

006117600

TN915.04
143
V1

CCIE 职业发展系列

TCP/IP 路由技术

(第一卷)

Routing TCP/IP, Volume I

[美] Jeff Doyle, CCIE#1919 著

葛建立 吴剑章 译



好好
QQ:835047
Email:mr.goodgood@hotmail.com



北航 C1014735

人民邮电出版社

4.24
3
V1
X

图书在版编目 (CIP) 数据

TCP/IP 路由技术. 第一卷 / (美) 多伊尔 (Doyle, J.) 著; 葛建立, 吴剑章译.
—北京: 人民邮电出版社, 2003.10
(CCIE 职业发展系列)
ISBN 7-115-11571-0

I. T… II. ①多… ②葛… ③吴… III. 计算机网络—通信协议—路由选择
IV. TN915.04

中国版本图书馆 CIP 数据核字 (2003) 第 080308 号

版 权 声 明

Jeff Doyle: Routing TCP/IP, Volume I (ISBN: 1-57870-041-8)

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

Copyright © 1998 by Macmillan Technical Publishing

All rights reserved.

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

版权所有, 侵权必究。

CCIE 职业发展系列 TCP/IP 路由技术 (第一卷)

- ◆ 著 [美] Jeff Doyle, CCIE#1919
- 译 葛建立 吴剑章
- 责任编辑 杨长青
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
- 邮编 100061 电子函件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 北京顺义振华印刷厂印刷
- 新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
- 印张: 41.25
- 字数: 1 005 千字
- 印数: 8 501 - 10 500 册
- 2003 年 10 月第 1 版
- 2005 年 9 月北京第 5 次印刷

著作权合同登记 图字: 01-2003-0659 号

ISBN 7-115-11571-0/TP · 3587

定价: 79.00 元

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

内 容 提 要

本书是第一本详细而又完整地介绍互联网络内部网关协议 (IGP) 的专业书籍, 堪称有关 IGP 方面不可多得的经典之作。本书共分三个部分。第一部分主要介绍了网络和路由选择的基本知识, 对 TCP/IP 和静态、动态路由选择技术作了一个整体的回顾。第二部分是本书的精华, 这一部分详细深入地讲述了各种常用的内部网关协议, 如静态路由、RIP、RIPv2、IGRP、EIGRP、OSPF、ISIS 等, 每一章除了对该协议的实现机制和参数详尽阐述, 使读者对协议的实现原理有一个清晰的理解外, 还通过在实际网络环境中的实例, 详细地论述了该协议在 Cisco 路由器上的配置和故障处理方法, 使读者获取大量解决实际问题的专业技能。第三部分介绍了如缺省路由、路由过滤等多种有效的路由控制工具, 用来创建和管理多个 IP 路由选择协议的协调工作。

本书不仅适合那些需要准备通过 CCIE 考试的考生, 而且也适合任何需要完整理解 TCP/IP 内部路由选择协议的网络设计和工程人员阅读。本书中对协议细节的讲解和对网络实例的探讨相信会让读者受益匪浅。

Forward for Chinese–Language Edition

Since the publication of *Routing TCP/IP* Volumes I and II, I have had many opportunities to visit the People's Republic of China. In no other country have I received as many warm compliments on these books as I have in China. I am therefore delighted that PT Press is now offering a version of Volume I in Mandarin.

China is aggressively expanding its Internet infrastructure, and along with it services such as mobile IP and online gaming. In the next few years, I predict that China will become the world's leader in the commercial implementation of IPv6. Already China exceeds Japan in the number of PCs, and it will soon have the world's largest mobile network. All of this expansion means that over the coming decade, there will be an enormous increase in the demand for IP networking experts. The Cisco Certified Internet Expert program provides the opportunity for potential employers to discern the best networking engineers, and it provides those holding the certification a testament to their expertise. Therefore the CCIE program plays an increasingly important role in the growing Chinese networking industry. I hope that you, the reader, will find this book useful in your preparation to earn this coveted certification.

Best Regards,

Jeff Doyle



中文版序

自从《TCP/IP 路由技术》第一卷和第二卷出版以来，我有了很多机会可以来到中国。这些书在其他国家都没有像在中国这样受到大家的热情关注。因此，我很高兴人民邮电出版社能够在中国出版本书第一卷的中文版本。

在中国，Internet 基础设施正在迅猛发展与普及，而且也提供了诸如移动 IP 和在线游戏之类的服务。在未来的几年，我预计中国将成为 IPv6 商业化部署的世界领导者。在 PC 机的拥有量上，中国已经超过了日本，并且在不远的将来，中国会拥有世界上最大的移动网络。所有这些巨大的发展都意味着在将来的 10 年里，中国对 IP 网络专家的需求量将会有很大的增长。Cisco 认证互联网络专家（CCIE）计划可以为企业管理者提供一个渠道，以便识别出最优秀的互联网络工程师；同时，它也为专业技能的证明提供了确实的依据。因此，CCIE 计划在中国互联网络产业的发展中将扮演日益重要的角色。我非常期待读者在获取令人羡慕的认证准备过程中，能从本书获益。

最诚挚的问候

Jeff Doyle



原书序

在当今的网络互联领域，担当关键业务的网络正在被设计用来为数据、语音和视频等业务提供服务。由于通信流量的模式和每种业务信息所要求的服务质量（Qos）各不相同，因此拥有丰富可靠的实践经验，对这些网络的管理、设计和故障诊断是很有必要的。

获取熟练的实践经验可以转化成对现代网络的原理、扩展性能和部署问题的深入理解。一些经验也可以形成分析流量模式的专门技术和何时、何处以及怎样应用协议和带宽的特性去增强性能。

为了进一步提高你的实践经验，Cisco Press 正在出版一系列关于 CCIE 专业开发的书籍。这个系列的书籍将有效地帮助你理解网络协议的概念，而且书内提供了大量现实生活中的实例和案例研究用来加强对理论概念的检验。我力荐读者把这些书作为手头的学习工具，并在 Cisco 公司的产品上实现书中的实例和案例研究。你甚至可以进一步修改一些配置参数，使用 Cisco 产品提供的强大的调试工具来观察网络发生了哪些变化。

本书是“CCIE 职业发展”系列图书的第一本书，在本书中，Jeff Doyle 出色地完成了从 IP 地址分类到协议度量分析等 TCP/IP 原理的讲解。每一章都包含实例、标注 IP 地址的网络拓扑、数据包的分析 and Cisco 调试（debug 命令）工具的输出信息。依我看来，书中最有价值的部分是案例研究，Jeff 通过增加或减少一些相似的网络拓扑来比较网络协议的不同特性，使读者对协议的概念和特性有深刻的理解。

我建议所有准备参加网络互联认证的同行能阅读本书，同时，我也相信本书将会成为一本优秀的大学网络课程教材。

CCIE 项目经理
Imran Qureshi

作者简介

Jeff Doyle 是位于科罗拉多州丹佛地区的国际网络服务 (INS) 的高级网络系统顾问。他是 Cisco 认证的互联网络专家 (CCIE#1919)，而且是 Cisco 认证的系统进师。他已经开发和讲述了多种网络和互联网络方面的课程。读者可以通过电子信箱 Jeff.Doyle@ins.com 和 Jeff 取得联系。

关于技术审稿人

Jennifer DeHaven Carroll 是 International Network Services 的首席顾问，她也是 CCIE——CCIE # 1402。Jennifer 在过去的 10 年中，利用 RIPv2、IGRP、EIGRP、OSPF 和 BGP 协议，规划、设计并实施了许多 IP 网络。她也开发和教授过所有 IP 路由选择协议的理论知识和 Cisco 的实际应用课程。

Michael Tibodeau 是 Cisco 系统公司的系统工程师，在过去的两年里，Michael 专门为客户和网络听众提供网络安全技术指导。他也关注于电子商务和服务质量保证（QoS）方面的研究。Michael 拥有弗吉尼亚大学系统工程专业的学士学位和系统工程与电信管理方面的硕士学位。

致 谢

如果没有许多对本书作出贡献的人们，没有他们的一致努力，本书是不可能完成的。请允许我在此对下面的各位以及他们对本书所作出的贡献表示真诚的谢意：

首先，我要感谢本书的开发编辑 Laurie McGuire，他不仅提高了本书的质量，而且也提高了我的写作水平。

感谢本书的技术编辑 Jenny DeHaven Carroll 和 Mike Tibodeau，感谢他们精心细致的编辑工作。

我也要感谢 Howard Berkowitz、Dave Katz、Burjiz Pithawala、Mikel Ravizza、Russ White 和 Man-Kit Yueng，他们给我提供了技术上的建议或审阅了本书中的一些章节。

感谢 Macmillan Technical Publishing 的 Tracy Hughes 和 Lynette Quinn，他们是本书的项目管理人员，还有执行编辑 Julie Fairweather。他们除了可以完全胜任自己的工作外，而且非常易于相处，任何人都会希望与他们一起工作。另外，要感谢助理出版人 Jim LeValley，他是第一个和我商洽写作本书的人。

当然，我还要感谢 Wandel & Golterman 公司和 Gary Archuleta 先生，Gary Archuleta 是 W&G 公司丹佛地区的销售经理，他积极安排使用他们优秀的协议分析仪提供的信息丰富了本书的内容。

最后，我想感谢我的妻子 Sara 和我的孩子们：Anna、Carol、James 和 Katherine。他们的耐心、鼓励和支持对于本书的完成是十分重要的。

前言

路由技术即使在最小的数据通信网络中也是基本的要素。在某种程度上，路由技术和路由器的配置是相当简单的。但是当互联网络的规模越来越大，并且越来越复杂的时候，路由选择问题就变得比较突出和难以控制了。或许，有点不恰当地说，作为一名网络系统顾问，我应该感谢当前出现的大规模路由技术难题，这些问题给了我谋生的手段。假设没有它们，“你何以为生？”这句习语可能就会不幸地成为我每天生活词汇的一部分了。

Cisco 认证互联网络专家 (CCIE) 在大型互联网络的设计、故障排除和管理能力方面得到广泛的认同。这种广泛的认同来自于这样一个事实：一个网络工作人员仅仅依赖参加一些课程的培训，并反复依赖记忆一些书面测试的内容是不可能成为一名 CCIE 的。一名 CCIE 必须通过一个众所周知、难度非常大的、并且需要亲自动手操作的实验室考试，从而使他或者她的专业技能得到提高。

本书的目标

本书是一系列设计用来帮助读者成为 Cisco 认证互联网络专家的丛书的第一本，也是专门讨论 TCP/IP 路由选择问题的两卷书中的第一本。在这个项目的早期，Cisco 公司的程序经理 Kim Lew 说过：“我们的目标是使人们成为 CCIE，而不是使人们通过 CCIE 实验室考试。”作者完全赞同这种观点，并且把它作为一种指导原则贯穿到本书的写作当中。虽然这本书包括了很多案例研究和练习可以帮助读者准备 CCIE 实验室考试，但是作者的主要目的还是提高读者对 IP 路由技术的理解——能有一个普通的水平并能够在 Cisco 的路由器上实现。

读者对象

本书的读者可以是任何需要完整理解 TCP/IP 内部路由选择协议的网络设计人员、管理人员或者工程人员。虽然本书的实践方面针对 Cisco IOS, 但是本书的资料也可以应用于任何路由选择平台。

这本书不仅仅是写给那些计划成为 Cisco 认证互联网络专家的读者阅读的, 而且是写给任何希望提高自己的 TCP/IP 路由选择技能的读者。这些读者可以划分为以下三类:

- “初学者”——具有基本的网络知识, 并且希望开始深入学习互联网络的读者;
- 中级水平的网络专业人员——具有一定的路由器 (Cisco 或其他厂商的产品) 操作经验, 并且计划提高自己的技能达到专家水平的读者;
- 经验丰富的网络专家——这些读者具有丰富和广泛的 Cisco 路由器的实践经验和专业技能, 并且准备参加 CCIE 实验室考试。但是, 这类读者需要自己制定一个复习表和一系列检验与确认自己技能的练习。

本书主要面向具有中级水平的网络专业人员。同时, 对于初学者, 本书提供了一个网络基本知识的概要。而对于网络方面的专家而言, 本书也提供了一些磨炼他们的专业技能所需要的挑战性内容。

本书的内容组织

本书共有 14 章, 分为 3 个部分。

第一部分回顾了网络和路由技术的基本知识。虽然一些水平较高的读者希望跳过开始的两个章节, 但是我建议这些读者至少应该浏览一下第 3 章“静态路由”和第 4 章“动态路由选择协议”的内容。

第二部分包括了 TCP/IP 路由选择的各种内部网关协议。讲解具体协议的每一章都是从该协议的实现机制和参数开始的, 并在读者对该协议有了一个总体的了解后, 接着通过多个不同的网络拓扑环境中的实例, 详细地讲述了该协议在 Cisco 路由器上的配置和故障排除方法。

外部网关协议, 还有组播路由选择、服务质量保证、路由器的安全与管理以及 IPv6 路由选择等一些主题, 将在第二卷中介绍。

第三部分介绍了多种有效的工具, 用来创建和管理多个 IP 路由选择协议的协调工作, 例如缺省路由、路由过滤等。这些章节和第二部分的每章一样, 也是先从概念开始讲解, 并随后给出多个不同的实例。

惯例和风格

大多数章节在结束时都配有一组复习题、配置练习和故障排除练习。复习题主要侧重于每章主题的概念理论方面, 而配置和故障排除练习主要侧重于该协议在 Cisco 设备上的实际实现。

在每章末尾还列出了一张命令总结表, 简要介绍了在这一章中使用到的 Cisco IOS 中的所有重要命令。这些命令使用的惯例和 Cisco IOS 命令参考中使用的惯例是一样的。命令参考中约定的这些惯例如下:

- 竖线 (|) 表示在几个选项中选择一项，并且这些项是互相排斥的；
- 方括号[]表示可选的参数；
- 大括号{}表示一个必需的选项；
- 方括号内嵌大括号[{}]表示在一个可选项里面的必需选项；
- **粗体字**表示实际需要键入的命令和关键字；
- *斜体字*表示需要用实际数值替换的参数；

在图 I-1 中，显示了本书的图示中所用到的惯例表示。

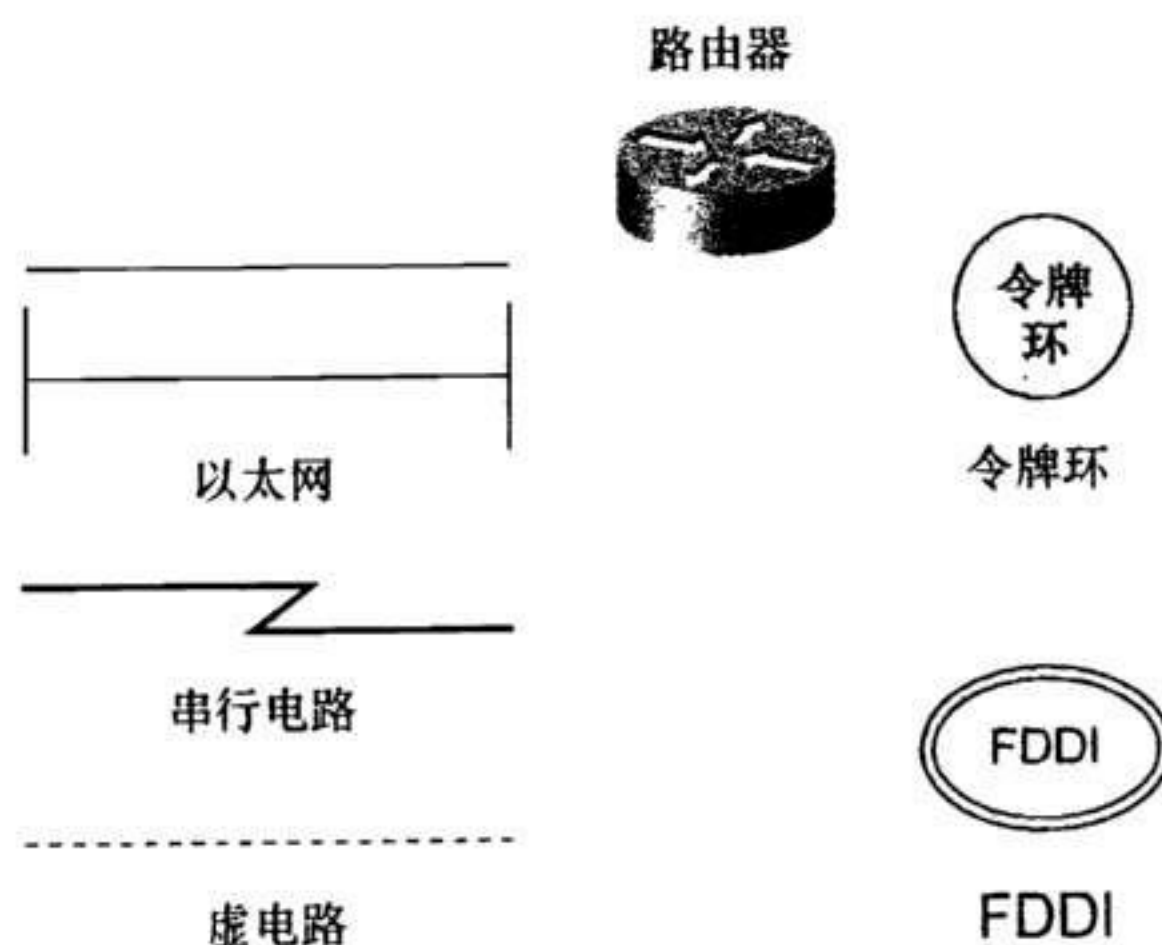


图 I-1 本书用到的惯例图示

本书显示的所有的协议分析仪信息都是使用 Wandel&Goltermann DA-320 DominoLAN 互联网络分析仪。

目 录

第一部分 路由技术基础知识

第 1 章 基本概念：互联网络、路由器和地址	5
1.1 带马达的自行车	6
1.2 数据链路地址	6
1.3 中继器和网桥	9
1.4 路由器	12
1.5 网络地址	14
1.6 展望	15
1.7 参考读物	16
1.8 复习题	16
第 2 章 TCP/IP 回顾	19
2.1 TCP/IP 协议层	19
2.2 IP 报文头	21
2.3 IP 地址	28
2.3.1 首个 8bit 字节规则	29
2.3.2 地址掩码 (Address Mask)	30
2.3.3 子网和子网掩码	32
2.3.4 子网规划	34
2.3.5 打破 8bit 字节界线	35
2.3.6 子网掩码的故障排除	37
2.4 ARP	38
2.4.1 代理 ARP	42
2.4.2 无故 ARP	43
2.4.3 反向 ARP	43
2.5 ICMP	44
2.6 主机到主机层	47
2.6.1 TCP	47
2.6.2 UDP	49
2.7 展望	50

2.8 总结表: 第 2 章命令回顾	50
2.9 推荐读物	51
2.10 复习题	51
2.11 配置练习	52
2.12 故障排除练习	52
第 3 章 静态路由	55
3.1 路由选择表	56
3.2 配置静态路由	58
3.2.1 案例研究: 简单静态路由	58
3.2.2 案例研究: 汇总路由	61
3.2.3 案例研究: 选择路由	61
3.2.4 案例研究: 浮动静态路由 (Floating Static Route)	63
3.2.5 案例研究: 均分负载	65
3.2.6 案例研究: 递归表查询	67
3.3 静态路由故障排除	69
3.3.1 案例研究: 追踪故障路由	69
3.3.2 案例研究: 协议冲突	72
3.4 展望	75
3.5 总结表: 第 3 章命令回顾	76
3.6 复习题	76
3.7 配置练习	76
3.8 故障排除练习	77
第 4 章 动态路由选择协议	83
4.1 路由选择协议基础	84
4.1.1 路径决策	84
4.1.2 度量	85
4.1.3 收敛	87
4.1.4 负载均衡	88
4.2 距离矢量路由选择协议	88
4.3 链路状态路由选择协议	95
4.3.1 邻居	95
4.3.2 链路状态泛洪扩散	96
4.3.3 SPF 算法	102
4.3.4 区域	105
4.4 内部和外部网关协议	106
4.5 静态或动态路由	107
4.6 展望	108
4.7 推荐读物	108

4.8 复习题	108
---------------	-----

第二部分 内部路由选择协议

第 5 章 路由选择信息协议 (RIP)	113
5.1 RIP 的操作	114
5.1.1 RIP 的计时器和稳定性	114
5.1.2 RIP 消息格式 (RIP Message Format)	116
5.1.3 请求消息类型 (Request Message Type)	118
5.1.4 有类别路由选择(Classful Routing)	118
5.2 配置 RIP	122
5.2.1 案例研究 1: 一个基本的 RIP 配置	122
5.2.2 案例研究 2: 被动接口 (Passive Interface)	124
5.2.3 案例研究 3: 配置单播更新(Unicast update)	125
5.2.4 案例研究 4: 不连续的子网	127
5.2.5 案例研究 5: 掌握 RIP 的度量	129
5.3 RIP 故障排除	132
5.4 展望	132
5.5 总结表: 第 5 章命令总结	132
5.6 推荐读物	133
5.7 复习题	133
5.8 配置练习	133
5.9 故障排除练习	134
第 6 章 内部网关路由选择协议 (IGRP)	141
6.1 IGRP 的操作	142
6.1.1 IGRP 的计时器和稳定性	144
6.1.2 IGRP 的度量	145
6.1.3 IGRP 的报文格式	150
6.2 配置 IGRP	153
6.2.1 案例研究 1: 一个基本的 IGRP 配置	153
6.2.2 案例研究 2: 非等价负载均衡 (一)	154
6.2.3 案例研究 3: 设置最大的路径数	157
6.2.4 案例研究 4: 多个 IGRP 进程	158
6.3 IGRP 故障排除	160
6.3.1 案例研究 5: 非等价负载均衡 (二)	160
6.3.2 案例研究 6: 被分段的网络 (Segmented Network)	162
6.4 展望	164
6.5 总结表: 第 6 章命令总结	164
6.6 推荐读物	164

6.7 复习题	165
6.8 配置练习	165
6.9 故障排除练习	168
第 7 章 路由选择信息协议——第 2 版 (RIPv2)	175
7.1 RIPv2 的操作	176
7.1.1 RIPv2 的消息格式	176
7.1.2 与 RIPv1 的兼容性	178
7.1.3 无类别路由查找	179
7.1.4 无类别路由选择协议	179
7.1.5 可变长子网掩码 (VLSM)	180
7.1.6 认证	182
7.2 配置 RIPv2	184
7.2.1 案例研究 1: 一个基本的 RIPv2 配置	185
7.2.2 案例研究 2: 与 RIPv1 的兼容性	185
7.2.3 案例研究 3: 使用 VLSM	187
7.2.4 案例研究 4: 不连续的子网和无类别路由选择	189
7.2.5 案例研究 5: 认证	191
7.3 RIPv2 故障排除	193
7.4 展望	198
7.5 总结表: 第 7 章命令总结	198
7.6 推荐读物	199
7.7 复习题	199
7.8 配置练习	199
7.9 故障排除练习	200
第 8 章 增强型内部网关路由选择协议 (EIGRP)	205
8.1 EIGRP 的操作	207
8.1.1 依赖于协议的模块 (Protocol-Dependent Modules)	207
8.1.2 可靠传输协议 (RTP)	208
8.1.3 邻居的发现和恢复	209
8.1.4 扩散更新算法 (Diffusing Update Algorithm)	210
8.1.5 EIGRP 的报文格式	227
8.1.6 地址聚合	232
8.2 配置 EIGRP	235
8.2.1 案例研究 1: 一个基本的 EIGRP 配置	235
8.2.2 案例研究 2: 和 IGRP 的重新分配	237
8.2.3 案例研究 3: 关闭自动路由汇总	239
8.2.4 案例研究 4: 地址聚合 (Address Aggregation)	240
8.2.5 案例研究 5: 认证	242

8.3 EIGRP 故障排除	242
8.3.1 案例研究 6: 邻居丢失 (A Missing Neighbor)	243
8.3.2 “卡”在活动状态的邻居 (Stuck-in-Active Neighbors)	248
8.4 展望	250
8.5 总结表: 第 8 章命令总结	250
8.6 复习题	251
8.7 配置练习	252
8.8 故障排除练习	254
第 9 章 开放最短路径优先协议 (OSPF)	257
9.1 OSPF 的操作	258
9.1.1 邻居和邻接关系	259
9.1.2 区域 (Area)	285
9.1.3 链路状态数据库	291
9.1.4 路由选择表	303
9.1.5 认证	307
9.1.6 按需电路上的 OSPF	307
9.1.7 OSPF 的报文格式	308
9.1.8 OSPF 的 LSA 格式	315
9.1.9 可选项字段	321
9.2 配置 OSPF	322
9.2.1 案例研究 1: 一个基本的 OSPF 配置	322
9.2.2 案例研究 2: 使用 Loopback 接口设置路由器的 ID	325
9.2.3 案例研究 3: 域名服务查询	327
9.2.4 案例研究 4: OSPF 和辅助地址	328
9.2.5 案例研究 5: 末梢区域	333
9.2.6 案例研究 6: 完全末梢区域	335
9.2.7 案例研究 7: NSSA 区域	336
9.2.8 案例研究 8: 地址汇总	342
9.2.9 案例研究 9: 认证	344
9.2.10 案例研究 10: 虚链路	346
9.2.11 案例研究 11: 运行在 NBMA 网络上的 OSPF	348
9.2.12 案例研究 12: 运行在按需电路上的 OSPF	355
9.3 OSPF 故障排除	356
9.3.1 案例研究 13: 孤立的区域	359
9.3.2 案例研究 14: 路由汇总配置错误	362
9.4 展望	364
9.5 总结表: 第 9 章命令总结	365
9.6 推荐读物	366
9.7 复习题	366

9.8 配置练习	367
9.9 故障排除练习	368
第 10 章 集成 IS-IS 协议	373
10.1 集成 IS-IS 协议的操作	375
10.1.1 IS-IS 区域	376
10.1.2 网络实体标题	378
10.1.3 IS-IS 的功能结构	379
10.1.4 IS-IS 的 PDU 格式	389
10.2 配置集成 IS-IS 协议	407
10.2.1 案例研究 1: 一个基本的集成 IS-IS 配置	408
10.2.2 案例研究 2: 更改路由器的类型	412
10.2.3 案例研究 3: 区域的迁移	415
10.2.4 案例研究 4: 路由汇总	418
10.2.5 案例研究 5: 认证	420
10.3 集成 IS-IS 协议的故障排除	422
10.3.1 IS-IS 邻接关系的故障排除	423
10.3.2 IS-IS 链路状态数据库的故障排除	424
10.3.3 案例研究 6: 运行于 NBMA 网络上的集成 IS-IS	427
10.4 展望	431
10.5 总结表: 第 10 章命令总结	431
10.6 复习题	432
10.7 配置练习	433
10.8 故障排除练习	434

第三部分 路由控制和互操作性

第 11 章 路由重新分配	439
11.1 重新分配的原则	441
11.1.1 度量	441
11.1.2 管理距离	442
11.1.3 从无类别协议向有类别协议重新分配	446
11.2 配置重新分配	449
11.2.1 案例研究: 重新分配 IGRP 和 RIP	451
11.2.2 案例研究: 重新分配 EIGRP 和 OSPF	452
11.2.3 案例研究: 重新分配和路由汇总	457
11.2.4 案例研究: 重新分配 IS-IS 和 RIP	461
11.2.5 案例研究: 重新分配静态路由	464
11.3 展望	466
11.4 总结表: 第 11 章命令回顾	466

11.5 复习题	467
11.6 配置练习	467
11.7 故障诊断练习	468
第 12 章 缺省路由和按需路由选择	471
12.1 缺省路由基本原理	472
12.2 按需路由基本原理	473
12.3 配置缺省路由和 ODR	475
12.3.1 案例研究: 静态缺省路由	476
12.3.2 案例研究: 缺省网络命令	478
12.3.3 案例研究: 缺省信息发生命令	481
12.3.4 案例研究: 配置按需路由	484
12.4 展望	485
12.5 总结表: 第 12 章命令回顾	485
12.6 复习题	485
第 13 章 路由过滤	489
13.1 配置路由过滤器	490
13.1.1 案例研究: 过滤特定路由	491
13.1.2 案例研究: 路由过滤和重新分配	494
13.1.3 案例研究: 协议迁移	496
13.1.4 多个重新分配点	501
13.1.5 案例研究: 使用距离设置路由器优先权	506
13.2 展望	507
13.3 总结表: 第 13 章命令回顾	507
13.4 配置练习	508
13.5 故障诊断练习	510
第 14 章 路由图	513
14.1 路由图的基本用途	513
14.2 配置路由图	515
14.2.1 案例研究: 策略路由选择	517
14.2.2 案例研究: 策略路由选择和服务质量路由	523
14.2.3 案例研究: 路由图和重新分配	525
14.2.4 案例研究: 路由标记	529
14.3 展望	534
14.4 总结表: 第 14 章命令回顾	534
14.5 复习题	535
14.6 配置练习	535
14.7 故障诊断练习	536

第四部分 附录

附录 A 教程：二进制和十六进制	541
A.1 二进制数	542
A.2 十六进制数	543
附录 B 教程：访问列表	547
B.1 访问列表基础知识	548
B.1.1 隐式拒绝一切	549
B.1.2 顺序性	549
B.1.3 访问列表类型	550
B.1.4 编辑访问列表	552
B.2 标准 IP 访问列表	552
B.3 扩展 IP 访问列表	554
B.3.1 TCP 访问列表	556
B.3.2 UDP 访问列表	557
B.3.3 ICMP 访问列表	557
B.4 调用访问列表	558
B.5 可供选择的關鍵字	560
B.6 命名访问列表	560
B.7 对过滤表放置的考虑	561
B.8 访问列表的监视和计费	563
附录 C CCIE 小提示	567
C.1 牢固的基础	568
C.2 实践经验	568
C.3 深入学习	569
C.4 最后 6 个月	569
C.5 参加考试	570
附录 D 复习题答案	573
附录 E 配置练习答案	587
附录 F 故障排除练习答案	623
索引	629

第一部分

路由技术基础知识

第1章 基本概念：互联网络、路由器和地址

第2章 TCP/IP 回顾

第3章 静态路由

第4章 动态路由选择协议

第 1 章

基本概念：互联网、路由器和地址

本章包括以下主题：

- 带马达的自行车
- 数据链路地址
- 中继器和网桥
- 路由器
- 网络地址

以前，计算处理和数据存储都采用集中化的方式。大型机通常被锁在恒温、高度安全的房间内，由身份特殊的 IS 管理员看守。与计算机打交道的典型方式是将一叠 Hollerith 卡片交给管理员，由他代替使用者向大型机输入卡片。

小型机的出现使得计算机走出了公司和大学专人看守的神秘殿堂，进入院系/部门开始使用。由于一台小型机的价格仅为 10 万或 20 万美金，因此工程部门、会计部门或其他需要数据处理的部门能够拥有他们自己的小型机。

紧跟小型机之后出现的是微型计算机，微机使得数据处理工作可以在桌面上完成。用得起、买得到的微型计算机将计算机的使用领域从部门级拓展到个人，并使个人计算机这一名词成为了一种大众词汇。

虽然桌面计算以一种令人难以置信的速度发展起来，但是并未立即替代集中式、基于大型主机的计算模式。仍然需要经过一个攀升期，当个人计算机的软硬件发展到人们普遍认可的水平后，桌面计算才有可能直接替代基于大型主机的集中式计算。

1.1 带马达的自行车

分散式计算的难点之一是用户与用户相互分离, 用户与公共数据、应用程序相互分离。当一个文件被建立后, 在同一办公大厅工作的 Tom、Dick 和 Harriet 如何共享该文件?

早期解决这一问题的方法是有名的人力网络 (SneakerNet): 由人将文件拷贝到软盘上, 然后携带到需要的目的地。但是当 Tom、Dick 和 Harriet 修改了各自的文件拷贝时会发生什么情况? 如何保证文件所有版本中的信息同步? 如果上面提及的 3 个合作者在不同楼层、不同建筑物或不同城市又会怎样呢? 如果文件在一天当中需要多次更新呢? 如果不是 3 个合作者, 而是 300 个呢? 如果所有 300 人临时需要打印一份他们作过修改的文件硬拷贝呢?

局域网, 或称 LAN, 实际上向集中方式又退回了一小步。LAN 是一种实现集中资源和共享资源的工具。服务器使大家可以访问一个公共文件或数据库; 而不再需要携带软盘步行传输信息, 更不必担心信息内容的不一致问题。打电话需要接电话者必须到场, 传统的邮政服务又因速度慢被称为蜗牛信件, 而电子邮件提供了一种介于二者之间的折中办法。打印机和调制解调器池的共享减少了每个办公桌对这些昂贵的、需周期性使用的服务的需求量。

当然在局域网出现的初期, 它遭受了来自于大型机制造商的嘲笑。在早先的几年, 通常会听到这样的嘲讽, “局域网不就像是一辆带马达的自行车, 我们要的可不是这种两用车!” 时光和金钱让 LAN 发生了天翻地覆的变化。

从物理上看, 局域网通过普通的共享介质或数据链路将一组设备互联起来, 以实现资源集中。这种介质可以是双绞线 (屏蔽或非屏蔽)、同轴电缆、光纤、红外线。无论介质是什么, 重要的是所有的设备一般都是通过某种网络接口连接到数据链路上。

仅有共享的物理介质是不够的, 还需要规则来控制如何共享数据链路。就像在任何一个社区中, 为了保障生活秩序, 有一套规则是很有必要的, 它可以使各团体行为规矩, 保证每个都可以公平地分享可用资源。对于局域网, 这套规则或协议通常称为介质访问控制 (MAC)。顾名思义, MAC 规定了每台机器如何访问和共享指定的介质。

到目前为止, 局域网被定义为是在公共的通信介质上, 遵循公共协议的一个设备群体, 这些设备包括 PC、打印机和服务器等设备, 其中公共协议控制着设备如何访问介质。还有一点, 就像在任何一个社区中一样, 每个个体都必须具有惟一的标识。

1.2 数据链路地址

在美国科罗拉多州某社区, 有两个人名字都叫 Jeff Doyle。其中一个人经常接到别人打给他同名的另一个人的电话, 这样打错的电话非常多, 因此 Jeff Doyle 聪明的妻子将另一个 Jeff Doyle 的电话号码贴在电话机旁, 以便直接告诉因重名而打错电话的人。换句话说, 因为不能从姓名上惟一地标识两个人, 所以不时会出现数据传递错误, 并且需要专门的程序来纠正这一错误。

在亲属、朋友和同事之间, 使用姓名来准确地区分不同的人往往是足够了。然而, 正像上面的例子所示, 在一个更大的人群中, 大多数姓名将不能准确地区别不同个体。因此, 需

要一个更加独特的标识来完成这项工作，例如美国社会保障号码。

在局域网设备上必须被唯一地标识，否则，就像同名的人一样，会收到错发给自己的报文。当数据在局域网上传递时，数据被封装在一个实体中，这是一种二进制的信封，叫做帧。如图 1-1 所示，数据封装是就像在信封中装信一样，只不过是数字化而已¹。目的地址和回复(源)地址被写在信封外面。没有收信人地址，邮局将不知道向哪里递送信件。同样的，当一个数据帧被放到数据链路上时，所有连接到数据链路上的设备都将“看”到它；因而，某种机制必须指明哪一台设备应该获取它并且查看被封装的数据。

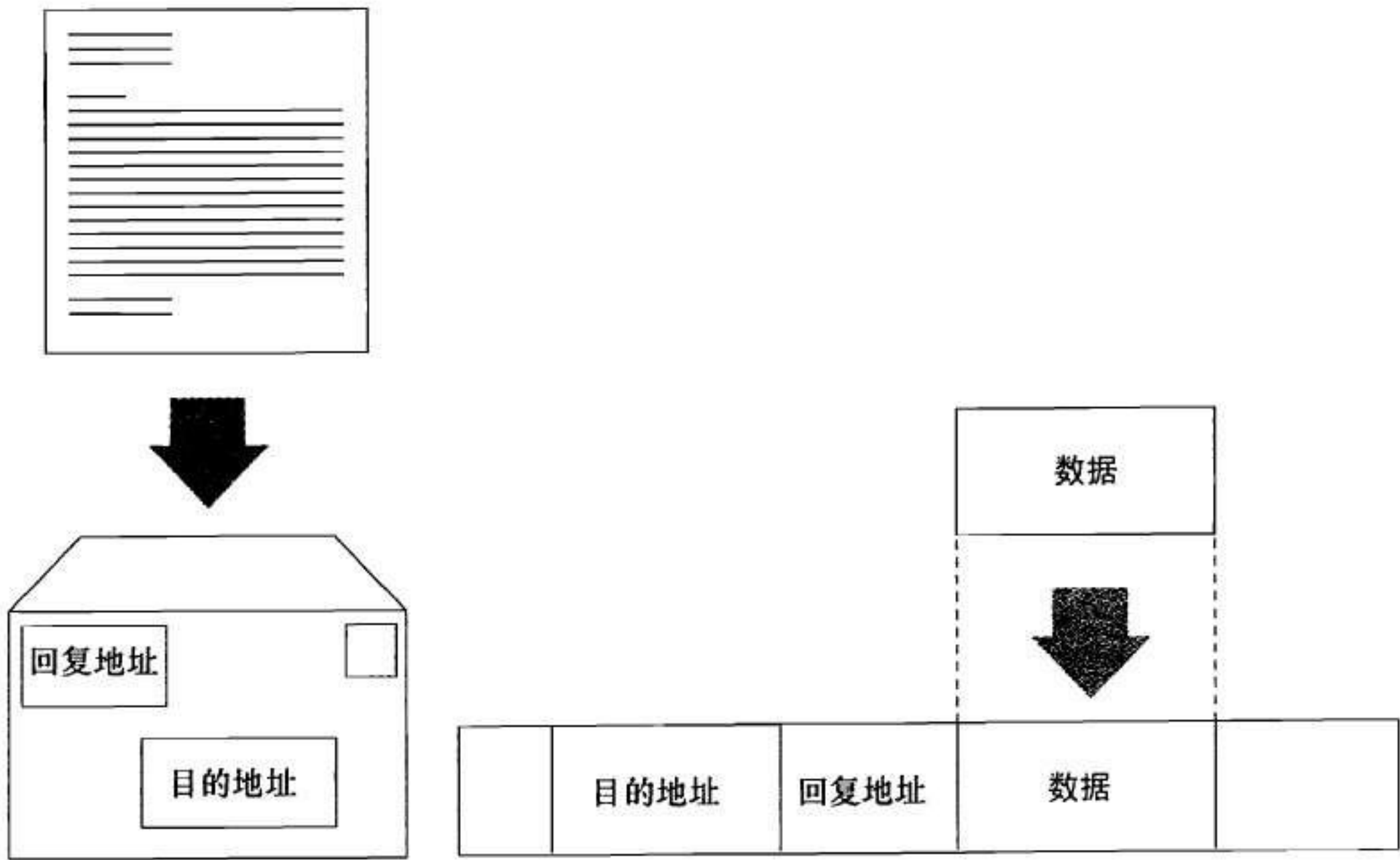


图 1-1 封装就是将数据放入数据帧中——一种用于传输的数字“信封”。

图 1-2 给出了一些最常用的局域网帧格式。注意，每一种格式中都包含目的地址和源地址。地址的格式依赖于特定的 MAC 协议，但是所有的地址都有相同的用途：唯一地标识帧的目标机器和发送帧的设备。

在局域网最通用的 3 种数据链路是以太网、令牌环和 FDDI。虽然 3 种类型各不相同，但是它们都共用一种通用的网络设备编址格式。这种格式最初被 Xerox’s Palo Alto Research Center (PARC)²标准化，目前由 Institute of Electrical and Electronics Engineers(IEEE)所管理，被称为固化地址³、物理地址或机器地址，更多的是称为 MAC 地址。

如图 1-3 所示，MAC 地址为 48 位，它可以唯一地标识地球上任何设备。大家可能听说过这样的传言，由于不道德的“克隆”公司或是“卡壳”程序的原因，生产出大批量固化地址相同的网络接口卡。虽然大多数这类故事仅是传言而已，但是可以想象，如果在局域网所有设备的 MAC 地址都相同将会发生什么：设想在一个城镇中所有居民都叫 Wessvick Smackley，男人、女人、孩子、狗和小猫的名字都是 Wessvick Smackley，那么人们的日常交流将会多么困难，更别提其他事务了。⁴

¹ 后面会看到，建立一个数据链路层的帧就像将一个信封放入一个更大的信封一样。
² 在目前的一些文章中，PARC 的全名是 The Now Famous Xerox PARC。
³ 地址通常被永久地编程或固化到网络接口的 ROM 内。
⁴ 在现实生活中，网络上 MAC 地址相同的情况最有可能由网络管理员使用局部可管理地址引起。这种情况在令牌环网络上十分普遍，以至于在令牌环插入过程中有专门的一步来进行地址冲突检查。

Ethernet

PREAMBLE	目的地址	源地址	类型	数据	帧校验 序列号
----------	------	-----	----	----	------------

IEEE 802.3

PREAMBLE	目的地址	源地址	长度	数据	帧校验 序列号
----------	------	-----	----	----	------------

IEEE 802.5/TOKEN RING

S D	A C	F C	目的地址	源地址	数据	帧校验 序列号	E D
--------	--------	--------	------	-----	----	------------	--------

FDDI

PREAMBLE	S D	F C	目的地址	源地址	数据	帧校验 序列号	E D	F S
----------	--------	--------	------	-----	----	------------	--------	--------

SD = 起始分界符

AC = 访问控制

FC = 帧控制

ED = 终止分界符

FS = 帧状态

图 1-2 几种常用的 LAN 数据链路帧格式

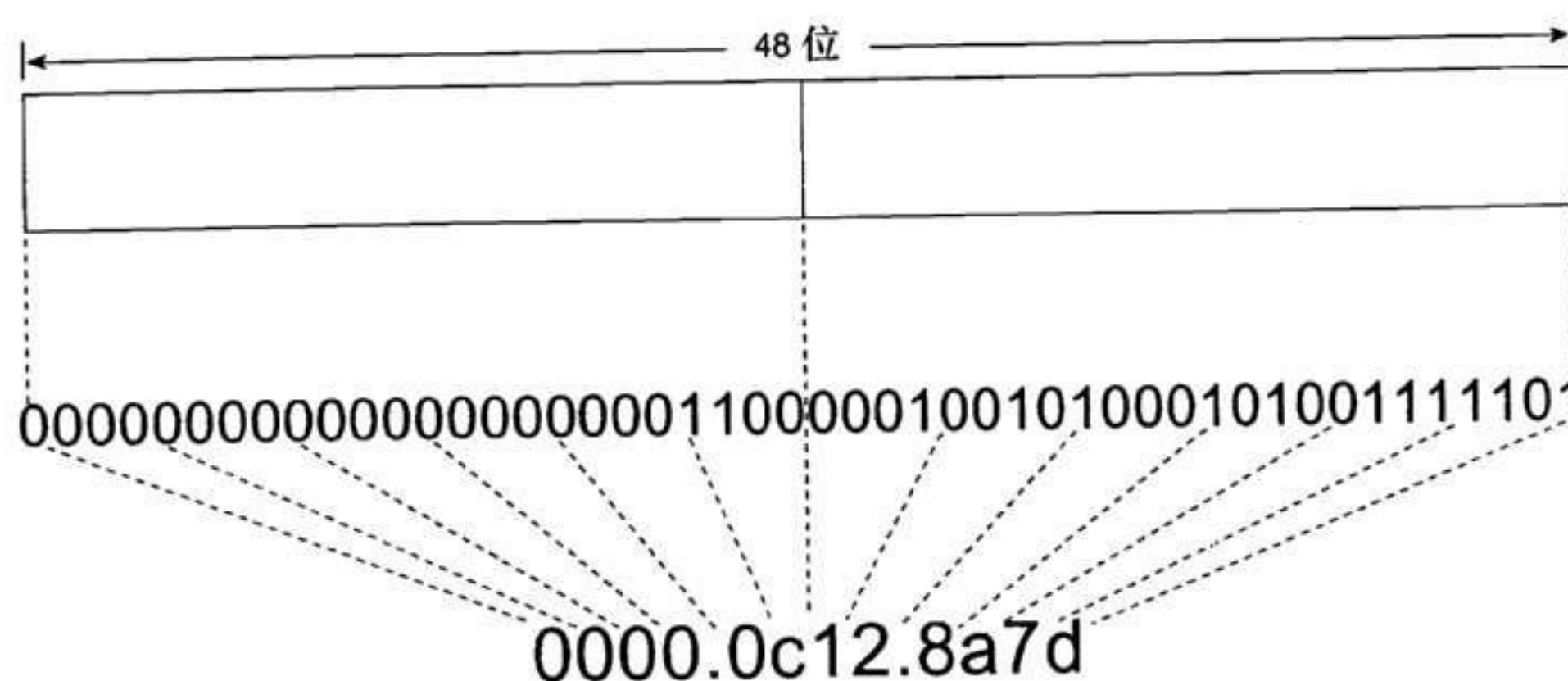


图 1-3 MAC 地址

虽然按照惯例 MAC 地址被称为地址，但 MAC 地址实际上是标识符。因为标识符被固化或永久地分配给设备，它已经成为设备的一部分，并且始终跟随着设备。¹

大多数人一生中都会有多个住址，但是很少会有多个姓名。名字可以标识一个实体——一个人或一台 PC。而地址则是描述一个人或一台 PC 所在的位置。

为了表达清楚，本书使用数据链路标识或 MAC 标识替代 MAC 地址，区分上述名字的原因将在后面说明。

¹ 虽然一些数据链路地址可能或是必须由管理员配置，但是关键之处在于它们是一个标识符，并且是全网唯一的标识符。

1.3 中继器和网桥

到目前为止，上面讲述的内容可以摘要如下：

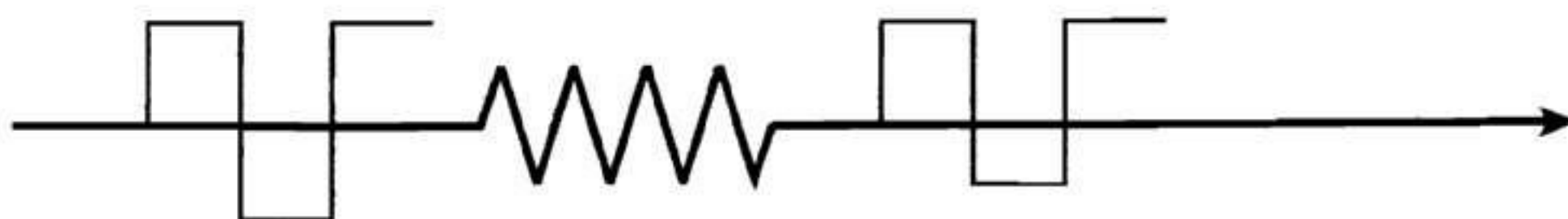
- 数据通信网络由一组通过普通共享介质连接起来的设备组成的。
- 这些设备遵循一套规则，通常叫做介质访问控制，或 MAC，它负责控制设备如何共享介质。
- 每一台设备都有一个标识，每一个标识符唯一地标识一台设备。
- 设备将发送数据封装在名叫帧的虚拟信封中，使用设备标识符进行通信。

因此，局域网是一个极好的资源共享工具，大家都想连接到局域网上。然而，困难也在于此。随着局域网的发展，它自身又出现了新的问题。

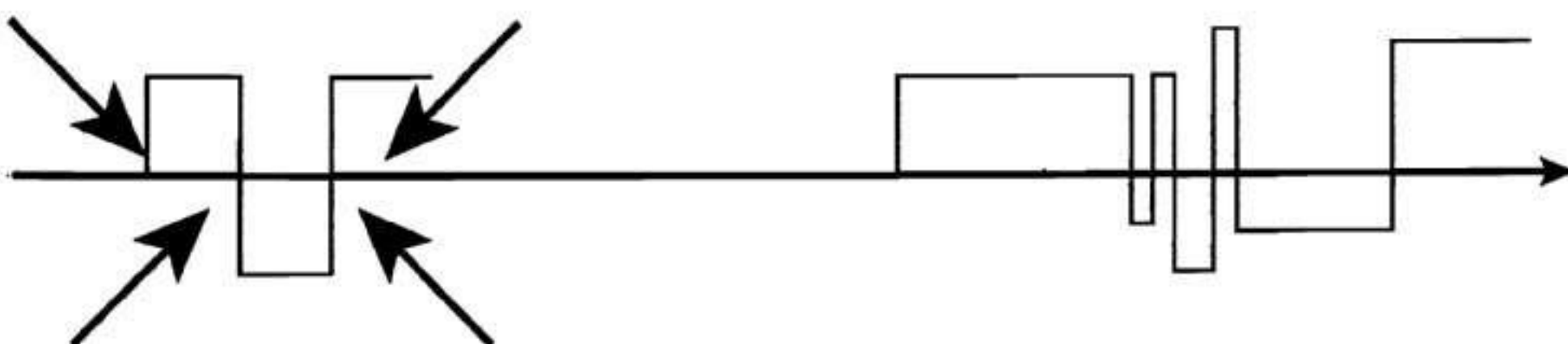
第一个问题是物理距离，图 1-4 给出了 3 个影响电信号的因素。这 3 个因素可能减少或消除信号所表示的信息：

- 衰减
- 干扰
- 失真

(a) 衰减



(b) 噪声



(c) 失真

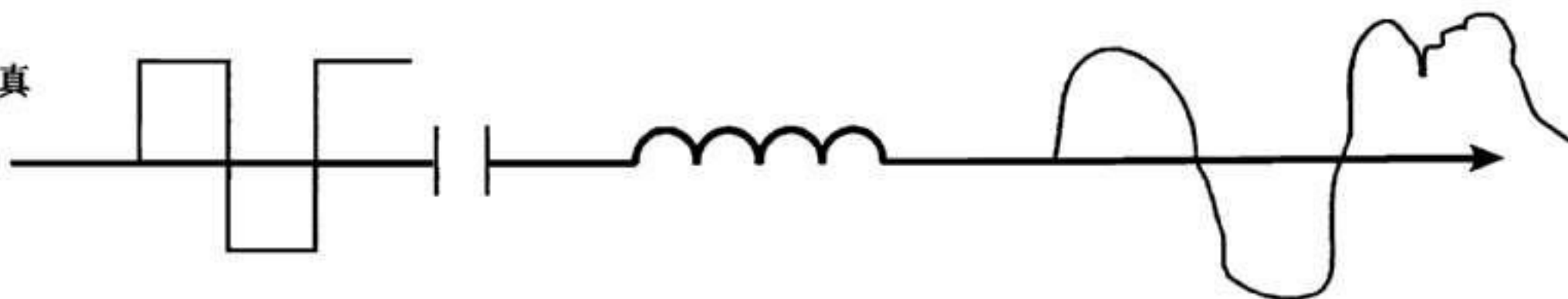


图 1-4 衰减、干扰和失真阻碍了信号在到达时仍然保持发送时的波形。衰减(a)是线路阻抗的函数。信号为克服阻抗需要消耗一部分能量。干扰(b)是外部影响——噪声——的函数，噪声在信号中引入了干扰特性。

失真(c)是线路在不同方式下对信号中不同频率部分的阻抗函数

随着信号沿线路传播距离的增加，这 3 种因素的影响也会减弱。虽然光脉冲沿光纤信道传播时不易受到干扰，但是仍然存在信号的衰减和失真。

每隔一定的间隔需要在线路中加入中继器来减轻超长距离带来的影响。中继器通常放置在距离信号源一定距离的地方，但是要能够正确地识别信号（见图 1-5）。中继器通过清除衰

弱的原始信号,再生成一个新的中继信号,因此叫中继器。

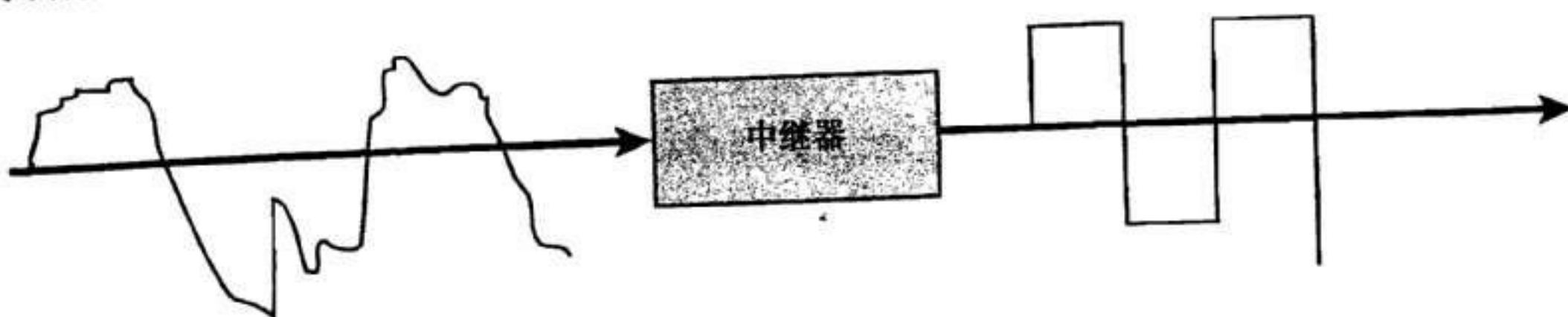


图 1-5 尽管存在衰减,干扰和失真,但是在距离信号源一定距离的地方依然可以对信号进行识别,

通过在这些地方放置中继器,可以再生信号,从而使线路长度加长

中继器可以认为是物理介质的一部分。中继器不带有任何智能,仅仅是再生信号;有时开玩笑地称数字中继器为吐比特唾沫的人。

不断扩大的局域网带来的第二个问题是拥塞。虽然,中继器可以实现线路距离的延伸,网络设备的添加;然而,使用局域网的主要原因是为了资源共享。当过于庞大的群体试图分享有限的资源时,礼貌相处的规则开始被打破,冲突将会爆发。这种情况如果在人类中发生,就可能导致贫穷、罪恶以及战争。在以太网中,冲突则会消耗有效的带宽。在令牌环网和 FDDI 网中,令牌的循环时间和抖动时间则可能会高得惊人。

解决网络过分拥挤的方法就是在局域网设备群体之间划分界限。使用网桥¹可以完成这一任务。

图 1-6 显示了最常用的网桥类型:透明网桥。它执行 3 种简单的功能:学习、转发和过滤。它的透明性在于终端站点不知道网桥的存在。

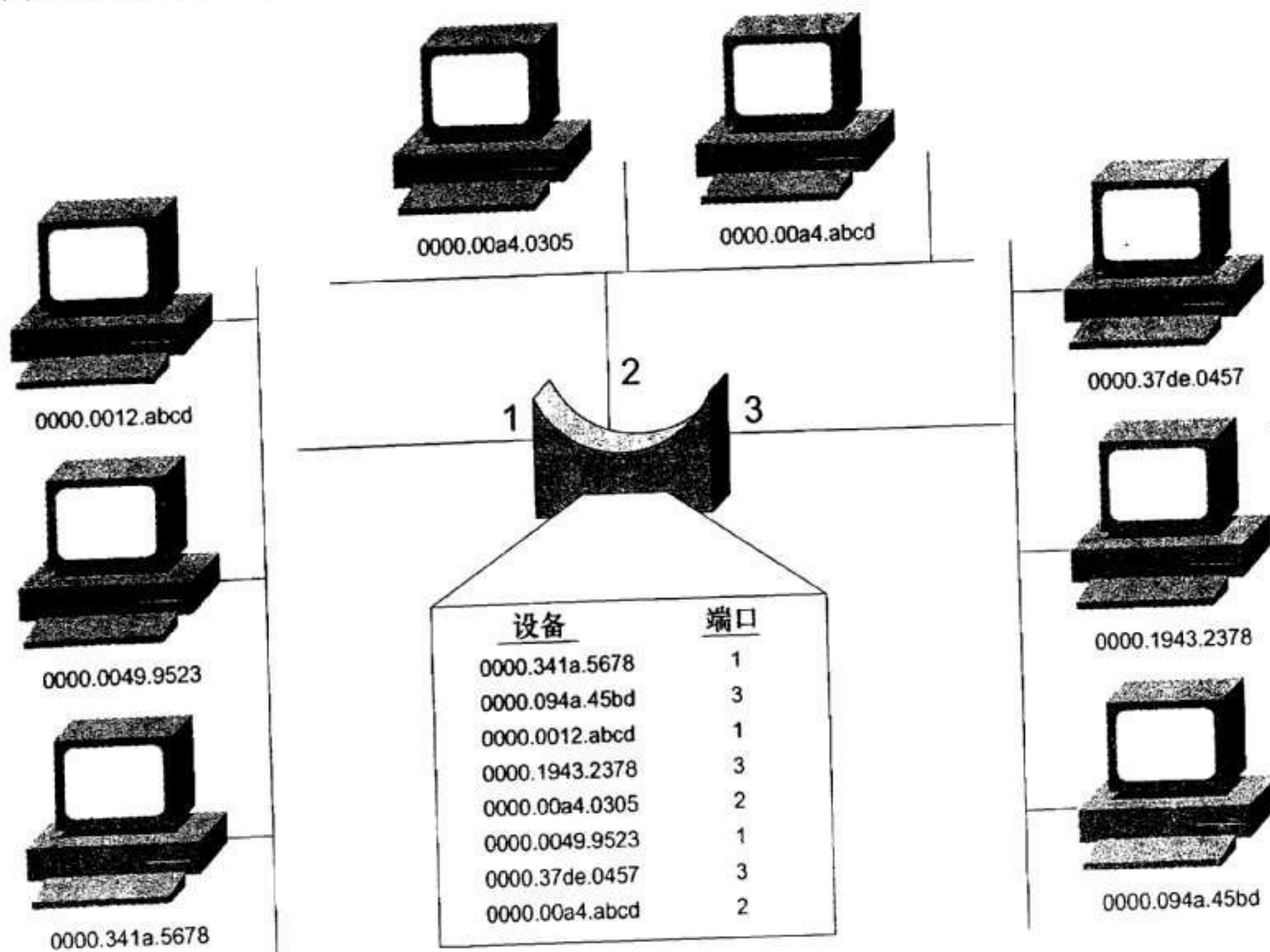


图 1-6 透明网桥将网络设备分割成可管理的群体。桥接表记录了每个群体中的成员,管理着群体之间的通信

¹ 如果撇开那些对现代以太网交换设备和令牌环交换设备进行包装的市场广告宣传,你会发现这些有用的工具只不过是高性能的网桥而已。

网桥是通过无目的地侦听所有端口来学习的。这就是说，每次站点传输一个数据帧时，网桥都去检查帧的源标识。接着，网桥将查到的标识连同它侦听到该帧的端口一起记录在桥接表中。从而网桥可以学习到哪些站点是来自端口1的，哪些是来自端口2的等等。

在图 1-6 中，当一个群体中的成员（例如，与端口 1 相通的站点）试图发送一个数据帧到另一个群体中的成员（例如，与端口 2 相通的站点）时，网桥使用它的桥接表中的信息进行帧的转发。

只具有学习和转发能力的网桥是没有用的。网桥的实际应用是它的第 3 种功能——过滤。图 1-6 显示出：如果一个与端口 2 相通的站点发送数据帧给另一个与端口 2 相通的站点，网桥会检查该帧。网桥查看桥接表，如果与目的设备相通的端口就是收到帧的端口，网桥将不转发该帧。该帧被过滤。

在维持同样带宽的情况下，使用网桥可以比不使用网桥（所有网络设备在单一群体中）允许更多的网络设备添加到网络中去。过滤意味着只有那些需要转发到另一群体中的帧才会被转发，因而节省了资源。使用网桥，以太网被分成了多个冲突域；令牌环和 FDDI 被分成了多个环网。

图 1-7 举例说明了透明网桥的两种应用场合。例子中的网桥是透明的，因为终端站点不知道它。同时，透明网桥也不了解一个网络拓扑结构的真实信息；它只知道从每个端口可以听到哪些标识。

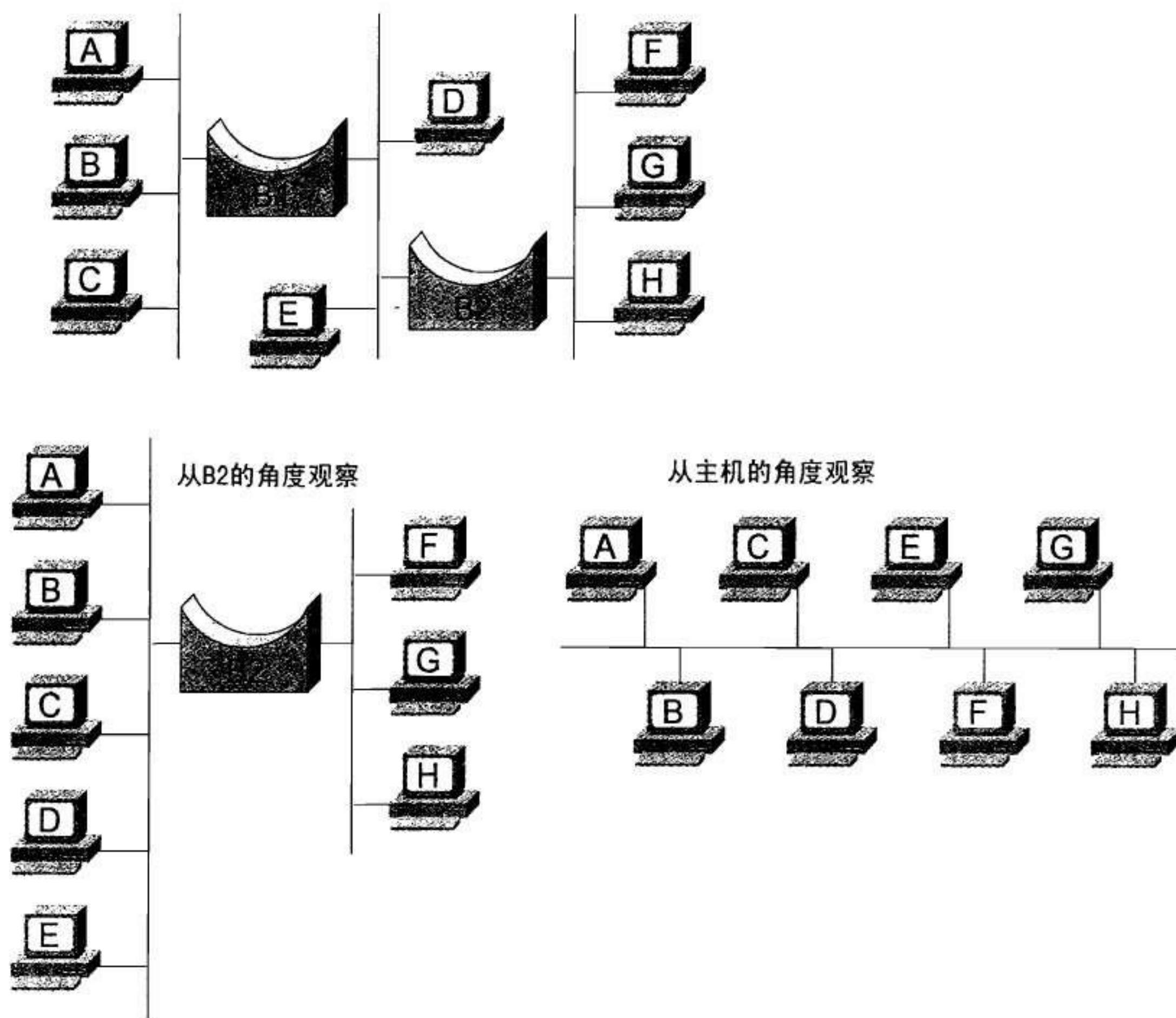


图 1-7 透明网桥的两种应用场合

网桥的一些其他类型包括：源路由网桥、源路由/透明网桥、转换式网桥、封装网桥 (encapsulating bridge)。有关网桥问题以及网桥功能性的完整讨论，参看 Perlman[1992]，在本章末尾的推荐读物列表中被列举。

局域网扩展引起的第 3 个问题是设备放置的地点。中继器可以延伸局域网的距离，但这也只是在一定的地理范围的限制之内。在城市或大于城市范围中扩展局域网意味着用于物理材料上的费用、用于工程结构中的费用以及一些合法的需要缴纳的费用（如破路费）将是无法支付的。像这样的网络距离就需要使用广域网（或称 WAN）技术。¹ 表 1-1 比较了局域网和广域网的异同。

第四个问题是可扩展性。网桥允许将一个网络分割成若干个站点群体；在这种方式中，虽然某些类型的数据帧是不能被局部化的，但是站点与站点之间的通信被局部化了。一些应用要求数据被广播——也就是说，数据必须被发送到网络上的所有站点。以太网、令牌环网和 FDDI 网为广播使用了一个保留的目的标识 (0xFFFF.FFFF.FFFF)。网桥必须在所有的端口上转发广播帧，以确保所有的站点都可以得到一份拷贝。当一个被网桥划分的网络变得越来越大时，越来越多的站点将会产生广播流量；很快，广播帧就会导致网络重新进入拥塞状态。

表 1-1

局域网与广域网的基本不同点

局 域 网	广 域 网
有限的地理范围	地理范围可以是城市或全世界
使用属于自己的并完全可以控制的介质	使用服务提供商提供的专用介质
丰富廉价的带宽	有限昂贵的带宽

为了管理广播流量和应对其他一些规模扩大方面的挑战，需要另一种边界划分方法。虽然网桥可以将网络划分成若干个站点群体，但是我们需要一种方法能够在一个大的网络中创建网络。这种网络的网络，我们称做互连网络 (internetwork)。路由器使互连网络成为可能。

1.4 路 由 器

路由器曾经有过多个名字。让我们回到当今 Internet 被仍叫做 ARPANET 的年代，那时路由器叫 IMP，²即接口报文处理器。近来，路由器更多地被叫做网关；按这种命名方式命名的术语仍然可以在边界网关协议 (BGP) 和内部网关路由选择协议 (IGRP)³中找到。在开放系统互联 (OSI) 领域，路由器被称之为中介系统 (IS)。

所有这些别名都描述了路由器某一方面功能。作为接口报文处理器，暗示了路由器在不同网络之间交换数据报文或分组的功能。作为网关，路由器是发送数据到达其他网络的网关。作为中介系统，路由器是端系统与端系统之间进行数据传送的中介。

作为名称，路由器可能是对现在这些设备所完成功能的最好描述。路由器沿着两个网

¹ 第 3 个术语，就是城域网，或称 MAN。该术语正逐渐地不再被使用。幸好该术语即将消失，它模糊了 LAN 和 WAN 之间的差异。一个 MAN 到底是一个大的 LAN 还是一个小的 WAN 呢？消失确实是一个不好的双关语，它还表明网桥可以确保 MAN 不会成为孤岛。

² 现代分组交换网的前身是 AlohaNet，它是 Norman Abramson 于 20 世纪 60 年代末在夏威夷大学创建的。因为在那时路由器被叫做 IMP，所以 Abramson 故意地把他的路由器 Menehune 命名为：夏威夷矮子。

³ 网关目前通常指应用层的网关，相对于路由器来说，它是一个网络层网关。

络之间的一条路由（路径）发送信息。这条路径可能经过一个路由器或者多个路由器。此外，在互联网络上还存在同时有多条路径到达相同目的地，现代的路由器是使用一组过程来确定并使用其中的最优路由。假如当某条路由不再是最优路由或者完全不可用时，路由器会选择下一条最优路由。路由器使用路由选择协议来确定和选择路由，并且与其他路由器共享网络可达信息和状态。

正像数据链路可以直接连接两个设备，路由器也可以在两个设备之间建立连接。如图 1-8 所示，在不同的网络中，路由器所提供通信路径是一条高层、逻辑的路径。相反，在两个共享公共数据链路的设备之间的通信路径是一条物理路径。

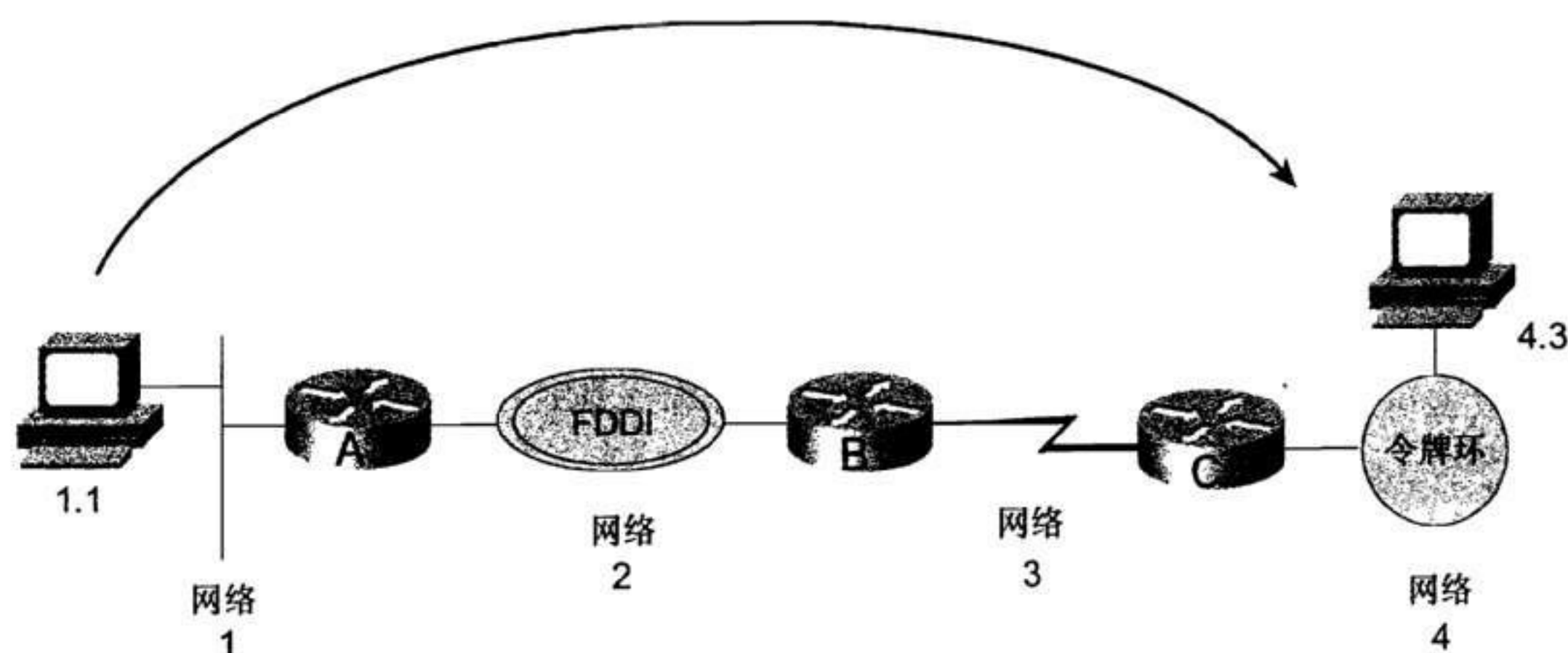


图 1-8 路由器在网络之间建立逻辑路径

这个概念对于理解路由器的功能是极其重要的。注意，在图 1-8 中，设备之间的逻辑路径（或路由）经过了下面几种类型的数据链路：以太网、FDDI 环、串行链路和令牌环。如前面谈到的，为了在数据链路的物理路径上传输数据，必须将数据封装在帧结构（一种数字信封）中。同样，当数据沿逻辑路径穿过可路由的互联网络时，也必须将数据封装在路由器所使用的数字信封中，这里叫报文。

如上所述，每一种数据链路都有它自己特殊的帧格式。图 1-8 中描述的互联网络路由虽然跨过多种数据链路，但是报文从头到尾保持不变。

这是如何成为可能的呢？图 1-9 显示了报文实际上是如何沿路由被传递的。

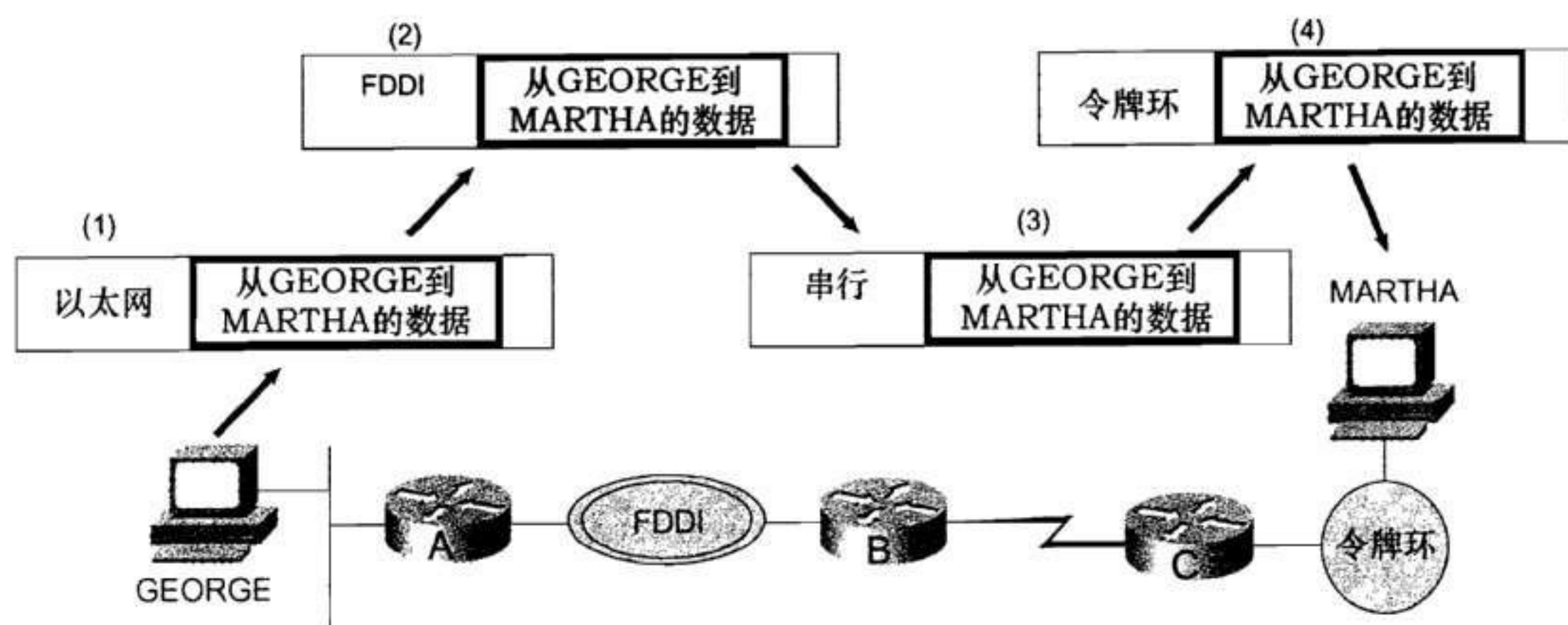


图 1-9 从一种数据链路到达另一种数据链路，帧格式需要发生变化，但是在跨越整个逻辑路径时，报文却保持不变

- 起始主机将数据封装在报文中进行传输。由于需要沿数据链路将报文传递给本地路由器¹——主机的缺省网关，所以主机要将报文再封装到帧中。这一操作等同于将一个信封放到一个更大的信封，例如，就像将一个装有信件的信封放入联邦快递公司的信封。帧的目的数据链路标识是路由器接口的标识，帧的源数据链路标识是主机。
- 路由器 A（图 1-9 中的路由器 A）将报文剥离以太帧；因为路由器知道沿路径的下一跳路由器是 B，并且可以通过它的 FDDI 接口到达，所以路由器 A 将报文封装在 FDDI 帧中。这时，帧的目的标识是路由器 B 的 FDDI 接口的标识，源标识是路由器 A 的 FDDI 接口标识。
- 路由器 B 将报文剥离 FDDI 帧，由于路由器知道沿路径的下一跳路由器是 C，并且可以沿串行链路到达，所以路由器将报文封装在正常的串行链路帧中，并向路由器 C 发送。
- 路由器 C 将报文剥离串行链路帧，并识别出目的站点是通过令牌环网直接连接在路由器上的；路由器 C 将报文封装在令牌环帧中，并把帧的目的标识和源标识分别设置为目的站点和路由器令牌环接口标识。然后发送报文。

理解全部过程的关键是注意帧及相关的数据链路标识，数据链路标识仅仅与各网络相关，并且报文每经过一个网络，它都发生变化。而报文从头到尾都保持不变。

除此之外，起始主机如何知道报文需要被传递到主机的缺省网关？路由器如何知道需要向哪里发送报文？

1.5 网络地址

对于直接连接到局域网上的设备，它们需要通过数据链路标识符被唯一地标识。如果建立一个可路由的互连网络——由多个网络组成的网络，那么每个网络成员都必须被唯一地标识。

大多数讨论可路由互连网络的基础规范都提到：为了保证路由器准确地传递报文到达它们的正确目的地，必须唯一地标识每一个网络或数据链路。这种唯一标识性就是网络地址的用途。

图 1-10 提出了一类网络地址。请注意，每个网络都有它自己的唯一地址。点到点串行链路同样也是。一个初学者的通病是忘记串行链路也是一个网络，而且也需要地址来完成路由。

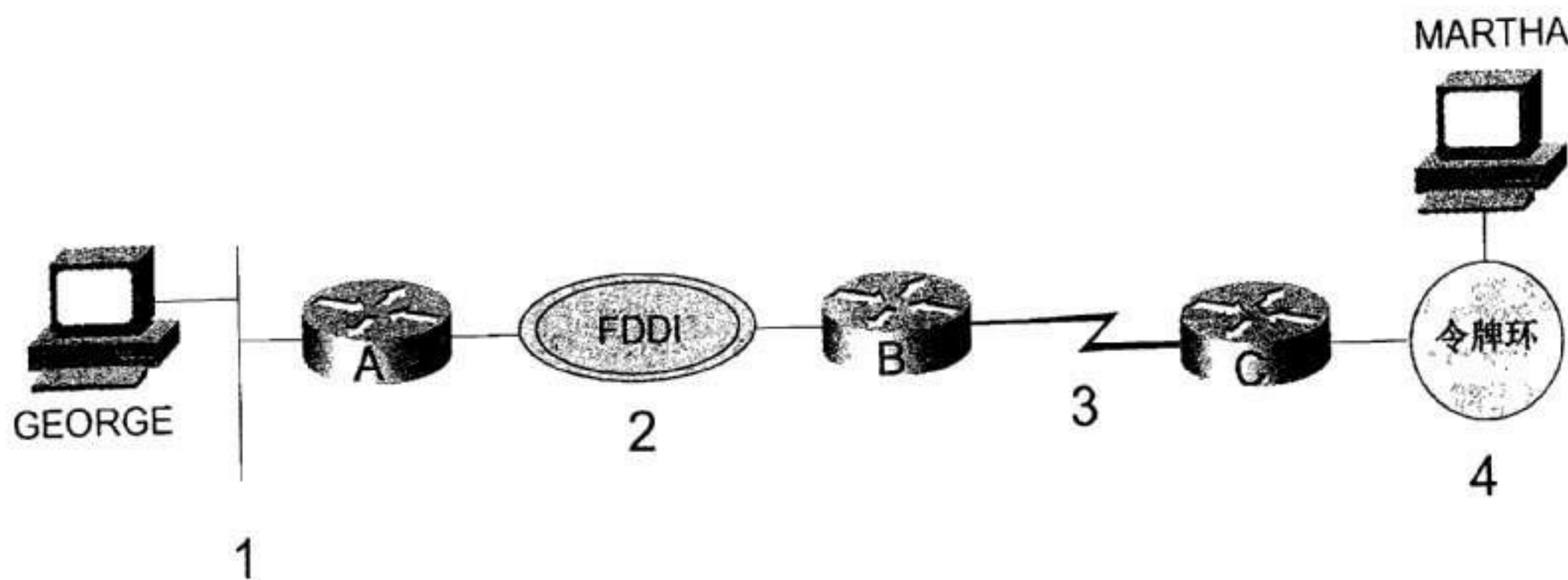


图 1-10 每一个网络都必须有一个唯一的标识地址

¹ 虽然由路由器来建立数据链路（网络）之间的通路，但是路由器也需要遵守所连接网络的协议规则。因此，连接到以太网上的路由器接口将具有 MAC 标识，并且遵守 CSMA/CD 规则，路由器的令牌环接口必须遵守令牌环规则，等等。换言之，路由器不仅仅是一个路由器，还是其连接网络的一个站点。

现在可以回答上一节结尾提出的两个问题中的一个：路由器之所以可传递报文是因为起始主机在报文中写入了目的地址。从路由器的角度看，目的地址是必要的。通常，路由器真正关心的是每个网络的位置。单独的设备与路由器是不相关的；路由器仅需要将报文传送到正确的目的网络。当报文到达目的网络时，路由器将使用数据链路标识把报文传递给网络上的单个设备。

路由器如何处理目的地址是很重要的，而且路由器还担负着中继的作用。路由器的用途就是传递报文至正确的目的网络。同样，路由器惟一关心的个体设备是其他路由器。当路由器发现报文的目的地址属于直连网络，它将担当本网络的一个站点，并且使用目的设备的数据链路标识在网络上进行报文（封装在帧中）传送。¹

随着对路由器和网络地址关系的理解，一个新问题会出现：当路由器看到报文的目的地址属于一个直连网络时，路由器如何知道该向哪里传送报文？毕竟在图 1-10 中起始主机没有提及目的站点的数据链路标识。

在本节结尾提出一个相关问题：起始主机如何知道报文需要被传递到路由选择的缺省网关呢？使用图 1-10 所示的网络地址来回答这两个问题显然是不够的。在一个网络上必须再一次惟一地标识每一个设备，这一次是作为特定网络的成员。网络地址必须包括网络标识和主机标识（图 1-11）。起始主机必须能够识别自身和其他主机的网络地址，实际上是：“我需要发送报文到设备 4.3，我的网络地址是 1.2；因此，我知道目的主机和我的主机位于不同的网络，我需要发送报文到本地路由器。”

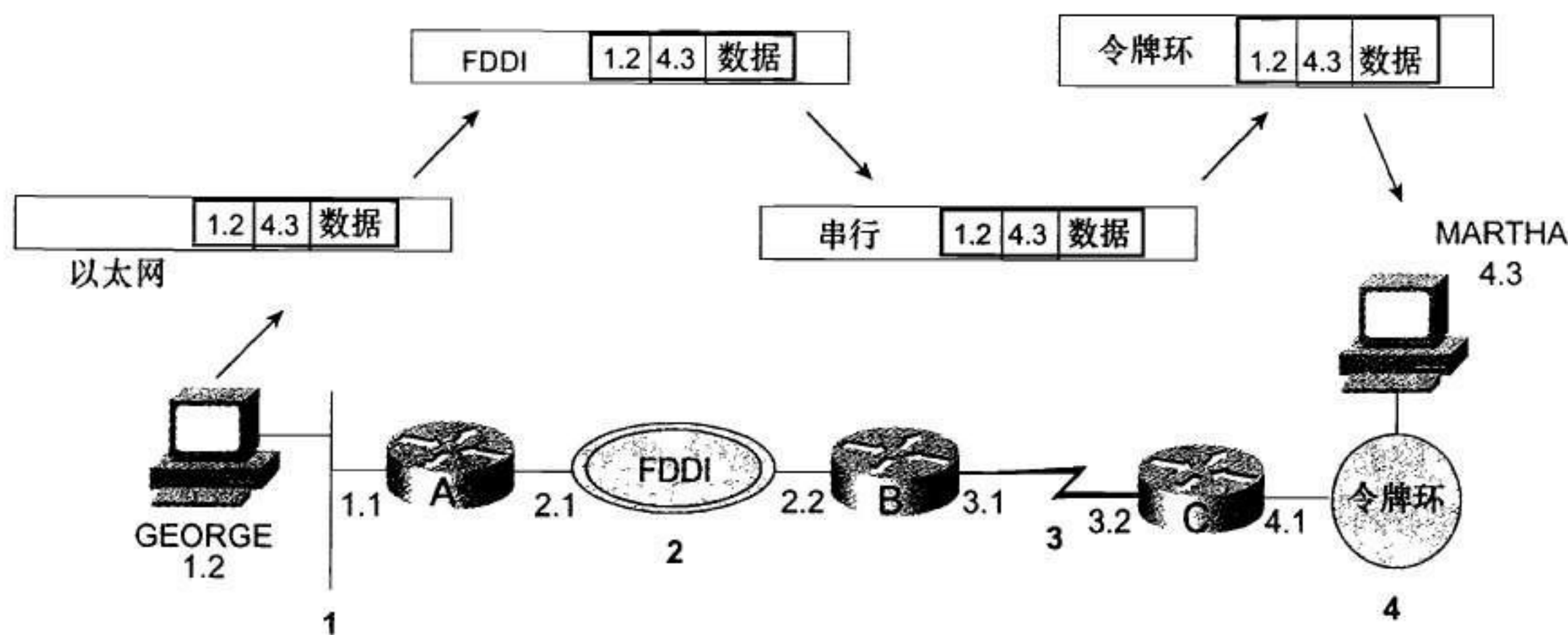


图 1-11 每一个网络都必须有一个惟一的标识地址

同样，路由器 C 必须能够识别，“已经收到目的地址为 4.3 的报文。因为我的令牌环接口的网络地址是 4.1，所以网络 4 是我的直连网络。作为网络 4 的成员，我知道站点 4.3 的 MAC 标识是 0000.2354.AC6B；我仅需要将报文放入令牌环帧中并发送它即可。”

1.6 展 望

本章讲述了以下概念：一个网络地址必须包括网络和主机部分，并且需要某种机制实现

¹ 应当指出的是，存在一种指向特定设备的路由——主机路由。本书后面将介绍。而在这里，主机路由只会把事情搞乱。

网络地址到数据链路标识的映射。第 2 章“TCP/IP 回顾”，将说明 IP 如何满足以上需求。在第 2 章中还会分析 IP 地址的格式，IP 实现网络到数据链路映射的方法，以及有关 IP 路由过程的一些重要机制。

1.7 参考读物

Perlman, R. *Interconnections: Bridges and Routers*. Reading, Massachusetts: Addison-Wesley; 1992.

作者 Radia Perlman 是互联网络界的天才之一，她写的这本书是一部杰作。这本书不仅是一本优秀的基础教科书，而且当 Perlman 讨论围绕着标准制订的政治观点时，她的讽刺性讲解也不容错过。

1.8 复习题

1. 局域网的用途是什么？
2. 什么是协议？
3. MAC 协议的用途是什么？
4. 什么是帧？
5. 所有帧类型的共同特性是什么？
6. 什么是 MAC 地址或 MAC 标识？
7. 为什么 MAC 地址不是真正的地址？
8. 在数据链路上，信号衰变的 3 个原因是什么？
9. 中继器的用途是什么？
10. 网桥的用途是什么？
11. 透明网桥是如何做到透明的？
12. 局域网和广域网的 3 个基本不同点是什么？
13. 广播 MAC 标识的用途是什么？如何使用十六进制或二进制表示广播 MAC 标识？
14. 网桥和路由器主要的相似点是什么？路由器和网桥主要的不同点是什么？
15. 什么是报文？帧和报文主要的相似点是什么？帧和报文的的不同点是什么？
16. 当一个报文穿过互联网络时，报文源地址发生变化吗？
17. 什么是网络地址？网络地址各部分的用途是什么？
18. 网络地址和数据链路标识的主要不同点是什么？

第 2 章

TCP/IP 回顾

本章包括以下主题：

- TCP/IP 协议层
- IP 报头
- IP 地址
- 地址解析协议（ARP）
- Internet 消息控制协议（ICMP）
- 主机到主机层

本章的目的是研究启动、控制和协助 TCP/IP 进行路由选择的协议细节，对 TCP/IP 协议族将不作深入讨论。本章最后的几本参考读物均对 TCP/IP 进行了详细的讲解，请读者至少阅读其中的一本。

早在 20 世纪 70 年代初期，Vint Cerf 和 Bob Kahn 就对 TCP/IP 及其分层协议框架进行了构思，它的提出先于 ISO 的 OSI 模型。在这一章，我们将会分析多种功能和服务，对 TCP/IP 协议层的简单回顾有助于读者理解它们是如何进行相互关联的。

2.1 TCP/IP 协议层

图 2-1 展示了 TCP/IP 协议族与 OSI 参考模型的相互关系。在 TCP/IP 协议族中，网络接口层对应于 OSI 的物理和数据链路层，但实际上在规范中并不存在这一层。然而，如图 2-1 所示，作为对物理和数据链路层的表示，它已经成为事实上的一个层次。在本节中，我们将使用 OSI 的术语——物理和数据链路层来描述它。

OSI	TCP/IP
应用层	应用层
表示层	
会话层	
传输层	主机到主机层
网络层	互联网络层
数据链路层	网络接口层
物理层	

图 2-1 TCP/IP 协议族

物理层包含了多种与物理介质相关的协议，这些物理介质用以支撑 TCP/IP 通信。物理层的协议按照正式的分类可以分为 4 类，这 4 类涵盖了物理介质的所有方面：

- 电子/光学协议——描述了信号的各种特性。例如，电压或光强度、位定时、编码和信号波形。
- 机械协议——规定了连接器的尺寸或电线的金属成份。
- 功能性协议——描述了做什么。例如，在 EIA-232-D 连接器第 4 管脚上，二进制 1 表示“请求发送”。
- 程序性协议——描述了如何做。例如，在 EIA-232-D 导线上，二进制 1 表示电压小于-3V。

在第 1 章“基本概念：互联网络、路由器和地址”中，讲述了数据链路层。数据链路层包含控制物理层的协议：如何访问和共享介质，怎样标识介质上设备，以及在介质上发送数据之前如何完成数据成帧。典型的数据链路协议有 IEEE 802.3/以太网、IEEE 802.5/令牌环以及 FDDI。

互联网络层与 OSI 的网络层相对应，如图 1-9，通过定义报文格式和地址格式，互联网络层主要负责为经过逻辑互联网络路径的数据进行路由选择。当然，互联网络层也是本书内容涉及最多的一层。

与 OSI 传输层相对应的是主机到主机层，它指定了控制互联网络层的协议，这就像数据链路层控制物理层一样。主机到主机层和数据链路层都定义了流控和差错控制机制。二者不同之处在于，数据链路层协议强调控制数据链路路上的流量，即连接两个设备的物理介质上的流量，而传输层控制逻辑链路路上的流量，即两个设备的端到端连接，这种逻辑连接可能跨越一连串数据链路。

应用层与 OSI 的会话层、表示层、应用层相对应。虽然一些路由协议（诸如 BGP、RIP）也在应用层，但是大多数应用层通用服务向用户应用提供访问网络的接口。

对于图 2-1 中所示的协议族和其他协议族来说，多路复用是一个通用功能。许多应用可以使用主机到主机层的一个服务，同样许多主机到主机层的服务也可以使用互联网络。多个

协议族（如 IP、IPX、AppleTalk）还可以通过数据链路协议共享一条物理链路。

2.2 IP 报文头

图 2-2 给出了 IP 报文头的格式，相应标准见 RFC791。报文中的大多数字段对路径选择都很重要。

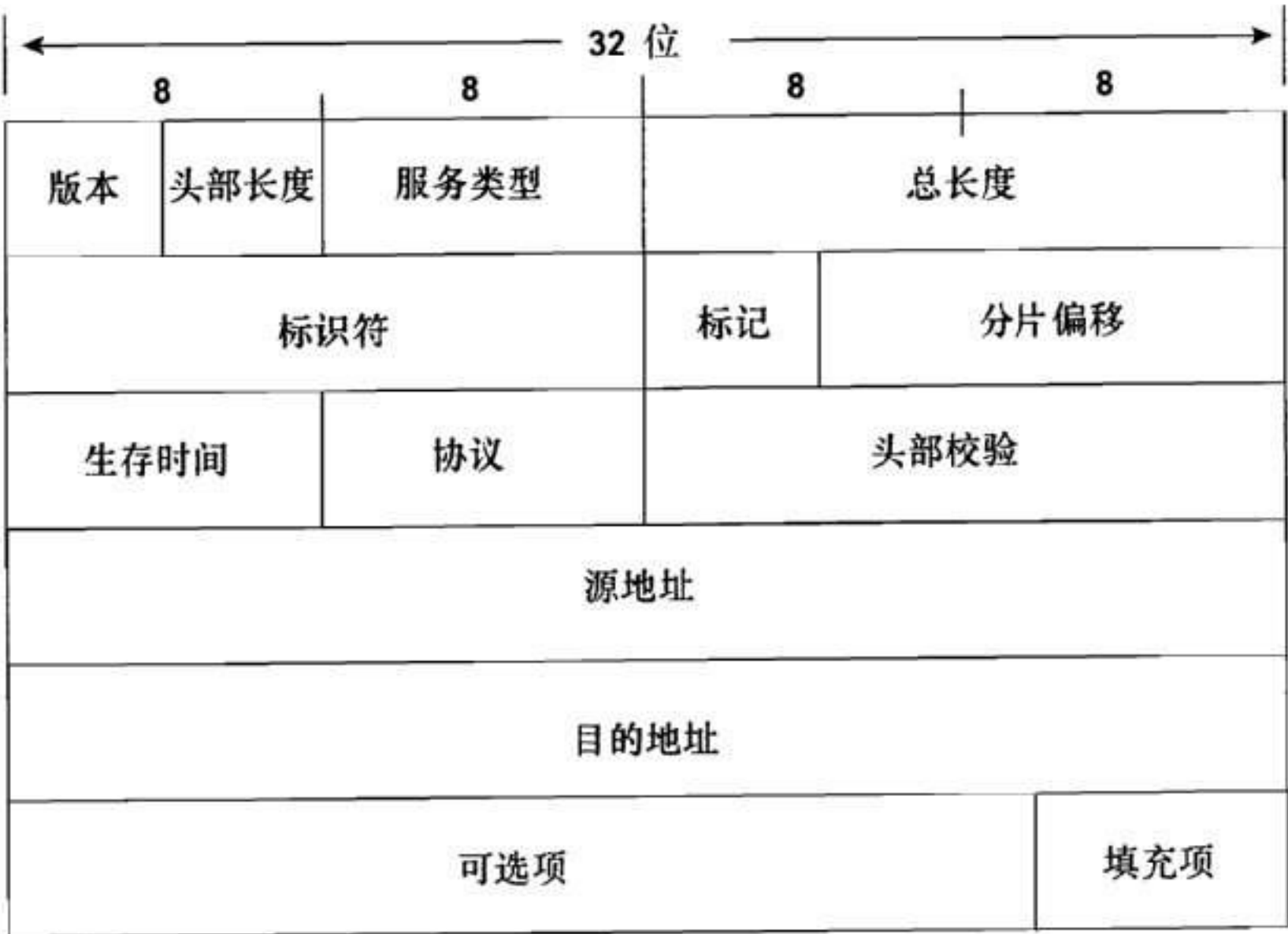


图 2-2 IP 报文协议

- **版本 (Version)** ——标识了报文的 IP 版本号。这个 4 位字段的值通常为二进制 0100；常用的 IP 版本号是 4 (IPv4)。新版的 IP 协议版本号是 6 (IPv6)，但还没有普遍使用，有时又叫做“下一代 IP” (IPng)。所有已分配的现行版本号及相关 RFC 见表 2-1。除 4 和 6（早期提出的简单 Internet 协议，SIP，也使用版本号 6）之外，所有其他版本号仅作为“文化”而存在，感兴趣的读者可以阅读相关的 RFC。
- **报头长度 (header length)** ——字段长度为 4 位，正如字段名所示，它表示 IP 报头的长度。设计报头长度字段的原因是报文的选择项字段（在本节后面部分会被讨论）大小会发生变化。IP 报头最小长度为 20 个 8bit 字节，最大为 24 个 8bit 字节。报头长度字段描述了以 32 比特的字为单位的报头长度，其中 5 表示 IP 报头的最小长度为 160 比特，6 表示最大。

表 2-1 IP 版本号

版本号	版本	RFC
0	保留	
1-3	未分配	
4	Internet 协议 (IP)	791
5	ST 数据报模式	1190
6	简单 Internet 协议 (SIP)	

续表

版本号	版本	RFC
6	IPng	1883
7	TP/LX	1475
8	P Internet 协议 (PIP)	1621
9	使用更大地址的 TCP 和 UDP (TUBA)	1347
10-14	未分配	
15	保留	

- **服务类型 (Type Of Service, TOS)** —— 字段长度为 8 位，它用来指定特殊的报文处理方式。服务类型字段实际上被划分为两个子字段：优先权和 TOS。优先权用来设置报文的优先级，这就像邮寄包裹一样，可以是平信、隔日送到或两日内送到。TOS 允许按照吞吐量、时延、可靠性和费用方式选择传输服务。虽然 TOS 字段通常不用（所有位均被设置为 0），但是在开放式最短路径优先协议 (OSPF) 的早期规范中还是提倡 TOS 路由选择的。在服务质量 (QoS) 应用中有时使用优先权位，图 2-3 简要地说明了 8 个 TOS 比特，更详细的信息可以参见 RFC1340 和 RFC1349。

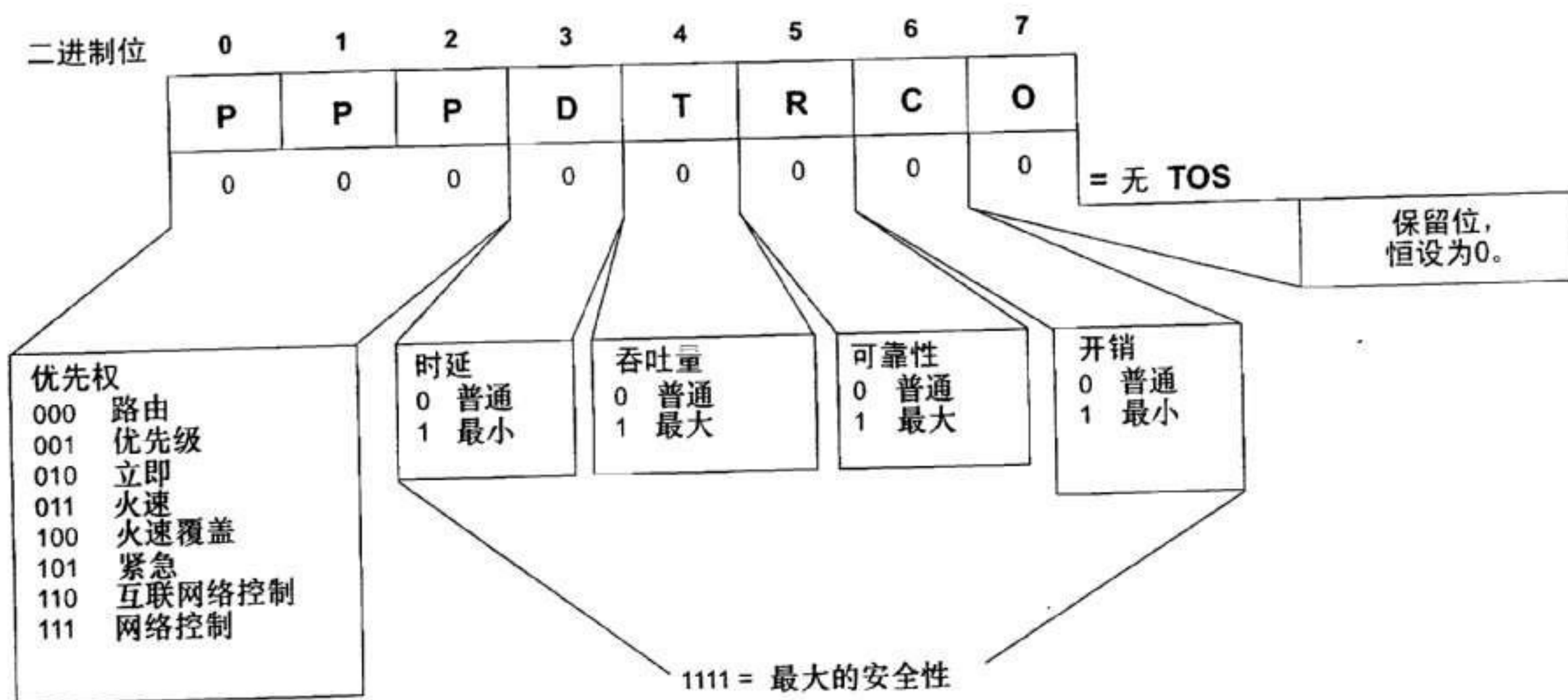


图 2-3 服务类型域

- **总长度 (Total Length)** —— 字段长度为 16 位，它是指整个 IP 报文的长度，以 8bit 字节 (octet) 为单位，其中包括 IP 报头。接收者用 IP 报文总长度减去 IP 报头长度，就可以确定报文数据有效载荷的大小。16 位长的二进制数用十进制表示最大可以为 65 535，所以 IP 报文的最大长度是 65 535。
- **标识符 (Identifier)** —— 字段长度为 16 位，通常与标记字段和分片偏移字段一起用于 IP 报文的分片。如果报文原始长度超过报文所要经过的数据链路的最大传输单位 (MTU)，那么必须将报文分片为更小的报文。例如，一个大小为 5000 字节 (byte) 的报文在穿过互联网络时，如果遇到一条 MTU 为 1500 字节的数据链路，即数据帧最多容纳大小为 1500 字节的报文。那么路由器需要在数据成帧之前将报文分片成多个报文，其中每个报文长度不得超过 1500 个 8bit 字节 (octet)。然后路由器在

每片报文的标识字段上打上相同的标记，以便接收设备可以识别出属于一个报文的分片。¹

- **标记字段 (Flag)**——长度为 3 位，其中第 1 位没有使用。第 2 位是不分片位 (DF)。当 DF 位被设置为 1 时，表示路由器不能对报文进行分片处理。如果报文由于不能被分片而未能被转发，那么路由器将丢弃该报文并向源点发送错误信息。这一功能可以在互联网络上用于测试 MTU 值。如图 2-4，在 Cisco 的路由器上，使用扩展 Ping 工具可以对 DF 进行设置。

```

Handy#ping
Protocol [ip]:
Target IP address: 172.16.113.17
Repeat count [5]: 1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address:
Type of service [0]:
Set DF bit in IP header? [no]: y
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: r
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]: y
Sweep min size [76]: 500
Sweep max size [18024]: 2000
Sweep interval [1]: 500
Type escape sequence to abort.
Sending 4, [500..2000]-byte ICMP Echos to 172.16.113.17, timeout is 2 seconds:
Packet has IP options: Total option bytes= 39, padded length=40
  Record route: <*> 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
                  0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0

Reply to request 0 (16 ms) (size 500). Received packet has options
  Total option bytes= 40, padded length=40
  Record route: 172.16.192.5 172.16.113.18 172.16.113.17 172.16.113.17
                172.16.192.6 172.16.192.5 <*> 0.0.0.0 0.0.0.0 0.0.0.0
  End of list

Reply to request 1 (24 ms) (size 1000). Received packet has options
  Total option bytes= 40, padded length=40
  Record route: 172.16.192.5 172.16.113.18 172.16.113.17 172.16.113.17
                172.16.192.6 172.16.192.5 <*> 0.0.0.0 0.0.0.0 0.0.0.0
  End of list

Reply to request 2 (28 ms) (size 1500). Received packet has options
  Total option bytes= 40, padded length=40
  Record route: 172.16.192.5 172.16.113.18 172.16.113.17 172.16.113.17
                172.16.192.6 172.16.192.5 <*> 0.0.0.0 0.0.0.0 0.0.0.0
  End of list

Unreachable from 172.16.192.6, maximum MTU 1478 (size 2000).
  Received packet has options
  Total option bytes= 39, padded length=40
  Record route: <*> 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
                0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0

Success rate is 75 percent (3/4), round-trip min/avg/max = 16/22/28 ms
Handy#

```

图 2-4 为了测试穿越互联网络的 MTU 值，Cisco 的扩展 Ping 工具允许设置 DF 位。在图中，到目的 172.16.113.17 路径的最大 MTU 为 1478 个 8bit 字节

第 3 位表示还有后继分片 (MF)，当路由器对报文进行分片时，除了最后一个分片的 MF 位设置为 0 外，其他所有分片的 MF 位均设置为 1，以便接收者直到收到 MF 位为 0 的分片为止。

¹ 被分片的报文不会在数据链路的另一端被重组，而是一直保持分片状态，直至到达最终目的地时才会被重组。

- **分片偏移 (Fragment Offset)** —— 字段长度为 13 位, 以 8 个 8bit 字节为单位, 用于指明分片起始点相对于报头起始点的偏移量¹。由于分片到达时可能错序, 所以分片偏移字段可以使接收者按照正确的顺序重组报文。

注意: 如果一个分片在传输中丢失, 那么必须在网络中同一点对整个报文重新分片并重新发送。因此, 容易发生故障的数据链路会造成时延不成比例。另外, 如果由于网络拥塞而造成分片丢失, 那么重传整组分片会进一步加重网络拥塞。

- **生存时间 (Time To Live, TTL)** —— 字段长度为 8 位, 在最初创建报文时 TTL 即被设置为某个特定值。当报文逐个沿路由器被传输时, 每个路由器都会降低 TTL 的数值。当 TTL 值减为 0 时, 路由器将会丢弃该报文并向源点发送错误信息。这个方法可以防止报文在互联网上无休止地被传输。

按照最初构想, TTL 值以 s (秒) 为单位。如果报文在路由器上被延迟的时间超过 1s, 路由器将会相应地调整 TTL 值。然而, 这种方法实施起来十分困难, 因而也很少被支持。大部分路由器不管实际时延是多少, 统统将 TTL 值减 1, 所以 TTL 实际上是表示跳数。虽然 TTL 的常用值为 15 和 32, 但是建议的缺省值是 64。

一些追踪工具, 如 Cisco 的 **trace** 命令, 使用 TTL 字段。如果路由器被告知需要追踪到达主机地址为 10.11.12.13 的路径, 路由器将发送 3 个报文, 其中 TTL 值被设置为 1; 第 1 个路由器将会把 TTL 值减少到 0, 而且在丢弃报文的同时向源点发送错误信息。源点路由器通过阅读错误信息从而得知发送错误信息的路由器即为路径上的第 1 个路由器。再一次被路由器发送的 3 个报文的 TTL 值被设置为 2。第 1 个路由器将 TTL 值减 1, 第 2 个路由器将 TTL 值再减 1 后为 0, 此时源点路由器将会接收到第 2 个路由器发送来的错误信息。第 3 次发送的报文 TTL 值被设置为 3, 依此类推, 直到目的地被发现。最终, 沿途所有的路由器都会被标识出来。图 2-5 给出了在 Cisco 路由器上的命令的输出结果。

```
ELVIS#trace www.cisco.com
Type escape sequence to abort.
Tracing the route to cio-sys.Cisco.COM (192.31.7.130)

 1 172.18.197.17 4 msec 4 msec
 2 ltlrichard-s1-13.hwy51.com (172.18.197.1) 36 msec 44msec 2536 msec
 3 cperkins-rtf-fr2.hwy51.com(10.168.204.3) 104 msec 60 msec *
 4 cberry.hwy51.com (10.168.193.1) 92 msec *
 5 jllewis-inner.hwy51.com (10.168.207.59) 44 msec * 44 msec
 6 bholly-fw-outer-rt.hwy51.com (10.168.207.94) 44 msec * 48 msec
 7 sl-stk-14-S10/0:6-512k.sprintlink.net (144.228.214.107) 92 msec *
 8 sl-stk-2-F1/0/0.sprintlink.net (144.228.40.2) 52 msec 1156 msec *
 9 sl-mae-w-H1/0-T3.sprintlink.net (144.228.10.46) 100 msec 124 msec 2340 msec
10 sanjose1-br1.bbnplanet.net (198.32.136.19) 2264 msec 164 msec *
11 paloalto-br2.bbnplanet.net (4.0.1.10) 64 msec 60 msec *
12 su-pr2.bbnplanet.net (131.119.0.218) 76 msec 76 msec 76 msec
13 cisco.bbnplanet.net (131.119.26.10) 2560 msec 76 msec 936 msec
14 sty.cisco.com (192.31.7.39) 84 msec 72 msec *
15 cio-sys.Cisco.COM (192.31.7.130) 60 Msec * 64 msec
ELVIS#
```

图 2-5 追踪工具使用 TTL 字段来标识沿途路由器。星号表示超时报文

- **协议 (Protocol)** —— 字段长度为 8 位, 它给出了主机到主机层或传输层协议的“地址”或协议号, 协议字段指定了报文中信息的类型。当前已分配了 100 多个不同的协议号, 表 2-2 给出了其中一些较常用的协议号。

¹ 为了使 13 位长的分片偏移字段可以表示的最大报文长度为 65 535 字节, 所以使用 8 个 8bit 字节作为本字段的单位。

表 2-2

一些众所周知的协议号

协 议 号	主机到主机层协议
1	Internet 消息控制协议 (ICMP)
2	Internet 组管理协议 (IGMP)
3	网关到网关协议 (GGP)
4	被 IP 协议封装的 IP
6	传输控制协议 (TCP)
8	外部网关协议 (EGP)
17	用户数据报协议 (UDP)
35	域间策略路由选择协议 (IDPR)
45	域间路由选择协议 (IDRP)
46	资源预留协议 (RSVP)
47	通用路由选择封装 (GRE)
54	NBMA 下一跳解析协议 (NHRP)
88	Cisco Internet 网关路由选择协议 (IGRP)
89	开放式最短路径优先 (OSPF)

- **报头校验和 (Header Checksum)** ——是针对 IP 报头的纠错字段。校验和的计算不使用被封装的数据内容，UDP、TCP 和 ICMP 都有各自的校验和。报头校验和字段包含一个 16 位二进制补码和，这是由报文发送者计算得到的。接收者将连同原始校验和重新进行 16 位二进制补码和计算。如果报文传输中没有发生错误，那么结果应该 16 位全部为 1。回忆前面所述内容，由于每个路由器都会降低报文的 TTL 值，所以每个路由器都必须重新计算校验和。RFC1141 讨论了一些简化计算的策略。
- **源地址和目的地址 (Source and Destination Address)** ——字段长度为 32 位，分别表示发送报文源点和目的地的 IP 地址。IP 地址的格式将会在下一节“IP 地址”中讨论。
- **可选项 (Options)** ——是一个变长字段，并且是可选的。可选项被添加在报头中，包括源点产生的信息和其他路由器加入的信息；可选项字段主要用于测试。常用的可选项如下：
 - **松散源选路选项 (Loose Source Routing)** ——它给出了一连串路由器接口的 IP 地址序列。报文必须沿着 IP 地址序列传送，但是允许在相继的两个地址之间跳过多个路由器。
 - **严格源选路选项 (Strict Source Routing)** ——它也给出了一系列路由器接口的 IP 地址序列。不同于松散源选路，报文必要严格按照路由转发。如果下一跳不再列表中，那么将会发生错误。
 - **记录路由选项 (Record Route)** ——当报文离开时为每个路由器提供空间记录报文的出站接口地址，以便保存报文经过的所有路由器的记录。记录路由选项提供了类似于路由追踪的功能，但是不同点在于这里记录了双向路径上的出站接口信息。
 - **时间戳选项 (Timestamp)** ——除了每个路由器还会记录一个时间戳之外，时间戳选项十分类似于记录路由选项，这样报文不仅可以知道自己到过哪里，而且还可以记录下到达的时间。

使用 Cisco 路由器上的扩展 Ping 工具可以调用所有这些选项。图 2-4 中使用了记录路由选项，图 2-6 使用了松散源选路和时间戳选项，严格源选路选项在图 2-7 中被使用。

```

Handy#ping
Protocol [ip]:
Target IP address: 172.16.113.9
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address:
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: 1
Source route: 172.16.113.14 172.16.113.10
Loose, Strict, Record, Timestamp, Verbose[LV]: t
Number of timestamps [ 6 ]: 2
Loose, Strict, Record, Timestamp, Verbose[LTV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.113.9, timeout is 2 seconds:
Packet has IP options: Total option bytes= 23, padded length=24
  Loose source route: <*> 172.16.113.14 172.16.113.10
  Timestamp: Type 0. Overflows: 0 length 12, ptr 5
    >>Current pointer<<
    Time= 0
    Time= 0

Request 0 timed out
Reply to request 1 (76 ms). Received packet has options
  Total option bytes= 24, padded length=24
  Loose source route: 172.16.113.13 172.16.192.6 <*>
  Timestamp: Type 0. Overflows: 6 length 12, ptr 13
    Time= 80FF4798
    Time= 80FF4750
    >>Current pointer<<
  End of list

Request 2 timed out
Reply to request 3 (76 ms). Received packet has options
  Total option bytes= 24, padded length=24
  Loose source route: 172.16.113.13 172.16.192.6 <*>
  Timestamp: Type 0. Overflows: 6 length 12, ptr 13
    Time= 80FF4FC0
    Time= 80FF4F78
    >>Current pointer<<
  End of list

Request 4 timed out
Success rate is 40 percent (2/5), round-trip min/avg/max = 76/76/76 ms
Handy#

```

图 2-6 可以使用 Cisco 的扩展 Ping 工具来设置 IP 报头中的选择字段的各项参数。在这个例子中，用到了松散源选路选项和时间戳选项

- **填充 (Padding)** ——该字段通过在选择字段后面添加 0 来补足 32 位，这样保证报头长度是 32 比特的倍数。图 2-8 显示了协议分析器捕获到的 IP 报头的信息。与图 2-2 中的信息作一下比较。


```

Handy#ping
Protocol [ip]:
Target IP address: 172.16.113.10
Repeat count [5]: 2
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address:
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: s
Source route: 172.16.192.6 172.16.113.17 172.16.113.10
Loose, Strict, Record, Timestamp, Verbose[SV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 2, 100-byte ICMP Echos to 172.16.113.10, timeout is 2 seconds:
Packet has IP options: Total option bytes= 15, padded length=16
  Strict source route: <*> 172.16.192.6 172.16.113.17 172.16.113.10

Reply to request 0 (80 ms). Received packet has options
  Total option bytes= 16, padded length=16
  Strict source route: 172.16.113.10 172.16.113.17 172.16.192.6 <*>
  End of list

Reply to request 1 (76 ms). Received packet has options
  Total option bytes= 16, padded length=16
  Strict source route: 172.16.113.10 172.16.113.17 172.16.192.6 <*>
  End of list

Success rate is 100 percent (2/2), round-trip min/avg/max = 76/78/80 ms
Handy#

```

图 2-7 这里使用扩展 ping 在 ping 报文中设置严格源选路选项

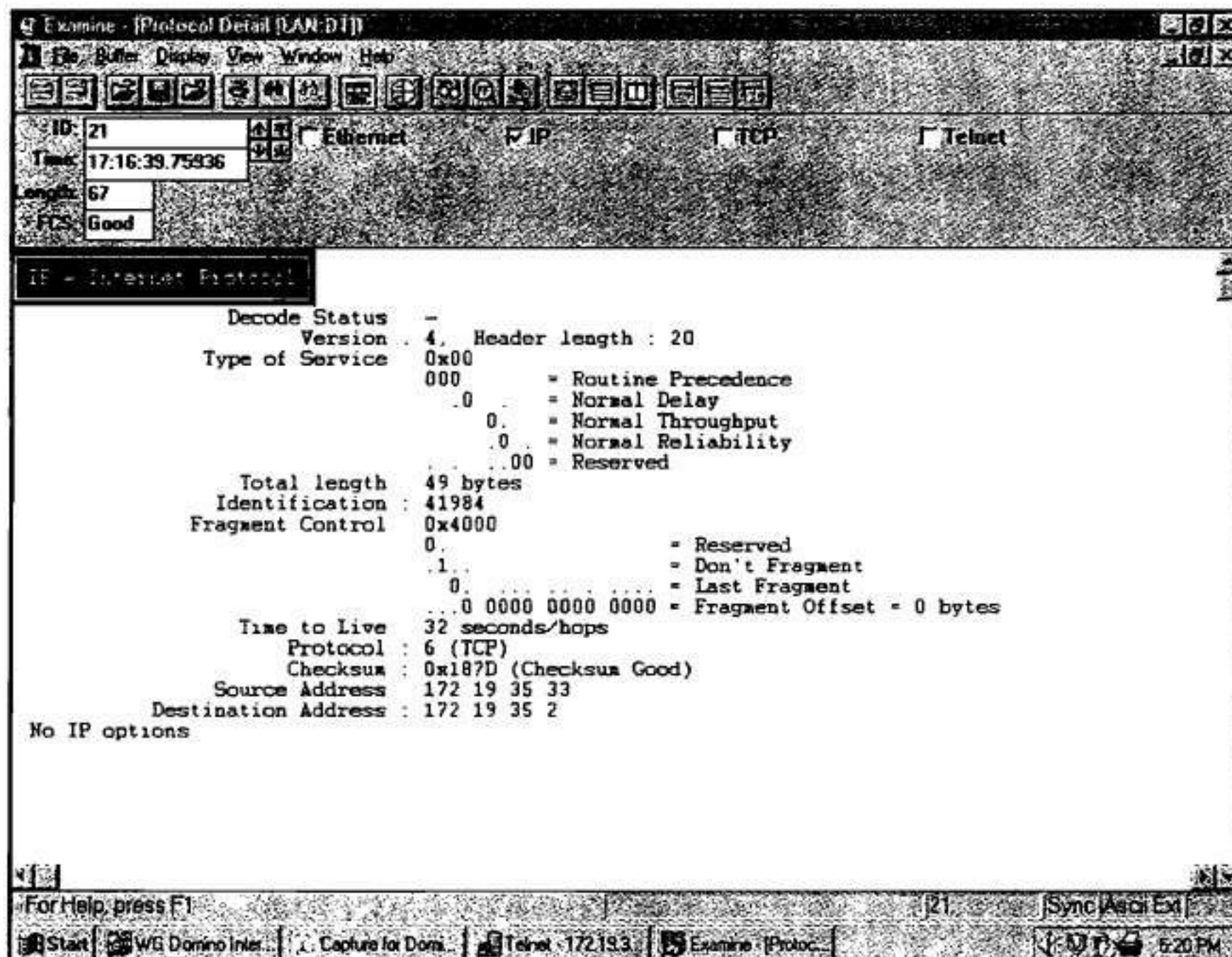


图 2-8 在协议分析器的窗口中，可以看到 IP 报文报头各字段及每个字段的值

2.3 IP 地址

IP 地址长度为 32 位。像所有其他网络层地址一样, IP 地址也包括网络号和主机号两部分。网络号对于连接到网络上的所有设备来说是公共的,它惟一地标识了数据链路(即网络)。而主机号惟一地标识了连接到网络上的特殊设备。

有几种方式可以表示 IP 地址的 32 比特。例如, 32 位 IP 地址 00001010110101100101011110000011 可以用十进制表示为 181 819 267。

可见用二进制表示 IP 地址十分麻烦,而用十进制表示 IP 地址计算起来又很耗时。图 2-9 给出了一个更好的表示方法。32 位的地址包含 4 个 8bit 字节,每个 8bit 字节均可以用 0~255 之间的十进制数表示,而每个十进制数之间用点号分隔。在图 2-9 中,将 32 位的地址映射到用点分十进制法表示的地址上。

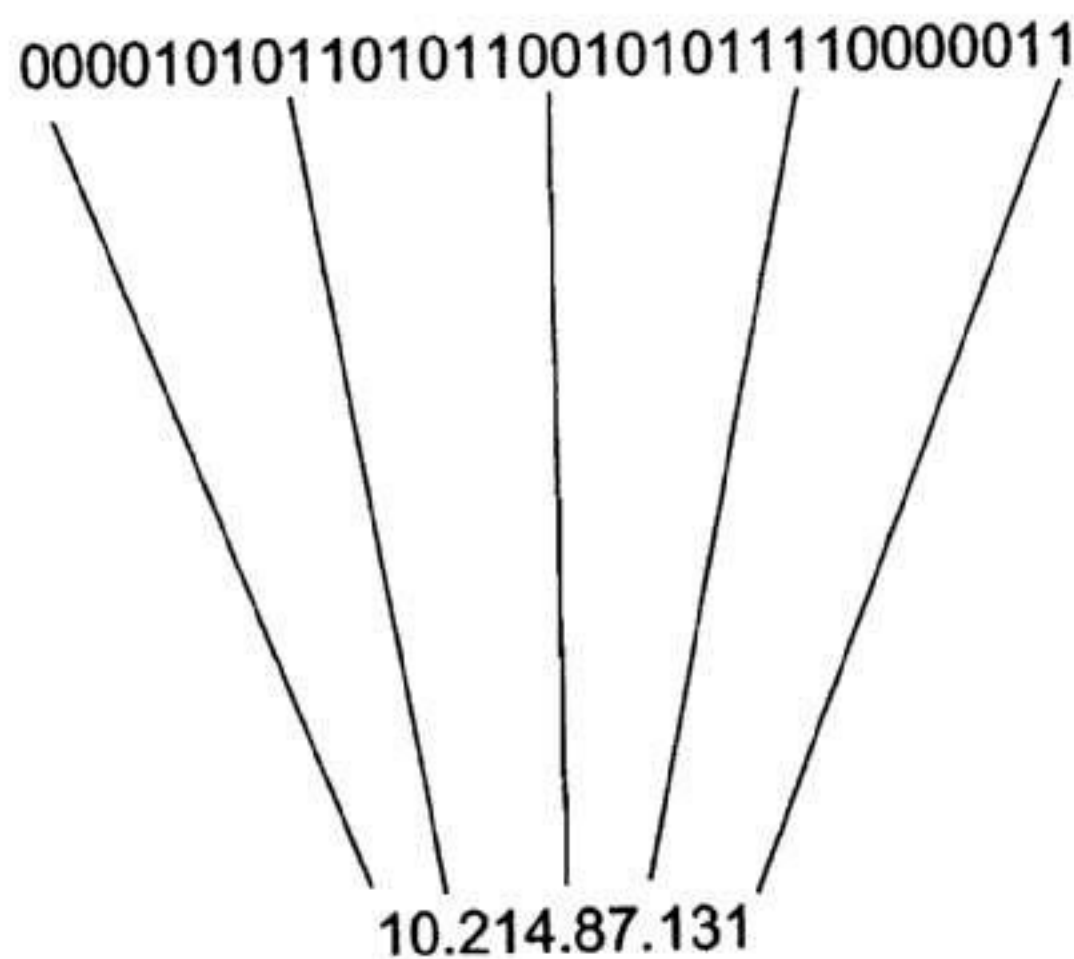


图 2-9 虽然使用点分十进制表示法书写 IP 地址十分方便,但是不要将它与路由器(或主机)所看到的 32 位字符串混淆起来

在使用 IP 地址时需要记住一点,点分十进制表示法便于人们阅读和书写,而路由器更适合使用 32 位二进制串,而不是按照 4 个 8bit 字节的方式读取地址。牢记这一点可以避免许多易犯的错误。

可能 IP 地址与众不同的特性就是 IP 地址不像其他网络层地址的网络号和主机号固定不变,IP 地址的网络号和主机号可以在 32 位的界线内发生变化。也就是说,IP 地址的网络号和主机号都有可能占据 32 位中的大多数位,也可能两者平分 32 位。例如 NetWare 和 AppleTalk 协议,由于它们主要用于相对较小的互联网络,¹所以协议的网络层地址的网络号和主机号长度固定。这样的安排的确使得工作更加容易,接收设备可以从地址中读入固定的比特来获取网络号,剩下的比特便是主机号。

然而,TCP/IP 从最初设计出来到现在可以灵活地应用于任何互联网络,从很简单的几个功能发展成为一个庞大的协议族。TCP/IP 这种适应性使得 IP 地址的管理更加困难。本节仅

¹ 然而,这些协议的普及性使得它们的应用规模远远超过了协议设计者所想象的;结果,在大型 Novell 和 Apple 互联网络中产生了大量令人关注的困难和挑战。

介绍了 IP 地址管理的一些基本内容，在第 7 章中将会介绍一些更高级的技术。

2.3.1 首个 8bit 字节规则

如果不对互联网络作太过精确的划分，那么互联网络可以按照主机数量分为 3 类：大型、中型和小型。

- **大型互联网络**——可以定义为包含大量主机的网络。大型互联网络的数量相对很少；
- **小型互联网络**——作为大型互联网络的对照，它仅仅包含很少数量的主机，但小型互联网络的数目很多；
- **中型互联网络**——相对于大型和小型互联网络来说，包含的主机数量中等，而且中型互联网络的数量也中等；

对于这 3 种规模的互联网络，高层的地址划分要求有 3 种类型网络地址。面向大型互联网络的地址需要有能够为大量的主机编址，但是由于大型互联网络的数量有限，所以大型互联网络仅需要少量网络地址。

而对小型互联网络来说情况又颠倒过来，因为小型互联网络数量庞大，所以需要大量的小型互联网络网络地址。但是小型互联网络主机有限，所以仅需要少量主机地址。

对于中等规模的互联网络来说，网络地址和主机地址的需求量均趋于中等水平。

图 2-10 给出了 3 类 IP 地址的网络号和主机号是怎样划分的。

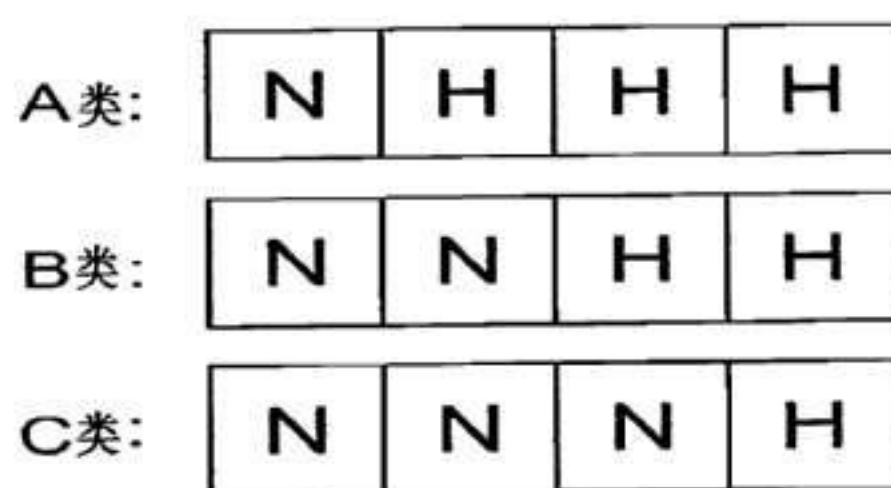


图 2-10 A、B 和 C 类 IP 地址格式

迄今为止，对于所描述的大型、中型和小型互联网络，是按照如下方式映射到各类地址的：

- **A 类 IP 地址**——用于大型互联网络，第 1 个 8bit 字节是网络号，后 3 个 8bit 字节是主机号。8 位的网络号最多可以表示 256 个网络，而每个网络地址的主机号可以提供的主机数量为 2^{24} 或 16 777 216。
- **B 类 IP 地址**——用于中型互联网络。前 2 个 8bit 字节表示网络号，后 2 个 8bit 字节表示主机号。网络号和主机号的数量均为 2^{16} 或 65 536 个。
- **C 类 IP 地址**——对应于 A 类 IP 地址。前 3 个 8bit 字节表示网络号，最后 1 个 8bit 字节表示主机号。

因为所有的 IP 地址都是 32 位二进制串，所以需要某种方法来区分一个特定地址到底是属于哪一类地址。图 2-11 所示的首个 8bit 字节规则提供了这种方法，如下所述：

- 对于 A 类地址，首个 8bit 字节的第 1 位，即 32 位字串最左边的 1 位，总是被设置为 0。因此通过设置首个 8bit 字节的剩余位为 0（最小）或为 1（最大），我们可以找到 A 类地址范围中的最小数和最大数。于是我们可以得到最小数和最大数分别为

0 和 127, 但是这里有几个例外: 0 被保留作为缺省地址部分 (第 12 章), 127 被保留为内部回送地址。¹剩下的十进制数则是 1~126。因此任何首个 8bit 字节落在 1 和 126 之间的 IP 地址均属于 A 类地址。

规则	最小值和最大值	十进制数范围
A 类: 首位恒为 0.	00000000 = 0 01111111 = 127	1 - 126* *0 和 127 是保留的.
B 类: 前两位恒为 10.	10000000 = 128 10111111 = 191	128 - 191
C 类: 前三位恒为 110.	11000000 = 192 11011111 = 223	192 - 223

图 2-11 首个 8bit 字节规则

- B 类地址总是把左边的第 1 位设置为 1, 第 2 位设置为 0。那么再次通过设置首个 8bit 字节的剩余位为 0 或为 1, 我们依然可以找到最小数和最大数。在图 2-9 中, 我们可以看到首个 8bit 字节落在 128 和 191 之间的 IP 地址属于 B 类地址。
- 在 C 类地址中, 前 2 位均被设置为 1, 第 3 位被设置为 0。这样设置的结果是首个 8bit 字节在 192 和 223 之间。²

到目前为止, IP 的编址看上去并不是十分困难。路由器和主机通过首个 8bit 字节规则能够很容易地确定 IP 地址的网络号。如果前 2 位是 10, 那么需要读取 16 比特; 如果前 3 位是 110, 则需求读取 24 比特才能获取网络号。不幸的是, 事情并不会这样简单。

2.3.2 地址掩码 (Address Mask)

表示整个数据链路的地址——非特指某台主机的网络地址, 可以用 IP 地址的网络部分来表示, 其中主机位全部为 0。例如 InterNIC, 作为 IP 地址的管理机构, 它可以将 172.21.0.0³分配给一个申请者。因为 172 在 128 和 191 之间, 所以这是一个 B 类地址, 其中后两个 8bit 字节作为主机位, 全部被设置为 0。虽然前 16 位 (172.21.) 已经被指定, 但是地址所有者有权决定后 16 位主机位的使用。

每一个设备和接口都将被分配一个惟一的、主机号明确的地址, 例如 172.21.35.17。不管设备是路由器还是主机, 显然都需要知道自身的地址, 而且它还需要能够确定它所属的网

¹ UNIX 主机使用内部回环地址 (典型的是 127.0.0.1) 向自己发送流量, 发送到该地址的数据将会被直接送回给发送进程, 而不会离开此设备。

² 注意, 223 并没有用完第一个 8bit 字节中所有可用的数。参加本章最后的配置练习 1。

³ 事实上, 这个地址决不会被分配, 因为它属于私有的保留地址: 本书中所用到的大多数地址都是保留地址, 见 RFC1918。保留地址包括: 10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255, 和 192.168.0.0-192.168.255.255。

络，在这个案例中，它属于 172.21.0.0。

这一任务通常由地址掩码来完成。地址掩码是一个 32 位的字串，与 IP 地址的每一位相对应。掩码也可以像 IP 地址一样用点分十进制表示。这种表示方法会成为某些初学者的绊脚石。虽然地址掩码可以用点分十进制书写，但是它并不是一个地址。表 2-3 给出了对应于 3 类 IP 地址的标准地址掩码；

表 2-3 A 类、B 类和 C 类地址的地址掩码

类	掩 码	点分十进制表示
A	11111111000000000000000000000000	255.0.0.0
B	11111111111111110000000000000000	255.255.0.0
C	11111111111111111111111100000000	255.255.255.0

对于 IP 地址的每一位，设备会拿它与地址掩码的对应位进行布尔（逻辑）AND 操作。AND 函数表述如下：

比较两位并得出结果。当且仅当两位全部为 1 时，结果为 1。如果两位中任意一位为 0，则结果为 0。

对于一个指定的 IP 地址，图 2-12 给出了怎样用地址掩码确定网络地址。地址掩码值为 1 的位对应于地址的网络位，值为 0 的位对应于主机位。因为 172.21.35.17 是 B 类地址，所以掩码前两个 8bit 字节必须全部设置为 1，后两个 8bit 字节，即主机号的所有位必须设置为 0。见表 2-3，这个掩码的点分十进制表示为 255.255.0.0。

逻辑“与”的真值表：

		0	1
0 AND 0 = 0	0	0	0
0 AND 1 = 0			
1 AND 0 = 0	1	0	1
1 AND 1 = 1			

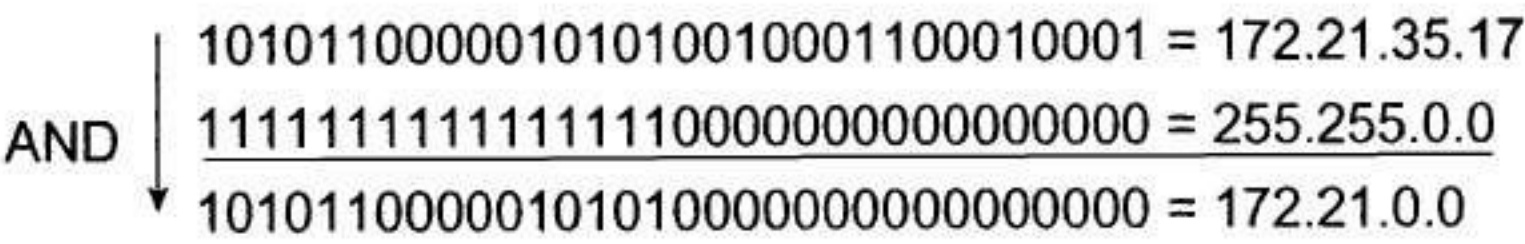


图 2-12 B 类地址的每一位与地址掩码的对应位进行 AND 操作，然后得到网络地址

在 IP 地址和地址掩码的每一位上执行 AND 操作，结果如图 2-12 所示。在结果中，网络位不变，所有主机位则变为 0。通过向接口分配地址 172.21.35.17 和掩码 255.255.0.0，设备将会知道接口属于网络 172.21.0.0。对 IP 地址和掩码应用 AND 操作总能够得到网络地址。

通过下面命令可以向 Cisco 路由器的接口分配地址和掩码：

```
Smokey(config)# interface ethernet 0
Smokey(config-if)# ip address 172.21.35.17 255.255.0.0
```

但是为什么要使用地址掩码？到目前为止，使用首个 8bit 字节规则看上去更简单一些。

2.3.3 子网和子网掩码

首先, 决不要忽略网络层地址的必要性。为了完成路径选择, 每个数据链路(网络)都必须有一个惟一的地址, 另外, 数据链路上的每个主机也必须有一个地址, 这个地址不仅标识主机为一个网络成员, 还可以把主机与网络上的其他主机区分开来。

到目前为止的定义中, 一个 A 类、B 类或 C 类地址仅仅能用于一个单一网络中, 为了建立一个互联网络, 每个数据链路都必须使用不同的地址, 以便这些网络可以被惟一地标识。如果每一个数据链路都使用一个单独的 A 类、B 类或 C 类地址, 那么即使用尽所有的 IP 地址, 也只能给少于 1700 万个数据链路分配地址。显然, 这种方法是不切实际的,¹在前面的例子中, 如果充分地使用主机地址空间, 那么在数据链路 172.21.0.0 中的设备数目可以超过 65000!

使 A 类、B 类或 C 类地址实用化的惟一方法是对主网地址进行划分, 例如将 172.21.0.0 划分为子网地址。请回忆两个事实:

- 地址的主机部分可以随意使用;
- IP 地址的网络号由分配给接口的地址掩码确定。

如图 2-13 所示, 分配给互联网络的地址为 B 类地址 172.21.0.0。5¹个数据链路通过路由器互连起来, 每个数据链路都需要一个网络地址。按照目前的情况, 172.21.0.0 必须分配给其中的一个数据链路, 那么另外 4 个数据链路还需要 4 个地址。

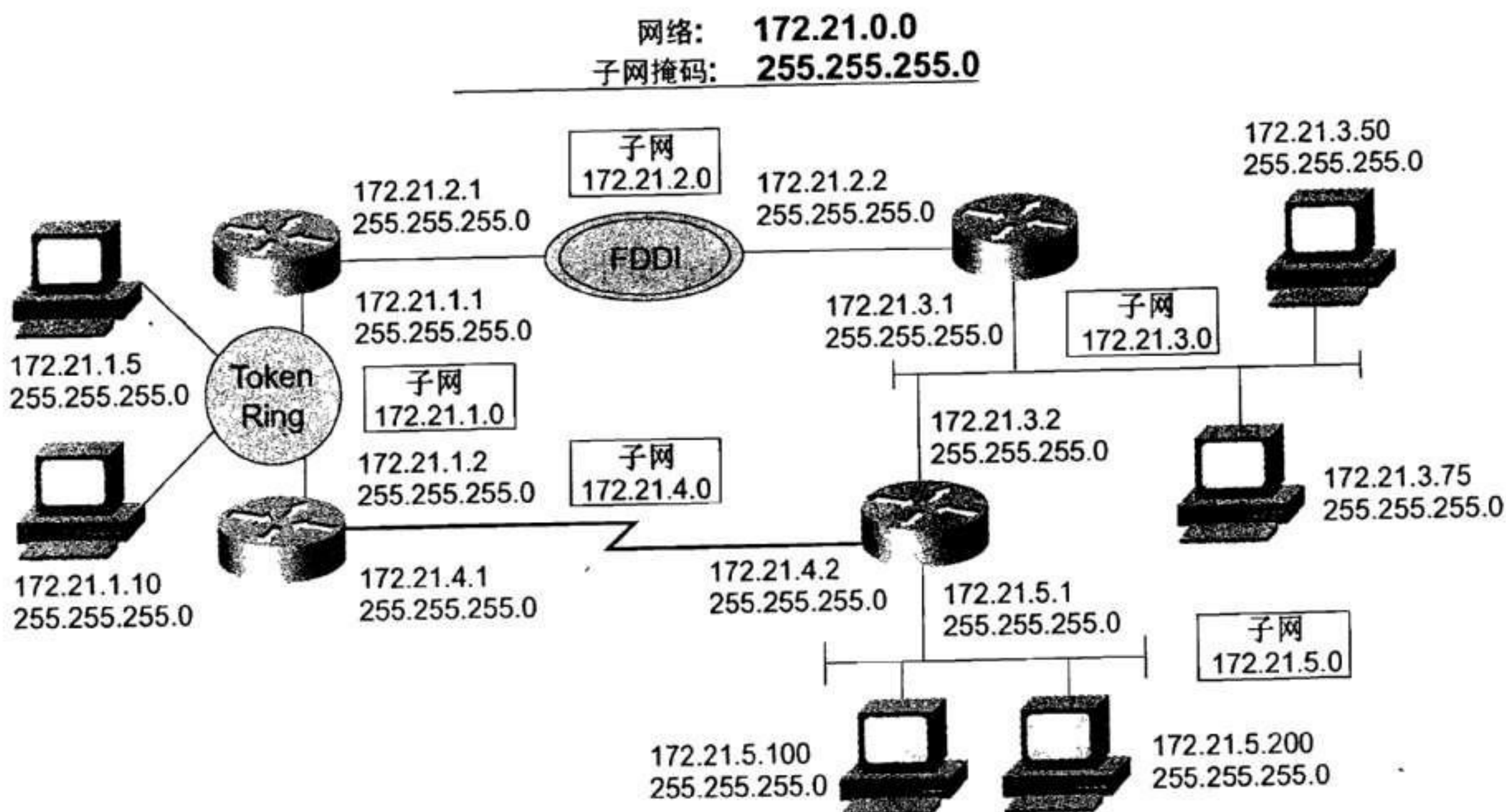


图 2-13 通过向主机位借位用作子网位, 子网掩码使一个单一的网络地址可以用于多个数据链路

注意图 2-13 所示, 地址掩码并不是标准的 16 位 B 类地址掩码; 而是被扩展了 8 位, 以便 IP 地址的前 24 位都被解释为网络位。换句话说, 掩码使路由器和主机把读取的前 8

¹ 1700 万个网络看上去很多, 但是你要考虑到, 一个中等规模的企业就可能有许多网络。

位主机位作为网络地址的一部分。结果，主网络地址应用于整个互连网络，而每一个数据链路则变为一个子网（subnet），一个子网是一个主 A 类、B 类或 C 类地址空间的一个子集。

现在，IP 地址包括 3 个部分：网络部分、子网部分和主机部分。地址掩码现在变为子网掩码，或比标准地址掩码长的掩码。地址的前两个 8bit 字节依然是 172.21，但是第 3 个 8bit 字节——主机位已经由子网位代替——的变化范围为 0~255。在图 2-12 中的互连网络有子网 1、2、3、4 和 5（172.21.1.0~172.21.5.0）。在单一 B 类地址下最多可以有 256 个子网，对应的掩码如图 2-13 所示。

下面给出两点告诫。首先，并不是所有路由选择协议都支持子网地址，即子网位全 0 或全 1。因为这些协议是有类别化协议，它们不能区分一个全 0 子网和主网络号。例如，在图 2-13 中子网 0 为 172.21.0.0；而主网 IP 地址也为 172.21.0.0。没有更多信息将无法区分二者。

同样，有类别路由选择协议也不能区分全 1 子网的广播地址和一个所有子网的广播地址。¹例如，图 2-13 中的全 1 子网为 172.21.255.0。对于这个子网，广播地址是 172.21.255.255，但是这也是在主网 172.21.0.0 的所有子网上所有主机的广播地址。没有更多的信息也无法区分二者。第 1 版 RIP 协议和 IGRP 协议都是有类别路由选择协议；第 7 章将会介绍无类别路由选择协议，这种路由选择协议才可以真正地使用全 0 或全 1 子网。

其次是与子网及其掩码的口头表述有关。在图 2-13 中，对 B 类地址的第 3 个 8bit 字节进行子网划分是非常普遍的，但还常常听到人们这样表述子网设计：“B 类地址使用 C 类地址掩码”，“将 B 类地址划分为 C 类地址”。这两种表述都是错误的。它们常常会对子网设计引起误解或者是不准确的理解。对于图 2-12 中所示的子网划分图解的正确表述应该是“一个使用 8 位进行子网划分的 B 类地址”或“一个带有 24 位掩码的 B 类地址”。

可以用 3 种格式表示子网掩码——点分十进制、位计数和十六进制，如图 2-14 所示。虽然位计数格式变得渐渐流行起来，但是点分十进制仍旧是最通用的格式。与点分十进制相比，位计数格式更容易书写（地址后面是 /，/ 后面紧跟着是网络部分的位计数）。另外，位计数格式可以更清楚地描述掩码的实际作用，因而可以避免前面段落出现的语义误解问题。许多 UNIX 系统使用十六进制格式。

点分十进制表示

255.255.255.0

位计数表示

172.21.0.0/24

十六进制表示

0xFFFFF00

图 2-14 在图 2-13 中的子网掩码可以使用 3 种格式表示

虽然在 Cisco 路由器中必须使用点分十进制方式表示地址掩码，但是在行配置模式下使

¹ 所有主机的 IP 广播地址是所有位全为 1: 255.255.255.255。特定子网的广播地址是所有主机位全为 1: 例如，子网 172.21.1.0 的广播地址是 172.21.1.255。最后，对于所有子网的所有主机来说，广播地址是子网为和主机位均为 1: 172.21.255.255。

用命令 **ip netmask-format[dec|hex|bit]**, 可以设置使用 3 种格式中的任何一种格式显示掩码。例如, 为使路由器以位计数格式显示掩码, 配置如下:

```
Gladys(config) # line vty 0 4
Gladys(config-line)# ip netmask-format bit
```

2.3.4 子网规划

如前面部分所述, 在有类别地址环境中, 子网位不能全部为 0 或全部为 1。同样, 一个主机的 IP 地址也不能将主机位全部设置为 0, 这种用法是为路由器保留的, 用于表示网络和子网自身。当然 IP 地址的主机位也不能全部被设置为 1, 因为它用于表示广播地址。所有这些限制无一例外地适用于 IP 地址的主机位, 并且这也是子网规划的起点。除了这些限制, 网络设计人员还需要根据地址空间与互联网络详细的匹配程度来选择最合理的子网划分方案。

在规划子网和子网掩码时, 可以使用相同的公式计算一个主网地址下可用的子网数以及每个子网内可用的主机数, 公式为: $2^n - 2$, 其中 n 表示子网位数或主机空间, 2 表示减去全 0 和全 1 两个不可用地址。例如, 给定一个 A 类地址 10.0.0.0, 子网掩码 10.0.0.0/16(255.255.255.0) 意味着有 8 位子网空间, 也就是可以产生 $2^8 - 2 = 254$ 个子网, 每个子网可以有 $2^{16} - 2 = 65\,534$ 个主机地址。另一方面, 掩码 10.0.0.0/24(255.255.255.0) 表示有 16 位子网空间, 可以产生 65 534 个子网, 其中 8 位主机空间可以在某个子网中产生 254 个主机地址。

下面是 IP 地址子网划分的步骤:

步骤 1: 确定需要多少个子网, 每个子网需要多少个主机。

步骤 2: 为了满足第 1 步提出的需求, 使用公式 $2^n - 2$ 确定子网位数和主机位数。如果存在多个子网掩码可以满足第 1 步需求, 那么选择最能够符合未来需求的一个。例如, 如果互联网络最有可能通过增加子网发展起来, 那么选择子网位最多的掩码; 如果互联网络最有可能借助增加现有子网内的主机数发展起来, 则选择主机位最多的掩码。为了避免所选择的方案中的子网及子网内的主机地址被迅速地用完, 需要为将来的发展预留一些空间。

步骤 3: 使用二进制进行计算, 在子网空间中确定所有的位组合方式; 在每种组合方式中, 将所有主机位都设置为 0。将得到的子网地址转换为点分十进制格式。最终结果就是子网地址。

步骤 4: 对于每一个子网地址, 再次使用二进制, 在保持子网位不变的情况下写出所有主机位组合, 并将结果转换成点分十进制格式。最终结果就是每个子网的可用主机地址。

这里没有过分强调在最后两步中使用二进制的重要性。当进行子网划分时, 最主要的单一错误根源就是在没有理解在二进制上会发生什么的情况下试图使用点分十进制方法。此外, 点分十进制表示法对于人们读写 IP 地址十分方便。但是路由器和主机却把地址看作 32 位二进制串; 为了顺利地完地址操作, 必须采用路由器和主机处理地址的方式。

就目前给出的例子而言, 最后一段看上去好像过分担心了, 在没有限定必须使用二进制方式表示地址和掩码的时候, 子网模式和主机地址看上去还是十分清楚的。下一部分讨论使用 4 个步骤完成子网规划, 在那里点分十进制表示法将十分不明确。

2.3.5 打破 8bit 字节界线

到目前为止，在给出的例子中，子网空间都是以 8bit 字节为界线的。但这并不总是最实用或最有效的选择。例如，如果你需要对一个 B 类地址进行子网划分，并满足以下需求：网络数为 500，每个网络内主机数不超过 100 个，应该怎么办？这样的需求很容易得以满足，只要使用 9 位子网位，就可以得到 $2^9 - 2 = 510$ 个子网，剩下 7 位做主机位，每个子网的可用主机数为 $2^7 - 2 = 126$ 。除此不再有其他位组合可以满足上面的需求。

注意：如果还是以 8bit 字节为界线的话，那么将无法对 C 类地址进行子网划分。如果要这样做就会占用最后 1 个 8bit 字节，那么就没有更多主机位了。因此，如下面的例子所示，子网位和主机位必须共享最后 1 个 8bit 字节

步骤 1：除了这里分配的地址是 C 类地址 192.168.100.0 之外，图 2-15 的互联网络与图 2-13 的互联网络完全相同。互联网络共有 5 个数据链路，因此至少需要划分出 5 个子网地址。图中还指明了每个子网需要分配的主机数。其中两个以太网最多需要 25 个主机地址。所以完整的子网划分最小需求是 5 个子网，每个子网至少需要 25 个主机地址。

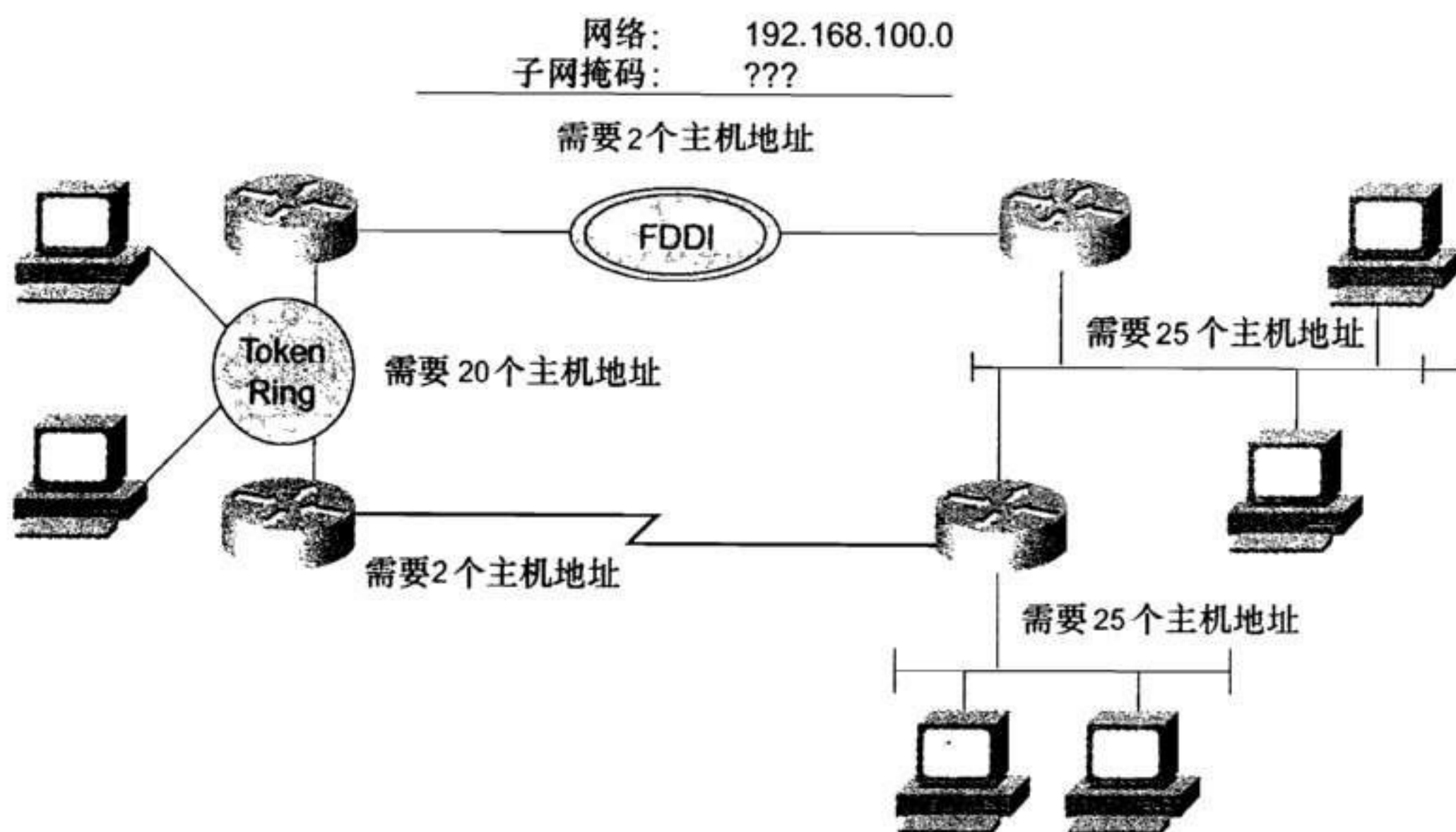


图 2-15 除了分配 C 类地址之外，这里沿用图 2-13 给出的网络。如果子网位占用整个 8bit 字节，那么将无法进行划分，因为主机位将没有空间

步骤 2：使用公式 $2^n - 2$ 可以计算出，3 个子网位和 5 个主机位即可以满足需求： $2^3 - 2 = 6$ ， $2^5 - 2 = 30$ 。带有 3 位子网位的 C 类地址掩码可以用点分十进制表示为 255.255.255.224。

步骤 3：图 2-16 给出了子网位的推导过程。用二进制表示第 2 步计算出的子网掩码，子网掩码下面是 IP 地址。垂直线标记了子网空间，从二进制 0 开始计数，在这一空间中的所有位组合均被写出。

网络位	主机位
11000000101010000110010000100000	= 192.168.100.32 ← 子网号
11000000101010000110010000100001	= 192.168.100.33
11000000101010000110010000100010	= 192.168.100.34
11000000101010000110010000100011	= 192.168.100.35
11000000101010000110010000100100	= 192.168.100.36
11000000101010000110010000100101	= 192.168.100.37
11000000101010000110010000100110	= 192.168.100.38
11000000101010000110010000100111	= 192.168.100.39
11000000101010000110010000101000	= 192.168.100.40
11000000101010000110010000101001	= 192.168.100.41
11000000101010000110010000101010	= 192.168.100.42
11000000101010000110010000101011	= 192.168.100.43
11000000101010000110010000101100	= 192.168.100.44
11000000101010000110010000101101	= 192.168.100.45
11000000101010000110010000101110	= 192.168.100.46
11000000101010000110010000101111	= 192.168.100.47
11000000101010000110010000110000	= 192.168.100.48
11000000101010000110010000110001	= 192.168.100.49
11000000101010000110010000110010	= 192.168.100.50
11000000101010000110010000110011	= 192.168.100.51
11000000101010000110010000110100	= 192.168.100.52
11000000101010000110010000110101	= 192.168.100.53
11000000101010000110010000110110	= 192.168.100.54
11000000101010000110010000110111	= 192.168.100.55
11000000101010000110010000111000	= 192.168.100.56
11000000101010000110010000111001	= 192.168.100.57
11000000101010000110010000111010	= 192.168.100.58
11000000101010000110010000111011	= 192.168.100.59
11000000101010000110010000111100	= 192.168.100.60
11000000101010000110010000111101	= 192.168.100.61
11000000101010000110010000111110	= 192.168.100.62
11000000101010000110010000111111	= 192.168.100.63 ← 广播地址

图 2-18 写出主机空间中所有位组合可以得到子网内的主机地址。这里是子网 192.168.100.32 的主机位

2.3.6 子网掩码的故障排除

在“解剖”一个给定的主机地址和掩码时，常常需要确定地址属于哪个子网。例如，如果在一个接口上配置了地址，一个很好的实践就是首先验证对于接口连接的子网来说该地址是否合法。

使用下面的步骤逆推一个 IP 地址：

步骤 1：用二进制写下给定的子网掩码。

步骤 2：用二进制写下主机 IP 地址；

步骤 3：在知道一个地址的类别后，掩码的子网位便是显然的了。根据掩码位，在最后网络位和第 1 个子网位之间画一条线，在最后子网位和第 1 个主机之间也画另一条线。

步骤 4：写下地址的网络位和子网位，设置所有的主机位为 0。最终的结果就是主机地址所属的子网地址。

步骤 5：再次写下地址的网络位和子网位，这次设置所有主机位为 1。结果就是本子网的广播地址。

步骤 6: 按照顺序可以知道第一个地址是子网地址, 最后一个地址是广播地址。而且还可以知道在这两个地址之间的所有地址都是合法的主机地址。

对于地址 172.30.141/25, 图 2-19 给出了以上步骤的示例。这个地址是 B 类地址, 所以前 16 位是网络位, 25 位掩码中的后 9 位是子网位。可以发现子网地址是 172.30.0.128, 广播地址是 172.30.0.255。在这两个地址之间的主机地址对于这个子网来说都是合法的, 如对子网 172.30.0.128 来说, 172.30.0.129~172.30.0.254 都是主机地址。

172.30.0.141/25		
(1) 写出子网掩码:	11111111111111111111111100000000 = 255.255.255.128	
(2) 写出 IP 地址:	101011000001111100000000010001101 = 172.30.0.141	
(3) 标记子网空间:	11111111111111111111111100000000 = 255.255.255.128	
	101011000001111100000000010001101 = 172.30.0.141	
导出 ...	11111111111111111111111100000000 = 255.255.255.128	
	101011000001111100000000010001101 = 172.30.0.141	
(4) 子网地址:	10101100000111110000000000000000 = 172.30.0.128	
(5) 广播地址:	10101100000111110000000011111111 = 172.30.0.255	

(6) 对于这个子网, 有效的主机地址是 172.30.0.129 – 172.30.0.254。

图 2-19 给定一个 IP 地址和子网掩码, 按照以下步骤可以找出子网地址、广播地址和主机地址

在这个例子中, 初次进行子网划分的人可能会受到以下几种情况的干扰。一种是地址的第 3 个 8bit 字节所有位都为 0。另一种是最后一个 8bit 字节仅一个子网位。一些人可能会认为广播地址看上去不合法。所有这些不舒服的感觉都源自地址的点分十进制表示法。当使用二进制表示地址和掩码时, 这些疑虑会被打消, 任何事看上去都一切正常, 掩码设定了 9 位子网空间——包括第 3 个 8bit 字节和第 4 个 8bit 字节的第 1 位。这个案例说明了如果使用二进制表示法时一切正常, 那么就不必担心看上去有些奇怪的点分十进制表示法。

2.4 ARP

第 1 章解释了通过读取和操作报文的网络地址, 路由器可以沿逻辑路径传送报文, 其中逻辑路径包括多个数据链路。沿独立的数据链路传送报文时, 需要把被报文封装在帧中, 并且使用数据链路标识 (如 MAC 地址) 让帧可以从链路的源点到达目的地。本书的主题之一是为了进行路由选择, 路由器利用何种机制发现并共享地址信息。类似的, 数据链路上的设备也需要一种方法发现邻居的数据链路标识, 以便将数据帧传送到正确的目的地。

有几种机制可以提供这些信息¹; IP 使用地址解析协议 (ARP), 详见 RFC826。图 2-20 给出了 ARP 的工作机制。当一个设备需要发现另一个设备的数据链路标识时, 它将建立一个 ARP 请求报文。这个请求报文中包括目标设备的 IP 地址以及请求设备 (发送者) 的源点 IP 地址和数据链路标识 (MAC)。然后 ARP 请求报文被封装在数据帧中, 其中带有作为源的发

¹ 例如, NetWare 把设备的 MAC 地址作为网络层地址的主机部分, 这是一个明智之举。

送者的 MAC 地址和作为目标的广播地址（图 2-21）。¹



图 2-20 ARP 用于把设备的数据链路标识映射到它的 IP 地址上

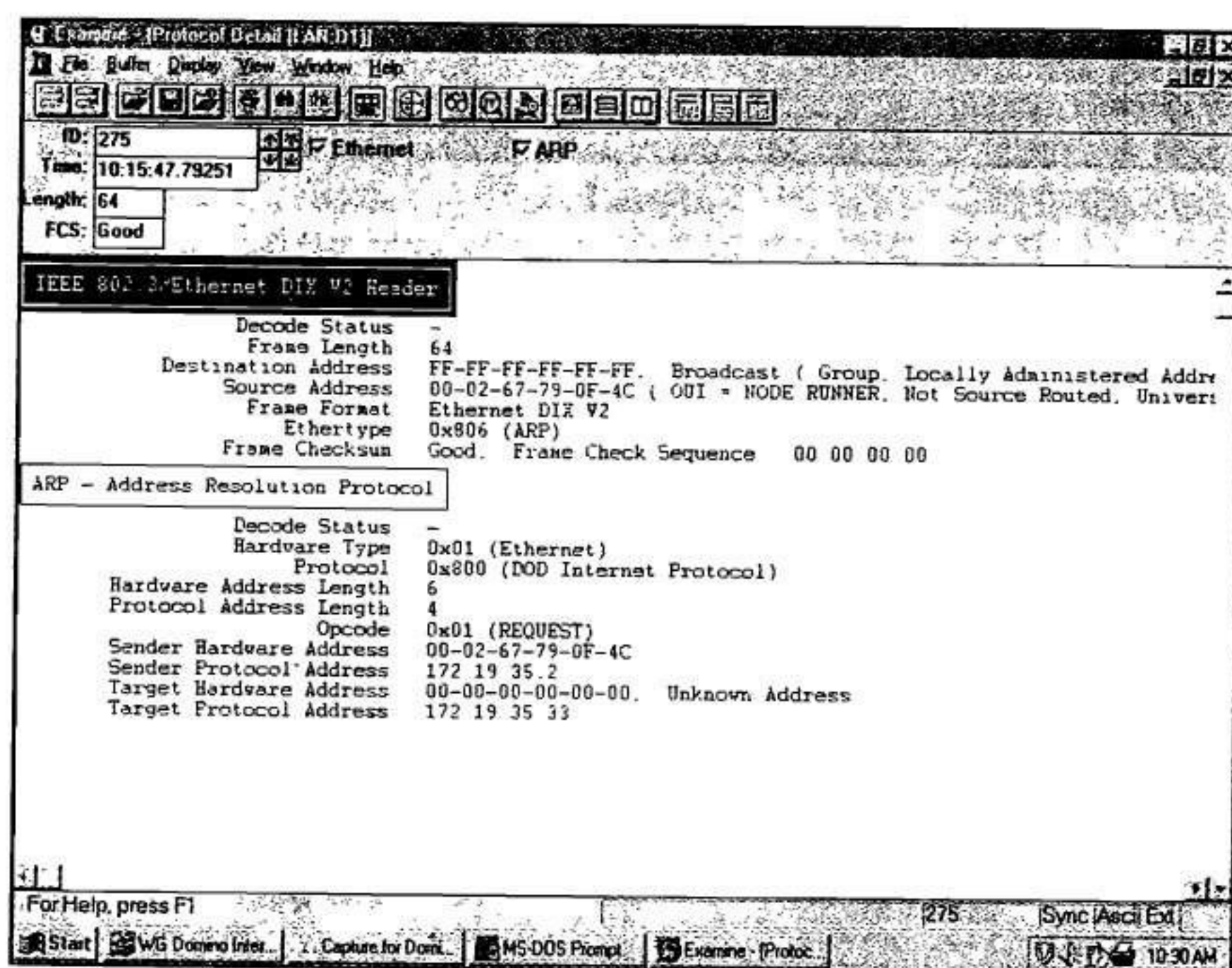


图 2-21 协议分析器捕捉到图 2-20 所描述的 ARP 请求报文及封装帧

广播地址意味着数据链路上的所有设备都将收到该帧，并且要检查帧内封装的报文。除了目标机可以识别此报文外，其他所有设备都会丢弃此报文。目标机将向源地址发送 ARP 响应报文，提供它的 MAC 地址（图 2-22）。

当调用调试功能 **debug arp** 时，Cisco 路由器可以显示 ARP 的活动情况，如图 2-23 所示。

图 2-24 给出了 ARP 报文的格式。这里可以把图中描述的各字段同图 2-21 和图 2-22 的 ARP 报文相对照。

- **硬件类型（Hardware Type）**——指定了硬件的类型，详见 RFC1700²。一些常用的类型见表 2-4。

¹ 类似于 IP 的广播地址，MAC 的广播地址也是所有位全部为 1：ffff.ffff.ffff。

² J.Postel 和 J.Reynolds。“指定的号码。” RFC1700，1994 年 10 月。RFC1700 指定了整个 TCP/IP 协议族中各字段所有在用的号码。这本大型文档（230 页）是一本很有价值的参考文献，身边最好保留一份拷贝以便参考。

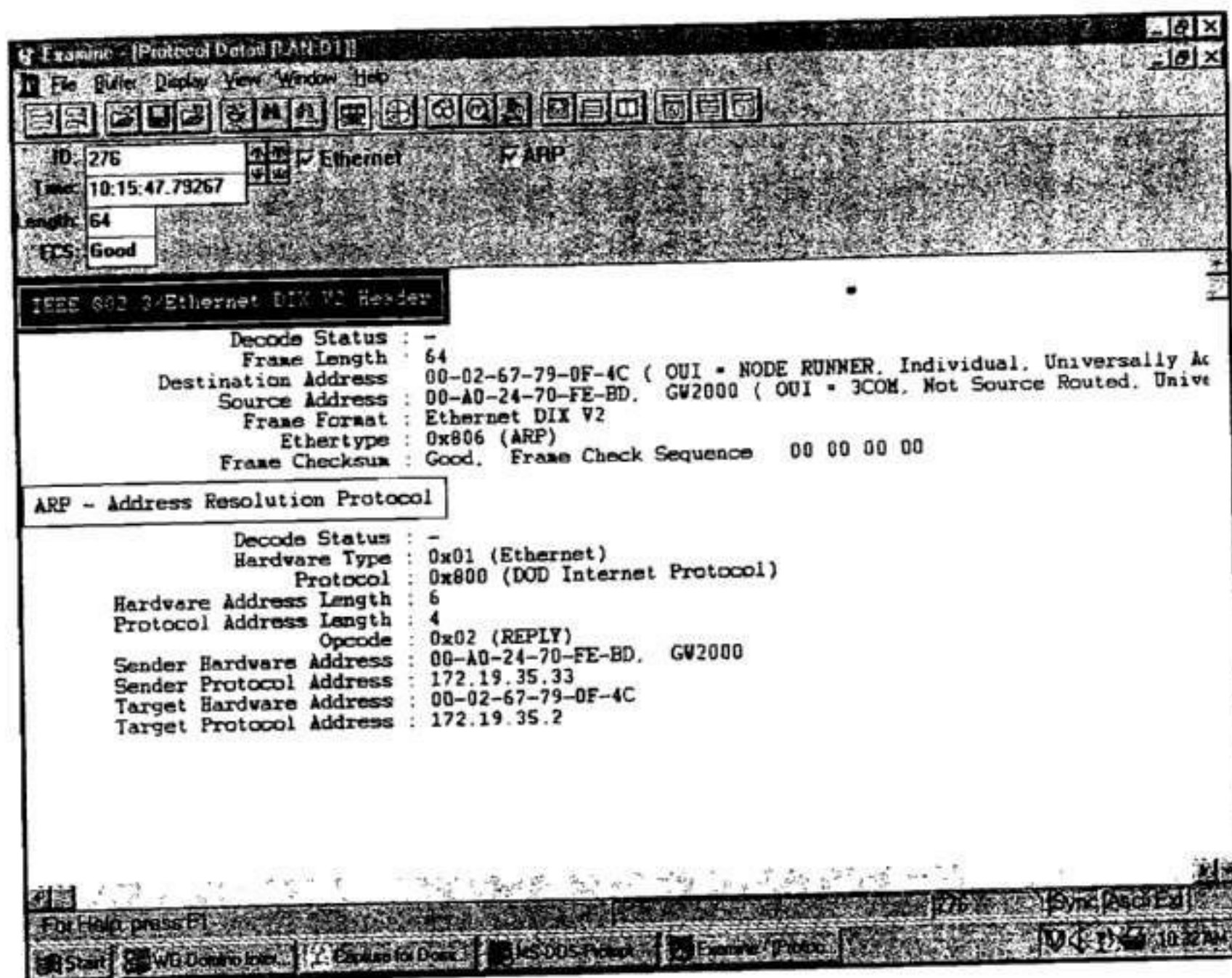


图 2-22 协议分析器捕捉的图 2-20 所描述的 ARP 响应报文

```
Aretha#debug arp
IP ARP: rcvd req src 172.19.35.2 0002.6779.0f4c, dst 172.21.5.1 Ethernet0
IP ARP: sent rep src 172.21.5.1 0000.0c0a.2aa9,
dst 172.19.35.2 0002.6779.0f4c Ethernet0
Aretha#
```

图 2-23 路由器 Aretha (172.21.5.1) 响应来自主机 172.19.35.2 的 ARP 请求



图 2-24 ARP 报文格式

表 2-4 常用的硬件类型码

编 号	硬 件 类 型
1	以太网
3	X.25
4	Proteon ProNET Token Ring
6	IEEE 802 网络
7	ARCnet
11	Apple LocalTalk
14	SMDS
15	帧中继
16	异步传输模式 (ATM)
17	高速数据链路控制 (HDLC)
18	光纤信道
19	异步传输模式 (ATM)
20	串行链路

- **协议类型 (Protocol Type)** ——指定了发送者映射到数据链路标识符的网络层协议的类型；IP 对应 0x0800。
- **硬件地址长度 (Hardware Address Length)** ——指定了数据链路标识符的长度，单位是 8bit 字节 (octet)。MAC 地址的长度为 6。
- **协议地址长度 (Protocol Address Length)** ——指定了网络层地址的长度，单位是 8bit 字节。IP 地址的长度为 4。
- **操作 (Operation)** ——指明了一个报文是 ARP 请求 (1) 还是 ARP 响应 (2)。这里还可以发现有其他的值表明 ARP 报文的其他用途。如反向 ARP 请求 (4)、反向 ARP 响应 (5)、反转 ARP 请求 (8)、反转 ARP 响应 (9)。

最后 20 个 8bit 字节是发送者和目标机的数据链路标识和 IP 地址。

在图 2-25 所示屏幕的最上面，命令 **show arp** 用于检查 Cisco 路由器内的 ARP 表。请注意年龄一栏，这一列表明为了防止陈旧信息充满 ARP 表，每经过一个特定的实际间隔，ARP 信息将会被刷新。Cisco 路由器保存 ARP 表项的时间为 4 个小时 (14 400s)；这个缺省值可以修改。下面的例子就是将 ARP 的超时值修改为 30min (1 800s)：

```
Martha(config)# interface Ethernet 0
Martha(config-if)# arp timeout 1800
```

图 2-25 所示屏幕的中间给出了 Windows 95 PC 的 ARP 表，屏幕底部给出了 Linux 机器的 ARP 表。虽然它们的格式不同于 Cisco 路由器的 ARP 表，但是 3 个表中的实质性信息是相同的。

ARP 表项还可以永久地保存在表中。为了实现地址 172.21.5.131 到硬件地址 0000.00a4.b74c 的静态映射，并且采用 SNAP 封装类型，可以使用一些命令完成：

```
Martha(config)# arp 172.21.5.131 0000.00a4.b74c snap
```

命令 **clear arp-cache** 可以从 ARP 表中强制删除所有动态表项。并且此命令也可以清除快速交换高速缓冲区和 IP 路由高速缓冲区中的内容。

ARP 还有几种变形；其中至少有一种对路由选择十分重要，它就是代理 ARP。

Martha#show arp						
Protocol	Address	Age (min)	Hardware	Addr	Type	Interface
Internet	10.158.43.34	2	0002 . 6779 . 0f4c		ARPA	Ethernet0
Internet	10.158.43.1	-	0000 . 0c0a . 2aa9		ARPA	Ethernet0
Internet	10.158.43.25	18	00a0 . 24a8 . a1a5		ARPA	Ethernet0
Internet	10.158.43.100	6	0000 . 0c0a . 2c51		ARPA	Ethernet0
Martha#						

C:\WINDOWS>arp -a		
Interface: 148.158.43.25		
Internet Address	Physical Address	Type
10.158.43.1	00-00-0c-0a-2a-a9	dynamic
10.158.43.34	00-02-67-79-0f-4c	dynamic
10.158.43.100	00-00-0c-0a-2c-51	dynamic

Linux:-# arp -a				
Address	HW type	HW address	Flags	Mask
10.158.43.1	10Mbps Ethernet	00:00:0C:0A:2A:A9	C	*
10.158.43.100	10Mbps Ethernet	00:00:0C:0A:2C:51	C	*
10.158.43.25	10Mbps Ethernet	00:A0:24:A8:A1:A5	C	*
Linux:-#				

图 2-25 连接到相同网络上的 3 台设备的 ARP 表: Cisco 路由器, Windows95 主机和 Linux 主机

2.4.1 代理 ARP

代理 ARP 有时也被叫做混杂 ARP, 详见 RFC925 和 RFC1027, 代理 ARP 被路由器作为向主机表明自身可用的一种手段。例如, 主机 192.168.12.5/24 需要向主机 192.168.20.101/24 发送报文, 但是它没有配置缺省网关信息, 因而也就不知道如何到达路由器。这时它可以向 192.168.20.101 发送一个 ARP 请求, 本地路由器收到这一请求后, 并且路由器知道如何到达

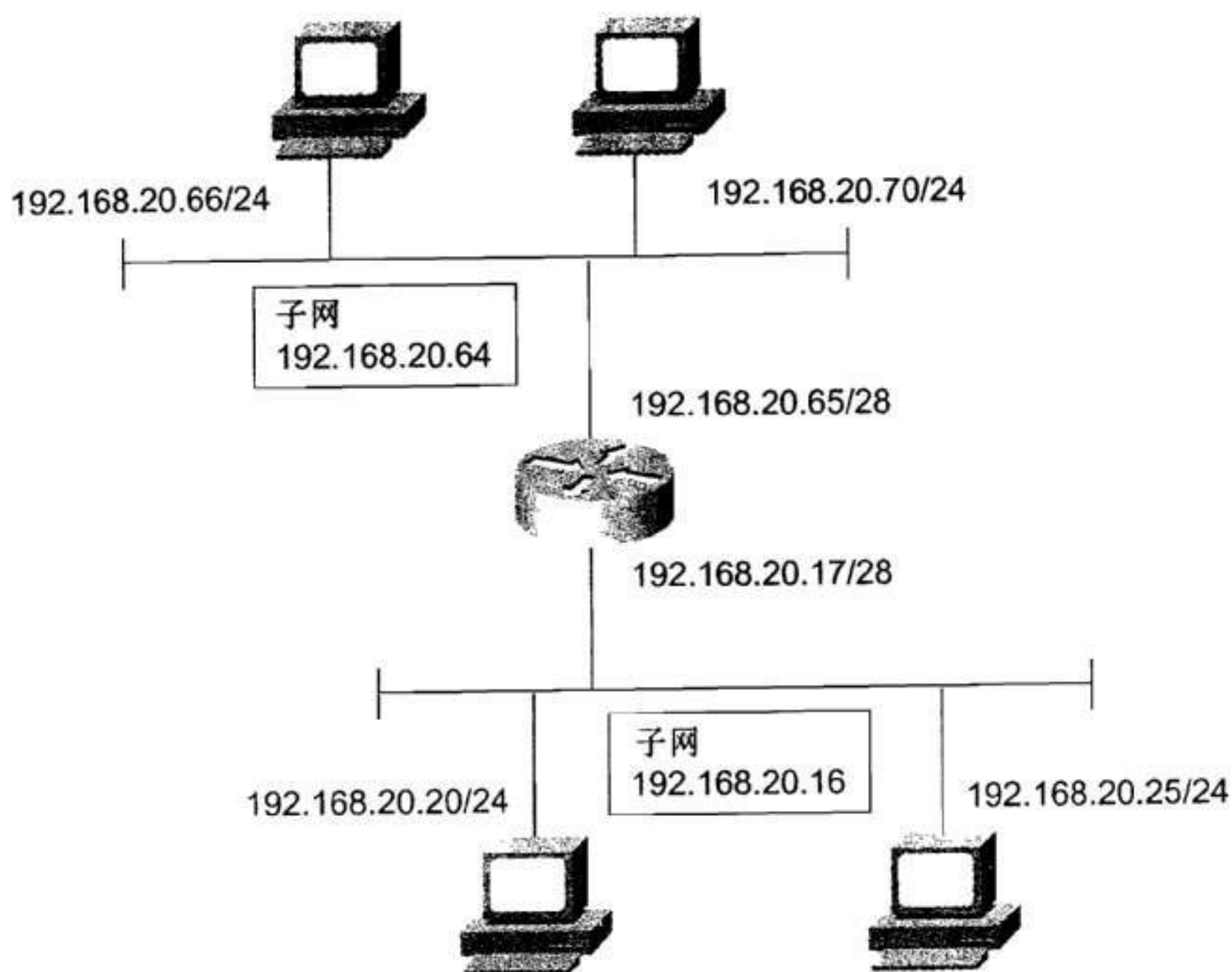


图 2-26 代理 ARP 实现了子网划分的透明性

网络 192.168.20.0，因此路由器将回复以上请求，其中把自己的数据链路标识作为 ARP 回复报文中的硬件地址。事实上，路由器欺骗了本地的主机，让它认为路由器的接口就是 192.168.20.101 的接口。最终所有发向 192.168.20.101 的报文都被送往路由器。

图 2-26 给出了代理 ARP 的另一种用途。这里特别关注的是地址掩码。路由器配置的掩码是 28 位掩码（4 个子网位的 C 类地址），而主机配置的是标准的 C 类地址掩码（24 位）。其结果是主机并不知道子网的存在。当主机 192.168.20.66 想发送报文到 192.168.20.25 时，它首先将发送 ARP 请求。这时路由器识别出报文的目标地址属于另一个子网，因而向请求主机回复自己的硬件地址。这种代理 ARP 使得子网化网络拓扑结构对主机来说是透明的。

图 2-27 所示的 ARP 高速缓冲暗示了代理 ARP 的又一用途。注意，有多个 IP 地址映射到单一的 MAC 标识符；其中 IP 地址对应着主机，而硬件 MAC 标识符属于路由器接口。

C:\WINDOWS>arp -a		
Interface: 192.168.20.66		
Internet Address	Physical Address	Type
192.168.20.17	00-00-0c-0a-2a-a9	dynamic
192.168.20.20	00-00-0c-0a-2a-a9	dynamic
192.168.20.25	00-00-0c-0a-2a-a9	dynamic
192.168.20.65	00-00-0c-0a-2c-51	dynamic
192.168.20.70	00-02-67-79-0f-4c	dynamic

图 2-27 图 2-26 中主机 192.168.20.66 的 ARP 表显示出多个 IP 地址映射到单一 MAC 标识符，这说明正在使用代理 ARP

在 Cisco 路由器上，缺省情况下代理 ARP 功能是打开的，当然也可以在每个接口上使用命令 **no ip proxy-arp** 关闭此功能。

2.4.2 无故 ARP

主机偶尔也会使用自己的 IP 地址作为目标地址发送 ARP 请求。这种 ARP 请求称为无故 ARP，主要有两个用途：

- 无故 ARP 可以用于检查重复地址。一个设备可以向自己的 IP 地址发送 ARP 请求，如果收到 ARP 响应则表明存在重复地址。
- 无故 ARP 还可以用于通告一个新的数据链路标识。当一个设备收到一个 ARP 请求，如果 ARP 高速缓冲中已有发送者的 IP 地址，那么与此 IP 地址相对应的硬件地址将会被发送者新的硬件地址所更新。这种无故 ARP 用途正是基于此事实。

许多 IP 实现中都没有实现无故 ARP 功能，但是读者应该知道它的存在。

2.4.3 反向 ARP

代替映射硬件地址到已知 IP 地址，反向 ARP (RARP) 可以实现 IP 地址到已知硬件地址的映射。某些设备，如无盘工作站在启动时可能不知道自己启动时的 IP 地址。嵌入这些设备固件中的 RARP 程序可以允许它们发送 RARP 请求，其中硬件地址为设备的硬件编入地址。RARP 服务器将会向这些设备回复相应的 IP 地址。

RARP 在很大程度上正在被自举协议 (BOOTP) 和其扩展协议动态主机配置协议 (DHCP) 所替代，不同于 RARP，这两种协议都可以提供 IP 地址以外的更多信息，而且还可以跨越本

地数据链路。

2.5 ICMP

Internet 消息控制协议 (ICMP) 指定了多种消息类型, 这些消息的共同目的就是管理互联网络, 详见 RFC792。ICMP 的消息可以分为错误消息、请求消息和响应消息。图 2-28 给出一般的 ICMP 报文格式。报文可以通过类型来标识, 许多报文类型都有多个指定的类型, 可以用代码字段来标识它们。表 2-5 列出了多种 ICMP 的报文类型和代码, 详见 RFC1700。

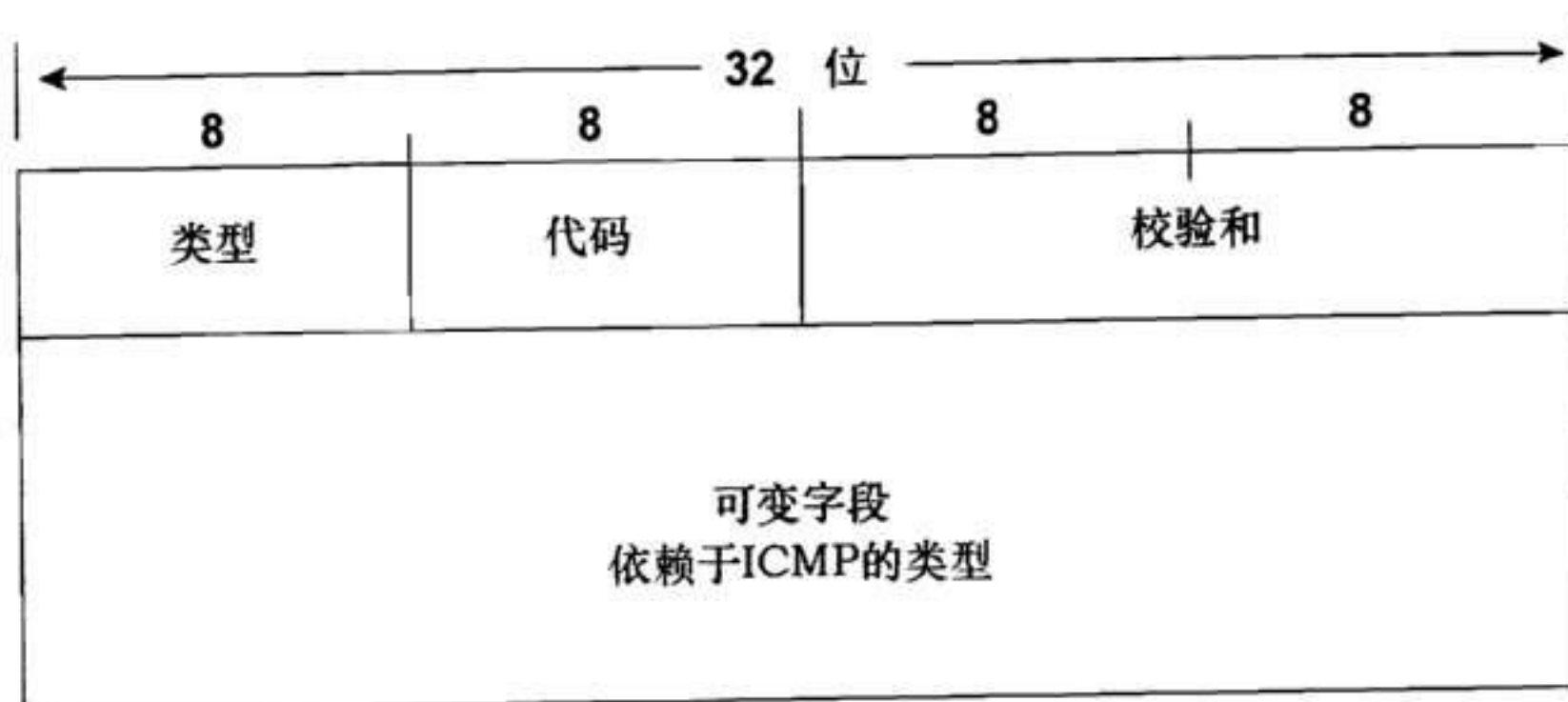


图 2-28 ICMP 报文头包括类型字段, 进一步标识某些类型的代码字段和校验和。剩余的字段依赖于特定类型和代码

表 2-5

ICMP 报文类型字段和代码字段

类 型	代 码	名 称
0	0	回应应答
3		目的地不可达
	0	网络不可达
	1	主机不可达
	2	协议不可达
	3	端口不可达
	4	需要分片和不需要分片标记置位
	5	源路由失败
	6	目的网络未知
	7	目的主机未知
	8	源主机被隔离
	9	与目的网络的通信被禁止
	10	目的主机的通信被禁止
	11	对请求的服务类型, 目的网络不可达
	12	对请求的服务类型, 目的主机不可达
4	0	源抑制 (Source Quench)
5		重定向
	0	为网络 (子网) 重定向数据报
	1	为主机重定向数据报
	2	为网络和服务类型重定向数据报
	3	为主机和服务类型重定向数据报
6	0	选择主机地址

续表

类 型	代 码	名 称
8	0	回应
9	0	路由器通告
10	0	路由器选择
11		超时
	0	传输中超出 TTL
	1	超出分片重组时间
12		参数问题
	0	指定错误的指针
	1	缺少需要的选项
	2	错误长度
13	0	时间戳
14	0	时间戳回复
15	0	信息请求 (废弃)
16	0	信息回复 (废弃)
17	0	地址掩码请求
18	0	地址掩码回复
30		跟踪路由
31		数据报会话错误
32		移动主机重定向
33		IPv6 你在哪里
34		IPv6 我在这里
35		移动注册请求
36		移动注册回复

图 2-29 和 2-30 给出了协议分析器捕捉到的两种众所周知的 ICMP 报文,它们常用在 ping 功能中。

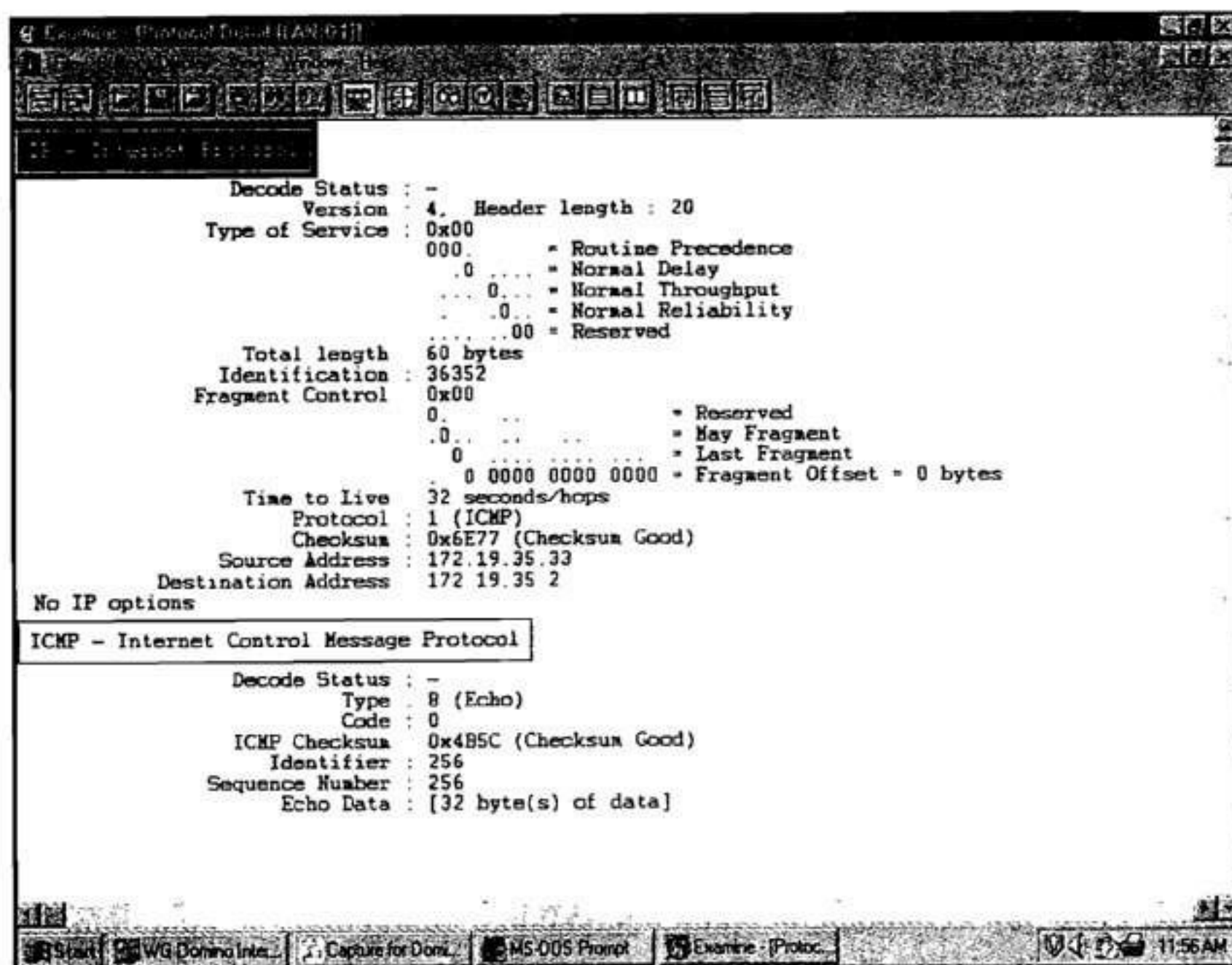


图 2-29 ICMP 回送报文及其 IP 头部

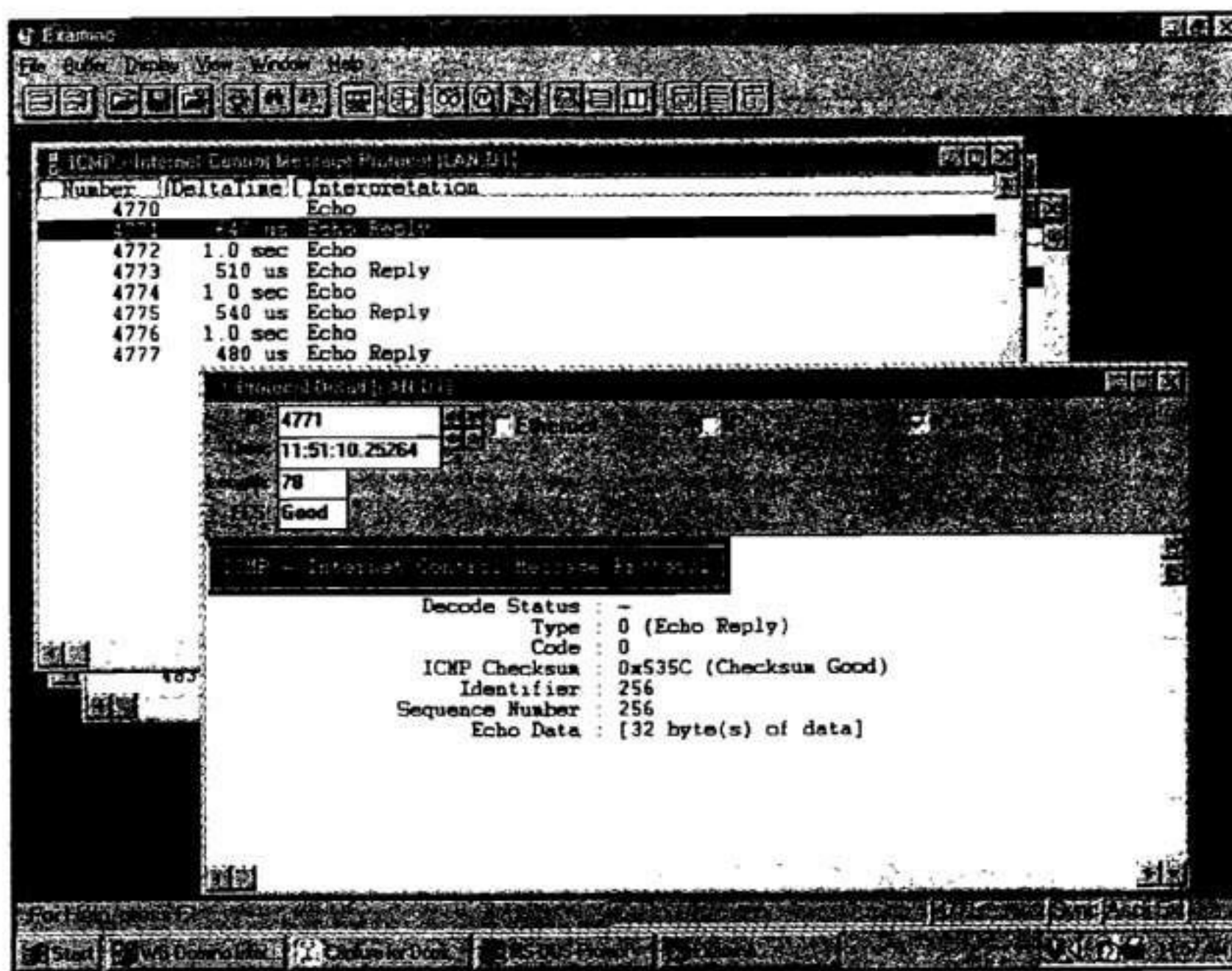


图 2-30 不带 IP 头的 ICMP 回送应答报文。在背景中的摘要窗口给出了 4 个回送/回送应答对，它们组成了 4 组 ping。

虽然大部分 ICMP 类型都与路由选择功能有关，但是有 3 个类型特别重要。

- **路由器通告 (Router Advertisement)** 和 **路由器选择 (Router Selection)** ——分别是类型 9 和类型 10，它们用于 ICMP 路由器发现协议 (IRDP)。
- **重定向 (Redirection)** ——类型 5，被路由器用于通知主机去往指定目标的网关，是数据链路上的另一个路由器。假设路由器 A 和 B 连接在相同的以太网上，主机 X 也在以太网上，而且 X 还把路由器 4 配置为自己的缺省网关。如果主机向路由器 A 发送报文，而 A 发现该报文目的地址需通过路由器 B 才可以到达（即路由器 A 必须在接收此报文的端口再次转发此报文）。路由器 A 不仅要向路由器 B 转发报文，而且还要向主机 X 发送 ICMP 重定向消息，通知它如果继续向特定的目标发送报文，那么请直接将报文发送给路由器 B。图 2-31 显示出路由器发送了一个重定向消息。

```
Pip#debug ip icmp
ICMP packet debugging is on
ICMP: redirect sent to 10.158.43.25 for dest 10.158.40.1, use gw
10.158.43.10
0
Pip#
```

图 2-31 使用调试功能 **debug ip icmp**，可以看到路由器向主机 10.158.43.25 发送了一个重定向消息，通知它到达目的地 10.158.40.1 的正确网关应该是路由器 10.158.43.10

当数据链路上连接多个路由器时，避免报文重定向的常用窍门是将每一个主机的缺省网关设置为主机自己的 IP 地址。于是主机对任何目的地址都会发送 ARP 请求，当目的地址不属于本地数据链路时，合适的路由器将通过代理 ARP 功能回复请求。使用这种策略避免重定向是有争议的，因为重定向会被减少或消除了，但是 ARP 的流量又增加了。

在 Cisco 路由器上，缺省状态下重定向功能是打开的。在接口上使用命令 **no ip redirects** 可以关闭此功能。

2.6 主机到主机层

TCP/IP 协议的主机到主机层的命名恰如其分。尽管互联网络层负责网络之间的逻辑路径，但主机到主机层是负责两个在完全不同网络¹上的主机之间的全程逻辑路径。从另一个角度看，主机到主机层向应用提供了一个到协议族下一层的接口，使应用不必关心它们的数据实际上是如何被传送的。

可以把这种服务比喻为公司的信件收发室。一个包裹被送到收发室，并附有邮寄要求（平信或隔日送到）。提出邮寄要求的人不需要知道或可能不关心实际是怎样邮寄此包裹的。收发室的工作人员将会安排合适的邮寄方式来满足其要求（送邮局邮寄、FedEx、交给骑自行车横穿城镇送快件的人）。

主机到主机层提供两个主要的服务：TCP 和 UDP

2.6.1 TCP

传输控制协议（TCP），向应用提供了可靠的、面向连接的服务，详见 RFC793。换句话说，TCP 提供了一个类似于点到点的连接。

点到点连接有两个特点：

- 仅存在一条路径到达目的地。进入连接的报文不会丢失，因为报文惟一可去的地方就是连接的另一端。
- 报文到达的顺序与发送顺序相同。

TCP 提供了一条看似点到点的连接，虽然实际上这条连接并不存在。TCP 利用的互联网络层可以提供无连接的、尽力转发的服务。这类似于邮政服务。一叠信一旦交给邮递员后，谁也不能保证信件将按照原先叠放的顺序依次送达，也不能保证信件都将在同一天送达，甚至不能保证全部送达。邮政服务仅仅能承诺尽最大努力邮寄这些信件。

同样，互联网络层不保证所有的报文使用相同的路径，因而也不保证报文到达时仍旧保持发送时的顺序和间隔或者全部到达。

另一方面，电话呼叫是一个面向连接的服务。数据必须顺序、可靠地到达，否则数据就会作废。像电话呼叫一样，TCP 首先必须建立连接，然后是传送数据，当数据传送完成后要拆除连接。

- 在无连接服务之上，TCP 使用了 3 种基础的机制实现面向连接的服务：
 - 使用序列号对报文进行标记，以便 TCP 接收服务在向目的应用传递数据之前修正错序的报文排序。
 - TCP 使用确认、校验和定时器系统提供可靠性。当接收者按照顺序识别出报文未能到达或发生错误时，接收者将通知发送者，或者接收者在特定时间内没有发送确认信息，那么发送者就认为在发送结束后报文没有到达接收方。在这两种情况下，发送者都会考虑重传报文。

¹ 类似的，主机到主机层也可以看作在传输层之上，功能上等同于 OSI 的会话层，在两个跨域互联网络的应用之间提供逻辑的、端到端的路径。

- TCP 使用窗口机制调整报文的流量；窗口机制可以减少因接收方缓冲区满而造成丢失报文的可能性。

TCP 在应用层数据上附加了一个报头，报头包括序列号字段和这些机制的其他一些必要信息，如叫做端口号的地址字段，该字段可以标识数据的源点和目标应用程序。为了传送数据，应用数据及附加的 TCP 报头被封装在 IP 报文内。图 2-32 显示了 TCP 数据报头字段，图 2-33 显示了协议分析仪获取的 TCP 报头信息。

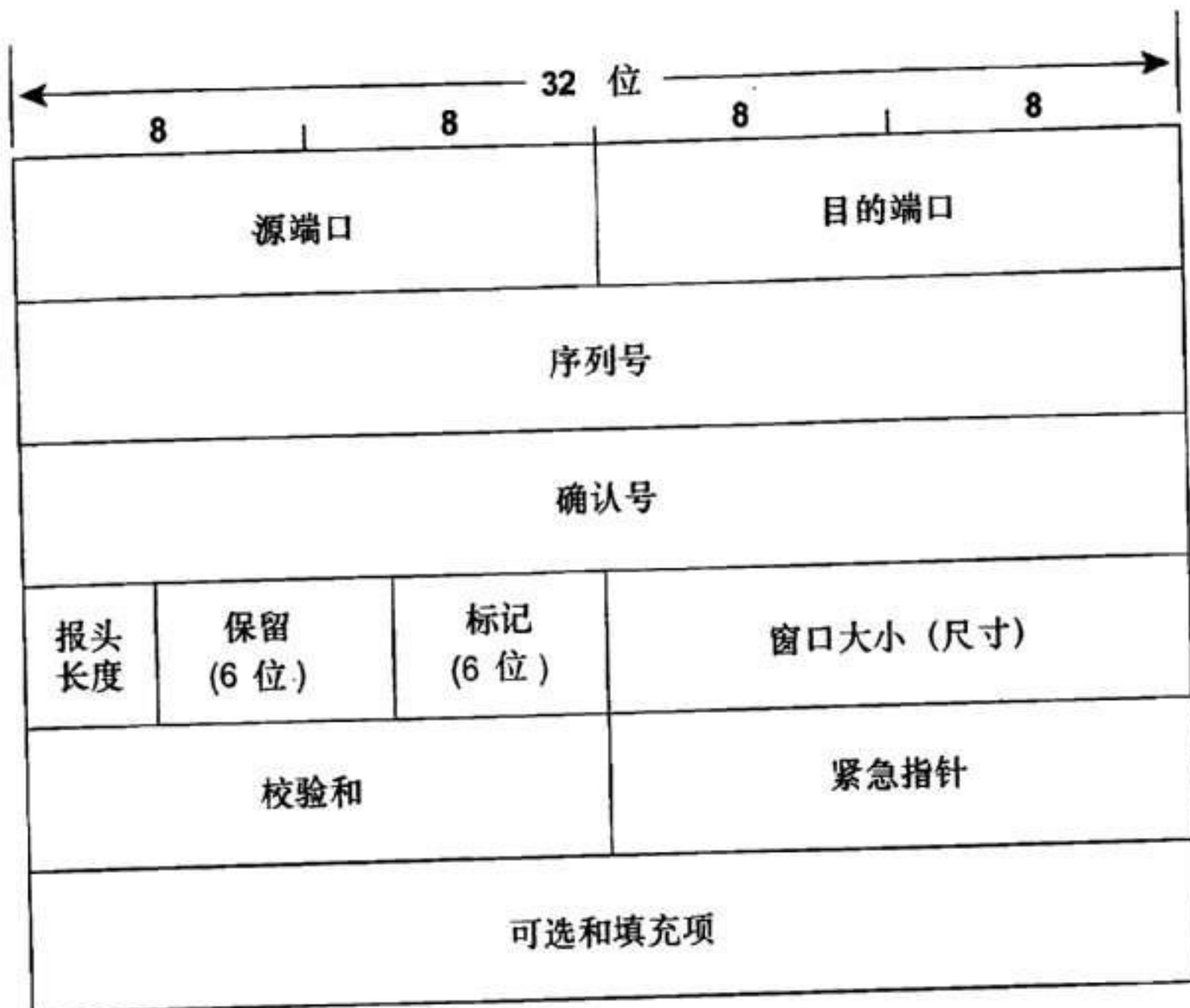


图 2-32 TCP 报头格式

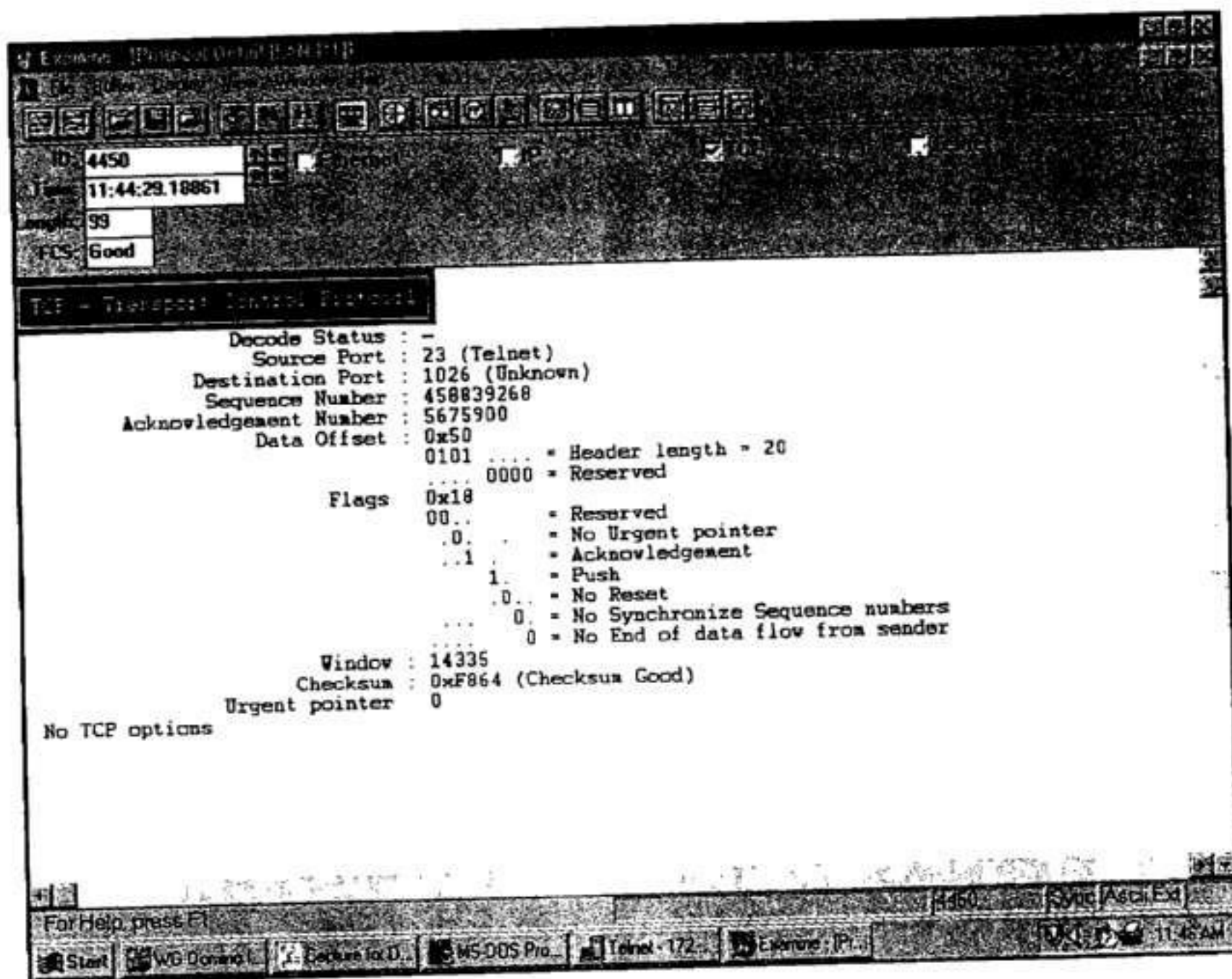


图 2-33 协议分析器显示出的 TCP 报头

- **源端口(Source Port)**和**目的端口(Destination Port)**——字段长度各为 16 位，它们为封装的数据指定了源和目的应用程序。像 TCP/IP 使用其他编号一样，RFC1700 描述了所有常用和不常用的端口号。应用程序的端口号加上应用程序所在主机的 IP 地址统称为套接字 (**Socket**)。在互联网上套接字惟一地标识了每一个应用程序。
- **序列号 (Sequence Number)**——字段长度为 32 位，序列号确定了发送方发送的数据流中被封装的数据所在位置。例如，如果本段数据的序列号为 1343，且数据段长 512 个 8bit 字节，那么下一数据段的序列号应该为 $1343 + 512 + 1 = 1856$ 。
- **确认号 (Acknowledgment Number)**——字段长度为 32 位，确认号确定了源点下一次希望从目标接收的序列号。如果主机收到的确认号与它下一次打算发送（或已发送）的序列号不符，那么主机将获悉报文不仅丢失，而且确切地知道丢失了哪一个报文。
- **报头长度(Header Length)**——又叫数据偏移量，长度为 4 位，报头长度指定了以 32 比特字为单位的报头长度。由于可选项字段的长度可变，所以这一字段标识出数据的起点是很有必要的。
- **保留 (Reserved)**——字段长度为 6 位，通常设置为 0。
- **标记 (Flag)**——包括 6 个 1 位标记，它们用于数据流控和连接控制。这些标记是紧急 (URG)、确认 (ACK)、弹出 (PSH)、复位 (RST)、同步 (SYN) 和结束 (FIN)。
- **窗口大小 (WindowSize)**——字段长度为 16 位，主要用于流控制。窗口大小指明了自确认号指定的 8bit 字节开始，接收方在必须停止传输并等待确认之前发送方可以接收的数据段的 8bit 字节长度。
- **校验和 (Checksum)**——字段长度为 16 位，它包括报头和被封装的数据，校验和允许错误检测。
- **紧急指针 (Urgent Point)**——字段仅当 URG 标记置位时才被使用。这个 16 位数被添加到序列号上用于指明紧急数据的结束。
- **可选项 (Options)**——字段用于指明 TCP 的发送进程要求的选项。最常用的可选项是最大段长度，最大段长度通知接收者发送者愿意接收的最大段长度。为了保证报头的长度是 32 个 8bit 字节的倍数，所以使用 0 填充该字段的剩余部分。

2.6.2 UDP

用户数据报协议 (UDP) 提供了一种无连接、尽力传送的报文转发服务，详见 RFC768。起初，对应用程序宁愿使用不可靠的转发服务，而不用面向连接的 TCP 服务，感觉很有疑问。然而 UDP 的优点是不花时间建立连接，直接发送数据。用 UDP 代替 TCP，可以使发送小数据量的应用取得更好的性能优势。

图 2-34 给出了 UDP 的另一个优点：UDP 报头长度远远小于 TCP 报头长度。UDP 报头中的源端口和目的端口字段与 TCP 完全相同，UDP 的长度指明了以 8bit 字节为单位的整个段长度。校验和包括整个段，但是不同于 TCP，在这里，校验和是可选的，当不使用校验和时，此字段全部设置为 0。在图 2-35 中显示出协议分析器捕捉到的 UDP 报头。

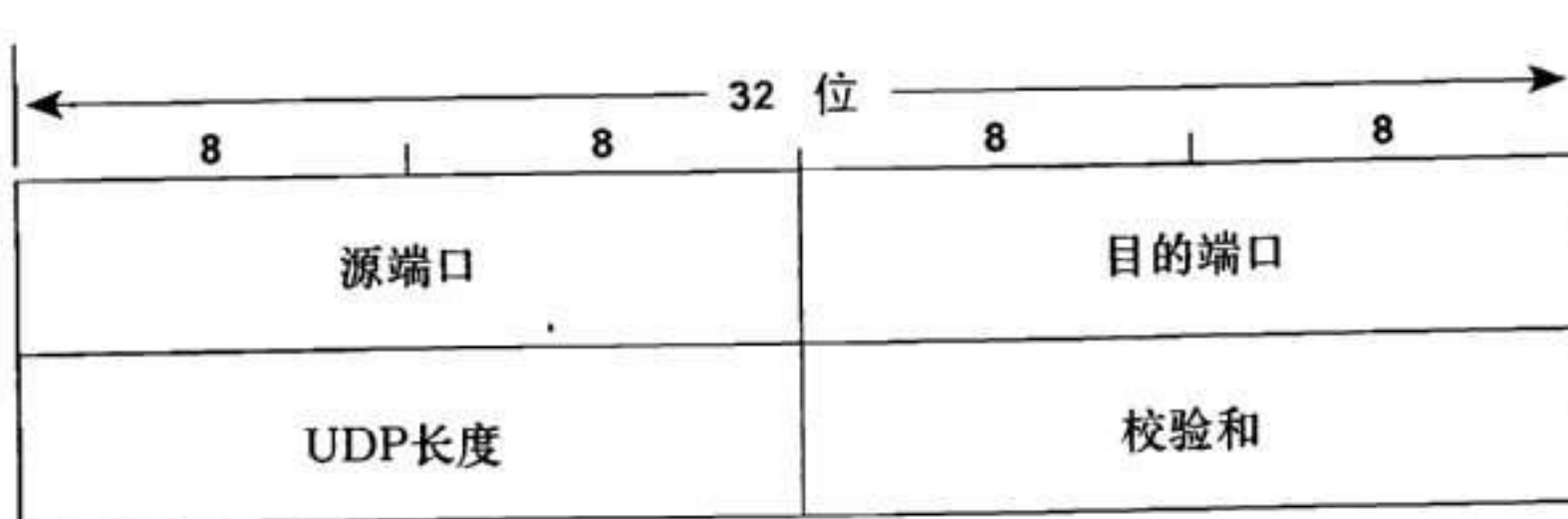


图 2-34 UDP 报头格式

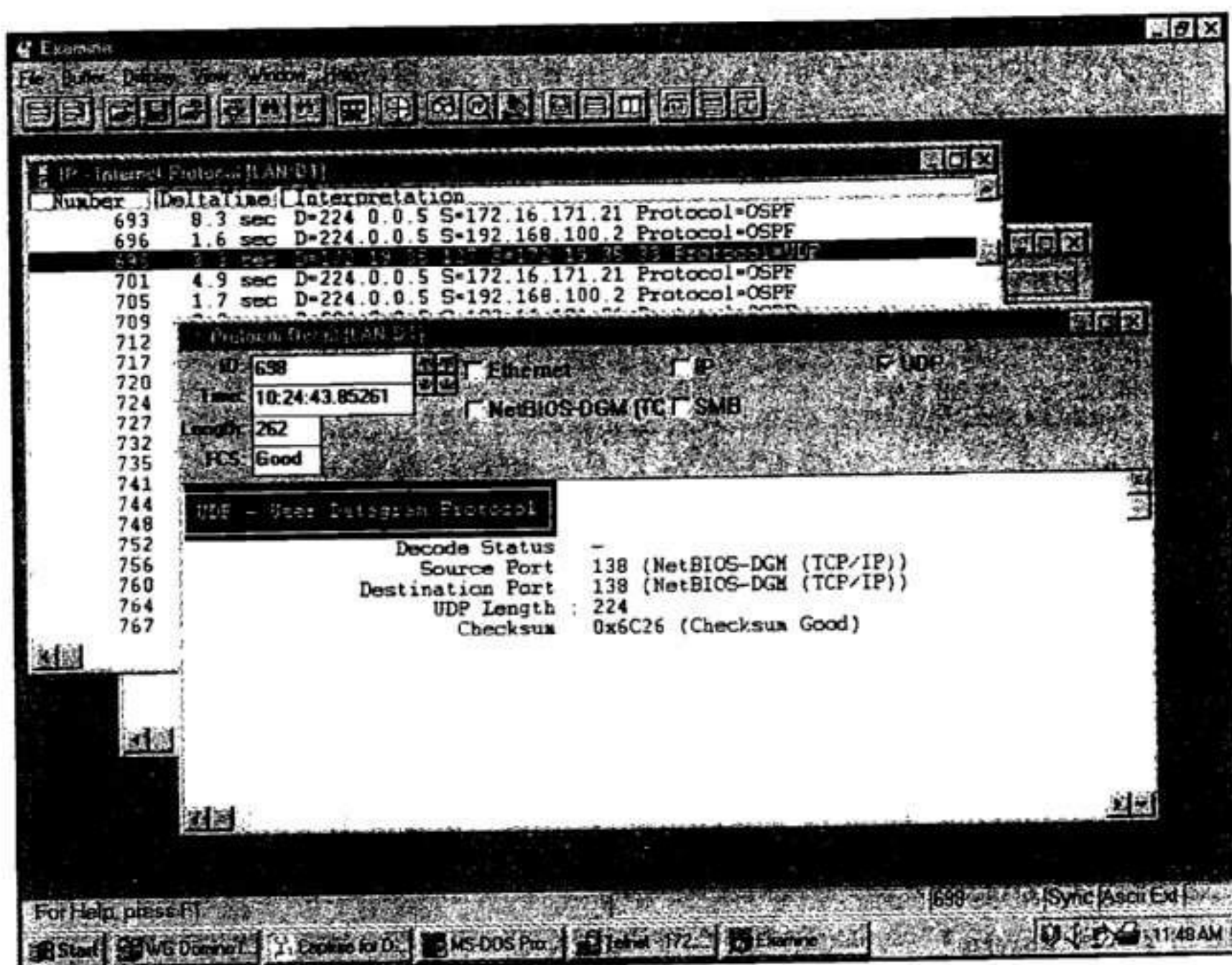


图 2-35 协议分析器显示的 UDP 报头

2.7 展 望

本章重点讲述了设备互连网络层（或 OSI 网络层）的自我标识机制以及网络层是怎样映射到网络接口层（OSI 数据链路）的。本章还分析了互连网络层的功能对路由选择的重要性。下一章将研究路由选择功能及路由器执行此功能所需要的信息。

2.8 总结表：第 2 章命令回顾

命 令	描 述
<code>arp ip-address hardware-address</code>	静态地映射 IP 地址类型（别名）到硬件地址
<code>arp timeout seconds</code>	设置 Cisco 路由器保留 ARP 表项的时间值
<code>clear arp-cache</code>	强制从 ARP 表中删除所有动态表项
<code>debug ip icmp</code>	显示在路由器上出现的 ICMP 事件

续表

命 令	描 述
ip address ip-address mask	为接口分配 IP 地址和掩码
ip netmask-format{bitcount decimal hexadecimal}	配置路由器，使路由器可以用位计数、点分十进制和十六进制方式显示 IP（地址，掩码）对
ip proxy-arp	启用代理 ARP
ip redirects	启用 ICMP 重定向功能

2.9 推荐读物

Baker, F., ed. "Requirements for IP Version 4 Routers," RFC 1812, June 1995.

这篇文章给出了对要运行 IP 协议的路由器的要求和建议。

Braden, R., ed. "Requirements for Internet Hosts—Communication Layers," RFC 1122, October 1989.

RFC 1812 的姊妹篇，中心内容是主机。

Comer, D. E. *Internetworking with TCP/IP*, Vol. 1. Englewood Cliffs, New Jersey: Prentice-Hall; 1991.

这本书，就像 Perlman 的著作一样，是一本经典书。尽管你不一定要把 Comer 和 Stenvens 的书都读了，但是如果都能阅读的话，对你绝不会有坏处。

Stevens, W. R. *TCP/IP Illustrated*, Vol. 1. Reading, Massachusetts: Addison-Wesley; 1994.

一本关于 TCP/IP 的好书。Stevens 在深入地介绍协议的同时，针对封二上的网络图提供了大量现实网络的细节。

2.10 复 习 题

1. TCP/IP 协议族的 5 个层次是什么？每一层的目的是什么？
2. 目前最常用的 IP 版本是什么？
3. 什么是分片？IP 报头的什么字段用于分片？
4. IP 报头中的 TTL 字段的用途是什么？TTL 过程是如何工作的？
5. 什么是首个 8bit 字节规则？
6. 怎样识别点分十进制表示的 A、B 和 C 类地址？怎样识别二进制表示的地址？
7. 什么是地址掩码？它是如何工作的？
8. 什么是子网？在 IP 环境中为什么使用子网？
9. 为什么在有类别路由选择环境中子网位不能全部为 0 或 1？
10. 什么是 ARP？
11. 什么是代理 ARP？
12. 什么是重定向？
13. TCP 和 UDP 的本质区别是什么？

14. TCP 提供面向连接服务的机制是什么?

15. 为了替代 ARP, Novell NetWare 用设备的 MAC 地址作为网络地址中的主机部分。为什么 IP 不能这样做?

16. NetWare 传输层服务与 TCP 很相似, 叫做顺序报文交换 (SPX), 但是没有类似 UDP 的服务。如果应用需要无连接服务, 它可直接访问网络层的无连接协议服务 IPX。那么 UDP 在无连接服务之上仍旧提供无连接服务的目的是什么?

2.11 配置练习

1. 首个 8bit 字节规则指出最高的 C 类地址是 223, 而我们知道 8bit 字节的最大十进制数是 255。因而还有两类地址, 一类是 D 类地址, 用于组播, 另一类是 E 类地址, 用于实验。其中 D 类地址的前 4 位为 1110。请问 D 类地址首个 8bit 字节的十进制数的范围是什么?

2. 为 10.0.0.0 选择一个子网掩码以便至少可以划分出 16 000 个子网, 并且每个子网至少拥有 700 个主机地址。为 172.27.0.0 选择子网掩码以便至少可以划分出 500 个子网, 并且每个子网至少拥有 100 个主机地址。

3. 如果 C 类地址有 6 个子网位, 那么可以划分出多个子网? 每个子网有多少个主机地址? 这样的子网规划有实际用途吗?

4. 对地址 192.168.147.0 进行子网划分, 子网掩码为 28 位, 请写出所有子网。试给出每个子网的可用主机地址。

5. 对地址 192.168.147.0 进行子网划分, 子网掩码为 29 位, 请写出所有子网。试给出每个子网的可用主机地址。

6. 对地址 172.16.0.0 进行子网划分, 子网掩码为 20 位, 试给出每个子网的可用主机地址。(地址按照最低到最高顺序给出)

2.12 故障排除练习

1. 根据以下主机地址和子网掩码, 试找出每个地址所属的子网, 并且找出该子网中的广播地址和可用主机地址的范围:

10.14.87.60/19
172.25.0.235/27
172.25.16.37/25

2. 请问接口上配置 IP 地址 192.168.13.175, 掩码 255.255.255.240, 会有问题吗? 如果有, 问题是什么?

第 3 章

静态路由

本章包括以下主题：

- 路由选择表
- 配置静态路由
 - 案例研究：简单静态路由
 - 案例研究：汇总路由
 - 案例分析：选择路由
 - 案例研究：浮动静态路由
 - 案例研究：均分负载
 - 案例研究：递归表查询
- 静态路由故障排除
 - 案例研究：追踪错误路由
 - 案例研究：协议冲突

在阅读完第 2 章“TCP/IP 回顾”之后，读者应该发现有一点十分重要，就是 OSI 模型定义数据链路层/物理层和传输层/网络层有着十分相似的职责：提供数据传输的手段，即沿某条路径将数据从源点传送到目的点。不同之处在于数据链路层/网络层提供跨越物理路径的通信服务，而传输层/网络层提供跨越一连串数据链路组成的逻辑或虚拟路径的通信服务。

此外，第 2 章讲述了沿物理路径进行通信，需要获取有关数据链路标识和数据封装的信息，并且要将它们保存在数据库中，如 ARP 高速缓冲。类似的，传输层/网络层为了完成本层的工作也必须对信息进行获取和保存。但这些信息被保存在路由选择表中，路由选择表又叫转发数据库。

本章所要研究的内容包括：需要什么样的信息来为报文选择路由，怎样在路由选择表中保存这些信息，如何向数据库中输入这些信息，以及通过向恰当的路由器的路由选择表

中输入正确的信息来建立一个可路由的互联网络相关技术。

3.1 路由选择表

为了理解路由选择表中存在的信息种类, 开始先分析一下当报文到达路由器的接口时会发生什么, 这对我们是很有帮助。首先路由器会检查数据帧目的地址字段中的数据链路标识。如果标识符是路由器接口标识符或广播标识符, 那么路由器将从帧中剥离出报文并传递给网络层。在网络层, 将检查报文的地址。如果目的地址是路由器接口的 IP 地址或是所有主机的广播地址, 那么需要再检查报文的协议字段, 然后再向适当的内部进程发送被封装的数据。¹

除此之外, 所有其他目的地址都需要进行路由选择。这里的地址可能是另一个网络上的主机地址, 该网络或者与路由器相连 (包括与那个网络相连接的路由器接口), 或者不直接连接到路由器上。目的地址还可能是一个定向的广播地址, 这种地址的网络地址或子网地址是不同的, 而主机位全部为 1。以上这些地址也是可以路由的。

如果报文是可以被路由的, 那么路由器将会查找路由选择表选择一个正确的路径。在数据库中的每个路由选择表项最少必须包括下面两个项目:

- **目的地址**——这是路由器可以到达的网络的地址。正像本章所解释的, 路由器可能会有多条路径到达相同的地址/或是到达相同主网 IP 地址下的一组等长或变长子网。
- **指向目的地的指针**——指针不是指向路由器的直连目的网络就是直连网络内的另一个路由器地址。更接近目标网络一跳的路由器叫**下一跳 (next hop)** 路由器。

路由器将会尽量地做最精确的匹配。²按精确程度递减的顺序, 可选地址排列如下:

- 主机地址 (主机路径);
- 子网;
- 一组子网 (一条汇总路由);
- 主网号;
- 一组主网号 (超网);
- 缺省地址。

本章将提供有关前 4 类的例子。超网将会在第 7 章中讨论。缺省地址是最不明确的地址, 只有当所有匹配都失败时才被使用。第 12 章会讨论缺省地址。

如果报文的地址不能匹配到任何一条路由选择表项, 那么报文将被丢弃, 同时目标网络不可达的 ICMP 消息将会被发送给源地址。

如图 3-1 所示, 这是一个简单的互联网络, 图中给出了每个路由器需要的路由选择表。这里最重要的是看这些路由选择表是如何作为一个整体来保证传输报文的工作准确高效地进行。路由选择表的网络栏目列出了路由器可达的网络地址。指向目标网络的指针在下一跳栏目。

¹ 有一种特殊情况, 目的地址是组播地址, 这时报文是发向一组设备, 而不是所有设备。D 类地址 224.0.0.5 就是一个组播地址, 它是为所有 OSPF-speaking 路由器保留的。

² 寻找最优匹配有两个基本过程, 它们依赖于路由器是否表现为有类别或无类别。有类别路由选择表的查找过程详见第 5 章, 无类别路由选择表查找见第 7 章

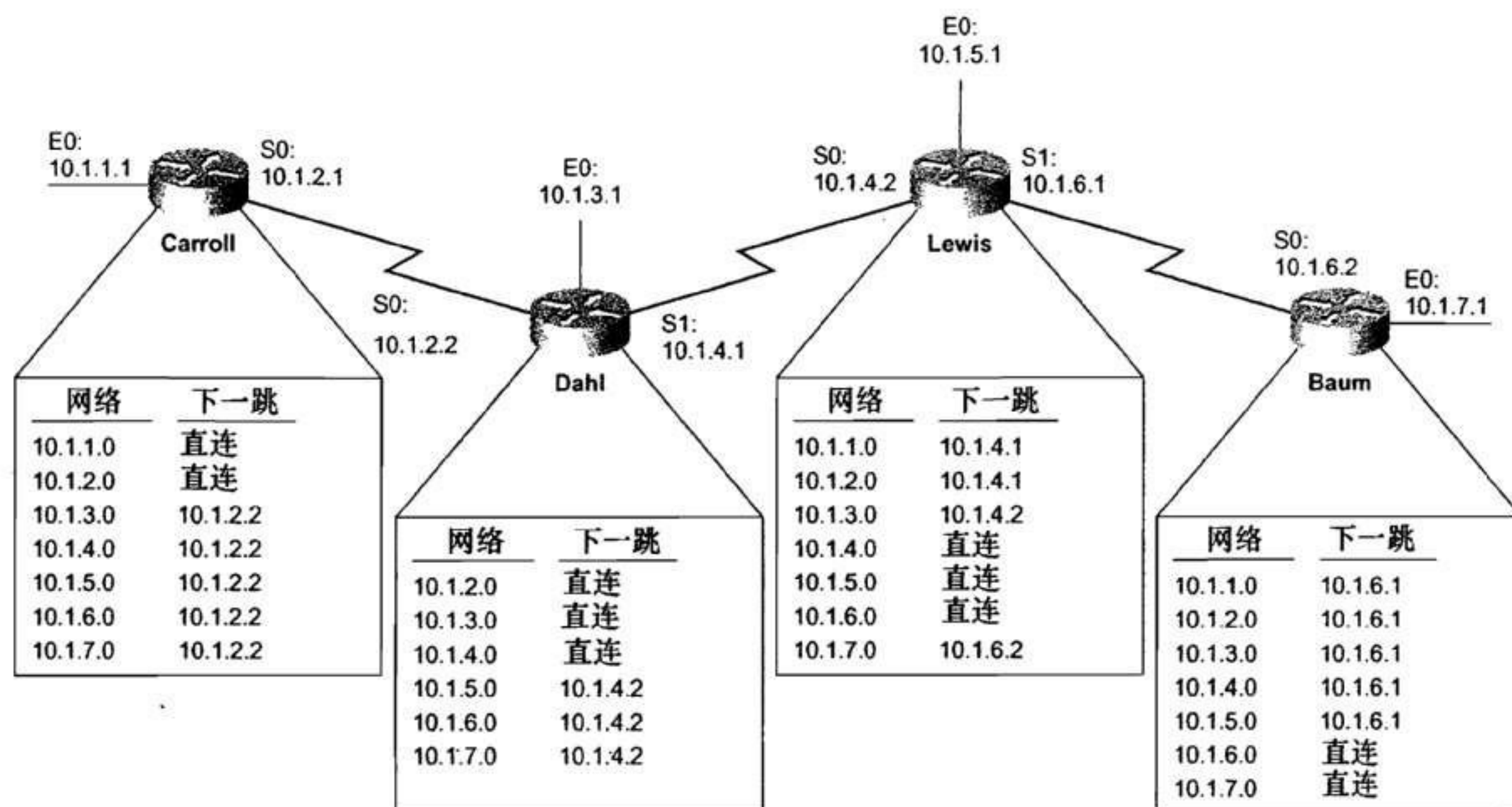


图 3-1 每一个路由选择表项目需要的信息至少应该包括目的地址和指向目的地址的指针

在图 3-1 中，如果路由器 Carroll 收到一个源地址为 10.1.1.97、目的地址为 10.1.7.35 的报文，那么路由选择表查询的结果对于目的地址 10.1.7.0 的最优匹配是子网 10.1.7.0，报文可以从接口 S0 出站经下一跳地址 10.1.2.2 去往目的地。接着报文被发送给路由器 Dahl，Dahl 查找路由选择表后发现报文应该从接口 S1 出站经下一跳 10.1.4.2 去往目的网络 10.1.7.0。此过程将一直持续到报文到达路由器 Baum。当 Baum 在接口 S0 接收到报文时，Baum 通过查找路由表发现目的地是连接在端口 E0 的一个直连网络。最终结束路由选择过程，报文被传递给以太网上的主机 10.1.7.35。

上面说明的路由选择过程是假设路由器可以将下一跳地址同它的接口匹配起来。例如，路由器 Dahl 必须知道通过接口 S1 可以到达 Lewis 的地址 10.1.4.2。首先 Dahl 从分配给接口 S1 的 IP 地址和子网掩码可以知道子网 10.1.4.0 直接连接在接口 S1 上。那么 Dahl 就可以得出结论 10.1.4.2 是子网 10.1.4.0 的成员，而且一定被连接到该子网上。

注意：为了正确地进行报文交换，每个路由器都必须保持信息的一致性和准确性。例如，图 3-1 中，在路由器 Dahl 的路由选择表中丢失了关于网络 10.1.1.0 的表项。从 10.1.1.97 到 10.1.7.35 的报文将被传送，但是当 10.1.7.35 向 10.1.1.97 回复报文时，报文从 Baum 到 Lewis 再到 Dahl。Dahl 查找路由选择表后发现没有关于子网 10.1.1.0 的路由表项，因此丢弃此报文，同时 Dahl 向主机 10.1.7.35 发送目标网络不可达的 ICMP 信息。

图 3-2 给出了在 Cisco 路由器中图 3-1 中路由器 Lewis 的实际路由选择表。在 Cisco 路由器中查看路由选择表的命令是 **show ip route**。

检查数据库的内容并把它与图 3-1 中路由器 Lewis 的普通路由选择表相比较。可以看到，表最上方的关键字是对路由选择表左侧的一列字母的解释。这些字母指明了每个路由表项是如何学习到的。在图 3-2 中，标记为 C 的路由表示直连网络，标记为 S 的路由选择表示静态路由。声明 “Gateway of last resort is not set” 指的是缺省路由。


```

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

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 7 subnets
S      10.1.3.0 [1/0] via 10.1.4.1
S      10.1.2.0 [1/0] via 10.1.4.1
S      10.1.1.0 [1/0] via 10.1.4.1
S      10.1.7.0 [1/0] via 10.1.6.2
C      10.1.6.0 is directly connected, Serial1
C      10.1.5.0 is directly connected, Ethernet0
C      10.1.4.0 is directly connected, Serial0
Lewis#

```

图 3-2 图 3-1 中路由器 Lewis 对应于 Cisco 路由器的路由选择表

路由选择表的最上面有一句声明主网络地址 10.0.0.0 有 7 个已知子网，掩码为 24 位。在 7 个路由表项中，每一个都给出了目标子网。对于不是直连网络的表项——报文必须转发到下一跳路由器——置于括号内的元组指明了路由的[管理距离/度量]。管理距离将会在本章后面部分介绍，在第 11 章还将详细讨论。

度量是通过优先权评价路由的一种手段，度量越低，路径越短，在第 4 章中将会详细讨论度量。注意，在图 3-2 中静态路由的度量为 0。最后，路由选择表还给出了下一跳路由器直接被连接的接口地址或者目标网络连接的接口地址。

3.2 配置静态路由

路由选择表获取信息的方式有两种，以静态路由表项的方式手工输入信息，或者通过几种自动信息发现和共享系统（动态路由选择协议）之一自动地获取信息。本书将花大量篇幅讲述动态 IP 路由选择协议，但是讨论一下静态路由配置可以为读者理解后继章节做好准备。

除了上述目的外，在某些场合，人们宁愿选用静态路由，而不是动态路由。对于任何程序而言，自动化程度越高，可控程度就越差。虽然动态（自动）路由要求更少的人为干涉，但静态路由允许在互联网的路由选择行为上实施非常精确的控制。然而为此付出的代价是每当网络拓扑结构发生变化时都需要重新进行手工配置。

3.2.1 案例研究：简单静态路由

如图 3-3 所示，互联网络有 4 个路由器和 6 个网络。注意，网络 10.0.0.0 的几个子网是不连续的——有一个不同主网的子网（在 Tigger-to-Piglet 链路上的 192.168.1.192）将子网 10.1.0.0 与 10.0.0.0 的其他子网分离开了。并且 10.0.0.0 的子网还是变长子网——整个互联网络中的子网掩码长度不一致。最后要说的是，路由器 Pooh 以太网链路上的子网地址是一个全 0 子网。在后面章节可以看到这种编址方式对于更简单的有类别路由选择协议，如 RIP 和

IGRP，将会产生一些问题；而静态路由则工作的很正常。

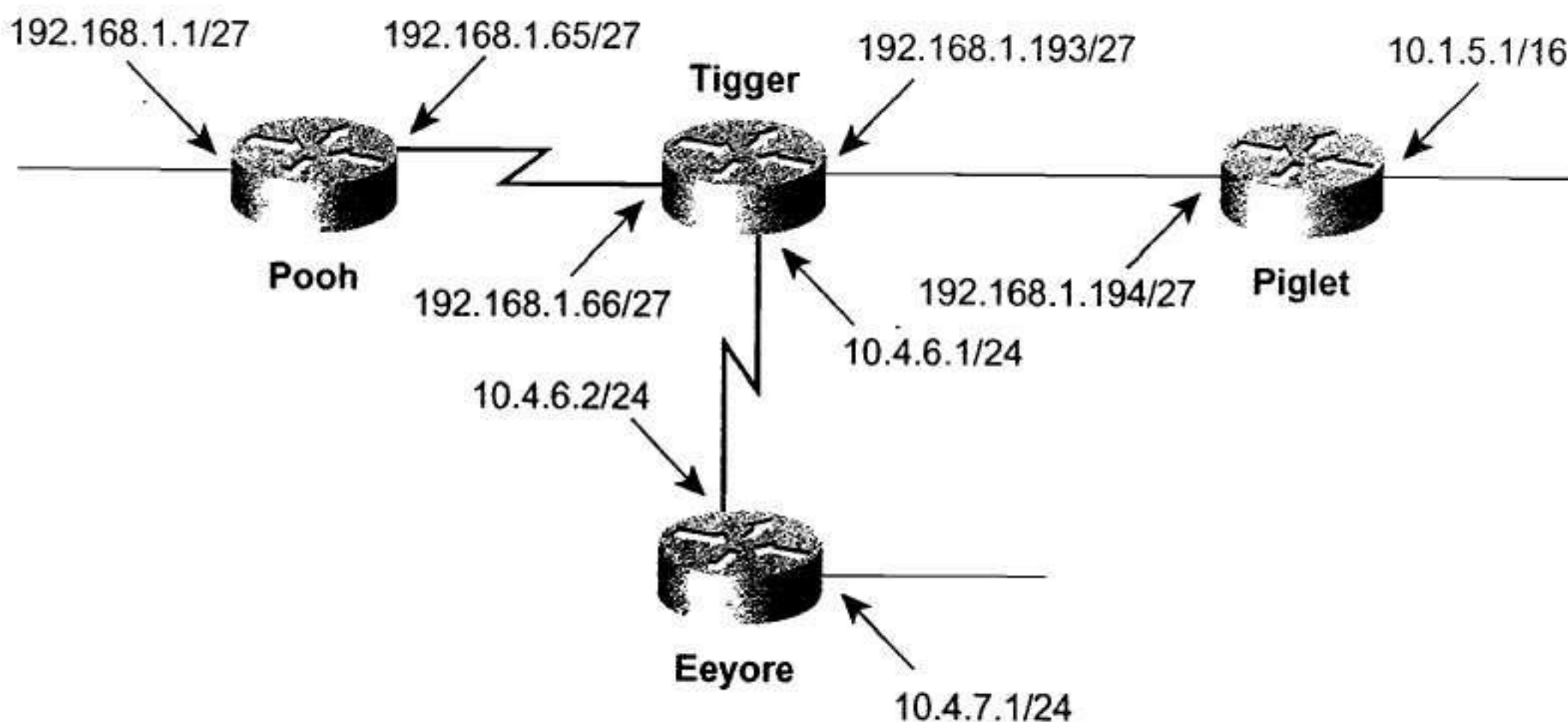


图 3-3 路由选择协议 RIP 和 IGRP 对于含有非连续和变长子网的互联网络来说不能进行正常的路由，而静态路由可以

在互联网上实施静态路由选择的过程共有 3 步：

步骤 1：为互联网络中的每个数据链路确定地址（包括子网和网络）。

步骤 2：为每个路由器标识所有非直连的数据链路。

步骤 3：为每个路由器写出关于每个非直连数据链路的路由说明。

这里没有必要写出有关直连数据链路的路由描述，因为在路由器接口上配置的地址和掩码可以使这些直连网络被记录在路由选择表中。

例如，在图 3-3 中的 6 个子网是：

- 10.1.0.0/16
- 10.4.6.0/24
- 10.4.7.0/24
- 192.168.1.192/27
- 192.168.1.64/27
- 192.168.1.0/27

为了在路由器 Piglet 上配置静态路由，将那些非直连的子网标识如下：

- 10.4.6.0/24
- 10.4.7.0/24
- 192.168.1.64/27
- 192.168.1.0/27

对于静态路由来说，这些子网必须被记录下来。在路由器 Piglet 上配置静态路由的命令如下：¹

```
Piglet(config)# ip route 192.168.1.0 255.255.255.224 192.168.1.193
Piglet(config)# ip route 192.168.1.64 255.255.255.224 192.168.1.193
Piglet(config)# ip route 10.4.6.0 255.255.255.0 192.168.1.193
Piglet(config)# ip route 10.4.7.0 255.255.255.0 192.168.1.193
```

¹ 为了本例及本章后续例子中的静态路由正常工作，必须附加两个全局配置命令：**ip classless** 和 **ip subnet-zero**。在这里提及这两个命令主要考虑到希望在实验室中尝试这个配置实例的读者，第 7 章将介绍这些命令。

对其他 3 个路由器也执行相同的步骤:

```
Pooh(config)# ip route 192.168.1.192 255.255.255.224 192.168.1.66
Pooh(config)# ip route 10.1.0.0 255.255.0.0 192.168.1.66
Pooh(config)# ip route 10.4.6.0 255.255.255.0 192.168.1.66
Pooh(config)# ip route 10.4.7.0 255.255.255.0 192.168.1.66

Tigger(config)# ip route 192.168.1.0 255.255.255.224 192.168.1.65
Tigger(config)# ip route 10.1.0.0 255.255.0.0 192.168.1.194
Tigger(config)# ip route 10.4.7.0 255.255.255.0 10.4.6.2

Eeyore(config)# ip route 192.168.1.0 255.255.255.224 10.4.6.1
Eeyore(config)# ip route 192.168.1.64 255.255.255.224 10.4.6.1
Eeyore(config)# ip route 192.168.1.192 255.255.255.224 10.4.6.1
Eeyore(config)# ip route 10.1.0.0 255.255.0.0 10.4.6.1
```

如果读者记住每条命令描述一个路由表项的话, 路由配置命令本身是很容易阅读的。命令 **ip route** 后面跟着的是将要被输入到路由选择表中的地址、确定地址网络号的掩码及直接连接下一跳路由器的接口地址。

配置静态路由还可以选择另一种命令, 这种命令用出站接口代替下一跳路由器地址, 其中通过出站接口可以到达目标网络。例如, 可以这样配置路由器 Tigger 的路由选择表:

```
Tigger(config)# ip route 192.168.1.0 255.255.255.224 S0
Tigger(config)# ip route 10.1.0.0 255.255.0.0 E0
Tigger(config)# ip route 10.4.7.0 255.255.255.0 S1
```

图 3-4 比较了两种配置方法所产生的路由选择表。注意, 这里引入了一个错误, 所有用

<pre>Tigger#show ip route Gateway of last resort is not set 10.0.0.0 is variably subnetted, 3 subnets, 2 masks C 10.4.6.0 255.255.255.0 is directly connected, Serial1 S 10.4.7.0 255.255.255.0 [1/0] via 10.4.6.2 S 10.1.0.0 255.255.0.0 [1/0] via 192.168.1.194 192.168.1.0 255.255.255.224 is subnetted, 3 subnets C 192.168.1.64 is directly connected, Serial0 S 192.168.1.0 [1/0] via 192.168.1.65 C 192.168.1.192 is directly connected, Ethernet0 Tigger#</pre>	<pre>Tigger#show ip route Gateway of last resort is not set 10.0.0.0 is variably subnetted, 3 subnets, 2 masks C 10.4.6.0 255.255.255.0 is directly connected, Serial1 S 10.4.7.0 255.255.255.0 is directly connected, Serial1 S 10.1.0.0 255.255.0.0 is directly connected, Ethernet0 192.168.1.0 255.255.255.224 is subnetted, 3 subnets C 192.168.1.64 is directly connected, Serial0 S 192.168.1.0 is directly connected, Serial0 C 192.168.1.192 is directly connected, Ethernet0 Tigger#</pre>
---	--

图 3-4 上面的路由选择表是指向下一跳路由器的静态路由表项产生的结果,

下面是指向出站接口到达目标网络的静态路由表项产生的路由选择表¹

¹ 为了表达清楚, 删除了路由器上方的关键字。

静态路由指明的网络，如果静态路由参照出站接口，那么它们将被作为直连网络输入到路由选择表中。这里所涉及到路由重新分配的问题将在第11章中讨论。

在图3-4中，令人感兴趣的一点是在10.0.0.0子网标题中指明了互联网络中使用了变长子网掩码。变长子网掩码（VLSM）是一种很有用的工具，我们会在第7章中讨论VLSM。

3.2.2 案例研究：汇总路由

汇总路由（Summary Route）是一个包含路由选择表中几个更加精确地址的地址。正是由于路由表项与地址掩码联合使用，使得静态路由的使用如此灵活。通过使用合适的子网掩码，有时可以为多个目标地址生成一条单一的汇总路由。

例如，在前面的案例研究中，为每个数据链路都使用了一条单独的路由。每个路由表项的掩码与连接数据链路的设备接口的地址掩码是一致的。再回想一下图3-3，读者可以发现对于路由器Piglet来说，可以使用经路由器Tigger可达10.4.0.0/16的单一表项来完成对子网10.4.6.0/24和10.4.7.0/24的说明。同样，也可以用指向192.168.1.0/24单一表项替代路由选择表中的子网192.168.1.0/27和192.168.1.64/27。10.4.0.0/16和192.16.1.0/24两个路由表项就是汇总路由。

使用汇总路由，路由器Piglet的静态路由选择表配置如下：

```
Piglet(config)# ip route 192.168.1.0 255.255.255.0 192.168.1.193
Piglet(config)# ip route 10.4.0.0 255.255.0.0 192.168.1.193
```

因为从路由器Pooh看，10.0.0.0网络的所有子网都可以经过路由器Tigger到达，所以对于10.0.0.0的所有子网仅需要该主网的一条单一的表项就可以表示：

```
Pooh(config)# ip route 192.168.1.192 255.255.255.224 192.168.1.66
Pooh(config)# ip route 10.0.0.0 255.0.0.0 192.168.1.66
```

从路由器Eeyore看，所有以192开头的目的地址都可以经过Tigger到达。因此可以用一条不恰好指明所有C类地址的单一路由表项：¹

```
Eeyore(config)# ip route 192.0.0.0 255.0.0.0 10.4.6.1
Eeyore(config)# ip route 10.1.0.0 255.255.0.0 10.4.6.1
```

通过对一组子网甚至主网汇总，可以使静态路由项的数目迅速减小——在本例中减少了三分之一。但是，在对地址进行汇总时需要小心，当汇总不正确时，可能会有意想不到的路由行为发生（见本章后面的案例研究：追踪错误路由）。第8章和第9章将会深入分析路由汇总及由此带来的问题。

3.2.3 案例研究：选择路由

在图3-5中，在Pooh和Eeyore之间新增加了一条链路。除了去往主机10.4.7.25的报文外，所有从Pooh到网络10.0.0.0的报文都将使用这条新的路径。下面通过一条策略可以使这

¹ 用小于特定类别地址标准掩码长度的掩码汇总一组主网地址的方法叫做超网化，详见第7章。

些报文改经 Tigger 去往目的地。在 Pooh 上, 相应的静态路由命令如下:

```
Pooh(config)# ip route 192.168.1.192 255.255.255.224 192.168.1.66
Pooh(config)# ip route 10.0.0.0 255.0.0.0 192.168.1.34
Pooh(config)# ip route 10.4.7.25 255.255.255.255 192.168.1.66
```

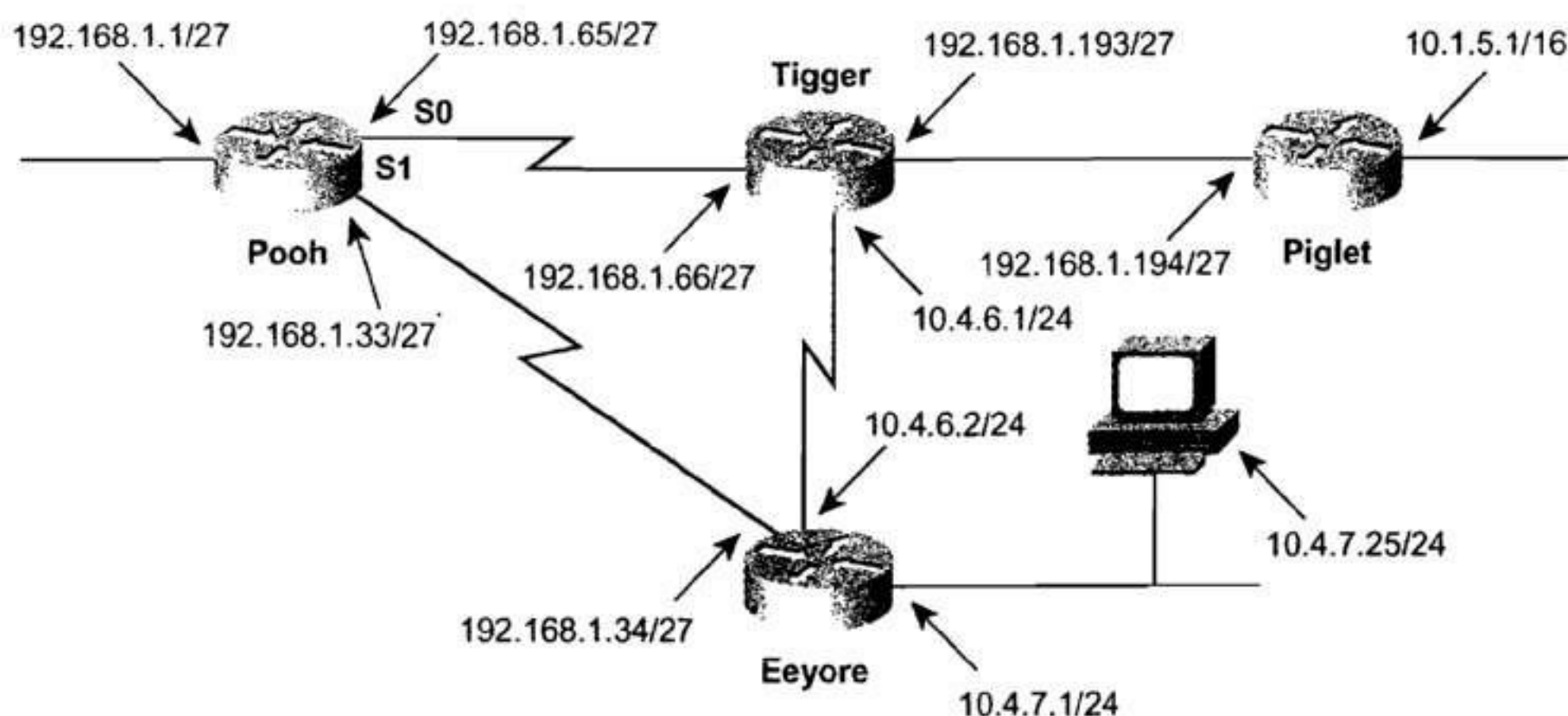


图 3-5 在互联网中添加了一条从 Pooh 到子网 10.4.0.0 更加直接的路径

目前, 除了第 2 条路由指向在 Eeyore 上新的接口 192.168.1.34 外, 前两个路由表项没有其他变化。第 3 条路由是一条指向单一主机 10.4.7.25 的主机路由, 这通过把地址掩码所有位都设置为 1 便可以实现。注意, 不同于 10.0.0.0 子网的其他条目, 这条主机路由指向 Tigger 的接口 192.168.1.66。

为了观察添加新路由条目之后报文从路由器获取的路径, 可以在路由器 Pooh 上打开调试功能 **debug ip packet** (图 3-6)。这时从主机 192.168.1.15 有一个报文被发向主机 10.4.7.25。前两条调试捕获信息显示了从接口 E0 进入路由器的报文被路由到出站 S0 后再送往下一跳路由器 192.168.1.66 (Tigger), 按照要求, 在接口 S0 接收到的回复报文被路由到出站接口 E0 后再送往主机 192.168.1.15。

```
Pooh#debug ip packet
IP packet debugging is on
Pooh#
IP: s=192.168.1.15 (Ethernet0), d=10.4.7.25 (Serial0), g=192.168.1.66, forward
IP: s=10.4.7.25 (Serial0), d=192.168.1.15 (Ethernet0), g=192.168.1.15, forward
Pooh#
IP: s=192.168.1.15 (Ethernet0), d=10.4.7.100 (Serial1), g=192.168.1.34, forward
IP: s=10.4.7.100 (Serial0), d=192.168.1.15 (Ethernet0), g=192.168.1.15, forward
Pooh#
```

图 3-6 调试信息证实了在 Pooh 上的新路由表项工作正常

下一个调试信息显示一个报文从主机 192.168.1.15 发往主机 10.4.7.100。除了主机 10.4.7.25 外, 去往 10.0.0.0 子网上所有主机的报文都将沿新链路到达 Eeyore 的接口 192.168.1.34。第 3 个调试信息证实了这一点。然而, 第 4 个调试信息最初看上去有点让人惊讶。从 10.4.7.100 去往 192.168.1.15 的响应报文自 Tigger 到达 Pooh 的接口 S0。

还记得在其他路由器中的路由表项与原来例子中路由表项相比没有发生改变。这样的结果是期望的也好, 不是期望的也好, 但是它体现出了静态路由的两个特性。第一, 如果互联


```
ip route 10.1.30.0 255.255.255.0 10.1.20.2 50
```

Rabbit 路由表项配置如下:

```
ip route 10.4.0.0 255.255.0.0 10.1.10.1
ip route 10.4.0.0 255.255.0.0 10.1.20.1 50
ip route 10.1.5.0 255.255.255.0 10.1.10.1
ip route 10.1.5.0 255.255.255.0 10.1.20.1 50
ip route 192.168.0.0 255.255.0.0 10.1.10.1
ip route 192.168.0.0 255.255.0.0 10.1.20.1 50
```

在 Piglet 上有两条路由指向 Rabbit 的网络 10.1.30.0; 一条指定 Rabbit 接口 S0 的地址作为下一跳地址, 另一条指定 Rabbit 接口 S1 的地址作为下一跳地址。Rabbit 也有两条类似的配置。

注意: 在所有使用子网 10.1.20.0 的静态路由后面都跟了 50 这个数字。这个数字指定了管理距离, 管理距离是一种优先级度量。当存在两条路径到达相同的网络时, 路由器将会选择管理距离较低的路径。这种思想初听起来有点像度量, 但度量指明了路径的优先级, 而管理距离指明了发现路由方式的优先级。

例如, 指向下一跳地址的静态路由的管理距离为 1, 而指向出站接口的静态路由的管理距离为 0。如果有两条静态路由指向相同的目标网络, 一条指向下一跳地址, 一个指向出站接口, 那么后一条路由——管理距离值较低的路由——被选择。

将经过子网 10.1.20.0 的静态路由的管理距离提高到 50, 可以使经过子网 10.1.10.0 的静态路由成为首选路由。图 3-8 反复给出了 Rabbit 路由选择表的 3 次迭代。在第 1 个路由选择表中, 所有指向非直连网络的路由都使用下一跳地址 10.1.10.1。在每条路由表项中, 括号内的数字指定了管理距离为 1, 度量为 0 (因为静态路由没有度量)。

```
Rabbit#show ip route
      10.0.0.0 is variably subnetted, 5 subnets, 2 masks
C       10.1.10.0 255.255.255.0 is directly connected, Serial0
S       10.4.0.0 255.255.0.0 [1/0] via 10.1.10.1
S       10.1.5.0 255.255.255.0 [1/0] via 10.1.10.1
C       10.1.30.0 255.255.255.0 is directly connected, Ethernet0
C       10.1.20.0 255.255.255.0 is directly connected, Serial1
S      192.168.0.0 255.255.0.0 [1/0] via 10.1.10.1
Rabbit#

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
%LINK-3-UPDOWN: Interface Serial0, changed state to down

Rabbit#show ip route

      10.0.0.0 is variably subnetted, 4 subnets, 2 masks
S       10.4.0.0 255.255.0.0 [50/0] via 10.1.20.0
S       10.1.5.0 255.255.255.0 [50/0] via 10.1.20.1
C       10.1.30.0 255.255.255.0 is directly connected, Ethernet0
C       10.1.20.0 255.255.255.0 is directly connected, Serial1
S      192.168.0.0 255.255.0.0 [50/0] via 10.1.20.1
Rabbit#

%LINK-3-UPDOWN: Interface Serial0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
```

待续


```

Rabbit#show ip route
10.0.0.0 is variably subnetted, 5 subnets, 2 masks
C    10.1.10.0 255.255.255.0 is directly connected, Serial0
S    10.4.0.0 255.255.0.0 [1/0] via 10.1.10.1
S    10.1.5.0 255.255.255.0 [1/0] via 10.1.10.1
C    10.1.30.0 255.255.255.0 is directly connected, Ethernet0
C    10.1.20.0 255.255.255.0 is directly connected, Serial1
S 192.168.0.0 255.255.0.0 [1/0] via 10.1.10.1
Rabbit#

```

图 3-8 当主链路 10.1.10.0 失败时，备份链路 10.1.20.0 被启用。当主链路恢复时，再次启用主链路。

接着，陷阱消息（trap message）通知连接到接口 S0 的主链路状态变为“链路故障（down）”，表明链路发生故障。查看路由选择表第 2 次迭代发现所有非直连网络路由都指向下一跳地址 10.1.20.1。由于原来的首选路由不再可用，所以路由器切换到管理距离为 50 的备份链路。而且因为子网 10.1.10.0 发生故障，所以路由选择表中不再把它作为直连网络。

在路由选择表的第 3 个迭代之前，陷阱消息提示主链路状态恢复为“链路正常（up）”，路由选择表中再次显示子网 10.1.10.0，而且路由器也再次使用 10.1.10.1 作为下一跳地址。

第 11 章将结合多种动态路由选择协议讨论管理距离，动态路由选择协议的管理距离远远高于 1。因此对于相同的目标网络，缺省的静态路由总是优先于动态路由被发现。

3.2.5 案例研究：均分负载

上一节所用配置方法的弊病是在正常情况下备份链路不能被利用。备份链路的可用带宽资源被浪费了。均分负载（Load Sharing），又叫负平衡，允许路由器利用多路径的优点，在所有可用的路径上发送报文。

均分负载可以是等价或非等价的，这里的代价（Cost）是一个通用术语，它指的是与路由相关联的度量。

- 等价均分负载（Equal-Cost Load Sharing）——将流量均等地分布到多条度量相同的路径上。
- 非等价均分负载（Unequal-Cost Load Sharing）——将报文分布到不同度量的多条路径上。各条路径上分布的流量与路由代价成反比。也就是说，代价越低的路径分配的流量越多，代价越高的路径分配的流量越少。

一些路由选择协议可以支持等价和非等价负载均衡两种方式，而其他一些路由选择协议仅支持等价方式。静态路由没有度量，所以仅支持等价负载均衡。

在图 3-7 中存在并行链路，为了使用静态路由实现负载均衡，Piglet 相应的路由条目如下：

```

ip route 192.168.1.0 255.255.255.0 192.168.1.193
ip route 10.4.0.0 255.255.0.0 192.168.1.193
ip route 10.1.30.0 255.255.255.0 10.1.10.1
ip route 10.1.30.0 255.255.255.0 10.1.20.1

```


Rabbit 的路由条目如下:

```
ip route 10.4.0.0 255.255.0.0 10.1.10.1
ip route 10.4.0.0 255.255.0.0 10.1.20.1
ip route 10.1.5.0 255.255.255.0 10.1.10.1
ip route 10.1.5.0 255.255.255.0 10.1.20.1
ip route 192.168.0.0 255.255.0.0 10.1.10.1
ip route 192.168.0.0 255.255.0.0 10.1.20.1
```

除了两条链路的缺省管理距离都为 1 以外, 这些路由条目在浮动静态路由一节中都被用到了。如图 3-9 所示, Rabbit 的路由选择表对于每个目标网络都存在两条路由。

```
Rabbit#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
       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
Gateway of last resort is not set

 10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.10.0/24 is directly connected, Serial0
S       10.1.5.0/24 [1/0] via 10.1.10.1
                  [1/0] via 10.1.20.1
S       10.4.0.0/16 [1/0] via 10.1.10.1
                  [1/0] via 10.1.20.1
C       10.1.20.0/24 is directly connected, Serial1
S      192.168.0.0/16 [1/0] via 10.1.10.1
                  [1/0] via 10.1.20.1
Rabbit#
```

图 3-9 这个路由选择表指明了到达相同目标网络存在两条路径。路由器将会多条路径之间进行负载均衡

负载均衡有两种方式: 基于目标网络和基于报文。

1. 基于目标网络的负载均衡和快速交换

基于目标网络的负载均衡是根据目的地址分配负载。假设到一个网络存在两条路径, 那么去往该网络中第一个目标的报文从第一条路径通过, 去往网络中第二个目标的报文从第二条路径走, 去往此网络中第三个目标的所有报文还从第一条路径走, 依此类推。当 Cisco 路由器工作在缺省交换模式下时, 即快速交换 (Fast Switching) 模式, 路由器将使用这种负载均衡方式。

快速交换 (Fast Switching) 的工作方式如下: 当路由器为第一个去往特定目标的报文进行交换处理时, 路由器将执行路由选择表查询并选择出站接口。然后获取有关被选接口的数据链路信息 (例如从 ARP 表), 这些信息对报文成帧是必需的, 最后封装报文并发送。前面获取的路由和数据链路信息被输入到快速交换的高速缓冲内, 一旦去往相同目的地的后继报文进入路由器, 高速缓冲中的信息允许路由器不必查找路由选择表和 ARP 高速缓冲, 而是立即交换报文。

快速交换意味着所有去往指定目的地的报文都从相同的接口被发送出去, 因此交换时间和处理器的占用大大降低。当去往相同网络内不同主机的报文进入路由器时, 路由器可能会在另一条路径上发送报文到目的地。因此路由器能够做得最好的就是基于目标网络的

均衡负载。

2. 基于报文的均分负载和过程交换

基于报文的均分负载就是第一个去往一个目标网络的报文在链路 1 上发送，下一个去往相同目标网络的报文在另一条链路上发送，依此类推，这里假定是等价路径。如果路径代价不相同，那么高代价路径上每发送一个报文，低代价路径上就要发送 3 个报文，或者依据代价比率采取其他比例。当 Cisco 路由器处于过程交换模式时，将采用基于报文的均分负载方式。

过程交换（Process Switching）就是对于每个报文，路由器都要进行路由选择表查询和接口选择，然后再查询数据链路信息。因为每一次为报文确定路由的过程都是相互独立的，所以不会强制去往相同目标网络的所有报文使用相同的接口。为了在接口上打开过程交换功能，可以使用命令 **no ip route-cache**。

在图 3-10 中，主机 192.168.1.15 向主机 10.1.30.25 发送了 6 个 ping 报文。在 Piglet 上使用 **debug ip packet** 可以观察到 ICMP 的回应请求和回应应答报文。通过查看出站接口和转发地址可以发现 Piglet 和 Rabbit 都在交替使用接口 S0 和 S1。注意命令 **debug ip packet** 仅允许观察过程交换的报文，快速交换的报文将不能被显示出来。

```
Piglet#debug ip packet
IP packet debugging is on
Piglet#
IP: s=192.168.1.15 (Ethernet0), d=10.1.30.25 (Serial0), g=10.1.10.2, forward
IP: s=10.1.30.25 (Serial0), d=192.168.1.15 (Ethernet0), g=192.168.1.193, forward
IP: s=192.168.1.15 (Ethernet0), d=10.1.30.25 (Serial1), g=10.1.20.2, forward
IP: s=10.1.30.25 (Serial1), d=192.168.1.15 (Ethernet0), g=192.168.1.193, forward
IP: s=192.168.1.15 (Ethernet0), d=10.1.30.25 (Serial0), g=10.1.10.2, forward
IP: s=10.1.30.25 (Serial0), d=192.168.1.15 (Ethernet0), g=192.168.1.193, forward
IP: s=192.168.1.15 (Ethernet0), d=10.1.30.25 (Serial1), g=10.1.20.2, forward
IP: s=10.1.30.25 (Serial1), d=192.168.1.15 (Ethernet0), g=192.168.1.193, forward
IP: s=192.168.1.15 (Ethernet0), d=10.1.30.25 (Serial0), g=10.1.10.2, forward
IP: s=10.1.30.25 (Serial0), d=192.168.1.15 (Ethernet0), g=192.168.1.193, forward
IP: s=192.168.1.15 (Ethernet0), d=10.1.30.25 (Serial1), g=10.1.20.2, forward
IP: s=10.1.30.25 (Serial1), d=192.168.1.15 (Ethernet0), g=192.168.1.193, forward
Piglet#
```

图 3-10 路由器交替使用接口 S0 和 S1 发送去往相同目标网络的报文。注意，在两条链路

另一端的路由器也以同样的方式回复报文

正如许多设计选择一样，基于报文的均分负载也是要付出代价的。这种方式虽然使流量的分布比前一种方式更均匀，但是较低的交换时间和处理器占用的优点也随之丧失了。

3.2.6 案例研究：递归表查询

所有路由条目不必一定指向下一跳路由器。图 3-11 给出了图 3-7 中互联网络的简化版。在这个互联网络中，Pooh 配置如下：

```
ip route 10.1.30.0 255.255.255.0 10.1.10.2
ip route 10.1.10.0 255.255.255.0 192.168.1.194
ip route 192.168.1.192 255.255.255.224 192.168.1.66
```


如果 Pooh 需要向主机 10.1.30.25 发送报文, Pooh 将查找路由选择表并发现经过 10.1.10.2 可以到达这个子网。因为这个地址不在直连网络中, 所以 Pooh 必须再次查找路由选择表并发现去往 10.1.10.0 需要途径 192.168.1.194。由于这个子网也不是直连子网, 因而需要进行第 3 次路由选择表查找。这次 Pooh 发现途径 192.168.1.66 可以到达 192.168.1.192, 并且 192.168.1.66 在一个直连子网中。现在可以对报文进行转发。

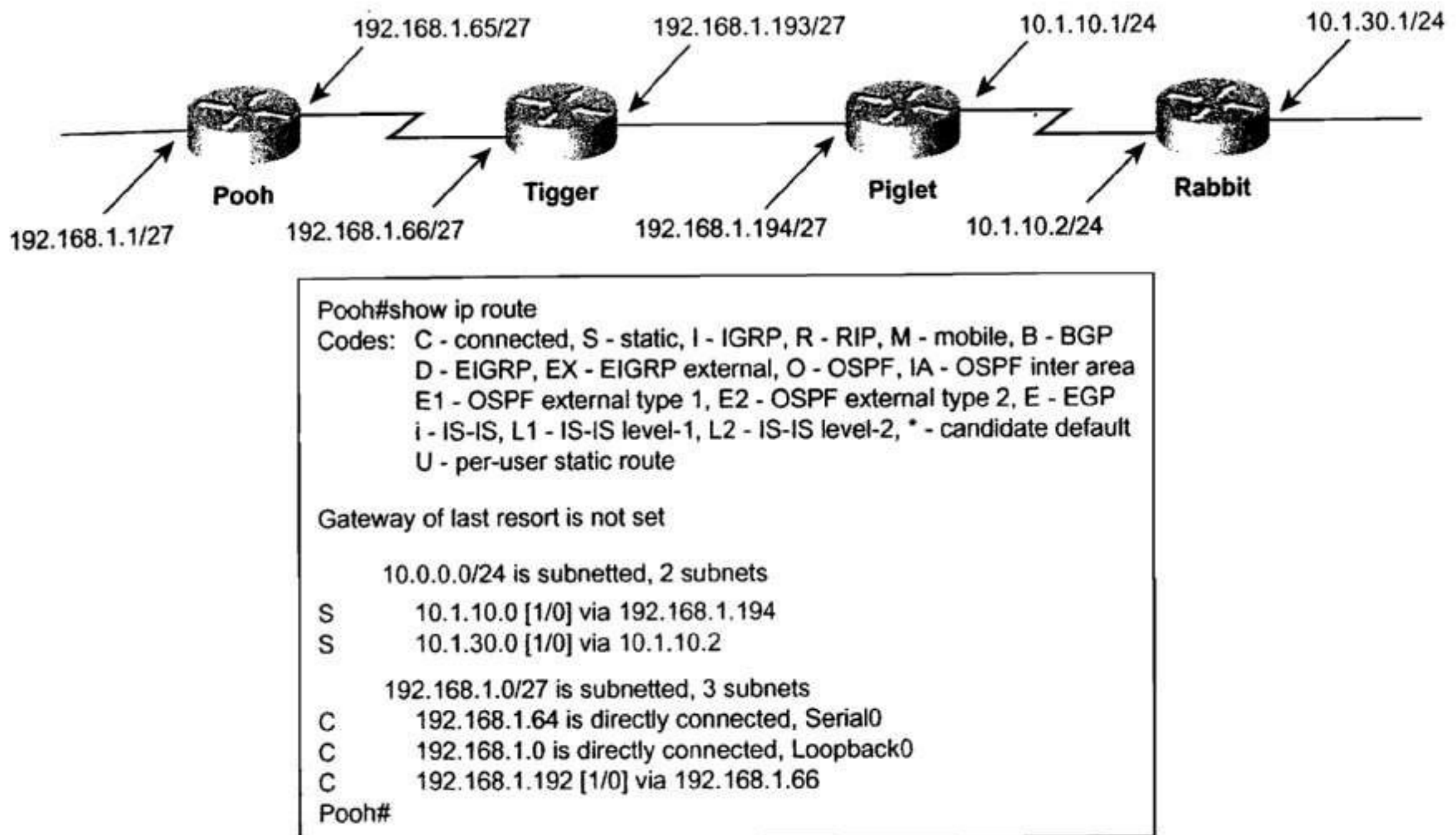


图 3-11 为了到达网络 10.1.30.0, Pooh 必须进行 3 次路由选择表查找

因为每次路由选择表查询都会花费处理器的时间, 所以在正常情况下强制路由器进行多次路由选择表查询是一种不好的设计决策。快速交换对递归查询进行了限制, 仅对去往每个目标网络的首报文进行递归查询, 从而有效地降低了这些不利的影响, 但是在使用这种设计方法之前仍然需要充分的理由。

图 3-12 给出了一个递归路由选择表查询的实例, 例子中的递归查询也许是有用的。在这里, Sanderz 途经 Heffalump 可以到达所有网络。然而, 网络管理员计划弃用 Heffalump, 改用 Woozle。Sanderz 中的前 12 条路由不再指向 Heffalump, 而是指向被连接到子网 10.87.14.0 上的合适路由器。最后一条路由指明经 Heffalump 可以到达子网 10.87.14.0。

使用下面的配置, 仅需要改动最后一条静态路由, 便可以使 Sanderz 的所有路由都重新指向 Woozel:

```
Sanderz(config)# ip route 10.87.14.0 255.255.255.0 10.23.5.95
Sanderz(config)# no ip route 10.87.14.0 255.255.255.0 10.23.5.20
```

如果以 10.23.5.20 作为所有静态路由条目的下一跳地址, 那么需要删除 13 条路由, 再重新输入 13 条新的路由。不过, 读者要仔细斟酌一下, 省去重新输入静态路由的麻烦与递归查询带给路由器的额外负担到底谁更重要。

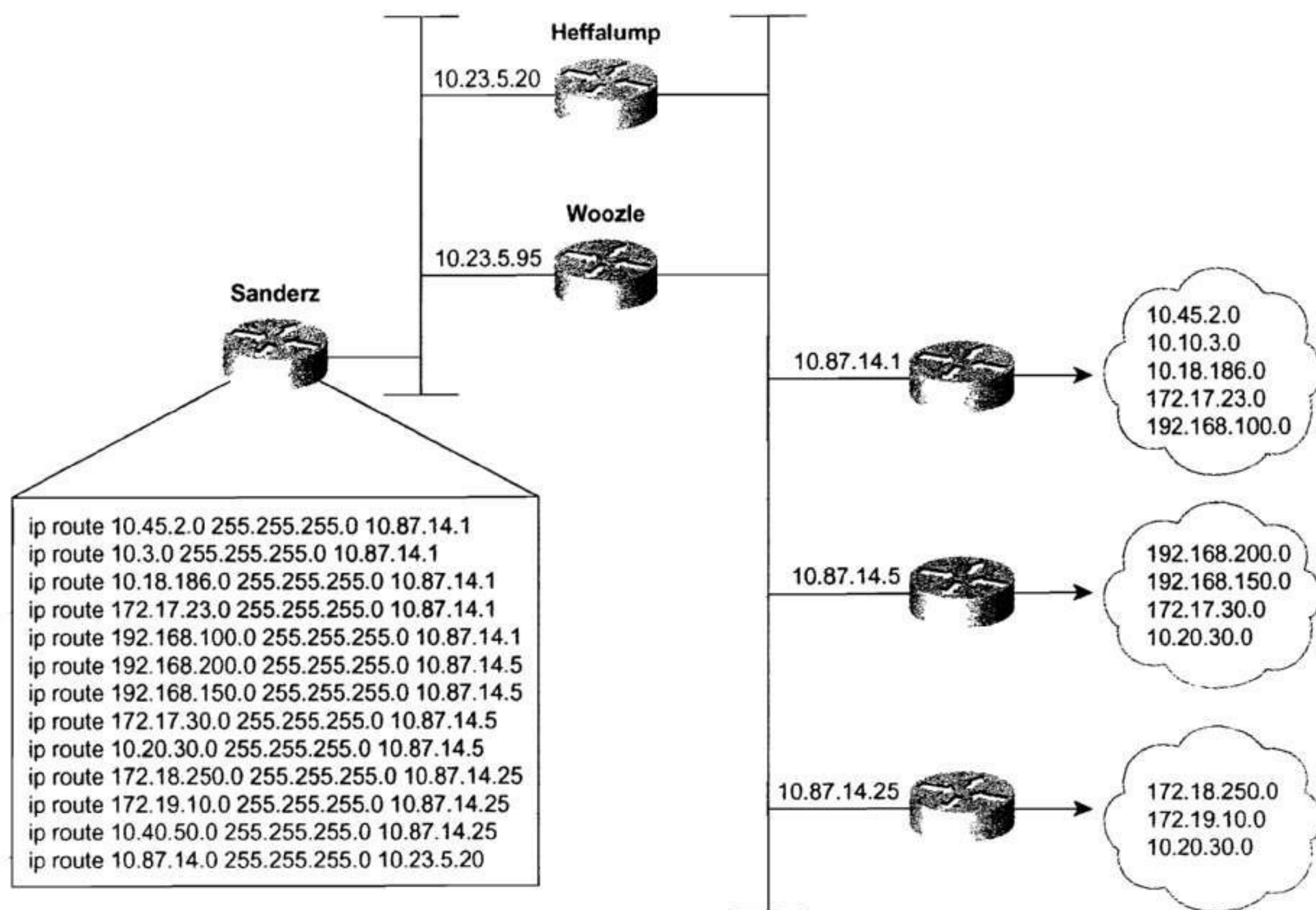


图 3-12 为 Sanderz 配置递归查询, 使网络管理员仅需要修改一条路由, 便可把离开 Sanderz 到 Heffaoump 的流量重定向到 Woozle

3.3 静态路由故障排除

了解过去 30 年各种美国政治丑闻的人们, 应该听说过一个国会调查员问过这样的问题, “他知道什么? 他什么时候知道的?” 这些问题对于互联网络调查员也同样适用。当排除路由故障时, 首要的一步就是检查路由选择表。路由器知道什么? 路由器知道怎样到达被提及的目标网络吗? 路由选择表中的信息准确吗? 为了顺利地排除互联网络的故障, 了解如何跟踪路由是十分必要的。

3.3.1 案例研究: 追踪故障路由

图 3-13 所示的互联网络在前面已经作过相应的配置, 其中包括每个路由器的静态路由。现在发现一个问题, 子网 192.168.1.0/27 连接在 Pooh 的以太网接口上, 该子网上的设备与子网 10.1.0.0/16 上的设备可以正常地通信。然而, 当 Pooh 向子网 10.1.0.0/16 发送 ping 报文时, 结果出现 ping 失败 (图 3-14)。这看上去很奇怪。如果 Pooh 可以成功地路由报文到达目标网络, 那么为什么源自 Pooh 的报文却传送失败呢?

