行为

OPEN 发送状态行为	原因
忽略收到的启用事件	启用事件只有在空闲状态才会被接收和确认，在连接状态收到的任何启用事件都会被忽略。
等待对等体的 OPEN 报文	BGP 对等体会停留在 OPEN 发送状态，直到以下情况发生： <ul style="list-style-type: none">• 收到一个有效的 OPEN 报文• 发生 TCP 连接中断• 收到一个通知报文• 发生一个终止事件• 保持计时器超时• 任意其他未定义的事件

有很多事件可以导致 BGP 发言人从 OPEN 发送状态转变到空闲状态。如前所述，如果发言人从它的对等体收到了一个无效的 OPEN 报文，将会产生一个 OPEN 报文错误，这时本

地的路由器会发送一个通知报文指明出错的原因，然后转变到空闲状态重新开始连接进程。

当收到 BGP 终止事件、保持计时器超时或是发生其他意外事件的时候，本地路由器也会发送一个通知报文同时转变回空闲状态，然后重新启用新的成功的 BGP 会话。图 7-19 显示了在 OPEN 发送状态可能发生的各种各样的事件。

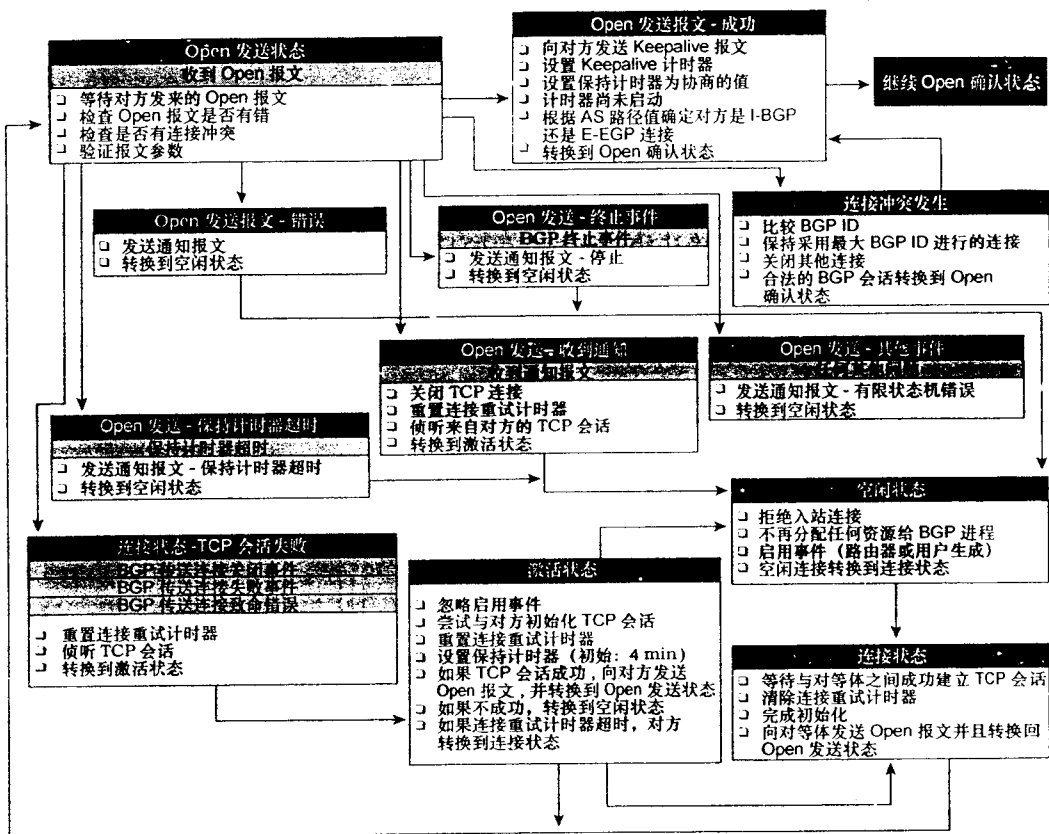


图 7-19 BGP OPEN 发送状态

BGP 对等体通常很少会在 OPEN 发送状态停留较长的时间，当本地路由器从它的对等体收到一个 OPEN 报文后，将给对等体发送一个保活报文然后转变到 OPEN 确认状态。

7.6.5 OPEN 确认状态

在 OPEN 确认状态，本地路由器等待从它的对等体收到一个保活报文，收到后 BGP 会话将转变到已建立状态。BGP 对等体可能会因为一系列的原因从 OPEN 发送状态转变为 OPEN 确认状态，表 7-15 列出了这些状态转变和 OPEN 确认状态的其他行为。

表 7-15

OPEN 确认状态行为

OPEN 确认状态行为	原因
忽略收到的启用事件	启用事件只有在空闲状态才会被接收和确认，在 OPEN 确认状态收到的任何启用事件都会被忽略

续表

OPEN 确认状态行为	原因
等待对等体的保活报文	BGP 对等体会停留在 OPEN 确认状态，直到以下情况发生： <ul style="list-style-type: none">收到一个有效的保活报文发生 TCP 连接中断收到一个通知报文发生一个终止事件保持计时器超时任意其他未定义的事件
如果 KEEPALIVE 计时器超时	KEEPALIVE 计时器可能会在保持计时器超时前 3 倍的保持时间重启，本地对等体将转变为空闲状态
如果对等体从 OPEN 确认状态返回空闲状态	BGP 连接被关闭 BGP 对等体会话的所有 BGP 资源被释放

图 7-20 显示了在 OPEN 确认状态可能发生的行动。本地路由器在收到一个保活报文后会成功地转变为已建立状态，或是在发生断开、终止或是通知事件的时候转变为空闲状态。

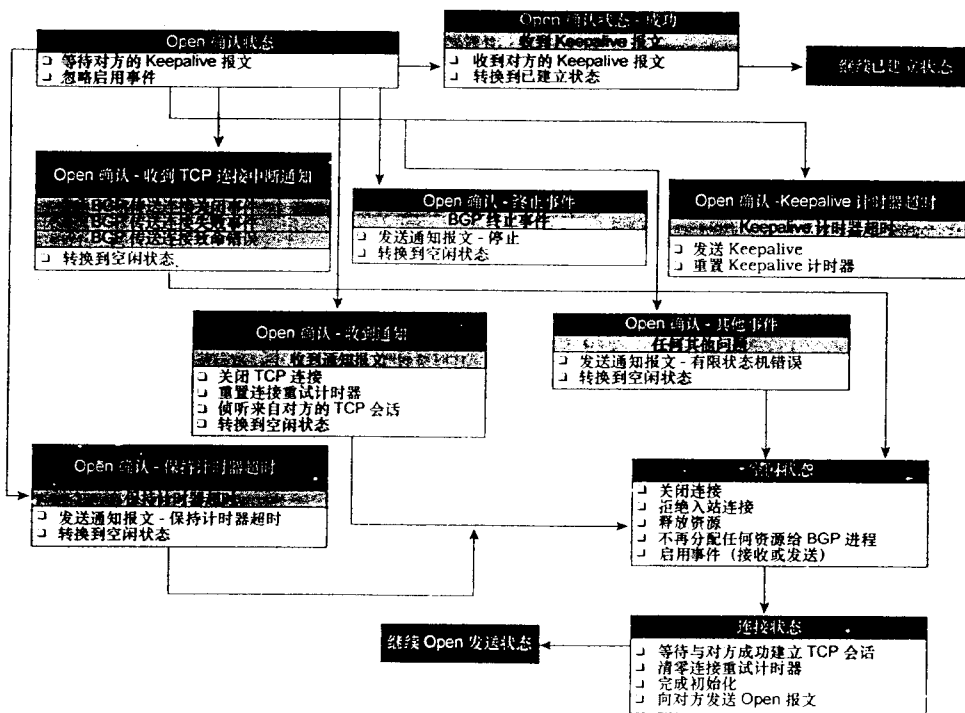


图 7-20 BGP OPEN 确认状态

BGP 对等体只会在 OPEN 确认状态停留一定的时间等待接收保活报文，如果没有在指定的保持时间内收到保活报文，会话将转变为空闲状态。

7.6.6 已建立状态

BGP 对等体在成功地交换了 OPEN 和保活报文后进入已建立状态，进入后它们会开

始发送包含路由信息的 UPDATE 报文和保活报文来验证 TCP 的连接状态。如果在对等体处于已建立状态的任何时候发生错误，本地对等体将发送一个包括错误原因的通知报文并且转变回空闲状态。图 7-21 显示了当发言人处于已建立状态时可能发生的各种各样的事件。

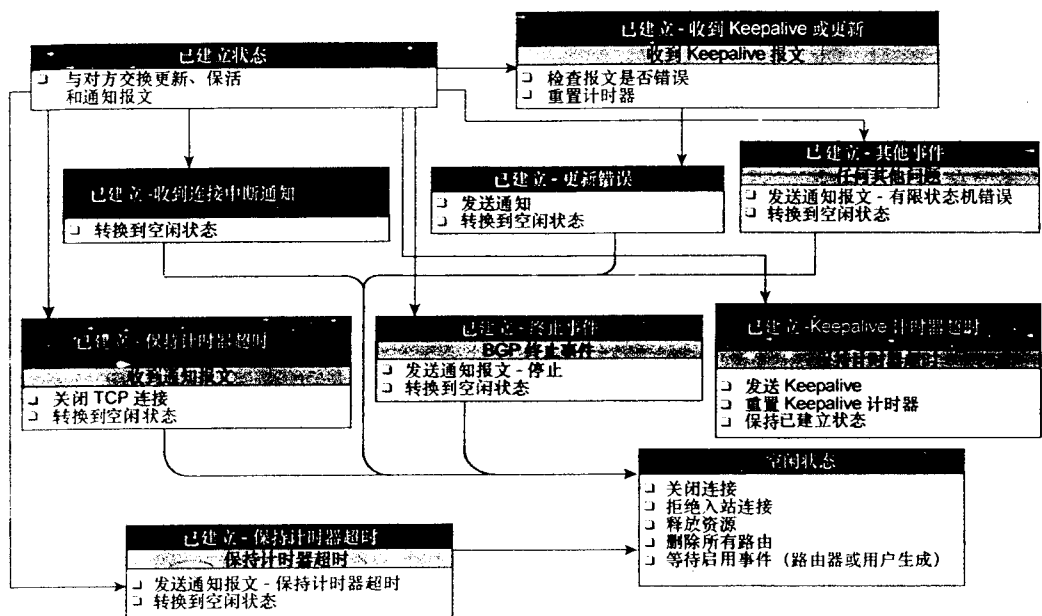


图 7-21 BGP 已建立状态

到目前为止我们已经介绍了基本的 BGP 运行过程，下面将讨论在 BGP UPDATE 报文中交换的各种各样的属性。

7.7 BGP 路径属性

BGP 路径属性包含了在 BGP UPDATE 报文中的路径的值，在 UPDATE 报文中包括的所有属性适用于 UPDATE 报文中 NLRI（网络层可达信息）字段指定的所有路径。

7.7.1 起源属性

路由的起源属性描述了路径被引入 BGP 的方式，它是公认必遵属性，也就是说所有的 BGP 实现都必须接受和理解起源属性的值，而且这个属性会传给其他的 BGP 对等体。表 7-16 列出了 3 种 BGP 起源代码。如果是从 I-BGP 学到的 BGP 路由，该路由的起源类型是 0：IGP，如果路由来自于外部网关协议（EGP）会话，那么路由的起源类型是 1：EGP，如果路由来自于未知（非 BGP）的路由进程，那么起源属性的值将是 3：不完整。

表 7-16

BGP 起源代码

起源代码	起源代码名	描述
0	IGP	路由来源于一个 BGP 路由器，这个路由类型包括了任意一条由 BGP 发言人的 BGP 进程产生的路由 在路由选择中 IGP 是最优先的起源类型，比 EGP 或是 Incomplete 要优先
1	EGP	路由来源于一个 EGP（外部网关协议，不是外部 BGP）会话 在起源类型中 EGP 比 Incomplete 优先
2	Incomplete	路由来源于非 BGP 的路由进程，通过手工重分发进入 BGP，比如说来自内部网关协议、静态路由或是直连路由 起源属性 Incomplete 没有 IGP 或是 EGP 优先

图 7-22 显示了起源属性为 0: IGP 的一条路由，它来自一个内部 BGP 会话。你会看到路由器 C 始发了到网段 10.2.1.0/24 和 10.2.2.0/24 的路由，所以从路由器 C 发给路由器 B 的 UPDATE 报文中可以看到路由器 C 赋予了值为 IGP 的起源属性给这些路由。

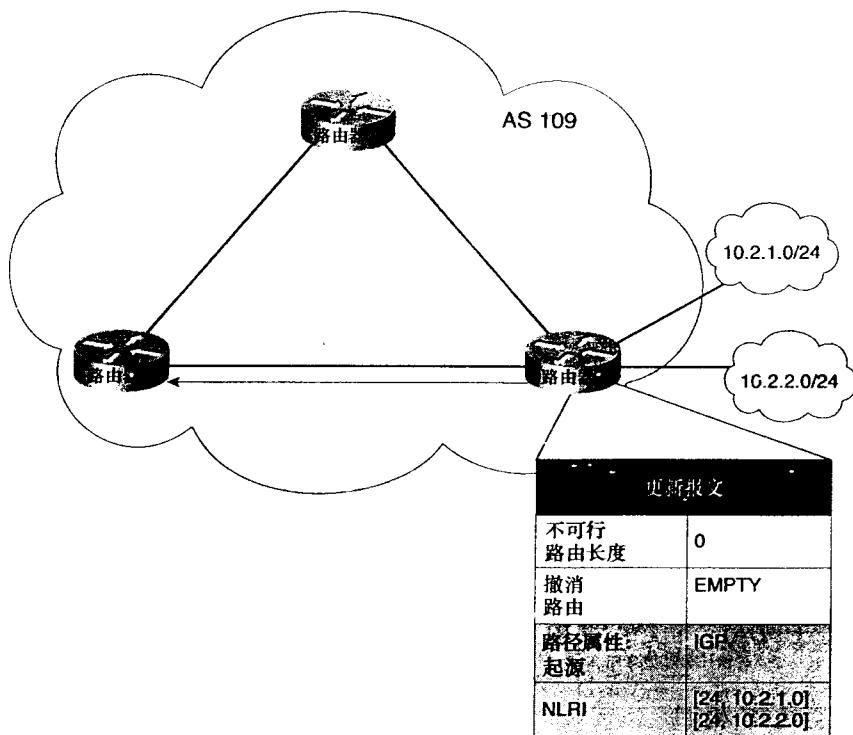


图 7-22 BGP 起源代码 IGP

图 7-23 阐述了如何使用 Incomplete 起源属性标识来源未知的路径。在图中，路由器 R 始发了自治系统 6565 的路由，但是由于这些路由是从 OSPF 进程中重分发的，所以发送出去路径的起源类型都是 Incomplete，每个下游的路由器都同样地转发这些起源类型是 Incomplete 的路径而不加修改。

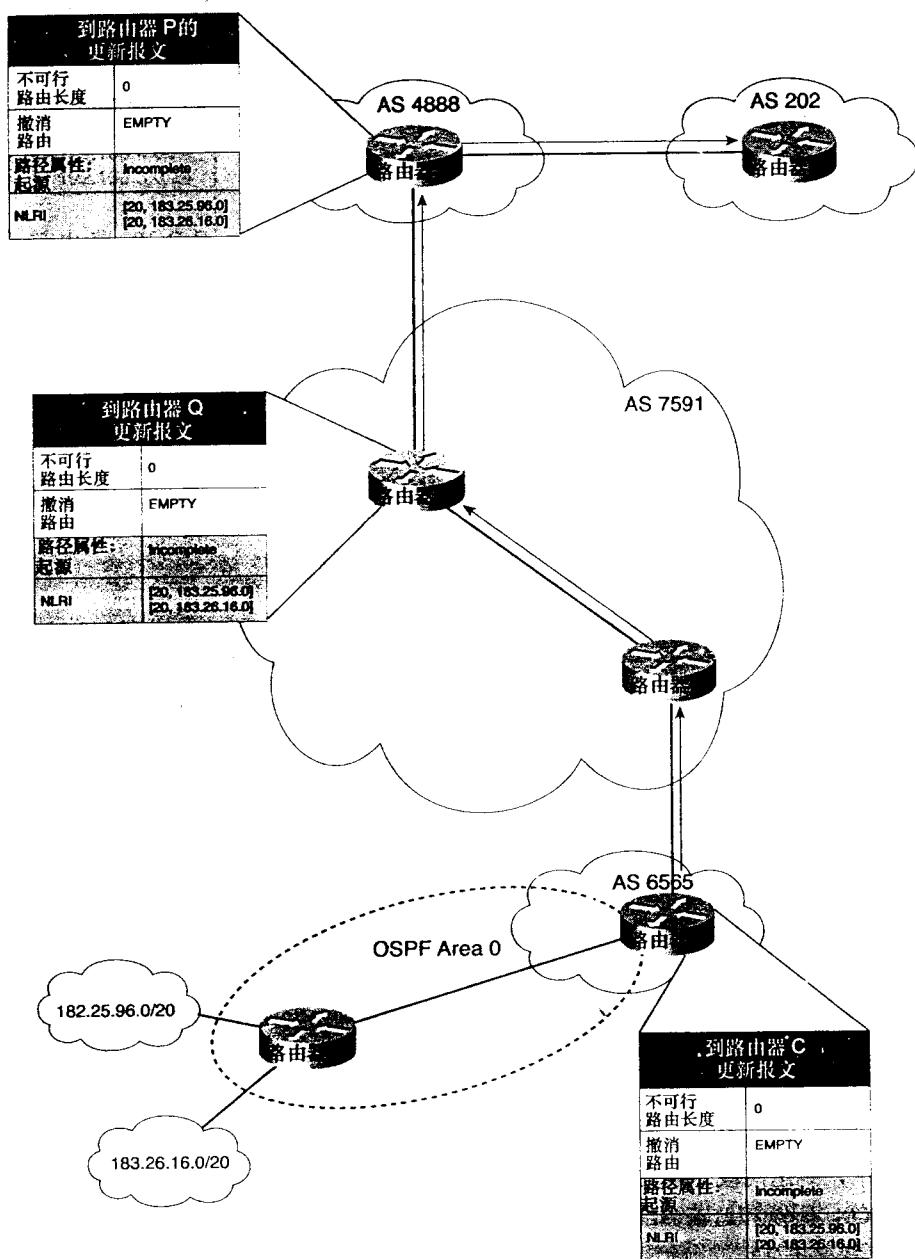


图 7-23 GP 起源代码 Incomplete

7.7.2 AS 路径属性

AS 路径属性是公认必遵属性，它描述了路由到达目的网段需要经过的路径。BGP AS 路径属性的主要目的是为了防止路由环路，当 BGP 对等体收到了在 AS 路径中包括本地自治系统号码的 UPDATE 报文时就知道发生了路由环路。当一个含有环路的更新收到后，UPDATE 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

报文将被忽略。

每个向 E-BGP 对等体发送已知路径更新的 AS 边界路由器都会在 AS 路径上加上它自己的 AS。AS 路径字段包括 3 部分：

- 路径段落类型，有两个可能的值：AS_SET 和 AS_SEQUENCE。
- 路径段落长度，指段落中 AS 的数量。
- 路径段落值包含 AS 号码的列表。

AS 路径的路径段落类型通常是 AS_SEQUENCE，每个 E-BGP 路由器在字段 AS_SEQUENCE 的最左边加上自己的自治系统号码。AS 路径包含了到达当前的自治系统已经穿过的自治系统的路径。图 7-24 描述了当路径段落类型是 AS_SEQUENCE 时 AS 路径的值是如何使用的。

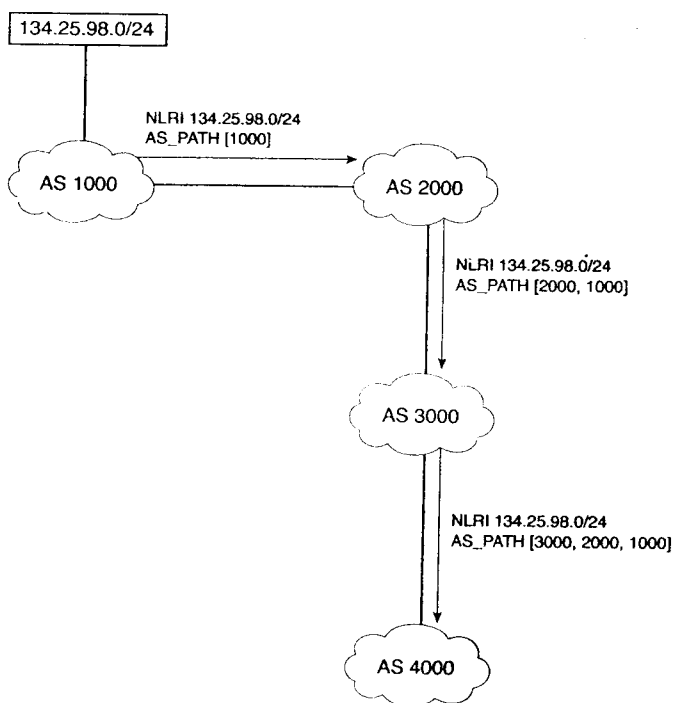


图 7-24 包含 AS_SEQUENCE 的 AS 路径属性

在范例中，到达网段 134.25.98.0/24 的路由起源于自治系统 1000。因为这条路由起源于自治系统 1000，所以和这条 NLRI 相关的 AS 路径中只有本地自治系统号码 1000。当自治系统 2000 收到更新并且它的自治系统 3000 边界路由器发送更新给它在自治系统 3000 中的对等体的时候，它会在 AS 路径中加入自己的自治系统号码，AS 3000 的边界路由器对它的自治系统 4000 中的对等体也会做同样的事情。这样 AS 路径中就包含了访问 134.25.98.0/24 网段必须经过的自治系统的序列，最左边的值是最近的自治系统号，最右边的是始发的自治系统号，中间是需要经过的自治系统。

AS_SET 值一般和聚约定时使用，当有不同的 AS 路径值的路由被聚合时使用路径段落类型 AS_SET。图 7-25 显示了在 AS 路径序列中如何使用 AS_SET 值来表示到达聚合网段 192.168.0.0/21 需要经过两个路径。

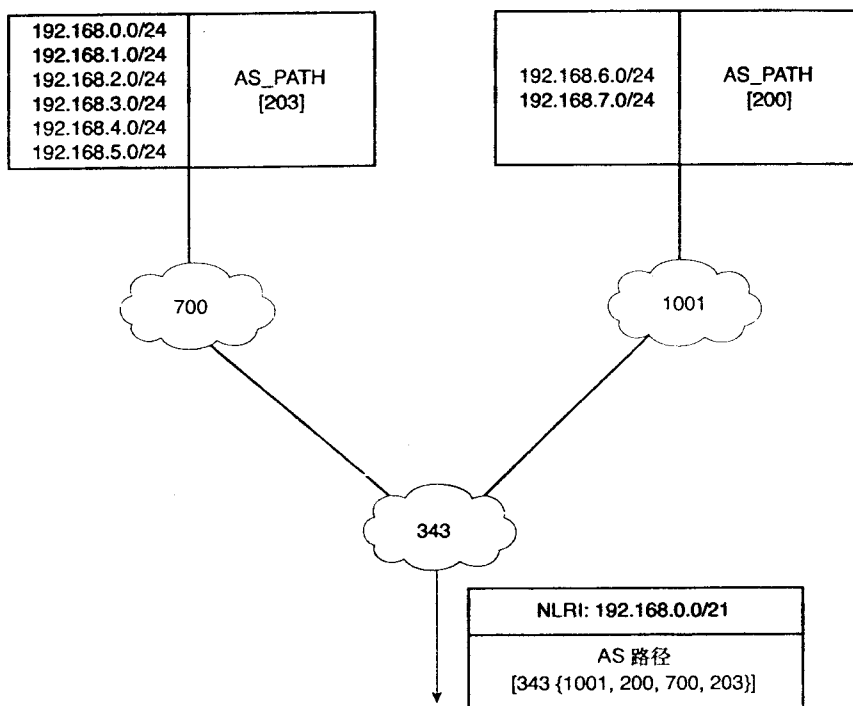


图 7-25 含 AS_SET 的 AS 路径属性

在这个范例中，AS 700 有 6 条路由：192.168.0.0/24、192.168.1.0/24、192.168.2.0/24、192.168.3.0/24、192.168.4.0/24 和 192.168.5.0/24，这些路由将被通告给在自治系统 343 中的 E-BGP 对等体，每条路由都起源于自治系统 203，当自治系统 700 的边界路由器给在自治系统 343 中的 E-BGP 邻居发送更新的时候会将自己的自治系统号码附加在 AS 路径上，所以自治系统 343 为了到达 192.168.0~5 将经过的完整的 AS 路径是[700, 203]。自治系统 1001 也同样地通告 AS 路径是[1001, 200]的网段 192.168.6.0/24 和 192.168.7.0/24。

当自治系统 343 会聚 192.168.0.0/21 范围的地址时，为了保持会聚路由的自治系统信息，必须使用 AS 路径段落代码 AS_SET 来列出到达目的网段无序排列的路径。

图 7-26 显示了当穿过不同的自治系统时网段 183.25.96.0/20 和 183.25.16.0/20 的 AS 路径属性是如何被修改的。

这个范例显示了路由器 R 从自己本地的 OSPF 路由进程学到了到网段 183.25.96.0/20 和 183.26.16.0/20 的路由，将起源设为 Incomplete，将 AS 路径的值设为本地的自治系统号 6565 发送给自治系统 7591。由于路由器 C 和路由器 A 同属于自治系统 7591，路由器 C 在通告路由给路由器 A 的时候不会附加自己的自治系统号码。由于路由器 A 发送路由给它的 E-BGP 邻居路由器 Q，所以路由器 A 会在 AS 路径上附加自治系统号码 7591。路由器 Q 收到了起源属性是 Incomplete 和 AS 路径是[7591, 6565]的路由后，将会在 AS 路径中加上自己的自治系统号码 4888 发送给自治系统 202 中的路由器 P。当自治系统 202 中的路由器想要访问网段 183.25.96.0/20 或是 183.26.16.0/20 时，它将遵循自治系统路径 4888、7591、6565，当数据包到达路由器 R 时，本地的 OSPF 进程会引导将它们发

送到路由器 M。

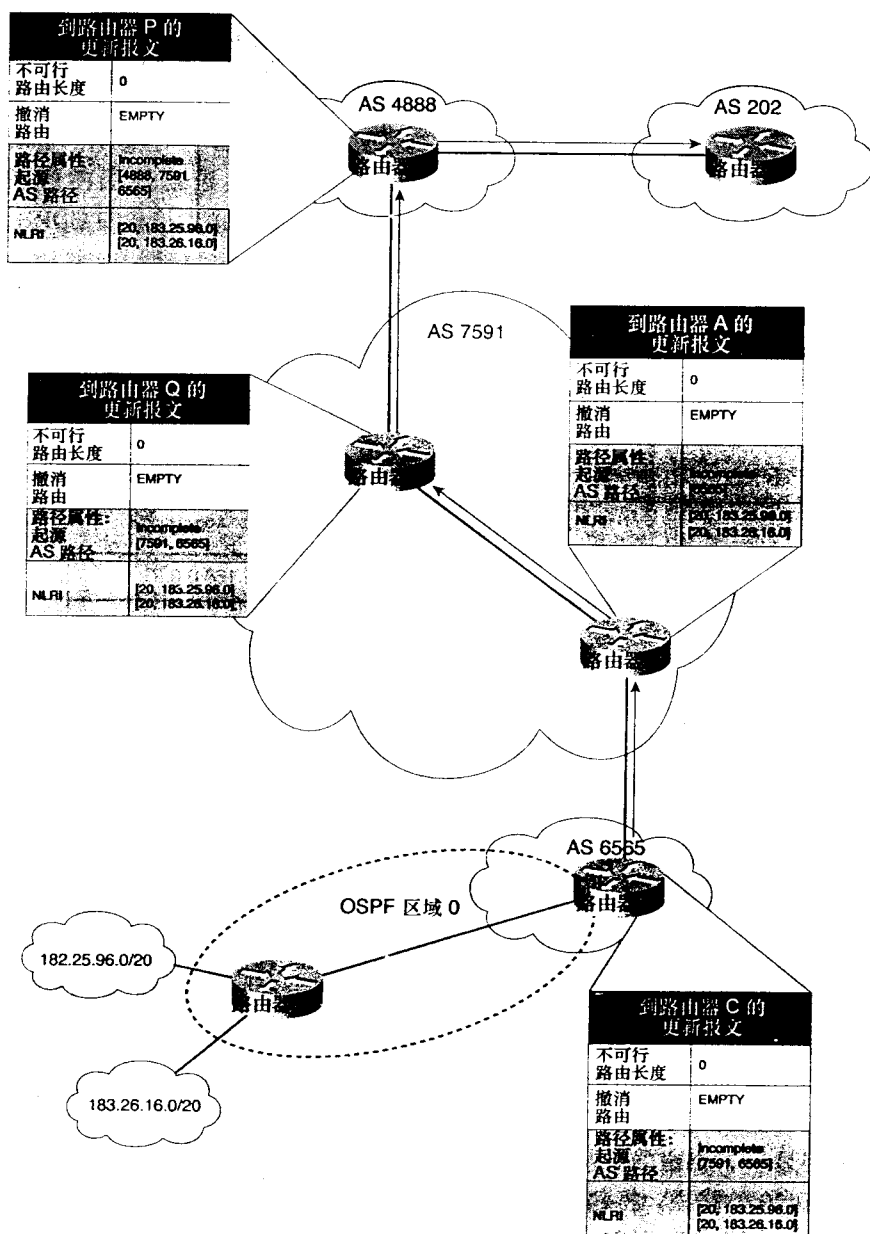


图 7-26 用起源和 AS 路径属性

7.7.3 下一跳属性

下一跳属性是公认必遵属性，指明了到达目的需要经过的下一跳 IP 地址。I-BGP 和 E-BGP 对下一跳的处理不同，因为前面提到过的同步原则，除非用 **next-hop-self** 命令特别指定，I-BGP

路由器不会修改下一跳属性，E-BGP 邻居会把下一跳地址修改为到达它们的 E-BGP 对等体的出口地址。在图 7-27 中，如果 Santa Fe 路由器想要到达 Roswell 路由器通告的任何网段，都必须使用 192.168.4.5 作为下一跳地址，同样 Roswell 路由器必须使用 192.168.4.4 作为下一跳地址来到达网段 207.23.12.0/22 和 207.23.24.0/22。

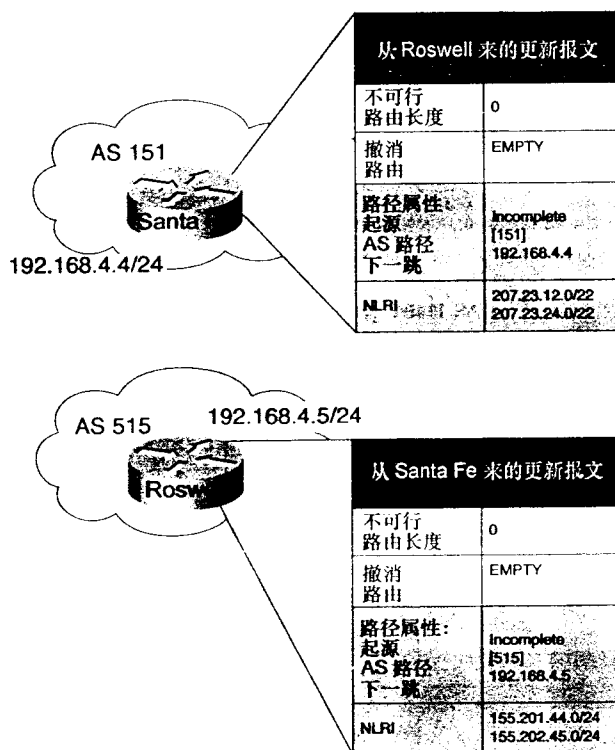


图 7-27 E-BGP 对等体和下一跳

为了使 I-BGP 对等体能够访问别的 I-BGP 对等体通告的路由，下一跳地址必须在主路由表中存在路由可达。如果由于某些原因 I-BGP 对等体没有到达下一跳地址的路由，可以使用 **next-hop-self** 命令来改变发送给那个对等体的 UPDATE 报文中指定的下一跳地址。

图 7-28 演示了 I-BGP 对等路由器之间是如何使用下一跳属性的。在这个范例中，自治系统 7995 中东部路由器与北部和西部路由器建立对等关系，同时也与自治系统 8245 中的南部路由器建立 E-BGP 会话，南部路由器通告网段 147.50.0.0/18 给东部路由器，东部路由器收到更新后将不加改变地转发路由给它的 I-BGP 对等体北部路由器。由于东部路由器没有修改 NLRI 147.50.0.0/18 的下一跳属性，所以这条路由的下一跳地址是自治系统 8245 的出口地址 217.200.8.1。因此北部和西部路由器会发发现到网段 147.50.0.0/18 的路由的下一跳地址是 217.200.8.1，由于下一跳地址是不可达的，所以这些路由器就不会将不可达路由通告给 E-BGP 对等体，也不会存入本地的主路由表。

然而，图 7-29 显示了如何在东部路由器上使用 **next-hop-self** 命令来避免这种情况。当命令执行后，西部路由器通告下一跳地址是 204.168.52.1 的 147.50.0.0/18 路由给北部路由器，

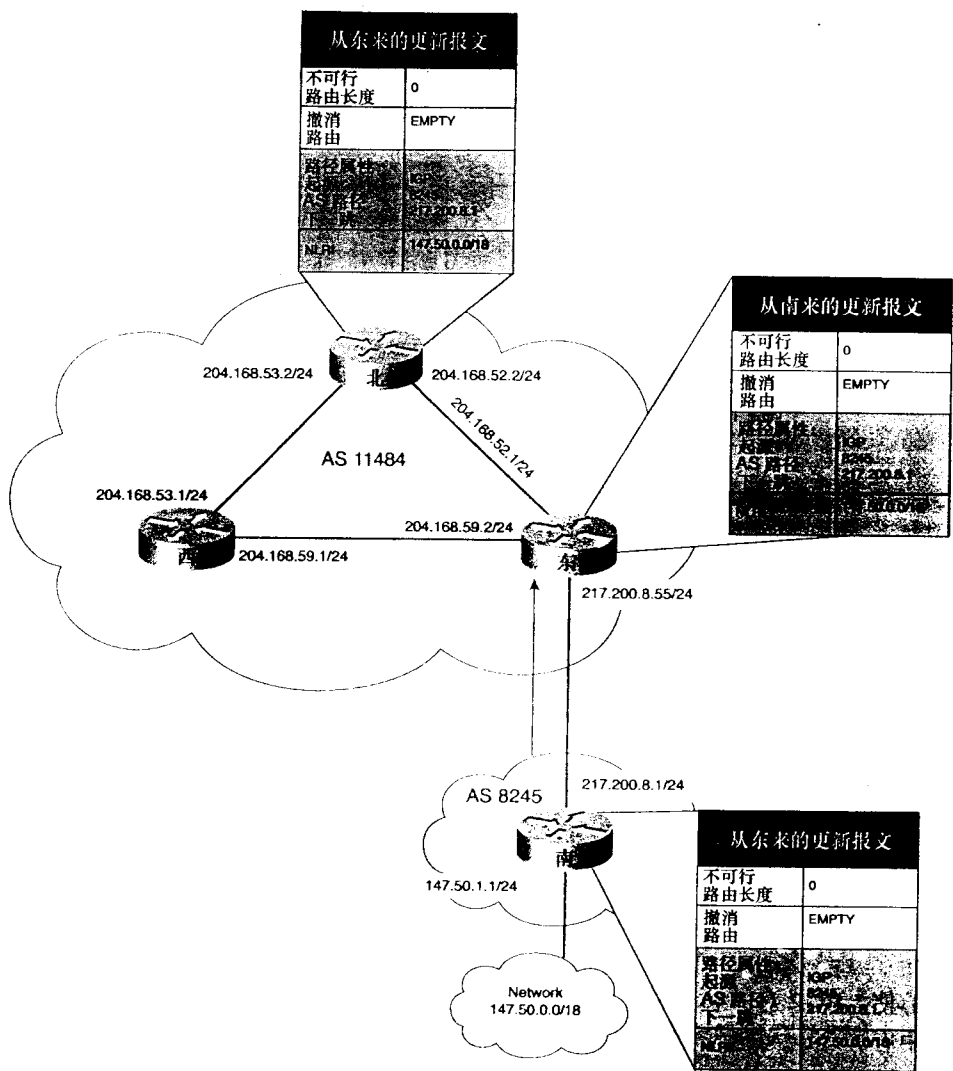


图 7-28 下一跳属性和 I-BGP 对等体

同时也通告下一跳地址是 204.168.59.2 的相同路由给西部路由器。由于这些下一跳地址都是可达的，北部和西部路由器接受这条路由，然后通告给相邻的 E-BGP 路由器，同时放入本地主路由表中。

7.7.4 多出口鉴别器 (MED) 属性

多出口鉴别器 (MED) 属性是可选非过渡属性，当存在到某个网段的多个入口的时候作为量度来指明最佳入口路径。多出口鉴别器属性是一个度量值，基本用来发送给其他相邻的自治系统关于优选的网络入口的信息。多出口鉴别器的取值范围是从 0~4 294 967 295，值越小越优选，多出口鉴别器的配置是基于邻居的，默认值为 0。多出口鉴别器属性只在相邻的

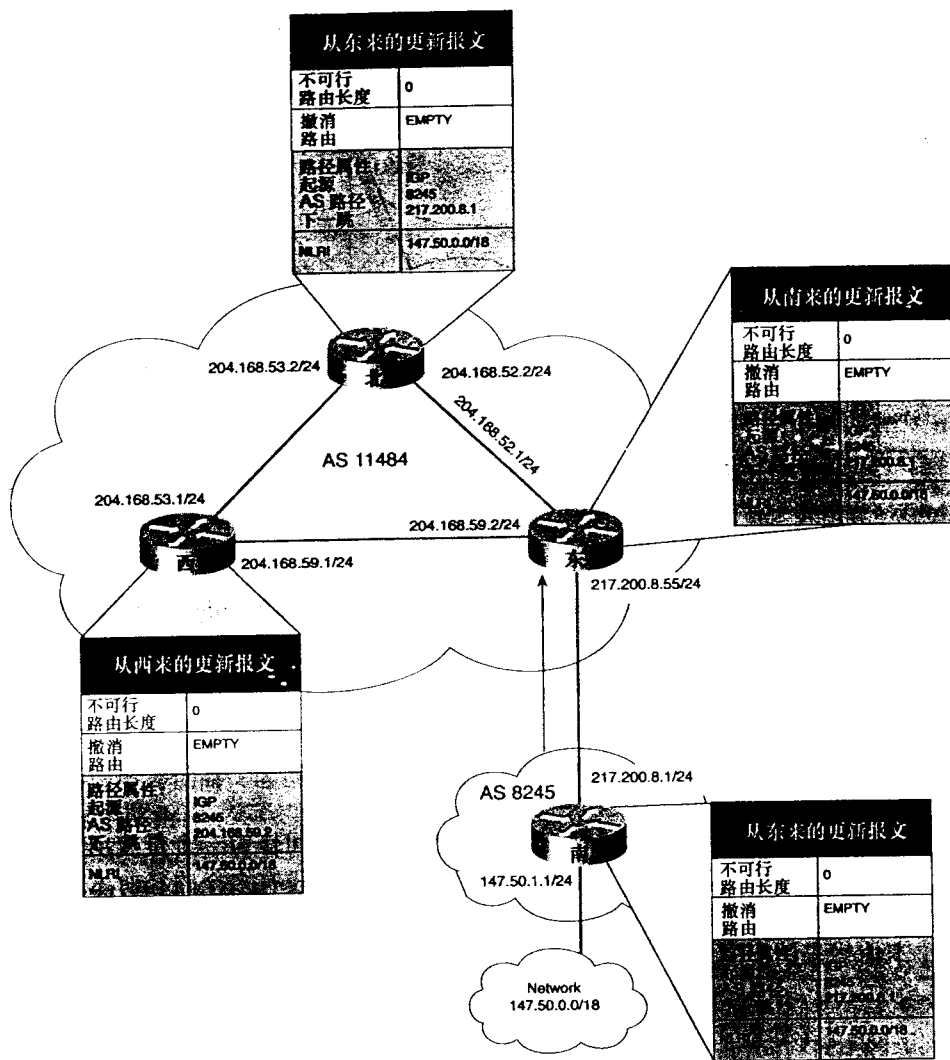


图 7-29 下一跳属性和 I-BGP 对等体

自治系统内部传送，只有当外部对等体属于同一个相邻的自治系统的时候才会比较多出口鉴别器值，这个量度只应用于配置的外部对等体之间的连接。在使用多出口鉴别器属性之前，应该先向你的服务提供商咨询，并且询问它们是否接受多出口鉴别器属性以及使用多出口鉴别器的优选方式。

图 7-30 显示了如何在自治系统 3898 和 8021 之间使用多出口鉴别器属性。在这个范例中，自治系统 3898 有两个出口，一个是在 Edge 1 和因特网路由器之间，网段是 211.146.2.248，使用 DS3 的连接；另外一个在 Edge 2 和因特网路由器之间，网段是 211.146.2.252，使用 T1 的连接。为了让自治系统 8021 中的因特网路由器优选 DS3 连接沿着 Edge 1 通告的路径去访问网段 123.45.67.0/24、123.45.68.0/24 和 123.45.69.0/24，在与因特网路由器通过 T1 连接使用 211.146.2.252 网段的 Edge 2 路由器上通告的多出口鉴别器值为 50。Edge 1 路由使用默认为

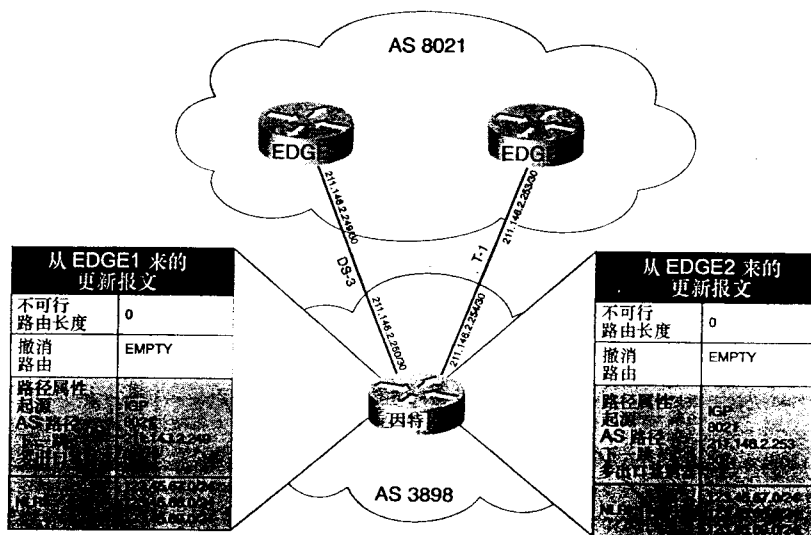


图 7-30 采用多出口鉴别器属性选择路径

0 的多出口鉴别器值通告同样的路由，当因特网路由器从 Edge 1 和 Edge 2 路由器收到路由后会优选从 Edge 1 路由器得到的路由，这是因为从 Edge 1 得到的路由有着较小的多出口鉴别器值。

7.7.5 本地优先 (LOCAL_PREF) 属性

本地优先属性是个公认自决属性，在 I-BGP 对等体之间作为一个量度用来指明在到目的网段的多个路径中优选的路径。当到一个外部目的网段有多个路径时本地优先属性用来指明对路径的优选程度，本地优先的取值范围是 0~4 294 967 295，和多出口鉴别器一样，本地优先的配置也是基于邻居的，它的默认值是 100，而且不会被传送给 E-BGP 对等体。

图 7-31 演示了如何使用本地优先来指明通过多个服务提供商到达因特网的优选路径。自治系统 3679 有两个因特网边界路由器：Internet 1 和 Internet 2，每个因特网边界路由器分别连接到不同的因特网服务提供商，在图中分别为运营商 1 和运营商 2。

运营商 1 和运营商 2 路由器通告同样的 3 条路由：123.45.67.0/24，123.45.68.0/24 和 123.45.69.0/24。因特网边界路由器 Internet 1 和 Internet 2 将这些路由转发给直接连接的 BGP 对等路由器 DC-01 和 DC-02，然而，Internet 1 在地理位置上靠近 DC-01，同样 Internet 2 靠近 DC-02，因此，除非 DC-01 与 Internet 1 之间的链路中断，DC-01 应该优选和使用从 Internet 1 发来的路由，类似的情况也应用于 Internet 2 和 DC-02。为了达到这个效果，当 Internet 1 发送路由给 DC-01 时，它将本地优先值从 0 改为 150，发送给 DC-02 的路由保持不变仍为默认值 100。通过这种方式，除非 Internet 1 和 DC-01 之间的连接中断，DC-01 总是会优选从 Internet 1 处学到的路由，如果连接真的失败，将会使用从 Internet 2 处得到的路由，这种方式同样适用于 DC-02 和 Internet 2。I-BGP 对等体总是优选具有最高的本地优先的路由。由于 Internet 1 和 Internet 2 之间的连接没有修改本地优先，它们各自总是优选

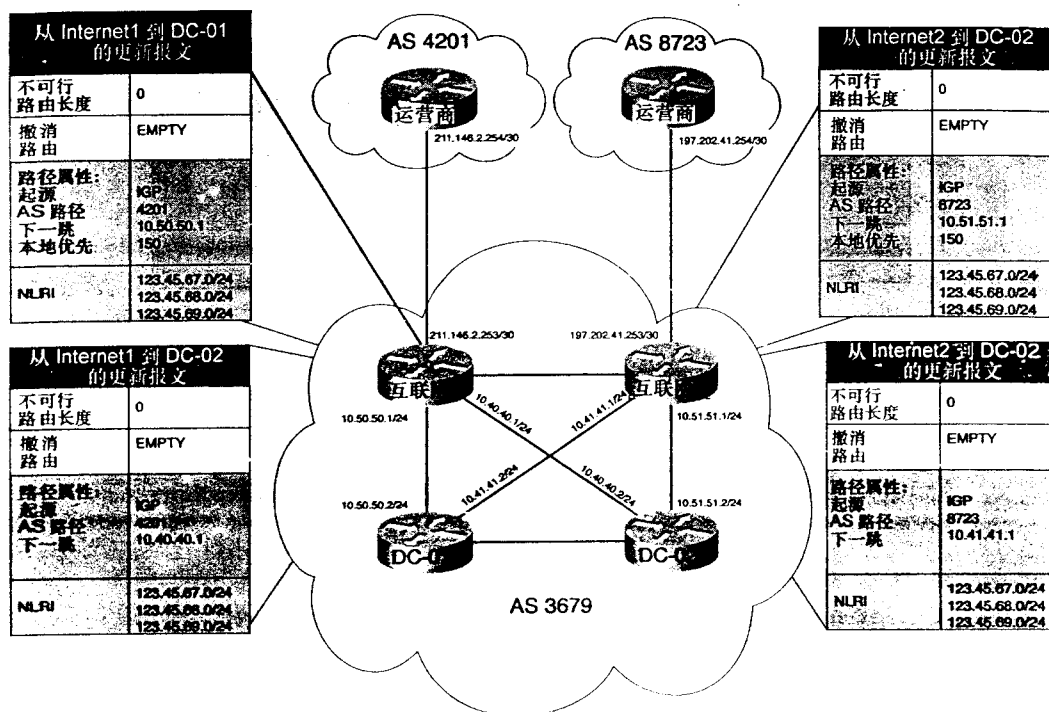


图 7-31 采用本地优先属性选择路径

从它们的上游服务提供商（运营商 1 和运营商 2）学到的 123.45.67.0/24、123.45.68.0/24 和 123.45.69.0/24 的路由。

7.7.6 权重（WEIGHT）属性

权重属性是本书中讨论的惟一只适用于思科路由器的属性。权重属性是当一个目的网段有多条出口路径时指定某条优选路径的另外一种方法。大的权重值比小的权重值优选，从相邻对等体收到的路由默认权重是 0，本地产生的路由的默认权重是 32 768，权重的取值范围是从 0~65 535。权重属性不会传递给任何路由器，无论是 E-BGP 还是 I-BGP，它严格来说是应用于本地 BGP 表中的路由的本地 BGP 策略。

注意：由于权重值是 BGP 路径选择过程中考虑的第一个项目，当创建本地 BGP 路由策略的时候修改权重属性会是一个非常有用的工具。

注意：到达一个目的网段可以使用多条路由并且可以在这些路由之间负载均衡。通过配置 **maximum-paths** 命令，可以设置使用多达 6 条路径到达同一个目的网段。

图 7-32 显示了当 BGP 表中到某个网段存在一条以上路由的时候如何修改权重属性来指定优先路由。在这个范例中，Factory 路由器通告给 Engineering 路由器两条可以到达 10.7.8.0/24 网段的路径。

在这种情况下，Engineering 路由器应该优选通过快速路由器的路由而不是通过慢速

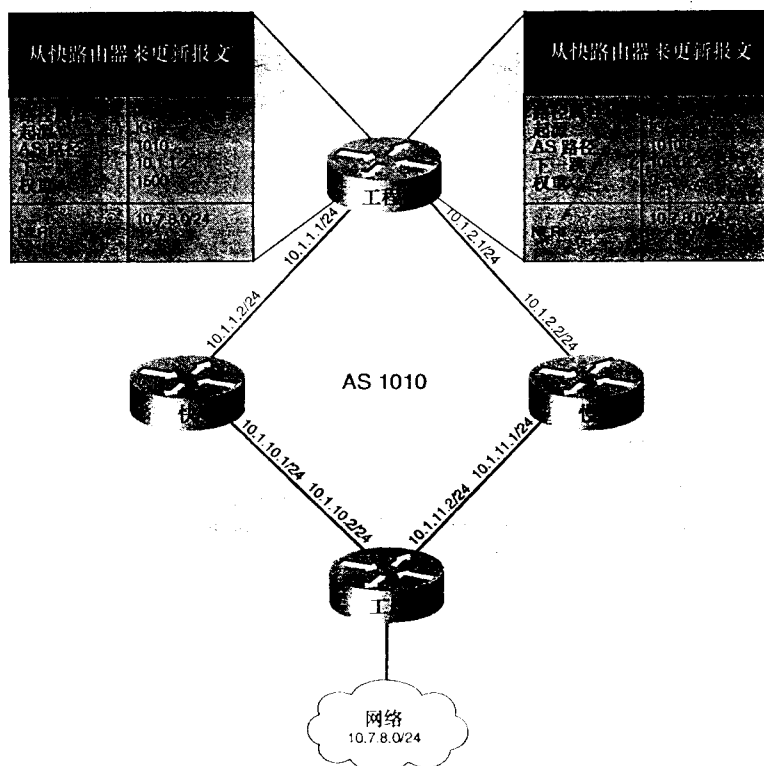


图 7-32 用权重在本地优选路由

路由器的路径。由于决定选择通过快速路由器而不是通过慢速路由器的路径是 Engineering 路由器的本地决策，所以我们可以修改从快速路由器得到的 10.7.8.0/24 路由的权重属性值为 1500。如果快速路由器失败，Engineering 和 Factory 路由器之间的流量仍然可以使用通过慢速路由器的路径，这是由于没有修改通过慢速路由器的路径的权重属性，仍然为默认值 0。

7.7.7 原子聚合 (ATOMIC_AGGREGATE) 属性

原子聚合是公认自决属性，用来通知下游的邻居丢失了一个特定路由的路径信息。当更精确的路由被会聚为不够精确的路由的时候会引起信息丢失，原子聚合属性只是 UPDATE 数据包中设置的一个标志位，它提醒下游路由器在聚合的过程中丢失了一些路径信息。当原子聚合属性被设置后，下游路由器不能删除这个属性或是发送到目的网段的更精确路由。

图 7-33 显示了一个范例，关于如何使用原子聚合属性来提醒 Showroom 路由器 Warehouse 路由器会聚了到网段 10.1.0.0/21 的 NLRI。Warehouse 路由器设置了原子聚合属性，通知 Showroom 路由器由于路径信息丢失，所以不允许发送更精确的到 10.1.0.0/21 网段的路由。

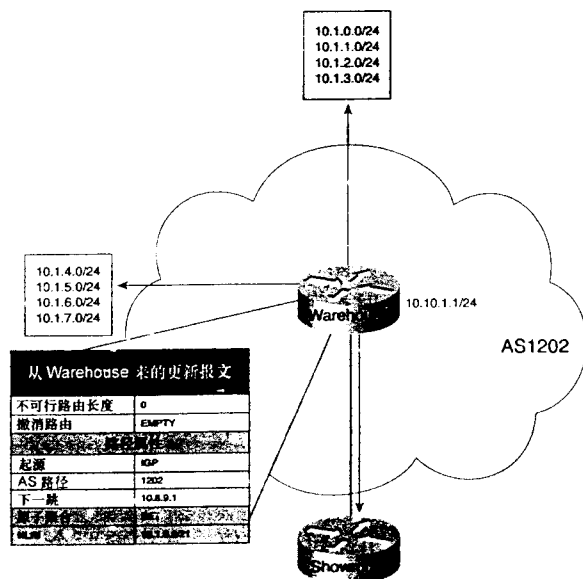


图 7-33 原子聚合属性

7.7.8 聚合者属性

聚合者属性是一个可选过渡属性，一般针对某个 NLRI 与原子聚合属性同时使用。聚合者属性包含了会聚路由的发言人的相关信息，属性中包含了创建会聚路由并且标记原子聚合属性的路由器的 BGP 识别符和自治系统号码。这些信息指明了非精确会聚路由的来源，可以用来找到更精确路由的源头。

图 7-34 显示了图 7-33 中到网段 10.1.0.0/21 的路由使用了聚合者属性。在这个范例

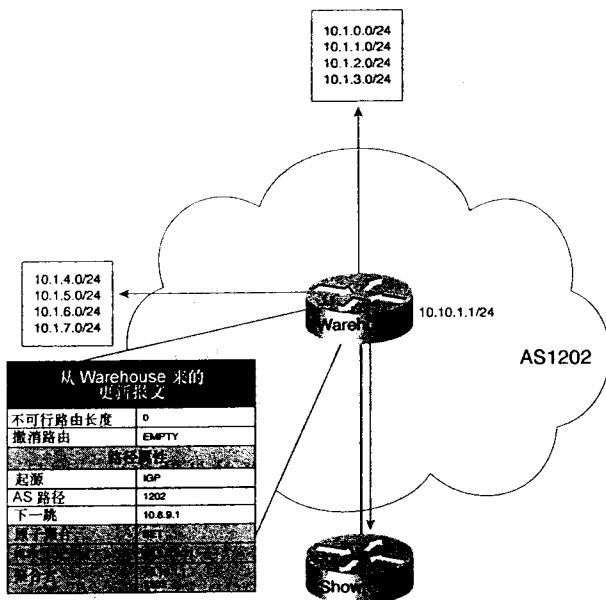


图 7-34 聚合者属性

中，加入的聚合者属性表明是自治系统 1202 中 BGP 识别符为 10.10.1.1 的路由器聚合了路由。

7.7.9 BGP 团体属性

BGP 团体属性在 RFC 1997 中定义（使用方法定义在 RFC 1998 和 2519 中），它是一个可选过渡属性，定义了遵循相同策略的组。团体拥有的策略影响了团体中的路由器对接收到的路由的处理，是接受还是拒绝。它们同时也可以用于指定对某个特定路由的优选，必须通过本地配置来指定路由属于某个团体，默认是所有支持团体属性的 BGP 发言人都属于 Internet 团体。如果收到了带有未知团体属性的路由，那么可能会增加一个新的团体。如果收到了一条已经设置了团体属性的路由，也有可能修改这个团体属性。由于 BGP 发言人不会自动转发团体属性，在发送团体属性给 E-BGP 对等体之前，必须首先和对方的相关人员合作协调被提议的团体属性使用方式。

BGP 团体属性是一个 32 位共 4 个八位组的值，其中前两个八位组是本地自治系统号码，后两个八位组是本地定义的值。团体属性可以有 3 种方式定义：十进制方式，取值范围是 1~4 294 967 295；十六进制方式，以 *aa: nn* 的格式表示，前面是本地自治系统号码，后面两个八位组是本地定义值；第三种方式是使用名字，使用公认的 BGP 团体名字之一。

表 7-17 列出了各种团体属性值和它们的相关描述。

表 7-17

公认 BGP 团体值

团体值（十六进制）	团体值（十进制）	团体名字	描述
0x00000000 到 0x0000FFFF	0~65535	保留	本范围的团体属性值被 IANA 保留
0xFFFF0000 到 0xFFFFFFFF	4294967041 ~ 4294967295	保留	本范围的团体属性值被 IANA 保留
0	0	Internet	默认的团体，所有支持 BGP 团体属性的路由器都属于本团体
0xFFFFFFF01	4294967041	NO_EXPORT	带有本团体属性的路由不会被通告给本自治系统或是联盟外的路由器
0xFFFFFFF02	4294967042	NO_ADVERTISE	带有本团体属性的路由不会被通告给任何其他对等体
0xFFFFFFF03	4294967043	LOCAL_AS	带有本团体属性的路由不会被通告给任何其他联盟外的路由器，请参考 RFC 1997 中的 NO_EXPORT-SUBCONFED

图 7-35 显示了如何使用 NO_EXPORT（0xFFFFFFF01）团体来防止内部网络路由被通告给公共互联网。在本范例中，边界路由器给路由 158.203.10.0/24、158.203.20.0/24 和 158.203.30.0/24 加上 NO_EXPORT 属性然后发送给路由器 ISP.com，当路由器 ISP.com 收到这些路由后可能会转发给它自己的自治系统 2501 内的其他路由器，但是自治系统 2501 内没有路由器会将这些路由转发出本地的自治系统。

在本章的后面我们会介绍两个只和路由反射器相关的属性：集群列表（CLUSTER-LIST）和起源者识别符（ORIGINATOR_ID）。在介绍完 BGP 路由反射器的运行后我们会介绍这些属性。

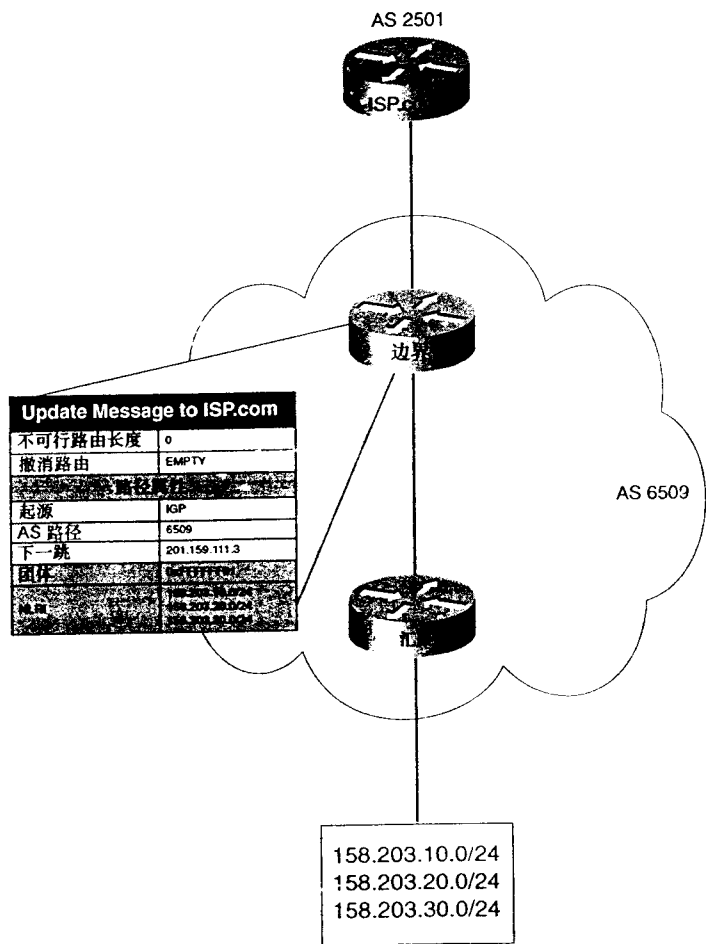


图 7-35 使用 NO_EXPORT 团体

7.8 路由反射器

在本章的前面介绍过，BGP-4 协议需要在同一自治系统内的所有 BGP 对等体互相之间都建立 I-BGP 会话，最早的 BGP 规范假设每个自治系统内都运行着内部网关协议来同步所有的 I-BGP 会话。但是自从这个规范制定后，越来越多的 BGP 用户不再使用 IGP 同步功能，同时让大型网络中的所有 I-BGP 路由器建立全网状结构也越来越困难。图 7-36 显示了在没有使用路由反射器和联盟的全网状情况下 6 台路由器之间需要建立多少 I-BGP 连接。

在范例中，6 个 I-BGP 发言人中的每一个都必须和本自治系统内的其他对等体建立 I-BGP 会话，你会发现这样的配置需要 $n * (n - 1) / 2$ ，也就是 15 个连接，这在使用昂贵的广域网连接的大型网络中会变得无法管理和无法接受。每个 I-BGP 会话都增加了 I-BGP 路由器需要支持的 BGP 总内存，同时也加重了处理器的使用负担，还给对 BGP 路由器提供支持的人员加大了管理难度。为了解决这个问题，创建了路由反射器（在 RFC 2796 中定

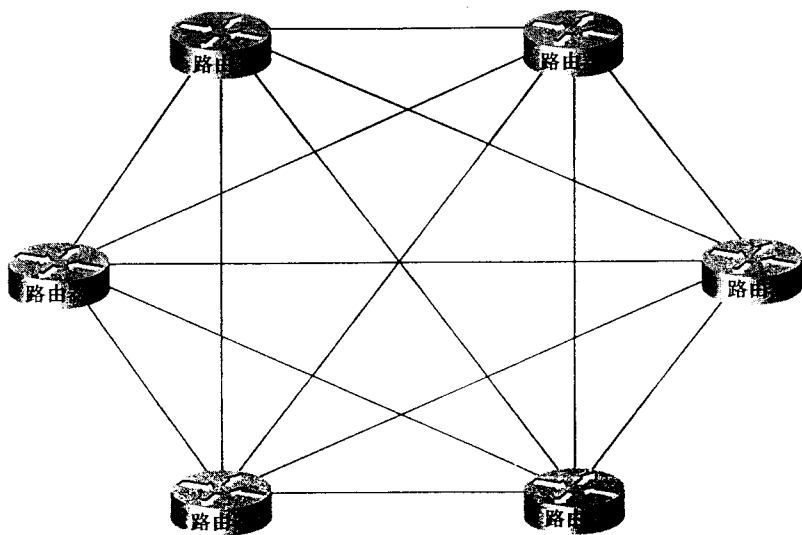


图 7-36 没有路由反射器的 I-BGP 全网状连接

义) 和联盟 (在 RFC 3065 中定义), 在本章的后续部分会介绍联盟。

路由反射器基本上是一个全功能的 I-BGP 发言人, 它和其他所有的 I-BGP 发言人建立 I-BGP 会话关系。然而, 路由反射器还有第二个作用: 它们转发从其他 I-BGP 发言人处学到的路由给路由反射器客户。路由反射器客户是只与路由发射器建立 I-BGP 会话的 BGP 路由器, 这就减少了 I-BGP 对等会话的数量, 同时简化了 BGP 路由的处理。图 7-37 显示了先前在图 7-36 中同样的网络, 在新的图中, 使用了路由反射器来减少 I-BGP 会话的数量。

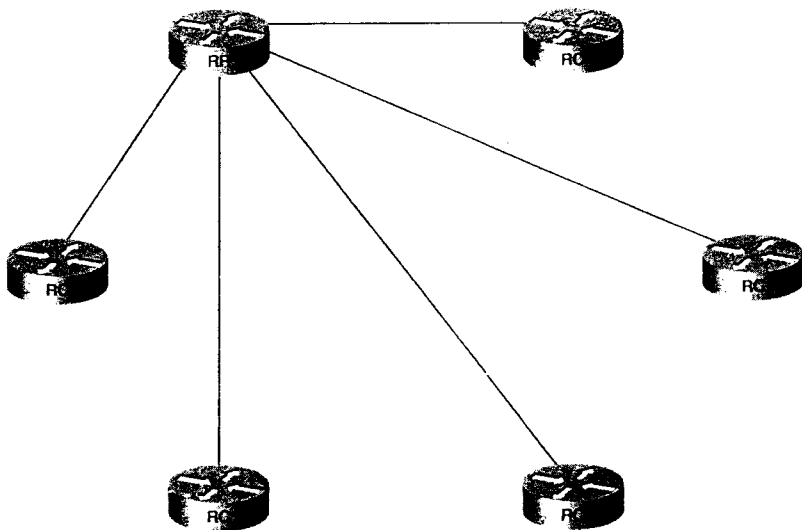


图 7-37 使用路由反射器来减少 I-BGP 会话的数量

在图中注意到 6 个 I-BGP 对等体中的 5 个被标记为 RC, 现在和标记为 RR 的路由反射器各建立一个 I-BGP 会话。

回顾一下，路由反射器通告 I-BGP 路由给 I-BGP 邻居，其中包括非路由反射器客户的全网状连接的邻居以及它们提供服务的路由反射器客户。尽管路由反射器将路由转发给客户端，但是除非特别配置，路由反射器客户不会转发路由给路由反射器。路由反射器和它的客户组成集群（CLUSTER），在一个自治系统中可以存在多个集群。任意一个不支持路由反射的 I-BGP 发言人都必须和除了路由反射器客户外其他所有的 I-BGP 路由器建立 I-BGP 会话，路由反射器客户当作自己通过路由反射器已经实现了全网状连接。路由反射器客户只需要与它们的路由反射器建立 I-BGP 会话，路由反射器将和其他非路由反射器客户的路由器建立 I-BGP 会话。

7.8.1 起源者识别符（ORIGINATOR_ID）

路由反射器集群通过 4 字节（32 位）的起源者识别符（ORIGINATOR_ID）来识别，它的值就是路由反射器的 BGP 识别符。起源者识别符通过路由反射器的 IP 地址来识别一个路由反射器集群从而防止路由环路。如果一个路由反射器在 UPDATE 报文中发现了自己的起源者识别符，它会假设发生了路由循环并且忽略报文。

起源者识别符是可选非过渡属性，RFC 2796 中描述其为用来防止路由环路的路由反射器集群的识别符。如果路由反射器收到了没有起源者识别符的路由，它将把自己的 BGP 识别符加到起源者识别符中，如果路由反射器在起源者识别符字段中发现了自己的 IP 地址，它将忽略这个更新。图 7-38 显示了自治系统内部的路由反射器是如何使用起源者识别符属性的。

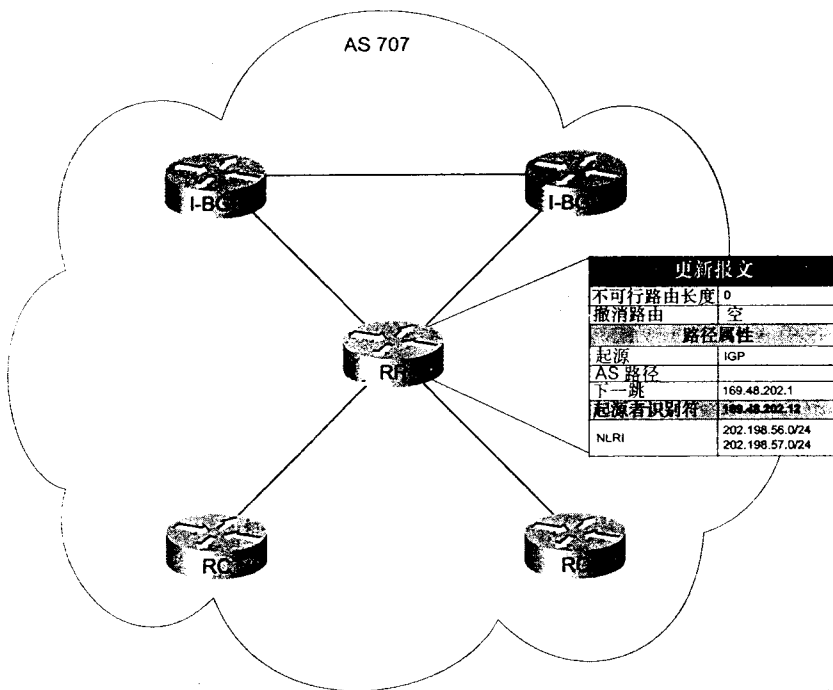


图 7-38 起源者识别符和路由反射器

7.8.2 集群列表 (CLUSTER-LIST)

同样在 RFC2796 中定义的集群列表属性是可选非过渡属性，当一个自治系统中存在多个路由反射器集群的时候用来防止环路的发生。集群列表的值是 4 个字节，与 AS 路径类似，包含了路由通过的反射路径的集群识别符列表。与起源者识别符类似，集群识别符也是路由器的 BGP 识别符，当路由反射器收到更新后，它会检查集群列表属性，如果集群列表字段是空的，就会将自己的集群识别符加入该字段，如果字段中已经有了别的记录，它将会把自己的集群识别符附加在列表的前面。如果路由反射器收到的更新在集群列表中有自己的集群识别符，那么就会假设发生了路由环路同时忽略这个更新。图 7-39 演示了在自治系统中如何将集群识别符附加到集群列表中来防止路由环路。

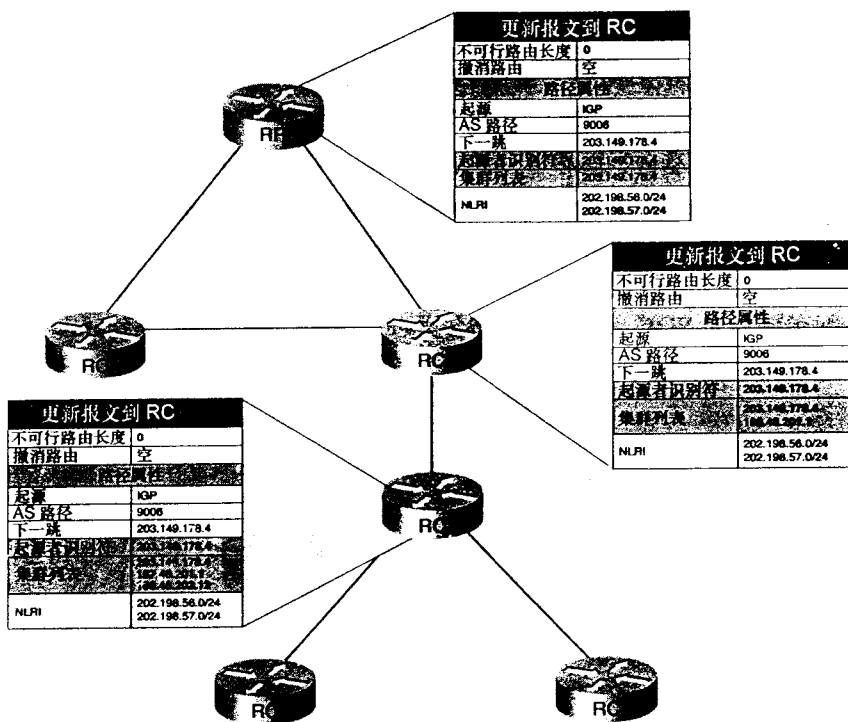


图 7-39 集群列表属性

7.9 联 盟

另外一个解决 I-BGP 全网状连接要求的办法是使用联盟，RFC 3065 定义了联盟是在一个主自治系统内创建的较小的子自治系统，通过这种方式来减少 I-BGP 对等体之间需要的 BGP 连接数量。图 7-40 列出了 6 台路由器在创建自治系统联盟前后的情况。

在上面的图中，自治系统 1765 中所有的 6 个对等体建立了全网状的 I-BGP 连接，也就是说 $n*(n-1)/2$ ， $6*(6-1)/2=15$ 个 I-BGP 对等体会话，同时路由器 A 和 D 与自治系统 2592

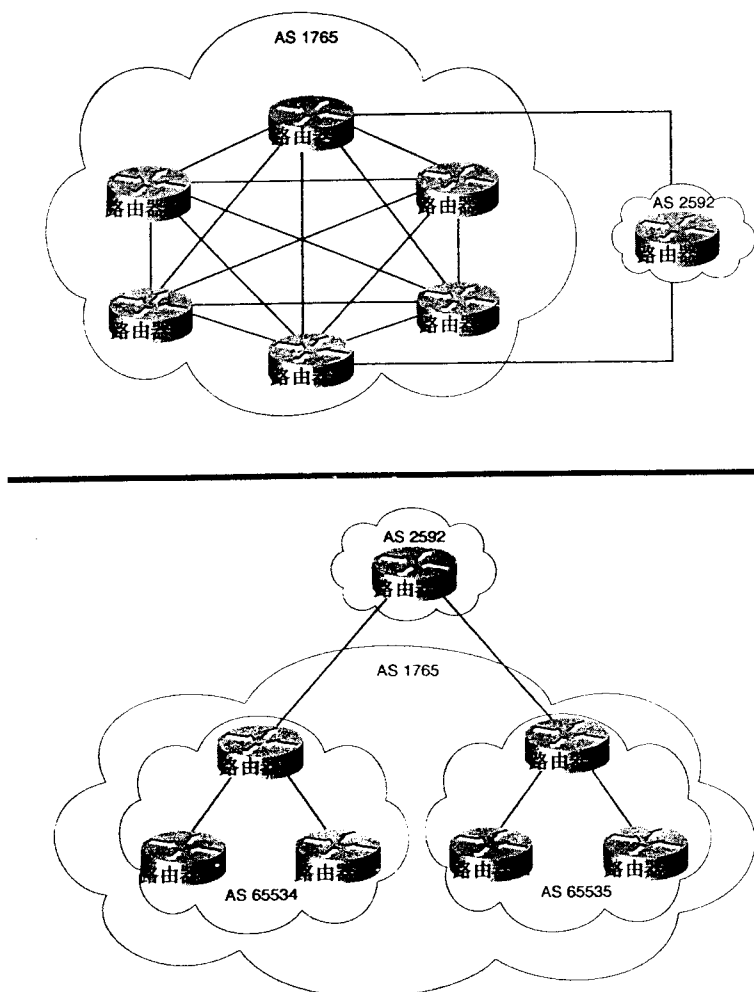


图 7-40 创建自治系统前后

中的路由器 Z 建立 E-BGP 对等体连接。在下面的图中，AS 1765 中建立了两个子自治系统：65 534 和 65 535，将每个子系统内的 I-BGP 会话减小到了 3 个，同时在子系统之间建立了 E-BGP 会话，路由器 A 和 D 仍然与自治系统 2592 中的路由器 Z 建立 E-BGP 会话，但是路由器 Z 完全不知道自治系统 1765 中存在自治系统联盟，不知道自治系统 1765 中有两个子自治系统 65 534 和 65 535。

所有的 BGP 联盟中的对等体与不属于联盟的 I-BGP 对等体遵循同样的规则，在子自治系统中每个对等体都必须和其他所有的 I-BGP 对等体建立 I-BGP 会话，同一个子自治系统中的对等体之间下一跳地址、AS 路径、多出口鉴别器和本地优先等属性保持不变。任何含有联盟的自治系统对外部 BGP 对等体表现为一个自治系统，每个子自治系统被分配自己的自治系统号码，这是私有号码，在子自治系统外不可见，被称为成员自治系统号码。属于自治系统联盟的子自治系统被称为成员自治系统。包含子自治系统的父自治系统仍然保持它自己的自治系统号码，当使用联盟的时候，这个号码被称为联盟识别符。由于在子自治系统中的对

等体与父自治系统中其他对等体有着不同的“我的自治系统”值，为了与父自治系统中的其他路由器建立通信，在联盟中最起码有一个成员需要和联盟外的成员建立 E-BGP 会话。当联盟中的成员给予自治系统外的对等体发送 BGP 更新的时候，发送方将使用它自己的子自治系统号码，当联盟中的对等体给 E-BGP 对等体发送更新的时候，它将使用父自治系统的联盟识别符来标识自己。

当使用联盟的时候，会应用两个新的 AS 路径属性之一：AS_CONFED_SET 或 AS_CONFED_SEQUENCE。它们描述了路由通过联盟时的路径，AS_CONFED_SET 是路由通过的子自治系统的无序列表，AS_CONFED_SEQUENCE 是路由通过的子自治系统的有序列表。当更新被发送给外部对等体的时候，AS_CONFED_SET 和 AS_CONFED_SEQUENCE 被替换为父自治系统的联盟识别符。图 7-41 演示了如何使用 AS_CONFED_SEQUENCE 来通告在离开父自治系统之前经过了多个子自治系统。

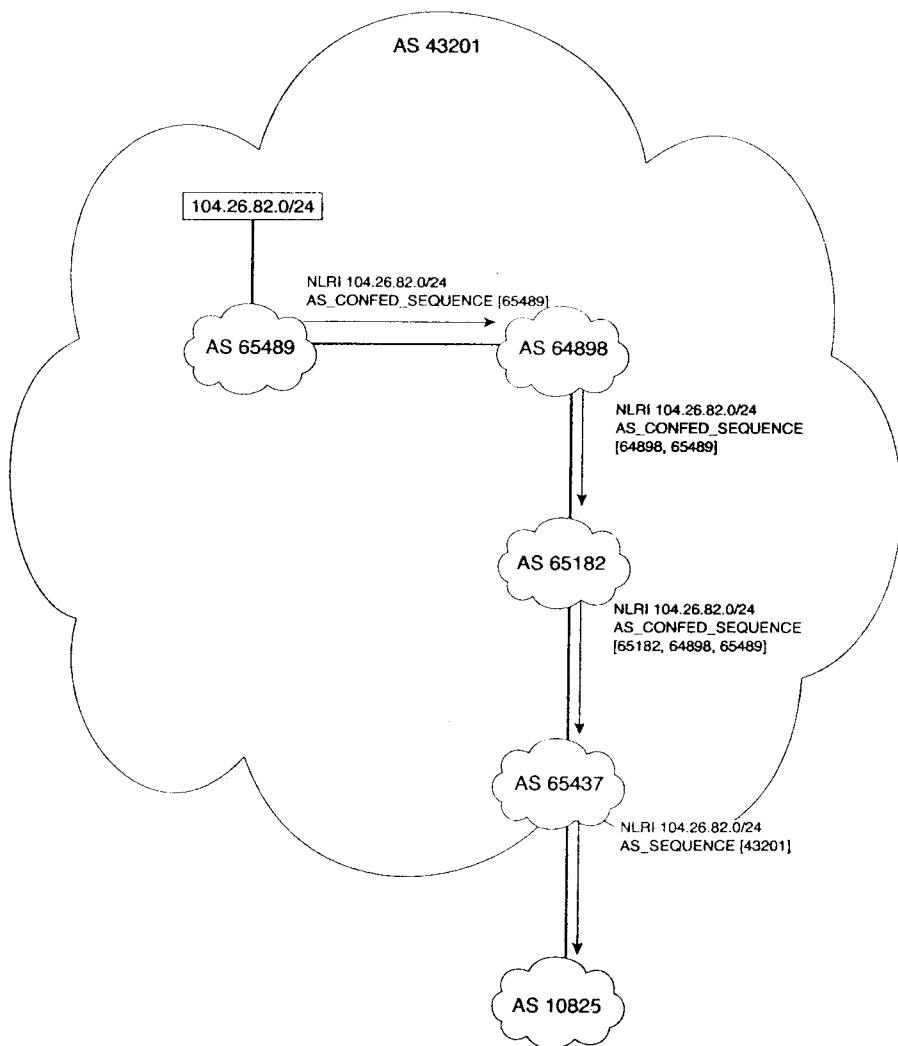


图 7-41 AS_CONFED_SEQUENCE 路径段落类型

图 7-42 描述了在联盟内部和外部的路由器充当的角色和不同的配置。

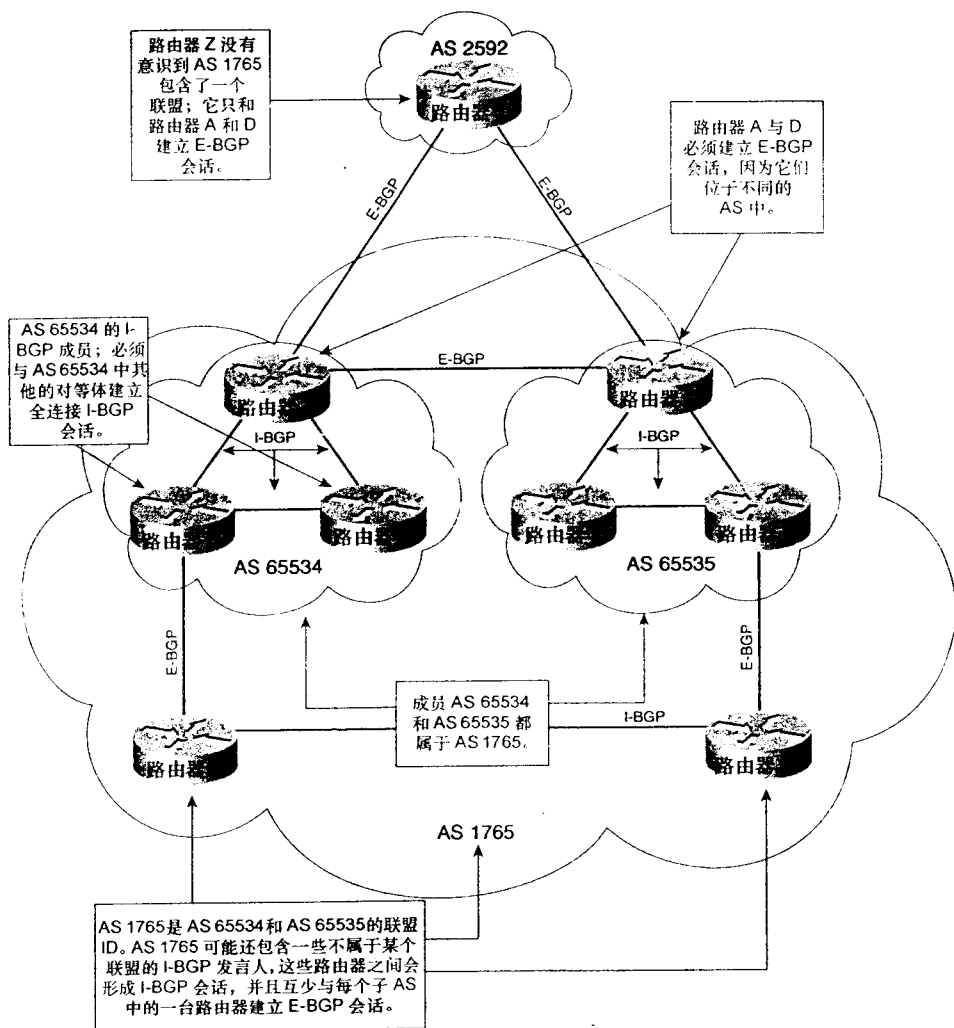


图 7-42 联盟如何工作

上图描述了一个包含联盟的自治系统的基本特征。子自治系统 65 534 包括了路由器 A、B 和 C，每个子自治系统中的路由器和其他路由器之间都建立 I-BGP 对等体关系，也就是全网状的关系，同样在子自治系统 65 535 中包括了路由器 D、E 和 F，它们之间也建立全网状的关系。路由器 A 和 D 之间建立了跨两个子自治系统的 E-BGP 会话，将两个子自治系统连接起来，同样路由器 B 和 F 也与路由器 Q 和 R 建立 E-BGP 会话，所有这些路由器都属于自治系统 65 534 和 65 535 的联盟 ID，也就是自治系统 1765。

路由器 A 和 D 同时也是自治系统 1765 中惟一和自治系统 2592 中路由器 Z 相连接的路由器，自治系统 2592 是自治系统 1765 惟一相邻的自治系统。在自治系统 1765 产生的路由被发送给自治系统 2592 之前，路由器 A 和 D 必须修改 AS 路径，使用值为 [1765] 的 AS_SEQUENCE 去替换值为 [65534] 或是 [65535] 的 AS_CONFED_SEQUENCE。

7.10 对等体组

当在一台路由器上配置多个 BGP 对等体关系时可能会很复杂，可以使用对等体组来简化配置和故障排查过程。对等体组通过建立组并将具有同样策略的邻居放入组中来创建，对等体组中的成员继承了组的策略。第 9 章将会介绍对等体组的配置和范例。

7.11 路由选择处理

既然你已经了解了 BGP 是如何运行的，属性是如何影响路由决策的，以及何时考虑更复杂的配置，现在该把这些信息放在一起考虑如何将路由放入到主路由表中。在一个 BGP 发言人将路由从 Adj-RIB-In 移到 Loc-RIB 表中之前，BGP 需要一系列很复杂的路由选择处理。除非已经显式地配置了使用多路径，BGP 只会将最优的那条路由放入主路由表中。只有 BGP 进程知道是可达的（通过内部网关协议或是直链路由）路由会被 BGP 路由选择进程处理。下面的 BGP 路由选择进程在思科的网站上 <http://www.cisco.com/warp/public/459/25.shtml> 有介绍。

- 第 1 步 选择具有最大的权重（范围从 0~65 535）的路径。记住，权重是思科专有的属性，它只对本地路由器起作用，不会被转发给其他任何对等体。
- 第 2 步 如果权重是一样的，那么选择有最大的本地优先值（范围从 0~4 294 967 295）的路径。
- 第 3 步 如果权重和本地优先值都是一样的，选择本地路由器始发的路由，这些路由可能是通过本地配置或是重分发产生的。
- 第 4 步 如果权重、本地优先以及路由的本地来源都是一致的，那么选择具有最短的 AS 路径的路由。
- 第 5 步 如果前面提到的所有属性都是一致的，选择最合适的起源，记住 IGP 是最优先的，EGP 比 Incomplete 优先。
- 第 6 步 如果前面提到的所有属性都是一致的，而且有多个网络出口路径，优选具有最小的多出口鉴别器值的路径（范围是 0~4 294 967 295）。
- 第 7 步 如果多出口鉴别器值相同或是没有使用，E-BGP 路径比 I-BGP 路径优先。
- 第 8 步 如果路径都是 E-BGP（或者路径都不是 E-BGP），优选具有最低的 IGP 量度的路径。如果使用了 BGP 多路径特性，同时从相邻的自治系统或是子自治系统学到了多条外部或是联盟外部路径，这时多条路径都被加入到 Loc-RIB 表中。当发送更新给其他路由器的时候最旧的路径被认为是最优的路径。
- 第 9 步 如果路径都是外部的，选择最旧的路径（即最早收到的路径）。
- 第 10 步 如果路径都是同时收到的，优选从具有较小的 BGP 识别符的对等体学到的路径。
- 第 11 步 如果路由来自于路由反射器，选择具有最小的集群识别符（路由反射器的 BGP 识别符）长度的路径。

第12步 如果路径来自于相同的主机，不论是对等体还是路由反射器，优选从具有最小的对等体 IP 地址（直连接口地址，如果没有直接连接就是最近的非直连接口地址）的邻居处学到的路径。

选定了最合适的路由后将其放入主路由表中并用来路由数据包。

7.12 总 结

BGP 是外部网关协议，使用了路径-向量逻辑来定义到目的网段的最佳路由。有两种类型的 BGP 关系：外部 BGP 和内部 BGP，它们都有不同的运行方式。BGP 对等体需要经过一系列的状态转变直到建立对等关系后才会选择路径，在建立对等体会话的过程中使用 OPEN 报文，当相邻的路由器成为对等体以后，它们交换保活报文来验证网络的连接并且通过 UPDATE 报文交换路由，当有严重错误时，发生问题的对等体会发送通知报文给邻居，说明错误的原因并且关闭 BGP 会话。在 UPDATE 过程中，BGP 使用一些属性类型来决定到目的网段的最佳路径，在选定了最佳路径后，将路径存放在主路由表中供将来使用。

7.13 进一步阅读资料

Internet Routing Architectures, Second Edition, by Sam Halabi.

Routing TCP/IP, Volume II, by Jeff Doyle and Jennifer Dehaven Carroll.

Cisco BGP-4 Command and Configuration Handbook, by Dr. William R. Parkhurst.

BGP4 Inter-Domair: Routing in the Internet, by John W. Stewart III.

RFC 1771, *A Border Gateway Protocol 4 (BGP-4)*, by Yakov Rekter and Tony Li.

RFC 1997, *BGP Communities Attribute*, by Ravi Chandra and Paul Triana.

RFC 1998, *An Application of the BGP COMMUNITY Attribute in Multi-Home Routing*, by Enke Chen and Tony Bates.

RFC 2395, *Protection of BGP Sessions via the TCP MD5 Signature Option*, by Andy Hefferman.

RFC 2519, *A Framework for Inter-Domain Route Aggregation*, by Enke Chen and John W. Stewart, III.

RFC 2892, *Capabilities Advertisement with BGP-4*, by Ravi Chandra and John G. Scudder.

RFC 2918, *Route Refresh Capability for BGP-4*, by Enke Chen.

RFC 2796, *BGP Route Reflection—An Alternative to Full Mesh IBGP*, by Tony Bates, Ravi Chandra, and Enke Chen.

第 8 章

BGP-4 配置介绍

在生产环境中配置边界网关协议（BGP）可能是网络专家在工作中感到最困难的任务之一。取决于你的 BGP 协议和配置的知识，BGP 对等连接的要求，网络的策略以及通常的网络可靠性，设计和实现一个稳定的 BGP 网络可能会使你面对很大的设计挑战。BGP 路由器的配置模式包含了数百条可能的命令，这使得 BGP 协议成为目前最可客户化的协议之一。BGP 同时使用了思科 IOS 软件中的其他一些特性来补充在 BGP 路由器配置模式下的命令，例如访问列表、路由映射、自治系统路径访问列表、团体列表和正则表达式等等，这些特性加上其他的 BGP 配置命令为 BGP 的配置建立了大的工具箱。在下面两章中，本书将覆盖这些命令并且显示如何使用这些命令来创建和实现可靠的 BGP 网络模式。

本章覆盖了基本的 BGP 配置先决条件，并且简单介绍了在思科路由器上运行的一些 BGP 进程，然后会通过一些实例详细展示一步一步如何配置 BGP 邻居和通告网段。在本章中配置 BGP 时，你可以有机会使用包括 BGP show 和 debug 命令在内的故障排查工具来分析和验证 BGP 配置。本章同时也会介绍一些 BGP 配置的提示以及其他一些可以用来减少故障排查时间和帮助你更加熟悉思科 IOS 软件的工具，这些工具可以用来仔细观察 BGP 的运行和解决常见的 BGP 问题。每个命令的输出都会详细列出，这样你就可以看到路由器在做什么。

本章是 BGP 的最后一章也就是第 9 章的基础，第 9 章介绍了包括路由反射器、联盟、重分发、路由会聚以及 BGP 调节等方面的内容。

8.1 BGP 配置的先决条件

的先决条件。必须考虑路由器上的可用内存和处理能力，还要考虑为了正确地建立网络模型所需要的软件特性。一个很好的经验是在配置 BGP 之前，必须确认路由器有能力运行 BGP。快速了解现有的操作环境，检查可用和已经使用的内存来验证调试 BGP 不会导致路由器死机。

如果路由器没有足够的内存，也没有方法增加内存的数量，你可以做一些事情来防止配置可能导致的失败。首先，使用 **show version** 命令检查你支持的特性集合。如果你使用的是企业版而且你不会使用如 IPX、AppleTalk 或是 DEC 等其他协议，那么你应该尝试去使用例如 IP 特性集合这样的简化思科 IOS 软件。其次，检查正在运行的进程和配置，看是否能够关闭一些协议或是特性来给 BGP 提供更多的可用内存。第三，关闭主控日志（记录到缓存或是 syslog），使用 **scheduler allocate** 命令防止路由器重启。最后，在调试之前保存你的配置，这样当路由器真的重启的时候你不会丢失你的配置。

8.1.1 评估路由器的 BGP 能力

思科的路由器配置 BGP 后会启用 4 个进程：BGP Open、BGP Scanner、BGP Router 和 BGP I/O。“BGP Open”进程用来在 BGP 发言人之间建立 TCP 会话，当会话成功建立后“BGP Open”进程将结束，所以只有在 BGP 会话建立的开始阶段才可以看到这个进程。“BGP I/O”进程进行所有 BGP 数据包的处理以及进行 BGP UPDATE 和保活报文的队列管理。“BGP Scanner”进程扫描或是遍历 BGP 表中一个叫“Radix Trie”的数据结构以修改下一跳地址的可达性。扫描器默认是每隔 60s 运行一次，当调试 BGP 的时候显示为 *as nettable_scan* 和 *nettable_walker*。最后，“BGP Router”进程进行实际的 BGP 决策处理，决定将哪个路由放入主 IP 路由表中，它也会处理新的路由以及将路由通告给对等体。范例 8-1 显示了使用 **show processes cpu | include BGP** 命令得到的 4 个 BGP 进程。

范例 8-1 4 个 BGP 进程

Alki# show processes cpu include BGP							
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%							
PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY Process
21	0	1	0	0.00%	0.00%	0.00%	0 BGP Open
84	81	6085	13	0.00%	0.02%	0.00%	0 BGP Router
85	693	13436	51	0.00%	0.00%	0.00%	0 BGP I/O
86	2547	201	12671	0.00%	0.06%	0.06%	0 BGP Scanner

在前例中当执行 **show processes cpu** 快照时有 4 个 BGP 进程在运行，当 BGP 被配置后“BGP Router”，“BGP I/O”和“BGP Scanner”会一直运行，而“BGP Open”进程只有在 BGP 触发了 TCP 会话的初始化时才开始运行，直到 TCP 会话建立后就结束，所以你可以断定上面的命令是在 BGP 刚刚配置后开始 BGP 会话的时候执行的。可以使用 **show processes history** 命令显示 CPU 利用率历史的图形化总结，这个命令可以用来在生产路由器上发现和解决性能问题。

提示：比如在范例 8-1 中显示的输出修饰语可以使你从命令输出中提取更简明的信息，在前面的范例中，输出修饰语 **| include BGP** 用来限制命令 **show processes cpu** 仅仅输出包含字符串“BGP”的行。输出修饰语是大小写敏感的，你可能需要试验输出字符串来找到

你需要的信息。当你将带有输出修饰语的命令和命令别名组合起来时，你就拥有了一个可以帮你客户化思科 IOS 软件使用的工具。我们将在本章的后面详细介绍别名和输出修饰语的使用。

在本范例中增加了命令输出的高亮部分来显示命令输出描述，当使用输出修饰语的时候除非指定它一般不会出现。范例 8-2 使用了 **show processes memory|include BGP** 命令来显示正在使用内存的 BGP 进程。

范例 8-2 命令 show processes memory | include bgp 的输出

```
Alki# show processes memory | include BGP
Total: 29184828, Used: 5148284, Free: 24036544
PID TTY Allocated Freed Holding Getbufs Retbufs Process
21 0 0 0 6928 0 0 BGP Open
84 0 52560 492 10324 0 0 BGP Router
85 0 0 0 6868 0 0 BGP I/O
86 0 116 0 9992 0 0 BGP Scanner
```

在前面的范例中，你可以看到路由器 Alki 分配给正在运行的 BGP 进程的内存大小。再一次提示，命令中的高亮部分显示了命令输出行的描述，如果在命令中加入了 **show processes memory** 的全部内容，那么输出将会有很多页，所以使用了输出修饰语来限制 **show** 命令的输出中仅仅包含 BGP 进程。范例 8-3 显示了命令 **show memory | include BGP** 使你输出当前分配给 BGP 进程的内存。命令输出的高亮部分显示了输出的描述。

范例 8-3 BGP 的内存使用

```
Alki# show memory | include BGP
Address Bytes Prev Next Ref PrevF NextF Alloc PC what
823A2F8C 0000000044 823A2D10 823A2FE4 001 ----- 813BC2E0 BGP Router
823C1C5C 0000005000 823C1830 823C3010 001 ----- 805A124C BGP rcache-
chunk
823C3010 0000005000 823C1C5C 823C43C4 001 ----- 805A1280 BGP fcache-
chunk
823C4406 0000060496 823C43C4 823D3084 001 ----- 805A12E8 BGP (0) attr
823D3084 0000000044 823C4408 823D30DC 001 ----- 813BC2E0 BGP Router
8241C8D4 0000000032 8241C7F8 8241C920 001 ----- 8045F35C BGP Router
8241D100 0000000072 8241D08C 8241D174 001 ----- 813B0548 BGP Router
8241D358 0000000072 8241D250 8241D3CC 001 ----- 813B0548 BGP Scanner
8241D704 0000032768 8241D6C0 82425730 001 ----- 805A12E8 BGP (1) attr
82425774 0000020000 82425730 8242A5C0 001 ----- 805A12E8 BGP (2) attr
8242A604 0000032768 8242A5C0 82432630 001 ----- 805A12E8 BGP (3) attr
82432630 0000003000 8242A604 82433214 001 ----- 805A1330 BGP attrlist
-chunk
82433214 0000001500 82432630 8243381C 001 ----- 805A1364 BGP worktype
-chunk
8243381C 0000005000 82433214 82434BD0 001 ----- 805A1398 BGP gwcach
-c
hunk
82434BD0 0000002000 8243381C 824353CC 001 ----- 805A13CC BGP NLRI-
chunk
824353CC 0000000432 82434BD0 824355A8 001 ----- 805A1400 BGP SNPA-
chunk
824355EC 0000065536 824355A8 82445618 001 ----- 805A146C BGP (0)
update
8244565C 0000065536 82445618 82455688 001 ----- 805A146C BGP (1)
```

(待续)

```
update
824556CC 0000065536 82455688 824656F8 001 ----- 805A146C BGP (2)
update
8246573C 0000065536 824656F8 82475768 001 ----- 805A146C BGP (3)
update
824757AC 0000065536 82475768 824857D8 001 ----- 805A146C BGP (4)
update
8248581C 0000065536 824857D8 82495848 001 ----- 805A146C BGP (5)
update
8249588C 0000065536 82495848 824A58B8 001 ----- 805A146C BGP (6)
update
824A58FC 0000065536 824A58B8 824B5928 001 ----- 805A146C BGP (7)
update
824B5928 0000065536 824A58FC 824C5954 001 ----- 805A14D4 BGP battr
chunk
824C5954 0000000264 824B5928 824C5A88 001 ----- 805A1508 BGP vpnv4
soo
```

前面命令的输出显示了 BGP 进程的内存地址，在这个范例中，路由器 Aiki 只有一个对等关系，在 BGP 表中只有 4 条路由。当路由器有很多对等体，有很多包含很多属性的路由的时候，命令 **show memory | include BGP** 将显示多页的信息。如果在网络模型中的路由器有多个对等体，那么最好给 BGP 提供较多的可用内存。在实验室环境中，BGP 可以在包括 BGP 的特性集合的任意路由器上运行；然而，BGP 的性能很大程度上取决于所选择的路由器平台、处理器、内存的数量和类型、背板的速度、从对等路由器收到的路由数以及路由器自己的配置。如果你为一个生产用的 BGP 网络建立模型，那么必须小心地选择生产用的路由平台来支持 BGP 处理和内存使用。如果你在配置一个运行全因特网路由表的 BGP 生产路由器，你最好先检查当前 BGP 因特网路由表的大小。必须确认你至少有全因特网路由表大小两倍的内存，以保证当全因特网路由表加倍后还能够不被中断地运行 BGP。

8.1.2 BGP 配置提示

当配置和检查 BGP 时，经常会使用一些命令。可以使用很多技巧来帮助自己更有效地使用思科 IOS 软件。比如，可以使用控制 (Ctrl) 键和键盘上其他字母的组合来作为编辑的快捷键。当你很着急或是心情不好不愿意多敲键的时候这些快捷键可以帮助你节省时间，另外当你使用的终端软件不支持上下键或是你经常使用的其他命令时这些命令也能帮上忙。表 8-1 列出了一些最流行的命令。

表 8-1 思科 IOS 软件快捷键

命令	描述	命令	描述
Ctrl+A	移到行的开始处	Ctrl+P	重复前一行
Ctrl+B	往回 (左) 移动一个字符	Ctrl+R	刷新行
Ctrl+E	移到行的结束处	Ctrl+U	删除整行
Ctrl+F	往前 (右) 移动一个字符	Ctrl+W	删除最后一个单词

另一个常常被忽略但是可以帮助你定制你使用思科 IOS 软件的命令是 **alias**，**alias** 命令使你能够创建命令别名来代表常用命令。在全局配置模式下使用下面的命令创建别名：

alias mode alias-name alias-string

在范例 8-4 中，你可以看到很多命令别名被用来作为各种各样的常用命令的快捷方式。

范例 8-4 命令别名

```
Alki# show alias
Exec mode aliases:
  h             help
  lo            logout
  p             ping
  r             resume
  s             show
  u             undebug
  un            undebug
  w             where
  cib           cle ip bgp *
  sb            show ip bgp sum
Router configuration mode aliases:
  net           network
```

注意有一些默认的别名：**h**，**lo**，**p**，**r**，**s**，**u**，**un** 和 **w**，我还加上了 3 个其他的别名，**cib** 代表命令 **clear ip bgp ***；**sb** 代表命令 **show ip bgp summary**；**net** 代表路由配置模式命令 **network**。别名不是必需的，但是它们对一个高效的网络工程师是有价值的工具。

在本章前面简单提到的另外一个工具是输出修饰语，输出修饰语已经出来很长时间了但是很少被使用，输出修饰语修改了 **show** 命令输出的显示方式，它对几乎每一个现有的 **show** 命令都有效。

表 8-2 输出修饰语

输出修饰语	描述
begin string	从指定的字符串开始显示命令输出
exclude string	显示除了字符串指定的信息以外的所有信息
include string	只显示符合字符串的项

范例 8-5 显示了如何使用每个不同的输出修饰语来从 **show ip bgp** 命令得到特定的输出。第一行是命令的未修改的输出，第二个高亮行显示了如何使用 **include** 语句来指定只显示用字符*>标记的最佳可达路由。注意，本范例中在*>字符串中的*字符前必须加上一个/字符变成/*>；如果没有加上/，路由器将会报错“% Failed to compile regular expression”，这是由于字符*本身是个通用表达式，代表和一连串的字符匹配，就像 DOS 中的“*.*”一样。第二个高亮的范例显示了如何使用 **exclude** 来排除在自治系统路径中有 600 的路由，最后一个高亮区显示了如何使用 **begin** 来防止在命令输出中显示输出描述信息。

范例 8-5 输出修饰语的范例

```
Alki# show ip bgp
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

(待续)

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.1.1.0/24	192.168.32.2	0		0 600	i
*> 10.2.2.0/24	192.168.32.2	0		0 600	i
*> 192.168.32.0/30	0.0.0.0	0		32768	I

Alki# show ip bgp | include /*
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.1.1.0/24	192.168.32.2	0		0 600	i
*> 10.2.2.0/24	192.168.32.2	0		0 600	i
*> 192.168.32.0/30	0.0.0.0	0		32768	I

Alki# show ip bgp | exclude 600
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.32.0/30	0.0.0.0	0		32768	I

Alki# show ip bgp | begin Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.1.1.0/24	192.168.32.2	0		0 600	i
*> 10.2.2.0/24	192.168.32.2	0		0 600	i
*> 192.168.32.0/30	0.0.0.0	0		32768	i

现在你已经知道了使 BGP 配置更容易的技巧，下面将把你对思科 IOS 软件的知识和技术结合起来学习如何在思科路由器上配置 BGP。

8.2 配置和故障排查 BGP 邻居关系

对每个 BGP 会话都必须完成 5 个主要任务，在本节中将检查每个任务，通过一个实例来展示每个任务中的所有项目，图 8-1 显示了本节中的范例使用的网络。

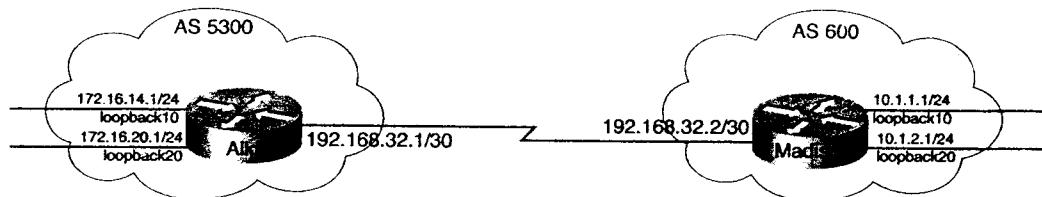


图 8-1 BGP 邻居配置

在进行 BGP 邻居会话配置之前，必须完成以下任务：

- 如果远端 BGP 对等体不在你的控制之下，那么必须找到远程对等体的远端接口地址和自治系统号码，在 E-BGP 中它们通常与网络的输出接口直接连接。
- 本地和远端的 BGP 对等体必须能够互相访问对方的 TCP 端口 179，因此，本地路由器的接口必须配置 IP 地址，而且路由器也必须有到达远端对等体的路径。
- 如果本地路由器没有和它的远端对等体直接连接，那么必须使用别的内部网关协议或是静态路由来提供建立 TCP 会话所必需的路由信息。

注意：在 BGP 会话建立之前 BGP 会话的两端都必须完全配置好。

在本例中，在路由器 Alki 和 Madison 之间的直连串行连接上配置 E-BGP 会话，路由器 Alki 的串口 0/0 上配置的 IP 地址是 192.168.32.1/30，路由器 Madison 的串口 0 的地址是 192.168.32.2/30，路由器 Alki 将通告网段 172.16.14.0/24 和 172.16.20.0/24，因此配置 loopback 10 为 172.16.14.1/24，loopback 20 为 172.16.20.1/24。路由器 Madison 将通告网段 10.1.1.0/24 和 10.1.2.0/24，于是配置 loopback 10 为 10.1.1.1/24，loopback 20 为 10.1.2.1/24。

第 1 步 验证本地 BGP 路由器能够访问远端路由器。**ping** 命令使你可以验证与远端路由器的连通性，但是如果在本地和远端路由器之间有访问列表或是防火墙，那么必须验证数据包过滤器将允许端口为 179 的 TCP 流量通过。

在这个时候，最好也使用命令 **show ip interface brief** 和 **show interface serial interface-number** 来验证两台路由器串口上的 IP 地址。在进行下一步之前，确认两个接口都在“**interface is up, line protocol is up**”状态。

验证 Alki 和 Madison 路由器之间能够互相访问。由于在两台路由器之间没有访问列表，我们可以放心地假设 **ping** 测试将会验证连接。范例 8-6 显示了串行接口的配置以及 **ping** 测试的结果。

范例 8-6 接口的配置和 ping 测试

```
Alki# show run | begin Serial0/0
interface Serial0/0
 ip address 192.168.32.1 255.255.255.252
Alki# ping 192.168.32.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.32.2, timeout is 2 seconds:
!!!!
Madison# show run | begin Serial0
interface Serial0
 ip address 192.168.32.2 255.255.255.252
Madison# ping 192.168.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.32.1, timeout is 2 seconds:
!!!!
```

当两个接口都是 up 状态而且第三层运行正常时就可以开始在每台路由器上配置 BGP。

第 2 步 在全局配置模式下使用 **router bgp as-number** 命令在思科 IOS 软件上启用 BGP，关键词 *as-number* 指的是本地自治系统号码。*as-number* 变量的取值范围是从 1~65 535，其中 64 512~65 535 保留给私有的自治系统使用。

```
router bgp as-number
```

这个命令启用了以下 BGP 进程（使用 **show processes cpu** 命令列出），同时给这些进程分配内存（使用 **show processes memory** 命令列出）：

- BGP Router;
- BGP I/O;
- BGP Scanner。

在 Alki 路由器上配置 BGP，Alki 路由器在自治系统 5300 中。

```
Alki(config)# router bgp 5300
Alki(config-router)#
```

可以使用 **show processes cpu | include BGP** 命令看到已经启用的 BGP 进程。

```
Alki(config-router)# do show processes cpu | include BGP
80      4      111      36 0.00% 0.00% 0.00% 0 BGP Router
84      0      1      0 0.00% 0.00% 0.00% 0 BGP I/O
85     44      4     11000 0.00% 0.06% 0.01% 0 BGP Scanner
```

注意：在前面的范例中，**do show processes cpu | include BGP** 命令用来显示当前的 BGP 进程。当你在 BGP 对等体之间 TCP 连接失败的路由器上执行该命令时，你将看到如下所示的 BGP Open 进程：

```
r2(config)# do show processes cpu | include BGP
78      0      179      0 0.00% 0.00% 0.00% 0 BGP Open
89      0      179      0 0.00% 0.00% 0.00% 0 BGP Open
99      0      179      0 0.00% 0.00% 0.00% 0 BGP Open
104    165252 3566960     46 0.00% 0.00% 0.00% 0 BGP Router
105      0      1      0 0.00% 0.00% 0.00% 0 BGP I/O
106     7108     890    7986 0.00% 0.03% 0.00% 0 BGP Scanner
107      0      179      0 0.00% 0.00% 0.00% 0 BGP Open
```

如果你执行 **show tcp brief all** 命令，你将发现路由器现在没有已建立的 TCP 会话，仍然在侦听进入的 TCP 会话请求，这是因为在 Alki 路由器上没有配置任何 BGP 对等体而且 Madison 路由器上也还没有配置 BGP。

```
Alki# show tcp brief all
TCB      Local Address      Foreign Address      (state)
8241BE64 *.*              *.*              LISTEN
```

在 Madison 路由器上配置 BGP，Madison 路由器在自治系统 600 中。

```
Madison(config)# router bgp 600
```

当命令 **router bgp as-number** 执行后，路由器进入了 BGP 路由配置模式，在这里可以输入范例 8-7 所示的 BGP 命令。这些命令将在本章和第 9 章中介绍。

范例 8-7 在思科 IOS 软件版本 12.2 (7) T 中可用的 BGP 命令

```
Madison(config-router)#?
Router configuration commands:
  address-family      Enter Address Family command mode
  aggregate-address    Configure BGP aggregate entries
  auto-summary         Enable automatic network number summarization
  bgp                  BGP specific commands
  default              Set a command to its defaults
  default-information  Control distribution of default information
  default-metric       Set metric of redistributed routes
  distance             Define an administrative distance
  distribute-list       Filter networks in routing updates
  exit                Exit from routing protocol configuration mode
  help                Description of the interactive help system
  maximum-paths        Forward packets over multiple paths
  neighbor             Specify a neighbor router
  network              Specify a network to announce via BGP
  no                  Negate a command or set its defaults
  redistribute         Redistribute information from another routing protocol
  synchronization      Perform IGP synchronization
  table-map            Map external entry attributes into routing table
  timers               Adjust routing timers
  traffic-share         How to compute traffic share over alternate paths
```

第 3 步 说明关于远端对等体的信息。使用命令 **neighbor ip-address remote-as remote-as-number** 输入远端对等体的信息，具体显示如下：

```
neighbor ip-address remote-as remote-as-number
```

这个命令指明了用来访问远端 BGP 对等体的 IP 地址以及远端对等体所属的自治系统号码。

使用 **neighbor** 命令配置 Alki 和 Madison 路由器的远端对等体信息，指明远端对等体的 IP 地址和远端的自治系统号。

```
Alki(config-router)# neighbor 192.168.32.2 remote-as 600
Madison(config-router)# neighbor 192.168.32.1 remote-as 5300
```

第 4 步 在配置了本地和远程的对等自治系统后，使用下面显示的 **network** 命令来配置每个 BGP 发言人将要通告给远端对等体的网段。

```
network network-address [mask subnet-mask] [route-map route-map-name]
[backdoor]
```

这个命令使你能够指定网段，如果网段不是有类的，你可以指定网段的子网掩码。*route-map* 选项允许 BGP 属性处理，关键字 **backdoor** 指明使用 BGP 后门，在本章的后面会详细介绍这些。

使用 **network** 命令配置 Alki 路由器以通告 172.16.14.0/24 和 172.16.20.0/24 网络。然后使用同样的命令配置 Madison 路由器来通告 10.1.1.0/24 和 10.1.2.0/24 网络。

```
Alki(config-router)# network 172.16.14.0 mask 255.255.255.0
Alki(config-router)# network 172.16.20.0 mask 255.255.255.0

Madison(config-router)# network 10.1.1.0 mask 255.255.255.0
Madison(config-router)# network 10.1.2.0 mask 255.255.255.0
```

第 5 步 在配置了本地和远端 BGP 对等体后，可以使用一些不同的 BGP **show** 和 **debug** 命令来监控 BGP 状态。

这时你应当在每台路由器上使用命令 **show tcp brief all** 验证一些项目，如范例 8-8 所示，你应该可以看到 Alki 和 Madison 路由器之间在端口 179 上已经建立的 TCP 会话，还应该看到路由器在端口 179 上侦听 TCP 的活动。

范例 8-8 使用命令 **show tcp brief all** 显示 TCP 连接状态

Alki# show tcp brief all			
TCB	Local Address	Foreign Address	(state)
8248F4BC	192.168.32.1.11003	192.168.32.2.179	ESTAB
820E59F0	*.179	192.168.32.2.*	LISTEN

通过使用命令 **show ip bgp**，如范例 8-9 所示，你应该可以看到关于 BGP 会话的信息以及两个对等体通告的网段。

范例 8-9 使用 **show ip bgp** 命令显示 BGP 路由

Alki# show ip bgp
BGP table version is 5, local router ID is 1.1.1.1

(待续)

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
  Network          Next Hop          Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.32.2          0           0 600 i
*> 10.1.2.0/24      192.168.32.2          0           0 600 i
*> 172.16.14.0/24   0.0.0.0              0          32768 i
*> 172.16.20.0/24   0.0.0.0              0          32768 i
```

在 Alki 路由器上，要注意的是你可以看到网段 10.1.1.0/24 和 10.1.2.0/24；它们的下一跳地址是 192.168.32.2，多出口鉴别器、本地优先和权重等属性的值都是默认的。同时你可以看到路由来源于自治系统 600，起源属性的值是 i 即 IGP，这是因为路由来源于 Madison 路由器本地。在每条路由的左边你都可以看到星号 (*)，表明 BGP Scanner 进程已经验证了路由是可达的，大于号 (>) 说明这个路由是到目的网段的最佳路由。当 BGP 有到某个网段的最佳路由的时候，它会把这条路由放入主 IP 路由表中，并且通告给它的所有其他外部 BGP 对等体。

你应该也可以使用 **show ip route** 命令来查看在主 IP 路由表中的 BGP 路由，并且 **ping** 每一个环回接口，范例 8-10 显示了路由器 Alki 上命令 **show ip route** 的输出，范例 8-11 显示了路由器 Alki 和 Madison 上 **ping** 的测试结果。

范例 8-10 使用 show ip route 命令查看主 IP 路由表

```
Alki# 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, 2 subnets
    C      172.16.20.0 is directly connected, Loopback20
    C      172.16.14.0 is directly connected, Loopback10
 10.0.0.0/24 is subnetted, 2 subnets
    B      10.1.2.0 [20/0] via 192.168.32.2, 00:05:30
    B      10.1.1.0 [20/0] via 192.168.32.2, 00:05:30
 192.168.32.0/30 is subnetted, 1 subnets
    C      192.168.32.0 is directly connected, Serial0/0
```

范例 8-11 在 Alki 和 Madison 路由器上成功的 ping 测试

```
Alki# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
Alki# ping 10.1.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/36 ms
```

使用 **debug ip bgp** 命令可以看到路由器建立 BGP 会话、通告网段、将网段放入 BGP 表中的过程。由于 BGP 在 UPDATE 报文中只会发送新增或是变化的路由，需要使用 **clear ip bgp**

命令来清除 BGP 会话。由于在这台路由器上只有一个 BGP 会话，如范例 8-12 所示，可以使用 * 字符来清除所有的 BGP 会话。

提示：在生产性网络中的路由器上必须小心使用 **clear ip bgp *** 命令，这条命令关闭所有的 BGP 会话，如果在生产网络中使用将会导致网络中断。

范例 8-12 调试 BGP

```
Alki# debug ip bgp
BGP debugging is on
Alki# clear ip bgp *
01:10:18: BGP: 192.168.32.2 went from Established to Idle
Comment: BGP cleared session
01:10:18: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Down User reset
Comment: the ADJCHANGE message indicates the session with the 192.168.32.2
neighbor is down due to a user reset
01:10:18: BGP: 192.168.32.2 closing
Comment: The BGP session is being closed
01:10:38: BGP: 192.168.32.2 went from Idle to Active
01:10:38: BGP: 192.168.32.2 open active, delay 26900ms
Comment: The router sent a active host TCP open message
connection request and is awaiting a TCP session request from its passive
neighbor.
01:10:48: BGP: Applying map to find origin for 172.16.14.0/24
01:10:48: BGP: Applying map to find origin for 172.16.20.0/24
Comment: BGP is finding the ORIGIN for the 172.16.14.0/24 and 172.16.20.0/24
routes, which will be i for I-BGP
01:11:05: BGP: 192.168.32.2 open active, local address 192.168.32.1
01:11:05: BGP: 192.168.32.2 went from Active to OpenSent
Comment: The remote BGP session transitioned from Active to OpenSent meaning a
TCP session has been established and OPEN message has been sent, the router is
now waiting to receive an OPEN message from its peer.
01:11:05: BGP: 192.168.32.2 sending OPEN, version 4, my as: 5300
Comment: The router sent an OPEN message to its peer, 192.168.32.2, and the
message contained the BGP version: 4 and the MY_AS value 5300
01:11:05: BGP: 192.168.32.2 send message type 1, length (incl. header) 45
01:11:05: BGP: 192.168.32.2 rcv message type 1, length (excl. header) 26
Comment: The remote router sent an OPEN (type-1) message to this peer and it was
successfully received
01:11:05: BGP: 192.168.32.2 rcv OPEN, version 4
01:11:05: BGP: 192.168.32.2 rcv OPEN w/ OPTION parameter len: 16
01:11:05: BGP: 192.168.32.2 rcvd OPEN w/ optional parameter type 2 (Capability)
len 6
01:11:05: BGP: 192.168.32.2 OPEN has CAPABILITY code: 1, length 4
01:11:05: BGP: 192.168.32.2 OPEN has MP_EXT CAP for afi/safi: 1/1
01:11:05: BGP: 192.168.32.2 rcvd OPEN w/ optional parameter type 2 (Capability)
len 2
01:11:05: BGP: 192.168.32.2 OPEN has CAPABILITY code: 128, length 0
01:11:05: BGP: 192.168.32.2 OPEN has ROUTE-REFRESH capability(old) for all
address-families
01:11:05: BGP: 192.168.32.2 rcvd OPEN w/ optional parameter type 2 (Capability)
len 2
01:11:05: BGP: 192.168.32.2 OPEN has CAPABILITY code: 2, length 0
01:11:05: BGP: 192.168.32.2 OPEN has ROUTE-REFRESH capability(new) for all
address-families
Comment: The remote peer's OPEN message contained the following data:
Comment: BGP version - 4
Comment: With Multiprotocol BGP and Route Refresh capabilities
01:11:05: BGP: 192.168.32.2 went from OpenSent to OpenConfirm
Comment: The session transitioned from OpenSent to OpenConfirm, the router is
```

(待续)

```

waiting on a KEEPALIVE message from its peer.
01:11:05: BGP: 192.168.32.2 send message type 4, length (incl. header) 19
01:11:05: BGP: 192.168.32.2 rcv message type 4, length (excl. header) 0
Comment: the router sent and received a KEEPALIVE (type-4) message and received a
message from its peer.
01:11:05: BGP: 192.168.32.2 went from OpenConfirm to Established
Comment: The session transitioned from OpenConfirm to Established, now routes can
be exchanged using UPDATE messages
01:11:05: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Up
Comment: The ADJCHANGED message indicating the BGP session with peer 192.168.32.2
is up

```

命令 `debug ip bgp event` 用来显示在路由器上发生的内部 BGP 事件的详细信息，范例 8-13 中的命令 `debug ip bgp updates` 用来显示路由器收到的 UPDATE 报文的详细信息。

范例 8-13 调试 BGP 更新

```

Alki# debug ip bgp updates
BGP updates debugging is on
Alki# clear ip bgp *
01:33:30: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Down User reset
Comment: The session was reset upon user request
01:34:12: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Up
Comment: The BGP session with peer 192.168.32.2 is back up
01:34:12: BGP(0): 192.168.32.2 rcvd UPDATE w/ attr: nexthop 192.168.32.2, origin
i, metric 0, path 600
Comment: The router received an update from peer 192.168.32.2 containing the
BGPAttributes:
Comment: NEXT_HOP 192.168.32.2
Comment: ORIGIN: i
Comment: MED: 0
Comment: AS_PATH 600
01:34:12: BGP(0): 192.168.32.2 rcvd 10.1.1.0/24
01:34:12: BGP(0): 192.168.32.2 rcvd 10.1.2.0/24
Comment: The update contained NLRI paths 10.1.1.0/24 and 10.1.2.0/24
01:34:12: BGP(0): Revise route installing 10.1.1.0/24 -> 192.168.32.2 to main IP
table
01:34:12: BGP(0): Revise route installing 10.1.2.0/24 -> 192.168.32.2 to main IP
table
Comment: BGP found the routes to networks 10.1.1.0/24 and 10.1.2.0/24 valid best
paths and is installing them in the main IP routing table
01:34:12: BGP(0): nettable_walker 172.16.14.0/24 route sourced locally
01:34:12: BGP(0): nettable_walker 172.16.20.0/24 route sourced locally
Comment: The BGP Scanner (nettable_walker) found networks 172.16.14.0/24 and
172.16.20.0/24 sourced locally
01:34:12: BGP(0): 192.168.32.2 computing updates, afi 0, neighbor version 0,
table version 5, starting at 0.0.0.0
01:34:12: BGP(0): 192.168.32.2 send UPDATE (format) 172.16.14.0/24, next
192.168.32.1, metric 0, path
Comment: The router is sending an UPDATE message to 192.168.32.2 containing the
route 172.16.14.0/24 with the attributes of NEXT_HOP: 192.168.32.2, MED: 0
01:34:12: BGP(0): 192.168.32.2 send UPDATE (prepend, chgflags: 0x208)
172.16.20.0/24, next 192.168.32.1, metric 0, path
Comment: The router is sending an UPDATE message to 192.168.32.2 containing the
route 172.16.20.0/24 with the attributes of NEXT_HOP: 192.168.32.2, MED: 0
01:34:12: BGP(0): 192.168.32.2 1 updates enqueued (average=56, maximum=56)
01:34:12: BGP(0): 192.168.32.2 update run completed, afi 0, ran for 4ms, neighbor
version 0, start version 5, throttled to 5
Comment: UPDATE messages were enqueued for transport and then sent successfully
the BGP table version has been changed to 5
01:34:12: BGP: 192.168.32.2 initial update completed
Comment: The update is complete

```

如果 BGP 对等体之间无法互相访问对方的 TCP 端口 179，可以使用一些 TCP 的故障排查命令来排除连接上的问题。一个很好的实践经验（将会大大减少你的头疼）是在故障排查一个错误之前验证路由器配置的准确性，可能最后你会发现问题是由于输入错误而导致的。

- 验证本地 BGP 自治系统号码配置正确。
- 验证远端对等体的 BGP 自治系统号码和 IP 地址配置正确。
- 验证连接到两端对等体的接口是 up 状态和可操作的。
- 如果对等体没有直接连接，验证它们都有有效的路由（往返两个方向）可以互相访问。
- 检查对等体之间的路由器是否有能够丢弃或重路由 BGP 流量的访问列表或是路由策略。
- 检查接口不稳定性记录。在 BGP 对等体之间的路由有抖动吗？BGP 对等体之间的连接通过的接口有严重的拥塞或是丢包吗？记住 BGP 的 OPEN 和保活报文使用很小的数据包，当其他较大的数据包独占了一个拥塞的接口时会导致这些 BGP 报文延时。
- 当两个 BGP 对等体之间的路径上有变化的时候，验证这些变化不会影响 BGP 会话，这些变化包括一个新的交换机配置、新的访问列表、防火墙、新的路由策略等等。

当不是 BGP 的问题的时候不用浪费时间去故障排查 BGP，建立一个通用的层次化的故障排查方式，当你碰到问题的时候它将是你的首选工具和最好的朋友。

第 1 步 第一层

- 检查你的接线，验证所有的线缆都正确地连接，所有接口的线路和协议都是处于工作状态。当你有第一层问题的时候不用浪费时间去故障排查 BGP。
- 如果你使用的是串行接口，首先确认你设置了正确的时钟速率。如果你使用的是 channel service unit/data service unit (CSU/DSU)，确认已经正确地配置了 CSU/DSU 而且线路已经工作。
- 如果你使用的是以太网接口，确认在路由器和交换机上设置了正确的速率和双工参数。
- 检查路由器和交换机的接口是否有错误；如果有错误，必须在继续故障排查之前解决这些错误。
如果你使用的是令牌环接口，必须确认路由器的环速率配置正确，路由器到 multistation access unit (MSAU) 或是交换机的连接正常。

第 2 步 第二层

- 如果你使用的是以太网连接，确认交换端口属于正确的虚拟局域网。
- 确认虚拟局域网配置正确，在交换机中没有生成树拓扑问题。
- 如果是 ATM 接口，验证在连接两端正确地配置了最大传输单元 (MTU)。
- 验证你使用了正确的虚路径识别符/虚通道识别符 (VPI/VCI) 对，已经配置了有效的第二层到第三层连接的 ATM 映射。
- 如果是帧中继连接，验证你的本地和远端的数据链路连接识别符 (DLCI) 和本地管理接口 (LMI) 类型正确设置为符合交换机配置的值。

- 验证 LMI 已经工作，接口也没有抖动。
- 如果你使用的是 PPP 连接，确认 PPP 在连接的两端已经配置。
- 在进行下一步之前，确认你的接口不是处于线路工作而协议没有工作的状态。

第 3 步 第三层

- 验证你在接口上配置了正确的 IP 地址和子网掩码，检查连接的另外一端，验证它在同一子网上（如果是直连的）或是如你期望的那样。
- 确保在 IP 路由表中有到你的目的地址的有效路由。沿着路径跟踪连接通过的所有路由器，验证它们都有供数据包访问你的源和目的网段所必需的往来路由。
- 检查静态路由是否有输入错误，确认重分发的路由已经被正确地传播。
- 如果使用了多路径，验证它们没有路由环路。
- 如果任何路由由协议使用了认证功能，确认它们都使用了正确的密码。
- 在例如 ATM 或是帧中继的非广播多路访问（NBMA）网络上，确认你有正确的二层到三层的映射，对类似于开放最短路径优先（OSPF）的协议配置了正确的网络类型。
- 在进入下一步之前，验证你能够从源网段访问目的网段，反之亦然。

第 4 步 第四层

- 检查是否有可能导致 TCP 数据包丢失的访问列表或是防火墙。
- 验证你在 TCP 端口 179 上有连接。一个 BGP 发言人是被动的 TCP 主机，将在端口 179 上收到 TCP 请求，另外一个 BGP 发言人（主动的 TCP 主机）将使用一个随机产生的 TCP 源端口（从 11 000 开始）来初始化 TCP 会话。
- 检查重传、数据包乱序或是其他的 TCP 症状，这些现象可能表明网络拥塞或是无效的配置。

在验证前面的条件都没有影响 BGP 会话后，继续使用 TCP **show** 和 **debug** 命令来缩小可能错误的范围。这些用来故障排查 BGP TCP 连接的工具在表 8-3 中列出。

表 8-3

TCP 连接故障排查工具

TCP 命令	命令描述
show tcp	显示了本地路由器和远端对等体建立的每个 TCP 会话，可以与 BGP 一起使用来显示本地和远端的 BGP 对等体是否有已连接的 TCP 会话以及会话的详细信息
show tcp[brief][all][include 179]	显示了本地路由器和远端对等体建立的每个 TCP 会话的简单状态，这是一个基本的总结命令，可以作为你用来验证 BGP 对等体的 TCP 连接的一个工具
debug ip tcp transatcions	这个命令在生产路由器中应该小心使用，它可以显示 TCP 会话变化的信息，使你可以检查 BGP 的 TCP 会话的错误，显示 TCP 重传和状态变化的信息
debug ip tcp packet[in out address IP-address[port port-number]	这个命令显示 TCP 数据包的详细信息，可以和 in、out、address 或是 port 参数一起使用来指定特别的流量，在生产路由器上必须特别小心地使用。通过这个命令，你可以监控本地路由器发送和接收的 TCP 数据包，这些信息使你可以判定不稳定的 BGP TCP 会话的原因，解决路由抖动或是通常的连接问题

如果 **show tcp** 命令的输出中 BGP 会话使用的对等体的 IP 地址状态不是 **ESTAB**，那么就故障排查 TCP 连接。范例 8-14 中的 **show tcp** 命令显示了 TCP 连接的详细信息，作为一个很好的实践经验，总是应该用作 TCP 会话的故障排查命令。

范例 8-14 show tcp 命令

```
Alki# show tcp
Stand-alone TCP connection to host 192.168.32.2
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 192.168.32.1, Local port: 11009
Foreign host: 192.168.32.2, Foreign port: 179
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x16681CC):
Timer           Starts      Wakeups      Next
Retrans         323          1            0x0
TimeWait        0            0            0x0
AckHold         320          164          0x0
SendWnd         0            0            0x0
KeepAlive       0            0            0x0
GiveUp          0            0            0x0
PmtuAger        0            0            0x0
DeadWait        0            0            0x0
iss: 3779523619 snduna: 3779529779 sndnxt: 3779529779 sndwnd: 16080
irs: 2902813429 rcvnxt: 2902819573 rcvwnd: 16099 delrcvwnd: 285
SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, nagle
Datagrams (max data segment is 1460 bytes):
Rcvd: 556 (out of order: 0), with data: 320, total data bytes: 6143
Sent: 492 (retransmit: 1, fastretransmit: 0), with data: 321, total data bytes:
6159
```

表 8-4 显示了 **show tcp** 命令输出的详细信息。你可能永远不会在每天的故障排查中使用所有的 20 行，但是当你排除类似过多重传等 TCP 连接问题的时候它们可以提供方便。

表 8-4 show tcp 命令的输出解释

命令输出	输出解释
Stand-alone TCP connection to host 192.168.32.2	识别从本地路由器到 192.168.32.2 的 TCP 连接
Connection state is ESTAB	<p>表明是一个已经建立的 TCP 会话，</p> <p>Connection state is 可以是以下任意值之一：</p> <p>LISTEN—表明路由器在侦听连接请求</p> <p>SYNSENT—表明路由器在等待发送过的请求（TCP-SYN 报文）的返回请求</p> <p>SYNRCVD—表明路由器已经发送和接收到了连接请求，现在正在等待连接确认（TCP-ACK 报文）</p> <p>ESTAB—表明一个已经建立的 TCP 会话（TCP-SYN 和 ACK 报文）</p> <p>FINWAIT1—表明路由器或者在等待一个结束请求或者确认前面发送的结束请求（TCP-FIN ACK 报文）</p> <p>FINWAIT2—说明路由器在等待来自远端对等体的结束请求（TCP-FIN 报文）</p> <p>CLOSEWAIT—说明路由器在等待用户端的结束请求（TCP-FIN 报文）</p> <p>CLOSING—说明路由器在等待远端主机的结束请求（TCP-FIN 报文）</p> <p>LASTACK—说明路由器在等待向远端对等体发送的结束请求的响应（TCP-FIN ACK 报文）</p> <p>TIMEWAIT—说明路由器在关闭连接前给远端主机一定的时间去接收连接结束请求</p> <p>CLOSED—表明没有连接</p> <p>一个成功的 BGP 会话的 TCP 会话必须始终在 ESTAB 状态</p>

续表

命令输出	输出解释
I/O status: 1	说明连接的状态
unread input bytes: 0	指明已经收到在等待处理的字节数
Local host : 192.168.32.1, Local port: 11009	显示了本地的 IP 地址和 TCP 端口号。你可以使用这个数字来判断是本地还是远端的对等体初始化了 BGP 连接。如果 TCP 端口号在 11 000 范围内，那就是本地路由器向远端的端口 179 发起了会话
Foreign host: 192.168.32.2, Foreign port: 179	显示连接的远端 IP 地址和 TCP 端口号，对 BGP 来说你看到的值总是 179 或是在 11 000 范围内的端口
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)	说明了等待重传的数据包个数，任意大于 0 的值表明有数据包重传和可能的 TCP 问题
Event Timers (current time is 0x16681cc): Timer Starts Wakeups Next Retrans 323 1 0x0 TimeWait 0 0 0x0 AckHold 320 164 0x0 SendWnd 0 0 0x0 KeepAlive 0 0 0x0 GiveUp 0 0 0x0 PmtuAger 0 0 0x0 DeadWait 0 0 0x0	本部分以计数值的方式显示了当前 TCP 会话的计时器信息(这个信息可以使用 clear tcp statistics 命令清除) <i>Event Timer</i> 显示了系统目前已经运行的时间，它的格式是毫秒 <i>Timer</i> 列描述了下面行列出的计时器 <i>Starts</i> 列描述了这个会话的计数器启用的次数 <i>Wakeups</i> 列描述了没有被响应的 KEEPALIVE 个数 <i>Next</i> 列描述了下次关闭计时器的时间 <i>Retrans</i> 计时器显示了用来计算没有被确认的数据包等待重传的时间的计时器值 <i>TimeWait</i> 计时器说明系统为了让远端系统接收连接结束请求将要等待的时间 <i>AckHold</i> 计时器的作用是延迟确认消息的传输，以避免网络拥塞 <i>SendWnd</i> 计时器可以防止远端系统由于丢失应答而导致的 TCP 会话失败 <i>KeepAlive</i> 计时器用来给保活报文之间的间隔计时 <i>GiveUp</i> 计时器是在放弃等待处理的决议请求前最少等待的时间 <i>PmtuAger</i> 计时器是用来跟踪路径 MTU 老化计时器的计时器，MTU 老化计时器可以通过 ip tcp path-mtu-discovery [age-timer {minutes indefinite}] 命令修改 <i>DeadWait</i> 计时器是 TCP 的 DeadWait 计时器
iss: 3779523619	显示了初始发送序列号，是在一个新的 TCP 会话中开始发送的序列号
snduna: 3779529779	显示了路由器已发送的最后的未确认序列号
sndnxt: 3779529779	显示了发送的下一个序列号
sndwnd: 16080	显示了远程主机的 TCP 窗口大小
irs: 2902813429	显示了初始的接收序列号
rcvnxt: 2902819573	显示已经接收并被确认的最后的序列号
rcvwnd: 16099	显示了本地路由器的 TCP 窗口大小
delrcvwnd: 285	显示了延迟的接收窗口，是接收窗口未计算的值
SRTT: 300 ms	平滑往返计时器是衡量数据包被发送和远端对等体响应的平均时间
RTT0: 303 ms	毫秒为单位的往返超时
RTV: 3 ms	毫秒为单位的往返时间的变化值
KRTT: 0ms	新的往返 (K 代表 Karn 的运算法则) 超时，它以毫秒为单位计算被重传的数据包的往返时间
minRTT: 20ms	最小的往返超时
maxRTT: 300ms	最大的往返超时
ACK hold: 200ms	应答延迟超时，用来延迟应答以获得将数据加入包中的时间

续表

命令输出	输出解释
Flags: higher precedence	指定在数据包中可能出现的 IP 优先级值
nagle	说明设置了 Nagle 标志
Datagrams (max data segment is 1460 bytes) :	以字节表示最大数据段
Rcvd: 556 (out of order: 0, total data bytes: 6143	收到的数据报的数目 收到的乱序的数据报的数目 收到数据的总字节数
Sent: 492 (retransmit: 1, fastretransmit: 0) , with data: 321, total data bytes: 6159	发送的数据报的数目 重传的数据报的数目 快速重传的数据报的数目 发送的包含数据的数据报的数目 发送的数据的总字节数

其他两个可以用来故障排查 TCP 连接但是常常会被忘记的工具是命令 `debug tcp transactions` 和 `debug tcp packet`，命令 `debug tcp transactions` 在范例 8-15 中列出。

范例 8-15 debug ip tcp transactions 命令

```
Alki# debug ip tcp transactions
TCP special event debugging is on
Alki# clear ip bgp *
01:53:24: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Down User reset
Comment: BGP session reset at user request
01:53:24: TCP0: state was ESTAB -> FINWAIT1 [179 -> 192.168.32.2(11005)]
Comment: TCP session transitioned from ESTAB to FINWAIT1
01:53:24: TCP0: sending FIN
01:53:24: TCP0: state was FINWAIT1 -> FINWAIT2 [179 -> 192.168.32.2(11005)]
01:53:26: TCP0: FIN processed
01:53:26: TCP0: state was FINWAIT2 -> TIMEWAIT [179 -> 192.168.32.2(11005)]
Comment: TCP session was gracefully torn down and the router is waiting to close
the session between the two hosts on ports 179 and 110005
01:54:03: TCB8252932C created
01:54:03: TCP0: state was LISTEN -> SYNRCVD [179 -> 192.168.32.2(11006)]
Comment: BGP was listening for TCP connection request and received it on port
11006
01:54:03: TCP0: Connection to 192.168.32.2:11006, received MSS 1460, MSS is 516
01:54:03: TCP: sending SYN, seq 1620953691, ack 2271616142
01:54:03: TCP0: Connection to 192.168.32.2:11006, advertising MSS 1460
01:54:03: TCP0: state was SYNRCVD -> ESTAB [179 -> 192.168.32.2(11006)]
Comment: The TCP session between the two routers on port 179 and 11006 was
successfully established
01:54:03: TCB820E59F0 callback, connection queue = 1
01:54:03: TCB820E59F0 accepting 8252932C from 192.168.32.2.11006
01:54:03: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Up
Comment: BGP session is ESTABLISHED
01:54:26: TCP0: state was TIMEWAIT -> CLOSED [179 -> 192.168.32.2(11005)]
01:54:26: TCB 0x82528E90 destroyed
Comment: The old TCP session between ports 179 and 11005 was closed the TCB
marker for the session was destroyed
```

在验证了两台路由器之间的 TCP 会话工作正常后，可以使用表 8-5 中列出的命令来验证或是故障排查 BGP 会话。

表 8-5

BGP 邻居的显示和调试工具

命令	描述
<code>show ip bgp [ip-address prefix]</code>	显示了 BGP 表，包括总结、表的版本和表中列出的路径属性。可选的 IP 地址和前缀用来限制命令返回的信息
<code>show ip bgp neighbors [ip-address]</code>	命令显示了本地路由器配置的对等连接的每个邻居的详细信息，包括邻居的 BGP 版本、BGP 路由器识别符、有限状态机的状态、收到的报文数目和详细的 TCP 连接信息。可选的 IP 地址和前缀用来限制命令返回的信息
<code>show ip bgp summary</code>	命令显示了每个 BGP 邻居的总结信息，包括邻居的 BGP 路由器 ID、表的版本、从邻居收到的路径信息、这些路径的属性、已被发送或是接收到的报文的数目、有限状态机的状态以及邻居保持在已建立状态的时间
<code>debug ip bgp [ip-address]</code>	<code>debug ip bgp</code> 命令显示了所有 BGP 对等体关系的实时信息，包括了有限状态机的状态、发送和接收的报文、能力的协商和收到的路由
<code>debug ip bgp events</code>	命令显示了 BGP 事件的实时信息，包括 BGP scanning、本地通告的路由表、计时器以及发送和接收的报文
<code>debug ip bgp [ip-address] updates [access-list] [in out]</code>	<code>debug ip bgp update</code> 命令显示了从对等 BGP 邻居收到的 UPDATE 报文中路径的实时信息，其中包括收到的路径、路径到主 IP 路由表中的安装以及发送给邻居路由器的更新。参数 <code>IP-address</code> 使你指定从某个特定的邻居收到的更新。 <code>access-list</code> 命令使你限制为特定更新的输出。 <code>in</code> 和 <code>out</code> 参数使你指定进入还是发出的更新
<code>debug ip bgp in [ip-address]</code>	命令显示了在 BGP 会话中发送的进入报文的实时信息以及本地路由器从它的邻居收到的路径
<code>debug ip bgp out [ip-address]</code>	命令显示了在 BGP 会话中发送的输出报文的实时信息以及本地路由器发送给它的邻居的路径
<code>debug ip bgp keepalives</code>	命令显示了本地路由器发送和接收到的保活报文的实时信息
<code>debug ip routing</code>	当 BGP 路由没有被加入主 IP 路由表的时候本命令能够帮助诊断问题

一、show ip bgp 命令

`show ip bgp` 命令是一个很方便的命令，使你可以验证本地 BGP 的配置，检查路径属性和故障排查 BGP 路径通告的问题。这个命令列出了每个路径的状态的简单总结；用来到达路径的下一跳；多出口鉴别器、本地优先、权重、AS 路径和起源属性。范例 8-16 列出了 `show ip bgp` 命令的输出，表 8-6 描述了命令的输出。

范例 8-16 show ip bgp 命令输出范例

Aiki# show ip bgp						
BGP table version is 5, local router ID is 172.16.20.1						
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal						
Origin codes: i - IGP, e - EGP, ? - incomplete						
Network	Next Hop	Metric	LocPrf	Weight	Path	
*> 10.1.1.0/24	192.168.32.2	0		0	600	i
*> 10.1.2.0/24	192.168.32.2	0		0	600	i
*> 172.16.14.0/24	0.0.0.0	0		32768		i
*> 172.16.20.0/24	0.0.0.0	0		32768		i

`show ip bgp regexp` 命令也可以和一个正则表达式一起使用来创建 AS 路径访问列表，或是仅仅用来找到所有来自于某个特定的自治系统的路由。AS 访问列表和正则表达式在第 9 章中会介绍。

表 8-6 show ip bgp 命令输出解释

命令输出	输出描述
BGP table version is 5	当前的 BGP 版本，当每次表变化的时候这个数字会增加
local router ID is 172.16.20.1	本地 BGP 的路由器识别符，除非显式配置，这个数字通常是最高的环回 IP 地址，BGP 的路由器识别符可以通过 bgp router-id 命令显式地指定 注意 BGP 本地路由器 ID 和路由器用来建立 BGP 会话的接口不一样，最好的方法是应该总是配置路由器使用一个特定的路由器识别符，以避免当你加上一个新的 BGP 对等体或是想通过多个 BGP 路径负载均衡的时候发生任何问题。当解决 BGP 的连接问题时，如果某个 BGP 对等体没有使用正确的 IP 地址（BGP 路由器识别符）来表示它的远端对等体，BGP 会话将无法建立。当你使用的路由器只通过一个直连端口连接一个 E-BGP 对等体的时候这可能不是一个问题，当路由器有非直接连接的多个 E-BGP 对等体时，你可能需要使用 ebgp-multihop 命令来指定没有直连的对等体，在本章的后面会介绍 ebgp-multihop 命令
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal	状态代码表示了 BGP 表中每个路径的状态 suppressed (s) —被本地 BGP 配置抑制的路由，仍旧在本地 BGP 表中，但是不会通告给远端对等体 dampened (d) —被远端对等体衰减的路由 history (h) —表示这条路由上已经启用了衰减 valid (*) —路由已经被验证是可达的，没有星号标记的路由不会被 BGP 使用，更不会被放入主路由表中 best (>) —到达目的网段的最佳路径。BGP 保存了到每一个网段的所有路径，但是它只会将最佳路径放入主路由表中，也只会通告最佳路径给它的邻居 internal (i) —BGP 通过一个内部网关协议进程学到的路由
Origin codes: i - IGP e - EGP ? - incomplete	起源代码是路由的起源属性，在本命令输出的每个路径的最右端就是起源代码 i - IGP —通过 I-BGP 会话学到，大多数的路由开始都是通过本地配置学到的，所以起源代码都是 i e - EGP —通过 EGP 会话学到，除非路由器和一个 EGP 对等体对接，否则很少会看到起源代码为 e 的路由 ? - INCOMPLETE —这些路由来自于未知的起源，这个起源通常用在路由是 BGP 从 IGP 重分发而学到的情况下
Network	网段在命令输出中用 IP 地址 / 子网掩码的格式表达
Next Hop	网段的下一跳属性，这是 BGP 用来到达网段的下一跳地址，如果下一跳不可达，路由将不会被标记为有效的 BGP 也会将下一跳属性传递给主 IP 路由表，当和 I-BGP 一起使用但是下游的路由器无法到达下一中继地址的时候可能会造成可达性的问题
Metric	当到某个网段有多个出口的时候使用多出口鉴别器属性，默认值为 0，必须被显式配置
LocPrf	路径的本地优先属性用来说明到某个网段的本地优选路径，I-BGP 对等体的默认本地优先是 100
Weight	路径的本地配置的权重属性，本地产生的路由的默认权重是 0，从其他对等体处学到的路由的默认权重是 32 768 记住权重属性是思科的专有属性，它只在本地有效，不会传递给其他任何对等体
Path	AS 路径属性列出了路由经过的 E-BGP 自治系统，AS 路径的最右边的记录是起源的自治系统 本地产生的路由，也就是本地自治系统起源的路径，在离开自治系统之前不包含自治系统路径记录

二、show ip bgp neighbors 命令

show ip bgp neighbors 命令是用来故障排查和验证 BGP 对等体会话的常用命令之一，这个命令显示了关于每个 BGP 对等体会话和每个会话的 TCP 参数的很多详细信息。当故障排查 BGP 问题的时候这个命令的一些输出行被证明是非常有用的，这个命令将是你使用 BGP 的最佳工具之一。范例 8-17 显示了 Alki 路由器上 **show ip bgp neighbors** 命令的输出。

范例 8-17 show ip bgp neighbors 命令输出

```
Alki# show ip bgp neighbors
BGP neighbor is 192.168.32.2, remote AS 600, external link
```

(待续)

```
BGP version 4, remote router ID 192.168.32.2
BGP state = Established, up for 01:15:35
Last read 00:00:34, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  Route refresh: advertised and received(old & new)
  Address family IPv4 Unicast: advertised and received
Received 168 messages, 0 notifications, 0 in queue
Sent 174 messages, 0 notifications, 0 in queue
Route refresh request: received 0, sent 0
Default minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
BGP table version 5, neighbor version 5
Index 1, Offset 0, Mask 0x2
2 accepted prefixes consume 72 bytes
Prefix advertised 12, suppressed 0, withdrawn 0
Number of NLRI in the update sent: max 2, min 0
Connections established 6; dropped 5
Last reset 01:16:14, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 192.168.32.1, Local port: 179
Foreign host: 192.168.32.2, Foreign port: 11006
Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)
Event Timers (current time is 0xADA668):
Timer           Starts    Wakeups    Next
Retrans          81         0         0x0
TimeWait         0         0         0x0
AckHold         79         40        0x0
SendWnd         0         0         0x0
KeepAlive        0         0         0x0
GiveUp          0         0         0x0
PmtuAger        0         0         0x0
DeadWait        0         0         0x0
iss: 1620953691 snduna: 1620955275 sndnxt: 1620955275 sndwnd: 16270
irs: 2271616141 rcvnxt: 2271617706 rcvwnd: 16289 delrcvwnd: 95
SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 1460 bytes):
Rcvd: 125 (out of order: 0), with data: 79, total data bytes: 1564
Sent: 122 (retransmit: 0, fastretransmit: 0), with data: 80, total data bytes:
1583
```

这个命令可以故障排查一个主机的问题，优化 BGP 的性能和验证配置。比如，通过输入命令 **show ip bgp neighbors | include BGP state** 可以看到当前 BGP 的状态和邻居关系已经建立的时间，通过命令 **show ip bgp neighbors | include accepted** 可以看到已经收到的前缀数目以及它们消耗的内存。通过命令 **show ip bgp neighbors | include Connections** 可以看到对等体已经建立和结束的连接数量，通过 **show ip bgp neighbors | include Last reset** 可以迅速发现上次连接重启的原因。命令 **show ip bgp neighbors** 输出的详细解释见表 8-7。

表 8-7

show ip bgp neighbors 命令输出的解释

命令输出	输出描述
BGP neighbor is 192.168.32.2	远端 BGP 对等体的 IP 地址
remote AS 600	远端 BGP 的自治系统号码
external link	BGP 会话类型

续表

命令输出	输出描述
BGP version 4	与 BGP 远端对等体之间会话的 BGP 版本号（双方都同意的）
remote router ID 192.168.32.2	远端 BGP 对等体的 BGP 路由器识别符，记住它不总是直连接口的 IP 地址
BGP state = Established	当前的 BGP 有限状态机的状态 以下为可能的状态： Idle Connect Active OpenSent OpenConfirm Established 你可能只会看到 Idle、Active 和 Established 状态
up for 01:15:35	当前 BGP 会话已经建立的时间（用“for: hours, minutes, and seconds”的格式）
Last read 00:00:34	最后一次从远端对等体收到和读取报文的时间
hold time is 180	当前的保持时间值，是从它的对等体接收报文的间隔，默认的保持时间是 180s，也就是 3 倍的 Keepalive 计时器值
keepalive interval is 60 seconds	这个会话的 Keepalive 计时器间隔。Keepalive 计时器指定了 BGP 对等体在发送一个保活报文之前需要等待的时间，如果在 3 个 Kcepalive 间隔内没有收到一个保活报文，那么保持计时器超时，一个通知报文将被发送，同时会话结束
Neighbor capabilities: Route refresh: advertised and received (old & new) Address family IPv4 Unicast: advertised and received	在本地和远端对等体之间的会话协商后的能力，关于 BGP 能力的列表请参考第 7 章的相关内容，路由刷新能力允许在不清除 BGP 会话的情况下动态地输入和输出更新的请求，根据你的配置在这个字段可能会出现不同的 IPv4 地址族： <ul style="list-style-type: none">• IPv4 unicast• IPv4 multicast• VPNv4 unicast IPv4 单播地址族能力允许传播和接收 IPv4 单播路径 IPv4 组播地址族能力允许传播和接收 IPv4 组播路径，是多协议 BGP 的功能 IPv4 虚拟网地址族能力允许传播和接收 IPv4 虚拟网单播路径
Received 168 messages	这个对等体收到的总 BGP 报文数，包括以下报文： <ul style="list-style-type: none">• OPEN• UPDATE• KEEPALIVE• NOTIFICATION
0 notifications	这个对等体收到的通知报文数 通知报文表明了错误的情况，当收到后应该检查、监控和记录
0 in queue	等待被处理的报文数，如果在队列中报文数过大可能表明拥塞，缺少内存和处理器时间，以及很多 BGP 对等体在不停地发送报文。当一个生产路由器正在与许多对等体交换更新报文的时候这个队列中通常包含报文，如果这种情况持续存在，这时就应该检查如何提高路由器的 BGP 处理性能

续表

命令输出	输出描述
Sent 174 messages	本地对等体发送给远端对等体的报文总数, 包括以下报文类型: <ul style="list-style-type: none">• OPEN• UPDATE• NOTIFICATION• KEEPALIVE
0 notifications	从本地路由发送给远端对等体的通知报文数
0 in queue	在队列中等待被传输的报文数
Route refresh request: received 0, sent 0	向远端对等体发送或是从远端对等体接收到的路由刷新报文数
Default minimum time between advertisement runs is 30 seconds.	在 UPDATE 报文之间的默认的最小时间
For address family: IPv4 Unicast	下个字段中提到的 BGP 表的地址族
BGP table version 5	当前的本地 BGP 表的版本, 每次发生变化后这个数字就会增加, 不一致的表的版本可能意味着 BGP 对等体之间的问题
neighbor version 5	当前的远端 BGP 表的版本
Index 1, Offset 0, Mask 0x2	内部 BGP 表的信息
2 accepted prefixes consume 72 bytes	本地对等体接受的前缀数目, 以及这些前缀消耗的内存字节数量
Prefix advertised 12	本地对等体通告的前缀数目
suppressed 0	本地对等体抑制的前缀数目
withdrawn 0	本地对等体已经撤销的前缀数目 大量的撤销路由表明路由的不稳定, 可以通过解决不稳定性或是增加往 null 接口的高管理距离的静态路由来更正
Number of NLRI in the update sent: max 2, min 0	在 UPDATE 报文中发送的 NLRI 或是路径数目 最大值—说明在单个 UPDATE 报文中曾经发送的最大的 NLRI 数目 最小值—说明单个 UPDATE 报文中曾经发送的最小的 NLRI 数目
Connections established 6; dropped 5	在最近的一次路由器重启后本地和远端对等体曾经建立的会话数目。较大的结束会话数字表示了路由振荡的情况, 应该解决来避免路由衰减
Last reset 01: 16: 14, due to User reset	最近一次 BGP 会话重启的时间 (用 hours: minutes: seconds 的格式) 和原因
Connection state is ESTAB, I/O status: 1, unread input bytes: 0 Local host: 192.168.32.1, Local port: 179 Foreign host: 192.168.32.2, Foreign port: 11006 Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes) Event Timers (current time is 0xADA668): Timer Starts Wakeups Next Retrans 81 0 0x0 TimeWait 0 0 0x0 AckHold 79 40 0x0 SendWnd 0 0 0x0 KeepAlive 0 0 0x0 GiveUp 0 0 0x0	show ip bgp neighbors 命令输出的其余部分和 show tcp 命令一致, 关于这些项目的细节请参考表 8-3

续表

命令输出	输出描述
<pre>PmtuAger 0 0 0x0 DeadWait 0 0 0x0 iss: 1620953691 snduna: 1620955275 sndnxt: 1620955275 sndwnd: 16270 irs: 2271616141 rcvnxt: 2271617706 rcvwnd: 16289 delrcvwnd: 95 SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms Flags: passive open, nagle, gen tcbs Datagrams (max data segment is 1460 bytes): Rcvd: 126 (out of order: 0), with data: 79. total data bytes: 1564 Sent: 122 (retransmit: 0, fastretransmit: 0), with data: 80, total data bytes: 1583</pre>	

当故障排查的时候，**show ip bgp neighbors** 命令的基本部分可以被输出修饰符（如果配置了的话也可以是命令别名）解析，以显示特定的命令输出部分。如范例 8-18 所示，也可以使用带有关键词 **ip-address advertised-networks** 和 **ip-address routes** 的这个命令来显示发送给某个特定的邻居或是从它那里接收到的信息。

范例 8-18 使用 **show ip bgp neighbors** 命令显示 BGP 路由通告

```
Madison# show ip bgp neighbors 192.168.32.1 advertised-routes
BGP table version is 3, local router ID is 10.1.1.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop              Metric LocPrf Weight Path
*> 6.0.0.0          0.0.0.0                  0           32768 i
Madison# show ip bgp neighbors 192.168.32.1 routes
BGP table version is 3, local router ID is 10.1.1.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop              Metric LocPrf Weight Path
*> 5.0.0.0          192.168.32.1            0           0 5300 i

Total number of prefixes 1
```

前面范例的第一部分显示了如何使用 **show ip bgp neighbors 192.168.32.1 advertised-routes** 命令来查看发送给对等体 192.168.32.1 的路由，范例的第二部分显示了如何使用 **show ip bgp neighbors 192.168.32.1 routes** 命令来查看从对等体 192.168.32.1 接收到的路由。当故障排查 BGP 路由策略的时候这些命令被证明非常有用。

三、**show ip bgp summary** 命令

show ip bgp summary 命令显示了包含每个邻居的 **show ip bgp neighbors** 命令输出的汇总信息，这个命令使你能够获得每个 BGP 对等会话状态的简单影像，能够故障排查连接或是性能问题，而且能够检查 BGP 用来存放路径信息的内存数量。范例 8-19 显示了 **show ip bgp summary** 命令的输出，表 8-8 显示了输出的详细描述。

范例 8-19 show ip bgp summary 命令的输出

```
Alki# show ip bgp summary
BGP router identifier 172.16.20.1, local AS number 5300
BGP table version is 5, main routing table version 5
4 network entries and 4 paths using 532 bytes of memory
2 BGP path attribute entries using 120 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.32.2    4    600     20     21       5     0     0 00:16:47      2
```

表 8-8 show ip bgp summary 命令输出的解释

命令输出	输出描述
BGP router identifier 172.16.20.1.	本地的 BGP 路由识别符
localAS number 5300	本地的自治系统号码
BGP table version is 5.	本地 BGP 表的版本
main routing table version 5	主 IP 路由表的版本
network entries and paths using 532 bytes of memory	网段记录数目，路径的数目，以及被这些记录消耗的内存
2 BGP path attribute entries using 120 bytes of memory	BGP 路径属性记录的数目以及这些记录消耗的内存数量
1 BGP AS-PATH entries using 24 bytes of memory	AS 路径记录的数目以及这些记录使用的内存数量
0 BGP route-map cache entries using 0 bytes of memory	路由映射缓存记录数目以及它们消耗的内存数量
0 BGP filter-list cache entries using 0 bytes of memory	过滤列表缓存记录数目以及这些记录消耗的内存数量
BGP activity 4/0 prefixes	本地 BGP 表中包含的前缀数目
4/0 paths	本地 BGP 表中包含的路径数目
scan interval 60 secs	BGP 扫描器扫描 BGP 表查找变化和可达性的间隔，默认值是 60s，可以使用 bgp scan-time 命令小心地将其改为 5~ 60s 之间的值
Neighbor 192.168.32.2	远端对等体的 IP 地址
V 4	远端对等体的 BGP 版本
AS 600	远端对等体的自治系统号码
MsgRcvd 20	从远端对等体接收到的报文数目（包括 OPEN、UPDATE、NOTIFICATION 和 KEEPALIVE）
MsgSent 21	发送给远端对等体的报文数目（包括 OPEN、UPDATE、NOTIFICATION 和 KEEPALIVE）
TblVer 5	最近一次发送给远端对等体的 BGP 表的版本
InQ 0	等待处理的接收到的报文数目
OutQ 0	等待传输的发送报文数目
Up/Down 00: 16: 47	两个对等体之间 BGP 会话起来或是结束的持续时间

续表

命令输出	输出描述
State/PfxRcd 2	当 BGP 会话建立后从远端对等体接收到的前缀数目，如果当前不是“已建立”状态那么 BGP 有限状态机的状态： <ul style="list-style-type: none">• 空闲• 连接• 激活• Oper: 发送• Open 确认

现在你已经将 BGP 的 `show` 和 `debug` 命令放入了故障排查工具箱，在下一小节我们将介绍和解释另外一个故障排查工具：使用 BGP 报文。

8.2.1 使用 BGP 报文作为征兆

使用 BGP 报文作为诊断工具是最好的故障排查 BGP 问题的方法之一，思科 IOS 软件根据情况可以有一些不同的方式来显示报文。作为一个常用的最优方法，可能要使用 `no logging console` 命令关闭主控日志而通过虚终端来进行所有的配置和故障排查。除非你每次故障排查的时候都使用 `terminal monitor` 命令，报文通常不会直接发送到虚终端上，你可能不会看到 BGP 报文的输出，可以使用 `logging buffered` 命令打开带缓冲的日志功能将报文存放在内存中。

另外一个容易忽略的思科 IOS 软件特性是日志配置，默认的行为是根据路由器的在线时间记录每个事件。你可能觉得这样比较好，你也可能想让路由器以日期/时间的格式显示报文，可以通过命令 `service timestamps debug datetime msec` 和 `service timestamps log datetime msec` 来配置。使用这些命令后，路由器将不会使用在线时间而是使用组合的日期/时间来显示事件，这使得故障排查在过去某天或是某个小时发生的事件变得很方便。

配置了路由器的日志风格后，可以使用软件进程产生的报文进行故障排查。思科 IOS 软件根据严重性的变化有 5 个主要的报文记录情况，如表 8-9 所示。

表 8-9 思科 IOS 软件事件情况

事件情况号	事件情况	情况描述
2	Critical	需要立刻采取行动的严重情况
3	Error	需要立刻采取行动的错误情况
4	Warning	警告情况表明发生了一个可能导致问题的事件
5	Notification	通知报文显示了关于一个重要但是尚且正常的事件的信息
6	Informational	情报报文是关于一个存在但是对路由器的运行不会有重要影响的问题

BGP 报文显示的格式如图 8-2 所示。

根据前面图中的输出显示，你可以看到与邻居 192.168.32.2 发生一次 BGP 邻接状态变化，BGP 邻接状态变为在线。范例 8-20 显示了 `show logging` 命令的输出如何使你能够在几秒之内诊断和故障排查一个 BGP 路由抖动的问题。

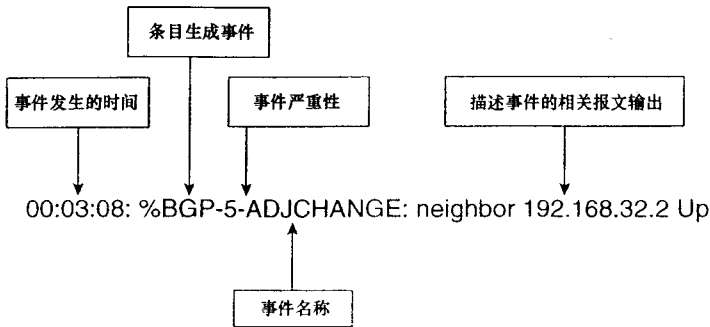


图 8-2 思科 IOS 软件报文格式

范例 8-20 来自 show logging 命令的报文

```
00:00:51: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
00:00:52: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to down
00:02:23: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
00:02:24: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
00:03:08: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Up
00:44:23: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
00:44:23: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Down Interface flap
00:44:24: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to down
00:46:49: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
00:46:50: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
state to up
00:47:22: %BGP-5-ADJCHANGE: neighbor 192.168.32.2 Up
```

在这个范例中，你可以看到串行接口 0/0 持续地在激活和断掉之间转换，导致和邻居 192.168.32.2 的 BGP 对等体关系也随之抖动。在 BGP 报文中显示的 LINK-3-UPDOWN 报文使你很容易地诊断 BGP 路由抖动问题的症状，在这个时候，很容易地将 BGP 路由抖动问题隔离为串行接口 0/0 的连接问题。表 8-10 列出了 BGP 报文和它们的描述。

表 8-10

BGP 报文

BGP 报文	报文描述
%BGP-2-INSUFMEM	这是一个严重的 BGP 报文，表明路由器没有足够的内存继续运行指定的操作，这个错误常常会在没有足够内存来处理 BGP 运行的路由器上发生（你可能会在 2500 系列的实验室路由器上打开 BGP 调试导致重启之前看到这个错误信息），为了解决这个情况，需要升级路由器，如果没有超过当前的内存配置，只需要在路由器上增加内存，或者使用 show memory 命令查找不必用的进程并且将其关闭，如果一个实验室路由器（请不要在生产路由器上）没有足够的能力来运行 BGP，可能需要在打开调试之前保存配置以免重启后配置丢失
%BGP-3-ADDRROUTE	这个错误报文表明路由器无法增加路由的错误情况
%BGP-3-BADMASK	这个错误报文表明由于错误报文中指定的前缀的子网掩码出错导致路由器无法将其加入到本地路由表中
%BGP-3-BADROUTEMAP	这个错误报文表明错误报文中指定的某个路由映射的使用方式不合适
%BGP-3-BGP_INCONSISTENT	这个错误表明 BGP 的数据结构不一致，这是一个内部 BGP 错误
%BGP-3-DELPATH	这个错误表明当试图删除一个路径时出错
%BGP-3-DELROUTE	这个错误表明试图从路由器的称为 Radix Trie 的内部 BGP 数据结构中删除路由时出错，这是一个内部 BGP 错误

续表

BGP 报文	报文描述
%BGP-3-INSUFCHUNKS	这个错误表明没有定义足够的块，思科 IOS 软件为进程分配块，这和内存的分配类似
%BGP-3-MARTIAN_IP	这个错误报文表明本地 BGP 发言人从远端路由器收到了一个无效的 IP 地址或是前缀的路由
%BGP-3-MAXPATHS	这个错误报文表明到一个目的网段有太多的等值路径，错误报文的输出包括和错误相关的 IP 前缀和掩码以及当前允许的最大路径数目，可以在 BGP 路由配置模式下使用 <code>maximum-paths</code> 命令来解决这个问题，可以指明一个路径数的较大值（从 1~6）
%BGP-3-MAXPFEXCEEDED and %BGP-4-MAXPFX:	这些报文表明相邻的 BGP 发言人发送了本地 BGP 发言人配置允许接收的前缀数，在报文的输出中显示了远端 BGP 发言人的 IP 地址和十进制的最大前缀数，%BGP-3 报文指明了最大的前缀数已经达到，连接将被断掉，%BGP-4 报文只是一个警告，说明允许的前缀数已经被超过，收到的报文类型取决于本地的 BGP 配置，用来配置最大前缀数目限制的命令将在第 9 章介绍
%BGP-3-NEGCOUNTER	这个报文表明当接收到的前缀计数器值小于 0 的时候发生 BGP 内部错误
%BGP-3-NOBITFIELD	这个错误报文表明路由器无法为报文输出中显示的对等体建立一个索引记录。当路由器没有足够的内存来与远端对等体建立 BGP 会话的时候通常发出这个报文，可以通过增加内存或是关闭不必要的进程来解决这个情况
%BGP-3-NOTIFICATION	这个错误报文表明路由器向报文中指定的远端对等体发送或是接收了通知报文，通知报文的类型也会显示在报文输出中，与远端对等体的会话结束
%BGP-3-RADIXINIT	这个错误报文表明本地路由器由于无法分配足够的内存所以无法建立 BGP Radix Trie，可以通过增加内存或是关闭不必要的进程来解决这个情况
%BGP-5-ADJCHANGE	这个通知报文表明与报文输出中指定的对等体的邻接关系发生了变化，它的输出同时还说明了 BGP 邻接关系是变化到了已建立还是空闲状态
%BGP-5-VERSION_WRAP	这个通知报文表明本地的 BGP 表超过了最大允许大小，有重叠发生
%BGP-6-AS-PATH	这个报文表明本地路由器收到了一个包含无效的 AS 路径的 UPDATE 报文，报文的输出中包括了错误的 AS 路径属性和发送者的 IP 地址
%BGP-6-NEXTHOP	这个报文当本地路由器收到了一个有非法的下一跳属性的更新时出现，当这个事件发生时，路由将被忽略，BGP 继续运行，报文的输出中包括了 UPDATE 报文中收到的前缀的 IP 地址以及发送这个报文的邻居

8.2.2 BGP 空闲/活动场景

你可能记得在前一章中我们提到 BGP 有限状态机（FSM）在 BGP 邻居到达实际开始发送和接收更新的已建立状态之前需要经过一些其他的状态，作为一个简单的回顾，图 8-3 显示了 BGP 有限状态机如何从空闲转变到已建立状态。

注意，如果在连接和 Open 发送状态之间发生错误，有限状态机会过渡到激活状态。如果有限状态机仍然无法从激活状态转变到连接或是 Open 发送状态，它将返回空闲状态。由于路由器在等待进入下一状态之前只保持连接和 Open 发送状态很短的时间，在故障排查的时候一个值得注意的症状就是对等体一直在激活和空闲状态之间变化。如果你注意到当一个 TCP 会话建立后对等体在连接和 Open 发送状态之间过渡，这个故障通常是由于 TCP 会话引起的。如果这时使用分层故障排查方法，对于对等体在空闲和激活状态之间持续变化的问题，你在检查 BGP 之前将验证第一层到第三层工作正常。

以图 8-4 中的 Alien 网络为例，在这个范例中，自治系统 22801 中的路由器 Mulder 和 Scully 被配置为互相对等连接，由于两台路由器都属于自治系统 22801，它们是 I-BGP 对等体，也不必须要直接相连。因此，路由器 Mulder 和路由器 Krycek 通过网段 148.201.100.0/24 相连，路由器 Krycek 和路由器 MrX 通过网段 148.202.100.0/24 相连，最终网段 148.202.100.0/24 连接到路由器 Scully 的 148.203.100.0/24 网段。

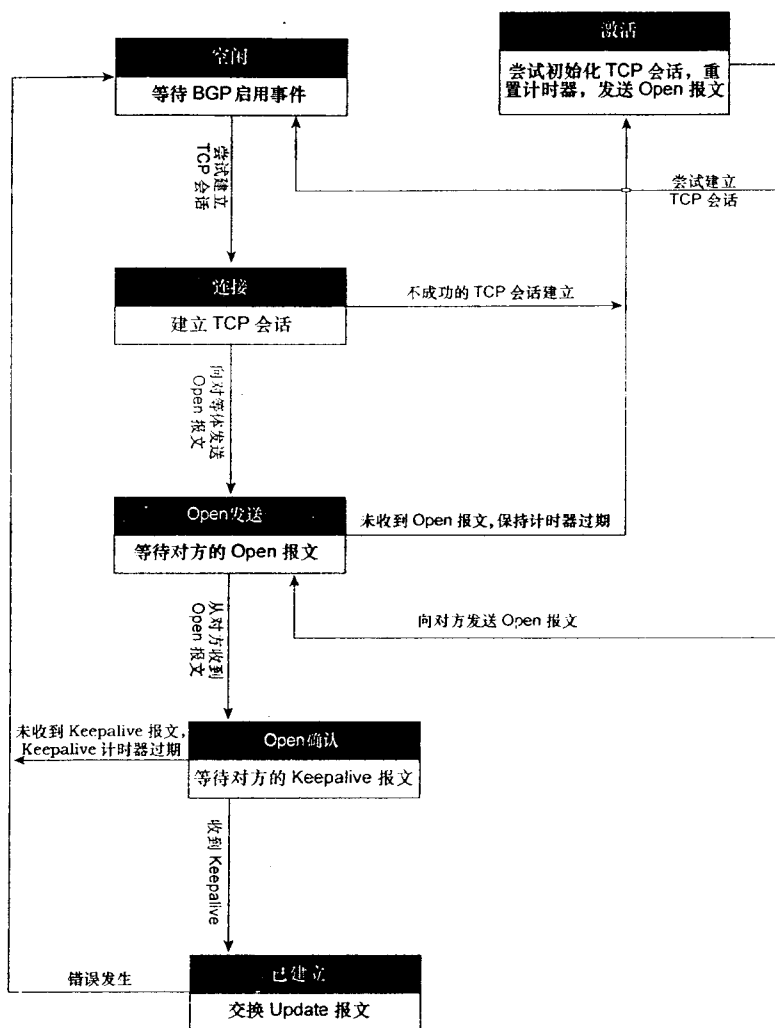


图 8-3 BGP 有限状态机回顾

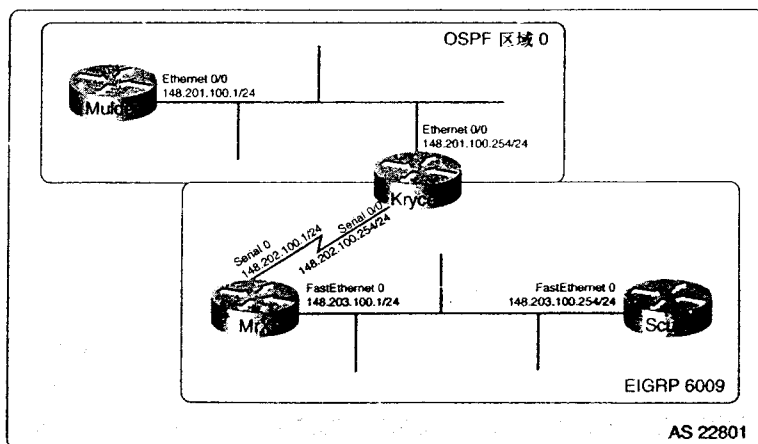


图 8-4 Alien 网络

在配置 BGP 后, 当输入 **show ip bgp summary** 命令后你会发现路由器停留在空闲和激活状态。范例 8-21 显示了路由器 Mulder 的配置, Mulder 路由器通过快速以太网接口 0 连接到路由器 Krycek, 接口运行在 OSPF 区域 0 内。

范例 8-21 路由器 Mulder 的配置

```
hostname Mulder
<text omitted>
!
interface Ethernet0
 ip address 148.201.100.1 255.255.255.0
!
router ospf 1
 network 148.201.100.0 0.0.0.255 area 0
!
router bgp 22801
 bgp log-neighbor-changes
 network 10.1.1.0 mask 255.255.255.0
 network 10.2.2.0 mask 255.255.255.0
 neighbor 148.203.100.254 remote-as 22801
```

路由器 Krycek 连接到路由器 Mulder 的以太网接口 0/0 上, 接口运行在 OSPF 区域 0 内, 路由器 Krycek 同时也通过一个串行接口连接到 MrX 路由器上。Mrx 路由器上运行着 EIGRP 进程 6009。范例 8-22 显示了路由器 Krycek 的配置以及关于往返 Mulder 和 Scully 网段的连接性的 **show ip route** 命令的输出。

范例 8-22 路由器 Krycek 的配置

```
hostname Krycek
<text omitted>
!
interface Ethernet0/0
 ip address 148.201.100.254 255.255.255.0
!
interface Serial0/0
 ip address 148.202.100.254 255.255.255.0
!
router eigrp 6009
 passive-interface Ethernet0/0
 network 148.202.0.0
 auto-summary
!
router ospf 1
 passive-interface Serial0/0
 network 148.201.100.0 0.0.0.255 area 0
!
Krycek# show ip route
 148.201.0.0/24 is subnetted, 1 subnets
C    148.201.100.0 is directly connected, Ethernet0/0
 148.202.0.0/24 is subnetted, 1 subnets
C    148.202.100.0 is directly connected, Serial0/0
D    148.203.0.0/16 [90/2172416] via 148.202.100.1, 00:45:21, Serial0/0
```

范例 8-23 显示了路由器 MrX 的配置, Mrx 通过串行接口 0 连接到路由器 Krycek, 通过快速以太网接口 0 连接到路由器 Scully。

范例 8-23 路由器 MrX 的配置

```
hostname MrX
<text omitted>
!
interface Serial0
 ip address 148.202.100.1 255.255.255.0
!
interface FastEthernet0
 ip address 148.203.100.1 255.255.255.0
!
router eigrp 6009
 network 148.202.0.0
 network 148.203.0.0
 auto-summary
```

最后，范例 8-24 显示了路由器 Scully 的配置。

范例 8-24 路由器 Scully 的配置

```
hostname Scully
<text omitted>
!
interface FastEthernet0
 ip address 148.203.100.254 255.255.255.0
!
router eigrp 6009
 network 148.203.0.0
 auto-summary
!
router bgp 22801
 bgp log-neighbor-changes
 network 192.168.8.0
 network 192.168.9.0
 neighbor 148.201.100.1 remote-as 22801
```

范例 8-25 显示了命令 **show ip bgp summary** 和 **show ip bgp neighbors** 的输出，对故障发生的原因给出了一些提示。

范例 8-25 故障排查命令细节

```
Scully# show ip bgp summary
BGP router identifier 192.168.1.1, local AS number 22801
BGP table version is 1, main routing table version 1
Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
148.201.100.1  4 22801      0       0        0    0    0 never    Active
Scully# show ip bgp neighbor
BGP neighbor is 148.201.100.1, remote AS 22801, internal link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active
  Last read 00:23:24, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 5 seconds
For address family: IPv4 Unicast
BGP table version 1, neighbor version 0
```

(待续)

```

Index 1, Offset 0, Mask 0x2
0 accepted prefixes consume 0 bytes
Prefix advertised 0, suppressed 0, withdrawn 0
Connections established 0; dropped 0
Last reset never
No active TCP connection

```

注意 **show ip bgp summary** 命令显示远端对等体 148.201.200.1 处于激活状态，在连接上没有发送或是接收任何报文，这表明在对等体之间还没有成功地建立过 BGP 会话。下面注意到 **show ip bgp neighbor** 命令的输出中没有远端主机的 BGP 路由识别符，这表明本地主机从来没有连接到远端主机去学到路由器识别符。同时注意到没有会话被建立或是关闭，没有被重启的连接，目前没有活动的 TCP 连接。如果你遵循本章前面提到的故障排查方法，那么应该使用以下步骤来调查 TCP 会话无法建立的原因。

第 1 步 验证第一层的连通性。

- 使用 **show** 命令验证路由器 Mulder 和 Scully 的以太网接口都是起来的。
- 验证在 Mulder 和 Scully 之间的每台路由器都是工作正常的。

第 2 步 验证第二层的连通性。

- 检查确认在路由器 Mulder 和 Scully 之间的任何路由器都没有第二层的问题。

第 3 步 验证第三层的连通性。

- 验证路由器 Mulder 和 Scully 之间的第三层的连通性。
- 从 Mulder 路由器 ping Scully 路由器，检查本地路由表中有没有到远端对等体网络的路由。

```

Mulder# ping 148.203.100.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 148.203.100.254, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Mulder# show ip route 148.203.100.0
% Network not in table

```

现在可以确定是 Mulder 和 Scully 网络之间的第三层路由有问题。由于 I-BGP 需要一个 IGP 来提供对等体之间下层的网络连通性，所以路由器 Mulder 和 Scully 之间无法建立 TCP 会话来满足完全连接的 BGP 对等体以及交换路由的需要。通过测试两个对等体之间的 IP 连通性，你会立刻发现路由器 Mulder 和 Scully 无法互相访问，你可以检查路由器 Krycek 的路由表并且作一些 ping 的测试。

```

Krycek# show ip route | begin Gateway
Gateway of last resort is not set
  148.201.0.0/24 is subnetted, 1 subnets
C    148.201.100.0 is directly connected, Ethernet0/0
  148.202.0.0/24 is subnetted, 1 subnets
C    148.202.100.0 is directly connected, Serial0/0
D    148.203.0.0/16 [90/2172416] via 148.202.100.1, 01:00:08, Serial0/0
Krycek# ping 148.201.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 148.201.100.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
Krycek# ping 148.203.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 148.203.100.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

```

现在你已经验证了路由器 Krycek 可以同时访问路由器 Mulder 和 Scully, 接下来你可以转到路由器 MrX 上验证 IP 的连通性。

```
MrX# show ip route | begin Gateway
Gateway of last resort is not set
  148.202.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       148.202.100.0/24 is directly connected, Serial0
D       148.202.0.0/16 is a summary, 01:17:13, Null0
  148.203.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       148.203.100.0/24 is directly connected, FastEthernet0
D       148.203.0.0/16 is a summary, 01:17:13, Null0
MrX# show ip route 148.201.100.0
% Network not in table
```

通过在路由器 MrX 上使用命令 `show ip route | begin Gateway`, 你会发现它没有到路由器 Mulder 的路由, 因此, 路由器 Scully 也没有到 148.201.100.0/24 网段的路由。在重新访问路由器 Krycek 后发现还没有配置 OSPF 和 EIGRP 之间的重分发, 当你采取行动解决这个问题后, 路由器 Mulder 和 Scully 之间的连接将会起来。

```
Mulder# show ip route
  10.0.0.0/24 is subnetted, 2 subnets
C       10.2.2.0 is directly connected, Loopback20
C       10.1.1.0 is directly connected, Loopback10
  148.201.0.0/24 is subnetted, 1 subnets
C       148.201.100.0 is directly connected, Ethernet0
  148.202.0.0/24 is subnetted, 1 subnets
O E1    148.202.100.0 [110/30] via 148.201.100.254, 00:02:26, Ethernet0
O E1    148.203.0.0/16 [110/30] via 148.201.100.254, 00:02:26, Ethernet0
Scully# show ip route
  148.201.0.0/24 is subnetted, 1 subnets
D EX    148.201.100.0 [170/2223616] via 148.203.100.1, 00:00:53, FastEthernet0
D       148.202.0.0/16 [90/2172416] via 148.203.100.1, 01:19:24, FastEthernet0
  148.203.0.0/24 is subnetted, 1 subnets
C       148.203.100.0 is directly connected, FastEthernet0
Scully# ping 148.201.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 148.201.100.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/37/40 ms
Scully# show ip bgp summary
BGP router identifier 192.168.1.1, local AS number 22801
BGP table version is 1, main routing table version 1
2 network entries and 2 paths using 266 bytes of memory
1 BGP path attribute entries using 60 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 2/0 prefixes, 4/2 paths, scan interval 15 secs
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down State/PfxRcd
148.201.100.1  4 22801      8        6        1    0    0 00:00:11      2
```

8.3 BGP 邻居的配置

在配置 BGP 之前, 理解 I-BGP 和 E-BGP 配置之间的基本规则是很重要的。在下一节我们会通过范例介绍这两种 BGP 类型, 说明如何配置 BGP 来支持不同的网络拓扑:

- 直接连接的 I-BGP 的配置;
- 通过 IGP 骨干的 I-BGP 连接的配置;
- 直接连接的 E-BGP;
- 多跳 E-BGP 的配置;

- E-BGP 传输自治系统的配置；
- 配置 BGP 对等体与 IGP 相互作用。

IBGP 对等关系

在第 7 章提到过，I-BGP 对等关系依赖于 I-BGP 发言人之间的全网状连接以及用来提供每个 BGP 对等体之间的基本路由的 IGP 路由协议产生的路由表。由于 I-BGP 对等体之间不需要直接连接，在两个 I-BGP 发言人之间可以有任意多个没有参加到 BGP 路由中的 IGP 路由器，只要这两个发言人有互相访问的路由，它们就可以建立 BGP 对等关系和交换 BGP 路由。

一、BGP 同步

作为一个规则，I-BGP 发言人在认为 BGP 路由可用之前必须使它们的 BGP 路由和 IGP 路由表同步。如果一个 I-BGP 对等体没有和它的 IGP 同步或是没有运行 IGP 进程，那么这个对等体将不会通告网段或是将 BGP 路由安装到主 IP 路由表中。有两个方法可以用来解决同步问题：首先，当有一个 IGP 已经在运行着但是你不想要用它同步，可以使用 **no synchronization** 命令；其次，如果你没有运行 IGP，可以使用 **no synchronization** 关闭 BGP/IGP 同步。

二、实际范例：I-BGP 同步试验

在这个范例中，I-BGP 用来通告以环回 IP 地址表示的远端 BGP 网段。本例演示了 IGP 同步是如何影响 BGP 路由的，以及 I-BGP 是如何在一个全网状的环境下运行的。图 8-5 显示了本例中的网络。

在这个范例中，使用的 IP 地址和 DLCI 见表 8-11。

表 8-11 实际范例的接口和 IP 地址

路由器	接口	串行封装和/或 DLCI	IP 地址
Sydney	Serial0	56 kbit/s PPP with Compression	15.1.15.1/24
Sydney	Loopback10	None	10.20.10.1/24
Sydney	Loopback20	None	10.20.20.1/24
Sloane	Serial0/0	56 kbit/s PPP with Compression	15.1.15.2/24
Sloane	Ethernet0/0	None	164.189.26.1/24
Khasinaw	FastEthernet0	None	164.189.26.2/24
Khasinaw	Serial1	Frame Relay DLCI 104	10.1.8.1/24
McCullough	Ethernet0	None	164.189.26.3/24
McCullough	Serial0	Frame Relay DLCI 105	10.1.9.1/24
Vaughn	Serial1	Frame Relay DLCI 401	10.1.8.2/24
Vaughn	Loopback10	None	192.168.40.1/24
Vaughn	Loopback20	None	192.168.60.1/24
Dixon	Serial1	Frame Relay DLCI 501	10.1.9.2/24
Dixon	Loopback10	None	10.50.5.1/24
Dixon	Loopback20	None	10.50.50.1/24

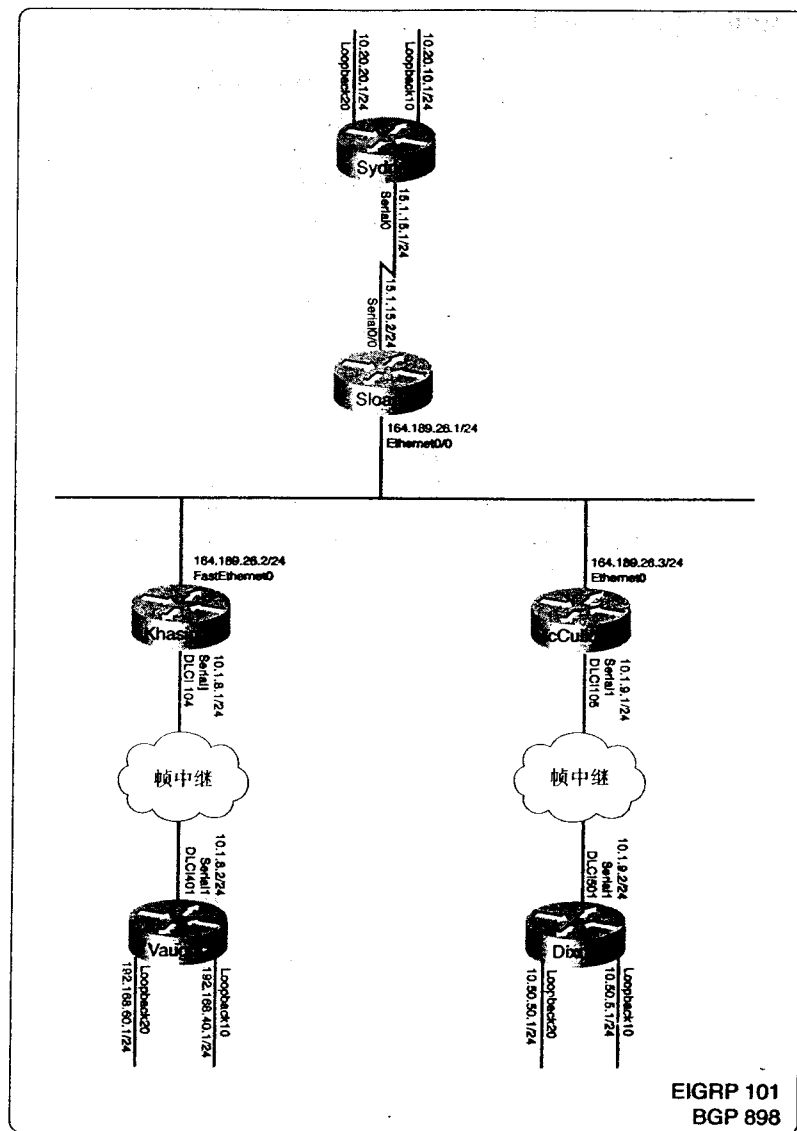


图 8-5 SD-6 网络

第 1 步 根据表 8-12 配置帧中继交换机。关于配置帧中继交换机的帮助，请参考《CCIE 实验指南（第 1 卷）》的第 1 章。

表 8-12 帧中继交换机的配置

接口	DLCI	接口	DLCI	接口	DLCI	接口	DLCI
串口 4	104	串口 2	401	串口 1	105	串口 3	501
串口 2	401	串口 4	104	串口 3	501	串口 1	105

范例 8-26 显示了帧中继交换机的配置以及配置完后呈现的帧中继路由。

范例 8-26 帧中继交换机的配置

```
hostname Frame-Relay-Switch
!
frame-relay switching
!
interface Serial1
 no ip address
 encapsulation frame-relay IETF
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 105 interface Serial3 501
!
interface Serial2
 no ip address
 encapsulation frame-relay IETF
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 401 interface Serial4 104
!
interface Serial3
 no ip address
 encapsulation frame-relay IETF
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 501 interface Serial1 105
!
interface Serial4
 no ip address
 encapsulation frame-relay IETF
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 104 interface Serial2 401
```

Frame-Relay-Switch# show frame-relay route

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial1	105	Serial3	501	active
Serial2	401	Serial4	104	active
Serial3	501	Serial1	105	active
Serial4	104	Serial2	401	active

第 2 步 使用表 8-11 中的 IP 地址和 DLCI 值配置 Khasinau 和 Vaughn 之间的帧中继，同时在路由器 Vaughn 上配置环回 IP 地址。在这个时候，你应该可以验证路由器 Khasinau 和 Vaughn 能够互相访问它们串行接口的 IP 地址。范例 8-27 显示了路由器 Khasinau 和 Vaughn 的帧中继配置。

范例 8-27 Khasinau 和 Vaughn 路由器的配置

```
hostname Khasinau
!
interface Serial1
 ip address 10.1.8.1 255.255.255.0
 encapsulation frame-relay IETF
 clockrate 1300000
 frame-relay map ip 10.1.8.2 104 broadcast
 frame-relay lmi-type ansi
!
hostname Vaughn
```

(待续)

```
!  
interface Loopback10  
  ip address 192.168.40.1 255.255.255.0  
!  
interface Loopback20  
  ip address 192.168.60.1 255.255.255.0  
!  
interface Serial1  
  ip address 10.1.8.2 255.255.255.0  
  encapsulation frame-relay IETF  
  clockrate 13000000  
  frame-relay map ip 10.1.8.1 401 broadcast  
  frame-relay lmi-type ansi
```

第 3 步 使用表 8-11 中的 IP 地址和 DLCI 值配置 McCullough 和 Dixon 之间的帧中继。此时，你也应该在 Dixon 路由器上配置环回 IP 地址，并且验证路由器 McCullough 和 Dixon 能够互相访问它们串行接口的 IP 地址。范例 8-28 显示了路由器 McCullough 和 Dixon 的帧中继配置。

范例 8-28 路由器 McCullough 和 Dixon 的配置

```
hostname McCullough  
!  
interface Serial1  
  ip address 10.1.9.1 255.255.255.0  
  encapsulation frame-relay  
  
  clockrate 13000000  
  frame-relay map ip 10.1.9.2 105 broadcast  
  frame-relay lmi-type ansi  
-----  
hostname Dixon  
!  
interface Loopback10  
  ip address 10.50.5.1 255.255.255.0  
!  
interface Loopback20  
  ip address 10.50.50.1 255.255.255.0  
!  
interface Serial1  
  ip address 10.1.9.2 255.255.255.0  
  encapsulation frame-relay IETF  
  clockrate 13000000  
  frame-relay map ip 10.1.9.1 501 broadcast  
  frame-relay lmi-type ansi
```

第 4 步 使用表 8-11 中的 IP 地址配置路由器 Sloane、Khasinau 和 McCullough 的以太网，然后在路由器 Sloane、Khasinau、Vaughn 和 McCullough 上启用 EIGRP 并将它们放入 EIGRP 自治系统 101 中。在路由器 Vaughn 和 Dixon 上不要将环回地址放入 EIGRP，在进入第 5 步之前验证所有的路由器都可以访问其他任意路由器的所有接口地址（环回地址除外）。范例 8-29 显示了路由器 Sloane、Khasinau、Vaughn、McCullough 和 Dixon 的以太网和 EIGRP 配置以及它们的路由表。

范例 8-29 路由器 Sloane、Khasinau、Vaughn、McCullough 和 Dixon 的以太网和 EIGRP 配置

```

hostname Sloane
!
interface Ethernet0/0
 ip address 164.189.26.1 255.255.255.0
!
router eigrp 101
 network 167.189.26.0 0.0.0.255
 no auto-summary
Sloane# show ip route
 10.0.0.0/24 is subnetted, 2 subnets
D    10.1.9.0 [90/2195456] via 164.189.26.3, 00:08:06, Ethernet0/0
D    10.1.8.0 [90/2195456] via 164.189.26.2, 00:01:50, Ethernet0/0
    164.189.0.0/24 is subnetted, 1 subnets
C    164.189.26.0 is directly connected, Ethernet0/0

hostname Khasinau
!
interface FastEthernet0
 ip address 164.189.26.2 255.255.255.0
!
router eigrp 101
 network 10.1.8.0 0.0.0.255
 network 164.189.26.0 0.0.0.255
 no auto-summary
Khasinau# show ip route
 10.0.0.0/24 is subnetted, 2 subnets
D    10.1.9.0 [90/2172416] via 164.189.26.3, 00:02:21, FastEthernet0
C    10.1.8.0 is directly connected, Serial0
    164.189.0.0/24 is subnetted, 1 subnets
C    164.189.26.0 is directly connected, FastEthernet0

hostname Vaughn
!
router eigrp 101
 network 10.1.8.0 0.0.0.255
 no auto-summary
Vaughn# show ip route
C    192.168.60.0/24 is directly connected, Loopback20
C    192.168.40.0/24 is directly connected, Loopback10
 10.0.0.0/24 is subnetted, 2 subnets
D    10.1.9.0 [90/2684416] via 10.1.8.1, 00:04:03, Serial1
C    10.1.8.0 is directly connected, Serial1
    164.189.0.0/24 is subnetted, 1 subnets
D    164.189.26.0 [90/2172416] via 10.1.8.1, 00:04:03, Serial1

hostname McCullough
!
interface Ethernet0
 ip address 164.189.26.3 255.255.255.0
!
router eigrp 101
 network 10.1.9.0 0.0.0.255
 network 164.189.26.0 0.0.0.255
 no auto-summary

```

(待续)

```

McCullough # show ip route
10.0.0.0/24 is subnetted, 2 subnets
C    10.1.9.0 is directly connected, Serial1
D    10.1.8.0 [90/2195456] via 164.189.26.2, 00:06:50, Ethernet0
164.189.0.0/24 is subnetted, 1 subnets
C    164.189.26.0 is directly connected, Ethernet0

-----

hostname Dixon
!
router eigrp 101
 network 10.1.9.0 0.0.0.255
 no auto-summary

-----

Dixon# show ip route
10.0.0.0/24 is subnetted, 4 subnets
C    10.1.9.0 is directly connected, Serial1
D    10.1.8.0 [90/2707456] via 10.1.9.1, 00:07:41, Serial1
C    10.50.50.0 is directly connected, Loopback20
C    10.50.5.0 is directly connected, Loopback10
164.189.0.0/24 is subnetted, 1 subnets
D    164.189.26.0 [90/2195456] via 10.1.9.1, 00:10:35, Serial1

```

第5步 配置路由器 Sydney 与 Sloane 之间的串行链路以及路由器 Sydney 的环回接口，然后启用 EIGRP 路由进程 101 来允许路由器 Sydney 可以 ping 通路由器 Vaughn 和 Dixon 上除了环回接口以外的所有接口。不允许路由器 Sydney 在 EIGRP 中通告它的环回接口。范例 8-30 显示了路由器 Sydney 和 Sloane 的配置以及路由表。

范例 8-30 路由器 Sydney 和 Sloane 的配置以及路由表

```

hostname Sydney
!
interface Loopback10
 ip address 10.20.10.1 255.255.255.0
!
interface Loopback20
 ip address 10.20.20.1 255.255.255.0
!
interface Serial0
 ip address 15.1.15.1 255.255.255.0
!
router eigrp 101
 network 15.1.15.0 0.0.0.255
 no auto-summary
!
Sydney# show ip route
10.0.0.0/24 is subnetted, 4 subnets
D    10.1.9.0 [90/2707456] via 15.1.15.2, 00:02:23, Serial0
D    10.1.8.0 [90/2707456] via 15.1.15.2, 00:02:23, Serial0
C    10.20.20.0 is directly connected, Loopback20
C    10.20.10.0 is directly connected, Loopback10
164.189.0.0/24 is subnetted, 1 subnets
D    164.189.26.0 [90/2195456] via 15.1.15.2, 00:02:23, Serial0
15.0.0.0/24 is subnetted, 1 subnets
C    15.1.15.0 is directly connected, Serial0

-----

hostname Sloane
!
interface Ethernet0/0

```

(待续)

```
ip address 164.189.26.1 255.255.255.0
!
interface Serial0/0
ip address 15.1.15.2 255.255.255.0
!
router eigrp 101
network 15.1.15.0 0.0.0.255
network 164.189.26.0 0.0.0.255
no auto-summary

Sloane# show ip route | begin Gateway
Gateway of last resort is not set

10.0.0.0/24 is subnetted, 2 subnets
D    10.1.9.0 [90/2195456] via 164.189.26.3, 00:07:09, Ethernet0/0
D    10.1.8.0 [90/2195456] via 164.189.26.2, 00:07:50, Ethernet0/0
164.189.0.0/24 is subnetted, 1 subnets
C    164.189.26.0 is directly connected, Ethernet0/0
15.0.0.0/24 is subnetted, 1 subnets
C    15.1.15.0 is directly connected, Serial0/0
```

第 6 步 在路由器 Sydney、Vaughn 和 Dixon 之间配置 BGP 来通告环回地址，将这些路由器放入 BGP 自治系统 898 中，不允许 BGP 对等体进行网络地址的自动汇总。使用 **show ip bgp** 命令来验证每个对等路由器的路由出现在 BGP 路由表中。范例 8-31 显示了每台路由器的 BGP 配置和它们的 BGP 路由表。