为了解决这个问题, 需要跟踪一下 ping 所经过的路线。首先, 检查一下 Pooh 的路由选择表 (图 3-15)。(根据路由选择表) 目的地址 10.1.5.1 可以匹配到路由表项 10.0.0.0/8, 该子网经下一跳地址 192.168.1.34 可达, 192.168.1.34 是 Eeyore 的一个接口。

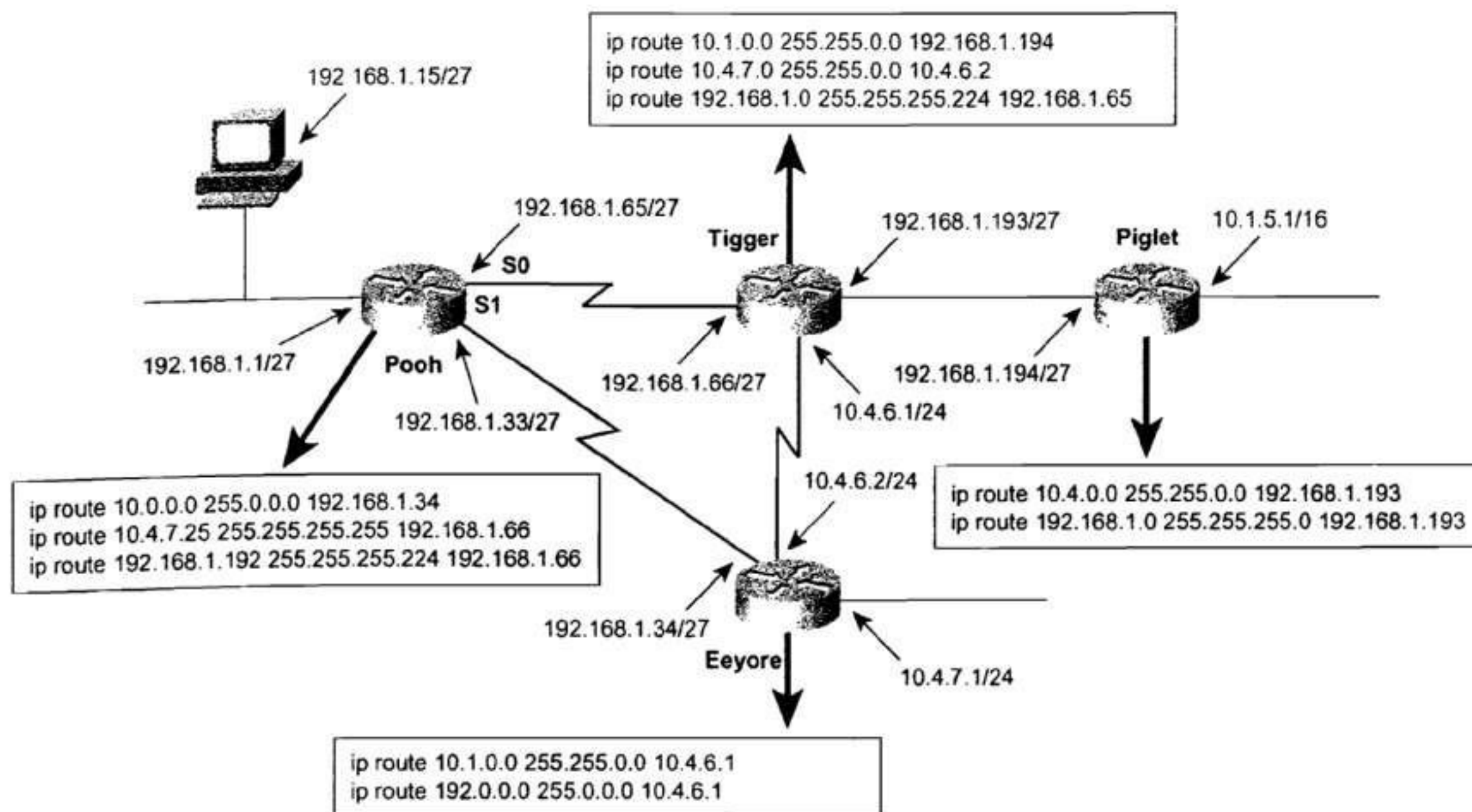


图 3-13 从子网 192.168.1.0/27 到子网 10.1.0.0/16 报文可以被正确地路由,但从 Pooh 却不能 ping 通子网 10.1.0.0/16 中的任何设备

```
C:\WINDOWS>ping 10.1.5.1
Pinging 10.1.5.1 with 32 bytes of data:
Reply from 10.1.5.1: bytes=32 time=22ms TTL=253
Reply from 10.1.5.1: bytes=32 time=22ms TTL=253
Reply from 10.1.5.1: bytes=32 time=22ms TTL=253
Reply from 10.1.5.1: bytes=32 time=22ms TTL=253
```

```
Pooh#ping 10.1.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 10.1.5.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Pooh#
```

图 3-14 子网 192.168.1.0/27 中的设备可以 ping 通 Piglet 的以太网接口,但是从 Pooh 进行 ping 操作却失败

```
Pooh#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
       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

Gateway of last resort is not set

  10.0.0.0 is variably subnetted, 2 subnets, 2 masks
S    10.0.0.0 255.0.0.0 [1/0] via 192.168.1.34
S    10.4.7.25 255.255.255.255 [1/0] via 192.168.1.66
  192.168.1.0 255.255.255.224 is subnetted, 4 subnets
C    192.168.1.64 is directly connected, Serial0
C    192.168.1.32 is directly connected, Serial1
C    192.168.1.0 is directly connected, Ethernet0
S    192.168.1.192 [1/0] via 192.168.1.66
Pooh#
```

图 3-15 去往目的地址 10.1.5.1 的报文在匹配到路由表项 10.0.0.0/8 之后被转发到下一跳路由器 192.168.1.34

然后，需要检查 Eeyore 的路由选择表（图 3-16）。目的地址 10.1.5.1 可以匹配到路由表项 10.1.0.0/16，该表项的下一跳地址为 10.4.6.1，它是 Tigger 的一个接口地址。

```
Eeyore#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
       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

Gateway of last resort is not set

  10.0.0.0 is variably subnetted, 3 subnets, 2 masks
C       10.4.6.0 255.255.255.0 is directly connected, Serial1
C       10.4.7.0 255.255.255.0 is directly connected, Ethernet0
S       10.1.0.0 255.255.0.0 [1/0] via 10.4.6.1
       192.168.1.0 255.255.255.224 is subnetted, 1 subnets
C       192.168.1.32 is directly connected, Serial0
S       192.0.0.0 255.0.0.0 [1/0] via 10.4.6.1
Eeyore#
```

图 3-16 10.1.5.1 在匹配到路由表项 10.1.0.0/16 之后被转发到 10.4.6.1

图 3-17 给出了 Tigger 的路由选择表。目标地址在匹配到路由表项 10.1.0.0/16 之后将被转发给 Piglet（192.168.1.194）。

```
Tigger#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
       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

Gateway of last resort is not set

  10.0.0.0 is variably subnetted, 3 subnets, 2 masks
C       10.4.6.0 255.255.255.0 is directly connected, Serial1
S       10.4.7.0 255.255.255.0 [1/0] via 10.4.6.2
S       10.1.0.0 255.255.0.0 [1/0] via 192.168.1.194
       192.168.1.0 255.255.255.224 is subnetted, 3 subnets
C       192.168.1.64 is directly connected, Serial0
S       192.168.1.0 [1/0] via 192.168.1.65
C       192.168.1.192 is directly connected, Ethernet0
Tigger#
```

图 3-17 10.1.5.1 在匹配到路由表项 10.1.0.0/16 后将被转发到 192.168.1.194

Piglet 的路由选择表（图 3-18）显示出目标网络 10.1.0.0 是一个直连网络。换言之，报文已经到达目的地了，因为目标地址 10.1.5.1 就是 Piglet 自己的接口地址。因为去往目标地址的路由经过验证是正确的，所以我们可以认为来自 Pooh 的 ICMP 回应报文可以到达目标网络。

再下一步是跟踪 ICMP 回应应答报文所经过的路径。为了跟踪这一路径，读者需要知道回应报文的源地址——这个地址将是回应应答报文的目的地。从路由器发出报文的源地址就是发送报文的接口地址。¹在本例中，最初由 Pooh 向 192.168.1.34 转发回应报文。图 3-13 显示出报文的源地址为 192.168.1.33。所以 Piglet 将要发送的回应应答报文的目标地址就是 192.168.1.33。

再次参考图 3-18 中 Piglet 的路由选择表，可以发现 192.168.1.33 将会匹配到路由表项

¹ 除非用扩展 ping 工具将源地址设置为其他地址。

192.168.1.0/24 并被转发到 192.168.1.193, 它是 Tigger 的另一个接口。重新检查图 3-17 中 Tigger 的路由选择表, 使我们想起还有一条关于 192.168.1.0 的路由。可是, 如果要根据这些信息准确地解释那里发生的实际情况, 那么需要十分小心。

```
Piglet#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
       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

Gateway of last resort is not set

 10.0.0.0 255.255.0.0 is subnetted, 2 subnets
C    10.1.0.0 is directly connected, Ethernet1
S    10.4.0.0 [1/0] via 192.168.1.193
    192.168.1.0 is variably subnetted, 2 subnets, 2 masks
S    192.168.1.0 255.255.255.0 [1/0] via 192.168.1.193
C    192.168.1.192 255.255.255.224 is directly connected, Ethernet0
Piglet#
```

图 3-18 目的网络 10.0.0.0 直接连接在 Piglet 上

比较一下 Tigger 路由选择表中有关 10.0.0.0 子网和 192.168.1.0 子网的路由表项。10.0.0.0 的标题表明其子网大小各不相同; 换句话说, 指向子网 10.4.7.0 Tigger 的静态路由使用了 24 位掩码, 指向子网 10.1.0.0 的静态路由使用了 16 位掩码。该路由选择表为每个子网都记录了正确的掩码。

192.168.1.0 的标题则不同, 它表明 Tigger 知道 192.168.1.0 有 3 个子网, 且掩码都为 255.255.255.224。用这个掩码可以确定目标地址 192.168.1.33 所属的目标网络为 192.168.1.32/27。但是路由选择表中只有关于 192.168.1.64/27、192.168.1.0/27 和 192.168.1.192/27 的路由表项, 而没有关于 192.168.1.32/27 的路由表项。因此路由器不知道该如何到达这个子网。

那么问题就很清楚了, ICMP 的回应应答报文是在 Tigger 处被丢弃的。一个解决办法是另外创建一条指向网络 192.168.1.32 的静态路由, 掩码为 255.255.255.224, 指向下一跳地址为 192.168.1.65 或 10.4.6.2。还有一个解决办法是将关于 192.168.1.0 的路由表项的掩码由 255.255.255.224 改为 255.255.255.0。

此案例的精髓就是当你跟踪路由时, 你必须考虑完整的通信过程。不仅要验证去往目标网络的路由是正确的, 还要验证返回的路径也是正确的。

3.3.2 案例研究: 协议冲突

如图 3-19 所示, 两台路由器被两个以太网连接起来, 其中一个以太网包括一个网桥。这个网桥同时还负责处理其他几条没有画出的链路的流量, 所以有时网桥会变得十分拥挤。主机 Milne 是一台承担重要任务的服务器, 网络管理员担心网桥会延误 Milne 的流量, 所以在 Roo 上添加了一条指向 Milne 的主机路由, 该路由指引报文使用图上方的以太网避开该网桥。

这个解决方案看上去是合理的, 但是实际并非如此。在添加上面那条静态路由后, 报文经 Roo 不但不能被路由到服务器, 而且报文经 Kanga 也不能被路由到服务器, 尽管没有对

Kanga 作过改动。

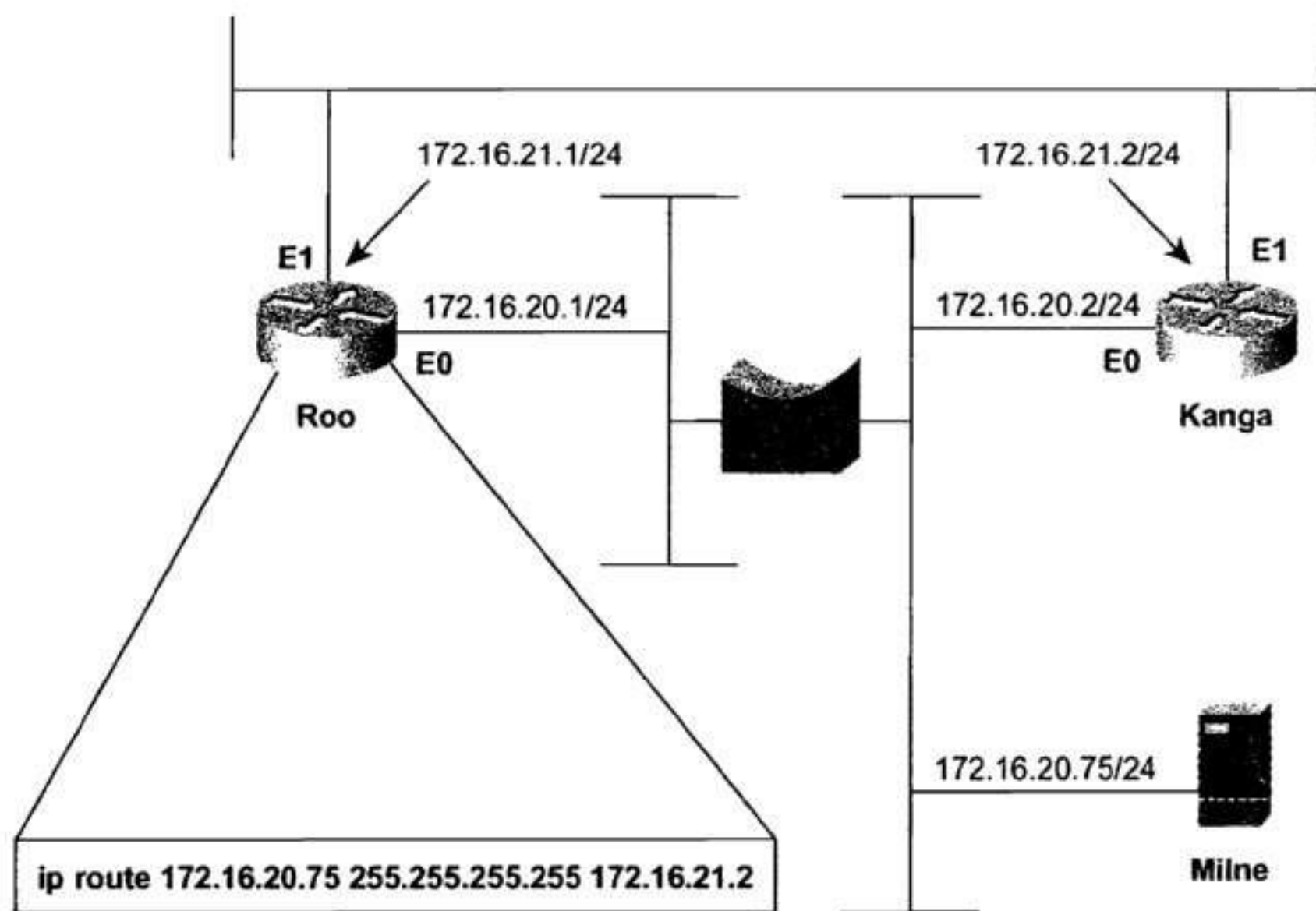


图 3-19 主机路由指引从 Roo 到 Milne 的报文经过上面的以太网，这样可以避开偶尔发生拥塞的网桥

照常第一步先检查路由选择表。Roo 的路由选择表(图 3-20)指明目标地址为 172.16.20.75 的报文实际上是被转发到 Kanga 的接口 E1 上，这正是我们所期望的。由于目标网络直接连接到 Kanga 上，所以不再需要进一步的路由。经过快速检查确定在 Kanga 和 Milne 上的两个以太网接口都工作正常。

```
Roo#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
       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
Gateway of last resort is not set

 172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.20.0/24 is directly connected, Ethernet0
C       172.16.21.0/24 is directly connected, Ethernet1
S       172.16.20.75/32 [1/0] via 172.16.21.2
Roo#
```

图 3-20 在 Roo 的路由选择表中显示有一条经 Kanga 接口 E1 指向 Milne 的静态主机路由

在图 3-21 中，从 Roo 向 Milne 执行跟踪命令，发现以下症状。Kanga 应该发向 Milne 的报文被发向 Roo 的接口 E0。Roo 又将报文发给 Kanga 接口 E1，接着 Kanga 再次将报文发回给 Roo。这看上去像是发生了路由环路，但是为什么呢？

此故障的可疑之处在于 Kanga 本不应该对报文进行路由选择，但事实恰恰相反。Kanga 应能够识别出报文的目标地址属于它的直连网络 172.16.20.0，然后使用数据链路向主机传送报文。因此疑点落在数据链路上。路由器是否具有经过某条逻辑路径到达目标网络的正确信息，这一点我们可通过查看路由选择表来获知，同样，我们应该检查 ARP 高速缓冲区，来确定路由器是否具有经过某条物理路径到达某个主机的正确信息。


```

Roo#trace 172.16.20.75
Type escape sequence to abort.
Tracing the route to 172.16.20.75

 0  172.16.21.2    0    msec    0    msec    0    msec
 1  172.16.20.1    4    msec    0    msec    0    msec
 2  172.16.21.2    4    msec    0    msec    0    msec
 3  172.16.20.1    0    msec    0    msec    4    msec
 4  172.16.21.2    0    msec    0    msec    4    msec
 5  172.16.20.1    0    msec    0    msec    4    msec
 6  172.16.21.2    0    msec    0    msec    4    msec
 7  172.16.20.1    0    msec    0    msec    4    msec
 8  172.16.21.2    4    msec    0    msec    4    msec
 9  172.16.20.1    4    msec    0    msec    4    msec
10  172.16.21.2    4    msec
11  172.16.21.2    4    msec
Roo#

```

图 3-21 从 Roo 到 Milne 进行跟踪发现 Kanga 将本应发向正确目的地的报文回给了 Roo

图 3-22 给出了 Kanga 的 ARP 高速缓冲。在 Kanga 的高速缓冲中, Milne 的 IP 地址与 MAC 标识符 00e0.1e58.dc39 相对应。但是当检查 Milne 的接口时, 发现 Milne 的 MAC 标识符为 0002.6779.0f4c, 因此断定 Kanga 一定获取了不正确的信息。

```

Kanga#show arp
Protocol Address      Age (min)  Hardware Addr  Type  Interface
Internet 172.16.21.1        2          00e0.1e58.dc3c ARPA  Ethernet1
Internet 172.16.20.2        -          00e0.1e58.dcb1 ARPA  Ethernet0
Internet 172.16.21.2        -          00e0.1e58.dcb4 ARPA  Ethernet1
Internet 172.16.20.75      2          00e0.1e58.dc39 ARPA  Ethernet0
Kanga#

```

图 3-22 在 Kanga 的 ARP 高速缓冲中, 有一个关于 Milne 的表项, 其中数据链路标识是错误的

再次查看 Kanga 的 ARP 高速缓冲, 令人感到疑惑的是与 Milne 相对应的 MAC 标识符类似于 Kanga 自己接口 (Cisco 产品) 的 MAC 标识符 (路由器接口的 MAC 地址没有相关的年代记录)。因为 Milne 不是 Cisco 产品, 所以 MAC 标识符的前 3 个 8bit 字节应该与 Kanga 接口的 MAC 标识符不同。互联网络上不是 Cisco 的产品惟有 Roo, 因此检查一下 Roo 的 ARP 高速缓冲 (图 3-23)。经查 Roo 接口 E0 的 MAC 标识符即为 00e0.1e58.dc39。

```

Roo#show arp
Protocol Address      Age (min)  Hardware Addr  Type  Interface
Internet 172.16.21.1        -          00e0.1e58.dc3c ARPA  Ethernet0
Internet 172.16.20.1        -          00e0.1e58.dc39 ARPA  Ethernet0
Internet 172.16.20.2        7          00e0.1e58.dcb1 ARPA  Ethernet0
Internet 172.16.21.2        7          00e0.1e58.dcb4 ARPA  Ethernet1
Roo#

```

图 3-23 Roo 的 ARP 高速缓冲显示出 Kanga 获取的 Milne 的 MAC 标识符实际上是 Roo 的接口 E0 的

所以 Kanga 错误地认为 Roo 的接口 E0 就是 Milne 的接口。Kanga 使用 00e0.1e58.dc39 作为发向 Milne 数据帧的目的标识符, Roo 接收到该帧, 在读取封装报文的目标地址之后,

又将报文路由回 Kanga。

但 Kanga 是如何得到这个错误信息的呢？答案是代理 ARP。当 Kanga 首次收到发往 Milne 的报文时，它将发送 ARP 请求，该请求将询问 Milne 的数据链路标识符。Milne 发回了响应，但是 Roo 也在接口 E0 收到了此 ARP 请求。由于在 Roo 上也有一条通向 Milne 的路径，但是这条路径所在的网络不是 Pooh 收到 ARP 请求的网络，所以 Pooh 发送了一个代理 ARP 应答。Kanga 收到 Milne 的 ARP 应答后将相关信息输入到 ARP 高速缓冲内。由于网桥的时延造成来自 Roo 代理 ARP 的应答随后到达 Kanga。这时 Kanga 用新信息覆盖了 ARP 缓冲内的原始信息。

解决办法有两个。一个是使用以下命令关闭 Roo E0 接口上的代理 ARP 功能：

```
Roo(config)#interface e0
Roo(config-if)#no ip proxy-arp
```

第二个办法是在 Kanga 上为 Milne 配置静态 ARP 表项：

```
Kanga(config)#arp 172.16.20.75 0002.6779.0f4c arpa
```

这个表项不会被任何 ARP 应答所覆盖。图 3-24 显示出正在输入静态 ARP 表项以及 Kanga 上 ARP 高速缓冲的相应结果。

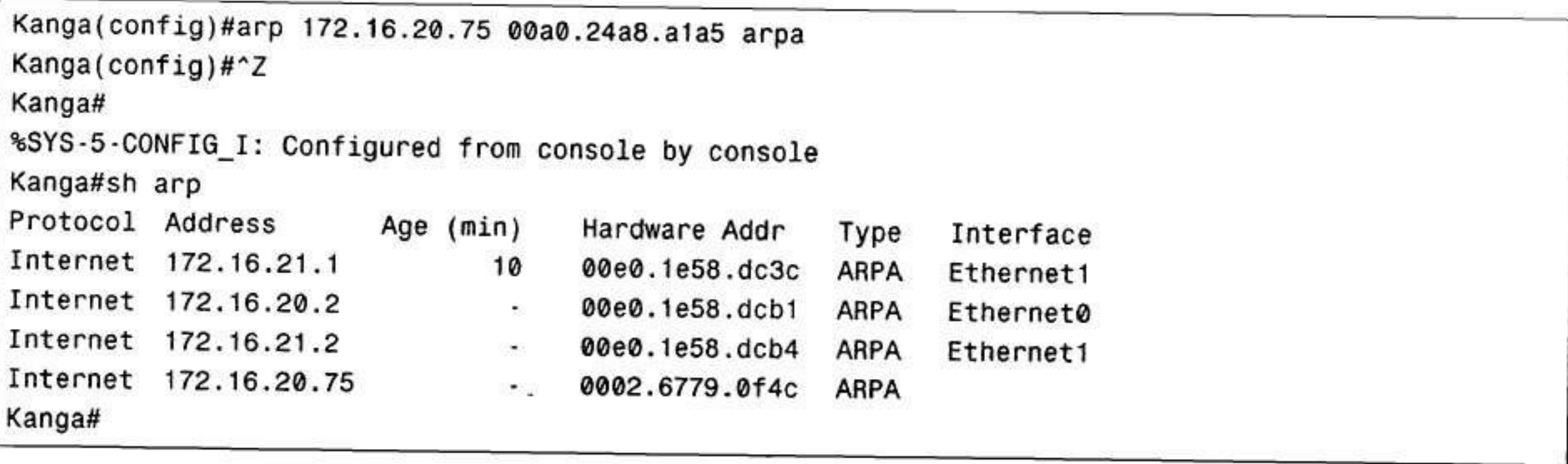


图 3-24 静态 ARP 纠正了代理 ARP 所造成的错误

两种方法哪一种最好取决于网络环境。使用静态 ARP 意味着如果 Milne 的网络接口被更换，那么需要修改相应的 ARP 表项反映新的 MAC 标识符。另一方面，如果没有主机使用代理 ARP 功能，关闭此功能也是一个很好的办法。

3.4 展 望

对于精确地控制互联网络的路由行为来说，静态路由不失为一个强有力的工具。然而，如果经常发生网络拓扑变化，那么手动配置方式导致静态路由的管理工作根本无法进行下去。动态路由选择协议能够使互联网络迅速并自动地响应网络拓扑的变化。在研究特定 IP 路由选择协议的细节之前，我们首先需要研究一些围绕着动态协议的常见问题。下一章将介绍动态路由选择协议。

3.5 总结表：第 3 章命令回顾

命 令	描 述
<code>arp ip-address hardware-address</code>	把 IP 类型 (别名) 地址静态地映射到硬件地址
<code>debug ip packet</code>	显示有关接收、生成、转发 IP 报文的信息。而关于快速交换的报文信息将不被显示
<code>ip proxy-arp</code>	打开代理 ARP 功能
<code>ip route prefix mask{address interface}[[distance]][permanent]</code>	向路由选择表添加静态路由
<code>ip route-cache</code>	在接口上配置快速交换高速缓冲

3.6 复 习 题

1. 路由选择表中需要保存哪些信息?
2. 当路由选择表指明对一个地址进行了变长子网划分时, 这意味着什么?
3. 什么是非连续子网?
4. 在 Cisco 路由器上使用什么命令检查路由选择表?
5. 在路由选择表中与非直连路由相关的括号内的两个数字表示什么?
6. 当使用出站接口代替静态路由中的下一跳地址时, 路由选择表会有什么不同?
7. 什么是汇总路由? 在静态路由选择的上下文中, 汇总路由怎样起作用?
8. 什么是管理距离?
9. 什么是浮动静态路由?
10. 等价均分负载与非等价均分负载之间有什么不同?
11. 接口上的交换模式怎样影响均分负载?
12. 什么是递归表查询?

3.7 配置练习

1. 如图 3-25 所示的互连网络, 为每台路由器配置静态路由。要求每个子网都有单独的表项。
2. 使用最少的路由表项重新配置练习 1 中的静态路由。¹ (提示: RTA 仅有两条静态路由)
3. 如图 3-26 的互连网络, 为每台路由器配置静态路由。假设所有链路的介质相同。为保证最大利用率和线路冗余, 请使用负载均衡和浮动路由技术。

¹ 如果在实验室做此练习, 请确保在所有 6 台路由器上都配置命令 `ip classless`。

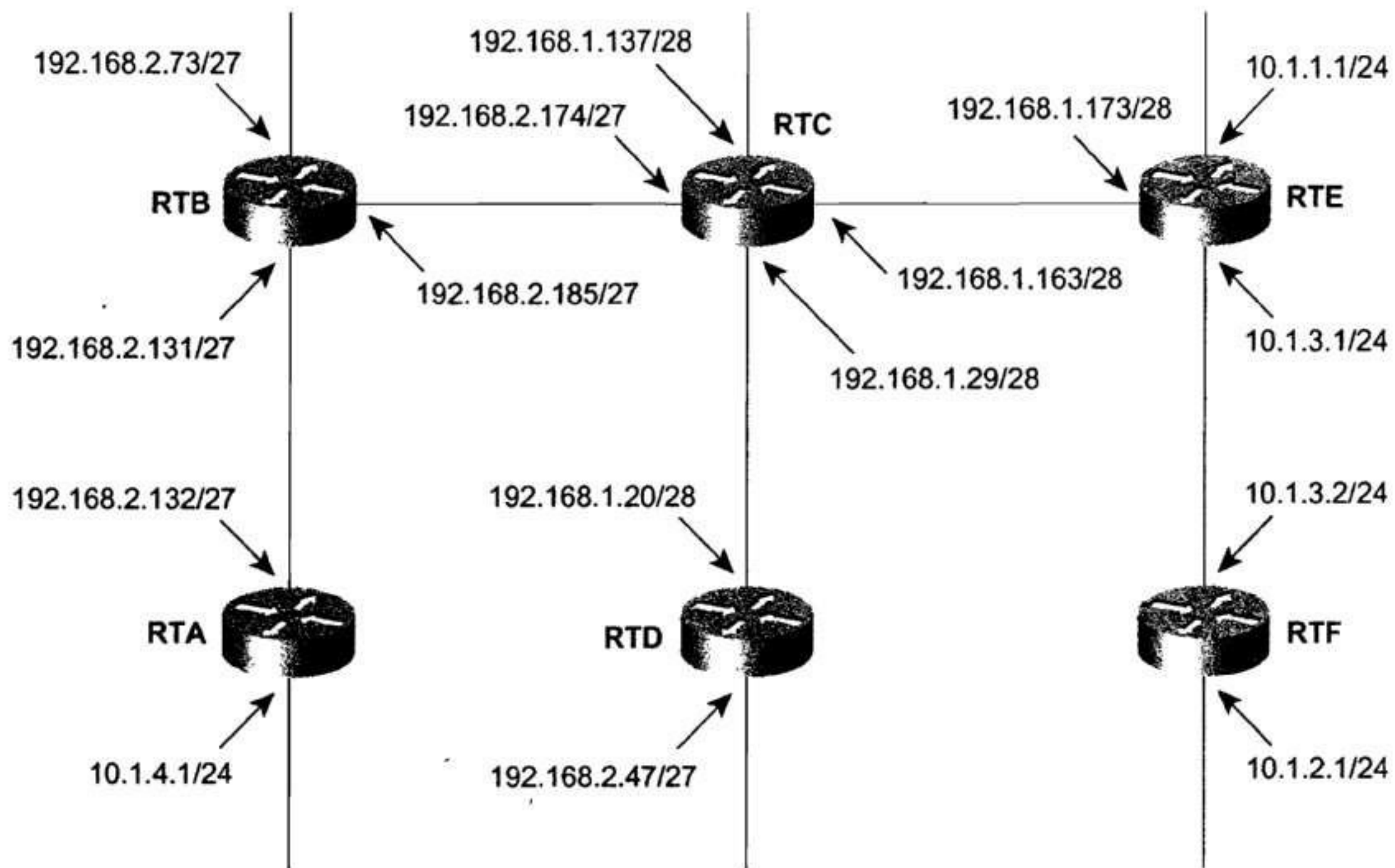


图 3-25 配置练习 1 和 2 中用到的互连网络

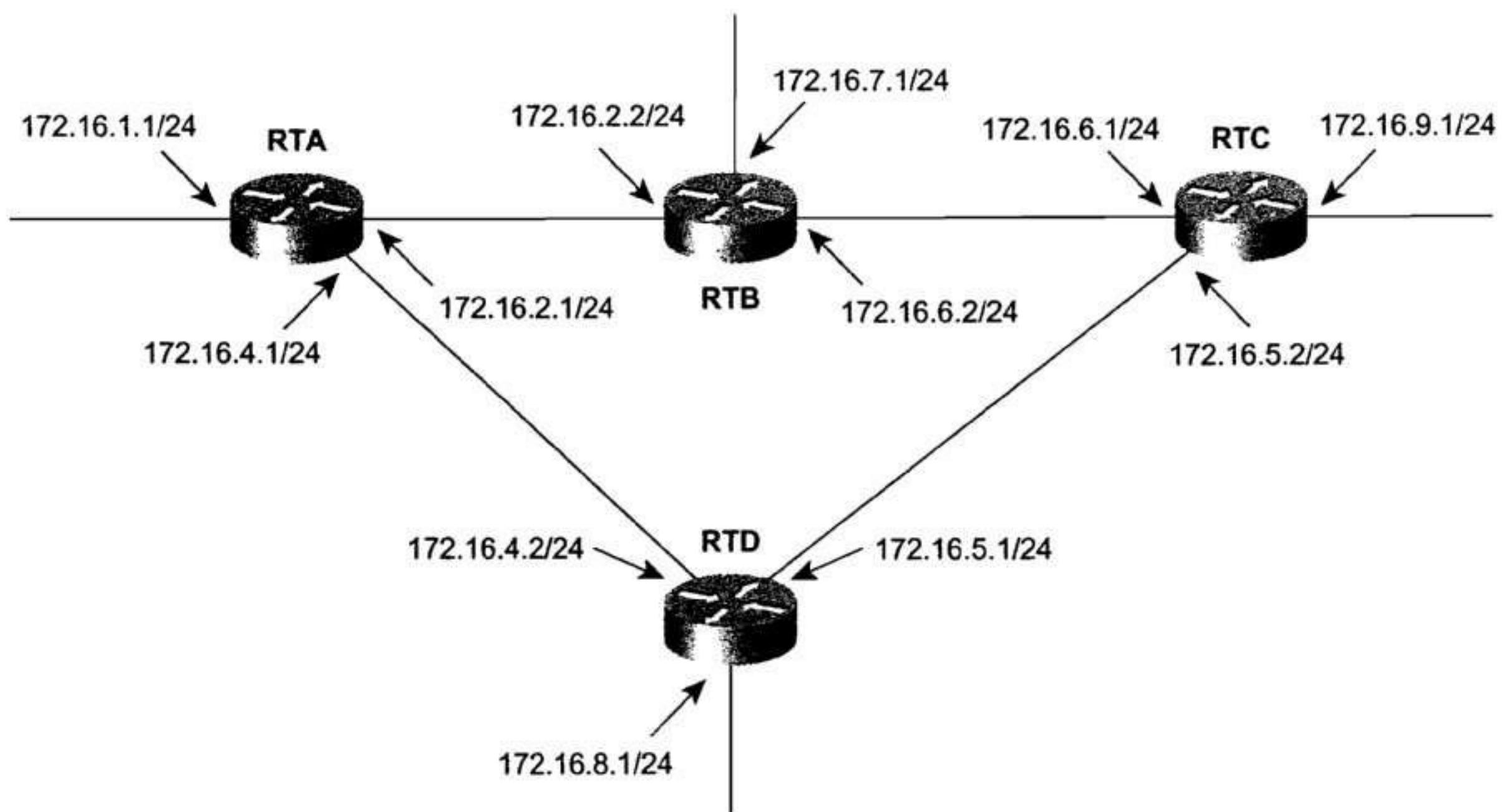


图 3-26 配置练习 3 用到的互连网络

3.8 故障排除练习

1. 在图 3-3 所示的互连网络和相关配置中，把 Piglet 的路由配置由

```
Piglet(config)# ip route 192.168.1.0 255.255.255.0 192.168.1.193
Piglet(config)# ip route 10.4.0.0 255.255.0.0 192.168.1.193
```


修改为

```
Piglet(config)# ip route 192.168.1.0 255.255.255.224 192.168.1.193
Piglet(config)# ip route 10.0.0.0 255.255.0.0 192.168.1.193
```

会发生什么情况?

2. 在图 3-27 中, 路由器的静态路由配置如下:

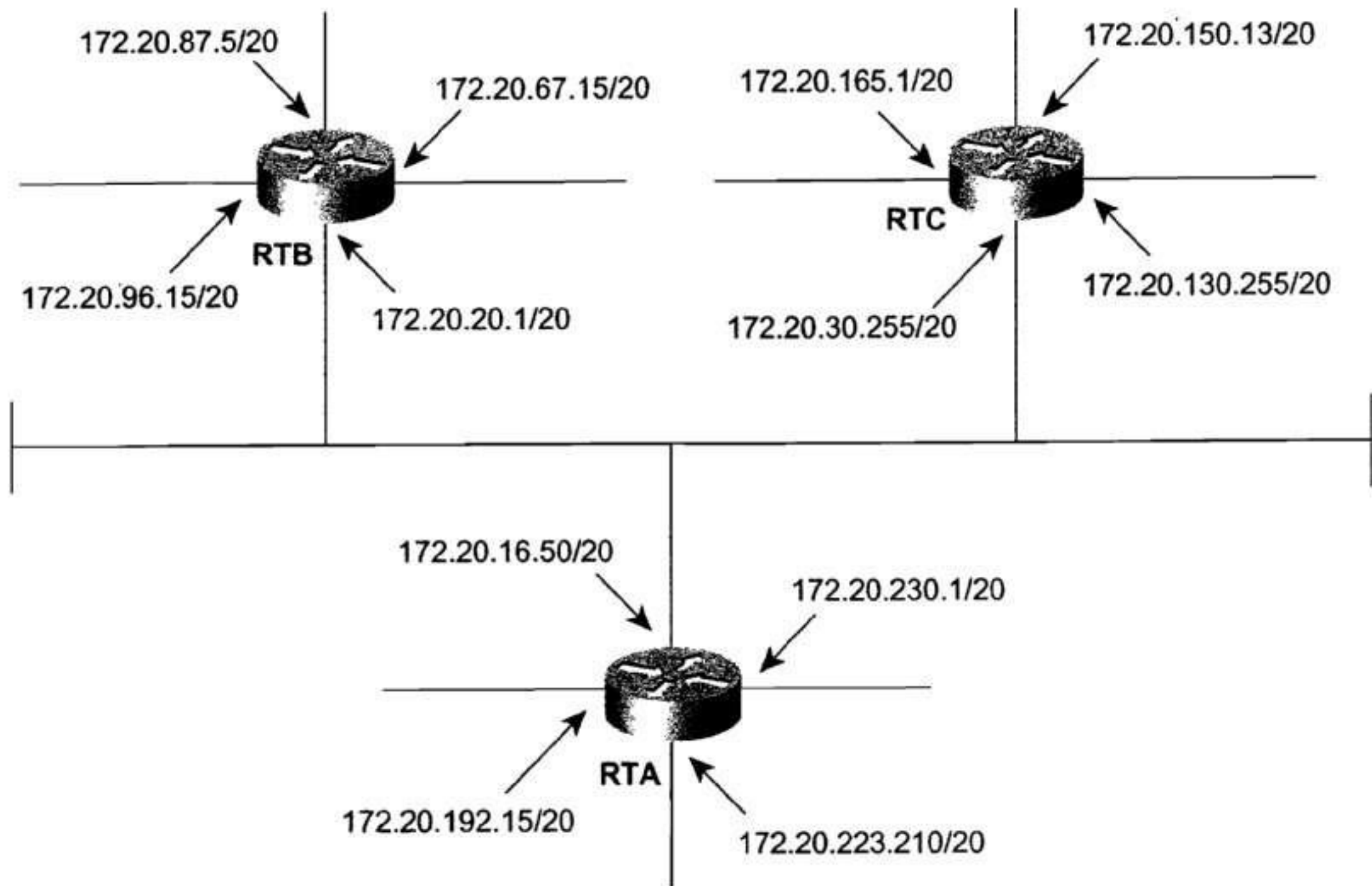


图 3-27 故障排除练习 2 用到的互联网络

路由器 RTA:

```
ip route 172.20.96.0 255.255.240.0 172.20.20.1
ip route 172.20.82.0 255.255.240.0 172.20.20.1
ip route 172.20.64.0 255.255.240.0 172.20.20.1
ip route 172.20.160.0 255.255.240.0 172.20.30.255
ip route 172.20.144.0 255.255.240.0 172.20.30.255
ip route 172.20.128.0 255.255.240.0 172.20.30.255
```

路由器 RTB:

```
ip route 172.20.192.0 255.255.240.0 172.20.16.50
ip route 172.20.224.0 255.255.240.0 172.20.16.50
ip route 172.20.128.0 255.255.240.0 172.20.16.50
ip route 172.20.160.0 255.255.240.0 172.20.30.255
ip route 172.20.144.0 255.255.240.0 172.20.30.255
ip route 172.20.128.0 255.255.240.0 172.20.30.255
```

路由器 RTC:

```
ip route 172.20.192.0 255.255.240.0 172.20.16.50
ip route 172.20.208.0 255.255.255.0 172.20.16.50
ip route 172.20.224.0 255.255.240.0 172.20.16.50
ip route 172.20.96.0 255.255.240.0 172.20.20.1
```



```
ip route 172.20.82.0 255.255.240.0 172.20.20.1
ip route 172.20.64.0 255.255.240.0 172.20.20.1
```

用户抱怨互联网络中存在一些连通性问题。请在静态路由配置中找出错误。

3. 图 3-28 给出了另一个互联网络，同样有用户抱怨存在连通性问题。图 3-29 到图 3-32 分别给出了 4 个路由器的路由选择表。请找出静态路由配置的错误。

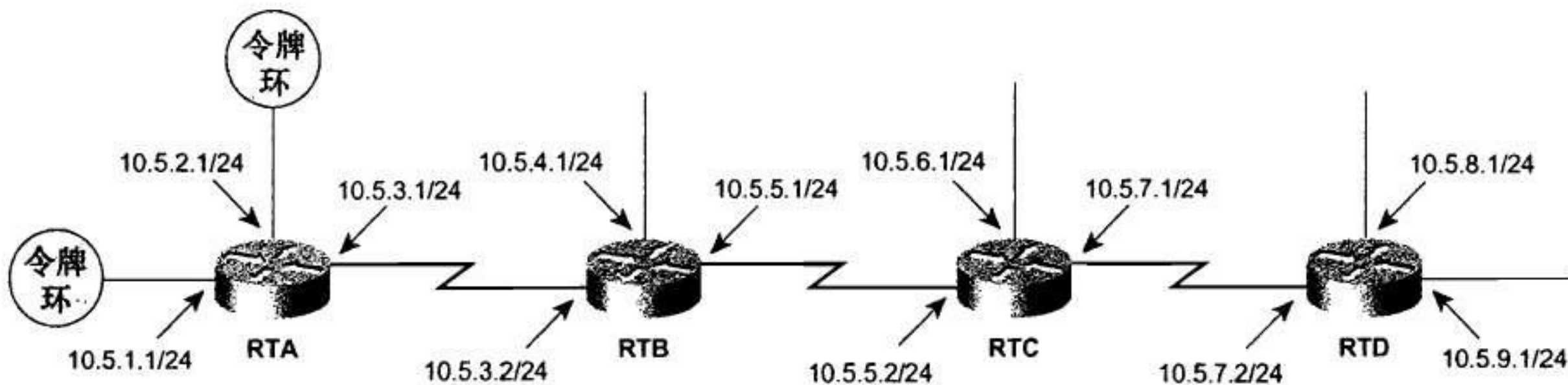


图 3-28 互联网络故障排除练习 3

```
RTA#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
       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

Gateway of last resort is not set

 10.0.0.0/8 is subnetted, 9 subnets
S    10.5.9.0 [1/0] via 10.5.3.2
S    10.5.8.0 [1/0] via 10.5.3.2
S    10.5.7.0 [1/0] via 10.5.3.2
S    10.5.6.0 [1/0] via 10.5.3.2
S    10.5.5.0 [1/0] via 10.5.3.2
S    10.5.4.0 [1/0] via 10.5.3.2
C    10.5.3.0 is directly connected, Serial0
C    10.5.2.0 is directly connected, TokenRing1
C    10.5.1.0 is directly connected, TokenRing0
RTA#
```

图 3-29 图 3-28 中 RTA 的路由选择表

```
RTB#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
       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

Gateway of last resort is not set

 10.0.0.0/8 is subnetted, 9 subnets
S    10.5.9.0 [1/0] via 10.5.5.2
S    10.5.8.0 [1/0] via 10.5.5.2
S    10.5.7.0 [1/0] via 10.5.5.2
S    10.5.6.0 [1/0] via 10.5.5.2
C    10.5.5.0 is directly connected, Serial1
C    10.5.4.0 is directly connected, TokenRing0
C    10.5.3.0 is directly connected, Serial0
S    10.5.2.0 [1/0] via 10.5.3.1
S    10.5.1.0 [1/0] via 10.5.3.1
RTB#
```

图 3-30 图 3-28 中 RTB 的路由选择表


```

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

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 8 subnets
S       10.5.9.0 [1/0] via 10.5.7.2
S       10.5.8.0 [1/0] via 10.5.5.1
C       10.5.7.0 is directly connected, Serial1
C       10.5.6.0 is directly connected, Ethernet0
S       10.1.1.0 [1/0] via 10.5.5.1
C       10.5.5.0 is directly connected, Serial0
S       10.5.3.0 [1/0] via 10.5.5.1
S       10.5.2.0 [1/0] via 10.5.5.1
RTC#

```

图 3-31 图 3-28 中 RTC 的路由选择表

```

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

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 9 subnets
C       10.5.9.0 is directly connected, Ethernet1
C       10.5.8.0 is directly connected, Ethernet0
C       10.5.7.0 is directly connected, Serial0
S       10.5.6.0 [1/0] via 10.5.7.1
S       10.5.5.0 [1/0] via 10.5.7.1
S       10.4.5.0 [1/0] via 10.5.7.1
S       10.5.3.0 [1/0] via 10.5.7.1
S       10.5.2.0 [1/0] via 10.5.7.1
S       10.5.1.0 [1/0] via 10.5.7.1
RTD#

```

图 3-32 图 3-28 中 RTD 的路由选择表

第 4 章

动态路由选择协议

本章包括以下主题：

- 路由选择协议基础
- 距离矢量路由选择协议
- 链路状态路由选择协议
- 内部和外部网关协议
- 静态或动态路由？

上一章说明了路由器为了正确地交换报文到达各自的目的地所需要知道的信息，以及怎样手工向路由选择表输入这些信息。本章将讨论路由器如何发现这些信息，并且借助动态路由选择协议与其他路由器共享这些信息。为了共享可达性信息和网络状态，路由选择协议被作为路由器之间进行相互交流的语言。

动态路由选择协议不仅执行路径决策和路由选择表更新功能，而且还要在最优路由不可用时决策下一条最优路由。动态路由选择相比静态路由选择而言最大的优势在于动态路由选择能够缓解拓扑变动带来的影响。

显然，为了使通信发生，通信双方必须使用相同的语言。有 8 种主要的 IP 路由选择协议可供选择，如果一台路由器使用 RIP 与另一台使用 OSPF 的路由器进行对话，那么它们将无法实现信息共享，因为它们没有使用相同的语言。

后面的章节将会分析所有目前在用的 IP 路由选择协议，甚至涉及如何使路由器“能说两种语言”，但是首先是研究所有路由选择协议共有的一些特性和问题——IP 或其他方面。

4.1 路由选择协议基础

所有路由选择协议都是围绕着一个算法构建的。一般地, 一个算法是一个逐步解决问题的过程。一个路由算法至少应指明以下内容:

- 向其他路由器传送网络可达信息的过程
- 从其他路由器接收可达信息的过程
- 基于现有可达信息决策最优路由的过程以及在路由选择表中记录这些信息的过程
- 响应、修正和通告互联网络中拓扑变化的过程

对所有路由选择协议来说, 共有的几个问题是路径决策、度量、收敛和负载均衡。

4.1.1 路径决策

在互联网内的所有网络都必须连接到一台路由器上, 如果路由器有一个接口连接到一个网络上, 那么这个接口必须具有一个属于该网络的地址。这个地址就是可达信息的起始点。

图 4-1 给出了一个包含 3 个路由器的互联网络。路由器 A 知道网络 192.168.1.0、192.168.2.0 和 192.168.3.0 的存在, 因为路由器有接口连接到这些网络上, 并且配置了相应的地址和掩码。同样, 路由器 B 知道网络 192.168.3.0、192.168.4.0、192.168.5.0 和 192.168.6.0 的存在, 由于每个接口都实现了所连接网络的数据链路和物理协议, 因此路由器也知道网络的状态 (工作正常 “up” 或发生故障 “down”)。

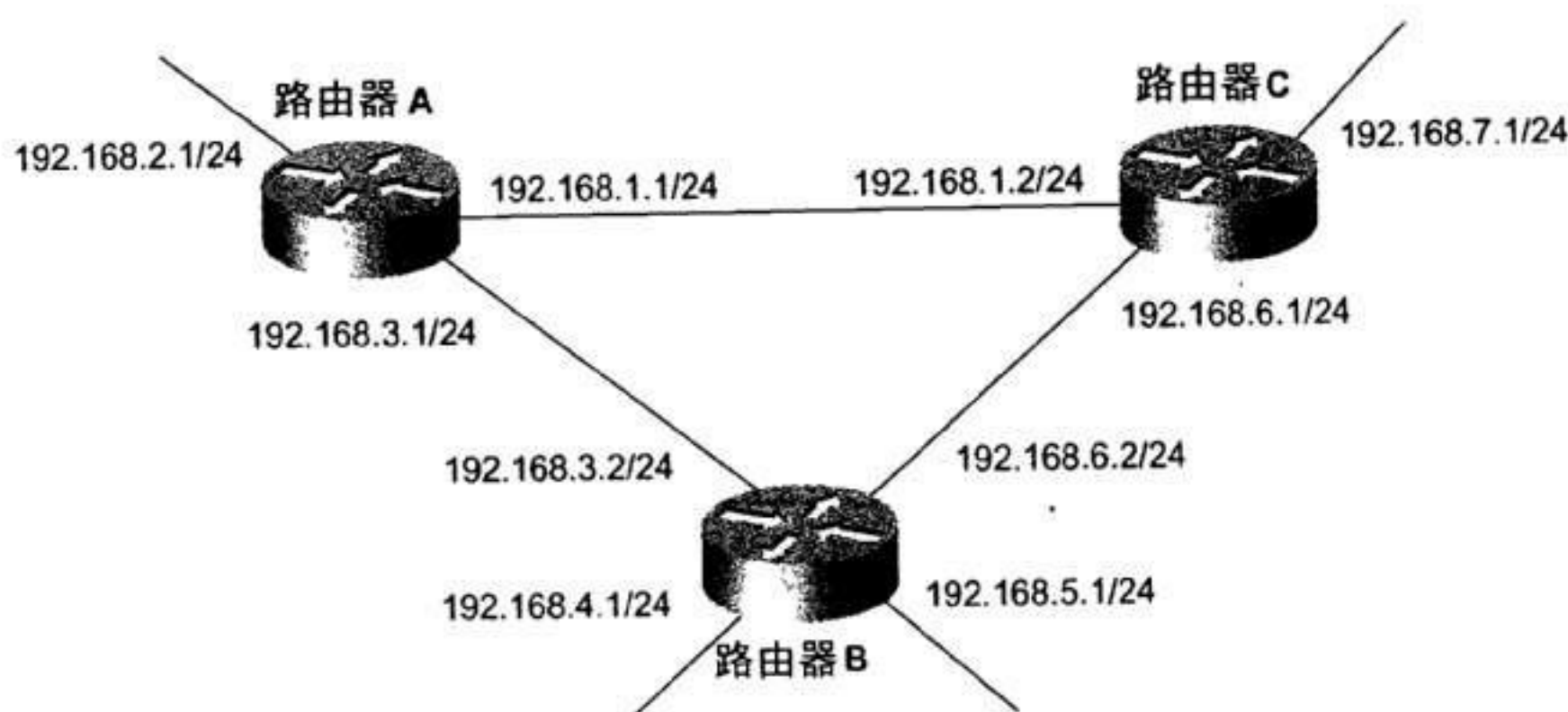


图 4-1 每个路由器从分配给它的接口的地址和掩码可以知道它的直连网络

信息共享过程乍看上去很简单。考虑路由器 A:

步骤 1: 路由器 A 检查自己的 IP 地址和相关掩码, 然后推导出与自身所连接的网络是 192.168.1.0、192.168.2.0 和 192.168.3.0。

步骤 2: 路由器 A 将这些网络连同标记一起保存到路由选择表中, 其中标记指明了网络是直连网络。

步骤 3: 路由器 A 向报文中加入以下信息: “我的直连网络是 192.168.1.0、192.168.2.0 和 192.168.3.0。”

步骤 4: 路由器 A 向路由器 B 和 C 发送这些路由信息报文的拷贝，这些路由信息或者叫做路由更新报文。

步骤 5: 路由器 B 和 C 执行与 A 完全相同的步骤，并且也向路由器 A 发送更新报文，其中报文包括与它们直连相连的网络。路由器 A 将接收到的信息连同发送该更新报文的路由器的源地址一起写入路由选择表。现在路由器 A 知道了所有的网络，而且还知道连接这些网络的路由器的地址。

这个过程看似非常简单。那么为什么路由选择协议比这更复杂呢？让我们重新看一下图 4-1。

- 路由器 A 将来自 B 和 C 的更新信息保存到路由选择表之后，它应该用这些信息作什么？例如，路由器 A 是否应该将 C 的信息传递给 B 并且将 B 的信息传递给 C 呢？
- 如果路由器 A 没有转发这些信息，那么就不能完成信息共享。例如，如果 B 和 C 之间的链路不存在，那么这两个路由器就无法知道对方的网络。因此路由器 A 必须转发那些更新信息，但是这样做又产生了新的问题。
- 如果路由器 A 从路由器 B 和 C 那里知道网络 192.168.4.0，那么为了到达该网络应该使用哪一个路由器呢？B 和 C 都合法吗？谁是最优路径呢？
- 什么机制可以确保所有路由器接收到所有的路由信息，而且这种机制还可以阻止更新报文在互联网中无休止地循环下去呢？
- 如果多个路由器可能共享某个直连网络（192.168.1.0、192.168.3.0 和 192.168.6.0），那么路由器是否仍旧应该通告这些网络呢？

这些问题同开头解释路由选择协议一样显得有点过分单纯，但是它们给读者的感觉却是：正是这些问题造成了协议的复杂性。每种路由选择协议无论如何都需要解决这些问题，这在后继的章节中将会变得更加清楚。

4.1.2 度量

当有多条路径到达相同目标网络时，路由器需要一种机制来计算最优路径。度量是指派给路由的一种变量，作为一种手段，度量可以按最好到最坏，或按最先选择到最后选择对路由进行等级划分。考虑下面的例子，了解为什么需要度量。

如图 4-1，假设在互联网中信息共享可以正常进行，并且路由器 A 中的路由选择表如表 4-1 所示。

表 4-1 有关图 4-1 中路由器 A 的一个不完善的路由选择表

网 络	下一跳路由器
192.168.1.0	直接被连接
192.168.2.0	直接被连接
192.168.3.0	B, C
192.168.4.0	B, C
192.168.5.0	B, C
192.168.6.0	B, C
192.168.7.0	B, C

路由选择表说明前 3 个网络直接连接到路由器，因而从路由器到达它们不要进行路由选

择。根据路由选择表, 后 4 个网络需要经过路由器 B 或 C 才能到达。这些信息都是正确的。但是如果通过 B 或 C 都可以到达网络 192.168.7.0, 那么优先选择那一条路径呢? 这时就需要度量对两条路径进行等级划分。

不同的路由选择协议使用不同的度量, 有时还使用多个度量。例如, RIP 定义含有路由器跳数最少的路径是最优路径; IGRP 基于路径沿路最小带宽和总时延定义最优路径。下面一节将给出这些度量和其他常用度量的基本定义。更复杂的内容——例如路由选择协议怎样使用多个度量以及如何处理度量值相同的路由——将在本书的后面章节讨论。

1. 跳数 (Hop Count)

跳数 (Hop Count) 度量可以简单地记录路由器跳数。例如, 如果报文从路由器 A 的接口 192.168.3.1 发出, 经过路由器 B 到达网络 192.168.5.0, 记为 1 跳; 如果从路由器接口 192.168.1.1 发出, 经路由器 C 和 B 到达网络 192.168.5.0, 记为 2 跳。假设仅使用跳数作为度量, 那么最优路径就是跳数最少的路线, 在本例中就是 A-B。

但 A-B 是真正的最优路径吗? 如果 A-B 是一条 DS-0 链路, A-C 和 C-B 都是 T1 链路, 那么跳数为 2 的路由则是最优路径, 因为带宽对业务通过网络的效率的影响最大。

2. 带宽 (Bandwidth)

带宽 (Bandwidth) 度量将会选择高带宽路径, 而不是低带宽路径。然而带宽本身可能不是一个好的度量。如果两条 T1 链路或其中一条被其他流量过多占用, 那么与一个 56K 的空闲链路相比到底谁好呢? 或者一条高带宽但时延也很大的链路又如何呢?

3. 负载 (Load)

负载 (Load) 度量反应了占用沿途链路的流量大小。最优路径应该是负载最低的路径。

不像跳数和带宽, 路径上的负载会发生变化, 因而度量也会跟着变化。这里需要当心。如果度量变化过于频繁, 路由翻动——最优路径频繁变化——可能就发生了。路由翻动会对路由器的 CPU、数据链路的带宽和全网稳定性产生负面影响。

4. 时延 (Delay)

时延 (Delay) 是度量报文经过一条路径所花费的时间。使用时延作度量的路由选择协议将会选择使用最低时延的路径作为最优路径。有多种方法可以度量时延。时延不仅要考虑链路时延, 而且还要考虑路由器的处理时延和队列时延等因素。另一方面, 路由的时延可能根本无法度量。因此, 时延可能是沿路径各接口所定义的静态延时量的总和, 其中每个独立的时延量是基于连接接口的链路类型估算而得到的。

5. 可靠性 (Reliability)

可靠性 (Reliability) 度量是用以度量链路在某种情况下发生故障的可能性, 可靠性可以是变化的或固定的。链路发生故障的次数或特定时间间隔内收到错误的次数都是可变可靠性度量的例子。固定可靠性度量是基于管理员确定的一条链路的已知量。可靠性最高的路径将会被最优先选择。

6. 代价 (Cost)

由管理员设置的代价 (Cost) 度量可以反应路由的等级。通过任何策略或链路特性可以对代价进行定义, 同时代价也可以反应出网络管理员意见的独断性。

每当谈论起路由选择的话题时，常常会把代价作为一个通用术语。例如，“RIP 基于跳数选择代价最低的路径”。但还有一个通用术语是最短，如“RIP 基于跳数选择最短路径。”当在这种情况下使用它们时，最小代价（最高代价）或最短（最长）仅仅指的是路由选择协议基于自己特定的度量对路径的一种看法。

4.1.3 收敛

动态路由选择协议必须包含一系列过程，这些过程用于路由器向其他路由器通告本地的直连网络，接收并处理来自其他路由器的同类信息，中继从其他路由器接收到的信息。此外，路由选择协议还需要定义决策最优路径的度量。

对路由选择协议来说另一个标准是互联网络上所有路由器的路由选择表中的可达信息必须一致。在图 4-1 中，如果路由器 A 确定了经过路由器 C 到达网络 192.168.5.0 是最优路径，而路由器 C 确定到达相同网络的最优路径是经过路由器 A，那么路由器 A 发向 192.168.5.0 的报文到达 C 后又被发回给 A，A 又再次发给 C，如此往复循环。我们称这种在两个或多个目标网络之间的持续流量循环为路由环路（Routing Loop）。

使所有路由选择表都达到一致状态的过程叫做收敛（Convergence）。全网实现信息共享以及所有路由器计算最优路径所花费的时间的总和就是收敛时间。

图 4-2 所示的互联网络已经收敛，但是现在拓扑又发生了变化。最左边两台路由器之间的链路发生故障，这两个直接相连的路由器都从数据链路协议获知链路故障，转而通知它们的邻居该链路不再可用。邻接路由器因而马上更新路由选择表并通知它们的邻居，这个过程一直持续到所有路由器都知道此变化。

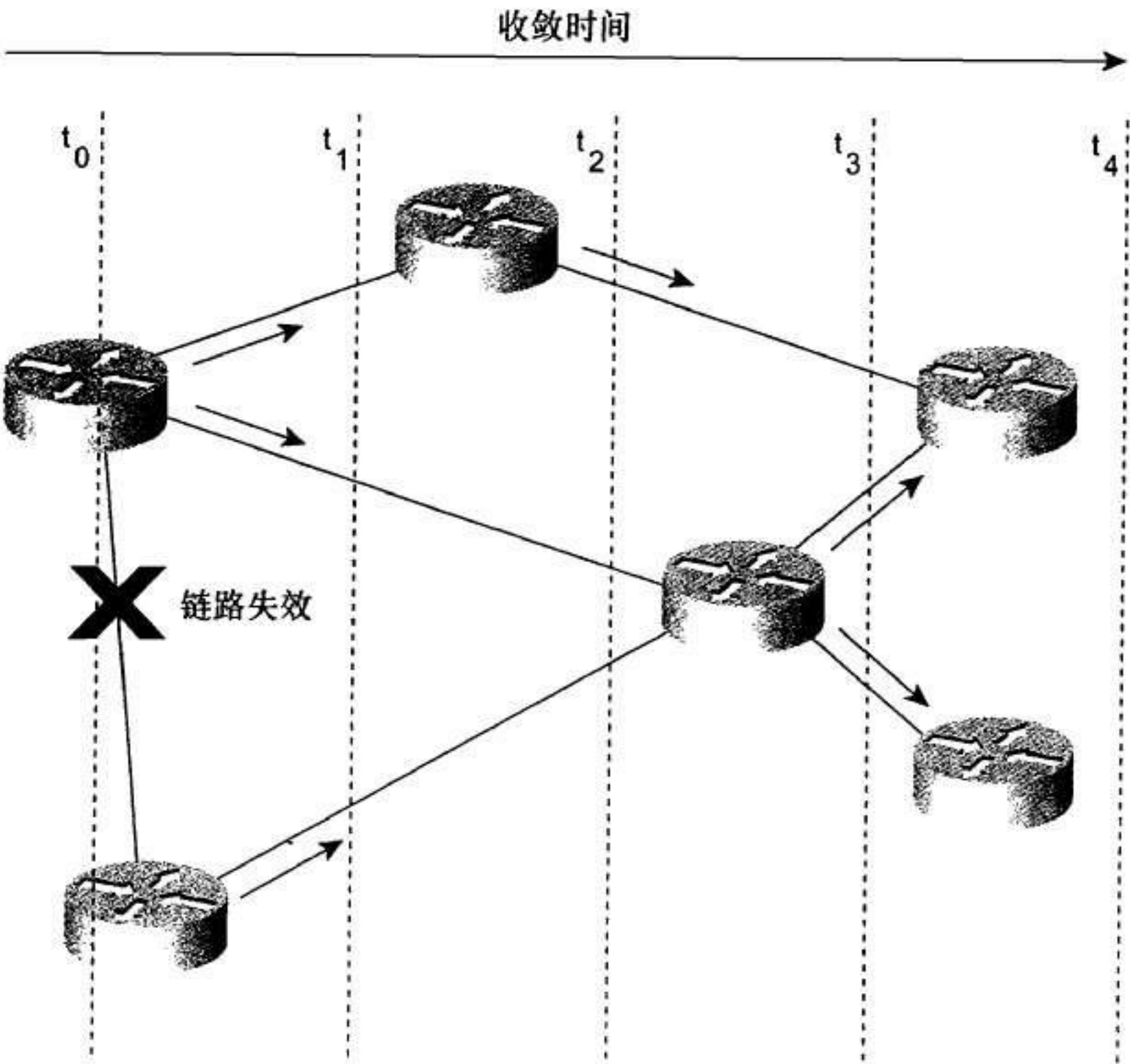


图 4-2 拓扑发生变化后重新收敛需要一定时间。当互联网络处于未收敛状态时，路由器易受到错误路由选择信息的影响

注意: 在 t_2 时刻, 最左边的 3 台路由器知道拓扑发生了变化, 但最右边 3 台路由器依然不知道。最右边 3 台路由器仍旧保存着原来的路由信息并继续交换报文。这时互连网络处于未收敛状态, 正是在这段时间里可能发生路由选择错误。因此在任何路由选择协议里收敛时间都是一个重要的因素。在拓扑发生变化之后, 一个网络收敛速度越快, 说明路由选择协议越好。

4.1.4 负载均衡

回忆一下第 3 章“静态路由”, 为了有效地使用带宽, 负载均衡作为一种手段, 将流量分配到通向相同目标网络的多条路径上。为了讨论负载均衡的有效性, 让我们再回过头看一下图 4-1。图中所有网络都存在两条可达路径。如果网络 192.168.2.0 上的设备向 192.168.6.0 上的设备发送一组报文流, 路由器 A 可以经过 B 或 C 发送这些报文。在这两种情况下, 到目的网络的距离都是 1 跳。然而, 在一条路径上发送所有的报文不能最有效地利用可用带宽。因此应该执行负载均衡交替使用两条路径。正如第 3 章所述, 负载均衡可以是等代价或不等代价, 可以是基于报文或基于目标地址的。

4.2 距离矢量路由选择协议

大多数路由选择协议都属于这两类之一: 距离矢量 (Distance Vector) 和链路状态 (Link State)。这里会分析距离矢量路由选择协议的基础内容, 在下一节将讨论链路状态路由选择协议。距离矢量算法是以 R.E.Bellman¹、L.R.Ford 和 D.R.Fulkerson²所做的工作为基础的, 由于这个原因, 所以有时距离矢量算法又称为 *Bellman-Ford* 或 *Ford-Fulkerson* 算法。

距离矢量名称的由来是因为路由是以矢量 (距离, 方向) 的方式被通告出去的, 其中距离是根据度量定义的, 方向是根据下一跳路由器定义的。例如, “朝下一跳路由器 X 的方向可以到达目标 A, 距此 5 跳之远”。这个表述隐含了每个路由器向邻接路由器学习它们所观察到的路由信息, 然后再向外通告自己观察到的路由信息。因为每个路由器在信息上都依赖于邻接路由器, 而邻接路由器又从它们的邻接路由器哪里学习路由, 依次类推, 所以距离矢量路由选择有时又被认为是“依照传闻进行路由选择”。

下面都属于距离矢量路由选择协议:

- IP 路由选择信息协议 (RIP)
- Xerox 网络系统的 XNS RIP
- Novell 的 IPX RIP
- Cisco 的 Internet 网关路由选择协议 (IGRP)
- DEC 的 DNA 阶段 4
- Apple Talk 的路由选择表维护协议 (RTMP)

1 R.E.Bellman. 动态规划。普林斯顿, 新泽西: 普林斯顿大学出版: 1957。

2 L.R.Ford Jr.和 D.R.Fulkerson. 网络中的流。普林斯顿, 新泽西: 普林斯顿大学出版: 1962。

通用属性

在一个使用路由选择算法的典型距离矢量路由选择协议中，路由器通过广播整个路由选择表，定期地向所有邻居发送路由更新信息。¹

上面这个表述包含了大量信息，下一节将会更详细地讨论。

1. 定期更新(Periodic Updates)

定期更新意味着每经过特定时间周期就要发送更新信息。这个时间周期从 10s (AppleTalk RTMP) 到 90s (Cisco 的 IGRP)。这里引起争论的是如果更新信息发送过于频繁可能会引起拥塞；但如果更新信息发送不频繁，收敛时间可能长的不能被接收。

2. 邻居(Neighbours)

在路由器的语境中，邻居通常意味着共享相同数据链路的路由器。距离矢量路由选择协议向邻接路由器²发送更新信息，并依赖邻居向它的邻居传递更新信息。因此，距离矢量路由选择被说成使用逐跳更新方式。

3. 广播更新(Broadcast Updates)

当路由器首次在网络上被激活时，路由器怎样寻找其他路由器呢？它又是怎样宣布自己的存在呢？这里有几种方法可用。最简单的方法是向广播地址发送(在 IP 网中，广播地址是 255.255.255.255)更新信息。使用相同路由选择协议的邻居路由器将会收到广播报文并且采取相应的动作。不关心路由更新信息的主机和其他设备仅仅丢弃该报文。

4. 包含整个路由选择表的更新信息

大多数距离矢量路由选择协议使用非常简单的方式告诉邻居它所知道的一切，该方式就是广播它的整个路由选择表，但在下一节中我们会讨论几个特例。邻居在收到这些更新信息之后，他们会收集自己需要的信息，其他则被丢弃。

5. 依照传闻进行路由选择

在图 4-3 中，正在执行一个距离矢量算法，其中使用跳数作为度量。在 t_0 时刻，路由器 A 到 D 正好可用。让我们沿最上面一行查看路由选择表，4 台路由器在 t_0 时刻所具有的惟一信息就是它们的直连网络。路由选择表标识了这些网络，并且指明它们没有经过下一跳路由器，是直接连接到路由器上的，所以跳数为 0。每台路由器都将在它所有的链路上广播这些信息。

在 t_1 时刻，路由器接收并处理第 1 个更新信息。在 t_1 时刻，查看路由器 A 的路由选择表，发现路由器 B 发给路由器 A 的更新信息说路由器 B 能够到达网络 10.1.2.0 和 10.1.3.0，而且距离都为 0 跳。如果这些目标网络距离 B 为 0 跳，那么距离 A 则为 1 跳。所以路由器 A 将跳数增加 1，然后检查自己的路由选择表。路由选择表中显示网络 10.1.2.0 已知，且距离为 0 跳，小于 B 通告的跳数，所以路由器 A 忽略此信息。

¹ Cisco 的增强型 IGRP 明显不同于这种协定。EIGRP 虽然是距离矢量协议，但是它既不定期发送更新信息，而且更新信息也不是整个路由选择表。第 8 章将会讨论 EIGRP。

² 这个表述不完全正确。在某些实现中主机也监听路由更新信息；但是在这个讨论中重要的是路由器如何工作。



图 4-3 距离矢量协议逐跳收敛

由于网络 10.1.3.0 对路由器 A 来说是新信息，所以 A 将其输入到路由选择表中。因为更新报文的源地址是路由器 B 的接口地址 (10.1.2.2)，因此该地址连同计算的跳数一起被保存到路由选择表中。

注意：在 t_1 时刻其他路由器也执行了类似的操作。例如，路由器 C 忽略了来自 B 关于 10.1.3.0 的信息以及来自 C 关于 10.1.4.0 的信息，但是保存了以下信息：经过 B 的接口地址 10.1.3.1 可以到达网络 10.1.2.0 以及经过 D 的接口地址 10.1.4.2 可以到达网络 10.1.5.0。经计算得知 C 到这两个网络的距离都为 1 跳。

在 t_2 时刻，随着更新周期再次到期，另一组更新报文被广播。路由器 B 发送了最新的路由选择表。路由器 A 再次将 B 通告的跳数加 1 后与自己的路由选择表相比较。像上次一样，A 又一次丢弃了关于 10.1.2.0 的信息。由于网络 10.1.3.0 已知且跳数没有发生变化，所以该信息也被丢弃。唯有 10.1.4.0 被作为新的信息输入到路由选择表中。

在 t_3 时刻，网络收敛。每台路由器都已经知道了每个网络以及到达每个网络的下一跳路由器地址和距离跳数。

这里打个比方。你正在新墨西哥洲北部的 Sangre de Cristo 山中漫步，如果你不会迷路的话，这里是一个迷人的地方。但是如果你迷路了。你偶遇到一个叉路口，一个路标指向西面，上面写着“陶斯镇，15 英里”。这时你除了相信这个路标外别无选择。你不知道 15 英里外的地形是什么，你也不知道是否有更好的路，或者这个路标是否正确。如果有人将路标转个方向，那么你不但不能去往安全的地方反而走向森林的更深处！

距离矢量算法提供了指向网络的路标。¹该算法给出了方向和距离，但是没有给出沿着这条路径行走的细节。就像叉路口的路标一样，它很容易受到意外或故意的破坏。下面是距离矢量算法所面临的一些困难及算法的改进。

6. 路由失效计时器

在图 4-3 中，既然互联网络已经收敛，那么当部分网络拓扑发生变化时，它怎样处理重新收敛问题呢？如果网络 10.1.5.0 发生故障，答案很简单——在下一个更新周期中，路由器 D 将网络标记为不可达并且发送该信息。

但是如果网络 10.1.5.0 没有故障，而是路由器 D 发生故障该怎么办？这时仍然保存在路由器 A、B 和 C 的路由选择表中关于网络 10.1.5.0 的信息将不再有用，但是却没有路由器通知它们。它们将不知不觉地向一个不可达的网络转发着报文——互联网络中打开了一个黑洞。

处理这个问题的办法是为路由选择表中的每个表项设置路由失效计时器。例如，当路由器 C 首次知道 10.1.5.0 并将其输入到路由选择表中时，路由器 C 将为该路由设置计时器。每隔一定时间间隔 C 都会收到 D 的更新信息，C 在丢弃有关 10.1.5.0 的信息的同时置位该路由的计时器。

如果路由器 D 发生故障，C 将不能接收到关于 10.1.5.0 的更新信息。这时计时器将会超时，C 将把该路由标记为不可达路由，并且将其附带在下一个更新信息中。

路由超时的典型周期范围是 3~6 个更新周期。路由器在丢失单个更新信息之后将不会使路由无效的，因为报文的损坏、丢失或者某种网络延时都会造成这种事件的发生。但是，如果路由失效周期太长，网络收敛速度将会过慢。

7. 水平分隔 (Split Horizon)

根据到目前为止所描述的距离矢量算法，每台路由器在每个更新周期都要向每个邻居发送它的整个路由选择表。但是这真的有必要吗？在图 4-3 中，路由器 A 知道的每个距离大于 0 跳的网络都是从路由器 B 学习来的。常识表明，如果路由器 A 将学自路由器 B 的网络再广播给 B，那么这是一种资源浪费。显然，B 已经知道这些网络。

逆向路由 (Reverse Route)——路由的指向与报文流动方向相反的路由。水平分隔是一种在两台路由器之间阻止逆向路由的技术。

这样做除了不会浪费资源，还有一个很重要的原因是不会把从路由器学习到的可靠信息再返回给这台路由器。动态路由选择协议最重要的功能是监测和抵消拓扑变化——如果到网络的最优路径不可用，协议必须寻找下一个最优路径。

再看一下图 4-3 中已收敛的网络，假设网络 10.1.5.0 发生故障。路由器 D 监测到该故障，D 将网络标记为不可达并在下一更新周期通知 C。然而在 D 的更新计时器触发更新之前，意想不到的事情发生了。C 的更新报文到达了 D，声明 C 可以到达网络 10.1.5.0，距离为 1 跳！还记得上面路标的比喻吗？路由器 D 不知道 C 通告的下一条最优路径并不合理，因而 D 将跳数加 1 并在路由选择表中记录以下信息：通过路由器 C 的接口 (10.1.4.1) 可以到达网络 10.1.5.0，距离为 2 跳。

此刻目标地址为 10.1.5.3 的报文到达路由器 C，C 查询路由选择表并将报文转发给 D。D 查询路由选择表又将报文转发给 C，C 再转回给 D，一直无穷尽地进行下去。路由环路发

¹ 一般是拿路标作比方。你可以在 Radia Perlman 的 *Interconnections* 一书中找到一个好的表达，见 205 页—210 页。

生了。

执行水平分隔可以阻止路由环路的发生。有两类水平分隔方法：简单水平分隔法和毒性逆转水平分隔法。

简单水平分隔的规则是，当更新报文被发送出某接口时，更新信息中不能包含从该接口接收的更新信息中获取到的网络。

在图 4-4 中，路由器执行简单的水平分隔。路由器 C 向 D 发送了关于网络 10.1.1.0、10.1.2.0 和 10.1.3.0 的更新信息。其中没有包含网络 10.1.4.0 和 10.1.5.0，因为它们是从路由器 D 获取的。同样，发送给 B 的更新信息包括了网络 10.1.4.0 和 10.1.5.0，而没有提及 10.1.1.0、10.1.2.0 和 10.1.3.0。

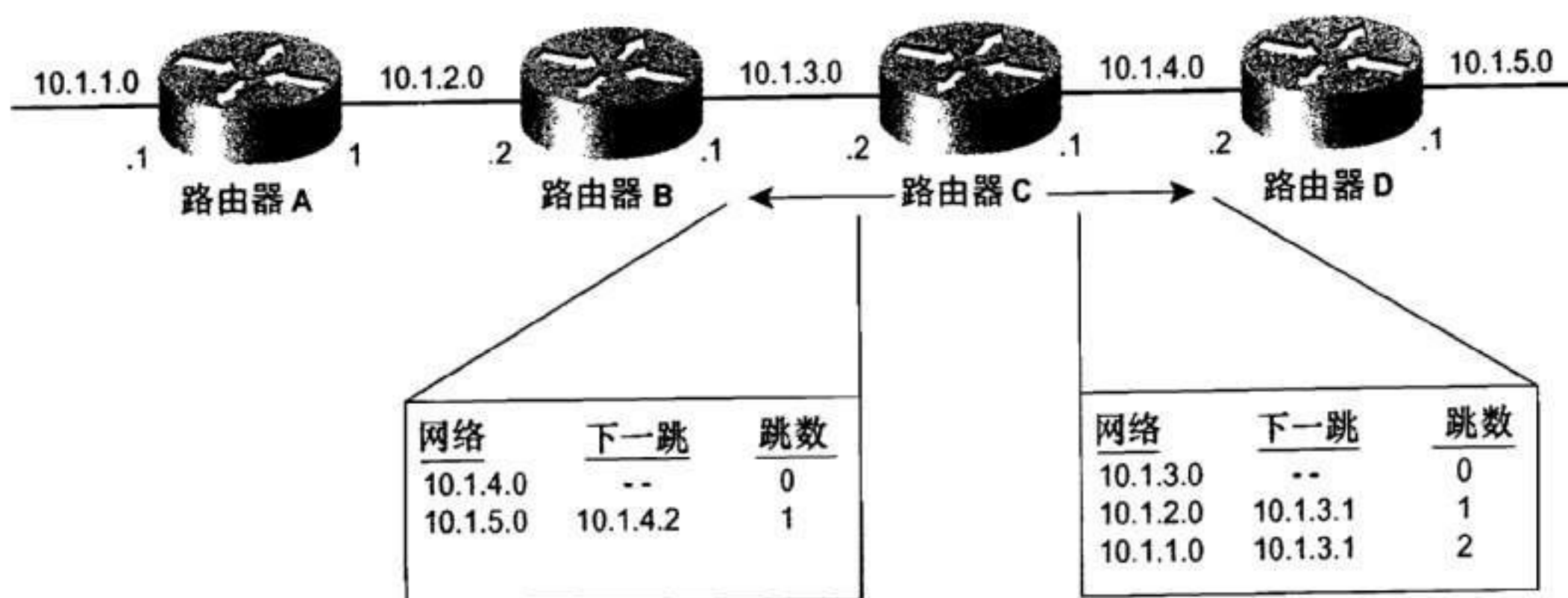


图 4-4 简单水平分隔没有把从邻居那里获取的路由通告给邻居

简单水平分隔采用抑制信息的工作方式。毒性逆转水平分隔法是一种改进方法，它可以提供更积极的信息。

毒性逆转水平分隔法的规则是，当更新信息被发送出某接口时，信息中将指定从该接口接收到的更新信息中获取的网络是不可达的。

在图 4-4 中，事实上，路由器 C 向 D 通告了网络 10.1.4.0 和 10.1.5.0，但是这些网络都被标记为不可达。图 4-5 给出了看似从 C 到 B 和 D 的路由选择表。注意，通过设置度量为无穷大可以标记网络不可达。换言之，网络无穷远。下一节将讨论路由选择协议中无穷大的概念。

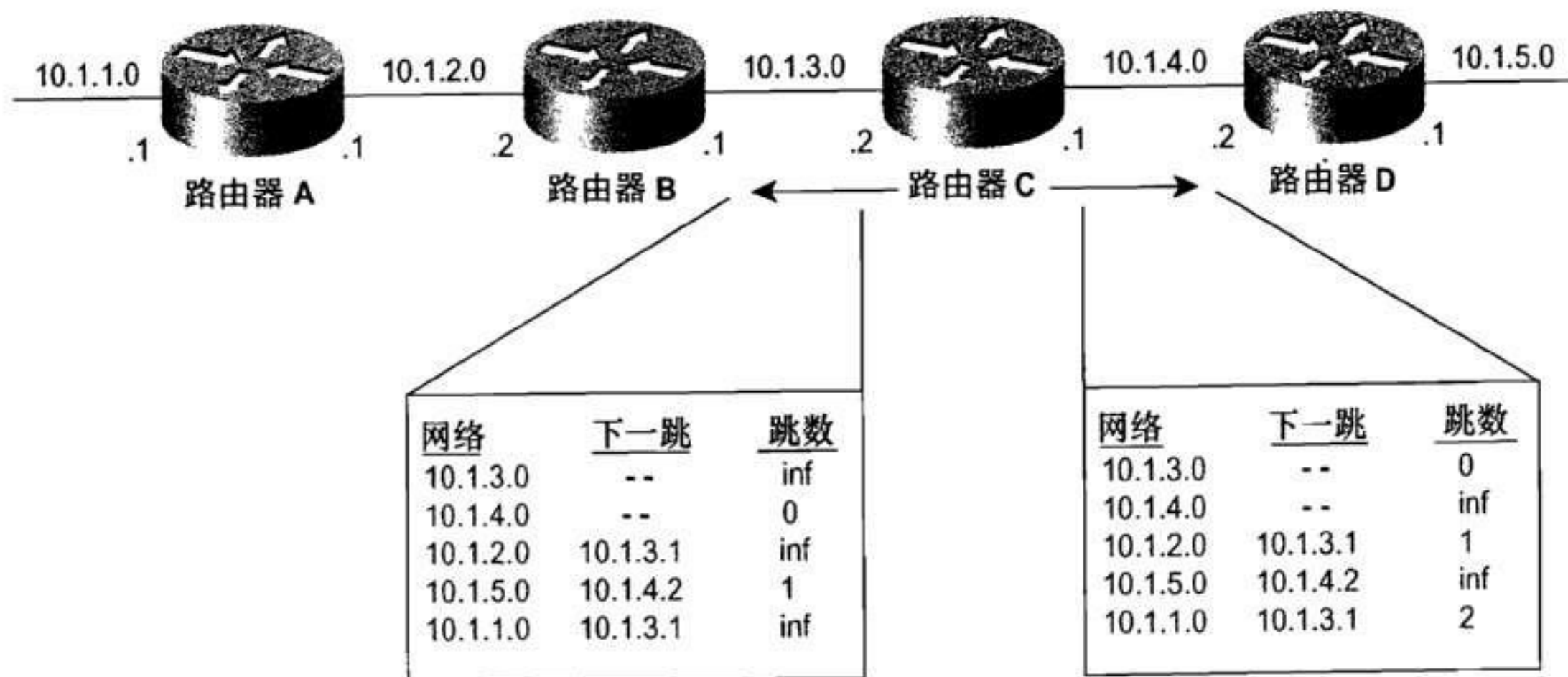


图 4-5 虽然毒性逆转水平分隔法通告逆向路由，但使用了不可达度量（无穷大）

毒性逆转水平分隔法被认为比简单水平分隔法更安全更健壮——一种“坏信息总比没消息好”的方法。例如在图 4-5 中, 假设路由器 B 收到错误信息使其相信经过路由器 C 可以到达子网 10.1.1.0。简单水平分隔法无法纠正这种错误理解, 而路由器 C 的毒性逆转更新信息可以立刻制止这种潜在的环路。正因如此, 大部分现代距离矢量算法的实现都使用了毒性逆转水平分隔法。这样做使路由更新报文更大了, 可能会加剧链路的拥塞问题。

8. 计数到无穷大

水平分隔法切断了邻居路由器之间的环路, 但是它不能割断网络中的环路, 如图 4-6, 这里还是 10.1.5.0 发生故障。路由器 D 向路由器 C (虚箭头) 和 B (实箭头) 发送了相应的更新信息。于是路由器 B 将经过 D 的路由标记为不可达, 而此时路由器 A 正在向外通告下一条到达 10.1.5.0 的最优路径, 距离为 3 跳。因此 B 在路由选择表中记录下此路由。

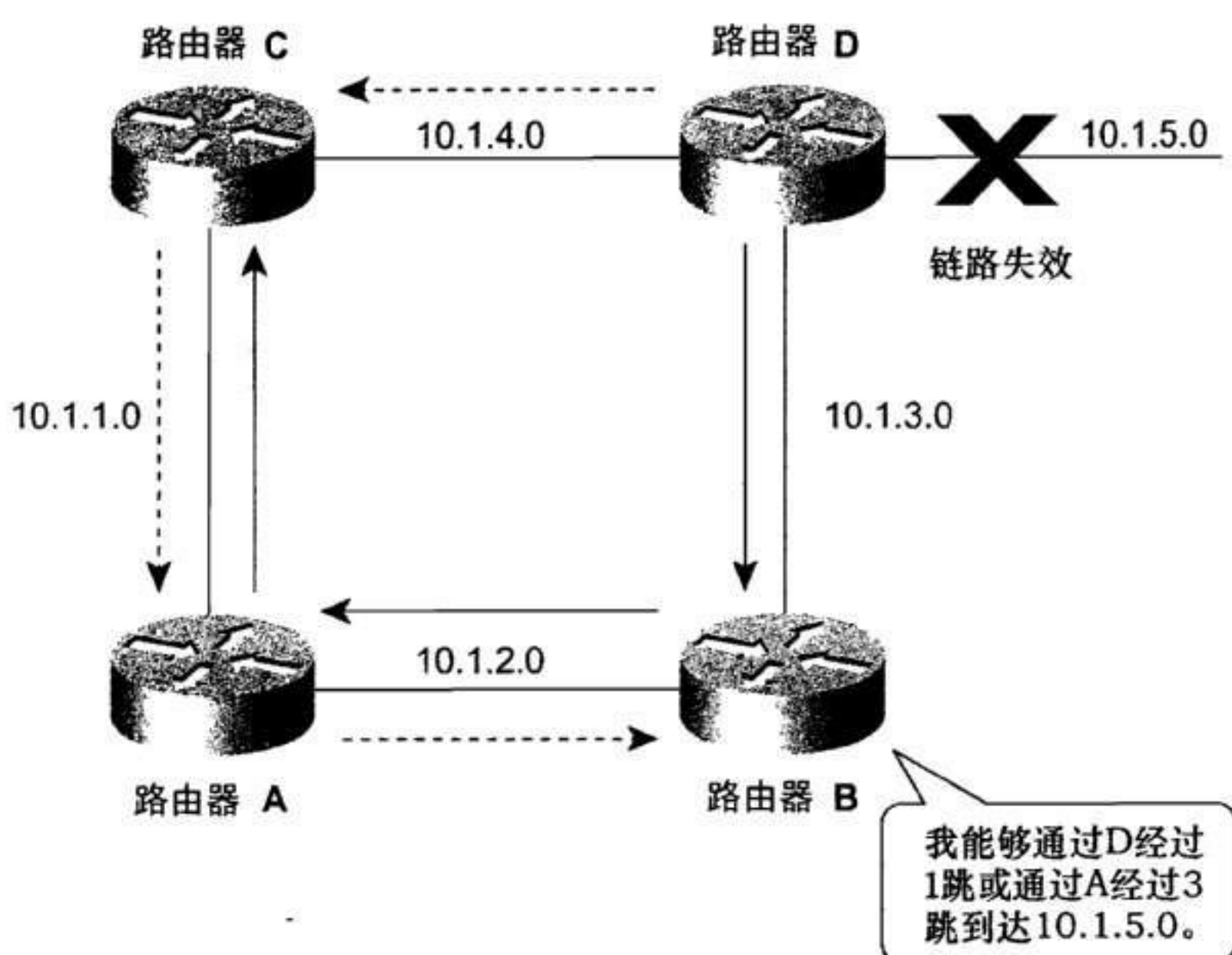


图 4-6 水平分隔法不能阻止这里的路由环路

B 现在又通知 D 它有另一条路由可以到达 10.1.5.0。于是 D 也记录下这个路由, 并通知 C 说它有一条距离 10.1.5.0 4 跳的路由。C 又告诉 A 距离 10.1.5.0 有 5 跳远。A 告诉 B 有 6 跳远。

“啊”, 路由器 B 想, “路由器 A 到网络 10.1.5.0 的路径在不断加长。不过它是惟一可用的路径, 所以我将使用这条路径!”

B 又将跳数加到 7, 并通知 D, 如此循环下去。这种情况就叫计数到无穷大, 因为到 10.1.5.0 的跳数将会持续增加到无穷大。虽然所有路由器都执行了水平分隔, 但对此无能为力。

减小计数到无穷大影响的方法是定义无穷大。大多数距离矢量协议定义无穷大为 16 跳。在图 4-6 中, 随着更新报文在路由器中转圈, 到 10.1.5.0 的跳数最终将增加到 16。那时网络 10.1.5.0 将被认为不可达。

这也是路由器如何通告网络不可达的一种方法。一个网络发生故障, 不管它是毒性逆转路由, 还是超过最大网络尺寸 15 的路由, 路由器将把所有跳数为 16 的路由看作不可达。

设置最大跳数 15 有助于解决计数到无穷大的问题, 但是收敛速度仍旧非常慢。假设更

新周期为 30s, 网络可能花 7.5min 达到收敛, 在这期间容易受到路由错误的影响。两种加快重新收敛速度的方法是触发更新和挂起计时器。

9. 触发更新(Triggered Update)

触发更新又叫快速更新, 非常简单: 如果一个度量变好或变坏, 那么路由器将立即发送更新信息, 而不等更新计时器超时。这样重新收敛的速度将会比每台路由器必须等待更新周期的方式快, 而且可以大大减少计数到无穷大所引发的问题, 虽然不能完全消除。定期更新和触发更新可以并存, 因而路由器可能会在收到来自触发更新的正确信息之后收到来自未收敛路由器的错误信息。这种情况表明, 当网络正在进行重新收敛时, 还会发生混乱和路由错误。但是触发更新将有助于更快地消除这些问题。

对触发更新进一步的改进是更新信息中仅包括实际触发该事件的网络, 而不是包括整个路由选择表。触发更新技术减少了处理时间和对网络带宽的占用。

10. 抑制计时器(Holddown Timer)

触发更新为正在重新进行收敛的网络增加了应变能力。为了降低接受错误路由信息的可能性, 抑制计时器引入了某种程度的怀疑量。

如果到一个目标的距离增加(例如, 跳数由 2 增加到 4), 那么路由器将为该路由设置抑制计时器。直到计时器超时, 路由器才可以接受有关此路由的更新信息。

显然, 这也是一种折衷办法。错误路由信息进入路由选择表的可能信被减小了, 但是重新收敛的时间也被耗费了。像其他计时器一样, 必须小心设置抑制计时器。如果挂起时间太长, 则不起作用; 如果太短, 正常路由会受到不利的影响。

11. 异步更新 (Asynchronous Update)

图 4-7 给出了一组连接在以太网骨干上的路由器。路由器将不同时广播更新信息, 如果发生这种情况, 更新报文会发生碰撞。但是当几台路由器共享一个广播网络时可能会发生这种情况。因为在路由器中, 更新处理所带来的系统时延导致更新计时器趋于同步。当几个路由器的计时器同步后, 碰撞随之发生, 这又进一步影响到系统时延, 最终共享广播网络的所有路由器都可能被同步起来了。

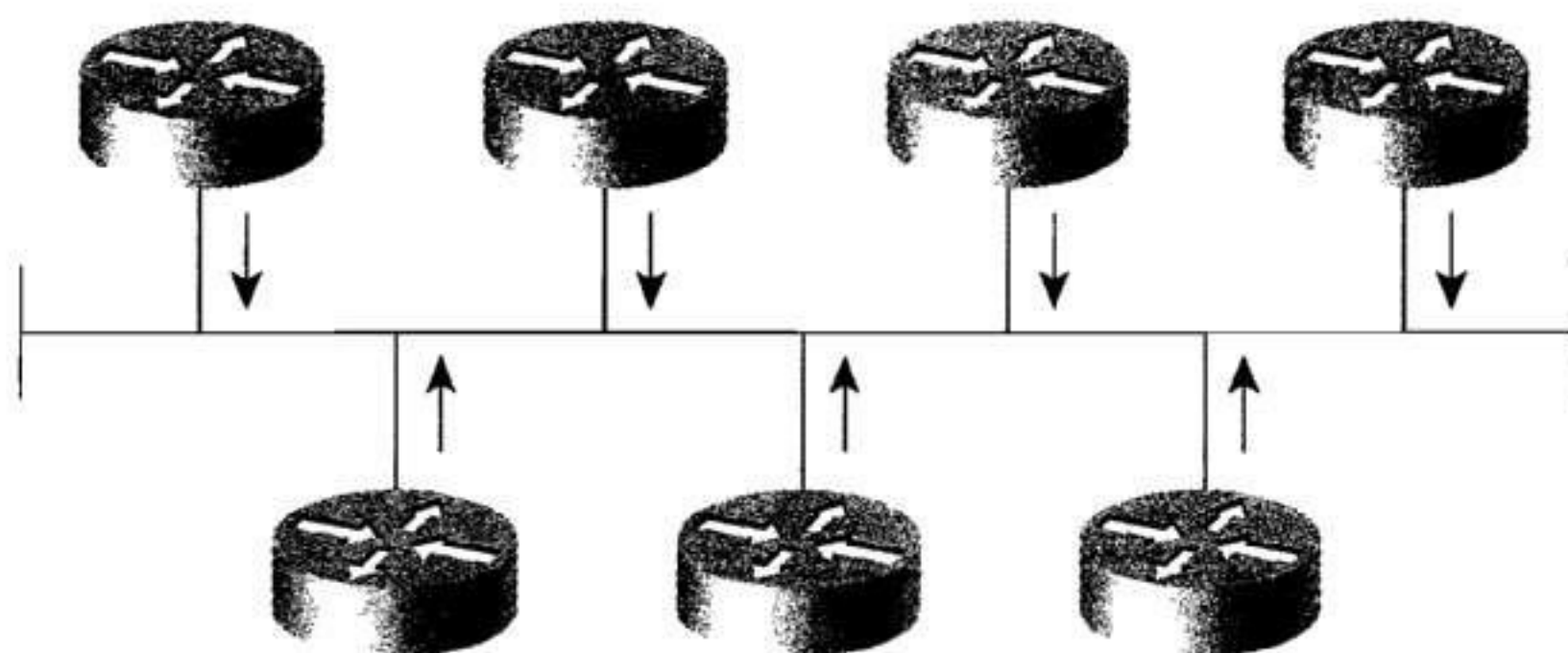


图 4-7 如果更新计时器同步, 就可能发生碰撞

可以使用下面两种方法维持异步更新:

- 每台路由器的更新计时器独立于路由进程, 因而不会受到路由器处理负载的影响。
- 在每个更新周期中加入一个小的随机时间或随机抖动作为偏移。

如果路由器实现了严格的、独立于系统的计时器方法，那么还必须按照随机的方式激活共享一个广播网络的所有路由器。因为并行启动整组路由器可能造成所有计时器同时发送更新信息。

如果随机变量与共享广播网络的路由器数量相比足够大，那么增加更新周期的随机性是有效的。Sally Floyd 和 Van Jacobson¹曾作过计算，在足够大的网络中，过小的随机化会被路由器所克服，为了保证有效性，更新计时器应该分布在中等更新周期的 50% 的范围内。

4.3 链路状态路由选择协议

距离矢量路由器所使用的信息可以比拟为由路标提供的信息。链路状态路由选择协议像是一张公路线路图。链路状态路由器是不容易被欺骗而做出错误的路由决策的，因为它有一张完整的网络图。链路状态不同于距离矢量依照传闻进行路由选择的工作方式，原因是链路状态路由器从对等路由器²那里获取第一手信息。每台路由器会产生一些关于自己、本地直连网络以及这些链路状态的信息。这些信息从一台路由器传送到另一台路由器，每台路由器都做一份信息拷贝，但是决不改动信息。最终目的是每台路由器都有一个相同的有关互连网络的信息，并且每台路由器可以独立地计算各自的最优路径。

链路状态协议，有时叫**最短路径优先协议**或**分布式数据库协议**，是围绕着图论中的一个著名算法——E.W.Dijkstra 的最短路径算法设计的。链路状态协议有以下一些：

- IP 开放式最短路径优先 (OSPF)
- CLNS 或 IP ISO 的中介系统到中介系统 (IS-IS)
- DEC 的 DNA 阶段 5
- Novell 的 Netware 链路服务协议 (NLSP)

虽然链路状态协议确实考虑的比距离矢量协议更复杂，但是基本功能却一点也不复杂。

步骤 1： 每台路由器与它的邻居之间建立联系，这种联系叫做邻接关系。

步骤 2： 每台路由器向每个邻居发送链路状态通告 (LSA)，有时叫链路状态报文 (LSP)。

对每条路由器链路都会生成一个 LSA，LSA 用于标识这条链路、链路状态、路由器接口到链路的代价度量值以及链路所连接的所有邻居。每个邻居在收到公告之后要依次向它的邻居转发这些通告 (泛洪)。

步骤 3： 每台路由器要在数据库中保存一份它所收到的 LSA 的备份。如果所有工作正常，所有路由器的数据库应该相同。

步骤 4： 完整的拓扑数据库，也叫链路状态库，描述了互连网络的地理图。每台路由器使用 Dijkstra 算法计算出到每个网络的最短路径，并将信息保存在路由选择表中。

4.3.1 邻居

邻居发现是建立链路状态环境并运转的第一步。与友邻术语技术一样，这一步使用 Hello

¹ S.Floyd 和 V.Jacobson。“周期性路由消息的同步。” ACM Sigcomm'93 Symposium, 1993 年 9 月。

² 就是所有使用系统路由选择协议的路由器。

协议 (Hello Protocol)。Hello 协议定义了一个 Hello 报文的格式和交换报文并处理报文内信息的过程。

Hello 报文至少应包含一个路由器 ID 和发送报文的网络地址。路由器 ID 可以将发送该报文的路由器与其他路由器唯一地区分开。例如, 路由器 ID 可以是路由器的一个接口的 IP 地址。报文的其他字段可以携带子网掩码、Hello 间隔、线路类型描述符和帮助建立邻居关系的标记, 其中 Hello 间隔是路由器在宣布邻居死亡之前等待的最大周期。

当两台路由器已经互相发现并将对方视为邻居时, 它们要进行数据库同步过程, 即交换和确认数据库信息直到数据库相同为止。数据库同步的细节见第 9 章和第 10 章。为了执行数据库同步, 邻居之间必须建立邻接关系, 即它们必须就某些特定的协议参数, 如计时器和可选能力的支持, 达成一致意见。通过使用 Hello 报文建立邻接关系, 链路状态协议就可以在受控的方式下交换信息。与距离矢量相对照, 这种方式仅在配置了路由选择协议的接口上广播更新信息。

除建立邻接关系之外, Hello 报文还可作为监视邻接关系的握手信号。如果在特定的时间内没有从邻接路由器收到 Hello 报文, 那么认为邻居路由器不可达, 随即邻接关系被解除。典型的 Hello 报文交换间隔为 10s, 典型的死亡周期是报文交换间隔的 4 倍。

4.3.2 链路状态泛洪扩散

在建立邻接关系之后, 路由器开始发送 LSA。从术语泛洪扩散 (Flooding) 我们可以得知, 通告被发送给每个邻居。路由器保存接收到的 LSA, 并依次向每个邻居转发, 除了发送该 LSA 的邻居之外。这个过程是链路状态优于距离矢量的原因。LSA 几乎是立刻被转发, 而距离矢量在发送路由更新之前必须运行算法并更新自身的路由选择表, 甚至对触发路由更新也是如此。因此, 当网络拓扑改变时, 链路状态协议收敛速度远远快于距离矢量协议。

泛洪扩散过程是链路状态协议中最复杂的一部分。有几种方式可以使泛洪扩散更高效和更可靠, 如使用单播和组播地址、校验和以及主动确认。在有关具体协议的章节中将讨论这些主题, 但是有两个过程对泛洪扩散是极其重要的: 排序和老化。

1. 序列号

到目前为止, 泛洪扩散的一个难点是当所有路由器收到所有 LSA 时, 泛洪扩散必须停止。报文中的 TTL 值仅仅能被依赖到超时, 但是直到 LSA 超时为止, TTL 几乎不可能有效地允许 LSA 在互联网中漫游。如图 4-8, 路由器 A 连接的子网 172.22.4.0 发生故障, 因而 A 向邻居 B 和 D 泛洪扩散一个 LSA, 以便通告该链路的新状态。B 和 D 忠实地向它们的邻居扩散该 LSA, 依次类推。

看一下在路由器 C 上会发生什么。在 t_1 时刻, 一个来自路由器 B 的 LSA 到达路由器 C, C 将相关信息输入到拓扑数据库, 并且向 F 进行转发。在 t_3 时刻, 相同 LSA 的另一份拷贝经 A-D-E-F-C 路径到达 C。路由器 C 发现数据库中已经存在该 LSA, 问题是 C 是否应该向 B 转发这个 LSA? 答案是不转发, 因为 B 已经收到了这个通告。由于 C 从 F 接收到的 LSA 的序列号与早先从 B 接受的 LSA 的序列号相同, 所以路由器 C 也知道这一情况。

当路由器 A 发送 LSA 时, 在每个拷贝中的序列号都是一样的。这个序列号同 LSA 的其他部分一起被保存在路由器的拓扑数据库中, 当路由器收到数据库中已有的 LSA 且序列号相

同时, 路由器将丢弃这些信息。如果信息相同但是序列号更大, 那么接收的信息和新序列号被保存到数据库中, 并且泛洪扩散这个 LSA。按照这种方式, 当所有路由器都收到 LSA 的最新拷贝时, 泛洪扩散将停止。

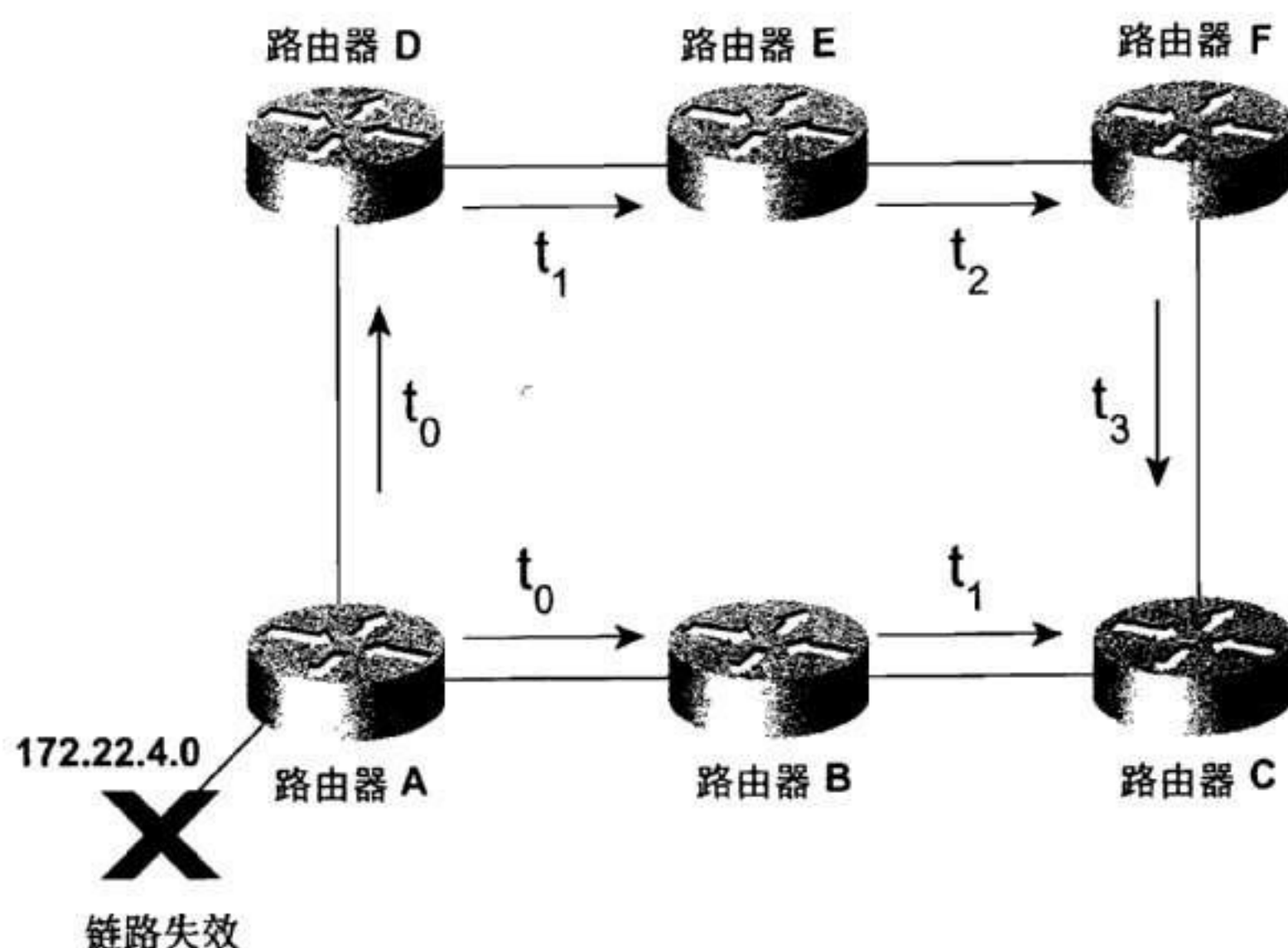


图 4-8 当拓扑发生变化时, 通告该变化的 LSA 在整个互联网络上扩散

按目前所描述, 路由器好像仅对数据库中 LSA 与新收到的 LSA 是否相同进行验证, 并据此作出扩散/丢弃决定, 并没有使用序列号。但是设想, 图 4-8 中的网络 172.22.4.0 在发生故障后马上恢复正常。路由器 A 发出通告网络故障的 LSA, 序列号为 166, 接着再发送通告网络正常的 LSA, 序列号为 167。路由器 C 先后接收到沿路径 A-B-C 扩散过来的 LSA, 分别是关于网络发生故障和故障恢复的通告, 但是接着路由器 C 又收到沿路径 A-D-E-F-C 扩散过来的关于网络故障的 LSA。使用序列号, C 的数据库将指明来自 A 的 LSA 的序列号为 167, 而后面到达的 LSA 序列号为 166, 因此该 LSA 被认为是过时的信息而被丢弃。因为序列号被携带在 LSA 中的一个固定字段内, 所以序列号一定有上界, 那么当序列号到达上界时会发生什么呢?

(1) 线性序列号空间

一个办法是使用一个非常大的线性序列号空间以至于根本不可能到达上界。例如, 如果使用 32 位长字段, 那么从 0 开始将有 $2^{32}=4\,294\,967\,296$ 个可用序列号。即使路由器每 10s 产生一个新的状态报文, 它也需要花 1361 年才能用尽所有序列号。几乎没有路由器被希望持续这样长的时间。

很不幸, 在这个不够完美的世界上, 故障依然会发生。如果一个链路状态进程用完了所有序列号, 那么它在重新使用最低序列号之前必须停止, 并等待它所发出的 LSA 在所有数据库中都不再被使用 (见本章“老化 (Aging)”部分)。

在路由器启动期间有一个更常见的困难。如果路由器 A 启动后, 它无法记得它上次使用的序列号, 必须重新使用 1。但是如果 A 的邻居仍然在数据库中保留了 A 上次的序列号, 那么越小的序列号也就是越老的序列号, 因而会被忽略。而且, 路由选择进程必须一直等到互联网络上所有陈旧的 LSA 都消失为止。假如最大的时间是 1 个小时或更长, 那么这种方法将不再有任何吸引力。

更好的解决办法是在泛洪扩散行为中添加新的规则：如果一个重新启动的路由器向邻居发送 LSA 的序列号比邻居保存的序列号还要老,那么邻居将向该路由器发送自己保存的 LSA 和序列号。这个路由器将知道启动前自己使用的序列号并作出相应的调整。

然而需要当心,最近使用过的序列号不能接近上界。否则,重新启动的路由器将不得不再次重新启动。必须设定规则限制路由器“跳跃”地使用序列号。例如,规则指明一次序列号的增加不能超过整个序列号空间的二分之一(实际公式比例子要复杂,因为要考虑年龄的限制)。

IS-IS 使用 32 位线性序列号空间。

(2) 循环序列号空间

另一种方法是使用循环序列号空间,数字是循环使用的——在 32 位空间内紧跟在 4 294 967 295 后面的是 0。然而在这里,故障也会让人左右为难,重新启动的路由器可能会遇到同线性序列号一样的问题。

循环序列编号建立了一个不合逻辑的奇特的位。如果 x 是 1 到 4 294 967 295 之间任意的一个数,那么 $0 < x < 0$ 。在运转正常的互网络中通过声明两条规则可以维持这种条件,其中声明的规则用来确定什么时候一个序列号大于或小于另一个序列号。假设序列号空间为 n ,有两个序列号 a 和 b ,如果满足以下任意一个条件,则认为 a 更新(数量更大):

- $a > b$ 且 $(a - b) \leq n/2$
- $a < b$ 且 $(b - a) > n/2$

为了简单起见,如图 4-9,我们使用一个 6 位序列号空间:

$$n = 2^6 = 64, \text{ 所以 } n/2 = 32。$$

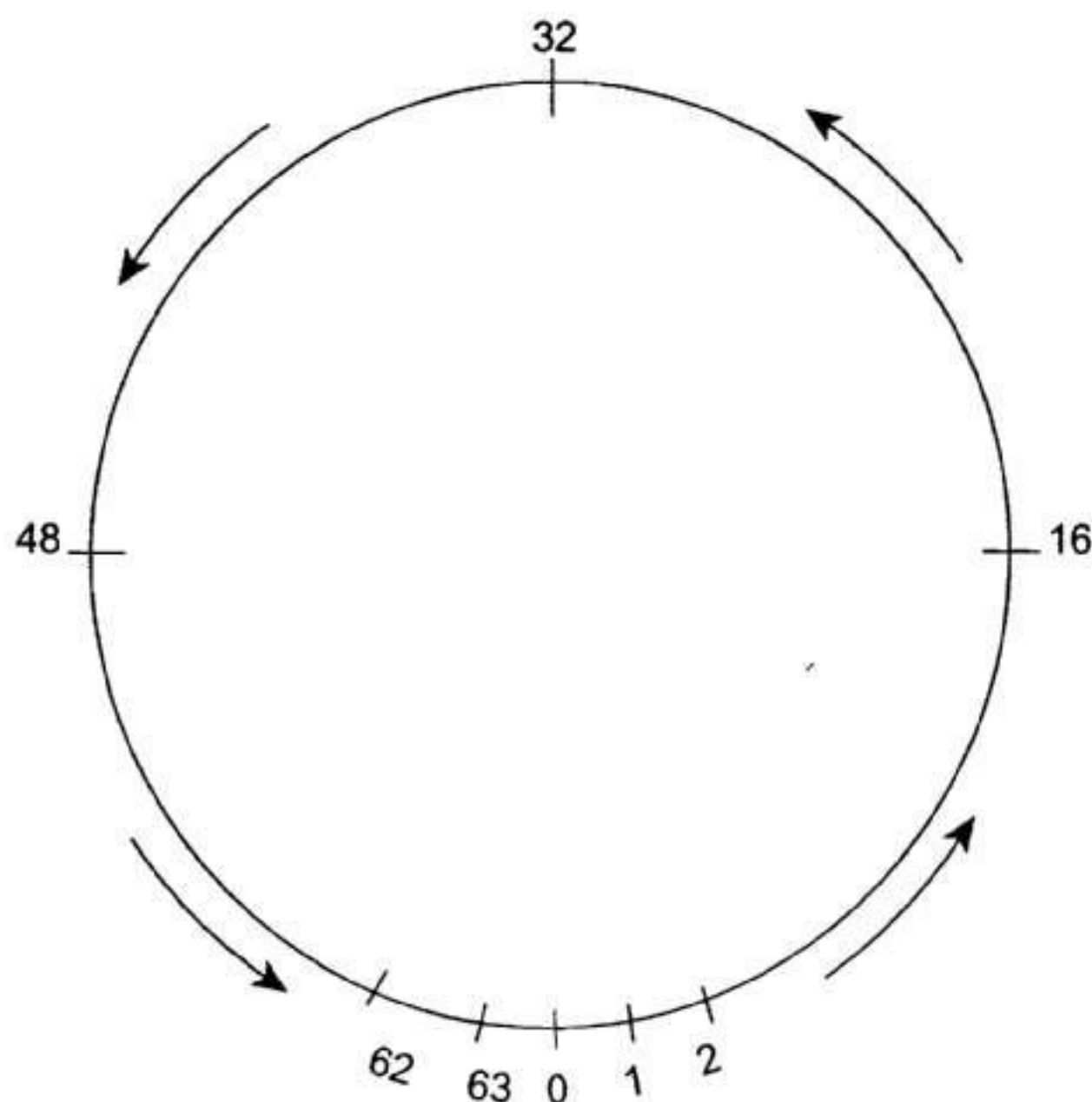


图 4-9 6 位循环地址空间

假设给定两个序列号 48 和 18,由规则 1 可得 48 较新:

$$48 > 18 \text{ 且 } (48 - 18) = 30, \text{ 且 } 30 < 32。$$

假设给定两个序列号 3 和 48,由规则 2 可得 3 较新:

$3 < 48$ 且 $(48-3) = 45$, 且 $45 > 32$ 。

假设给定两个序列号 3 和 18, 由规则 1 可得 18 较新:

$18 > 3$ 且 $(18-3) = 15$, 且 $15 < 32$ 。

可以看出规则强制序列号循环使用。

但是在运行状况不是很正常的互联网络上又会如何? 设想在一个互联网络使用 6 位序列号空间。现在其中的一台路由器决定离线, 当它在离线之时, 它突然发送了 3 个相同且序列号为 44 (101100) 的 LSA。不幸的是, 一个邻居也发生了故障——丢掉了几位数据, 丢失的数据位是第 2 个 LSA 和第 3 个 LSA 序列号中的 1 位。随即该邻居路由器向外泛洪扩散了这 3 个 LSA。结果造成 3 个 LSA 序列号各不相同。

44	(101100)
40	(101000)
8	(001000)

应用循环规则可得 44 比 40 更新, 40 比 8 更新, 8 又比 44 更新! 这个结论将使 3 个 LSA 都持续扩散下去, 数据库也不断地被最新的 LSA 更新, 直到数据库缓冲被塞满, CPU 超载为止, 最终整个互联网络崩溃。

这一连锁事件听上去好像有些牵强。然而它确实是真实的。现代互联网络的前驱 ARPANET 早期曾经使用过带有 6 位序列号空间的链路状态协议。在 1980 年 10 月 27 号, 两台路由器发生了上面所说的故障, 从而造成 ARPANET 的停顿。¹

(3) 棒棒糖形序列号空间

这个古怪的名称是由 Radia Perlman 博士提出的。²棒棒糖形序列号空间是线性序列号空间和循环序列号空间的综合; 如果你考虑一下, 你会发现棒棒糖形序列号空间有一个线性组件和一个圆形组件。圆形空间的缺点是不存在一个数小于其他所有的数。线性空间的缺点是不能循环使用序列号, 即序列号是有限的。

当路由器 A 重新启动时, 它将从小于其他所有数的 a 开始。邻居将会识别出这个数, 这时如果邻居数据库中还保留有上次序列号为 b 的 LSA, 那么它们会发送 b 给 A, 则 A 将跳至该序列号。在知道自己重启前使用的序列号之前, 路由器 A 可能会发送不止一个 LSA。因此, 在邻居通知 A 上次使用的序列号或带有上次序列号的 LSA 消失之前, 必须准备充足的启动数以便 A 不会用光所有序列号。

这些线性重启序列号形成了棒棒糖的棒。在这些数用完或邻居提供了使 A 跳转的序列号之后, A 进入循环数空间, 也就是棒棒糖的糖体部分。

在设计棒棒糖地址空间时使用了有符号数, 即 $-k < 0 < k$ 。从 $-k$ 到 1 的负数形成棒棒, 从 0 到 k 的正数形成循环空间。Perlman 的序列号规则如下。假设给出两个数 a 、 b 及一个序列号空间 n , 当且仅当:

- ① $a < 0$ 且 $a < b$, 或
- ② $a > 0, a < b$, 且 $(b-a) < n/2$ 或
- ③ $a > 0, b > 0, a > b$, 且 $(a-b) > n/2$ 。

认为 a 比 b 更新。

¹ E.C.Rose. “网络控制协议的脆弱性: 一个例子。” 计算机通信回顾, 1981 年 7 月。

² R.Perlman. “路由信息的容错广播。” 计算机网络, 第 7 卷, 1983 年 12 月, 395—405 页。

图 4-10 给出了棒棒糖形序列号空间的一个实现。使用 32 位有符号数 N 可以产生 2^{31} 个正数和 2^{31} 个负数。 $-N$ (-2^{31} 或 $0x80000000$) 和 $N-1$ ($2^{31}-1$ 或 $0x7FFFFFFF$) 没有被使用。路由器在线时将从 $-N+1$ ($0x80000001$) 开始使用序列号, 一直增加到 0, 这时将进入循环数空间。当序列号到达 $N-2$ ($0x7FFFFFFE$) 时, 序列号将返回到 0 (不使用 $N-1$)。

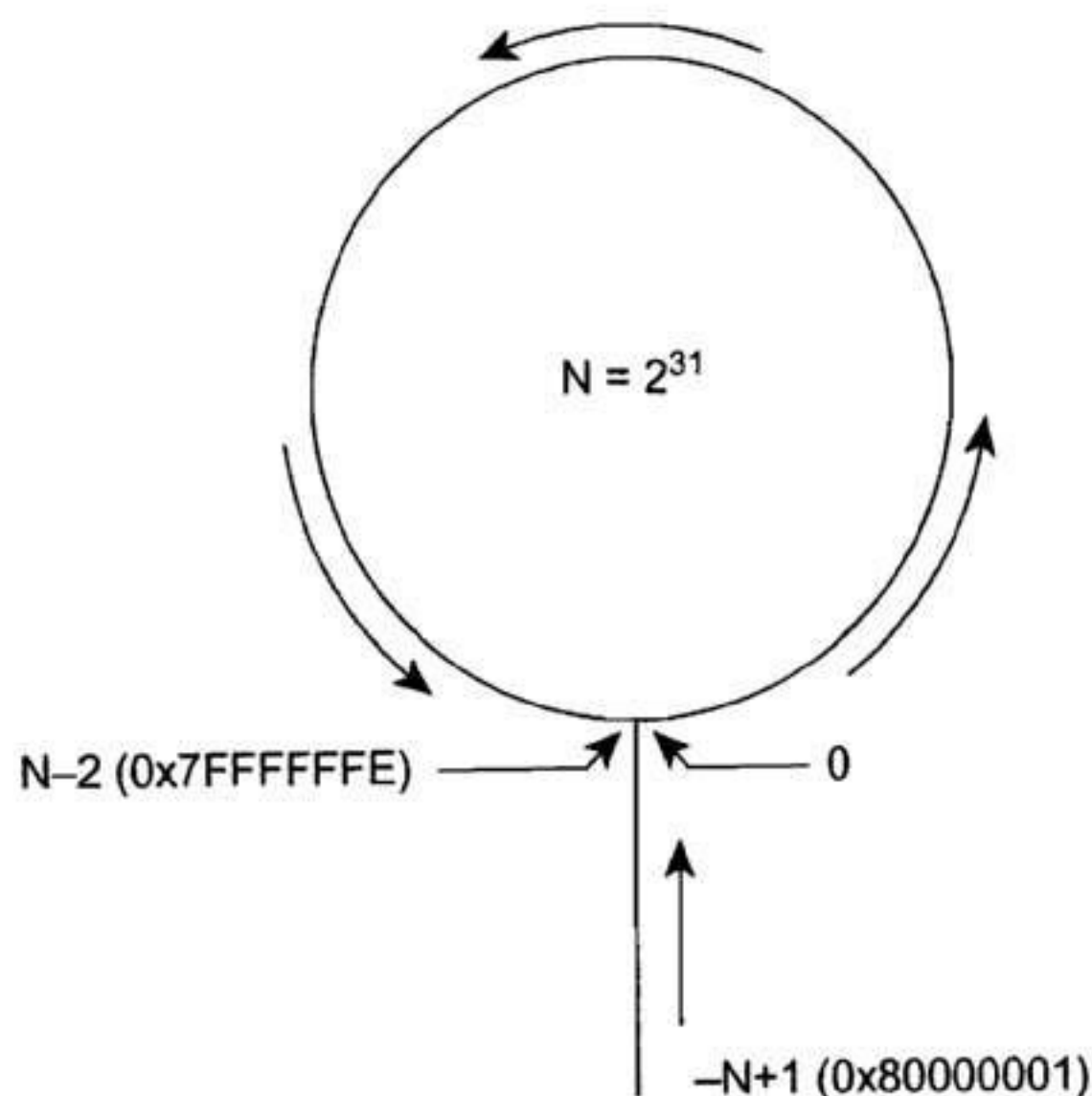


图 4-10 棒棒糖形序列号空间

假设路由器再次重新启动, 在启动之前最后一个 LSA 的序列号 $0x00005de3$ (循环序列号空间的一部分) 被发送给路由器。路由器重启后, 当它与邻居同步数据库时, 路由器发送序列号为 $0x80000001$ ($-N+1$) 的 LSA。邻居检查它的数据库发现该路由器重启前 LSA 的序列号为 $0x00005de3$, 这时邻居将把这个 LSA 发给重新启动的路由器, 实际上是说, “这是你离开时的位置。”接着重启路由器记录下这个序列号为正数的 LSA。如果下一时刻路由器需要发送新的 LSA, 它将使用序列号 $0x00005de6$ 。

棒棒糖形序列号空间是用在 OSPF 的最初版本 OSPFv1 (RFC1131) 中的。虽然使用有符号数是对线性数空间的一个改进, 但是发现棒棒糖形序列号空间的循环部分与纯粹的循环空间具有相同的弱点。OSPFv1 的部署一直处于试验阶段。当前使用的 OSPF 版本 OSPFv2 (见 RFC1247) 采用了线性和棒棒糖形序列号空间的最好的特性。OSPFv2 像棒棒糖形序列号数一样使用了从 $0x80000001$ 开始的有符号数空间。但是当序列号数变为正数时, 序列号空间持续保持线性直至到达最大数 $0x7FFFFFFF$ 为止。

2. 老化 (Aging)

LSA 的格式中将要包含一个通告的年龄字段。当 LSA 被建立时, 路由器将该字段设置为 0。随着报文的扩散, 每个路由器都会增加通告中的年龄。¹

老化过程为泛洪扩散过程增加了另一层可靠性, 该协议为互联网络定义了一个最大年龄差距 (MaxAgeDiff) 值。路由器可能接收到一个 LSA 的多个拷贝, 其中序列号相同, 年龄

¹ 当然, 另一个选型是从某个最大年龄开始, 然后递减。OSPF 是递增; IS-IS 是递减。

不同。如果年龄的差距小于 MaxAgeDiff, 那么认为是由于网络的正常时延造成了年龄的差异, 因此数据库原有的 LSA 继续保存, 新收到的 LSA (年龄更大) 不被扩散。如果年龄差距超过 MaxAgeDiff, 那么认为互连网络发生异常, 因为新被发送的 LSA 的序列号值没有增加。在这种情况下, 较新的 LSA 被记录下来, 并将报文扩散出去。典型的 MaxAgeDiff 值为 15min (OSPF 使用)。

若 LSA 驻留在数据库中, 则 LSA 的年龄会不断增加。如果链路状态记录的年龄增加到某个最大年龄值 (MaxAge)——由特定的路由选择协议定义——那么一个带有 MaxAge 值的 LSA 被泛洪扩散到所有邻居, 邻居随即从数据库中删除相关记录。

当 LSA 的年龄到达 MaxAge 时, 将被从所有的数据库中删除, 这需要有一种机制可以定期地确认 LSA 并且在达到最大年龄之前将它的计时器复位。链路状态计时器 (LSRefreshTime) 就是做此用途的;¹一旦计时器超时, 路由器将向所有邻居泛洪扩散新的 LSA, 收到的邻居会把有关记录的年龄设置为新接收到的年龄。OSPF 定义 MaxAge 为 1 小时, LSRefreshTime 为 30min。

3. 链路状态数据库

除了泛洪扩散 LSA 和发现邻居, 链路状态路由选择协议的第 3 个主要任务是建立链路状态数据库。链路状态或拓扑数据库把 LSA 作为一连串记录保存下来。虽然还保存了与 LSA 相关的序列号和年龄, 但这些变量主要用于管理泛洪扩散进程。对于最短路径的决策来说, 通告路由器的 ID、连接的网络和邻居路由器以及与网络 and 邻居相应的代价是很重要的。正如前面句子所隐含的, LSA 还包括两类通用信息:²

- 路由器链路信息——使用三元组 (路由器 ID、邻居 ID、代价) 通告路由器的邻居路由器, 这里的代价是指链路到邻居的代价;
- 末梢网络信息——使用三元组 (路由器 ID、网络 ID、代价) 通告路由器直接连接的末梢网络 (没有邻居的网络)。

最短路径优先 (SPF) 算法对路由器链路信息进行一次计算以建立到每台路由器的最短路径, 然后使用末梢网络信息向路由器添加网络。图 4-11 给出的互连网络包括路由器及路由器之间的链路, 为简单起见没有给出末梢网络。注意, 在几条链路两端的代价各不相同。因为代价与接口的出站方向有关。例如, 从 RB 到 RC 的链路代价为 1, 但是对相同的链路, 从 RC 到 RB 方向的代价却为 5。

对于图 4-11 的互连网络, 表 4-2 给出了一个通用的链路状态数据库, 在每台路由器中都保存了一份拷贝。当你阅读这个数据库时, 你将会发现数据库完整地描述了互连网络。现在通过运行 SPF 算法可以计算出一棵到达每台路由器的最短路径树。

表 4-2

关于图 4-11 中互连网络的拓扑数据库

路由器 ID	邻 居	代 价
RA	RB	2
RA	RD	4
RA	RE	4
RB	RA	2
RB	RC	1

¹ LSRefreshTime、MaxAge 和 MaxAgeDiff 是 OSPF 架构中的常量。

² 实际上, 信息的种类不止两种, 而且还包括多种链路状态报文类型。这些在关于特定的路由选择协议的章节会提到。

续表

路由器 ID	邻 居	代 价
RB	RE	10
RC	RB	5
RC	RF	2
RD	RA	4
RD	RE	3
RD	RG	5
RE	RA	5
RE	RB	2
RE	RD	3
RE	RF	2
RE	RG	1
RE	RH	8
RF	RC	2
RF	RE	2
RF	RH	4
RG	RD	5
RG	RE	1
RH	RE	8
RH	RF	6

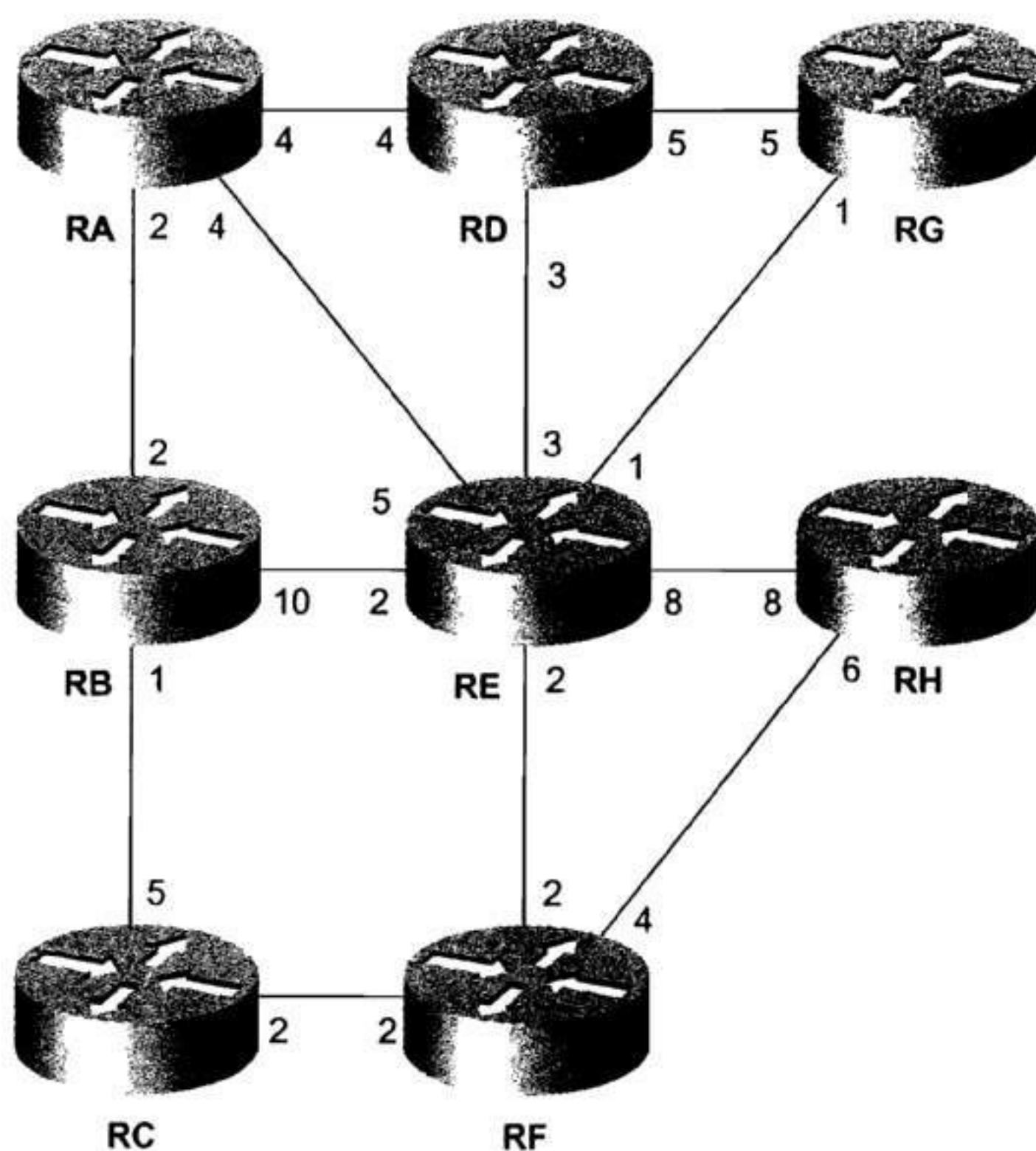


图 4-11 链路代价是按照流出接口的方向计算的，并且在一个网络中所有接口的代价没有必要完全相同

4.3.3 SPF 算法

在路由选择领域里，很不幸的是 Dijkstra 算法常常被认为就是最短路径优先算法。毕竟每个路由选择协议的目标都是计算最短路径。另一个不幸的是 Dijkstra 算法常常被描述的比

实际复杂得多。因为许多作者都使用集合论符号讨论它。最清晰的描述来自 E.W.Dijkstra 的原稿。这里将使用他的原话，并插入针对链路状态路由协议的解释：

构造一个树[a]，使 n 个节点之间的总长最小（树是一个在每两个节点之间仅有一条路径的图）。

在我们给出的构造过程中，分枝被分成 3 个集合：

- I. 被明确分配给构造中的树的分支（它们将在子树中）；
- II. 这个分支的隔壁分支被添加到集合 I；
- III. 剩余的分支（抛弃或不考虑）。

节点被分成 2 个集合：

- A. 被集合 I 中的分支连接的节点；
- B. 剩余的节点（集合 II 中有且仅有一个分支将指向这些节点中的每一个节点）。

下面我们开始构造树，首先选择任意一个节点作为集合 A 中仅有的成员，并将所有拿这个节点做端点的分支放入集合 II 中。开始集合 I 是空的。然后我们重复执行下面两步。

步骤 1： 集合 II 中最短的分支被移出并加入集合 I。结果，一个节点被从集合 B 传送到集合 A。

步骤 2： 考虑从这个节点（刚才被传送到集合 A 中的节点）通向集合 B 中节点的分支。如果构建中的分支长于集合 II 中相应的分支，那么分支被丢弃；否则，用它替代集合 II 中相应的分支，并且丢弃后者。

接着我们回到第 1 步并重复此过程直到集合 II 和 B 为空。集合 I 中的分支形成所要求的树。¹

配合路由器的算法，第一点注意的是，Dijkstra 描述了 3 个分枝集合：I、II 和 III。在路由器中，使用 3 个数据库表示 3 个集合：

- **树数据库**

树数据库用来表示集合 I。通过向数据库中添加分枝实现向最短路径树中添加链路（分枝）。当算法完成时，这个数据库将可以描述最短路径树。

- **候选对象数据库**

候选对象数据库对应集合 II。按照规定的顺序从链路状态数据库向该列表中复制链路，作为向树中添加的候选对象。

- **链路状态数据库**

按照前面的描述，这里保存所有链路，这个拓扑数据库对应集合 III。

Dijkstra 还指定了两个节点集合——A 和 B。这里的节点是路由器。这些路由器被明确地用路由器链路三元组（路由器 ID、邻居 ID、代价）中的邻居 ID 表示。集合 A 由树数据库中链路所连接的路由器组成。集合 B 是所有其他的路由器。由于整个要点是发现到每个路由器的最短路径，所以当算法接收时集合 B 应该为空。

下面是路由器中采用的 Dijkstra 算法版本：

步骤 1： 路由器初始化树数据库，将自己作为树的根。这表明路由器作为它自己的邻居，代价为 0。

步骤 2： 在链路状态数据库中，所有描述通向根路由器邻居链路的三元组被添加到候选

¹ E.W.Dijkstra. “关于连通图中两个问题的注释。” Numerische Mathematik. 卷 1, 1959 年, 269—271 页。

对象数据库中。

步骤 3: 计算从根到每条链路的代价, 候选对象数据库中代价最小的链路被移到树数据库中。如果两个或更多的链路离根的最短代价相同, 选择其中一条。

步骤 4: 检查添加到树数据库中的邻居 ID。除了邻居 ID 已在树数据库中的三元组之外, 链路状态数据库中描述路由器邻居的三元组被添加到候选对象数据库中。

步骤 5: 如果候选对象中还有剩余的表项, 回到第 3 步。如果候选数据库为空, 那么终止算法。在算法终止时, 在树数据库中, 一个单一的邻居 ID 表项将表示每台路由器, 并且最短路径树构造完毕。

表 4-3 总结了应用 Dijkstra 算法为图 4-11 中的网络构建最短路径树的过程和结果。路由器 RA 正在运行算法, 并使用表 4-2 中的链路状态数据库。图 4-12 给出了通过该算法为路由器 A 构造的最短路径树。在每台路由器完成自己最短路径树的计算之后, 它能够检查其他路由器的网络链路信息, 并且完成向树中添加末梢网络这种相对容易的任务。根据这些信息, 表项可以被制作为路由选择表。

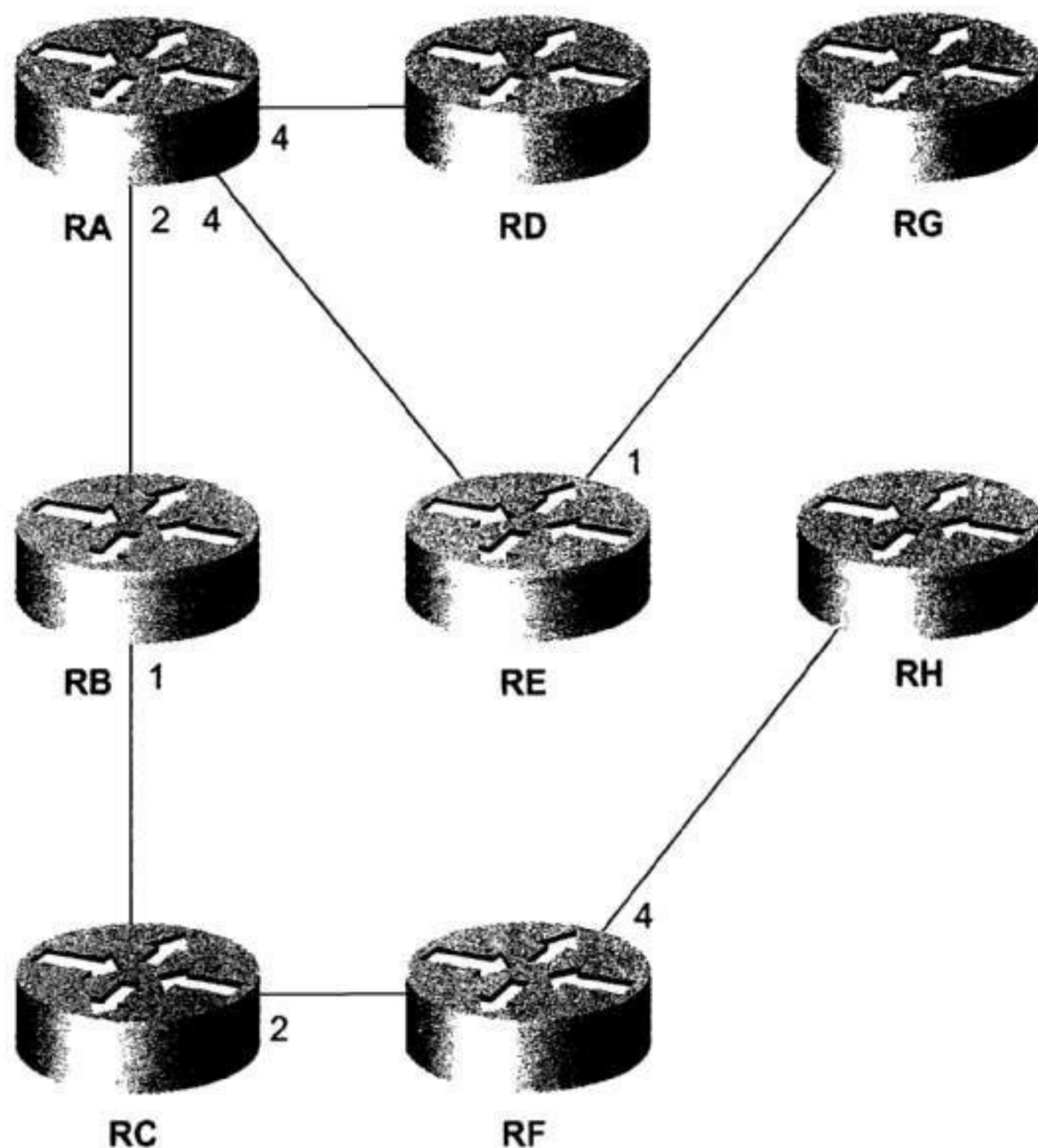


图 4-12 用表 4-3 中的算法得到的最短路径树

表 4-3

Dijkstra 算法应用到表 4-1 的数据库

候 选 对 象	到根的代价	树	描 述
		RA, RA, 0	路由器 A 把自己作为树的根
RA, RB, 2	2	RA, RA, 0	到 RA 所有邻居的链路被添加到候选对象列表
RA, RD, 4	4		
RA, RE, 4	4		
RA, RD, 4	4	RA, RA, 0 RA, RB, 2	(RA, RB, 2) 是候选列表中代价最小的链路, 所以被添加到树中。所有 RB 的邻居除了已在树中的都将被添加到候选列表。(RA, RE, 4) 到 RE 的代价比 (RB, RE, 10) 小, 所以后者被从候选列表中丢弃
RA, RE, 4	4		
RB, RC, 1	3		
RB, RE, 10			

续表

候 选 对 象	到根的代价	树	描 述
RA, RD, 4 RA, RE, 4 RC, RF, 2	4 4 5	RA, RA, 0 RA, RB, 2 RB, RC, 1	(RB, RC, 1) 是候选列表中代价最小的链路, 所以被添加进树中。所以 RC 的邻居除了已在树中的都将变为候选对象
RA, RE, 4 RC, RF, 2 RD, RE, 3 RD, RG, 5	4 5 7 9	RA, RA, 0 RA, RB, 2 RB, RC, 1 RA, RD, 4	(RA, RD, 4) 和 (RA, RE, 4) 离 RA 的代价都为 4; (RC, RF, 2) 代价为 5。 (RA, RD, 4) 被添加到树中, 它的邻居成为候选对象。在候选列表中有两条路径到 RE, 从 RA 出发 (RD, RE, 3) 因代价更高而被丢弃
RC, RF, 2 RD, RG, 5 RE, RF, 2 RE, RG, 1 RE, RH, 8	5 9 6 5 12	RA, RA, 0 RA, RB, 2 RB, RC, 1 RA, RD, 4 RA, RE, 4	(RF, RE, 1) 被添加到树中, 所有 RE 的不在树中的邻居都被添加进候选列表。到 RG 代价最高的链路被丢弃
RE, RF, 2 RE, RG, 1 RE, RH, 8 RF, RH, 4	6 5 12 9	RA, RA, 0 RA, RB, 2 RB, RC, 1 RA, RD, 4 RA, RE, 4 RC, RF, 2	(RC, RF, 2) 被添加进树中并且它的邻居被添加进候选列表。由于从 RA 出发 (RE, RG, 1) 代价相同 (5), 所以使用 (RE, RG, 1) 替代。到 RH 代价更高的路径被丢弃
RF, RH, 4		RA, RA, 0 RA, RB, 2 RB, RC, 1 RA, RD, 4 RA, RE, 4 RC, RF, 2 RE, RG, 1	(RE, RG, 1) 被添加到树中。RG 所有邻居都在树中, 所以没有对象被添加到候选列表中
		RA, RA, 0 RA, RB, 2 RB, RC, 1 RA, RD, 4 RA, RE, 4 RC, RF, 2 RE, RG, 1 RF, RH, 4	(RF, RH, 4) 是候选列表中代价最小的链路, 所以被添加到树中, 候选列表中不再有候选对象, 所以算法终止。最短路径树构造完毕

4.3.4 区域

一个区域是构成一个互联网络的路由器的一个子集。将互联网络划分为区域是针对链路状态协议的 3 个不利影响所采取的措施。

- 必要的数据库要求内存的数量比距离矢量协议更多。
- 复杂的算法要求 CPU 时间比距离矢量协议更多。
- 链路状态泛洪扩散报文对可用带宽带来了不利的影响, 特别是不稳定的互联网络。

目前设计的链路状态协议和使用该协议的路由器都能够减少这些影响, 但是却没有消除这些影响。在最后一节我们分析了在含有 8 台路由器的互联网络中, 链路状态数据看上去是什么样, SPF 算法是如何工作的。但要记住, 对于连接到 8 台路由器上的末梢网络, 并由此形成的 SPF 树的叶节点, 在这里没有对此进行考虑。

现在我们假设一个有 8000 台路由器的互联网络, 你就能理解对内存、CPU 和带宽影响的利害关系了。

正如图 4-13, 通过划分区域可以减小这些影响。当一个互连网络被划分为区域, 在一个区域内的路由器仅需要在本区域扩散 LSA, 因而只需要维护本区域的链路状态数据库。数据库越小, 意味着需要内存越少, 运行 SPF 算法需要的 CPU 周期也越少。如果拓扑改变频繁发生, 引起的扩散将被限制在不稳定的区域。

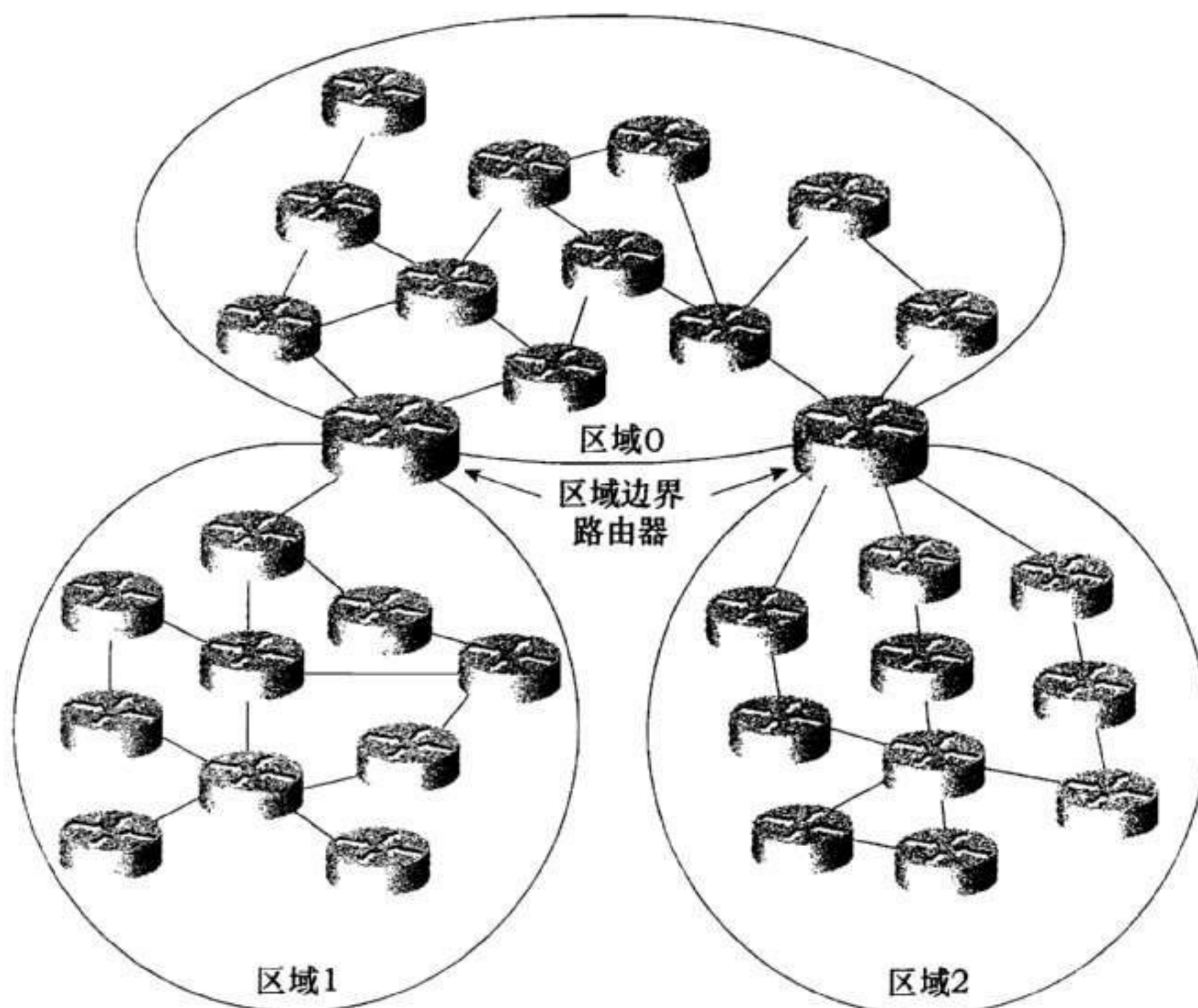


图 4-13 区域的使用减少了链路状态对系统资源的需求

区域边界路由器是连接两个区域的路由器, 它属于所连接的两个区域, 而且必须为每个区域维护各自的拓扑数据库。就像网络上的一个想往其他网络发送报文的主机一样, 它仅需要知道如何找到本地路由器, 在一个区域内想往其他区域发送报文的路由器仅需要知道怎样找到本地区域边界路由器。换句话说, 区域间路由器/区域内路由器之间的关系就如同主机/路由器之间的关系, 除了等级更高一些。

距离矢量协议, 如 RIP 和 IGRP, 不使用区域。假设这些协议不要任何手段, 而要把一个大型的互连网络看作一个单一的实体, 那么它就必须计算到每个网络的路由, 并且每 30s 或 90s 就需要广播这个巨大的路由选择表。很明显, 利用区域的链路状态协议实际上可以节省系统资源。

4.4 内部和外部网关协议

区域向互连网络体系结构中引入了一个新的层次, 在此基础上, 将一组区域组成一个更大的区域又向互连网络体系结构中引入了另一个新的层次。这些更高层的区域在 IP 网中叫自主系统, 在 ISO 模型中叫路由选择域。

一个自主系统被定义为在共同管理域下的一组运行相同路由选择协议的路由器。假设现

代互联网络的活动存在变移性,那么该定义的后半部分将不是非常准确。部门、分公司乃至整个公司常常会合并,随着它们的合并原来设计使用不同路由选择协议的互联网络也合并在一起。结果,现今许多互联网络使用多种不精确的等级将多种路由选择协议结合在一起,并处于共同的管理之下。所以自主系统的当前定义应该是在共同管理下的互联网络。

运行在一个自主系统内的路由选择协议称为内部网关协议(IGP)。在本章中,所有作为距离矢量和链路状态协议例子的都是 IGP。

在自主系统之间和路由选择域之间进行路由选择的协议称为外部网关协议(EGP)。IGP 发现网络之间的路径,而 EGP 发现自主系统之间的路径。下面这些都是 EGP:

- 边界网关协议(BGP)
- 外部网关协议(EGP)
- ISO 的域间路由选择协议(IDRP)

Novell 也把 EGP 功能合并到 NLSP 中,叫 3 层路由。

在给出这些定义的同时,还必须要提到的是,术语自主系统(Autonomous System)的通常用法并不是绝对的。各种标准文档、文献和人们都趋向于给这个术语多种含义。其结果是,理解听到或读到该术语的上下文是十分重要的。

本书在两种上下文中使用自主系统:

- 如本节开始定义的,自主系统可以指路由选择域。在这个上下文中,一个自主系统是一个有一个或多个 IGP 的系统,它完全自主于其他 IGP 系统。在这些自主系统之间使用 EGP。
- 自主系统也可以认为是一个过程域,或一个 IGP 进程,它自主于其他 IGP 进程。例如,使用 OSPF 的路由器可能被认为是 OSPF 自主系统。在关于 IGRP 和 EIGRP 的章节中也在这种上下文中使用自主系统。在这些自主系统之间使用了路由的再分配。

在本书中,上下文将指明在不同地方所讨论的自主系统是哪一种形式的自主系统。

4.5 静态或动态路由

在阅读完动态路由选择协议的细节之后,留下的印象一定是动态路由选择协议比静态路由选择协议好。动态路由选择协议的主要任务就是自动监测和适应互联网络中拓扑的变化,记住这一点很重要。但是,为实现自动化需要在带宽、队列空间、内存和处理时间上付出代价。

静态路由选择常见的缺陷是管理困难。这对于存在许多选择路由的中型和大型网络来说是真实的,但对于几乎没有选择路由的小型互联网络来说就不适用了。

在图 4-14 中,在较小的互联网络中很流行中心辐射式(hub-and-spoke)拓扑。如果到路由器的一辐发生中断,是否有另一条路由供动态路由选择协议选择?因此对于这种互联网络来说,十分适合使用静态路由选择协议。在中心路由器上为每个辐条上的路由器配置一条静态路由,而在辐条路由器上配置一条指向中心路由器的缺省路由,这样互联网络就可以工作了(缺省路由将在第 12 章介绍)。

在设计互联网络时,最简单的解决方法常常是最好的办法。在确定静态路由选择协议不能满足设计要求之后,可以选择动态路由选择协议。

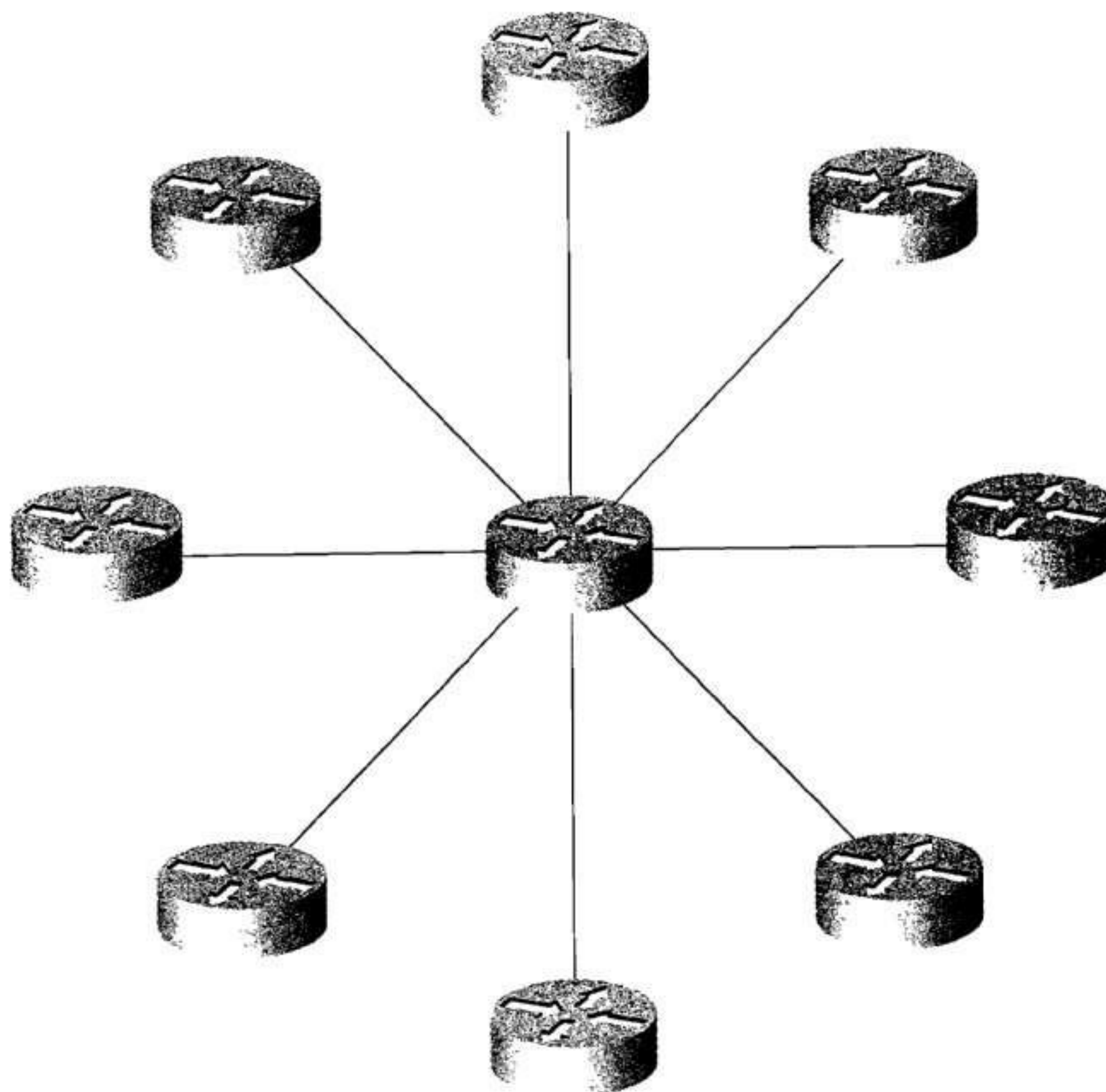


图 4-14 中心辐射式互联网络适合使用静态路由选择

4.6 展 望

既然动态路由选择协议的基础知识已经分析完了, 现在该是讨论具体的路由选择协议的时候了。下一章是 **RIP**, 最古老最简单的动态路由选择协议。

4.7 推荐读物

Perlman, R. *Interconnections: Bridges and Routers*. Reading, Massachusetts: Addison-Wesley; 1992.

这本书已经在第 1 章中提到过。如果你没读过这本书, 那么你应该读一下。

4.8 复 习 题

1. 什么是路由选择协议?
2. 路由选择算法执行的基本过程是什么?

3. 为什么路由选择协议使用度量?
4. 什么是收敛时间?
5. 什么是负载均衡? 给出负载均衡的4种不同类型。
6. 距离矢量路由选择协议是什么?
7. 指出距离矢量协议存在的几个问题。
8. 邻居是什么?
9. 路由失效计时器的作用是什么?
10. 解释简单水平分隔与毒性逆转水平分隔的区别?
11. 什么是计数到无穷大问题? 如何控制?
12. 什么是抑制计时器? 它们如何工作?
13. 距离矢量和链路状态协议的区别是什么?
14. 拓扑数据库的作用是什么?
15. 解释链路状态互连网络中收敛所包含的基本步骤。
16. 在链路状态协议中序列号为什么重要?
17. 在链路状态协议中老化服务的作用是什么?
18. 解释 SPF 算法是怎样工作的。
19. 区域对链路状态网络的好处是什么?
20. 什么是自主系统?
21. IGP 和 EGP 的区别是什么?

第二部分

内部路由选择协议

第 5 章 路由选择信息协议 (RIP)

第 6 章 内部网关路由选择协议 (IGRP)

第 7 章 路由选择信息协议——第 2 版 (RIPv2)

第 8 章 增强型内部网关路由选择协议 (EIGRP)

第 9 章 开放最短路径优先协议 (OSPF)

第 10 章 集成 IS-IS 协议

第 5 章

路由选择信息协议 (RIP)

本章包括以下主题：

- RIP 的操作
 - RIP 协议的计时器和稳定性
 - RIP 协议的消息格式
 - RIP 协议的请求消息类型
 - 有类别路由选择
- RIP 协议的配置
 - 案例研究：一个基本的 RIP 配置
 - 案例研究：被动接口
 - 案例研究：配置单播更新
 - 案例研究：不连续的子网
 - 案例研究：掌握 RIP 的度量
- RIP 协议的故障排除

RIP 协议作为最早的距离矢量型 IP 路由选择协议依然被广泛地使用着，当前存在着两个版本。本章介绍的是 RIP 协议的第 1 版，第 7 章“路由选择信息协议第 2 版（RIPv2）”中涵盖了 RIP 协议的第 2 版的内容，而后者对第 1 版的功能作了部分增强。版本 1 和版本 2 最主要的区别是，RIPv1 是有类别路由选择协议，而 RIPv2 是无类别路由选择协议。本章介绍有类别路由，第 7 章介绍无类别路由。

距离矢量协议基于 Bellman¹、Ford 和 Fulkerson²开发的路由算法，早在 1969 年的一些早期网络(ARPANET 和 CYCLADES)中就已经实现了。在上世纪 70 年代中期，Xerox

¹ R. E. Bellman. *Dynamic Programming*. Princeton, New Jersey: Princeton University Press; 1957.

² L. R. Ford Jr. and D. R. Fulkerson. *Flows in Networks*. Princeton, New Jersey: Princeton University Press; 1962.

公司开发了一个叫做 PARC 通用协议¹, 或称为 PUP 协议, 运行在它的可以说是现代以太网前身的 3Mbit/s 带宽的试验网络上。PUP 使用网关信息协议(GWINFO)来选路, 并发展成为 Xerox 网络系统(XNS)协议族。同时, GWINFO 协议发展成为 XNS 路由选择信息协议(XNS RIP)。XNS RIP 也随之成为一些常用的路由选择协议的前身, 这些协议如 Novell 的 IPX RIP、AppleTalk 的 RTMP(路由选择表维护协议), 当然还有 IP RIP。

1982 年, 伯克利发布的 UNIX 4.2BSD 版中, 通过一个称为“routed”的驻留进程实现了 RIP 协议; 很多后来的 UNIX 版本都是基于流行的 UNIX 4.2BSD 版本的, 并且也都通过一个称为“routed”或“gated”²的进程支持 RIP 协议。有意思的是, 直到 1988 年才发布了一个 RIP 协议的标准, 那时 RIP 已经被广泛应用了。这个标准就是 RFC 1058, 由 Charles Hedrick 撰写, 是 RIP 协议第 1 版的比较正式的标准。

由于人们阅读的文献不同, 从而对 RIP 协议褒贬不一。RIP 协议虽然没有后来一些路由选择协议功能强大, 但它简单易用, 已有广泛的应用, 这意味着 RIP 协议在实际网络的实施中碰到的兼容性问题会比较少。在小型网络数据互联的设计中, RIP 协议还是相当单一的设计。在这些限定的条件下, 尤其是许多 UNIX 环境下, RIP 协议依然是一个受欢迎的路由选择协议。

5.1 RIP 的操作

RIP 协议的处理是通过 UDP 520 端口来操作的。所有的 RIP 消息都被封装在 UDP 数据报文中, 其中数据报文的源和目的端口字段被设置为 520。RIP 定义了两种消息类型: 请求消息(Request messages)和响应消息(Response messages)。请求消息用来向邻居路由器发送一个更新(Update), 响应消息用来传送路由更新。RIP 的度量是基于“跳”数(hop count)的, 1 跳表示的是与发出通告的路由器相直连的网络, 16 跳表示网络不可到达。

开始时, RIP 从每个启动 RIP 协议的接口广播出带有请求消息的数据包。接着, RIP 程序进入一个循环状态, 不断地侦听来自其他的路由器的 RIP 请求或响应消息, 而接收请求的邻居路由器则回送包含它们的路由选择表的响应消息。

当发出请求的路由器收到响应消息时, 它将开始处理附加在响应消息中的路由更新信息。如果路由更新中的路由条目是新的, 路由器则将新的路由连同通告路由器的地址一起加入到自己的路由选择表中, 这里通告路由器的地址可以从更新数据包的源地址字段读出。如果网络路由已经在路由选择表中存在, 那么只有在新的路由拥有更小的跳数时才能替换原来存在的路由条目。如果路由更新通告的跳数大于路由选择表已记录的跳数, 并且更新来自于已记录条目的下一跳路由器, 那么该路由将在一个指定的抑制时间段(Holddown period)内被标记为不可到达。如果在抑制时间超时后, 同一个邻居路由器仍然通告这个有较大跳数的路由, 路由器则接受该路由新的度量值。³

5.1.1 RIP 的计时器和稳定性

路由器启动后, 平均每隔 30s 从每个启动 RIP 协议的接口不断地发送出响应消息。除了

¹ Palo Alto Research Center.

² 读作“route-dee”和“gate-dee”。

³ Holddowns 用于 Cisco IOS, 但它不是 RFC 1058 指定的稳定性特性之一。

被水平分隔法则抑制的路由条目之外，响应消息（或称为更新消息）包含了路由器的整个路由选择表。这个周期性的更新由更新计时器(Update Timer)进行初始化，并且包含一个随机变量用来防止表的同步。¹结果，一个典型的 RIP 处理单个更新的时间大约是 25~35s。RIP_JITTER 是 Cisco IOS 中专有的一个随机变量，它缩短到一般更新时间的 15% (即 4.5s)。因此，Cisco 路由器的更新时间在 25.5~30s 之间变化(图 5-1)。路由更新的目的地址是到所有主机的广播地址 255.255.255.255。²

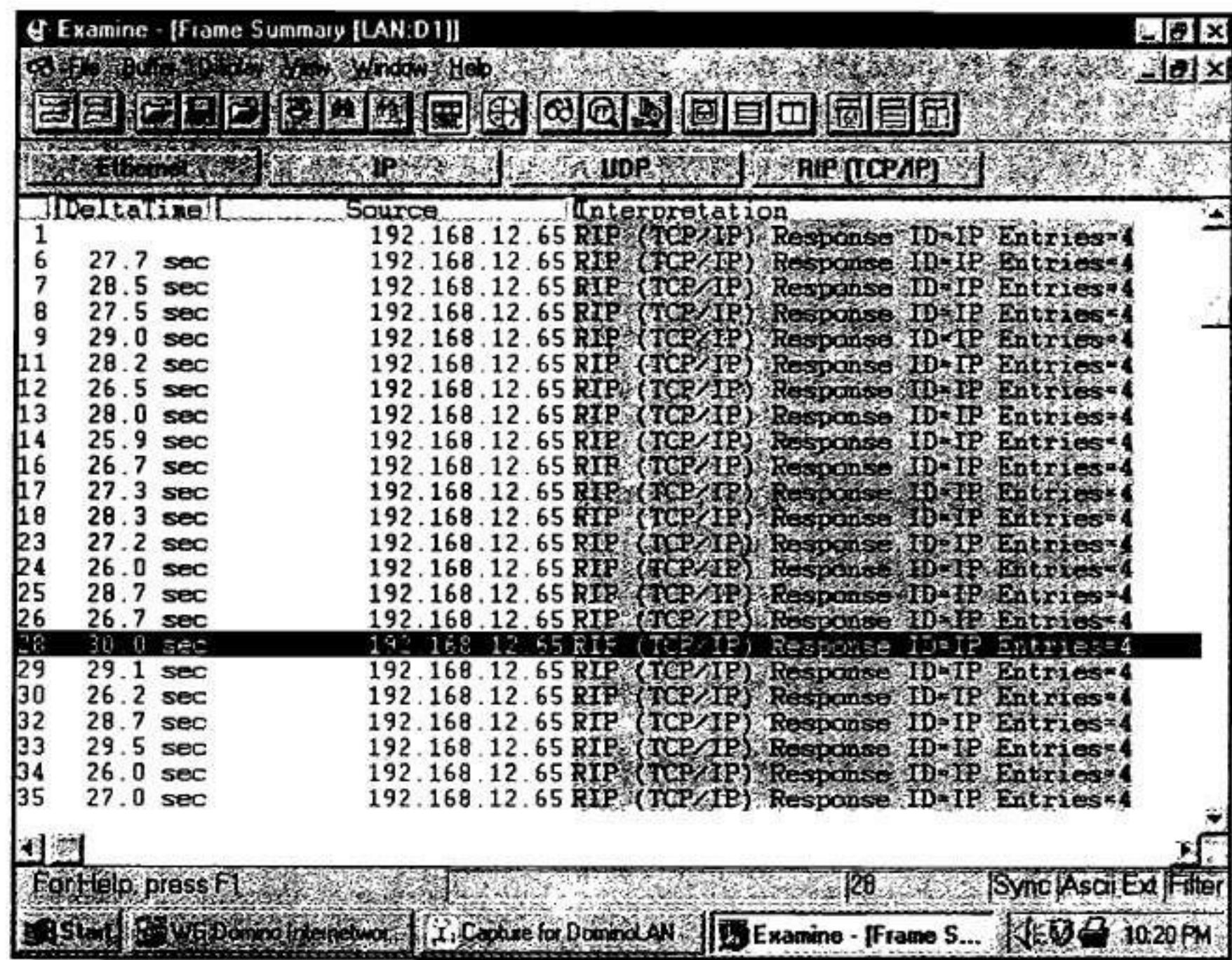


图 5-1 RIP 协议在每次重置计时器时，会增加一个小随机变量到更新计时器以避免路由选择表的同步。在 Cisco 的路由器中，RIP 的更新时间在 25.5~30s 之间变化，从图中可以看出这些更新的增量时间

RIP 也使用一些其他的计时器。回忆一下第 4 章“动态路由选择协议”，距离矢量协议用到的无效计时器(Invalidation Timer)用来限制停留在路由选择表中的路由未被更新的时间。RIP 称这个计时器为限时计时器(Expiration Timer)或超时计时器(Timeout Timer)。在 Cisco IOS 中，称为无效计时器(Invalid Timer)。无论什么时候，当有一条新的路由建立成功后，超时计时器就会被初始化为 180s，而每当接收到这条路由的更新报文时，超时计时器又将被重置成计时器的初始化值，即 180s。如果一条路由的更新在 180s(6 个更新周期)内还没有收到，那么这条路由的跳数将变成 16，也就是标记为不可到达的路由。

另一种计时器，称为垃圾收集(Garbage Collection)或刷新计时器(flush timer)，它们所设置的时间长度一般比限时计时器的时间长 240~60s。³如果垃圾收集计时器也超时了，则该路由将被通告为一条度量值为不可到达的路由，同时从路由选择表中删除该路由项。图 5-2 显示了路由选择表中有一条被标记为不可到达的路由，但还没有被刷新掉。

¹ 路由选择表的同步在第 4 章中讨论。
² RIP 的一些实现方式也可以只在广播型介质网络上广播，而在点到点的链路上直接发送给对端直连的邻居路由器。Cisco 路由器中，如果不改变配置成其他方式的话，RIP 更新将在任何类型的链路上广播。
³ Cisco 路由器使用 60s 的垃圾收集计时器，虽然 RFC 1058 规定为 120s。


```

Mayberry#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
       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

Gateway of last resort is not set

    10.0.0.0 255.255.0.0 is subnetted, 4 subnets
C      10.2.0.0 is directly connected, Serial0
R      10.3.0.0 255.255.0.0 is possibly down,
        routing via 10.1.1.1, Ethernet0
C      10.1.0.0 is directly connected, Ethernet0
R      10.4.0.0 [120/1] via 10.2.2.2, 00:00:00, Serial0
Mayberry#

```

图 5-2 这台路由器经过 6 个更新周期的时间还没有收到关于子网 10.3.0.0 的更新。

因而这条路由被标记为不可到达，但还没有被从路由选择表中刷新掉

第 3 个计时器是抑制计时器(Holddown Timer)。虽然 RFC 1058 没有关于 Holddowns 的介绍，但在 Cisco 的路由器中运行的 RIP 协议使用了它们。如果一条路由更新的跳数大于路由选择表已记录的该路由的跳数，那么将会引起该路由进入长达 180s（即 6 个路由更新周期）的抑制状态阶段。

这 4 个计时器可以通过下面的指令来操作：

```
timers basic update invalid holddown flush
```

这个命令适用于 RIP 协议整个进程的运行处理。如果一个路由器的计时被改变了，那么这个 RIP 域中的所有路由器的计时都必须改变。因此，如果没有特别的原因和慎重考虑，不应该改变这些计时器的缺省值。

RIP 使用带毒性逆转(Poison Reverse)的水平分隔(Split Horizon)和触发更新(Triggered updates)。不像普通的定期的更新，触发更新在只要有路由的度量值发生改变时就会产生，而且触发更新不会引起接收路由器重置它们的更新计时器；因为如果这么做的话，网络拓扑的改变会导致很多路由器在同一时间重置，从而引起定期的路由更新变得同步。为了避免拓扑改变后造成触发更新“风暴”，还需要使用另外一个计时器。当一个触发更新传播时，这个计时器被随机的设置为 1~5s 之间的数值；在这个计时器计时超时前不能发送并发的触发更新。

一些主机可以在“静”模式下使用 RIP。这些所谓的“静”主机不产生 RIP 的更新报文，而只侦听 RIP 的更新消息，从而更新它们自己的路由选择表。举一个例子，在一台 UNIX 主机上可以使用带“-q”选项的“routed”启动“静”模式下的 RIP。

5.1.2 RIP 消息格式 (RIP Message Format)

RIP 的消息格式如图 5-3 所示。每条消息包含一个命令标识(Command)、一个版本号 (Version Number)和路由条目(最大 25 条)。每个路由条目包括地址族标识(Address Family Identifier)、路由可达的 IP 地址和路由的跳数。如果某台路由器必须发送大于 25 条路由的更

新消息, 那么必须产生多条 RIP 消息。注意, RIP 消息的开始部分(头部)占用 4 个 8bit 字节, 而每个路由条目占用 20 个 8bit 字节。因此, RIP 消息的大小最大为 $4+(25\times 20)=504$ 个 8bit 字节, 再加上 8 个字节的 UDP 头部, RIP 数据报文的大小 (不含 IP 包的头部) 最大可达 512 个 8bit 字节。

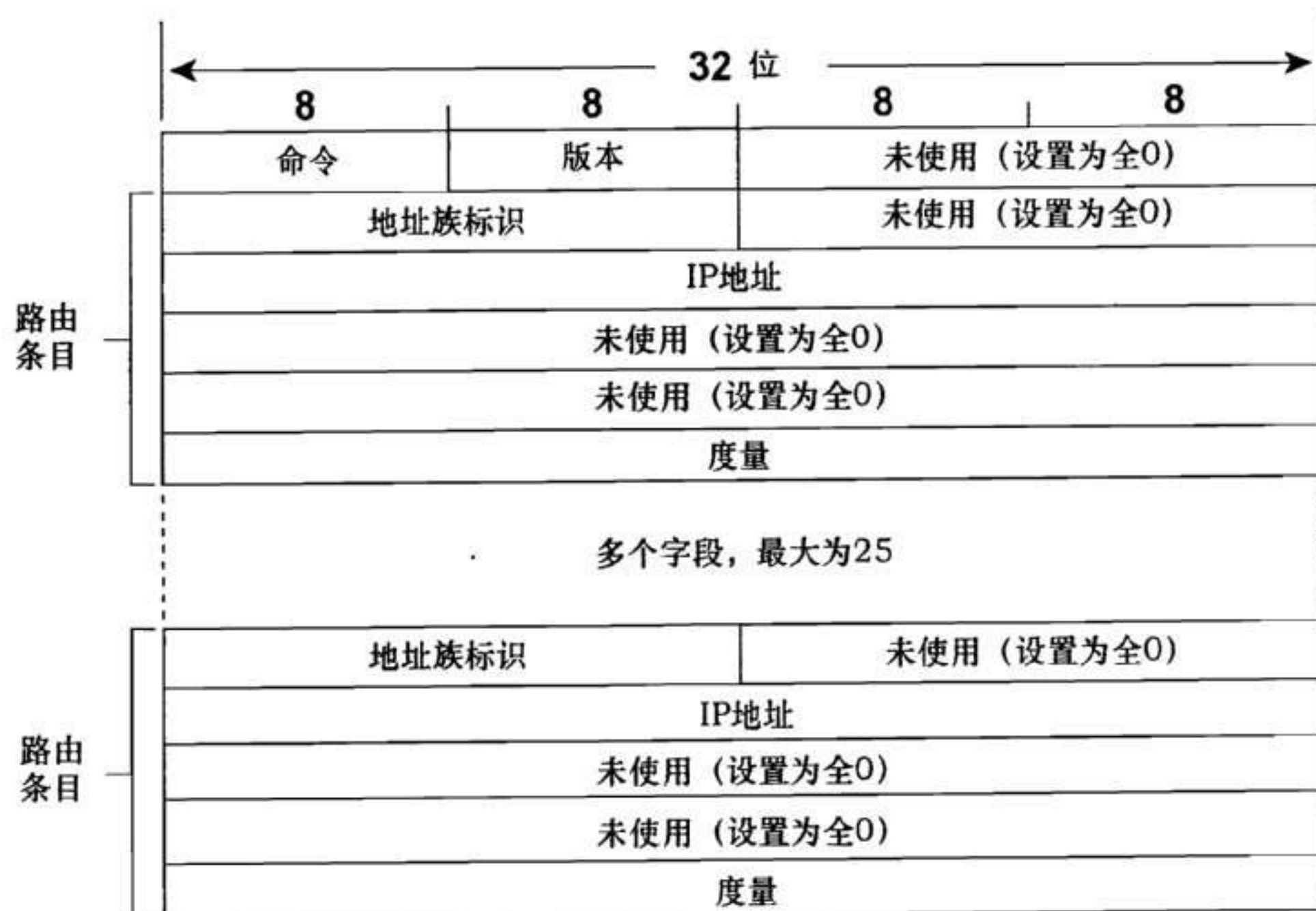


图 5-3 RIP 的消息格式

- **命令 (Command)** ——只取值 1 或 2, 1 表示该消息是请求消息, 2 表示该消息是响应消息。其他的取值都不被使用或保留用作私有用途。
- **版本号 (Version)** ——对于 RIPv1, 该字段的值设置为 1。
- **地址族标识 (Address Family Identifier, AFI)** ——对于 IP 该项设置为 2。只有一个例外情况, 该消息是路由器 (或主机) 整个路由选择表的请求, 这将在后面的章节讨论。
- **IP 地址 (IP Address)** ——路由的目的地址。这一项可以是主网络地址、子网地址或主机路由地址。在“有类别路由查找”一节, 将说明如何区分这 3 种类型的路由。
- **度量 (Metric)** ——正如前面的章节所说, Metric 在 RIP 里面就是指跳数, 该字段的取值范围在 1~16 之间。

图 5-4 显示了用协议分析仪观察到的一个 RIP 消息的解码。

由于一些历史的影响造成 RIP 消息的格式不尽合理, 消息格式中没有使用的 bit 空间远远大于所使用的。这些影响从 RIP 最初发展到后来就都存在, RIP 最初来自于 XNS 协议, 开发人员试图使它适合更广泛的地址族, 后来还有 BSD 的影响, 它所使用的套接字 (socket) 地址需要 RIP 的字段是增大到 32 位字的边界长度。

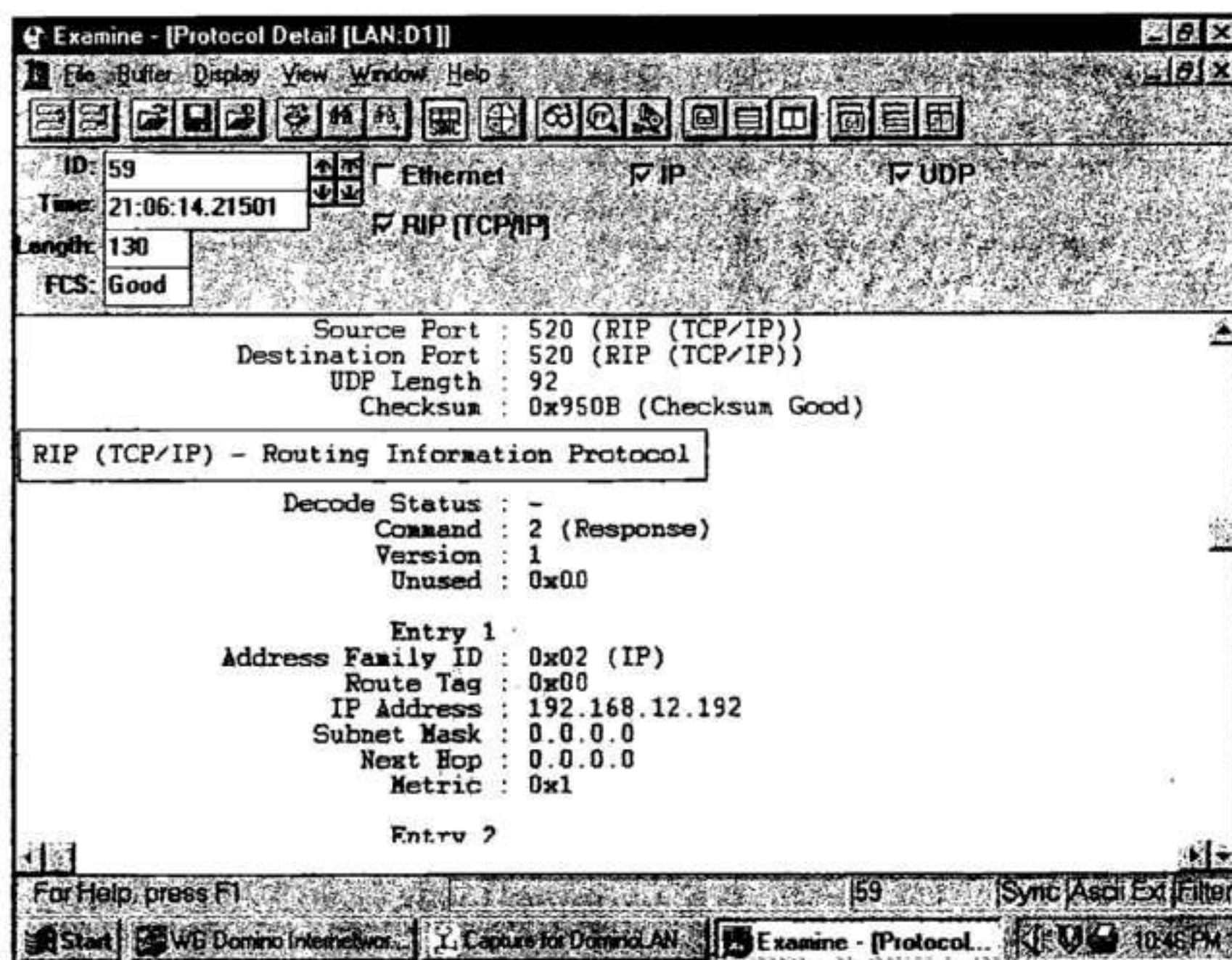


图 5-4 从协议分析仪可以看出, RIPv1 不使用子网掩码(Subnet Mask)和下一跳(Next Hop)字段。

这些字段在 RIPv2 中使用, 将在第 7 章中讲述

5.1.3 请求消息类型 (Request Message Type)

RIP 请求消息可以请求整个路由选择表信息, 也可以仅请求某些具体路由的信息。就前一种情况而言, 请求消息含有一个地址族标识字段为 0 (地址为 0.0.0.0) 度量值为 16 的单条路由, 接收到这个请求的设备将通过单播方式向发出请求的地址回送它的整个路由选择表, 并遵循一些规则如水平分隔 (Split Horizon) 和边界汇总 (Boundary Summarization, 在本章的后面一节“有类别路由选择: 在边界路由器上的路由汇总”中讲述)。

一些诊断测试程序可能需要知道某个或某些具体路由的信息。这样的话, 请求消息可以附加上相关被关注的地址的路由条目一起发送。接收到这个请求的设备将根据请求消息逐个处理这些条目, 构成一个响应消息。如果该设备的路由选择表中已有请求消息中地址相对应的路由条目, 则把它自己路由条目的度量值填入 metric 字段。如果没有, metric 字段就被设置为 16。在不考虑水平分隔或边界汇总的情况下, 响应将正确地告诉这台路由器了解的信息。

前面已经提到, 主机可以在“静”模式下运行 RIP。这种方法允许网络中的主机不需要发送无用的 RIP 响应消息, 也可以通过侦听来自路由器的 RIP 更新来保证自己的路由选择表保持最新。但是, 网络诊断程序可能需要检查这些“静”主机的路由选择表, 因此, RFC 1508 指出: 如果一个“静”主机接收到一个来自于 UDP 端口的请求消息, 而不是来自于标准的 RIP 520 端口, 那么主机就必须发送一个响应。

5.1.4 有类别路由选择(Classful Routing)

图 5-5 中的路由选择表包含了始发于 RIP 的路由, 这可以从每个路由条目的左边的关键

字识别出来。这些路由条目的权值由方括号中的元组来表示,与第3章“静态路由”中讨论的一样,第一个数字表示管理距离 (Administrative Distance),第二个表示度量值 (Metric)。从图中很容易看出, RIP 的管理距离为 120,如前所述, RIP 的度量是基于跳数的。因此,网络 10.8.0.0 通过 E0 或 S1 需要 2 跳可到达。如果到达同一个目的网络有多条跳数相等的路由,那么 RIP 进行等价路径代价的负载均衡。图 5-5 中的路由选择表就包含了多条等价路径代价的路由条目。

```
MtPilate#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
       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

Gateway of last resort is not set

  10.0.0.0 255.255.0.0 is subnetted, 9 subnets
R       10.10.0.0  [120/3] via 10.5.5.1, 00:00:20, Serial1
           [120/3] via 10.1.1.1, 00:00:21, Ethernet0
R       10.11.0.0  [120/3] via 10.5.5.1, 00:00:21, Serial1
           [120/3] via 10.1.1.1, 00:00:21, Ethernet0
R       10.8.0.0   [120/2] via 10.1.1.1, 00:00:21, Ethernet0
           [120/2] via 10.5.5.1, 00:00:21, Serial1
R       10.9.0.0   [120/2] via 10.5.5.1, 00:00:21, Serial1
           [120/2] via 10.1.1.1, 00:00:21, Ethernet0
R       10.3.0.0   [120/1] via 10.1.1.1, 00:00:21, Ethernet0
           [120/1] via 10.5.5.1, 00:00:21, Serial1
C       10.1.0.0 is directly connected, Ethernet0
R       10.6.0.0   [120/1] via 10.1.1.1, 00:00:21, Ethernet0
           [120/1] via 10.5.5.1, 00:00:22, Serial1
R       10.7.0.0   [120/2] via 10.1.1.1, 00:00:22, Ethernet0
           [120/2] via 10.5.5.1, 00:00:22, Serial1
C       10.5.0.0 is directly connected, Serial1
  172.25.0.0 255.255.255.0 is subnetted, 3 subnets
R       172.25.153.0 [120/1] via 172.25.15.2, 00:00:03, Serial0
R       172.25.131.0 [120/1] via 172.25.15.2, 00:00:03, Serial0
C       172.25.15.0 is directly connected, Serial0
```

图 5-5 这个路由选择表包含了主网络 10.0.0.0 和 172.25.0.0 的子网,所有这些非直连的子网都是从 RIP 协议学习得到的

当一个数据包进入宣告 RIP 的路由器后,路由器将执行路由选择表的查询,逐步排除数据包的路由选择范围,直到只剩下一条惟一的路径。路由器首先读出目的地址的网络部分(基于有类别路由选择协议的主网络号),察看这个网络部分在路由选择表中是否有其匹配的条目。根据有类别路由选择表查询规则的第一步,读出基于 A 类、B 类或 C 类主网分类的主网络号。如果没有匹配的主网络,这个数据包就被丢弃,同时发出一个 ICMP 目的不可达的信息给发出该数据包的源。如果存在匹配该数据包网络部分的主网络,那么路由选择表中会列出匹配这个主网络的子网,并进一步在这些子网中进行查询,这时,如果能找到一个匹配的子网条目,那么该数据包将被路由器转发,否则,该数据包将被丢弃并发出一个 ICMP 目的不可达的信息。

1. 有类别路由选择: 直连的子网络

有类别路由查询可以用下面 3 个例子来表述(参照图 5-5):

(1) 假设有一个目的地址为 192.168.35.3 的数据包进入这台路由器,由于该路由器在路

由选择表中没有发现和网络 192.168.35.0 匹配的条目, 因此这个数据包将被丢弃。

(2) 假设有一个目的地址为 172.25.33.89 数据包进入这台路由器, 在路由选择表中有一个和 B 类网络 172.25.0.0/24 匹配的条目, 那么进一步检查路由选择表中列出的网络 172.25.0.0/24 的子网条目, 显然没有和网络 172.25.33.0 匹配的子网条目, 因此这个数据包也被丢弃。

(3) 最后一个例子, 假设要到达地址 172.25.153.220 的数据包进入这台路由器, 这时, 路由选择表中有和网络 172.25.0.0/24 匹配的条目, 也进一步检查到有和子网 172.25.153.0 匹配的条目, 因此, 这个数据包将被转发到下一跳地址 172.25.15.2。

另外, 请注意观察图 5-3 中所显示的一个情况, RIP 协议的消息中并没有随同路由条目一起通告子网掩码, 从而路由器中也没有和单独的子网相关联的掩码。因此, 一台路由器的转发数据库如同图 5-5 中所显示的那样, 当它收到了一个目的地址为 172.25.131.23 的数据包, 即使这个地址是被完全子网化的, 路由器也没有确切的方法来识别子网网络位的结束位置和主机位的开始位置。

路由器惟一可以借助的就是, 假定在整个互联网络中, 对于同一个主网络地址使用相同的掩码, 这个掩码就是配置在与网络 172.25.0.0 相连的某一个路由器接口之上的。对于从目的地址的子网部分得出的主网络 172.25.0.0, 它将使用自己的掩码。正如本章所有图示中所显示的路由选择表那样, 如果一个网络是和路由器直连的, 那么路由器将在路由选择表中作为一个标题条目列出该网络 and 该网络所连接的接口的子网掩码, 然后列出它所知道的关于这个网络的所有子网。如果一个网络和路由器不是直接相连的, 那么路由选择表将仅仅列出这个网络的主网络而不列出与它相关联的掩码。

因为在有类别路由选择协议进行路由选择的情况下, 数据包的目的地址是通过在路由器接口本地配置的子网掩码来识别的, 所以在同一个主类别网络范围中的所有子网掩码应该是一致的。

2. 有类别路由选择: 在边界路由器上的路由汇总

在前面的讨论中产生了这样一个问题, 就是当一个网络没有和路由器的任何接口相连接时, RIP 协议应该怎么识别一个主类别网络的子网呢? 如果没有一个接口和该目的地址对应的 A 类、B 类或 C 类主网络相关联, 那么路由器将没有办法正确地识别出所使用的子网掩码, 也没有办法正确地标识该子网。

解决方法很简单: 如果路由器没有和某个目的网络有直接连接, 那么该路由器仅仅需要有一条简单的路由指向一个直接相连的路由器。

图 5-6 显示了一个处于两个主网络边界上的路由器, 这两个主网络是 A 类网络 10.0.0.0 和 C 类网络 192.168.115.0。这个“边界”路由器不会把其中一个主网络子网的具体信息发送给另一个主网。正如所显示的那样, 该路由器执行了自动路由汇总, 或称为子网屏蔽 (Subnet Hiding), 仅仅会把地址 10.0.0.0 通告给网络 192.168.115.0, 同样的会把地址 192.168.115.0 通告给网络 10.0.0.0。

这种方式下, 在网络 192.168.115.0 内的路由器的路由选择表中, 只包含一个单独的路由条目用来引导把到达目的网络 10.0.0.0 的数据包转发到边界路由器, 而边界路由器则具有和网络 10.0.0.0 直连的接口, 因此它具有一个带子网掩码的子网来为在网络 10.0.0.0 “云” 内的数据包来选路。图 5-7 显示了一个在网络 192.168.115.0 中的路由器的路由选择表, 在路由选

择表中可以看到有一条网络 10.0.0.0 的路由条目，它看起来像是一条单独的、没有子网掩码的路由条目。

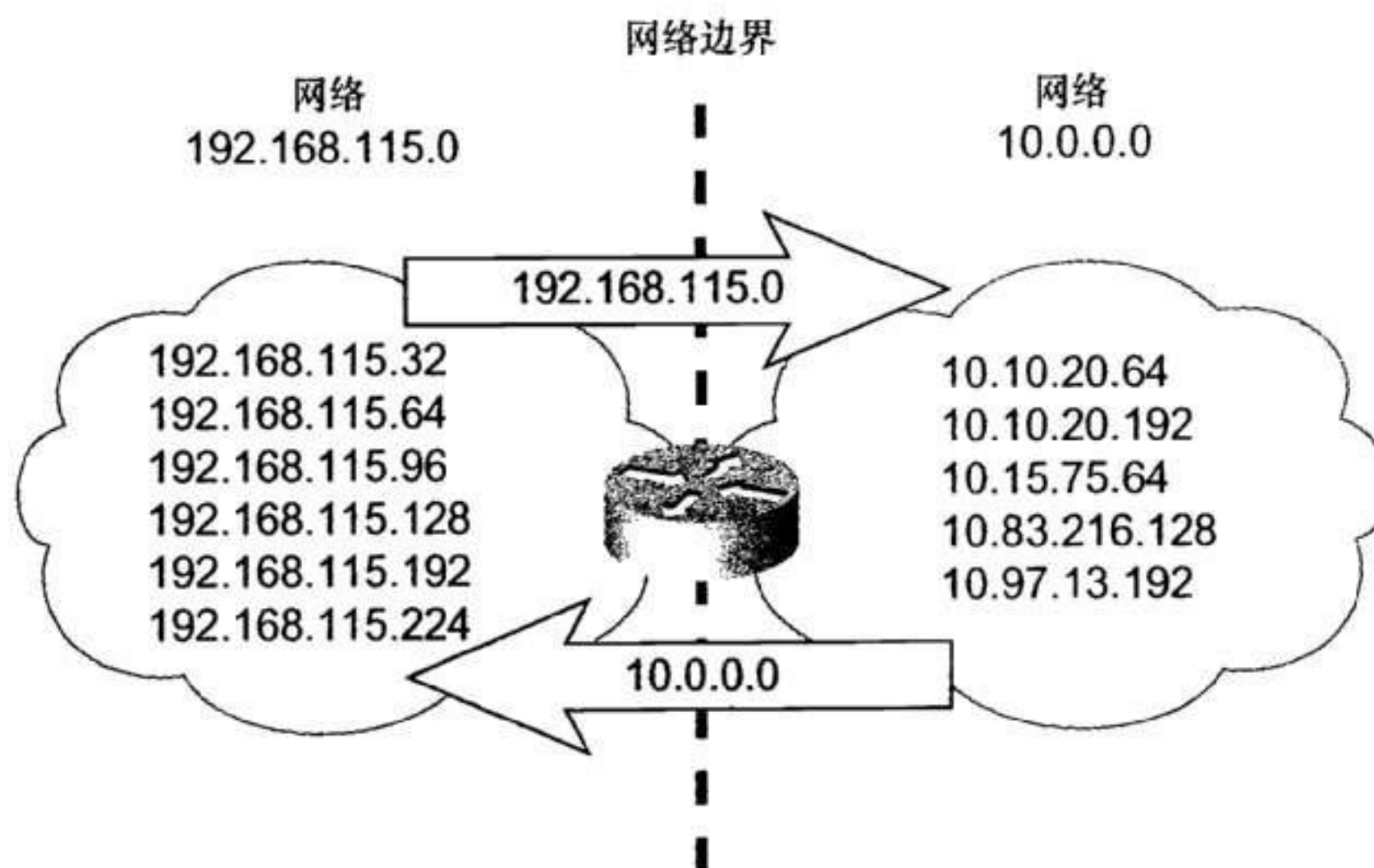


图 5-6 处在两个主网络边界上的路由器不会把其中一个主网络子网通告给另一个主网

```
Raleigh#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
       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

Gateway of last resort is not set

R    10.0.0.0 [120/1] via 192.168.115.40, 00:00:10, Ethernet1
    192.168.115.0 255.255.255.240 is subnetted, 6 subnets
C    192.168.115.32 is directly connected, Ethernet1
R    192.168.115.64 [120/1] via 192.168.115.99, 00:00:13, Ethernet0
C    192.168.115.96 is directly connected, Ethernet0
C    192.168.115.128 is directly connected, Serial0
R    192.168.115.192 [120/1] via 192.168.115.99, 00:00:13, Ethernet0
R    192.168.115.224 [120/1] via 192.168.115.130, 00:00:25, Serial0
Raleigh#
```

图 5-7 这个路由器有一个单独的路由条目指向网络 10.0.0.0，因为表中可以看出该网络可以经过 1 跳到达，所以它的下一跳地址就是边界路由器

第 3 章简要讨论了一个主网络被另一个不同的主网络分割成不连续的子网的情况。这里要注意，这种情况在像 RIP、IGRP 等有类别路由协议中会带来一些问题，不连续的子网在网络边界的地方会被自动汇总。本章在 RIP 配置的一节中描述了这类问题和解决这个问题的办法。

3. 有类别路由选择：小结

有类别路由选择协议的一个基本特征，是在通告目的地址时不能随之一起通告它的地址掩码。因此，有类别路由选择协议首先必须匹配一个与该目的地址对应于 A 类、B 类或 C 类的主网络号。对于每一个通过这台路由器的数据包：

(1) 如果目的地址是一个和路由器直接相连的主网络的成员，那么该网络的路由器接口上配置的子网掩码将被用来确定目的地址的子网。因此，在那个主网络中必须自始至终地统

一使用这个相同的子网掩码。

(2) 如果目的地址不是一个和路由器直接相连的主网络的成员, 那么路由器将仅仅尝试去匹配该目的地址对应于 A 类、B 类或 C 类的主网络号。

5.2 配置 RIP

与 RIP 协议简易的特点相对应的, RIP 协议的配置工作也是一项比较简单的事情。首先, 用一条命令来启动 RIP 进程, 另外还有一条命令用来指定运行 RIP 协议的每一个网络。在此之后, 就只有很少的几个配置选项了。

5.2.1 案例研究 1: 一个基本的 RIP 配置

配置一个 RIP 协议, 只需要两个必要的步骤:

步骤 1: 使用 **router rip** 命令启动 RIP 进程;

步骤 2: 使用 **network** 命令指定每一个需要运行 RIP 协议的主网络。

图 5-8 显示了一个含有 4 台路由器的互连网络, 它包含 4 个主网络号。路由器 Goober 和网络 172.17.0.0 的两个子网相连, 因此, 下面的命令是启动 RIP 协议所必需的:

```
Goober(config)#router rip
Goober(config-router)#network 172.17.0.0
```

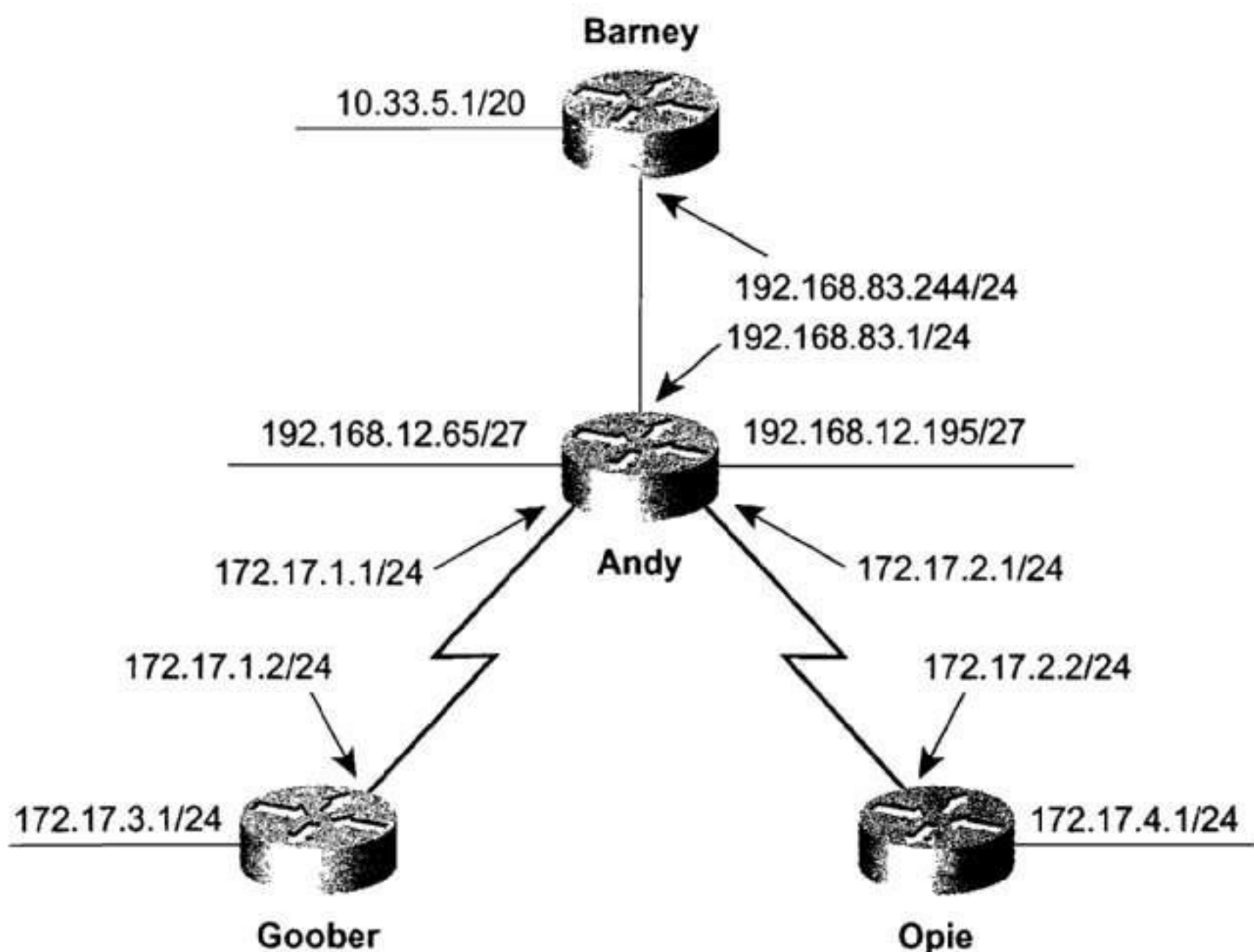


图 5-8 在按照主网络分类的层面上, 路由器 Andy 和 Barney 都是网络之间的边界路由器

类似的, 路由器 Opie 和同一个网络 172.17.0.0 的两个子网相连, 相应的配置命令如下:

```
Opie(config)#router rip
Opie(config-router)#network 172.17.0.0
```


使用任何一个 **router** 命令都需要让路由器进入到 **config-router** 配置模式下,这可以通过提示符辨别出来。由于 RIP 协议具有有类别路由选择的特性,从而在网络边界上会出现子网屏蔽的情形,这意味着 **network** 命令中不需要指定子网,而仅仅需要指定相对应的 A 类、B 类或 C 类的主网络地址。任何一个接口,只要它的配置地址属于 **network** 命令指定的网络,都将会运行 RIP。

路由器 Barney 和两个主网络——10.0.0.0 和 192.168.83.0 相连。因此,这两个主网络都需要被指定:

```
Barney(config)#router rip
Barney(config-router)#network 10.0.0.0
Barney(config-router)#network 192.168.83.0
```

路由器 Andy 和网络 192.168.83.0 的一个子网相连,和网络 192.168.12.0 的两个子网相连,和网络 172.17.0.0 的两个子网相连。因此,它们的配置为:

```
Andy(config)#router rip
Andy(config-router)#network 172.17.0.0
Andy(config-router)#network 192.168.12.0
Andy(config-router)#network 192.168.83.0
```

如图 5-9 所示,在路由器 Andy 上打开了 RIP 协议的调试命令 **debug ip rip**。这里要特别注意的是,路由器 Andy 执行了子网屏蔽。由于 E0 和 E2 接口都是和网络 192.168.12.0 相连的,因而子网 192.168.12.64 和 192.168.12.192 可以在这两个接口上通告出去,但是在和其他不同的网络相连的 E1、S0 和 S1 接口上,这两个子网则会被进行路由汇总后才通告出去。同样,网络 192.168.83.0 和网络 172.17.0.0 在通过有类别网络边界时也会被路由汇总后通告出去。注意,路由器 Andy 也正在接收一条来自于路由器 Barney 关于网络 10.0.0.0 的汇总路由。最后,从图中也可以观察到水平分隔的情况。例如,从 E1 接口通告给路由器 Barney 的路由中就不再包含网络 10.0.0.0 或 192.168.83.0 的路由条目。

```
Andy#debug ip rip
RIP protocol debugging is on
Andy#
RIP: sending update to 255.255.255.255 via Ethernet0 (192.168.12.65)
      subnet 192.168.12.192, metric 1
      network 10.0.0.0, metric 2
      network 192.168.83.0, metric 1
      network 172.17.0.0, metric 1
RIP: sending update to 255.255.255.255 via Ethernet1 (192.168.83.1)
      network 192.168.12.0, metric 1
      network 172.17.0.0, metric 1
RIP: sending update to 255.255.255.255 via Ethernet2 (192.168.12.195)
      subnet 192.168.12.64, metric 1
      network 10.0.0.0, metric 2
      network 192.168.83.0, metric 1
      network 172.17.0.0, metric 1
RIP: sending update to 255.255.255.255 via Serial0 (172.17.1.1)
      subnet 172.17.4.0, metric 2
      subnet 172.17.2.0, metric 1
      network 10.0.0.0, metric 2
      network 192.168.83.0, metric 1
```

待续


```

network 192.168.12.0, metric 1
RIP: sending update to 255.255.255.255 via Serial1 (172.17.2.1)
subnet 172.17.1.0, metric 1
subnet 172.17.3.0, metric 2
network 10.0.0.0, metric 2
network 192.168.83.0, metric 1
network 192.168.12.0, metric 1
RIP: received update from 172.17.1.2 on Serial0
172.17.3.0 in 1 hops
RIP: received update from 192.168.83.244 on Ethernet1
10.0.0.0 in 1 hops
RIP: received update from 172.17.2.2 on Serial1
172.17.4.0 in 1 hops

```

图 5-9 这些 debug 信息显示了路由器 Andy 上接收和发送的 RIP 更新信息，并可以观察到网络路由的汇总和水平分隔的效果

5.2.2 案例研究 2：被动接口 (Passive Interface)

在图 5-10 所示的互连网络中增加了一台路由器 Floyd，但不希望在路由器 Floyd 和 Andy 之间交换 RIP 协议的通告信息，这在路由器 Floyd 上可以很容易的实现：

```

Floyd(config)#router rip
Floyd(config-router)#network 192.168.100.0

```

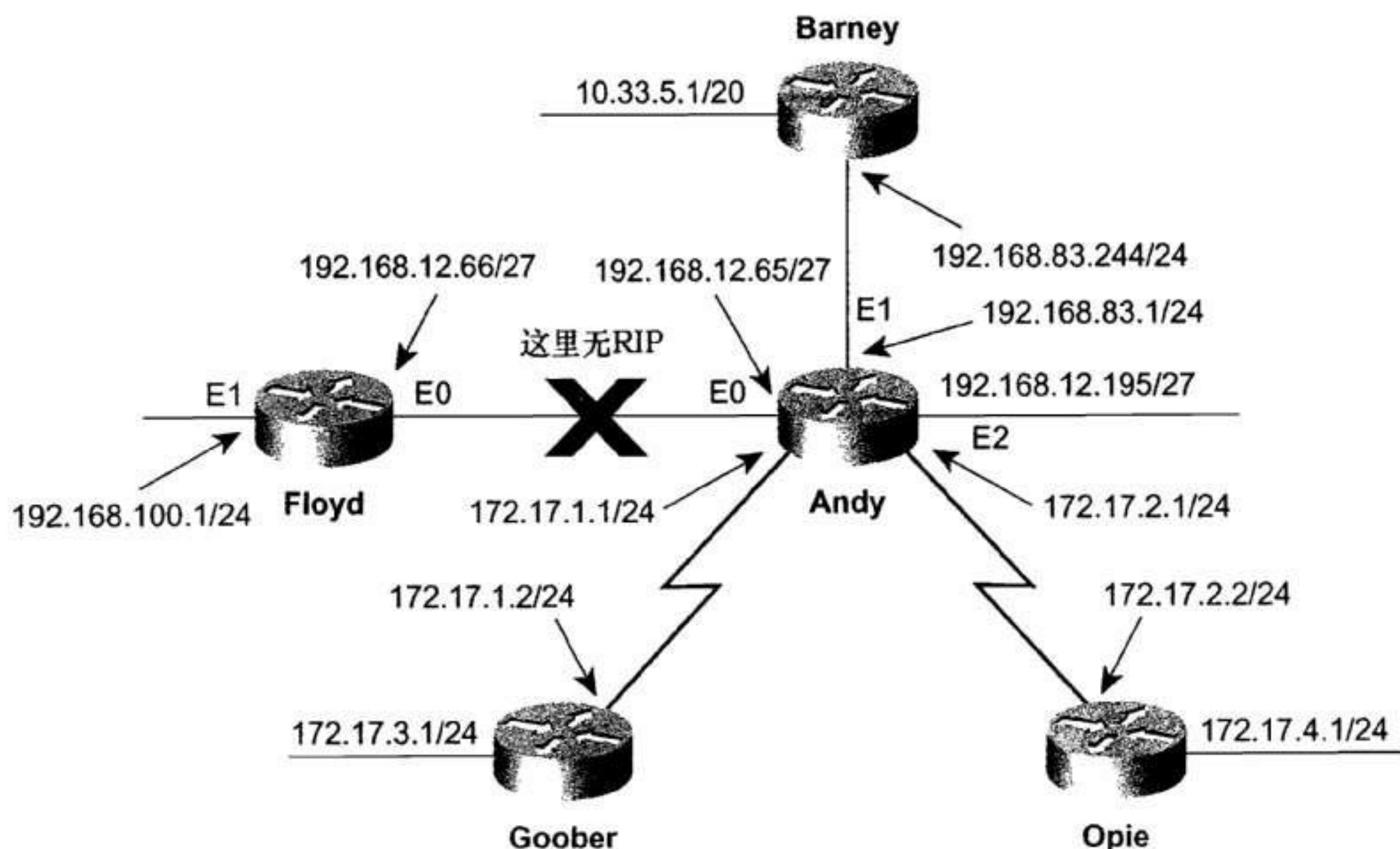


图 5-10 网络的路由策略要求在路由器 Andy 和 Floyd 之间没有 RIP 信息的交换

由于没有包含网络 192.168.12.0 的 **network** 命令语句，路由器 Floyd 将不在接口 192.168.12.66 上通告 192.168.12.0。然而，路由器 Andy 有两个和网络 172.17.0.0 相连的接口，因而网络 172.17.0.0 必须包含在 RIP 协议中。而如果一个路由器接口和启动 RIP 协议的网络的子网相连的话，路由器就会在该接口上发出 RIP 广播，为了阻塞这样的 RIP 广播，在 RIP

的处理中就需要增加一条 **passive-interface** 命令。路由器 Andy 中 RIP 的配置是：

```
router rip
  passive-interface Ethernet0
  network 172.17.0.0
  network 192.168.12.0
  network 192.168.83.0
```

Passive-interface 命令不是 RIP 协议专有的命令，它可以在所有的 IP 路由选择协议中配置使用。使用 **passive-interface** 命令实际上可以说是在一条特定的数据链路上，将路由器作为一台“静”主机来看待。像其他的“静”主机一样，它只是在该特定的链路上侦听 RIP 的广播，从而更新自己的路由选择表。如果希望避免路由器从一条链路上学到路由信息，就必须使用更复杂的路由更新控制才能实现，这种路由更新控制称为出站更新过滤 (filtering out updates)(路由过滤的内容将在第 13 章中讨论)。和“静”主机不同的是，路由器并不在被动接口上响应收到的请求消息。

5.2.3 案例研究 3：配置单播更新(Unicast update)

接下来，增加一台新的路由器 Bea，并连接到路由器 Andy 和 Floyd 之间的以太网共享链路上 (见图 5-11)。在原来路由器 Andy 和 Floyd 之间的链路上依然保留不启动 RIP 协议的路由策略，但在路由器 Bea 和 Andy 之间，Bea 和 Floyd 之间现在都必须交换 RIP 通告信息。

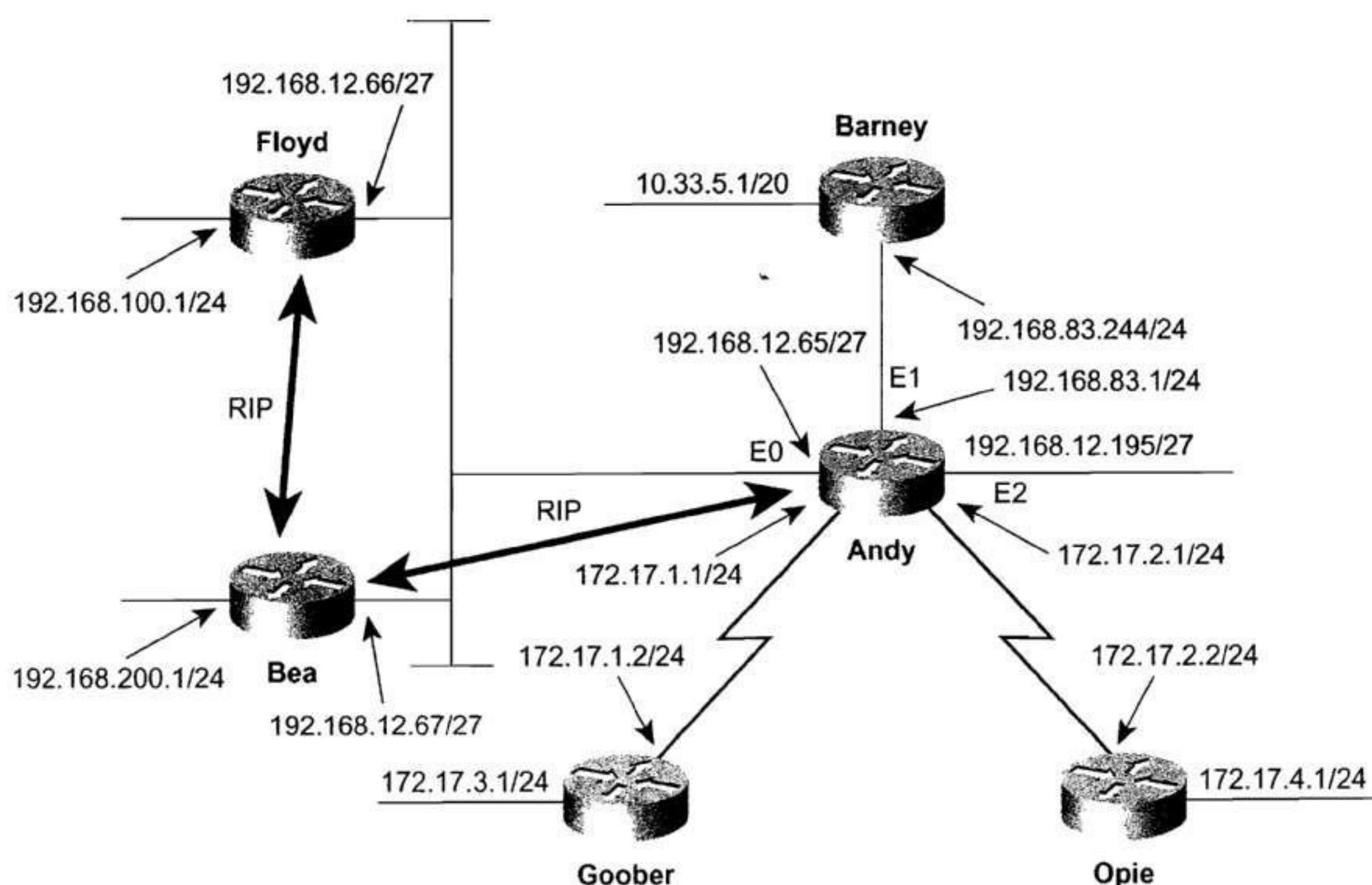


图 5-11 路由器 Andy 和 Floyd 不进行 RIP 更新的交换，但是路由器 Andy 和 Floyd 都与路由器 Bea 交换 RIP 更新

路由器 Bea 的配置很简单:

```
router rip
  network 192.168.12.0
  network 192.168.200.0
```

在路由器 Andy 的 RIP 处理中增加一条额外的命令 **neighbor**, 使 RIP 协议能够以单播方式发送通告给路由器 Bea 的接口, 而这时, 路由器 Andy 上的 **passive-interface** 命令仍继续防止在该链路上广播更新。¹

路由器 Andy 的配置如下:

```
router rip
  passive-interface Ethernet0
  network 172.17.0.0
  network 192.168.12.0
  network 192.168.83.0
  neighbor 192.168.12.67
```

因为路由器 Floyd 现在必须发送 RIP 通告给路由器 Bea, 因此也必须增加一条通告 192.168.12.0 的 **network** 命令。为了防止广播 RIP 更新, 也要增加一条 **Passive-interface** 命令, 并且要增加 **neighbor** 命令以单播方式通告 RIP 更新给路由器 Bea:

```
router rip
  passive-interface Ethernet0
  network 192.168.12.0
  network 192.168.100.0
  neighbor 192.168.12.67
```

在路由器 Andy 上启动调试命令 **debug ip rip events**, 可以用来验证更改后的新配置的效果 (图 5-12)。路由器 Andy 可以从路由器 Bea 收到路由更新, 但是无法从 Floyd 上收到, 并且正在以单播方式直接给路由器 Bea 的接口发送路由更新, 但是却不在它的 E0 接口上进行广播。

```
Andy#debug ip rip events
RIP event debugging is on
Andy#
RIP: received update from 192.168.12.67 on Ethernet0
RIP: Update contains 1 routes
RIP: sending update to 255.255.255.255 via Ethernet1 (192.168.83.1)
RIP: Update contains 4 routes
RIP: sending update to 255.255.255.255 via Ethernet2 (192.168.12.195)
RIP: Update contains 6 routes
RIP: sending update to 255.255.255.255 via Serial0 (172.17.1.1)
RIP: Update contains 7 routes
RIP: sending update to 255.255.255.255 via Serial1 (172.17.2.1)
RIP: Update contains 7 routes
RIP: sending update to 192.168.12.67 via Ethernet0 (192.168.12.65)
RIP: Update contains 4 routes
RIP: received update from 172.17.1.2 on Serial0
```

待续

¹ **neighbor** 的另外应用是在像帧中继这样的非广播介质型网络上发送单播更新。


```
RIP: Update contains 1 routes
RIP: received update from 172.17.2.2 on Serial1
RIP: Update contains 1 routes
RIP: received update from 192.168.12.67 on Ethernet0
RIP: Update contains 1 routes
```

图 5-12 路由器 Andy 在 E0 接口发送出的唯一路由更新是到路由器 Bea 的单播更新。路由器 Andy 可以收到来自路由器 Bea 的更新，但收不到来自路由器 Floyd 的更新

虽然路由器 Bea 可以从路由器 Andy 和路由器 Floyd 学到路由，而且在共享的以太网上广播更新，但由于水平分隔法则依然适用，因此可以防止路由器 Bea 通告从那两个路由器学到的路由重新广播到它们之间的以太网上。

5.2.4 案例研究 4：不连续的子网

在图 5-13 中，另有一台路由器被增加到原来的互联网络上，它通过一个 E1 接口和子网 10.33.32.0/20 相连。现在问题出现了，网络 10.0.0.0 的另一个子网——子网 10.33.0.0/20 是和路由器 Barney 相连的，它和子网 10.33.32.0/20 之间只有一条惟一的经过网络 192.168.83.0 和 192.168.12.0 的路由路径，而这是它们完全不同的两个网络。结果，网络 10.0.0.0 变成不连续的了。

路由器 Barney 会认为自己是网络 10.0.0.0 和网络 192.168.83.0 之间的边界路由器；同样，路由器 Ernest_T 也会认为自己是网络 10.0.0.0 和网络 192.168.12.0 之间的边界路由器。它们都将宣告一条网络 10.0.0.0 的汇总路由，结果路由器 Andy 将会“傻乎乎”地认为它有两条相等代价的路径可以到达同一个网络。在这种情况下，路由器 Andy 将在与路由器 Barney 和 Ernest_T 相连的链路上进行均分负载，因而，要到达网络 10.0.0.0 的数据包现在只有 50% 的机会可以转发到正确的子网上。

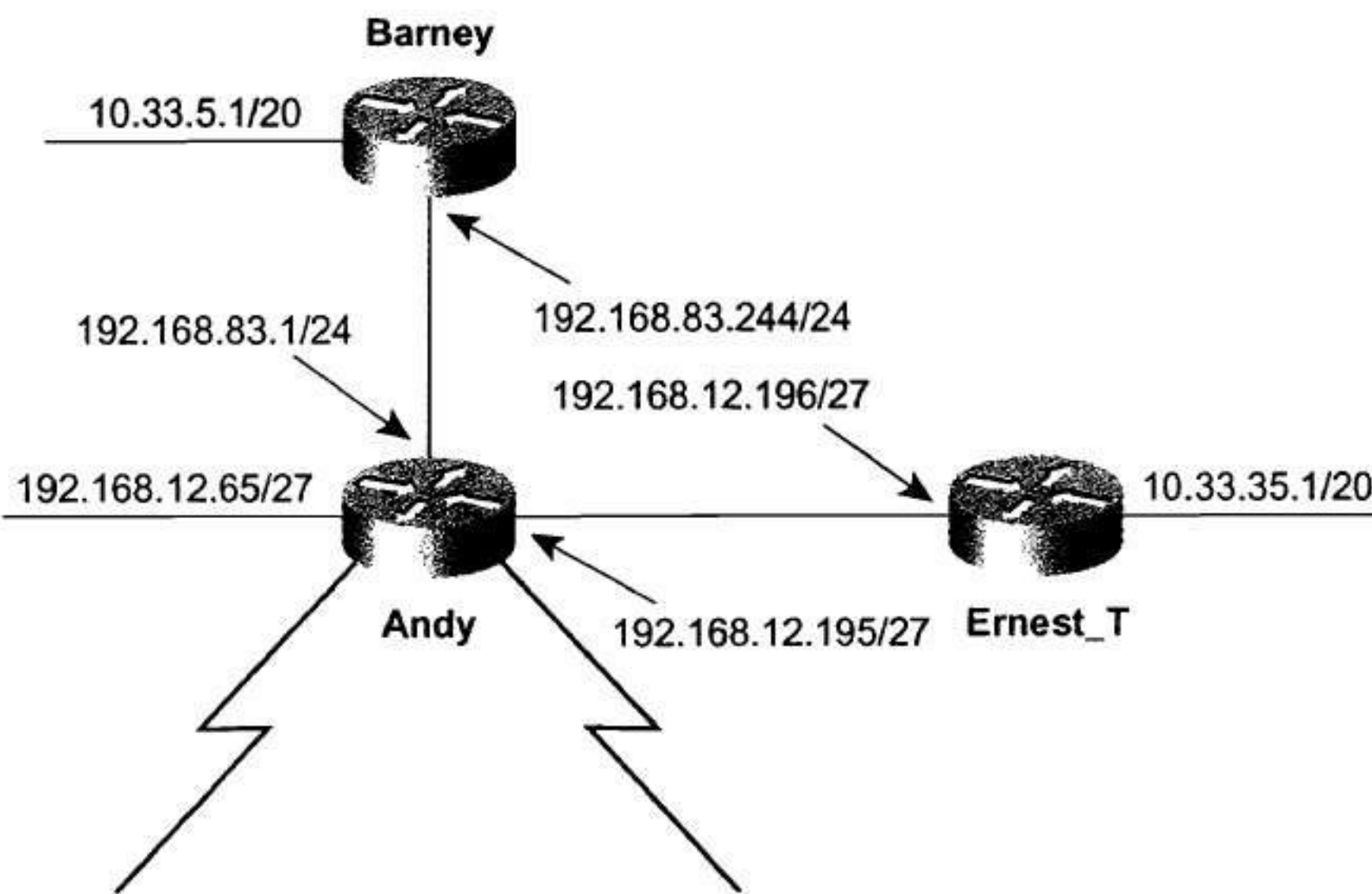


图 5-13 像 RIP 和 IGRP 等有类别路由选择协议不能对图中所示类型的网络拓扑进行路由选择，因为其中的网络 10.0.0.0 的子网被不同的主网络分开了

解决方法是在网络 192.168.83.0/24 和 192.168.12.192/27 所在的同一条链路上配置网络 10.0.0.0 的子网, 这可以通过在路由器接口上配置辅助 IP 地址 (Secondary IP Address) 实现, 配置如下:

```
Barney(config)#interface e0
Barney(config-if)#ip address 10.33.55.1 255.255.240.0 secondary

Andy(config)#interface e1
Andy(config-if)#ip address 10.33.55.2 255.255.240.0 secondary
Andy(config-if)#interface e2
Andy(config-if)#ip address 10.33.75.1 255.255.240.0 secondary
Andy(config-if)#router rip
Andy(config-router)#network 10.0.0.0

Ernest_T(config)#interface e0
Ernest_T(config-if)#ip address 10.33.75.2 255.255.240.0 secondary
```

因为路由器 Andy 在前面的配置中没有和网络 10.0.0.0 相连的接口, 所以在 RIP 配置中增加了一条网络声明 (**network 10.0.0.0**)。配置的效果可以从图 5-14 中看到, 原有的逻辑网络结构依然保留, 只是在其网络结构上“叠加 (overlaid)”了一个连续的网络 10.0.0.0。

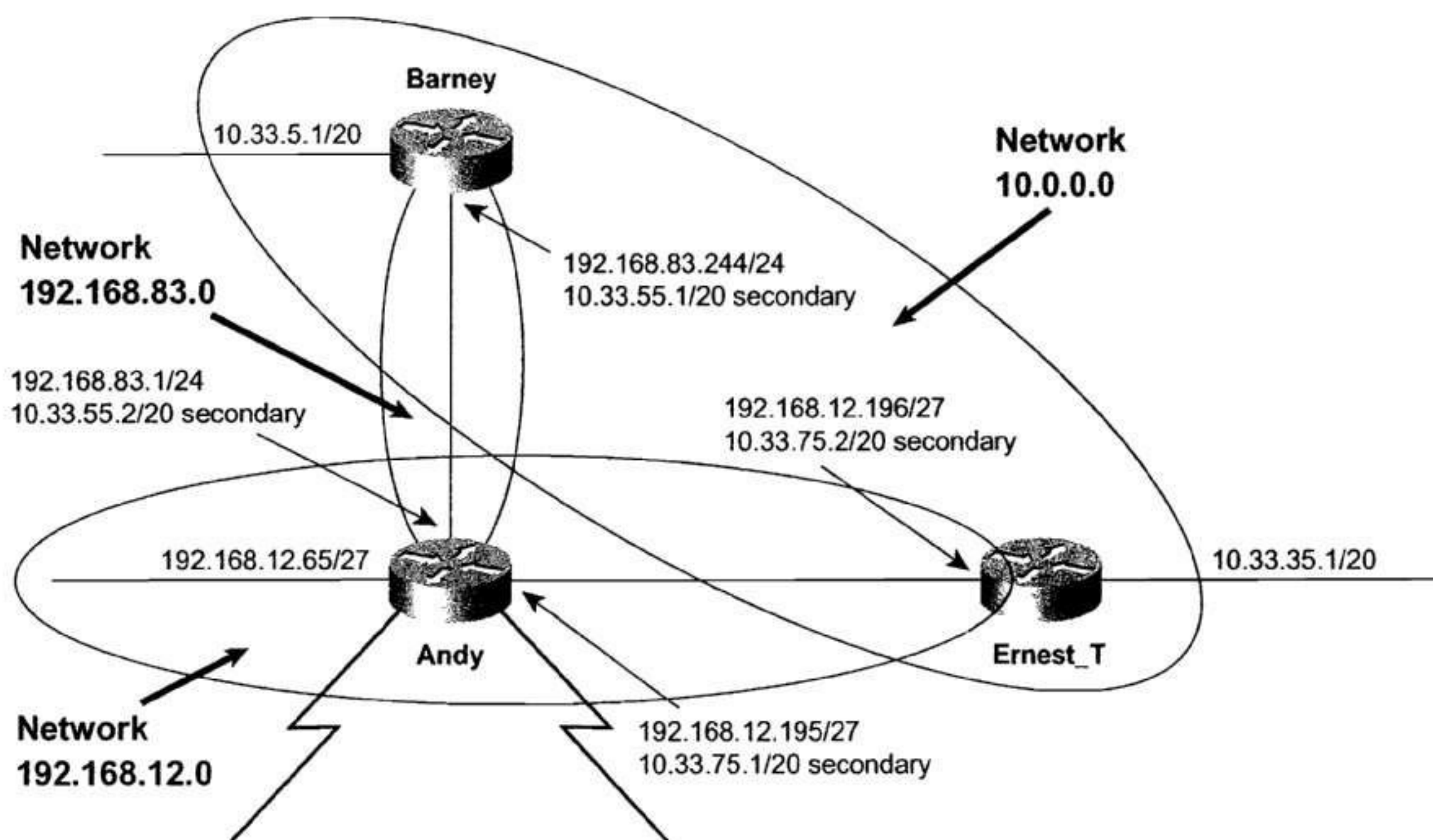


图 5-14 辅助地址被用来在已有其他网络地址的同一条链路上连接网络 10.0.0.0 的子网

图 5-15 显示了路由器 Ernest_T 的路由选择表。这里值得注意的是, 两条等价路由分别和下一跳地址 10.33.75.1 及 192.168.12.195 相关联。

由于路由选择进程会把辅助地址看作是单独的数据链路, 所以在 RIP 协议或 IGRP 协议网络的设计中要很小心地使用。各自的 RIP 更新会在每一个子网里进行广播, 如果路由更新比较多而且物理链路的带宽有限 (例如串行链路), 那么大量的路由更新会造成网络的拥塞。在本章后面的图 5-17 中, 可以观察到这种配置了辅助地址的链路上产生的大量路由更新。

在插入辅助地址时一定要十分小心, 如果不小心忽略了关键字 “secondary”, 路由器将

会认为该接口的主地址被一个新的地址代替了。在一个正在提供服务的网络接口上犯这个错误将会带来严重的后果。

```
Ernest_T#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
       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

Gateway of last resort is not set

 10.0.0.0 255.255.240.0 is subnetted, 4 subnets
C    10.33.32.0 is directly connected, Ethernet1
R    10.33.48.0    [120/1] via 10.33.75.1, 00:00:05, Ethernet0
R    10.33.0.0     [120/2] via 10.33.75.1, 00:00:05, Ethernet0
C    10.33.64.0 is directly connected, Ethernet0
R   192.168.83.0   [120/1] via 192.168.12.195, 00:00:05, Ethernet0
                   [120/1] via 10.33.75.1, 00:00:05, Ethernet0
192.168.12.0 255.255.255.224 is subnetted, 2 subnets
R    192.168.12.64 [120/1] via 192.168.12.195, 00:00:05, Ethernet0
C    192.168.12.192 is directly connected, Ethernet0
R   192.168.200.0  [120/2] via 192.168.12.195, 00:00:05, Ethernet0
                   [120/2] via 10.33.75.1, 00:00:05, Ethernet0
R   172.17.0.0    [120/1] via 192.168.12.195, 00:00:06, Ethernet0
                   [120/1] via 10.33.75.1, 00:00:06, Ethernet0
Ernest_T#
```

图 5-15 这台路由器的路由选择进程把子网 192.168.12.192/27 和 10.33.64.0/20 看作是和分开的链路相连的，但是它们实际上是和同一个物理接口相连的

5.2.5 案例研究 5：掌握 RIP 的度量

如图 5-16 所示，在路由器 Ernest_T 和 Barney 之间增加一条串行链路用作备份链路，只有当路由经过路由器 Andy 失败时这条链路才被使用。现在的问题在于，路由器 Barney 的子网 10.33.0.0 和路由器 Ernest_T 的子网 10.33.32.0 之间经过这条串行链路的路径是 1 跳，而经过那条优选的以太网链路的路径却是 2 跳。在正常的情况下，RIP 会首先选择串行链路。

可以通过 **offset-list** 命令来改变路由的度量值，它指定一个数值来加大路由的度量值，并且参照一个访问列表（access list）¹来决定哪些路由条目需要改变。这条命令的语法格式是：

```
offset-list {access-list-number | name} {in| out} offset [type number]
```

路由器 Ernest_T 的配置如下：

```
Ernest_T(config)#access-list 1 permit 10.33.0.0 0.0.0.0
Ernest_T(config)#router rip
Ernest_T(config-router)#network 192.168.12.0
Ernest_T(config-router)#network 10.0.0.0
Ernest_T(config-router)#offset-list 1 in 2 Serial0
```

访问列表的配置确定了关于子网 10.33.0.0 的路由，偏移列表（offset list）语句的含义是说：“先检查从 S0 接口接收进来的 RIP 通告，如果存在和访问列表 1 指定的地址相匹配的

¹ 参照附录 B 中关于访问列表的讲述。

路由条目, 那么就把该路由条目的度量值加大 2 跳。”

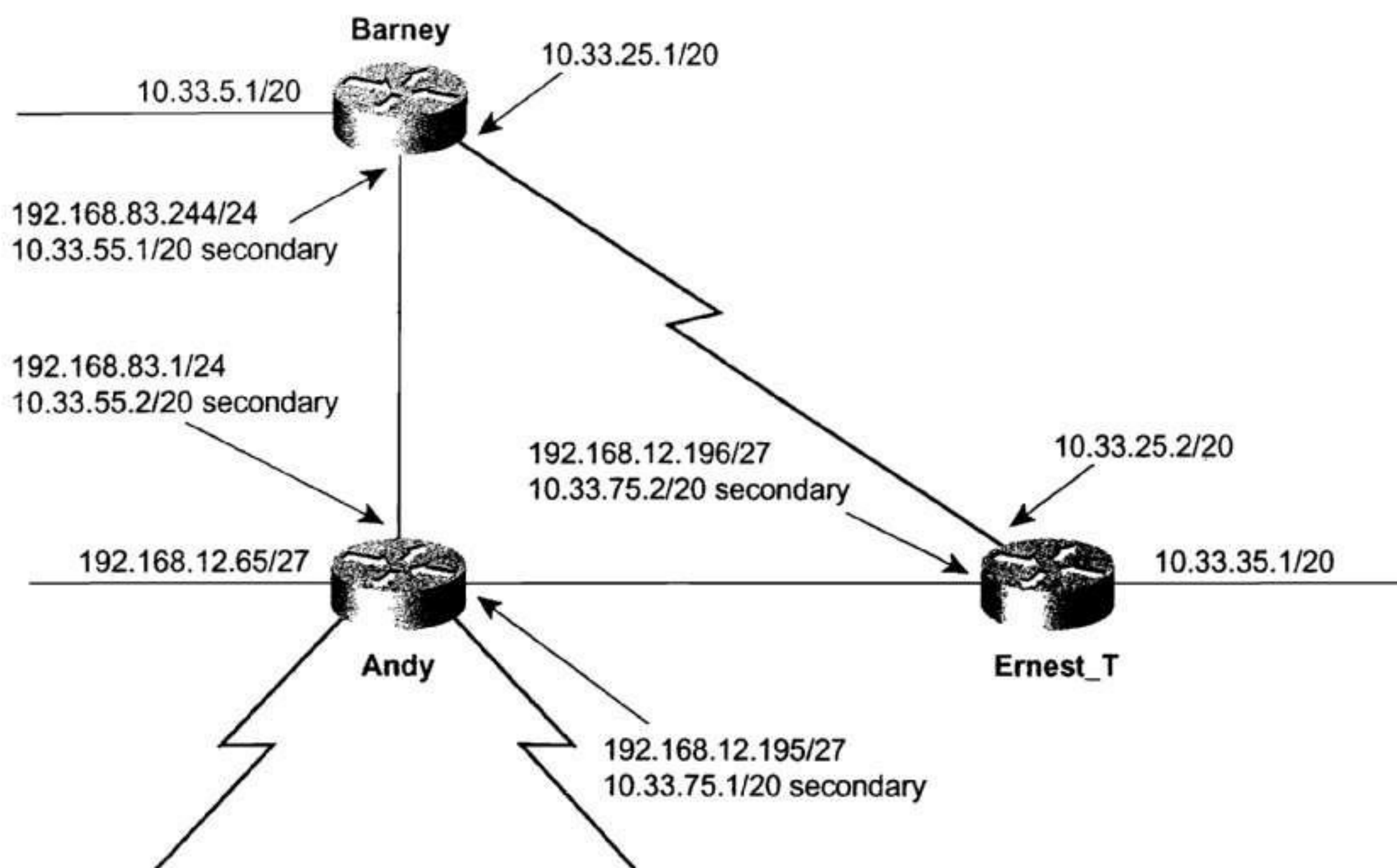


图 5-16 RIP 的度量值必须要更改, 以便使路由器 Barney 和 Ernest_T 之间 2 跳的以太网路由优先于 1 跳的串行链路路由

路由器 Barney 更新配置后, 在它的配置文件中包含了下面的语句:

```
router rip
  offset-list 5 in 2 Serial0
  network 10.0.0.0
  network 192.168.83.0
!
```

在图 5-17 中显示了路由器 Ernest_T 的配置所产生的效果。

```
Ernest_T#debug ip rip
RIP protocol debugging is on
Ernest_T#
RIP: received update from 192.168.12.195 on Ethernet0
  192.168.12.64 in 1 hops
  10.0.0.0 in 1 hops
  192.168.83.0 in 1 hops
  192.168.200.0 in 2 hops
  172.17.0.0 in 1 hops
RIP: received update from 10.33.75.1 on Ethernet0
  10.33.48.0 in 1 hops
  10.33.0.0 in 2 hops
  192.168.83.0 in 1 hops
  192.168.12.0 in 1 hops
  192.168.200.0 in 2 hops
  172.17.0.0 in 1 hops
RIP: received update from 10.33.25.1 on Serial0
  10.33.32.0 in 3 hops
```

待续


```

10.33.48.0 in 1 hops
10.33.0.0 in 3 hops
192.168.83.0 in 1 hops
192.168.200.0 in 3 hops
172.17.0.0 in 2 hops
RIP: sending update to 255.255.255.255 via Ethernet0 (192.168.12.196)
network 10.0.0.0, metric 1
RIP: sending update to 255.255.255.255 via Ethernet0 (10.33.75.2)
subnet 10.33.32.0, metric 1
subnet 10.33.16.0, metric 1
RIP: sending update to 255.255.255.255 via Ethernet1 (10.33.35.1)
subnet 10.33.48.0, metric 2
subnet 10.33.0.0, metric 3
subnet 10.33.16.0, metric 1
subnet 10.33.64.0, metric 1
network 192.168.83.0, metric 2
network 192.168.12.0, metric 1
network 192.168.200.0, metric 3
network 172.17.0.0, metric 2
RIP: sending update to 255.255.255.255 via Serial0 (10.33.25.2)
subnet 10.33.32.0, metric 3
subnet 10.33.0.0, metric 3
subnet 10.33.64.0, metric 1
network 192.168.12.0, metric 1
network 192.168.200.0, metric 3
network 172.17.0.0, metric 2

```

图 5-17 偏移列表指定额外增加的跳数，把子网 10.33.0.0/20 经过 S0 接口的路由度量由 1 跳变成了 3 跳，
但经过 E0 接口的路由的跳数没有改变，还是 2 跳

作为另一种选择，路由器也可以通过配置去更改向外通告的出站（outgoing）路由更新的度量，替代上述两台路由器对从链路上接收的入站（incoming）路由更新的度量的更改。下面的配置可以和前面的配置达到同样的效果：

路由器 ERNEST_T:

```

router rip
  offset-list 3 out 2 Serial0
  network 192.168.12.0
  network 10.0.0.0
!
access-list 3 permit 10.33.32.0 0.0.0.0

```

路由器 BARNEY:

```

router rip
  offset-list 7 out 2 Serial0
  network 10.0.0.0
  network 192.168.83.0
!
access-list 7 permit 10.33.0.0 0.0.0.0

```


偏移列表的其他几个选项在配置时也是有用的。如果不指定使用偏移列表的接口,那么偏移列表将在所有与访问列表匹配的接口上更改所有的入站更新或出站更新。如果不调用访问列表(使用 0 作为访问列表的序列号)来做匹配,偏移列表将更改所有的入站更新或出站更新。

当在入站或出站更新的通告上选择是否使用偏移列表时,有些需要注意的地方。例如,在一个多于两台路由器的广播型网络上,就必须考虑清楚,到底是需要某个单独的路由器向它所有的邻居路由器广播偏移更改后的通告,还是需要某个单独的路由器接收偏移更改后的通告。

在正在运行使用的路由上实施偏移列表时也需要特别注意。因为当一个偏移列表引起下一跳路由器通告的度量值比它正在通告的路由更新的度量值更高时,将直到抑制计时器(holddown timer)超时前,这条路由都会被标记成不可到达。

5.3 RIP 故障排除

RIP 协议的故障排除相对来说是比较简单的。对于 RIP 这样的有类别路由选择协议来说,最困难的排错就是出现子网掩码配置错误或者子网不连续的情形。如果路由选择表包含了不准确的或被丢失的路由,那么就应该检查邻近的所有子网和所有子网掩码的一致性。

最后,有一条命令在一台高速路由器向一台低速路由器发送大量 RIP 信息时可能比较有用。在这种情况下,低速路由器可能不能像接收一样快地处理这些路由更新,因此就可能会丢失路由信息。这种情况可以通过在 RIP 的处理中使用 **Output-delay** 命令来设置一个 8~50ms 的发包之间的延迟间隙(缺省为 0ms)来解决。

5.4 展 望

RIP 协议的简单、成熟和使用的广泛性确保了它还将会使用许多年。然而, RIP 协议由于过于简单也限制了它的使用范围,使得它仅能适用于小型的、单一的互连网络。下一章开始讲述 IGRP 协议,在 Cisco 公司的这个私有协议中将会讨论一些 RIP 协议的局限性。

5.5 总结表: 第 5 章命令总结

命 令	描 述
debug ip rip [events]	简要地显示路由器收发的 RIP 信息
ip address ip-address mask secondary	在接口上指定一个 IP 地址作为辅助地址
neighbor ip-address	通过指定接口邻居的 IP 地址来建立邻接关系
network network-number	指定一个需要运行 RIP 的网络
offset-list {access-list-number name} {in out} offset [type number]	指定路由选择表中一个与指定的访问列表匹配的路由条目,将自己的度量值增加一个 offset 指定的偏移量
output-delay delay	设定一个指定延迟长度的延迟间隙,以便协调高速路由器和低速路由器之间的延迟问题
passive-interface type number	在指定类型和序列号的接口上阻止 RIP 广播
router rip	启动 RIP 进程
timers basic update invalid holddown flush	更改指定的计时器的值

5.6 推荐读物

Hedrick, C. "Routing Information Protocol", RFC 1058, 1988 年 6 月。

5.7 复习题

1. RIP 协议使用什么端口?
2. RIP 协议使用什么度量? 怎样用度量来表示一个不可达的网络?
3. RIP 协议的更新周期是多少?
4. 在一条路由被标记成不可到达之前, 必须忽略多少更新?
5. 垃圾收集计时器的用途是什么?
6. 为什么触发更新要使用一个随机计时器? 这个计时器的大小范围是什么?
7. RIP 协议的请求消息和响应消息之间有哪些不同之处?
8. RIP 协议使用哪两种类型的请求消息?
9. 在什么情况下会发出一个 RIP 协议的响应消息?
10. 为什么 RIP 协议在主网络的边界处会屏蔽子网?

5.8 配置练习

1. 写出图 5-18 中所示的 6 台路由器的有关配置, 使它们可以利用 RIP 协议来为所有的子网进行路由选择。

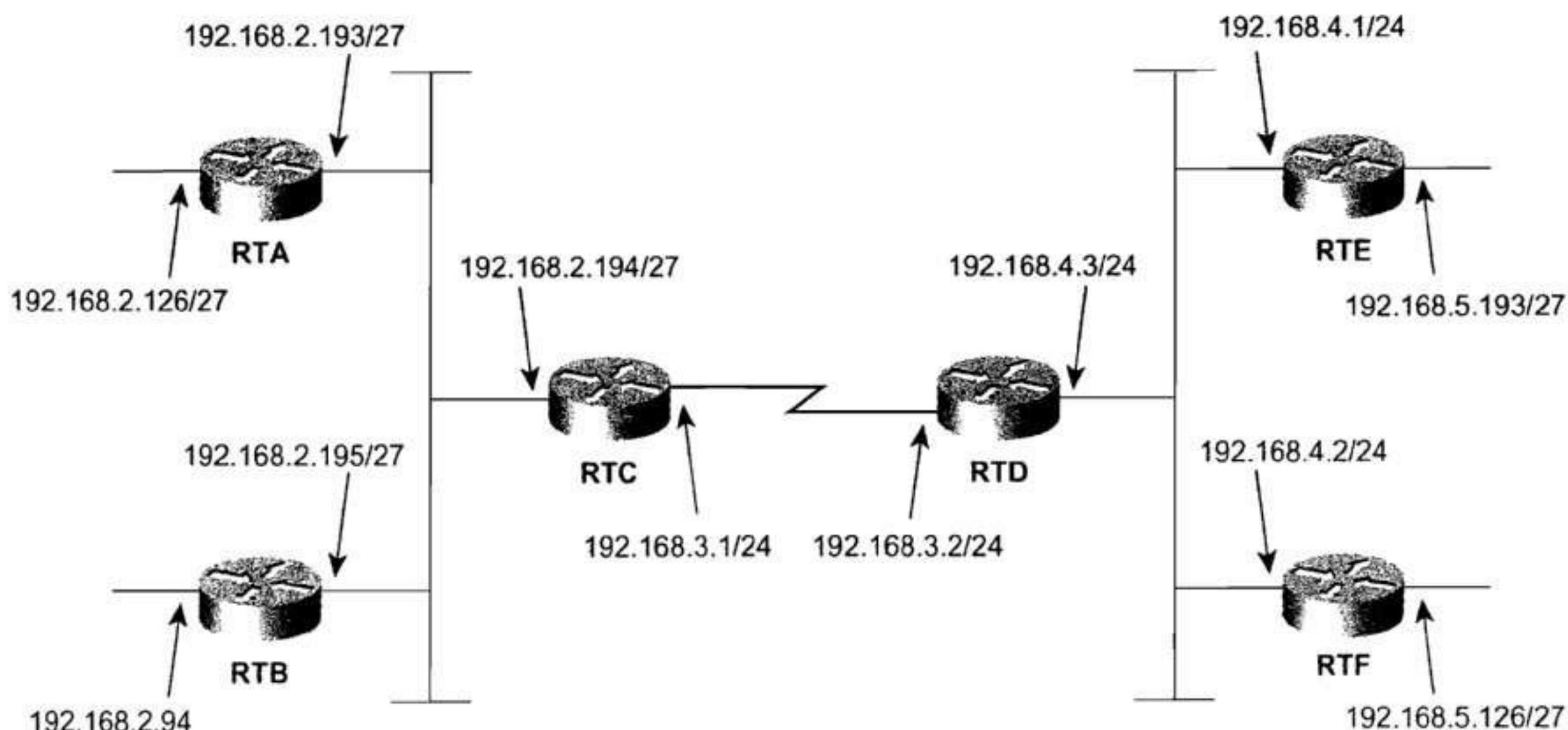


图 5-18 配置练习 1~4 的互联网络

2. 更改配置练习 1 中的配置, 使路由器 RTC 和 RTD 之间 RIP 更新的通告由广播方式改为单播方式。

3. 在图 5-18 中, 路由器 RTC 和 RTD 之间的串行链路的带宽十分有限, 请进一步调整 RIP 协议的配置, 使得经过这条链路的 RIP 更新能够每两分钟发送一次。这里要仔细考虑的是, 必须要更改哪些计时器? 以及必须要更改哪些路由器上的计时器?

4. 制定一个路由策略, 使网络 192.168.4.0 在路由器 RTA 看来是不可到达的, 而网络 192.168.5.0 在路由器 RTB 看来是不可到达的, 可以利用偏移列表来实现这个策略。

5. 依照“有类别路由选择: 直连的子网络”一节所述, 在一个主类别分类的网络中的所有子网掩码必须是一致的。但那一节中却没有强调一个主类别分类的网络内的子网掩码必须是相同的。图 5-19 中的那两台路由器的 RIP 配置如下:

```
router rip
network 192.168.20.0
```

在这个小型的互联网络中, 数据包可以被正确地路由转发吗? 解释一下为什么可以或者为什么不可以。

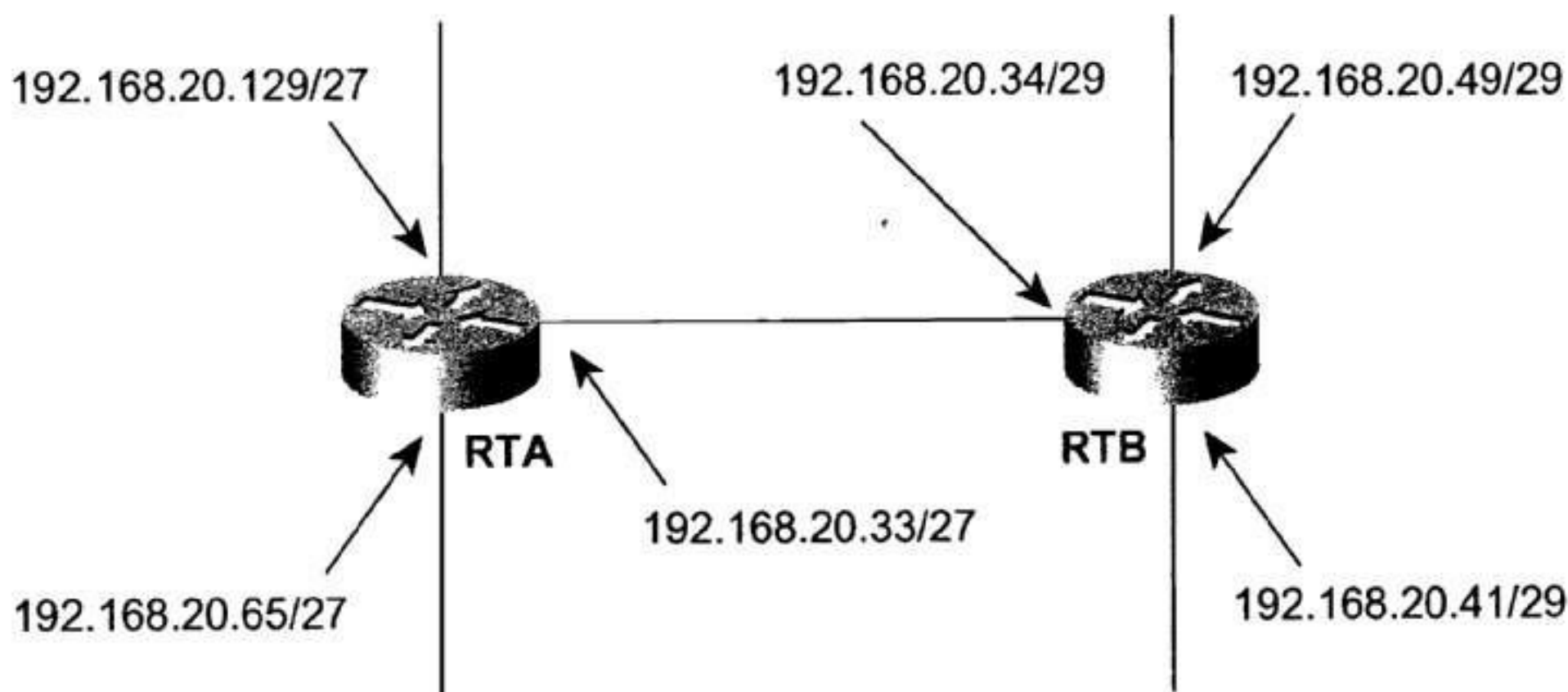


图 5-19 配置练习 5 使用的互联网络

5.9 故障排除练习

1. 在第一个偏移列表的例子中, 路由器 Barney 上的访问列表由

```
access-list 5 permit 10.33.32.0 0.0.0.0
```

更改成

```
access-list 5 deny 10.33.32.0 0.0.0.0
access-list 5 permit any
```

会出现什么结果?

2. 在图 5-20 中显示了一个互联网络, 其中有一台路由器的 IP 地址掩码配置错误了。在图 5-21~图 5-23 中分别显示了路由器 RTA、RTB 和 RTC 的路由选择表。请读者根据前面所

了解的关于 RIP 协议通告和接收路由更新的知识,解释一下路由器 RTB 的路由选择表中的每一个路由选择表项。并请解释一下路由器 RTB 的路由选择表中子网 172.16.26.0 的掩码为什么是 32 位的? 在所有的路由选择表中, 如果存在被丢失的路由条目, 请解释为什么?

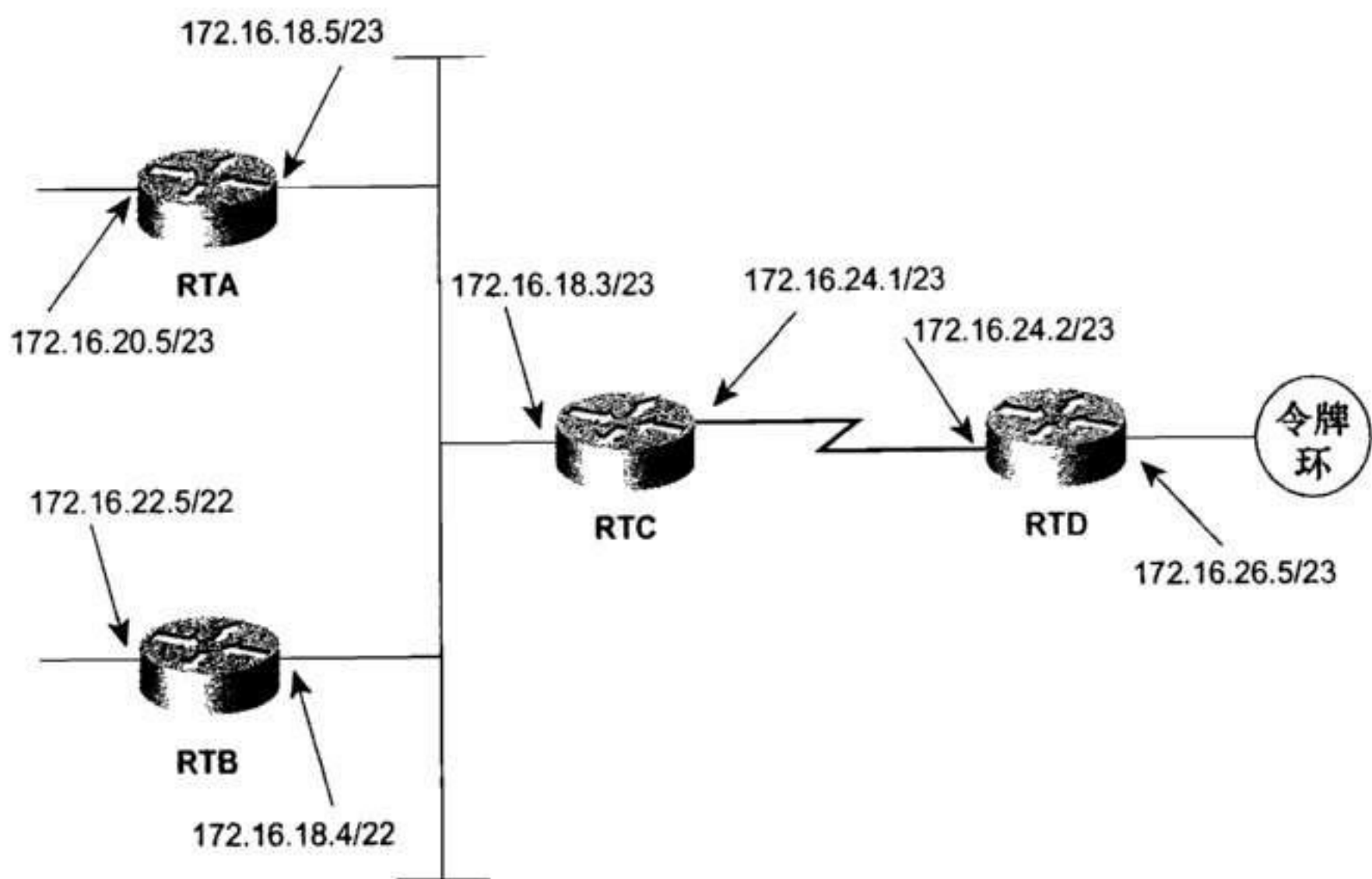


图 5-20 故障排除练习 2 和 3 的互联网络

```
RTA#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
       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
Gateway of last resort is not set
    172.16.0.0/16 is subnetted, 4 subnets
R       172.16.24.0 [120/1] via 172.16.18.3, 00:00:01, Ethernet0
R       172.16.26.0 [120/2] via 172.16.18.3, 00:00:01, Ethernet0
C       172.16.20.0 is directly connected, Ethernet1
C       172.16.18.0 is directly connected, Ethernet0
RTA#
```

图 5-21 图 5-20 中路由器 RTA 的路由选择表

```
RTB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default,
       U - per-user static route, o - ODR
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
R       172.16.24.0/22 [120/1] via 172.16.18.3, 00:00:20, Ethernet0
R       172.16.26.0/32 [120/2] via 172.16.18.3, 00:00:20, Ethernet0
C       172.16.20.0/22 is directly connected, Ethernet1
C       172.16.16.0/22 is directly connected, Ethernet0
RTB#
```

图 5-22 图 5-20 中路由器 RTB 的路由选择表


```

RTC#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

    172.16.0.0/23 is subnetted, 4 subnets
C       172.16.24.0 is directly connected, Serial0
R       172.16.26.0 [120/1] via 172.16.24.2, 00:00:09, Serial0
R       172.16.20.0 [120/1] via 172.16.18.5, 00:00:25, Ethernet0
C       172.16.18.0 is directly connected, Ethernet0
RTC#

```

图 5-23 图 5-20 中路由器 RTC 的路由选择表

3. 如图 5-20 所示, 在子网 172.16.18.0/23 上的用户一直抱怨和子网 172.16.26.0/23 的连接总是时断时续的——有时通, 有时不通 (路由器 RTB 上配置错误的掩码已经改为正确的了)。起初检查路由器 RTC 和 RTD 的路由选择表 (如图 5-24) 也没有什么问题, 所有的子网都在路由选择表中。然而经过 1min 或更长一点的时间, 发现路由器 RTC 显示的子网 172.16.26.0/23 变得不可到达了 (如图 5-25), 而路由器 RTD 依然显示出所有的子网。再经过几分钟后, 路由器 RTC 的路由选择表中又看到了那个子网 (如图 5-26)。在显示这 3 张图示的每一个的时候, 路由器 RTD 的路由选择表都没有变化。请仔细检查图 5-24~图 5-26 中的路由选择表所隐含的问题, 看看是什么问题。

```

RTC#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

    172.16.0.0/23 is subnetted, 5 subnets
C       172.16.24.0 is directly connected, Serial0
R       172.16.26.0 [120/1] via 172.16.24.2, 00:02:42, Serial0
R       172.16.20.0 [120/1] via 172.16.18.5, 00:00:22, Ethernet0
R       172.16.22.0 [120/1] via 172.16.18.4, 00:00:05, Ethernet0
C       172.16.18.0 is directly connected, Ethernet0

```

```

RTD#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
       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

Gateway of last resort is not set

    172.16.0.0/16 is subnetted, 5 subnets
C       172.16.24.0 is directly connected, Serial0
C       172.16.26.0 is directly connected, TokenRing0
R       172.16.20.0 [120/2] via 172.16.24.1, 00:00:00, Serial0
R       172.16.22.0 [120/2] via 172.16.24.1, 00:00:00, Serial0
R       172.16.18.0 [120/1] via 172.16.24.1, 00:00:00, Serial0

```

图 5-24 图 5-20 中路由器 RTC 和 RTD 的路由选择表


```

RTC#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

    172.16.0.0/23 is subnetted, 5 subnets
C       172.16.24.0 is directly connected, Serial0
R       172.16.26.0/23 is possibly down,
        routing via 172.16.24.2, Serial0
R       172.16.20.0 [120/1] via 172.16.18.5, 00:00:19, Ethernet0
R       172.16.22.0 [120/1] via 172.16.18.4, 00:00:24, Ethernet0
C       172.16.18.0 is directly connected, Ethernet0

```

```

RTD#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
       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

Gateway of last resort is not set

    172.16.0.0/16 is subnetted, 5 subnets
C       172.16.24.0 is directly connected, Serial0
C       172.16.26.0 is directly connected, TokenRing0
R       172.16.20.0 [120/2] via 172.16.24.1, 00:00:15, Serial0
R       172.16.22.0 [120/2] via 172.16.24.1, 00:00:15, Serial0
R       172.16.18.0 [120/1] via 172.16.24.1, 00:00:15, Serial0

```

图 5-25 在图 5-24 显示大约 60s 后检查得到的路由器 RTC 和 RTD 的路由选择表

```

RTC#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

    172.16.0.0/23 is subnetted, 5 subnets
C       172.16.24.0 is directly connected, Serial0
R       172.16.26.0 [120/1] via 172.16.24.2, 00:00:09, Serial0
R       172.16.20.0 [120/1] via 172.16.18.5, 00:00:11, Ethernet0
R       172.16.22.0 [120/1] via 172.16.18.4, 00:00:18, Ethernet0
C       172.16.18.0 is directly connected, Ethernet0

```

待续


```
RTD#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
       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

Gateway of last resort is not set

  172.16.0.0/16 is subnetted, 5 subnets
C      172.16.24.0 is directly connected, Serial0
C      172.16.26.0 is directly connected, TokenRing0
R      172.16.20.0 [120/2] via 172.16.24.1, 00:00:19, Serial0
R      172.16.22.0 [120/2] via 172.16.24.1, 00:00:19, Serial0
R      172.16.18.0 [120/1] via 172.16.24.1, 00:00:19, Serial0
```

图 5-26 在图 5-25 显示大约 120s 后检查得到的路由器 RTC 和 RTD 的路由选择表

第 6 章

内部网关路由选择 协议 (IGRP)

本章包括以下主题：

- IGRP 的操作
IGRP 的计时器和稳定性
IGRP 的度量
IGRP 的报文格式
- IGRP 的配置案例
案例研究：一个基本的 IGRP 配置
案例研究：非等价负载均衡
案例研究：设置最大的路径数
案例研究：多个 IGRP 进程
- IGRP 的故障排除
案例研究：非等价负载均衡
案例研究：被分段的网络

80 年代中期，作为对 RIP 协议局限性的回应，Cisco 公司开发了 IGRP 协议，其中最重要的变化就是跳数的度量和 15 跳对互联网络口径大小的限制。IGRP 通过多样的路由变量参数计算出一个复合型的度量，并提供一些“旋钮”给这些变量参数施以权重，以便反映和衡量互联网络的一些特征和需求。虽然跳数并不作为这些变量参数之一，但 IGRP 协议还是延续使用了跳数，并且能够支持网络口径最大为 255 跳的互联网络。

IGRP 协议相对于 RIP 协议还有其他一些优点，如非等价负载均衡 (Unequal-cost load balancing)、比 RIP 协议长 3 倍的更新周期以及更有效率的更新报文格式等等。但是，IGRP

协议的一个主要缺点是它仅是 Cisco 公司的私有协议, 因此, 只能在 Cisco 公司的产品平台上使用, 而 RIP 协议则可以作为所有平台上的任何 IP 路由选择处理的一部分来使用。

Cisco 公司开发 IGRP 协议的主要目的是要创建一个功能强大的通用协议, 以便使它能适应可路由选择协议族的多样性。虽然 IGRP 协议在 IP 网络上已经被证明是一个十分受欢迎的路由选择协议, 但是它还适用于另一个可路由选择的协议, 也就是 ISO 无连接网络协议 (Connectionless Network Protocol, CLNP)。查看一下 Cisco 公司的网络配置手册, 可以获取关于 IGRP 协议在 CLNS 网络上进行路由选择的进一步信息。

6.1 IGRP 的操作

从宏观的角度来看, IGRP 协议继承了许多 RIP 协议的操作特点。IGRP 协议也是一个有类别距离矢量型协议, 除了被水平分隔法则抑制的路由外, IGRP 将不断地周期性地向邻居路由器广播它的整张路由选择表。像 RIP 协议一样, 路由器启动时, IGRP 在所有运行 IGRP 协议的接口上广播出一个请求报文, 并对收到的更新执行一个完整性的检查, 用来验证更新报文的源地址是否和收到更新的那个子网属于同一个子网。¹新的带有可达路由度量值的路由更新条目将会被放置在路由选择表中, 并且仅当它所带的度量值小于到达相同目的地址的原有路由条目的度量值时, 才能替代原有的路由条目。IGRP 协议同样使用带毒性反转的水平分隔法则、触发更新和抑制计时器等手段来保证它的稳定性; 同 RIP 协议一样, IGRP 协议也在网络边界上进行路由汇总。

与 RIP 协议不同, IGRP 协议不使用 UDP 来交换报文, 它直接通过 IP 层的端口号 9 来进行报文交换。

IGRP 协议使用了一个称为自主系统 (Autonomous System) 的概念。回忆第 4 章“动态路由选择协议”中所述, 自主系统可以定义为一个路由选择域 (Routing Domain), 也可以定义为一个进程域 (Process Domain)。IGRP 自主系统是一个进程域——一组使用 IGRP 协议作为共同的路由选择协议的路由器。

通过定义和跟踪多个自主系统, IGRP 协议允许在一个 IGP 环境里面运行多个进程域, 这样可以把一个域内部的通信和另一个域内部的通信孤立开来。域间的通信量可以通过路由重新分配 (Redistribution) (第 11 章“路由重新分配”) 和路由过滤 (Route Filtering) (第 13 章“路由过滤”) 来严密地控制。

图 6-1 显示了进程域和路由选择域的对照。这里定义了两个自主系统 (AS): AS 10 和 AS 40, 这些系统是路由选择域——在一个共同的管理机构下运行一个或多个 IGP 协议的路由器集合。它们通过外部网关协议 (Exterior Gateway Protocol, 在这个实例中, 是边界网关协议, 或称为 BGP 协议) 来通信。

在自主系统 AS 10 中有两个 IGRP 进程域: IGRP 20 和 IGRP 30。在 IGRP 协议内, 定义了两个自主系统号 20 和 30, 就此处而言, 这些数字是用来区分同一个路由选择域内的两个不同路由选择进程的。进程域 IGRP 20 和进程域 IGRP 30 是通过和这两个进程域都相连的一台路由器来进行通信的。这台路由器同时运行两个 IGRP 进程, 并且在这两个进程之间进行

¹ 这个完整性检查可以用命令 `no validate-update-sourced` 来禁止它生效。

路由再分配。在本章关于 IGRP 配置的一节里包含了一个这方面的案例, 用来说明多个 IGRP 进程域的配置。

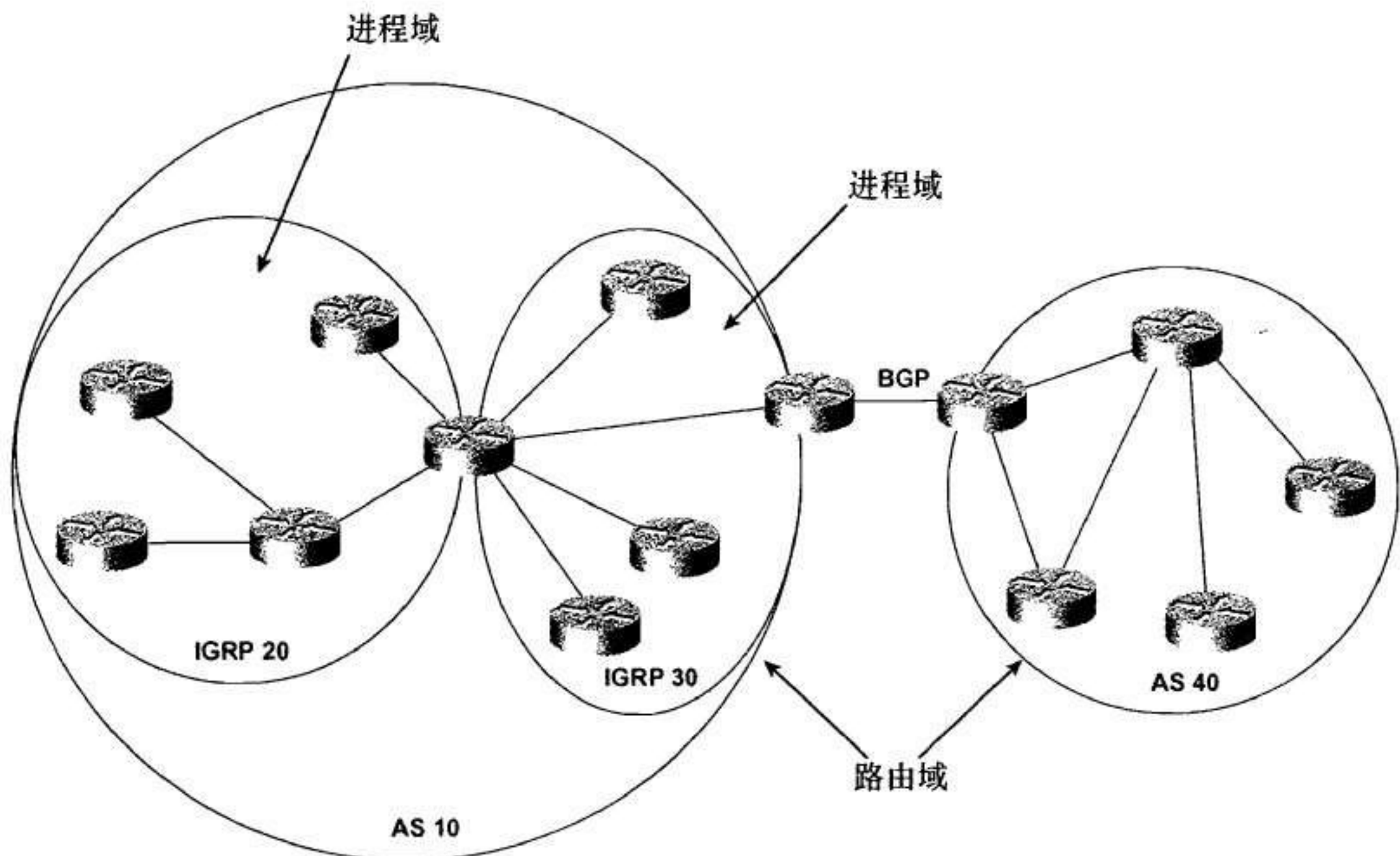


图 6-1 一个自主系统号可以指定一个路由选择域, 即在一个单一的管理域内运行一个或多个 IGP 协议的一组路由器集合。

一个自主系统号也可以指定一个进程域, 即依赖于一个单一的路由选择进程共享路由选择信息的一组路由器

在 IGRP 的路由更新报文中, IGRP 把路由条目分成 3 类: 内部路由 (Interior Route)、系统路由 (System Route) 和外部路由 (Exterior Route), 每个 IGRP 的路由条目都属于这 3 个类别中的一个。

- **内部路由 (Interior Route)** ——是指到达属于某个主网络的子网地址的路径, 这里的主网络是指正在广播这条路由更新的数据链路的主网络地址。换句话说, 作为内部路由被通告的子网对于通告路由器和接收路由器共同相连的主网络来说是“本地”的。
- **系统路由 (System Route)** ——是指到达在网络边界路由器上被汇总的网络地址的路径。
- **外部路由 (Exterior Route)** ——是到达被标记成缺省网络 (Default Network) 的路径。对于缺省网络, 路由器将直接发送所有的数据包而不对更具体的目的网络进行查找匹配。¹缺省网络和其配置方法将在第 12 章中讲述。

图 6-2 显示了 IGRP 协议是怎样使用这 3 类路由的。路由器 LeHand 和 Tully 都与子网 192.168.2.64/26 相连, 所以主网络 192.168.2.0 被认为是这两台路由器“公共”的“本地”网络。而路由器 LeHand 和子网 192.168.2.192/26 相连, 显然子网 192.168.2.192/26 是连接路由器 LeHand 和 Tully 的主网络 192.168.2.0 的另一个子网。因此, 路由器 LeHand 把子网 192.168.2.192/26 作为内部路由通告给路由器 Tully。

然而, 与路由器 LeHand 和路由器 Thompson 相连的本地网络却是 192.168.3.0。由于路

¹ 把缺省路由网络分类归到外部路由是 IGRP 和 EIGRP 协议所独有的。像 RIP 和 OSPF 等开放性的协议用地址 0.0.0.0 通告缺省路由网络。

由器 LeHand 是主网络 192.168.2.0 和网络 192.168.3.0 的边界路由器, 因此网络 192.168.2.0 被作为一条系统路由通告给路由器 Thompson, 同样, 网络 192.168.3.0 也被作为一条系统路由通告给路由器 Tully。

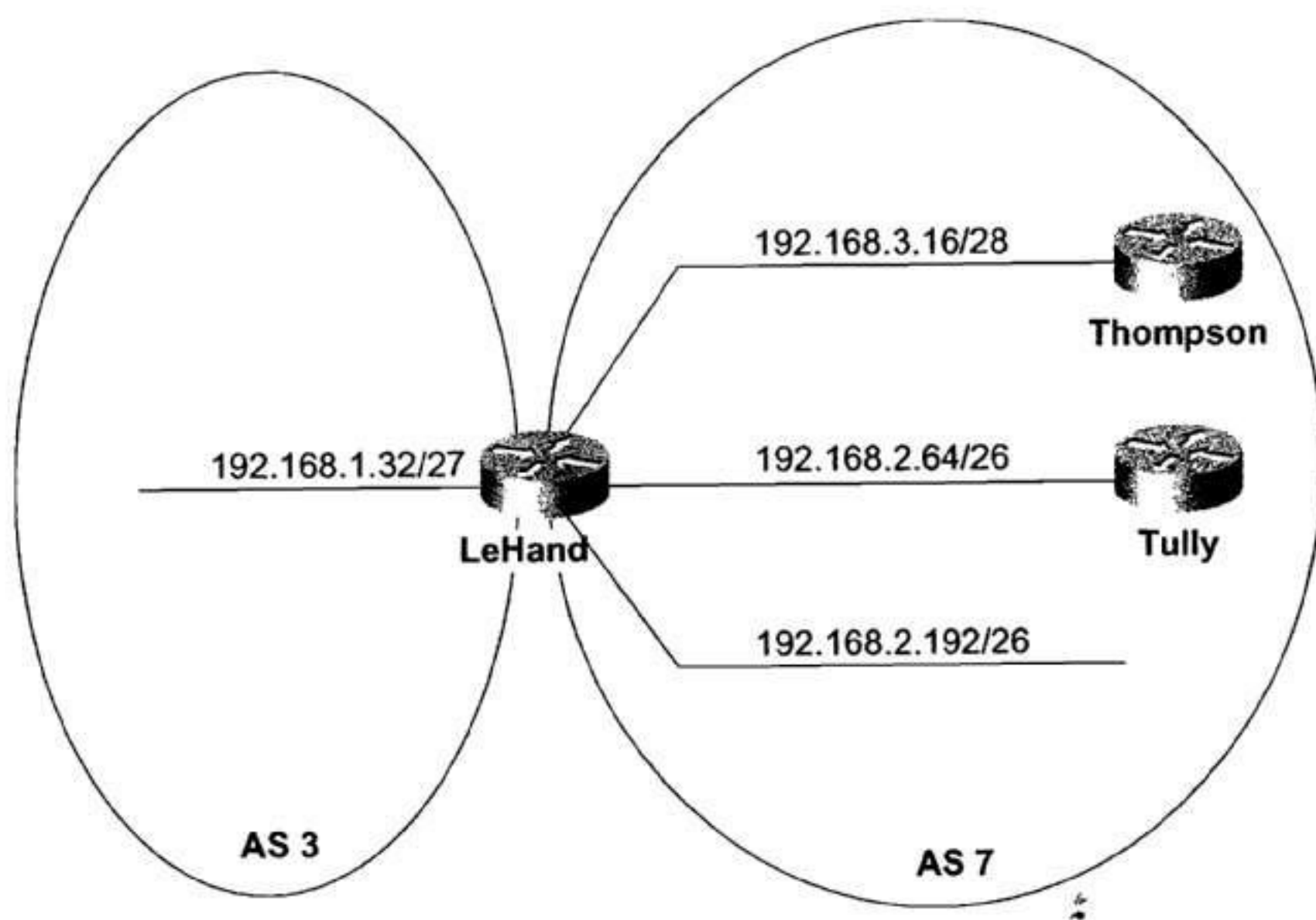


图 6-2 路由器 LeHand 把子网 192.168.2.192/26 作为一条内部路由通告给路由器 Tully, 把网络 192.168.3.0 作为一条系统路由通告给路由器 Tully, 并把网络 192.168.1.0 作为一条外部路由通告给路由器 Tully

网络 192.168.1.0 属于另一个自主系统, 因而路由器 LeHand 把它作为一条缺省网络地址来通告。因此, 网络 192.168.1.0 就被作为一条外部路由通告给路由器 Tully 和 Thompson。

6.1.1 IGRP 的计时器和稳定性

IGRP 协议的更新周期是 90s。为了防止更新计时器的同步, IGRP 针对每一个更新时间减掉一个最大为其 20% 的随机抖动变量。因此, 每个更新周期所需要的时间将在 72~90s 之间变化。

当一条路由首次被学到时, 这条路由的无效计时器就会被设置成 270s, 即更新周期时间的 3 倍长。同时, 刷新计时器设置成 630s, 即更新周期时间的 7 倍长。每次接收路由器收到该路由的更新报文后, 这些计时器都将被重新初始化。如果在收到一条更新报文之前无效计时器的计时超时了, 这条路由就会被标记成不可到达。但是, 在路由器的刷新计时器超时前, 这条路由还会被保留在路由选择表中, 并且作为不可达的路由通告出去, 如果刷新计时器超时了, 这条路由才会从路由选择表中删除掉。

和 RIP 协议时长为 30s 的计时器相比, IGRP 协议使用了 90s 的计时器, 这意味着 IGRP 协议的周期性更新将比 RIP 协议占用更少的带宽。然而, 代价是在同样的情况下, IGRP 协议比 RIP 协议的收敛更慢。例如, 如果一台路由器离线了, IGRP 协议需要 3 倍于 RIP 协议的时间才能检测到这个已不存在的邻居。

如果一条路由的目的地址变为不可达的, 或下一跳路由器增大了到达目的地址的度量以至于引起一个触发更新的话, 那么这条路由将会进入一个 280s (3 倍的更新周期加上 10s) 的抑制时间状态。直到抑制计时器超时之前, 有关这个目的地址新的信息都不会被路由器接

受。IGRP 协议的抑制特性可以用命令 **no metric holddown** 来禁止。在一个没有路由环路的网络拓扑中，抑制特性没有实际的意义，禁止掉这个特性将有助于减少 IGRP 的收敛时间。

缺省的计时器可以用下面的命令来改变：

```
timers basic update invalid holddown flush [sleeptime]
```

除了 **sleeptime** 选项，这条命令曾在改变 RIP 协议的计时器时使用过。**Sleeptime** 是一个周期性的毫秒 (ms) 级的计时器，在收到一条触发更新后，它被用来延迟一个正常的路由更新。

计时器的缺省值应当只在网络发生了明显的问题，并且仔细考虑了更改计时器所带来的后果之后才能加以改变。例如，在一个不稳定的网络拓扑中，可以缩减更新周期的时间来加速收敛，当然，代价是增加了更新报文的通信量，这将有可能在一条低带宽的链路导致拥塞，并且需要路由器增加大量的 CPU 周期来处理这些更新。这里需要注意，要确保在整个自主系统中统一地调整计时器，如果将来有任何路由器加入到这个自主系统里来，也必须确保它们的配置是更改后的计时器的值。

6.1.2 IGRP 的度量

在 IGRP 协议中，IGRP 将根据链路的特性计算出一个“复合”的度量值，这些链路特性是链路带宽、时延、负载和可靠性。缺省条件下，IGRP 选用路由路径的链路带宽和时延作为度量值。如果把数据链路想象成一个管道，那么带宽就像是这个管道的宽度，而时延就像是这个管道的长度。换句话说，带宽是数据传送能力的量度，而时延是端一端传送时间的量度。链路的另外两个特性——负载和可靠性只有在路由器上进行人工配置后才会被应用。虽然 IGRP 协议不使用 MTU (Maximum Transmission Unit, 最大传输单元) 作为计算复合度量值的参数，但 IGRP 也会跟踪每条路由路径上的最小 MTU 的大小。如图 6-3，可以通过命令 **show interfaces** 来观察一个特定接口上相关 IGRP 的复合度量的值大小。

```
Newfoundland#show interface fddi0
Fddi0 is administratively down, line protocol is down
Hardware is DAS FDDI, address is 00e0.1e8e.d1d9 (bia 00e0.1e8e.d1d9)
Internet address is 172.20.50.1/24
MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
Encapsulation SNAP, loopback not set, keepalive not set
ARP type: SNAP, ARP Timeout 04:00:00
Phy-A state is off, neighbor is Unknown, status no signal
Phy-B state is off, neighbor is Unknown, status no signal
ECM is out, CFM is isolated, RMT is isolated
Requested token rotation 5000 usec, negotiated 5017 usec
Configured tvx is 3400 usec, using 5242.90 usec, ring not operational
0 SMT frames processed, 0 dropped, 20 SMT buffers
Upstream neighbor 0000.f800.0000, downstream neighbor 0000.f800.0000
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Queuing 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
```

待续


```

0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
2 transitions, 0 traces
Newfoundland#

```

图 6-3 **show interface** 命令的输出包含了关于这个 FDDI 接口的度量的统计。这个 FDDI 接口的度量值显示,

MTU=4470 字节, 带宽=100Mbit/s, 时延=100μs, 可靠性=100%, 负载=.39% (最小负载)

1. 带宽 (Bandwidth)

带宽用 kbit/s 单位来表示, 它在计算链路的度量值时仅作为一个静态的值, 没有必要反映出链路实际使用的带宽, 也就是说, 带宽不需要动态地去量度。例如, 不论和串行接口相连的链路是 T1 的还是 56K 的, 串行接口的缺省带宽都是 1544kbit/s。这个缺省的带宽值可以通过 **bandwidth** 命令来更改。

IGRP 的更新报文使用 3 个 8bit 字节来表示 IGRP “带宽”, 在本书里用 BW_{IGRP} 表示, 它是用因子 10^7 除以带宽得来的, 因此, 如果接口的带宽是 1544, 那么

$$BW_{IGRP} = 10^7 / 1544 = 6476, \text{ 或 } 0x00194C.$$

2. 时延 (Delay)

时延, 像带宽一样, 也是一个静态特征的度量值, 不需要动态地去量度。时延可以通过 **show interface** 命令显示的 DLY 参数来表示, 单位是 μs (微秒)。一个接口的缺省时延可以通过命令 **delay** 进行更改, 并以 10μs 作为命令配置的最小计量单位。图 6-4 显示了用 **bandwidth** 和 **delay** 命令更改图 6-3 中的缺省带宽和时延的情况。

```

Newfoundland(config)#interface fddi0
Newfoundland(config-if)#bandwidth 75000
Newfoundland(config-if)#delay 5
Newfoundland(config-if)#^Z
Newfoundland#
%SYS-5-CONFIG_I: Configured from console by console
Newfoundland#show interface fddi0
Fddi0 is administratively down, line protocol is down
  Hardware is DAS FDDI, address is 00e0.1e8e.d1d9. (bia 00e0.1e8e.d1d9)
  Internet address is 172.20.50.1/24
  MTU 4470 bytes, BW 75000 Kbit, DLY 50 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive not set
  ARP type: SNAP, ARP Timeout 04:00:00
  Phy-A state is off, neighbor is Unknown, status no signal
  Phy-B state is off, neighbor is Unknown, status no signal
  ECM is out, CFM is isolated, RMT is isolated
  Requested token rotation 5000 usec, negotiated 5017 usec
  Configured txx is 3400 usec, using 5242.90 usec, ring not operational
  0 SMT frames processed, 0 dropped, 20 SMT buffers
  Upstream neighbor 0000.f800.0000, downstream neighbor 0000.f800.0000
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never

```

待续


```
Queuing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 collisions, 2 interface resets
  0 output buffer failures, 0 output buffers swapped out
  2 transitions, 0 traces
```

图 6-4 通过 **bandwidth** 和 **delay** 命令改变了接口 FDDI0 的缺省度量值，使用 **show interface** 命令可以查看更改后的新度量值

在 IGRP 的更新报文中，时延也是用 3 个 8bit 字节来表示，并可以通过命令 **delay** 来改变，同样，是以 10μs 作为命令设置的最小计量单位。为了避免误解，这个数值用 DLY_{IGRP} 来表示，以便区别于 DLY，DLY_{IGRP} 可以通过命令 **show interface** 来观察，单位是 μs（微秒）。例如，假如 DLY 的值是 50，那么

$DLY_{IGRP} = DLY / 10 = 50 / 10 = 5$ ，或 0x000005。

IGRP 通过设定 DLY_{IGRP}=0xFFFFF 来标识一条不可到达的路由路径，这个数值大约为 167.8s，因此，一条 IGRP 路由端一端的最大时延是 167s。

因为 IGRP 协议使用带宽和时延来作为它的缺省度量值，因此，这些特性参数必须配置正确，并且要在所有的 IGRP 路由器的接口上统一规划配置。除非有一个很好的理由，并且对更改这些参数的配置后产生的结果有个清楚地理解，否则最好不要更改一个接口的带宽和时延参数。在大多数的情况下，最好保留使用这些参数的缺省值而不要加以改变。有一个值得注意的例外就是串行接口，正如本节前面所提及的，在 Cisco 路由器上，不管串行接口相连的是什么带宽的数据链路，它都会把串行接口的缺省带宽设置为 1544。这时，应该使用命令 **bandwidth** 在接口上设置链路的实际带宽。

这里请注意，很重要的一点是在 OSPF 协议中也使用带宽来计算它的度量值。因此，在一个同时运行 IGRP 和 OSPF 协议的互联网络中，如果要改变 IGRP 的度量值，应该使用 **delay** 来影响 IGRP 的度量值，如果改变带宽将会同时影响到 IGRP 和 OSPF。

表 6-1 中列出了一些常用接口的带宽和时延（注意，串行接口的缺省带宽总是 1544；图 6-1 显示了使用 **bandwidth** 命令来反映实际相连的链路带宽的效果）。

表 6-1 常用 BW_{IGRP} 和 DLY_{IGRP} 的数值

介 质	带宽 (Bandwidth)	BW _{IGRP}	时延 (Delay)	DLY _{IGRP}
100M ATM	100000kbit/s	100	100μs	10
快速以太网 (Fast Ethernet)	100000 kbit/s	100	100μs	10
FDDI	100000 kbit/s	100	100μs	10
HSSI	45045 kbit/s	222	20000μs	2000
16M 令牌环 (Token Ring)	16000 kbit/s	625	630μs	63
以太网 (Ethernet)	10000 kbit/s	1000	1000μs	100
T1	1544 kbit/s	6476	20000μs	2000

续表

介 质	带宽 (Bandwidth)	BW _{IGRP}	时延 (Delay)	DLY _{IGRP}
DS0	64 kbit/s	156250	20000μs	2000
56K	56 kbit/s	178571	20000μs	2000
Tunnel	9 kbit/s	1111111	500000μs	50000

3. 可靠性 (Reliability)

可靠性是一个动态量度的度量参数, 它使用一个 8 位的数字来表达, 255 表示 100% 的可靠链路, 而 1 表示最低可靠的链路。在命令 **show interface** 的输出中, 可靠性被表示成 255 的分数, 例如, 234/255 (图 6-5)。

```
Casablanca#show interface ethernet0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c76.5b7c (bia 0000.0c76.5b7c)
  Internet address is 172.20.1.1 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 234/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:28, output 0:00:06, output hang never
  Last clearing of "show interface" counters 0:06:05
  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
    22 packets input, 3758 bytes, 0 no buffer
    Received 21 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 input packets with dribble condition detected
    125 packets output, 11254 bytes, 0 underruns
    39 output errors, 694 collisions, 0 interface resets, 0 restarts
    0 output buffer failures, 0 output buffers swapped out
Casablanca#
```

图 6-5 这个接口显示了该接口的可靠性是 234/255, 或 91.8%

4. 负载 (Load)

在 IGRP 的更新里, 负载是一个 8 位的数字。在 **show interface** 命令的输出中表示成一个 255 的分数, 例如, 40/255 (图 6-6); 1 表示最小的负载链路, 255 表示 100% 的负载链路。

如果可靠性或负载被用来作为一个度量或复合度量的一部分, 计算度量的算法应当不允许在出错的比率或信道的占用突然发生变化来影响度量值, 以免造成互联网络的不稳定。举一个例子来说明, 如果在一个可能出现突发变化的网络上, 突发的大流量可能会导致路由进入抑制 (holddown) 状态, 而通信量的突然降低可能会触发一个更新。为了防止度量频繁的改变, 可靠性和负载是基于 5min 时间常数的指数加权平均计算的, 它们每 5s 被更新一次。

对于每个 IGRP 路由的复合度量, 可以用下面的公式计算:

$$\text{metric} = [k1 * BW_{IGRP(\min)} + (k2 * BW_{IGRP(\min)}) / (256 - \text{LOAD}) + k3 * DLY_{IGRP(\text{sum})}] * [k5 / (\text{RELIABILITY} + k4)]$$

在这里, $BW_{IGRP(\min)}$ 是沿着路由路径到达目的网络的所有出站接口的 BW_{IGRP} 带宽中的最小值, 而 $DLY_{IGRP(\text{sum})}$ 是这条路由路径 DLY_{IGRP} 时延的总和。

系数 $k1$ 到 $k5$ 是可配置的加权值, 它们的缺省值是: $k1=k3=1$, $k2=k4=k5=0$ 。这些缺省值可以通过下面的命令来更改:


```
Yalta#show interface serial 1
Serial1 is up, line protocol is up
Hardware is HD64570
Internet address is 172.20.20.2 255.255.255.0
MTU 1500 bytes, BW 56 Kbit, DLY 20000 usec, rely 255/255, load 40/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 0:00:08, output 0:00:00, output hang never
Last clearing of "show interface" counters 0:05:05
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 10000 bits/sec, 1 packet/sec
5 minute output rate 9000 bits/sec, 1 packet/sec
  456 packets input, 397463 bytes, 0 no buffer
  Received 70 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  428 packets output, 395862 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets, 0 restarts
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

图 6-6 这个接口显示了一个 40/255 的负载链路，或 15.7%

metric weights tos k1 k2 k3 k4 k5¹

如果 k5 被设置为 0，则 $[k5/(RELIABILITY+k4)]$ 这一项将不使用。²

使用 k1~k5 给定的缺省值，IGRP 的复合度量的计算公式将简化成缺省的度量：

$\text{metric} = BW_{IGRP(\min)} + DLY_{IGRP(\text{sum})}$ 。

图 6-7 中的例子显示了网络的每个路由器接口的带宽和时延的配置，以及计算出 IGRP 路由度量的路由器之一的路由转发表。³

路由选择表本身只显示了已经计算出来的度量，如果需要查看 IGRP 记录的每条路由实际的参数值，可以用 **show ip route address** 命令，如图 6-8。这里路由器 Casablanca 到子网 172.20.40.0/24 的路由路径上的最小带宽是 512kbit/s，最小带宽的链路接口位于路由器 Quebec 的出站接口。该路由时延的总和是 $(1000+20000+20000+5000)=46000\mu\text{s}$ 。

$$BW_{IGRP(\min)} = 10^7 / 512 = 19531$$

$$DLY_{IGRP(\text{sum})} = 46000 / 10 = 4600$$

$$\text{metric} = BW_{IGRP(\min)} + DLY_{IGRP(\text{sum})} = 19531 + 4600 = 24131$$

图 6-8 也显示了 IGRP 记录的沿着路由路径上最小的 MTU 和跳数。MTU 不用来作度量的计算。跳数是下一跳路由器报告的跳数，仅仅用来限制网络规模的口径大小。缺省条件下，最大的跳数是 100，也可以通过命令 **metric maximum-hops** 配置成 1~255 之间的数值。如果一条路由超过了设置的最大跳数，那么它的时延将被设置成 0xFFFFFFFF，而变成一条不可达的路由。

注意：这里所有的度量都是基于沿着路由路径方向的路由器出站接口计算的。例如，路由器 Yalta 到达子网 172.20.40.0/24 的路由度量是不同于路由器 Casablanca 到达子网 172.20.40.0/24

¹ tos 是 Cisco 公司最早打算使用 IGRP 协议来提供支持服务类型的路由选择时遗留下来的参数，但这个计划从来没有被采纳过，因而 tos 的值总是被设定为 0。

² 译者注：原来的度量计算公式将变为 $\text{metric} = k1 * BW_{IGRP(\min)} + (k2 * BW_{IGRP(\min)}) / (256 - \text{LOAD}) + k3 * DLY_{IGRP(\text{sum})}$ 。

³ 这里也要注意，IGRP 协议的管理距离是 100。

的路由度量的。这是因为, 路由器 Yalta 和路由器 Quebec 之间的链路带宽的配置不同, 而且到达这两个子网的出站接口上时延也不同。

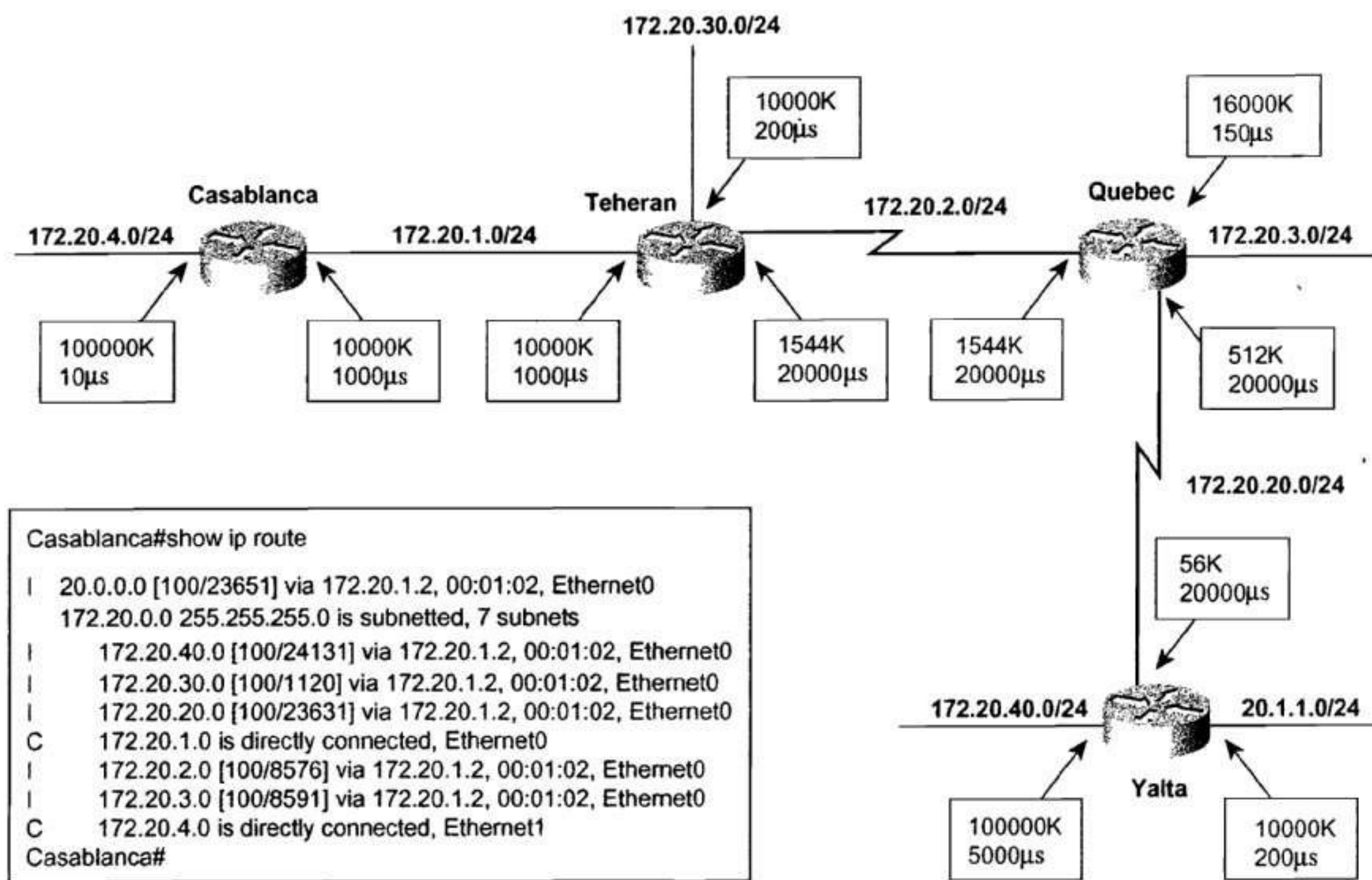


图 6-7 缺省条件下, IGRP 的度量是由时延的总和加上最小的带宽计算得来的

```

Casablanca#show ip route 172.20.40.0
Routing entry for 172.20.40.0 255.255.255.0
Known via "igrp 1", distance 100, metric 24131
Redistributing via igrp 1
Advertised by igrp 1 (self originated)
Last update from 172.20.1.2 on Ethernet0, 00:00:54 ago
Routing Descriptor Blocks:
* 172.20.1.2, from 172.20.1.2, 00:00:54 ago, via Ethernet0
Route metric is 24131, traffic share count is 1
Total delay is 46000 microseconds, minimum bandwidth is 512 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 2
  
```

图 6-8 路由器 Casablanca 到子网 172.20.40.0/24 的路由的度量由 512kbit/s 的最小带宽和 46000μs 的路由时延总和计算得出

6.1.3 IGRP 的报文格式

IGRP 协议的报文格式如图 6-9 所示。相对于图 5-3 中 RIP 协议的报文格式, 很显然这是一个高效率的设计。同时, IGRP 更新提供了比 RIP 协议更多的信息, 它可以发送比发送者路由选择表的快照 (snapshot) 更多的信息。IGRP 报文的所有字段都被使用了, 在 12 个 8bit 字节长的报文头后跟着相继出现的就是一条条单个的路由条目。与 RIP 协议对比, IGRP 协议不用填充无用的数据给每个路由条目以使它们达到 32bit 字的边界。每个更新报文可以携带最大 104 个路由条目, 每个路由条目的大小是 14 个 8bit 字节。因此, 加上 12 个 8bit 字节

的报文头，一个最大的 IGRP 报文的大小是 $12+(104\times14)=1468$ 个 8bit 字节，再加上一个 32 个 8bit 字节的 IP 头部，一个最大的 IGRP 报文的大小增大到 1500 字节。

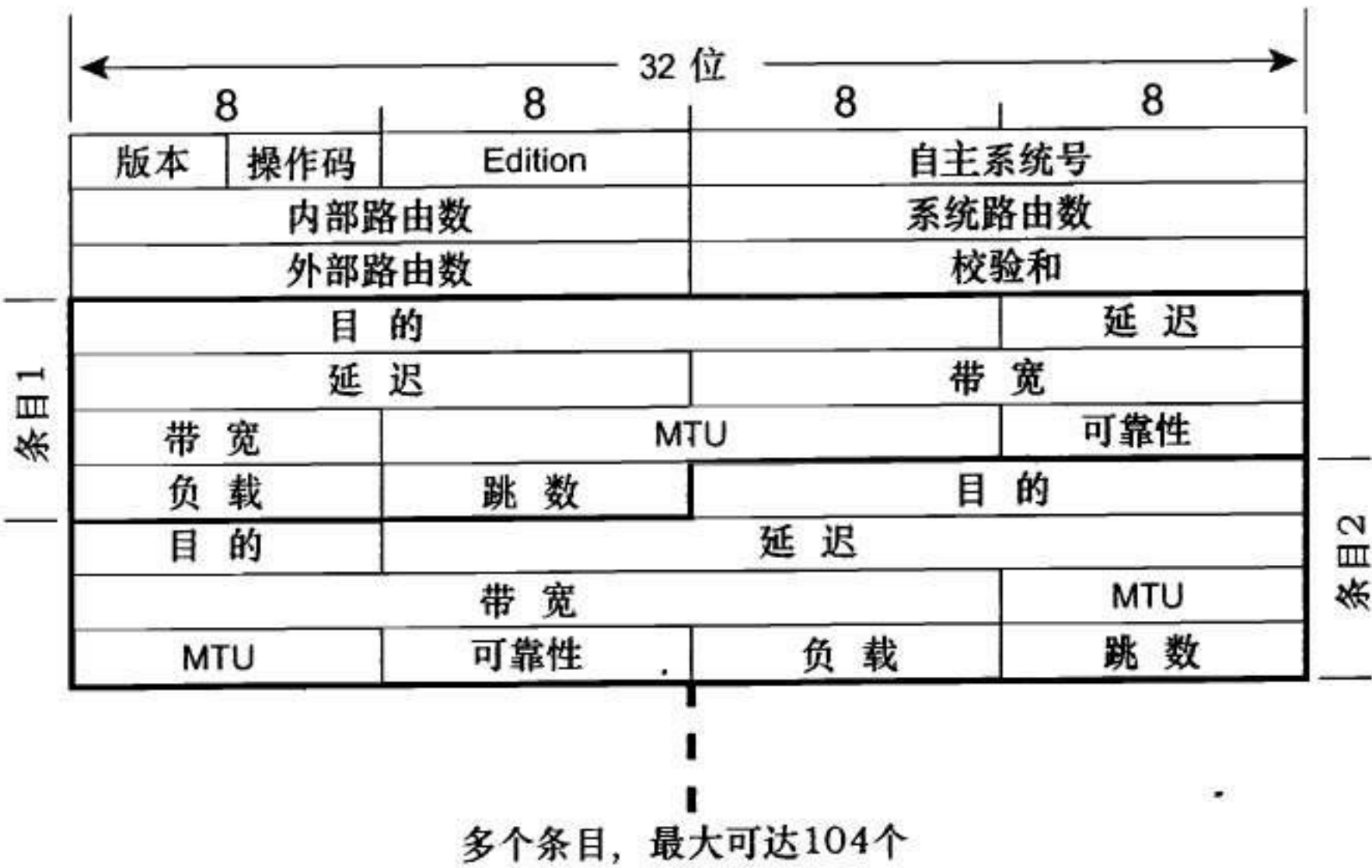


图 6-9 IGRP 的报文格式

- 版本号 (**Version**) ——永远设置为 1。
- 类型代码 (**Opcode**) ——1 标识是一个 IGRP 请求报文，2 标识是一个 IGRP 更新报文。一个请求报文可以是一个不带路由条目的报文头组成。
- 更新版本 (**Edition**) ——只要路由选择信息发生变化时更新的发送者就会增大这个参数值。更新版本号可以帮助路由器在接收新的路由更新后，避免再接收老的路由更新。
- 自主系统号 (**Autonomous System Number**) ——确切地说是 IGRP 进程的 ID 号。这个标记允许多个 IGRP 进程在同一条数据链路上交换路由信息。
- 内部路由数 (**Number of Interior Routes**) ——在路由更新中，直连网络的子网条目的个数，如果直连网络没有子网，这个字段就设置为 0。内部路由条目总是在更新中首先显示出来。这个字段，和后面紧接的系统路由数字段以及外部路由数字段一起，可以告诉 IGRP 进程有多少个 14 个 8bit 字节的条目包含在报文里，因此可以计算出报文的长度。
- 系统路由数 (**Number of System Routes**) ——指出了那些非直连网络的路由条数 —— 换句话说，就是在边界路由器上被汇总的路由条数。如果有的话，路由条数计入这个字段，并跟在内部路由条目的后面。
- 外部路由数 (**Number of Exterior Routes**) ——指出了那些到达被标识为缺省网络的路由条数。如果有的话，路由条数计入这个字段，并出现在路由更新的最后。
- 校验和 (**Checksum**) ——校验和是基于 IGRP 报文头部和所有的路由条目的信息计算的。为了计算校验和，这个校验字段首先被设置为 0，并且计算这个报文的 16 位 1 的补码和 (不含 IP 头部)，然后把计算所得的 16 位 1 的补码和存储在这个校验和字段中，这样一直到这个报文被接收后将再次计算这个 16 位 1 的补码，这次计算要包含被传送的校验和字段。当在一个报文被无差错地传送时，执行校验和计算后结果应该是 16 位全 1 (0xFFFF)。

- **目的地址 (Destination)** —— 是每个路由条目的第一个字段。乍一看, 这个字段好像不合常规, 对于给定 4 个 8bit 字节的 IP 地址, 它只有 3 个 8bit 字节的长度。但是事实上, 由于 IGRP 路由分类的原因, 结果发现 3 个 8bit 字节的目的地址也可以被辨认出来。假如一个路由条目是内部路由, 那么至少 IP 地址的第 1 个 8bit 字节可以通过接收路由更新的接口的地址得知, 因此, 内部路由条目的目的地址字段只需要包含 IP 地址的最后 3 个 8bit 字节即可。同样, 如果一个路由条目是系统路由和外部路由, 该路由将进行路由汇总, 因而这条路由的目的地址至少最后一个 8bit 字节将全部为 0。因此, 系统路由和外部路由的目的地址字段将只需要包含目的地址的开始 3 个 8bit 字节。

举一个例子, 假设在接口 172.20.1.1/24 上收到一条目的地址字段为 20.40.0 的内部路由, 那么将会认为这条内部路由的地址是子网 172.20.40.0/24。同样, 如果系统路由 192.168.14 和 20.0.0 被收到, 那么 IGRP 将会辨认出这些目的地址是主网络 192.168.14.0 和 20.0.0.0。

- **时延 (Delay)** —— 就是前面讲述的 24 位的 $DLY_{IGRP(sum)}$, 是以 10 μ s 为单位进行配置的时延总和。
- **带宽 (Bandwidth)** —— 就是前面讲述的 24 位的 $BW_{IGRP(min)}$, 用 10 000 000 除以沿此路由路径的所有接口上配置的最小带宽。
- **最大传输单元 (MTU)** —— 是沿此路由到达目的地址的路径上所有链路中最小的 MTU。虽然它也是一个包括在内的参数, 但是从来没有在 IGRP 协议的度量计算中使用过。
- **可靠性 (Reliability)** —— 是一个 0x01~0xFF 之间的数字, 它反映了沿此路由路径上的接口出站数据包的出错比率的总和, 使用一个每 5min 计量一次的指数加权平均值来计算。

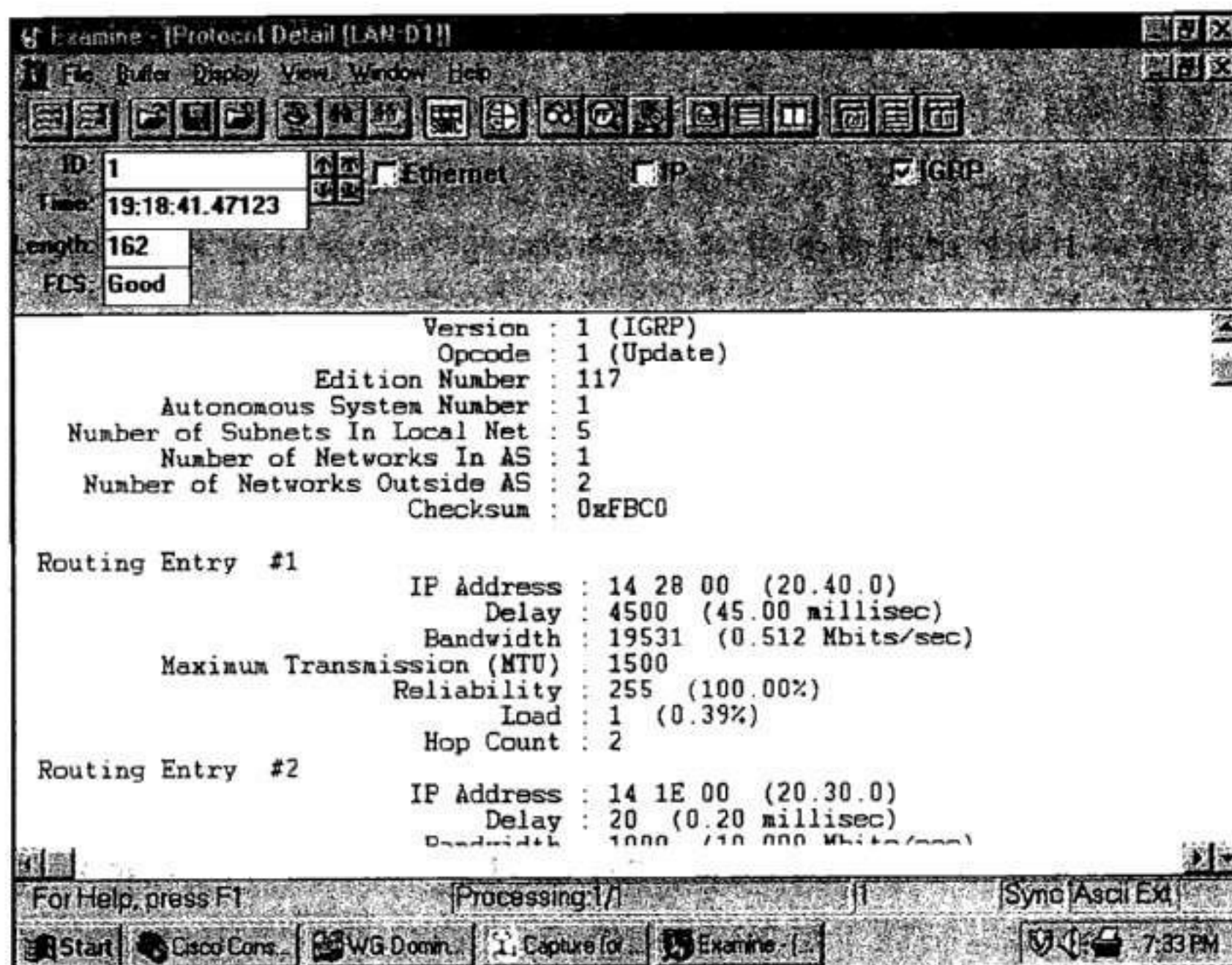


图 6-10 可以在协议分析解码器里看出一个 IGRP 更新报文的头部和第一个路由条目

- **负载 (Load)** ——也是一个在 0x01~0xFF 之间的数字，它反映了沿此路由路径上的接口出站负载的总和，同样使用一个每 5min 计量一次的指数加权平均值来计算。
- **跳数 (Hop Count)** ——是一个表示到达目的地址跳数的数字，数值范围是 0x01~0xFF。路由器使用“0 跳”通告直连的网络，其后相继的路由器将记录和通告相对于下一跳路由器的路由。例如，在图 6-8 中的路由器 Casablanca 显示了到达子网 172.20.40.0 是两跳。图 6-7 显示了跳数的含义：子网 172.20.40.0 从下一跳路由器 Teheran 算起是两跳。

在图 6-10 中显示了在协议分析仪上观察到的一个 IGRP 更新报文的部分解码。

6.2 配置 IGRP

相比 RIP 协议而言，虽然 IGRP 协议有更多一些的配置选项可用，但是基本的配置都是相当简单的：**router** 命令用来创建一个路由选择进程，**network** 命令用来指定哪些网络需要运行 IGRP。和 RIP 协议一样，IGRP 协议也是个有类别路由选择协议，因而也只能指定主网络号。

neighbor 命令用来发送单播更新，而 **passive-interface** 命令用来防止在选定的子网上广播更新，这在第 5 章“路由选择信息协议 (RIP)”中已经介绍，它们可以像在 RIP 中一样使用。

Offset-list 命令也已经在第 5 章中介绍过。当在 IGRP 协议中使用时，偏移变量使用 *delay* 替代了 *hops*。

IGRP 和 RIP 有一个重要的不同之处，就是 IGRP 使用进程 ID 号 (process ID)，因而允许多个 IGRP 进程运行在同一台路由器上。

6.2.1 案例研究 1：一个基本的 IGRP 配置

配置 IGRP 只需要两个必要的步骤：

步骤 1：使用 **router igrp process-id** 命令启动 IGRP 进程；

步骤 2：使用 **network** 命令来指定运行 IGRP 协议的每个主网络。

进程 ID 号是由更新报文的 16 位的自主系统字段来传送的。进程 ID 号的选择是任意的。如果使用某个具体的 IGRP 进程的所有路由器必须共享路由信息，那么那个 IGRP 进程的 ID 号应当是一致的，它可以是 1~65535 (0 不允许使用) 之间的任何一个数字。图 6-11 显示了一个简单的互联网络，图中 3 台路由器的配置如下：

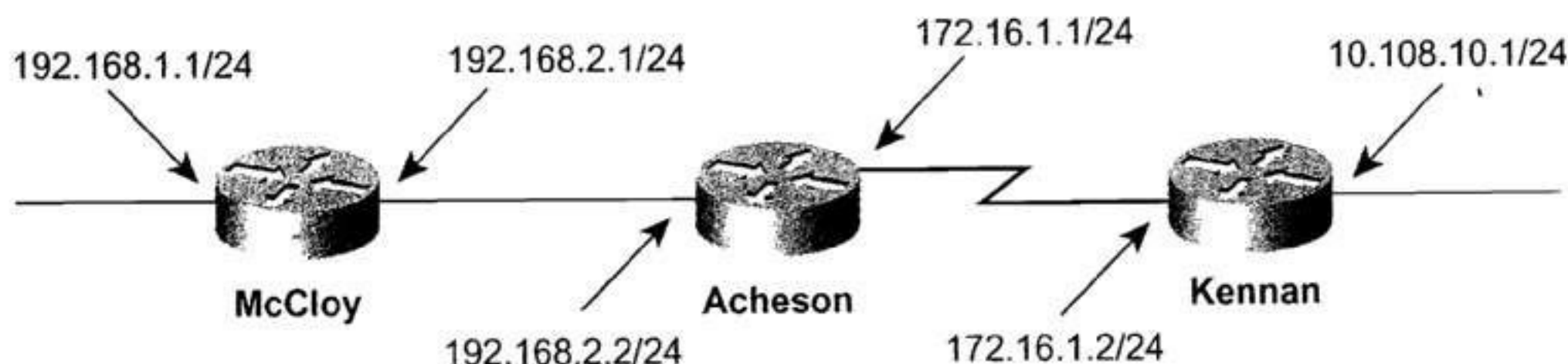


图 6-11 IGRP 将在这 3 台网络边界路由器上进行地址汇总


```

McCloy(config)#router igrp 10
McCloy(config-router)#network 192.168.1.0
McCloy(config-router)#network 192.168.2.0

Acheson(config)#router igrp 10
Acheson(config-router)#network 192.168.2.0
Acheson(config-router)#network 172.16.0.0

Kennan(config)#router igrp 10
Kennan(config-router)#network 172.16.0.0
Kennan(config-router)#network 10.0.0.0

```

IGRP 协议在网络边界上执行子网屏蔽或路由汇总。在图 6-11 的例子中, 这 3 台路由器全都是网络边界路由器。

6.2.2 案例研究 2: 非等价负载均衡 (一)

在和 RIP 协议同样的快速交换或处理交换 (fast/process switching) 转发机制的限制下, IGRP 可以在给定最多 6 条等价路由¹上执行等价负载均衡。但与 RIP 协议不同的是, IGRP 协议也可以执行非等价负载均衡。图 6-12 中, 在路由器 Acheson 和路由器 Kennan 中间增加了一条额外的串行接口, 它的带宽配置为 256kbit/s。这样做的目的, 是为了使路由器 Acheson 可以执行非等价负载均衡, 即在这两条链路之间根据它们的链路度量大小的反比来分配这两条链路上数据流量的负载大小。

察看路由器 Acheson 通过 S0 接口到达网络 10.0.0.0 的路由, 可以看出最小的带宽是 1544 kbit/s (假定路由器 Kennan 的以太网接口使用的缺省带宽为 10 Mbit/s)。根据表 6-1 中所标明的, 串行接口和以太网接口的 $DLY_{IGRP(sum)}$ 为 $2000+100=2100$, $BW_{IGRP(min)}$ 为 $10^7/1544=6476$, 因此, 这条路由的复合度量值是 $6476+2100=8576$ 。

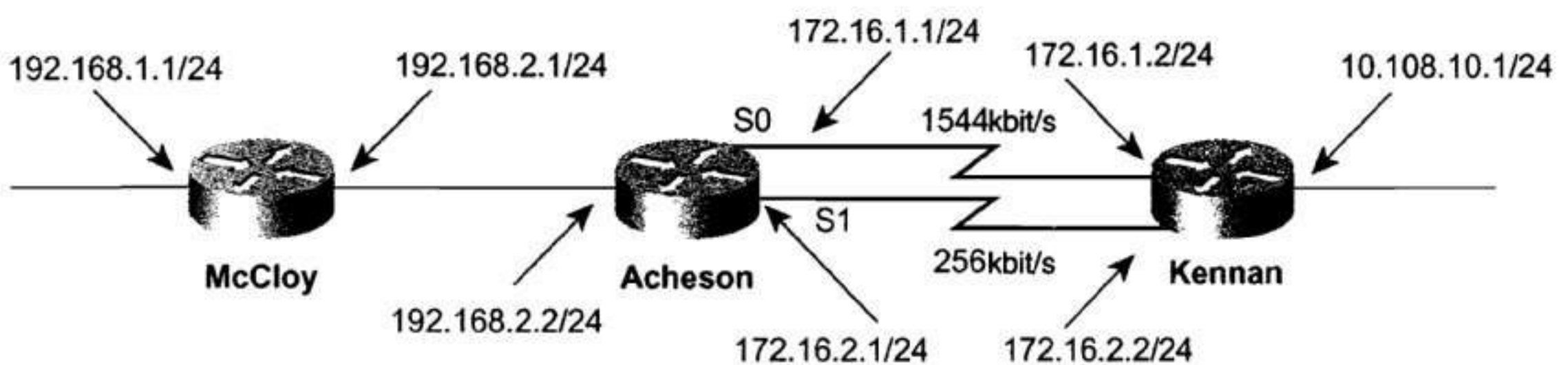


图 6-12 可以配置 IGRP 协议在链路之间运行非等价负载均衡, 例如路由器 Acheson 和 Kennan 之间的链路

路由器 Acheson 通过 S1 接口到达网络 10.0.0.0 的最小带宽是 256kbit/s。它的 $DLY_{IGRP(sum)}$ 与上面的第一条路由相同, 因此, 这条路由的复合度量值是 $10^7/256+2100=41162$ 。不做进一步的配置, IGRP 协议将会简单地选择路径代价最小的路径。图 6-13 显示了路由器 Acheson 仅仅使用度量值为 8576 的那条路径。

差异变量 (Variance)

variance 命令用来确定哪些路由路径在非等价负载均衡中是可以使用的。**Variance** 定义

¹ 缺省的路径条数是 4 条, 可以参看设置最大的路径条数的案例研究来获取进一步的详细信息。

了一个倍数因子,用来表示一条路由路径的度量值和最小代价路由路径的不同或差别的程度。任何路由路径的度量值如果超过了最小代价路由路径的度量乘以 **Variance** 的值,那么这条路由路径将不被使用。

```
Acheson#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
       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

Gateway of last resort is not set

I 10.0.0.0 [100/8576] via 172.16.1.2, 00:00:06, Serial0
I 192.168.1.0 [100/1600] via 192.168.2.1, 00:00:06, Ethernet0
C 192.168.2.0 is directly connected, Ethernet0
  172.16.0.0 255.255.255.0 is subnetted, 2 subnets
C    172.16.1.0 is directly connected, Serial0
C    172.16.2.0 is directly connected, Serial1
Acheson#
```

图 6-13 路由器 Acheson 仅仅使用最小路径代价的链路到达网络 10.0.0.0。要运行非等价负载均衡则需要额外的配置

Variance 的缺省值是 1, 这意味着如果要想实现负载均衡, 那么多条路由路径的度量值必须是相同的。Variance 必须是整数。

路由器 Acheson 通过 S1 接口的路由路径的度量值是通过 S0 接口的路由路径的 $41162/8576=4.8$ 倍。因此, 为了使路由器 Acheson 可以实现非等价负载均衡, 路由器 Acheson 上的 variance 的值应该是 5。IGRP 的配置如下:

```
router igrp 10
 network 172.16.0.0
 network 192.168.2.0
 variance 5
```

在路由器 Acheson 上将 variance 指定为 5 以后, 它的路由选择表就包含了第二条代价较高的路由 (图 6-14)。在非等价负载均衡中的路由经常会碰到以下 3 种情况:

```
Acheson(config)#router igrp 10
Acheson(config-router)#variance 5
Acheson(config-router)#^Z
Acheson#
%SYS-5-CONFIG_I: Configured from console by console
Acheson#clear ip route *
Acheson#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
       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

Gateway of last resort is not set

I 10.0.0.0 [100/8576] via 172.16.1.2, 00:00:07, Serial0
      [100/41162] via 172.16.2.2, 00:00:07, Serial1
I 192.168.1.0 [100/1600] via 192.168.2.1, 00:00:07, Ethernet0
C 192.168.2.0 is directly connected, Ethernet0
  172.16.0.0 255.255.255.0 is subnetted, 2 subnets
C    172.16.1.0 is directly connected, Serial0
C    172.16.2.0 is directly connected, Serial1
Acheson#
```

图 6-14 到达网络 10.0.0.0 的第二条路径的复合度量值是 41162, 或者说是最小代价的路由路径的 4.8 倍。如果 variance 的值设置为不小于 5 的话, 那么 IGRP 协议将把第二条路径加入到路由选择表中

(1) 增加到负载共享“组”中的路由路径条数不能超过最大路径条数 (maximum-paths) 的限制。

(2) 下一跳路由器必须在度量上更接近目的网络。换句话说, 在下一跳路由器上到达目的网络的度量值必须小于本地路由器到达该目的网络的度量值。到达目的网络更近的下一跳路由器, 通常被称为下游路由器 (Downstream Router)。

(3) 最小路径代价的路由的度量值乘以 variance 后, 必须大于所增加的非最小代价路由的度量值。

关于按每目的地 (per destination) 和按每数据包 (per packet) 进行负载均衡的规则, 已经在第 3 章“静态路由”中讨论了, 同样也适用于这里。如果数据包转发是快速交换 (fast switching) 的, 就按照每目的地进行负载均衡; 如果数据包转发是处理交换 (process switching) 的, 就按照每数据包进行负载均衡。图 6-15 显示了从路由器 Acheson 发出 20 个 ping 包的调试输出结果, 在这里快速交换已经通过命令 **no ip route-cache** 关闭了, 因此路由器将执行按每数据包的非等价负载均衡。每 5 个数据包通过 1544kbit/s 的链路 (下一跳是 172.16.1.2) 发送过后, 就会有 1 个数据包通过 256kbit/s 的链路 (下一跳是 172.16.2.2) 发送, 这和这两条路径大约 5:1 的度量比值是相对应的。

```
Acheson#debug ip packet
IP packet debugging is on
Acheson#
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial1), g=172.16.2.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial1), g=172.16.2.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial1), g=172.16.2.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
IP: s=192.168.2.1 (Ethernet0), d=10.108.10.1 (Serial0), g=172.16.1.2, forward
Acheson#
```

图 6-15 路由器将执行按每数据包的负载均衡, 即每通过低代价的链路发送 5 个数据包, 就会通过高代价的链路发送 1 个数据包

如果 **variance** 设置为 1, 那么 IGRP 将只会把到达目的网络最小代价的路由加入到它的路由选择表中。然而, 在一些情况下, 例如, 有时为了减小收敛的时间或帮助故障排错, 即使没有发生负载均衡, 也要将所有可用的路由都加入到路由选择表中。所有的数据包应该使用路径代价最小的路由, 只有当主路径失效时, 才被切换到下一个最好的路径。这里有一个隐含的缺省命令 **traffic-share balanced** (也就是说, 这个命令存在, 但是不需要保留在配置文件里面)。在路由选择表中存在多条路径时, 为了使路由器只使用最小代价的路径, 可以把缺省的配置改成 **traffic-share min**。如果有多条相等的最小代价的路径, 并且配置了

traffic-share min, 那么 IGRP 将执行等价负载均衡。

6.2.3 案例研究 3: 设置最大的路径数

IGRP 协议可以进行负载均衡的路由路径的最大条数可以用 **maximum-paths** 命令来设置。在 11.0 版及后续版本的 IOS 软件中, 路径条数可以是 1~6 之间的任何值, 而在这之前的早期版本中路径条数可以是 1~4 之间的任何值。所有版本的缺省值是 4。

图 6-16 显示了从路由器 McCloy 到达网络 172.18.0.0 的 3 条不同代价的并行路由。网络管理员可能仅仅希望在这些路由中的两条路由上进行负载均衡, 只有当这些路由中的任何一条被确认失效时, 才使用第 3 条路由替代它。

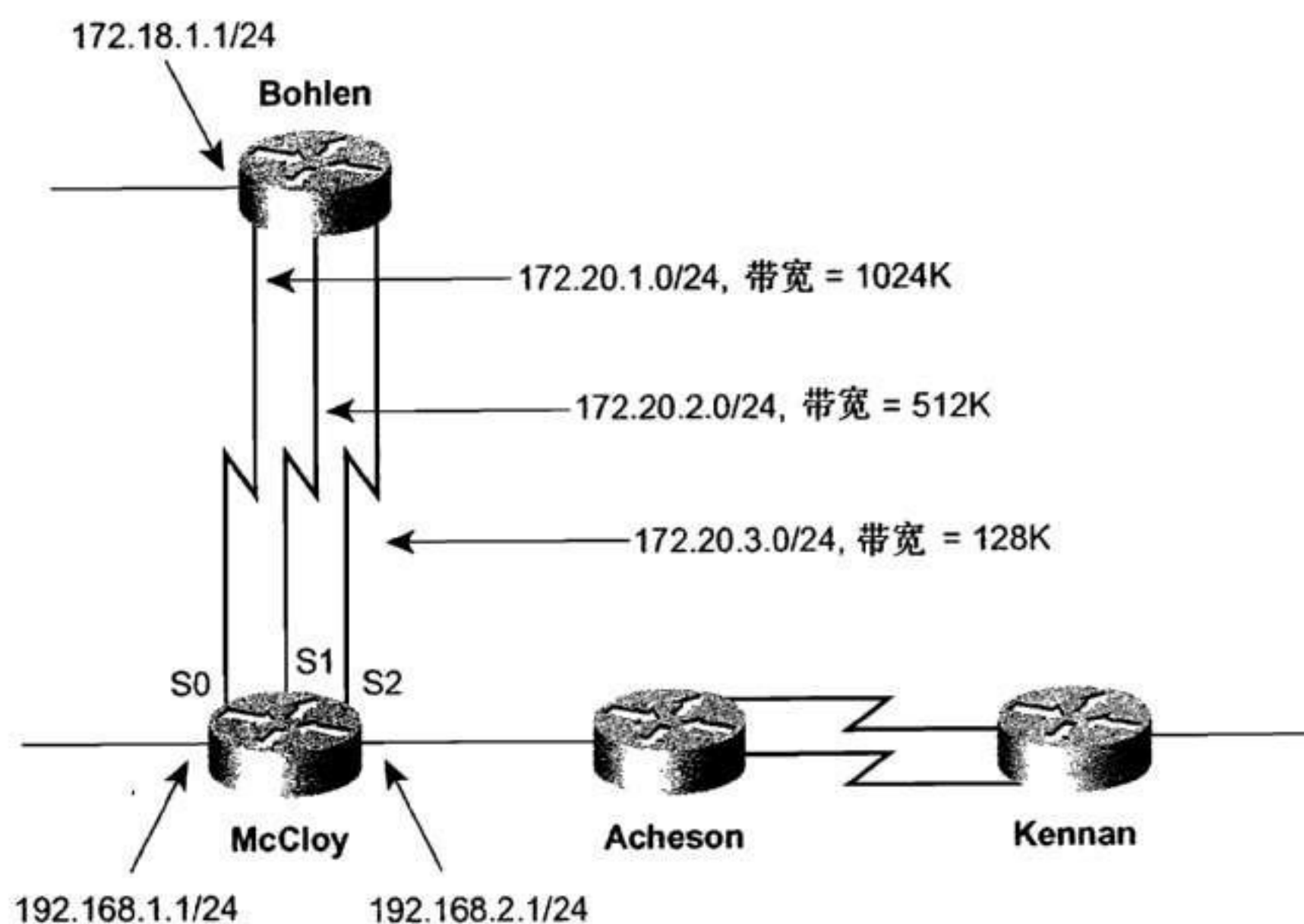


图 6-16 同时配置 **maximum-paths** 和 **variance** 命令, 用来在路由器 McCloy 和 Bohlen 之间 3 条链路中的其中两条上进行负载均衡。如果任何一条链路失效, 第 3 条链路将替代失效的链路

从路由器 McCloy 上来看这 3 条链路的度量值分别是:

- 通过 S0 接口: $9765 + (2000 + 100) = 11865$;
- 通过 S1 接口: $19531 + (2000 + 100) = 21631$;
- 通过 S2 接口: $78125 + (2000 + 100) = 80225$ 。

通过 S2 接口的度量是最小代价路径的度量的 6.76 倍, 因此, **variance** 的值应该是 7。在路由器 McCloy 上 IGRP 的配置为:

```
router igrp 10
 variance 7
 network 172.20.0.0
 network 192.168.1.0
 network 192.168.2.0
 maximum-paths 2
```

Variance 命令确保到达网络 172.18.0.0 的 3 条路由都是可用的; **maximum-paths** 命令用

来限制负载均衡“组”中最多只有两条最佳的路由。在图 6-17 中可以看到这种配置的结果。第一个路由选择表显示了路由器 McCloy 在通过 S0 和 S1 接口的两条链路上进行负载均衡的情形, 这里的两条链路的度量是 3 条链路中最低的。在第二个路由选择表中, 显示了链路 S1 失效后, 路由器又在通过 S0 和 S2 接口的两条链路上进行负载均衡的情形。在每一种情况下, 路由器都是执行和这两条路径的度量值成反比的负载均衡。

```
McCloy#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
       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

Gateway of last resort is not set

I    10.0.0.0 [100/8676] via 192.168.2.2, 00:00:02, Ethernet1
C    192.168.1.0 is directly connected, Ethernet0
C    192.168.2.0 is directly connected, Ethernet1
     172.20.0.0 255.255.255.0 is subnetted, 3 subnets
C       172.20.1.0 is directly connected, Serial0
C       172.20.2.0 is directly connected, Serial1
C       172.20.3.0 is directly connected, Serial2
I    172.18.0.0 [100/11865] via 172.20.1.2, 00:00:17, Serial0
     [100/21631] via 172.20.2.2, 00:00:17, Serial1

McCloy#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to down
%LINK-3-UPDOWN: Interface Serial1, changed state to down
McCloy#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
       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

Gateway of last resort is not set

I    10.0.0.0 [100/8676] via 192.168.2.2, 00:00:02, Ethernet1
C    192.168.1.0 is directly connected, Ethernet0
C    192.168.2.0 is directly connected, Ethernet1
     172.20.0.0 255.255.255.0 is subnetted, 2 subnets
C       172.20.1.0 is directly connected, Serial0
C       172.20.3.0 is directly connected, Serial2
I    172.18.0.0 [100/11865] via 172.20.1.2, 00:00:08, Serial0
     [100/80225] via 172.20.3.2, 00:00:08, Serial2

McCloy#
```

图 6-17 显示了同时使用 **variance** 和 **maximum-paths** 命令配置到达网络 172.18.0.0 的负载均衡时, 在 3 条链路的其中一条链路失效前后, 路由器 McCloy 的路由选择表

6.2.4 案例研究 4: 多个 IGRP 进程

增添两台新的路由器 Lovett 和 Harriman 到前面用来举例的互连网络中 (图 6-18)。这样在互连网络中创建了两个 IGRP 自主系统“域”, 这两个自主系统“域”之间没有通信。图 6-19 显示了这两个自主系统“域”和其相关的链路。

路由器 Bohlen、Lovett、McCloy 和 Kennan 的配置都是相当简单的, 路由器 Bohlen、Lovett 和 McCloy 运行 IGRP 10, 而路由器 Kennan 将运行 IGRP 15。在路由器 Acheson 上的配置如下:


```
network 172.16.0.0
```

每一个 IGRP 进程都只在指定网络的接口上。在路由器 Harriman 上, 两个接口都属于网络 10.0.0.0:

```
router igrp 10
  passive-interface TokenRing0
  network 10.0.0.0
!
router igrp 15
  passive-interface Serial0
  network 10.0.0.0
```

使用 **passive-interface** 命令是为了防止 IGRP 的更新广播到不属于它们的 IGRP 自主系统“域”中去。

6.3 IGRP 故障排除

像 RIP 协议一样, IGRP 协议的故障排除通常也是一项简单的事情。在大多数的实例中, 通常是在互连网络中追溯某条路由的路径, 检查路由路径经过的每一跳路由器的路由选择表, 直到发现故障的源头。故障的产生通常和地址或掩码配置错误以及不连续的编址有关。

当更改 IGRP 进程的计时器和其他一些参数变量的缺省值时, 就会增大引起故障的可能性。特别是在没有完全理解改变这些值后产生的影响时则更有可能带来问题。下面的第一个案例研究演示了这种情况——IGRP 协议做了它应该做的, 但是故障来自于操作者的错误。第二个案例研究演示了不连续的地址范围, 但这并不一定是配置错误造成的。

6.3.1 案例研究 5: 非等价负载均衡 (二)

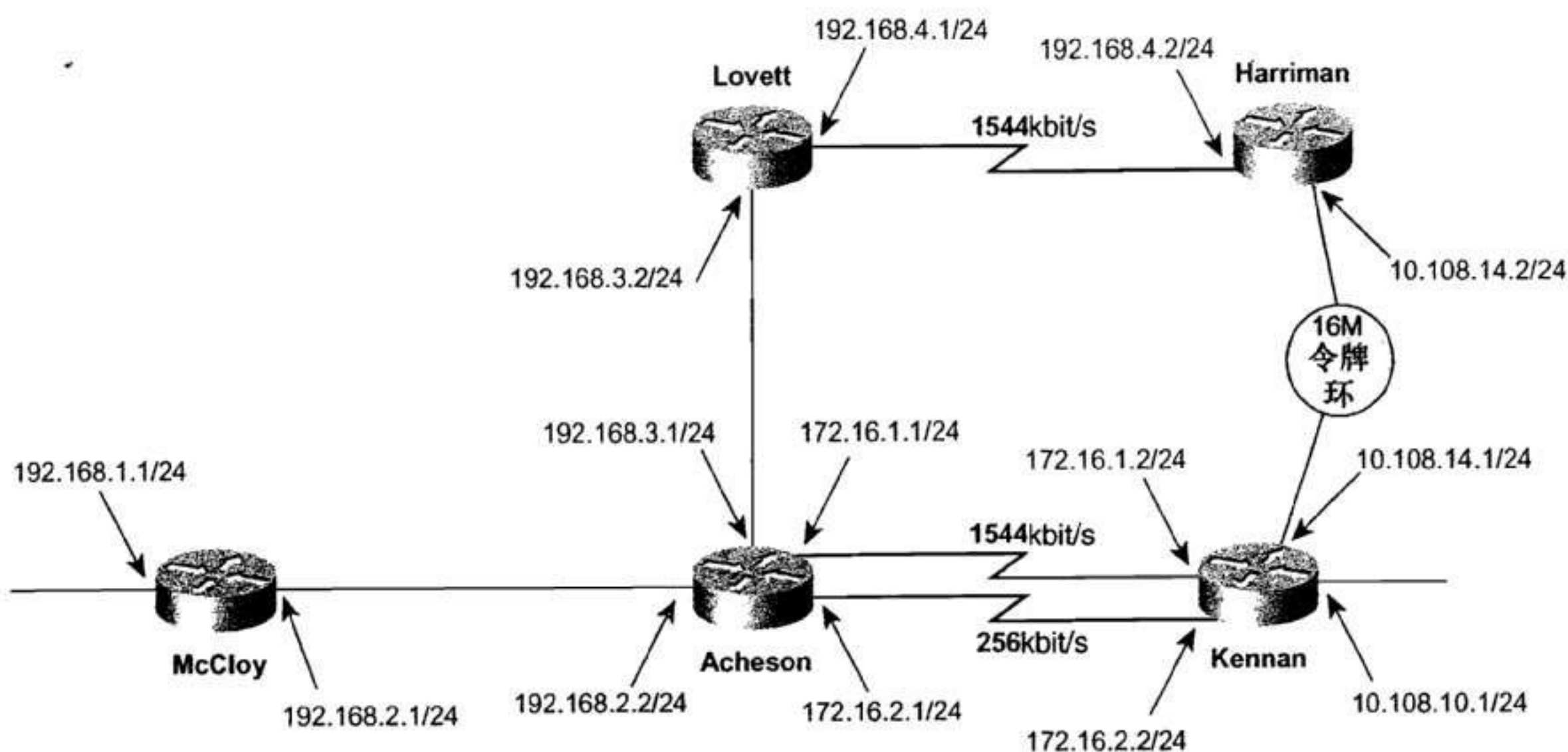
图 6-20 中的整个互连网络都是在单个 IGRP 进程中进行路由的, 串行链路的带宽配置已经标注在图上。这里使用缺省的时延。注意, 路由器 Lovett 和路由器 Harriman 之间的链路的地址是不同于前面的例子的。由于路由器 Acheson 不仅可以通过两条串行链路, 而且可以通过到达路由器 Lovett 的以太网链路到达目的网络 10.0.0.0, 因此, 网络管理员可以在这 3 条路由之间按比例分配业务量。

指向目的网络 10.0.0.0 的访问点有:

- 路由器 Kennan 的令牌环接口;
- 路由器 Kennan 的以太网接口;
- 路由器 Harriman 的令牌环接口。

路由器 Kennan 到达目的网络 10.0.0.0 的两个接口, 将通告令牌环接口上的最低时延。这 3 条路由最小的带宽是串行接口的带宽。从路由器 Acheson 上来看, 这 3 条路由的度量值分别是:

- 通过 S0: $6476 + (2000 + 63) = 8539$;
- 通过 S1: $39062 + (2000 + 63) = 41125$;



- 通过 E1: $6476 + (100 + 2000 + 63) = 8639$ 。

最大的度量值是最小的度量值的 4.8 倍，因此，variance 的值设置成 5。

Variance 配置好了，但是网络管理员发现负载均衡却没有像预期的那样工作（图 6-21）。路由选择表中仅仅显示了通过路由器 Kennan 的两条路由，通过路由器 Lovett 的路由的度量值在最大和最小的度量代价之间，却没有包含在路由选择表中。

```
Acheson#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
       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

Gateway of last resort is not set

I    10.0.0.0 [100/8539] via 172.16.1.2, 00:00:11, Serial0
      [100/41125] via 172.16.2.2, 00:00:11, Serial1
I    192.168.1.0 [100/1600] via 192.168.2.1, 00:00:11, Ethernet0
C    192.168.2.0 is directly connected, Ethernet0
C    192.168.3.0 is directly connected, Ethernet1
I    192.168.4.0 [100/8576] via 192.168.3.2, 00:00:11, Ethernet1
      172.16.0.0 255.255.255.0 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial0
C      172.16.2.0 is directly connected, Serial1

Acheson#
```

图 6-21 路由选择表中并没有包含通过路由器 Lovett 到达目的网络 10.0.0.0 的, 并且可以进行负载均衡的路由

回忆一下关于在一个负载均衡“组”里包含一条路由的3条规则，这些规则在配置非等价负载均衡的案例研究一节中已经作过讲述。在这个例子中，它违反了第二条规则，也就是下一跳路由器在度量上必须比本地路由器最好的路由更加靠近目的地。在路由器 Lovett 中，到达目的网络 10.0.0.0 的路由的度量值是 $6476 + (2000 + 63) = 8539$ ，这和路由器 Acheson 中的最佳路由路径的度量值相同，但并不是更好的路由。

在路由器 Lovett 上，对于串行链路，可以通过增加一点点带宽或减少一点点时延来使路

由的度量值小于 8539。在这个实例中, 时延减小了 10 μ s:

```
Lovett(config)#interface serial 0
Lovett(config-if)#delay 1999
```

路由选择表的显示结果在图 6-22 中。

```
Acheson#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
       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

Gateway of last resort is not set

I    10.0.0.0 [100/8539] via 172.16.1.2, 00:00:14, Serial0
      [100/41125] via 172.16.2.2, 00:00:14, Serial1
      [100/8638] via 192.168.3.2, 00:00:14, Ethernet1
I    192.168.1.0 [100/1600] via 192.168.2.1, 00:01:02, Ethernet0
C    192.168.2.0 is directly connected, Ethernet0
C    192.168.3.0 is directly connected, Ethernet1
I    192.168.4.0 [100/8575] via 192.168.3.2, 00:00:14, Ethernet1
      172.16.0.0 255.255.255.0 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial0
C      172.16.2.0 is directly connected, Serial1
Acheson#
```

图 6-22 将路由器 Lovett 的串行接口的时延减少 10 μ s 后, 路由器 Acheson 则接受了路由器 Lovett 到达目的网络 10.0.0.0 的路由

当更改度量时, 一定要十分注意查看结果。如果一条路径的代价没有合适地反映出链路的实际传送能力, 那么流量负载就可能出现误差——低带宽的链路可能出现超载, 或高带宽的链路出现利用率不足的情况。

6.3.2 案例研究 6: 被分段的网络 (Segmented Network)

图 6-22 中的互联网络在负载均衡的方式工作得还不错, 但在运行了一段时间后, 用户开始抱怨通过路由器 Acheson 到达网络 10.108.14.0 的数据流总是断断续续的。当网络管理员向这个目的网络的某个地址发出 100 个 ping 包时, 他们确认这条路径上的流量确实是时断时续的 (图 6-23)。

从图 6-23 可以看出, 这里 ping 包的结果并不是随机的方式——每 5 个 ping 包成功就交替有 6 个不成功的 ping 包。打开数据包的调试功能, 并发送更多的 ping 包来察看到底是发生了什么 (图 6-24)。路由器 Acheson 应该是按照 5/5/1 的比例模式进行负载均衡的 (5 个数据包通过 E1 接口发送, 5 个包通过 S0 接口发送, 1 个包通过 S1 接口发送)。通过 E1 接口发送的数据包都是成功的, 而所有通过串行链路发送的数据包都失败了。

进一步的探索发现路由器 Kennan 的令牌环接口处的电缆没有连好, 于是修复了这个故障。现在的问题是: 为什么通过串行接口的路由依然存在? 它们并没有被标记成不可达的路由, 这是因为路由器 Kennan 的以太网接口依然是正常工作的。路由器在向网络 172.16.0.0 通告路由时对网络 10.0.0.0 进行了路由汇总, 因此, 路由器没有办法把那个失效的子网通知给其他路由器。


```

Acheson#ping
Protocol [ip]:
Target IP address: 10.108.14.83
Repeat count [5]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echoes to 10.108.14.83, timeout is 2 seconds:
!!!!!!.....!!!!!!.....!!!!!!.....!!!!!!.....!!!!!!.....!!!!!!
.....!!!!!!.....!!!!!!.....!
Success rate is 46% (46/100), round-trip min/avg/max = 32/34/40 ms
Acheson#

```

图 6-23 到达子网 10.108.14.0 的流量断断续续的情形可以通过这些 ping 包的结果观察到，可以看出只有 45% 的 ping 包成功

```

Acheson#debug ip packet
IP packet debugging is on
Acheson#ping
Protocol [ip]:
Target IP address: 10.108.14.83
Repeat count [5]: 15
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 15, 100-byte ICMP Echoes to 10.108.14.83, timeout is 2 seconds:

IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.2.1 (local), d=10.108.14.83 (Serial1), len 100, sending.!!!!!!
IP: s=192.168.3.1 (local), d=10.108.14.83 (Ethernet1), len 100, sending
IP: s=10.108.14.83 (Ethernet1), d=192.168.3.1 (Ethernet1), len 114, rcvd 3
IP: s=192.168.3.1 (local), d=10.108.14.83 (Ethernet1), len 100, sending
IP: s=10.108.14.83 (Ethernet1), d=192.168.3.1 (Ethernet1), len 114, rcvd 3
IP: s=192.168.3.1 (local), d=10.108.14.83 (Ethernet1), len 100, sending
IP: s=10.108.14.83 (Ethernet1), d=192.168.3.1 (Ethernet1), len 114, rcvd 3
IP: s=192.168.3.1 (local), d=10.108.14.83 (Ethernet1), len 100, sending
IP: s=10.108.14.83 (Ethernet1), d=192.168.3.1 (Ethernet1), len 114, rcvd 3
IP: s=192.168.3.1 (local), d=10.108.14.83 (Ethernet1), len 100, sending
IP: s=10.108.14.83 (Ethernet1), d=192.168.3.1 (Ethernet1), len 114, rcvd 3
IP: s=192.168.3.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.1.1 (local), d=10.108.14.83 (Serial0), len 100, sending.
IP: s=172.16.2.1 (local), d=10.108.14.83 (Serial1), len 100, sending.
Success rate is 33% (5/15), round-trip min/avg/max = 36/36/40 ms
Acheson#

```

图 6-24 打开了数据包调试功能，发送 15 个 ping 包来寻找一些线索。根据经过路由器 Lovett 的路由转发的数据包都可以成功地到达目的地，而经过路由器 Kennan 的数据包全都转发失败了

6.4 展 望

最后一个案例研究和其他的一些例子展示了有类别路由选择协议的一些局限性和缺点。第 7 章将阐述第一个无类别路由选择协议 RIPv2。更为重要的是,第 7 章讨论了无类别协议怎样克服了像子网不连续和网络被分段这样的问题,同时,也说明了怎样利用可变长子网掩码 (variable-length subnet mask) 来支持更有效率的地址空间的设计。

6.5 总结表: 第 6 章命令总结

命 令	描 述
bandwidth <i>kilobits</i>	在接口上指定带宽参数,单位是 kbit/s。在一些路由选择协议中用来计算度量值,但它不影响数据链路实际的带宽
delay <i>tens-of-microseconds</i>	在接口上指定时延参数,单位是 10 μ s。在一些路由选择协议中用来计算度量值,但它不影响数据链路实际的时延
ip address <i>ip-address mask</i> [secondary]	在接口上指定 IP 地址和地址掩码
maximum-paths <i>maximum</i>	指定 IP 路由选择协议所能支持的并行路由路径数的最大值,设定范围可以是 1~6,缺省值是 4
metric holddown	打开或关闭 IGRP 抑制 (holddown)
metric maximum-hops <i>hops</i>	指定一条路由被标记成不可达之前 IGRP 协议所能通告的最大跳数,设定范围最大是 255,缺省是 100
metric weights <i>tos k1 k2 k3 k4 k5</i>	指定在 IGRP 和 EIGRP 协议中计算复合度量值时,对带宽、负载、时延和可靠性等参数所使用的权重
neighbor <i>ip-address</i>	定义一个 RIP、IGRP 或 EIGRP 协议发送路由更新信息的目的单播地址
network <i>network-number</i>	指定覆盖一个或多个运行 IGRP、EIGRP 或 RIP 协议的接口的网络地址
offset-list { <i>access-list-number</i> <i>name</i> }[<i>in</i> <i>out</i>] <i>offset</i> [<i>type number</i>]	指定一个跳数 (对于 RIP 协议) 或时延 (对于 IGRP 协议) 来增大入站或出站路由通告的度量值
passive-interface <i>type number</i>	抑制一个接口发出路由更新信息
router igrp <i>autonomous-system</i>	在路由器上启动一个指定自主系统号的 IGRP 进程
show interface [<i>type number</i>]	显示一个接口的配置和监控信息
show ip route [<i>address</i> [<i>mask</i>]][<i>protocol</i> [<i>process-ID</i>]]	显示当前的整个路由选择表和某条路由的信息
timers <i>basic update invalid holddown flush</i> [<i>sleeptime</i>]	调节 EGP、RIP 或 IGRP 处理的计时器
traffic-share { <i>balanced</i> <i>min</i> }	指定 IGRP 协议或 EIGRP 协议路由选择进程是否使用非等价负载均衡或只使用等价负载均衡
validate-update-source	打开或关闭 RIP 和 IGRP 路由选择进程对源地址有效性的验证
variance <i>multiplier</i>	指定一个倍数来表示一条路由与最小代价路径的度量值所差别的程度,确定是否可以依然包含在非等价负载均衡“组”中

6.6 推荐读物

“An Introduction to IGRP” 由 Rutgers University 大学的 Hedrick, C. L 于 1991 年 8 月编写,这篇文章可以从网址 <http://cco.cisco.com/warp/public/103/5.html> 上下载。虽然有点旧了,

但它依然是关于 IGRP 协议技术指导最好的公开文章。

6.7 复习题

1. 哪个 UDP 端口号被用来访问 IGRP 协议的信息?
2. 基于跳数的度量, IGRP 协议最大支持多大口径的网络?
3. IGRP 协议缺省的更新周期是多少?
4. 为什么 IGRP 协议要指定一个自主系统号?
5. 参考图 6-11, 路由器 McCloy 将会把网络 192.168.1.0 当成是内部路由、系统路由还是外部路由来通告? 类似地, 路由器 Acheson 会把网络 172.16.0.0 当成什么类型的路由来通告呢?
6. IGRP 协议缺省的抑制时间是多少?
7. IGRP 协议可以通过哪些变量参数计算它的复合度量值?
8. 在一个单一的 IGRP 更新报文中能够携带多少个路由条目?

6.8 配置练习

1. 写出图 6-25 中 6 台路由器的配置, 使它们可以通过 IGRP 协议来进行路由选择。自主系统号使用 50。

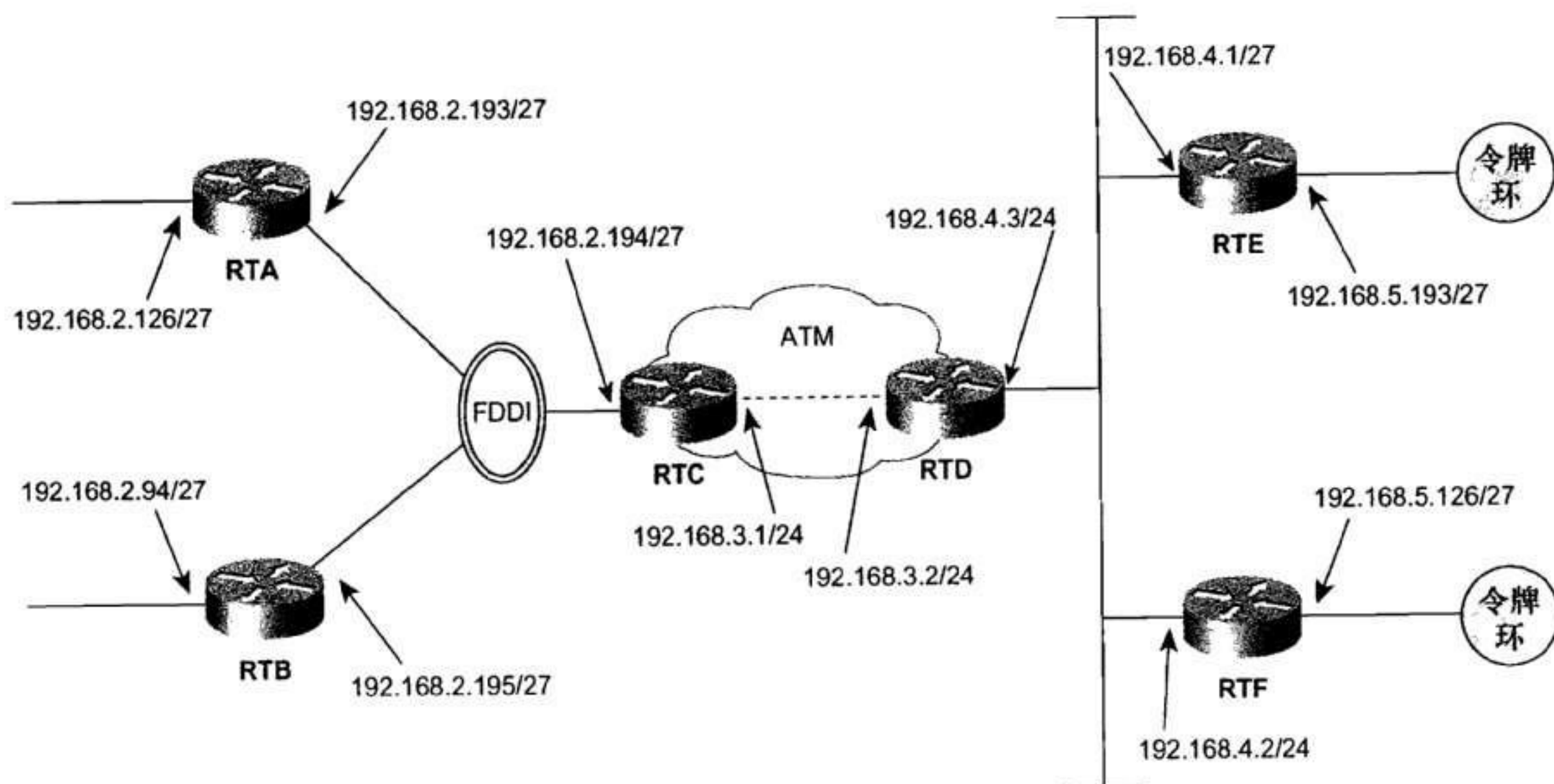


图 6-25 配置练习 1~3 的互连网络图

2. 图 6-26~6-29 显示了从子网 192.168.2.96/27 到子网 192.168.5.96/27 的路由路径上的出站接口的信息。假定 IGRP 协议使用缺省的带宽和时延作为它的度量, 请计算这条路由的复

合度量值。

```

RTA#show interface fddi0
Fddi0 is up, line protocol is up
  Hardware is DAS FDDI, address is 00e0.1e8e.d1d9 (bia 00e0.1e8e.d1d9)
  Internet address is 192.168.2.193/27
  MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive not set
  ARP type: SNAP, ARP Timeout 04:00:00
  Phy-A state is off, neighbor is Unknown, status no signal
  Phy-B state is off, neighbor is Unknown, status no signal
  ECM is out, CFM is isolated, RMT is isolated
  Requested token rotation 5000 usec, negotiated 5017 usec
  Configured tvx is 3400 usec, using 5242.90 usec, ring not operational
  0 SMT frames processed, 0 dropped, 20 SMT buffers
  Upstream neighbor 0000.f800.0000, downstream neighbor 0000.f800.0000
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queuing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort.
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    0 output buffer failures, 0 output buffers swapped out
    2 transitions, 0 traces
RTA#

```

图 6-26 图 6-25 中路由器 RTA 的 FDDI 接口

```

RTC# show interfaces atm 3/1
ATM3/1 is up, line protocol is up
  Hardware is cxBus ATM
  Internet address is 192.168.3.1, subnet mask is 255.255.255.0
  MTU 4470 bytes, BW 155000 Kbit, DLY 70 usec, rely 255/255, load 1/255
  Encapsulation ATM, loopback not set, keepalive set (10 sec)
  Encapsulation(s): AAL5, PVC mode
  256 TX buffers, 256 RX buffers, 1024 Maximum VCs, 1 Current VCs
  Signalling vc = 1, vpi = 0, vci = 5
  ATM NSAP address: 14.84D3.01.6A3A23.8340.DEAC.F021.8357.2192.A78E.13
  Last input 0:00:05, output 0:00:05, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
    144 packets input, 3148 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    154 packets output, 4228 bytes, 0 underruns
    0 output errors, 0 collisions, 1 interface resets, 0 restarts

```

图 6-27 图 6-25 中路由器 RTC 的 ATM 接口


```

RTD#show interface ethernet1/2
Ethernet1/2 is up, line protocol is up
  Hardware is Lance, address is 0000.0c0a.2c51 (bia 0000.0c0a.2c51)
  Internet address is 192.168.4.3/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:06, 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
    85496 packets input, 8284044 bytes, 0 no buffer
    Received 85421 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 input packets with dribble condition detected
    38594 packets output, 3478807 bytes, 0 underruns
    0 output errors, 1 collisions, 5 interface resets
    0 babbles, 0 late collision, 15 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
RTD#

```

图 6-28 图 6-25 中路由器 RTD 的以太网接口

```

RTF#show interface tokenring0
TokenRing0 is up, line protocol is up
  Hardware is TMS380, address is 0000.3090.c7df (bia 0000.3090.c7df)
  Internet address is 192.168.5.126/27
  MTU 4464 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive set (10 sec)
  ARP type: SNAP, ARP Timeout 04:00:00
  Ring speed: 16Mbps
  Single ring node, Transparent Bridge capable
  Group Address: 0x00000000, Functional Address: 0x08000000
  Ethernet Transit OUI: 0x000000
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  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
    29245 packets input, 1934430 bytes, 0 no buffer
    Received 75700 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    31612 packets output, 2220089 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    0 output buffer failures, 0 output buffers swapped out
    5 transitions
RTF#

```

图 6-29 图 6-25 中路由器 RTF 的令牌环接口

3. 在图 6-25 中 6 台路由器的 IGRP 配置中, 增加了命令 **metric weights 0 1 1 0 1 1**。请重新计算从子网 192.168.2.96/27 到子网 192.168.5.96/27 的路由的复合度量值。

4. 图 6-30 中的两台路由器运行 IGRP 协议, 它们接口上配置的带宽和时延已经显示在各自的链路上。路由器中必须增加什么命令才能使所有的链路都能够进行非等价负载均衡?

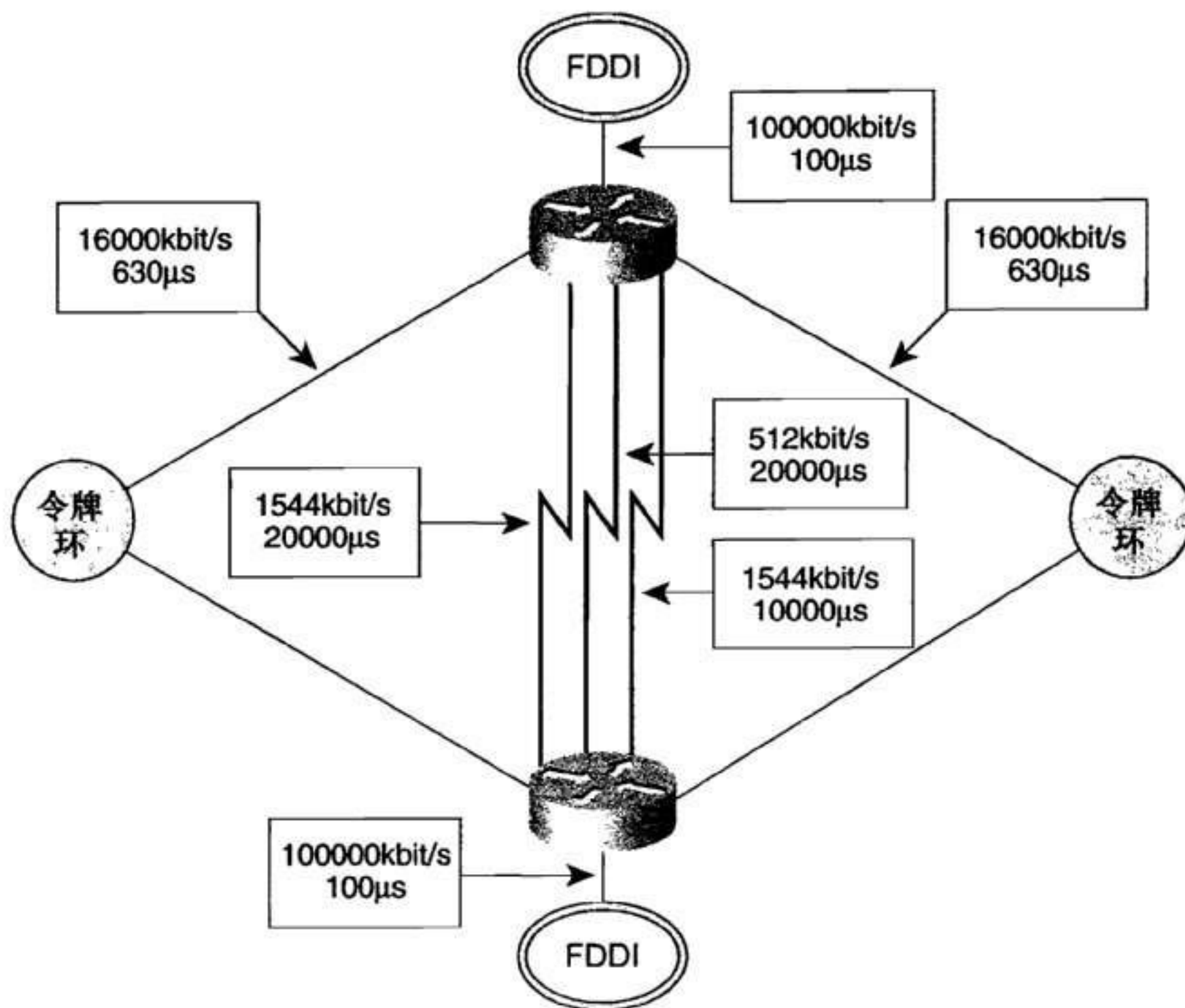


图 6-30 配置练习 4 的互联网络图

6.9 故障排除练习

1. 图 6-31 显示了图 6-32 中路由器 RTA 的路由选择表。在这个互联网络中, 虽然没有出现路由可达性的问题, 但是路由器 RTA 的路由选择表中却包含了不应该出现的路由条目——网络 192.168.3.0 可以通过串行链路可达, 而这条低带宽的串行链路却不是原来所期望的。图 6-33~图 6-36 显示了在这 4 台路由器上使用 debug 功能所观察到的 IGRP 更新。尽管通过这些调试信息并不能确定引起故障的原因, 但是能够发现最初导致发生故障的路由器。根据给出的这些信息, 作出一个假设, 来解释最可能引起故障发生的原因。

```
RTA#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
```

待续


```

10.0.0.0/24 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Serial1
I 192.168.1.0/24 [100/8676] via 172.17.16.56, 00:00:38, Ethernet0
I 192.168.2.0/24 [100/1200] via 172.17.16.56, 00:00:38, Ethernet0
I 192.168.3.0/24 [100/12476] via 172.16.17.9, 00:00:19, Serial0
172.16.0.0/30 is subnetted, 1 subnets
C    172.16.17.8 is directly connected, Serial0
172.17.0.0/28 is subnetted, 1 subnets
C    172.17.16.48 is directly connected, Ethernet0
RTA#

```

图 6-31 图 6-32 中路由器 RTA 的路由选择表

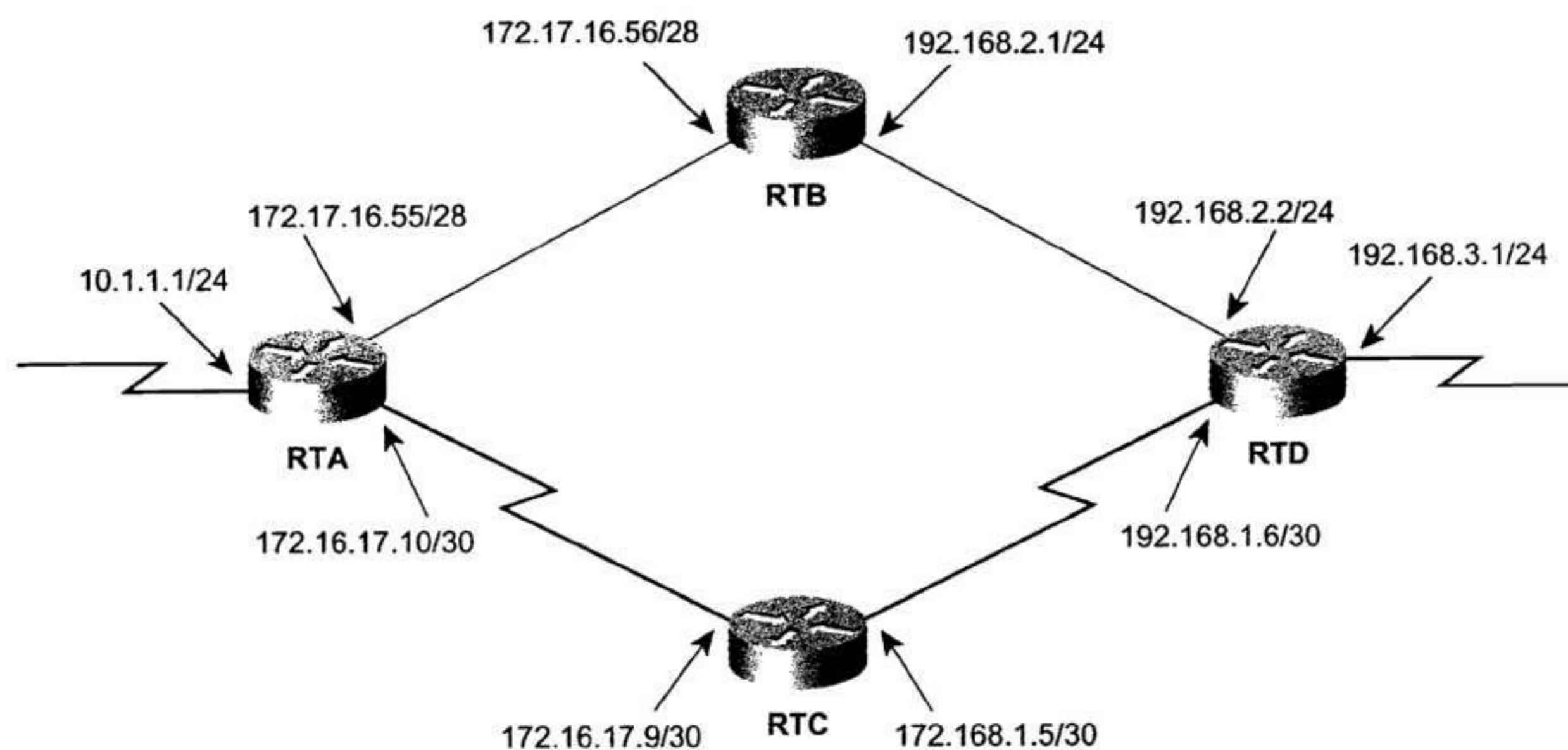


图 6-32 故障排除练习 1 的互连网络图

```

RTA#debug ip igrp transactions
IGRP protocol debugging is on
RTA#
IGRP: received update from 172.17.16.56 on Ethernet0
network 192.168.1.0, metric 8676 (neighbor 8576)
network 192.168.2.0, metric 1200 (neighbor 1100)
IGRP: sending update to 255.255.255.255 via Ethernet0 (172.17.16.55)
network 10.0.0.0, metric=8476
network 192.168.3.0, metric=12476
network 172.16.0.0, metric=8476
IGRP: sending update to 255.255.255.255 via Serial0 (172.16.17.10)
network 10.0.0.0, metric=8476
network 192.168.1.0, metric=8676
network 192.168.2.0, metric=1200
network 172.17.0.0, metric=1100
IGRP: sending update to 255.255.255.255 via Serial1 (10.1.1.1)
network 192.168.1.0, metric=8676
network 192.168.2.0, metric=1200
network 192.168.3.0, metric=12476
network 172.16.0.0, metric=8476

```

待续


```

network 172.17.0.0, metric=1100
IGRP: received update from 172.16.17.9 on Serial0
network 192.168.1.0, metric 10476 (neighbor 8476)
network 192.168.2.0, metric 10576 (neighbor 8576)
network 192.168.3.0, metric 12476 (neighbor 10476)

```

图 6-33 图 6-32 中路由器 RTA 接收和发送的 IGRP 更新

```

RTB#debug ip igrp transactions
IGRP protocol debugging is on
RTB#
IGRP: received update from 172.17.16.55 on Ethernet0
network 10.0.0.0, metric 8576 (neighbor 8476)
network 192.168.3.0, metric 12576 (neighbor 12476)
network 172.16.0.0, metric 8576 (neighbor 8476)
IGRP: sending update to 255.255.255.255 via Ethernet0 (172.17.16.56)
network 192.168.1.0, metric=8576
network 192.168.2.0, metric=1100
IGRP: sending update to 255.255.255.255 via Ethernet1 (192.168.2.1)
network 10.0.0.0, metric=8576
network 172.16.0.0, metric=8576
network 172.17.0.0, metric=1100
IGRP: received update from 192.168.2.2 on Ethernet1
network 192.168.1.0, metric 8576 (neighbor 8476)
network 192.168.3.0, metric 8576 (neighbor 8476)

```

图 6-34 图 6-32 中路由器 RTB 接收和发送的 IGRP 更新

```

RTC#debug ip igrp transactions
IGRP protocol debugging is on
RTC#
IGRP: sending update to 255.255.255.255 via Serial0 (172.16.17.9)
network 192.168.1.0, metric=8476
network 192.168.2.0, metric=8576
network 192.168.3.0, metric=10476
IGRP: sending update to 255.255.255.255 via Serial1 (192.168.1.5)
network 10.0.0.0, metric=10476
network 172.16.0.0, metric=8476
network 172.17.0.0, metric=8576
IGRP: received update from 172.16.17.10 on Serial0
network 10.0.0.0, metric 10476 (neighbor 8476)
network 192.168.1.0, metric 10676 (neighbor 8676)
network 192.168.2.0, metric 8676 (neighbor 1200)
network 172.17.0.0, metric 8576 (neighbor 1100)
IGRP: received update from 192.168.1.6 on Serial1
network 10.0.0.0, metric 10676 (neighbor 8676)
network 192.168.2.0, metric 8576 (neighbor 1100)
network 192.168.3.0, metric 10476 (neighbor 8476)
network 172.16.0.0, metric 10676 (neighbor 8676)
network 172.17.0.0, metric 8676 (neighbor 1200)

```

图 6-35 图 6-32 中路由器 RTC 接收和发送的 IGRP 更新


```

RTD#debug ip igrp transactions
IGRP protocol debugging is on
RTD#
IGRP: received update from 192.168.2.1 on Ethernet0
  network 10.0.0.0, metric 8676 (neighbor 8576)
  network 172.16.0.0, metric 8676 (neighbor 8576)
  network 172.17.0.0, metric 1200 (neighbor 1100)
IGRP: sending update to 255.255.255.255 via Ethernet0 (192.168.2.2)
  network 192.168.1.0, metric=8476
  network 192.168.3.0, metric=8476
IGRP: sending update to 255.255.255.255 via Serial0 (192.168.1.6)
  network 10.0.0.0, metric=8676
  network 192.168.2.0, metric=1100
  network 192.168.3.0, metric=8476
  network 172.16.0.0, metric=8676
  network 172.17.0.0, metric=1200
IGRP: sending update to 255.255.255.255 via Serial1 (192.168.3.1)
  network 10.0.0.0, metric=8676
  network 192.168.1.0, metric=8476
  network 192.168.2.0, metric=1100
  network 172.16.0.0, metric=8676
  network 172.17.0.0, metric=1200
IGRP: received update from 192.168.1.5 on Serial0
  network 10.0.0.0, metric 12476 (neighbor 10476)
  network 172.16.0.0, metric 10476 (neighbor 8476)
  network 172.17.0.0, metric 10576 (neighbor 8576)

```

图 6-36 图 6-32 中路由器 RTD 接收和发送的 IGRP 更新

2. 图 6-37 中在子网 172.16.1.8/29 和 172.16.2.16/29 网段的用户反映他们不能连接到服务器所在的子网 172.17.1.8/29 上。图 6-38~图 6-41 显示了协议分析仪在这两条以太链路上捕获到的 IGRP 更新信息。每一屏都显示了 IP 报文的头部部分。请确认每个更新信息是哪一个路由器发出的，并找出问题的所在。

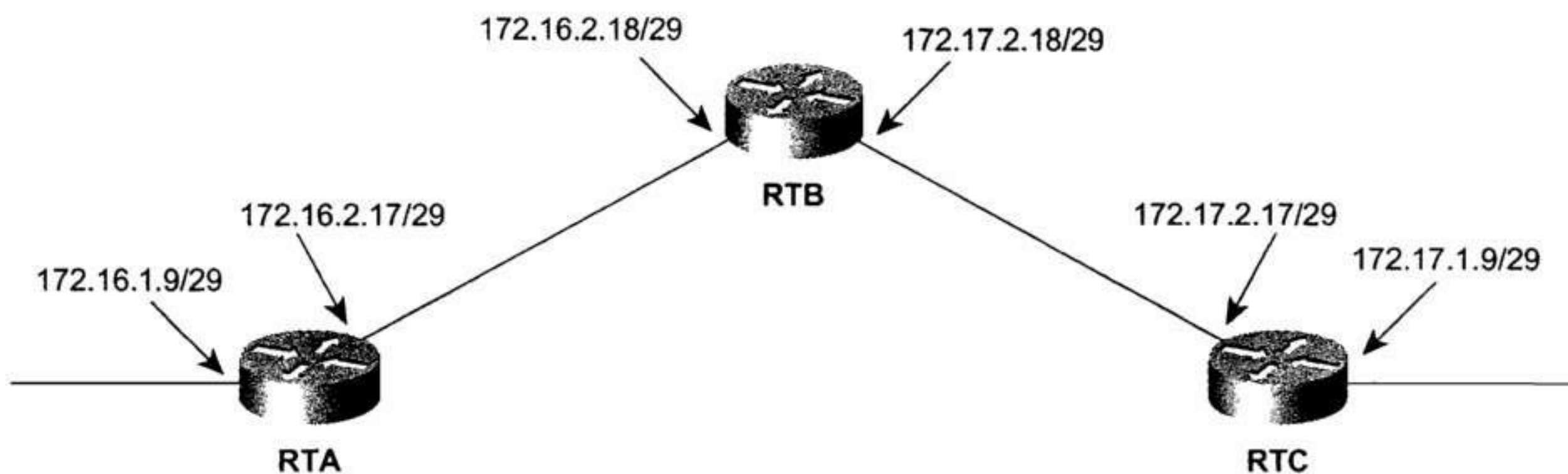


图 6-37 故障排除练习 2 的互联网络图

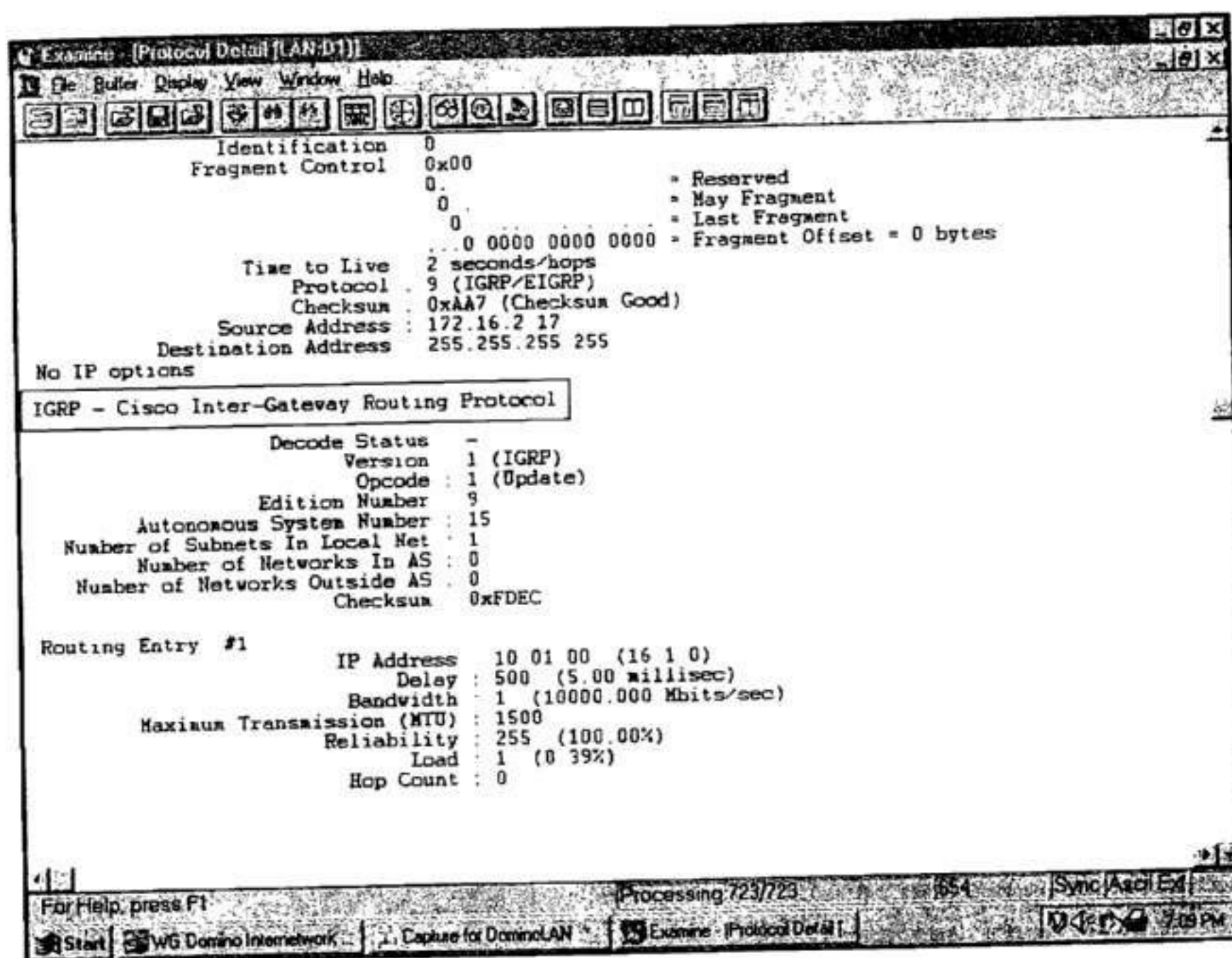


图 6-38 协议分析仪在图 6-37 所示的互联网络上捕获的 IGRP 更新信息

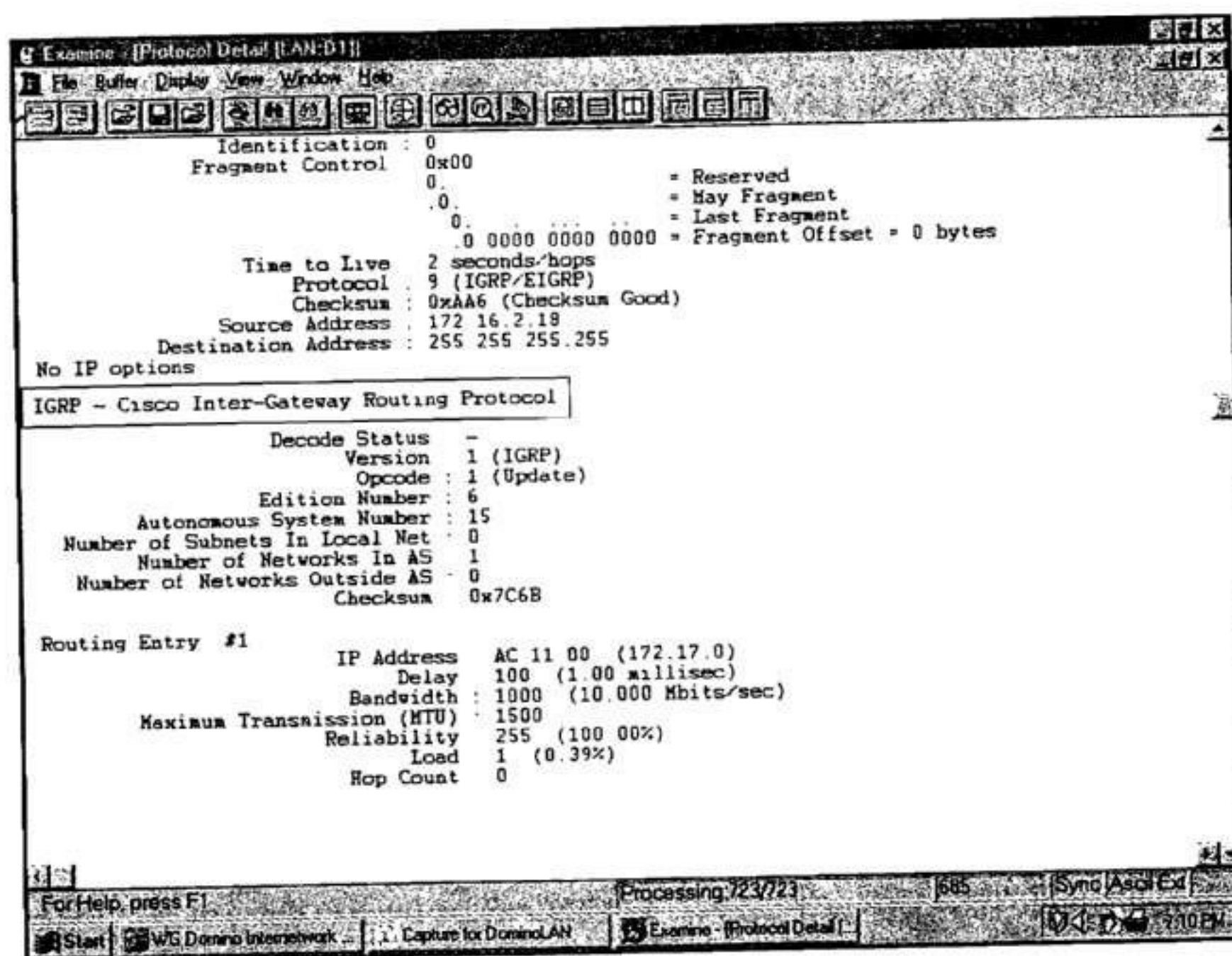


图 6-39 协议分析仪在图 6-37 所示的互联网络上捕获的 IGRP 更新信息

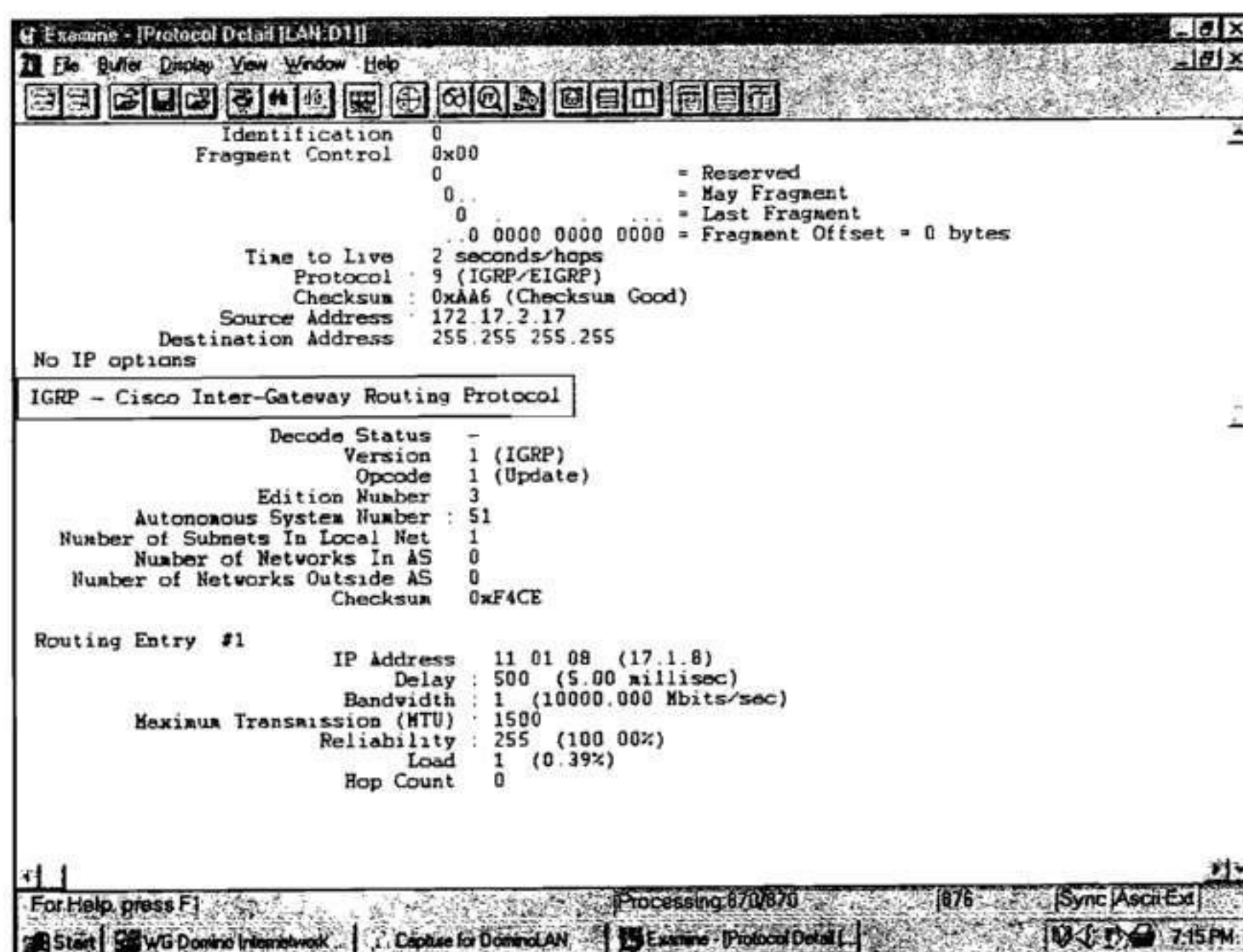


图 6-40 协议分析仪在图 6-37 所示的互联网络上捕获的 IGRP 更新信息

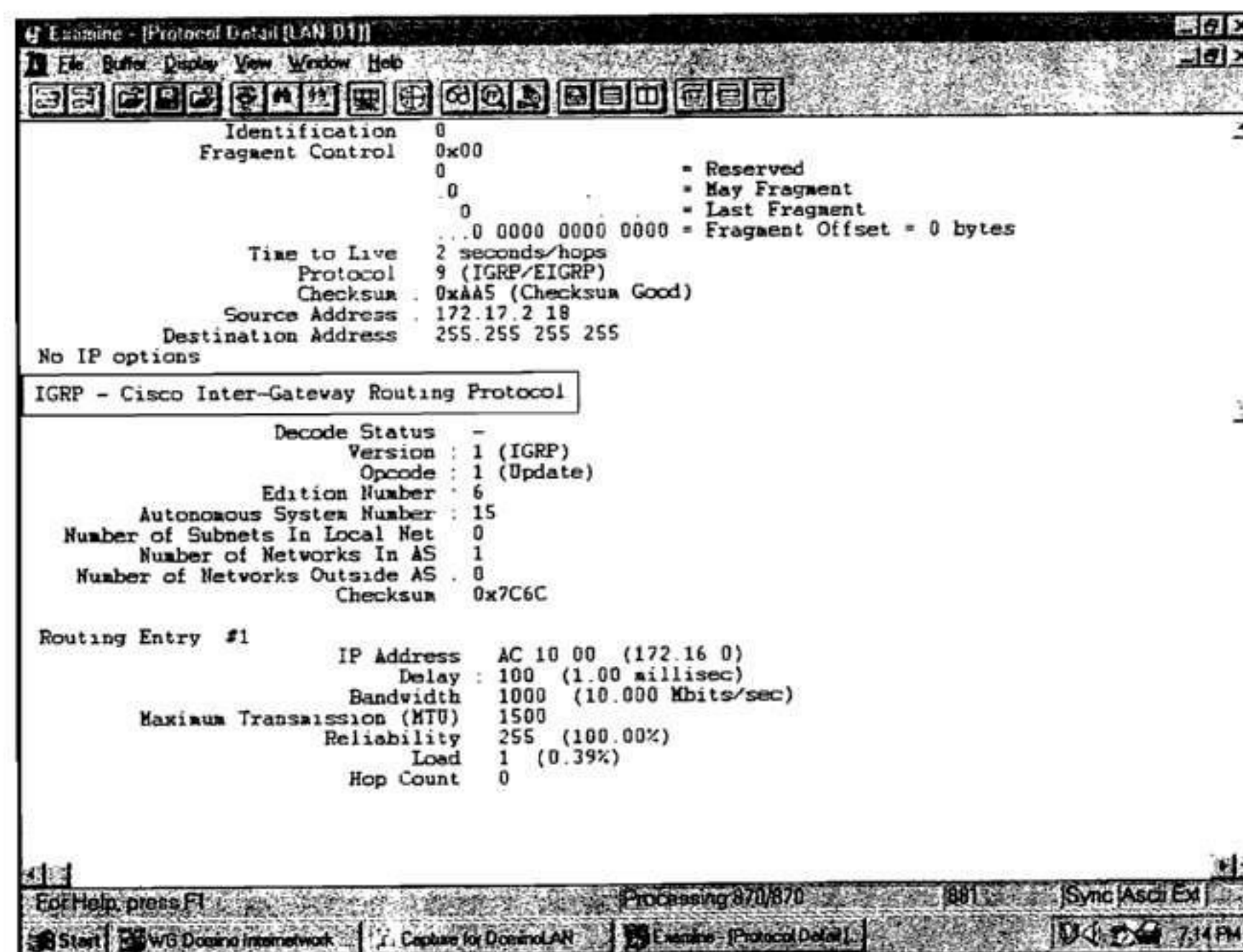


图 6-41 协议分析仪在图 6-37 所示的互联网络上捕获的 IGRP 更新信息

第 7 章

路由选择信息协议 ——第 2 版 (RIPv2)

本章包括以下主题：

- RIPv2 的操作
 - RIPv2 的消息格式
 - 与 RIPv1 的兼容性
 - 无类别路由查找
 - 无类别路由选择协议
 - 可变长子网掩码
 - 认证
- RIPv2 的配置
 - 案例研究：一个基本的 RIPv2 配置
 - 案例研究：使用 VLSM
 - 案例研究：不连续的子网和无类别路由选择
 - 案例研究：认证
- RIPv2 的故障排除
 - 配置错误的 VLSM

RIPv2 协议在 RFC1723¹中进行了定义，并在 Cisco IOS 11.1 版及后续的版本中得到支持。确切地说，RIPv2 协议不是一个新的协议，它只是在 RIPv1 协议的基础上增加了一些扩展特性，以适用于现代网络的路由选择环境，这些扩展特性有：

- 每个路由条目都携带自己的子网掩码；
- 路由选择更新具有认证功能；
- 每个路由条目都携带下一跳地址；

¹ 对这个 RFC 的补充还有 RFC 1721——“RIP Version 2 Protocol Analysis”和 RFC 1722——“RIP Version 2 rotocol Applicability Statement”。

- 外部路由标志;
- 组播路由更新。

在这些扩展特性中,最重要的一项就是路由更新条目增加了子网掩码的字段,因而 RIPv2 协议可以使用可变长的子网掩码,从而使 RIPv2 协议变成了一个无类别的路由选择协议。

RIPv2 协议是本书中讲述的第一个无类别路由选择协议。因此,本章将同时介绍无类别路由选择和 RIPv2 协议。

7.1 RIPv2 的操作

所有在 RIPv1 中运用的操作过程、计时器和稳定特性都同样可以在版本 2 中使用,其中只有一个例外,就是路由更新的广播。RIPv2 协议使用组播的方式向其他宣告 RIPv2 的路由器发出更新报文,它所使用的组播地址是保留的 D 类地址 224.0.0.9。使用组播方式的好处在于,本地网络上相连的和 RIP 路由选择无关的设备不再需要花费时间对路由器广播的更新报文进行解析。组播更新将在“与 RIPv1 的兼容性”一节进一步阐述。

先来看一下 RIP 协议的消息格式中提供了哪些版本 2 的扩展特性,这一节将主要关注 RIPv2 的操作和这些新增的扩展特性所带来的好处。

7.1.1 RIPv2 的消息格式

RIPv2 的消息格式如图 7-1,它的基本结构和 RIPv1 相同。所有相对于原来协议的扩展特性都是由未使用的字段提供的。和版本 1 一样,RIPv2 的更新报文最大可以包含 25 个路由条目。同样,与版本 1 相同的是,RIPv2 的操作使用 UDP 的 520 端口号,并且数据报文的大小(包括一个 8 字节的 UDP 头部)最大为 512 个 8bit 字节。



图 7-1 RIPv2 利用了版本 1 的消息格式中的未使用字段, 因而这些扩展没有改变版本 1 的基本消息格式

- **命令 (Command)** ——只取值 1 和 2, 1 表示本消息是请求消息, 2 表示本消息是响应消息。
- **版本号 (Version)** ——对于 RIPv2, 该字段的值设置为 2。如果设置为 0 或者虽设置为 1 但消息是无效的 RIPv1 格式, 那么这个消息将被丢弃。RIPv2 处理无效的 RIPv1 消息。
- **地址族标识 (Address Family Identifier, AFI)** ——对于 IP 该项设置为 2。只有一个例外情况, 本消息是对路由器 (或主机) 整个路由选择表的请求时, 这个字段将被设置成 0。
- **路由标志 (Route Tag)** ——提供这个字段用来标记外部路由或重分配到 RIPv2 协议中的路由。默认的情况是使用这个 16 位的字段来携带从外部路由选择协议注入到 RIP 中的路由的自主系统号。虽然 RIP 协议自己并不使用这个字段, 但是在多个地点和某个 RIP 域相连的外部路由, 可能需要使用这个路由标记字段通过 RIP 域来交换路由信息。这个字段也可以用来把外部路由编成“组”, 以便于在 RIP 域中更容易地控制这些路由。关于路由标志的使用将在第 14 章“路由图”中进一步论述。
- **IP 地址 (IP Address)** ——路由条目的目的地址, 它可以是主网络地址、子网地址或主机路由。
- **子网掩码 (Subnet Mask)** ——是一个确认 IP 地址的网络和子网部分的 32 位的掩码。这个字段的意义将在“可变长子网掩码”一节中论述。
- **下一跳 (Next Hop)** ——如果存在的话, 它标识一个比通告路由器的地址更好的下一跳地址。换句话说, 它指出的下一跳地址, 其度量值比在同一个子网上的通告路由器更靠近目的地。如果这个字段设置为全 0 (0.0.0.0), 说明通告路由器的地址是最好的下一跳地址。在本节的最后将有一个例子说明这个字段的用处。
- **度量 (Metric)** ——是一个在 1~16 之间的跳数。

图 7-2 显示了 4 台连接在同一个以太网子网链路上的路由器。¹ 路由器 Jicarilla、Mescalero 和 Chiricahua 都属于自主系统 65501, 并相互通告 RIPv2。路由器 Chiricahua 是自主系统 65501 和自主系统 65502 之间的边界路由器, 在第二个自主系统中, 它通告 BGP 给路由器 Lipan。

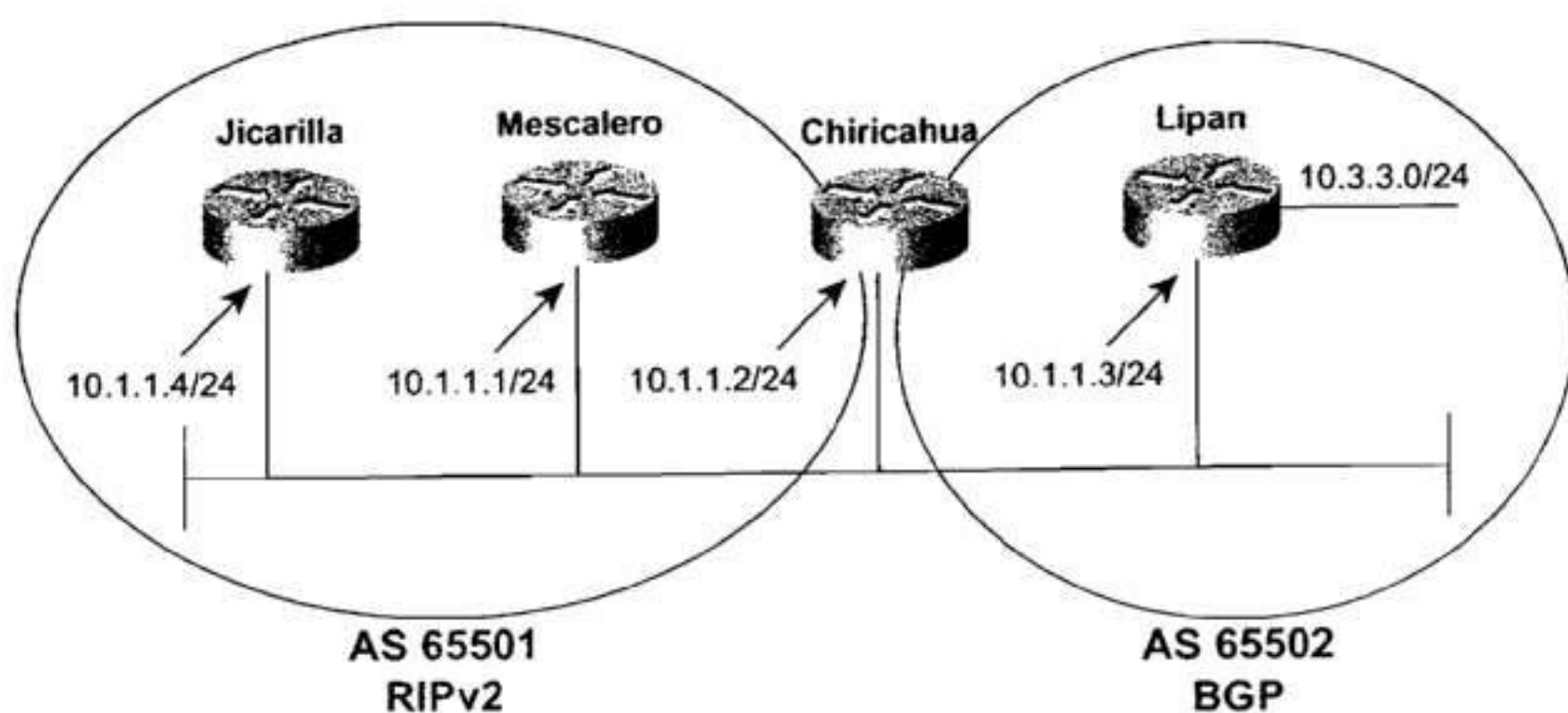


图 7-2 虽然它们共享一条公共的数据链路, 但路由器 Jicarilla 和 Mescalero 仅仅宣告 RIPv2, 而路由器 Lipan 仅仅宣告 BGP。路由器 Chiricahua 则负责把从后者学习到的路由通告给前面两台路由器

¹ 这个图示是根据 RFC1722 中 Gary Malkin 演示的一个例子改编的。

这里, 路由器 Chiricahua 正在通告它从 BGP 协议学习到的路由给宣告 RIP 的路由器 (图 7-3)。¹ 在路由器 Chiricahua 的 RIPv2 通告里, 它将使用路由标记字段来指出子网 10.3.3.0 (掩码是 255.255.255.0) 是位于自主系统 65502 (0xFFDE) 之中的。路由器 Chiricahua 也将使用下一跳字段告诉路由器 Jicarilla 和 Mescalero, 到达子网 10.3.3.0 最好的下一跳地址是路由器 Lipan 的接口 10.1.1.3, 而不是它们自己的接口。注意, 由于路由器 Lipan 不运行 RIP 协议, 而路由器 Jicarilla 和 Mescalero 不运行 BGP 协议, 这样即使是在同一个子网上路由器 Lipan 是可达的, 路由器 Jicarilla 和 Mescalero 也没有办法直接知道路由器 Lipan 是最好的下一跳路由器。

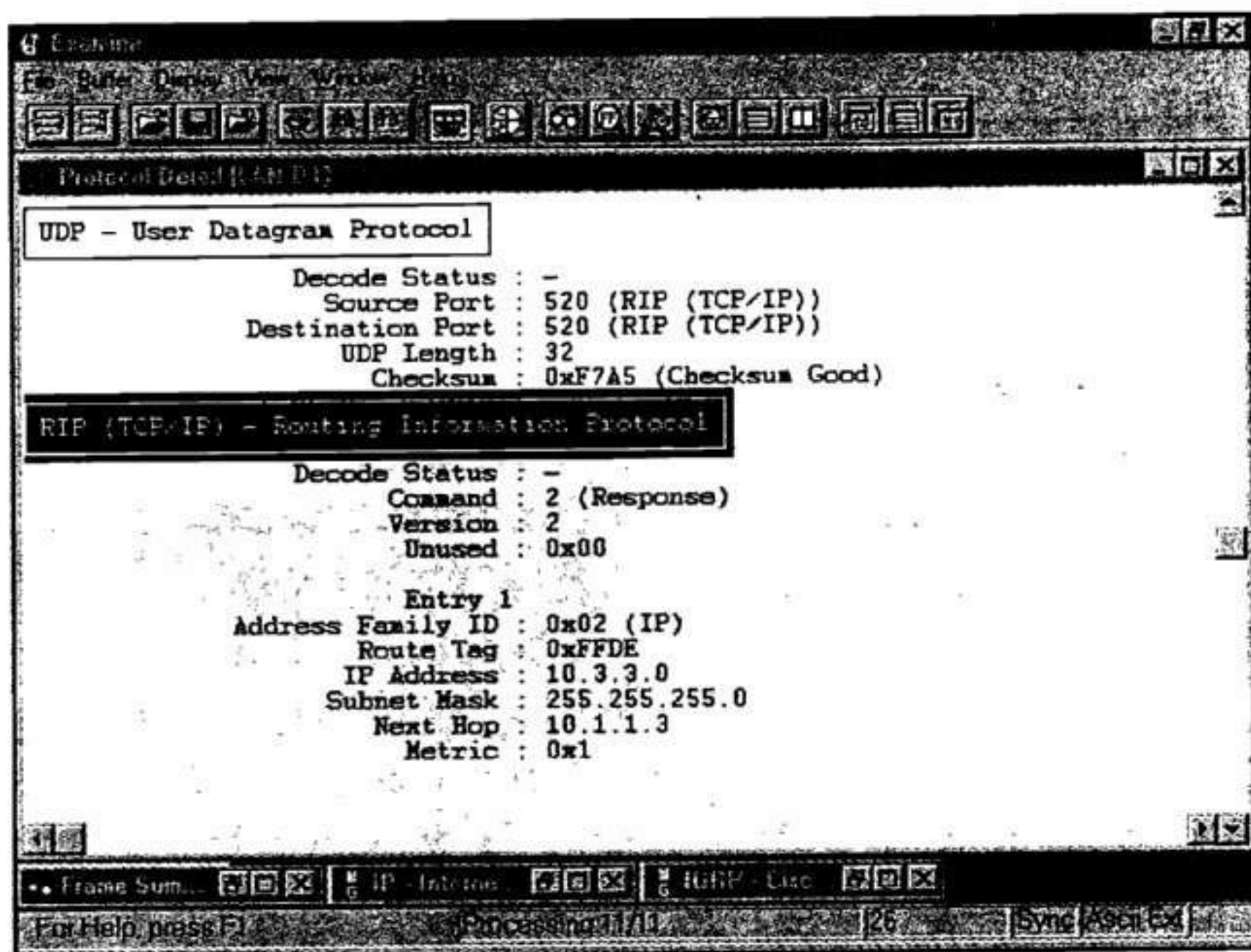


图 7-3 这个从路由器 Chiricahua 上捕获到的 RIPv2 更新显示了通告子网 10.3.3.0 时所使用的路由标记、子网掩码和下一跳等字段

7.1.2 与 RIPv1 的兼容性

RIPv1 使用了灵活的方式来进行路由更新。如果更新报文的版本字段指出是 RIP 的版本 1, 但所有未使用字段 (unused field) 的所有位都被设置为 1, 那么这个更新报文将被丢弃; 如果版本字段设置大于 1, 在版本 1 中定义为未使用的字段将被忽略, 并且处理这个消息。结果, 像 RIPv2 这样新版本的协议就可以向后兼容 RIPv1。

RFC1723 用 4 个设置定义了一个“兼容性开关”, 用来允许版本 1 和版本 2 之间的互操作:

- **RIP-1**——只有 RIPv1 的消息传送;
- **RIP-1 兼容性**——使 RIPv2 使用广播方式代替组播方式来通告消息, 以便 RIPv1 可以接收它们;

¹ 重分配是指把从一个路由协议学习到的路由通告给另一个路由选择协议的情况, 这将在第 11 章“路由重新分配”中详细论述。

- **RIP-2**——RIPv2 使用组播方式通告消息到目的地址 224.0.0.9;
- **None**——不发送更新。

RFC 建议这个开关基于每一个接口上配置。在“配置 RIPv2”一节中阐述了 Cisco 的命令为设置 1~3, 设置 4 的功能已经使用 **passive-interface** 命令完成了。

另外, RFC1723 定义了一个“接收控制开关”来控制更新的接收。对于这个开关, 4 个被建议的设置是:

- **RIP-1 Only;**
- **RIP-2 Only;**
- **Both;**
- **None。**

这个开关也应该是基于每一个接口上配置的。在本章“配置 RIPv2”一节中阐述了 Cisco 的命令为设置 1~3, 设置 4 的功能可以通过使用访问列表去过滤 UDP 源端口号 520, 或者在该接口上不包含 **network** 语句¹, 或者配置一个路由过滤列表完成, 后者将在第 13 章“路由过滤”中论述。

7.1.3 无类别路由查找

第 5 章阐述了有类别路由的查找方法——首先将目的地址与路由选择表中的主网络地址匹配, 然后匹配主网络的子网。如果经过这些步骤找不到匹配项, 这个数据包就会被丢弃。

即使对于像 RIPv1 和 IGRP 这样的有类别路由选择协议, 这种缺省的方式也能够通过全局命令 **ip classless** 更改。当路由器执行无类别路由查找时, 它不会注意目的地址的类别, 替代的方式是, 它在目的地址和所有已知的路由之间执行一位一位 (bit-by-bit) 的最佳匹配。当和缺省路由一起工作时, 这个性能变得非常有用, 这将在第 12 章“缺省路由和按需路由选择”讲述。当再加上无类别路由选择协议的其他一些特性时, 无类别路由查找的功能将显得更加强大。

7.1.4 无类别路由选择协议

无类别路由选择协议最根本的特点, 是它可以在路由通告中携带子网掩码。每条路由拥有子网掩码的一个好处就是, 全 0 和全 1 的子网现在可以利用了。第 2 章“TCP/IP 回顾”说明了有类别路由选择协议不能区分全 0 子网 (例如 172.16.0.0) 和主网络号 (172.16.0.0); 同样地, 它们也不能区分全 1 子网的广播 (172.16.255.255) 和全部子网的广播 (172.16.255.255)。

如果包含了子网掩码, 这个困难就不存在了。我们可以容易地看出网络 172.16.0.0/16 是一个主网络号, 而 172.16.0.0/24 则是一个全 0 的子网。172.16.255.255/16 和 172.16.255.255/24 也可以同样地区分开来。

在缺省的条件下, 即使正在运行无类别路由选择协议, Cisco IOS 软件也将拒绝尝试把一个全 0 的子网配置为有效的地址/掩码组合。为了忽略这个缺省的行为, 可以使用全局命令 **ip subnet-zero**。

¹ 这种方法只适用于下面的情形, 路由器上没有和同一个主网络相连的其他接口运行 RIP 协议。

每条路由拥有子网掩码的另一个很大的好处,就是可以使用可变长子网掩码和利用一条单一的聚合地址来汇总一组主网络地址。可变长子网掩码将在下面一节讲述,地址聚合(或超网)(address aggregation (or supernetting))将在第 8 章“增强型内部网关路由选择协议(EIGRP)”中介绍。

7.1.5 可变长子网掩码 (VLSM)

如果每一个目的地址都可以单独地携带相关联的子网掩码通告到整个网络中,那么就没有什么理由要求所有的掩码必须是等长的了。这个事实就是可变长子网掩码(VLSM)的基础。

图 7-4 显示了可变长子网掩码的一个简单应用。图中互连网络的每一条数据链路都必须有一个单独的可确认的子网掩码,同时,每一个子网地址段必须包含足够的主机地址数提供给这条数据链路上相连的设备使用。

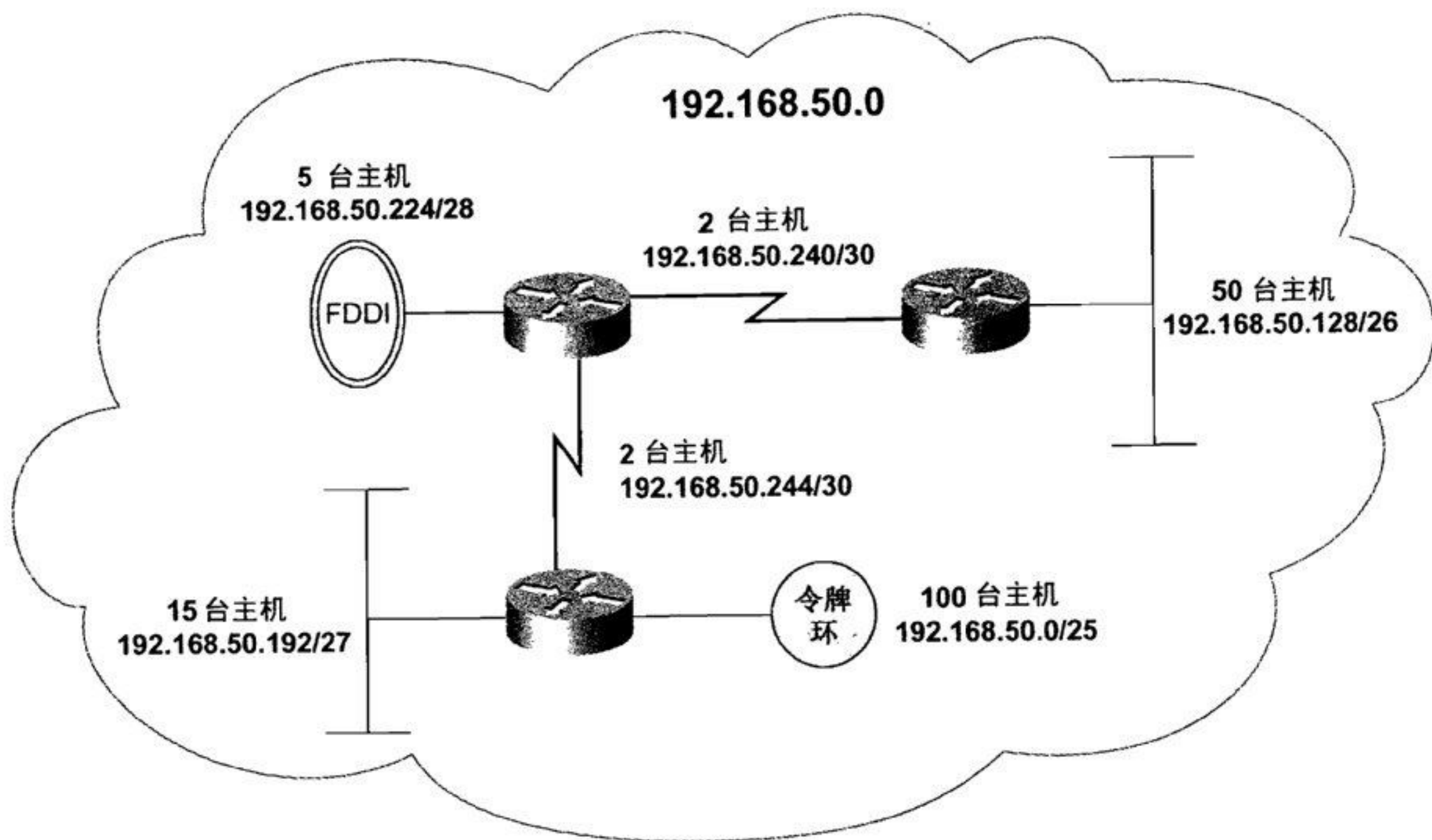


图 7-4 使用 VLSM, C 类地址可以被分成子网, 以便提供给这个互连网络和它的每一条数据链路上的主机使用

给定一个 C 类的网络地址分配给这个网络, 没有 VLSM 划分子网是不能完成的。这里令牌环的那个网需要 100 个主机地址, 也就是需要一个 25 位的掩码(有 1 位被子网化), 任何更长一点的掩码都将无法保留有足够的主机位。但是如果所有的子网掩码都必须都是等长的话, 那么从这个 C 类的地址中只能再分出一个子网。¹这样将没有办法提供所需要的足够的子网数。

使用 VLSM, 在图 7-4 的互连网络中多样的主机地址需求只需要 1 个 C 类的网络地址就可以满足了。表 7-1 显示了这些子网和每个子网内可用的地址范围。

¹ 这个说法假定全 0 和全 1 的子网可以被路由的, 因为对于只有 1 位被子网化时, 全 0 和全 1 子网是唯一可用的子网。

表 7-1

图 7-4 的子网

子网/掩码	地址范围	广播地址
192.168.50.0/25	192.168.50.1-192.168.50.126	192.168.50.127
192.168.50.128/26	192.168.50.129-192.168.50.190	192.168.50.191
192.168.50.192/27	192.168.50.193-192.168.50.222	192.168.50.223
192.168.50.224/28	192.168.50.225-192.168.50.238	192.168.50.239
192.168.50.240/30	192.168.50.241-192.168.50.242	192.168.50.243
192.168.50.244/30	192.168.50.245-192.168.50.246	192.168.50.247

很多人,包括许多使用 VLSM 的人利用这项技术解决的问题比上述例子复杂得多。使用 VLSM 技术的关键之处就是: 当一个网络地址依据标准的方式被划分子网以后, 那些子网本身也能够进一步被子网化。事实上, 有时候我们会偶尔听到 VLSM 关于“子网的子网化”的说法。

仔细察看表 7-1 的地址 (通常是二进制的), 就可以发现 VLSM 是怎样工作的。¹首先, 一个 25 位的掩码用来把一个网络地址划分成两个子网: 192.168.50.0/25 和 192.168.50.128/25。第一个子网可以提供 126 个主机地址满足图 7-4 中令牌环网段的需要。

根据第 2 章所讲述的, 我们了解到划分子网可以包括扩展的缺省网络掩码, 因此一些主机位可以被视为网络位。这种做法同样可以应用于剩下的子网 192.168.50.128/25。有一个以太网段需要 50 个主机地址, 因而余下的子网的掩码应该扩展到 26 位, 这一步提供了两个子网的子网 (sub-subnets) ——192.168.50.128/26 和 192.168.50.192/26, 每一个小子网都可以提供 62 个可用的主机地址。第一个小子网可以给前面那个大的以太网段使用, 剩下的第二个小子网可以进一步子网化提供给其他数据链路使用。

这个过程再重复两次, 就可以提供必需的满足大小的子网给小的以太网段和 FDDI 环网段使用。剩余的子网 192.168.50.240/28 可以用作两个串行链路所需要的子网。对于任何点到点的链路, 最普遍的情况只需要两个主机地址——每端一个地址。使用 30 位的掩码可以创建这两个串行链路的子网, 每个子网正好包括两个可用的主机地址。

点到点的链路需要子网地址, 但每个子网只需要两个主机地址, 这就是使用 VLSM 的一个理由。例如, 图 7-5 显示了一个典型的广域网络的拓扑, 它将每一个远程路由器都通过帧中继 PVC 虚电路与中心路由器 (hub router) 相连。现代网络设计的经验通常建议在点到点的子接口²上配置每个 PVC 电路。如果没有 VLSM 技术, 将必须需要相同大小的子网, 而这个子网的大小却是主机设备数量最多的子网所需要的地址数。

假设图 7-5 中的网络使用了一个 B 类的地址, 并且每一台路由器都和几个局域网相连, 而这些局域网连接的设备数量最大为 175 个。包含每个 PVC 电路的子网都应该需要一个 24 位的掩码, 这样对互连网络中的每一个 PVC 链路来说, 就有 252 个地址浪费了。使用 VLSM 技术, 选用单个子网并使用 30 位的掩码对子网进行子网化就可以满足需要了, 可以划分足够的子网最多可以满足 64 个点到点的链路 (图 7-6)。

VLSM 地址设计的例子在本章和后续的章节中都有讲述。第 8 章介绍了使用 VLSM 技术的另一个主要用途, 即层次化的编址和地址聚合。

¹ 强烈建议读者把这个例子的所有地址转换成二进制的。

² 子接口超出了本书的讲解范围, 还不熟悉这些有用的工具的读者可以参考 Cisco 配置手册。

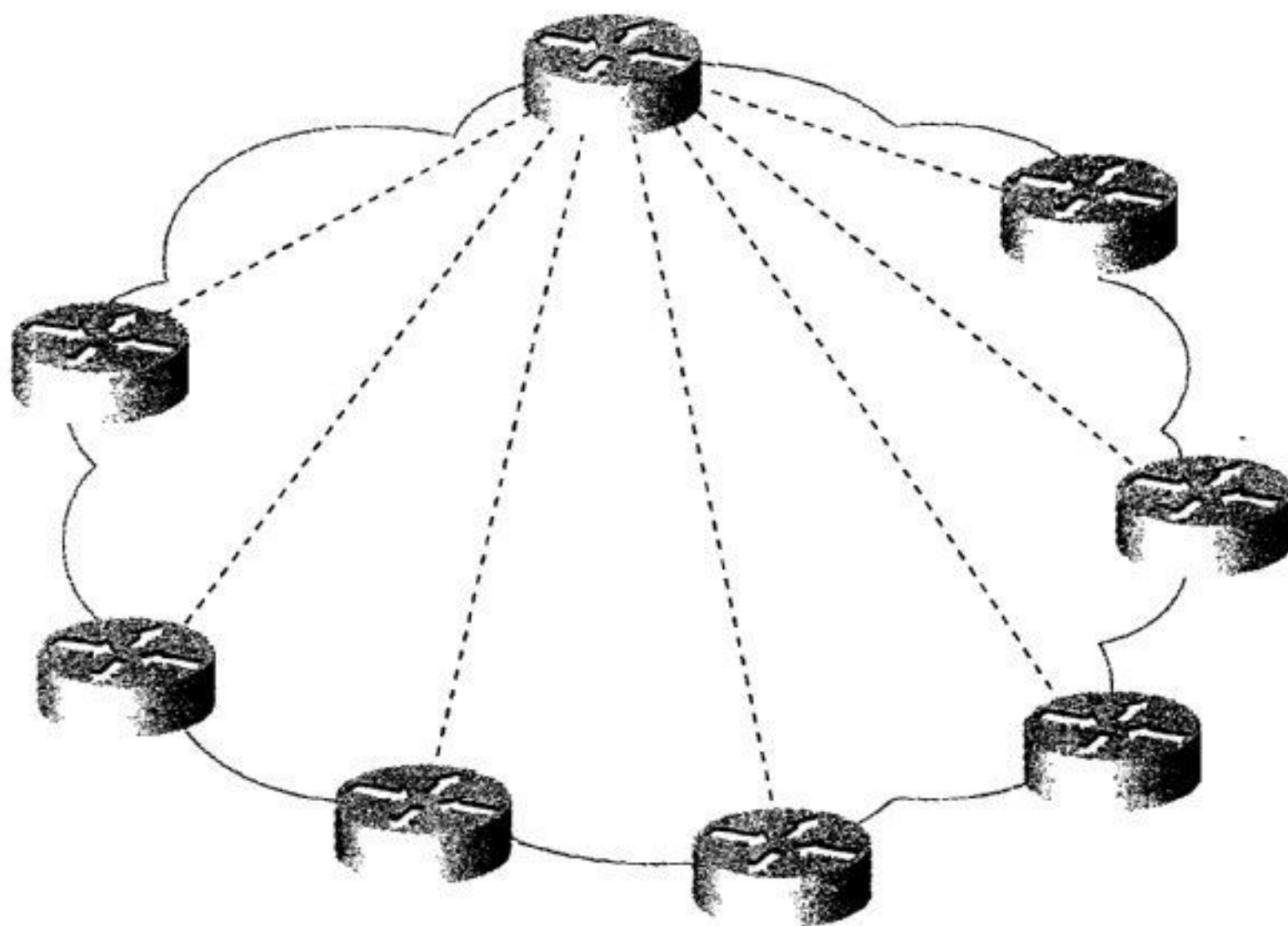


图 7-5 VLSM 允许这些 PVC 虚电路中的每一个电路都配置一个单独的子网，而不浪费主机地址

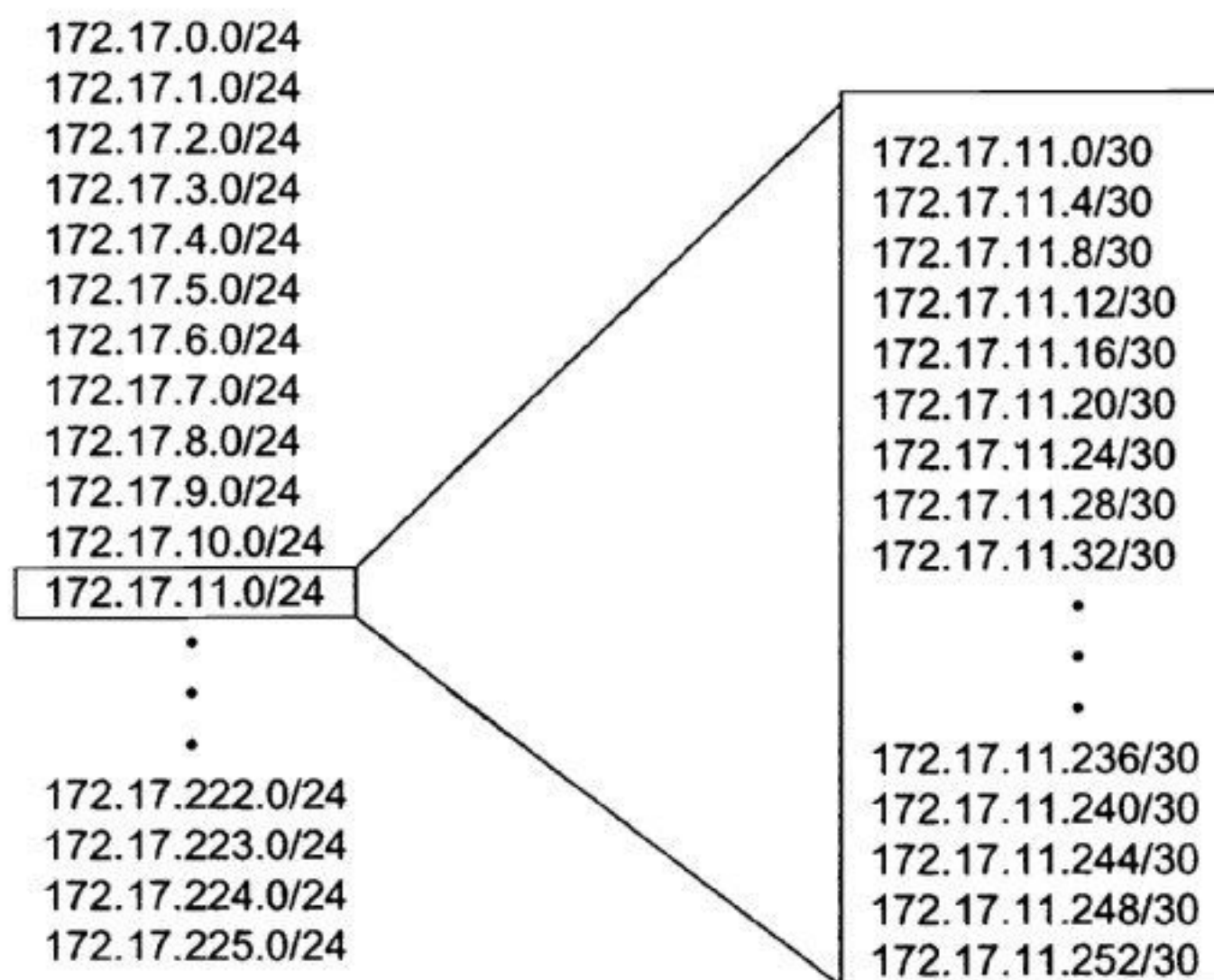


图 7-6 这个 B 类地址使用了 24 的掩码来划分子网，对子网 172.17.11.0 使用 30 位的掩码进一步地划分更小的子网，结果有 64 个子网可以提供给相关的点到点链路使用

7.1.6 认证

涉及到所有路由选择协议的安全问题，是指一台路由器接受非法路由选择更新报文的可能性。非法的更新报文可能来自于试图破坏互联网络的攻击者，或试图通过欺骗路由器发送数据包到一个错误的目的地的方法来捕获数据包。更普遍的有害更新报文来自于出现故障的路由器。RIPv2 协议能够通过更新报文所包含的口令来验证某个路由选择更新报文的源的合法性。

RIPv2 是通过更改 RIP 消息中原来正常情况下应该是第一个路由条目的字段来支持认证的，如图 7-7。在含有认证的单个更新报文中，最大可以携带的路由条目就被减少到了 24 个。

认证的识别是通过设置地址族标识字段为全1 (0xFFFF) 来标识的。对于简单的口令认证, 认证的类型 (Authentication Type) 是2 (0x0002), 剩余的16个8bit字节字段携带一个最多有16个字符的口令, 可以由数字和字母混合组成。口令在字段中按照左对齐的方式, 如果一个口令小于16个8bit字节的长度, 那么字段中没有使用的位被设置成0来填充。

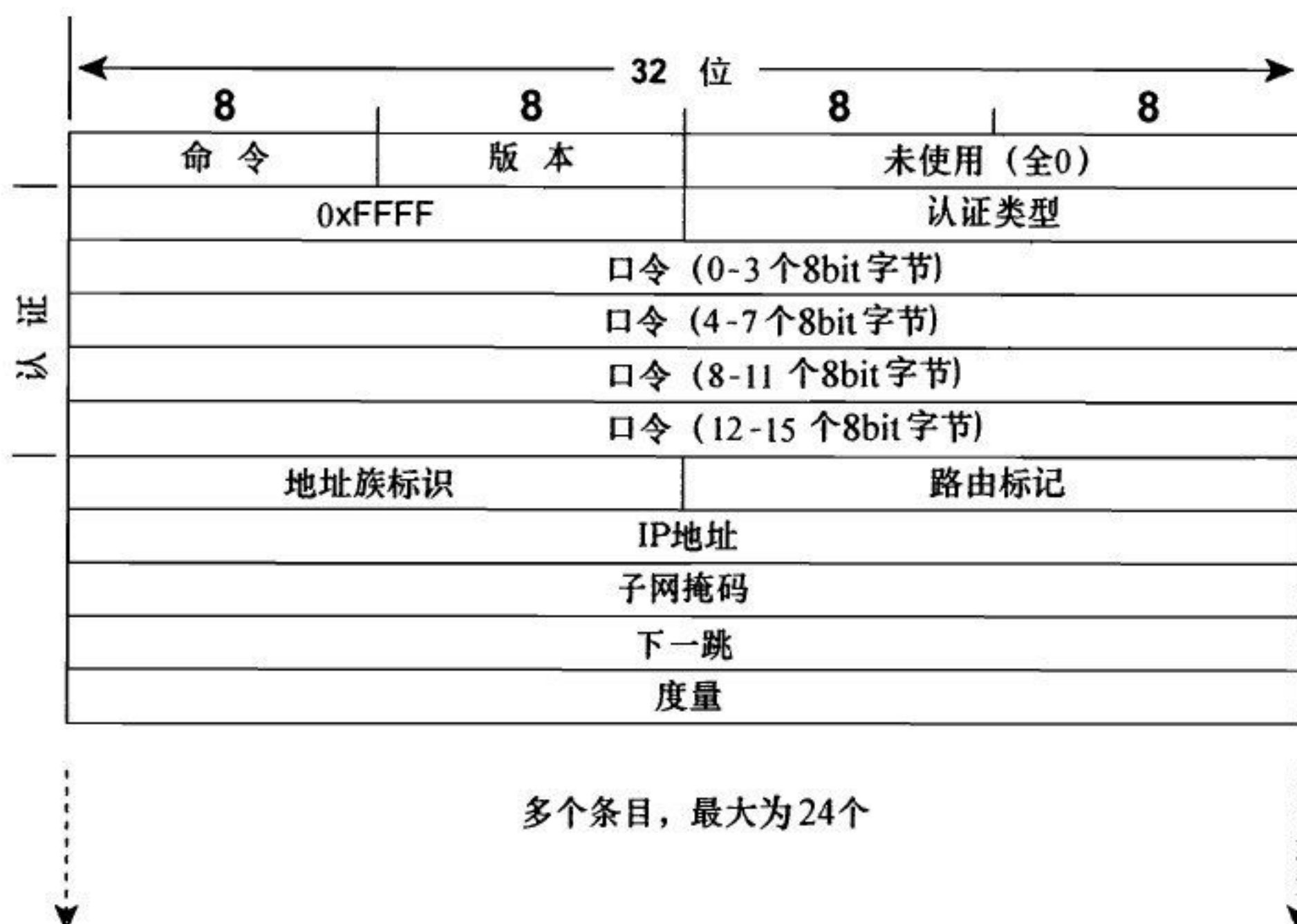


图 7-7 RIPv2 的认证信息, 配置是加载在第一个路由条目的字段空间上的

图 7-8 显示了协议分析仪捕获到的一个包含认证的 RIPv2 消息。这个图形也显示了使用缺省的 RIP 认证的一个安全隐患: 口令是明文传输的。任何人捕获到包含 RIPv2 更新消息的数据包, 都可以读出它的认证口令。

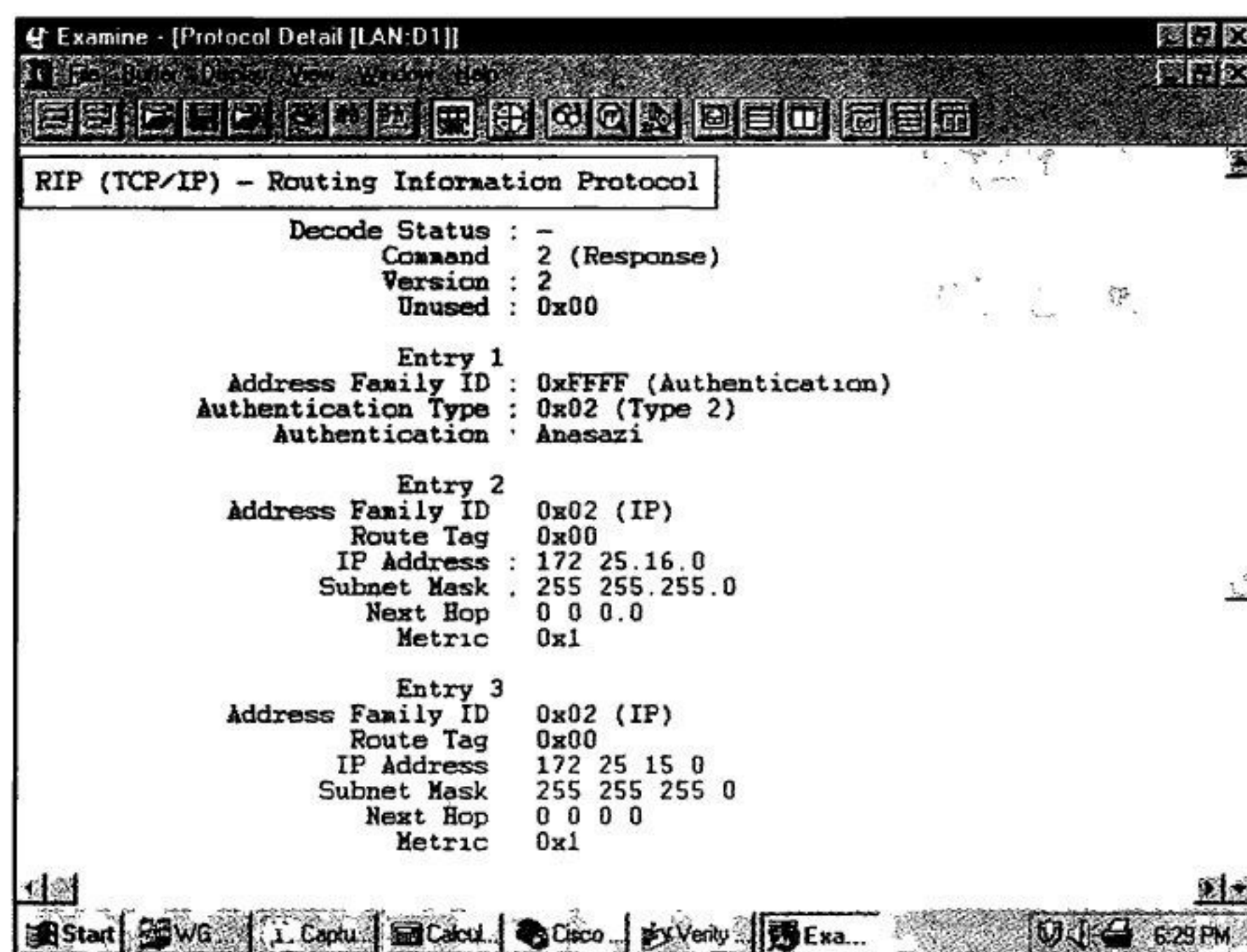


图 7-8 当使用简单的口令认证时, 口令是明文传送的, 因而可以被任何人通过探测包含更新的数据包读出口令

虽然 RFC1723 只描述了简单口令认证, 但却很有远见地提供了一个认证类型 (Authentication Type) 字段。Cisco IOS 软件就是利用这个特定字段, 提供了用 MD5 认证来替代简单口令认证的选项。¹ Cisco 使用了第一个和最后一个路由条目的字段空间, 从而达到了 MD5 认证的目的。

MD5 是一个单向的消息摘要 (message digest) 算法或安全散列函数 (secure hash function), 由 RSA Data Security, Inc 提出。有时候 MD5 也被作为一个加密校验和 (cryptographic checksum), 因为它的工作方式和算术校验和 (arithmetic checksum) 有点相似。MD5 算法是通过一个随意长度的明文消息 (例如, 一个 RIPv2 的更新报文) 和口令计算出一个 128 位的 hash 值的。这个“指纹”随同消息一起传送。拥有相同口令的接收者会计算它自己的 hash 值, 如果消息的内容没有被更改, 接收者的 hash 值应该和消息中发送者的 hash 值相匹配。

图 7-9 显示了图 7-8 中的同一台路由器的更新报文, 但是含有了 MD5 认证。这里的认证类型是 3, 并且看不到口令。注意, Cisco 使用了第一个和最后一个路由条目的字段空间来承载认证信息。因为这种用法不是开放的 RIPv2 协议标准的一部分, 协议分析仪显示出“认证在字段空间范围外”。

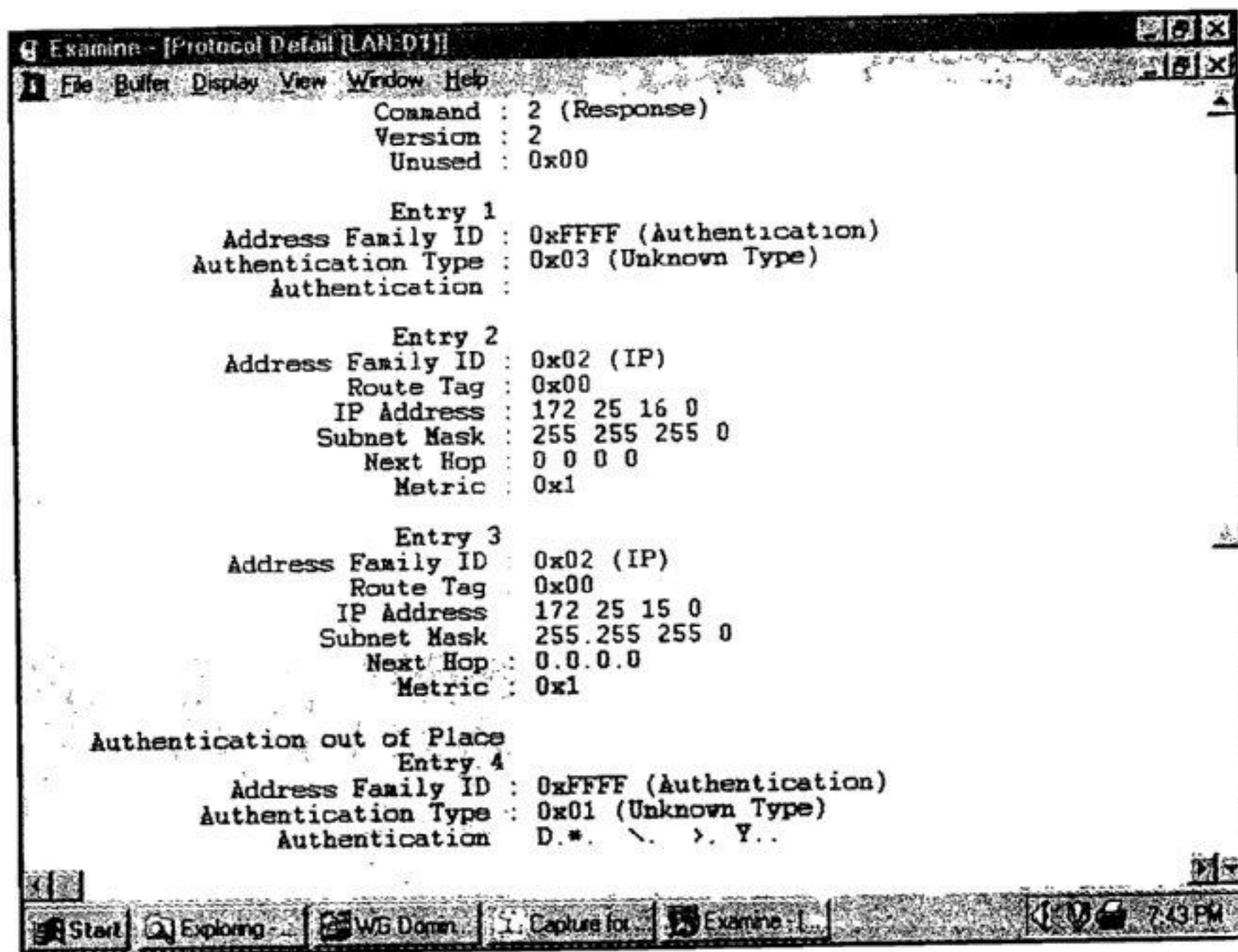


图 7-9 这个更新报文和图 7-8 中的更新来自于同一个路由器, 但是它使用了 MD5 认证

7.2 配置 RIPv2

由于 RIPv2 只不过是 RIPv1 的增强版, 而不是一个单独的协议, 因此, 在第 5 章中介绍的某些命令都可以以同样的方法在 RIPv2 中正确使用, 例如计时器和度量的操作、单播更新或根本不发出更新的配置等。在浏览一下 RIPv2 协议的配置后, 本节余下来的部分将集中讲

¹ MD5 是在 RFC1321 中描述的。欲更好地了解 MD5, 请参阅下面这本书: 《Network Security: Private Communication in a Public World》的第 120~122 页, 由 Charlie Kaufman、Radia Perlman 和 Mike Spencer 撰写, 1995 年 Prentice Hall 出版。

述一些新的扩展特性的配置。

7.2.1 案例研究 1: 一个基本的 RIPv2 配置

缺省时,在 Cisco 路由器上配置一个 RIP 进程将只发送 RIPv1 的消息,但是同时接收 RIPv1 和 RIPv2 的消息。这个缺省的配置可以通过命令 **version** 来更改,就像下面的例子一样:

```
router rip
  version 2
  network 172.25.0.0
  network 192.168.50.0
```

在这种配置方式下,路由器只发送和接收 RIPv2 的消息。同样地,路由器也可以配置成只发送和接收 RIPv1 消息的方式:

```
router rip
  version 1
  network 172.25.0.0
  network 192.168.50.0
```

可以在路由器配置模式 (config-router mode) 下键入命令 **no version** 恢复到原来的缺省方式。

7.2.2 案例研究 2: 与 RIPv1 的兼容性

RFC1723 中建议的基于接口级别 (interface-level) 的“兼容性开关”,在 Cisco IOS 软件中可以通过命令 **ip rip send version** 和 **ip rip receive version** 来实现。

在图 7-10 所示的互联网络里包含了同时宣告 RIPv1 和 RIPv2 的路由器。另外,主机 Pojoaque 是一台运行“routed”进程的 Linux 主机,它只能理解和处理 RIPv1。路由器 Taos 的配置是:

```
interface Ethernet0
  ip address 192.168.50.129 255.255.255.192
  ip rip send version 1
  ip rip receive version 1
!
interface Ethernet1
  ip address 172.25.150.193 255.255.255.240
  ip rip send version 1 2
!
interface Ethernet2
  ip address 172.25.150.225 255.255.255.240
!
router rip
  version 2
  network 172.25.0.0
  network 192.168.50.0
```

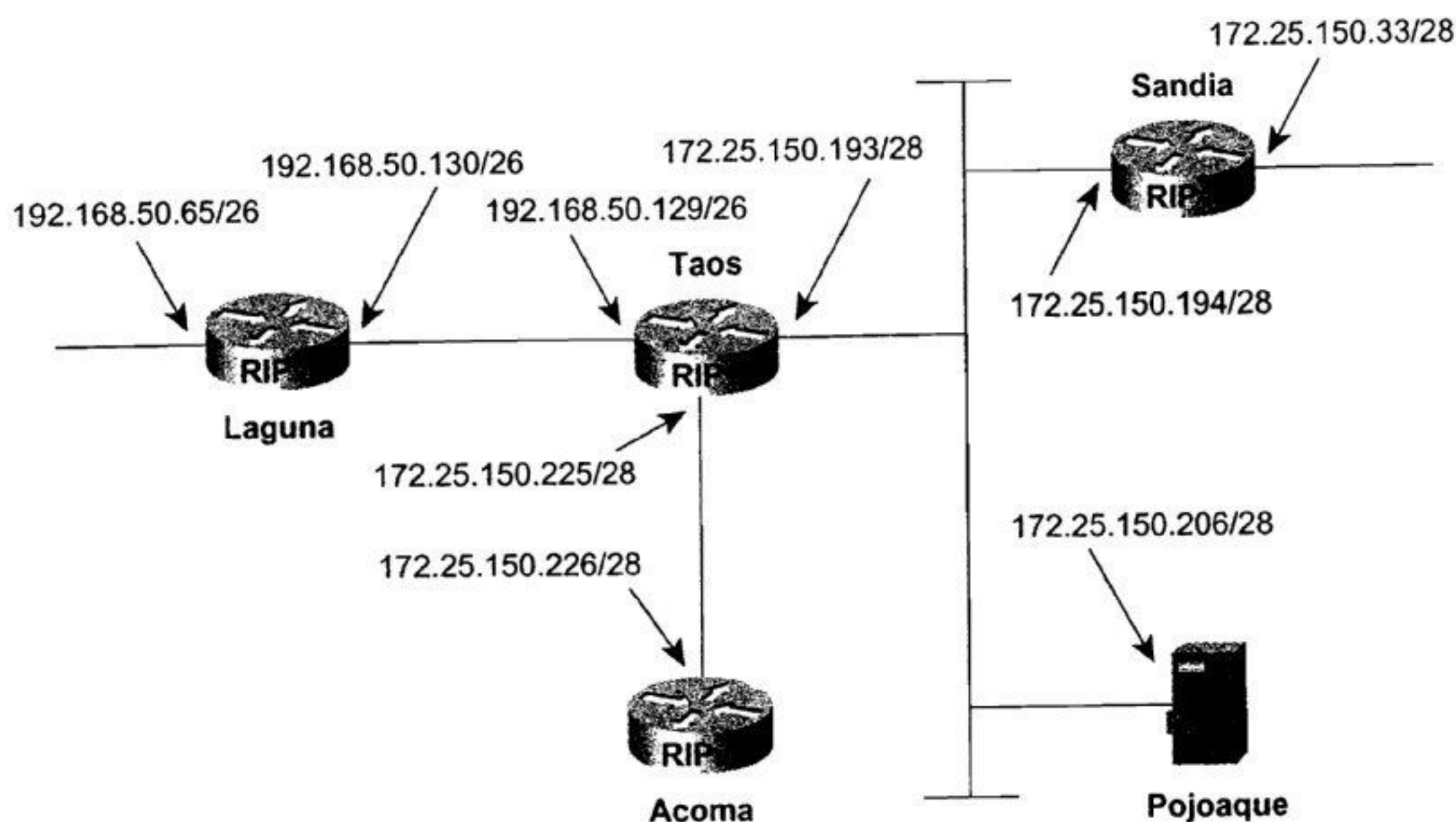



图 7-10 路由器 Taos 正在运行 RIPv2, 但是也必须宣告版本 1 给其他一些设备

因为路由器 Laguna 是 RIPv1 的一个宣告者, 所以路由器 Taos 的 E0 接口要配置成接收和发送 RIPv1 更新的方式, 而 E1 接口则配置成同时支持版本 1 和版本 2 更新的方式, 以便发送给运行 RIPv1 的路由器 Pojoaque 和运行 RIPv2 的路由器 Sandia。E2 接口没有什么特别的配置, 它将按照缺省方式发送和接收版本 2 的更新。

在图 7-11 中, 使用 **debug ip rip** 命令观察路由器 Taos 发送和接收的消息报文。这里有几个有趣的地方。首先, 注意观察 RIPv1 和 RIPv2 消息携带的内容不同。RIPv2 的更新中可以看到地址掩码、下一跳和路由标志 (在这个实例中, 都被设置成全 0)。其次, 可以观察到, E1 接口正在以广播方式发送 RIPv1 更新, 而以组播方式发送 RIPv2 更新。第三, 由于路由器 Taos 没有配置接收 RIPv1, 从而来自路由器 Pojoaque (172.25.150.206) 的更新就被忽略了 (路由器 Pojoaque 被配置错误了, 并且正在广播它的路由选择表)。¹

图 7-11 中, 可能最需要关注的就是广播到主机 Pojoaque 的更新了, 它没有包含子网 172.25.150.32。路由器 Taos 是通过组播方式的 RIPv2 更新从路由器 Sandia 学习这个子网的。但是主机 Pojoaque 由于只宣告 RIPv1 而不能接收这些组播。此外, 虽然路由器 Taos 得知这个子网, 但是水平分隔法则禁止路由器 Taos 把从这个接口学到的路由再从相同的接口通告出去。

因此, 主机 Pojoaque 无法得知子网 172.25.150.32。这里有可供使用的两种修正的方法: 第一, 路由器 Sandia 可以配置成可以同时发送 RIP 协议的两个版本; 第二, 可以通过下面的配置在路由器 Taos 的 E1 接口上关闭水平分隔:

```
interface Ethernet1
  ip address 172.25.150.193 255.255.255.240
  ip rip send version 1 2
  no ip split-horizon
```

¹ 实际上, 这个例子中使用 “routed -d” 选项故意配置错误的。


```

Taos#debug ip rip
RIP protocol debugging is on
Taos#
RIP: received v2 update from 172.25.150.194 on Ethernet1
      172.25.150.32/28 - 0.0.0.0 in 1 hops
RIP: ignored v1 packet from 172.25.150.206 (illegal version)
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.50.129)
      network 172.25.0.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Ethernet1 (172.25.150.193)
      subnet 172.25.150.224, metric 1
      network 192.168.50.0, metric 1
RIP: sending v2 update to 224.0.0.9 via Ethernet1 (172.25.150.193)
      172.25.150.224/28 - 0.0.0.0, metric 1, tag 0
      192.168.50.0/24 - 0.0.0.0, metric 1, tag 0
RIP: sending v2 update to 224.0.0.9 via Ethernet2 (172.25.150.225)
      172.25.150.32/28 - 0.0.0.0, metric 2, tag 0
      172.25.150.192/28 - 0.0.0.0, metric 1, tag 0
      192.168.50.0/24 - 0.0.0.0, metric 1, tag 0
RIP: received v1 update from 192.168.50.130 on Ethernet0
      192.168.50.64 in 1 hops
RIP: received v2 update from 172.25.150.194 on Ethernet1
      172.25.150.32/28 - 0.0.0.0 in 1 hops

```

图 7-11 使用调试功能，可以观察到路由器 Taos 上接收和发送的 RIP 版本

图 7-12 显示了更改配置后的结果。现在路由器 Taos 在它的更新里包含了子网 172.25.150.32。可以预见到关闭水平分隔后的一些可能的结果：路由器 Taos 现在不仅通告子网 172.25.150.32 给主机 Pojoaque，而且把这个子网通告回路由器 Sandia。

```

Taos#debug ip rip
RIP protocol debugging is on
Taos#
RIP: ignored v1 packet from 172.25.150.206 (illegal version)
RIP: received v2 update from 172.25.150.194 on Ethernet1
      172.25.150.32/28 -> 0.0.0.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.50.129)
      network 172.25.0.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Ethernet1 (172.25.150.193)
      subnet 172.25.150.32, metric 2
      subnet 172.25.150.224, metric 1
      subnet 172.25.150.192, metric 1
      network 192.168.50.0, metric 1
RIP: sending v2 update to 224.0.0.9 via Ethernet1 (172.25.150.193)
      172.25.150.32/28 -> 172.25.150.194, metric 2, tag 0
      172.25.150.224/28 -> 0.0.0.0, metric 1, tag 0
      172.25.150.192/28 -> 0.0.0.0, metric 1, tag 0
      192.168.50.0/24 -> 0.0.0.0, metric 1, tag 0

```

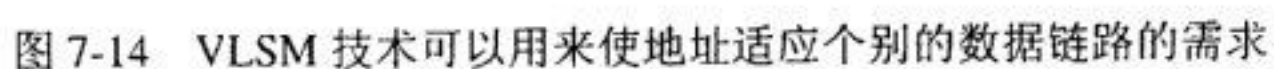
图 7-12 在 E1 接口上关闭水平分隔后，路由器 Taos 在通告给主机 Pojoaque 的更新中包含了子网 172.25.150.32

7.2.3 案例研究 3：使用 VLSM

参见图 7-10，子网 172.25.150.0/24 已经分配给图中的网络，这个子网通过扩展到 28 位的掩码进一步子网化以满足不同的数据链路。图 7-13 中显示了以二进制和点分十进制表示的

11111111111111111111111111111111	=	255.255.255.240
10101100000110011001011100000000	=	172.25.150.0/28
101011000001100110010111000010000	=	172.25.150.16/28
101011000001100110010111000100000	=	172.25.150.32/28
101011000001100110010111000110000	=	172.25.150.48/28
101011000001100110010111001000000	=	172.25.150.64/28
101011000001100110010111001010000	=	172.25.150.80/28
101011000001100110010111001100000	=	172.25.150.96/28
101011000001100110010111001110000	=	172.25.150.112/28
101011000001100110010111010000000	=	172.25.150.128/28
101011000001100110010111010010000	=	172.25.150.144/28
101011000001100110010111010100000	=	172.25.150.160/28
101011000001100110010111010110000	=	172.25.150.176/28
101011000001100110010111011000000	=	172.25.150.192/28
101011000001100110010111011010000	=	172.25.150.208/28
101011000001100110010111011100000	=	172.25.150.224/28
101011000001100110010111011110000	=	172.25.150.240/28

在图 7-14 中，路由器 Taos 增加了一个包含 60 台主机的令牌环网段。为了给这个数据链路分配地址，需要有至少 6 个主机位的子网。有类别路由选择协议如果使用次一级的地址，



Created with Unregistered
Version of FotoBatch
Get it at keksoft.com

应需要 5 个图 7-13 中的子网分配给令牌环网段【 $5 \times (2^4 - 2) = 70$ 】。利用无类别路由选择协议和 VLSM 技术, 只要将图 7-13 中的 4 个子网合成单个 26 位掩码的子网就可以满足需求了, 这一个步骤可以提供 6 个主机位 (62 个主机地址), 并且没有必要使用次一级的编址。这 4 个子网 172.25.150.64/28~172.25.150.112/28 可以合并成单个 26 位掩码的子网 172.25.150.64/26。注意, 这里的 4 个子网并不是随意选择的, 它们是在这 16 个子网中具有相同的和惟一的起始 26 位掩码位的那些子网。¹

还是参看图 7-14, 图中互联网络增加了 4 个路由器和 4 条串行链路。没有 VLSM 技术的话, 4 条串行链路就需要使用图 7-13 中的 4 个子网; 有了 VLSM 技术, 在图 7-13 中使用单个子网就可以满足所有 4 条串行链路的需求了。选用子网 172.25.150.240, 利用 30 位的掩码创建图 7-15 中的子网, 4 个小的子网的每一个都包含两个主机地址。

划分子网的基本目的总是相同的: 路由器必须能够使用惟一的地址来标识每一条数据链路, 以区别于互联网络中的其他地址, 这也是前面两个例子的共同目的。在第一个例子中, 通过减小掩码的大小, 将多个地址合并成一个地址, 这个操作一直进行到余下的所有的地址都拥有共同的位。注意, 这种情况在子网被汇总成主网络地址时也会发生。在第二个例子中, 通过扩展子网掩码将单个子网划分为多个更小的子网。

11111111111111111111111111111100	=	255.255.255.252
1010110000011001100101101110000	=	172.25.150.240/30
1010110000011001100101101110100	=	172.25.150.244/30
1010110000011001100101101111000	=	172.25.150.248/30
1010110000011001100101101111100	=	172.25.150.252/30

图 7-15 30 位的掩码应用于子网 172.25.150.240

7.2.4 案例研究 4: 不连续的子网和无类别路由选择

图 7-16 显示了 4 台新添路由器中的每一台路由器都和两个以太网相连。在每一处, 其中的一个以太网都是子网 172.25.150.0/24 的成员, 而且都不超过 12 台主机。这个需求很容易满足, 选用图 7-13 中 4 个未用的子网进行分配即可。

在每一处的另一个以太网是网络 192.168.50.0 的成员, 并且都不超过 25 台主机。子网 192.168.50.64/26 和 192.168.50.128/26 正在被其他链路使用, 只剩下了子网 192.168.50.0/26 和 192.168.50.192/26。通过增加掩码位到 27 位, 这两个子网就被划分成了 4 个子网, 每个子网有 5 个主机位——每个子网可以提供 30 个主机地址。图 7-17 显示了二进制表示的这 4 个子网。

分配完所有的子网地址后, 下一个所关注的就是这样一个事实: 网络 192.168.50.0 的子网是不连续的。第 5 章展示了一个不连续子网的实例, 并演示了是怎样使用辅助接口连接它们的。而无类别路由选择协议没有关于不连续子网的这些困难。因为每一条路由更新都包含一个子网掩码, 因而一个主网络的子网能够通告给另一个主网络。

但是, RIPv2 协议缺省的行为要在主网络边界上进行路由汇总, 这一点和 RIPv1 相同。为了关闭路由汇总功能以允许被通告的子网通过主网络边界, 可以在 RIP 的处理中使用 **no**

¹ 将几个地址合并成一个地址的技巧将在第 8 章的地址聚合中介绍。

必须要关闭路由汇总。回忆图 7-10 中，路由器 Laguna 正运行的是 RIPv1，为了使这个配置可以工作，必须更改成版本 2。

应该仔细考虑一下，可变长子网掩码对仍旧运行 RIPv1 的主机 Pojoaque 产生了什么影响。图 7-18 的调试信息显示了路由器 Taos 发送到子网 172.25.150.192/28 上的版本 1 和版本 2 的更新信息。版本 1 的更新信息仅仅包含那些 28 位掩码的子网，这与正在广播更新的子网的掩码是一样的。虽然主机 Pojoaque 不接收子网 172.25.150.64/26 或所有串行链路的子网的通告，但是在这个实例中，关于那些子网地址的分析显示，主机 Pojoaque 依然可以正确地识别这些不同于它本身子网的地址，因而，到达这些子网的数据包将会通过路由选择送到路由器 Taos 上。

```
Taos#debug ip rip
RIP protocol debugging is on
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (172.25.150.193)
  subnet 172.25.150.0, metric 3
  subnet 172.25.150.16, metric 3
  subnet 172.25.150.32, metric 2
  subnet 172.25.150.48, metric 3
  subnet 172.25.150.128, metric 3
  subnet 172.25.150.192, metric 1
  subnet 172.25.150.224, metric 1
  network 192.168.50.0, metric 1
RIP: sending v2 update to 224.0.0.9 via Ethernet0 (172.25.150.193)
  172.25.150.0/28 -> 0.0.0.0, metric 3, tag 0
  172.25.150.16/28 -> 0.0.0.0, metric 3, tag 0
  172.25.150.32/28 -> 0.0.0.0, metric 2, tag 0
  172.25.150.48/28 -> 0.0.0.0, metric 3, tag 0
  172.25.150.64/26 -> 0.0.0.0, metric 1, tag 0
  172.25.150.128/28 -> 0.0.0.0, metric 3, tag 0
  172.25.150.192/28 -> 0.0.0.0, metric 1, tag 0
  172.25.150.224/28 -> 0.0.0.0, metric 1, tag 0
  172.25.150.240/30 -> 0.0.0.0, metric 2, tag 0
  172.25.150.244/30 -> 0.0.0.0, metric 2, tag 0
  172.25.150.248/30 -> 0.0.0.0, metric 2, tag 0
  172.25.150.252/30 -> 0.0.0.0, metric 2, tag 0
  192.168.50.0/27 -> 0.0.0.0, metric 3, tag 0
  192.168.50.32/27 -> 0.0.0.0, metric 3, tag 0
  192.168.50.64/26 -> 0.0.0.0, metric 2, tag 0
  192.168.50.128/26 -> 0.0.0.0, metric 1, tag 0
  192.168.50.192/27 -> 0.0.0.0, metric 3, tag 0
  192.168.50.224/27 -> 0.0.0.0, metric 3, tag 0
```

图 7-18 虽然来自路由器 Taos 的 RIPv2 更新包含了图中网络所有的子网，但是 RIPv1 的更新信息却仅仅包含到达网络 192.168.50.0 的汇总路由和网络 172.25.150.0 的一些子网（这些子网的掩码和发送这些更新的接口的掩码相同）

7.2.5 案例研究 5：认证

Cisco 实现 RIPv2 的消息报文的认证包含了两种选择——简单的口令或 MD5 认证。另外，也包含了在一个“钥匙链”上定义多个钥匙或口令的选项。这样路由器就可以在不同的时候配置不同的钥匙。

设置 RIPv2 认证的步骤如下：

步骤 1：定义一个带名字的钥匙链：

- 步骤 2: 定义在钥匙链上的钥匙;
- 步骤 3: 在接口上启动认证并指定使用的钥匙链;
- 步骤 4: 指定这个接口使用明文认证还是 MD5 认证;
- 步骤 5: 可选地配置钥匙的管理。

在下面的例子中, 路由器 Taos 上配置了一个名为 “Tewa” 的钥匙链, 这个钥匙链上惟一的一个钥匙是 “Key 1”, 它含有一个口令 “Kachina”。接着, E0 接口利用 MD5 认证的这个钥匙去验证来自路由器 Laguna 的更新报文。

```
Taos(config)#key chain Tewa
Taos(config-keychain)#key 1
Taos(config-keychain-key)#key-string Kachina
Taos(config-keychain-key)#interface ethernet 0
Taos(config-if)#ip rip authentication key-chain Tewa
Taos(config-if)#ip rip authentication mode md5
```

即使只有一个钥匙, 也必须配置钥匙链。虽然交换带认证的更新报文的所有路由器必须拥有相同的口令, 但是钥匙链的名字却只在本地路由器上有意义。例如, 路由器 Laguna 可以有一个名为 Keres 的钥匙链, 但是宣告给路由器 Taos 的钥匙的字符串必须是 Kachina。

如果没有添加口令 **ip rip authentication mode md5**, 接口将使用缺省的明文认证。虽然在与一些 RIPv2 的设备通信时明文认证可能是必要的, 但是只要有可能, 几乎总是明智地使用安全性能好得多的 MD5 认证。

钥匙管理 (Key management) 用来做从一个认证钥匙到另一个认证钥匙的迁移工作的。在下面的例子中, 路由器 Laguna 的配置是, 在 1997 年 11 月 28 日下午 4:30 开始使用第一个钥匙, 使用的时长是 12 小时 (43200s); 第二个钥匙从 1997 年 11 月 29 日凌晨 4:00 开始生效, 并一直使用到 1998 年 4 月 15 日下午 1:00; 第三个钥匙从 1998 年 4 月 15 日下午 12:30 开始生效, 并在这个时间以后永久有效。

```
key chain Keres
key 1
key-string Kachina
accept-lifetime 16:30:00 Nov 28 1997 duration 43200
send-lifetime 16:30:00 Nov 28 1997 duration 43200
key 2
key-string Kiva
accept-lifetime 04:00:00 Nov 29 1997 13:00:00 Apr 15 1998
send-lifetime 04:00:00 Nov 29 1997 13:00:00 Apr 15 1998
key 3
key-string Koshare
accept-lifetime 12:30:00 Apr 15 1998 infinite
send-lifetime 12:30:00 Apr 15 1998 infinite
!
interface Ethernet0
ip address 198.168.50.130 255.255.255.192
ip rip authentication key-chain Keres
ip rip authentication mode md5
```


正如配置所显示的, 从其他路由器接受的口令和发送消息所使用的口令在管理上是分离的。因此, 使用 **accept-lifetime** 和 **send-lifetime** 命令都应该含有一个指定的开始时间和一个指定的持续时间或结束时间, 或者指定关键字 *infinite*。钥匙的号码按照从最低到最高的顺序检查, 使用第一个有效的钥匙。

虽然这个配置可以使用 30min 的时间重叠来在不同的系统时钟之间进行校正, 但是, 这里强烈建议在对钥匙的管理时, 使用像网络时钟协议 (Network Time Protocol, NTP) 这样的时钟同步协议 (Time Synchronization Protocol)。¹

7.3 RIPv2 故障排除

对于 RIPv2 协议, 有两个配置问题, 即版本不匹配和认证配置错误。这两个难点都可以比较容易地从调试信息中发现, 如图 7-19 所示。

```
Jemez#debug ip rip events
RIP event debugging is on
Jemez#
RIP: ignored v1 packet from 172.25.150.249 (illegal version)
RIP: ignored v2 packet from 172.25.150.249 (invalid authentication)
Jemez#
```

图 7-19 通过调试信息来发现不匹配的版本和配置错误的认证问题

对于 RIPv2 协议或者任何无类别路由选择协议来说, 更有可能出问题的原因是配置了一个错误的可变长子网掩码。VLSM 并不难, 但是如果 VLSM 的规划没有小心地设计和管理, 它就会带来一些奇怪的路由选择困难。

案例研究: 配置错误的 VLSM

图 7-20 中的主机 C 不能通过互联网络进行通信, 甚至无法在本地数据链路上 ping 通其他的主机或路由器。而主机 A 和主机 B 的相互通信没有问题, 也和互联网络上的其他主机能够正常通信, 但是它们都不能和主机 C 通信。这里, 所有的主机都把 172.19.35.1 配置成自己的缺省网关地址。

如图 7-21, 当从主机 A 或主机 B 上试图去 ping 主机 C 时, 发现第一个 ping 包是成功的, 但是后续的 ping 包是失败的。显然, 至少有一个 ICMP 的 Echo 请求包能够到达主机 C, 并且至少有一个 Echo 响应包可以返回给 ping 包的源主机, 这个事实说明, 网络的故障与硬件或数据链路没有太大关系。

这个奇怪的 ping 的行为可以让我们作出这样一个假设: 在第一个 ping 包成功后, 后续的 ping 包——不论是主机 B 发出的 Echo 请求包, 还是主机 C 返回的 Echo 响应包不知由于什么原因被误导了。因为这个情况发生在本地的数据链路上, 因此应该检查一下 ARP (Address Resolution Protocol, 地址解析协议) 的缓冲区 (Cache)。

图 7-22 和图 7-23 分别显示了主机 B 和主机 C 的 ARP 缓冲区。关于 ARP 的猜疑在这里得到证实, 主机 C 的 ARP 缓冲区包含了主机 B 的正确 MAC 地址 (00a0.2470.febd), 但是主

¹ NTP 协议超出了本书的讲述范围, 请参考 Cisco 配置手册以便获取更多的信息。

机 B 的缓冲区有一个和主机 C 的 IP 地址相关联的 MAC 地址 (0000.0c0a.2aa9)。进一步地观察这两个缓冲区, 显示出 MAC 地址 0000.0c0a.2aa9 是路由器 San_Felipe 的本地接口的 MAC 地址, 这个信息可以从以下事实推导出来: 通过路由器 San_Felipe 可以到达的目的 IP 地址和 IP 地址 172.19.35.2 映射到相同的 MAC 地址上了。

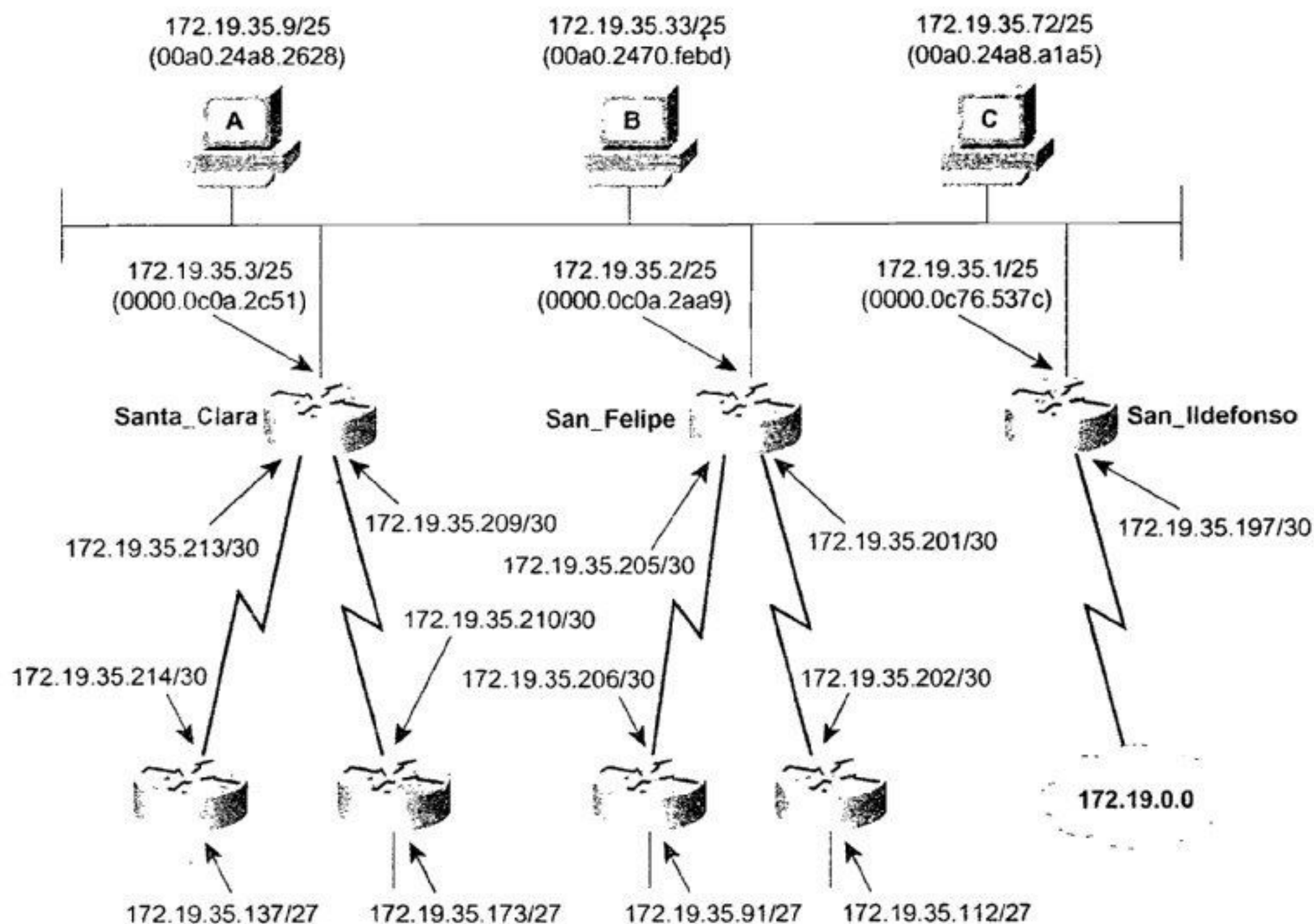


图 7-20 主机 A 和主机 B 能够通过互连网络进行通信, 但是都不能和主机 C 通信

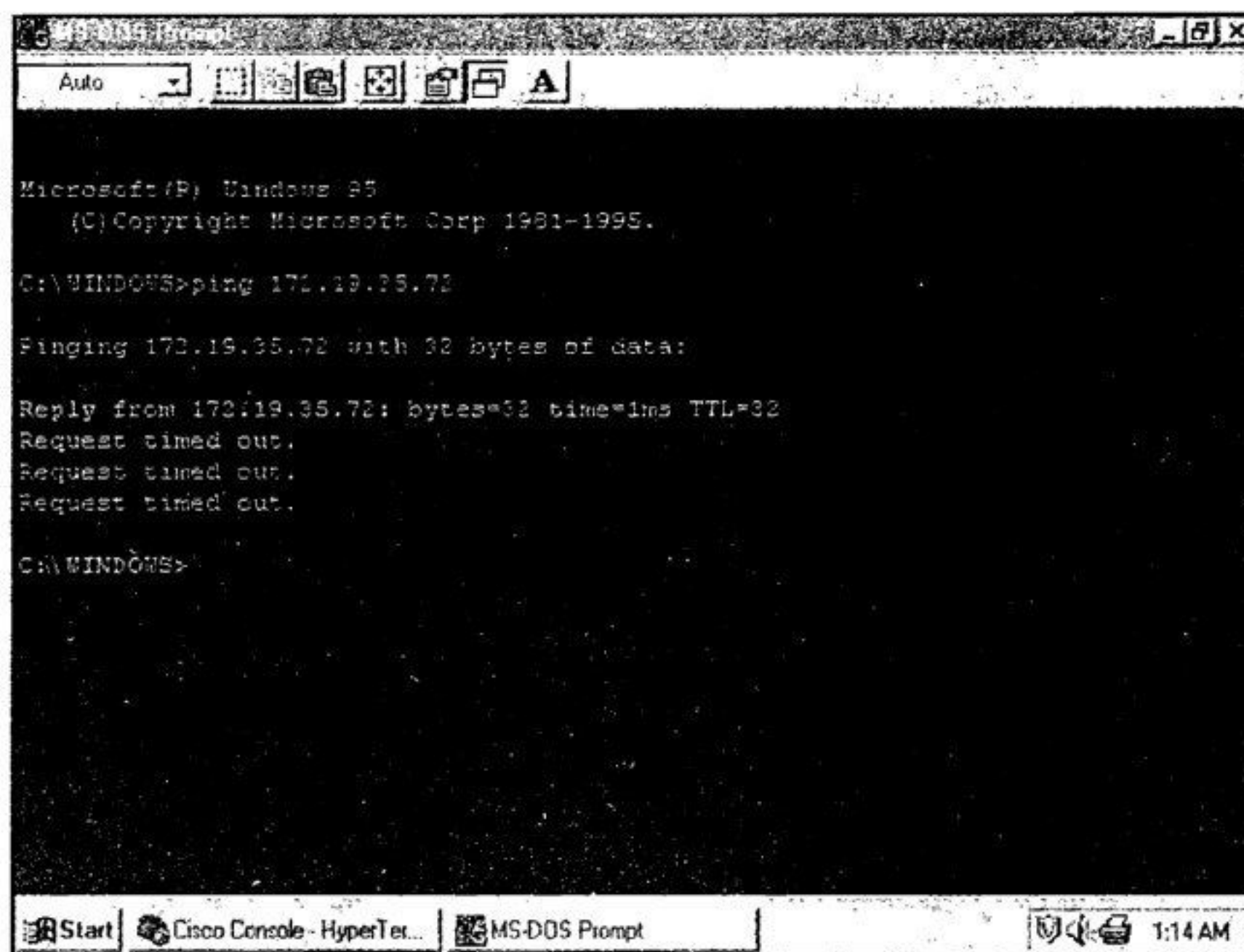


图 7-21 当主机 B 试图 ping 主机 C 时, 第一个 ping 包是成功的, 而后续的 ping 包是失败的


```
Linux 1.2.13 (Zuni.Pueblo.com) (tty0)

Zuni login: root
Password:
Last login: Sat Nov 29 11:21:57 on tty1
Linux 1.2.13.
You have mail.
Zuni:~# arp -a
```

Address	HW type	HW address	Flags	Mask
172.19.35.112	10Mbps Ethernet	00:00:0C:0A:2A:A9	C	*
172.19.35.1	10Mbps Ethernet	00:00:0C:76:5B:7C	C	*
172.19.35.33	10Mbps Ethernet	00:A0:24:70:FE:BD	C	*
172.19.35.2	10Mbps Ethernet	00:00:0C:0A:2A:A9	C	*
172.19.35.3	10Mbps Ethernet	00:00:0C:0A:2C:51	C	*
172.19.35.9	10Mbps Ethernet	00:A0:24:A8:26:28	C	*
172.19.35.91	10Mbps Ethernet	00:00:0C:0A:2A:A9	C	*

```
Zuni:~#
```

图 7-22 主机 C 的 ARP 缓冲区正确地显示了和所有地址相关的 MAC 地址



图 7-23 主机 B 的 ARP 缓冲区显示，主机 C 的 IP 地址被映射到了路由器 San_Felipe 的本地接口 172.19.35.2 的 MAC 地址上

现在 ping 的结果就比较清楚了。首先，主机 B 广播了一个 IP 地址为 172.19.35.72 的 ARP 请求，然后主机 C 发送一个 ARP 响应包，因而主机 B 发送的第一个 ping 包是正确的。在这期间，路由器 San_Felipe 也收到了那个 ARP 的请求包，很显然它认为自己有一条到达地址 172.19.35.72 的路由，于是路由器 San_Felipe 就用 ARP 代理（Proxy ARP）作出响应（滞后于主机 C 是因为路由器最初不得不执行一次路由的查找），这就导致主机 B 覆盖了主机 C 的 MAC 地址。后来的 Echo 请求包就被发送给路由器 San_Felipe 了，而路由器 San_Felipe 将把这个请求包从本地链路路由出去，最终丢弃掉，连接在这个以太网链路上的协议分析仪证实了这一点（如图 7-24）。

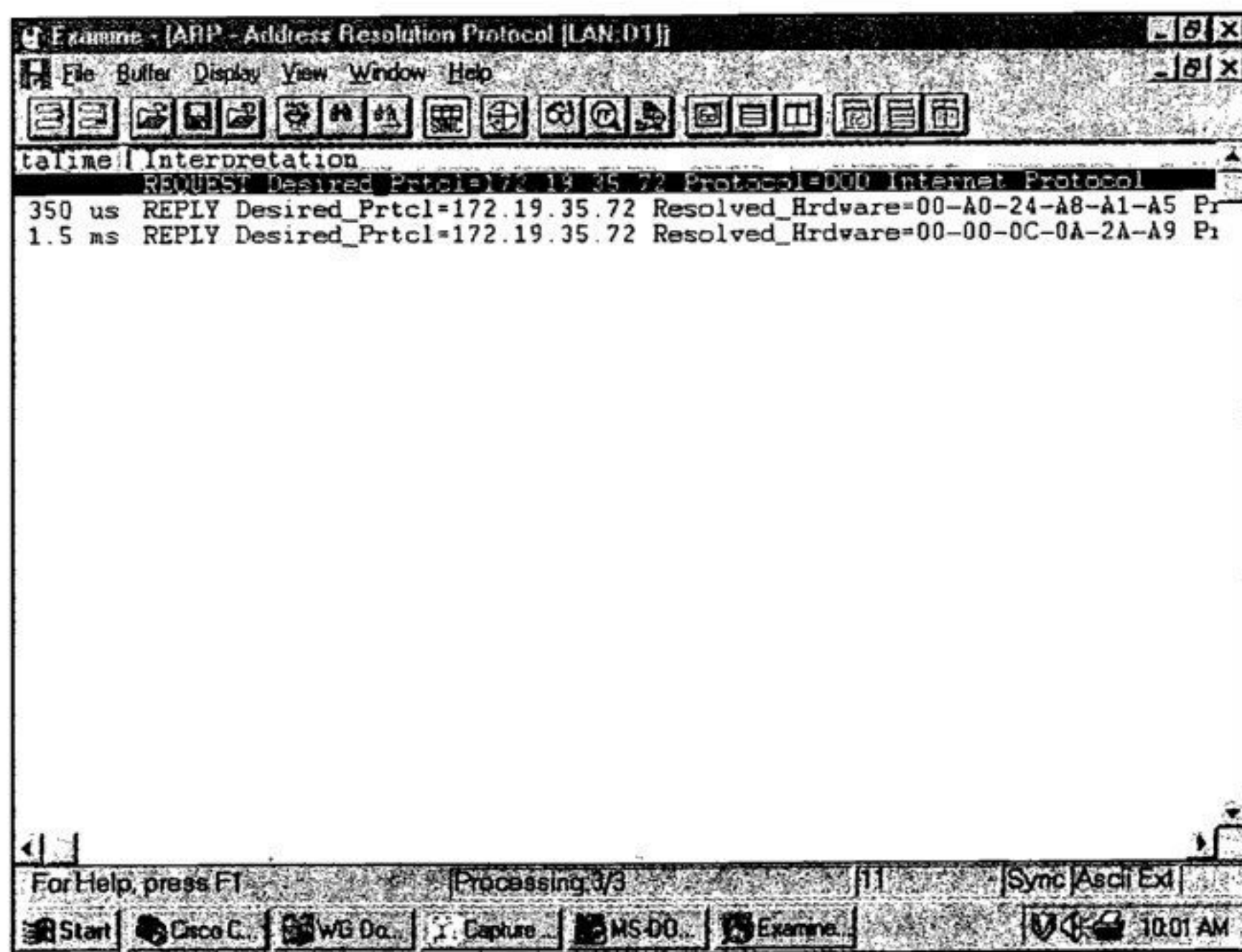


图 7-24 协议分析仪过滤出的 ARP 数据包显示, 主机 B 到主机 C 的 ARP 请求和从主机 C (00a0.24a8.a1a5) 与路由器 San_Felipe (0000.0c0a.2aa9) 返回的响应

如果了解了故障是由路由选择问题引起的, 那么余下的工作就只是要找出引起路由选择问题的原因了。首先, 应该确定一下每一条数据链路的子网地址, 如图 7-25。其次, 基于二进制的表示, 应该把主机 C 的 IP 地址和从路由器 San_Felipe 可达的所有子网相对照, 来找出所有的地址冲突。如图 7-26, 用粗体字显示了子网地址的最后一个地址位组的子网位。

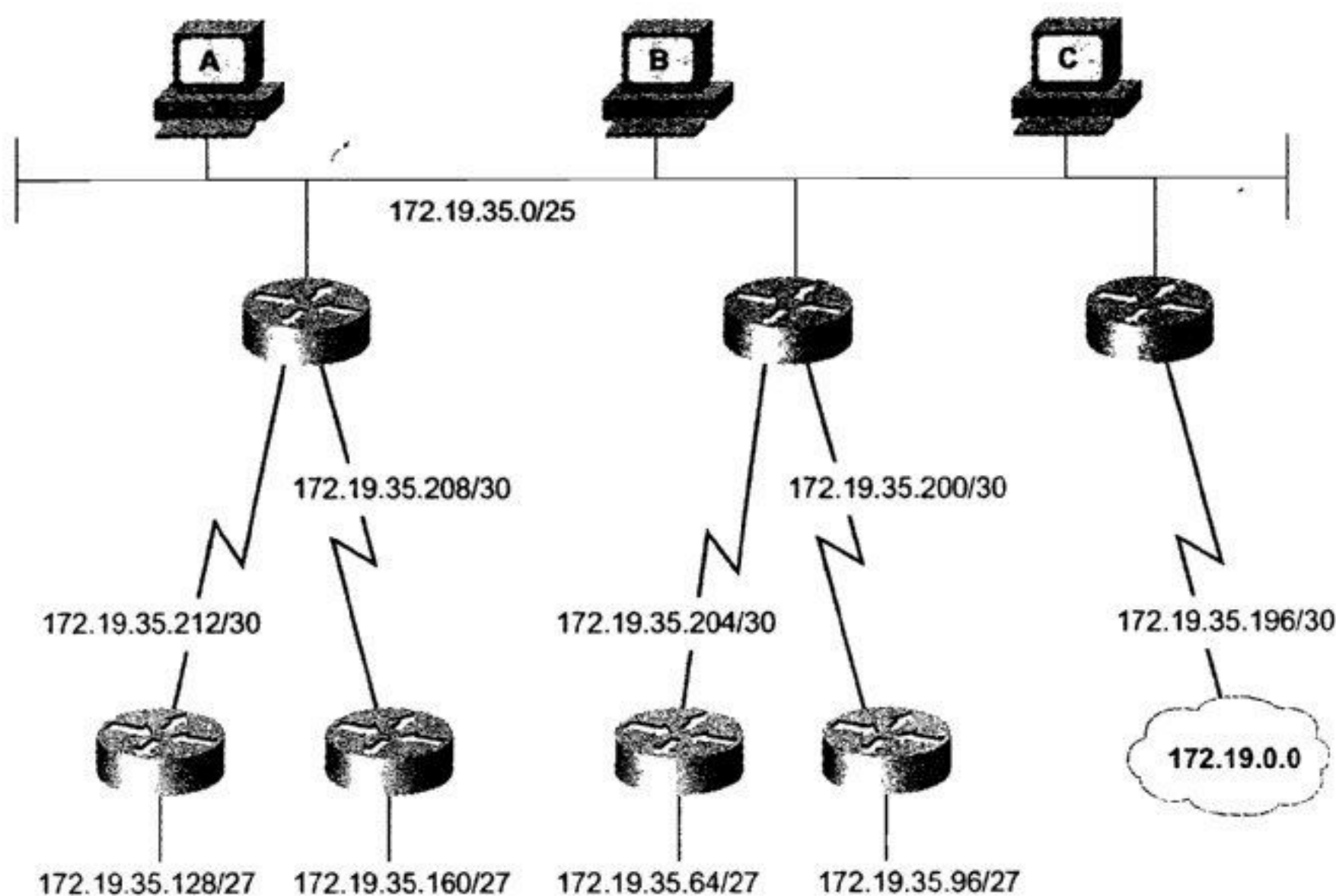


图 7-25 当分析任何地址编址的规划时, 特别是 VLSM 的设计, 应该先确定每一条数据链路的子网地址, 这样才能发现地址冲突和重叠的问题

经过比较以后, 显示出子网 172.19.35.72/25 的前 3 位和 172.19.35.64/27 的前 3 位是匹配的。路由器 San_Felipe 的路由选择表同时拥有 172.19.35.0/25 和 172.19.35.64/27 的路由 (如

图 7-27)。当路由器收到一个要到达主机 C 的数据包时,它可以和子网 172.19.35.0/25 匹配 1 个 bit 位,但却可以和子网 172.19.35.64/27 匹配 3 个 bit 位。结果,路由器将选择更具体的子网路由而把数据包从本地的数据链路上路由出去,最后丢弃这个数据包。

10101100000100110010001101001000	=	172.19.35.72/25
10101100000100110010001100000000	=	172.19.35.0/25
10101100000100110010001101000000	=	172.19.35.64/27
10101100000100110010001101100000	=	172.19.35.96/27
10101100000100110010001111001000	=	172.19.35.200/30
10101100000100110010001111001100	=	172.19.35.204/30

图 7-26 粗体字突出显示了主机 C 的 IP 地址的最后一个位组的子网位

```
San_Felipe#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
       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
Gateway of last resort is 172.19.35.1 to network 0.0.0.0
*
 172.19.0.0/16 is variably subnetted, 10 subnets, 3 masks
R   172.19.35.128/27 [120/1] via 172.19.35.3, 00:00:07, Ethernet0
R   172.19.35.160/27 [120/1] via 172.19.35.3, 00:00:08, Ethernet0
R   172.19.35.212/30 [120/1] via 172.19.35.3, 00:00:08, Ethernet0
R   172.19.35.208/30 [120/1] via 172.19.35.3, 00:00:08, Ethernet0
C   172.19.35.204/30 is directly connected, Serial0
C   172.19.35.200/30 is directly connected, Serial1
R   172.19.35.196/30 [120/1] via 172.19.35.1, 00:00:17, Ethernet0
C   172.19.35.0/25 is directly connected, Ethernet0
R   172.19.35.64/27 [120/1] via 172.19.35.206, 00:00:11, Serial0
R   172.19.35.96/27 [120/1] via 172.19.35.202, 00:00:23, Serial1
R*  0.0.0.0/0 [120/1] via 172.19.35.1, 00:00:18, Ethernet0
San_Felipe#
```

图 7-27 路由器 San_Felipe 同时拥有 172.19.35.0/25 和 172.19.35.64/27 的路由,第二个路由比第一个路由能够更好地匹配主机 C 的地址

这个故障的解决办法是,要么更改主机 C 的地址,要么更改子网 172.19.35.64。这一措施理论上听起来比较容易,但在实际的网络中,它会带来一些困难的决策,因为它涉及到基于这个案例研究的互联网络上的用户。

图 7-28 显示了网络 172.19.35.0 基于 27 位掩码的所有子网。这样做的目的是为了把最初的 4 个连续的子网捆绑成一个 25 位掩码的单个子网,以便容纳“骨干”以太网段上的最多 85 台主机。这个做法是有效的,因为地址的编组所使用的所有子网的首个子网位都为 0,因而没有其他的地址能够引起冲突;其次,子网 172.19.35.192/27 使用 30 位的掩码来划分更小的子网,用来给串行链路使用,又一次证明这个做法是有效的。子网 172.19.35.128/27 和子网 172.19.35.160/27 已经被使用了,当选择子网 172.19.35.64/27 和子网 172.19.35.96/27 给两个“远程”网段使用时,错误就产生了,因为这两个子网已经被宣告了。

这个困难的决策来自于,究竟是放弃骨干以太网段的地址空间进行从新编址,还是放弃那两个远程子网各自的地址空间进行从新编址?我们选择后者,利用一个 28 位的掩码将 172.19.35.224/27 划分成两个子网,供那两个远程子网使用。

11111111111111111111111111111111	=	255.255.255.224
10101100000100110010001100000000	=	172.19.35.0/27
10101100000100110010001100100000	=	172.19.35.32/27
10101100000100110010001101000000	=	172.19.35.64/27
10101100000100110010001101100000	=	172.19.35.96/27
10101100000100110010001110000000	=	172.19.35.128/27
10101100000100110010001110100000	=	172.19.35.160/27
10101100000100110010001111000000	=	172.19.35.192/27
10101100000100110010001111100000	=	172.19.35.224/27

图 7-28 一个 27 位的掩码应用于子网 172.19.35.0

7.4 展 望

虽然 RIPv2 协议比 RIPv1 协议有了很大的改进,但它依旧是受到最大 15 跳的跳数限制,因此也就仅仅适用于小型互联网络。第 8 章“增强型内部网关路由选择协议 (EIGRP)”、第 9 章“开放最短路径优先协议 (OSPF)”和第 10 章“集成 IS-IS 协议”将阐述在大型互联网络中使用的 3 种路由选择协议,再结合像 VLSM 这样的设计策略,就可以成为控制大型网络的强有力的工具。

7.5 总结表: 第 7 章命令总结

命 令	描 述
accept-lifetime <i>start-time</i> (<i>infinite</i>) <i>end-time</i> duration <i>seconds</i>	设置一个时间段, 用来指定钥匙链上的认证钥匙可被接受的有效时间
auto-summary	在网络边界打开或关闭自动路由汇总功能
debug ip rip [<i>events</i>]	打开 RIP 处理消息的调试功能
ip classless	使无类路由特性有效, 即路由器在不予考虑到达目的地址的类别的情况下, 能够找到最佳的匹配路由去转发数据包到目的地址
ip rip authentication key-chain <i>name-of-chain</i>	使接口上的 RIPv2 认证有效, 并指定一个所用的钥匙链的名字
ip rip authentication mode { <i>text</i> <i>md5</i> }	指定在一个接口上使用的是明文还是 MD5 认证
ip rip receive version [<i>1</i>] [<i>2</i>]	指定一个接口可以接收的 RIP 的版本
ip rip send version [<i>1</i>] [<i>2</i>]	指定一个接口可以发送的 RIP 的版本
ip split-horizon	在接口上打开或关闭水平分隔特性
ip subnet-zero	允许接口的地址和路由选择更新使用全 0 子网
key <i>number</i>	指定在钥匙链上的一个钥匙
key chain <i>name-of-chain</i>	指定一组钥匙
key-string <i>text</i>	指定钥匙使用的认证字符串或口令
network <i>network-number</i>	指定覆盖一个和多个运行 IGRP、EIGRP 或 RIP 协议进程的接口的网络地址
passive-interface <i>type</i> <i>number</i>	使一个接口不再发送路由选择更新
router rip	在路由器上启动 RIP 路由选择进程

续表

命 令	描 述
<code>send-lifetime start-time {infinite end-time duration seconds}</code>	设置一个时间段，用来指定钥匙链上的认证钥匙可被发送的有效时间
<code>show ip route [address [mask]][protocol [process-ID]]</code>	显示当前路由选择表的全部或某条路由
<code>Version</code>	指定 RIP 路由选择进程的版本号

7.6 推荐读物

Malkin, G. S. "RIP Version 2: Carrying Additional Information." RFC 1723, November 1994

7.7 复 习 题

1. RIPv2 的消息格式中包含了哪 3 个新的字段？
2. 除了上述的问题 1 中 3 个字段定义的扩展特性外，RIPv2 相比 RIPv1 还有哪两个主要的改变？
3. RIPv2 协议使用的组播地址是什么？组播方式通告消息报文与广播方式相比有什么好处？
4. 在 RIPv2 中的消息报文中，路由标记字段的目的是什么？
5. 在 RIPv2 中的消息报文中，下一跳字段的目的是什么？
6. RIPv2 协议使用的 UDP 端口号是什么？
7. 哪一个特性要求一个路由选择协议必须是一个无类别的路由选择协议？
8. 哪一个特性要求一个路由选择协议不得使用 VLSM？
9. 在 Cisco 的 RIPv2 中，有哪两种类型的认证可以使用？它们都在 RFC 1723 中定义了吗？

7.8 配置练习

1. 在图 7-10 的例子里，路由器 Taos 配置成可以发送版本 1 和版本 2 的更新报文，因此 Linux 主机 Pojoaque 上的“routed”进程可以理解来自路由器 Taos 的更新。除了使用 `ip rip send version` 命令还有另外的方法配置路由器 Taos 吗？
2. 一个互互联网分配到地址 192.168.100.0，划分这个地址来满足下面的需求：
 - 含有 50 台主机的 1 个子网
 - 含有 10 台主机的 5 个子网
 - 含有 25 台主机的 1 个子网
 - 含有 5 台主机的 4 个子网
 - 10 条串行链路

3. 配置图 7-29 中的 4 台路由器运行 RIP 协议。路由器 RTC 运行的是 IOS 10.3, 并由于策略原因不能升级。
4. 配置图 7-29 的路由器 RTB 和 RTD, 使其在串行链路上对交换的 RIP 更新进行认证。
5. 配置图 7-29 的路由器 RTB 和 RTD, 使其在配置 4 的认证钥匙生效后的 3 天改用一个新的认证钥匙, 这个新钥匙生效 10 小时后再改用另一个的钥匙。

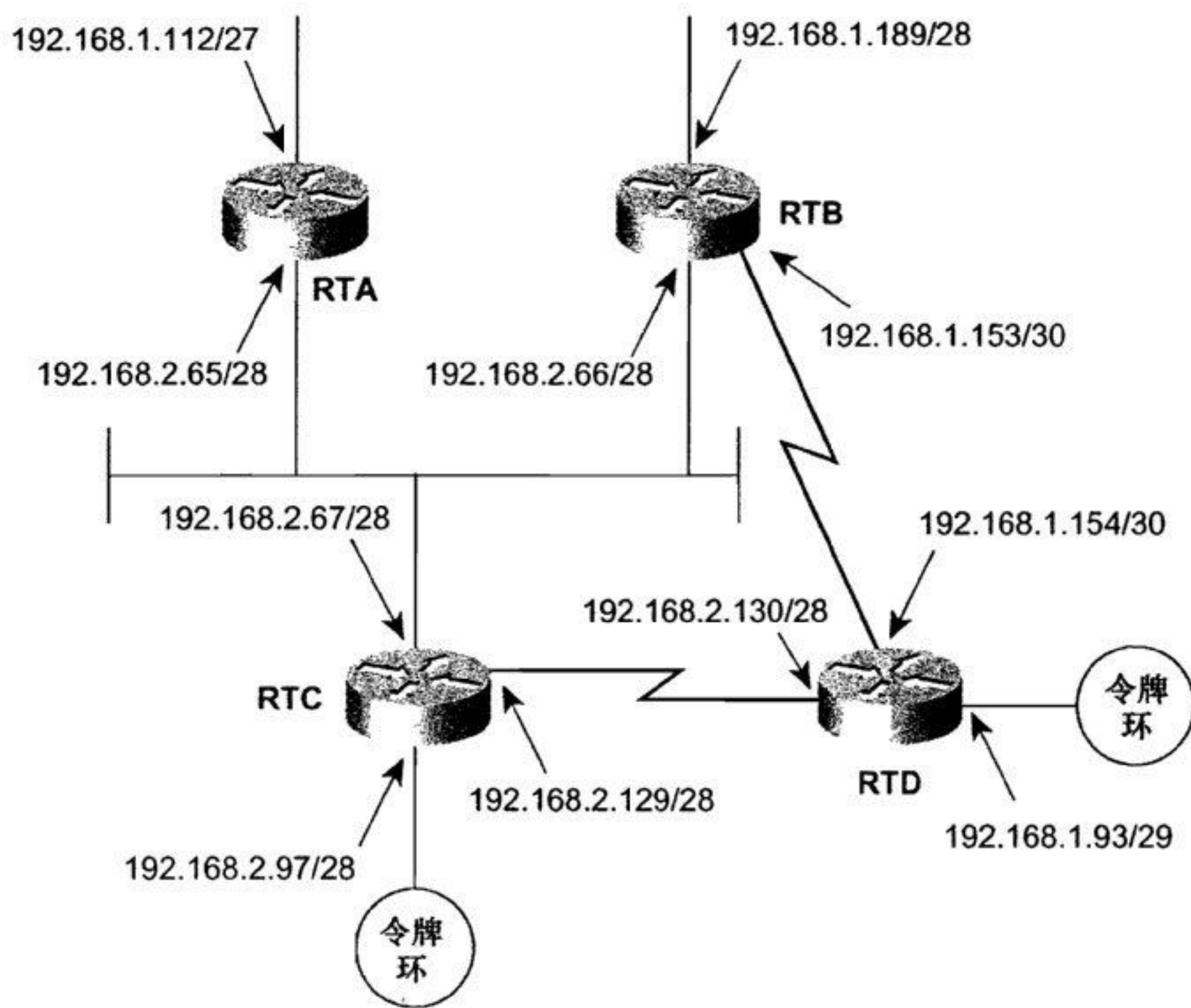


图 7-29 配置练习 3-5 的互联网络

7.9 故障排除练习

1. 图 7-31~图 7-33 显示了图 7-30 中的 3 台路由器的配置。哪些子网在每一个路由器的路由选择表中是都出现的？在每一台路由器上，哪些子网是可达的？哪些子网（如果有的话）是不可达的？

2. 图 7-30 中的路由器 RTA 和 RTB 的配置改变如下：¹

```
interface TokenRing0
ip address 192.168.13.35 255.255.255.224
ip rip receive version 1 2
ring-speed 16
```

这个改变配置的结果会有一些子网增加到路由选择表中吗？解释一下为什么有或为什么没有？

¹ 显示的是路由器 RTB 的配置，除了 IP 地址 192.168.13.34 外，路由器 RTA 的配置和路由器 B 是相同的。

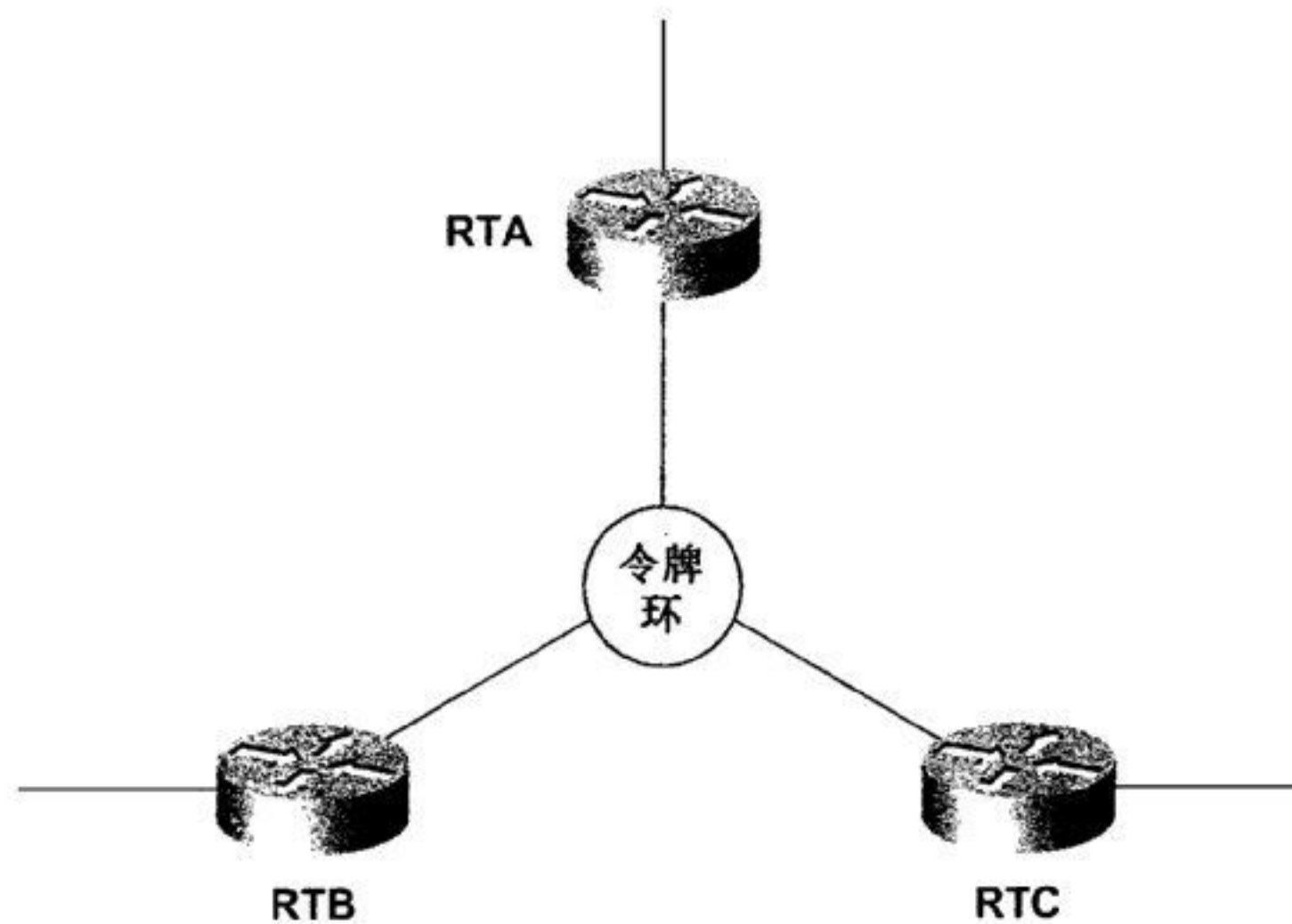


图 7-30 故障排除练习 1~2 的互连网络

```

RTA#show running-config
Building configuration...

Current configuration:
!
version 11.2
no service udp-small-servers
no service tcp-small-servers
!
hostname RTA
!
!
!
interface Ethernet0
 ip address 192.168.13.86 255.255.255.248
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface TokenRing0
 ip address 192.168.13.34 255.255.255.224
 ring-speed 16
!
router rip
 version 2
 network 192.168.13.0
!
no ip classless
!
!
line con 0
line aux 0
line vty 0 4

```

待续


```
login
!  
end  
RTA#
```

图 7-31 图 7-30 中路由器 RTA 的配置

```
RTB#show running-config  
Building configuration...  
  
Current configuration:  
!  
version 11.2  
no service udp-small-servers  
no service tcp-small-servers  
!  
hostname RTB  
!  
!  
!  
interface Ethernet0  
 ip address 192.168.13.90 255.255.255.240  
!  
interface Serial0  
 no ip address  
 shutdown  
!  
interface Serial1  
 no ip address  
 shutdown  
!  
interface TokenRing0  
 ip address 192.168.13.35 255.255.255.224  
 ring-speed 16  
!  
router rip  
 version 2  
 network 192.168.13.0  
!  
no ip classless  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
 login  
!  
end  
  
RTB#
```

图 7-32 图 7-30 中路由器 RTB 的配置


```
RTC#show running-config
Building configuration...

Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname RTC
!
!
!
interface Ethernet0
 ip address 192.168.13.75 255.255.255.224
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface TokenRing0
 ip address 192.168.13.33 255.255.255.224
 ring-speed 16
!
router rip
 network 192.168.13.0
!
no ip classless
!
line con 0
line 1 8
line aux 0
line vty 0 4
 login
!
end
RTC#
```

图 7-33 图 7-30 中路由器 RTC 的配置

第 8 章

增强型内部网关路由 选择协议（EIGRP）

本章包括以下主题：

- EIGRP 的操作
依赖于协议的模块
可靠传输协议
邻居的发现和恢复
扩散更新算法
EIGRP 的报文格式
地址聚合
- EIGRP 的配置
案例研究：一个基本的 EIGRP 配置
案例研究：和 IGRP 的重分配
案例研究：关闭自动路由汇总
案例研究：地址聚合
案例研究：认证
- EIGRP 的认证
- EIGRP 的故障排除
案例研究：邻居丢失
“卡”在活动状态的邻居

EIGRP 协议是在 Cisco IOS 9.21 版中首次发布的，顾名思义，它是 IGRP 协议的增强版。这个命名是比较恰当的，因为它不像 RIPv2 协议那样，EIGRP 协议对 IGRP 协议所增加的扩展特性远远多于 RIPv2 对 RIPv1 的扩展。EIGRP 协议依然是一个距离矢量型协议，并且使用了 IGRP 协议所用的复合度量。除此之外，EIGRP 协议和 IGRP 协议几乎没有更多的相似之处。

EIGRP 协议有时也被描述成一个具有链路状态协议行为特性的距离矢量协议。在这里, 对第 4 章中的广泛论述作一个扼要的重述: 距离矢量协议是路由器之间共享路由器所知道的所有信息, 但仅仅限于在与之直连的邻居之间共享; 而链路状态协议虽然只通告它们直连链路的信息, 但是链路状态协议可以在它们的路由选择域或区域内的所有路由器上共享这些信息。

到目前为止, 所讨论的所有距离矢量协议的运行都是基于 Bellman-Ford (或 Ford-Fulkerson) 算法或其一些派生算法的基础之上的。因此, 这些协议易于产生路由选择环路和计数无穷大的问题。结果, 这些协议必须要采取一些避免路由选择环路的措施, 像水平分隔、路由毒性逆转和抑制计时器等。由于每一台路由器在向它的邻居传送路由信息之前, 都必须对收到的路由信息运行路由选择算法, 因此大型互联网络的收敛可能会变得比较慢。更为重要的是, 距离矢量协议在通告路由时, 如果网络核心的关键链路发生了变化, 就意味着有很多发生变化的路由要进行通告。

相对于距离矢量协议, 链路状态协议受到路由选择环路和有害路由选择信息的影响就小得多了。首先, 链路状态报文的转发不依赖于路由计算的执行, 因而大型互联网络就可以更快速地收敛。其次, 它只通告链路和链路的状态, 而不是通告路由, 这意味着链路的变化不会引起使用这条链路的所有路由都被通告。然而, 相对于距离矢量的算法, 复杂的 Dijkstra 算法和相关的数据库会耗费更多的 CPU 和内存。

其他的路由选择协议执行路由的计算——不论是在给它的邻居发送距离矢量更新之前, 还是在生成网络拓扑的数据库之后, 它们的共同特性都是单独地进行路由的计算。相反, EIGRP 协议使用了一个称为扩散计算 (diffusing computations) 的方法——在多台路由器之间通过一个并行的方式执行路由的计算, 从而在保持无环路的拓扑时可以随时获取较快的收敛。

虽然 EIGRP 更新仍然是把距离矢量传送给它直连的邻居, 但是 EIGRP 更新是非周期的、部分的和有边界的。“非周期的”意思是指更新不是按照规则的时间间隔发送的, 而是在度量或网络拓扑发生变化时才发送更新。“部分的”意思是指更新只包含发生变化的路由条目, 而不是路由器的所有路由条目。“有边界的”意思是指更新仅仅发送给受到影响的路由器。这些特性意味着, EIGRP 协议比典型的距离矢量协议所使用的带宽少得多, 这个特点对路由选择协议在带宽较低而费用较高的广域网链路上运行时显得特别重要。

另外一个需要关注的是, 如果是在低带宽的广域网链路上进行路由选择时, 在路由收敛期间, 路由选择信息的流量会显得比较大, 这时要考虑可以使用的最大带宽。缺省情况下, EIGRP 协议使用的带宽不超过链路总带宽的 50%。后来 IOS 发布的版本允许使用命令 `ip bandwidth-percent eigrp` 来改变这个缺省的百分比。

EIGRP 协议是一个无类别的协议 (也就是说, 在它的路由更新里的每一个路由条目都包含子网掩码)。EIGRP 协议不仅可以利用可变长子网掩码进行子网的划分 (正如第 7 章所讲述的), 而且可以利用可变长子网掩码进行地址的聚合 (Address Aggregation), 即一组主网络地址的汇总。

从 IOS 11.3 版本开始, EIGRP 协议就能够使用 MD5 加密校验和对 EIGRP 报文进行认证。基本的认证和 MD5 认证已经在第 7 章中讲述过了, 本章仅讲述一个配置 EIGRP 认证的实例。

最后, EIGRP 协议的一个主要特性是它不仅可以进行 IP 协议的路由选择, 而且可以进行 IPX 协议和 AppleTalk 协议的路由选择。

8.1 EIGRP 的操作¹

EIGRP 协议使用和 IGRP 协议相同的公式来计算它的复合度量值。但是, EIGRP 协议使用一个 256 的倍数因子扩展了度量参数, 使它具有了更好的度量颗粒度。因此, 如果在一条到达目的地的路径上, 配置的最小带宽是 512K, 并且配置的总延迟是 46000 μ s, 那么 IGRP 协议计算出的复合度量值应该是 24131。而 EIGRP 协议将使用 256 倍数乘以带宽和延迟参数, 从而计算出的度量值是 $256 \times 24131 = 6177536$ 。要查看关于 IGRP 协议复合度量的计算的详细论述, 请参见第 6 章“内部网关路由选择协议 (IGRP)”。

如图 8-1, EIGRP 协议包含以下 4 个部件:

- 依赖于协议的模块
- 可靠传输协议 (RTP)
- 邻居发现和恢复模块
- 扩散更新算法 (DUAL)

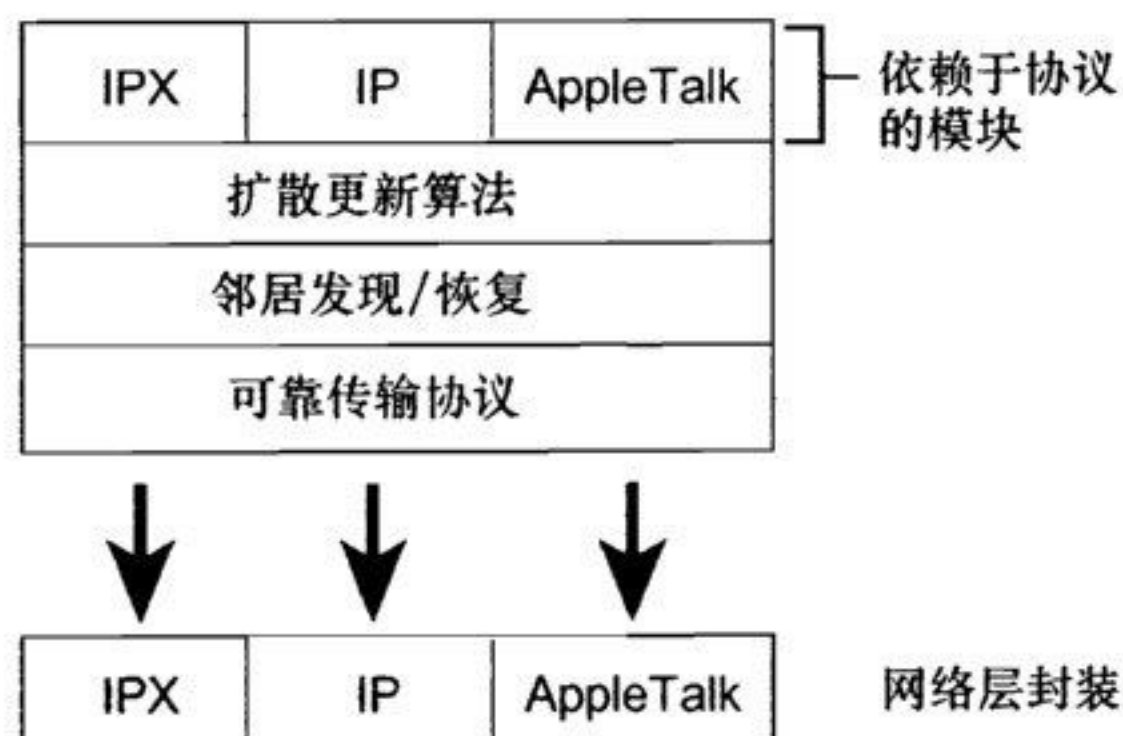


图 8-1 EIGRP 协议的 4 个主要部件。RTP 和邻居发现是使 DUAL 正确操作的更低层次上的协议。

DUAL 可以在多个可路由的协议上执行路由计算

本节将讲述 EIGRP 的每一个部件, 并特别注重地讲解 DUAL, 最后再讨论地址聚合的内容。

8.1.1 依赖于协议的模块 (Protocol-Dependent Modules)

EIGRP 协议实现了 IP 协议、IPX 协议和 AppleTalk 协议的模块, 它可以担负起某一特定协议的路由选择任务。例如, IPX EIGRP 模块可以负责在 IPX 网络上与其他 IPX EIGRP 进程进行路由信息交换的任务, 并且把这些信息传给 DUAL。另外, IPX 模块也接收和发送 SAP 信息。

正如图 8-1 所示, 每个单独模块的通信量被封装在它们各自的网络层协议里面, 例如,

¹ EIGRP 协议几个主要的修订版本是 IOS 10.3(11)、11.0(8)和 11.1(3)。后来的几个版本由于性能和稳定性的提高, 比原来的老版本更加实用。

对于 IPX 协议的 EIGRP 通过 IPX 协议格式数据包传输。

EIGRP 协议在很多情况下和其他路由选择协议自动进行路由重新分配:

- IPX EIGRP 将自动地和 IPX RIP 协议、NLSP 协议进行路由重新分配;
- AppleTalk EIGRP 协议将自动地和 AppleTalk RTMP 协议进行路由重新分配;
- 如果 IGRP 进程和 EIGRP 进程在同一个自主系统内, 那么 IP EIGRP 协议也将自动地和 IGRP 协议进行路由重新分配。

在配置一节中包含了一个 IGRP 协议和 EIGRP 协议之间进行路由重新分配的例子。和其他 IP 路由选择协议之间的路由重新分配将是第 11 章“路由重新分配”的主题。

配置关于 IPX 和 AppleTalk 的 EIGRP 协议超出了本书的讲解范围, 请参阅 Cisco 配置手册以获取更多的信息。

8.1.2 可靠传输协议 (RTP)

可靠传输协议 (Reliable Transport Protocol) 用来管理 EIGRP 报文的发送和接收。可靠的发送是指发送是有保障的而且报文是有序的发送的。有保障的发送是依赖 Cisco 公司私有的算法来实现的, 这个私有的算法被称为“可靠组播 (reliable multicast)”, 它使用保留的 D 类地址 224.0.0.10。每一个接收可靠组播报文的邻居都会发送一个单播的确认报文。

有序的发送是通过在每个报文中包含两个序列号来实现的。每一个报文都包含一个由发送该报文的路由器分配的序列号, 这个序列号在每台路由器发送一个新的报文时递增 1。另外, 发送路由器会把最近从目的路由器收到的报文的序列号放在该报文中。

在一些实例中, RTP 也可以使用不可靠的发送, 不需要确认, 而且在使用不可靠发送的报文中不包含序列号。

EIGRP 协议使用多种类型的报文, 所有这些报文都通过 IP 报文头部的协议号 88 来标识。

- **Hello 报文 (Hello)**——用于邻居的发现和恢复的过程。Hello 报文使用组播方式发送, 而且使用不可靠的发送方式。
- **确认报文 (Acknowledgments, ACKs)**——是不包含数据的 Hello 报文。ACKs 报文总是使用单播方式和不可靠的发送方式。
- **更新报文 (Update)**——用于传递路由更新信息。不像 RIP 协议和 IGRP 协议的更新报文, EIGRP 协议的这些更新报文只在必要的时候传递必要的信息, 而且仅仅传递给需要路由信息的路由器。当只有某一指定的路由器需要路由更新时, 更新报文就是单播发送的; 当有多台路由器需要路由更新时, 更新报文就是组播发送的, 例如, 路由的度量和拓扑发生变化时。更新报文总是使用可靠的发送方式。
- **查询 (Query) 和答复 (Reply) 报文**——是 DUAL 有限状态机 (DUAL finite state machine) 用来管理它的扩散计算的。查询报文可以使用组播方式或者单播方式发送, 而回复报文总是单播方式发送的。查询和回复报文都使用可靠的发送方式。
- **请求报文 (Request)**——最初是打算提供给路由服务器使用的报文类型。但是这个应用从来没有实现过, 在这里提到请求报文主要是因为有一些老的文档中可能会提及它们。

如果任何报文通过可靠的方式组播出去, 而没有从邻居那里收到一个 ACK 报文, 那么这个报文就会以单播方式被重新发送给那个没有响应的邻居。如果经过 16 次这样的单播重传

还没有收到一个 ACK 报文的话, 那么这个邻居就会被宣告为无效。

在从组播方式切换到单播方式之前等待一个 ACK 报文的时间可以由组播流计时器 (multicast flow timer) 指定。后续的单播之间的时间可以由重传超时 (retransmission timeout, RTO) 指定。对于每一个邻居, 组播流计时器和重传超时都可以通过平均回程时间 (smooth round-trip time, SRTT) 来计算。SRTT 是一个用来衡量路由器发送 EIGRP 报文到邻居和从邻居那里接收到该报文的确认报文为止所花费的平均时间, 以毫秒 (ms) 为单位。关于 SRTT、RTO 和组播流计时器的精确值的计算公式是有私有版权的。

下面两个后续的小章节将讲述使用不同报文类型的 EIGRP 的部件。

8.1.3 邻居的发现和恢复

因为 EIGRP 协议的更新报文是非周期的, 因此有一个发现和跟踪邻居的方法是非常重要的, 在这里, 邻居是指网络上直连的通告 EIGRP 的路由器。在大多数的网络中, Hello 报文是以组播方式每 5s 发送一次的, 其中减掉一个很小的随机时间差用来防止更新的同步。在多点的 X.25、帧中继和 ATM 接口上, 由于它们的接入链路速率通常是 T1 或更低的速率, 因此它们的 Hello 报文是以单播方式每 60s 发送一次的。¹这个比较长的 Hello 报文时间间隔也缺省地使用在 ATM SVC 和 ISDN PRI 的接口上。在所有的实例中, Hello 报文都是不进行确认的。缺省的 Hello 报文的时间间隔可以在每个接口上使用命令 **ip hello-interval eigrp** 进行更改。

当一台路由器从它的邻居路由器收到一个 Hello 报文时, 这个报文将包含一个抑制时间 (holdtime)。这个抑制时间会告诉本路由器, 在它收到后续的 Hello 报文之前等待的最长时间。如果抑制计时器超时了, 路由器还没有收到 Hello 报文, 那么将宣告这个邻居不可到达, 并且通知 DUAL 这个邻居丢失了。在缺省的情况下, 抑制时间是 Hello 报文时间间隔的 3 倍长, 也就是说, 对于低速的非广播多路访问 (NBMA) 网络来说是 180s, 对于其他所有的网络来说是 15s。这个缺省值可以通告在每个接口上配置命令 **ip hold-time eigrp** 来更改。EIGRP 协议具有在 15s 以内检测邻居丢失的能力, 相对照 RIP 协议的 180s 和 IGRP 协议的 270s 所花费的时间, 显然这是一个对 EIGRP 的快速收敛起很大作用的因素。

每一个邻居的相关信息都记录在一个邻居表中。如图 8-2 显示, 邻居表 (neighbor table) 记录了邻居路由器的 IP 地址和收到邻居的 Hello 报文的接口。邻居通告的抑制时间也作为 SRTT 和邻居关系建立时间 (uptime) 也记录在邻居表中, 这里的邻居关系建立时间是指从邻居第一次被添加到邻居表后到现在所经过的时间。重传超时 RTO 是指在一个组播方式的报文发送失败后, 路由器等待一个单播方式发送的报文的确认报文的时间, 单位是毫秒 (ms)。如果一个 EIGRP 的更新、查询或答复报文被发送出去, 那么这个报文的一个拷贝就会放在一个重传队列里排队。如果 RTO 超时了还没有收到确认报文, 那么重传队列中报文的另一个拷贝将被再次发送出去。队列计数 (Q Count) 就是标识在这个重传队列中等待发送的报文数量的。从邻居收到的最新的更新、查询或答复报文的序列号也记录在了邻居表中。可靠传输协议 RTP 就跟踪这些序列号, 以确保来自于邻居的报文不是无序收到的。最后, H 列记录了这台路由器所学到的邻居的顺序号。

¹ 点到点的子接口每 5s 发送一次 Hello 报文。


```

Wright#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address   Interface   Hold Uptime      SRTT   RTT    Q      Seq
                               (sec)          (ms)          Cnt    Num
3   10.1.1.2   Et0         10 09:01:27      12     200    0      5
2   10.1.4.2   Se1         13 09:02:11      23     200    0     11
1   10.1.2.2   Et1         14 09:02:12       8     200    0     15
0   10.1.3.2   Se0         12 09:02:12      21     200    0     13
Wright#

```

图 8-2 show ip eigrp neighbors 命令用来观察 IP EIGRP 的邻居表

8.1.4 扩散更新算法 (Diffusing Update Algorithm)

DUAL 算法背后的设计思想是,即使暂时的路由选择环路也会对一个互联网络的性能造成损害。DUAL 最初是由 E. W. Dijkstra 和 C. S. Scholten 提议的¹,指的是为了随时能够打破路由环路,而使用扩散计算去执行一个分布式的最短路径的路由选择。虽然很多研究人员对 DUAL 算法的发展作出了贡献,但是最显著的贡献来自于 J. J. Garcia-Luna-Aceves 的工作。²

1. DUAL: 预备概念

为了能够正确地操作 DUAL,较低层的协议必须确保满足下面的几个条件:³

- 一个节点需要在有限的时间内检测到一个新的邻居的存在或一个相连邻居的丢失;
- 在一个正在运行的链路上传送的所有消息应该在一个有限的时间内正确地收到,并且包含正确的序列号;
- 所有的消息,包括改变链路的代价、链路失败和发现新邻居的通告,都应该在一个有限的时间内一次一个地处理,并且应该被有序地检测到。

Cisco 的 EIGRP 协议使用邻居发现/恢复和可靠传输协议 RTP 来确定这些前提条件。在介绍 DUAL 之前,先来介绍几个术语和概念。

(1) 邻接 (Adjacency)

刚启动时,路由器使用 Hello 报文发现它的邻居和标识自己给邻居识别。当邻居被发现时,EIGRP 协议将试图和它的邻居形成一个邻接。邻接是指两个互相交换路由信息的邻居之间形成的一条虚链路。一旦邻接成功地建立,路由器就可以从它们的邻居接收路由更新信息了。这里的路由更新信息包括发送路由器所知道的所有路由和这些路由的度量值。对于每一条路由,路由器都将会基于它的邻居通告的距离 (distance) 和到它的邻居的链路代价计算出一个距离。

(2) 可行距离 (Feasible distance)

到达每一个目的地的最小度量将作为那个目的网络的可行距离 (Feasible Distance)。例如,路由器可能得到 3 条不同的路由可以到达子网 172.16.5.0,这 3 条路由计算所得的度量分别是 380672、12381440 和 660868。那么 380672 就成为可行距离 FD,因为它是经计算

¹ Edsger W. Dijkstra 和 C. S. Scholten 编写的 "Termination Detection for Diffusing Computations." Information Processing Letters, Vol. 11, No. 1, pp. 1-4: 29 August 1980.

² J. J. Garcia-Luna-Aceves. "A Unified Approach for Loop-Free Routing Using Link States or Distance Vectors," ACM SIGCOMM Computer Communications Review, Vol. 19, No. 4, pp. 212-223: September 1989.

³ J.J. Garcia-Luna-Aceves. "Area-Based, Loop-Free Internet Routing." Proceedings of IEEE INFOCOMM 94. Toronto, Ontario, Canada, June 1994.

到达子网 172.16.5.0 的最小度量。

(3) 可行性条件 (Feasibility condition)

可行性条件 (Feasibility Condition, FC) 就是需要满足下面这样的条件——本地路由器的一个邻居路由器所通告的到达一个目的网络的距离是否小于本地路由器到达相同目的网络的可行距离 FD。

(4) 可行后继路由器 (Feasible successor)

如果本地路由器的邻居路由器所通告的到达目的网络的距离满足了可行性条件 FC, 那么这个邻居就会成为那个目的网络的一个可行后继路由器。¹例如, 假定一个路由器到达子网 172.16.5.0 的可行距离是 380672, 而它的邻居路由器所通告的到达该目的子网的路由路径的距离是 355072, 那么这个邻居路由器就满足可行性条件 FC, 因而成为一个可行后继路由器; 如果邻居路由器所通告的距离是 380928, 那么这个邻居路由器就不满足可行性条件 FC, 因而也就不能成为一个可行后继路由器。

可行后继路由器和可行性条件 FC 的概念是避免环路的一个核心技术, 因为可行后继路由器总是“下游路由器 (downstream)” (也就是说, 可行后继路由器到达目的地的度量距离比本地路由器的可行距离 FD 更短), 所以路由器从来不会选择一条导致反过来还要经过它本身的路由路径——像这样的路径一般有一个大于本地路由器可行距离 FD 的距离。

存在一个或多个可行后继路由器的每一个目的网络将连同下面的每项一起被记录在一个称为“拓扑结构表 (topological table)”的表中:

- 目的网络的可行距离;
- 所有的可行后继路由器;
- 每一个可行后继路由器所通告的到达目的网络的通告距离;
- 本地路由器所计算的经过每一个可行后继路由器到达目的网络的距离, 也就是基于可行后继路由器所通告的到达目的子网的距离和本地路由器与该可行后继路由器之间相连链路的代价计算所得的距离;
- 与发现每一个可行后继路由器的网络相连的接口。²

(5) 后继路由器 (Successor)

对于在拓扑结构表中列出的每一个目的网络, 将选用拥有最小度量值的路由并放置到路由选择表中。通告这条路由的邻居就成为一个后继路由器 (Successor), 或者是到达目的网络的数据包的下一跳路由器。

举一个例子可以帮助我们澄清这些术语, 但先来简要地描述一下本节这个例子中用到的互连网络还是必要的。图 8-3 显示了一个基于 EIGRP 的互连网络, 这个网络将使用在本节和后续的 3 个小章节中。³把命令 **metric weights 0 0 0 1 0 0** 添加到 EIGRP 的进程处理当中, 因此只使用延迟来进行路由的度量计算。命令 **delay** 使用图中每一个链路上标注的数值。例如, 路由器 Wright 和路由器 Langley 的接口和子网 10.1.3.0 相连, 那么接口配置的延迟就是 2。这些做法将给下面的例子带来方便和简化。

¹ 后继路由器简单的理解就是指到达目的网络更近一跳的路由器, 换句话说, 就是下一跳路由器。

² 事实上, 这个接口并不是明确地显示在路由选择表中的。更确切地说, 它是邻居路由器自身的属性。这个约定意味着通过多条并行链路的相同路由器将被 EIGRP 协议看作是多个邻居。

³ 在本节和后续几个章节演示的几个图例以及使用的网络示例改编自 Garcia-Luna 先生的“使用扩散计算的无环路路由选择”, 并得到了他的许可。

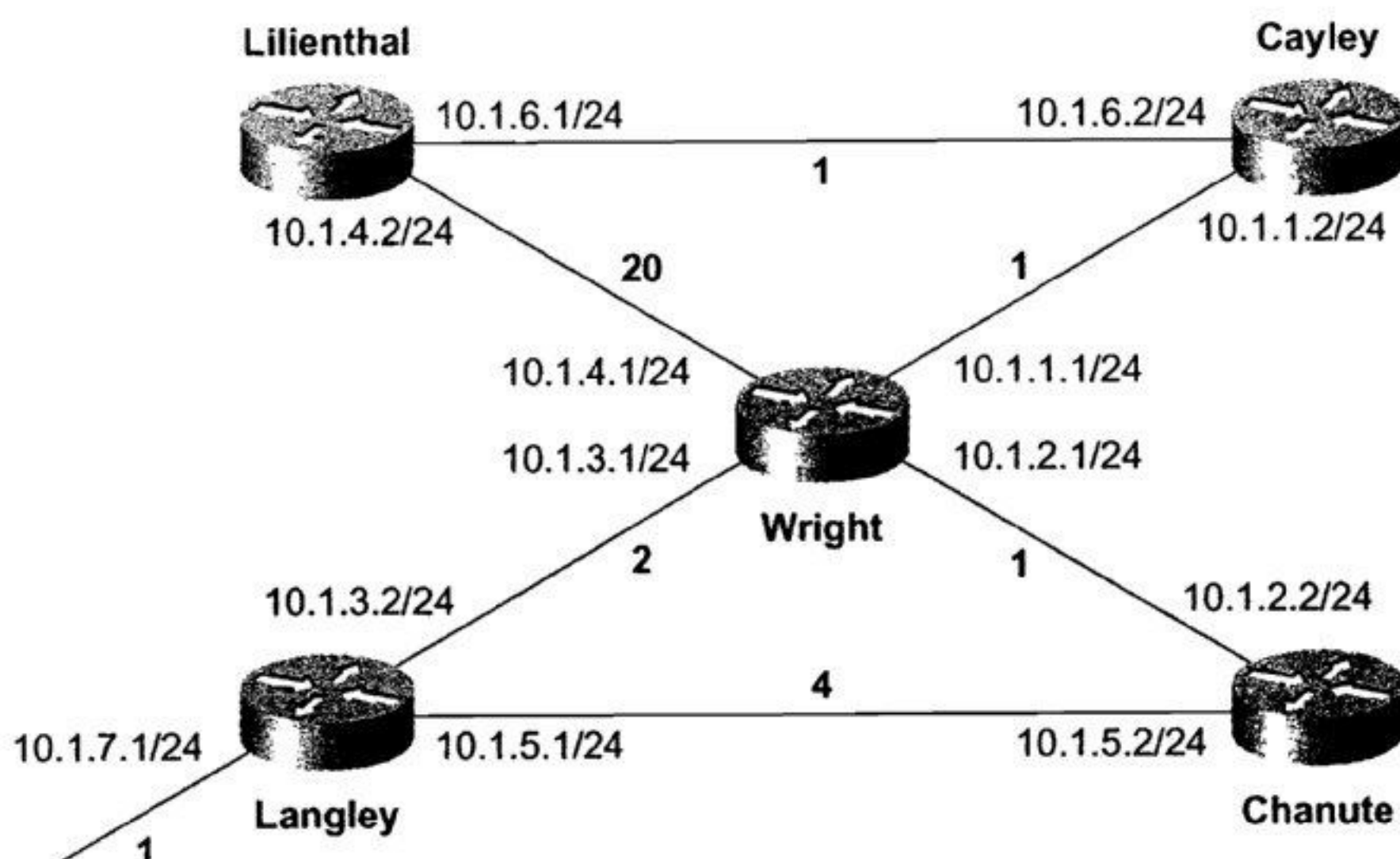


图 8-3 本节和后续的两个章节的图例和例子都是基于 EIGRP 协议的网络

需要指出的是, 虽然这里使用的延迟参数为了简化而不太实用, 但是进行度量的方法还是很实用的。很多参数是通过接口的命令 **bandwidth** 所指定的带宽进行计算的。有的, 像命令 **ip bandwidth-percent eigrp** 就是直接地应用于 EIGRP 的; 还有的, 像 OSPF 的代价并不是直接地应用于 OSPF 的。因此, 除了需要设置串行链路的带宽和它们的实际带宽相一致外, 应该避免改变带宽的配置。如果需要改变一个接口的度量来影响 EIGRP (或 IGRP 协议) 的路由选择, 应该使用命令 **delay**。这样可以避免很多令人头痛的问题。

在图 8-4 中, 可以使用命令 **show ip eigrp topology** 查看路由器 Langley 的拓扑结构表。图 8-3 中显示的 7 个子网的每个子网和这些子网的可行后继路由器一起, 都在拓扑结构表中列出了。例如, 子网 10.1.6.0 的可行后继路由器是 10.1.3.1 (路由器 Wright) 和 10.1.5.2 (路由器 Chanute), 并分别通过接口 S0 和 S1 到达。

```
Langley#show ip eigrp topology
IP-EIGRP Topology Table for process 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.3.0/24, 1 successors, FD is 512
   via Connected, Serial0
P 10.1.2.0/24, 1 successors, FD is 768
   via 10.1.3.1 (768/256), Serial0
   via 10.1.5.2 (1280/256), Serial1
P 10.1.1.0/24, 1 successors, FD is 768
   via 10.1.3.1 (768/256), Serial0
   via 10.1.5.2 (1536/512), Serial1
P 10.1.7.0/24, 1 successors, FD is 256
   via Connected, Ethernet0
P 10.1.6.0/24, 1 successors, FD is 1024
   via 10.1.3.1 (1024/512), Serial0
   via 10.1.5.2 (1792/768), Serial1
P 10.1.5.0/24, 1 successors, FD is 1024
   via Connected, Serial1
P 10.1.4.0/24, 1 successors, FD is 5632
   via 10.1.3.1 (5632/5120), Serial0
   via 10.1.5.2 (6400/5376), Serial1
Langley#
```

图 8-4 路由器 Langley 的路由拓扑结构表

圆括号中的两个度量也都是和每个可行后继路由器相关联的。第一个数字是本地路由器计算得出的路由器 Langley 到达目的网络的度量值。第二个数字是邻居通告的度量值。例如, 图 8-3 中路由器 Langley 经过路由器 Wright 到达子网 10.1.6.0 的度量值是 $256 \times (2+1+1) = 1024$, 而邻居路由器 Wright 通告的到达这个目的子网的度量值是 $256 \times (1+1) = 512$ 。通过路由器 Chanute 到达上面相同的子网的这两个度量值分别是 $256 \times (4+1+1+1) = 1792$ 和 $256 \times (1+1+1) = 768$ 。

可见, 从路由器 Langley 到达子网 10.1.6.0 的最小度量值是 1024, 因此, 这个度量值就是可行距离 FD。图 8-5 中显示了路由器 Langley 的路由选择表, 可以从中看出所选用的后继路由器。

```
Langley#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
       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

Gateway of last resort is not set

10.0.0.0/8 is subnetted, 7 subnets
C      10.1.3.0 is directly connected, Serial0
D      10.1.2.0 [90/768] via 10.1.3.1, 00:32:06, Serial0
D      10.1.1.0 [90/768] via 10.1.3.1, 00:32:07, Serial0
C      10.1.7.0 is directly connected, Ethernet0
D      10.1.6.0 [90/1024] via 10.1.3.1, 00:32:07, Serial0
C      10.1.5.0 is directly connected, Serial1
D      10.1.4.0 [90/5632] via 10.1.3.1, 00:32:07, Serial0
Langley#
```

图 8-5 基于最小的度量距离计算, 路由器 Langley 的路由选择表显示了每个可行的目的网络选用的单个后继路由器

对于路由器 Langley 的每一条路由, 只有一个后继路由器, 而图 8-6 中路由器 Cayley 的拓扑结构表却显示了到达目的网络 10.1.4.0 的两个后继路由器。这是因为在路由器 Cayley 所计算的度量值中, 有两条路由路径的度量值都匹配它的可行距离 FD。因此这两条路由都存在于图 8-7 中的路由选择表中, 并且路由器 Cayley 执行等价负载均衡。

```
Cayley#show ip eigrp topology
IP-EIGRP Topology Table for process 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.3.0/24, 1 successors, FD is 768
    via 10.1.1.1 (768/512), Ethernet0
P 10.1.2.0/24, 1 successors, FD is 512
    via 10.1.1.1 (512/256), Ethernet0
P 10.1.1.0/24, 1 successors, FD is 256
    via Connected, Ethernet0
P 10.1.7.0/24, 1 successors, FD is 1024
    via 10.1.1.1 (1024/768), Ethernet0
P 10.1.6.0/24, 1 successors, FD is 256
    via Connected, Serial0
P 10.1.5.0/24, 1 successors, FD is 1536
    via 10.1.1.1 (1536/1280), Ethernet0
P 10.1.4.0/24, 2 successors, FD is 5376
    via 10.1.6.1 (5376/5120), Serial0
    via 10.1.1.1 (5376/5120), Ethernet0
Cayley#
```

图 8-6 路由器 Cayley 的路由拓扑结构表显示了到达目的子网 10.1.4.0 的两个后继路由器


```

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

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 7 subnets
D    10.1.3.0 [90/768] via 10.1.1.1, 00:01:19, Ethernet0
D    10.1.2.0 [90/512] via 10.1.1.1, 00:01:19, Ethernet0
C    10.1.1.0 is directly connected, Ethernet0
D    10.1.7.0 [90/1024] via 10.1.1.1, 00:01:19, Ethernet0
C    10.1.6.0 is directly connected, Serial0
D    10.1.5.0 [90/1536] via 10.1.1.1, 00:01:19, Ethernet0
D    10.1.4.0 [90/5376] via 10.1.1.1, 00:01:19, Ethernet0
           [90/5376] via 10.1.6.1, 00:01:19, Serial0
Cayley#

```

图 8-7 在到达目的子网 10.1.4.0 的两个后继路由器之间将执行等价负载均衡

图 8-8 中的路由器 Chanute 的拓扑结构表显示了几条仅仅拥有一个可行后继路由器的路由路径。例如, 到达子网 10.1.6.0 的路由路径拥有一个距离为 768 的可行距离 FD, 因而路由器 Wright (10.1.2.1) 是惟一的可行后继路由器。路由器 Langley 有一条到达子网 10.1.6.0 的路由路径, 但是它的度量值是 $256 \times (2+1+1) = 1024$, 大于可行距离 FD, 因此, 路由器 Langley 到达子网 10.1.6.0 的路由不满足可行性条件 FC, 因而路由器 Langley 也就没有资格成为一个可行后继路由器。

```

Chanute#show ip eigrp topology
IP-EIGRP Topology Table for process 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.3.0/24, 1 successors, FD is 768
    via 10.1.2.1 (768/512), Ethernet0
    via 10.1.5.1 (1536/512), Serial0
P 10.1.2.0/24, 1 successors, FD is 256
    via Connected, Ethernet0
P 10.1.1.0/24, 1 successors, FD is 512
    via 10.1.2.1 (512/256), Ethernet0
P 10.1.7.0/24, 1 successors, FD is 1024
    via 10.1.2.1 (1024/768), Ethernet0
    via 10.1.5.1 (1280/256), Serial0
P 10.1.6.0/24, 1 successors, FD is 768
    via 10.1.2.1 (768/512), Ethernet0
P 10.1.5.0/24, 1 successors, FD is 1024
    via Connected, Serial0
P 10.1.4.0/24, 1 successors, FD is 5376
    via 10.1.2.1 (5376/5120), Ethernet0
Chanute#

```

图 8-8 从路由器 Chanute 可达的几个子网都只有惟一的一个可行后继路由器

如果一个可行后继路由器通告的一条路由在本地路由器上所计算的度量比当前后继路由器的度量小, 那么这个可行后继路由器就成为后继路由器。下面的情况可能会引起这种情形的发生:

- 发现一条新的路由;
- 一条后继路由器的路由的度量值增加后超过了可行后继路由器的度量值;
- 一条可行后继路由器的路由的度量值减小后小于后继路由器的度量值。

例如, 图 8-9 中显示了路由器 Lilienthal 到达子网 10.1.3.0 的后继路由器是路由器 Cayley (10.1.6.2)。假设路由器 Lilienthal 和路由器 Wright 之间的链路代价减小到 1, 那么由于路由器 Wright (10.1.4.1) 正在通告一条到达子网 10.1.3.0 的距离是 512 的路由, 同时根据路由器 Lilienthal 和路由器 Wright 之间新的链路代价值, 路由器 Lilienthal 计算出经过路由器 Wright 到达目的子网的度量现在变成了 768。因此, 路由器 Wright 将替代路由器 Cayley 成为到达子网 10.1.3.0 的后继路由器。