范例 8-31 路由器 Sydney、Vaughn 和 Dixon 的 BGP 配置和 BGP 路由表

```
Sydney# show run | begin bgp
router bgp 898
bgp log-neighbor-changes
network 10.20.10.0 mask 255.255.255.0
network 10.20.20.0 mask 255.255.255.0
neighbor 10.1.8.2 remote-as 898
neighbor 10.1.9.2 remote-as 898
no auto-summary

Sydney# show ip bgp
BGP table version is 3, local router ID is 10.20.20.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 10.20.10.0/24    0.0.0.0              0           32768 i
*> 10.20.20.0/24    0.0.0.0              0           32768 I
* i10.50.5.0/24     10.1.9.2              0         100      0 i
* i10.50.50.0/24    10.1.9.2              0         100      0 i
* i192.168.40.0     10.1.8.2              0         100      0 i
* i192.168.60.0     10.1.8.2              0         100      0 i

Vaughn# show run | begin bgp
router bgp 898
bgp log-neighbor-changes
network 192.168.40.0
network 192.168.60.0
neighbor 10.1.9.2 remote-as 898
neighbor 15.1.15.1 remote-as 898
no auto-summary
Vaughn# show ip bgp
```

(待续)

```
BGP table version is 3, local router ID is 196.168.60.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i10.20.10.0/24	15.1.15.1	0	100	0	i
* i10.20.20.0/24	15.1.15.1	0	100	0	i
* i10.50.5.0/24	10.1.9.2	0	100	0	i
* i10.50.50.0/24	10.1.9.2	0	100	0	i
*> 192.168.40.0	0.0.0.0	0		32768	i
*> 192.168.60.0	0.0.0.0	0		32768	i

```
Dixon# show run | begin bgp
router bgp 898
  bgp log-neighbor-changes
  network 10.50.5.0 mask 255.255.255.0
  network 10.50.50.0 mask 255.255.255.0
  neighbor 10.1.8.2 remote-as 898
  neighbor 15.1.15.1 remote-as 898
  no auto-summary
```

```
Dixon# show ip bgp
```

```
BGP table version is 3, local router ID is 10.50.50.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i10.20.10.0/24	15.1.15.1	0	100	0	i
* i10.20.20.0/24	15.1.15.1	0	100	0	i
*> 10.50.5.0/24	0.0.0.0	0		32768	i
*> 10.50.50.0/24	0.0.0.0	0		32768	I
* i192.168.40.0	10.1.8.2	0	100	0	i
* i192.168.60.0	10.1.8.2	0	100	0	i

如果你将每个 BGP 对等体都配置为全网状连接，你会注意到每台路由器都收到了到它们的对等体的环回接口的路由。然而，没有一台路由器会将这些到环回接口的路由存为最佳（>）路由，这是因为环回接口路由没有和主 IP 路由表中的路由同步。为了确定路由是否同步，可以使用 **show ip bgp** 命令来查找显示为最优（>）的路由，BGP 只会将有效的路由存放在主路由表中，同时也只会将有效的（*）最优的（>）路由发送给对等连接的 BGP 发言人。

第 7 步 现在你能够看到同步在 I-BGP 对等体上起的作用，关闭 BGP 同步，重启对等体之间的 BGP 会话，然后再检查 BGP 表。范例 8-32 显示了 **no synchronization** 命令在路由器 Sydney 上起的作用。

范例 8-32 在路由器 Sydney 上关闭 BGP 同步

```
Sydney(config)# router bgp 898
Sydney(config-router)# no synchronization
Sydney# show ip bgp
BGP table version is 7, local router ID is 10.20.20.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.20.10.0/24	0.0.0.0	0		32768	i

（待续）

*> 10.20.20.0/24	0.0.0.0	0	32768	I
*>i10.50.5.0/24	10.1.9.2	0	100	0 i
*>i10.50.50.0/24	10.1.9.2	0	100	0 i
*>i192.168.40.0	10.1.8.2	0	100	0 i
*>i192.168.60.0	10.1.8.2	0	100	0 i

第 8 步 为了使 BGP 路由器能够 ping 它们的对等体的环回接口，需要配置 BGP 和 EIGRP 之间的重分发。为了实现这个目的，首先需要进入 BGP 配置模式使用 **bgp redistribute-internal** 命令来启用 BGP 到 IGP 的重分发，然后同样地在 EIGRP 进程中启用 BGP 重分发。当 EIGRP 再收敛后，你将在所有路由器的主路由表中看到环回网段的路由，你应该可以 ping 所有路由器上的所有地址。在路由器 Sydney、Vaughn 和 Dixon 上 EIGRP 外部路由将代替 BGP 路由，这时因为 EIGRP 外部路由比 BGP 路由有着更低的管理距离（external EIGRP 170，I-BGP 200）。范例 8-33 显示了路由器 Sydney 的最终配置和路由表。

范例 8-33 路由器 Sydney 的最终配置和路由表

```
hostname Sydney
!
interface Loopback10
 ip address 10.20.10.1 255.255.255.0
!
interface Loopback20
 ip address 10.20.20.1 255.255.255.0
!
interface Serial0
 ip address 15.1.15.1 255.255.255.0
!
router eigrp 101
 redistribute bgp 898 metric 56 200 255 1 1500
 network 15.1.15.0 0.0.0.255
 no auto-summary
!
router bgp 898
 no synchronization
 bgp redistribute-internal
 bgp log-neighbor-changes
 network 10.20.10.0 mask 255.255.255.0
 network 10.20.20.0 mask 255.255.255.0
 neighbor 10.1.8.2 remote-as 898
 neighbor 10.1.9.2 remote-as 898
 no auto-summary

Sydney# show ip route | begin Gateway
Gateway of last resort is not set
D EX 192.168.60.0/24 [170/2758656] via 15.1.15.2, 00:00:25, Serial0
D EX 192.168.40.0/24 [170/2758656] via 15.1.15.2, 00:00:25, Serial0
    10.0.0.0/24 is subnetted, 6 subnets
D      10.1.9.0 [90/2707456] via 15.1.15.2, 00:37:45, Serial0
D      10.1.8.0 [90/2707456] via 15.1.15.2, 00:38:26, Serial0
D EX   10.50.50.0 [170/2758656] via 15.1.15.2, 00:08:21, Serial0
```

(待续)

```

C      10.20.20.0 is directly connected, Loopback20
C      10.20.10.0 is directly connected, Loopback10
D EX   10.50.5.0 [170/2758656] via 15.1.15.2, 00:08:21, Serial0
       164.189.0.0/24 is subnetted, 1 subnets
D      164.189.26.0 [90/2195456] via 15.1.15.2, 00:39:36, Serial0
       15.0.0.0/24 is subnetted, 1 subnets
C      15.1.15.0 is directly connected, Serial0

Sydney# ping 10.50.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.50.5.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

Sydney# ping 192.168.40.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/41/44 ms

```

范例 8-34 显示了路由器 Sloane 的完整配置和路由表，范例 8-35 显示了路由器 Khasinau 的完整配置和路由表，范例 8-36 显示了路由器 McCullough 的同样的信息。

范例 8-34 路由器 Sloane 的完整配置和路由表

```

hostname Sloane
!
interface Ethernet0/0
 ip address 164.189.26.1 255.255.255.0
!
interface Serial0/0
 ip address 15.1.15.2 255.255.255.0
!
router eigrp 101
 network 15.1.15.0 0.0.0.255
 network 164.189.26.0 0.0.0.255
 no auto-summary

Sloane# show ip route | include vialis
Gateway of last resort is not set
D EX 192.168.60.0/24 [170/2246656] via 164.189.26.3, 00:16:58, Ethernet0/0
D EX 192.168.40.0/24 [170/2246656] via 164.189.26.3, 00:16:58, Ethernet0/0
   10.0.0.0/24 is subnetted, 6 subnets
D      10.1.9.0 [90/2195456] via 164.189.26.3, 00:54:18, Ethernet0/0
D      10.1.8.0 [90/2195456] via 164.189.26.2, 00:54:59, Ethernet0/0
D EX   10.50.50.0 [170/2246656] via 164.189.26.3, 00:24:54, Ethernet0/0
D EX   10.20.20.0 [170/46277376] via 15.1.15.1, 00:26:04, Serial0/0
D EX   10.20.10.0 [170/46277376] via 15.1.15.1, 00:26:04, Serial0/0
D EX   10.50.5.0 [170/2246656] via 164.189.26.3, 00:24:54, Ethernet0/0
       164.189.0.0/24 is subnetted, 1 subnets
C      164.189.26.0 is directly connected, Ethernet0/0
       15.0.0.0/24 is subnetted, 1 subnets
C      15.1.15.0 is directly connected, Serial0/0

```

范例 8-35 路由器 Khasinau 的完整配置和路由表

```

hostname Khasinau
!
interface FastEthernet0

```

(待续)


```

ip address 164.189.26.2 255.255.255.0
!
interface Serial1
ip address 10.1.8.1 255.255.255.0
encapsulation frame-relay IETF
clockrate 1300000
frame-relay map ip 10.1.8.2 104 broadcast
frame-relay lmi-type ansi
!
!
router eigrp 101
network 10.1.8.0 0.0.0.255

network 164.189.26.0 0.0.0.255
no auto-summary

Khasinaw# show ip route | include vialis
Gateway of last resort is not set
D EX 192.168.60.0/24 [170/2223616] via 164.189.26.3, 00:21:11, FastEthernet0
D EX 192.168.40.0/24 [170/2223616] via 164.189.26.3, 00:21:11, FastEthernet0
    10.0.0.0/24 is subnetted, 6 subnets
D      10.1.9.0 [90/2172416] via 164.189.26.3, 00:58:31, FastEthernet0
C      10.1.8.0 is directly connected, Serial1
D EX   10.50.50.0 [170/2223616] via 164.189.26.3, 00:29:07, FastEthernet0
D EX   10.20.20.0 [170/46279936] via 164.189.26.1, 00:30:17, FastEthernet0
D EX   10.20.10.0 [170/46279936] via 164.189.26.1, 00:30:17, FastEthernet0
D EX   10.50.5.0 [170/2223616] via 164.189.26.3, 00:29:07, FastEthernet0
    164.189.0.0/24 is subnetted, 1 subnets
C      164.189.26.0 is directly connected, FastEthernet0
    15.0.0.0/24 is subnetted, 1 subnets
D      15.1.15.0 [90/2172416] via 164.189.26.1, 00:59:15, FastEthernet0

```

范例 8-36 路由器 McCullough 的完整配置和路由表

```

hostname McCullough
!
interface Ethernet0
ip address 164.189.26.3 255.255.255.0
!
interface Serial1
ip address 10.1.9.1 255.255.255.0
encapsulation frame-relay IETF
clockrate 1300000
frame-relay map ip 10.1.9.2 105 broadcast
frame-relay lmi-type ansi
!
router eigrp 101
network 10.1.9.0 0.0.0.255
network 164.189.26.0 0.0.0.255
no auto-summary

McCullough# show ip route | include vialis
Gateway of last resort is not set
D EX 192.168.60.0/24 [170/2221056] via 10.1.9.2, 00:23:34, Serial1
D EX 192.168.40.0/24 [170/2221056] via 10.1.9.2, 00:23:34, Serial1
    10.0.0.0/24 is subnetted, 6 subnets
C      10.1.9.0 is directly connected, Serial1
D      10.1.8.0 [90/2172416] via 164.189.26.2, 01:00:59, Ethernet0
D EX   10.50.50.0 [170/2221056] via 10.1.9.2, 00:31:30, Serial1
D EX   10.20.20.0 [170/46279936] via 164.189.26.1, 00:32:40, Ethernet0
D EX   10.20.10.0 [170/46279936] via 164.189.26.1, 00:32:40, Ethernet0
D EX   10.50.5.0 [170/2221056] via 10.1.9.2, 00:31:30, Serial1

```

(待续)

```
164.189.0.0/24 is subnetted, 1 subnets
C    164.189.26.0 is directly connected, Ethernet0
15.0.0.0/24 is subnetted, 1 subnets
D    15.1.15.0 [90/2172416] via 164.189.26.1, 01:00:59, Ethernet0
```

范例 8-37 显示了路由器 Vaughn 的最终配置、BGP 表和路由表，范例 8-38 显示了路由器 Dixon 的同样的信息。

范例 8-37 路由器 Vaughn 的最终配置和路由表

```
hostname Vaughn
!
interface Loopback10
 ip address 192.168.40.1 255.255.255.0
!
interface Loopback20
 ip address 192.168.60.1 255.255.255.0
!
interface Serial1
 ip address 10.1.8.2 255.255.255.0
 encapsulation frame-relay IETF
 clockrate 1300000
 frame-relay map ip 10.1.8.1 401 broadcast
 frame-relay lmi-type ansi
!
router eigrp 101
 redistribute bgp 898 metric 1544 200 255 1 1500
 network 10.1.8.0 0.0.0.25
 no auto-summary
!
router bgp 898
 no synchronization
 bgp redistribute-internal
 network 192.168.40.0
 network 192.168.60.0
 neighbor 10.1.9.2 remote-as 898
 neighbor 15.1.15.1 remote-as 898

Vaughn# show ip bgp | begin Network
      Network      Next Hop      Metric LocPrf Weight Path
*>i10.20.10.0/24    15.1.15.1          0    100      0 i
*>i10.20.20.0/24    15.1.15.1          0    100      0 i
*>i10.50.5.0/24     10.1.9.2           0    100      0 i
*>i10.50.50.0/24    10.1.9.2           0    100      0 i
*> 192.168.40.0     0.0.0.0            0           32768 i
*> 192.168.60.0     0.0.0.0            0           32768 i

Vaughn# show ip route | include vialis
Gateway of last resort is not set
C    192.168.60.0/24 is directly connected, Loopback20
C    192.168.40.0/24 is directly connected, Loopback10
    10.0.0.0/24 is subnetted, 6 subnets
D    10.1.9.0 [90/2684416] via 10.1.8.1, 01:05:52, Serial1
C    10.1.8.0 is directly connected, Serial1
D EX  10.20.20.0 [170/46791936] via 10.1.8.1, 00:39:46, Serial1
D EX  10.50.50.0 [170/2735616] via 10.1.8.1, 00:38:36, Serial1
D EX  10.20.10.0 [170/46791936] via 10.1.8.1, 00:39:46, Serial1
D EX  10.50.5.0 [170/2735616] via 10.1.8.1, 00:38:36, Serial1
    164.189.0.0/24 is subnetted, 1 subnets
D    164.189.26.0 [90/2172416] via 10.1.8.1, 01:05:52, Serial1
    15.0.0.0/24 is subnetted, 1 subnets
D    15.1.15.0 [90/2684416] via 10.1.8.1, 01:05:53, Serial1
```

范例 8-38 路由器 Dixon 的最终配置和路由表

```

hostname Dixon
!
interface Loopback10
 ip address 10.50.5.1 255.255.255.0
!
interface Loopback20
 ip address 10.50.50.1 255.255.255.0
!
interface Serial1
 ip address 10.1.9.2 255.255.255.0
 encapsulation frame-relay IETF
 frame-relay map ip 10.1.9.1 501 broadcast
 frame-relay lmi-type ansi
!
router eigrp 101
 redistribute bgp 898 metric 1544 200 255 1 1500
 network 10.1.9.0 0.0.0.255
 no auto-summary
!
router bgp 898
 no synchronization
 bgp redistribute-internal
 bgp log-neighbor-changes
 network 10.50.5.0 mask 255.255.255.0
 network 10.50.50.0 mask 255.255.255.0
 neighbor 10.1.8.2 remote-as 898
 neighbor 15.1.15.1 remote-as 898
Dixon# show ip bgp | begin Network
      Network                Next Hop           Metric LocPrf Weight Path
*>i10.20.10.0/24             15.1.15.1           0      100      0 i
*>i10.20.20.0/24             15.1.15.1           0      100      0 i
*> 10.50.5.0/24              0.0.0.0             0           32768 i
*> 10.50.50.0/24             0.0.0.0             0           32768 i
*>i192.168.40.0              10.1.8.2            0      100      0 i
*>i192.168.60.0             10.1.8.2            0      100      0 i

Dixon# show ip route | include vialis
Gateway of last resort is not set
B    192.168.60.0/24 [200/0] via 10.1.8.2, 00:33:41
B    192.168.40.0/24 [200/0] via 10.1.8.2, 00:33:41
    10.0.0.0/24 is subnetted, 6 subnets
C      10.1.9.0 is directly connected, Serial1
D      10.1.8.0 [90/2684416] via 10.1.9.1, 01:08:24, Serial1
D EX   10.20.20.0 [170/46791936] via 10.1.9.1, 00:42:47, Serial1
C      10.50.50.0 is directly connected, Loopback20
D EX   10.20.10.0 [170/46791936] via 10.1.9.1, 00:42:47, Serial1
C      10.50.5.0 is directly connected, Loopback10
    164.189.0.0/24 is subnetted, 1 subnets
D      164.189.26.0 [90/2172416] via 10.1.9.1, 01:08:24, Serial1
    15.0.0.0/24 is subnetted, 1 subnets
D      15.1.15.0 [90/2684416] via 10.1.9.1, 01:08:24, Serial1
    
```

注意：在 BGP 和 IGP 之间重分发路由，反之亦然，将对路由性能产生严重的影响，在生产网络中使用 BGP/IGP 重分发要小心。

三、I-BGP next-hop self 命令

在多归路的 BGP 网络中经常碰到的一个问题就是不可达的 BGP 路由，当刚刚配置了子书仅限试看之用，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

E-BGP 到 I-BGP 的关系而且下游的 I-BGP 发言人无法到达与发送 E-BGP 更新的路由器直接对等连接的路由器通告的下一跳地址时，通常会发生这个问题。尽管与上游的 E-BGP 对等体直接对等连接的路由器能够到达它的 E-BGP 对等体的地址，但是其他在它下游的路由器没有到 E-BGP 对等体的路由，所以这些路由器不能到达 BGP 更新中通告的下一跳地址。这是符合设计的行为，它发生的原因是 I-BGP 发言人在转发路由给其他 I-BGP 对等体的时候没有修改下一跳属性。图 8-6 显示从上游路由器 Chunk 和 Sloth 发送的路由通过路由器 Mikey 到达 Data 和 Brand 时下一跳属性没有被修改。

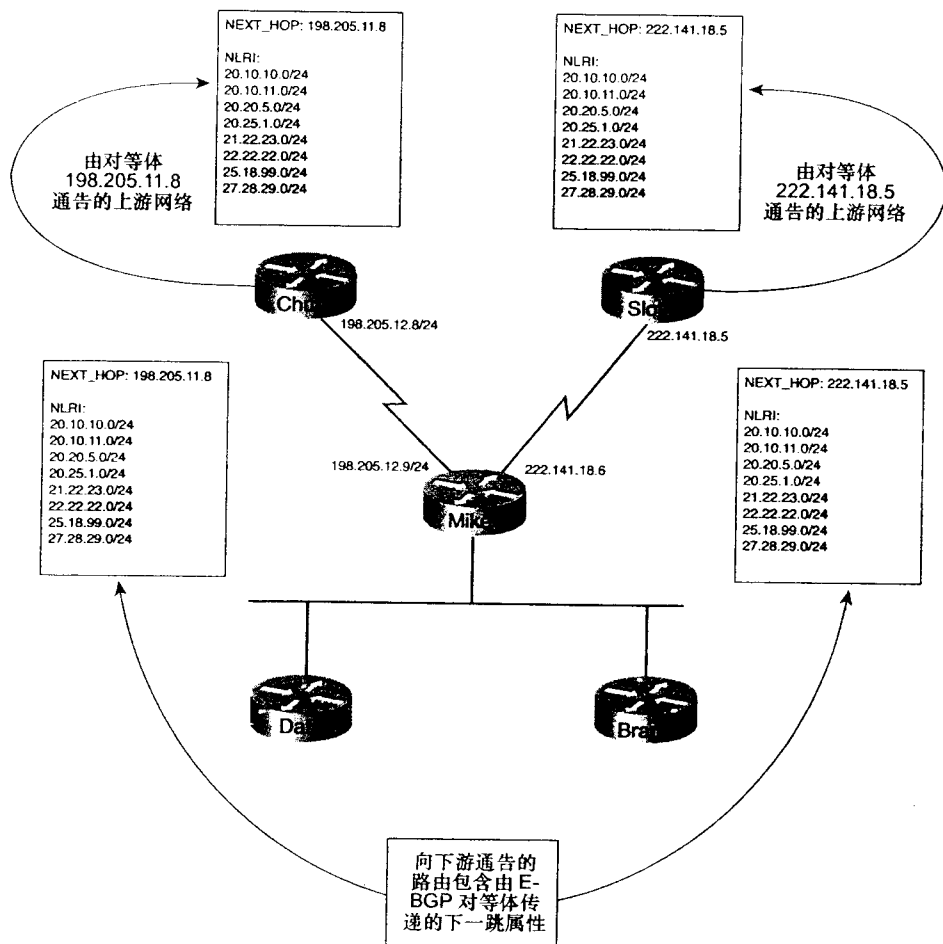


图 8-6 I-BGP 下一跳行为

只需要完成三步就可以在一个 I-BGP 路由器上将下一跳属性的值修改为本地路由器的地址。

第 1 步 启用 BGP 路由。

```
Mikey(config)# router bgp 10101
```

第 2 步 配置 BGP 邻居关系。

```
Mikey(config-router)# neighbor 198.205.12.8 remote-as 811      -- E-BGP peer
Mikey(config-router)# neighbor 222.141.18.5 remote-as 945      -- E-BGP peer
Mikey(config-router)# neighbor 192.168.1.2 remote-as 10101     -- I-BGP peer
Mikey(config-router)# neighbor 192.168.1.3 remote-as 10101     -- I-BGP peer
```

第 3 步 使用 **neighbor ip-address next-hop-self** 命令修改下一跳属性。

```
Mikey(config-router)# neighbor 192.168.1.2 next-hop-self      -- Change attribute
Mikey(config-router)# neighbor 192.168.1.3 next-hop-self      -- Change attribute
```

对下一跳属性的修改通过 **show ip bgp** 命令可以看到。范例 8-39 显示了当 **next-hop-self** 命令在路由器 Mikey 上使用之前路由器 Data 上下一跳属性是怎样的，范例 8-40 显示了在路由器 Mikey 上加上 **next-hop-self** 配置后在同样的路由器上使用同样的命令的输出。

范例 8-39 在修改下一跳属性之前

```
Data# show ip bgp | begin Network
Network      Next Hop      Metric LocPrf Weight Path
*> 2.0.0.0    157.68.90.1    0      100      0 3456 i
*> 3.0.0.0    157.68.90.1    0      100      0 3456 i
```

范例 8-40 使用 **next-hop-self** 命令之后

```
Data# show ip bgp | begin Network
Network      Next Hop      Metric LocPrf Weight Path
*>i2.0.0.0    192.168.1.1    0      100      0 3456 i
*>i3.0.0.0    192.168.1.1    0      100      0 3456 i
```

四、实际范例：I-BGP 下一跳地址处理

本范例显示了一个自治系统中 **next-hop-self** 命令对 I-BGP 路由的影响。本例需要 5 台思科路由器，接口配置如表 8-13。

表 8-13

路由器接口需求

路由器	以太、快速以太或令牌环接口	串行接口	路由器	以太、快速以太或令牌环接口	串行接口
Skinner	0	1	Byers	1	0
Kritchgau	0	1	Frohike	1	0
Langle	1	2			

在配置任何路由器之前，确认路由器已经接好线，如图 8-7 所示。本例需要两个背对背的串行电缆和 3 个连接到集线器、交换机或是 MSAU 的以太网线。如果你使用交换机，所有的接口都要放在同一个虚拟局域网中。

第 1 步 如图 8-7 所示配置所有的 IP 地址，在进入第 2 步之前验证所有的接口都是起来的。在自治系统 123 内所有的 I-BGP 发言人上配置 OSPF，将这些路由器的所有接口放入区域 0，不要在路由器 Skinner 和 Kritchgau 上配置 OSPF。范例 8-41 显示了路由器 Skinner、Langle、Byers 和 Frohike 上的 IP 地址和 OSPF 配置。

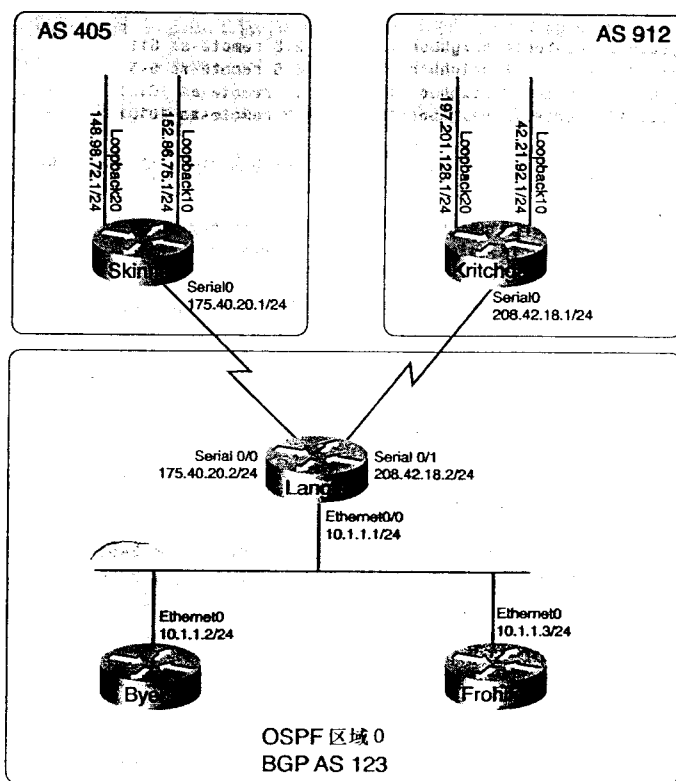


图 8-7 Conspiracy 网络图

范例 8-41 第 1 步在路由器 Skinner、Langle 和 Byers 上的配置

```

Skinner# show run | begin Loopback
interface Loopback10
 ip address 152.86.75.1 255.255.255.0
!
interface Loopback20
 ip address 148.98.72.1 255.255.255.0
!
interface Serial0
 ip address 175.40.20.1 255.255.255.0

```

```

Kritchgau# show run | begin Loopback
interface Loopback10
 ip address 42.21.92.1 255.255.255.0
!
interface Loopback20
 ip address 197.201.128.1 255.255.255.0
!
interface Serial0
 ip address 208.42.18.1 255.255.255.0

```

```

Langle# show run | begin Ethernet
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
!
interface Serial0/0
 ip address 175.40.20.2 255.255.255.0

```

(待续)

```
!
interface Serial0/1
 ip address 208.42.18.2 255.255.255.0
 clock rate 1300000
!
```

```
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
```

```
Byers# show run | begin Ethernet
interface Ethernet0
 ip address 10.1.1.2 255.255.255.0
!
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
```

```
Frohike# show run | begin Ethernet
interface Ethernet0
 ip address 10.1.1.3 255.255.255.0
!
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
```

第 2 步 在路由器 Skinner 和 Langle 以及路由器 Kritchgau 和 Langle 之间配置 E-BGP 会话。配置路由器 Skinner 和 Kritchgau 通过 BGP 通告它们的环回地址，让路由器 Langle 通告 10.1.1.0/24 网段给它的两个 E-BGP 对等体。在进入第 3 步之前，确认路由器 Langle 能够 ping 到路由器 Skinner 和 Kritchgau 的环回接口的所有 IP 地址。范例 8-42 显示了每个 BGP 路由器的 BGP 配置以及路由器 Langle 的路由表。

范例 8-42 路由器 Skinner、Kritchgau 和 Langle 的 BGP 配置

```
Skinner# show run | begin bgp
router bgp 405
 bgp log-neighbor-changes
 network 148.98.72.0 mask 255.255.255.0
 network 152.86.75.0 mask 255.255.255.0
 neighbor 175.40.20.2 remote-as 123
 no auto-summary
```

```
Kritchgau# show run | begin bgp
router bgp 912
 bgp log-neighbor-changes
 network 42.21.92.0 mask 255.255.255.0
 network 197.201.128.0 mask 255.255.255.0
 neighbor 208.42.18.2 remote-as 123
 no auto-summary
```

```
Langle# show run | begin bgp
router bgp 123
 bgp log-neighbor-changes
 network 10.1.1.0 mask 255.255.255.0
 neighbor 175.40.20.1 remote-as 405
 neighbor 208.42.18.1 remote-as 912
 no auto-summary
Langle# show ip route | begin Gateway
Gateway of last resort is not set
 1.0.0.0/32 is subnetted, 1 subnets
 C       1.1.1.1 is directly connected, Loopback0
```

(待续)

```
B 197.201.128.0/24 [20/0] via 208.42.18.1, 00:01:54
152.86.0.0/24 is subnetted, 1 subnets
B 152.86.75.0 [20/0] via 175.40.20.1, 00:05:21
175.40.0.0/24 is subnetted, 1 subnets
C 175.40.20.0 is directly connected, Serial0/0
42.0.0.0/24 is subnetted, 1 subnets
B 42.21.92.0 [20/0] via 208.42.18.1, 00:01:54
10.0.0.0/24 is subnetted, 1 subnets
C 10.1.1.0 is directly connected, Ethernet0/0
148.98.0.0/24 is subnetted, 1 subnets
B 148.98.72.0 [20/0] via 175.40.20.1, 00:05:22
C 208.42.18.0/24 is directly connected, Serial0/1
```

第 3 步 在路由器 Langle、Byers 和 Frohike 之间配置 I-BGP 连接。在进入下个步骤之前，验证 Byers 和 Frohike 从路由器 Skinner 和 Kritchgau 收到了 E-BGP 路由。范例 8-43 显示了路由器 Langle 的 BGP 配置和 BGP 表，范例 8-44 显示了路由器 Byers 的同样的信息，范例 8-45 显示了路由器 Frohike 的配置和 BGP 数据。

范例 8-43 路由器 Langle 的 BGP 配置和 BGP 表

```
Langle# show run | begin bgp
router bgp 123
  bgp log-neighbor-changes
  network 10.1.1.0 mask 255.255.255.0
  neighbor 10.1.1.2 remote-as 123
  neighbor 10.1.1.3 remote-as 123
  neighbor 175.40.20.1 remote-as 405
  neighbor 208.42.18.1 remote-as 912

Langle# show ip bgp | begin Network
Network        Next Hop        Metric LocPrf Weight Path
*> 10.1.1.0/24  0.0.0.0          0           32768 i
*> 42.21.92.0/24 208.42.18.1      0           0 912 i
*> 148.98.72.0/24 175.40.20.1      0           0 405 i
*> 152.86.75.0/24 175.40.20.1      0           0 405 i
*> 197.201.128.0 208.42.18.1      0           0 912 i
```

范例 8-44 路由器 Byers 的 BGP 配置和 BGP 表

```
Byers# show run | begin bgp
router bgp 123
  bgp log-neighbor-changes
  neighbor 10.1.1.1 remote-as 123
  neighbor 10.1.1.3 remote-as 123
Byers# show ip bgp | begin Network
Network        Next Hop        Metric LocPrf Weight Path
*>i10.1.1.0/24  10.1.1.1          0    100      0 i
* i42.21.92.0/24 208.42.18.1      0    100      0 912 i
* i148.98.72.0/24 175.40.20.1      0    100      0 405 i
* i152.86.75.0/24 175.40.20.1      0    100      0 405 i
* i197.201.128.0 208.42.18.1      0    100      0 912 i
```

范例 8-45 路由器 Frohike 的 BGP 配置和 BGP 表

```
Frohike# show run | begin bgp
router bgp 123
  bgp log-neighbor-changes
```

(待续)


```

neighbor 10.1.1.1 remote-as 123
neighbor 10.1.1.3 remote-as 123
Frohike# show ip bgp | begin Network

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>10.1.1.0/24	10.1.1.1	0	100	0	i
* 142.21.92.0/24	208.42.18.1	0	100	0	912 i
* 1148.98.72.0/24	175.40.20.1	0	100	0	405 i
* 1152.86.75.0/24	175.40.20.1	0	100	0	405 i
* 1197.201.128.0	208.42.18.1	0	100	0	912 i

第4步 在路由器 Langle、Byers 和 Frohike 上配置了 BGP 后，你可能注意到路由器 Byers 和 Frohike 从它的上游 E-BGP 对等体路由器 Langle 收到的路由没有被加入到路由表中，这是因为路由器 Langle 通告的下一跳地址使用的 IP 地址不可达。为了解决这个问题，在路由器 Langle 的每个 I-BGP 会话上使用 **next-hop-self** 命令，然后使用 **clear ip bgp *** 命令重启 BGP 会话。当 BGP 会话重新建立后路由器 Langle 通告从它的上游路由器来的路由时，将修改所有发送给 Byers 和 Frohike 路由器的路由的下一跳属性。范例 8-46 显示了加入 **next-hop-self** 命令后路由器 Langle 的配置，范例 8-47 显示了路由器 Byers 和 Frohike 上随之产生的 BGP 和 IP 路由表。

范例 8-46 路由器 Langle 的 BGP 配置

```

Langle# show run | begin bgp
router bgp 123
no synchronization
bgp router-id 177.164.8.5
bgp log-neighbor-changes
network 10.1.1.0 mask 255.255.255.0
neighbor 10.1.1.2 remote-as 123
neighbor 10.1.1.2 next-hop-self
neighbor 10.1.1.3 remote-as 123
neighbor 10.1.1.3 next-hop-self
neighbor 175.40.20.1 remote-as 405
neighbor 208.42.18.1 remote-as 912
no auto-summary

```

范例 8-47 随之产生的 BGP 和 IP 路由表

```

Byers# show ip bgp
BGP table version is 6, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>10.1.1.0/24	10.1.1.1	0	100	0	I
*>142.21.92.0/24	10.1.1.1	0	100	0	912 i
*>1148.98.72.0/24	10.1.1.1	0	100	0	405 i
*>1152.86.75.0/24	10.1.1.1	0	100	0	405 i
*>1197.201.128.0	10.1.1.1	0	100	0	912 i

```

Byers# show ip route | begin Gateway
Gateway of last resort is not set
B 197.201.128.0/24 [200/0] via 10.1.1.1, 00:01:09
152.86.0.0/24 is subnetted, 1 subnets
B 152.86.75.0 [200/0] via 10.1.1.1, 00:01:09
42.0.0.0/24 is subnetted, 1 subnets

```

(待续)

```
B    42.21.92.0 [200/0] via 10.1.1.1, 00:01:09
    10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Ethernet0
    148.98.0.0/24 is subnetted, 1 subnets
B    148.98.72.0 [200/0] via 10.1.1.1, 00:01:09
Byers# ping 197.201.128.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 197.201.128.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
Byers# ping 152.86.75.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 152.86.75.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/40 ms

Frohike# show ip bgp | begin Network
      Network          Next Hop           Metric LocPrf Weight Path
*>i10.1.1.0/24         10.1.1.1             0      100      0 i
*>i42.21.92.0/24       10.1.1.1             0      100      0 912 i
*>i148.98.72.0/24      10.1.1.1             0      100      0 405 i
*>i152.86.75.0/24      10.1.1.1             0      100      0 405 i
*>i197.201.128.0       10.1.1.1             0      100      0 912 i
Frohike# show ip route | begin Gateway
Gateway of last resort is not set
B    197.201.128.0/24 [200/0] via 10.1.1.1, 00:02:24
    152.86.0.0/24 is subnetted, 1 subnets
B    152.86.75.0 [200/0] via 10.1.1.1, 00:02:24
    42.0.0.0/24 is subnetted, 1 subnets
B    42.21.92.0 [200/0] via 10.1.1.1, 00:02:24
    10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Ethernet0
    148.98.0.0/24 is subnetted, 1 subnets
B    148.98.72.0 [200/0] via 10.1.1.1, 00:02:24
Frohike# ping 42.21.92.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 42.21.92.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
Frohike# ping 152.86.75.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 152.86.75.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/40 ms
```

现在你能够看到使用 I-BGP 全网状配置、BGP 同步和 `next-hop-self` 命令的影响，下面我们来看看 E-BGP 对等体配置和在进行 E-BGP 配置时将碰到的问题。

8.4 E-BGP 对等关系

不容置疑，E-BGP 对等关系是企业网络专家碰到的最常用的 BGP 对等关系。不管一个 BGP 发言人有多少个对等体，在 E-BGP 对等体之间只会有几种连接类型。

- **直连对等体**——直接连接的对等体，通常经过在客户和服务提供商之间的广域网连接或是在穿越对等体之间。

- 非直连对等体——必须通过一个或是多个非 BGP 发言人的路由器来互相访问的 E-BGP 对等体。

配置直连的 E-BGP 连接是极其直接的过程，仅仅需要以下三步：

第 1 步 使用命令 **router bgp as-number** 启用 BGP 路由。

第 2 步 使用命令 **neighbor ip-address remote-as remote-as-number** 配置 BGP 对等体。如果在配置 **neighbor** 命令时输入的自治系统号码和本地配置的自治系统号码不一样，就会建立 E-BGP 对等关系。

第 3 步 （可选）指定本地对等体使用命令 **network network [mask subnet-mask]** 通告的网段。和 EIGRP **network** 命令类似，BGP **network** 命令定义了本地对等体将要通告的网段，如果这些网段没有严格地落在分类的边界上，子网掩码将定义这些网段。

图 8-8 显示了在路由器 Sideshow 和 Crusty 之间直接连接的 E-BGP 配置范例。

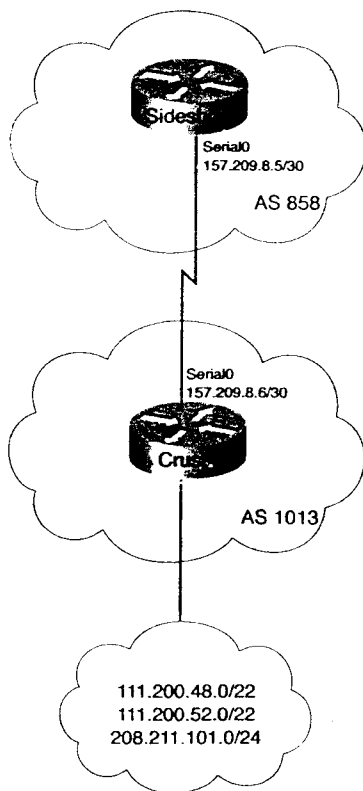


图 8-8 直连的 E-BGP 对等体

在这个范例中，路由器 Sideshow 和 Crusty 通过 157.209.8.4/30 网段的串行连接建立 E-BGP 对等连接。路由器 Sideshow 在自治系统 858 中，它没有通告 BGP 网段；路由器 Crusty 属于自治系统 1013，并且通告网段 111.200.48.0/22、111.200.52.0/22 和 208.211.101.0/24。范例 8-48 显示了路由器 Sideshow 的配置和它看到的路由，范例 8-49 显示了路由器 Crusty 的配置。

范例 8-48 路由器 Sideshow 的配置

```
Sideshow# show run | begin bgp
router bgp 858
  bgp log-neighbor-changes
  neighbor 157.209.8.6 remote-as 1013
  no auto-summary

Sideshow# show ip bgp
BGP table version is 8, local router ID is 157.209.8.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
*> 111.200.48.0/22   157.209.8.6             0             0 1013 i
*> 111.200.52.0/22   157.209.8.6             0             0 1013 i
*> 208.211.101.0     157.209.8.6             0             0 1013 i
```

范例 8-49 路由器 Crusty 的配置

```
Crusty# show run | begin bgp
router bgp 1013
  bgp log-neighbor-changes
  network 111.200.48.0 mask 255.255.252.0
  network 111.200.52.0 mask 255.255.252.0
  network 208.211.101.0
  neighbor 157.209.8.5 remote-as 858
  no auto-summary
```

使用 E-BGP Multihop 超越 BGP 的限制

因为 BGP-4 的规范不允许 E-BGP 发言人在非直连的时候建立对等关系，必须规划非直接连接的外部 BGP 的配置。你需要知道当对等体必须通过其他路由器才能建立对等关系和交换更新报文的时候，是否需要特别的设计考虑来保证 BGP 正常工作。

neighbor ip-address ebgp-multihop 命令表明 **neighbor** 语句指定的远端对等体没有直接相连，当 E-BGP 发言人必须穿过一个或是多个节点才能建立成功的 E-BGP 会话时就使用这个命令。**neighbor ip-address ebgp-multihop** 命令使用以下的语法：

```
neighbor ip-address ebgp-multihop [ number-of-hops ] .
```

需要指明到达一个邻居必须穿过的跳数（范围从 1~255），如果你不清楚需要的跳数，可以使用默认值，但是由于默认值可能会允许长路径之间的非最优路由，所以不是总推荐使用默认值。

在对等体之间建立成功的 E-BGP 对等关系，必须完成 5 个步骤：

- 第 1 步 在配置 BGP 之前使用命令 **show ip route neighbor-ip-address** 验证本地和远端的路由器都有路由可以相互到达。
- 第 2 步 使用 **router bgp as-number** 命令启用本地 BGP 进程。
- 第 3 步 使用命令 **neighbor ip-address remote-as remote-as-number** 配置远端对等体的 IP 地址和自治系统号码。
- 第 4 步 使用 **network** 命令配置本地对等体将要通告的网段。

第 5 步 使用 **neighbor ip-address ebgp-multihop number-of-hops** 命令启用 E-BGP multihop。

在图 8-9 所示的网络中，注意路由器 Murtagh 和 Geilis 通过路由器 Willoughby 相连。路由器 Murtagh 属于自治系统 1743，路由器 Geilis 属于自治系统 1968，它们必须通过没有运行 BGP 的路由器 Willoughby 发送 BGP 报文来建立 E-BGP 对等关系。

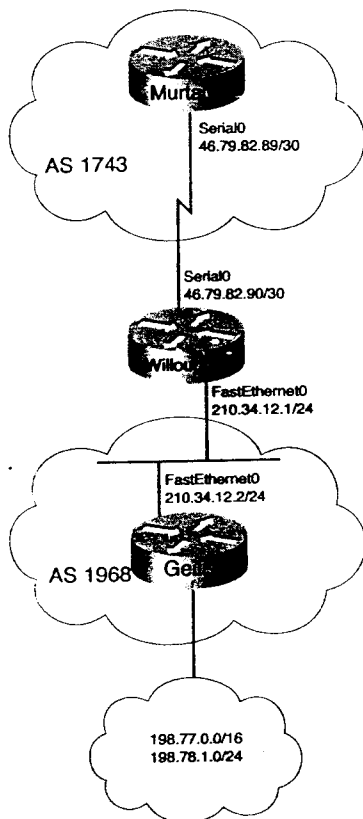


图 8-9 在多跳之间运行 E-BGP

下面显示了如何使用 **ebgp-multihop** 关键字在路由器 Murtagh 和 Geilis 之间启用 BGP 路由。注意在范例 8-50 中使用了命令 **neighbor 210.34.12.2 ebgp-multihop 2** 来说明到路由器 Geilis 不超过两跳，同时使用了一个静态路由告诉路由器 Murtagh 如何到达路由器 Geilis 所在的 210.32.12.0/24 网段。

范例 8-50 路由器 Murtagh 的配置

```

Murtagh# show run | begin bgp
router bgp 1743
  bgp log-neighbor-changes
  neighbor 210.34.12.2 remote-as 1968
  neighbor 210.34.12.2 ebgp-multihop 2
!
ip classless
ip route 210.34.12.0 255.255.255.0 46.79.82.90

```

为了验证 E-BGP multihop 的配置正在正常工作，使用 **show ip bgp neighbor** 命令（或是 **show ip bgp neighbors | i external|state|hops** 看 E-BGP 邻居的简化总结），并且寻找已建立的连接。范例 8-51 显示了路由器 Murtagh 上命令 **show ip bgp neighbors** 和 **show ip bgp neighbors | i external|state|hops** 的输出。

范例 8-51 命令 **show ip bgp neighbors** 的输出

```
Murtagh# show ip bgp neighbors
BGP neighbor is 210.34.12.2, remote AS 1968, external link

  BGP version 4, remote router ID 198.78.1.1
  BGP state = Established, up for 00:16:08
  Last read 00:00:08, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Received 25 messages, 0 notifications, 0 in queue
  Sent 25 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5
  Index 1, Offset 0, Mask 0x2
  2 accepted prefixes consume 72 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0
  Number of NLRI in the update sent: max 0, min 0
  Connections established 2; dropped 1
  Last reset 00:16:53, due to Peer closed the session
  External BGP neighbor might be up to 2 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 46.79.82.89, Local port: 179
Foreign host: 210.34.12.2, Foreign port: 11020
Byers# show ip bgp neighbors | i external|state|hops
BGP neighbor 210.34.12.2, remote AS 1968, external link
  BGP state = Established, up for 00:16:08
  External BGP neighbor might be up to 2 hops away.
```

如果没有为每个非直连的 E-BGP 会话使用 **ebgp-multihop** 关键词，**show ip bgp neighbors** 命令将显示关于出现的问题的很多提示，如范例 8-52 所示。

范例 8-52 诊断非直连的 E-BGP 对等连接问题

```
Murtagh# show ip bgp neighbors
BGP neighbor is 210.34.12.2, remote AS 1968, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:00:09, hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
  Sent 0 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0
  Index 1, Offset 0, Mask 0x2
  0 accepted prefixes consume 0 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0
  Number of NLRI in the update sent: max 0, min 0
  Connections established 0; dropped 0
```

（待续）

```

Last reset never
External BGP neighbor not directly connected.
No active TCP connection

```

例如，第一个高亮行说明本地 BGP 发言人不知道远端对等体的 BGP 路由器识别符，也就是说本地路由器从来没有看到过远端对等体的 BGP 路由器识别符。而且，注意 BGP 会话处于空闲状态，这通常说明在对等体之间建立 TCP 会话的时候存在问题。没有建立或是结束会话以及发送和接收的 BGP 报文为 0 表明没有发送 BGP 报文给远端对等体，而且没有从远端对等体收到任何 BGP 报文。问题的原因明白地显示在“External BGP neighbor not directly connected.”这行上，除此之外，命令输出的最后一行明白地显示了在对等体之间没有活动的 TCP 连接。如果你在连接 E-BGP 对等体的时候发生了问题，应该总是使用 **show ip bgp neighbors** 命令来帮助诊断错误情况。范例 8-53 列出了路由器 Willoughby 和 Geilis 的配置。

范例 8-53 路由器 Willoughby 和 Geilis 的配置

```

hostname Willoughby
!
interface Serial0
 ip address 46.79.82.90 255.255.255.252
!
interface FastEthernet0
 ip address 210.34.12.1 255.255.255.0
!
router ospf 1
 network 46.79.82.88 0.0.0.3 area 0
 network 210.34.12.0 0.0.0.255 area 0

hostname Geilis
!
interface Loopback10
 ip address 198.77.1.1 255.255.0.0
!
interface Loopback20
 ip address 198.78.1.1 255.255.255.0
!
interface FastEthernet0
 ip address 210.34.12.2 255.255.255.0
!
router ospf 1
 network 210.34.12.0 0.0.0.255 area 0
!
router bgp 1968
 bgp log-neighbor-changes
 network 198.77.0.0 mask 255.255.0.0
 network 198.78.1.0 mask 255.255.255.0
 neighbor 46.79.82.89 remote-as 1743
 neighbor 46.79.82.89 ebgp-multihop 2
 no auto-summary

```

现在你对如何配置 BGP 和故障排查 BGP 连接问题有了实际的了解，下面学习 BGP 是如何与其他路由协议互相作用的，BGP 是如何在表中存放路由的，BGP 是如何被配置来通告本地网段的。

8.5 BGP 和 IGP 的相互作用

当使用 BGP 作为自治系统的路由协议的时候，始终应该牢记的一件事就是 BGP 是路径-向量路由协议，它和距离-向量协议以及包括 OSPF 和 EIGRP 在内的链路状态协议不同。BGP 不是基于跳数、开销或是其他 IGP 协议的度量值来路由数据包的，它是基于自治系统路径来路由数据包的。记住 BGP 和 IGP 协议的不同之处将会大大节省你故障排查问题的时间。

当和其他 IGP 协议一起使用 BGP 的时候记住下面这些规则：

- BGP 不会将无法验证可达性的路由放入主 IP 路由表中。
- 路由器为了能够成功地使用 BGP 路由，必须在主 IP 路由表中有到达下一跳地址的路由。
- 除非特别配置，BGP 只会将到目的网段的最佳路径放入主 IP 路由表中，但是你可以使用在第 9 章介绍的 BGP **maximum-paths** 命令来配置多于一条的路径。
- BGP 只通告到目的网段的最佳路径，可以通过 BGP 属性来控制 BGP 路径的选择，你可以使用第 9 章介绍的一些特定的思科 IOS 软件 BGP 配置命令来控制最佳路径选择的过程。
- BGP 使用自己的最佳路径决策程序找到最有效的路由并且存入主路由表中。
- BGP 只会和显式配置的对等体建立对等关系，也只会通告被显式配置要通告的网段。
- 除非被显式配置，否则 BGP 不会向 IGP 重分发路由。
- BGP 是一个极其容易客户化的协议，它可以被配置成动态化或是静态的，可以使用很多不同的方法通告和控制路由策略。

将 BGP 作为一个路由协议

可以有很多不同的方式来使用 BGP 补充你现有的 IGP 协议。设计 BGP 网络最简单的方法是首先分析你的 IP 地址规划，验证你创建的网络设计允许路由聚合和路由表的保持。例如，假设你负责设计一个全国性的企业网，你得到了一个可以使用在网络中/22 的公共 IP 地址块，这时你必须决定将主要的数据中心放在哪里，如何分配地址来利用路由协议。在这个过程中，必须创建策略来指定需要过滤哪些路由，如何进行路由聚合和汇总，如何通告这些路由（给内部对等体、外部对等体以及因特网）。

假设你的公司已经决定建立与两个提供因特网路由的服务提供商对等连接的 4 个主要的数据中心，同时你将使用 OSPF 作为内部路由协议。你得到的 IP 地址块是 109.248.4.0/22，自治系统号码是 444，你能将 IP 地址分为 4 个/24 的网段，每个数据中心使用一个。表 8-14 显示了如何将/22 网段分为 4 个/24 网段，同时散布全国到 LosAngeles、Dallas、Chicago 和 Boston 站点。

为了给新网络提供分层路由，你需要在每个数据中心的每个因特网边界路由器会聚这些

表 8-14 一个全国性的企业网的地址分配

洛杉矶	达拉斯	芝加哥	波士顿
109.248.4.0/24	109.248.5.0/24	109.248.6.0/24	109.248.7.0/24

地址 并且向每个服务提供商通告。为了提供运营商的冗余，每个数据中心至少需要两个 E-BGP 连接，同时为了建立 I-BGP 的全网状连接，需要在自治系统内的每个因特网边界路由器之间建立 I-BGP 连接。为了给你的公司提供一个成功的设计，需要确认 OSPF 配置为发送更新给 BGP 路由器，每个因特网边界路由器都有通过 OSPF 路由进程学到的路由信息。你必须这样做，只有这样，当某个因特网边界路由器不可用后，其他 3 台路由器能够继续成功地向因特网通告你的网段。图 8-10 显示了一个高级范例，你可以看到本例中的自治系统边界路由器是如何处理数据中心提供路由的每个州的路由的。

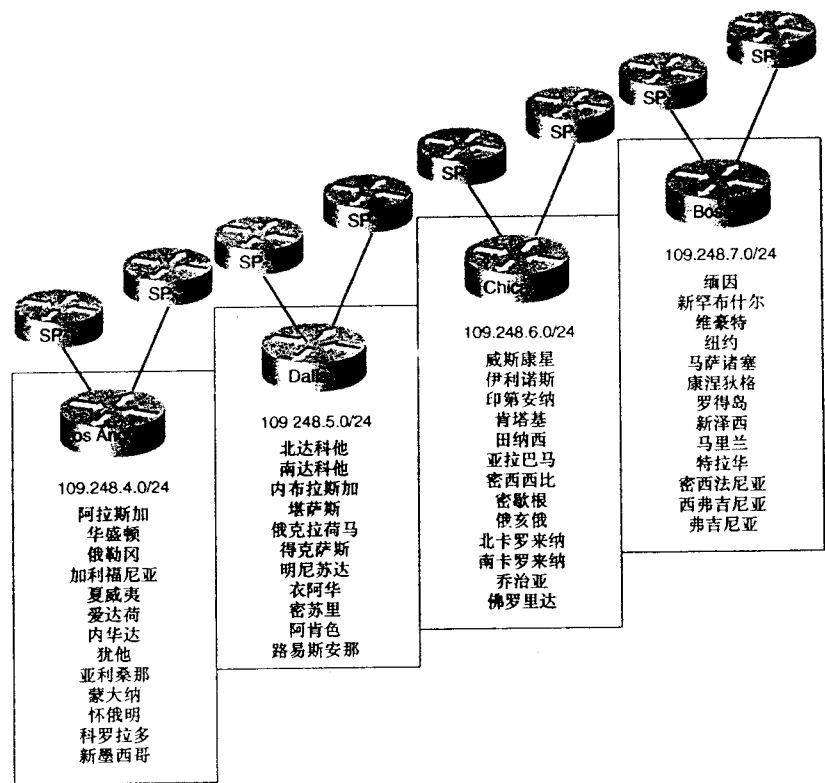


图 8-10 有 4 个数据中心的全国网的自治系统边界路由器的安排

在这个范例中, OSPF 使你可以通告和汇总较小的站点使用的/27 或是/28 网段。使用 OSPF ASBR 路由器将这些较小的网段会聚为/24 的地址块, 通过防火墙发送到因特网边界路由器上, 以供将来通告给因特网。

现在你已经知道如何在一个真实的网络中使用 BGP, 让我们看看 BGP 是如何利用 IP 路由表来存储和通告它的路由的, IGP 是如何学习到 BGP 路由的, 以及如何配置 BGP 来通告不同的网络类型。

8.6 BGP 和 IP 路由表

在第7章你看到了一个关于 BGP 如何使用它的表来存储通告路由的简单介绍，也了解到 BGP 更新用来转发流量的主 IP 路由表的过程，现在你将看到如何配置 BGP 来执行上述功能以及需要做什么来控制路由策略。

BGP 如何存储路由

在 BGP 通告路由给对等体之前，它总是检查路由的有效性。因此，如果路由是本地始发的，BGP 将检查路由在本地主 IP 路由表中是否存在；如果路由是从对等路由器接收到的，它会验证是否能够到达路由的下一跳地址。如果这两种情况都不满足，那么路由器只会将路由存放在自己的 BGP 路由表中，你通过命令 `show ip bgp` 可以看到，路由器不会将这些路由放入主 IP 路由表中或是通告给其他任何对等体。

注意：在进行 BGP 故障排查之前总是要先检查你的输入。思科 IOS 软件允许你使用 `network` 命令输入任何有效的 IP 地址作为网段，如果你碰巧输错了一个网段地址（比如将 `10.1.1.0 mask 255.255.255.0` 敲作 `10.1.1.1 mask 255.255.255.0`），路由器将接受网络配置，你就可能花费时间去检查为什么 BGP 没有通告 `10.1.1.0/24` 网段，而你的实际配置却是通告 `10.1.1.1/24` 网段。

8.7 通告本地网段

有很多种方式可以通告网段给 BGP 对等体，用来通告网段的命令取决于一些其他的变量。比如，你可能想要 BGP 精确控制通告给它的远端对等体的网段，你可能想通告所有路由器直接相连的网段，或者你可能想要通告静态路由，将路由固定住，这样不管到该网段的路径如何变化，BGP 通告给上游路由器的都是同样的路由。或者在某些特定的情况下，你可能想要通告全部的 IGP 路由表给它的远端对等体，BGP 通过给你关于路由来源的不同选择使你可以控制如何通告网段。下面列出了这些选择：

- 使用 `network` 命令；
- 重分发连接网段；
- 重分发静态路由；
- 重分发 IGP 路由。

本节展示了如何使用这里列出的命令来向 BGP 对等体通告网段，下面的范例中使用了图 8-11 所示的网络。

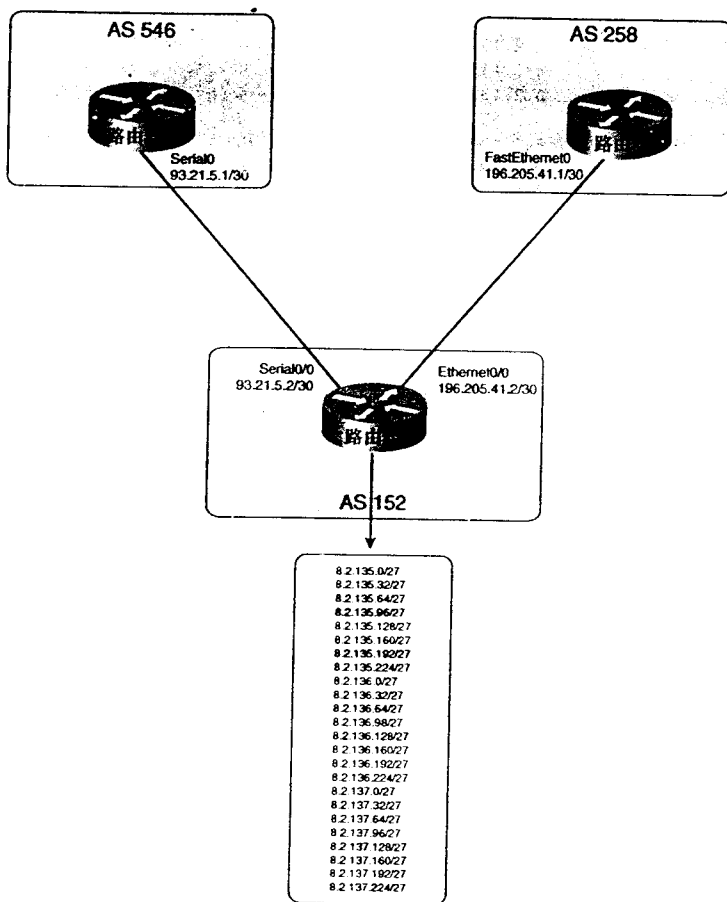


图 8-11 Reservoir 网络

8.7.1 通告连接网段

如前所述，如果你需要动态地通告直接连接的网段，你可能需要向本地的 BGP 进程重分发连接网段，这样可以限制静态配置的数量。以范例 8-54 中的路由器为例，有超过 20 个环回接口需要通过 BGP 通告。

范例 8-54 通告很多直连网段

```
Black# show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	196.205.41.2	YES	manual	up	up
Serial0/0	93.21.5.2	YES	manual	up	up
Loopback2	8.2.135.1	YES	manual	up	up
Loopback3	8.2.135.33	YES	manual	up	up
Loopback4	8.2.135.65	YES	manual	up	up
Loopback5	8.2.135.97	YES	manual	up	up
Loopback6	8.2.135.129	YES	manual	up	up
Loopback7	8.2.135.161	YES	manual	up	up
Loopback8	8.2.135.193	YES	manual	up	up

(待续)

Loopback9	8.2.135.225	YES manual up	up
Loopback10	8.2.136.1	YES manual up	up
Loopback11	8.2.136.33	YES manual up	up
Loopback12	8.2.136.65	YES manual up	up
Loopback13	8.2.136.97	YES manual up	up
Loopback14	8.2.136.129	YES manual up	up
Loopback15	8.2.136.161	YES manual up	up
Loopback16	8.2.136.193	YES manual up	up
Loopback17	8.2.136.225	YES manual up	up
Loopback18	8.2.137.1	YES manual up	up
Loopback19	8.2.137.33	YES manual up	up
Loopback20	8.2.137.65	YES manual up	up
Loopback21	8.2.137.97	YES manual up	up
Loopback22	8.2.137.129	YES manual up	up
Loopback23	8.2.137.161	YES manual up	up
Loopback24	8.2.137.193	YES manual up	up
Loopback25	8.2.137.225	YES manual up	up

可以使用 BGP **network** 命令来通告所有这些网段，但是这样需要很多的配置而且无法动态地增加和减少路由，同时也可能会有很多的拼写错误，如范例 8-55 所示。

范例 8-55 使用 network 命令通告网段

```
Black# show run | begin bgp
router bgp 152
  bgp log-neighbor-changes
  network 8.2.135.0 mask 255.255.255.224
  network 8.2.135.32 mask 255.255.255.224
  network 8.2.135.64 mask 255.255.255.224
  network 8.2.135.96 mask 255.255.255.224
  network 8.2.135.128 mask 255.255.255.224
  network 8.2.135.160 mask 255.255.255.224
  network 8.2.135.192 mask 255.255.255.224
  network 8.2.135.224 mask 255.255.255.224
  network 8.2.136.0 mask 255.255.255.224
  network 8.2.136.32 mask 255.255.255.224
  network 8.2.136.64 mask 255.255.255.224
  network 8.2.136.96 mask 255.255.255.224
  network 8.2.136.128 mask 255.255.255.224
  network 8.2.136.160 mask 255.255.255.224
  network 8.2.136.192 mask 255.255.255.224
  network 8.2.136.224 mask 255.255.255.224
  network 8.2.137.0 mask 255.255.255.224
  network 8.2.137.32 mask 255.255.255.224
  network 8.2.137.64 mask 255.255.255.224
  network 8.2.137.96 mask 255.255.255.224
  network 8.2.137.128 mask 255.255.255.224
  network 8.2.137.160 mask 255.255.255.224
  network 8.2.137.192 mask 255.255.255.224
  network 8.2.137.224 mask 255.255.255.224
  neighbor 93.21.5.1 remote-as 546
  neighbor 196.205.41.1 remote-as 258

Black# show ip bgp
BGP table version is 32, local router ID is 8.2.137.225
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
*> 8.2.135.0/27    0.0.0.0             0           32768 i
*> 8.2.135.32/27   0.0.0.0             0           32768 i
```

(待续)

Loopback9	8.2.135.225	YES manual up	up
Loopback10	8.2.136.1	YES manual up	up
Loopback11	8.2.136.33	YES manual up	up
Loopback12	8.2.136.65	YES manual up	up
Loopback13	8.2.136.97	YES manual up	up
Loopback14	8.2.136.129	YES manual up	up
Loopback15	8.2.136.161	YES manual up	up
Loopback16	8.2.136.193	YES manual up	up
Loopback17	8.2.136.225	YES manual up	up
Loopback18	8.2.137.1	YES manual up	up
Loopback19	8.2.137.33	YES manual up	up
Loopback20	8.2.137.65	YES manual up	up
Loopback21	8.2.137.97	YES manual up	up
Loopback22	8.2.137.129	YES manual up	up
Loopback23	8.2.137.161	YES manual up	up
Loopback24	8.2.137.193	YES manual up	up
Loopback25	8.2.137.225	YES manual up	up

可以使用 BGP **network** 命令来通告所有这些网段，但是这样需要很多的配置而且无法动态地增加和减少路由，同时也可能会有很多的拼写错误，如范例 8-55 所示。

范例 8-55 使用 network 命令通告网段

```
Black# show run | begin bgp
router bgp 152
  bgp log-neighbor-changes
  network 8.2.135.0 mask 255.255.255.224
  network 8.2.135.32 mask 255.255.255.224
  network 8.2.135.64 mask 255.255.255.224
  network 8.2.135.96 mask 255.255.255.224
  network 8.2.135.128 mask 255.255.255.224
  network 8.2.135.160 mask 255.255.255.224
  network 8.2.135.192 mask 255.255.255.224
  network 8.2.135.224 mask 255.255.255.224
  network 8.2.136.0 mask 255.255.255.224
  network 8.2.136.32 mask 255.255.255.224
  network 8.2.136.64 mask 255.255.255.224
  network 8.2.136.96 mask 255.255.255.224
  network 8.2.136.128 mask 255.255.255.224
  network 8.2.136.160 mask 255.255.255.224
  network 8.2.136.192 mask 255.255.255.224
  network 8.2.136.224 mask 255.255.255.224
  network 8.2.137.0 mask 255.255.255.224
  network 8.2.137.32 mask 255.255.255.224
  network 8.2.137.64 mask 255.255.255.224
  network 8.2.137.96 mask 255.255.255.224
  network 8.2.137.128 mask 255.255.255.224
  network 8.2.137.160 mask 255.255.255.224
  network 8.2.137.192 mask 255.255.255.224
  network 8.2.137.224 mask 255.255.255.224
  neighbor 93.21.5.1 remote-as 546
  neighbor 196.205.41.1 remote-as 258

Black# show ip bgp
BGP table version is 32, local router ID is 8.2.137.225
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
*> 8.2.135.0/27    0.0.0.0              0           32768 i
*> 8.2.135.32/27   0.0.0.0              0           32768 i
```

(待续)

*> 8.2.135.64/27	0.0.0.0	0	32768	i
*> 8.2.135.96/27	0.0.0.0	0	32768	i
*> 8.2.135.128/27	0.0.0.0	0	32768	i
*> 8.2.135.160/27	0.0.0.0	0	32768	i
*> 8.2.135.192/27	0.0.0.0	0	32768	i
*> 8.2.135.224/27	0.0.0.0	0	32768	i
*> 8.2.136.0/27	0.0.0.0	0	32768	i
*> 8.2.136.32/27	0.0.0.0	0	32768	i
*> 8.2.136.64/27	0.0.0.0	0	32768	i
*> 8.2.136.128/27	0.0.0.0	0	32768	i
*> 8.2.136.160/27	0.0.0.0	0	32768	i
*> 8.2.136.192/27	0.0.0.0	0	32768	i
*> 8.2.136.224/27	0.0.0.0	0	32768	i
*> 8.2.137.0/27	0.0.0.0	0	32768	i
*> 8.2.137.32/27	0.0.0.0	0	32768	i
*> 8.2.137.64/27	0.0.0.0	0	32768	i
Network	Next Hop	Metric	LocPrf	Weight Path
*> 8.2.137.96/27	0.0.0.0	0	32768	i
*> 8.2.137.128/27	0.0.0.0	0	32768	i
*> 8.2.137.160/27	0.0.0.0	0	32768	i
*> 8.2.137.192/27	0.0.0.0	0	32768	i
*> 8.2.137.224/27	0.0.0.0	0	32768	i

另外一个方法是，可以使用 **redistribute connected** 命令来告诉 BGP 自动重分发所有的直连网段，如范例 8-56 所示。

范例 8-56 使用 redistribute connected 命令

Black# show run begin bgp				
router bgp 152				
no synchronization				
bgp log-neighbor-changes				
redistribute connected				
neighbor 93.21.5.1 remote-as 546				
neighbor 196.205.41.1 remote-as 258				
Black# show ip bgp				
BGP table version is 5, local router ID is 8.2.137.225				
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal				
Origin codes: i - IGP, e - EGP, ? - incomplete				
Network	Next Hop	Metric	LocPrf	Weight Path
*> 8.0.0.0	0.0.0.0	0	32768	?
*> 93.0.0.0	0.0.0.0	0	32768	?
*> 196.205.41.0	0.0.0.0	0	32768	?

注意，当你如前面的范例中那样使用 **redistribute connected** 命令时，BGP 会自动地在网段的分类边界进行汇总。但是很少有网段可以正好在它们的分类边界被汇总，为了克服 BGP 的这个默认行为，可以使用 **no auto-summary** 命令来告诉 BGP 不要自动汇总网段，如范例 8-57 所示。

范例 8-57 使用 BGP no auto-summary 命令

Black# show run begin bgp				
router bgp 152				
bgp log-neighbor-changes				
redistribute connected				
neighbor 93.21.5.1 remote-as 546				

(待续)

```

neighbor 196.205.41.1 remote-as 258
no auto-summary

Black# show ip bgp
BGP table version is 28, local router ID is 8.2.137.225
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop          Metric LocPrf Weight Path
*> 1.1.1.1/32        0.0.0.0              0           32768 ?
*> 8.2.135.0/27       0.0.0.0              0           32768 ?
*> 8.2.135.32/27      0.0.0.0              0           32768 ?
*> 8.2.135.64/27      0.0.0.0              0           32768 ?
*> 8.2.135.96/27      0.0.0.0              0           32768 ?
*> 8.2.135.128/27     0.0.0.0              0           32768 ?
*> 8.2.135.160/27     0.0.0.0              0           32768 ?
*> 8.2.135.192/27     0.0.0.0              0           32768 ?
*> 8.2.135.224/27     0.0.0.0              0           32768 ?
*> 8.2.136.0/27       0.0.0.0              0           32768 ?
*> 8.2.136.32/27      0.0.0.0              0           32768 ?
*> 8.2.136.64/27      0.0.0.0              0           32768 ?
*> 8.2.136.96/27      0.0.0.0              0           32768 ?
*> 8.2.136.128/27     0.0.0.0              0           32768 ?
*> 8.2.136.160/27     0.0.0.0              0           32768 ?
*> 8.2.136.192/27     0.0.0.0              0           32768 ?
*> 8.2.136.224/27     0.0.0.0              0           32768 ?
*> 8.2.137.0/27       0.0.0.0              0           32768 ?
   Network          Next Hop          Metric LocPrf Weight Path
*> 8.2.137.32/27      0.0.0.0              0           32768 ?
*> 8.2.137.64/27      0.0.0.0              0           32768 ?
*> 8.2.137.96/27      0.0.0.0              0           32768 ?
*> 8.2.137.128/27     0.0.0.0              0           32768 ?
*> 8.2.137.160/27     0.0.0.0              0           32768 ?
*> 8.2.137.192/27     0.0.0.0              0           32768 ?
*> 8.2.137.224/27     0.0.0.0              0           32768 ?
*> 93.21.5.0/30       0.0.0.0              0           32768 ?
*> 196.205.41.0/30    0.0.0.0              0           32768 ?

```

8.7.2 通告静态路由

一种使用 BGP 向因特网通告非常稳定的 BGP 路由的方式是固定路由，配置较高的管理距离的发往 null0 的静态路由，这将使路由器通告由静态路由指定的网段给它的邻居。由于发往 null0 的路由有较高的管理距离（比如 253），任何由其他路由协议接收到的路由都在主 IP 路由表中优先使用。通过 IGP 邻居学到的动态路由可能会改变甚至消失，BGP 仍然会不断地通告静态的固定路由。范例 8-58 显示了如何使用 **redistribute static** 命令、发往 null0 的静态路由以及 **no auto-summary** 命令来创建稳定的面向因特网的路由。

范例 8-58 重分发静态路由

```

Black# show run | begin bgp
router bgp 152
  no synchronization
  bgp log-neighbor-changes
  redistribute static
  neighbor 93.21.5.1 remote-as 546
  neighbor 196.205.41.1 remote-as 258

```

（待续）

```
no auto-summary
!
ip classless
ip route 8.2.135.0 255.255.255.224 Null0 254
ip route 8.2.135.32 255.255.255.224 Null0 254
ip route 8.2.135.64 255.255.255.224 Null0 254
ip route 8.2.135.96 255.255.255.224 Null0 254
ip route 8.2.135.128 255.255.255.224 Null0 254
ip route 8.2.135.160 255.255.255.224 Null0 254
ip route 8.2.135.192 255.255.255.224 Null0 254
ip route 8.2.135.224 255.255.255.224 Null0 254
ip route 8.2.136.0 255.255.255.224 Null0 254
ip route 8.2.136.32 255.255.255.224 Null0 254
ip route 8.2.136.64 255.255.255.224 Null0 254
ip route 8.2.136.96 255.255.255.224 Null0 254
ip route 8.2.136.128 255.255.255.224 Null0 254
ip route 8.2.136.160 255.255.255.224 Null0 254
ip route 8.2.136.192 255.255.255.224 Null0 254
ip route 8.2.136.224 255.255.255.224 Null0 254
ip route 8.2.137.0 255.255.255.224 Null0 254
ip route 8.2.137.32 255.255.255.224 Null0 254
ip route 8.2.137.64 255.255.255.224 Null0 254
ip route 8.2.137.96 255.255.255.224 Null0 254
ip route 8.2.137.128 255.255.255.224 Null0 254
ip route 8.2.137.160 255.255.255.224 Null0 254
ip route 8.2.137.192 255.255.255.224 Null0 254
ip route 8.2.137.224 255.255.255.224 Null0 254
```

Black# show ip bgp

BGP table version is 25, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 8.2.135.0/27	0.0.0.0	0		32768	?
*> 8.2.135.32/27	0.0.0.0	0		32768	?
*> 8.2.135.64/27	0.0.0.0	0		32768	?
*> 8.2.135.96/27	0.0.0.0	0		32768	?
*> 8.2.135.128/27	0.0.0.0	0		32768	?
*> 8.2.135.160/27	0.0.0.0	0		32768	?
*> 8.2.135.192/27	0.0.0.0	0		32768	?
*> 8.2.135.224/27	0.0.0.0	0		32768	?
*> 8.2.136.0/27	0.0.0.0	0		32768	?
*> 8.2.136.32/27	0.0.0.0	0		32768	?
*> 8.2.136.64/27	0.0.0.0	0		32768	?
*> 8.2.136.96/27	0.0.0.0	0		32768	?
*> 8.2.136.128/27	0.0.0.0	0		32768	?
*> 8.2.136.160/27	0.0.0.0	0		32768	?
*> 8.2.136.192/27	0.0.0.0	0		32768	?
*> 8.2.136.224/27	0.0.0.0	0		32768	?
*> 8.2.137.0/27	0.0.0.0	0		32768	?
*> 8.2.137.32/27	0.0.0.0	0		32768	?
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 8.2.137.64/27	0.0.0.0	0		32768	?
*> 8.2.137.96/27	0.0.0.0	0		32768	?
*> 8.2.137.128/27	0.0.0.0	0		32768	?
*> 8.2.137.160/27	0.0.0.0	0		32768	?
*> 8.2.137.192/27	0.0.0.0	0		32768	?
*> 8.2.137.224/27	0.0.0.0	0		32768	?

注意每条路由都存放在可以通告给其他任何对等体的 BGP 表中；如有 IGP 路由存在，路由器将转发前往 **redistribute static** 指定的网段的流量到正确的目的，使用 **redistribute static** 只是为了保证 IGP 路由的修改或消失不会中断 BGP 服务。需要牢记的是，如果你使用了目

的为 null0 的静态路由，你仍然需要有较低的管理距离的到目的网段的路由，否则路由器实际上会将路由转发给接口 null0 比特桶（bit bucket）。

8.7.3 通告由 IGP 学到的路由

最后也是最不期望的向 BGP 通告本地始发的路由的方式是动态地向 BGP 重分发 IGP 路由。由于 IGP 路由往往变化频繁，你（和你对等连接的任何其他路由器）不会希望 BGP 经常不停地增加、修改或是删除 IGP 重分发的路由，所以在实践中不推荐使用向 BGP 动态重分发 IGP 路由。尽管如此，你还是可以使用 *redistribute protocol* 命令将 IGP 路由直接重分发进入 BGP。范例 8-59 显示了 OSPF 进程通告的路由如何动态地重分发进入 BGP。这个范例显示了通过 OSPF 接收到的路由、OSPF/BGP 的配置以及最后的 BGP 表。

范例 8-59 将 IGP 路由重分发进入 BGP

```
Black# show run | begin ospf
router ospf 1
 log-adjacency-changes
 network 8.2.138.0 0.0.0.3 area 0
Black# show ip route
 196.205.41.0/30 is subnetted, 1 subnets
C    196.205.41.0 is directly connected, Ethernet0/0
 8.0.0.0/8 is variably subnetted, 25 subnets, 2 masks
O    8.2.137.129/32 [110/65] via 8.2.138.2, 00:02:29, Serial0/1
O    8.2.136.129/32 [110/65] via 8.2.138.2, 00:02:29, Serial0/1
O    8.2.135.129/32 [110/65] via 8.2.138.2, 00:02:29, Serial0/1
O    8.2.137.161/32 [110/65] via 8.2.138.2, 00:02:29, Serial0/1
O    8.2.136.161/32 [110/65] via 8.2.138.2, 00:02:30, Serial0/1
O    8.2.135.161/32 [110/65] via 8.2.138.2, 00:02:30, Serial0/1
O    8.2.137.193/32 [110/65] via 8.2.138.2, 00:02:30, Serial0/1
O    8.2.136.193/32 [110/65] via 8.2.138.2, 00:02:30, Serial0/1
O    8.2.135.193/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.137.225/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.136.225/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.135.225/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
C    8.2.138.0/30 is directly connected, Serial0/1
O    8.2.137.1/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.136.1/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.135.1/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.137.33/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.136.33/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.135.33/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.137.65/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.136.65/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.135.65/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.137.97/32 [110/65] via 8.2.138.2, 00:02:31, Serial0/1
O    8.2.136.97/32 [110/65] via 8.2.138.2, 00:02:32, Serial0/1
O    8.2.135.97/32 [110/65] via 8.2.138.2, 00:02:32, Serial0/1
 93.0.0.0/30 is subnetted, 1 subnets
C    93.21.5.0 is directly connected, Serial0/0
Black# show run | begin bgp
router bgp 152
 no synchronization
 bgp log-neighbor-changes
 redistribute ospf 1 match internal external 1 external 2
 neighbor 93.21.5.1 remote-as 546
 neighbor 196.205.41.1 remote-as 258
 no auto-summary
```

（待续）

```
Black# show ip bgp
BGP table version is 26, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
*> 8.2.135.1/32    8.2.138.2           65             32768 ?
*> 8.2.135.33/32   8.2.138.2           65             32768 ?
*> 8.2.135.65/32   8.2.138.2           65             32768 ?
*> 8.2.135.97/32   8.2.138.2           65             32768 ?
*> 8.2.135.129/32  8.2.138.2           65             32768 ?
*> 8.2.135.161/32  8.2.138.2           65             32768 ?
*> 8.2.135.193/32  8.2.138.2           65             32768 ?
*> 8.2.135.225/32  8.2.138.2           65             32768 ?
*> 8.2.136.1/32    8.2.138.2           65             32768 ?
*> 8.2.136.33/32   8.2.138.2           65             32768 ?
*> 8.2.136.65/32   8.2.138.2           65             32768 ?
*> 8.2.136.97/32   8.2.138.2           65             32768 ?
*> 8.2.136.129/32  8.2.138.2           65             32768 ?
*> 8.2.136.161/32  8.2.138.2           65             32768 ?
*> 8.2.136.193/32  8.2.138.2           65             32768 ?
*> 8.2.136.225/32  8.2.138.2           65             32768 ?
*> 8.2.137.1/32    8.2.138.2           65             32768 ?
*> 8.2.137.33/32   8.2.138.2           65             32768 ?
   Network        Next Hop           Metric LocPrf Weight Path
*> 8.2.137.65/32   8.2.138.2           65             32768 ?
*> 8.2.137.97/32   8.2.138.2           65             32768 ?
*> 8.2.137.129/32  8.2.138.2           65             32768 ?
*> 8.2.137.161/32  8.2.138.2           65             32768 ?
*> 8.2.137.193/32  8.2.138.2           65             32768 ?
*> 8.2.137.225/32  8.2.138.2           65             32768 ?
*> 8.2.138.0/30    0.0.0.0             0              32768 ?
```

注意在前面的范例中，IGP 和 BGP 之间的重分发是非常直接的过程，只需要一个或是两个命令（根据你对 **auto-summary** 的要求）。然而，重分发进入 BGP 的路由数量可能会很大，只有在 IGP 通告的网段稳定的时候这些路由才会稳定。除非绝对需要，最好不要使用这个命令。

8.8 实验 14: BGP 路由

在本章中你已经学到在一个生产网络中可以有很多方法来使用 BGP，最常用的方式是为了获得因特网的连接使用 BGP 将一个网段多归路到两个或是更多的服务提供商。下面的实验侧重于不同的 BGP 连接类型和使用话音在 IP 上传输（VoIP）的应用作为 BGP 路由的测试。

8.8.1 实验练习

在这个实验中，配置 “I-Scream for Coffee” 32-flavor 网络的 BGP 对等关系，使用 BGP 路由协议作为骨干来路由包括自治系统 203 中的 Mint 路由器与自治系统 507 中的 Chocolate 路由器在内的外部网络，内部网络是在自治系统 409 中的 Vanilla、Strawberry、Latte 和 Americano 路由器。为了测试你的 BGP 路由配置能力，需要在路由器 Chocolate 和 Latte 的电话之间发送测试呼叫。

8.8.2 实验目的

- 使用 E-BGP 和 I-BGP 以及相关命令来实现自治系统之间的路由。
- 允许 BGP 路由通过访问列表。
- 在 IGP 路由器的周边配置 BGP，不在所有的路由器上启用 BGP。
- 使用 BGP 提供的路由在不同自治系统中的路由器之间连接的电话上进行测试呼叫。

8.8.3 需要的设备

- 7 台思科路由器（其中两台需要可以进行 VoIP 测试的话音模块）。
- 6 台路由器只需要一个或是两个串行接口，3 台路由器需要串行和以太或是令牌环接口。
- 一个集线器、交换机或是 MSAU，用来连接 3 台多路访问的路由器。

8.8.4 物理布局和预规划

- 如图 8-12 所示连接路由器，其中 Mint、Chocolate、Vanilla 和 Strawberry 可以通过背对背串行电缆连接。

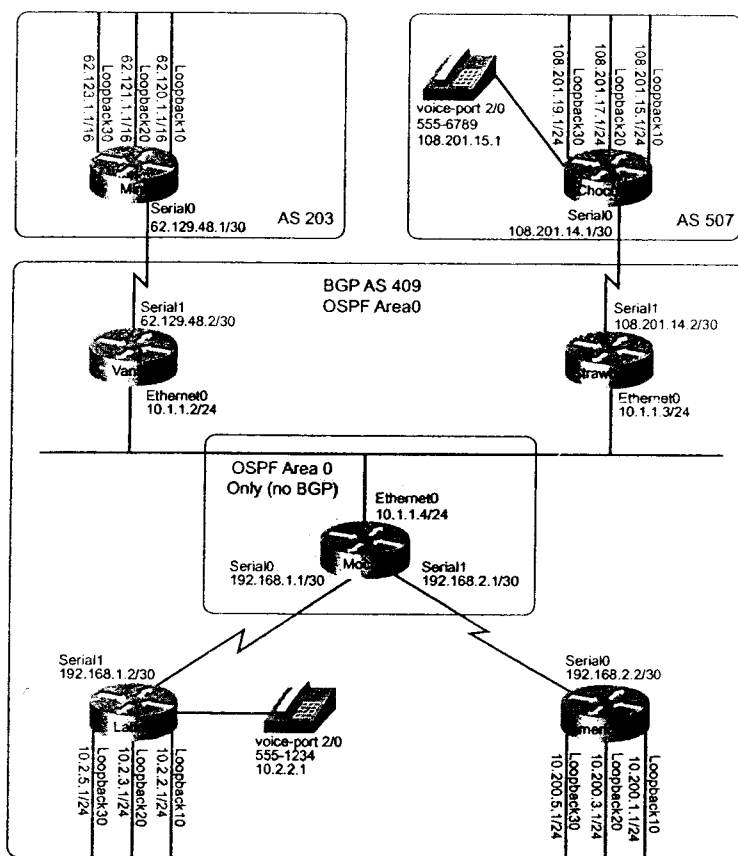


图 8-12 I-Scream for Coffee 网络

- 路由器 Vanilla、Strawberry 和 Mocha 需要背对背串行和以太（或是令牌环）连接。
- 路由器 Mocha、Latte 和 Americano 也需要背对背串行连接。
- 如图 8-12 所示在所有的环回接口、串行接口和以太接口上配置 IP 地址。
- 在除了 Mint 和 Chocolate 以外的所有路由器上启用 OSPF 路由，除了环回接口以外，这些路由器上的所有接口都属于 OSPF 区域 0，确认 OSPF 通告不会发送到非 OSPF 的接口上。

为了成功地完成这个实验，请按以下步骤执行：

- 第 1 步** 在路由器 Mint 和 Vanilla 之间配置 E-BGP 对等会话，将 Mint 路由器放入自治系统 203 中，将 Vanilla 路由器放入自治系统 409 中。在路由器 Mint 和 Vanilla 上配置完 BGP 后，在路由器 Chocolate 和 Strawberry 上配置 BGP，将路由器 Chocolate 放入自治系统 507 中，将路由器 Strawberry 放入自治系统 409 中。不使用 **network** 命令通告从自治系统 203 和 507 来的所有的外部环回接口地址，不允许 BGP 路由器进行自动汇总。使用 **show ip bgp** 和 **show ip bgp summary** 命令来测试 BGP 路由器的配置，使用 **show ip bgp neighbors** 和 **show tcp brief all** 命令来验证 TCP 的可达性。
- 第 2 步** 在路由器 Strawberry 和 Vanilla 之间配置 I-BGP 会话，验证路由器 Mint、Vanilla、Chocolate 和 Strawberry 都能访问彼此的 BGP 路由。
- 第 3 步** 在路由器 Vanilla、Latte 和 Americano 之间以及路由器 Strawberry、Latte 和 Americano 之间配置 I-BGP。配置路由器 Latte 和 Americano 使用 BGP 通告它们的环回接口和串行接口的 IP 地址。验证所有的 BGP 路由器都可以访问其他所有的路由器。
- 第 4 步** 通过在路由器 Chocolate 和 Latte 之间进行语音呼叫来测试配置。在路由器 Chocolate 和 Latte 上配置 VoIP，创建拨号对等体，增加目的模式和 IP 地址或是物理接口，然后从电话上拨出（关于 VoIP 配置的更多信息，请参考《CCIE 实验指南（第 1 卷）》）。

8.8.5 实验步骤

所有的路由器接线完毕后，使用命令 **show cdp neighbors** 和 **show ip interface brief** 验证连接性，这将大大节省故障排查接线和时钟速率的问题的时间。根据图 8-12 的信息为每台路由器配置 IP 地址，然后使用 **ping** 命令来验证直接相连网络的第三层的连接性。现在你已经验证了所有的路由器都能相互访问，然后在所有的路由器上都启用 OSPF，将所有的接口都放入区域 0，每台路由器都应该使用最大的非环回接口的 IP 地址作为 OSPF 的路由器识别符。启用 OSPF 后，使用命令 **show ip route**、**show ip ospf neighbors** 和 **show ip ospf interfaces** 验证所有的路由器都有到其他 OSPF 路由器的路由，在进入第 1 步之前验证它们可以相互 ping 通。

- 第 1 步** 在路由器 Mint 和 Vanilla 之间配置 E-BGP 对等会话，将路由器 Mint 放入自治系统 203 中，将路由器 Vanilla 放入自治系统 409 中。然后在路由器 Chocolate 和 Strawberry 上配置 BGP，将路由器 Chocolate 放入自治系统 507 中，将路由器 Strawberry 放入自治系统 409 中。不使用 **network** 命令通告从自治系统 203

和 507 来的所有的外部环回接口地址，不允许 BGP 路由器进行自动汇总。使用 **show ip bgp** 和 **show ip bgp summary** 命令来测试 BGP 路由器的配置，使用 **show ip bgp neighbors** 和 **show tcp brief all** 命令来验证 TCP 的可达性。范例 8-60 显示了路由器 Mint 和 Vanilla 的配置，范例 8-61 显示了路由器 Chocolate 和 Strawberry 的配置。

范例 8-60 路由器 Mint 和 Vanilla 的配置

```
Mint# show run | begin bgp
router bgp 203
  no synchronization
  bgp log-neighbor-changes
  redistribute connected
  neighbor 62.129.48.2 remote-as 409
  no auto-summary
```

```
Vanilla# show run | begin bgp
router bgp 409
  no synchronization
  bgp log-neighbor-changes
  neighbor 62.129.48.1 remote-as 203
  no auto-summary
Vanilla# show ip bgp
BGP table version is 17, local router ID is 62.129.48.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 62.120.0.0/16    62.129.48.1             0             0 203 ?
*> 62.121.0.0/16    62.129.48.1             0             0 203 ?
*> 62.123.0.0/16    62.129.48.1             0             0 203 ?
*> 62.129.48.0/30   62.129.48.1             0             0 203 ?
```

范例 8-61 路由器 Chocolate 和 Strawberry 的配置

```
Chocolate# show run | begin bgp
router bgp 507
  no synchronization
  bgp log-neighbor-changes
  redistribute connected
  neighbor 108.201.14.2 remote-as 409
  no auto-summary
```

```
Strawberry# show run | begin bgp
router bgp 409
  no synchronization
  bgp log-neighbor-changes
  neighbor 108.201.14.1 remote-as 507
  no auto-summary
Strawberry# show ip bgp
BGP table version is 11, local router ID is 108.201.14.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 108.201.14.0/30   108.201.14.1             0             0 507 ?
*> 108.201.15.0/24   108.201.14.1             0             0 507 ?
*> 108.201.17.0/24   108.201.14.1             0             0 507 ?
*> 108.201.19.0/24   108.201.14.1             0             0 507 ?
```

第 2 步 在路由器 Strawberry 和 Vanilla 之间配置 I-BGP 会话, 验证路由器 Mint、Vanilla、Chocolate 和 Strawberry 都能访问彼此的 BGP 路由。范例 8-62 显示了路由器 Vanilla 和 Strawberry 上是如何配置 BGP 的, 以及路由器之间是如何交换路由的。

范例 8-62 路由器 Strawberry 和 Vanilla 的 I-BGP 配置

```
Strawberry(config)# router bgp 409
Strawberry(config-router)# neighbor 10.1.1.2 remote-as 409
Strawberry(config-router)# neighbor 10.1.1.2 next-hop-self
Strawberry# show ip bgp | begin Network
  Network      Next Hop      Metric LocPrf Weight Path
*> 62.120.0.0/16 10.1.1.2        0    100    0 203 ?
*> 62.121.0.0/16 10.1.1.2        0    100    0 203 ?
*> 62.123.0.0/16 10.1.1.2        0    100    0 203 ?
*> 62.129.48.0/30 10.1.1.2        0    100    0 203 ?
*> 108.201.14.0/30 108.201.14.1    0          0 507 ?
*> 108.201.15.0/24 108.201.14.1    0          0 507 ?
*> 108.201.17.0/24 108.201.14.1    0          0 507 ?
*> 108.201.19.0/24 108.201.14.1    0          0 507 ?

Vanilla(config)#router bgp 409
Vanilla(config-router)#neighbor 10.1.1.3 remote-as 409
Vanilla(config-router)# neighbor 10.1.1.3 next-hop-self
Vanilla# show ip bgp | begin Network
  Network      Next Hop      Metric LocPrf Weight Path
*> 62.120.0.0/16 62.129.48.1      0          0 203 ?
*> 62.121.0.0/16 62.129.48.1      0          0 203 ?
*> 62.123.0.0/16 62.129.48.1      0          0 203 ?
*> 62.129.48.0/30 62.129.48.1      0          0 203 ?
*> i108.201.14.0/30 10.1.1.3        0    100    0 507 ?
*> i108.201.15.0/24 10.1.1.3        0    100    0 507 ?
*> i108.201.17.0/24 10.1.1.3        0    100    0 507 ?
*> i108.201.19.0/24 10.1.1.3        0    100    0 507 ?
```

在前面的范例中也演示了如何使用 **neighbor ip-address next-hop-self** 命令来改变 I-BGP 对等体之间传递的路由的下一跳属性。同时, 注意到在路由器 Vanilla 和 Strawberry 上配置 BGP 路由后, 尽管路由器在它们的 BGP 表中有了解有效的路由, 但是路由器 Vanilla 不能 ping 路由器 Chocolate 的网段, 路由器 Strawberry 不能 ping 路由器 Mint 的网段, 如下所示:

```
Vanilla# ping 108.201.14.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 108.201.14.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Strawberry# ping 62.129.48.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 62.129.48.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Strawberry#
```

在验证了路由器 Mint 能够 ping 通路由器 Vanilla 而且路由器 Strawberry 能够 ping 通路由器 Chocolate 后, 可以断定问题出在路由器 Chocolate 上 (或是 Mint 上, 取决于你先看哪个), 即路由器 Chocolate 不知道如何到达 10.0.0.0/8 网段 (路由器 Vanilla 和 Strawberry ping 的源 IP

地址)。为了解决这个问题，在路由器 Strawberry 和路由器 Vanilla 上增加一个网段描述，将 10.1.1.0/24 网段通告给路由器 Mint 和 Chocolate 然后再试。范例 8-63 显示了路由器 Strawberry 增加的 BGP 网络配置以及随后导致的路由器 Chocolate 的 IP 路由表的变化。这个范例同时也表明当路由器 Chocolate 收到了前往 10.1.1.0/24 网段的路由后，所有的 4 个 BGP 路由器能够 ping 所有的 BGP 网段。

范例 8-63 增加到 10.1.1.0/24 网段的路由

```
Strawberry(config)#router bgp 409
Strawberry(config-router)# network 10.1.1.0 mask 255.255.255.0
Chocolate# show ip route | begin Gateway
Gateway of last resort is not set
  10.0.0.0/24 is subnetted, 1 subnets
B       10.1.1.0 [20/0] via 108.201.14.10, 00:00:32
  108.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
S       108.201.14.10/32 [1/0] via 108.201.14.2
C       108.201.15.0/24 is directly connected, Loopback10
C       108.201.14.0/30 is directly connected, Serial0
C       108.201.17.0/24 is directly connected, Loopback20
C       108.201.19.0/24 is directly connected, Loopback30
Chocolate# ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/42/44 ms
Vanilla# ping 108.201.14.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 108.201.14.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

第 3 步 在路由器 Vanilla、Latte 和 Americano 之间以及路由器 Strawberry、Latte 和 Americano 之间配置 I-BGP。配置路由器 Latte 和 Americano 使用 BGP 通告它们的环回接口和串行接口的 IP 地址。验证所有的 BGP 路由器都可以访问其他所有的路由器。范例 8-64 显示了路由器 Vanilla 的配置和 BGP 表。

范例 8-64 路由器 Vanilla 的配置和 BGP 表

```
Vanilla# show run | begin bgp
router bgp 409
no synchronization
bgp log-neighbor-changes
network 10.1.1.0 mask 255.255.255.0
neighbor 10.1.1.3 remote-as 409
neighbor 10.1.1.3 next-hop-self
neighbor 62.129.48.1 remote-as 203
neighbor 192.168.1.2 remote-as 409
neighbor 192.168.2.2 remote-as 409
no auto-summary
Vanilla# show ip bgp
BGP table version is 435, local router ID is 62.129.48.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
* i10.1.1.0/24    10.1.1.3            0      100      0 i
*>                0.0.0.0             0              32768 i
```

(待续)

```
*>i10.2.2.0/24      192.168.1.2      0      100      0 ?
*>i10.2.3.0/24      192.168.1.2      0      100      0 ?
*>i10.2.5.0/24      192.168.1.2      0      100      0 ?
*>i10.200.1.0/24     192.168.2.2      0      100      0 ?
*>i10.200.3.0/24     192.168.2.2      0      100      0 ?
*>i10.200.5.0/24     192.168.2.2      0      100      0 ?
*> 62.120.0.0/16     62.129.48.1      0              0 203 ?
*> 62.121.0.0/16     62.129.48.1      0              0 203 ?
*> 62.123.0.0/16     62.129.48.1      0              0 203 ?
*> 62.129.48.0/30    62.129.48.1      0              0 203 ?
*>i108.201.14.0/30   10.1.1.3          0      100      0 507 ?
*>i108.201.15.0/24   10.1.1.3          0      100      0 507 ?
*>i108.201.17.0/24   10.1.1.3          0      100      0 507 ?
*>i108.201.19.0/24   10.1.1.3          0      100      0 507 ?
*>i192.168.1.0/30    192.168.1.2      0      100      0 ?
*>i192.168.2.0/30    192.168.2.2      0      100      0 ?
```

现在看看路由器 Latte，注意路由器 Latte 没有将外部 BGP 路由存为可达的，它们有*，但是没有>，这意味着它们是有效的，但是不可达，如范例 8-65 所示。

范例 8-65 路由器 Latte 的 BGP 表

```
Latte# show ip bgp
BGP table version is 6, local router ID is 10.2.5.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop        Metric LocPrf Weight Path
*>i10.1.1.0/24     10.1.1.3         0      100      0 i
* i               10.1.1.2         0      100      0 i
*> 10.2.2.0/24     0.0.0.0          0              32768 ?
*> 10.2.3.0/24     0.0.0.0          0              32768 ?
*> 10.2.5.0/24     0.0.0.0          0              32768 ?
* i62.120.0.0/16   62.129.48.1      0      100      0 203 ?
* i62.121.0.0/16   62.129.48.1      0      100      0 203 ?
* i62.123.0.0/16   62.129.48.1      0      100      0 203 ?
* i62.129.48.0/30  62.129.48.1      0      100      0 203 ?
* i108.201.14.0/30 108.201.14.1     0      100      0 507 ?
* i108.201.15.0/24 108.201.14.1     0      100      0 507 ?
* i108.201.17.0/24 108.201.14.1     0      100      0 507 ?
* i108.201.19.0/24 108.201.14.1     0      100      0 507 ?
*> 192.168.1.0/30  0.0.0.0          0              32768 ?
   Network        Next Hop        Metric LocPrf Weight Path
*>i192.168.2.0/30  192.168.2.2      0      100      0 i
```

这些路由不可达是因为上游的 BGP 邻居通告它们的时候带有的最初 E-BGP 下一跳地址是 62.129.48.1 和 108.201.14.1，而不是路由器 Latte 和 Americano 通过 OSPF 学到的本地可达的网段。关于这个问题的回答是很简单的，只需要不多的几个步骤——在所有的 I-BGP 发言路由器上增加 **next-hop-self** 描述，清除 BGP 进程，在路由器 Mocha 上增加两条路由，告诉它如何到达 62.0.0.0/8 和 108.201.0.0/16 网段，同时把所有的 I-BGP 发言路由器上关闭同步，这样它们就不用等待到这些网段的 OSPF 路由。在完成这些配置改变之后再重新检查路由，范例 8-66 显示了解决 I-BGP 路由问题的步骤和方案。

范例 8-66 解决 I-BGP 路由问题的步骤

```
Vanilla# show run | begin bgp
router bgp 409
```

(待续)


```

no synchronization
bgp log-neighbor-changes
network 10.1.1.0 mask 255.255.255.0
neighbor 10.1.1.3 remote-as 409
neighbor 10.1.1.3 next-hop-self
neighbor 62.129.48.1 remote-as 203
neighbor 192.168.1.2 remote-as 409
neighbor 192.168.1.2 next-hop-self
neighbor 192.168.2.2 remote-as 409
neighbor 192.168.2.2 next-hop-self

```

```

Strawberry# show run | begin bgp
router bgp 409
no synchronization
bgp log-neighbor-changes
network 10.1.1.0 mask 255.255.255.0
neighbor 10.1.1.2 remote-as 409
neighbor 10.1.1.2 next-hop-self
neighbor 108.201.14.1 remote-as 507
neighbor 192.168.1.2 remote-as 409
neighbor 192.168.1.2 next-hop-self
neighbor 192.168.2.2 remote-as 409
neighbor 192.168.2.2 next-hop-self
no auto-summary

```

```

Mocha# show run | begin ip route
ip route 62.0.0.0 255.0.0.0 10.1.1.2
ip route 108.201.0.0 255.255.0.0 10.1.1.3

```

```

Latte# show run | begin bgp
router bgp 409
no synchronization
bgp log-neighbor-changes
network 10.2.2.0 mask 255.255.255.0
network 10.2.3.0 mask 255.255.255.0
network 10.2.5.0 mask 255.255.255.0
network 192.168.1.0 mask 255.255.255.252
neighbor 10.1.1.2 remote-as 409
neighbor 10.1.1.2 next-hop-self
neighbor 10.1.1.3 remote-as 409
neighbor 10.1.1.3 next-hop-self
neighbor 192.168.2.2 remote-as 409
neighbor 192.168.2.2 next-hop-self
no auto-summary

```

```

Americano# show run | begin bgp
router bgp 409
no synchronization
network 10.200.1.0 mask 255.255.255.0
network 10.200.3.0 mask 255.255.255.0
network 10.200.5.0 mask 255.255.255.0
network 192.168.2.0 mask 255.255.255.252
neighbor 10.1.1.2 remote-as 409
neighbor 10.1.1.2 next-hop-self
neighbor 10.1.1.3 remote-as 409
neighbor 10.1.1.3 next-hop-self
neighbor 192.168.1.2 remote-as 409
neighbor 192.168.1.2 next-hop-self
no auto-summary

```

范例 8-67 显示了在改变前路由器 Latte 的 BGP 表，范例 8-68 显示了改变后的 BGP 表。

在第一个范例中，注意网段 62.120.0.0/16、62.121.0.0/16、62.122.0.0/16、62.129.48.0/30、

108.201.14.0/30、108.201.15.0/24、108.201.17.0/24 和 108.201.19.0/24 是不可达的，在第二个范例中，当你增加 **next-hop-self** 描述解决了错误的下一跳路由问题后，这些网段都可达了。

范例 8-67 在增加 next-hop-self 之前路由器 Latte 的 BGP 表

```
Latte# show ip bgp
BGP table version is 6, local router ID is 10.2.5.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i10.1.1.0/24	10.1.1.3	0	100	0	i
* i	10.1.1.2	0	100	0	i
*> 10.2.2.0/24	0.0.0.0	0		32768	?
*> 10.2.3.0/24	0.0.0.0	0		32768	?
*> 10.2.5.0/24	0.0.0.0	0		32768	?
* i62.120.0.0/16	62.129.48.1	0	100	0	203 ?
* i62.121.0.0/16	62.129.48.1	0	100	0	203 ?
* i62.123.0.0/16	62.129.48.1	0	100	0	203 ?
* i62.129.48.0/30	62.129.48.1	0	100	0	203 ?
* i108.201.14.0/30	108.201.14.1	0	100	0	507 ?
* i108.201.15.0/24	108.201.14.1	0	100	0	507 ?
* i108.201.17.0/24	108.201.14.1	0	100	0	507 ?
* i108.201.19.0/24	108.201.14.1	0	100	0	507 ?
*> 192.168.1.0/30	0.0.0.0	0		32768	?
Network	Next Hop	Metric	LocPrf	Weight	Path
*>i192.168.2.0/30	192.168.2.2	0	100	0	I

范例 8-68 在增加 next-hop-self 之后路由器 Latte 的 BGP 表

```
Latte# show ip bgp
BGP table version is 15, local router ID is 10.2.5.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i10.1.1.0/24	10.1.1.2	0	100	0	i
* i	10.1.1.3	0	100	0	i
*> 10.2.2.0/24	0.0.0.0	0		32768	?
*> 10.2.3.0/24	0.0.0.0	0		32768	?
*> 10.2.5.0/24	0.0.0.0	0		32768	?
*>i62.120.0.0/16	10.1.1.2	0	100	0	203 ?
*>i62.121.0.0/16	10.1.1.2	0	100	0	203 ?
*>i62.123.0.0/16	10.1.1.2	0	100	0	203 ?
*>i62.129.48.0/30	10.1.1.2	0	100	0	203 ?
*>i108.201.14.0/30	10.1.1.3	0	100	0	507 ?
*>i108.201.15.0/24	10.1.1.3	0	100	0	507 ?
*>i108.201.17.0/24	10.1.1.3	0	100	0	507 ?
*>i108.201.19.0/24	10.1.1.3	0	100	0	507 ?
*> 192.168.1.0/30	0.0.0.0	0		32768	?
Network	Next Hop	Metric	LocPrf	Weight	Path
*>i192.168.2.0/30	192.168.2.2	0	100	0	I

```
Latte# ping 108.201.14.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 108.201.14.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/42/44 ms
```

第 4 步 通过在路由器 Chocolate 和 Latte 之间进行语音呼叫来测试配置。在路由器 Chocolate 和 Latte 上配置 VoIP，创建拨号对等体，增加目的模式和 IP 地址或是物理接口，

然后从电话上拨出（关于 VoIP 配置的更多信息，请参考《CCIE 实验指南（第1卷）》）。范例 8-69 显示了路由器 Chocolate 上的语音配置，范例 8-70 显示了路由器 Latte 上的语音配置。

范例 8-69 路由器 Chocolate 上的语音配置

```
Chocolate# show run | begin dial
dial-peer voice 5551234 voip
 destination-pattern 5551234
 session target ipv4:10.2.2.1
!
dial-peer voice 5556789 pots
 destination-pattern 5556789
 port 2/0
```

范例 8-70 路由器 Latte 上的语音配置

```
Latte# show run | begin dial
dial-peer voice 5556789 voip
 destination-pattern 5556789
 session target ipv4:108.201.15.1
!
dial-peer voice 5551234 pots
 destination-pattern 5551234
 port 2/0
```

现在已经介绍了 BGP 的配置和故障排查命令，下面看看使 BGP 成为 IP 路由的最强大工具的特性。第 9 章包含了高级 BGP 特性，例如 BGP 属性的使用、路由过滤和策略、路由聚合、最佳路径选择过程的操纵以及 BGP 的调节。范例 8-71 显示了本实验中所有路由器的最终配置。

范例 8-71 实验 11 的最终路由器配置

```
hostname Mint
!
interface Loopback10
 ip address 62.120.1.1 255.255.0.0
!
interface Loopback20
 ip address 62.121.1.1 255.255.0.0
!
interface Loopback30
 ip address 62.123.1.1 255.255.0.0
!
interface Serial0
 ip address 62.129.48.1 255.255.255.252
 clockrate 13000000
!
router bgp 203
 no synchronization
 bgp log-neighbor-changes
 redistribute connected
 neighbor 62.129.48.2 remote-as 409
 no auto-summary
-----
hostname Vanilla
!
```

（待续）

```
interface Ethernet0
 ip address 10.1.1.2 255.255.255.0
!
interface Serial1
 ip address 62.129.48.2 255.255.255.252
!
router ospf 1
 router-id 10.1.1.2
 log-adjacency-changes
 passive-interface Serial1
 network 10.1.1.0 0.0.0.255 area 0
 network 62.129.48.0 0.0.0.3 area 0
!
router bgp 409
 no synchronization
 bgp log-neighbor-changes
 network 10.1.1.0 mask 255.255.255.0
 neighbor 10.1.1.3 remote-as 409
 neighbor 10.1.1.3 next-hop-self
 neighbor 62.129.48.1 remote-as 203
 neighbor 192.168.1.2 remote-as 409
 neighbor 192.168.1.2 next-hop-self
 neighbor 192.168.2.2 remote-as 409
 neighbor 192.168.2.2 next-hop-self
 no auto-summary

hostname Chocolate
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer voice 5551234 voip
 destination-pattern 5551234
 session target ipv4:10.2.2.1
!
dial-peer voice 5556789 pots
 destination-pattern 5556789
 port 2/0
!
interface Loopback10
 ip address 108.201.15.1 255.255.255.0
!
interface Loopback20
 ip address 108.201.17.1 255.255.255.0
!
interface Loopback30
 ip address 108.201.19.1 255.255.255.0
!
interface Serial0
 ip address 108.201.14.1 255.255.255.252
!
router bgp 507
 no synchronization
 bgp log-neighbor-changes
 redistribute connected
 neighbor 108.201.14.2 remote-as 409
 no auto-summary
!

hostname Strawberry
!
```

(待续)

```
interface Ethernet0
 ip address 10.1.1.3 255.255.255.0
!
interface Serial1
 ip address 108.201.14.2 255.255.255.252
 clockrate 1300000
!
router ospf 1
 router-id 10.1.1.3
 log-adjacency-changes
 passive-interface Serial1
 network 10.1.1.0 0.0.0.255 area 0
 network 108.201.14.0 0.0.0.3 area 0

!
router bgp 409
 no synchronization
 bgp log-neighbor-changes
 network 10.1.1.0 mask 255.255.255.0
 neighbor 10.1.1.2 remote-as 409
 neighbor 10.1.1.2 next-hop-self
 neighbor 108.201.14.1 remote-as 507
 neighbor 192.168.1.2 remote-as 409
 neighbor 192.168.1.2 next-hop-self
 neighbor 192.168.2.2 remote-as 409
 neighbor 192.168.2.2 next-hop-self
 no auto-summary
```

```
hostname Mocha
!
interface Ethernet0
 ip address 10.1.1.4 255.255.255.0
!
interface Serial0
 ip address 192.168.1.1 255.255.255.252
 clock rate 1300000
!
interface Serial1
 ip address 192.168.2.1 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 10.1.1.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.3 area 0
 network 192.168.2.0 0.0.0.3 area 0
!
ip classless
ip route 62.0.0.0 255.0.0.0 10.1.1.2
ip route 108.201.0.0 255.255.0.0 10.1.1.3
```

```
hostname Latte
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer voice 5556789 voip
 destination-pattern 5556789
 session target ipv4:108.201.15.1
!
dial-peer voice 5551234 pots
 destination-pattern 5551234
```

(待续)

```

port 2/0
!
interface Loopback10
 ip address 10.2.2.1 255.255.255.0
!
interface Loopback20
 ip address 10.2.3.1 255.255.255.0
!
interface Loopback30
 ip address 10.2.5.1 255.255.255.0
!
interface Serial0
 ip address 192.168.1.2 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 10.2.2.0 0.0.0.255 area 0
 network 10.2.3.0 0.0.0.255 area 0
 network 10.2.5.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.3 area 0
!
router bgp 409
 no synchronization
 bgp log-neighbor-changes
 redistribute connected
 network 10.200.2.0 mask 255.255.255.0
 network 10.200.3.0 mask 255.255.255.0
 network 10.200.5.0 mask 255.255.255.0
 network 192.168.1.0 mask 255.255.255.252
 neighbor 10.1.1.2 remote-as 409
 neighbor 10.1.1.2 next-hop-self
 neighbor 10.1.1.3 remote-as 409
 neighbor 10.1.1.3 next-hop-self
 neighbor 192.168.2.2 remote-as 409
 neighbor 192.168.2.2 next-hop-self
 no auto-summary

```

```

hostnameAmericano
!
interface Loopback10
 ip address 10.200.1.1 255.255.255.0
!
interface Loopback20
 ip address 10.200.3.1 255.255.255.0
!
interface Loopback30
 ip address 10.200.5.1 255.255.255.0
!
interface Serial0
 ip address 192.168.2.2 255.255.255.252
 clockrate 1300000
!
router ospf 1
 log-adjacency-changes
 network 10.200.1.0 0.0.0.255 area 0
 network 10.200.3.0 0.0.0.255 area 0
 network 10.200.5.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.3 area 0
!
router bgp 409
 no synchronization
 network 10.200.1.0 mask 255.255.255.0

```

(待续)

```
network 10.200.3.0 mask 255.255.255.0
network 10.200.5.0 mask 255.255.255.0
network 192.168.2.0 mask 255.255.255.252
neighbor 10.1.1.2 remote-as 409
neighbor 10.1.1.2 next-hop-self
neighbor 10.1.1.3 remote-as 409
neighbor 10.1.1.3 next-hop-self
neighbor 192.168.1.2 remote-as 409
neighbor 192.168.1.2 next-hop-self
no auto-summary
```

8.9 进一步阅读资料

Cisco IOS Configuration Fundamentals, by Cisco Systems Inc., Riva Technologies.

TCP/IP Principles, Protocols, and Architectures, by Douglas E.Comer.

Internet Routing Architectures, Second Edition, by Sam Halabi with Danny McPherson.

Routing TCP/IP, Volume II, by Jeff Doyle and Jennifer DeHaven Carroll.

Cisco BGP-4 Command and Configuration Handbook, by William R. Parkhurst.

第 9 章

高级 BGP 配置

前面的章节讨论了许多 BGP 故障排查的概念，探究了简单的 BGP 设计，同时显示了如何通告各种类型的网段。前面两章在一起提供了基础，或者说是 BGP 概念的回顾，在本章中对高级内容进行更多的技术讨论。本章演示了使用 BGP 来支持更大、更稳定的网络的方法，并且解释了如何实现高级路由策略。本章包括以下主题：

- BGP 路由认证；
- 如何使用路由反射器和联盟简化大型网络实现；
- 有效地使用 BGP 对等体组；
- 高级的 BGP 重分发方式；
- BGP 路由过滤、抑制和条件通告；
- 路由衰减；
- 路由聚合和策略；
- BGP 后门的使用；
- 如何配置 BGP 来支持不同的路由表大小以及如何维护对称路由；
- 调节 BGP 性能。

9.1 BGP 邻居认证

在 BGP 网络中减小安全风险的最简单的方法就是使用 BGP 对等体认证。思科的 BGP 实现使用了 RFC 2385 定义的 TCP MD-5 签名，这个逻辑是将配置的时候输入的密码作为关键值，对其进行 MD-5 的哈希运算，将得到的哈希值发送给远端对等体，密码本身不会通过连接发送。

使用 BGP MD-5 密码认证只需要一个配置步骤，那就是

rd password 启用密码认证，具体如下：

```
neighbor {ip-address | peer-group} password [0-7] password-string
```

这个命令也有可选参数，可以通过指定使用密码级别 7 来使用一个以前已经加密过的密码。

```
SlyDog(config-router)# neighbor 8.8.9.1 password 7 1511021F0725
```

一个认证的 BGP 对等会话的两端都必须使用同样的密码。如果一台路由器收到了带有无效密码的 BGP OPEN 报文，它将发送一个通知报文，说明由于认证失败导致 OPEN 报文出错。范例 9-1 显示了如何在两个 E-BGP 对等体之间使用密码认证来保护一个会话。

范例 9-1 BGP MD-5 密码认证

```
Mariner# show run | begin bgp
router bgp 5151
  bgp log-neighbor-changes
  neighbor 217.204.187.8 remote-as 1578
  neighbor 217.204.187.8 password cisco

OtherGuys# show run | begin bgp
router bgp 1578
  bgp log-neighbor-changes
  neighbor 217.204.187.7 remote-as 5151
  neighbor 217.204.187.7 password cisco
```

尽管使用 MD-5 认证不能完全保证 BGP 会话的安全，但是的确可以减小 BGP 会话被攻击的危险。

9.2 简化大型 BGP 网络

在几乎每一个大型的 BGP 网络中最终都会出现的问题就是设计的复杂性。当你有大量的 BGP 发言人路由器，它们又有内部或是外部的大量的 BGP 对等体，你最终需要重新评估网络设计以找到创建更简单、更可扩展的网络的方式。主动的网络专家将规划他们的网络使得每台路由器都有能力保持很大的 BGP 路由信息库（RIB），他们在规划网络的未来发展时也会考虑影响网络设计和实现的多种因素，下面列出了其中的一些因素：

- 参加 E-BGP 对等会话的路由器数和必须配置的对等体数；
- 对等连接的路由器之间发送的 BGP UPDATE 的数量、大小和频率；
- 由于多路径导致的非同步路由；
- 在网络收敛之前对等体之间必须发送的路径数和网络应用中收敛时间的延迟；
- 由于路由不稳定导致路由衰减的可能性；
- BGP 对等体的全网状连接的要求；
- 又长又复杂的路由器配置以及在路由器配置过程中人为出错的可能性。

你可以有很多方法处理每个不同的问题。本节讲述了如何使用路由反射器和联盟来帮助解决 I-BGP 全网状连接的问题，以及如何使用对等体组和路由聚合帮助控制大小和大型 BGP 实现的复杂性。

9.2.1 路由反射器

BGP 路由反射器在 RFC 1966 中定义，提供了在大型 I-BGP 实现中 I-BGP 全网状连接问题的一个简单解决方案。作为一个快速回顾，在路由反射器的情况下有两个实体：服务器和客户端。每个路由反射服务器需要与它的每个客户端建立 I-BGP 对等连接，然而每个客户端只需要与路由反射服务器建立一个连接即可。服务器通过 I-BGP 连接给它的每个客户端发送更新，排除了全网状拓扑的要求。

图 9-1 显示了需要帮助的一个 I-BGP 网络在使用路由反射器前后的样子。

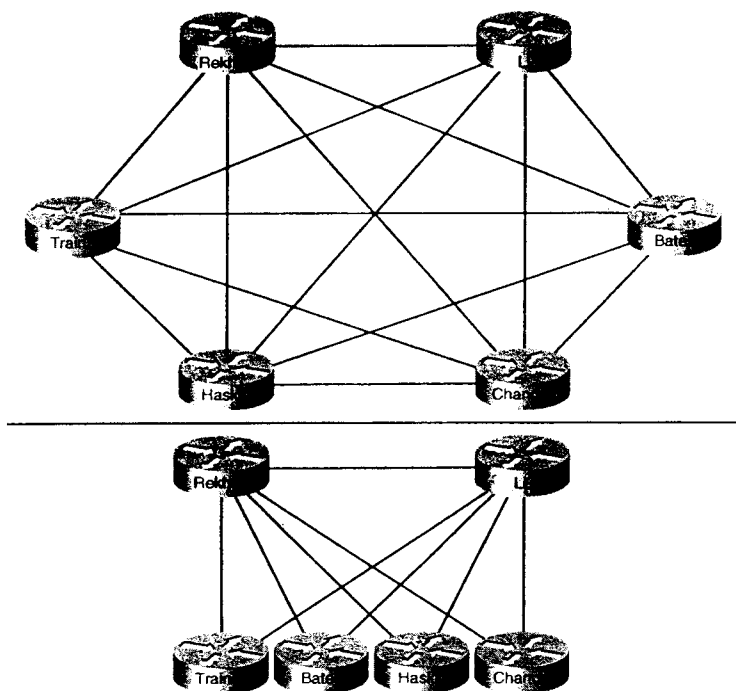


图 9-1 使用路由反射器前后

在图的第一部分，6 台路由器中的每一个都和它的所有对等体建立 I-BGP 对等连接，总共建立了 15 个 I-BGP 连接。图的第二部分显示了路由反射器如何简化这 6 台路由器的 I-BGP 配置——使用路由器 Rekhter 和 Li 作为路由反射服务器，路由 Traina、Haskin、Bates 和 Chandra 作为路由器 Rekhter 和 Li 的客户端。当路由反射器客户端连接到两个或是两个以上路由反射服务器的时候，路径的冗余性仍然得到了维持，同时配置也被大大简化了。

必须完成两个步骤来创建一个路由反射服务器，有时简称为路由服务器。我们使用图 9-2 中的网络来演示这个过程。

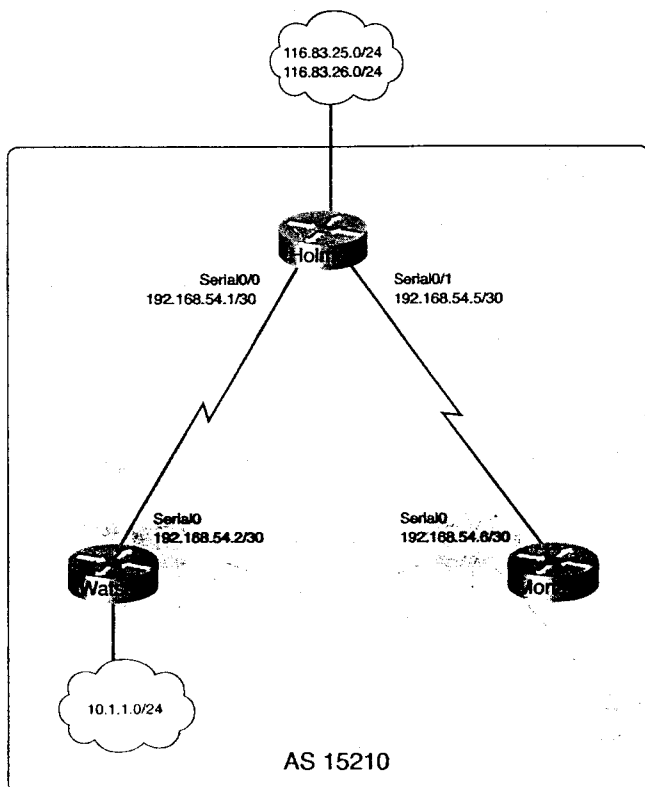


图 9-2 路由反射

第 1 步 为路由反射服务器要对等连接的每个 BGP 对等体配置 I-BGP。范例 9-2 显示了路由器 Holmes 的初始 BGP 配置。

范例 9-2 将路由器 Holmes 配置为路由反射服务器

```
Holmes# show run | begin bgp
router bgp 15210
no synchronization
neighbor 192.168.54.2 remote-as 15210
neighbor 192.168.54.6 remote-as 15210
```

第 2 步 在路由反射服务器上使用命令 **neighbor ip-address route-reflector-client** 配置充当路由反射客户端的每个邻居。范例 9-3 显示了路由器 Holmes 上的路由反射服务器配置。

范例 9-3 路由器 Holmes 上的路由反射器配置

```
neighbor 192.168.54.2 route-reflector-client
neighbor 192.168.54.6 route-reflector-client
```

不需要特别的配置来使路由器充当路由反射客户端，所有你需要做的就是配置客户端与路由反射服务器的对等连接。范例 9-4 显示了路由器 Watson 和 Moriarty 的路由反射客户端的

BGP 配置。

范例 9-4 路由反射客户端的 BGP 配置

```
Watson# show run | begin bgp
router bgp 15210
no synchronization
neighbor 192.168.54.1 remote-as 15210

Moriarty# show run | begin bgp
router bgp 15210
no synchronization
neighbor 192.168.54.5 remote-as 15210
```

命令 **show ip bgp neighbors | include BGP neighbor|Route-Reflector** 显示了由路由反射服务器提供路由的路由器的简单总结，如范例 9-5 所示。

范例 9-5 显示路由反射器客户端总结

```
Holmes# show ip bgp neighbors | include BGP neighbor|Route-Reflector
BGP neighbor is 192.168.54.2, remote AS 15210, internal link
Route-Reflector Client
BGP neighbor is 192.168.54.6, remote AS 15210, internal link
Route-Reflector Client
```

可以使用如范例 9-6 所示的命令 **show ip bgp ip-prefix** 来验证从路由反射服务器学到的路由。

范例 9-6 显示路由反射服务器信息

```
Moriarty# show ip bgp 10.1.1.0/24
BGP routing table entry for 10.1.1.0/24, version 8
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x208
Not advertised to any peer
Local
192.168.54.2 from 192.168.54.5 (10.1.1.1)
Origin IGP, metric 0, localpref 100, valid, internal, best
Originator: 10.1.1.1, Cluster list: 116.83.26.1
```

在前面的范例中，路由器 Moriarty 显示了到网段 10.1.1.0/24 的路由包含两个新的 BGP 属性：起源者属性指明了始发本路由的路由器的 BGP 路由器识别符，集群列表属性指明了路由的 BGP 集群识别符。BGP 集群识别符是始发路由的路由反射服务器的 BGP 路由器识别符，集群列表是一种避免环路的机制，设计用来避免属于一个路由反射集群的路由器从属于其他集群的路由器那里接收始发于本集群的路由。如果路由反射器接收到了在集群列表中包含自己的集群识别符的路由，它将忽略这个路由。

注意：如果一个路由通过了一个以上的路由反射集群，那么在路由的集群列表中将有一个以上的集群识别符，每个路由反射器在向自己的客户端转发路由时会将它自己本地的集群识别符加到集群列表中。请参考第 7 章的“路由反射器”一节获取关于这些 BGP 属性的更多信息。

9.2.2 联盟

另外一个实现 I-BGP 对等体的全网状连接需求的办法是配置 BGP 联盟。作为一个快速回顾，BGP 联盟将大型的 I-BGP 自治系统分割成小型的更易于管理的子自治系统，这些子自治系统也称为成员自治系统。将前面图 9-1 中所示的范例和图 9-3 所示的联盟解决方案相比较，你可以看到同样的网络如何使用 BGP 联盟来重新配置。

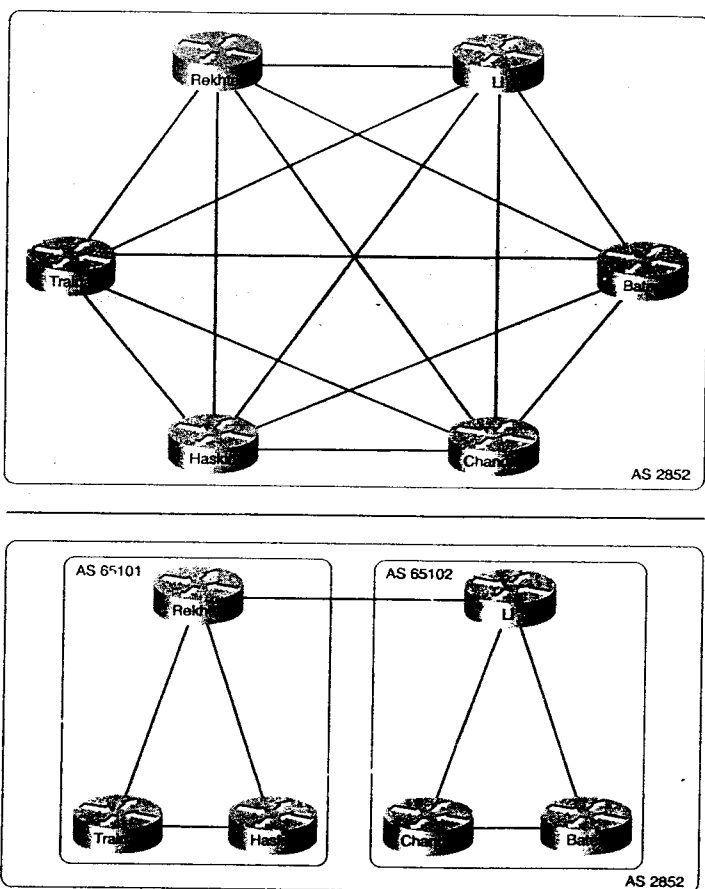


图 9-3 配置 BGP 联盟前后

注意在本例中联盟将路由器 Rekhter、Traina 和 Haskin 划入成员自治系统 65 101，将路由器 Li、Chandra 和 Bates 划入成员自治系统 65 102。所有在自治系统 65 101 和 65 102 中的路由器仍然属于自治系统 2852，减少了必须配置的 I-BGP 对等连接数量，同时注意一个子自治系统中的每个 I-BGP 路由器仍然和同一子自治系统中其他的 I-BGP 对等体成全网状连接。这带来了需要注意的关于联盟使用的一个关键点：尽管联盟是一个 I-BGP 全网状连接问题的简单解决方案，它们在每个子自治系统内还是需要全网状的对等关系，所以它们必须被小心地设计以满足发展的要求。

在 BGP 自治系统中配置联盟必须完成 5 个步骤。下面描述的过程使用如图 9-4 所示的网络。

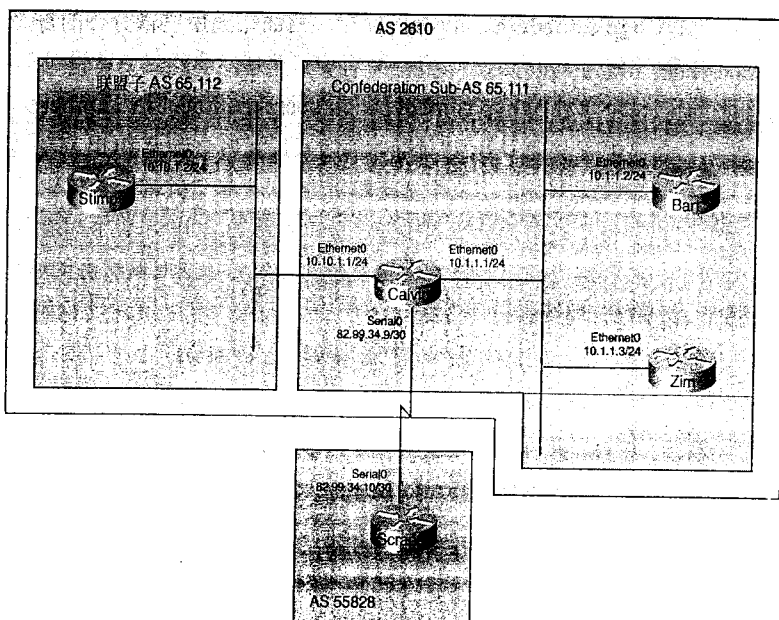


图 9-4 Good-Old-Boy 网络

第 1 步 如下所示在路由器 Calvin 上使用成员自治系统作为 BGP 自治系统启用 BGP 路由：

```
Calvin(config)# router bgp 65111
```

在这个范例中，路由器 Calvin 属于 BGP 子自治系统（成员自治系统）65 111，所以本地 BGP 路由进程使用自治系统号码 65 111 开始。

第 2 步 配置联盟识别符，这是当与其他外部 BGP 邻居对等连接时使用的父自治系统的自治系统号码。

```
Calvin(config-router)# bgp confederation identifier 2610
```

BGP 联盟识别符定义了两个子自治系统 AS 65 111 和 65 112 都属于的父自治系统。

第 3 步 使用子自治系统号码作为所有内部 I-BGP 对等体的远端自治系统号码，建立全网状的 I-BGP 子自治系统邻居关系。在下面的范例中，路由器 Calvin 和它的 I-BGP 邻居路由器 Bart 和 Zim 在 BGP 子自治系统 65 111 中建立对等关系：

```
Calvin(config-router)# neighbor 10.1.1.2 remote-as 65111
Calvin(config-router)# neighbor 10.1.1.3 remote-as 65111
```

第 4 步 将在同一个父自治系统内但不是同一个联盟子自治系统的其他 BGP 邻居配置为外部邻居，指定它们的子自治系统号码作为 BGP 的远端自治系统号码。其他的不同子自治系统的联盟对等体也必须通过命令 **bgp confederation peers sub-AS number** 配置为外部的联盟对等体，如路由器 Calvin 所示：

```
Calvin(config-router)# neighbor 10.10.1.2 remote-as 65112
Calvin(config-router)# bgp confederation peers 65112
```

可以使用 **bgp confederation peers** 命令来定义多个联盟对等自治系统。这个命令有两种使用方式，每种方式的结果都一样。

- 输入 **bgp confederation peers** 命令，后面跟随用空格分开的每个联盟对等自治系统号码。
- 输入每个单独的 **bgp confederation peers member-AS number**。

第5步 使用你通常配置其他 E-BGP 对等体的方式配置所有的 E-BGP 对等体（既不属于父自治系统又不属于子自治系统的对等体）。每个外部对等体将使用父自治系统号码和每个内部的联盟对等体连接，外部的 BGP 邻居不知道 I-BGP 联盟的信息，当联盟内部对等体发送更新给外部对等体的时候所有的联盟相关信息将从 AS 路径中删除。

```
Calvin(config-router)# neighbor 82.99.34.10 remote-as 55828
```

由于路由器 Calvin 属于父自治系统 2610，它将使用本地的联盟识别符与路由器 Scrappy 建立 E-BGP 对等连接。同样，路由器 Scrappy 必须使用路由器 Calvin 的父自治系统号码（联盟识别符）来建立对等连接，因为这是路由器 Scrappy 知道的惟一的自治系统号码。

注意：当配置属于自治系统联盟的路由器的时候，总是需要注意每个对等体所属的自治系统类型。当使用联盟的时候，记住下面 3 个简单规则：

- 成员自治系统对等体（属于同一个子自治系统的对等体）只需要通常的 I-BGP 邻居定义，使用的命令是 **neighbor ip-address remote-as remote-AS-number**。
- 外部联盟对等体（属于同一个 I-BGP 父自治系统，但是在不同的成员自治系统中的对等体）需要两个步骤：使用命令 **neighbor ip-address remote-as remote-ASN** 和 **bgp confederation peers remote-AS-number** 定义一个对等体。
- 外部 BGP 对等体通过标准的 E-BGP 命令配置，但是远端 E-BGP 对等体将不会知道任何 BGP 联盟的信息，所以必须总是确认使用 **bgp confederation identifier parent-AS-number** 命令定义了父自治系统。

使用命令 **show ip bgp neighbors** 验证每个 BGP 联盟对等体的配置，这个命令将显示被共同管理的子自治系统内的每个邻居，如下所示：

```
Calvin# show ip bgp neighbors 10.1.1.2
BGP neighbor is 10.1.1.2, remote AS 65111, internal link
BGP version 4, remote router ID 10.1.1.2
Neighbor under common administration
BGP state = Established, up for 00:00:45
Last read 00:00:45, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  Route refresh: advertised and received(old & new)
  Address family IPv4 Unicast: advertised and received
Received 3 messages, 0 notifications, 0 in queue
Sent 4 messages, 0 notifications, 0 in queue
Route refresh request: received 0, sent 0
Default minimum time between advertisement runs is 5 seconds
```

现在你已经知道 BGP 联盟如何创建 sub-AS 来简化 I-BGP 配置，下面来看一个实际 BGP 联盟范例。

9.3 实际范例：BGP 联盟

65 500 和 65 501。这个范例探究了 BGP 联盟配置的一些实情，它告诉你如何进行下面这些配置：

- 在成员自治系统内部任何配置对等体。
- 配置特别的 E-BGP 类型的对等体，它们在同一个父自治系统但是不同的成员自治系统中。
- 配置联盟对等体与标准 E-BGP 对等体的相互作用。

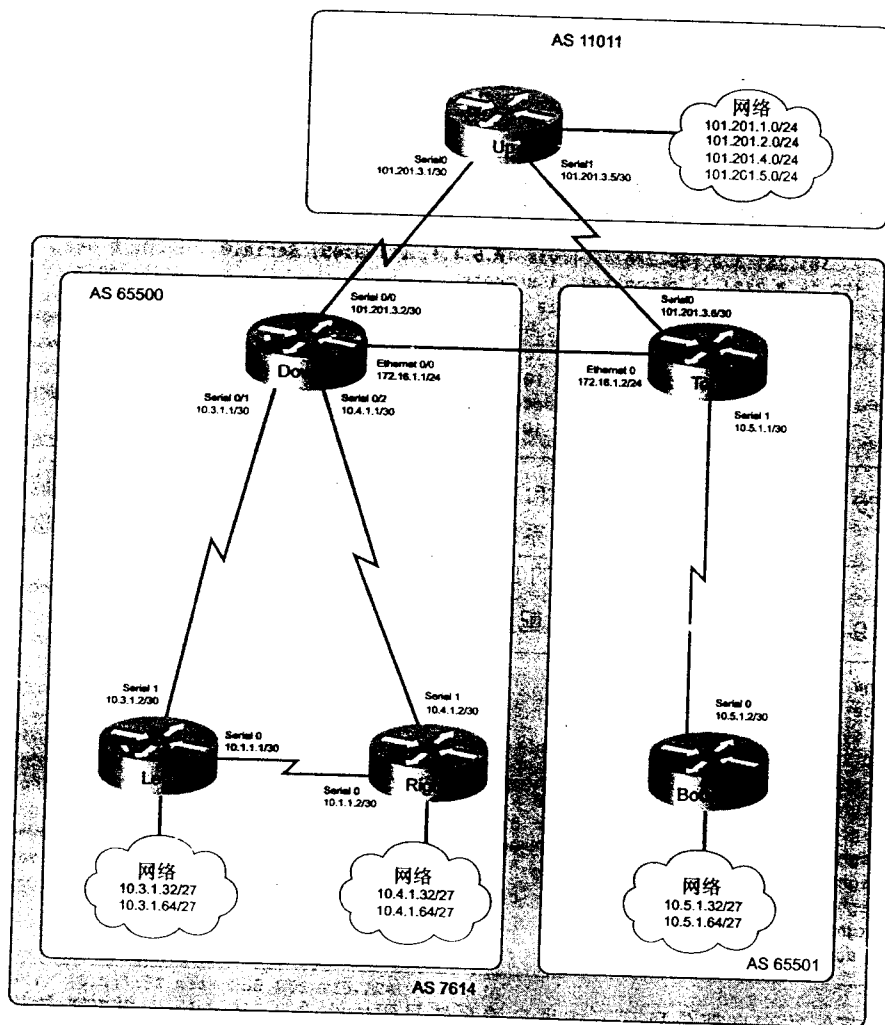


图 9-5 使用联盟来简化一个自治系统

这个范例需要 5 台思科路由器，接口要求如表 9-1 所示。

表 9-1

路由器接口要求

路由器名称	以太、快速以太或令牌环接口	串行接口	路由器名称	以太、快速以太或令牌环接口	串行接口
Up	0	2	Right	0	2
Down	1	3	Left	0	2
Top	1	2	Bottom	0	1

在配置任何路由器之前，确认它们物理上已经如图 9-5 所示连接好。这个范例需要 6 个背对背的串行电缆和两个以太网或是令牌环电缆连接到一个集线器、交换机或是多点访问单元（MSAU）。如果你使用的是交换机，所有的接口都必须放入同一个虚拟局域网中。

第 1 步 按前面的图所示配置所有的 IP 地址。将自治系统 7614 的所有成员放入 EIGRP 进程 1709 中，不要向路由器 UP 发送 EIGRP 更新。在进入第 2 步之前验证所有的接口都是在线的，而且 EIGRP 1709 中的所有路由器都可以互相 ping 通。范例 9-7 显示了结束上述步骤后路由器 Bottom 的路由表应该是怎样的。

范例 9-7 路由器 Bottom 的路由表

```
Bottom# show ip route | include islvia
 101.0.0.0/30 is subnetted, 2 subnets
D    101.201.3.4 [90/2681856] via 10.5.1.1, 00:09:45, Serial0
D    101.201.3.0 [90/2707456] via 10.5.1.1, 00:09:45, Serial0
 172.16.0.0/24 is subnetted, 1 subnets
D    172.16.1.0 [90/2195456] via 10.5.1.1, 00:09:45, Serial0
 10.0.0.0/30 is subnetted, 4 subnets
D    10.3.1.0 [90/2707456] via 10.5.1.1, 00:09:45, Serial0
D    10.1.1.0 [90/3219456] via 10.5.1.1, 00:08:53, Serial0
C    10.5.1.0 is directly connected, Serial0
D    10.4.1.0 [90/2707456] via 10.5.1.1, 00:09:46, Serial0
```

第 2 步 在路由器 Down、Right 和 Left 上配置 BGP 路由。将所有的路由器放入成员自治系统 65 500 和父自治系统 7614 中，BGP 路由不能在分类边界汇总。范例 9-8 显示了路由器 Down 上的 BGP 配置结果。

范例 9-8 路由器 Down 的 BGP 配置

```
Down# show run | begin bgp
router bgp 65500
  no synchronization
  bgp log-neighbor-changes
bgp confederation identifier 7614
 neighbor 10.3.1.2 remote-as 65500
 neighbor 10.3.1.2 route-reflector-client
 neighbor 10.3.1.2 next-hop-self
 neighbor 10.4.1.2 remote-as 65500
 neighbor 10.4.1.2 route-reflector-client
 neighbor 10.4.1.2 next-hop-self
 no auto-summary
```

前面范例的高亮部分显示了成员自治系统号码由命令 **router bgp 65500** 定义，父自治系统由命令 **bgp confederation identifier 7614** 定义。如果没有使用这个描述，那么路由器将只能参加私有自治系统 65 500 而不能成为父自治系统的一部分。命令 **next-hop-self** 将发给对等体的 BGP 下一跳属性改为本地 BGP 发言人的 IP 地址，命令 **route-reflector-client** 转发通过 I-BGP 对等会话学到的路由，这样成员自治系统 65 500 中的每个 I-BGP 路由器都将有到每个网段的两条路由。范例 9-9 显示了路由器 Right 和 Left 在第 2 步结束后的 BGP 配置。

第 3 步 在路由器 Top 和 Bottom 上配置 BGP 路由，将它们放入成员自治系统 65 501 和父自治系统 7 614 中，这些路由器也都不会自动汇总路由。范例 9-10 显示了路由器 Top 和 Bottom 的配置结果。

范例 9-9 路由器 Left 和 Right 的配置

```
Left# show run | begin bgp
router bgp 65500
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 7614
network 10.3.1.32 mask 255.255.255.224
network 10.3.1.54 mask 255.255.255.224
neighbor 10.1.1.2 remote-as 65500
neighbor 10.1.1.2 route-reflector-client
neighbor 10.1.1.2 next-hop-self
neighbor 10.3.1.1 remote-as 65500
neighbor 10.3.1.1 route-reflector-client
neighbor 10.3.1.1 next-hop-self
no auto-summary
```

```
Right# show run | begin bgp
router bgp 65500
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 7614
network 10.4.1.32 mask 255.255.255.224
network 10.4.1.64 mask 255.255.255.224
neighbor 10.1.1.1 remote-as 65500
neighbor 10.1.1.1 route-reflector-client
neighbor 10.1.1.1 next-hop-self
neighbor 10.4.1.1 remote-as 65500
neighbor 10.4.1.1 route-reflector-client
neighbor 10.4.1.1 next-hop-self
```

范例 9-10 路由器 Top 和 Bottom 的 BGP 配置

```
Top# show run | begin bgp
router bgp 65501
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 7614
neighbor 10.5.1.2 remote-as 65501
neighbor 10.5.1.2 next-hop-self
no auto-summary
```

```
Bottom# show run | begin bgp
router bgp 65501
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 7614
network 10.5.1.32 mask 255.255.255.224
network 10.5.1.64 mask 255.255.255.224
neighbor 10.5.1.1 remote-as 65501
no auto-summary
```

第 4 步 在路由器 Up、Down 和 Top 之间配置 BGP 路由。验证路由器 Up 从路由器 Down 和 Top 那里收到正确的自治系统号码，路由器 Right、Left 和 Bottom 可以访问路由器 Up 通告的路由。范例 9-11 显示了路由器 Up 上的 BGP 配置和 BGP RIB。当与联盟成员配置 E-BGP 对等关系的时候，始终记住对远端自治系统使用父自治系统号码。

范例 9-12 显示了 Down 路由器的配置结果。

范例 9-11 路由器 Up 的 BGP 配置和 BGP RIB

```
Up# show run | begin bgp
router bgp 11011
no synchronization
bgp log-neighbor-changes
network 101.201.1.0 mask 255.255.255.0
network 101.201.2.0 mask 255.255.255.0
network 101.201.4.0 mask 255.255.255.0
network 101.201.5.0 mask 255.255.255.0
neighbor 101.201.3.2 remote-as 7614
neighbor 101.201.3.6 remote-as 7614
no auto-summary
Up# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
* 10.3.1.32/27    101.201.3.6        0       0       0 7614 i
*>                101.201.3.2        0       0       0 7614 i
* 10.3.1.64/27    101.201.3.6        0       0       0 7614 i
*>                101.201.3.2        0       0       0 7614 i
* 10.4.1.32/27    101.201.3.6        0       0       0 7614 i
*>                101.201.3.2        0       0       0 7614 i
* 10.4.1.64/27    101.201.3.6        0       0       0 7614 i
*>                101.201.3.2        0       0       0 7614 i
* 10.5.1.32/27    101.201.3.2        0       0       0 7614 i
*>                101.201.3.6        0       0       0 7614 i
* 10.5.1.64/27    101.201.3.2        0       0       0 7614 i
*>                101.201.3.6        0       0       0 7614 i
*> 101.201.1.0/24  0.0.0.0            0       32768  0 i
*> 101.201.2.0/24  0.0.0.0            0       32768  0 i
*> 101.201.4.0/24  0.0.0.0            0       32768  0 i
*> 101.201.5.0/24  0.0.0.0            0       32768  0 i
```

范例 9-12 路由器 Down 的 BGP 配置和 BGP 路由表

```
Down# show run | begin bgp
router bgp 65500
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 7614
bgp confederation peers 65501
neighbor 10.3.1.2 remote-as 65500
neighbor 10.3.1.2 route-reflector-client
neighbor 10.3.1.2 next-hop-self
neighbor 10.4.1.2 remote-as 65500
neighbor 10.4.1.2 route-reflector-client
neighbor 10.4.1.2 next-hop-self
neighbor 101.201.3.1 remote-as 11011
neighbor 172.16.1.2 remote-as 65501
neighbor 172.16.1.2 next-hop-self
no auto-summary
Down# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
* i10.3.1.32/27   10.1.1.1          0      100     0 i
*>i               10.3.1.2          0      100     0 i
* i10.3.1.64/27   10.1.1.1          0      100     0 i
*>i               10.3.1.2          0      100     0 i
*>i10.4.1.32/27    10.4.1.2          0      100     0 i
* i               10.1.1.2          0      100     0 i
*>i10.4.1.64/27    10.4.1.2          0      100     0 i
* i               10.1.1.2          0      100     0 i
```

(待续)

```
*> 10.5.1.32/27 172.16.1.2 0 100 0 (65501) i
*> 10.5.1.64/27 172.16.1.2 0 100 0 (65501) i
* 101.201.1.0/24 172.16.1.2 0 100 0 (65501) 11011 i
*> 101.201.1.0/24 101.201.3.1 0 0 0 11011 i
* 101.201.2.0/24 172.16.1.2 0 100 0 (65501) 11011 i
*> 101.201.2.0/24 101.201.3.1 0 0 0 11011 i
* 101.201.4.0/24 172.16.1.2 0 100 0 (65501) 11011 i
*> 101.201.4.0/24 101.201.3.1 0 0 0 11011 i
* 101.201.5.0/24 172.16.1.2 0 100 0 (65501) 11011 i
*> 101.201.5.0/24 101.201.3.1 0 0 0 11011 i
```

为了使路由器 Down 与路由器 Top 建立特别的 E-BGP 联盟对等关系，需要有 bgp confederation peer 65501 描述。这个描述告诉路由器自治系统 65501 也是父自治系统 7614 内的一个成员自治系统。范例 9-13 显示了路由器 Top 的 BGP 配置和 show ip bgp RIB 信息。

范例 9-13 路由器 Top 的 BGP 配置和产生的 BGP RIB

```
Top# show run | begin bgp
router bgp 65501
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 7614
bgp confederation peers 65500
neighbor 10.5.1.2 remote-as 65501
neighbor 10.5.1.2 next-hop-self
neighbor 101.201.3.5 remote-as 11011
neighbor 172.16.1.1 remote-as 65500
neighbor 172.16.1.1 next-hop-self
no auto-summary
Top# show ip bgp | begin Network
Network Next Hop Metric LocPrf Weight Path
*> 10.3.1.32/27 172.16.1.1 0 100 0 (65500) i
*> 10.3.1.64/27 172.16.1.1 0 100 0 (65500) i
*> 10.4.1.32/27 172.16.1.1 0 100 0 (65500) i
*> 10.4.1.64/27 172.16.1.1 0 100 0 (65500) i
*> 10.5.1.32/27 10.5.1.2 0 100 0 i
*> 10.5.1.64/27 10.5.1.2 0 100 0 i
*> 101.201.1.0/24 101.201.3.5 0 0 0 11011 i
* 172.16.1.1 0 100 0 (65500) 11011 i
*> 101.201.2.0/24 101.201.3.5 0 0 0 11011 i
* 172.16.1.1 0 100 0 (65500) 11011 i
*> 101.201.4.0/24 101.201.3.5 0 0 0 11011 i
* 172.16.1.1 0 100 0 (65500) 11011 i
*> 101.201.5.0/24 101.201.3.5 0 0 0 11011 i
* 172.16.1.1 0 100 0 (65500) 11011 i
```

这时可以在所有的路由器上 ping 所有的接口地址。如果每个 I-BGP 发言人有始发于子自治系统 65 500 的到所有网段的两条路由，而且你也能够在每台路由器上 ping 通每个接口，你就已经完成了这个范例。范例 9-14 显示了本实验中每台路由器的完整配置。

范例 9-14 路由器的完整配置

```
Up# show run | begin int
interface Loopback100
ip address 101.201.1.1 255.255.255.0
!
```

(待续)

```
interface Loopback101
 ip address 101.201.2.1 255.255.255.0
!
interface Loopback102
 ip address 101.201.4.1 255.255.255.0
!
interface Loopback103
 ip address 101.201.5.1 255.255.255.0
!
interface Serial0
 ip address 101.201.3.1 255.255.255.252
!
interface Serial1
 ip address 101.201.3.5 255.255.255.252
!
router bgp 11011
 no synchronization
 bgp log-neighbor-changes
 network 101.201.1.0 mask 255.255.255.0
 network 101.201.2.0 mask 255.255.255.0
 network 101.201.4.0 mask 255.255.255.0
 network 101.201.5.0 mask 255.255.255.0
 neighbor 101.201.3.2 remote-as 7614
 neighbor 101.201.3.6 remote-as 7614
 no auto-summary
```

```
Down# show run | begin int
interface Ethernet0/0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0
 ip address 101.201.3.2 255.255.255.252
!
interface Serial0/1
 ip address 10.3.1.1 255.255.255.252
 clock rate 1300000
!
interface Serial0/2
 ip address 10.4.1.1 255.255.255.252
 clock rate 1300000
!
router eigrp 1709
 passive-interface Serial0/0
 network 10.3.1.0 0.0.0.3
 network 10.4.1.0 0.0.0.3
 network 101.201.3.0 0.0.0.3
 network 172.16.1.0 0.0.0.255
 no auto-summary
!
router bgp 65500
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 7614
 bgp confederation peers 65501
 neighbor 10.3.1.2 remote-as 65500
 neighbor 10.3.1.2 route-reflector-client
 neighbor 10.3.1.2 next-hop-self
 neighbor 10.4.1.2 remote-as 65500
 neighbor 10.4.1.2 route-reflector-client
 neighbor 10.4.1.2 next-hop-self
 neighbor 101.201.3.1 remote-as 11011
```

(待续)

```

neighbor 172.16.1.2 remote-as 65501
neighbor 172.16.1.2 next-hop-self
no auto-summary

```

```

Top# show run | begin int
interface Ethernet0
 ip address 172.16.1.2 255.255.255.0
!
interface Serial0
 ip address 101.201.3.6 255.255.255.252
 clockrate 1300000
!
interface Serial1
 ip address 10.5.1.1 255.255.255.252
!
router eigrp 1709
 passive-interface Serial0
 network 10.5.1.0 0.0.0.3
 network 101.201.3.4 0.0.0.3
 network 172.16.1.0 0.0.0.255
 no auto-summary
!
router bgp 65501
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 7614
 bgp confederation peers 65500
 neighbor 10.5.1.2 remote-as 65501
 neighbor 10.5.1.2 next-hop-self
 neighbor 101.201.3.5 remote-as 11011
 neighbor 172.16.1.1 remote-as 65500
 neighbor 172.16.1.1 next-hop-self
 no auto-summary

```

```

Left# show run | begin int
interface Loopback100
 ip address 10.3.1.33 255.255.255.224
!
interface Loopback200
 ip address 10.3.1.65 255.255.255.224
!
interface Serial0
 ip address 10.1.1.1 255.255.255.252
 clockrate 1300000
!
interface Serial1
 ip address 10.3.1.2 255.255.255.252
!
router eigrp 1709
 network 10.1.1.0 0.0.0.3
 network 10.3.1.0 0.0.0.3
 no auto-summary
!
router bgp 65500
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 7614
 network 10.3.1.32 mask 255.255.255.224
 network 10.3.1.64 mask 255.255.255.224
 neighbor 10.1.1.2 remote-as 65500
 neighbor 10.1.1.2 route-reflector-client
 neighbor 10.1.1.2 next-hop-self

```

(待续)

```
neighbor 10.3.1.1 remote-as 65500
neighbor 10.3.1.1 route-reflector-client
neighbor 10.3.1.1 next-hop-self
no auto-summary
```

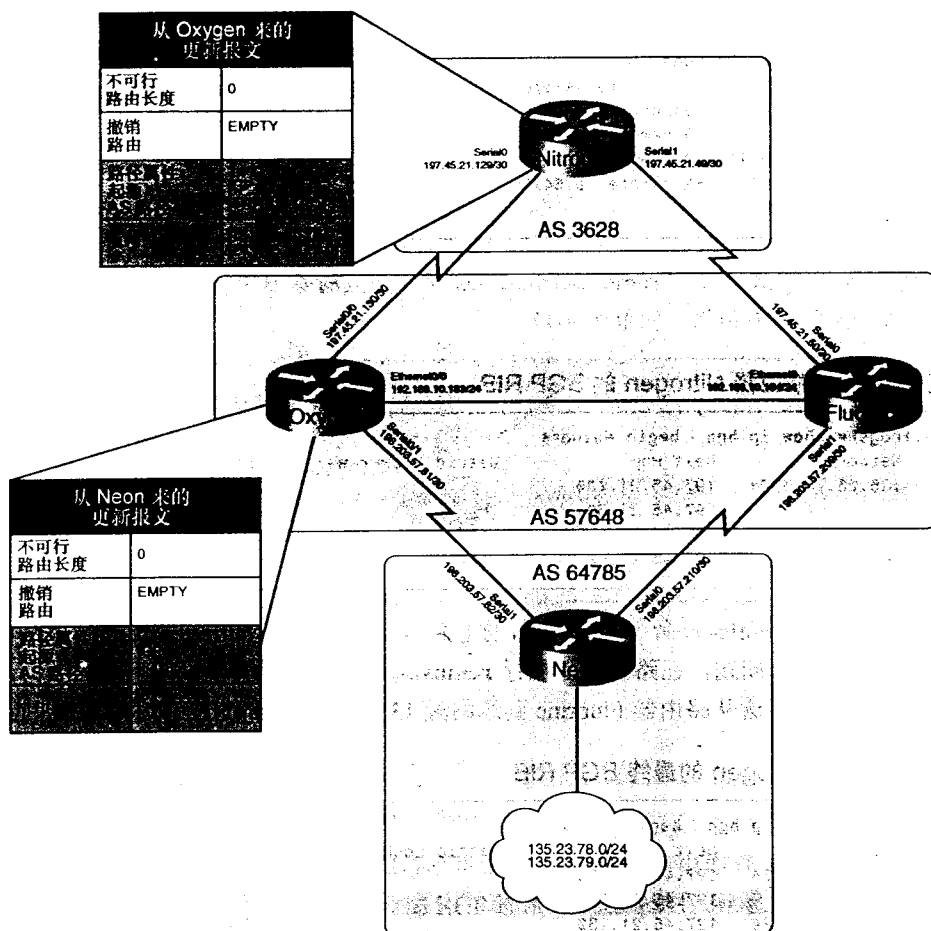
```
Right# show run | begin int
interface Loopback100
 ip address 10.4.1.33 255.255.255.224
!
interface Loopback200
 ip address 10.4.1.65 255.255.255.224
!
interface Serial0
 ip address 10.1.1.2 255.255.255.252
!
interface Serial1
 ip address 10.4.1.2 255.255.255.252
!
router eigrp 1709
 network 10.1.1.0 0.0.0.3
 network 10.4.1.0 0.0.0.3
 no auto-summary
!
router bgp 65500
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 7614
 network 10.4.1.32 mask 255.255.255.224
 network 10.4.1.64 mask 255.255.255.224
 neighbor 10.1.1.1 remote-as 65500
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 next-hop-self
 neighbor 10.4.1.1 remote-as 65500
 neighbor 10.4.1.1 route-reflector-client
 neighbor 10.4.1.1 next-hop-self
 no auto-summary
```

```
Bottom# show run | begin int
interface Loopback100
 ip address 10.5.1.33 255.255.255.224
!
interface Loopback200
 ip address 10.5.1.65 255.255.255.224
!
interface Serial0
 ip address 10.5.1.2 255.255.255.252
 clockrate 1300000
!
router eigrp 1709
 network 10.5.1.0 0.0.0.3
 no auto-summary
!
router bgp 65501
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 7614
 network 10.5.1.32 mask 255.255.255.224
 network 10.5.1.65 mask 255.255.255.224
 neighbor 10.5.1.1 remote-as 65501
 no auto-summary
```

9.3.1 私有自治系统

与 RFC 1918 的私有 IP 地址类似，一些自治系统号码也被保留给不需要公共自治系统号码的网络使用。私有自治系统（范围从 64 512~65 535）通常有两种使用方式：它们可以在两个私有的 BGP 网络之间，或是作为 BGP 联盟的成员自治系统。如果你回想一下第 7 章，你会记得 BGP 联盟的默认行为是 AS 路径在通告给 E-BGP 邻居之前成员自治系统号码会被事先删除。尽管当路由离开自治系统之前不需要手工在联盟成员上删除私有自治系统号码（路由器会帮你做），但是在私有的 BGP 网络的情况下当向公共互联网通告路由时还是需要删除私有的自治系统号码。

可以在路径通告给外部对等体之前在自治系统的出口将私有自治系统号码从 AS 路径中删除。需要为每个配置的 E-BGP 对等体使用命令 **neighbor ip-address remove-private-as** 来删除私有自治系统。以图 9-6 所示的网络为例，注意到路由器 Neon 通告网段 135.23.78.0/24 和 135.23.79.0/24 的路由给自治系统 57 548 中的路由器，AS 路径为 64 785。



上游的路由器 Nitrogen 接收了这些网段的更新，其中 AS 路径的值为[57648, 64785]。为了从路径中删除自治系统 64785，在路由器 Oxygen 和 Fluorine 的 E-BGP 邻居配置上加上了命令 **remove-private-as**，还需要在这些路由器上清除 BGP 以使改变生效。在删除私有自治系统号码之前，路由器 Nitrogen 上的 BGP RIB 如范例 9-15 所示。

范例 9-15 Nitrogen BGP RIB

```
Nitrogen# show ip bgp | begin Network
Network      Next Hop      Metric LocPrf Weight Path
* 135.23.78.0/24 197.45.21.130      0 57648 64785 i
*>           197.45.21.50      0 57648 64785 i
* 135.23.79.0/24 197.45.21.130      0 57648 64785 i
*>           197.45.21.50      0 57648 64785 i
```

范例 9-16 显示了使用 **remove-private-as** 命令以后路由器 Oxygen 的 BGP 配置。

范例 9-16 在路由器 Oxygen 上使用 remove-private-as 命令

```
Oxygen# show run | begin bgp
router bgp 57648
no synchronization
bgp log-neighbor-changes
neighbor 192.168.10.184 remote-as 57648
neighbor 192.168.10.184 next-hop-self
neighbor 197.45.21.129 remote-as 3628
neighbor 197.45.21.129 remove-private-as
neighbor 198.203.57.82 remote-as 64785
no auto-summary
```

在路由器 Oxygen 上加上 **remove-private-as** 命令并且清除 BGP 会话后，私有自治系统号码 64785 从 AS 路径中删除，如范例 9-17 所示。

范例 9-17 路由器 Nitrogen 的 BGP RIB

```
Nitrogen# show ip bgp | begin Network
Network      Next Hop      Metric LocPrf Weight Path
*> 135.23.78.0/24 197.45.21.130      0 57648 i
*             197.45.21.50      0 57648 64785 I
*> 135.23.79.0/24 197.45.21.130      0 57648 i
*             197.45.21.50      0 57648 64785 I
```

现在 **remove-private-as** 命令已经执行，你也能够看到路由器 Nitrogen 优选最短路径的新路由。为了解决这个问题，在路由器上执行 **remove-private-as** 命令，清除 BGP 会话，路由器 Nitrogen 将再次优选从路由器 Fluorine 到达网段 135.23.78.0/24 和 135.23.79.0/24 的路径。

范例 9-18 Nitrogen 的最终 BGP RIB

```
Nitrogen# show ip bgp | begin Network
Network      Next Hop      Metric LocPrf Weight Path
* 135.23.78.0/24 197.45.21.130      0 57648 i
*>           197.45.21.50      0 57648 i
* 135.23.79.0/24 197.45.21.130      0 57648 i
*>           197.45.21.50      0 57648 i
```

9.3.2 使用对等体组简化配置

在很多高级的 BGP 实现中会出现很大很复杂的配置。针对你配置的每个单独的对等体，你可能需要一个 **neighbor** 描述、**next-hop-self** 描述、路由过滤、路由聚合、属性修改等等，这使配置变得极其复杂和难以阅读。解决这个问题方法就是使用 BGP 对等体组。

在思科 IOS 软件中，与 BGP 共同使用的 BGP 对等体组主要是用于简化 BGP 配置，可以将那些需要不断重复的配置命令合并到一个或多个对等体组中。每个邻居都被分配一个对等体组，同一个对等体组中路由器的配置一致。

创建一个对等体组需要 3 个步骤：

第 1 步 使用命令 **neighbor peer-group-name peer-group** 来创建对等体组。

第 2 步 使用命令 **neighbor peer-group-name** 给对等体加上你希望的组范围的配置成员。

第 3 步 使用命令 **neighbor ip-address peer-group** 将具有共同特性的 BGP 对等体放入对等体组中。

图 9-7 中显示的网络是使用对等体组的一个很好的范例。

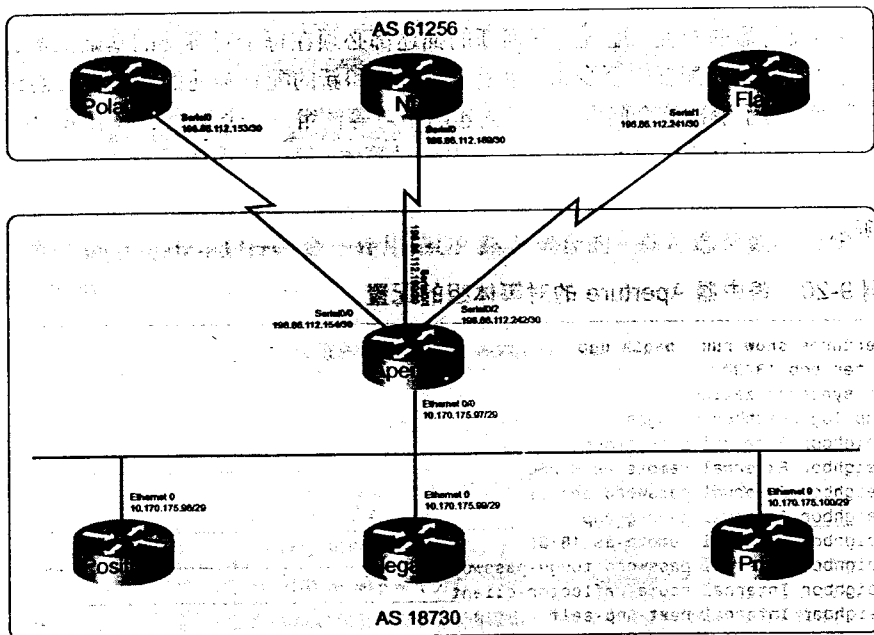


图 9-7 Shadow-Box 网络

在这个范例中，路由器 Aperture 和自治系统 61 256 中的路由器有 3 个外部 BGP 连接，和自治系统 18 730 中的对等体有 3 个内部 BGP 连接。如范例 9-19 所示，每个 BGP 对等会话都有相同的基本配置。

范例 9-19 路由器 Aperture 的 BGP 配置

```
Aperture# show run | begin bgp
router bgp 18730
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.170.175.98 remote-as 18730
  neighbor 10.170.175.98 password tough-password
  neighbor 10.170.175.98 route-reflector-client
  neighbor 10.170.175.98 next-hop-self
  neighbor 10.170.175.99 remote-as 18730
  neighbor 10.170.175.99 password tough-password
  neighbor 10.170.175.99 route-reflector-client
  neighbor 10.170.175.99 next-hop-self
  neighbor 10.170.175.100 remote-as 18730
  neighbor 10.170.175.100 password tough-password
  neighbor 10.170.175.100 route-reflector-client
  neighbor 10.170.175.100 next-hop-self
  neighbor 196.86.112.153 remote-as 61256
  neighbor 196.86.112.153 password secret
  neighbor 196.86.112.189 remote-as 61256
  neighbor 196.86.112.189 password secret
  neighbor 196.86.112.241 remote-as 61256
  neighbor 196.86.112.241 password secret
  no auto-summary
```

前面的配置显示了路由器 Aperture 为 6 个 BGP 会话配置了 18 行命令。每个外部会话有一个远端自治系统和密码的配置，每个内部会话有一个 **remote-as**、**password**、**next-hop-self** 描述，以及路由反射器的配置。配置上任何新的描述都必须在每个对等体的基础上增加，增加任何新的对等体都需要在配置中至少添加两行。对等体和新的配置描述的组合将会创建一个很冗长乏味的配置，为了解决这个问题，可以创建两个对等体组，一个是为了自治系统 61 256 的外部对等体，另一个是为了自治系统 18 730 的内部对等体。每个对等体组的每个配置描述被加入对等体组的配置，当这些组配置完成后，每个外部或是内部的邻居只需要一行即可配置，如范例 9-20 所示。

范例 9-20 路由器 Aperture 的对等体组的配置

```
Aperture# show run | begin bgp
router bgp 18730
  no synchronization
  bgp log-neighbor-changes
  neighbor External peer-group
  neighbor External remote-as 61256
  neighbor External password secret
  neighbor Internal peer-group
  neighbor Internal remote-as 18730
  neighbor Internal password tough-password
  neighbor Internal route-reflector-client
  neighbor Internal next-hop-self
  neighbor 10.170.175.98 peer-group Internal
  neighbor 10.170.175.99 peer-group Internal
  neighbor 10.170.175.100 peer-group Internal
  neighbor 196.86.112.153 peer-group External
  neighbor 196.86.112.189 peer-group External
  neighbor 196.86.112.241 peer-group External
  no auto-summary
```

现在你已经看到了如何使用路由反射器、联盟和对等体组来简化大型网络的实现，下面来学习如何使用 BGP 路由聚合技术来简化路由表。

9.4 路由聚合

另外一个简化大型 BGP 实现的方法是通过聚合 BGP 路由来减小 BGP RIB 的大小。路由聚合是一个可以帮助保持因特网路由表的大小，减少更新的时候在相邻的 BGP 路由器之间传递的路由数量的简单方法。本节将介绍下面的路由聚合配置：

- 常规的路由聚合；
- 使用过滤的路由聚合；
- 路由抑制；
- 条件路由通告。

默认的情况下如果有更精确的路由在主 IP 路由表中存在，BGP 将只会通告聚合路由。如果你为一些路由的集合指定了聚合路由，但是 BGP 扫描器不知道这些路由，那么聚合路由将不会被通告。默认的情况下聚合路由丢失了更精确的个体路由的属性值，然而你可以将包含路由列表和属性的路由映射应用在聚合路由上来改变这种行为。使用路由聚合来控制 BGP 更新流量是一个简单的易于配置的过程，它只需要三步：

第 1 步 使用 **network** 命令指定要被聚合的网段。

第 2 步 使用 **aggregate-address** 命令指明网段应该被汇总的方式。在思科 IOS 软件版本 12.2 (12) T 中 **aggregate-address** 命令的语法如下：

```
aggregate-address ip-address aggregate-mask [advertise-map route-map-name ]
[as-set] [attribute-map route-map-name] [route-map route-map-name] [summary-
only] [suppress-map route-map-name]
```

第 3 步 （可选）指明将要使用的任意附加聚合配置。

可以通过 **aggregate-address** 命令使用 BGP 路由聚合的一些可选参数，表 9-2 显示了这些可选命令的值以及它们的描述。

表 9-2 可选的 aggregate-address 命令

命令名	描述
advertise-map	指明了包含将被应用一个 AS_SET 属性的路由列表的路由映射，这个命令也可以用来指定将要被聚合的路由
as-set	为聚合路由创建一个 AS_SET 属性，当路径包含不同的 AS 路径值的时候在其中存放 AS 路径的聚合子集
attribute-map	允许根据用户定义的信息客户化定义 BGP 属性
route-map	与 attribute-map 命令类似，这个命令允许控制聚合属性
summary-only	限制 BGP 仅仅通告输出聚合路由——过滤所有用来创建聚合的单独路由
suppress-map	根据用户在路由映射中定义的信息指明需要被抑制的更精确的路由

当一个聚合路由被创建后，新的聚合路由和所有其他更精确的路由都被通告给每个 BGP 对等体。如果这不是你希望的效果，可以使用命令 **summary-only** 来控制这种行为。新的路由默认包括原子聚合和聚合者属性，原子聚合属性说明了路由已经被聚合而且精确路由的路

径属性已经丢失。聚合者属性提供了最初聚合路由的路由器的信息。

要保持被聚合的路径的 AS 路径属性，一种可能的方法是在聚合点使用 **as-set** 命令，它将在包含聚合路由信息的 UPDATE 报文的 AS 路径字段创建路径段落类型 **AS_SET**。

范例 9-21 显示了路由聚合如何将如图 9-8 所示的路由器 Day 和 Night 之间的网段 156.202.148.x 汇总成一个聚合网段 156.202.148.0/24。

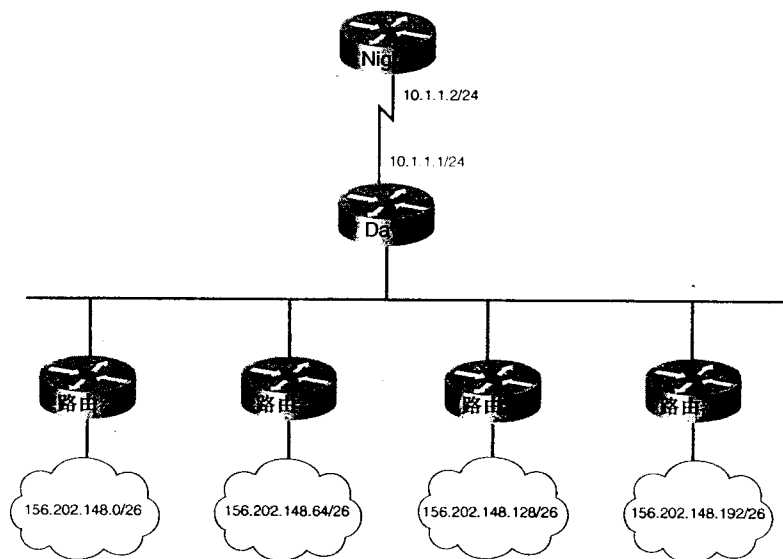


图 9-8 Day 和 Night 网络

范例 9-21 聚合路由和过滤精确路由

```
Day# show run | begin bgp
router bgp 8
  bgp log-neighbor-changes
  network 10.1.1.0 mask 255.255.255.0
  network 156.202.148.0 mask 255.255.255.192
  network 156.202.148.64 mask 255.255.255.192
  network 156.202.148.128 mask 255.255.255.192
  network 156.202.148.192 mask 255.255.255.192
  aggregate-address 156.202.148.0 255.255.255.0 summary-only
  neighbor 10.1.1.2 remote-as 9
```

在这个范例中，**aggregate-address** 命令将 4 个 156.202.148.0/26 网段聚合成一个 156.202.148.0/24 汇总路由，**summary-only** 描述告诉路由器抑制创建汇总路由的单独路由，只向远端对等体通告 156.202.148.0/24 网段。为了验证这个命令工作正常，可以在路由器 Day 上使用 **show ip bgp** 和 **show ip bgp neighbors 10.1.1.2 advertised-routes** 命令，如范例 9-22 所示。

注意聚合网段 156.202.148.0/24 的精确路由的子网掩码是/26，在前例中高亮显示，它们带有的 **>** 字符说明是被抑制的路由；聚合路由 156.202.148.0/24 显示带有 ***** 字符，表明它是最有效的路由。同时注意当使用 **show ip bgp neighbors 10.1.1.2 advertised-routes** 命令后，你会看到路由器只是通告了 156.202.148.0/24 汇总网段。范例 9-23 显示了关于 156.202.148.0/24 网段的特定的 BGP 信息。

范例 9-22 路由器 Day 上 show ip bgp 命令的输出

```
Day# show ip bgp | begin Network
  Network      Next Hop      Metric LocPrf Weight Path
s> 156.202.148.0/26 0.0.0.0      0          32768 i
*> 156.202.148.0/24 0.0.0.0      0          32768 i
s> 156.202.148.64/26
      0.0.0.0      0          32768 i
s> 156.202.148.128/26
      0.0.0.0      0          32768 i
s> 156.202.148.192/26
      0.0.0.0      0          32768 i
Day# show ip bgp neighbors 10.1.1.2 advertised-routes | begin Network
  Network      Next Hop      Metric LocPrf Weight Path
*> 10.1.1.0/24  0.0.0.0      0          32768 i
*> 156.202.148.0/24 0.0.0.0      0          32768 i
```

范例 9-23 路由器 Day 上 show ip bgp 156.202.148.0/24 命令的输出

```
Day# show ip bgp 156.202.148.0/24
BGP routing table entry for 156.202.148.0/24, version 7
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
    10.1.1.2
  Local, (aggregated by 8 10.1.1.1)
    0.0.0.0 from 0.0.0.0 (10.1.1.1)
      Origin IGP, localpref 100, weight 32768, valid, aggregated, local,
      atomic-aggregate, best
```

注意，网段 156.202.148.0/24 的路由包含了聚合和原子聚合属性，说明是路由器 Day（自治系统 8 中的 10.1.1.1）聚合了路由，在聚合过程中路由的路径信息可能发生了丢失。as-set 参数也可以和 aggregate-address 命令一起使用来存储路由的 AS_SET 路径信息。图 9-9 的范

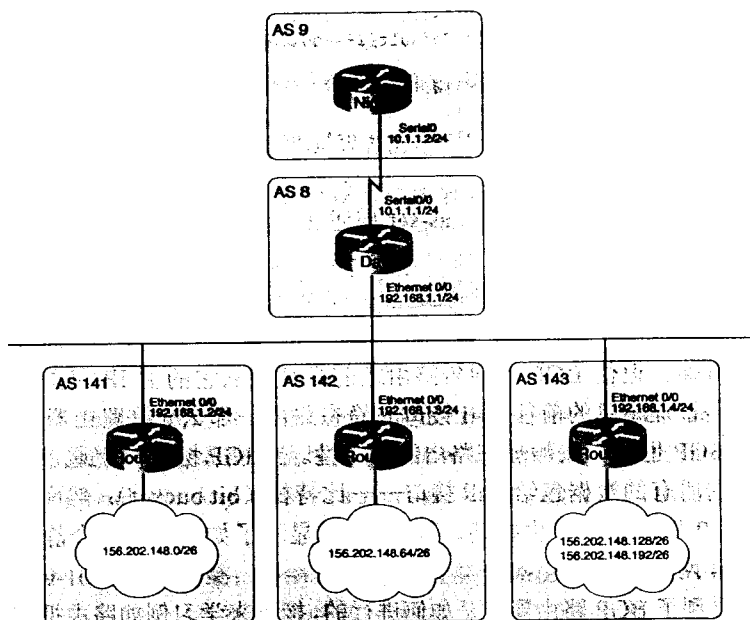


图 9-9 Day 和 Night 网络详图

例中路由器 Day 将 156.202.148.0/26 网段聚合成更大的 156.202.148.0/24 网段通告给路由器 Night。在这个范例中，每个 156.202.148.0/26 网段来源于不同的自治系统。**as-set** 关键词可以和 **aggregate-address** 命令一起使用，将在聚合过程中被删除的单独的自治系统号码列表加入到聚合路由的 AS 路径属性中。范例 9-24 显示了当 **as-set** 关键词使用之前路由器 Night 上 156.202.148.0/24 网段的 BGP RIB 记录，范例 9-25 显示了路由器 Day 上的配置修改以及在路由器 Night 上导致的 BGP 路由变化。

范例 9-24 路由器 Night 的 156.202.148.0/24 网段的 BGP 记录（修改前）

```
Night# show ip bgp 156.202.148.0/24
BGP routing table entry for 156.202.148.0/24, version 13
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  8, (aggregated by 8 10.1.1.1)
    10.1.1.1 from 10.1.1.1 (10.1.1.1)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
```

范例 9-25 使用 AS_SET 值保留单独的 AS 路径值

```
Day# show run | begin bgp
router bgp 8
no synchronization
bgp log-neighbor-changes
aggregate-address 156.202.148.0 255.255.255.0 summary-only
neighbor 10.1.1.2 remote-as 9
neighbor 192.168.1.2 remote-as 141
neighbor 192.168.1.3 remote-as 142
neighbor 192.168.1.4 remote-as 143

Night# show ip bgp 156.202.148.0/24
BGP routing table entry for 156.202.148.0/24, version 18
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  8 {141,142,143}, (aggregated by 8 10.1.1.1)
    10.1.1.1 from 10.1.1.1 (10.1.1.1)
      Origin IGP, localpref 100, valid, external, best
```

当在 **aggregate-address** 命令上加入 **as-set** 描述并且清除 BGP 会话后，路由器 Night 上将显示关于 156.202.148.0/24 路由的更详细的 AS 路径记录。该路由现在在 AS 路径属性中列出了自治系统号码 141、142 和 143，该列表通常被称为 AS_SET。

可以使用高管理距离的前往 null 接口的静态路由来防止由于被聚合的单独网段的不稳定而导致的路由振荡。记住 BGP 在通告路由之前必须先从它的主 IP 路由表中学到这条路由。如果你使用高管理距离的前往 null 接口的静态路由，那么允许路由器优选从 IGP 协议学到的路由，而 BGP 也可以依赖静态路由的稳定性。当 IGP 协议停止通告路由的时候，路由器将会开始发送所有的数据包给 null 接口——比特桶 (bit bucket)；然而，路由器通告给上游路由器的 BGP 路由不会发生抖动。范例 9-26 显示了如何使用一个静态路由来帮助聚合 189.28.145.0/24 网段。

现在你已经看到了 BGP 路由聚合是如何进行的，接下来学习例如路由抑制和条件通告等更高级的 BGP 路由聚合和通告方式。

范例 9-26 为了路由稳定使用前往 null 接口的静态路由

```
Doh# show run | begin bgp
router bgp 104
  no synchronization
  bgp router-id 10.1.1.1
  bgp log-neighbor-changes
  network 189.28.145.0 mask 255.255.255.128
  network 189.28.145.128 mask 255.255.255.128
  aggregate-address 189.28.145.0 255.255.255.0 summary-only
  neighbor 10.1.1.2 remote-as 9
  no auto-summary
!
ip route 189.28.145.0 255.255.255.128 Null0 253 permanent
ip route 189.28.145.128 255.255.255.128 Null0 253 permanent
```

```
Doh# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
s> 189.28.145.0/25 0.0.0.0           0           32768 i
*> 189.28.145.0/24 0.0.0.0           32768 i
s> 189.28.145.128/25
                   0.0.0.0           0           32768 i
```

9.4.1 聚合与路由抑制

另外一个控制聚合路由通告的方式是使用路由抑制来抑制特定网段的通告。被抑制的路由也可以在邻居对邻居的基础上解除抑制，可以和命令 **aggregate-address** 配合使用可选的 **summary-only** 以抑制所有的更精确的路由，可以使用 **suppress maps** 和 **unsuppress maps** 来指定哪些路由应该或是不应该被抑制。通过在路由聚合中使用路由抑制，能够从聚合的路由通告中过滤特定的更长的前缀。

在路由聚合中使用路由抑制需要 4 个步骤：

- 第 1 步 启用 BGP，配置邻居关系以及将要通告的网段，需要的话使用 **no auto-summary** 命令关闭分类路由汇总。
- 第 2 步 使用一个访问列表或是前缀列表来指定要被抑制的网段。
- 第 3 步 创建一个路由映射用作聚合网段的抑制映射，这个路由映射应该指定访问列表或是前缀列表，告诉路由器要抑制哪些前缀。
- 第 4 步 使用带有 **suppress-map** 描述的 **aggregate-address** 命令来配置路由聚合，指明聚合和抑制路由。在路由聚合中使用路由抑制的命令结构如下：

```
aggregate-address ip-prefix mask [suppress-map route-map-name]
```

使用命令 **show ip bgp** 或是 **show ip bgp neighbors neighbor-address advertised-routes** 验证属于聚合路由范围的较长前缀被正确地抑制。**show ip bgp** 在被抑制的路由的状态字段显示字符 **s>**，**show ip bgp neighbors ip-address advertised-routes** 命令只显示实际通告给指定的邻居的路由。

考虑例如图 9-10 所示的网络，路由器 Rainier 连接到路由器 Adams 和 Vernon，在它的每个通告中发送两个前缀，一个是到网段 194.69.12.0/22 的会聚路由，另外一个的是到 194.69.14.0/24 网段的更精确的路由。

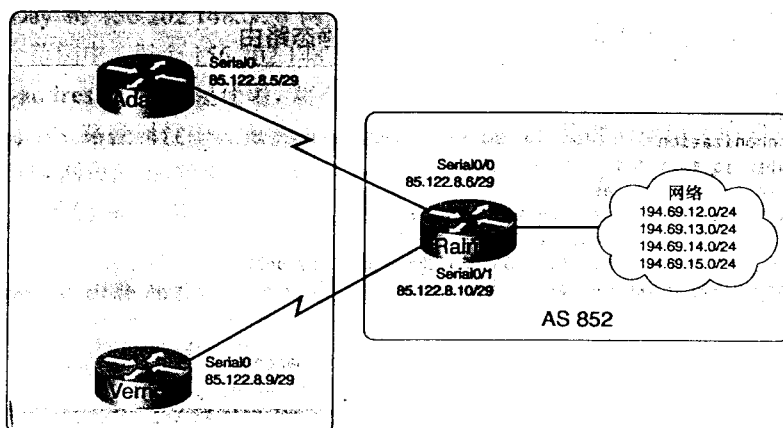


图 9-10 路由抑制和 Volcano 网络

范例 9-27 显示了命令 **aggregate-address** 是如何指定前缀 194.69.12.0/22 的，抑制映射 **hide-me** 指明了任何包含在 IP 前缀列表 10 中的网段都会被抑制，但是路由器 Rainier 仍然通告到网段 194.69.14.0/24 的更精确的路由。当你想只通告一个汇总路由和一些特别的精确路由时，可以使用 **suppress-map** 命令。

范例 9-27 使用了 **summary only** 描述和一个抑制映射

```
Rainier# show run | begin bgp
router bgp 852
 no synchronization
 bgp log-neighbor-changes
 network 194.69.12.0
 network 194.69.13.0
 network 194.69.14.0
 network 194.69.15.0
 aggregate-address 194.69.12.0 255.255.252.0 suppress-map hide-me
 neighbor 85.122.8.5 remote-as 7518
 neighbor 85.122.8.5 description Adams Peer
 neighbor 85.122.8.9 remote-as 7518
 neighbor 85.122.8.9 description Vernon Peer
 no auto-summary
!
ip prefix-list 10 seq 5 permit 194.69.12.0/24
ip prefix-list 10 seq 10 permit 194.69.13.0/24
ip prefix-list 10 seq 15 permit 194.69.15.0/24
!
route-map hide-me permit 10
 match ip address prefix-list 10
```

在范例 9-28 中，注意路由器 Rainier 的 BGP RIB 包含了 3 个被抑制的路由和两个有效的最优路由，这个结果是通过前面范例 9-27 中使用带有 **suppress-map** 描述的路由聚合来实现的。

为了使路由器对特定的对等体抑制某些路由，但是对其他对等体仍然通告这些路由，可以使用命令 **neighbor ip-address unsuppress-map route-map-name** 命令。范例 9-29 显示了如何使用这条命令将所有精确的 194.69.x.0 路由通告给路由器 Vernon，同时仍然对路由器 Adams 使用路由抑制。

范例 9-28 路由器 Rainier 的 BGP RIB

```
Rainier# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
s> 194.69.12.0    0.0.0.0           0          32768 i
*> 194.69.12.0/22 0.0.0.0           0          32768 i
s> 194.69.13.0    0.0.0.0           0          32768 i
*> 194.69.14.0    0.0.0.0           0          32768 i
s> 194.69.15.0    0.0.0.0           0          32768 i
```

范例 9-29 使用 unsuppress-map 来释放前面抑制的路由

```
Rainier# show run | begin bgp
router bgp 852
no synchronization
bgp log-neighbor-changes
network 194.69.12.0
network 194.69.13.0
network 194.69.14.0
network 194.69.15.0
aggregate-address 194.69.12.0 255.255.252.0 suppress-map hide-me
neighbor 85.122.8.5 remote-as 7518
neighbor 85.122.8.5 description Adams Peer
neighbor 85.122.8.9 remote-as 7518
neighbor 85.122.8.9 description Vernon Peer
neighbor 85.122.8.9 unsuppress-map hide-me
no auto-summary
!
ip prefix-list 10 seq 5 permit 194.69.12.0/24
ip prefix-list 10 seq 10 permit 194.69.13.0/24
ip prefix-list 10 seq 15 permit 194.69.15.0/24
!
route-map hide-me permit 10
match ip address prefix-list 10
```

在前面的范例中，曾经被用作 **hide-me suppress map** 的 **hide-me unsuppress map** 说明了在 IP 前缀列表 10 中的所有路由将不会对邻居路由器 Vernon 85.122.8.9 抑制，可以在路由器 Vernon 上使用命令 **show ip bgp** 来验证。范例 9-30 显示了路由器 Vernon 和 Adams 上产生的 BGP 表。

范例 9-30 路由器 Rainier 通告给路由器 Vernon 和 Adams 的路由

```
Vernon# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*> 194.69.12.0    85.122.8.10       0          0 852 i
*> 194.69.12.0/22 85.122.8.10       0          0 852 i
*> 194.69.13.0    85.122.8.10       0          0 852 i
*> 194.69.14.0    85.122.8.10       0          0 852 i
*> 194.69.15.0    85.122.8.10       0          0 852 i

Adams# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*> 194.69.12.0/22 85.122.8.6        0          0 852 i
*> 194.69.14.0    85.122.8.6        0          0 852 i
```

现在你已经看到了如何使用路由抑制来抑制或是在邻居的基础上释放路由，下一小节将

探究如何使用条件路由通告来有条件地向 BGP 邻居通告路由。

9.4.2 条件路由通告

条件路由通告提供了允许对路由通告进行更多控制的用户定义路由通告方式，条件路由通告使你可以通过称为 **non-exist-map** 的路由映射指定一系列的条件来跟踪某个路由的状态。如果该路由不存在，通告由另外一个称为 **advertise-map** 的路由映射指定的路由。当使用 **aggregate-address** 命令来指定当路由由聚合的时候应该包含 AS 路径属性是 **AS_SET** 的路由时，通告映射可以用来提供条件路由通告或是作为一个条件来通告会聚路由。

non-exist-map 指定了在 BGP RIB 中需要跟踪的网段。当 **non-exist-map** 中的路由存在的时候，**advertise-map** 中指定的路由将不会被通告。如果 **non-exist-map** 中指定的路由被撤消，那么在 **non-exist-map** 中指定的路由重新出现之前 **advertise-map** 中指定的路由将会被通告。条件路由通告可以用在多归路的网络中以防止不对称的路由，或者单独使用来提供额外的路由功能。

配置条件路由通告需要以下四步：

- 第 1 步 在路由通告需要涉及的路由器上配置 BGP 对等连接。
- 第 2 步 使用标准的路由映射描述创建 **non-exist-map**。这个路由映射应该使用一个访问列表或是前缀列表来指明被跟踪的网段前缀，确认配置路由映射中指定的访问列表或是前缀列表。
- 第 3 步 使用标准的路由映射描述创建 **advertise-map**，这个路由映射应该使用一个访问列表或是前缀列表来指明当 **non-exist-map** 指定的网段从 BGP RIB 中撤消时需要通告的网段前缀。而且，创建访问列表或是前缀列表来指定应该被通告的前缀。
- 第 4 步 使用命令 **neighbor ip-address advertise-map route-map-name non-exist-map route-map-name** 将路由映射应用到 BGP 邻居上。

在图 9-11 的范例中，路由器 Speedy 通过以太网连接到路由器 Tom 和 Jerry 上。路由器 Tom 通告网段 129.40.18.0/24，路由器 Jerry 通告网段 129.40.20.0/24，路由器 Speedy 将这两个网段通告给自治系统 714 中的路由器 Tweety。

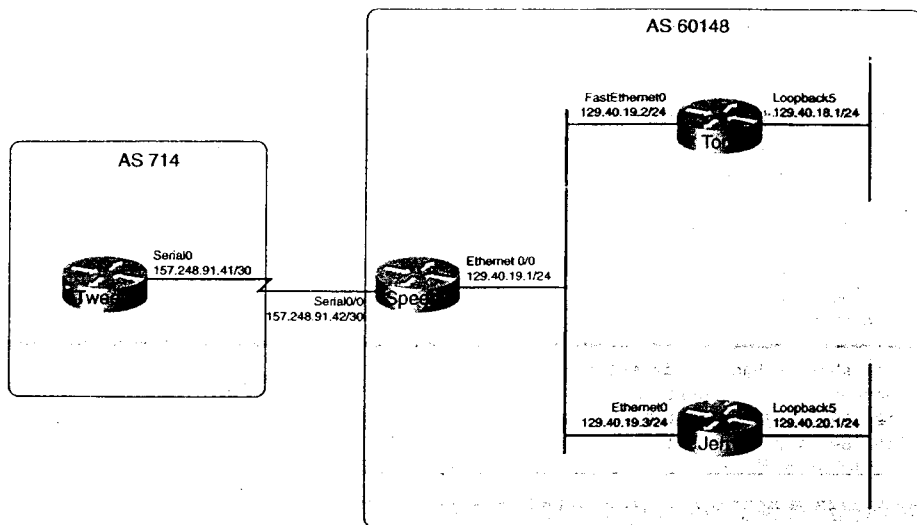


图 9-11 Cartoon 网络中的条件路由通告

范例 9-31 显示了路由器 Tweety 的 BGP RIB 记录。在这个范例中，路由器 Tweety 从路由器 Speedy 收到了所有路由（129.40.18.0/24、129.40.19.0/24 和 129.40.20.0/24）。

范例 9-31 路由器 Tweety 的 BGP RIB

```
Tweety# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*> 129.40.18.0/24  157.248.91.42          0          0 60148 i
*> 129.40.19.0/24  157.248.91.42          0          0 60148 i
*> 129.40.20.0/24  157.248.91.42          0          0 60148 i
```

范例 9-32 显示了一个条件路由通告如何控制路由器 Speedy 通告的路由。

范例 9-32 路由器 Speedy 上的条件路由通告

```
Speedy# show ip bgp | begin bgp
router bgp 60148
no synchronization
network 129.40.19.0 mask 255.255.255.0
neighbor 129.40.19.2 remote-as 60148
neighbor 129.40.19.2 description Tom Router
neighbor 129.40.19.3 remote-as 60148
neighbor 129.40.19.3 description Jerry Router
neighbor 157.248.91.41 remote-as 714
neighbor 157.248.91.41 description Tweety Router
neighbor 157.248.91.41 advertise-map advertise-me non-exist-map not-in-table
no auto-summary
!
ip prefix-list 1000 seq 5 permit 129.40.18.0/24
!
ip prefix-list 1001 seq 5 permit 129.40.20.0/24
!
route-map not-in-table permit 10
match ip address prefix-list 1001
!
route-map advertise-me permit 10
match ip address prefix-list 1000
```

路由映射 not-in-table 作为 non-exist-map 指定了网段 129.40.20.0/24，路由映射 advertise-me 用来指定有条件地通告的网段。只要路由 129.40.20.0/24 在路由器 Speedy 的路由表中存在，路由 129.40.18.0/24 将不会被通告。如果 129.40.20.0/24 路由被撤消，129.40.18.0/24 路由将会代替它被通告。范例 9-33 显示了加上条件路由通告配置后路由器 Tweety 的路由表。

范例 9-33 条件路由后路由器 Tweety 的路由表

```
Tweety# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*> 129.40.19.0/24  157.248.91.42          0          0 60148 i
*> 129.40.20.0/24  157.248.91.42          0          0 60148 i
```

在前面的范例中你可以看到，路由器 Speedy 被配置为有条件地通告网段 129.40.18.0/24 后，它开始抑制对网段 129.40.18.0/24 的通告。如果路由器 Jerry 停止通告网段 129.40.20.0/24，路由器 Speedy 将撤消 129.40.20.0/24 网段的通告而开始通告替代的 129.40.18.0/24 网段。范例

9-34 显示了当路由器 Jerry 上的环回接口 5 被停用后路由器 Speedy 有条件地路由 129.40.18.0/24 网段。

范例 9-34 有条件地通告 129.40.18.0/24 网段

Jerry(config)# interface loopback 5					
Jerry(config-if)# shutdown					
Speedy# show ip bgp begin Network					
Network	Next Hop	Metric	LocPrf	Weight	Path
*>129.40.18.0/24	129.40.19.2	0	100	0	i
*> 129.40.19.0/24	0.0.0.0	0		32768	i
Tweety# show ip bgp begin Network					
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 129.40.18.0/24	157.248.91.42				0 60148 i
*> 129.40.19.0/24	157.248.91.42	0			0 60148 i

如范例 9-35 所示，可以使用命令 `show ip bgp neighbors ip-address [| begin Condition]` 监控条件路由通告。

范例 9-35 使用 show ip bgp neighbors 命令监控条件路由通告

Speedy# show ip bgp neighbors 157.248.91.41 begin Condition	
Condition-map not-in-table, Advertise-map advertise-me, status: Withdraw	

当 `non-exist-map` 指定的条件不满足时，条件通告的状态是 `Advertise`，通告映射中指定的路由被通告给对等体。

现在你已经理解了 BGP 路由抑制和聚合，了解如何使用 BGP 路由过滤来帮助定义网络策略也很重要。下一节介绍了路由过滤，它后面的内容介绍了如何与 BGP 属性一起使用路由过滤来过滤路由以及修改路径选择。

9.5 过滤 BGP 路由

你能够使用许多方法过滤 BGP 路由，可以通告分发列表、路由映射和前缀列表，利用 BGP 属性或是 BGP 团体属性过滤从邻居收到或是发给邻居的路由。本节介绍了使用路由映射、分发列表和前缀列表的基本 BGP 路由过滤。

基本的 BGP 路由过滤和 IGP 使用的路由过滤的配置很相象。使用访问列表或是前缀列表创建一系列的网段前缀，这些信息被应用到某个或是某些指定的邻居、对等体组或是所有的 BGP 对等体上。BGP 与 IGP 的路由过滤的主要区别是 BGP 提供了路由过滤选择标准的很多选项。

9.5.1 使用分发列表来过滤网段前缀

过滤 BGP 路由的最简单的方法是使用一个分发列表，可以作为一个空的描述应用到所有的对等体上，也可以使用 `neighbor` 描述应用到特定的对等体上。按照下面这些步骤将一个分

发列表应用到所有的对等体上来控制收到或是发送的路由。

第 1 步 创建一个访问列表或是前缀列表，指明需要过滤的流量。

第 2 步 在 BGP 路由配置模式下创建用来过滤所有收发的 UPDATE 报文的分发列表。

```
distribute-list { access-list-number | access-list-name | gateway prefix-list-name
                 | prefix prefix-list-name [gateway prefix-list-name] } {in [ interface-name
interface-number] | out [ interface-name interface-number | bgp | connected | egp
| eigrp | igmp | ospf | rip | static]}
```

注意: `distribute-list gateway prefix-list-name` 命令中的可选 `gateway` 描述使你能够过滤从某个特定的对等体来的所有路由，这些对等体由前缀列表定义。

可以在任何时候向进入和离开的（单独或是同时）更新应用一个分发列表，通过使用列表最后的可选 `interface-name` 和 `number` 描述可以将分发列表应用到从某个特定接口收到的 UPDATE 报文上。例如，路由器 Willis 现在接收到所有网段的路由，如范例 9-36 所示。

范例 9-36 Willis 的 BGP RIB

Willis# show ip bgp begin Network				
Network	Next Hop	Metric	LocPrf	Weight Path
*> 23.75.18.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.19.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.20.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.21.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.22.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.23.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.24.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.25.0/24	62.128.47.6			0 11151 5623 i
*> 23.75.26.0/24	62.128.47.6			0 11151 5623 i
*> 189.168.56.0/23	62.128.47.198	0		0 645 i
*> 189.168.58.0/23	62.128.47.198	0		0 645 i
*> 189.168.60.0/23	62.128.47.198	0		0 645 i
*> 189.168.62.0/23	62.128.47.198	0		0 645 i
*> 189.168.64.0/23	62.128.47.198	0		0 645 i
*> 189.168.66.0/23	62.128.47.198	0		0 645 i
*> 189.168.68.0/23	62.128.47.198	0		0 645 i
*> 189.168.70.0/23	62.128.47.198	0		0 645 i
Network	Next Hop	Metric	LocPrf	Weight Path
*> 189.168.72.0/23	62.128.47.198	0		0 645 i
*> 189.168.74.0/23	62.128.47.198	0		0 645 i
*> 189.168.76.0/23	62.128.47.198	0		0 645 i
*> 189.168.78.0/23	62.128.47.198	0		0 645 i
*> 189.168.80.0/23	62.128.47.198	0		0 645 i
*> 189.168.82.0/23	62.128.47.198	0		0 645 i
*> 189.168.84.0/23	62.128.47.198	0		0 645 i
*> 189.168.86.0/23	62.128.47.198	0		0 645 i
*> 189.168.88.0/23	62.128.47.198	0		0 645 i

为了过滤除了 23.75.0.0/16 以外的所有路由，需要建立一个指定 23.75.0.0/16 网段前缀的访问列表，然后在分发列表中使用这个访问列表来过滤所有进入的路由信息。范例 9-37 显示了路由器 Willis 的 BGP 配置以及相应的结果，在这个范例中，分发列表全局应用到所有的 BGP 邻居。

范例 9-37 路由器 Willis 的配置以及 BGP RIB 的紧接配置