```
Lilienthal#show ip eigrp topology
IP-EIGRP Topology Table for process 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.3.0/24, 1 successors, FD is 1024
    via 10.1.6.2 (1024/768), Serial0
    via 10.1.4.1 (5632/512), Serial1
P 10.1.2.0/24, 1 successors, FD is 768
    via 10.1.6.2 (768/512), Serial0
    via 10.1.4.1 (5376/256), Serial1
P 10.1.1.0/24, 1 successors, FD is 512
    via 10.1.6.2 (512/256), Serial0
    via 10.1.4.1 (5376/256), Serial1
P 10.1.7.0/24, 1 successors, FD is 1280
    via 10.1.6.2 (1280/1024), Serial0
    via 10.1.4.1 (5888/768), Serial1
P 10.1.6.0/24, 1 successors, FD is 256
    via Connected, Serial0
P 10.1.5.0/24, 1 successors, FD is 1792
    via 10.1.6.2 (1792/1536), Serial0
    via 10.1.4.1 (6400/1280), Serial1
P 10.1.4.0/24, 1 successors, FD is 5120
    via Connected, Serial1
Lilienthal#
```

图 8-9 路由器 Lilienthal 的拓扑结构表

其次, 我们假定路由器 Lilienthal 发现了一个新的邻居, 并且这个邻居正在通告一条到达子网 10.1.3.0 距离为 256 的路由。由于这个距离小于当前的可行距离 FD, 因而这个新的邻居就成为一个可行后继路由器。进一步假定路由器 Lilienthal 到达这个新邻居的链路代价是 256, 那么路由器 Lilienthal 将计算得出经过这个新邻居到达子网 10.1.3.0 的度量值为 512。这个度量值小于经过路由器 Wright 的度量值, 因此这个新的邻居路由器将成为到达子网 10.1.3.0 的后继路由器。

由于可行后继路由器减少了扩散计算的数量, 提高了网络的性能, 因此可行后继路由器十分重要。可行后继路由器也对降低重新收敛的次数有一定的贡献。如果到达后继路由器的一条链路失效了, 或者链路的代价增加并超过了可行距离 FD, 那么这台路由器将首先在它的拓扑结构表中查找可行后继路由器, 如果发现存在一台可行后继路由器, 它就成为后继路由器。路由器只有在找不到任何一台可行后继路由器的情况下, 才开始进行扩散计算。

下面的章节将给出一个更正式的规则集合, 来说明路由器是何时和怎么样查找可行后继路由器的。这个规则集合称为 DUAL 有限状态机。

2. DUAL 有限状态机

当一个 EIGRP 的路由器不执行扩散计算时, 每一条路由都处于被动状态 (passive state)。

参见前面章节出现的所有的拓扑结构表，每一条路由左边的关键字就是用来指出路由的被动状态的。

在产生输入事件 (input event) 的任何时候，路由器都会重新评估一条路由的可行后继路由器的列表，这将在本章最后一节中讲述。一个输入事件可以是：

- 直连链路的代价发生变化；
- 直连链路的状态 (up 或 down) 发生变化；
- 收到一个更新报文；
- 收到一个查询报文；
- 收到一个答复报文。

路由器重新评估的第一步是在本地路由器上执行一个本地计算 (local computation)，也就是对于所有的可行后继路由器，重新计算到达目的地的距离。可能的结果有下面几种：

- 如果拥有最低的度量距离的可行后继路由器和已经存在的后继路由器不同，那么可行后继路由器将成为后继路由器；
- 如果新的度量距离小于可行距离 FD，那么就更新可行距离 FD；
- 如果新的度量距离和已经存在的度量距离不同，那么将向所有的邻居发送更新。

当路由器执行一个本地计算时，路由依然保持被动状态。如果本地路由器发现了一台可行后继路由器，那么将发送一个更新报文给它所有的邻居，但不改变路由的状态。

如果在拓扑结构表中没有发现任何一台可行后继路由器的话，那么路由器将开始进行扩散计算，而且路由器的路由状态改变成活动状态 (active state)。在扩散计算完成和路由的状态返回到被动状态之前，路由器不能：

- 改变路由的后继路由器；
- 改变正在通告的路由的距离；
- 改变路由的可行距离 FD；
- 开始进行路由的另一个扩散计算。

如图 8-10 所示，路由器是通过向它所有的邻居发送查询报文来开始一个扩散计算的，查询报文中包含一个到达目的地的新的本地路由器计算的距离。收到查询报文后，每一台邻居路由器将执行它自己的本地计算：

- 如果该邻居拥有到达目的地的一台或多台可行后继路由器，它将发送一个答复报文给原来发送查询报文的路由器。答复报文将包含这台邻居路由器所计算的它到达目的网络的最小距离；
- 如果一个邻居没有可行后继路由器，它将把路由的状态改变为活动状态，并且开始进行扩散计算。

对于每一台接收查询报文的邻居路由器，本地路由器将设置一个答复状态标记 (reply status flag (r)) 来不断跟踪所有未处理的查询报文。当本地路由器收到所有发送到邻居路由器的查询报文的答复报文时，扩散计算就完成了。

在一些实例中，路由器没有收到发出的每一个查询报文的答复报文。例如，这有可能发生在一个拥有很多低速带宽或质量较差的链路的大型网络当中。在扩散计算的开始，一个活动计时器 (active timer) 被设置为 3min。¹如果在活动计时器计时超时后还没有收到希望收到

¹ 在一些早期的 IOS 软件版本中，缺省的活动计时器设置为 1min。

的所有答复, 那么这条路由就被宣告“卡”在活动状态 (stuck-in-active, SIA)。这些没有答复的邻居将从邻居表中删除, 并且扩散计算认为这个邻居回应了一个无穷大的度量。

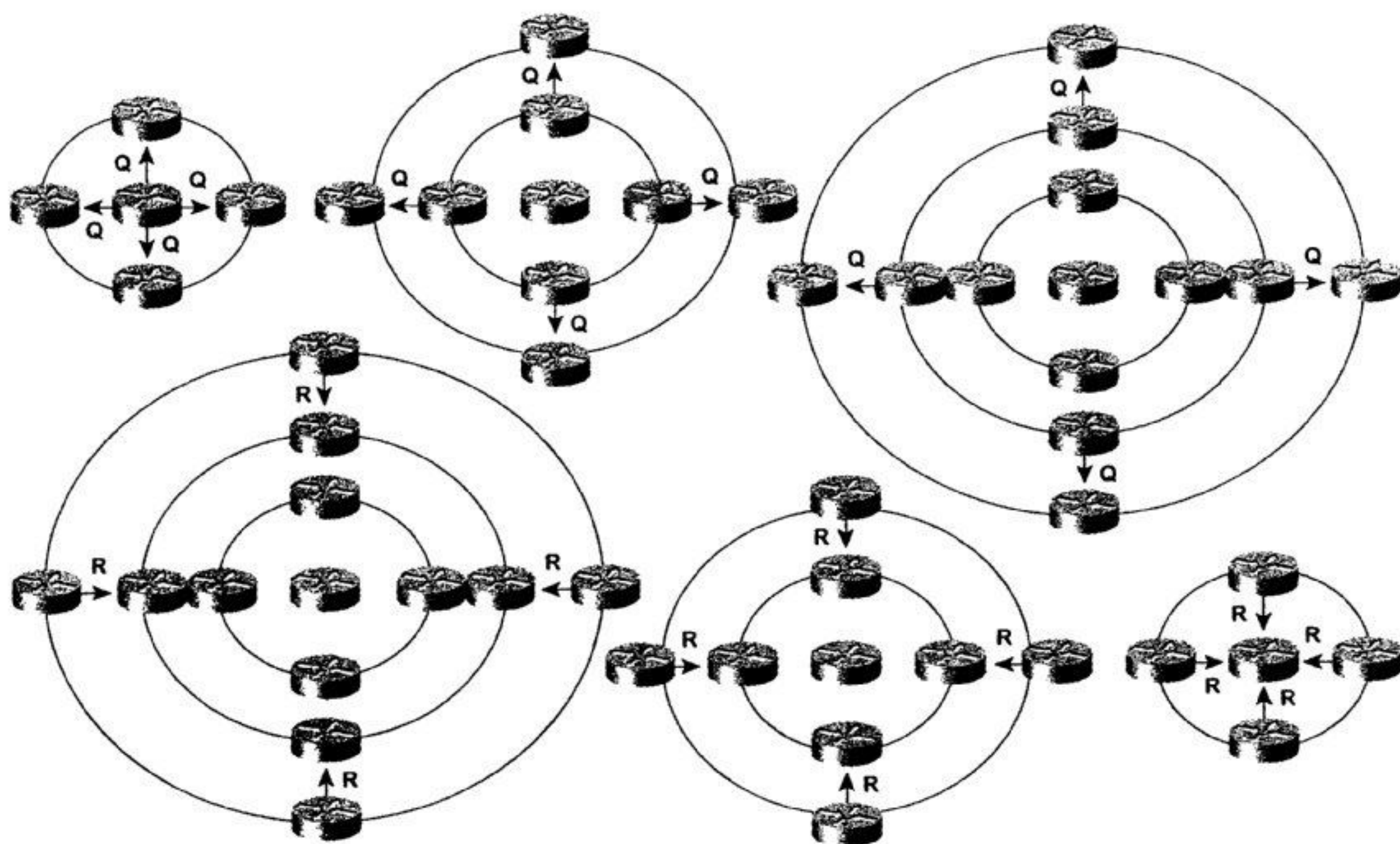


图 8-10 扩散计算在查询报文被发送时扩大, 而在答复报文被收到时收缩

活动计时器的缺省配置是 3min, 这个时长可以通过命令 **timers active-time** 来改变或使其无效。由于查询报文的丢失而造成邻居的删除显然会带来负面的影响, 而在一个稳定的、设计良好的互连网络中, SIA 的情形是从来不会发生的。本章故障排除一节中将更详细地讨论 SIA。

在扩散计算完成的时候, 最初始发的路由器会把可行距离 **FD** 设置成无穷大, 这样可以确保任何答复到达目的地是有限距离的邻居路由器都满足可行性条件 **FC**, 因而成为一台可行后继路由器。对于这些答复报文, 度量都是由答复报文中所通告的距离加上和发送答复报文的邻居路由器相连的链路代价计算得出的。选择一台后继路由器是基于最低的度量值的, 而且这个最低的度量值也就被设置为可行距离 **FD**。任何一台可行后继路由器如果不满足关于这个新的可行距离 **FD** 的可行性条件 **FC** 的话, 就会从拓扑结构表中被删除。注意, 在收到所有的答复报文之前不会选择后继路由器。

因为有多类型的输入事件 (input events) 能够引起一条路由改变它的状态, 所以当一条路由处于活动状态时就说明可能发生了一些类型的输入事件。DUAL 定义了多种活动状态。查询始发标记 (Query origin flag (O)) 用来指出当前的状态。图 8-11 和表 8-1 中显示了完整的 DUAL 有限状态机。

有两个例子可以帮助我们阐明 DUAL 的处理过程。图 8-12 中显示了例子的网络拓扑, 这里只需要关注到达子网 10.1.7.0 的每一个路由器的路径; 参考图 8-3 中指定的地址。在数据链路上, 箭头指出了每一台路由器用来到达子网 10.1.7.0 的后继路由器。在圆括号中显示的分别是每一台路由器到达子网的本地计算距离、路由器的可行距离 **FD**、答复状态标记 (reply status flag (r)) 和查询始发标记 (Query origin flag (O))。活动路由器 (active router) 用一个圆圈指示。

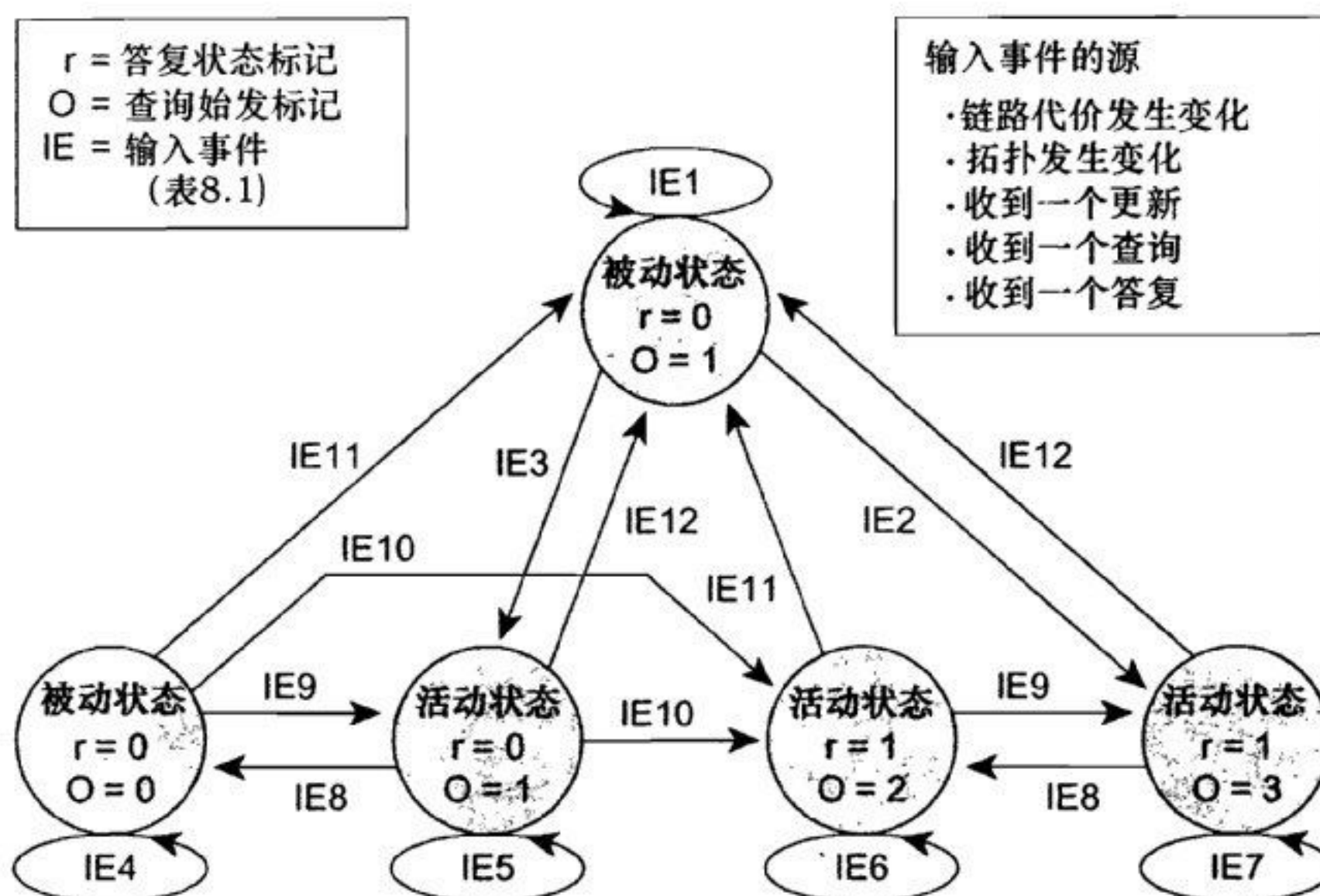


图 8-11 DUAL 有限状态机。查询始发标记 (Query origin flag (O)) 标识出扩散计算当前的状态。参看表 8-1 关于每一个输入事件 (IE) 的解释

表 8-1

DUAL 有限状态机的输入事件

输入事件	描述
IE1	满足可行性条件 FC 或者目的地不可到达的任何输入事件
IE2	从后继路由器收到了查询报文：不满足可行性条件 FC
IE3	除了来自于后继路由器的查询的其他输入事件：不满足可行性条件 FC
IE4	除了最新的答复或来自于后继路由器的查询的输入事件
IE5	除了最新的答复、来自于后继路由器的查询或到达目的地距离的增加的输入事件
IE6	除了最新答复的输入事件
IE7	除了最新的答复或到达目的地距离的增加的输入事件
IE8	到达目的地距离的增加
IE9	收到了最新的答复：当前可行距离 FD 不满足可行性条件 FC
IE10	从后继路由器收到了查询
IE11	收到了最新的答复：可行性条件 FC 和当前的可行距离 FD 匹配
IE12	收到了最新的答复：设置可行距离 FD 为无限大

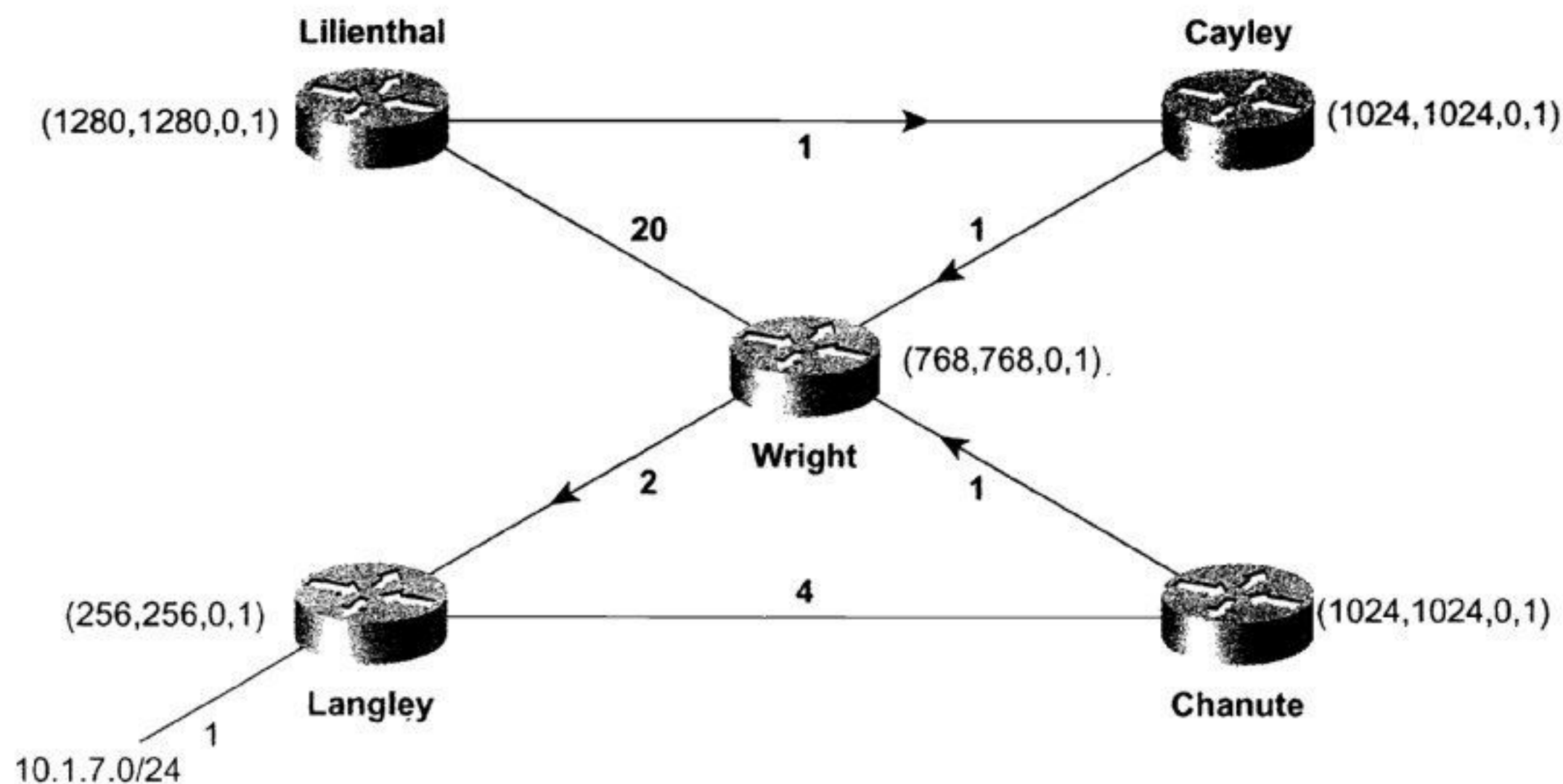


图 8-12 达到子网 10.1.7.0 的所有路由都是被动状态的，通过 r=0 和 O=1 指示

3. 扩散计算: 范例 1

这个例子关注的是路由器 Cayley 和它到达子网 10.1.7.0 的路由。在图 8-13 中, 路由器 Cayley 和 Wright (10.1.1.1) 之间的链路是失效的。EIGRP 把失效的链路当作一条拥有无穷大距离的链路。¹ 路由器 Cayley 检查它的拓扑结构表, 来查找到达子网 10.1.7.0 的可行后继路由器, 但是没有找到, 如图 8-6 所示。

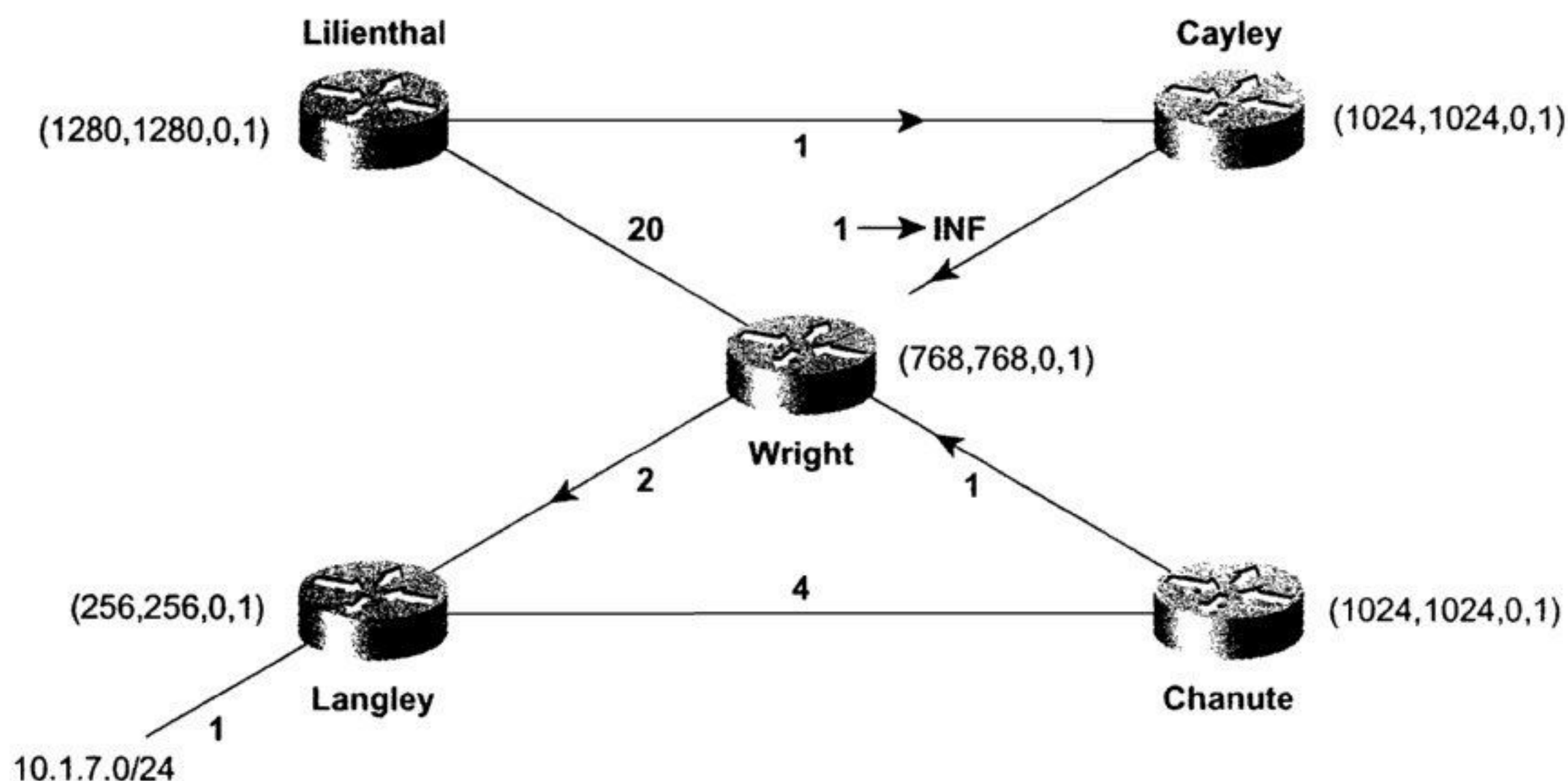


图 8-13 路由器 Cayley 和 Wright 间的链路是失效的, 因而路由器 Cayley 没有一个到达子网 10.1.7.0 的可行后继路由器

如图 8-14 所示, 路由器 Cayley 的路由变成了活动状态。这条路由的距离和可行距离 FD 也变为不可到达的了, 并且路由器 Cayley 把一个包含新的距离的查询报文发送给它的邻居路由器 Lilienthal。路由器 Cayley 关于路由器 Lilienthal 的答复状态标记被设置为 1, 用来指出期待一个答复报文, 因此输入事件是一个查询的未接受状态 (IE3), O=1。

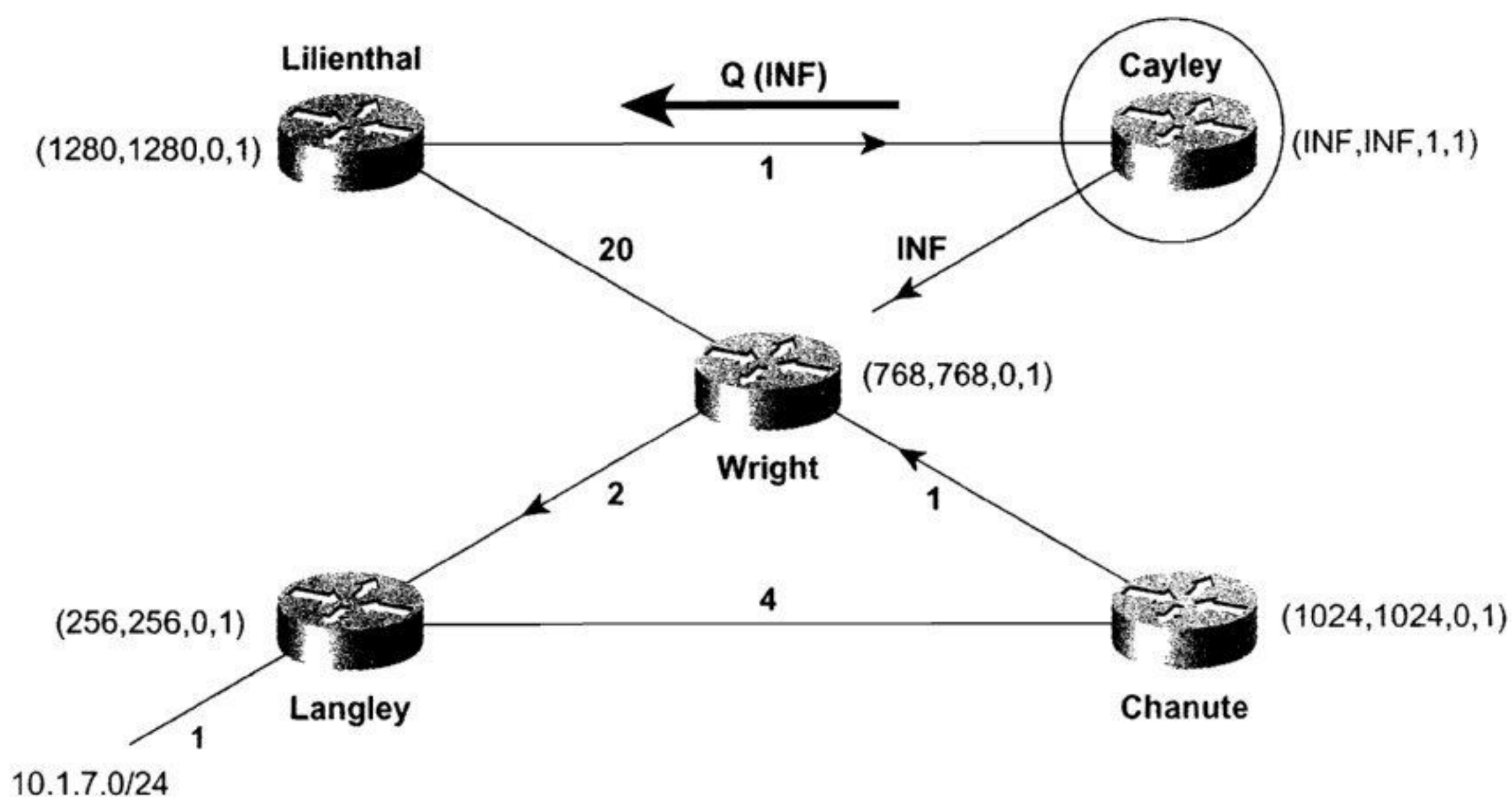


图 8-14 路由器 Cayley 到达子网 10.1.7.0 的路由转变成活动状态了, 并且为了确定可行后继路由器而去查询路由器 Lilienthal

¹ 无穷大距离可以用 0xFFFFFFFF 或 4294967295 来表示。

一旦收到了查询报文, 路由器 Lilienthal 将执行一个本地计算, 如图 8-15 所示。参见图 8-9, 由于路由器 Lilienthal 拥有到达子网 10.1.7.0 的可行后继路由器, 因而它的路由将不会变为活动状态。路由器 Wright 成为新的后继路由器, 而且路由器 Lilienthal 将发送一个含有它经过路由器 Wright 到达子网 10.1.7.0 的距离的答复报文。因为到达子网 10.1.7.0 的距离已经增加了, 而且路由不处在活动状态, 因此路由器 Lilienthal 的可行距离 FD 没有变化。

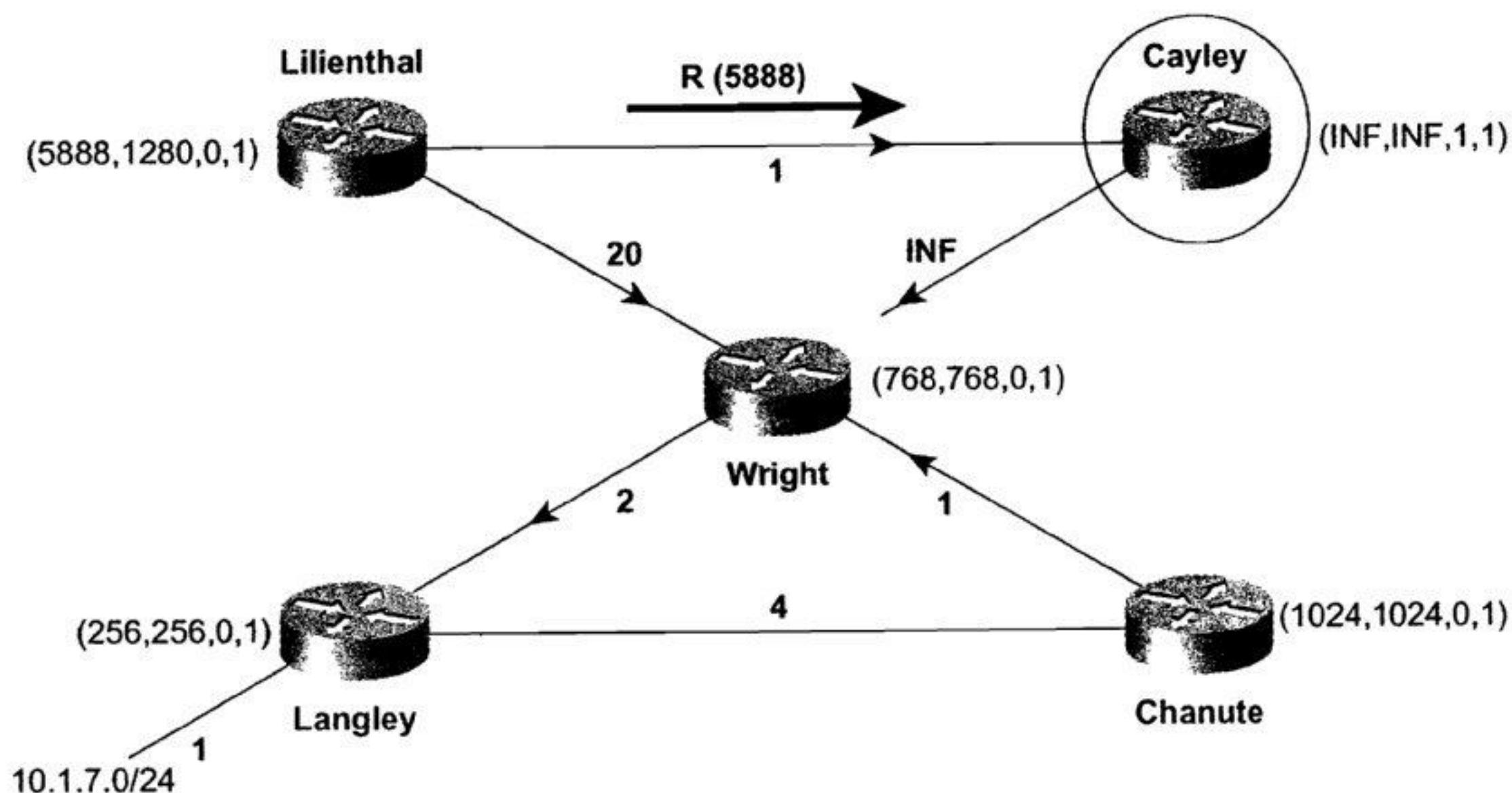


图 8-15 路由器 Lilienthal 有一个到达子网 10.1.7.0 的可行后继路由器。它执行一个本地的计算, 向路由器 Cayley 发送一个包含它经过路由器 Wright 到达目的子网的距离的答复报文, 而且发送一个更新给路由器 Wright

一旦从路由器 Lilienthal 收到一个答复报文, 路由器 Cayley 就设置 $r=0$, 路由也就变成了被动状态, 如图 8-16 所示。路由器 Lilienthal 成为路由器 Cayley 的新后继路由器, 可行距离 FD 也将设置成新的距离。最后, 路由器 Cayley 发送给路由器 Lilienthal 一个包含它的本地计算度量的更新。路由器 Lilienthal 也发送新的更新报文来通告它的新度量值。

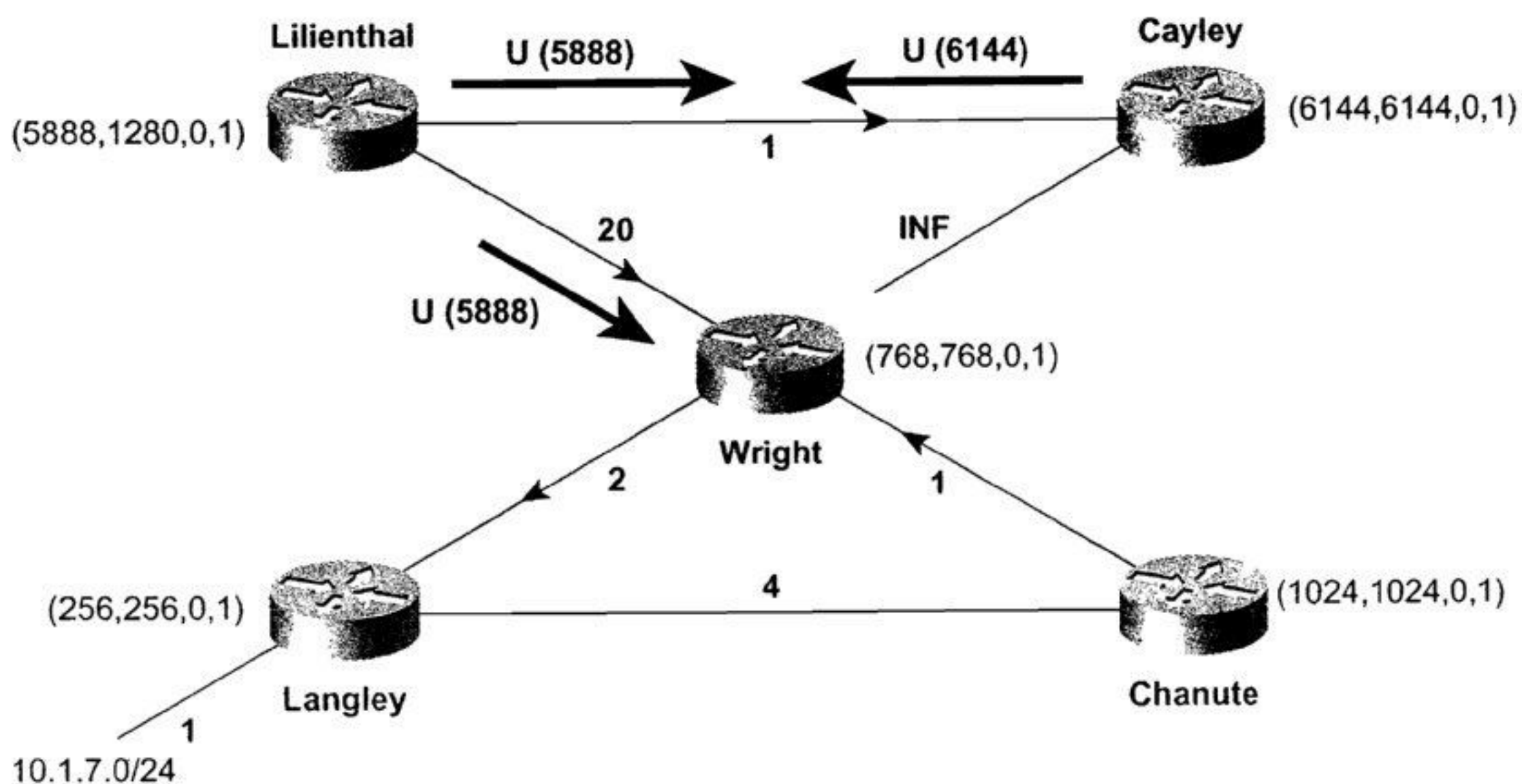


图 8-16 路由器 Cayley 到达子网 10.1.7.0 的路由变为被动状态, 而且发送一个更新报文给路由器 Lilienthal

EIGRP 协议报文的行为可以通过调试命令 **debug eigrp packets** 来观察。缺省情况下, 路由器将会显示所有的 EIGRP 报文。由于大量 Hello 报文和 ACK 报文的调试信息的输出可能很难去跟踪, 因此这个命令允许使用关键字选项以便只显示指定的报文类型。在图 8-17 中, 命令 **debug eigrp packets query reply update** 用来在路由器 Cayley 上观察本例中描述的事件的报文行为。

```
Cayley#debug eigrp packet update query reply
EIGRP Packets debugging is on
  (UPDATE, QUERY, REPLY)
B#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
EIGRP: Enqueueing QUERY on Serial0 iibq un/rely 0/1 serno 45-49
EIGRP: Enqueueing QUERY on Serial0 nbr 10.1.6.1 iibq un/rely 0/0 peerQ un/rely
0/0 serno 45-49
EIGRP: Sending QUERY on Serial0 nbr 10.1.6.1
  AS 1, Flags 0x0, Seq 45/64 idbQ 0/0 iibq un/rely 0/0 peerQ un/rely 0/1 serno
45-49
EIGRP: Received REPLY on Serial0 nbr 10.1.6.1
  AS 1, Flags 0x0, Seq 65/45 idbQ 0/0 iibq un/rely 0/0 peerQ un/rely 0/0
EIGRP: Enqueueing UPDATE on Serial0 iibq un/rely 0/1 serno 50-54
EIGRP: Enqueueing UPDATE on Serial0 nbr 10.1.6.1 iibq un/rely 0/0 peerQ un/rely
0/0 serno 50-54
EIGRP: Sending UPDATE on Serial0 nbr 10.1.6.1
  AS 1, Flags 0x0, Seq 46/66 idbQ 0/0 iibq un/rely 0/0 peerQ un/rely 0/1 serno
50-54
EIGRP: Received UPDATE on Serial0 nbr 10.1.6.1
  AS 1, Flags 0x0, Seq 67/46 idbQ 0/0 iibq un/rely 0/0 peerQ un/rely 0/1
```

图 8-17 在本例中描述的 EIGRP 报文事件可以通过这些调试信息进行观察

- **标记(Flags)**——在输出的调试信息中, 指出 EIGRP 报文头部的标记的状态, EIGRP 报文头部请参阅本章后面讲述的“EIGRP 报文头部”一节。0x0 表示没有标记被设置。0x1 表示设置了初始化(initialization)位, 在一个新的邻居关系中, 当附加的路由条目是首个时, 这个标记就被设置。0x2 表示设置了条件接收位(conditioned receive bit), 这个标记用在私有的可靠组播算法中。
- **序列号(Seq)**——表示一个报文序列号/确认报文序列号;
- **idbq**——表示在接口上的输入队列报文数/输出队列报文数;
- **iibq**——表示在接口上等待传送的不可靠组播报文数/等待传送的可靠组播报文数;
- **peerQ**——表示在接口上等待传送的不可靠单播报文数/等待传送的可靠单播报文数;
- **serno**——表示一个指向某条路由的双重连接的序列号的指针。这个指示器使用在内部(或私有)的机制, 用来在一个快速变化的拓扑中跟踪正确的路由信息。

4. 扩散计算: 范例 2

这个例子所关注的是路由器 Wright 和它到达子网 10.1.7.0 的路由。虽然在这里描述的输入事件(在扩散更新计算的期间链路的延迟变化了两次)在现实的网络中未必会发生, 但是这个例子显示了 DUAL 算法是怎样控制多种度量的变化的。

在图 8-18 中, 把路由器 Wright 和 Langley 之间的链路代价由 2 改变成 10。经过路由器 Langley 到达 10.1.7.0 的距离现在超过了路由器 Wright 的可行距离, 这将引起路由器 Wright 开始进行本地度量计算。这时度量将被更新, 除了和链路代价发生改变的链路相连的邻居路

由器 Langley 外, 路由器 Wright 会向它所有的邻居发送更新报文, 如图 8-19 所示。

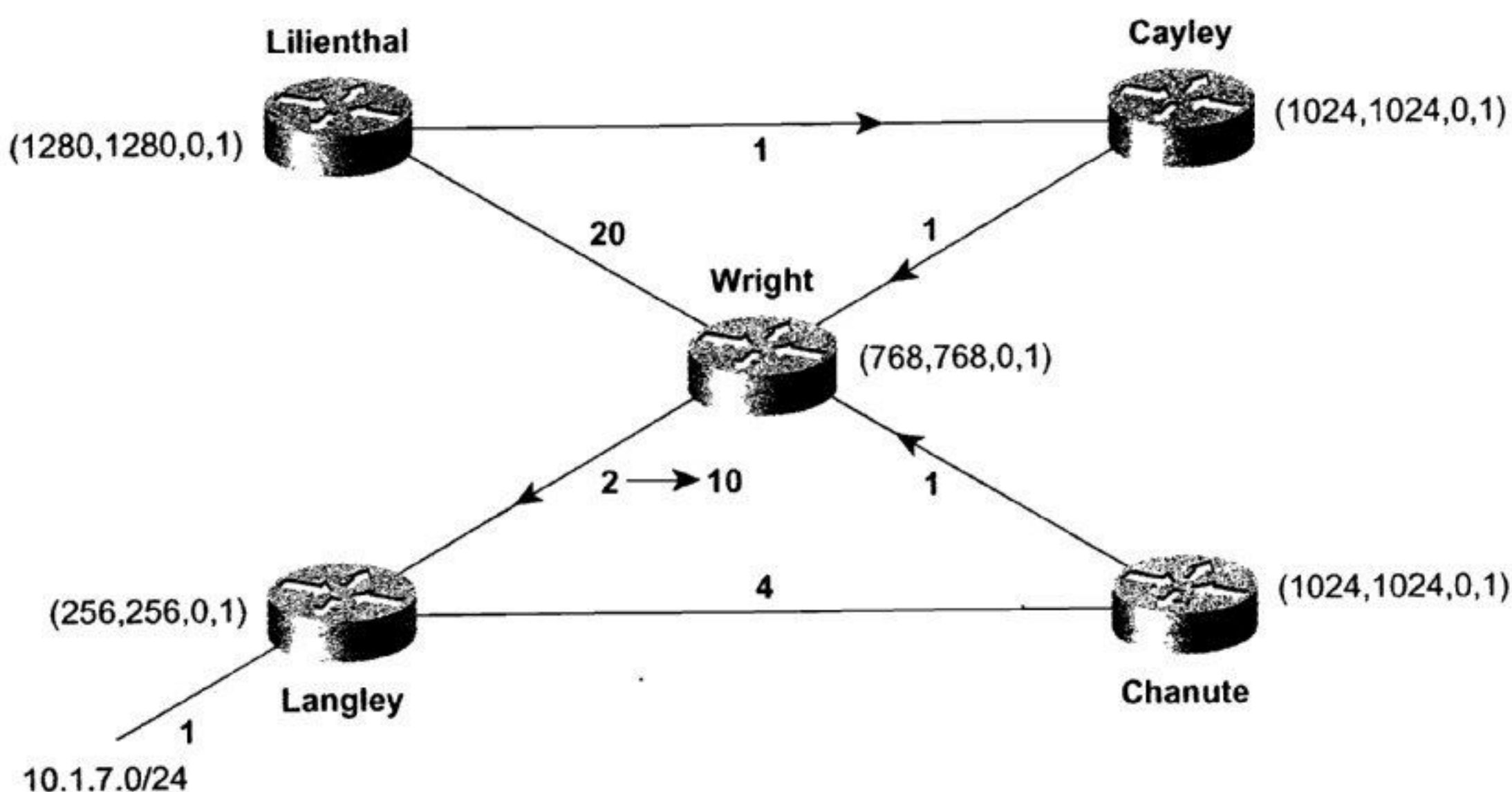


图 8-18 路由器 Cayley 到达子网 10.1.7.0 的路由变为被动状态, 并且向路由器 Lilienthal 发送一个更新报文

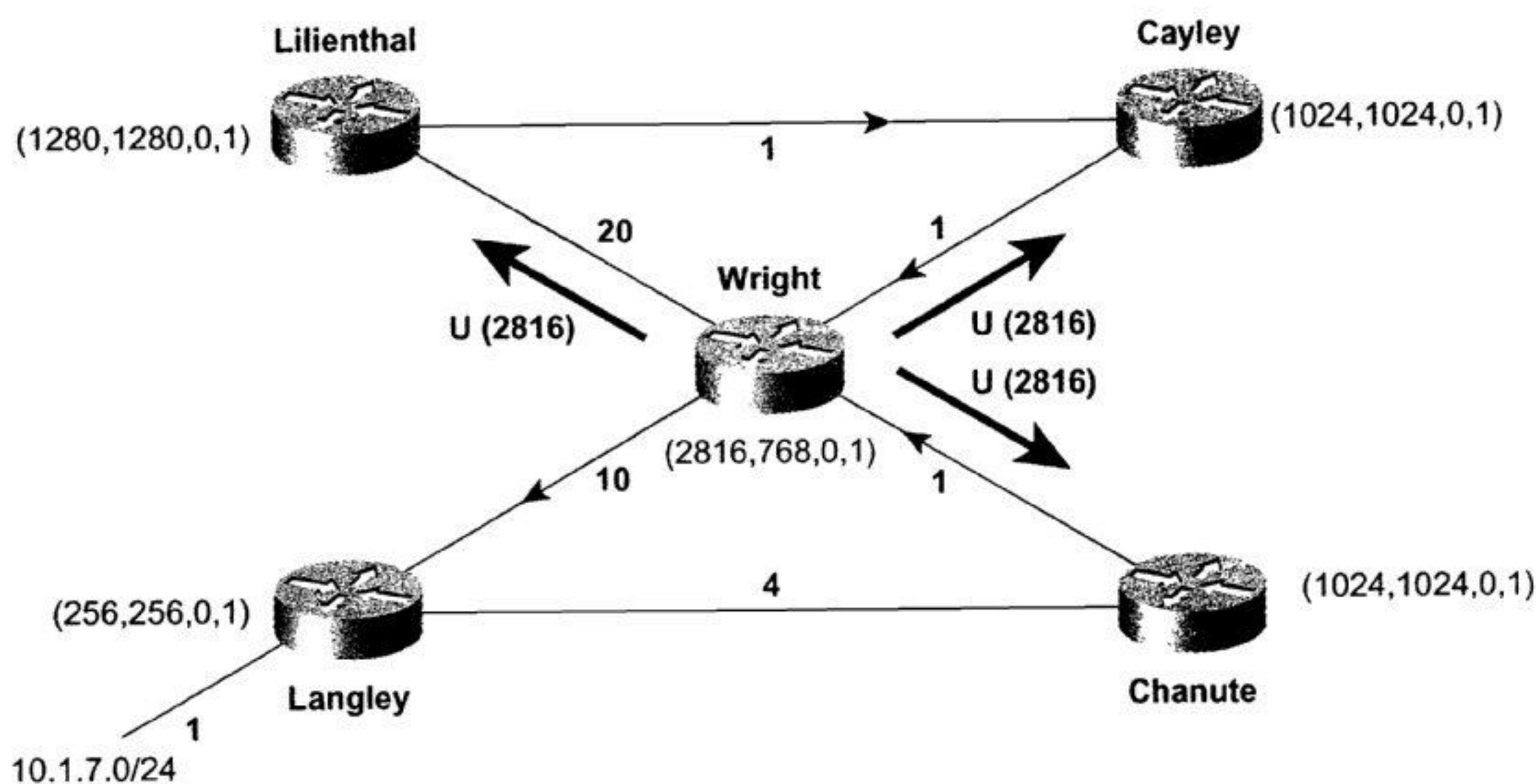


图 8-19 路由器 Wright 发送含有新度量值的更新报文到除了路由器 Langley 之外的所有邻居路由器

注意: 路由器 Langley 是到达子网 10.1.7.0 的惟一可行后继路由器, 这是因为路由器 Chanute 的本地计算度量高于路由器 Wright 的可行距离 FD ($1024 > 768$)。路由器 Wright 和 Langley 之间链路度量的增加引起路由器 Wright 在它的拓扑结构表中去查找一台新的后继路由器。由于路由器 Wright 在它的拓扑结构表中可以查到的惟一可行后继路由器是路由器 Langley, 因此, 路由器 Wright 到达子网 10.1.7.0 的路由将变为活动状态。查询报文也被发送到邻居路由器, 如图 8-20 所示。

同时, 在图 8-19 中, 路由器 Wright 发出的更新信息将引起路由器 Cayley、Lilienthal 和 Chanute 分别执行一个本地计算。

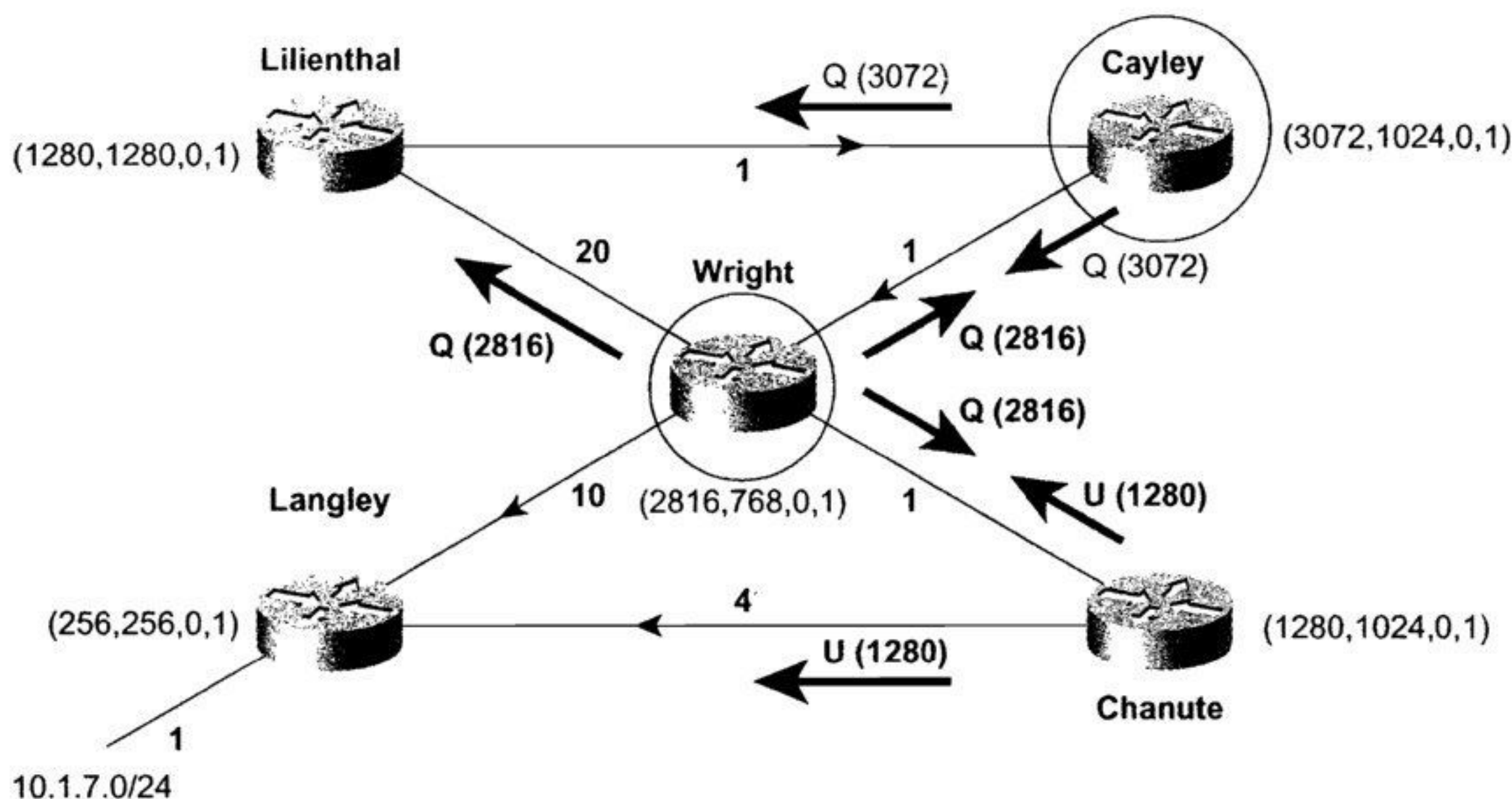


图 8-20 路由器 Wright 到达 10.1.7.0 的路由变为活动状态, Wright 向它的邻居发出查询, 以便选择一个可行后继路由器。为了响应来自路由器 Wright 的早期报文, 路由器 Cayley 使它的路由成为活动状态并向它的邻居路由器发出查询报文; 同样, 路由器 Chanute 也改变了它的度量值并发送出更新

在路由器 Cayley 上, 现在经过路由器 Wright 的路由距离超过了路由器 Cayley 的可行距离 FD ($2816 > 1024$)。因而, 路由变为活动状态并向它的邻居路由器发送查询报文。

在图 8-20 中, 路由器 Lilienthal 正在使用路由器 Cayley 作为一台后继路由器, 但是还没有收到从路由器 Cayley 发出的查询报文。因此, 路由器 Lilienthal 只不过重新计算了经过路由器 Wright 的路径的度量值, 发现它不再满足可行性条件 FC, 从而从拓扑结构表中删除了这条路径。

在路由器 Chanute 看来, 路由器 Wright 是它的后继路由器。因为路由器 Wright 所通告的距离不再满足路由器 Chanute 的可行性条件 FC ($2816 > 1024$), 并且因为路由器 Chanute 还有一台可行后继路由器 (参见图 8-8), 因此, 路由器 Chanute 将会把路由器 Wright 从它的拓扑结构表中删除。随后, 路由器 Langley 变成了路由器 Chanute 的后继路由器, 度量值被更新后, 路由器 Chanute 向它的邻居路由器发送更新报文 (参见图 8-20)。而路由器 Chanute 上的路由则从来没有变为活动状态。

路由器 Cayley、Lilienthal 和 Chanute 都分别以不同的答复报文来响应发源于路由器 Wright 的查询报文, 如图 8-21 所示。

路由器 Cayley 已经是活动状态, 因为输入事件是来自于后继路由器的查询, 这个查询最初标记为 2 ($O=2$), 参见图 8-11 和表 8-1。

路由器 Lilienthal 一旦收到路由器 Wright 的查询, 就发送一个含有经过路由器 Cayley 的距离的答复报文。然而, 就在刚发出这个答复报文后, 路由器 Lilienthal 收到了来自于路由器 Cayley 的查询, 这时, 路由器 Lilienthal 的可行距离超出了, 因而度量值将被更新, 路由器 Lilienthal 使路由变为活动状态, 并向它的邻居路由器发出查询。

路由器 Chanute 已经把它的路由器切换到路由器 Langley, 并且仅仅发出一个回复报文。

当上述的一切正在继续进行的时候, 图 8-21 中显示了路由器 Wright 和 Langley 之间的链

路由器 Lilienthal 在收到了它发送的所有查询报文的答复后, 将把路由的状态转变成被动状态, 如图 8-23。这时, 路由就可以设置新的可行距离 FD 了。由于路由器 Cayley 所通告的路由距离低于路由器 Lilienthal 的可行距离, 因而它仍然保持是后继路由器。路由器 Lilienthal 也发送一个答复报文来响应路由器 Cayley 的查询。

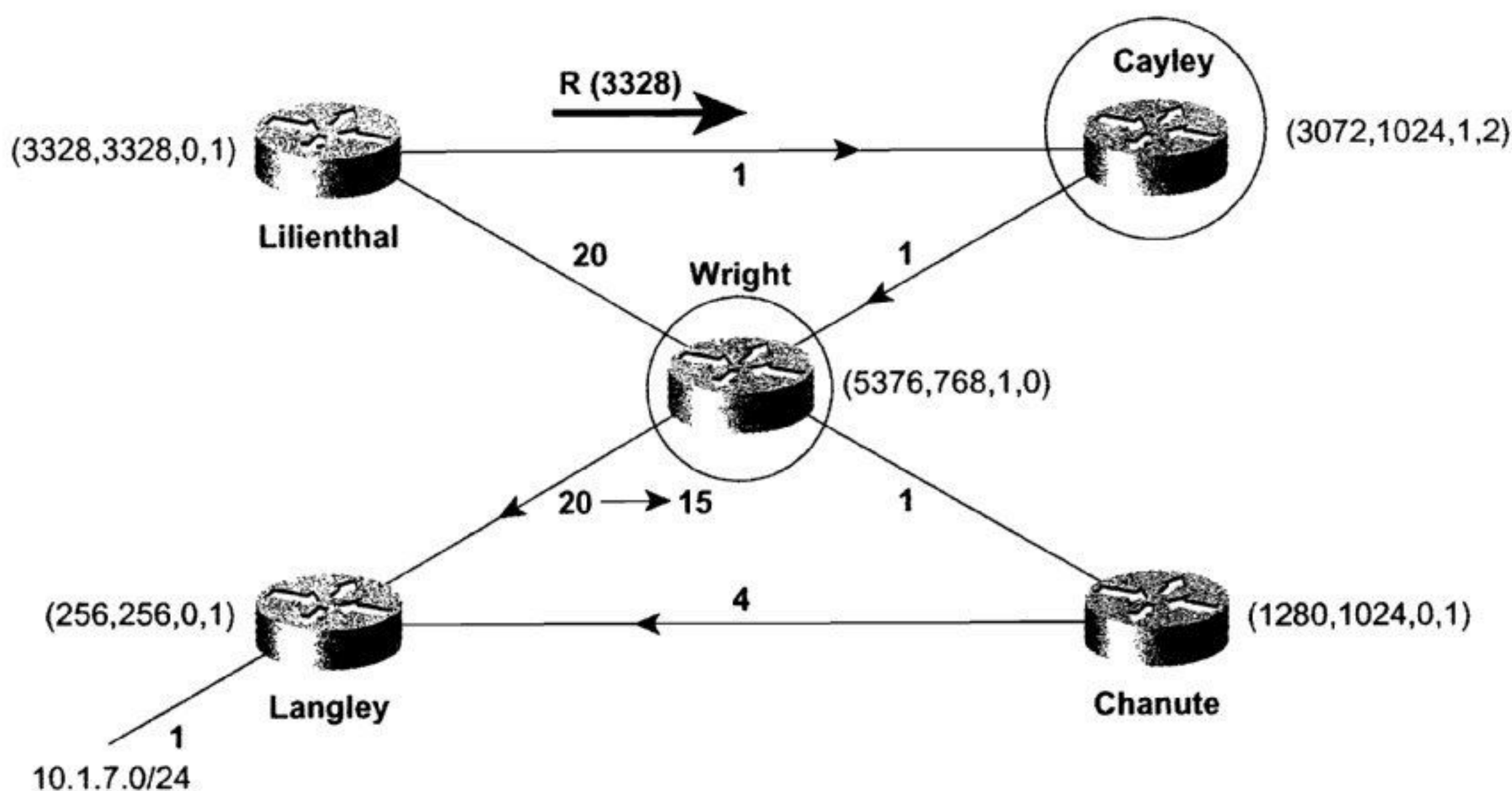


图 8-23 收到了所希望得到的最后一个答复后, 路由器 Lilienthal 将使它的路由转变为被动状态 ($r=0$, $O=1$)

图 8-23 中显示了路由器 Wright 和 Langley 之间的链路距离再次从 20 改变成 15。路由器 Wright 也再次计算出它的路由的本地距离为 4096, 如图 8-24。如果在路由变为被动状态之前, 路由器 Wright 收到一个查询报文, 那么它将仍然通告含有距离为 2816 的路由, 2816 是在路由变为活动状态时的距离值。

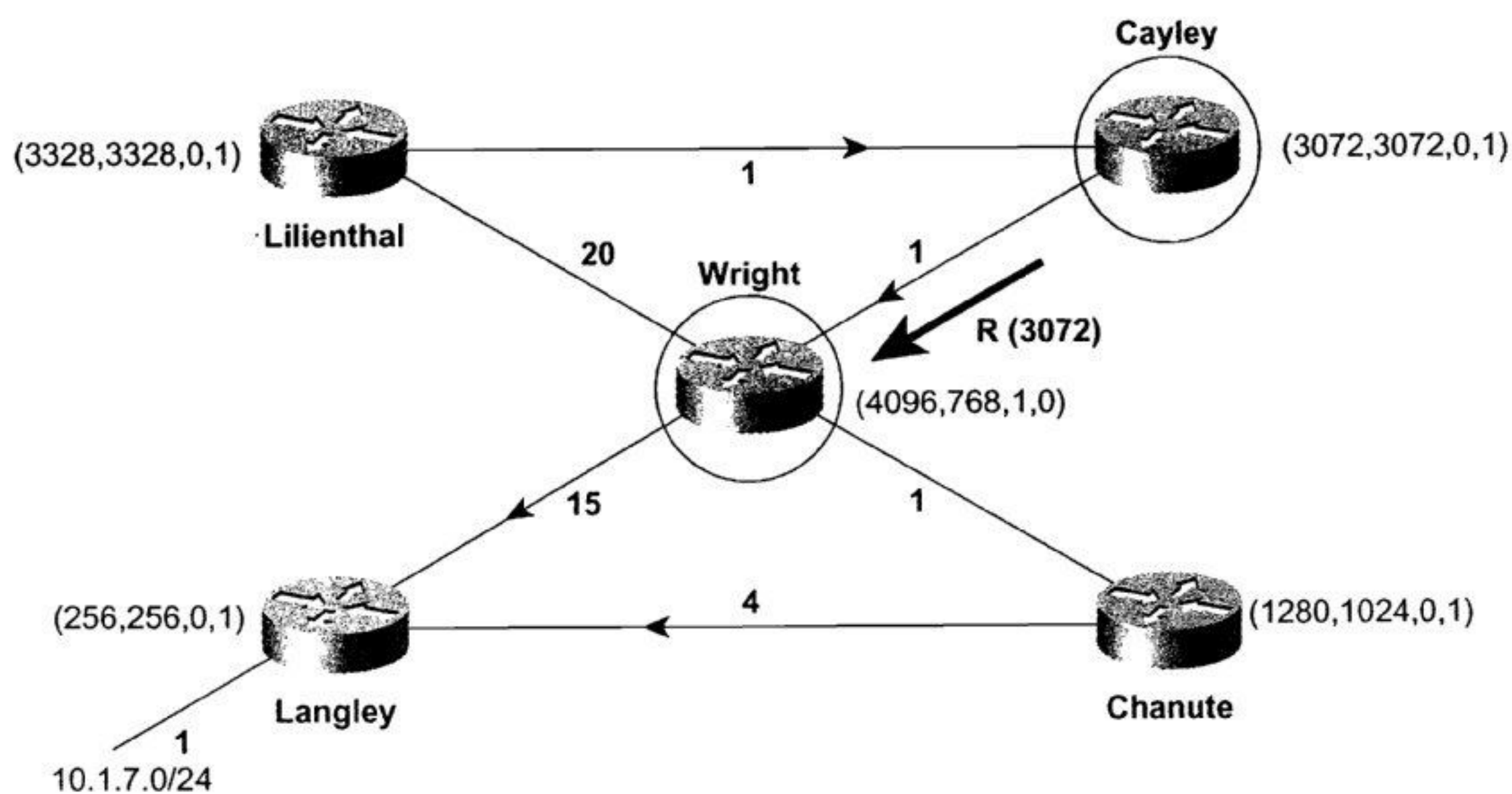


图 8-24 收到了所希望得到的最后一个答复后, 路由器 Cayley 将使它的路由状态转变为被动状态

当路由器 Cayley 收到它所发送的查询报文的答复时, 它到达子网 10.1.7.0 的路由也将变成被动状态, 如图 8-24 所示, 将设置一个新的可行距离 FD。虽然路由器 Wright 在本地计算的度

量值是 4096, 但是它所通告的最新度量值却是 2816。因此, 路由器 Wright 满足路由器 Cayley 的可行性条件 FC, 从而变为到达子网 10.1.7.0 的后继路由器, 并发送一个答复给路由器 Wright。

在图 8-25 中, 路由器 Wright 收到了它所发出的每一个查询的答复后, 它的路由状态就变为被动状态了。路由器 Wright 选择了路由器 Chanute 作为它的新后继路由器, 并且把可行距离 FD 改变为路由器 Chanute 所通告的距离与它和邻居路由器 (Chanute) 之间的链路代价的总和。路由器 Wright 发送一个更新报文给它所有的邻居路由器, 并通告它在本地所计算的新度量值。

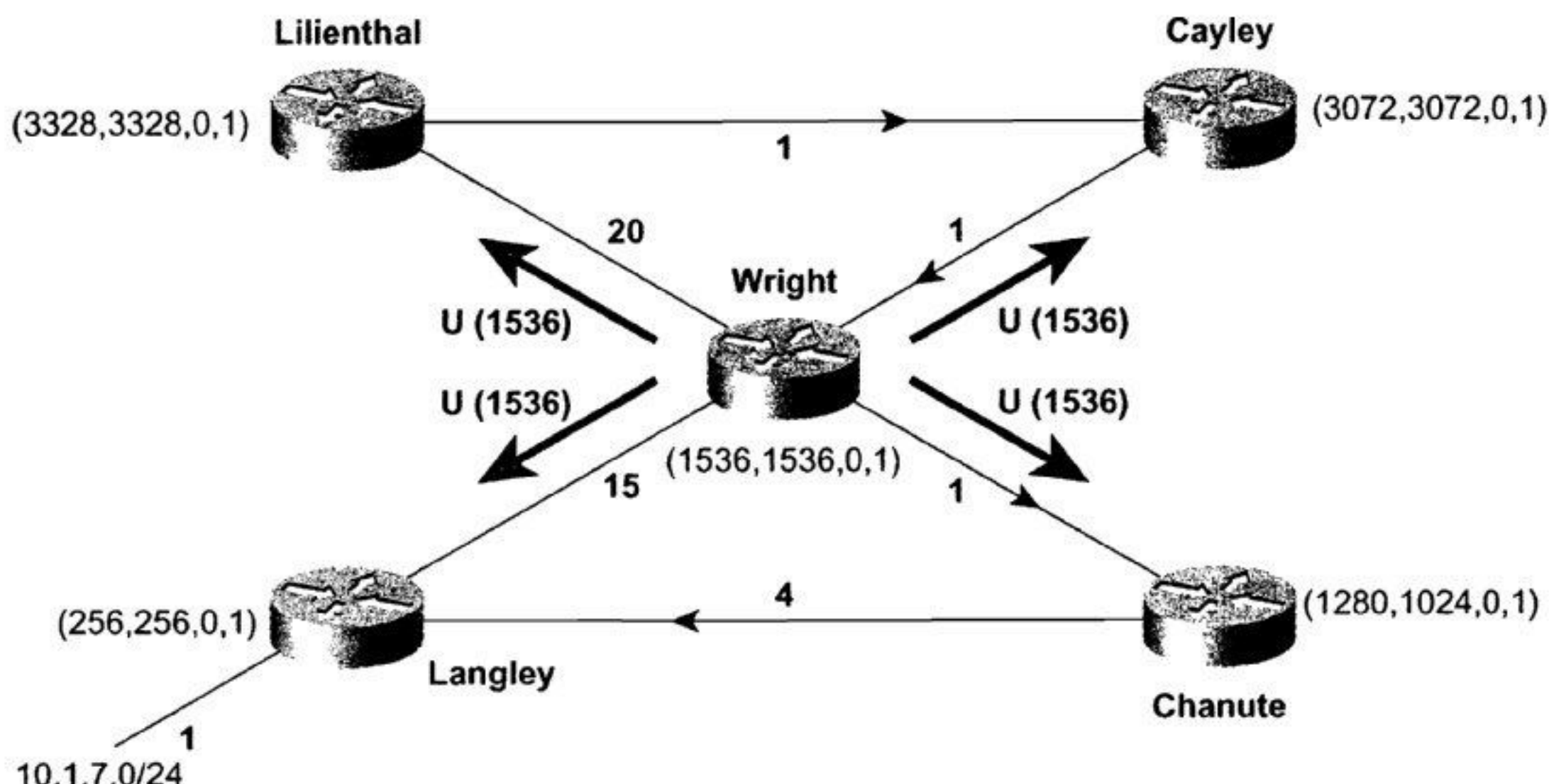


图 8-25 路由器 Wright 转换到被动状态, 选择路由器 Chanute 作为它的后继路由器, 同时改变可行距离 FD, 并且更新所有的邻居路由器

路由器 Cayley 使用路由器 Wright 作为它的后继路由器。当它收到一个来自于路由器 Wright 的并有较低代价的更新时, 它将改变它在本地的计算度量和可行距离 FD, 并且更新它的邻居路由器, 如图 8-26 所示。

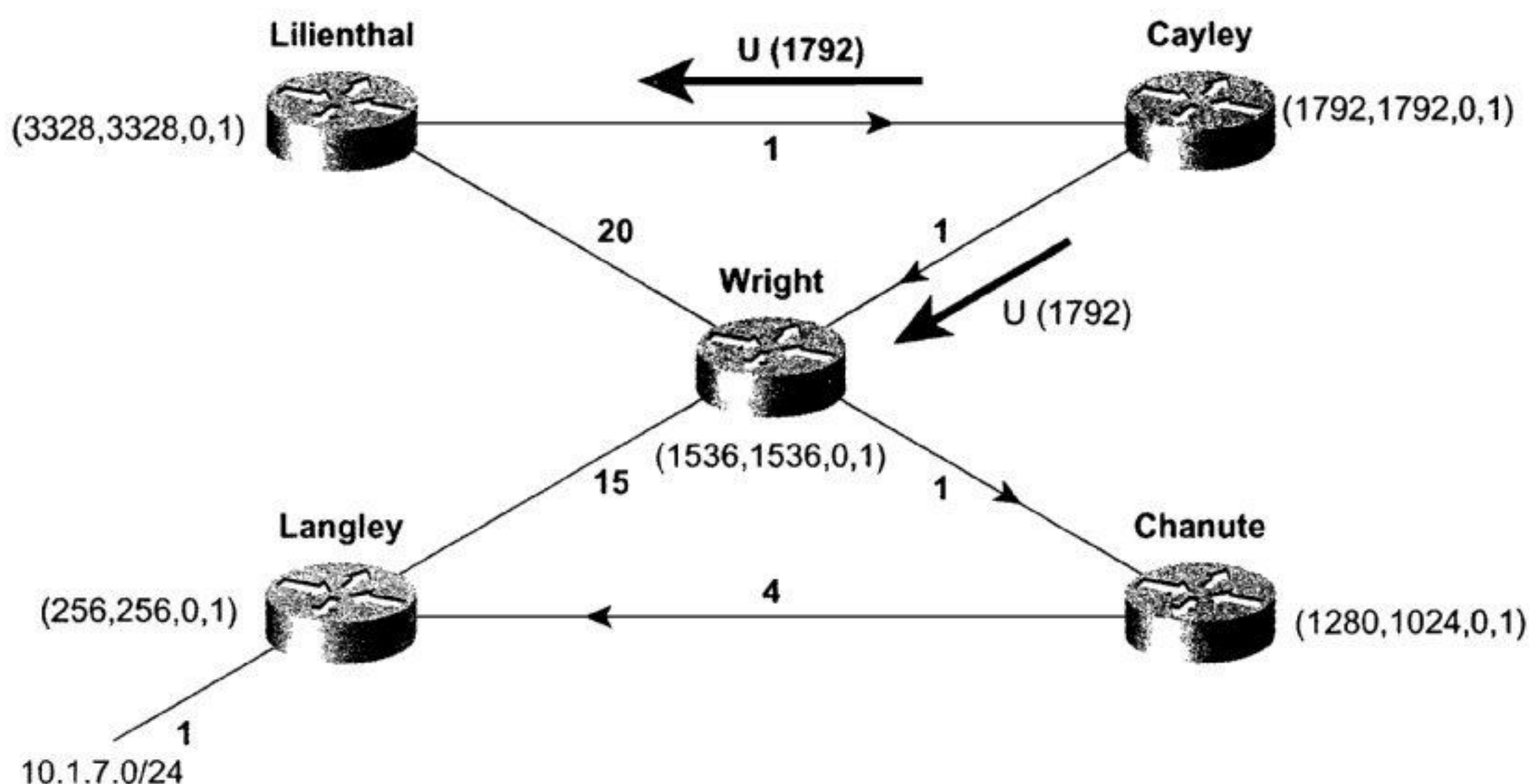


图 8-26 路由器 Cayley 重新计算它的度量值, 根据路由器 Wright 通告的较低的代价更改它的可行距离 FD, 并更新它的邻居路由器

来自路由器 Cayley 的更新并不影响路由器 Wright，这是因为路由器 Cayley 不满足路由器 Wright 那儿的可行性条件 FC。在路由器 Lilienthal 上的更新会引起一个本地计算。

如图 8-27 所示，路由器 Lilienthal 减小了它的度量值，减小了可行距离 FD，并更新它的邻居。

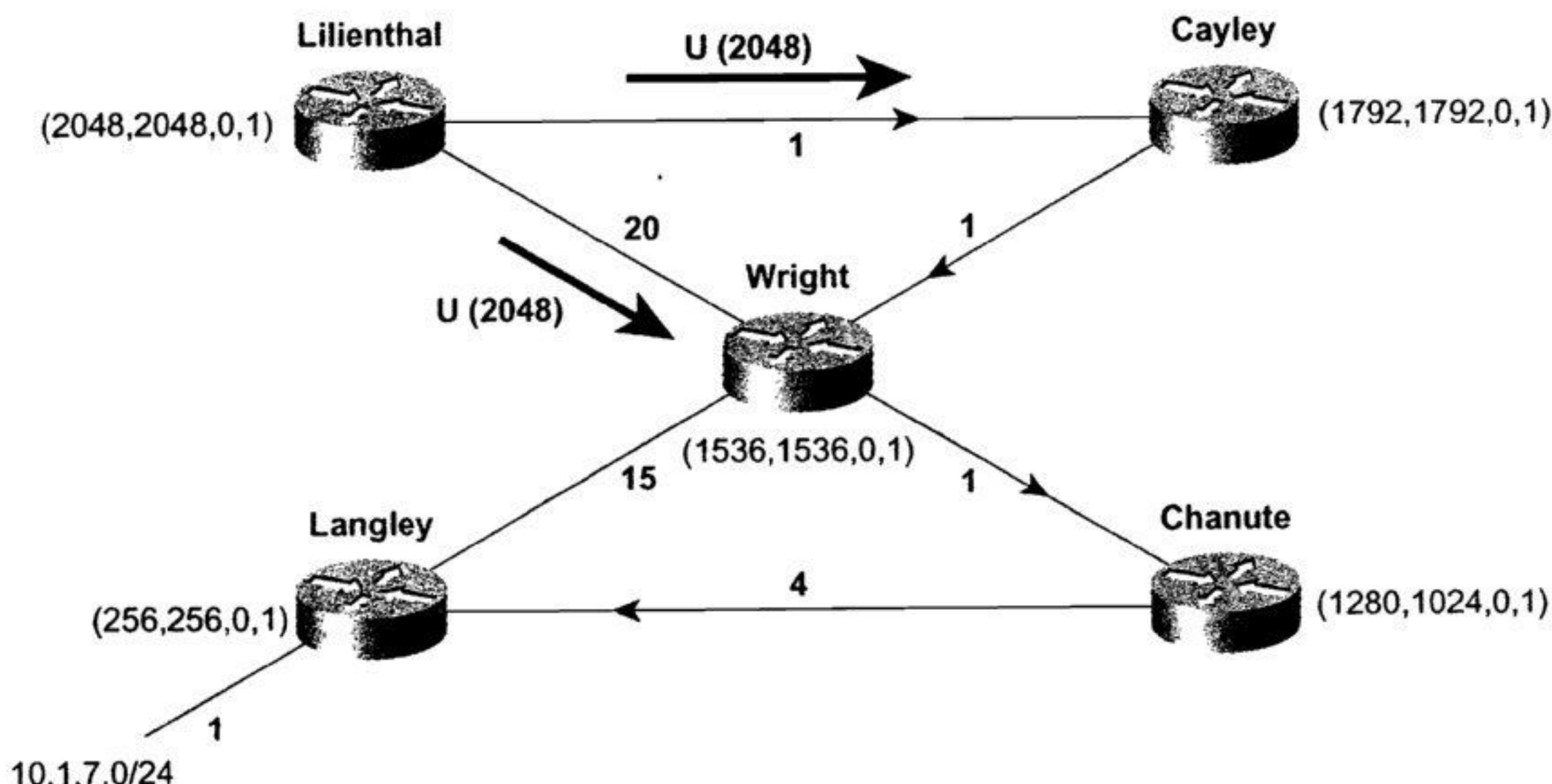


图 8-27 路由器 Lilienthal 重新计算它的度量，基于来自路由器 Cayley 的更新改变它的可行距离 FD，并更新它的邻居路由器

虽然，扩散计算算法可以通过更加详细的描述或阅读一些读物以便完全的理解，但是在这里所讲述的内容和前面的例子也包含了扩散计算算法的主要中心内容：

- 任何时间，一个输入事件发生了，就会执行一个本地的计算；
- 如果在路由器的拓扑结构表中发现了一台或多台可行后继路由器，那么将使用具有最低度量代价的可行后继路由器作为它的后继路由器；
- 如果没有发现可行后继路由器，那么将使它的路由变成活动状态，向它的邻居路由器发送查询报文，以便确定一个可行后继路由器；
- 在所有的查询报文被答复报文响应之前，或者活动计时器计时超时之前，将保持路由的状态是活动状态；
- 如果扩散计算的结果无法发现一个可行后继路由器，那么将宣告这个目的地不可到达。

8.1.5 EIGRP 的报文格式

EIGRP 协议报文的 IP 报文头部指定它的协议号是 88，报文的最大长度可以是 IP 的最大传输单元 (MTU) 的大小——通常是 1500 个 8bit 字节。紧接着 IP 头部后面的是 EIGRP 协议头部，EIGRP 协议头部后面是类型/长度/数值 (Type/Length/Value, TLV) 这 3 个参数的不同组合。这些 TLV 不仅携带路由条目的信息，而且提供多个字段来管理 DUAL 算法的处理、组播的先后次序和 IOS 软件版本。

1. EIGRP 报文的头部

图 8-28 中显示了 EIGRP 报文的头部，它是每个 EIGRP 报文的开始部分。

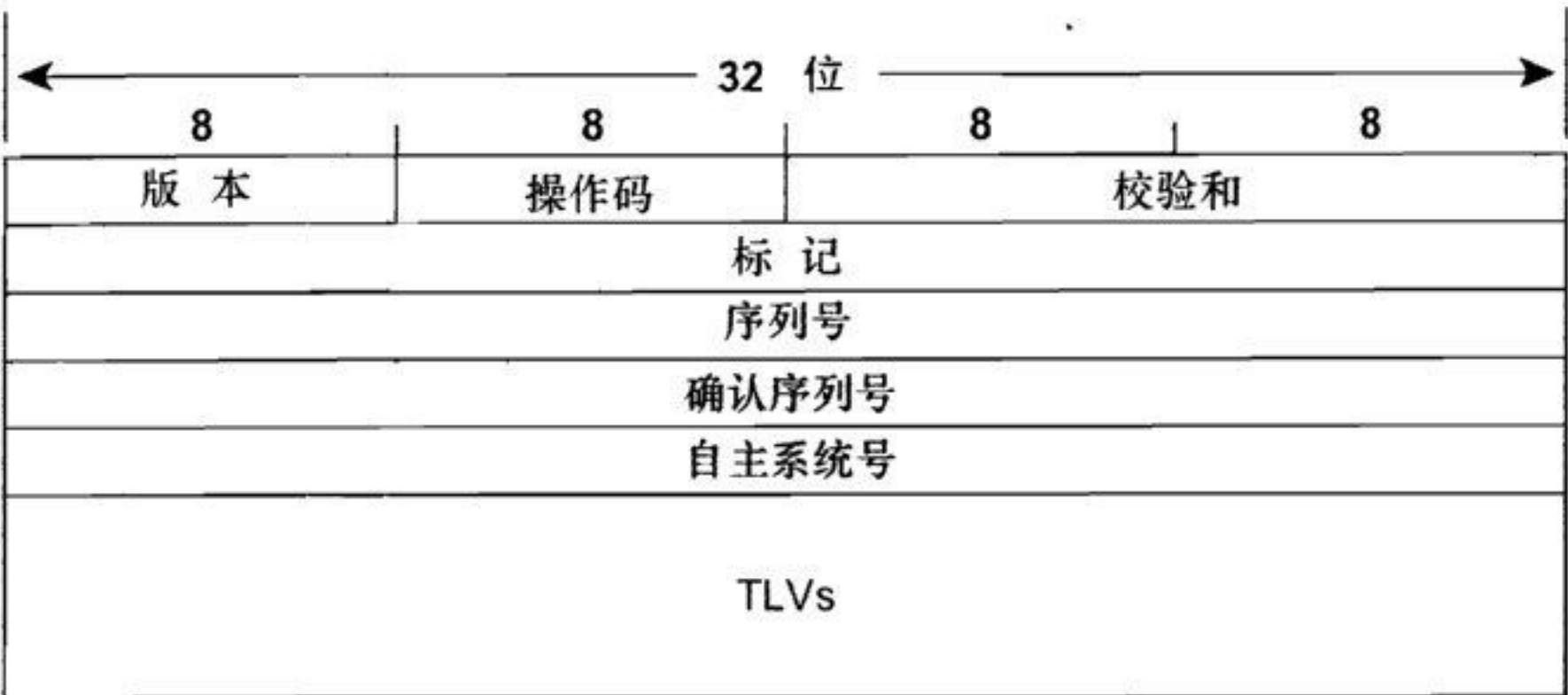


图 8-28 EIGRP 报文的头部

- **版本号 (Version)** ——指出始发 EIGRP 进程处理的具体版本。虽然关于 EIGRP 协议的两个软件版本目前都是可用的，¹但是 EIGRP 协议本身的版本却自发布后还没有改变过。
- **操作码 (Opcode)** ——指出 EIGRP 报文的类型，这显示在表 8-2 中。虽然表中包含了 IPX SAP 报文类型，但 IPX EIGRP 报文的讨论已经超出了本书的范围。

表 8-2 EIGRP 的报文类型

操作码 (Opcode)	类型 (Type)
1	更新 (Update)
3	查询 (Query)
4	答复 (Reply)
5	问候 (Hello)
6	IPX SAP

- **校验和 (Checksum)** ——标准的 IP 校验和。它是基于除了 IP 头部的整个 EIGRP 报文来计算的。
- **标记 (Flags)** ——目前包括两个标记。大部分的位设置为是 Init 位，也就是设置为 0x00000001，指出附加的路由条目是新的邻居关系的开始。第二位设置为 0x00000002，表示条件接收位 (Conditional Receive Bit)，并使用在一个私有的可靠组播算法中。
- **序列号 (Sequence)** ——是一个用在 RTP 中的 32 位序列号。
- **确认序列号 (ACK)** ——是本地路由器从邻居路由器那里收到的最新的一个 32 位序列号。一个包含有非零的 ACK 字段的 Hello 报文将被看作是一个 ACK 报文，而不看作一个 Hello 报文。注意，如果报文本身是单播的，这里的 ACK 字段只能是非零的，因为确认报文从来都不是组播的。
- **自主系统号 (Autonomous System Number)** ——指定一个 EIGRP 协议域的标识号。

跟在 EIGRP 头部后面的就是 TLV 字段，表 8-3 中列出了多种类型的 TLV 字段。虽然在本书中不讲 IPX 协议和 AppleTalk 协议类型，但在下面的表中也包含了。每一个 TLV 字段都包含一个表 8-3 中列出的 2 个 8bit 字节的类型号、一个指定 TLV 字段长度的 2 个 8bit 字节

¹ 从 IOS10.3 (11)、11.0 (8) 和 11.1 (3) 开始，由于软件稳定性的增强，因此，强烈建议使用 EIGRP 的后来版本。

的字段和一个由类型决定其格式的可变字段。

表 8-3 类型/长度/数值 (TLV) 的类型

数 值	TLV 类型
一般的 TLV 类型	
0x0001	EIGRP 参数
0x0003	序列 (Sequence)
0x0004	软件版本 ¹
0x0005	下一个组播序列
IP 特有的 TLV 类型	
0x0102	IP 内部路由
0x0103	IP 外部路由
AppleTalk 特有的 TLV 类型	
0x0202	AppleTalk 内部路由
0x0203	AppleTalk 外部路由
0x0204	Apple Talk 电缆配置
IPX 特有的 TLV 类型	
0x0302	IPX 内部路由
0x0303	IPX 外部路由

2. 一般的 TLV 字段

这些 TLV 字段可以携带 EIGRP 的管理信息而不需要指定任何一个可路由的协议。带参数的 TLV 用来传递度量权重和抑制时间, 如图 8-29 所示。序列、软件版本和下一个组播序列等 TLV 是用于 Cisco 的私有可靠性组播算法的, 超出了本书的讲述范围。

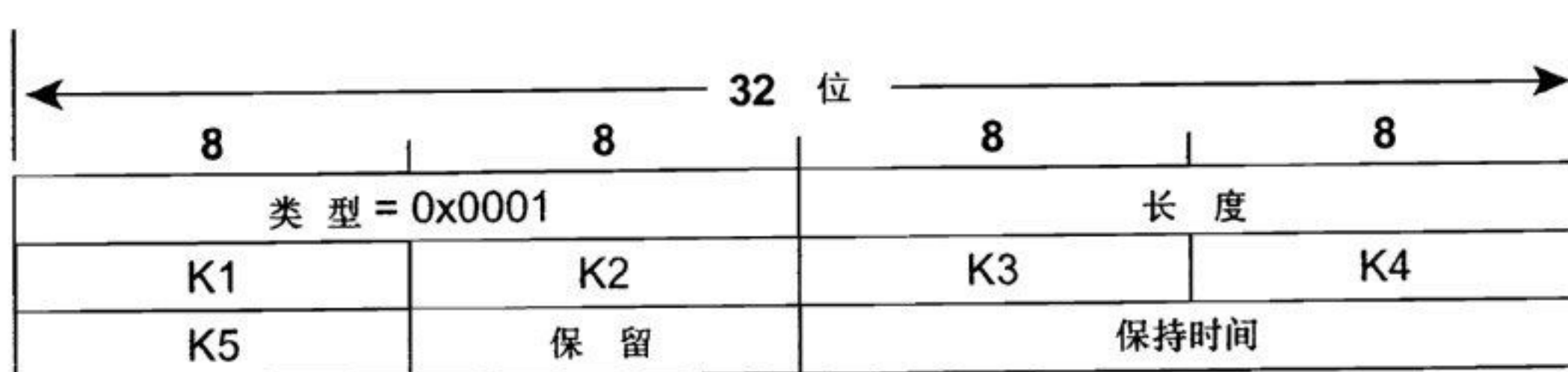


图 8-29 带 EIGRP 参数的 TLV

3. IP 特有的 TLV 字段

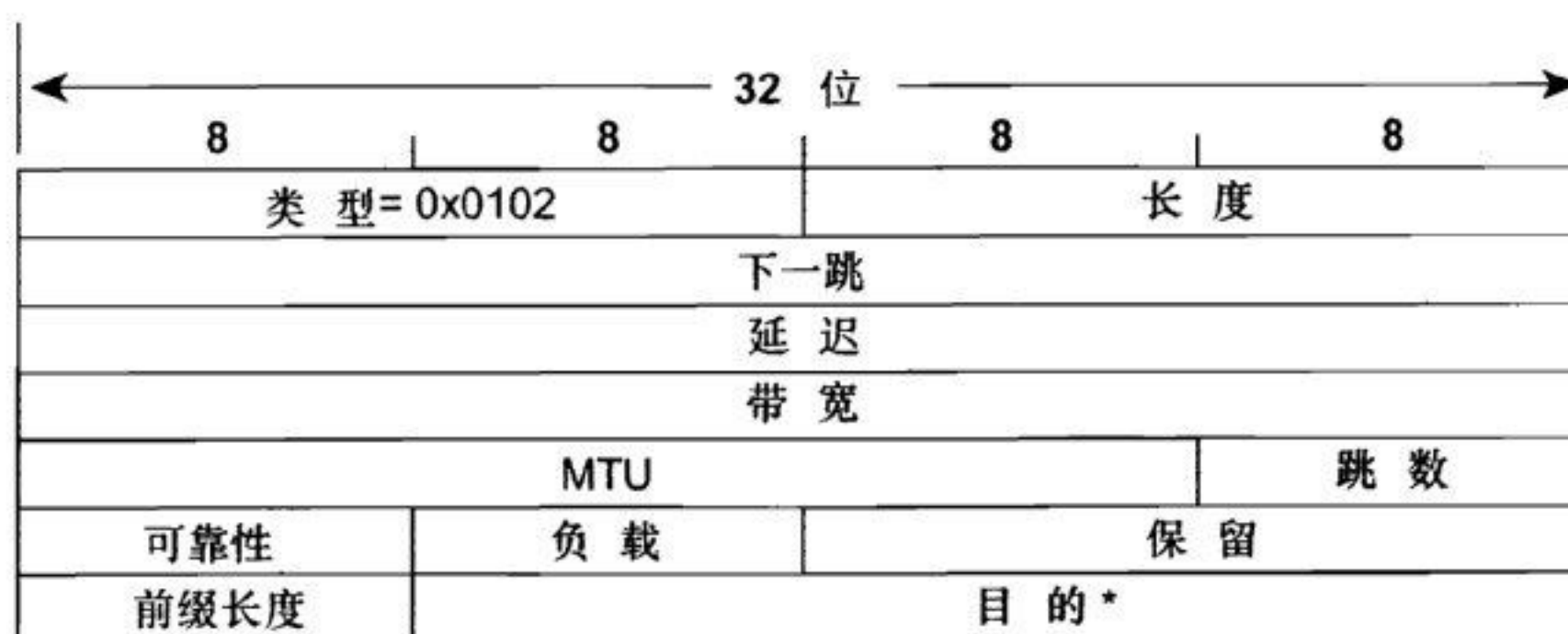
每一个内部路由和外部路由的 TLV 都包含一个路由条目。每个更新、查询和答复报文都至少包含一个路由 TLV。

内部路由和外部路由的 TLV 包括了路由的度量信息。就像早先提到的, EIGRP 协议使用的度量是和 IGRP 协议使用的度量相同的, 只是扩大了 256 倍。关于 IGRP 协议度量更详细的讲述, 连同复合度量的计算都已在第 6 章中讲述了。

(1) IP 内部路由的 TLV

内部路由是指在 EIGRP 自主系统内部可以到达目的地的路径。内部路由的 TLV 格式如图 8-30 所示。

¹ 这个报文指出的是软件的老版本在运行 (软件版本为 0) 还是软件从 IOS10.3(11)、11.0(8)和 11.1(3)起的新版本 (软件版本为 1) 在运行。

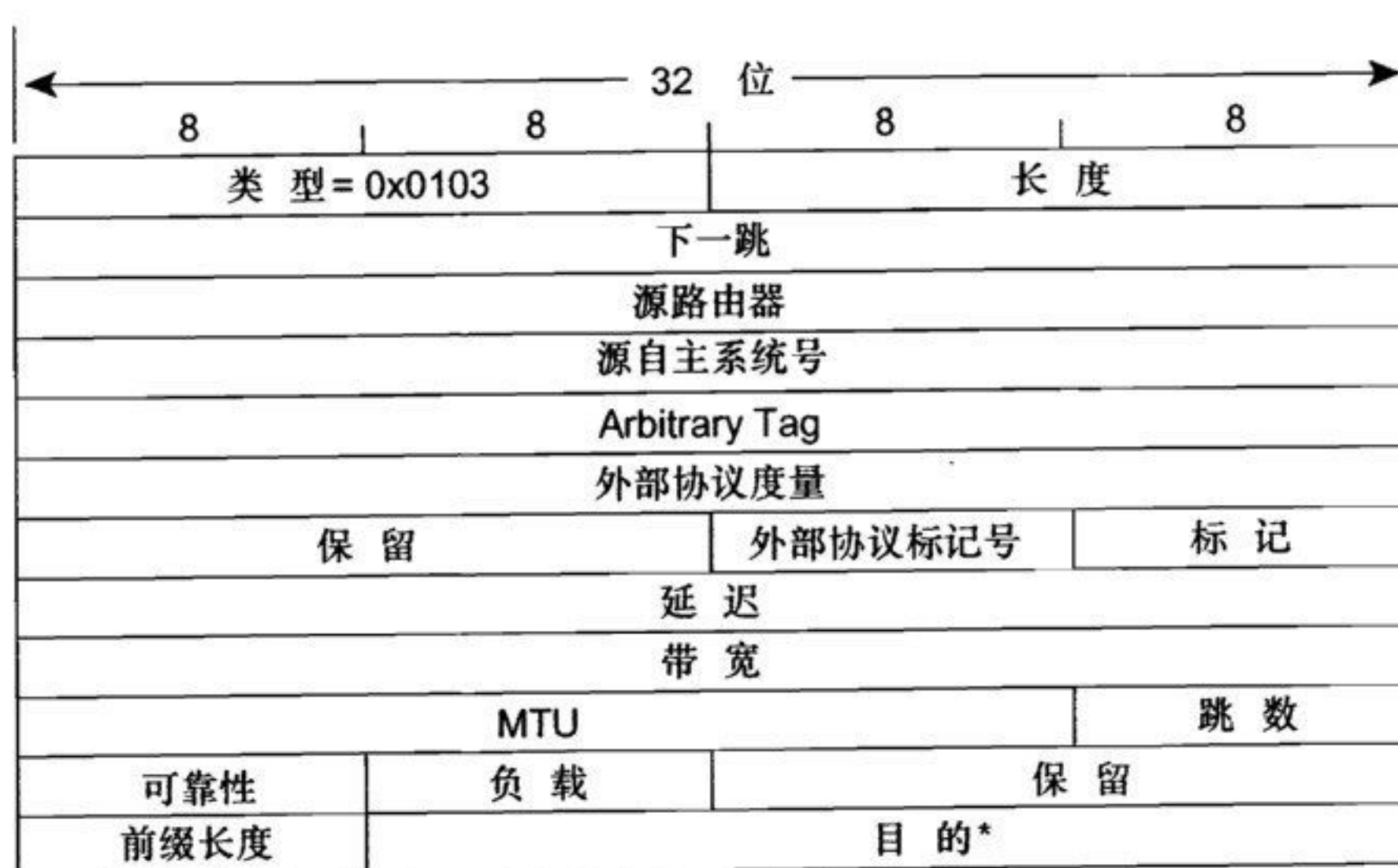


*这个字段是可变的。如果它小于或者大小 3 个 8bit 字节长度，那么将会使用 0 来填充 TLV，以使它达到下一个 4 个 8bit 字节的边界。例如，假设目的地址是 10.1，那么目的字段将是 2 个 8bit 字节和一个 0x00 的填充项。如果目的地址是 192.168.16.64，那么目的字段将是 4 个 8bit 字节和一个 0x000000 的填充。

图 8-30 IP 内部路由 TLV

- **下一跳 (Next Hop)**——是指下一跳 IP 地址。这个地址可能是、也可能不是始发路由器的地址；
- **延迟 (Delay)**——是指所配置的以 10μs 为单位表示的延迟总和。注意，不像 IGRP 报文中 24 位的字段，这个字段是 32 位的。这个更大的字段可以容纳 EIGRP 使用的大 256 倍的延迟。一个 0xFFFFFFFF 的延迟标识一个不可到达的路由。
- **带宽 (Bandwidth)**——就是 $256 \times BW_{IGRP(\min)}$ ，或者用 2 560 000 000 除以沿着路由路径方向的所有接口所配置的最小带宽。像延迟一样，这个字段也比 IGRP 的带宽字段多 8 位。
- **MTU**——是指沿着到达目的地的路由路径上所有链路中最小的最大传输单元。虽然 EIGRP 报文中包含了这个参数，但是它从来没有在度量值的计算中使用过。
- **跳数 (Hop Count)**——是一个在 0x01~0xFF 之间的数字，表示到达目的地的路由的跳数。路由器将通告和它直连的网络的跳数为 0 跳，后续的路由器将记录并通告相对于下一跳路由器的路由。
- **可靠性 (Reliability)**——是一个在 0x01~0xFF 之间的数字，用来反映沿着到达目的地的路由路径上接口的出站误码率的总和，每 5min 通过一个指数的加权平均来计算。0xFF 表示 100% 的可靠链路。
- **负载 (Load)**——是一个在 0x01~0xFF 之间的数字，用来反映沿着到达目的地的路由路径上接口的出站负载的总和，每 5min 通过一个指数的加权平均来计算。0x01 表示一条最小负载的链路。
- **保留字段 (Reserved)**——保留位，未使用的字段并且总是设置为 0x0000。
- **前缀长度 (Prefix Length)**——指出一个地址掩码中的网络位的个数。
- **目的地址 (Destination)**——表示一个路由的目的地址。虽然在图 8-30 和图 8-31 中显示的字段只是一个 3 个 8bit 字节长的字段，但是这个地段针对不同的地址是可变的。例如，假如有一条到达目的地址 10.1.0.0/16 的路由，它的前缀长度是 16，因而目的地址只需要一个包含 10.1 的 2 个 8bit 字节的字段。假如一条到达目的地址 192.168.17.64/27 的路由，它的前缀长度是 27，因而目的地址将需要一个包含

192.168.17.64 的 4 个 8bit 字节的字段。如果这个字段没有 3 个 8bit 字节的长度, 那么 TLV 将增加 0 来填充这个字段, 以便使这个字段达到 4 个 8bit 字节的边界长。



*这个字段是可变的。如果它小于或者大于 3 个 8bit 字节长度, 那么将会使用 0 来填充 TLV, 以使它达到下一个 4 个 8bit 字节的边界。例如, 假设目的地址是 10.1, 那么目的字段将是 2 个 8bit 字节和一个 0x00 的填充项。如果目的地址是 192.168.16.64, 那么目的字段将是 4 个 8bit 字节和一个 0x000000 的填充。

图 8-31 IP 外部路由的 TLV

(2) IP 外部路由的 TLV

外部路由是指到达 EIGRP 自主系统外部的目的地址的一条路径, 或者是一条通过路由重新分配注入到 EIGRP 域内的路由路径。图 8-31 显示了外部路由 TLV 字段的格式。

- **下一跳 (Next Hop)**——就是路由的下一跳 IP 地址。在一个多路访问的网络中, 正在通告路由的路由器可能不是到达目的地的最佳的下一跳路由器。例如, 一个在以太网链路上宣告 EIGRP 路由的路由器也可能宣告 BGP 的路由, 同时也可能把从 BGP 学到的路由通告到 EIGRP 的自主系统。因为以太网链路上的其他路由器并不宣告 BGP 路由, 因此, 它们也无法得知 BGP 宣告者的接口是一个最佳的下一跳地址。下一跳字段允许同时宣告两种路由协议的路由器告诉它的 EIGRP 邻居——“使用地址 A.B.C.D 代替我的接口地址作为它的下一跳”。
- **源路由器 (Originating Router)**——是一个 IP 地址, 或者重分配外部路由到 EIGRP 自主系统的路由器 ID。
- **源自主系统号 (Originating Autonomous System Number)**——是指始发路由的路由器所在的自主系统号。
- **Arbitrary Tag**——可以用来携带一组路由图的标记。如要了解路由图的使用, 请参见第 14 章“路由图”。
- **外部协议度量 (External Protocol Metric)**——顾名思义, 这是一个外部协议的度量。在和 IGRP 协议之间进行重分配时, 这个字段用来跟踪 IGRP 协议的度量值。
- **保留字段 (Reserved)**——保留位, 未使用的字段并且总是设置为 0x0000。

- 外部协议标识号 (**External Protocol ID**) ——用来标识外部路由是从哪一个协议学习到的。表 8-4 列出了这个字段的可能值。

表 8-4

外部协议标识号字段的数值

代 码	外 部 协 议
0x01	IGRP
0x02	EIGRP
0x03	静态路由
0x04	RIP
0x05	Hello
0x06	OSPF
0x07	IS-IS
0x08	EGP
0x09	BGP
0x0A	IDRP
0x0B	直连链路

- 标记 (**Flags**) ——目前仅定了两个标记。如果这个八位字段最右边的第一位设置了 (0x01)，该路由就是外部路由。如果右边的第二位设置了 (0x02)，该路由就是一个候选的缺省路由。缺省路由的讲述请参见第 12 章“缺省路由和按需路由选择”。

其余的字段描述了度量和目的地址。这些字段的含义和在内部路由 TLV 中讲述的相同字段的含义是相同的。

8.1.6 地址聚合

在第 2 章“TCP/IP 回顾”中介绍了子网划分的操作方法——为了使多条链路可以使用一个主网络地址编址，将地址掩码扩展到了主机地址空间。第 7 章介绍了可变子网掩码的操作方法——地址掩码的使用扩展到了子网中，甚至在子网中又创建了更多的新子网。

从相反的观点来看，子网地址也可以考虑成为一组更小的子网的汇总，而一个主网络地址可以看作一组子网的汇总。在每个这样的实例中，汇总都是通过减少子网掩码的长度来完成的。

地址聚合是打破主网络地址分类限制的进一步汇总措施。聚合的地址表示了一组数字上连续的网络地址，或称为超网 (supernet)。¹图 8-32 中显示了一个聚合地址的例子。

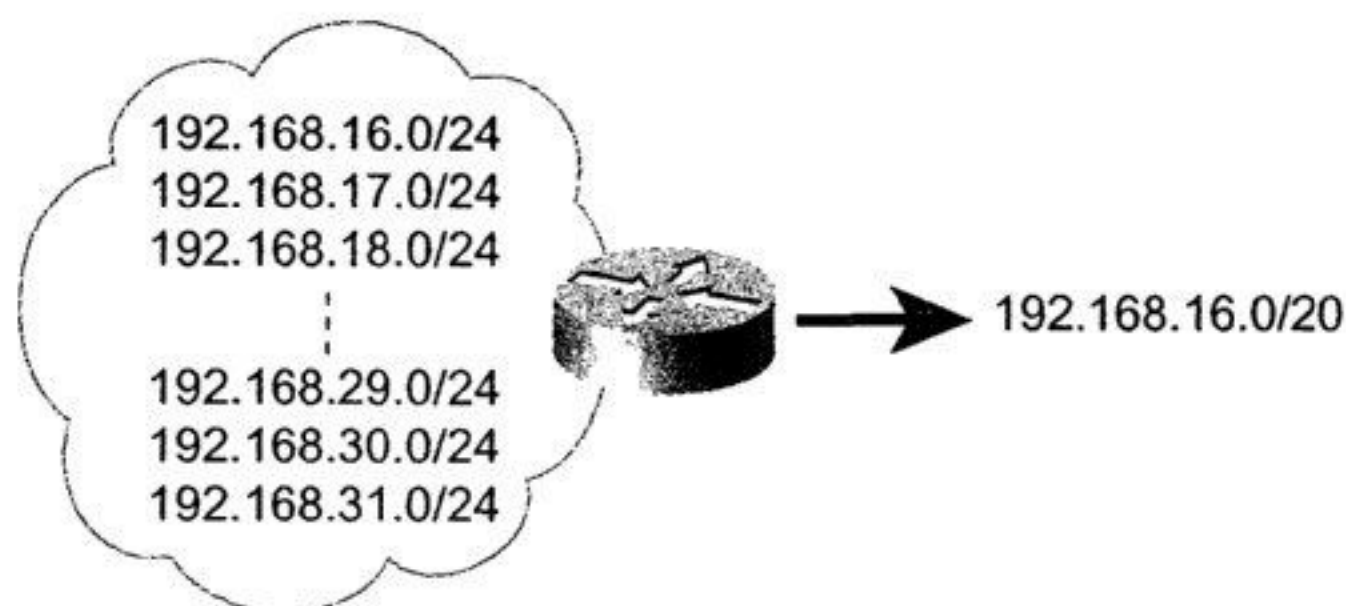


图 8-32 这组网络地址可以看作是单个的聚合地址或聚合子网

图 8-33 显示了是怎样得出图 8-32 中的聚合地址的。对于一组网络地址，寻找出所有网

¹ 更正确地说，聚合应该是任何一组地址的汇总。这里声明，本书中所提及的聚合地址是指一组主网络地址的汇总。

络地址的共同位并对这些位使用掩码。被掩码覆盖的部分就是聚合地址。

```

111111111111111111111111111100000000 = 24位掩码
11000000101010000001000000000000 = 192.168.16.0/24
11000000101010000001000100000000 = 192.168.17.0/24
11000000101010000001001000000000 = 192.168.18.0/24
11000000101010000001001100000000 = 192.168.19.0/24
11000000101010000001010000000000 = 192.168.20.0/24
11000000101010000001010100000000 = 192.168.21.0/24
11000000101010000001011000000000 = 192.168.22.0/24
11000000101010000001011100000000 = 192.168.23.0/24
11000000101010000001100000000000 = 192.168.24.0/24
11000000101010000001100100000000 = 192.168.25.0/24
11000000101010000001101000000000 = 192.168.26.0/24
11000000101010000001101100000000 = 192.168.27.0/24
11000000101010000001110000000000 = 192.168.28.0/24
11000000101010000001110100000000 = 192.168.29.0/24
11000000101010000001111000000000 = 192.168.30.0/24
11000000101010000001111100000000 = 192.168.31.0/24
11000000101010000001000000000000 = 192.168.16.0/20

```

图 8-33 聚合地址是由对一组数字上连续的网络地址的所有共同位进行掩码而得出的

当设计一个超网时，有一点很重要，就是超网的成员地址应该由原来掩码位的一个完整和连续的地址集合组成。例如，在图 8-33 中，聚合地址的 20 位掩码比成员地址的掩码少 4 位。对于这 4 个“不同”的位，注意，它们包括了 0000~FFFF 之间二进制位组合的每一种可能性。按照这个设计规则如果失败的话，将会引起编址的安排冲突，减小聚合路由的性能，并且可能导致路由选择环路和路由选择“黑洞”。

汇总寻址的一个明显的好处就是对网络资源的节省。由于通告更少的路由从而节省了带宽，而处理更少的路由则节省了 CPU 的周期。更为重要的是，由于路由选择表的大小缩减而使内存的使用也变得节省了。

无类别路由选择、VLSM 和聚合寻址一起都是通过创建层次化的地址来达到最大限度地节省网络资源的目的。与 IGRP 协议不同，EIGRP 协议支持所有这些寻址策略。在图 8-34 中，Treetop Aviation 的工程部门分配了 16 个 C 类的地址，这些地址已经根据需要分配到不同的子部门中去了。

发动机、电力和水力部门的聚合地址是它们自己聚合到单个地址 192.168.16.0/21 中去的。这个地址和机身部门的聚合地址一起被聚合到一个单一的地址 192.168.16.0/20 中，这个地址也表示了整个工程部门的地址。

其他的部门也可以类似地表示。例如，假设 Treetop Aviation 总共有 8 个部门，而每个部门都和工程部门有相似的地址分配，那么在最高层次的骨干路由器只有很少的 8 条路由，如图 8-35 所示。

层次化的地址设计将继续应用于每个部门的每一个子部门中，通过子网化划分成更小的单独的网络地址，VLSM 也可以用来进一步分割子网。路由选择协议将在网络的边界上自动地汇总子网，这和前面章节的讲述是一致的。

在 Internet 上，地址聚合也允许地址的节省和地址的分层。对于指数速度增长的 Internet，有两个越来越需要关注的是：可供使用的 IP 地址（尤其是 B 类地址）的消耗和存储 Internet 路由选择信息所需要的巨大的数据库。

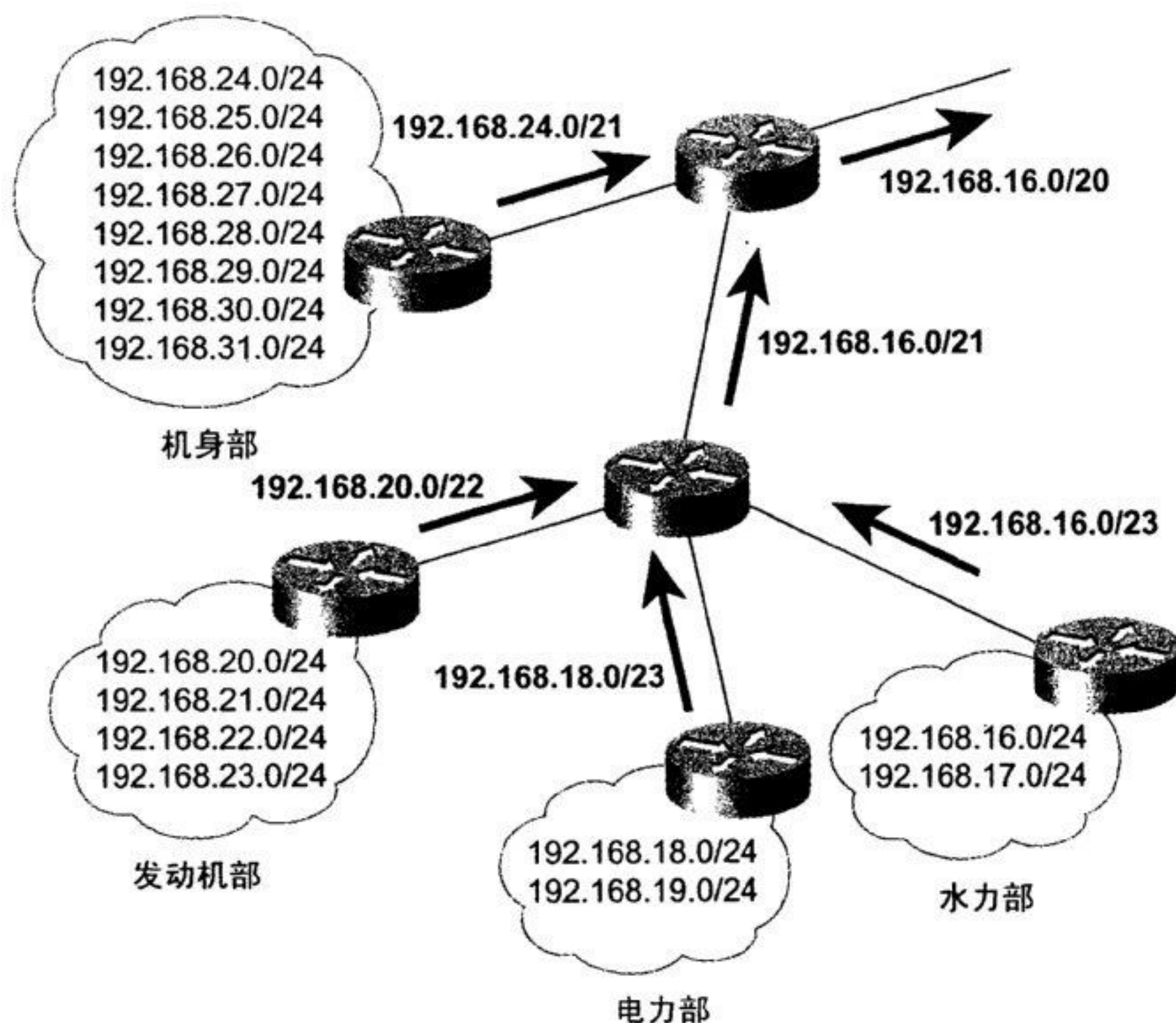


图 8-34 在 Treetop Aviation 中, 一个更大的部门中的几个子部门正在聚合地址。依次地, 整个部门将通过单个聚合地址 192.168.16.0/20 通告出去

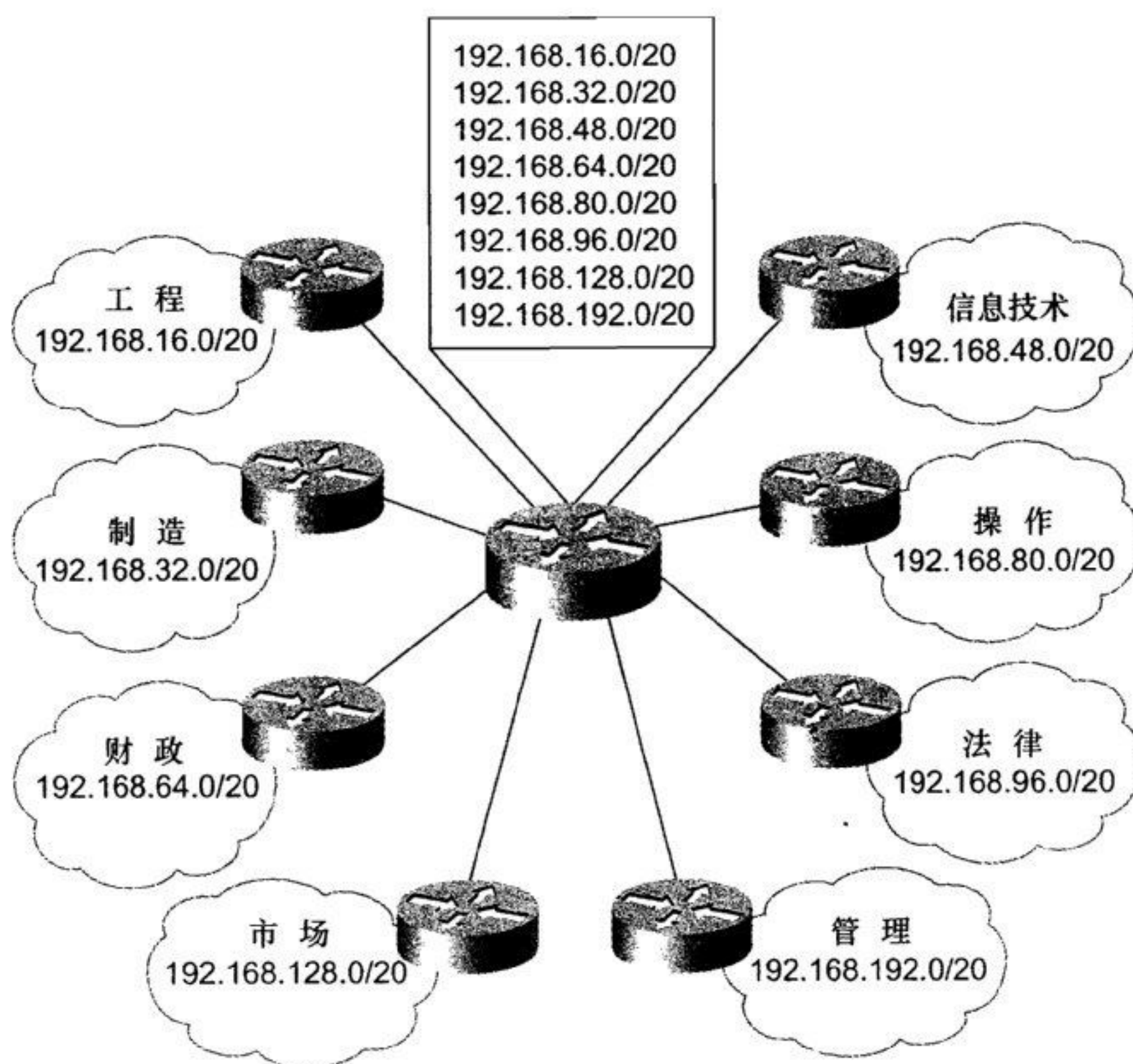


图 8-35 虽然在这个互联网络中有 128 个主网络地址并且可能覆盖了 32 000 台主机, 但是骨干路由器的路由选择表中只有 8 条聚合地址

这个问题的一个解决方案就是使用称为无类别域间路由选择 (CIDR) 的方法。¹在 CIDR 下, C 类地址的集合被 InterNIC 机构分配给不同的国际上的地址分配权威机构, 像美国的 Network Solutions 和欧洲的 R seaux IP Europ ens (RIPE) 机构。这些地址集合是按照地域来组织的, 如表 8-5 所示。

表 8-5 CIDR 地址按国际上的地理区域分配

地 区	地 址 范 围
Multiregional	192.0.0.0-193.255.255.255
欧洲	194.0.0.0-195.255.255.255
其他	196.0.0.0-197.255.255.255
北美	198.0.0.0-199.255.255.255
中、南美	200.0.0.0-201.255.255.255
环太平洋地区 (Pacific Rim)	202.0.0.0-203.255.255.255
其他	204.0.0.0-205.255.255.255
其他	206.0.0.0-207.255.255.255

这些地址分配权威机构轮流地分配它们自己的那部分地址给本地的互联网络服务提供商 (ISP)。当一个组织申请 IP 地址并且所需的地址小于 32 个子网和 4096 台主机时, 将可以分配给它一组连续的 C 类地址, 称为 CIDR 块 (CIDR block)。

以这种方式, 单个组织的 Internet 路由器可以通告单一的汇总地址给它们的 ISP。反过来, ISP 也可以聚合它自己所有的地址, 而世界某个区域内的所有 ISP 的地址就可以汇总到表 8-5 中所表示的地址中去。

本章的案例研究包含了一些地址聚合的例子, 更深一步的例子将在第 9 章中讲解。

8.2 配置 EIGRP

EIGRP 协议的基本配置和 IGRP 协议的基本配置十分相似, 因此有时一些讲师就这么来教授初学者: “按照 IGRP 配置, 但是增加一个 E 字”。正如前面章节所提及的, 命令 **metric weights** 在 EIGRP 协议和 IGRP 协议中的使用方法是一样的, 命令 **traffic-share** 和命令 **variance** 在这两种协议之间的使用也是相同的。如果需要复习这些命令, 请参考第 6 章。

本节的案例研究演示了一个 EIGRP 协议的基本配置, 然后讲述了路由汇总的技巧和与 IGRP 协议之间的互操作性。

8.2.1 案例研究 1: 一个基本的 EIGRP 配置

和 IGRP 协议一样, EIGRP 也仅仅需要两个步骤就可以启动一个 EIGRP 的路由选择进程:

步骤 1: 使用 **router eigrp process-id** 命令启动 EIGRP 进程;

步骤 2: 使用 **network** 命令来指定运行 EIGRP 协议的每个主网络。

¹ V. Fuller, T. Li, J. I. Yu, and K. Varadhan. "Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy." RFC 1519, 1993 年 9 月。

EIGRP 进程 ID 号可以是 1~65535 (0 不允许使用) 之间的任何一个数字, 只要在必须共享路由信息的所有路由器上的 EIGRP 进程 ID 号是一致的, 那么网络管理员可以随意地选用进程 ID 号。另外一个选择, 这个进程号也可以使用 Inter NIC 分配的自主系统号。图 8-36 显示了一个简单的互连网络, 图中 3 台路由器的配置如下:

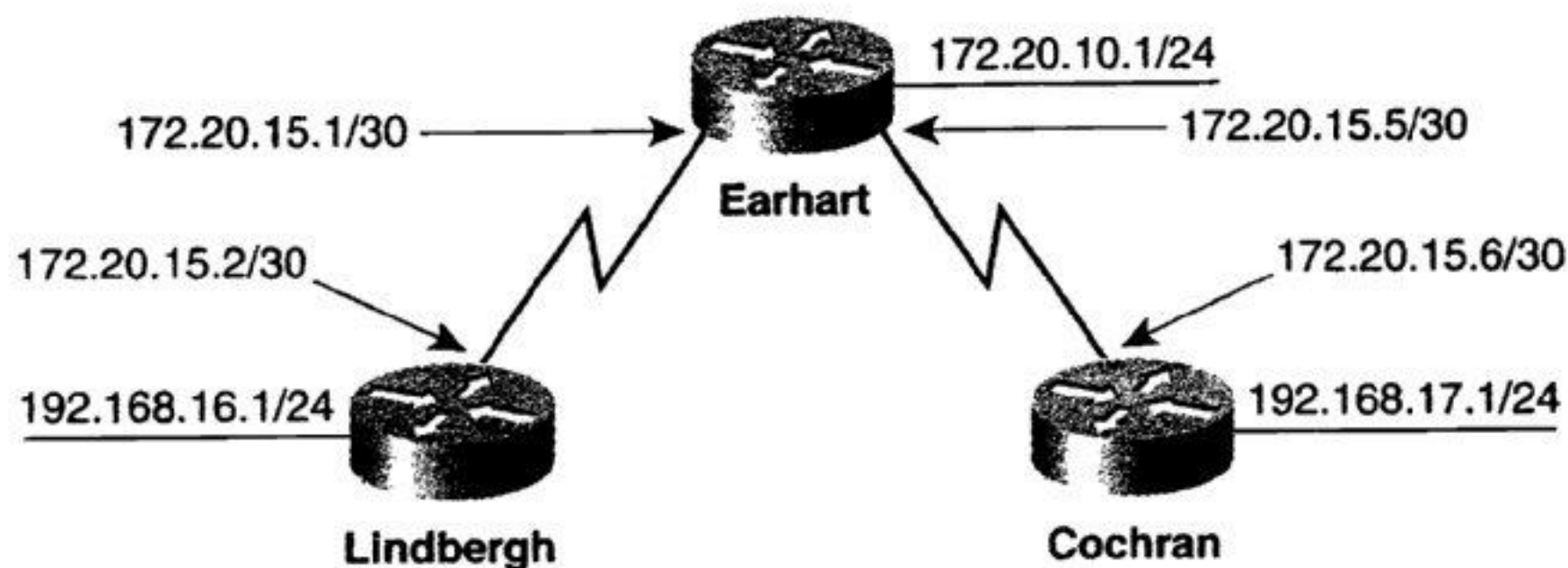


图 8-36 与 IGRP 协议不同, EIGRP 协议支持这个互连网络的 VLSM 需求

路由器 Earhart:

```
router eigrp 15
network 172.20.0.0
```

路由器 Cochran:

```
router eigrp 15
network 172.20.0.0
network 192.168.17.0
```

路由器 Lindbergh:

```
router eigrp 15
network 172.20.0.0
network 192.168.16.0
```

路由器 Earhart 的路由选择表如图 8-37 所示。这个路由选择表显示了 EIGRP 协议缺省的管理距离是 90, 并且表中的网络 172.20.0.0 被划分为不同的子网。

```
Earhart#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

D    192.168.16.0/24 [90/2195456] via 172.20.15.2, 00:02:06, Serial0
D    192.168.17.0/24 [90/2195456] via 172.20.15.6, 00:02:06, Serial1
D    172.20.0.0/16 is variably subnetted, 3 subnets, 2 masks
C    172.20.10.0/24 is directly connected, Ethernet0
C    172.20.15.4/30 is directly connected, Serial1
C    172.20.15.0/30 is directly connected, Serial0
Earhart#
```

图 8-37 路由器 Earhart 的路由选择表

与本章前面的一些例子不同,图 8-36 中的互联网络使用了缺省度量,因此在实际的环境中复习一下 EIGRP 的度量计算是很有用的。

跟踪从路由器 Earhart 到达网络 192.168.16.0 的路由,这条路由的路径穿过了一个串行接口和一个以太网接口,每个接口都配置为它们的缺省度量值。EIGRP 度量的计算与第 6 章中讲述的 IGRP 协议的度量计算一样,只是 EIGRP 的度量需要在 IGRP 度量计算的最后结果上乘以一个 256 的倍数。这条路由路径的最小带宽是串行接口上的带宽,¹延迟是这两个接口延迟的总和。请参考表 6-1:

$$BW_{EIGRP(min)} = 256 \times 6476 = 1657856$$

$$DLY_{EIGRP(sum)} = 256 \times (2000 + 100) = 537600$$

因此,

$$Metric = 1657856 + 537600 = 2195456$$

8.2.2 案例研究 2: 和 IGRP 的重新分配

路由选择协议之间的重新分配将在第 11 章中讲解,但这里值得注意的是,如果一个 IGRP 协议和一个 EIGRP 协议进程有一个相同的进程 ID 号,那么它们将自动地进行路由重新分配。在图 8-38 中,路由器 Curtiss 的配置如下:

```
router igrp 15
 network 172.25.0.0
 network 172.20.0.0
```

路由器 Earhart 的配置如下:

```
router eigrp 15
 passive-interface Ethernet0
 network 172.20.0.0
!
router igrp 15
 passive-interface Serial0
 passive-interface Serial1
 network 172.20.0.0
```

路由器 Earhart 宣告 IGRP 协议信息到路由器 Curtiss,同时宣告 EIGRP 协议信息到路由器 Lindbergh 和 Cochran。注意,路由器 Earhart 的接口地址都在网络 172.20.0.0 的地址范围内,因此要使用命令 **passive-interface** 来限制不必要的路由选择协议的报文通信量。对于 EIGRP,这个命令只需要阻断不必要的 Hello 报文。如果在一个接口上没有发现邻居,那么也就不会发送其他的 EIGRP 报文。

图 8-39 中显示了路由器 Curtiss 的路由选择表。这里注意,路由选择表中不仅仅显示了到达网络 192.168.16.0 和 192.168.17.0 的路由,而且还显示了路由重新分配后经过调整的度量值,也就是除去了 EIGRP 的倍数因子 256。相反地,如果从 IGRP 协议重新分配到 EIGRP 协议中,路由的度量值则需要乘以一个倍数因子 256。

¹ 记住串行接口的缺省带宽是 1544kb/s。

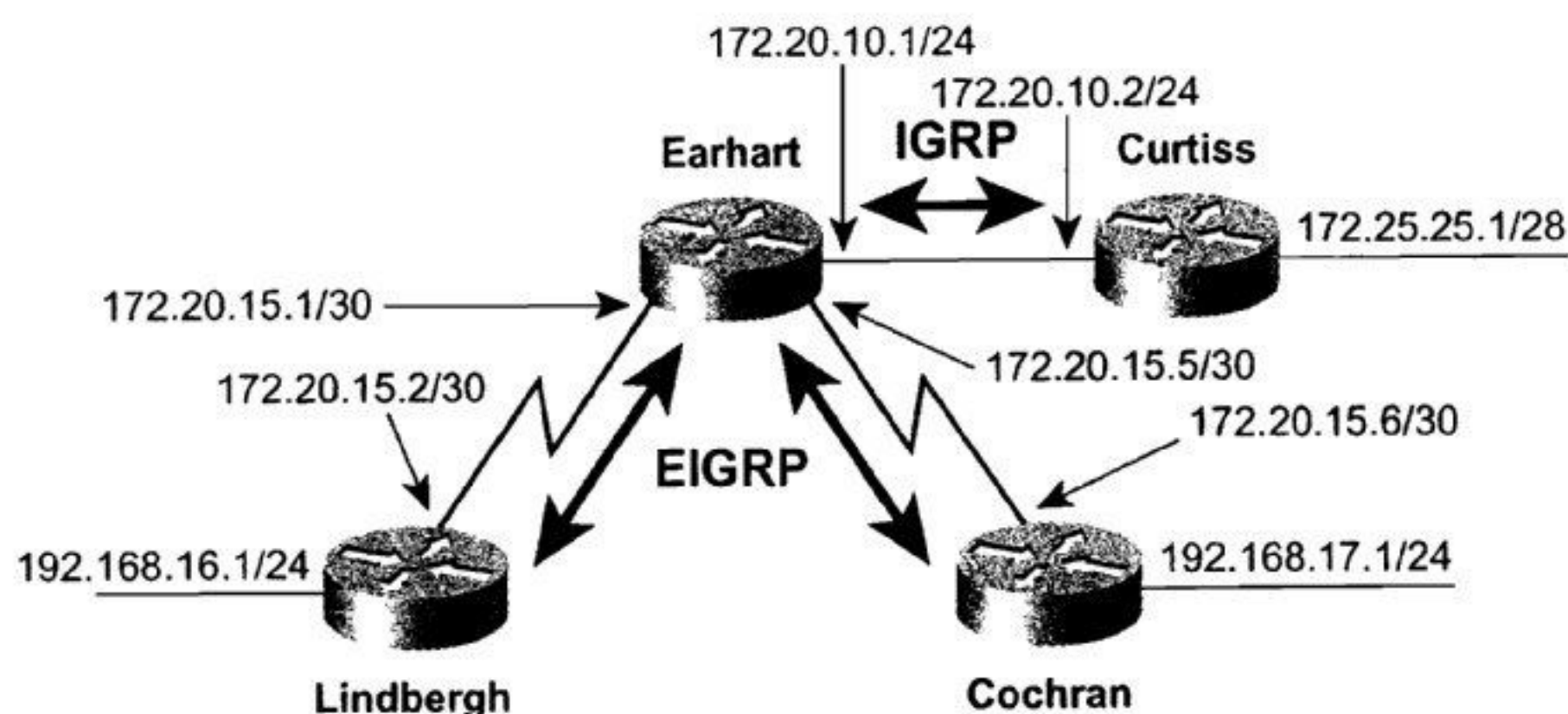


图 8-38 如果路由器 Earhart 同时配置了 IGRP 和 EIGRP 协议, 并使用了相同的进程 ID 号, 那么路由信息将进行路由重新分配

```
Curtiss#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

I    192.168.16.0/24 [100/8676] via 172.20.10.1, 00:00:06, Ethernet0
I    192.168.17.0/24 [100/8676] via 172.20.10.1, 00:00:06, Ethernet0
     172.25.0.0/28 is subnetted, 1 subnets
C       172.25.25.0 is directly connected, Ethernet1
     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.10.0 is directly connected, Ethernet0
Curtiss#
```

图 8-39 在路由器 Earhart 上增加了 IGRP 协议进程后, 路由器 Curtiss 的路由选择表

图 8-39 中也显示一些信息丢失了。在路由器 Earhart 上, 有类别的 IGRP 进程不接收到达子网 172.20.15.0/30 和子网 172.20.15.4/30 的可变长子网路由。使用命令 **ip summary-address eigrp**, 可以配置路由器 Earhart 发送一个汇总的路由通告给路由器 Curtiss:

```
interface Ethernet0
 ip address 172.20.10.1 255.255.255.0
 ip summary-address eigrp 15 172.20.15.0 255.255.255.0
!
router eigrp 15
 passive-interface Ethernet0
 network 172.20.0.0
!
router igrp 15
 passive-interface Serial0
 passive-interface Serial1
 network 172.20.0.0
```


路由器 Curtiss 的 IGRP 进程将可以接收到 EIGRP 的汇总通告, 图 8-40 中的路由选择表显示了这个结果。

```
Curtiss#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

I    192.168.16.0/24 [100/10676] via 172.20.10.1, 00:00:18, Ethernet0
I    192.168.17.0/24 [100/8676] via 172.20.10.1, 00:00:18, Ethernet0
     172.25.0.0/28 is subnetted, 1 subnets
C      172.25.25.0 is directly connected, Loopback0
     172.20.0.0/24 is subnetted, 2 subnets
C      172.20.10.0 is directly connected, Ethernet0
I      172.20.15.0 [100/8576] via 172.20.10.1, 00:00:18, Ethernet0
Curtiss#
```

图 8-40 在配置了路由器 Earhart 发送一个汇总的路由后, 路由器 Curtiss 现在就能够到达这两条串行链路了

图 8-41 中显示了路由器 Cochran 带有重分配的 IGRP 路由的路由选择表。正如这个路由选择表所显示的, EIGRP 明确地标记了从外部学习到的路由, 这个信息在阅读路由选择表时有一定的帮助, 因为这样很容易辨认通过路由重新分配学到的外部路由。

```
Cochran#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
       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

Gateway of last resort is not set

D    192.168.16.0/24 [90/3219456] via 172.20.15.5, 00:41:41, Serial0
C    192.168.17.0/24 is directly connected, Ethernet0
     192.168.18.0/24 is variably subnetted, 2 subnets, 2 masks
D EX 172.25.0.0/16 [170/2221056] via 172.20.15.5, 00:41:48, Serial0
     172.20.0.0/16 is variably subnetted, 3 subnets, 2 masks
D    172.20.10.0/24 [90/2195456] via 172.20.15.5, 00:41:48, Serial0
C    172.20.15.4/30 is directly connected, Serial0
D    172.20.15.0/30 [90/2681856] via 172.20.15.5, 00:41:48, Serial0
D    172.20.0.0/16 is a summary, 00:00:09, Null0
```

图 8-41 正如路由器 Cochran 中的路由选择表所显示的, EIGRP 标记了所学到的外部路由

图 8-41 中路由选择表的最后一个条目也很有趣, 其中有一条汇总路由指向了一个空的接口 (NULL)。这条路由可以帮助我们在使用缺省路由和汇总路由时, 防止潜在的路由选择黑洞。这个技巧将在第 11 章和第 12 章中讲述。

8.2.3 案例研究 3: 关闭自动路由汇总

缺省条件下, EIGRP 协议在网络边界和前面章节所讲述的协议一样进行路由汇总。但是

和前面那些协议不同, EIGRP 的自动路由汇总可以被关闭。图 8-42 中显示了一种情形, 说明关闭汇总功能是有用的。

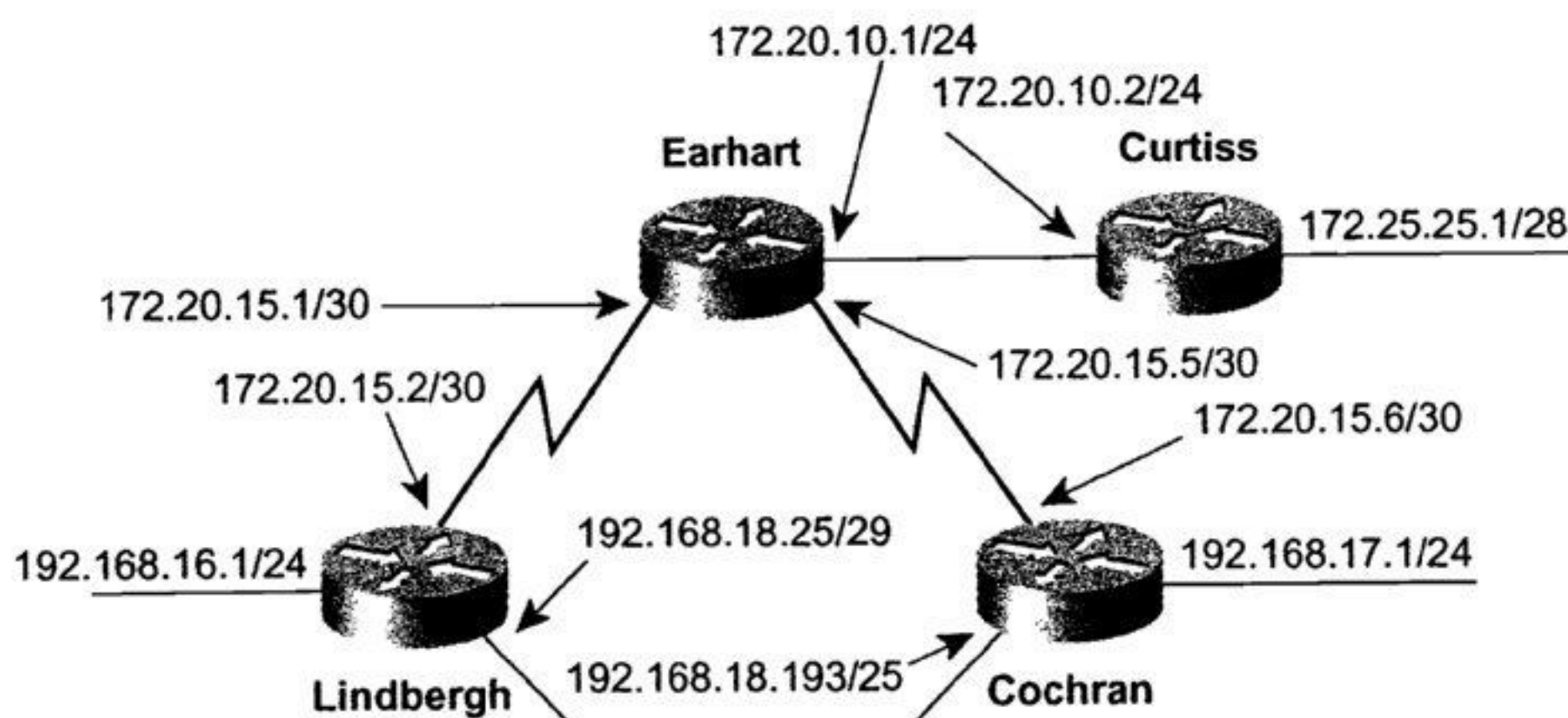


图 8-42 在路由器 Cochran 和 Lindbergh 上关闭自动路由汇总功能来防止到达网络 192.168.18.0 的不确定的路由

在路由器 Cochran 和路由器 Lindbergh 上分别添加了新的以太网链路, 并且给它们分配的地址形成了不连续的子网。这两台路由器的缺省行为是把它们自己当作主网络 192.168.18.0 和 172.20.0.0 之间的边界路由器。结果, 路由器 Earhart 将在它的两个串行接口上收到到达网络 192.168.18.0 的汇总路由通告。这个结果会产生一个路由选择不明确的情况: 路由器 Earhart 记录了到达网络 192.168.18.0 的两条等价路径, 要到达那些以太子网之一的数据包可能会、也可能不会被路由到正确的链路上去。

使用命令 **no auto-summary** 来关闭自动汇总功能。例如, 路由器 Lindbergh 的配置将是:

```
router eigrp 15
 network 172.20.0.0
 network 192.168.16.0
 network 192.168.18.0
 no auto-summary
```

在路由器 Lindbergh 和 Cochran 上关闭了自动路由汇总功能后, 分离的子网 192.168.18.24/29 和 192.168.18.128/25 将可以被通告到网络 172.20.0.0 中去, 这样就在路由器 Earhart 上排除了不确定的路由。

8.2.4 案例研究 4: 地址聚合 (Address Aggregation)

在图 8-43 中的互联网络上增添了一个新的路由器。路由器 Earhart 必须通告给路由器 Yeager 的 5 个网络地址可以汇总成两个聚合地址。路由器 Earhart 的配置将是:

```
interface Ethernet1
 ip address 10.15.15.254 255.255.255.252
 ip summary-address eigrp 15 172.0.0.0 255.0.0.0
 ip summary-address eigrp 15 192.168.16.0 255.255.240.0
```

命令 **ip summary-address eigrp** 将会自动抑制更详细的网络地址通告给路由器 Yeager。图 8-44 中显示了在路由器 Earhart 上配置了地址聚合前后路由器 Yeager 的路由选择表。即使

在这么一个小小的互联网络中, EIGRP 学习到的路由条目也减少了一半。在一个大型的互联网络中, 路由选择表和存储路由选择表所需的内存的缩减将变得很有意义。

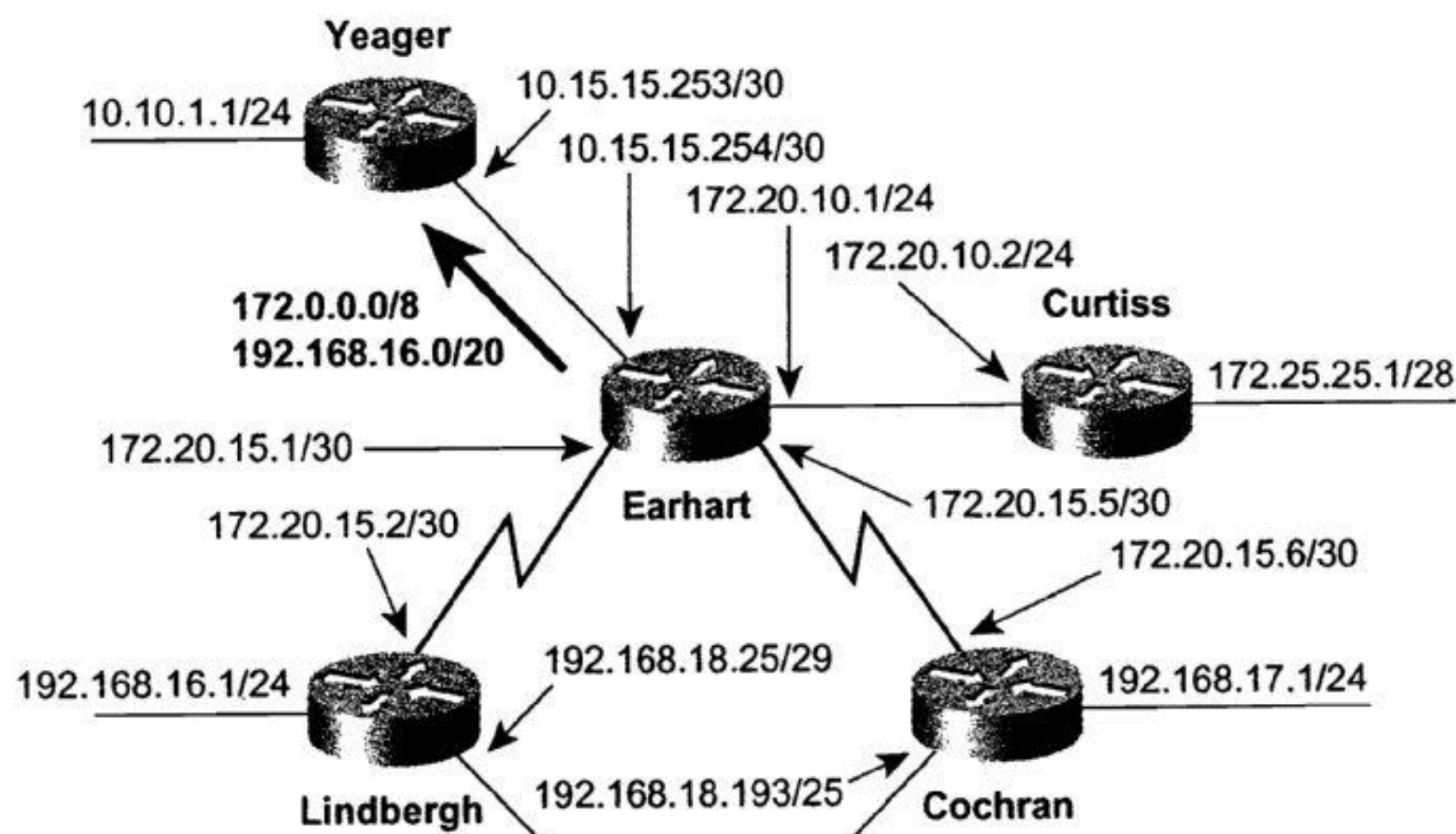


图 8-43 路由器 Earhart 正在通告两个聚合地址给路由器 Yeager

```
Yeager#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
       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
```

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.10.1.0/24 is directly connected, Ethernet1
C    10.15.15.252/30 is directly connected, Ethernet0
D    192.168.16.0/24 [90/2733056] via 10.15.15.254, 00:00:13, Ethernet0
D    192.168.17.0/24 [90/2221056] via 10.15.15.254, 00:00:13, Ethernet0
192.168.18.0/24 is variably subnetted, 2 subnets, 2 masks
D    192.168.18.24/29 [90/2221056] via 10.15.15.254, 00:00:13, Ethernet0
D    192.168.18.128/25 [90/2323456] via 10.15.15.254, 00:00:13, Ethernet0
D EX 172.25.0.0/16 [170/332800] via 10.15.15.254, 00:00:13, Ethernet0
D    172.20.0.0/16 [90/307200] via 10.15.15.254, 00:00:14, Ethernet0
```

```
Yeager#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
       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
```

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.10.1.0/24 is directly connected, Ethernet1
C    10.15.15.252/30 is directly connected, Ethernet0
D    192.168.16.0/20 [90/435200] via 10.15.15.254, 00:00:26, Ethernet0
D    172.0.0.0/8 [90/307200] via 10.15.15.254, 00:00:26, Ethernet0
```

图 8-44 在路由器 Earhart 上配置了地址聚合前后路由器 Yeager 的路由选择表

8.2.5 案例研究 5: 认证

EIGRP 协议报文的认证是在 IOS 软件的 11.3 版和后续的版本中才支持的。MD5 加密校验和是 EIGRP 协议唯一支持的认证方式, 初看起来, 这和 RIPv2 或 OSPF 协议相比灵活性较差, 因为后两种协议同时支持 MD5 认证和明文口令认证。然而, 明文口令认证应该只使用在邻居设备不支持比较安全的 MD5 认证的时候。由于 EIGRP 协议仅会在两台 Cisco 公司的设备之间互相宣告, 因此这种情况不会发生。

配置 EIGRP 协议的认证有以下几个步骤:

步骤 1: 定义一个带有名字的钥匙链;

步骤 2: 在钥匙链上定义一个或一组钥匙;

步骤 3: 在接口上启用认证并指定使用的钥匙链;

步骤 4: 可选地配置钥匙的管理。

钥匙链的配置和管理已经在第 7 章中讲述过了。EIGRP 协议认证是在接口上配置命令 **ip authentication key-chain eigrp** 和命令 **ip authentication mode eigrp md5** 来启用和连接到一个钥匙链上的。¹

参考图 8-43, 在路由器 Cochran 到路由器 Earhart 的接口上启动 EIGRP 认证的配置如下:
路由器 Cochran:

```
key chain Edwards
  key 1
    key-string PanchoBarnes
  !
interface Serial0
  ip address 172.20.15.6 255.255.255.252
  ip authentication key-chain eigrp 15 Edwards
  ip authentication mode eigrp 15 md5
```

在路由器 Earhart 上也要作相似的配置。关于钥匙链的管理命令 **accept-lifetime** 和 **send-lifetime** 的使用方法已经在第 7 章中讲述过了。

8.3 EIGRP 故障排除

IGRP 协议或 RIP 协议的路由信息交换的故障排除是一个相当简单的过程。路由选择更新要么传播出去了要么没有传播出去, 要么包含了精确的路由信息要么没有包含。EIGRP 协议复杂性的增加也意味着故障排除的过程增加了复杂性。在 EIGRP 协议的故障排除中, 必须验证邻居表和邻接关系的正确性, 接着要检查 DUAL 算法的查询/响应过程的正确性, 还必须考虑在自动汇总的地方对 VLSM 的影响。

本节的案例研究讲述了一系列典型的事件, 可以用来追踪一个 EIGRP 的故障。随后的一

¹ 虽然 MD5 认证是 EIGRP 协议目前惟一可用的认证模式, 但是使用命令 **ip authentication mode eigrp md5** 可以为将来出现其他可用的认证模式作预先考虑。

个案例研究将讲述在一个大型的 EIGRP 互连网络中引起网络不稳定的一些偶然原因。

8.3.1 案例研究 6: 邻居丢失 (A Missing Neighbor)

图 8-45 中显示了一个小型的 EIGRP 互连网络。用户抱怨子网 192.168.16.224/28 无法到达。仔细察看路由选择表, 发现在路由器 Grissom 上有一些错误, 如图 8-46 所示。¹

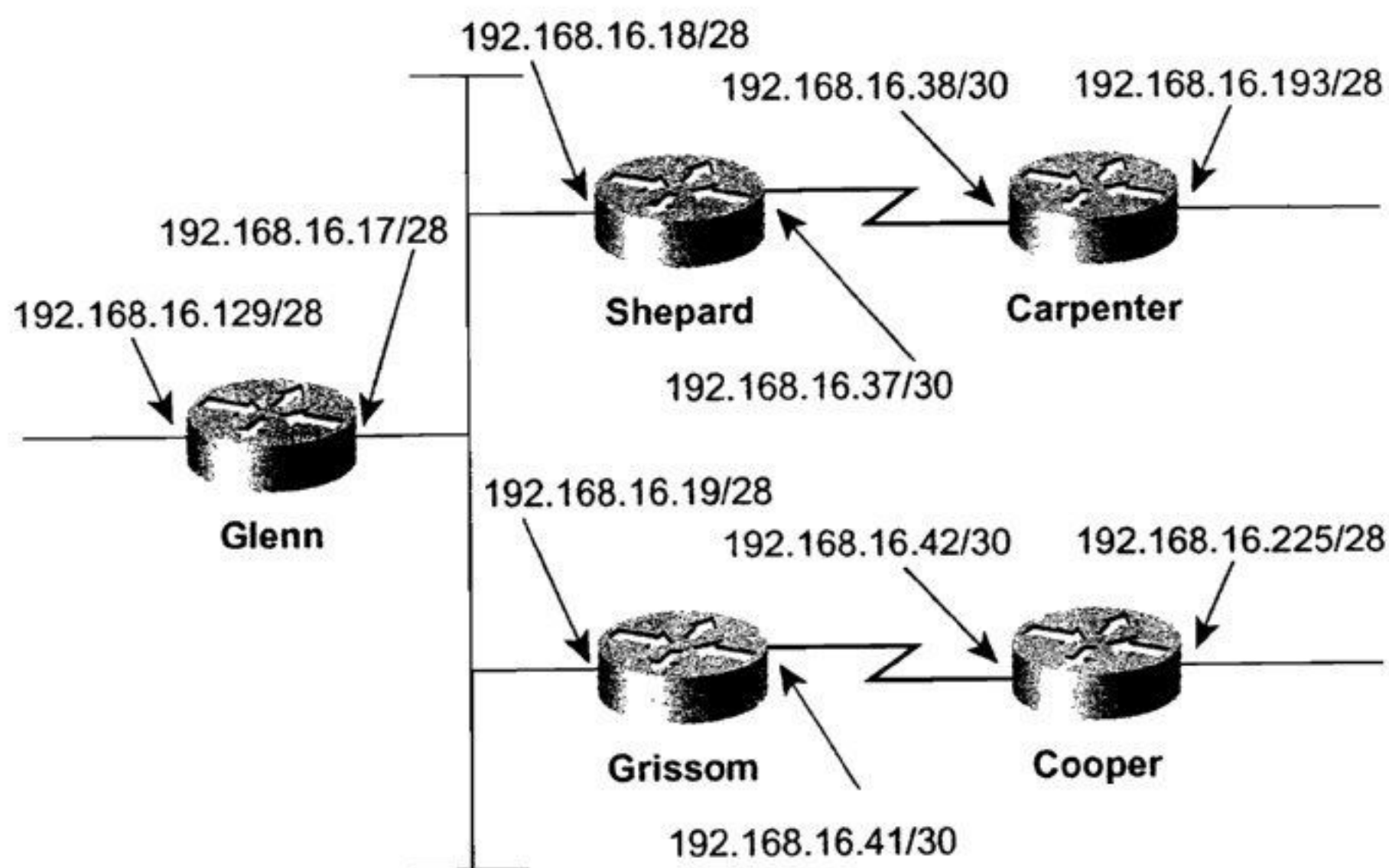


图 8-45 在这个例子的 EIGRP 互连网络中, 通过路由器 Grissom 不能到达子网 192.168.16.224/28

从图 8-46 的两个路由选择表中可以得到以下的信息:

- 路由器 Shepard 的路由选择表中没有子网 192.168.16.40/30 和子网 192.168.16.224/28 的信息, 虽然路由器 Grissom 的路由选择表中含有这些信息;
- 路由器 Grissom 的路由选择表中没有包含任何应该由路由器 Glenn 或 Shepard 通告的子网;
- 路由器 Shepard 的路由选择表中包含了由路由器 Glenn 通告的子网 (并且路由器 Glenn 的路由选择表中包含了路由器 Shepard 通告的子网, 虽然它的路由选择表没有包括在这个图表中)。

从以上的观察信息可以得出结论, 路由器 Grissom 没有正确地接收和通告关于子网 192.168.16.16/28 的路由。

在这些可能引起故障的原因中, 应该首先来察看一些最简单和直接的原因。这些原因有:

- 接口地址或者掩码配置不正确;
- EIGRP 的进程 ID 号不正确;
- **network** 语句遗漏或者不正确。

在这个实例中, 没有发现 EIGRP 或地址的配置错误。

接下来, 仔细检查一下邻居表, 分别在路由器 Grissom、Shepard 和路由器 Glenn 上察看邻居表, 如图 8-47 所示, 有两个事实比较明显:

¹ 当排除一个互连网络的故障时, 检查所有路由器接口的地址配置是否属于正确的子网是一个好习惯。


```
Grissom#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

```
192.168.16.0/24 is variably subnetted, 3 subnets, 2 masks
C    192.168.16.40/30 is directly connected, Serial0
C    192.168.16.16/28 is directly connected, Ethernet0
D    192.168.16.224/28 [90/2195456] via 192.168.16.42, 01:07:26, Serial0
```

```
Shepard#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

```
192.168.16.0/24 is variably subnetted, 4 subnets, 2 masks
C    192.168.16.36/30 is directly connected, Serial0
C    192.168.16.16/28 is directly connected, Ethernet0
D    192.168.16.192/28 [90/2297856] via 192.168.16.38, 01:07:20, Serial0
D    192.168.16.128/28 [90/307200] via 192.168.16.17, 01:07:20, Ethernet0
```

图 8-46 路由器 Shepard 和 Grissom 的路由选择表显示了路由器 Grissom 的 EIGRP 进程没有通告和接收到关于子网 192.168.16.16/28 的路由

```
Grissom#show ip eigrp neighbors
IP-EIGRP neighbors for process 75
H   Address                Interface    Hold Uptime    SRTT    RTO    Q    Seq
                               (sec)              (ms)          Cnt  Num
0   192.168.16.42           Se0         11 05:27:11    23     200    0    8
```

```
Shepard#show ip eigrp neighbors
IP-EIGRP neighbors for process 75
H   Address                Interface    Hold Uptime    SRTT    RTO    Q    Seq
                               (sec)              (ms)          Cnt  Num
1   192.168.16.19           Et0         12 00:01:01     0     5000    1    0
2   192.168.16.17           Et0         11 05:27:33     8      200    0    6
0   192.168.16.38           Se0         14 05:27:34    22      200    0   10
```

```
Glenn#show ip eigrp neighbors
IP-EIGRP neighbors for process 75
H   Address                Interface    Hold Uptime    SRTT    RTO    Q    Seq
                               (sec)              (ms)          Cnt  Num
1   192.168.16.19           Et0         14 00:00:59     0     8000    1    0
2   192.168.16.18           Et0         10 05:30:11     9       20    0    7
0   192.168.16.129          Et1         12 05:30:58     6       20    0    7
```

图 8-47 在这个例子的 EIGRP 互联网络中, 通过路由器 Grissom 不能到达子网 192.168.16.224/28

- 路由器 Grissom (192.168.16.19) 在它的邻居的邻居表中, 但是它的邻居却不在路由器 Grissom 的邻居表中;
- 整个互联网络已经运行了 5 个多小时了, 这个信息可以从除了路由器 Grissom 外的所有邻居的 uptime 统计信息反映出来。然而, 路由器 Grissom 的 uptime 显示大约是 1min。

假设路由器 Grissom 在路由器 Shepard 的邻居表中, 则路由器 Shepard 必须接收来自于路由器 Grissom 的 Hello 报文。然而, 路由器 Grissom 显然不接收来自于路由器 Shepard 的 Hello 报文。没有 Hello 报文的双向交换, 就不能形成一个邻接关系, 路由信息也就无法进行交换。

更仔细地检查路由器 Shepard 和路由器 Glenn 的邻居表, 更加证实了这个假设:

- 路由器 Grissom 的 SRTT 为 0, 表示从来没有一个数据包在路由路径上进行过往返 (round-trip);
- 路由器 Grissom 的 RTO 分别增加到了 5s 和 8s;
- 有一个关于路由器 Grissom 的数据包在队列中等待发送;
- 关于路由器 Grissom 记录的序列号为 0, 表示从来没有从路由器 Grissom 接收到可靠的数据包。

这些因素都表明, 这两台路由器都在试图向路由器 Grissom 发送可靠的报文, 但是却没有收到一个 ACK 确认报文。

图 8-48 中, 在路由器 Shepard 上使用命令 **debug eigrp packets** 可以更好地观察到底发生了什么。这样的话, 所有的 EIGRP 报文类型都将显示出来了, 但是可以使用第二个调试命令:

```
Shepard#debug eigrp packets
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK)
Shepard#debug ip eigrp neighbor 75 192.168.16.19
IP Neighbor target enabled on AS 75 for 192.168.16.19
IP-EIGRP Neighbor Target Events debugging is on
EIGRP: Sending UPDATE on Ethernet0 nbr 192.168.16.19, retry 14, RTO 5000
  AS 75, Flags 0x1, Seq 22/0 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno
1-4
EIGRP: Received HELLO on Ethernet0 nbr 192.168.16.19
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
EIGRP: Sending UPDATE on Ethernet0 nbr 192.168.16.19, retry 15, RTO 5000
  AS 75, Flags 0x1, Seq 22/0 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno
1-4
EIGRP: Received HELLO on Ethernet0 nbr 192.168.16.19
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
EIGRP: Sending UPDATE on Ethernet0 nbr 192.168.16.19, retry 16, RTO 5000
  AS 75, Flags 0x1, Seq 22/0 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno
1-4
EIGRP: Received HELLO on Ethernet0 nbr 192.168.16.19
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
EIGRP: Retransmission retry limit exceeded
EIGRP: Received HELLO on Ethernet0 nbr 192.168.16.19
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0
EIGRP: Enqueueing UPDATE on Ethernet0 nbr 192.168.16.19 iidbQ un/rely 0/1 peerQ
un/rely 0/0 serno 1-4
EIGRP: Sending UPDATE on Ethernet0 nbr 192.168.16.19
  AS 75, Flags 0x1, Seq 23/0 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno
1-4
```

图 8-48 调试命令 **debug ip eigrp neighbor** 用来控制命令 **debug eigrp packet** 所显示的报文

debug ip eigrp neighbor 75 192.168.16.19。这个命令是在第一个命令的基础上增加了一个过滤。它告诉 **debug eigrp packet** 只需要显示 EIGRP 75 (图 8-45 中那些路由器的进程 ID 号) 的报文并且只显示和邻居 192.168.16.19 (路由器 Grissom) 有关的那些报文。

图 8-48 中显示了路由器 Shepard 正在接收来自于路由器 Grissom 的 Hello 报文。它也显示了路由器 Shepard 正在试图发送更新报文给路由器 Grissom, 而路由器 Grissom 却不确认它们。这样进行了第 16 次重试后, 将显示一条“重传尝试次数限制超出”(“Retransmission retry limit exceeded”)的消息。这个超出限制的计数可以从路由器的邻居表中看出, 因为所显示的路由器 Grissom 的 uptime 时间都较低——一旦超过重传尝试次数的限制, 路由器 Grissom 将从路由器 Shepard 的邻居表中删除。但是, 由于路由器 Shepard 依然可以接收到来自于路由器 Grissom 的 Hello 报文, 所以路由器 Grissom 又将在路由器 Shepard 的邻居表中迅速地再次出现, 这个处理过程又将再次开始。

图 8-49 显示了在路由器 Shepard 上调试命令 **debug eigrp neighbors** 的输出信息。这个命令没有指定具体的 IP 地址, 而是改为显示 EIGRP 的邻居事件了。在这里, 显示了前面段落里描述的邻居事件的两种情况: 路由器 Grissom 在超出重传次数限制时被宣告丢失, 但一旦收到下一个 Hello 报文时又立即“找回来”了。

```
Shepard#debug eigrp neighbors
EIGRP Neighbors debugging is on
Shepard#
EIGRP: Retransmission retry limit exceeded
EIGRP: Holdtime expired
EIGRP: Neighbor 192.168.16.19 went down on Ethernet0
EIGRP: New peer 192.168.16.19
EIGRP: Retransmission retry limit exceeded
EIGRP: Holdtime expired
EIGRP: Neighbor 192.168.16.19 went down on Ethernet0
EIGRP: New peer 192.168.16.19
```

图 8-49 命令 **debug eigrp neighbors** 显示了邻居事件

虽然图 8-48 中显示正在向路由器 Grissom 发送更新报文, 但是在图 8-50 的路由器 Grissom 上, 通过对 EIGRP 报文的观察来看, 这个路由器并没有收到那些发送给它的报文。因为路由器 Grissom 可以和路由器 Cooper 成功地交换 Hello 报文, 因而路由器 Grissom 的 EIGRP 进程肯定是在运行的。因此就轮到怀疑是路由器 Grissom 的以太网接口故障了。观察路由器的配置文件, 发现在以太接口 E0 处配置了一个作为入站过滤的访问列表:

```
interface Ethernet0
 ip address 192.168.16.19 255.255.255.240
 ip access-group 150 in
!
!
access-list 150 permit tcp any any established
access-list 150 permit tcp any host 192.168.16.238 eq ftp
access-list 150 permit tcp host 192.168.16.201 any eq telnet
access-list 150 permit tcp any host 192.168.16.230 eq pop3
access-list 150 permit udp any any eq snmp
access-list 150 permit icmp any 192.168.16.224 0.0.0.15
```

待续


```

Grissom#debug eigrp packets
EIGRP Packets debugging is on
  (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK)
Grissom#
EIGRP: Sending HELLO on Serial0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Received HELLO on Serial0 nbr 192.168.16.42
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
EIGRP: Sending HELLO on Ethernet0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Sending HELLO on Serial0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Received HELLO on Serial0 nbr 192.168.16.42
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
EIGRP: Sending HELLO on Ethernet0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Sending HELLO on Serial0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Sending HELLO on Ethernet0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Received HELLO on Serial0 nbr 192.168.16.42
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
EIGRP: Sending HELLO on Serial0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
EIGRP: Sending HELLO on Ethernet0
  AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0

```

图 8-50 路由器 Grissom 正在和路由器 Cooper 通过接口 S0 交换 Hello 报文，并正在从接口 E0 发送出 Hello 报文。然而，路由器 Grissom 却没有在接口 E0 上收到任何 EIGRP 的报文

当在路由器 Grissom 的 E0 接口上收到 EIGRP 的报文时，这些报文首先要通过访问列表 150 的过滤。由于这些报文和访问列表中的任何一项都不匹配，因此它们就被丢弃了。这个问题可以通过在访问列表后附加一条匹配条目解决，如图 8-51 所示：

```

Grissom#show ip eigrp neighbors
IP-EIGRP neighbors for process 75
H  Address                Interface    Hold Uptime    SRTT    RTO    Q    Seq
                               (sec)        (ms)          Cnt    Num
2  192.168.16.17            Et0         10 00:06:20     4      200    0     41
1  192.168.16.18            Et0         14 00:06:24    15      200    0     85
0  192.168.16.42            Se0         10 06:22:56    22      200    0     12
Grissom#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

192.168.16.0/24 is variably subnetted, 6 subnets, 2 masks
C    192.168.16.40/30 is directly connected, Serial0
D    192.168.16.36/30 [90/2195456] via 192.168.16.18, 00:06:27, Ethernet0
C    192.168.16.16/28 is directly connected, Ethernet0
D    192.168.16.224/28 [90/2195456] via 192.168.16.42, 00:06:12, Serial0
D    192.168.16.192/28 [90/2323456] via 192.168.16.18, 00:06:27, Ethernet0
D    192.168.16.128/28 [90/307200] via 192.168.16.17, 00:06:12, Ethernet0
Grissom#

```

图 8-51 当在访问列表后增加一个条目来允许 EIGRP 报文通过时，路由器 Grissom 的邻居表和路由选择表显示出了它现在拥有可达所有子网的路由


```
access-list 150 permit eigrp 192.168.16.16 0.0.0.15 any
```

8.3.2 “卡”在活动状态的邻居 (Stuck-in-Active Neighbors)

当本地路由器的一条路由变为活动状态并且向它的邻居路由器发送查询报文时,在本地路由器收到每个查询报文的答复报文之前,这条路由将一直保持活动状态。但是如果一个邻居失效了或其他无效的情况下没有办法作出答复,那么究竟会发生什么呢?答案是这条路由将会永久地停留在活动状态。活动计时器将设计用来防止这种情况的发生。当发送一个查询报文时,活动计时器就被设置了。如果在收到查询报文的答复报文之前,活动计时器超时了,这条路由就被宣告“卡”在活动状态 (Stuck-in-active),这个邻居也就被推断为失效了,并且从邻居表中刷新掉。¹SIA 路由和任何其他经过这个邻居的路由也都会从路由选择表中删除。DUAL 算法将会认为这个邻居已经答复了一个含有无穷大度量的报文。

事实上,这一系列的事件应该从来不会发生。在活动计时器超时很久以前,丢失的 Hello 报文就应该识别出一个无效的邻居了。

但是在一个大型的 EIGRP 网络中,究竟发生了什么,使一个查询报文就像电池广告中的小兔子一样,一直保持向前继续?回忆一下,查询会引起扩散计算的不断增大,反之,答复报文会引起扩散计算的不断减小,如图 8-10 所示。这样,查询最终将必然会到达互联网络的边界,而答复最终也必然开始往回收缩,但是,如果扩散计算的直径增大到足够的大,活动计时器将可能在收到所有的答复前超时。结果,从邻居表中刷新掉一个合法的邻居将明显会带来网络的不稳定。

当一个邻居神秘地从邻居表中消失了,随后又重新出现,或者用户抱怨总是断断续续地不能到达目的地时,SIA 路由可能就是故障所在了。检查路由器的错误记录日志是找出是否出现 SIA 路由的一个好途径,如图 8-52 所示。

```
Gagarin#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level debugging, 3369 messages logged
  Monitor logging: level debugging, 0 messages logged
  Trap logging: level informational, 71 message lines logged
  Buffer logging: level debugging, 3369 messages logged

Log Buffer (4096 bytes):
...
...
...
DUAL: dual_rcvupdate(): 10.51.1.0/24 via 10.1.2.1 metric 409600/128256
DUAL: Find FS for dest 10.51.1.0/24. FD is 4294967295, RD is 4294967295 found
DUAL: RT installed 10.51.1.0/24 via 10.1.2.1
DUAL: Send update about 10.51.1.0/24. Reason: metric chg
DUAL: Send update about 10.51.1.0/24. Reason: new if
DUAL: dual_rcvupdate(): 10.52.1.0/24 via 10.1.2.1 metric 409600/128256
DUAL: Find FS for dest 10.52.1.0/24. FD is 4294967295, RD is 4294967295 found
%DUAL-3-SIA: Route 10.11.1.0/24 stuck-in-active state in IP-EIGRP 1. Cleaning up
Gagarin#
```

图 8-52 这个错误记录日志中的最后一条记录显示了一条 SIA 消息

¹ 正如前面所提及的,缺省的活动计时时间是 3min,并且可以通过命令 `timer active-time` 来更改。

当追踪 SIA 路由产生的原因时, 应该仔细关注路由器的拓扑结构表。如果路由处于活动状态了, 那么就应该注意邻居路由器仍然没有收到查询报文。例如, 在图 8-53 中显示了一个含有几条处于活动状态的路由的拓扑结构表。注意, 这里大多数的路由已经有 15s 的时间处于活动状态了, 而另一个路由 (10.6.1.0) 进入活动状态则已经有 41s 的时间了。

```
Gagarin#show ip eigrp topology
IP-EIGRP Topology Table for process 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply
       r - Reply Status

A 10.11.1.0/24, 0 successors, FD is 3072128000, Q
  1 replies, active 00:00:15, query-origin: Local origin
  Remaining replies:
    via 10.1.2.1, r, Ethernet0
A 10.10.1.0/24, 0 successors, FD is 3584128000, Q
  1 replies, active 00:00:15, query-origin: Local origin
  Remaining replies:
    via 10.1.2.1, r, Ethernet0
A 10.9.1.0/24, 0 successors, FD is 4096128000, Q
  1 replies, active 00:00:15, query-origin: Local origin
  Remaining replies:
    via 10.1.2.1, r, Ethernet0
A 10.2.1.0/24, 1 successors, FD is Inaccessible, Q
  1 replies, active ve 00:00:15, query-origin: Local origin
  Remaining res:
    via 10.1.2.1, r, Ethernet0
P 10.1.2.0/24, 1 successors, FD is 281600
  via Connected, Ethernet0
A 10.6.1.0/24, 0 successors, FD is 3385160704, Q
  1 replies, active 00:00:41, query-origin: Local origin
  Remaining replies:
    via 10.1.2.1, r, Ethernet0
A 10.27.1.0/24, 0 successors, FD is 3897160704, Q
--More--
```

图 8-53 这个拓扑结构表显示了几个处于活动状态的路由, 它们都在等待来自邻居路由器 10.1.2.1 的答复

也注意到在每个条目中, 邻居路由器 10.1.2.1 都带有一个答复状态标记 (r), 这表明该邻居的答复仍然没有被收到。邻居路由器本身或者和邻居路由器相连的链路可能没有问题, 但是在互联网的拓扑中这个信息指出了应该将追查继续下去的方向。

通常在一个大型的 EIGRP 互联网络里引起 SIA 的原因是网络拥塞严重、数据链路带宽较低和路由器内存过低或 CPU 利用率负荷过大等等。如果这些有限的资源必须处理数量很大的查询报文的话, 这个问题将会进一步恶化。

冒然地调整接口上的带宽参数可能会引起另外的 SIA 路由。回忆一下在设计 EIGRP 网络时, EIGRP 使用的带宽一般不超过链路可用带宽的 50%。这个限制意味着 EIGRP 的调节是和所配置的带宽相关联的。如果试图在处理路由选择时人为地降低带宽, EIGRP 进程处理所需求的带宽将可能会极度缺乏。假如运行的是 IOS 软件的 11.2 版本或后续的版本, 则可以使用命令 **ip bandwidth-percent eigrp** 来调整带宽使用的百分比。

例如, 假设一个接口和一个带宽为 56K 的串行链路相连, 但是链路带宽被设置成了 14kbit/s。EIGRP 协议应该限制自己使用的带宽在这个数值的 50% 之内, 或者说就是在 7kbit/s 之内。下面的命令将把 EIGRP 协议使用的带宽百分比调整到 200%, 即 14kbit/s 的 200%, 也就是

实际 56K 链路带宽的 50%:

```
interface Serial 3
 ip address 172.18.107.210 255.255.255.240
 bandwidth 14
 ip bandwidth-percent eigrp 1 200
```

也可以使用命令 **timers active-time** 来增大活动计时器的周期, 这样在某些情况下可以帮助避免 SIA 路由, 但是采取这个办法应当仔细考虑对网络路由收敛的影响。

一个好的互连网络设计应该是解决像 SIA 路由这类网络不稳定性的最好方法。通盘考虑使用灵活的地址分配、路由过滤、缺省路由和路由汇总, 在一个大型的互连网络里创建一些边界来限制扩散计算的大小尺寸和范围。第 13 章“路由过滤”将包含一个这种网络设计的例子。

8.4 展 望

当比较 EIGRP 协议和 OSPF 协议时, 人们经常说 EIGRP 协议的优势在于它的配置比较简单。这个看法在很多互连网络的情况中是正确的, 但是本章故障处理一节的讲述显示了当互连网络的规模增大时, 更多的努力将要放在划分 EIGRP 网络的拓扑上了。相反地, 正如下一章所描述的, 非常复杂的 OSPF 在配置一个大型的互连网络时反而显得简单了。

8.5 总结表: 第 8 章命令总结

命 令	描 述
accept-lifetime <i>start-time{infinitelend-timeduration seconds}</i>	设置一个时间段, 用来指定钥匙链上的认证钥匙可被接受的有效时间
auto-summary	在网络边界上打开或关闭自动路由汇总功能, 这个命令缺省的配置是打开
bandwidth <i>kilobits</i>	在接口上指定带宽参数, 单位是 kbit/s。在一些路由选择协议中用来计算度量值, 但它不影响数据链路实际的带宽
debug eigrp packets	显示 EIGRP 报文的活动行为
debug ip eigrp neighbor <i>process-id address</i>	在命令 debug eigrp packets 基础上增加一个过滤, 告诉路由器只显示选定的进程 ID 号和邻居的 IP 报文
delay <i>tens-of-microseconds</i>	在接口上指定延迟参数, 单位是 10μs。在一些路由选择协议中用来计算度量值, 但它不影响数据链路实际的延迟
ip authentication key-chain eigrp <i>process-id key-chain</i>	在一个运行 EIGRP 协议的接口上配置一个钥匙链, 并指定一个钥匙链所使用的名字
ip authentication mode eigrp <i>process-id md5</i>	在一个接口上指定 EIGRP 协议使用的认证类型
ip bandwidth-percent eigrp <i>process-id percent</i>	配置 EIGRP 协议所使用的带宽百分比, 缺省配置是 50%
ip hello-interval eigrp <i>process-id seconds</i>	配置 EIGRP 的 Hello 报文的时间间隔
ip hold-time eigrp <i>process-id seconds</i>	配置 EIGRP 的抑制时间
ip summary-address eigrp <i>process-id address mask</i>	配置路由器发送一个汇总的 EIGRP 通告
key number	指定在钥匙链上一个钥匙
key chain <i>name-of-chain</i>	指定一组认证钥匙
key-string <i>text</i>	指定钥匙使用的认证字符串或口令

续表

命 令	描 述
metric weights <i>tos k1 k2 k3 k4 k5</i>	指定在 IGRP 和 EIGRP 协议中计算复合度量值时, 对带宽、负载、延迟和可靠性等参数所使用的权重
network <i>network-number</i>	指定覆盖一个或多个运行 IGRP、EIGRP 或 RIP 协议进程的接口的网络地址
passive-interface <i>type number</i>	使一个接口不再发送广播的或组播的路由选择更新
router eigrp <i>process-id</i>	在路由器上启动 EIGRP 路由选择进程
send-lifetime <i>start-time{infinite end-time duration seconds}</i>	设置一个时间段, 用来指定钥匙链上的认证钥匙可被发送的有效时间
show ip eigrp neighbors [<i>type number</i>]	用来显示 EIGRP 的邻居表
show ip eigrp topology [<i>process-id</i>][<i>ip address</i>] <i>mask</i>]	用来显示 EIGRP 的拓扑结构表
timers active-time (<i>minutes</i> <i>disabled</i>)	改变或关闭缺省的 3min 的活动状态计时
traffic-share (<i>balanced</i> <i>min</i>)	指定 IGRP 协议或 EIGRP 协议路由选择进程是否使用非等价负载均衡或只使用等价负载均衡
variance <i>multiplier</i>	指定一个倍数来表示一条路由与最小代价路径的度量值所差别的程度, 确定是否可以依然包含在非等价负载均衡“组”中

8.6 复 习 题

1. EIGRP 协议是一个距离矢量协议还是一个链路状态路由选择协议?
2. EIGRP 协议在一条链路上使用的可配置最大带宽是多少? 这个带宽百分比可以更改吗?
3. EIGRP 协议和 IGRP 协议在计算复合度量时有什么不同?
4. EIGRP 协议的 4 个基本部件是什么?
5. 通过 EIGRP 协议的上下文, 术语“可靠的分发 (reliable delivery)”是什么意思? 有哪两种方法可以确保 EIGRP 报文的可靠分发?
6. 有什么机制可以确保路由器正在接收的是最新的路由条目?
7. EIGRP 协议使用的组播 IP 地址是什么?
8. EIGRP 协议使用的报文类型是什么?
9. 缺省情况下, EIGRP 协议发送 Hello 报文的时间间隔是多少?
10. 缺省的抑制时间是多少?
11. 邻居表和拓扑结构表的不同之处是什么?
12. 什么是可行距离?
13. 什么是可行性条件?
14. 什么是可行后继路由器?
15. 什么是后继路由器?
16. 处于活动状态的路由和处于被动状态的路由有什么不同之处?
17. 引起一个被动状态的路由变成活动状态的条件是什么?
18. 引起一个活动状态的路由变成被动状态的条件是什么?
19. Stuck-in-active 是什么意思?
20. 子网划分和地址聚合有什么不同之处?

8.7 配置练习

1. 在图 8-42 和相关的案例研究中, 自动路由汇总功能在路由器 Cochran 和路由器 Lindbergh 上关闭了。结果, 在路由器 Earhart 的路由选择表中就可以得到变长掩码的子网 192.168.18.24/29 和 192.168.18.128/25。为了使路由器 Curtiss 上的有类别协议 IGRP 进程能够正确地路由到这些子网, 需要做哪些进一步的配置?

2. 写出图 8-54 中路由器 A、B 和 C 的 EIGRP 配置, 并使用进程 ID 号 5。

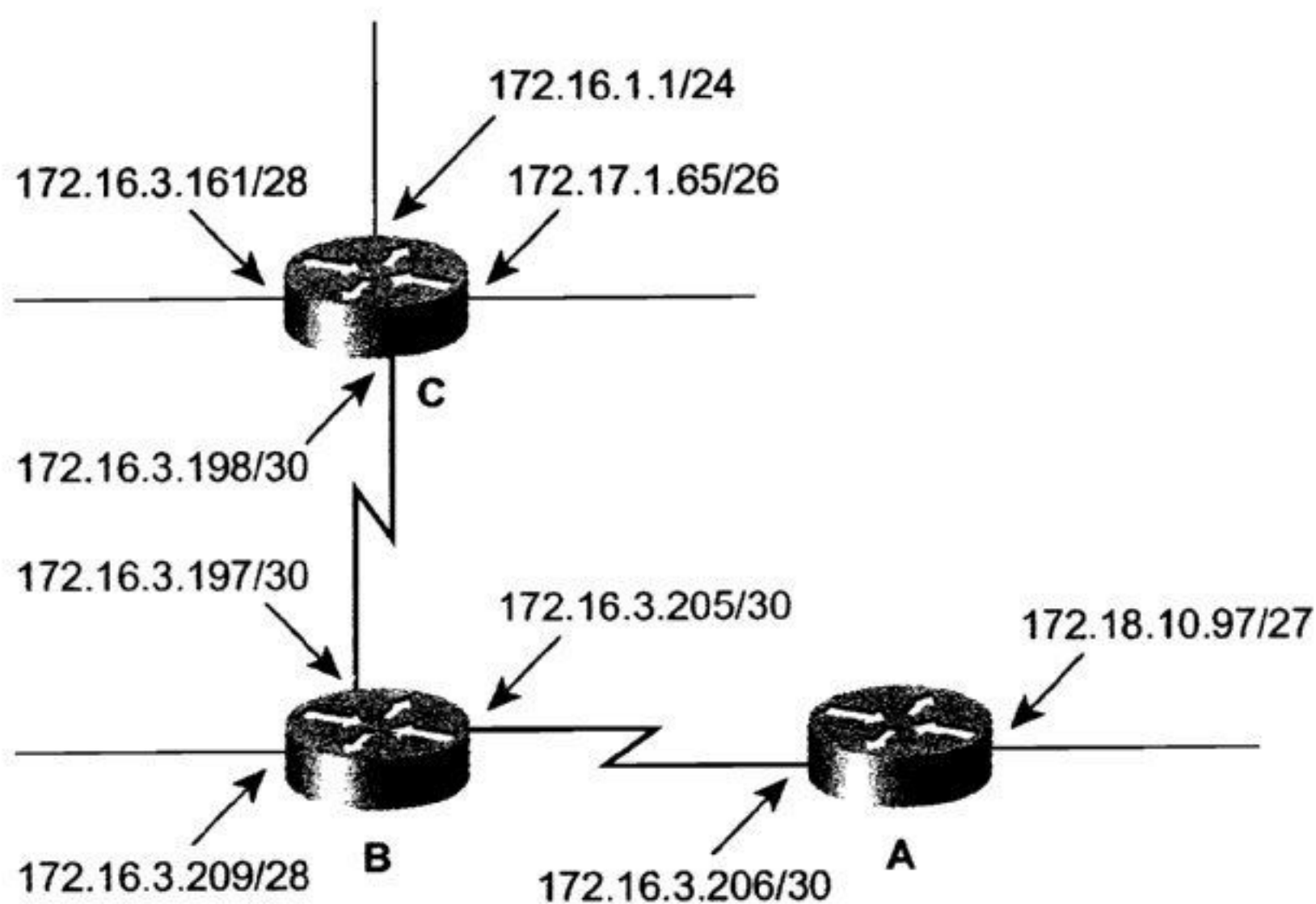


图 8-54 配置练习 2 和 3 的互联网络

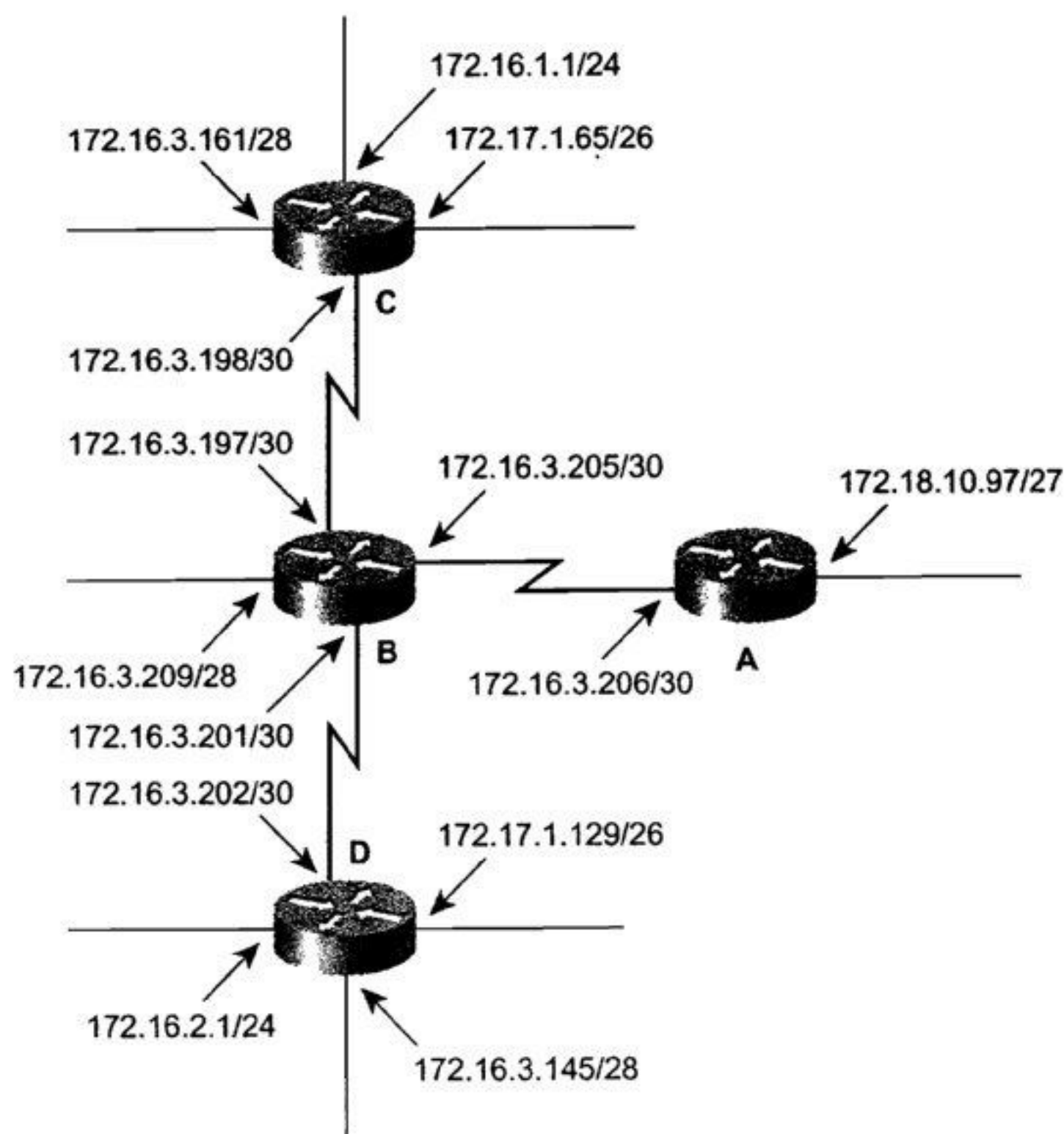


图 8-55 配置练习 4 的互联网络

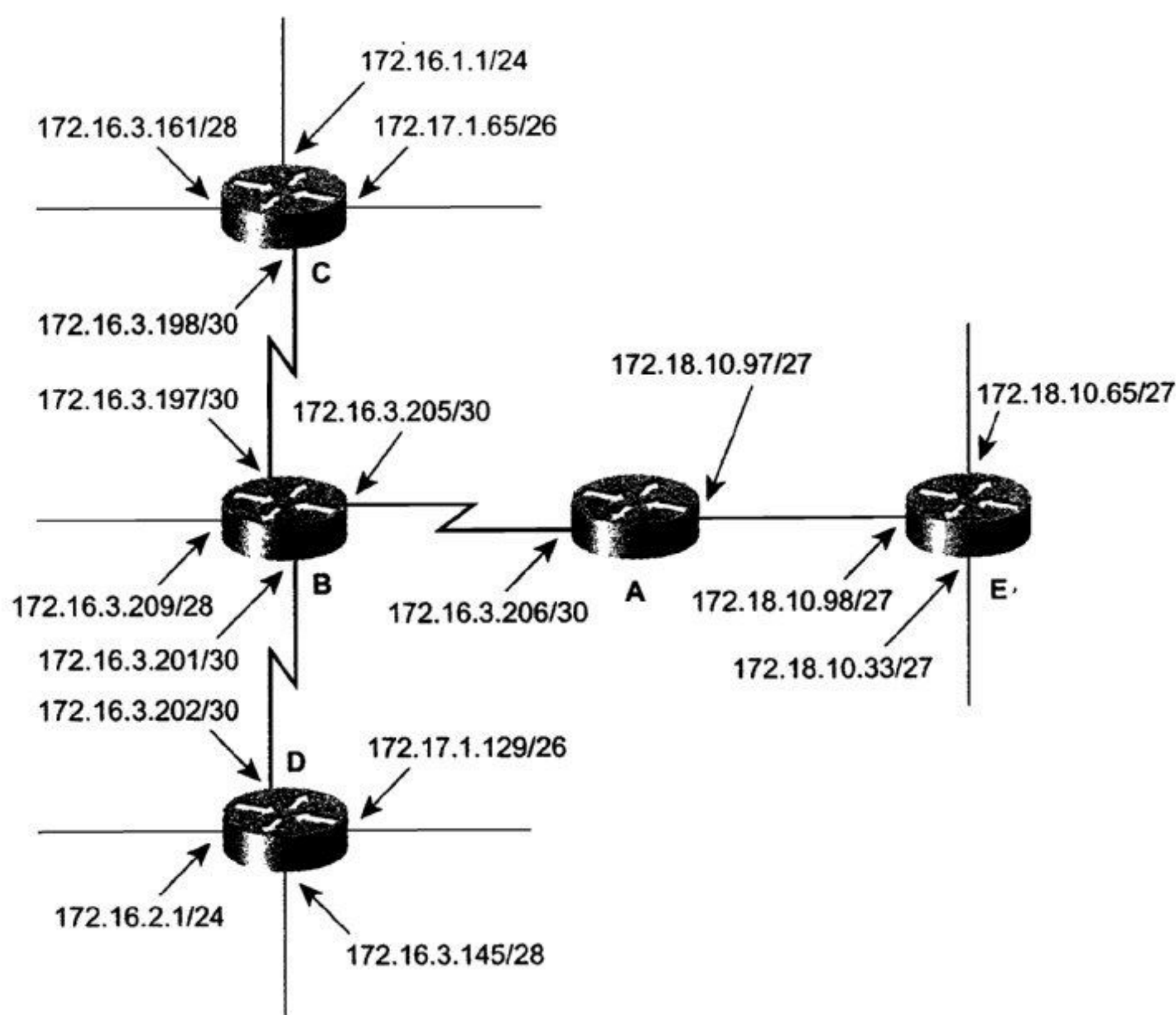


图 8-56 配置练习 5 的互联网络

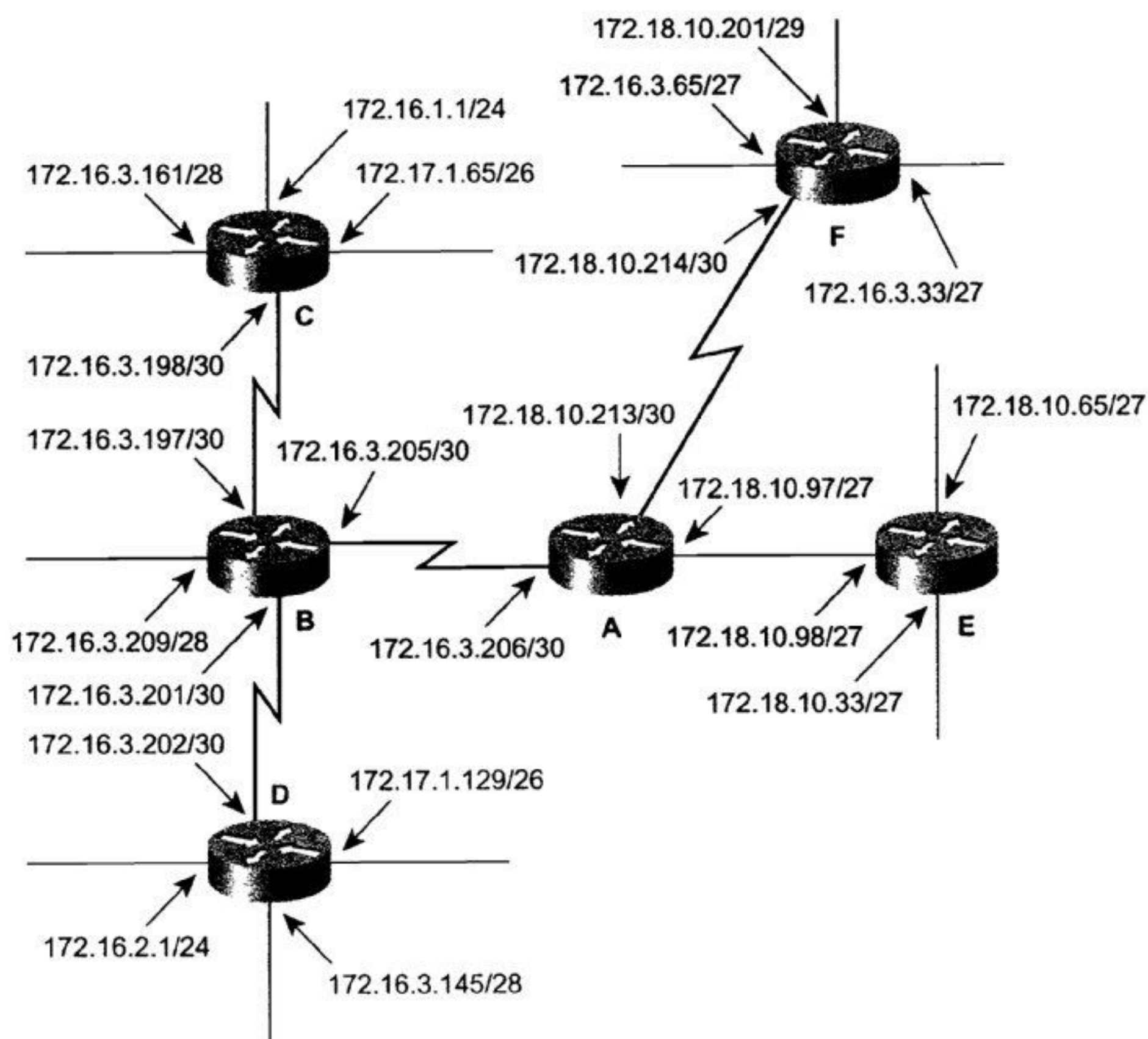


图 8-57 配置练习 6 和 7 的互联网络

3. 在图 8-54 中, 连接路由器 A 和 B 的串行接口都是 S0 接口。配置这两台路由器之间的认证, 假设从今天开始, 有 2 天的时间使用第一个钥匙。然后配置第二个钥匙, 在第一个钥匙使用过后 30 天开始使用。
4. 在图 8-55 中增添了路由器 D。增加这台路由器到配置练习 3 所写的配置中。
5. 在图 8-56 中增添了路由器 E, 并且它只运行 IGRP 协议。增加这台路由器到配置练习 3 和 4 所写的配置中。
6. 在图 8-57 中增添了路由器 F, 配置这台路由器, 在与配置练习 3、4 和 5 中配置的路由器之间运行 EIGRP 协议。
7. 在图 8-57 的互联网络上任何可能的地方配置路由汇总。

8.8 故障排除练习

1. 在一个路由器上配置了 EIGRP 协议和 IGRP 协议的路由重新分配, 如下所示:

```
router eigrp 15
 network 192.168.5.0
 no auto-summary
 metric weights 0 1 1 0 1 1
!
router igrp 5
 network 172.16.0.0
 metric weights 0 0 0 1 1 1
```

EIGRP 域内的路由器没有学习到 IGRP 域的路由, 而 IGRP 域内的路由器没有学习到 EIGRP 域的路由, 出现了什么错误?

2. 表 8-6 显示了在图 8-58 中的每一个接口上使用 **show interface** 命令显示的数值。哪一台路由器将作为路由器 F 到达子网 A 的后继路由器?

表 8-6 使用 **show interface** 命令显示的图 8-58 中所有接口的度量值

路 由 器	接 口	带宽 BW(k)	延迟 DLY(μs)
A	E0	10000	1000
	F0	100000	100
	T0	16000	630
	S0	512	20000
	S1	1544	20000
B	T0	16000	630
	S0	1544	20000
C	E0	10000	1000
	S0	1544	20000
D	E0	10000	1000
	F0	100000	100
	S0	1544	20000
E	E0	10000	1000
	S0	1544	20000
F	F0	100000	100
	S0	512	20000
G	E0	10000	1000
	E1	10000	1000
	F0	100000	100
	S0	56	20000

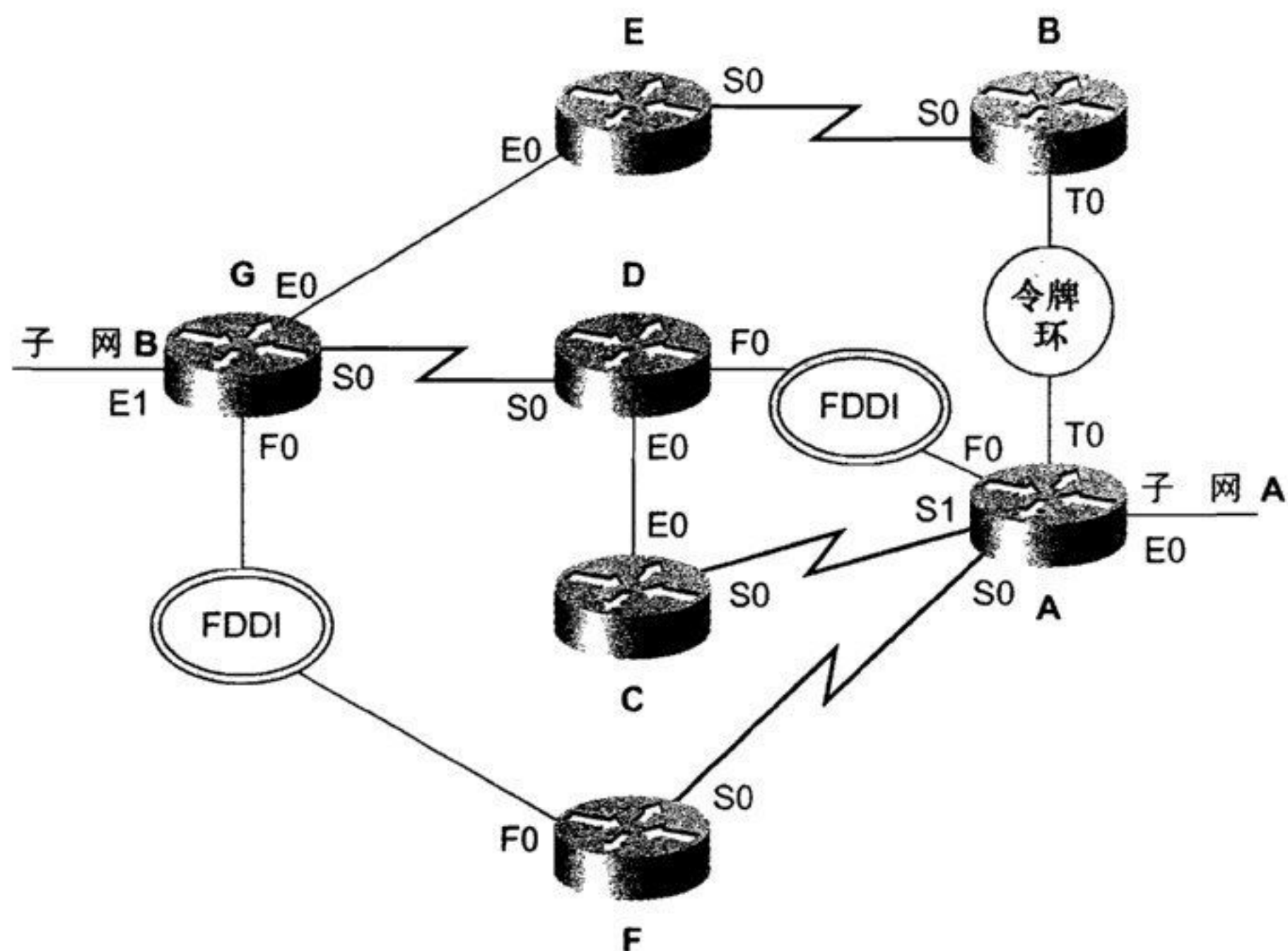


图 8-58 故障排除练习 2~6 的互联网络

3. 在图 8-58 中, 路由器 C 到达子网 A 的可行距离是什么?
4. 在图 8-58 中, 路由器 G 到达子网 A 的可行距离是什么?
5. 在图 8-58 中, 在路由器 G 的拓扑结构表中, 有哪些路由器是作为可行后继路由器的?
6. 在图 8-58 中, 路由器 A 到达子网 B 的可行距离是什么?

第 9 章

开放最短路径优先 协议（OSPF）

本章主要包括以下主题：

- OSPF 协议的操作
 - 邻居和邻接关系
 - 区域
 - 链路状态数据库
 - 路由选择表
 - 认证
 - OSPF 的报文格式
 - OSPF 的 LSA 格式
 - 可选项字段
- OSPF 协议的配置
 - 案例研究：一个基本的 OSPF 配置
 - 案例研究：使用 Loopback 接口设置路由器的 ID
 - 案例研究：域名服务查询
 - 案例研究：OSPF 和辅助地址
 - 案例研究：末梢区域
 - 案例研究：完全末梢区域
 - 案例研究：NSSA 区域
 - 案例研究：地址汇总
 - 案例研究：认证
 - 案例研究：虚链路
 - 案例研究：运行在 NBMA 网络上的 OSPF
 - 案例研究：运行在按需电路上的 OSPF
- OSPF 协议的故障排除
 - 案例研究：孤立的区域

案例研究: 路由汇总配置错误

开放最短路径优先协议(Open Shortest Path First, OSPF)是由 Internet 工程任务组(Internet Engineering Task Force, IETF)开发的路由选择协议, 用来替代存在一些问题的 RIP 协议。现在, OSPF 协议是 IETF 组织建议使用的内部网关协议(IGP)。OSPF 协议是一个链路状态协议, 正如它的命名所描述的, OSPF 使用 Dijkstra 的最短路径优先算法(SPF), 而且是开放的。这里所说的开放是指它不属于任何一个厂商或组织所私有。OSPF 协议的发展经过了几个 RFC, 所有这些相关的 RFC 都是由 John Moy 撰写的。RFC 1131 详细说明了 OSPF 协议的版本 1, 这个版本从来没有在实验平台以外使用过。OSPF 协议的版本 2, 也就是目前仍然使用的版本, 最初的说明是在 RFC 1247 里描述的, 最新的 OSPF 协议说明是 RFC2328。¹

像所有的链路状态协议一样, OSPF 协议和距离矢量协议相比, 一个主要的改善就在于它的快速收敛, 这样使 OSPF 协议可以支持更大型的互联网络, 并且不容易受到有害路由选择信息的影响。OSPF 协议的其他一些特性有:

- 使用了区域的概念, 这样可以有效地减少路由选择协议对路由器的 CPU 和内存的占用, 划分区域还可以降低路由选择协议的通信量, 这使构建一个层次化的互联网络拓扑成为可能;
- 完全无类别地处理地址问题, 排除了像不连续的子网这样的有类别路由选择协议的问题;
- 支持无类别的路由选择表查询、VLSM 和用来进行有效地址管理的超网技术;
- 支持无大小限制的、任意的度量值;
- 支持使用多条路由路径的效率更高的等价负载均衡;²
- 使用保留的组播地址来减小对不宣告 OSPF 报文的设备的影响;
- 支持更安全的路由选择认证;
- 使用可以跟踪外部路由的路由标记。

OSPF 协议也支持具有服务类型 (Type of Service, TOS) 的路由选择能力, 但是它从来没有被广泛地实施过。基于这个原因, RFC 2328 已经在 OSPF 协议中删除了这个 TOS 路由选择选项。

9.1 OSPF 的操作³

从一个非常宏观的角度来看, OSPF 协议的操作是比较容易解释的:

1. 宣告 OSPF 的路由器从所有启动 OSPF 协议的接口上发出 Hello 报文。如果两台路由器共享一条公共数据链路, 并且能够相互成功协商它们各自 Hello 报文中所指定的某些参数, 那么它们就成为了邻居 (Neighbor)。
2. 邻接关系 (Adjacency), 可以想象成为一条点到点的虚链路, 它是在一些邻居路由器

¹ 正在编写本章的时候, RFC2328 发布了, 并废弃了 RFC2178。

² 更准确地说, RFC 建议称为等价多路径——多条等价路径的发现和使用, 但是 RFC 并没有指明路由选择协议应该怎么样路由转发单个数据包通过这些多条不同的路径。在 Cisco 的路由器上, OSPF 的实现执行的是前面章节描述过的等价负载均衡。

³ 由于 OSPF 协议的术语和概念的相互关系, 本章在完整地描述它们的定义之前将会频繁地使用这些术语和概念。建议读者多阅读几遍本章的内容, 而不是仅仅阅读一遍, 以便确保对 OSPF 协议的操作有一个完全的理解。复习一下第 4 章“动态路由选择协议”中的“链路状态路由选择协议”一节也是很有帮助的。

之间构成的。OSPF 协议定义了一些网络类型和一些路由器类型的邻接关系。邻接关系的建立是由交换 Hello 报文信息的路由器类型和交换 Hello 报文信息的网络类型决定的。

3. 每一台路由器都会在所有形成邻接关系的邻居之间发送链路状态通告 (Link State Advertisement, LSA)。LSA 通告描述了路由器所有的链路信息 (或接口) 和链路状态信息。这些链路可以是到一个末梢网络 (stub network, 是指没有和其他路由器相连的网络) 的链路、到其他 OSPF 路由器的链路、到其他区域网络的链路, 或是到外部网络 (从其他的路由选择进程学习到的网络) 的链路。由于这些链路状态信息的多样性, OSPF 协议定义了许多 LSA 类型。

4. 每一个收到从邻居路由器发出的 LSA 通告的路由器都会把这些 LSA 通告记录在它的链路状态数据库当中, 并且发送一份 LSA 的拷贝给该路由器的其他所有邻居。

5. 通过 LSA 泛洪到整个区域, 所有的路由器都会形成同样的链路状态数据库。

6. 当这些路由器的数据库都完全相同时, 每一台路由器都将以它本身为根, 使用 SPF 算法去计算一个无环路的拓扑图, 来描述它所知道的到达每一个目的地的最短路径 (最小的路径代价)。这个拓扑图就是 SPF 算法树。

7. 最后, 每一台路由器都将从 SPF 算法树中构建出自己的路由选择表。¹

当所有的链路状态信息泛洪到一个区域内的所有路由器上——也就是说, 链路状态数据库同步了——并且成功创建路由选择表时, OSPF 协议就变成了一个“安静”的协议。邻居之间交换的 Hello 报文称为 keepalive 报文, 并且每隔 30min 重传一次 LSA。如果互联网络的拓扑是稳定的, 那么网络中将不会有什么活动或行为发生。

9.1.1 邻居和邻接关系

在发送任何 LSA 通告之前, OSPF 路由器都必须首先发现它们的邻居路由器并建立起邻接关系。邻居路由器, 连同每一台邻居路由器所在的链路 (接口) 和维护邻居路由器的一些必要的其他信息都被记录在一个邻居表里, 如图 9-1 所示。

Monet#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.30.70	1	FULL/DR	00:00:34	192.168.17.73	Ethernet0
192.168.30.254	1	FULL/DR	00:00:34	192.168.32.2	Ethernet1
192.168.30.70	1	FULL/BDR	00:00:34	192.168.32.4	Ethernet1
192.168.30.30	1	FULL/-	00:00:33	192.168.17.50	Serial0.23
192.168.30.10	1	FULL/-	00:00:32	192.168.17.9	Serial1
192.168.30.68	1	FULL/-	00:00:39	192.168.21.134	Serial2.824
192.168.30.18	1	FULL/-	00:00:30	192.168.21.142	Serial2.826
192.168.30.78	1	FULL/-	00:00:36	192.168.21.170	Serial2.836

图 9-1 邻居表记录了所有宣告 OSPF 协议的邻居路由器

一台 OSPF 路由器对其他 OSPF 路由器的跟踪需要每台路由器都提供一个路由器的 ID 号 (RouterID), 路由器 ID 是在 OSPF 区域内惟一标识一台路由器的 IP 地址。Cisco 的路由器通

¹ 受到路由过滤影响的是利用链路状态数据库计算路由的基本过程, 而不是邻居之间交换路由的基本过程。如需了解更详细的内容, 请参见第 13 章“路由过滤”。

过下面的方法得到它们的路由器 ID:

(1) 首先, 路由器选取它所有的 loopback 接口上数值最高的 IP 地址;

(2) 如果路由器没有配置 IP 地址的 loopback 接口, 那么路由器将选取它所有的物理接口上数值最高的 IP 地址。用作路由器 ID 的接口不一定非要运行 OSPF 协议。

使用 loopback 地址作为路由器 ID 有两个好处:

- loopback 接口比任何其他物理接口都更稳定。一旦路由器启动成功, 这个环回接口就处于活动状态了, 只有整个路由器失效时它才会失效;
- 网络管理员在预先分配和识别作为路由器 ID 的地址时有更多的回旋余地。

在 Cisco 的路由器上, 即使路由器的这个用作路由器 ID 的物理接口随后失效了或被删除了, OSPF 协议也会继续使用原来的物理接口作为路由器 ID (参见本章后面的小节“案例研究: 使用 loopback 接口设置路由器 ID”)。因此, loopback 接口的稳定性只是一个次要的优点, loopback 接口的一个主要好处在于它具有更好控制路由器 ID 的能力。

OSPF 路由器利用 Hello 报文通告它的路由器 ID 来开始建立和邻居的关系。

1. Hello 报文协议

Hello 报文协议服务于以下几个目的:

- 它是发现邻居路由器的方法;
- 在两台路由器成为邻居之前, 需要通过 Hello 报文协议通告这两台路由器必须相互认可的几个参数;
- Hello 报文在邻居路由器之间担当 keepalive 的角色;
- 它确保了邻居路由器之间的双向通信;
- 它用来在一个广播网络或非广播多址(NBMA)的网络上选取指定路由器(Designated Router, DR)和备份指定路由器(Backup Designated Router, BDR)。

宣告 OSPF 的路由器周期性地从启动 OSPF 协议的每一个接口上发送出 Hello 报文。这个周期性的时间段称为 Hello 时间间隔 (HelloInterval), 它的配置是基于路由器的每一个接口的。在 Cisco 路由器上, 使用的缺省 Hello 时间间隔是 10s。¹这个值可以通过命令 **ip ospf hello-interval** 来更改。如果一台路由器在一个称为路由器无效时间间隔 (RouterDeadInterval) 的时间段内还没有收到来自于邻居的 Hello 报文, 那么它将宣告它的邻居路由器无效。在 Cisco 的路由器中, 路由器无效时间间隔的缺省值是 Hello 时间间隔的 4 倍, 并且这个值可以通过命令 **ip ospf dead-interval** 来更改。²

每一个 Hello 报文都包含以下的信息:

- 始发路由器的路由器 ID (RouterID);
- 始发路由器接口的区域 ID (AreaID);
- 始发路由器接口的地址掩码;
- 始发路由器接口的认证类型和认证信息;
- 始发路由器接口的 Hello 时间间隔;
- 始发路由器接口的路由器无效时间间隔;
- 路由器的优先级;

¹ 在 NBMA 的网络上缺省值是 30s。

² 在 RFC2328 里没有为 HelloInterval 或者 RouterDeadInterval 设置一个必需的值, 虽然它建议采用 10s 和 4×HelloInterval。

- 指定路由器 (DR) 和备份指定路由器 (BDR)；
- 标识可选的性能的 5 个标记位；
- 始发路由器的所有有效邻居的路由器 ID。这个列表仅仅包含这样一些所谓有效的邻居路由器——即在最近的路由器无效时间间隔内，始发路由器接口可以从其接收到 Hello 报文的那些邻居。

本节概述了上面列出的多数信息的含义和用法。随后的章节将会详细地讲述指定路由器 DR、备份指定路由器 BDR、路由器的优先级和阐明 Hello 报文的详细格式。当一台路由器从它的邻居路由器收到一个 Hello 报文时，它将检验这个 Hello 报文携带的区域 ID、认证信息、网络掩码、Hello 间隔时间、路由器无效时间间隔以及可选项的数值是否和接收接口上配置的对应值相匹配。如果它们不匹配，那么这个 Hello 报文将被丢弃，而且邻接关系也无法建立。

如果所有的参数都匹配，那么这个 Hello 报文就被认为是有效的。而且，如果始发路由器的路由器 ID 已经在接收该 Hello 报文的接口的邻居表中列出了，那么路由器无效时间间隔计时器将被重置。如果始发路由器的路由器 ID 没有在邻居表中列出，那么就把这个路由器 ID 加入到它的邻居表里。

无论何时，路由器发送一个 Hello 报文时，都会在这个报文里列出传送该报文的链路上所出现的所有邻居的路由器 ID。如果一台路由器收到了一个有效的 Hello 报文，并在这个 Hello 报文中发现了它自己的路由器 ID，那么这台路由器就认为双向通信 (two-way communication) 建立成功了。

一旦双向通信成功建立，邻接关系也就可能建立了。然而，正如前面所提及的，并不是所有的邻居路由器都会成为邻接对象。一个邻接关系的形成与否是依赖于和这两台互为邻居的路由器所连网络的类型的。另外，网络类型也影响 OSPF 报文传送的方式。因此，在讲述邻接关系之前，讲述网络类型是必要的。

2. 网络类型

OSPF 协议定义了以下 5 种网络的类型：

- 点到点网络 (Point-to-Point)；
- 广播型网络 (Broadcast)；
- 非广播多址 (NBMA) 网络；
- 点到多点网络 (Point-to-Multipoint)；
- 虚链路 (Virtual Links)。

(1) 点到点网络 (Point-to-Point)

点到点网络，像 T1 链路或其子速率的链路，是连接单独的一对路由器的。在点到点网络上的有效邻居总是可以形成邻接关系。在这些网络上的 OSPF 报文的目的地址也总是保留的 D 类地址 224.0.0.5，这个组播地址称为 AllSPFRouters。¹

(2) 广播型网络 (Broadcast)

广播型网络，像以太网、令牌环网和 FDDI，也可以更确切地定义为广播型多址网络，以便区别于 NBMA 网络。广播型网络是多址的网络，因而它们可以连接多于两台设备。而且由于它们是广播型的，因而连接在这种网络上的所有设备都可以接收到个别传送的报文。

¹ 这个规则的一个例外是重传的 LSA 报文，它们在所有的网络类型的网络上都是使用单播方式发送的。这个例外的情况将在后面的“可靠的泛洪：确认”一节中讲述。

在广播型网络上的 OSPF 路由器正如下一节“指定路由器和备份指定路由器”中所讲述的，会选举一个指定路由器 DR 和一个备份指定路由器 BDR。Hello 报文像所有始发于 DR 和 BDR 的 OSPF 报文一样，是以组播方式发送到 AllSPFRouters（目的地址是 224.0.0.5）的。携带这些报文的数据帧的目的介质访问控制（MAC）地址是 0100.5E00.0005。其他所有的路由器都将以组播方式发送链路状态更新报文和链路状态确认报文（将在后面讲述）到保留的 D 类地址 224.0.0.6，这个组播地址称为 AllDRouters。携带这些报文的数据帧的目的 MAC 地址是 0100.5E00.0006。

（3）非广播多址（NBMA）网络

NBMA 网络，像 X.25、帧中继和 ATM 等，可以连接两台以上的路由器，但是它们没有广播数据包的能力。一个在 NBMA 网络上的路由器发送的报文将不能被其他与之相连的路由器收到。结果，在这些网络上的路由器有必要增加额外的配置来获得它们的邻居。在 NBMA 网络上的 OSPF 路由器需要选举 DR 和 BDR，并且所有的 OSPF 报文都是单播的。

（4）点到多点网络（Point-to-Multipoint）

点到多点网络是 NBMA 网络的一个特殊配置，可以被看作是一群点到点链路的集合。在这些网络上的 OSPF 路由器不需要选举 DR 和 BDR，因为这些网络可以被看作点到点链路，而且 OSPF 报文是组播的。

（5）虚链路（Virtual Links）

虚链路将在后面一节讲述，它可以被路由器认为是没有编号的点到点网络的一种特殊配置。在虚链路上 OSPF 报文是以单播方式发送的。

（6）传送网络（Transit Network）和末梢网络（Stub Network）

除了以上那 5 种网络类型外，应该注意的是，所有的网络也都可以归纳到下面两种更普通的网络类型之一：

- **传送网络（Transit Network）**——和两台或两台以上的路由器相连。这种网络仅仅传送那些“只需仅仅通过”的数据包，也就是这样的一些数据包——它们的始发网络和目的网络都不同于当前的传送网络。
- **末梢网络（Stub Network）¹**——仅仅和一台路由器相连。末梢网络上的数据包总是有一个源地址或者目的地址属于这个末梢网络。也就是说，末梢网络上的所有数据包要么始发于这个末梢网络上的某个设备，要么终止于这个末梢网络上的某个设备。OSPF 协议在末梢网络上通告主机路由（就是网络掩码为 255.255.255.255 的路由）。Loopback 接口也可以认为是末梢网络，并当作主机路由来通告。²

3. 指定路由器和备份指定路由器

对于 OSPF 协议来说，在多址网络上有关 LSA 的泛洪（flooding，将在后面的章节讲述）方面还存在两个问题：

（1）在构建相关路由器之间的邻接关系时，会创建很多不必要的 LSA。假设在一个多址网络上有 n 台路由器，那么就会构成 $n(n-1)/2$ 个邻接关系，如图 9-2。每台路由器都会通告出 $n-1$ 条 LSA 信息到与之存在邻接关系的邻居路由器，再加上 1 个网络 LSA，这样计算的最终

¹ 注意，不要把 stub 网络和本章后面章节讲述的 stub 区域的概念相混淆。

² 从 IOS 11.3 版本开始，这个缺省的行为可以在 loopback 接口上增加使用命令 `ip ospf network point-to-point` 来改变。这将可以使 loopback 接口的地址作为一个子网地址来通告。

结果是，这个网络上将产生出 n^2 个 LSA 通告。

(2) 在多址网络上，它本身的泛洪显得比较混乱。某一台路由器向与它存在邻接关系的所有邻居发出 LSA，同样地，这些邻接的邻居路由器又向与它自己有邻接关系的邻居的邻居发出这个 LSA，这样将会在同一个网络上创建很多个相同 LSA 的拷贝。

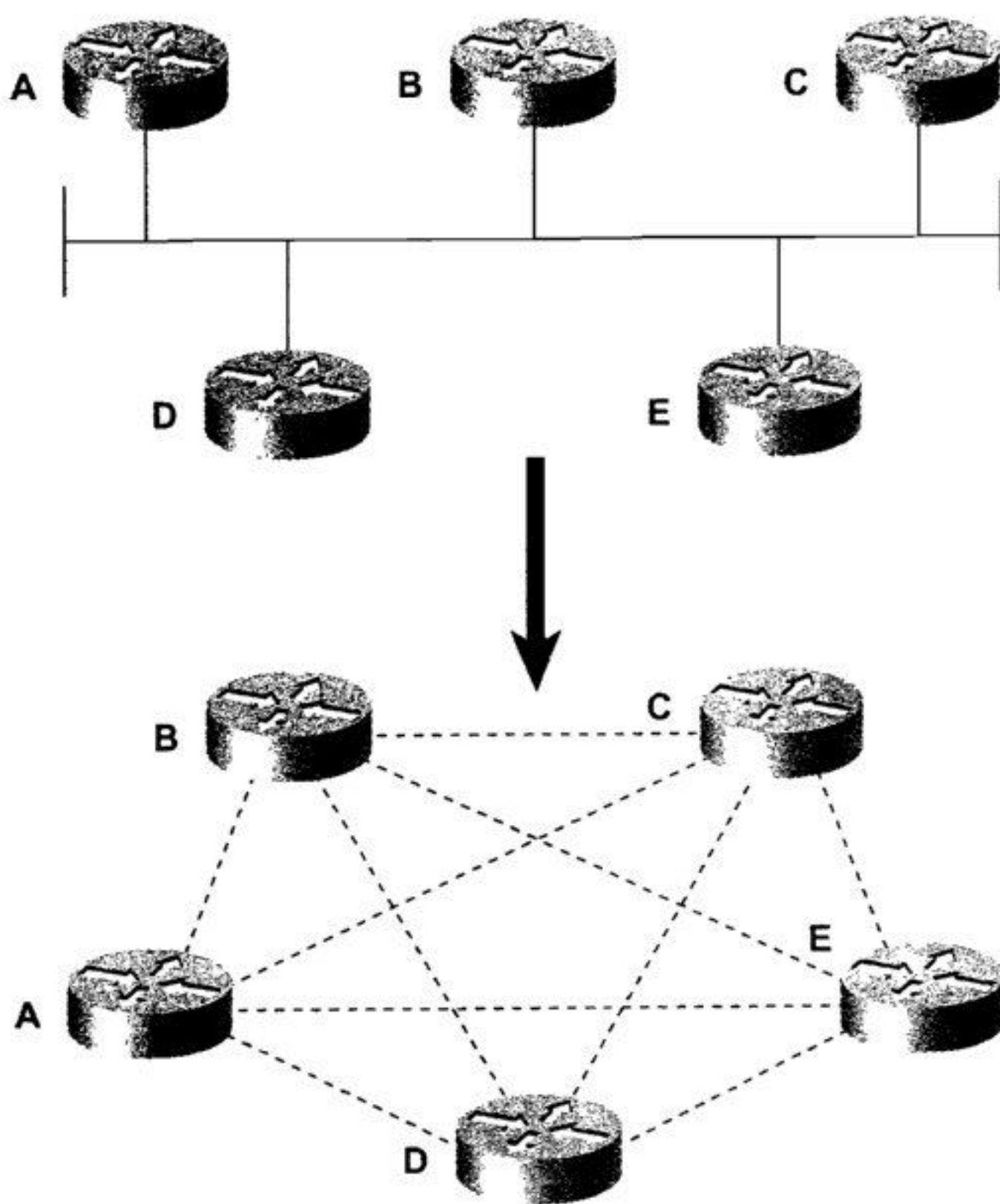


图 9-2 在这个 OSPF 网络上，如果要在每一台路由器和它的邻居路由器之间形成完全网状的 OSPF 邻接关系，那么这 5 台路由器之间将需要形成 10 个邻接关系：这个网络还将产生 25 条 LSA 通告

为了在一个多址网络避免这些问题的发生，可以在多址网络上选举一个指定路由器。这个指定路由器将完成以下工作：

- 描述这个多址网络和该网络上剩下的其他相关路由器；
- 管理这个多址网络上的泛洪过程。

网络本身也将指定路由器看作是网络上的一个“伪节点”，或者是一个虚拟路由器。网络上的每一台路由器都将和这个被描述为伪节点的指定路由器构成一个邻接关系。在这里，只有指定路由器才发送 LSA 到网络中其余的路由器。请记住，一台路由器可能是其中一个与它相连的多址网络的指定路由器，但可能不是其他与它相连的多址网络的指定路由器。换句话说，指定路由器是一个路由器接口的特性，而不是整个路由器的特性。

到目前所描述的为止，可以看出关于指定路由器的一个重要问题是，如果一个指定路由器失效了，就必须选取一个新的指定路由器。同时，网络上的所有路由器也要重新建立新的邻接关系，并且网络上所有的路由器必须根据新选出的指定路由器进行同步它们的网络数据库（邻接关系创建过程中的一部分）。当所有上述的过程发生时，网络将无法有效地传送数据包。

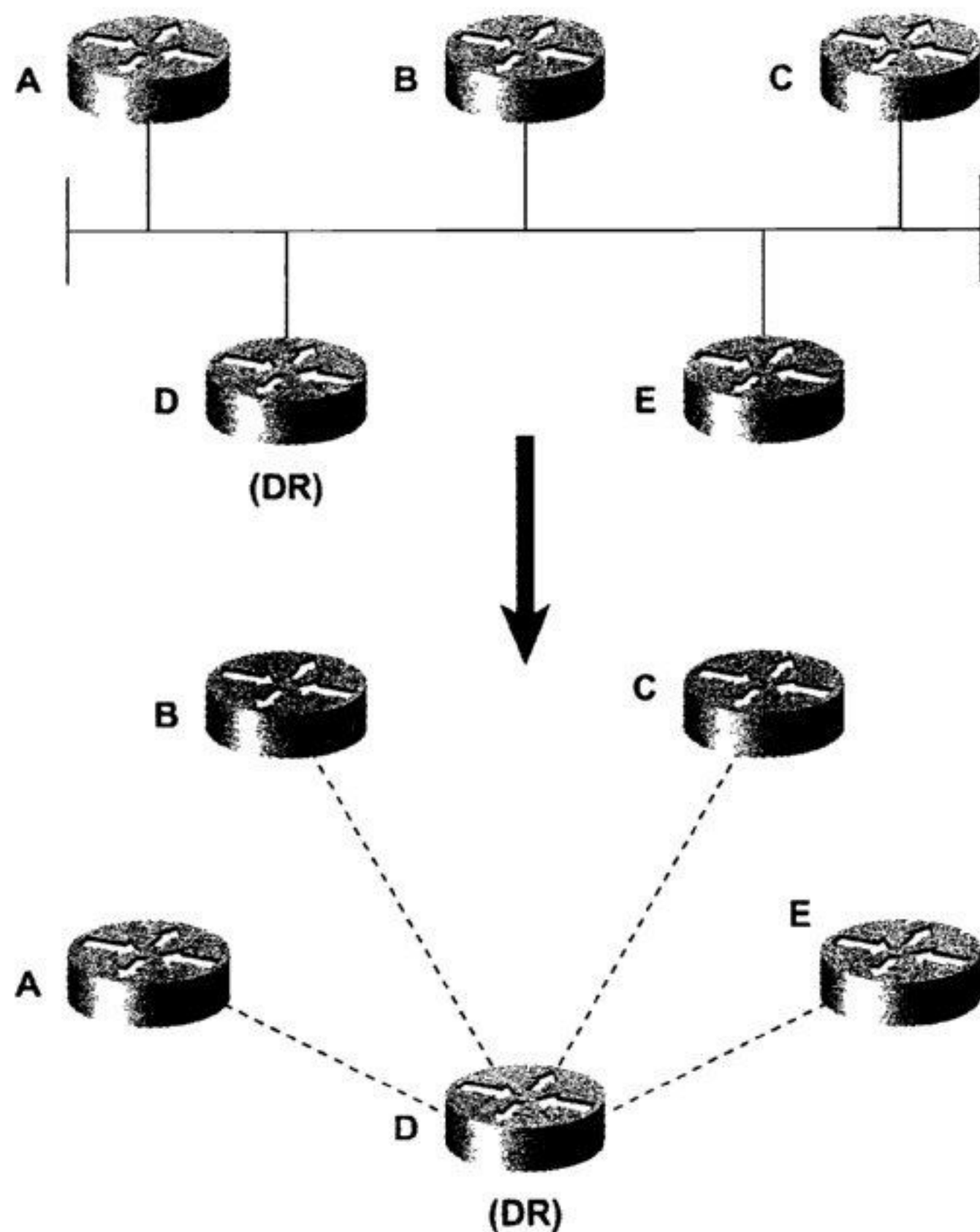


图 9-3 指定路由器描述了一个多址网络。网络上的其他路由器都将和这个指定路由器 DR 构成邻接关系，而不是它们互相之间构成邻接关系

为了避免这个问题，在网络上除了选取指定路由器，还再选取一个备份指定路由器 (Backup Designated Router, BDR)。这样，网络上所有的路由器将和指定路由器 DR 与备份指定路由器 BDR 同时形成邻接关系。DR 和 BDR 之间也将互相形成邻接关系。这时，如果 DR 失效了，BDR 将成为新的 DR。但是由于网络上其余的路由器已经和 BDR 形成了邻接关系，因此网络可以将无法传送数据的影响降低到最小。

DR 和 BDR 的选取是通过一个接口状态机的方式触发的，接口状态机将在后面的小节中讲述。为了能够使选取的处理过程可以进行，需要满足以下一些前提条件：

- 每台路由器的每一个多点访问的接口都有一个路由器的优先级 (Router Priority)，用一个 8 位的无符号整数来表示，大小范围是 0~255。在 Cisco 的路由器上，缺省的优先级是 1。基于每一个多点访问的接口都可以通过命令 `ip ospf priority` 来更改。具有 0 优先级的路由器将不能成为 DR 或者 BDR。
- Hello 报文包含了表示始发路由器指定的路由器优先级的字段，也包含了表示路由器认为可能是 DR 和 BDR 的相关接口的 IP 地址的字段。
- 当一个接口在一个多址网络上开始有效时，它将把它的 DR 和 BDR 的地址设置为 0.0.0.0。同时它也设置等待计时器 (Wait Timer) 的值等于路由器无效时间间隔 (RouterDeadInterval)。
- 在多址网络上已经存在的接口将把 DR 和 BDR 的地址记录入一个接口数据结构表中，接口数据结构表将在后面的章节中讲述。

DR 和 BDR 的选取过程如下描述:

(1) 在路由器和它的邻居路由器之间首先建立成功双向通信 (2-way communication), 接着检查每台邻居路由器发送的 Hello 报文的优先级、DR 和 BDR 等字段。列出所有具有 DR 和 BDR 选取资格的路由器的列表 (也就是说, 路由器的优先级要大于 0, 并且它的邻居状态至少要是 2-way 的); 接着, 所有的路由器将宣称自己是 DR 路由器 (Hello 报文的 DR 字段是它们自身接口的地址); 所有的路由器也将宣称它们自己是 BDR 路由器 (Hello 报文的 BDR 字段是它们自身接口的地址)。除非没有选取资格, 路由器计算时也将在这个具有选取资格路由器的列表中包括它本身。

(2) 从具有选取资格的路由器的列表中, 创建一个还没有宣告为 DR 路由器的所有路由器的子集 (宣告自己为 DR 路由器的路由器不能被选取为 BDR 路由器)。

(3) 如果在这个子集中的一个或者多个邻居路由器, 它们在 Hello 报文的 BDR 字段包含了它们自己的接口地址, 那么具有最高优先级的邻居路由器将被宣告为 BDR 路由器。在优先级相同的条件下, 具有最高的路由器 ID 的邻居路由器将被选作 BDR 路由器。

(4) 如果在这个子集中没有路由器宣称自己是 BDR 路由器, 那么具有最高优先级的邻居路由器将被宣告为 BDR 路由器。在优先级相同的条件下, 具有最高的路由器 ID 的邻居路由器将被选作 BDR 路由器。

(5) 如果一个或多个具有选取资格的路由器在 Hello 报文的 DR 字段包含它们自己的接口地址, 那么具有最高优先级的邻居路由器将被宣告为 DR 路由器。在优先级相同的条件下, 具有最高的路由器 ID 的邻居路由器将被选作 DR 路由器。

(6) 如果没有路由器宣称自己是 DR 路由器, 那么新选取的 BDR 路由器将成为 DR 路由器。

(7) 如果正在执行计算的路由器是新选取的 DR 或 BDR 路由器, 或者它不再是 DR 或 BDR 路由器了, 那么将重复以上的 2~6 步骤。

简单地说, 当一台 OSPF 路由器有效 (active) 启动并去发现它的邻居路由器时, 它将去检查有效的 DR 和 BDR 路由器。如果 DR 和 BDR 路由器存在的话, 这台路由器将接受已经存在的 DR 和 BDR 路由器。如果 BDR 路由器不存在, 将执行一个选取过程, 选出具有最高优先级的路由器作为 BDR 路由器。如果存在多个路由器具有相同的优先级, 那么在数值上具有最高的路由器 ID 的路由器将被选中。如果没有有效的 DR 路由器存在, 那么 BDR 路由器将被推举为 DR 路由器, 然后再执行一个选取过程选取 BDR 路由器。

这里需要注意的是, 路由器的优先级可以影响一个选取过程, 但是它不能强制更换已经有效的 DR 或 BDR 路由器。也就是说, 在已经选取了 DR 和 BDR 路由器后, 如果一台具有更高优先级的路由器变为有效的了, 那么这台新的路由器将不会替换 DR 或 BDR 路由器的任何一个。因此, 在一个多址网络上, 最先初始化启动的两台具有 DR 选取资格的路由器将成为 DR 和 BDR 路由器。

一旦 DR 和 BDR 路由器选取成功, 其他的路由器 (称为 DRothers) 将只和 DR 及 BDR 路由器之间形成邻接关系。所有的路由器将继续以组播方式发送 Hello 报文到 AllSPFRouters (组播地址是 224.0.0.5), 因此它们能够跟踪它们的邻居路由器, 但是 DRothers 路由器只以组播方式发送更新报文到 AllDRouters (组播地址是 224.0.0.6)。只有 DR 和 BDR 路由器去侦听这个地址, 反过来, DR 路由器将使用组播地址 224.0.0.5 泛洪更新报文到 DRothers。

请注意, 如果在一个多址网络上只有惟一的一台具有选取资格的路由器相连, 那么这台

路由器将成为 DR 路由器,而且在这个网络上没有 BDR 路由器。其他所有的路由器都将只和这个 DR 路由器建立邻接关系。如果没有具有选取资格的路由器和一个多址网络相连,那么这个网络上将没有 DR 或者 BDR 路由器,而且也不建立任何邻接关系。在这种情况下,网络上所有路由器的邻居状态都将停留在“2-Way”状态(将在后面的“邻居状态机”一节中阐述)。

关于 DR 和 BDR 路由器所需要完成的更多功能将在随后的章节作更多地完整描述。

4. OSPF 接口

链路状态协议的基本要点是它涉及到了路由器之间的链路和那些链路的状态。在 Hello 报文发送之前,在邻接关系建立之前,以及在 LSA 通告发送之前,一个 OSPF 路由器必须了解它自己的链路情况。OSPF 协议了解链路信息的手段是借助于路由器的接口信息的,所以碰到下面这样的情况就不足为奇了,就是在讲述 OSPF 协议时有时会混用接口和链路这两个术语,而不仔细区分它们含义的不同。本小节将讲述 OSPF 协议接口的数据结构和 OSPF 协议接口的不同状态。

(1) OSPF 接口数据结构

运行 OSPF 协议的路由器将为每一个启动 OSPF 协议的接口维护一个数据结构。如图 9-4 所示,使用 **show ip ospf interface** 命令观察路由器接口的数据结构的内容。

```
Renoir#show ip ospf interface Serial1.738
Serial1.738 is up, line protocol is up
Internet Address 192.168.21.21/30, Area 7
Process ID 1, Router ID 192.168.30.70, Network Type POINT_TO_POINT, Cost: 781
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 192.168.30.77
Message digest authentication enabled
Youngest key id is 10
```

图 9-4 可以使用命令 **show ip ospf interface** 来观察路由器接口相关 OSPF 协议的具体信息。

在这个例子中,接口是和点到点类型的网络相连的

路由器接口的数据结构的信息如下说明:

- **IP Address and Mask (IP 地址和掩码)**——这个信息是路由器接口所配置的 IP 地址和掩码。始发于这个接口的 OSPF 数据包将把这个地址作为源地址。如图 9-4 所示,这个地址/掩码组合是 192.168.21.21/30。
- **Area ID (区域 ID)**——就是接口所在的区域,也就是这个接口所属的网络指定的区域 ID。始发于这个接口的 OSPF 报文将使用这个区域 ID。如图 9-4 所示,这里所显示的接口的区域 ID 是 7。
- **Process ID (进程 ID)**——这个特性是 Cisco 公司特有的属性,不是 OSPF 协议开放标准的一部分。Cisco 的路由器依赖这个特性能够在同一台路由器中运行多个 OSPF 的进程,并且使用这个进程 ID 来区分这些 OSPF 进程。进程 ID 的概念仅在所配置的路由器上有效,而在该路由器之外没有意义。如图 9-4 所示,这里的进程 ID 是 1。
- **Router ID (路由器 ID)**——如图 9-4 所示,这里的路由器 ID 是 192.168.30.70。

- **Network Type (网络类型)**——和这个接口相连的网络的类型：广播型、点到点类型、NBMA、点到多点类型或虚链路等。在图 9-4 中，网络的类型是点到点。¹
- **Cost(代价)**——是指从该接口发送出去的数据包的出站接口代价。链路代价是 OSPF 协议的度量，并使用 16 位的无符号的整数表示，大小范围在 1~65535 之间。Cisco 公司使用的缺省代价是 $10^8/\text{BW}$ ，表示为一个整数，在这里 BW 是指在接口上配置的带宽，而 10^8 是 Cisco 路由器使用的参考带宽。图 9-4 中所示的接口配置了一个 128kbit/s 的带宽（图中没有显示），所以它的代价是 $10^8/128\text{kbit/s}=781$ 。路由器接口的代价值可以通过命令 **ip ospf cost** 来改变，当在一个多家厂商产品的网络环境中配置 Cisco 的路由器时，这个命令变得十分重要。例如，Bay 公司或其他厂商的路由器在其所有的接口上使用的缺省代价是 1（实际上就是把 OSPF 的代价映射为跳数）。如果网络中所有的路由器没有使用同一种计算代价的方式来指定 OSPF 的代价，那么 OSPF 协议将不能正确地进行路由选择。

使用 10^8 作为接口的参考带宽在现代一些带宽高于 100M（例如 OC-3 或吉比特以太网 GE）的网络介质中会产生一个问题。 $10^8/100\text{M}=1$ ，这就意味着更高带宽的传输介质在 OSPF 协议中将会计算出一个小于 1 的分数，这在 OSPF 协议中是不允许的。因此，从 IOS11.2 版本开始，Cisco 就使用命令 **ospf auto-cost reference-bandwidth** 修正了这个问题，这个命令允许管理者更改缺省的参考带宽。

- **InfTransDelay**——这个信息是指 LSA 通告从路由器的接口发送后经历的时间，以秒数计算，当 LSA 通告从路由器接口发出后将会引起这个参数值不断地增大。如图 9-4 所示，在图中它是以 Transmit Delay 来显示的，并且在 Cisco 的路由器上缺省值为 1s。InfTransDelay 可以通过命令 **ip ospf transmit-delay** 来改变。
- **State (状态)**——这个接口的功能状态将在后面的“OSPF 接口状态机”一节中讲述。
- **Router Priority (路由器优先级)**——用来选择 DR 和 BDR 的一个 8 位无符号整数，大小范围是 0~255。在图 9-4 中没有显示路由器的优先级是因为这里的网络类型是点到点的，而在这种网络类型中不需要选取 DR 和 BDR。如图 9-5 所示，它显示了同一台路由器上另一个 OSPF 接口，从图中可以看出，这个接口是和一个广播型网络相连接的，因此，在这个网络上将会选取 DR 和 BDR。在 Cisco 公司的路由器上，路由器的优先级缺省为 1，并且可以通过命令 **ip ospf priority** 来改变。

```
Renoir#show ip ospf interface Ethernet0
Ethernet0 is up, line protocol is up
  Internet Address 192.168.17.73/29, Area 0
  Process ID 1, Router ID 192.168.30.70, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.30.70, Interface address 192.168.17.73
  Backup Designated router (ID) 192.168.30.80, Interface address 192.168.17.74
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.30.80 (Backup Designated Router)
  Message digest authentication enabled
  Youngest key id is 10
```

图 9-5 这里所显示的接口和一个广播型网络相连，并且这台路由器是该广播型网络上的指定路由器 DR

¹ 请注意，在这里和这个接口相连的是帧中继网络。但是由于这是一个点到点的子接口，因此，OSPF 协议使用点到点类型替代了 NBMA 的网络类型。

- **Designated Router (指定路由器)**——对于和路由器接口相连的网络的指定路由器, 路由器将同时记录下指定路由器的 Router ID 和它与这个共享网络相连的接口地址信息。注意, 在图 9-4 中并没有显示出指定路由器, 因为只有在多址网络类型中才会显示出指定路由器。如图 9-5 所示, 这里的指定路由器是 192.168.30.70, 它所连接的接口地址是 192.168.17.73。在图 9-5 中查看一下 Router ID、接口地址和接口状态, 就会发现路由器 Renoir 就是指定路由器。
- **Backup Designated Router (备份指定路由器)**——对于和路由器接口相连的网络的 BDR, 路由器也将同时记录下指定路由器的 Router ID 和它与这个共享网络相连的接口地址信息。如图 9-5 所示, 这里的 BDR 是 192.168.30.80, 而它所连接的接口地址是 192.168.17.74。
- **HelloInterval**——是指在接口上传送两个 Hello 报文之间的周期性间隔时间, 以秒 (s) 来表示。这个周期时间是在从接口发送的 Hello 报文中通告的。在 Cisco 公司的路由器上, 这个周期时间缺省为 10s, 并且可以通过命令 `ip ospf hello-interval` 来改变。如图 9-5 所示, HelloInterval 在图中是通过 Hello 表示的, 并在这里使用了路由器配置的缺省值。
- **RouterDeadInterval**——是指在宣告邻居路由器无效之前, 本地路由器从与一个接口相连的网络上侦听到来自于邻居路由器的一个 Hello 报文所经历的时间, 以秒 (s) 来表示。RouterDeadInterval 是在从接口发送的 Hello 报文中通告的。在 Cisco 公司的路由器上, 这个时间缺省的是 HelloInterval 的 4 倍, 并且可以通过命令 `ip ospf dead-interval` 来改变。如图 9-5 所示, RouterDeadInterval 在图中是通过 Dead 表示的, 并在这里使用了路由器配置的缺省值。
- **Wait Timer (等待计时器)**——在开始选取 DR 和 BDR 之前, 路由器等待邻居路由器的 Hello 报文通告 DR 和 BDR 的时长。等待计数器的时间长度就是 RouterDeadInterval 的时间。在图 9-4 中的等待时间是没有意义的, 因为那个接口是和一点到点的网络相连的, 没有 DR 和 BDR 的选取问题。
- **RxmtInterval**——是指在没有得到确认的情况下, 路由器重传 OSPF 报文将要等待的时间长度, 以秒 (s) 来表示。如图 9-5 所示, 这个时间是通过 retransmit 来表示的, 并在这里使用了 Cisco 路由器缺省配置的时间——5s。路由器接口的 RxmtInterval 可以通过命令 `ip ospf retransmit-interval` 来改变。
- **Hello Timer (Hello 计时器)**——这个计时器的初始值由 HelloInterval 来设置。当它计时超时后, 路由器将从接口上发送出一个 Hello 报文。在图 9-5 中显示了 Hello Timer 将在 3s 后超时。
- **Neighboring Routers (邻居路由器)**——是指和这个接口相连的网络上有效邻居路由器 (这里的有效邻居路由器是指在 RouterDeadInterval 的时间内可以收到来自于它们的 Hello 报文的那些邻居路由器) 的列表。如图 9-6 显示了同一台路由器另一个接口的信息。在这里, 和这个接口相连的网络上应该可以学习到 5 台邻居路由器, 但是只有两台邻居路由器是有邻接关系的 (只有建立邻接关系的邻居路由器的 router ID 才会显示出来)。作为一个 DR 路由器, 在这个网络只能和 DR 与 BDR 路由器建立邻接关系, 这和多址网络中使用 DR 的规则是一致的。


```

Renoir#show ip ospf interface Ethernet1
Ethernet1 is up, line protocol is up
  Internet Address 192.168.32.4/24, Area 78
  Process ID 1, Router ID 192.168.30.70, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 192.168.30.254, Interface address 192.168.32.2
  Backup Designated router (ID) 192.168.30.80, Interface address 192.168.32.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
  Neighbor Count is 5, Adjacent neighbor count is 2
    Adjacent with neighbor 192.168.30.80 (Backup Designated Router)
    Adjacent with neighbor 192.168.30.254 (Designated Router)
  Message digest authentication enabled
  Youngest key id is 10

```

图 9-6 在这个网络上，路由器可以看到 5 台邻居路由器，但是只和 DR 与 BDR 路由器建立了邻接关系

- **AuType**——描述了在网络上使用的认证类型。OSPF 协议认证的类型可以是 Null（没有认证）、简单口令或者加密认证（消息摘要 MD 认证）。在图 9-6 中可以看出这里使用了消息摘要 MD 认证方式。如果使用了 Null 认证方式，在使用命令 **show ip ospf interface** 时将不会显示认证方式和密钥信息。
- **Authentication Key**（认证密钥）——如果在路由器的接口上启用的是简单认证方式，那么认证密钥就是一个 64 位的口令；如果在路由器的接口上启用的是加密认证方式，那么认证密钥就是一个消息摘要密钥。图 9-6 中显示了“youngest key ID”是 10。这里说明了一个事实，就是加密认证允许在路由器的一个接口上配置多个密钥，从而可以保证便捷、安全地改变密钥。

图 9-7 显示了一个和 NBMA 网络相连的接口。注意，在这里 HelloInterval 的值是 NBMA 网络类型缺省值 30s，而 RouterDeadInterval 的值缺省是 4 倍的 HelloInterval。

```

Renoir#show ip ospf interface Serial3
Serial3 is up, line protocol is up
  Internet Address 192.168.16.41/30, Area 0
  Process ID 1, Router ID 192.168.30.105, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 192.168.30.210, Interface address 192.168.16.42
  Backup Designated router (ID) 192.168.30.105, Interface address 192.168.16.41
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:08
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.30.210 (Designated Router)

```

图 9-7 这个接口是和 NBMA 网络类型的帧中继网络相连的，并且它是这个网络的 BDR

花费一些时间来比较一下图 9-4~图 9-7 中所示的信息会很有收获的。所有这 4 个接口都是在同一台路由器上的，但是它们却又在不同类型的网络环境中，从而扮演不同的角色。在每一个实例中所示的接口状态都表明了不同网络上 OSPF 路由器的不同角色。在下一个章节中，将会讲述多种接口状态和接口状态机。

(2) OSPF 接口状态机

一个启用 OSPF 协议的接口在它变成完全有效之前，将会在几种接口状态中间发生转换。这些接口状态是失效、点到点、等待、DR、备份、DRother 和 Loopback 等。

表 9-1

接口状态机的输入事件

输入事件	描 述
IE1	低层协议指明该网络接口是可操作的
IE2	低层协议指明该网络接口是不可操作的
IE3	网络管理系统或低层协议指明该网络接口打环路后是有效的 (looped up)
IE4	网络管理系统或低层协议指明该网络接口打环路后是无效的 (looped down)
IE5	收到 Hello 报文, 在 Hello 报文中, 要么始发邻居路由器把自身作为 BDR 列出, 要么始发邻居路由器把自身作为 DR 列出而不指明 BDR
IE6	等待计时器超时
IE7	路由器被所在的网络选取为 DR 路由器
IE8	路由器被所在的网络选取为 BDR 路由器
IE9	路由器没有被所在的网络选取为 DR 或者 BDR 路由器
IE10	在网络中一组有效的邻居路由器发生了变化。这些变化可能是下列变化之一: (1) 和一个邻居路由器之间建立了双向通信; (2) 和一个邻居路由器之间丢失了双向通信; (3) 收到一个 Hello 报文, 在该 Hello 报文中始发路由器重新把它自身作为 DR 或 BDR 路由器列出; (4) 收到来自于 DR 路由器的 Hello 报文, 在该 Hello 报文中路由器不再把它自身作为 DR 列出; (5) 收到来自于 BDR 路由器的 Hello 报文, 在该 Hello 报文中路由器不再把它自身作为 BDR 列出; (6) 在 RouterDeadInterval 超时后, 还没有从 DR 或 BDR 收到 Hello 报文

5. OSPF 邻居

前面的章节已经讨论了路由器和与之相连的网络之间的关系。虽然一台路由器与其他路由器之间的相互操作和关系在讲述 DR 和 BDR 路由器选取的章节中已经做了一些讨论, 但是在那些章节介绍 DR 路由器选取过程的主要目的还是围绕着建立一种与网络的联系。本节的重点将是讲述网络中的路由器与它的邻接路由器之间的关系。邻居之间建立关联关系的最终目的是为了形成路由器邻居之间的邻接关系, 最终可以顺利地传送路由选择信息。

要成功建立一个邻接关系, 通常需要下面 4 个阶段:

- 邻居路由器发现阶段;
- 双向通信阶段 (Bidirectional Communication) ——当两台互为邻居的路由器在它们的 Hello 报文中都互相列出了它们对方的路由器 ID (Router ID) 时, 路由器就认为双向通信完成了;
- 数据库同步阶段 (Database Synchronization) ——路由器之间将进行交换数据库描述 (Database Description)、链路状态请求和链路状态更新报文 (将在后续的章节讲述) 信息, 以便确保在邻居路由器的链路状态数据库中包含有相同的数据库信息。执行这一步骤的目的是使其中一台邻居路由器成为“主路由器” (master), 而使另一台路由器成为“从路由器” (slave)。“主路由器”将控制数据库描述报文的信息交换;
- 完全邻接阶段 (full adjacency)。

在前面的介绍中, 邻居关系的建立和维持都是通过交换 Hello 报文来实现的。在广播类型和点到点类型的网络里, Hello 报文以组播方式发送给组播地址 AllSPFRouters (224.0.0.5)。在 NBMA 类型、点到多点和虚链路类型的网络里, Hello 报文以单播方式发送给每个单独的邻居路由器。单播的发送方式就意味着, 路由器首先必须知道邻居路由器的存在, 这可以通过手工配置的方式或使用像逆向地址解析协议 (Inverse ARP) 之类的低层协议来发现。关于在这些网络类型中的邻居的配置方法将会在相应的章节中讲述。

在 NBMA 类型的网络中,路由器是每经过 PollInterval 的时间给它邻居状态为 down 的邻居发送一次 Hello 报文,但是在其他的各种网络类型中,路由器都是每经过 HelloInterval 的时间给它的邻居路由器发送一次 Hello 报文。在 Cisco 的路由器中, NBMA 网络里 PollInterval 的缺省值是 60s。

(1) 邻居数据结构

OSPF 路由器在每个 OSPF 接口的接口数据结构中保存的信息可以用来为每一种类型的网络构成 Hello 报文的内容。路由器通过发送包含这些信息的 Hello 报文,可以将自己通告给它的邻居路由器。同样的,对于每一个邻居路由器来说,路由器也将维护一个邻居数据结构表,用来表示从其他路由器学习到的 Hello 报文的信息。路由器和邻居路由器之间的这种双向的信息交换可以认为是一种会话 (conversation)。

如图 9-9 所示,使用命令 **show ip ospf neighbor** 可以观察到路由器单个邻居的邻居数据结构中的一些信息。¹

```
Seurat#show ip ospf neighbor 192.168.30.105
Neighbor 192.168.30.105, interface address 192.168.16.41
  In the area 0 via interface Serial0
  Neighbor priority is 1, State is FULL
  Poll interval 60
  Options 2
  Dead timer due in 00:01:40
```

图 9-9 OSPF 路由器通过邻居数据结构来描述与每个邻居的每次会话

事实上,每个邻居路由器的数据结构中所记录的信息要比图 9-9 中所显示的信息更多。邻居数据结构所含信息如下面所述:

- **Neighbor ID (邻居路由器 ID)**——邻居路由器的标识。如图 9-9 所示,邻居路由器 ID 是 192.168.30.105。
- **Neighbor IP Address (邻居 IP 地址)**——是指和网络相连的邻居路由器的接口 IP 地址。当 OSPF 报文以单播方式发送给邻居路由器时,这个地址就是目的地址。如图 9-9 所示,这里的邻居路由器 IP 地址是 192.168.16.41。
- **Area ID (区域 ID)**——为了使两台路由器能够互为邻居路由器,路由器收到的 Hello 报文中所带的区域 ID 必须和路由器接收接口配置的区域 ID 要匹配。图 9-9 中所示的邻居路由器的区域 ID 是 0 (0.0.0.0)。
- **Interface (接口)**——是指与邻居路由器所在的网络相连的接口,也就是说,邻居路由器可以通过该接口到达。在图 9-9 中,该邻居路由器是通过 S0 接口到达的。
- **Neighbor Priority (邻居优先级)**——这一项表示邻居的路由器优先级,并在邻居路由器的 Hello 报文中通告。所谓的优先级是在 DR 和 BDR 的选取过程使用的。在图 9-9 的邻居路由器的优先级是 1,这是 Cisco 路由器的缺省值。
- **State (状态)**——这一项指的是邻居路由器的状态,邻居的状态将在下面的章节“邻居状态机”中讲述。图 9-9 中邻居的状态为 Full。
- **PollInterval**——这个值只用于 NBMA 网络上相关的邻居路由器。因为在 NBMA 网络上,邻居路由器可能无法自动地被本地路由器发现,因此,如果邻居状态是失效

¹ 请比较这里和图 9-1 中的用法不同。

(Down) 的, 那么路由器将每经过 PollInterval 的时间就会发送一个 Hello 报文给它的邻居路由器。这里的 PollInterval 的时长比 HelloInterval 的时间要长一些。在图 9-9 中所示的 NBMA 网络上的邻居路由器显示出它的 PollInterval 时间是 60s——这是 Cisco 路由器的缺省值。

- **Neighbor Options**(邻居路由器可选项)——这是邻居路由器支持的一些可选的 OSPF 性能。关于这些可选项的介绍将在 Hello 报文格式的讲述中讨论。
- **Inactivity Timer** (失效计时器)——这是一个时长为 RouterDeadInterval (这个参数是在接口数据结构中定义的) 的计时器。无论何时, 只要从邻居路由器收到一个 Hello 报文, 这个计时器就会被重新设置。如果在这个失效计时器超时了还没从邻居路由器那里收到一个 Hello 报文, 那么那个邻居路由器将被宣告为失效 (down) 了。如图 9-9 所示, 在这里失效计时器用 Dead Timer 来表示, 并且将在 100s 后超时。

在邻居数据结构中还有一些信息内容使用命令 **show ip ospf neighbor** 没有显示出来, 这些没有显示的信息也说明如下:

- **Designated Router** (指定路由器)——这个地址包含在邻居路由器发送的 Hello 报文的 DR 字段里面。
- **Backup Designated Router** (备份指定路由器)——这个地址包含在邻居路由器发送的 Hello 报文的 BDR 字段里面。
- **Master/Slave** (主/从)——在 ExStart 状态下, 邻居之间协商的主/从关系将用来控制数据库的同步问题。
- **DD Sequence Number** (数据库描述序列号)——是指当前正在向邻居路由器发送的数据库描述序列号。
- **Last Received Database Description Packet** (最后收到的数据库描述报文)——这个报文记录了初始化位 (Initialize)、后继位 (More) 和主/从位 (Master/Slave), 可选项, 以及最后收到的数据库描述报文的序列号等信息。这个信息可以用来确定下一个数据库描述报文是否是重复的。
- **Link State Retransmission List** (链路状态重传列表)——这是在邻接关系建立后 OSPF 已经进行泛洪 (flood) 但还没有得到确认的 LSA 的列表。当 LSA 通告还没有得到确认或邻接关系被破坏的时候, LSA 通告将每经过 RxmtInterval 的时间就重传一次, 这里的 RxmtInterval 是在接口的数据结构里面定义的。
- **Database Summary List** (数据库摘要列表)——这一项是指在数据库同步期间, 数据库描述报文中向邻居路由器发送的 LSA 列表。当路由器进入到信息交换状态 (Exchange state) 时, 这些 LSA 通告将会构成链路状态数据库。
- **Link State Request List** (链路状态请求列表)——这个列表记录了来自于邻居路由器的数据库描述报文的 LSA 通告, 这些 LSA 通告要比在路由器链路状态数据库中的 LSA 通告更加新。而链路状态请求报文发送给邻居这些 LSA 通告的拷贝。当路由器通过链路状态更新报文收到请求的 LSA 通告时, 请求列表就会减小, 最终将变为空列表。

(2) 邻居状态机

OSPF 路由器需要邻居路由器在几种邻居状态之间转换后 (在邻居数据结构中讲述), 才

能形成邻居之间的完全邻接关系 (Full Adjacent)。

- **失效状态 (Down)** ——这是一个邻居会话的初始状态, 用来指明在最近一个 RouterDeadInterval 的时间内还没有收到来自于邻居路由器的 Hello 报文。除非在 NBMA 网络中的那些邻居路由器, 否则, Hello 报文是不会发送给那些失效的邻居路由器的。在 NBMA 网络的环境中, Hello 报文是每隔 PollInterval 的时间发送一次的。如果一台邻居路由器从其他更高一些的邻居状态转换到了失效状态, 那么路由器将会清空链路状态重传列表、数据库摘要列表和链路状态请求列表。
- **尝试状态 (Attempt)** ——这种状态仅仅适用于 NBMA 网络上的邻居, 在 NBMA 网络上邻居路由器是手工来配置的。当 NBMA 网络上具有 DR 选取资格的路由器和其邻居路由器相连的接口开始变为有效 (Active) 时, 或者当这台路由器成为 DR 或 BDR 时, 这台具有 DR 选取资格的路由器将会把邻居路由器的状态转换到 Attempt 状态。在 Attempt 的状态下, 路由器将使用 HelloInterval 的时间代替 PollInterval 的时间来作为向邻居发送 Hello 报文的时间间隔。
- **初始状态 (Init)** ——这一状态表明在最近的 RouterDeadInterval 时间里路由器收到了来自于邻居路由器的 Hello 报文, 但是双向通信仍然没有建立起来。路由器将会在 Hello 报文的邻居字段中包含这种状态下或更高状态的所有邻居路由器的路由器 ID。
- **双向通信状态 (2-Way)** ——这一状态表明本地路由器已经在来自于邻居路由器的 Hello 报文的邻居字段中看到了它自己的路由器 ID, 这也就意味着, 一个双向通信的会话已经成功建立了。在多址网络中, 邻居路由器必须在这个状态或更高的状态时才能有资格被选作该网络上的 DR 或 BDR。如果在 Init 状态下从邻居路由器那里收到一个数据库描述报文, 也可以引起邻居状态直接转换到 2-Way 状态。
- **信息交换初始状态 (ExStart)** ——在这一状态下, 本地路由器和它的邻居将建立起主/从关系, 并确定数据库描述报文的序列号, 以便为数据库描述报文的信息交换作好准备。在这里, 具有最高的接口地址的邻居路由器将成为“主”路由器。
- **信息交换状态 (Exchange)** ——在这一状态下, 本地路由器将向它的邻居路由器发送可以描述它整个链路状态数据库信息的数据库描述报文。同时, 在这个 Exchange 的状态下, 本地路由器也会发送链路状态请求报文给它的邻居路由器, 用来请求最新的 LSA 通告。
- **信息加载状态 (Loading)** ——在这一状态下, 本地路由器将会向它的邻居路由器发送链路状态请求报文, 用来请求最新的 LSA 通告。虽然在 Exchange 状态下已经发现了这些最新的 LSA 通告, 但是本地路由器还没有收到这些 LSA 通告。
- **完全邻接状态 (Full)** ——在这一状态下, 邻居路由器之间将建立起完全邻接关系, 这种邻接关系出现在路由器 LSA 和网络 LSA 中。

在图 9-10~图 9-12 中, 显示了 OSPF 协议的邻居状态和引起这些邻居状态发生转换的输入事件。在表 9-2 中对这些输入事件作了详细描述, 并在表 9-3 中定义了点。在图 9-10 中显示了从最初的功能状态到最完全的功能状态一步一步向更高状态转换的一般过程。在图 9-11 和图 9-12 中显示了 OSPF 协议邻居状态机的整个转换过程。

(3) 建立一个邻接关系

除非邻居路由器之间 Hello 报文的参数不匹配, 一般情况下, 在点到点、点到多点和虚

链路类型的网络上的邻居路由器之间总是可以形成邻接关系的。而在广播型网络和 NBMA 网络上，将需要选取 DR 和 BDR 路由器，DR 和 BDR 路由器将和所有的邻接路由器形成邻接关系，但是在 DRothers 路由器之间没有邻接关系存在。

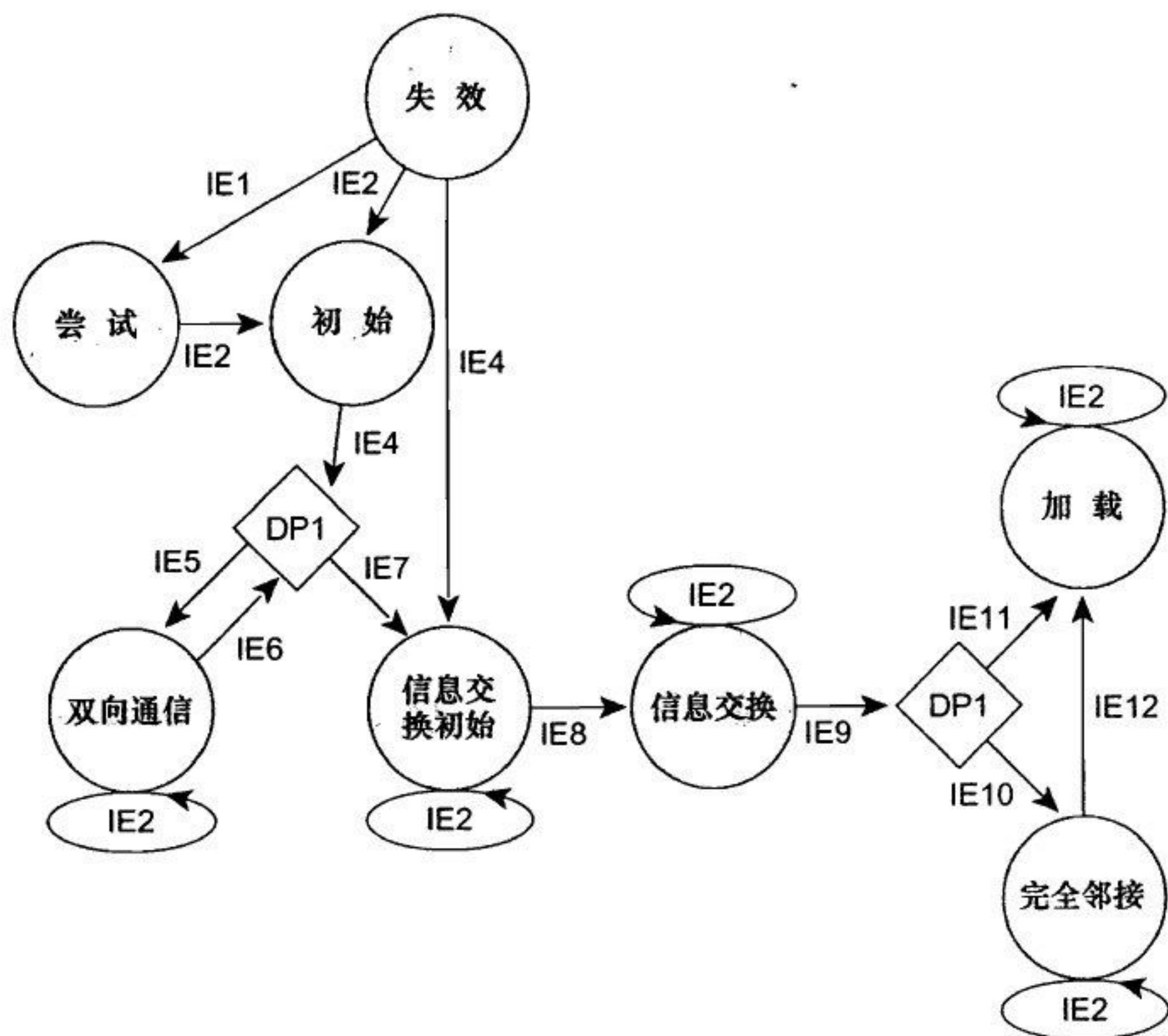


图 9-10 在 OSPF 协议的邻居状态机中，一个邻居路由器从失效状态到完全邻接状态所经过的一系列状态转换

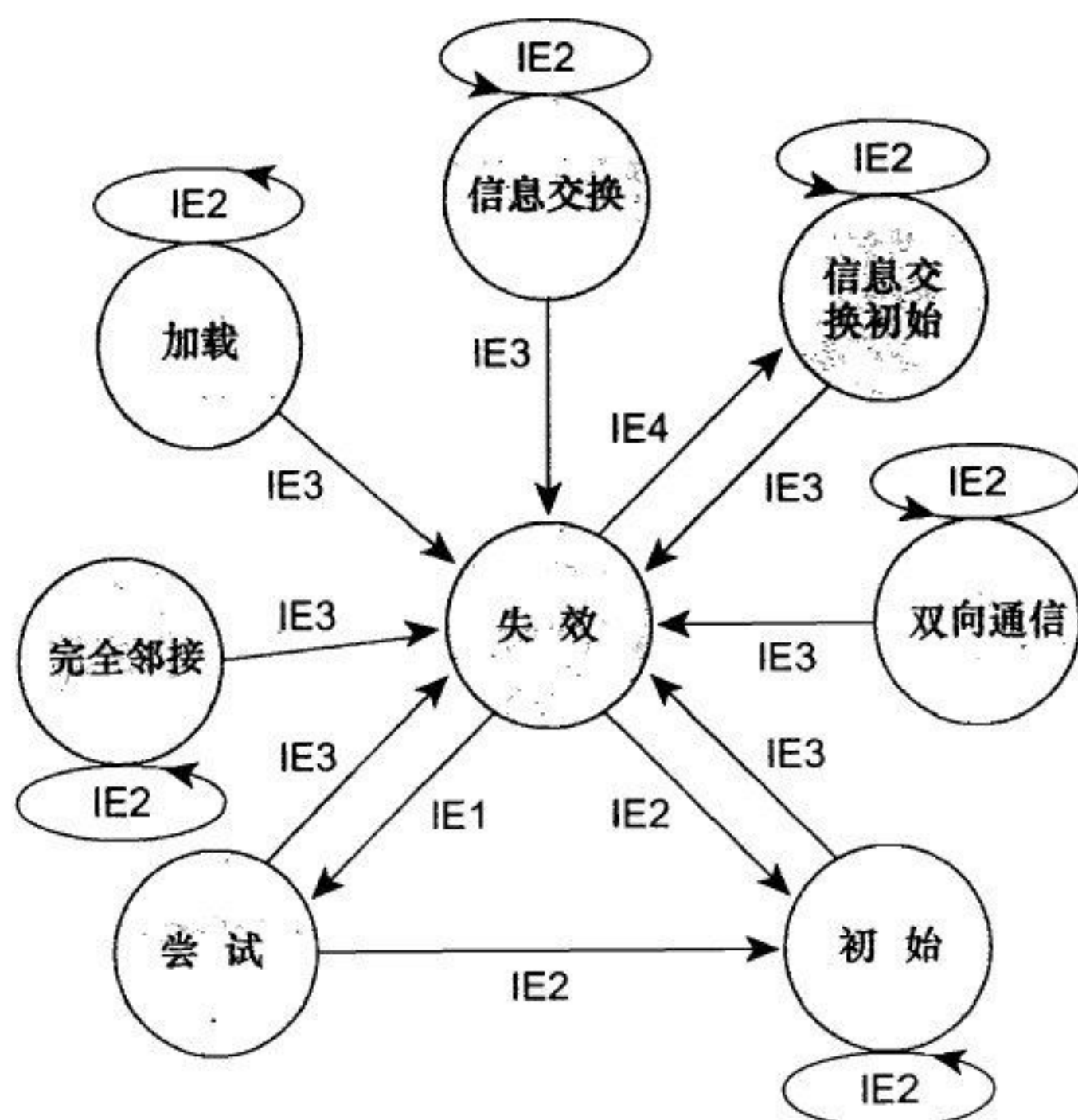


图 9-11 在 OSPF 协议的邻居状态机中，一个邻居路由器从失效状态到初始状态的转换

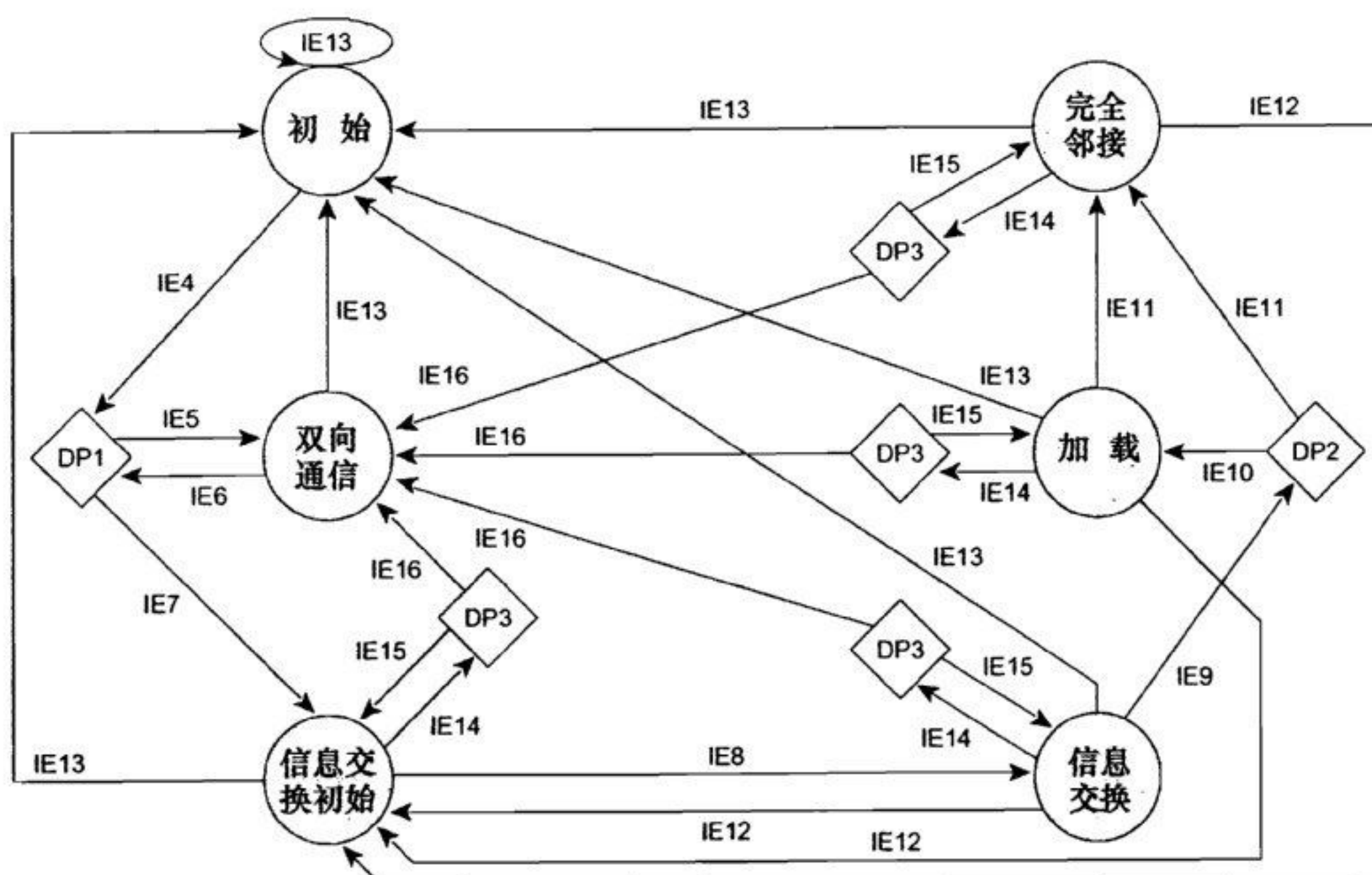


图 9-12 在 OSPF 协议的邻居状态机中, 一个邻居路由器从初始状态到完全邻接状态的转换

表 9-2

图 9-10、图 9-11 和图 9-12 的输入事件

输入事件	描述
IE1	这个输入事件只发生在与 NBMA 网络相连的邻居路由器上, 并可以通过下列所述的情况之一触发该输入事件: (1) 与 NBMA 网络相连的路由器接口开始变为有效 (active), 并且邻居路由器有资格进行 DR 的选取; (2) 本地路由器变为 DR 路由器或者 BDR 路由器, 并且邻居路由器没有资格进行 DR 的选取
IE2	从邻居路由器那里收到一个有效的 Hello 报文
IE3	根据低层的协议、来自于 OSPF 进程本身的明确指令或者无效计时器的超时等影响使邻居路由器不再可达
IE4	本地路由器在邻居路由器发送的 Hello 报文的邻居字段列表中开始看到它自己的路由器 ID, 或者是从邻居路由器收到了数据库描述报文
IE5	邻居路由器不能建立邻接关系
IE6	这个输入事件可以由下面两种情况的任何一个触发: (1) 邻居状态开始转换到 2-way 状态; (2) 接口状态发生变化
IE7	不能和这个邻居路由器形成邻接关系
IE8	已经成功建立主/从关系, 并且已经交换数据库描述序列号
IE9	完成了数据库描述报文的信息交换
IE10	链路状态请求列表非空, 存在要请求的条目
IE11	链路状态请求列表为空
IE12	邻接关系将被中断并接着重新开始。这个输入事件可以由下面几种情况的任何一个触发: (1) 接收到一个数据库描述序列号不匹配的数据库描述报文; (2) 接收到一个所含可选项字段的设置和最后一个数据库描述报文的可选项字段设置不同的数据库描述报文; (3) 接收到一个所含初始状态位 (Init 位) 的设置和最初的报文不同的数据库描述报文; (4) 接收到一个所含 LSA 不在本地路由器的链路状态数据库里的链路状态请求报文
IE13	从邻居路由器收到一个 Hello 报文, 但是这个 Hello 报文的邻居字段中没有列出接收该报文的路由器的路由器 ID
IE14	当接口状态变化时将产生这个输入事件
IE15	与该邻居之间现有的或者形成的邻接关系应该继续
IE16	与该邻居之间现有的或者形成的邻接关系不应该继续

表 9-3

图 9-10~图 9-12 中的判定点

判定点	描 述
DP1	是否应该与这台邻居路由器建立一个邻接关系？如果满足下面所描述的条件中的一个或多个，那么将应该建立邻接关系： (1) 网络类型是点到点的； (2) 网络类型是点到多点的； (3) 网络类型是虚链路； (4) 本地路由器是邻接路由器所在的网络上的 DR； (5) 本地路由器是邻接路由器所在的网络上的 BDR； (6) 邻居路由器是 DR； (7) 邻居路由器是 BDR
DP2	关于这台邻居路由器的链路状态请求列表是否是空的
DP3	与这台邻居路由器之间现有的或者形成的邻接关系是否应该继续

在一个邻接关系的创建过程中，OSPF 协议使用以下 3 种报文类型：

- 数据库描述报文（类型 2）
- 链路状态请求报文（类型 3）
- 链路状态更新报文（类型 4）

这些报文类型的格式将在后续的章节“OSPF 报文格式”中详细讲解。

数据库描述报文对于邻接关系的建立过程来说非常重要。正如它的名字所暗示的，该报文携带了始发路由器的链路状态数据库中的每一个 LSA 通告的一个简要描述。这些描述不是关于 LSA 通告的完整描述，而仅仅是它们的头部——这些信息对于接收路由器判定在它自己的数据库中的 LSA 通告是否是最新的拷贝来说已经是足够的了。另外，在数据库描述报文里面有 3 个标记位用来管理邻接关系的建立过程：

- I 位，或称为初始位（Initial bit），当需要指明所发送的是第一个数据库描述报文时，该位设置为 1；
- M 位，或称为后继位（More bit），当需要指明所发送的还不是最后一个数据库描述报文时，该位设置为 1；
- MS 位，或称为主/从位（Master/Slave bit），当数据库描述报文始发于一个“主”路由器时，该位设置为 1。

当两台邻居路由器在 ExStart 状态开始进行主/从关系协商时，它们都将通过发送一个 MS 位设置为 1 的空的数据库描述报文来宣称自己是“主”路由器。这两个数据库描述报文的数据库描述序列号是由发出这两个报文的路由器根据当时应该顺次使用到的序列号来确定的。具有较低的路由器 ID 的邻居路由器将成为“从”路由器，并且回复一个 MS 位设置为 0 的数据库描述报文——这个数据库描述报文的序列号设置为“主”路由器的序列号。同时，这个数据库描述报文也将是第一个携带 LSA 摘要信息的报文。当主/从关系协商完成后，邻居状态也将转换到 Exchange 状态了。

在 Exchange 状态，邻居路由器开始同步它们的链路状态数据库，同步链路状态数据库的操作是通过描述它们各自的链路状态数据库的所有条目来实现的。数据库摘要列表由路由器的链路状态数据库中所有的 LSA 通告的头部组成，而本地路由器将向它的邻居路由器发送包含这些 LSA 头部列表的数据库描述报文。

如果本地路由器发现它的邻居路由器有一条 LSA 通告不在它自己的链路状态数据库当中，或者邻居路由器含有比已知 LSA 通告更新的拷贝，那么本地路由器将把这条 LSA 放入

它的链路状态请求列表中去。随后,本地路由器将发出一个链路状态请求报文去请求一个关于刚才讨论的这个 LSA 的完整拷贝。链路状态更新报文将会传送这些被请求的 LSA 的信息。当本地路由器收到关于这些被请求的 LSA 之后,它将从自己的链路状态请求列表中删除这些 LSA 的条目。

在更新报文中传送的所有的 LSA 必须单独地进行确认。因此,路由器将把这些传送的 LSA 放入它的链路状态重传列表当中。当这些 LSA 被确认后,路由器就从它的链路状态重传列表中删除它们。LSA 可以通过下面两种方法之一来确认:

- **显式确认 (Explicit Acknowledgment)**——确认收到包含这个 LSA 头部的链路状态确认报文;
- **隐式确认 (Implicit Acknowledgment)**——确认收到包含这个 LSA 的相同实例(没有其他更加新的 LSA)的更新报文。

“主”路由器将控制数据库的同步过程,并确保每次只有一个数据库描述报文是未处理的。当“从”路由器收到一个从“主”路由器发出的数据库描述报文后,“从”路由器将通过发送一个具有相同序列号的数据库描述报文来确认那个报文。如果“主”路由器在 RxmtInterval (这个参数在接口数据结构一节中已经介绍)的时间内没有收到一个关于未处理的数据库描述报文的确认的话,那么“主”路由器将会发送该报文的一个新拷贝。

“从”路由器发送数据库描述报文仅仅用来响应它从“主”路由器那里收到的数据库描述报文。如果它所收到的数据库描述报文具有一个新的序列号,那么“从”路由器将发送一个具有相同序列号的数据库描述报文。如果它所收到的数据库描述报文的序列号和在这之前已确认的数据库描述报文相同,那么这个确认报文就是重发的。

当数据库同步过程完成后,将会出现下面两种状态转换的其中一种:

- 如果链路状态请求列表中仍然还有一些 LSA 条目,那么路由器将把邻居的状态转换到加载 (Loading) 状态;
- 如果链路状态请求列表为空,那么路由器将会把邻居的状态转换到完全邻接 (Full) 状态。

如果“主”路由器已经发送过可以完整地描述它自己的链路状态数据库所必要的所有数据库描述报文,并且从“从”路由器收到一个 M 位设置为 0 的数据库描述报文,那么这时“主”路由器就认为数据库的同步过程完成了。如果“从”路由器接收到一个 M 位设置为 0 的数据库描述报文,并且向“主”路由器发送一个确认的 M 位也设置为 0 的数据库描述报文的话(也就是说,“从”路由器已经完全描述了它自己的链路状态数据库),那么这时“从”路由器就认为数据库的同步过程完成了。由于“从”路由器必须确认每一个收到的数据库描述报文,因此“从”路由器总是最先得知同步过程完成了。

图 9-13 中显示了一个邻接关系的创建过程。这个例子是直接从 RFC2328 中引用而来的。在图 9-13 中演示了链路状态数据库同步过程中的下列步骤:

(1) 在多路访问的网络上,路由器 RT1 变为有效状态,并发送一个 Hello 报文。由于它还没有学习到任何邻居,因而这个 Hello 报文的邻居字段是空的,而 DR 和 BDR 字段设置为 0.0.0.0。

(2) 一旦从路由器 RT1 收到上面的 Hello 报文,路由器 RT2 就会为 RT1 创建一个邻居数据结构,并将 RT1 的状态设置为初始状态 (Init)。路由器 RT2 将发送一个 Hello 报文给路由器 RT1,并在这个 Hello 报文的邻居字段里设置 RT1 的路由器 ID。同样的,作为 DR,路由

器 RT2 也将把 Hello 报文的 DR 字段设置成它自己的接口地址。

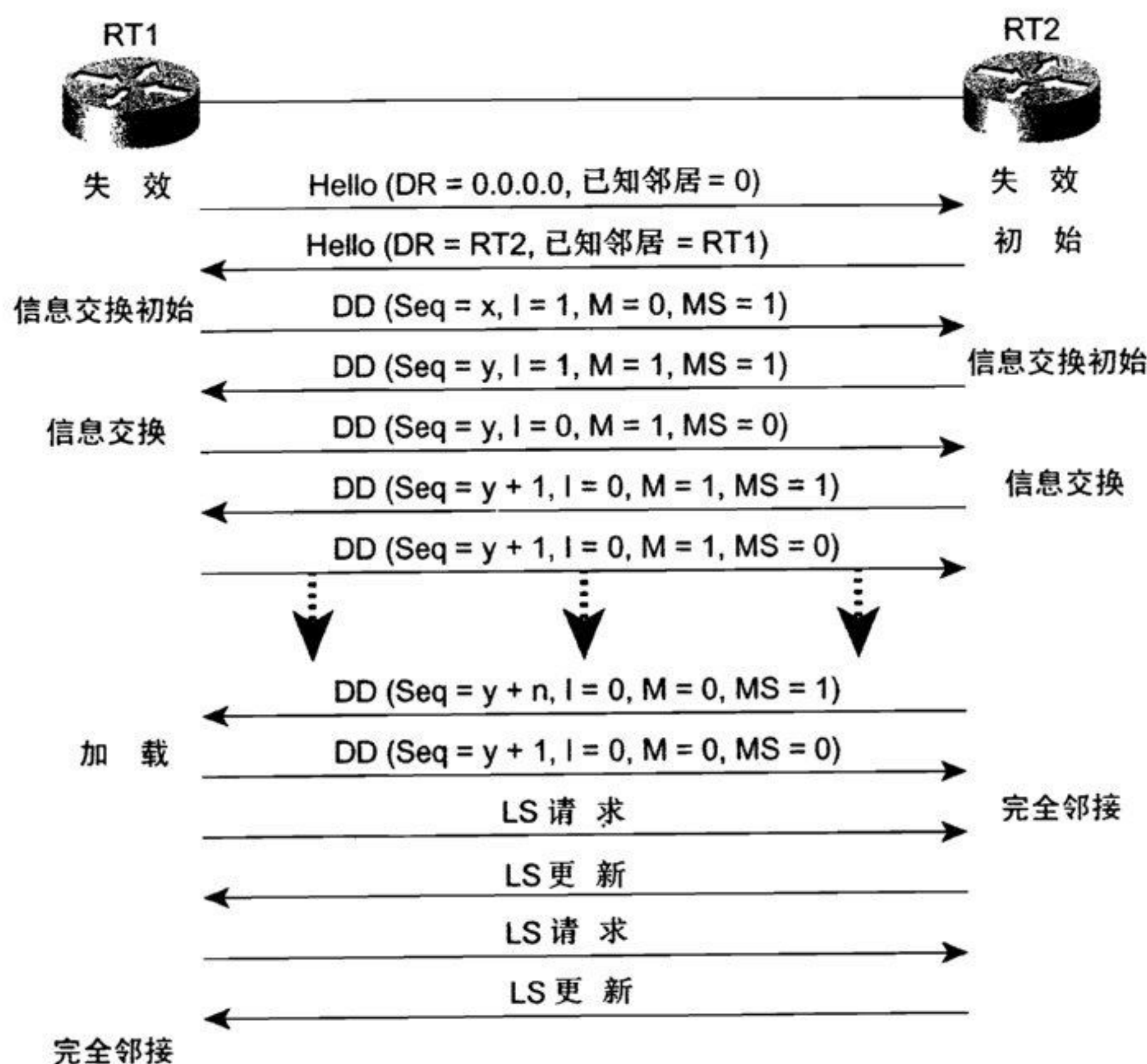


图 9-13 链路状态数据库同步过程和相关的邻居状态

(3) 当路由器 RT1 接收到来自于路由器 RT2 的 Hello 报文，并看到自己的路由器 ID 时 (参见表 9-2 的输入事件 IE4)，RT1 将为路由器 RT2 创建一个邻居数据结构，并把 RT2 的状态设置为 ExStart 状态，以便开始进行主/从关系的协商。接着，路由器 RT1 产生一个空的数据库描述报文 (没有包含 LSA 的摘要)，并把数据库描述的序列号设置为 x。同时设置初始位 (I 位) 来指明这个报文是路由器 RT1 用来进行本次信息交换 (Exchange) 的最初的数据库描述报文，并设置后继位 (M 位) 来指明这个报文不是最后的数据库描述报文，最后还要设置主从位 (MS 位) 来指明路由器 RT1 声称自己是“主”路由器。

(4) 路由器 RT2 一旦收到来自 RT1 的数据库描述报文，就会把 RT1 的状态转换到 ExStart 状态。接着，它将发送一个响应的数据库描述报文，并把这个数据库描述报文的序列号设置为 y。由于路由器 RT2 拥有比 RT1 更高的路由器 ID，因此它将把自己的 MS 位设置为 1。就像最初的那个数据库描述报文一样，这个报文用来进行主从关系协商的，因此也是空的。

(5) 当这两台邻居路由器同意 RT2 是主路由器后，路由器 RT1 就把路由器 RT2 的状态转换为 Exchange 状态。路由器 RT1 将产生一个数据库描述报文，这个报文的序列号使用 RT2 的数据库描述报文的序列号 y，并设置 MS 位为 0 用来指明 RT1 是“从”路由器。同时，该报文将会传送路由器 RT1 的链路状态摘要列表中的 LSA 头部。

(6) 路由器 RT2 一旦收到来自于 RT1 的数据库描述报文，就会把它的邻居状态转换到 Exchange 状态。接着，它将发送一个数据库描述报文，这个报文包含路由器 RT2 自己的链路状态摘要列表中的 LSA 头部，并使它的数据库描述序列号增加到 y+1。

(7) 当路由器 RT1 从路由器 RT2 收到上述的数据库描述报文后, 路由器 RT1 就会发送一个包含相同序列号的确认报文。这个过程将一直延续, 路由器 RT2 发送一个单一的数据库描述报文, 接着等待从 RT1 发出的包含相同序列号的确认报文, 然后 RT2 再发送下一个数据库描述报文, 直到路由器 RT2 发出包含最后一个 LSA 摘要的数据库描述报文, 并把这个报文的 M 位设置为 0。

(8) 收到上述这个报文并且确信它所发出的确认报文包含它自己最后的 LSA 摘要后, 路由器 RT1 就会认为 Exchange 过程已经完成。然而, 路由器 RT1 的链路状态请求列表中还存在 LSA 条目, 因此, 它将转换到信息加载状态 (Loading)。

(9) 当路由器 RT2 收到 RT1 的最后一个数据库描述报文时, 路由器 RT2 将把 RT1 的状态转换为完全邻接状态 (Full), 这是因为在它的链路状态请求列表中已经没有 LSA 条目了。

(10) 路由器 RT1 发送链路状态请求报文, 而路由器 RT2 通过链路状态更新报文发送被请求的 LSA 通告, 这个过程一直持续到路由器 RT1 的链路状态请求列表变成空表。然后, 路由器 RT1 也将把路由器 RT2 的状态转换到完全邻接状态。

这里要注意, 如果路由器的链路状态请求列表中还有 LSA 条目, 它并不需要等待 Loading 状态才发送链路状态请求报文。事实上, 当邻居状态还依旧是 Exchange 状态时路由器就可以发送链路状态请求报文了。因此, 同步过程也许不像图 9-13 中描绘的那么整齐有序, 但是却更有效率。

图 9-14 中显示了使用协议分析仪捕获到的两个邻居路由器之间正在创建邻接关系的过程。虽然链路状态请求报文和链路状态更新报文正在被发送, 但这时两个邻居仍然处于 Exchange 状态, 注意这里的初始位、后继位、主从位和序列号反映了实际网络环境中的处理过程, 这和图 9-13 中描绘的一般过程是一致的。

Number	Packet Type	Router ID	I-bit	M-bit	MS-bit	Sequence Number
8	Hello	192.168.30.70	-	-	-	-
10	Hello	192.168.30.175	-	-	-	-
11	Database Description	192.168.30.70	1	1	1	0x20E0
12	Database Description	192.168.30.175	1	1	1	0xB17
13	Database Description	192.168.30.70	0	1	0	0xB17
14	Database Description	192.168.30.175	0	1	1	0xB18
15	Link State Request	192.168.30.175	-	-	-	-
16	Database Description	192.168.30.70	0	0	0	0xB18
17	Link State Request	192.168.30.70	-	-	-	-
18	Link State Update	192.168.30.70	-	-	-	-
19	Database Description	192.168.30.175	0	0	1	0xB19
20	Link State Update	192.168.30.175	-	-	-	-
21	Database Description	192.168.30.70	0	0	0	0xB19
22	Link State Update	192.168.30.175	-	-	-	-
23	Link State Update	192.168.30.175	-	-	-	-
24	Link State Acknowledgement	192.168.30.175	-	-	-	-
25	Link State Acknowledgement	192.168.30.70	-	-	-	-
26	Link State Update	192.168.30.70	-	-	-	-
28	Link State Update	192.168.30.175	-	-	-	-
30	Link State Acknowledgement	192.168.30.70	-	-	-	-
33	Link State Update	192.168.30.70	-	-	-	-
34	Link State Acknowledgement	192.168.30.175	-	-	-	-
40	Hello	192.168.30.70	-	-	-	-
46	Hello	192.168.30.175	-	-	-	-

图 9-14 这个协议分析仪的捕获界面显示了一个邻接关系正在被创建

在图 9-15 中, 使用调试命令 `debug ip ospf adj` 得到的输出结果显示了图 9-14 中邻接关系正在创建, 这是在其中一台路由器 (路由器 ID 是 192.168.30.175) 上观察到的结果。