```
Willis# show run | begin bgp
router bgp 2001
  no synchronization
  bgp log-neighbor-changes
  neighbor 62.128.47.6 remote-as 11151
  neighbor 62.128.47.194 remote-as 645
  neighbor 62.128.47.198 remote-as 645
  distribute-list 1 in
  no auto-summary
!
access-list 1 permit 23.75.0.0 0.0.255.255
Willis# show ip bgp | begin Network
      Network          Next Hop              Metric LocPrf Weight Path
*> 23.75.18.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.19.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.20.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.21.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.22.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.23.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.24.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.25.0/24       62.128.47.6                      0 1.151 5623 i
*> 23.75.26.0/24       62.128.47.6                      0 11151 5623 i
```

如前所述，也可以和 **distribute-list** 命令一起使用一个 **neighbor** 描述，过滤从一个特定的邻居或是对等体组接收或是发送的流量。可以使用下面的命令来实现这种类型的 BGP 路由过滤：

```
neighbor {ip-address | peer-group} distribute-list {access-list-number |
access-list-name} {in | out}
```

使用前面范例中的 BGP 配置和一个邻居分发列表，可以过滤来自对等体 62.128.47.6 的除了两条路由以外的所有路由。范例 9-38 显示了需要的命令和随之产生的 BGP 路由。

范例 9-38 过滤来自一个特定对等体的路由

```
Willis# show run | begin bgp
router bgp 2001
  no synchronization
  bgp log-neighbor-changes
  neighbor 62.128.47.6 remote-as 11151
  neighbor 62.128.47.6 distribute-list 50 in
  neighbor 62.128.47.194 remote-as 645
  neighbor 62.128.47.198 remote-as 645
  no auto-summary
!
access-list 50 permit 23.75.18.0 0.0.0.255
access-list 50 permit 23.75.19.0 0.0.0.255
Willis# show ip bgp neighbors 62.128.47.6 routes | begin Network
      Network          Next Hop              Metric LocPrf Weight Path
*> 23.75.18.0/24       62.128.47.6                      0 11151 5623 i
*> 23.75.19.0/24       62.128.47.6                      0 11151 5623 i
```

9.5.2 使用前缀列表过滤 BGP 路由

作为一个更简单更容易理解的路由过滤配置，可以使用命令 **neighbor {ip-address |**

`peer-group} prefix-list prefix-list-name {in | out}` 将前缀列表直接应用到 BGP 对等体上。

IP 前缀列表提供了一个更简单直观的替代访问列表的方法，IP 前缀列表使你能够使用一个列表名字或是数字来指定一系列的 **permit** 或是 **deny** 描述，通过指明前缀列表的序列号，可以单独地编辑 IP 前缀列表中的各个描述而不需要删除和重新应用全部列表。前缀列表也去除了通配符掩码 (wildcard mask) 计算的负担。如果你想要指定一个特定的主机 IP——例如，110.80.8.118/32——输入以下内容：

```
ip prefix-list bad-host seq 100 deny 110.80.8.118/32
```

如果你在路由器 Willis 的本地 BGP 配置上加上一些 62.128.0.0/23 网段，然后使用命令 `show ip bgp neighbor 62.128.47.6 advertised-routes`，你将看到在范例 9-39 中通告的路由。

范例 9-39 当前通告给对等体 62.128.47.6 的网段

Willis# show ip bgp neighbors 62.128.47.6 advertised-routes begin Network						
Network	Next Hop	Metric	LocPrf	Weight	Path	
*> 62.128.60.0/23	0.0.0.0	0		32768	i	
*> 62.128.64.0/23	0.0.0.0	0		32768	i	
*> 62.128.68.0/23	0.0.0.0	0		32768	i	
*> 62.128.72.0/23	0.0.0.0	0		32768	i	
*> 62.128.76.0/23	0.0.0.0	0		32768	i	
*> 189.168.56.0/23	62.128.47.198	0		0	645	i
*> 189.168.58.0/23	62.128.47.198	0		0	645	i
*> 189.168.60.0/23	62.128.47.198	0		0	645	i
*> 189.168.62.0/23	62.128.47.198	0		0	645	i
*> 189.168.64.0/23	62.128.47.198	0		0	645	i
*> 189.168.66.0/23	62.128.47.198	0		0	645	i
*> 189.168.68.0/23	62.128.47.198	0		0	645	i
*> 189.168.70.0/23	62.128.47.198	0		0	645	i
*> 189.168.72.0/23	62.128.47.198	0		0	645	i
*> 189.168.74.0/23	62.128.47.198	0		0	645	i
*> 189.168.76.0/23	62.128.47.198	0		0	645	i
*> 189.168.78.0/23	62.128.47.198	0		0	645	i
*> 189.168.80.0/23	62.128.47.198	0		0	645	i
*> 189.168.82.0/23	62.128.47.198	0		0	645	i
*> 189.168.84.0/23	62.128.47.198	0		0	645	i
*> 189.168.86.0/23	62.128.47.198	0		0	645	i
*> 189.168.88.0/23	62.128.47.198	0		0	645	i

现在假设你想只允许本地的 62.128.x.0 网段被通告给邻居 62.128.47.6，为了完成这个任务，加上一个 IP 前缀然后使用 **neighbor** 命令调用这个列表，如范例 9-40 所示。

这个 IP 前缀列表提供了与一个通配符掩码为 0.0.1.255 的访问列表相同的功能，62.128.0.0/16 **le** 23 前缀列表允许任何以 62.128.x.x 开始带有小于等于 23 位子网掩码的网段。如果你决定从使用访问列表转变为使用 IP 前缀列表，需要在将前缀列表应用到一个邻居之前仔细地检查你的语法。记住，和访问列表一样，前缀列表以一个隐式的拒绝作为结束，如果你在列表的开始使用一个 **deny** 描述，那么必须在列表中的某个位置包括一个 **permit** 描述来允许其他的流量。可能一开始你会对 **ge** 和 **le** 命令参数感到迷惑；记住用于前缀的掩码必须与所有被过滤的路由前缀精确匹配。**ge/le** 参数与一个子网范围匹配，其作用就像反转掩码。参见附录 D 以获得更多配置 IP 前缀列表方面的帮助信息。

范例 9-40 使用前缀列表过滤 BGP 路由

```

Willis# show run | begin bgp
router bgp 2001
no synchronization
bgp log-neighbor-changes
network 62.128.60.0 mask 255.255.254.0
network 62.128.64.0 mask 255.255.254.0
network 62.128.68.0 mask 255.255.254.0
network 62.128.72.0 mask 255.255.254.0
network 62.128.76.0 mask 255.255.254.0
neighbor 62.128.47.6 remote-as 11151
neighbor 62.128.47.6 prefix-list route-filter out
neighbor 62.128.47.194 remote-as 645
neighbor 62.128.47.198 remote-as 645
no auto-summary
!
ip prefix-list route-filter seq 5 permit 62.128.0.0/16 le 23
Willis# show ip bgp neighbors 62.128.47.6 advertised-routes | begin Network

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 62.128.60.0/23	0.0.0.0	0		32768	i
*> 62.128.64.0/23	0.0.0.0	0		32768	i
*> 62.128.68.0/23	0.0.0.0	0		32768	i
*> 62.128.72.0/23	0.0.0.0	0		32768	i
*> 62.128.76.0/23	0.0.0.0	0		32768	i

9.5.3 采用路由映射过滤 BGP 路由

另一种复杂的方法是在路由映射中通过 **neighbor** 声明来实现路由过滤。在路由映射方式下有多种基本方法可以过滤 BGP 路由：通过属性、网络地址前缀、下一跳地址或者路由类型。当实施 BGP 路由过滤时，**match** 命令用来匹配指定的条目，然后路由映射被应用于 BGP 邻居或对等体组。表 9-3 列出了 BGP 中支持的路由映射 **match** 命令类型。

表 9-3 BGP 相关的路由映射 match 命令

match 命令	描述
as-path as-path-access-list-number	匹配 as-path-access-list number 指定的 AS 路径属性（范围从 1~199）。AS 路径访问控制列表和其他的 AS 路径的功能在本章的后面进行详细的讨论
community community-list-number [exact-match]	匹配团体列表指定的团体的值，有两种类型的团体列表：标准的（范围从 1~99）和扩展的（范围从 100~199）。 exact-match 命令用于指定一个准确的匹配。团体列表和其他的 BGP 团体属性的功能在本章的后面进行讨论
ip address {access-list-number access-list-name prefix-list prefix-list-name}	匹配访问控制列表或者前缀列表指定的 IP 前缀
ip next-hop {access-list-number access-list-name prefix-list prefix-list-name}	匹配一条路由的下一跳属性。下一跳的值是由尾部的访问控制列表或者前缀列表指定的 下一跳属性和它的使用在本章的后面进行讨论
ip route-source { access-list-number access-list-name prefix-list prefix-list-name }	匹配发送这条路由的对方的源 IP 地址。对方的 IP 地址是由访问控制列表或者前缀列表指定的。 match ip route-source 命令只在发送方向的路由映射中支持
metric metric-value	匹配一个 MULT_EXIT_DISC （多出口鉴别器）的值，匹配的度量对于输入或者输出方向的路由过滤不支持 多出口鉴别器属性和它的使用在本章的后面进行讨论

续表

match 命令	描述
route-type {internal external local}	匹配一条本地产生的路由（使用 show ip bgp，看到源是 0.0.0.0）。match route-type 命令只对输出方向的路由过滤支持 确保测试 route-type local 命令产生的结果。这条命令只匹配任何本地产生的路由，包括由路由重分发进入到 BGP 进程中的路由
tag tag-value	匹配一个标记值 BGP 标记的使用在前面第 2 章中介绍过

使用路由映射配置 BGP 的基本路由过滤只需要两个步骤：

- 第 1 步 使用 route-map 命令建立一个路由映射，从路由映射的配置模式中，使用 match 命令来指定需要匹配的属性（路由映射的配置在第 2 章中介绍过）。
- 第 2 步 使用下面的命令将路由映射绑定到邻居或者对等体组上。

```
neighbor { ip-address | peer-group-name } route-map route-map-name {in | out}
```

下面的范例显示了用户如何使用一个简单的路由映射来限制只通告本地产生的路由。范例

9-41 显示了在应用路由映射过滤之前，Willis 路由器当前正在给对等体 62.128.47.6 通告的路由。

范例 9-41 在应用路由映射之前 Willis 路由器给对等体 62.128.47.6 通告的路由

Willis# show ip bgp neighbors 62.128.47.6 advertised-routes begin Network						
Network	Next Hop	Metric	LocPrf	Weight	Path	
*> 23.75.18.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.19.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.20.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.21.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.22.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.23.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.24.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.25.0/24	62.128.47.6			0	11151 5623 i	
*> 23.75.26.0/24	62.128.47.6			0	11151 5623 i	
*> 62.128.0.0/23	0.0.0.0	0		32768	i	
*> 62.128.4.0/23	0.0.0.0	0		32768	i	
*> 62.128.8.0/23	0.0.0.0	0		32768	i	
*> 62.128.12.0/23	0.0.0.0	0		32768	i	
*> 62.128.16.0/23	0.0.0.0	0		32768	i	
*> 62.128.20.0/23	0.0.0.0	0		32768	i	
*> 62.128.24.0/23	0.0.0.0	0		32768	i	
*> 62.128.28.0/23	0.0.0.0	0		32768	i	
*> 62.128.32.0/23	0.0.0.0	0		32768	i	
*> 62.128.36.0/23	0.0.0.0	0		32768	i	
*> 62.128.40.0/23	0.0.0.0	0		32768	i	
*> 62.128.44.0/23	0.0.0.0	0		32768	i	
*> 62.128.48.0/23	0.0.0.0	0		32768	i	
Network	Next Hop	Metric	LocPrf	Weight	Path	
*> 62.128.52.0/23	0.0.0.0	0		32768	i	
*> 62.128.56.0/23	0.0.0.0	0		32768	i	
*> 62.128.60.0/23	0.0.0.0	0		32768	i	
*> 62.128.64.0/23	0.0.0.0	0		32768	i	
*> 62.128.68.0/23	0.0.0.0	0		32768	i	
*> 62.128.72.0/23	0.0.0.0	0		32768	i	
*> 62.128.76.0/23	0.0.0.0	0		32768	i	
*> 189.168.56.0/23	62.128.47.198	0		0	645 i	
*> 189.168.58.0/23	62.128.47.198	0		0	645 i	
*> 189.168.60.0/23	62.128.47.198	0		0	645 i	

(待续)

*> 189.168.62.0/23	62.128.47.198	0	0	645	i
*> 189.168.64.0/23	62.128.47.198	0	0	645	i
*> 189.168.66.0/23	62.128.47.198	0	0	645	i
*> 189.168.68.0/23	62.128.47.198	0	0	645	i
*> 189.168.70.0/23	62.128.47.198	0	0	645	i
*> 189.168.72.0/23	62.128.47.198	0	0	645	i
*> 189.168.74.0/23	62.128.47.198	0	0	645	i
*> 189.168.76.0/23	62.128.47.198	0	0	645	i
*> 189.168.78.0/23	62.128.47.198	0	0	645	i
*> 189.168.80.0/23	62.128.47.198	0	0	645	i
*> 189.168.82.0/23	62.128.47.198	0	0	645	i
*> 189.168.84.0/23	62.128.47.198	0	0	645	i
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 189.168.86.0/23	62.128.47.198	0	0	645	i
*> 189.168.88.0/23	62.128.47.198	0	0	645	i

范例 9-42 显示了如何使用一个简单的小的路由映射就可以过滤除了本地产生的路由之外来自任何源的路由，将过滤后的路由通告给属于 all-peers 对等体组的所有用户。

范例 9-42 使用 route-type local 命令来过滤路由

```
Willis# show run | begin bgp
router bgp 2001
no synchronization
bgp log-neighbor-changes
network 62.128.60.0 mask 255.255.254.0
network 62.128.64.0 mask 255.255.254.0
network 62.128.68.0 mask 255.255.254.0
network 62.128.72.0 mask 255.255.254.0
network 62.128.76.0 mask 255.255.254.0
neighbor all-peers peer-group
neighbor all-peers route-map route-filter out
neighbor 62.128.47.6 remote-as 11151
neighbor 62.128.47.6 peer-group all-peers
neighbor 62.128.47.194 remote-as 645
neighbor 62.128.47.194 peer-group all-peers
neighbor 62.128.47.198 remote-as 645
neighbor 62.128.47.198 peer-group all-peers
no auto-summary
!
route-map route-filter permit 10
match route-type local
```

当完成这个配置后，Willis 路由器将只通告范例 9-43 中显示的路由给所有属于 all-peers 对等体组的成员。这个范例使用了 `show ip bgp neighbors peer-group advertised-routes` 命令来显示被通告给 all-peers 对等体组的路由。

范例 9-43 show ip bgp neighbors peer-group advertised-routes 命令

```
Willis# show ip bgp neighbors 62.128.47.6 advertised-routes | begin Network
Network      Next Hop      Metric LocPrf Weight Path
*> 62.128.60.0/23 0.0.0.0      0        32768 i
*> 62.128.64.0/23 0.0.0.0      0        32768 i
*> 62.128.68.0/23 0.0.0.0      0        32768 i
*> 62.128.72.0/23 0.0.0.0      0        32768 i
*> 62.128.76.0/23 0.0.0.0      0        32768 i
```

你可能注意到路由映射 **set** 命令没有在表 9-3 中显示；这是因为路由映射 **set** 命令提供了更高级的 BGP 功能——BGP 属性的操作。BGP 路由映射的另外一个更强大的使用就是对 BGP 属性的操作和 BGP 的路由惩罚。每个主题都会在本章的后面讨论。BGP 属性值可以在路由映射的配置模式下使用 **set** 命令来操作，并且使用 **neighbor {ip-address | peer-group} route-map route-map-name {in | out}** 命令将其绑定到邻居或者对等体上。下面的列表显示了在下一节中讨论的 **set** 命令的简短描述：

- **as-path prepend** *as-path-number*
- **as-path tag** *as-path-string*
- **comm-list** *community-list-number* [delete]
- **community** [*community-value-decimal* | *aa: nn-format*]
- **community additive**
- **community internet**
- **community local-as**
- **community no-advertise**
- **community no-export**
- **community none**
- **dampening** *half-life-value reuse-penalty-value suppress-penalty-value*
- **ip default next-hop** *ip-address*
- **ip default next-hop verify-availability**
- **local-preference** *value*
- **metric** [+ | -] *metric-value*
- **origin** {*egp as-number* | *igp* | *incomplete*}
- **tag** *tag-value*
- **weight** *weight-value*

9.6 使用 BGP 属性来建立路由策略

在前面的一些章节中，本书讨论了 BGP 的机制、邻居的配置、MD-5 的验证和路由聚合。本节教会用户如何使用 BGP 属性来将前面学到的技术综合到一起并且将 BGP 用作强壮的路由协议，就像它号称的那样。这一节探讨不同属性类型的配置，以及它们可能和 BGP 使用的许多方式，包括如何做下面的这些事情：

- 过滤接收或者发送方向的路由；
- 定制路由重分发；
- 特殊的路由聚合；
- 操作 BGP 路由选择过程；
- 在进口或者出口点，指定最好的路由；
- 下一跳的修改；
- 修改上游或者下游的对等体如何传播特定的路由。

可以在思科的路由器上对属性使用很多方法来修改 BGP 的路由——使用路由映射、属性

为了将路由映射应用到一个聚合的网络中，使用 **aggregate-address ip-prefix subnet-mask attribute-map route-map-name [summary-only]** 命令。

为了将路由映射应用到来自或者去往某个邻居或者对等体组的所有宣告的路由，使用 **neighbor {ip-address|peer-group-name} route-map route-map-name {in|out}** 命令。

注意：当你做出配置变化时，可能需要复位 BGP 的进程来使得变化生效。为了清除一个 BGP 进程而无需复位所有的会话，使用 **clear ip bgp * soft [in|out]** 命令。

为了测试在 Willis 路由器上，起源属性的变化对 BGP 路由选择过程的影响效果，如图 9-13 所示，使用 **neighbor ip-address route-map route-map-name** 命令改变所有发送方向上的 BGP 路由更新。范例 9-45 显示了在变化发生之前 189.168.x.0 网络的 BGP 表项。

范例 9-45 Willis 路由器对于 189.168.x.0 网络的 BGP 表

```
Willis# show ip bgp 189.168.0.0/16 longer-prefixes
BGP table version is 119, local router ID is 62.128.47.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 189.168.56.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.58.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.60.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.62.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.64.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.66.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.68.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.70.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.72.0/23	62.128.47.198	0		0	645 i
Network	Next Hop	Metric	LocPrf	Weight	Path
*>	62.128.47.194	0		0	645 i
* 189.168.74.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.76.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.78.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.80.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.82.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.84.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.86.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i
* 189.168.88.0/23	62.128.47.198	0		0	645 i
*>	62.128.47.194	0		0	645 i

注意：为了简单起见，在图 9-13 中显示的 BGP 属性网络会在这一小节所有的范例中使用。

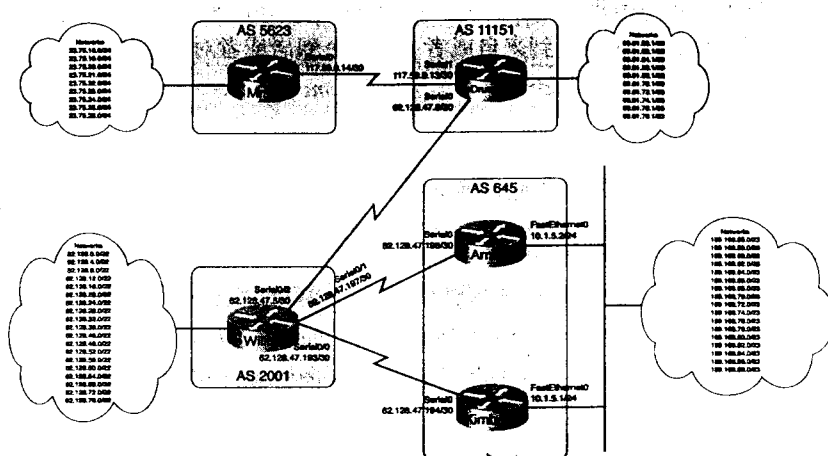


图 9-13 BGP 属性网络

范例 9-46 显示了 Kimberly 路由器的配置。在这个范例中，Kimberly 路由器已经被配置发送所有本地产生的路由给邻居 62.128.47.97，即 Willis 路由器，但是起源属性被修改为 INCOMPLETE。Willis 路由器在范例 9-47 中显示。

范例 9-46 Kimberly 路由器的初始 BGP 配置

```
Kimberly# show run | begin bgp
router bgp 645
no synchronization
bgp router-id 10.1.5.1
bgp log-neighbor-changes
network 189.168.56.0 mask 255.255.254.0
network 189.168.58.0 mask 255.255.254.0
network 189.168.60.0 mask 255.255.254.0
network 189.168.62.0 mask 255.255.254.0
network 189.168.64.0 mask 255.255.254.0
network 189.168.66.0 mask 255.255.254.0
network 189.168.68.0 mask 255.255.254.0
network 189.168.70.0 mask 255.255.254.0
network 189.168.72.0 mask 255.255.254.0
network 189.168.74.0 mask 255.255.254.0
network 189.168.76.0 mask 255.255.254.0
network 189.168.78.0 mask 255.255.254.0
network 189.168.80.0 mask 255.255.254.0
network 189.168.82.0 mask 255.255.254.0
network 189.168.84.0 mask 255.255.254.0
network 189.168.86.0 mask 255.255.254.0
network 189.168.88.0 mask 255.255.254.0
neighbor 10.1.5.2 remote-as 645
neighbor 10.1.5.2 route-reflector-client
neighbor 10.1.5.2 next-hop-self
neighbor 62.128.47.193 remote-as 2001
neighbor 62.128.47.193 route-map change-origin out
no auto-summary
!
route-map change-origin permit 10
match route-type local
set origin incomplete
```

范例 9-47 Willis 路由器在起源属性变化后的 BGP RIB 表

```
Willis# show ip bgp | include 645
*> 189.168.56.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.58.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.60.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.62.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.64.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.66.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.68.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.70.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.72.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.74.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.76.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.78.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.80.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.82.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.84.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.86.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
*> 189.168.88.0/23 62.128.47.198 0 0 645 i
* 62.128.47.194 0 0 645 ?
```

而且，注意 Willis 路由器现在偏爱所有来自 Arnold 路由器的路由，即 62.128.47.198。范例 9-48 显示了 Willis 路由器的 IP 路由表。

范例 9-48 Willis 路由器的 IP 路由表

```
Willis# show ip route | include 189
189.168.0.0/23 is subnetted, 17 subnets
B 189.168.60.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.62.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.56.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.58.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.84.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.86.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.80.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.82.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.88.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.68.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.70.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.64.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.66.0 [20/0] via 62.128.47.198, 00:02:48
B 189.168.76.0 [20/0] via 62.128.47.198, 00:02:48
```

(待续)

```
B      189.168.78.0 [20/0] via 62.128.47.198, 00:02:48
B      189.168.72.0 [20/0] via 62.128.47.198, 00:02:48
B      189.168.74.0 [20/0] via 62.128.47.198, 00:02:48
```

就像你看到的，起源属性可以用来修改 BGP 的路径选择过程。既然你已经看到了起源属性修改的范例，接下来了解如何使用 AS 路径属性来修改路径选择的决策。虽然起源属性可以修改来改变最佳路径选择，但是起源属性的修改对于 BGP 路径选择不是最好的方法。

9.6.2 使用 AS 路径属性来影响路径选择

每次当路由更新从一个 AS 传递到另外一个 AS 时，AS 路径属性就会被修改，以存储这条路由到达当前位置沿途通过的路径。你可能想起在第 7 章中，BGP UPDATE 报文中的 AS 路径字段含有 AS 路径，以从右到左的方式，开始的是起始的 AS，如范例 9-49 所示。

范例 9-49 显示一条 BGP 路由的 AS 路径属性

```
MrsG# show ip bgp 189.168.88.0/23
BGP routing table entry for 189.168.88.0/23, version 699
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer
Please add shading to next line
 11151 2001 645
    117.59.0.13 from 117.59.0.13 (117.59.0.13)
      Origin IGP, localpref 100, valid, external, best
```

在先前的范例中，你可能看到到达 189.168.88.0/23 网络的路由起始于 Arnold 路由器的 AS 645，接着通过 AS 2001，即 Willis 路由器，接着穿过 AS 11 151，即 MrDrummand 路由器，最终到达它当前的位置，即 MrsG 路由器。AS 路径信息主要是一种 BGP 环路的检测机制，如果一台路由器在路径列表中看到了它自己，那么这条路由就会认为是环路，被忽略掉。

注意： `neighbor ip-address allowas-in [number-of-occurrences]` 命令允许运行思科 IOS 软件的路由器在接收的 BGP 更新的 AS 路径属性中，接受最多 10 次本地 AS 号码的重复。使用这个命令时，要特别注意，因为它会关闭掉 BGP 主要的环路预防方法。

AS 路径信息也用于提供几种新的特性，包括 BGP AS 路径过滤，使用常规表达式的 BGP RIB 查找，以及影响 BGP 选路策略的 AS 路径信息。记住，BGP 选路进程是基于下面这些表项来选择路由：

1. 最大的权重属性；
2. 最大的本地优先属性；
3. 本地产生的路由（BGP RIB 中下一跳为 0.0.0.0）；
4. 最短的 AS 路径属性；
5. 最佳路由起源属性：IGP，EGP，INCOMPLETE；
6. 最低的多出口鉴别器属性；
7. E-BGP 路由优于 I-BGP 路由（也具有较低的管理距离）；
8. 最老的路由（越老的路由越稳定）；

9. 从路由器始发的路由具有最低的 BGP router ID;

10. 如果路由器是路由反射器，最低的 CLUSTER_ID 属性长度;

11. 从对等体收到的具有最低 IP 地址的路由。

这是一个常用的但并不推荐的方法，即使用 AS 路径作为在因特网中路径选择的决定因素。作为一个实验，到因特网站点 looking-glass 上，你会发现含有重复几次的相同 AS 号码的 AS 路径的路由；这被称为 *AS 路径添加*。AS 路径添加在 AS 路径中的当前位置（AS 路径最左边的位置）添加由用户指定的多个本地 AS 号码。这种方法通常不推荐，因为因特网路由通常会穿过许多自治系统，并且当每条路由离开每一个 AS 时，这个 AS 的边界路由器也会在路径中添加它们的本地 AS 号码，所以没有办法保证你原来在路径中添加的 AS 号码将会产生你所期望的效果。当探究因特网路由表时，你甚至會注意到某些路由在 AS 路径中有 20 个表项。这很可能是因为两个或者更多个边界路由器在 AS 路径中添加了它们的 AS 号码，当你查看路由时，它可能已经通过了几个自治系统。

为了在思科的路由器上操作 AS 路径，在路由映射中使用 **set as-path prepend as-number** 命令并且指定你想添加到路由中的 AS 值。只需要两步将 AS 号码添加到 AS 路径中去。

第 1 步 建立一个路由映射，在访问控制列表或者前缀列表指定将要添加 AS 列表的网络，并且识别要添加到路径中的 AS 号码。为了改变所有的本地产生路由的 AS 路径，使用 **match route-type local** 命令，它可以匹配所有的本地路由器产生的路由（这在大型公共的网络上不是一个很好的主意）。

第 2 步 将路由映射绑定到适当的邻居或者对等体组上。

注意：虽然可以添加任何随机的 AS 数值到 AS 路径中来增加 AS 路径的尺寸，但这实际上不是一个很好的经验。添加本地的 AS 号码不会对本地网络产生任何伤害，也不会对直连的对等体的网络产生任何伤害，但是路由器添加任意随机的 ASN 号码，可能会在通过 AS 的过程中，正好遇到那个你随机加入的 AS 号码的 AS，导致严重的问题（也非常尴尬）。许多服务提供商对于 AS 添加都有相关的策略。在配置 BGP 属性之前，总是去咨询你的服务提供商。如果你计划在网络中使用 AS 添加的功能，成为一个因特网的好邻居，并且只是添加适合条件的 AS 号码。

因为思科的 BGP 实施比较 AS 路径的长度（作为第 4 个最佳路径的决定因素），当一个 AS 有超过一个出口点时，可以使用 AS 路径添加的功能，使得一条路径的长度大于另外一条的长度。这使得上游 BGP 的对等体更偏爱具有较小的 AS 路径属性的路由。如果 Kimberly 路由器将发送给 Willis 路由器的所有本地路由添加它自己的 AS 号码 (AS 645)，在这个范例中，这会导致 Willis 路由器偏爱来自 Arnold 路由器的路由。如果 Willis 和 Arnold 之间的连接断掉了，Willis 路由器将清除来自 Arnold 路由器的路由，使用来自 Kimberly 路由器的路由。当 Willis 和 Arnold 之间的连接修复后，并且交换了 BGP 的路由，Willis 路由器将再次使用来自 Arnold 路由器的路由。范例 9-50 显示了自治系统添加的功能是如何应用在属性网络中的。在这个范例中，自治系统 645 有两个出口点：一个是 Arnold 路由器，另外一个为 Kimberly 路由器。

在尝试下一个范例之前，删除先前的范例中使用的路由映射 **change-origin**。

当 Willis 路由器收到来自 Kimberly 路由器的路由更新后，它不再使用来自 Kimberly 路由器的路由，它有最低的 BGP router ID 和 IP 地址。这是因为来自 Kimberly 路由器的路由的 AS 路径长度长于 Arnold 路由器始发的路由的 AS 路径长度。范例 9-51 显示了来自 Willis 路

由器的 BGP 路由范例。

范例 9-50 给 AS 路径添加 ASN

```
Kimberly# show run | begin bgp
router bgp 645
no synchronization
bgp router-id 10.1.5.1
bgp log-neighbor-changes
network 189.168.56.0 mask 255.255.254.0
network 189.168.58.0 mask 255.255.254.0
network 189.168.60.0 mask 255.255.254.0
network 189.168.62.0 mask 255.255.254.0
network 189.168.64.0 mask 255.255.254.0
network 189.168.66.0 mask 255.255.254.0
network 189.168.68.0 mask 255.255.254.0
network 189.168.70.0 mask 255.255.254.0
network 189.168.72.0 mask 255.255.254.0
network 189.168.74.0 mask 255.255.254.0
network 189.168.76.0 mask 255.255.254.0
network 189.168.78.0 mask 255.255.254.0
network 189.168.80.0 mask 255.255.254.0
network 189.168.82.0 mask 255.255.254.0
network 189.168.84.0 mask 255.255.254.0
network 189.168.86.0 mask 255.255.254.0
network 189.168.88.0 mask 255.255.254.0
neighbor 10.1.5.2 remote-as 645
neighbor 62.128.47.193 remote-as 2001
neighbor 62.128.47.193 route-map prepend out
no auto-summary
!
route-map prepend permit 10
match route-type local
set as-path prepend 645
```

范例 9-51 一个被添加路由的 BGP 路由信息

```
Willis# show ip bgp 189.168.56.0/23
BGP routing table entry for 189.168.56.0/23, version 276
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
Advertised to non peer-group peers:
62.128.47.6 62.128.47.194
645
62.128.47.198 from 62.128.47.198 (10.1.5.2)
Origin IGP, metric 0, localpref 100, valid, external, best
645 645
62.128.47.194 from 62.128.47.194 (10.1.5.1)
Origin IGP, metric 0, localpref 100, valid, external
```

9.6.3 使用 AS 路径属性过滤 BGP 路由

过滤大量路由的一种最简单的方法就是使用 AS 路径访问控制列表通过 AS 号码来过滤路由。如果你不熟悉常规表达式，这是你第一次使用 AS 路径访问控制列表，你可能会发现 AS 路径过滤进程是相当混淆的，导致不可预料的结果。建立一个完美的 AS 路径访问控制列表需要你首先熟悉常规表达式的使用。但是，放松点，接着读下去，因为你现在准备轻松地学习常规表达式。

注意：思科 IOS 软件使用许多和 UNIX/Linux 世界中相同的常规表达式。如果你不熟悉常规表达式的话，你可以参考附录，直接阅读有相关内容的书。

一、如何使用常规表达式

常规表达式让人觉得比较陌生的原因之一就是它使用看起来很奇怪的结构。如果你像这里的许多非数学专家一样，你可能发现表达式例如`^400$`看起来更像外币，而不是 AS 路径的值。然而，这个常规表达式只是简单地意味着下面的事情：

`^` = “开始于”

`$` = “结束于”

或者开始并且结束于 ASN 400。

所以，这个语句只是简单地意味着开始和结束于号码 400；这个常规表达式只匹配 AS 号码 400 这一个实例。到目前为止，你可能会问，为什么不能只是输入“400”，难道不行吗？原因是号码 400 匹配任何开始、结束或者含有号码 400 的字符串。在常规表达式中有许多方法使用特殊的字符来代表不同的字符串。找到你所需要的 AS 路径序列号的最好方法就是使用 **show ip bgp regexp regular-expression** 命令。当使用这个命令时，在路由过滤中使用最好的表达式之前就可以对每一个常规表达式进行测试，以找到所有可能的匹配。范例 9-52 显示了 **show ip bgp regexp** 命令是如何找到 AS path 645 的任何实例的。

范例 9-52 show ip bgp regexp 命令

```
Willis# show ip bgp regexp _645_
  Network        Next Hop        Metric LocPrf Weight Path
* 10.1.1.0/24    62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 400 i
* 10.2.2.0/24    62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 100 400 i
* 10.3.3.0/24    62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 400 400 100 i
*> 189.168.56.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.58.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.60.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.62.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.64.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.66.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.68.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.70.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.72.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.74.0/23 62.128.47.194          0           0 645 645 645 645 i
*> 189.168.76.0/23 62.128.47.194          0           0 645 645 645 645 i
  Network        Next Hop        Metric LocPrf Weight Path
*> 189.168.78.0/23 62.128.47.194          0           0 645 645 645 645 i
* 189.168.80.0/23 62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 645 645 645 i
* 189.168.82.0/23 62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 645 645 645 i
* 189.168.84.0/23 62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 645 645 645 i
* 189.168.86.0/23 62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 645 645 645 i
* 189.168.88.0/23 62.128.47.198          0           0 645 800 234 6768 i
*>               62.128.47.194          0           0 645 645 645 645 i
```

提示：如果你在使用 **show ip bgp regexp** 命令时，发现一个特定的常规表达式不工作，即使你绝对相信它应当工作，再次检查！你可能无意中在常规表达式的尾部输入了一个空格，这样做会改变常规表达式的含义，并且阻止它找到相应的匹配。这就是为什么我们在使用任何一个常规表达式做测试运行后，再把它应用到实际的生产性网络中是一个很好的主意。

表 9-4 显示了可以在常规表达式中使用的一些特殊字符、字符的定义和它们的使用的范例。

表 9-4

常规表达式中使用的特殊字符

字符	含义	范例
^ 用于表达式的开始	开始于什么	^1 =开始于 1。这意味着任何以 1 开始的字符串都匹配这个字符串。例如：1 400 500 或者 123 456 7891
\$ 美元符号 用于表达式的结束	终止于什么	400\$ =结束于 400 这意味着任何以 400 作为结束的字符串都匹配这个常规表达式。 例如：6:5 400 或者 645 100 400 400 然而，常规表达式 ^400\$ 意味着开始并且结束于 400。 ^\$ 匹配一个空的 AS 路径
* 星号 用于一个表达式的尾部	0 或者多个什么	40* =含有 0 或者多个字符串 4 的实例： 可以匹配的有： 645 645 400 645 100 4 645 400 400 100 44 645 775 801 212 ^645* 匹配任何以 645 开始的字符串： 例如： 645 100 400 645 645 645
. 点 用于表达式的任何地方	任何字母（包括空格）	645 匹配任何字符串 645 的实例，例如： 1645 645 645 777 645 645 645 645. 匹配任何包含 645 的字符串。例如： 645 645 645 100 645 400 189 201 13645 .* 匹配任何 as 的路径，包括一个空的路径
+ 加号 不能用于表达式的开始	1 个或者多个在+号之前的内容的重复	645+ 匹配 1 个或者多个 645 字符串的实例： 例如： 6451 645 400 100 400 100 645 645 645 645
- 用于中括号之间指定一个范围	用于一个起始和终止点之间的范围	用于中括号指定的一个范围： [x-x] *参看中括号 [] 的用法
? 问号 用于一个表达式的尾部。需要使用 CTRL-v 键才能使得？可以作为一个字母使用	0 或者 1 个什么的重复	645? 匹配任何含有 645 的字符串；例如： 645 645 645 645 645 645 400 123 400 400 645 ^645? 开始于 645，可以结束于任何内容

续		
字符	含义	范例
<u>下划线</u>	匹配任何特殊的字符，例如下面的： 。 逗号 () 小括号 { } 大括号 字符串的开始 字符串的结束 空格	用于建立复杂的表达中带有一些特殊的字符。 例如， <u>645</u> 匹配任何含有 645 的路径。 645 645 645 645 645 800 234 645
() 小括号	匹配 AS 路径中联盟的路径，也用于建立数字序列	(65501) \$ 匹配任何以 (65501) 结束的。 径。 例如： 101 (65501)
[] 中括号	字母的范围	[0-9] 匹配任何数字串，但是不匹配空的。 径。 例如： 645 645 400 100 11151 2001 5623 11151 2001 [058] \$ 匹配任何含有字母 0、5 或者 8 的。 径。 例如： 645 645 645 800 234 6768 645 400 400 100 ^356_[0-9] 匹配任何以 356 开始的 AS 并且 后面具有至少一个尾部的 ASN。例如： 356 789 012 356 012 356 356

注意：当输入？字符时，不要忘记使用 CTRL-V 的键盘序列，否则，你将会持续 句思
科 IOS 软件请求帮助。

当你已经非常适应常规表达式之后，可以使用这些表达式来建立 AS 路径的访问 制列
表。

二、AS 路径访问控制列表和常规表达式

与用于 IP 流量的常规的带数字的访问控制列表类似，AS 路径访问控制列表也 数字
的访问控制列表，它可以基于 AS 路径的值来匹配流量。这个 AS 的数值是使用常 式
来指定的。而且，非常类似于 IP 访问控制列表，每一个 AS 路径访问控制列表都结 一个
显示的 deny any。AS 路径访问控制列表是使用下面的命令建立的：

```
ip as-path access-list list-number {permit | deny} regular-expression
```

例如，假设 Willis 路由器有一个新的需求，就是阻塞所有含有 AS 路径的值为 6 的网
络前缀。这可以很容易地用 AS 路径的访问列表来完成，拒绝在 AS 路径中含有 645 任何
实例，如范例 9-53 所示。

范例 9-53 使用 AS 路径访问控制列表来过滤含有 645 的 BGP 路由

```
Willis# show run | include as-path
ip as-path access-list 1 deny _645_
ip as-path access-list 1 permit .*
```

在先前的范例中，AS 路径访问控制列表 1 用于拒绝任何含有字符串 645 的 AS ，而

其他的流量都是允许的。常规表达式 `645` 描述了任何含有数值 `645` 的字符串，而 `*` 常规表达式允许任何其他的路径值。

就像 BGP 中的许多参数一样，有两种方法来应用 AS 路径访问控制列表：使用路由映射或者使用过滤列表。两种方法都会在本小节中介绍。首先考虑路由映射的配置。

使用路由映射，需要 3 个步骤来配置 AS 路径的前缀过滤：

第 1 步 建立一个 AS 路径的访问控制列表，它将用于指定 AS 路径的常规表达式。

第 2 步 建立一个路由映射，来告诉路由器如何使用这个 AS 路径的访问控制列表。

第 3 步 使用 `neighbor {ip-address | peer-group} route-map route-map-name {in | out}` 命令，将路由映射绑定到一个 BGP 的邻居或者对等体上。

如果你准备在路由映射中使用访问控制列表的话，必须定义一个路由映射来告诉路由器如何使用 AS 路径的访问控制列表。就像我们在表 9-3 中所提到的那样，`match as-path as-path-access-list-number` 命令指定了用于匹配的 AS 路径。对于这个范例，如范例 9-54 所示，`route-map filter-as` 用于匹配 AS 路径访问控制列表 1。

范例 9-54 使用一个具有 AS 路径访问控制列表的路由映射

```
Willis# show run | begin route-map
route-map filter-as permit 10
match as-path 1
```

当建立路由映射后，可以将它绑定到一个邻居或者对等体组上。范例 9-55 显示了对于 Willis 路由器的已完成的 AS 路径访问列表的过滤配置。

范例 9-55 对 BGP 的对等体应用一个路由映射

```
Willis# show run | begin bgp
router bgp 2001
no synchronization
bgp router-id 62.128.47.5
bgp log-neighbor-changes
network 62.128.0.0 mask 255.255.252.0
network 62.128.4.0 mask 255.255.252.0
network 62.128.8.0 mask 255.255.252.0
network 62.128.12.0 mask 255.255.252.0
network 62.128.16.0 mask 255.255.252.0
network 62.128.20.0 mask 255.255.252.0
network 62.128.24.0 mask 255.255.252.0
network 62.128.28.0 mask 255.255.252.0
network 62.128.32.0 mask 255.255.252.0
network 62.128.36.0 mask 255.255.252.0
network 62.128.40.0 mask 255.255.252.0
network 62.128.48.0 mask 255.255.252.0
network 62.128.52.0 mask 255.255.252.0
network 62.128.56.0 mask 255.255.252.0
network 62.128.60.0 mask 255.255.252.0
network 62.128.64.0 mask 255.255.252.0
network 62.128.68.0 mask 255.255.252.0
network 62.128.72.0 mask 255.255.252.0
network 62.128.76.0 mask 255.255.252.0
aggregate-address 62.128.44.0 255.255.255.252
neighbor 62.128.47.6 remote-as 11151
neighbor 62.128.47.6 route-map filter-as out
```

(待续)

```

neighbor 62.128.47.194 remote-as 645
neighbor 62.128.47.198 remote-as 645
no auto-summary
!
ip as-path access-list 1 deny _645_
ip as-path access-list 1 permit .*
!
route-map filter-as permit 10
match as-path 1

```

在先前的范例中，路由映射 filter-as 用于拒绝到 MrDrummand 路由器的任何带有 ASN 645 的发送方向的路由更新。**permit .*** 常规表达适用于允许所有其他的 AS 号码。

BGP 过滤列表提供了一个简单的、基于 AS 路径访问控制列表过滤的途径。过滤列表只用于基于 AS 路径来过滤 BGP 路由。

只需要两个步骤来基于 AS 路径配置 BGP 的路由过滤。

第 1 步 建立一个 AS 路径访问控制列表来指定要匹配的 AS 路径。

第 2 步 使用下面的命令将路由映射应用到 BGP 的邻居或者对等体上。

```

neighbor { ip-address | peer-group } filter-list as-path-access-list-number
{in | out}

```

范例 9-56 显示了 **filter list** 命令如何完成和范例 9-55 用路由映射完成的相同的效果。

范例 9-56 使用过滤列表来基于 AS 路径过滤 BGP 流量

```

Willis# show run | begin bgp
router bgp 2001
no synchronization
bgp router-id 62.128.47.5
bgp log-neighbor-changes
network 62.128.0.0 mask 255.255.252.0
network 62.128.4.0 mask 255.255.252.0
network 62.128.8.0 mask 255.255.252.0
network 62.128.12.0 mask 255.255.252.0
network 62.128.16.0 mask 255.255.252.0
network 62.128.20.0 mask 255.255.252.0
network 62.128.24.0 mask 255.255.252.0
network 62.128.28.0 mask 255.255.252.0
network 62.128.32.0 mask 255.255.252.0
network 62.128.36.0 mask 255.255.252.0
network 62.128.40.0 mask 255.255.252.0
network 62.128.48.0 mask 255.255.252.0
network 62.128.52.0 mask 255.255.252.0
network 62.128.56.0 mask 255.255.252.0
network 62.128.60.0 mask 255.255.252.0
network 62.128.64.0 mask 255.255.252.0
network 62.128.68.0 mask 255.255.252.0
network 62.128.72.0 mask 255.255.252.0
network 62.128.76.0 mask 255.255.252.0
aggregate-address 62.128.44.0 255.255.255.252
neighbor 62.128.47.6 remote-as 11151
neighbor 62.128.47.6 filter-list 1 out
neighbor 62.128.47.194 remote-as 645
neighbor 62.128.47.198 remote-as 645
no auto-summary

```

(待续)

```
1
ip as-path access-list 1 deny _645_
ip as-path access-list 1 permit .*
```

可以使用一些方法，利用 AS 路径访问控制列表来过滤网络前缀：

- 在多归路的情况下，`^$`常规表达式用于防止本地的自治系统成为两个上游的服务提供商之间的过渡自治系统，只允许具有空的 AS 路径属性的路由更新数据包被宣告出去。
- 通过使用`^AS$`常规表达式，只给下游的邻居提供部分 BGP RIB 的路由更新。
- 通过使用`_AS_`常规表达式，利用一个 AS 路径的访问控制列表，只允许本地起源的路由更新发送给上游的邻居。
- 通过使用复杂的常规表达式和 AS 路径访问控制列表的组合，过滤接收或者发送方向的路由更新包中含有的 AS 路径。

9.6.4 对于路径操作修改下一跳属性

可以使用几种方法，利用 BGP 属性来操作路由。最简单的一种方法就是修改一条路由的下一跳。就像你在前面的章节所学习到的，每次当路由跨过一个 AS 的边界时，这个路由的下一跳属性就会改变，但是在一个 AS 内部，路由的下一跳属性不会改变。例如，在图 9-14 中，Eany 路由器在 AS 12 512 中，而 Meany、Miney 和 Moe 这 3 台路由器在 AS 61 382 中。

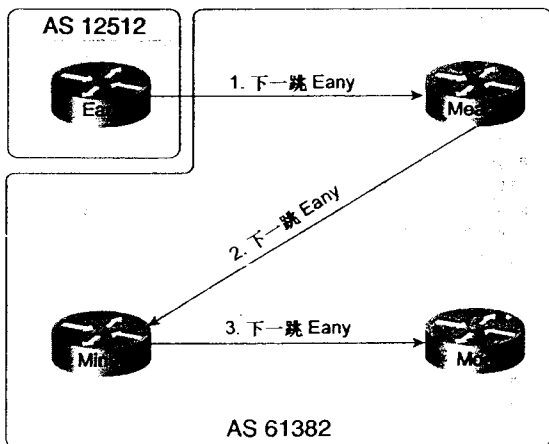


图 9-14 BGP 如何修改下一跳属性

这个图形逻辑表示了当路由通过在不同的自治系统中的路由器时，下一跳属性是如何变化的。首先，路由在 Eany 和 Meany 路由器之间转发时，通过两个自治系统。在这个范例里，路由的下一跳属性在 Eany 路由器的出口改变了，Eany 路由器修改了下一跳属性并且将路由传递给 Meany 路由器。默认情况下，Meany 路由器在将路由传递给 Miney 路由器之前，并不改变下一跳属性的值，这是因为路由始发于一个外部的 AS。当 Miney 路由器将路由通告给 Moe 路由器时，它也不改变下一跳属性，这是因为，除非特别指定，I-BGP 发言人不修改下一跳属性。

注意：下一跳属性在第 7 章的“下一跳属性”一节中详细地介绍过。

当一个 I-BGP 发言人将它通过一个 E-BGP 对等体学到的路由传递给另外一个 I-BGP 发言人时，通常需要改变路由的下一跳属性。除非 I-BGP 会话者已经配置了网关指向它上游的 I-BGP 对等体，否则它将不能到达 E-BGP 路由器的 IP 地址。可用 3 种方法改变这一情况：

- 使用 **neighbor {ip-address | peer-group} default-originate** 命令产生一条默认路由。
- 将 BGP 的路由重分发到 IGP 中（如果 IGP 正在使用）。
- 使用 **next-hop-self** 命令来改变 I-BGP 路由的下一跳属性。

下一跳属性可以使用 **neighbor {ip-address | peer-group} next-hop-self** 命令完成。有时，你可能并不想修改一条外出路由的下一跳属性，在这种情况下，可以使用 **neighbor {ip-address | peer-group} next-hop-unchanged** 命令。所以，你可能会问，如何用其他的方法来改变一条路由的下一跳属性？很简单，可以使用路由映射来改变下一跳属性。

注意：当改变一条路由的下一跳属性时要特别注意。如果那条路径失败了，流量可能不会正确地重新路由。

需要 3 个步骤来手动修改路由的下一跳属性：

第 1 步 建立一个访问控制列表或者前缀列表，在其中指定需要做属性调整的网络。如果到达某个特定的邻居或者对等体组的所有路由都需要做改变的话，这一步可以忽略。

第 2 步 建立一个路由映射来引用在第 1 步中建立的访问控制列表或者前缀列表，并且使用 **set next-hop {ip-address|peer-address| verify-availability}** 命令。

注意：**verify-availability** 命令只能对接收的路由使用。

第 3 步 使用 **neighbor {ip-address | peer-group} route-map route-map-name {in | out}** 命令将路由映射绑定到一个邻居或者对等体组上。

如果两台或者更多台路由器需要添加到 AS 645 中，例如如图 9-15 所示，Arnold 和 Kimberly 路由器需要配置和新的路由器成为邻居，提供被反射的路由，并且对所有外部始发的路由修改下一跳属性。

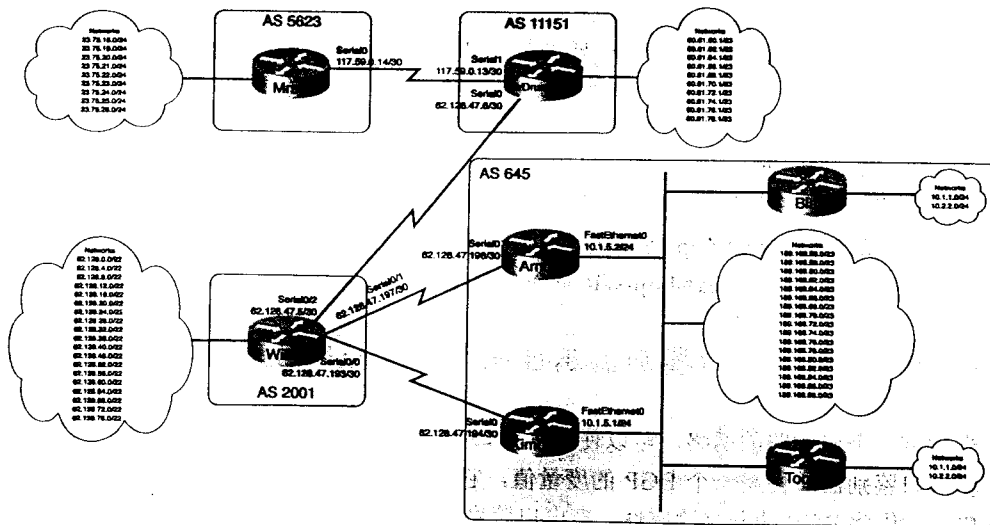


图 9-15 给 Mix 增加两条新的路由

在这个范例中，很容易通过 **next-hop-self** 命令来使得 Blair 和 Tootie 路由器到达外部网络。不过为了说明问题，人为采用路由映射来实现。虽然大多数时候 **next-hop-self** 命令更容易，但是你仍然需要注意修改下一跳地址，而不是采用通过 **next-hop-self** 命令产生的下一跳值。例如，你可能希望下一跳地址指向外部的防火墙，这个防火墙地址没有被 I-BGP 对等体通告；在这种情况下，需要采用路由映射方式来手工设置下一跳地址。范例 9-57 显示了如何通过简单的路由映射来改变与一个特定邻居绑定的所有路由的下一跳属性。

范例 9-57 使用路由映射来修改下一跳属性

```
Arnold# show run | begin bgp
router bgp 645
no synchronization
bgp router-id 10.1.5.2
bgp log-neighbor-changes
network 189.168.56.0 mask 255.255.254.0
network 189.168.58.0 mask 255.255.254.0
network 189.168.60.0 mask 255.255.254.0
network 189.168.62.0 mask 255.255.254.0
network 189.168.64.0 mask 255.255.254.0
network 189.168.66.0 mask 255.255.254.0
network 189.168.68.0 mask 255.255.254.0
network 189.168.70.0 mask 255.255.254.0
network 189.168.72.0 mask 255.255.254.0
network 189.168.74.0 mask 255.255.254.0
network 189.168.76.0 mask 255.255.254.0
network 189.168.78.0 mask 255.255.254.0
network 189.168.80.0 mask 255.255.254.0
network 189.168.82.0 mask 255.255.254.0
network 189.168.84.0 mask 255.255.254.0
network 189.168.86.0 mask 255.255.254.0
network 189.168.88.0 mask 255.255.254.0
neighbor 10.1.5.3 remote-as 645
neighbor 10.1.5.1 route-reflector-client
neighbor 10.1.5.1 next-hop-self
neighbor 10.1.5.3 route-reflector-client
neighbor 10.1.5.3 route-map next-hop out
neighbor 10.1.5.4 remote-as 645
neighbor 10.1.5.4 route-reflector-client
neighbor 10.1.5.4 route-map next-hop out
neighbor 62.128.47.197 remote-as 2001
no auto-summary
!
route-map next-hop permit 10
set ip next-hop 10.1.5.2
```

在先前的范例中，**next-hop** 路由映射用于改变 Arnold 路由器的快速以太接口的下一跳属性。同样的效果也可以用 **next-hop-self** 命令来完成。

9.6.5 使用多出口鉴别器属性来指定最佳路径

当你有一个多归路的网络，可以使用多出口鉴别器属性来指定对于一个 AS 的最佳入口点。多出口鉴别器属性是一个 BGP 的度量值，它可以使 E-BGP 的邻居意识到一个网络的最佳入口点。作为 BGP 的非过渡属性，多出口鉴别器不会宣告到它所直连的 AS 以外，只应用于对等体的基础上。

注意：多出口鉴别器属性的技术文档在第 7 章的“多出口鉴别器属性”一节中介绍过。

只需要三步来设置一个 AS 的多出口鉴别器属性。你可能会可选性地对于每一个 AS 的边界路由器设置不同的多出口鉴别器的值，或者修改 BGP 的决策过程如何使用多出口鉴别器属性。

第 1 步 （可选）建立一个访问控制列表，指定多出口鉴别器属性应当应用在哪些流量上。

第 2 步 建立一个路由映射，对于入口点指定多出口鉴别器的值。默认的多出口鉴别器的值为 0，它可以修改为 1~4 294 967 295 之间的任何一个数值，最低的数值是最好的。这个度量值可以在路由映射的配置模式下使用 `set metric [+|-metric-value]` 命令进行设置。可选的 + 和 - 参数可以改变一个预先存在的度量值。

第 3 步 使用 `neighbor {ip-address | peer-group} route-map route-map-name {in | out}` 命令将路由映射绑定到邻居上。

第 4 步 （可选）使用 `bgp always-compare-med`、`bgp bestpath med confed`、`bgp bestpath med missing-as-worst` 或者 `bgp deterministic-med` 命令来修改 BGP 在最佳路径选择进程中如何使用多出口鉴别器属性来影响其对最佳路径的决策。

表 9-5 显示了每一个命令是如何应用的以及什么时候去使用它们。

表 9-5 最佳路径多出口鉴别器修改

多出口鉴别器命令	命令定义
<code>bgp always-compare-med</code>	允许 BGP 最佳路径选择进程比较属于不同的自治系统的 E-BGP 对等体的多出口鉴别器属性
<code>bgp bestpath med confed</code>	允许 BGP 比较来自联盟对等体的多出口鉴别器属性
<code>bgp bestpath med missing-as-worst</code>	指定在多出口鉴别器属性没有出现的情况下，BGP 应当将这条路径认为是最坏的
<code>bgp deterministic-med</code>	允许 BGP 比较来自同一自治系统中不同的 E-BGP 对等体的多出口鉴别器的值

注意：虽然多出口鉴别器属性可以应用于接收或者发送的路径，你应当总是使用多出口鉴别器来指定对于 E-BGP 对等体的最佳的网络入口点，而本地优先属性指定对于 I-BGP 对等体的最佳的网络出口点。

当你将新的度量值应用到邻居之后，可以在远端邻居上通过 `show ip bgp` 命令来验证它的用途。多出口鉴别器属性显示的是多出口鉴别器的值，如范例 9-58 所示。

范例 9-58 使用 `show ip bgp` 命令来验证多出口鉴别器属性

Willis# show ip bgp regexp ^645\$					
Network	Next Hop	Metric	LocPrf	Weight	Path
* 10.1.1.0/24	62.128.47.194	100		0	645 i
*>	62.128.47.198	50		0	645 i
* 10.2.2.0/24	62.128.47.194	100		0	645 i
*>	62.128.47.198	50		0	645 i
* 189.168.56.0/23	62.128.47.194	100		0	645 i
*>	62.128.47.198	50		0	645 i
* 189.168.58.0/23	62.128.47.194	100		0	645 i
*>	62.128.47.198	50		0	645 i
* 189.168.60.0/23	62.128.47.194	100		0	645 i
*>	62.128.47.198	50		0	645 i
* 189.168.62.0/23	62.128.47.194	100		0	645 i
*>	62.128.47.198	50		0	645 i
* 189.168.64.0/23	62.128.47.194	100		0	645 i

(待续)

>	62.128.47.198	50	0 645 i
* 189.168.66.0/23	62.128.47.194	100	0 645 i
>	62.128.47.198	50	0 645 i
* 189.168.68.0/23	62.128.47.194	100	0 645 i

为了解释多出口鉴别器属性的使用，可以把它应用到处于 AS 645 中的 Arnold 和 Kimberly 路由器上。在多出口鉴别器属性应用到由这两个 AS645 的边界路由器通告的路径之前，Willis 路由器使用由 Kimberly 路由器宣告的路径，原因是它有一个较低的 IP 地址。通过改变多出口鉴别器的属性值，这个属性在 BGP 的决策进程中占有较高的位置，通过将 Arnold 路由器的多出口鉴别器属性修改得比 Kimberly 路由器还要低，到达 AS 645 的最佳路径就会改变。范例 9-59 显示了多出口鉴别器属性是如何在 Arnold 和 Kimberly 路由器上进行修改的。

范例 9-59 对 AS 645 中的 Arnold 和 Kimberly 路由器修改多出口鉴别器属性

```

Arnold# show run | begin bgp
router bgp 645
  no synchronization
  bgp router-id 10.1.5.2
  bgp log-neighbor-changes
  network 189.168.56.0 mask 255.255.254.0
  network 189.168.58.0 mask 255.255.254.0
  network 189.168.60.0 mask 255.255.254.0
  network 189.168.62.0 mask 255.255.254.0
  network 189.168.64.0 mask 255.255.254.0
  network 189.168.66.0 mask 255.255.254.0
  network 189.168.68.0 mask 255.255.254.0
  network 189.168.70.0 mask 255.255.254.0
  network 189.168.72.0 mask 255.255.254.0
  network 189.168.74.0 mask 255.255.254.0
  network 189.168.76.0 mask 255.255.254.0
  network 189.168.78.0 mask 255.255.254.0
  network 189.168.80.0 mask 255.255.254.0
  network 189.168.82.0 mask 255.255.254.0
  network 189.168.84.0 mask 255.255.254.0
  network 189.168.86.0 mask 255.255.254.0
  network 189.168.88.0 mask 255.255.254.0
  neighbor 10.1.5.1 remote-as 645
  neighbor 10.1.5.1 route-reflector-client
  neighbor 10.1.5.1 next-hop-self
  neighbor 10.1.5.3 remote-as 645
  neighbor 10.1.5.3 route-reflector-client
  neighbor 10.1.5.3 next-hop-self
  neighbor 10.1.5.4 remote-as 645
  neighbor 10.1.5.4 route-reflector-client
  neighbor 10.1.5.4 next-hop-self
  neighbor 62.128.47.197 remote-as 2001
  neighbor 62.128.47.197 route-map MED out
  no auto-summary
!
route-map MED permit 10
  set metric 50

Kimberly# show run | begin bgp
router bgp 645
  no synchronization
  bgp router-id 10.1.5.1
  bgp log-neighbor-changes
  network 189.168.56.0 mask 255.255.254.0
    
```

(待续)

```
network 189.168.58.0 mask 255.255.254.0
network 189.168.60.0 mask 255.255.254.0
network 189.168.62.0 mask 255.255.254.0
network 189.168.64.0 mask 255.255.254.0
network 189.168.66.0 mask 255.255.254.0
network 189.168.68.0 mask 255.255.254.0
network 189.168.70.0 mask 255.255.254.0
network 189.168.72.0 mask 255.255.254.0
network 189.168.74.0 mask 255.255.254.0
network 189.168.76.0 mask 255.255.254.0
network 189.168.78.0 mask 255.255.254.0
network 189.168.80.0 mask 255.255.254.0
network 189.168.82.0 mask 255.255.254.0
network 189.168.84.0 mask 255.255.254.0
network 189.168.86.0 mask 255.255.254.0
network 189.168.88.0 mask 255.255.254.0
neighbor 10.1.5.2 remote-as 645
neighbor 10.1.5.2 route-reflector-client
neighbor 10.1.5.2 next-hop-self
neighbor 10.1.5.3 remote-as 645
neighbor 10.1.5.3 route-reflector-client
neighbor 10.1.5.3 next-hop-self
neighbor 10.1.5.4 remote-as 645
neighbor 10.1.5.4 route-reflector-client
neighbor 10.1.5.4 next-hop-self
neighbor 62.128.47.193 remote-as 2001
neighbor 62.128.47.193 route-map MED out
no auto-summary
!
route-map MED permit 10
set metric 100
```

9.6.6 使用本地优先属性来指定网络的出口点

本地优先属性 (LOCAL_PREF) 用于一个 AS 内部，对于不止一条路径可以离开这个 AS 的情况修改最佳的出口点。就像它的名字所意味的那样，本地优先属性只在 I-BGP 对等体之间传递，本地优先属性不会转发给外部的对等体。

注意：有时很难记住本地优先和多出口鉴别器属性之间的不同点。记住属性做什么样的事情的一种简单方法就是看它们的名字——本地优先只应用在本地的对等体之间，而多出口鉴别器属性告诉外部的对等体到达 AS 的最佳入口点。多出口鉴别器属性不比较来自 I-BGP 对等体的路由，而本地优先属性不比较来自 E-BGP 对等体的路由。

就像多出口鉴别器一样，本地优先属性可以使用路由映射，基于对等体来应用。默认的本地优先的属性值为 100，并且它可以改变为 1~4 294 967 295 之间的任意值。最大的本地优先的值总是最好的。需要 3 个步骤来修改一条路径的本地优先的值。

第 1 步 (可选) 建立一个访问控制列表或者前缀列表，指定本地优先将应用的网络。

第 2 步 建立一个路由映射，在路由映射的配置模式下使用 **set local-preference value** 命令来分配本地优先的值。

第 3 步 使用 **neighbor {ip-address|peer-group} route-map route-map-name {in|out}** 命令将路由映射绑定到邻居或者对等体组上。

注意：记住本地优先属性不会传递给外部的对等体，所以，如果你想修改本地优先属性子书仅限试看之用，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

来用于外部的网络，必须将路由映射应用于接收的流量。

为了演示本地优先属性的用法，把它应用在 Arnold 和 Kimberly 路由器上来自 Willis 路由器的所有进入的路由，当它们被传递给 Blair 和 Tootie 路由器时。在这个范例中，Arnold 路由器告诉 Blair 路由器使用来自它的路由，而 Kimberly 路由器告诉 Tootie 路由器使用来自它的路由。Arnold 和 Kimberly 路由器也使用默认的本地优先值将路由发送给其他的路由器。范例 9-60 显示了 Arnold 和 Kimberly 路由器的配置。

范例 9-60 在 Arnold 和 Kimberly 路由器上设置本地优先的配置

```
Arnold# show run | begin bgp
router bgp 645
  no synchronization
  bgp router-id 10.1.5.2

<networks excluded>
  bgp log-neighbor-changes
  neighbor 10.1.5.1 remote-as 645
  neighbor 10.1.5.1 route-reflector-client
  neighbor 10.1.5.1 next-hop-self
  neighbor 10.1.5.3 remote-as 645
  neighbor 10.1.5.3 route-reflector-client
  neighbor 10.1.5.3 next-hop-self
  neighbor 10.1.5.3 route-map local-pref out
  neighbor 10.1.5.4 remote-as 645
  neighbor 10.1.5.4 route-reflector-client
  neighbor 10.1.5.4 next-hop-self
  neighbor 62.128.47.197 remote-as 2001
  no auto-summary
!
route-map local-pref permit 10
  set local-preference 500
```

```
Kimberly# show run | begin bgp
router bgp 645
  no synchronization
  bgp router-id 10.1.5.1
  bgp cluster-id 3181926401
  bgp log-neighbor-changes
  neighbor 10.1.5.2 remote-as 645
  neighbor 10.1.5.2 route-reflector-client
  neighbor 10.1.5.2 next-hop-self
  neighbor 10.1.5.3 remote-as 645
  neighbor 10.1.5.3 route-reflector-client
  neighbor 10.1.5.3 next-hop-self
  neighbor 10.1.5.4 remote-as 645
  neighbor 10.1.5.4 route-reflector-client
  neighbor 10.1.5.4 next-hop-self
  neighbor 10.1.5.4 route-map local-pref out
  neighbor 62.128.47.193 remote-as 2001
  no auto-summary
!
route-map local-pref permit 10
  set local-preference 500
```

在先前的范例中，路由映射 local-pref 将所有从 Arnold 路由器发送给 Blair 路由器的路由的本地优先值设为 500，同样，对于 Kimberly 和 Tootie 路由器也是如此。范例 9-61 显示了 Blair 和 Tootie 路由器的 BGP RIB 的内容。

范例 9-61 当改变本地优先属性后，Blair 和 Tootie 路由器的 BGP RIB 表

```

Blair# show ip bgp regexp _11151_
BGP table version is 95, local router ID is 10.2.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete

```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i23.75.18.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>i	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.19.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>i	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.20.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>i	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.21.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>i	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.22.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>i	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.23.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>j	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.24.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>l	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.25.0/24	10.1.5.1		100	0	2001 11151 5623 i
*>l	10.1.5.2		500	0	2001 11151 5623 i
* i23.75.26.0/24	10.1.5.1		100	0	2001 11151 5623 i
Network	Next Hop				
*>i	10.1.5.2		500	0	2001 11151 5623 I

```

Tootie# show ip bgp regexp _11151_
BGP table version is 307, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i23.75.18.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.19.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.20.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.21.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.22.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.23.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.24.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.25.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i
*>i23.75.26.0/24	10.1.5.1		500	0	2001 11151 5623 i
* i	10.1.5.2		100	0	2001 11151 5623 i

注意在两个范例中，路由器都优选具有较大本地优先属性的路由。Blair 路由器优选来自 Arnold 路由器的路由，而 Tootie 路由器优选来自 Kimberly 路由器的路由。

9.6.7 使用权重属性来影响路径选择

不像多出口鉴别器和本地优先属性，思科专有的权重属性指定的是一个本地最佳的路

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

径，只具有本地的意义，这个属性不会传递给任何对等体。权重属性是一个 0~65 535 之间的值。本地产生的路由默认的权重值为 32 768，所有的其他路由默认的权重值为 0。

设置一条路径的权重需要三步：

- 第 1 步 (可选) 建立一个访问控制列表或者前缀列表，指定用于匹配权重属性的路径。
- 第 2 步 建立一个路由映射来应用访问控制列表或者前缀列表，使用 **set weight value** 命令来设置权重属性。
- 第 3 步 使用 **neighbor {ip-address | peer-group} route-map route-map-name in** 命令将路由映射绑定到邻居或者对等体组上。

注意：即使思科 IOS 软件允许用户利用路由映射来修改发送方向的路由的权重属性，这个命令实际上也是没有效果的，因为权重属性是不会传递给任何对等体的。

例如，假设在这个范例中，Tootie 路由器应当总是使用来自 Kimberly 路由器 (10.1.5.1) 的路由，除非那台路由器失效了。完成这个任务的一种简单方法就是将所有来自 Kimberly 路由器的路由的权重属性设置成一个较高的值。范例 9-62 显示了如何使用权重属性来完成这个任务。

范例 9-62 使用权重属性来设置路由的优先

```
Tootie# show run | begin bgp
router bgp 645
  no synchronization
  bgp log-neighbor-changes
  network 10.1.1.0 mask 255.255.255.0
  network 10.2.2.0 mask 255.255.255.0
  neighbor 10.1.5.1 remote-as 645
  neighbor 10.1.5.1 next-hop-self
  neighbor 10.1.5.1 route-map Heavy-Routes in
  neighbor 10.1.5.2 remote-as 645
  neighbor 10.1.5.2 next-hop-self
  neighbor 10.1.5.3 remote-as 645
  neighbor 10.1.5.3 next-hop-self
  no auto-summary
  !
  route-map Heavy-Routes permit 10
    set weight 150
```

在先前的范例中，路由映射 Heavy-Routes 将权重值设置为 150。这个路由映射接着应用在所有来自 Arnold 路由器的接收方向的路由上，使得这些路由成为最期望的路由，产生了范例 9-63 所示的结果。

范例 9-63 在权重属性修改后，Tootie 路由器的 BGP RIB

```
Tootie# show ip bgp regexp _5623_
BGP table version is 111, local router ID is 10.1.5.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop              Metric LocPrf Weight Path
*>123.75.18.0/25    10.1.5.1                    100   150 2001 11151 5623 i
```

(待续)

* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.19.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.20.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.21.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.22.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.23.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.24.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.25.0/24	10.1.5.1	100	150	2001	11151	5623	i
* i	10.1.5.2	100	0	2001	11151	5623	i
*>i23.75.26.0/24	10.1.5.1	100	150	2001	11151	5623	i
Network	Next Hop	Metric	LocPrf	Weight	Path		
* i	10.1.5.2	100	0	2001	11151	5623	i

注意：在配置这个范例之前，local-pref 路由映射已经从 Kimberly 和 Arnold 路由器上清除了，然而，权重属性依旧比本地优先属性的优先级要高（即使本地优先和权重属性完成相同的事情），这是因为它在 BGP 路径选择进程中的排列顺序要优于本地优先属性。

因为权重属性在 BGP 路径选择进程中是排在第一位的，所以修改权重属性会导致 Tootie 路由器使用具有较高权重属性的路由，它优于具有较高本地优先属性的路由。

9.6.8 团体属性的许多用法

BGP 团体属性是现有的 BGP 属性中非常强大的属性之一。通过团体数值、团体列表或者通过给一条路由增加一个共知的团体值，团体属性可以过滤或者修改路由。可以通过给一条路由设置团体属性以备后用或者匹配一个预定义的团体值来过滤路由。有标准的具有号码方式的团体数值，也有名字方式的团体值，可以使用它们来给一条路径分配一个更可读的团体值。表 9-6 显示了在第 7 章中提到的共知的 BGP 团体属性的回顾。

表 9-6 共知的 BGP 团体值

团体值（十六进制）	团体值（十进制）	团体的名字	描述	思科 IOS 设置团体命令
0x0000000 到 0x0000FFF	0~65535	保留	这个范围的团体属性值已被 IANA 保留	十进制数字的范围为 0~65535 或者 aa: nn 的格式
0xFFFF0000 到 0xFFFFFFFF	4294967041~4294967295	保留	这个范围的团体属性值已被 IANA 保留	十进制数字的范围为 65535~4294967295 或者 aa: nn 的格式
0	0	Internet	默认的团体，所有支持 BGP 团体属性的路由器都属于它	internet
0xFFFFFFF01	4294967041	NO_EXPORT	具有这种团体属性值的路由不能宣告到本地自治系统或者联盟之外	no_export
0xFFFFFFF02	4294967042	NO_ADVERTISE	具有这种团体属性值的路由不能宣告给任何对等体	no_advertise
0xFFFFFFF03	4294967043	LOCAL_AS	具有这种团体属性值的路由不能宣告给任何外部的联盟对等体，在 RFC 1997 中被称为 NO_EXPORTSU-B-CONFED	local-as

需要 5 个步骤来设置 BGP 团体属性的值：

- 第 1 步 （可选）建立一个访问控制列表或者前缀列表，指定将要修改的路径。如果网络没有用 **match** 语句指定，那么路由映射将应用到所有的路由上。
- 第 2 步 建立一个路由映射，指定 **set community** 语句来改变团体属性，使用的命令为 **set community {decimal-number | aa: nn-format | additive | internet | local-as | no-advertise | no-export | none}**。
- 第 3 步 如果你使用的是 aa:nn 的团体格式，确保使用的是 **ip bgp-community new-format** 命令。这个命令可以将思科 IOS 软件显示团体值的默认的十六进制形式改变为新的 aa:nn 的格式。
- 第 4 步 使用 **neighbor {ip-address | peer-group} route-map route-map-name {in | out}** 命令将路由映射绑定到邻居或者对等体组上。
- 第 5 步 使用 **neighbor {ip-address | peer-group} send-community** 命令启用团体属性的宣告。

就像以前所提到的，团体属性值可以在路由映射中使用 **set** 语句来设置，表 9-7 显示了在思科 IOS 软件版本 12.2 (12) T 中在一个路由映射中可以设置的可能的团体值。

表 9-7 路由映射中 set COMMUNITY 命令

命令	描述
Community number in decimal-number format	1 到 4 294 967 295 之间的一个数字
Community number in aa: nn-format	一个 BGP 团体属性的数值，以 aa: nn 的格式表示
additive	给现有的团体属性值增加一个新的数值
internet	将团体属性值设置为共知的 Internet 数值——所有的 BGP 发言人默认的团体值
local-as	一个共知的团体属性，它指定匹配这个团体属性的路径不能宣告到本地自治系统之外
no-advertise	一个共知的团体属性，它指定匹配这个团体属性的路径不能宣告给任何对等体
no-export	一个共知的团体属性，它指定匹配这个团体属性的路径不能宣告给任何外部对等体
none	清除团体属性

下一个范例显示了用户如何使用 BGP NO_EXPORT 团体属性来防止一个 BGP 邻居传播一条特定的路由。在这个范例中，Arnold 路由器正在宣告带有 BGP 共知的团体值 NO_EXPORT 的两条网络 10.1.1.0/24 和 10.2.2.0/24。范例 9-64 显示了 Arnold 路由器的 BGP 配置。

在先前的范例中，配置 Arnold 路由器来宣告带有 NO_EXPORT 团体属性的 10.1.1.0/24 和 10.2.2.0/24 网络，通过建立“community”的路由映射来引用 local-list 的前缀列表，这个前缀列表又引用了 10.1.1.0/24 和 10.2.2.0/24 两条网络。使用 **set community no-export** 命令给这两条网络分配 NO_EXPORT 的团体属性值，这个路由映射绑定到邻居 62.128.47.197 上，即 Willis 路由器，并且 BGP 团体属性通告是使用 **send-community** 命令来启用的。范例 9-65 显示了这个配置在 Willis 和 MrDrummand 路由器上的效果。

注意 Willis 路由器现在显示的是路由 *not advertised to EBGp peer*（不宣告给 EBGp 对等体）。这就是 NO_EXPORT 团体属性应用的直接效果。还要注意，MrDrummand 路由器在变

范例 9-64 使用 BGP 共知的 NO_EXPORT 团体属性

```
Arnold# show run | begin bgp
router bgp 645
no synchronization
bgp router-id 10.1.5.2
bgp log-neighbor-changes
neighbor 10.1.5.1 remote-as 645
neighbor 10.1.5.1 route-reflector-client
neighbor 10.1.5.1 next-hop-self
neighbor 10.1.5.3 remote-as 645
neighbor 10.1.5.3 route-reflector-client
neighbor 10.1.5.3 next-hop-self
neighbor 10.1.5.4 remote-as 645
neighbor 10.1.5.4 route-reflector-client
neighbor 10.1.5.4 next-hop-self
neighbor 62.128.47.197 remote-as 2001
neighbor 62.128.47.197 send-community
neighbor 62.128.47.197 route-map community out
no auto-summary
!
ip prefix-list local-list seq 5 permit 10.1.1.0/24
ip prefix-list local-list seq 10 permit 10.2.2.0/24
!
route-map community permit 10
match ip address prefix-list local-list
set community no-export
```

范例 9-65 在团体过滤后，Willis 路由器的 BGP RIB 表项

```
Willis# show ip bgp 10.1.1.0/24
BGP routing table entry for 10.1.1.0/24, version 191
Paths: (2 available, best #2, table Default-IP-Routing-Table, not advertised to
EBGP peer)
Not advertised to any peer
645
62.128.47.194 from 62.128.47.194 (10.1.5.1)
Origin IGP, metric 100, localpref 100, valid, external
645
62.128.47.198 from 62.128.47.198 (10.1.5.2)
Origin IGP, localpref 100, valid, external, best
Community: no-export
Willis# show ip bgp 10.2.2.0/24
BGP routing table entry for 10.2.2.0/24, version 192
Paths: (2 available, best #2, table Default-IP-Routing-Table, not advertised to
EBGP peer)
Not advertised to any peer
645
62.128.47.194 from 62.128.47.194 (10.1.5.1)
Origin IGP, metric 100, localpref 100, valid, external
645
62.128.47.198 from 62.128.47.198 (10.1.5.2)
Origin IGP, localpref 100, valid, external, best
Community: no-export

MrDrummand# show ip bgp 10.1.1.0/24
% Network not in table
MrDrummand# show ip bgp 10.2.2.0/24
% Network not in table
```

化发生后，没有收到任何 10.1.1.0/24 或者 10.2.2.0/24 网络的通告。先前的范例中演示了 BGP 团体属性如何使用共知的团体值来过滤路由。下面的小节将告诉用户如何使用 BGP 团体列表来指定匹配多个 BGP 团体值的路由。

团体列表

BGP 团体列表提供了一种方法来指定要匹配的 BGP 团体属性值的列表。有 4 种不同类型的 BGP 团体列表，列表类型、命令语法和描述显示在表 9-8 中。

表 9-8 团体列表指南

团体列表类型	语法	描述
标准带号码方式的	<code>ip community-list number {permit deny} {decimal-number aa: nn-number internet local-as no-advertise no-export}</code>	一个带号码方式的访问控制列表，范围从 1~99，它列出来 BGP 团体值，以数字的方式或者共知的名字方式列出来
扩展带号码方式的	<code>ip community-list number {permit deny} regular-expression</code>	一个带号码方式的访问控制列表，范围从 100~199，它使用常规表达式来列出 BGP 的团体值
标准带名字方式的	<code>ip community-list standard list-name {permit deny} {decimal-number aa: nn-number internet local-as no-advertise no-export}</code>	一个带名字方式的访问控制列表，范围从 1~99，它列出来 BGP 团体值，以数字的方式或者共知的名字方式列出来
扩展带名字方式的	<code>ip community-list expanded list-name {permit deny} regular-expression</code>	一个带名字方式的访问控制列表，范围从 100~199，它使用常规表达式来列出 BGP 的团体值

`show ip community-list` 命令允许用户显示本地的团体列表的配置，而 `show ip bgp community community` 命令列出来 RIB 表中任何匹配特定的团体值的 BGP 路径。`show ip bgp community-list {list-name | list-number}` 命令显示匹配特定的团体列表的 BGP RIB 表项。范例 9-66 显示了每一种团体列表类型的样例。

范例 9-66 团体列表的样例

```
ip community-list 1 permit no-export
ip community-list 100 permit ^645
ip community-list standard my-community permit local_as
ip community-list expanded your-community permit 645$
```

第一个团体列表匹配在 RIB 表中具有 NO_EXPORT 团体属性值的表项。第二个团体列表即列表 100，匹配任何团体值开始于字符串 645 的 RIB 表项。第三个团体列表即列表 my-community，匹配任何团体值为 LOCAL-AS 的 RIB 表项，最后一个团体列表匹配任何团体值结束于 645 的 RIB 表项。团体列表是在路由映射中使用 `match` 语句来指定的。表 9-9 显示了团体中 `match` 命令和它们的描述。

表 9-9 共知的 BGP 团体 match 语句

命令	描述
<code>match community {standard-list-number expanded-list-number list-name}</code>	匹配一个预定义的团体列表：标准的团体列表的范围从 1~99，扩展的团体列表的范围从 100~199
<code>match extcommunity {standard-list-number expanded-list-number list-name}</code>	匹配扩展的多协议的 BGP 团体列表：标准的团体列表的范围从 1~99，扩展的团体列表的范围从 100~199

下一个范例显示了如何使用 BGP 团体属性来设置和过滤 BGP 的团体值。在范例 9-67 中，你可以看到 Kimberly 路由器正在使用 community 路由映射来设置两个团体。

范例 9-67 在 Kimberly 路由器上使用路由映射设置团体值

```
Kimberly# show run | begin bgp
router bgp 645
  no synchronization
  bgp router-id 10.1.5.1
  bgp log-neighbor-changes
  network 189.168.56.0 mask 255.255.254.0
  network 189.168.58.0 mask 255.255.254.0
  network 189.168.60.0 mask 255.255.254.0
  network 189.168.62.0 mask 255.255.254.0
  neighbor 10.1.5.2 remote-as 645
  neighbor 10.1.5.2 route-reflector-client
  neighbor 10.1.5.2 next-hop-self
  neighbor 10.1.5.3 remote-as 645
  neighbor 10.1.5.3 route-reflector-client
  neighbor 10.1.5.3 next-hop-self
  neighbor 10.1.5.4 remote-as 645
  neighbor 10.1.5.4 route-reflector-client
  neighbor 10.1.5.4 next-hop-self
  neighbor 62.128.47.193 remote-as 2001
  neighbor 62.128.47.193 send-community
  neighbor 62.128.47.193 route-map community out
  no auto-summary
!
ip bgp-community new-format
!
ip prefix-list 1 seq 5 permit 189.168.56.0/22
!
ip prefix-list 2 seq 5 permit 189.168.60.0/22
!
route-map community permit 10
  match ip address prefix-list 1
  set community 645:100
!
route-map community permit 20
  match ip address prefix-list 2
  set community 645:200
!
route-map community permit 30
```

在先前的范例中，Kimberly 路由器使用路由映射 community 将 189.168.56.0/22 网络的团体值设置为 645:100，将 189.168.60.0/22 网络的团体值设为 645:200。接着这个 community 的路由映射就应用到 62.128.47.193 这个邻居上，即 Willis 路由器，并且团体属性是使用 **send-community** 命令发送的。范例 9-68 显示了 Willis 路由器是如何使用由 Kimberly 路由器通告过来的团体值来过滤路由的。

在先前的范例中，Willis 路由器使用路由映射 use-community 序列号为 10 的语句来匹配含有团体值 645:100 的路由，并且使用 NO_ADVERTISE 这个团体属性来通告它们。这个路由映射的序列号 20 也将任何含有团体值 645:200 的路由通告为 NO-EXPORT 的团体属性值；所有其他的路由都被设置为默认的 Internet 团体值。use-community 的路由映射接着绑定到 MrDrummand 路由器 (62.128.47.6) 上。可以在 MrDrummand 路由器上使用 **show ip bgp ip-prefix** 命令来进行验证，如范例 9-69 所示。

范例 9-68 在 Willis 路由器上使用团体属性来过滤路由

```

Willis# show run | begin bgp
router bgp 2001
  no synchronization
  bgp log-neighbor-changes
  neighbor 62.128.47.6 remote-as 11151
  neighbor 62.128.47.6 send-community
  neighbor 62.128.47.6 route-map use-community out
  neighbor 62.128.47.194 remote-as 645
  neighbor 62.128.47.198 remote-as 645
  no auto-summary
!
ip bgp-community new-format
ip community-list 1 permit 645:100
ip community-list 2 permit 645:200
!
route-map use-community permit 10
  match community 1
  set community no-advertise
!
route-map use-community permit 20
  match community 2
  set community no-export
!
route-map use-community permit 30
  set community internet

```

范例 9-69 MrDrummand 路由器上的 BGP RIB 表项

```

MrDrummand# show ip bgp 189.168.56.0/23
BGP routing table entry for 189.168.56.0/23, version 137
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to
any peer)
  Not advertised to any peer
    2001 645
      62.128.47.5 from 62.128.47.5 (62.128.76.1)
        Origin IGP, localpref 100, valid, external, best
        Community: no-advertise
MrDrummand# show ip bgp 189.168.58.0/23
BGP routing table entry for 189.168.58.0/23, version 138
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to
any peer)
  Not advertised to any peer
    2001 645
      62.128.47.5 from 62.128.47.5 (62.128.76.1)
        Origin IGP, localpref 100, valid, external, best
        Community: no-advertise
MrDrummand# show ip bgp 189.168.60.0/23
BGP routing table entry for 189.168.60.0/23, version 115
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to
EBGP peer)
  Not advertised to any peer
    2001 645
      62.128.47.5 from 62.128.47.5 (62.128.76.1)
        Origin IGP, localpref 100, valid, external, best
        Community: no-export
MrDrummand# show ip bgp 189.168.62.0/23
BGP routing table entry for 189.168.62.0/23, version 116
Paths: (1 available, best #1, table Default-IP-Routing-Table, not advertised to

```

(待续)

```
EBGP peer)
Not advertised to any peer
2001 645
  62.128.47.5 from 62.128.47.5 (62.128.76.1)
  Origin IGP, localpref 100, valid, external, best
  Community: no-export
```

就像你所看到的，MrDrummand 路由器从 Willis 路由器收到了设置了属性值的路由。MrDrummand 当前没有通告 189.168.56.0/22 这条路由，是因为这条路由被标记为 **no-advertise**，到 189.168.60.0/22 网络的路由也没有通告，因为 MrDrummand 路由器没有任何 I-BGP 的邻居，使其可以转发这个 NO-EXPORT 属性。

下一个范例显示了团体如何使得用户改变其他的 BGP 属性。在这个范例中，Kimberly 路由器正在给 Willis 路由器发送含有 645：600 BGP 团体属性值的路由。

范例 9-70 Kimberly 路由器的配置

```
Kimberly# show run | begin bgp
router bgp 645
  no synchronization
  bgp router-id 10.1.5.1
  bgp log-neighbor-changes
<networks omitted>
  neighbor 10.1.5.2 remote-as 645
  neighbor 10.1.5.2 route-reflector-client
  neighbor 10.1.5.2 next-hop-self
  neighbor 10.1.5.3 remote-as 645
  neighbor 10.1.5.3 route-reflector-client
  neighbor 10.1.5.3 next-hop-self
  neighbor 10.1.5.4 remote-as 645
  neighbor 10.1.5.4 route-reflector-client
  neighbor 10.1.5.4 next-hop-self
  neighbor 62.128.47.193 remote-as 2001
  neighbor 62.128.47.193 send-community
  neighbor 62.128.47.193 route-map change-attr out
  no auto-summary
!
ip bgp-community new-format
!
route-map change-attr permit 10
  set community 645:600
```

就像你可以看到的，Kimberly 路由器使用路由映射 **change-attr** 将所有路由更新的团体属性值设置为 645：600。那个路由映射接着绑定到 Willis 路由器（62.128.47.193）上。当 Willis 路由器收到路由后，一个本地路由映射匹配了 645：600 团体值的路由，并为所有来自 Kimberly 路由器的路由设置本地优先的值，如范例 9-71 所示。

范例 9-71 使用团体属性来改变本地优先属性

```
Willis# show run | begin bgp
router bgp 2001
  no synchronization
  bgp log-neighbor-changes
  neighbor 62.128.47.6 remote-as 11151
  neighbor 62.128.47.194 remote-as 645
  neighbor 62.128.47.194 route-map change-pref in
```

(待续)

```
neighbor 62.128.47.198 remote-as 645
no auto-summary
!
ip bgp-community new-format
ip community 11st standard change-pref permit 645:600,000:100,000,000:100,000,000
!
route-map change-pref permit 10
match community standard change-pref1
set local-preference 250
```

在先前的范例中，Willis 路由器使用 change-pref 路由映射将所有来自 Kimberly 路由器的接收路由的本地优先属性值设置为 250。这使得 Willis 路由器优选使用 Kimberly 路由器到达所有位于 AS 645 的网络。范例 9-72 显示了 Willis 路由器上对于 AS 645 的 BGP RIB 表。

范例 9-72 Willis 路由器的本地 BGP RIB 表

Willis# show ip bgp regexp ^645\$						
Network	Next Hop	Metric	LocPrf	Weight	Path	
*> 10.1.1.0/24	62.128.47.194		250	0	645	i
*	62.128.47.198			0	645	i
*> 10.2.2.0/24	62.128.47.194		250	0	645	i
*	62.128.47.198			0	645	i
* 189.168.56.0/23	62.128.47.198	0		0	645	i
*>	62.128.47.194	0	250	0	645	i
* 189.168.58.0/23	62.128.47.198	0		0	645	i
*>	62.128.47.194	0	250	0	645	i
* 189.168.60.0/23	62.128.47.198	0		0	645	i
*>	62.128.47.194	0	250	0	645	i
* 189.168.62.0/23	62.128.47.198	0		0	645	i
*>	62.128.47.194	0	250	0	645	i
* 189.168.64.0/23	62.128.47.198	0		0	645	i
*>	62.128.47.194	0	250	0	645	i
* 189.168.66.0/23	62.128.47.198	0		0	645	i
*>	62.128.47.194	0	250	0	645	i
* 189.168.68.0/23	62.128.47.198	0		0	645	i

9.7 使用多路径

在一个企业级的 BGP 网络中，一个网络有一个或多个服务提供商是一件非常普通的事情。可以将多归路的网络配置为以下 3 种：

- 一台路由器有多条链路连接到同一个服务提供商上；
- 一台路由器有多条链路连接到多个服务提供商上；
- 不止一台路由器多归路到一个服务提供商上；
- 不止一台路由器多归路到多个服务提供商上。

虽然有许多方法可以配置一个多归路的网络，但是每次最好总是遵循相同的规则。在多归路的网络中仔细规划可以获得最好的效果，在试图多归路你的网络之前，你总是想要验证上游的服务提供商支持你的配置。许多服务提供商都有 BGP 的策略，将这些策略提供给想做多归路网络的用户，这里列出了一些策略：

- 使用 **ebgp-multihop** 命令（具有或者不具有负载均衡）；
- 支持的 BGP 属性的列表；

- 公有 IP 地址和 AS 号码策略；
- 服务提供商 IP 地址和私有 ASN 号码的使用；
- 路由过滤策略；
- 路由聚合策略（许多服务提供商将不接受小于/24 的路由）；
- BGP 的版本号；
- 验证的方法、策略和口令；
- 路由惩罚的策略。

当你决定网络需求并且获取了所有必需的地址和电路以后，你可以开始设计你的多归路解决方案。因为环回地址是永远都不会失效的，它们通常被用作多归路的导航设备。多归路的一个最常用的实践经验就是使用环回地址作为 BGP 的更新源。多归路网络的另外一个需求就是 AS 路径的过滤——你不想让上游的服务提供商将你的网络作为过渡的 AS。你也必须过滤任何私有的地址空间，并且在通告之前聚合你的内部网络。需要几种基本的任务来实现一个多归路的网络：

- 第 1 步 建立 E-BGP 对等体的路由，你的网络非常可能是和不在你的管理控制之下的一台路由器建立对等的关系，所以你必须提前规划一个路由策略。
- 第 2 步 如果路由器和另外一台路由器有不止一个的连接，你应当给远端的服务提供商提供环回接口的地址，并且使用这个环回接口的地址作为你的更新源。这可以通过 **neighbor {ip-address | peer-group} update-source interface-name interface-number** 命令来完成。如果你准备使用 **update-source** 命令，最好也配置那台路由器使用这个 IP 地址作为 BGP 的 router ID，可以使用 **bgp router-id ip-address** 命令并且指定环回的 IP 地址。
- 第 3 步 如果因为你使用了环回接口，而导致你没有和一台路由器直接进行连接的话，那么必须使用 **neighbor {ip-address | peer-group} ebgp-multihop number-of-hops** 命令。因为当你使用这个命令时，可以指定允许几跳的数量，所以需要特别注意，你的服务提供商可能会终止你的跨过两跳以外试图到达另外一个接口的流量。当使用 **ebgp-multihop** 命令时，总是指定最大的跳数。
- 第 4 步 如果你准备使用不止一个接口实现负载分担的话，使用 **maximum-paths number-of-paths** 命令。这个命令允许 BGP 的进程使用多条路径，而不是一条最佳路径实现负载分担。
- 第 5 步 如果你正在使用不止一台路由器作为过渡的对等点，那么在 I-BGP 对等体之间使用 **next-hop-self** 命令，以使在路由中通告的是一个可达的下一跳属性。
- 第 6 步 如果你正在使用不止一台路由器和不止一个服务提供商进行对等连接的话，那么使用只包含空的 AS 路径（**^\$**）的 AS 路径列表来过滤所有的外部路由，这可以防止某个服务提供商使用你的 AS 作为一个过渡的 AS，来到达另外一个服务提供商的网络。
- 第 7 步 验证你的路由器没有传播任何私有的 RFC 1918 的地址，使用访问控制列表和分发列表（**distribute list**）或者路由映射来指定这个私有的地址。
- 第 8 步 在宣告路由给上游的服务提供商之前，执行路由聚合。为了节省因特网路由表的空间，总是尽可能发送最小的前缀。
- 第 9 步 配置 BGP 属性用于路径选择和路由策略。对于 I-BGP 的路由，设置本地优先。

对于 E-BGP AS 的出口，设置多出口鉴别器，并且设置任何你在路由策略中需要用到的团体属性。

例如，看看图 9-16 所示的网络。在这个范例中，Internal_Border 路由器和它的上游路由器即 External 路由器有两条连接。为了使得 Internal_Border 路由器可以成功地同时使用两条串行链路，它必须配置使用先前列出的步骤。

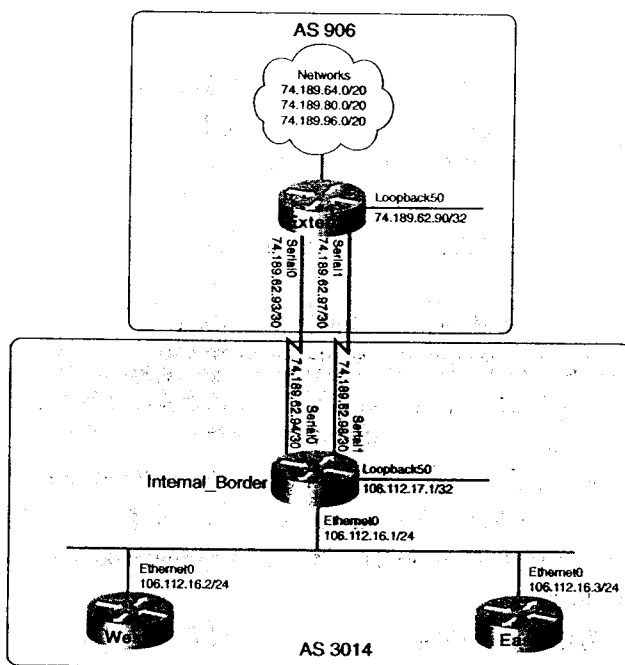


图 9-16 和同一个服务提供商实现多归路

范例 9-73 显示了在 Internal_Border 和 External 边界路由器上的配置。在这个范例中，Internal_Border 路由器使用了环回接口 50 和 External 路由器的环回接口建立对等关系。

范例 9-73 在 Internal_Border 路由器和 External 路由器之间实现多归路的连接

```
External# show run | begin bgp
router bgp 906
no synchronization
bgp router-id 74.189.62.90
network 74.189.62.92 mask 255.255.255.252
network 74.189.62.96 mask 255.255.255.252
network 74.189.64.0 mask 255.255.240.0
network 74.189.80.0 mask 255.255.240.0
network 74.189.96.0 mask 255.255.240.0
neighbor 106.112.17.1 remote-as 3014
neighbor 106.112.17.1 ebgp-multihop 2
neighbor 106.112.17.1 update-source Loopback50
no auto-summary
!
ip route 106.112.17.1 255.255.255.255 74.189.62.94
ip route 106.112.17.1 255.255.255.255 74.189.62.98
```

```
Internal_Border # show run | begin bgp
```

(待续)

```

router bgp 3014
no synchronization
bgp router-id 106.112.17.1
bgp log-neighbor-changes
network 106.112.16.0 mask 255.255.255.0
aggregate-address 106.112.16.0 255.255.248.0 summary-only
neighbor 74.189.62.90 remote-as 906
neighbor 74.189.62.90 ebgp-multihop 2
neighbor 74.189.62.90 update-source Loopback50
neighbor 106.112.16.2 remote-as 3014
neighbor 106.112.16.2 route-reflector-client
neighbor 106.112.16.2 next-hop-self
neighbor 106.112.16.3 remote-as 3014
neighbor 106.112.16.3 route-reflector-client
neighbor 106.112.16.3 next-hop-self
no auto-summary
!
ip route 74.189.62.90 255.255.255.255 74.189.62.93
ip route 74.189.62.90 255.255.255.255 74.189.62.97

```

先前的范例中显示了 External 路由器如何使用 **ebgp-multihop 2** 命令来指定远端的邻居 74.189.62.90 可能在两跳以外。**update-source loopback 50** 命令告诉路由器使用 loopback50 接口的 IP 地址作为 BGP 报文的 IP 地址。当使用这个命令时，update-source 的接口，通常也是环回接口，也被通告为所有路由的下一跳地址。External 和 Internal_Border 路由器都需要路由来告诉它们如何对于 BGP 的会话找到环回接口。