```
Degas#debug ip ospf adj
OSPF adjacency events debugging is on
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0x20E0 opt 0x2 flag 0x7 len 32
state INIT
OSPF: 2 Way Communication to 192.168.30.70 on Ethernet0,
state 2WAY
OSPF: Neighbor change Event on interface Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 192.168.30.70
OSPF: Elect DR 192.168.30.175
DR: 192.168.30.175 (Id) BDR: 192.168.30.70 (Id)
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x7 len 32
OSPF: First DBD and we are not SLAVE
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x2 len 92
state EXSTART
OSPF: NBR Negotiation Done. We are the MASTER
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x3 len 72
OSPF: Database request to 192.168.30.70
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x0 len 32
state EXCHANGE
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x1 len 32
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x0 len 32
state EXCHANGE
OSPF: Exchange Done with 192.168.30.70 on Ethernet0
OSPF: Synchronized with 192.168.30.70 on Ethernet0,
state FULL
```

图 9-15 调试命令的输出结果显示了图 9-14 中邻接事件，这是从其中一台路由器上观察到的结果

图 9-14 中，在同步过程结束的时候，可以观察到一系列的链路状态更新报文和链路状态确认报文。这些报文都是 LSA 泛洪过程的一部分，这将在下面一节中讲述。

6. 泛洪 (Flooding)

整个 OSPF 的拓扑图可以描绘成一组互连的路由器或一组互连的节点，这里所说的互连不是指物理的链路而是指逻辑的邻接关系（如图 9-16 所示）。为了使这些节点能够在这个逻辑的拓扑上完全地进行路由选择，每一个节点都必须拥有一个关于这个拓扑结构的相同的拓扑图。这个拓扑图就是拓扑数据库。

OSPF 拓扑数据库更熟知的一个叫法是链路状态数据库。这个数据库由路由器可以接收到的所有 LSA 组成。在拓扑图中发生的一个变化将可以表示为一条或多条 LSA 的变化。泛洪 (Flooding) 过程就是将这些变化的或新的 LSA 发送到整个网络中去，以确保每一个节点的数据库都可以更新，最终保持所有其他节点的数据库的同一性的过程。

泛洪过程将会使用到下面两种类型的 OSPF 报文：

- 链路状态更新报文 (Link State Update packets, 类型 4)；
- 链路状态确认报文 (Link State Acknowledgment packets, 类型 5)。

正如图 9-17 中所显示的那样，每一个链路状态更新报文和确认报文都可以携带多个 LSA。虽然 LSA 本身是泛洪到整个互联网络的，但是更新报文和确认报文却只在具有邻接关系的两个节点之间传送。

在点到点的网络中，路由器是以组播方式将更新报文发送到组播地址 AllSPFRouters (224.0.0.5) 的。在点到多点和虚链路的网络上，路由器是以单播方式将更新报文发送到邻接邻居的接口地址的。

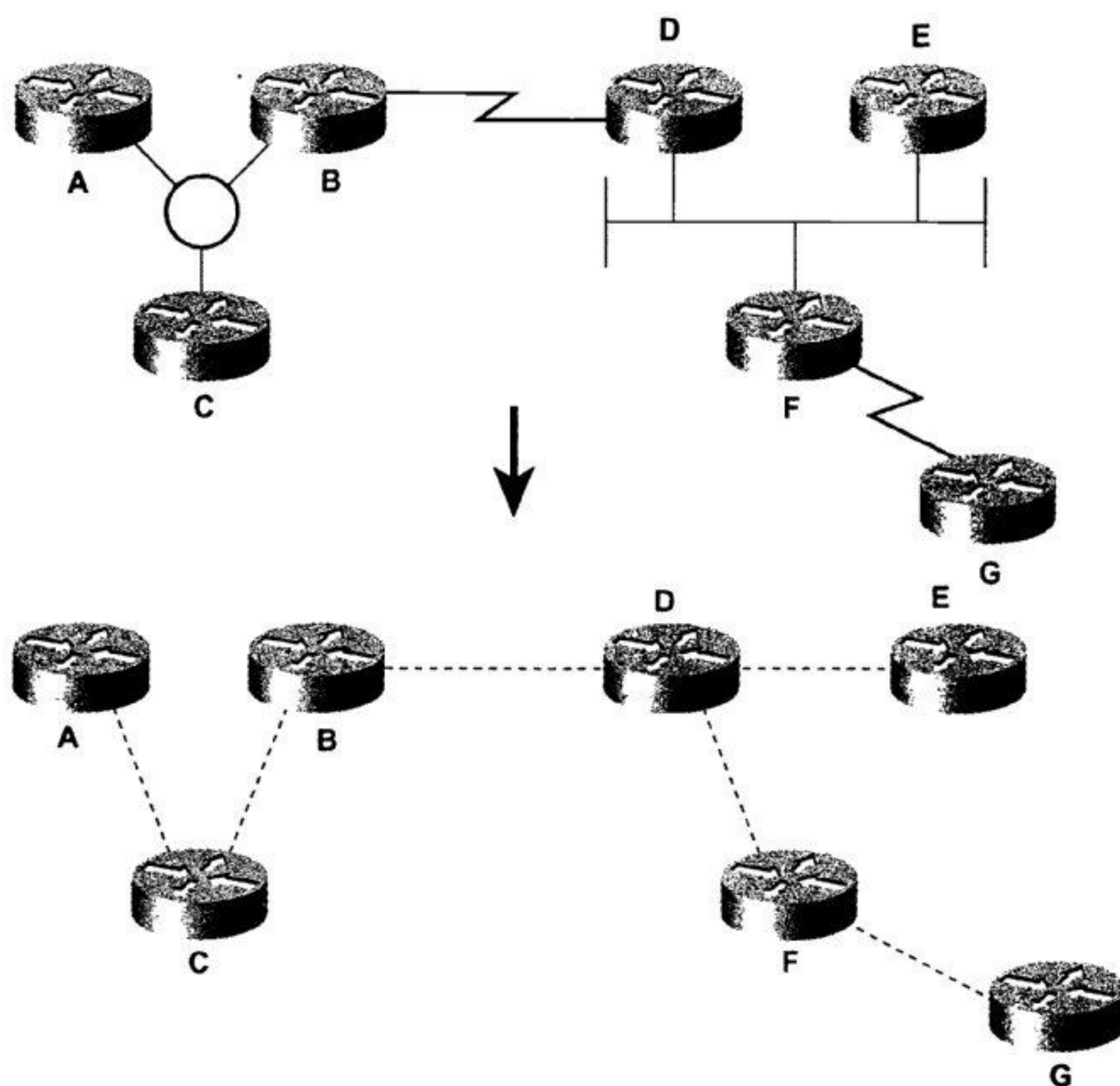


图 9-16 OSPF 协议把一组通过数据链路相连的路由器看作是一组逻辑上通过邻接关系相连的节点



图 9-17 LSA 可以在链路状态更新报文里面发送，从而穿过节点之间的邻接

在广播型的网络上，DRothers 路由器只能和 DR 与 BDR 路由器形成邻接关系，因此，更新报文将发送到组播地址 AllDRouters (224.0.0.6)。相应地，DR 路由器也将以组播方式发送包含 LSA 的更新报文到网络上所有与之建立邻接关系的路由器，这里使用的组播地址是 AllSPFRouters。接着，所有的路由器将从它们所有其他的接口上泛洪出去 LSA 通告（如图 9-18 所示）。虽然 BDR 路由器也使用组播方式收到和记录了来自于 DRothers 路由器的 LSA 通告，但是它不会再重复泛洪或者确认这些 LSA，除非 DR 路由器失效了它才会这么做。在 NBMA 网络上存在着同样的 DR/BDR 的功能特性，只是 LSA 通告是以单播方式从 DRothers 路由器发送给 DR 和 BDR 路由器的，并且 DR 路由器也是以单播方式发送这个 LSA 的拷贝到所有与之建立邻接关系的邻居路由器的。

因为完全相同的链路状态数据库信息是正确操作 OSPF 最基本的前提，因此 LSA 的泛洪必须是可靠的。发送出 LSA 的路由器必须要确认它们发出的 LSA 是否被成功接收了，而接收 LSA 的路由器也必须确认它们正在接收的 LSA 信息是正确的。

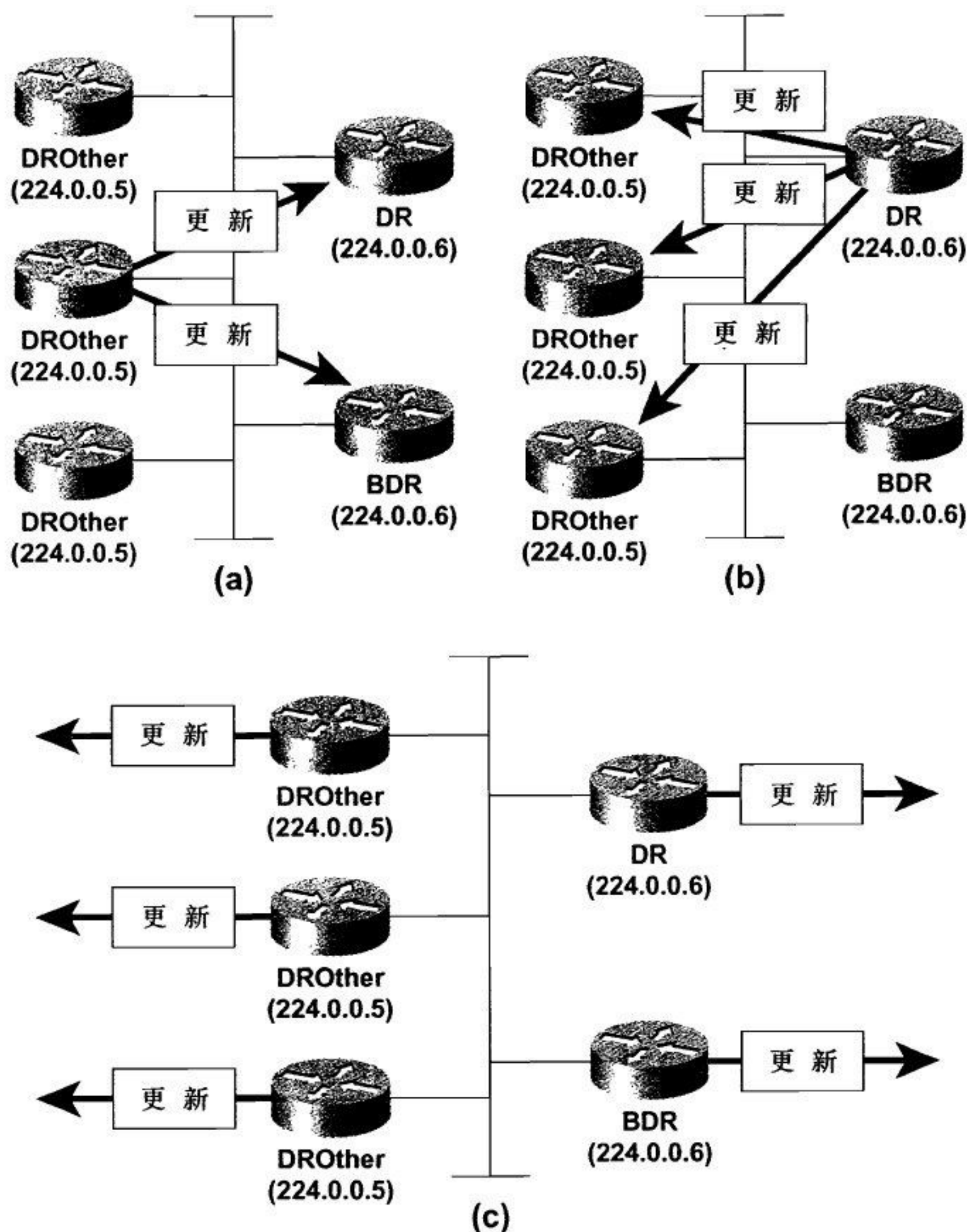


图 9-18 在广播型网络上, DROthers 路由器只向 DR 和 BDR 路由器发送 LSA (a); 而 DR 路由器将再把这个 LSA 泛洪到所有的与之有邻接关系的邻居路由器 (b); 接着, 所有的路由器在它们其他所有的接口上泛洪这个 LSA (c)

(1) 可靠的泛洪: 确认

对于可靠的泛洪来讲, 每一个单独传送的 LSA 都必须被确认。在这里, 确认可以有隐式确认 (Implicit Acknowledgment) 或显式确认 (Explicit Acknowledgment) 两种方式来完成。

邻居路由器可以通过向始发更新报文的路由器回送包含那个 LSA 的复制信息的更新报文, 来作为对所收到的 LSA 的隐式确认。在一些情况下, 隐式确认方式比显式确认方式更有效率。例如, 当邻居路由器正打算向始发路由器发送更新报文的时候。

邻居路由器的显式确认是指通过发送一个链路状态确认报文来确认收到的 LSA 的方式。而且, 可以使用单个链路状态确认报文多个 LSA 通告。这个链路状态确认报文不需要携带完整的 LSA 信息, 而只是需要携带 LSA 的头部就足以完全识别这些 LSA 了。

当一台路由器开始发送一个 LSA 时, 会把这个 LSA 的一个拷贝放进它所发送的每个邻居的链路状态重传列表中。这个 LSA 通告每隔 RxmtInterval 的时间重传一次, 一直到该 LSA 得到确认, 或者一直到这个邻接关系中断。不论是哪一种网络类型, 包含重传的链路状态更

新报文总是以单播方式发送的。

确认可以是有时延的 (delayed) 或直接 (direct) 的。通过延迟一个确认的方法, 更多的 LSA 通告可以通过单个链路状态确认报文来确认。在一个广播型的网络上, 来自于多个邻居路由器的 LSA 可以由单个组播的链路状态确认报文来确认。一个被延迟的确认报文的延迟时间必须小于 RxmtInterval 的时长, 从而避免不必要的报文重传。一般的情况下, 在不同的网络类型上使用于链路状态更新报文的单播/组播地址约定也可以适用于链路状态确认。

直接的确认总是立即发送并且是单播方式发送的。直接的确认将在出现下面的两种情况下发送:

- 从邻居路由器收到了重复的 LSA, 可能表明邻居还没有收到这个 LSA 的一个确认;
- LSA 的老化时间 (Age) 达到最大生存时间 (MaxAge, 将在下一节介绍) 了, 说明在接收路由器的链路状态数据库里已经没有这个 LSA 的实例 (instance)。

(2) 可靠的泛洪: 序列号、校验和、老化时间

每一个 LSA 都包含 3 个值用来确保在每个数据库中保存的 LSA 是最新的。这 3 个数值是序列号、校验和以及老化时间 (age)。

OSPF 协议使用线性的序列号空间 (在第 4 章“动态路由选择协议”中已经讲述) 和 32 位有符号的序列号, 这里序列号的大小范围从 InitialSequenceNumber (0x80000001) 到 MaxSequenceNumber (0x7fffffff)。当一台路由器始发一条 LSA 通告时, 它将设置这个 LSA 的序列号为 InitialSequenceNumber。每次这台路由器产生了这个 LSA 的一个新实例 (Instance) 时, 该路由器就会将它的序列号增加 1。

如果当前 LSA 的序列号是最大值 MaxSequenceNumber 并且又必须创建这个 LSA 的一个新实例时, 这台路由器就必须开始从所有的数据库中清除老的 LSA。这一操作是通过设置现有 LSA 的年龄或老化时间 (Age) 为最大生存时间 (MaxAge, 将在后面的章节介绍) 并且重新泛洪它到所有的邻接节点来实现的。一旦所有的邻接的邻居路由器确认过这个“提前老化”的 LSA 后, 也就可以泛洪这个 LSA 的一个含有 InitialSequenceNumber 序列号的新实例了。

校验和是一个使用 Fletcher 算法计算得到的 16 位整数。¹这个校验和的计算除了 Age 字段 (因为这个 age 字段在 LSA 从一个节点到另一个节点时都会发生变化, 因此如果校验和也计算这个字段的话, 将在每一个节点上都需要重新计算校验和) 外, 将覆盖整个 LSA 报文。驻留在链路状态数据库中的每个 LSA 的校验和每 5min 也将检验一次, 以便确保这个 LSA 在数据库中没有被破坏。

老化时间 (age) 是一个用来指明 LSA 的生存时间的 16 位无符号整数, 以秒为单位计, 大小范围是 0~3600 (1h, 也就是最大生存时间)。一台路由器在始发一个 LSA 时, 它就把老化时间设置为 0。而当泛洪的 LSA 经过一台路由器时, LSA 的老化时间就会增加一个由 InfTransDelay 设定的秒数。在 Cisco 的路由器中, InfTransDelay 设定的缺省值为 1s, 这个数值可以通过命令 **ip ospf transmit-delay** 来改变。当 LSA 驻留在路由器的数据库中时, LSA 的老化时间同样也会增大。

当一条 LSA 通告的老化时间达到最大生存时间时, LSA 将被重新泛洪, 并且随后会从路由器的数据库中清除该条 LSA。当一台路由器需要从所有路由器的数据库中清除一条 LSA

¹ Alex McKenzie, "ISO Transport Protocol Specification ISO DP 8073," RFC 905, April 1984, Annex B.

时, 它会提前把这条 LSA 的老化时间设置为最大生存时间并重新泛洪这个 LSA。在这里, 只有始发这条 LSA 的路由器才可以提前使这条 LSA 老化。

图 9-19 中显示了一个链路状态数据库的部分信息, 从图中可以观察到每一条 LSA 的老化时间、序列号和校验和。有关链路状态数据库和不同类型的 LSA 的详细讨论将在本章后面的“链路状态数据库”一节中介绍。

```
Manet#show ip ospf database
```

```
OSPF Router with ID (192.168.30.43) (Process ID 1)
```

```
Router Link States (Area 3)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link Count
192.168.30.13	192.168.30.13	910	0x80000F29	0xA94E	2
192.168.30.23	192.168.30.23	1334	0x80000F55	0x8D53	3
192.168.30.30	192.168.30.30	327	0x800011CA	0x523	8
192.168.30.33	192.168.30.33	70	0x80000AF4	0x94DD	3
192.168.30.43	192.168.30.43	1697	0x80000F2F	0x1DA1	2

图 9-19 在链路状态数据库中记录了每一条 LSA 的老化时间、序列号和校验和。老化时间是以秒来计算的

当收到某条相同的 LSA 的多个实例时, 路由器将通过下面的算法来确定哪个是最新的 LSA 实例:

- a. 比较 LSA 实例的序列号。拥有最大的序列号的 LSA 就是最新的 LSA;
- b. 如果 LSA 实例的序列号相同, 那么将会比较它们的校验和。拥有最大的无符号校验和的 LSA 就是最新的 LSA;
- c. 如果 LSA 实例的校验和也相同, 那么将进一步比较它们的老化时间。如果只有一条 LSA 拥有大小为最大生存时间的老化时间, 那么就认为这条 LSA 是最新的 LSA; 还有一个比较是:
- d. 如果这些 LSA 的老化时间之间的差别多于 15min (称做 MaxAgeDiff), 那么拥有较小的老化时间的 LSA 将是最新的 LSA;
- e. 如果上述的条件都无法区分最新的 LSA, 那么这两个 LSA 就被认为是相同的。

9.1.2 区域 (Area)

到目前为止, 读者应该对 OSPF 协议有一定的了解了。OSPF 协议由于使用了多个数据库和复杂的算法, 因而相比前面几章介绍的路由选择协议而言, 它将会耗费路由器更多的内存和更多的 CPU 处理。当互联网络的规模不断增大时, 这些对路由器的性能要求就会显得比较重要甚至达到了路由器性能的极限。另一方面, 虽然 LSA 的泛洪比 RIP 协议和 IGRP 协议中周期性的、全路由选择表的更新更加有效率, 但是对于一个大型的互联网络来说, 它依然给大量数据链路带来了无法承受的负担。SPF 算法本身并没有特别的解决办法。像 LSA 的泛洪和数据库的维护等这些相关的处理仍然大大加重了 CPU 的负担。

OSPF 协议可以利用区域的概念来缩小这些不利的影响。在 OSPF 协议的环境下, 区域 (Area) 是一组逻辑上的 OSPF 路由器和链路, 它可以有效地把一个 OSPF 域分割成几个子域 (如图 9-20 所示)。在一个区域内的路由器将不需要了解它们所在区域外部的拓扑细节。在这

种环境下:

- 路由器仅仅需要和它所在区域的其他路由器具有相同的链路状态数据库, 而没有必要和整个互连网络内的所有路由器共享相同的链路状态数据库。因此, 在这种情况下, 链路状态数据库大小的缩减就降低了对路由器内存的消耗。
- 链路状态数据库的减小也就意味着处理较少的 LSA 通告, 从而也就降低了对路由器 CPU 的消耗。
- 由于链路状态数据库只需要在一个区域内进行维护, 因此, 大量的 LSA 泛洪也就被限制在一个区域里面了。

区域是通过一个 32 位的区域 ID (Area ID) 来识别的。正如图 9-20 所显示的, 区域 ID 可以表示成一个十进制的数字, 也可以表示成一个点分十进制的数字。在 Cisco 的路由器中这两种表示方式都可以使用。到底选用哪一种格式来标识一个具体的区域 ID, 通常是根据使用的方便性来选择。例如, 区域 0 和区域 0.0.0.0 的使用效果是相同的, 还有区域 16 和 0.0.0.16, 区域 271 和 0.0.1.15 等等。在上述的这些实例当中, 我们可能应该首先选用十进制的表示方式。然而, 如果要在区域 3232243229 和区域 192.168.30.29 两种表示方式中选择一种格式的话, 那么后面一种格式可能是比较好的一种选择。

对于和区域相关的通信量定义了下面 3 种通信量的类型:

- **域内通信量 (Intra-Area Traffic)** ——是指由在单个域内的路由器之间交换的数据包构成的通信量;
- **域间通信量 (Inter-Area Traffic)** ——是指由在不同区域的路由器之间交换的数据包构成的通信量;

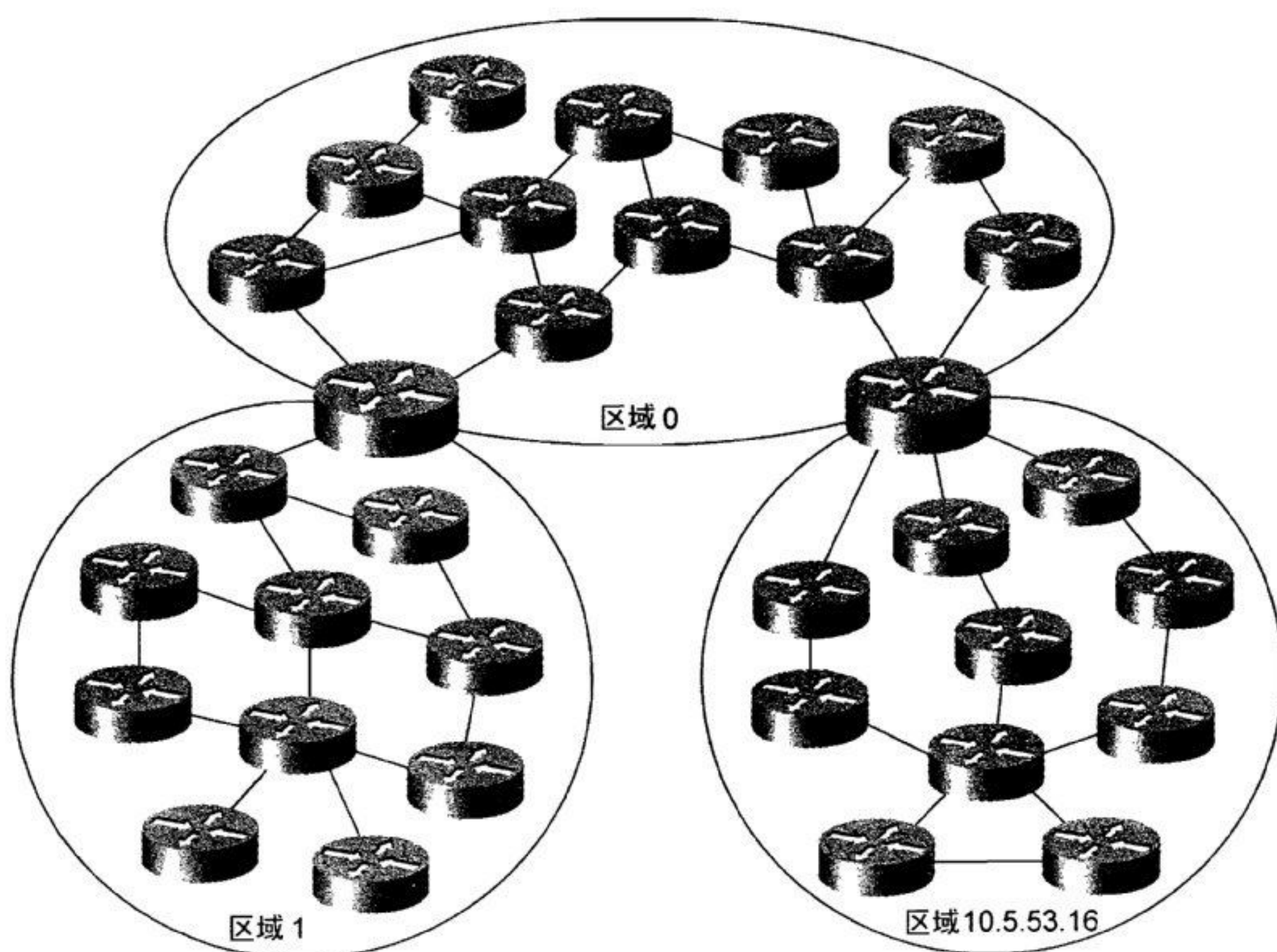


图 9-20 OSPF 区域是一组逻辑上的 OSPF 路由器。每一个区域都是通过它自己的链路状态数据库来描述的, 而且每台路由器也都只需要维护路由器本身所在的区域的链路状态数据库

- 外部通信量 (External Traffic) ——是指由 OSPF 域内的路由器和另一个自主系统内的路由器之间交换的数据包构成的通信量。

区域 0 (或者区域 0.0.0.0) 是为骨干域保留的区域 ID 号。骨干区域 (Backbone Area) 的任务是汇总每一个区域的网络拓扑路由到其他所有的区域。正是由于这个原因, 所有的域间通信量都必须通过骨干区域, 非骨干区域之间不能直接交换数据包。

另外还有一种特殊的区域类型是末梢区域 (Stub Area)。因为在讲述 LSA 的多种类型之前很难描述清楚末梢区域的概念, 因此, 本章将安排在“链路状态数据库”一节中再介绍这个主题。

大多数 OSPF 协议的设计者对于单个区域所能支持的路由器的最大数量都有一个个人认为较适当的粗略的经验值。单个区域所支持的路由器最大数量的范围大约是 30~200。但是, 在一个区域内实际加入的路由器数量要比单个区域所能容纳的路由器最大数量小一些。这是因为还有更为重要的一些因素影响这个数量, 诸如一个区域内链路的数量, 网络拓扑的稳定性, 路由器的内存和 CPU 性能, 路由汇总的有效使用和注入到这个区域的汇总 LSA 的数量等等。正是由于这些因素, 有时在一些区域里包含 25 台路由器可能都已经显得比较多了, 而在另一些区域内却可以容纳多于 500 台的路由器。

只使用单个区域来设计一个小型的 OSPF 网络是非常合理的。不论区域数量的多少, 如果一个区域的数据链路非常之少, 以至于没有冗余链路存在的话, 那么一些潜在的故障就会发生了。如果这样一个区域被分割开来, 那么就有可能使网络的通信服务中断。被分割的区域 (或称为分段区域, Partitioned Area) 将在后面一个小节中更详细地介绍。

1. 路由器的类型

路由器也像通信量一样可以被分成和区域相关的几个类型。所有的 OSPF 路由器都是下面 4 个路由器类型中的一个, 如图 9-21 所示。

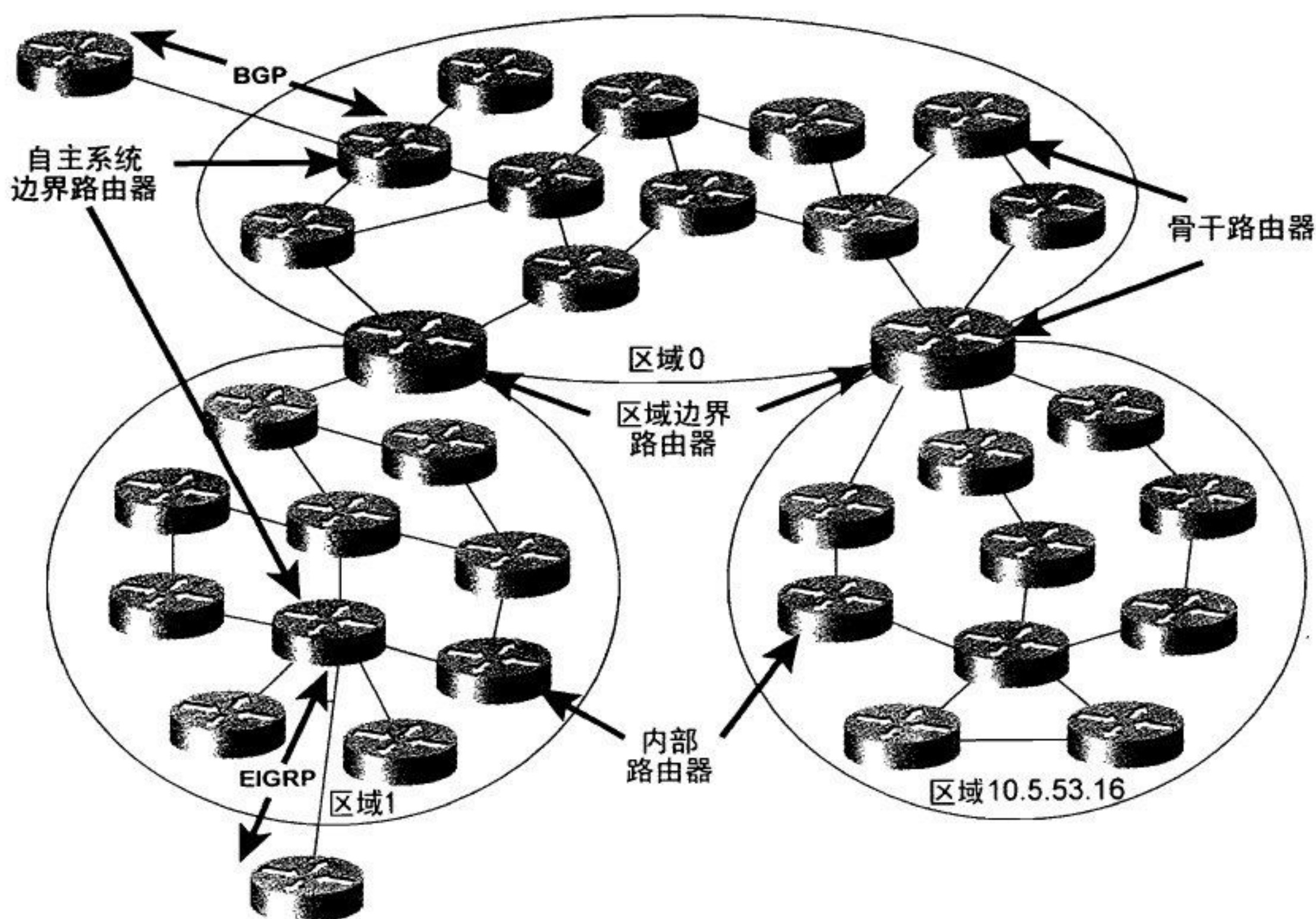


图 9-21 所有的 OSPF 路由器都可以被归类到下面 4 种路由器类型之一——内部路由器、骨干路由器、区域边界路由器 (ABR) 或自主系统边界路由器 (ASBR)。注意前 3 种路由器类型的任何一种也都可能成为一个 ASBR

- **内部路由器 (Internal Router)** ——是指所有接口都属于同一个区域的路由器。
- **区域边界路由器 (Area Border Routers ABR)** ——是指连接一个或者多个区域到骨干区域的路由器, 并且这些路由器会作为域间通信量的路由网关。因而, ABR 路由器总是至少有一个接口是属于骨干区域的, 而且必须为每一个与之相连的区域维护不同的链路状态数据库。正因为这个原因, ABR 路由器通常需要比一般的内部路由器更多的内存和更高性能的路由处理器。ABR 路由器将会汇总与它相连的区域的拓扑信息给骨干区域, 然后将这些汇总信息传送给其他的区域。
- **骨干路由器 (Backbone Router)** ——是指至少有一个接口是和骨干区域相连的路由器。这个定义意味着 ABR 路由器也可以是骨干路由器, 但是, 如图 9-21 中显示, 并不是所有的骨干路由器都是 ABR 路由器。另外, 如果一个内部路由器的所有接口都属于区域 0, 那么这个内部路由器也是一个骨干路由器。
- **自主系统边界路由器 (Autonomous System Boundary Router, ASBR)** ——可以认为是 OSPF 域外部的通信量进入 OSPF 域的网关路由器, 也就是说, ASBR 路由器是用来把其他路由选择协议 (例如, 图 9-21 中显示的 BGP 协议和 EIGRP 协议进程) 学习到的路由通过路由选择重分配的方式注入到 OSPF 域的路由器。一个 ASBR 路由器可以是位于 OSPF 域的自主系统内部的任何路由器, 它可以是一台内部路由器、骨干路由器或者 ABR 路由器。

2. 分段区域

分段区域 (Partitioned Area) 是指一个区域由于链路的失效而使这个区域的一个部分和其他部分隔离开来的情形。如果一个非骨干的区域变成分段区域, 并且在这个分段区域的任何一段区域里的所有路由器当中都还能发现一个 ABR 路由器, 如图 9-22 所示, 那么这个分段区域将不会产生中断通信服务的情况。骨干区域仅仅会把这个分段区域看作两个单独的区域。但是, 从这个分段区域的一段区域到另一段区域的域内通信量将变为域间通信量了, 这些通信量将通过骨干区域而绕开一下这个分段区域。这里要注意, 分段区域和孤立区域 (Isolated Area) 是不同的, 孤立区域没有链路路径和互连网络相连。

如果一个骨干区域本身变成了分段区域, 那么将会带来更加麻烦的问题。如图 9-23 中所示, 一个分段的骨干区域将把原来的骨干区域隔离成两个部分区域, 并在这两个部分区域上创建两个单独的 OSPF 域。

如图 9-24 所示, 图中显示了一些更好的区域设计方法。在区域 0 和区域 2 之间设计了两条数据链路连接, 这样任何一条链路失效了都不会使它们变成分段的区域。但是, 区域 2 也有一个设计缺陷就是如果 ABR 路由器失效了, 那么这个区域就会被孤立了。区域 3 使用了两台 ABR 路由器, 在这里, 任何单条链路的失效或单个 ABR 的失效都不会隔离这个区域的任何部分。

3. 虚链路

虚链路 (Virtual Link) 是指一条通过一个非骨干区域连接到骨干区域的链路。虚链路主要应用于以下几种目的:

- 通过一个非骨干区域连接一个区域到骨干区域 (如图 9-25 所示);
- 通过一个非骨干区域连接一个分段的骨干区域两边的部分区域 (如图 9-26 所示)。

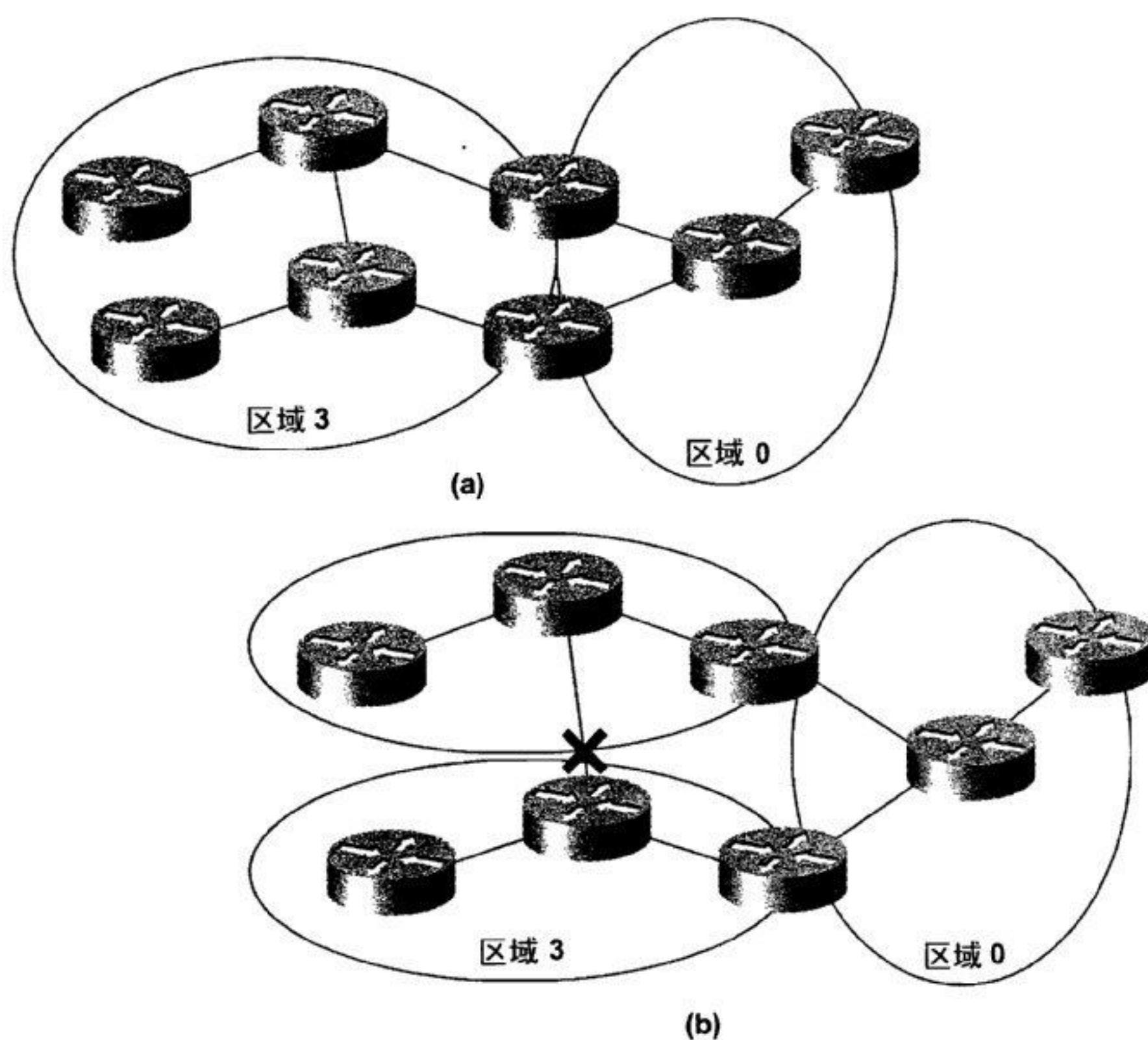


图 9-22 (a) 区域 3 通过两台 ABR 路由器和骨干区域 (区域 0) 相连。(b) 区域 3 的一条链路失效了将会创建一个分段区域, 但是区域 3 内的所有路由器都仍然可以到达一个 ABR 路由器。在这种情况下, 数据的通信量仍然可以在这个分段区域的两边之间进行转发

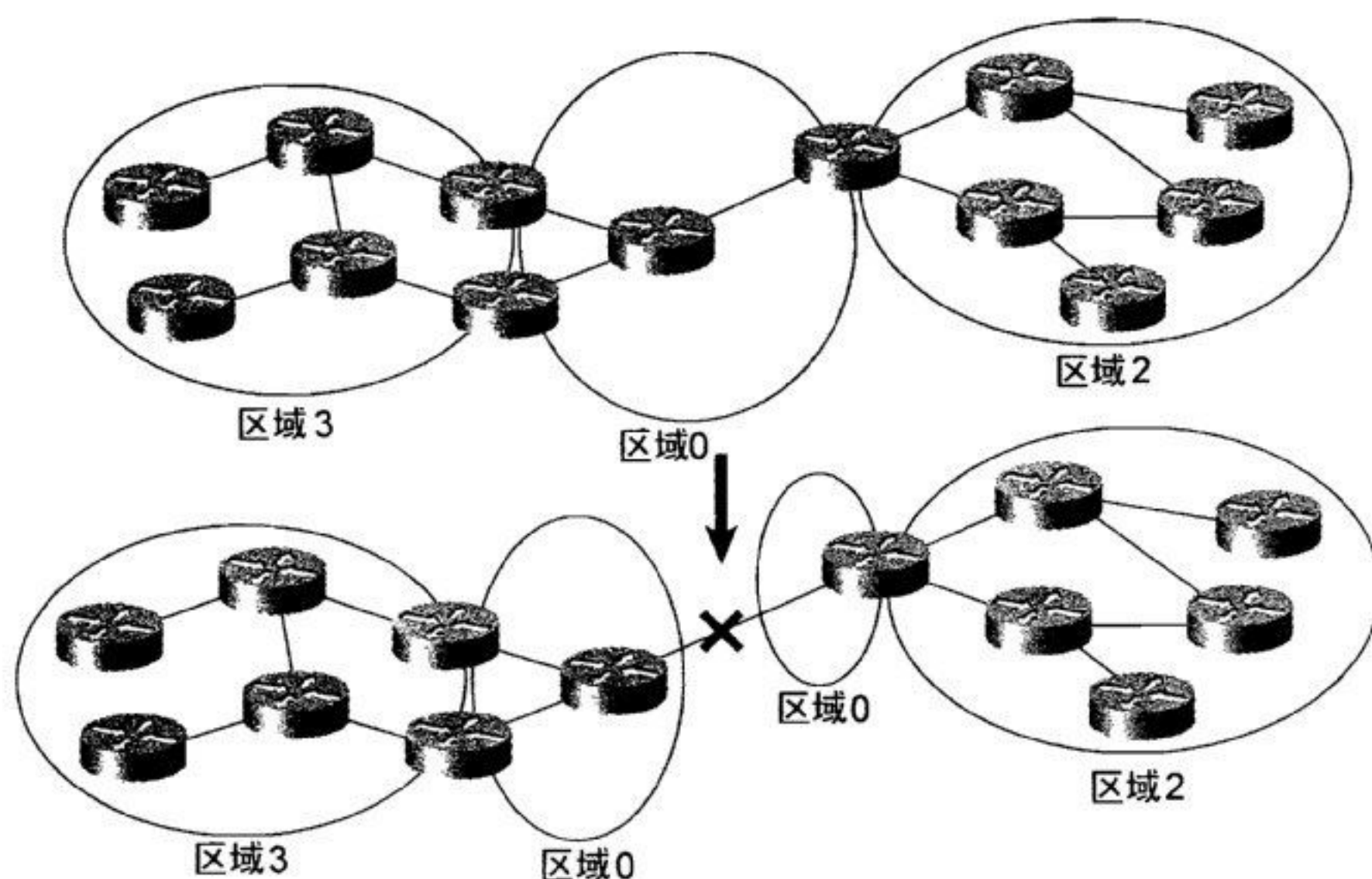


图 9-23 如果一个骨干区域变成分段的区域, 那么这个分段的骨干区域的每一边和与之相连的区域都将和另外一边的部分隔离开来

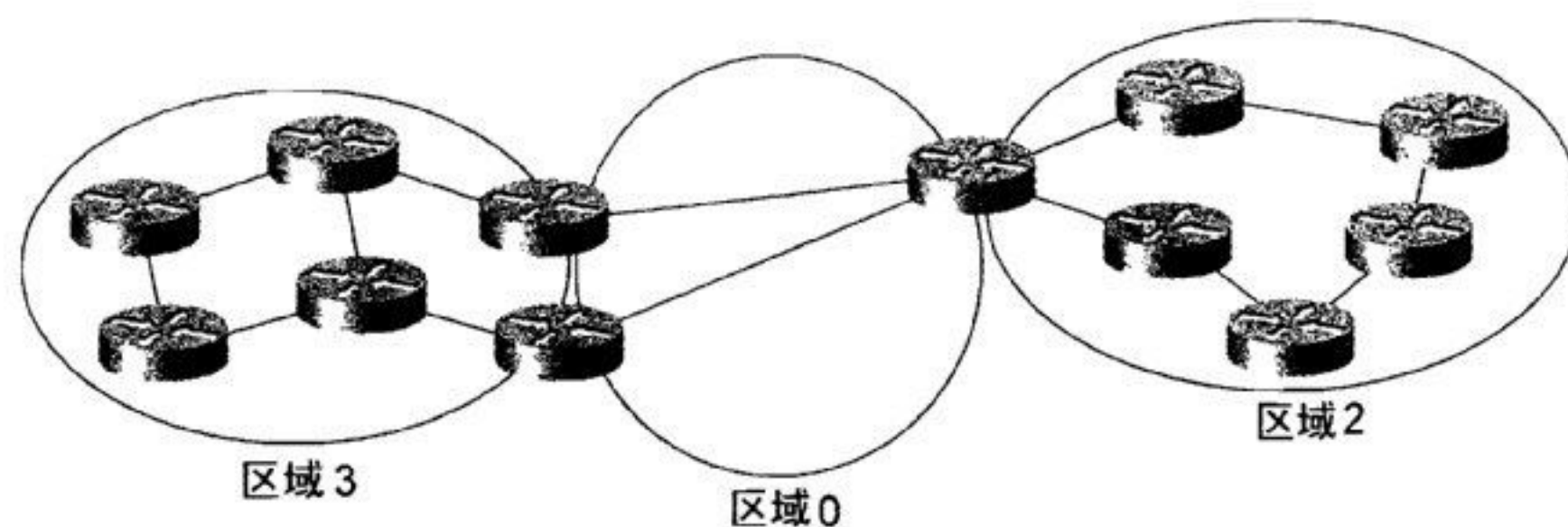


图 9-24 在区域 0 和区域 2 之间，单条链路的失效不会隔离这个区域。在区域 3 中，单条链路或单个 ABR 路由器的失效也不会隔离区域 3

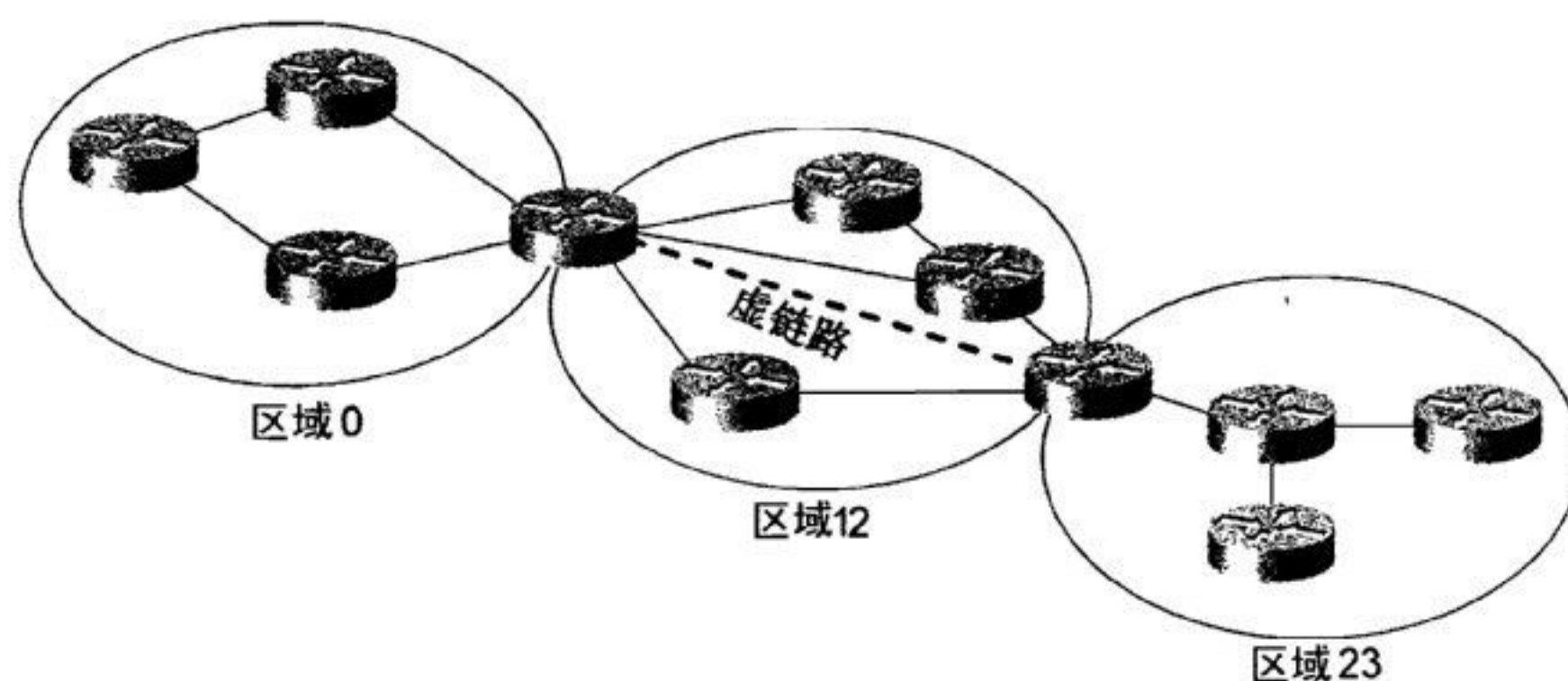


图 9-25 一条虚链路用来把区域 23 经由区域 12 连接到骨干区域

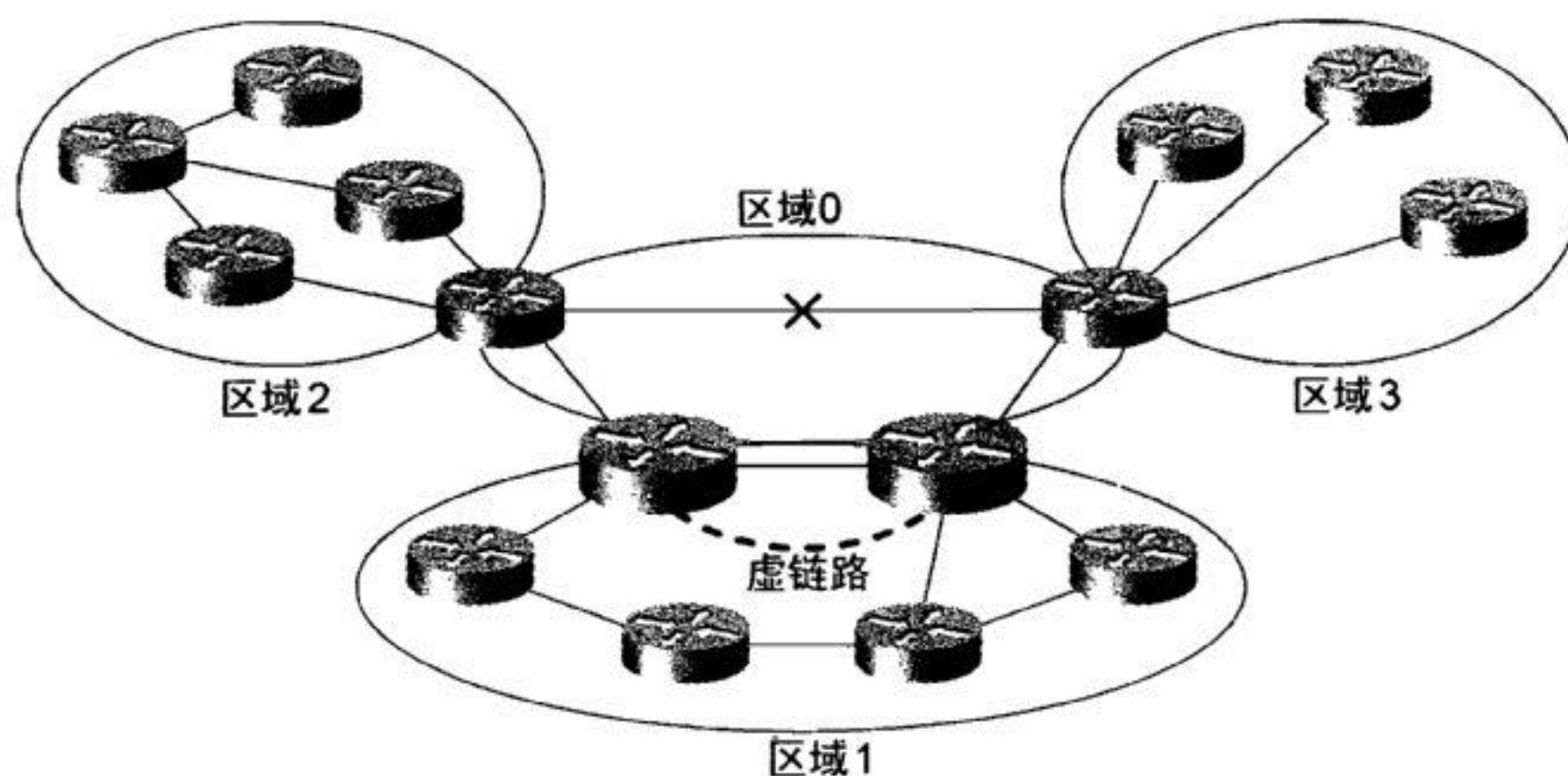


图 9-26 一条虚链路穿过一个非骨干区域重新连接一个分段的骨干区域

在这两个例子中，虚链路和具体的物理链路没有关系。虚链路事实上是一个逻辑通道 (tunnel)，数据包可以通过选择最优的路由路径从一端到达另一端。

在配置虚链路的时候，有几条相关的规则，说明如下：

- 虚链路必须配置在两台 ABR 路由器之间；
- 配置了虚链路所经过的区域必须拥有全部的路由选择信息，这样的区域又被称为传送区域 (Transit Area)；
- 传送区域不能是一个末梢区域。

正如前面所提及的, OSPF 协议也把虚链路归类作为一个网络类型。更特别的, 虚链路可以看作是在两台 ABR 路由器之间的一个无编码的——也就是说是无编址地址的链路, 并且它是属于骨干区域的。这些 ABR 路由器之间虽然没有物理的数据链路相连, 但是它们可以看作是通过它们之间的虚链路逻辑上虚拟连接的邻居。在每一个 ABR 路由器的路由选择表中, 当发现有到达邻居的 ABR 路由器的路由时, 虚链路将转换到完全可操作的点到点接口状态。这条虚链路的代价就是到达它的邻居路由器的路由的代价。当接口状态变为点到点状态时, 一个邻接关系将通过这条虚链路建立成功。

显然, 虚链路的存在增加了互联网络的复杂程度, 而且使故障的排除更加困难。因此, 应该最好避免使用虚链路, 而应该在区域上, 特别是骨干区域上设计冗余链路来确保防止分段区域的产生。当有两个或多个互联网络要合并时, 预先要制定好充分的计划, 以便确保那些没有直连链路到达骨干区域的区域不被遗漏。

如果配置了一条虚链路, 设计者应该仅仅把它用来作为修复无法避免的网络拓扑问题的一个临时手段。虚链路可以看作是一个标明互联网络的某个部分是否需要重新设计的标志。事实上, 永久虚链路的存在总是一个设计比较糟糕的互联网络的标志。

9.1.3 链路状态数据库

一台路由器中所有有效的 LSA 通告都被存放在它的链路状态数据库当中的。正确的 LSA 通告将可以描述出一个 OSPF 区域网络拓扑的结构。因为一个区域中的每一台路由器都要利用这个数据库的信息来计算它自己的最短路径树, 因此, 所有区域数据库的统一性对于正确的路由选择来说就变得十分重要。

如图 9-27 所示, 要观察一个链路状态数据库中的所有 LSA 的列表可以通过命令 **show ip ospf database** 来实现。图中所显示的列表并不是数据库中存储的关于每个 LSA 的全部信息, 而仅仅是这些 LSA 的头部信息。这里要注意, 这个数据库如果是包含多个区域的 LSA 的信息的, 那么就表明这个路由器是 ABR 路由器。

图 9-27 中的大多数条目出于简化的目的已经被删除了, 真实的链路状态数据库包含了 1 445 个 LSA 条目和 4 个区域, 如图 9-28 所示。

正如早前在“可靠的泛洪: 序列号、校验和、老化时间”一节中所提及的, 当 LSA 通告驻留在路由器的链路状态数据库中的时候, 它们的老化时间是增大的。如果这些 LSA 通告达到了最大生存时间 (1h), 那么它们将从 OSPF 域中清除掉。这就意味着, 在这里必须有一个机制来防止正常的 LSA 通告达到最大生存时间而被清除掉。这个机制就是链路状态重刷新 (Link State Refresh)。每隔 30min (这个时间称为 LSRefreshTime) 始发这条 LSA 通告的路由器就将泛洪这条 LSA 的一个新拷贝, 并将它的序列号增加 1, 老化时间设置为 0。其他的 OSPF 路由器一旦收到这个新拷贝, 就会用这个新拷贝替换该条 LSA 通告原来的拷贝, 并且使这个新的拷贝的老化时间开始增加。

虽然链路状态重刷新的机制是用来确保每条 LSA 通告的活动状态的, 但是, 它还带来一个额外的好处是, 任何一个在路由器的链路状态数据库中可能已经被破坏的 LSA 通告都可以被正常 LSA 通告新刷新的拷贝来替换。

由于每一个 LSA 通告都与一个独自的重刷新计时器相关联, 这意味着每 30min, LSA 通告的 LSRefreshTime 将不会一下子都突然超时, 从而重新泛洪所有的 LSA 通告。作为替换做

法, 重新泛洪将在一个半随机的模式 (semirandom pattern) 下传播出去。这种方法带来的问题是每一个单独的 LSA 通告都会在它增加的 LSRefreshTime 超时的时候被重新泛洪, 这使链路带宽的使用没有效率。更新报文只能携带一些, 甚至单个 LSA 通告传送。

```
Homer#show ip ospf database

      OSPF Router with ID (192.168.30.50) (Process ID 1)

      Router Link States (Area 0)

Link ID        ADV Router    Age      Seq#          Checksum      Link count
192.168.30.10  192.168.30.10 1010     0x80001416    0xA818        3
192.168.30.20  192.168.30.20 677      0x800013C9    0xDE18        3
192.168.30.70  192.168.30.70 857      0x80001448    0xFD79        3
192.168.30.80  192.168.30.80 1010     0x800014D1    0xEB5C        5

      Net Link States (Area 0)

Link ID        ADV Router    Age      Seq#          Checksum
192.168.17.18  192.168.30.20 677      0x800001AD    0x849A
192.168.17.34  192.168.30.60 695      0x800003E2    0x4619
192.168.17.58  192.168.30.40 579      0x8000113C    0xF0D
192.168.17.73  192.168.30.70 857      0x8000044F    0xB0E7

      Summary Net Link States (Area 0)

Link ID        ADV Router    Age      Seq#          Checksum
172.16.121.0   192.168.30.60 421      0x8000009F    0xD52
172.16.121.0   192.168.30.70 656      0x8000037F    0x86A
10.63.65.0     192.168.30.10 983      0x80000004    0x1EAA
10.63.65.0     192.168.30.80 962      0x80000004    0x780A

      Summary ASB Link States (Area 0)

Link ID        ADV Router    Age      Seq#          Checksum
192.168.30.12  192.168.30.20 584      0x80000005    0xFC4C
192.168.30.12  192.168.30.30 56        0x80000004    0x45BA
172.20.57.254  192.168.30.70 664      0x800000CE    0xF2CF
172.20.57.254  192.168.30.80 963      0x80000295    0x23CC

      Router Link States (Area 4)

Link ID        ADV Router    Age      Seq#          Checksum      Link count
192.168.30.14  192.168.30.14 311      0x80000EA5    0x93A0        7
192.168.30.24  192.168.30.24 685      0x80001333    0x6F56        6
192.168.30.50  192.168.30.50 116      0x80001056    0x42BF        2
192.168.30.54  192.168.30.54 1213     0x80000D1F    0x3385        2

      Summary Net Link States (Area 4)

Link ID        ADV Router    Age      Seq#          Checksum
172.16.121.0   192.168.30.40 1231     0x80000D88    0x73BF
172.16.121.0   192.168.30.50 34        0x800003F4    0xF90D
10.63.65.0     192.168.30.40 1240     0x80000003    0x5110
10.63.65.0     192.168.30.50 42        0x80000005    0x1144

      Summary ASB Link States (Area 4)

Link ID        ADV Router    Age      Seq#          Checksum
192.168.30.12  192.168.30.40 1240     0x80000006    0x6980
192.168.30.12  192.168.30.50 42        0x80000008    0xC423
172.20.57.254  192.168.30.40 1241     0x8000029B    0xEED8
172.20.57.254  192.168.30.50 43        0x800002A8    0x9818

      AS External Link States

Link ID        ADV Router    Age      Seq#          Checksum      Tag
10.83.10.0     192.168.30.60 459      0x80000D49    0x9C0B        0
10.1.27.0      192.168.30.62 785      0x800000EB    0xB5CE        0
10.22.85.0     192.168.30.70 902      0x8000037D    0x1EC0        65502
10.22.85.0     192.168.30.80 1056     0x800001F7    0x6B4B        65502

Homer#
```

图 9-27 使用命令 `show ip ospf database` 来显示一个链路状态数据库中所有 LSA 通告的列表

在 IOS 软件 11.3 版本之前, Cisco 的路由器只选用单个 LSRefreshTime 和整个链路状态数据库相关联。每隔 30min, 每台路由器将重刷新它始发的所有 LSA 通告, 而不管这些 LSA 通告实际的老化时间。虽然这种策略避免了链路带宽使用低效的问题, 但是它再次引入了本

应解决的独自重刷新计时器的问题。如果一个链路状态数据库很大,那么每隔 30min,网络上就会产生一个区域通信量和 CPU 利用率的高峰。

```
Homer#show ip ospf database database-summary

OSPF Router with ID (192.168.30.50) (Process ID 1)

Area ID      Router  Network  Sum-Net  Sum-ASBR  Subtotal  Delete  Maxage
0            8        4       185     27        224       0        0
4            7        0       216     26        249       0        0
5            7        0       107     13        127       0        0
56           2        1       236     26        265       0        0
AS External              580       0        0
Total        24        5       744     92        1445
Homer#
```

图 9-28 使用命令 `show ip ospf database database-summary` 来显示一个链路状态数据库当中基于区域和 LSA 类型分类的 LSA 通告的数量

在 IOS 软件 11.3AA 版本里,提供了一种称为 LSA 组步调的机制,作为 LSA 独自使用重刷新计时器和使用单个统一的计时器问题之间的一种折衷办法。每一个 LSA 通告都有属于自己的重刷新计时器,但是当它们独自使用的重刷新计时器超时的时候,会引入一个时延来延迟这些 LSA 通告的泛洪。通过延迟重刷新时间,可以在泛洪之前将更多的 LSA 通告共同编成一组,从而可以让更新报文携带更大数量的 LSA 通告。缺省条件下,一个组步调 (group-pacing) 的间隔时间是 240s (4min)。这个间隔时间可以通过命令 `timers lsa-group-pacing` 来改变。如果链路状态数据库非常大,那么减小组步调的间隔时间是有好处的;而如果链路状态数据库很小,那么增加组步调的间隔时间会比较有用。在这里,组步调计时器的大小范围是 10~1800s。

1. LSA 的类型

由于 OSPF 协议定义了多种路由器的类型,因而定义多种 LSA 通告的类型也是必要的。例如,一台 DR 路由器必须通告多路访问链路和所有与这条链路相连的路由器,而其他类型的路由器将不需要通告这种类型的信息。在图 9-27 和图 9-28 中已经显示了多种类型的 LSA 通告。每一种 LSA 通告类型都描述了 OSPF 网络的一个不同情况。表 9-4 中列出了 LSA 通告的类型和标识这些 LSA 类型的代码。

表 9-4 LSA 类型

类型代码	描述
1	路由器 LSA
2	网络 LSA
3	网络汇总 LSA
4	ASBR 汇总 LSA
5	AS 外部 LSA
6	组成员 LSA
7	NSSA 外部 LSA
8	外部属性 LSA
9	Opaque LSA (本地链路范围)
10	Opaque LSA (本地区域范围)
11	Opaque LSA (AS 范围)

- **路由器 LSA (Router LSA)**——每一台路由器都会产生路由器 LSA 通告 (如图 9-29 所示)。这个最基本的 LSA 通告列出了路由器所有的链路或接口, 并指明了它们的状态和沿每条链路方向出站的代价。这些 LSA 通告只会在始发它们的区域内部进行泛洪。通过命令 **show ip ospf database router** 可以查看数据库中列出了所有路由器 LSA 通告。如图 9-30 所示, 显示了这条命令, 并在命令后加了一个变量指定一个路由器 ID, 从而观察到单个路由器 LSA 通告的详细信息。在这个及其后面的一些图示中, 显示了记录在链路状态数据库中的完整的 LSA 信息。关于对所有 LSA 字段的介绍, 请参考本章后面“OSPF 报文格式”一节中的讲述。

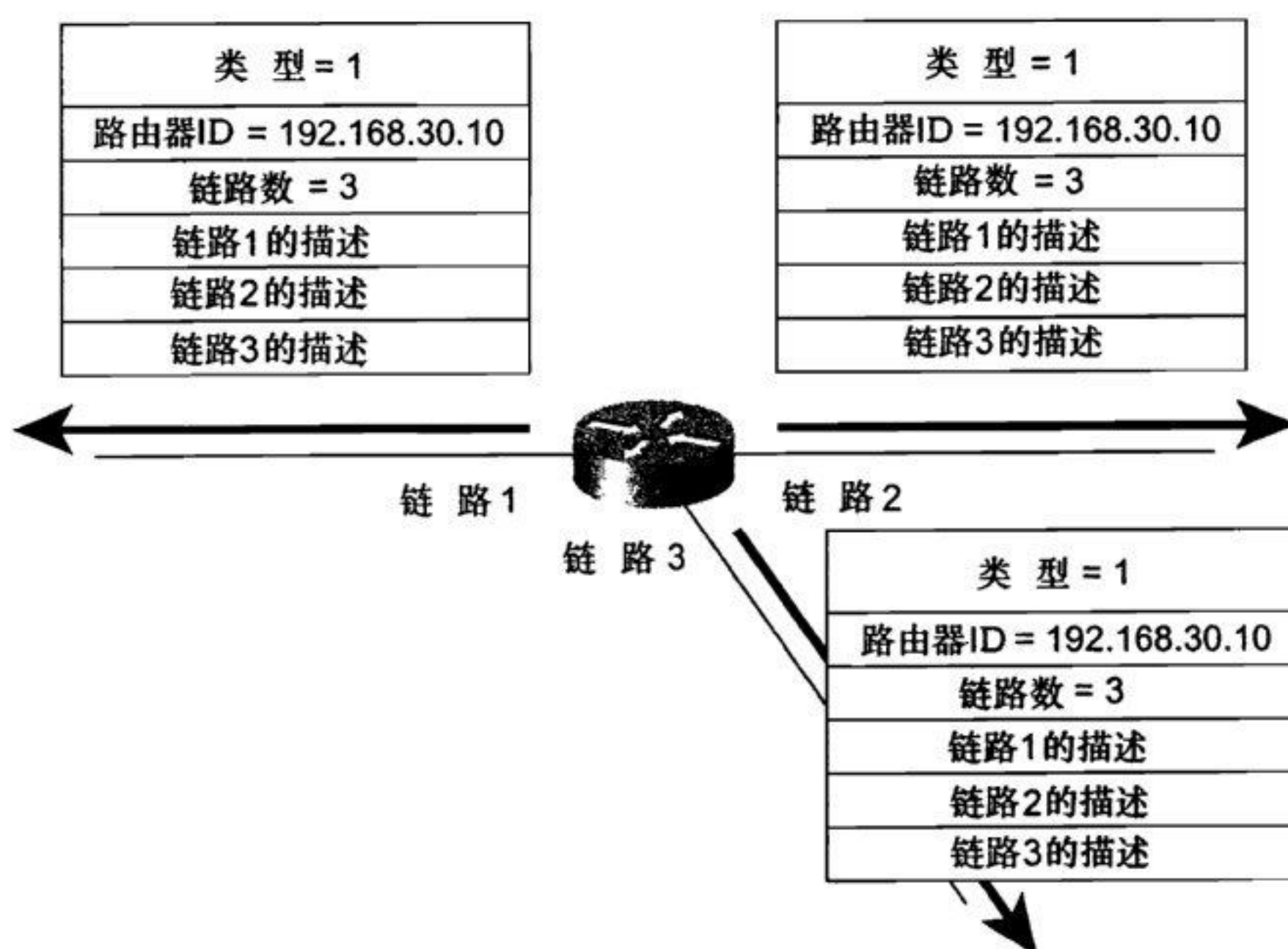


图 9-29 路由器 LSA 描述了所有的路由器接口

```
Homer#show ip ospf database router 192.168.30.10

OSPF Router with ID (192.168.30.50) (Process ID 1)

Router Link States (Area 0)

Routing Bit Set on this LSA
LS age: 680
Options: (No TOS-capability)
LS Type: Router Links
Link State ID: 192.168.30.10
Advertising Router: 192.168.30.10
LS Seq Number: 80001428
Checksum: 0x842A
Length: 60
Area Border Router
Number of Links: 3

Link connected to: another Router (point-to-point)
(Link ID) Neighboring Router ID: 192.168.30.80
```

待续


```
(Link Data) Router Interface address: 192.168.17.9
Number of TOS metrics: 0
TOS 0 Metrics: 64

Link connected to: a Stub Network
(Link ID) Network/subnet number: 192.168.17.8
(Link Data) Network Mask: 255.255.255.248
Number of TOS metrics: 0
TOS 0 Metrics: 64

Link connected to: a Transit Network
(Link ID) Designated Router address: 192.168.17.18
(Link Data) Router Interface address: 192.168.17.17
Number of TOS metrics: 0
TOS 0 Metrics: 10

Homer#
```

图 9-30 通过命令 **show ip ospf database router** 可以显示出一个链路状态数据库中的路由器 LSA 通告

- **网络 LSA (Network LSA)** ——每一个多路访问网络中的指定路由器 DR 将会产生网络 LSA 通告, 如图 9-31 所示。正如前面讨论的, DR 路由器可以看作一个“伪”节点, 或是一个虚拟路由器, 用来描绘一个多路访问网络和与之相连的所有路由器。从这个角度来看, 一条网络 LSA 通告也可以描绘一个逻辑上的“伪”节点, 就像一条路由器 LSA 通告描绘一个物理上的单个路由器一样。网络 LSA 通告列出了所有与之相连的路由器, 包括 DR 路由器本身。像路由器 LSA 一样, 网络 LSA 也仅仅在产生这条网络 LSA 的区域内部进行泛洪。如图 9-32 所示, 使用命令 **show ip ospf database network** 可以查看一条网络 LSA 通告的信息。
- **网络汇总 LSA (Network Summary LSA)** ——是由 ABR 路由器始发的。ABR 路由器将发送网络汇总 LSA 到一个区域, 用来通告该区域外部的目的地址 (如图 9-33 所示)。实际上, 这些网络汇总 LSA 就是 ABR 路由器告诉在与之相连的区域内的内部路由器它所能到达的目的地址的一种方法。一台 ABR 路由器也可以通过网络汇总 LSA 向骨干区域通告与它相连的区域内部的目的地址。在一个区域外部但是仍然在一个 OSPF 自主系统内部的缺省路由也可以通过这种 LSA 类型来通告。使用命令 **show ip ospf database summary** 可以显示链路状态数据库中的网络汇总 LSA 信息, 如图 9-34 所示。

当一台 ABR 路由器始发一条网络汇总 LSA 时, 它将包括从它本身到正在通告的这条 LSA 的目的地址所耗费的代价。ABR 路由器即使知道它有多条路由可以到达目的地, 它也只能为这个目的地始发单条网络汇总 LSA 通告。因此, 如果一台 ABR 路由器在与它本身相连的区域内部有多条路由可以到达目的地, 那么它将只会始发单一的一条网络汇总 LSA 到骨干区域, 而且这条网络汇总 LSA 是上述多条路由中代价最低的。同样地, 如果一台 ABR 路由器经过骨干区域从其他的 ABR 路由器收到多条网络汇总 LSA, 那么这台始发的 ABR 路由器将会选择这些 LSA 通告中代价最低的 LSA, 并且将把这个 LSA 的最低代价通告给与它相连的非骨干区域。

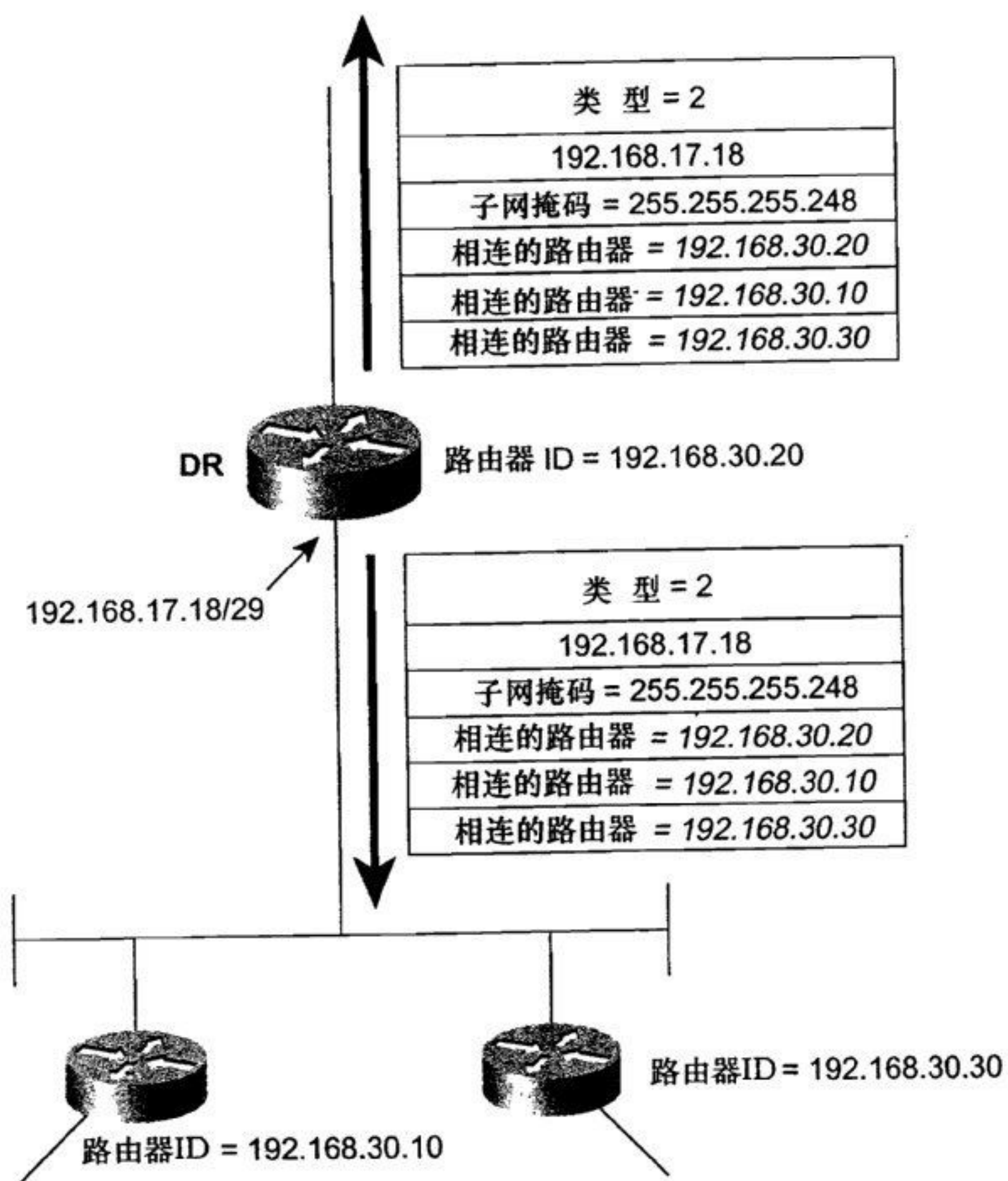


图 9-31 DR 路由器始发了一条可以描绘一个多路访问网络和与之相连的所有路由器的网络 LSA 通告

```
Homer#show ip ospf database network 192.168.17.18

OSPF Router with ID (192.168.30.50) (Process ID 1)

Net Link States (Area 0)

Routing Bit Set on this LSA
LS age: 244
Options: (No TOS-capability)
LS Type: Network Links
Link State ID: 192.168.17.18 (address of Designated Router)
Advertising Router: 192.168.30.20
LS Seq Number: 800001BF
Checksum: 0x60AC
Length: 32
Network Mask: /29
    Attached Router: 192.168.30.20
    Attached Router: 192.168.30.10
    Attached Router: 192.168.30.30

Homer#
```

图 9-32 网络 LSA 通告可以通过命令 `show ip ospf database network` 来查看

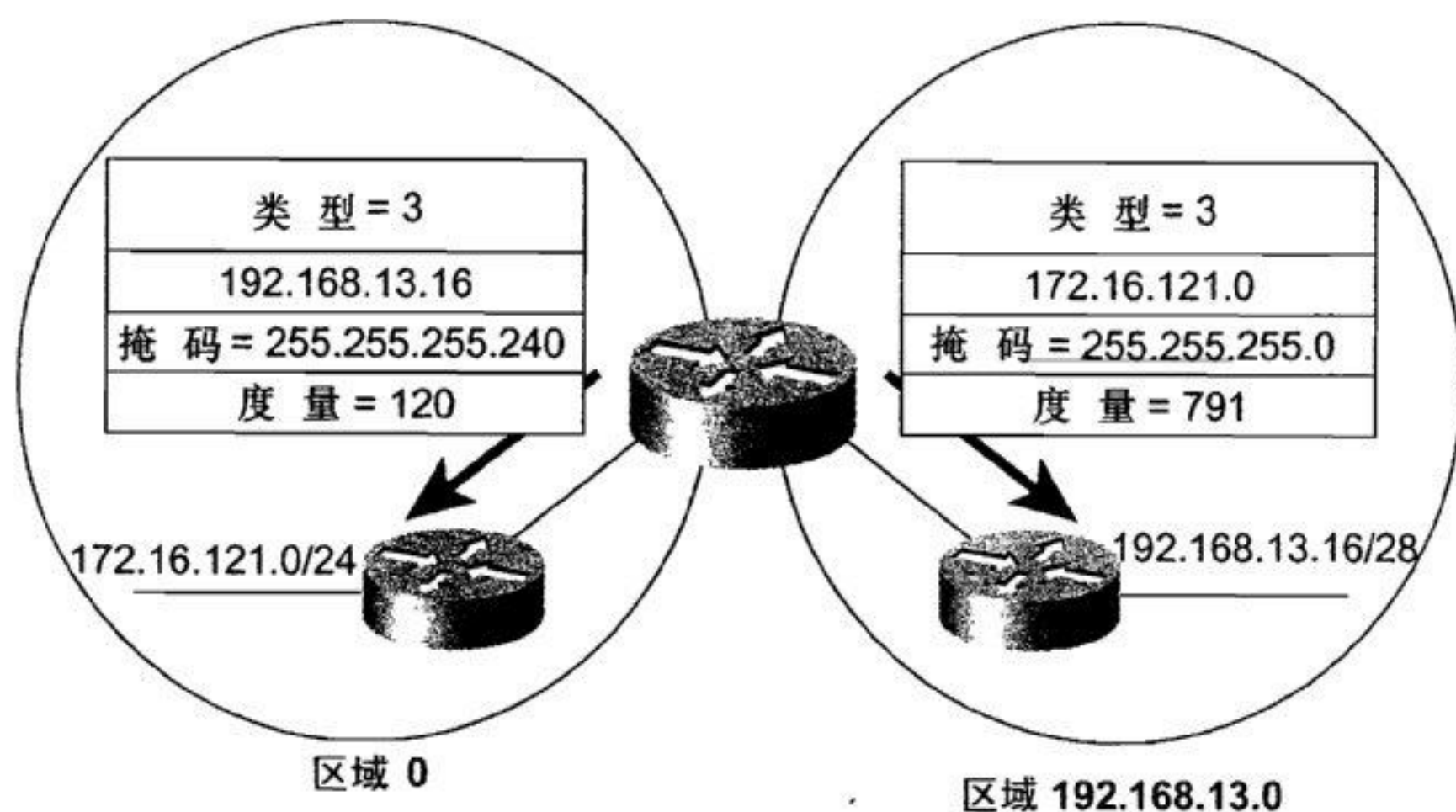


图 9-33 一台 ABR 路由器始发了一条网络汇总 LSA 来描述域间通信的目的地址

```
Homer#show ip ospf database summary 172.16.121.0

OSPF Router with ID (192.168.30.50) (Process ID 1)

Summary Net Link States (Area 0)

Routing Bit Set on this LSA
LS age: 214
Options: (No TOS-capability)
LS Type: Summary Links(Network)
Link State ID: 172.16.121.0 (summary Network Number)
Advertising Router: 192.168.30.60
LS Seq Number: 800000B1
Checksum: 0xE864
Length: 28
Network Mask: /24
TOS: 0 Metric: 791
```

图 9-34 可以通过命令 `show ip ospf database summary` 来查看网络汇总 LSA 通告的信息

当其他的路由器从一台 ABR 路由器收到一条网络汇总 LSA 通告时，它并不运行 SPF 算法。相反地，它只是简单地加上从它到那台 ABR 路由器之间路由的代价，并将这个代价包含在这个 LSA 通告当中。通过 ABR 路由器，到达所通告的目的地的路由连同所计算的代价一起被记录进了路由选择表。这个行为——依赖中间路由器代替确定到达目的地的全程路由 (full route) 的做法其实是距离矢量协议的行为。因此，虽然在一个区域内部 OSPF 协议是一个链路状态协议，但是它却使用了距离矢量的算法来查找域间路由。¹

- **ASBR 汇总 LSA (ASBR Summary LSA)** ——也是由 ABR 路由器始发出的。ASBR 汇总 LSA 除了所通告的目的地是一个 ASBR 路由器而不是一个网络外，其他的和网络汇总 LSA 都是一样的，如图 9-35 所示。使用命令 `show ip ospf database asbr-summary` 可以查看 ASBR 汇总 LSA 的信息，如图 9-36 所示。这里要注意，在这个图示中，目的地是一个主机地址，并且掩码是 0；通过 ASBR 汇总 LSA 通告的

¹ 这个距离矢量的行为就是 OSPF 协议为什么需要一个骨干区域和为什么需要所有的域间通信量都必须通过骨干区域的原因。通过把这些区域构成一个本质上像 hub-and-spoke 一样的网络拓扑，可以避免距离矢量协议中易于出现的路由环路。

目的地将总是一个主机地址，因为它是一条到达一台路由器的路由。

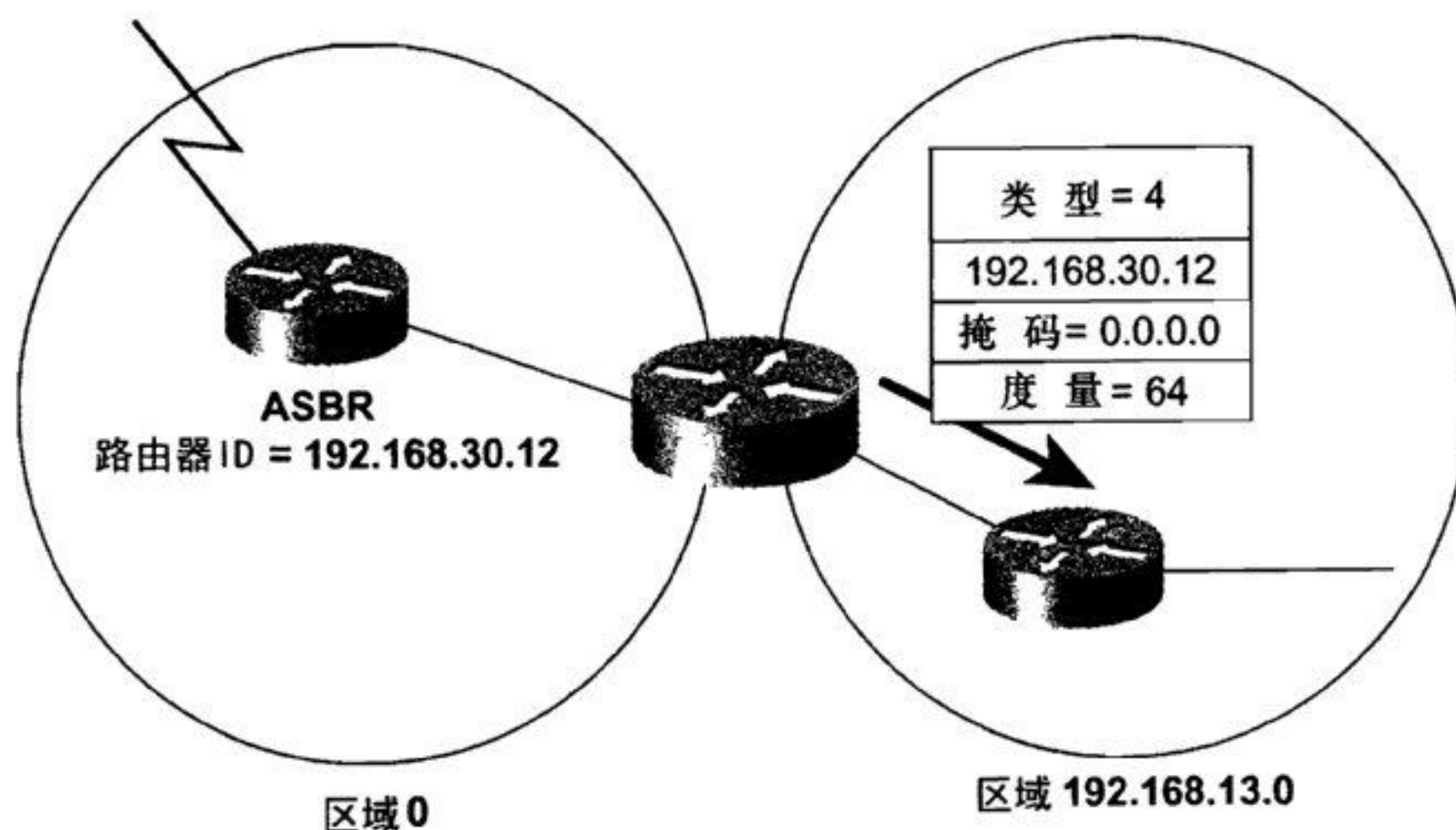


图 9-35 ASBR 汇总 LSA 通告到达 ASBR 路由器的路由

```
Homer#show ip ospf database asbr-summary

OSPF Router with ID (192.168.30.50) (Process ID 1)

Summary ASB Link States (Area 0)

Routing Bit Set on this LSA
LS age: 1640
Options: (No TOS-capability)
LS Type: Summary Links (AS Boundary Router)
Link State ID: 192.168.30.12 (AS Boundary Router address)
Advertising Router: 192.168.30.20
LS Seq Number: 80000009
Checksum: 0xF450
Length: 28
Network Mask: /0
TOS: 0 Metric: 64

--More--
```

图 9-36 可以通过命令 `show ip ospf database asbr-summary` 来查看 ASBR 汇总 LSA 通告的信息

- **自主系统外部 LSA (Autonomous System External LSA)** ——或者称为外部 LSA (External LSA)，是始发于 ASBR 路由器的，用来通告到达 OSPF 自主系统外部的目的地或者 OSPF 自主系统外部的缺省路由¹的 LSA，如图 9-37 所示。回顾一下图 9-27，读者可以发现自主系统外部 LSA 是链路状态数据库中惟一不和具体的区域相关联的 LSA 通告。外部 LSA 通告将在整个自主系统中进行泛洪。使用命令 `show ip ospf database external` 可以查看 AS 外部 LSA 的信息，如图 9-38 所示。
- **组成员 LSA (Group Membership LSA)** ——是使用在 OSPF 协议的一个增强版本

¹ 缺省路由是指在路由选择表中，如果没有更具体的路由存在就可以选用的路由。OSPF 协议和 RIP 协议使用 IP 地址 0.0.0.0 来表示一个缺省路由。更详细的信息请参考第 12 章的内容。

——组播 OSPF 协议 (MOSPF 协议) 中的。¹MOSPF 协议将数据包从一个单一的源地址转发到多个目的地, 或者是一组共享 D 类组播地址的成员。虽然 Cisco 的路由器支持其他的组播路由选择协议, 但是在编写本书时还不支持 MOSPF 协议。由于这个原因, 本书将不介绍 MOSPF 协议, 也不介绍组成员 LSA 通告。

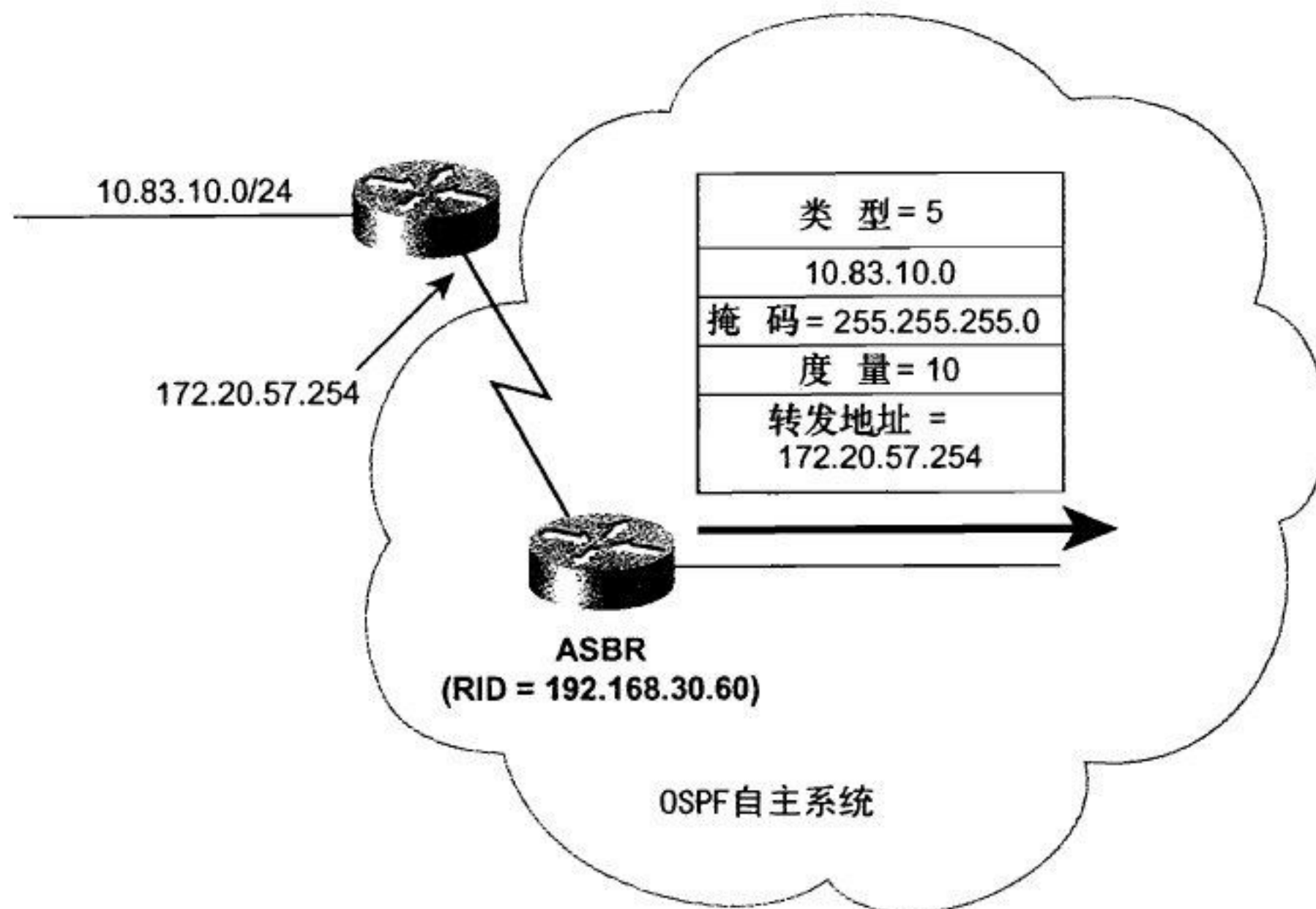


图 9-37 自主系统外部 LSA 用来通告到达 OSPF 自主系统外部的目的地

```
Homer#show ip ospf database external 10.83.10.0

OSPF Router with ID (192.168.30.50) (Process ID 1)

AS External Link States

Routing Bit Set on this LSA
LS age: 1680
Options: (No TOS-capability)
LS Type: AS External Link
Link State ID: 10.83.10.0 (External Network Number)
Advertising Router: 192.168.30.60
LS Seq Number: 80000D5A
Checksum: 0x7A1C
Length: 36
Network Mask: /24
Metric Type: 1 (Comparable directly to link state metric)
TOS: 0
Metric: 10
Forward Address: 172.20.57.254
External Route Tag: 0

Homer#
```

图 9-38 可以通过命令 `show ip ospf database external` 来查看自主系统外部 LSA 通告的信息

- **NSSA 外部 LSA (NSSA External LSA)**——是指在非纯末梢区域 (not-so-stubby area, NSSA) 内始发于 ASBR 路由器的 LSA 通告。NSSA 区域将在后面的章节介绍。正

¹ John Moy, "Multicast Extensions to OSPF," RFC 1584, 1994 年 3 月。

像 OSPF 报文格式一节中所介绍的那样, NSSA 外部 LSA 通告几乎和自主系统外部 LSA 通告是相同的。只是不像自主系统外部 LSA 通告那样在整个 OSPF 自主系统内进行泛洪, NSSA 外部 LSA 通告仅仅在始发这个 NSSA 外部 LSA 通告的非纯末梢区域内部进行泛洪。可以通过命令 **show ip ospf database nssa-external** 来显示 NSSA 外部 LSA 通告的信息, 如图 9-39 所示。

- **外部属性 LSA (External Attributes LSA)** ——是被提议作为运行内部 BGP 协议 (iBGP 协议) 的另一种选择, 以便用来传送 BGP 协议的信息穿过一个 OSPF 域。在本书编写的时候, 类型 8 的 LSA 还没有实现, 也没有关于这个主题的 RFC 或 Internet 草案 (Internet Draft) 发布。

```
Morisot#show ip ospf database nssa-external

      OSPF Router with ID (10.3.0.1) (Process ID 1)

                Type-7 AS External Link States (Area 15)

LS age: 532
Options: (No TOS-capability, No Type 7/5 translation, DC)
LS Type: AS External Link
Link State ID: 10.0.0.0 (External Network Number)
Advertising Router: 10.3.0.1
LS Seq Number: 80000001
Checksum: 0x9493
Length: 36
Network Mask: /16
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 100
    Forward Address: 10.3.0.1
    External Route Tag: 0

--More--
```

图 9-39 可以通过命令 **show ip ospf database nssa-external** 来查看 NSSA 外部 LSA 通告的信息

- **Opaque LSA** ——是一个被提议的 LSA 类别, 由标准的 LSA 头部后面跟随特殊应用 (application-specific) 的信息组成。¹这个信息字段可以直接由 OSPF 协议使用, 或者由其他应用分发信息到整个 OSPF 域间接使用。在本书编写的时候, Opaque LSA 类型还没有实现。

2. 末梢 (Stub) 区域

一个学习到外部目的路由信息的 ASBR 路由器将通过在整个 OSPF 自主系统中泛洪自主系统外部 LSA 来通告那些外部的目的路由信息。在大多数的实际案例中, 这些外部 LSA 通告可能会在每台路由器的链路状态数据库中构成较大百分比的 LSA 数量。例如, 在图 9-28 显示的那个链路状态数据库中有 580 个 LSA (约 40%) 是外部 LSA 通告。

如图 9-40 所示, 并不是每一台路由器都需要了解所有外部目的地的信息的。不管外部的目的地可能在哪里, 在区域 2 中的路由器都必须发送数据包到达 ABR 路由器, 以便到达那个 ASBR 路由器。在这种情况下, 区域 2 可以被配置成为一个末梢区域。

¹ Rob Coltun, "The OSPF Opaque LSA Option," RFC 2370, 1998 年 7 月。

末梢区域是一个不允许 AS 外部 LSA 通告在其内部进行泛洪的区域。如果在一个区域里没有学到类型 5 的 LSA 通告, 那么类型 4 的 LSA 通告也是不必要的了, 因此这些 LSA 通告也将被阻塞。位于末梢区域边界的 ABR 路由器将使用网络汇总 LSA 向这个区域通告一个简单的缺省路由 (目的地址是 0.0.0.0)。在区域内部路由器上, 所有和域内或域间路由不能匹配的目的地地址都将最终匹配这条缺省路由。由于缺省路由是由类型 3 的 LSA 通告传送的, 因此它不会被通告到这个区域的外部去。

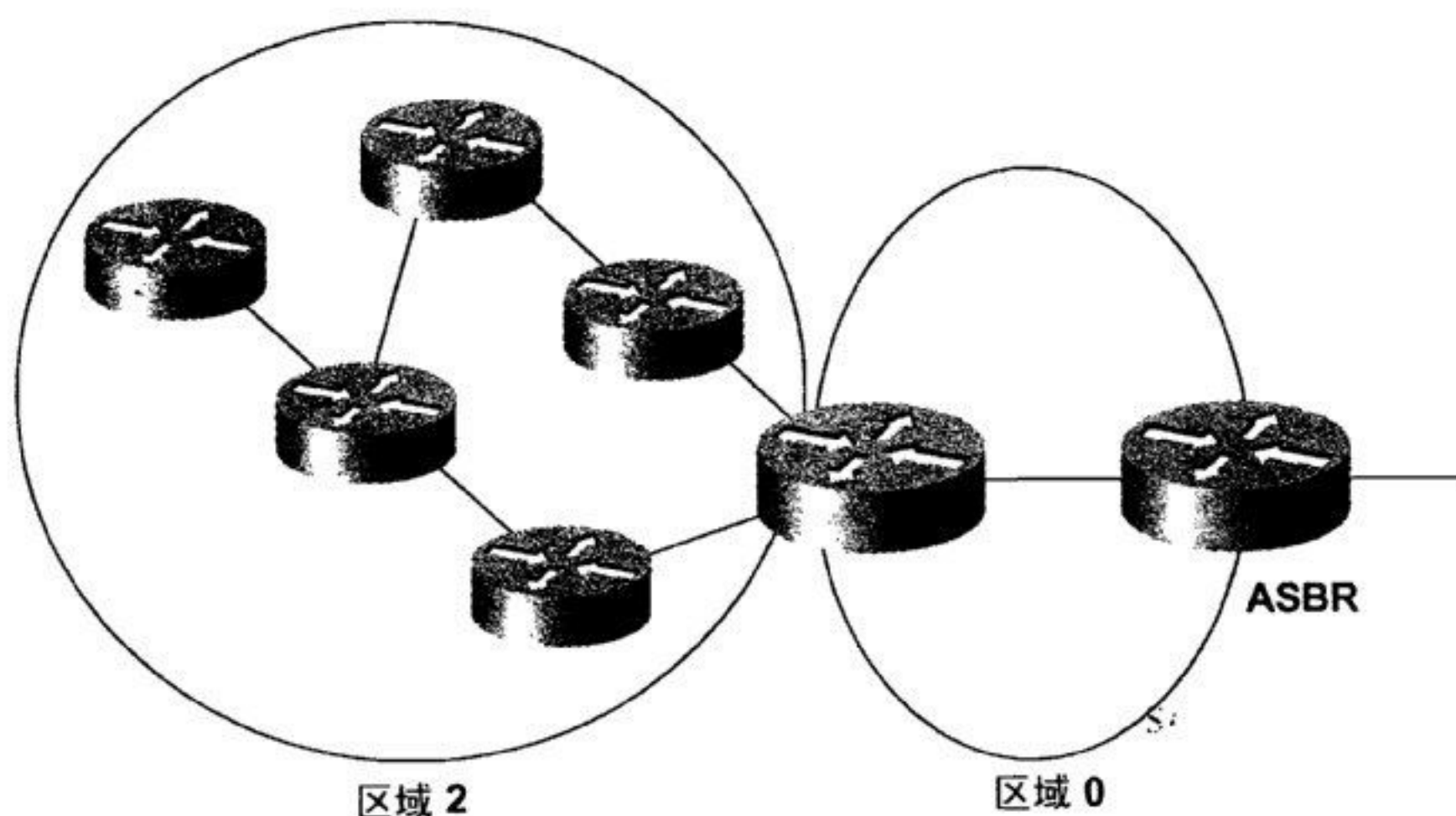


图 9-40 可以通过使区域 2 成为一个末梢区域来节省内存和提高性能

由于在一个末梢区域里, 路由器的链路状态数据库的大小被减小了, 因此, 这些路由器的性能将得到提高, 并且内存也得到节省。当然, 在一个含有大量类型 5 的 LSA 通告的互联网络里, 这种改进将更加显著。然而, 在末梢区域中也有 4 个限制条件:

- 和所有的区域一样, 一个末梢区域内部的所有路由器也必须拥有相同的链路状态数据库。为了确保满足这个条件, 所有末梢区域内的路由器都会在它们的 Hello 报文中设置一个标志——就是 E-bit 位, 并将它设置为 0。这样, 这些末梢区域路由器将不接受其他路由器发送的任何 E-bit 为 1 的 Hello 报文。结果, 没有配置成一个末梢区域路由器的任何路由器之间将无法建立成功邻接关系。
- 虚链路不能在一个末梢区域内进行配置, 也不能穿过一个末梢区域。
- 末梢区域内的路由器不能是 ASBR 路由器。这个限制条件是很容易直观地理解的, 因为 ASBR 路由器会产生类型 5 的 LSA 通告, 而在一个末梢区域内不能存在类型 5 的 LSA 通告。
- 一个末梢区域可以拥有多台 ABR 路由器, 但是因为缺省路由的原因, 区域内部路由器将不能确定哪一台路由器才是到达 ASBR 路由器的最优的网关。

(1) 完全末梢区域

如果通过阻塞类型 5 和类型 4 的 LSA 传播到一个区域的方法来节省内存的话, 那么要是能够把类型 3 的 LSA 也阻塞掉, 不是可以节省更多的内存吗? 对于这个问题, Cisco 借助于末梢区域的概念提出了称为完全末梢区域的概念。

完全末梢区域 (Totally Stubby Area) 不仅使用缺省路由到达 OSPF 自主系统外部的目的地地址, 而且使用缺省路由到达这个区域外部的所有目的地地址。一个完全末梢区域的 ABR 将不仅阻塞 AS 外部 LSA, 而且阻塞所有的汇总 LSA——除了通告缺省路由的那一条类型

3 的 LSA。

(2) 非纯末梢区域

在图 9-41 中;带有一些末梢网络的某台路由器必须通过区域 2 的其中一台路由器和图中的 OSPF 网络相连。但是,该路由器仅支持 RIP 协议,因此,区域 2 的那台路由器将同时运行 RIP 协议和 OSPF 协议,并利用路由重新分配的方法把该路由器的那些末梢网络注入到 OSPF 域。不幸的是,这个配置将使区域 2 的那台路由器成为一台 ASBR 路由器,因此,区域 2 也就不再是一个末梢区域了。

在这里,RIP 协议的宣告者并不需要学习 OSPF 域的路由,而只需要有一条缺省路由指向那台区域 2 的路由器都足够了。但是,OSPF 域内的路由器为了能够正确转发数据包到达 RIP 路由器的那些目的地址,它们必须要学习到这些和 RIP 路由器相连的目的网络。

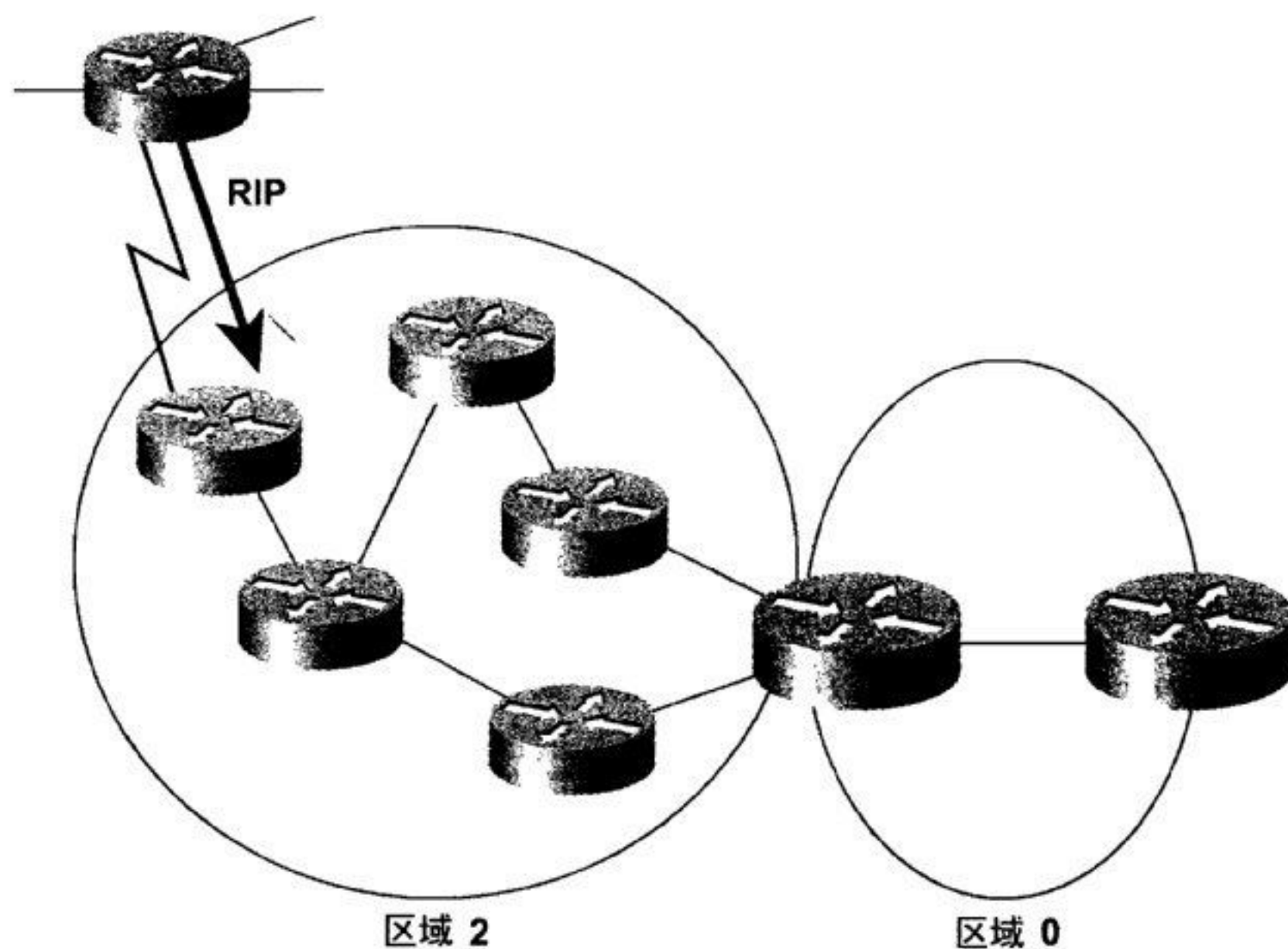


图 9-41 因为有一些 OSPF 域外部的目的网络必须在区域 2 的路由器上通过路由重新分配的方式注入到 OSPF 域中,因此区域 2 就不再满足末梢区域的条件了

非纯末梢区域 (Not-so-stubby-area, NSSA)¹ 允许外部路由通告到 OSPF 自主系统内部,而同时保留自主系统的其余部分的末梢区域特征。为了做到这一点,在 NSSA 区域内的 ASBR 将始发类型 7 的 LSA 来通告那些外部的目的网络。这些 NSSA 外部 LSA 将在整个 NSSA 区域中进行泛洪,但是会在 ABR 路由器的地方被阻塞。

NSSA 外部 LSA 在它的报文头部有一个称为 P-bit 位的标志。NSSA ASBR 路由器可以设置或清除这个 P-bit 位。如果一台 NSSA ABR 路由器收到一条 P-bit 设置为 1 的类型 7 的 LSA 报文,那么它将把这条类型 7 的 LSA 转换成为类型 5 的 LSA,并且将这条 LSA 泛洪到其他的区域中去 (请参见图 9-42)。如果这个 P-bit 位被设置为 0,那么将不会转换这条类型 7 的 LSA,而且这条类型 7 的 LSA 携带的目的地址也不能通告到这个 NSSA 区域的外部。

NSSA 区域是在 IOS 软件 11.2 版及其以后的版本才支持的。

表 9-5 总结了每一种区域内允许泛洪的 LSA 类型。

¹ Rob Coltun and Vince Fuller, "The OSPF NSSA Option," RFC 1587, 1994 年 3 月。

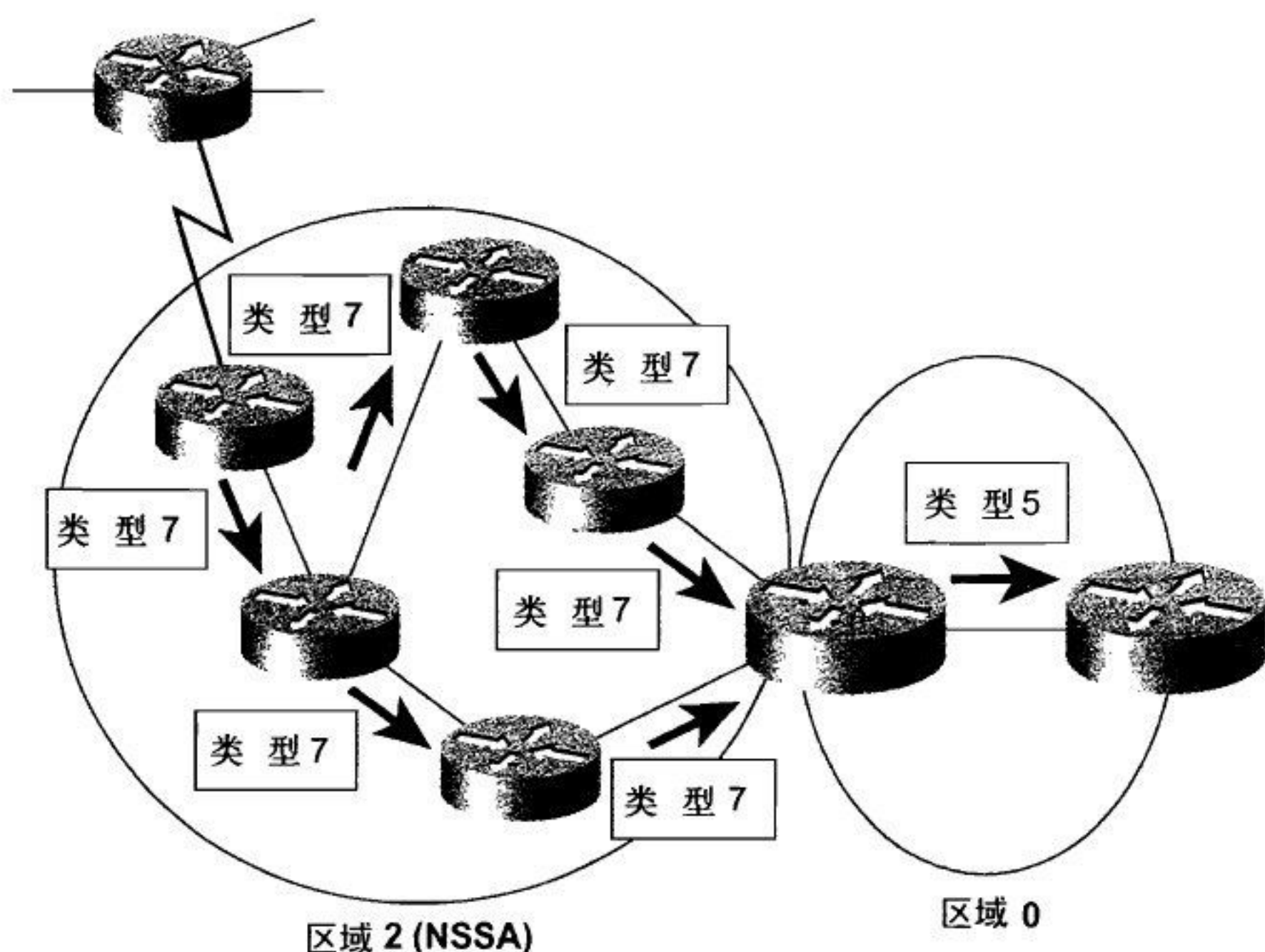


图 9-42 在 NSSA 区域内的 ASBR 路由器将会始发 NSSA 外部 LSA。如果一条 NSSA 外部 LSA 的 P-bit 位设置了，那么 ABR 路由器将会把这条 NSSA 外部 LSA 转换为一条 AS 外部 LSA

表 9-5 每一种区域内允许泛洪的 LSA 类型

区域类型	1&2	3&4	5	7
骨干区域 (区域 0)	允许	允许	允许	不允许
非骨干区域, 非末梢区域	允许	允许	允许	不允许
末梢区域	允许	允许	不允许	不允许
完全末梢区域	允许	不允许*	不允许	不允许
NSSA	允许	允许	不允许	允许

* 只有一个例外，就是在每一台 ABR 路由器上利用一个类型 3 的 LSA 来通告缺省路由。

9.1.4 路由选择表

根据链路状态数据库中的 LSA 信息，路由器使用 Dijkstra 算法来计算一棵最短路径树。第 4 章中已经比较详细地讲述了 Dijkstra 算法，如果需要查看关于 OSPF 协议计算 SPF 树的完整描述，请参考 RFC2328 中的第 16.1 节。SPF 算法一般要运行两次，第一次运行将根据连接到区域内的每一个节点（路由器）的链路创建 SPF 树的分支；接着，第二次运行 SPF 算法来增加这棵 SPF 树的叶节点——也就是和每台路由器相连的末梢网络。

OSPF 协议是基于路由器的每一个接口指定的度量值来决定最短路径的，这里的度量值指的是接口指定的代价 (Cost)。一条路由的代价是指沿着到达目的网络的路由路径上所有出站接口的代价之和。RFC2328 没有专门为代价指定任何值。Cisco 的路由器使用 $10^8/BW$ 作为 OSPF 协议缺省的代价，这里的 BW 是指在路由器接口上配置的带宽。如果计算所得的结果是分数的话就采用四舍五入的方法变为数值最接近的整数。接口上的代价可以通过命令 `ip ospf cost` 进行改变。LSA 使用一个 16 位的字段来记录这个代价值，因此，一个接口的代价的大小范围是 1~65535。表 9-6 中显示了一些典型的接口的缺省代价。

表 9-6

Cisco 路由器的缺省接口代价

接口类型	代价 (10 ⁶ /BW)
FDDI、快速以太网 (Fast Ethernet)	1
HSSI (45M)	2
16M 令牌环	6
以太网 (10M)	10
4M 令牌环	25
T1 (1.544M)	64
DS0 (64kbit/s) *	1562
56K *	1785
Tunnel (9kbit/s)	11111

* 假定串行接口的缺省带宽已被改变。

1. 目的类型

每一个路由条目都可以被归类到目的类型 (Destination Type) 中去。目的类型可以是网络也可以是路由器。

网络条目 (Network Entries) 是数据包所要转发的目的网络地址。这些网络条目就是记录到路由选择表中的目的网络地址, 如图 9-43 所示。

```
Homer#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
       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

Gateway of last resort is 192.168.32.2 to network 0.0.0.0

O E1 192.168.118.0/24 [110/94] via 192.168.17.74, 02:15:01, Ethernet0
O E1 10.0.0.0/8 [110/84] via 192.168.17.41, 02:15:01, Serial0.19
O E1 192.168.119.0/24 [110/94] via 192.168.17.74, 02:15:01, Ethernet0
O E2 172.19.0.0/16 [110/21] via 192.168.32.2, 02:15:01, Ethernet1
    172.21.0.0/16 is variably subnetted, 2 subnets, 2 masks
O E2 172.21.0.0/16 [110/801] via 192.168.21.6, 02:15:01, Serial1.724
O 172.21.121.0/24 [110/791] via 192.168.21.6, 04:18:30, Serial1.724
    172.16.0.0/16 is variably subnetted, 104 subnets, 7 masks
O 172.16.21.48/30 [110/844] via 192.168.21.10, 04:18:48, Serial1.725
O IA 172.16.30.61/32 [110/856] via 192.168.17.74, 02:15:19, Ethernet0
O IA 172.16.35.0/24 [110/865] via 192.168.17.74, 02:15:19, Ethernet0
C 172.16.32.0/24 is directly connected, Ethernet1
O 172.16.17.48/29 [110/74] via 192.168.17.74, 06:19:46, Ethernet0
O E1 172.16.46.0/24 [110/30] via 192.168.32.2, 02:15:19, Ethernet1
O 172.16.45.0/24 [110/20] via 192.168.32.2, 3d10h, Ethernet1
O IA 172.16.30.54/32 [110/1061] via 192.168.17.74, 02:15:21, Ethernet0
O 172.16.17.56/29 [110/84] via 192.168.17.74, 06:19:48, Ethernet0
O 172.16.54.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
O 172.16.55.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
O 172.16.52.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
O 172.16.53.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
C 172.16.25.28/30 is directly connected, Tunnel29
--More--
```

图 9-43 在路由选择表中的 OSPF 条目是网络目的类型

路由器条目 (Router Entries) 是到达 ABR 和 ASBR 路由器的路由。如果一台路由器需要发送数据包到一个区域外的目的地, 那么它就必须知道怎么到达一个 ABR 路由器; 同样, 如果一台路由器需要发送数据包到一个 OSPF 域外部的目的地, 那么它就必须知道怎么到达一个 ASBR 路由器。路由器条目就包含了这个信息, 并且路由器把路由器条目放在了一个单独的内部路由选择表里面。这个路由选择表可以通过命令 **show ip ospf border-routers** 来看, 如图 9-44 所示。

正如图 9-44 所显示的, 这个内部的路由选择表看上去和其他的普通路由选择表十分相似: 也包含目的地、度量值、下一跳地址和出口接口。这里所不同的是, 在这个内部的路由选择表中的所有目的地都是 ABR 和 ASBR 的路由器 ID。每一个条目都被打上区域内 (i) 或区域间 (I) 的标志, 用来表明这个条目的目的地是一台 ABR, 或是一台 ASBR, 或两者都是。所在的区域也被记录了, 以便用来进行 SPF 算法的迭代查找。

```
Homer#show ip ospf border-routers

OSPF Process 1 internal Routing Table

Codes:  i - Intra-area route, I - Inter-area route
i 192.168.30.10 [74] via 192.168.17.74, Ethernet0, ABR, Area 0, SPF 391
I 192.168.30.12 [148] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
I 192.168.30.18 [205] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
i 192.168.30.20 [84] via 192.168.17.74, Ethernet0, ABR, Area 0, SPF 391
i 192.168.30.27 [781] via 192.168.21.6, Serial1.724, ASBR, Area 7, SPF 631
i 192.168.30.30 [74] via 192.168.17.74, Ethernet0, ABR/ASBR, Area 0, SPF 391
I 192.168.30.38 [269] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
i 192.168.30.37 [390] via 192.168.21.10, Serial1.725, ASBR, Area 7, SPF 631
i 192.168.30.40 [84] via 192.168.17.74, Ethernet0, ABR/ASBR, Area 0, SPF 391
i 192.168.30.47 [400] via 192.168.21.10, Serial1.725, ASBR, Area 7, SPF 631
i 192.168.30.50 [74] via 192.168.17.41, Serial0.19, ABR/ASBR, Area 0, SPF 391
I 192.168.30.62 [94] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
i 192.168.30.60 [64] via 192.168.17.41, Serial0.19, ABR/ASBR, Area 0, SPF 391
i 192.168.30.60 [790] via 192.168.21.10, Serial1.725, ABR/ASBR, Area 7, SPF 631
i 192.168.30.80 [10] via 192.168.32.5, Ethernet1, ABR/ASBR, Area 78, SPF 158
i 192.168.30.80 [10] via 192.168.17.74, Ethernet0, ABR/ASBR, Area 0, SPF 391
i 172.20.57.254 [10] via 192.168.32.2, Ethernet1, ASBR, Area 78, SPF 158
Homer#
```

图 9-44 路由器条目放置在一个和网络条目相分开的内部表中, 用来表示到达 ABR 和 ASBR 路由器的路由

2. 路径类型

每一条到达一个网络目的地的路由都可以被归类到 4 种路径类型中的一种。这些路径类型 (Path Type) 是区域内路径、区域间路径、类型 1 的外部路径和类型 2 的外部路径。

- **区域内路径 (Intra-area path)** ——是指在路由器所在的区域内就可以到达目的地的路径。
- **区域间路径 (Inter-area path)** ——是指目的地在其他区域但是还在 OSPF 自主系统内的路径。在图 9-43 中, 打上了 IA 标志的条目就是区域间路径, 它总是至少通过一台 ABR 路由器。
- **类型 1 的外部路径 (Type 1 external path, E1)** ——是指目的地在 OSPF 自主系统外部的路径, 在图 9-43 中表示为 E1。当一条外部路由重新分配到任何自主系统的时候, 它都必须指定一个对那个自主系统中的路由选择协议有意义的度量值。在 OSPF 协议里, ASBR 路由器的责任是要给它们所要通告的外部路由指定一个代价

值。对于类型 1 的外部路径来说,这个代价值是这条路由的外部代价加上到达 ASBR 路由器的路径代价之和。关于配置一台 ASBR 路由器使用 E1 类型的度量来通告一条外部路由(路由重新分配)的介绍将在第 11 章“路由重新分配”中讲述。

- **类型 2 的外部路径 (Type 2 external path, E2)**——也是指目的地在 OSPF 自主系统外部的路径,但是在计算外部路由的度量时不再计入到达 ASBR 路由器的路径代价。E2 的路由类型将提供给网络管理员一个选择,可以告诉 OSPF 协议只需要考虑外部路由在 OSPF 外部的代价,而忽略到达 ASBR 路由器的内部代价。OSPF 外部路由在缺省条件下是类型 2 的外部路径,即 E2 路径。

在图 9-45 中,路由器 A 有两条到达外部目的网络 10.1.2.0 的路径。如果目的地址通过 E1 类型来通告,那么路径 A-B-D 的代价是 35(5+20+10),这条路径将比代价为 50(30+10+10)的路径 A-C-D 优先。如果目的地址通过 E2 类型来通告,那么到达 ASBR 路由器的那两条内部路径的代价将被忽略。在这个实例中,路径 A-B-D 的代价是 30(20+10),而路径 A-C-D 的代价为 20(10+10)。这时,后者反而是优先的路径。

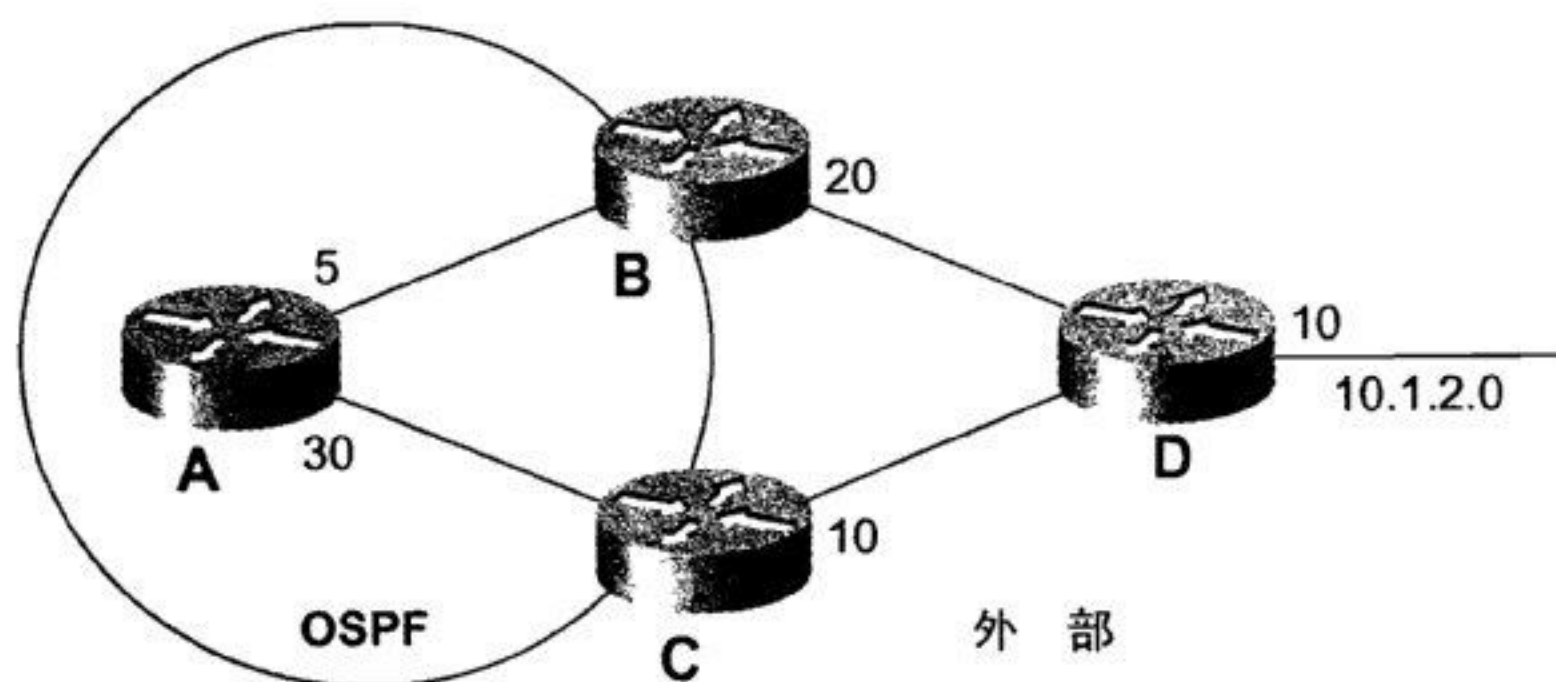


图 9-45 如果使用 E1 类型的度量来通告到达外部网络 10.1.2.0 的路由,那么路由器 A 将选择路由器 B 作为最近的 ASBR。

如果使用 E2 类型的度量来通告外部目的网络,那么路由器 C 将被选为最近的 ASBR

3. 路由选择表的查找

当一台 OSPF 路由器检查一个数据包的目的地址时,它将通过下面的步骤来选择最优的路由:¹

(1) 选择可以和目的地址最精确匹配的路由。例如,如果路由选择表中存在路由条目 172.16.64.0/18、172.16.64.0/24 和 172.16.64.192/27,而目的地址是 172.16.64.205,那么最后一个路由条目将被选中。最精确的匹配应该总是最长匹配——拥有最长的地址掩码的路由。路由条目可以是主机地址、子网地址、网络地址、超网地址,或者缺省地址。如果路由器没有发现匹配的条目,那么它将发送一个 ICMP 目的不可达的消息给那个数据包的源地址,并且把这个数据包丢弃。

(2) 通过排除次优的路径类型来剪除(prune)可选择条目的集合。路径类型根据下面的次序排列优先级,1 表示最高的优先级,而 4 表示最低的优先级:

- 1 区域内路径;
- 2 区域间路径;

¹ 这里描述的路由选择表查找过程是和 RFC2328 一致的。早期的 OSPF RFC 规定首先要创建一个匹配路由的集合,然后再选择优先的路径类型,最后再进行最长匹配。

3 E1 外部路径;

4 E2 外部路径。

如果在最后的路由子集中还有多条等价代价的、等价路径类型的路由存在,那么 OSPF 协议将会利用它们。缺省条件下, Cisco 路由器可以在最多 4 条等价代价的路径上实现负载均衡,这个数值可以通过命令 **maximum-paths** 来改变,改变的范围是 1~6。

9.1.5 认证

OSPF 协议在邻居路由器之间的所有报文的交换都具有认证的能力。认证可以是简单的口令认证或 MD5 加密校验和认证。这些认证的方法在第 7 章中已经讲述过了,本章将在后面的配置一节中给出一个配置 OSPF 认证的例子。

9.1.6 按需电路上的 OSPF

OSPF 协议每隔 10s 发送一次 Hello 报文,并且每隔 30min 重新刷新一次它的 LSA。这些功能都维护在邻接关系之间,以便确保链路状态数据库的精确,而且它比 RIP 或 IGRP 协议使用的带宽要少得多。然而,即使是这样一个最小的通信量,在按需电路上也是不希望有的——例如像在 X.25 SVC、ISDN 和拨号电路等即用即连 (usage-sensitive connection) 的电路上。像这些链路可能根据连接次数或通信量或两者皆有来循环计算费用,因而,促使网络管理员最小化地减少它们的上线时间。

一个在按需电路上实用的 OSPF 协议的增强特性是使 OSPF 具有抑制 Hello 报文和 LSA 重刷新的能力,以便链路不需要永久的有效。¹虽然这个增强特性是为即用即连的链路设计的,但是它在任何带宽有限的链路上也可能是有用的。²按需电路上的 OSPF 在 IOS 11.2 版本及其后续版本中得到支持。

按需电路上的 OSPF 将会激活一条按需链路去执行最初的数据库同步,随后只会激活这条链路去泛洪产生某些变化的 LSA。这些变化是:

- LSA 的可选字段发生了变化;
- 在老化时间达到 MaxAge 时收到了一个已经存在的 LSA 通告的新实例 (Instance);
- LSA 头部的长度字段发生了变化;
- LSA 的内容发生了变化,但不包括 20 个 8bit 字节的头部、校验和或者序列号。

由于没有周期性的 Hello 报文可以交换 (Hello 报文只有在链路激活时才能使用), OSPF 协议必须有一个可达性的假定 (presumption of reachability)。也就是说, OSPF 协议必须假定在需要链路连接的时候那条按需电路是可用的和有效的。但是在一些情况下,链路可能并不能立即可用和有效。例如,一个拨号链路可能是正在用的,一个 BRI 链路的两个 B 信道可能都在用,或者 X.25 所允许使用的最大的 SVC 电路数也都是在用的。在这些情形下,链路并不是因为链路失效而变得不可用的,而是因此链路太忙而变得不可用的,这也是这种链路的正常特点,可以称为链路过忙 (oversubscribed)。

¹ John Moy, "Extending OSPF to Support Demand Circuits," RFC 1793, 1995 年 4 月。

² 虽然按需电路上的 OSPF 可能被配置在任何接口上,但是,由于在多路访问类型的网络上不能抑制 Hello 报文的发送——如果这么做会阻碍 DR 处理的一些相应功能。结果,这个增强特性仅仅在点到点和点到多点类型的网络上才有实际用途。

OSPF 协议将不会把链路过忙的按需链路报告为失效, 因而数据包转发到一条过忙的链路上将会被丢弃而不是放入缓冲队列排队。这样做是有道理的, 因为 OSPF 没有办法预先知晓那条繁忙的链路什么时候可以再次变得可用, 一连串数据包转发到这个不可用的接口上就会导致它们的缓冲区溢出。

关于接口状态机和邻居状态机, 以及泛洪过程的处理必须有几处修改, 以便支持 OSPF 协议运行在按需电路上 (更详细的内容请参考 RFC1793)。在 LSA 通告的报文格式里, 也有两个地方需要修改。

首先, 如果 LSA 通告没有通过按需电路进行周期性的重复刷新, 那么经过 LSA 的最大生存时间 (MaxAge) 后, 在这条链路的另一端将不会有路由器宣称这条 LSA 无效。OSPF 协议更改了 LSA 的 Age 字段, 它将 Age 字段的更高一位指定为 DoNotAge 位, 以便解决这种情况的发生。当一条 LSA 在按需电路上进行泛洪时, 传送的路由器将把 DoNotAge 设置为 1。这样, 当这条 LSA 要泛洪到这条链路另一端的所有路由器时, Age 字段通常会增加一个 InfTransDelay 指定的秒数。¹但是, 当一条 LSA 被安置到路由器的链路状态数据库中后, 这条 LSA 将不再像其他 LSA 一样老化。

第二个修改来源于第一个修改。因为所有的路由器必须能够正确地识别这个更改的 DoNotAge 位, 因此在所有的 LSA 中增加了一个新的标志, 称为 DemandCircuit 位 (DC-bit)。通过在所有 LSA 发起的时候设置这个标志位, 路由器就可以通知其他路由器它是能够支持按需电路上的 OSPF 协议的。

9.1.7 OSPF 的报文格式

OSPF 报文是由多重封装构成的, 解析一个 OSPF 报文就像给洋葱剥皮一样。正如图 9-46 所示, 这个“洋葱”的外面一层是 IP 包的头部。在 Cisco 路由器中, OSPF 报文大小的最大值是 1500 个 8bit 字节。封装在 IP 头部内的是 5 种 OSPF 报文类型中的一种。每一种报文类型都是由一个 OSPF 报文头部开始的, 这个 OSPF 报文头部对于所有的报文类型都是相同的。紧跟 OSPF 报文头部之后的是 OSPF 报文数据, 并且根据报文类型的不同会有所不同。每一种报文类型都有许多的特有类型字段 (type-specific fields), 后跟更多的报文数据。在 Hello 报文中, 这些报文数据包含的是邻居路由器的列表。在链路状态请求报文中包含的是一系列描述被请求的 LSA 的字段。在链路状态更新报文中包含的是一个 LSA 的列表, 如图 9-46 中所示的那样。这些 LSA 依次都含有它们自己的头部和特有类型数据字段。数据库描述和链路状态确认报文将包含有一个 LSA 头部的列表。

这里要注意, 在一个网络上, OSPF 报文只能在邻居节点之间进行交换信息, 它们从来都不会转发到始发它们的节点所在的网络。

图 9-47 显示了一台协议分析仪捕获到的传送 OSPF 数据的报文的 IP 头部, 表明 OSPF 的协议号是 89。从这儿可以看出, OSPF 报文是和一个设置为 1 的 TTL 一起发送的。既然一个 OSPF 报文永远不应该跃过最近的邻居路由器转发, 那么设置 TTL 为 1 可以帮助 OSPF 报文确保自己的转发不会超过一跳。有一些路由器通过设置优先位 (Precedence bit) 来运行一些具有优先次序的报文处理。例如, 这里的优先位可能是加权公平队列 (Weighted Fair

¹ 注意, 这意味着 MaxAge 实际上是 MaxAge + DoNotAge。

Queuing, WFQ) 和加权随机预先检测 (Weighted Random Early Detection, WRED) 等。正如图 9-47 所显示的, OSPF 将设置优先位为互连网络控制 (Internetwork Control, 110b), 以便这些具有优先次序的报文处理给 OSPF 报文一个高的优先级。

这一节将从报文的头部开始, 详细介绍这 5 种类型的 OSPF 报文。随后的章节将详细介绍 LSA 报文类型。在 Hello 报文、数据库描述报文和所有的 LSA 报文中都含有一个可选字段 (Option), 这个字段的格式在所有的实例中都是一样的, 因此将单独列出一节来详细介绍它。

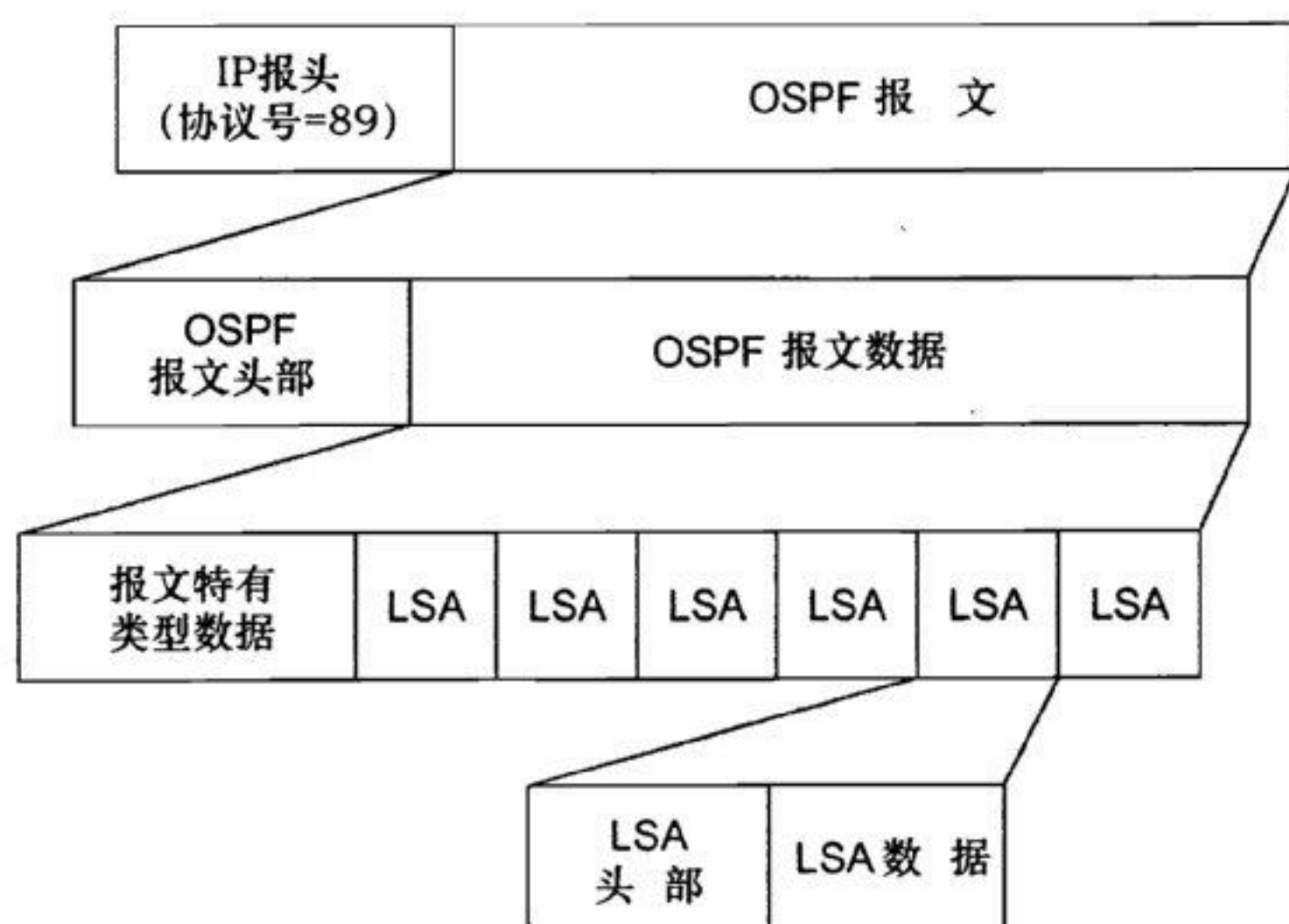


图 9-46 一个 OSPF 报文由一系列封装组成

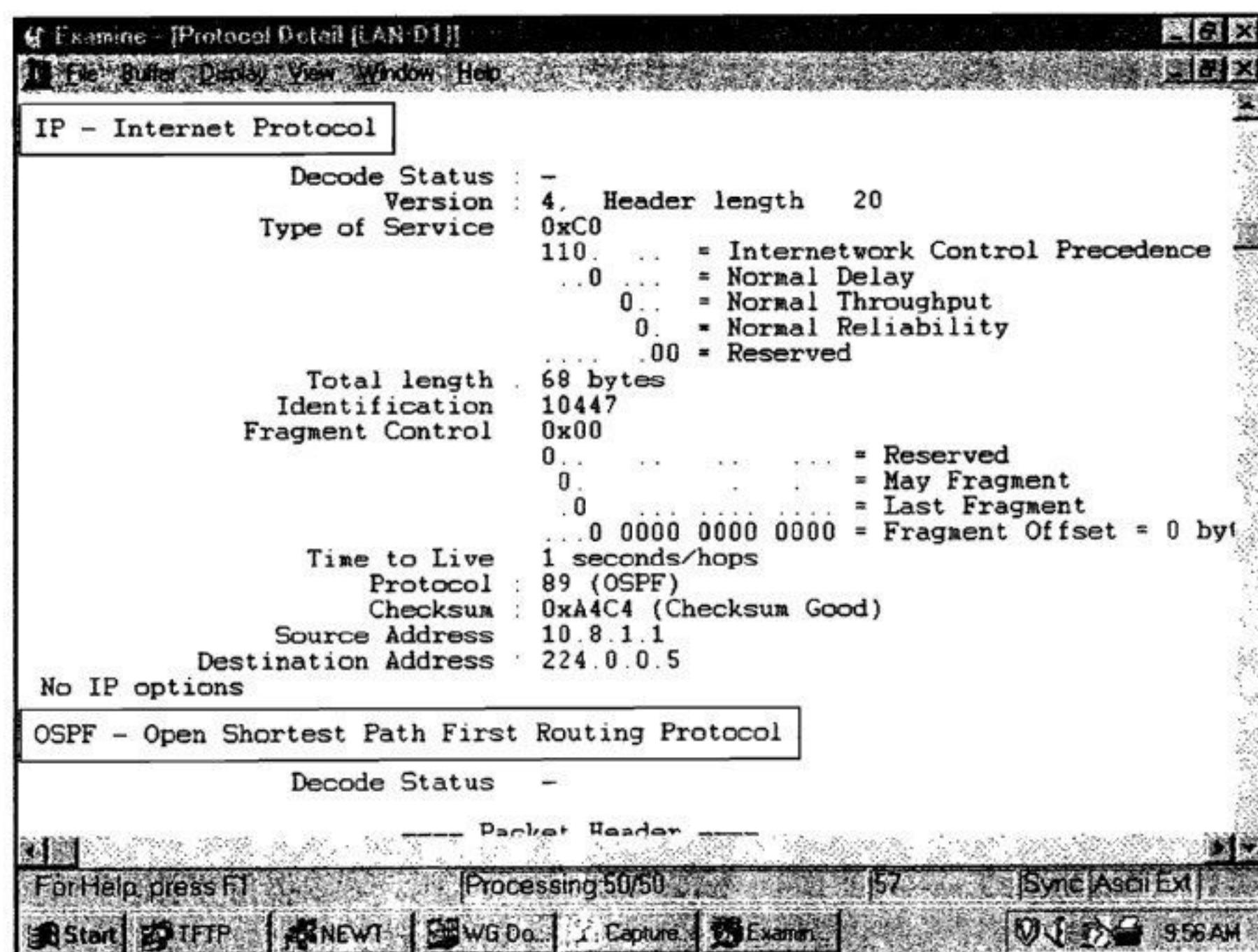
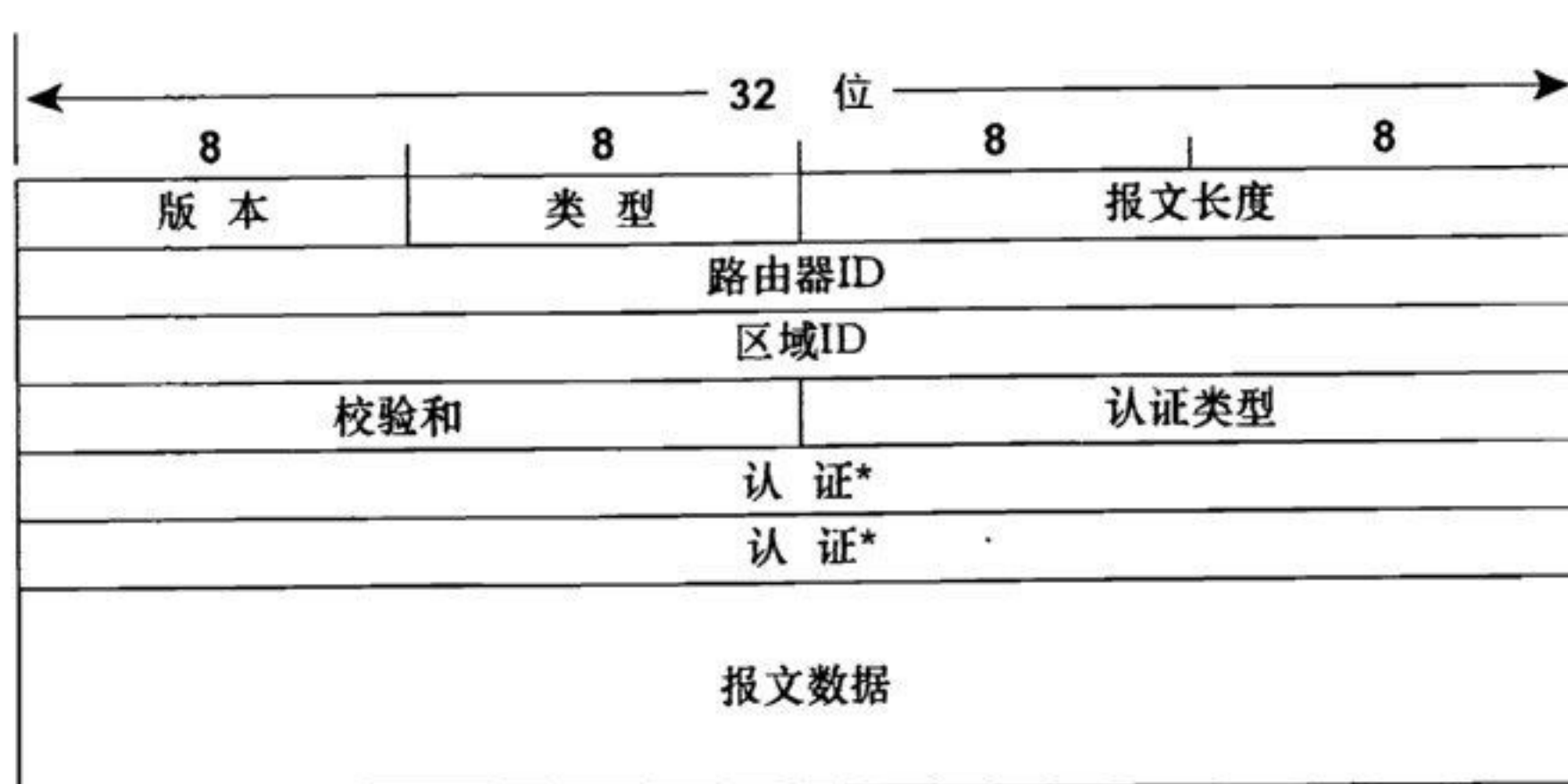


图 9-47 OSPF 使用的协议号是 89。OSPF 协议报文在 IP 头部的 TTL 值设置为 1, 并且把优先位设置成互连网络控制

1. 报文头部

所有 OSPF 报文都是由一个 24 个 8bit 字节的头部开始的, 如图 9-48 所示。



* 如果认证类型=2, 那么认证字段就是:

0x0000	密钥ID	认证数据长度
加密序列号		

图 9-48 OSPF 报文头部

- **版本 (Version)** ——是指 OSPF 的版本号。在本书编写的时候, 最新的 OSPF 版本号是 2。
- **类型 (Type)** ——指出跟在头部后面的报文类型。根据出现在类型字段的数字, 表 9-7 列出了这 5 种报文类型。

表 9-7

OSPF 报文类型

类 型 代 码	描 述
1	Hello
2	数据库描述
3	链路状态请求
4	链路状态更新
5	链路状态确认

- **报文长度 (Packet Length)** ——是指 OSPF 报文的长度, 包括报文头部的长度, 以 8bit 字节计。
- **路由器 ID (Router ID)** ——是指始发路由器的 ID。
- **区域 ID (Area ID)** ——是指始发报文的路由器所在的区域。如果报文是在一个虚链路上发送的, 那么区域 ID 就为 0.0.0.0, 也就是骨干区域的 ID, 因为虚链路被认为是骨干的一部分。
- **校验和 (Checksum)** ——是指一个对整个报文 (包括报文头部) 的标准 IP 校验和。
- **认证类型 (AuType)** ——是指正在使用的认证模式。表 9-8 中列出了可能的认证模式。

表 9-8

OSPF 认证类型

认证类型代码 (AuType)	认 证 类 型
0	空 (没有认证)
1	简单 (明文) 口令认证
2	加密校验和 (MD5)

- **认证 (Authentication)** ——是指报文认证的必要信息，认证可以是 AuType 字段中指定的任何一种认证模式。如果 AuType=0，将不检查这个认证字段，因此可以包含任何内容。如果 AuType=1，这个字段将包含一个最长为 64 位的口令。如果 AuType=2，这个认证字段将包含一个 Key ID、认证数据长度和一个不减小的加密序列号。这个消息摘要附加在 OSPF 报文的尾部，不作为 OSPF 报文本身的一部分。
- **密钥 ID (Key ID)** ——标识认证算法和创建消息摘要使用的安全密钥。
- **认证数据长度 (Authentication Data Length)** ——指明附加在 OSPF 报文尾部的消息摘要的长度，以 8bit 字节 (octet) 计。
- **加密序列号 (Cryptographic Sequence Number)** ——是一个不会减小的数字，用来防止重现攻击 (replay attacks)。

2. Hello 报文

如图 9-49 所示，Hello 报文是用来建立和维护邻接关系的。为了形成一个邻接关系，Hello 报文携带的参数必须和它的邻居一致。

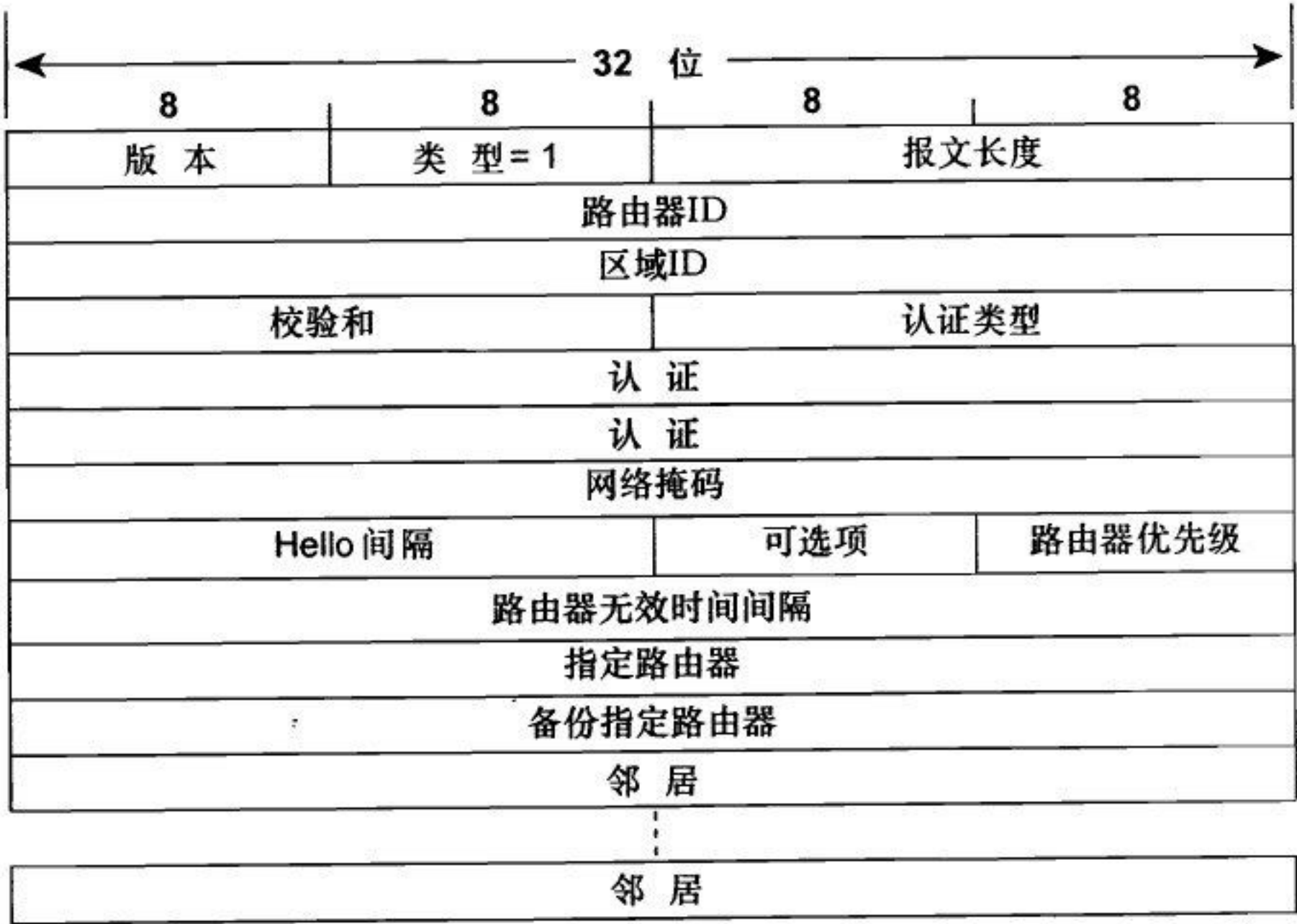


图 9-49 OSPF 协议 Hello 报文

- **网络掩码 (Network Mask)** ——是指发送报文的接口的网络掩码。如果这个掩码和接收该报文的接口的网络掩码不匹配，那么该报文将被丢弃。这个技术性的措施可以确保路由器之间只有在它们共享网络的地址精确匹配时才能互相成为邻居。
- **Hello 时间间隔 (Hello Interval)** ——和早期讲述的一样，是指接口上 Hello 报文传送之间的时间间隔，也是一个周期性的时间段，并以秒来计。对于这个参数，如果发送和接收路由器没有相同的值，那么它们就不能建立一个邻居关系。
- **可选项 (Option)** ——这个字段将在本章后面的“可选项字段”一节中讲述。Hello 报文中包含的这个字段可以用来确保邻居之间的兼容性问题。一台路由器可以拒绝

一个兼容性不匹配的邻居路由器。

- **路由器优先级 (Router Priority)** ——是用来做 DR 和 BDR 路由器的选取的。如果该字段设置为 0, 那么始发路由器将没有资格选取成为 DR 和 BDR 路由器。
- **路由器无效时间间隔 (Router Dead Interval)** ——是指始发路由器在宣告邻居路由器无效之前, 将要等待的从邻居路由器发出的 Hello 报文的时长, 以秒数计。如果路由器收到的 Hello 报文所带的这个秒数和接收接口配置的 RouterDeadInterval 不匹配, 那么这个 Hello 报文将被丢弃。这种做法可以确保邻居之间的这个参数的一致性。
- **指定路由器 (DR)** ——是指网络上指定路由器的接口的 IP 地址, 注意, 这里指的不是指定路由器的路由器 ID。在选取 DR 的过程中, 这可能只是始发路由器所认为的 DR, 而不是最终选取出来的 DR。如果没有 DR (因为 DR 可能还没有选出或者网络类型根本不需要 DR), 那么这个字段就会被设置为 0.0.0.0。
- **备份指定路由器 (BDR)** ——是指网络上备份指定路由器的接口的 IP 地址。同样, 在选取 BDR 的过程中, 这可能只是始发路由器所认为的 BDR, 而不是最终选取出来的 BDR。如果没有 BDR, 那么这个字段就会被设置为 0.0.0.0。
- **邻居 (Neighbor)** ——是一个循环重复的字段, 它列出了始发路由器在过去的一个 RouterDeadInterval 时间内收到有效 Hello 报文的网络上的所有邻居。

3. 数据库描述报文

如图 9-50 所示, 数据库描述报文用于正在建立的邻接关系 (请参考本章前面介绍的“建立一个邻接关系”一节)。数据库描述报文的一个主要目的是描述始发路由器数据库中的一些或者全部 LSA 信息, 以便接收路由器能够确定所收到的 LSA 在它自己数据库中是否已经有一个匹配的 LSA。这个操作只需要列出 LSA 的头部就可以完成了。由于在这个处理过程中, 可能需要交换多个数据库描述报文, 因此数据库描述报文中包含了一个主/从控制关系的标志, 用来管理这些报文的交换。

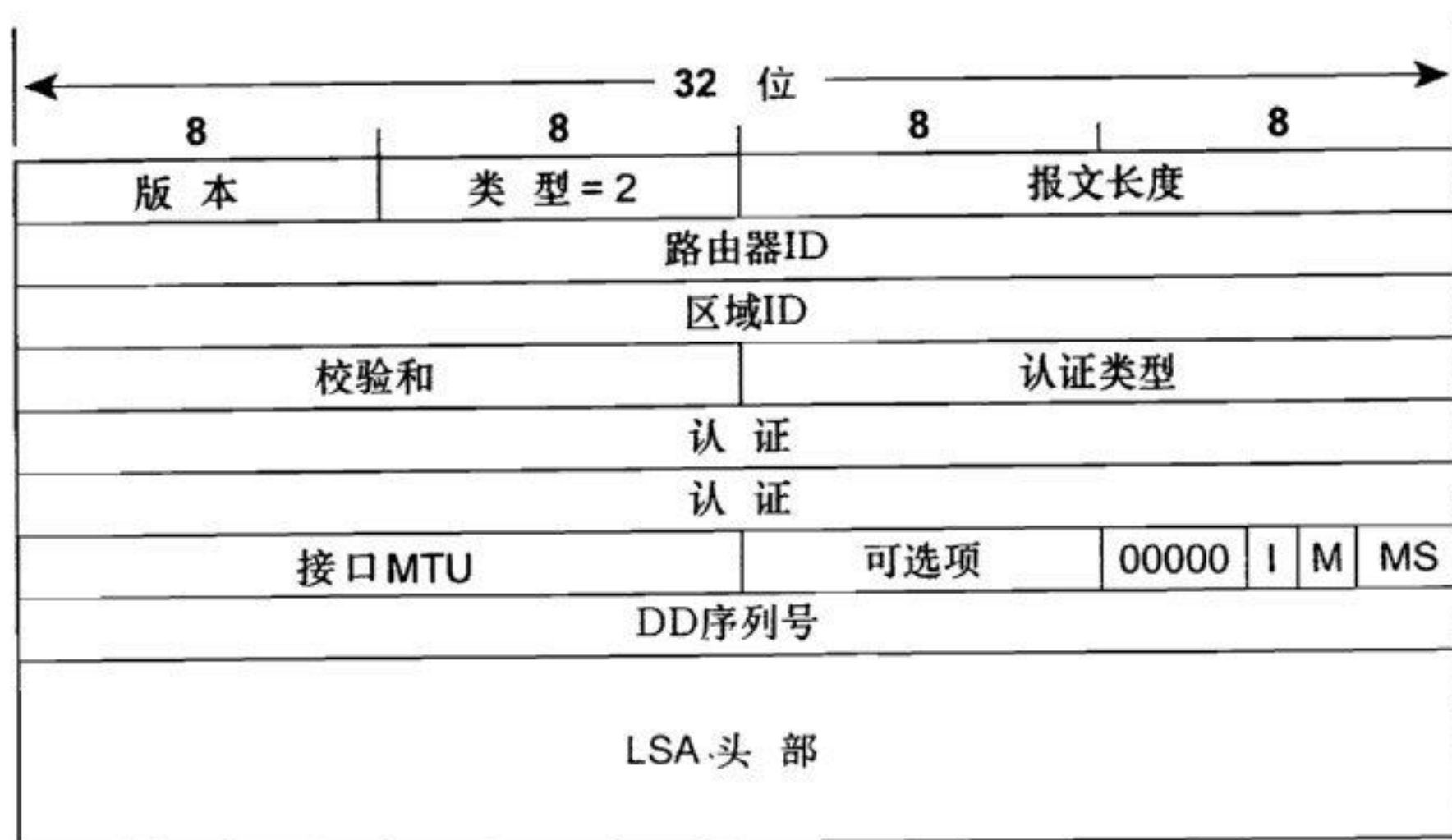


图 9-50 OSPF 数据库描述报文

- **接口 MTU (Interface MTU)** ——是指在报文不分段的情况下, 始发路由器接口可

以发送的最大 IP 报文大小, 以 8bit 字节 (octet) 计。当报文在虚链路上传送时, 这个字段的值设置为 0x0000。

- **可选项 (Option)** ——这个字段将在本章后面的“可选项字段”一节中讲述。该字段包含在数据库描述报文中, 使路由器可以选择不转发某些 LSA 到那些没有必要的支持能力的邻居路由器。

报文下一个 8bit 字节的前 5 位没有被使用而总是设置为 00000b。

- **I 位, 或称为初始位 (Initial bit)** ——当发送的是一系列数据库描述报文中的最初一个报文时, 该位设置为 1。后续的数据库描述报文将把该位设置为 0, 即 I-bit=0。
- **M 位, 或称为后继位 (More bit)** ——当发送的报文还不是一系列数据库描述报文中的最后一个报文时, 将该位设置为 1。最后的一个数据库描述报文将把该位设置为 0, 即 M-bit=0。
- **MS 位, 或称为主/从位 (Master/Slave bit)** ——在数据库同步的过程中, 该位设置为 1, 用来指明始发数据库描述报文的路由器是一个“主”路由器 (也就是说, 是主从关系协商过程的控制者)。“从”路由器将该位设置为 0, 即 MS-bit=0。
- **数据库描述序列号 (DD Sequence Number)** ——在数据库的同步过程中, 用来确保路由器能够收到完整的数据库描述报文序列。这个序列号将由“主”路由器在最初发送的数据库描述报文中设置一些惟一的数值, 而后续报文的序列号将依次增加。
- **LSA 头部 (LSA Header)** ——列出了始发路由器的链路状态数据库中部分或全部 LSA 头部。参见“链路状态头部”, 那里有一个关于 LSA 头部的完整描述。在 LSA 头部里包含有足够的信息可以惟一地标识一个 LSA 和一个 LSA 的具体实例。

4. 链路状态请求报文

在数据库同步过程中如果收到了数据库描述报文, 路由器将会查看数据库描述报文里有哪些 LSA 不在自己的数据库中, 或者有哪些 LSA 比自己数据库中的 LSA 更新。然后, 将把这些 LSA 记录在链路状态请求列表中。接着, 路由器会发送一个或多个链路状态请求报文去向它的邻居请求发送这些 LSA 的拷贝, 如图 9-51 所示。这里要注意, 一个报文可以根据一个 LSA 头部的类型、ID 和通告路由器进行唯一的标识, 但是它不能请求这个 LSA 的具体实例 (LSA 的具体实例由 LSA 头部的序列号、校验和以及老化时间标识)。因此, 不论请求路由器是否知道是 LSA 的哪个具体实例, 它所请求的都是 LSA 的最新实例。

- **链路状态类型 (Link State Type)** ——是一个链路状态类型号, 用来指明 LSA 标识是一个路由器 LSA、一个网络 LSA 还是其他类型的 LSA 等等。表 9-4 中列出了这些类型号。
- **链路状态 ID (Link State ID)** ——是 LSA 头部中和类型无关的字段。请参见“链路状态头部”和具体介绍 LSA 的章节可以得到关于不同类型的 LSA 是怎样使用该字段的完整描述。
- **通告路由器 (Advertising Router)** ——是指始发 LSA 通告的路由器的路由器 ID。

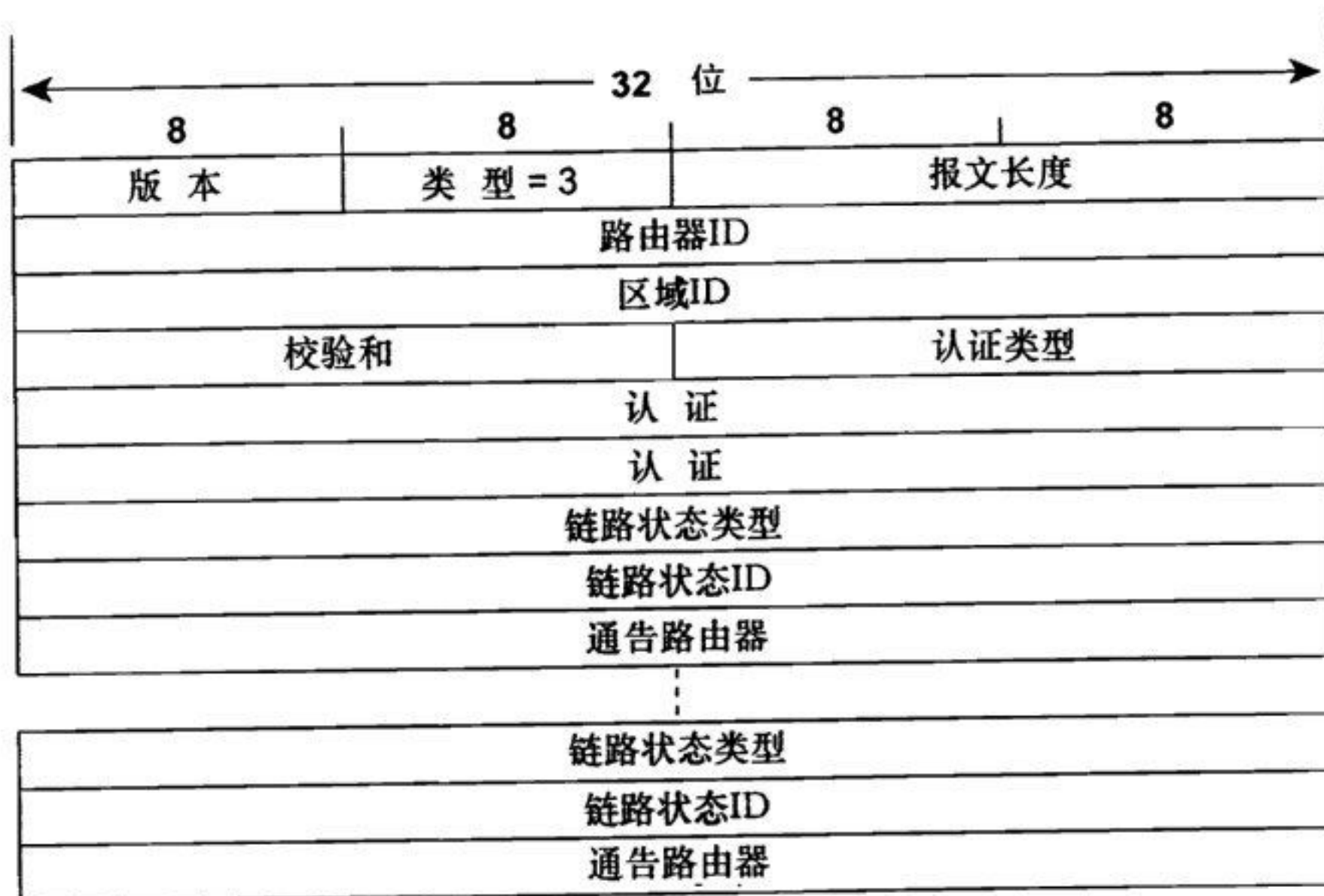


图 9-51 OSPF 链路状态请求报文

5. 链路状态更新报文

如图 9-52 所示, 链路状态更新报文是用于 LSA 的泛洪和发送 LSA 去响应链路状态请求报文的。请记住, OSPF 报文是不能离开发起它们的网络的。因此, 一个链路状态报文可以携带一个或多个 LSA, 但是只能携带这些 LSA 传送到始发它们的路由器的下一跳。接收 LSA 的邻居路由器将负责在新的链路状态报文中重新封装相关的 LSA, 从而进一步进行 LSA 的泛洪。

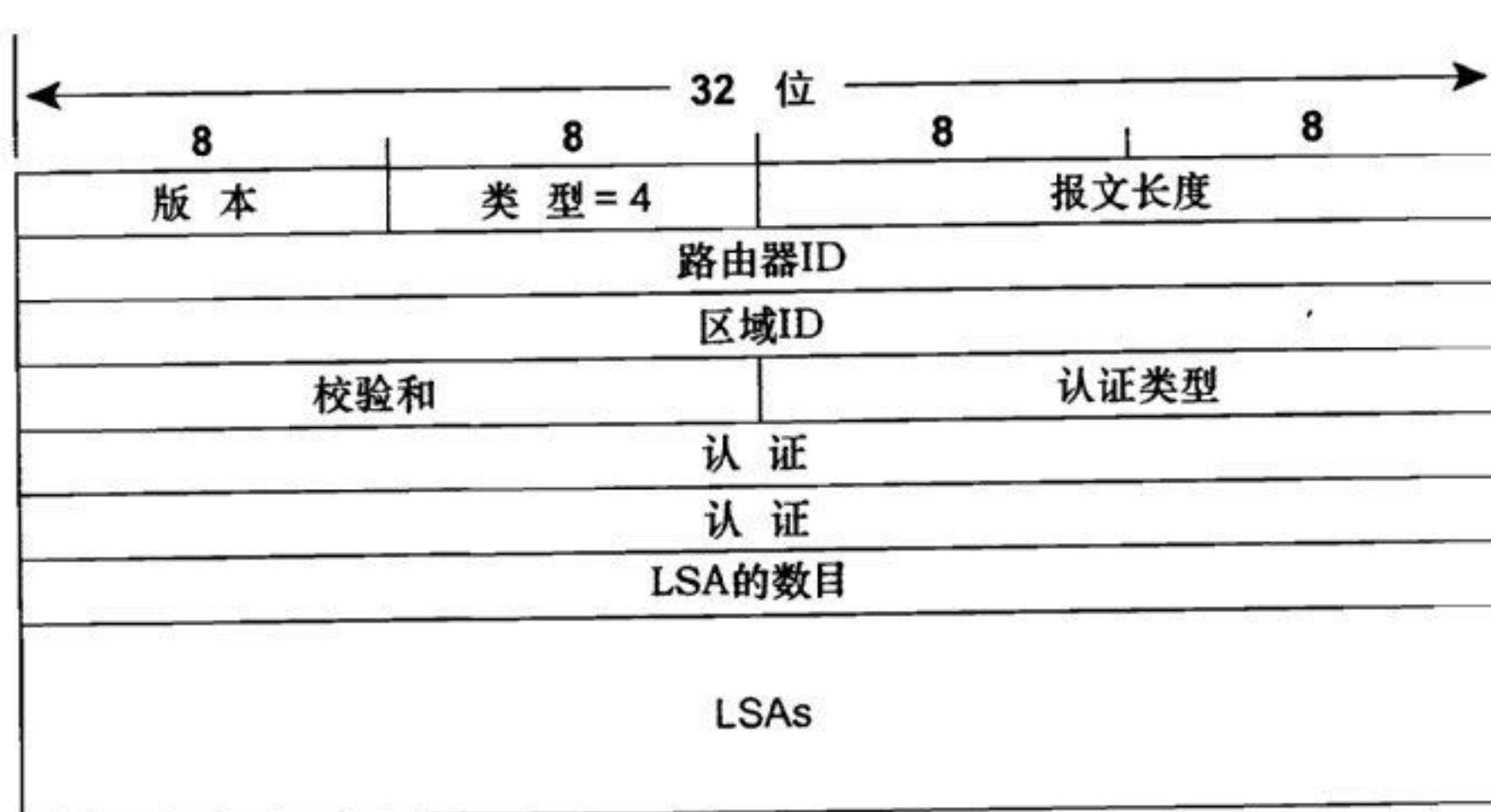


图 9-52 OSPF 链路状态更新报文

- **LSA 数量 (Number of LSA)** ——指出这个报文中包含的 LSA 的数量。
- **链路状态通告 (LSA)** ——是指在 OSPF 协议的 LSA 报文格式中描述的全部 LSA。每一个更新报文都可以携带多个 LSA, 它的大小可以达到传送这个报文的链路所允许的最大报文尺寸。

6. 链路状态确认报文

链路状态确认报文是用来进行 LSA 可靠的泛洪的。一台路由器从它的邻居路由器收到的每一个 LSA 都必须在链路状态确认报文中进行明确的确认。被确认的 LSA 是根据在链路状态确认报文里包含它的头部来辨别的，并且多个 LSA 可以通过单个报文来确认。正如图 9-53 所显示的，一个链路状态确认报文的组成除了 OSPF 报文头部和一个 LSA 头部的列表之外，就没有其他多余的内容了。

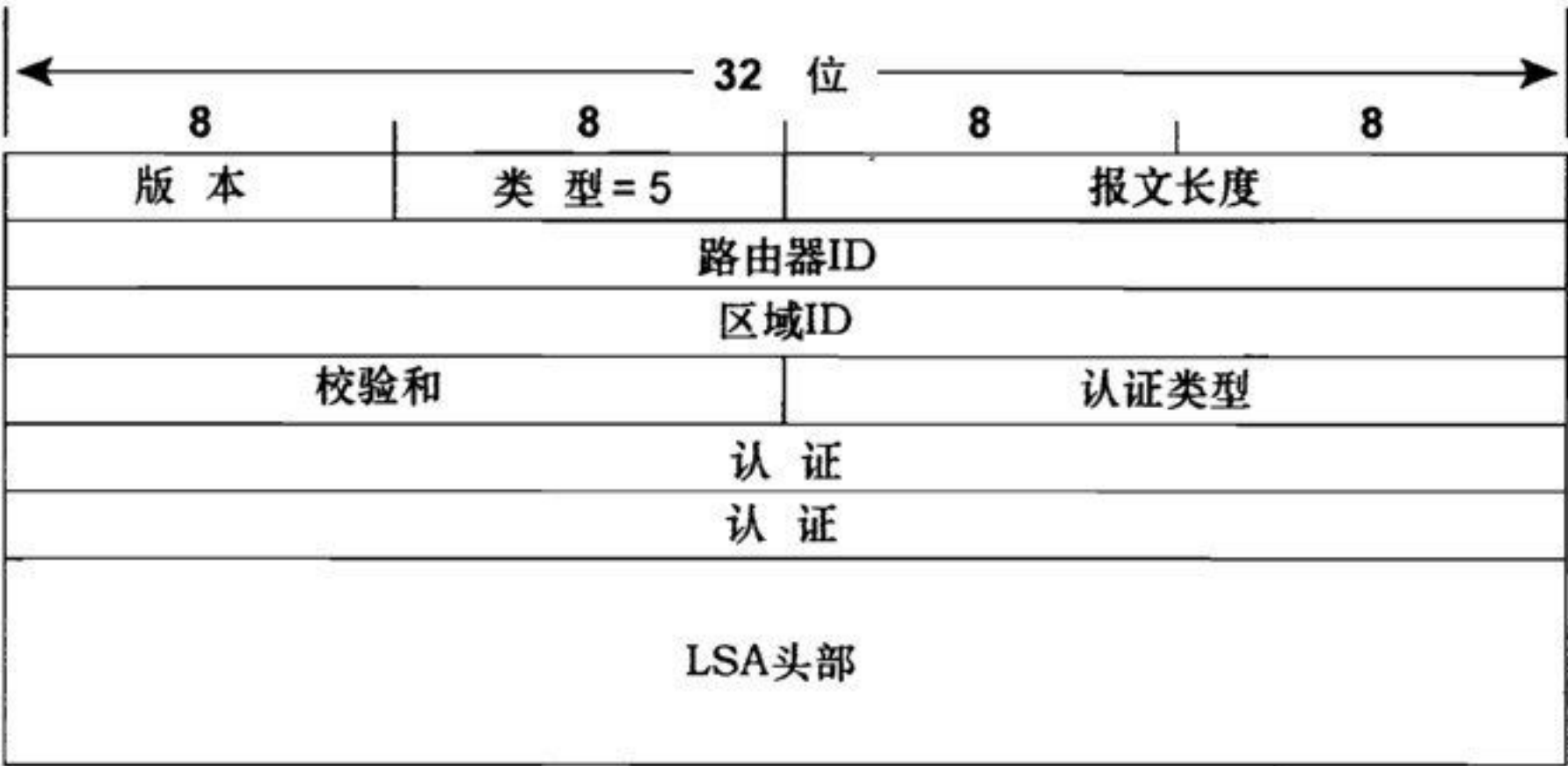


图 9-53 OSPF 链路状态确认报文

9.1.8 OSPF 的 LSA 格式

本节将详细描述每一种 LSA 类型的字段含义，但不包括组成员 LSA（类型 6）。因为 MOSPF 协议已经超出了本书的讲述范围，所以与之相关的 LSA 类型也就不在本书详细描述了。由于类型 8~类型 11 的 LSA 在目前还仅仅是提议，并且还没有部署，因此本书也没有涵盖这些内容。

1. LSA 的头部

LSA 头部在所有 LSA 的开始处，如图 9-54 所示。在数据库描述报文和链路状态确认报文里也使用了 LSA 的头部本身。在 LSA 头部中有 3 个字段可以惟一地识别每个 LSA：类型、链路状态 ID 和通告路由器。另外，还有其他 3 个字段可以惟一地识别一个 LSA 的最新实例：老化时间、序列号和校验和。



图 9-54 OSPF 协议 LSA 头部

- **老化时间 (Age)** ——是指自从发出 LSA 后所经历的时间, 以秒数计。当泛洪 LSA 时, 在从每一个路由器接口转发出来时, LSA 的老化时间都会增加一个 InfTransDelay 的秒数。当然, 当 LSA 驻留在链路状态数据库内时, 这个老化时间也会增大。
- **可选项 (Option)** ——这个字段将在本章后面的“可选项”一节中讲述。在 LSA 的头部中, 该字段指出了在部分 OSPF 域中 LSA 能够支持的可选项性能。
- **类型 (Type)** ——就是 LSA 的类型。一些类型的代码可以参见表 9-4。
- **链路状态 ID (Link State ID)** ——用来指定 LSA 所描述的部分 OSPF 域。这个字段的特殊用法根据 LSA 的类型而会有所不同。每一个 LSA 的描述都包含了一个怎样使用这个字段的描述。
- **通告路由器 (Advertising Router)** ——是指始发 LSA 的路由器的 ID。
- **序列号 (Sequence Number)** ——当 LSA 每次有新的实例产生时, 这个序列号就会增加。这个序列号的更新可以帮助其他路由器识别最新的 LSA 实例。
- **校验和 (Checksum)** ——这是一个除了 Age 字段之外, 关于 LSA 的全部信息的校验和。因为如果包含了 Age 字段, 那么这个校验和将会随着老化时间的增大而每次都需要进行重新计算。
- **长度 (Length)** ——是一个包含 LSA 头部在内的 LSA 的长度, 用 octet 表示。

2. 路由器 LSA

如图 9-55 所示, 路由器 LSA 是由每一台路由器产生的。它列出了一台路由器的链路或接口, 同时也列出了这些接口的状态和每一条链路的出站代价。这些路由器 LSA 只能在始发它们的 OSPF 区域内进行泛洪。使用命令 **show ip ospf database router** 可以列出链路状态数据库里的路由器 LSA (请参见图 9-30)。在这里要注意, 路由器 LSA 是把主机路由作为末梢网络来通告的, 它的链路 ID 字段携带的是主机的 IP 地址, 而链路数据字段携带的是主机地址的掩码——255.255.255.255。



图 9-55 OSPF 的路由器 LSA

- **链路状态 ID (Link State ID)** —— 路由器 LSA 的链路状态 ID 是指始发路由器的路由器 ID。
- **V, 或虚链路端点位 (Virtual Link Endpoint bit)** —— 设置为 1 时, 说明始发路由器是一条或多条具有完全邻接关系的虚链路的一个端点, 这里被描述的区域是传送区域。
- **E, 或外部位 (External bit)** —— 当始发路由器是一个 ASBR 路由器时, 设置该位为 1。
- **B, 或边界位 (Border bit)** —— 当始发路由器是一个 ABR 路由器时, 设置该位为 1。
- **链路数量 (Number of Links)** —— 标明一个 LSA 所描述的路由器链路数量。对于 LSA 进行泛洪的区域, 路由器 LSA 必须描述始发路由器的所有链路或接口。

在路由器 LSA 后续的字段里描述了每一条链路, 并且出现的次数和前面链路的数量字段中的数量是一致的。虽然这个字段是出现在链路数据字段之后的, 但是在这里将首先讲述链路类型字段。这是因为链路 ID 和链路数据字段的描述会根据链路类型字段的值而有所变化, 因此首先理解链路类型是必要的。

- **链路类型 (Link Type)** —— 描述了链路所提供的连接的一般类型。表 9-9 列出了这个字段可能的值和相关的连接类型。

表 9-9 链路类型的值

链 路 类 型	连 接
1	点到点连接到另一台路由器
2	连接到一个传送网络
3	连接到一个末梢网络
4	虚链路

- **链路 ID (Link ID)** —— 用来标识链路连接的对象。这个字段依赖于表 9-10 中的链路类型字段。注意, 当连接的对象是另一台路由器时, 链路 ID 和在邻居路由器的 LSA 头部的链路状态 ID 是相同的。在计算路由选择表的期间, 这个值可以用来发现链路状态数据库中邻居的 LSA。

表 9-10 链路 ID 的值

链 路 类 型	链路 ID 字段的值
1	邻居路由器的路由器 ID
2	DR 路由器的接口的 IP 地址
3	IP 网络或子网地址
4	邻居路由器的路由器 ID

- **链路数据 (Link Data)** —— 也是依赖于链路类型字段的值的字段, 如表 9-11 所示。

表 9-11 链路数据的值

链 路 类 型	链路数据字段的值
1	和网络相连的始发路由器接口的 IP 地址*
2	和网络相连的始发路由器接口的 IP 地址
3	网络的 IP 地址或子网掩码
4	始发路由器的接口的 MIB-II ifIndex 值

* 如果点到点链路是无编号的, 那么这个字段将替代携带接口的 MIB-II ifIndex 值。

- **TOS 号（Number of TOS）**——为列出的这条链路指定服务类型度量的编号。虽然 RFC2328 已经不再支持 TOS，但是为了向前兼容早期部署的 OSPF，仍旧保留这个字段。如果没有 TOS 度量和一条链路相关联，那么这个字段就设置为 0x00。
- **度量（Metric）**——是指一条链路（接口）的代价。

接下来的两个与链路相关联的字段是和 TOS 号（#）字段一致的。如果 TOS 的 #=3，那么将有 3 个 32 位字包含这些字段的 3 个实例。如果 TOS 的 #=0，那么将没有这些字段的实例。

注意：Cisco 路由器只支持 TOS=0。

- **TOS**——指定了后面提及的度量所提到的服务类型。¹表 9-12 列出了 TOS 的值（在 RFC1349 中指定的）、在 IP 报文头部中相应的 bit 值和在 OSPF TOS 字段中使用的相应值。

表 9-12 OSPF TOS 的值

RFC TOS 的值	IP 报文头部 TOS 字段	OSPF 的 TOS 编码
正常的服务	0000	0
最小的成本代价	0001	2
最大的可靠性	0010	4
最大的吞吐量	0100	8
最小的时延	1000	16

- **TOS 度量（TOS Metric）**——和指定的 TOS 值相关联的度量。

3. 网络 LSA

如图 9-56 所示，网络 LSA 是始发于指定路由器（DR）的。这些网络 LSA 将通告一个多路访问网络和与这个网络相连的所有路由器（包括 DR）。像路由器 LSA 一样，网络 LSA 也只能在始发这条网络 LSA 的区域内进行泛洪。可以使用 `show ip ospf database network` 来查看一条网络 LSA，如图 9-32 所示。

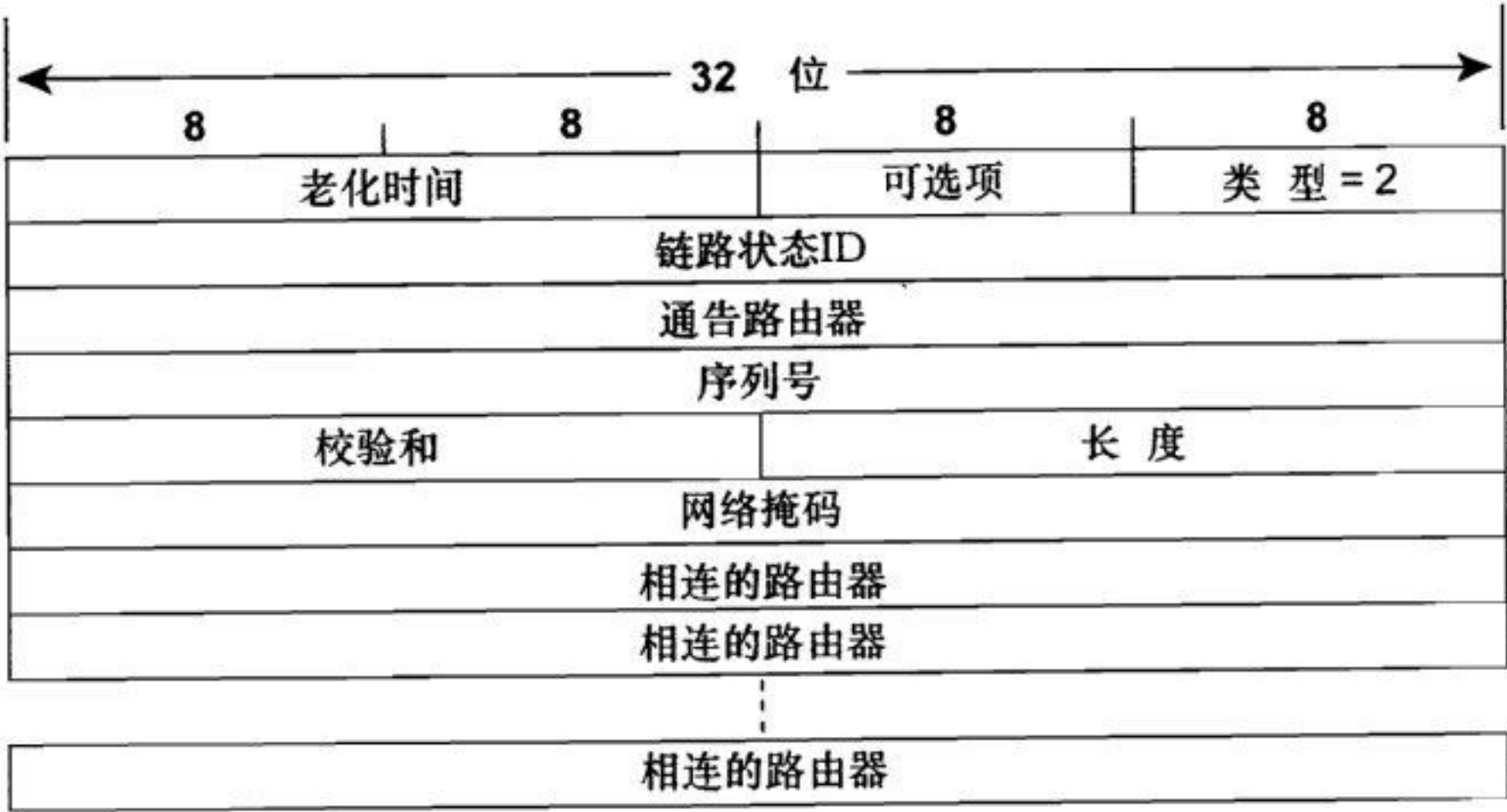


图 9-56 OSPF 的网络 LSA

¹ Philip Almquist, "Type of Service in the Internet Protocol Suite," RFC 1349, 1992 年 7 月。

- **链路状态 ID (Link State ID)** ——网络 LSA 的链路状态 ID 是指网络中 DR 路由器接口上的 IP 地址。
- **网络掩码 (Network Mask)** ——指定这个网络上使用的地址或子网的掩码。
- **关联路由器 (Attached Router)** ——列出了网络上所有与 DR 形成完全邻接关系的路由器的路由器 ID，以及 DR 路由器本身的路由器 ID。这个字段的实例的数量（因此也是列出的路由器的数量）可以由 LSA 头部的长度字段推断出来。

4. 网络汇总 LSA 和 ASBR 汇总 LSA

网络汇总 LSA（类型 3）和 ASBR 汇总 LSA（类型 4）具有同样的格式，如图 9-57 所示。在它们的字段内容里，惟一的不同之处是它们所指的类型和链路状态 ID。ABR 路由器将产生这两种类型的汇总 LSA。网络汇总 LSA 通告的是一个区域外部的网络（包括缺省路由），而 ASBR 汇总 LSA 通告的是一个区域外部的 ASBR 路由器。这两种类型的 LSA 都只能泛洪到单个区域。使用命令 `show ip ospf database summary` 可以查看一台路由器的链路状态数据库中的网络汇总 LSA，如图 9-34 所示。而使用命令 `show ip ospf database asbr-summary` 可以查看链路状态数据库中的 ASBR 汇总 LSA，如图 9-36 所示。

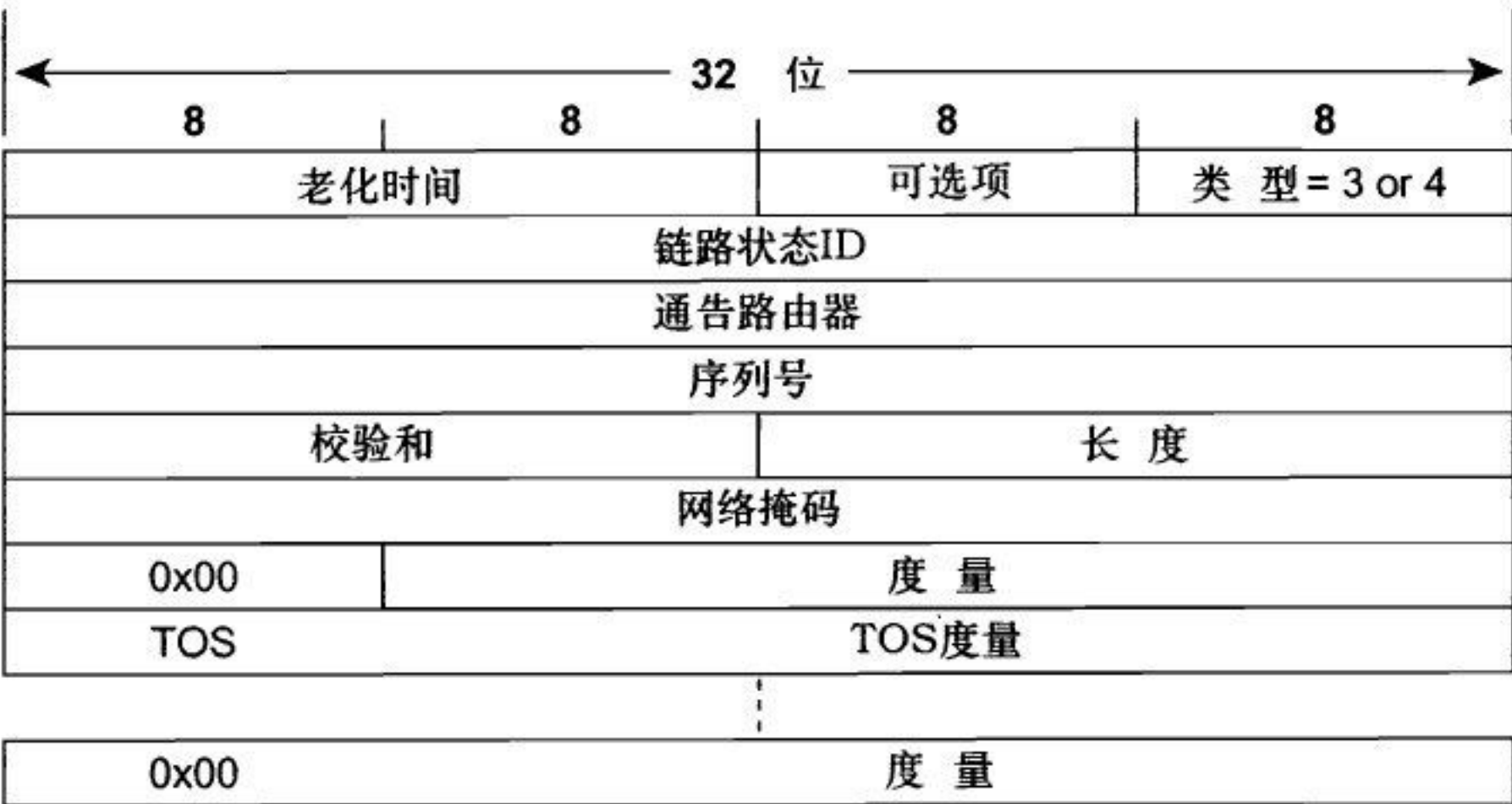


图 9-57 OSPF 的汇总 LSA。类型 3 和类型 4 的汇总 LSA 具有同样的格式

- **链路状态 ID (Link State ID)** ——对于类型 3 的 LSA 来说，它是所通告的网络或子网的 IP 地址。对于类型 4 的 LSA 来说，链路状态 ID 是所通告的 ASBR 路由器的路由器 ID。
 - **网络掩码 (Network Mask)** ——在类型 3 的 LSA 中，是指所通告的网络的子网掩码或地址。在类型 4 的 LSA 中，这个字段没有什么实际意义，并被设置为 0.0.0.0。如果一条类型 3 的 LSA 通告的是一条缺省路由，那么链路状态 ID 和网络掩码字段都将是 0.0.0.0。
 - **度量 (Metric)** ——是指到达目的地的路由的代价。
- TOS 字段和 TOS 度量字段都是可选字段，并且已经在“路由器 LSA”一节里描述过了。另外提醒一下，Cisco 路由器只支持 TOS=0。

5. 自主系统外部 LSA

如图 9-58 所示，自主系统外部 LSA 是由 ASBR 路由器始发的。这些自主系统外部 LSA

是用来通告 OSPF 自主系统外部的目的网络的, 这里也包括到达外部目的网络的缺省路由。自主系统外部 LSA 可以泛洪到 OSPF 域中所有非末梢的区域当中去。可以使用命令 **show ip ospf database external** 来查看 AS 外部 LSA, 如图 9-38 所示。

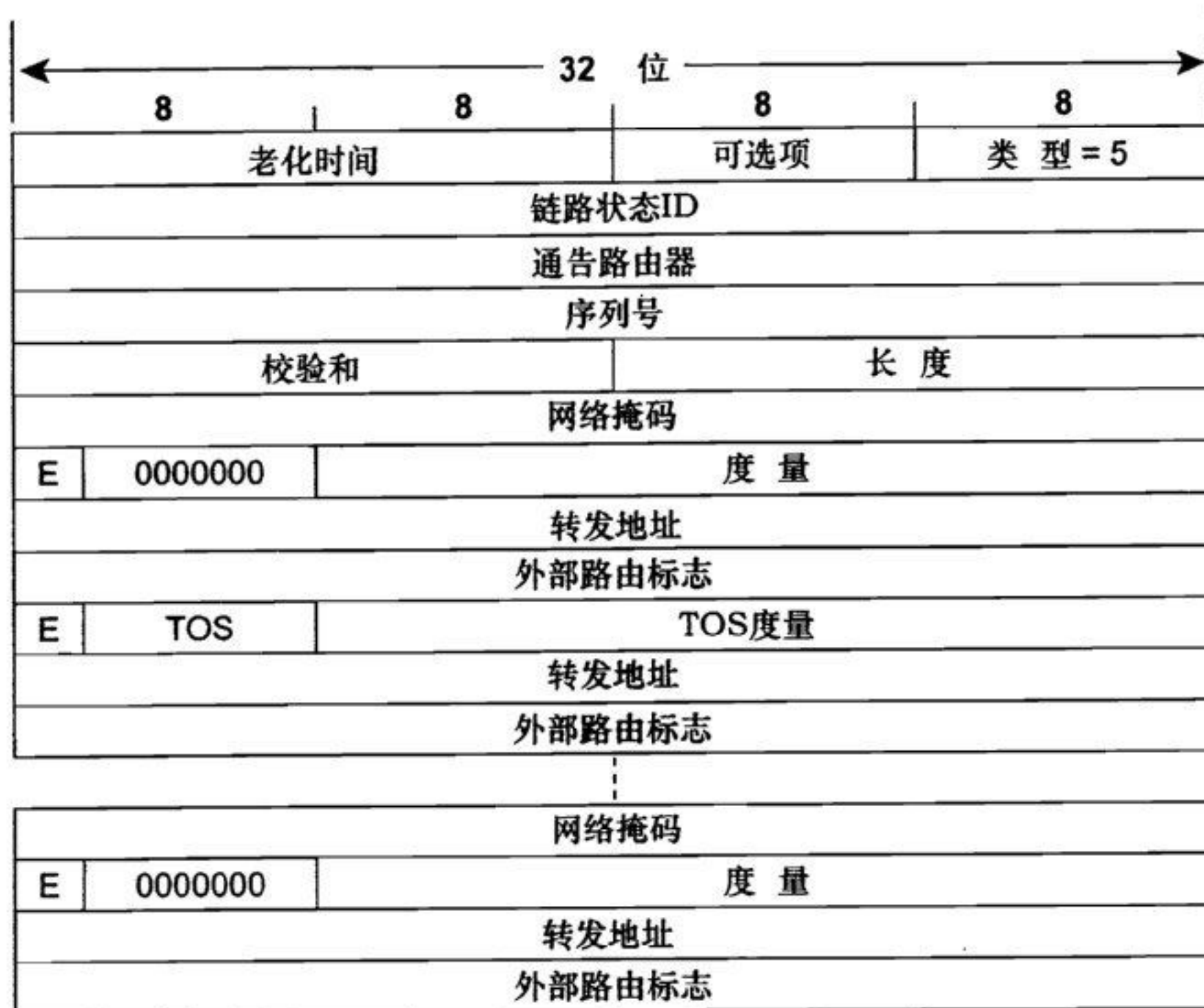


图 9-58 OSPF 的自主系统外部 LSA

- **链路状态 ID**——自主系统外部 LSA 的链路状态 ID 是指目的地的 IP 地址。
- **网络掩码**——是指所通告的目的地的子网掩码或地址。

如果类型 5 的 LSA 正在通告的是一条缺省路由, 那么链路状态 ID 和网络掩码字段都将被设置为 0.0.0.0。

- **E, 或称外部度量位 (External Metric bit)**——用来指定这条路由使用的外部度量的类型。如果该 E-bit 设置为 1, 那么度量类型就是 E2; 如果该 E-bit 设置为 0, 那么度量类型就是 E1。请参照本章前面讲述的“路径类型”一节, 那里可以找到关于 E1 和 E2 外部度量类型的更多信息。
- **度量**——是指路由的代价, 由 ASBR 路由器设定。
- **转发地址 (Forwarding Address)**——是指到达所通告的目的地的数据包应该被转发到的地址。如果转发地址是 0.0.0.0, 那么数据包将被转发到始发 ASBR 上。
- **外部路由标志 (External Route Tag)**——是一个应用于外部路由的任意标志。OSPF 协议本身并不使用这个字段, 而是由外部路由来管理和控制的。像这样的标志的设定和用法将在第 14 章“路由图”中介绍。

可选地, TOS 字段也可以和某个目的地相关联。这些字段和前面讲述的是相同的, 只是每一个 TOS 度量也都有自己的 E-bit、转发地址和外部路由标志。

6. NSSA 外部 LSA

NSSA 外部 LSA 是由一个 NSSA 区域内的 ASBR 路由器始发的。如图 9-59 所示, 除了

转发地址字段外, NSSA 外部 LSA 的所有字段都是和一个 AS 外部 LSA 的字段相同的。不像 AS 外部 LSA 那样是在整个 OSPF 自主系统中进行泛洪的, NSSA 外部 LSA 仅仅在始发它们的一个非纯末梢区域中进行泛洪。使用命令 **show ip ospf database nssa-external** 可以显示出 NSSA 外部 LSA 的信息, 如图 9-39 所示。



图 9-59 OSPF 的 NSSA 外部 LSA

- **转发地址**——如果网络是在一个 NSSA ASBR 路由器和邻接的自主系统之间是作为一条内部路由通告的, 那么这个转发地址就是指这个网络的下一跳地址。如果网络不是作为一条内部路由通告的, 那么这个转发地址将是 NSSA ASBR 路由器的路由器 ID。

9.1.9 可选项字段

如图 9-60 所示, 可选字段是出现在每一个 Hello 报文、数据库描述报文和每一个 LSA 中的。可选字段允许路由器和其他路由器进行一些可选性能的通信。



图 9-60 OSPF 的可选项字段

- **星号 (*) 位**——表明这一位是不使用的, 通常设置为 0。
- **DC 位**——当始发路由器具有支持按需电路上的 OSPF 的能力时, 该位将被设置。
- **EA 位**——当始发路由器具有接收和转发外部属性 LSA 的能力时, 该位将被设置。这些 LSA 还没有一般的用法, 因此本书将不介绍它们。
- **N 位**——只用在 Hello 报文中。一台路由器设置 N-bit=1 表明它支持 NSSA 外部 LSA。如果设置 N-bit=0, 那么路由器将不接受和发送 NSSA 外部 LSA。邻居路由器如果

错误配置了 N-bit 将不会形成邻接关系, 这个限制可以确保一个区域内的所有路由器都同样地具有支持 NSSA 的能力。如果 N-bit=1, 那么 E-bit 必须设置为 0。

- **P 位**——只用在 NSSA 外部 LSA 的头部 (由于这种情况, N-bit 和 P-bit 可以使用在同一位置)。这一位将告诉一个非纯末梢区域中的 ABR 路由器将类型 7 的 LSA 转换为类型 5 的 LSA。
- **MC 位**——当始发路由器具有转发 IP 组播数据包的能力时, 该位将被设置。这一位使用在 MOSPF 协议当中。
- **E 位**——当始发路由器具有接受 AS 外部 LSA 的能力时, 该位将被设置。在所有的 AS 外部 LSA 和所有始发于骨干区域以及非末梢区域的 LSA 里面, 该位将设置为 1。而在所有始发于末梢区域的 LSA 当中, 该位设置为 0。另外, 可以在 Hello 报文中使用该位来表明一个接口具有接收和发送类型 5 的 LSA 的能力。E-bit 配置错误的邻居路由器将不能形成邻接关系, 这个限制可以确保一个区域的所有路由器都同样地具有支持末梢区域的能力。
- **T 位**——当始发路由器具有支持 TOS 的能力时, 该位将被设置。

9.2 配置 OSPF

在一个大型的 IP 互联网络里, 有很多可用的 OSPF 选项和配置变量经常在 IGP 使用。然而, 也偶尔会听到这样一种观点, 认为在一个小型的互联网络里使用 OSPF 协议不是一个好的选择, 因为 OSPF 协议的配置显得“太复杂”了。这纯粹是无稽之谈。正如下面将要讲述的第一个配置案例中所显示的, 完成一个基本的 OSPF 配置并使 OSPF 协议可以正常地运行, 只需要在 **network** 命令中额外地敲几下键盘而已。如果对 OSPF 的操作已经有了相当的理解, 那么所敲入的这些额外的内容也显得十分直观和自然了。

9.2.1 案例研究 1: 一个基本的 OSPF 配置

配置一个基本的 OSPF 的过程含有以下 3 个必要的步骤:

步骤 1: 确定和每一个路由器接口相连的区域;

步骤 2: 使用 **router ospf process-id** 命令来启动一个 OSPF 进程;

步骤 3: 使用 **network area** 命令来指定运行 OSPF 协议的接口和它们所在的区域。

和 IGRP 与 EIGRP 协议中相关的进程 ID 不同, OSPF 协议的进程 ID 不是一个自主系统号。OSPF 的这个进程 ID 可以是任何正整数, 并且仅在配置它的路由器内有意义。Cisco IOS 软件允许同一台路由器中运行多个 OSPF 进程,¹ 进程 ID 不过是在同一台设备中用来区分一个进程与另一个不同的进程而已。

在前面讲述的路由选择协议里, 命令 **network** 只允许用来指定一个主网络地址。如果在这个网络中的一些接口不应该运行该路由选择协议的话, 那么就在这些协议中使用 **passive-interface** 命令来抑制这些接口。使用命令 **network area** 就比较灵活了, 它可以反映

¹ 虽然可以在同一台路由器上配置多个 OSPF 进程, 但是并不十分鼓励这么做, 因为这样的话, 运行多个数据库将要求占用较多的路由器资源。

完全的 OSPF 的无类别特性。任何一个地址范围都能够使用一个（地址，反向掩码）对来指定。这里的反向掩码（Inverse mask）和在访问列表中使用反向掩码是一样的。¹对于区域的指定，既可以使用一个十进制数字表示，也可以使用一个点分十进制来表示。

如图 9-61 显示了一个 OSPF 的互联网络。注意，在这里每一个区域都具有一个指定的 IP 地址，这个地址来源于它的子网。限制一个区域为一个地址或子网是不必要的，但是这样做会带来一些有效的好处，这在后面的一个案例研究“地址汇总”中将会看到。注意，这个例子也是设计用来演示多个区域的配置的。在现实的网络环境中，将图中这样的一个小网络设计成单个区域应该是一种更聪明的做法。更进一步来说，这里的单个区域也不必一定是区域 0。OSPF 的规则是所有的区域都必须连接到骨干区域。因此，只有当有多于一个的区域时才需要骨干区域。

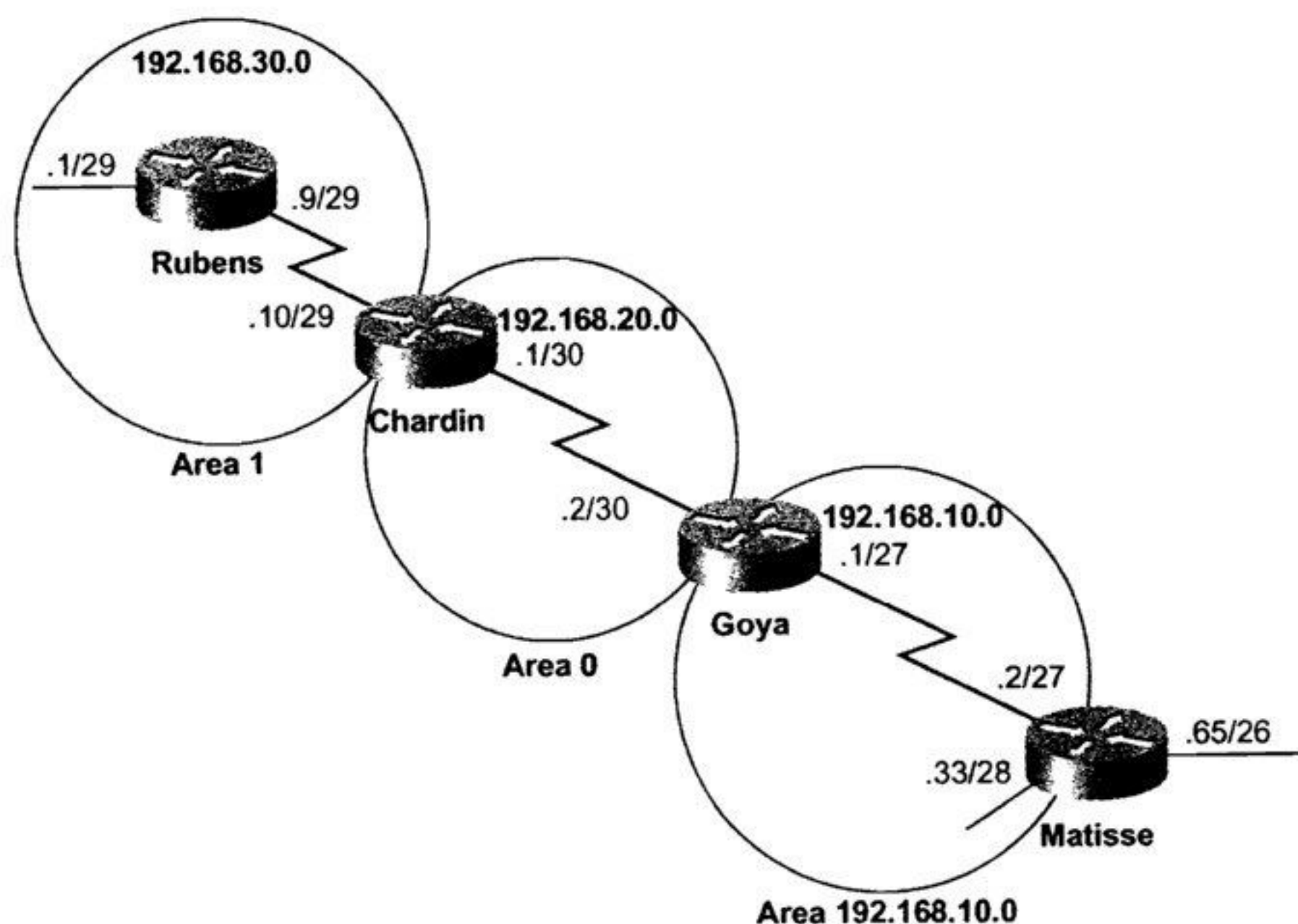


图 9-61 路由器 Chardin 和 Goya 都是 ABR 路由器，而路由器 Rubens 和 Matisse 是内部路由器

在图 9-61 中的 4 台路由器上，每一台路由器的配置都不同，这样做是为了更好地说明 **network area** 命令的灵活性。相关配置如下：

路由器 Rubens:

```
router ospf 10
network 0.0.0.0 255.255.255.255 area 1
```

路由器 Chardin:

```
router ospf 20
network 192.168.30.0 0.0.0.255 area 1
network 192.168.20.0 0.0.0.255 area 0
```

路由器 Goya:

```
router ospf 30
network 192.168.20.0 0.0.0.3 area 0.0.0.0
network 192.168.10.0 0.0.0.31 area 192.168.10.0
```

¹ 请参考附录 B “教程：访问列表”，那里有关于反向掩码的用法说明。

路由器 Matisse:

```
router ospf 40
 network 192.168.10.2 0.0.0.0 area 192.168.10.0
 network 192.168.10.33 0.0.0.0 area 192.168.10.0
```

在这里要注意的第一件事情就是,在每一台路由器上配置的进程 ID 都是不同的。一般情况下,为了保持在一个互连网络里配置的一致性,这些数字是相同的。而在这儿,配置不同的进程 ID 号只不过是为了演示这样一个事实——它们在本地路由器之外是没有意义的。当然,这 4 个不同编号的进程是可以相互通信的。

要注意的第二件事情就是 **network area** 命令的格式。紧跟在 **network** 之后的部分是一个 IP 地址和一个反向掩码。当 OSPF 进程第一次启动时,它将根据第一个网络语句(即第一个 **network** 语句的)的(地址,反向掩码)对来“启动”所有有效接口的 IP 地址。所有匹配的接口将被分配到根据 **network** 命令的 **area** 部分指定的区域。接着,这个过程根据第二条 **network** 语句继续匹配启动所有不匹配第一条 **network** 语句的接口。这样,根据一条条 **network** 语句运行 IP 地址的过程一直持续到所有的接口都匹配了,或者所有的 **network** 语句都使用了为止。这里要注意有一点非常重要,就是这个过程从第一条 **network** 语句开始是有顺序的处理的。结果,正如在后面故障排除一节中所描述的, **network** 语句的次序变得很重要。

路由器 Rubens 的 **network** 语句将匹配路由器上所有的接口。地址 0.0.0.0 实际上仅仅是一个占位符。在这儿,反向掩码 255.255.255.255 才是所有操作的核心。当“可以忽略”的位占据了所有 4 个 8bit 字节时,掩码将认为是和任何地址相匹配的,并且把相应的接口指定到区域 1 里面。这种方法提供了一种最粗略地控制接口运行 OSPF 的手段。

路由器 Chardin 是区域 1 和区域 0 之间的 ABR 路由器。这个事实是由在它们上面配置的 **network** 语句看出的。在这里,(地址,反向掩码)对将把与主网络 192.168.30.0 的任何子网相连的所有接口放置到区域 1 里,并把与主网络 192.168.20.0 的任何子网相连的所有接口放到骨干区域。

路由器 Goya 也是一个 ABR 路由器。在这里,(地址,反向掩码)对将只匹配两个接口上配置的具体子网。这里也要注意,骨干区域是用点分十进制来指定的。这种格式和路由器 Chardin 上配置的十进制格式是兼容的,它们都会使 OSPF 报文格式中相应的区域字段设置为 0x00000000。

路由器 Matisse 有一个接口 192.168.10.65/26,但这个接口并不运行 OSPF 协议。这台路由器上的 **network** 语句配置的是每个单独的接口地址,因而它的反向掩码指明所有 32 位都必须精确地匹配。这种方法提供了一种最精确地控制接口运行 OSPF 的手段。

最后,请注意虽然路由器 Matisse 的接口 192.168.10.65/26 没有运行 OSPF 协议,但是这个地址在数值上是该路由器上最高的 IP 地址。结果,路由器 Matisse 的路由器 ID 就是 192.168.10.65,如图 9-62 所示。

```
Matisse#show ip ospf 40
Routing Process "ospf 40" with ID 192.168.10.65
Supports only single TOS(TOS0) routes
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
```

待续


```

Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Area 192.168.10.0
  Number of interfaces in this area is 2
  Area has no authentication
  SPF algorithm executed 3 times
  Area ranges are
  Link State Update Interval is 00:30:00 and due in 00:27:59
  Link State Age Interval is 00:20:00 and due in 00:17:58
  Number of DCbitless LSA 1
  Number of indication LSA 1
  Number of DoNotAge LSA 0

```

Matisse#

图 9-62 使用命令 `show ip ospf process-id` 显示了指定进程的信息。这里的第一行显示的是路由器 ID 192.168.10.65

9.2.2 案例研究 2: 使用 Loopback 接口设置路由器的 ID

假设图 9-61 中的路由器 Matisse 已经在设备配置中心配置好了, 并且被发送到了要安装的地点。在这台路由器启动的过程中, 路由器报告它无法定位一个路由器 ID, 并且看上去好像在报告 `network area` 命令出现配置错误, 如图 9-63 所示。更麻烦的是, OSPF 命令也不再是可运行的配置了。

```

Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-J-L), Version 11.2(7a), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Tue 01-Jul-97 15:31 by kuong
Image text-base: 0x0303E1EC, data-base: 0x00001000

cisco 2509 (68030) processor (revision C) with 16384K/2048K bytes of memory.
Processor board ID 01210416, with hardware revision 00000000
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
TN3270 Emulation software.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read ONLY)

OSPF: Could not allocate router id
network 192.168.10.2 0.0.0.0 area 192.168.10.0
^
% Invalid input detected at '^' marker.

network 192.168.10.334 0.0.0.0 area 192.168.10.0
^
% Invalid input detected at '^' marker.

Press RETURN to get started!

```

图 9-63 如果找不到一个有效的 IP 地址作为它的路由器 ID, OSPF 将不会启动

这儿出现的问题是, 在路由器启动期间, 路由器上所有的接口都是管理关闭 (administratively shutdown) 的。如果 OSPF 不能发现一个有效的 IP 地址作为它的路由器 ID, 那么 OSPF 将不会启动。再进一步, 如果 OSPF 进程没有启动, 那么随后的 **network area** 命令也将是无效的。

解决这个问题的方法 (在这里假定关闭所有的物理接口是有合理的原因的) 是使用一个 loopback 接口 (环回接口)。Loopback 接口是一个仅在软件上有意义的、虚拟的接口, 并且它总是有效 (up) 的。因此, loopback 接口的 IP 地址也总是有效的。

在 OSPF 路由器上使用 loopback 接口还有一个更普遍的原因是这些接口运行网络管理员对路由器 ID 进行控制。当 OSPF 进程查找一个路由器 ID 时, OSPF 将越过所有物理接口的 IP 地址, 优先选用 loopback 接口的 IP 地址, 而且不论 IP 地址在数值上的高低次序。如果路由器具有多个带 IP 地址的 loopback 接口, 那么 OSPF 将选用在数值上最高的 loopback 地址。

控制路由器 ID 使单个 OSPF 路由器更加容易识别, 从而使网络的管理和故障排除更加容易。路由器 ID 的管理通常使用以下两种方法之一:

- 单独使用合法的网络或子网地址作为路由器 ID;
- 使用一段“伪造”的 IP 地址段。

第一种方法的缺点是需要使用合法的网络地址空间。第二种方法将可以节省合法的地址, 但是有一点要记住, 在一个互联网络里伪造的地址在另一个网络里可能是合法的地址。只要你记得这些地址不是合法的地址, 那么使用一些简单、易于识别的地址例如 1.1.1.1、2.2.1.1 等等将是比较好的做法。必须要小心的是, 伪造的地址千万不能泄漏到公共的 Internet 网络上去。

在前面一节的配置中使用 loopback 地址更改如下:

路由器 Rubens:

```
interface Loopback0
  ip address 192.168.50.1 255.255.255.255
!
router ospf 10
  network 192.168.30.0 0.0.0.255 area 1
```

路由器 Chardin:

```
interface Loopback0
  ip address 192.168.50.2 255.255.255.255
!
router ospf 20
  network 192.168.30.0 0.0.0.255 area 1
  network 192.168.20.0 0.0.0.255 area 0
```

路由器 Goya:

```
interface Loopback0
  ip address 192.168.50.3 255.255.255.255
!
router ospf 30
  network 192.168.20.0 0.0.0.3 area 0.0.0.0
  network 192.168.10.0 0.0.0.31 area 192.168.10.0
```


路由器 Matisse:

```
interface Loopback0
  ip address 192.168.50.4 255.255.255.255
!
router ospf 40
  network 192.168.10.2 0.0.0.0 area 192.168.10.0
  network 192.168.10.33 0.0.0.0 area 192.168.10.0
```

对于这个例子, 网络地址 192.168.50.0 独自使用来作为路由器 ID。因此, 在这个网络中, 路由器 ID 可以容易地和其他 IP 地址区分开来。

这里要注意的第一件事情是, 在这个配置中 loopback 地址所使用的地址掩码: 每一个掩码都配置成一个主机地址。这一步其实不是必要的, 因为 OSPF 会把一个 loopback 接口作为一个末梢网络来看待。无论 (地址, 掩码) 对配置成什么, loopback 接口的地址都将被作为一条主机路由来通告。主机掩码仅仅用来保持一种整齐的格式, 并且用来反映所通告的地址的一种方式。

然而, 第二个需要引起注意的地方和第一个有点不相关。请记住, OSPF 协议虽然使用一个接口的 IP 地址作为它的路由器 ID, 但是并不一定需要在这个接口上运行 OSPF。事实上, OSPF 所通告的 loopback 地址不过是创建了一个不必要的 LSA 而已。在上面一个例子的显示中要注意, 那里的 **network area** 语句并没有涉及到 loopback 地址。事实上, 路由器 Rubens 上的配置不得不更改。路由器 Rubens 在前面例子中的命令 **network 0.0.0.0 255.255.255.255 area 1** 应该已含有 loopback 地址。

另外, 为了对网络的管理和故障排除能有所帮助, 使用 loopback 接口也可以使一个 OSPF 网络更加稳定。如果一个作为路由器 ID 的物理接口出现了硬件故障¹, 或者这个接口被管理关闭 (administratively shutdown) 了, 或者这个 IP 地址被无意中删除了, 那么 OSPF 进程将必须获取一个新的路由器 ID。因此, 路由器必须过早地老化和泛洪它原来的 LSA, 并且接着要泛洪包含新的路由器 ID 的 LSA。Loopback 接口却没有硬件上的故障问题。但是, 如果 loopback 接口或它的 IP 地址被无意删除了, 路由器 ID 仍然会被重新计算和指定。但是, 改变一个 loopback 接口的可能性相对来说是很少的, 这是因为路由的配置操作不需要过多关注这个接口。

9.2.3 案例研究 3: 域名服务查询

loopback 地址由于可以提供预先设计好的路由器 ID, 从而使一个 OSPF 网络的管理和故障排除变得很简单。为了进一步得到简化, 可以把路由器 ID 记录到一个域名服务 (DNS) 数据库当中。在路由器上可以配置域名服务, 使路由器向一台域名服务器请求相关“地址到名称”的映射, 或称为反向 DNS 查找, 从而可以通过显示路由器的名称来代替路由器的 ID, 如图 9-64 所示。

路由器 Goya 可以作如下的配置来执行 DNS 的查找:

```
ip name-server 172.19.35.2
!
ip ospf name-lookup
```

¹ 如果仅仅是断开接口连接并不会引起路由器 ID 的改变。


```
Goya#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
chardin	1	FULL/ -	00:00:38	192.168.20.1	Serial0
matisse	1	FULL/ -	00:00:36	192.168.10.2	Serial1

```
Goya#show ip ospf database
```

```
OSPF Router with ID (192.168.50.3) (Process ID 30)
```

```
Router Link States (Area 0.0.0.0)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
192.168.50.2	chardin	151	0x80000097	0x1B3F	2
192.168.50.3	goya	1568	0x8000000C	0x2A1C	3

```
Summary Net Link States (Area 0.0.0.0)
```

Link ID	ADV Router	Age	Seq#	Checksum
192.168.10.0	goya	1568	0x80000009	0xA35E
192.168.10.33	goya	1568	0x80000009	0x1DA3
192.168.30.1	chardin	1058	0x80000009	0x6984
192.168.30.8	chardin	1059	0x80000009	0xEEFF

```
Router Link States (Area 192.168.10.0)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
192.168.50.3	goya	1569	0x8000001C	0xF9E8	2
192.168.50.4	matisse	688	0x8000000B	0xB597	3

```
--More--
```

图 9-64 OSPF 可以配置成使用 DNS 来实现路由器 ID 到名称的映射, 并可以使用一些 **show** 命令来显示

第一个命令用来指定一个 DNS 服务器的地址, 第二个命令用来启动 OSPF 进程执行 DNS 查找。在一些实际案例中, 一台路由器是通过一个接口地址来识别的, 而不是路由器 ID。这种情况下, 可以为路由器的这个接口增加一个条目到 DNS 数据库当中, 例如 rubens-e0。这样做也就可以使路由器通过这个接口的名字来识别了, 而这是与路由器 ID 不同的。

在这个例子中使用的域名服务器的地址是不属于图 9-61 中显示的某一个子网的。到达这个网络的方法将是下一个案例研究的主题。

9.2.4 案例研究 4: OSPF 和辅助地址

在一个 OSPF 的环境中, 辅助地址的用法有以下两个相关的规则:

- 只有在主网络或子网 (primary network or subnet) 也运行 OSPF 协议的时候, OSPF 才会通告一个辅助网络或辅助子网;
- OSPF 将把辅助地址看作是末梢网络 (这些网络上没有 OSPF 邻居), 从而不会在这些网络上发送 Hello 报文。因此, 在辅助网络上也就无法建立邻接关系;

图 9-65 中显示了一个 DNS 服务器, 并另外增添了一台路由器连接到路由器 Matisse 的 E0 接口上。这台 DNS 服务器和新加的路由器都放在子网 172.19.35.0/25 中, 并给路由器 Matisse 的 E0 接口分配一个辅助地址 172.19.35.15/25:


```

interface Ethernet0
  ip address 172.19.35.15 255.255.255.128 secondary
  ip address 192.168.10.33 255.255.255.240
!
router ospf 40
  network 192.168.10.2 0.0.0.0 area 192.168.10.0
  network 192.168.10.33 0.0.0.0 area 192.168.10.0
  network 172.19.35.15 0.0.0.0 area 192.168.10.0

```

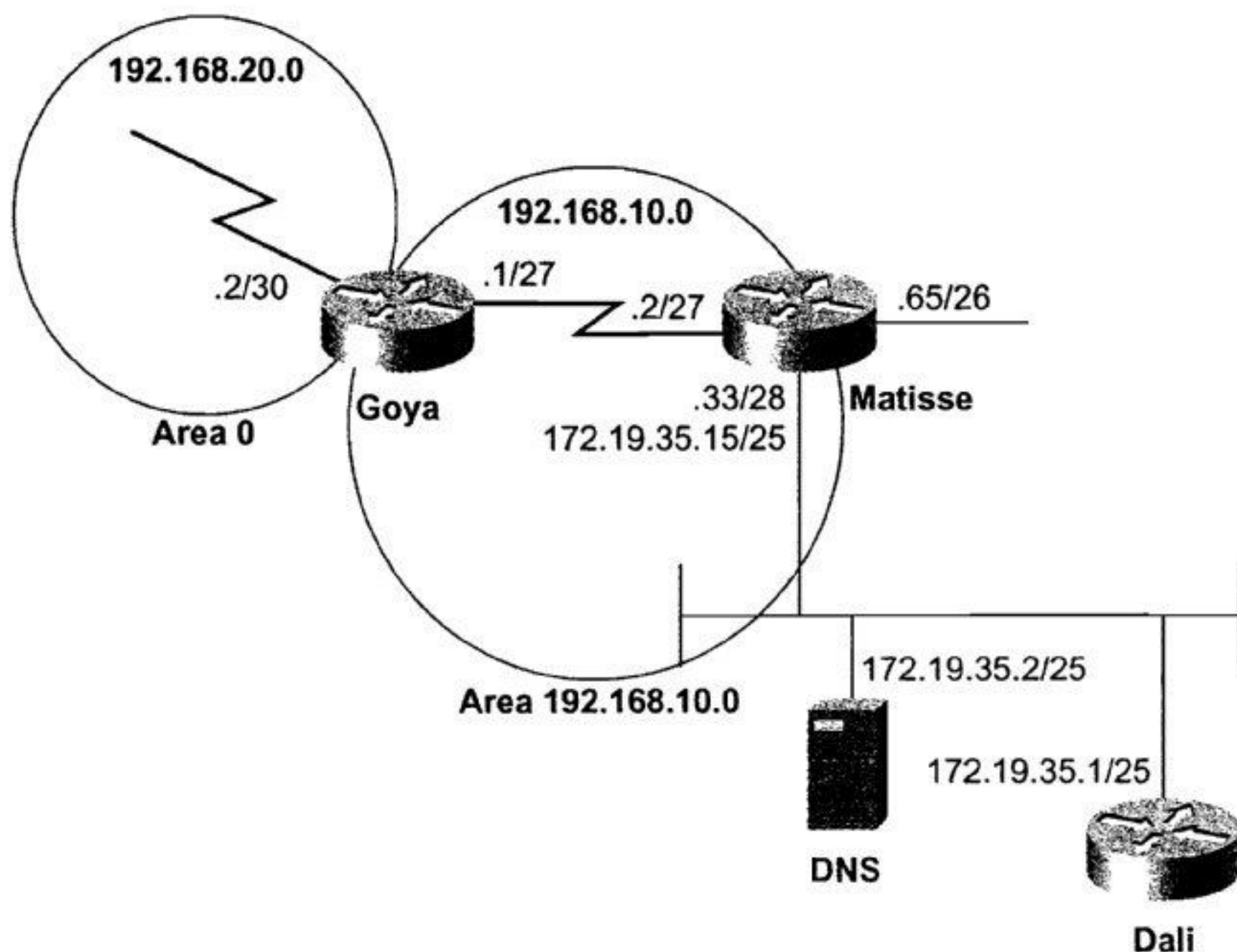


图 9-65 路由器 Dali 和 DNS 服务器都不是 OSPF 域的一部分，并且它们是通过一个辅助地址连接到路由器 Matisse 上的

根据这个配置，路由器 Matisse 将向它的邻居路由器通告子网 172.19.35.0/25。但是，如果关于主地址 192.168.10.33 的 **network area** 语句被删除了的话，那么子网 172.19.35.0/25 也将不再被通告了。

由于路由器 Matisse 是通过一个辅助地址和子网 172.19.35.0/25 相连的，因此它不能和这个子网上的任何路由器建立邻接关系，如图 9-66 所示。但是，DNS 服务器使用了路由器 Dali 作为它的缺省网关。因此，路由器 Matisse 和路由器 Dali 之间必须能够互相转发数据包。

到目前为止，对于上述的网络可以得出以下的结论：

- 子网 172.19.35.0/25 正在被通告到 OSPF 域中，一个目的地址是 172.19.35.2 的数据包将被转发到路由器 Matisse 的 E0 接口，并且从那儿直接转发到 DNS 服务器，如图 9-67 所示；
- 由于 DNS 服务器必须发一些回复 (reply) 到与它不在同一网段的网络，因而它会根据路由发送这些回复到路由器 Dali；
- 路由器 Dali 和路由器 Matisse 之间并不交换路由选择信息，因此它将不知道怎样到达 OSPF 域内的网络。

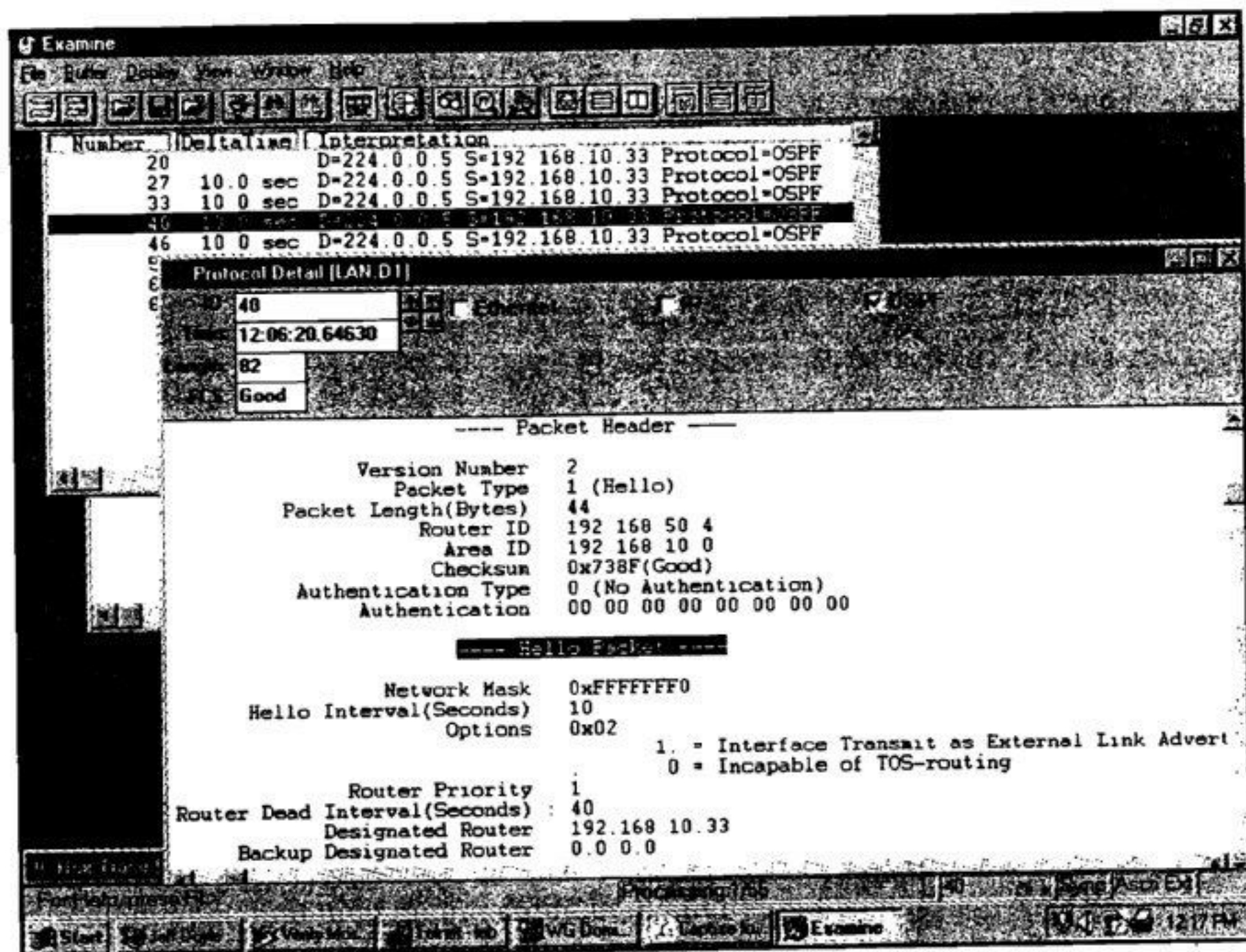


图 9-66 这台协议分析仪显示的信息是从与路由器 Matisse、Dali 和 DNS 服务器相连的网络上捕获得。小一点的窗口里面显示出 Hello 报文只能由路由器 Matisse 的主地址 192.168.10.33 发送出。大一点的窗口显示了一个 Hello 报文的解码信息

```
Matisse#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.10.33 - 0000.0c0a.2c51 ARPA Ethernet0
Internet 172.19.35.15 - 0000.0c0a.2c51 ARPA Ethernet0
Internet 172.19.35.1 167 0000.0c0a.2aa9 ARPA Ethernet0
Internet 172.19.35.2 26 0002.6779.0f4c ARPA Ethernet0
Matisse#
```

图 9-67 在路由器 Matisse 的 ARP 缓存中记录了 DNS 服务器的 MAC 地址标识, 这表明这台 DNS 服务器是可以直接到达的。但是, 如果到达 DNS 服务器的数据包必须通过路由器 Dali 路由转发才能到达的话, 那么在这个缓存中, 这台 DNS 服务器和路由器 Dali 的 MAC 地址将应该都是 0000.0c0a.2aa9

因此, 这里需要一个步骤去“连通一条电路”, 来告诉路由器 Dali 怎样才能到达 OSPF 域内的网络。这一步可以通过配置一条静态路由很容易得完成:

```
Dali(config)#ip route 192.168.0.0 255.255.0.0 172.19.35.15
```

这里需要注意的是, 静态路由是无类别的路由, 因而它可以使用超网的条目来匹配 OSPF 自主系统内的所有地址。

在这个例子当中, 路由器 Matisse 并不是一个 ASBR 路由器。这是因为, 虽然它发送了数据包到 OSPF 自主系统外部的目的地, 但是它并没有接受外部目的地的任何信息, 因此, 也没有始发任何类型 5 的 LSA 通告。

图 9-68 中显示了通过路由器 Dali 到达的一组新的目的地址。路由器 Matisse 现在就必须变成一个 ASBR 路由器, 并且要通告这些路由到 OSPF 域内了。但是, 它首先必须学习到这些路由。这个工作可以通过配置一系列静态路由或者运行一个在辅助网络上通信的路由选择协议来完成。无论在上述哪种情况下, 路由器都必须重新分配这些路由到 OSPF 域中。

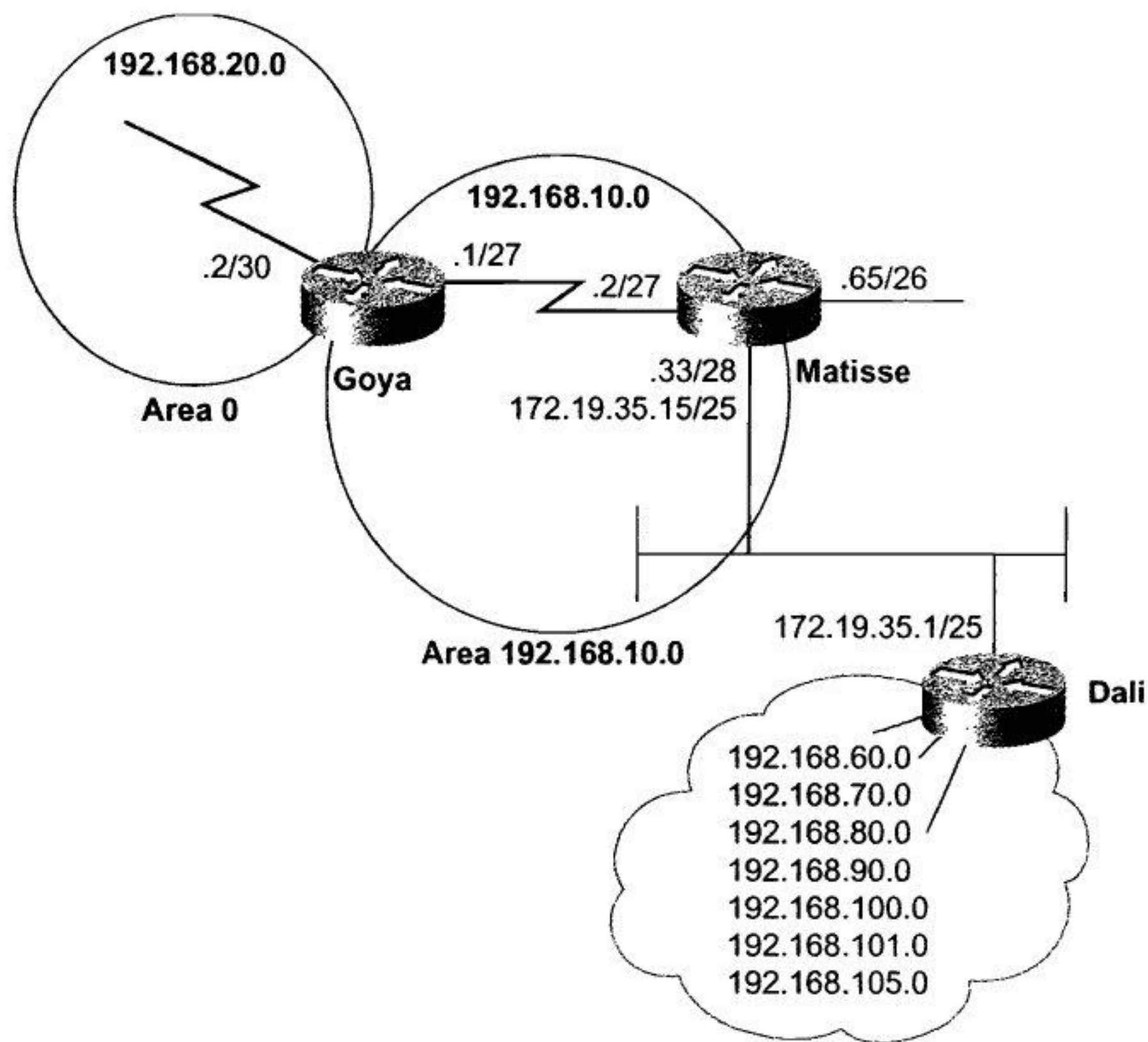


图 9-68 OSPF 自主系统必须学习到这些通过路由器 Dali 可达的目的地址，但是路由器 Matisse 到达路由器 Dali 的辅助地址妨碍了这两台路由器共享 OSPF 信息

RIP 协议在辅助地址上使用没有什么困难。因此，在路由器 Matisse 上可以选用 RIP 协议作为与路由器 Dali 的通信。路由器 Matisse 的配置如下：

```
interface Ethernet0
  ip address 172.19.35.15 255.255.255.128 secondary
  ip address 192.168.10.33 255.255.255.240
!
router ospf 40
  redistribute rip metric 10
  network 192.168.10.2 0.0.0.0 area 192.168.10.0
  network 192.168.10.33 0.0.0.0 area 192.168.10.0
!
router rip
  network 172.19.0.0
```

上述配置可以在 E0 接口的辅助网络上启用 RIP 协议，它可以让路由器 Matisse 从路由器 Dali 那里学习到路由，如图 9-69 所示。这些路由重新分配到 OSPF 域中（此时这些路由已不再运行在辅助地址上了），并且给它们指定一个 OSPF 的度量代价为 10，这是通过命令 **redistribute rip metric 10** 来实现的。关于路由重新分配的更加详细的介绍请参考第 11 章。如图 9-70 所示，这里通告到 OSPF 域内的路由是使用缺省的外部类型度量的，即外部类型 2 (E2)。注意观察在路由器 Rubens 上，这些路由的代价仍然是 10。路由器 Matisse 将使用类型 5 的 LSA 通告这些外部的目的地址，并使自己成为一个 ASBR 路由器（如图 9-71 所示）。


```

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

Gateway of last resort is not set

R 192.168.105.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R 192.168.100.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.101.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.70.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.90.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.80.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.60.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
  192.168.50.0/32 is subnetted, 1 subnets
C    192.168.50.4 is directly connected, Loopback0
  192.168.10.0/24 is variably subnetted, 3 subnets, 3 masks
C    192.168.10.64/26 is directly connected, Ethernet1
C    192.168.10.32/28 is directly connected, Ethernet0
C    192.168.10.0/27 is directly connected, Serial1
  192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
O IA 192.168.30.1/32 [110/193] via 192.168.10.1, 01:16:02, Serial1
O IA 192.168.30.8/29 [110/192] via 192.168.10.1, 01:16:02, Serial1
  192.168.20.0/30 is subnetted, 1 subnets
O IA 192.168.20.0 [110/128] via 192.168.10.1, 01:16:02, Serial1
  172.19.0.0/25 is subnetted, 1 subnets
C    172.19.35.0 is directly connected, Ethernet0

```

图 9-69 路由器 Dali 通过 RIP 协议传递它的路由选择信息给路由器 Matisse

```

Rubens#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
       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

Gateway of last resort is not set

O E2 192.168.105.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.100.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.101.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.70.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.90.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.80.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.60.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
  192.168.50.0/24 is subnetted, 1 subnets
C    192.168.50.1 is directly connected, Loopback1
  192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
O IA 192.168.10.32/28 [110/202] via 192.168.30.10, 02:01:21, Serial1
O IA 192.168.10.0/27 [110/192] via 192.168.30.10, 02:01:22, Serial1
  192.168.30.0/24 is subnetted, 2 subnets
C    192.168.30.0 is directly connected, Ethernet0
C    192.168.30.8 is directly connected, Serial1
  192.168.20.0/24 is subnetted, 1 subnets
O IA 192.168.20.0 [110/128] via 192.168.30.10, 02:01:22, Serial1
  172.19.0.0/16 is subnetted, 1 subnets
O IA 172.19.35.0 [110/202] via 192.168.30.10, 02:01:22, Serial1
Rubens#

```

图 9-70 通过 RIP 协议学习到的路由将作为路径类型 E2 的路由重新分配到 OSPF 自主系统当中


```
Rubens#show ip ospf border-routers

OSPF Process 10 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route
i   192.168.50.2 [64] via 192.168.30.10, Serial1, ABR, Area 1, SPF 60
I   192.168.50.4 [192] via 192.168.30.10, Serial1, ASBR, Area 1, SPF 60
Rubens#
```

图 9-71 路由器 Matisse (RID=192.168.50.4) 现在变成了一个 ASBR 路由器,

这是因为它正在始发自主系统外部 LSA 来通告外部路由

9.2.5 案例研究 5: 末梢区域

由于在区域 1 里面没有始发类型 5 的 LSA, 因而它可以配置成一个末梢区域。注意, 当一个相连的区域被配置成一个末梢区域时, 路由器始发的 Hello 报文进入那个区域后, 它的可选字段中的 E 位将会设置为 0, 即 E=0。其他没有同样配置的所有路由器收到这些 Hello 报文后将丢弃这些报文, 并且不能在这些路由器之间建立邻接关系。即使已经存在一个邻接关系, 它也会被阻断掉。因此, 如果需要把一个正在运作的区域重新配置成一个末梢区域的话, 应该计划好路由阻断的时间, 因此在这期间路由将被阻断, 一直到所有的路由器都重新配置后才能恢复。

一个末梢区域的配置可以通过在 OSPF 进程中增加命令 **area stub** 来完成, 具体配置如下:

路由器 Rubens:

```
router ospf 10
 network 0.0.0.0 255.255.255.255 area 1
 area 1 stub
```

路由器 Chardin:

```
router ospf 20
 network 192.168.30.0 0.0.0.255 area 1
 network 192.168.20.0 0.0.0.255 area 0
 area 1 stub
```

比较路由器 Rubens 在配置成末梢区域前 (如图 9-72) 后 (如图 9-73) 的链路状态数据库, 可以显示出所有的自主系统外部 LSA 和 ASBR 汇总 LSA 都已经从这个数据库当中清除了。在这种情况下, 数据库的大小也缩减了 50%。

```
Rubens#show ip ospf database database-summary

OSPF Router with ID (192.168.50.1) (Process ID 10)

Area ID      Router  Network  Sum-Net  Sum-ASBR  Subtotal  Delete  Maxage
1            2       0        4        1         7         0        0
AS External          7         0        0
Total        2       0        4        1        14
Rubens#
```

图 9-72 在区域 1 配置成末梢区域前, 路由器 Rubens 的数据库总共包括 14 条 LSA


```

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

```

Gateway of last resort is not set

```

R 192.168.105.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R 192.168.100.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.101.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.70.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.90.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.80.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
R 192.168.60.0/24 [120/1] via 172.19.35.1, 00:00:14, Ethernet0
  192.168.50.0/32 is subnetted, 1 subnets
C    192.168.50.4 is directly connected, Loopback0
  192.168.10.0/24 is variably subnetted, 3 subnets, 3 masks
C    192.168.10.64/26 is directly connected, Ethernet1
C    192.168.10.32/28 is directly connected, Ethernet0
C    192.168.10.0/27 is directly connected, Serial1
  192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
O IA 192.168.30.1/32 [110/193] via 192.168.10.1, 01:16:02, Serial1
O IA 192.168.30.8/29 [110/192] via 192.168.10.1, 01:16:02, Serial1
  192.168.20.0/30 is subnetted, 1 subnets
O IA 192.168.20.0 [110/128] via 192.168.10.1, 01:16:02, Serial1
  172.19.0.0/25 is subnetted, 1 subnets
C    172.19.35.0 is directly connected, Ethernet0

```

图 9-69 路由器 Dali 通过 RIP 协议传递它的路由选择信息给路由器 Matisse

```

Rubens#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
        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

```

Gateway of last resort is not set

```

O E2 192.168.105.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.100.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.101.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.70.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.90.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.80.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
O E2 192.168.60.0/24 [110/10] via 192.168.30.10, 01:21:35, Serial1
  192.168.50.0/24 is subnetted, 1 subnets
C    192.168.50.1 is directly connected, Loopback1
  192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
O IA 192.168.10.32/28 [110/202] via 192.168.30.10, 02:01:21, Serial1
O IA 192.168.10.0/27 [110/192] via 192.168.30.10, 02:01:22, Serial1
  192.168.30.0/24 is subnetted, 2 subnets
C    192.168.30.0 is directly connected, Ethernet0
C    192.168.30.8 is directly connected, Serial1
  192.168.20.0/24 is subnetted, 1 subnets
O IA 192.168.20.0 [110/128] via 192.168.30.10, 02:01:22, Serial1
  172.19.0.0/16 is subnetted, 1 subnets
O IA 172.19.35.0 [110/202] via 192.168.30.10, 02:01:22, Serial1
Rubens#

```

图 9-70 通过 RIP 协议学习到的路由将作为路径类型 E2 的路由重新分配到 OSPF 自主系统当中


```
Rubens#show ip ospf border-routers
```

```
OSPF Process 10 internal Routing Table
```

```
Codes: i - Intra-area route, I - Inter-area route
```

```
i 192.168.50.2 [64] via 192.168.30.10, Serial1, ABR, Area 1, SPF 60
```

```
I 192.168.50.4 [192] via 192.168.30.10, Serial1, ASBR, Area 1, SPF 60
```

```
Rubens#
```

图 9-71 路由器 Matisse (RID=192.168.50.4) 现在变成了一个 ASBR 路由器,

这是因为它正在始发自主系统外部 LSA 来通告外部路由

9.2.5 案例研究 5: 末梢区域

由于在区域 1 里面没有始发类型 5 的 LSA, 因而它可以配置成一个末梢区域。注意, 当一个相连的区域被配置成一个末梢区域时, 路由器始发的 Hello 报文进入那个区域后, 它的可选字段中的 E 位将会设置为 0, 即 E=0。其他没有同样配置的所有路由器收到这些 Hello 报文后将丢弃这些报文, 并且不能在这些路由器之间建立邻接关系。即使已经存在一个邻接关系, 它也会被阻断掉。因此, 如果需要把一个正在运作的区域重新配置成一个末梢区域的话, 应该计划好路由阻断的时间, 因此在这期间路由将被阻断, 一直到所有的路由器都重新配置后才能恢复。

一个末梢区域的配置可以通过在 OSPF 进程中增加命令 **area stub** 来完成, 具体配置如下:

路由器 Rubens:

```
router ospf 10
 network 0.0.0.0 255.255.255.255 area 1
 area 1 stub
```

路由器 Chardin:

```
router ospf 20
 network 192.168.30.0 0.0.0.255 area 1
 network 192.168.20.0 0.0.0.255 area 0
 area 1 stub
```

比较路由器 Rubens 在配置成末梢区域前 (如图 9-72) 后 (如图 9-73) 的链路状态数据库, 可以显示出所有的自主系统外部 LSA 和 ASBR 汇总 LSA 都已经从这个数据库当中清除了。在这种情况下, 数据库的大小也缩减了 50%。

```
Rubens#show ip ospf database database-summary
```

```
OSPF Router with ID (192.168.50.1) (Process ID 10)
```

Area ID	Router	Network	Sum-Net	Sum-ASBR	Subtotal	Delete	Maxage
1	2	0	4	1	7	0	0
AS External					7	0	0
Total	2	0	4	1	14		

```
Rubens#
```

图 9-72 在区域 1 配置成末梢区域前, 路由器 Rubens 的数据库总共包括 14 条 LSA


```
Rubens#show ip ospf database database-summary
```

OSPF Router with ID (192.168.50.1) (Process ID 10)

Area ID	Router	Network	Sum-Net	Sum-ASBR	Subtotal	Delete	Maxage
1	2	0	5	0	7	0	0
AS External					0	0	0
Total	2	0	5	0	7		

```
Rubens#
```

图 9-73 在配置成末梢区域后, 使路由器 Rubens 从它的数据库中清除了 7 条类型 5 的 LSA 通告和 1 条类型 4 的 LSA, 而只增加了一条通告缺省路由的类型 3 的 LSA

当一个末梢区域和一台 ABR 路由器相连时, 路由器将会通过一条网络汇总 LSA 自动地通告一个缺省路由 (目的地址是 0.0.0.0) 到这个区域。图 9-73 中所示的数据库汇总信息表明了这个额外的类型 3 的 LSA。路由器 Rubens 的路由选择表 (如图 9-74) 中的最后一个条目显示了这条缺省路由是由路由器 Chardin 通告的。

```
Rubens#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
 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

Gateway of last resort is 192.168.30.10 to network 0.0.0.0

```

192.168.50.0/24 is subnetted, 1 subnets
C      192.168.50.1 is directly connected, Loopback1
192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
O IA   192.168.10.32/28 [110/202] via 192.168.30.10, 00:05:43, Serial1
O IA   192.168.10.0/27 [110/192] via 192.168.30.10, 00:05:43, Serial1
192.168.30.0/24 is subnetted, 2 subnets
C      192.168.30.0 is directly connected, Loopback0
C      192.168.30.8 is directly connected, Serial1
192.168.20.0/24 is subnetted, 1 subnets
O IA   192.168.20.0 [110/128] via 192.168.30.10, 00:05:43, Serial1
172.19.0.0/16 is subnetted, 1 subnets
O IA   172.19.35.0 [110/202] via 192.168.30.10, 00:05:43, Serial1
O*IA 0.0.0.0/0 [110/65] via 192.168.30.10, 00:05:44, Serial1
Rubens#
```

图 9-74 路由器 Rubens 的路由选择表显示, 所有的外部路由都被清除了 (请将这里所显示的和图 9-70 显示的相比较), 但增加了一条缺省路由

ABR 路由器将通告代价为 1 的缺省路由。在路由器 Rubens 和 Chardin 之间的串行链路的代价是 64, 图 9-74 中显示了路由器 Rubens 到达缺省路由的总代价将是 $64+1=65$ 。这里的缺省代价可以通过命令 **area default-cost** 来改变。例如, 路由器 Chardin 可以配置成通告一条代价为 20 的缺省路由, 配置如下:

```

router ospf 20
 network 192.168.30.0 0.0.0.255 area 1
 network 192.168.20.0 0.0.0.255 area 0
 area 1 stub
 area 1 default-cost 20
```


结果,从图 9-75 中可以看出,这条缺省路由的代价增大了—— $64+20=84$ 。在这里改变缺省路由的代价没有什么实际意义,但是在多于一个 ABR 路由器的末梢区域里将可能会比较有用。通常情况下,每一台内部路由器都只会选择代价最低的缺省路由。通过操纵和控制所通告的代价,网络管理员可以促使所有的内部路由器都使用同一台 ABR 路由器。关于第二台 ABR 路由器,可以通告比较高的代价,而使它仅仅在第一台 ABR 路由器失效时才被使用。

```
Rubens#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
       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

Gateway of last resort is 192.168.30.10 to network 0.0.0.0

192.168.50.0/24 is subnetted, 1 subnets
C      192.168.50.1 is directly connected, Loopback1
192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
O IA   192.168.10.32/28 [110/202] via 192.168.30.10, 00:01:08, Serial1
O IA   192.168.10.0/27 [110/192] via 192.168.30.10, 00:01:08, Serial1
192.168.30.0/24 is subnetted, 2 subnets
C      192.168.30.0 is directly connected, Ethernet0
C      192.168.30.8 is directly connected, Serial1
192.168.20.0/24 is subnetted, 1 subnets
O IA   192.168.20.0 [110/128] via 192.168.30.10, 00:01:08, Serial1
172.19.0.0/16 is subnetted, 1 subnets
O IA   172.19.35.0 [110/202] via 192.168.30.10, 00:01:08, Serial1
O*IA 0.0.0.0/0 [110/84] via 192.168.30.10, 00:01:03, Serial1
Rubens#
```

图 9-75 路由器 Rubens 的路由选择表反映了更改缺省路由的代价以后的结果

9.2.6 案例研究 6: 完全末梢区域

完全末梢区域的配置可以通过在命令 **area stub** 的末端增加关键字 **no-summary** 来实现。这一步的配置操作只有在 ABR 路由器上才是必需的,在内部路由器上使用标准的末梢区域配置就可以了。为了在前面例子的网络中把区域 1 配置成一个完全末梢区域,路由器 Chardin 上的配置应该如下:

```
router ospf 20
 network 192.168.30.0 0.0.0.255 area 1
 network 192.168.20.0 0.0.0.255 area 0
 area 1 stub no-summary
```

如图 9-76 所示,可以看出路由器 Rubens 的数据库中 LSA 的数量已经减少到 3 条了。在图 9-77 中显示了它的路由选择表。

```
Rubens#show ip ospf database database-summary

OSPF Router with ID (192.168.50.1) (Process ID 10)

Area ID  Router  Network  Sum-Net  Sum-ASBR  Subtotal  Delete  Maxage
1         2         0        1        0         3         0       0
AS External
Total    2         0        1        0         3
Rubens#
```

图 9-76 改变区域 1 成为一个完全末梢区域,又消除了除了一条类型 3 的 LSA 之外的所有 LSA (缺省路由器)


```

Rubens#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
       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

Gateway of last resort is 192.168.30.10 to network 0.0.0.0

    192.168.50.0/24 is subnetted, 1 subnets
C      192.168.50.1 is directly connected, Loopback1
    192.168.30.0/24 is subnetted, 2 subnets
C      192.168.30.0 is directly connected, Ethernet0
C      192.168.30.8 is directly connected, Serial1
O*IA 0.0.0.0/0 [110/65] via 192.168.30.10, 00:03:33, Serial1
Rubens#

```

图 9-77 在完全末梢区域中的路由选择表将只包含区域内路由和缺省路由

9.2.7 案例研究 7: NSSA 区域

在前面的案例研究“OSPF 和辅助地址”中,讨论到路由器 Matisse 接受了通过 RIP 从路由器 Dali 学习到的路由,并把这些路由重新分配到 OSPF 域中(请参照图 9-68)。这一步操作使路由器 Matisse 成为一个 ASBR 路由器,更扩展一步来说,也使区域 192.168.10.0 无法满足成为一个末梢区域或完全末梢区域的条件了。然而,这里并不需要 AS 外部 LSA 从骨干区域通告到这个区域。因此,可以把区域 192.168.10.0 配置成一个 NSSA,在路由器 Matisse 上的配置如下:

```

router ospf 40
 redistribute rip metric 10
 network 192.168.10.2 0.0.0.0 area 192.168.10.0
 network 192.168.10.33 0.0.0.0 area 192.168.10.0
 area 192.168.10.0 nssa
!
router rip
 network 172.19.0.0

```

这里显示的 area nssa 语句可以同样地配置在路由器 Goya 上面。由于路由器 Goya 是一个 ABR 路由器,因此它将把与 NSSA 区域相连的接口收到的类型 7 的 LSA 转换成类型 5 的 LSA。这些转换过的 LSA 将泛洪到整个骨干区域中去,从而也泛洪到其他的区域中去。比较路由器 Goya 和 Chardin 的路由选择表,可以看出路由器 Goya 把外部路由标记成 NSSA1 (如图 9-78 所示)。而路由器 Chardin 把外部路由标记成 E2 (如图 9-79 所示),这表明它们是从类型 5 的 LSA 学到的。

这个转换也可以通过检查路由器 Goya 的链路状态数据库来观察。如图 9-80 所示,数据库中同时包含了相同外部路由的类型 7 和类型 5 的 LSA。只不过类型 7 的 LSA 是始发于路由器 Matisse 的,而类型 5 的 LSA 是始发于路由器 Goya 的。

¹ N2 表明的是和 E2 相同的度量计算——也就是说,只使用外部的代价。随后的一个例子将演示 E1 和 N1 的度量类型。


```
Goya#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 N2 192.168.105.0/24 [110/10] via 192.168.10.2, 00:38:32, Serial1
O N2 192.168.100.0/24 [110/10] via 192.168.10.2, 00:38:33, Serial1
O N2 192.168.101.0/24 [110/10] via 192.168.10.2, 00:38:33, Serial1
O N2 192.168.70.0/24 [110/10] via 192.168.10.2, 00:38:33, Serial1
O N2 192.168.90.0/24 [110/10] via 192.168.10.2, 00:38:33, Serial1
O N2 192.168.80.0/24 [110/10] via 192.168.10.2, 00:38:33, Serial1
O N2 192.168.60.0/24 [110/10] via 192.168.10.2, 00:38:33, Serial1
  192.168.50.0/32 is subnetted, 1 subnets
C    192.168.50.3 is directly connected, Loopback0
  192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
O    192.168.10.32/28 [110/74] via 192.168.10.2, 00:38:33, Serial1
C    192.168.10.0/27 is directly connected, Serial1
  192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
O IA  192.168.30.1/32 [110/129] via 192.168.20.1, 00:38:33, Serial0
O IA  192.168.30.8/29 [110/128] via 192.168.20.1, 00:38:35, Serial0
  192.168.20.0/30 is subnetted, 1 subnets
C    192.168.20.0 is directly connected, Serial0
```

图 9-78 从路由器 Matisse 学到的外部路由在路由器 Goya 上被标记成 NSSA 路由

```
Chardin#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
       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

Gateway of last resort is not set

O E2 192.168.105.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
O E2 192.168.100.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
O E2 192.168.101.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
O E2 192.168.70.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
O E2 192.168.90.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
O E2 192.168.80.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
O E2 192.168.60.0/24 [110/10] via 192.168.20.2, 00:00:58, Serial0
  192.168.50.0/24 is subnetted, 1 subnets
C    192.168.50.2 is directly connected, Loopback0
  192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
O IA  192.168.10.32/28 [110/138] via 192.168.20.2, 00:00:59, Serial0
O IA  192.168.10.0/27 [110/128] via 192.168.20.2, 00:01:10, Serial0
  192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
O    192.168.30.1/32 [110/65] via 192.168.30.9, 00:01:30, Serial1
C    192.168.30.8/29 is directly connected, Serial1
  192.168.20.0/24 is subnetted, 1 subnets
C    192.168.20.0 is directly connected, Serial0
```

图 9-79 路由器 Chardin 把同样的路由标记成 E2 类型，表明它们是从自主系统外部 LSA 学习到的

Type-7 AS External Link States (Area 192.168.10.0)						
Link ID	ADV Router	Age	Seq#	Checksum	Tag	
192.168.60.0	192.168.50.4	1476	0x800000E6	0xD907	0	
192.168.70.0	192.168.50.4	1485	0x800000E6	0x6B6B	0	
192.168.80.0	192.168.50.4	1494	0x800000E6	0xFCCF	0	
192.168.90.0	192.168.50.4	1503	0x800000E6	0x8E34	0	
192.168.100.0	192.168.50.4	1512	0x800000E6	0x2098	0	
192.168.101.0	192.168.50.4	1521	0x800000E6	0x15A2	0	
192.168.105.0	192.168.50.4	1530	0x800000E6	0xE8CA	0	
Type-5 AS External Link States						
Link ID	ADV Router	Age	Seq#	Checksum	Tag	
192.168.60.0	192.168.50.3	2695	0x80000001	0x4091	0	
192.168.70.0	192.168.50.3	2704	0x80000001	0xD1F5	0	
192.168.80.0	192.168.50.3	2713	0x80000001	0x635A	0	
192.168.90.0	192.168.50.3	2722	0x80000001	0xF4BE	0	
192.168.100.0	192.168.50.3	2731	0x80000001	0x8623	0	
192.168.101.0	192.168.50.3	2740	0x80000001	0x7B2D	0	
192.168.105.0	192.168.50.3	2749	0x80000001	0x4F55	0	
Goya#						

图 9-80 路由器 Goya 的链路状态数据库表明了来自路由器 Matisse (192.168.50.4) 的类型 7 LSA 已经被 Goya (192.168.50.3) 转换成类型 5 的 LSA 了

对于 ABR 路由器来说还有几个可用的配置选项。第一个是 **no-summary** 选项，可以和命令 **area nssa** 一起用来阻塞类型 3 和类型 4 的 LSA 泛洪到 NSSA 里面。这个配置可以使区域 192.168.10.0 变成一个名字有点怪异的区域——“完全非纯末梢区域” (Totally stubby not-so-stubby area)。这时，路由器 Goya 上的配置应该是：

```
router ospf 30
 network 192.168.20.0 0.0.0.3 area 0
 network 192.168.10.0 0.0.0.31 area 192.168.10.0
 area 192.168.10.0 nssa no-summary
```

路由器 Matisse 的路由选择表如图 9-81 所示，可以看出所有的区域间路由都被消除了，而另外增加了一条由路由器 Goya 通告的缺省路由。

在图 9-82 中，路由器 Dali 的链路从路由器 Matisse 移到 Goya 的接口上，相关的 IP 地址也作相应的变动。路由器 Goya 现在成为了一个 ASBR 路由器，并把 RIP 学习到的路由重新分配到 OSPF 当中。

当一台 ABR 路由器也是一台 ASBR 路由器，并且和一个非纯末梢区域相连时，它的缺省行为是通告重新分配的路由到这个 NSSA 中去，如图 9-83 所示。

这个在 ABR/ASBR 路由器上缺省的路由重新分配行为可以通过在命令 **area nssa** 之后增加参数 **no-redistribution** 来关闭。这样，在所述例子的网络中，就不应该有类型 3、4、5 或 7 的 LSA 从 ABR 路由器发送到区域 192.168.10.0。所需要的路由重新分配可以通过下面的配置在路由器 Goya 上完成：¹

¹ 注意，在 **redistribution** 命令中使用了 **metric-type 1** 参数。这个参数的作用是使外部的目的路由利用 E1 度量来通告。在 NSSA 中，度量类型变成了 N1，如图 9-82 所示。


```

Matisse#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 192.168.10.1 to network 0.0.0.0

R    192.168.105.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.100.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.101.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.70.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.90.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.80.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.60.0/24 [120/1] via 172.19.35.1, 00:00:13, Ethernet0
R    192.168.50.0/32 is subnetted, 1 subnets
C      192.168.50.4 is directly connected, Loopback0
R    192.168.10.0/24 is variably subnetted, 3 subnets, 3 masks
C      192.168.10.64/26 is directly connected, Loopback1
C      192.168.10.32/28 is directly connected, Ethernet0
C      192.168.10.0/27 is directly connected, Serial1
R    172.19.0.0/25 is subnetted, 1 subnets
C      172.19.35.0 is directly connected, Ethernet0
O*IA 0.0.0.0/0 [110/65] via 192.168.10.1, 00:36:50, Serial1
Matisse#

```

图 9-81 所有的区域间路由都被一条到达 ABR 的缺省路由替代了

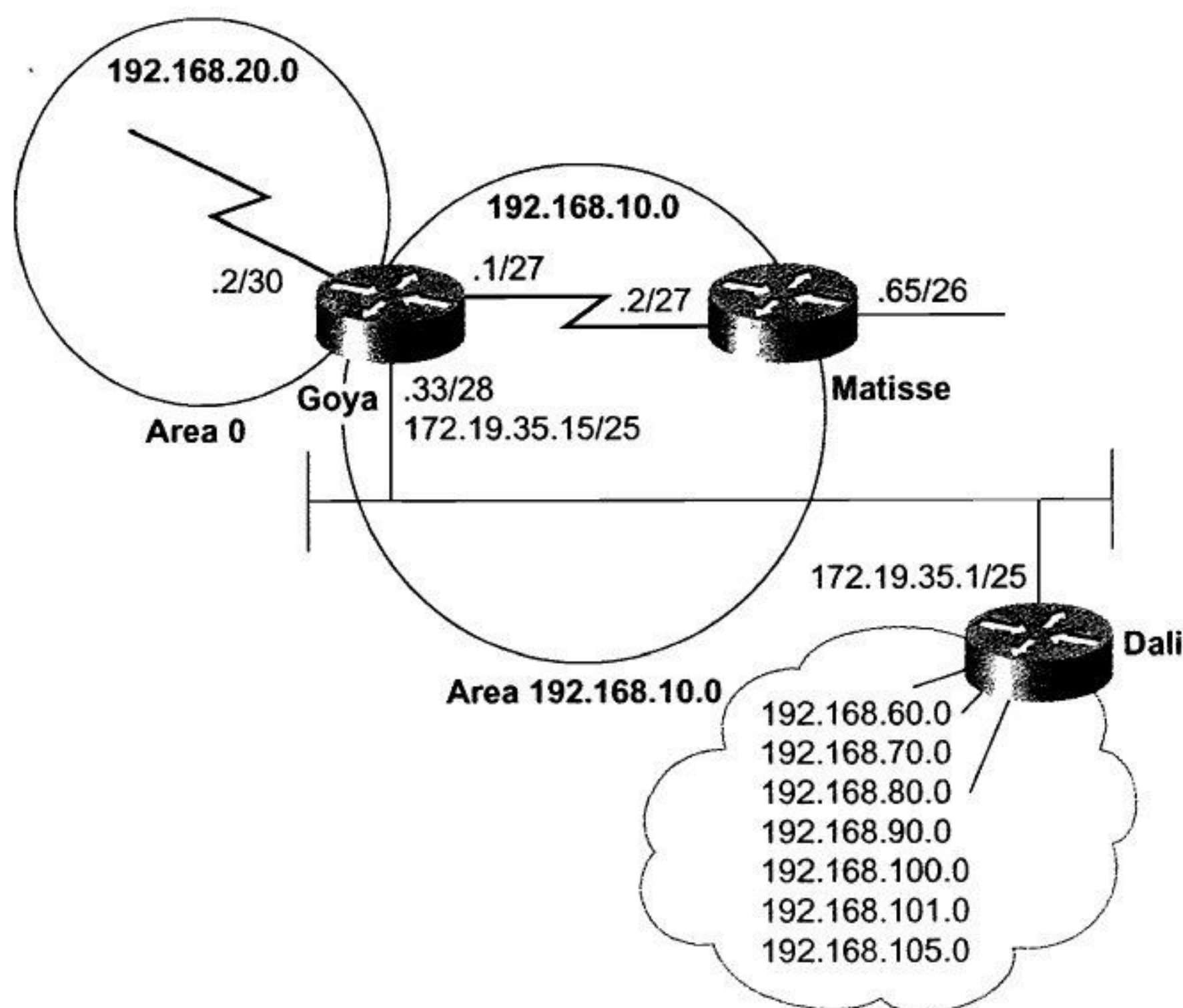


图 9-82 路由器 Dali 的链路移到了路由器 Goya 上，现在变成路由器 Goya 来宣告 RIP 到
路由器 Dali，并且重新分配学到的路由到 OSPF


```

Matisse#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 192.168.10.1 to network 0.0.0.0

O N1 192.168.105.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
O N1 192.168.100.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
O N1 192.168.101.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
O N1 192.168.70.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
O N1 192.168.90.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
O N1 192.168.80.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
O N1 192.168.60.0/24 [110/74] via 192.168.10.1, 00:03:03, Serial1
    192.168.50.0/32 is subnetted, 1 subnets
C      192.168.50.4 is directly connected, Loopback0
    192.168.10.0/24 is variably subnetted, 3 subnets, 3 masks
C      192.168.10.64/26 is directly connected, Loopback1
C      192.168.10.32/28 is directly connected, Ethernet0
C      192.168.10.0/27 is directly connected, Serial1
O*IA 0.0.0.0/0 [110/65] via 192.168.10.1, 00:03:04, Serial1
Matisse#

```

图 9-83 一台 ABR 路由器同时也是一台 ASBR 路由器时，将会利用类型 7 的 LSA 通告外部路由到 NSSA 区域。

在这个例子中，路由器 Goya 正在使用 N1 的度量类型通告外部路由

```

interface Ethernet0
 ip address 172.19.35.15 255.255.255.128
!
router ospf 30
 redistribute rip metric 10 metric-type 1
 network 192.168.20.0 0.0.0.3 area 0
 network 192.168.10.0 0.0.0.31 area 192.168.10.0
 area 192.168.10.0 nssa no-redistribution no-summary
!
router rip
 network 172.19.0.0

```

在这里，**area nssa** 命令阻塞了类型 5 的 LSA 通过路由器 Goya 进入到该区域，而 **no-redistribution** 参数阻塞了类型 7 的 LSA，还有 **no-summary** 参数阻塞了类型 3 和 4 的 LSA。和前面一样，**no-summary** 参数也使路由器 Goya 发送一条类型 3 的 LSA 来向该区域通告一个缺省路由。如图 9-84 所示，显示了在路由器 Goya 禁止了类型 7 的路由重新分配之后路由器 Matisse 的路由选择表。注意，即使外部网络不在这个路由选择表中，它们依然是可达的，这是因为路由选择表中含有缺省路由的缘故。

在最后的这个例子中，假设需要路由器 Goya 允许类型 3 和类型 4 的 LSA 泛洪到 NSSA 区域，但是不允许类型 5 和类型 7 的 LSA 泛洪到该区域。这里的问题是当在路由器上去除了 **no-summary** 参数后，ABR 路由器将不再始发一条类型 3 的 LSA 通告缺省路由了。没有缺省路由，外部网络也就无法从 NSSA 区域内部到达了。要解决这个问题，可以在命令 **area nssa** 后面增加一个 **default-information-originate** 参数，这就可以使 ABR 路由器来通告一条缺省

路由到这个 NSSA 区域了。这时，它是使用类型 7 的 LSA 来通告这条缺省路由的。要使用这个参数，路由器 Goya 上的 OSPF 配置如下：

```
router ospf 30
 redistribute rip metric 10 metric-type 1
 network 192.168.20.0 0.0.0.3 area 0
 network 192.168.10.0 0.0.0.31 area 192.168.10.0
 area 192.168.10.0 nssa no-redistribution default-information-originate
```

```
Matisse#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 192.168.10.1 to network 0.0.0.0

192.168.50.0/32 is subnetted, 1 subnets
C      192.168.50.4 is directly connected, Loopback0
192.168.10.0/24 is variably subnetted, 3 subnets, 3 masks
C      192.168.10.64/26 is directly connected, Loopback1
C      192.168.10.32/28 is directly connected, Ethernet0
C      192.168.10.0/27 is directly connected, Serial1
O*IA 0.0.0.0/0 [110/65] via 192.168.10.1, 00:00:10, Serial1
Matisse#
```

图 9-84 在路由器 Goya 的 `area nssa` 命令中增加了 `no-redistribution` 参数后，

图 9-83 中的路由选择表将不再包括从类型 7 的 LSA 中学到的路由

```
Matisse#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 192.168.10.1 to network 0.0.0.0

192.168.50.0/32 is subnetted, 1 subnets
C      192.168.50.4 is directly connected, Loopback0
192.168.10.0/24 is variably subnetted, 3 subnets, 3 masks
C      192.168.10.64/26 is directly connected, Loopback1
C      192.168.10.32/28 is directly connected, Ethernet0
C      192.168.10.0/27 is directly connected, Serial1
192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
O IA   192.168.30.1/32 [110/193] via 192.168.10.1, 00:00:13, Serial1
O IA   192.168.30.8/29 [110/192] via 192.168.10.1, 00:00:13, Serial1
192.168.20.0/30 is subnetted, 1 subnets
O IA   192.168.20.0 [110/128] via 192.168.10.1, 00:00:14, Serial1
O*N2 0.0.0.0/0 [110/1] via 192.168.10.1, 00:00:14, Serial1
Matisse#
```

图 9-85 在命令 `area nssa` 中增加 `default-information-originate` 参数后，可以使 ABR 路由器通告一条缺省路由到这个 NSSA 区域

如图 9-85 所示，显示了做过重新配置后的路由器 Matisse 的路由选择表。在这里，路由

选择表中包括了区域间路由和一条缺省路由, 这条缺省路由带有 N2 标志, 表明这条路由是从类型 7 的 LSA 学习到的。

9.2.8 案例研究 8: 地址汇总

虽然末梢区域可以通过防止某些 LSA 进入该区域, 从而达到在一个非骨干的区域里节省资源的目的, 但是从骨干区域上来看, 这些区域除了节省资源外并没有做其他任何事情。一个区域内所有的地址仍然会通告到骨干区域当中。这种情形可以通过地址汇总来帮助解决。像末梢区域一样, 地址汇总也是通过减少泛洪的 LSA 数量来达到节省资源的目的。另外, 它还可以通过屏蔽一些网络不稳定的细节来节省资源。例如, 一个忽好忽坏的不稳定的子网在它每一次状态发生转变的时候, 都会引起 LSA 在整个互连网络中进行泛洪。但是, 如果这个子网地址被汇总包含在一个汇总地址中的话, 那么单独的子网和它的稳定性就不再被通告出去了。

在 Cisco 的路由器上可以执行两种类型的地址汇总: 区域间路由汇总和外部路由汇总。区域间路由汇总 (Inter-area summarization), 顾名思义, 是指在区域之间的地址汇总。这种类型的汇总通常是配置在 ABR 路由器上的。外部路由汇总 (External route summarization) 允许一组外部地址汇总为一条汇总地址通过重新分配注入到一个 OSPF 域中, 这种类型的汇总通常是配置在 ASBR 路由器上的。区域间路由汇总将在本小节介绍, 而外部路由汇总将在第 11 章讲述。

在图 9-86 中, 区域 15 包含了 8 个子网: 10.0.0.0/16~10.7.0.0/16。图 9-87 显示了这些子网地址可以使用单个汇总地址 10.0.0.0/13 来表示。

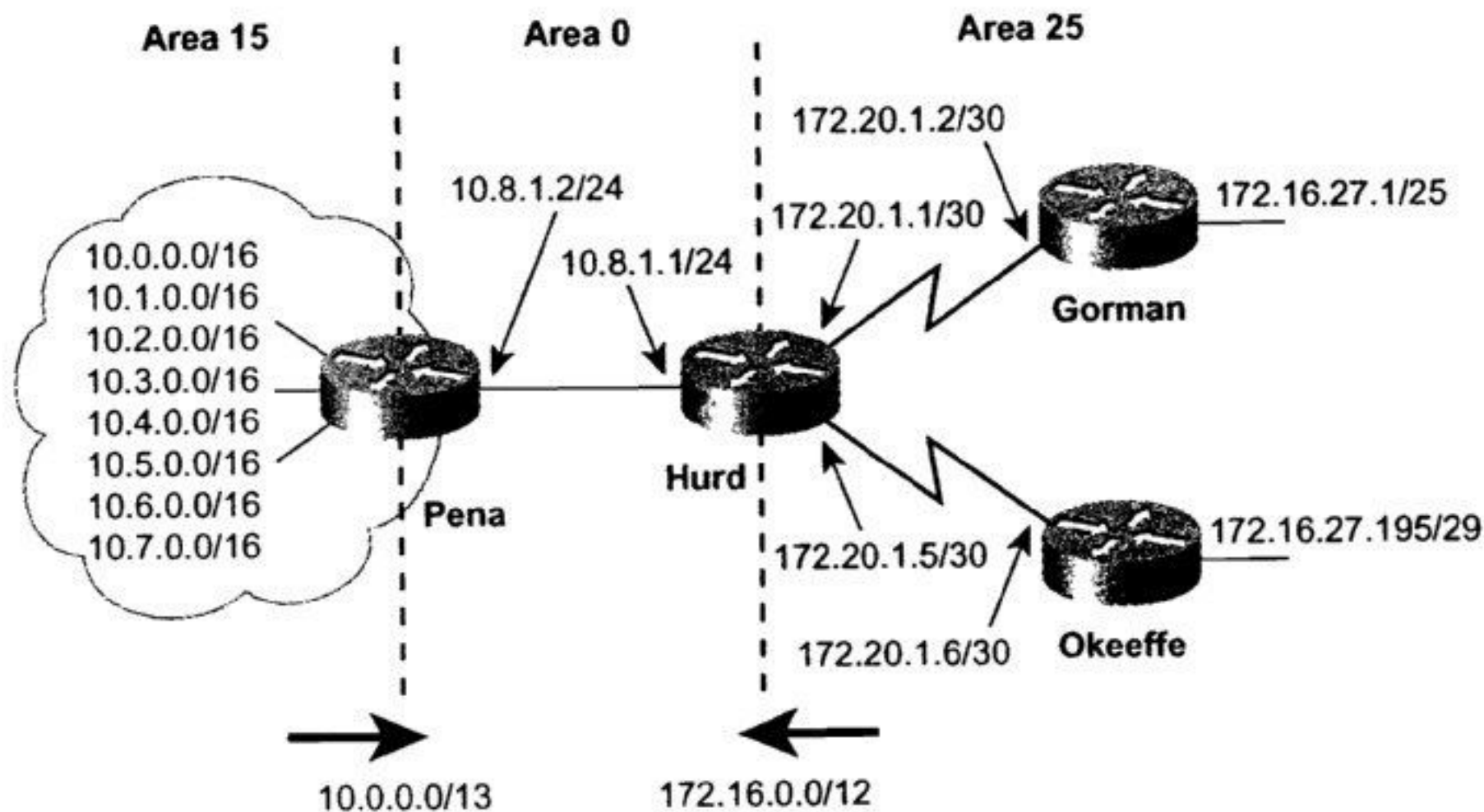


图 9-86 在区域 15 和 25 中的地址能够汇总到骨干区域里

11111111111111110000000000000000	= 16-bit mask
00001010000000000000000000000000	= 10.0.0.0/16
00001010000000010000000000000000	= 10.1.0.0/16
00001010000000100000000000000000	= 10.2.0.0/16
00001010000000110000000000000000	= 10.3.0.0/16
00001010000001000000000000000000	= 10.4.0.0/16
00001010000001010000000000000000	= 10.5.0.0/16
00001010000001100000000000000000	= 10.6.0.0/16
00001010000001110000000000000000	= 10.7.0.0/16
00001010000000000000000000000000	= 10.0.0.0/13

图 9-87 可以使用汇总地址 10.0.0.0/13 来表示地址 10.0.0.0/16~10.7.0.0/16 的地址范围

在一台 ABR 路由器上配置一个汇总地址，既可以通告到骨干区域，也可以通告到一个非骨干的区域。最好的方法是，一个非骨干区域的地址应该通过它自己的 ABR 路由器汇总到骨干区域。而与之相对的是使其他所有的 ABR 路由器汇总这个区域到它们各自的区域内。然后，在骨干区域中，被汇总的地址将穿越骨干区域并通告到其他区域中去。这两种方法都简化了路由器的配置并减小了骨干区域内链路状态数据库的大小。

area range 命令指定了汇总地址所属的区域、汇总地址和地址掩码。回忆第 8 章“增强型内部网关路由选择协议 (EIGRP)”，在那里，当为 EIGRP 协议配置一条汇总路由时，会有一条指向空接口 (NULL Interface) 的路由被自动地加入路由选择表中，以便用来避免路由黑洞和路由环路。¹和 EIGRP 协议不同，OSPF 不会自动地加入这条路由。因此，无论何时，读者在一个 OSPF 域内配置汇总路由时，就应该确认为这条汇总地址增加了一条静态路由指向空接口。

路由器 Pena 的 OSPF 配置如下：

```
router ospf 1
 network 10.0.0.0 0.7.255.255 area 15
 network 10.8.0.0 0.7.255.255 area 0
 area 15 range 10.0.0.0 255.248.0.0
!
ip route 10.0.0.0 255.248.0.0 Null0
```

图 9-87 中显示出 10.0.0.0/13 表示的地址范围是连续的，也就是说，被汇总的 3 个二进制位构成了每一个 000~111 的组合。而在区域 25 中的地址不同，它们不能形成一个连续的地址范围。但是，它们仍然可以通过如下的配置在路由器 Hurd 上进行汇总：

```
router ospf 1
 network 10.8.0.0 0.0.255.255 area 0
 network 172.20.0.0 0.0.255.255 area 25
 area 25 range 172.16.0.0 255.240.0.0
!
ip route 172.16.0.0 255.240.0.0 Null0
```

即使在这个汇总地址范围内的地址出现在这个互联网络以外的地址，该汇总路由也是可以正常工作的。在图 9-88 中，网络 172.17.0.0/16 在区域 15 内，尽管这个地址是属于来自区域 25 汇总的那一段地址的。路由器 Pena 将通告这个地址到骨干区域，而路由器 Hurd 将学到它并通告到区域 25 中去。跟随这个地址的网络掩码比汇总地址 172.16.0.0/12 的掩码更具体（也就是说，是更长的），又因为 OSPF 协议是无类别的路由选择协议，因此，OSPF 将能够转发属于网络 172.17.0.0/16 的目的地址到正确的目的地。

尽管图 9-88 中的地址配置方法可以工作，但是这并不是一个值得推荐的设计方法。汇总地址的地方是节省了资源，但对网络 172.17.0.0/16 的通告还是必须独立于网络 172.16.0.0/12，并穿过骨干区域。在像这样的网络设计中，如果使用了缺省路由，还可能会有产生路由环路的隐患。这个问题会在第 12 章“缺省路由和按需路由选择”进行详细讲述。

这里也要注意图 9-88 中，子网 172.16.27.0/25（和路由器 Gorman 相连）和子网 172.16.27.192/29（和路由器 Okeeffe 相连）是不连续的。又因为 OSPF 协议是一个无类别路

¹ 增加这条路由的理由将在第 11 章“路由重新分配”和第 12 章“缺省路由和按需路由选择”中结合实例，作更详细的讲述。

由选择协议, 因而路由器 Gorman 和 Okeeffe 不会扮演网络边界路由器的角色。这些子网和它们的掩码将会通告到网络 172.20.0.0 中去, 并且不会有路由选择不确定的情况发生。

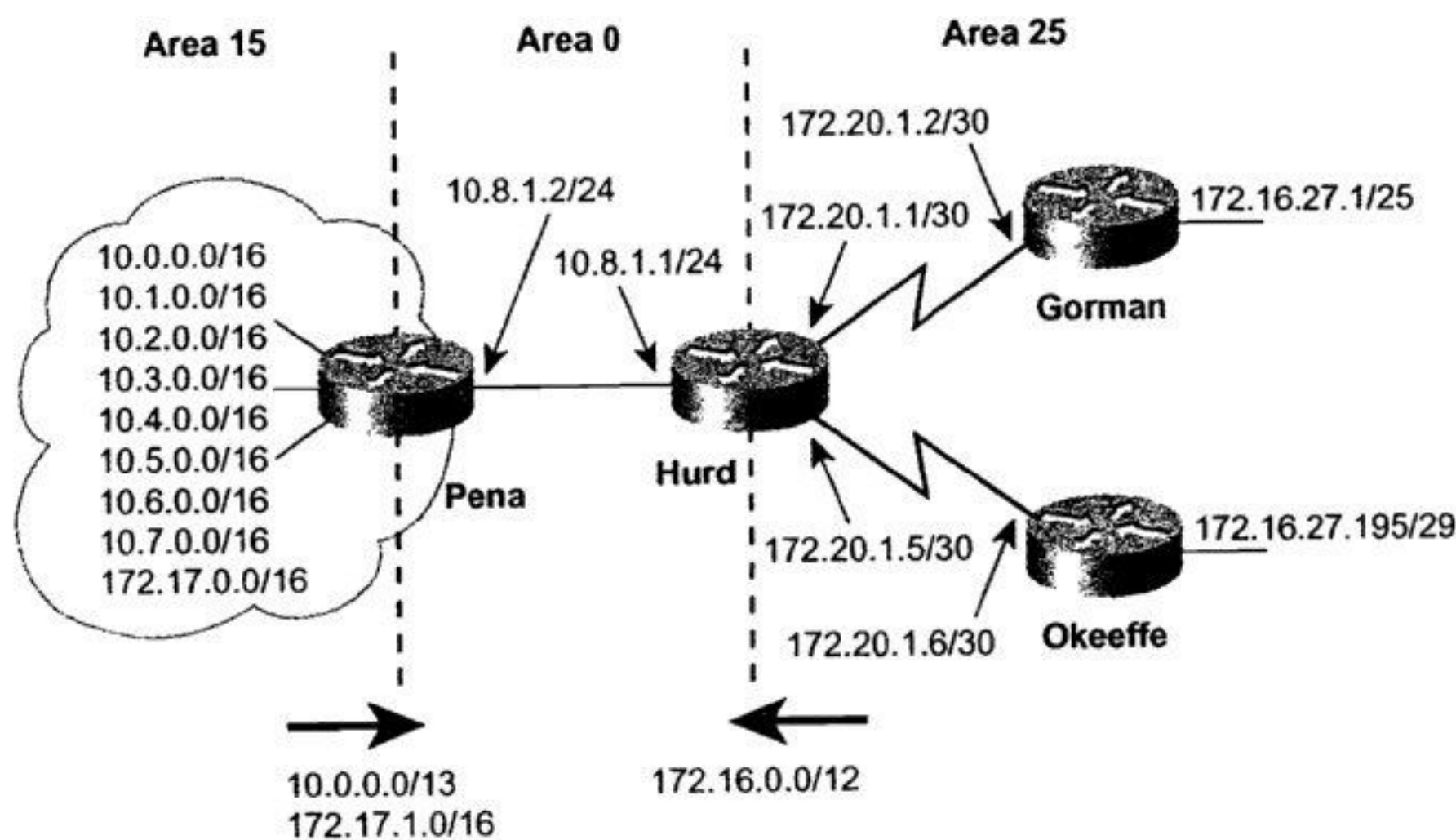


图 9-88 网络 172.17.0.0 在区域 15 内, 尽管这个地址是属于来自区域 25 汇总的那一段地址 172.16.0.0/12 的

9.2.9 案例研究 9: 认证

OSPF 报文可以通过认证来防止有意或无意地引入有害路由信息的情况发生。表 9-8 中列出了有效的认证类型。Null 认证 (类型 0) 是路由器缺省使用的类型, 表示在报文头部没有包含认证信息, 也就是说, 不需要认证。在路由器上可以使用简单的明文口令 (类型 1) 或者 MD5 加密校验和 (类型 2) 来配置认证。如果一个区域内某处配置了认证, 那么就必须在整个区域内都配置认证。

如果网络管理者是以增加网络的安全性为目标的, 那么只有在 OSPF 区域内的设备不能支持更安全的类型 2 的认证时才考虑使用类型 1 的认证。明文认证会在互联网络上给网络攻击者留下一个安全漏洞, 因为网络上传送的报文能够被协议分析仪捕获, 并读出所设置的口令 (请参考第 7 章, 并请注意图 7-8)。但是, 类型 1 的认证在进行 OSPF 的重新配置时会变得比较有用。例如, 不同的口令可以用在作重新配置时“旧”OSPF 路由器和“新”OSPF 路由器上, 从而避免它们在共享一个公共广播网络的情况下相互通信。

在一个区域上配置类型 1 的认证方式, 可以使用命令 **ip ospf authentication-key** 来为和该区域相连的每一个接口分配一个口令, 这个口令最长为 8 个 8bit 字节长。所分配的口令不必要在整个区域上都相同, 但是在—对邻居路由器之间必须相同。然后, 在 OSPF 协议的配置下添加 **area authentication** 命令来使类型 1 的认证方式有效。

参考图 9-88, 在区域 0 和区域 25 上启用类型 1 的认证。路由器 Hurd 的有关配置如下:

```
interface Ethernet0
  ip address 10.8.1.1 255.255.255.0
  ip ospf authentication santafe
!
interface Serial0
  ip address 172.20.1.1 255.255.255.252
```



```
ip ospf authentication taos
!
interface Serial1
ip address 172.20.1.5 255.255.255.252
ip ospf authentication abiquiu
!
router ospf 1
network 10.8.0.0 0.0.255.255 area 0
network 172.20.0.0 0.0.255.255 area 25
area 25 range 172.16.0.0 255.240.0.0
area 0 authentication
area 25 authentication
```

在这里，口令“santafe”使用在路由器 Hurd 和 Pena 之间；口令“taos”使用在路由器 Hurd 和 Gorman 之间；而口令“abiquiu”使用在路由器 Hurd 和 Okeeffe 之间。

MD5 算法使用在类型 2 的认证方式中。它用来为 OSPF 报文内容和一个口令（或密钥）计算一个散列值（Hash Value）。这个散列值将和一个密钥 ID，以及一个不变小的序列号一起在报文中传送。拥有相同口令的接收路由器将会计算它自己的散列值。如果传送的消息中什么内容都没改变，那么接收路由器的散列值应该和发送路由器在消息报文中传送的散列值相匹配。密钥 ID 允许路由器指定多个口令，这样可以使口令的改变比较容易，并具有更好的安全性。在这个案例研究中包含了一个口令迁移的例子。序列号用来防止“重现攻击（replay attacks）”，以避免 OSPF 报文被捕获、更改和重新传送给一台路由器。

在一个区域上配置类型 2 的认证方式，可以使用命令 **ip ospf message-digest-key md5** 来为和该区域相连的每一个接口分配一个口令和一个密钥 ID（Key ID），这里的口令最长为 16 个 8bit 字节，而密钥 ID 在 1~255 之间。和类型 1 的认证方式相同，所分配的口令不必要在整个区域上都相同，但是在一对邻居路由器之间的密钥 ID 和口令都必须相同。然后，在 OSPF 协议的配置下添加 **area authentication message-digest** 命令来使类型 2 的认证方式有效。

在路由器 Hurd 上启用类型 2 的认证方式，有关配置如下：

```
interface Ethernet0
ip address 10.8.1.1 255.255.255.0
ip ospf message-digest-key 5 md5 santafe
!
interface Serial0
ip address 172.20.1.1 255.255.255.252
ip ospf message-digest-key 10 md5 taos
!
interface Serial1
ip address 172.20.1.5 255.255.255.252
ip ospf message-digest-key 15 md5 abiquiu
!
router ospf 1
network 10.8.0.0 0.0.255.255 area 0
network 172.20.0.0 0.0.255.255 area 25
area 25 range 172.16.0.0 255.240.0.0
area 0 authentication message-digest
area 25 authentication message-digest
```


密钥允许路由器在不需要使认证无效的情况下更改它的口令。例如,为了在路由器 Hurd 和 Okeeffe 之间更改口令,新的口令可以和一个不同的密钥一起配置。这时,路由器 Hurd 的配置应该是:

```
interface Serial1
  ip address 172.20.1.5 255.255.255.252
  ip ospf message-digest-key 15 md5 abiquiu
  ip ospf message-digest-key 20 md5 steiglitz
```

路由器 Hurd 现在将会在 S1 接口为每一个 OSPF 报文发送两个重复的拷贝,一个使用密钥 15 认证,另一个使用密钥 20 认证。当路由器 Hurd 开始从路由器 Okeeffe 那里收到使用密钥 20 认证的 OSPF 报文时,它就会停止发送密钥为 15 的认证报文。一旦新的密钥可以使用了,原来的密钥就可以使用下面的命令从这两台路由器上移掉:

```
no ip ospf message-digest-key 15 md5 abiquiu
```

一个在运行的互连网络中的口令从来不应该向这些例子中的口令那样可以预先得知。在所有使用认证的路由器的配置文件里增加命令 **service password-encryption** 是比较明智的。这样做可以使路由器对配置文件中任何显示的口令进行加密,因此可以保护口令不被别人通过简单地查看路由器配置的文本拷贝就可以得知的隐患。如果已经给路由器 Hurd 的配置口令加密,那么,它的配置应该像下面这样的显示:

```
service password-encryption
!
interface Ethernet0
  ip address 10.8.1.1 255.255.255.0
  ip ospf message-digest-key 5 md5 7 001712008105A0D03
!
interface Serial0
  ip address 172.20.1.1 255.255.255.252
  ip ospf message-digest-key 10 md5 7 03105A0415
!
interface Serial1
  ip address 172.20.1.5 255.255.255.252
  ip ospf message-digest-key 20 md5 7 070E23455F1C1010
```

9.2.10 案例研究 10: 虚链路

如图 9-89 所示,显示了一个骨干区域设计的比较糟糕的互连网络。如果路由器 Hokusai 和 Hiroshige 之间的链路失效了,那么这个网络的骨干区域将被分割成两部分。结果,路由器 Sesshiu 和 Okyo 不能相互通信。即使这两台路由器是分离的区域的 ABR,区域间的通信量也将会在这些区域之间被阻塞。

在这个实例中,最有效的解决办法是在路由器 Sesshiu 和 Okyo 之间为骨干区域增加另外一条链路。在这个骨干区域得到改进之前,作为一种过渡方案,可以在路由器 Hokusai 和 Hiroshige 之间建立一条穿过区域 100 的虚链路。

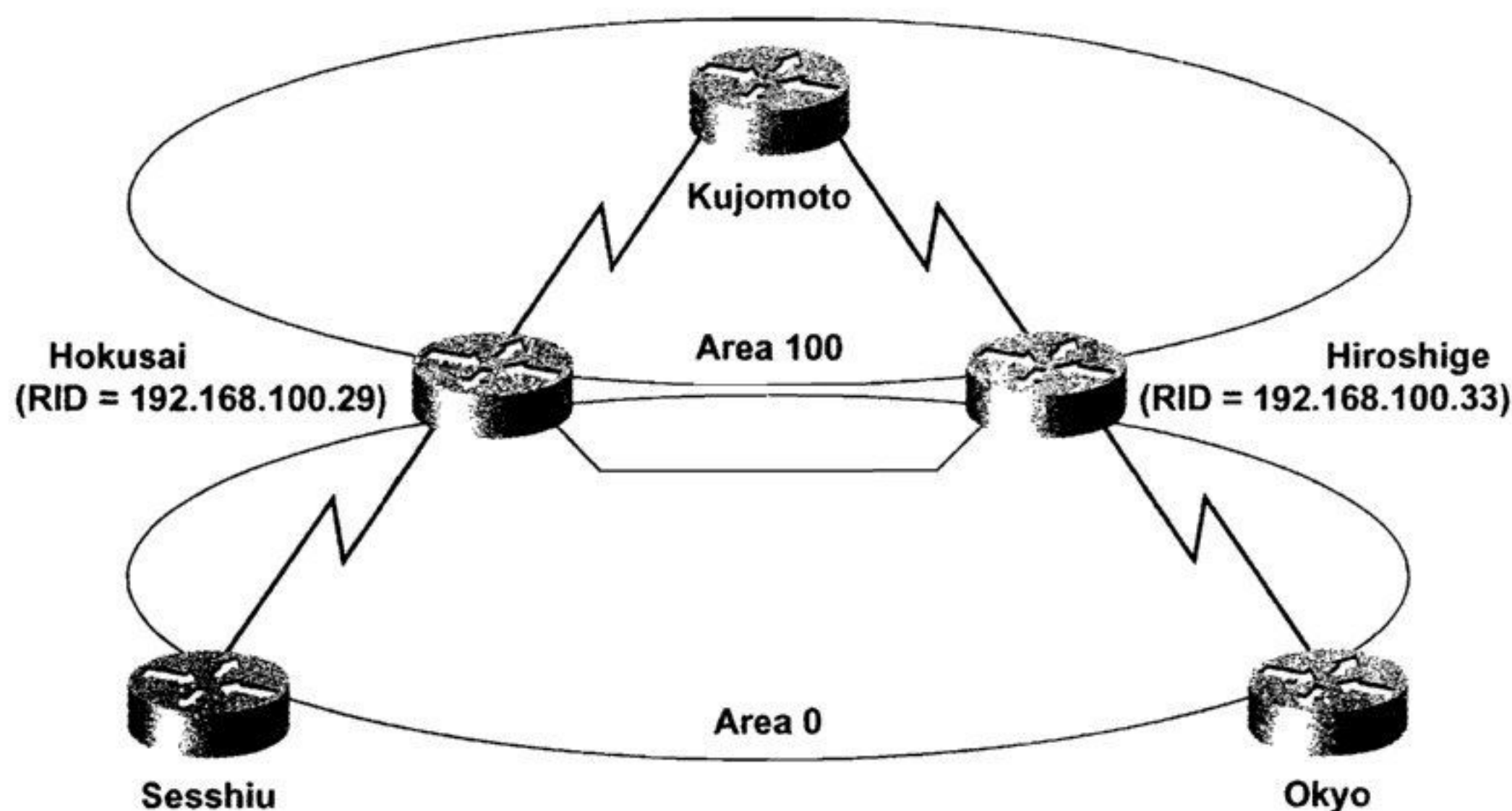


图 9-89 路由器 Hokusai 和 Hiroshige 之间的链路失效了将会使它们的骨干区域变成分段区域

虚链路总是建立在 ABR 路由器之间的，至少它们之中有一个 ABR 路由器是必须和区域 0 相连的。¹在每一台 ABR 路由器的 OSPF 配置里，通过添加 **area virtual-link** 命令来配置一条虚链路，并指定了这条虚链路要穿过的区域和这条链路远端的 ABR 的路由器 ID。在路由器 Hokusai 和 Hiroshige 之间建立一条虚链路的配置如下：

路由器 Hokusai:

```
router ospf 10
 network 192.168.100.1 0.0.0.0 area 0
 network 192.168.100.29 0.0.0.0 area 0
 network 192.168.100.21 0.0.0.0 area 100
 area 100 virtual-link 192.168.100.33
```

路由器 Hiroshige:

```
router ospf 10
 network 192.168.100.2 0.0.0.0 area 0
 network 192.168.100.33 0.0.0.0 area 0
 network 192.168.100.25 0.0.0.0 area 100
 area 100 virtual-link 192.168.100.29
```

```
Hokusai#show ip ospf virtual-link
Virtual Link OSPF_VL1 to router 192.168.100.33 is up
Run as demand circuit
DoNotAge LSA not allowed (Number of DCbitless LSA is 2).
Transit area 100, via interface Serial0, Cost of using 128
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:00
Adjacency State FULL (Hello suppressed)
Hokusai#
```

图 9-90 使用命令 **show ip ospf virtual-link** 可以查看一条虚链路的状态

¹ 当一条虚链路穿过一个非骨干的区域连接某个区域到骨干区域时，其中一个 ABR 路由器将位于这两个非骨干的区域之间。

完成以上的配置后,在正常情况下,路由器 Sesshiu 和 Okyo 之间的数据包访问可以通过路由器 Hokusai 和 Hiroshige 之间的骨干区域上的链路进行转发。但是,如果那条链路失效的话,将会利用虚链路作数据包的转发。如图 9-90 所示,虽然每一台路由器都把这条链路看作是一条无编号的点到点网络,但实际上数据包的转发是通过路由器 Kujomoto 的。

9.2.11 案例研究 11: 运行在 NBMA 网络上的 OSPF

在一些非广播多址网络上,例如 X.25、帧中继和 ATM 等,运行 OSPF 协议会产生一个问题。“多址”意味着一个 NBMA 的网络“云”是多个设备共同相连的单个网络,和以太网或者令牌环网络一样(如图 9-91)。但是它又和以太网、令牌环网等广播型网络不同,它是非广播的。“非广播”意味着发送到这个网络上的报文不一定会被和该网络相连的其他所有的路由器看到。这样,由于 NBMA 网络是多址的,OSPF 协议将需要选取一个 DR 路由器和 BDR 路由器。但是由于 NBMA 网络又是非广播的,它不能保证所有相连的路由器都能收到其他所有路由器发送的 Hello 报文。因此,所有的路由器不一定能够自动地了解它的所有邻居,因而不一定能够正确地进行 DR 的选取。

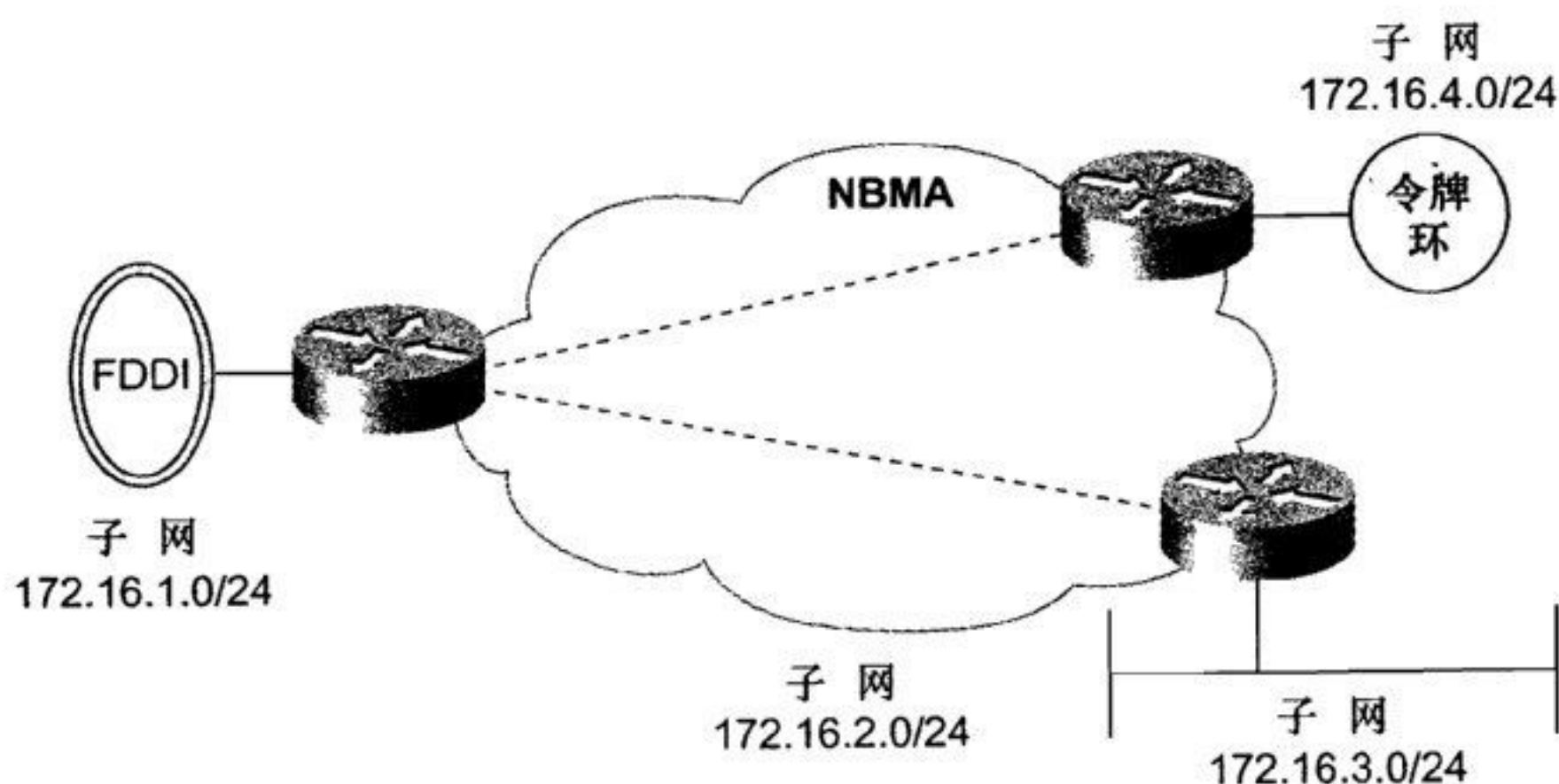


图 9-91 路由选择协议会把 NBMA 网络看作是有多个设备相连的一个子网。但是当一个 NBMA 网络是部分网状连接时,正如这里所示的,不是所有相连的路由器都和其他所有的路由器有直接连接的

本节内容将阐述几个解决 NBMA 网络问题的方案。具体方案的选择依赖于实现该解决方案的互联网络的特征。

最早的解决方案出现在 Cisco IOS 10.0 版之前的有关版本里面。它是使用 **neighbor** 命令以手工的方式指定每一台路由器的邻居并创建 DR 的。如图 9-92 所显示的一个和 4 台路由器相连的帧中继网络。

在图 9-92 中,由于 PVC 电路的配置是部分网状连接的 hub-and-spoke 的星型结构,因此,路由器 Rembrandt 必须成为一个 DR 路由器。作为中心的“hub”,它是惟一的和其他所有的路由器直接相连的路由器。这 4 台路由器的配置如下:

路由器 Rembrandt:

```
interface Serial0
  encapsulation frame-relay
```



```
ip address 172.16.2.1 255.255.255.0
frame-relay map ip 172.16.2.2 100
frame-relay map ip 172.16.2.3 300
frame-relay map ip 172.16.2.4 500
!
router ospf 1
network 172.16.0.0 0.0.255.255 area 0
neighbor 172.16.2.2
neighbor 172.16.2.3
neighbor 172.16.2.4
```

路由器 Hals:

```
interface Serial0
encapsulation frame-relay
ip address 172.16.2.2 255.255.255.0
frame-relay map ip 172.16.2.1 600
frame-relay map ip 172.16.2.3 600
frame-relay map ip 172.16.2.4 600
!
router ospf 1
network 172.16.0.0 0.0.255.255 area 0
neighbor 172.16.2.1 priority 10
```

路由器 Vandyck:

```
interface Serial0
encapsulation frame-relay
ip address 172.16.2.3 255.255.255.0
frame-relay map ip 172.16.2.1 400
frame-relay map ip 172.16.2.2 400
frame-relay map ip 172.16.2.4 400
!
router ospf 1
network 172.16.0.0 0.0.255.255 area 0
neighbor 172.16.2.1 priority 10
```

路由器 Brueghel:

```
interface Serial0
encapsulation frame-relay
ip address 172.16.2.4 255.255.255.0
frame-relay map ip 172.16.2.1 200
frame-relay map ip 172.16.2.2 200
frame-relay map ip 172.16.2.3 200
!
router ospf 1
network 172.16.0.0 0.0.255.255 area 0
neighbor 172.16.2.1 priority 10
```

在路由器 Rembrandt 上, 使用 **neighbor** 命令配置了它的 3 个邻居的接口 IP 地址。缺省的优先级是 0, 在路由器 Rembrandt 上不改变这个缺省值, 没有邻居有资格成为一个 DR 或者 BDR。

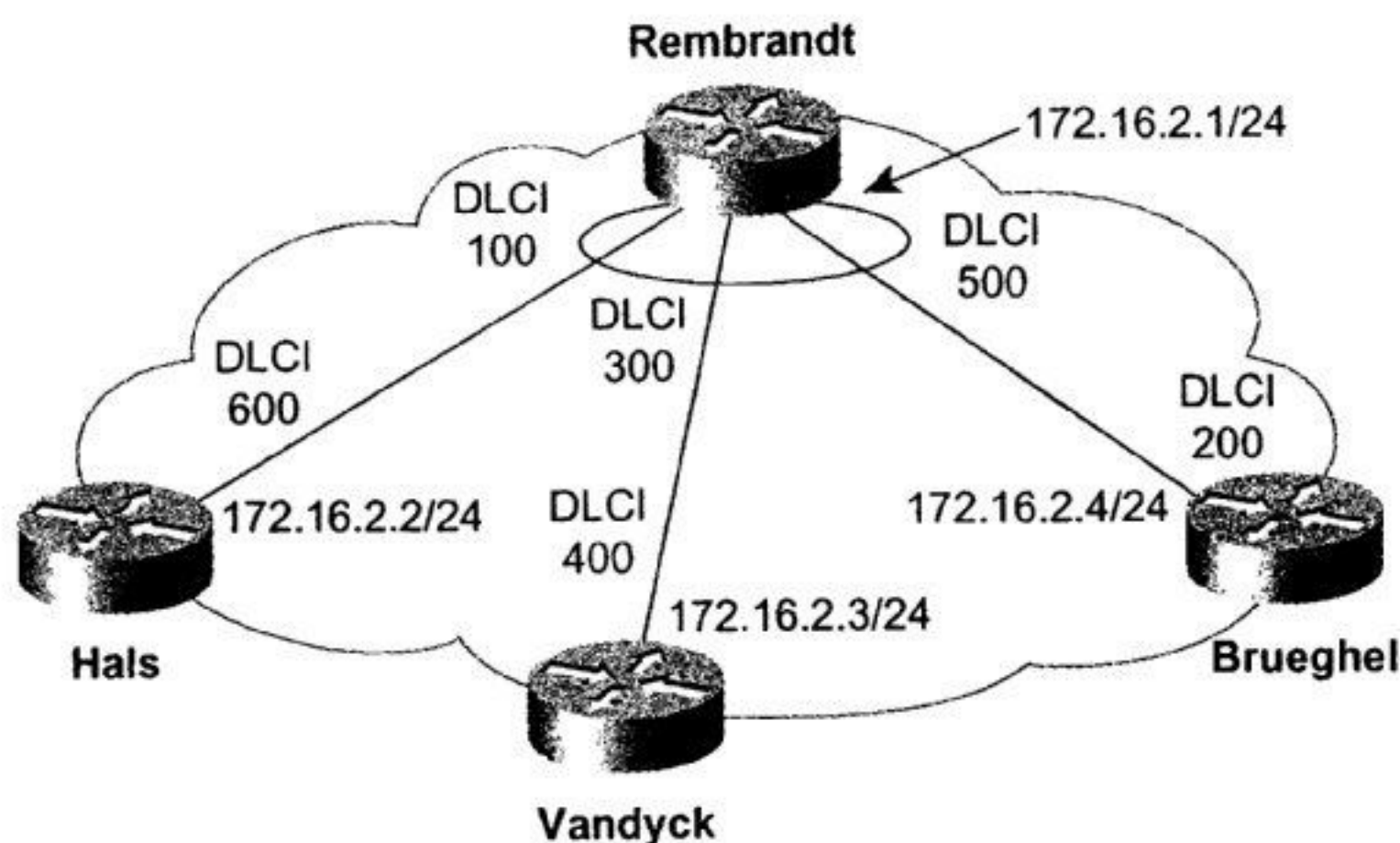


图 9-92 在这个 NBMA 网络中配置 OSPF 存在的几种选择

另外 3 台路由器仅把路由器 Rembrandt 作为它们各自的邻居来配置，并把优先级设置为 10，这意味着路由器 Rembrandt 将成为 DR 路由器。通过使路由器 Rembrandt 成为 DR 路由器，这些 PVC 电路可以精确地仿效假设这 4 台路由器连接到一个广播型多址网络上时所形成的邻接关系。现在，OSPF 报文将以单播的方式转发到所配置的邻居地址上。

再次重申一下，**neighbor** 命令只是在老的 IOS 版本（10.0 版本以前）上才是必要的。而新的解决方案中，使用 **ip ospf network** 命令可以改变缺省的 OSPF 网络类型。这条命令的一个选项是改变网络类型为广播型，这可以在每一个帧中继接口上使用 **ip ospf network broadcast** 来实现。这个网络类型的更改将会让 OSPF 把这个 NBMA 视为一个广播型网络，在这种方案里，4 台路由器的配置如下：

路由器 Rembrandt:

```
interface Serial0
  encapsulation frame-relay
  ip address 172.16.2.1 255.255.255.0
  ip ospf network broadcast
  ip ospf priority 10
  frame-relay map ip 172.16.2.2 100 broadcast
  frame-relay map ip 172.16.2.3 300 broadcast
  frame-relay map ip 172.16.2.4 500 broadcast
!
router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
```

路由器 Hals:

```
interface Serial0
  encapsulation frame-relay
  ip address 172.16.2.2 255.255.255.0
  ip ospf network broadcast
  ip ospf priority 0
  frame-relay map ip 172.16.2.1 600 broadcast
  frame-relay map ip 172.16.2.3 600 broadcast
  frame-relay map ip 172.16.2.4 600 broadcast
```



```
!  
router ospf 1  
  network 172.16.0.0 0.0.255.255 area 0
```

路由器 Vandyck:

```
interface Serial0  
  encapsulation frame-relay  
  ip address 172.16.2.3 255.255.255.0  
  ip ospf network broadcast  
  ip ospf priority 0  
  frame-relay map ip 172.16.2.1 400 broadcast  
  frame-relay map ip 172.16.2.2 400 broadcast  
  frame-relay map ip 172.16.2.4 400 broadcast  
!  
router ospf 1  
  network 172.16.0.0 0.0.255.255 area 0
```

路由器 Brueghel:

```
interface Serial0  
  encapsulation frame-relay  
  ip address 172.16.2.4 255.255.255.0  
  ip ospf network broadcast  
  ip ospf priority 0  
  frame-relay map ip 172.16.2.1 200 broadcast  
  frame-relay map ip 172.16.2.2 200 broadcast  
  frame-relay map ip 172.16.2.3 200 broadcast  
!  
router ospf 1  
  network 172.16.0.0 0.0.255.255 area 0
```

注意: 在这个例子中, 路由器 Rembrandt 的接口的优先级设置为 10, 而其他接口的优先级设置为 0。这将可以再次确保路由器 Rembrandt 能够成为一个 DR 路由器。注意, 静态的帧中继映射命令也设置为转发到广播和组播地址上了。

影响 DR 选取的另一种方案是实现一个全网状连接的拓扑结构, 即每一台路由器都有 PVC 电路和其他所有的路由器相连。站在路由器本身的角度来看, 这种方案实际上是所有 NBMA 网络实现中最有效的方案。但是这种方法的一个显而易见的缺点就是金钱的花费比较高昂。假设有 n 台路由器, 那么为了创建一个全网状连接的拓扑结构将必须要有 $n(n-1)/2$ 条 PVC 电路才能实现。例如, 图 9-92 中的 4 台路由器要创建一个全网状连接的拓扑应该需要 6 条 PVC 电路, 而 16 台路由器则需要 120 条 PVC 电路。

另外一种方法, 就是通过改变网络类型为点到多点网络, 这样就可以避免 DR/BDR 选取的处理。点到多点网络把 PVC 当作一个点到点链路的集合, 因此就没有 DR/BDR 的选取发生。在有多个厂家设备的网络环境里, 点到多点类型可能是广播型网络之外惟一的一种选择。

在下面的配置中, 把和每一个接口相关的 OSPF 网络类型改变成了点到多点类型:

路由器 Rembrandt:


```
interface Serial0
  encapsulation frame-relay
  ip address 172.16.2.1 255.255.255.0
  ip ospf network point-to-multipoint
!
router ospf 1
  network 172.16.0.0 0.0.255.255 area
```

路由器 Hals:

```
interface Serial0
  encapsulation frame-relay
  ip address 172.16.2.2 255.255.255.0
  ip ospf network point-to-multipoint
!
router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
```

路由器 Vandyck:

```
interface Serial0
  encapsulation frame-relay
  ip address 172.16.2.3 255.255.255.0
  ip ospf network point-to-multipoint
!
router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
```

路由器 Brueghel:

```
interface Serial0
  encapsulation frame-relay
  ip address 172.16.2.4 255.255.255.0
  ip ospf network point-to-multipoint
!
router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
```

在这些配置中,利用帧中继网络逆向 ARP 解析功能动态地把网络层的地址映射到 DLCI 上,这种方法替代了上面例子中使用的静态映射命令。当然,如果想用,静态映射依然可以使用。

OSPF 点到多点的网络类型把下层的网络看作一组点到点链路的集合,而不是一个多址网络,并且 OSPF 报文以组播的方式发送到它的邻居。这种解决方案在那些动态连接的网络上会产生问题,像帧中继 SVC 或者 ATM SVC 等。自 IOS 11.3AA 开始,这个问题可以通过同时声明一个网络是点到多点 (**point-to-multipoint**) 和非广播 (**non-broadcast**) 的方式而得到解决,配置如下:

路由器 Rembrandt:

```
interface Serial0
  ip address 172.16.2.1 255.255.255.0
  encapsulation frame-relay
```



```
ip ospf network point-to-multipoint non-broadcast
map-group Leiden
frame-relay lmi-type q933a
frame-relay svc
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.2 cost 30
 neighbor 172.16.2.3 cost 20
 neighbor 172.16.2.4 cost 50
```

路由器 Hals:

```
interface Serial0
 ip address 172.16.2.2 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint non-broadcast
 map-group Haarlem
 frame-relay lmi-type q933a
 frame-relay svc
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.1 priority 10
```

路由器 Vandyck:

```
interface Serial0
 ip address 172.16.2.3 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint non-broadcast
 map-group Antwerp
 frame-relay lmi-type q933a
 frame-relay svc
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.1 priority 10
```

路由器 Brueghel:

```
interface Serial0
 ip address 172.16.2.4 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint non-broadcast
 map-group Brussels
 frame-relay lmi-type q933a
 frame-relay svc
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.1 priority 10
```

由于网络是非广播的, 邻居将不会被自动地发现, 因此必须要手工地配置。另一个在 IOS

11.3AA 中引入的特性可以从路由器 Rembrandt 的配置中看出来：就是可以利用 **neighbor** 命令基于每一个 VC 来指定它们的代价大小。

最后一种解决方案就是，使用它们各自本身的子网把每一个 PVC 连接作为一个单独的点到点网络，如图 9-93 所示。这种解决方法可以通过子接口来完成：

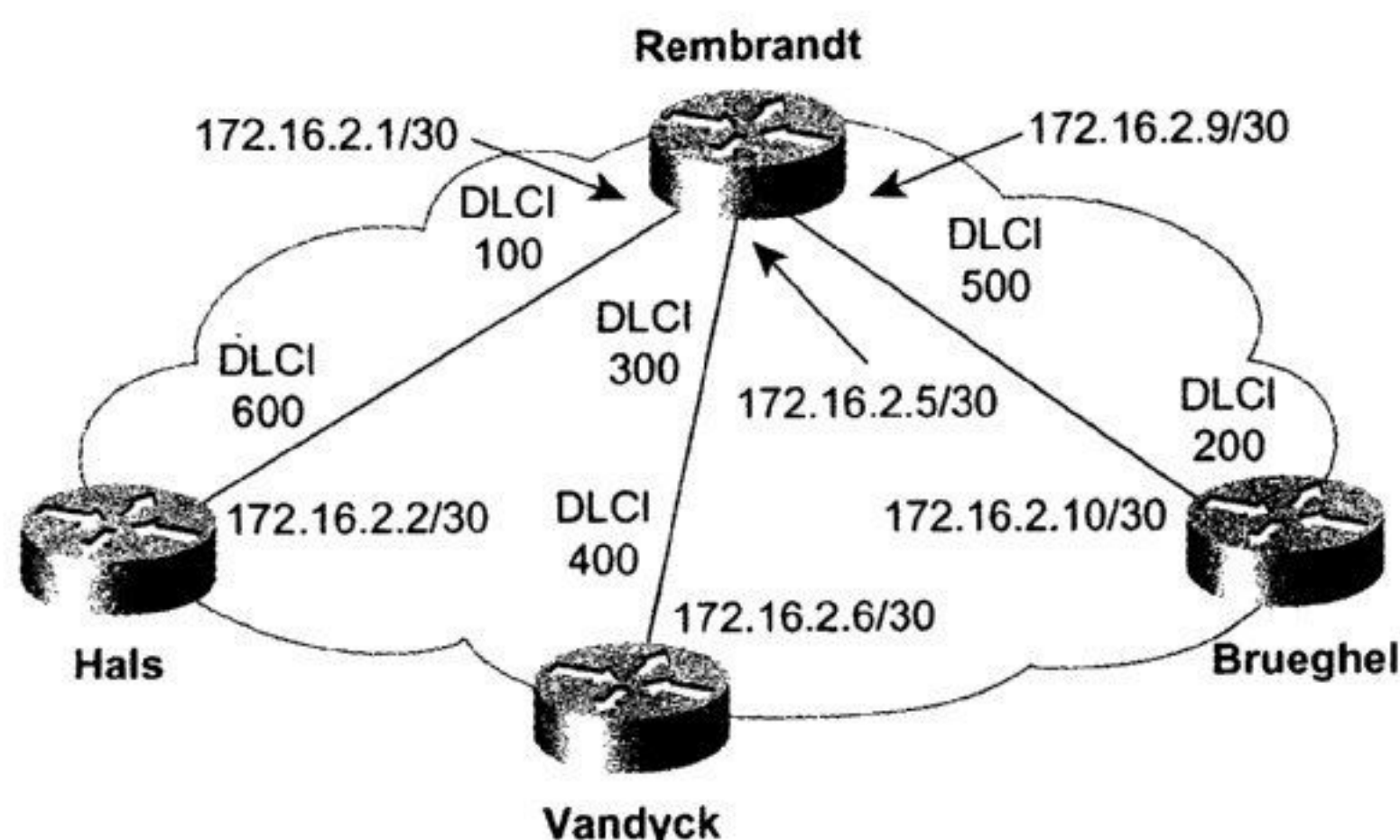


图 9-93 点到点网络的子接口允许每一条 PVC 配置成一个单独的子网，并在 NBMA 网络上排除了 DR/BDR 的选取问题

路由器 Rembrandt:

```
interface Serial0
  no ip address
  encapsulation frame-relay
interface Serial0.100 point-to-point
  description ----- to Hals
  ip address 172.16.2.1 255.255.255.252
  frame-relay interface-dlci 100
interface Serial0.300 point-to-point
  description ----- to Vandyck
  ip address 172.16.2.5 255.255.255.252
  frame-relay interface-dlci 300
interface Serial0.500 point-to-point
  description ----- to Brueghels
  ip address 172.16.2.9 255.255.255.252
  frame-relay interface-dlci 500
!
router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
```

路由器 Hals:

```
interface Serial0
  no ip address
  encapsulation frame-relay
interface Serial0.600
  description ----- to Rembrandt
  ip address 172.16.2.2 255.255.255.252
  frame-relay interface-dlci 600
```



```
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
```

路由器 Vandyck:

```
interface Serial0
 no ip address
 encapsulation frame-relay
interface Serial0.400
 description ----- to Rembrandt
 ip address 172.16.2.6 255.255.255.252
 frame-relay interface-dlci 400
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
```

路由器 Brueghel:

```
interface Serial0
 no ip address
 encapsulation frame-relay
interface Serial0.200
 description ----- to Rembrandt
 ip address 172.16.2.10 255.255.255.252
 frame-relay interface-dlci 200
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
```

对于所有运行在 NBMA 网络上 OSPF 配置来说, 这种配置是最容易管理的。这种配置代码的一些优点是显而易见的, 例如可以使用一个接口号对应于帧中继的 DLCI, 并且包括一个解释行。然而, 还有一个主要的优点就是可以在路由器之间建立简单的一对一的对应关系。

使用子接口的方法有一个不太经常碰到的缺点是, 每一个 PVC 电路都必须拥有自己的子网地址。在大多数情况下, 这个需求不应该会产生问题, 因此 OSPF 协议是支持 VLSM 的。正如这里的例子所显示的, 可以从一个子网地址中创建更小的子网来分配给这个网络“云”是一件容易的事情。而且, 由于 PVC 电路现在是点到点的链路, 也可以使用无 IP 地址编号的方法作为子网地址需求的另一种选择。一个更加谨慎的考虑是子接口会占用更多的内存。在一些内存有限的小型路由器上可能会给路由器带来较大的负担。

9.2.12 案例研究 12: 运行在按需电路上的 OSPF

配置运行在按需电路上的 OSPF 是很简单的, 这可以在与按需电路相连的接口上, 通过增加 **ip ospf demand-circuit** 命令来实现。而且只需要在点到点电路的一端, 或者点到多点电路的多点的那一边宣告为按需电路就可以了。在一般情况下, 运行在按需电路上的 OSPF 不应该在一个广播型介质上实现。这是因为, 在像这样的网络上不能抑制 Hello 报文的发送, 从而使链路一直保持是活动 (up) 的。

如果图 9-92 中的虚电路是帧中继 SVC 电路, 路由器 Rembrandt 可以作如下配置:

```
interface Serial0
```



```
ip address 172.16.2.1 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint non-broadcast
ip ospf demand-circuit
 map-group Leiden
frame-relay lmi-type q933a
frame-relay svc
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.2 cost 30
 neighbor 172.16.2.3 cost 20
 neighbor 172.16.2.4 cost 50
```

在按需电路上实现 OSPF，需要记住以下几点：

- 只有在一个区域的链路状态数据库中的所有 LSA 都设置了 DC-bit 位以后，设置 DoNotAge 位的 LSA 才被允许进入该区域。这种设置方法可以确保该区域的所有路由器都具有识别 DoNotAge 位的能力；
- 实现按需电路上的 OSPF 的区域内的所有路由器都必须具有支持这种配置方式的能力；
- 如果运行在按需电路上的 OSPF 是在一个非末梢区域里实现的，那么在所有的非末梢区域内的路由器都必须支持这种方式。原因是 DC-bit 位的设置是在类型 5 的 LSA 中实现的，而这种类型的 LSA 是泛洪到所有的非末梢区域中的；
- 应该尽量限制在末梢区域、完全末梢区域或 NSSA 区域上实现按需电路的 OSPF。这种实现方式可以取消要求 OSPF 域内的所有路由器支持按需电路上的 OSPF 的需要。这种方式也将因其他区域拓扑改变而引起改变的 LSA 的数量减少到最小，并可以防止过多地使按需电路处于活动状态；
- 如果配置了按需电路上的 OSPF 并且配置了一条虚链路要穿过这条按需电路，那么这条虚链路也将被看作是一条按需电路。另外，这条虚链路的通信流量将会保持按需电路为活动 (up)；
- 每隔 30min，OSPF 协议就会重新刷新一次它的 LSA，以防止这些 LSA 驻留在链路状态数据库时变得无效。由于设置 DoNotAge 位的 LSA 在穿过按需电路的时候不会重新刷新，因此也就丧失了 OSPF 的这个稳定特性；
- 重新刷新过程可以在一条按需电路两端外的其他所有接口上发生，但是 LSA 在通过这条按需电路时不能进行重新刷新。结果，这条链路两端的同一个 LSA 的各自序列号可能会是不同的。网络管理工作站可以使用某些 MIB 变量¹去验证数据库的同步；如果序列号在数据库中不匹配，那么将会错误地报告一个错误。

9.3 OSPF 故障排除

OSPF 协议的故障排除有时是令人恐怖的，这在一个大型互联网络上显得尤其明显。但

¹ 具体来说，就是 ospfExternLSACksumSum 和 ospfAreaLSACksumSum。这些是单独的 LSA 校验和字段的总和。由于校验和的计算包括序列号，并且序列号可能不同，因此，校验和也就可能不同。

是, OSPF 产生的路由选择问题和其他任何路由选择协议的路由选择问题没有什么不同, 故障可能是下列某种原因之一引起的:

- 路由信息丢失;
- 错误的或不精确的路由信息。

对路由器的检查仍然是获取故障排除信息的主要来源。使用命令 **show ip ospf database** 查看不同的 LSA 也会得到重要的信息。例如, 如果一条链路是不稳定的, 那么通告它的 LSA 将会变得频繁变化。这种情况反映在它的序列号会比其他 LSA 的序列号明显的偏高。网络不稳定的另外一个迹象是 LSA 的老化时间从来不会变得很大。

请记住, 一个区域内所有路由器的链路状态数据库都是相同的。因此, 除非你怀疑某些路由器的链路状态数据库本身变得有问题, 否则就可以通过检查某一台路由器的链路状态数据库来检查整个区域的链路状态数据库。另外一个好的经验是为每一个区域的链路状态数据库做一个拷贝 (软拷贝或硬拷贝)。

当检查单独一个路由器的配置时, 需要考虑以下几点:

- 所有接口配置的地址和掩码是否正确?
- **network area** 语句使用的反向掩码是否正确? 是否匹配正确的接口?
- **network area** 语句是否把所有的接口都指定到正确的区域中了?
- **network area** 语句的使用次序正确吗?

当检查邻接关系时 (或者缺少邻接关系时), 请考虑下面这些问题:

- 从这两个邻居路由器中有 Hello 报文正在发送吗?
- 这两个邻居路由器之间的计时器设置相同吗?
- 这两个邻居路由器之间的可选性能字段设置相同吗?
- 接口是配置在同一个子网上的吗 (也就是说, 它们的地址/掩码对是否属于同一个子网)?
- 邻居路由器的接口是否是同一种网络类型?
- 一台路由器是否正在试图和它的邻居路由器的辅助地址形成一个邻接关系?
- 如果使用了认证, 那么在邻居路由器之间的认证类型是否相同? 口令和密钥 (在 MD5 的实例中) 是否相同? 该区域内的所有路由器是否都启用认证了?
- 在所有的访问列表中, 是否有正在阻塞 OSPF 的访问列表?
- 如果邻居关系穿过一条虚链路, 那么这条链路是否配置在了一个末梢区域内了?

如果怀疑某个邻居或者某个邻接关系变得不稳定了, 那么可以通过 **debug ip ospf adj** 命令来监控这些邻接关系。然而, 这个命令经常会产生比所需要的信息多得多的内容, 如图 9-94 所示。它不仅非常详细地记录了邻居状态的变化, 而且也记录了普通的 Hello 报文处理。如果在一个扩展期间执行了监控, 这些正常的信息将会溢出路由器的内部缓冲区。从 IOS11.2 版本开始, 路由器可以在它的 OSPF 配置里增加命令 **ospf log-adjacency-changes** 来监控邻接关系, 这条命令将会记录邻接关系变化的一个简单日志, 如图 9-95 所示。

如果怀疑一个链路状态数据库出现问题, 或者这两个数据库不同步, 那么可以使用命令 **show ip ospf database database-summary** 命令来观察每一台路由器中数据库的 LSA 数量。对于给定的一个区域, 在所有的路由器上, 每一种 LSA 类型的数量都应该是相同的。下一条命令 **show ip ospf database** 将显示出一台路由器数据库的每一条 LSA 的校验和。在一个给定的区域内, 在每一台路由器的数据库中的每一条 LSA 的校验和都应该相同。除非是在一个非常

小的数据库里, 否则校验这个情况的正确与否将非常乏味和痛苦。不过幸运的是, 还有 MIB¹, MIB 可以在一个 SNMP 网络管理平台上面报告一个数据库校验和的总和。如果在一个区域的所有数据库都同步了, 那么每一个数据库中的这个总和都应该是相同的。

当检查一个区域层面上的问题时, 请记住以下几个问题:

- ABR 路由器是否配置正确?
- 对于相同区域的类型是否所有的路由器都配置了? 例如, 如果一个区域是末梢区域, 那么所有的路由器都必须使用 **area stub** 命令。
- 如果配置了地址汇总, 那么配置的正确吗?

```
Hurd#debug ip ospf adj
OSPF adjacency events debugging is on
Hurd#
OSPF: Rcv hello from 172.20.1.2 area 25 from Serial0 172.20.1.2
OSPF: End of hello processing
OSPF: Rcv hello from 10.3.0.1 area 0 from Ethernet0 10.8.1.2
OSPF: Cannot see ourself in hello from 10.3.0.1 on Ethernet0, state INIT
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from FULL to INIT, 1-Way
OSPF: Neighbor change Event on interface Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 0.0.0.0
OSPF: Elect DR 172.20.1.5
      DR: 172.20.1.5 (Id) BDR: none
OSPF: End of hello processing
OSPF: Build router LSA for area 0, router ID 172.20.1.5
OSPF: Build network LSA for Ethernet0, router ID 172.20.1.5
OSPF: No full nbrs to build Net Lsa
OSPF: Flush network LSA on Ethernet0 for area 0
OSPF: Schedule SPF to remove network route
OSPF: Rcv hello from 172.20.1.2 area 25 from Serial0 172.20.1.2
OSPF: End of hello processing
OSPF: Rcv hello from 10.3.0.1 area 0 from Ethernet0 10.8.1.2
OSPF: End of hello processing
OSPF: Rcv DBD from 10.3.0.1 on Ethernet0 seq 0x2653 opt 0x2 flag 0x7 len 32
state INIT
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from INIT to 2WAY, 2-Way
Received
OSPF: 2 Way Communication to 10.3.0.1 on Ethernet0, state 2WAY
OSPF: Neighbor change Event on interface Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 10.3.0.1
OSPF: Elect DR 172.20.1.5
      DR: 172.20.1.5 (Id) BDR: 10.3.0.1 (Id)
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from 2WAY to EXSTART,
AdjOK?
OSPF: Send DBD to 10.3.0.1 on Ethernet0 seq 0x25D7 opt 0x2 flag 0x7 len 32
OSPF: First DBD and we are not SLAVE
OSPF: Rcv DBD from 10.3.0.1 on Ethernet0 seq 0x25D7 opt 0x2 flag 0x2 len 312
state EXSTART
OSPF: NBR Negotiation Done. We are the MASTER
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from EXSTART to EXCHANGE,
Negotiation Done
OSPF: Send DBD to 10.3.0.1 on Ethernet0 seq 0x25D8 opt 0x2 flag 0x3 len 292
OSPF: Database request to 10.3.0.1
OSPF: sent LS REQ packet to 10.8.1.2, length 36
OSPF: Rcv DBD from 10.3.0.1 on Ethernet0 seq 0x25DB opt 0x2 flag 0x0 len 32
state EXCHANGE
OSPF: Send DBD to 10.3.0.1 on Ethernet0 seq 0x25D9 opt 0x2 flag 0x1 len 32
OSPF: Rcv DBD from 10.3.0.1 on Ethernet0 seq 0x25D9 opt 0x2 flag 0x0 len 32
state EXCHANGE
OSPF: Exchange Done with 10.3.0.1 on Ethernet0
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from EXCHANGE to LOADING,
Exchange Done
OSPF: Synchronized with 10.3.0.1 on Ethernet0, state FULL
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from LOADING to FULL,
Loading Done
OSPF: Build router LSA for area 0, router ID 172.20.1.5
OSPF: Build network LSA for Ethernet0, router ID 172.20.1.5
OSPF: Rcv hello from 172.20.1.2 area 25 from Serial0 172.20.1.2
OSPF: End of hello processing
OSPF: Rcv hello from 10.3.0.1 area 0 from Ethernet0 10.8.1.2
OSPF: Neighbor change Event on interface Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 10.3.0.1
OSPF: Elect DR 172.20.1.5
      DR: 172.20.1.5 (Id) BDR: 10.3.0.1 (Id)
OSPF: End of hello processing
OSPF: Build router LSA for area 0, router ID 172.20.1.
```

图 9-94 使用命令 **debug ip ospf adj** 输出的调试信息显示, 当一个邻居路由器的以太接口暂时断开随后又重新连接上的结果

¹ 也就是 `ospfExternLsaChecksumSum` 和 `ospfAreaLsaChecksumSum`。


```

Hurd#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level debugging, 19 messages logged
  Monitor logging: level debugging, 0 messages logged
  Trap logging: level informational, 23 message lines logged
  Buffer logging: level debugging, 19 messages logged

Log Buffer (4096 bytes):

%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from FULL to INIT, 1-Way
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from INIT to 2WAY, 2-Way Received
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from 2WAY to EXSTART, AdjOK?
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from EXSTART to EXCHANGE, Negotiation Done
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from EXCHANGE to LOADING, Exchange Done
%OSPF-5-ADJCHG: Process 1, Nbr 10.3.0.1 on Ethernet0 from LOADING to FULL, Loading Done
Hurd#

```

图 9-95 这些日志信息的记录结果来自于命令 `ospf log-adjacency-changes`，显示了图 9-94 中的同一个邻居路由器失效的信息，但是没有上面的调试信息那么详细

如果是路由器的性能出现了问题，那么可以在路由器上检查它们的 CPU 和内存的使用情况。如果内存的使用率在 70% 以上，那么可能是链路状态数据库太大了；如果 CPU 的利用率一直保持在 60% 以上，那么网络的拓扑可能存在不稳定的情况。如果内存或 CPU 的利用率超过了 50% 的警戒线，网络管理员就应该开始分析网络性能加重的原因，并基于得出的分析结果，来制定改善网络的升级计划。

末梢区域和地址汇总能够帮助减小链路状态数据库的大小并能容忍网络的一些不稳定。最能加重一台 OSPF 路由器负担的是对 LSA 的处理，而不是 SPF 算法的计算。在个别情况下，类型 1 和类型 2 的 LSA 对路由器处理器的影响比汇总 LSA 更大。但是，类型 1 和类型 2 的 LSA 可以编成组发送，而汇总 LSA 却只能在单个报文中发送。结果，汇总 LSA 实际上对路由器处理器的影响更大。

下面的案例研究演示了进行 OSPF 协议的故障排除时，使用最频繁的技巧和工具。

9.3.1 案例研究 13：孤立的区域

区域内的数据包可以在图 9-96 中的区域 1 内进行路由转发，但是所有的区域间通信的尝试都失败了。这种情况下，首先应该马上怀疑是区域 1 的 ABR 路由器出现了故障。而且由于内部路由器没有关于 ABR 路由器的路由器入口，这个事实更加坚信可能是 ABR 路由器出现了问题（如图 9-97）所示。

下一步就是验证连接到 ABR 路由器的物理链路是否正常和 OSPF 协议是否在正常工作。如图 9-98 所示，在上述的那一台内部路由器的邻居表中，显示关于 ABR 路由器的邻居状态都是完全邻接的（Full），这表明邻接关系是存在的。事实上，这个 ABR 路由器是这里令牌环网络的 DR 路由器。邻接关系的存在证实链路是正常的，而且也可以交换含有正确参数的 OSPFHello 报文。

在路由器 National 的链路状态数据库和它的路由选择表中可以发现一些其他的和这个故障有关的迹象。如图 9-99 所示，路由器 National 的数据库中只包含了路由器 LSA（类型 1）和网络 LSA（类型 2），但是没有记录通告区域外部目的地址的网络汇总 LSA（类型 3）。同

时, 出现了由路由器 Whitney (1.1.1.1) 始发的 LSA。这个信息再次表明路由器 Whitney 与路由器 National 是有邻接关系的, 但是却没有信息从区域 0 通过它到达区域 1。

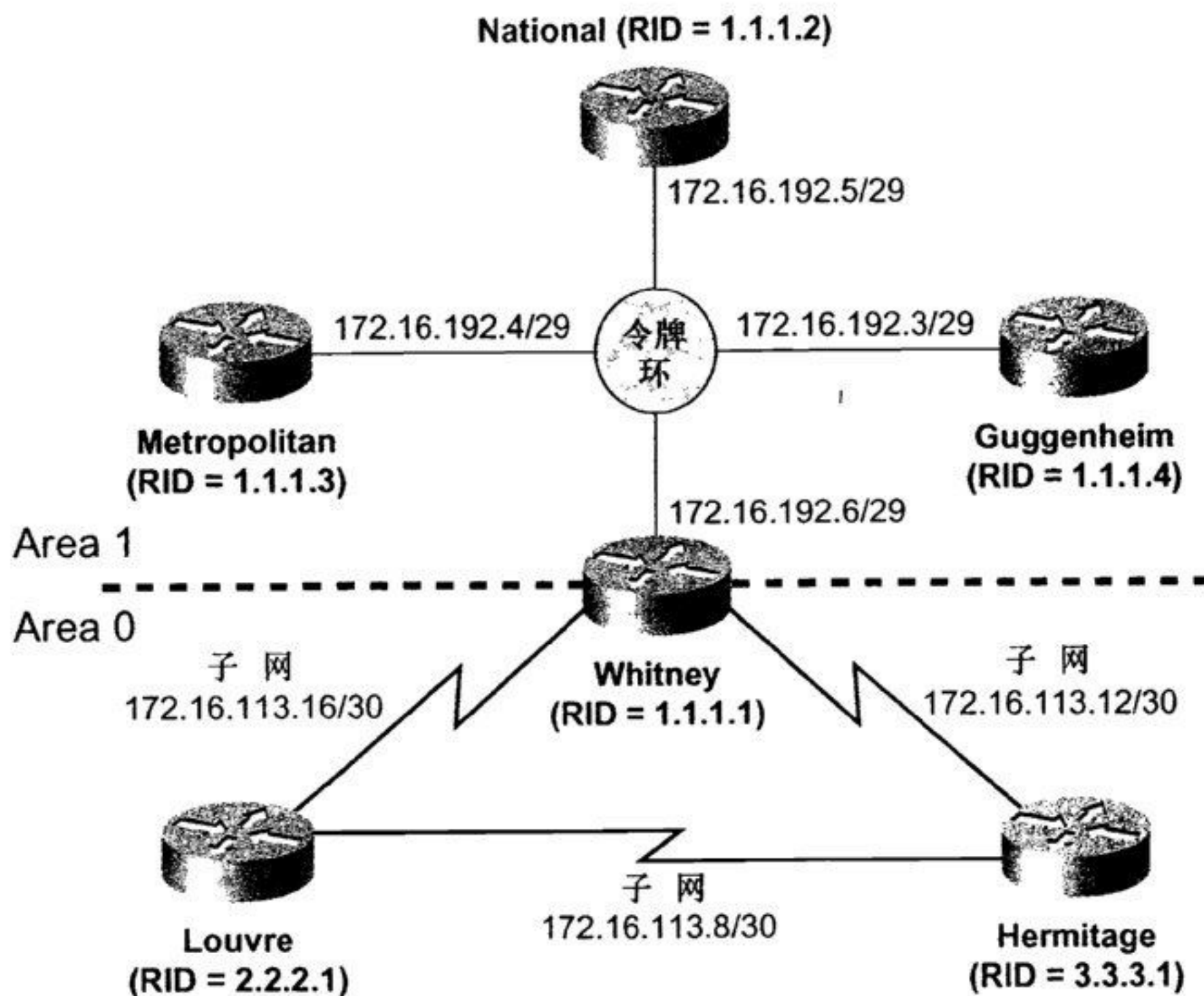


图 9-96 区域 1 内的系统和路由器通信正常, 但是没有通信流量可以通过或者来自区域 0

```
National#show ip ospf border-routers

OSPF Process 8 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route

National#
```

图 9-97 使用命令 **show ip ospf border-router** 可以检查内部路由器的内部路由选择表信息。图中显示没有关于 ABR 路由器的路由入口

```
National#show ip ospf neighbor

Neighbor ID    Pri   State       Dead Time   Address        Interface
1.1.1.1        1     FULL/DR     00:00:33   172.16.192.6   TokenRing0
1.1.1.3        1     FULL/BDR    00:00:34   172.16.192.4   TokenRing0
1.1.1.4        1     FULL/-      00:00:30   172.16.192.3   TokenRing0
National#
```

图 9-98 路由器 National 的邻居表表明与 ABR 路由器 (1.1.1.1) 的邻接关系是完全邻接的

如图 9-100 所示, 在路由器 National 的路由选择表中, 区域 1 外部惟一的地址是与路由器 Whitney 相连的串行链路地址。然而, 在这儿还显示了另外一个线索: 这些路由条目是被标记成区域内路由 (O) 的。而依照如图 9-96 所示, 这些路由应该是在区域 0 内的, 因此, 它们应该是被标记成区域间路由 (O IA)。现在, 问题显然出在 ABR 路由器的区域 0 的一端。


```
National#show ip ospf database

      OSPF Router with ID (1.1.1.2) (Process ID 8)

          Router Link States (Area 1)

Link ID      ADV Router  Age      Seq#          Checksum      Link count
172.16.192.6  1.1.1.1      132      0x800000034   0xAC4D        3
172.16.219.120 1.1.1.2 1 458      0x80000002B   0x6B46        2

          Net Link States (Area 1)

Link ID      ADV Router  Age      Seq#          Checksum
172.16.192.6  1.1.1.1      132      0x80000002E   0x2078
National#
```

图 9-99 路由器 National 的链路状态数据库也显示, 和路由器 Whitney 之间是有邻接的, 但是却没有通告区域间的目的地址

```
National#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
       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

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
C       172.16.219.112/28 is directly connected, Serial0
C       172.16.192.0/29 is directly connected, TokenRing0
O       172.16.113.12/30 [110/70] via 172.16.192.6, 03:01:43, TokenRing0
O       172.16.113.16/30 [110/70] via 172.16.192.6, 03:01:43, TokenRing0
National#
```

图 9-100 路由器 Whitney 正在通告它的串行接口的子网, 但是它们正在被作为区域内的目的地址通告

如图 9-101 所示, 虽然还没有发现问题产生的最终原因, 但是通过对路由器 Whitney 的串行链路的检查已经发现了问题所在。这两个串行接口应该在区域 0 内, 但都被替代为区域 1 了。它们都和网络逻辑拓扑上的邻居 (路由器 Louvre 和 Hermitage) 相连, 但是却没有记录 OSPF 邻居。正在有规律地显示的错误信息表明路由器 Whitney 正在接收来自于路由器 Louvre 和 Hermitage 的 Hello 报文, 而这些报文的区域字段设置为 0, 因而引起不匹配的情况发生。

路由器 Whitney 的 OSPF 配置如下:

```
router ospf 8
 network 172.16.0.0 0.0.255.255 area 1
 network 172.16.113.0 0.0.0.255 area 0
```

乍一看, 这个配置好像是没有问题的。但是, 回忆一下第一个案例研究中提到的, **network area** 命令是连续地执行的。第二条 **network area** 命令只可能影响到那些和第一条 **network area** 命令不匹配的接口。在这样的配置下, 所有匹配第一个 **network area** 命令语句的接口就都被设置到区域 1 了。而第二条命令并没有被应用。

正确的配置应该是:

```
router ospf 8
```



```
network 172.16.192.0 0.0.0.255 area 1
network 172.16.113.0 0.0.0.255 area 0
```

当然, 这里可以有多种有效的正确配置。但重要的一点是第一个 **network area** 命令必须足够精确地只去匹配区域 1 的地址, 而不含有区域 0 接口的地址。

```
Whitney#show ip ospf interface serial0
Serial0 is up, line protocol is up
Internet Address 172.16.113.18/30, Area 1
Process ID 8, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:05
Neighbor Count is 0, Adjacent neighbor count is 0
Whitney#show ip ospf interface serial 1
Serial1 is up, line protocol is up
Internet Address 172.16.113.14/30, Area 1
Process ID 8, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:09
Neighbor Count is 0, Adjacent neighbor count is 0
Whitney#
%OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area
must be virtual-link but not found from 172.16.113.13, Serial1
%OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area
must be virtual-link but not found from 172.16.113.17, Serial0
```

图 9-101 路由器 Whitney 的串行接口被配置成区域 1, 从而替代了区域 0:
这个配置在接收区域 0 的 Hello 报文时会引起错误信息

9.3.2 案例研究 14: 路由汇总配置错误

如图 9-102 所示, 显示了一个骨干区域和与之相连的 3 个区域。为了减小链路状态数据库的大小和增强网络的稳定性, 在区域之间使用了路由汇总。

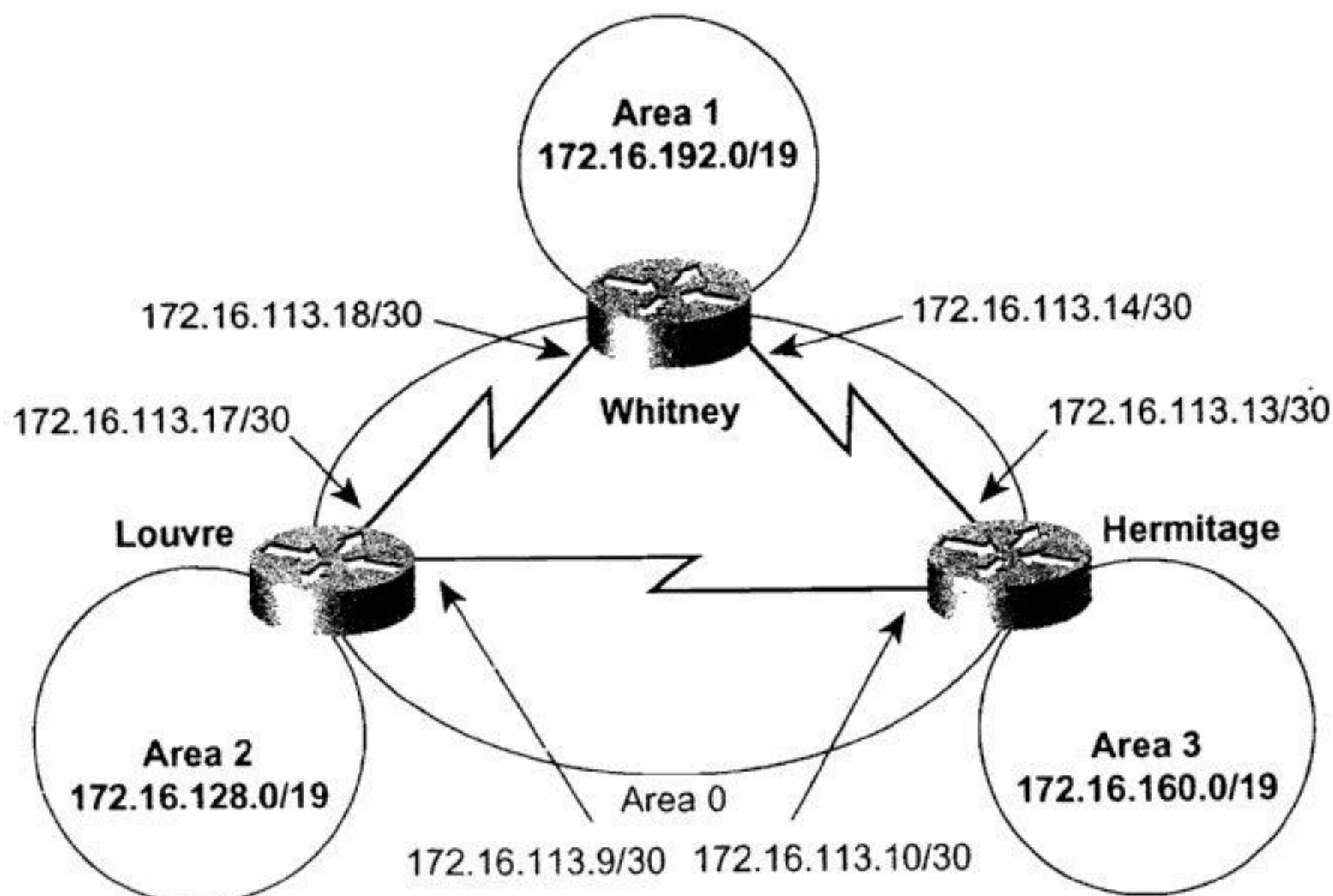


图 9-102 这里显示了被通告到区域 0 内的每一个区域的汇总地址。区域 0 的子网地址也被汇总到其他的区域中

图中显示的每个区域的地址汇总了这3个非骨干区域内的个别子网。例如, 区域1内的一些子网可能是:

172.16.192.0/29
172.16.192.160/29
172.16.192.248/30
172.16.217.0/24
172.16.199.160/29
172.16.210.248/30

图9-103中显示了这些子网地址能够被汇总成地址172.16.192.0/19。

```

10101100000100001100000000000000 = 172.16.192.0/29
10101100000100001100000011111000 = 172.16.192.248/30
10101100000100001101100100000000 = 172.16.217.0/24
10101100000100001100011110100000 = 172.16.199.160/29
10101100000100001101001011111000 = 172.16.210.248/30
10101100000100001100000000000000 = 172.16.192.0/19

```

图9-103 一些子网地址能够被汇总成地址172.16.192.0/19。粗体字部分表明了每一个地址的网络位

路由器 Whitney 的配置为:

```

router ospf 8
 network 172.16.192.0 0.0.0.255 area 1
 network 172.16.113.0 0.0.0.255 area 0
 area 1 range 172.16.192.0 255.255.224.0
 area 0 range 172.16.113.0 255.255.224.0

```

在其他3个ABR路由器上也做类似地配置。每台ABR路由器将向区域0通告与之相连的非骨干区域的汇总地址, 并且也把区域0的子网地址汇总到了非骨干区域中去。

图9-104中显示出了一个问题。在查看区域1的某个内部路由器的路由选择表时, 发现区域0的子网地址没有被正确地汇总(讲得再清楚一点, 区域1的内部子网也没有显示出来)。虽然关于区域2和区域3的汇总地址被显示出来, 但是在路由选择表中区域0的汇总地址却被它内部单独的子网替代了。

```

National#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
       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

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 7 subnets, 4 masks
O IA   172.16.160.0/19 [110/80] via 172.16.192.6, 1d22h, TokenRing0
O IA   172.16.128.0/19 [110/80] via 172.16.192.6, 1d22h, TokenRing0
C       172.16.192.0/29 is directly connected, TokenRing0
O IA   172.16.113.12/30 [110/70] via 172.16.192.6, 00:39:46, TokenRing0
O IA   172.16.113.8/30 [110/134] via 172.16.192.6, 00:39:46, TokenRing0
O IA   172.16.113.16/30 [110/70] via 172.16.192.6, 00:39:46, TokenRing0
National#

```

图9-104 在区域1内的某个内部路由器的路由选择表中记录了区域0的个别子网, 而替代了应该出现的汇总地址

当网络管理员以二进制的表示方式检查区域0的3个子网时,就会发现关于区域0的 **area range** 命令可能有问题 (如图 9-105)。

```

10101100000100000111000100001000 = 172.16.113.8/30
10101100000100000111000100001100 = 172.16.113.12/30
10101100000100000111000100010000 = 172.16.113.16/30
11111111111111111110000000000000 = 255.255.224.0
10101100000100000110000000000000 = 172.16.96.0

```

图 9-105 区域0的子网、所配置的汇总掩码和正确的汇总地址

从上面可以看出,这里的问题是 **area range** 命令指定汇总地址 (172.16.113.0) 比它携带的掩码 (255.255.224.0) 更具体了。对于这个 19 位的掩码,正确的地址应该是 172.16.96.0:

```

router ospf 8
 network 172.16.192.0 0.0.0.255 area 1
 network 172.16.113.0 0.0.0.255 area 0
 area 1 range 172.16.192.0 255.255.224.0
 area 0 range 172.16.96.0 255.255.224.0

```

图 9-106 中显示了更改配置后的路由选择表的结果。当然,对于汇总区域0内的地址也有可以选择的其他地址。例如,172.16.113.0/24 和 172.16.113.0/27 也都是可用的。最恰当的汇总地址是依赖于互联网络设计的优先性选择的。例如这个实例,在图 9-99 中显示的互联网络里,选用 172.16.96.0/19 是为了保持配置的一致性——所有的汇总地址都使用了 19 位的掩码。另一方面,为了网络更好的扩展性,应该选用 172.16.113.0/27 作为汇总地址。在这个汇总地址里,还可以增加 5 个子网用于骨干区域,而剩余的更为广泛的地址范围可以用于网络上的其他地方。

```

National#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
       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

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
O IA 172.16.160.0/19 [110/80] via 172.16.192.6, 00:38:11, TokenRing0
O IA 172.16.128.0/19 [110/80] via 172.16.192.6, 1d23h, TokenRing0
C     172.16.192.0/29 is directly connected, TokenRing0
O IA 172.16.96.0/19 [110/70] via 172.16.192.6, 00:00:23, TokenRing0
National#

```

图 9-106 区域0现在可以被正确地汇总了

9.4 展 望

当提到链路状态路由选择协议的时候,大多数人们首先想到的是 OSPF 协议。但是,OSPF 协议并不是 IP 网络上惟一的链路状态协议。ISO 的中间系统—中间系统 (IS-IS) 虽然是设计

用来为其他网络协议进行路由选择的,但是它也可以为 IP 网络进行路由选择。第 10 章将讲述这个鲜为人知的链路状态路由选择协议。

9.5 总结表: 第 9 章命令总结

命 令	描 述
area area-id authentication [message-digest]	使一个区域的类型 1 或者类型 2 的认证有效
area area-id default-cost cost	为 ABR 路由器发送到一个末梢区域的缺省路由指定一个代价值
area area-id nssa [no-redistribution][default-information-originate][no-summary]	配置一个区域为非纯末梢区域 (NSSA)
area area-id range address mask	汇总地址进入或离开一个区域
area area-id stub [no-summary]	配置一个区域作为末梢区域或者完全末梢区域
area area-id virtual-link router-id	在 ABR 路由器之间定义一条虚链路
debug ip ospf adj	显示有关一个 OSPF 邻接关系的创建或中断的事件
ip ospf authentication-key password	使用类型 1 的认证方式分配一个口令给一个 OSPF 接口
ip ospf cost cost	在一个 OSPF 接口上指定出站接口的代价大小
ip ospf dead-interval seconds	在一个接口上指定 OSPF 的 RouterDeadInterval 大小
ip ospf demand-circuit	配置一个接口作为 OSPF 按需链路
ip ospf hello-interval seconds	为一个接口指定 OSPF 的 HelloInterval 的值
ip ospf message-digest-key key-id md5 key	使用类型 2 的认证方式指定一个接口的密钥 ID 和密钥 (口令)
ip ospf name-lookup	使域名的反向 DNS 域名查找有效,以便在某些 show 命令匹配路由器 ID
ip ospf network [broadcast] [nonbroadcast][point-to-multipoint]	配置 OSPF 网络类型
ip ospf priority number	设定一个接口的路由器优先级,以便用来选取 DR 和 BDR 路由器
ip ospf retransmit-interval seconds	设置一个接口的 OSPF RxmtInterval 值
ip ospf transmit-delay seconds	设置一个接口的 OSPF InfTransDelay 值
maximum-paths	设置 OSPF 执行负载均衡的最大路径数量
neighbor ip-address [priority number] [poll-interval seconds][cost cost]	在一个非广播网络上手工配置邻居路由器
network address inverse-mask area area-id	指定运行 OSPF 协议的接口,并指定这些接口相连的 OSPF 区域
ospf auto-cost reference-bandwidth reference-bandwidth	为计算链路的代价,改变缺省的 OSPF 参考带宽
ospf log-adjacency-changes	记录邻居状态的改变
router ospf process-id	启动一个 OSPF 路由选择进程
show ip ospf [process-id]	显示有关 OSPF 路由选择进程的一般信息
show ip ospf border-routers	显示一个路由器的内部 OSPF 路由选择表
show ip ospf [process-id area-id] database	显示 OSPF 链路状态数据库中的所有条目
show ip ospf [process-id area-id] database router [link state-id]	显示 OSPF 链路状态数据库中的类型 1 的 LSA
show ip ospf [process-id area-id] database network [link state-id]	显示 OSPF 链路状态数据库中的类型 2 的 LSA
show ip ospf [process-id area-id] database summary [link state-id]	显示 OSPF 链路状态数据库中的类型 3 的 LSA
show ip ospf [process-id area-id] database asbr-summary [link state-id]	显示 OSPF 链路状态数据库中的类型 4 的 LSA
show ip ospf [process-id area-id] database nssa-external [link state-id]	显示 OSPF 链路状态数据库中的类型 7 的 LSA
show ip ospf [process-id area-id] database external [link state-id]	显示 OSPF 链路状态数据库中的类型 5 的 LSA
show ip ospf [process-id area-id] database database-summary	根据类型和区域 ID 显示 OSPF 链路状态数据库中 LSA 的数量
show ip ospf interface [type number]	显示一个接口具体的 OSPF 信息

续表

命 令	描 述
<code>show ip ospf neighbor [type number][neighbor-id][detail]</code>	显示 OSPF 邻居表的信息
<code>show ip ospf virtual-link</code>	显示有关 OSPF 虚链路的信息
<code>timer lsa-group-pacing pacing-time</code>	在重刷新计时器超时的两组 LSA 之间设定的最小步调时间

9.6 推荐读物

John Moy, “OSPF Version 2”, RFC 2328, 1998 年 4 月。

John Moy, *OSPF: Anatomy of an Internet Routing Protocol*. Reading, Massachusetts: Addison-Wesley, 1998.

9.7 复 习 题

1. 什么是 OSPF 邻居?
2. 什么是 OSPF 邻接关系?
3. OSPF 报文的 5 种类型是什么? 每一种类型的用途是什么?
4. 什么是 LSA? 怎样区分一个 LSA 和一个 OSPF 更新报文的不同?
5. LSA 的类型 1 到类型 5, 以及类型 7 分别是什么? 每一种类型的用途是什么?
6. 什么是链路状态数据库? 链路状态数据库的同步是什么意思?
7. 什么是缺省的 HelloInterval?
8. 什么是缺省的 RouterDeadInterval?
9. 什么是路由器 ID? 怎么样确定一个路由器 ID?
10. 什么是区域?
11. 区域 0 的含义是什么?
12. 什么是最大生存时间 (MaxAge)?
13. OSPF 协议的 4 种路由器类型是什么?
14. OSPF 协议的 4 种路径类型是什么?
15. OSPF 协议的 5 种网络类型是什么?
16. 什么是指定路由器 DR?
17. 在 Cisco 的路由器上是怎样计算一个接口的出站代价的?
18. 什么是分段的区域?
19. 什么是虚链路?
20. 末梢区域、完全末梢区域和非纯末梢区域之间有什么不同?
21. OSPF 网络条目和 OSPF 路由器条目之间有什么不同之处?
22. 为什么类型 2 的认证方式比类型 1 的认证方式更好?
23. 在 LSA 头部中哪 3 个字段是用来区分不同的 LSA 的? 另外, 在 LSA 头部中哪 3 个字段是用来区分相同 LSA 的不同实例的?

9.8 配置练习

1. 表 9-13 显示了 14 台路由器的接口和地址，其中也表明了每一个接口相连的 OSPF 区域。根据表中提供的信息，假定下面的事实：

- 每一台路由器的所有接口都显示在表中了。
- 如果没有区域显示 (-)，就表示在相关的接口上不运行 OSPF 协议。
- 子网地址的第二个 8bit 字节和区域 ID 相同。
- 每一个 OSPF 接口地址的前 16 位指定一个区域。例如，前缀是 10.30.x.x 的地址将只能在区域 30 里出现。

请写出表 9-13 中的路由器关于 OSPF 的配置（提示：可以首先画一个路由器和子网的拓扑图）。

表 9-13 关于配置练习 1~6 的路由器信息

路 由 器	接 口	地址/掩码	区域 ID
A	L0	10.100.100.1/32	-
	E0	10.0.1.1/24	0
	E1	10.0.2.1/24	0
	E2	10.0.3.1/24	0
	E3	10.0.4.1/24	0
B	L0	10.100.100.2/32	-
	E0	10.0.1.2/24	0
	E1	10.5.1.1/24	5
	S0	10.5.255.13/30	5
	S1	10.5.255.129/30	5
C	L0	10.100.100.3/32	-
	E0	10.0.2.2/24	0
	E1	10.10.1.1/24	10
	S0	10.30.255.249/30	30
D	L0	10.100.100.4/32	-
	E0	10.0.3.2/24	0
	E1	10.20.1.1/24	20
E	L0	10.100.100.5/32	-
	E0	10.0.4.2/24	0
	S0	10.15.255.1/30	15
F	L0	10.100.100.6/32	-
	E0	10.5.5.1/24	5
	S0	10.5.255.130/30	5
	S1	10.5.255.65/30	5
G	L0	10.100.100.7/32	-
	E0	10.10.1.58/24	10
	S0	10.10.255.5/30	-
H	L0	10.100.100.8/32	-
	E0	10.20.1.2/24	20
	E1	10.20.100.100/27	20
	S0	10.20.255.225/30	-
I	L0	10.100.100.9/32	-
	E0	10.35.1.1/24	35
	S0	10.5.255.66/30	5
J	L0	10.100.100.10/32	-
	E0	10.15.227.50/24	15
	S0	10.15.225.2	15

续表

路 由 器	接 口	地址/掩码	区域 ID
K	L0	10.100.100.11/32	-
	E0	10.30.1.1/24	30
	S0*	10.30.254.193/26	30
L	L0	10.100.100.12/32	-
	E0	10.30.2.1/24	30
	S0*	10.30.254.194/26	30
M	L0	10.100.100.13/32	-
	E0	10.30.3.1/24	30
	S0*	10.30.254.195/26	30
	S1	10.30.255.250/30	30
N	L0	10.100.100.14/32	-
	E0	10.30.4.1/24	30
	S0*	10.30.254.196/26	30

* 表示帧中继封装。

2. 在表 9-13 中的所有 ABR 路由器上配置路由汇总。
3. 更改配置, 使区域 15 成为一个末梢区域。
4. 更改配置, 使区域 30 成为一个完全末梢区域。
5. 路由器 H 的 S0 接口和一个运行其他路由选择协议的路由器相连, 并将那个路由选择协议学习到的路由重新分配到 OSPF 域中。更改必要的配置, 以便使这些重新分配的路由器可以在整个 OSPF 域内通告, 但是不允许任何类型 5 的 LSA 通告到区域 20 里面。
6. 在路由器 C 和路由器 M 之间的串行链路是一条带宽很低的链路。更改配置, 以便使 OSPF 协议把这条链路作为一个按需链路看待。

9.9 故障排除练习

1. 在两台路由器上的 OSPF 不能工作。当打开 **debug** 命令进行调试后, 发现每 10s 就会收到图 9-107 中显示的信息。请问网络发生了什么故障?

```

RTR_EX1#debug ip ospf adj
OSPF adjacency events debugging is on
RTR_EX1#
OSPF: Rcv pkt from 172.16.27.1, TokenRing0, area 0.0.0.25 : src not on the same network
OSPF: Rcv pkt from 172.16.27.1, TokenRing0, area 0.0.0.25 : src not on the same network
OSPF: Rcv pkt from 172.16.27.1, TokenRing0, area 0.0.0.25 : src not on the same network
OSPF: Rcv pkt from 172.16.27.1, TokenRing0, area 0.0.0.25 : src not on the same network
OSPF: Rcv pkt from 172.16.27.1, TokenRing0, area 0.0.0.25 : src not on the same network

```

图 9-107 故障排除练习 1 的调试信息

2. 根据图 9-108 中显示的调试信息, 查明网络出现的故障。
3. 根据图 9-109 中显示的错误信息, 查明网络出现的故障。
4. 根据图 9-110 中显示的错误信息查明网络出现的故障。
5. 根据图 9-111 中显示的错误信息查明网络出现的故障。
6. 在图 9-112 中所示的路由器上做如下配置。


```

RTR_EX2#debug ip ospf adj
OSPF adjacency events debugging is on
RTR_EX2#
OSPF: Hello from 172.16.27.195 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.20.1.1 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.16.27.195 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.20.1.1 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.16.27.195 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.20.1.1 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.16.27.195 with mismatched Stub/Transit area option bit
OSPF: Hello from 172.20.1.1 with mismatched Stub/Transit area option bit

```

图 9-108 故障排除练习 2 的调试信息

```

RTR_EX3#
OSPF: Send with youngest Key 10
OSPF: Rcv pkt from 10.8.1.1, Ethernet0 : Mismatch Authentication type. Input
packet specified type 0, we use type 2
OSPF: Send with youngest Key 10
OSPF: Rcv pkt from 10.8.1.1, Ethernet0 : Mismatch Authentication type. Input
packet specified type 0, we use type 2
RTR_EX3#

```

图 9-109 故障排除练习 3 的错误信息

```

RTR_EX4#
OSPF: Send with youngest Key 10
OSPF: Rcv pkt from 10.8.1.1, Ethernet0 : Mismatch Authentication Key - Message D
igest Key 10
OSPF: Send with youngest Key 10
OSPF: Rcv pkt from 10.8.1.1, Ethernet0 : Mismatch Authentication Key - Message D
igest Key 10
RTR_EX4#

```

图 9-110 故障排除练习 4 的错误信息

```

RTR_EX5#
%OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area must be virtual-link
but not found from 10.8.1.1, Ethernet0
%OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area must be virtual-link
but not found from 10.8.1.1, Ethernet0
RTR_EX5#

```

图 9-111 故障排除练习 5 的错误信息

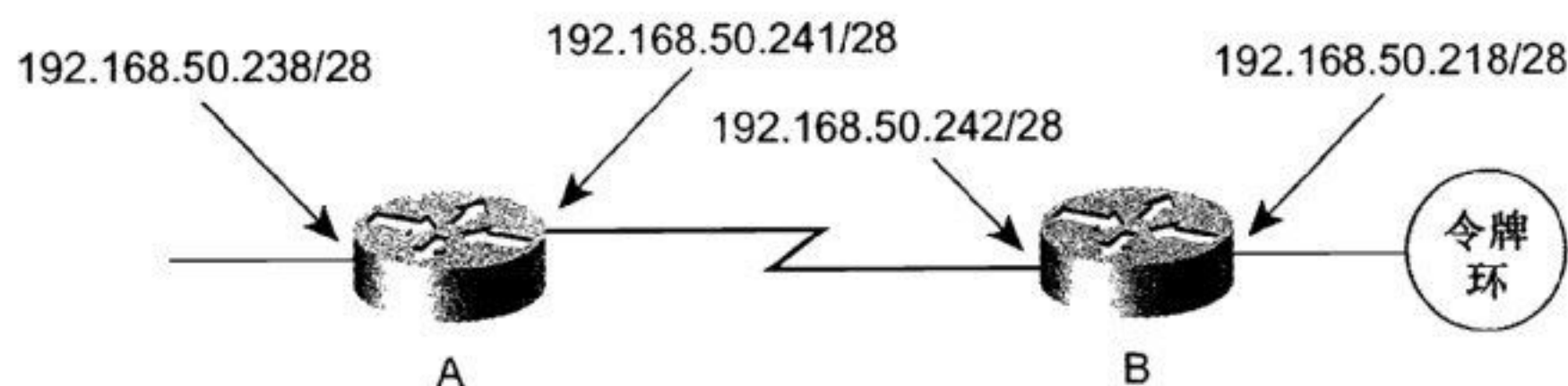


图 9-112 故障排除练习 6 的互联网络

路由器 A:

```
router ospf 15
```



```

network 192.168.50.224 0.0.0.31 area 192.168.50.0
network 192.168.50.240 0.0.0.15 area 0.0.0.0
area 192.168.50.0 authentication message-digest

```

路由器 B:

```

router ospf 51
network 192.168.50.0 0.0.0.255 area 0

```

在这里, 路由器 A 和 B 不能形成一个邻接关系。请问发生了什么问题?

7. 图 9-113 显示了某个区域里的一个链路状态数据库, 其中这个区域里存在一条不稳定的链路。基于图中所示的信息, 哪一条链路看起来像是有问题的链路?

```
RTR_EX7#show ip ospf database
```

```
OSPF Router with ID (10.8.20.1) (Process ID 1)
```

```
Router Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.3.0.1	10.3.0.1	18	0x8000001B	0x6AF8	5
10.8.5.1	10.8.5.1	15	0x80000267	0xFDA0	6
10.8.20.1	10.8.20.1	478	0x800000	1E 0xD451	4

```
Net Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum
10.8.1.2	10.3.0.1	18	0x80000013	0xA747

```
RTR_EX2#
```

图 9-113 故障排除练习 7 的链路状态数据库

第 10 章

集成 IS-IS 协议

本章包括以下主题：

- 集成 IS-IS 协议的操作
 - IS-IS 区域
 - 网络实体标题
 - IS-IS 的功能结构
 - IS-IS 的 PDU 格式
- 集成 IS-IS 协议的配置
 - 案例研究：一个基本的 IS-IS 配置
 - 案例研究：更改路由器的类型
 - 案例研究：区域的迁移
 - 案例研究：路由汇总
 - 案例研究：认证
- 集成 IS-IS 协议的故障排除
 - IS-IS 邻接关系的故障排除
 - IS-IS 链路状态数据库的故障排除
 - 案例研究：运行于 NBMA 网络上的集成 IS-IS

当人们一提到链路状态协议和 IP 协议的术语时，大多数人会立即想到 OSPF 协议。一些人会说：“哦，是的，也有 IS-IS 协议，但是用得不多。”只有少数人会认真地考虑使用集成的 IS-IS 协议替换 OSPF 协议。但是，IS-IS 协议虽然为数不多但毕竟还是存在，并且运行在一些互联网络上——包括一些 ISP 运营商——也使用 IS-IS 协议进行 IP 路由选择。

IS-IS 的意思是表示中间系统到中间系统，并且是为 ISO 无连接网络协议（ISO's Connectionless Network Protocol, CLNP）设计的路由选择协议。IS-IS 协议是由 ISO10589 定义

和解释的。¹这个协议是由数字设备公司 DEC 的 DECnet PhaseV 发展而来的。

ISO 发展 IS-IS 协议的时间和 IAB (Internet Architecture Board, Internet 体系结构委员会) 发展 OSPF 协议的时间基本是同一时期, 只是稍早或稍迟一点而已。并且有一个提议, 建议采用 IS-IS 协议替代 OSPF 协议作为 TCP/IP 协议的路由选择协议。这种提法是由这样一种观点驱动的, 即 TCP/IP 协议只是一个过渡的协议族, 并且最终会被 OSI 协议族代替。加强这种向着 OSI 发展的推动力来自于一些技术规范, 例如 GOSIP 和 EPHOS 等。GOSIP (United States' Government Open Systems Interconnection Profile) 是指美国政府开放系统互连规范, EPHOS (European Procurement Handbook for Open Systems) 是指欧洲国家关于开放系统的采购指南。

为了支持从 TCP/IP 协议向 OSI 协议这个可以预见的转换, 又提议出一个扩展的 IS-IS 协议,²称为集成 IS-IS 协议。提出集成 IS-IS 协议的目的是为了把它作为一个具有双重功能的 IS-IS 协议, 即利用单一一个路由选择协议同时为 CLNS 协议³和 IP 协议提供路由选择的能力。这个协议可以设计用来在一个单纯的 CLNS 环境, 一个单纯的 IP 环境, 或者一个 CLNS/IP 的混和环境中运行。

说是画了一条战线可能有点过分夸张, 但是至少是形成了两个鲜明的派别——ISO 的支持者和 OSPF 的支持者。读者应该阅读和对比一些经典的书籍, 其中关于 OSPF 和 IS-IS 的讲述是很有启发作用的。这些书籍是由 Christian Huitema——IAB 的前任主席⁴, 还有 Radia Perlman——IS-IS 的首席设计师⁵编写的。最后, IETF 组织 (Internet 工程任务组) 采用了 OSPF 协议作为建议使用的 IGP 协议。技术上的不同的确会影响到决心, 但是有时, 这也会有行政上的因素。ISO 的标准化是一个缓慢的处理过程, 它一般需要 4 个步骤, 并且依赖多个委员会最终投票表决同意。而另一方面, IETF 组织却灵活快捷得多。IETF 组织在 1992 年有一个声明可以作为它们非正式的格言: “我们拒绝君主、总统和投票, 我们相信大致描述的多数人同意和运行的代码。”⁶可以看出, 通过 RFC 的程序, 发展 OSPF 协议要比采纳拘泥于形式化的 IS-IS 协议更有意义。

不考虑行政上的争议, OSPF 工作组实际上学习和利用了很多 IS-IS 设计中的基本机制。从表面看来, OSPF 协议和 IS-IS 协议有很多共同的特性:

- 它们都维护一个链路状态数据库, 并且这个数据库都是来自于一个基于 Dijkstra 的 SPF 算法计算的一棵最短路径树;
- 它们都利用 Hello 报文来形成和维护邻接关系;
- 它们都使用区域的概念来构成一个两级层次化的拓扑结构;
- 它们都具有在区域之间提供地址汇总的能力;
- 它们都是无类别路由选择协议;
- 它们都通过选取一个指定路由器来描述广播型网络;
- 它们都具有认证的能力。

¹ 国际标准化组织, “中间系统到中间系统域内路由选择信息交换协议 提供无连接模型网络服务 (ISO 8473)” ISO/IEC 10589, 1992 年。

² Ross Callon, “Use of OSI IS-IS for Routing in TCP/IP and Dual Environments,” RFC 1195, 1990 年 12 月。

³ 无连接模型的网络服务——CLNP 的网络层协议。

⁴ Christian Huitema, Routing in the Internet, Prentice Hall PTR, Englewood Cliffs, NJ, 1995 年。

⁵ Radia Perlman, Interconnections: Bridges and Routers, Addison-Wesley, Reading, MA, 1992 年。

⁶ Dave Clark, 引用于 Huitema 的第 23 页。

除了这些类似之处外，它们也有明显的不同。本章将通过检查它们的这些不同之处开始讲述。本书只把集成 IS-IS 协议（以下简称为 IS-IS 协议）作为一个 IP 路由选择协议来讲述，只有在使用 IS-IS 协议为 IP 协议路由选择时和 CLNS 协议有关的地方才讲述 CLNS 协议。

10.1 集成 IS-IS 协议的操作

ISO 组织经常使用不同的术语来描述 IETF 所描述的相同概念实体，这种情况有时会引起混淆。ISO 的术语将在本节介绍和定义，但是在一般情况下，本章将使用本书其余章节使用的更类似于 IETF 的术语¹。有一些 ISO 的术语是非常基本的，因此，在具体讲述 IS-IS 协议的所有术语之前先介绍一下这些术语。

一台路由器就是一个中间系统（Intermediate System, IS），而一台主机就是一台端系统（End System, ES）。因此，提供主机与路由器之间通信的协议称为 ES-IS 协议，而被路由器用来进行相互宣告的协议（路由选择协议）称为 IS-IS 协议（如图 10-1）。虽然 IP 协议使用路由器发现机制，例如 Proxy ARP 或 IRDP，或者在主机上配置简单的缺省网关，但是 CLNP 协议还使用 ES-IS 协议来形成端系统和中间系统之间的邻接关系。对于 IP 协议来说，ES-IS 协议和 IS-IS 协议没有什么相关之处，因此本书将不包括有关 ES-IS 协议的讨论。

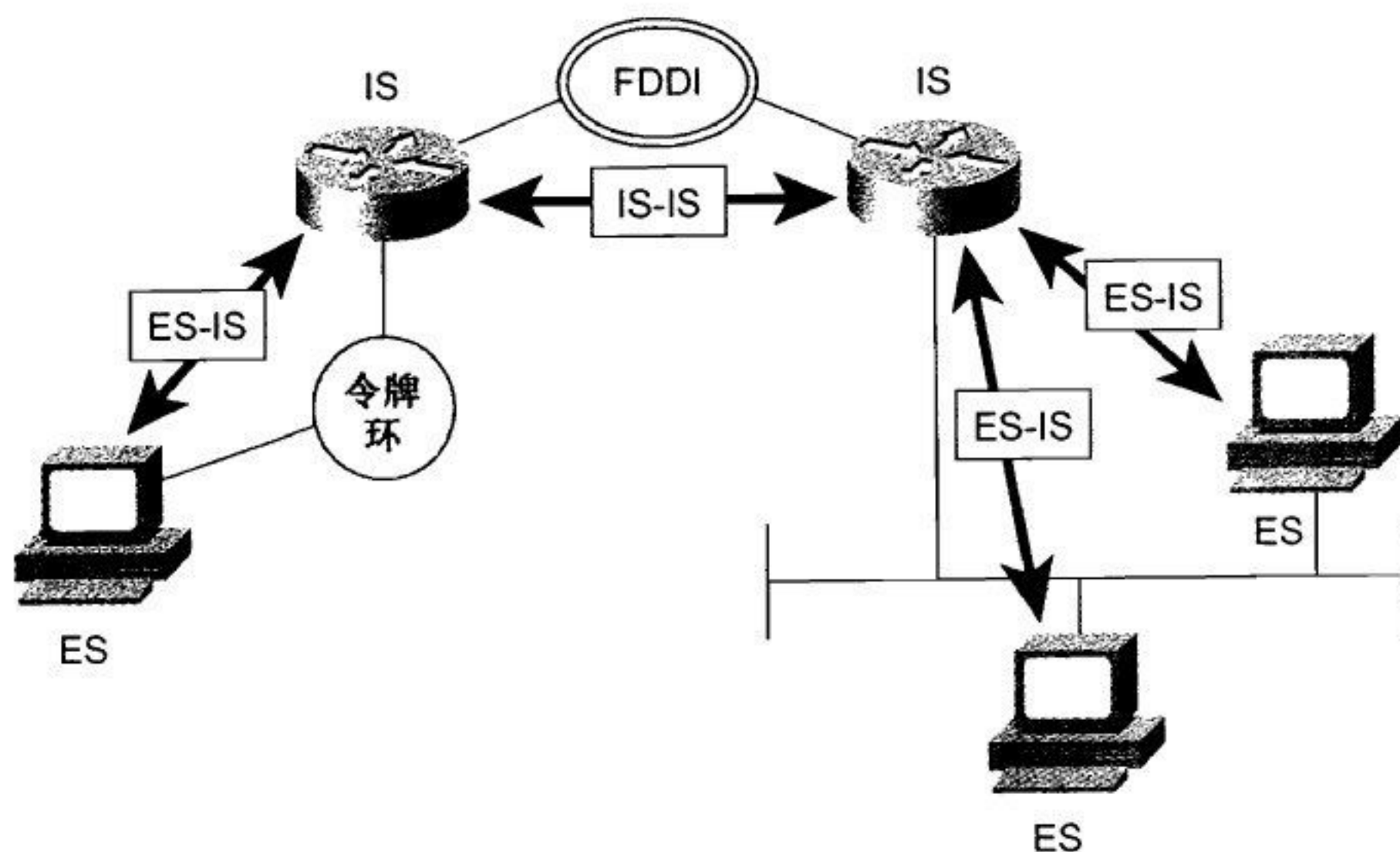


图 10-1 在 ISO 的术语里面，主机是端系统，而路由器是中间系统

与一个子网相连的接口称为子网连接点（Subnetwork Point of Attachment, SNPA）。SNPA 有一些概念化，因为它实际上是定义了一个提供子网服务的“点”，而不是一个实际的物理接口。SNPA 的基本概念特性和子网本身的基本概念特性是相符合的，它可以由数据链路交换机相连的多个数据链路组成。

从一个节点的 OSI 层到另一个节点对等的 OSI 层的数据单元称为协议数据单元（Protocol Data Unit, PDU）。因此，一个帧就是一个数据链路 PDU（DLPDU），而一个数据包（或者分组）就是一个网络层协议数据单元（NPDU）。执行与 OSPF 协议中的 LSA 等价功能的数据单

¹ 对于某些共同术语的 ISO/欧洲拼法，如“routeing”与“neighbour”是可以不用的。

元称为链路状态 PDU (LSP)¹。但与 LSA 不同, LSA 是封装在 OSPF 头部之后的, 并且都被封装在一个 IP 数据包内, 而一个 LSP 本身就是一个数据包。

10.1.1 IS-IS 区域

虽然 IS-IS 协议和 OSPF 协议都使用区域的概念来创建两级的层次化网络拓扑结构, 但是它们存在一个基本的不同之处, 就是这两种协议在定义区域的方法上不一样。正如图 10-2 所示, OSPF 协议的区域边界是通过路由器来划分的。某些接口属于一个区域, 而另一些接口属于其他区域。如果一台 OSPF 路由器具有的接口分布在多于一个的区域里, 那么这台路由器就是一个区域边界路由器, 即 ABR 路由器。

如图 10-3 所示, 这和图 10-2 显示的网络拓扑完全一样, 只是把它设计成了 IS-IS 区域。这里请注意, 所有的路由器都完全处在一个区域内部, 并且区域的边界是在链路上, 而不是在路由器上。与区域相连的路由器称为层 2 路由器 (Level 2 Router), 而那些与其他区域没有直接连接的路由器称为层 1 路由器 (Level 1 Router)。

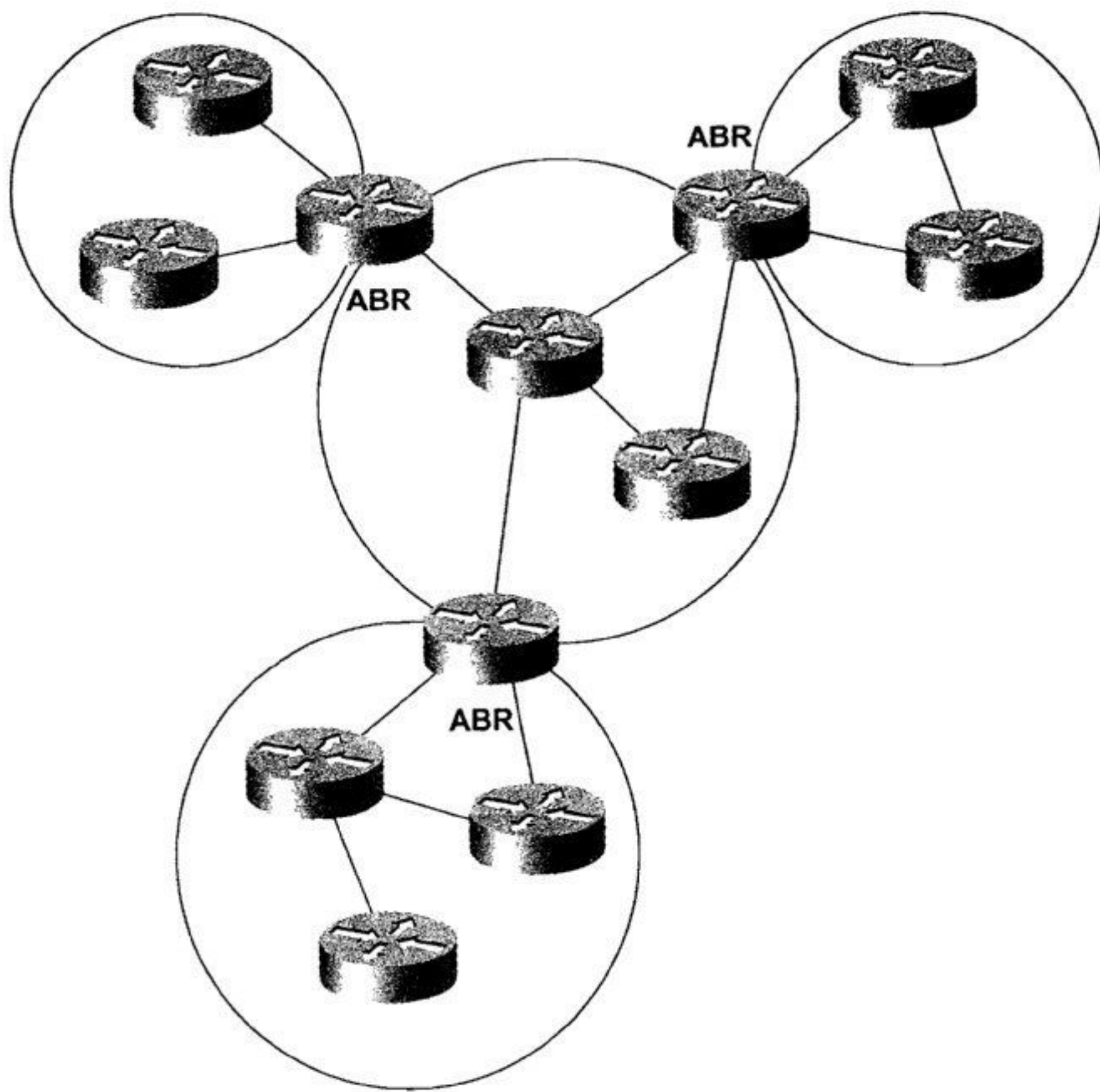


图 10-2 OSPF 区域边界是在路由器上, 而与不同区域相连的路由器是 ABR 路由器

一个中间系统可以是一个层 1 类型的路由器 (L1)、一个层 2 类型的路由器 (L2) 或者两种类型皆是的路由器 (L1/L2)。L1 路由器类似于 OSPF 协议中的非骨干内部路由器, 而 L2 路由器类似于 OSPF 协议中的骨干路由器, 同样地, L1/L2 路由器类似于 OSPF 协议中的

¹ 在某些文档中, 例如 RFC1195, 把 LSP 定义为一个链路状态数据包 (Link State Packet)。

ABR 路由器。在图 10-3 中, L1/L2 路由器和 L1 路由器以及 L2 路由器相连。这些 L1/L2 路由器必须同时维护一个 L1 的链路状态数据库和一个 L2 的链路状态数据库, 这种方式 and OSPF 协议中 ABR 路由器必须维护与之相连的每一个区域各自的数据库相类似。

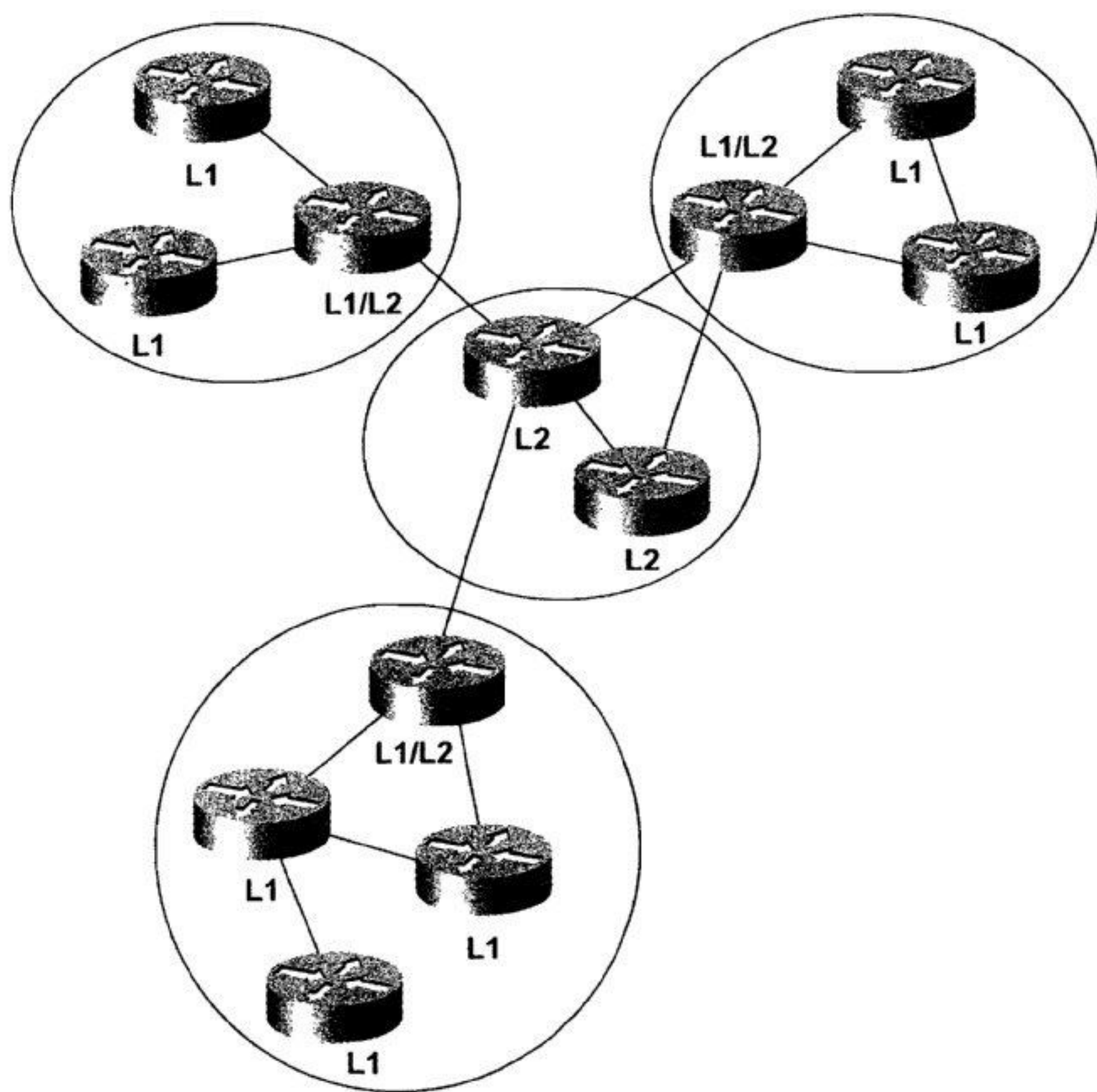


图 10-3 IS-IS 区域的边界在链路上, 而与区域相连的路由器是层 2 路由器

由 L2 路由器（包括 L1/L2 路由器）和它们之间的互连链路一起构成 IS-IS 的骨干；与 OSPF 协议相同，区域间的通信量都必须经过这个骨干。在一个区域内的每台 L1 路由器（包括区域内的 L1/L2 路由器）都会维护一个同样的链路状态数据库。但与 OSPF 协议中的 ABR 路由器不同，L1/L2 路由器不需要通告 L2 类型的路由给 L1 类型的路由器。因此，一台 L1 路由器无法知晓它自己所在区域之外的目的路由。在这个意义上，一台 L1 路由器就相当于 OSPF 协议中一台属于完全末梢区域内的路由器。为了路由转发数据包到其他的区域，L1 路由器必须转发数据包到一台 L1/L2 路由器上。当 L1/L2 路由器发送它的层 1LSP 进入一个区域时，它将通过在 LSP 中设置一个称为“区域关联位（Attached, ATT）”¹的二进制位来通知其他 L1 路由器它可以到达其他的区域。

回忆第 9 章“开放最短路径优先协议（OSPF）”，在那里 OSPF 协议是通过运行一个 SPF 算法来计算一个区域内的路由的，但是区域间的路由却是使用距离矢量算法来计算的。这种情况在 IS-IS 协议中并不完全一样。L1/L2 路由器将要分别维护一个层 1 类型的链路状态数据库和一个层 2 类型的链路状态数据库，并且使用不同的 SPF 树来反映层 1 和层 2 的拓扑结构。

ISO10589 描述了 IS-IS 协议路由器可以利用虚链路来修复被分段的区域，这和 OSPF 是

¹ 这里实际上是 4 个 ATT 位，并与不同的度量相关联。这些位的进一步解释在“IS-IS PDU 格式”一节中介绍。

一样的。但是这个特性在 Cisco 路由器和其他大多数厂商的路由器上都不支持，因而不在这里作进一步的描述。

由于一个 IS-IS 路由器可以完全地处于一个单一的区域内，因此区域 ID（或区域地址）将和整个路由器相关联，而不是和某一个接口相关联。IS-IS 协议有一个独特的特性，就是一台路由器可以最多具有 3 个区域地址，这在区域过渡期间是很有用的。在本章配置部分的案例研究“一个区域的迁移”中演示了多个区域地址的使用。每个 IS-IS 路由器还要有一种方法在它所在的路由选择域内唯一地标识它本身。这个唯一标识就是系统标识的功能，系统标识(System ID)类似于 OSPF 协议中的路由器 ID。在一台 IS-IS 路由器上可以通过一个单一的地址同时定义区域标识和系统标识，这个地址就是网络实体标题 (Network Entity Title, NET)。

10.1.2 网络实体标题

即使 IS-IS 协议只用来为 TCP/IP 协议进行路由选择时，它也依然是一个 ISO CLNP 协议。因此，IS-IS 协议对等体之间的通信数据报文是 CLNS PDU，也就是说即使是在一个纯 IP 环境中，一台 IS-IS 路由器也必须有一个 ISO 地址。这个 ISO 地址是一个网络地址，称为网络实体标题 (NET)，并在 ISO8348 中给以描述。¹一个 NET 地址的长度范围可以是 8~20 个 8bit 字节，并可以描述为区域标识和一个设备的系统标识两部分，如图 10-4 所示。

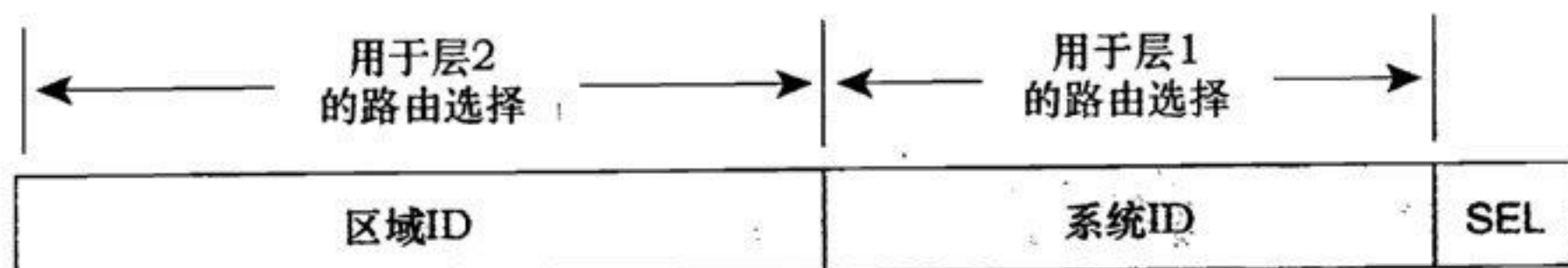


图 10-4 NET 地址指定了区域 ID 和一个 IS 或者 ES 的系统标识

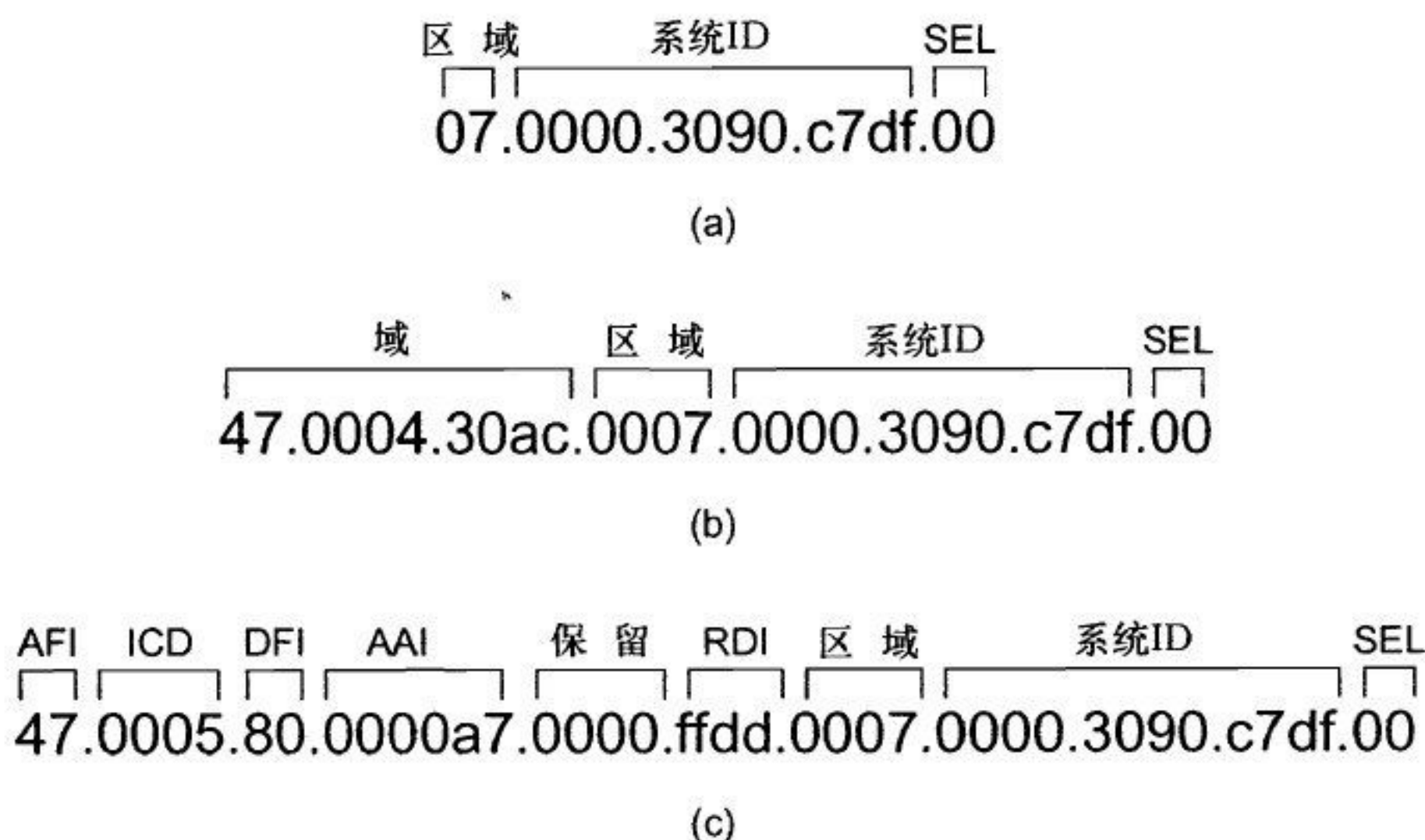
ISO 设计的 NET 地址可以在许多系统中做很多事情，这依赖于你个人的看法，要么认为这个地址的格式是非常灵活的和可扩展的，要么认为这个地址格式是个麻烦的容易搞糊涂的变量字段。如图 10-5 所示，图中仅仅显示了一个 ISO NET 地址可能具有的多种格式中的 3 种。虽然，在每一个例子中系统标识前面的域是不同的，但是系统标识本身都是相同的。ISO10589 指定了这个域的长度可以从 1~8 个 8bit 字节，但是在一个路由选择域内的所有节点的系统标识必须使用相同的长度。最普遍的情况是，这个系统标识的长度是 6 个 8bit 字节²，并且通常是这台设备上的某个接口的 MAC 地址 (Media Access Control, 介质访问控制)。对于路由选择域内的每一个节点，这个系统标识必须是惟一的。

在图 10-5 中的例子中还有一个需要注意的地方，就是 NSAP 选择符 (SEL)。在所有的情况下，这个 1 个 8bit 字节的字段都被设置为 0x00。一个网络服务接入点 (NSAP) 所描述的都和某个节点在网络层上的一种特有服务相关联。因而，在一个 ISO 地址中，SEL 设置为大于 0x00 的某些值时，这个地址就是一个 NSAP 地址。这种情况和一个 IP 数据包内的 IP 目的地址与 IP 协议号的组合有些类似，它表明一个具体设备的 TCP/IP 协议栈的网络层上的一个具体服务。而在一个 ISO 地址的 SEL 设置为 0x00 时，这个地址就是一个 NET 地址，指明

¹ 国际标准化组织，“Network Service Definition Addendum 2: Network Layer Addressing”，ISO/IEC 8348/Add.2, 1988 年。

² Cisco 公司的 IS-IS 实现需要一个 6 个 8bit 字节的系统标识。

了某个节点网络层本身的地址。



AFI: Authority and Format Identifier
 ICD: International Code Designator
 DFI: Domain Specific Part (DSP) Format Identifier
 AAI: Administrative Authority Identifier
 RDI: Routing Domain Identifier (Autonomous System Number)
 SEL: Network Service Access Point (NSAP) Selector

图 10-5 3 种 NET 格式：一个是简单的 8 个 8bit 字节区域 ID/系统标识格式 (a)；

一个是 OSI NSAP 格式 (b)；还有一个是 GOSIP NSAP 格式¹ (c)

图 10-5 中只是显示了 NET 的多种格式，但关于 NET 配置的更详细的讨论已经超出了本书的讨论范围。如需要进一步的学习，RFC1237 是一个不错的参考。²在大多数的情况下，集成 IS-IS 会运行在一个 CLNP/IP 的混和环境中，因而，NET 地址的设定将基于 CLNP 的需要。在一个只有 IP 的环境中，NET 地址的设定可以基于某个标准，例如 GOSIP。如果你可以在一个纯 IP 环境中自由地选择任何 NET 地址的格式，那么将可以根据实际网络的需要选择最简单的格式。

无论是何种格式的地址，都需要满足下面的 3 个规则：

- NET 地址必须以一个单个 8bit 字节的域开始（例如，47.xxxx...）；
- NET 地址必须以一个单个 8bit 字节的域结束，并且应该设置为 0x00 (...xxxx.00)。如果 SEL 是非零的，IS-IS 也会起作用，但是在一个 CLNP/IP 混和的路由器可能会出现一些问题；
- 在 Cisco 的路由器上，NET 地址的系统标识必须是 6 个 8bit 字节；

10.1.3 IS-IS 的功能结构

像 ISO 模型一样，之所以有一个分层的网络体系结构，其中有一个最主要的目的是为了

¹ GOSIP Advanced Requirements Group (GOSIP 高级需求组), "Government Open Systems Interconnection Profile (GOSIP) Version 2.0 [Final Text]", Federal Information Processing Standard (联邦信息处理标准), U.S. Department of Commerce (美国商务部), National Institute of Standards and Technology (美国国家标准和技术协会), 1990 年 10 月。

² Richard Colella, Ella Gardner 和 Ross Callon, "Guidelines for OSI NSAP Allocation in the Internet", RFC 1237, 1991 年 7 月。

使每一层的功能都独立于它下面的一层。例如,网络层必须适应大多数类型的数据链路或子网络。为了进一步满足这种适应性,网络层又由两个子层组成(如图 10-6 所示)。子网独立子层(Subnetwork Independent Sublayer)为传输层提供了一致的和统一的网络服务。而子网依赖子层(Subnetwork Dependent Sublayer)则为子网独立子层的需求而去存取数据链路层提供的服务。正如这两个命名所暗示的,子网依赖子层依赖于数据链路的具体类型,而子网独立子层则能够独立于数据链路。

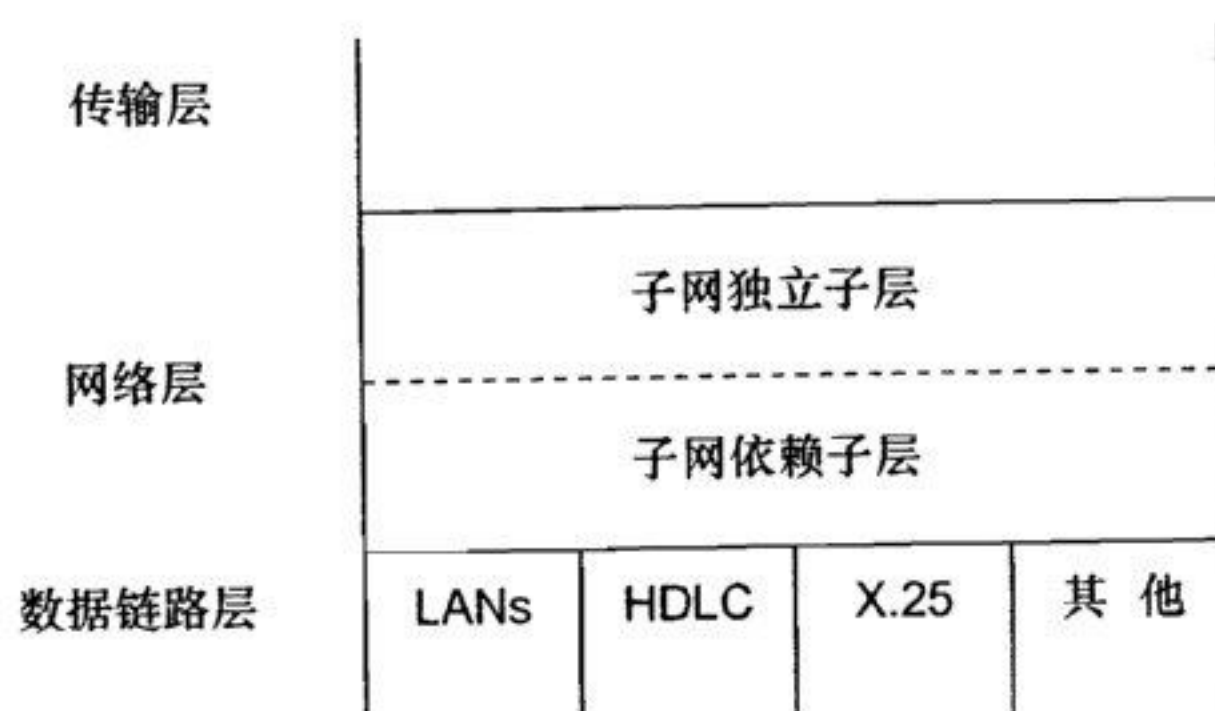


图 10-6 OSI 网络层由两个子层组成

网络层的结构是在 ISO8648¹中指定的,它实际上要比图 10-6 中显示的结构更为复杂。在这里之所以提到这两个基本的子层,是因为在 ISO10589 中对这些子层的功能架构里面有关 IS-IS 的操作做了大量描述。

1. 依赖于子网的功能

依赖于子网的功能当然是指子网的功能依赖于它的下层。它们的功能是为子网独立子层的功能层面隐藏掉不同种类的数据链路(子网)的特征。下面的依赖于子网的功能对于路由选择是非常重要的:

- PDU 报文的传送和接收是在一个具体相连的子网上;
- 通过 IS-IS 的 Hello PDU 报文来发现邻居路由器并建立这个子网上的邻接关系;
- 邻接关系的维护;
- 链路的复用,或者说对于 OSI 协议处理转换为 OSI PDU 报文,而对于 IP 处理转换为 IP 报文;

(1) IS-IS 网络类型

相对于 OSPF 协议中定义的 4 种网络类型,IS-IS 协议只定义了两种类型:广播型子网和点到点或一般拓扑子网。广播型子网的定义是和 OSPF 协议中的定义一样的——就是支持多路广播的多路访问数据链路。而点到点子网(非广播子网)则可能是永久链路,例如 T1 链路,或者动态链接链路,例如 X.25 的 SVC 链路。

(2) 邻居路由器和邻接关系

IS-IS 路由器是通过交换 IS-IS Hello PDU 报文信息来发现邻居并形成邻接关系的。Hello 报文每隔 10 秒传送一次,在 Cisco 的路由器中,这个时间间隔可以基于每个接口通过命令 `isis`

¹ 国际标准化组织,“Internal Organisation of the Network Layer”, ISO 8648, 1990 年。

hello-interval 来改变。虽然 IS-IS Hello 报文对于广播型子网和点到点子网有些小的差别，但是 Hello 报文中包含的基本信息还是相同的，这将在“IS-IS PDU 格式”一节中讲述。一台 IS-IS 路由器使用它的 Hello PDU 报文可以标识它本身和它的性能 (capabilities)，以及描述发送这个 Hello 报文的接口的一些参数。如果两台邻居路由器关于它们各自的性能和接口的参数协商一致，那么它们就形成了邻接关系。

对于层 1 类型与层 2 类型的邻居路由器，IS-IS 协议可以形成不同的邻接关系。L1 路由器可以和 L1 以及 L1/L2 邻居形成 L1 邻接关系，而 L2 路由器可以和 L2 以及 L1/L2 邻居形成 L2 邻接关系。L1/L2 路由器和它的邻居既可能形成 L1 邻接关系也可能形成 L2 邻接关系。但是，一台 L1 路由器和一台 L2 路由器不能形成一个邻接关系。

一旦邻接关系建立成功，Hello 报文将担当保活 (keepalive) 的功能。每一台路由器都在它的 Hello 报文中发送一个抑制时间 (Hold Time)，用来通知它的邻居路由器在宣告这台路由器无效之前，应该等待多长的时间去侦听下一个 Hello 报文。在 Cisco 的路由器上，缺省的抑制时间是 Hello 时间间隔的 3 倍长，并且可以基于每一个接口通过命令 **isis hello-multiplier** 来改变。

如图 10-7 所示，IS-IS 的邻居表可以通过命令 **show clns is-neighbors** 来查看。图中显示的开始 4 列表明了每一台邻居路由器的系统标识、和邻居相连的本地接口、邻接关系的状态以及邻接关系的类型。这里的状态要么是 Init——表明邻居路由器是学习到了但是还没有形成邻接关系，要么是 Up——表明和邻居路由器成功建立邻接关系。优先级是指在广播型网络上用来选举指定路由器的路由器优先级，这将在下一节介绍。

```
Brussels#show clns is-neighbors
```

System Id	Interface	State	Type	Priority	Circuit Id	Format
0000.0C04.DCC0	Se0	Up	L1	0	06	Phase V
0000.0C04.DCC0	Et1	Up	L1	64	0000.0C76.5B7C.03	Phase V
0000.0C0A.2C51	Et0	Up	L2	64	0000.0C76.5B7C.02	Phase V
0000.0C0A.2AA9	Et0	Up	L1L2	64/64	0000.0C76.5B7C.02	Phase V

```
Brussels#
```

图 10-7 IS-IS 的邻居表可以通过命令 **show clns is-neighbors** 来显示

第 6 列显示的是电路 ID (Circuit ID)，这是一个 1 个 8bit 字节的数字，路由器用它来唯一地标识这个 IS-IS 接口。如果这个接口是和一个广播型多址网络相连的，那么这个电路 ID 是和该网络上的指定路由器的系统标识相连的，并把这个完全的数字称为 LAN ID。在这种用法中，更为正确的叫法应该把电路 ID 称为伪节点 ID (Pseudonode ID)。例如，在图 10-7 中，与接口 E0 相连的链路的 LAN ID 是 0000.0c76.5b7c.02。在这里，指定路由器的系统标识是 0000.0c76.5b7c，而伪节点 ID 是 02。

最后一列指出了邻接关系的格式。对于集成的 IS-IS 协议，这个格式将永远是 Phase V，用来说明是 OSI/DECnet Phase V。另外一个惟一的格式是 DECnet Phase IV。

(3) 指定路由器

IS-IS 协议在一个广播型多址网络上选取指定路由器 (更为准确地说，是一个指定 IS) 的原因同 OSPF 协议的一样。把网络本身看作是一台路由器或一个伪节点，要比局域网内的每一台路由器都要与该网络上相连的其余每台路由器形成一个邻接关系的方法好得多。包括指定路由器在内的每一台路由器都只需要通告单条链接到伪节点。指定路由器作为伪节点

的代表也会通告一条链接到与之相连的所有路由器。

然而, 与 OSPF 协议不同的是, 与广播型多址网络相连的 IS-IS 路由器要和网络上它的所有邻居建立邻接关系, 而不仅仅是指定路由器。每一台路由器将以组播方式发送它的 LSP 报文给它所有的邻居路由器, 并且指定路由器使用一个称为序列号 PDU (Sequence Number PDU, SNP) 的报文来确保 LSP 的泛洪是可靠的。这个可靠的泛洪过程和 SNP 报文将在后面的“更新过程”一节中介绍。

IS-IS 协议的指定路由器的选取过程非常简单。每一个 IS-IS 路由器接口都被指定一个 L1 类型的优先级和 L2 类型的优先级, 它们的范围是 0~127。Cisco 路由器接口的优先级对于 L1 和 L2 类型的缺省值都是 64, 并且可以通过命令 **isis priority** 来改变这个值。

路由器通过它的每一个接口发送出 Hello 报文, 并在 Hello 报文中通告它的优先级——在 L1 类型的 Hello 报文中通告 L1 类型的优先级, 在 L2 类型的 Hello 报文中通告 L2 类型的优先级。如果优先级是 0, 那么路由器将没有资格成为一个指定路由器。对于非广播型网络上的接口, 不需要选举指定路由器, 因此也将它们的优先级设置为 0 (注意一下图 10-7 中显示的串行接口的优先级)。拥有最高优先级的路由器将成为指定路由器。如果路由器的优先级相同, 那么在数值上具有最高的系统标识的路由器将成为一个指定路由器。

与 L1 和 L2 类型的优先级相对应的是, 需要在一个网络上为 L1 和 L2 分别选取单独的指定路由器。这种做法是必需的, 因为在一个单一的局域网中存在着各自不同的 L1 和 L2 类型的邻接关系, 如图 10-8 所示。由于一个接口对于每一个层具有单独不同的优先级, 因此在同一个局域网上的 L1 类型的 DR 路由器和 L2 类型的 DR 路由器可能是同一台路由器, 也可能不是。

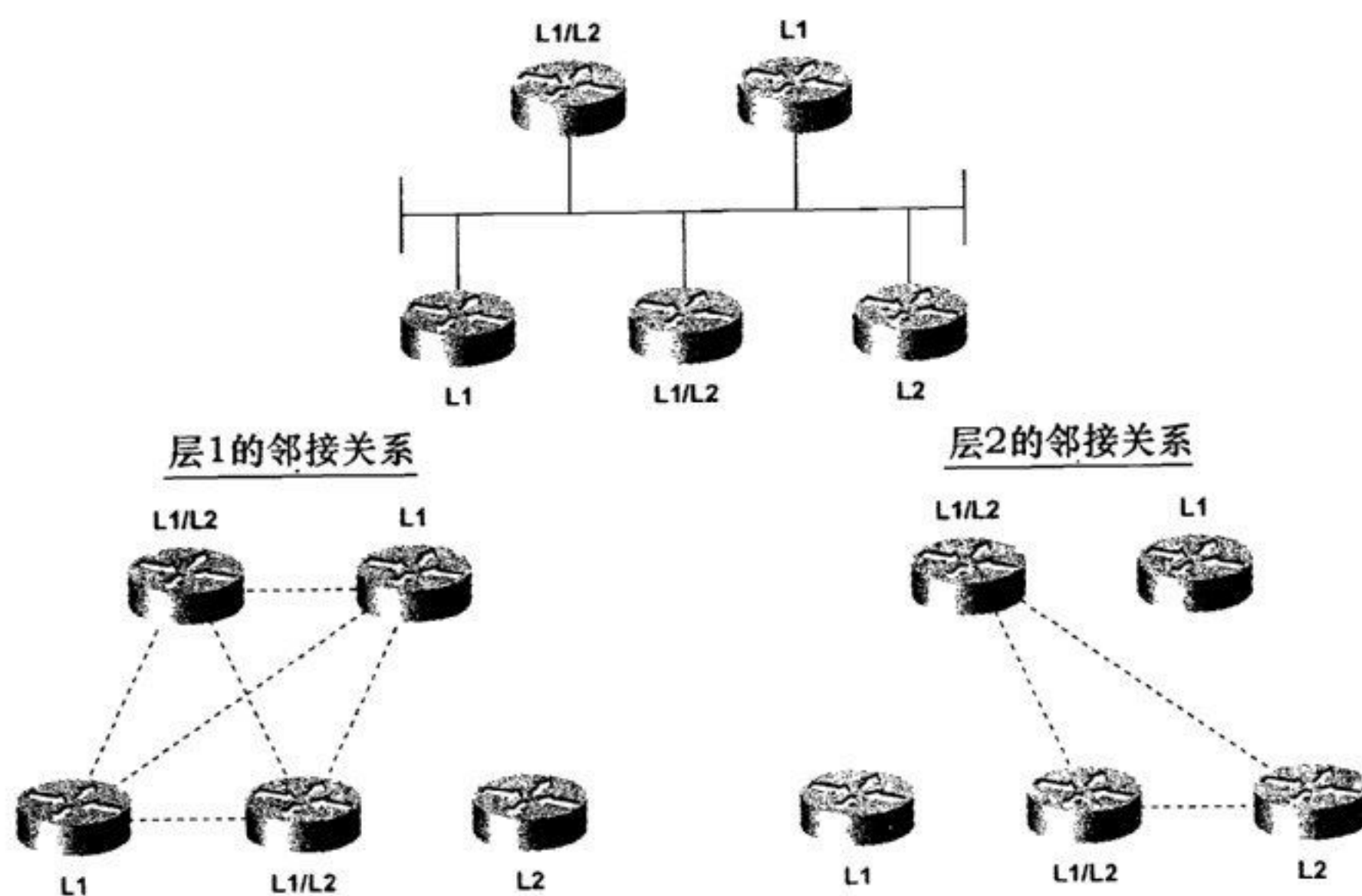


图 10-8 对于层 1 和层 2 建立的邻接关系是不同的, 因而也就必须为层 1 和层 2 选取各自的指定路由器

指定路由器分配了所在网络的 LAN ID。正如前面章节所讨论的, LAN ID 是由该网络上的指定路由器的系统标识和它的伪节点 ID 连在一起得到的。该网络上其他所有的路由器都将使用指定路由器分配的这个 LAN ID。

如图 10-9 所示, 图中显示了一台路由器的 E0 接口的邻居表, 这个 E0 接口是和图 10-7 中显示的路由器的 E0 接口连接在同一个网络上的。通过比较这两个邻居表, 可以发现总共

有 3 台路由器连接在这个以太网上：0000.0c0a.2aa9、0000.0c0a.2c51 和 0000.0c76.5b7c。由于它们所有的优先级都是 64，因此在数值上具有最高的系统标识的路由器将成为指定路由器，也就是路由器 0000.0c76.5b7c，并且它在这里利用设置为 2 的电路 ID 来标识这个网络。因此，图 10-7 和图 10-9 中显示的 LAN ID 都是 0000.0c76.5b7c.02。

London#show clns is-neighbors

System Id	Interface	State	Type	Priority	Circuit Id	Format
0000.0C76.5B7C	Et0	Up	L2	64	0000.0C76.5B7C.02	Phase V
0000.0C0A.2AA9	Et0	Up	L2	64	0000.0C76.5B7C.02	Phase V
0000.3090.6756	Se0	Up	L1	0	02	Phase V

London#

图 10-9 这个路由器的 E0 接口是和图 10-7 中的路由器的 E0 接口连接在同一个网络上的

那个附加在系统标识后面的电路 ID 是必需的，因为同一台路由器可以是多个网络的指定路由器。注意在图 10-7 中，与那台路由器的 E0 接口和 E1 接口相连的两个网络的指定路由器都是同一台路由器。与 E0 接口相连的网络的电路 ID 设置为 02，而与 E1 接口相连的网络的电路 ID 设置为 03，这样就可以使每一个网络的 LAN ID 保持惟一性。

IS-IS 协议的指定路由器处理过程相比 OSPF 协议的指定路由器处理来说，有两个方面是十分粗糙的（或者说是不够复杂的，这个观点因人而异）。首先，IS-IS 协议不选取备份指定路由器。如果 IS-IS 的指定路由器失效了，那么将选取一个新的指定路由器。其次，IS-IS 的指定路由器相对 OSPF 的指定路由器来说是不稳定的。如果一个 OSPF 路由器在一个已经存在指定路由器的网络上变成活动的了，即使它的优先级或路由器 ID 更高，新的路由器也不会成为一个指定路由器。结果，OSPF 的指定路由器通常是网络上处于活动状态很长的路由器。与 OSPF 的规则相比，如果一个新的 IS-IS 路由器具有比现有指定路由器更高的优先级，或者优先级相同但是具有更高的系统标识，那么这个新的 IS-IS 路由器将成为新的指定路由器。这样，每次指定路由器的更改都必须有一组新的 LSP 报文进行泛洪。

2. 独立于子网的功能

子网独立子层的功能定义了 CLNS 怎样分发报文通过整个 CLNP 的互联网络和怎样把这些服务提供给上一层传输层。路由选择功能本身又被分成 4 个处理过程：更新处理过程、决策处理过程、转发处理过程和接收处理过程。正如后面两个处理过程的名称所暗示的，转发处理过程（Forwarding process）的职责是传送 PDU 报文，而接收处理过程（Receive process）的职责是接收 PDU 报文。这两个处理过程是在 ISO10589 中描述的，并且相比 IP 报文来说，它们和 CLNS NPDU 报文的关系更密切，因此不再做进一步的讲述。

(1) 更新处理过程

更新处理过程（Update Process）的职责是构建 L1 和 L2 的链路状态数据库。为了做到这一点，L1 的 LSP 将在整个区域内进行泛洪，而 L2 的 LSP 将会在所有 L2 的邻接上进行泛洪。关于 LSP 报文的具体字段的详细描述将在后面的“IS-IS PDU 格式”一节中讲述。

每一个 LSP 报文都包含一个剩余生存时间、一个序列号和一个校验和。剩余生存时间（Remaining Lifetime）是一个老化时间或使用期限（Age）。IS-IS LSP 报文的剩余生存时间和 OSPF 协议中 LSA 报文的老化时间参数的一个不同是：LSA 报文的老化时间是从 0 到最大生存时间依次递增，而 LSP 报文的剩余生存时间则是从最大生存时间开始，并递减到 0。在这

里, IS-IS 的最大生存时间 (MaxAge) 是 1200s (20min)。像 OSPF 协议一样, 当 LSP 驻留在路由器的链路状态数据库中时, IS-IS 会随着时间的推移老化每一个 LSP, 即递减它的剩余生存时间。并且, 始发路由器必须周期性地刷新它的 LSP 以防止它的剩余生存时间减小到 0。IS-IS 的刷新时间间隔是 15min 减去一个最大不超过 25% 的随机抖动变量。如果剩余生存时间减小到 0 了, 那么这个过期的 LSP 将还会在路由器的链路状态数据库当中保留 60s 的时间, 这个时间称为“零老化生存时间” (ZeroAgeLifetime)。

如果一台路由器收到了一个带有错误校验和的 LSP, 那么这台路由器可以通过设置 LSP 的剩余生存时间为 0 来清除这条 LSP, 并进行重新泛洪。清除的行为将会引起始发这条 LSP 的路由器发送一个新的关于这条 LSP 的实例 (new instance)。这个处理过程也是 IS-IS 协议和 OSPF 协议相比另一个不同之处, 因为在 OSPF 协议里始发路由器仅仅清除这个 LSA。

在一个可能出现错误的子网上, 允许接收路由器启动清除 LSP 的功能会显著地增加 LSP 的流量 (也就是说, 会引起接收路由器不断地清除 LSP, 并且始发路由器会不断地重发新的 LSP 实例)。为了忽略这个行为, 可以在 IS-IS 协议的路由选择配置里增加一条命令 **ignore-lsp-errors**。当一台启动了这个参数选项的路由器收到一条被破坏的 LSP 时, 它就会忽略它而不是清除它。但是, 这条被破坏的 LSP 的始发路由器仍然会利用 SNP 了解到这条 LSP 没有被收到。SNP 将在本节后面的部分讲述。

序列号是一个 32 位的无符号线性数字。当一台路由器开始始发一条 LSP 时, 它将使用设置为 1 的序列号, 并且这条 LSP 的每一个后续实例的序列号都会递增 1。如果序列号递增达到了最大值 (0xFFFFFFFF), 那么这个 IS-IS 进程必须失效至少 21min (最大生存时间 + 零老化生存时间), 以便允许使这条旧的 LSP 从所有的链路状态数据库中清除掉。

在一个点到点的子网上, 路由器将直接发送 L1 和 L2 的 LSP 给它们的邻居路由器。在一个广播型的子网上, LSP 将以组播的方式发送到它所有的邻居路由器。运载 L1 LSP 的帧会有一个 0180.c200.0014 的目的 MAC 标识, 称为 A11L1ISs。运载 L2 LSP 的帧会有一个 0180.c200.0015 的目的 MAC 标识, 称为 A11L2ISs。

IS-IS 协议使用序列号报文 (SNP) 来了解 LSP 的接收情况和维护链路状态数据库的同步情况。在这里有两种类型的序列号报文: 部分序列号报文 (PSNP) 和完全序列号报文 (CSNP)。在一个点到点的子网上, 路由器使用 PSNP 报文来明确地确认每一个 LSP 报文是否收到¹。PSNP 报文是通过下面的信息来描述正在被确认的 LSP:

- LSP 标识 (LSP ID);
- LSP 的序列号;
- LSP 的校验和;
- LSP 的剩余生存时间。

当一台路由器在一个点到点的子网上发送一条 LSP 时, 它会设置一个周期为 `minimumLSPTransmissionInterval` 的计时器。如果该计时器超时了, 路由器还没有收到一个关于确认收到这条 LSP 的 PSNP 报文, 那么将会发送一个新的 LSP 报文。在 Cisco 的路由器上, `minimumLSPTransmissionInterval` 的缺省值是 5s, 并且可以基于每个接口使用命令 **isis retransmit-interval** 来更改。

在一个广播型的子网上, LSP 不需要每一台接收它的路由器确认。作为替代, 指定路由

¹ 有一个例外是, 路由器收到了一条比它链路状态数据库中的相同 LSP 的实例更旧的 LSP 实例。在这种情况下, 该路由器将回复一个新的 LSP。

器将会周期性地以组播方式发送 CSNP 报文，用来描述链路状态数据库中的每一个 LSP。发送 CSNP 的周期缺省的是 10s，并且可以通过命令 **isis csnp-interval** 来更改。L1 CSNP 以组播方式发送到 AllL1ISs (0180.c200.0014)，而 L2 CSNP 以组播方式发送到 AllL2ISs (0180.c200.0015)。

在一台路由器收到一个 CSNP 报文时，它会把这个 PDU 报文中的 LSP 摘要与自己数据库中的 LSP 进行比较。如果发现在该路由器的数据库中存在 CSNP 报文中没有的 LSP，或者比 CSNP 报文中更新的 LSP 实例，那么该路由器将以组播方式在网络上发送这条 LSP。但是，如果其他的路由器开始发送更新的 LSP，那么该路由器将不会发送相同 LSP 的另一个拷贝。如果路由器的数据库中没有包含 CSNP 报文中列出的所有 LSP，或者数据库中包含的是某条 LSP 的旧实例，那么这台路由器将会以组播方式发送一个 PSNP 报文，这个报文中列出该路由器所需要的所有 LSP。虽然 PSNP 报文是以组播方式发送的，但是只有指定路由器才会使用包含相应 LSP 的报文来响应。

IS-IS 具有一个有趣的特性，如果运行它的设备因内存不足而不能记录完整的链路状态数据库时，它具有通知其他路由器的能力。导致内存溢出或超载的原因可能是因为该路由器所在的区域变得过于庞大，路由器的内存不足，或者一些瞬间情况——像指定路由器失效等。像在上面这些情况下，路由器如果不能完整地存储链路状态数据库，那么它将会在它发送的 LSP 报文中设置一个称为超载 (Overload, OL) 的位。

OL 位用来指示路由器可能不能再进行正确的路由选择决策了，因为它的链路状态数据库已经不再完整。其他的路由器将仍然会转发数据包到这台超载路由器的直连网络上，但是，在这台超载的路由器发送一个清除 OL 位的 LSP 报文之前，其他路由器不会再利用这台路由器转发经过它传送的数据流了。因为设置 OL 位可以避免超载的路由器被用作一条路由的下一跳，因此这个位又经常被称做 hippity 位 (或称为 hippity-hop，随个人习惯叫法)。

一般来说，路由器的内存应该平等地分配给 L1 和 L2 的数据库，但是，路由器能够在其中一个层的内存超载时，而保持其他层的内存处于正常状态。如果希望设置一台 IS-IS 路由器仅仅作为端节点的话，那么可以通过命令 **set-overload-bit** 来手工设置 OL 位。

使用命令 **show isis database** 可以显示一个 IS-IS 链路状态数据库的摘要，如图 10-10 所示。在这个图示中路由器 Brussels 是一台 L1/L2 路由器，因此它包含了 L1 和 L2 的数据库。在第一列显示的 LSP ID 是由始发路由器的系统标识连接了 2 个 8bit 字节构成的。在这里，跟在系统标识后的第一个 8bit 字节是伪节点标识。如果这个 8bit 字节是非零的，LSP 则是由一台 DR 路由器始发的。这时，系统标识和非零的伪节点标识一起构成了一个广播型子网的 LAN ID。

LSP ID 的最后一个 8bit 字节是 LSP 的编号。有时候一个 LSP 可能会很大，以至于超出了路由器缓冲区或数据链路所支持的 MTU 大小。在这种情况下，LSP 将被分段传送——也就是说，LSP 的信息可以在多个 LSP 报文中传送。这些 LSP 报文的 LSP 标识也就由 3 部分组成：相同的系统标识和伪节点 ID，还有不同的 LSP 编号。

LSP ID 后面紧跟的星号表示这条 LSP 报文是始发于正在查看的数据库所在的路由器。例如，在图 10-10 中显示的数据库是来自于路由器 Brussels 的。因此，LSP 标识为 0000.0c76.5b7c.00-00 的 L1 LSP 是始发于路由器 Brussels 的。

数据库的第 2 列和第 3 列显示了每一个 LSP 的序列号和校验和。第 4 列显示的是 LSP

抑制时间,也就是 LSP 的剩余生存时间,以秒数计。如果连续不断地重复敲入 **show isis database** 命令,就会发现这个数字在不断减小。当重新刷新一条 LSP 时,LSP 的剩余生存时间将被重新设置为 1200s,并将序列号递加 1。

```
Brussels#show isis database
IS-IS Level-1 Link State Database
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C04.DCC0.00-00  0x00000036  0x78AE       1152          0/0/0
0000.0C0A.2AA9.00-00  0x0000011B  0x057B       416           0/0/0
0000.0C76.5B7C.00-00* 0x00000150  0xD5D4       961           1/0/0
0000.0C76.5B7C.02-00* 0x00000119  0xD9C3       407           0/0/0
0000.0C76.5B7C.03-00* 0x000000FA  0x896E       847           0/0/0

IS-IS Level-2 Link State Database
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2AA9.00-00  0x0000013E  0x319A       666           0/0/0
0000.0C0A.2C51.00-00  0x00000133  0x762D       654           0/0/0
0000.0C76.5B7C.00-00* 0x0000014C  0x4E91       886           0/0/0
0000.0C76.5B7C.02-00* 0x0000011F  0x3CC3       1174          0/0/0
0000.3090.C7DF.00-00  0x0000011A  0xDDF0       858           0/0/0
Brussels#
```

图 10-10 IS-IS 的数据库可以通过命令 **show isis database** 来查看

最后一列指明了每一个 LSP 的区域关联位 (ATT 位)、区域分段位 (Partition, P 位) 和超载位 (OL 位)。L2 和 L1/L2 路由器设置 ATT 位为 1 来指明它们含有到达其他区域的路由。P 位指出始发路由器具有支持区域分段修复的能力。Cisco 公司 (和大多数其他厂商) 并不支持这个功能,因此该位总是设置为 0。OL 位设置为 1 说明始发路由器正处于内存超载状态,而这时链路状态数据库是不完整的。

如图 10-11 所示,使用带 level 和 LSP 标识参数的 **show isis database detail** 命令可以查看一个 LSP 的完整信息。LSP 中每一个单独字段的具体含义参见“IS-IS PDU 格式”一节。

(2) 决策处理过程

一旦更新处理过程建立了链路状态数据库,决策处理过程就将使用数据库中的这些信息去计算一个最短路径树。接着,这个处理过程使用生成的最短路径树去构建一个转发数据库 (路由选择表)。对于 L1 路由和 L2 路由,路由器将会执行不同的 SPF 计算。

```
London#show isis database detail level-2 0000.0C0A.2C51.00-00

IS-IS Level-2 LSP 0000.0C0A.2C51.00-00
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2C51.00-00* 0x0000013B  0x6635       815           0/0/0
  Area Address: 47.0001
  NLPID:         0x81 0xCC
  IP Address:   10.1.3.2
  Metric: 10 IS 0000.0C76.5B7C.02
  Metric: 10 IP 10.1.3.0 255.255.255.0
  Metric: 20 IP 10.1.2.0 255.255.255.0
  Metric: 10 IP 10.1.255.4 255.255.255.252
  Metric: 20 IP 10.1.255.0 255.255.255.0
  Metric: 30 IP 10.1.255.8 255.255.255.252
London#
```

图 10-11 使用命令 **show isis database detail** 可以查看数据库中 LSP 的完整信息

ISO10589 规定 IS-IS 协议使用下面的度量（一项是必须的，三项是可选的）来计算最短路径：

- 缺省度量（**Default**）——这是每一台 IS-IS 路由器都必须支持和理解的度量；
- 时延度量（**Delay**）——这是一个可选项，反映一个子网的传输时延；
- 代价度量（**Expense**）——这是一个可选项，反映一个子网的成本代价；
- 差错度量（**Error**）——这是一个可选项，反映一个子网的出错概率，类似于 IGRP/EIGRP 协议中的可靠度量。

每一种度量都使用一个范围在 0~63 之间的整数表示，并且每个路由都要为每种度量进行单独地计算。因此，如果一个系统同时支持这 4 种度量类型，那么路由器必须为 L1 的路由和 L2 的路由各运行 4 次 SPF 计算。由于对于每一个目的路由都可能需要进行多次反复地计算，结果会产生多个不同的路由选择表，而且因为可选的度量是用来支持根本没有发展起来的服务类型（TOS）的路由选择使用的，因而 Cisco 公司只支持缺省度量。

在 Cisco 的路由器上，不论接口的类型如何，都会指定每一个接口的缺省度量为 10。使用命令 **isis metric** 可以修改这个缺省度量的值，而且可以分别为层 1 和层 2 的接口修改它们的缺省值。如果对于每一个接口都保留使用它的缺省度量 10，那么每个子网的度量都可以被认为是等价的，并且每个子网的 IS-IS 度量可以看作是一个简单的跳数，其中每一跳的代价为 10。

这种情况下，一条路由的总代价就可以看作是沿此路由路径方向的每一个出站接口的单独度量简单相加。对于任何一条路由，IS-IS 最大的度量值是 1023。这个比较小的最大度量值经常被认为是 IS-IS 协议的一个限制，因为在一个大型的互联网络上它的度量尺度显得有点小了。但是，在批评这个限制的同时，我们也可以看到它的另一方面好处，就是 1023 的度量限制使 SPF 算法变得更有效率了。

IS-IS 协议的路由不仅分 L1 路由和 L2 路由，而且分内部路由和外部路由。内部路由是指到达 IS-IS 路由选择域内的目的地的路由，而外部路由是指到达 IS-IS 路由选择域外的目的地的路由。虽然 L2 路由可能是内部路由，也可能是外部路由，但 L1 路由却总是内部路由。

如果到达某个具体的目的地存在多条可能的路由，那么 L1 的路由将优先于 L2 的路由。在同一种 level 的多条路由中，支持可选度量的路由要优先于只支持缺省度量的路由（再次提示，Cisco 的路由器仅仅支持缺省度量，因此第二个优先顺序的排序和 Cisco 的路由器不相关）。在每一种 level 所支持的度量中，具有最低度量的路由优先。如果经过这个决策处理过后发现多条路径在同一层里是等价的，那么它们都会被放入路由选择表中。在 Cisco 公司的 IS-IS 协议的实现中将执行等价代价的负载均衡，并且最大支持 6 条等价负载均衡的路径。

在前面一节“更新处理过程”中谈到 LSP 标识的最后一个 8bit 字节是 LSP 编号（LSP Number），并用来跟踪分段的 LSP。决策处理过程关注这个 LSP 编号有几个原因。首先，如果一个 LSP 编号为 0 并且剩余生存时间不为 0 的 LSP 不在路由器的数据库当中，那么决策处理过程将不会处理任何具有同样的系统标识但 LSP 编号不为 0 的 LSP。例如，假设在数据库中存在 LSP 标识为 0000.0c76.5b7c.00-01 和 0000.0c76.5b7c.00-02 的两条 LSP，但是在该数据库中没有包含 LSP 标识为 0000.0c76.5b7c.00-00 的 LSP，那么路由器将不会处理前面的两条 LSP。这种方法可以确保不会因不完整的 LSP 而导致不精确的路由选择决策。

另外，决策处理过程将仅从 LSP 编号为 0 的 LSP 中接受下面的信息：

- 数据库中超载位的设置信息；

- IS 类型字段的设置信息;
- 区域地址可选字段的设置信息。

但是在 LSP 编号不为 0 的 LSP 中, 决策处理过程将忽略这些设置信息。换句话说, 在一系列被分段的 LSP 中, 将由第一个 LSP 来宣告所有分段 LSP 的这三个设置信息。

如图 10-12 所示, 图中显示了一台 Cisco 的 IS-IS 路由器的路由选择表。在这里我们注意到存在 L1 和 L2 的路由, 并且有 3 个到达目的地的路由具有多条路径。每一条路由都带有一个相应的掩码, 这表明它们是支持 VLSM 技术的。最后, 在路由选择表中也指出了 IS-IS 路由的管理距离是 115。

```
Brussels#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
       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

Gateway of last resort is not set

    10.0.0.0 is variably subnetted, 8 subnets, 3 masks
C       10.1.3.0 255.255.255.0 is directly connected, Ethernet0
i L2    10.1.2.0 255.255.255.0 [115/30] via 10.1.3.2, Ethernet0
        [115/30] via 10.1.3.3, Ethernet0
i L1    10.1.5.0 255.255.255.0 [115/20] via 0.0.0.0, Serial0
        [115/20] via 10.1.4.2, Ethernet1
C       10.1.4.0 255.255.255.0 is directly connected, Ethernet1
i L2    10.1.255.4 255.255.255.252 [115/20] via 10.1.3.2, Ethernet0
i L2    10.1.255.0 255.255.255.0 [115/30] via 10.1.3.2, Ethernet0
i L1    10.1.255.8 255.255.255.252 [115/20] via 10.1.3.3, Ethernet0
i L1    10.1.6.240 255.255.255.240 [115/20] via 0.0.0.0, Serial0
        [115/20] via 10.1.4.2, Ethernet1

Brussels#
```

图 10-12 这个路由选择表显示了层 1 和层 2 IS-IS 路由

对于 L1 的路由器来说, 决策处理过程还有另外一个功能, 就是为区域间路由选择计算到达最近的 L2 路由器的路径。正如前面所提到的, 当一台 L2 或 L1/L2 路由器和其他的区域相连时, 路由器将通过在它的 LSP 中设置 ATT 位为 1 来通告这种情况。对于 L1 路由器, 决策处理过程将选择度量最近的 L1/L2 路由器作为它缺省的区域间路由器。当使用 IS-IS 协议进行 IP 协议的路由选择时, 在路由器中会记录一条到达 L1/L2 路由器的 IP 缺省路由。例如, 在图 10-13 中显示了一台 L1 路由器的链路状态数据库和相应的路由选择表。LSP 0000.0c0a.2c51.00-00 的 ATT 位设置为 1。基于这个信息, 决策处理过程将选择系统标识为 0000.0c0a.2c51 的路由器作为缺省的区域间路由器。在路由选择表中还显示了一条经过 10.1.255.6 可达的缺省路由 (0.0.0.0), 它的度量是 10。虽然在图 10-13 的两个表中显示的信息不太容易对照, 但实际上地址为 10.1.255.6 和系统标识为 0000.0c0a.2c51.00-00 的路由器指的是同一台路由器。

在图 10-13 中显示的信息突出了采用 IS-IS 协议的一个基本问题, 特别是在做故障排除时这个问题更加突出。虽然 TCP/IP 协议是被路由转发的协议, 但是决定路由选择策略的协议, 包括所有路由的控制报文和地址却都是 CLNS 协议。有时要把 CLNS 协议的信息和 IP 协议相关的信息关联起来是比较困难的。解决这种情况, 有一个很有用的命令是 **which-route**。


```

Paris#show isis database
IS-IS Level-1 Link State Database
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2C51.00-00  0x0000016D  0xA093        730           1/0/0
0000.3090.6756.00-00* 0x00000167  0xC103        813           0/0/0
0000.3090.6756.04-00* 0x0000014E  0x227F        801           0/0/0
0000.3090.C7DF.00-00  0x00000158  0x78A6        442           0/0/0
Paris#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
       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

Gateway of last resort is 10.1.255.6 to network 0.0.0.0

    10.0.0.0 is variably subnetted, 5 subnets, 2 masks
i L1    10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C       10.1.2.0 255.255.255.0 is directly connected, TokenRing0
i L1    10.1.255.4 255.255.255.252 [115/20] via 10.1.255.6, Serial0
C       10.1.255.0 255.255.255.0 is directly connected, Serial0
i L1    10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
i*L1 0.0.0.0 0.0.0.0 [115/10] via 10.1.255.6, Serial0
Paris#

```

图 10-13 如果 ATT 位设置为 1, 集成 IS-IS 就会增加一条到达最近的 L1/L2 路由器的 IP 缺省路由

这条命令主要用来确定某个具体的 CLNS 目的地址在路由选择表的定位。但是, 利用 **which-route** 命令也可以了解到一些关于某个具体 CLNS 地址和相关 IP 地址的有用信息。如图 10-14 所示, 显示了使用系统标识/电路标识 0000.0c0a.2c51.00 作为参数的 **which-route** 命令的输出信息, 这里的系统标识 0000.0c0a.2c51.00 指的就是在前面图 10-13 显示的数据库中 ATT=1 的那个 LSP。从输出结果可以看出, 要查询的系统标识的下一跳 IP 地址是 10.1.255.6。

```

Paris#which-route 0000.0C0A.2C51.00
Route look-up for destination 00.000c.0a2c.5100
Using route to closest IS-IS level-2 router

Adjacency entry used:
System Id      SNPA                Interface  State  Holdtime  Type Protocol
0000.0C0A.2C51 *HDLC*            Se0       Up     26        L1    IS-IS
Area Address(es): 47.0001
IP Address(es):  10.1.255.6
Uptime: 22:08:52
Paris#

```

图 10-14 使用 **which-route** 命令可以发现一些 CLNS 地址和 IP 地址的关联信息

10.1.4 IS-IS 的 PDU 格式

IS-IS 协议使用 9 种 PDU 类型来进行它的控制信息处理, 并使用一个 5 位二进制位的类型号来标识每一个 PDU 报文。所有的 PDU 报文可以归纳分为 3 类, 如表 10-1 所示。

在所有 IS-IS PDU 报文起始的 8 个 8bit 字节都是该报文的头部字段, 并且对于所有的 PDU 报文类型来说都是公用的部分, 如图 10-15 所示。这里将先讲述这些起始字段, 特有的 PDU

字段将在后续的章节讲述。

表 10-1

S-IS 协议的 PDU 报文类型

IS-IS PDU 报文	类 型 号
Hello PDU 报文	
层 1 LAN 的 IS-IS Hello PDU 报文	15
层 2 LAN 的 IS-IS Hello PDU 报文	16
点到点的 IS-IS Hello PDU 报文	17
链路状态 PDU 报文	
层 1 LSP 报文	18
层 2 LSP 报文	20
序列号 PDU 报文	
层 1 完全序列号报文	24
层 2 完全序列号报文	25
层 1 部分序列号报文	26
层 2 部分序列号报文	27

				长度, 8bit 字节数
域内路由选择协议鉴别符				1
长度标识符				1
版本/协议标识扩展				1
标识长度				1
R	R	R	PDU 类 型	1
版 本				1
保 留				1
最大区域地址数				1
PDU 专有字段				
可变长度字段				

图 10-15 IS-IS PDU 报文起始的 8 个 8bit 字节

- 域内路由选择协议鉴别符 (**Intradomain Routeing Protocol Discriminator**) —— 这是由 ISO9577¹ 分配的一个恒定不变的数值, 用来标识网络层协议数据单元 (NPDU)。在所有的 IS-IS PDU 报文中, 该字段的值都是 0x83。
- 长度标识符 (**Length Indicator**) —— 标识该报文固定报文头部字段的长度, 以 8bit 字节数表示。
- 版本/协议标识扩展 (**Version/Protocol ID Extension**) —— 当前始终设置为 1。
- 标识长度 (**ID Length**) —— 用来标识该路由选择域内使用的 NSAP 地址和 NET 的系统标识 (**System ID**) 的长度。该字段的取值可以是以下几个数值之一:
 - 1~8 的整数, 表示系统标识字段具有相同长度的 8bit 字节数;

¹ 国际标准化组织, “Protocol Identification in the Network Layer”, ISO/IEC TR 9577, 1990 年。

- 0, 表示系统标识字段的长度为 6 个 8bit 字节;
- 255, 表示系统标识字段为空 (0 个 8bit 字节)。

在 Cisco 路由器中系统标识字段的长度固定为 6 个 8bit 字节, 因此, 在由 Cisco 路由器始发的 PDU 报文中, 这个标识长度字段的值将始终是 0。

- **PDU 类型**——是一个 5 位的字段, 取值范围可以是表 10-1 中显示的 PDU 报文类型号中的任何一个。该字段的前 3 位 (R) 作为保留位, 始终为 0。
- **版本号 (Version)**——当前始终设置为 1, 这和第 3 个 8bit 字节中的版本/协议标识扩展字段是一致的。
- **保留位**——当前设置为全 0。
- **最大区域地址数 (Maximum Area Addresses)**——表示该 IS 区域所允许的最大区域地址数量。这个字段的值可以是下面数值之一:
 - 1~254 的整数, 表示该区域实际所允许的最大区域地址数;
 - 0, 表示该 IS 区域最大只支持 3 个区域地址数。

Cisco IOS 软件最大支持 3 个区域地址, 因此, 在由 Cisco 路由器始发的 IS-IS PDU 报文中, 该最大区域地址数字段的值始终是 0。

在图 10-16 中, 显示了使用协议分析仪捕获到的某个 IS-IS PDU 报文起始的 8 个 8bit 字节。紧跟在公共报文头部字段之后的专有 PDU 字段也是报文头部的一部分。根据 PDU 报文类型的不同它们会有所变化, 这将在讲述具体 PDU 类型的章节中介绍。

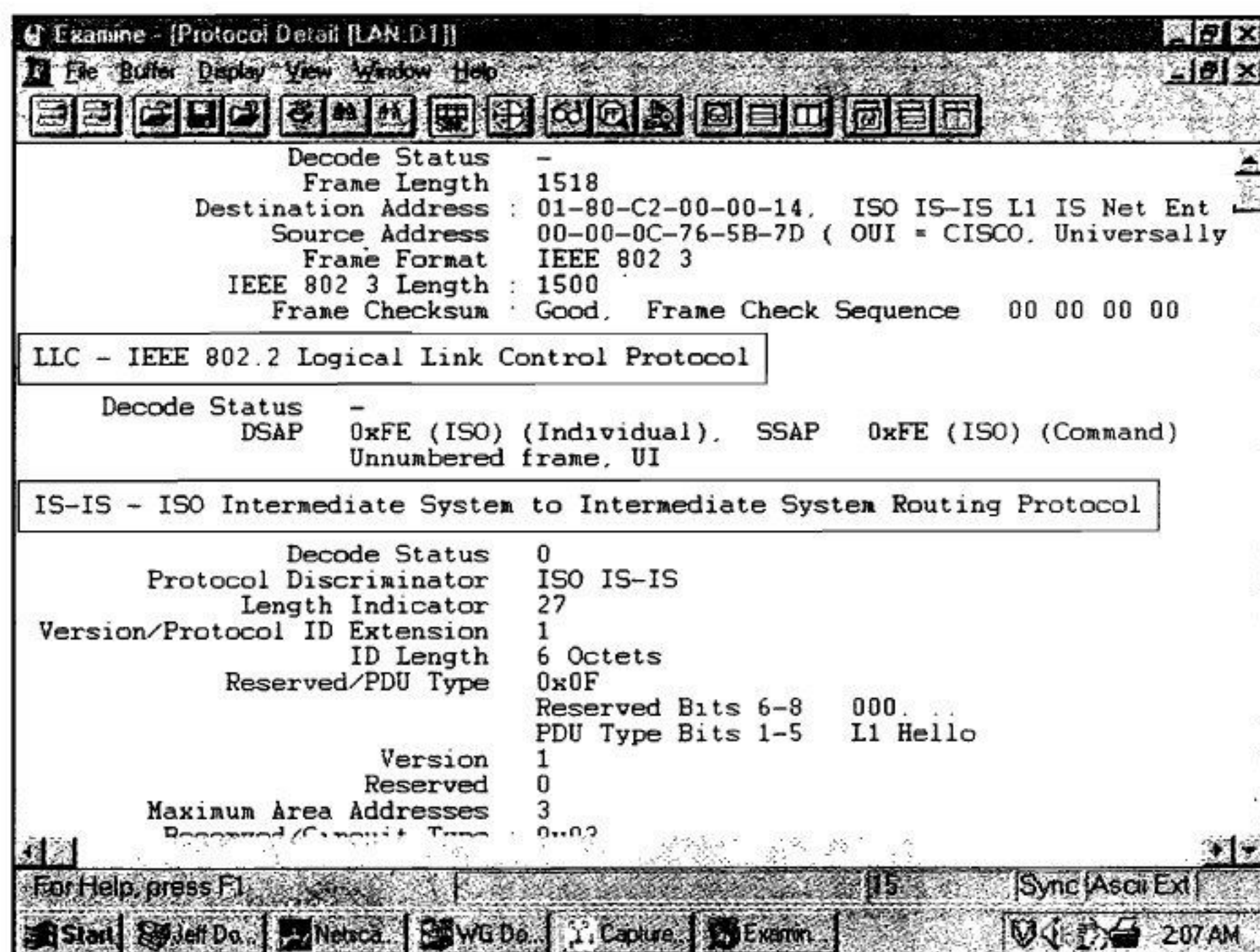


图 10-16 使用协议分析仪捕获到的某个 IS-IS PDU 报文起始的 8 个 8bit 字节

1. CLV 字段

紧跟在专有 PDU 字段之后的可变长度字段是代码/长度/值 (Code/Length/Value, CLV)¹

¹ 在 ISO10589 中并不使用 CLV 这样的字母缩写, 在这里使用只不过为了方便而已。实际上, 读者可以利用前面第 8 章“增强型内部网关路由选择协议 (EIGRP)”中讲述 EIGRP TLV 的概念来熟悉 CLV 的概念。而且, 在 RFC1195 中, 也把集成 IS-IS CLV 称为 TLV。

这 3 个参数的不同组合,如图 10-17 所示。代码是一个指定值字段信息类型的数字,长度用来指定值字段的长度,而值字段就是它本身的信息内容。注意,长度字段只用了一个 8bit 字节来表示,这也就意味着值字段大小的最大值为 255 个 8bit 字节。

	长度, 8bit 字节数
代 码	1
长 度	1
值	长 度

图 10-17 IS-IS 的代码/长度/值参数的组合在 IS-IS 协议中的功能和
类型/长度/值参数的组合在 EIGRP 协议中起到的功能相同

表 10-2 中列出了所有 IS-IS 协议里 CLV 的代码。在这个表中也指出了哪些 CLV 是 ISO10589 指定的,哪些是 RFC1195 指定的。ISO 指定的 CLV 是设计在 CLNP 协议使用的,但它们中的大多数也使用在 IP 协议中。RFC 指定的 CLV 只设计用在 IP 协议当中。如果一台路由器不能识别一个特有的 CLV 代码,那么它将忽略这个 CLV。这种设计方法可以允许在同一个 PDU 报文中携带支持 CLNP 协议的 CLV、支持 IP 协议的 CLV、或者同时携带这两种 CLV。

表 10-2 IS-IS 协议中使用的 CLV 代码

代 码	CLV 的类型	ISO 10589	RFC 1195
1	区域地址	X	
2	中间系统邻居 (LSP)	X	
3	终端系统邻居*	X	
4	区域分段指定的层 2 中间系统**	X	
5	前缀邻居*	X	
6	中间系统邻居 (Hello)	X	
8	填充项	X	
9	LSP 条目	X	
10	认证信息	X	
128	IP 内部可达性信息		X
129	支持的协议		X
130	IP 外部可达性信息		X
131	域间路由选择协议信息		X
132	IP 接口地址		X
133	认证信息***		X

*终端系统邻居和前缀邻居的 CLV 与 IP 路由选择没有关系,因此不在本书中讲述。

**这个 CLV 用作分段区域的修复, Cisco 的路由器并不支持。

***RFC1195 为 IP 的认证指定了这个代码,但 Cisco 的路由器中使用的是 ISO 指定的代码 10。

虽然大多数的 CLV 都在不止一种 IS-IS PDU 报文类型中使用,但是只有一个 CLV (认证信息 CLV) 是在所有的 PDU 报文中都使用的。在下面讲述 IS-IS PDU 报文格式的章节中,将会列出每一种 PDU 报文用到的 CLV。每一种 CLV 的格式将只在它第一次出现时讲述一次。表 10-3 中总结了每种 PDU 所用到的 CLV。

2. IS-IS Hello PDU 报文

IS-IS Hello PDU 报文是同一条链路上的 IS-IS 路由器用来发现它们的 IS-IS 邻居路由器

的。一旦路由器发现了它的邻居路由器并且建立成功邻接关系，Hello PDU 报文的工作就可以只担当 keepalive 的功能，从而维护已有的邻接关系并将邻接关系中任何参数的变化通知邻居。

表 10-3 每一种 IS-IS PDU 报文所用到的 CLV

CLV 的类型	PDU 报文类型								
	15	16	17	18	20	24	25	26	27
区域地址	X	X	X	X	X				
中间系统邻居(LSP)				X	X				
终端系统邻居				X					
区域分段指定的层 2 中间系统					X				
前缀邻居					X				
中间系统邻居 (Hello)	X	X							
填充项	X	X	X						
LSP 条目						X	X	X	X
认证信息	X	X	X	X	X	X	X	X	X
IP 内部可达性信息				X	X				
支持的协议	X	X	X	X	X				
IP 外部可达性信息					X				
域间路由选择协议信息					X				
IP 接口地址	X	X	X	X	X				

一个 IS-IS PDU 报文大小的上限可以由始发路由器的缓冲区大小或者传输这个 PDU 报文的数据链路的 MTU 值来确定。ISO10589 规定 IS-IS Hello 报文必须填充一个 8bit 字节小于这个最大值，以便使一台路由器可以和它的邻居路由器进行彼此 MTU 大小的通信。更为重要的是，发送达到或接近链路 MTU 大小的 Hello 报文可以帮助检测这样一种链路的失效情形：较小的 PDU 报文可以通过，但是较大的 PDU 报文会被丢弃。这种设计方式的好处是有争议的，因为在一些低速的串行链路上发送像这样大的 Hello 报文费用是比较大的。

有两种类型的 IS-IS Hello 报文：LAN Hello 报文和点到点 Hello 报文。LAN Hello 报文可以进一步分为 L1 和 L2 的 LAN Hello 报文。但是，这两种类型的 LAN Hello 报文的格式是相同的，如图 10-18 所示。图 10-19 显示了协议分析仪捕获到的一个层 2 的 LAN Hello 报文。

- **电路类型 (Circuit Type)**——是一个 2 位的字段（前面 6 位是保留位，始终设置为 0），用来指定该路由器是 L1 路由器（01）、L2 路由器（02），还是 L1/L2 路由器（11）。如果这两位都为 0（00），那么这个 PDU 报文整个都会被忽略；
- **源标识 (Source ID)**——是指始发该 Hello 报文的路由器的系统 ID；
- **抑制时间 (Holding Time)**——是指邻居路由器在宣告始发路由器失效之前，它所等待接收下一个 Hello 报文的时间间隔；
- **PDU 报文长度**——是指整个 PDU 报文长度的 8bit 字节数；
- **优先级**——是一个用来选取 DR 路由器的 7 位字段。这个字段可以设置成 0~127 之间的数值，而且数值越大就表示优先级越高。L1 的指定路由器是通过 L1 LAN Hello 报文中的优先级特性选取出的，而 L2 的指定路由器是通过 L2 LAN Hello 报文中的优先级特性选取出的。

- **LAN ID**——就是指定路由器的系统 ID 再加上一个 8bit 字节 (伪节点 ID)，用来区分这个 LAN ID 和同一台指定路由器上的其他 LAN ID。

在 IS-IS LAN Hello 报文中可以使用下面的多种 CLV：¹

							长度, 8bit字节数
域内路由选择协议鉴别符							1
长度标识符							1
版本/协议标识扩展							1
标识长度							1
R	R	R	PDU类型				1
版 本							1
保 留							1
最大区域地址数							1
R	R	R	R	R	R	电路类型	1
源标识							标识长度
抑制时间							2
PDU长度							2
R	优先级						2
LAN ID							标识长度+1
可变长度字段							

图 10-18 IS-IS LAN Hello PDU 报文的格式

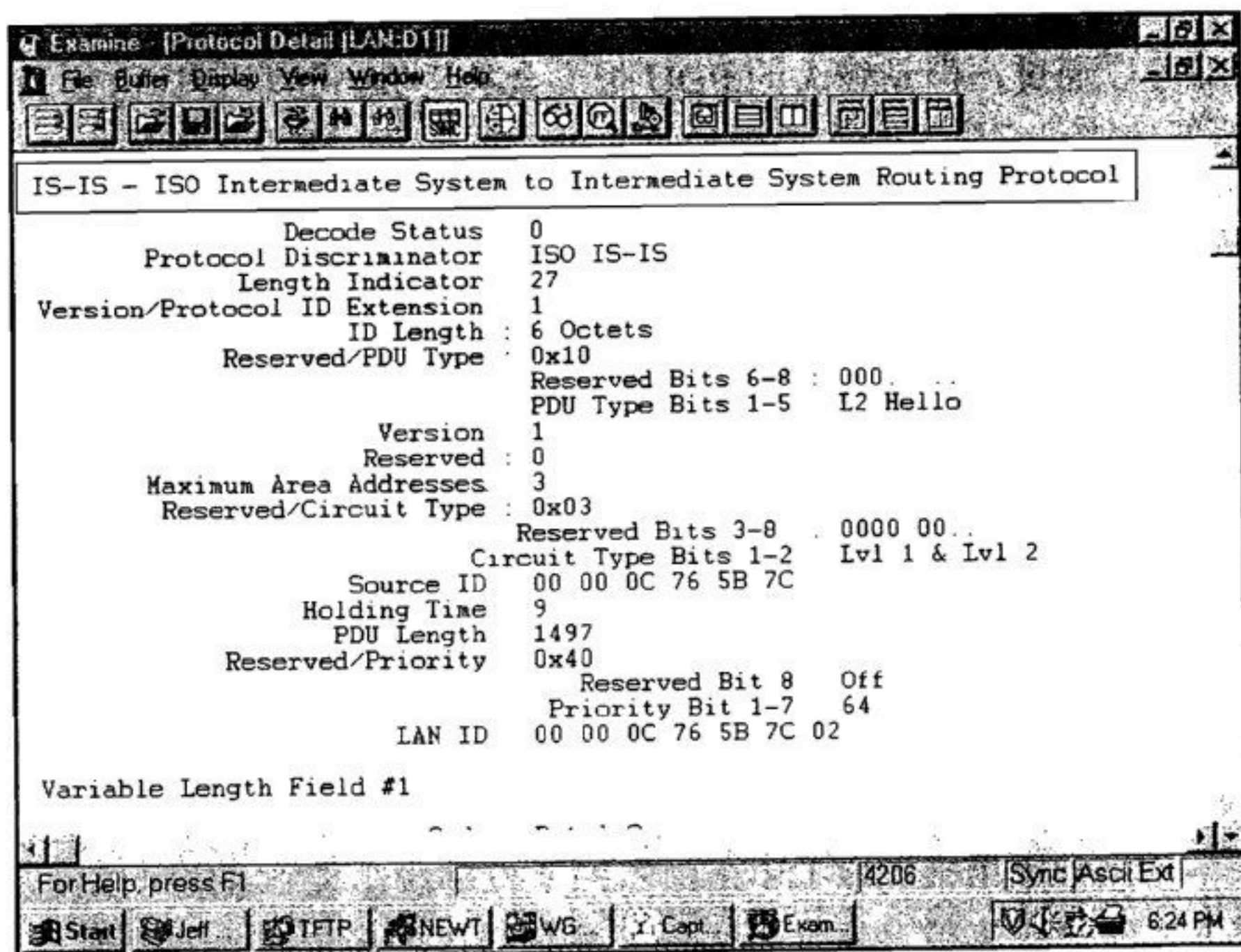


图 10-19 通过协议分析仪捕获到的 LAN Hello 报文显示了只有 Hello PDU 报文才有的字段

¹ 作为一个提示, RFC1195 规定了认证信息 CLV 的类型代码是 133。而 Cisco 路由器使用 ISO 规定的类型代码 10 来标识认证信息 CLV。

- 区域地址 CLV（类型 1）；
- 中间系统邻居 CLV（类型 6）；
- 填充 CLV（类型 8）；
- 认证信息 CLV（类型 10）；
- 支持的协议 CLV（类型 129）；
- IP 接口地址 CLV（类型 132）。

如图 10-20 所示，显示了 IS-IS 点到点 Hello PDU 报文格式。点到点 Hello 报文的格式和 LAN Hello 报文相比，除了没有优先级字段外基本上是一样的，而且它用本地电路 ID 替代了 LAN ID 字段。和 LAN Hello 报文不同，L1 和 L2 的信息是在同一个点到点 Hello PDU 报文中传送的。

							长度，8bit字节数
域内路由选择协议鉴别符							1
长度标识符							1
版本/协议标识扩展							1
标识长度							1
R	R	R	PDU类型				1
版 本							1
保 留							1
最大区域地址数							1
R	R	R	R	R	R	电路类型	1
源标识							标识长度
抑制时间							2
PDU长度							2
本地电路标识							1
可变长度字段							

图 10-20 IS-IS 点到点 Hello PDU 报文的格式

- **本地电路 ID（Local Circuit ID）**——是一个 1 个 8bit 字节的标识字段，由始发路由器发送该 Hello 报文时分配给这条电路，并且在路由器的接口上是惟一的。在点到点链路的另一端，Hello 报文中的本地电路 ID 可能包含，也可能不包含同样的值。

IS-IS 点到点 Hello 报文不使用中间系统邻居 CLV。除了这个不同，其他 LAN Hello 报文中用到的 CLV 也同样地用在点到点 Hello 报文中。

（1）区域地址 CLV

如图 10-21 所示，区域地址 CLV 是在始发路由器上配置的，并用来通告该区域的地址。正如多个地址长度/区域地址字段所表示的，一台路由器可以配置多个区域地址。但是，在从 Cisco 路由器始发出来的 PDU 报文中，地址长度/区域地址字段的数目从来不会超过 3 个，因为在 Cisco 的路由器中所支持区域地址的最大数目是 3 个。

在图 10-22 中显示了一个 IS-IS Hello PDU 报文的部分信息。“Variable Length Field #3”部分显示的就是一个区域地址 CLV 的信息，它总共有 6 个 8bit 字节长，包括两个区域地址：

47.0002 (3 个 8bit 字节) 和 0 (1 个 8bit 字节)。

	长度, 8bit字节数
代 码 = 1	1
长 度	1
地址长度	1
区域地址	地址长度
多个字段	
地址长度	1
区域地址	地址长度

图 10-21 区域地址 CLV 的格式

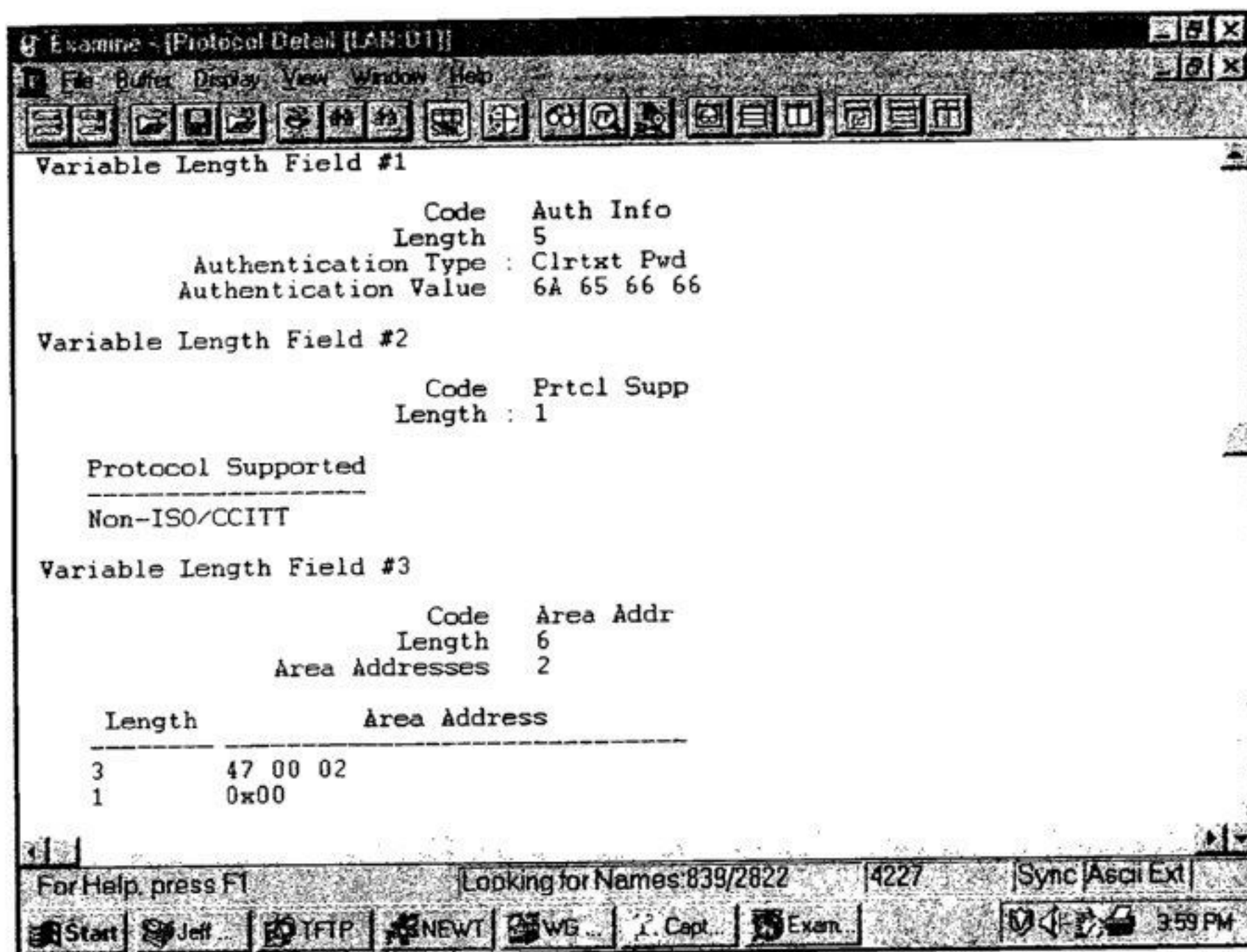


图 10-22 “Variable Length Field #3”部分显示的就是一个区域地址 CLV 的信息。为了便于观察, 分析仪也在 CLV 里面指出了所列出的所有地址数, 但是这个信息并不是 CLV 字段的一部分

(2) 中间系统邻居 CLV

中间系统邻居 CLV 列出了本地路由器所有邻居的系统标识, 但是这些邻居路由器必须满足在最新的一个抑制时间间隔内, 能够被本地路由器接收到它们发出的 Hello 报文。这里可以注意到, 这个 CLV 字段对于 IS-IS LAN Hello 报文所起到的功能, 其实在 OSPF 协议中也有相类似的功能: 为了验证双向通信, 本地路由器列出了最近所有发出 Hello 报文并能够被它收到的邻居路由器。

这个 CLV 只能用在 LAN Hello 报文中。由于点到点 Hello 报文中没有指定路由器的选取过程, 因而在点到点 Hello 报文中没有这个 CLV。同时, 这里的 CLV 也和 LSP 报文中使用中间系统邻居 CLV 有所不同, 这可以通过它们的类型代码区分开来。L1 LAN Hello 报文只列出了 L1 的邻居, 而 L2 LAN Hello 报文也只列出了 L2 的邻居。虽然携带系统 ID 的字段长

度经常发生变化,但这个 CLV 字段的长度却始终是 6 个 8bit 字节长。这个 CLV 字段的长度之所以固定是因为系统 ID 总是属于局域网上的路由器的,因而也就是 6 个 8bit 字节长的 MAC 地址标识。图 10-24 中显示的“Variable Length Field #5”部分表示了一个中间系统邻居 CLV,它只列出了一个邻居——0000.0c0a.2aa9。

	长度, 8bit 字节数
代 码 = 6	1
长 度	1
LAN 长度	6
多个字段	
LAN 地址	6

图 10-23 Hello PDU 报文中的中间系统邻居 CLV 的格式

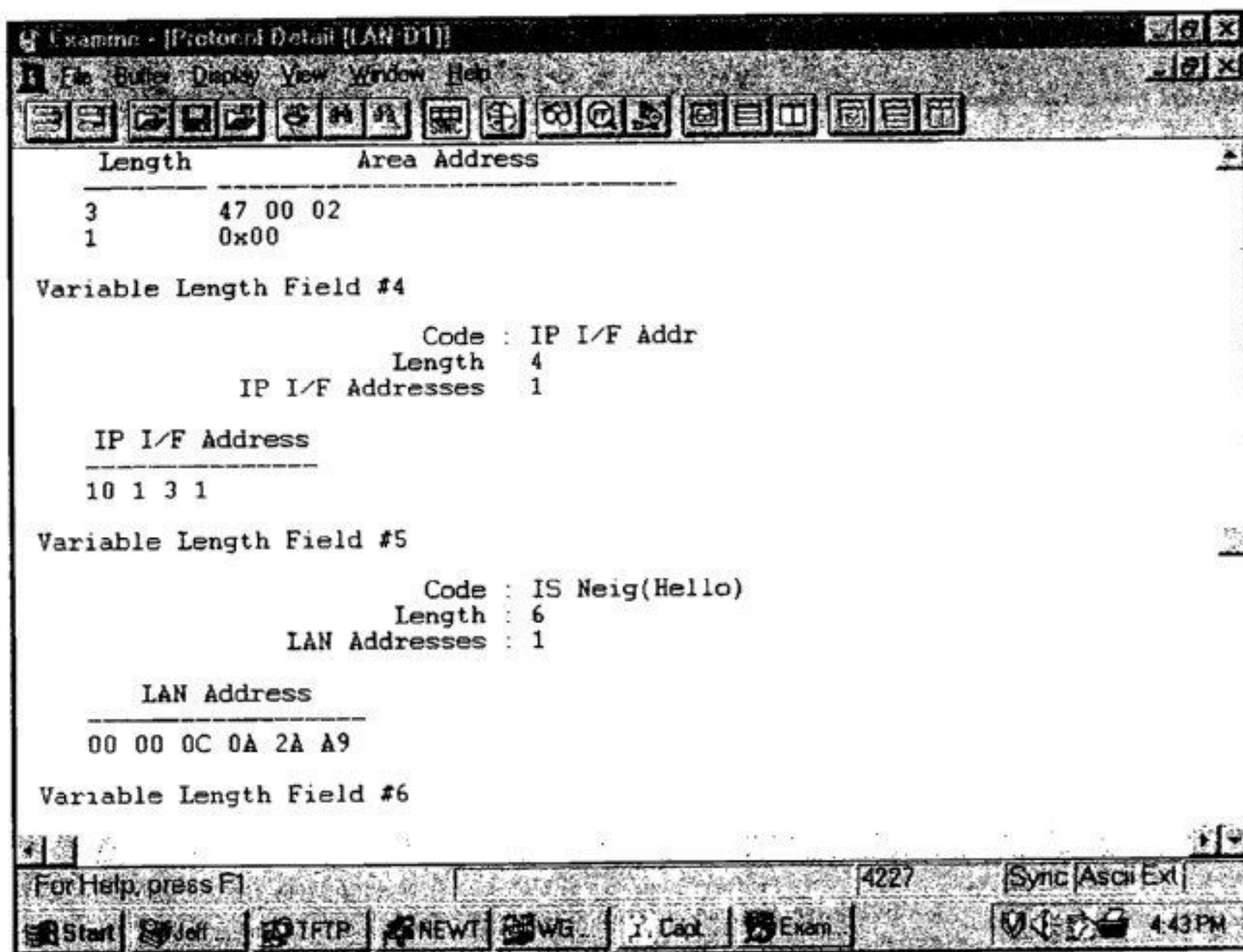


图 10-24 “Variable Length Field #5”部分显示了一个中间系统邻居 CLV

(3) 填充 CLV

填充 CLV 是用来填充一个 Hello PDU 报文以使它达到可允许的最大报文长度。由于一个值字段的最大长度为 255 个 8bit 字节,因此经常会使用多个填充 CLV 字段。由于填充 CLV 的内容是被路由器忽略的,因此在它的值字段内可以设置任意的数值。在 Cisco 的路由器中这些位的值都被设置为 0,如图 10-25 所示。

(4) 认证信息 CLV

如图 10-26 所示,认证信息 CLV 只有在配置认证时才会使用。认证类型字段包含了一个 0~255 之间的数字,用来指定认证使用的类型,因此也指定认证值字段所包含的认证信息的类型。目前 ISO10589 只定义了一种认证类型,并且 Cisco 路由器也仅支持的这种认证类型是

明文口令认证, 也就是认证类型 1。

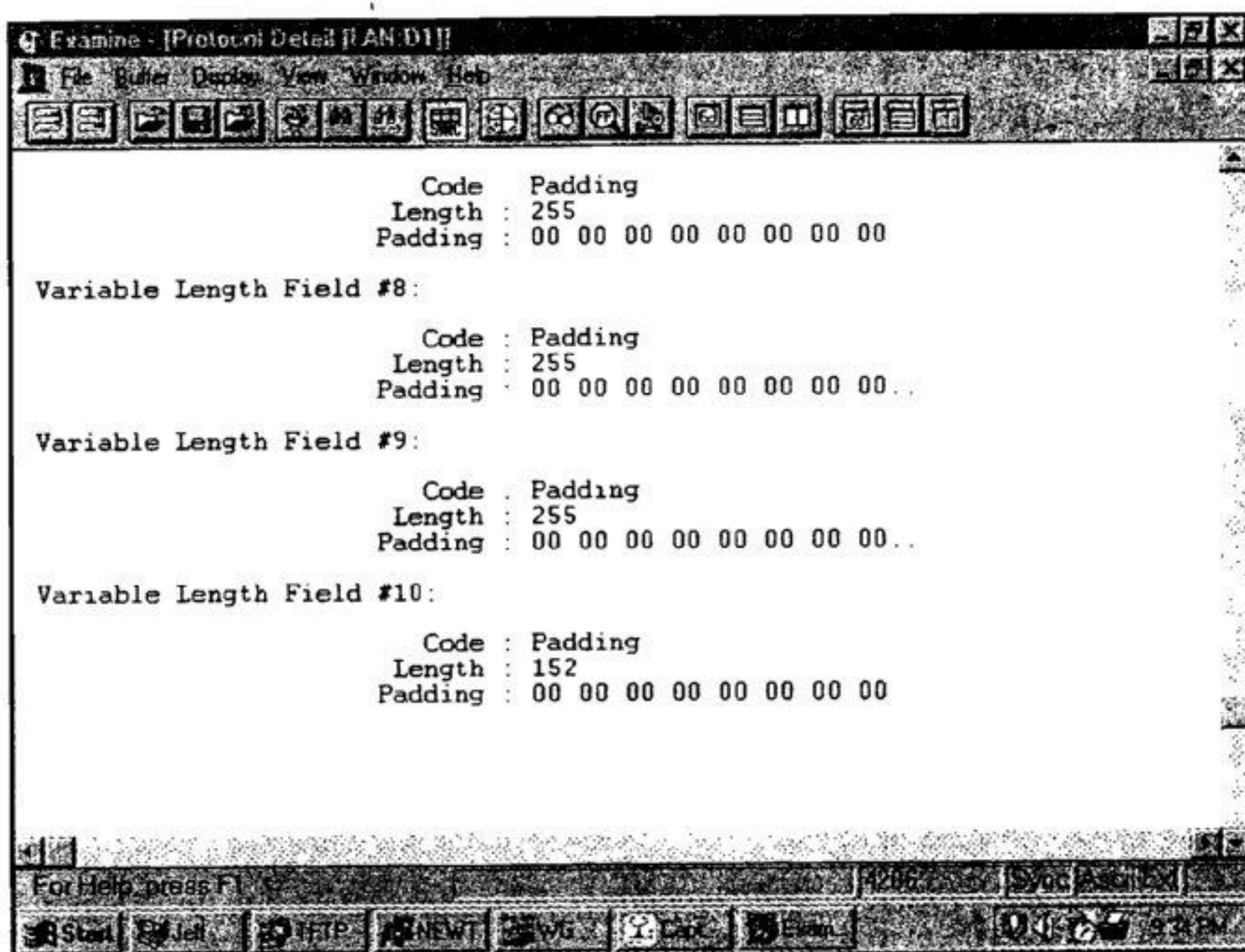


图 10-25 图 10-19 中显示的 Hello PDU 报文末尾的可变长字段就是填充 CLV, 它把图 10-19 中显示的 PDU 报文大小增大到 1497 个 8bit 字节。再加上 3 个 8bit 字节的 LLC 头部和 18 个 8bit 字节的以太网头部, 整个帧的大小就是最大为 1518 个 8bit 字节的以太网 MTU 的大小了

	长度, 8bit 字节数
代 码 = 10	1
长 度	1
认证类型	1
认证值	长度 - 1

图 10-26 认证信息 CLV 的格式

如图 10-22 所示, “Variable Length Field #1”部分显示的就是一个认证信息 CLV。在这里, 口令是 “jeff”, 并使用十六进制的 ASCII 码字符来表示。

(5) 支持的协议 CLV

如图 10-27 所示, 支持的协议 CLV 是由 RFC1195 规定的, 它用来指定 PDU 报文的始发路由器所支持的协议——仅支持 CLNP、仅支持 IP 或同时支持这两种协议。对于每一种所支持的协议, 在 CLV 字段里都会有一个相应的 1 个 8bit 字节的网络层协议标识符(Network Layer Protocol Identifier, NLPID), 这个标识符是由 ISO/TR 9577 规定的。IP 协议的网络层协议标识符是 0x81。在图 10-22 中的“Variable Length Field #2”部分显示的就是一个支持的协议 CLV。

(6) IP 接口地址 CLV

如图 10-28 所示, IP 接口地址 CLV 是指发出 PDU 报文的接口地址或 IP 地址。因为长度

字段是 1 个 8bit 字节，因而 IS-IS 路由器的接口理论上最多可以有 63 个 IP 地址。在图 10-24 中的“Variable Length Field #4”部分显示的就是一个 IP 接口地址 CLV，表明所捕获的 Hello PDU 报文是由地址为 10.1.3.1 的接口发出的。

长度，8bit字节数	
代 码 = 129	1
长 度	1
NLPID	1
多个字段	
NLPID	1

图 10-27 支持的协议 CLV 格式

长度，8bit字节数	
代 码 = 132	1
长 度	1
IP地址	4
多个字段	
IP地址	4

图 10-28 IP 接口地址 CLV 的格式

3. IS-IS 协议链路状态 PDU 报文格式

IS-IS 协议中 LSP 的功能在本质上和 OSPF 协议中 LSA 的功能是一样的。一台 L1 路由器把 L1 类型的 LSP 泛洪到整个区域，用来确定它的邻接关系和这些邻接关系的状态。一台 L2 路由器把 L2 类型的 LSP 泛洪到整个层 2 的域，用来确定它与其他 L2 路由器的邻接关系，并标识出通告 L2 路由器能够到达的地址前缀。

如图 10-29 所示，显示一个 IS-IS LSP 的格式。这个格式对于 L1 LSP 和 L2 LSP 都是相同的。

- **PDU 长度**——是指整个 PDU 报文的长度，用 8bit 字节数表示。
- **剩余生存时间 (Remaining Lifetime)**——在确认一个 LSP 过期之前等待的秒数。
- **LSP ID**——可以是系统 ID、伪节点 ID 或 LSP 报文的 LSP 编号。LSP ID 在“更新处理过程”一节中有更为详细的描述。
- **序列号**——是一个 32 位的无符号整数。
- **校验和**——对 LSP 内容的校验和。
- **P 位**——是指分段区域的修复位。虽然这一位存在于 L1 和 L2 的 LSP 报文中，但是它实际上只和 L2 的 LSP 报文有关。当该位被设置为 1 时，表明始发路由器支持自动地修复区域的分段情况。Cisco IOS 软件并不支持这个功能，因此 Cisco 路由器始发的 LSP 报文中，该位始终设置为 0。

					长度, 8bit字节数
域内路由选择协议鉴别符					1
长度标识符					1
版本/协议标识扩展					1
标识长度					1
R	R	R	PDU类型		1
版 本					1
保 留					1
最大区域地址数					1
PDU长度					2
剩余生存时间					2
LSP ID					标识长度+2
序列号					4
校验和					2
P	ATT		OL	IS 类 型	1
可变长度字段					

图 10-29 IS-IS LSP 的格式

- **区域关联位 (ATT)**——是一个 4 位的字段, 用来指明始发路由器是与一个或多个其他区域相连的。虽然区域关联位也存在于 L1 和 L2 的 LSP 报文中, 但是它们实际上只和 L1/L2 路由器始发的 L1 LSP 报文有关。这 4 位用来表明相连的区域究竟使用哪一种度量方式。从左到右这 4 位依次表示:
 - 位 7: 差错度量 (Error);
 - 位 6: 代价度量 (Expense);
 - 位 5: 时延度量 (Delay);
 - 位 4: 缺省度量 (Default)。

Cisco IOS 软件仅支持缺省度量, 因此位 5~位 7 始终设置为 0。

- **超载位(OL)**——链路状态数据库的超载位。在一般情况下, 该位设置为 0。如果始发路由器正处于一个内存超载的情形, 将会把超载位设置为 1。收到超载位设置为 1 的 LSP 报文的路由器将仍然会转发数据包到与这台始发超载信号的路由器直连的网络上, 但是, 不会再使用这台始发路由器作为转发数据包的过渡路由器 (transit router) 了。
- **中间系统类型 (IS Type)**——是一个 2 位的字段, 用来指明始发路由器是 L1 路由器还是 L2 路由器:
 - 00=未使用的值;
 - 01=L1;
 - 10=未使用的值;
 - 11=L2。

一台 L1/L2 路由器可以根据收到的 LSP 是 L1 类型的还是 L2 类型的 LSP 来确定设置这

两位的值。

在 L1 的 LSP 报文中可以使用下面的 CLV 字段：

- 区域地址（类型 1）；
- 中间系统邻居（类型 2）；
- 终端系统邻居（类型 3）；
- 认证信息（类型 10）；
- IP 内部可达性信息（类型 128）；
- 支持的协议（类型 129）；
- IP 接口地址（类型 132）。

在 L2 的 LSP 报文中可以使用下面的 CLV 字段：

- 区域地址（类型 1）；
- 中间系统邻居（类型 2）；
- 区域分段指定的层 2 中间系统（类型 4）；
- 前缀邻居（类型 5）；
- 认证信息（类型 10）；
- IP 内部可达性信息（类型 128）；
- IP 外部可达性信息（类型 130）；
- 域间路由选择协议信息（类型 131）；
- 支持的协议（类型 129）；
- IP 接口地址（类型 132）。

如图 10-30 所示，显示了 L1/L2 路由器始发的一条 L1 类型的 LSP 报文。

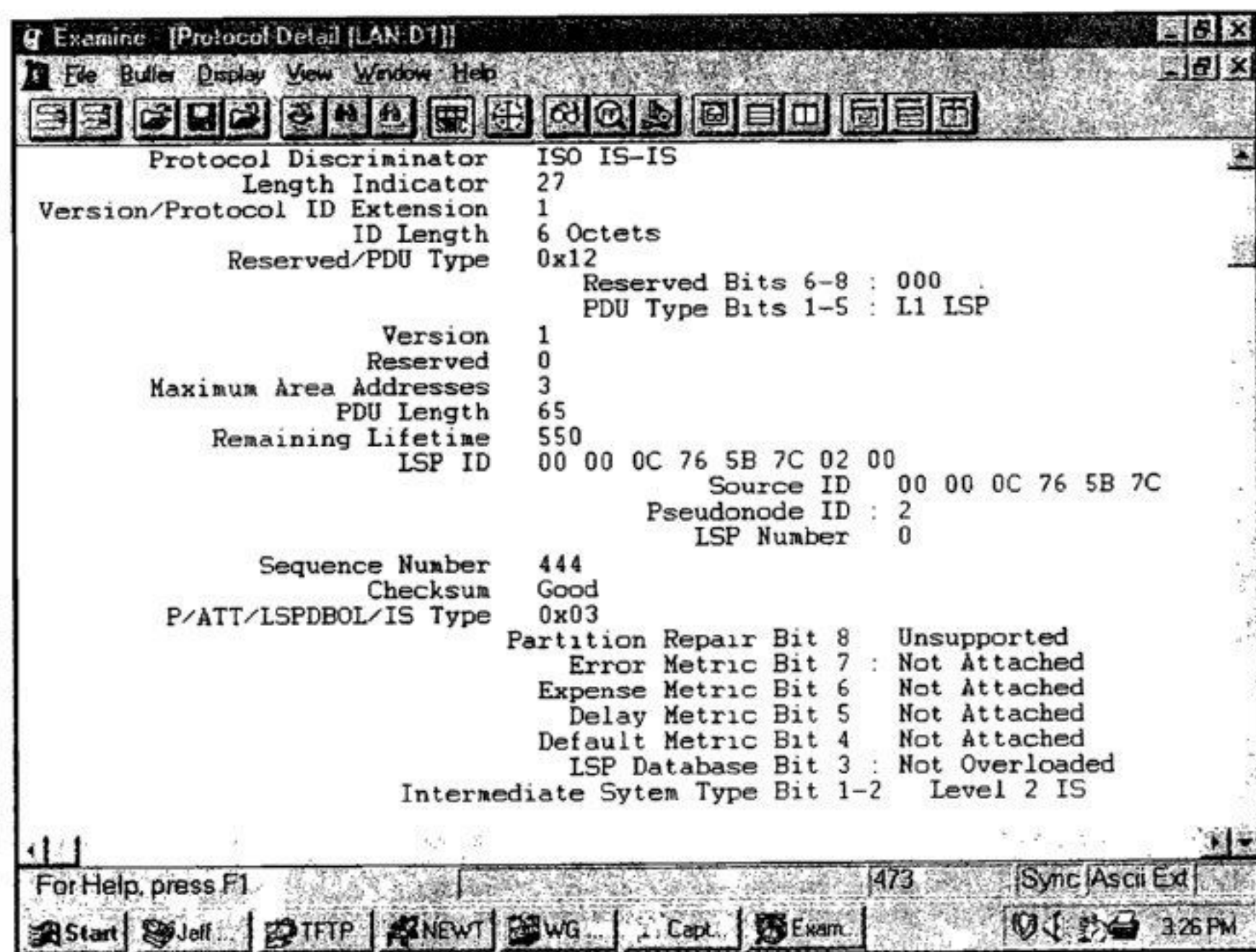


图 10-30 协议分析仪捕获到的 LSP 报文信息

(1) 中间系统邻居 CLV (LSP)

中间系统邻居 CLV 用来在 LSP 报文中列出始发路由器的 IS-IS 邻居（包括伪节点），如图 10-31 所示。同时，它也列出了到达每一个邻居路由器的链路的度量。

			长度, 8bit字节数
代 码 = 2			1
长 度			1
虚拟标记			1
R	I/E	缺省度量	1
S	I/E	时延度量	1
S	I/E	开销度量	1
S	I/E	差错度量	1
邻居标识			标识长度+1
多个字段			
R	I/E	缺省度量	1
S	I/E	时延度量	1
S	I/E	开销度量	1
S	I/E	差错度量	1
邻居标识			标识长度+1

图 10-31 LSP 报文中的中间系统邻居 CLV 的格式

- **虚拟标记(Virtual Flag)**——这个字段虽然有 8 位长,但取值只有 0x01 或者 0x00。当这个字段设置为 0x01 时,表示该链路是一个用来修复分段区域的层 2 类型的虚链路。这时该字段只和支持区域分段能力的 L2 路由器相关,由于 Cisco 路由器并不支持这一特性,因此在 Cisco 路由器始发的 LSP 报文中该字段始终设置为 0x00。
- **R 位**——是一个保留位,始终设置为 0。
- **I/E 位**——该位和每个度量有关,用来指明相关的度量是内部度量还是外部度量。该位在中间系统邻居 CLV 字段中没有意义,因为对 IS-IS 域来说所有的邻居路由器都被定义为内部的了。因此,在中间系统邻居 CLV 字段中该位始终设置为 0。
- **缺省度量**——是一个 6 位的缺省度量,用来表示始发路由器到达所列出的邻居的链路度量,大小范围在 0~63 之间。
- **S 位**——该位和每一个可选度量有关,用来指明相关的度量是被支持(0)的,还是不被支持(1)的。由于 Cisco 路由器不支持其他所有的 3 种可选度量,因此这一位总是被设置为 1,并把相关的长度为 6 位的度量字段全部设置为 0。
- **邻居标识**——是指邻居的系统 ID,再加上一个额外的 8bit 字节。如果该邻居是一台路由器,末尾的那个 8bit 字节就设置为 0x00。如果该邻居是一个伪节点,那么系统 ID 就是指定路由器,末尾的那个 8bit 字节就是伪节点的 ID。

图 10-32 中显示了一个中间系统邻居 CLV 的部分信息。

(2) IP 内部可达性信息 CLV

如图 10-33 所示,IP 内部可达性信息 CLV 列出了与通告该 LSP 报文的路由器直连的路由选择域内的 IP 地址和相关的掩码信息。这个 CLV 使用在 L1 和 L2 类型的 LSP 报文中,但是从来不会出现在伪节点的 LSP 报文中。度量字段的含义是和中间系统邻居 CLV 中的度量字段一样的,但是它没有与可选度量相关联的 I/E 位。

作为替代，这一位作为保留位并总是设置为 0。像中间系统邻居 CLV 一样，这个 CLV 字段中的 I/E 位也总是设置为 0，因为在这个 CLV 字段中通告的地址总是内部地址。如图 10-34 所示，显示了协议分析仪捕获到的一个 IP 内部可达性信息 CLV 的信息。

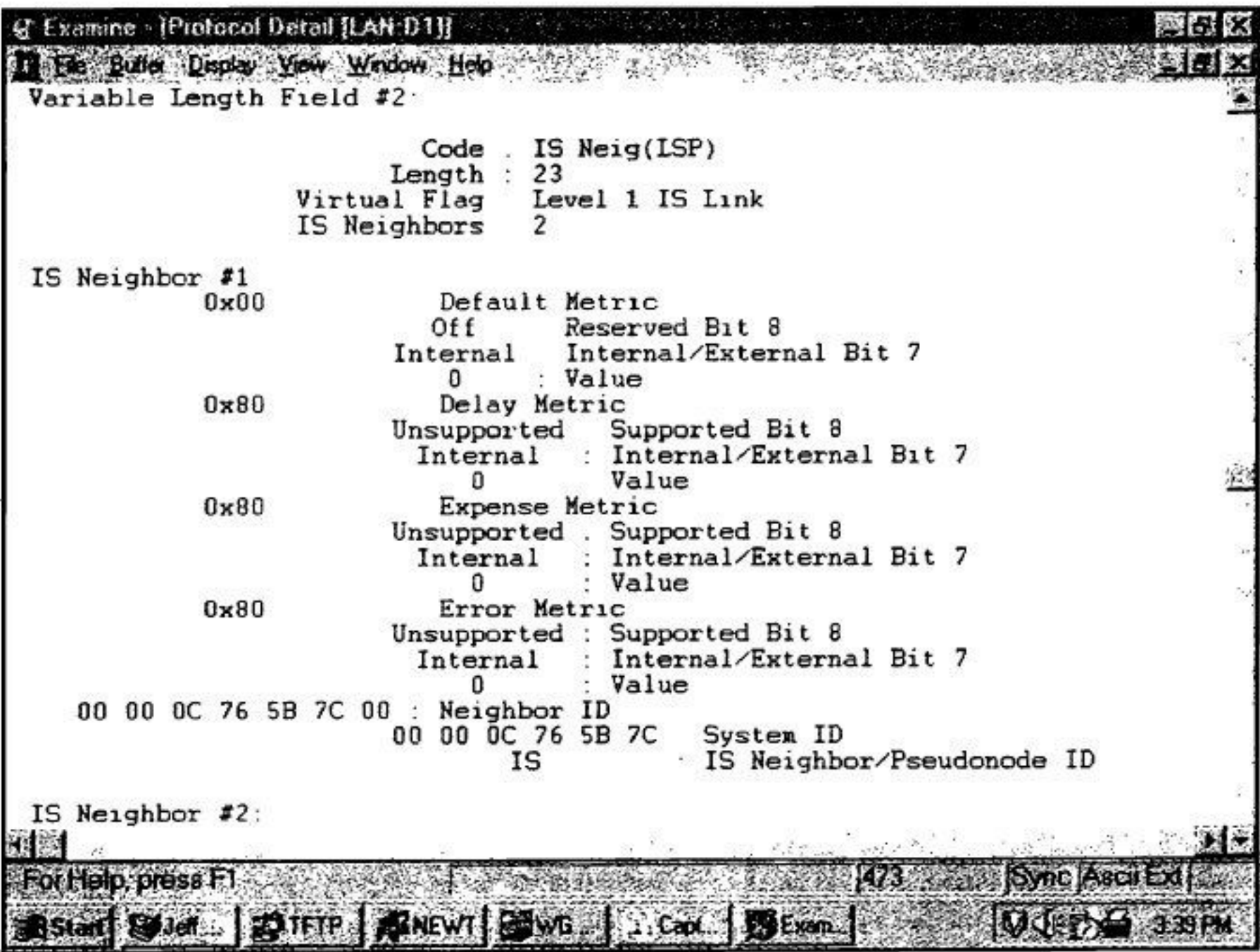


图 10-32 一个 LSP 报文中的中间系统邻居 CLV 的部分信息

			长度，8bit字节数
代 码 = 128			1
长 度			1
R	I/E	缺省度量	1
S	R	时延度量	1
S	R	开销度量	1
S	R	差错度量	1
IP地址			4
子网掩码			4
多个字段			
R	I/E	缺省度量	1
S	R	时延度量	1
S	R	开销度量	1
S	R	差错度量	1
IP地址			4
子网掩码			4

图 10-33 IP 内部可达性信息 CLV 的格式

(3) IP 外部可达性信息 CLV

IP 外部可达性信息 CLV 列出了到达 IS-IS 路由选择域外部的 IP 地址和相关的掩码，这

些外部的目的地址可以通过始发路由器的某个接口到达。IP 外部可达性信息 CLV 的格式和图 10-33 中显示的 IP 内部可达性信息 CLV 的格式是相同的, 但有一个例外, 就是它的类型代码是 130。但是, 它和 IP 内部可达性信息 CLV 不同, IP 外部可达性信息 CLV 只能用于 L2 类型的 LSP 报文。其中 I/E 位用来确定所有 4 种度量的类型——I/E=0 表示内部度量, 而 I/E=1 表示外部度量。

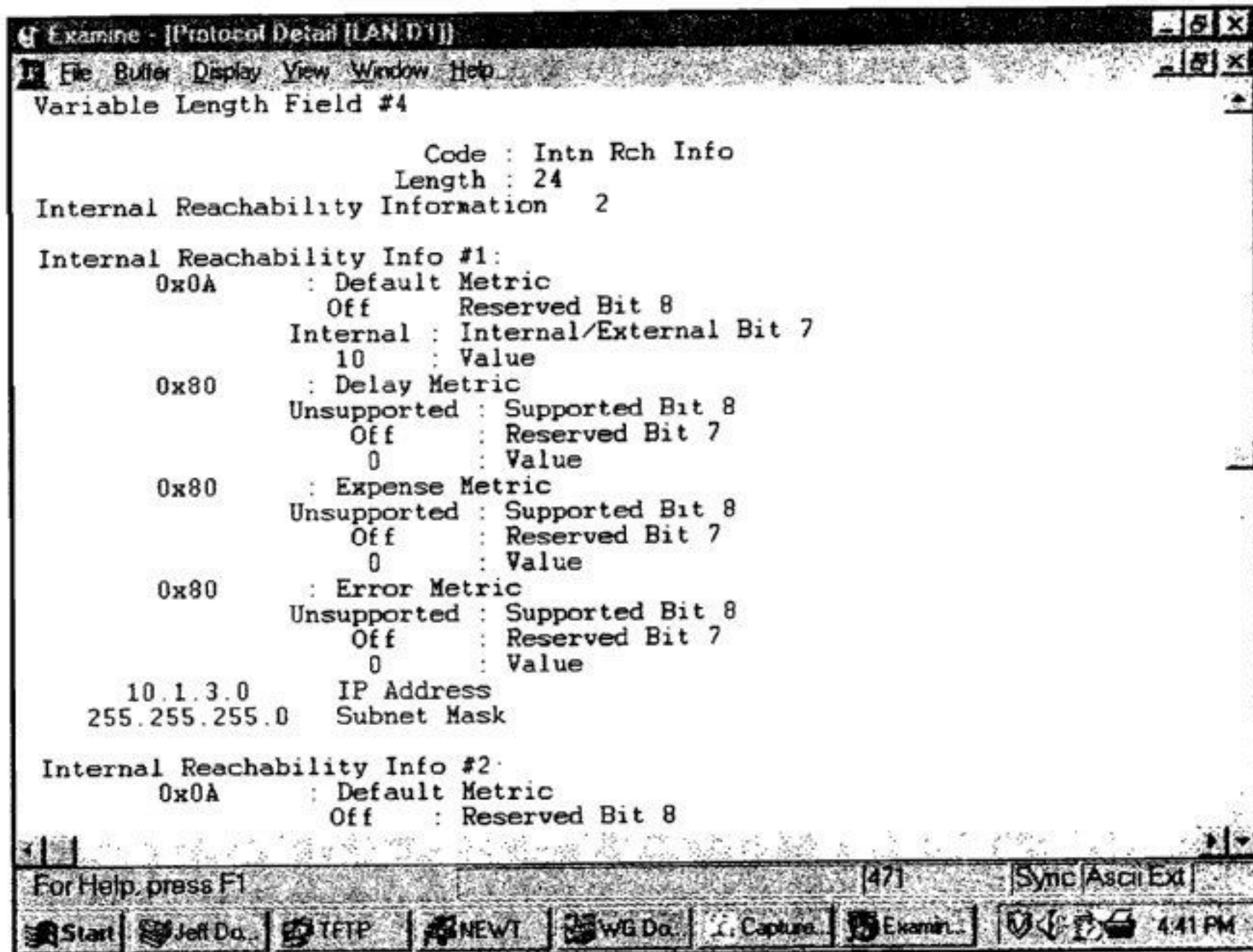


图 10-34 协议分析仪捕获到的一个 IP 内部可达性信息 CLV 的信息

(4) 域间路由选择协议信息 CLV

如图 10-35 所示, 域间路由选择协议信息 CLV 允许 L2 类型的 LSP 报文利用 IS-IS 域来透传来自于外部路由选择协议的信息。这个 CLV 字段具有一个相同的用途, 就是提供给 RIPv2、EIGRP 和 OSPF 协议报文的路由标签 (Route Tag) 字段信息使用。路由标签将在第 14 章“路由图”中介绍。

- **域间信息类型 (Inter-Domain Information Type)** ——指定了在可变长度的外部信息字段中包含的信息类型。如果该类型字段设置为 0x01, 那么外部信息就会使用本地域间路由选择协议的方式。第 14 章包含了一个使用路由图来设置这种本地信息的例子。如果该类型字段设置为 0x02, 那么外部信息就是一个 16 位的自主系统号, 用来标记所有后续的外部 IP 可达性条目, 直到 LSP 报文的结尾或者下一个域间路由选择协议信息 CLV 的出现。

	长度, 8bit 字节数
代 码 = 131	1
长 度	1
域间信息类型	1
外部信息	可变的

图 10-35 域间路由选择协议信息 CLV 的格式

4. IS-IS 协议序列号 PDU 报文

SNP 报文通过描述数据库中的部分或者全部 LSP 的信息，对 IS-IS 链路状态数据库进行维护。如图 10-36 所示，一台指定路由器将会周期性地以组播方式发送完全序列号报文 (CSNP) 来描述在伪节点的数据库当中的所有 LSP 信息。由于存在 L1 类型和 L2 类型的数据库，因此完全序列号报文也可能是 L1 类型或者 L2 类型的。有时，某些链路状态数据库的信息量太大，以至于 LSP 的信息无法使用单个完全序列号报文来描述。基于这个原因，完全序列号报文头部的最后两个字段作为起始 LSP ID (Start LSP ID) 字段和结束 LSP ID (End LSP ID) 字段，一起用来说明完全序列号报文中描述的 LSP 的范围。如图 10-37 所示，图中显示了这两个字段是怎样使用的。在这个完全序列号报文中将会描述完整的数据库信息，因此，LSP ID 的范围将开始于 0000.0000.0000.00.00，并结束于 ffff.ffff.ffff.ff.ff。如果需要两个完全序列号报文来描述这个数据库，那么第一个完全序列号报文的范围可以是 0000.0000.0000.00.00 ~ 0000.0c0a.1234.00.00，而第二个完全序列号报文的范围可以是 0000.0c0a.1235.00.00~ffff.ffff.ffff.ff.ff。

				长度, 8bit字节数
域内路由选择协议鉴别符				1
长度标识符				1
版本/协议标识扩展				1
标识长度				1
R	R	R	PDU类型	1
版 本				1
保 留				1
最大区域地址数				1
PDU长度				2
源标识				标识长度+1
起始LSP标识				标识长度+2
结束LSP标识				标识长度+2
可变长度字段				

图 10-36 IS-IS 协议完全序列号报文的格式

如图 10-38 所示，一个部分序列号报文除了像前面描述的只是携带部分 LSP 的信息，而不是整个数据库的信息外，其他与完全序列号报文都相似。因此，不必要像完全序列号报文那样需要起始和结束字段。一台路由器可以在一个点到点的子网上发送部分序列号报文来确认收到的 LSP 报文。在一个广播型的子网上，部分序列号报文将会用来请求丢失的或者最新的 LSP 报文。和完全序列号报文一样，部分序列号报文也存在 L1 类型和 L2 类型。

在 SNP 报文里面，不论是完全序列号报文还是部分序列号报文，也不管是 L1 类型还是 L2 类型的报文，它们都只用到了两个 CLV 字段：

- LSP 条目（类型 9）；

- 认证信息 (类型 10)。

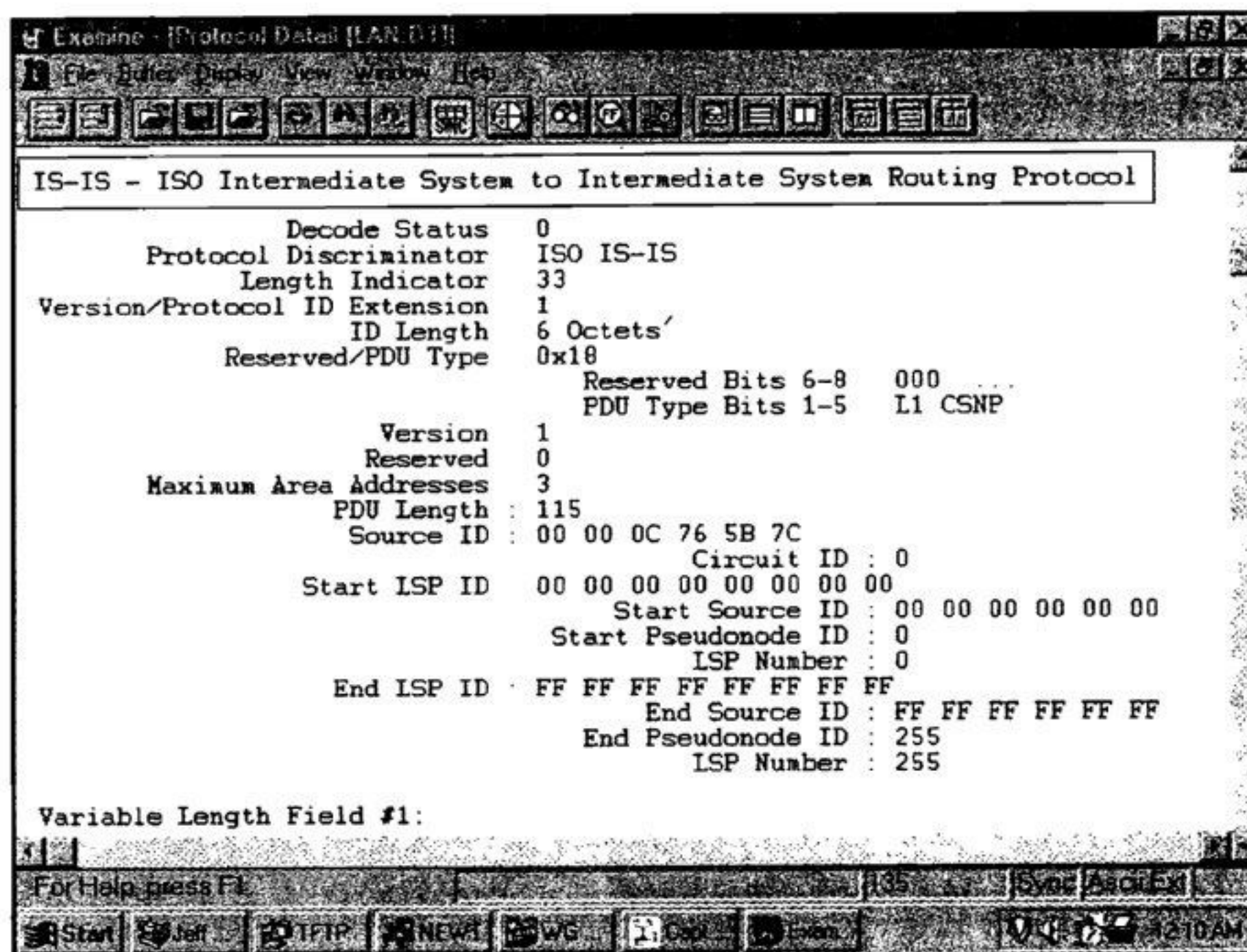


图 10-37 协议分析仪的捕获信息显示了一个 L1 类型的完全序列号报文头部信息

				长度, 8bit字节数
域内路由选择协议鉴别符				1
长度标识符				1
版本/协议标识扩展				1
标识长度				1
R	R	R	PDU类型	1
版 本				1
保 留				1
最大区域地址数				1
PDU长度				2
源标识				标识长度+1
可变长度字段				

图 10-38 IS-IS 协议部分序列号报文的格式

LSP 条目 CLV

如图 10-39 所示, LSP 条目 CLV 总结了一个 LSP 报文中列出的该 LSP 的剩余生存时间、LSP ID、序列号和校验和。这些字段信息不仅可以确定某个 LSP 报文, 而且可以完全地确定某个 LSP 的一个具体实例。如图 10-40 所示, 显示了一个 LSP 条目 CLV 的部分信息。

	长度, 8bit 字节数
代 码 = 9	1
长 度	1
剩余生存时间	2
LSP 标识	标识长度+2
LSP 序列号	4
校验和	2
多个字段	
剩余生存时间	2
LSP 标识	标识长度+2
LSP 序列号	4
校验和	2

图 10-39 LSP 条目 CLV 的格式

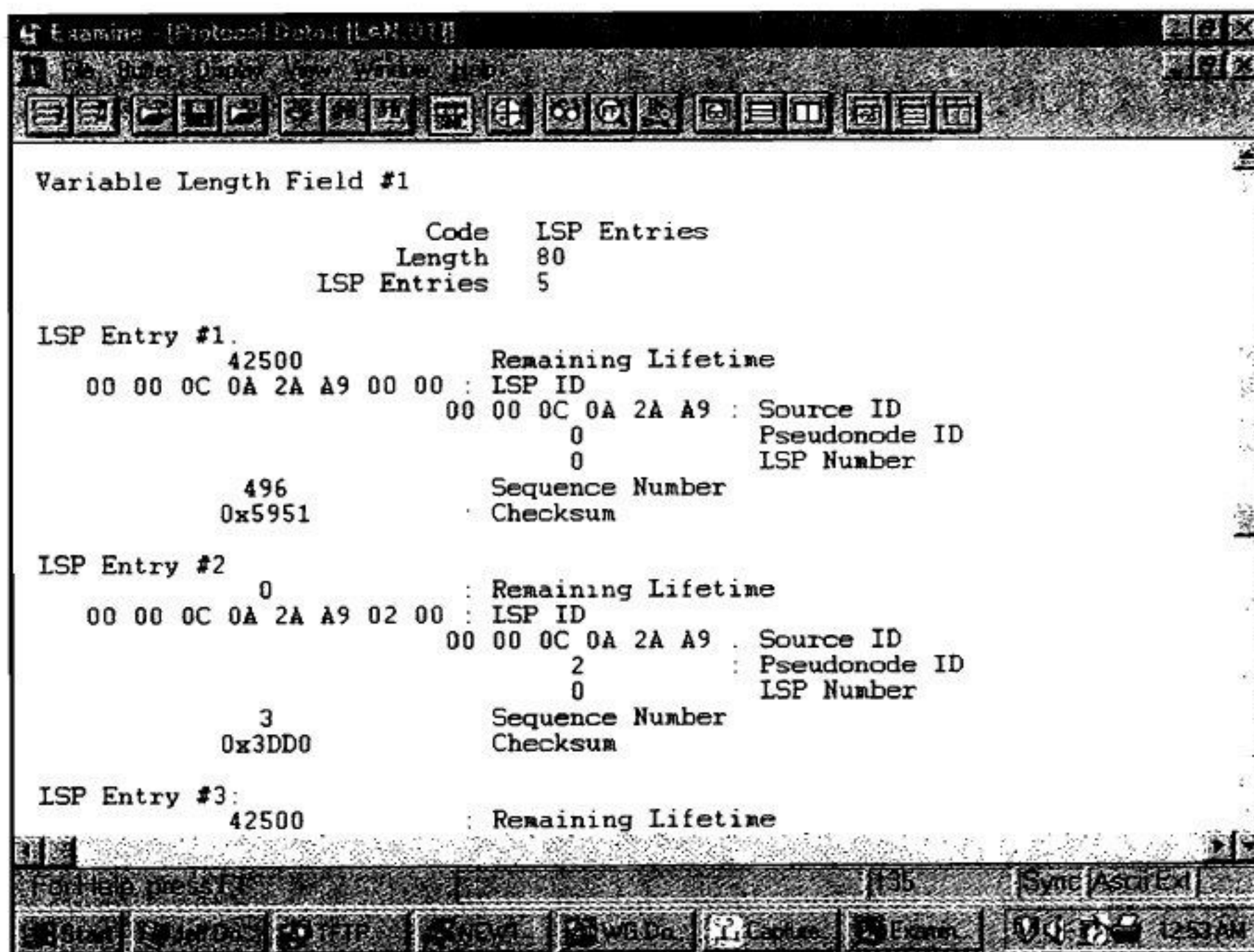


图 10-40 完全序列号报文的 LSP 条目 CLV 的部分信息, 这个完全序列号报文如图 10-37 所示

10.2 配置集成 IS-IS 协议

集成 IS-IS 协议在本书介绍的 IP 路由选择协议中显得比较独特, 这有两方面原因。首先, IS-IS 协议是惟一一个必须作为一个进程启动又要在单独的接口上启动的协议。其次, IS-IS 协议是惟一的一个开始并不是为 IP 协议设计的 IP 路由选择协议。由于集成 IS-IS 协议使用 CLNS PDU 报文而不是 IP 报文, 因此 IS-IS 协议的配置就不如其他路由选择协议的配置那么清楚直观了。

集成 IS-IS 协议作为一个 CLNS 协议的事实产生了一个有趣的影响——邻居路由器的 IP 地址对邻接关系的格式没有影响。结果, IS-IS 没有关于辅助 IP 地址的邻接关系的限制, 而

这在 OSPF 协议中是存在的。但是, 另一个结果是, 如果两个接口的 IP 地址属于完全不同的子网, 它们也会形成邻接关系。在这种情况下, IP 应该不会工作, 但事实上邻接关系会使链路处于“半阻断 (half broken)”状态, 这种情况在做故障排除时会引起一些困惑。

10.2.1 案例研究 1: 一个基本的集成 IS-IS 配置

在一台 Cisco 路由器上配置一个集成 IS-IS, 需要以下 4 个步骤:

步骤 1: 确定路由器所在的区域和启动 IS-IS 协议的接口;

步骤 2: 使用 **router isis** 命令来启动一个 IS-IS 进程;¹

步骤 3: 使用 **net** 命令来配置 NET 地址;

步骤 4: 使用命令 **ip router isis** 在相应的接口上启动集成 IS-IS。这个命令不仅在转发接口 (和 IS-IS 邻居相连的接口) 上必须增加, 而且在一个和末梢网络相连的接口也必须要配置, 这里的末梢网络是指需要 IS-IS 协议来通告的 IP 地址。

如图 10-41 所示, 显示了一个包括 6 台路由器的小型互连网络, 它被分成了两个区域。使用 NET 地址表示方式, 区域 1 和 2 将分别表示为 00.0001 和 00.0002, 而它们各自的系统 ID 是每台路由器 E0 或 TO0 接口的 MAC 地址标识符。表 10-4 中显示了使用该信息进行编码得到的 NET 地址。

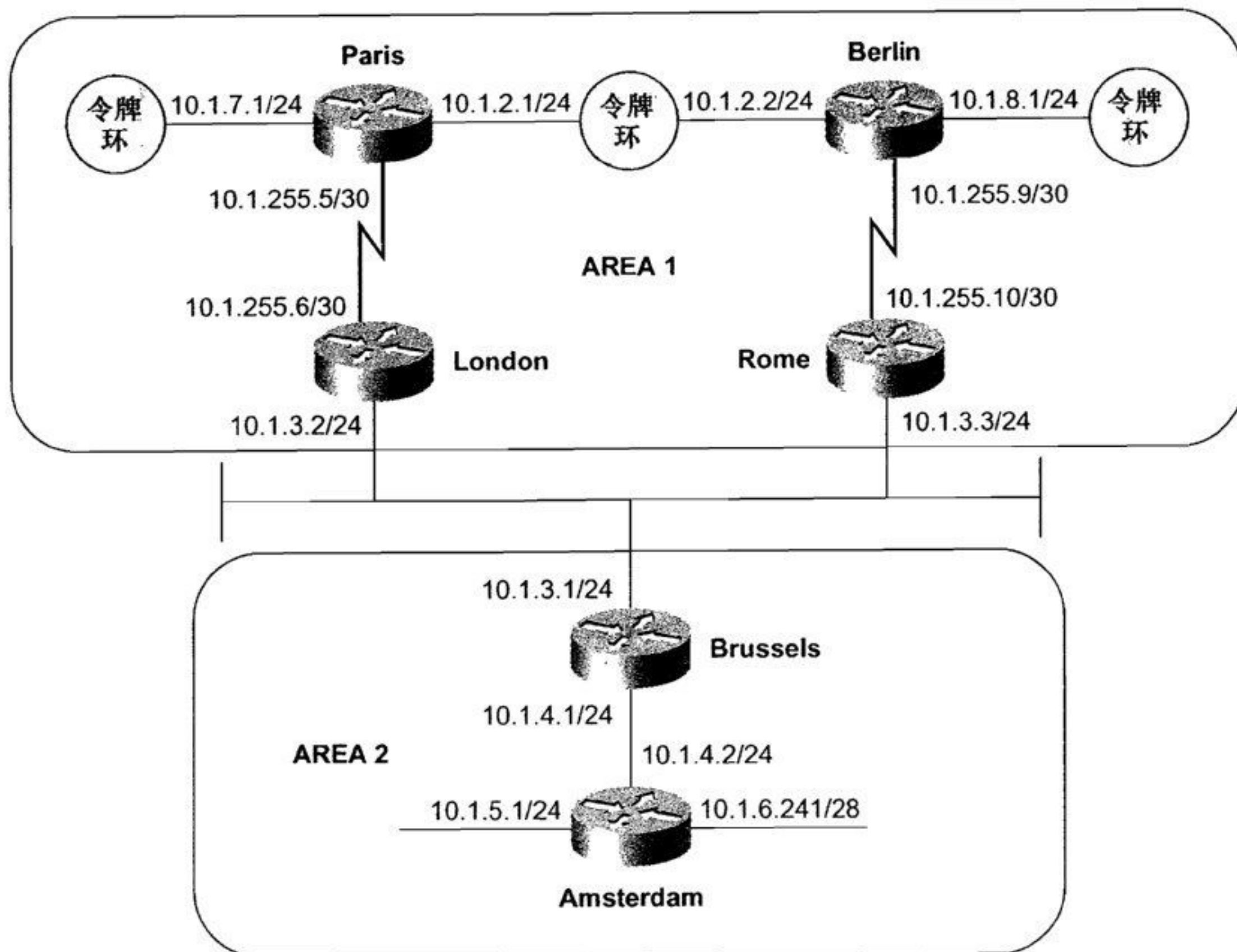


图 10-41 区域 1 表示成 NET 地址方式是 00.0001, 区域 2 表示成 NET 地址方式是 00.0002。

每一个 NET 的系统 ID 都是接口 E0 或 TO0 的 MAC 地址标识符

¹ **router isis** 命令也可以跟一个名字, 例如 **router isis Warsaw**。如果 IS-IS 和 ISO-IGRP 协议配置在同一台路由器上, 那么必须为其中一个进程或者同时为两个进程命名。如果没有配置 ISO-IGRP 协议, 则命名不是必须的。

表 10-4

图 10-41 中路由器 IS-IS 配置用到的 NET 地址

路 由 器	区 域	MAC	NET
Paris	00.0001	0000.3090.6756	00.0001.0000.3090.6756.00
Berlin	00.0001	0000.3090.c7df	00.0001.0000.3090.c7df.00
London	00.0001	0000.0c0a.2c51	00.0001.0000.0c0a.2c51.00
Rome	00.0001	0000.0c0a.2aa9	00.0001.0000.0c0a.2aa9.00
Brussels	00.0002	0000.0c76.5b7c	00.0002.0000.0c76.5b7c.00
Amsterdam	00.0002	0000.0c04.dcc0	00.0002.0000.0c04.dcc0.00

在路由器 Paris、London、Brussels 和 Amsterdam 上的配置如下：

路由器 Paris:

```

clns routing
!
interface Serial0
 ip address 10.1.255.5 255.255.255.252
 ip router isis
!
interface TokenRing0
 ip address 10.1.2.1 255.255.255.0
 ip router isis
 ring-speed 16
!
interface TokenRing1
 ip address 10.1.7.1 255.255.255.0
 ip router isis
 ring-speed 16
!
router isis
 net 00.0001.0000.3090.6756.00

```

路由器 London:

```

clns routing
!
interface Ethernet0
 ip address 10.1.3.2 255.255.255.0
 ip router isis
!
interface Serial0
 ip address 10.1.255.6 255.255.255.252
 ip router isis
!
router isis
 net 00.0001.0000.0c0a.2c51.00

```

路由器 Brussels:

```

clns routing
!
interface Ethernet0

```



```

ip address 10.1.3.1 255.255.255.0
ip router isis
!
interface Ethernet1
ip address 10.1.4.1 255.255.255.0
ip router isis
!
router isis
net 00.0002.0000.0c76.5b7c.00

```

路由器 Amsterdam:

```

clns routing
!
interface Ethernet0
ip address 10.1.4.2 255.255.255.0
ip router isis
!
interface Ethernet1
ip address 10.1.5.1 255.255.255.0
ip router isis
!
interface Ethernet2
ip address 10.1.6.241 255.255.255.240
ip router isis
!
router isis
net 00.0002.0000.0c04.dcc0.00

```

路由器 Berlin 和 Rome 的配置基本相似。在这里有一个需要注意的配置细节，就是在这些路由器的配置中都启动了 CLNS 协议的路由选择功能。CLNS 路由选择对于 IS-IS 协议处理 CLNS PDU 报文是必需的。但是，**clns routing** 命令在 IS-IS 路由选择的配置中却不是必要的步骤，这是因为在启动 IS-IS 协议进程的时候，路由器已经自动地加入了这个命令。

如图 10-42 所示，图中显示了路由器 Paris 的路由选择表信息。请注意，这里路由选择表中同时包含了 L1 路由和 L2 路由。在缺省条件下，Cisco 路由器默认是 L1/L2 路由器。这个事实也可以通过查看路由器的 IS 邻居表清楚地看出来（如图 10-43 所示）。

由于在图 10-41 的互连网络中的每一台路由器都是 L1/L2 类型的，因此，每一台路由器都会同时形成 L1 类型的邻接关系和 L2 类型的邻接关系。也正因为如此，每一台路由器也将会同时维护一个 L1 类型的链路状态数据库和一个 L2 类型的链路状态数据库。例如，图 10-44 中显示的路由器 Amsterdam 的链路状态数据库。其中，L1 类型的数据库中包含了一条始发于路由器 Amsterdam (0000.0c04.dcc0.00-00)¹ 的 LSP 报文和一条始发于路由器 Brussels (0000.0c76.5b7c.00-00) 的 LSP 报文。同时，它还包含了一条始发于路由器 Brussels 的伪节点 LSP 报文 (0000.0c76.5b7c.03-00)，用来描述路由器 Brussels 和 Amsterdam 之间的以太网链路。请记住，作为伪节点 LSP 报文的 LSP ID 是可以辨别出来的，因为伪节点 LSP 报文的 LSP ID 的倒数第二个 8bit 字节（也就是伪节点 ID）是非零的。

¹ 正如前面讲到的，这个 LSP ID 后面的星号表示该 LSP 是由这台路由器本身始发的。


```

Paris#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
       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

Gateway of last resort is not set

    10.0.0.0 is variably subnetted, 9 subnets, 3 masks
i L1   10.1.8.0 255.255.255.0 [115/20] via 10.1.2.2, TokenRing0
i L1   10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C      10.1.2.0 255.255.255.0 is directly connected, TokenRing0
C      10.1.7.0 255.255.255.0 is directly connected, TokenRing1
i L2   10.1.5.0 255.255.255.0 [115/40] via 10.1.255.6, Serial0
i L2   10.1.4.0 255.255.255.0 [115/30] via 10.1.255.6, Serial0
C      10.1.255.4 255.255.255.252 is directly connected, Serial0
i L1   10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
i L2   10.1.6.240 255.255.255.240 [115/40] via 10.1.255.6, Serial0
Paris#

```

图 10-42 路由器 Paris 的路由选择表中同时显示了 L1 路由和 L2 路由，表明这台路由器是一个 L1/L2 路由器

```

Berlin#show clns is-neighbors

System Id      Interface  State  Type Priority  Circuit Id      Format
0000.0C0A.2AA9 Se0        Up     L1L2 0 /0     03              Phase V
0000.3090.6756 To0        Up     L1L2 64/64    0000.3090.6756.04 Phase V
Berlin#

```

图 10-43 路由器 Berlin 的 IS 邻居表显示出路由器 Paris 和 Rome 都是 L1/L2 路由器

这 3 条 LSP 表明路由器 Amsterdam 实际上只有一个惟一的 L1 类型的邻接关系，也就是它与路由器 Brussels 形成的 L1 邻接关系。这个单一的邻接关系是可以预料到的，因为在区域 2 中路由器 Brussels 是惟一的其他路由器。对比表 10-4 中的系统 ID 来观察路由器 Amsterdam 的 L2 链路状态数据库，可以发现路由器 Amsterdam 和 IS-IS 域内的每一台路由器都形成了一条 L2 类型的邻接关系，当然这也是可以预料到的，因为每一台路由器都是一台 L1/L2 类型的路由器。

```

Amsterdam#show isis database
IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C04.DCC0.00-00* 0x00000025  0x3E6C        1078          0/0/0
0000.0C76.5B7C.00-00 0x00000023  0xD30E        1074          1/0/0
0000.0C76.5B7C.03-00 0x00000020  0x3F93        1074          0/0/0

IS-IS Level-2 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C04.DCC0.00-00* 0x0000005B  0x9A66        1080          0/0/0
0000.0C0A.2AA9.00-00 0x00000034  0xE971        371           0/0/0
0000.0C0A.2C51.00-00 0x00000031  0x732C        1135          0/0/0
0000.0C76.5B7C.00-00 0x0000002F  0xBEC6        1078          0/0/0
0000.0C76.5B7C.02-00 0x0000001F  0x3FC1        366           0/0/0
0000.0C76.5B7C.03-00 0x00000021  0xCC8D        1073          0/0/0
0000.3090.6756.00-00 0x0000002C  0xEF9F        365           0/0/0
0000.3090.6756.04-00 0x0000001D  0x1941        1143          0/0/0
0000.3090.C7DF.00-00 0x0000002D  0x4C01        359           0/0/0
Amsterdam#

```

图 10-44 路由器 Amsterdam 同时具有一个层 1 类型的链路状态数据库和一个层 2 类型的链路状态数据库，

这表明该路由器是一台 L1/L2 路由器

10.2.2 案例研究 2: 更改路由器的类型

在如图 10-41 这样的小型互连网络中, 在路由器上保留它们缺省的 IS-IS 类型是可以接受的。但是, 当网络规模不断增大时, 使用这些缺省的类型将越来越不能被接受。因为, 这不仅要消耗大量的路由器 CPU 和内存去处理和维持两个链路状态数据库, 而且要消耗大量的缓存和带宽去处理和泛洪每一台路由器始发的 L1 和 L2 类型的 IS-IS PDU 报文。

在图 10-41 中的路由器 Paris、Berlin 和 Amsterdam 可以配置成 L1 路由器, 因为它们都没有和其他区域直连的链路。在路由器上可以使用命令 **is-type** 来更改缺省的路由器类型。例如, 要把路由器 Berlin 变成一台 L1 类型的路由器, 只需做如下配置:

```
router isis
 net 00.0001.0000.3090.c7df.00
 is-type level-1
```

路由器 Paris 和 Amsterdam 的配置也和上面类似。比较一下路由器 Paris 在图 10-45 中的路由选择表和在图 10-42 中的路由选择表, 可以发现 L2 类型的路由已经被删除了。同样地, 比较一下路由器 Amsterdam 在图 10-46 和在图 10-44 中的路由选择表, 可以发现现在路由器 Amsterdam 只剩下 L1 类型的链路状态数据库了。

```
Paris#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
       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

Gateway of last resort is not set

10.0.0.0 is variably subnetted, 6 subnets, 2 masks
i L1   10.1.8.0 255.255.255.0 [115/20] via 10.1.2.2, TokenRing0
i L1   10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C      10.1.2.0 255.255.255.0 is directly connected, TokenRing0
C      10.1.7.0 255.255.255.0 is directly connected, TokenRing1
C      10.1.255.4 255.255.255.252 is directly connected, Serial0
i L1   10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
Paris#
```

图 10-45 当路由器 Paris 配置成一台 L1 路由器后, 它的路由选择表中就只包含到达它本身所在区域内的目的地址的路由了

```
Amsterdam#show isis database
IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C04.DCC0.00-00* 0x0000002C  0x2E77        726           0/0/0
0000.0C76.5B7C.00-00  0x0000002A  0xC515        733           1/0/0
0000.0C76.5B7C.03-00  0x00000026  0x3399        733           0/0/0
Amsterdam#
```

图 10-46 当路由器 Amsterdam 配置成一台 L1 路由器后, 它就只包含层 1 类型的链路状态数据库了

到目前为止, 在所显示的 L1 类型的配置, IP 路由选择的功能还是不完全的。回忆一下, 在前面曾经讲述过 LSP 报文中的区域关联位 (ATT 位), 一台 L1/L2 路由器会通过设置 ATT 位来告知 L1 路由器它具有区域间的连接。如图 10-47 所示, 图中显示出路由器 London 的 LSP

(0000.0c0a.2c51.00-00) 和路由器 Rome 的 LSP (0000.0c0a.2aa9.00-00) 都将 ATT 位设置为 1 了, 即 ATT=1。因而, 路由器 Paris 将会知道把区域间的通信量发送到路由器 London 或者 Rome。换句话说, 路由器 Paris 会有一条到达路由器 London 或 Rome 的缺省路由, 而且到达路由器 London 的路径将成为优先路径, 因为路由器 Paris 到达它的度量更小。但不幸的是, 在图 10-45 中, 并没有在路由器 Paris 的路由选择表中显示出这条缺省路由 (0.0.0.0)。

```
Paris#show isis database
IS-IS Level-1 Link State Database
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2AA9.00-00  0x0000000F   0x63B0        837            1/0/0
0000.0C0A.2C51.00-00  0x00000013   0x8922        784            1/0/0
0000.0C0A.2C51.01-00  0x0000000A   0x69D2        646            0/0/0
0000.3090.6756.00-00* 0x00000016   0x4A66        650            0/0/0
0000.3090.6756.04-00* 0x0000000E   0xA53D        864            0/0/0
0000.3090.C7DF.00-00  0x00000014   0x047E        1119           0/0/0
Paris#
```

图 10-47 路由器 London 和 Rome 始发的 L1 LSP 中设置了 ATT=1, 这表明它们具有到达其他区域的连接

在这里, 出现问题的原因是因为 ATT 位是一个 CLNS 的特性, IP 协议并不能直接理解该位。解决这个问题有两个方法。第一种解决方法是在路由器的接口上除了启用 IP 协议的 IS-IS, 另外再启用 CLNS 协议的 IS-IS。例如, 修改路由器 London 和 Paris 的串行接口的配置如下:

路由器 London:

```
interface Serial0
 ip address 10.1.255.6 255.255.255.252
 ip router isis
 clns router isis
```

路由器 Paris:

```
interface Serial0
 ip address 10.1.255.5 255.255.255.252
 ip router isis
 clns router isis
```

在图 10-48 中, 显示出路由器 Paris 现在有了一条指向路由器 London 的 IP 缺省路由, 并且可以成功地 ping 通区域间的目的地址。

```
Paris#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
       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

Gateway of last resort is 10.1.255.6 to network 0.0.0.0

10.0.0.0 is variably subnetted, 6 subnets, 2 masks
i L1   10.1.8.0 255.255.255.0 [115/20] via 10.1.2.2, TokenRing0
i L1   10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
```

待续


```

C      10.1.2.0 255.255.255.0 is directly connected, TokenRing0
C      10.1.7.0 255.255.255.0 is directly connected, TokenRing1
C      10.1.255.4 255.255.255.252 is directly connected, Serial0
i L1   10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
i*L1 0.0.0.0 0.0.0.0 [115/10] via 10.1.255.6, Serial0
Paris#ping 10.1.6.241
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.6.241, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/36/40 ms
Paris#

```

图 10-48 将路由器 London 和 Paris 配置成 CLNP/IP 混和路由器后, 路由器 Paris 就能够理解 ATT 位, 并且在它的路由选择表中增加一条缺省路由了

第一种解决方法需要 IS-IS 协议运行在一个 CLNP/IP 的混和环境里, 但是如果 IS-IS 协议仅仅是作为一个单一的 IP 路由选择协议使用的话, 那么启用 CLNS 路由选择仅仅为了生成一条 IP 缺省路由就显得十分没必要了。而第二种解决缺省路由问题的方法是在 L1/L2 路由器上配置一条静态路由, 并且使用命令 **default-information originate** 配置 IS-IS 协议来通告这条缺省路由。在图 10-41 的区域 2 中使用这个方法, 路由器 Brussels 的配置如下:

```

router isis
 net 00.0002.0000.0c76.5b7c.00
 default-information originate
!
 ip route 0.0.0.0 0.0.0.0 Null0

```

如图 10-49 所示, 路由器 Amsterdam 的路由选择表现在出现了一条缺省路由, 这条缺省路由是由路由器 Brussels 通告的, 这时也就可以成功 ping 通一个区域间的目的地址了。关于缺省路由和 **default-information originate** 命令更为详细的介绍将在第 12 章中讲述。

```

Amsterdam#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
       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

Gateway of last resort is 10.1.4.1 to network 0.0.0.0

10.0.0.0 is variably subnetted, 4 subnets, 2 masks
i L1   10.1.3.0 255.255.255.0 [115/20] via 10.1.4.1, Ethernet0
C      10.1.5.0 255.255.255.0 is directly connected, Ethernet1
C      10.1.4.0 255.255.255.0 is directly connected, Ethernet0
C      10.1.6.240 255.255.255.240 is directly connected, Ethernet2
i*L1 0.0.0.0 0.0.0.0 [115/10] via 10.1.4.1, Ethernet0
Amsterdam#ping 10.1.8.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.8.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/36/40 ms
Amsterdam#

```

图 10-49 路由器 Amsterdam 的路由选择表中包含了在路由器 Brussels 上以静态路由方式配置的缺省路由

10.2.3 案例研究 3: 区域的迁移

在 OSPF 协议中如果更改区域的地址, 就必须考虑和预计好网络中断的时间。但是, 在 IS-IS 协议的设计中, 能够在网络不中断的情况下允许更改区域地址。正如在“集成 IS-IS 的操作”一节中讲述的, Cisco 路由器可以最多配置 3 个区域地址。为了使两台路由器能够形成一个 L1 类型的邻接关系, 它们必须至少具有一个公用的区域地址。在允许具有多个区域地址的情况下, 新的邻接关系能够在旧的邻接关系中断时替代它。这种方法在某些情况下会显得非常有用。例如, 在合并区域或拆分区城的时候、在为一个区域重新编号的时候, 或者在同一个 IS-IS 域内同时运行多个编址机构分配的区域地址的时候等等。

举个例子, 在图 10-50 (a) 中的路由器都具有一个区域地址 01 (这些路由器当中的任何一个设备的 NET 地址看上去应该像 010000.0c12.3456.00 一样)。在图 10-50 (b) 中, 这些路由器另外分配了一个区域地址 03。虽然实际上并没有形成多个邻接关系, 但是这些路由器还是可以识别出它们具有多个公共的区域地址。在图 10-50 (c) 中, 区域 01 已经从某一个路由器上移走了。这 3 台路由器仍然保留着邻接关系, 因为它们至少还有一个公共的区域地址。最后, 在图 10-50 (d) 中, 区域地址 01 从这 3 台路由器上全部移走了, 这时, 这 3 台路由器就只在区域 03 中了。可以看出, 在区域迁移的期间并没有任何时候丢失邻接关系。

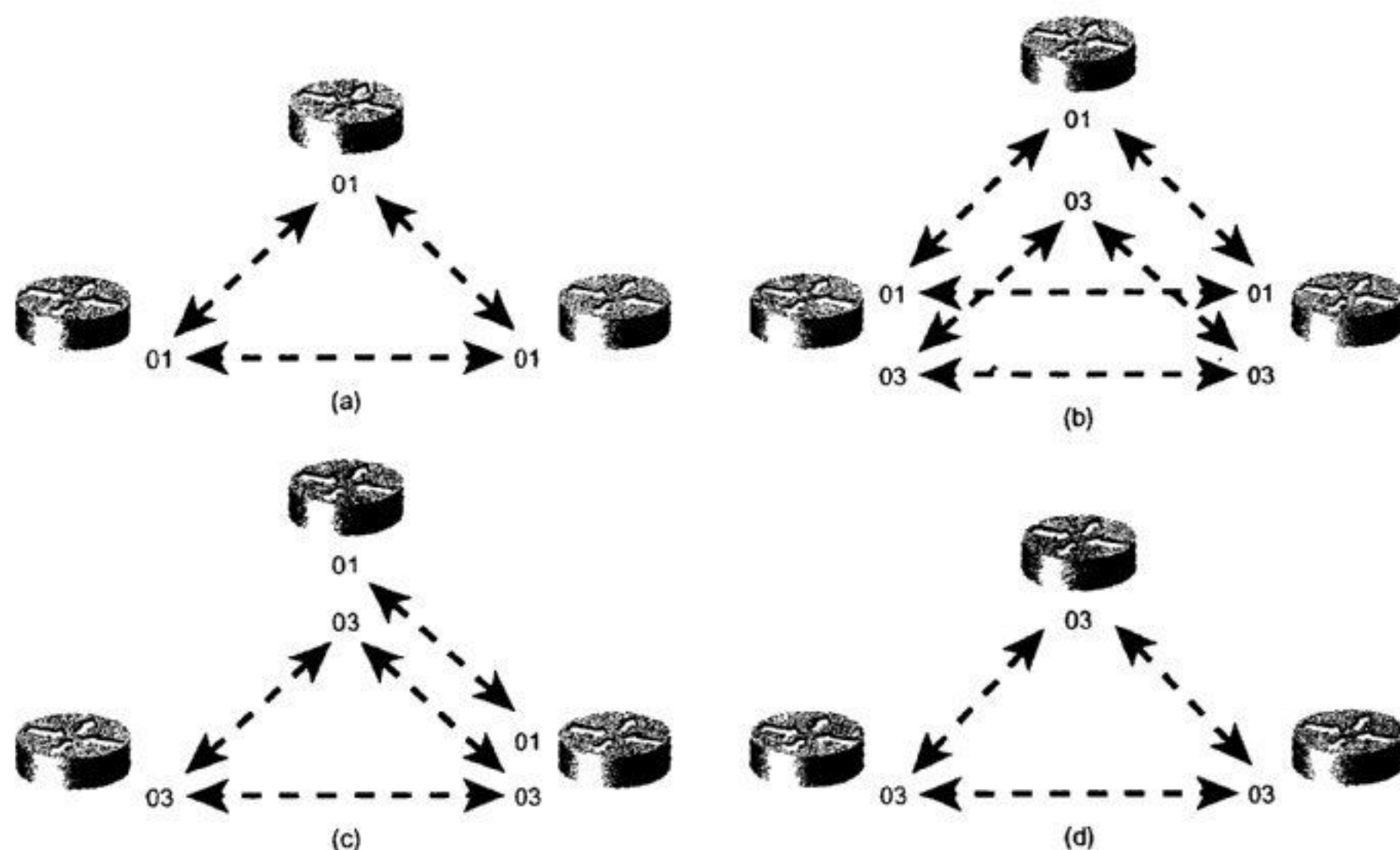


图 10-50 每台路由器都支持多个区域地址可以使区域的更改变得比较容易

假定因为某些原因强令, 认为图 10-41 中的网络正在使用的区域地址安排不合法, 而被要求必须要符合 GOSIP。这时通过注册 U.S.GSA 机构后, 可以使用下面的资源来组成新的 NET 地址:

```

AFI: 47
IDI: 0005
DFI: 80
AAI: 00ab7c
Reserved: 0000
RDI: fffe9

```


Areas: 0001 (area 1), 0002 (area 2)
根据上述信息, 表 10-5 中显示了新的 NET 地址。

表 10-5 图 10-41 中的路由器分配到了新的 GOSIP 格式的 NET 地址

路 由 器	NET 地址
Paris	47.0005.80.00ab7c.0000.ffe9.0001.0000.3090.6756.00
Berlin	47.0005.80.00ab7c.0000.ffe9.0001.0000.3090.c7df.00
London	47.0005.80.00ab7c.0000.ffe9.0001.0000.0c0a.2c51.00
Rome	47.0005.80.00ab7c.0000.ffe9.0001.0000.0c0a.2aa9.00
Brussels	47.0005.80.00ab7c.0000.ffe9.0002.0000.0c76.5b7c.00
Amsterdam	47.0005.80.00ab7c.0000.ffe9.0002.0000.0c04.dcc0.00

更改区域地址的第一步是增加新的 NET 地址到路由器上, 但是不改变原来的 NET 地址。
路由器 Rome 上的 IS-IS 配置如下:

```
router isis
net 00.0001.0000.0c0a.2aa9.00
net 47.0005.8000.ab7c.0000.ffe9.0001.0000.0c0a.2aa9.00
```

其他 5 台路由器的配置也和上面类似。可以使用带关键字 **detail** 的命令 **show isis database** 来查看结果 (如图 10-51 所示), 或者也可以使用命令 **show clns is-neighbors** 来查看结果 (如图 10-52 所示)。在这两个数据库中, 可以看出网络内的每一台路由器都是和多个区域相连的。

```
Rome#show isis database detail
IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2AA9.00-00* 0x00000059   0x705E        592           1/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.3.3
  Metric: 10 IP 10.1.3.0 255.255.255.0
  Metric: 10 IP 10.1.255.8 255.255.255.252
  Metric: 10 IS 0000.0C0A.2C51.01
  Metric: 10 IS 0000.3090.C7DF.00
  Metric: 0  ES 0000.0C0A.2AA9
0000.0C0A.2C51.00-00 0x00000059   0xD495        652           1/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.3.2
  Metric: 10 IP 10.1.3.0 255.255.255.0
  Metric: 10 IP 10.1.255.4 255.255.255.252
  Metric: 10 IS 0000.0C0A.2C51.01
  Metric: 10 IS 0000.3090.6756.00
  Metric: 0  ES 0000.0C0A.2C51
0000.0C0A.2C51.01-00 0x00000052   0xD81B        507           0/0/0
  Metric: 0  IS 0000.0C0A.2C51.00
  Metric: 0  IS 0000.0C0A.2AA9.00
0000.3090.6756.00-00 0x0000005C   0xDB0D        678           0/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.7.1
```

待续


```

Metric: 10 IP 10.1.7.0 255.255.255.0
Metric: 10 IP 10.1.255.4 255.255.255.252
Metric: 10 IP 10.1.2.0 255.255.255.0
Metric: 10 IS 0000.3090.6756.04
Metric: 10 IS 0000.0C0A.2C51.00
Metric: 0 ES 0000.3090.6756
0000.3090.6756.04-00 0x00000054 0x1983 835 0/0/0
Metric: 0 IS 0000.3090.6756.00
Metric: 0 IS 0000.3090.C7DF.00
0000.3090.C7DF.00-00 0x0000005B 0x18A5 545 0/0/0
Area Address: 00.0001
Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
--More--

```

图 10-51 在路由器 Rome 的链路状态数据库中, LSP 显示出图 10-41 中网络的所有路由器都正在通告两个区域地址

```
Rome#show clns is-neighbors detail
```

System Id	Interface	State	Type	Priority	Circuit Id	Format
0000.0C76.5B7C	Et0	Up	L2	64	0000.0C76.5B7C.02	Phase V
Area Address(es): 00.0002 47.0005.8000.ab7c.0000.ffe9.0002						
IP Address(es): 10.1.3.1						
Uptime: 0:27:22						
0000.0C0A.2C51	Et0	Up	L1L2	64/64	0000.0C0A.2C51.01	Phase V
Area Address(es): 00.0001 47.0005.8000.ab7c.0000.ffe9.0001						
IP Address(es): 10.1.3.2						
Uptime: 0:27:21						
0000.3090.C7DF	Se0	Up	L1	0	02	Phase V
Area Address(es): 00.0001 47.0005.8000.ab7c.0000.ffe9.0001						
IP Address(es): 10.1.255.9						
Uptime: 0:27:24						

Rome#

图 10-52 路由器 Rome 的 IS-IS 邻居表中也显示出每个邻居路由器是与多个地址相关联的

区域迁移的最后一步是从所有的路由器上删除原来的 NET 地址语句。例如, 在路由器 Rome 的 IS-IS 配置中输入命令 **no net 00.0001.0000.0c0a.2aa9.00**。在图 10-53 中, 显示了在从路由器 Rome 上删除了以前的 NET 语句后, 该路由器数据库中的一些 LSP 信息。

```

Rome#show isis data detail
IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2AA9.00-00* 0x00000069  0x02C4        809           1/0/0
Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
NLPID:         0x81 0xCC
IP Address:    10.1.3.3
Metric: 10 IP 10.1.3.0 255.255.255.0
Metric: 10 IP 10.1.255.8 255.255.255.252
Metric: 10 IS 0000.0C0A.2C51.01
Metric: 10 IS 0000.3090.C7DF.00
Metric: 0 ES 0000.0C0A.2AA9
0000.0C0A.2C51.00-00 0x0000006C  0x75E9        719           1/0/0
Area Address: 47.0005.8000.ab7c.0000.ffe9.0001

```

待续


```

NLPID:      0x81 0xCC
IP Address:  10.1.3.2
Metric: 10 IP 10.1.3.0 255.255.255.0
Metric: 10 IP 10.1.255.4 255.255.255.252
Metric: 10 IS 0000.0C0A.2C51.01
Metric: 10 IS 0000.3090.6756.00
Metric: 0 ES 0000.0C0A.2C51
0000.0C0A.2C51.01-00 0x0000005F 0xBE28 628 0/0/0
Metric: 0 IS 0000.0C0A.2C51.00
Metric: 0 IS 0000.0C0A.2AA9.00
0000.3090.6756.00-00 0x00000067 0x9936 896 0/0/0
Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
NLPID:      0x81 0xCC
IP Address:  10.1.7.1
Metric: 10 IP 10.1.7.0 255.255.255.0
Metric: 10 IP 10.1.255.4 255.255.255.252
Metric: 10 IP 10.1.2.0 255.255.255.0
Metric: 10 IS 0000.3090.6756.04
Metric: 10 IS 0000.3090.6756.05
Metric: 10 IS 0000.0C0A.2C51.00
Metric: 0 ES 0000.3090.6756
0000.3090.6756.04-00 0x0000005B 0x0B8A 730 0/0/0
Metric: 0 IS 0000.3090.6756.00
Metric: 0 IS 0000.3090.C7DF.00
0000.3090.6756.05-00 0x00000004 0xDF01 857 0/0/0
Metric: 0 IS 0000.3090.6756.00
0000.3090.C7DF.00-00 0x00000069 0xECC6 646 0/0/0
Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
NLPID:      0x81 0xCC
IP Address:  10.1.8.1
Metric: 10 IP 10.1.8.0 255.255.255.0
Metric: 10 IP 10.1.255.8 255.255.255.252
Metric: 10 IP 10.1.2.0 255.255.255.0
Metric: 10 IS 0000.3090.C7DF.05
--More--

```

图 10-53 路由器 Rome 的数据库中的 LSP 显示出只有一个区域地址

10.2.4 案例研究 4: 路由汇总

在第 9 章中, 已经介绍了在链路状态协议的区域之间如何进行路由汇总。关于路由汇总更完整的讲述将会在第 12 章讲述的缺省路由部分介绍。这里先简要地描述一下汇总路由的好处:

- 汇总路由可以有效地减小 LSP 报文的大小, 这样也就减小了链路状态数据库的大小, 从而也节省了路由器的 CPU 和内存消耗;
- 汇总路由可以隐藏掉区域内部网络的不稳定影响。如果仅仅是一个汇总地址范围内的地址发生了改变或一条链路的状态发生变化, 那么并不会通告到做汇总的区域外部;

当然，汇总路由也有一些不利之处，主要如下所述：

- 汇总路由的效果依赖于能够进行汇总的连续的 IP 地址范围，因此地址分配必须仔细规划；
- 汇总路由由于隐藏区域内的细节因而减少了路由的精确性。如果具有多条进入汇总区域的路径，那么将无法确定最佳的路径。

路由汇总可以在 IS-IS 的配置下使用命令 **summary-address** 来启动。配置了这条语句后，任何在汇总地址范围内的更具体的目的地址都将被抑制，而汇总路由的度量会选择它所有更具体的地址中更小的度量。

在图 10-54 中，显示了包含了 3 个区域的一个 IS-IS 网络。在这里，区域 1 内的地址可以汇总为 172.16.0.0/21，而区域 3 内的地址可以汇总为 172.16.16.0/21。路由器 Zurich、Madrid 和 Bonn 的配置如下：¹

路由器 Zurich:

```
router isis
net 01.0000.0c76.5b7c.00
summary-address 172.16.0.0 255.255.248.0
```

路由器 Madrid:

```
router isis
net 02.0000.3090.6756.00
is-type level-2-only
```

路由器 Bonn:

```
router isis
net 03.0000.0c0a.2aa9.00
summary-address 172.16.16.0 255.255.248.0
```

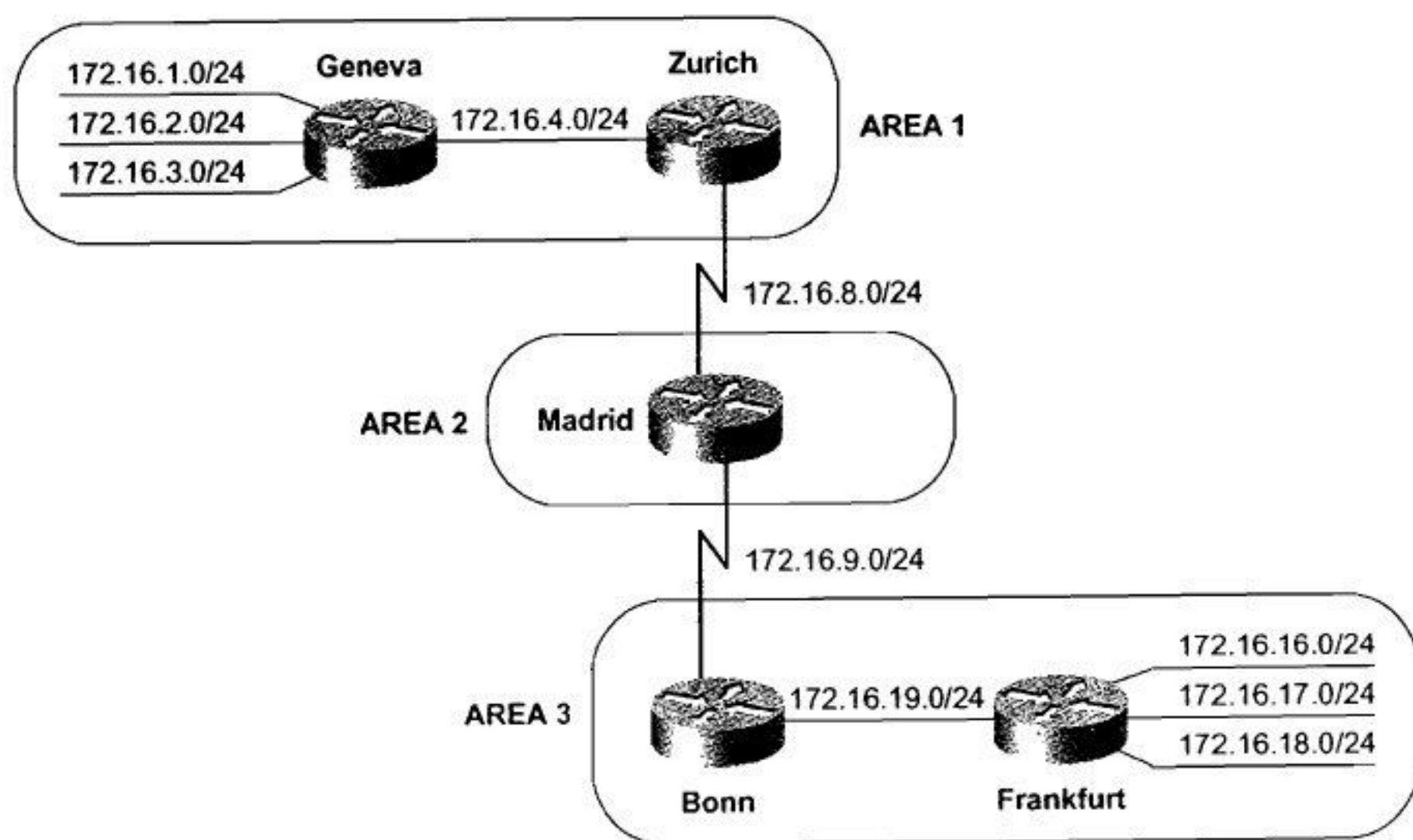


图 10-54 路由器 Zurich 和 Bonn 正在汇总区域 1 和区域 3 到区域 2

¹ 请注意，路由器 Madrid 由于没有 L1 类型的邻居，因而配置成一台 L2 路由器。

这里注意, 路由器 Madrid 由于没有 L1 类型的邻居, 因而配置成一台 L2 路由器。路由器 Zurich 和 Bonn 正在汇总它们各自的区域到层 2 类型的骨干。如图 10-55 所示, 在路由器 Madrid 的路由选择表中显示出了路由汇总的结果。

```
Madrid#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
       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

Gateway of last resort is not set

    172.16.0.0 is variably subnetted, 4 subnets, 2 masks
i L2   172.16.16.0 255.255.248.0 [115/20] via 172.16.9.2, Serial0
C      172.16.8.0 255.255.255.0 is directly connected, Serial1
C      172.16.9.0 255.255.255.0 is directly connected, Serial0
i L2   172.16.0.0 255.255.248.0 [115/20] via 172.16.8.2, Serial1
Madrid#
```

图 10-55 路由器 Madrid 的路由选择表显示出路由器 Bonn 和 Zurich 通告的汇总地址

10.2.5 案例研究 5: 认证

IS-IS 协议的认证仅限于明文口令。这种认证方式对于防止来自网络的攻击来说, 只是提供了一个很弱的安全机制, 但是对于防止因为配置出错或未受权的路由器造成网络服务的中断还是比较有效的。

Cisco IOS 软件支持 3 个级别的 IS-IS 认证: 邻居之间、区域范围和 IS-IS 域范围。这 3 个级别的认证可以单独使用也可以一起使用。IS-IS 认证的规则如下:

- 当认证配置在邻居之间时, 互相连接的路由器接口必须配置相同的口令;
- 当认证配置在邻居之间时, 必须分别为 L1 和 L2 类型的邻接关系配置各自的认证;
- 当认证在整个区域范围内有效时, 区域内的每一台路由器都必须执行认证, 并且必须使用相同的口令;
- 当认证在整个 IS-IS 域范围内有效时, IS-IS 域内的每一个 L2 和 L1/L2 类型的路由器都必须执行认证, 并且必须使用相同的口令。

如果要在两个邻居之间配置认证, 可以在这两个邻居相连的接口上使用命令 **isis password** 来配置一个口令。这个命令将会指定一个口令, 并且指定这个口令是 L1 类型的邻接关系使用, 还是 L2 类型的邻接关系使用。在一个接口上可以指定任何一层或两层类型的口令, 并且每一层类型的口令可以相同也可以不同。当配置了认证信息后, IS-IS 邻居之间将会在 L1 或 L2 类型的 Hello 报文中的认证信息 CLV 里携带所配置的口令。

举个例子, 在图 10-54 中的路由器 Geneva、Zurich 和 Madrid 上配置了口令认证, 这 3 台路由器的相关配置如下:

路由器 Geneva:

```
interface Ethernet0
ip address 172.16.4.1 255.255.255.0
ip router isis
```



```
isis password Alps level-1
```

路由器 Zurich:

```
interface Ethernet0
  ip address 172.16.4.2 255.255.255.0
  ip router isis
  isis password Alps level-1
!
interface Serial0
  ip address 172.16.8.2 255.255.255.0
  ip router isis
  isis password Pyrenees level-2
```

路由器 Madrid:

```
interface Serial1
  ip address 172.16.8.1 255.255.255.0
  ip router isis
  isis password Pyrenees level-2
```

因为路由器 Geneva 和 Zurich 之间的邻接关系是 L1 类型的, 因此只指定了一个层 1 类型的口令 (Alps)。路由器 Zurich 和 Madrid 之间只存在 L2 类型的邻接关系, 因此只指定了一个层 2 类型的口令 (Pyrenees)。这里要注意, 如果没有指定关键字 **level-1** 或者 **level-2**, 那么命令 **isis password** 将会默认为是层 1 类型。

如果要在一个区域内进行认证, 可以在 IS-IS 配置下面使用命令 **area-password** 来指定一个口令。然而, **isis password** 命令指定的口令是在 Hello 报文传送的, 而 **area-password** 命令指定的口令是在所有的 L1 LSP 报文、完全序列号报文 (CSNP) 和部分序列号报文 (PSNP) 中传送的。因此, 邻居级别的口令只可以用来验证邻接关系的建立, 而区域级别的口令则可以用来验证层 1 类型的链路状态信息的交换。如果区域认证没有正确, 路由器将仍然可以形成邻接关系, 但是不会进行 L1 LSP 报文的交换。

在图 10-54 中的区域 3 上配置区域口令, 路由器 Bonn 和 Frankfurt 的配置如下:

路由器 Bonn:

```
router isis
  net 03.0000.0c0a.2aa9.00
  area-password Rhine
  summary-address 172.16.16.0 255.255.248.0
```

路由器 Frankfurt:

```
router isis
  net 03.0000.0c04.dcc0.00
  is-type level-1
  area-password Rhine
```

如果要在 IS-IS 域范围内配置认证信息, 可以使用命令 **domain-password** 来指定一个口令。这个指定的口令将在所有的 L2 LSP 报文、完全序列号报文 (CSNP) 和部分序列号报文 (PSNP) 中传送。因而, IS-IS 域认证可以用来验证层 2 类型路由信息的交换。像区域认证一样, IS-IS 域认证不会去验证 L2 类型的邻接关系, 但是会验证 L2 LSP 报文的

交换。

在图 10-54 中的网络上配置 IS-IS 域认证, 只需要在路由器 Zurich、Madrid 和 Bonn 上配置就可以了, 因为路由器 Geneva 和 Frankfurt 是 L1 类型的路由器。相关的配置如下:

路由器 Zurich:

```
router isis
 net 01.0000.0c76.5b7c.00
 domain-password BlackForest
 area-password Switzerland
 summary-address 172.16.0.0 255.255.248.0
```

路由器 Madrid:

```
router isis
 net 02.0000.3090.6756.00
 is-type level-2-only
 domain-password BlackForest
```

路由器 Bonn:

```
router isis
 net 03.0000.0c0a.2aa9.00
 domain-password Blackforest
 area-password Rhine
 summary-address 172.16.16.0 255.255.248.0
```

10.3 集成 IS-IS 协议的故障排除

IS-IS 协议故障排除的基本方法和第 9 章讲述的 OSPF 协议的故障排除方法十分相似。集成 IS-IS 协议和其他 IP 路由选择协议的故障排除相比, 一个主要的不同之处是 IS-IS 协议使用的是 CLNS PDU 报文, 而不是 IP 报文。如果你是在对协议本身做故障排除的话, 记住你是在做 CLNS 协议的故障排除, 而不是 IP 协议。

正如所有的路由选择协议一样, 故障排除的第一步是检查路由选择表来获取精确的信息。如果一个预期的路由条目在路由选择表中丢失了或变得不正确, 那么故障排除剩下来的任务就是确定引起故障的源头了。

在检查过路由选择表之后, 查看链路状态数据库就是获取故障排除信息的一个最重要的来源。正如在第 9 章中所建议的, 一个比较好的实际经验是为每一个区域保存一份 L1 类型的链路状态数据库拷贝, 并保存一份 L2 类型的链路状态数据库的拷贝。这些保存的数据库拷贝应该有规律地进行更新, 并作为日常工作的一部分。这样在网络出现问题或错误时, 这些保存的数据库拷贝将可以提供一个稳定状态的参考。在检查一个单独的路由器配置时, 可以考虑以下问题:

- 在 IS-IS 协议配置下面的 **net** 语句是否指定了正确的 NET 地址? 在该路由器上配置的区域 ID 和系统 ID 是否正确无误? 配置的 NET 地址是否符合所在网络上正在使用的 CLNS 编址约定?
- 是否在正确的接口上使用命令 **ip router isis** 启动 IS-IS 协议?