同时，也要注意 Internal_Border 路由器的配置。就像 External 路由器一样，Internal_Border 路由器使用了 **ebgp-multihop 2** 和 **update-source loopback 50** 命令来指定路由器将使用它的 loopback50 的 IP 地址来发送 BGP 报文，同时它也指定远端对等体的 IP 地址距它自己有两跳这么远。Internal_Border 路由器也配置了通告它的 Ethernet0 IP 地址作为发送给位于 AS 3014 中 East 和 West 的 I-BGP 对等体的路由更新的下一跳，这些路由器也是路由反射器的客户方。Internal_Border 路由器在发送路由给 External 路由器之前，也对所有通告的网络进行了路由聚合。范例 9-74 显示了 External 路由器的路由表。

范例 9-74 External 路由器的路由表

```

External# show ip bgp | begin Network
  Network          Next Hop          Metric LocPrf Weight Path
*> 74.189.62.92/30  0.0.0.0              0           32768 i
*> 74.189.62.96/30  0.0.0.0              0           32768 i
*> 74.189.64.0/20   0.0.0.0              0           32768 i
*> 74.189.80.0/20   0.0.0.0              0           32768 i
*> 74.189.96.0/20   0.0.0.0              0           32768 i
*> 106.112.16.0/21  106.112.17.1          0 3014 i
External# show ip route | include vialis
  106.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B    106.112.16.0/21 [20/0] via 106.112.17.1, 00:00:43
S    106.112.17.1/32 [1/0] via 74.189.62.98
      [1/0] via 74.189.62.94
  74.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C    74.189.62.90/32 is directly connected, Loopback50
C    74.189.62.92/30 is directly connected, Serial0
C    74.189.96.0/20 is directly connected, Loopback30
C    74.189.80.0/20 is directly connected, Loopback20
C    74.189.62.96/30 is directly connected, Serial1
C    74.189.64.0/20 is directly connected, Loopback10

```

在这个范例中，你可以看到到 106.112.16.0/21 网络的路由可以通过 74.189.62.94 或者 74.189.62.98 的下一跳 IP 地址到达。因此，如果一个接口失效了，另外一个接口可以在很小的或者根本没有中断的情况下继续进行 BGP 的路由。范例 9-75 显示了 **debug ip routing** 命令在一个仿真的接口失效的情况下的输出。

注意：在生产性的路由器上执行 **debug** 命令要特别注意。试图使用访问控制列表，关闭控制台的记录，并且使用一个日志服务器来捕捉日志的输出，这些做法可以限制命令的输出。在生产性的路由器上使用调试的命令很容易就可以使整台路由器崩溃。

范例 9-75 在一个接口故障的情况下，debug 的输出

```
Internal_Border(config)# interface serial0
Internal_Border(config-if)# shutdown
01:59:37: is_up: 0 state: 6 sub state: 1 line: 0
01:59:37: RT: interface Serial0 removed from routing table
01:59:37: RT: del 74.189.62.92/30 via 0.0.0.0, connected metric [0/0]
01:59:37: RT: delete subnet route to 74.189.62.92/30
Comment: routes using Serial 0 interface are removed
01:59:37: RT: add 74.189.62.92/30 via 74.189.62.90, bgp metric [20/0]
01:59:38: RT: del 74.189.62.90/32 via 74.189.62.93, static metric [1/0]
Comment: route to External router loopback over Serial 0 is removed
01:59:39: %LINK-5-CHANGED: Interface Serial0, changed state to administratively
down
01:59:39: is_up: 0 state: 6 sub state: 1 line: 0
01:59:40: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state
to down
01:59:40: is_up: 0 state: 6 sub state: 1 line: 0
01:59:41: RT: del 74.189.62.92/30 via 74.189.62.90, bgp metric [20/0]
01:59:41: RT: delete subnet route to 74.189.62.92/30
00:47:14: RT: del 74.189.64.0/20 via 74.189.62.90, bgp metric [20/0]
00:47:14: RT: delete subnet route to 74.189.64.0/20
00:47:14: RT: del 74.189.80.0/20 via 74.189.62.90, bgp metric [20/0]
00:47:14: RT: delete subnet route to 74.189.80.0/20
00:47:14: RT: del 74.189.96.0/20 via 74.189.62.90, bgp metric [20/0]
00:47:14: RT: delete subnet route to 74.189.96.0/20
00:47:38: RT: del 74.189.62.90/32 via 74.189.62.93, static metric [1/0]
00:47:38: RT: del 74.189.62.90/32 via 74.189.62.93, static metric [1/0]
00:48:14: RT: add 74.189.64.0/20 via 74.189.62.90, bgp metric [20/0]
00:48:14: RT: add 74.189.80.0/20 via 74.189.62.90, bgp metric [20/0]
00:48:14: RT: add 74.189.96.0/20 via 74.189.62.90, bgp metric [20/0]
```

范例 9-76 显示了在 Internal_Border 路由器上当接口出现问题时路由器的路由表。注意，所有的路由依旧出现在路由表中，并且指向环回接口；惟一的变化就是路由指向环回接口。

范例 9-76 当接口出现问题时，路由器的路由表

```
Internal_Border# show ip route
106.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B    106.112.16.0/21 [200/0] via 0.0.0.0, 00:13:18, Null0
C    106.112.16.0/24 is directly connected, Ethernet0
C    106.112.17.0/24 is directly connected, Loopback50
S    74.189.62.90/32 [1/0] via 74.189.62.97
B    74.189.96.0/20 [20/0] via 74.189.62.90, 00:45:00
B    74.189.80.0/20 [20/0] via 74.189.62.90, 00:45:00
C    74.189.62.96/30 is directly connected, Serial1
B    74.189.64.0/20 [20/0] via 74.189.62.90, 00:45:01
```

9.8 实际范例：多归路一个 BGP 网络

本范例演示了在台路由器具备多个路径通往两个服务提供商的环境下实现 BGP 网络多归路所需的全部任务。演示了多归路相关的命令，以及这些命令在实际情况下是如何使用的。图 9-17 显示了在本范例中所采用的网络。

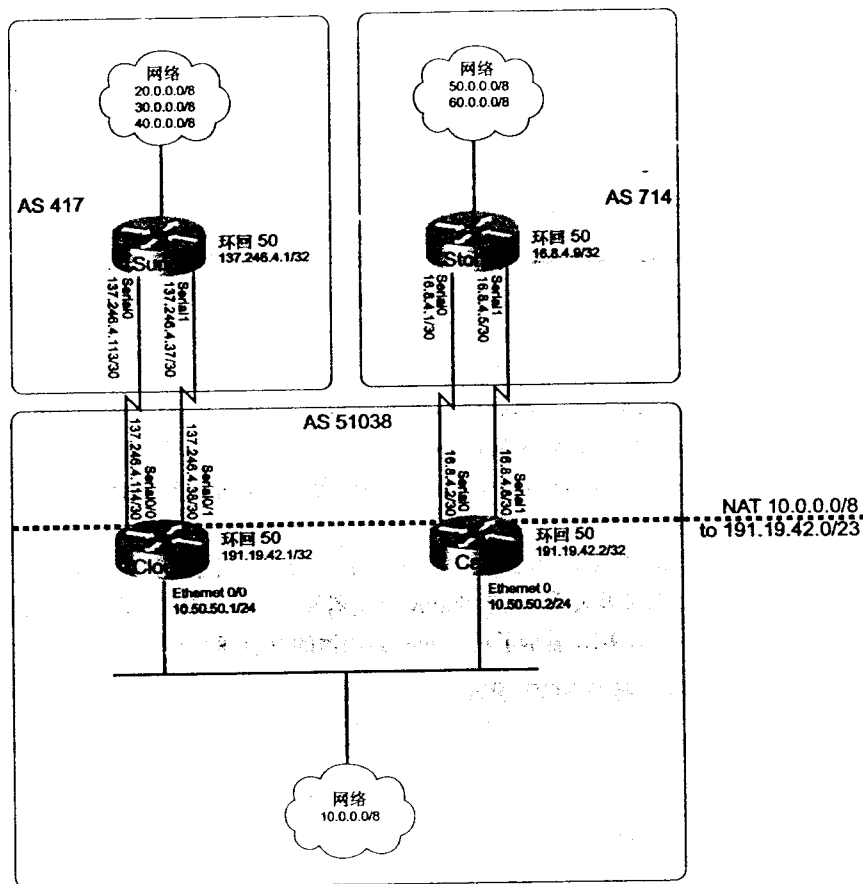


图 9-17 All-Weather 网络

这个范例需要 4 台思科的路由器，每一台路由器都需要两个串行接口，其中两台路由器还需要以太接口。这个范例中的路由器使用了如表 9-10 所示的 IP 地址和接口的分配。

第 1 步 配置 IP 地址，并且在进行第 2 步之前，验证每一台路由器都能够 ping 通它直连的下一跳。

第 2 步 在配置 BGP 之前，确保 Sunny 和 Cloudy 路由器能够连通彼此的 Loopback50 的 IP 地址。不要使用 IGP 的协议。下面的范例显示了配置在两台路由器上的路由。范例 9-77 显示了配置在 Sunny 和 Cloudy 路由器上的静态路由。

表 9-10

IP 地址和接口的分配

路由器	接口名字/号码	IP 地址	路由器	接口名字/号码	IP 地址
Sunny	Loopback5	20.0.0.1/8	Stormy	Serial0	16.8.4.1/30
	Loopback10	30.0.0.1/8		Serial1	16.8.4.5/30
	Loopback15	40.0.0.1/8	Cloudy	Ethernet0/0	10.50.50.1/24
	Loopback50	137.246.4.1/32		Serial0/0	137.246.4.114/30
	Serial0	137.246.4.113/30		Serial0/1	137.246.4.38/30
	Serial1	137.246.4.37/30		Loopback50	191.19.42.1/32
Stormy	Loopback5	50.0.0.1/8	Calm	Ethernet0/0	10.50.50.2/24
	Loopback10	60.0.0.1/8		Loopback50	191.19.42.2/32
	Loopback15	70.0.0.1/8		Serial0	16.8.4.2/30
	Loopback50	16.8.4.9/32		Serial1	16.8.4.6/30

范例 9-77 在 Sunny 和 Cloudy 路由器上配置静态路由

```
Sunny# show run | begin ip route
ip route 191.19.42.1 255.255.255.255 137.246.4.114
ip route 191.19.42.1 255.255.255.255 137.246.4.38

Cloudy# show run | begin ip route
ip route 137.246.4.1 255.255.255.255 137.246.4.37
ip route 137.246.4.1 255.255.255.255 137.246.4.113
```

在这个范例中，每一台路由器上都添加了两条非常特殊的静态路由，允许路由器无需指定整个网络前缀就可以到达对方的环回接口。

第 3 步 在 Sunny 路由器上配置 BGP 的路由。将这台路由器分配到 ASN 417 中，并且使用 Loopback50 的 IP 地址作为 BGP router ID。也要关闭掉自动汇总。使用 **network** 语句，以环回地址作为下一跳，宣告 3 条网络。Sunny 路由器应当使用环回接口和 Cloudy 路由器建立对等关系。配置 Sunny 路由器和 Cloudy 路由器的 Loopback50 接口成为对等体。范例 9-78 显示了对 Sunny 路由器的 BGP 配置。

范例 9-78 Sunny 路由器的 BGP 配置

```
Sunny# show run | begin bgp
router bgp 417
 synchronization
  bgp router-id 137.246.4.1
  bgp log-neighbor-changes
  network 20.0.0.0
  network 30.0.0.0
  network 40.0.0.0
  neighbor 191.19.42.1 remote-as 51038
  neighbor 191.19.42.1 ebgp-multihop 2
  neighbor 191.19.42.1 update-source Loopback50
  no auto-summary
!
ip route 191.19.42.1 255.255.255.255 137.246.4.114
ip route 191.19.42.1 255.255.255.255 137.246.4.38
```

在先前的范例中，BGP 被配置使用了 **ebgp-multihop** 命令，允许 E-BGP 邻居之间的跳数有两跳之远，环回接口使用了 **update-source** 命令来指定，而 BGP router ID 是使用 **bgp router-id**

命令来改变的。

第 4 步 配置 Cloudy 路由器运行在 AS 51 038 中，并且配置这台路由器和 Sunny 路由器的环回接口建立对等关系。验证两台路由器可以成功地启用和维护 BGP 的会话。范例 9-79 显示了在 Cloudy 路由器上的 BGP 配置，这个范例也显示了在 Cloudy 和 Sunny 路由器上 **show ip bgp summary** 命令的输出。

范例 9-79 Cloudy 路由器的 BGP 配置

```
Cloudy# show run | begin bgp
router bgp 51038
  synchronization
  bgp router-id 191.19.42.1
  bgp log-neighbor-changes
  neighbor 137.246.4.1 remote-as 417
  neighbor 137.246.4.1 ebgp-multihop 2
  neighbor 137.246.4.1 update-source Loopback50
  no auto-summary
!
ip route 137.246.4.1 255.255.255.255 137.246.4.37
ip route 137.246.4.1 255.255.255.255 137.246.4.113
Cloudy# show ip bgp summary
BGP router identifier 191.19.42.1, local AS number 51038
BGP table version is 4, main routing table version 4
3 network entries and 3 paths using 411 bytes of memory
1 BGP path attribute entries using 60 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 3/0 prefixes, 3/0 paths, scan interval 60 secs

Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
137.246.4.1    4   417      7      6        4    0    0 00:02:13      3

Sunny# show ip bgp summary
BGP router identifier 137.246.4.1, local AS number 417
BGP table version is 4, main routing table version 4
3 network entries and 3 paths using 411 bytes of memory
1 BGP path attribute entries using 60 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 3/0 prefixes, 3/0 paths, scan interval 60 secs

Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
191.19.42.1    4 51038      6      7        4    0    0 00:02:43      0
```

就像 Sunny 路由器一样，Cloudy 路由器被配置使用环回接口 **ebgp-multihop** 和一个预定义的 BGP router ID。

第 5 步 配置 Cloudy 路由器，使其和 Calm 路由器成为对等体的关系；对于 I-BGP 的对等体路由，不要使用 IGP。无需使用任何路由过滤，防止 Cloudy 路由器给 Sunny 路由器宣告任何 RFC 1918 的网络。当配置完成后，Cloudy 路由器之后的网络应当能够 ping 通任何上游的路由器。使用 192.19.42.0/23 网络的一半完成这个目的，但是要配置 BGP 使用一条路由来实现整个 23 位地址块的通告。验证 Cloudy 路由器可以使用 10.50.50.1 的源地址来连通 Sunny 路由器的环回接口。

范例 9-80 显示了 Cloudy 路由器的配置。

范例 9-80 Cloudy 路由器对于第 5 步的配置

```

Cloudy# show run | begin interface Ethernet0/0
interface Ethernet0/0
ip address 10.50.50.1 255.255.255.0
ip nat inside
!
interface Serial0/0
ip address 137.246.4.114 255.255.255.252
ip nat outside
!
interface Serial0/1
ip address 137.246.4.38 255.255.255.252
ip nat outside
clockrate 13000000
Cloudy# show run | begin bgp
router bgp 51038
no synchronization
bgp router-id 191.19.42.1
bgp log-neighbor-changes
network 191.19.42.0 mask 255.255.255.0
neighbor 10.50.50.2 remote-as 51038
neighbor 10.50.50.2 next-hop-self
neighbor 137.246.4.1 remote-as 417
neighbor 137.246.4.1 ebgp-multihop 2
neighbor 137.246.4.1 update-source Loopback50
no auto-summary
!
ip nat pool public 191.19.42.3 191.19.42.254 prefix-length 24
ip nat inside source list 8 pool public
ip route 137.246.4.1 255.255.255.255 137.246.4.37
ip route 137.246.4.1 255.255.255.255 137.246.4.113
ip route 191.19.42.0 255.255.255.0 Null0 253
!
access-list 8 permit 10.0.0.0 0.255.255.255

```

no synchronization 命令允许 Cloudy 路由器使用 BGP 的路由，而无需使用 IGP。使用 IGP 的网络地址翻译 (NAT)，Cloudy 路由器可以对 Sunny 路由器隐藏内部的 RFC 1918 网络 10.50.50.0/24。建立一个名为 public 的 NAT 地址池来将 10.0.0.0/8 的剩余网络，在此图中没有显示出来，翻译成共有的网络 191.19.42.0/24。头两个 IP 地址被忽略，这是因为它们已经被使用了。一个指向 Null0 的静态路由，由于管理距离非常高，确保到 191.19.42.0/23 网络的路由存在于主 IP 路由表中，所以此网络可以通过 BGP 通告到 Sunny 路由器。如果你使用 NAT 有任何故障的话，使用 **debug ip nat** 命令可以跟踪 NAT 的翻译，使用 **show ip bgp neighbor ip-address advertised-routes** 命令可以验证 Sunny 路由器正在接收到 191.19.42.0/23 网络的路由。可以使用扩展 ping 来验证连通性。**debug ip nat**、**show ip bgp neighbor 137.246.4.1 advertised-routes** 的输出和扩展 ping 的测试在范例 9-81 中显示。

范例 9-81 验证第 5 步

```

Cloudy# show ip bgp neighbors 137.246.4.1 advertised-routes | begin Network
  Network      Next Hop      Metric LocPrf Weight Path
*> 191.19.42.0/23 0.0.0.0          32768 i
Cloudy# debug ip nat
Cloudy# ping
Protocol [ip]:

```

(待续)


```

Target IP address: 20.0.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.50.50.1
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 20.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
Cloudy#
*Mar 5 06:16:51.307: NAT: s=10.50.50.1->191.19.42.3, d=20.0.0.1 [165]
*Mar 5 06:16:51.327: NAT*: s=20.0.0.1, d=191.19.42.3->10.50.50.1 [165]
*Mar 5 06:16:51.331: NAT: s=10.50.50.1->191.19.42.3, d=20.0.0.1 [166]
*Mar 5 06:16:51.347: NAT*: s=20.0.0.1, d=191.19.42.3->10.50.50.1 [166]
*Mar 5 06:16:51.351: NAT: s=10.50.50.1->191.19.42.3, d=20.0.0.1 [167]
*Mar 5 06:16:51.371: NAT*: s=20.0.0.1, d=191.19.42.3->10.50.50.1 [167]
*Mar 5 06:16:51.371: NAT: s=10.50.50.1->191.19.42.3, d=20.0.0.1 [168]
*Mar 5 06:16:51.391: NAT*: s=20.0.0.1, d=191.19.42.3->10.50.50.1 [168]
*Mar 5 06:16:51.395: NAT: s=10.50.50.1->191.19.42.3, d=20.0.0.1 [169]
*Mar 5 06:16:51.415: NAT*: s=20.0.0.1, d=191.19.42.3->10.50.50.1 [169]

```

第 6 步 在 Stormy 和 Calm 路由器上对 loopback50 地址配置静态路由。在进行到第 7 步之前，验证这些路由器的 Loopback50 接口之间的连通性。范例 9-82 显示了配置在 Stormy 和 Calm 路由器上的静态路由。

范例 9-82 Stormy 和 Calm 路由器上的静态路由

```

stormy# show run | include ip route
ip route 191.19.42.2 255.255.255.255 16.8.4.2
ip route 191.19.42.2 255.255.255.255 16.8.4.6

Calm# show run | include ip route
ip route 16.8.4.9 255.255.255.255 16.8.4.1
ip route 16.8.4.9 255.255.255.255 16.8.4.5

```

Stormy 和 Calm 路由器上环回之间的路由是通过特定的静态路由配置的。

第 7 步 现在在 Stormy 路由器上配置 BGP 路由。将这台路由器分配到 ASN 714，并且使用 Loopback50 的 IP 地址作为 BGP router ID。关掉自动汇总。配置 Stormy 路由器只使用 Loopback50 的接口和 Calm 路由器成为对等体，使用 **network** 语句宣告 3 个环回接口所在的网段。范例 9-83 显示了 Stormy 路由器的 BGP 配置。

范例 9-83 Stormy 路由器的 BGP 配置

```

stormy# show run | begin bgp
router bgp 714
no synchronization
bgp router-id 16.8.4.9

```

(待续)

```

bgp log-neighbor-changes
network 50.0.0.0
network 60.0.0.0
network 70.0.0.0
neighbor 191.19.42.2 remote-as 51038
neighbor 191.19.42.2 ebgp-multihop 2
neighbor 191.19.42.2 update-source Loopback50
no auto-summary
!
ip route 191.19.42.2 255.255.255.255 16.8.4.2
ip route 191.19.42.2 255.255.255.255 16.8.4.6
    
```

类似于 Sunny 路由器，Stormy 路由器是使用 **bgp router-id**、**ebgp-multihop** 和 **update-source** 命令进行配置的。

第 8 步 在 Calm 路由器上配置 BGP 的路由，配置这台路由器与 Stormy 和 Cloudy 路由器成为对等体。记住，Cloudy 路由器不允许使用 IGP 来做 I-BGP 的路由。Calm 和 Stormy 路由器应当互相和对方的 Loopback50 IP 地址成为对等体。配置 Calm 路由器向 Stormy 路由器宣告 191.19.42.0/23 的网络。验证 Cloudy 路由器从 Sunny 和 Stormy 路由器收到完全的表，所有的路由器互相可以 ping 通彼此，在进行到第 9 步之前，这可能需要另外一个 NAT 的翻译。范例 9-84 显示了 Calm 路由器上的 BGP 配置。

范例 9-84 Calm 路由器的 BGP 配置

```

Calm# show run | begin bgp
router bgp 51038
no synchronization
bgp router-id 191.19.42.2
bgp log-neighbor-changes
network 191.19.43.0 mask 255.255.255.0
aggregate-address 191.19.42.0 255.255.254.0 summary-only
neighbor 10.50.50.1 remote-as 51038
neighbor 10.50.50.1 next-hop-self
neighbor 16.8.4.9 remote-as 714
neighbor 16.8.4.9 ebgp-multihop 2
neighbor 16.8.4.9 update-source Loopback50
no auto-summary
!
ip nat pool public 191.19.43.3 191.19.43.254 prefix-length 24
ip nat inside source list 8 pool public
ip route 16.8.4.9 255.255.255.255 16.8.4.5
ip route 16.8.4.9 255.255.255.255 16.8.4.1
ip route 191.19.43.0 255.255.255.0 Null0 253
!
access-list 8 permit 10.0.0.0 0.255.255.255
    
```

Calm 路由器和 Cloudy 路由器使用的是相同的配置。接下来，NAT 被启用，使用一个 NAT 的地址池和一个访问控制列表，然后应用到内口和外口上，一个指向 Null0 的静态路由将这个路由添加到 IGP 的路由表中，所以公共的网络可以宣告到 Stormy 路由器上。接着，**next-hop-self** 命令被添加到 Calm 路由器上，确保 Calm 和 Cloudy 路由器可以宣告一个有效的、可达的下一跳，如范例 9-85 所示。

扩展 ping 和 **show ip nat translations** 命令允许用户验证所有的路由和 NAT 语句已经正确地配置，如范例 9-86 所示。

范例 9-85 Calm 路由器的 BGP RIB

```

Calm# show ip bgp | begin Network
      Network      Next Hop      Metric LocPrf Weight Path
*>i20.0.0.0        10.50.50.1          0    100      0 417 i
*>i30.0.0.0        10.50.50.1          0    100      0 417 i
*>i40.0.0.0        10.50.50.1          0    100      0 417 i
*> 50.0.0.0        16.8.4.9            0           0 714 i
*> 60.0.0.0        16.8.4.9            0           0 714 i
*> 70.0.0.0        16.8.4.9            0           0 714 i
*> 191.19.42.0/23  0.0.0.0              32768 i
* i               10.50.50.1          100      0 i
s> 191.19.43.0/24  0.0.0.0              32768 i

```

范例 9-86 在 Calm 路由器上验证 BGP 和 NAT 的配置

```

Calm# ping
Protocol [ip]:
Target IP address: 20.0.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.50.50.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/30/40 ms
Calm#

```

```

Cloudy# ping
Protocol [ip]:
Target IP address: 50.0.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.50.50.1
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 50.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/12/32 ms
Calm# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 191.19.42.3 20.0.0.1 10.50.50.1 10.50.50.1

```

第 9 步 配置一个路由过滤来防止 Sunny 和 Stormy 路由器使用 AS 51 038 中的任何路由

器作为一个过渡的网络到达彼此的网络。范例 9-87 显示了 Cloudy 和 Calm 路

由器上的过滤配置。

范例 9-87 过滤过渡的路由

```
Cloudy# show run | begin bgp
router bgp 51038
  no synchronization
  bgp router-id 191.19.42.1
  bgp log-neighbor-changes
  network 191.19.42.0 mask 255.255.255.0
  aggregate-address 191.19.42.0 255.255.254.0 summary-only
  neighbor 10.50.50.2 remote-as 51038
  neighbor 10.50.50.2 next-hop-self
  neighbor 137.246.4.1 remote-as 417
  neighbor 137.246.4.1 ebgp-multihop 2
  neighbor 137.246.4.1 update-source Loopback50
  neighbor 137.246.4.1 filter-list 8 out
  no auto-summary
!
ip nat pool public 191.19.42.3 191.19.42.254 prefix-length 24
ip nat inside source list 8 pool public
ip classless
ip route 137.246.4.1 255.255.255.255 137.246.4.37
ip route 137.246.4.1 255.255.255.255 137.246.4.113
ip route 191.19.42.0 255.255.255.0 Null0 253
ip as-path access-list 8 permit ^$
```

```
Calm# show run | begin bgp
router bgp 51038
  no synchronization
  bgp router-id 191.19.42.2
  bgp cluster-id 1253916250
  bgp log-neighbor-changes
  network 191.19.43.0 mask 255.255.255.0
  aggregate-address 191.19.42.0 255.255.254.0 summary-only
  neighbor 10.50.50.1 remote-as 51038
  neighbor 10.50.50.1 next-hop-self
  neighbor 16.8.4.9 remote-as 714
  neighbor 16.8.4.9 ebgp-multihop 2
  neighbor 16.8.4.9 update-source Loopback50
  neighbor 16.8.4.9 filter-list 8 out
  no auto-summary
!
ip nat pool public 191.19.43.3 191.19.43.254 prefix-length 24
ip nat inside source list 8 pool public
ip route 16.8.4.9 255.255.255.255 16.8.4.1
ip route 16.8.4.9 255.255.255.255 16.8.4.5
ip route 191.19.43.0 255.255.255.0 Null0 253
ip as-path access-list 8 permit ^$
!
access-list 8 permit 10.0.0.0 0.255.255.255
```

建立一个 AS 路径访问控制列表 8 只允许本地产生的路由，它含有一个空的自治系统路径(由^\$的常规表达式表示)，被宣告给 Cloudy 和 Calm 路由器的 E-BGP 对等体。这防止 Sunny 和 Stormy 路由器彼此接收对方的网络，防止 AS 51 038 如这里所示成为一个过渡的 AS。范例 9-88 显示了当 AS 路径过滤应用后 Sunny 和 Stormy 路由器的 BGP 表。

先前的实验回顾了本章中的许多内容，包括使用环回实现稳定性的多归路网络，使用 AS 路径访问控制列表来过滤 ASN，以及使用指向空接口的路由来通告一个不在 IGP 路由表中的路由。范例 9-89 显示了对于这个范例的完整的路由器配置。

范例 9-88 查看 Sunny 和 Stormy 路由器上最终的 BGP 表

Sunny# show ip bgp begin Network					
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 20.0.0.0	0.0.0.0	0		32768	i
*> 30.0.0.0	0.0.0.0	0		32768	i
*> 40.0.0.0	0.0.0.0	0		32768	i
*> 191.19.42.0/23	191.19.42.1	0		0 51038	i

Stormy# show ip bgp begin Network					
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 50.0.0.0	0.0.0.0	0		32768	i
*> 60.0.0.0	0.0.0.0	0		32768	i
*> 70.0.0.0	0.0.0.0	0		32768	i
*> 191.19.42.0/23	191.19.42.2			0 51038	i

范例 9-89 对于本实验的完整的路由器配置

```
Sunny# show run | begin Loopback
interface Loopback5
 ip address 20.0.0.1 255.0.0.0
!
interface Loopback10
 ip address 30.0.0.1 255.0.0.0
!
interface Loopback15
 ip address 40.0.0.1 255.0.0.0
!
interface Loopback50
 ip address 137.246.4.1 255.255.255.255
!
interface Serial0
 ip address 137.246.4.113 255.255.255.252
!
interface Serial1
 ip address 137.246.4.37 255.255.255.252
!
router bgp 417
 synchronization
 bgp router-id 137.246.4.1
 bgp log-neighbor-changes
 network 20.0.0.0
 network 30.0.0.0
 network 40.0.0.0
 neighbor 191.19.42.1 remote-as 51038
 neighbor 191.19.42.1 ebgp-multihop 2
 neighbor 191.19.42.1 update-source Loopback50
 no auto-summary
!
ip route 191.19.42.1 255.255.255.255 137.246.4.114
ip route 191.19.42.1 255.255.255.255 137.246.4.38
```

```
Cloudy# show run | begin Loopback
interface Loopback50
 ip address 191.19.42.1 255.255.255.255
!
interface Ethernet0/0
 ip address 10.50.50.1 255.255.255.0
 ip nat inside
!
interface Serial0/0
```

(待续)

```
ip address 137.246.4.114 255.255.255.252
ip nat outside
!
interface Serial0/1
ip address 137.246.4.38 255.255.255.252
ip nat outside
clockrate 1300000
!
router bgp 51038
no synchronization
bgp router-id 191.19.42.1
bgp log-neighbor-changes
network 191.19.42.0 mask 255.255.255.0
aggregate-address 191.19.42.0 255.255.254.0 summary-only
neighbor 10.50.50.2 remote-as 51038
neighbor 10.50.50.2 next-hop-self
neighbor 137.246.4.1 remote-as 417
neighbor 137.246.4.1 ebgp-multihop 2
neighbor 137.246.4.1 update-source Loopback50
neighbor 137.246.4.1 filter-list 8 out
no auto-summary
!
ip nat pool public 191.19.42.3 191.19.42.254 prefix-length 24
ip nat inside source list 8 pool public
ip route 137.246.4.1 255.255.255.255 137.246.4.37
ip route 137.246.4.1 255.255.255.255 137.246.4.113
ip route 191.19.42.0 255.255.255.0 Null0 253
ip as-path access-list 8 permit ^$
!
access-list 8 permit 10.0.0.0 0.255.255.255
```

```
stormy# show run | begin Loopback
interface Loopback5
ip address 50.0.0.1 255.0.0.0
!
interface Loopback10
ip address 60.0.0.1 255.0.0.0
!
interface Loopback15
ip address 70.0.0.1 255.0.0.0
!
interface Loopback50
ip address 16.8.4.9 255.255.255.255
!
interface Serial0
ip address 16.8.4.1 255.255.255.252
clockrate 1300000
!
interface Serial1
ip address 16.8.4.5 255.255.255.252
clockrate 1300000
!
router bgp 714
no synchronization
bgp router-id 16.8.4.9
bgp log-neighbor-changes
network 50.0.0.0
network 60.0.0.0
network 70.0.0.0
neighbor 191.19.42.2 remote-as 51038
neighbor 191.19.42.2 ebgp-multihop 2
neighbor 191.19.42.2 update-source Loopback50
```

(待续)

```

no auto-summary
!
ip route 191.19.42.2 255.255.255.255 16.8.4.2
ip route 191.19.42.2 255.255.255.255 16.8.4.6

Calm# show run | begin Loopback
interface Loopback50
 ip address 191.19.42.2 255.255.255.255
!
interface Ethernet0
 ip address 10.50.50.2 255.255.255.0
 ip nat inside
!
interface Serial0
 ip address 16.8.4.2 255.255.255.252
 ip nat outside
!
interface Serial1
 ip address 16.8.4.6 255.255.255.252
 ip nat outside
!
router bgp 51038
 no synchronization
 bgp router-id 191.19.42.2
 bgp log-neighbor-changes
 network 191.19.43.0 mask 255.255.255.0
 aggregate-address 191.19.42.0 255.255.254.0 summary-only
 neighbor 10.50.50.1 remote-as 51038
 neighbor 10.50.50.1 next-hop-self
 neighbor 16.8.4.9 remote-as 714
 neighbor 16.8.4.9 ebgp-multihop 2
 neighbor 16.8.4.9 update-source Loopback50
 neighbor 16.8.4.9 filter-list 8 out
 no auto-summary
!
ip nat pool public 191.19.43.3 191.19.43.254 prefix-length 24
ip nat inside source list 8 pool public
ip route 16.8.4.9 255.255.255.255 16.8.4.5
ip route 16.8.4.9 255.255.255.255 16.8.4.1
ip route 191.19.43.0 255.255.255.0 Null0 253
ip as-path access-list 8 permit ^$
!
access-list 8 permit 10.0.0.0 0.255.255.255

```

9.9 管理距离和在 BGP 上的效果

当 BGP 和 IGP 一起使用来实现 IP 路由时，它们通常会使用在企业级的网络中，有时，你可能想让路由器更愿意使用 IGP 的路由，而不是 E-BGP 的路由。在正常的情况下，这是不可能的，因为路由器总是会优先使用 E-BGP 的路由，这是因为它们有一个较低的管理距离。思科 IOS 软件使用的管理距离的值如表 9-11 所示。

可以使用一系列的方法来处理这些情况。可以使用 **distance distance-value** 命令来增加 IGP 协议的管理距离(或者对于 E-BGP 的路由使用 **distance bgp external-distance internal-distance local-distance** 命令)；然而，这个命令的效果相当广，可能产生和预想不一样的结果。

一个更灵活的途径是使用 **bgp backdoor** 命令在每一个网络的基础上修改路由。

表 9-11

默认的管理距离

管理距离	协议	管理距离	协议
0	直接连接的网络	115	IS-IS
1	静态路由	120	RIP
20	E-BGP	170	外部的 EIGRP
90	内部的 EIGRP	200	I-BGP
100	IGRP	255	未知
110	OSPF		

后门是什么和如何使用它们

*BGP 后门*主要设计用来改变 E-BGP 的管理距离，以允许 IGP 的路由在 IP 路由表中占优先的位置。**BGP backdoor**命令基本上是将特定的 E-BGP 路由的管理距离从 20 修改为 200，这个管理距离和 I-BGP 路由是一样的，从而允许 IGP 的路由在路由表中更具有优先级。在图 9-18 中，对于这个范例，Pike 路由器有两条路径到达 102.231.6.0/29 的网络——一条是通过 Pine 路由器，另外一条是通过 Union 路由器。

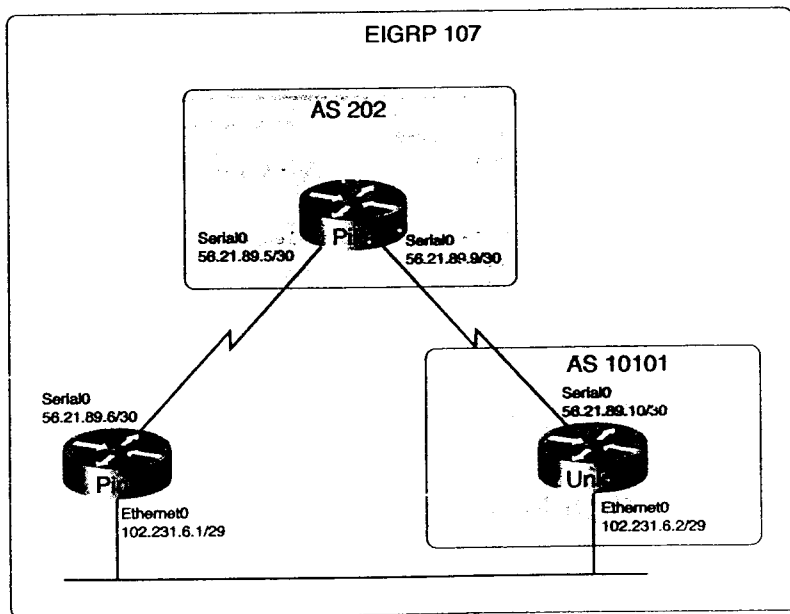


图 9-18 在下游网络上的管理距离和路由

因为 Pike 和 Pine 路由器不是 BGP 的邻居，Pike 路由器只存储了一条到达 102.231.6.0/29 网络的路由。Pike 路由器忽略 EIGRP 路由的原因是路由的管理距离是 90，它比 Union 路由器的 E-BGP 路由的管理距离 20 大，如范例 9-90 所示。

为了允许 Pike 路由器使用两条 EIGRP 路由到达 102.231.6.0/29 的网络，可以对那个网络配置 BGP 的后门。BGP 的后门是通过使用 **network network-prefix mask network-mask backdoor** 命令配置的。你可能会认为 BGP **network** 命令不能用于 BGP 去通告一个非直连的

范例 9-90 在后门配置之前的 Pike 路由器的路由表

```
Pike# show ip route | begin subnet
102.0.0.0/29 is subnetted, 1 subnets
B    102.231.6.0 [20/0] via 56.21.89.10, 00:05:49
56.0.0.0/30 is subnetted, 2 subnets
C    56.21.89.4 is directly connected, Serial0
C    56.21.89.8 is directly connected, Serial1
```

路由，这点是正确的；然而，在本例的情况下，**network** 命令用于本地去改变后门路由的管理距离。BGP 并不将这条路由通告为本地路由，只是将这条路由的管理距离简单地修改了，这样使得 EIGRP 的路由在路由表中优先出现。范例 9-91 显示了 BGP **backdoor** 命令是如何对于 102.231.6.0/29 的网络改变 IP 路由的优先级的。

范例 9-91 使用 BGP 后门来改变管理距离

```
Pike# show run | begin eigrp
router eigrp 107
 network 56.21.89.4 0.0.0.3
 network 56.21.89.8 0.0.0.3
 maximum-paths 2
 no auto-summary
 no eigrp log-neighbor-changes
!
router bgp 202
 no synchronization
 bgp log-neighbor-changes
 network 56.21.89.8 mask 255.255.255.252
 network 102.231.6.0 mask 255.255.255.248 backdoor
 neighbor 56.21.89.10 remote-as 10101
 no auto-summary
```

范例 9-92 显示了 IP 路由表最终变化的结果。当应用这个配置后，BGP 路由的管理距离被改变了，E-BGP 的路由会从主 IP 路由表中清除。这时，两条 EIGRP 的路由会添加到路由表中，这是因为它们有较低的管理距离。而且，注意到 **show ip bgp 102.231.6.0/29** 命令还会显示出这条路由是最佳的路由，BGP 的网段仍然不通告给任何对等体。

范例 9-92 在 BGP 后门配置后 Pike 路由器的配置

```
Pike# show ip route | begin subnet
102.0.0.0/29 is subnetted, 1 subnets
D    102.231.6.0 [90/2195456] via 56.21.89.10, 00:01:14, Serial1
      [90/2195456] via 56.21.89.6, 00:01:14, Serial0
56.0.0.0/30 is subnetted, 2 subnets
C    56.21.89.4 is directly connected, Serial0
C    56.21.89.8 is directly connected, Serial1
Pike# show ip bgp 102.231.6.0/29
BGP routing table entry for 102.231.6.0/29, version 6
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x800
Not advertised to any peer
10101
56.21.89.10 from 56.21.89.10 (10.2.2.1)
Origin IGP, metric 0, localpref 100, valid, external, best
```

现在你已经了解到有许多方法可以配置 BGP 来实现路由和策略的增强,下面来看看 BGP 是如何控制因特网路由表的稳定性的,这是通过路由衰减的方法和其他的一些方法,可以调整 BGP 来执行得更有效率。

9.10 BGP 路由衰减

*BGP 路由衰减*可以控制 E-BGP 对等体之间路由波动的效果。路由衰减通常可以帮助服务提供商防止一个用户的路由或者电路的问题影响服务提供商网络的稳定性,通过回退问题的 BGP 路由来实现。有两种方法可以启用路由衰减:第一种是使用 **bgp dampening** 命令对所有的 BGP 对等体全局启用路由衰减;第二种方法是使用路由映射来指定某些路由应当配置衰减,并将参数应用到被衰减的网络上。下面的语法显示了 **bgp dampening** 命令和它的可选参数。

```
bgp dampening [[route-map route-map-name] [[ half-life] | reuse-limit start-suppress suppress-duration]]
```

使用 **bgp dampening** 命令,路由衰减可以配置 3 种方式:

- 全局路由衰减,使用默认的参数;
- 全局路由衰减,使用定制的参数;
- 特定的路由衰减,使用定制的参数。

表 9-12 显示了可选的 **bgp dampening** 命令参数和它们的描述。

表 9-12 BGP 路由衰减参数

衰减命令	描述
<i>half-life</i>	等待的时间来减少衰减值,范围从 1~45 min,默认的半衰期是 15 min
<i>reuse-limit</i>	1~20 000 之间的一个值,它和衰减值作比较来决定路由的可用性,如果衰减值大于抑制,路由将被抑制住,如果没有,它将会被再次使用,默认的抑制范围是 750
<i>start-suppress</i>	1~20 000 之间的一个值,指定如果路由被抑制了,那么就使用衰减,默认的路由衰减是对于每个路由波动 2000 个衰减值
<i>suppress-duration</i>	这个值指定了路由将被抑制的最大时间,范围从 1~255 min。默认的抑制时间是半衰期的 4 倍,即换句话说就是 60 min
<i>route-map route-map-name</i>	指定一个路由映射将来指定路由衰减的参数。路由映射用于指定路由应当应用的衰减策略,当路由映射使用时,可以应用相同的路由衰减参数

当路由波动被激活后,会对波动的路由实施一个 1000 点的衰减值。路由器会对每一个波动的路由维护一个历史的记录,这个历史记录会维护每一条路由的衰减信息。*half-life* 值用于在一条路由波动后通过时间减少衰减值,因此,如果路由停止了波动,它就不会再被衰减,历史记录最终会被清除。如果路由再次波动,就会出现下一次衰减,当达到了 *suppress-limit* 时,路由就会被衰减。当路由已经被衰减时,它就不会再宣告给其他的 BGP 对等体,直到 *suppress-duration* 的时间过后。

注意: BGP 路由的初始衰减值被设置为 1000 点,并且不能被修改;然而,其他的参数

都是用户可以配置的，你可以接受默认的值，或者根据特定的网络需求，建立你自己的定制的衰减策略。

看一下图 9-19 所示的网络图。在这个图中，AS 18 901 中的 Service_Provider 路由器被配置为使用路由衰减的策略，衰减路由使用的是默认的衰减参数，除了 half-life 这个值。在这种情况下，half-life 被修改为 5min，如范例 9-93 所示。

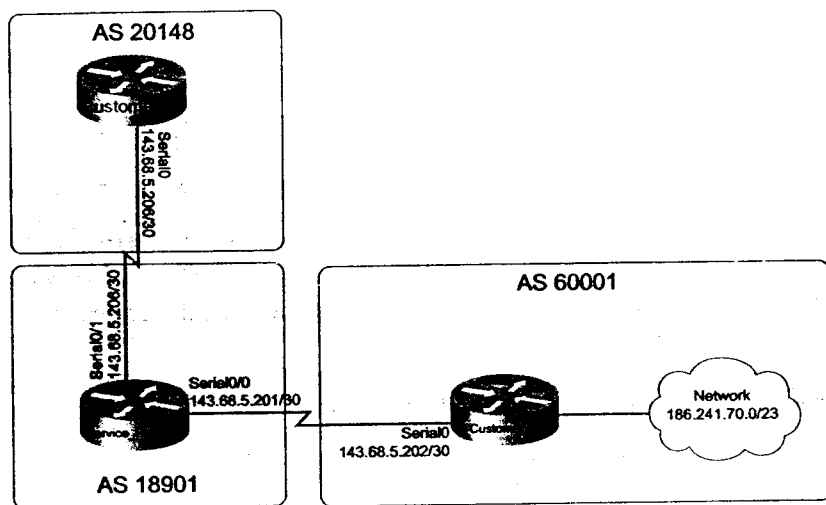


图 9-19 服务提供商到用户的网络

范例 9-93 Service_Provider 路由器的 BGP 配置

```
Service_Provider# show run | begin bgp
router bgp 18901
no synchronization
bgp log-neighbor-changes
bgp bestpath dampening 5
bgp dampening 5
network 143.68.5.200 mask 255.255.255.252
network 143.68.5.204 mask 255.255.255.252
neighbor 143.68.5.202 remote-as 60001
neighbor 143.68.5.206 remote-as 20148
no auto-summary
```

在默认的情况下，在后来的思科 IOS 软件 12.2 版本中 **bgp dampening** 命令发出后，**bgp best path dampening** 命令会自动地输入。这个命令也用于启用和关闭路由衰减。有几种方法可以验证和跟踪 BGP 的路由衰减的配置，其中最详细的就是 **show ip bgp dampened parameters** 命令。范例 9-94 使用了 **show ip bgp dampening parameters** 命令来显示对 Service_Provider 路由器的 BGP 路由衰减参数。

这个命令显示了对于本地 BGP 路由衰减策略的所有参数，并且在这种情况下，显示 Service_Provider 路由器配置了 5 min 的 half-life。half-life 参数的修改也改变了最大的抑制时间，所以被抑制的路由不再被严厉地衰减。范例 9-95 显示了默认的 BGP 路由衰减参数。

范例 9-94 show ip bgp dampening parameters 命令

```
Service_Provider# show ip bgp dampening parameters
dampening 5 750 2000 20
Half-life time      : 5 mins      Decay Time          : 775 secs
Max suppress penalty: 12000       Max suppress time: 20 mins
Suppress penalty    : 2000        Reuse penalty       : 750
```

范例 9-95 默认的 BGP 路由衰减参数

```
Service_Provider# show ip bgp dampening parameters
dampening 15 750 2000 60 (DEFAULT)
Half-life time      : 15 mins     Decay Time          : 2320 secs
Max suppress penalty: 12000       Max suppress time: 60 mins
Suppress penalty    : 2000        Reuse penalty       : 750
```

show ip bgp dampening flap-statistics 命令显示了关于所有被衰减的路由的详细信息。在这种情况下，Service_Provider 路由器衰减了 186.241.70.0/23 的网络，这是因为它已经波动了 4 次。范例 9-96 使用了 **show ip bgp dampening flap-statistics** 命令来显示路由已经被衰减了 3 分 34 秒，并且在 10 分 20 秒后会被重新使用。

范例 9-96 show ip bgp dampening flap-statistics 命令

```
Service_Provider# show ip bgp dampening flap-statistics
BGP RIB version is 13, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          From          Flaps Duration Reuse      Path
*d 186.241.70.0/23  143.68.5.202    4      00:03:34 00:10:20 60001
```

clear ip bgp dampening 命令允许用户清除被衰减的路由和与路由相关的波动的统计数字。

可以使用本章前面提到的技术，采取一些步骤来防止 BGP 路由由波动的发生；例如，Customer_B 路由器应当配置使用下面的一些 BGP 特性：

- 多条链路和环回接口来防止网络故障；
- 将路由聚合成更小、更稳定的前缀，使得一个网络故障不会影响宣告给上游路由器的路由；
- 建立一个指向 Null0 的静态路由来防止不稳定的 IGP 路由。

服务提供商总是很感兴趣，在他们的网络中使用 BGP 路由衰减来维持网络的稳定性。他们的策略可能无意中会影响一个设计得很差的网络，所以，你总是应当尽可能将你的网络配置成最稳定的、最冗余的 BGP 配置。

9.11 调整 BGP 的性能

对 BGP 的会话进行配置和故障排查可能会很消耗时间。每次当你改变一个 BGP 的参数时，你都必须清除会话来使变化生效。使用 **clear ip bgp ip-address *** 命令可以清除 BGP 的会话，这个命令是很耗时的，并且导致网络重新复位。在过去，**neighbor {ip-address | peer-group}**

soft-reconfiguration inbound 和 **clear ip bgp * ip-address in** 命令可以允许对接收的 BGP 路由进行“软”配置来解决这个问题。这意味着 BGP 的对等体必须要在内存中存储接收的 BGP 路由表，那么在路由器上的 BGP 配置增加了路由器的负荷。

随着路由刷新能力的引入，它在 RFC 2918 中描述，并且在思科 IOS 软件版本 12.2 (6) T 以后介绍进来，动态的接收和发送的软复位现在被允许了。为了了解一个对等体是否支持软件刷新的能力，使用 **show ip bgp neighbors ip-address | begin capabilities** 命令，如范例 9-97 所示。

范例 9-97 show ip bgp neighbors | begin capabilities 命令

```
Service_Provider# show ip bgp neighbors 143.68.5.202 | begin capabilities
Neighbor capabilities:
  Route refresh: advertised and received(old & new)
  Address family IPv4 Unicast: advertised and received
Received 341 messages, 2 notifications, 0 in queue
Sent 312 messages, 0 notifications, 0 in queue
Default minimum time between advertisement runs is 30 seconds
For address family: IPv4 Unicast
BGP table version 251, neighbor version 251
Index 1, Offset 0, Mask 0x2
Route refresh request: received 7, sent 1
1 accepted prefixes consume 40 bytes
Prefix advertised 462, suppressed 0, withdrawn 2
```

注意，先前的范例显示了 143.68.5.202 的邻居支持路由刷新的能力，并且已经使用它刷新了路由 7 次。当你确信路由的刷新能力可支持后，可以开始使用新的 **clear ip bgp * soft[in | out]** 命令，如范例 9-98 所示。

范例 9-98 在路由刷新的请求过程中，跟踪 IP BGP

```
Service_Provider# clear ip bgp * soft
*Mar 1 09:18:01.817: BGP: service reset requests
*Mar 1 09:18:01.821: BGP: 143.68.5.202 sending REFRESH_REQ(5) for afi/safi: 1/1
*Mar 1 09:18:01.821: BGP: 143.68.5.202 send message type 5, length (incl.
header) 23
```

当一条路由刷新信息被发送时，如果在一个 BGP 会话中两个对等体都支持远端刷新的能力，那么远端的对等体会重新发送外出的 BGP 更新，而无需清除整个 BGP 的会话。如果远端的对等体不支持路由刷新的能力，对等体会忽略这个请求，你要么需要那个邻居使用 **soft-reconfiguration** 命令，要么需要标准的 **clear ip bgp {* | ip-address | peer-group}** 命令来复位 BGP 的会话。远端的对等体还会接收到路由刷新的请求，但是不能够使用它；然而，因为路由器不能够理解这个请求，它将会忽略包含在路由刷新请求中的消息和随后的路由刷新能力的通告，如范例 9-99 所示。

范例 9-99 跟踪一个忽略的路由刷新

```
Older_Router# debug ip bgp
BGP debugging is on
00:20:58: BGP: 10.1.1.1 unrecognized OPEN parameter (0x2/0x6)
00:20:58: BGP: 10.1.1.1 unrecognized OPEN parameter (0x2/0x2)
```

(待续)

```

Older_Router# show ip bgp neighbors
BGP neighbor is 10.1.1.1, remote AS 8, internal link
Index 2, Offset 0, Mask 0x4
Inbound soft reconfiguration allowed
BGP version 4, remote router ID 10.1.1.1
BGP state = Established, table version = 1, up for 00:00:53
Last read 00:00:52, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 5 seconds
Received 10 messages, 0 notifications, 0 in queue
Sent 8 messages, 0 notifications, 0 in queue
Prefix advertised 0, suppressed 0, withdrawn 0
Connections established 2; dropped 1
Last reset 00:01:00, due to Soft reconfig change
0 accepted prefixes consume 0 bytes
0 denied but saved prefixes consume 0 bytes
0 history paths consume 0 bytes

```

通过 BGP 的配置节省内存

BGP 是一个对内存和处理器都非常消耗的协议。在你的职业生涯中的某个时候，你很有可能陷入这样的情况，就是你必须在一个没有足够资源来支持现有的 BGP 系统需求的路由器上运行 BGP 协议。有一系列的选择可以帮助你解决这个问题：升级内存，升级路由器，过滤进入的路由，或者限制 BGP 可以接收的路由前缀的数量。假设你不能够立刻升级路由器、内存或者处理器，最好的选择就是路由过滤或者限制进入的 BGP 前缀的数量。范例 9-100 显示了在一个真正的因特网路由器上获取的 `show ip bgp summary` 命令的输出（IP 地址已经被改变了）。

范例 9-100 因特网路由表统计数字

```

BGP router identifier 6.6.6.6, local AS number 123
BGP table version is 8438778, main routing table version 8438778
114591 network entries and 337412 paths using 23262159 bytes of memory
82050 BGP path attribute entries using 4923540 bytes of memory
15 BGP rrinfo entries using 360 bytes of memory
40359 BGP AS-PATH entries using 1046148 bytes of memory
162 BGP community entries using 7100 bytes of memory
54353 BGP route-map cache entries using 869648 bytes of memory
21745 BGP filter-list cache entries using 260940 bytes of memory
Dampening enabled. 79 history paths, 20 dampened paths
BGP activity 227228/2798971 prefixes, 8600655/8263243 paths, scan interval 15 secs

```

一、通过使用部分 BGP 路由表来最小化内存的使用

限制 BGP RIB 尺寸的最好的方法之一就是使用路由过滤来只接受部分的 BGP RIB 路由更新。有两种方法运行 BGP，产生部分表：请求服务提供商来过滤发送给你的网络的路由，只发送给你部分路由；或者你可以过滤你的接收路由。配置部分 BGP RIB 的最简单和最安全的方法就是使用 AS 路径访问控制列表，通过一个过滤列表来匹配开始和终止于你的服务提供商 AS 的 AS 路径。

例如，使用图 9-19 所示的网络，Customer_B 路由器的内存不够用了，不再能够处理 Service_Provider 路由器发送的完整的因特网路由表。为了解决这个问题，可以使用 AS 路径访问控制列表来限制从上游路由器的 E-BGP 邻居收到的 AS 路径的数量，如范例 9-101 所示。

并且上游的服务提供商可以给你发送一个默认的路由，使得你的路由器还可以到达其他的因特网网络。

范例 9-101 对部分的 BGP RIB 实施过滤

```
Customer_B# show run | begin bgp
router bgp 60001
  no synchronization
  bgp log-neighbor-changes
  network 186.241.70.0 mask 255.255.254.0
  neighbor 143.68.5.201 remote-as 18901
  neighbor 143.68.5.201 filter-list 101 in
  no auto-summary
!
ip as-path access-list 101 permit ^18901$
```

在这个范例中，AS 路径访问控制列表 101 用于过滤没有起始和终止于 AS 18 901 的任何路由，它可以将接收的路由的数量限制到 63 条，如范例 9-102 所示。

范例 9-102 当实施接收的路由过滤后，Customer_2 路由器的 BGP RIB

```
Customer_B# show ip bgp summary | begin Neighbor
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
143.68.5.201   4    18901    116    123    248    0    0 01:33:35      63
```

有一些不同的方法可以处理内存的问题（按照对内存的最低的利用率排列顺序）

- 只从服务提供商接收默认路由；
- 只接收默认路由和来自每一个服务提供商本地产生的路由；
- 只接收默认路由和来自每一个服务提供商和客户产生的路由。

这些实施的选择取决于你。一定要记住，如果你不想接受完整的路由表，为了到达因特网的网络，你必须接受一个默认的路由。

二、配置接收的 BGP 前缀限制

限制接收的 BGP 路由数量的另外一种方法就是使用 **maximum-prefix** 命令。当使用 **maximum-prefix** 命令时，当最大的前缀数量达到后，你有两种选择：自动关闭 BGP 的会话，或者发送一个警告的消息。如果你绝对不允许路由器超过一个特定的路由数量，那么可以使用 **maximum-prefixes** 命令来关闭 BGP 的会话，不冒犯 BGP 对等体，使用 **neighbor {ip-address | peer-group} maximum-prefix limitation-number** 命令，使用一个 1~4 294 967 295 范围内的数字。范例 9-103 显示了当 **maximum-prefix** 命令用在 Customer_B 路由器上时所发生的情况。

范例 9-103 使用 maximum-prefix 命令来关闭 BGP 的会话

```
Customer_B# show run | begin bgp
router bgp 60001
  no synchronization
  bgp log-neighbor-changes
  network 186.241.70.0 mask 255.255.254.0
  neighbor 143.68.5.201 remote-as 18901
```

（待续）

```

neighbor 143.68.5.201 maximum-prefix 50
neighbor 143.68.5.201 filter-list 101 in
no auto-summary
!
ip as-path access-list 101 permit ^18901$
Customer_B# show ip bgp summary | begin Neighbor
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
143.68.5.201   4  18901   138    147      0    0    0 00:02:20 Idle (PfxCt)
Customer_2# show logging | include %BGP
*Mar 1 02:48:01.731: %BGP-5-ADJCHANGE: neighbor 143.68.5.197 Down Neighbor
deleted
*Mar 1 02:48:53.927: %BGP-3-MAXPFXEXCEED: No. of prefix received from
143.68.5.201 (afi 0): 63 exceed limit 50
*Mar 1 03:08:05.507: %BGP-3-MAXPFXEXCEED: No. of prefix received from
143.68.5.201 (afi 0): 63 exceed limit 50
*Mar 1 03:33:04.307: %BGP-3-MAXPFXEXCEED: No. of prefix received from
143.68.5.201 (afi 0): 63 exceed limit 50
*Mar 1 03:33:04.307: %BGP-5-ADJCHANGE: neighbor 143.68.5.201 Down BGP
Notification sent
*Mar 1 03:33:04.307: %BGP-3-NOTIFICATION: sent to neighbor 143.68.5.201 3/1
(update malformed) 0 bytes

```

在先前的范例中，如果对等体 143.68.5.201 发送的路由超过了 50 个路由前缀，BGP 会话将会被关闭，并且会记录一条 %BGP-3-MAXPFXEXCEED 的消息。在这种情况下，BGP 会话不会被重新初始化，直到这个会话被手动复位，而且接收的最大数量的路由没有超过限制。当条件重新满足，BGP 的连接重新启用后，这个连接会重新起来。另外一个温和一点的方法就是使用带有可选的 **warning-only** 参数的 **maximum-prefix** 命令，当最大的前缀数量达到后，这个命令只会发出一个警告消息。当这个命令和系统日志报告的能力结合起来以后，你可以监控 BGP 前缀的数量，并且在收到系统日志的消息后采取措施。范例 9-104 显示了在 50 条最大前缀的 80% 的数量限制达到后，**maximum-prefix warning-only** 命令是如何发送一个警告消息给 186.241.70.89 的系统日志服务器的。

范例 9-104 使用 maximum-prefix 警告来发送警告消息

```

router bgp 60001
no synchronization
bgp log-neighbor-changes
network 186.241.70.0 mask 255.255.254.0
neighbor 143.68.5.201 remote-as 18901
neighbor 143.68.5.201 maximum-prefix 50 80 warning-only
neighbor 143.68.5.201 filter-list 101 in
maximum-paths 2
no auto-summary
!
ip as-path access-list 101 permit ^18901$
!
logging 186.241.70.89
Customer_2# show logging | include %BGP
*Mar 1 04:04:40.462: %BGP-4-MAXPFX: No. of prefix received from 143.68.5.201
(afi 0) reaches 41, max 50
*Mar 1 04:04:40.470: %BGP-3-MAXPFXEXCEED: No. of prefix received from
143.68.5.201 (afi 0): 51 exceed limit 50

```


9.12 实验 15: 多归路一个 BGP 网络

先前的一些章节很少介绍 BGP 的理论以及基本的和高级的 BGP 配置，只是简单地推荐了在生产型网络中优化 BGP 的因特网路由的一些方法。下面的实验集中于一个多归路 BGP 的配置，使用一个真实的 BGP 场景来测试高级的 BGP 配置内容，HTTP web 流量测试了最终网络的可达性。

9.12.1 实验练习

在这个实验的场景中，配置一个仿真的因特网 web 浏览服务，使用一个 24 小时网络骨干和两个上游的服务提供商网络。这个 24 小时网络有两个因特网边界路由器和 3 个属于两个上游服务提供商网络的路由器成为对等体。这个实验要求你仿真一个因特网连接，使用通常的负载均衡技术来最大化网络资源的利用，实施通常的安全经验来减轻简单的安全隐患，并且从 24 小时网络的一个内部 PC 使用 HTTP web 的浏览来测试网络的连通性。

9.12.2 实验目的

这个实验演示了在前面 3 章中介绍过的许多内容，以及如何在冗余的网络设计中使用它们：

- BGP 多归路；
- 在两个自治系统之间负载分担；
- 路由聚合；
- BGP MD-5 验证；
- 对于 I-BGP，使用路由反射器；
- I-BGP 网络出口的优先级；
- 重分发静态路由；
- 使用对等体组来简化配置；
- 使用 AS 路径和团体值来过滤路由；
- 在 BGP 中使用 DHCP 和 NAT 来隐藏内部的 RFC 1918 的网络地址。

9.12.3 需要的设备

- 一台思科的路由器，具有 5 个串口，充当帧中继交换机；
- 6 台思科路由器，具有至少一个串行接口和一个以太网接口；
- 一台思科路由器，具有两个串行接口（其中一台路由器需要一个以太网接口）；

- 一个交换机，连接 5 个在单独 VLAN 中的多点访问路由器；
- 一个有以太 NIC 网卡的 PC，能够运行 TCP/IP，可以运行 DHCP 和 web 浏览器；
- 这个实验的一部分最适合于思科 IOS 软件版本为 12.2 (11) T 或更高版本。

9.12.4 物理布局和预规划

对于这个实验，可以使用图 9-20 所示的网络设计。在 AS 104 和 AS 60 中的路由器可以仿真因特网服务提供商网络，即因特网服务提供商 1 和因特网服务提供商 2。Drazen 和 Palmer 路由器是一个 24 小时的网络边界路由器，所有其他的路由器都是内部的 24 小时的网络路由器。

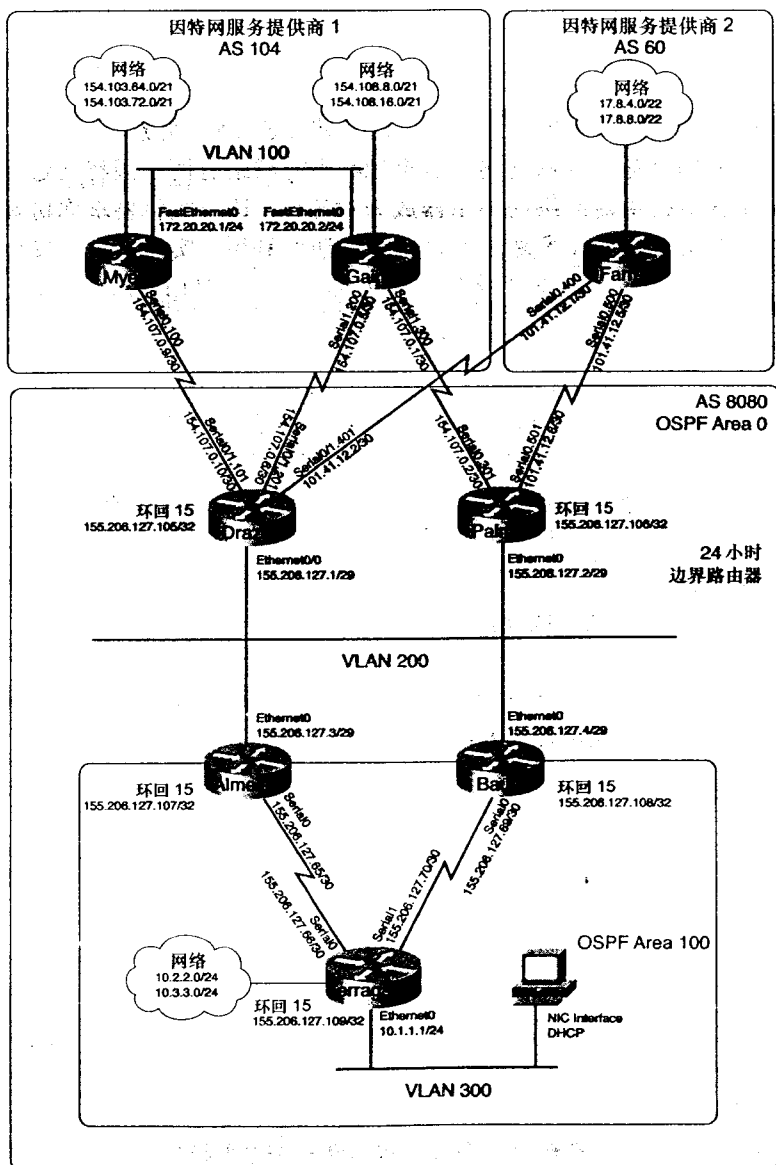


图 9-20 24 小时网络

- 按照图 9-20 所示，将路由器进行布线。Myers、Gaines、Farrell、Drazen 和 Palmer 路由器应当按照表 9-13 所示的接口号码连接到帧中继交换机上，并且使用背对背的串行线缆连接。
- 按照表 9-13 所示，使用接口和 DLCI 号码来配置帧中继交换机。

表 9-13 帧中继交换机的参数

帧中继交换机接口	路由器接口	帧中继交换机接口	路由器 DLCI	路由器接口	路由器 DLCI
Serial1	Myers Serial0.100	Serial0	100	Drazen Serial0/1.101	101
Serial2	Gaines Serial1.200	Serial0	200	Drazen0/1.201	201
Serial2	Gaines Serial1.300	Serial3	300	Palmer Serial0.301	301
Serial4	Farrell Serial0.400	Serial0	400	Drazen Serial0/1.401	401
Serial4	Farrell Serial0.500	Serial3	500	Palmer Serial0.501	501

范例 9-105 显示了来自帧中继交换机的 `show frame relay route` 命令的输出。

范例 9-105 帧中继交换机的配置

Frame-Relay-Switch # show frame-relay route				
Input Intf	Input DlcI	Output Intf	Output DlcI	Status
Serial0	101	Serial1	100	active
Serial0	201	Serial2	200	active
Serial0	401	Serial4	400	active
Serial1	100	Serial0	101	active
Serial2	200	Serial0	201	active
Serial2	300	Serial3	301	active
Serial3	301	Serial2	300	active
Serial3	501	Serial4	500	active
Serial4	400	Serial0	401	active
Serial4	500	Serial3	501	active

- 将 Myers、Gaines、Drazen、Palmer、Almeida 和 Bauer 路由器的以太接口连接到以太网交换机上，正如先前的图 9-20 所示。
- 按照图 9-20 所示，将 Almeida 和 Bauer 路由器与 Ferragamo 路由器相连。
- 验证每一台路由器的每一个接口都是 up/up 的状态。
- 不要在 Ferragamo 路由器或者 PC 上配置 DHCP。

9.12.5 实验练习

第 1 步 按照表 9-14 所示，配置所有的 IP 地址，并且按照这个表将所有的以太接口分配到相应的 VLAN 中。

第 2 步 对 Drazen、Palmer、Almeida、Bauer 和 Ferragamo 路由器配置 OSPF 的路由，只将 Drazen、Palmer、Almeida 和 Bauer 路由器的以太接口放入到 OSPF area 0 里。

表 9-14

对于这个网络模型的 IP 地址

路由器的名字	路由器的接口	IP 地址	以太 VLAN
Myers	FastEthernet0	172.20.20.1/24	100
	Loopback100	154.103.64.1/21	
	Loopback200	154.103.72.1/21	
	Serial0.100	154.107.0.9/30	
Gaines	FastEthernet0	172.20.20.2/24	100
	Loopback100	154.108.8.1/21	
	Loopback200	154.108.16.0/21	
	Serial1.200	154.107.0.5/30	
	Serial1.300	154.107.0.1/30	
Farrell	Loopback100	17.8.4.1/22	50
	Loopback200	17.8.8.0/22	
	Serial0.400	101.41.12.1/30	
	Serial0.500	101.41.12.5/30	
Drazen	Ethernet0/0	155.206.127.1/29	200
	Loopback15	155.206.127.105/32	
	Serial0/1.101	154.107.0.10/30	
	Serial0/1.201	154.107.0.6/30	
	Serial0/1.401	101.41.12.2/30	
Palmer	Ethernet0	155.206.127.2/29	200
	Loopback15	155.206.127.106/32	
	Serial0.301	154.107.0.2/30	
	Serial0.501	101.41.12.6/30	
Almeida	Ethernet0	155.206.127.3/29	200
	Loopback15	155.206.127.107/32	
	Serial0	155.206.127.65/30	
Bauer	Ethernet0	155.206.127.4/29	200
	Loopback15	155.206.127.108/32	
	Serial0	155.206.127.69/30	
Ferragamo	Ethernet0	10.1.1.1/24	300
	Loopback15	155.206.127.109/32	
	Loopback100	10.2.2.1/24	
	Loopback200	10.3.3.1/24	
	Serial0	155.206.127.66/30	
	Serial1	155.206.127.70/30	
PC	Ethernet NIC	DHCP	300

- 将 Drazen 和 Palmer 路由器的环回接口也放入到 area 0 中，Ferragamo 路由器以及 Almeida 和 Bauer 路由器上的串口应当放入到 area 1 中。
- 对于每一个 OSPF 的路由器，使 Loopback15 的接口 IP 地址作为每一台路由器的 OSPF router ID。

- 使得 Almeida 和 Bauer 路由器给所有下游的邻居发送默认路由。
- 第 3 步** 在 Ferragamo 路由器上配置负载均衡，使得 OSPF 可以同时使用两个上游的串口给 155.206.127.0/29 网络转发数据包。使用适当的命令来启用负载均衡，使得属于同一股流的数据包走相同的路径。
- 第 4 步** 将 Ferragamo 路由器配置成为 10.1.1.0/24 网络的 DHCP 服务器。这台路由器也应当给它们的 DHCP 客户分配 fiction.org 的域名。当在路由器上配置完 DHCP 的服务后，配置 PC 使得它们可以从路由器请求一个 DHCP 的地址，并且通过 ping Drazen 路由器的环回接口来验证这个配置。
- 第 5 步** 当配置完内部网络后，增加主机，并且启用路由，你现在可以集中于这个实验的 BGP 部分了。开始配置位于 AS 104 中的外部的服务提供商路由器，即 Myers 和 Gaines 路由器。在 Myers 和 Gaines 路由器上启用 BGP 路由。当你完成这个任务后，每一台路由器应当能够看到/21 的网络，这是路由器之间内部通告的网络。
- 第 6 步** 接下来，在位于 AS 104 中的服务提供商 1 的路由器和位于 AS 8080 中的 24 小时路由器之间配置 E-BGP 的路由。使用对等体组来简化 BGP 的配置。
 - 使得 AS 8080 的边界路由器使用它们的 Loopback15 的 IP 地址作为 BGP 路由器 ID，并且使用它们的环回地址作为对等体的地址。在这个实例中，在 AS 104 的路由器上，允许每一台路由器对每一个邻居配置一条静态路由。
 - 不要允许服务提供商 1 的路由器通告 172.20.20.0/24 的网络给任何外部的对等体。不能使用一个分布式列表来完成这个任务。
 - 不要允许服务提供商路由器使用 AS 8080 的边界路由器作为一个过渡网络到达彼此的/21 网络。
 - 当这个步骤完成后，在 AS 8080 中的路由器应当可以看到位于 AS 104 路由器后面的所有/21 网络。
- 第 7 步** 为了完成 E-BGP 因特网对等体的会话，需要在位于 AS 60 中的 Farrell 路由器和 24 小时边界路由器之间配置一个 BGP 会话。这些 BGP 会话应当使用在第 6 步中指定的所有规则。
 - 使用对等体组来允许进一步的对等体添加。
 - 使得 AS 8080 的边界路由器使用它们的 Loopback15 IP 地址作为 BGP router ID，在 Farrell 路由器上允许对每一个邻居配置一条静态路由。
 - 不要允许服务提供商路由器使用 AS 8080 的边界路由器作为一个过渡网络来到达彼此的网络。
 - 当这一步完成后，位于 AS 8080 中的路由器应当可以看到由服务提供商路由器通告的所有外部网络。
- 第 8 步** 如果在 24 小时路由器和它们的对等体路由器（即 Almeida 和 Bauer 路由器）之间没有配置 I-BGP 的连接，那么 BGP 对等体的配置就不是完整的。
 - 在这些路由器之间配置 I-BGP 对等关系，使用 Loopback15 的接口地址作为对等体的地址。
 - 使用对等体组来简化边界路由器的配置，在这个网络中不要全冗余这些路

由器。

- 在 AS 8080 边界路由器上汇总所有的 155.206.127.0 的网络，不要宣告任何小于/24 的路由。
- 通过从 Ferragamo 路由器上 ping 这些因特网网络来验证配置。

第 9 步 为了最有效地使用边界路由器和服务提供商路由器之间的连接，配置服务提供商 1 的路由器使用来自 Drazen 路由器的路由，配置服务提供商 2 的路由器使用来自 Palmer 路由器的路由，多出口鉴别器或者 AS 路径属性都不可以用于完成这个任务。本地产生的路由应当总是具有最高的优先级：

- Drazen 路由器应当优先使用来自 Myers 路由器的路由，其次选用来自 Farrell 路由器的路由；而 Palmer 路由器应当优先使用来自 Farrell 路由器的路由，其次选用 Gaines 路由器的路由，最后是 Myers 路由器的路由。本地产生的路由应当总是具有最高的优先级。

第 10 步 作为一个安全上的注意事项，应当在 24 小时边界路由器上关掉任何 CDP、HTTP web 访问和任何不必要的特性。

- 也建立一个反欺骗的访问控制列表，它可以防止任何 RFC 1918 的私有的 IP 地址和内部地址。
- 确保 OSPF 路由不允许通告到 24 小时网络之外。
- 在 Internet-facing 路由器上启用 HTTP web 服务，它们将用于仿真因特网 web 服务器。
- 配置 HTTP 的服务使用 Loopback100 接口的 IP 地址。

第 11 步 为了从因特网上隐藏 RFC 1918 的私有地址，配置 24 小时边界路由器将所有的内网地址翻译成因特网可路由的公网地址，如表 9-15 所示。

- 确保所有的 IP 地址块可以聚合成最好的地址。所有的具体路由都应当被抑制住；只有聚合的路由可以通告给外部的邻居。
- 为了验证用户到因特网的连接性，使用一个 web 浏览器来进入每一个服务提供商网络的 HTTP web 配置站点。

表 9-15 内部到外部 NAT 地址

内部网络	外部网络	内部网络	外部网络
10.1.1.0/24	155.206.124.0/24	10.3.3.0/24	155.206.126.0/24
10.2.2.0/24	155.206.125.0/24		

9.12.6 实验步骤

第 1 步 按照表 9-14 所示，配置所有的 IP 地址，并且将所有的以太接口分配到同一表所示的 VLAN 中去。

第 2 步 对 Drazen、Palmer、Almeida、Bauer 和 Ferragamo 路由器配置 OSPF 的路由。只将 Drazen、Palmer、Almeida 和 Bauer 路由器的以太接口放入到 OSPF area 0 里。

- 将 Drazen 和 Palmer 路由器的环回接口也放入到 area 0 中，Ferragamo 路由

器以及 Almeida 和 Bauer 路由器上的串口应当放入到 area 1 中。

- 将 Loopback15 的接口 IP 地址作为每一个 OSPF 路由器的 OSPF 路由器 ID。
- 使得 Almeida 和 Bauer 路由器给所有下游的邻居发送默认路由。

这个任务对内部的 24 小时网络构建了一个 IGP 的路由解决方案。当 OSPF 已经被配置完成后，所有的内部路由器都应当能够连通其他的内部路由器，除了 Internet-facing 串行接口是个例外。这个过程开始于 Almeida 和 Bauer 路由器。这个步骤中一个隐藏的任务就是需要对因特网边界路由器配置一条指向 HSRP IP 地址的默认的路由。当默认的路由被配置完成后，应当启用 OSPF 协议，并且接口应当分配到先前指定的区域中。default-information originate 命令可以发送一条默认路由给其他的 OSPF 邻居。范例 9-106 显示了对 Almeida 路由器的 OSPF 配置。

范例 9-106 Almeida 路由器的 OSPF 配置

```
Almeida# show run | begin ospf
router ospf 1
  router-id 155.206.127.107
  log-adjacency-changes
  area 1 stub
  network 155.206.127.0 0.0.0.7 area 0
  network 155.206.127.64 0.0.0.3 area 1
  network 155.206.127.107 0.0.0.0 area 0
  default-information originate always metric-type 1
!
ip route 0.0.0.0 0.0.0.0 155.206.127.5
```

当 OSPF 被配置完成后，所有的内部路由器应当能够连通所有的 OSPF 启用的接口。默认的路由也已经被通告了，这就产生了一个小问题，除非你在 Drazen 或者 Palmer 路由器上配置一个分布式列表 (distribution list) 来过滤接收的路由，否则它们将会收到一条默认路由，它们源自 Almeida 和 Bauer 路由器的 LSA。当你配置并且应用了分发列表来拒绝默认路由即 0.0.0.0/32 后，这个问题就得到了解决。可以在 Ferragamo、Drazen 和 Palmer 路由器上使用 show ip route 和 ping 命令来测试 OSPF 的配置。范例 9-107 显示了来自 Drazen 和 Ferragamo 路由器的路由表。

范例 9-107 Drazen 和 Ferragamo 的路由表

```
Drazen# show ip route | begin Gateway
Gateway of last resort is not set
  155.206.0.0/16 is variably subnetted, 7 subnets, 3 masks
C       155.206.127.0/29 is directly connected, Ethernet0/0
O       155.206.127.106/32 [110/11] via 155.206.127.2, 00:31:55, Ethernet0/0
O       155.206.127.107/32 [110/11] via 155.206.127.3, 00:31:55, Ethernet0/0
C       155.206.127.105/32 is directly connected, Loopback15
O       155.206.127.108/32 [110/11] via 155.206.127.4, 00:31:55, Ethernet0/0
O IA    155.206.127.64/30 [110/74] via 155.206.127.3, 00:31:55, Ethernet0/0
O IA    155.206.127.68/30 [110/74] via 155.206.127.4, 00:31:55, Ethernet0/0
  101.0.0.0/30 is subnetted, 1 subnets
C       101.41.12.0 is directly connected, Serial0/1.401
  154.107.0.0/30 is subnetted, 2 subnets
```

(待续)

```

C    154.107.0.4 is directly connected, Serial0/1.201
C    154.107.0.8 is directly connected, Serial0/1.101
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA  10.1.1.0/24 [110/84] via 155.206.127.3, 00:31:56, Ethernet0/0
      [110/84] via 155.206.127.4, 00:31:56, Ethernet0/0
O IA  10.3.3.1/32 [110/75] via 155.206.127.3, 00:31:56, Ethernet0/0
      [110/75] via 155.206.127.4, 00:31:56, Ethernet0/0
O IA  10.2.2.1/32 [110/75] via 155.206.127.3, 00:31:56, Ethernet0/0
      [110/75] via 155.206.127.4, 00:31:56, Ethernet0/0

```

```

Ferragamo# show ip route | begin Gateway
Gateway of last resort is 155.206.127.65 to network 0.0.0.0
155.206.0.0/16 is variably subnetted, 7 subnets, 3 masks
O IA  155.206.127.0/29 [110/74] via 155.206.127.69, 00:35:02, Serial1
      [110/74] via 155.206.127.65, 00:35:02, Serial0
O IA  155.206.127.106/32 [110/75] via 155.206.127.69, 00:32:22, Serial1
      [110/75] via 155.206.127.65, 00:32:22, Serial0
O IA  155.206.127.107/32 [110/65] via 155.206.127.65, 00:35:02, Serial0
O IA  155.206.127.105/32 [110/75] via 155.206.127.65, 00:33:44, Serial0
      [110/75] via 155.206.127.69, 00:33:44, Serial1
O IA  155.206.127.108/32 [110/65] via 155.206.127.69, 00:35:02, Serial1
C    155.206.127.64/30 is directly connected, Serial0
C    155.206.127.68/30 is directly connected, Serial1
10.0.0.0/24 is subnetted, 3 subnets
C    10.3.3.0 is directly connected, Loopback200
C    10.2.2.0 is directly connected, Loopback100
C    10.1.1.0 is directly connected, Ethernet0
O*E1 0.0.0.0/0 [110/84] via 155.206.127.65, 00:35:03, Serial0
      [110/84] via 155.206.127.69, 00:35:03, Serial1

```

第3步 在 Ferragamo 路由器上配置负载均衡，使得 OSPF 可以同时使用两个上游的串口给 155.206.127.0/29 网络转发数据包。使用适当的命令来启用负载均衡，使得属于同一股流的数据包走相同的路径。

这个步骤实际上不需要太多的配置。默认情况下，OSPF 在路由表中存储 4 条等长路径。为了在两个串行接口上启用基于目的的负载均衡，必须使用 **ip cef** 命令启用 CEF 的交换。再次强调一下，在默认情况下，**ip cef** 命令启用 CEF 的交换，使用通用的基于目的的算法来实现负载均衡。可以使用 IP 路由表和 CEF 表来验证配置。范例 9-108 显示了 Ferragamo 路由器的 IP 路由表和 **show ip cef summary** 命令的输出。

范例 9-108 Ferragamo 路由器的路由表和 CEF 汇总

```

Ferragamo# show ip route | include vialis
Gateway of last resort is 155.206.127.69 to network 0.0.0.0
155.206.0.0/16 is variably subnetted, 7 subnets, 3 masks
O IA  155.206.127.0/29 [110/74] via 155.206.127.65, 00:18:00, Serial0
      [110/74] via 155.206.127.69, 00:18:00, Serial1
O IA  155.206.127.106/32 [110/75] via 155.206.127.65, 00:18:00, Serial0
      [110/75] via 155.206.127.69, 00:18:00, Serial1
O IA  155.206.127.107/32 [110/65] via 155.206.127.65, 00:18:00, Serial0
O IA  155.206.127.105/32 [110/75] via 155.206.127.65, 00:18:00, Serial0
      [110/75] via 155.206.127.69, 00:18:00, Serial1
O IA  155.206.127.108/32 [110/65] via 155.206.127.69, 00:18:00, Serial1
C    155.206.127.64/30 is directly connected, Serial0
C    155.206.127.68/30 is directly connected, Serial1
10.0.0.0/24 is subnetted, 3 subnets
C    10.3.3.0 is directly connected, Loopback200
C    10.2.2.0 is directly connected, Loopback100

```

(待续)


```

C      10.1.1.0 is directly connected, Ethernet0
O*E1 0.0.0.0/0 [110/84] via 155.206.127.69, 00:18:01, Serial1
      [110/84] via 155.206.127.65, 00:18:01, Serial0
Ferragamo# show ip cef summary
IP CEF with switching (Table Version 28), flags=0x0
 28 routes, 0 reresolve, 0 unresolved (0 old, 0 new)
 31 leaves, 18 nodes, 22734 bytes, 31 inserts, 0 invalidations
 4 load sharing elements, 1264 bytes, 4 references
 universal per-destination load sharing algorithm, id CD1F18C5
 2 CEF resets, 0 revisions of existing leaves
 refcounts: 4907 leaf, 4864 node
Adjacency Table has 3 adjacencies

```

第 4 步 将 Ferragamo 路由器配置成为 10.1.1.0/24 网络的 DHCP 服务器。这台路由器也应当给它们的 DHCP 客户分配 fiction.org 的域名。当在路由器上配置完 DHCP 的服务后，配置 PC 使得它们可以从路由器请求一个 DHCP 的地址，并且通过 ping Drazen 路由器的环回接口来验证这个配置。

DHCP 的配置是一个很直接的步骤，当建立 DHCP 的地址池并且给这个地址池分配了 DHCP 的参数后，惟一需要做的就是排除 Ferragamo 路由器的以太网接口的 IP 地址。当 DHCP 服务器的配置完成并且 PC 被配置请求 DHCP 的 IP 地址后，它应当能够立刻 ping 通 Drazen 路由器的 IP 地址。范例 9-109 显示了 ipconfig 命令的输出和一个来自 Windows PC 的成功的 ping。

范例 9-109 来自 PC 的 ipconfig 和 ping 命令

```

G:\>ipconfig
Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix  . : fiction.org
    IP Address. . . . . : 10.1.1.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.1.1.1
G:\>ping 155.206.127.105
Pinging 155.206.127.105 with 32 bytes of data:
Reply from 155.206.127.105: bytes=32 time=20ms TTL=253
Reply from 155.206.127.105: bytes=32 time<10ms TTL=253
Reply from 155.206.127.105: bytes=32 time<10ms TTL=253
Reply from 155.206.127.105: bytes=32 time<10ms TTL=253
Ping statistics for 155.206.127.105:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 20ms, Average = 5ms

```

第 5 步 当配置完内部网络后，增加主机，并且启用路由，你现在可以集中于这个实验的 BGP 部分了。开始配置位于 AS 104 中的外部的服务提供商路由器，即 Myers 和 Gaines 路由器。在 Myers 和 Gaines 路由器上启用 BGP 路由。当你完成这个任务后，每一台路由器应当能够看到/21 的网络，这是路由器之间内部通告的网络。

在 Myers 和 Gaines 路由器之间的 I-BGP 配置只依赖于一个关键的因素：关闭 IGP 的同步。当启用 BGP 后，配置了网络和邻居，同步被关闭了，每一台路由器都应当连通它的对等体的/21 网络。范例 9-110 显示了 Myers 路由器的 IP 路由表。

范例 9-110 Myers 路由器的 IP 路由表

```

Myers# show ip route | include islvia
Gateway of last resort is not set
  154.103.0.0/21 is subnetted, 2 subnets
C       154.103.72.0 is directly connected, Loopback200
C       154.103.64.0 is directly connected, Loopback100
  154.108.0.0/21 is subnetted, 2 subnets
B       154.108.16.0 [200/0] via 172.20.20.2, 00:07:57
B       154.108.8.0 [200/0] via 172.20.20.2, 00:07:57
  154.107.0.0/30 is subnetted, 1 subnets
C       154.107.0.8 is directly connected, Serial0.100
  172.20.0.0/24 is subnetted, 1 subnets
C       172.20.20.0 is directly connected, FastEthernet0

```

第 6 步 接下来，在位于 AS 104 中的服务提供商 1 的路由器和位于 AS 8080 的 24 小时路由器之间配置 E-BGP 的路由。使用对等体组来简化 BGP 的配置。

- 使得 AS 8080 的边界路由器使用它们的 Loopback15 的 IP 地址作为 BGP 路由器 ID，并且使用它们的环回地址作为对等体的地址。在这个实例中，在 AS 104 的路由器上，允许每一台路由器对每一个邻居配置一条静态路由。
- 不要允许服务提供商 1 的路由器通告 172.20.20.0/24 的网络给任何外部的对等体。不能使用一个分布式列表来完成这个任务。
- 不要允许服务提供商路由器使用 AS 8080 的边界路由器作为一个过渡网络到达彼此的/21 网络。
- 当这个步骤完成后，在 AS 8080 中的路由器应当可以看到位于 AS 104 路由器后面的所有/21 网络。

这个步骤需要几个子任务，它们必须准确地完成，才能使得这个实验的剩余部分可以适当地完成。为了实现服务提供商 1 路由器和 24 小时边界路由器之间的 BGP 路由，必须在服务提供商 1 路由器上使用 **ebgp-multihop** 命令，在 24 小时路由器上使用 **update-source Loopback 15** 命令。如果你没有使用这些命令，这些路由器之间的 BGP 会话将不能够启用，你会在服务提供商 1 路由器上看到下面的这些信息：

```

Connections established 0; dropped 0
Last reset never
External BGP neighbor not directly connected.
No active TCP connection

```

如果你在服务提供商 1 路由器上对每一个环回 IP 地址添加了一条静态路由，当 **multihop** 和 **update-source** 命令被添加到适当的路由器后，就会启用一个 BGP 的会话。范例 9-111 显示了对 Gaines 和 Drazen 路由器的 BGP 配置。

范例 9-111 多归路 Gaines 和 Drazen 路由器

```

Gaines# show run | begin bgp
router bgp 104
no synchronization
bgp log-neighbor-changes
network 154.108.8.0 mask 255.255.248.0
network 154.108.16.0 mask 255.255.248.0
network 172.20.20.0 mask 255.255.255.0
neighbor AS8080 peer-group
neighbor AS8080 remote-as 8080

```

(待续)

```
neighbor AS8080 ebgp-multihop 2
neighbor 155.206.127.105 peer-group AS8080
neighbor 155.206.127.106 peer-group AS8080
neighbor 172.20.20.1 remote-as 104
no auto-summary
!
ip route 155.206.127.105 255.255.255.255 154.107.0.6
ip route 155.206.127.106 255.255.255.255 154.107.0.2
```

```
Drazen# show run | begin bgp
router bgp 8080
no synchronization
bgp log-neighbor-changes
network 154.107.0.8 mask 255.255.255.252
network 154.206.127.0 mask 255.255.255.248
neighbor AS104 peer-group
neighbor AS104 remote-as 104
neighbor AS104 update-source Loopback15
neighbor 154.107.0.5 peer-group AS104
neighbor 154.107.0.9 peer-group AS104
no auto-summary
```

当 BGP 的会话建立后，路由也已经交换了，你需要找到一种方法来阻止服务提供商的路由器向外部的 AS 对等体宣告 172.20.20.0/24 的私有网络。因为你不能使用路由过滤器来完成这一任务，所以只有另外一种方法来隐藏那个网络：给服务提供商的路由器分配 local AS 团体属性。这个属性允许路由在内部通告，但是阻止它被发送到外部的 BGP 对等体。范例 9-112 显示了对 Myers 路由器的 BGP 配置。

范例 9-112 在 Myers 路由器上使用共知的 LOCAL_AS 团体属性

```
Myers# show run | begin bgp
router bgp 104
no synchronization
bgp log-neighbor-changes
network 154.103.64.0 mask 255.255.248.0
network 154.103.72.0 mask 255.255.248.0
network 172.20.20.0 mask 255.255.255.0 route-map hide-network
neighbor AS8080 peer-group
neighbor AS8080 remote-as 8080
neighbor AS8080 ebgp-multihop 2
neighbor 155.206.127.105 peer-group AS8080
neighbor 172.20.20.2 remote-as 104
no auto-summary
!
ip route 155.206.127.105 255.255.255.255 154.107.0.10
!
route-map hide-network permit 10
set community local-as
```

正如你看到的，Myers 路由器使用 hide-network 路由映射来对 172.20.20.0/24 网络设置 local AS 的团体属性，因为 local AS 的团体属性并不需要宣告到本地 AS 之外，所以不需要使用 send-community 命令。

第 3 步的最后一部分指出，不能允许服务提供商网络使用 AS 8080 作为一个过渡的网络来到达内部生成的路由。这个任务需要在 24 小时路由器上添加一个 AS 路径过滤列表。一个简单的只有一行的 AS 路径过滤列表，它使用 ^\$ 的常规表达式，对所有的发送路由指定一个

空的 AS 路径，只允许通告内部生成的路由，这就达到了我们的目的。这可以在范例 9-113 中看到，它显示了 Palmer 路由器的 BGP 配置。

范例 9-113 在 Palmer 路由器上应用过滤列表

```
Palmer# show run | begin bgp
router bgp 8080
no synchronization
bgp log-neighbor-changes
network 155.206.127.0 mask 255.255.255.248
neighbor AS104 peer-group
neighbor AS104 remote-as 104
neighbor AS104 update-source Loopback15
neighbor AS104 filter-list 100 out
neighbor 154.107.0.1 peer-group AS104
no auto-summary
!
ip as-path access-list 100 permit ^$
```

第7步 为了完成 E-BGP 因特网对等体的会话，需要在位于 AS 60 中的 Farrell 路由器和 24 小时边界路由器之间配置一个 BGP 会话。这些 BGP 会话应当使用在第 6 步中指定的所有规则。

- 使用对等体组来允许进一步的对等体添加。
- 使得 AS 8080 的边界路由器使用它们的 Loopback15 IP 地址作为 BGP router ID，在 Farrell 路由器上允许对每一个邻居配置一条静态路由。
- 不要允许服务提供商路由器使用 AS 8080 的边界路由器作为一个过渡网络来到达彼此的网络。
- 当这一步完成后，位于 AS 8080 中的路由器应当可以看到由服务提供商路由器通告的所有外部网络。

如果你使用和先前的步骤中相同的步骤来配置这些路由器，那么你应当在 Drazen、Palmer 和 Farrell 路由器之间建立两个新的 BGP 会话。Myers 和 Gaines 路由器应当能够连通 155.206.127.0/29 的网络和 AS 8080 边界路由器上每一个串口的网络，但是它们不应当有任何到达 17.8.4.0/22 或者 17.8.8.0/22 网络的路由。范例 9-114 显示了 Myers 路由器的 BGP RIB 表。

范例 9-114 当应用过滤列表后 Myers 路由器的 BGP RIB

```
Myers# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
* i101.41.12.0/30 155.206.127.105   0      100      0 8080 i
*>                155.206.127.105   0              0 8080 i
*> 154.103.64.0/21 0.0.0.0           0              32768 i
*> 154.103.72.0/21 0.0.0.0           0              32768 i
* i154.107.0.4/30 155.206.127.105   0      100      0 8080 i
*>                155.206.127.105   0              0 8080 i
* i154.107.0.8/30 155.206.127.105   0      100      0 8080 i
*>                155.206.127.105   0              0 8080 i
*>i154.108.8.0/21 172.20.20.2       0      100      0 i
*>i154.108.16.0/21 172.20.20.2       0      100      0 i
* i155.206.127.0/29 155.206.127.106   0      100      0 8080 i
* i172.20.20.0/24 172.20.20.2       0      100      0 i
*>                0.0.0.0           0              32768 I
```

Drazen 和 Palmer 路由器应当有到达它们的外部 BGP 邻居的网路的路由，除了在 AS 104 中的 172.20.20.0/24 网络是个例外。范例 9-115 显示了 Drazen 路由器的 BGP RIB 表。

范例 9-115 当应用过滤列表后 Drazen 路由器的 BGP RIB

```
Drazen# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*> 17.8.4.0/22     101.41.12.1       0          0 60 i
*> 17.8.8.0/22     101.41.12.1       0          0 60 i
*> 101.41.12.0/30  0.0.0.0           0          32768 i
* 154.103.64.0/21 154.107.0.5       0          0 104 i
*>                154.107.0.9       0          0 104 i
* 154.103.72.0/21 154.107.0.5       0          0 104 i
*>                154.107.0.9       0          0 104 i
*> 154.107.0.4/30  0.0.0.0           0          32768 i
*> 154.107.0.8/30  0.0.0.0           0          32768 i
* 154.108.8.0/21  154.107.0.5       0          0 104 i
*>                154.107.0.9       0          0 104 i
* 154.108.16.0/21 154.107.0.5       0          0 104 i
*>                154.107.0.9       0          0 104 i
```

最终, Farrell 路由器的 BGP RIB 应当含有 Drazen 和 Palmer 路由器通告的所有网络表项, 除了到达 AS 104 中网路的路由, 如范例 9-116 所示。

范例 9-116 当应用过滤列表后 Farrell 路由器的 BGP RIB 表

```
Farrell# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*> 17.8.4.0/22     0.0.0.0           0          32768 i
*> 17.8.8.0/22     0.0.0.0           0          32768 i
*> 101.41.12.0/30  155.206.127.105   0          0 8080 i
*> 154.107.0.4/30  155.206.127.105   0          0 8080 i
*> 154.107.0.8/30  155.206.127.105   0          0 8080 i
*> 155.206.127.0/29 155.206.127.106   0          0 8080 I
```

第 8 步 如果在 24 小时路由器和它们的对等体路由器 (即 Almeida 和 Bauer 路由器) 之间没有配置 I-BGP 的连接, 那么 BGP 对等体的配置就不是完整的。

- 在这些路由器之间配置 I-BGP 对等关系, 使用 Loopback15 的接口地址作为对等体的地址。
- 使用对等体组来简化边界路由器的配置, 在这个网络中不要全冗余这些路由器。
- 在 AS 8080 边界路由器上汇总所有的 155.206.127.0 的网络, 不要宣告任何小于/24 的路由。
- 通过从 Ferragamo 路由器上 ping 这些因特网网络来验证配置。

这个步骤需要几步来完成成功的网络 ping 测试验证。首先, 必须在 Drazen 和 Palmer 路由器上配置一个对等体组。这个对等体组应当含有所有的特性, 它们可以应用于添加到这个对等体组 (Almeida 和 Bauer 路由器) 中的所有邻居。每一个边界路由器都需要对下游的 24 小时路由器充当一个路由反射器, 需要使用 **update-source** 和 **next-hop-self** 来实现完整的 BGP 路由能力。范例 9-117 显示了对 Drazen 路由器的 I-BGP 配置。

当配置完边界路由器后, 接下来对 Almeida 和 Bauer 路由器配置 I-BGP。Bauer 和 Almeida 路由器的配置很直接, 每个对等体只需要两个命令: **remote-as** 和 **update-source** 命令。范例

9-118 显示了对 Bauer 路由器的 BGP 配置和 BGP RIB 表。

范例 9-117 对 Drazen 路由器的 I-BGP 配置

```
Drazen# show run | include AS8080
neighbor AS8080 peer-group
neighbor AS8080 remote-as 8080
neighbor AS8080 update-source Loopback15
neighbor AS8080 route-reflector-client
neighbor AS8080 next-hop-self
neighbor 155.206.127.106 peer-group AS8080
neighbor 155.206.127.107 peer-group AS8080
neighbor 155.206.127.108 peer-group AS8080
```

范例 9-118 Bauer 路由器的 I-BGP 配置和 BGP RIB 表

```
Bauer# show run | begin bgp
router bgp 8080
no synchronization
bgp log-neighbor-changes
network 155.206.127.68 mask 255.255.255.0
neighbor 155.206.127.105 remote-as 8080
neighbor 155.206.127.105 update-source Loopback15
neighbor 155.206.127.106 remote-as 8080
neighbor 155.206.127.106 update-source Loopback15
no auto-summary
Bauer# show ip bgp | begin Network
Network          Next Hop          Metric LocPrf Weight Path
*>i17.8.4.0/22    155.206.127.105   0      100     0 60 i
* i              155.206.127.106   0      100     0 60 i
*>i17.8.8.0/22    155.206.127.105   0      100     0 60 i
* i              155.206.127.106   0      100     0 60 i
* i101.41.12.0/30 155.206.127.105   0      100     0 i
*>i              155.206.127.105   0      100     0 i
*>i154.103.64.0/21 155.206.127.105   0      100     0 104 i
* i              155.206.127.106   0      100     0 104 i
*>i154.103.72.0/21 155.206.127.105   0      100     0 104 i
* i              155.206.127.106   0      100     0 104 i
*>i154.107.0.0/30 155.206.127.105   0      100     0 104 i
* i              155.206.127.106   0      100     0 104 i
*>i154.107.0.4/30 155.206.127.105   0      100     0 i
* i              155.206.127.105   0      100     0 i
* i154.107.0.8/30 155.206.127.105   0      100     0 i
*>i              155.206.127.105   0      100     0 i
*>i154.108.8.0/21 155.206.127.105   0      100     0 104 i
* i              155.206.127.106   0      100     0 104 i
*>i154.108.16.0/21 155.206.127.105   0      100     0 104 i
* i              155.206.127.106   0      100     0 104 i
* i155.206.127.0/24 155.206.127.106   100     0 i
*>i              155.206.127.106   100     0 i
```

I-BGP 配置中的最后一步需要聚合 155.206.127.0/24 的网络，应当使用 `summary` 的参数在边界路由器上执行来抑制具体路由。注意 Ferragamo 路由器直到这个步骤完成，才能够连通任何外部的服务提供商的网络。这是因为上游的服务提供商没有到达 155.206.127.64/30 和 155.206.127.68/30 网络的路由（你应当永远都不要给服务提供商发送/30 的路由，它们通常不会接受任何小于/24 的路由）。当聚合完网络之后，你会看到 Ferragamo 路由器可以使用它的默认路由，ping 通所有的服务提供商的网络，使用的配置类似于范例 9-119 所示。

范例 9-119 Palmer 路由器的路由聚合配置

```
Palmer# show run | begin bgp
router bgp 8080
  no synchronization
  bgp router-id 154.206.127.106
  bgp cluster-id 2614001514
  bgp log-neighbor-changes
  network 155.206.127.0 mask 255.255.255.248
  aggregate-address 155.206.127.0 255.255.255.0 summary-only
  neighbor AS104 peer-group
  neighbor AS104 remote-as 104
  neighbor AS104 update-source Loopback15
  neighbor AS104 filter-list 100 out
  neighbor AS60 peer-group
  neighbor AS60 remote-as 60
  neighbor AS60 update-source Loopback15
  neighbor AS60 filter-list 100 out
  neighbor AS8080 peer-group
  neighbor AS8080 remote-as 8080
  neighbor AS8080 update-source Loopback15
  neighbor AS8080 route-reflector-client
  neighbor AS8080 next-hop-self
  neighbor 101.41.12.5 peer-group AS60
  neighbor 154.107.0.1 peer-group AS104
  neighbor 155.206.127.105 peer-group AS8080
  neighbor 155.206.127.107 peer-group AS8080
  neighbor 155.206.127.108 peer-group AS8080
  no auto-summary
```

当聚合的配置添加到边界路由器之后，因特网服务提供商路由器应当可以接收到一条到 155.206.127.0/24 网络的路由，而 Ferragamo 路由器应当可以 ping 通服务提供商的所有 155.206.127.0 的网络，如范例 9-120 所示。

范例 9-120 Farrell 的聚合 BGP RIB 和 Ferragamo 路由器的 ping 测试

```
Farrell# show ip bgp | begin Network
  Network      Next Hop      Metric LocPrf Weight Path
*> 17.8.4.0/22  0.0.0.0          0         32768 i
*> 17.8.8.0/22  0.0.0.0          0         32768 i
*> 155.206.127.0/24 155.206.127.106      0 8080 i
*               155.206.127.105      0 8080 i

Ferragamo# ping
Protocol [ip]:
Target IP address: 154.103.64.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 155.206.127.66
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 154.103.64.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/16 ms
```

第9步 为了最有效地使用边界路由器和服务提供商路由器之间的连接，配置服务提供商 1 的路由器使用来自 Drazen 路由器的路由，配置服务提供商 2 的路由器使用来自 Palmer 路由器的路由，多出口鉴别器或者 AS 路径属性都不可以用于完成这个任务。本地产生的路由应当总是具有最高的优先级：

- Drazen 路由器应当优先使用来自 Myers 路由器的路由，其次选用来自 Farrell 路由器的路由；而 Palmer 路由器应当优先使用来自 Farrell 路由器的路由，其次选用 Gaines 路由器的路由，最后是 Myers 路由器的路由。本地产生的路由应当总是具有最高的优先级。

有几种方法可以在 BGP 中设置优先的路由；其中一种最简单和最常用的方法就是给次优的路径添加 AS 路径信息，或者给次优的路径设置多出口鉴别器属性。当外部的对等体收到了具有新属性的路由后，BGP 的路径选择算法会优选具有最短 AS 路径的路由，或者具有最低的多出口鉴别器属性的路由。另外一种解决这个问题方法就是设置并且匹配一个特定的 BGP 团体属性，在接收端使用路由映射将权重属性设置为一个较高的值，使得这样的路由更具有优选性。范例 9-121 显示了 Drazen 路由器如何使用 **route map external-pref** 来将所有由 **match route-type local** 命令产生的本地路由的团体属性值设置为 104:8080，而将所有其他发送路由的团体属性值设置为 104:111。**ip bgp-community new-format** 命令允许使用更可读的 **aa:nn** 的团体格式。

范例 9-121 在 Drazen 路由器上改变团体属性

```
Drazen# show run | include AS104new-format
neighbor AS104 peer-group
neighbor AS104 remote-as 104
neighbor AS104 update-source Loopback15
neighbor AS104 send-community
neighbor AS104 route-map external-pref out
neighbor AS104 filter-list 100 out
neighbor 154.107.0.5 peer-group AS104
neighbor 154.107.0.9 peer-group AS104
ip bgp-community new-format
Drazen# show run | begin route-map external-pref permit 10
route-map external-pref permit 10
match route-type local
set community 104:8080
!
route-map external-pref permit 20
set community 104:111
```

当在 AS 104 中的外部对等体收到具有新的团体属性值的路由后，它们同样可以在路由映射中使用相同的类型来设置 WEIGHT（权重）属性。范例 9-122 显示了 Gaines 路由器如何使用 IP 团体列表 10、11 和 80 来匹配接收的团体属性值并且根据这些值来设置权重。

在先前的范例中，路由映射语句 10 匹配团体属性值 104:8080，这个值来自团体列表 10，并且将匹配的路由的权重属性值从默认的 0 增加到一个新的值 10 000。路由映射语句 20 匹配来自团体列表 11 中的团体属性值 104:111，而路由映射 30 的语句匹配默认的 Internet 属性，并不修改其他属性。如果没有路由映射 30 语句，这个路由映射将充当一个访问控制列表并且拒绝所有其他的路由。范例 9-123 显示了来自 Gaines 路由器的最终结果的 BGP RIB。

范例 9-122 在 Gaines 路由器上使用团体属性来改变权重

```
Gaines# show run | begin AS8080
neighbor AS8080 peer-group
neighbor AS8080 remote-as 8080
neighbor AS8080 ebgp-multihop 2
neighbor AS8080 route-map preference in
neighbor 155.206.127.105 peer-group AS8080
neighbor 155.206.127.106 peer-group AS8080
neighbor 172.20.20.1 remote-as 104
no auto-summary
!
ip bgp-community new-format
ip community-list 10 permit 104:8080
ip community-list 11 permit 104:111
ip community-list 80 permit internet
!
route-map preference permit 10
match community 10
set weight 10000
!
route-map preference permit 20
match community 11
set weight 2000
!
route-map preference permit 30
match community 80
```

范例 9-123 当调整新的权重属性后，Gaines 路由器的 BGP RIB

```
Gaines# show ip bgp | begin Network
Network      Next Hop      Metric LocPrf Weight Path
*>i154.103.64.0/21 172.20.20.1      0    100      0 i
*>i154.103.72.0/21 172.20.20.1      0    100      0 i
*> 154.108.8.0/21 0.0.0.0          0          32768 i
*> 154.108.16.0/21 0.0.0.0          0          32768 i
* 155.206.124.0/22 155.206.127.106      0          0 8080 i
* i           155.206.127.105      100         0 8080 i
*>           155.206.127.105      2000 8080 i
* i172.20.20.0/24 172.20.20.1      0    100      0 i
*>           0.0.0.0          0          32768 i
```

这个步骤的第二部分需要给进入到 24 小时网络的路由配置路由优先级。乍一看，你可能想要使用本地优先属性来改变这些路由的优先级；然而，如果你认真地读了问题的话，你会注意到本地优先属性在这种情况下是不工作的，因为本地优先属性会传递给位于 AS 8080 中的所有邻居，它不会产生所需要的结果。完成这个任务的另外一种方法就是使用 **set** 和 **match** 来应用团体属性，并且使用那个属性来改变路由的权重属性，就像你在这个步骤中的第一部分所做的那样。这次，这个任务稍微麻烦一点，这是因为优先级的 3 种顺序。范例 9-124 显示了这在 Drazen 路由器上是如何完成的。

范例 9-124 在 Drazen 路由器上修改路由的优先级

```
Drazen# show run | include AS104|AS60
neighbor AS104 peer-group
neighbor AS104 remote-as 104
```

(待续)

```

neighbor AS104 update-source Loopback15
neighbor AS104 send-community
neighbor AS104 route-map internal-pref in
neighbor AS104 route-map external-pref out
neighbor AS104 filter-list 100 out
neighbor AS60 peer-group
neighbor AS60 remote-as 60
neighbor AS60 update-source Loopback15
neighbor AS60 send-community
neighbor AS60 route-map internal-pref in
neighbor AS60 route-map external-pref2 out
neighbor AS60 filter-list 100 out
neighbor 101.41.12.1 peer-group AS60
neighbor 154.107.0.5 peer-group AS104
neighbor 154.107.0.9 peer-group AS104
Drazen# show run | include community-list
ip community-list 4 permit 104:104
ip community-list 10 permit internet
ip community-list 14 permit 104:222
ip community-list 44 permit 104:333
Drazen# show run | begin route-map internal-pref permit 10
route-map internal-pref permit 10
  match community 4
  set weight 10000
!
route-map internal-pref permit 20
  match community 14
  set weight 2000
!
route-map internal-pref permit 30
  match community 44
  set weight 1000
!
route-map internal-pref permit 40
  match community 10

```

在先前的范例中，路由映射 internal-pref 指定权重将分配给每一个带有团体属性值的路由。路由映射 internal-pref 10 使用团体列表 4 将所有本地产生的路由（在 Myers 和 Gaines 路由器上设置的具有团体属性值 104:104 的路由）的权重值设置为 1000。这个路由映射的下一个实例就是匹配始发于 Myers 路由器的流量（这个值在 Myers 路由器上被设置为 104:22），这个路由映射的再下一个实例指定来自 Gaines 路由器的路由（这个值在 Gaines 路由器上被设置为 104:333），最后一条语句允许任何其他的路由，这些路由的团体属性值没有变化。范例 9-125 显示了最终的 BGP RIB。

范例 9-125 在 Drazen BGP RIB 中指定优先级

```

Drazen# show ip bgp | begin Network

```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 117.8.4.0/22	155.206.127.106	0	100	0 60	i
*>	101.41.12.1	0		0 60	i
* 117.8.8.0/22	155.206.127.106	0	100	0 60	i
*>	101.41.12.1	0		0 60	i
* 1154.103.64.0/21	155.206.127.106		100	0 104	i
*>	154.107.0.9	0		10000 104	i
*	154.107.0.5			1000 104	i

(待续)

```

* i154.103.72.0/21 155.206.127.106 100 0 104 i
*> 154.107.0.9 0 10000 104 i
* 154.107.0.5 1000 104 i
* i154.108.8.0/21 155.206.127.106 0 100 0 104 i
* 154.107.0.9 2000 104 i
*> 154.107.0.5 0 10000 104 i
* i154.108.16.0/21 155.206.127.106 0 100 0 104 i
* 154.107.0.9 2000 104 i
*> 154.107.0.5 0 10000 104 i
s> 155.206.124.0/24 0.0.0.0 0 32768 i
* i155.206.124.0/22 155.206.127.106 100 0 i
*> 0.0.0.0 32768 i
s> 155.206.125.0/24 0.0.0.0 0 32768 i
s> 155.206.126.0/24 0.0.0.0 0 32768 i
r>i155.206.127.64/30
Network Next Hop Metric LocPrf Weight Path
#
155.206.127.107 0 100 0 i

```

第 10 步 作为一个安全上的注意事项，应当在 24 小时边界路由器上关掉任何 CDP、HTTP web 访问和任何不必要的特性。

- 也建立一个反欺骗的访问控制列表，它可以防止任何 RFC 1918 的私有的 IP 地址和内部地址。
- 确保 OSPF 路由不允许通告到 24 小时网络之外。
- 在 Internet-facing 路由器上启用 HTTP web 服务；它们将用于仿真因特网 web 服务器。
- 配置 HTTP 的服务使用 Loopback100 接口的 IP 地址。

范例 9-126 显示了可能已经发出的某些命令，这取决于思科 IOS 软件的版本。

范例 9-126 关闭思科路由器上的服务

```

no service pad
no service dhcp
no ip identd
no service finger
no ip source-route
no ip bootp
no service tcp-small-servers
no service tcp-small-servers
!
interface Ethernet0/0
no mop enabled
no cdp enable
ip access-group 101 in
!
router ospf 1
passive-interface Serial0/1
passive-interface Serial0/1.101
passive-interface Serial0/1.201
passive-interface Serial0/1.401
!
no ip http server
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any

```

(待续)

```
access-list 101 deny ip 172.0.0.0 0.31.255.255 any
access-list 101 deny ip 154.206.127.0 0.0.0.255 any
access-list 101 permit any any
!
no cdp run
```

第 11 步 为了从因特网上隐藏 RFC 1918 的私有地址，配置 24 小时边界路由器将所有的内网地址翻译成因特网可路由的公网地址，如表 9-15 所示。

- 确保所有的 IP 地址块可以聚合成最好的地址。所有的具体路由都应当被抑制住；只有聚合的路由可以通告给外部的邻居。
- 为了验证用户到因特网的连接性，使用一个 web 浏览器来进入每一个服务提供商网络的 HTTP web 配置站点。

这个步骤需要一些 NAT 和 BGP 的配置正常工作。首先，必须配置 NAT 使得任何内部的路由网络都被静态地翻译成一个外部的 IP 地址，如果这个步骤没有正确地配置，数据包将不会被转发和正确地返回。为了对于这种情况配置 NAT，需要配置一个静态网络地址翻译，如范例 9-127 所示。

范例 9-127 Drazen 路由器的 NAT 配置

```
Drazen# show run ! include nat inside source
ip nat inside source static network 10.1.1.0 155.206.124.0 /24
ip nat inside source static network 10.2.2.0 155.206.125.0 /24
ip nat inside source static network 10.3.3.0 155.206.126.0 /24
```

可以使用 **show ip nat translations** 命令来检查一个成功的 NAT 的翻译。当 PC 发出一个 ping 数据包，目的是任何因特网的 IP 地址时，你应当在其中的一个边界路由器上看到成功的翻译。范例 9-128 显示了对于 Drazen 路由器的 NAT 的翻译。

范例 9-128 Drazen 路由器的 NAT 表

```
Drazen# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 155.206.124.2      10.1.1.2         ---               ---
Subnet translation:
Inside global  Inside local  Outside local  Outside global /prefix
155.206.124.0  10.1.1.0     ---           ---           /24
155.206.125.0  10.2.2.0     ---           ---           /24
155.206.126.0  10.3.3.0     ---           ---           /24
```

为了使得上游的服务提供商网络能够连通新的翻译后的 IP 地址，它们必须在边界路由器上通过 BGP 通告出去。当你在 BGP 的进程中增加 155.206.124.0/24、155.206.125.0/24 和 155.206.126.0/24 网络时，这些网络可以聚合成一个更大的网络——155.206.124.0/22。范例 9-129 显示了在 Drazen 路由器上的新的 BGP 配置。

范例 9-129 Drazen 路由器 NAT/BGP 配置变化

```
network 155.206.124.0 mask 255.255.255.0
network 155.206.125.0 mask 255.255.255.0
network 155.206.126.0 mask 255.255.255.0
aggregate-address 155.206.124.0 255.255.252.0 summary-only
```

范例 9-131 完整的路由器配置

```
hostname Frame-Relay-Switch
!
frame-relay switching
!
interface Serial0
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 101 interface Serial1 100
frame-relay route 201 interface Serial2 200
frame-relay route 401 interface Serial4 400
!
interface Serial1
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 100 interface Serial0 101
!
interface Serial2
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 200 interface Serial0 201
frame-relay route 300 interface Serial3 301
!
interface Serial3
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 301 interface Serial2 300
frame-relay route 501 interface Serial4 500
!
interface Serial4
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 400 interface Serial0 401
frame-relay route 500 interface Serial3 501
```

Myers# show run | begin hostname

```
hostname Myers
!
interface Loopback100
ip address 154.103.64.1 255.255.248.0
!
interface Loopback200
ip address 154.103.72.1 255.255.248.0
!
interface FastEthernet0
ip address 172.20.20.1 255.255.255.0
!
interface Serial0
no ip address
encapsulation frame-relay
clockrate 1300000
```

(待续)

```

frame-relay lmi-type ansi
!
interface Serial0/100 multipoint
 ip address 154.107.0.9 255.255.255.252
 frame-relay map ip 154.107.0.10 100 broadcast
!
router bgp 104
 no synchronization
 bgp log-neighbor-changes
 network 154.103.64.0 mask 255.255.248.0
 network 154.103.72.0 mask 255.255.248.0
 network 172.20.20.0 mask 255.255.255.0 route-map hide-network
 neighbor AS8080 peer-group
 neighbor AS8080 remote-as 8080
 neighbor AS8080 ebgp-multihop 2
 neighbor AS8080 send-community
 neighbor AS8080 route-map preference in
 neighbor AS8080 route-map external-pref out
 neighbor 155.206.127.105 peer-group AS8080
 neighbor 172.20.20.2 remote-as 104
 no auto-summary
!
ip route 155.206.127.105 255.255.255.255 154.107.0.10
ip http server
ip bgp-community new-format
ip community-list 11 permit 104:111
ip community-list 80 permit internet
!
route-map preference permit 10
 match community 11
 set weight 2000
!
route-map preference permit 20
 match community 80
!
route-map external-pref permit 10
 match route-type local
 set community 104:104
!
route-map external-pref permit 20
 set community 104:222
!
route-map hide-network permit 10
 set community local-as

```

```

Gaines# show run | begin host
hostname Gaines
!
!
interface Loopback100
 ip address 154.108.8.1 255.255.248.0
!
interface Loopback200
 ip address 154.108.16.1 255.255.248.0
!
interface FastEthernet0
 ip address 172.20.20.2 255.255.255.0
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 1300000

```

(待续)

```

frame-relay lmi-type ansi
!
interface Serial1.200 multipoint
 ip address 154.107.0.5 255.255.255.252
 frame-relay map ip 154.107.0.6 200 broadcast
!
interface Serial1.300 multipoint
 ip address 154.107.0.1 255.255.255.252
 frame-relay map ip 154.107.0.2 300 broadcast
!
router bgp 104
 no synchronization
 bgp log-neighbor-changes
 network 154.108.8.0 mask 255.255.248.0
 network 154.108.16.0 mask 255.255.248.0
 network 172.20.20.0 mask 255.255.255.0 route-map hide-network
 neighbor AS8080 peer-group
 neighbor AS8080 remote-as 8080
 neighbor AS8080 ebgp-multihop 2
 neighbor AS8080 send-community
 neighbor AS8080 route-map preference in
 neighbor AS8080 route-map external-pref out
 neighbor 155.206.127.105 peer-group AS8080
 neighbor 155.206.127.106 peer-group AS8080
 neighbor 172.20.20.1 remote-as 104
 no auto-summary
!
ip route 155.206.127.105 255.255.255.255 154.107.0.6
ip route 155.206.127.106 255.255.255.255 154.107.0.2
ip http server
ip bgp-community new-format
ip community-list 10 permit 104:8080
ip community-list 11 permit 104:111
ip community-list 80 permit internet
!
route-map preference permit 10
 match community 10
 set weight 10000
!
route-map preference permit 20
 match community 11
 set weight 2000
!
route-map preference permit 30
 match community 80
!
route-map external-pref permit 10
 match route-type local
 set community 104:104
!
route-map external-pref permit 20
 set community 104:333
!
route-map hide-network permit 10
 set community local-as

```

```

Farrell# show run | begin host
hostname Farrell
!
interface Loopback100
 ip address 17.8.4.1 255.255.252.0

```

(待续)

```

!
interface Loopback200
  ip address 17.8.8.1 255.255.252.0
!
interface Serial0
  no ip address
  encapsulation frame-relay
  clockrate 1300000
  frame-relay lmi-type ansi
!
interface Serial0.400 multipoint
  ip address 101.41.12.1 255.255.255.252
  frame-relay map ip 101.41.12.2 400 broadcast
!
interface Serial0.500 multipoint
  ip address 101.41.12.5 255.255.255.252
  frame-relay map ip 101.41.12.6 500 broadcast
!
router bgp 60
  no synchronization
  bgp log-neighbor-changes
  network 17.8.4.0 mask 255.255.252.0
  network 17.8.8.0 mask 255.255.252.0
  neighbor AS8080 peer-group
  neighbor AS8080 remote-as 8080
  neighbor AS8080 ebgp-multihop 2
  neighbor AS8080 send-community
  neighbor AS8080 route-map preference in
  neighbor AS8080 route-map external-pref out
  neighbor 155.206.127.105 peer-group AS8080
  neighbor 155.206.127.106 peer-group AS8080
  no auto-summary
!
ip route 155.206.127.105 255.255.255.255 101.41.12.2
ip route 155.206.127.106 255.255.255.255 101.41.12.6
ip http server
ip bgp-community new-format
ip community-list 11 permit 60:111
ip community-list 60 permit internet
!
route-map preference permit 10
  match community 11
  set weight 2000
!
route-map preference permit 20
  match community 60
!
route-map external-pref permit 10
  match route-type local
  set community 60:60
!
route-map external-pref permit 20
  set community 60:222

```

```

Drazen# show run | begin host
hostname Drazen
!
no ip source-route
!
no ip bootp server
!
interface Loopback15

```

(待续)


```
ip address 155.206.127.105 255.255.255.255
!
interface Ethernet0/0
ip address 155.206.127.1 255.255.255.248
ip nat inside
!
interface Serial0/1
no ip address
encapsulation frame-relay
clockrate 1300000
frame-relay lmi-type ansi
!
interface Serial0/1.101 multipoint
ip address 154.107.0.10 255.255.255.252
ip access-group 101 in
ip nat outside
frame-relay map ip 154.107.0.9 101 broadcast
!
interface Serial0/1.201 multipoint
ip address 154.107.0.6 255.255.255.252
ip access-group 101 in
ip nat outside
frame-relay map ip 154.107.0.5 201 broadcast
!
interface Serial0/1.401 multipoint
ip address 101.41.12.2 255.255.255.252
ip access-group 101 in
ip nat outside
frame-relay map ip 101.41.12.1 401 broadcast
!
router ospf 1
router-id 155.206.127.105
log-adjacency-changes
passive-interface Serial0/1
passive-interface Serial0/1.101
passive-interface Serial0/1.201
passive-interface Serial0/1.401
network 155.206.127.0 0.0.0.7 area 0
network 155.206.127.105 0.0.0.0 area 0
distribute-list f in
!
router bgp 8080
no synchronization
bgp log-neighbor-changes
network 154.206.127.0 mask 255.255.255.248
network 155.206.124.0 mask 255.255.255.0
network 155.206.125.0 mask 255.255.255.0
network 155.206.126.0 mask 255.255.255.0
aggregate-address 155.206.124.0 255.255.252.0 summary-only
neighbor AS104 peer-group
neighbor AS104 remote-as 104
neighbor AS104 update-source Loopback15
neighbor AS104 send-community
neighbor AS104 route-map internal-pref in
neighbor AS104 route-map external-pref out
neighbor AS104 filter-list 100 out
neighbor AS60 peer-group
neighbor AS60 remote-as 60
neighbor AS60 update-source Loopback15
neighbor AS60 send-community
neighbor AS60 route-map internal-pref in
neighbor AS60 route-map external-pref2 out
```

(待续)

```

neighbor AS60 filter-list 100 out
neighbor AS8080 peer-group
neighbor AS8080 remote-as 8080
neighbor AS8080 update-source Loopback15
neighbor AS8080 route-reflector-client
neighbor AS8080 next-hop-self
neighbor 101.41.12.1 peer-group AS60
neighbor 154.107.0.5 peer-group AS104
neighbor 154.107.0.9 peer-group AS104
neighbor 155.206.127.106 peer-group AS8080
neighbor 155.206.127.107 peer-group AS8080
neighbor 155.206.127.108 peer-group AS8080
no auto-summary
!
ip nat inside source static network 10.1.1.0 155.206.124.0 /24
ip nat inside source static network 10.2.2.0 155.206.125.0 /24
ip nat inside source static network 10.3.3.0 155.206.126.0 /24
ip route 155.206.124.0 255.255.255.0 Null0 254
ip route 155.206.125.0 255.255.255.0 Null0 254
ip route 155.206.126.0 255.255.255.0 Null0 254
no ip http server
ip bgp-community new-format
ip community-list 4 permit 104:104
ip community-list 10 permit internet
ip community-list 14 permit 104:222
ip community-list 44 permit 104:333
ip as-path access-list 100 permit ^$
!
access-list 1 deny 0.0.0.0
access-list 1 permit any
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
access-list 101 deny ip 172.0.0.0 0.31.255.255 any
access-list 101 deny ip 154.206.127.0 0.0.0.255 any
access-list 101 permit ip any any
no cdp run
!
route-map external-pref2 permit 10
match route-type local
set community 60:8080
!
route-map external-pref2 permit 20
set community 60:111
!
route-map internal-pref permit 10
match community 4
set weight 10000
!
route-map internal-pref permit 20
match community 14
set weight 2000
!
route-map internal-pref permit 30
match community 44
set weight 1000
!
route-map internal-pref permit 40
match community 10
!
route-map external-pref permit 10
match route-type local

```

(待续)

```
set community 104:8080
!
route-map external-pref permit 20
set community 104:111
!
Palmer# show run | begin host
hostname Palmer
!
no ip source-route
!
interface Loopback15
ip address 155.206.127.106 255.255.255.255
!
interface Ethernet0
ip address 155.206.127.2 255.255.255.248
ip nat inside
!
interface Serial0
no ip address
encapsulation frame-relay
clockrate 1300000
frame-relay lmi-type ansi
!
interface Serial0.301 multipoint
ip address 154.107.0.2 255.255.255.252
ip access-group 101 in
ip nat outside
frame-relay map ip 154.107.0.1 301 broadcast
!
interface Serial0.501 multipoint
ip address 101.41.12.6 255.255.255.252
ip access-group 101 in
ip nat outside
frame-relay map ip 101.41.12.5 501 broadcast
!
router ospf 1
router-id 155.206.127.106
log-adjacency-changes
passive-interface Serial0
passive-interface Serial0.301
passive-interface Serial0.501
network 155.206.127.0 0.0.0.7 area 0
network 155.206.127.106 0.0.0.0 area 0
distribute-list 1 in
!
router bgp 8080
no synchronization
bgp router-id 154.206.127.106
bgp log-neighbor-changes
network 155.206.124.0 mask 255.255.255.0
network 155.206.125.0 mask 255.255.255.0
network 155.206.126.0 mask 255.255.255.0
network 155.206.127.0 mask 255.255.255.248
aggregate-address 155.206.124.0 255.255.252.0 summary-only
neighbor AS104 peer-group
neighbor AS104 remote-as 104
neighbor AS104 update-source Loopback15
neighbor AS104 send-community
neighbor AS104 route-map internal-pref in
neighbor AS104 route-map external-pref out
neighbor AS104 filter-list 100 out
```

(待续)

```

neighbor AS60 peer-group
neighbor AS60 remote-as 60
neighbor AS60 update-source Loopback15
neighbor AS60 send-community
neighbor AS60 route-map internal-pref in
neighbor AS60 route-map external-pref2 out
neighbor AS60 filter-list 100 out
neighbor AS8080 peer-group
neighbor AS8080 remote-as 8080
neighbor AS8080 update-source Loopback15
neighbor AS8080 route-reflector-client
neighbor AS8080 next-hop-self
neighbor 101.41.12.5 peer-group AS60
neighbor 154.107.0.1 peer-group AS104
neighbor 155.206.127.105 peer-group AS8080
neighbor 155.206.127.107 peer-group AS8080
neighbor 155.206.127.108 peer-group AS8080
no auto-summary
!
ip nat inside source static network 10.1.1.0 155.206.124.0 /24
ip nat inside source static network 10.2.2.0 155.206.125.0 /24
ip nat inside source static network 10.3.3.0 155.206.126.0 /24
ip route 155.206.124.0 255.255.255.0 Null0 254
ip route 155.206.125.0 255.255.255.0 Null0 254
ip route 155.206.126.0 255.255.255.0 Null0 254
no ip http server
ip bgp-community new-format
ip community-list 10 permit internet
ip community-list 11 permit 60:60
ip community-list 11 permit 104:104
ip community-list 14 permit 104:333
ip community-list 60 permit 60:222
ip as-path access-list 100 permit ^$
!
access-list 1 deny 0.0.0.0
access-list 1 permit any
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
access-list 101 deny ip 172.0.0.0 0.31.255.255 any
access-list 101 deny ip 154.206.127.0 0.0.0.255 any
access-list 101 permit ip any any
no cdp run
!
route-map external-pref2 permit 10
match route-type local
set community 60:8080
!
route-map external-pref2 permit 20
set community 60:111
!
route-map internal-pref permit 10
match community 11
set weight 10000
!
route-map internal-pref permit 20
match community 60
set weight 2000
!
route-map internal-pref permit 30
match community 14
set weight 1000

```

(待续)

```
!  
route-map internal-pref permit 40  
  match community 10  
!  
route-map external-pref permit 10  
  match route-type local  
  set community 104:8080  
!  
route-map external-pref permit 20  
-----  
Almeida# show run | begin host  
hostname Almeida  
!  
ip cef  
!  
interface Loopback15  
  ip address 155.206.127.107 255.255.255.255  
!  
interface Ethernet0  
  ip address 155.206.127.3 255.255.255.248  
!  
interface Serial0  
  ip address 155.206.127.65 255.255.255.252  
  clockrate 1300000  
!  
router ospf 1  
  router-id 155.206.127.107  
  log-adjacency-changes network 155.206.127.0 0.0.0.7 area 0  
  network 155.206.127.64 0.0.0.3 area 1  
  network 155.206.127.107 0.0.0.0 area 0  
  default-information originate always metric-type 1  
!  
router bgp 8080  
  no synchronization  
  bgp log-neighbor-changes  
  network 155.206.127.64 mask 255.255.255.252  
  neighbor 155.206.127.105 remote-as 8080  
  neighbor 155.206.127.105 update-source Loopback15  
  neighbor 155.206.127.106 remote-as 8080  
  neighbor 155.206.127.106 update-source Loopback15  
  no auto-summary  
!  
ip route 0.0.0.0 0.0.0.0 155.206.127.5  
-----  
Bauer# show run | begin host  
hostname Bauer  
!  
ip cef  
!  
interface Loopback15  
  ip address 155.206.127.108 255.255.255.255  
!  
interface Ethernet0  
  ip address 155.206.127.4 255.255.255.248  
!  
interface Serial0  
  ip address 155.206.127.69 255.255.255.252  
  clockrate 1300000  
!  
router ospf 1  
  router-id 155.206.127.108
```

(待续)

```
log-adjacency-changes network 155.206.127.0 0.0.0.7 area 0
network 155.206.127.68 0.0.0.3 area 1
network 155.206.127.108 0.0.0.0 area 0
default-information originate always metric-type 1
!
router bgp 8080
no synchronization
bgp log-neighbor-changes
network 155.206.127.68 mask 255.255.255.0
neighbor 155.206.127.105 remote-as 8080
neighbor 155.206.127.105 update-source Loopback15
neighbor 155.206.127.106 remote-as 8080
neighbor 155.206.127.106 update-source Loopback15
no auto-summary
!
ip route 0.0.0.0 0.0.0.0 155.206.127.5

Ferragamo# show run | begin host
hostname Ferragamo
!
ip dhcp excluded-address 10.1.1.1
!
ip dhcp pool workstations
network 10.1.1.0 255.255.255.0
default-router 10.1.1.1
domain-name fiction.org
!
interface Loopback100
ip address 10.2.2.1 255.255.255.0
!
interface Loopback200
ip address 10.3.3.1 255.255.255.0
!
interface Ethernet0
ip address 10.1.1.1 255.255.255.0
!
interface Serial0
ip address 155.206.127.66 255.255.255.252
!
interface Serial1
ip address 155.206.127.70 255.255.255.252
!
router ospf 1
log-adjacency-changes network 10.1.1.0 0.0.0.255 area 1
network 10.2.2.0 0.0.0.255 area 1
network 10.3.3.0 0.0.0.255 area 1
network 155.206.127.64 0.0.0.3 area 1
network 155.206.127.68 0.0.0.3 area 1
```

9.13 进一步阅读资料

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*, by A. Heffeman

Cisco IOS Dial Solutions, by Cisco Systems, Inc.

www.apnic.net—Asia Pacific Network Information Centre

www.arin.net—The American Registry for Internet Numbers

www.ripe.net—RIPE Network Coordination Centre

www.isoc.org—The Internet Society

www.nanog.org—The North American Network Operators' Group

第六部分

CCIE 练习实验

第 10 章

CCIE 准备和练习实验

10.1 CCIE 准备

为了获得成功，我们愿意做任何工作。我们必须付出代价，但是这样的代价是值得的，必须付出这种代价才有可能获得成功。最重要的是必须坚持下去。牺牲、坚定不移、竞争的动力、无私和尊敬权威，是我们每个人必须为成功所付出的代价。这使你能够忽略次要的伤害、对手的压力和暂时的失败。

加入 CCIE 行列意味着你正在成为世界上最精英和能力最强的网络工程师团队中的一员。获得这个成员资格要付出很高的代价。承诺你自己达到一个很少人能做到的水平。考试准备这段时间的压力可能是巨大的，你必须在这个压力下几乎毫无瑕疵地完成任务。

很幸运，你不是独自寻求。随着我们全速向信息时代的前进，即使仍然有.com 的倒闭，对高技能的网络工程师的需求一直存在。随着越来越多的工程师准备 CCIE 和其他类似考试，有了更多可以利用的工具。学习组，如 routerie.com 和 groupstudy.com，是从其他正在准备考试的人（能感觉你的痛苦的人）获得帮助的主要地方。关于路由协议、交换、安全和很多其他论题的新书每年都会出版。

正如你所看到的，成为一名 CCIE 需要在时间、金钱和个人付出方面承担很多。

模拟 CCIE 实验室所需的设备可能非常贵。诸如 Ascolta Training、Skyline Computer、Network Learning 和其他一些公司以合理的费用提供实验室、ISDN 交换机和 CCIE 准备资料。关于建立 CCIE 实验室的详细细节，参考《CCIE 实验指南（第 1 卷）》第 1 章。

本。下面列出了你学习时很有价值的一些书籍，其中一部分已由人民邮电出版社翻译或影印出版，详情请查阅人民邮电出版社网站：www.ptpress.com.cn。

Stevens: *TCP/IP Illustrated*, Volume I

Comer: *Internetworking with TCP/IP*

Pearlman: *Interconnections: Routers and Bridges*, Second Edition

Doyle: *Routing TCP/IP*, Volume I

Doyle/Carroll: *Routing TCP/IP*, Volume II

Solie: *CCIE Practical Studies*, Volume I

Solie/Lynch: *CCIE Practical Studies*, Volume II

Halabi: *Internetwork Routing Architectures*, Second Edition

Clark/Hamilton: *Cisco LAN Switching*

Caslow: *Bridges, Routers, and Switches*

Cisco Press: *CCIE Design and Case Studies*, Second Edition

Diker-Pildush: *Cisco ATM Solutions*

Cisco Press: *Troubleshooting IP Routing Protocols*

Cisco IOS Software 12.1 and 12.2 configuration guides (尽力多读)

下面的列表决不是 CCIE 学习论题的全部列表。然而，它是一系列论题的切入点，CCIE 投考者应该对此非常熟悉。

- 帧中继
 - 帧中继交换
 - 帧中继子接口
 - 点对点链路和多点链路
 - 帧中继映射综述：bridge、LLC、DLSW 和其他关键字
 - RFC 1490 封装
 - Bridging over Frame
 - Voice over Frame
 - PPP over Frame
 - 帧中继 ARP 和反 ARP 操作
 - 帧中继流量整形
- HDLC
 - 压缩类型
- PPP
 - PPP 认证：PAP/CHAP
 - PPP 回叫
 - PPP 链路
 - DDR 技术
 - 虚拟拨号器预置文件
 - 压缩类型
 - IPCP
- ISDN

- 拨号器映射/DDR
- 了解如何处理基于 ISDN 的路由协议，如 RIP、EIGRP、OSPF 等
- 瞬态路由
- 拨号器查看
- OSPF 请求电路
- BGP
 - BGP 理论，包括思科路由器上的 BGP 操作
 - I-BGP 与 E-BGP 的对比
 - BGP 同步规则
 - 路由反射器
 - 隐藏自治系统号和创建私有自治系统
 - 认证
 - BGP 后门
 - 路由映射和路由重分发
 - 自治系统路径过滤
 - BGP 路径选择过程和路径操作：多出口鉴别器、本地优先、权重等
 - BGP 联盟
 - BGP 团体
 - 超网通告，汇总
 - BGP 和 IGP 交互
 - BGP 属性
 - 自治系统路径和团体过滤，包括规则表达式
 - 前缀抑制
 - 条件路由通告
 - 路由衰减
- OSPF
 - 不同路由协议间互相重分发
 - 用 summary address 和 area range 语句来汇总
 - OSPF over Frame
 - OSPF 请求电路
 - OSPF 的路由映射和路由过滤
 - OSPF 代价和管理距离
 - 末梢区域、NSS 区域、骨干区域和 LSA 传播
 - 认证：类型 I 和类型 II
 - 认证 area 0
 - 指定路由器和 BDR 的选择：priority 命令
 - 默认路由传播
- EIGRP
 - 基于 IP 的 EIGRP
 - 不同路由协议间互相重分发

- 汇总
- EIGRP 的路由映射和路由过滤
- MD5 认证
- 基于 ISDN 的 EIGRP
- 多点网络的水平分割问题
- 所有路由协议的管理距离
- EIGRP 末梢网络
- RIP
 - 不同路由协议间互相重分发
 - 基于 ISDN 的瞬态路由/RIP
 - 多点网络的水平分割问题
 - RIPv1, 缺乏 VLSM 支持的问题
 - RIPv2
 - RIP 单播更新
- IS-IS
 - 不同路由协议间互相重分发
 - CLNS
 - 基于帧中继的 IS-IS
 - IS-IS 类型 1 和类型 2 路由
- DLSw
 - TCP、FST、direct 和帧中继对等体
 - 备份对等体
 - 混杂对等体
 - 边界对等体及对等体组
 - Costed 对等体
 - 用 DLSw LSAP 过滤器做探测器控制和 LLC 控制
- 桥
 - 透明桥接
 - 生成树控制
 - IEEE 802.1w 和 IEEE 802.1s
 - 基于帧中继的桥接
 - 源路由桥接
 - 远程源路由桥接
 - 转换桥接
 - 探测器控制和泛洪
 - LSAP 过滤器
 - 集成的路由和桥接
 - 默认网关
- 控制路由和流量
 - 标准访问列表

- 扩展访问列表
- 命名访问列表
- 定时访问列表
- 动态反身访问列表
- 路由映射和策略路由
- 传播默认路由
- 队列
 - 一般和帧中继流量整形
 - RSVP, WRED 基本配置
 - 检查路由器配置最优化
 - 路由交换：进程、快速、CEF、NetFlow、优化和分布式
 - 压缩技术——前向和堆栈
 - 快速的 ATM PVC 原理和配置回顾，包括新的 IOS atm 命令
 - ATM 与 Frame Relay 的比较
 - ATM 服务质量
 - RSVP 的集成服务
 - 基于 IP ToS、Precedence、DSCP 和 WRED 的区分服务
 - 先进先出队列
 - 加权公平队列
 - 优先级队列
 - 定制队列
 - 基于类别的加权公平队列
 - 低延迟队列
 - IP RTP 优先级
 - 常规整形、帧中继流量整形和分类整形
 - 流量策略
 - 承诺的访问速率
- 通用的 IOS
 - 访问服务器配置
 - 跳数寄存器配置
 - Catalyst 和路由器的口令恢复
 - EXEC 控制：超时设定、权限等级等
 - 安全：加密隧道、CONS 和 vty 访问
 - 控制台和系统日志
- IOS 特性
 - NAT：动态、静态和地址池以及 TCP 超载
 - NTP：NTP 认证和层设置
 - DNS
 - HSRP：跟踪和优先级
 - IDRP

- DHCP
- 瞬态路由
- 拨号器查看
- 移动 IP
- ARP 操作
- SNMP: read/write 关键字, 设置和获取 traps
- UDP 泛洪: **ip forward** 命令
- GRE 隧道和认证
- Catalyst
 - 创建 Catalyst 3550 VLAN
 - VTP 域
 - 高级生成树控制
 - 生成树: IEEE 802.1b、IEEE 802.1w 和 IEEE 802.1s
 - 端口安全和 IP 访问控制
 - VLAN 映射
 - ISL, 802.1Q 骨干
 - 骨干上的 VLAN 传播和控制
 - VLAN 间的路由
 - 组播路由
 - SVI 和路由端口
 - 三层交换/路由
 - 基于 802.1s 的 STP 负载分担
 - Voice VLAN
 - Layer 2 和 Layer 3 以太通道
- 组播路由
 - 加入组播组
 - 稀疏和密集模式操作
 - IGMP 和 CGMP
 - Catalyst 3550 的组播问题
- ATM
 - ATM 上的传统 IP, 路由
 - VPI、VCD 和 VCI 定义
 - ARP 控制
 - PVC 映射
- Voice
 - Voice over IP
 - Voice over Frame
 - Voice over ATM
 - FXO 和 FXS 和 E&M 电路
 - H.323

- VPN（通常为安全实验考试）
 - 加密类型
 - IPSec 保护的 GRE 隧道
 - IPSec 传输和 tunnel 模式
 - 转换设置，加密映射
 - “Key” 认证
 - CA 认证
- 删除的论题（下面的论题已在 2003 年删除）
 - ATM LANE
 - AppleTalk
 - LAT
 - DECnet
 - Apollo
 - Banyan VINES
 - ISO CLNS
 - XNS
 - X.25
 - IGRP
 - IPX
 - 令牌环和令牌环交换
 - Catalyst 5500 或 CAT-OS 配置

CCIE 路由和交换实验（2003 年 11 月）正式的设备列表如下：

- 2600 系列路由器
- 3600 系列路由器
- Catalyst 3550 系列交换机
- 3700 系列路由器
- 从 2003 年 7 月 7 日到 2003 年 8 月 31 日，CCIE 项目将迁移到思科 IOS 软件 12.2 版本。在迁移期间，所有的考试仍然基于思科 IOS 软件 12.1 的内容和目的。注意：思科 IOS 软件 12.2 的特性和命令 2003 年 9 月 1 日前不考。

10.2 CCIE 练习实验

CCIE 练习实验是为了给你一个 CCIE 实验考试实际情况的正确描述。一些实验室是完全的 CCIE 实验室，要求硬件支持语音、ATM 和 2 台思科 3550；其他的没有很严格的硬件要求。你可以根据自己的硬件环境，对实验做方便的改动。我们知道不是每个人都有机会使用 ATM、语音和 3550，因此每个实验有不同的硬件需求。

在每个实验之前，提供全部的设备列表和前一阶段的信息——如帧中继交换配置、骨干

路由器配置等。使用这个信息模拟你自己的 CCIE 实验。

在创作《CCIE 实验指南（第 1 卷）》时关于是否要包含练习实验的答案有一些争议。很多人，包括我自己，认为提供了答案，人们会更关注于匹配答案本身，而不在做实验上。然而，我们确实认识到有时看看答案是非常有帮助的。定制队列问题就是一个好的范例。鉴于这些原因以及依据读者的要求，我们决定把答案含在 CD-ROM 中。

关于更多的实验信息和更新，请点击思科出版社网站：www.ciscopress.com。

不要忘记实践是你学习中最关键的部分。你第一次花费数个小时解决其中的一个问题是常见的。事实上，如果我们不在某处难住你，那么我们就没有做好我们的工作。路由环路、路由反馈问题、水平分割等都是些常见问题，我们称这些为“CCIE 地雷”。没有实际地做实验，你会错过很多这些有趣的冒险经历。如果你被一个问题难住，尽力抵制看答案的诱惑。有时最好的学习方法是通过花费很长时间来寻找和理解方案。目的不是提出一个解决方案；那是隐含的。最终目的是实践，实践，再实践。

牢记这点，我们提供给你 5 个 CCIE 练习实验。

10.3 CCIE 练习实验：Broken Arrow

设备列表：

- 1 台帧中继交换机：4 个串行接口
- 具有 2 个 BRI 端口的 ISDN 模拟器/交换机
- 具有 2 个 ATM 接口的 ATM 交换机
- 3 台实验路由器：1 个以太网口和 1 个串行接口
- 1 台实验路由器：1 个以太网口，1 个 ATM 接口，1 个串行接口，1 个 ISDN BRI 接口
- 1 台实验路由器：1 个以太网口，1 个 ISDN BRI 接口
- 1 台实验路由器：2 个以太网口
- 1 台实验路由器：1 个以太网口，1 个 ATM 接口
- 2 台装有 EMI 软件的以太网 3550 交换机，2 个光口或交叉电缆，用于网络互连

10.3.1 准备阶段——帧中继交换机和 ATM 配置

按照图 10-1 中的描述配置帧中继交换机的 PVC。不要计算你自己做这部分实验的时间。帧中继交换机的配置是一个全局的配置，会用在除第 3 个实验以外的所有实验中。不是图中所有的 PVC 都会在这个实验中使用。用实线表示的 PVC 是这个实验中使用的 PVC，用点线表示的 PVC 在这个实验中不使用。现在使用 ATM 配置来配置 ATM 交换机，配置一个接口到路由器 3，一个接口到路由器 7。范例 10-1 列出了帧中继和 ATM 交换机的配置。

帧中继DLCI映射
不是所有的PVC和接口都被使用

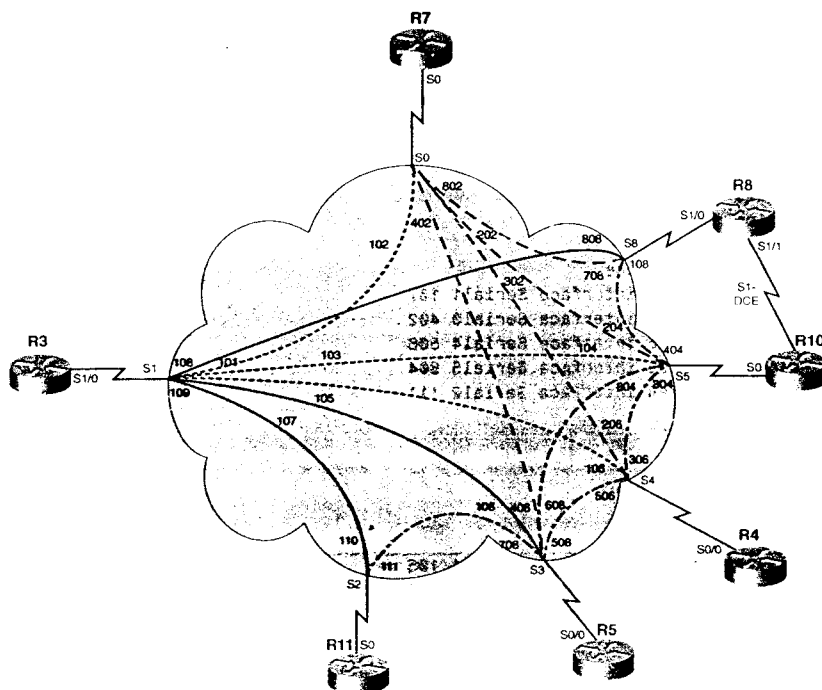


图 10-1 帧中继交换机配置

范例 10-1 帧中继和 ATM 交换机配置

```
hostname frame_switch
!
frame-relay switching
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
clockrate 2000000
frame-relay intf-type dce
frame-relay route 102 interface Serial1 101
frame-relay route 202 interface Serial5 204
frame-relay route 302 interface Serial4 206
frame-relay route 402 interface Serial3 408
frame-relay route 802 interface Serial8 708
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 2000000
frame-relay intf-type dce
frame-relay route 101 interface Serial0 102
frame-relay route 103 interface Serial5 104
frame-relay route 105 interface Serial4 106
frame-relay route 107 interface Serial3 108
frame-relay route 108 interface Serial8 808
frame-relay route 109 interface Serial2 110
```

(待续)


```

!
interface Serial2
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 110 interface Serial1 109
 frame-relay route 111 interface Serial3 708
!
interface Serial3
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 108 interface Serial1 107
 frame-relay route 408 interface Serial0 402
 frame-relay route 508 interface Serial4 506
 frame-relay route 608 interface Serial5 804
 frame-relay route 708 interface Serial2 111
!
interface Serial4
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 106 interface Serial1 105
 frame-relay route 206 interface Serial0 302
 frame-relay route 306 interface Serial5 304
 frame-relay route 506 interface Serial3 508
!
interface Serial5
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 104 interface Serial1 103
 frame-relay route 204 interface Serial0 202
 frame-relay route 304 interface Serial4 306
 frame-relay route 404 interface Serial8 108
 frame-relay route 804 interface Serial3 608
!
interface Serial8
 no ip address
 encapsulation frame-relay
 clockrate 64000
 frame-relay intf-type dce
 frame-relay route 108 interface Serial5 404
 frame-relay route 708 interface Serial0 802
 frame-relay route 808 interface Serial1 108
!
no ip classless
!
end

```

LIGHTSTREAM CONFIGURATION

```

hostname r12_ls1010
!
atm address 47.0091.8100.0000.0061.705b.4001.0061.705b.4001.00
!
interface ATM0/0/0

```

(待续)

```

no keepalive
!
interface ATM0/0/1
no keepalive
atm pvc 1 88 interface ATM0/0/0 1 77
!
interface ATM0/0/2
no keepalive
!
interface ATM2/0/0
no ip address
no keepalive
atm maxvp-number 0
!
interface Ethernet2/0/0
no ip address
!
no ip classless
!
line con 0
line aux 0
line vty 0 4
login
!
end

```

下面的实验部分要计时，在所有硬件配置和物理安装后开始。

10.3.2 规则

- 除非特别规定，不允许使用静态路由或浮动静态路由。
- 严格按照指示去做。注意只在要求的地点和时间传播路由。只能按照说明中的指示使用 PVC。
- 可以使用配置向导和 CD-ROM 作为惟一的参考资料。
- 你有 8.5 小时来完成这部分实验。在这个阶段不要和任何人谈话。
- 建议你在开始前阅读整个实验。

10.3.3 第 I 部分：IP 设置

1. 在路由器 11 的 E0 接口使用 IP 子网 145.10.1.19/27。
2. 用下面的子网创建虚接口：
 - 路由器 11 上的 LB20- 145.10.128.64/26
 - 路由器 10 上的 LB20- 172.19.1.0/24 和 LB21-172.18.1.0/24
 - 路由器 5 上的 LB20- 206.191.1.0/24
 - 交换机 15_3550 上的 VLAN X – 145.10.192.15/24
 - 交换机 15_3550 上的 VLAN Y – 145.10.193.15/24
3. 所有其他的子网和主机地址使用网络 145.10.0.0：
 - VLAN A: 27-bit 子网
 - VLAN B: 29-bit 子网

- VLAN D: 24-bit 子网
- VLAN F: 24-bit 子网

10.3.4 第 II 部分：Catalyst 配置

1. 在交换机 15_3550 和交换机 16_3550 之间配置 802.1Q 骨干。使用 Gig 0/1 和 Gig 0/2 接口做冗余。不要把 IP 地址设在 Gigabit 接口（这个实验可以使用 2 个 100BASE-T 接口）。
2. 把交换机 16_3550 配置为 VTP 服务器，交换机 15_3550 是客户端。使用 PSV2 作为 VTP 域名，ccie 作为 VTP 密码。
3. 按照图 10-2 的描述配置 VLAN。不要使用 VLAN 1。

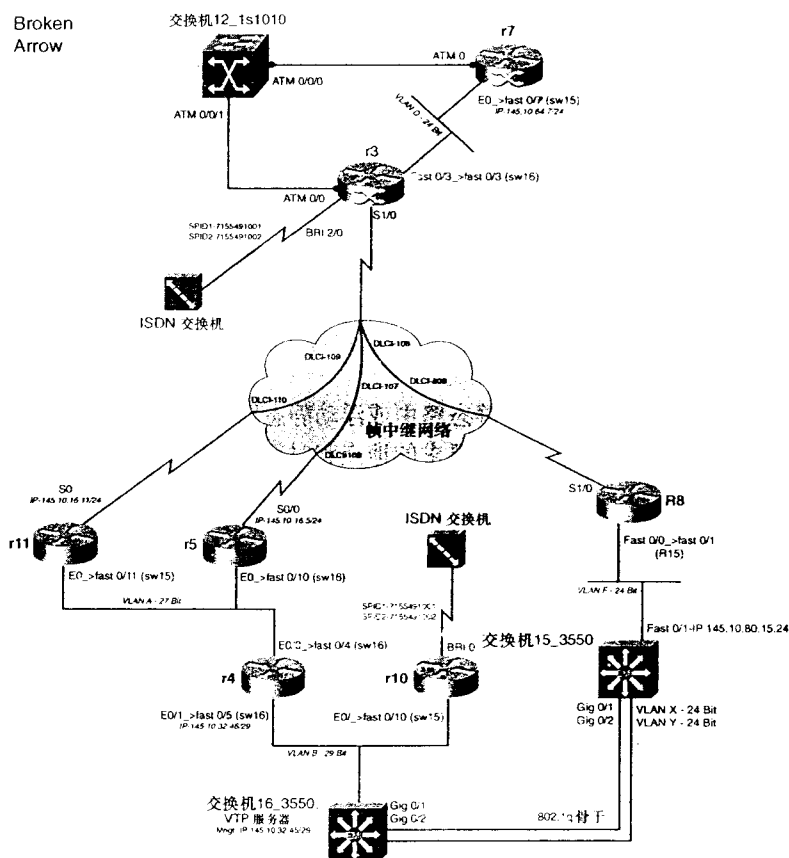


图 10-2 Broken Arrow 网络图

4. 配置交换机 16_3550 作为现在和以后所有 VLAN 的根，除了 VLAN 800 以外。交换机 15_3550 是 VLAN 800 的根和其余 VLAN 的备份根交换机。如果创建一个新的 VLAN，它要遵循这些 STP 指导，没有额外的配置。
5. 配置到所有交换机的远程登录（Telnet）访问，不要使用 VLAN 1。在实验中管理地址能从任何地方到达。交换机 15_3550 应该被 VLAN F 管理，VLAN B 用来管理交换机 16_3550。使用 cisco 作为密码。

6. 在两台交换机之间配置 IEEE 802.1w RSTP。保证如果 1 个吉比特以太网骨干 down 掉，99% 的流量仍然能通过。也就是说，RSTP 能够在 1s 内收敛网络，而不是 50s。在吉比特以太网链路断路的情况下，用扩展 ping 测试路由器 11 到路由器 5 的连接。99% 的成功率表示 RSTP 在工作。RSTP 应该在所有端口快速收敛，包括接路由器的端口（两个吉比特接口可以用两个快速以太网接口代替，这不会改变实验的功能性）。

7. 保证所有在用的端口正在运用 802.1w，包括主机/路由器的端口。

10.3.5 第 III 部分：OSPF、RIP 和帧中继

1. 在路由器 3、路由器 11 和路由器 5 之间配置帧中继网络，使得它们共享相同的 IP 子网 145.10.16.0/24。

2. 在路由器 3、路由器 11 和路由器 5 之间配置帧中继网络，使其在 OSPF area 0 中。不要配置静态 OSPF 邻居。

3. 配置 VLAN A 在 OSPF area 100 中。路由器 11、路由器 5 和路由器 4 都有一个以太网接口在 area 100 中。配置 VLAN D 和帧中继网络在 OSPF area 0 中。

4. 在路由器 3 和路由器 8 之间配置帧中继网络。配置这个网络和 VLAN F 在一个 RIP 域中。

5. 交换机 15_3550 的接口 FastEthernet 0/1 使用 IP 地址 145.10.80.15。配置这个接口和路由器 8 交换单播 RIP 的更新。

6. 如果有必要，配置 3 层交换，使得全部 IP 可达的 VLAN 能够彼此 ping 通。确保能从路由器 11 ping 到 VLAN X 和 VLAN Y 上的地址。

7. 确保 OSPF 域和 RIP 域之间的全部 IP 可达。

8. 为重分发到 OSPF 中的路由，配置一个等于自治系统边界路由器（ASBR）的主机名的标签。例如，如果路由器 2 是一个 ASBR，当你在路由器 2 上重分发任何路由协议到 OSPF 时，为那些路由设置一个值为 2 的标签。

9. 配置路由器 3 和路由器 8，使得所有的 RIP 路由有 95 的管理距离。

10.3.6 第 IV 部分：EIGRP 集成

1. 在路由器 10、路由器 4 和交换机 16_3550 之间用 AS 2003 配置 EIGRP。

2. 通过 EIGRP 通告路由器 10 上的环回网络 LB21-172.19.1.0/24 和 LB20-172.18.1.0/24。阻止 RIP 域看到 172.19.1.0/24 路由。路由器 7 应看到 2 个 172 路由。

3. 确保 EIGRP、OSPF 和 RIP 域之间的全部 IP 可达。保证路由器 10 能 ping 到路由器 7，交换机 15_3550 上的 VLAN X 和 VLAN Y。

10.3.7 第 V 部分：流量控制和 ISDN

1. 配置路由器 4，使得从 VLAN B 去往 VLAN D 的远程登录流量将穿过路由器 5。从 VLAN B 去往 VLAN D 的 ping 将穿过路由器 11。所有其他流量应沿着路由/转发表中的方向。

2. 在路由器 10 和路由器 3 之间配置 ISDN 网络。使用以下指导：

— 当任何方式的 IP 连接失败时，配置路由器 10 来建立呼叫。

— 使用 PPP CHAP 作认证；用 cisco 作为密码。

- 拨号器不应由于路由协议的原因而一直保持 up 状态。
- 不要使用静态路由；路由应为动态的。
- 为了穿过 ISDN 链路，可以配置额外的路由协议。
- 达到最小门限值时路由器 10 就启用第二条 B 信道。
- 3min 的空闲时间后链路断开。

10.3.8 第 VI 部分：BGP

1. 在路由器 4 和路由器 10 之间配置 BGP。

- 在 BGP 表中所有 I-BGP 路由可达；不能改变下一跳属性。不能使用路由反射器或联盟。
- 把两台路由器放到 AS 144 中。
- 每台路由器必须使用它的 VLAN B 的 IP 地址作为 BGP 标识。
- 路由器 4 应当只通告 145.10.0.0/18 和 206.191.1.0/24 网络。这一项可以使用一条仅指向一个接口（没有 IP 地址）的静态路由。
- 路由器 10 应当只通告 145.10.64.0/18 和 145.10.128.0/18 网络。
- 两台路由器都不能通告私有地址空间。
- 两台路由器都应该显式使用 BGP 软设置。
- 两台路由器都有有效的可达 BGP 路由到它们邻居的聚合网络。

2. 在路由器 7 与 AS 144 中的两台路由器之间配置 BGP。

- 把路由器 7 放到 AS 12501 中。
- 配置这台路由器使用它的以太网 IP 地址作为 BGP 路由器 ID。
- AS 144 路由器应该也与以太网 IP 地址对等。
- 创建两个环回接口：一个在 193.164.80.0/20 网络中，一个在 214.148.12.0/22 网络中。
- 通告那些网络到 E-BGP 对等体。
- 使路由器 7 优选从路由器 4 到 145.10.0.0/18、145.10.64.0/18 和 206.191.1.0/24 网络的路由；这个任务中不能使用 AS 路径属性。
- 把路由器 3 加到 AS 12501 中：使用直连接口作为每个 BGP 邻居的对等点。不要从这台路由器通告新的路由。

3. 把路由器 11 和路由器 5 加到 AS 144 中；使用直连接口作为每个 BGP 邻居的对等点。不要从这些路由器通告新的路由。这些路由器应该使用它们的串口作为它们的 BGP 路由器 ID。

- 使路由器 11 与路由器 3、路由器 4 和路由器 5 对等。
- 使路由器 5 与路由器 3、路由器 11 和路由器 4 对等。
- 路由器 8 和交换机不应该参与 BGP 路由或者学习 BGP 路由。所有的 BGP 路由器应该能够 ping 到任何其他 BGP 通告的网络。

10.3.9 第 VII 部分：服务质量和 ATM

在路由器 7 和路由器 3 之间配置 ATM 接口。

- 路由器 7 应使用 VPI/VCI 1/77，路由器 3 应使用 1/88。
- 两台路由器应该能够在将来的某些时候增加到这条电路的其他多点连接。
- 两台路由器必须有显式的 PVC 配置；ATM 交换不能依赖 PVC 的配置。
- 两台路由器对突发数据流量应使用最高的 ATM 服务等级，SCR 为 1.544，PCR 为 2.048bit/s。
- 使用 BGP 来通告 /20 ATM 网络汇总，不要使用 **network** 命令。不要通过 IGP 协议通告这个网络。记住不要通告私有网络。
- 在拥塞时，帧中继路由器应该基于 IP 优先级值丢弃数据包；来自网络 145.10.32.0/29 的流量应该有最高的非受控优先级值。
- 配置这些路由器采用最佳拥塞避免算法来阻止基于 IP 优先级值的尾丢弃。

10.3.10 第 VIII 部分：DLSW+

1. 在路由器 10 的 VLAN B 和路由器 3 的 VLAN D 之间配置一个 DLSw TCP 对等体。当 ISDN 链路收敛时，对等体保持激活和不断开。
2. 配置另一个 DLSw TCP 对等体从路由器 5 的 VLAN A 到路由器 3 的 VLAN D。这个对等体只是在 VLAN A 上有 NetBIOS 流量发起时才激活。最后的电路断开 3min 后对等体断掉。
3. 不能在路由器 3 上配置 **remote-peer** 语句。

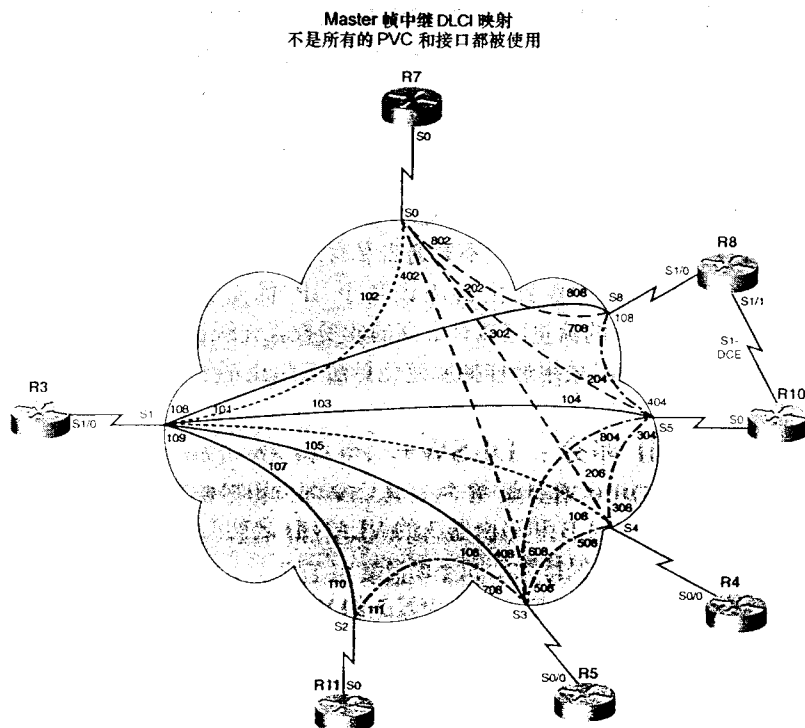
10.4 CCIE 练习实验:!!! Boom...

设备列表：

- 1 台帧中继交换机：4 个串行端口
- 具有 2 个 BRI 端口的 ISDN 模拟器/交换机
- 2 台实验路由器：1 个以太网接口
- 1 台实验路由器：1 个快速以太网口，1 个串行接口，1 个 ATM 接口，1 个 ISDN BRI 接口
- 1 台实验路由器：1 个以太网接口，1 个 ISDN BRI 接口，1 个串行接口
- 1 台实验路由器：2 个以太网接口
- 1 台实验路由器：1 个串行接口和 1 个以太网接口
- 1 台装有 EMI 软件的以太网 3550 交换机，2 个光口或交叉电缆，用于网络互连
- 1 台有快速或吉比特以太信道的 35xx 以太网交换机

10.4.1 准备阶段——帧中继交换机、骨干路由器和 ATM 配置

按照图 10-3 中的描述配置帧中继交换机的 PVC。不要计算你自己做这部分实验的时间。不是图中所有的 PVC 都要使用。用实线表示的 PVC 是你使用的。还要配置骨干路由器 5 和路由器 11 以及 ATM 交换机。范例 10-2 列出了帧中继和 ATM 交换机的配置。范例 10-3 列出了骨干路由器 5 和路由器 11 的配置。



```
frame-relay route 107 interface Serial3 108
frame-relay route 108 interface Serial8 808
frame-relay route 109 interface Serial2 110
!
interface Serial2
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 110 interface Serial1 109
frame-relay route 111 interface Serial3 708
!
interface Serial3
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 108 interface Serial1 107
frame-relay route 408 interface Serial0 402
frame-relay route 508 interface Serial4 506
frame-relay route 608 interface Serial5 804
frame-relay route 708 interface Serial2 111
!
interface Serial4
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 106 interface Serial1 105
frame-relay route 206 interface Serial0 302
frame-relay route 306 interface Serial5 304
frame-relay route 506 interface Serial3 508
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 104 interface Serial1 103
frame-relay route 204 interface Serial0 202
frame-relay route 304 interface Serial4 306
frame-relay route 404 interface Serial8 110
frame-relay route 804 interface Serial3 608
!
interface Serial6
no ip address
!
interface Serial7
no ip address
!
interface Serial8
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 108 interface Serial5 404
frame-relay route 708 interface Serial0 802
frame-relay route 808 interface Serial1 108
!
interface Serial9
no ip address
```

(待续)


```

shutdown.
!
interface BRI0
  no ip address
  shutdown
!
no ip classless
!
line con 0
line aux 0
line vty 0 4
  login
!
end

..... backbone routers .....→

```

范例 10-3 骨干路由器 5 和骨干路由器 11 配置

```

hostname backbone_router_r5
!
clns routing
!
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
dls w local-peer peer-id 141.200.5.5 promiscuous
dls w icanreach netbios-name backbone_rtr5
dls w bridge-group 1
!
interface Ethernet0/0
  ip address 141.200.5.5 255.255.255.0
  ip router isis
  bridge-group 1
!
interface Serial0/0
  no ip address
  encapsulation frame-relay
  no ip mroute-cache
!
interface Serial0/0.1 point-to-point
  ip address 140.200.1.1 255.255.255.0
  ip router isis
  no ip mroute-cache
  frame-relay interface-dlci 108
!
interface Serial0/1
  no ip address
  shutdown
  clns router isis
!
router isis
  redistribute connected metric 30 metric-type internal level-1
  distance 140
  net 00.0001.0050.736b.7800.00

```

(待续)

```

!
ip classless
!
!
bridge 1 protocol ieee
!
end
----->
hostname backbone_router_r11
!
ip subnet-zero
!
isdn voice-call-failure 0
!
interface Loopback20
 ip address 192.200.16.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback21
 ip address 192.200.17.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback22
 ip address 192.200.18.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback23
 ip address 192.200.19.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback24
 ip address 192.200.20.11 255.255.255.0
 no ip directed-broadcast
!
interface Ethernet0
 description to fast 0/11 on sw15_3550
 ip address 129.200.17.11 255.255.255.0
 no ip directed-broadcast
!
<<<text omitted>>>
!
router rip
 network 129.200.0.0
 network 192.200.16.0
 network 192.200.17.0
 network 192.200.18.0
 network 192.200.19.0
 network 192.200.20.0
!
end

```

下面的实验部分要计时，在所有硬件配置和物理安装后开始。

10.4.2 规则

- 除非特别规定，不允许使用静态路由或浮动静态路由。
- 严格按照指示去做。注意只在要求的地点和时间传播路由。只能按照说明中的指示使用 PVC。
- 可以使用配置向导和 CD-ROM 作为惟一的参考资料。

- ### 10.4.3 第 I 部分：IP 设置

!!!BOOM

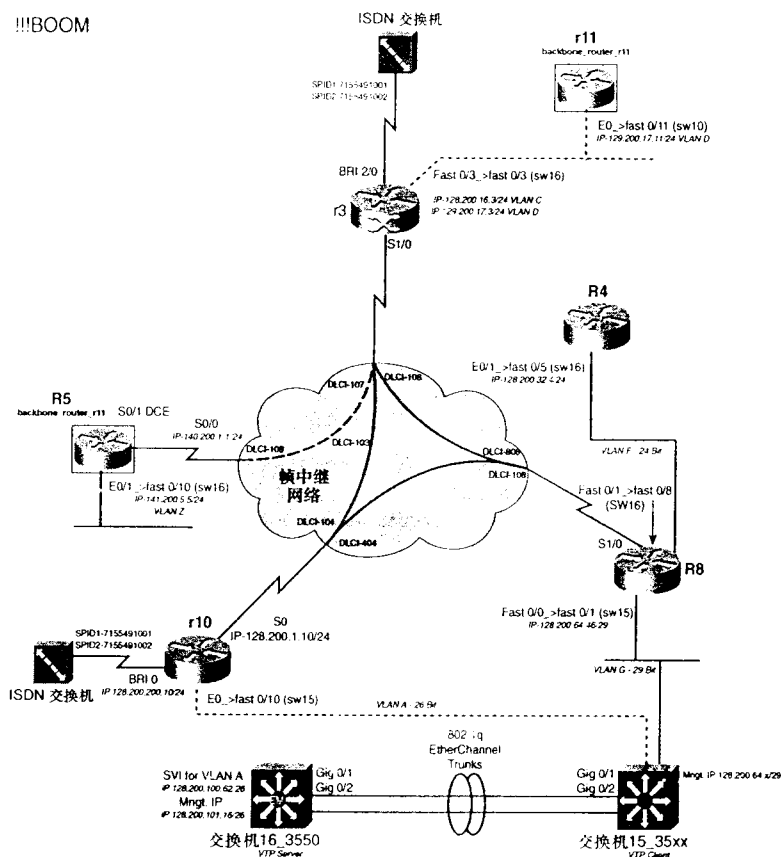


图 10-4 !!! Boom... 网络图

2. 路由器 3 的 Fast 3/0 接口使用 IP 地址 128.200.16.3/24 和 129.200.17.3/24。子网 128.200.16.0/24 使用 VLAN C, 129.200.17.0/24 子网使用 VLAN D。
3. 路由器 4 的 E0/1 使用 IP 地址 128.200.32.4/24。这个接口在 VLAN F 中。
4. 路由器 8 的 Fast 0/1 接口在 VLAN F 中, Fast 0/0 接口在 VLAN G 中。Fast 0/0 接口使用 IP 地址 128.200.64.46/29。
5. 路由器 10 的 s0 接口将使用 IP 地址 128.200.1.10/24。
6. 所有其他的子网和主机地址使用网络 128.200.0.0:
 - VLAN A: 26-bit 子网
 - VLAN C、D、F、X、Z: 24-bit 子网

— VLAN G: 6 个可用的主机地址

10.4.4 第 II 部分: Catalyst 配置

1. 使用 Gig 0/1 和 Gig 0/2 接口在交换机 15_35xx 和交换机 16_3550 之间配置一个 802.1Q 吉比特以太信道骨干（这个实验中你能以 100BASE-T 接口替代）。不要把 IP 地址设在吉比特接口上。以太信道骨干应配置 PAGP。

2. 按照图 10-4 中的描述配置 VLAN。

3. 将交换机 16_3550 配置为 VTP 服务器，交换机 15_35xx 为客户端。使用 ccie 作为 VTP 域名和密码做保护。

4. 通过下面的操作，允许访问交换机的所有配置权限：

- 在交换机 16_3550 的 VLAN X 上配置一个管理地址 128.200.101.16/24。在交换机 15_3550 的 VLAN G 上配置一个管理地址。用户应当使用用户名 ccie，密码 psv2 来认证。
- 每个交换机只允许 2 个远程登录会话。如果到同一台交换机的第三个远程登录会话被打开，它将失败。实验中从所有的路由器都可以配置和到达交换机。

10.4.5 第 III 部分: OSPF、三层交换和帧中继

1. 在路由器 3、路由器 10 和路由器 8 之间配置一个全网状连接的帧中继网络，使得它们共享相同的 IP 子网 128.200.1.0/24。只能使用路由器 3 上的子接口。不能改变帧中继接口上的 IP OSPF 网络类型。

2. 在路由器 3、路由器 10 和路由器 8 之间配置帧中继网络，使其在 OSPF area 0 中。

3. 配置 VLAN A 使其在 OSPF area 200 中。

4. 不要在到路由器 11 的 VLAN D、路由器 8 的 VLAN F 和路由器 8 的 VLAN G 的骨干上运行 OSPF。

5. 配置交换机 16_3550 的 VLAN X，使其在 OSPF area 300 中。配置交换机 16_3550 的 VLAN A，使其在 OPSF area 200 中。

6. 当路由器上创建了一条链路-状态类型 5，它应当用创建它的路由器编号作标记。例如，如果路由器 4 创建了一条链路-状态类型 5，它应当有一个标签 4。

确保 OSPF 域到 RIP、EIGRP、IS-IS 域全 IP 连接。

10.4.6 第 IV 部分: RIP、EIGRP、IS-IS 集成

1. 配置 VLAN D 在与骨干路由器 11 相连的 RIP 域中。当连接骨干路由器 11 时，应接收到下面的 RIP 路由：192.200.16.0/24，192.200.17.0/24，192.200.18.0/24，192.200.19.0/24 和 192.200.20.0/24。确保所有的 OSPF 路由器能够到达这些路由。

2. 只在 VLAN F 和 VLAN G 上配置 EIGRP。不要使用被动的接口命令来完成这个任务。EIGRP、OSPF 和 RIP 域之间全部可达。

3. 在路由器 3 和骨干路由器 5 之间配置 IS-IS。确保你从骨干路由器 5 能看到 IS-IS 路由

141.200.5.0/24。

4. 确保所有的路由域能够彼此到达。确保交换机 16_3550 能够发送 100 个直接 ping 包到所有的 OSPF 和 IS-IS 帧中继接口、IS-IS 路由 141.200.5.0/24 以及来自 RIP 域的 192.200.x.x 路由。

10.4.7 第 V 部分：路由过滤和控制

1. 阻止骨干路由器 11 看到任何 IS-IS 路由，140.200.1.0/24 和 141.200.5.0/24。用一个两行的访问控制列表来实现。

2. 只允许路由器 4 看到来自 RIP 域的偶数子网。用一个两行的访问控制列表来实现。

10.4.8 第 VI 部分：ISDN

在路由器 10 和路由器 3 之间配置 ISDN 网络。使用下面的指导：

- 在路由器 10 上使用 IP 地址 128.200.200.10/24。这个子网应该在 OSPF area 0 中。
- 拨号器不应由于路由协议的原因而一直保持 up 状态。配置路由器 10 仅当失去帧中继服务时才建立呼叫。
- 用 PPP CHAP 作认证，用 cisco_isdn 作为密码。
- 不要使用静态路由，路由应该为动态的。
- 当链路的流出流量超过 32 kbit/s 时，路由器 10 应该启用第二个 B 信道。
- 空闲 5min 后链路断开。

10.4.9 第 VII 部分：BGP

除非另外说明，这部分不允许使用静态路由。不要通告 BRI 接口到 BGP 中。除非另外说明，BGP 路由不能被重分发到 IGP 路由协议。所有路由器优先选用 IGP 路由，而不用任何 BGP 路由。在被通告到 E-BGP 邻居之前，所有 BGP 路由应该聚合成最小网络前缀。每个 BGP 对等体关系使用一个静态更新源和 BGP 路由器 ID。所有路由器应使用尽可能最小的配置行来进行 BGP 配置。按照自治系统编号来组织 BGP 对等体。

1. 为路由器 3、路由器 8 和路由器 10 配置 BGP 路由；把所有这些路由器放在 AS 5300 中。在帧中继网络中，使这些路由器中的每一台与路由器 5 对等。所有 AS 5300 的路由器应该通告所有直接连接的网络，通告到外部对等体的路由应使用最小的网络前缀编号汇总。

2. 除了前面的配置项，路由器 3 应配置成与路由器 11 对等，传播所有路由器 11 的路由到它的 I-BGP 对等体。

3. 在路由器 5 上配置 BGP 路由；把这台路由器放在 AS 12 中，把它配置成与 AS 5300 中的路由器对等。通告连接的网络；然后在 4.0.0.0/8 和 5.5.0.0/16 网络创建环回接口，通告这些网络到所有的 BGP 邻居。

4. 路由器 11 上的 BGP 路由器在 AS 500 中。把它配置成与路由器 3 对等。这台路由器对所有的邻居使用 BGP 认证，使用密码 abc123。在路由器 11 上创建 2 个环回，分配到网络 11.0.0.0/8 和 12.0.0.0/8 中，通告这些网络到所有的 BGP 对等体。配置这台路由器使得发送到 AS 5300 中的路由器的路由不传播；不能改变 AS 5300 中的路由器来支持这个配置。

5. 在路由器 4 上配置 BGP 路由。把这台路由器放在 AS 101 中，把它配置成与路由器 8 对等。在路由器 4 上创建两个环回接口，分配一个到 118.116.0.0/24 网络，另一个到 117.116.115.0/24 网络；通告这个网络和其他所有其他连接到路由器 8。

6. 配置路由器 8 阻止来自路由器 5 的 117.116.115.0/24 网络广播到路由器 4，不改变路由器 3 或路由器 10，阻止其他 AS 5300 的路由器通告那个网络到任何对等体。

7. 在交换机 16_3550 上配置一条单一的静态路由，指向骨干路由器 5 的 141.200.5.0/24 网络。不要使用默认路由。

10.4.10 第 VIII 部分：服务质量

1. 用一个策略配置路由器 8，根据数据包大小对其帧中继接口限制带宽消耗。使用表 10-1 中显示的数据包大小和带宽百分比。分配带宽限制时遵循思科的接口带宽和队列建议。

表 10-1

路由器 8 策略参数

数据包大小	带宽限制	数据包大小	带宽限制
64~127	28%	512~767	9%
128~255	10%	768~1024	6%
256~511	18%	其他	以 WFQ 排队

10.4.11 第 IX 部分：DLSW+

1. 在路由器 10 的 VLAN A 和骨干路由器 5 的 141.200.5.5 之间配置一个 DLSw TCP 对等体。做完了这个，你应看到 backbone_rtr5 在 DLSW 可达缓存中。

2. 如果到 141.200.5.5 对等体的连接失败了，路由器 5 上另一个到路由器 4 的 VLAN F 的对等体应激活，备份服务器驻留在这里。不能在路由器 5 上使用 **remote-peer** 语句。

3. 当路由器 10 到路由器 4 的对等体激活后，路由器 4 应当通告 NetBIOS 主机备份路由器 4。当到主对等体的连接恢复 3min 后，这个备份对等体应当 down 掉。

10.5 CCIE 练习实验：The Intimidator

设备列表：

- 1 台帧中继交换机：4 个串行端口
- 7 台实验路由器：1 个以太网口，2 个串行接口
- 2 台实验路由器：1 个快速以太网口，2 个串行接口和 1 个语音口（1750s）
- 1 台有扩展的 VLAN 的以太网 35xx 以太网交换机

10.5.1 准备阶段——帧中继交换机和骨干路由器配置

按照图 10-5 中的描述配置帧中继交换机的 PVC。不要计算你自己做这部分实验的时间。

配置骨干路由器 bb-1、bb-2 和 bb-3。范例 10-4 列出了帧中继交换机和骨干路由器 bb-1、bb-2 和 bb-3 的配置。

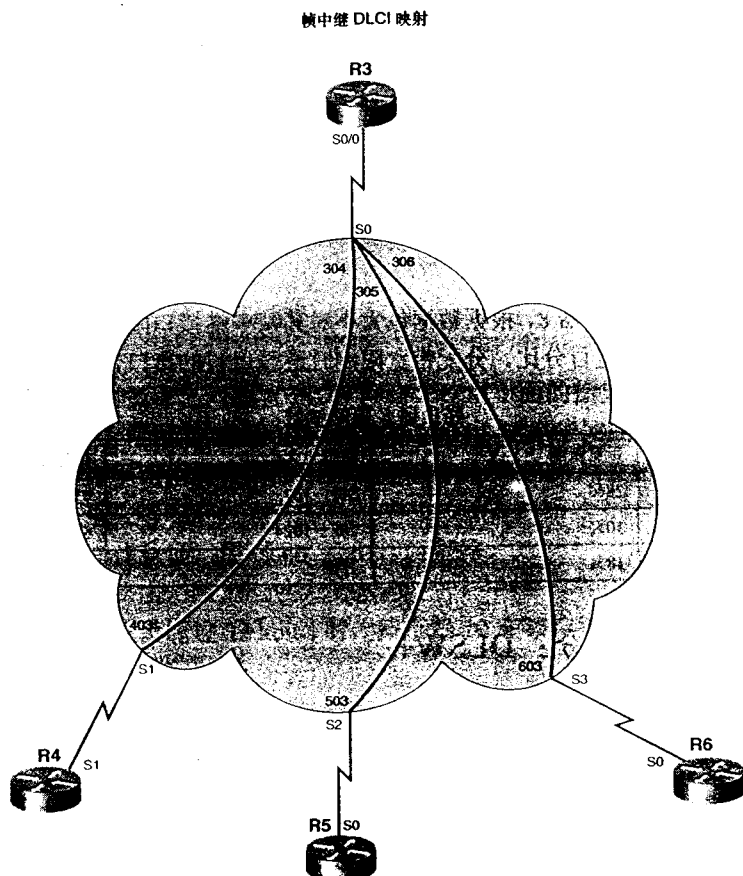


图 10-5 帧中继交换机配置

范例 10-4 帧中继和骨干路由器配置

```
hostname frame_switch
!
ip subnet-zero
!
no ip domain-lookup
!
frame-relay switching
!
interface Serial0
no ip address
encapsulation frame-relay IETF
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 304 interface Serial1 403
frame-relay route 305 interface Serial2 503
frame-relay route 306 interface Serial3 603
!
```

(待续)

```
interface Serial1
no ip address
encapsulation frame-relay IETF
clockrate 1300000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 403 interface Serial0 304
!
interface Serial2
no ip address
encapsulation frame-relay IETF
clockrate 1300000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 503 interface Serial0 305
!
interface Serial3
no ip address
encapsulation frame-relay IETF
logging event dlci-status-change
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 603 interface Serial0 306
!
no cdp run
!
end
----- bb-1 config ----->
hostname bb-1
!
logging buffered 4096 debugging
no logging console
ip subnet-zero
no ip source-route
!
no ip domain lookup
!
interface Loopback10
ip address 177.164.12.1 255.255.252.0
!
interface Loopback20
ip address 177.164.16.1 255.255.252.0
!
interface Loopback30
ip address 2.0.0.1 255.0.0.0
!
interface Loopback40
ip address 8.0.0.1 255.0.0.0
!
interface Loopback50
ip address 16.0.0.1 255.0.0.0
!
interface Ethernet0/0
ip address 55.9.6.1 255.255.255.248
half-duplex
!
interface Serial0/0
ip address 177.164.8.5 255.255.255.252
clockrate 1300000
!
interface Serial0/1
ip address 177.164.8.9 255.255.255.252
```

(待续)


```

!
interface Serial0/2
  no ip address
  shutdown
!
ip classless
no ip http server
!
end
----- bb-2 config ----->
hostname bb-2
no logging console
!
ip subnet-zero
no ip domain lookup
!
interface Loopback10
  ip address 55.9.8.1 255.255.248.0
!
interface Loopback20
  ip address 55.9.16.1 255.255.248.0
!
interface Loopback30
  ip address 2.0.0.2 255.0.0.0
!
interface Loopback40
  ip address 8.0.0.2 255.0.0.0
!
interface Loopback50
  ip address 16.0.0.2 255.0.0.0
!
interface Ethernet0
  ip address 55.9.6.2 255.255.255.248
!
interface Serial0
  ip address 55.9.5.6 255.255.255.252
  clockrate 1300000
!
interface Serial1
  ip address 55.9.5.10 255.255.255.252
!
ip classless
ip http server
!
end
----- bb-3 config ----->
hostname bb-3
!
logging buffered 4096 debugging
no logging console
!
ip subnet-zero
!
no ip domain lookup
!
interface Loopback10
  ip address 168.101.12.1 255.255.252.0
!
interface Loopback20
  ip address 168.101.16.1 255.255.252.0
!