- IP 地址和子网掩码配置是否正确？在一个集成 IS-IS 环境里检查这些配置显得加倍重要，因为配置错误的 IP 地址不会妨碍建立一个 IS-IS 邻接关系。

10.3.1 IS-IS 邻接关系的故障排除

使用命令 **show clns is-neighbors** 可以显示 IS-IS 的邻居表。缺省条件下显示的是整张邻居表，当然也可以指定显示一个具体接口的邻居表。从这个表中，我们可以看出所有预期的邻居是否都出现了？并且它们是否是正确的类型？为了获取更详细的信息，可以使用命令 **show clns is-neighbors detail** 来显示像与每一个邻居相关联的区域地址和 IP 地址，以及每一个邻居的上线时间等等。

在检查邻接关系时，可以考虑以下问题：

- 路由器的层（level）是否配置正确？L1 路由器只能和 L1 与 L1/L2 类型的路由器建立邻接关系，而 L2 路由器只能和 L2 与 L1/L2 类型的路由器建立邻接关系；
- 是否这两个邻居路由器都正在发送 Hello 报文？Hello 报文的层（level）是否正确？它们的 Hello 报文包含的参数是否正确？调试命令 **debug isis adj-pachets** 是查看 Hello 报文的一个比较有用的命令，如图 10-56 所示；
- 在邻居之间通过 **isis hello-interval** 和 **isis hello-multiplier** 命令设置的值是否相同？
- 如果使用了认证，那么在邻居之间的口令是否相同？记住区域（层 1）和域（层 2）认证是验证邻接关系的，它们只验证 LSP 报文的交换；
- 是否存在任何阻塞 IS-IS 或者 CLNS 协议的访问列表？

```
Bonn#debug isis adj-packets
IS-IS Adjacency related packets debugging is on
Bonn#
ISIS-Adj: Sending serial IIH on Serial0
ISIS-Adj: Rec L1 IIH from 0000.0c04.dcc0 (Ethernet0), cir type 1, cir id 0000.0C
0A.2AA9.02
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Rec serial IIH from *HDLC* on Serial0, cir type 2, cir id 02
ISIS-Adj: rcvd state 0, old state 0, new state 0
ISIS-Adj: Action = 2, new_type = 0
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Sending L2 IIH on Ethernet0
ISIS-Adj: Sending serial IIH on Serial0
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Rec serial IIH from *HDLC* on Serial0, cir type 2, cir id 02
ISIS-Adj: rcvd state 0, old state 0, new state 0
ISIS-Adj: Action = 2, new_type = 0
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Rec L1 IIH from 0000.0c04.dcc0 (Etheir type 1, cir id 0000.0C0A.2AA9.0
2
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Sending L2 IIH on Ethernet0
ISIS-Adj: Sending serial IIH on Serial0
```

图 10-56 可以使用命令 **debug isis adj-packets** 来查看 IS-IS Hello（IIHs）的详细信息。

图中显示的是图 10-54 中路由器 Bonn 的信息

10.3.2 IS-IS 链路状态数据库的故障排除

IS-IS 链路状态数据库的信息可以通过命令 **show isis database** 来查看。如果一台路由器是 L1/L2 路由器, 那么缺省情况下将会同时显示 L1 和 L2 类型的数据库。如果只需要查看其中一个数据库, 可以使用 **level-1** 或 **level-2** 关键字。如果需要查看 LSP 更详尽的信息, 可以使用 **detail** 关键字。如果指定一个 LSP ID, 也可以查看单个 LSP 的信息, 如图 10-57 所示。

```
Zurich#show isis database detail 0000.3090.6756.00-00

IS-IS Level-2 LSP 0000.3090.6756.00-00
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.3090.6756.00-00 0x00000080   0x9EA1        480           0/0/0
Auth:                Length: 12
Area Address: 02
NLPID:               0xCC
IP Address: 172.16.8.1
Metric: 10 IS 0000.0C76.5B7C.00
Metric: 10 IS 0000.0C0A.2AA9.00
Metric: 10 IP 172.16.8.0 255.255.255.0
Metric: 10 IP 172.16.9.0 255.255.255.0
Zurich#
```

图 10-57 这个 LSP 来自于图 10-54 中路由器 Zurich 的 L2 类型的数据库

如果一个序列号显著地高于其他 LSP 的序列号, 就可能表明是这个区域不稳定或者是层 2 类型的骨干不稳定。另一个网络不稳定的提示是某个 LSP 的抑制时间从来不会变得很小。如果怀疑网络不稳定, 可以使用命令 **show isis spf-log** 列出这台路由器上最近执行的所有 SPF 计算。

在图 10-58 中, 显示了图 10-54 中路由器 Geneva 的 SPF 日志。在显示这个日志大约 3min 后, 除了每隔 15min 由数据库重新刷新触发的周期性的 SPF 计算, 并没有显示什么内容。在那个时间, 频繁的 SPF 计算开始产生, 这表明网络发生了频繁的变化。¹

```
Geneva#sh isis spf-log

Level 1 SPF log
When    Duration  Nodes  Count  Last trigger LSP  Triggers
02:43:09    12      3      1
02:28:08    12      3      1
02:13:06    12      3      1
01:58:05    12      3      1
01:43:03    12      3      1
01:28:02    12      3      1
01:13:00    12      3      1
00:57:59    12      3      1
00:42:58    12      3      1
```

待续

¹ 开始的 4 个触发事件是由路由器 Zurich 的串行接口几次状态变化引起的, 接下来的 3 个事件是由于删除和接着错误配置了链路命令引起的, 最后一个事件是在配置了正确的口令时引起的。

00:27:56	12	3	1		PERIODIC
00:12:55	12	3	1		PERIODIC
00:03:08	8	3	1	0000.0C76.5B7C.00-00	LSPHEADER
00:02:35	8	3	1	0000.0C76.5B7C.00-00	LSPHEADER
00:02:23	8	3	1	0000.0C76.5B7C.00-00	LSPHEADER
00:01:50	8	3	1	0000.0C76.5B7C.00-00	LSPHEADER
00:01:14	4	1	1	0000.0C0A.2C51.00-00	TLVCONTENT
00:00:46	4	2	2	0000.0C0A.2C51.04-00	NEWLSP TLVCONTENT
00:00:20	4	1	3	0000.0C0A.2C51.00-00	NEWADJ TLVCONTENT
00:00:08	8	3	1	0000.0C76.5B7C.02-00	TLVCONTENT

Geneva#

图 10-58 这个 SPF 的 log 记录揭示了图 10-54 中区域 area 1 的不稳定性

为了进一步跟踪 SPF 日志暴露的网络震荡问题, 还有 3 个有用的调试命令可以使用。图 10-59、图 10-60 和图 10-61 中显示了这 3 个调试命令的输出结果。在每一个结果中, 从路由器 Geneva 的角度来看, 调试信息都显示了图 10-54 中路由器 Zurich 的串行接口断开和重新连接的结果。首先, **debug isis spf-triggers** 命令显示了与触发一个 SPF 计算有关的事件消息, 如图 10-59 所示。第二个命令是 **debug isis spf-events**, 这个命令显示了触发事件引起 SPF 计算的一个详细报告, 如图 10-60 所示。第三个命令是 **debug isis spf-statistics**, 它显示了有关 SPF 计算本身的信息, 如图 10-61 所示。这里值得特别注意的是, 这次进行的是完全的计算, 这可能给路由器带来性能问题。

```
Geneva#debug isis spf-triggers
IS-IS SPF triggering events debugging is on
Geneva#
ISIS-SPF-TRIG: L1, LSP fields changed 0000.0C76.5B7C.00-00
ISIS-SPF-TRIG: L1, LSP fields changed 0000.0C76.5B7C.00-00
Geneva#
```

图 10-59 debug isis spf-triggers 命令显示了触发一个 SPF 计算的事件消息

```
Geneva#debug isis spf-events
IS-IS SPF events debugging is on
Geneva#
ISIS-SPF: L1 LSP 3 (0000.0C76.5B7C.00-00) flagged for recalculation
from 34F561A
ISIS-SPF: Calculating routes for L1 LSP 3 (0000.0C76.5B7C.00-00)
ISIS-SPF: Add 172.16.4.0/255.255.255.0 to IP route table, metric 20
ISIS-SPF: Next hop 0000.0C76.5B7C/172.16.4.2 (Ethernet0) (rejected)
ISIS-SPF: Add 0000.0C76.5B7C to L1 route table, metric 10
ISIS-SPF: Next hop 0000.0C76.5B7C (Ethernet0)
ISIS-SPF: Aging L1 LSP 3 (0000.0C76.5B7C.00-00), version 132
ISIS-SPF: Aging IP 172.16.8.0/255.255.255.0, next hop 172.16.4.2
ISIS-SPF: Deleted NDB
ISIS-SPF: Compute L1 SPT
ISIS-SPF: Move 0000.0C0A.2C51.00-00 to PATHS, metric 0
ISIS-SPF: thru 2147483647/2147483647/2147483647, delay 0/0/0, mtu 2147483647/214
7483647/2147483647, hops 0/0/0, ticks 0/0/0
ISIS-SPF: Add 0000.0C76.5B7C.02-00 to TENT, metric 10
ISIS-SPF: Next hop local
ISIS-SPF: Add 0000.0C0A.2C51 to L1 route table, metric 0
```

待续


```

ISIS-SPF: Move 0000.0C76.5B7C.02-00 to PATHS, metric 10
ISIS-SPF: thru 2147483647/2147483647/2147483647, delay 0/0/0, mtu 2147483647/214
7483647/2147483647, hops 0/0/0, ticks 0/0/0
ISIS-SPF: considering adj to 0000.0C76.5B7C (Ethernet0) metric 10
ISIS-SPF: (accepted)
ISIS-SPF: Add 0000.0C76.5B7C.00-00 to TENT, metric 10
ISIS-SPF: Next hop 0000.0C76.5B7C (Ethernet0)
ISIS-SPF: Move 0000.0C76.5B7C.00-00 to PATHS, metric 10
ISIS-SPF: Add 172.16.4.0/255.255.255.0 to IP route table, metric 20
ISIS-SPF: N0C76.5B7C/172.16.4.2 (Ethernet0) (rejected)
ISIS-SPF: Add 0000.0C76.5B7C to L1 route table, metric 10
ISIS-SPF: Next hop 0000.0C76.5B7C (Ethernet0)
ISIS-SPF: Aging L1 LSP 1 (0000.0C0A.2C51.00-00), version 126
ISIS-SPF: Aging L1 LSP 2 (0000.0C76.5B7C.02-00), version 127
ISIS-SPF: Aging L1 LSP 3 (0000.0C76.5B7C.00-00), version 133

```

图 10-60 debug isis spf-events 显示了一个 SPF 计算的详细信息

```

Geneva#debug isis spf-statistics
IS-IS SPF Timing and Statistics Data debugging is on
Geneva#
ISIS-Stats: Compute L1 SPT
ISIS-Stats: Complete L1 SPT, Compute time 0.008, 3 nodes, 2 links on SPT, 0 suspends
ISIS-Stats: Compute L1 SPT
ISIS-Stats: Complete L1 SPT, Compute time 0.008, 3 nodes, 2 links on SPT, 0 suspends

```

图 10-61 debug isis spf-statistics 显示了有关 SPF 计算本身的统计信息

在一个区域内的每一台路由器都必须维护一个同样的链路状态数据库。另外，IS-IS 域内的每一个 L1/L2 和 L2 路由器都必须维护一个同样的 L2 类型的数据库。如果你怀疑某台路由器的链路状态数据库不能正确同步，可以检查它的 LSP ID 和它的校验和。相同的 LSP ID 应该存在于每一个数据库中，并且在每一个数据库中的每一个 LSP 的校验和也都应该相同。

有两个调试命令可以帮助我们查看数据库的同步处理过程。第一个命令是 **debug isis update-packets**，它显示了路由器接收和发送 SNP 与 LSP 报文的有关信息，如图 10-62 所示。第二个命令是 **debug isis snp-packets**，它显示了路由器接收和发送某个指定的 CSNP 与 PSNP 报文的有关信息，如图 10-63 所示。

```

Geneva#debug isis update-packets
IS-IS Update related packet debugging is on
Geneva#
ISIS-Update: Rec L1 LSP 0000.0C76.5B7C.00-00, seq A7, ht 1199,
ISIS-Update: from SNPA 0000.0c76.5b7c (Ethernet0)
ISIS-Update: LSP newer than database copy
ISIS-Update: Important fields changed
ISIS-Update: Populating FastPSNP cache (index 245 lspix 3 chksm EF1A)
ISIS-Update: Full SPF required
ISIS-SNP: Rec L1 CSNP from 0000.0C76.5B7C (Ethernet0)
ISIS-SNP: Rec L1 CSNP from 0000.0C76.5B7C (Ethernet0)
Geneva#

```

图 10-62 debug isis update-packets 显示了路由器接收和发送 SNP 与 LSP 报文的有关信息


```

Geneva#debug isis snp-packets
IS-IS CSNP/PSNP packets debugging is on
Geneva#
ISIS-SNP: Rec L1 CSNP from 0000.0C76.5B7C (Ethernet0)
ISIS-SNP: CSNP range 0000.0000.0000.00-00 to FFFF.FFFF.FFFF.FF-FF
ISIS-SNP: Same entry 0000.0C0A.2C51.00-00, seq 82
ISIS-SNP: Same entry 0000.0C76.5B7C.00-00, seq A7
ISIS-SNP: Same entry 0000.0C76.5B7C.02-00, seq 65
ISIS-SNP: Rec L1 CSNP from 0000.0C76.5B7C (Ethernet0)
ISIS-SNP: CSNP range 0000.0000.0000.00-00 to FFFF.FFFF.FFFF.FF-FF
ISIS-SNP: Same entry 0000.0C0A.2C51.00-00, seq 82
ISIS-SNP: Entry 0000.0C76.5B7C.00-00, seq AD is newer than ours (seq A8), sending PSNP
ISIS-SNP: Same entry 0000.0C76.5B7C.02-00, seq 65
ISIS-SNP: Rec L1 CSNP from 0000.0C76.5B7C (Ethernet0)
ISIS-SNP: CSNP range 0000.0000.0000.00-00 to FFFF.FFFF.FFFF.FF-FF
ISIS-SNP: Same entry 0000.0C0A.2C51.00-00, seq 82
ISIS-SNP: Same entry 0000.0C76.5B7C.00-00, seq AE
ISIS-SNP: Same entry 0000.0C76.5B7C.02-00, seq 65
ISIS-SNP: Rec L1 CSNP from 0000.0C76.5B7C (Ethernet0)
ISIS-SNP: CSNP range 0000.0000.0000.00-00 to FFFF.FFFF.FFFF.FF-FF
ISIS-SNP: Same entry 0000.0C0A.2C51.00-00, seq 82
ISIS-SNP: Same entry 0000.0C76.5B7C.00-00, seq AE
ISIS-SNP: Same entry 0000.0C76.5B7C.02-00, seq 65

```

图 10-63 debug isis snp-packets 路由器接收和发送 CSNP 与 PSNP 报文的有关详细信息

10.3.3 案例研究 6: 运行于 NBMA 网络上的集成 IS-IS

在图 10-64 中, 显示了 4 台运行 IS-IS 协议的路由器, 它们之间通过一个部分网状连接的帧中继网络相连。IP 地址、DLCI 和 NET 地址都标注在图上了。所有路由器的 IS-IS 配置都已经确认是正确的, 而且没有配置任何认证信息。

这个网络的故障是无法发现路由, 如图 10-65 所示。邻居路由器的帧中继接口的 IP 地址可以 ping 通, 但是邻居路由器上的其他接口地址都不能 ping 通, 如图 10-66 所示。这些 ping 的结果表明帧中继 PVC 是正常运作的, IP 也是工作的, 但是路由器却没有路由。

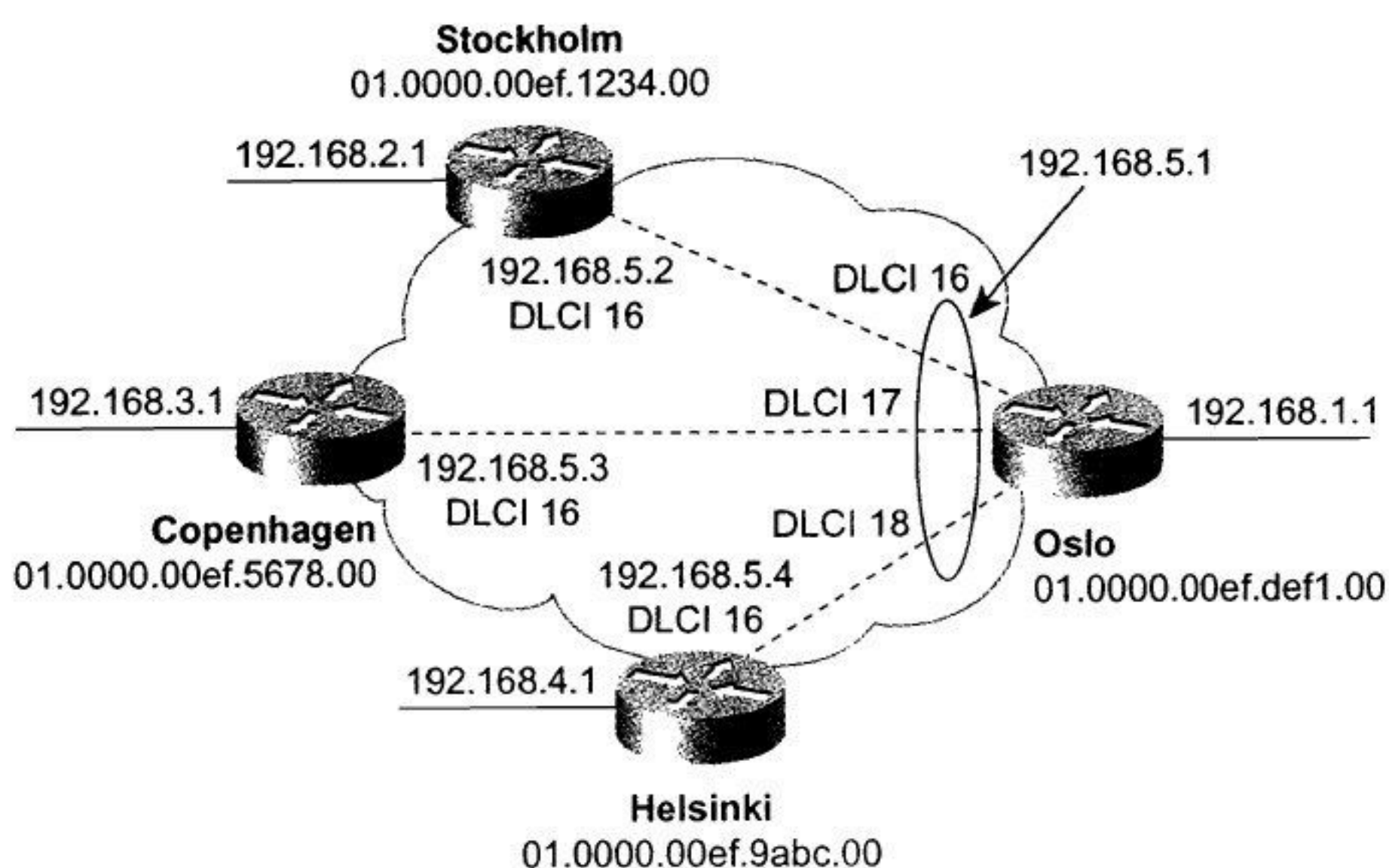


图 10-64 IS-IS 协议在帧中继网络上没有建立邻接关系


```

Oslo#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
       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

Gateway of last resort is not set

C    192.168.1.0 is directly connected, TokenRing0
C    192.168.5.0 is directly connected, Serial0
Oslo#

```

图 10-65 图 10-64 中路由器 Oslo 的路由选择表没有包含任何 IS-IS 路由

```

Oslo#ping 192.168.5.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms
Oslo#ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Oslo#ping 192.168.5.3
Type escape sequence to 5, 100-byte ICMP Echos to 192.168.5.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms
Oslo#ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Oslo#ping 192.168.5.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms
Oslo#ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Oslo#

```

图 10-66 与帧中继网络相连的其他接口可以 ping 通, 但是路由器的可达地址却不能 ping 通

接下来的一步就是检查 IS-IS 的邻居表了。路由器 Oslo 的邻居表显示已经收到 Hello 报文 (如图 10-67 所示), 而且这台路由器能够学习到邻居路由器的系统 ID。同时, 邻居表也显示了邻居路由器的 IP 地址和区域地址是正确的。但是, 所有邻居路由器的状态都是 Init, 这表明还没有建立一个完全的邻接关系。看一下链路状态数据库并确认不存在的邻接关系, 路由器 Oslo 的数据库中的惟一 LSP 就是路由器本身的 LSP, 如图 10-68 所示。

事实上, 路由器正在接收 Hello 报文, 但邻接关系尚未建立成功, 因此问题就指向了 Hello 报文本身。如果 Hello 报文中的参数不正确, 这个 PDU 就会被丢弃。因此可以使用 **debug isis adj-packets** 命令来观察 Hello 报文。在这里需要特别注意的是, 调试信息输出了“封装错误 (encapsulation failed)”的消息, 如图 10-69 所示。这些消息显示路由器显然不能解释收到的 Hello 报文, 因而丢弃了这些 Hello 报文。


```
Oslo#show clns is-neighbors detail
```

```
System Id      Interface  State  Type Priority  Circuit Id      Format
0000.00EF.5678 Se0      Init   L1L2 0 /0    0000.0000.0000.00 Phase V
  Area Address(es): 01
  IP Address(es):   192.168.5.3
  Uptime: 1:11:20
0000.00EF.1234 Se0      Init   L1L2 0 /0    0500.0000.0000.00 Phase V
  Area Address(es): 01
  IP Address(es):   192.168.5.2
  Uptime: 1:11:15
0000.00EF.9ABC Se0      Init   L1L2 0 /0    0700.0000.0000.00 Phase V
  Area Address(es): 01
  IP Address(es):   192.168.5.4
  Uptime: 1:11:20
Oslo#
```

图 10-67 路由器 Oslo 的 IS-IS 邻居表显示了收到 Hello 报文，但是不能完全建立邻接关系

```
Oslo#show isis database
```

```
IS-IS Level-1 Link State Database
```

LSPID	LSP SeqChecksum	LSP Holdtime	ATT/P/OL
0000.00EF.DEF1.00-00*	0x0000001F	0x8460	947 0/0/0
0000.00EF.DEF1.02-00*	0x00000010	0x695E	896 0/0/0
0000.00EF.DEF1.04-00*	0x00000002	0x2F2E	887 0/0/0
0000.00EF.DEF1.05-00*	0x00000008	0x1C3A	847 0/0/0

```
IS-IS Level-2 Link State Database
```

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime	ATT/P/OL
0000.00EF.DEF1.00-00*	0x00000013	0x81BE	829	0/0/0

```
Oslo#
```

图 10-68 路由器 Oslo 的链路状态数据库没有包含任何来自于邻居的 LSP

```
Oslo#debug isis adj-packets
```

```
IS-IS Adjacency related packets debugging is on
```

```
Oslo#
```

```
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Encapsulation failed for level 2 IIH on Serial0
ISIS-Adj: Rec serial IIH from DLCI 17 on Serial0, cir type 3, cir id 00
ISIS-Adj: rcvd state 2, old state 1, new state 1
ISIS-Adj: Action = 1, new_type = 3
ISIS-Adj: Sending L2 IIH on TokenRing0
ISIS-Adj: Encapsulation failed for level 1 IIH on Serial0
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Rec serial IIH from DLCI 18 on Serial0, cir type 3, cir id 07
ISIS-Adj: rcvd state 2, old state 1, new state 1
ISIS-Adj: Action = 1, new_type = 3
ISIS-Adj: Encapsulation failed for level 2 IIH on Serial0
ISIS-Adj: Sending L2 IIH on TokenRing0
ISIS-Adj: Encapsulation failed for level 1 IIH on Serial0
ISIS-Adj: Rec serial IIH from DLCI 16 on Serial0, cir type 3, cir id 05
ISIS-Adj: rcvd state 2, old state 1, new state 1
ISIS-Adj: Action = 1, new_type = 3
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Encapsulation failed for level 2 IIH on Serial0no debu
ISIS-Adj: Sending L2 IIH on TokenRing0
ISIS-Adj: Encapsulation failed for level 1 IIH on Serial0g a
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Rec serial IIH from DLCI 17 on Serial0, cir type 3, cir id 00
ISIS-Adj: rcvd state 2, old state 1, new state 1
ISIS-Adj: Action = 1, new_type = 3
ISIS-Adj: Encapsulation failed for level 2 IIH on Serial0ll
```

图 10-69 打开调试命令 **debug isis adj-packets**，输出结果显示了由于封装失败而正在丢弃 Hello 报文

任何时候看到封装失败的消息都应该怀疑数据链路的问题和所连接的接口问题。使用命令 **show interface serial** 检查接口, 没有发现存在严重的误码率, 因此不像是帧中继 PVC 链路破坏了 Hello 报文。下一步就要检查接口的配置了。图 10-64 中的 4 台路由器的接口配置如下:

路由器 Oslo:

```
interface Serial0
  ip address 192.168.5.1 255.255.255.0
  ip router isis
  encapsulation frame-relay
  frame-relay interface-dlci 16
  frame-relay interface-dlci 17
  frame-relay interface-dlci 18
```

路由器 Stockholm:

```
interface Serial0
  no ip address
  encapsulation frame-relay
  !
interface Serial0.16 point-to-point
  ip address 192.168.5.2 255.255.255.0
  ip router isis
  frame-relay interface-dlci 16
```

路由器 Copenhagen:

```
interface Serial0
  no ip address
  encapsulation frame-relay
  !
interface Serial0.16 point-to-point
  ip address 192.168.5.3 255.255.255.0
  ip router isis
  frame-relay interface-dlci 16
```

路由器 Helsinki:

```
interface Serial0
  no ip address
  encapsulation frame-relay
  !
interface Serial0.16 point-to-point
  ip address 192.168.5.4 255.255.255.0
  ip router isis
  frame-relay interface-dlci 16
```



```
Oslo#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
       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

Gateway of last resort is not set

C    192.168.1.0 is directly connected, TokenRing0
i L1 192.168.2.0 [115/20] via 192.168.5.5, Serial0.16
i L1 192.168.3.0 [115/20] via 192.168.5.9, Serial0.17
i L1 192.168.4.0 [115/20] via 192.168.5.13, Serial0.18
    192.168.5.0 is variably subnetted, 4 subnets, 2 masks
C    192.168.5.12 255.255.255.252 is directly connected, Serial0.18
C    192.168.5.8 255.255.255.252 is directly connected, Serial0.17
C    192.168.5.4 255.255.255.252 is directly connected, Serial0.16
C    192.168.5.0 255.255.255.0 is directly connected, Serial0
Oslo#
```

图 10-70 在路由器 Oslo 配置成点到点接口后，并且进行重新编址以使每一个 PVC 成为一个单独的子网后，IS-IS 路由选择就起作用了

通过比较这些配置发现了一个问题，虽然这个问题不一定是很明显。路由器 Stockholm、Copenhagen 和 Helsinki 都配置成了点到点的子接口，但路由器 Oslo 没有使用子接口。在缺省情况下，Cisco 路由器的串行接口在封装成帧中继时是一个多点接口。因此，路由器 Stockholm、Copenhagen 和 Helsinki 发送点到点 IS-IS Hello 报文，而路由器 Oslo 发送 L1 和 L2 类型的 IS-IS LAN Hello 报文。

IS-IS 协议的配置中没有一个类似于 OSPF 协议中的 `ip ospf network` 命令的配置选项，因此，路由器 Oslo 必须重新配置成点到点接口，并且必须更改 IP 地址，以便使每一个 PVC 链路都在不同的子网中，如图 10-70 所示。

10.4 展 望

到现在为止，所有的 IP IGP 协议都已经论述过了，下一步将开始研究一些有效的工具来帮助控制我们的互联网络。第三部分涵盖了路由再分配、缺省路由、按需路由选择、路由过滤和路由图。

10.5 总结表：第 10 章命令总结

命 令	描 述
<code>area-password password</code>	配置 IS-IS 区域（层 1）认证
<code>clns routing</code>	使 CLNS PDU 的路由选择有效
<code>debug isis adj-packets</code>	显示 IS-IS Hello PDU 报文的行为
<code>debug isis spf-events</code>	显示触发一个 IS-IS SPF 计算的事件的详细信息
<code>debug isis snp-packets</code>	显示路由器接收和发送 SNP 报文的有关信息
<code>debug isis spf-statistics</code>	显示一个有关 IS-IS SPF 计算的统计信息
<code>debug isis spf-triggers</code>	显示触发 IS-IS SPF 计算的事件
<code>debug isis update-packets</code>	显示路由器接收和发送 LSP、CSNP 和 PSNP 报文的有关信息

续表

命 令	描 述
default-information originate [route-map <i>map-name</i>]	生成一条进入 IS-IS 域的缺省 IP 路由
domain-password <i>password</i>	配置 IS-IS 域 (层 2) 认证
ignore-lsp-errors	配置一台 IS-IS 路由器忽略错误的 LSP 而不是触发一个 LSP 的清除
ip router isis [<i>tag</i>]	在一个路由器接口上启动 IS-IS 路由选择
isis csnp-interval <i>seconds</i> { <i>level-1</i> <i>level-2</i> }	指定一台 IS-IS 指定路由器发送 CSNP 报文的时间间隔, 以秒数计
isis hello-interval <i>seconds</i> { <i>level-1</i> <i>level-2</i> }	指定 IS-IS Hello PDU 报文重传的时间间隔, 以秒数计
isis hello-multiplier <i>multiplier</i> { <i>level-1</i> <i>level-2</i> }	指定一个邻居路由器在宣告它与始发路由器邻接关系失效之前, 必须错过的 IS-IS Hello PDU 报文的数目
isis metric default-metric { <i>level-1</i> <i>level-2</i> }	指定一个接口的 IS-IS 缺省度量
isis password <i>password</i> { <i>level-1</i> <i>level-2</i> }	在两台 IS-IS 邻居路由器之间配置认证
isis priority <i>value</i> { <i>level-1</i> <i>level-2</i> }	指定一个接口用来选取指定路由器的优先级
isis retransmit-interval <i>seconds</i>	指定一台路由器在一个点到点链路上发送一条 LSP 后需要等待确认的时间, 如果超过这个时间还没收到确认, 路由器将会重传这条 LSP
is-type { <i>level-1</i> <i>level-1-2</i> <i>level-2-only</i> }	配置一台路由器作为一台 L1、L1/L2 或者 L2 类型的 IS-IS 路由器
net <i>network-entity-title</i>	配置一台 IS-IS 路由器的 NET 地址
router isis [<i>tag</i>]	启动一个 IS-IS 路由选择进程
show clns is-neighbor [<i>type number</i>][<i>detail</i>]	显示一个 IS-IS 邻居表
show isis database [<i>level-1</i>][<i>level-2</i>][<i>l1</i>][<i>l2</i>][<i>detail</i>] [<i>lspid</i>]	显示一个 IS-IS 链路状态数据库
show isis spf-log	显示路由器怎样以及为什么要运行一个完全的 SPF 计算
summary-address <i>address-mask</i> { <i>level-1</i> <i>level-1-2</i> <i>level-2</i> }	配置 IP 地址汇总
which-route { <i>nsap-address</i> <i>clns-name</i> }	查找并显示一个指定的 CLNS 目的地址在路由选择表中对应的 IP 地址和区域地址的详细信息

10.6 复 习 题

1. 什么是中间系统?
2. 什么是网络协议数据单元?
3. L1、L2 和 L1/L2 类型的路由器有什么不同?
4. 说明一个 IS-IS 区域和一个 OSPF 区域的不同之处。
5. 什么是网络实体标题 (NET)?
6. 在 NET 中必须将 NSAP 选择符设置成什么值?
7. 系统 ID 的用途是什么?
8. 一台路由器是怎样确定它所在的区域的?
9. IS-IS 协议在一个广播型子网上选取备份指定路由器吗?
10. 伪节点 ID 的用途是什么?
11. 一个 IS-IS LSP 报文的最大老化时间 (也就是最大生存时间 MaxAge) 是多少?
12. OSPF 协议老化它的 LSA 和 IS-IS 协议老化它的 LSP 的方式有什么基本的不同?
13. 一台 IS-IS 路由器多长时间重刷新一次它的 LSP?
14. 什么是完全序列号报文 (CSNP)? 它有什么用途?
15. 什么是部分序列号报文 (PSNP)? 它有什么用途?

16. 超载位 (OL) 的用途是什么?
17. 区域关联位 (ATT) 的用途是什么?
18. ISO 为 IS-IS 规定的度量有哪些? 在 Cisco IOS 中又支持多少?
19. IS-IS 缺省度量的最大值是多少?
20. 一条 IS-IS 路由的最大度量值是多少?
21. 层 1 类型的 IS-IS 度量和层 2 类型的 IS-IS 度量有什么不同?
22. 内部 IS-IS 度量和外部 IS-IS 度量有什么不同?

10.7 配置练习

1. 表 10-6 中显示了 11 台路由器的接口、接口地址和子网掩码。这个表还指定了属于同一个区域的路由器。使用下面的指导策略, 写出每台路由器的集成 IS-IS 的配置:

- 为每台路由器配置自己的系统 ID;
- 使用尽可能短的 NET 地址;
- 适当地把这些路由器配置成 L1、L2 或者 L1/L2 类型的路由器。

提示: 首先画一张路由器和子网的图形。

表 10-6

配置练习 1~5 的路由器信息

路 由 器	区 域	接 口	地址/掩码
A	0	E0	192.168.1.17/28
		E1	192.168.1.50/28
B	0	E0	192.168.1.33/28
		E1	192.168.1.51/28
C	0	E0	192.168.1.49/28
		S0	192.168.1.133/30
D	2	S0	192.168.1.134/30
		S1	192.168.1.137/30
E	2	S0	192.168.1.142/30
		S1	192.168.1.145/30
		S2	192.168.1.138/30
F	2	S0	192.168.1.141/30
		S1	192.168.1.158/30
G	1	E0	192.168.1.111/27
		S0	192.168.1.157/30
H	1	E0	192.168.1.73/27
		E1	192.168.1.97/27
I	3	E0	192.168.1.225/29
		E1	192.168.1.221/29
		S0	192.168.1.249/30
		S1	192.168.1.146/30
J	3	E0	192.168.1.201/29
		E1	192.168.1.217/29
K	3	E0	192.168.1.209/29
		S0	192.168.1.250/30

2. 在表 10-6 中区域 2 内的所有路由器上配置认证。路由器 D 和 E 之间使用口令“Eiffel”，路由器 D 和路由器 F 之间使用“Tower”。
3. 在表 10-6 中区域 1 上配置层 1 类型的认证，使用口令“Scotland”。
4. 在表 10-6 的路由器上配置层 2 类型的认证，使用口令“Vienna”。
5. 配置表 10-6 中区域 0、1 和 3 的 L1/L2 路由器来汇总它们区域的地址。

10.8 故障排除练习

1. 在图 10-71 和图 10-72 中，显示了路由器 A 和路由器 B 的 IS-IS 邻居表，这两台路由器是通过令牌环网络相连的。IS-IS 可以在它们之间交换路由，并能够把路由放入到它们的路由选择表中，但是没有 IP 的通信量通过这两台路由器。出现了什么错误？

2. 在图 10-73 中，显示了一台路由器的调试信息，这台路由器没有和它 TO0 接口上的邻居路由器成功建立邻接关系。出现了什么错误？

```
Router_A#show clns is-neighbors detail
```

System Id	Interface	State	Type	Priority	Circuit Id	Format
0000.00EF.DCBA	To0	Up	L1L2	64/64	0000.00EF.DCBA.04	Phase V
Area Address(es): 01						
IP Address(es): 192.168.11.2						
Uptime: 0:09:25						
0000.00EF.5678	Se0.17	Up	L1L2	0 / 0	00	Phase V
Area Address(es): 01						
IP Address(es): 192.168.5.9						
Uptime: 1:28:22						
0000.00EF.9ABC	Se0.18	Up	L1L2	0 / 0	07	Phase V
Area Address(es): 01						
IP Address(es): 192.168.5.13						
Uptime: 1:29:45						
0000.00EF.1234	Se0.16	Up	L1L2	0 / 0	06	Phase V
Area Address(es): 01						
IP Address(es): 192.168.5.5						
Uptime: 1:29:45						

Router_A#

图 10-71 故障排除练习 1，路由器 A 的 IS-IS 邻居表

```
Router_B#show clns is-neighbors detail
```

System Id	Interface	State	Type	Priority	Circuit Id	Format
0000.00EF.DEF1	To0	Up	L1L2	64/64	0000.00EF.DCBA.04	Phase V
Area Address(es): 01						
IP Address(es): 192.168.1.1						
Uptime: 0:11:06						

Router_B#

图 10-72 故障排除练习 1，路由器 B 的 IS-IS 邻居表


```
Router_B#debug isis adj-packets
IS-IS Adjacency related packets debugging is on
Router_B#
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Rec L2 IIH from 0000.3090.c7df (TokenRing0), cir type 2, cir id 0000.0
0EF.DCBA.04
ISIS-Adj: is-type mismatch
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Rec L2 IIH from 0000.3090.c7df (TokenRing0), cir type 2, cir id 0000.0
0EF.DCBA.04
ISIS-Adj: is-type mismatch
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
ISIS-Adj: Sending L1 IIH on TokenRing0
ISIS-Adj: Sending L1 IIH on TokenRing1
```

图 10-73 故障排除练习 2 的调试输出信息

第三部分

路由控制和互 操作性

第 11 章 路由重新分配

第 12 章 缺省路由和按需路由选择

第 13 章 路由过滤

第 14 章 路由图

第 11 章

路由重新分配

本章包括以下主题：

- 重新分配的概念
度量
管理距离
从无类别化协议向有类别协议重新分配
- 配置重新分配
案例研究：重新分配 IGRP 和 RIP
案例研究：重新分配 EIGRP 和 OSPF
案例研究：重新分配和路由汇总
案例研究：重新分配 IS-IS 和 RIP
案例研究：重新分配静态路由

当路由器使用路由选择协议进行路由通告时，如果该路由是通过其他方式获取的，那么路由器将要执行重新分配。这里所谓的其他方式可能是另外一个路由选择协议、静态路由或直连目标网络。例如，路由器可能同时运行 OSPF 进程和 RIP 进程。如果设置 OSPF 进程通告来自 RIP 进程的路由，这就叫做重新分配 RIP。

在整个 IP 互联网络中，如果从配置管理和故障管理的角度看，我们通常更愿意运行一种路由选择协议，而不是多种路由选择协议。然而，现代的互联网络又常常强迫我们接受多协议 IP 路由选择域这一现实。当部门、分公司乃至整个公司合并时，必须统一它们原来的自主互联网络。

在大部分案例中，将要被合并的互联网络在实现和发展上都不相同，并且它们满足了不同的需要，是不同设计理念的产物。这种差异性使得向单一路由选择协议的迁移成为一项复杂的任务。在某些案例中，公司的策略可能会强制使用多种路由选择协议。而在少数场合还会出现因网络管理员不

能很好地相处而采用多种路由选择协议。

多厂商环境是需要重新分配路由的另一个因素。例如, 一个运行 Cisco IGRP 和 EIGRP 的互连网络可能会与使用另一个厂商路由器的互连网络合并, 而这种路由器仅支持 RIP 和 OSPF。如果没有重新分配, 那么 Cisco 路由器需要重新配置一种公开的协议或者用 Cisco 路由器替代非 Cisco 路由器。

当多种路由协议被拼凑在一起时, 使用重新分配是很有必要的, 而且重新分配也是严谨的互连网络设计的一部分。图 11-1 给出了一个例子, 这里把两个 OSPF 进程域连接在一起, 但是 OSPF 进程之间并不直接通信, 取而代之的是在每台路由器上配置静态路由, 静态路由指向其他 OSPF 域内的被选网络。

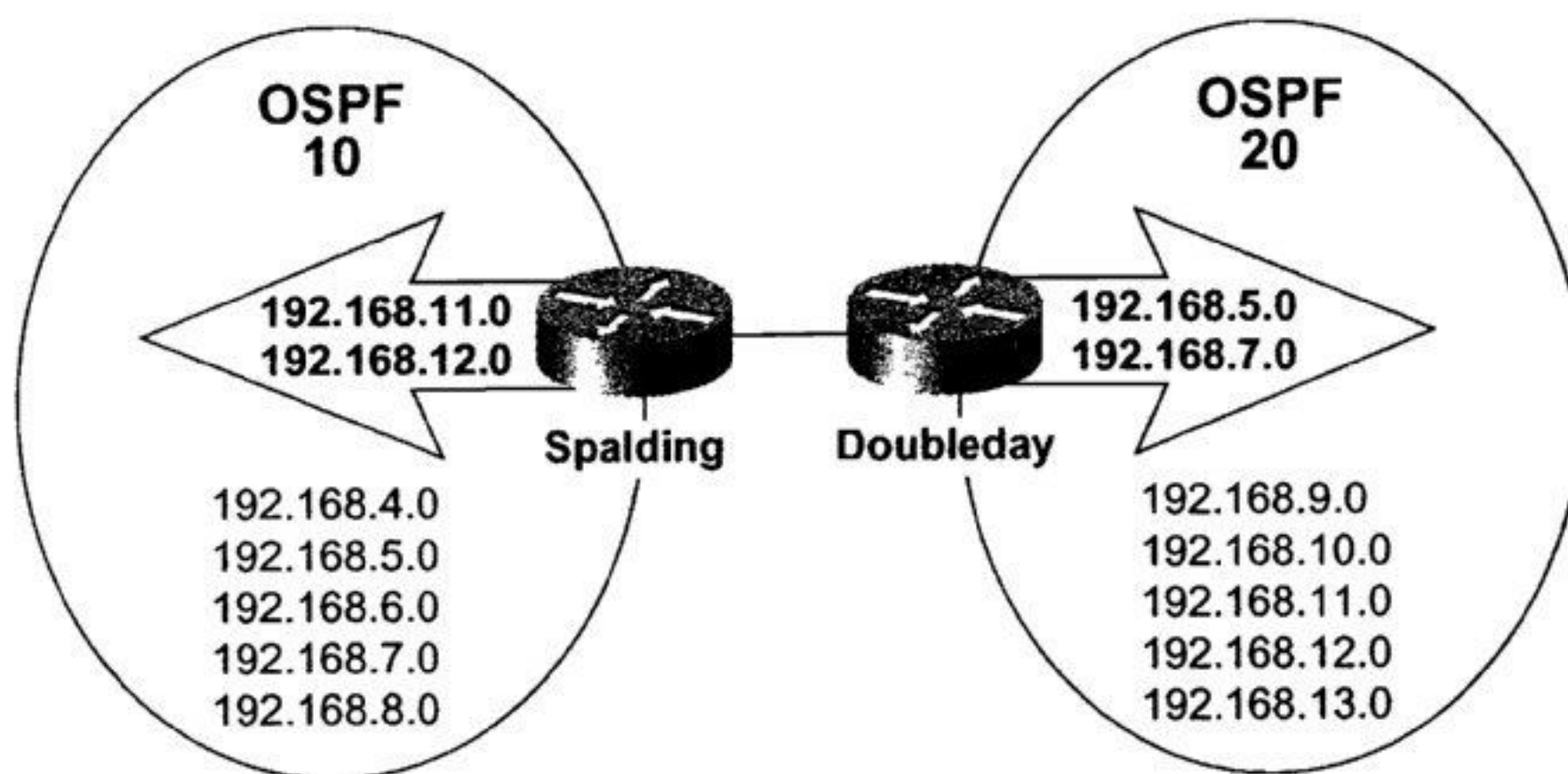


图 11-1 在这个互连网络中, 在每台路由器上都配置了静态路由, 并且向 OSPF 重新分配这些路由。

结果, 可以做到精细地控制在两个 OSPF 域之间的网络通告

例如, 路由器 Spalding 包含指向网络 192.168.11.0 和 192.168.12.0 的路由, 所以 Spalding 把这些静态路由重新分配到 OSPF 中, OSPF 又向 OSPF 10 内的其他路由器通告这些路由。这样做的结果是向 OSPF 10 隐瞒了 OSPF 20 中的其他网络。重新分配使得 OSPF 的动态特性和静态路由的精确控制性融合在一起。

如果不是非要使用动态路由选择协议的话, 在拨号环境中向动态路由选择协议重新分配静态路由也是非常有用的。动态协议周期性的管理流量会导致拨号线路始终保持接通状态。而通过阻止路由更新和 Hello 信息通过线路, 并在两边配置静态路由, 管理员可以确保线路在有用户流量通过时才接通。而且向动态路由选择协议重新分配静态路由, 可以使拨号线路两边的所有路由器都知道链路对方的所有网络。

注意: 除了少数特例外,¹在相同路由器上存在不止一种路由选择协议并不意味着重新分配自动发生。重新分配必须被明确地配置。在没有使用重新分配的单一路由器上配置多种路由选择协议的方法叫做午夜航船 (Ships In the Night, SIN) 路由选择。路由器将会在每个进程域内向它的对等路由器传递路由, 但是进程域之间却一无所知——这就好比黑暗中航行的船只一样。虽然 SIN 路由选择法通常指的是在相同路由器上多种路由选择协议为多种可路由

¹ 在 IP 中, 自主系统号相同的 IGRP 和 EIGRP 进程可以自动重新分配。在第 8 章“增强型内部网关路由选择协议 (EIGRP)”的“案例分析: 使用 IGRP 重新分配”一节中给出了这样的例子。

协议进行路由选择（例如 OSPF 为 IP 和 NLSP 为 IPX 进行路由选择），但是它也可以指在单一路由器上两个 IP 协议为单独的 IP 域进行路由选择。

11.1 重新分配的原则

IP 路由选择协议的能力相差非常大。对重新分配影响最大的协议特性是度量和距离的差异性以及每种协议的有类别和无类别能力。在重新分配时如果忽略了对这些差异的考虑将导致以下后果，最好情况会出现某些或全部路由交换失败，最坏情况将造成路由环路和黑洞。

11.1.1 度量

图 11-1 中的路由器正在向 OSPF 重新分配静态路由，然后它们会向其他 OSPF 路由器通告这些路由。虽然静态路由没有相关联的度量，但每条 OSPF 路由必须有一个代价值。有关度量冲突的另一个例子是向 IGRP 重新分配 RIP 路由，RIP 的度量是跳数，而 IGRP 使用带宽和时延。在这两种情况中，接收被重新分配路由的协议必须能够将自己的度量与这些路由联系起来。

所以执行重新分配的路由器必须为被重新分配的路由指派度量。图 11-2 给出了一个例子，这里 EIGRP 被重新分配进入 OSPF，同时 OSPF 也被重新分配进入 EIGRP。OSPF 不能理解 EIGRP 的复合度量，EIGRP 也不能理解 OSPF 的代价度量。因此在向 OSPF 传递 EIGRP 路由之前，路由器的重新分配进程必须为每一条 EIGRP 路由分配代价度量。同样，路由器在向 EIGRP 传递 OSPF 路由之前也必须为每一条 OSPF 路由分配带宽、时延、可靠性、负载和 MTU 度量值。如果分配了不正确的度量，重新分配将会失败。

本章后面的案例分析将会讨论为了进行重新分配，应该如何配置路由器来达到分配度量的目的。

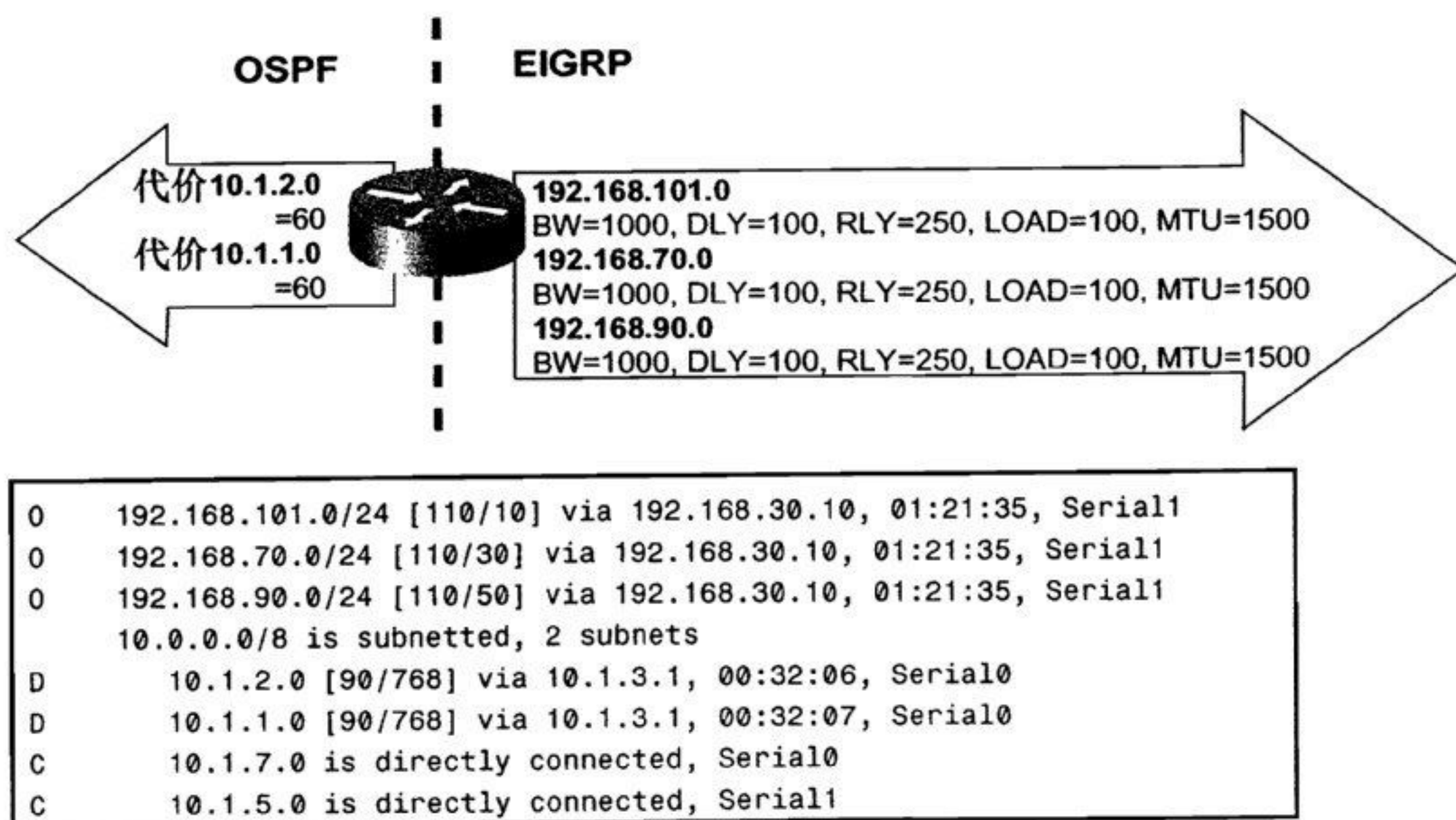


图 11-2 当重新分配路由时，必须为路由分配一个接收协议可以理解的度量值

11.1.2 管理距离

度量的差异性产生了另一个问题：如果路由器正在运行多个路由选择协议，并从每个协议都学习到一条到达相同目标网络的路由，那么应该选择哪一条路由呢？每一个路由选择协议均使用自己的度量方案定义最优路径。比较不同度量的路由，例如代价和跳数，就像比较苹果和橙子一样。

这个问题的答案是管理距离。正像为路由分配度量以便可以确定首选路径一样，因为要确定首选路由源，所以需要向路由源分配管理距离。把管理距离看作可信度的一个量度，管理距离越小，协议的可信度越高。

例如，假设运行 RIP 和 EIGRP 的路由器从邻居 RIP 路由器那里学习到一条指向网络 192.168.5.0 的路由，从邻居 EIGRP 路由器那里学习到一条指向相同网络的路由。由于 EIGRP 的复合度量，使得该协议更有可能确定最佳路由。因此，EIGRP 比 RIP 更可信。¹

表 11-1 列出了缺省的 Cisco 管理距离。EIGRP 的管理距离为 90，而 RIP 的管理距离为 120。因此可以认为 EIGRP 比 RIP 更值得信赖。

表 11-1

Cisco 缺省管理距离

路 由 源	管 理 距 离
直连接口	0
静态路由	1
EIGRP 汇总路由	5
外部 BGP	20
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
外部 EIGRP	170
内部 BGP	200
未知	255

虽然管理距离帮助解决了不同度量带来的混乱，但是它又给重新分配带来了问题。例如，在图 11-3 中 Gehrig 和 Ruth 都正在向 IGRP 重新分配 RIP 路由。Gehrig 通过 RIP 知道网络 192.168.1.0，并且将其通告给 IGRP 域。结果，Ruth 不仅通过 RIP 从 Combs 处学习到网络 192.168.1.0，而且还通过 IGRP 从 Meusel 那里也学习到该网络。

图 11-4 给出了 Ruth 的路由选择表。注意，指向网络 192.168.1.0 的路由是一条 IGRP 路由。Ruth 之所以选择 IGRP 路由是因为 IGRP 比 RIP 具有更小的管理距离。Ruth 将经过 Meusel 沿着这条“风景优美的路线”发送所有报文，代替直接向 Combs 发送报文。

水平分隔阻止了在图 11-3 互联网络中路由环路的发生。Gehrig 和 Ruth 最初都向 IGRP 域通告网络 192.168.1.0，并且最终 4 台 IGRP 路由器都收敛到一条到达该网络的路径。然而，这种收敛是不可预知的。重新启动 Lazzeri 和 Meusel 可以看到这一情形。在重新启动后，Ruth

¹ 回忆一下第 3 章，当静态路由使用接口替代下一跳地址时，目的网络即被认为是直连网络。

的路由选择表显示到达网络 192.168.1.0 的下一跳路由器是 Combs（图 11-5）。

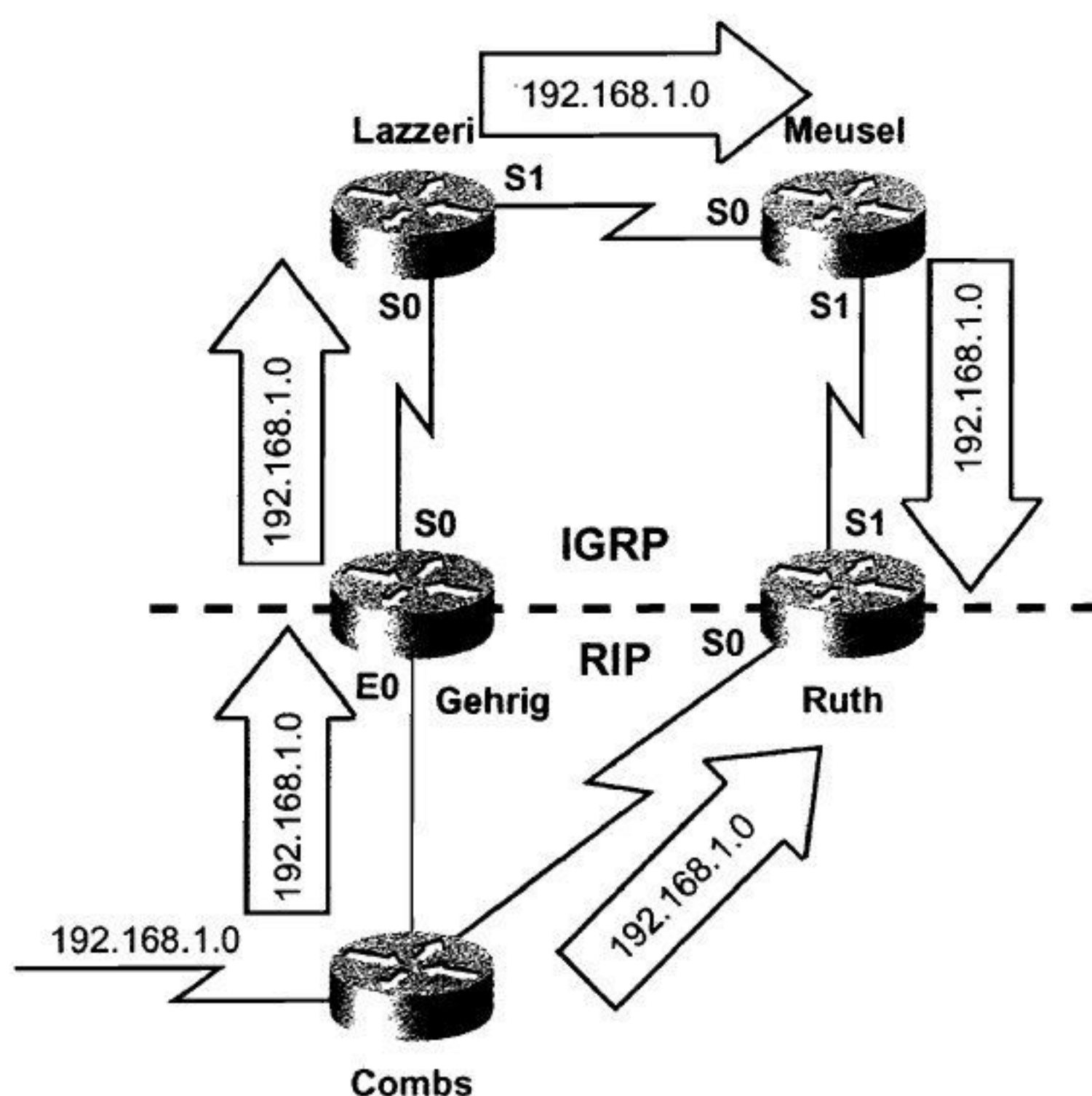


图 11-3 通过 RIP 和 IGRP，网络 192.168.1.0 被通告给 Ruth

```
Ruth#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
       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

Gateway of last resort is not set

I    192.168.1.0/24 [100/16100] via 192.168.5.1, 00:00:00, Serial1
I    192.168.2.0/24 [100/12576] via 192.168.5.1, 00:00:00, Serial1
I    192.168.3.0/24 [100/12476] via 192.168.5.1, 00:00:01, Serial1
I    192.168.4.0/24 [100/10476] via 192.168.5.1, 00:00:01, Serial1
C    192.168.5.0/24 is directly connected, Serial1
C    192.168.6.0/24 is directly connected, Serial0
Ruth#
```

图 11-4 虽然从 Ruth 到达网络 192.168.1.0 的最佳路径是从 S0 出发经过 Combs 去往目标网络，但是 Ruth 却选择了从 S1 出发经 Meusel 的路径

在重新启动后收敛不仅难以预知，而且很慢。图 11-6 显示了重新启动完毕大约又经过 3min 后 Gehrig 的路由选择表。它使用 Lazzeri 作为到达网络 192.168.1.0 的下一跳路由器，但是 ping 该网络中的一个在线地址却发生失败。从 Lazzeri 的路由选择表（图 11-7）可以看出问题所在：Lazzeri 使用 Gehrig 作为下一跳路由器，因此存在路由环路。

下面是导致环路的事件顺序：

1. 当 Lazzeri 和 Meusel 重新启动时，Gehrig 和 Ruth 的路由条目显示经 Combs 可以到达网络 192.168.1.0。


```

Ruth#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
       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

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.6.2, 00:00:23, Serial0
I    192.168.2.0/24 [100/12576] via 192.168.5.1, 00:00:22, Serial1
I    192.168.3.0/24 [100/12476] via 192.168.5.1, 00:00:22, Serial1
I    192.168.4.0/24 [100/10476] via 192.168.5.1, 00:00:22, Serial1
C    192.168.5.0/24 is directly connected, Serial1
C    192.168.6.0/24 is directly connected, Serial0
Ruth#

```

图 11-5 图 11-3 中的互联网的收敛不可预知。在路由器重新启动后, Ruth 现在选择经过 Combs 到达网络 192.168.1.0

```

Gehrig#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

I    192.168.1.0/24 [100/16100] via 192.168.3.2, 00:02:38, Serial0
C    192.168.2.0/24 is directly connected, Ethernet0
C    192.168.3.0/24 is directly connected, Serial0
I    192.168.4.0/24 [100/10476] via 192.168.3.2, 00:00:29, Serial0
I    192.168.5.0/24 [100/12476] via 192.168.3.2, 00:00:29, Serial0
I    192.168.6.0/24 [100/14476] via 192.168.3.2, 00:00:39, Serial0
Gehrig#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Gehrig#

```

图 11-6 重新启动后不久, Gehrig 为报文选择经 Lazzeri 到达网络 192.168.1.0 的路径

```

Lazzeri#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
       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

Gateway of last resort is not set

I    192.168.1.0/24 [100/12100] via 192.168.3.1, 00:04:21, Serial0
I    192.168.2.0/24 [100/8576] via 192.168.3.1, 00:00:33, Serial0
C    192.168.3.0/24 is directly connected, Serial0
C    192.168.4.0/24 is directly connected, Serial1
I    192.168.5.0/24 [100/10476] via 192.168.4.2, 00:00:53, Serial1
I    192.168.6.0/24 [100/12100] via 192.168.3.1, 00:02:32, Serial0
Lazzeri#

```

图 11-7 Lazzeri 路由为报文选择经 Gerig 到达 192.168.1.0 的路径, 因而产生路由环路。注意路由的年龄

2. 随着 Lazzeri 和 Meusel 启动完毕, Gehrig 和 Ruth 发送包括网络 192.168.1.0 的 IGRP 更新信息, 仅仅由于运气, Ruth 比 Gehrig 更早一点发送了更新信息。

3. Meuse 接收到 Ruth 的更新信息, 把 Ruth 作为下一跳路由器, 并且向 Lazzeri 发送更新信息。

4. Lazzeri 接收到 Meusel 的更新信息, 把 Meusel 作为下一跳路由器。

5. Lazzeri 和 Gehrig 在差不多相同的时刻互相发送了更新信息。Lazzeri 把 Gehrig 作为到达网络 192.168.1.0 的下一跳路由器, 因为这条路由比 Meusel 的路由更接近目标。Gehrig 把 Lazzeri 作为到达网络 192.168.1.0 的下一跳路由器, 因为 Lazzeri 的 IGRP 通告的管理距离比 Combs 的 RIP 通告的小。至此环路产生。

水平分隔和失效计时器最终将会解决这一问题。虽然 Lazzeri 正在向 Meusel 通告 192.168.1.0, 但是 Meusel 仍会继续使用经过 Ruth 的路径。由于 Ruth 是下一跳路由器, 所以在 Meusel 的接口 S1 上, 水平分隔对 192.168.1.0 有效。Meusel 还向 Lazzeri 通告 192.168.1.0, 但是 Lazzeri 认为 Gehrig 更接近目的网络。

由于 Lazzeri 和 Gehrig 都将对方看作是去往 192.168.1.0 的下一跳路由器, 所以它们相互不通告此路由。保存在它们路由选择表中的这条路由一直老化到失效计时器超时为止 (图 11-8)。

```
Lazzeri#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
       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

Gateway of last resort is not set

I    192.168.1.0/24 is possibly down, routing via 192.168.3.1, Serial0
I    192.168.2.0/24 [100/8576] via 192.168.3.1, 00:00:57, Serial0
C    192.168.3.0/24 is directly connected, Serial0
C    192.168.4.0/24 is directly connected, Serial1
I    192.168.5.0/24 [100/10476] via 192.168.4.2, 00:01:25, Serial1
I    192.168.6.0/24 is possibly down, routing via 192.168.3.1, Serial0
Lazzeri#
```

图 11-8 在指向 192.168.1.0 路由的失效计时器超时后, 这条路由将会被声明不可达, 并且抑制计时器被启动

在 Lazzeri 的失效计时器超时后, 指向 192.168.1.0 的路由将被抑制。虽然 Meusel 正在通告指向该网络的路由, 但是 Lazzeri 直到抑制计时器超时才能接收该通告。图 11-9 给出了 Lazzeri 最终接收了这条来自 Meusel 的路由, 图 11-10 显示了 Gehrig 通过 Lazzeri 可以成功地到达 192.168.1.0。但是这两台路由器花费了 9min 时间才得以收敛, 而且还使用了一条非最佳路由。

管理距离导致的问题会比前面例子所述的非最佳路径、不可预知的行为以及慢收敛问题更严重。例如, 图 11-11 给出的互联网络与图 11-3 中的互联网络本质上是相同, 除了 IGRP 路由器之间的链路为帧中继永久虚电路。帧中继接口上的 IP 水平分隔功能在缺省情况下是关闭的, 结果是在 Lazzeri 和 Gehrig 之间以及 Meuse 和 Ruth 之间会形成永久的路由环路, 并且从 IGRP 域无法到达网络 192.168.1.0。


```
Lazzeri#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
       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

Gateway of last resort is not set

I   192.168.1.0/24 [100/14100] via 192.168.4.2, 00:00:27, Serial1
I   192.168.2.0/24 [100/8576] via 192.168.3.1, 00:00:02, Serial0
C   192.168.3.0/24 is directly connected, Serial0
C   192.168.4.0/24 is directly connected, Serial1
I   192.168.5.0/24 [100/10476] via 192.168.4.2, 00:00:28, Serial1
I   192.168.6.0/24 [100/12476] via 192.168.4.2, 00:00:28, Serial1
Lazzeri#
```

图 11-9 在网络 192.168.1.0 的抑制计时器超时后, Lazzeri 接收被 Meusel 通告的路由

```
Gehrig#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

I   192.168.1.0/24 [100/16100] via 192.168.3.2, 00:00:32, Serial0
C   192.168.2.0/24 is directly connected, Ethernet0
C   192.168.3.0/24 is directly connected, Serial0
I   192.168.4.0/24 [100/10476] via 192.168.3.2, 00:00:33, Serial0
I   192.168.5.0/24 [100/12476] via 192.168.3.2, 00:00:33, Serial0
I   192.168.6.0/24 [100/14476] via 192.168.3.2, 00:00:33, Serial0
Gehrig#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/72/108 ms
Gehrig#
```

图 11-10 Gehrig 现在可以通过 Lazzeri 到达网络 192.168.1.0

在重新分配时有几种工具和策略可以避免路由环路, 可以使用操作管理距离、路由过滤和路由图。第 13 章、14 章将涉及路由过滤和路由图。这两章还示范了修改管理距离的技术。

11.1.3 从无类别协议向有类别协议重新分配

从无类别路由进程域向有类别域重新分配路由会产生那些影响, 这值得我们仔细地考虑。为了理解为什么这样做, 首先有必要理解有类别路由选择协议怎样应对变长子网划分。回想第 5 章, 有类别路由选择协议不能通告携带子网掩码的路由。对于有类别路由器所接收到的每一条路由, 无外乎是下面两种情况之一:

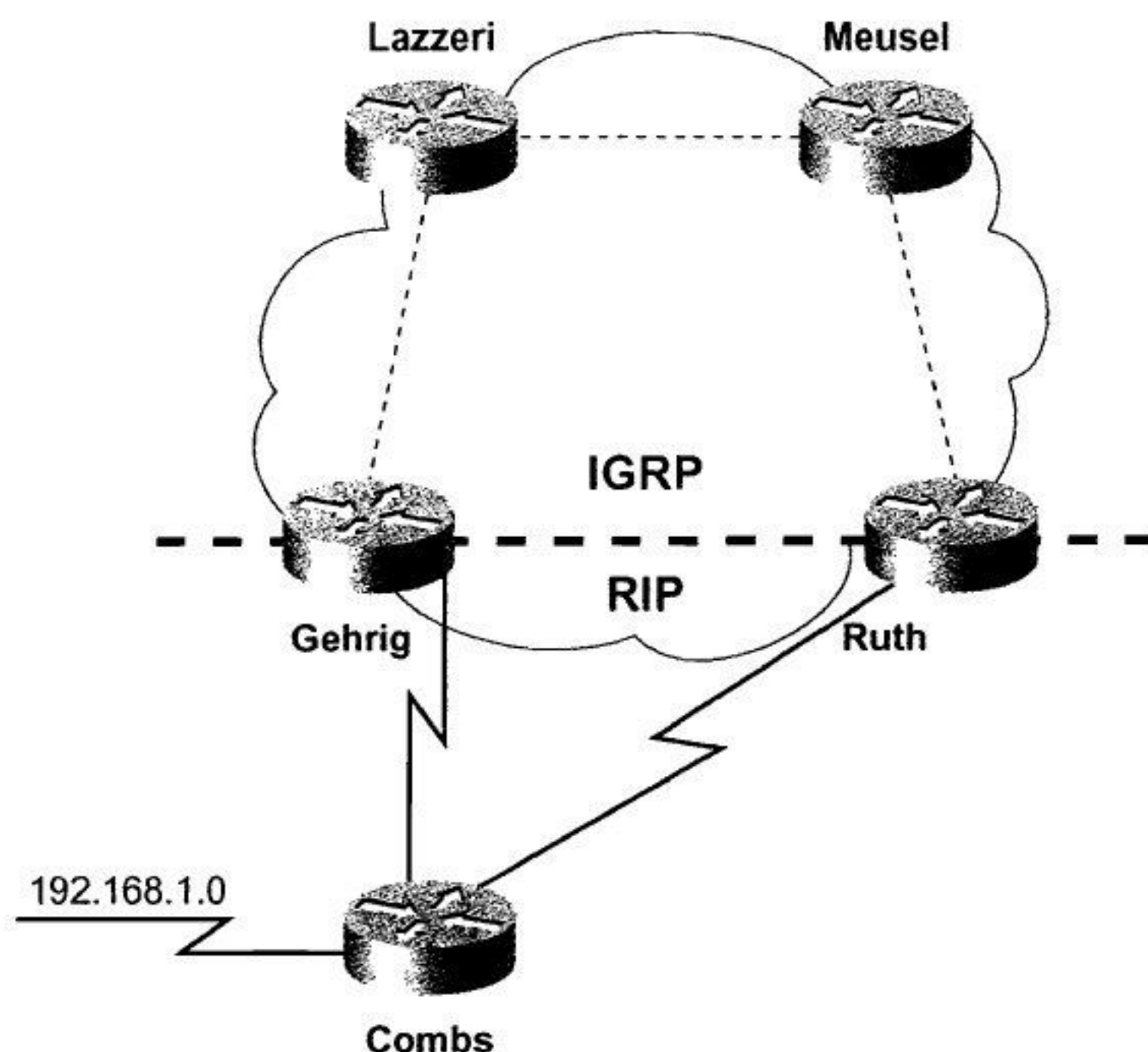


图 11-11 因为在帧中继接口上水平分隔功能在缺省状态下是关闭的，所以在这个互联网络上将会形成永久路由环路

- 路由器有一个或多个接口连接到主网上；
- 路由器没有接口连接到主网络上。

在第一种情况下，为了正确地确定报文目标地址的子网，路由器必须使用自身为主网配置的掩码。在第二种情况下，公告信息中仅包含主网络地址，因为路由器不知道使用哪一个子网掩码。

图 11-12 给出了路由器有 4 个接口分别连接到 192.168.0.0 的各子网上。该主网络采用了变长子网划分——两个接口的子网掩码为 27 位，另两个为 30 位。如果路由器运行有类别协议，例如 IGRP，那么它将不能从 27 位掩码推出 30 位掩码的子网，并且也不能从 30 位掩码推出 27 位掩码的子网。因此，协议如何处理冲突的掩码呢？

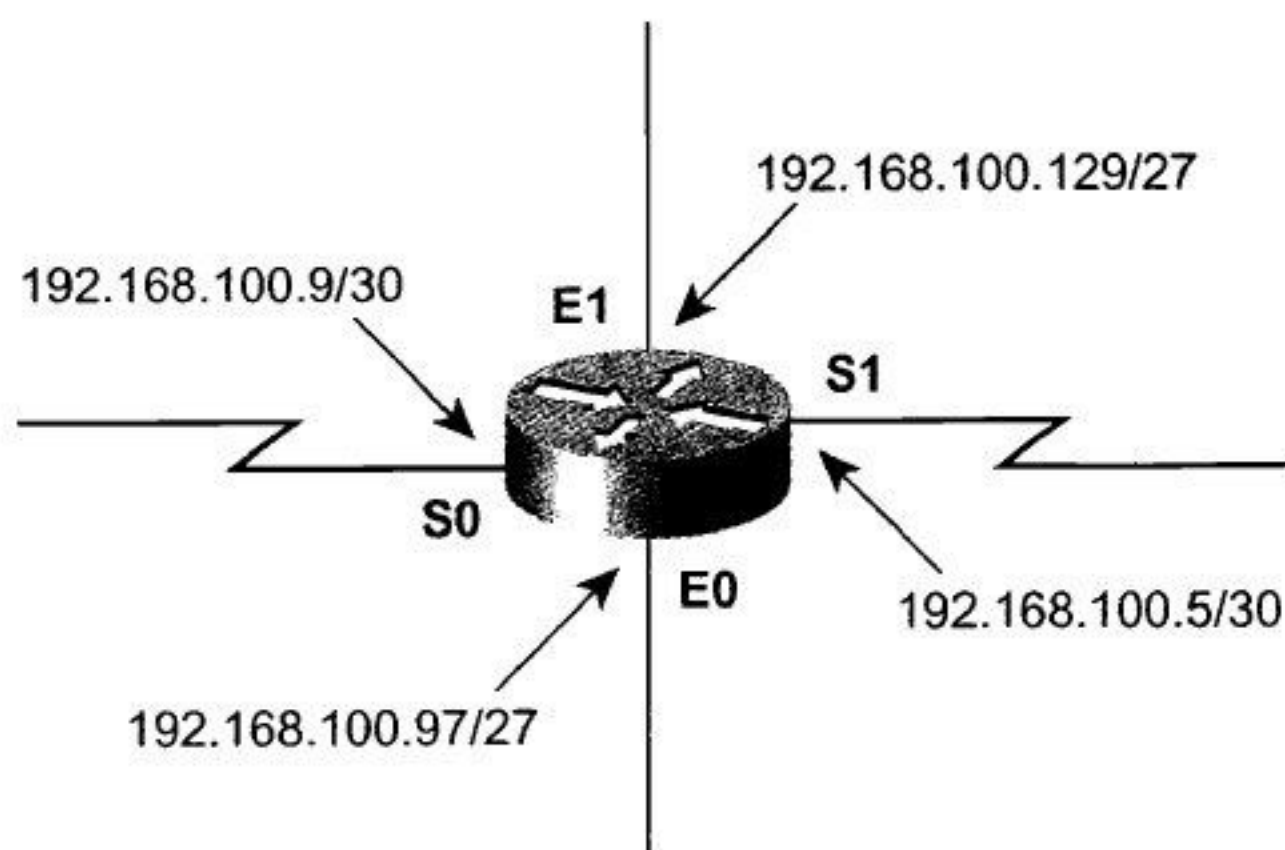


图 11-12 如果路由器运行有类别路由选择协议，它应该选择什么掩码呢？

在图 11-13 中，使用调试手段观察图 11-2 中路由器发出的 IGRP 公告。注意，子网 192.168.100.128/27 的公告从接口 E0 发出，网络使用了 27 位掩码，但是没有从该接口发出 192.168.100.4/30 和 192.168.100.8/30 公告。类似的，192.168.100.8/30 的公告从接口 S1 发出，掩码位 30 位，但是没有从该接口发送 192.168.100.96/27 和 192.168.100.128/27 公告。相同的情形同

样适用于所有 4 个接口。从接口通告的子网仅包括在 192.168.100.0 的子网中, 仅仅那些子网掩码与接口掩码相同的子网, 才会从此接口通告。最后结果是接口 E0 和 E1 的 IGRP 邻居路由器不知道掩码为 30 位的子网, 接口 S0 和 S1 的 IGRP 邻居路由器也不知道掩码为 27 位的子网。

```
O'Neil#debug ip igrp transactions
IGRP protocol debugging is on
O'Neil#
IGRP: sending update to 255.255.255.255 via Ethernet0 (192.168.100.97)
      subnet 192.168.100.128, metric=1100
IGRP: sending update to 255.255.255.255 via Ethernet1 (192.168.100.129)
      subnet 192.168.100.96, metric=1100
IGRP: sending update to 255.255.255.255 via Serial0 (192.168.100.9)
      subnet 192.168.100.4, metric=8476
IGRP: sending update to 255.255.255.255 via Serial1 (192.168.100.5)
      subnet 192.168.100.8, metric=8476
O'Neil#
```

图 11-13 有类别路由选择协议将不在掩码不匹配的接口之间通告路由

仅在掩码相同的接口之间通告路由这一特性, 在从无类别路由选择协议向有类别路由选择协议重新分配时也会使用到。在图 11-14 中, OSPF 域的子网是经过变长子网划分得到的, Paige 将来自 OSPF 的路由信息向 IGRP 重新分配。

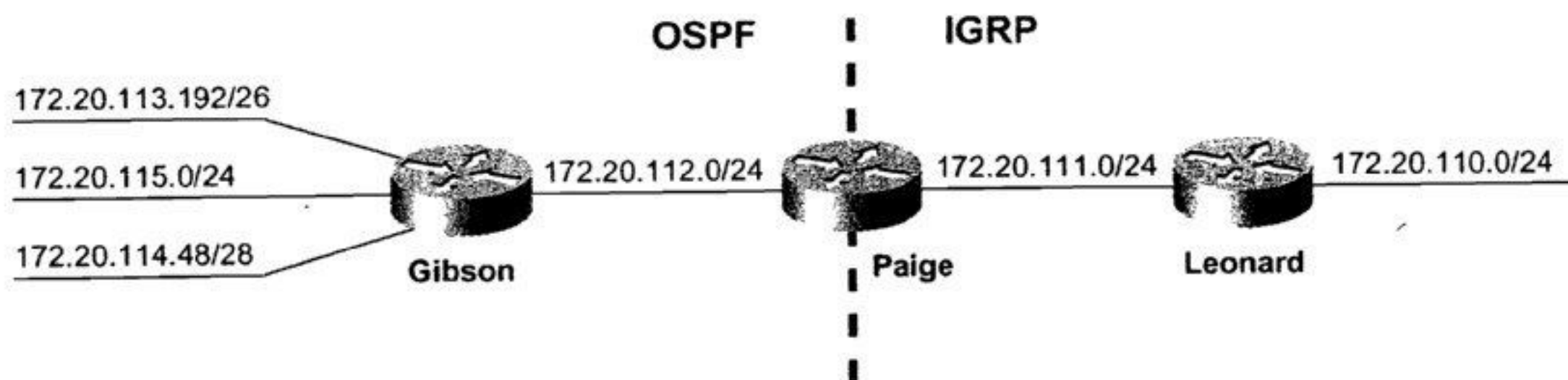


图 11-14 Paige 将来自 OSPF 的路由重新分配进 IGRP

如图 11-15 所示, Paige 知道 OSPF 和 IGRP 域内的所有子网。因为 OSPF 是无类别协议, 所以路由器知道连接到 Gibson 的每个子网的相应掩码。由于 Paige 的 IGRP 进程使用 24 位掩码, 因此 172.20.113.192/26 和 172.20.114.48/28 不一致, 所以不能被通告 (图 11-16)。注意, IGRP 对 172.20.112.0/24 和 172.20.115.0/24 进行通告, 结果在 OSPF 域内 Leonard 仅知道掩码为 24 位的子网 (图 11-17)。

```
Paige#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

172.20.0.0/16 is variably subnetted, 6 subnets, 3 masks
O       172.20.113.192/26 [110/74] via 172.20.112.1, 00:01:35, Ethernet1
C       172.20.112.0/24 is directly connected, Ethernet1
O       172.20.115.0/24 [110/80] via 172.20.112.1, 00:01:35, Ethernet1
I       172.20.110.0/24 [100/1600] via 172.20.111.1, 00:00:33, Ethernet0
C       172.20.111.0/24 is directly connected, Ethernet0
O       172.20.114.48/28 [110/74] via 172.20.112.1, 00:01:35, Ethernet1
Paige#
```

图 11-15 Paige 知道图 11-14 中的所有 6 个子网, 它们来自 OSPF、IGRP 和直接连接


```

Paige#debug ip igrp transactions
IGRP protocol debugging is on
Paige#
IGRP: received update from 172.20.111.1 on Ethernet0
      subnet 172.20.110.0, metric 1600 (neighbor 501)
IGRP: sending update to 255.255.255.255 via Ethernet0 (172.20.111.2)
      subnet 172.20.112.0, metric=1100
      subnet 172.20.115.0, metric=1100
Paige#

```

图 11-16 在来自 OSPF 的路由信息中，仅掩码为 24 位的路由被成功地重新分配进入 IGRP 域，该 IGRP 域也使用 24 位掩码

```

Leonard#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

      172.20.0.0/24 is subnetted, 4 subnets
I       172.20.112.0 [100/1200] via 172.20.111.2, 00:00:48, Ethernet1
I       172.20.115.0 [100/1200] via 172.20.111.2, 00:00:48, Ethernet1
C       172.20.110.0 is directly connected, Ethernet0
C       172.20.111.0 is directly connected, Ethernet1
Leonard#

```

图 11-17 在 OSPF 域内，Leonard 仅知道掩码为 24 位的子网，从该域均不能到达网络 172.20.113.192/26 和 172.20.114.48/28

配置章节包括的案例研究将示范从无类别路由选择协议向有类别路由选择协议可靠地进行重新分配的方法。

11.2 配置重新分配

配置重新分配分为两步：

步骤 1：在路由选择协议中配置接收重新分配的路由，其中使用命令 **redistribute** 指定路由源点。

步骤 2：为重新分配的路由指定度量值。

例如，在图 11-14 中，Paige 的 IGRP 配置如下：

```

router igrp 1
 redistribute ospf 1 metric 10000 100 255 1 1500
 passive-interface Ethernet1
 network 172.20.0.0

```

上面的配置把 OSPF 进程 1 发现的路由向 IGRP 进程 1 重新分配。命令的度量部分为路由分配了 IGRP 度量。按照顺序，命令中各数字分别表示：

- 带宽，单位是 kbit/s；

- 时延, 单位是 10 μ s;
- 可靠性, 为 255 的若干分之一;
- 负载, 为 255 的若干分之一;
- MTU, 单位为 8bit 字节。

Paige 的 OSPF 配置如下:

```
router ospf 1
 redistribute igmp 1 metric 30 metric-type 1 subnets
 network 172.20.112.2 0.0.0.0 area 0
```

上面的配置把 IGRP 进程 1 发现的路由重新分配进入 OSPF 进程 1。命令的度量部分为每一条被重新分配的路由分配度量, 度量是 OSPF 代价值 30。重新分配使得 Paige 成为 OSPF 域的 ASBR, 并且被重新分配的路由是作为外部路由进行通告的。命令的 **metric-type** 部分指明了外部路由的类型为 E1。关键字 **subnets** 仅当向 OSPF 分布路由时使用, 它指明了重新分配的子网细节, 没有它, 仅重新分配主网地址。在案例研究中会更多地讨论关键字 **subnets**。

另一种分配度量的方法是使用 **default-metric** 命令。例如, 前面的 OSPF 配置也可以改写为以下方式:

```
router ospf 1
 redistribute igmp 1 metric-type 1 subnets
 default-metric 30
 network 172.20.112.2 0.0.0.0 area 0
```

该配置同前面配置所产生的结果完全相同。当重新分配来自多个源点的路由时, 命令 **default-metric** 显得十分有用。例如在图 11-14 中, 假设路由器 Paige 不仅运行 IGRP 和 OSPF, 而且还运行 RIP 和 EIGRP。OSPF 的配置如下:

```
router ospf 1
 redistribute igmp 1 metric-type 1 subnets
 redistribute rip metric-type 1 subnets
 redistribute eigrp 2 metric-type 1 subnets
 default-metric 30
 network 172.20.112.2 0.0.0.0 area 0
```

在上面的配置中, 对于所有来自 IGRP、RIP 和 EIGRP 路由, 所分配的度量均为 OSPF 代价 30。

在这里, 两种分配度量的方法也可以相互使用。例如, 假设配置 Paige 向 IGRP 重新分配 OSPF、RIP 和 EIGRP 路由, 但要求对 RIP 路由进行通告时, 所使用的度量要不同于 OSPF 和 EIGRP, 根据需求配置如下:

```
router igmp 1
 redistribute ospf 1
 redistribute rip metric 50000 500 255 1 1500
 redistribute eigrp 2
 default-metric 10000 100 255 1 1500
 passive-interface Ethernet1
 network 172.20.0.0
```


在上面的配置中，首先在命令 **redistribute** 中使用关键字 **metric** 分配度量值，然后使用 **default-metric** 命令分配度量值。来自 RIP 的路由被通告到 IGRP，这些路由所使用的度量在配置行 **redistribute rip** 中指明。而来自 OSPF 和 EIGRP 的路由则使用 **default-metric** 命令指定的度量。

如果关键字 **metric** 和命令 **default-metric** 都没有指定度量，那么被重新分配到 OSPF 的路由的度量缺省值为 20，而其他协议路由度量的缺省值为 0。IS-IS 可以理解 0 度量，但是 RIP 不能，因为它的跳数在 1 到 16 之间。0 度量与 IGRP 和 EIGRP 的多度量格式也不兼容。因此这 3 种协议都必须为重新分配的路由分配合适的度量，否则重新分配将不能进行。下面的案例研究将会分析向各种 IP IGRP 重新分配路由的配置方法。此外，案例中还更多地安排了关于从有类别向无类别、从无类别向无类别以及从无类别向有类别重新分配路由等常见问题的分析。

11.2.1 案例研究：重新分配 IGRP 和 RIP

在图 11-18 的互联网络中，Ford 运行 IGRP，Berra 运行 RIP。Mantle 的路由配置如下：

```
router rip
 redistribute igrp 1 metric 5
 passive-interface Ethernet1
 network 10.0.0.0
!
router igrp 1
 redistribute rip
 default-metric 1000 100 255 1 1500
 passive-interface Ethernet0
 network 10.0.0.0
```

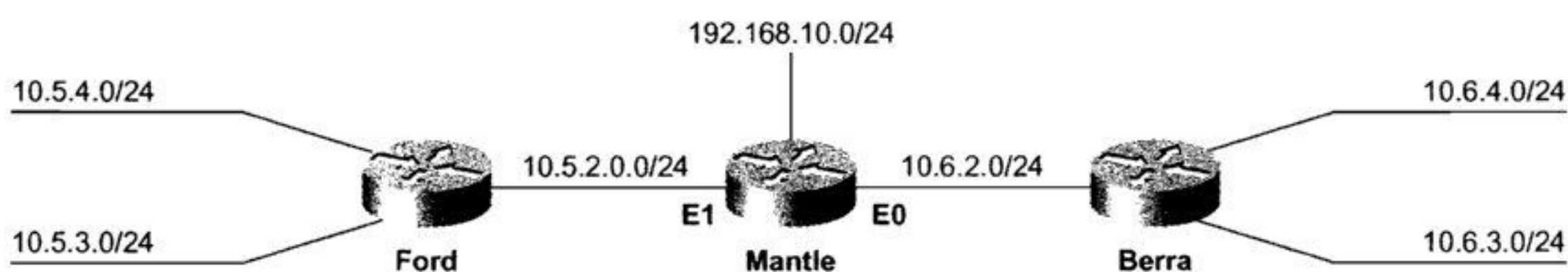


图 11-18 Ford 运行 IGRP，Berra 运行 RIP。Mantle 正在进行路由重新分配

这里同时使用两种分配度量的方法是出于示范目的，在大部分情况下，如果重新分配方案和本例一样简单的话，可以使用一种方法。

注意：Mantle 还被连接到一个末梢网络（192.168.10.0/24）。在本案例中，要求向 IGRP 域通告末梢网络，但是不能向 RIP 域通告。一种实现方法是仅在 IGRP 中添加适当的网络表述。然而，这样做会在末梢网络中造成不必要的 IGRP 广播。另一个实现办法是使用重新发配。

```
router rip
 redistribute igrp 1 metric 5
```



```
passive-interface Ethernet1
network 10.0.0.0
!
router igrp 1
 redistribute connected
 redistribute rip
 default-metric 1000 100 255 1 1500
 passive-interface Ethernet0
 network 10.0.0.0
```

命令 **redistribute connected** 将会重新分配所有直连网络。如果要向 IGRP 域和 RIP 域通告网络 192.168.10.0/24, 那么配置如下:

```
router rip
 redistribute connected metric 5
 redistribute igrp 1 metric 5
 passive-interface Ethernet1
 network 10.0.0.0
!
router igrp 1
 redistribute connected
 redistribute rip
 default-metric 1000 100 255 1 1500
 passive-interface Ethernet0
 network 10.0.0.0
```

11.2.2 案例研究: 重新分配 EIGRP 和 OSPF

图 11-19 的互连网络中, 有一个 OSPF 域和两个 EIGRP 域。路由器 Hodges 运行 OSPF 进程 1, Podres 运行 EIGRP 进程 1, Snider 和 Campanella 运行 EIGRP 进程 2。Robinson 的配置如下:

```
router eigrp 1
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 2
 passive-interface Ethernet0
 network 192.168.3.0
!
router eigrp 2
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 1
 network 192.168.4.0
 network 172.16.0.0
!
router ospf 1
 redistribute eigrp 1 metric 50
 redistribute eigrp 2 metric 100
 network 192.168.3.33 0.0.0.0 area 0
```

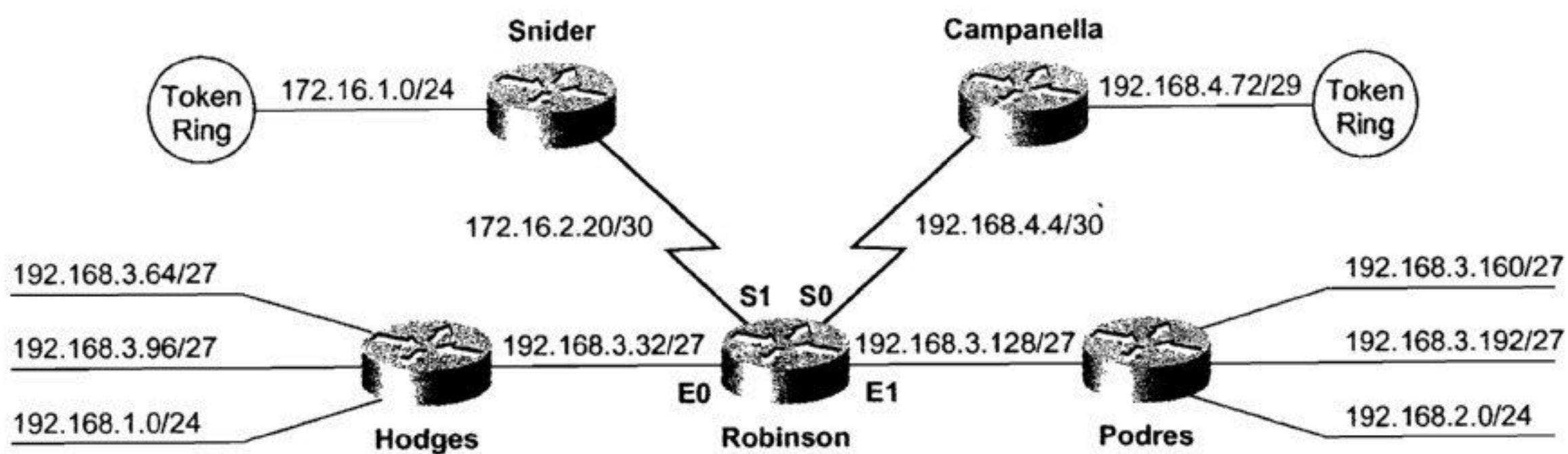



图 11-19 Hodges 运行 OSPF, Podre 运行 EIGRP1, Snider 和 Campanella 运行 EIGRP2

注意: 尽管在 EIGRP 进程之间必须配置重新分配, 但是不需要配置度量。因为这些进程使用相同的度量, 所以能够穿过重新分配边界准确地跟踪度量。图 11-20 显示了 Podres 的路由选择表, 重新分配的路由被标记为 EIGRP 外部路由。

```
Podres#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
       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

Gateway of last resort is not set

D EX 192.168.1.0/24 [170/2611200] via 192.168.3.129, 00:39:14, Ethernet0
C    192.168.2.0/24 is directly connected, Ethernet3
    192.168.3.0/24 is variably subnetted, 7 subnets, 3 masks
D EX  192.168.3.96/27 [170/2611200] via 192.168.3.129, 00:41:18, Ethernet0
D EX  192.168.3.64/27 [170/2611200] via 192.168.3.129, 00:41:18, Ethernet0
D    192.168.3.32/27 [90/307200] via 192.168.3.129, 00:44:06, Ethernet0
D    192.168.3.0/24 is a summary, 00:52:21, Null0
C    192.168.3.192/27 is directly connected, Ethernet2
C    192.168.3.160/27 is directly connected, Ethernet1
C    192.168.3.128/27 is directly connected, Ethernet0
    192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
D EX  192.168.4.72/29 [170/2211584] via 192.168.3.129, 00:07:25, Ethernet0
D EX  192.168.4.4/30 [170/281600] via 192.168.3.129, 00:07:25, Ethernet0
D EX  192.168.4.0/24 [170/2195456] via 192.168.3.129, 00:07:25, Ethernet0
    172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
D EX  172.16.2.20/30 [170/281600] via 192.168.3.129, 00:07:27, Ethernet0
D EX  172.16.0.0/16 [170/2195456] via 192.168.3.129, 00:07:27, Ethernet0
D EX  172.16.1.0/24 [170/2211584] via 192.168.3.129, 00:07:27, Ethernet0
Podres#
```

图 11-20 图 11-19 中 Podres 的路由选择表

图 11-21 显示了 Hodges 的路由选择表, 这里有一些问题。回忆一下第 9 章“开放最短路径优先协议 (OSPF)”, 被重新分配进入 OSPF 域的路由类型可以是类型 1 (E1) 或类型 2 (E2) 外部路由。在这里, 唯有一条指向主网地址 192.168.2.0/24 且标记为 E2 的路由, 好像被重新分配过。造成这种现象的原因是在 Robinson 的配置表述中缺少关键字 **subnets**。如果没有这个关键字, 那么被重新分配的地址仅包括那些没有直接连接到重新分配路由器的主网地址。


```

Hodges#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

C    192.168.1.0/24 is directly connected, Ethernet2
O E2 192.168.2.0/24 [110/50] via 192.168.3.33, 00:11:59, Ethernet0
     192.168.3.0/27 is subnetted, 3 subnets
C      192.168.3.96 is directly connected, Ethernet1
C      192.168.3.64 is directly connected, Ethernet3
C      192.168.3.32 is directly connected, Ethernet0
Hodges#

```

图 11-21 Hodges 的路由选择表仅包括一条被 E2 标记指定的重新分配路由

修改 Robinson 的配置，使其包含关键字 **subnets**:

```

router eigrp 1
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 2
 passive-interface Ethernet0
 network 192.168.3.0
!
router eigrp 2
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 1
 network 192.168.4.0
 network 172.16.0.0
!
router ospf 1
 redistribute eigrp 1 metric 50 subnets
 redistribute eigrp 2 metric 100 subnets
 network 192.168.3.33 0.0.0.0 area 0

```

修改的结果是图 11-19 中的所有子网都出现在 Hodges 的路由选择表中 (图 11-22)。

```

Hodges#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

C    192.168.1.0/24 is directly connected, Ethernet2
O E2 192.168.2.0/24 [110/50] via 192.168.3.33, 00:17:31, Ethernet0
     192.168.3.0/27 is subnetted, 6 subnets

```

待续


```

C      192.168.3.96 is directly connected, Ethernet1
C      192.168.3.64 is directly connected, Ethernet3
C      192.168.3.32 is directly connected, Ethernet0
O E2   192.168.3.192 [110/50] via 192.168.3.33, 00:02:51, Ethernet0
O E2   192.168.3.160 [110/50] via 192.168.3.33, 00:02:51, Ethernet0
O E2   192.168.3.128 [110/50] via 192.168.3.33, 00:00:36, Ethernet0
      192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
O E2   192.168.4.72/29 [110/100] via 192.168.3.33, 00:00:19, Ethernet0
O E2   192.168.4.4/30 [110/100] via 192.168.3.33, 00:00:19, Ethernet0
O E2   192.168.4.0/24 [110/100] via 192.168.3.33, 00:00:19, Ethernet0
      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
O E2   172.16.2.20/30 [110/100] via 192.168.3.33, 00:00:20, Ethernet0
O E2   172.16.0.0/16 [110/100] via 192.168.3.33, 00:00:20, Ethernet0
O E2   172.16.1.0/24 [110/100] via 192.168.3.33, 00:00:20, Ethernet0
Hodges#

```

图 11-22 在关键字 **subnets** 被添加到 Robinson 的重新分配配置中后, Hodges 便可以知道所有子网信息

缺省情况下, 外部路由作为类型 2 路由被重新分配到 OSPF。正如第 9 章所讨论的, E2 路由仅包括路由的外部代价。如图 11-23 所示, 当存在不止一条外部路由可以到达单一目标网络时, 这一事实将显得十分重要。在图 11-23 的互联网络中, 一台路由器正在重新分配代价为 50, 指向 10.2.3.0/24 的路由, 另一台路由器也正在重新分配代价为 100 并且指向相同目标网络的另一条路由。如果这条路由被作为 E2 通告, 那么在 OSPF 域内的链路代价将不会被计入。结果在 OSPF 域内的路由器将会选择路由 1 到达 10.2.3.0/24。

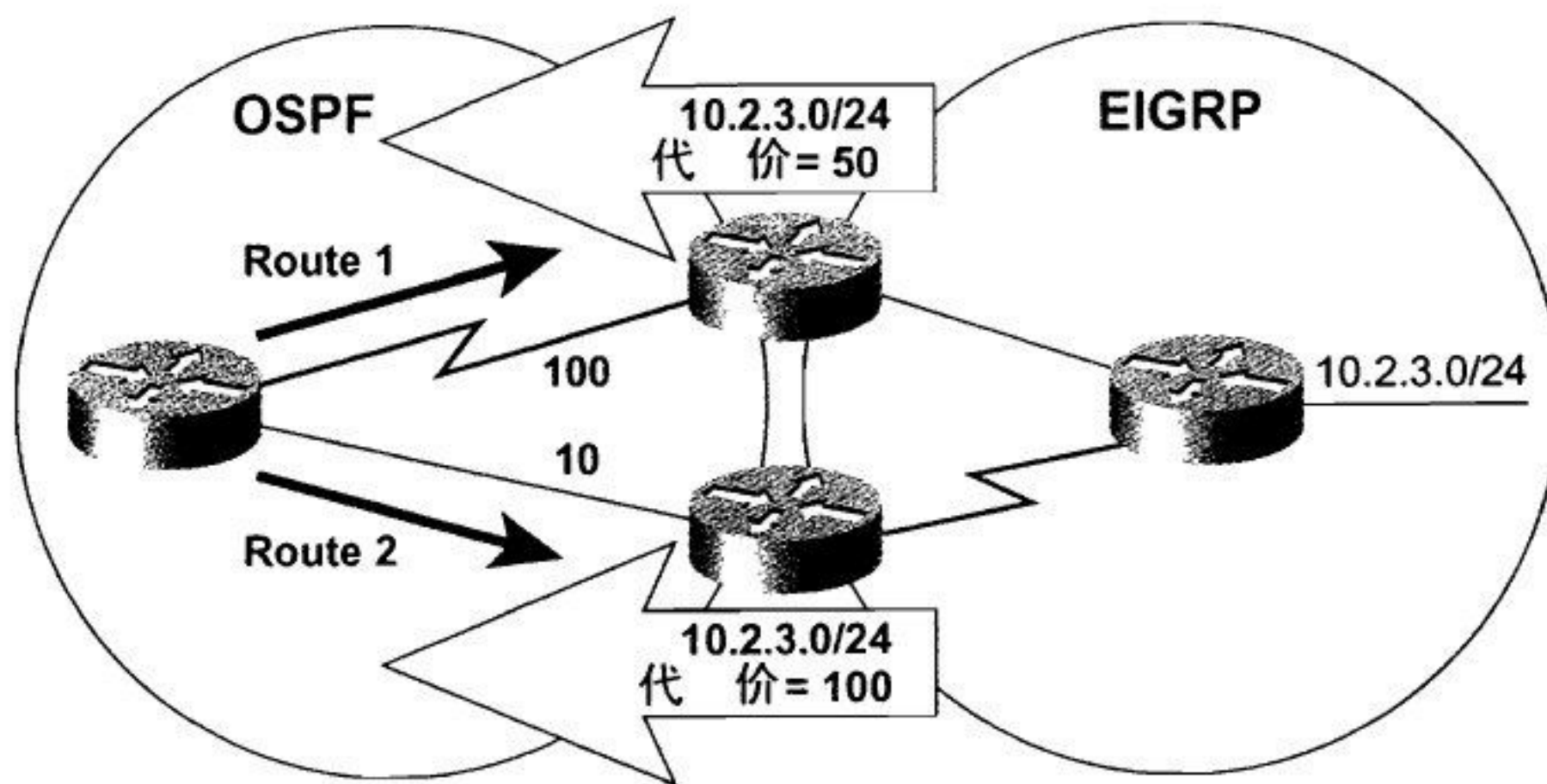


图 11-23 如果去往 10.2.3.0/24 的路由被作为 E2 通告, 那么路由 1 的代价将会是 1, 路由 2 的代价为 100。

如果该路由被作为 E1 通告, 那么路由 1 的代价为 150, 路由 2 的代价为 110

如果在图 11-23 中, 指向 10.2.3.0/24 的路由被作为 E1 重新分配, 那么在 OSPF 域内的链路代价将被计入重新分配代价。结果, OSPF 域内的路由器将选择路由 2, 代价为 110 (100 + 10); 而不是路由 1, 代价为 150 (100 + 50)。

图 11-19 中的路由器 Robinson 正在重新分配代价为 50 的 EIGRP 1 和代价为 100 的 EIGRP 2。图 11-22 显示出, 在 Hodges, 指向 EIGRP 1 子网的路由代价仍然为 50, 而指向 EIGRP 2 子网的路由代价为 100。因此在 Hodges 和 Robinson 之间的以太网链路代价没有被计入。

为了将路由作为 E1 重新分配到 OSPF, 可以在重新分配命令中添加关键字 **metric-type 1**。在下面的配置中, Robinson 继续把 EIGRP 1 作为 E2 重新分配, 但是把 EIGRP 2 作为 E1 重新分配:

```
router eigrp 1
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 2
 passive-interface Ethernet0
 network 192.168.3.0
!
router eigrp 2
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 1
 network 192.168.4.0
 network 172.16.0.0
!
router ospf 1
 redistribute eigrp 1 metric 50 subnets
 redistribute eigrp 2 metric 100 metric-type 1 subnets
 network 192.168.3.33 0.0.0.0 area 0
```

在重新配置 Robinson 之后, 图 11-24 给出了 Hodges 的路由选择表。在 EIGRP 1 域内所有指向目标网络的路由代价仍旧为 50, 但是在 EIGRP 2 域内指向目标网络的路由代价现在为 110 (重新分配代价加上 Robinson 和 Hodges 之间以太网链路的缺省代价 10)。

```
Hodges#sh ip rou
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

C    192.168.1.0/24 is directly connected, Ethernet2
O E2 192.168.2.0/24 [110/50] via 192.168.3.33, 00:21:20, Ethernet0
     192.168.3.0/27 is subnetted, 6 subnets
C     192.168.3.96 is directly connected, Ethernet1
C     192.168.3.64 is directly connected, Ethernet3
C     192.168.3.32 is directly connected, Ethernet0
O E2 192.168.3.192 [110/50] via 192.168.3.33, 00:06:40, Ethernet0
O E2 192.168.3.160 [110/50] via 192.168.3.33, 00:06:40, Ethernet0
O E2 192.168.3.128 [110/50] via 192.168.3.33, 00:04:24, Ethernet0
     192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
O E1 192.168.4.72/29 [110/110] via 192.168.3.33, 00:00:54, Ethernet0
O E1 192.168.4.4/30 [110/110] via 192.168.3.33, 00:00:54, Ethernet0
O E1 192.168.4.0/24 [110/110] via 192.168.3.33, 00:00:54, Ethernet0
     172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
O E1 172.16.2.20/30 [110/110] via 192.168.3.33, 00:00:55, Ethernet0
O E1 172.16.0.0/16 [110/110] via 192.168.3.33, 00:00:55, Ethernet0
O E1 172.16.1.0/24 [110/110] via 192.168.3.33, 00:00:55, Ethernet0
Hodges#
```

图 11-24 修改 Robinson 的配置以便在通告时把子网 192.168.4.0 和 172.16.0.0 作为类型 1 外部路由

11.2.3 案例研究：重新分配和路由汇总

Cisco 的 EIGRP、OSPF 和 IS-IS 的实现都可以对被重新分配的路由进行汇总。这个案例研究将分析对 EIGRP 和 OSPF 的汇总；下面的案例研究将研究 IS-IS 的汇总。

第一个要注意的事情是汇总要起作用，先决条件是 IP 子网地址已为汇总进行过规划。例如，在图 11-19 中 OSPF 域内，192.168.3.0 的子网全部都被汇总地址 192.168.3.0/25 所包含。在 EIGRP 1 域内相同主网地址的所有子网也都被汇总地址 192.168.3.128/25 覆盖。如果子网 192.168.3.0/27 被连接到 Podres，那么必须从汇总地址中将这个单一目标分离出来之后，才能进行通告。虽然通告单一目标几乎不会有什么不利影响，但是通告大量汇总地址范围之外的子网将会减少汇总的好处。

命令 **summary-address** 为 OSPF 进程指定了一个汇总地址和掩码。任何落在指定汇总地址范围内的更精确的地址都会被禁止。注意，此命令仅用在 ASBR 汇总外部路由；在 ABR 内部 OSPF 路由的汇总可以通过命令 **area range** 实现，详见第 9 章。

在图 11-19 中路由器 Robinson 上，EIGRP 1 的子网在进入 OSPF 域时被汇总为 192.168.3.128/25，EIGRP 2 的子网被汇总为 172.16.0.0/16：

```
router eigrp 1
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 2
 passive-interface Ethernet0
 network 192.168.3.0
!
router eigrp 2
 redistribute eigrp 1
 network 192.168.4.0
 network 172.16.0.0
!
router ospf 1
 summary-address 192.168.3.128 255.255.255.128
 summary-address 172.16.0.0 255.255.0.0
 redistribute eigrp 1 metric 50 subnets
 redistribute eigrp 2 metric 100 metric-type 1 subnets
 network 192.168.3.33 0.0.0.0 area 0
```

比较图 11-25 和 11-24，在图 11-25 中，Hodges 的路由选择表包含指定的汇总地址，汇总地址范围内的子网地址在重新分配点被禁止。注意，由于没有对 192.168.4.0/24 进行汇总配置，所以该主网地址的所有子网仍出现在路由选择表中。

对于 EIGRP 的汇总是指定接口的。也就是不在路由进程下指明汇总地址和掩码，而是在独立的接口下指明。这个系统提供了更大的灵活性，可以在同一进程的不同接口通告不同的汇总地址。命令 **ip summary-address eigrp process-id** 指定了汇总地址、掩码和汇总所要通告的 EIGRP 进程。

在下面的配置中，Robinson 将向 EIGRP 1 通告总结地址 192.168.3.0/25、172.16.0.0/16 和 192.168.4.0/24：


```

Hodges#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

C    192.168.1.0/24 is directly connected, Ethernet2
O E2 192.168.2.0/24 [110/50] via 192.168.3.33, 00:25:01, Ethernet0
      192.168.3.0/24 is variably subnetted, 4 subnets, 2 masks
C      192.168.3.96/27 is directly connected, Ethernet1
C      192.168.3.64/27 is directly connected, Ethernet3
C      192.168.3.32/27 is directly connected, Ethernet0
O E2 192.168.3.128/25 [110/50] via 192.168.3.33, 00:01:45, Ethernet0
      192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
O E1 192.168.4.72/29 [110/110] via 192.168.3.33, 00:04:35, Ethernet0
O E1 192.168.4.4/30 [110/110] via 192.168.3.33, 00:04:35, Ethernet0
O E1 192.168.4.0/24 [110/110] via 192.168.3.33, 00:04:36, Ethernet0
O E1 172.16.0.0/16 [110/110] via 192.168.3.33, 00:04:36, Ethernet0
Hodges#

```

图 11-25 Robinson 正在汇总 192.168.3.128/25 和 172.16.0.0/26, 因此在 Hodges 的路由选择表中不会出现在此范围内的更精确的地址

```

interface Ethernet0
 ip address 192.168.3.33 255.255.255.224
!
interface Ethernet1
 ip address 192.168.3.129 255.255.255.224
 ip summary-address eigrp 1 192.168.3.0 255.255.255.128
 ip summary-address eigrp 1 172.16.0.0 255.255.0.0
 ip summary-address eigrp 1 192.168.4.0 255.255.255.0
!
interface Serial0
 ip address 192.168.4.5 255.255.255.252
 ip summary-address eigrp 2 192.168.3.0 255.255.255.0
!
interface Serial1
 ip address 172.16.2.21 255.255.255.252
 ip summary-address eigrp 2 192.168.0.0 255.255.0.0
!
router eigrp 1
 redistribute ospf 1 metric 1000 100 1 255 1500
 redistribute eigrp 2
 passive-interface Ethernet0
 network 192.168.3.0
!
router eigrp 2
 redistribute eigrp 1
 network 192.168.4.0
 network 172.16.0.0
!

```



```

router ospf 1
summary-address 192.168.3.128 255.255.255.128
summary-address 172.16.0.0 255.255.0.0
redistribute eigrp 1 metric 50 subnets
redistribute eigrp 2 metric 100 metric-type 1 subnets
network 192.168.3.33 0.0.0.0 area 0

```

图 11-26 给出了 Podres 的路由选择表。正如 OSPF 汇总一样，EIGRP 的汇总禁止通告汇总范围以内的子网。但与 OSPF 不一样的是，Podres 的路由选择表显示向 EIGRP 通告的汇总路由没有被标记为外部路由。

Robinson 正在向 Campanella 通告 EIGRP 汇总路由 192.168.3.0/24，同时向 Snider 通告 192.168.0.0/16。图 11-27 给出了 Campanella 的路由选择表，图 11-28 给出了 Snider 的路由选择表。

在 Snider 路由选择表中令人感兴趣的是指向 192.168.4.0/24 的路由条目。你可能会预料到这条路由被汇总地址 192.168.0.0/16 所禁止。然而，192.168.4.0/24 在 EIGRP 2 进程域内部；汇总仅应用于被重新分配进入进程域的路由。

回过头再看图 11-26，注意指向 192.168.3.128/25 的汇总路由，它可能会使你感到惊讶，因为汇总地址被通告到 OSPF，而不是 EIGRP。另外还要注意，这条路由被标记为外部路由，这表明它已经被重新分配到 EIGRP 域了。这里所发生的情况是：汇总路由被通告到 OSPF，接着又从 OSPF 域被重新分配到 EIGRP。所以在 Podres 出现了不期望的路由条目。

```

Podres#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
       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

Gateway of last resort is not set

D EX 192.168.1.0/24 [170/2611200] via 192.168.3.129, 00:00:52, Ethernet0
C    192.168.2.0/24 is directly connected, Ethernet3
     192.168.3.0/24 is variably subnetted, 6 subnets, 3 masks
D    192.168.3.0/24 is a summary, 00:00:52, Null0
D    192.168.3.0/25 [90/307200] via 192.168.3.129, 00:00:52, Ethernet0
C    192.168.3.192/27 is directly connected, Ethernet2
C    192.168.3.160/27 is directly connected, Ethernet1
D EX 192.168.3.128/25 [170/2611200] via 192.168.3.129, 00:00:52, Ethernet0
C    192.168.3.128/27 is directly connected, Ethernet0
D    192.168.4.0/24 [90/281600] via 192.168.3.129, 00:00:52, Ethernet0
D    172.16.0.0/16 [90/281600] via 192.168.3.129, 00:00:52, Ethernet0
D EX 192.168.0.0/16 [170/281600] via 192.168.3.129, 00:00:53, Ethernet0
Podres#

```

图 11-26 Podres 的路由选择表显示了汇总路由 192.168.3.0/25、192.168.4.0/24 和 192.172.16.0.0/16

现在假设子网 192.168.3.192/27 变为不可访问。Podres 将按照较精确的路由 192.168.3.128/25 转发去往该子网的报文。报文将被发送到 OSPF 域，你可能会认为在 OSPF 域中汇总路由 192.168.3.128/25 将导致报文被送回 Podres。

事实上，这种情形不会发生，Robinson 的路由选择表（见图 11-29）中有许多汇总路由

条目都把 Null0 接口作为连接接口。空接口是一个不知道去哪里的软件接口——路由到它的报文将会被丢弃。除了某些特例外,¹每当路由器产生一条汇总路由,路由器同时还会生成一条指向空接口的路由。如果 Robinson 接收到一个去往 192.168.3.192/27 的报文并且该子网不再可达,那么路由器将转发报文至空接口。路由环路将在这一跳被打断。

```
Campanella#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
       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

Gateway of last resort is not set

D EX 192.168.2.0/24 [170/2323456] via 192.168.4.5, 00:03:15, Serial0
D    192.168.3.0/24 [90/2169856] via 192.168.4.5, 00:04:26, Serial0
    192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.4.72/29 is directly connected, TokenRing0
C    192.168.4.4/30 is directly connected, Serial0
D    172.16.0.0/16 [90/2681856] via 192.168.4.5, 00:03:13, Serial0
Campanella#
```

图 11-27 在 Robinson 配置汇总之后的 Campanellade 路由选择表

```
Snider#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
       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

Gateway of last resort is not set

D    192.168.4.0/24 [90/2681856] via 172.16.2.21, 00:05:26, Serial0
    172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C    172.16.2.20/30 is directly connected, Serial0
D    172.16.0.0/16 is a summary, 00:05:24, Null0
C    172.16.1.0/24 is directly connected, TokenRing0
D    192.168.0.0/16 [90/2169856] via 172.16.2.21, 00:07:37, Serial0
Snider#
```

图 11-28 在 Robinson 配置汇总之后的 Snider 路由选择表

指向空接口的汇总路由对于防止环路非常有用,关于空接口的使用详见第 12 章。然而,重新分配不正确的路由信息是一点都不允许发生的。假设 Podres 到 Robinson 不是 1 跳而是 10 跳,那么方向错误的报文要经过很长的路线之后才会被丢弃。这个例子证明了在互相进行重新分配时(也就是当两个路由选择协议互相向对方重新分配它们各自的路由时)需要仔细地控制路由通告。在这种情况下,使用路由过滤(详见第 13 章)或路由图(详见第 14 章)是绝对必要的。

前面的情景还展示了为使用汇总而付出的代价。虽然路由选择表的尺寸被减少,节约了内存和处理器循环周期,但路由的精度也被降低了,随着互联网络变得更加复杂,细节的损

¹ 例如,OSPF 内部区域汇总并不自动生成到空接口的汇总路由。它必须被静态配置,见第 9 章示例。

失将会增加路由错误的可能性。

```

Robinson#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    192.168.1.0/24 [110/74] via 192.168.3.34, 02:28:09, Ethernet0
D    192.168.2.0/24 [90/409600] via 192.168.3.130, 02:04:15, Ethernet1
    192.168.3.0/24 is variably subnetted, 9 subnets, 4 masks
O    192.168.3.96/27 [110/11] via 192.168.3.34, 02:28:09, Ethernet0
O    192.168.3.64/27 [110/74] via 192.168.3.34, 02:28:09, Ethernet0
C    192.168.3.32/27 is directly connected, Ethernet0
D    192.168.3.0/24 is a summary, 00:58:14, Null0
D    192.168.3.0/25 is a summary, 00:58:14, Null0
D    192.168.3.192/27 [90/435200] via 192.168.3.130, 02:04:15, Ethernet1
D    192.168.3.160/27 [90/460800] via 192.168.3.130, 02:04:15, Ethernet1
O    192.168.3.128/25 is a summary, 01:21:18, Null0
C    192.168.3.128/27 is directly connected, Ethernet1
    192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
D    192.168.4.72/29 [90/2185984] via 192.168.4.6, 00:58:15, Serial0
C    192.168.4.4/30 is directly connected, Serial0
D    192.168.4.0/24 is a summary, 01:21:08, Null0
    172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C    172.16.2.20/30 is directly connected, Serial1
D    172.16.0.0/16 is a summary, 00:58:16, Null0
D    172.16.1.0/24 [90/2185984] via 172.16.2.22, 01:21:10, Serial1
D    192.168.0.0/16 is a summary, 01:21:08, Null0
Robinson#

```

图 11-29 Robinson 的路由选择表。由于路由器产生了许多汇总路由，因此相应地有许多连接到空接口的汇总路由条目。这样可以防止路由环路

11.2.4 案例研究：重新分配 IS-IS 和 RIP

在图 11-30 的互连网络中，Aaron 运行 IS-IS，Williams 运行 RIPv1，Mays 正在进行重新分配。Mays 的 IS-IS 配置如下：

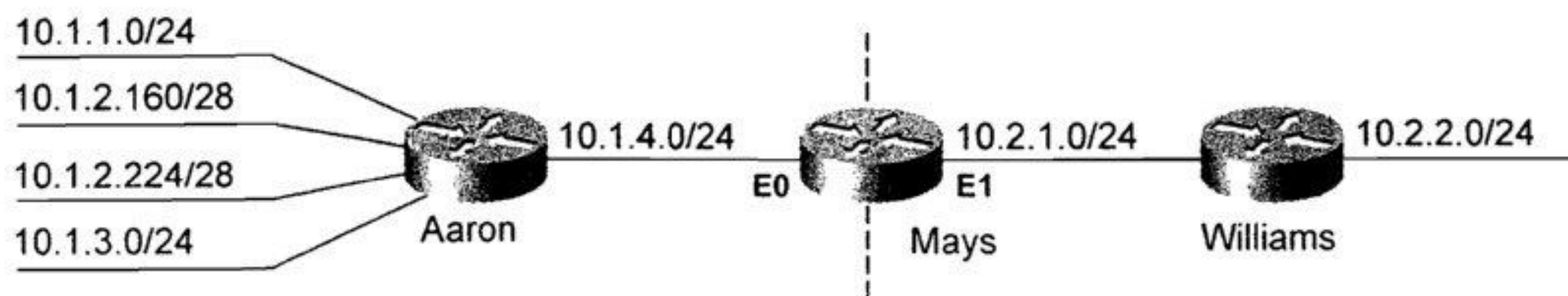


图 11-30 路由器 Mays 正在向 IS-IS 重新分配 RIP 路由，同时向 RIP 重新分配 IS-IS 路由


```

router isis
 redistribute rip metric 0 metric-type internal level-2
 net 01.0001.0000.0c76.5432.00
!
router rip
 redistribute isis level-1-2 metric 1
 passive-interface Ethernet0
 network 10.0.0.0

```

路由可能被作为内部或外部路由（缺省是内部）、1 级或 2 级（缺省是 1 级）路由向 IS-IS 重新分配。在所示例子中，把 RIP 路由作为内部 2 级路由，然后使用缺省度量 0 对路由进行重新分配。图 11-31 显示了在 Aaron 路由选择表中被重新分配的路由。

```

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

Gateway of last resort is not set

 10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C       10.1.3.0/24 is directly connected, Ethernet4
i L2    10.2.1.0/24 [115/10] via 10.1.4.2, Ethernet0
i L2    10.2.2.0/24 [115/10] via 10.1.4.2, Ethernet0
C       10.1.1.0/24 is directly connected, Ethernet1
C       10.1.4.0/24 is directly connected, Ethernet0
C       10.1.2.160/28 is directly connected, Ethernet2
C       10.1.2.224/28 is directly connected, Ethernet3
Aaron#

```

图 11-31 Aaron 的路由选择表显示了被重新分配的 RIP 路由

由于 RIP 路由在 IS-IS 路由选择域外部，为了最好地反映这一点，因此把 RIP 路由作为外部路由重新分配进入该域：

```

router isis
 redistribute rip metric 0 metric-type external level-2
 net 01.0001.0000.0c76.5432.00
!
router rip
 redistribute isis level-1-2 metric 1
 passive-interface Ethernet0
 network 10.0.0.0

```

图 11-32 给出了配置修改过后的 Aaron 的路由选择表。对图 11-31 惟一的改动就是将被重新分配路由的度量提高到大于 64，这指明它是外部路由（在这个小互连网络中）。

再次查看图 11-30 发现，RIP 域的两个子网被汇总为单一地址 10.2.0.0/16。进入 IS-IS 的汇总路由所使用的配置命令与 OSPF 相同，都是 **summary-address**。但是，还必须要指定汇总路由要送达的路由级别。在下面的配置中，RIP 路由作为 1 级路由被重新分配和汇总。


```

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

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.1.3.0/24 is directly connected, Ethernet4
i L2    10.2.1.0/24 [115/138] via 10.1.4.2, Ethernet0
i L2    10.2.2.0/24 [115/138] via 10.1.4.2, Ethernet0
C       10.1.1.0/24 is directly connected, Ethernet1
C       10.1.4.0/24 is directly connected, Ethernet0
C       10.1.2.160/28 is directly connected, Ethernet2
C       10.1.2.224/28 is directly connected, Ethernet3
Aaron#

```

图 11-32 在把路由作为外部路由通告之后，指向 10.2.1.0/24 和 10.2.2.0/24 的路由度量被改为 138

```

router isis
summary-address 10.2.0.0 255.255.0.0 level-1
redistribute rip metric 0 metric-type external level-1
net 01.0001.0000.0c76.5432.00
!
router rip
redistribute isis level-1-2 metric 1
passive-interface Ethernet0
network 10.0.0.0

```

图 11-33 给出了在 Aaron 路由选择表中的汇总路由。同 OSPF 和 EIGRP 一样，汇总导致汇总范围以内更精确的路由被禁止。

```

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

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
i L1    10.2.0.0/16 [115/138] via 10.1.4.2, Ethernet0
C       10.1.3.0/24 is directly connected, Ethernet4
C       10.1.1.0/24 is directly connected, Ethernet1
C       10.1.4.0/24 is directly connected, Ethernet0
C       10.1.2.160/28 is directly connected, Ethernet2
C       10.1.2.224/28 is directly connected, Ethernet3
Aaron#

```

图 11-33 Aaron 路由选择表中有汇总路由指向 RIP 域内的子网

在向另一种协议重新分配 IS-IS 时，必须指明路由级别。到目前为止在所给的例子中，

对于向 RIP 重新分配的路由, 1 级和 2 级路由都指定过它们。

11.2.5 案例研究: 重新分配静态路由

图 11-34 给出了在图 11-30 中 Williams 的路由选择表。注意缺少了子网 10.1.2.160/28 和 10.1.2.224/28。这些子网的掩码与配置在 Mays 接口 E1 上的 24 位掩码不一致, 所以从该接口发送的 RIP 更新信息中没有包含这些路由。这个例子再一次说明了从无类别协议向有类别协议重新分配变长子网路由所存在的问题, 这在本章开头曾讨论过。

```
Williams#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
       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

Gateway of last resort is not set

    10.0.0.0/8 is subnetted, 5 subnets
R       10.1.3.0 [120/1] via 10.2.1.2, 00:00:01, Ethernet0
C       10.2.1.0 is directly connected, Ethernet0
R       10.1.1.0 [120/1] via 10.2.1.2, 00:00:02, Ethernet0
C       10.2.2.0 is directly connected, Ethernet1
R       10.1.4.0 [120/1] via 10.2.1.2, 00:00:02, Ethernet0
Williams#
```

图 11-34 没有向 RIP 域重新分配掩码不是 24 位的子网路由

解决这个问题的一种方案是使用 24 位地址 10.1.2.0/24 汇总两个 28 位子网。因为 RIP 没有这个汇总命令, 所以实现该汇总的办法是配置一条指向汇总地址的静态路由, 接着向 RIP 重新分配该路由。

```
router isis
summary-address 10.2.0.0 255.255.0.0 level-1
redistribute rip metric 0 metric-type external level-1
net 01.0001.0000.0c76.5432.00
!
router rip
redistribute static metric 1
redistribute isis level-1-2 metric 1
passive-interface Ethernet0
network 10.0.0.0
!
ip route 10.1.2.0 255.255.255.0 10.1.4.1
```

图 11-35 给出了 Williams 的路由选择表, 其中包含了该总结路由。

正如第 3 章“静态路由”所讨论的, 静态路由的一种变化形式是使用出站端口代替路由条目中的下一跳地址。这种静态路由也可以被重新分配, 但是配置上稍微有一点不同。例如, Mays 的配置如下:


```

Williams#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
       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

Gateway of last resort is not set

10.0.0.0/8 is subnetted, 6 subnets
R      10.1.3.0 [120/1] via 10.2.1.2, 00:00:03, Ethernet0
R      10.1.2.0 [120/1] via 10.2.1.2, 00:00:03, Ethernet0
C      10.2.1.0 is directly connected, Ethernet0
R      10.1.1.0 [120/1] via 10.2.1.2, 00:00:03, Ethernet0
C      10.2.2.0 is directly connected, Ethernet1
R      10.1.4.0 [120/1] via 10.2.1.2, 00:00:03, Ethernet0
Williams#

```

图 11-35 使用 10.1.2.0/24 汇总子网 10.1.2.160/28 和 10.1.2.224/28

```

router isis
summary-address 10.2.0.0 255.255.0.0 level-1
redistribute rip metric 0 metric-type external level-1
net 01.0001.0000.0c76.5432.00
!
router rip
redistribute isis level-1-2 metric 1
passive-interface Ethernet0
network 10.0.0.0
!
ip route 10.1.2.0 255.255.255.0 Ethernet0

```

这里的静态路由指向 Mays 的接口 E0，而不是指向下一跳地址 10.1.4.1。由于在 RIP 的配置模式下不再使用命令 **redistribute static**，所以 Williams 的路由选择表看上去和图 11-35 一样。

这个静态路由仍然会被重新分配的原因是当静态路由指向出站接口时，目标网络被认为是直接连接到路由器（图 11-36），又因为网络 10.0.0.0 的表述出现在 RIP 的配置中，所以 RIP 将通告 10.0.0.0 的直连子网。

假设 Williams 接收到报文的目的地址是 10.1.2.5，并且报文匹配到汇总地址 10.1.2.0/24，那么报文将被转发给 Mays。在 Mays，目的地址不能匹配到更精确的子网，因而最终匹配到这条静态路由。Mays 将在接口 E0 发送 ARP 请求，试图发现主机 10.1.2.5（或者发现将发送一个代理 ARP 回应的路由器）。如果没有发现，那么路由器不知道该如何处理此报文。ICMP 目标不可达的信息将不会被发送给源点。

回忆一下，当使用汇总命令时，它们会在路由选择表中创建一个指向空接口的路由条目。对于静态路由，也应该做同样的工作：

```

router isis
summary-address 10.2.0.0 255.255.0.0 level-1
redistribute rip metric 0 metric-type external level-1
net 01.0001.0000.0c76.5432.00
!

```



```

router rip
 redistribute isis level-1-2 metric 1
 passive-interface Ethernet0
 network 10.0.0.0
!
ip route 10.1.2.0 255.255.255.0 Null0

```

现在, 在路由器 Mays, 任何不能发现最精确匹配的目的地地址都会被路由到空接口丢弃, 同时目标不可达的 ICMP 消息将会被发送给源点。

```

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

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
i L1  10.1.3.0/24 [115/20] via 10.1.4.1, Ethernet0
S      10.1.2.0/24 is directly connected, Ethernet0
C      10.2.1.0/24 is directly connected, Ethernet1
i L1  10.1.1.0/24 [115/20] via 10.1.4.1, Ethernet0
R      10.2.2.0/24 [120/1] via 10.2.1.1, 00:00:21, Ethernet1
C      10.1.4.0/24 is directly connected, Ethernet0
i L1  10.1.2.160/28 [115/20] via 10.1.4.1, Ethernet0
i L1  10.1.2.224/28 [115/20] via 10.1.4.1, Ethernet0
Mays#

```

图 11-36 Mays 认为汇总地址 10.1.2.0/24 被直接连接到接口 Ethernet 0 上

11.3 展 望

本章讨论了在重新分配路由时会出现的几个问题。为了避免和纠正故障, 除了最简单的重新分配方案之外, 几乎在所有的方案中都常常包括对路由过滤和路由图的使用, 它们分别第 13 章和第 14 章讨论。这些章节包括了更加复杂的重新分配方案以及如何故障诊断。首先, 第 12 章将研究缺省路由——缺省路由被认为是汇总路由的最普遍的形式。

11.4 总结表: 第 11 章命令回顾

命 令	描 述
default-metric <i>bandwidth deafly reliability load mtu</i>	为重新分配到 IGRP 和 EIGRP 的路由指定缺省度量
default-metric <i>number</i>	为重新分配到 OSPF 和 RIP 的路由指定缺省度量
ip summary-address <i>eigrp autonomous-system-number address mask</i>	在接口上配置一条 EIGRP 汇总路由
redistribute <i>connected</i>	重新分配所有直连网络

续表

命 令	描 述
<code>redistribute protocol [process-id] {level-1 level-1-2 level-2} [metric metric-value][metric-type type-value][match {internal external 1 external 2}][tag tag-value] [route-map map-tag][weight weight][subnets]</code>	将重新分配配置到路由选择协议，并且指定重新分配路由的源
<code>summary-address address mask {level-1 level-1-2 level-2} prefix mask [not-advertise] [tag tag]</code>	为 IS-IS 和 OSPF 配置路由汇总

11.5 复 习 题

- 1. 从什么样的源点可以重新分配路由？
- 2. 管理距离的目的是什么？
- 3. 在重新分配时管理距离是如何导致故障的？
- 4. 从无类别路由选择协议向有类别路由选择协议重新分配是怎样导致故障的？
- 5. 哪一种 IP 的 IGP 可以使用缺省重新分配度量，为了重新分配工作正常，哪一种 IGP 必须配置度量？
- 6. 使用带关键字 **metric** 的 **redistribute** 命令和 **default-metric** 命令有什么区别？
- 7. 在重新分配 OSPF 时，关键字 **subnets** 的作用是什么？
- 8. 在汇总路由时空接口是如何起作用的？

11.6 配置练习

1. 在图 11-37 中，路由器 A 运行 IGRP，路由器 C 运行 RIPv1。为了使所有子网相互连通，请给出路由器 B 的配置。

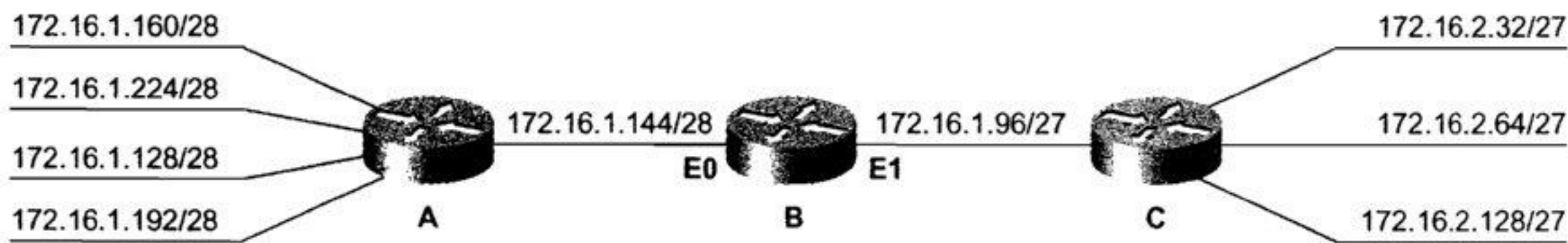


图 11-37 配置练习 1、2 和 3 所使用的互连网络

- 2. 在图 11-37 中路由器 A 运行 OSPF，路由器 C 运行 RIPv1。为了使所有子网相互连通，请给出路由器 B 的配置。
- 3. 在图 11-37 中路由器 A 运行 EIGRP，路由器 C 运行 IS-IS。在路由器 B，所有 IS-IS 的路由级别均为 1。请在路由器 B 配置相互重新分配时尽可能使用汇总路由。EIGRP 路由要求被作为外部路由被通告到 IS-IS 域。

11.7 故障诊断练习

1. 在“案例研究：重新分配 IGRP 和 RIP”中，下面给出图 11-18 中路由器 Mantle 的配置：

```
router rip
 redistribute igrp 1 metric 5
 passive-interface Ethernet1
 network 10.0.0.0
!
router igrp 1
 redistribute connected
 redistribute rip
 default-metric 1000 100 255 1 1500
 passive-interface Ethernet0
 network 10.0.0.0
```

由于路由可以被重新分配到 IGRP，接着再次被重新分配到 RIP，那么 RIP 域可以知道末梢网络 192.168.10.0/24 吗？

2. 在练习 1 中，如果在 IGRP 的配置中取消命令 **redistribute rip**，那么为了向 RIP 域进行通告，命令 **redistribute connected** 是否可以满足要求？

3. 在图 11-20 中，为什么子网 192.168.3.32/27 没有被标记为 EIGRP 外部路由？

4. 在图 11-20 中，有一条指向 192.168.3.0 的汇总路由，是什么导致产生了这一条目？

5. 在图 11-27 中，为什么 192.168.1.0/24 不在 Campanella 的路由选择表中？

第 12 章

缺省路由和按需 路由选择

本章包括以下主题：

- 缺省路由基本原理
 - 按需路由基本原理
 - 配置缺省路由和 ODR
- 案例研究：静态缺省路由
案例研究：缺省网络命令
案例研究：缺省信息发生命令
案例研究：配置按需路由

到目前为止，我们已经在几个章节中对路由汇总进行了讨论。汇总可以减小路由选择表的尺寸，减少路由通告内容，从而节省了互联网络的资源。路由选择表越小、越简单，那么管理和故障诊断也越容易。

一个汇总地址可以表示几个或更多个更加精确的地址。例如，下面的 4 个子网可以用单一 192.168.200.128/25 来汇总。

```
192.168.200.128/27
192.168.200.160/27
192.168.200.192/27
192.168.200.224/27
```

当使用二进制方式查看地址时，我们可以看出汇总地址不太准确，因为汇总地址所包含的网络和子网位要比原地址少。因此，如果用粗略的方式表达，可以说向主机空间添加越多的 0 位，可用的网络位就越少，那么可以汇总的地址就越多。如果许多 0 位被添加到主机空间以至于没有网络位剩余将会怎么样？如果汇总地址包括 32 个 0 位又会怎样呢？这个地址将会汇总所有可能的 IP 地址。

0.0.0.0 就是这个 IP 缺省地址, 指向 0.0.0.0 的路由就是缺省路由¹。其他每个 IP 地址都比这个地址更准确, 所以当路由选择表中存在缺省路由时, 如果不能寻找到一个更加准确的路由, 那么都会匹配到该路由上。

12.1 缺省路由基本原理

当把路由器连入 Internet 时, 缺省路由是非常有用的。没有缺省路由, 路由器将不得不为 Internet 上每一个可达网络记录一条路由, 按照这样的记录方式, 路由选择表将会包括 55 000 多个路由条目。而使用了缺省路由, 路由器仅需要知道它自己内部管理系统中的目标网络。缺省路由将把去往其他地址的报文转发给 Internet 服务提供商。在处理大型路由选择表时, 拓扑变化所产生的影响远远大于对内存的需求。在大型互联网络中, 拓扑的变化频繁发生, 从而导致通告以及处理这些变化的系统活动明显增加。使用缺省路由可以有效地隐藏更精确路由的变化, 使得具有缺省路由的互联网络更加稳定。

在单一自主系统中, 缺省路由在更小的程度上还是有用的。在小型互联网络中, 缺省路由可以减少对内存和 CPU 的使用, 尽管随着路由数目的减少这种好处也会相应减弱。

缺省路由在中心辐条式拓扑结构中也非常有用, 如图 12-1。在这里, 中心路由器包含指向每一个远程子网的静态路由。每当一个新的子网在线, 中心路由器上就会被输入一条新的静态路由。虽然这一管理任务是微不足道, 但是要在每个辐条路由器上添加路由可能会很耗时。如果在辐条路由器上使用缺省路由, 那么仅中心路由器需要有关每个子网的路由。当辐条路由器收到去往未知目标网络的报文时, 它将把报文转发至中心路由器, 中心路由器接着将报文发送到正确的目的地。

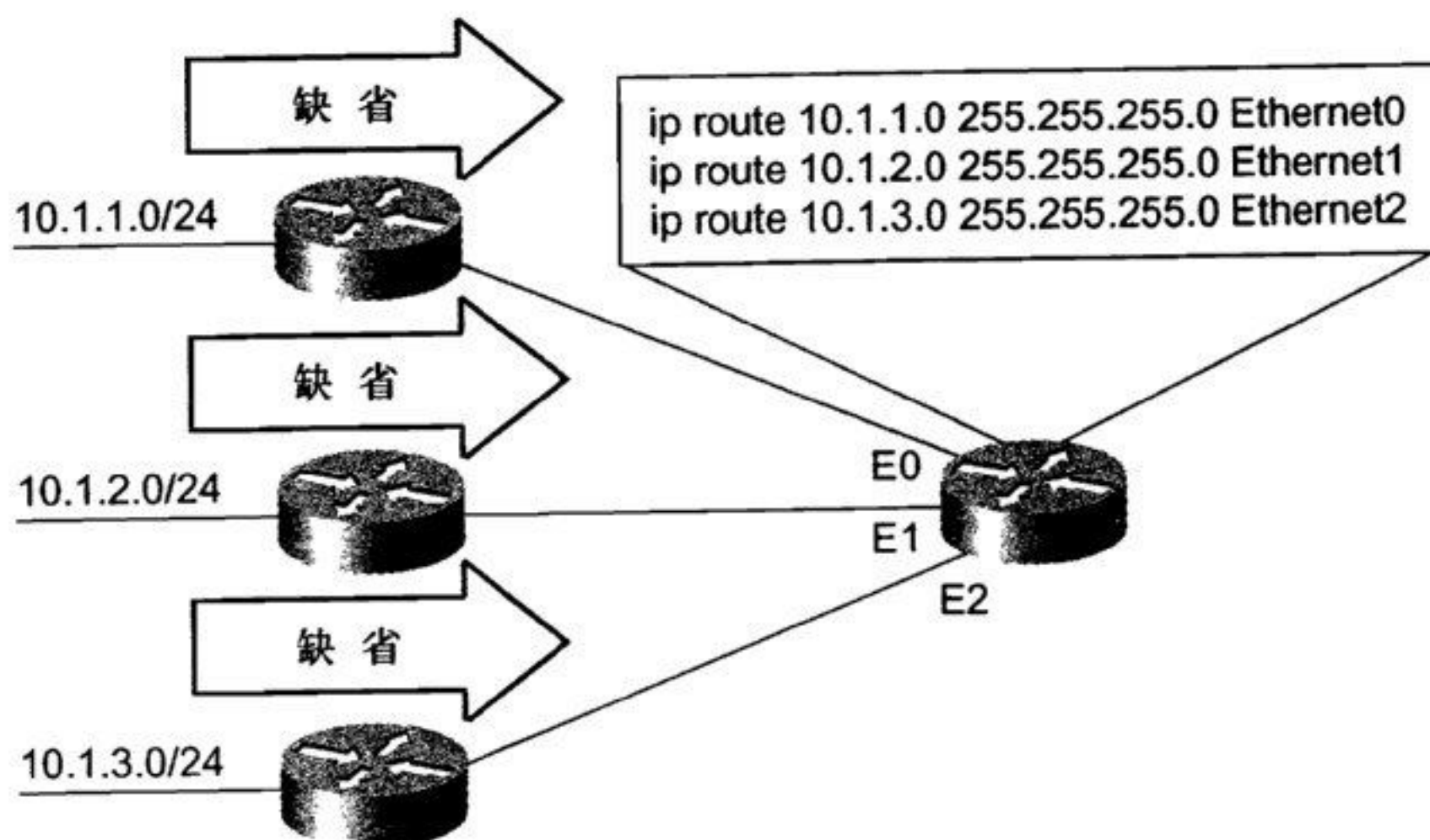


图 12-1 缺省路由大大地简化了中心辐条式互联网络的静态路由管理

在图 12-1 中, 辐条路由器更正确地应该称为末梢路由器。末梢路由器到其他路由器仅存在一条连接。在这种设备上路由的决策就变得非常简单: 目标网络要么是路由器各直连网络

¹ 在所有开放式 IP 路由选择协议中均使用这个地址。Cisco 的 IGRP 和 EIGRP 使用一个真实的网络地址, 作为外部路由进行通告。

(末梢网络)之一, 要么经邻居路由器可达。而且如果这个邻居路由器是下一跳路由器的惟一选择, 那么末梢路由器将不需要详细的路由选择表, 一条缺省路由就足够了。

正如使用汇总路由一样, 缺省路由也会造成路由细节的损失。例如在图 12-1 中的末梢路由器将不知道某个目标网络是否可达。所有去往未知目标网络的报文都要被转发到中心路由器, 然后确定网络是否可达。在互联网中, 很少会发生向不存在的地址发送报文的情况。但如果万一发生, 那么让末梢路由器指定全部路由选择表以便尽快地确定未知网络可能是一个更好的设计选择。

图 12-2 给出了路由细节损失所引起的另一个问题。这些路由器形成了一个全国性的骨干网络, 而且把大型本地互联网络也连接到这些骨干路由器上。洛杉矶 (Los Angeles) 骨干路由器正在接收来自旧金山 (San Francisco) 和圣地亚哥 (San Diego) 的缺省路由。如果洛杉矶必须要向西雅图转发报文, 而它仅有两条缺省路由, 那么它无法得知经过旧金山才是最佳的路由。洛杉矶有可能会向圣地亚哥转发报文, 在这种情况下, 报文会被延误到达目标网络, 而且在报文到达目的地之前, 报文还将使用部分非常昂贵的链路带宽。可见在骨干网中使用缺省路由是一个不好的设计决策,¹除此之外还说明了使用缺省路由隐藏路由细节会导致不理想的路由选择。

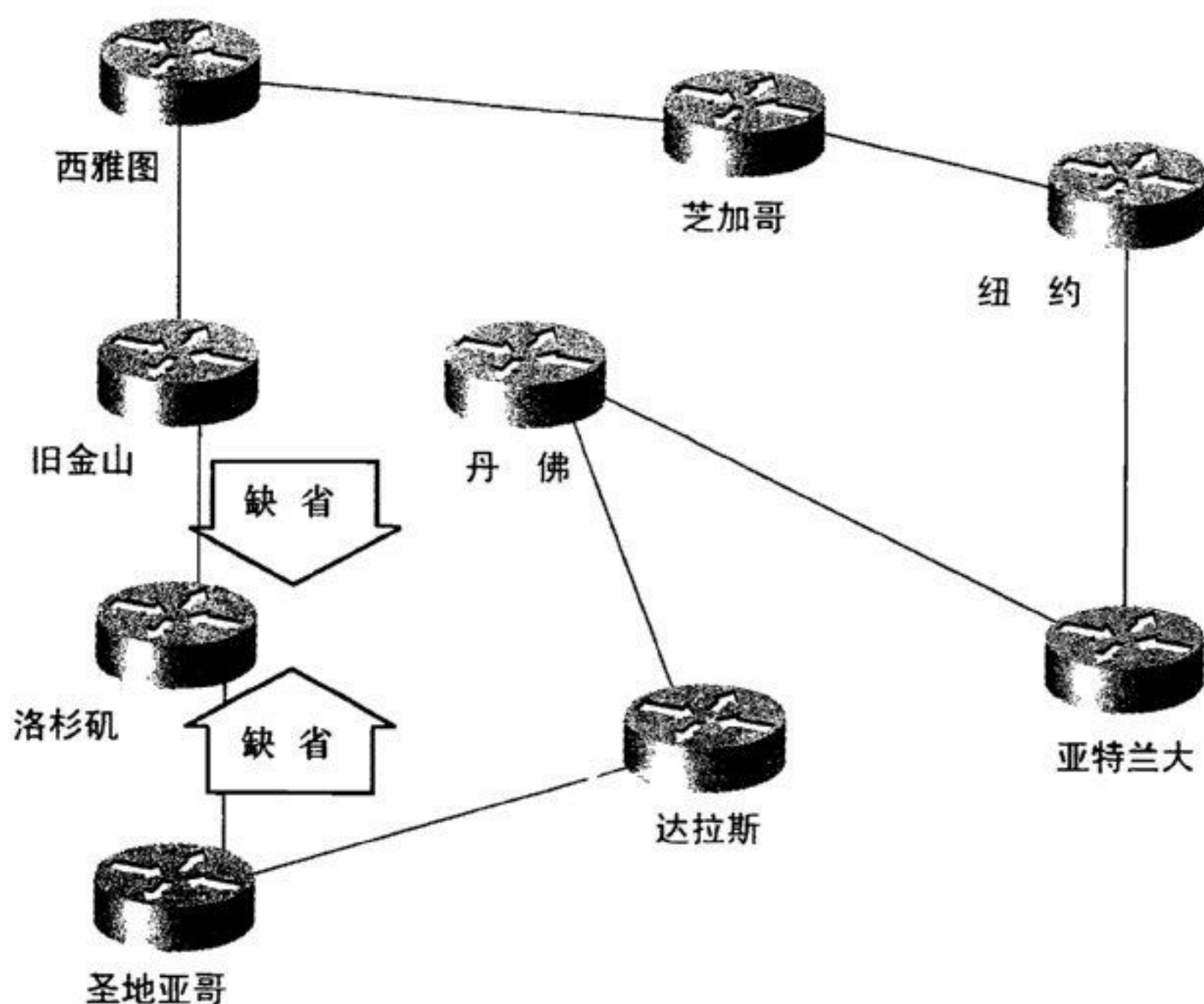


图 12-2 如果洛杉矶仅知道旧金山和圣地亚哥通告的缺省路由, 而不知道这两个路由器后面拓扑结果的更多细节, 那么它将不能够有效地进行路由选择

12.2 按需路由基本原理

如图 12-1, 虽然在中心路由器上配置静态路由非常简单, 但是许多网络管理员仍然不喜

¹ 另一方面, 让每个骨干路由器仅向它的本地互联网络通告一条缺省路由, 这将会是一个非常好的设计选择。

欢使用静态路由。困难不在于每当新的末梢网络在线时需要添加路由，而是在末梢网络或末梢路由器离线时忘记删除路由。从 IOS 11.2 起，Cisco 开始向中心路由器提供另一个专有技术，叫做按需路由（On-Demand Routing, ODR）。

当末梢路由器仍然使用指向中心路由器的缺省路由时，中心路由器使用 ODR 可以自动地发现末梢网络。ODR 传送地址前缀，即地址的网络号，而不是整个地址，因此路由器必须支持 VLSM。由于仅有非常少的路由信息在末梢路由器和中心路由器之间的链路上传输，所以节省了带宽。

ODR 不是真正意义上的路由选择协议。它可以发现有关末梢网络的信息，但是 ODR 不能向末梢路由器提供任何到末梢路由器的路由选择信息。数据链路协议完成了链路信息的传输，因而没有从末梢路由器到中心路由器做进一步的传输。然而，正如后面案例研究将显示的，ODR 发现的路由可以被重新分配到动态路由选择协议中。

图 12-3 给出了一个包含 ODR 条目的路由选择表，该表显示的管理距离为 160，路由的度量为 1。因为 ODR 路由总是从中心路由器到末梢路由器，所以度量（跳数）将永远不会超过 1。路由还显示出支持 VLSM。

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

Gateway of last resort is not set

    192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
o      192.168.1.40/30 [160/1] via 192.168.1.37, 00:00:27, Serial0
C      192.168.1.36/30 is directly connected, Serial0
C      192.168.1.192/27 is directly connected, Ethernet1
o      192.168.3.0/24 [160/1] via 192.168.1.37, 00:00:27, Serial0
    192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
o      192.168.4.48/29 [160/1] via 192.168.1.37, 00:00:27, Serial0
o      192.168.4.128/27 [160/1] via 192.168.1.37, 00:00:27, Serial0
Router#
```

图 12-3 这个路由选择表显示了几个 ODR 条目

ODR 路由的传输机制是 Cisco 发现协议（Cisco Discovery Protocol, CDP），CDP 是一种专用的数据链路协议，它可以收集有关邻居网络设备¹的信息。图 12-4 给出了 CDP 所收集信息的类型。

CDP 运行在任何支持子网访问协议（SNAP）的介质上，这意味着 ODR 还依赖于 SNAP 的支持。虽然在所有运行 IOS 10.3 或更高版本 IOS 的 Cisco 设备的所有接口上，CDP 缺省是被启用的，但是从 IOS 11.2 才开始支持 ODR。配置案例研究将会显示 ODR 仅能配置在中心路由器上，为了中心路由器能发现末梢路由器所连接的网络，末梢路由器必须运行 IOS 11.2 或更高版本的操作系统。

¹ CDP 不仅可以运行在路由器上，而且还可以运行在 Cisco 的交换机和接入服务器上。


```
reg75k2#show cdp neighbor detail
-----
Device ID: WPI72k
Entry address(es):
  IP address: 192.168.5.2
  Novell address: BA5.0008.f417.1f88
Platform: cisco 7206, Capabilities: Router Source-Route-Bridge
Interface: TokenRing5/1/0, Port ID (outgoing port): TokenRing2/1
Holdtime : 133 sec
Version :
Cisco Internetwork Operating System Software
IOS (tm) 7200 Software (C7200-DR-M), Version 11.1(14)CA1, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)
Synced to mainline version: 11.1(14)
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Tue 30-Sep-97 16:49 by susingh
-----
Device ID: REG75K1
Entry address(es):
  IP address: 172.23.109.2
  Novell address: AA08.0006.e2b4.8c46
  DECnet address: 16.2
Platform: cisco RSP4, Capabilities: Router Source-Route-Bridge
Interface: TokenRing5/1/3, Port ID (outgoing port): TokenRing3/2
Holdtime : 134 sec
Version :
Cisco Internetwork Operating System Software
IOS (tm) GS Software (RSP-JV-M), Version 11.1(16)CA, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)
Synced to mainline version: 11.1(16)
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Sat 20-Dec-97 04:21 by tej
-----
```

图 12-4 CDP 收集有关邻居 Cisco 网络设备的信息

12.3 配置缺省路由和 ODR

缺省路由可以配置在每个需要缺省路由的路由器上，或者配置在向其对等路由器依次通告路由的路由器上。本节的案例研究将分析这两种方法。

回忆一下第 5 章讨论的有类别路由查询，路由器将首先匹配主网地址，然后匹配子网。如果子网不匹配，那么报文将被丢弃。有类别路由查询是 Cisco 路由器的缺省模式，但查询也可以通过命令 **ip classless** 变换到无类别（甚至对有类别路由选择协议）查询方式。

任何使用缺省路由的路由器必须执行无类别路由查询，图 12-5 解释了这样做的原因。在这个互联网络中，Memphis 同 Tanis、Giza 使用动态路由选择协议，但是 Memphis 不从 Thebes 接收路由。为了向 BigNet 路由报文，Memphis 有一条指向 Thebes 的缺省路由。如果 Memphis 接收到目的地址是 192.168.1.50 的报文，并且它正在执行有类别路由查询，那么它将会首先匹配主网地址 192.168.1.0，在路由选择表中存在该主网地址的几个子网。接着 Memphis 试图寻找有关子网 192.168.1.48/28 的路由，但是因为 Memphis 不从 Thebes 接收路由，所以该子

网不在路由选择表中, 因此报文会被丢弃。

如果 Memphis 配置了 **ip classless**, 那么它首先不会匹配主网络, 而是为 192.168.1.48/28 寻找最精确的匹配。如果在路由选择表中没有发现, 那么它将匹配到缺省路由, 最终报文被转发到 Thebes。

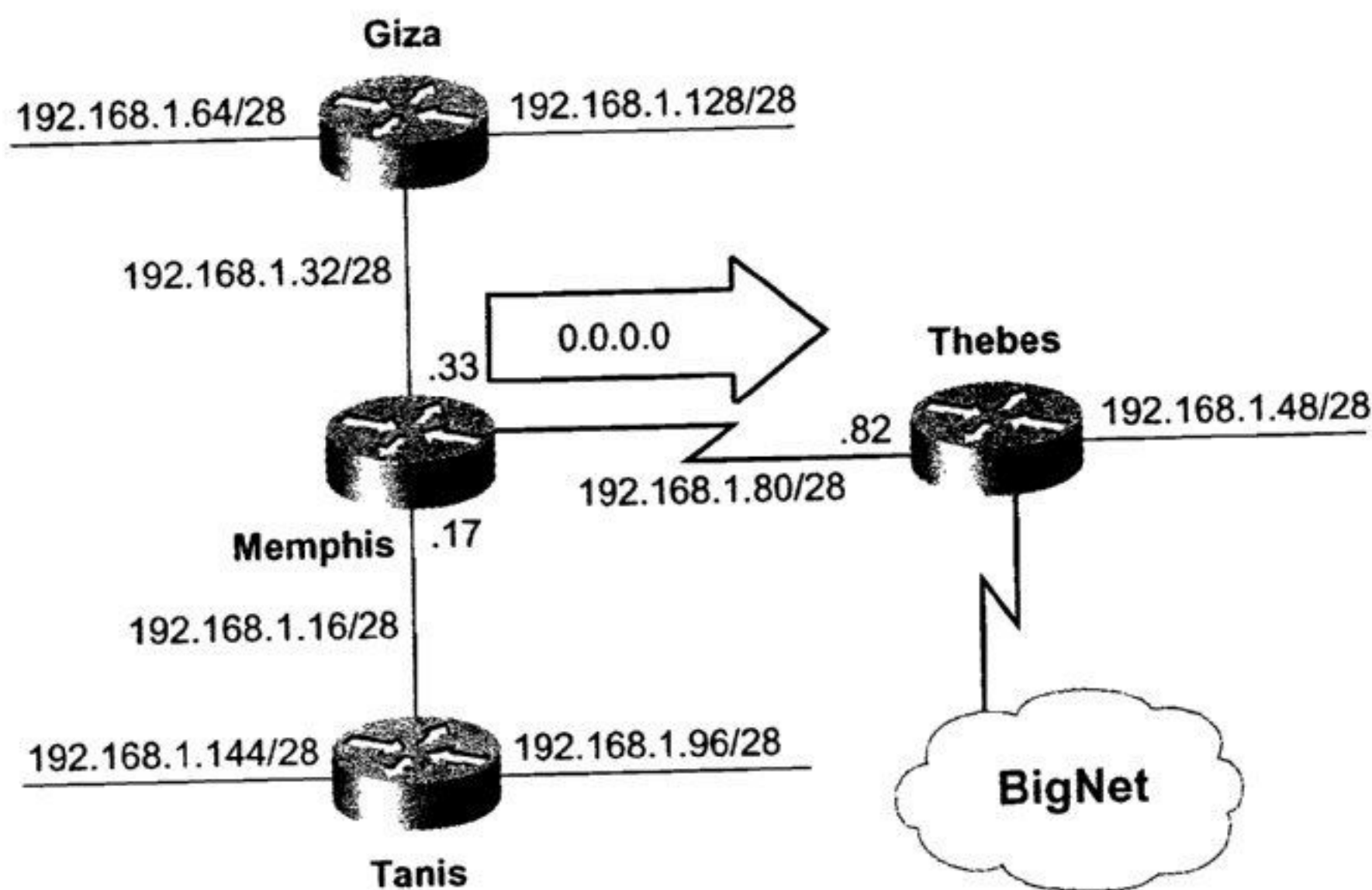


图 12-5 Memphis 使用缺省路由向 Thebes 转发报文。如果 Memphis 使用有类别路由查询, 子网 192.168.1.48/28 将不可达

12.3.1 案例研究: 静态缺省路由

图 12-5 中 Memphis 的配置如下:

```
router rip
 network 192.168.1.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.82
```

静态路由配置了缺省路由地址 0.0.0.0, 并且使用的掩码也是 0.0.0.0。第一次配置缺省路由的人们常犯的一个错误是使用全 1 掩码, 而不是使用全 0 掩码, 例如:

```
ip route 0.0.0.0 255.255.255.255 192.168.1.82
```

全 1 掩码将会设置一条指向 0.0.0.0 的主机路由, 惟有那些目的地址为 0.0.0.0 的报文才能匹配到该地址。另一方面, 全 0 掩码全部是由“不关心”位组成, 它可以在任意位置匹配到任意位。本章开头曾讨论过缺省地址是汇总地址的一种极端形式, 所以每一个位都会被用 0 汇总。这里缺省路由的掩码也是汇总掩码的一种极端形式。

Memphis 缺省路由的下一跳地址在 Thebes 上。这个下一跳地址指的是最后可选网关或缺省路由器。图 12-6 给出了 Memphis 的路由选择表。指向 0.0.0.0 的路由被标记为候选缺省, 在表的开头指明了最后可选网关。


```

Memphis#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 192.168.1.82 to network 0.0.0.0

    192.168.1.0/28 is subnetted, 7 subnets
R      192.168.1.96 [120/1] via 192.168.1.18, 00:00:15, Ethernet0
R      192.168.1.64 [120/1] via 192.168.1.34, 00:00:27, Ethernet1
C      192.168.1.80 is directly connected, Serial0
C      192.168.1.32 is directly connected, Ethernet1
C      192.168.1.16 is directly connected, Ethernet0
R      192.168.1.128 [120/1] via 192.168.1.34, 00:00:27, Ethernet1
R      192.168.1.144 [120/1] via 192.168.1.18, 00:00:15, Ethernet0
S* 0.0.0.0/0 [1/0] via 192.168.1.82
Memphis#

```

图 12-6 Memphis 的路由选择表，给出了缺省路由和最后可选网关

Memphis 将会向 Tanis 和 Gize 通告缺省路由（图 12-7）。首先，这个动作看上去有些令人惊讶，因为在 Memphis 没有配置重新分配。然而，静态路由并不是实际被重新分配的路由。在路由选择表中一旦缺省路由被标识，RIP、IGRP 和 EIGRP 将会自动通告它。但 OSPF 和 IS-IS 还需要一些额外的配置，这将在后面的案例研究中看到。

```

Tanis#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 192.168.1.17 to network 0.0.0.0

    192.168.1.0/28 is subnetted, 9 subnets
C      192.168.1.96 is directly connected, Ethernet1
R      192.168.1.64 [120/2] via 192.168.1.17, 00:00:01, Ethernet0
R      192.168.1.80 [120/1] via 192.168.1.17, 00:00:01, Ethernet0
R      192.168.1.32 [120/1] via 192.168.1.17, 00:00:01, Ethernet0
R      192.168.1.48 [120/2] via 192.168.1.17, 00:00:01, Ethernet0
C      192.168.1.16 is directly connected, Ethernet0
R      192.168.1.224 [120/1] via 192.168.1.17, 00:00:01, Ethernet0
R      192.168.1.128 [120/2] via 192.168.1.17, 00:00:01, Ethernet0
C      192.168.1.144 is directly connected, Ethernet2
R* 0.0.0.0/0 [120/1] via 192.168.1.17, 00:00:02, Ethernet0
Tanis#

```

图 12-7 Tanis 的路由选择表显示了缺省路由是通过 RIP 协议从 Memphis 那里学习到的

缺省路由对于连接无类别路由选择域也是非常有用的，在图 12-8 中，Chimu 将一个 RIP 域和一个 EIGRP 域连接到一起。虽然在 RIP 域主网 192.168.25.0 的掩码是一致的，但

是在 EIGRP 域对该主网进行了变长子网划分。此外, VLSM 方式对进入 RIP 的汇总没有任何帮助。

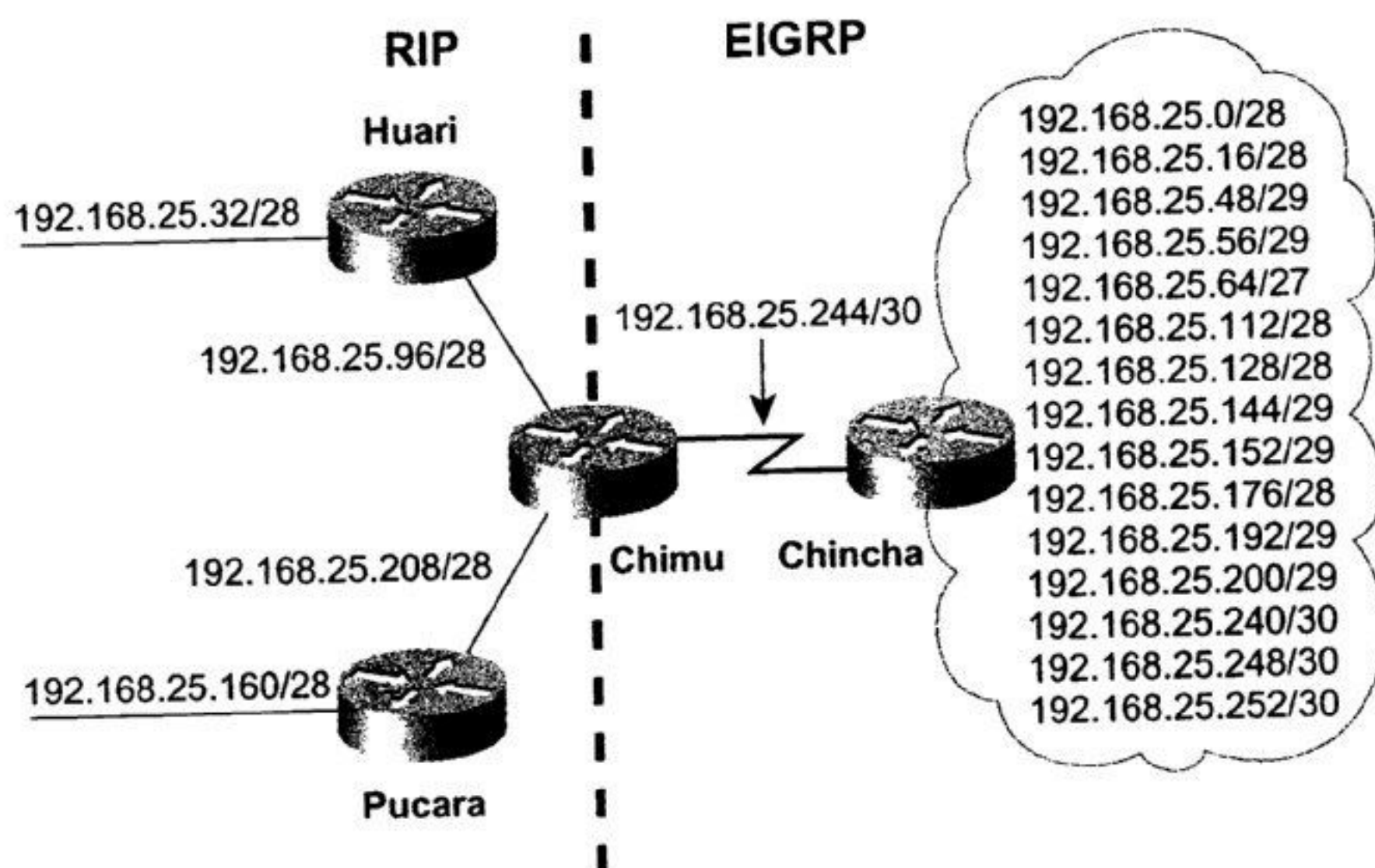


图 12-8 缺省路由使 RIP 可以路由进入变长子网化的 EIGRP 域

Chimu 的配置如下:

```
router eigrp 1
 redistribute rip metric 1000 100 255 1 1500
 passive-interface Ethernet0
 passive-interface Ethernet1
 network 192.168.25.0

!
router rip
 passive-interface Serial0
 network 192.168.25.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 Null0
```

Chimu 有一套来自 EIGRP 域的完整路由,但是 Chimu 没有将它们重新分配进入 RIP。相反, Chimu 仅通告了一条缺省路由。RIP 路由器将向 Chimu 转发所有去往未知网络的报文,然后 Chimu 查找路由选择表,寻找一条进入 EIGRP 域的最精确的路由。

Chimu 的静态路由不是指向下一跳地址,而是指向空接口。如果转发给 Chimu 的报文的目标属于一个不存在的子网,例如 192.168.25.224/28,那么报文不会被转发到 EIGRP 域,而是被丢弃。

12.3.2 案例研究: 缺省网络命令

配置缺省路由的另一种方法是使用命令 **ip default-network**。该命令指明了用作缺省网络的主网地址。静态路由指明的这个网络可以是直连网络,或者是通过动态路由选择协议发现

的网络。

图 12-9 中 Athens 的配置如下：

```
router rip
 network 172.16.0.0
!
ip classless
default-network 10.0.0.0
```

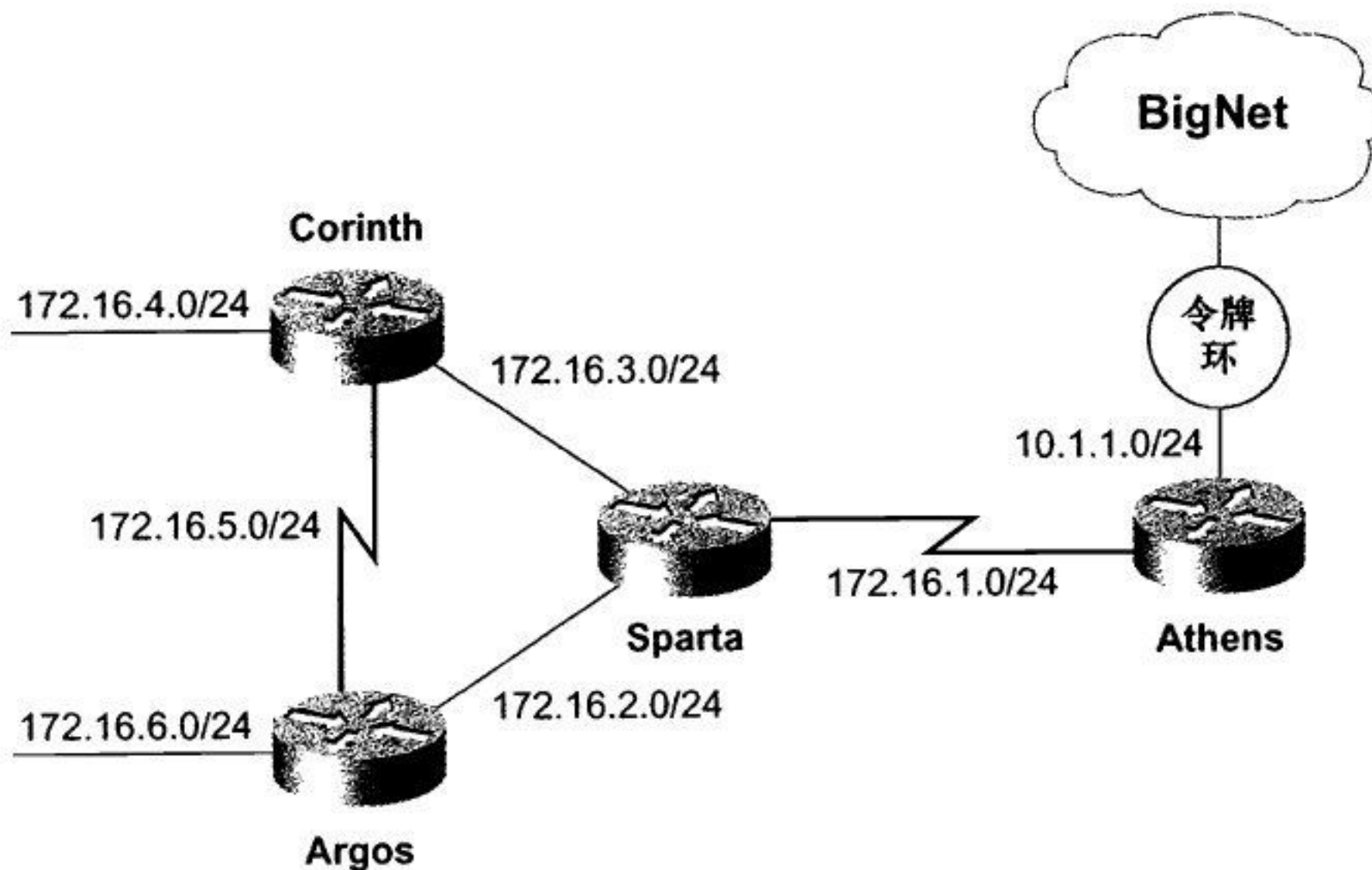


图 12-9 在 Athens 上使用命令 **default-network** 可以产生一个缺省网络通告

如图 12-10 所示，在 Athens 的路由选择表中网络 10.0.0.0 被标记为候选缺省路由，但是注意，没有指定最后可选网关，原因是 Athens 是到缺省网络的网关。即使在 RIP 配置中不存在有关网络 10.0.0.0 的表述（图 12-11），**ip default-network** 也将导致 Athens 通告一个缺省网络。

```
Athens#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
       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

Gateway of last resort is not set

*    10.0.0.0/8 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, TokenRing0
R    192.168.1.0/24 [120/2] via 172.16.1.2, 00:00:12, Serial0
    172.16.0.0/16 is subnetted, 6 subnets
R      172.16.4.0 [120/2] via 172.16.1.2, 00:00:12, Serial0
R      172.16.5.0 [120/2] via 172.16.1.2, 00:00:12, Serial0
R      172.16.6.0 [120/2] via 172.16.1.2, 00:00:12, Serial0
C      172.16.1.0 is directly connected, Serial0
R      172.16.2.0 [120/1] via 172.16.1.2, 00:00:12, Serial0
R      172.16.3.0 [120/1] via 172.16.1.2, 00:00:12, Serial0
Athens#
```

图 12-10 在 Athens 的路由选择表中网络 10.0.0.0 被标记作为候选缺省路由

对于 IGRP 和 EIGRP 来说缺省路由稍微有一些不同。这些协议不能理解地址 0.0.0.0。它们更适合通告一个真实地址作为外部路由（见第 6 章“内部网关路由选择协议（IGRP）”和第 8 章“增强型内部网关路由选择协议（EIGRP）”）。作为外部路由通告进入 IGRP 和 EIGRP 的目标网络被理解为缺省路由。

```
Sparta#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 172.16.1.1 to network 0.0.0.0

R    192.168.1.0/24 [120/1] via 172.16.2.2, 00:00:10, Ethernet0
      [120/1] via 172.16.3.2, 00:00:14, Ethernet1
      172.16.0.0/24 is subnetted, 6 subnets
R    172.16.4.0 [120/1] via 172.16.3.2, 00:00:14, Ethernet1
R    172.16.5.0 [120/1] via 172.16.3.2, 00:00:14, Ethernet1
R    172.16.6.0 [120/1] via 172.16.2.2, 00:00:10, Ethernet0
C    172.16.1.0 is directly connected, Serial0
C    172.16.2.0 is directly connected, Ethernet0
C    172.16.3.0 is directly connected, Ethernet1
R*   0.0.0.0/0 [120/1] via 172.16.1.1, 00:00:17, Serial0
Sparta#
```

图 12-11 Sparta 的路由选择表显示出 Athens 正在通告一条缺省路由 0.0.0.0，而且 Athens 是 Sparta 的最后可选网关

如果设置图 12-9 中的路由器运行 IGRP，那么 Athen 的配置将变为：

```
router igrp 1
 network 10.0.0.0
 network 172.16.0.0
!
ip classless
ip default-network 10.0.0.0
```

命令 **ip default-network** 保持不变，但是注意，在 IGRP 的配置中添加了关于网络 10.0.0.0 的表述。因为 IGRP 使用真实网络地址，所以必须配置该地址进行通告，见图 12-12。因为 Corinth 从 Aparta 那里学习到缺省路由，所以 Sparta 是 Corinth 的最后可选网关。如果到 Sparta 的链路发生故障，那么 Corinth 将使用 Argos 作为最后可选网关。

```
Corinth#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
       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

Gateway of last resort is 172.16.3.1 to network 10.0.0.0

I* 10.0.0.0/8 [100/8639] via 172.16.3.1, 00:00:17, Ethernet0
```

待续


```

172.16.0.0/16 is subnetted, 6 subnets
C    172.16.4.0 is directly connected, Ethernet1
C    172.16.5.0 is directly connected, Serial0
I    172.16.6.0 [100/1700] via 172.16.3.1, 00:00:18, Ethernet0
I    172.16.6.0 [100/1700] via 172.16.3.1, 00:00:18, Ethernet0
I    172.16.2.0 [100/1200] via 172.16.3.1, 00:00:18, Ethernet0
C    172.16.3.0 is directly connected, Ethernet0

```

图 12-12 IGRP 和 EIGRP 使用真实网络地址，而不是 0.0.0.0，作为缺省网络。

Corinth 的路由选择表显示出网络 10.0.0.0 被标记为缺省网络

12.3.3 案例研究：缺省信息发生命令

OSPF 的 ASBR 和 IS-IS 域间路由器不能向它们的路由选择域自动通告缺省路由，即使在缺省路由存在的时候也一样。例如，假设图 12-9 中的 Athens 配置了 OSPF，并且设置了一条指向 BigNet 的缺省路由：

```

router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.2

```

图 12-13 给出了 Athens 和 Sparta 的路由选择表。虽然静态路由使得在 Athen 上设置了最后可选网关，但是 Aparta 却不知道缺省路由。缺省路由必须以类型 5 的 LSA 方式被通告进入 OSPF 域，这意味着 Athens 必须是一个 ASBR。然而到目前为止，Athens 的配置并没有告诉它执行这些功能。

default-information originate 是重新分配命令的一个特例，它将导致一条缺省路由被重新分配进入 OSPF 和 IS-IS。同 **redistribute** 一样，命令 **default-information originate** 通知 OSPF 路由器它成为一个 ASBR 或通知 IS-IS 路由器它成为一个域间路由器。而且还指明被重新分配的缺省路由的度量，可以是 OSPF 外部度量类型，或者是 IS-IS 级别。为了向 OSPF 重新分配缺省路由，并且要求缺省路由的度量值为 10，外部度量类型为 E1，Athens 的配置将为：

```

router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 default-information originate metric 10 metric-type 1
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.2

```

图 12-14 给出的缺省路由正在被重新分配进入 OSPF。而且还可以在 Sparta 的 OSPF 数据库中观察到这个路由。

命令 **default-information originate** 还可以向 OSPF 和 IS-IS 重新分配被其他路由进程发现的缺省路由。在下面的配置中，指向网络 0.0.0.0 的静态路由被删除，而且 Athens 与 BigNet 中的一个路由器使用 BGP 通信：


```
Athens#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
       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
```

Gateway of last resort is 10.1.1.2 to network 0.0.0.0

```
10.0.0.0 255.255.255.0 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, TokenRing0
172.16.0.0 is variably subnetted, 6 subnets, 2 masks
O      172.16.5.0 255.255.255.0 [110/138] via 172.16.1.2, 00:04:17, Serial0
O      172.16.4.1 255.255.255.0 [110/75] via 172.16.1.2, 00:04:17, Serial0
O      172.16.6.1 255.255.255.0 [110/75] via 172.16.1.2, 00:04:17, Serial0
C      172.16.1.0 255.255.255.0 is directly connected, Serial0
O      172.16.2.0 255.255.255.0 [110/74] via 172.16.1.2, 00:04:17, Serial0
O      172.16.3.0 255.255.255.0 [110/74] via 172.16.1.2, 00:04:17, Serial0
S* 0.0.0.0 0.0.0.0 [1/0] via 10.1.1.2
```

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

Gateway of last resort is not set

```
172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
O      172.16.5.0/24 [110/74] via 172.16.2.2, 00:06:00, Ethernet1
       [110/74] via 172.16.3.2, 00:06:00, Ethernet0
O      172.16.4.1/24 [110/11] via 172.16.3.2, 00:06:00, Ethernet0
O      172.16.6.1/24 [110/11] via 172.16.2.2, 00:06:00, Ethernet1
C      172.16.1.0/24 is directly connected, Serial0
C      172.16.2.0/24 is directly connected, Ethernet1
C      172.16.3.0/24 is directly connected, Ethernet0
```

图 12-13 在 Athens 的 OSPF 进程不能向 OSPF 域自动通告缺省路由

```
Sparta#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 172.16.1.1 to network 0.0.0.0

```
172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
O      172.16.5.0/24 [110/74] via 172.16.2.2, 00:14:46, Ethernet0
O      172.16.4.1/32 [110/75] via 172.16.2.2, 00:14:46, Ethernet0
O      172.16.6.1/32 [110/11] via 172.16.2.2, 00:14:46, Ethernet0
C      172.16.1.0/24 is directly connected, Serial0
C      172.16.2.0/24 is directly connected, Ethernet0
C      172.16.3.0/24 is directly connected, Ethernet1
O* E1 0.0.0.0/0 [110/74] via 172.16.1.1, 00:02:55, Serial0
Sparta#
```

图 12-14 在 Athens 配置 **default-information originate** 之后, 缺省路由将被重新分配进入 OSPF 域


```

router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
  default-information originate metric 10 metric-type 1
!
router bgp 65501
  network 172.16.0.0
  neighbor 10.1.1.2 remote-as 65502
!
ip classless

```

现在 Athens 从它的 BGP 邻居路由器那里学习到指向 0.0.0.0 的路由，并且使用类型 5 的 LSA 向 OSPF 通告该路由（图 12-16）。

缺省路由或汇总路由的好处是提供互联网络的稳定性，但是如果缺省路由自身不稳定会发生什么呢？例如在图 12-15 中，假设通告给 Athens 的缺省路由在波动，也就是频繁地在可达与不可达之间变换。伴随着每一个变化，Athens 都必须向 OSPF 域发送一条新的类型 5 的 LSA，这个 LSA 将会被通告到所有非末梢区域。虽然这种泛洪和再泛洪对系统资源影响不大，但是这不是网络管理所期望的。解决办法是使用关键字 **always**。¹

```

Sparta#show ip ospf database external

      OSPF Router with ID (172.16.3.1) (Process ID 1)

          Type-5 AS External Link States

Routing Bit Set on this LSA
LS age: 422
Options: (No TOS-capability, No DC)
LS Type: AS External Link
Link State ID: 0.0.0.0 (External Network Number )
Advertising Router: 172.16.1.1
LS Seq Number: 80000002
Checksum: 0x5238
Length: 36
Network Mask: /0
    Metric Type: 1 (Comparable directly to link state metric)
    TOS: 0
    Metric: 10
    Forward Address: 0.0.0.0
    External Route Tag: 1
Sparta#

```

图 12-15 像 ASBR 通告的其他外部路由一样，缺省路由以类型 5 的 LSA 方式被通告

```

router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
  default-information originate always metric 10 metric-type 1
!
router bgp 65501
  network 172.16.0.0
  neighbor 10.1.1.2 remote-as 65502
!
ip classless

```

¹ 这个关键字仅在 OSPF 下可用，在 IS-IS 下不支持。


```

Athens#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
       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

Gateway of last resort is 10.1.1.2 to network 0.0.0.0

    10.0.0.0/8 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, TokenRing0
    172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
O IA   172.16.4.1/32 [110/139] via 172.16.1.2, 00:16:45, Serial0
O IA   172.16.5.0/24 [110/138] via 172.16.1.2, 00:16:45, Serial0
O IA   172.16.6.1/32 [110/75] via 172.16.1.2, 00:16:45, Serial0
C      172.16.1.0/24 is directly connected, Serial0
O IA   172.16.2.0/24 [110/74] via 172.16.1.2, 00:16:45, Serial0
O IA   172.16.3.0/24 [110/74] via 172.16.1.2, 00:16:45, Serial0
B* 0.0.0.0/0 [20/0] via 10.1.1.2, 00:12:02
Athens#

```

图 12-16 在 BigNet 中使用 BGP 的路由器向 Athens 通告缺省路由

使用这种配置方法, Athens 将始终通告一条缺省路由进入 OSPF, 不管它实际上是否有一条指向 0.0.0.0 的路由。如果在 OSPF 域内的路由器向 Athens 转发了一个报文, 而 Athens 没有缺省路由, 那么它将向源点发送目标不可达的 ICMP 信息并且丢弃该报文。

当 OSPF 域外仅有单一的缺省路由时, 关键字 **always** 可以安全地被使用。如果不止一个 ASBR 通告了缺省路由, 那么缺省应该是动态的即缺省路由的丢失将会被通告。如果一个 ASBR 在它没有缺省路由时却声明有缺省路由, 那么报文将会被转发到它那里, 而不是被转发到合理的 ASBR。

12.3.4 案例研究: 配置按需路由

使用命令 **router odr** 可以启用 ODR, 不需要指明网络和其他参数。CDP 仅在因某种原因被关闭的情况下才需要被启用, 缺省情况下 CDP 是被启用的。在路由器上启用 CDP 进程的命令是 **cdp run**。为了在特定接口上启用 CDP, 要使用命令 **cdp enable**。

图 12-17 给出了一个典型的中心辐条式拓扑结构。为了配置 ODR, 中心路由器必须配置命令 **router odr**。只要所有路由器都运行 IOS 11.2 或更高版本, 并且连接介质支持 SNAP (例如给出的帧中继或 PVC), ODR 就可以运行, 而且中心路由器将会学习到末梢网络。在末梢路由器上惟一需要配置的是一条指向中心路由器的静态缺省路由。

ODR 还可以被重新分配。在图 12-17 中, 如果 Baghdad 需要向 OSPF 通告 ODR 发现的路由, Baghdad 的配置如下:

```

router odr
!
router ospf 1
 redistribute odr metric 100
 network 172.16.0.0 0.0.255.255 area 5

```

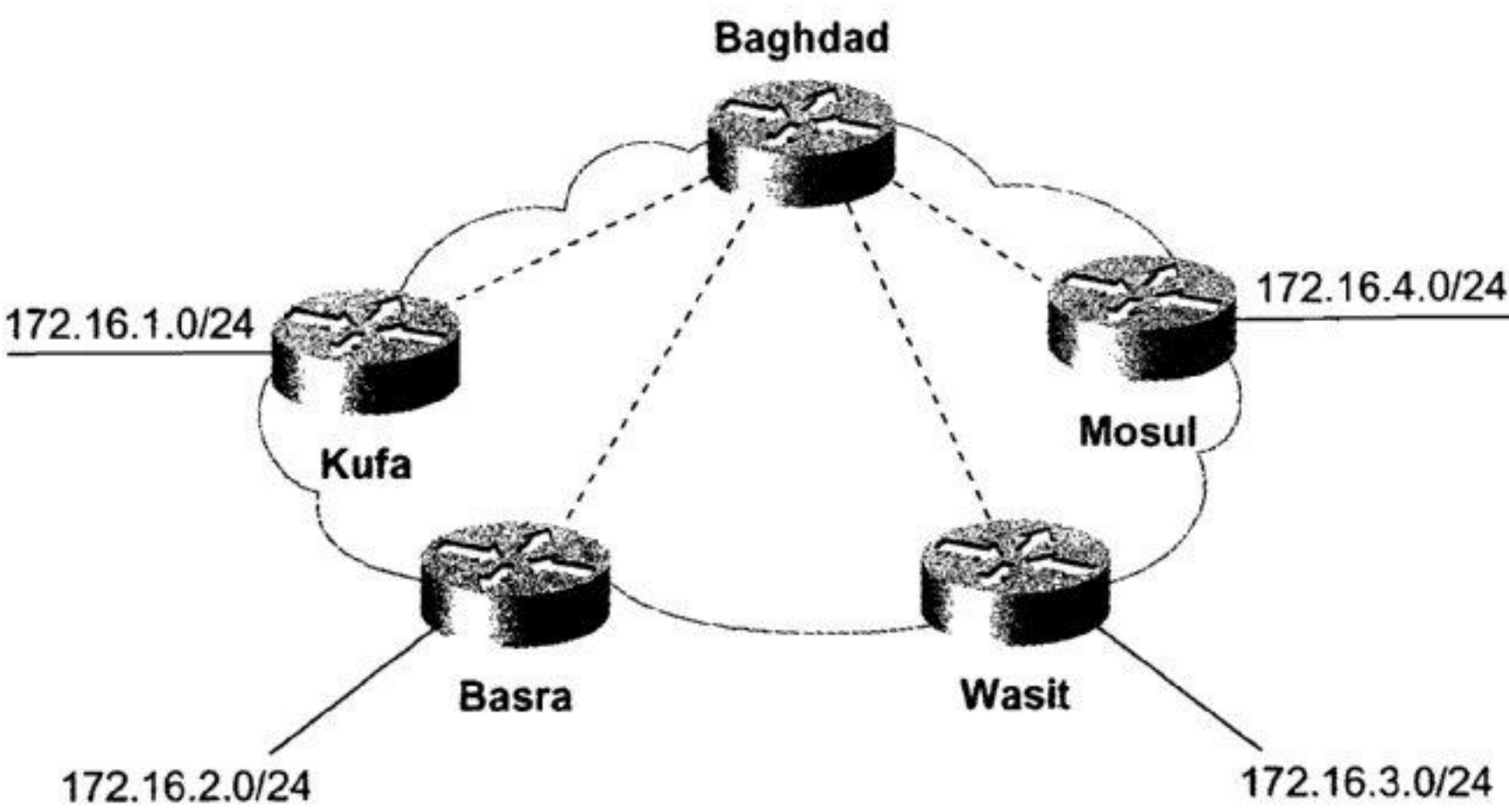



图 12-17 向这样的中心辐条式拓扑结构在帧中继网络中很普遍

12.4 展 望

在简单的没有环路的互联网络中配置和故障诊断缺省路由是一个微不足道的任务。当拓扑结构更加复杂时，特别是包含环路时，由缺省路由和重新分配所带来的潜在问题将会增加。第 13 章“路由过滤”和第 14 章“路由图”将讨论在复杂拓扑结构中控制路由行为的重要工具。

12.5 总结表：第 12 章命令回顾

命 令	描 述
<code>cdp enable</code>	在接口上启用 CDP
<code>cdp run</code>	在路由器上全局启用 CDP
<code>default-information originate [always][metric metric-value] [metric-type type-value]{level-1 level-1-2 level-2}[route-map map-name]</code>	向 OSPF 和 IS-IS 路由域选择输入缺省路由
<code>ip classless</code>	启用无类别路由查询，以便路由器可以向直连网络的未知子网转发报文
<code>ip default-network network-number</code>	在确定最后可选网关时指定一个网络作为候选路由
<code>ip route prefix mask {address interface}[distance][tag tag][permanent]</code>	指定缺省路由条目
<code>router odr</code>	启用按需路由选择

12.6 复 习 题

1. 开放协议使用的缺省路由的目标地址是什么？
2. IGRP 和 EIGRP 如何通告和标识缺省路由？

3. 在运行 IGRP 的路由器上可以使用指向 0.0.0.0 的静态路由作为缺省路由吗?
4. 什么是末梢路由器? 什么是末梢网络?
5. 使用缺省路由代替完整的路由选择表的好处是什么?
6. 使用完整的路由选择表代替缺省路由的好处是什么?
7. 按需路由使用什么样的数据链路协议发现路由?
8. 在使用 ODR 时, 对 IOS 有什么限制?
9. 在使用 ODR 时, 对介质有什么限制?

第 13 章

路由过滤

本章包括以下主题：

- 使用路由过滤器
- 配置路由过滤器

案例研究：过滤特定路由

案例研究：路由过滤和重新分配

案例研究：协议迁移

案例研究：多个重新分配点

案例研究：使用距离设置路由器的优先权

第 11 章“路由重新分配”曾介绍过在特殊路由器上，重新分配可能会在几种情况下导致不必要或不正确的路由。例如在图 11-3 及相关讨论中，一个或多个路由器选择了一条经过互联网络的非最佳路由。在这个例子中，问题出在路由器更加信任 IGRP，因为 IGRP 的管理距离比 RIP 要小。更加普遍的是，任何时间指向相同目标网络的路由都会被多个路由器重新分配进入路由选择域，其中可能会存在错误的路由。在某些情况下，可能会发生路由环路和黑洞。

图 11-26 给出了另一个关于不必要路由或意外路由的例子。在这个案例中，不仅向 OSPF 通告了汇总路由 192.168.3.128/25，而且还向 EIGRP 域重新分配了该路由，但是在 EIGRP 域内已经存在被汇总地址的子网。被通告的路由沿错误方向穿过重新分配路由器的现象叫做路由回馈。

路由过滤可以使网络管理员对路由通告施加严格的控制。任何时刻如果路由器执行相互重新分配——在两个或多个路由选择协议之间相互共享路由——为了确保沿着惟一的方向通告路由，那么应该使用路由过滤器。

图 13-1 给出了路由过滤器的另一种用途，在这里，一个路由选择域被分割为多个子域，每个子域包含多个路由器。

连接两个子域的路由器将对路由进行过滤以便子域 B 中的路由器仅知道子域 A 中的部分路由。这种过滤可能是处于安全考虑,以便 B 域中的路由器仅知道已被授权的子网。或者这样作只不过是减少不必要的路由来维持 B 域内路由器的路由选择表和更新信息的大小。

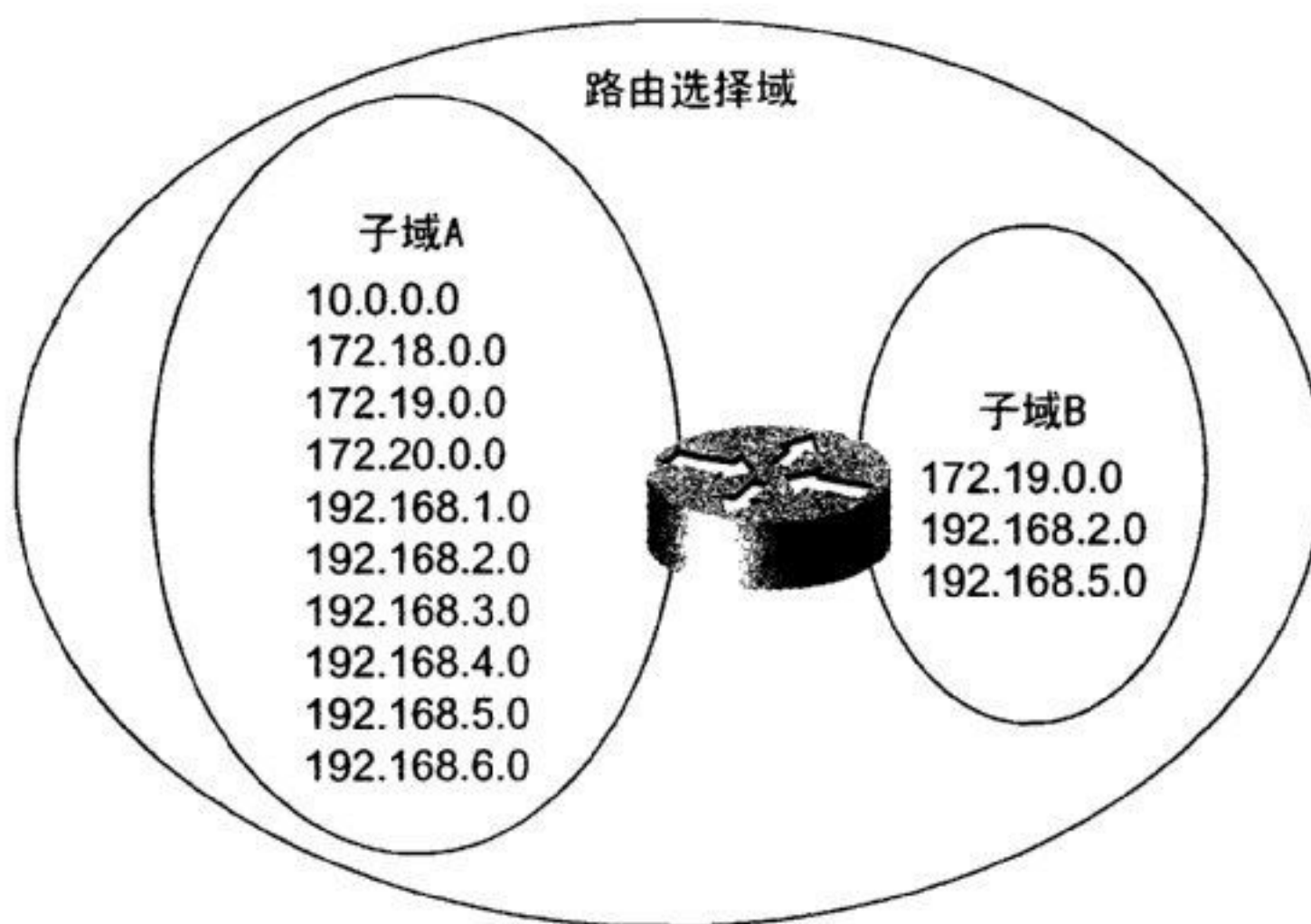


图 13-1 路由过滤器可以用于建立路由子域,以便仅向子域通告路由选择域内的部分地址

此外,路由过滤器的另一个常见用途是建立路由防火墙。公司部门或政府机关常常需要被互连在一起,然而它们却处于独立的管理控制之下。如果你不能控制这个互连网络所有部分,那么你很容易受到错误配置的影响甚至恶意路由的攻击。如果在互连网络中路由器上使用路由过滤,那么将确保路由器仅接收合法的路由。这种方法又是一种安全的方式,但是在这种情况下,管制的是出站路由,而不是入站路由。

外部的路由可以进入到路由表中,路由表中的路由也可以被通告出去,那么路由过滤器正是通过管制这些出入路由表的路由来工作的。路由过滤器对链路状态路由选择协议的影响和对距离矢量路由选择协议的影响稍微有点不同。运行距离矢量协议的路由器是基于自身路由表通告路由的,其结果是路由过滤选择将会对路由器通告给邻居路由器的路由有影响。

另一方面,运行链路状态协议的路由器是基于自身链路状态数据库的信息确定它们的路由,而不是基于被邻居路由器通告的路由条目。路由过滤器对链路状态的通告或链路状态数据库¹没有影响。所以路由过滤器会对配置了过滤的路由器的路由表产生影响,但不会对邻居路由器的路由条目有任何影响。正因为这种特性,路由过滤器主要被用在进入链路状态域的重新分配点上,例如 OSPF 的 ASBR,在那里路由过滤器可以控制那些进入或离开该域的路由。在链路状态域内,路由过滤器的效用是受限制的。

13.1 配置路由过滤器

使用下面所给出的任意一种方法都可以实现路由过滤:

¹ 请记住,链路状态协议的基本要求是一个区域内的所有路由器必须具有一致的链路状态数据库。如果路由过滤阻挡了一些 LSA,那么将违反上面的要求。

- 使用命令 **distribute-list** 过滤特定路由
- 使用命令 **distance** 操作路由的管理距离

13.1.1 案例研究：过滤特定路由

图 13-2 显示出一个运行 RIPv2 互联网络的一部分。Barkis 经 Traddles 提供了到互联网络其余部分的连接。除了 BigNet 内 700 个明确的路由之外，Traddles 还向 Barkis 通告了一条缺省路由。由于这条缺省路由，Barkis、Micawber、Peggotty 和 Heep 不再需要知道 BigNet 中 700 条以外的路由。因此在 Barkis 上配置过滤器的目标是从 Traddles 仅接收缺省路由，并拒绝其他所有路由。Barkis 的配置如下：

```
router rip
  version 2
  network 192.168.75.0
  distribute-list 1 in Serial1
  !
ip classless
  access-list 1 permit 0.0.0.0
```

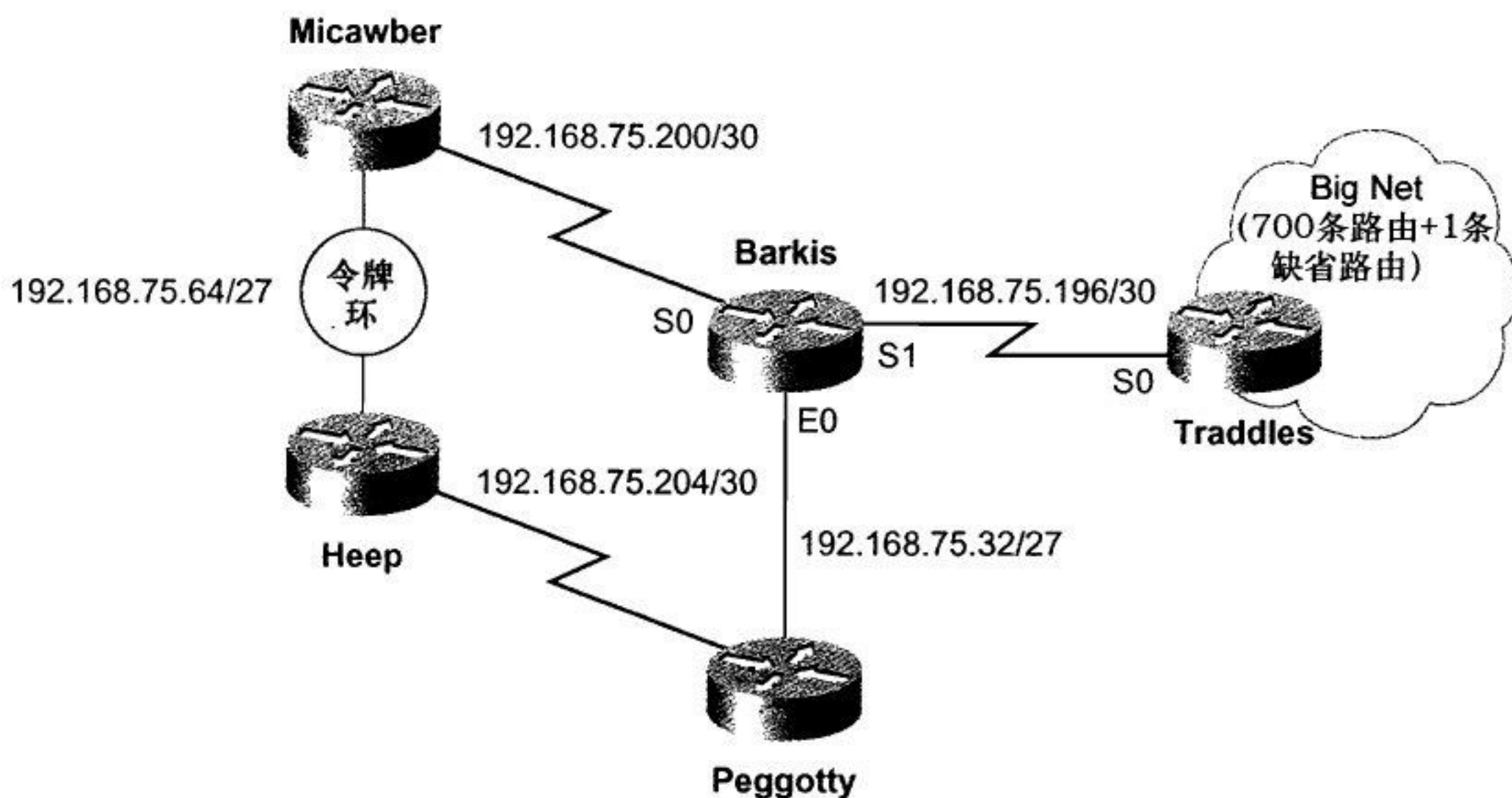


图 13-2 在 Barkis 上，路由过滤仅接收来自 Traddles 的缺省路由，并拒绝 BigNet 所有其他路由

该路由过滤器检查从接口 S1 入站的路由，其中 S1 是连接 Traddles 的接口。路由过滤器指定 Barkis 的 RIP 进程仅接收那些被访问列表 1 许可的路由，其中访问列表 1 指明仅允许 0.0.0.0。¹访问列表隐含地拒绝了所有其他路由。图 13-3 给出了 Barkis 的路由表。

沿串行链路通告的 700 条路由没想到会在链路的远端被全部丢弃，这自然是对带宽的浪费。因此更好的配置方法是将过滤器放在 Traddles 上，仅允许向 Barkis 通告缺省路由：

```
router rip
  version 2
```

¹ 注意没有给出反码。访问列表的缺省反码是 0.0.0.0，这对于本配置是正确的。


```

network 192.168.63.0
network 192.168.75.0
network 192.168.88.0
distribute-list 1 out Serial0
!
ip classless
access-list 1 permit 0.0.0.0

```

```

Barkis#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 192.168.75.198 to network 0.0.0.0

192.168.75.0/24 is variably subnetted, 5 subnets, 2 masks
C    192.168.75.32/27 is directly connected, Ethernet0
R    192.168.75.64/27 [120/1] via 192.168.75.201, 00:00:23, Serial0
C    192.168.75.196/30 is directly connected, Serial1
C    192.168.75.200/30 is directly connected, Serial0
R    192.168.75.204/30 [120/1] via 192.168.75.34, 00:00:13, Ethernet0
R*   0.0.0.0/0 [120/10] via 192.168.75.198, 00:00:03, Serial1
Barkis#

```

图 13-3 在来自 Traddles 的路由中, 0.0.0.0 是惟一被接受的

这里的过滤配置看上去与原来的配置差不多相同, 但它是过滤出站路由, 而不是进站路由。图 13-4 显示出从 Traddles 沿串行链路被通告的路由中仅包括缺省路由。

```

Barkis#debug ip rip
RIP protocol debugging is on
Barkis#
RIP: received v2 update from 192.168.75.198 on Serial1
    0.0.0.0/0 -> 0.0.0.0 in 10 hops
RIP: sending v2 update to 224.0.0.9 via Ethernet0 (192.168.75.33)
    192.168.75.64/27 -> 0.0.0.0, metric 2, tag 0
    192.168.75.196/30 -> 0.0.0.0, metric 1, tag 0
    192.168.75.200/30 -> 0.0.0.0, metric 1, tag 0
    0.0.0.0/0 -> 0.0.0.0, metric 11, tag 0
RIP: sending v2 update to 224.0.0.9 via Serial0 (192.168.75.202)
    192.168.75.32/27 -> 0.0.0.0, metric 1, tag 0
    192.168.75.196/30 -> 0.0.0.0, metric 1, tag 0
    192.168.75.204/30 -> 0.0.0.0, metric 2, tag 0
    0.0.0.0/0 -> 0.0.0.0, metric 11, tag 0
RIP: sending v2 update to 224.0.0.9 via Serial1 (192.168.75.197)
    192.168.75.32/27 -> 0.0.0.0, metric 1, tag 0
    192.168.75.64/27 -> 0.0.0.0, metric 2, tag 0
    192.168.75.200/30 -> 0.0.0.0, metric 1, tag 0
    192.168.75.204/30 -> 0.0.0.0, metric 2, tag 0
RIP: received v2 update from 192.168.75.34 on Ethernet0
    192.168.75.64/27 -> 0.0.0.0 in 2 hops
    192.168.75.204/30 -> 0.0.0.0 in 1 hops
RIP: received v2 update from 192.168.75.201 on Serial0
    192.168.75.64/27 -> 0.0.0.0 in 1 hops
    192.168.75.204/30 -> 0.0.0.0 in 2 hops

```

图 13-4 在 Traddles 上的过滤器仅允许向 Barkis 通告缺省路由

在这两种配置中，Barkis 将向 Micawber 和 Peggotty 通告缺省路由，没有配置可以影响 Barkis 向 Traddles 通告的路由。

当在链路状态协议（例如 OSPF）下配置命令 **distribute-list** 时，关键字 **out** 不能与接口联合使用¹，因为不像距离矢量协议，链路状态协议不从自身的路由表中通告路由，所以没有更新信息被过滤。所以命令 **distribute-list 1 out Serial1** 在链路状态协议下是没有意义的。

在图 13-5 中，还连接了另一组路由器。这部分互连网络——ThemNet，在独立的管理控制之下，Creakle 也一样。因为 BigNet 的管理员不能访问和控制 ThemNet 内的路由器，所以应使用路由过滤器将来自 Creakle 的错误路由信息的可能性减到最少。例如，ThemNet 正在使用缺省路由（或许为了访问内部 Internet 连接）。如果缺省路由被通告到 BigNet，那么它将导致报文被误转到 ThemNet 内，从而产生黑洞。

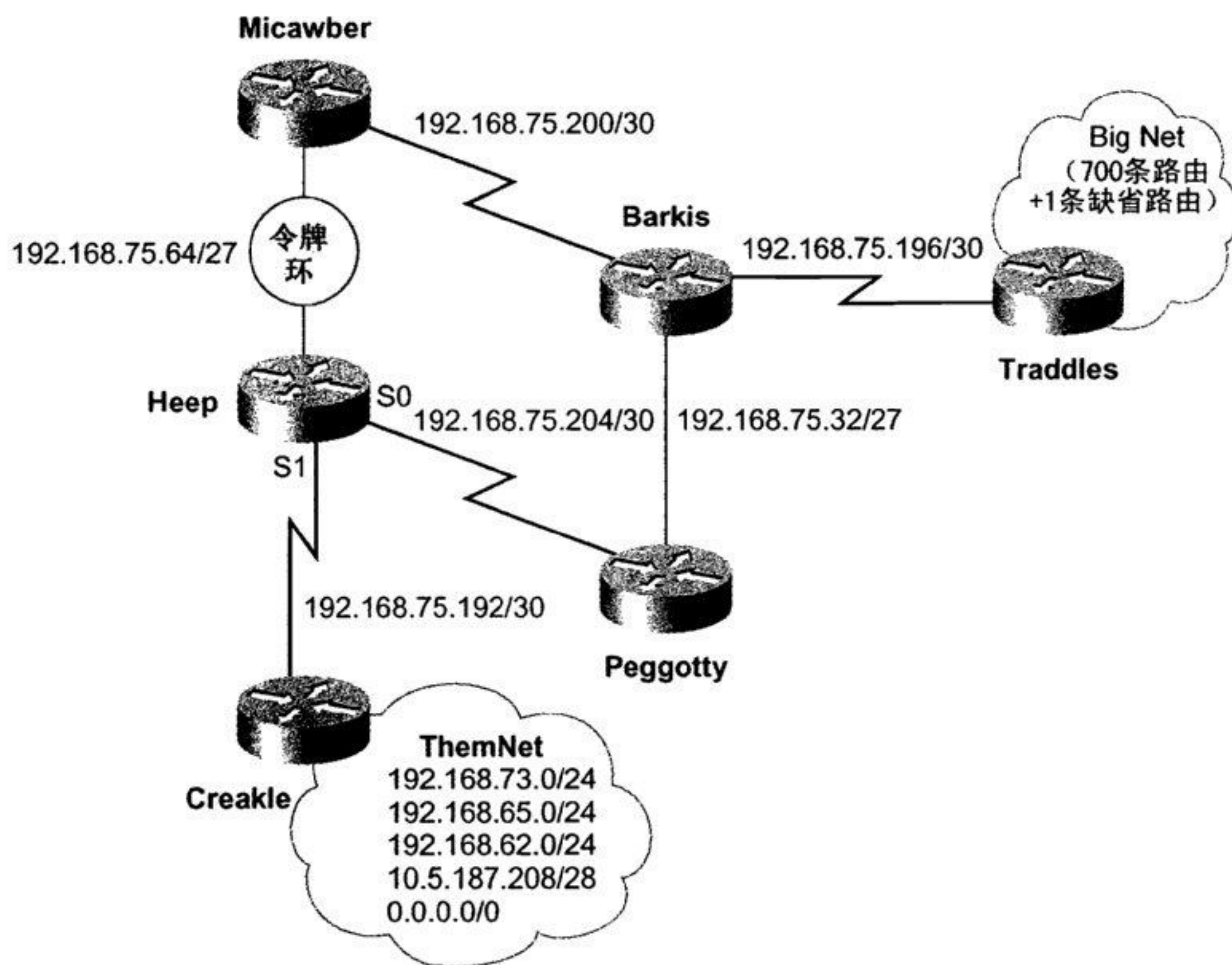


图 13-5 互连网络 ThemNet 不在 BigNet 管理员的控制之下

为了与 ThemNet 进行通信，仅允许必要的路由通过，Heep 相应的配置如下：

```
router rip
  version 2
  network 192.168.75.0
  distribute-list 2 out Serial1
  distribute-list 1 in Serial1
!
ip classless
access-list 1 permit 192.168.73.0
```

¹ 关键字 **out** 可以与一种路由选择协议联合使用，这将在下一个案例研究中讨论。


```

access-list 1 permit 192.168.65.0
access-list 1 permit 192.168.62.0
access-list 1 permit 10.5.187.208
access-list 2 deny 0.0.0.0
access-list 2 permit any

```

分布表 1 仅允许接受访问列表 1 指定的来自 Creakle 的路由。分布表阻挡了缺省路由和任何其他路由, 这些路由可能会被不正确地插入到 ThemNet 路由表中。

分布表 2 在适当的位置用于确保 BigNet 是一个正常的邻居, 它阻挡了 BigNet 的缺省路由, 否则该缺省路由将导致 ThemNet 内出现问题, 但是它允许 BigNet 的所有其他路由通过。

13.1.2 案例研究: 路由过滤和重新分配

路由器在任何时候执行相互重新分配时, 路由回馈都可能会存在。例如, 在图 13-6 中, 来自 RIP 方的路由会被重新分配到 OSPF, 并且再从 OSPF 重新分配回到 RIP。因此, 使用路由过滤器控制路由通告的方向将是一个明智的方法。

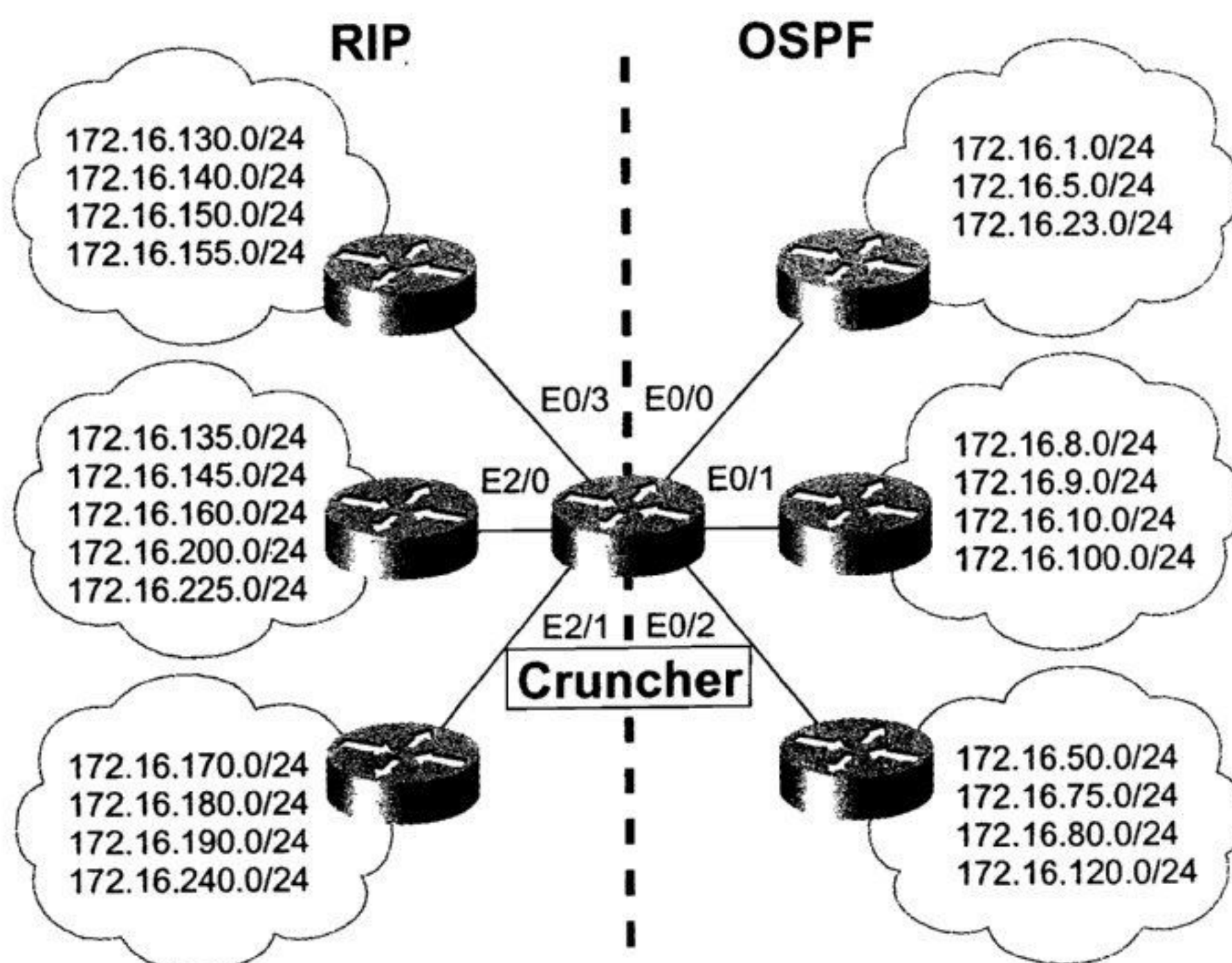


图 13-6 Cruncher 正在向 OSPF 重新分配 RIP 路由, 而且还向 RIP 重新分配 OSPF 路由。路由过滤将被用于阻止路由回馈

图 13-6 中的 Cruncher 可以在几个接口上同时使用 RIP 和 OSPF, Cruncher 的配置如下:

```

router ospf 25
 redistribute rip metric 100
 network 172.16.1.254 0.0.0.0 area 25
 network 172.16.8.254 0.0.0.0 area 25
 network 172.16.50.254 0.0.0.0 area 25
 distribute-list 3 in Ethernet0/0
 distribute-list 3 in Ethernet0/1

```



```

distribute-list 3 in Ethernet0/2
!
router rip
 redistribute ospf 25 metric 5
 passive-interface Ethernet0/0
 passive-interface Ethernet0/1
 passive-interface Ethernet0/2
 network 172.16.0.0
 distribute-list 1 in Ethernet0/3
 distribute-list 1 in Ethernet2/0
 distribute-list 1 in Ethernet2/1
!
ip classless
access-list 1 permit 172.16.128.0 0.0.127.255
access-list 3 permit 172.16.0.0 0.0.127.255

```

在上面的配置中，访问列表逻辑允许某些路由但拒绝所有其他路由。该逻辑也可以被颠倒过来，拒绝某些路由但允许所有其他路由。

```

access-list 1 deny 172.16.0.0 0.0.127.255
access-list 1 permit any
access-list 3 deny 172.16.128.0 0.0.127.255
access-list 3 permit any

```

第二个访问列表的作用同第一个访问列表相同。在这两种情况下，到 OSPF 域内目标网络的路由将不会从 RIP 向 OSPF 通告，同样到 RIP 域内目标网络的路由也不会从 OSPF 向 RIP 通告。然而，第二个访问列表的配置不易于管理，因为在表的末尾存在 **permit any**。为了向访问列表添加新的条目，整个表首先必须被删除以便在 **permit any** 之前可以放置新的条目。

为了容易地进行汇总，对图 13-6 中的子网地址作了精心的分配，所以出现小型访问列表是可能的，但牺牲了精确性。对路由的控制越精确，意味着访问列表越大、越精确，这是以增加管理的注意力为代价的。

在重新分配点配置路由过滤器的另一种方法是借助路由进程进行过滤，而不是接口。例如，下面的配置仅允许重新分配图 13-6 中的某些路由：

```

router ospf 25
 redistribute rip metric 100
 network 172.16.1.254 0.0.0.0 area 25
 network 172.16.8.254 0.0.0.0 area 25
 network 172.16.50.254 0.0.0.0 area 25
 distribute-list 10 out rip
!
router rip
 redistribute ospf 25 metric 5
 passive-interface Ethernet0/3
 passive-interface Ethernet2/0
 passive-interface Ethernet2/1
 network 172.16.0.0
 distribute-list 20 out ospf 25
!
ip classless

```



```

access-list 10 permit 172.16.130.0
access-list 10 permit 172.16.145.0
access-list 10 permit 172.16.240.0
access-list 20 permit 172.16.23.0
access-list 20 permit 172.16.9.0
access-list 20 permit 172.16.75.0

```

在 OSPF 配置下的路由过滤器允许 OSPF 通告 RIP 协议发现的路由, 但这些路由必须是访问表 10 许可的路由。同样, 在 RIP 配置下的路由过滤器允许 RIP 通告 OSPF25 发现的路由, 但这些路由必须是访问列表 20 许可的路由。在两种情况下, 路由过滤器对其他协议发现的路由没有影响。例如, 如果 OSPF 重新分配 RIP 和 EIGRP 的路由, 前面的分布表将不会应用到 EIGRP 发现的路由上。

当借助进程进行过滤时, 仅允许使用关键字 **out**。在 OSPF 下使用 **distribute-list 10 in rip** 是没有意义的, 因为路由已经通过 RIP 进入到路由表中了, OSPF 要么通告它 (out), 要么不通告。

注意: 虽然借助路由选择协议进行过滤对于指定那些将要被重新分配的路由是很有用处的, 但是它并不是防止路由回馈的好办法。例如, 考虑下面在图 13-6 中 Cruncher 的配置:

```

router ospf 25
 redistribute rip metric 100
 network 172.16.1.254 0.0.0.0 area 25
 network 172.16.8.254 0.0.0.0 area 25
 network 172.16.50.254 0.0.0.0 area 25
 distribute-list 1 out rip
!
router rip
 redistribute ospf 25 metric 5
 passive-interface Ethernet0/3
 passive-interface Ethernet2/0
 passive-interface Ethernet2/1
 network 172.16.0.0
 distribute-list 3 out ospf 25
!
ip classless
access-list 1 permit 172.16.128.0 0.0.127.255
access-list 3 permit 172.16.0.0 0.0.127.255

```

假设一条来自 RIP 域的路由, 如 172.16.190.0/24, 被重新分配到 OSPF 域, 然后又被通告回到 Cruncher。虽然在 RIP 配置下的分布表将会阻止路由被通告回 RIP 域, 但是它却不能阻止路由以 OSPF 域发生的路由的身份进入 Cruncher 的路由表。事实上, 过滤器认为路由已经通过 OSPF 进入路由表。为了阻止路由回馈, 必须在路由进入路由表之前, 在路由进站时进行过滤。

13.1.3 案例研究: 协议迁移

命令 **distance** 可以为路由指定管理距离, 这些路由是从一个特殊路由选择协议那里学

习到的。命令在使用时不带任何可选参数。在最初考虑时，该操作看上去不像路由过滤功能，但是当运行多个路由选择协议时就不同了，这时候将会基于路由的管理距离来确定是否接受或拒绝路由。

在图 13-7 中，互连网络运行 RIP，并且计划将路由选择协议转换为 EIGRP。可以有好几种办法完成这样的路由迁移。一种方法是在每一个路由器上关闭老的协议，然后打开新的协议。虽然这种方法对类似于图 13-7 的小型互连网络来说是合适的，但在更大的互连网络中，这种停工其是不切实际的。

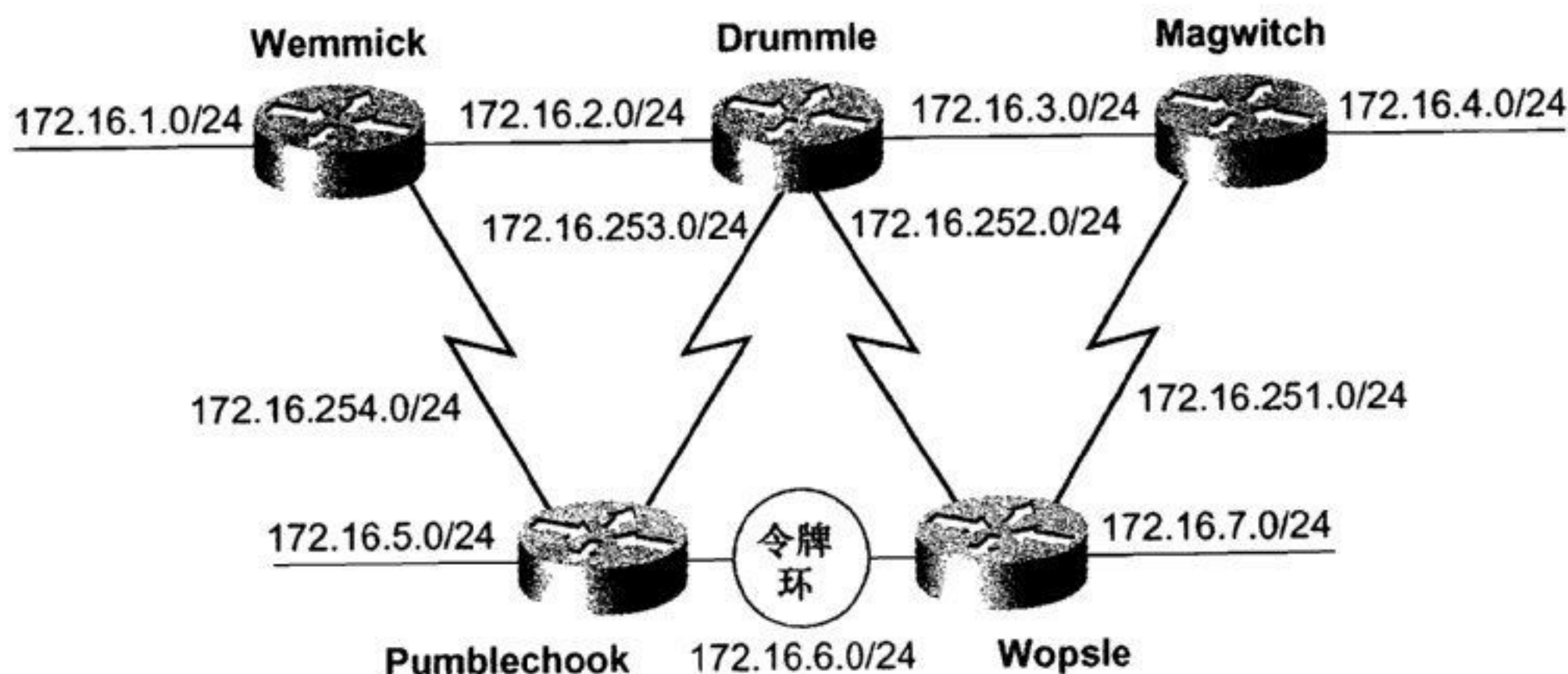


图 13-7 这些运行 RIP 的路由器将会被转为运行 EIGRP

另一种选择是不删除旧协议的同时添加新的协议。如果新协议的缺省管理距离小于老协议，那么每个路由器将选择新协议通告的路由。随着路由器转换完毕，互连网络将收敛到新的协议上。在整个互连网络收敛到新的协议之后，则可以从所有路由器上删除旧的协议。

参照表 11-1，RIP 的缺省管理距离是 120，EIGRP 的缺省管理距离为 90。除了 RIP 之外，如果 EIGRP 被添加到每个路由器，那么路由器将随着邻居路由器开始使用 EIGRP 也开始选择使用 EIGRP 路由。当所有 RIP 路由从路由表中消失时，互连网络将收敛到 EIGRP。RIP 进程接着会从路由器上被删除。

这种方法的问题是在重新配置的时候可能会存在路由环路和黑洞。在图 13-7 中，大约几分钟就可以完成对 5 个路由器的重新配置和转换，所以这里不会像大型互连网络一样关注环路问题。

对这种双协议方法的改进是使用命令 **distance**，以确保禁止新协议的路由一直到所有路由器为变换做好准备。这个过程中第一步是在所有路由器上降低 RIP 的管理距离：

```
router rip
network 172.16.0.0
distance 70
```

注意，管理距离仅与单一路由器的路由进程相关。当 RIP 仍然是惟一正在运行的协议时，管理距离的改变不会影响到路由。

下一步是重新访问路由器，添加 EIGRP 进程：

```
router eigrp 1
network 172.16.0.0
```



```
!
router rip
network 172.16.0.0
distance 70
```

由于 EIGRP 的缺省管理距离是 90, 因此 RIP 的路由被选择 (图 13-8)。因为没有路由器选择 EIGRP 路由, 所以在配置期间不必确定停工时间表。这种方式使网络管理员在转换之前有时间重新检查每一个路由器的新配置。

```
Drummle#show ip route
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, 11 subnets
C    172.16.252.0 is directly connected, Serial0
C    172.16.253.0 is directly connected, Serial1
R    172.16.254.0 [70/1] via 172.16.2.253, 00:00:16, Ethernet0
      [70/1] via 172.16.253.253, 00:00:05, Serial1
R    172.16.251.0 [70/1] via 172.16.252.253, 00:00:08, Serial0
      [70/1] via 172.16.3.253, 00:00:01, Ethernet1
R    172.16.4.0 [70/1] via 172.16.3.253, 00:00:01, Ethernet1
R    172.16.5.0 [70/1] via 172.16.253.253, 00:00:05, Serial1
R    172.16.6.0 [70/1] via 172.16.252.253, 00:00:08, Serial0
      [70/1] via 172.16.253.253, 00:00:05, Serial1
R    172.16.7.0 [70/1] via 172.16.252.253, 00:00:08, Serial0
R    172.16.1.0 [70/1] via 172.16.2.253, 00:00:17, Ethernet0
C    172.16.2.0 is directly connected, Ethernet0
C    172.16.3.0 is directly connected, Ethernet1
Drummle#
```

图 13-8 由于为 RIP 路由分配的管理距离为 70, 所以优先选择的是 RIP 路由, 而不是 EIGRP 路由

最后, 再一次访问每个路由器, 将 RIP 的距离改回到 120。这一步要计划停工时间。因为新的 RIP 更新路由的管理距离被指派为 120, 所以管理距离为 70 的 RIP 路由将会超时 (图 13-9)。210s 之后, 宣布 RIP 路由失效 (图 13-10), 最终 EIGRP 的路由将被选择 (图 13-11)。

```
Drummle#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, 11 subnets
C    172.16.252.0 is directly connected, Serial0
C    172.16.253.0 is directly connected, Serial1
R    172.16.254.0 [70/1] via 172.16.2.253, 00:02:31, Ethernet0
```

待续


```

          [70/1] via 172.16.253.253, 00:02:18, Serial1
R   172.16.251.0 [70/1] via 172.16.252.253, 00:02:27, Serial0
          [70/1] via 172.16.3.253, 00:02:32, Ethernet1
R   172.16.4.0 [70/1] via 172.16.3.253, 00:02:32, Ethernet1
R   172.16.5.0 [70/1] via 172.16.253.253, 00:02:19, Serial1
R   172.16.6.0 [70/1] via 172.16.252.253, 00:02:27, Serial0
          [70/1] via 172.16.253.253, 00:02:19, Serial1
R   172.16.7.0 [70/1] via 172.16.252.253, 00:02:27, Serial0
R   172.16.1.0 [70/1] via 172.16.2.253, 00:02:32, Ethernet0
C   172.16.2.0 is directly connected, Ethernet0
C   172.16.3.0 is directly connected, Ethernet1

```

图 13-9 在 RIP 的管理距离被改回到 120 之后, 管理距离为 70 的路由开始老化。这里所有 RIP 路由老化时间都已超过 2min

虽然使用这种方法仍然有可能产生路由环路和黑洞, 但是因为仅需要修改管理距离, 所以转换速度会更快, 所犯的人为错误也越小。

这种方法的另一优点是万一出现问题, 可以很容易停止切换工作。在所有路由器中 RIP 进程仍然处于合适的位置, 退回 RIP 需要做的所有工作就是将管理距离改回 120。一旦新的 EIGRP 被测试并证明是稳定后, 可以从所有路由器上删除 RIP 进程而不会有进一步的服务中断。

```

Drummler#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, 11 subnets
C       172.16.252.0 is directly connected, Serial0
C       172.16.253.0 is directly connected, Serial1
R       172.16.254.0/24 is possibly down,
         routing via 172.16.253.253, Serial1
R       172.16.251.0/24 is possibly down,
         routing via 172.16.252.253, Serial0
R       172.16.4.0/24 is possibly down,
         routing via 172.16.3.253, Ethernet1
R       172.16.5.0/24 is possibly down,
         routing via 172.16.253.253, Serial1
R       172.16.6.0/24 is possibly down,
         routing via 172.16.253.253, Serial1
R       172.16.7.0/24 is possibly down,
         routing via 172.16.252.253, Serial0
R       172.16.1.0/24 is possibly down,
         routing via 172.16.2.253, Ethernet0
C       172.16.2.0 is directly connected, Ethernet0
C       172.16.3.0 is directly connected, Ethernet1

```

Drummler#

图 13-10 经过 3.5min, RIP 路由将被宣布失效

在使用双协议方法之前需要考虑一件事情, 即同时在每个路由器上运行两个协议可能会影响路由器的内存用量和处理速度。如果内存利用率、处理利用率或两者利用率平均值超过 50% 或 60%, 那么在提交转换工作之前应该进行仔细的实验室测试和仿真, 以确保路由器可以处理这些额外的负载。如果路由器不行, 那么在配置新协议之前需要更加复杂的过程来删除旧协议, 这可能是惟一的选择。

```
Drummler#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, 11 subnets
C      172.16.252.0 is directly connected, Serial0
C      172.16.253.0 is directly connected, Serial1
D      172.16.254.0 [90/2195456] via 172.16.2.253, 00:01:11, Ethernet0
D      172.16.251.0 [90/2195456] via 172.16.3.253, 00:01:06, Ethernet1
D      172.16.4.0 [90/307200] via 172.16.3.253, 00:01:06, Ethernet1
D      172.16.5.0 [90/2195456] via 172.16.253.253, 00:01:11, Serial1
D      172.16.6.0 [90/2185984] via 172.16.252.253, 00:01:11, Serial0
           [90/2185984] via 172.16.253.253, 00:01:11, Serial1
D      172.16.7.0 [90/2195456] via 172.16.252.253, 00:01:07, Serial0
D      172.16.1.0 [90/307200] via 172.16.2.253, 00:01:11, Ethernet0
C      172.16.2.0 is directly connected, Ethernet0
C      172.16.3.0 is directly connected, Ethernet1
Drummler#
```

图 13-11 缺省管理距离为 90 的 EIGRP 路由代替了 RIP 路由

与上面过程不同之处在于, 这里是增加新协议的管理距离, 而不是降低旧协议的管理距离, 接着在转换时再降低新协议的管理距离。但是, 要确保在输入任何网络命令之前输入命令 **distance**, 以便不会用新协议缺省的管理距离激活新协议。

回顾一下表 11-1, 注意 EIGRP 有两个管理距离: 对内部路由为 90, 对外部路由为 170。因此, 对 EIGRP 来说, 命令 **distance** 还有些不同。如, 用提高 EIGRP 的管理距离代替降低 RIP 的管理距离, 相应配置如下:

```
router eigrp 1
 network 172.16.0.0
 distance eigrp 130 170
!
router rip
 network 172.16.0.0
```

添加关键字 **eigrp** 以说明指定的 EIGRP 管理距离。内部 EIGRP 路由的管理距离被改为 130, 而外部路由的管理距离仍为 170。

使用双协议迁移到新的路由选择协议还需注意的一点是: 确认你已经理解两种协议的行为。例如, 某些协议 (如 EIGRP) 不会老化自身的路由条目。因此, 如果要取代 EIGRP, 需要增加额外的一步, 就是在改变完管理距离之后使用命令 **clear ip route *** 清除路由表。

13.1.4 多个重新分配点

图 13-12 给出的互联网络非常类似于图 11-3 给出的互联网络。回忆一下在第 11 章关于多个重新分配点问题的讨论，由多个重新分配点可以引起路由器选择非最佳路由的问题。在某些情况下，还会导致路由环路和黑洞。例如，在 Bumble 的路由表中，指向网络 192.168.6.0 的路由的下一跳路由器是 Blathers，而不是 Monks。

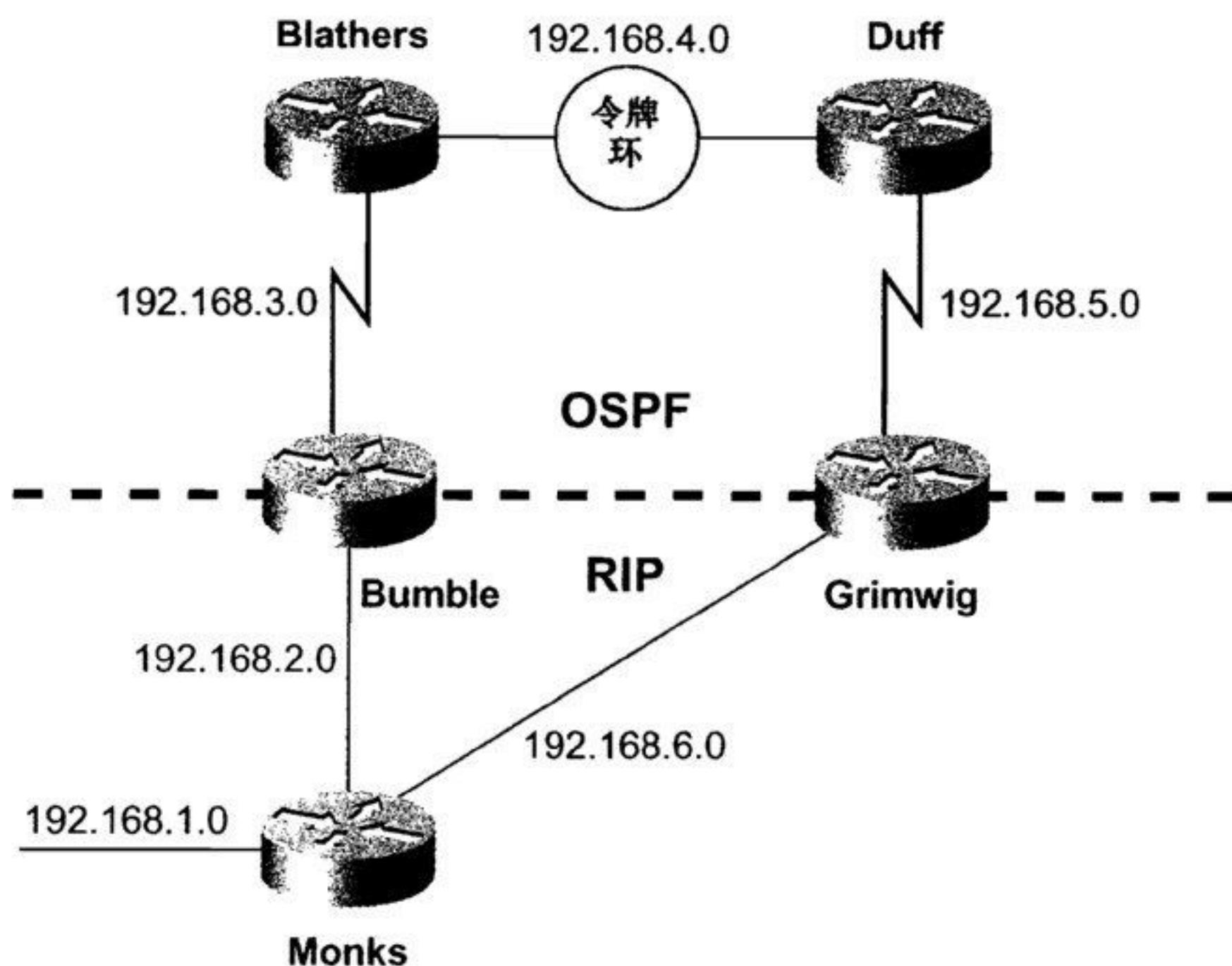


图 13-12 当在多个点进行互相重新分配时，管理距离会导致非最佳路径选择、路由环路和黑洞

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

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:00, Ethernet0
C    192.168.2.0/24 is directly connected, Ethernet0
C    192.168.3.0/24 is directly connected, Serial0
O    192.168.4.0/24 [110/70] via 192.168.3.2, 00:05:09, Serial0
O    192.168.5.0/24 [110/134] via 192.168.3.2, 00:05:09, Serial0
O E2 192.168.6.0/24 [110/100] via 192.168.3.2, 00:05:09, Serial0
Bumble#
```

图 13-13 使用 Bumble 的路由可以经 Blathers (192.168.3.2) 到达 192.168.6.0，这里包括两条串行链路和一个令牌环

解决这个问题一个办法是在重新分配点使用命令 **distribute-list** 控制路由源点。Bumble 和 Grimwig 的配置如下：

路由器 Bumble:


```

router ospf 1
 redistribute rip metric 100
 network 192.168.3.1 0.0.0.0 area 0
 distribute-list 1 in
!
router rip
 redistribute ospf 1 metric 2
 network 192.168.2.0
 distribute-list 2 in
!
ip classless
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.5.0
access-list 2 permit 192.168.1.0
access-list 2 permit 192.168.6.0

```

路由器 Grimwig:

```

router ospf 1
 redistribute rip metric 100
 network 192.168.5.1 0.0.0.0 area 0
 distribute-list 1 in
!
router rip
 redistribute ospf 1 metric 2
 network 192.168.6.0
 distribute-list 2 in
!
no ip classless
access-list 1 permit 192.168.3.0
access-list 1 permit 192.168.4.0
access-list 2 permit 192.168.1.0
access-list 2 permit 192.168.2.0

```

在上面两个配置中, 访问列表 1 仅允许 OSPF 接受 OSPF 域内的网络, 访问 2 仅允许接受 RIP 域内的网络。在配置了路由过滤器之后, 图 13-14 给出了 Bumble 的路由表。

```

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

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:12, Ethernet0
C    192.168.2.0/24 is directly connected, Ethernet0
C    192.168.3.0/24 is directly connected, Serial0
O    192.168.4.0/24 [110/70] via 192.168.3.2, 00:00:22, Serial0
O    192.168.5.0/24 [110/134] via 192.168.3.2, 00:00:22, Serial0
R    192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:13, Ethernet0
Bumble#

```

图 13-14 在配置路由过滤之后, Bumble 将使用最佳路由到达网络 192.168.6.0

使用这个配置方法的问题是消除了多个重新分配点内在的冗余性。在图 13-15 中, Bumble 的以太网链路被断开。由于在 OSPF 中过滤掉指向 RIP 网络的路由, 因而所有路由现在都不可达。

```
Bumble#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
Bumble#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

C    192.168.3.0/24 is directly connected, Serial0
O    192.168.4.0/24 [110/70] via 192.168.3.2, 00:06:45, Serial0
O    192.168.5.0/24 [110/134] via 192.168.3.2, 00:06:45, Serial0
Bumble#
```

图 13-15 当 Bumble 的以太网链路发生故障后, RIP 网络变得不可达。路由过滤器可阻止 OSPF 向路由表中输入替代的路由

一个更好的方法是使用命令 **distance** 的两种方式设置首选路由。Bumble 和 Grimwig 的配置如下:

路由器 Bumble:

```
router ospf 1
 redistribute rip metric 100
 network 192.168.3.1 0.0.0.0 area 0
 distance 130
 distance 110 0.0.0.0 255.255.255.255 1
!
router rip
 redistribute ospf 1 metric 2
 network 192.168.2.0
 distance 130
 distance 120 192.168.2.1 0.0.0.0 2
!
ip classless
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.5.0
access-list 2 permit 192.168.1.0
access-list 2 permit 192.168.6.0
```

路由器 Grimwig:

```
router ospf 1
 redistribute rip metric 100
 network 192.168.5.1 0.0.0.0 area 0
 distance 130
 distance 110 0.0.0.0 255.255.255.255 1
!
```



```

router rip
 redistribute ospf 1 metric 2
 network 192.168.6.0
 distance 130
 distance 120 192.168.6.1 0.0.0.0 2
!
ip classless
access-list 1 permit 192.168.3.0
access-list 1 permit 192.168.4.0
access-list 2 permit 192.168.1.0
access-list 2 permit 192.168.2.0

```

在这两个配置文件中, 第一个 **distance** 命令设置了 OSPF 和 RIP 的管理距离为 130。第二个 **distance** 命令根据被指定的通告路由器和参考访问列表来设定一个不同的管理距离。例如, 由 Monks (192.168.6.1) 通告并且被访问列表 2 许可的路由, Grimwig 的 RIP 进程为这些路由指定的管理距离为 120, 所有其他路由的管理距离为 130。注意, 这里与通告路由器地址一起使用的是反码。

在 OSPF 的配置中会存在更多的问题。通告路由器的地址不必是下一跳路由器的接口地址, 而是产生 LSA 路由器的标识号, 路由就是根据 LSA 进行计算的。因此对命令 **distance** 来说, 地址和反码是 0.0.0.0 255.255.255.255, 它们可以指定任意路由器。¹访问列表 1 许可的 OSPF 路由被指派的管理距离为 110, 所有其他路由的管理距离为 130。

结果如图 13-16 所示, 第一个路由表表示如果要到达 OSPF 域内的所有网络, Grimwig 选择经过 Duff 的路径, 如果要到达 RIP 域内的所有网络, Grimwig 则选择经过 Monks 路径。对 OSPF 路由管理距离为 110, 对 RIP 管理距离为 120。接着, 如果 Grimwig 的以太网链路发生故障。第二个路由表显示所有网络均通告 Duff 可达。指向 RIP 域内网络的路由的管理距离为 130。当以太网链路恢复后, 来自 Monks 且管理距离为 120 的 RIP 通告将取代管理距离为 130 的 OSPF 通告。

```

Grimwig#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
       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

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.6.1, 00:00:19, Ethernet0
R    192.168.2.0/24 [120/1] via 192.168.6.1, 00:00:19, Ethernet0
O    192.168.3.0/24 [110/134] via 192.168.5.2, 00:15:06, Serial0
O    192.168.4.0/24 [110/70] via 192.168.5.2, 00:15:06, Serial0
C    192.168.5.0/24 is directly connected, Serial0
C    192.168.6.0/24 is directly connected, Ethernet0
Grimwig#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
Grimwig#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

```

待续

¹ 相同的“任意”地址也可以和 RIP 一起使用, 而使用一个特殊地址完全是为了示范目的。


```

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
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

Gateway of last resort is not set

O E2 192.168.1.0/24 [130/100] via 192.168.5.2, 00:00:08, Serial0
O E2 192.168.2.0/24 [130/100] via 192.168.5.2, 00:00:08, Serial0
O   192.168.3.0/24 [110/134] via 192.168.5.2, 00:16:23, Serial0
O   192.168.4.0/24 [110/70] via 192.168.5.2, 00:16:23, Serial0
C   192.168.5.0/24 is directly connected, Serial0
O E2 192.168.6.0/24 [130/100] via 192.168.5.2, 00:00:08, Serial0
Grimwig#

```

图 13-16 在到 Monks 以太网链路发生故障之前和之后的 Grimwig 的路由表

在图 13-12 中，如果其中一条串行链路发生故障，那么将会发生相反的事情。穿过 RIP 域将可以到达 OSPF 域内的网络，而且管理距离再次是 130（图 3-17）。然而，不像 OSPF 可以快速地收敛，RIP 要花几分钟才能收敛。这种慢收敛是由于在 Monks 处 RIP 的水平分割所导致的。Monks 将不会向 Bumble 和 Grimwig 通告 OSPF 路由，直到这两个路由器停止通告相同的路由并且现存的路由超时为止。

解决问题的办法是关闭 Monks 两个以太网接口上的水平分割功能，这可以使用命令 **no ip split-horizon** 完成。虽然关闭水平分割缩短了收敛时间，但同时也失去了环路保护功能，不过还是值得的。在 Bumble 和 Grimwig 上基于管理距离的路由过滤可以阻止所有多跳环路，而且在两个路由器的相同以太网接口上的水平分割功能还可以打断单跳环路。

```

Grimwig#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
       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

Gateway of last resort is not set

R   192.168.1.0/24 [120/1] via 192.168.6.1, 00:00:04, Ethernet0
R   192.168.2.0/24 [120/1] via 192.168.6.1, 00:00:04, Ethernet0
O   192.168.3.0/24 [110/134] via 192.168.5.2, 00:00:12, Serial0
O   192.168.4.0/24 [110/70] via 192.168.5.2, 00:00:12, Serial0
C   192.168.5.0/24 is directly connected, Serial0
C   192.168.6.0/24 is directly connected, Ethernet0
Grimwig#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
%LINK-3-UPDOWN: Interface Serial0, changed state to down
Grimwig#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
       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

```

待续


```

Gateway of last resort is not set

R   192.168.1.0/24 [120/1] via 192.168.6.1, 00:00:07, Ethernet0
R   192.168.2.0/24 [120/1] via 192.168.6.1, 00:00:07, Ethernet0
R   192.168.3.0/24 [130/3] via 192.168.6.1, 00:00:07, Ethernet0
R   192.168.4.0/24 [130/3] via 192.168.6.1, 00:00:07, Ethernet0
R   192.168.5.0/24 is possibly down, routing via 192.168.6.1, Ethernet0
C   192.168.6.0/24 is directly connected, Ethernet0
Grimwig#

```

图 13-17 到 Duff 串行链路发生故障之前和之后的 Grimwig 的路由表

13.1.5 案例研究：使用距离设置路由器优先权

在图 13-12 中，假设策略规定使用 Grimwig 作为到 OSPF 域的主路由器，仅当 Grimwig 不可达时才选择经过 Bumble 的路由。目前，为了到达 OSPF 的网络，Monks 正在 Grimwig 和 Bumble 之间执行等价负载均衡（图 3-18）。

```

Monks#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

C   192.168.1.0/24 is directly connected, Ethernet2
C   192.168.2.0/24 is directly connected, Ethernet0
R   192.168.3.0/24 [120/2] via 192.168.2.2, 00:00:26, Ethernet0
                        [120/2] via 192.168.6.2, 00:00:23, Ethernet1
R   192.168.4.0/24 [120/2] via 192.168.2.2, 00:00:26, Ethernet0
                        [120/2] via 192.168.6.2, 00:00:23, Ethernet1
R   192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:26, Ethernet0
                        [120/2] via 192.168.6.2, 00:00:23, Ethernet1
C   192.168.6.0/24 is directly connected, Ethernet1
Monks#

```

图 13-18 Monks 把从 Bumble 和 Grimwig 到 OSPF 域内所有网络都看成是等距离的

通过降低来自 Grimwig 的路由的管理距离，可以使得 Monks 优先选择 Grimwig:

```

router rip
 network 192.168.1.0
 network 192.168.2.0
 network 192.168.6.0
 distance 100 192.168.6.2 0.0.0.0

```

在这里，命令 **distance** 没有参考访问列表。所有 Grimwig 通告的路由的管理距离都将被指定为 100。所有其他路由都被指定为 RIP 的缺省管理距离 120。因此将优先选择 Grimwig 的路由。

图 13-19 给出了结果。第一个路由表显示了 Monks 仅路由到 Grimwig。当到 Grimwig 的连接发生故障时，路由表结尾显示 Monks 将切换到 Bumble (192.168.2.2)。

```
Monks#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

C    192.168.1.0/24 is directly connected, Ethernet2
C    192.168.2.0/24 is directly connected, Ethernet0
R    192.168.3.0/24 [100/2] via 192.168.6.2, 00:00:12, Ethernet1
R    192.168.4.0/24 [100/2] via 192.168.6.2, 00:00:12, Ethernet1
R    192.168.5.0/24 [100/2] via 192.168.6.2, 00:00:12, Ethernet1
C    192.168.6.0/24 is directly connected, Ethernet1
Monks#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1, changed state to down
Monks#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

C    192.168.1.0/24 is directly connected, Ethernet2
C    192.168.2.0/24 is directly connected, Ethernet0
R    192.168.3.0/24 [120/2] via 192.168.2.2, 00:00:00, Ethernet0
R    192.168.4.0/24 [120/2] via 192.168.2.2, 00:00:00, Ethernet0
R    192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:00, Ethernet0
R    192.168.6.0/24 [120/2] via 192.168.2.2, 00:00:00, Ethernet0
Monks#
```

图 13-19 到 Grimwig 的以太网链路发生故障之前和之后的 Monks 的路由表

13.2 展 望

对控制互联网络的行为来说，路由过滤器是一个非常有用的工具。在大型互联网络中，路由过滤器几乎是不可缺少的。但是路由过滤器的所有效用仅限于允许或不允许路由。第 14 章“路由图”介绍另一个强大的工具路由图，路由图不仅可以标识路由，还可以主动地修改路由。

13.3 总结表：第 13 章命令回顾

命 令				描 述
access-list	access-list-number	{deny permit}	source	定义标准访问列表中的一条
[source-wildcard]				

续表

命 令	描 述
distance weight [<i>address mask</i> [<i>access-list-number</i> name]]	定义管理距离, 该值不同于缺省值的
distance eigrp internal-distance external-distance	为 EIGRP 内部和外部路由定义管理距离, 该值不同于缺省值
distribute-list { <i>access-list-number</i> name} in [<i>interface-name</i>]	过滤入站更新路由
distribute-list { <i>access-list-number</i> name} out [<i>interface-name</i> routing-process autonomous-system-number]	过滤出站更新路由
redistribute protocol [<i>process-id</i>]{ <i>level-1</i> <i>level-1-2</i> <i>level-2</i> }[<i>metric metric-value</i>][<i>metric-type type-value</i>][<i>match</i> { <i>internal</i> <i>external 1</i> <i>external 2</i> }][<i>tag tag-value</i>] [<i>route-map map-tag</i>][<i>weight weight</i>][<i>subnets</i>]	配置向一路由选择协议重新分配路由, 并指明被重新分配路由的源点

13.4 配置练习

1. 在图 13-20 中, 路由器 A 的路由配置如下:

```

router rip
 redistribute igrp 1 metric 3
 passive-interface Ethernet0
 passive-interface Ethernet1
 network 172.16.0.0
!
router igrp 1
 redistribute rip metric 10000 1000 255 1 1500
 passive-interface Ethernet2
 passive-interface Ethernet3
 network 172.16.0.0

```

请在 A 上配置路由过滤器, 防止除 E 之外的其他路由器知道子网 172.16.12.0/24。

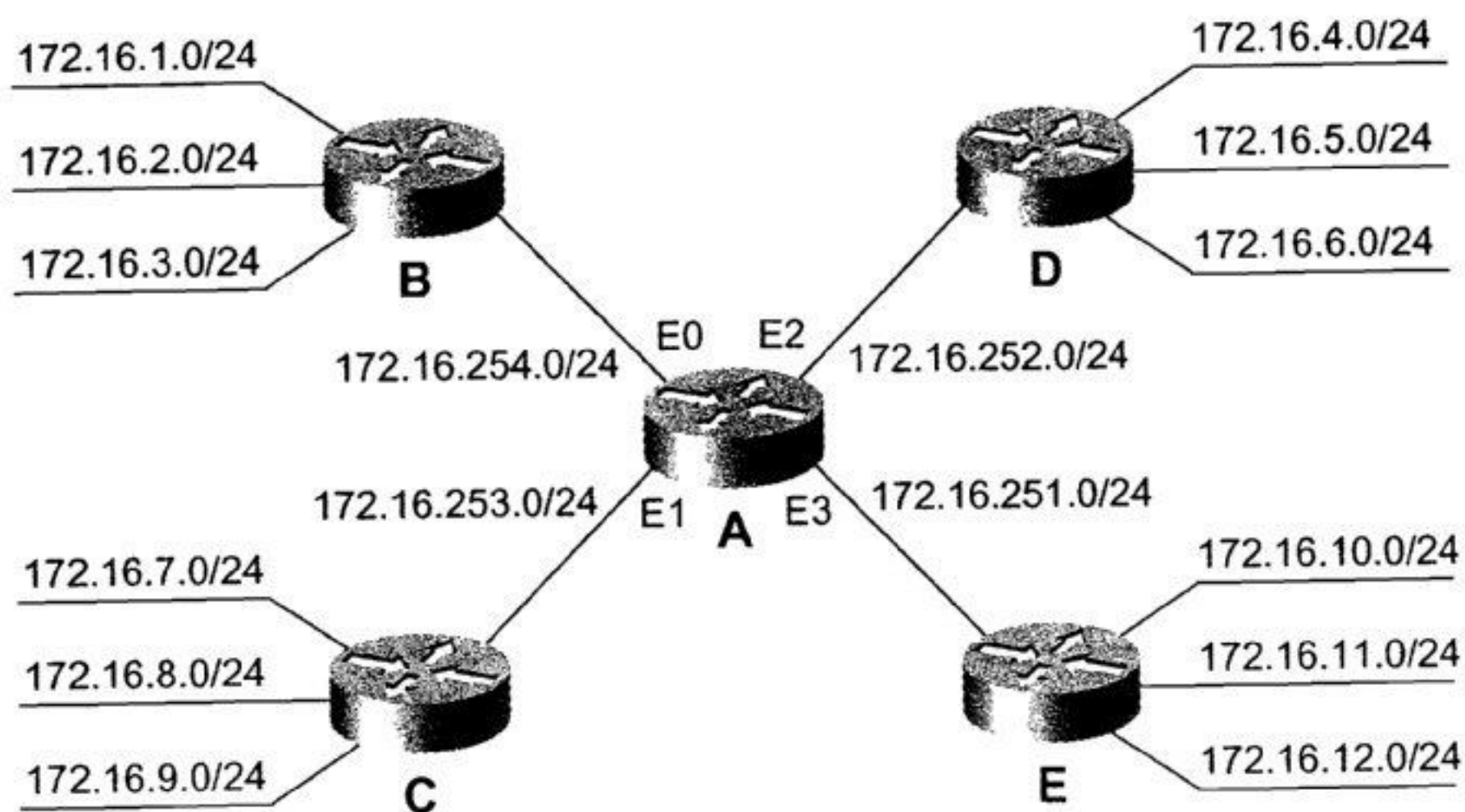


图 13-20 配置练习 1~4 中的互连网络

2. 图 13-20 中, 在路由器 A 上配置路由过滤器, 阻止 D 知道子网 172.16.10.0/24。
3. 图 13-20 中, 在路由器 A 上配置路由过滤器, 仅允许向 RIP 域通告子网 172.16.2.0/24、

172.16.8.0/24 和 172.16.9.0/24。

4. 图 13-20 中，在路由器 A 上配置路由过滤器，阻止 B 学习到 RIP 域内的任何子网。

5. 表 13-1 给出了图 13-21 中所有路由器的接口地址。路由器 A 和 B 运行 EIGRP，E 和 F 运行 IS-IS。C 和 D 正在重新分配。为 C 和 D 配置命令 **disntance**，阻止路由环路和路由回馈，但允许存在冗余路由。

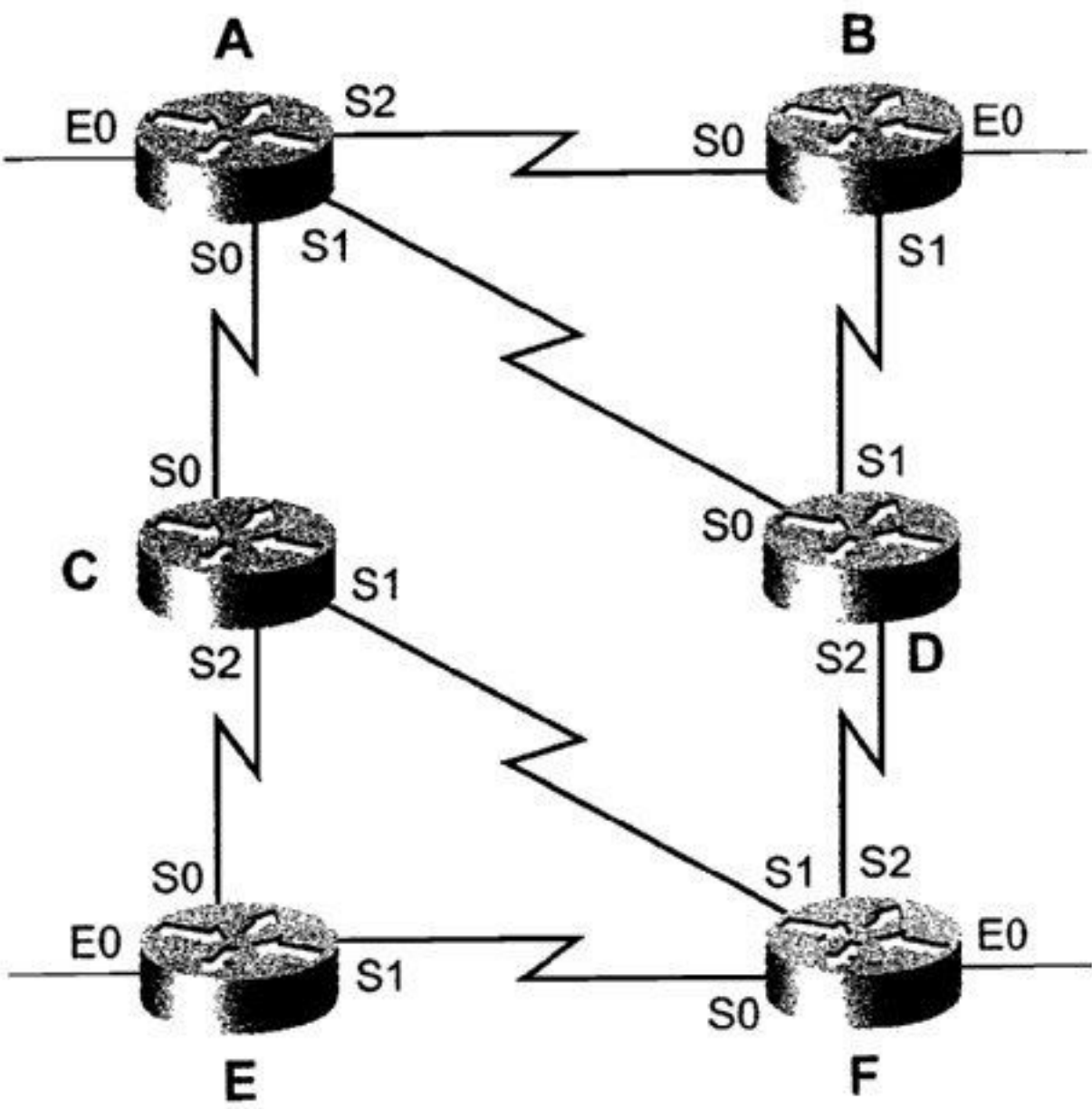


图 13-21 配置练习 5~7 的互联网络

表 13-1

图 13-21 中所有路由器的接口地址

路 由 器	接 口	地 址	掩 码
A	E0	192.168.1.1	255.255.255.0
	S0	192.168.10.254	255.255.255.252
	S1	192.168.10.249	255.255.255.252
	S2	192.168.10.245	255.255.255.252
B	E0	192.168.2.1	255.255.255.0
	S0	192.168.10.246	255.255.255.252
	S1	192.168.10.241	255.255.255.252
C	S0	192.168.10.253	255.255.255.252
	S1	192.168.10.234	255.255.255.252
	S2	192.168.10.225	255.255.255.252
D	E0	192.168.10.250	255.255.255.252
	S1	192.168.10.242	255.255.255.252
	S2	192.168.10.237	255.255.255.252
E	E0	192.168.4.1	255.255.255.0
	S0	192.168.10.226	255.255.255.252
	S1	192.168.10.229	255.255.255.252
F	E0	192.168.3.1	255.255.255.0
	S0	192.168.10.230	255.255.255.252
	S1	192.168.10.233	255.255.255.252
	S2	192.168.10.238	255.255.255.252

6. 在图 13-21 中，使用命令 **distance** 配置路由器 D，使它仅接受来自路由器 A 的 EIGRP 路由。如果到 A 的链路发生故障，那么 D 将不接受来自 B 的路由，虽然 D 仍旧向 B 通告路由。

7. 删除练习 6 中添加到 D 的配置, 配置图 13-21 中的路由器 C, 使它通过路由器 A 可以到达所有目标网络, 包括 IS-IS 域内的所有子网。仅在 A 发生故障时, C 才选择经过 E 和 F 进行路由。

13.5 故障诊断练习

1. 路由器的配置如下:

```
router igrp 1
 network 10.0.0.0
 distribute-list 1 in Ethernet5/1
!
access-list 1 deny 0.0.0.0 255.255.255.255
access-list 1 permit any
```

上面配置的目的是阻止进入接口 E5/1 的缺省路由, 而允许进入该接口的所有其他路由。但是, 在接口 E5/1 路由器却没有接受任何路由, 错误在哪里?

2. 在图 13-12 中, Grimwig 的配置如下:

```
router ospf 1
 redistribute rip metric 100
 network 192.168.5.1 0.0.0.0 area 0
 distance 255
 distance 110 0.0.0.0 255.255.255.255 1
!
router rip
 redistribute ospf 1 metric 2
 network 192.168.6.0
 distance 255
 distance 120 192.168.6.1 0.0.0.0 2
!
ip classless
access-list 1 permit 192.168.3.0
access-list 1 permit 192.168.4.0
access-list 2 permit 192.168.1.0
access-list 2 permit 192.168.2.0
```

该配置对 Grimwig 的路由起什么作用?

3. 在图 13-22 中, 路由器运行 OSPF, 路由器 B 的配置如下:

```
router ospf 50
 network 0.0.0.0 255.255.255.255 area 1
 distribute-list 1 in
!
access-list 1 deny 172.17.0.0
access-list 1 permit any
```

上面配置的目的是阻止路由器 B 和 C 具有关于网络 172.17.0.0 的路由条目。看上去在路

由器 B 上正在实施该计划，但是路由器 C 还是具有关于 172.17.0.0 的路由条目，为什么？

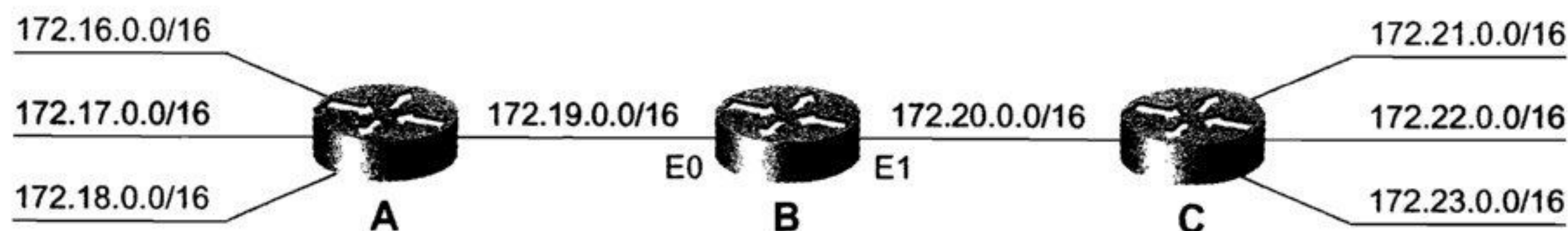


图 13-22 故障诊断练习 3 和 4 中的互联网络

4. 在图 13-22 中，路由器运行 RIP。路由器 B 的配置如下：