```

(待续)

```
interface Loopback30
 ip address 2.0.0.3 255.0.0.0
!
interface Loopback40
 ip address 8.0.0.3 255.0.0.0
!
interface Loopback50
 ip address 16.0.0.3 255.0.0.0
!
interface FastEthernet0
 ip address 55.9.6.3 255.255.255.248
 speed auto
!
interface Serial0
 ip address 192.168.2.1 255.255.255.252
!
interface Serial1
 ip address 168.101.8.1 255.255.255.252
 clockrate 1300000
!
ip classless
no ip http server
!
call rsvp-sync
!
voice-port 2/0
!
voice-port 2/1
!
dial-peer cor custom
!
!
end
```

下面的实验部分要计时，在所有硬件配置和物理安装后开始。

10.5.2 规则

- 除非特别规定，不允许使用静态路由或浮动静态路由。这个实验允许使用数量非常有限的静态路由。当你可以使用静态路由时，它会清楚地标注出来。
- 严格按照指示去做。注意只在要求的地点和时间传播路由。只能按照说明中的指示使用 PVC。
- 可以使用配置向导和 CD-ROM 作为惟一的参考资料。
- 你有 8.5 小时来完成这部分实验。在这个阶段不要和任何人谈话。
- 建议你在开始前阅读整个实验。

10.5.3 第 I 部分：IP 设置

按照图 10-6 中的描述使用 IP 地址，据此为网络分配地址。注意：此时不是所有的 IP 地址都可以被分配。

- VLAN A 使用 IP 子网 10.12.13.0/24，连接路由器 1、路由器 2 和路由器 3。
- 路由器 5 的 E0 端口使用 IP 地址 10.12.64.5。这个接口在 VLAN C 中。

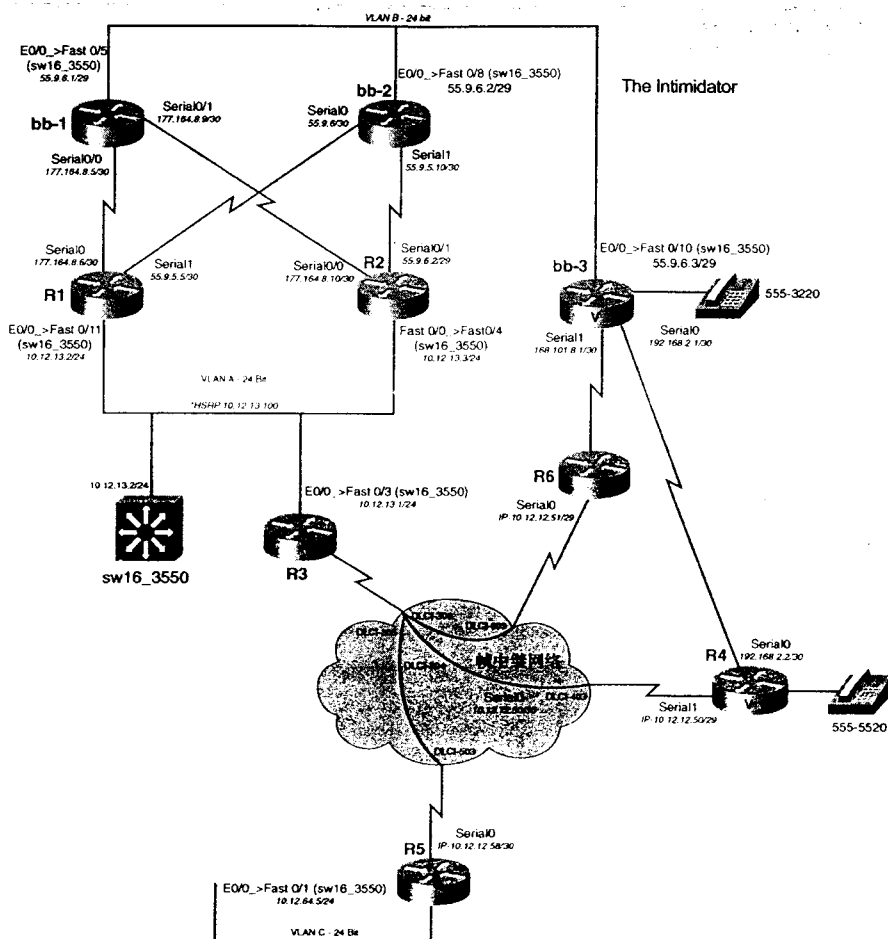


图 10-6 Intimidator 的网络图

- 路由器 4 的 s0 接口使用 IP 地址 192.168.2.2/30 连接骨干路由器 bb-3。
- 路由器 6、路由器 4 和路由器 3 在广域网上共享相同的 IP 子网，应这样配置。路由器 6 的串行接口 0 有一个 IP 地址 10.12.12.51/29，路由器 4 的串行接口有一个 IP 地址 10.12.12.50/29。
- 所有其他的子网和主机地址使用网络 10.12.0.0：
VLAN A、B、C：24-bit 子网

10.5.4 第 II 部分：Catalyst 配置

1. 按照图 10-6 中的描述配置所有的 VLAN。不要使用 VLAN 1。可以使用的有效的 VLAN 范围是 2000~3000。
2. 命名 VTP 域名为 labx。配置 STP，使得如果新的交换机加入到骨干子网 55.9.6.0/29 中，交换机 16_3550 仍然是根交换机。
3. 以 IP 地址 10.12.13.2/24 配置交换机。配置交换机使得它能通过 IP 可达。如果路由器 1、路由器 2 或路由器 3 down 掉，交换机仍然可达。

10.5.5 第 III 部分: OSPF 和帧中继

1. 在路由器 3、路由器 6 和路由器 4 之间配置一个部分网状连接的帧中继网络，使得它们共享相同的 IP 子网。只能使用路由器 3 的子接口。
2. 配置 VLAN A 使其在 OSPF area 0 中。
3. 在路由器 3、路由器 6 和路由器 4 之间配置帧中继网络，使其在 OSPF area 100 中。不能使用 **neighbor** 语句。
4. 配置 area 100 使得路由器 6 和路由器 4 上所有的外部链路状态显示为类型 7。

10.5.6 第 IV 部分: EIGRP 集成

1. 在 VLAN C 上配置 EIGRP，在路由器 3 和路由器 5 之间配置帧中继网络。
2. 配置路由器 5 为 EIGRP 末梢路由器。确保路由器 5 通告 VLAN C。EIGRP 和 OSPF 域之间完全可达。确保路由器 5 能 ping 到 bb-3 的串行接口和路由器 1、路由器 2 的局域网接口。

10.5.7 第 V 部分: HSRP

1. 为 VLAN A 配置 HSRP，使得路由器 1 是主路由器。使用 IP 地址 10.12.13.100 作为共享 IP 地址。
2. 如果路由器 1 的串口失效，路由器 2 将成为主路由器。如果路由器 1 和路由器 2 的串口都失效了，路由器 3 将成为主路由器。

10.5.8 第 VI 部分: BGP

1. 每台路由器应使用一个显式配置的 BGP 路由器 ID。这个 ID 应该是在本地产生的公有地址空间中最小的 IP 地址。例如，bb-1 使用 177.164.8.5 作它的 BGP 路由器 ID。所有的 BGP 发言人应该用最大可用的更新数据包大小。不允许骨干路由器 (bb-1、bb-2 或 bb-3) 用实验路由器 (路由器 1、路由器 2 和路由器 6) 作为过渡。
2. 使用表 10-2 中的信息为骨干路由器配置 BGP。

表 10-2

骨干 BGP 配置

路由器	自治系统编号	远端对等体	通告的网络
bb-1	65	bb-2 的 Ethernet0 接口	177.164.8.0/22 177.164.12.0/22 177.164.16.0/22
		路由器 2 的串口 0/0	177.164.8.0/22 177.164.12.0/22 177.164.16.0/22 2.0.0.0/8 8.0.0.0/8 16.0.0.0/8

续表

路由器	自治系统编号	远端对等体	通告的网络
		路由器 1 的串口 0	177.164.8.0/22 177.164.12.0/22 177.164.16.0/22 2.0.0.0/8 8.0.0.0/8 16.0.0.0/8
bb-2	104	bb-1 的 Ethernet0/0 接口	55.9.0.0/21 55.9.8.0/21 55.9.16.0/21
		路由器 1 的串口 1	55.9.0.0/21 55.9.8.0/21 55.9.16.0/21 2.0.0.0/8 8.0.0.0/8 16.0.0.0/8
		路由器 2 的串口 0/1	55.9.0.0/21 55.9.8.0/21 55.9.16.0/21 2.0.0.0/8 8.0.0.0/8 16.0.0.0/8
bb-3	12	路由器 6 的串口 1	168.101.8.0/22 168.101.12.0/22 168.101.16.0/22 2.0.0.0/8 8.0.0.0/8 16.0.0.0/8
		bb-2 的 Ethernet0 接口	168.101.8.0/22 168.101.12.0/22 168.101.16.0/22
		bb-1 的 Ethernet0/0 接口	168.101.8.0/22 168.101.12.0/22 168.101.16.0/22

3. 无论怎样，网络前缀应该聚合到最小的掩码大小。

4. 路由器 bb-1 和 bb-3 应使用 bb-2 作为过渡网到达彼此。

5. 启用路由器 1 上的 BGP 路由。

6. 把这台路由器放在 AS 10142 中，通告本地连接的 196.200.32.0/20 网络到所有的邻居。

7. 这台路由器应与 bb-1、bb-2 和路由器 2 对等；每个对等体应设置为使用直连 IP 地址作 BGP 对等。

8. 本地发起的路由应收敛到最小的前缀大小。

9. 这台路由器也通告来自路由器 2 和路由器 6 的其他 196.200.x.0 网络；但是，通告这些路由使得它们的外部对等体优选产生这些对等的路由器的路由。这一步不能使用 AS 路径属性。这一步可以为这台路由器增加两条静态路由。

10. 启用路由器 2 的 BGP 路由。

11. 把这台路由器放在 AS 10142 中，通告本地连接的 196.200.48.0/20 网络到所有的邻居。

12. 这台路由器应与 bb-1、bb-2 和路由器 1 对等；每个对等体应设置为使用直连 IP 地址作 BGP 对等。

13. 本地发起的路由应收敛到最小的前缀大小。

14. 这台路由器也通告来自路由器 1 和路由器 6 的其他 196.200.x.0 网络；但是，通告这些路由使得它们的外部对等体优选产生这些对等的路由器的路由。这一步不能使用 AS 路径

属性。这一步可以为这台路由器增加两条静态路由。

15. 配置路由器 1 使它优选从 bb-1 到 2.0.0.0/8 和 8.0.0.0/8 网络的路由，以及从 bb-2 到 16.0.0.0/8 网络的路由。这些设置不能传递到任何路由器，这一步不能使用 AS 路径属性。

16. 启用路由器 6 上的 BGP 路由。

17. 把这台路由器放在 AS 10142 中，通告本地连接的 196.200.64.0/20 网络到所有的邻居。

18. 这台路由器应与 bb-3、路由器 1 和路由器 2 对等；每个对等体应设置为使用直连 IP 地址作 BGP 对等。

19. 本地发起的路由应收敛到最小的前缀大小。

20. 这台路由器也通告来自路由器 1 和路由器 2 的其他 196.200.x.0 网络；但是，通告这些路由使得它们的外部对等体优选产生这些对等的路由器的路由。这一步不能使用 AS 路径属性。这一步可以为这台路由器增加两条静态路由。

21. 在路由器 1 和路由器 6 以及路由器 2 和路由器 6 之间配置 BGP 路由。配置这些路由器使用二层 VPN 接口到达彼此本地产生的 BGP 网络。

10.5.9 第 VII 部分: Voice

1. 在这两台路由器之间使用 192.168.2.0/30 网络配置 Voice over IP，如下。

2. 为路由器 4 的端口 2/0 配一部电话。为这部电话分配 555-5520 电话号码。

3. 为 bb-3 的端口 2/0 配一部电话。这部电话使用 555-3220 电话号码。

4. 每一个语音连接使用 g723r63 编码。

5. 配置路由器 5，使得当拿起电话机时，它自动呼叫 bb-3。

6. 配置 bb-3，使得无论何时拨打 555-5520 或 811，它都呼叫路由器 5。

10.5.10 第 VIII 部分: 服务质量

1. 配置路由器 1、路由器 2 和路由器 6 上的每个输出骨干连接，使得它们在拥塞时基于 IP 优先级值丢弃部分流量。

2. 用以下策略配置路由器 3：

- TCP 80 端口的所有流量应限制到 Ethernet0/0 接口带宽的 20%。任何 HTTP 流量应当采用 WRED 优先丢弃。
- 所有其他流量应当采用加权公平队列。
- 配置 RSVP 用于两个 Voice over IP 的呼叫者之间的所有语音呼叫；RSVP 只为两个呼叫者提供足够的带宽，EF-PHB 用于所有呼叫。
- 配置每个 Voice over IP 会话请求保证速率服务质量，并用 EF-PHB 控制所有进入呼叫。

10.5.11 第 IX 部分: DLSW+

1. 在路由器 3 的 VLAN A 和路由器 5 的 VLAN C 之间配置一个 DLSw+对等体。配置对等体，使其支持 RFC 1490，有可靠的传输和本地确认。

2. 配置对等体使其只允许 SNA 流量穿过 DLSw+连接。

10.6 CCIE 练习实验：Enchilada II

设备列表：

- 1 台帧中继交换机：5 个串行接口。
- 具有 2 个 BRI 端口的 ISDN 模拟器/交换机。
- 具有 2 个 ATM 接口的 ATM 交换机。
- 2 台实验路由器：1 个以太网口，1 个串行接口。
- 1 台实验路由器：1 个快速以太网口，1 个串行接口，1 个 ATM 口，1 个 ISDN BRI 口。
- 1 台实验路由器：1 个以太网口，1 个 ISDN BRI 口，1 个串行接口。
- 1 台实验路由器：2 个以太网口，1 个串行接口。
- 1 台实验路由器：1 个 ATM 口。
- 1 台实验路由器：1 个以太网口。
- 1 台装有 EMI 软件的以太网 3550 交换机，2 个光口或交叉电缆，用于网络互连。
- 1 台以太网 35xx 以太网交换机。

10.6.1 准备阶段——帧中继交换机、骨干路由器和 ATM 的配置

按照图 10-7 中的描述配置帧中继交换机的 PVC。不要计算你自己做这部分实验的时间。

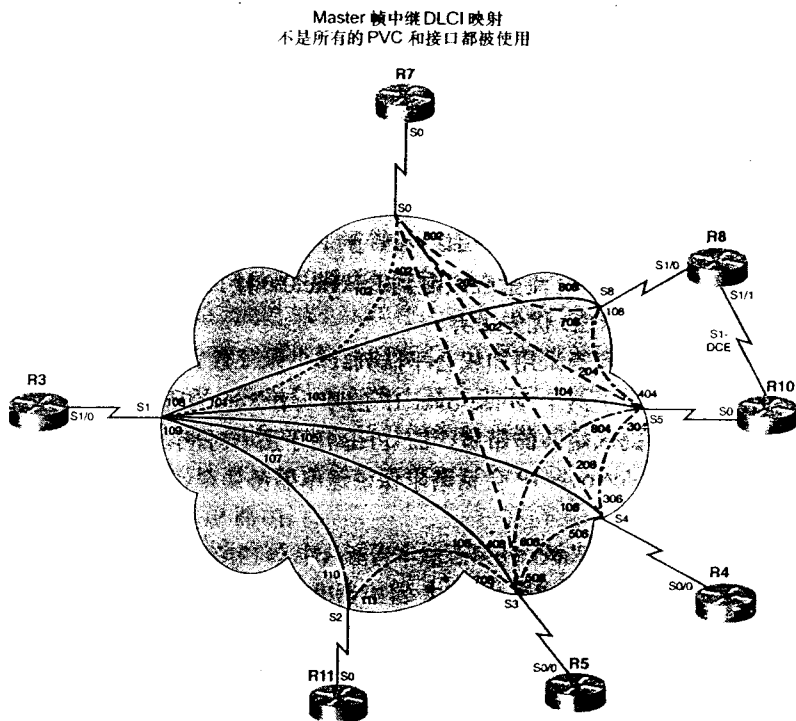


图 10-7 帧中继交换机配置

不是图中的所有 PVC 都要使用。同样，配置骨干路由器 5、路由器 11 和 ATM 交换机。范例 10-5 列出了帧中继和 ATM 交换机的配置。范例 10-6 列出了骨干路由器 5 和路由器 11 的配置。

范例 10-5 帧中继和 ATM 交换机配置

```
hostname frame_switch
!
frame-relay switching
!
interface Ethernet0
  no ip address
  shutdown
!
interface Serial0
  no ip address
  encapsulation frame-relay
  no fair-queue
  clockrate 2000000
  frame-relay intf-type dce
  frame-relay route 102 interface Serial1 101
  frame-relay route 202 interface Serial5 204
  frame-relay route 302 interface Serial4 206
  frame-relay route 402 interface Serial3 408
  frame-relay route 802 interface Serial8 708
!
interface Serial1
  no ip address
  encapsulation frame-relay
  clockrate 2000000
  frame-relay intf-type dce
  frame-relay route 101 interface Serial0 102
  frame-relay route 103 interface Serial5 104
  frame-relay route 105 interface Serial4 106
  frame-relay route 107 interface Serial3 108
  frame-relay route 108 interface Serial8 808
  frame-relay route 109 interface Serial2 110
!
interface Serial2
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 110 interface Serial1 109
  frame-relay route 111 interface Serial3 708
!
interface Serial3
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 108 interface Serial1 107
  frame-relay route 408 interface Serial0 402
  frame-relay route 508 interface Serial4 506
  frame-relay route 608 interface Serial5 804
  frame-relay route 708 interface Serial2 111
!
interface Serial4
  no ip address
  encapsulation frame-relay
```

(待续)


```

clockrate 64000
frame-relay intf-type dce
frame-relay route 106 interface Serial1 105
frame-relay route 206 interface Serial0 302
frame-relay route 306 interface Serial5 304
frame-relay route 506 interface Serial3 508
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 104 interface Serial1 103
frame-relay route 204 interface Serial0 202
frame-relay route 304 interface Serial4 306
frame-relay route 404 interface Serial8 110
frame-relay route 804 interface Serial3 608
!
interface Serial6
no ip address
!
interface Serial7
no ip address
!
interface Serial8
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 108 interface Serial5 404
frame-relay route 708 interface Serial0 802
frame-relay route 808 interface Serial1 108
!
interface Serial9
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
no ip classless
!
end
----- ATM ----->

hostname ls1010
!
!
atm address 47.0091.8100.0000.0061.705b.4001.0061.705b.4001.00
!
interface ATM0/0/0
no keepalive
no atm auto-configuration
no atm address-registration
no atm ilmi-enable
no atm ilmi-lecs-implied
!
interface ATM0/0/1
no keepalive
no atm auto-configuration
no atm address-registration

```

(待续)

```

no atm ilmi-enable
no atm ilmi-lecs-implied
atm pvc 1 101 interface ATM0/0/0 1 102
!
interface ATM0/0/2
no keepalive
!
interface ATM0/0/3
no keepalive
!
!
interface ATM1/1/3
no keepalive
!
interface ATM2/0/0
no ip address
no keepalive
atm maxvp-number 0
!
interface Ethernet2/0/0
no ip address
!
no ip classless
logging buffered
!
line con 0
line aux 0
line vty 0 4
login
!
end
----- backbone routers ----->

```

范例 10-6 骨干路由器 5 和骨干路由器 11 的配置

```

hostname backbone_router_r5
!
ip tcp path-mtu-discovery
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
interface Loopback0
ip address 201.201.5.5 255.255.255.0
!
interface Loopback4
ip address 4.4.4.4 255.0.0.0
!
interface Loopback6
ip address 6.6.6.6 255.0.0.0
!
interface Loopback12
ip address 12.1.1.1 255.0.0.0
!
interface Loopback55
ip address 5.5.5.5 255.255.0.0

```

(待续)

```
!
interface Ethernet0/0
 ip address 10.1.2.5 255.255.255.0
!
interface Serial0/0
 ip address 10.1.1.5 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-point
 no ip mroute-cache
 frame-relay interface-dlci 108
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 2003
 network 10.1.0.0 0.0.255.255 area 500
 area 500 stub
!
router bgp 65001
 no synchronization
 bgp router-id 10.1.1.5
 bgp confederation identifier 10001
 bgp confederation peers 65002
 network 4.0.0.0
 network 5.5.0.0 mask 255.255.0.0
 network 6.0.0.0
 network 12.0.0.0
 neighbor AS65001 peer-group
 neighbor AS65001 remote-as 65001
 neighbor AS65001 route-reflector-client
 neighbor AS65001 update-source Serial0/0
 neighbor AS65001 next-hop-self
 neighbor 10.1.1.3 peer-group AS65001
 no auto-summary
!
ip classless
!
logging buffered 4096 debugging
!
end

----->
hostname backbone_router_r11
!
ip subnet-zero
ip tcp path-mtu-discovery
!
isdn voice-call-failure 0
!
interface Loopback20
 ip address 192.200.16.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback21
 ip address 192.200.17.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback22
 ip address 192.200.18.11 255.255.255.0
 no ip directed-broadcast
```

(待续)

```

!
interface Loopback23
 ip address 192.200.19.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback24
 ip address 192.200.20.11 255.255.255.0
 no ip directed-broadcast
!
interface Loopback88
 ip address 88.8.8.8 255.255.0.0
 no ip directed-broadcast
!
interface Ethernet0
 description to fast 0/11 on sw15_3550
 ip address 192.168.2.11 255.255.255.0
 no ip directed-broadcast
 ip ospf message-digest-key 2 md5 trustno1
!
interface Serial0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 shutdown
!
interface Serial1
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 2003
 area 0 authentication message-digest
 network 192.168.2.11 0.0.0.0 area 0
 network 192.200.0.0 0.0.255.255 area 200
!
router bgp 96
 bgp router-id 192.168.2.11
 bgp cluster-id 2177372427
 network 88.8.0.0 mask 255.255.0.0
 neighbor 192.168.2.1 remote-as 10001
 neighbor 192.168.2.1 password :)router
 neighbor 192.168.2.1 update-source Ethernet0
!
ip classless
 no ip http server
!
end

```

下面的实验部分要计时，在所有硬件配置和物理安装后开始。

10.6.2 规则

- 除非特别规定，不允许使用静态路由或浮动静态路由。
- 严格按照指示去做。注意只在要求的地点和时间传播路由。只能按照说明中的指示使用 PVC。
- 可以使用配置向导和 CD-ROM 作为惟一的参考资料。
- 你有 8.5 小时来完成这部分实验。在这个阶段不要和任何人谈话。

- 建议你在开始前阅读整个实验。
- 画一个正确的精确的网络图。

10.6.3 第 I 部分：IP 设置

1. 按照图 10-8 中的描述使用 IP 地址，据此为网络分配地址。注意：此时不是所有的 IP 地址都可以被分配。

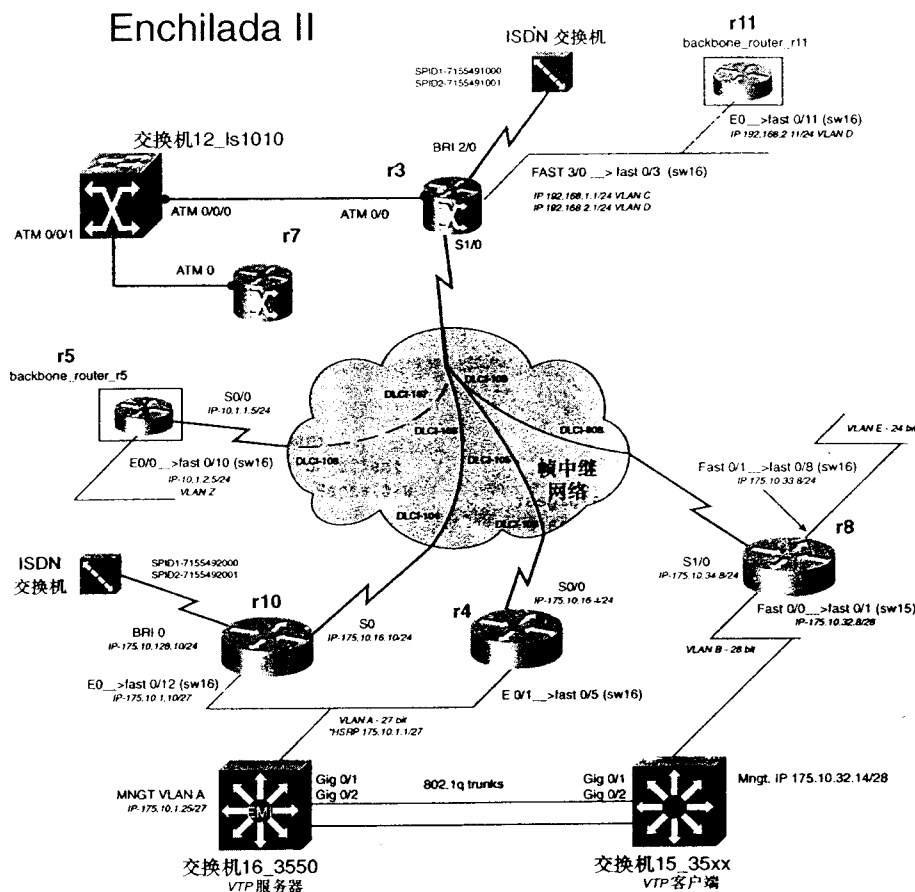


图 10-8 Enchilada II 的网络图

2. 在路由器 3 的 Fast 3/0 接口使用 IP 子网 192.168.1.0/24 和 192.168.2.0/24。子网 192.168.1.0/24 使用 VLAN C，子网 192.168.2.0/24 使用 VLAN D。
3. 路由器 10 的 E0 接口使用 IP 地址 175.10.1.10/27。这个接口同路由器 4 的 e0/1 接口一起在 VLAN A 中。
4. 路由器 8 的 Fast 0/1 接口在 VLAN E 中，Fast 0/0 接口在 VLAN B 中。Fast 0/1 接口使用 IP 地址 175.10.33.8/24，Fast 0/0 接口使用 175.10.32.8/28。
5. 所有其他子网和主机地址使用网络 175.10.0.0:
 - VLAN: 27-bit 子网;

- VLAN B: 28-bit 子网;
- VLAN C、D、E、Z: 24-bit 子网。

10.6.4 第 II 部分: Catalyst 配置

1. 使用 Gig 0/1 和 Gig 0/2 接口在交换机 15_35xx 和交换机 16_3550 之间配置 802.1Q 吉比特骨干 (在这个实验中你可以用 100BASE-T 接口代替)。不要把 IP 地址设在吉比特接口上。
2. 按照图 10-8 中的描述配置 VLAN。不要使用 VLAN 1。
3. 把交换机 16_3550 配置为 VTP 服务器, 交换机 15_35xx 为客户端。使用 PSV2 作为 VTP 域名, 用密码 cisco 验证 VTP。
4. 配置交换机 16_3550 使其支持 802.1w RSTP 和 802.1s MSTP。创建 3 个 STP 范例; 使用下面的指导:
 - 例 0: VLAN 1, STP 优先级 8192;
 - 例 1: VLAN 100~200, STP 优先级 4096;
 - 例 2: VLAN 2~99, 201~4094, STP 优先级 16834;
 - 确保 802.1w 和 802.1d 都工作在交换机 15_35xx 上。也就是说, 前面标注的 VLAN 的 VLAN 优先级应该与交换机 15_35xx 上的相同;
 - 确保连接到主机的交换机上的端口配置了 802.1w。
5. 使用 IP 地址 175.10.1.25/27 可以连接到交换机 16_3550, 使用 IP 地址 175.10.32.14/28 可以连接到交换机 15_35xx。在交换机 16_3550 上不能配置默认或静态路由。

10.6.5 第 III 部分: EIGRP、三层交换和帧中继

1. 在路由器 3、路由器 10 和路由器 4 之间配置一个部分网状连接的帧中继网。只能使用路由器 3 上的子接口。
2. 在路由器 3、路由器 10 和路由器 4 之间的帧中继网上配置 EIGRP。使用自治系统 ID 2003。
3. 在路由器 10 和路由器 4 之间的 VLAN A 和交换机 16_3550 上配置 EIGRP。启用交换机 16_3550 的三层交换来完成此任务。

10.6.6 第 IV 部分: RIP、OSPF 集成

1. 在路由器 3 和骨干路由器 11 之间配置 OSPF。配置 VLAN C 在 OSPF area 100 中, VLAN D 在 OSPF area 0 中。
用类型 II 认证方式做 OSPF area 0 认证。
2. 当你连接到骨干路由器 11 时, 你应当收到下面的 OSPF 路由: 192.200.16.0/24, 192.200.17.0/24, 192.200.18.0/24, 192.200.19.0/24 和 192.200.20.0/24。确保所有的路由器能够到达这些路由, 包括 RIP 和 EIGRP 域。
3. 在帧中继网络上路由器 3 和骨干路由器 5 之间配置 OSPF。配置帧中继网络使其在

area 500 中。area 500 应该配置为一个末梢区域。

4. 在路由器 3 和路由器 8 之间配置 RIPv2。VLAN E 和 VLAN B 也应运行 RIPv2。帧中继链路路上的 RIP 更新使用 MD5 认证。

10.6.7 第 V 部分：路由过滤和 HSRP

1. 路由器 10 和路由器 4 应运行 EIGRP 外部路由 192.200.16.0/24, 192.200.17.0/24, 192.200.18.0/24, 192.200.19.0/24 和 192.200.20.0/24。路由器 10 应该只传播奇数的 192.200.0.0 子网到交换机 16_3550。路由器 4 应该只传播偶数的 192.200.0.0 子网到交换机 16_3550。

2. 在路由器 10、路由器 4 和交换机 16_3550 之间配置 HSRP。使用 175.10.1.1/27 作 HSRP 的地址。

在所有设备之间做 HSRP 更新认证。使用密码 trustno1。

路由器 10 应该是默认的主路由器。如果路由器 10 的串口失败，路由器 4 应成为主路由器。如果路由器 4 的串口失败，路由器 10 的串口宕掉，交换机 16_3550 应成为 HSRP 的主路由器。

10.6.8 第 VI 部分：ISDN

在路由器 10 和路由器 3 之间配置 ISDN 网络。使用下面的指导：

- 路由器 10 使用 IP 地址 175.10.128.10/24。这个子网应该在 EIGRP 域；
- 拨号器不应由于路由协议的原因而一直保持 up 状态。配置路由器 10 仅当到 192.168.2.0/24 和 192.168.1.0/24 的路由失败时才建立呼叫；
- 使用 CHAP 作认证；使用 cisco_isdn 作为密码；
- 不要使用静态路由；路由应该是动态的；
- 5min 的空闲时间后链路断掉。

10.6.9 第 VII 部分：ATM

1. 配置一个 ATM PVC 从路由器 3 的 atm0/0 端口到路由器 7 的 atm0 端口；这里使用子接口。
2. 使用 ATM 封装方法，它最适合于突发数据流量。
3. 配置 ATM 电路，支持突发延时承受 VBR 流量；这条电路应该被配置为使用 8 个 T1 不变的信元速率，峰值信元速率支持全部接口带宽。
4. 使用 62.1.8.0 网络，30 比特子网掩码。

10.6.10 第 VIII 部分：BGP

1. 所有 BGP 路由器应该使用静态分配的 BGP 路由器 ID 彼此对等；BGP 路由更新应使用最大可能的数据包大小。除非另外说明，不能使用路由反射器来完成这个实验的任务。BGP 只用来通告环回网络；不要配置 BGP 来通告任何 10 网络。当路由器在相同的自治系统中有不止一个对等体时，使用一个对等体组简化配置。这部分完成后，所有 BGP 路由器上的所有

BGP 路由可达。增加和通告下面表 10-3 中的网络。

表 10-3 实验 4 BGP 网络

通告 路由器	网络	通告 路由器	网络
路由器 3	62.1.8.0/24 3.0.0.0/8	路由器 7	52.1.1.0/24 54.1.0.0/16 62.1.8.0/30
路由器 4	32.1.1.0/24 32.2.2.0/24	路由器 10	22.1.1.0/24 24.24.24.24/24
路由器 5	4.0.0.0/8 5.5.0.0/16 6.0.0.0/8 12.0.0.0/8		

2. 启用路由器 3、路由器 5 和路由器 7 上的 BGP 路由。在 AS 65001 中配置所有这些路由器彼此对等；这些路由器也应该属于父 AS 10001。

在 AS 96 中配置路由器 3 与路由器 11 对等；这些路由器应使用 BGP 认证，使用密码“:) router”。

路由器 3 也应该在 ATM 网络上与路由器 7 对等，在帧中继网络上与路由器 5 对等；允许在路由器 3、路由器 5 和路由器 7 上使用一条路由反射器语句。

路由器 7 应该能够到达所有路由器 3 能够到达的网络；允许在路由器 7 上使用一条默认路由。

3. 在路由器 4 和路由器 10 上配置 BGP 路由；把这些路由器放在 AS 65002 中；这些路由器也应该属于父 AS 10001。

在 AS 65001 中路由器 4 也应该与路由器 3 对等。AS 65002 中的所有路由器应该收到并且能够到达所有路由器 3 发送的 BGP 路由，反之亦然。

10.6.11 第 IX 部分：DLSw+

1. 在路由器 4 的 VLAN A 和路由器 8 的 VLAN B 之间配置一个 DLSw TCP 对等体。来自路由器 4 VLAN A 的探测器和 DLSw 流量只允许到路由器 8 的 VLAN B 上。

2. 在路由器 3 的 VLAN D 和路由器 8 的 VLAN E 之间配置一个 DLSw TCP 对等体。只有来自 VLAN D 的探测器和 DLSw 流量能够到达路由器 8 的 VLAN E。

3. 来自这两个对等体的 DLSw 流量彼此不能互相影响。

10.6.12 第 X 部分：NAT

配置 NAT，使得当访问任何内部的实验设备时，VLAN B 的所有用户共享一个单一的 IP 地址。举例来讲，如果交换机 15_35xx 发一个 ping 到路由器 3，它应该被翻译。

10.6.13 第 XI 部分：组播路由

1. 在路由器 3、路由器 10 和路由器 7 上配置组播路由。

2. 使用集合地址 175.10.16.3。路由器 10 和路由器 3 应该都能 ping 路由器 7 的 ATM 接口上的组播地址 224.0.10.10。

10.7 CCIE 练习实验：Kobayashi Maru

设备列表

- 1 台帧中继交换机：4 个串行接口
- 具有 2 个 BRI 端口的 ISDN 模拟器/交换机
- 具有 2 个 ATM 接口的 ATM 交换机
- 1 台实验路由器：1 个以太网接口，1 个串行接口
- 1 台实验路由器：1 个以太网接口，1 个串行接口和 1 个 FXS 语音端口
- 1 台实验路由器：1 个快速以太网口，1 个串行接口，1 个 ATM 口和 1 个 ISDN BRI
- 1 台实验路由器：1 个以太网口，1 个 ISDN BRI 和 2 个串行接口
- 1 台实验路由器：2 个以太网口和 1 个 FXS 语音端口
- 1 台实验路由器：2 个以太网口和 1 个串行接口
- 1 台实验路由器：1 个 ATM 口
- 1 台装有 EMI 软件的以太网 3550 交换机，1 个光口或交叉电缆，用于网络互连
- 1 台以太网 35xx 以太网交换机

10.7.1 准备阶段——帧中继交换机、骨干路由器和 ATM 配置

按照图 10-9 中的描述配置帧中继交换机的 PVC。不要计算你自己做这部分实验的时间。

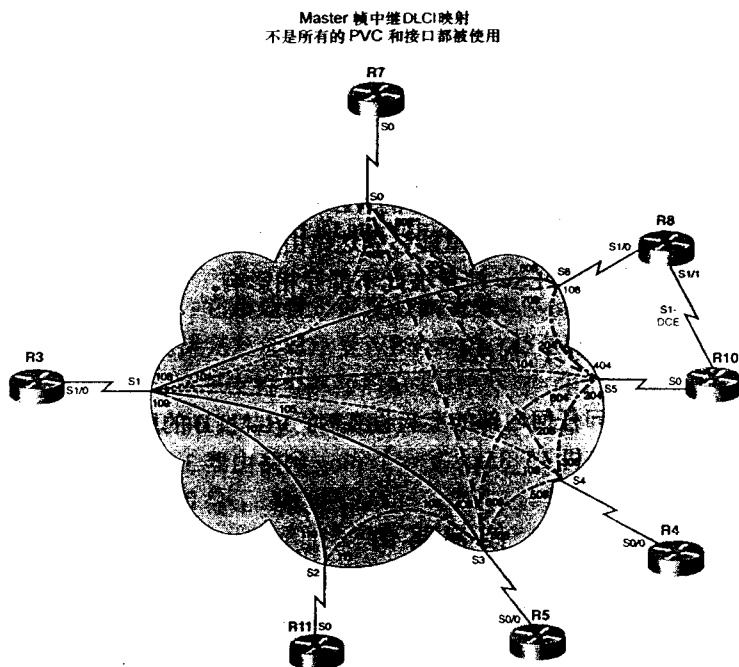


图 10-9 帧中继交换机配置

不是图中所有的 PVC 都要使用。范例 10-7 列出了帧中继和 ATM 交换机的配置。

范例 10-7 帧中继和 ATM 交换机配置

```
hostname frame_switch
!
frame-relay switching
!
interface Ethernet0
  no ip address
  shutdown
!
interface Serial0
  no ip address
  encapsulation frame-relay
  no fair-queue
  clockrate 2000000
  frame-relay intf-type dce
  frame-relay route 102 interface Serial1 101
  frame-relay route 202 interface Serial5 204
  frame-relay route 302 interface Serial4 206
  frame-relay route 402 interface Serial3 408
  frame-relay route 802 interface Serial8 708
!
interface Serial1
  no ip address
  encapsulation frame-relay
  clockrate 2000000
  frame-relay intf-type dce
  frame-relay route 101 interface Serial0 102
  frame-relay route 103 interface Serial5 104
  frame-relay route 105 interface Serial4 106
  frame-relay route 107 interface Serial3 108
  frame-relay route 108 interface Serial8 808
  frame-relay route 109 interface Serial2 110
!
interface Serial2
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 110 interface Serial1 109
  frame-relay route 111 interface Serial3 708
!
interface Serial3
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 108 interface Serial1 107
  frame-relay route 408 interface Serial0 402
  frame-relay route 508 interface Serial4 506
  frame-relay route 608 interface Serial5 804
  frame-relay route 708 interface Serial2 111
!
interface Serial4
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay intf-type dce
  frame-relay route 106 interface Serial1 105
```

(待续)

```

frame-relay route 206 interface Serial0 302
frame-relay route 306 interface Serial5 304
frame-relay route 506 interface Serial3 508
!
interface Serial5
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 104 interface Serial1 103
frame-relay route 204 interface Serial0 202
frame-relay route 304 interface Serial4 306
frame-relay route 404 interface Serial8 110
frame-relay route 804 interface Serial3 608
!
interface Serial6
no ip address
!
interface Serial7
no ip address
!
interface Serial8
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 108 interface Serial5 404
frame-relay route 708 interface Serial0 802
frame-relay route 808 interface Serial1 108
!
interface Serial9
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
no ip classless
!
end
----- ATM Switch ----->
hostname ls1010
!
atm address 47.0091.8100.0000.0061.705b.4001.0061.705b.4001.00
!
interface ATM0/0/0
no keepalive
no atm auto-configuration
no atm address-registration
no atm ilmi-enable
no atm ilmi-lecs-implied
!
interface ATM0/0/1
no keepalive
no atm auto-configuration
no atm address-registration
no atm ilmi-enable
no atm ilmi-lecs-implied
atm pvc 1 101 interface ATM0/0/0 1 102
atm pvc 3 103 interface ATM0/0/0 7 107
!

```

(待续)

```
interface ATM0/0/2
no keepalive
end
```

下面的实验部分要计时，在所有硬件配置和物理安装后开始。

10.7.2 规则

- 除非特别规定，不允许使用静态路由或浮动静态路由。
- 严格按照指示去做。注意只在要求的地点和时间传播路由。只能按照说明中的指示使用 PVC。
- 可以使用配置向导和 CD-ROM 作为惟一的参考资料。
- 你有 8.5 小时来完成这部分实验。在这个阶段不要和任何人谈话。
- 建议你在开始前阅读整个实验。

10.7.3 第 I 部分：IP 设置

1. 按照图 10-10 中的描述使用 IP 地址，据此为网络分配地址。注意：此时不是所有的 IP 地址都可以被分配。

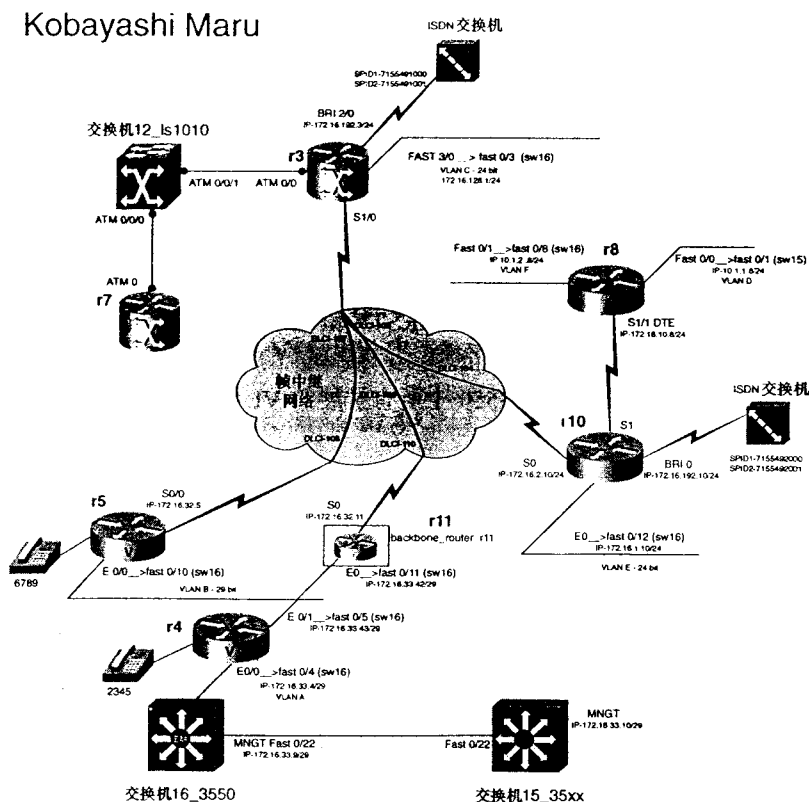


图 10-10 Kobayashi Maru 网络图

2. 路由器 3 的 Fast 3/0 接口使用 IP 地址 172.16.128.1。子网 172.16.128.0/24 在 VLAN C 中。
3. 路由器 11 的 E0 接口使用 IP 地址 172.16.33.42/29。这个接口同路由器 4 的 e0/1 接口和路由器 5 的 e0/0 接口一起在 VLAN B 中。
4. 路由器 8 的 Fast 0/1 接口在 VLAN F 中，Fast 0/0 接口在 VLAN D 中。Fast 0/1 接口使用 IP 地址 10.1.2.8/24，Fast 0/0 接口使用 10.1.1.8/24。
5. 路由器 10 的 e0 接口在 VLAN E 中；它使用 IP 地址 172.16.1.10/24。
6. 路由器 4 的 e0/0 接口在 VLAN A 中，使用 IP 地址 172.16.33.4/29。
7. 所有其他的子网和主机地址使用网络 172.16.0.0：
 - VLAN A：29-bit 子网
 - VLAN B：29-bit 子网
 - VLAN C、D、E、F：24-bit 子网

10.7.4 第 II 部分：Catalyst 配置

1. 按照图 10-8 中的描述配置 VLAN。不要使用 VLAN 1：
 - VLAN A = VLAN 2034
 - VLAN B = VLAN 2033
 - VLAN C = VLAN 1026
 - VLAN D = VLAN 10（在交换机 15_35xx 上）
 - VLAN E = VLAN 1025
 - VLAN F = VLAN 10
2. 通过背对背电缆把 Catalyst 交换机连起来。确保两台交换机是可达的，使用下面的地址：交换机 16_3550 = 172.16.33.9/29 和交换机 15_35xx = 172.16.33.10/29。不要配置 802.1Q 或 ISL 骨干。交换机 16_3550 上不能配置默认或静态路由。
3. 配置交换机 16_3550 和交换机 15_35xx 使用 ccie_psv2 作为 VTP 域名。选择最适合你的网络设计需要的 VTP 模式。
4. 配置交换机 16_3550 使其支持 802.1w RSTP 和 802.1s MSTP。配置所有的主机端口支持 RSTP。
5. 配置 MSTP，使得交换机 16_3550 上所有扩展的 VLAN 将是生成树的根。标准范围的 VLAN 应使用默认的 STP 值。
6. 在交换机 16_3550 的 VLAN 2034 上配置 MAC 地址 0001.0001.aaaa。

10.7.5 第 III 部分：OSPF、EIGRP、三层交换和帧中继

1. 在路由器 3、路由器 5 和路由器 11 之间配置一个部分网状连接的帧中继网络。你可以在你想要的任何地方使用子接口。
2. 在路由器 3、路由器 5 和路由器 11 之间的帧中继网络上配置 OSPF。帧中继网络在 OSPF area 0 中。配置路由器 3 的 VLAN C 在 area 51 中。
3. 在路由器 5、路由器 11 和路由器 4 之间配置 VLAN B，使其在 OSPF area 100 中。
4. 改变路由器 5 的 S0/0 接口的 OSPF hello 计时器为 60s。

5. 在路由器 4 和交换机 16_3550 之间的 VLAN A 上配置 EIGRP。使用自治系统 ID 2003。启用交换机 16_3550 的 3 层交换来完成这个任务。

6. EIGRP 发起的路由应该作为 OSPF 外部类型 1 路由出现，在所有的 OSPF 路由器上带有标记 4。

7. 确保 EIGRP 和 OSPF 域的所有 IP 可达。交换机 15_35xx 应能 ping 到 VLAN C，反之亦然。

10.7.6 第 IV 部分：IS-IS 和 RIP 集成

1. 在帧中继网络上路由器 3 和路由器 10 之间配置 IS-IS。通过 IS-IS 通告 VLAN E。

2. 在路由器 10 和路由器 8 之间配置串行链路。配置链路使其支持基于 Lempel-Ziv (LZ) 的压缩运算法则。

3. 在路由器 10 和路由器 8 之间配置 RIP。不要通过 RIP 通告 VLAN D 和 VLAN F。RIP 不应使用广播路由更新。

4. 把 RIP 和 IS-IS 完全集成到现有的 OSPF/EIGRP 域。确保所有路由域之间完全可达。

5. 在路由器 3 上，以原始管理距离 0、标签 3333 标记所有重分发的路由。以原始管理距离 115、标签 3 标记重分发路由，以原始管理距离 1、标签 777 标记路由。

10.7.7 第 V 部分：NAT 和 DHCP

1. 使用下面的指导在路由器 8 上配置 NAT：

- VLAN D, 10.1.1.0/24 上的用户共享 5 个 IP 地址 (172.16.16.2 到 172.16.16.6)。
- 路由器 8 的 Fast 0/0 IP 地址, 10.1.1.8, 始终被翻译为 172.16.16.100。
- VLAN F 上的用户使用 PAT。

2. 确保 VLAN D 和 VLAN F 上的用户能够 ping 到交换机 16_3550 和交换机 15_35xx，并据此被翻译。

3. 配置路由器 3 的 VLAN C 的用户使用 DHCP。服务器应该通告 172.16.128.1 作为默认网关。在 DHCP 地址池中保留 VLAN C 的 4 个主机地址为以后使用。

10.7.8 第 VI 部分：组播路由和 NTP

1. 配置路由器 8 作为 NTP 服务器，交换机 16_3550 接收 NTP 更新。当交换机 16_3550 与服务器同步时，它的等级是 6。

2. 在路由器 3、路由器 4 和路由器 5 上配置组播路由。使用稀疏模式，在路由器 3 的 Fast 3/0 接口配置组播地址 224.0.10.3。

3. 配置路由器 4 和路由器 5，使得它们能够 ping 到组播地址 224.0.10.3。

10.7.9 第 VII 部分：ISDN

在路由器 10 和路由器 3 之间配置 ISDN 网络。使用下面的指导：

- 路由器 10 上使用 IP 地址 172.16.192.10/24。

- 拨号器不应由于路由协议的原因而一直保持 up 状态。
- 配置路由器 10 仅当到 172.16.128.0/24 的路由/VLAN C 失败时才建立呼叫。两个 B 信道应立即启用。
- 使用 PPP CHAP 认证；使用 cisco_isdn 作为密码。
- 可以使用静态路由。
- 5min 空闲时间后链路断开。

10.7.10 第 VIII 部分：ATM

1. 配置一个 ATM PVC 从路由器 3 的 atm0/0 端口到路由器 7 的 atm0 端口；在这里使用子接口。
2. 使用 ATM 封装方式，它最适合于突发数据流量。
3. 配置 ATM 电路，以一个未指定的比特率来支持突发延时承受流量；配置这条电路的峰值信元速率支持全部接口带宽。
4. 使用 10.55.1.8 网络，30 比特子网掩码。

10.7.11 第 IX 部分：BGP

1. 所有 BGP 路由器应该使用静态分配的 BGP 路由器 ID 彼此对等；BGP 路由更新应使用最大可能的数据包大小。BGP 只用来通告环回网络；不要配置 BGP 来通告任何 10 网络。当路由器在相同的自治系统中有不止一个对等体时，使用一个对等体组简化配置。这部分完成后，所有 BGP 路由器上的所有 BGP 路由可达。增加和通告表 10-4 中显示的网络。

表 10-4

实验 5 BGP 网络

通告路由器	网络	通告路由器	网络
路由器 3	198.201.5.0/24 109.201.11.0/24 10.55.1.8/30	路由器 5	36.101.11.0/24 37.101.12.0/24
路由器 4	164.8.8.0/24 164.10.10/24	路由器 7	208.161.8.0/24 208.164.8.0/24

2. 在路由器 3 和路由器 7 上配置 BGP；把这两台路由器放在 AS 97 中。
3. 配置路由器 7 与路由器 3 通过 ATM 接口对等。配置路由器 7，使得 208.164.8.0/24 网络不在 AS 97 以外传播；在路由器 3 上允许一个配置行。
4. 在 AS 148 中帧中继网络上，路由器 3 应该与路由器 5 和路由器 11 对等。
5. 在路由器 5 和路由器 11 上配置 BGP。
6. 路由器 5 应该与路由器 3、路由器 4 和路由器 11 对等。
7. 路由器 11 应该与路由器 3、路由器 4 和路由器 5 对等。
8. 配置路由器 11，使来自路由器 4 的路由在广播给其他路由器时作为低优先级被选路由。
9. 在路由器 4 上配置 BGP；把这台路由器放在 AS 65 中，配置它使其在以太网口与路

由器 5 和路由器 11 对等。

10.7.12 第 X 部分：语音

1. 在路由器 5 和路由器 4 之间配置 Voice over IP。
2. 路由器 4 的 1/0/0 语音端口使用 2345 电话号码。这个实验要求你使用 164.8.8.1 IP 地址作语音呼叫。
3. 路由器 5 的 1/0/0 端口使用 6789 电话号码，对所有的语音呼叫必须使用 36.101.11.1 IP 地址。
4. 当拨打 411 电话号码时，来自路由器 4 的主叫也能够到达路由器 5；在这里路由器 4 上只允许使用一条命令。

10.7.13 第 XI 部分：DLSW+

1. 在路由器 10 的 VLAN E 和路由器 5 的 VLAN B 之间配置一个 DLSw Fast Sequence Transport 对等体。配置 DLSw 使得只有 NetBIOS 流量能够穿过对等体。

第七部分

附 录

附录 A 思科 IOS 软件的限制和约束

附录 B RFC

附录 C 参考书目

附录 D IP 前缀列表

- 如果数据包到达输入接口没有越界，修改组播边界访问列表不能阻止数据包被在访问列表修改前已经存在的任何组播路由所转发。然而，违犯组播边界访问列表更新版本的新组播路由不会被学习，在更新的访问列表的违犯中的任何组播路由如果已经超时，也不会再学习。

更新组播边界之后，工作区将使用 **clear ip mroute** 特权模式命令删除违犯更新的边界的所有现存的组播路由。（错误代码：CSCdr79083）

- 当收到一个带有循环冗余校验（CRC）错误的数据包时，每个数据包的每个 DSCP 计数器（从 DSCP 0）会增加。正常网络不应该有 CRC 错误的数据包（错误代码：CSCdr85898）。
- **mac-address** 接口配置命令不能正确地给接口分配一个 MAC 地址。Catalyst 3550 交换机不支持这个命令（错误代码：CSCds11328）。
- 如果你配置动态主机配置协议（DHCP）服务器从一个地址池给交换机分配地址，网络上的两台设备可能有相同的 IP 地址。地址池的地址被临时分配给一台设备，当地址不用了，就被送回地址池。如果在交换机收到这样一个地址之后你保存配置文件，共享地址就被保存了，重启之后交换机不再尝试访问 DHCP 服务器来接收一个新的 IP 地址。结果，两台设备可能有相同的 IP 地址。
工作区确保你配置通过交换机的硬件地址绑定到每台交换机的 DHCP 服务器保留租约（错误代码：CSCds55220）。
- **show ip mroute count** 特权模式命令可能显示不正确的数据包数。在某个瞬时状态（例如，在路由学习过程中，当组播数据流仅仅转发到 CPU，CPU 正在把这条路由编制程序到硬件时），组播流数据包数可能被计算 2 次。在这个瞬时状态不要相信计数器（错误代码：CSCds61396）。
- 当修改吉比特以太网口的链路速率从 1000Mbit/s 到 100Mbit/s 时，这里有一点可能性是这个端口会停止发送数据包。如果出现这个问题，使用 **shutdown** 和 **no shutdown** 接口配置命令，关闭端口再重新启用（错误代码：CSCds84279）。
- 在 IP 组播路由和回退桥接（fallback bridging）中，某些硬件特性被用来为输出方向骨干端口的不同 VLAN 复制数据包。如果输入速率是线性速率，那么输出接口不能复制这个速率（因为数据包的复制）。结果，某些复制的数据包被丢弃（错误代码：CSCdt06418）。
- 当使用 **no interface port-channel global** 配置命令删除一个以太通道组时，端口组中的端口变为管理 down 状态。
当删除一个以太通道组时，在属于端口组的接口上输入 **no shutdown** 接口配置命令，把它们置回在线状态（错误代码：CSCdt10825）。
- **show interface interface-id** 特权 EXEC 模式命令显示的输出中，*Output Buffer Failures* 项显示复制前数据包丢包数，然而，*Packets Output* 项显示复制后成功发送的数据包。要确定实际丢弃的帧，可以用输出缓冲器的失败数乘以复制组播数据的 VLAN 数（错误代码：CSCdt26928）。
- 按照服务质量（QoS）分类来匹配服务质量策略映射中区分服务编码点（DSCP）的值和服务等级（CoS）的值的互联网组管理协议（IGMP）数据包可能只修改 DSCP 特性，保留 CoS 值为 0（错误代码：CSCdt27705）。

- 如果你用 **wrr-queue threshold** 接口配置命令分配两个尾丢弃阈值百分比为 100%，用 **show mls qos interface statistics** 特权模式命令显示这个接口的服务质量信息，那么丢包数统计总是 0，即使超过了阈值。为显示数据包丢包总数，使用 **show controllers ethernet-controllers interface-id** 特权模式 EXEC 命令。在显示中，丢帧数包括超过尾丢弃阈值时被丢弃的帧（错误代码：CSCdt29703）。
- 交换虚接口（SVI）端口的开放最短路径优先（OSPF）路径代价和内部网关协议（IGRP）度量是不正确的。可以用 **bandwidth** 接口配置命令手工配置 SVI 带宽。当 SVI 作为一个输出接口时，改变接口带宽就改变路由的度量（错误代码：CSCdt29806）。
- Catalyst 3550 的冷启用和热启用 trap 不是一直发送。（错误代码：CSCdt33779）
- 物理接口上的远程监控（RMON）采集功能在以太信道和 SVI 上不支持（错误代码：CSCdt36101）。
- 当 IGMP 监听禁用时，组播路由器信息显示在 **show ip igmp snooping mrouter** 特权模式 EXEC 命令中。组播 VLAN 注册（MVR）和 IGMP 监听使用相同的命令显示组播路由器信息。在这个范例中，MVR 被启用，IGMP snooping 禁用（错误代码：CSCdt48002）。
- 当一个 VLAN 接口被禁用并使用 **shutdown** 和 **no shutdown** 接口配置命令重新启用多次时，接口可能不会遵循 **no shutdown** 命令重启。要重启这个接口，再次输入一个 **shutdown** 和 **no shutdown** 命令序列（错误代码：CSCdt54435）。
- 当配置 **ip pim spt-threshold infinity** 接口配置命令时，需要所有指定组的源使用共享树，而不使用源树。然而，交换机不能自动开始使用共享树。虽然没有出现连接问题，但是交换机继续对已经装在组播路由表中的组播组条目使用最短路径树。可以输入 **clear ip mroute *** 特权模式 EXEC 命令来强制变成共享树（错误代码：CSCdt60412）。
- 如果在交换机上配置的组播路由数比交换机所能支持的大，它可能会用完可用的内存，这将导致重启。这是一个平台-无关编码的限制。
工作区不能配置交换机以超过所支持的组播路由最大值运行。你可以使用 **show sdm prefer** 和 **show sdm prefer routing** 特权模式 EXEC 命令来看一下当前 SDM 模板和路由模板的近似最大值配置指导（错误代码：CSCdt63354）。
- 配置太多的组播组可能导致内存非常低的情形，并导致软件控制数据结构不同步，引起不可预知的转发行为。内存资源只能通过执行 **clear ip mroute** 特权模式 EXEC 命令恢复。为防止出现这种情况，在交换机上不要配置多于推荐的组播路由（错误代码：CSCdt63480）。
- **bridge bridge-group protocol** 全局配置命令不支持 **dec** 关键字。如果两台 Catalyst 3550 交换机通过一个接口互连，这个接口配置了 IP 路由和回退桥接，并且用 **bridge bridge-group protocol dec** 命令配置了桥接组，那么这两台交换机好像担当了生成树根。因此，生成树循环可能未被发现（错误代码：CSCdt63589）。
- 当你在一台 Catalyst 3550 和一台 Catalyst 1900 交换机之间配置一个以太通道时，以太通道中的一些 Catalyst 3550 链路可能 down 掉，但是只要通道中的一条链路保持 up 状态，就会维持连通性。

- 工作区用 **channel-group channel-group-number** 模式接口配置命令来停用两台设备上的端口聚合协议 (PAgP)。这两台设备间的 PAgP 协商是不可靠的 (错误代码: CSCdt78727)。

- 当交换机运行相等-代价路由，并且它必须学习多于它所能支持的单播路由时，CPU 可能用完内存，交换机会出故障。

工作区在建议文档中，有限支持 (错误代码: CSCdt79172)。

- 服务质量软件访问控制列表 (ACL) 的行为与服务质量硬件访问控制列表不同。在 Catalyst 3550 交换机上，当服务质量硬件重写数据包的 DSCP 时，这个项的重写发生在运行在 CPU 上的软件检查数据包之前，CPU 只看到新的值，而不是原始的 DSCP 值。当安全硬件访问控制列表匹配输入数据包时，匹配使用原始的 DSCP 值。对输出安全访问控制列表，安全访问控制列表硬件应匹配最终可能会修改的、由服务质量硬件设置的 DSCP 值。在某些环境里，与硬件安全访问控制列表的匹配防止了服务质量重写 DSCP，导致 CPU 使用原始的 DSCP 值。

如果一个安全访问控制列表应用在软件中 (因为访问控制列表不适合硬件，数据包被发送到 CPU 进行检查)，匹配可能使用新的 DSCP 值作为服务质量硬件最终的值，无论访问控制列表应用在输入还是输出中。当数据包被访问控制列表记入日志时，这个问题也会影响匹配是否被 CPU 记入日志，即使访问控制列表适合硬件，允许或拒绝过滤在硬件中完成。

为避免这些问题，无论何时交换机重写任何数据包的 DSCP 为一个与原始 DSCP 不同的值，安全访问控制列表不应检查任何它们的访问控制元素 (ACE) 中的 DSCP 值，无论访问控制列表是被应用到一个 IP 访问组还是一个 VLAN 映射。这个限制不适合用于服务质量分级映射的访问控制列表。

如果交换机没有配置为重写任何数据包的 DSCP 值，这样不改变 DSCP，而 IP 访问组或 VLAN 映射的访问控制列表中的 DSCP 是会改变的，因为当数据包被交换机处理时，DSCP 不改变。

IP 数据包的 DSCP 项包含两个项，它们是最初指定的优先级和 ToS (服务类型)。

与 DSCP 相关的语句同样适用于 IP 优先级或 IP ToS。 (错误代码: CSCdt94355)

- 使用 **speed nonegotiate** 接口配置命令停用吉比特接口转换器 (GBIC) 接口的自动协商，可能导致接口显示物理链路是 up 状态，即使当它没有连接的时候 (错误代码: CSCdv29722)。
- 如果你配置一个骨干接口，动态骨干协议 (DTP) 为非协商模式，并使用 **switchport trunk encapsulation** 接口配置命令将封装类型从内部交换链路 (ISL) 改变为 802.1Q，那么端口变为一个访问口，不再是骨干 (错误代码: CSCdv46715)。
- 在 Catalyst 3550-24 交换机的早期版本，如果交换机的一个 10/100BASE-TX 端口通过一个 ISL 骨干以 100 Mbit/s 连接到 Catalyst 2820 或 Catalyst 1900 交换机，不能建立双向通信。Catalyst 2820 或 Catalyst 1900 交换机把 Catalyst 3550-24 交换机看作一个思科发现协议 (CDP) 的邻居，但是 Catalyst 3550-24 交换机不能认出 Catalyst 2820 或 Catalyst 1900 交换机。在这些交换机上，你不能在 Catalyst 3550-24 和一台 Catalyst 2820 或 Catalyst 1900 交换机之间使用 ISL 骨干。配置链路作为一条访问链路而不是骨干链路。这个问题在 Catalyst 3550-24 交换机的硬件中用装配号 73-5700-08 或更新的主板已

经解决了。为确定交换机的主板级别，输入 **show version** 特权模式 EXEC。母板信息出现在输出显示的最后（错误代码：CSCdv68158）。

- 当启用了 IGMP 过滤，你使用 **ip igmp** 预置全局配置命令创建一个 IGMP 过滤器，保留组播地址不能被过滤。因为 IGMP 过滤只使用 3 层地址来过滤 IGMP 报告，并且由于 3 层组播地址和以太网组播地址之间的映射，始终允许保留组（224.0.0.x）通过交换机。另外，别名组可以漏过交换机。例如，如果允许一个用户接收来自组 225.1.2.3 的报告，但是不能接收来自组 230.1.2.3 的报告，别名将导致用户接收到来自 230.1.2.3 的报告。保留地址的别名意味着所有 y.0.0.x 格式的组允许通过（错误代码：CSCdv73626）。如果你使用 **ip igmp max-groups** 接口配置命令，将一个接口的 IGMP 组最大编号设置为 0，端口仍然接收到来自保留组播组（224.0.0.x）和它们的 2 层别名（y.0.0.x）的组报告（错误代码：CSCdv79832）。

- 交换机在执行 **no snmp-server host** 全局配置命令时可以重新装载。这是一种很少见的情况，可能发生在如果启用了 SNMP trap 或信息，正当它从配置中被删除时，SNMP 代理尝试发送一个 trap 到主机，以及如果主机（或到达主机的网关）IP 地址没有被地址解析协议（ARP）解析的情况下。

所做的这些是为了确保在从 SNMP 配置中将目标主机或到达目标主机的下一跳网关移去之前，它们都存在于 ARP 缓存中（如，通过发一个 **ping** 命令）。作为选择，可以在从 SNMP 配置中删除任何主机之前，禁用所有 SNMP trap 和信息（错误代码：CSCdw44266）。

- 当你访问 CISCO-STACK-MIB 端口表时，映射可能被一个来自交换机给出的映射中断。表中的对象由 2 个数字索引：portModuleIndex 和 portIndex。portModuleIndex 允许的取值是 1~16。因为 0 是不允许的值，所以 1 表示模块 0。

工作区使用 1 来表示模块 0（错误代码：CSCdw71848）。

- 如果运行多生成树协议（MSTP）的 Catalyst 3550 交换机的一个端口连接到另一台属于不同的多生成树（MST）域的交换机，当你使用 **clear spanning-tree detected-protocols interface interface-id** 特权模式 EXEC 命令启用协议迁移过程时，Catalyst 3550 端口不能被识别为一个边界端口。这个问题仅出现在根桥上，当清除根桥时，边界端口不显示，因为指定的端口不能接收任何桥协议数据单元（BPDU），除非发生拓扑改变。这是预期的行为。

工作区使用 **spanning-tree mode pvst** 全局配置命令桥，配置 Catalyst 3550 交换机采用每个 VLAN 独立的生成树（PVST），然后使用 **spanning-tree mode mst** 全局配置命令把它变成 MSTP（错误代码：CSCdx10808）。

- 如果应用访问控制列表到一个附加了服务质量策略映像的接口，并且配置访问控制列表使得数据包被 CPU 转发，或者如果配置的访问控制列表不能适合三重内容可寻址内存（TCAM），那么所有从这个接口收到的数据包被转发到 CPU。因为转发到 CPU 的流量不能被配置在接口上的策略管辖，所以这个流量不能精确地限速到配置的策略速率。

当为一个接口配置了服务质量限速时，工作区将配置应用的访问控制列表，使得数据包不被 CPU 转发或减小访问控制列表中的 ACE 数，使得它能适合 TCAM（错误代码：CSCdx30485）。

- 当流量监管时 Catalyst 3550 交换机不顾及 Preamble 和 Inter Frame Gap (IFG)，对大量的小数据帧，在 Preamble 和 IFG 的比率对于帧的大小更重要的情况下，这将导致轻微地不准确的策略速率。在不同大小混合帧情况下，这不是一个问题。
- 当你正在退出 VLAN 配置模式（通过输入 **vlan database** 特权模式 EXEC 命令）时，如果交换机由于任何原因出故障了，有一个很小的可能性是 VLAN 数据库崩溃。交换机重置之后，你可以在控制台上看到这些消息：

```
%SW_VLAN-4-VTP_INVALID_DATABASE_DATA: VLAN manager received bad data of type
device
type: value 0 from vtp database
$SW_VLAN-3-VTP_PROTOCOL_ERROR: VTP protocol code internal error
```

工作区将用 **delete flash: vlan.dat** 特权模式 EXEC 命令删除崩溃的 VLAN 数据库，然后使用 **reload** 特权模式 EXEC 命令重新装载交换机。（错误代码：CSCdx19540）

- 当一台思科 RPS 300 冗余电源系统为一台交换机提供电源时，交换机电源供给恢复后，RPS 300 继续提供电源直到按了 RPS 模式按钮。至今，一些交换机的重启依赖于交换机内部电源供给恢复运行的快慢（错误代码：CSCdx81023）。
- 在交换机中插入 GigaStack 吉比特接口转换器（GBIC）模块，导致 CPU 利用率的增加（错误代码：CSCdx90515）。
- 热备路由协议（HSRP）不支持在不同的 VPN 路由和转发（VRF）表中重叠地址的配置（错误代码：CSCdy14520）。
- 当配置 1000 个 VLAN 和多于 40 个骨干端口时，生成树模式从 MSTP 变为 PVST，反之亦然，在控制台上出现这个消息：

```
%ETHCNR-3-RA_ALLOC_ERROR: RAM Access write pool I/O memory allocation failure
```

没有工作区。无论如何，建议你使用 **reload** 特权模式 EXEC 命令重新装载交换机。为避免这个问题，以更少的 VLAN 和骨干端口配置系统，或者使用 **switchport trunk allowed vlan** 接口配置命令减少每个骨干端口的活动 VLAN 数（错误代码：CSCdx20106）。

A.2 集群的限制和约束

这些限制适用于集群配置：

- 当从 **cluster active** 命令交换机转换成 **standby** 命令交换机时，Catalyst 1900、Catalyst 2820 和 Catalyst 2900 4-MB 集群成员交换机可能失去它们的集群配置。必须手工将这些交换机添加回集群（错误代码：CSCds32517，CSCds44529，CSCds55711，CSCds55787，CSCdt70872）。
- 当一台 Catalyst 2900 XL 或 Catalyst 3500 XL **cluster** 命令交换机连接到一台 Catalyst 3550 交换机时，如果它不是集群的成员，命令交换机就不能越过 Catalyst 3550 交换机发现任何集群候选者。必须添加 Catalyst 3550 交换机到集群，然后你能看到与它连接的所有集群候选者（错误代码：CSCdt09918）。
- 当启用了 **clustering**，不要配置超过 59 字节的 SNMP 团体字符串，否则 **clustering**

SNMP 可能不正常工作（错误代码：CSCdt39616）。

- 如果 **active** 命令交换机和 **standby** 命令交换机同时发生故障，集群不能自动重新创建。即使有第三台从命令交换机，它也不可能再创建所有的集群成员，因为它不可能有全部最新的集群配置信息。如果 **active** 和 **standby** 命令交换机同时发生故障，必须手工重新创建集群（错误代码：CSCdt43501）。

A.3 集群管理组限制和约束

这些限制适用于 Cluster Management Suite (CMS) 配置：

- 在集群命令交换机、成员交换机或候选交换机上，包括逗号的主机名和域名系统 (DNS) 服务器名称能导致 CMS 运转异常。可以通过不在主机名或 DNS 名称中使用逗号来避免这个接口的不稳定性。在 CMS 的 IP 管理窗口的 IP 配置表中，输入多个 DNS 名称时也不输入逗号。
- 在标准访问控制列表中，包含 **host** 关键字的 ACE 先于所有其他的 ACE。可以用一个限制在标准访问控制列表中重新安置 ACE：带有 **any** 关键字或通配符掩码的 ACE 不能够先于带有 **host** 关键字的 ACE。
- 在一台 Solaris 机器上，如果拓扑图打开几个小时，CMS 性能会降低。原因可能是一个内存的漏洞。
工作区将关闭浏览器，重新打开它，再启用 CMS。（错误代码：CSCds29230）
- 如果你正在打印一张包含很多设备的拓扑图或前面板图，并且正在运行 Solaris 2.6 JDK1.2.2，你可能得到“Out of Memory”错误消息。
工作区将关闭浏览器，重新打开它，再启用 CMS。在你执行任何其他工作之前，提出你需要打印的图，在 CMS 菜单中点击 Print。（错误代码：CSCds80920）
- 如果运行 CMS 的 PC 内存很低，CMS 连续运行 2~3 天，PC 会用完内存。
工作区将重新启用 CMS（错误代码：CSCdv88724）。
- 当已经配置好一个 VLAN 或 VLAN 范围，并且为 SPAN 会话指定了 VLAN 过滤器，当前的会话配置将被新的输入项改写。虽然 CLI 在已有的输入项后面追加新的输入项，但是 CMS 重新创建整个会话，改写当前的输入项，并且每个输入项只提供一个单一的 VLAN 过滤器。
工作区将使用 CLI；在一个 Switched Port Analyzer (SPAN) 会话中，这是指定多个 VLAN 用于过滤的惟一方法（错误代码：CSCdw93904）。

A.4 重要注释

A.4.1 思科 IOS 软件注释

这些注释适用于思科 IOS 软件配置：

- 如果你在一台有 VLAN 映射或配置了输入路由器访问控制列表的交换机的物理接口上配置一个端口访问控制列表，或者如果你在一台配置了端口访问控制列表的交换机上配置一个 VLAN 映射或输入路由器访问控制列表，会产生一个“CONFLICT”消息，但是配置被接受了。端口访问控制列表对该端口的作用优先于路由器访问控制列表或应用在这个端口所属的 VLAN 的 VLAN 映射的作用。
结果是那个物理端口上收到的数据包将会基于端口访问控制列表作用被允许或拒绝，而不关心在路由器访问控制列表或 VLAN 映射中的任何 **permit** 或者 **deny** 语句，尽管这个 VLAN 中的其他物理端口接收的数据包仍然基于路由器访问控制列表或应用于 VLAN 的 VLAN 映射被允许或拒绝。如果端口访问控制列表应用于一个骨干端口，它不考虑应用于骨干端口上所有 VLAN 的任何其他输入访问控制列表。
- Catalyst 3550 交换机流量的默认系统最大传输单元 (MTU) 是 1500 字节。802.1Q 隧道特性以 4 字节帧大小增加。因此，当你配置 802.1Q 隧道时，必须配置在 802.1Q 网络中的所有交换机能够处理最大帧，通过将交换机系统 MTU 的大小增加到至少 1504 字节。使用 **system mtu** 全局配置命令配置系统 MTU 大小。
- 从思科 IOS 软件版本 12.1(8)EA1 开始，配置流量抑制(以前是通过使用 **switchport broadcast**、**switchport multicast** 和 **switchport unicast** 接口配置命令配置的)，使用 **storm-control {broadcast | multicast | unicast} level level [level]** 接口配置命令。关于这些命令更多的信息，参考 *Catalyst 3550 多层交换命令参考*。
- 当你使用 GigaStack GBIC 正在配置一个 Catalyst 3550 交换机层叠堆栈，并且需要在堆栈中包含不止一个 VLAN 时，使用 **switchport mode trunk** 接口配置命令确保配置所有的 GigaStack GBIC 接口作为骨干端口，使用 **switchport encapsulation {isl | dot1q}** 接口配置命令保证使用相同的封装方式。关于这些命令更多的信息，参考 *Catalyst 3550 多层交换命令参考*。
- 如果不能安全地插入 1000BASE-T GBIC (WS-G5482)，**show interface** 特权模式 EXEC 命令输入后交换机可能不能识别它或者显示一个不正确的介质类型。如果出现这个问题，删除并重新插入 GBIC。
- 从思科 IOS 软件版本 12.1(11)EA1 开始，**mac address-table aging-time** 命令取代 **mac-address-table aging-time** 命令（带有连字号）。**mac-address-table aging-time** 命令（带有连字号）在将来的版本中将不再使用。
- 从思科 IOS 软件版本 12.1(11)EA1 开始，**vtp** 特权模式 EXEC 命令关键字可以用在 **vtp** 全局配置命令中。**vtp** 特权模式 EXEC 命令在将来的版本中将不再使用。

A.4.2 集群注释

此注释适用于集群配置：

- **cluster setup** 特权模式 EXEC 命令和 **standby mac-address** 接口配置命令因为不能正确运行，已经从 CLI 和文件中删除。

A.4.3 CMS 注释

这些注释适用于 CMS 配置：

- 如果你在 Windows 2000 上使用 CMS 会话改变配置的同时，通过 CLI 方式改变了启用口令，那么你通过 CMS 改变的配置可能不会生效。必须重新启用 CMS 并在提示时输入新的密码。除了 Windows 2000 以外的平台当它被修改时提示你输入新的密码。
- CMS 不显示通过 CLI 创建的服务质量分级，如果这些分级有多重 **match** 语句。使用 CMS 时，不能创建匹配不止一个 **match** 语句的分级。CMS 不显示有这种分级的策略。
- 如果你使用 Internet Explorer version 5.5 并选择一个地址最后带有非标准端口的 URL（例如，**www.add.com: 84**），那么必须输入 **http://** 作为 URL 前缀。否则，不能启用 CMS。
- 在一个访问控制列表中，可以改变带有 **host** 关键字的 ACE 的顺序。但是，因为这样的 ACE 彼此独立，这个改变对访问控制列表过滤流量的方法没有影响。
- 如果你使用 Netscape 浏览器查看 CMS GUI，当 CMS 初始化时你调整浏览器窗口的大小，CMS 不调整大小来适应窗口。
当 CMS 不忙的时候再调整浏览器窗口。
- 如果计算机的临时目录内存溢出，CMS 就不能启用。出现这个问题是由于 Java 插件程序 1.2.2 版本的一个 bug。只要运行 CMS，插件程序就在目录中创建临时文件，目录最终用完插件空间。
工作区从临时目录中删除所有的 **jar_cache*.tmp** 文件。不同的操作系统的目录路径不同：
 - Solaris: **/var/tmp**
 - Windows NT 和 Windows 2000: **\TEMP**
 - Windows 95 和 98: **\Windows\Temp**

A.4.4 CMS 中的只读模式

CMS 提供对配置选项的两级访问。如果你的特权级别是 15，你有对 CMS 的读写访问权。如果你的特权级别是 1~14，你有对 CMS 的只读访问权。在只读模式中，一些数据不显示，当交换机运行这些软件版本时会出现一个错误信息：

- Catalyst 2900 XL 或 Catalyst 3500 XL 成员交换机，运行 12.0 (5) WC2 或更早版本。
- Catalyst 2950 成员交换机，运行 12.0 (5) WC2 或更早版本。
- Catalyst 3550 成员交换机，运行 12.1 (6) EA1 或更早版本。

在前面板图或拓扑图中，CMS 不显示错误信息。在前面板图中，如果交换机正在运行前面列出的软件版本之一，设备 LED 不出现。在拓扑图中，如果成员是长距离传输以太网 (LRE) 交换机，连接到交换机的用户驻地设备 (CPE) 就不会出现。带宽和链路图表也不在这些图中显示。

为了看交换机的信息，你需要升级成员交换机软件。关于升级交换机软件，看“下载软件”部分。

A.4.5 版本 12.1 (11) EA1 中不支持的 CLI 命令

这一部分列出了在 Catalyst 3550 交换机提示符下输入问号 (?) 能够显示，但是在这个版本中不支持的一些 CLI 命令，或者因为它们未经测试，或者因为 Catalyst 3550 硬件限制。这不是全部列表。按照软件特性和命令模式列出不支持的命令。

1. 访问控制列表：不支持特权模式 EXEC 命令

```
access-enable [host] [timeout minutes]
access-template [access-list-number | name] [dynamic-name] [source] [destination]
[timeout minutes]
clear access-template [access-list-number | name] [dynamic-name] [source] [destination]
```

2. 地址解析协议：不支持全局配置命令

```
arp ip-address hardware-address smds
arp ip-address hardware-address srp-a
arp ip-address hardware-address srp-b
```

3. 地址解析协议：不支持接口配置命令

```
arp probe
ip probe proxy
```

4. 回退桥接：不支持特权模式 EXEC 命令

```
clear bridge [bridge-group] multicast [router-ports | groups | counts] [group-address]
[interface-unit] [counts]
clear vlan statistics
show bridge [bridge-group] circuit-group [circuit-group] [-mac-address]
[dst-mac-address]
show bridge [bridge-group] multicast [router-ports | groups] [group-address]
show bridge vlan
show interfaces crb
show interfaces {ethernet | fastethernet} [interface | slot/port] irb
show subscriber-policy range
```

5. 回退桥接：不支持全局配置命令

```
bridge bridge-group bitswap_l3_addresses
bridge bridge-group bridge ip
bridge bridge-group circuit-group circuit-group pause milliseconds
bridge bridge-group circuit-group circuit-group source-based
bridge cmf
bridge crb
bridge bridge-group domain domain-name
bridge irb
bridge bridge-group mac-address-table limit number
bridge bridge-group multicast-source
bridge bridge-group route protocol
bridge bridge-group subscriber policy policy
subscriber-policy policy [[no | default] packet [permit | deny]]
```

6. 回退桥接：不支持接口配置命令

```
bridge-group bridge-group cbus-bridging
bridge-group bridge-group circuit-group circuit-number
bridge-group bridge-group input-address-list access-list-number
bridge-group bridge-group input-lat-service-deny group-list
```

```
bridge-group bridge-group input-lat-service-permit group-list
bridge-group bridge-group input-lsap-list access-list-number
bridge-group bridge-group input-pattern-list access-list-number
bridge-group bridge-group input-type-list access-list-number
bridge-group bridge-group lat-compression
bridge-group bridge-group output-address-list access-list-number
bridge-group bridge-group output-lat-service-deny group-list
bridge-group bridge-group output-lat-service-permit group-list
bridge-group bridge-group output-lsap-list access-list-number
bridge-group bridge-group output-pattern-list access-list-number
bridge-group bridge-group output-type-list access-list-number
bridge-group bridge-group sse
bridge-group bridge-group subscriber-loop-control
bridge-group bridge-group subscriber-trunk
bridge bridge-group lat-service-filtering
frame-relay map bridge dlci broadcast
interface bvi bridge-group
x25 map bridge x.121-address broadcast [options-keywords]
```

7. 热备路由协议：不支持全局配置命令

```
interface Async
interface BVI
interface Dialer
interface Group-Async
interface Lex
interface Multilink
interface Virtual-Template
interface Virtual-Tokenring
```

8. 热备路由协议：不支持接口配置命令

```
mtu
standby mac-refresh seconds
standby use-bia
```

9. 热备路由协议：接口配置命令

```
switchport broadcast level
switchport multicast level
switchport unicast level
```

注意：这些命令在思科 IOS 软件版本 12.1 (8) EA1 中被 **storm-control {broadcast | multicast | unicast} level level [level]** 接口配置命令取代。

10. IP 组播路由：不支持特权模式 EXEC 命令

```
debug ip packet
```

显示交换机 CPU 接收的数据包。不显示硬件交换的数据包。

```
debug ip mcache
```

影响交换机 CPU 接收的数据包。不显示硬件交换的数据包。

```
debug ip mpacket [detail] [access-list-number [group-name-or-address]]
```

只影响交换机 CPU 接收的数据包。因为大部分组播数据包是硬件交换的，只有当你知道路由将转发数据包到 CPU 时才使用这个命令。

```
debug ip pim atm
show frame-relay ip rtp header-compression [interface type number]
show ip mcache
```

不涉及 CPU 交换，可以使用这个命令，但是组播数据包信息不显示。

```
show ip mpacket
```

支持但是仅对交换机 CPU 接收的数据包有用。如果路由是硬件交换的，命令没有影响，因为 CPU 不接收数据包，不能显示它。

```
show ip pim vc [group-address | name] [type number]
show ip rtp header-compression [type number] [detail]
```

显示 PIM 和 RTP 报头压缩信息。

11. IP 组播路由：不支持全局配置命令

```
ip pim accept-rp {address | auto-rp} [group-access-list-number]
ip pim message-interval seconds
```

12. IP 组播路由：不支持接口配置命令

```
frame-relay ip rtp header-compression [active | passive]
frame-relay map ip ip-address dlcil [broadcast] compress
frame-relay map ip ip-address dlcil rtp header-compression [active | passive]
ip igmp helper-address ip-address
ip multicast helper-map {group-address | broadcast} {broadcast-address | multicast-address} extended-access-list-number
ip multicast rate-limit {in | out} [video | whiteboard] [group-list access-list] [source-list access-list] kbps
ip multicast use-functional
ip pim minimum-vc-rate pps
ip pim multipoint-signalling
ip pim nbma-mode
ip pim vc-count number
ip rtp compression-connections number
ip rtp header-compression [passive]
```

13. IP 单播路由：不支持特权模式 EXEC 或用户模式 EXEC 命令

```
clear ip accounting [checkpoint]
clear ip bgp {* | address | peer-group-name} soft [in | out]
clear ip bgp dampening
clear ip bgp address flap-statistics
clear ip bgp prefix-list
show cef [drop | not-cef-switched]
show ip accounting [checkpoint] [output-packets | access-violations]
show ip bgp dampened-paths
show ip bgp flap-statistics
show ip bgp inconsistent-as
show ip bgp regexp regular expression
show ip prefix-list regular expression
```

14. IP 单播路由：不支持全局配置命令

```
ip accounting-list ip-address wildcard
ip as-path access-list
ip accounting-transits count
ip cef accounting [per-prefix] [non-recursive]
ip cef traffic-statistics [load-interval seconds] [update-rate seconds]
ip flow-aggregation
ip flow-cache
ip flow-export
ip gratuitous-arps
ip local
ip prefix-list
ip reflexive-list
router bgp
router ebgp
router isis
```

```
router iso-igrp
router mobile
router odr
router static
```

15. IP 单播路由：不支持接口配置命令

```
ip accounting
ip load-sharing [per-packet]
ip mtu bytes
ip route-cache
ip verify
ip unnumbered type number
```

所有 ip security 命令。

16. 不支持 BGP 路由器配置命令

注意：这些边界网关协议（BGP）命令未对 Catalyst 3550 进行过测试，并且不被思科 IOS 软件版本 12.1（11）EAI 的交换机支持。这不是一个全部列表。

```
address-family vpnv4
address-family ipv4 [multicast | unicast]
default-information originate
neighbor advertise-map
neighbor advertisement-interval
neighbor allowas-in
neighbor default-originate
neighbor description
neighbor distribute-list
neighbor prefix-list
neighbor route-reflector client
neighbor soft-reconfiguration
neighbor version
network backdoor
table-map
```

17. 不支持 VPN 配置命令

全部

注意：交换机不支持这个版本的命令参考中显示的多 VPN 路由/转发（多 VRF）命令。

18. 不支持路由映射命令

```
match route-type { level-1 | level-2 }
set as-path {tag |prepend as-path-string}
set automatic-tag
set dampening half-life reuse suppress max-suppress-time
set ip destination ip-address mask
set ip next-hop
set ip precedence value
set ip qos-group
set metric-type internal
set tag tag-value
```

19. MSDP：不支持特权模式 EXEC 命令

```
show access-expression
show exception
show location
show pm LINE
show smf [interface-id]
show subscriber-policy [policy-number]
show template [template-name]
```

20. MSDP: 不支持全局配置命令

```
ip msdp default-peer ip-address | name [prefix-list list]
```

因为不支持 BGP/Multiprotocol BGP (MBGP)，所以使用 **ip msdp peer** 命令代替这个命令。

21. RADIUS: 不支持全局配置命令

```
aaa nas port extended  
radius-server attribute nas-port  
radius-server configure  
radius-server extended-portnames
```

22. SNMP: 不支持全局配置命令

```
snmp-server enable informs
```

23. 生成树: 不支持全局配置命令

```
spanning-tree etherchannel guard misconfig
```

24. VLAN: 不支持用户模式 EXEC 命令

```
ifindex  
private-vlan
```

附录 B

RFC

表 B-1 列出了贯穿本书的一些较通用的 RFC。你可以在 www.rfc-editor.org/cgi-bin/rfcsearch.pl 在线找到所有的 RFC。在查找项中输入 RFC 编号即可。

表 B-1 本书的 RFC 参考

文件	标题	更新注意
RFC 3392	<i>BGP-4 的通告性能</i>	
RFC 3260	<i>区分服务的新术语和澄清</i>	
RFC 3248	<i>RFC 2598 延时范围修订版</i>	
RFC 3065	<i>BGP 的自治系统联盟</i>	
RFC 2918	<i>BGP-4 的路由更新性能</i>	
RFC 2892	<i>BGP-4 的通告性能</i>	
RFC 2796	<i>BGP 路由反射——全网状连接 IBGP 的一种替代</i>	
RFC 2750	<i>RSVP 对策略控制的扩展</i>	
RFC 2697	<i>单一速率三色标记</i>	
RFC 2598	<i>加速转发 PHB</i>	由 RFC 3246 更新
RFC 2597	<i>确保转发 PHB 组</i>	由 RFC 3260 更新
RFC 2519	<i>域间路由聚合构架</i>	
RFC 2475	<i>一个区分服务的体系结构</i>	由 RFC 3260 更新
RFC 2474	<i>IPv4 和 IPv6 报头中区分服务项 (DS 项) 的定义</i>	由 RFC 3260 更新
RFC 3392	<i>BGP-4 的通告性能</i>	
RFC 2385	<i>经 TCP MD5 签名选项的 BGP 会话保护</i>	
RFC 2362	<i>协议独立的组播-稀疏模式</i>	
RFC 2309	<i>互联网中队列管理和拥塞避免建议</i>	
RFC 2330	<i>IP 性能度量构架</i>	
RFC 2205	<i>资源预留协议 (RSVP) ——版本 1 功能规范</i>	由 RFC 2750 更新
RFC 1998	<i>BGP 团体属性在多主路由中的应用</i>	
RFC 1105	<i>边界网关协议 (BGP)</i>	由 RFC 1163 废弃
RFC 1075	<i>距离向量组播路由协议</i>	

附录 C

参考书目

下表提供了本书创作过程中查阅的有关原始资料信息。

资源	标题	Web 页	章	作者
<i>Bridging and IBM Networking Command Reference, Cisco IOS Software Release 12.0</i>				Cisco
<i>Cisco — Configuring IP Multicast Guides</i>				Cisco
<i>Cisco — Understanding Service Access Point Access Control Lists</i>	“Understanding Service Access Point Access Control Lists”			Cisco
<i>Cisco IOS Desktop Switching Software Configuration Guide</i>	“Creating and Maintaining VLANs”		Chapter 5	Cisco
<i>Router Products Configuration Guide</i>	“Configuring DSLw+”		Chapter 30	Cisco
<i>Software Configuration Guide — Release 5.4</i>	“Configuring Fast EtherChannel and Gigabit EtherChannel”		Chapter 7	Cisco
<i>Software Configuration Guide — Release 6.1</i>			Chapter 9 Chapter 12	Cisco
<i>Cisco IOS 12.1 and 12.2 Configuration Guides and Command Reference</i>				Cisco
<i>Software Configuration Guide, Release 5.2</i>	“Configuring Spanning Tree”		Chapter 8	Cisco
<i>Statement of Direction</i>	“10 Gigabit Ethernet Position Statement”			Cisco
Web 站点	“Understanding and Configuring FastEtherChannel on Cisco Switching and Routing Devices”	www.cisco.com		Cisco
Web 站点	“Understanding and Configuring Spanning-Tree Protocol (STP) on Catalyst Switches”	Cisco.com/warp/public/473/5.html		

续表

资源	标题	Web 页	章	作者
Web 站点	"Using the border Gateway Protocol for Interdomain Routing"	www.cisco.com		
Web 站点	"Configuring a Gateway of Last Resort Using IP Commands"	Cisco.com/warp/public/105/default.html		
Data sheet	"Cisco 1000BASE-T GBIC"			Cisco
<i>Router Products Configuration and Reference</i>	"Configuring Transparent Bridging"		Chapter 1	
Web 站点	"Connectors and Cables"	Cisco.com/univercd/cc/td/doc/product/lan/c2900x1/gbic/ig_gbic/mamopins.html		
<i>Layer 3 Switching Software Feature and Configuration Guide</i>	"Configuring Bridging"			
Web 站点	"Configuring BGP"	Cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/np1_c/lcbgp.htm#xtocid2382823		
Web 站点	"Configuring ISO CLNS"	Cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/np3_c/3cclns.htm		
Web 站点	"The American Registry for Internet Numbers"	www.arin.net		
Web 站点	"The Internet Society"	www.isoc.org		
Web 站点	"The North American Network Operators' Group"	www.nanog.org		
Web 站点	"Asia Pacific Network Information Centre"	www.apnic.net		
Web 站点	"RIPE Network Coordination Centre"	www.ripe.net		
	"BGP4 Inter-Domain Routing in the Internet"			John W. Stewart III
Web 站点	"Catalyst 3550 limitation and Restrictions"	www.cisco.com		
<i>CCIE Practical Studies, Volume 1*</i>				Karl Solie
<i>Cisco BGP-4 Command and Configuration Handbook</i>				Dr. William R. Parkhurst
<i>Cisco Catalyst 3550 Software and Hardware Configuration Guides and Command Reference</i>	"Configuring 802.1s and 802.1w STP"	www.cisco.com		Cisco website
<i>Cisco Internetwork Troubleshooting*</i>				Laura Chappell Dan Farkas

续表

资源	标题	Web 页	章	作者
<i>Cisco IOS 12.0 Quality of Service</i>				Cisco
<i>Cisco IOS Configuration Fundamentals</i>				
<i>Cisco IOS Dial Solutions</i>				Cisco
<i>CCIE Professional Development: Cisco LAN Switching*</i>				Kennedy Clark Kevin Hamilton
<i>Cisco Voice Over Frame Relay, ATM, and IP</i>		www.cisco.com		Cisco
<i>Converged Network Architectures</i>				Oliver C. Ibe
<i>Deploying Cisco Voice Over IP Solutions</i>		www.cisco.com		Cisco
<i>Developing IP Multicast Networks, Volume I</i>				Beau Williamson
<i>Integrating Voice and Data Networks</i>				Scott Keagy
<i>Interconnections: Bridges, Routers, Switches, and Internetworking Protocols</i>				Radia Perlman
<i>Internet Performance Survival Guide</i>				Geoff Huston
<i>Internet Routing Architectures, Second Edition*</i>				Sam Halabi Danny McPherson
<i>Internetworking SNA with Cisco Solutions</i>				George Sackett Nancy Sackett
<i>Internetworking Troubleshooting Handbook, Third Edition</i>				Faraz Shamim Zaheer Aziz Johnson Liu Abe Martey
<i>Internetworking with TCP/IP, Volume I*</i>				Douglas Comer
<i>IP Quality of Service</i>				Srinivas Vegesna
<i>IP Telephony</i>				Bill Douskalis
<i>Managing Cisco Network Security*</i>				Michael Wenstrom
<i>Network Consultants Handbook*</i>				Matthew J. Castelli
<i>Network Routing Architectures</i>				Sam Halabi
<i>Performance and Fault Management</i>				Paul L. Della Maggiara Christopher E. Elliott Robert L. Payone, Jr. Kent J. Phelps James M. Thompson
<i>Putting VoIP to Work: Softswitch Network Design and Testing</i>				Bill Douskalis
<i>Routing TCP/IP, Volume I*</i>				Jeff Doyle
<i>Routing TCP/IP, Volume II*</i>				Jeff Doyle Jennifer DeHaven Carroll
<i>TCP/IP Principle, Protocols, and Architectures</i>				Douglas E. Comer
<i>The Protocols TCP/IP Illustrated, Volume I</i>				W. Richard Stevens

*号表示该书已由人民邮电出版社翻译或影印出版，详情请查阅人民邮电出版社网站：www.ptpress.com.cn。

附录 D

IP 前缀列表

前缀列表在思科 IOS 软件版本 12.0 (3) T 中可用。对于用路由协议的路由通告过滤，你可以使用前缀列表作为代替标准 IP 访问列表的简化。虽然前缀列表通常用于边界网关协议 (BGP) 配置中，本附录演示了其他的方法，你可以使用前缀列表支持其他路由协议，如增强内部网关协议 (EIGRP)。通过下面这些规则，前缀列表介绍了一种为网络前缀通告创建过滤器的更新型的方法：

- 如同访问列表，前缀列表也是从头到尾按顺序处理。当出现一个匹配时，处理过程停止，不再读取剩下的输入项。
- 任何时候都可以添加输入项到前缀列表中。
- 默认空的前缀列表允许所有的前缀。
- 前缀列表不使用通配符掩码，像访问列表那样；它们使用子网长度掩码（例如，/24）。
- 与访问列表不同，前缀列表中的行可以使用序号编辑。
- 前缀列表包含一个隐含的 **deny any** 在每个列表的最后。
- 序号是自动生成的；但是，可以停止自动序号生成。在全局配置模式下使用下面的命令配置前缀列表：

```
ip prefix-list list-name [list-number] [sequence  
sequence-value] deny | permit network-address/length [ge  
ge-value] [le le-value]
```

表 D-1 说明了前缀列表语法的含义。

正如前面所讨论的，你可以使用与路由器配置模式中的分发列表有关的前缀列表，来过滤路由通告。IP 前缀列表的配置是简单的；同样，修改前缀列表相对配置来讲也是简单的。图 D-1 利用网络提供了对前缀列表配置按步骤

表 D-1 IP 前缀列表语法

命令/参数	描述
<i>list-name list-number</i>	指定前缀列表的名称或编号
<i>seq sequence-value</i>	(可选的) 序号。如果没有手工输入序号，就会产生一个自动序号。这些号码从 5 开始顺序生成，增量为 5
<i>deny permit</i>	指定前缀对一个匹配是允许还是拒绝
<i>network-address</i>	匹配的网络地址，以带点十进制格式输入
<i>/length</i>	以 bit 表示的子网掩码长度
<i>ge ge-value</i>	(可选) 指定匹配的最小前缀范围
<i>le le-value</i>	(可选) 指定匹配的最大前缀范围

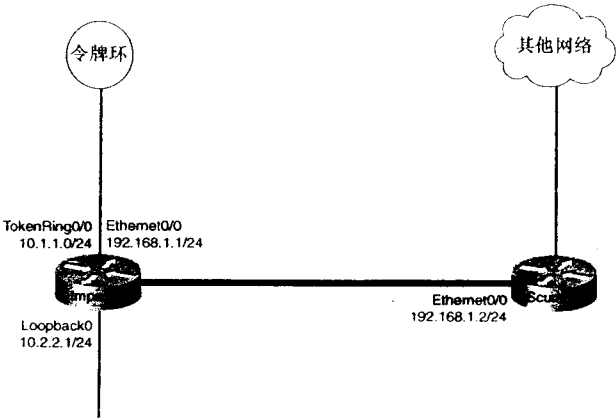


图 D-1 Artista 网络

下面的范例说明了如何使用前缀列表来过滤 EIGRP 路由协议的进入路由更新：

第 1 步 定义前缀列表：在这个范例中，前缀列表 Internal 用来指定 8 个 192.168.0.0/24 网络前缀：

```
ip prefix-list Internal seq 5 deny 192.168.0.0/24
ip prefix-list Internal seq 10 deny 192.168.1.0/24
ip prefix-list Internal seq 15 deny 192.168.2.0/24
ip prefix-list Internal seq 20 deny 192.168.3.0/24
ip prefix-list Internal seq 25 deny 192.168.4.0/24
ip prefix-list Internal seq 30 deny 192.168.5.0/24
ip prefix-list Internal seq 35 deny 192.168.6.0/24
ip prefix-list Internal seq 40 deny 192.168.7.0/24
```

第 2 步 创建一个分发列表，指定以前已配置的前缀列表：

```
router eigrp 100
 distribute-list prefix Internal in
```

为验证前缀列表工作了，从另一台路由器发一个 `show ip route` 命令。范例 D-1 显示了分发列表配置前的路由表。

范例 D-1 配置分发列表以前的路由表

Impasto# show ip route eigrp	
D	192.168.10.0/24 [90/409600] via 192.168.1.2, 00:00:03, Ethernet0/0
D	192.168.11.0/24 [90/409600] via 192.168.1.2, 00:00:03, Ethernet0/0

(待续)

```
D 192.168.4.0/24 [90/409600] via 192.168.1.2, 00:00:47, Ethernet0/0
D 192.168.5.0/24 [90/409600] via 192.168.1.2, 00:00:47, Ethernet0/0
D 192.168.6.0/24 [90/409600] via 192.168.1.2, 00:00:47, Ethernet0/0
D 192.168.7.0/24 [90/409600] via 192.168.1.2, 00:00:47, Ethernet0/0
D 192.168.2.0/24 [90/409600] via 192.168.1.2, 00:00:47, Ethernet0/0
D 192.168.3.0/24 [90/409600] via 192.168.1.2, 00:00:47, Ethernet0/0
```

范例 D-2 显示了应用分发列表和清空来自 Impasto 路由器的路由之后相同的路由表。

范例 D-2 应用分发列表之后的路由表

```
Impasto# clear ip route *
Impasto# show ip route eigrp
D 192.168.10.0/24 [90/409600] via 192.168.1.2, 00:00:41, Ethernet0/0
D 192.168.11.0/24 [90/409600] via 192.168.1.2, 00:00:41, Ethernet0/0
```

注意前缀列表提及的路由已经从路由表中删除了。范例 D-3 显示了这个范例中所使用的 Impasto 路由器的全部配置。

范例 D-3 使用 IP 前缀列表

```
interface Loopback0
 ip address 10.2.2.1 255.255.255.0
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface TokenRing0/0
 ip address 10.1.1.1 255.255.255.0
!
router eigrp 100
 network 10.0.0.0
 network 192.168.1.0
 distribute-list prefix Internal in
 no auto-summary
!
ip prefix-list Internal seq 5 deny 192.168.0.0/24
ip prefix-list Internal seq 10 deny 192.168.1.0/24
ip prefix-list Internal seq 15 deny 192.168.2.0/24
ip prefix-list Internal seq 20 deny 192.168.3.0/24
ip prefix-list Internal seq 25 deny 192.168.4.0/24
ip prefix-list Internal seq 30 deny 192.168.5.0/24
ip prefix-list Internal seq 35 deny 192.168.6.0/24
ip prefix-list Internal seq 40 deny 192.168.7.0/24
ip prefix-list Internal seq 45 permit 0.0.0.0/0 le 32
```

范例 D-4 演示了如何使用 **ge** 和 **le** 参数基于最小和最大前缀匹配来过滤路由。对这个范例，你需要有相同配置的两台相同的路由器。在 Impasto 上，用地址 11.1.1.1/24、11.2.1.1/16、11.30.1.1/13 和 11.200.1.1/10 创建 4 个环回接口。Impasto 和 Scumble 路由器在自治系统 100 中运行 EIGRP；Impasto 路由器将通告网络 10.0.0.0、192.168.1.0 和 11.0.0.0；两台路由器上应禁用汇总。

范例 D-5 显示了新增的 11.0.0.0 网络以及 Scumble 路由器的路由表。

范例 D-4 准备 Impasto 路由器

```

interface Loopback0
 ip address 10.2.2.1 255.255.255.0
 no ip directed-broadcast
!
interface Loopback10
 ip address 11.1.1.1 255.255.255.0
!
interface Loopback11
 ip address 11.2.1.1 255.255.0.0
!
interface Loopback12
 ip address 11.30.1.1 255.248.0.0
!
interface Loopback13
 ip address 11.200.1.1 255.192.0.0
!
interface Ethernet0/0
 ip address 192.168.1.2 255.255.255.0
!
router eigrp 100
 network 10.0.0.0
 network 11.0.0.0
 network 192.168.1.0 0.0.0.255
 no auto

```

范例 D-5 路由器 2 的路由表

```

Scumble# show ip route | include islvia
Gateway of last resort is not set
C    192.168.10.0/24 is directly connected, Loopback10
C    192.168.11.0/24 is directly connected, Loopback20
C    192.168.4.0/24 is directly connected, Loopback2
C    192.168.5.0/24 is directly connected, Loopback3
C    10.0.0.0/24 is subnetted, 2 subnets
D    10.2.2.0 [90/156160] via 192.168.1.1, 00:02:02, FastEthernet0
D    10.1.1.0 [90/178688] via 192.168.1.1, 00:02:02, FastEthernet0
C    192.168.6.0/24 is directly connected, Loopback4
C    11.0.0.0/8 is variably subnetted, 4 subnets, 4 masks
D    11.2.0.0/16 [90/156160] via 192.168.1.1, 00:02:02, FastEthernet0
D    11.1.1.0/24 [90/156160] via 192.168.1.1, 00:02:02, FastEthernet0
D    11.24.0.0/13 [90/156160] via 192.168.1.1, 00:02:02, FastEthernet0
D    11.192.0.0/10 [90/156160] via 192.168.1.1, 00:02:02, FastEthernet0
C    192.168.7.0/24 is directly connected, Loopback5
C    192.168.1.0/24 is directly connected, FastEthernet0
C    192.168.2.0/24 is directly connected, Loopback0
C    192.168.3.0/24 is directly connected, Loopback1

```

创建环回接口并验证 EIGRP 运转之后，创建一个 IP 前缀列表，它只允许 Impasto 路由器通告 11.1.0.0 网络，前缀范围从/16 到/32。应用这个前缀列表来过滤 EIGRP 路由，保持 Impasto 路由器如范例 D-6 所示。

范例 D-6 应用 IP 前缀列表

```

ip prefix-list Trial-2 seq 5 permit 11.1.0.0/16 le 32
!
router eigrp 100
 distribute-list prefix Trial-2 out

```

在 Impasto 路由器上应用前缀列表之后，Scumble 路由器的路由表将只包含到 11.1.1.0/24 网络的路由。掩码范围从 16~32 比特的其他 11.0.0.0 网络已经被删除，网络 10.2.2.0/24 也被删除，如范例 D-7 所示。

范例 D-7 配置 IP 前缀列表后 Scumble 路由器的路由表

```
Scumble# show ip route | include is|via
Gateway of last resort is not set
C    192.168.10.0/24 is directly connected, Loopback10
C    192.168.11.0/24 is directly connected, Loopback20
C    192.168.4.0/24 is directly connected, Loopback2
C    192.168.5.0/24 is directly connected, Loopback3
C    192.168.6.0/24 is directly connected, Loopback4
C    11.0.0.0/24 is subnetted, 1 subnets
D      11.1.1.0 [90/156160] via 192.168.1.1, 00:02:30, FastEthernet0
C    192.168.7.0/24 is directly connected, Loopback5
C    192.168.1.0/24 is directly connected, FastEthernet0
C    192.168.2.0/24 is directly connected, Loopback0
C    192.168.3.0/24 is directly connected, Loopback1
```

现在，删除 11.1.1.1/24 接口，在 Impasto 路由器的配置中添加环回接口 11.1.1.0/29、11.1.1.32/29 和 11.1.1.64/29；再次检查 Scumble 路由器上的路由表。它应该像范例 D-8 那样。

范例 D-8 以 IP 前缀列表进行实验

```
Impasto(config)# interface loopback 11
Impasto(config-if)# ip address 11.1.1.1 255.255.255.248
Impasto(config-if)# interface loopback 14
Impasto(config-if)# ip address 11.1.1.33 255.255.255.248
Impasto(config-if)# interface loopback 15
Impasto(config-if)# ip address 11.1.1.65 255.255.255.248

Impasto# show ip route | include is|via
Gateway of last resort is not set
D    192.168.10.0/24 [90/409600] via 192.168.1.2, 00:06:53, Ethernet0/0
D    192.168.11.0/24 [90/409600] via 192.168.1.2, 00:06:53, Ethernet0/0
C    10.0.0.0/24 is subnetted, 2 subnets
C      10.2.2.0 is directly connected, Loopback0
C      10.1.1.0 is directly connected, TokenRing0/0
C    11.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
C      11.2.0.0/16 is directly connected, Loopback11
C      11.1.1.0/29 is directly connected, Loopback10
C      11.24.0.0/13 is directly connected, Loopback12
C      11.1.1.32/29 is directly connected, Loopback14
C      11.1.1.64/29 is directly connected, Loopback15
C      11.192.0.0/10 is directly connected, Loopback13
C      192.168.1.0/24 is directly connected, Ethernet0/0
```

为了这个实验的下一部分，从 EIGRP 100 中删除流出方向的 Trial-2 前缀，更改前缀列表为所有 11.0.0.0/16 网络前缀大于 25 位（这包括前述步骤刚创建的环回接口，但是允许任何其他流量）。编辑了前缀列表之后，再运用它，如范例 D-9 所示。

应用了前缀列表的变更之后，Scumble 路由器的路由表应该再现 10.0.0.0 网络和掩码大于 16 的 11.0.0.0 网络。前面步骤中创建的环回应该已被删除，如范例 D-10 所示。

范例 D-9 实验继续

```

router eigrp 100
  no distribute-list prefix- Trial-2 out

ip prefix-list Trial-2 seq 5 deny 11.1.0.0/16 ge 25
ip prefix-list Trial-2 seq 10 permit 0.0.0.0/0 le 32

router eigrp 100
  distribute-list prefix- Trial-2 out

```

范例 D-10 改变前缀列表 Trial-2 后 Scumble 路由器的路由表

```

Scumble# clear ip route *
Scumble# show ip route | include is|via
Gateway of last resort is not set
C    192.168.10.0/24 is directly connected, Loopback10
C    192.168.11.0/24 is directly connected, Loopback20
C    192.168.4.0/24 is directly connected, Loopback2
C    192.168.5.0/24 is directly connected, Loopback3
C    10.0.0.0/24 is subnetted, 2 subnets
D      10.2.2.0 [90/156160] via 192.168.1.1, 00:00:16, FastEthernet0
D      10.1.1.0 [90/178688] via 192.168.1.1, 00:00:16, FastEthernet0
C    192.168.6.0/24 is directly connected, Loopback4
C    11.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D      11.2.0.0/16 [90/156160] via 192.168.1.1, 00:00:16, FastEthernet0
D      11.24.0.0/13 [90/156160] via 192.168.1.1, 00:00:16, FastEthernet0
D      11.192.0.0/10 [90/156160] via 192.168.1.1, 00:00:16, FastEthernet0
C    192.168.7.0/24 is directly connected, Loopback5
C    192.168.1.0/24 is directly connected, FastEthernet0
C    192.168.2.0/24 is directly connected, Loopback0
C    192.168.3.0/24 is directly connected, Loopback1

```

范例 D-11 显示了 Impasto 路由器完整的配置。

范例 D-11 Impasto 路由器的全部配置

```

interface Loopback0
  ip address 10.2.2.1 255.255.255.0
!
interface Loopback10
  ip address 11.1.1.1 255.255.255.248
!
interface Loopback11
  ip address 11.2.1.1 255.255.0.0
!
interface Loopback12
  ip address 11.30.1.1 255.248.0.0
!
interface Loopback13
  ip address 11.200.1.1 255.192.0.0
!
interface Loopback14
  ip address 11.1.1.33 255.255.255.248
!
interface Loopback15
  ip address 11.1.1.65 255.255.255.248

```

(待续)


```

!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface TokenRing0/0
 ip address 10.1.1.1 255.255.255.0
!
router eigrp 100
 network 10.0.0.0
 network 11.0.0.0
 network 192.168.1.0
 neighbor 192.168.1.2
 distribute-list prefix Trial-2 out
 distribute-list prefix Internal in
 no auto-summary
!
ip prefix-list Internal seq 5 deny 192.168.0.0/24
ip prefix-list Internal seq 10 deny 192.168.1.0/24
ip prefix-list Internal seq 15 deny 192.168.2.0/24
ip prefix-list Internal seq 20 deny 192.168.3.0/24
ip prefix-list Internal seq 25 deny 192.168.4.0/24
ip prefix-list Internal seq 30 deny 192.168.5.0/24
ip prefix-list Internal seq 35 deny 192.168.6.0/24
ip prefix-list Internal seq 40 deny 192.168.7.0/24
ip prefix-list Internal seq 45 permit 0.0.0.0/0 le 32
!
ip prefix-list Trial-2 seq 5 deny 11.1.0.0/16 ge 25
ip prefix-list Trial-2 seq 10 permit 0.0.0.0/0 le 32

```

用少量的练习，你可以使用更简单的前缀列表代替所有路由协议的访问列表，而不只是 BGP。