```
router rip
 network 172.19.0.0
 network 172.20.0.0
 distribute-list 1 out Ethernet0
 distribute-list 2 out Ethernet1
 !
 access-list 1 permit 172.18.0.0
 access-list 2 permit 172.22.0.0
```

上面配置的目的是向路由器 A 仅通告网络 172.22.0.0，向路由器 C 仅通告网络 172.18.0.0。但是，在 A 和 C 的路由表中均没有 RIP 路由条目。错误在哪里？

第 14 章

路由图

本章包括以下主题：

- 路由图的基本用途
- 配置路由图

案例研究：策略路由选择

案例研究：策略路由选择和服务质量路由

案例研究：路由图和重新分配

案例研究：路由标记

路由图与访问列表十分相似，它们都包含匹配确定报文细节的准则及许可、拒绝这些报文的操作。但是路由图不像访问列表，它可以向匹配准则中加入设置准则，设置准则可以按照指定的方式真正地对报文和路由信息进行修改。

14.1 路由图的基本用途

路由图可以用于路由重新分配和策略路由，而且还常常用在大规模边界网关协议（BGP）的运行中。虽然在前面几章对重新分配进行了广泛地讨论，但本章介绍的主题是策略路由。

策略路由只不过是复杂的静态路由。静态路由是基于报文的目的地址，将报文转发到指定的下一跳路由器，但策略路由是基于报文源地址转发报文至指定的下一跳路由器。策略路由还可以链接到扩展 IP 访问列表，以便路由器可以基于像协议类型和端口号这样的标志进行路由选择。同静态路由一样，策略路由会对路由器的路由选择产生影响，但仅限于那些配置了策略路由的路由器。

策略路由选择

图 14-1 给出了一个典型的策略路由应用的例子。AbnerNet 经路由器 Dogpatch 连接到两个 Internet 服务提供商。AbnerNet 的公司策略规定一些用户的 Internet 流量经 ISP1 发送，而其他用户流量要经 ISP2 发送。

如果其中一个 ISP 不可达，那么流经该提供商的流量将被转发给另一个提供商。在 Dogpatch 上的策略路由会根据本地策略对 Internet 流量进行分配。流量的分配可以基于子网、特殊的用户甚至是用户应用。

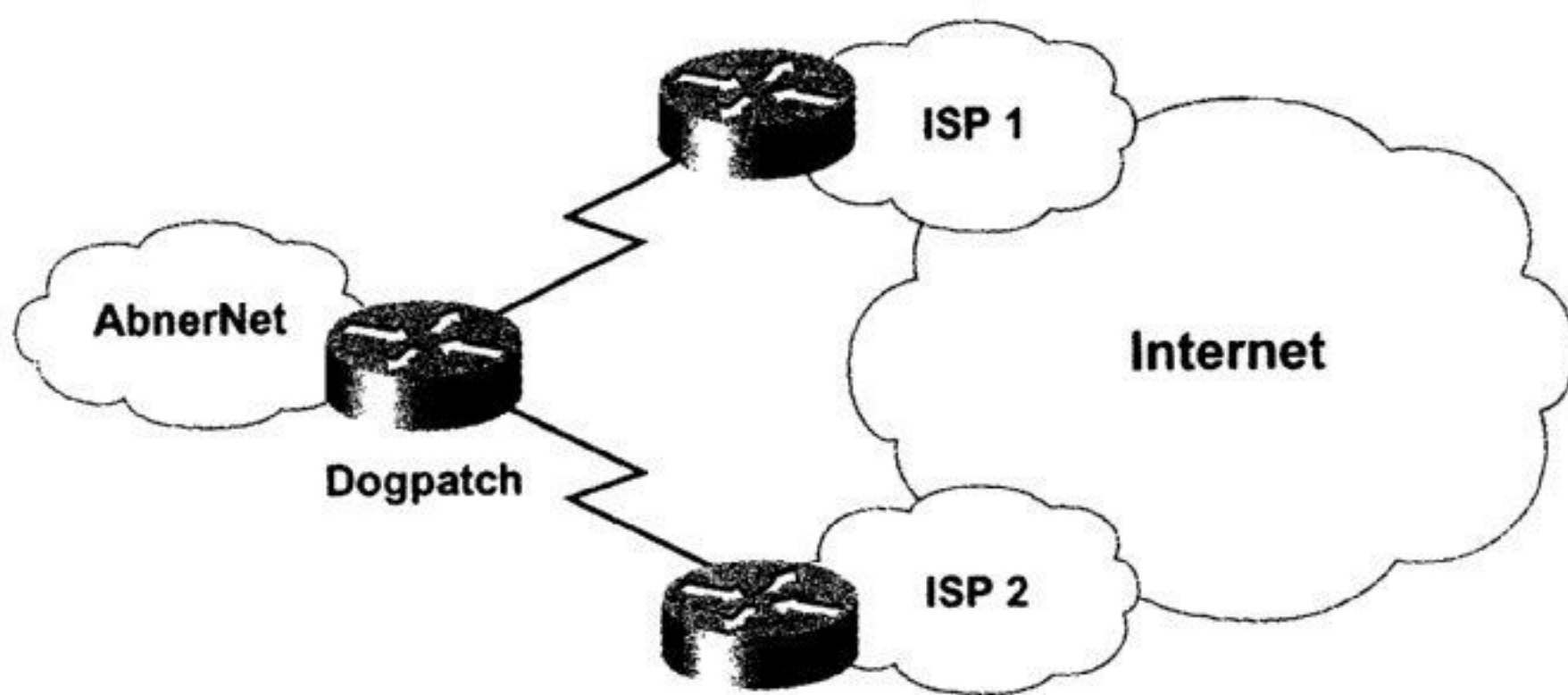


图 14-1 策略路由允许来自 AbnerNet 的流量被路由到两个 Internet 服务提供商中的一个，流量的分配可以基于诸如源地址、源/目的地址组合、报文大小、应用层端口或平均报文长度等参数

图 14-2 给出了策略路由的另一种用法。右边的一个系统监测来自行星 Mongo 的入侵军队，而另一个系统保存了过去的 Dilbert 连环漫画。我们可以配置策略路由使从 Mongo 系统到 Flash_G 的关键流量经过 FDDI 链路，而优先级较低的 Dilbert 流量经过 56K 链路。反之亦然。

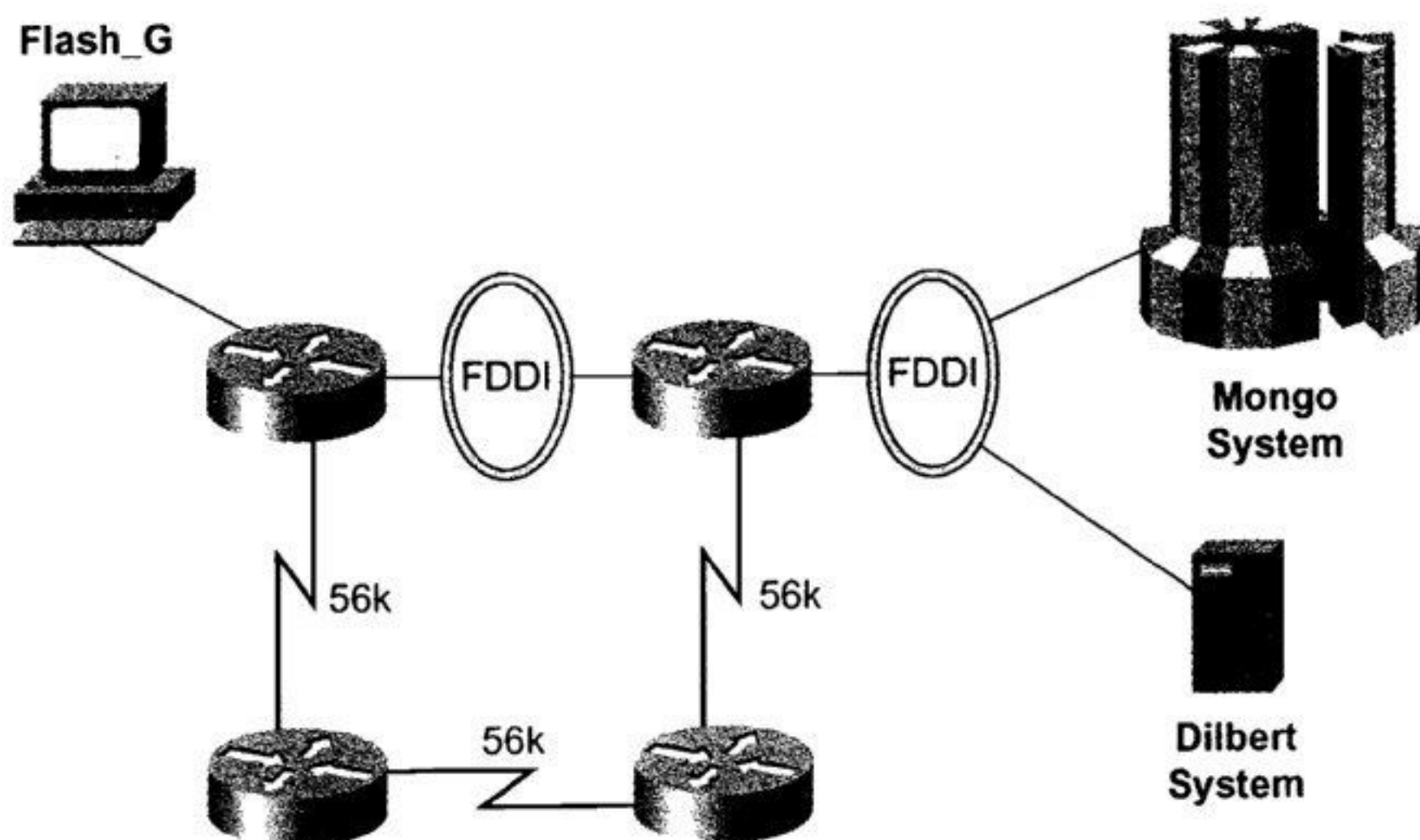


图 14-2 策略路由允许来自 Mongo 系统优先级较高的流量使用 FDDI 链路，而来自 Dilbert 系统优先级较低的流量使用 56K 链路

表 14-1 和 14-2 给出了与重新分配一起使用的命令 **match** 和 **set**，表 14-3 和 14-4 给出了与策略路由一起使用的命令 **match** 和 **set**。

表 14-1 match 命令可以与重新分配一起使用

命 令	描 述
match interface <i>type number</i> [... <i>type number</i>]	匹配被指定下一跳接口的路由
match ip address { <i>access-list-numbername</i> } [... <i>access-list-numbername</i>]	匹配被访问列表指明目标地址的路由
match ip next-hop { <i>access-list-numbername</i> } [... <i>access-list-numbername</i>]	匹配被访问列表指明下一跳路由器地址的路由
match ip route-source { <i>access-list-numbername</i> }[.. <i>access-list-numbername</i>]	匹配路由，其中通告该路由的路由器被访问列表指明
match metric <i>metric-value</i>	匹配带有指定度量的路由
match route-type { <i>internalexternal</i> [<i>type-1type-2</i>][<i>level-1level-2</i>]	匹配指定类型的 OSPF、EIGRP 或 IS-IS 路由
match tag <i>tag-value</i> [... <i>tag-value</i>]	匹配带有指定标记的路由

表 14-2 set 命令可以与重新分配一起使用

命 令	描 述
set level { <i>level-1level-2level-1-2stub-areaalbackbone</i> }	设置 IS-IS 级别或 OSPF 区域，其中匹配成功的路由将要被重新分配进入该区域
set metric { <i>metric-valuebandwidth delay reliability loading mtu</i> }	为匹配成功的路由设置度量值
set metric-type { <i>internalexternaltype-1type-2</i> }	为匹配成功的路由设置度量类型，该路由将要被重新分配进入 IS-IS 或 OSPF
set next-hop <i>next-hop</i>	为匹配成功的路由设置下一跳路由器地址
set tag <i>tag-value</i>	为匹配成功的路由设置标记值

表 14-3 match 命令可以与策略路由一起使用

命 令	描 述
match ip address { <i>access-list-numbername</i> }[.. <i>access-list-numbername</i>]	匹配带有标准或扩展访问列表中指定特征的报文
match length <i>min max</i>	匹配报文的层 3 长度

表 14-4 set 命令可以与策略路由一起使用

命 令	描 述
set default interface <i>type number</i> [... <i>type number</i>]	当不存在指向目标网络的显式路由时，为匹配成功的报文设置出站接口
set interface <i>type number</i> [... <i>type number</i>]	当存在指向目标网络的显式路由时，为匹配成功的报文设置出站接口
set ip default next-hop <i>ip-address</i> [... <i>ip-address</i>]	当不存在指向目标网络的显式路由时，为匹配成功的报文物设置下一跳路由器地址
set ip next-hop <i>ip-address</i> [... <i>ip-address</i>]	当存在指向目标网络的显式路由时，为匹配成功的报文设置下一跳路由器地址
set ip precedence <i>precedence</i>	为匹配成功的报文设置服务类型字段的优先级位
set ip tos <i>type-of-service</i>	为匹配成功的报文设置服务类型字段的 TOS 位

14.2 配置路由图

同访问列表一样（见附录 B，“教程：访问列表”），路由图本身不会对任何东西产生影响，它们必须被某些命令所调用。这些命令不是策略路由选择命令就是重新分配命令。策略路由选择将报文发送到路由图，而重新分配是将路由发送到路由图。本节的案例分析将给出在重新分配和策略路由选择中使用路由图的例子。

路由图是通过名字来标识的。例如, 下面的路由图被命名为 Hagar:

```
route-map Hagar permit 10
  match ip address 110
  set metric 100
```

每条路由图表述都包含“许可”、“拒绝”操作及一个序列号。这个路由图给出了一个许可操作和序列号 10。这些设置都是缺省的——也就是在配置路由图时如果没有指定操作或序列号, 那么路由图的操作和序列号缺省值就是序列号 10。

序列号允许说明和编辑多个表述, 考虑下面的配置步骤:

```
Linus(config)#route-map Hagar 20
Linus(config-route-map)#match ip address 111
Linus(config-route-map)#set metric 50
Linus(config-route-map)#route-map Hagar 15
Linus(config-route-map)#match ip address 112
Linus(config-route-map)#set metric 80
```

这里, 向路由图 Hagar 添加了第二组和第三组路由图表述, 其中每组表述都包含自己的 **match** 和 **set** 设置表述。注意, 第一次配置的序列号是 20, 第二次配置的序列号是 15。在最终的配置文件中, IOS 将把表述 15 放在表述 20 之前, 尽管表述 15 是在后面被输入的:¹

```
route-map Hagar permit 10
  match ip address 110
  set metric 100
!
route-map Hagar permit 15
  match ip address 112
  set metric 80
!
route-map Hagar permit 20
  match ip address 111
  set metric 50
```

序列号还允许删除个别表述。例如, 表述:

```
Linus(config)#no route-map Hagar 15
```

删除了表述 15, 而其他表述被完整无缺地保留下来:

```
route-map Hagar permit 10
  match ip address 110
  set metric 100
!
route-map Hagar permit 20
  match ip address 111
  set metric 50
```

在编辑路由图时必须谨慎。在这个例子中, 如果输入了 **no route-map Hagar**, 而没有指明序列号, 那么整个路由图将会被删除。同样, 如果在添加 **match** 和 **set** 表述时没有指定序

¹ 还要注意一点, 在配置中没有指定操作, 所以缺省操作“许可”出现在最终的配置中。

列号, 那么它们仅会修改表述 10。

报文和路由是顺序地通过路由图表述的。如果匹配成功, 那么将执行所有 **set** 表述及许可或拒绝操作。如同使用访问列表一样, 当匹配发生时, 处理马上停止, 指定的操作将被执行。路由或报文不再经过后继表述, 考虑下面的路由图:

```
route-map Sluggo permit 10
  match ip route-source 1
  set next-hop 192.168.1.5
!
route-map Sluggo permit 20
  match ip route-source 2
  set next-hop 192.168.1.10
!
route-map Sluggo permit 30
  match ip route-source 3
  set next-hop 192.168.1.15
```

如果路由不能匹配到表述 10, 那么它将被传递到表述 20。如果在表述 20 匹配发生, 那么将执行设置命令, 并且路由被允许。匹配成功的路由将不会被传递给表述 30。

拒绝操作的行为依赖于路由图被用于策略路由还是重新分配。如果用于重新分配并且与路由匹配的表述操作为拒绝, 那么路由将不会被重新分配。如果用于策略路由选择并且与报文匹配的表述操作为拒绝, 那么将不会为报文进行策略路由选择, 但是报文会被传递给常规路由进行转发。

与访问列表一样, 如果报文或路由没有匹配到任何一个表述, 那么对路由图来说必须执行一个缺省操作。在每个路由图的结尾都隐含了一个拒绝操作。如果路由经过重新分配路由图而没有发生匹配, 那么路由将不被重新分配。如果报文经过策略路由图而没有发生匹配, 那么报文将被发送到常规路由进程。

如果在路由图表述中没有配置 **match** 表述, 那么缺省操作是匹配所有报文和路由。

每个路由图表述可能有多个 **match** 和 **set** 表述, 如下面配置所示:

```
route-map Garfield permit 10
  match ip route-source 15
  match interface Serial0
  set metric-type type-1
  set next-hop 10.1.2.3
```

在这个案例中为了执行 **set** 表述, 在每个 **match** 表述中都必须发生匹配。

14.2.1 案例研究: 策略路由选择

使用命令 **ip policy route-map** 可以定义策略路由选择。该命令是在接口上进行配置, 它仅会对入站报文有影响。

在图 14-3 中, 假设在 Linus 上实现策略, 以便来自 172.16.6.0/24 的业务被转发到 Lucy, 而来自 172.16.7.0/24 的流量被转发到 Pigpen。Linus 的相应配置如下:

```
interface Serial0
```



```

ip address 172.16.5.1 255.255.255.0
ip policy route-map Sally
!
access-list 1 permit 172.16.6.0 0.0.0.255
access-list 2 permit 172.16.7.0 0.0.0.255
!
route-map Sally permit 10
match ip address 1
set ip next-hop 172.16.4.2
!
route-map Sally permit 15
match ip address 2
set ip next-hop 172.16.4.3

```

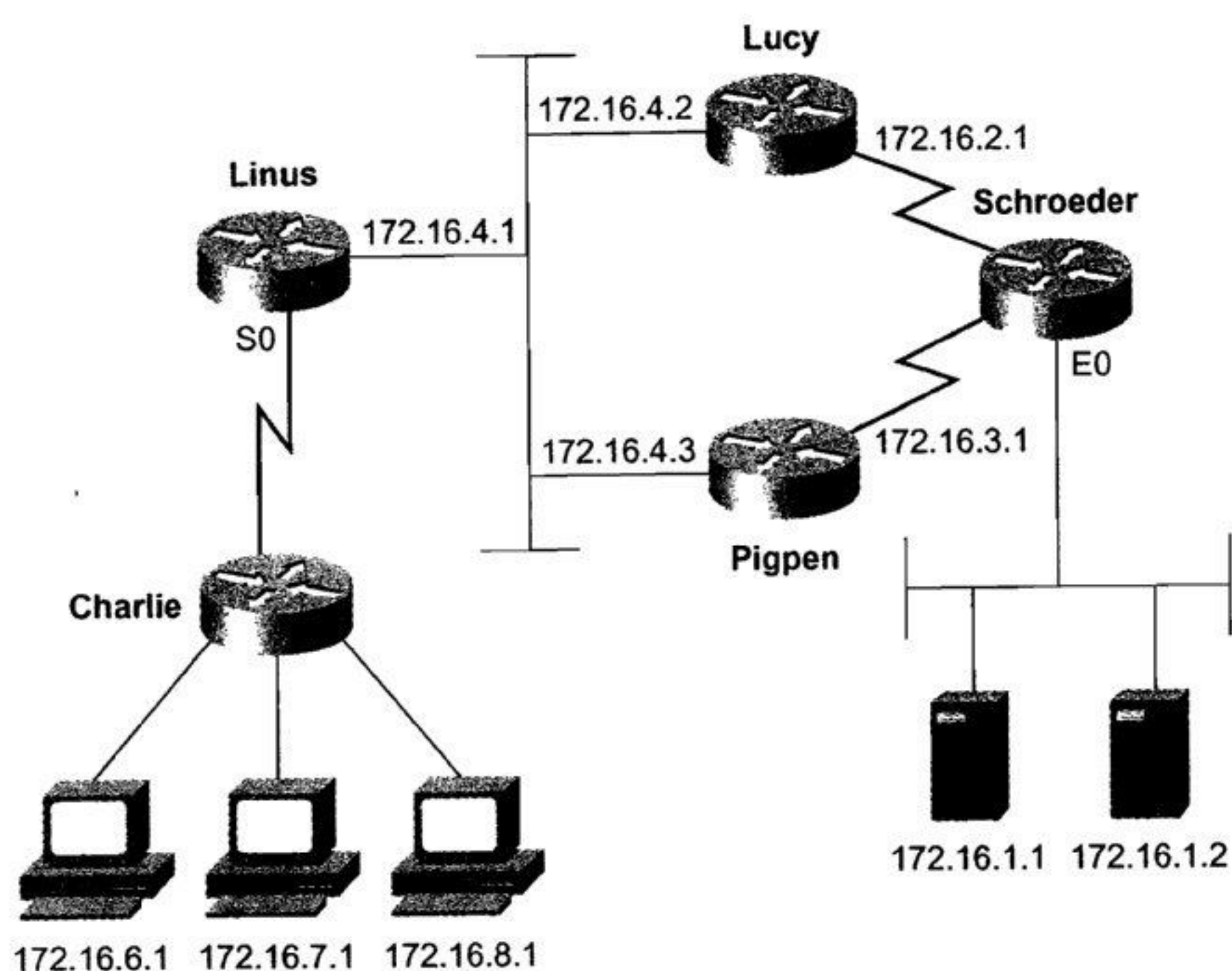


图 14-3 在 Linus 上配置策略路由，使某些报文经过 Lucy，而其他报文经过 Pigpen

S0 上的策略路由选择命令把入站报文发送给路由图 Sally。路由图 Sally 的表述 10 使用访问列表 1 标识来自子网 172.16.6.0/24 的源地址。如果匹配成功，报文将被转发到 Lucy，其中报文的下一跳接口地址是 172.16.4.2。如果匹配不成功，报文被发送给表述 15。该表述使用访问列表 2 匹配来自子网 172.16.7.0/24 的源地址。如果匹配成功，报文被转发给 Pigpen (172.16.4.3)。任何没有匹配到表述 15 的报文，例如来自子网 172.16.8.0/24 的报文，将会被正常地路由。图 14-4 给出了策略路由的结果。¹

当仅按照源地址进行策略路由选择时可以使用标准 IP 访问列表。如果需要借助源地址和目标地址进行路由，那么要使用扩展访问列表。如果报文从任意子网到主机 172.16.1.1，那么下面的配置将把报文转发到 Lucy，而从主机 172.16.7.2 到 172.16.1.2 的报文将被转发给 Pigpen。其他所有报文被正常地路由。

¹ 注意命令 `debug ip packet` 参考了访问列表 5。为了使调试功能对不感兴趣的流量不作显示，该访问列表仅允许连接在路由器 Charlie 的子网。


```

Linus#debug ip packet 5
IP packet debugging is on for access list 5
Linus#
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward

```

图 14-4 配置在 Linus 接口 S0 上的策略路由将来自子网 172.16.6.0/24 的报文路由到 Lucy (172.16.4.2)，将来自子网 172.16.7.0/24 的报文路由到 Pigpen (172.16.4.3)。来自 172.16.8.0/24 的报文因没有匹配到策略路由，所以被正常地路由（在 Lucy 和 Pigpen 之间进行负载均衡）

```

interface Serial0
 ip address 172.16.5.1 255.255.255.0
 ip policy route-map Sally
!
access-list 101 permit ip any host 172.16.1.1
access-list 102 permit ip host 172.16.7.1 host 172.16.1.2
!
route-map Sally permit 10
 match ip address 101
 set ip next-hop 172.16.4.2
!
route-map Sally permit 15
 match ip address 102
 set ip next-hop 172.16.4.3

```

这里再次使用路由图 Sally，除了 **match** 表述现在是参照访问列表 101 和 102。图 14-5 给出了结果。

```

Linus#debug ip packet 5
IP packet debugging is on for access list 5
Linus#
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.2, len 60, forward

```

图 14-5 从主机 172.16.7.1 到主机 172.16.1.1 的报文匹配到路由图 Sally 中的表述 10，报文被转发给 Lucy。从相同主机到主机 172.16.1.2 的报文被转发到 Pigpen。从子网 172.16.7.0/24 上的另一个地址到主机 172.16.1.2 的报文没有被 Sally 匹配到，因而被正常地路由

其次,假设你的策略规定从子网 172.16.1.0/24 上的服务器发出的 FTP 流量被转发到 Lucy,从相同服务器发出的 Telnet 流量将被转发到 Pigpen。这个计划使大容量 FTP 流量和突发、活跃的 Telnet 在 Schroeder 处被分离到两条串行链路上。Schroeder 的相应配置如下:

```
interface Ethernet0
 ip address 172.16.1.4 255.255.255.0
 ip policy route-map Rerun
!
access-list 105 permit tcp 172.16.1.0 0.0.0.255 eq ftp any
access-list 105 permit tcp 172.16.1.0 0.0.0.255 eq ftp-data any
access-list 106 permit tcp 172.16.1.0 0.0.0.255 eq telnet any
!
route-map Rerun permit 10
 match ip address 105
 set ip next-hop 172.16.2.1
!
route-map Rerun permit 20
 match ip address 106
 set ip next-hop 172.16.3.1
```

访问列表 105 和 106 不仅检查源地址和目的地址,而且还检查源端口。在图 14-6 中,选项 **detail** 与 **debug ip packet** 一起使用,可以观察到被 Schroeder 转发的报文类型。访问列表 10 对所显示的报文作出了限制,仅允许从 172.16.1.1 到 172.16.6.1 的报文被显示。

```
Schroeder#debug ip packet detail 10
IP packet debugging is on (detailed) for access list 10
Schroeder#
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1064, forward
  TCP src=20, dst=1047, seq=3702770065, ack=591246297, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 64, forward
  TCP src=21, dst=1046, seq=3662108731, ack=591205663, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
  TCP src=20, dst=1047, seq=3702771089, ack=591246297, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
  TCP src=23, dst=1048, seq=3734385279, ack=591277873, win=14332 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 52, forward
  TCP src=23, dst=1048, seq=3734385279, ack=591277873, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
  TCP src=23, dst=1048, seq=3734385291, ack=591277876, win=14332 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 60, forward
  ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
  ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 60, forward
  ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
  ICMP type=0, code=0
```

图 14-6 FTP 报文 (TCP 端口 20 和 21) 被转发到 Lucy, 而源地址和目的地地址相同的 Telnet 报文 (TCP 端口 23) 被转发到 Pigpen。

回应应答报文 (ICMP 类型 0) 在策略路由中没有找到匹配表述, 将被正常地路由

分割批量和交互式流量的目的是使小报文特性的交互式流量不会被大报文特性的批量流量所延迟, 最后一个例子也是这样的。在本例中该方法的缺点是, 如果需要隔离的流量类型很多, 那么通过目标端口标识流量会使访问列表变得惊人的庞大。

如果目标是将小报文从大报文中分离出来，那么可以对报文长度进行匹配：

```
interface Ethernet0
 ip address 172.16.1.4 255.255.255.0
 ip policy route-map Woodstock
!
route-map Woodstock permit 20
 match length 1000 1600
 set ip next-hop 172.16.2.1
!
route-map Woodstock permit 30
 match length 0 400
 set ip next-hop 172.16.3.1
```

在这里，**match length** 表述指明了报文长度的最小值和最大值。路由图的表述 20 使所有长度在 1000~1600 个 8bit 字节的报文经过串行链路到达 Lucy。表述 30 使所有长度大于 400 个 8bit 字节的报文经过串行链路到达 Pigpen。长度在 400~1000 个 8bit 字节的报文则被正常地路由。

图 14-7 给出了新路由图的结果。从 172.16.1.2 到 172.16.6.1 的 FTP、Telnet 和回应应答报文现在将根据它们的大小被路由，而不是按照报文的地址和端口。

```
Schroeder#debug ip packet detail 10
IP packet debugging is on (detailed) for access list 10
Schroeder#
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
  TCP src=20, dst=1063, seq=1528444161, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
  TCP src=20, dst=1063, seq=1528442725, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
  TCP src=20, dst=1063, seq=1528444161, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 840, forward
  TCP src=20, dst=1063, seq=1528445597, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
  TCP src=21, dst=1062, seq=1469372904, ack=601897901, win=14329 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 54, forward
  TCP src=21, dst=1062, seq=1469372904, ack=601897901, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
  TCP src=21, dst=1062, seq=1469372918, ack=601897901, win=14335 ACK FIN
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 44, forward
  TCP src=23, dst=1064, seq=1712116521, ack=602140570, win=14335 ACK SYN
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 43, forward
  TCP src=23, dst=1064, seq=1712116522, ack=602140570, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
  TCP src=23, dst=1064, seq=1712116525, ack=602140573, win=14332 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 52, forward
  TCP src=23, dst=1064, seq=1712116525, ack=602140573, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
  ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
  ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
  ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
  ICMP type=0, code=0
```

图 14-7 长度大于等于 1000 个 8bit 字节的报文被路由到 Lucy，而长度小于等于 400 个 8bit 字节的报文被路由到 Pigpen。
所有长度在 400~1000 个 8bit 字节的报文将被正常地路由

到目前为止, 所示例的策略路由都是对从一个特殊接口进入路由器的报文产生影响, 但是路由器自己产生的报文又会怎样? 使用命令 **ip local policy route-map** 可以使这些报文也按策略进行路由。命令 **ip policy route-map** 是配置在接口上的, 与此命令不同的是, 命令 **ip local policy route-map** 全局地配置在路由器上。

为了在 Schroeder 产生的报文上应用前面示范的策略, 相应的配置如下:

```
interface Ethernet0
  ip address 172.16.1.4 255.255.255.0
  ip policy route-map Woodstock
!
ip local policy route-map Woodstock
!
access-list 120 permit ip any 172.16.1.0 0.0.0.255
access-list 120 permit ospf any any
!
route-map Woodstock permit 10
  match ip address 120
!
route-map Woodstock permit 20
  match length 1000 1600
  set ip next-hop 172.16.2.1
!
route-map Woodstock permit 30
  match length 0 400
  set ip next-hop 172.16.3.1
```

这里特别感兴趣的是表述 10, 该表述没有 **set** 表述, 而是仅允许匹配到访问列表 120 的报文。访问列表 120 依次允许 OSPF 报文和所有到子网 172.16.1.0/24 的报文。如果没有访问列表的第一行, 那么由 Schroeder 产出的报文和到子网 172.16.1.0/24 的报文将会被表述 20 和 30 转发到错误的接口。图 14-8 说明了为什么有必要配置第二行。Schroeder 的 OSPF Hello 报

Number	Delta time	Interpretation
26	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
32	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
39	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
44	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
52	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
57	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
61	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
67	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
74	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
79	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
87	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
92	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
96	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
102	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
109	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
114	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
122	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
127	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
133	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
139	10.0 sec	Hello Len=44 Router_ID=172.16.3.2
146	10.0 sec	Hello Len=44 Router_ID=172.16.3.2

图 14-8 可以在分析器中看到 OSPF Hello 报文的长度

文长度为 44 个 8bit 字节，如果没有包括表述 10，那么 OSPF Hello 报文将匹配到表述 30 并被转发到 Pigpen，这将切断 Lucy 与 Schroeder 之间的邻接关系。如果匹配到表述 10，OSPF 报文将被允许并被正常地转发，邻接关系不会发生改变。

14.2.2 案例研究：策略路由选择和服务质量路由

虽然服务质量（QoS）路由超出了本卷的范围，但是这里必须注意，策略路由选择可以是 QoS 的一个完整的部分。带有 QoS 的策略路由选择可以在报文进入路由器接口时，通过设置报文 IP 头内字段中的优先级和服务类型位来实现。图 14-9 给出了 TOS 字段的位信息。虽然在现代互联网络中很少使用 TOS 位，但是在 QoS 应用中优先级位焕发出新的生命力。TOS 位可用于改变路由器为报文选择的路由。而优先级位则用于区分路由器中报文的优先次序。

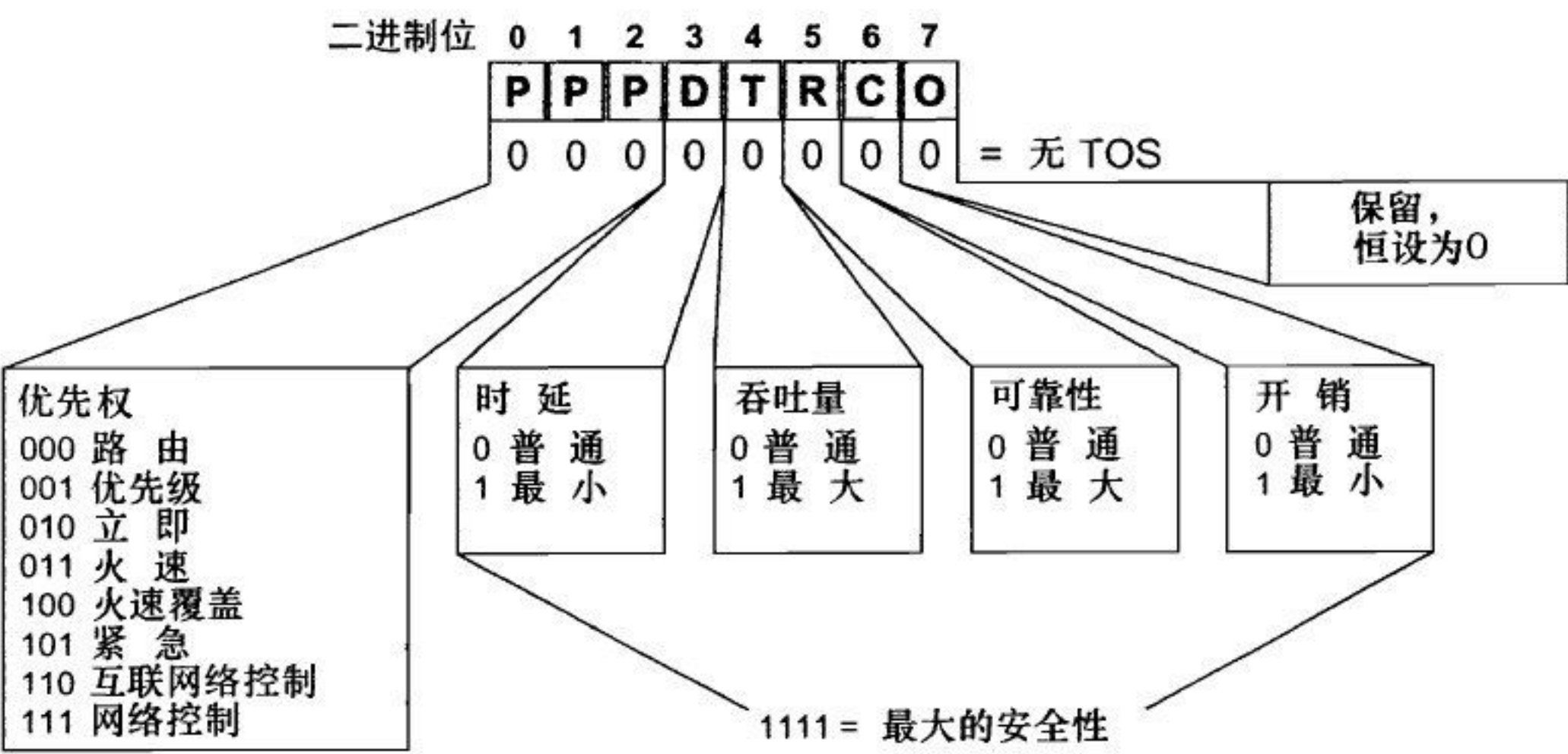


图 14-9 IP 报头服务类型字段中的优先级和 TOS 位

在路由图中使用 **set ip precedence** 表述可以设置优先级位。可以通过指定 3 个优先级位的十进制等价数或关键字来设置优先级。表 14-5 给出了将会用到的十进制数和关键字。

表 14-5 命令 set ip precedence 使用到的优先级值和关键字

比 特	数 字	关 键 字
000	0	路由
001	1	优先级
010	2	立即
011	3	火速
100	4	火速—覆盖
101	5	紧急
110	6	互联网络
111	7	网络

TOS 位可以通过表述 **set ip tos** 来设置，同优先级表述一样，表述的参数可以是数字和关键字，如表 14-6 所示。与优先级不同的是，你可以使用组合 TOS 值。例如，指定 TOS 为 12 (1100b) 表明最小带宽和最大吞吐量。由于仅能使用一个关键字，因此为了设置组合 TOS 值，

必须使用数字。

表 14-6

命令 set ip tos 用到的 TOS 值和关键字

比特	数字 (0-15)	关键字
0000	0	正常
0001	1	最小金钱的花费
0010	2	最大可靠性
0100	4	最大吞吐量
1000	8	最小时延

图 14-10 给出了在 QoS 路由选择中怎样使用策略路由的例子。这里，路由器 Pogo 在互联网 OkefenokeeNet 的边界。通过在 Pogo 的串行链路上配置策略路由，可以改变入站报文的优先级位或 TOS 位以便将 IP 流量区别为几种流量类型。例如：

```

interface Serial0
 ip address 10.1.18.67 255.255.255.252
 ip policy route-map Albert
!
interface Serial1
 ip address 10.34.16.83 255.255.255.252
 ip policy route-map Albert
!
access-list 1 permit 172.16.0.0 0.0.255.255
access-list 110 permit tcp any eq www any
!
route-map Albert permit 10
 match ip address 1 110
 set ip precedence critical
!
route-map Albert permit 20
 set ip tos 10
 set ip precedence priority

```

表述 10 说明如果报文匹配到访问列表 1 和 110，那么优先级被设置为紧急。注意表述 20 没有 **match** 表述，那么该表述将匹配所有在表述 10 没有发生匹配的报文。在表述 20 中还有两个 **set** 表述，这些表述将设置 TOS 位为最小延迟和最大可靠性，并且设置优先级为优先。图 14-11 给出了在 OkefenokeeNet 中捕获到的报文，该报文被 Pogo 上的路由图修改过。

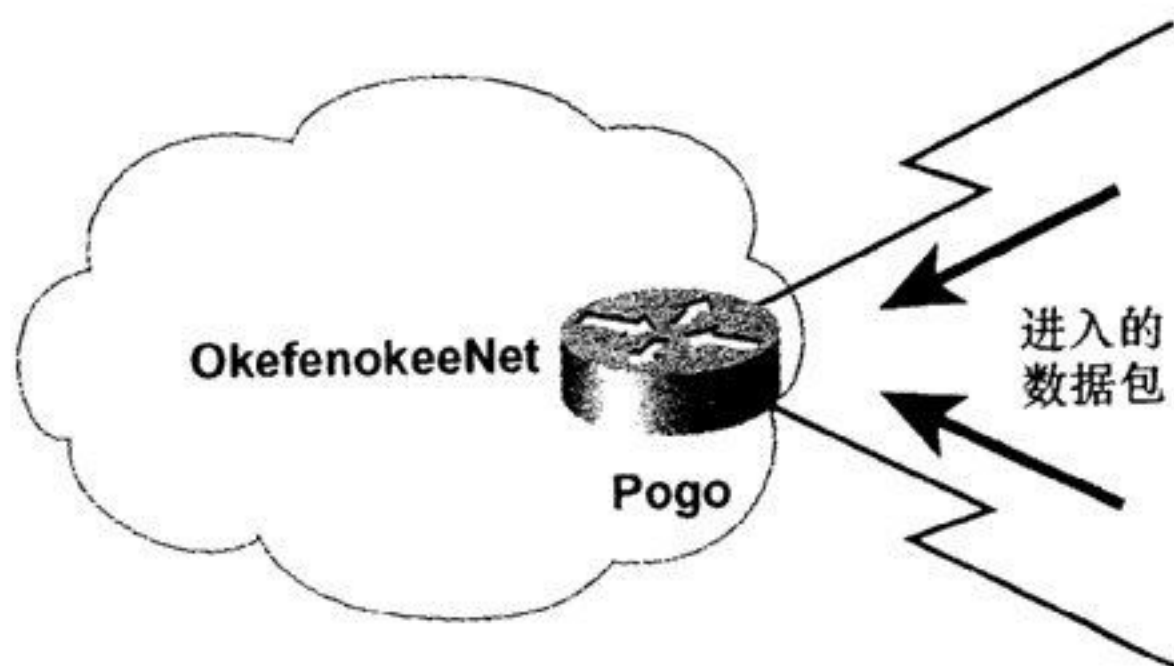


图 14-10 策略路由可以用于设置进入互联网报文的优先级位和 TOS 位，
然后，互联网中的路由器将基于对这些位的设置进行 QoS 决策

在设置好进入互连网络报文的优先级和 TOS 位之后,互连网络内的路由器将基于这些位所定义的部分或全部服务类别进行 QoS 决策。例如,为了对流量划分优先等级,可以根据优先级和 TOS 位配置优先级、自定义或加权公平队列。在某些实现中,优先级被用于拥塞控制机制,例如加权随机早期检测 (Weighted Random Early Detection, WRED)。或者通过配置访问列表,使其可以基于优先级或 TOS 位允许和拒绝报文经过某条链路,以便实现粗服务类别路由选择 (Class of Service, CoS)。

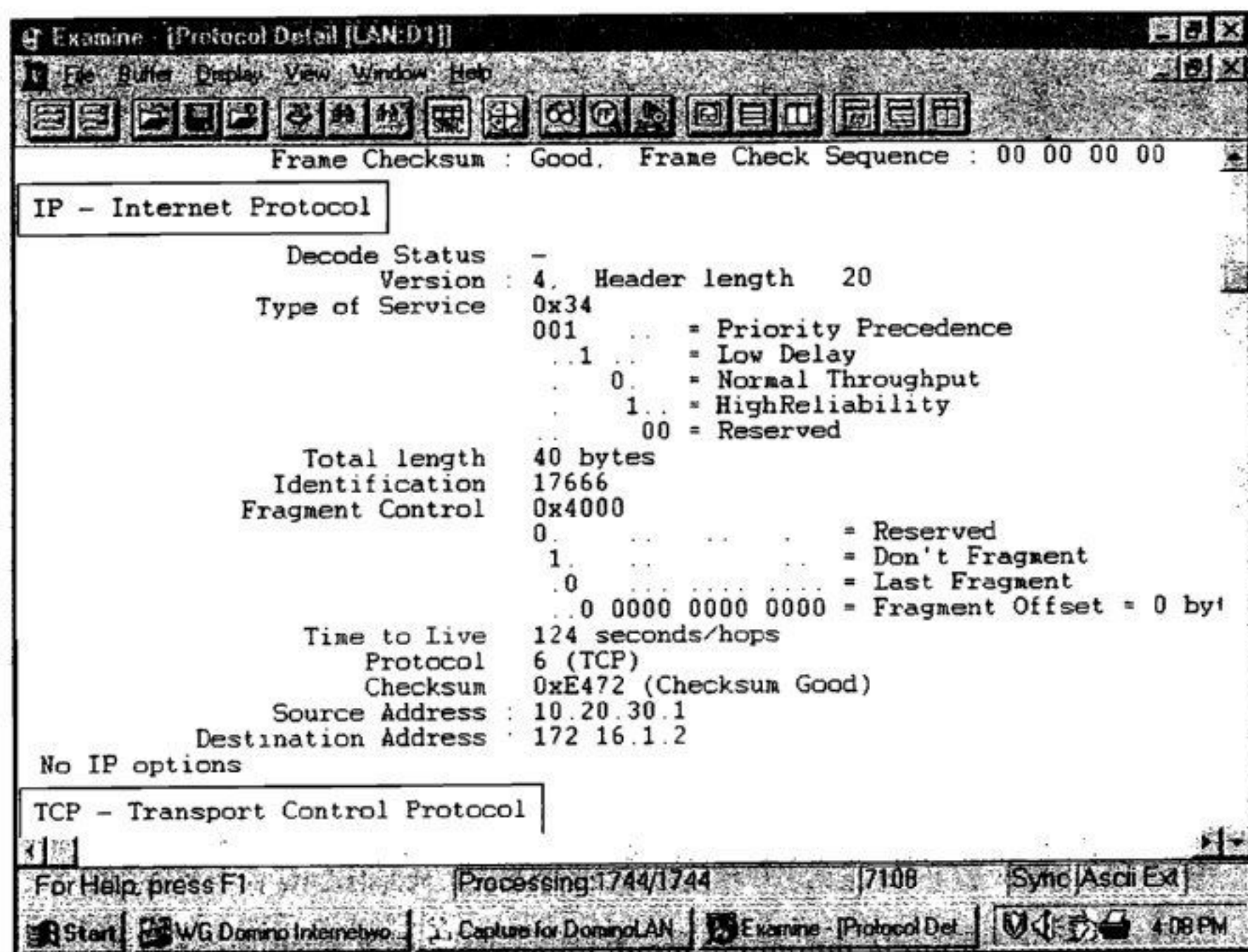


图 14-11 Pogo 的策略路由将这个报文的优先级位设置为 001b, TOS 位设置为最小延迟和最大可靠性 (1010b)

14.2.3 案例研究: 路由图和重新分配

通过在 **redistribute** 命令中添加对路由图的调用就可以使路由图和重新分配一起使用。在图 14-12 给出的互连网络中,在路由器 Zippy 上,IS-IS 和 OSPF 路由正在进行相互的路由重新分配。例子中列出了网络和子网的地址,其中仅第 3 个 8bit 字节为奇数的子网被重新分配。Zippy 的配置如下:

```
router ospf 1
 redistribute isis level-1 metric 20 subnets route-map Griffy
 network 172.16.10.2 0.0.0.0 area 5
!
router isis
 redistribute ospf 1 metric 25 route-map Toad metric-type internal level-2
 net 47.0001.1234.5678.9056.00
!
access-list 1 permit 192.168.2.0
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.6.0
access-list 2 permit 172.16.1.0
```



```

access-list 2 permit 172.16.3.0
access-list 2 permit 172.16.5.0
access-list 2 permit 172.16.7.0
access-list 2 permit 172.16.9.0
!
route-map Griffy deny 10
  match ip address 1
!
route-map Griffy permit 20
!
route-map Toad permit 10
  match ip address 2

```

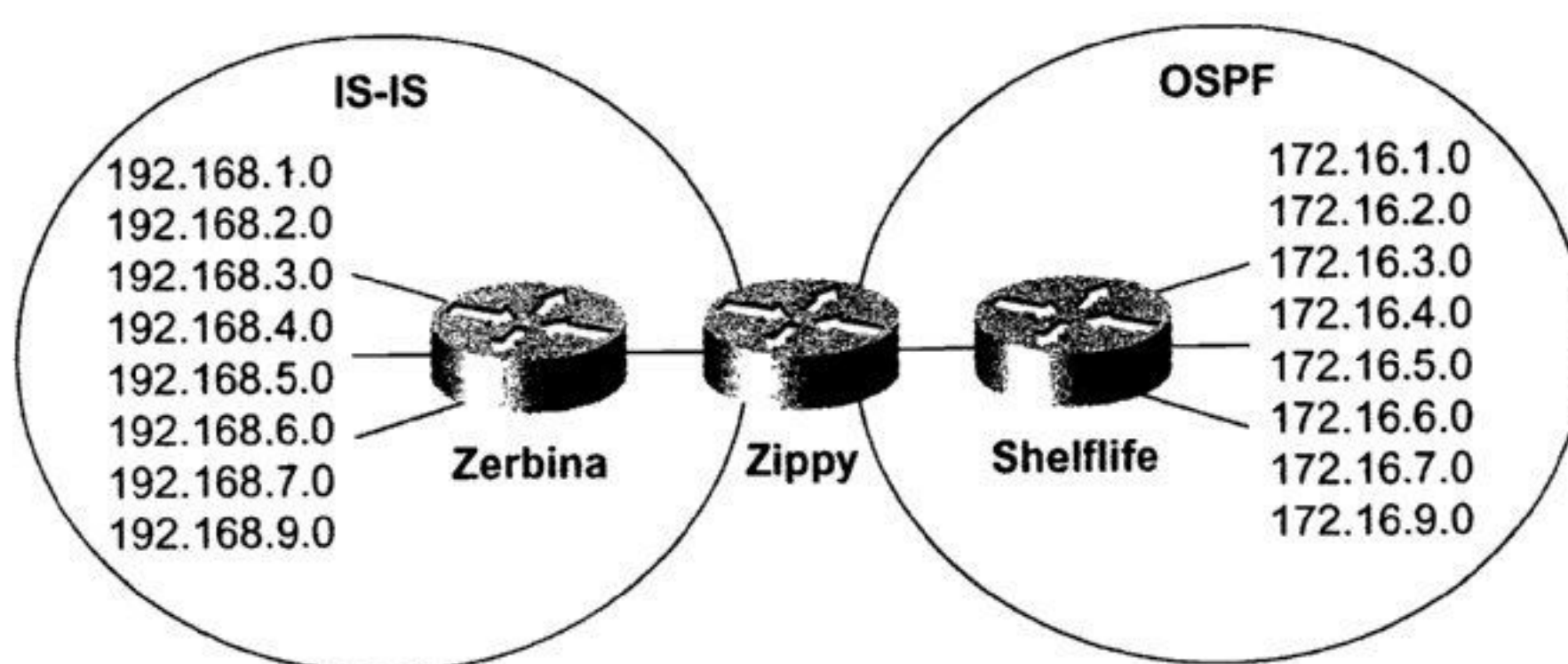


图 14-12 OSPF 和 IS-IS 正在进行相互的路由重新分配。**redistribute** 命令和路由图一起被用作简单的路由过滤，或者说路由图可以用于修改被重新分配路由的属性

路由图 Griffy 和 Toad 执行相同的功能，但是它们的逻辑却不相同。Griffy 使用否定逻辑，它标识那些不被重新分配的路由，而 Toad 使用肯定逻辑，它标识将要被重新分配的路由。

Griffy 的表述 10 拒绝访问列表 1 许可的任何路由（第 3 个 8bit 字节为偶数的地址）。因为第 3 个 8bit 字节为奇数的地址不能匹配到表述 10，因此它们被传递到表述 20。表述 20 没有 **match** 命令，因而缺省匹配所有地址。表述 20 具有允许操作，所以奇数地址的路由被许可。结果如图 14-13 所示。

```

Shelflife#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
       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

Gateway of last resort is not set

O E2 192.168.9.0 [110/20] via 172.16.10.2, 00:24:46, Ethernet0
O E2 192.168.1.0 [110/20] via 172.16.10.2, 00:24:46, Ethernet0
O E2 192.168.3.0 [110/20] via 172.16.10.2, 00:24:46, Ethernet0
O E2 192.168.5.0 [110/20] via 172.16.10.2, 00:24:47, Ethernet0
O E2 192.168.7.0 [110/20] via 172.16.10.2, 00:24:47, Ethernet0
    172.16.0.0 255.255.255.0 is subnetted, 9 subnets
C      172.16.9.0 is directly connected, Serial0

```

待续


```

C    172.16.10.0 is directly connected, Ethernet0
O    172.16.4.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O    172.16.5.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O    172.16.6.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O    172.16.7.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O    172.16.1.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O    172.16.2.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O    172.16.3.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
Shelflife#

```

图 14-13 对于那些在 IS-IS 域内的目标网络，仅那些第 3 个 8bit 字节为奇数的网络被包含在 Shelflife 的路由选择表中

路由图有一条单一表述允许访问列表 2 许可的路由（第 3 个 8bit 字节为奇数）。第 3 个 8bit 字节为偶数的地址不能在访问列表 2 中找到匹配。当重新分配时，缺省的路由图表述是拒绝所有路由，所以不能被访问列表 2 匹配的地址将不会被重新分配。图 14-14 给出了路由图 Toad 的结果。

```

Zerbina#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

C    192.168.9.0/24 is directly connected, Serial0
C    192.168.10.0/24 is directly connected, Ethernet0
i L1 192.168.1.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.2.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.3.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.4.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.5.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.6.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.7.0/24 [115/15] via 192.168.9.2, Serial0
     172.16.0.0/24 is subnetted, 5 subnets
i L2   172.16.9.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.5.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.7.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.1.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.3.0 [115/35] via 192.168.10.2, Ethernet0
Zerbina#

```

图 14-14 对于那些在 OSPF 域内的目标网络，仅那些第 3 个 8bit 字节为奇数的网络被包含在 Zerbina 的路由选择表中

另一个配置将实现相同的目的，例如路由图 Toad 使用下面访问列表将起到同样的作用：

```

access-list 2 deny 172.16.2.0
access-list 2 deny 172.16.4.0
access-list 2 deny 172.16.6.0
access-list 2 permit any

```

虽然路由图可以像简单的路由过滤一样工作得很好，但是它的能力体现在可以按照多种

方式改变路由。考虑图 14-12 中 Zippy 的配置:

```

router ospf 1
 redistribute isis level-1 metric 20 subnets route-map Griffy
 network 172.16.10.2 0.0.0.0 area 5
!
router isis
 redistribute ospf 1 metric 25 route-map Toad metric-type internal level-2
 net 47.0001.1234.5678.9056.00
!
ip classless
access-list 1 permit 192.168.2.0
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.6.0
access-list 2 permit 172.16.9.0
access-list 2 permit 172.16.5.0
access-list 2 permit 172.16.7.0
access-list 2 permit 172.16.1.0
access-list 2 permit 172.16.3.0
!
route-map Griffy permit 10
 match ip address 1
 set metric-type type-1
!
route-map Griffy permit 20
!
route-map Toad permit 10
 match ip address 2
 set metric 15
 set level level-1
!
route-map Toad permit 20

```

路由图 Griffy 的表述 10 允许在访问列表 1 中的地址, 并且将它们作为外部 1 型路由向 OSPF 重新分配。表述 20 允许所有其他路由, 并把它作为外部 2 型路由重新分配。图 14-15 给出了结果。

指向访问列表 2 所定义的地址的路由, 在路由图 Toad 中是被表述 10 允许的, 并且这些路由被作为 1 级路由重新分配到 IS-IS, 度量为 15。表述 20 允许所有其他路由, 在 IS-IS 配置下的 **redistribute** 命令把这些路由作为 2 级路由重新分配, 度量为 25 (图 14-16)。

```

Shelflife#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
       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

Gateway of last resort is not set

O E2 192.168.9.0 [110/20] via 172.16.10.2, 00:13:43, Ethernet0
O E2 192.168.1.0 [110/20] via 172.16.10.2, 00:13:43, Ethernet0
O E1 192.168.2.0 [110/30] via 172.16.10.2, 00:13:43, Ethernet0

```

待续


```

0 E2 192.168.3.0 [110/20] via 172.16.10.2, 00:13:44, Ethernet0
0 E1 192.168.4.0 [110/30] via 172.16.10.2, 00:13:44, Ethernet0
0 E2 192.168.5.0 [110/20] via 172.16.10.2, 00:13:44, Ethernet0
0 E1 192.168.6.0 [110/30] via 172.16.10.2, 00:13:44, Ethernet0
0 E2 192.168.7.0 [110/20] via 172.16.10.2, 00:13:44, Ethernet0
    172.16.0.0 255.255.255.0 is subnetted, 9 subnets
C      172.16.9.0 is directly connected, Serial0
C      172.16.10.0 is directly connected, Ethernet0
0      172.16.4.0 [110/159] via 172.16.9.2, 15:49:29, Serial0
0      172.16.5.0 [110/159] via 172.16.9.2, 15:49:30, Serial0
0      172.16.6.0 [110/159] via 172.16.9.2, 15:49:30, Serial0
0      172.16.7.0 [110/159] via 172.16.9.2, 15:49:30, Serial0
0      172.16.1.0 [110/159] via 172.16.9.2, 15:49:30, Serial0
0      172.16.2.0 [110/159] via 172.16.9.2, 15:49:30, Serial0
0      172.16.3.0 [110/159] via 172.16.9.2, 15:49:30, Serial0
Shelflife#

```

图 14-15 如果路由的目标网络在 IS-IS 域且地址的第 3 个 8bit 字节为奇数，则路由为 E1，如果为偶数，则为 E2

```

Zerbina#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

C      192.168.9.0/24 is directly connected, Serial0
C      192.168.10.0/24 is directly connected, Ethernet0
i L1 192.168.1.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.2.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.3.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.4.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.5.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.6.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.7.0/24 [115/15] via 192.168.9.2, Serial0
    172.16.0.0/24 is subnetted, 8 subnets
i L1   172.16.9.0 [115/25] via 192.168.10.2, Ethernet0
i L2   172.16.4.0 [115/35] via 192.168.10.2, Ethernet0
i L1   172.16.5.0 [115/25] via 192.168.10.2, Ethernet0
i L2   172.16.6.0 [115/35] via 192.168.10.2, Ethernet0
i L1   172.16.7.0 [115/25] via 192.168.10.2, Ethernet0
i L1   172.16.1.0 [115/25] via 192.168.10.2, Ethernet0
i L2   172.16.2.0 [115/35] via 192.168.10.2, Ethernet0
i L1   172.16.3.0 [115/25] via 192.168.10.2, Ethernet0
Zerbina#

```

图 14-16 如果路由的目标网络在 OSPF 域且地址第 3 个 8bit 字节为奇数，则路由为 L2，如果为偶数，则为 L1。

被重新分配的奇数路由度量为 15，偶数为 25（加 10 是因为从 Zippy 到 Zebbina 的跳数为 10）

14.2.4 案例研究：路由标记

图 14-17 表明，来自多个路由选择域的路由被重新分配到一个运行 OSPF 的传输域，其

中每个域都运行单独的路由选择协议。在 OSPF 云的另一边, 路由必须被重新分配回到它们各自的域。在从 OSPF 云进入每个域的出口点, 可以使用路由过滤仅允许属于该域的路由通过。然而, 如果每个域的路由很多或变动很大, 那么路由过滤可能会很难管理。

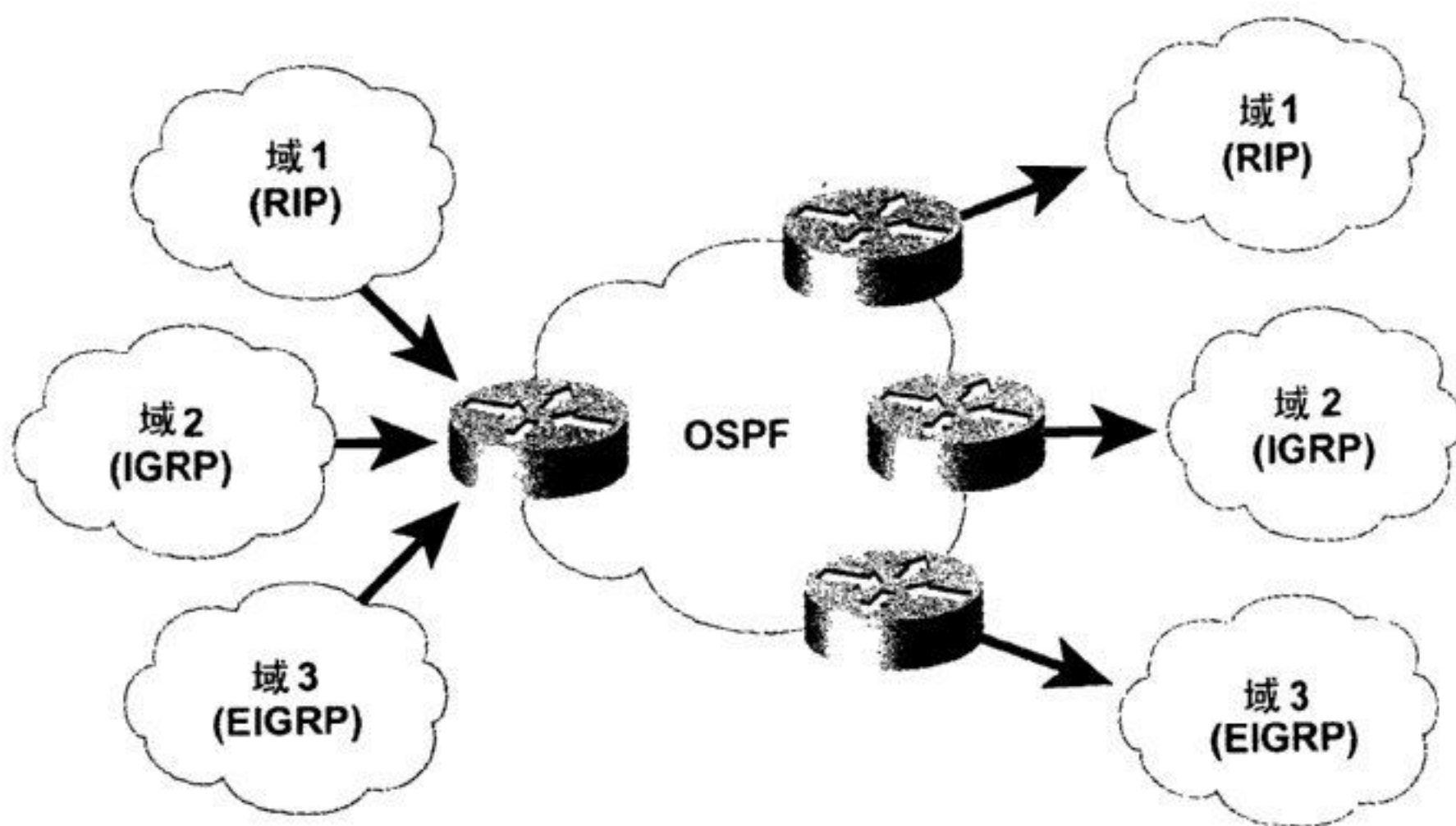


图 14-17 图左边 3 个域的路由被重新分配到一个运行 OSPF 的传输域。图右边域的路由必须被重新分配回它们的源域

处理这个问题的另一种方法是在 OSPF 传输域的入口对路由进行标记, 该标记在每个域内均是惟一的。在出口处, 将借助标记重新分配路由, 而不是通过明确的地址。传输网络的路由选择协议不会使用该标记, 而仅仅是向外部网络来回传送它们。RIPv2、EIGRP、集成 IS-IS 和 OSPF 都支持路由标记。BGP 也支持路由标记。RIPv1 和 IGRP 不支持标记。

回顾一下第 7 章、第 8 章、第 9 章的报文格式, 可以看出 RIPv2 消息支持 16 位标记, 而 EIGRP 外部路由 TLV 和 OSPF 5 型 LSA 支持 32 位标记。这些标记可以用十进制数表示, 所以 RIPv2 所携带的标记在 0~65535 之间, EIGRP 和 OSPF 携带的标记在 0~4 294 967 295 之间。

在图 14-18 中, 路由器正在接受来自 3 个不同路由选择域的路由, 并且把它们重新分配到 OSPF 域。目的是标记来自每个域的路由以便在 OSPF 云内可以标识它们的源点域。来自域 1 的路由标记为 1, 域 2 的标记为 2, 等等。

Dagwood 配置如下:

```
router ospf 1
 redistribute igrp 1 metric 10 subnets tag 1
 redistribute rip metric 10 subnets route-map Dithers
 network 10.100.200.1 0.0.0.0 area 0
!
router rip
 network 10.0.0.0
!
router igrp 1
 network 10.0.0.0
!
access-list 1 permit 10.1.2.3
access-list 2 permit 10.1.2.4
!
```



```

route-map Dithers permit 10
  match ip route-source 1
  set tag 2
!
route-map Dithers permit 20
  match ip route-source 2
  set tag 3

```

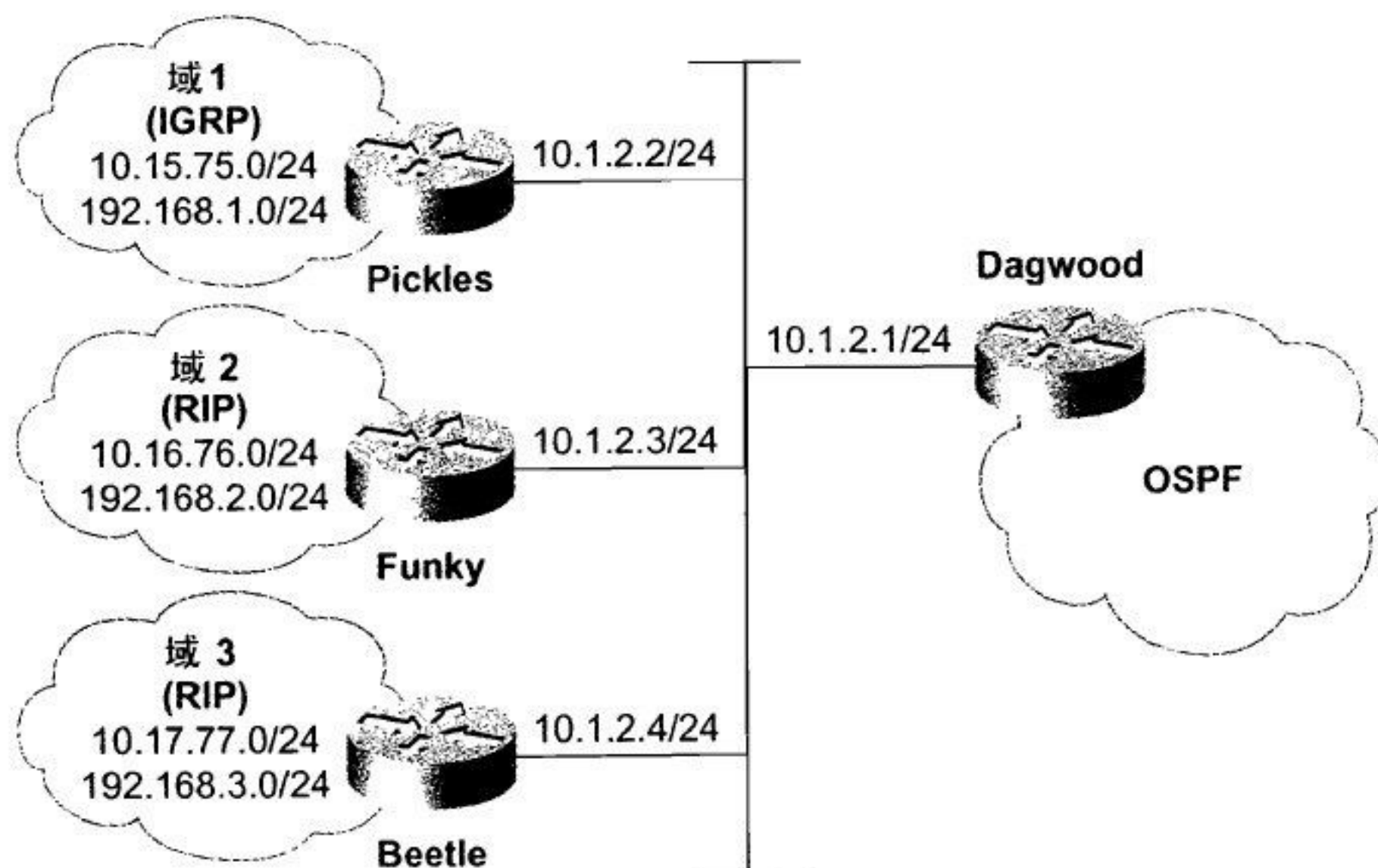


图 14-18 配置 Dagwood，使得来自这 3 个域的路由在被重新分配到 OSPF 时被标记

首先，注意 OSPF 配置下的命令 **redistribute igrp**。Dagwood 正在接受来自 IGRP 域的路由，由于仅有一个 IGRP 域，所以直接在 **redistribute** 命令上设置标记为 1。然而，RIP 路由是来自两个 RIP 域的，因此这里需要路由图。路由图 Dithers 设置 RIP 路由的标记为 2 或 3，具体依赖于路由是学自 Funky (10.1.2.3) 还是 Beetle (10.1.2.4)。图 14-19 给出了一个 LSA 通告，被通告的路由是从 RIP 那里学到的，标记被设置为 2。在 OSPF 的链路状态数据库中也可以观察到路由标记（图 14-20）。

在图 14-21 中，Blondie 必须仅向 Alley 重新分配域 2 的路由，仅向 Oop 重新分配域 1 的路由。因为这些路由在进入 OSPF 传输域时已经被标记过，按如下方式很容易实现：

```

router ospf 1
  network 10.100.200.2 0.0.0.0 area 0
!
router rip
  redistribute ospf 1 match external 2 route-map Daisy
  passive-interface Ethernet0
  passive-interface Serial1
  network 10.0.0.0
  default-metric 5
!
router igrp 1
  redistribute ospf 1 match external 2 route-map Herb
  passive-interface Ethernet0

```

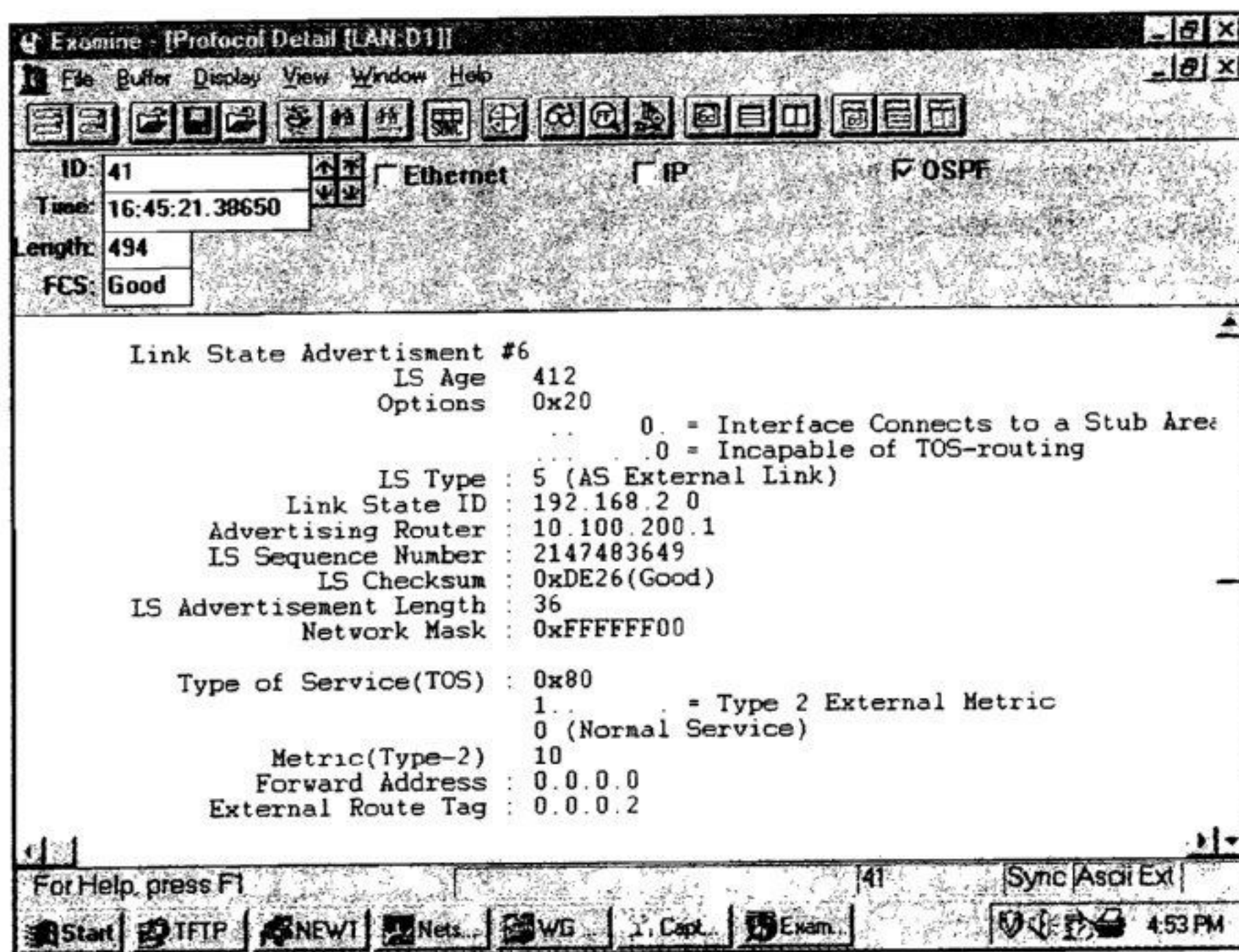



图 14-19 这个 5 型 LSA 正在通告域 2 内的网络 192.168.2.0, 域 2 在 OSPF 域内。路由标记见最后一行

```
Blondie#show ip ospf database
```

```
OSPF Router with ID (10.100.200.2) (Process ID 1)
```

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.100.200.2	10.100.200.2	39	0x80000002	0x6FF5	1
10.100.200.1	10.100.200.1	40	0x80000033	0x33E1	1

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
10.100.200.1	10.100.200.1	40	0x80000001	0xB0A7

AS External Link States

Link ID	ADV Router	Age	Seq#	Checksum	Tag
192.168.2.0	10.100.200.1	641	0x80000028	0x904D	2
10.17.77.0	10.100.200.1	642	0x80000028	0xC817	3
192.168.3.0	10.100.200.1	642	0x80000028	0x9744	3
10.15.75.0	10.100.200.1	642	0x80000028	0xD213	1
10.1.2.0	10.100.200.1	642	0x80000028	0xA19B	1
10.16.76.0	10.100.200.1	642	0x80000028	0xCD15	2
192.168.1.0	10.100.200.1	644	0x80000028	0x8956	1
10.100.200.0	10.100.200.1	644	0x80000028	0x6EA4	1

```
Blondie#
```

图 14-20 OSPF 链路状态数据库指明了每个外部路由的标记, 该标记是由 Dagwood 的重新分配进程设置的


```

passive-interface Serial0
network 10.0.0.0
default-metric 10000 1000 255 1 1500
!
route-map Daisy permit 10
match tag 2
!
route-map Herb permit 10
match tag 1

```

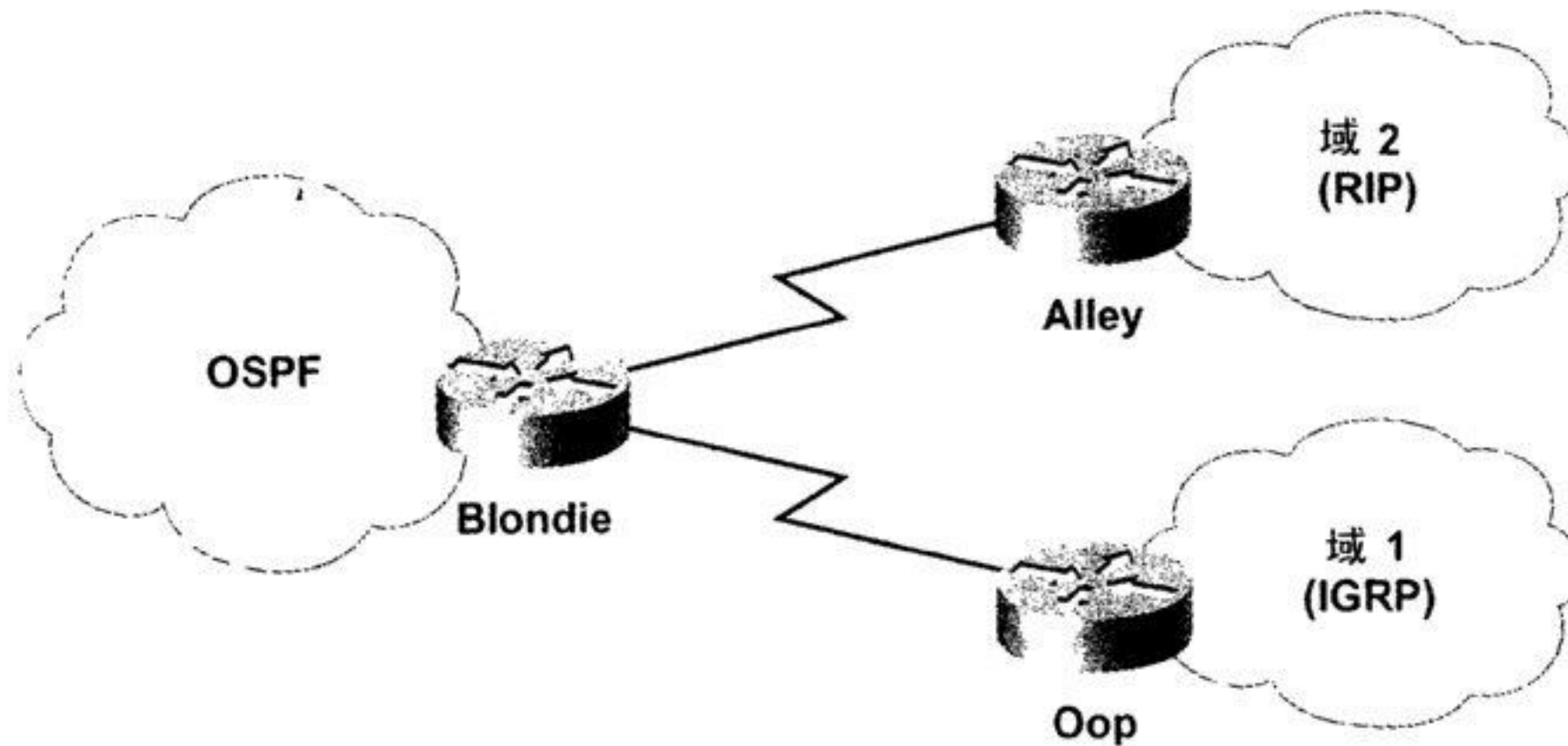


图 14-21 Blondie 使用路由图根据路由标记重新分配路由

图 14-22 给出了在 Alley 和 Oop 上的结果路由。使用路由标记过滤路由的一个缺点是不能通过接口过滤路由。例如，如果 Blondie 必须向域 2 和域 3 发送路由，而且这两个域都运行 RIP。不可能配置路由图向一个 RIP 进程发送某些路由，而向另一个 RIP 进程发送其他的路由。这些路由必须使用命令 **distribute-list** 借助地址来过滤。

```

Alley#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
       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

Gateway of last resort is not set

10.0.0.0 255.255.255.0 is subnetted, 4 subnets
C      10.1.3.0 is directly connected, Serial0
R      10.1.4.0 [120/1] via 10.1.3.1, 00:00:19, Serial0
R      10.16.76.0 [120/5] via 10.1.3.1, 00:00:19, Serial0
R      10.100.200.0 [120/1] via 10.1.3.1, 00:00:19, Serial0
R      192.168.2.0 [120/5] via 10.1.3.1, 00:00:19, Serial0
Alley#

```

```

Oop#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
       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

```

待续


```

Gateway of last resort is not set

    10.0.0.0 255.255.255.0 is subnetted, 5 subnets
I       10.1.3.0 [100/10476] via 10.1.4.1, 00:00:22, Serial0
I       10.1.2.0 [100/8676] via 10.1.4.1, 00:00:22, Serial0
C       10.1.4.0 is directly connected, Serial0
I       10.15.75.0 [100/9176] via 10.1.4.1, 00:00:22, Serial0
I       10.100.200.0 [100/8576] via 10.1.4.1, 00:00:22, Serial0
I       192.168.1.0 [100/9176] via 10.1.4.1, 00:00:22, Serial0
Oop#

```

图 14-22 在图 14-21 中, Alley 和 Oop 的路由选择表显示了 Blondie 的配置结果

14.3 展 望

本章结束了本书对 TCP/IP 路由中内部网关协议的深入讨论, 如果你正准备成为 CCIE, 那么你当然想在考试之前知道本书的主题。你可以使用每章后面的问题测试一下你的理解水平和准备水平。如果你没有学习过有关外部网关协议的 TCP/IP 路由, 那么在你下一步安排中, 学习这部分内容是较为合理的。

14.4 总结表: 第 14 章命令回顾

命 令	描 述
access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>]	定义标准 IP 访问列表中的一条
access-list <i>access-list-number</i> { deny permit } <i>protocol</i> <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [log]	定义扩展 IP 访问列表中的一条
ip local policy <i>route-map</i> <i>map-tag</i>	为路由器产生的报文定义策略路由
ip policy <i>route-map</i> <i>map-tag</i>	为经过路由器的报文定义策略路由
match interface <i>type number</i> [... <i>type number</i>]	匹配路由, 其中该路由的下一跳是这些接口之一
match ip address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	匹配路由, 其中该路由的目标地址被访问列表指明
match ip next-hop { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	匹配路由, 其中该路由的下一跳路由器地址被访问列表指明
match ip route-source { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	匹配路由, 其中通告该路由的路由器被访问列表指明
match length <i>min max</i>	匹配报文的层 3 长度
match metric <i>metric-value</i>	匹配指定度量的路由
match route-type { internal external [type-1 type-2] [level-1 level-2]	匹配指定类型的 OSPF、EIGRP 或 IS-IS 路由
match tag <i>tag-value</i> [... <i>tag-value</i>]	匹配指定标记的路由
redistribute <i>protocol</i> [<i>process-id</i>] { level-1 level-1-2 level-2 } [metric <i>metric-value</i>] [metric-type <i>type-value</i>] [match { internal external 1 external 2}] [tag <i>tag-value</i>] [route-map <i>map-tag</i>] [weight <i>weight</i>] [subnets]	配置进入路由选择协议的重新分配, 并且指明被重新分配路由的源点
set level { level-1 level-2 level-1-2 stub-area backbone }	设置 IS-IS 级别或设置 OSPF 区域, 其中匹配成功的路由将要被重新分配进入该区域
set default interface <i>type number</i> [... <i>type number</i>]	当不存在到达目标网络的显式路由时, 为匹配成功的报文设置出站接口

续表

命 令	描 述
<code>set interface type number [...type number]</code>	当存在到达目标网络的显式路由时，为匹配成功的报文设置出站接口
<code>set ip default next-hop ip-address [...ip-address]</code>	当不存在到达目标网络的显式路由时，为匹配成功的报文设置下一跳路由器地址
<code>set ip next-hop ip-address [...ip-address]</code>	当存在到达目标网络的显式路由时，为匹配成功的报文设置下一跳路由器地址
<code>set ip precedence precedence</code>	设置被匹配 IP 报文服务类型字段中的优先级比特
<code>set ip tos type-of-service</code>	设置被匹配报文服务类型字段中 TOS 比特
<code>set metric {metric-value bandwidth delay reliability loading mtu}</code>	为被匹配路由设置度量值
<code>set metric-type {internal external type-1 type-2}</code>	为将要被重新分配进入 IS-IS 或 OSPF 的被匹配路由设置度量类型
<code>set next-hop next-hop</code>	为被匹配路由设置下一跳路由器地址
<code>set tag tag-value</code>	为被匹配路由设置标记值

14.5 复 习 题

- 1. 路由图有哪些方面类似于访问列表？两者有什么不同？
- 2. 策略路由是什么？
- 3. 路由标记是什么？
- 4. 路由标记以什么样的方式影响路由选择协议？

14.6 配置练习

- 1. 在图 14-23 中，请为路由器 A 配置策略路由，使得从子网 172.168.1.0/28 出发经过 172.16.1.112/28 到达 A 的报文被转发到路由器 D，而从子网 172.16.1.128/28 出发经 172.16.1.240/28 到达 A 的报文被转发到路由器 E。
- 2. 在图 14-23 中，为路由器 A 配置策略路由，使得从子网 172.16.1.64/28 出发经过 172.16.1.112/28 到达路由器 C 的报文被转发到路由器 D，而相同的报文如果到达路由器 D，则被转发到 E。所有其他报文将被正常地转发。
- 3. 在图 14-23 中，为路由器 A 配置策略路由，使得所有经 172.16.1.240/28 去往子网 172.16.1.0/28 且源端口为 SMTP 的报文被转发到路由器 C，任何其他去往相同子网的 UDP 报文被转发到 B。通过策略路由或常规路由没有向 C 或 B 转发其他的报文。
- 4. 在图 14-24 中，路由器的 OSPF 和 EIGRP 的配置如下：

```
router eigrp 1
network 192.168.100.0
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 16
```

配置路由器把内部EIGRP路由作为度量为 10 的E1路由向 OSPF 重新分配，把外部EIGRP

路由作为度量为 50 的 E2 路由向 OSPF 重新分配。EIGRP 域内的所有网络和子网除了 10.201.100.0/24 外, 都将被重新分配。

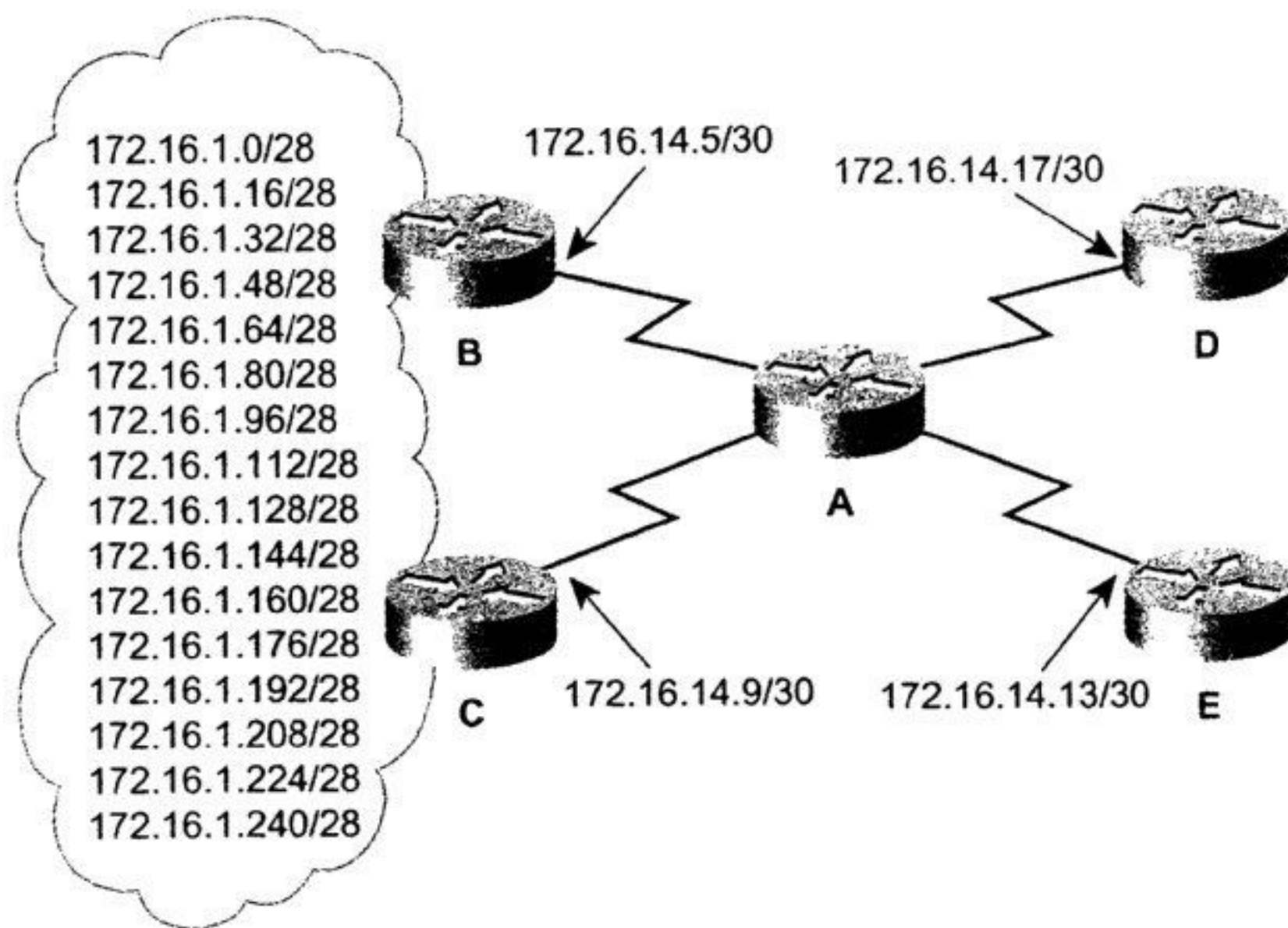


图 14-23 配置练习 1~3 中的互联网络

5. 配置图 14-24 中的路由器, 向 EIGRP 重新分配内部 OSPF 路由, 其中内部 OSPF 路由的时延要小于外部 OSPF 路由的时延。仅允许 OSPF 域内的 3 个 C 类网络被重新分配。

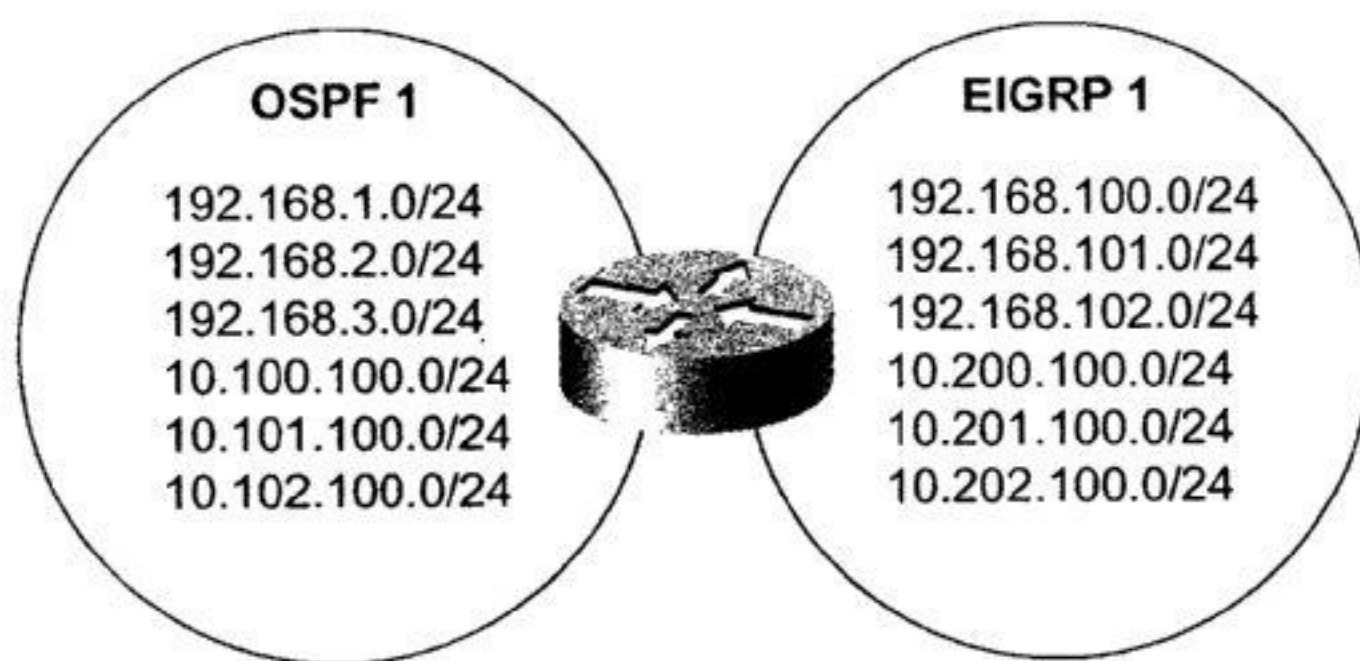


图 14-24 练习 4 和 5 的路由器配置

14.7 故障诊断练习

1. 已知下面配置:

```
interface TokenRing1
ip address 192.168.15.254 255.255.255.0
ip policy route-map Ex1
!
access-list 1 permit 192.168.0.0 0.0.255.255
access-list 101 permit host 192.168.10.5 any eq telnet
!
```



```
route-map Ex1 permit 5
  match ip address 1
  set ip next-hop 192.168.16.254
!
route-map Ex1 permit 10
  match ip address 101
  set ip next-hop 192.168.17.254
```

上面配置的目的是对所有源地址前缀在 192.168.0.0 ~ 192.168.255.255 中的报文进行策略路由。除了来自主机 192.168.10.5 的 Telnet 报文外, 所有其他报文将被转发到 192.168.17.254。在配置中有两个错误导致策略路由工作不正常, 请问错误是什么?

第四部分

附录

附录 A 教程：二进制和十六进制

附录 B 教程：访问列表

附录 C CCIE 小提示

附录 D 复习题答案

附录 E 配置练习答案

附录 F 故障排除练习答案

附录 A

教程：二进制和十六进制

理解二进制和十六进制的最佳方法是先透彻领悟十进制计数系统。十进制 (decimal) 是基于 10 的计数系统 (词根 deci-表示 10)。“基于 10”指的是由 10 个数位 (digit) 0 到 9 来表示数。通常我们使用十进制, 这是因为我们的祖先用他们的手指 (finger) 来计算牛、孩子、敌人 (事实上 digit 的意思就是“finger”)。

使用“位值 (place value)”, 可以用不多的几个数位 (如 10 个十进制数位) 来表示很大的数。所有计数系统的位值从最右边开始, 是基数的 0 次幂。从右往左, 基数的幂依次增大 1:

$$B^4 B^3 B^2 B^1 B^0$$

基数是 10 时, 前 5 个位值是:

$$10^4 10^3 10^2 10^1 10^0$$

对任何基数, 前两个位值是最容易计算的。任何数的 0 次幂是 1, 所以 $10^0=1$ 。任何数的 1 次幂就是它本身, 所以 $10^1=10$ 。从最左边的位值开始, 往右依次乘以基数:

$$10^0 = 1$$

$$10^1 = 1 \times 10 = 10$$

$$10^2 = 10 \times 10 = 100$$

$$10^3 = 100 \times 10 = 1000$$

$$10^4 = 1000 \times 10 = 10\,000$$

所以, 对于基数是 10 的计数系统, 前 5 个位值是:

$$10\,000 \quad 1\,000 \quad 100 \quad 10 \quad 1$$

根据位值来读一个数, 比如 57 258, 指的是有 5 个 10 000, 7 个 1 000, 2 个 100, 5 个 10, 以及 8 个 1。就是说:

$$5 \times 10\,000 = 50\,000$$

$$7 \times 1\,000 = 7\,000$$

$$2 \times 100 = 200$$

$$5 \times 10 = 50$$

$$8 \times 1 = 8$$

将这些结果相加, 结果是 $50\,000 + 7\,000 + 200 + 50 + 8 = 57\,258$ 。

我们对十进制都非常熟悉了, 所以, 我们很少会去想把一个数分解成位值。但是, 这种方法对于阐明其他进制的数是非常至关重要的。

A.1 二进制数

计算机从最底层来看, 只不过是电子开关的集合而已。而数字和字符是由这些开关的状态来表示的。由于一个开关仅有两种状态——开或者关, 所以它使用二进制 (binary), 或者说基数为 2 的计数系统 (词根 *bi* 表示 2)。一个基数为 2 的系统仅仅有两个数位: 0 和 1。计算机通常将这两个数位集成 8 个位值, 即一个字节 (byte) 或 8bit 字节 (octet)。这 8 个位值是:

$$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$$

位值这样计算:

$$2^0 = 1$$

$$2^1 = 1 \times 2 = 2$$

$$2^2 = 2 \times 2 = 4$$

$$2^3 = 4 \times 2 = 8$$

$$2^4 = 8 \times 2 = 16$$

$$2^5 = 16 \times 2 = 32$$

$$2^6 = 32 \times 2 = 64$$

$$2^7 = 64 \times 2 = 128$$

所以一个二进制 8 位组的位值是: 128 64 32 16 8 4 2 1

因此, 二进制 8 位组 10010111 可以这样理解:

$$1 \times 128 = 128$$

$$0 \times 64 = 0$$

$$0 \times 32 = 0$$

$$1 \times 16 = 16$$

$$0 \times 8 = 0$$

$$1 \times 4 = 4$$

$$1 \times 2 = 2$$

$$1 \times 1 = 1$$

或者 $128 + 16 + 4 + 2 + 1 = 151$

对于二进制数, 因为每一个位值要么就是该值本身, 要么就没有, 所以比较简单。另外一个例子: $11101001 = 128 + 64 + 32 + 8 + 1 = 233$ 。就是说, 将二进制转为十进制仅仅是一个

将位值相加的过程，将十进制转为二进制仅仅是将位值相减的过程。例如，要将十进制数 178 转为二进制，首先把 178 减去最高的位值：

- 1. 178 大于 128，我们就知道在该位值上有一个 1：178-128=50。
- 2. 50 比 64 小，该位值上有一个 0。
- 3. 50 比 32 大，所以该位值上有一个 1：50-32=18。
- 4. 18 比 16 大，该位值上有一个 1：18-16=2。
- 5. 2 比 8 小，该位值上有一个 0。
- 6. 2 比 4 小，该位值上有一个 0。
- 7. 2 等于 2，该位值上有一个 1：2-2=0。
- 8. 0 小于 1，该位值上有一个 0。

把这些步骤的结果综合起来，用二进制表示 178 就是 10110010

另外一个例子可能会有帮助。给出 110：

- 1. 110 比 128 小，所以在位值上有一个 0。
- 2. 110 比 64 大，所以在位值上有一个 1：110-64=46。
- 3. 46 比 32 大，所以在位值上有一个 1：46-32=14。
- 4. 14 比 16 小，所以在位值上有一个 0。
- 5. 14 比 8 大，所以在位值上有一个 1：14-8=6。
- 6. 6 比 4 大，所以在位值上有一个 1：6-4=2。
- 7. 第 2 个位值上有一个 1：2-2=0。
- 8. 0 比 1 小，所以在位值上有一个 0。

所以，110 用二进制表示就是 01101110。

A.2 十六进制数

写一个二进制 8 位组并不有趣。对于经常要使用这些数字的人来说，受欢迎的是更简洁的表示法。一个可能的表示法是为每一个可能的 8 位组分配一个单独的字符。但是，8 位有 $2^8=256$ 种不同的组合，所以，用单独的字符表示所有 8 位组需要 256 个数位，或者说一个基数为 256 的计数系统。

将一个 8 位组看作是两个各 4 位的组合或许会更简单一些。例如，11010011 可以看作是 1101 和 0011。对 4 个位来说，有 $2^4=16$ 种不同的组合，所以有基数 16，或者说十六进制 (hexadecimal) 计数系统，一个 8 位组可以用 2 位来表示 (词根 *hex* 的意思是“six”，*deci* 的意思是“ten”)。表 A.1 列出了十六进制数以及相应的十进制数和二进制数。

表 A.1 十六、十、二进制数

十六进制	十进制	二进制
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101

续表

十六进制	十进制	二进制
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

因为十六进制和十进制的前 10 个数字是一样的, 所以我们有意在一个十六进制数前面加 0x, 或者在后面加一个 h, 以便和十进制数区别开。例如, 十六进制数 25 应该写成 0x25 或者 25h。本书使用 0x 表示法。

刚才学过二进制的表示法, 很容易写出一个 4 位二进制数的十进制表达形式。同样也很容易将一个十进制数转为十六进制。于是, 我们可以很容易地通过 3 个步骤将一个二进制 8 位组转为十六进制:

1. 将 8 位组分成 2 个 4 位的二进制数
2. 将每个 4 位二进制数转为十进制
3. 把每个十进制数用十六进制来表示

例如: 把 11010011 转为十六进制:

1. 11010011 变成 1101 和 0011
2. $1101=8+4+1=13$, $0011=2+1=3$
3. $13=0xD$, $3=0x3$

所以, 11010011 用十六进制表示就是 0xD3。

把十六进制转为二进制是上述 3 步的简单逆序。例如, 把 0x7B 转为二进制:

1. $0x7=7$, $0xB=11$
2. $7=0111$, $11=1011$
3. 把 2 个 4 位二进制数写在一起就是 $0x7B=01111011$

附录 B

教程：访问列表

目前对访问列表的命名可能并不是很恰当，访问列表最初的目的是许可或拒绝访问进入、离开或经过路由器的报文。而现在访问列表已成为控制帧和报文行为的强大工具，他们的用途可分为 3 类（图 B-1）：

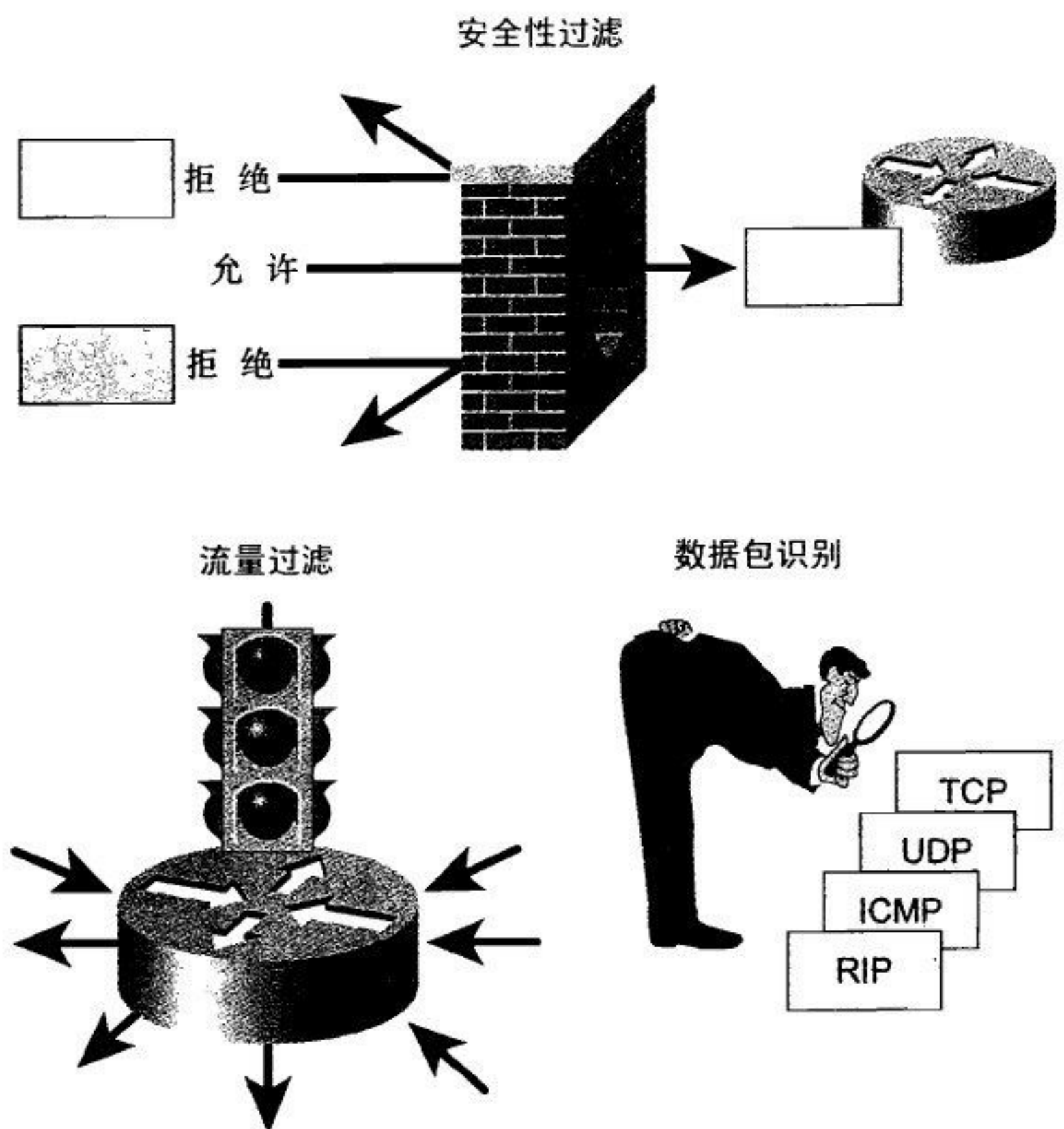


图 B-1 访问列表可以用做安全过滤表、流量过滤表或用于报文标识

- 安全过滤可以保护路由器及路由器传递流量所到达网络的完整性。一个安全过滤表许可少数清晰的报文

通过, 而拒绝其他任何报文通过。

- 流量过滤可以阻止不必要的报文通过带宽有限的链路。这些过滤表的外表和行为同安全过滤表很相似, 但是逻辑一般是颠倒的: 流量过滤表拒绝少数不必要的报文而许可其他任何报文通过。
- 为了能够发挥正常的作用, 在 Cisco 路由器上的许多工具, 例如拨号表、路由过滤表、路由图和队列表, 都必须能够标识确定的报文。访问列表可以链接到这些或其他工具上, 并且提供报文标识功能。

B.1 访问列表基础知识

一个访问列表是一连串顺序的过滤规则。每个过滤规则包括某种匹配准则和一个操作。操作不是许可就是拒绝。匹配准则可能像源地址一样简单, 或者有可能是一个更加复杂的源目地址、协议类型、端口号或管套和某些标记状态说明 (TCP ACK 比特) 的组合。

一个报文从栈顶进入过滤表 (图 B-2), 在每个过滤规则中的匹配准则被应用, 如果匹配发生, 那么指定的许可或拒绝操作将被执行。如果匹配没有发生, 报文将顺着向下移动到栈中下一个过滤规则并再次重复匹配过程。

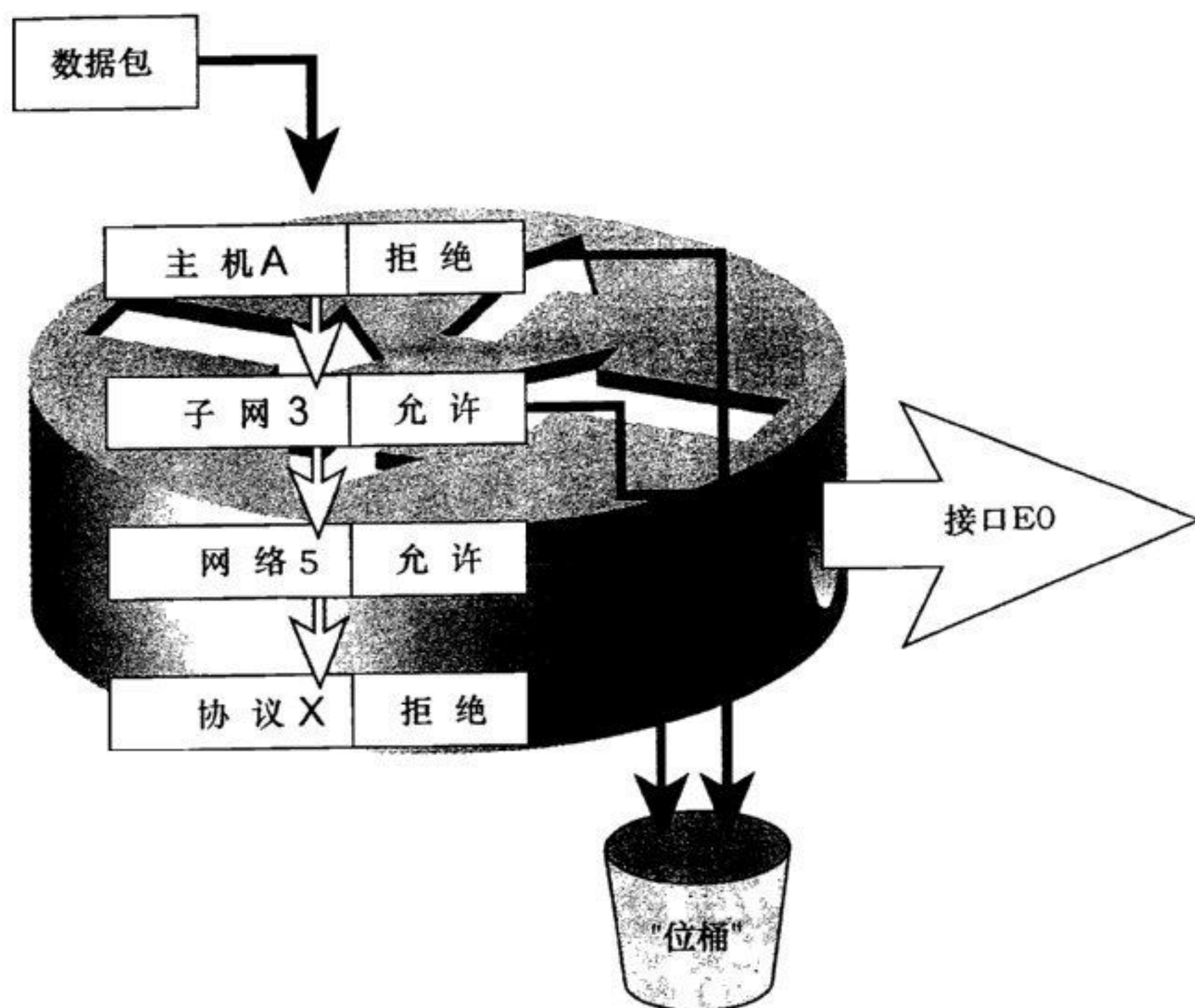


图 B-2 一个访问列表是一连串顺序的过滤规则, 每个过滤规则定义了一个匹配准则和一个操作

在图 B-2 中, 许可意味着报文将被允许离开接口 E0; 拒绝意味着报文将被丢弃。例如, 源地址为主机 A 的报文将在第一个过滤规则中被丢弃。假设报文的源地址是网络 5 中子网 2 上的主机 D, 而第一个过滤规则指明了对主机 A 的匹配准则, 因此没有匹配发生, 报文进入第二层。在第二个过滤规则中又指明了子网 3, 因此还是没有匹配发生。报文进入第三个过

滤规则，该规则指明了网络 5，因此匹配成功。由于本层的操作是许可，所以报文被允许离开接口 E0。

B.1.1 隐式拒绝一切

如果报文经过所有过滤规则都没有发生匹配，那会出现什么情况？在这种情况下路由器必须知道该如何处理报文，也就是必须有一个缺省操作。缺省操作可以是允许所有没有匹配的报文通过或拒绝它们通过。Cisco 选择了拒绝它们通过：对任何经过访问列表的报文，如果没有发生匹配将会被自动丢弃。

这种方法是一个正确的工程选择，特别是在访问列表用于安全控制时。丢弃那些本不应该被丢弃的报文比起许可那些你因疏忽而没有进行过滤的报文，应该更好。

最后一个过滤规则叫做隐式的拒绝一切（implicit deny any）（图 B-3）。正如名字所暗指的，在你创建的任何一个访问列表中都不会显示这一规则。它仅仅是一个缺省操作，并且它在所有访问列表中都处于最后。

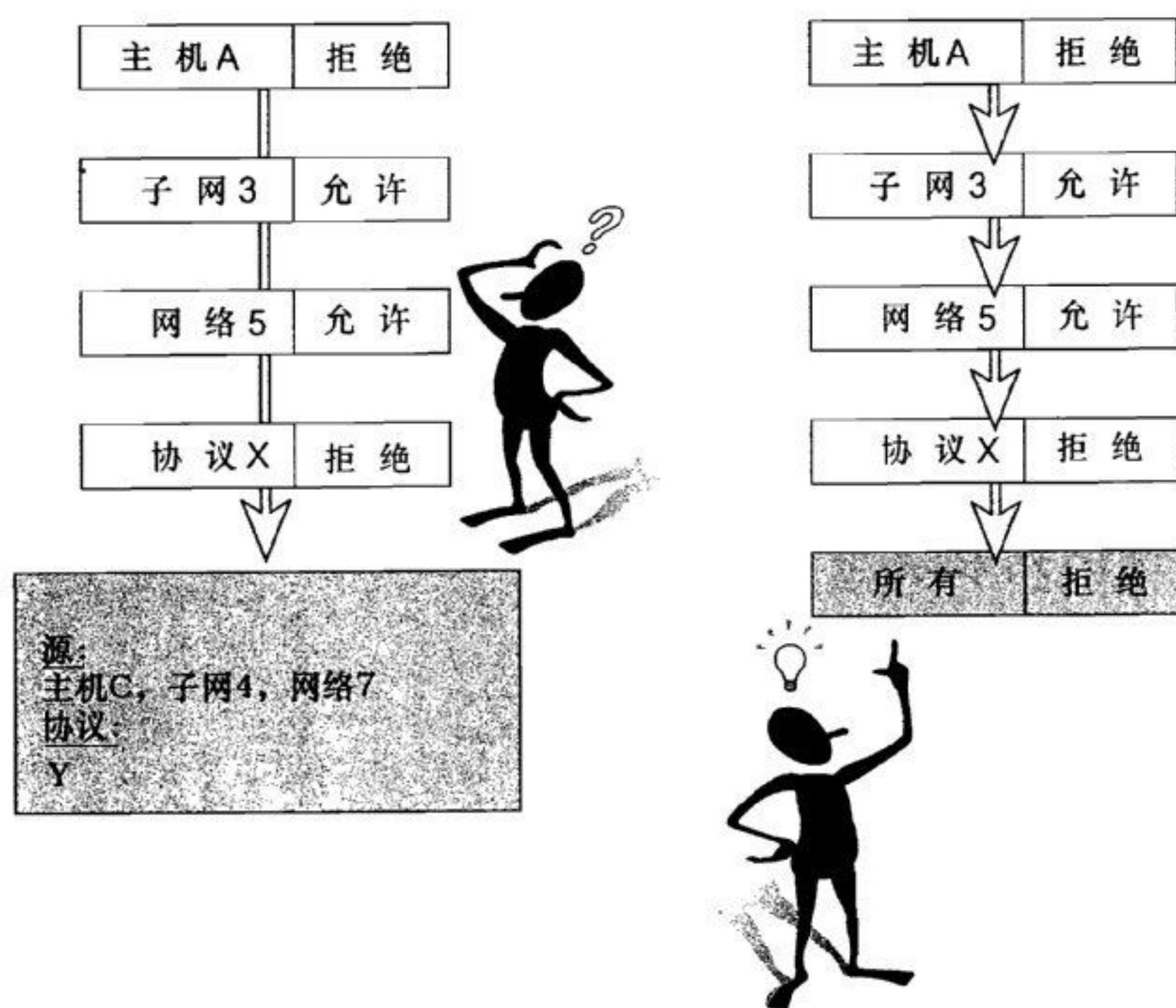


图 B-3 所有以隐式拒绝一切规则结尾的访问列表将丢弃那些在表中没有匹配的报文

通过在最后一行建立显式的许可一切（explicit permit any）可以覆盖这个缺省操作。这里隐含了一点，即经过其他所有过滤规则的报文在到达缺省的拒绝一切操作之前，会匹配到许可一切规则，因而没有匹配到任何规则的报文将被许可——不会有报文到达隐式的拒绝操作。

B.1.2 顺序性

访问列表是从上到下顺序被执行的。这一概念很重要：或许访问列表发生故障的最普遍的原因就是过滤规则的放置顺序出现错误。

在图 B-4 中, 子网 10.23.147.0/24 应该被拒绝, 而网络 10.0.0.0 其余的子网应该被许可。左边的访问列表顺势有错误, 网络 10.0.0.0 (包括它的子网 10.23.147.0) 将匹配到第 1 行并且被许可。而被拒绝的子网报文根本不会到达第 2 行。

右边的访问列表是准确的。子网 10.23.147.0 匹配到第 1 行并且被拒绝, 而 10.0.0.0 的其余子网将会匹配到第 2 行并且被许可。

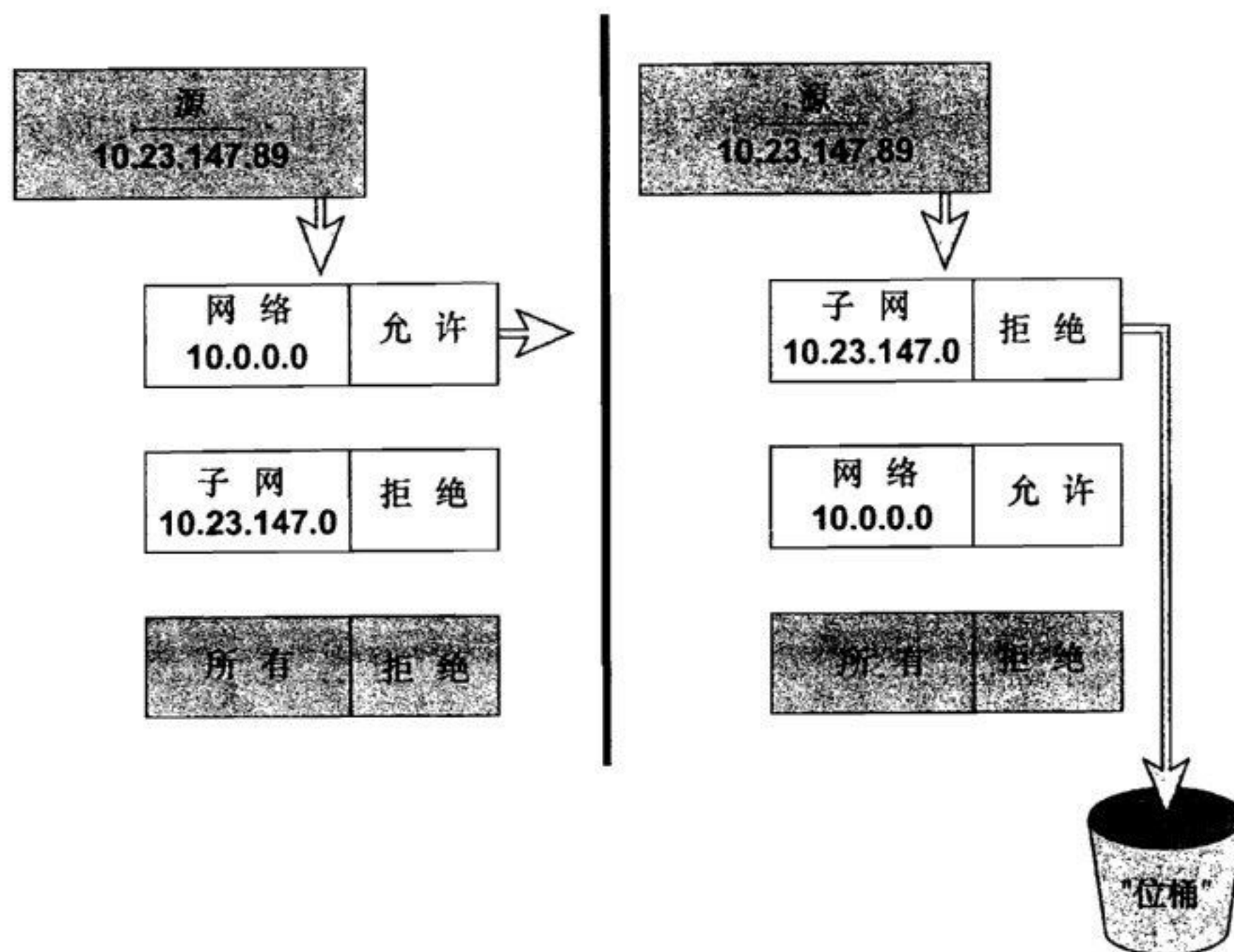


图 B-4 如果访问列表中的个别过滤规则没有被按照正确的顺序配置, 那么访问列表将不能正常工作

B.1.3 访问列表类型

下面是图 B-4 中右边的访问列表的实际配置:

```
access-list 9 deny 10.23.147.0 0.0.0.255
access-list 9 permit 10.0.0.0 0.255.255.255
```

一个配置行表示访问列表的一个过滤层。这里将简略地讨论一下访问列表中的各种组件, 但请先注意在两行中都出现的数字 9, 这个数字是访问列表编号, 它主要有两个用途:

- 将访问列表中的所有行都连接在一些, 使得该访问列表区别于路由器配置中的其他访问列表 (通常在一个路由器中存在多个访问列表)。
- 路由器必须有办法区分访问列表类型。Cisco IOS 有 IP、IPX、AppleTalk、DEC、NetBIOS、桥接和许多其他协议的访问列表, 而且其中许多协议都有多种访问列表类型。访问列表编号可以表明路由器访问列表的类型。

访问列表类型可以通过数字和名字来标识。表 B-1 给出了编号的访问列表类型和每种类型访问列表可用的编号范围。例如, 因为编号 1010 在 1000~1099 之间, 所以 access list 1010 标识了 IPX SAP。

在一个范围内, 访问列表编号不必按照任何特殊顺序, 也就是说, 在路由器中不必按第一个 AppleTalk 表为 600, 第二个为 601 等等进行编号。它们可以使用 600~699 中任意一个

数字，只要保证在一个路由器中每个访问列表编号惟一就可以。

表 B-1 Cisco 访问列表编号

访问列表类型	范 围
标准 IP	1-99
扩展 IP	100-199
以太网类型代码	200-299
以太网地址	700-799
透明桥接（协议类型）	200-299
透明桥接（厂商代码）	700-799
扩展的透明桥接	1100-1199
DECnet 和扩展的 DECnet	300-399
XNS	400-499
扩展 XNS	500-599
AppleTalk	600-699
源路由桥接（协议类型）	200-299
源路由桥接（厂商代码）	700-799
标准 IPX	800-899
扩展 IPX	900-999
IPX SAP	1000-1099
NLSP 路由汇总	1200-1299
标准 VINES	1-99
扩展 VINES	100-199
简单 VINES	200-299

此外，请注意，不同协议的一些编号范围是相同的，例如以太网类型代码、源路由桥接和简单 VINES。在这种情况下，路由器将会通过访问列表自身的格式来区分访问列表类型。

可以使用名字代替数字来标识下面的访问列表类型：

- Apollo 域
- 标准 IP
- 扩展 IP
- ISO CLNS
- 源路由桥接 NetBIOS
- 标准 IPX
- 扩展 IPX
- IPX Sap
- IPX NetBIOS
- NLSP 路由汇总

在下面的例子中，名字为 Boo 的访问列表用来标识 IPX NetBIOS：

```
netbios access-list host Boo deny Atticus
netbios access-list host Boo deny Scout
netbios access-list host Boo deny Jem
netbios access-list host Boo permit *
```

注意：虽然标准和扩展 IP 访问列表通常使用编号，但是它们也可以使用名字。在 IOS 11.2 及更高的版本中支持这一协定。在某些环境中，可能会使用大量 IP 表配置路由器，用名字代

替数字, 可以更加容易地标识单独的访问列表。而且, 命名访问列表打破了 99 个标准 IP 访问列表和 100 个扩展 IP 表的限制。

命名 IP 访问列表当前仅可以与报文和路由过滤一起使用。更详细的信息请参考 Cisco 配置指南。

B.1.4 编辑访问列表

任何一个曾经从控制台上编辑过多行访问列表的人, 都会告诉你这是一个在挫折中经受锻炼的过程。在控制台上, 没有办法向表的中间添加一行。所有新输入的行都被添加到表尾。如果你发现输入错误并且试图删去其中特定的一行, 例如,

```
no access-list 101 permit tcp 10.2.5.4 0.0.0.255 192.168.3.0 0.0.0.255 eq 25
```

那么访问列表 101 所有行将与这一行一起被删除。

更方便的办法是剪切和粘贴访问列表到你 PC 的记事本, 或者上载配置到 TFTP 服务器上, 然后在那里编辑访问列表。当编辑结束后, 新的访问列表将被载入路由器。但是, 提醒一下, 所有新输入的行都会被添加到表尾。记住, 始终将 **no access-list #** 加在被编辑访问列表的开始, 其中 # 是你正在编辑的访问列表编号。例如:

```
no access-list 5
access-list 5 permit 172.16.5.4 0.0.0.0
access-list 5 permit 172.16.12.0 0.0.0.255
access-list deny 172.16.0.0 0.0.255.255
access-list permit any
```

no access-list 5 将会在添加新访问列表之前, 从配置文件中删除旧的访问列表 5。如果你省略了这一步, 那么新表仅仅会被添加到旧表的结尾。

B.2 标准 IP 访问列表

标准访问列表的格式如下:

access-list *access-list-number* {deny|permit} *source* [*source-wildcard*]

命令根据表 B-1 指定了在 1~99 之间的访问列表编号、操作 (许可和拒绝)、源 IP 地址、掩码通配符 (或反码)。一个标准 IP 访问列表的例子如下:

```
access-list 1 permit 172.22.30.6 0.0.0.0
access-list 1 permit 172.22.30.95 0.0.0.0
access-list 1 deny 172.22.30.0 0.0.0.255
access-list 1 permit 172.22.0.0 0.0.31.255
access-list 1 deny 172.22.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

例子中的前两行允许源地址属于指定主机 172.22.30.6 和 172.22.30.96 的报文通过。这看

似很恰当，尽管反码 0.0.0.0 可能没有意义。第 3 行拒绝子网 172.22.30.0 上所有其他主机。这也是相当直观的。但第 4 行的目的就不是很明显了，它允许地址范围是 172.22.0.1~172.22.31.255 的主机通过，反码指定了本行的地址范围。第 5 行拒绝 B 类网络 172.22.0.0 的所有子网，最后一行允许所有其他地址。

为了完全地理解访问列表，你必须理解反码。

回忆一下 IP 地址掩码的作用：为了从主机地址导出网络地址或子网地址，掩码中的 1 对应着网络位，0 对应着主机位。在每一位上执行 Boolean AND 操作可以得到网络或子网号。图 B-5(a)包括了 AND 函数的真值表和用英文表达的函数状态。

拿两位作比较，当且仅当两位都为 1 时结果为 1。

布尔与		
	0 1	
0	0 0	172.22.30.13 = 10101100000101100001111000001101
1	0 1	255.255.255.0 = 11111111111111111111111110000000
		172.22.30.0 = 10101100000101100001111000000000

(a)

布尔或		
	0 1	
0	0 1	172.22.30.0 = 10101100000101100001111000001101
1	1 1	0.0.0.255 = 00000000000000000000000011111111
		172.22.30.255 = 10101100000101100001111011111111

(b)

图 B-5 真值表、Boolean AND 和 Boolean OR 的例子

布尔或（Boolean OR）是布尔与（Boolean AND）的反函数，真值表如图 B-5(b)所示：拿两位作比较，当且仅当两位都为 0 时结果为 0。

一个反码（inverse mask）（Cisco 更喜欢用术语通配掩码（wildcard mask））把对应于地址位中将要被准确匹配的位设置为 0，其他位为 1——值为 1 的位经常叫做不关心位。接着反码将同地址进行 OR 操作。

注意，图 B-5(b)中的 OR 操作结果是 172.22.30.255。在 IP 术语中这一结果意指子网 172.22.30.0 上的所有主机地址。来自 172.22.30.0 的任何指定地址都将匹配到该地址/反码组合上。

图 B-6 给出了两种书写标准 IP 访问列表的快捷方法。图 B-6(a)给出的全 0 反码指明被讨论地址的所有 32 位都必须准确匹配到 172.22.30.6。对标准 IP 访问列表来说，缺省掩码是 0.0.0.0。所以给出的替换表述除了不带指定掩码外，同第一个表述完全相同。注意，这个缺省掩码不能应用到下一节要讨论的扩展 IP 访问列表。

图 B-6(b)给出了允许一切的地址/反码组合。地址 0.0.0.0 实际上仅是一个占位符，真正起作用的是掩码 255.255.255.255。通过将每一位都设置为 1，该掩码将匹配任何地址。可以替换该表述的方法是使用关键字 **any**，它于第一个表述的意思相同。

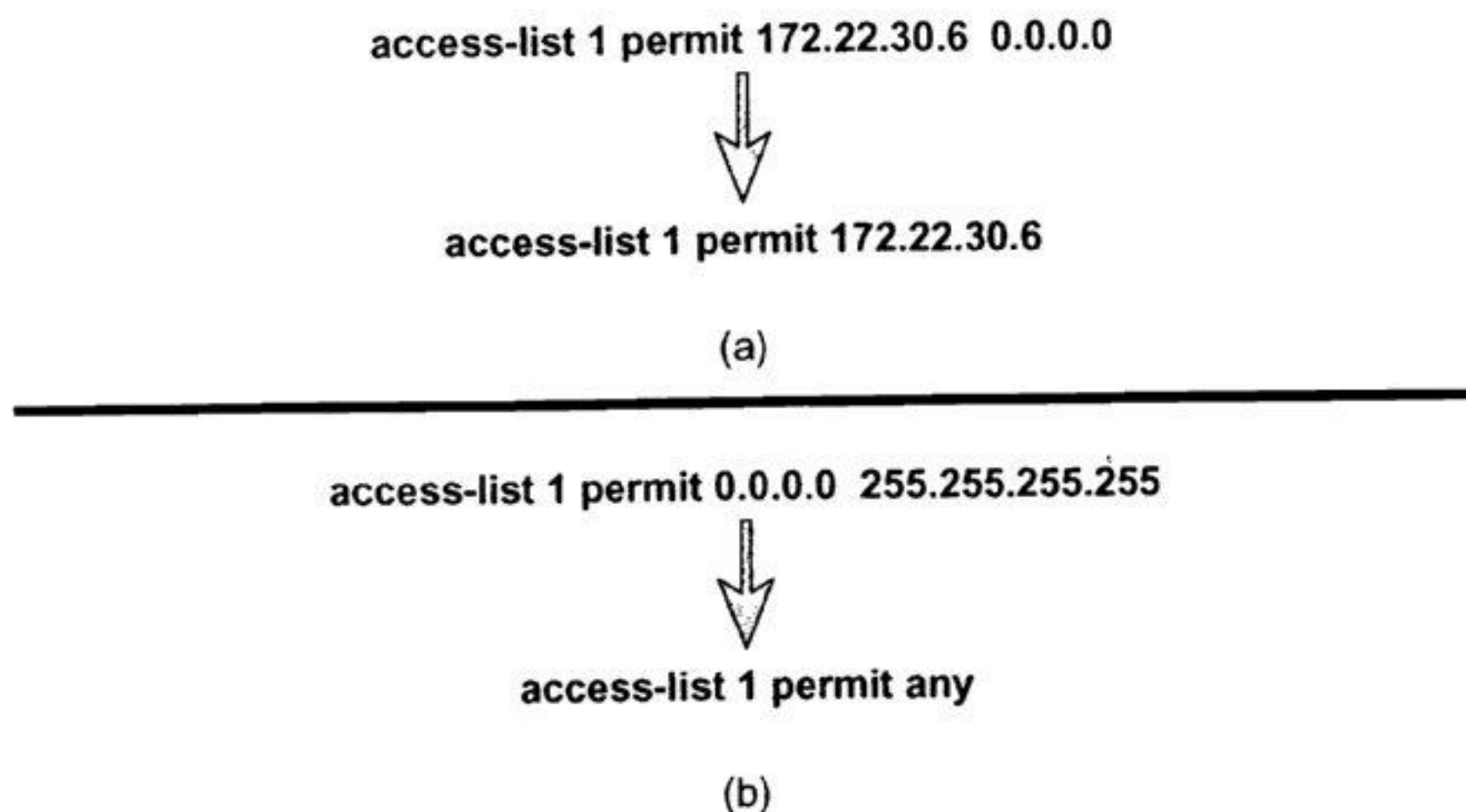


图 B-6 在配置标准 IP 访问列表时可以使用的两种快捷方法

B.3 扩展 IP 访问列表

扩展 IP 访问列表在指定过滤内容方面提供了更多的灵活性。扩展 IP 访问列表的基本格式如下：

```
access-list access-list-number {deny|permit} protocol source source-wildcard
destination destination-wildcard [precedence precedence] [tos] [log]
```

这里有一些特性已经很熟悉了，但还有一些新的特性：

- 对于扩展 IP 访问列表来说，**access-list-number** 范围在 100~199 之间。
- **protocol** 是一个新变量，它可以在 IP 报头的协议字段寻找匹配，可选择的关键字是 **eigrp**、**gre**、**icmp**、**igmp**、**igrp**、**ip**、**ipinip**、**nos**、**ospf**、**tcp** 和 **udp**。还可以使用 0~255 中的一个整数表示 IP 协议号。**ip** 是一个通用关键字，它可以匹配任意和所有的 IP 协议，同样地，反码 255.255.255.255 将可以匹配所有地址。
- 注意为了进行匹配，报文的 **source** 地址和 **destination** 地址都将被检查；它们有各自的反码。
- **precedence** 和 **tos** 是可选变量，它们可以在 IP 报头中的优先级字段和服务类型字段寻找匹配。优先级取值范围在 0~7 之间，TOS 在 0~15 之间，或者用关键字表示，可参见 Cisco 可用关键字列表文档。
- **log** 是一个可选项，它指定打开信息日志功能。

下面是一个扩展 IP 表的例子：

```
access-list 101 permit ip 172.22.30.6 0.0.0.0 10.0.0.0 0.255.255.255
access-list 101 permit ip 172.22.30.95 0.0.0.0 10.11.12.0 0.0.0.255
access-list 101 deny ip 172.22.30.0 0.0.0.255 192.168.18.27 0.0.0.0
access-list 101 permit ip 172.22.0.0 0.0.31.255 192.168.18.0 0.0.0.255
access-list 101 deny ip 172.22.0.0 0.0.255.255 192.168.18.64 0.0.0.63
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```


第 1 行: 源地址是 172.22.30.6 且目的地址属于网络 10.0.0.0 的报文被允许通过。

第 2 行: 源地址是 172.22.30.95 且目的地址属于子网 10.11.12.0/24 的报文被允许通过。

第 3 行: 源地址属于子网 172.22.30.0/24 且目的地址是 192.168.18.27 的报文被拒绝通过。

第 4 行: 源地址在 172.22.0.0~172.22.31.255 之间且目的地址属于网络 192.168.18.0 的报文被允许通过。

第 5 行: 源地址属于网络 172.22.0.0 且目的地址前 26 位是 192.168.18.64 的报文被拒绝通过。

第 6 行: 从任意源地址到任意目的地址的 IP 报文被允许通过。

图 B-7 给出了两种书写扩展 IP 访问列表的快捷方法。回想一下标准 IP 访问列表曾使用过缺省掩码 0.0.0.0, 但是这个缺省掩码不能用于扩展 IP 访问列表, 因为路由器无法正确地进行解释。但是对扩展表也存在一个替代表述。在图 B-7(a)中, 如果报文的源地址是主机 172.22.30.6 且目的地址为 10.20.30.40, 那么报文被允许通过。在扩展 IP 访问列表中出现的掩码 0.0.0.0, 可以通过在地址之前添加关键字 **host** 来代替。

```
access-list 101 permit ip 172.22.30.6 0.0.0.0 10.20.30.40 0.0.0.0
```



```
access-list 101 permit ip host 172.22.30.6 host 10.20.30.40
```

(a)

```
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```



```
access-list 101 permit ip any any
```

(b)

图 B-7 书写扩展 IP 访问列表的两种快捷方法

在图 B-7(b)的例子中, 允许从任意源点到任意目标的报文通过。正像标准访问列表一样, 对源地址、目的地址或源目地址都可以使用关键字 **any** 代替地址/反码组合 0.0.0.0 255.255.255.255。

扩展访问列表比标准访问列表更强大, 因为前者不仅检查报文的源地址, 而是检查所有有价值的内容, 但是任何事情都是有代价的。使用扩展表所要付出的代价是增加处理负担(图 B-8)。因为访问列表中每一行都需要检查报文中的多个字段, 因而会发生多次 CPU 中断。如果访问列表非常庞大或路由器很忙, 那么这个要求可能会对性能产生不利的影响。

尽可能保持访问列表的长度较小可以减轻路由器的处理负担。而且还要注意, 在匹配发生时, 指定的操作会被调用, 这时处理会停止。因此, 如果你能够使大多数匹配都发生在你所给出访问列表的前几行, 那么性能将会被提升。虽然这种方式并不总是可行的, 但是在设计访问列表时需要记住这一点。

作为练习, 请根据本节开始例子给出一个更精彩的访问列表配置。也就是说, 用尽可能少的行数重写访问列表, 但又不损失任何功能(提示: 一个相同功能的访问列表仅需要 3 行)。

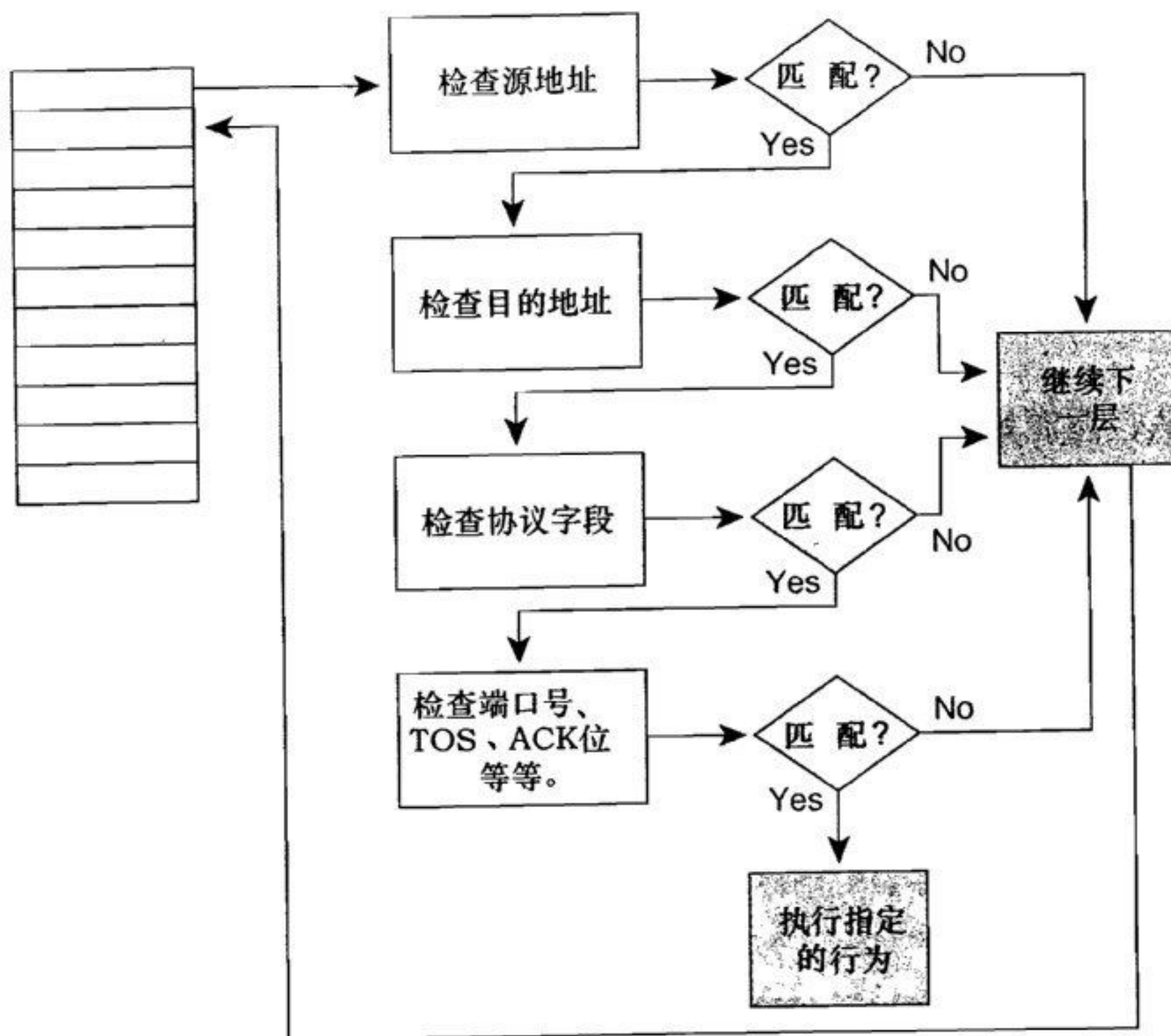


图 B-8 访问列表的决策流程图

B.3.1 TCP 访问列表

检查 TCP 端的扩展访问列表的行格式如下：

```
access-list access-list-number {deny|permit} tcp source source-wildcard
[operator port [port]] destination destination-wildcard [operator port [port]]
[established] [precedence precedence] [tos] [log]
```

注意：协议变量是 **tcp**。这里可能最重要的特性是，访问列表可以检查 TCP 段头中的源和目的端口号。其结果是，我们过滤报文的选项不仅是去往或来自一个特殊地址，而且还可以是去往或来自一个特殊管套（一个 IP 地址/应用端口的组合）。

- **operator** 指定逻辑操作。选项可以是 **eq**(等于)、**neq**(不等于)、**gt**(大于)、**lt**(小于)和 **range**（指明包括的端口范围）。如果使用 **range** 运算符，那么要指定两个端口号。
- **port** 指明被匹配的应用层端口号。几个常用的端口号是 Telnet(23)、FTP(20 和 21)、SMTP(25)和 SNMP(169)。完整的 TCP 端口号列表参见 RFC1700。

假设你实现了一个访问列表可以阻止外部发起的 TCP 会话进入到你的网络中，但是你又想让内部发起的 TCP 会话的响应通过，那应该怎么办？通过检查 TCP 段头内的 ACK 和 RST 标记，关键字 **established** 可以实现这一点。如果这两个标记都没有被设置，表明源点正在向目标建立 TCP 连接，那么匹配不会发生。最终报文将会在访问列表中的后继行中被拒绝。

TCP 访问列表的例子如下：

```
access-list 110 permit tcp any 172.22.0.0 0.0.255.255 established
```



```
access-list 110 permit tcp any host 172.22.15.83 eq 25
access-list 110 permit tcp 10.0.0.0 0.255.255.255 172.22.114.0 0.0.0.255 eq 23
```

第 1 行：如果连接是从网络 172.22.0.0 发起的，那么允许从任意源点到该网络的 TCP 报文通过。

第 2 行：允许来自任意源点，且目标端口号是主机 172.22.16.83 的端口 25(SMTP)的 TCP 报文通过。

第 3 行：允许来自网络 10.0.0.0，去向网络 172.22.114.0/24 且目标端口为 23(telnet)的 TCP 报文通过。

B.3.2 UDP 访问列表

检查 UDP 端的扩展访问列表的行格式如下：

```
access-list access-list-number {deny|permit} udp source source-wildcard
[operator port [port]] destination destination-wildcard [operator port [port]]
[precedence precedence][tos][log]
```

除了协议变量是 **udp** 外，该格式与 TCP 的格式非常相似。另一个不同之处是没有关键字 **established**。原因是 UDP 提供无连接传输服务，在主机之间没有建立连接。

通过向前面例子中添加额外的 3 行可以得到下面的例子：

```
access-list 110 permit tcp any 172.22.0.0 0.0.255.255 established
access-list 110 permit tcp any host 172.22.15.83 eq 25
access-list 110 permit tcp 10.0.0.0 0.255.255.255 172.22.114.0 0.0.0.255 eq 23
access-list 110 permit udp 10.64.32.0 0.0.0.255 host 172.22.15.87 eq 69
access-list 110 permit udp any host 172.22.15.85 eq 53
access-list 110 permit udp any any eq 161
```

第 4 行：允许从子网 10.64.32.0/24 到主机 172.22.15.87 且目标端口为 69 (TFTP) 的 UDP 报文通过。

第 5 行：允许从任意源点到主机 172.22.15.85 且目标端口为 53 (域名服务器) 的 UDP 报文通过。

第 6 行：允许从任意源点到任意目标的 SNMP 报文通过。

B.3.3 ICMP 访问列表

检查 ICMP 报文的扩展访问列表的行格式如下：

```
access-list access-list-number {deny|permit} icmp source source-wildcard
destination destination-wildcard [icmp-type[icmp-code]][precedence
precedence][tos][log]
```

icmp 目前在协议字段，注意这里没有源端口或目的端口，因为 ICMP 是一种网络层协议。该行可以用于过滤所有 ICMP 信息，或者读者可以使用下面的选项过滤指定的 ICMP 信息：

- **icmp-type** 编码范围是 0~255。所有 ICMP 类型编号见 RFC1700 以及本书表 2-5。
- 过滤粒度可以通过指定 **icmp-code** 进行增加。一个 ICMP 代码指定了 ICMP 报文类型的一个子集；

ICMP 访问列表的例子如下：

```
access-list 111 deny icmp 172.22.0.0 0.0.255.255 any 0
access-list 111 deny icmp 172.22.0.0 0.0.255.255 any 3 9
access-list 111 deny icmp 172.22.0.0 0.0.255.255 any 3 10
access-list 111 permit ip any any
```

第 1 行：拒绝从网络 172.22.0.0 到任意目标的 ICMP ping 响应（回应应答，ICMP 类型 0）通过。

第 2 行：拒绝从网络 172.22.0.0 到任意目标的 ICMP 目的网络不可达报文通过，其中代码号为 9。

第 3 行：拒绝从网络 172.22.0.0 到任意目标的 ICMP 目的网络不可达报文通过，其中代码号为 10。

第 4 行：允许所有其他 IP 报文通过。

B.4 调用访问列表

如果不通过调用命令使报文发送到访问列表，访问列表是不进行任何处理的，这里所用的命令定义了如何使用访问列表。命令如下所示：

```
ip access-group access-list-number {in|out}
```

在接口上配置这个命令可以建立安全过滤表或流量过滤表，并且可以应用于进出流量。如果 **in** 或 **out** 都没有被指定，那么缺省值是出站。当然，访问列表编号指定了接收该命令所发送报文的访问列表。图 B-9 给出了该命令的两个配置：

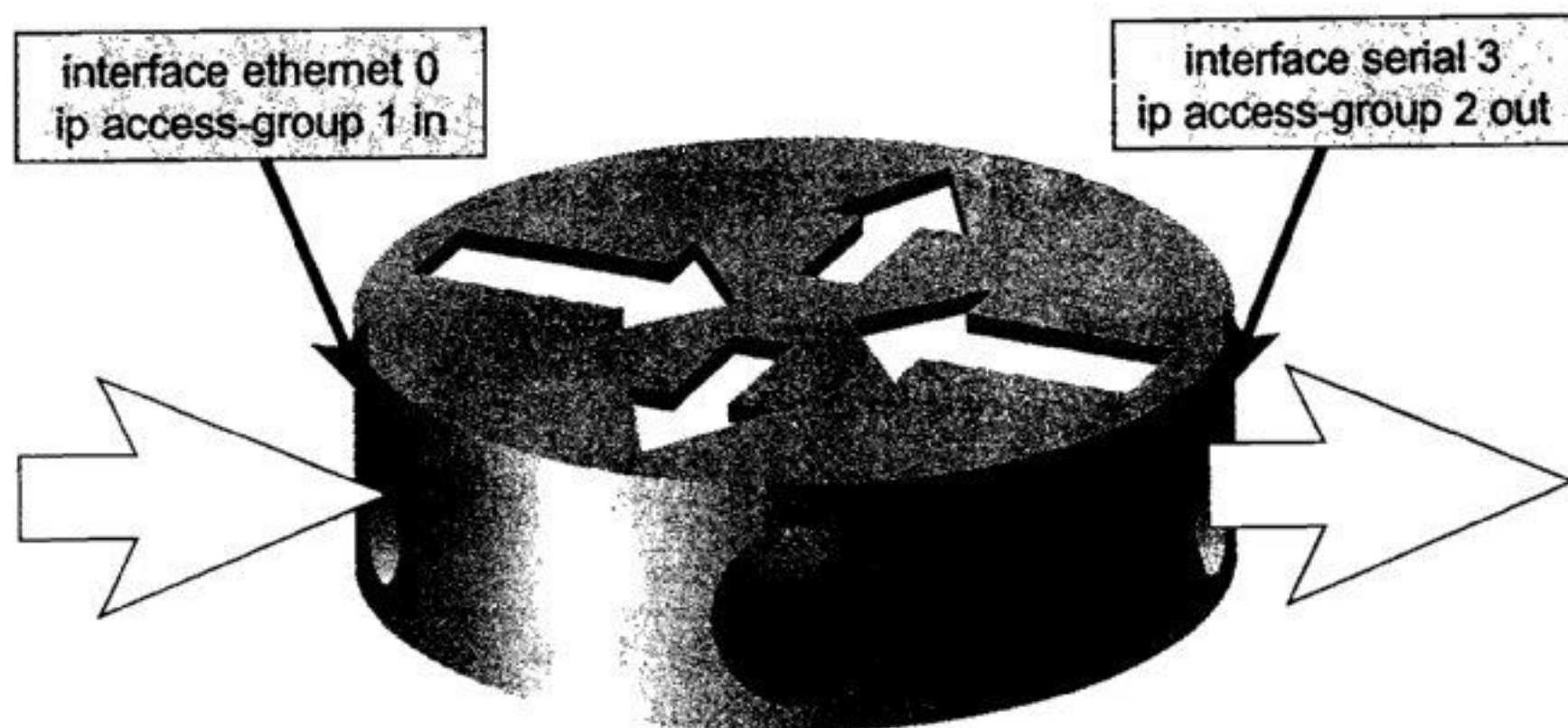


图 B-9 命令 **ip access-group** 使用指定的访问列表在接口上针对进出流量建立过滤器

图 B-9 中的访问列表 1 过滤进入接口 E0 的 IP 报文，它对于出站报文和其他协议（如 IPX）产生的报文不起作用。访问列表 2 过滤离开接口 S3 的 IP 报文，它对于入站报文和其他协议产生的报文不起作用。

多个接口可以调用相同的访问列表，但是在任意一个接口上，对每一种协议仅能有一个进入和离开的访问列表。

在图 B-10 中，前面例子中给出的 TCP、UDP 和 ICMP 访问列表被用作过滤器。源自前面两个例子中的访问列表 110 被应用到令牌环接口 0 上，检查入站流量。应用到相同接口的访问列表 111 检查出站流量。仔细分析一下两个访问列表，包括它们之间的相互关系，再考虑下面问题：

- 从 172.23.12.5 到 10.64.32.7 的 ping 响应报文想从接口 TO0 出站，是否允许通过？
- 想在主机 172.22.67.4 上 ping 网络 10.64.32.20 上的一台设备，可以 ping 通吗？

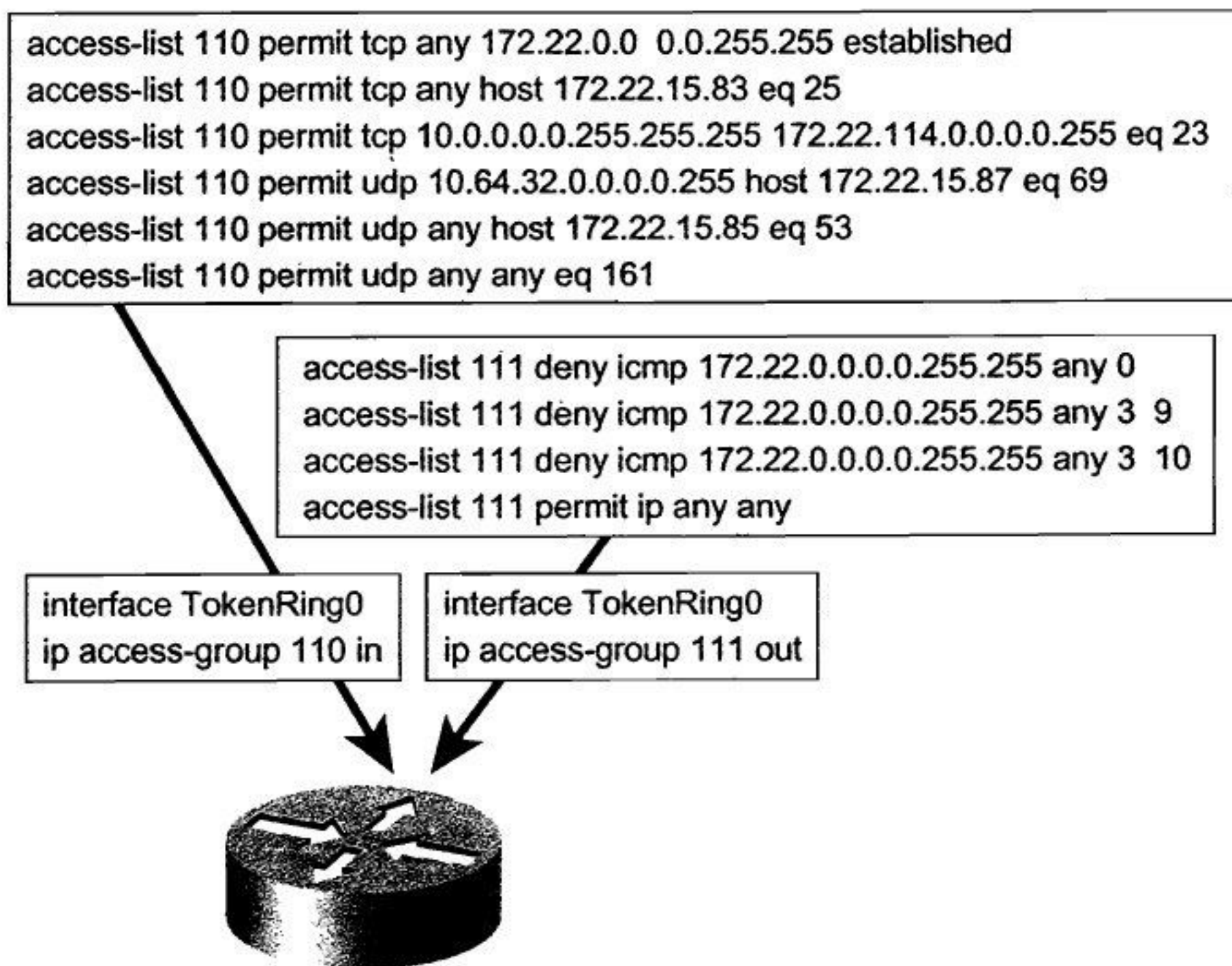


图 B-10 这里访问列表 110 用于过滤令牌环接口上的入站报文。访问列表 111 用于过滤该接口上的出站报文

另一种调用访问列表的命令是 **access-class**。该命令用于控制到达路由器或由路由器虚拟终端线路发起的 telnet 会话，而不进行报文过滤。命令格式如下：

```
access-class access-list-number {in|out}
```

图 B-11 给出了使用命令 **access-class** 的例子，访问列表 3 控制路由器 VTY 线路将要接受的 telnet 会话的源点地址。访问列表 4 控制路由器 VTY 线路可以连接的目标地址。

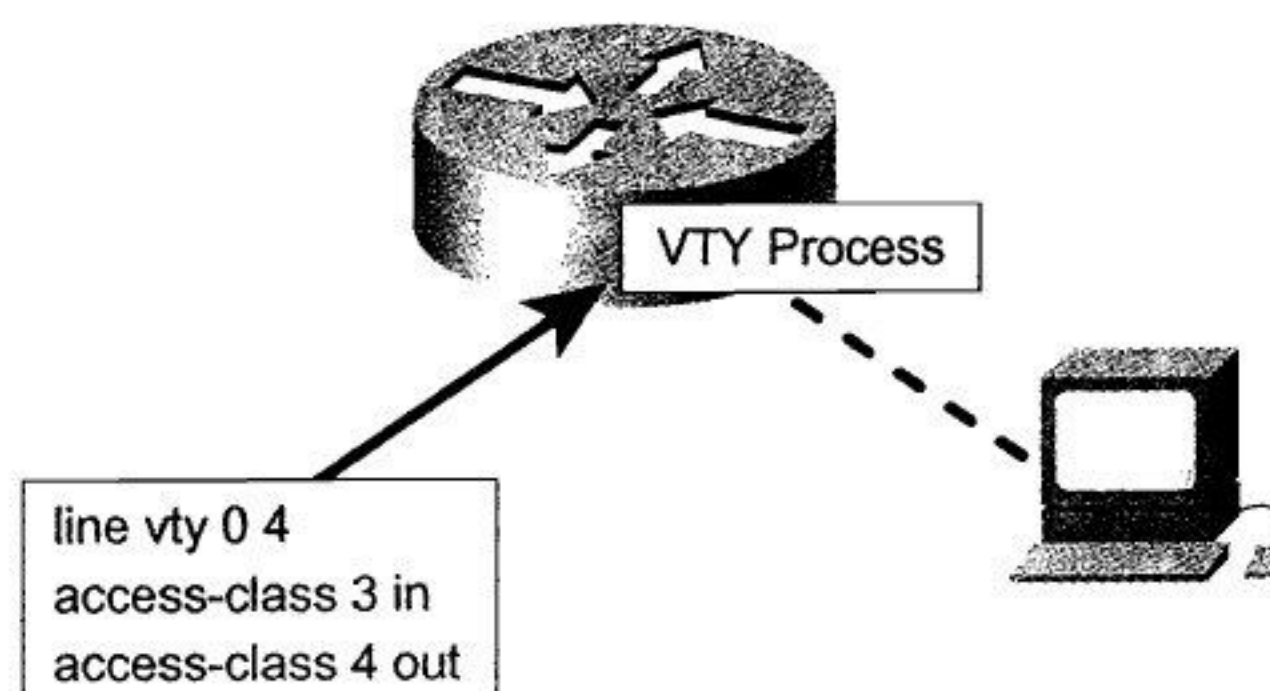


图 B-11 命令 **access-class** 使用访问列表控制到达和由路由器虚拟终端线路发起的 telnet 流量

命令 **access-class** 对于路由器传输的 telnet 流量不起作用, 它仅影响到达路由器以及由路由器发起的 telnet 会话。

B.5 可供选择的關鍵字

大多数网络专业人员都知道一些较常用的 TCP 端口号和一些 UDP 端口号。很少有人可以说出有关 ping 或目标不可达的 ICMP 类型是什么, 知道目标不可达类型的 ICMP 代码的人就更少了。从 IOS 10.3 开始, 配置访问列表可以使用关键字来代替端口、类型或代码编号。使用关键字, 访问列表 110 和 111 可以表述如下:

```
access-list 110 permit tcp any 172.22.0.0 0.0.255.255 established
access-list 110 permit tcp any host 172.22.15.83 eq smtp
access-list 110 permit tcp 10.0.0.0 0.255.255.255 172.22.114.0 0.0.0.255 eq telnet
access-list 110 permit udp 10.64.32.0 0.0.0.255 host 172.22.15.87 eq tftp
access-list 110 permit udp any host 172.22.15.85 eq domain
access-list 110 permit udp any any eq snmp
!
access-list 111 deny icmp 172.22.0.0 0.0.255.255 any echo-reply
access-list 111 deny icmp 172.22.0.0 0.0.255.255 any net-unreachable
      administratively-prohibited
access-list 111 deny icmp 172.22.0.0 0.0.255.255 any host-unreachable
      administratively-prohibited
access-list 111 permit ip any any
```

注意: 如果你将路由器从 10.3 以前的版本升级到新版本, 紧接着重启路由器, 那么路由器将会使用新的文法重写配置文件中的访问列表, 包括关键字。如果你随后又需要重载最初 10.3 之前的镜像文件, 那么路由器将无法理解被修改的访问列表。记住在任何情况下, 升级之前要将原来的配置文件上载到 TFTP 服务器上。

B.6 命名访问列表

对于每个路由器, 99 个标准访问列表或 100 个扩展访问列表的限制看上去已经够用了, 但是有些情况 (比如动态访问列表¹) 可能就不够了。从 IOS 11.2 开始提供的命名访问列表打破了这种限制, 它的另一个优点是描述名可以使数量庞大的访问列表更加便于管理。

为了使用命名访问列表, 访问列表的第 1 行应采用以下格式:

```
ip access-list {standard|extended} name
```

因为没有编号区分访问列表类型, 所以该行可以将表指定为标准 IP 表或扩展 IP 表。下面紧接着可以使用许可或拒绝表述, 标准表的文法如下:

```
{deny|permit} source [source-wildcard]
```

¹ 本教程中没有涉及动态访问列表。详细内容可参见 Cisco 文档。

基础扩展表的文法如下：

```
{deny|permit} protocol source source-wildcard destination destination-  
wildcard [precedence precedence][tos tos][log]
```

在这两种情况下，都没有出现命令的 **access-list access-list-number** 部分，但是其他部分完全相同。在相同路由器上标准和扩展访问列表不能同名。除了在接口上建立命名访问列表的命令涉及到用名字替代编号外，在其他方式中命令均保持不变。图 B-12 使用命名格式对图 B-10 中的访问列表进行转换。

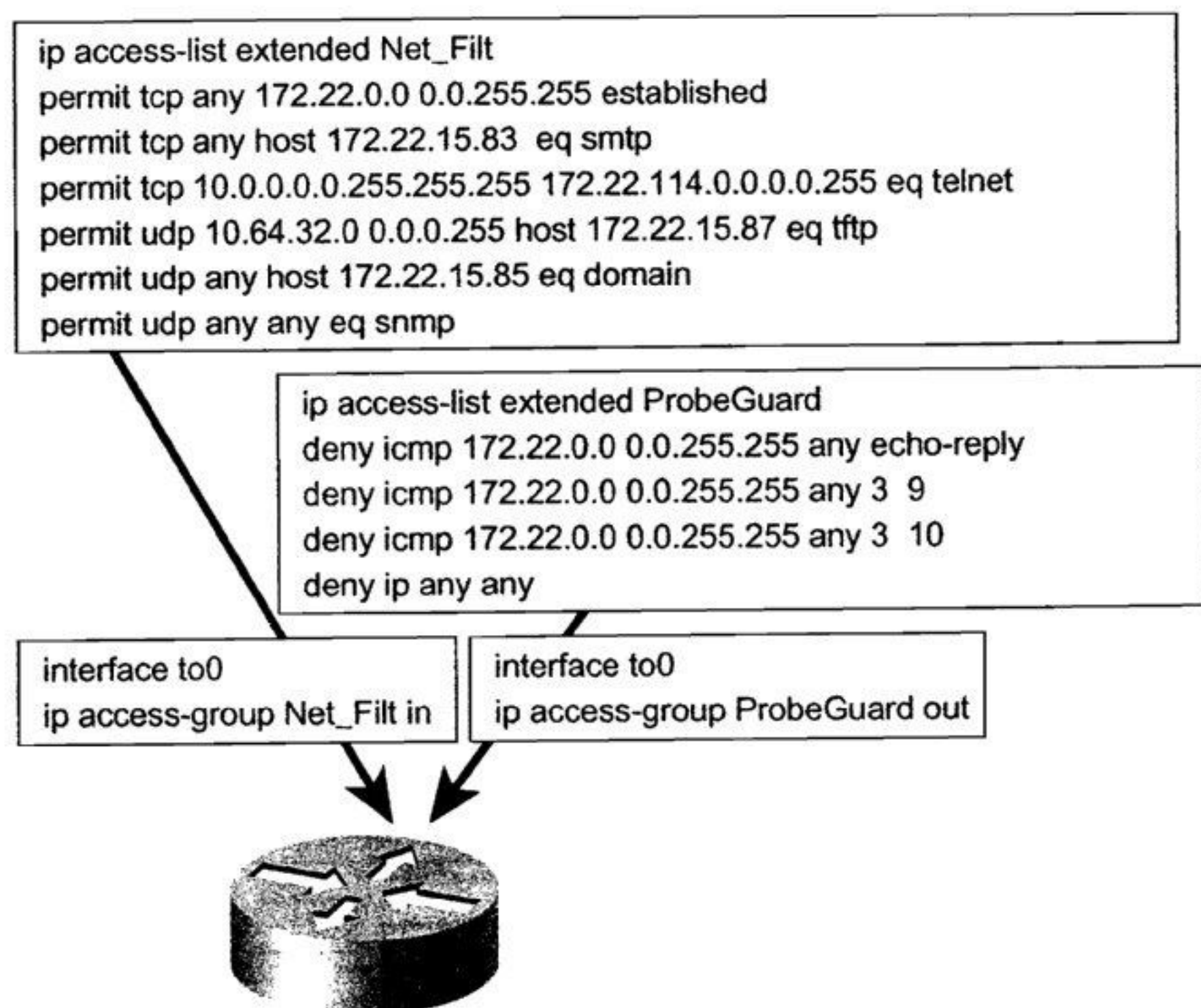


图 B-12 图 B-10 中的访问列表现在被配置为命名访问列表

B.7 对过滤表放置的考虑

为了达到最佳的性能，你不仅需要考虑到访问列表自身的有效设计，而且还要考虑过滤表在路由器和互连网络中如何去布置。

凭经验估计，安全过滤通常是作为入站过滤器。在不必要或不被信任的报文到达路由进程之前将它们过滤掉，阻止欺骗攻击——报文欺骗路由进程，使其认为它来自某处，但它实际并不是来自那里。另外一方面，流量过滤通常是出站过滤器。当你考虑在某一点的流量过滤器可以阻止不必要的报文占用某条数据链路时，这种方法将很有意义。

除了这两种凭经验估计的情况外，要考虑的另一个因素是访问列表和路由器进程将要使用的 CPU 周期数。入站过滤表是在路由进程之前被调用，而出站过滤表是在路由进程之后被调用（图 B-13）。如果大多数经过路由进程的报文被访问列表拒绝，那么入站过滤表可以节省一些处理周期。

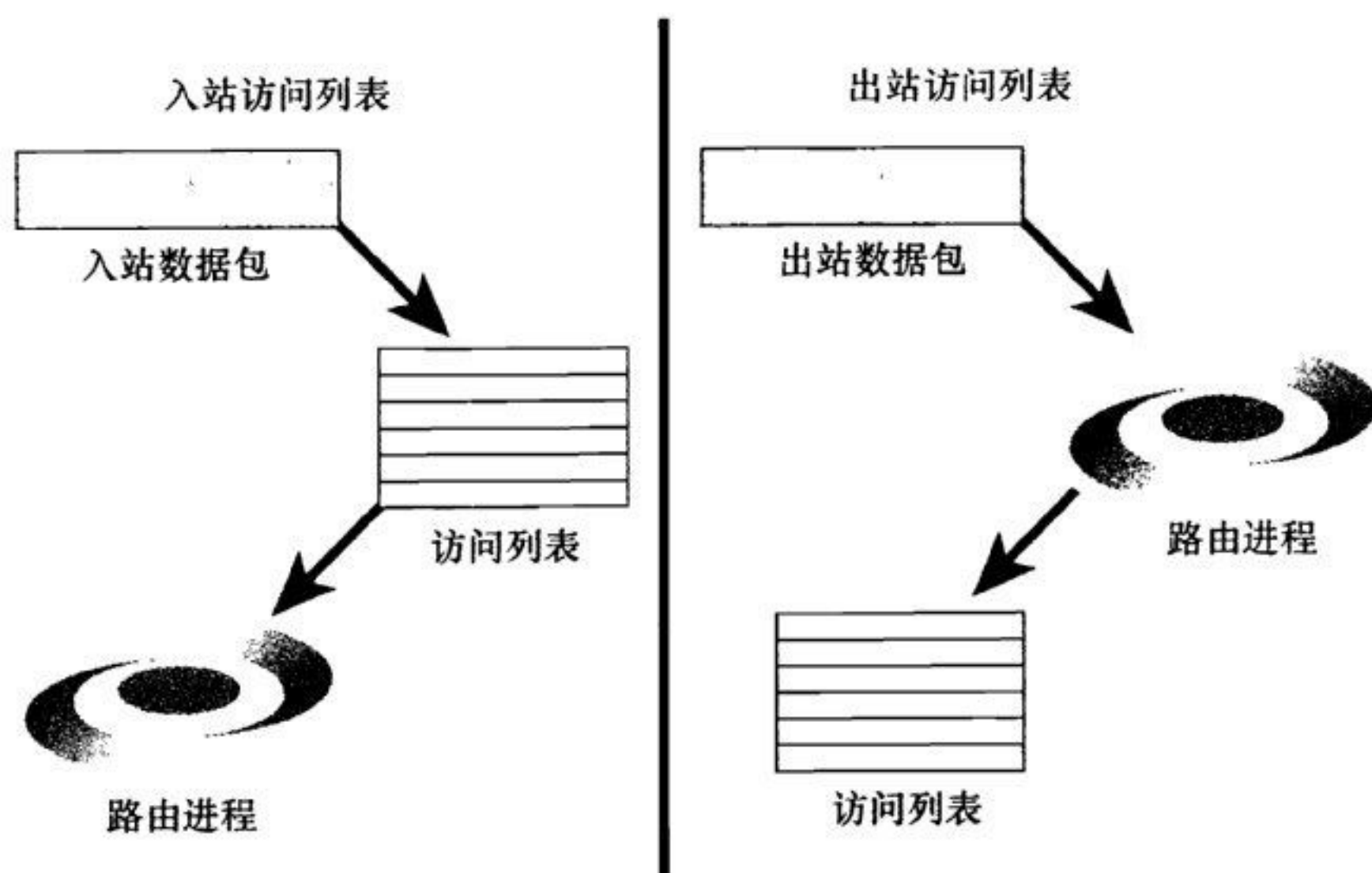


图 B-13 入站过滤表是在路由进程之前被调用，而出站过滤表是在路由进程之后被调用

标准 IP 访问列表仅能够过滤源地址。因而，使用标准表的过滤器必须被放置在尽可能靠近目标的地方以便源点还可以访问到其他没有被过滤的目标（图 B-14(a)）。其结果是带宽和 CPU 周期被浪费在最终将要被丢弃的报文身上。

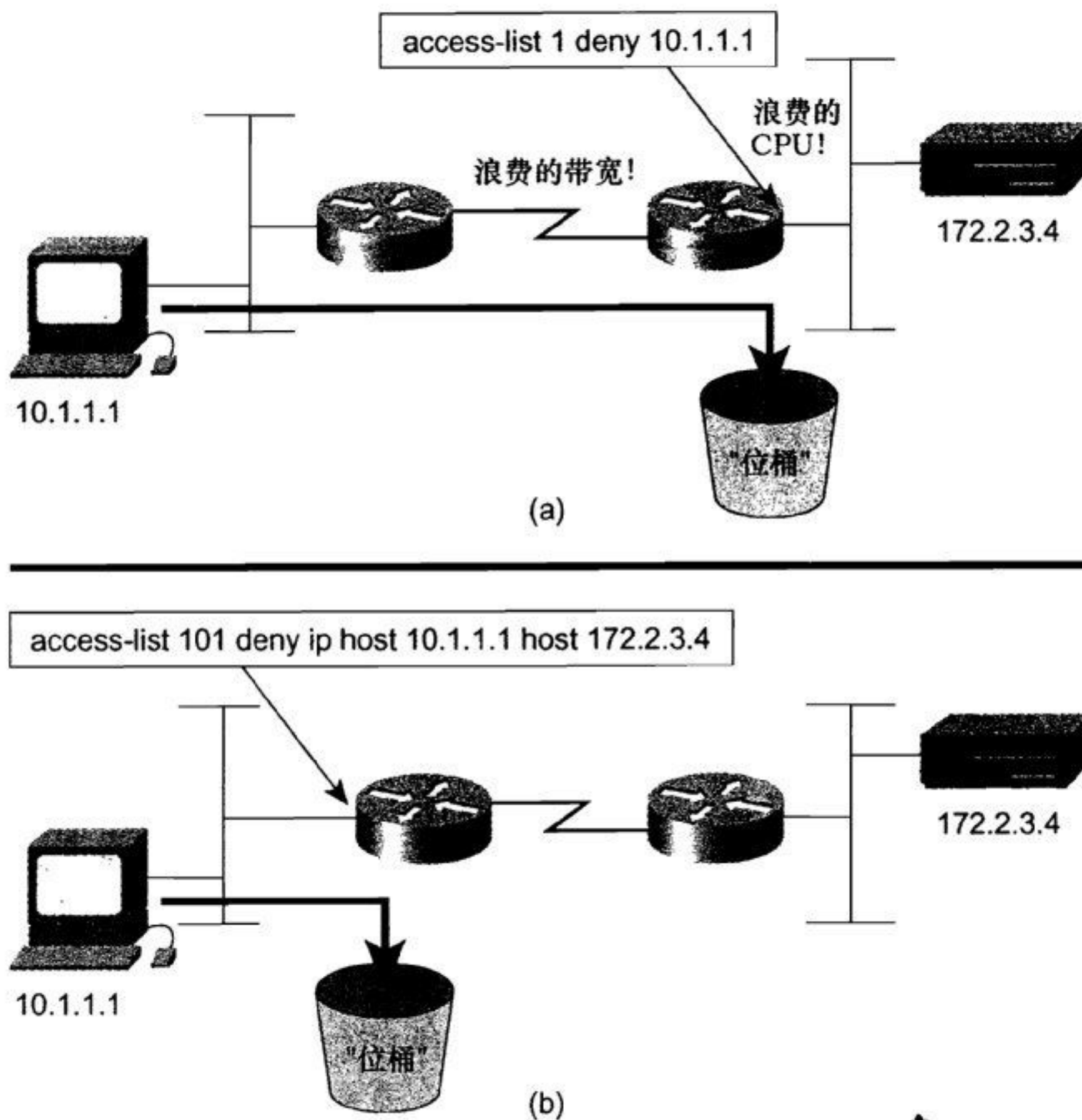


图 B-14 使用标准表的过滤器必须被放置在尽可能靠近目标的地方(a)，而扩展访问列表可以放置在靠近源点的地方(b)

因为扩展 IP 访问列表可以非常准确地标识报文特性，所以它们应该被放置在尽可能靠近

源点的地方，这样可以防止带宽和 CPU 浪费在传输无用的报文上（图 B-14(b)）。另一方面，扩展表的复杂性意味着更多的处理负担。当决定在那里放置过滤表时，需要权衡考虑。

你还必须理解访问列表将怎样影响路由器上的交换。例如，使用扩展访问列表的接口不能被独立地交换；动态访问列表不能被硅交换，而且可能影响硅交换的性能。在 11.2 之前 IOS 根本不支持命名访问列表。

在骨干或核心路由器上，访问列表对交换的影响可能是很重要的。通过阅读 Cisco 关于 IOS 的配置指南可以确保全面地研究和理解访问列表所产生的影响。在某些情况下，一个报文过滤路由器——一个专门用于报文过滤的较小的路由器——可能被用来减轻重要路由器的负担。

B.8 访问列表的监视和计费

在不必显示路由器的整个配置的前提下，能够检查一个甚至所有访问列表是非常有用的。命令 **show ip access-list** 可以显示路由器上所有 IP 访问列表的简化文法。如果要观察一个特定的访问列表，可以通过指定名字和标号（图 B-15）。如果你没有输入关键字 **ip**（**show access-list**），那么所有访问列表都将被显示出来。

```
Woody#show ip access-list 110
Extended IP access list 110
  permit tcp any 172.22.0.0 0.0.255.255 established
  permit tcp any host 172.22.15.83 eq smtp
  permit tcp 10.0.0.0 0.255.255.255 172.22.114.0 0.0.0.255 eq telnet
  permit udp 10.64.32.0 0.0.0.255 host 172.22.15.87 eq tftp
  permit udp any host 172.22.15.85 eq domain
  permit udp any any eq snmp
Woody#
```

图 B-15 命令 **show ip access-list** 可以显示访问列表的简化文法

跟踪被访问列表拒绝的报文作为安全计划或编制策略能力的一部分，这是很有用的。命令 **ip accounting access-violations** 可以配置在独立的接口上，用来建立所有在该接口被访问列表拒绝的报文的数据库。可以使用命令 **show ip accounting access-violations** 来检查数据库，结果可以显示出匹配该地址的报文数、字节数以及拒绝该报文的访问列表编号（图 B-16）。命令 **clear ip accounting** 可以清除计费数据库。

```
Woody#show ip accounting access-violations
Source          Destination          Packets          Bytes          ACL
10.1.4.1         255.255.255.255     13              936            110
10.1.4.1         172.22.1.1          12              1088           110

Accounting data age is 10
Woody#
```

图 B-16 访问列表计费数据库可以使用命令 **show ip accounting access-violations** 来查看

计费功能将关闭接口上的独立交换功能和硅交换功能。在需要这些交换模式的接口上不能使用计费功能。

作为最后一个窍门,你应该知道被表尾隐含的拒绝一切规则所丢弃的报文,它们是不能被计费跟踪的。为了跟踪这些报文,仅需在表尾配置一条拒绝一切的规则:

```
access-list 110 permit tcp any 172.22.0.0 0.0.255.255 established
access-list 110 permit tcp any host 172.22.15.83 eq smtp
access-list 110 permit tcp 10.0.0.0 0.255.255.255 172.22.114.0 0.0.0.255 eq telnet
access-list 110 permit udp 10.64.32.0 0.0.0.255 host 172.22.15.87 eq tftp
access-list 110 permit udp any host 172.22.15.85 eq domain
access-list 110 permit udp any any eq snmp
access-list 110 deny ip any any 1
```

本教程没有设计动态访问列表,详细内容参见 Cisco 配置文档。

附录 C

CCIE 小提示

要成为 Cisco 认证互联网络专家 (CCIE) 远远不像其他一些工业认证, 只是“读一本书, 参加一次考试”就行了。你需要在一场极其艰难的实验室实践考试中展示你的专业才能。虽然你必须对 Cisco 的配置命令很熟悉, 但实验室考试中最大的挑战是与 Cisco 无关的, 他们测试的是你对交换机、路由器以及路由选择协议的理解深度。正因为如此, CCIE 们才被认为是千里挑一的互联网络专家。

有层次地构建互联网络包括 4 个步骤。这 4 个步骤同样也适用在准备 CCIE 实验室考试的过程中。

计划: 冷静、认真地看待你现在的经验水平和不足之处。算一算你每天能用来学习的时间。检查你的资源, 包括实验室设备、资金、书籍、训练的时间以及能请教问题的熟人、老师、专家。衡量你自身的长处和短处: 你擅于考试吗? 你在压力下能很好地工作吗? 你对挫折和失落的反应是什么? 你学习习惯好吗? 你能很好地通过阅读来学习吗? 好好利用以上的答案, 写一个列表, 制定一个计划, 把你的长处发挥到极至, 最大限度地去掉你的短处。

设计: 设计一个满足你需要、符合你的时间表和资源的个人计划。与尽可能多的 CCIE 们交流, 问问他们的准备方案。找出哪些有用哪些没用。你的计划应该能在确定的时限内分阶段把你现在的水平提升至 CCIE Lab 的水平。通过一系列具有明确目标的小工程来架构一个大的项目。实实在在去编制时间表, 充分考虑你的工作和个人生活中的可能性。你的雇主和家庭是否支持对于决定你的准备计划应该紧张或是松弛是一个重要的因素。不要超过你身边事物的容忍限度, 否则你的计划会受到很大影响。

实现：有很多失败的例子是因为在设计完成之前就忙于实施。应该清楚地写下你的准备计划，规划好方案的每一步，而不要草率实施。一旦你开始了，就要坚持不懈。不要放弃，也不要气馁，更不要懒惰。检查核对你要完成的每一步。

优化：你的准备计划应是一个鲜活的文档。当你前进时，会遇到比你的预计或困难或简单的问题。尽管继续向前，但要灵活机动，增加额外的任务来帮助你掌握每一个主题。

只有你才能设计最适合自己的准备计划。在下面的几节里，我给出的建议并不是叫你死板地遵循，而是给你一些点子，让你创建自己的学习计划。一些小的技巧来源于我作为一个 CCIE 和 Cisco 系统讲师的经验，也来源于我的那些成功通过 CCIE 考试的同事们的经验。

C.1 牢固的基础

如果你是一个初学者，或者你的联网经验有限，你的第一步就是牢牢掌握网络互联和 Cisco 路由器的基础部分。这个过程包括课堂训练和自学。

Cisco 通过它的培训伙伴提供了很多实践培训课程。只要你的时间和资源允许，尽可能多地参加这些培训。其中一些培训特别重要：

- Cisco 路由器配置入门 (Introduction to Cisco Router Configuration , ICRC)
- Cisco 路由器高级配置 (Advanced Cisco Router Configuration , ACRC)
- Cisco 局域网交换配置 (Cisco LAN Switch Configuration , CLSC)
- Cisco 互联网络故障排除 (Cisco Internetwork Troubleshooting)

充分利用你参加的每一节课，多问老师问题，多和同学们一起讨论。最重要的是，好好利用你接触设备的机会。不要光满足于表面，一定要充分理解“为什么”和“如何”。当你做完一个实验，不要停止，多多摸索设备，看看有哪些配置和调试命令，多尝试使用它们。如果有时间，多配几次直到熟练。

上课会让你找出知识上的不足。多读书，填补你对网络互联协议和技术方面知识的空白。Internet 上有很多好的技术指南，当你开始学习一个新的知识点，一定要记得在 Web 上搜索一通。

C.2 实践经验

几乎所有的 CCIE 都会告诉你，实践经验在准备实验室考试的过程中是无法衡量的一个部分。决不要放过任何一个配置和调试路由器的机会。如果你现在的工作无法接触到路由器或者交换机，那么和你们公司的网络工程师打好交道，向他们说说你的目标，尽可能地在工作上帮助他们。

如果你能接触实验室设备，就一定要充分利用它们。在实验室取得的经验是不可替代的，在实验室，你可以随心所欲地进行配置，引入各种各样的问题，而不用担心实际环境中的网络崩溃。

在一些大城市，Cisco 的办事处可能会提供网络实验室，你可以预定使用，具体情况请咨询本地的 Cisco 机构。

还有一个选择, 就是建立自己的实验室。虽然代价昂贵, 但作为一个 CCIE, 你所拿的薪水会让你的投资非常划算。许多旧 Cisco 设备的价格是很合理的。请订阅 Cisco 新闻组, comp.dcom.sys.cisco; 经常有人在上面卖旧路由器, 你或许可以与他们做上一笔划算的买卖。你至少需要 4 台路由器, 其中一个应该有 4 个或者更多的串口, 这样你可以配置它为帧中继或者 X.25 交换机。记住你不需要顶级设备, 过时的路由器就足够了。例如, 你可以以 800\$ 到 1500\$ 的价格买到 AGS+, 如果你能忍受它的噪音, 它会是一台非常优秀的实验室路由器。

这本书中的所有配置都是基于 5 个 2500 系列路由器以及一个具有以太口、令牌环、串口和 FDDI 接口的 AGS+ 路由器。

C.3 深入学习

有了基础知识, 同时有实际的动手经验, 你应该开始深入学习互联网络协议了。至少你应该阅读本书中列出的 RFC, 相关的 RFC 读得越多越好。Internet 上很多站点都可以获得它们。www.ietf.com 是最好的站点之一。

当然, 不是所有的互联网络协议在 RFC 中都有描述。寻找更高级的非 IP 协议, 譬如 SNA、AppleTalk、IPX 和 Banyan VINES 的教程、白皮书和指南。你也应该学习 Ethernet、Token Ring、FDDI 和 WAN 协议, 比如 T-1、ISDN、X.25、Frame Relay 和 ATM。你可以在 www.cisco.com 上找到非常多的资料。

一个很好地结合了理论和实践知识的学习工具是 Cisco 新闻组 comp.dcom.sys.cisco。在上面提交你遇到的具有挑战性的难题, 你先自己解答, 然后等待由 CCIE 们以及 Cisco 工程师们给出的答案, 看看你的答案是不是正确, 如果不正确, 找出问题所在。如果你在 Cisco 的 CCO 上注册了, 还有一个好的信息园地是开放论坛 www.cisco.com/openf/openproj.shtml。

最后, 如果你找到一些志同道合者, 组成一个学习小组吧。在 International Network Services 中, CCIE 学习小组非常有效, 从中诞生了很多 CCIE。

C.4 最后 6 个月

具有坚实的理论和实践背景, 你最后 6 个月的准备应该包括仔细阅读“Cisco IOS 配置指南”。阅读每一章时, 回顾“Cisco IOS 命令参考”中的相关章节, 确保对于每一个协议, 你已经熟悉 IOS 中的所有配置命令。然后在你的实验室中用不同的方法来配置章节中所覆盖的协议。尝试不同的假设, 并使协议运行在不常见的环境下。你会发现, 当一个配置如你所料运行时, 你并不能得到最好的经验, 在与之相反的情况下, 你却往往能学到更多的东西。

为你的配置和想法作笔记, 记下它们是如何工作或者不工作的。记得去发掘记录所有与协议有关的调试工具, 譬如 **debug** 和 **show** 命令。

在每一章的最后, 你的目标是至少配置协议的最本质部分。当你参加 CCIE 实验室考试时, 动很少的脑筋去配置相对简单的部分, 抽出时间去思考困难的问题, 这是很重要的。你也应该了解一个协议是怎样运作的, 存在哪些配置选项, 以及怎样调试该协议。当你达到每一章的目标之后, 进入下一章。

最后 6 个月的开始, 你应该参加 CCIE 的笔试。不要被这个笔试所迷惑。它只是想淘汰那些完全没有准备的人, 好让他们仅仅损失参加笔试所花的钱, 而不是浪费参加实验室考试所需的更大的费用。如果你好好准备了, 你会发现其实笔试并不是很难。

一些 Cisco 培训伙伴提供 CCIE 的准备课程, 给你一些在类似 CCIE 实验室考试的条件下做实验的经验。不要以这些课程取代勤奋的学习和实践。如果你选择参加这些课程, 你首先应该认真准备了 CCIE 考试。从这些商业培训中得到的最大的好处就是“在枪口下”的感觉——在紧迫的时间限制里解决困难问题。

C.5 参加考试

CCIE 实验室考试不仅仅测试你的实践和理论知识, 而且测试你在压力下运用这些知识的能力。既然已经来了, 就不要无谓地为自己再增加压力。

- 大多数人第一次参加 CCIE 实验室考试都会失败, 因此要有参加第二次考试的准备。这能帮助你在参加第一次考试时保持冷静, 或许, 你就可能一次成功。
- 安排好你的行程, 提前到达考场。考试之后再安排回去的行程。你显然不愿意考试期间考虑行程安排。
- 考试的前一晚确定考场的位置, 不要因为迷了路慌慌张张出现在考场上。
- 考前一晚最多做做小复习, 如果你想填鸭式的恶补, 只能使自己焦虑不安、失眠。
- 吃一顿好的晚餐, 不要喝酒, 美美睡一觉。
- 考试当天, 吃一顿好的早餐。吃得好会帮你表现更出色, 这是事实。
- 穿舒服一点, 穿得好不会给你加分。

考试之前, 会要求你在一张表格上签名, 表明你不会泄漏考试的题目。考试长达 16h, 分两天完成 (午餐每天都是安排好的)。虽然其他的考生在同一个房间里考试, 但你得独立完成。通常, 其他考生的题目和你的是不同的。前一天半 (12h) 会让你根据要求构建一个互联网络。一定要记录下你的工作, 因为你可能会被要求解释你的设计和配置。

通过配置部分之后, 你可以去吃午饭。在此期间, 会在你构建的网络中人为设置一些故障。你有 4h 的时间来排除, 记录这些故障。记住, 一定要记录下你调试的过程。和前面一个部分一样, 你会被要求解释你的方法。

在考试的任何一个部分, 如果你不明白某个特定的要求, 不要犹豫, 去问监考官, 他会帮助你。最重要的是, 放松、集中注意力。

当你通过 CCIE 考试, 你就完成了值得你万分骄傲的一件事。如果本书或者附录中的一些建议曾经帮助你达成你的目标, 请 email 我, 让我共享你的骄傲。

附录 D

复习题答案

第 1 章

1. 局域网的基本目标是资源共享。资源可以是设备、应用程序或者信息。共享资源的例子有文件、数据库、email、调制解调器和打印机。

2. 协议指的是一组被广泛认可的规则。在数据通信中，这些规则通常规定了一个过程或者数据的格式。

3. 介质访问控制（MAC）协议定义了 LAN 介质的共享方式，连接到介质的设备的标识方式，以及传输到介质的帧的格式。

4. 帧可以看作是数字“信封”，它提供足够的信息，以便在数据链路上传送数据。帧的典型组成部分有：数据链路上源的地址、目标的地址、帧中封装的数据类型，以及差错校验信息。

5. 所有类型的帧都有一个共同特征，即有一个标识数据链路上设备的格式。

6. MAC 地址提供了一种手段，使得在数据链路上的单独的设备能够被惟一标识，这样它们之间可以互相传输数据。

7. 地址定义一个特定的位置。但 MAC 地址不是一个真正的地址，因为它被永久分配给了一个指定设备的接口，并且随设备移动。MAC 标识了设备，并不是这个设备的位置。

8. 数据链路上的信号退化有 3 种原因：衰减、干扰和失真。衰减是由于介质电阻的物理特性所致；干扰是由于进入介质的噪声所致；失真是由于介质的反应特性所致：对于不

同频率的信号, 它的反应也不同。

9. 中继器的作用是扩展物理介质的传输范围: 它读出一个已经退化的信号, 然后“干净”地复制它。

10. 网桥的作用是增加局域网的容量。网桥把数据链路分成段, 并且仅仅转发在一个段中产生而去往另外一个段的流量。通过控制和限制数据链路上的流量, LAN 上可以连接更多的设备。

11. 透明网桥在它每一个端口上侦听所有报文。就是说, 它检查在它所连接的所有介质上的所有帧。它在一个桥接表中记录帧的源地址, 以及收到该帧的端口号。以后, 它就参考该表以决定是否转发或者丢弃一个帧。我们说网桥是透明的, 是因为它的这种学习机制独立于产生帧的设备。终端设备本身并不知道网桥的存在。

12. 局域网和广域网的 3 个基本区别是:

- 局域网局限于比较小的地理范围之内, 譬如一栋楼或者一个校园。广域网覆盖了大的地理区域, 从一个城市到世界范围。
- 局域网内的设备通常都是完全私有的。而广域网一些部分, 比如交换网络或者点到点串行链路, 通常是向服务提供商租用的。
- 局域网提供高带宽并且费用相对低廉。跨越广域网的带宽则要昂贵得多。

13. 一个帧的目标地址是 MAC 广播地址时, 表明了数据将传送到连接至数据链路的所有设备。这个广播地址是二进制全 1, 对十六进制, 是 FFFF.FFFF.FFFF。

14. 网桥和路由器的最基本共同点是, 它们都增加了可连接至一个通信网络中的主机数量。它们的区别在于网桥对一个网络中的独立网段进行互连, 而路由器连接不同的网络。

15. 数据包标明了数据从一个网络传送到另一个的方式。帧和包的相同点是它们都封装了数据, 并且提供了一种寻址机制。不同点在于帧在两个共享数据链路的设备间传送数据, 而包通过逻辑通路, 或者叫路由, 横跨多个数据链路来传送数据。

16. 从源到目的的过程中, 包的源地址和目的地址都不变。

17. 包使用网络地址。每个地址有网络部分, 定义了特定的数据链路, 而主机部分则定义了由网络部分定义的数据链路上的一个特定设备。

18. 包从整个互联网络的视角标识一个设备, 而帧从一个单独的数据链路的角度去标识设备。由于横跨互联网络的两台设备之间的连接是逻辑的, 所以网络地址是逻辑地址。由于跨越数据链路的两台设备之间的连接是物理的, 所以数据链路标识符是物理地址。

第 2 章

1. TCP/IP 协议族的 5 个层是:

- 物理层
- 数据链路层
- IP 层
- 主机至主机层
- 应用层

物理层——包含关于物理介质的协议。

数据链路层——包含了如何控制物理层的一些协议：介质是怎样存取和共享的，介质上的设备是怎样标识的，数据在介质上传播之前是怎样成帧的。

IP 层——包含一些将数据链路逻辑分组至互连网络以及在互连网络之间进行通信的协议。

主机至主机层——包含一些定义并控制互连网络中逻辑的、端到端的路径的一些协议。

应用层——对应于 OSI 中的会话层，表示层以及应用层。

2. 现在使用最普遍的 IP 版本是版本 4。

3. 当包的长度超过它所要去的那个数据链路的 MTU (Maximum Transmission Unit) 时，路由器要将它分段。包中的数据将被分成小段，每一段被封装在独立的包中。接收端使用包标识符 (Identifier)、分段偏移以及标记 (Flag) 域的 MF 位来进行重组。

4. TTL 域防止丢失的报文在 IP 互连网络中无休止地传播。该域包含一个 8 位整数，此数由产生报文的主机设定。报文每经过一个路由器，TTL 值将被减 1。如果一个路由器将 TTL 减至 0，它将丢弃该报文并送出一个 ICMP 超时消息给报文的源地址。

5. IP 地址的分类如下：

- A 类：第一个 8bit 字节的第 1 位是 0
- B 类：第一个 8bit 字节的前 2 位是 10
- C 类：第一个 8bit 字节的前 3 位是 110
- D 类：第一个 8bit 字节的前 4 位是 1110
- E 类：第一个 8bit 字节的前 4 位是 1111

6. A、B、C 类 IP 地址用点分十进制以及二进制表示如下：

类	第一个 8bit 字节 二进制范围	第一个 8bit 字节 十进制范围
A	00000001-01111110	1-126
B	10000000-10111111	128-191
C	11000000-11011111	192-223

7. IP 地址掩码定义了 IP 地址的网络部分。32 位掩码中的 1 标识了 IP 地址中相应的网络位，0 标识了主机位。将 IP 地址和掩码进行布尔运算 AND；结果是，IP 地址中对应于掩码网络部分的那一段不变，而对应于主机部分的全变成 0。

8. 子网化是对 A、B、C 类地址进行子分组。如果没有子网化，A、B、C 类的主 IP 地址（譬如 C 类的 202.112.25.0）的网络部分将只能标识一个数据链路。子网化使用主 IP 地址中的一些主机位作为网络位，允许一个单独的主地址被划分为多个网络地址。

9. 有类别路由选择协议不能区分全 0 子网和主 IP 地址，也不能区分全 1 子网和主 IP 地址的全主机、全子网广播地址。

10. ARP（地址解析协议）的作用是将数据链路上接口的 IP 地址映射到相应的 MAC 地址。

11. 代理 ARP 是 IP 路由器的功能之一。如果路由器收到一个 ARP 请求，并且

- 目标网络或子网在路由器的路由选择表中，且
- 路由选择表指出目标可以通过某一个接口可达，该接口不同于接受到 ARP 请求的那个接口，那么

- 路由器将用自己的 MAC 地址对该 ARP 请求进行回应

12. 重定向是 IP 路由器的功能之一。如果一个设备送给路由器一个报文, 而且该路由器必须转发该报文至同一数据链路上的下一跳路由器, 那么该路由器将送出一个重定向消息, 通知源设备说: 你能直接到达下一跳路由器, 不必经过我了。

13. TCP 在无连接的 IP 层之上提供了面向连接的服务。UDP 则提供了无连接服务。

14. 序列号保证了准确的排序; 校验、确认、计时器以及重传机制保证了可靠性; 滑动窗口机制保证了流量控制。

15. MAC 地址是固定长度的二进制整数。如果用 MAC 地址作为 IP 地址的主机部分, 那么子网化将不能得以实现。因为不可能灵活地使用一些主机位作为网络位。

16. UDP 头的惟一目的是增加源端口及目的端口号。

第 3 章

1. 路由选择表中的每一个表项至少要包括目标地址和下一跳的路由器地址, 或者表明目标地址是直接相连的。

2. 这意味着路由器知道, 对于相同的主 IP 地址的不同子网, 有多个子网掩码。

3. 非连续子网指的是被一个不同的主 IP 地址分隔开的一个主 IP 网络地址的两个或多个子网。

4. **show ip route** 命令用来查看 Cisco 路由器的路由选择表。

5. 第一个数字是学习到该路由的路由选择协议的管理距离, 第二个数字是该路由的度量值。

6. 在静态路由选择表项中, 如果使用本地接口来代替下一跳的地址, 目标地址将作为直接连接的地址进入路由选择表。

7. 汇总路由是一个单独的路由选择表项, 指向多个子网或主 IP。对于静态路由, 汇总路由能减少需要配置的静态路由选择表项。

8. 管理距离是一个路由选择协议或者静态路由的优先等级。每一个路由选择协议和静态路由都有管理距离值。当一个路由器从多个路由选择协议得知到达同一目标地址的多个路由选择表项, 它将使用管理距离最小的那条路由。

9. 浮动静态路由是到达目标地址的备用路由。它的管理距离被设得很高, 这样只有当别的优先级高的路由均不可用时, 它才被派上用场。

10. 等价负载均衡把流量分布在具有相同度量值的多条路径上; 非等价负载均衡把流量分布在具有不同度量值的多条路径上。流量将根据路由代价分配, 代价高的分配得少, 代价低的分配得多。

11. 如果一个接口是快速交换, 将执行每目标负载均衡; 如果一个接口是处理交换, 将执行每报文负载均衡。

12. 当路由器需要转发报文, 但通过单一路由选择表查找却得不到它所需的所有信息, 则会发生递归路由选择表查找。例如, 路由器执行一次查找得到去往一个目标的路由, 下一跳是路由器 A, 但它却不知道该如何到达 A, 于是执行另外一次查找得到去往 A 的路由。

第 4 章

1. 路由选择协议是路由器之间所讲的一种“语言”，用来共享网络目标地址的信息。
2. 一个路由选择协议至少要定义以下过程：
 - 传递网络的可达信息给其他路由器；
 - 从其他路由器接收可达信息；
 - 根据它所拥有的可达信息决定最佳路由，在路由选择表里记录该信息；
 - 反应、补偿、宣告网络中的拓扑变化。
3. 路由的度量值，也叫做路由代价或者路由距离，用来决定到达一个目的的最佳路径。
最佳路由所使用的度量值类型定义。
4. 收敛时间：一组路由器所花费用来完成路由信息交换的时间。
5. 负载均衡是通过多条路径向同一目标地址传送报文的过程。有 4 种类型的负载均衡：
 - 等价，每报文
 - 等价，每目的
 - 非等价，每报文
 - 非等价，每目的
6. 距离矢量协议：每个路由器依据它邻居的路由来计算路由，然后传递给其他的邻居。
7. 距离矢量协议存在的一些问题是：
 - 因为它依靠邻居得到正确路由信息，所以易产生不正确的信息；
 - 收敛慢；
 - 路由环路；
 - 计数无限。
8. 邻居指的是连接至同一数据链路的路由器。
9. 当路由超出特定期限之后，路由无效计时器将它们从路由器中删除。
10. 使用简单的水平分隔，将不会发送路由信息到产生该路由信息的源点；具有毒性逆转的水平分隔虽然发送至源点，但把度量值设成不可达。
11. 当路由器环路更新路由时会产生计数到天穷大的问题；每个路由器增加路由的度量值直到度量值达到无穷大。可以把“无穷大”定义为合理的较低的度量值，这样可以很快达到无穷大，路由也就宣布为不可达，由此可以控制计数到无穷大的影响。
12. 抑制计时器可用来预防路由环路。如果一条路由宣布为不可达或者度量值超过特定的阈值，路由器将停止接受有关该路由的其他信息，直到抑制计时器超时。这样，可以防止路由器在网络收敛时接受错误的路由信息。
13. 距离矢量路由器发送它整个的路由选择表，但它仅仅发送给直接连接的邻居。链路状态路由器仅仅发送有关它直接相连链路的信息，但它广播该信息至整个网络区域。距离矢量协议通常使用不同的 Bellman—Ford 算法来计算路由，链路状态协议通常使用不同的 Dijkstra 算法来计算路由。

14. 拓扑数据库保存了路由选择域中所有路由器产生的链路状态信息。
15. 每个路由器广播链路状态信息通告, 该通告描述其自身的链路, 自身链路的状态, 以及整个网络区域中连接至这些链路的所有邻居。所有的路由器在链路状态数据库中保存所有收到的这些链路状态通告。每个路由器根据拓扑数据库中的信息计算最短路径树, 并且基于此树往路由选择表中添加路由。
16. 序列号帮助路由器区分同一个链路状态通告的多个拷贝, 并防止泛洪的链路状态通告在整个网络中无限循环。
17. Aging 防止老的、可能过期的链路状态信息在拓扑数据库中停留时间过长, 或被一个路由器接受。
18. 构建最短路径树时, 路由器首先将它自己作为根, 然后使用拓扑数据库中的信息, 建立所有与它直连的邻居列表。到一个邻居的代价最小的路径成为树的一个树枝, 该邻居的所有邻居也加进列表, 检查该列表, 看是否有重复路径, 如果有, 代价高的路径将从列表中删除, 列表中代价低的路由器将被加入树, 该路由器的邻居也加入列表, 再检查该列表, 看是否有重复路径。此过程不断重复, 直到列表中没有路由器为止。
19. 在一个路由选择域中, 区域是子域。由于限制了区域中每个路由器的链路状态数据库的大小, 区域使得链路状态路由更加高效。
20. 根据使用时的需要, 一个自主系统可被定义为一个在相同管理域中的网络, 或者是一个单独的路由选择域。
21. 内部网关协议是在自主系统内部进行路由的路由选择协议; 外部网关协议是在自主系统间进行路由的协议。

第 5 章

1. RIP 使用 UDP 端口 520。
2. RIP 的度量值是跳数。不可达网络的跳数是 16, RIP 将它视为无限距离。
3. 每隔 30s 减去一个小的随机变量, RIP 周期性地发送更新。这样可以防止和邻居的更新同步。
4. 如果 6 次更新都未收到, 则一个路由选择表项被标记为不可达。
5. 当一条路由宣布为不可达时, 启动垃圾收集计时器, 或者称为刷新计时器。当计时器超时, 该路由才从路由选择表中删除。这个过程使得一条不可达的路由在路由选择表中停留足够长的时间, 以便邻居都能够被通知到。
6. 随机计时器, 定时范围 1~5s, 防止了在拓扑改变时“触发更新”风暴。
7. 请求消息要求路由器进行更新。响应消息就是一个更新。
8. 请求消息可能要求一个完全的更新, 或者在一些特殊情况下, 它会要求特定的路由。
9. 当更新计时器超时, 或者当接受到一条请求消息时, 发出一响应消息。
10. RIP 更新不包括目标地址的子网掩码, 因此 RIP 路由器靠它自己接口的子网掩码来判断主网络地址怎样被子网化。如果不配置特定的主网络地址, 路由器将不知道主网络是被子网化的。因此, 将不会通告主网络地址的子网给其他的主网络。

第 6 章

1. IGRP 不使用 UDP 端口。它直接使用网络层，协议号为 9。
2. IGRP 网络的最大半径是 255 跳。
3. 缺省的 IGRP 更新周期是 90s。
4. IGRP 指定了自主系统号，这样在同一个路由选择域中，甚至在同一个路由器上，可以启动多个 IGRP 进程。
5. McCloy 将 192.168.1.0 作为系统路由通告至 Acheson，这是因为该地址被通告到了另外一个主网络。Acheson 将 172.16.0.0 作为系统路由通告至 McCloy，作为内部路由通告给 Kennan。
6. 缺省的 IGRP 抑制时间是 280s。
7. IGRP 能使用带宽、延迟、负载、可靠性来计算度量值。缺省只使用带宽和延迟。
8. 一个 IGRP 更新报文最多可携带 104 条路由选择表项。

第 7 章

1. 路由标记字段、子网掩码字段和下一跳字段是 RIPv2 的扩展，而在 RIPv1 消息中没有。RIP 消息的基本格式并没有改变。版本 2 仅仅添加了一些域。
2. 除了使用一些新的域，RIPv2 支持身份验证和组播更新。
3. RIPv2 使用组播地址 224.0.0.9。组播路由消息比广播好，因为主机和非 RIPv2 路由器将忽略组播消息。
4. 当另外的路由选择协议将 RIPv2 域作为它的传输域时，该协议便可以使用 RIPv2 的路由标记字段与 RIPv2 域另一端的对等体进行通信。
5. 下一跳字段用来把下一跳的地址通知给在相同多址网络上的其他路由器，这个下一跳地址到目标在度量值上要比源路由器近。
6. RIPv2 和 v1 使用相同的 UDP 端口，端口号 520。
7. 无类别路由选择协议不考虑在路由发现过程中的主网络地址，仅仅寻找最长匹配。
8. 为支持 VLSM，路由选择协议在其更新中必须包括每个目标地址的子网掩码。
9. RIPv2 的 Cisco 实现支持明文验证和 MD5 验证。RFC1723 仅仅定义了明文验证。

第 8 章

1. EIGRP 是一种距离矢量协议。
2. 缺省时，EIGRP 使用不超过 50% 的链路带宽，带宽配置在路由器的接口上。这个百分比可以通过命令 **ip bandwidth-percent eigrp** 来改变。
3. EIGRP 和 IGRP 使用相同的公式来计算复合度量值。但是 EIGRP 通过一个因子 256

来扩展度量值。

4. EIGRP 的 4 个基本部分是：

- 依赖于协议的模块
- 可靠传输协议
- 邻居发现和恢复模块
- 扩散更新算法

5. 可靠传输意味着 EIGRP 报文可以有保证、按次序地传输。RTP 使用可靠组播，收到的报文均被确认，以保证传输；使用序列号保证它们被有次序地传输。

6. 序列号保证路由器接收的是最新的路由选择表项。

7. EIGRP 使用组播地址 224.0.0.10。

8. EIGRP 使用的报文类型是：

- Hello 报文
- 确认报文
- 更新报文
- 查询报文
- 请求报文

9. 缺省的 EIGRP Hello 间隔是 5s，而在一些较慢的接口（T-1 或更低）上，这个时间是 60s。

10. EIGRP 的缺省抑制时间是 Hello 间隔的 3 倍。

11. 邻居表存储了有关 EIGRP-speaking 邻居的信息；拓扑表列出了所有具有可行后继的已知路由。

12. 到一个目标网络的可行距离是路由器计算出的到目标的最小距离。

13. 对一个目标网络来说，可行后继是通过可行条件选出的。如果一个邻居通告的到一个目标网络的距离比收到该通告的路由器到此目标网络的可行距离要小，那么就满足可行条件。也就是说，如果一个路由器的邻居到目标网络的距离比该路由器要近，那么这个邻居就是满足可行条件的。还有一种说法，即相对于目标网络而言，邻居是位于下游的。

14. 可行后继指的是一个满足可行条件的邻居。

15. 后继指的是目前正用来作为下一跳的可行后继。

16. 在特定路由器上，如果它已经查询了它的邻居来寻找可行后继，但是还没有收到任何一个邻居的回应时，我们说，这个路由器上的路由是活跃的；当查询全部完成，路由则是非活跃的。

17. 当拓扑表里没有可行后继时，路由变为活跃的。

18. 从被查询的邻居收到一条回应时，路由从活跃变为非活跃。

19. 当路由器在 active time（缺省 3min）内没有收到被查询邻居的回应，路由被宣布为 stuck-in-active。收到一条代表邻居的具有无限量值的回应，以满足 DUAL，然后该邻居从邻居表中删除。

20. 从一个 IP 网络地址产生一组子网地址，叫做子网化。地址聚合指的是从一组网络或子网地址汇总出一个 IP 网络地址。

第 9 章

1. 从 OSPF 路由器的角度来说, 邻居指的是直接与该 OSPF 路由器相邻的其他 OSPF 路由器。
2. OSPF 邻接是指到一个邻居的概念上的链路, 可以通过该链路传送 LSA。
3. 有 5 种 OSPF 报文类型, 它们的作用如下:
 - Hello, 用来发现邻居, 建立和保持邻接。
 - Update, 用来在邻居间发送 LSA。
 - 数据库描述报文, 被路由器用来在数据库同步的过程中向邻居描述它的链路状态数据库。
 - 链路状态请求报文, 被路由器用来向邻居的链路状态数据库请求 LSA。
 - 链路状态确认报文, 用来保证可靠的 LSA 传输。
4. 路由器产生一个链路状态通告来描述一个或多个目标网络。OSPF 的 Update 报文将 LSA 从一个邻居传送至另外一个。虽然 LSA 在整个 OSPF 域或区域(area)内广播, 但 Update 报文不会离开单个的数据链路。
5. 最常用的 LSA 类型和作用如下:
 - 类型 1 (Router LSA) 由每个路由器产生, 用来描述产生它的路由器, 该路由器的直连链路和状态, 以及该路由器的邻居。
 - 类型 2 (Network LSA) 由多点接入链路上的指定路由器 (Designated Router) 产生, 描述了该链路以及所有相连的邻居。
 - 类型 3 (Network Summary LSA) 由区域边界路由器 (Area Border Router) 产生, 描述了区域间的目标网络。
 - 类型 4 (ASBR Summary LSA) 由区域边界路由器 (Area Border Router) 产生, 描述了区域外的自主系统边界路由器 (Autonomous System Boundary Router)。
 - 类型 5 (AS External LSA) 由自主系统边界路由器 (Autonomous System Boundary Router) 产生, 描述 OSPF 域之外的目标网络。
 - 类型 7 (NSSA External LSA) 由非末梢区域内的自主系统边界路由器 (Autonomous System Boundary Router) 产生。
6. 路由器使用链路状态数据库来存储所有它知道的 OSPF LSA, 包括它自己的。数据库同步指的是保证一个区域内所有的路由器都有一致的链路状态数据库的过程。
7. 缺省的 OSPF HelloInterval 是 10s。
8. 缺省的 OSPF RouterDeadInterval 是 HelloInterval 的 4 倍。
9. 路由器 ID 是一个 OSPF 路由器用来标识它自己的地址。它或者是路由器所有 loopback 接口的最高地址; 或者如果没有配置 loopback 接口, 就是路由器所有 LAN 接口的最高地址。
10. 一个区域是一个 OSPF 子域, 在区域内, 所有的路由器都有一致的链路状态数据库。
11. 区域 0 是主干区域。其他所有的区域必须通过主干区域来发送它们的区域内流量。
12. MaxAge, 1h, 是 LSA 被认为过期的时限。
13. 4 种 OSPF 路由器类型是:

- 内部路由器 (IR): 它所有的 OSPF 接口属于同一个区域;
 - 主干路由器 (BR): 是区域 0 内的内部路由器;
 - 区域边界路由器 (ABR): 有 OSPF 接口在多个区域内;
 - 自主系统边界路由器 (ASBR): 宣告外部路由至 OSPF 域。
14. 4 种 OSPF 路径类型是:
- 域内路径
 - 域间路径
 - 类型 1 外部路径
 - 类型 2 外部路径
15. 5 种 OSPF 网络类型为:
- 点到点网络
 - 广播型网络
 - 非广播型多址网络
 - 点到多点网络
 - 虚链路
16. 指定路由器: 代表了一个多址网络以及连接至这个网络和 OSPF 域的其余部分的路由器。
17. Cisco IOS 这样计算一个接口的出站代价: $10^8/BW$, 其中 BW 是该接口上配置的带宽。
18. 如果一个区域内的一个或者多个路由器不通过向区域外送报文就不能向区域内的其他路由器送报文, 那么这个区域应该划分子区域。
19. 虚链路指的是通过一个非主干区域扩展 OSPF 主干连接的通道。
20. 末梢区域 (Stub Area) 是指类型 5 LSA 不能广播进的区域。完全末梢区域指的是类型 3、4 或 5 LSA 不能广播进的区域 (除了用来通告缺省路由的类型 3 LSA)。通过非末梢区域, 外部的目标网络能够被通告至 OSPF 域内, 但是类型 5 LSA 不能被 ABR 送至非末梢区域。
21. OSPF 网络表项是路由选择表中的表项, 描述了 IP 目标网络。OSPF 路由器表项是在一个单独的路由选择表中的表项, 该路由选择表只记录了到达 ABR 和 ASBR 的路由。
22. 类型 2 的认证使用 MD5 加密, 类型 1 认证使用明文密码。
23. LSA 头中区别不同 LSA 的 3 个字段分别是类型、通告路由器、链路状态 ID。LSA 头中区别相同 LSA 的不同实例的 3 个字段分别是序列号、老化时间、校验和。

第 10 章

1. 中间系统是 ISO 称呼路由器的术语。
2. 网络协议数据单元是 ISO 称呼报文的术语。
3. L1 路由器与其他的区域没有直接连接。L2 路由器仅仅路由区域间的流量。L1/L2 路由器路由区域间和区域内的流量, 并且为 L1 路由器充当区域间网关。
4. IS-IS 区域的边界在路由器之间, 在链路之上。OSPF 区域的边界由路由器自己定义。

5. 网络实体标题是路由器标识自己和它所在区域的一个地址。
6. 在一个 NET 内 NSAP 选择符应该设成 0x00。
7. 系统 ID 在 IS-IS 域内惟一地标识了一个路由器。
8. NET 最后 7 个 8bit 字节前面的部分是区域地址。
9. IS-IS 不选取 BDR。
10. 伪节点 ID 是 LAN ID 的最后一个 8bit 字节。它的作用是为了区别由一个路由器产生的多个 LAN ID, 该路由器在多个 LAN 中是 DR。
11. IS-IS LSP 的 MaxAge 是 1200s (20min)。
12. OSPF 增加 age 至 MaxAge; IS-IS 减小 age 至 0。一个新的 OSPF LSA 有 age 值 0, 而一个新的 IS-IS LSP 有 age 值 MaxAge。
13. IS-IS 路由器的刷新率是 900s (15min)。
14. 一个完全序列号报文 (CSNP) 列出了一个数据库中所有的 LSP。在一个广播网络上, 指定路由器周期性地发送 CSNP 来保持数据库的同步。
15. 一个部分序列号报文列出了一个或多个 LSP。有两个用途: 在点到点网络上, 它用来确认 LSP 的接收; 在广播网络上, 它用来请求 LSP。
16. IS-IS 路由器使用超载位通知它的邻居它的内存过载了, 并且不能存储整个链路状态数据库。
17. L1/L2 路由器使用 Attached 位通知 L1 路由器它与 L2 主干有连接。
18. ISO 指定 4 个度量值: 缺省度量 (Default)、开销度量 (Expense)、时延度量 (Delay)、差错度量 (Error)。Cisco 只支持 Default。
19. 任何 IS-IS 度量值的最大值是 63。
20. 一个 IS-IS 路由的最大度量值是 1023。
21. L1 IS-IS 度量值作用于区域内路由, L2 IS-IS 度量值作用于区域间路由。
22. 内部度量值作用于目标网络在 IS-IS 域内的路由; 外部度量值作用于目标网络在 IS-IS 域外的路由。

第 11 章

1. 从其他的路由选择协议、静态路由或者直连网络学来的路由可以被重分配进一个路由选择域。
2. 度量值用来在同一个路由选择协议产生的到达同一目标网络的多条路由中间决定最佳路径; 而管理距离用来在不同路由选择协议产生的到达同一目标网络的多条路由中间决定最佳路径。
3. 在一个具有较高管理距离值的路由选择域中, 一条路由可以被重新分配到具有较低管理距离值的路由选择域。如果该路由被分配回高管理距离域, 报文可能会误路由至低管理距离域。
4. 从无类域到有类域重新分配可变子网化的目标网络地址可能会有问题。
5. OSPF 和 IS-IS 能理解缺省度量值, 而 RIP、IGRP 和 EIGRP 则不能。
6. **metric** 命令为特定的重新分配语句引入了度量值。**default-metric** 语句为所有不包括

metric 命令的重新分配语句引入度量值。

7. 如果没有 **subnets** 关键字, 只有不与路由器直连的主网络地址将被重新分配。

8. 产生汇总路由的路由器应该使用 **null** 接口作为汇总路由的下一跳。如果有一个报文, 只能匹配汇总路由, 但不能匹配更加具体的、能够到达目的地址的路由, 那么这个报文将被丢弃。这防止了路由器转发丢失的报文。

第 12 章

1. 缺省路由地址是 0.0.0.0。

2. IGRP 和 EIGRP 通告一个缺省地址作为外部地址类型。

3. 是的。

4. 末梢路由器是指只有一条链路连至其他路由器的路由器。末梢网络是指只连至一台路由器的网络。

5. 使用缺省路由 (而不是完整的路由选择表) 能够使路由选择表很小, 节省路由器内存, 并且能限制必须被处理的路由信息, 节省路由器处理周期。

6. 使用完整的路由选择表 (而不是缺省路由) 能使路由匹配更加精确。

7. ODR 使用 Cisco 发现协议 (CDP) 来发现路由。

8. ODR 在 IOS 11.2 和以后版本中可用。

9. ODR 运行的介质必须支持 SNAP。

第 14 章

1. 路由图和访问列表相似, 它们都定义了匹配的标准, 以及匹配后采取的动作。它们的不同点在于路由图不光定义了匹配标准, 还定义了设置 (set) 标准。设置标准能改变一条路由或者根据报文的参数路由一个报文。

2. 策略路由是一种静态路由, 它使用路由图来决定哪些报文应该转发, 应该往哪儿转发。

3. 路由标签是路由信息报文中的一些位域, 它们允许在路由选择域内部能够携带外部信息。

4. 路由标签不会影响携带它们的路由选择协议。

附录 E

配置练习答案

第 2 章

1. 如果 D 类地址的前 4 位是 1110, 那么第一个 8bit 字节最小是 11100000, 最大是 11101111。用十进制数表示, 分别为 224 和 239。所以, D 类地址的第一个 8bit 字节取值从 224 到 239。

2. (a) 需要足够的子网位数 n , 使得 $2^n - 2 \geq 16\ 000$; 需要足够的主机位数 h , 使得 $2^h - 2 \geq 700$ 。子网掩码 255.255.252.0 为一个 A 类地址提供 16 382 个子网, 为其中每一个子网提供 1022 个主机地址。此掩码是惟一的答案。如果多一个子网位 (255.255.254.0), 将没有足够的主机地址。如果少一个子网位 (255.255.248.0), 将没有足够的子网。

(b) 需要足够的子网位数 n , 使得 $2^n - 2 \geq 500$; 需要足够的主机位数 h , 使得 $2^h - 2 \geq 100$ 。子网掩码 255.255.255.128 为一个 B 类地址提供 510 个子网, 为其中每一个子网提供 126 个主机地址。同样, 此掩码是惟一的答案。

3. 用 6 位来子网化, 一个 C 类地址将有 $2^6 - 2 = 62$ 个子网, 每个子网有 $2^2 - 2 = 2$ 个主机地址。以这种模式, 一个 C 类地址可以被 62 个点-to-点链路所使用。一个点对-点链路只需要 2 个主机地址——链路的每一端一个。

4. 有 28 位掩码的 C 类地址可以有 14 个子网, 每个子网有 14 个主机地址。首先给出子网。

这些子网是:

下面给出每个子网的主机。每个子网的广播地址同样给出。
每个子网的主机地址是：

```

11000000101010001001001100010000 = 192.168.147.16 (subnet)
11000000101010001001001100010001 = 192.168.147.17
11000000101010001001001100010010 = 192.168.147.18
11000000101010001001001100010011 = 192.168.147.19
11000000101010001001001100010100 = 192.168.147.20
11000000101010001001001100010101 = 192.168.147.21
11000000101010001001001100010110 = 192.168.147.22
11000000101010001001001100010111 = 192.168.147.23
11000000101010001001001100011000 = 192.168.147.24
11000000101010001001001100011001 = 192.168.147.25
11000000101010001001001100011010 = 192.168.147.26
11000000101010001001001100011011 = 192.168.147.27
11000000101010001001001100011100 = 192.168.147.28
11000000101010001001001100011101 = 192.168.147.29
11000000101010001001001100011110 = 192.168.147.30
11000000101010001001001100011111 = 192.168.147.31 (broadcast)
11000000101010001001001100100000 = 192.168.147.32 (subnet)
11000000101010001001001100100001 = 192.168.147.33
11000000101010001001001100100010 = 192.168.147.34
11000000101010001001001100100011 = 192.168.147.35
110000001010100010010011001000100 = 192.168.147.36
110000001010100010010011001000101 = 192.168.147.37
110000001010100010010011001000110 = 192.168.147.38
110000001010100010010011001000111 = 192.168.147.39
110000001010100010010011001001000 = 192.168.147.40
110000001010100010010011001001001 = 192.168.147.41
110000001010100010010011001001010 = 192.168.147.42
110000001010100010010011001001011 = 192.168.147.43
110000001010100010010011001001100 = 192.168.147.44
110000001010100010010011001001101 = 192.168.147.45
110000001010100010010011001001110 = 192.168.147.46
110000001010100010010011001001111 = 192.168.147.47 (broadcast)
11000000101010001001001100110000 = 192.168.147.48 (subnet)

```


110000001010100010010011**0011**0001 = 192.168.147.49
110000001010100010010011**0011**0010 = 192.168.147.50
110000001010100010010011**0011**0011 = 192.168.147.51
110000001010100010010011**0011**0100 = 192.168.147.52
110000001010100010010011**0011**0101 = 192.168.147.53
110000001010100010010011**0011**0110 = 192.168.147.54
110000001010100010010011**0011**0111 = 192.168.147.55
110000001010100010010011**0011**1000 = 192.168.147.56
110000001010100010010011**0011**1001 = 192.168.147.57
110000001010100010010011**0011**1010 = 192.168.147.58
110000001010100010010011**0011**1011 = 192.168.147.59
110000001010100010010011**0011**1100 = 192.168.147.60
110000001010100010010011**0011**1101 = 192.168.147.61
110000001010100010010011**0011**1110 = 192.168.147.62
110000001010100010010011**0011**1111 = 192.168.147.63 (broadcast)
110000001010100010010011**0100**0000 = 192.168.147.64 (subnet)
110000001010100010010011**0100**0001 = 192.168.147.65
110000001010100010010011**0100**0010 = 192.168.147.66
110000001010100010010011**0100**0011 = 192.168.147.67
110000001010100010010011**0100**0100 = 192.168.147.68
110000001010100010010011**0100**0101 = 192.168.147.69
110000001010100010010011**0100**0110 = 192.168.147.70
110000001010100010010011**0100**0111 = 192.168.147.71
110000001010100010010011**0100**1000 = 192.168.147.72
110000001010100010010011**0100**1001 = 192.168.147.73
110000001010100010010011**0100**1010 = 192.168.147.74
110000001010100010010011**0100**1011 = 192.168.147.75
110000001010100010010011**0100**1100 = 192.168.147.76
110000001010100010010011**0100**1101 = 192.168.147.77
110000001010100010010011**0100**1110 = 192.168.147.78
110000001010100010010011**0100**1111 = 192.168.147.79 (broadcast)
110000001010100010010011**0101**0000 = 192.168.147.80 (subnet)
110000001010100010010011**0101**0001 = 192.168.147.81
110000001010100010010011**0101**0010 = 192.168.147.82
110000001010100010010011**0101**0011 = 192.168.147.83
110000001010100010010011**0101**0100 = 192.168.147.84
110000001010100010010011**0101**0101 = 192.168.147.85
110000001010100010010011**0101**0110 = 192.168.147.86
110000001010100010010011**0101**0111 = 192.168.147.87
110000001010100010010011**0101**1000 = 192.168.147.88
110000001010100010010011**0101**1001 = 192.168.147.89
110000001010100010010011**0101**1010 = 192.168.147.90
110000001010100010010011**0101**1011 = 192.168.147.91
110000001010100010010011**0101**1100 = 192.168.147.92
110000001010100010010011**0101**1101 = 192.168.147.93
110000001010100010010011**0101**1110 = 192.168.147.94
110000001010100010010011**0101**1111 = 192.168.147.95 (broadcast)
110000001010100010010011**0110**0000 = 192.168.147.96 (subnet)
110000001010100010010011**0110**0001 = 192.168.147.97
110000001010100010010011**0110**0010 = 192.168.147.98
110000001010100010010011**0110**0011 = 192.168.147.99
110000001010100010010011**0110**0100 = 192.168.147.100

110000001010100010010011**01100101** = 192.168.147.101
110000001010100010010011**01100110** = 192.168.147.102
110000001010100010010011**01100111** = 192.168.147.103
110000001010100010010011**01101000** = 192.168.147.104
110000001010100010010011**01101001** = 192.168.147.105
110000001010100010010011**01101010** = 192.168.147.106
110000001010100010010011**01101011** = 192.168.147.107
110000001010100010010011**01101100** = 192.168.147.108
110000001010100010010011**01101101** = 192.168.147.109
110000001010100010010011**01101110** = 192.168.147.110
110000001010100010010011**01101111** = 192.168.147.111 (broadcast)
110000001010100010010011**01110000** = 192.168.147.112 (subnet)
110000001010100010010011**01110001** = 192.168.147.113
110000001010100010010011**01110010** = 192.168.147.114
110000001010100010010011**01110011** = 192.168.147.115
110000001010100010010011**01110100** = 192.168.147.116
110000001010100010010011**01110101** = 192.168.147.117
110000001010100010010011**01110110** = 192.168.147.118
110000001010100010010011**01110111** = 192.168.147.119
110000001010100010010011**01111000** = 192.168.147.120
110000001010100010010011**01111001** = 192.168.147.121
110000001010100010010011**01111010** = 192.168.147.122
110000001010100010010011**01111011** = 192.168.147.123
110000001010100010010011**01111100** = 192.168.147.124
110000001010100010010011**01111101** = 192.168.147.125
110000001010100010010011**01111110** = 192.168.147.126
110000001010100010010011**01111111** = 192.168.147.127 (broadcast)
110000001010100010010011**10000000** = 192.168.147.128 (subnet)
110000001010100010010011**10000001** = 192.168.147.129
110000001010100010010011**10000010** = 192.168.147.130
110000001010100010010011**10000011** = 192.168.147.131
110000001010100010010011**10000100** = 192.168.147.132
110000001010100010010011**10000101** = 192.168.147.133
110000001010100010010011**10000110** = 192.168.147.134
110000001010100010010011**10000111** = 192.168.147.135
110000001010100010010011**10001000** = 192.168.147.136
110000001010100010010011**10001001** = 192.168.147.137
110000001010100010010011**10001010** = 192.168.147.138
110000001010100010010011**10001011** = 192.168.147.139
110000001010100010010011**10001100** = 192.168.147.140
110000001010100010010011**10001101** = 192.168.147.141
110000001010100010010011**10001110** = 192.168.147.142
110000001010100010010011**10001111** = 192.168.147.143 (broadcast)
110000001010100010010011**10010000** = 192.168.147.144 (subnet)
110000001010100010010011**10010001** = 192.168.147.145
110000001010100010010011**10010010** = 192.168.147.146
110000001010100010010011**10010011** = 192.168.147.147
110000001010100010010011**10010100** = 192.168.147.148
110000001010100010010011**10010101** = 192.168.147.149
110000001010100010010011**10010110** = 192.168.147.150
110000001010100010010011**10010111** = 192.168.147.151
110000001010100010010011**10011000** = 192.168.147.152

110000001010100010010011**1001**1001 = 192.168.147.153
110000001010100010010011**1001**1010 = 192.168.147.154
110000001010100010010011**1001**1011 = 192.168.147.155
110000001010100010010011**1001**1100 = 192.168.147.156
110000001010100010010011**1001**1101 = 192.168.147.157
110000001010100010010011**1001**1110 = 192.168.147.158
110000001010100010010011**1001**1111 = 192.168.147.159 (broadcast)
110000001010100010010011**1010**0000 = 192.168.147.160 (subnet)
110000001010100010010011**1010**0001 = 192.168.147.161
110000001010100010010011**1010**0010 = 192.168.147.162
110000001010100010010011**1010**0011 = 192.168.147.163
110000001010100010010011**1010**0100 = 192.168.147.164
110000001010100010010011**1010**0101 = 192.168.147.165
110000001010100010010011**1010**0110 = 192.168.147.166
110000001010100010010011**1010**0111 = 192.168.147.167
110000001010100010010011**1010**1000 = 192.168.147.168
110000001010100010010011**1010**1001 = 192.168.147.169
110000001010100010010011**1010**1010 = 192.168.147.170
110000001010100010010011**1010**1011 = 192.168.147.171
110000001010100010010011**1010**1100 = 192.168.147.172
110000001010100010010011**1010**1101 = 192.168.147.173
110000001010100010010011**1010**1110 = 192.168.147.174
110000001010100010010011**1010**1111 = 192.168.147.175 (broadcast)
110000001010100010010011**1011**0000 = 192.168.147.176 (subnet)
110000001010100010010011**1011**0001 = 192.168.147.177
110000001010100010010011**1011**0010 = 192.168.147.178
110000001010100010010011**1011**0011 = 192.168.147.179
110000001010100010010011**1011**0100 = 192.168.147.180
110000001010100010010011**1011**0101 = 192.168.147.181
110000001010100010010011**1011**0110 = 192.168.147.182
110000001010100010010011**1011**0111 = 192.168.147.183
110000001010100010010011**1011**1000 = 192.168.147.184
110000001010100010010011**1011**1001 = 192.168.147.185
110000001010100010010011**1011**1010 = 192.168.147.186
110000001010100010010011**1011**1011 = 192.168.147.187
110000001010100010010011**1011**1100 = 192.168.147.188
110000001010100010010011**1011**1101 = 192.168.147.189
110000001010100010010011**1011**1110 = 192.168.147.190
110000001010100010010011**1011**1111 = 192.168.147.191 (broadcast)
110000001010100010010011**1100**0000 = 192.168.147.192 (subnet)
110000001010100010010011**1100**0001 = 192.168.147.193
110000001010100010010011**1100**0010 = 192.168.147.194
110000001010100010010011**1100**0011 = 192.168.147.195
110000001010100010010011**1100**0100 = 192.168.147.196
110000001010100010010011**1100**0101 = 192.168.147.197
110000001010100010010011**1100**0110 = 192.168.147.198
110000001010100010010011**1100**0111 = 192.168.147.199
110000001010100010010011**1100**1000 = 192.168.147.200
110000001010100010010011**1100**1001 = 192.168.147.201
110000001010100010010011**1100**1010 = 192.168.147.202
110000001010100010010011**1100**1011 = 192.168.147.203
110000001010100010010011**1100**1100 = 192.168.147.204


```

11000000101010001001001111001101 = 192.168.147.205
11000000101010001001001111001110 = 192.168.147.206
11000000101010001001001111001111 = 192.168.147.207 (broadcast)
11000000101010001001001111010000 = 192.168.147.208 (subnet)
11000000101010001001001111010001 = 192.168.147.209
11000000101010001001001111010010 = 192.168.147.210
11000000101010001001001111010011 = 192.168.147.211
11000000101010001001001111010100 = 192.168.147.212
11000000101010001001001111010101 = 192.168.147.213
11000000101010001001001111010110 = 192.168.147.214
11000000101010001001001111010111 = 192.168.147.215
11000000101010001001001111011000 = 192.168.147.216
11000000101010001001001111011001 = 192.168.147.217
11000000101010001001001111011010 = 192.168.147.218
11000000101010001001001111011011 = 192.168.147.219
11000000101010001001001111011100 = 192.168.147.220
11000000101010001001001111011101 = 192.168.147.221
11000000101010001001001111011110 = 192.168.147.222
11000000101010001001001111011111 = 192.168.147.223 (broadcast)
11000000101010001001001111100000 = 192.168.147.224 (subnet)
11000000101010001001001111100001 = 192.168.147.225
11000000101010001001001111100010 = 192.168.147.226
11000000101010001001001111100011 = 192.168.147.227
11000000101010001001001111100100 = 192.168.147.228
11000000101010001001001111100101 = 192.168.147.229
11000000101010001001001111100110 = 192.168.147.230
11000000101010001001001111100111 = 192.168.147.231
11000000101010001001001111101000 = 192.168.147.232
11000000101010001001001111101001 = 192.168.147.233
11000000101010001001001111101010 = 192.168.147.234
11000000101010001001001111101011 = 192.168.147.235
11000000101010001001001111101100 = 192.168.147.236
11000000101010001001001111101101 = 192.168.147.237
11000000101010001001001111101110 = 192.168.147.238
11000000101010001001001111101111 = 192.168.147.239 (broadcast)

```

5. 这里给出此题的答案, 可以看出比上一题短, 因为没有写出每一个子网的每一个主机。

有 29 位掩码的 C 类地址意味着将有 30 个子网, 每个子网有 6 个主机地址。这些子网是:

```

11111111111111111111111111111000 = 255.255.255.248 (mask)
11000000101010001001001100001000 = 192.168.147.8
11000000101010001001001100010000 = 192.168.147.16
11000000101010001001001100011000 = 192.168.147.24
11000000101010001001001100100000 = 192.168.147.32
11000000101010001001001100101000 = 192.168.147.40
11000000101010001001001100110000 = 192.168.147.48
11000000101010001001001100111000 = 192.168.147.56
11000000101010001001001101000000 = 192.168.147.64
11000000101010001001001101001000 = 192.168.147.72

```



```

11000000101010001001001101010000 = 192.168.147.80
11000000101010001001001101011000 = 192.168.147.88
11000000101010001001001101100000 = 192.168.147.96
11000000101010001001001101101000 = 192.168.147.104
11000000101010001001001101100000 = 192.168.147.112
11000000101010001001001101111000 = 192.168.147.120
11000000101010001001001110000000 = 192.168.147.128
11000000101010001001001110001000 = 192.168.147.136
11000000101010001001001110010000 = 192.168.147.144
11000000101010001001001110011000 = 192.168.147.152
11000000101010001001001110100000 = 192.168.147.160
11000000101010001001001110101000 = 192.168.147.168
11000000101010001001001110110000 = 192.168.147.176
11000000101010001001001110111000 = 192.168.147.184
11000000101010001001001111000000 = 192.168.147.192
11000000101010001001001111001000 = 192.168.147.200
11000000101010001001001111010000 = 192.168.147.208
11000000101010001001001111011000 = 192.168.147.216
11000000101010001001001111100000 = 192.168.147.224
11000000101010001001001111101000 = 192.168.147.232
11000000101010001001001111110000 = 192.168.147.240

```

下面给出每个子网的广播地址（将每个子网的主机位全置 1）。

子网广播地址是：

```

11000000101010001001001100001111 = 192.168.147.15
11000000101010001001001100010111 = 192.168.147.23
11000000101010001001001100011111 = 192.168.147.31
11000000101010001001001100100111 = 192.168.147.39
11000000101010001001001100101111 = 192.168.147.47
11000000101010001001001100110111 = 192.168.147.55
11000000101010001001001100111111 = 192.168.147.63
11000000101010001001001101000111 = 192.168.147.71
11000000101010001001001101001111 = 192.168.147.79
11000000101010001001001101010111 = 192.168.147.87
11000000101010001001001101011111 = 192.168.147.95
11000000101010001001001101100111 = 192.168.147.103
11000000101010001001001101101111 = 192.168.147.111
11000000101010001001001101100111 = 192.168.147.119
11000000101010001001001101111111 = 192.168.147.127
11000000101010001001001110000111 = 192.168.147.135
11000000101010001001001110001111 = 192.168.147.143
11000000101010001001001110010111 = 192.168.147.151
11000000101010001001001110011111 = 192.168.147.159
11000000101010001001001110100111 = 192.168.147.167
11000000101010001001001110101111 = 192.168.147.175
11000000101010001001001110110111 = 192.168.147.183
11000000101010001001001110111111 = 192.168.147.191
11000000101010001001001111000111 = 192.168.147.199
11000000101010001001001111001111 = 192.168.147.207
11000000101010001001001111010111 = 192.168.147.215
11000000101010001001001111011111 = 192.168.147.223

```


11000000101010001001001111100111 = 192.168.147.231
 11000000101010001001001111101111 = 192.168.147.239
 11000000101010001001001111110111 = 192.168.147.247

最后, 给出每个子网的主机地址, 它们是介于子网地址和子网广播地址之间的地址。

Subnet	Broadcast	Host Addresses
192.168.147.8	192.168.147.15	192.168.147.9 - 192.168.147.14
192.168.147.16	192.168.147.23	192.168.147.17 - 192.168.147.22
192.168.147.24	192.168.147.31	192.168.147.25 - 192.168.147.30
192.168.147.32	192.168.147.39	192.168.147.33 - 192.168.147.38
192.168.147.40	192.168.147.47	192.168.147.41 - 192.168.147.46
192.168.147.48	192.168.147.55	192.168.147.49 - 192.168.147.54
192.168.147.56	192.168.147.63	192.168.147.57 - 192.168.147.62
192.168.147.64	192.168.147.71	192.168.147.65 - 192.168.147.70
192.168.147.72	192.168.147.79	192.168.147.73 - 192.168.147.78
192.168.147.80	192.168.147.87	192.168.147.81 - 192.168.147.86
192.168.147.88	192.168.147.95	192.168.147.89 - 192.168.147.94
192.168.147.96	192.168.147.103	192.168.147.97 - 192.168.147.102
192.168.147.104	192.168.147.111	192.168.147.105 - 192.168.147.110
192.168.147.112	192.168.147.119	192.168.147.113 - 192.168.147.118
192.168.147.120	192.168.147.127	192.168.147.121 - 192.168.147.126
192.168.147.128	192.168.147.135	192.168.147.129 - 192.168.147.134
192.168.147.136	192.168.147.143	192.168.147.137 - 192.168.147.142
192.168.147.144	192.168.147.151	192.168.147.145 - 192.168.147.150
192.168.147.152	192.168.147.159	192.168.147.153 - 192.168.147.158
192.168.147.160	192.168.147.167	192.168.147.161 - 192.168.147.166
192.168.147.168	192.168.147.175	192.168.147.169 - 192.168.147.174
192.168.147.176	192.168.147.183	192.168.147.177 - 192.168.147.182
192.168.147.184	192.168.147.191	192.168.147.185 - 192.168.147.190
192.168.147.192	192.168.147.199	192.168.147.193 - 192.168.147.198
192.168.147.200	192.168.147.207	192.168.147.201 - 192.168.147.206
192.168.147.208	192.168.147.215	192.168.147.209 - 192.168.147.214
192.168.147.216	192.168.147.223	192.168.147.217 - 192.168.147.222
192.168.147.224	192.168.147.231	192.168.147.225 - 192.168.147.230
192.168.147.232	192.168.147.239	192.168.147.233 - 192.168.147.238
192.168.147.240	192.168.147.247	192.168.147.241 - 192.168.147.246

6. 一个有 20 位掩码的 B 类地址将有 14 个子网, 每个子网有 4 094 个主机地址。这些子网是:

11111111111111111111000000000000 = 255.255.240.0 (mask)
 10101100000100000001000000000000 = 172.16.16.0
 10101100000100000010000000000000 = 172.16.32.0
 10101100000100000011000000000000 = 172.16.48.0
 10101100000100000100000000000000 = 172.16.64.0
 10101100000100000101000000000000 = 172.16.80.0
 10101100000100000110000000000000 = 172.16.96.0
 10101100000100000111000000000000 = 172.16.112.0
 10101100000100001000000000000000 = 172.16.128.0

10101100000100001001000000000000 = 172.16.144.0
10101100000100001010000000000000 = 172.16.160.0
10101100000100001011000000000000 = 172.16.176.0
10101100000100001100000000000000 = 172.16.192.0
10101100000100001101000000000000 = 172.16.208.0
10101100000100001110000000000000 = 172.16.224.0

子网广播地址是:

10101100000100000001111111111111 = 172.16.31.255
10101100000100000010111111111111 = 172.16.47.255
10101100000100000011111111111111 = 172.16.63.255
10101100000100000100111111111111 = 172.16.79.255
10101100000100000101111111111111 = 172.16.95.255
10101100000100000110111111111111 = 172.16.111.255
10101100000100000111111111111111 = 172.16.127.255
10101100000100001000111111111111 = 172.16.143.255
10101100000100001001111111111111 = 172.16.159.255
10101100000100001010111111111111 = 172.16.175.255
10101100000100001011111111111111 = 172.16.191.255
10101100000100001100111111111111 = 172.16.207.255
10101100000100001101111111111111 = 172.16.223.255
10101100000100001110111111111111 = 172.16.239.255

使用以上的子网和广播地址, 主机地址是:

Subnet	Broadcast	Host Addresses
172.16.16.0	172.16.31.255	172.16.16.1-172.16.31.254
172.16.32.0	172.16.47.255	172.16.32.1-172.16.47.254
172.16.48.0	172.16.63.255	172.16.48.1-172.16.63.254
172.16.64.0	172.16.79.255	172.16.64.1-172.16.79.254
172.16.80.0	172.16.95.255	172.16.80.1-172.16.95.254
172.16.96.0	172.16.111.255	172.16.96.1-172.16.111.254
172.16.112.0	172.16.127.255	172.16.112.1-172.16.127.254
172.16.128.0	172.16.143.255	172.16.128.1-172.16.143.254
172.16.144.0	172.16.159.255	172.16.144.1-172.16.159.254
172.16.160.0	172.16.175.255	172.16.160.1-172.16.175.254
172.16.176.0	172.16.191.255	172.16.176.1-172.16.191.254
172.16.192.0	172.16.207.255	172.16.192.1-172.16.207.254
172.16.208.0	172.16.223.255	172.16.208.1-172.16.223.254
172.16.224.0	172.16.239.255	172.224.16.1-172.16.239.254

第 3 章

1. 首先确定每个链路的子网地址, 然后写出静态路由。记住路由器的路由选择表中将缺省包含直接相连的子网。静态路由是:

RTA

ip route 192.168.2.64 255.255.255.224 192.168.2.131
ip route 192.168.2.160 255.255.255.224 192.168.2.131


```
ip route 192.168.1.144/28 255.255.255.240 192.168.2.131
ip route 192.168.1.16 255.255.255.240 192.168.2.131
ip route 192.168.2.32 255.255.255.224 192.168.2.131
ip route 192.168.1.160 255.255.255.240 192.168.2.131
ip route 10.1.1.0 255.255.255.0 192.168.2.131
ip route 10.1.3.0 255.255.255.0 192.168.2.131
ip route 10.1.2.0 255.255.255.0 192.168.2.131
```

RTB

```
ip route 10.1.4.0 255.255.255.0 192.168.2.132
ip route 192.168.1.128 255.255.255.240 192.168.2.174
ip route 192.168.1.16 255.255.255.240 192.168.2.174
ip route 192.168.2.32 255.255.255.224 192.168.2.174
ip route 192.168.1.160 255.255.255.240 192.168.2.174
ip route 10.1.1.0 255.255.255.0 192.168.2.174
ip route 10.1.3.0 255.255.255.0 192.168.2.174
ip route 10.1.2.0 255.255.255.0 192.168.2.174
```

RTC

```
ip route 10.1.4.0 255.255.255.0 192.168.2.185
ip route 192.168.2.128 255.255.255.224 192.168.2.185
ip route 192.168.2.64 255.255.255.224 192.168.2.185
ip route 192.168.2.32 255.255.255.224 192.168.1.20
ip route 10.1.1.0 255.255.255.0 192.168.1.173
ip route 10.1.3.0 255.255.255.0 192.168.1.173
ip route 10.1.2.0 255.255.255.0 192.168.1.173
```

RTD

```
ip route 10.1.4.0 255.255.255.0 192.168.1.29
ip route 192.168.2.128 255.255.255.224 192.168.1.29
ip route 192.168.2.64 255.255.255.224 192.168.1.29
ip route 192.168.2.160 255.255.255.224 192.168.1.29
ip route 192.168.1.128 255.255.255.240 192.168.1.29
ip route 192.168.1.160 255.255.255.240 192.168.1.29
ip route 10.1.1.0 255.255.255.0 192.168.1.29
ip route 10.1.3.0 255.255.255.0 192.168.1.29
ip route 10.1.2.0 255.255.255.0 192.168.1.29
```

RTE

```
ip route 10.1.4.0 255.255.255.0 192.168.1.163
ip route 192.168.2.128 255.255.255.224 192.168.1.163
ip route 192.168.2.64 255.255.255.224 192.168.1.163
ip route 192.168.2.160 255.255.255.224 192.168.1.163
ip route 192.168.1.128 255.255.255.240 192.168.1.163
ip route 192.168.1.16 255.255.255.240 192.168.1.163
ip route 192.168.2.32 255.255.255.224 192.168.1.163
ip route 10.1.2.0 255.255.255.0 10.1.3.2
```

RTF

```
ip route 10.1.4.0 255.255.255.0 10.1.3.1
ip route 192.168.2.128 255.255.255.224 10.1.3.1
ip route 192.168.2.64 255.255.255.224 10.1.3.1
```



```
ip route 192.168.2.160 255.255.255.224 10.1.3.1
ip route 192.168.1.128 255.255.255.240 10.1.3.1
ip route 192.168.1.16 255.255.255.240 10.1.3.1
ip route 192.168.2.32 255.255.255.224 10.1.3.1
ip route 192.168.1.160 255.255.255.240 10.1.3.1
ip route 10.1.1.0 255.255.255.0 10.1.3.1
```

2. 静态路由是:

RTA

```
ip route 192.168.0.0 255.255.0.0 192.168.2.131
ip route 10.1.0.0 255.255.0.0 192.168.2.131
```

RTB

```
ip route 10.1.4.0 255.255.255.0 192.168.2.132
ip route 192.168.0.0 255.255.0.0 192.168.2.174
ip route 10.1.0.0 255.255.0.0 192.168.2.174
```

RTC

```
ip route 10.1.4.0 255.255.255.0 192.168.2.185
ip route 192.168.2.0 255.255.255.224 192.168.2.185
ip route 192.168.2.32 255.255.255.224 192.168.1.20
ip route 10.1.0.0 255.255.0.0 192.168.1.173
ip route 192.168.2.64 255.255.255.224 192.168.2.185
ip route 192.168.2.168 255.255.255.224 192.168.2.185
```

RTD

```
ip route 10.1.0.0 255.255.0.0 192.168.1.29
ip route 192.168.0.0 255.255.0.0 192.168.1.29
```

RTE

```
ip route 10.1.4.0 255.255.255.0 192.168.1.163
ip route 192.168.0.0 255.255.0.0 192.168.1.163
ip route 10.1.2.0 255.255.255.0 10.1.3.2
```

RTF

```
ip route 10.1.0.0 255.255.0.0 10.1.3.1
ip route 192.168.0.0 255.255.0.0 10.1.3.1
```

3. 静态路由是:

RTA

```
ip route 172.16.7.0 255.255.255.0 172.16.2.2
ip route 172.16.7.0 255.255.255.0 172.16.4.2 50
ip route 172.16.6.0 255.255.255.0 172.16.2.2
ip route 172.16.6.0 255.255.255.0 172.16.4.2 50
ip route 172.16.8.0 255.255.255.0 172.16.4.2
ip route 172.16.8.0 255.255.255.0 172.16.2.2 50
ip route 172.16.5.0 255.255.255.0 172.16.4.2
ip route 172.16.5.0 255.255.255.0 172.16.2.2 50
ip route 172.16.9.0 255.255.255.0 172.16.2.2
ip route 172.16.9.0 255.255.255.0 172.16.4.2
```


RTB

```

ip route 172.16.1.0 255.255.255.0 172.16.2.1
ip route 172.16.1.0 255.255.255.0 172.16.6.1 50
ip route 172.16.4.0 255.255.255.0 172.16.2.1
ip route 172.16.4.0 255.255.255.0 172.16.6.1 50
ip route 172.16.9.0 255.255.255.0 172.16.6.1
ip route 172.16.9.0 255.255.255.0 172.16.2.1 50
ip route 172.16.5.0 255.255.255.0 172.16.6.1
ip route 172.16.5.0 255.255.255.0 172.16.2.1 50
ip route 172.16.8.0 255.255.255.0 172.16.6.1
ip route 172.16.8.0 255.255.255.0 172.16.2.1

```

RTC

```

ip route 172.16.1.0 255.255.255.0 172.16.6.2
ip route 172.16.1.0 255.255.255.0 172.16.5.1
ip route 172.16.4.0 255.255.255.0 172.16.5.1
ip route 172.16.4.0 255.255.255.0 172.16.6.2 50
ip route 172.16.2.0 255.255.255.0 172.16.6.2
ip route 172.16.2.0 255.255.255.0 172.16.5.1 50
ip route 172.16.7.0 255.255.255.0 172.16.6.2
ip route 172.16.7.0 255.255.255.0 172.16.5.1 50
ip route 172.16.8.0 255.255.255.0 172.16.5.1
ip route 172.16.8.0 255.255.255.0 172.16.6.2 50

```

RTD

```

ip route 172.16.1.0 255.255.255.0 172.16.4.1
ip route 172.16.1.0 255.255.255.0 172.16.5.2 50
ip route 172.16.2.0 255.255.255.0 172.16.4.1
ip route 172.16.2.0 255.255.255.0 172.16.5.2 50
ip route 172.16.9.0 255.255.255.0 172.16.5.2
ip route 172.16.9.0 255.255.255.0 172.16.4.1 50
ip route 172.16.6.0 255.255.255.0 172.16.5.2
ip route 172.16.6.0 255.255.255.0 172.16.4.1 50
ip route 172.16.7.0 255.255.255.0 172.16.5.2
ip route 172.16.7.0 255.255.255.0 172.16.4.1

```

第 5 章

1. 除了这里给出的 RIP 配置, 在 RTE 和 RTF 之间还必须使用 2 级地址配置 192.168.5.0 的一个子网。否则子网 192.168.5.192/27 和 192.168.5.96/27 不连续。RIP 配置是:

RTA

```

router rip
network 192.168.2.0

```

RTB

```

router rip

```



```
network 192.168.2.0
```

RTC

```
router rip
network 192.168.2.0
network 192.168.3.0
```

RTD

```
router rip
network 192.168.3.0
network 192.168.4.0
```

RTE

```
router rip
network 192.168.4.0
network 192.168.5.0
```

RTF

```
router rip
network 192.168.4.0
network 192.168.6.0
```

2. 为了在 RTC 和 RTD 之间单播 RIP 更新, 配置为:

RTC

```
router rip
network 192.168.2.0
neighbor 192.168.3.2
```

RTD

```
router rip
network 192.168.3.0
neighbor 192.168.3.1
```

3. 更新时间作用于整个 RIP 进程。如果串行链路的更新时间变了, 路由器其他链路的更新时间也要变。这就意味着邻居路由器的计时器要变化, 邻居路由器的邻居的计时器也要顺次变化, 等等。在一个路由器上改变更新计时器的级联效应导致 RIP 域中每个路由器的计时器都要变化。在每个路由器上增加 RIP 更新周期至 2min 的命令为:

```
timers basic 120 360 360 480
```

因为更新计时器变了, 所以无效计时器、抑制计时器和刷新计时器也必须改变。像缺省的那样, 把无效计时器和抑制计时器设为更新计时器的 6 倍, 将使得这个网络的转换时间很长。所以, 把无效计时器和抑制计时器设为更新计时器的 3 倍。刷新计时器必须比抑制计时器长, 所以这儿把它设成长 60s。

4. 从 RTA 到网络 192.168.4.0 有 2 跳, 所以给度量值增加 14 将会使路由的度量值达到 16 (不可达)。记得在配置练习 1 中, 192.168.5.0 的一个子网必须在和 192.168.4.0 相同的链路上使用 2 级地址来配置, 这样 192.168.5.0 的子网才连续。因此, 从 RTB 到 192.168.5.0 也是 2 跳。假设 RTA 和 RTB 上连接到 RTC 的接口都是 E0, 配置为:

RTA

```
router rip
  offset-list 1 in 14 Ethernet0
  network 192.168.2.0
!
access-list 1 permit 192.168.4.0 0.0.0.0
```

RTB

```
router rip
  offset-list 1 in 14 Ethernet0
  network 192.168.2.0
!
access-list 1 permit 192.168.5.0 0.0.0.0
```

5. RTB 有更长一点的掩码, 能正确解释所有子网。问题在 RTA 上。RTA 有 27 位掩码, 它把 RTB 的子网 192.168.20.40/29 和 192.168.20.49/29 解释为 192.168.20.32/27——与它的直连链路相同的子网。因此, RTA 没有整个网络的正确视图。

但是, 如果代理 ARP 启用了, 报文仍然能被路由。例如, 假设 RTA 要路由一个目的地是 192.168.20.50 的报文。RTA 错误地把这个地址当作是它的子网 192.168.20.32/27 的成员, 并且在该子网上发出 ARP 请求, 要求得到 192.168.20.50 的 MAC 地址。RTB 听到 ARP 请求, 它正确地解释这个地址为它的子网 192.168.20.48/29 的一个成员, 并且回应它在 192.168.20.32/29 上的接口的 MAC 地址。RTA 然后转发报文给 RTB, RTB 转发这个报文至正确目的地。如果代理 ARP 没有启用, 报文将不会被正确地从 RTA 转发至 RTB。

第 6 章

1. 使用 2 级 IP 地址, 192.168.5.0 的一个子网必须被添加至 RTE 和 RTF 的以太网接口, 以连接这两个路由器令牌环接口上的不连续子网。IGRP 配置是:

RTA

```
router igrp 50
  network 192.168.2.0
```

RTB

```
router igrp 50
  network 192.168.2.0
```

RTC

```
router igrp 50
  network 192.168.2.0
  network 192.168.3.0
```

RTD

```
router igrp 50
```



```
network 192.168.3.0
network 192.168.4.0
```

RTE

```
router igrp 50
network 192.168.4.0
network 192.168.5.0
```

RTF

```
router igrp 50
network 192.168.4.0
network 192.168.5.0
```

2. 路由的最小带宽是 RTD 的以太网接口的带宽。整个度量值是每个接口的带宽和延迟之和:

$$1000 + 10 + 7 + 100 + 63 = 1180$$

3. 路由的最小带宽是 RTD 的以太网接口的带宽。因为 K5 非 0, 所以公式为:

$$\text{metric} = [K1 * BW_{\text{IGRP}(\text{min})} + (K2 * BW_{\text{IGRP}(\text{min})}) / (256 - \text{LOAD}) + K3 * DLY_{\text{IGRP}(\text{sum})}] * [K5 / (\text{RELIABILITY} + K4)]$$

K1=K2=K4=K5=1, 且 K3=0, 所以, 代入公式得:

$$\begin{aligned} \text{metric} &= [1000 + 1000/255 + 0] * [1/256] \\ &= 3.922 \end{aligned}$$

丢弃度量值的小数部分, 所以从 192.168.2.96/27 到 192.168.5.96/27 的路由的度量值是 3。

4. 两个 FDDI 网络之间的 5 条路径的最低度量值是 698, 最高度量值是 21541。21541/698 = 30.86, 所以两个路由器的 IGRP 配置中必须增加的命令是: **maximum-paths 5** 以及 **variance 31**。

第 7 章

1. Taos 的 RIP 配置中可增加 **neighbor 172.25.150.206** 语句, 使 Taos 单播 RIP 更新至该地址。该措施仅当下面情况满足时有效。Pojoaque 的 RIPv1 进程符合这样的规则: 版本号高于 1 的 RIP 消息的未用域被忽略, 剩下的报文被处理。

2. 首先计算有最大主机数的子网。然后利用未用的子网位, 计算有次最大主机数的子网, 等等。记住, 当用位组来表示子网时, 没有连续的子网能以相同的位组开始。例如, 如果第一个子网开始于 00, 所有后续的子网必须开始于 01、10 或 11。如果第二个子网开始于 010, 后续的子网不能开始于 010。

一种解答如下 (子网位用黑体表示):

```
00000000 = 192.168.100.0/26 (62 hosts)
01000000 = 192.168.100.64/27 (30 hosts)
01100000 = 192.168.100.96/28 (14 hosts)
01110000 = 192.168.100.112/28 (14 hosts)
```



```

10000000 = 192.168.100.128/28 (14 hosts)
10010000 = 192.168.100.144/28 (14 hosts)
10100000 = 192.168.100.160/28 (14 hosts)
10110000 = 192.168.100.176/29 (8 hosts)
10111000 = 192.168.100.184/29 (8 hosts)
11000000 = 192.168.100.192/29 (8 hosts)
11001000 = 192.168.100.200/29 (8 hosts)
11010000 = 192.168.100.208/29 (8 hosts)
11011000 = 192.168.100.216/30 (2 hosts)
11011100 = 192.168.100.220/30 (2 hosts)
11100000 = 192.168.100.224/30 (2 hosts)
11100100 = 192.168.100.228/30 (2 hosts)
11101000 = 192.168.100.232/30 (2 hosts)
11101100 = 192.168.100.236/30 (2 hosts)
11110000 = 192.168.100.240/30 (2 hosts)
11110100 = 192.168.100.244/30 (2 hosts)
11111000 = 192.168.100.248/30 (2 hosts)
11111100 = 192.168.100.252/30 (2 hosts)

```

3. RTA、RTB 和 RTD 的 RIP 配置中有 **version 2** 语句。另外, RTA 和 RTB 的 RIP 配置中包含 **no auto-summary** 语句。RTA 和 RTB 上连接到子网 192.168.2.64/28 的接口配置了语句 **ip rip send version 1 2** 和 **ip rip receive version 1 2**。RTD 上连接到子网 192.168.2.128/28 的接口配置了语句 **ip rip send version 1** 和 **ip rip receive version 1**。

4. 下面的解答在 RTB 和 RTD 上使用一个密钥链名 *CCIE* 以及一个密钥串 *exercise4*。假设两个路由器的接口都是 S0, RTB 和 RTD 的配置都如下:

```

key chain CCIE
  key 1
    key-string exercise4
  !
interface Serial0
  ip address 192.168.1.15X 255.255.255.252
  ip rip authentication mode md5
  ip rip authentication key-chain CCIE

```

5. 这里的解答假设配置练习 4 中的验证密钥在 1998 年 10 月 31 日午夜启动生效。这里使用的第二个密钥串是 *exercise5a*, 第三个是 *exercise5b*。

```

key chain CCIE
  key 1
    key-string exercise4
    accept-lifetime 00:00:00 Oct 31 1998 00:00:00 Nov 3 1998
    send-lifetime 00:00:00 Oct 31 1998 00:30:00 Nov 3 1998
  key 2
    key-string exercise5a
    accept-lifetime 00:00:00 Nov 3 1998 duration 36000
    send-lifetime 00:00:00 Nov 3 1998 duration 36000
  key 3
    key-string exercise5b
    accept-lifetime 10:00:00 Nov 3 1998 infinite

```



```
send-lifetime 10:00:00 Nov 3 1998 infinite
!  
interface Serial0  
ip address 192.168.1.15X 255.255.255.252  
ip rip authentication mode md5  
ip rip authentication key-chain CCIE
```

第 8 章

1. 不需要进一步的配置，因为 Earhart 上已经启用了自动汇总。
2. EIGRP 配置是：

RTA

```
router eigrp 5  
network 172.16.0.0  
network 172.18.0.0
```

RTB

```
router eigrp 5  
network 172.16.0.0
```

RTC

```
router eigrp 5  
network 172.16.0.0  
network 172.17.0.0
```

3. 下面的解答使用了一个密钥链名 *CCIE* 以及密钥串 *exercise3a* 和 *exercise3b*。假设今天的日期是 1998 年 11 月 30 日，第一个密钥 8:30AM 开始启用，串口的配置为：

```
key chain CCIE  
key 1  
key-string exercise3a  
accept-lifetime 08:30:00 Dec 2 1998 08:30:00 Jan 1 1999  
send-lifetime 08:30:00 Dec 2 1998 08:30:00 Jan 1 1999  
key 2  
key-string exercise3b  
accept-lifetime 08:30:00 Jan 1 1999 infinite  
send-lifetime 08:30:00 Jan 1 1999 infinite  
!  
interface Serial0  
ip address 172.16.3.19X 255.255.255.252  
ip authentication key-chain eigrp 5 CCIE  
ip authentication mode eigrp 5 md5
```

4. RTD 的 EIGRP 配置为：

```
router eigrp 5  
network 172.16.0.0  
network 172.17.0.0
```



```
no auto-summary
```

在 RTC 上自动汇总必须也关掉。

5. 先前的 EIGRP 配置不需要变化。RTE 上的配置为:

```
router igrp 5
network 172.18.0.0
```

RTA 上加入一个同样的配置。如果你愿意, RTA 上也可以使用 **passive-interface** 语句。

6. RTF 上的 EIGRP 配置为:

```
router eigrp 5
network 172.16.0.0
network 172.18.0.0
no auto-summary
```

RTA 在子网 172.18.10.96/27 上的接口上加入语句 **ip summary-address eigrp 5 172.18.10.192 255.255.255.224**。而且, 注意到自动汇总在 RTA 上必须关掉, 因为 RTF 上出现了网络 172.16.0.0。

7. RTA 能送给 RTF 172.16.3.128/25 的汇总地址, 能送给 RTB 172.16.3.0/25 的汇总地址。所有其他的汇总都是自动执行。

第 9 章

1. OSPF 配置为:

RTA

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
```

RTB

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.5.0.0 0.0.255.255 area 5
area 5 virtual-link 10.100.100.9
```

RTC

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.10.0.0 0.0.255.255 area 10
network 10.30.0.0 0.0.255.255 area 30
```

RTD

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.20.0.0 0.0.255.255 area 20
```

RTE

```
router ospf 1
```



```
network 10.0.0.0 0.0.255.255 area 0
network 10.15.0.0 0.0.255.255 area 15
```

RTF

```
router ospf 1
network 10.5.0.0 0.0.255.255 area 5
```

RTG

```
router ospf 1
network 10.10.1.58 0.0.0.0 area 5
```

RTH

```
router ospf 1
network 10.20.100.100 0.0.0.0 area 20
```

RTI

```
router ospf 1
network 10.5.0.0 0.0.255.255 area 5
network 10.35.0.0 0.0.255.255 area 35
area 5 virtual-link 10.100.100.2
```

RTJ

```
router ospf 1
network 10.15.0.0 0.0.255.255 area 15
```

RTK 到 RTN 有帧中继接口。到帧中继网络的 4 个接口都在同一个子网上，因此 OSPF 网络类型必须是广播或是点到多点的：

RTK

```
interface Serial0
encapsulation frame-relay
ip address 10.30.254.193 255.255.255.192
ip ospf network point-to-multipoint
!
router ospf 1
network 10.30.0.0 0.0.255.255 area 30
```

RTL

```
encapsulation frame-relay
ip address 10.30.254.194 255.255.255.192
ip ospf network point-to-multipoint
!
router ospf 1
network 10.30.0.0 0.0.255.255 area 30
```

RTM

```
encapsulation frame-relay
ip address 10.30.254.195 255.255.255.192
ip ospf network point-to-multipoint
!
router ospf 1
network 10.30.0.0 0.0.255.255 area 30
```


RTN

```
encapsulation frame-relay
ip address 10.30.254.196 255.255.255.192
ip ospf network point-to-multipoint
!
router ospf 1
network 10.30.0.0 0.0.255.255 area 30
```

2. ABR 配置为:**RTB**

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.5.0.0 0.0.255.255 area 5
area 5 virtual-link 10.100.100.9
area 0 range 10.0.0.0 255.255.0.0
area 5 range 10.5.0.0 255.255.0.0
```

RTC

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.10.0.0 0.0.255.255 area 10
network 10.30.0.0 0.0.255.255 area 30
area 0 range 10.0.0.0 255.255.0.0
area 10 range 10.10.0.0 255.255.0.0
area 30 range 10.30.0.0 255.255.0.0
```

RTD

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.20.0.0 0.0.255.255 area 20
area 0 range 10.0.0.0 255.255.0.0
area 20 range 10.20.0.0 255.255.0.0
```

RTE

```
router ospf 1
network 10.0.0.0 0.0.255.255 area 0
network 10.15.0.0 0.0.255.255 area 15
area 0 range 10.0.0.0 255.255.0.0
area 15 range 10.15.0.0 255.255.0.0
```

RTI

```
router ospf 1
network 10.5.0.0 0.0.255.255 area 5
network 10.35.0.0 0.0.255.255 area 35
area 5 virtual-link 10.100.100.2
area 0 range 10.0.0.0 255.255.0.0
area 5 range 10.5.0.0 255.255.0.0
area 35 range 10.35.0.0 255.255.0.0
```


3. 配置为:

RTE

```
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.15.0.0 0.0.255.255 area 15
 area 15 stub
```

RTJ

```
router ospf 1
 network 10.15.0.0 0.0.255.255 area 15
 area 15 stub
```

4. 配置为:

RTC

```
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.10.0.0 0.0.255.255 area 10
 network 10.30.0.0 0.0.255.255 area 30
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0
 area 30 stub no-summary
 area 30 range 10.30.0.0 255.255.0.0
```

RTK

```
router ospf 1
 network 10.30.0.0 0.0.255.255 area 30
 area 30 stub
```

RTL

```
router ospf 1
 network 10.30.0.0 0.0.255.255 area 30
 area 30 stub
```

RTM

```
router ospf 1
 network 10.30.0.0 0.0.255.255 area 30
 area 30 stub
```

RTN

```
router ospf 1
 network 10.30.0.0 0.0.255.255 area 30
 area 30 stub
```

5. 配置为:

RTD

```
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.20.0.0 0.0.255.255 area 20
 area 20 nssa
```


RTH

```
router ospf 1
 network 10.20.100.100 0.0.0.0 area 20
 area 20 nssa
```

6. 点到点线路只有一个端点需要被配置为按需电路。在本题解答中, RTC 被选择:

```
interface Serial0
 ip address 10.30.255.249 255.255.255.252
 ip ospf demand-circuit
!
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.10.0.0 0.0.255.255 area 10
 network 10.30.0.0 0.0.255.255 area 30
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0
 area 30 range 10.30.0.0 255.255.0.0
```

第 10 章

1. 本解答使用 System ID 0000.1234.abcX, X 使得路由器在 IS-IS 域中惟一。System ID 有 6 个 8bit 字节, 是 Cisco IOS 所要求的。最小的 NET 长度是 8 个 8bit 字节, 其中一个是 SEL, 所以区域地址是一个 8bit 字节的数字, 对应于表 10.6 中的区域号。配置为:

RTA

```
interface Ethernet0
 ip address 192.168.1.17 255.255.255.240
 ip router isis
!
interface Ethernet1
 ip address 192.168.1.50 255.255.255.240
 ip router isis
 clns router isis
!
router isis
 net 00.0000.1234.abcl.00
 is-type level-1
```

RTB

```
interface Ethernet0
 ip address 192.168.1.33 255.255.255.240
 ip router isis
!
interface Ethernet1
 ip address 192.168.1.51 255.255.255.240
 ip router isis
```



```
    clns router isis
!
router isis
  net 00.0000.1234.abc2.00
  is-type level-1
```

RTC

```
interface Ethernet0
  ip address 192.168.1.49 255.255.255.240
  ip router isis
  clns router isis
!
interface Serial0
  ip address 192.168.1.133 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc3.00
```

RTD

```
interface Serial0
  ip address 192.168.1.134 255.255.255.252
  ip router isis
!
interface Serial1
  ip address 192.168.1.137 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc4.00
  is-type level-2-only
```

RTE

```
interface Serial0
  ip address 192.168.1.142 255.255.255.252
  ip router isis
!
interface Serial1
  ip address 192.168.1.145 255.255.255.252
  ip router isis
!
interface Serial2
  ip address 192.168.1.138 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc5.00
  is-type level-2-only
```

RTF

```
interface Serial0
  ip address 192.168.1.141 255.255.255.252
```



```
ip router isis
!
interface Serial1
ip address 192.168.1.158 255.255.255.252
ip router isis
!
router isis
net 00.0000.1234.abc6.00
is-type level-2-only
```

RTG

```
interface Ethernet0
ip address 192.168.1.111 255.255.255.224
ip router isis
ip router clns
!
interface Serial0
ip address 192.168.1.157 255.255.255.252
ip router isis
!
router isis
net 00.0000.1234.abc7.00
```

RTH

```
interface Ethernet0
ip address 192.168.1.73 255.255.255.224
ip router isis
!
interface Ethernet1
ip address 192.168.1.97 255.255.255.224
ip router isis
ip router clns
!
router isis
net 00.0000.1234.abc8.00
is-type level-1
```

RTI

```
interface Ethernet0
ip address 192.168.1.225 255.255.255.248
ip router isis
!
interface Ethernet1
ip address 192.168.1.221 255.255.255.248
ip router isis
clns router isis
!
interface Serial0
ip address 192.168.1.249 255.255.255.252
ip router isis
clns router isis
!
```



```
interface Serial1
  ip address 192.168.1.146 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc9.00
```

RTJ

```
interface Ethernet0
  ip address 192.168.1.201 255.255.255.248
  ip router isis
!
interface Ethernet1
  ip address 192.168.1.217 255.255.255.248
  ip router isis
  clns router isis
!
router isis
  net 00.0000.1234.abca.00
  is-type level-1
```

RTK

```
interface Ethernet0
  ip address 192.168.1.209 255.255.255.248
  ip router isis
!
interface Serial0
  ip address 192.168.1.250 255.255.255.252
  ip router isis
  clns router isis
!
router isis
  net 00.0000.1234.abcb.00
  is-type level-1
```

2. 配置为:

RTD

```
interface Serial0
  ip address 192.168.1.134 255.255.255.252
  ip router isis
!
interface Serial1
  ip address 192.168.1.137 255.255.255.252
  ip router isis
  isis password Eiffel level-2
!
router isis
  net 00.0000.1234.abc4.00
  is-type level-2-only
```

RTE


```
interface Serial0
  ip address 192.168.1.142 255.255.255.252
  ip router isis
  isis password Tower level-2
!
interface Serial1
  ip address 192.168.1.145 255.255.255.252
  ip router isis
!
interface Serial2
  ip address 192.168.1.138 255.255.255.252
  ip router isis
  isis password Eiffel level-2
!
router isis
  net 00.0000.1234.abc5.00
  is-type level-2-only
```

RTF

```
interface Serial0
  ip address 192.168.1.141 255.255.255.252
  ip router isis
  isis password Tower level-2
!
interface Serial1
  ip address 192.168.1.158 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc6.00
  is-type level-2-only
```

3. 配置为:

RTG

```
interface Ethernet0
  ip address 192.168.1.111 255.255.255.224
  ip router isis
  ip router clns
!
interface Serial0
  ip address 192.168.1.157 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc7.00
  area-password Scotland
```

RTH

```
interface Ethernet0
  ip address 192.168.1.73 255.255.255.224
  ip router isis
```



```
!  
interface Ethernet1  
  ip address 192.168.1.97 255.255.255.224  
  ip router isis  
  ip router clns  
!  
router isis  
  net 00.0000.1234.abc8.00  
  is-type level-1  
  area-password Scotland
```

4. 配置为:

RTC

```
interface Ethernet0  
  ip address 192.168.1.49 255.255.255.240  
  ip router isis  
  clns router isis  
!  
interface Serial0  
  ip address 192.168.1.133 255.255.255.252  
  ip router isis  
!  
router isis  
  net 00.0000.1234.abc3.00  
  domain-password Vienna
```

RTD

```
interface Serial0  
  ip address 192.168.1.134 255.255.255.252  
  ip router isis  
!  
interface Serial1  
  ip address 192.168.1.137 255.255.255.252  
  ip router isis  
  isis password Eiffel level-2  
!  
router isis  
  net 00.0000.1234.abc4.00  
  is-type level-2-only  
  domain-password Vienna
```

RTE

```
interface Serial0  
  ip address 192.168.1.142 255.255.255.252  
  ip router isis  
  isis password Tower level-2  
!  
interface Serial1  
  ip address 192.168.1.145 255.255.255.252  
  ip router isis  
!
```



```
interface Serial2
  ip address 192.168.1.138 255.255.255.252
  ip router isis
  isis password Eiffel level-2
!
router isis
  net 00.0000.1234.abc5.00
  is-type level-2-only
  domain-password Vienna
```

RTF

```
interface Serial0
  ip address 192.168.1.141 255.255.255.252
  ip router isis
  isis password Tower level-2
!
interface Serial1
  ip address 192.168.1.158 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc6.00
  is-type level-2-only
  domain-password Vienna
```

RTG

```
interface Ethernet0
  ip address 192.168.1.111 255.255.255.224
  ip router isis
  ip router clns
!
interface Serial0
  ip address 192.168.1.157 255.255.255.252
  ip router isis
!
router isis
  net 00.0000.1234.abc7.00
  area-password Scotland
  domain-password Vienna
```

RTI

```
interface Ethernet0
  ip address 192.168.1.225 255.255.255.248
  ip router isis
!
interface Ethernet1
  ip address 192.168.1.221 255.255.255.248
  ip router isis
  clns router isis
!
interface Serial0
  ip address 192.168.1.249 255.255.255.252
```



```
ip router isis
clns router isis
!
interface Serial1
ip address 192.168.1.146 255.255.255.252
ip router isis
!
router isis
net 00.0000.1234.abc9.00
domain-password Vienna
```

5. 配置为:

RTC

```
interface Ethernet0
ip address 192.168.1.49 255.255.255.240
ip router isis
clns router isis
!
interface Serial0
ip address 192.168.1.133 255.255.255.252
ip router isis
!
router isis
net 00.0000.1234.abc3.00
domain-password Vienna
summary-address 192.168.1.0 255.255.255.192
```

RTG

```
interface Ethernet0
ip address 192.168.1.111 255.255.255.224
ip router isis
ip router clns
!
interface Serial0
ip address 192.168.1.157 255.255.255.252
ip router isis
!
router isis
net 00.0000.1234.abc7.00
area-password Scotland
domain-password Vienna
summary-address 192.168.1.64 255.255.255.192
```

RTI

```
interface Ethernet0
ip address 192.168.1.225 255.255.255.248
ip router isis
!
interface Ethernet1
ip address 192.168.1.221 255.255.255.248
ip router isis
```



```

    clns router isis
!
interface Serial0
    ip address 192.168.1.249 255.255.255.252
    ip router isis
    clns router isis
!
interface Serial1
    ip address 192.168.1.146 255.255.255.252
    ip router isis
!
router isis
    net 00.0000.1234.abc9.00
    domain-password Vienna
    summary-address 192.168.1.192 255.255.255.192

```

第 11 章

1. 当运行有类别路由选择协议时, 图 11.37 中的互联网络困难在于, 网络 172.16.0.0 是被可变子网化的。解决办法是在 RTB 的 E1 接口上使用 28 位掩码, 同时 RTC 在相同的子网上仍使用 27 位掩码。此办法可行, 因为 172.16.1.96 在图 11.37 中的互联网络环境中惟一, 无论它是用 27 位还是 28 位掩码被子网化。

RTB 的配置为:

```

interface Ethernet0
    ip address 172.16.1.146 255.255.255.240
!
interface Ethernet1
    ip address 172.16.1.98 255.255.255.240
!
router rip
    redistribute igrp 1 metric 1
    passive-interface Ethernet0
    network 172.16.0.0
!
router igrp 1
    redistribute rip metric 10000 1000 255 1 1500
    passive-interface Ethernet1
    network 172.16.0.0

```

2. 这个配置将 RTB 的 E1 掩码改回了 27 位。但是, RIP 不宣告子网 172.16.1.144/28。为了纠正这个问题, 对这个子网加一条带 27 位掩码的静态路由。因为此掩码涉及 RTB 的 E0 接口, 所以该路由被自动分配至 RIP 中。

```

interface Ethernet0
    ip address 172.16.1.146 255.255.255.240
!
interface Ethernet1

```



```
ip address 172.16.1.98 255.255.255.224
!
router ospf 1
 redistribute rip metric 50 subnets
 network 172.16.1.0 0.0.0.255 area 0
!
router rip
 redistribute ospf 1 metric 2
 passive-interface Ethernet0
 network 172.16.0.0
!
ip classless
ip route 172.16.1.128 255.255.255.224 Ethernet0
```

3. RTB 的配置为:

```
interface Ethernet0
 ip address 172.16.1.146 255.255.255.240
 ip summary-address eigrp 1 172.16.2.0 255.255.255.0
!
interface Ethernet1
 ip address 172.16.1.98 255.255.255.224
!
router eigrp 1
 redistribute isis level-2 metric 10000 1000 255 1 1500
 passive-interface Ethernet1
 network 172.16.0.0
!
router isis
 net 00.0000.1234.abcd.00
 summary-address 172.16.1.128 255.255.255.128
 redistribute eigrp 1 metric 20 metric-type external level-2
```

第 13 章

1. RTA 的配置为:

```
router rip
 redistribute igrp 1 metric 3
 passive-interface Ethernet0
 passive-interface Ethernet1
 network 172.16.0.0
 distribute-list 1 in Ethernet3
!
router igrp 1
 redistribute rip metric 10000 1000 255 1 1500
 passive-interface Ethernet2
 passive-interface Ethernet3
 network 172.16.0.0
!
```



```
access-list 1 deny 172.16.12.0
access-list 1 permit any
```

2. RTA 的配置为:

```
router rip
 redistribute igrp 1 metric 3
 passive-interface Ethernet0
 passive-interface Ethernet1
 network 172.16.0.0
 distribute-list 2 out Ethernet2
!
router igrp 1
 redistribute rip metric 10000 1000 255 1 1500
 passive-interface Ethernet2
 passive-interface Ethernet3
 network 172.16.0.0
!
access-list 2 deny 172.16.10.0
access-list 2 permit any
```

3. RTA 的配置为:

```
router rip
 redistribute igrp 1 metric 3
 passive-interface Ethernet0
 passive-interface Ethernet1
 network 172.16.0.0
 distribute-list 3 out igrp 1
!
router igrp 1
 redistribute rip metric 10000 1000 255 1 1500
 passive-interface Ethernet2
 passive-interface Ethernet3
 network 172.16.0.0
!
access-list 3 permit 172.16.2.0
access-list 3 permit 172.16.8.0
access-list 3 permit 172.16.9.0
```

4. RTA 的配置为:

```
router rip
 redistribute igrp 1 metric 3
 passive-interface Ethernet0
 passive-interface Ethernet1
 network 172.16.0.0
!
router igrp 1
 redistribute rip metric 10000 1000 255 1 1500
 passive-interface Ethernet2
 passive-interface Ethernet3
 network 172.16.0.0
```



```

    distribute-list 4 out Ethernet0
!
access-list 4 permit 172.16.1.0
access-list 4 permit 172.16.2.0
access-list 4 permit 172.16.3.0
access-list 4 permit 172.16.7.0
access-list 4 permit 172.16.8.0
access-list 4 permit 172.16.9.0

```

5. EIGRP 为外部路由分配高距离值 170，给内部路由分配低距离值 90。如果 EIGRP 域内的任何目标地址从 IS-IS 被重新分配至 EIGRP，将会被忽略，除非内部路由失效。所以，不要求手动操作 EIGRP 距离。RTC 和 RTD 的 IS-IS 配置中的 **distance** 语句为：

RTC

```

distance 115
distance 170 192.168.10.254 0.0.0.0 1
!
access-list 1 permit any

```

RTD

```

distance 115
distance 170 192.168.10.249 0.0.0.0 1
distance 170 192.168.10.241 0.0.0.0 1
!
access-list 1 permit any

```

6. RTD 中 IS-IS 配置的 **distance** 语句是：

```

distance 115
distance 255 192.168.10.241 0.0.0.0 1
!
access-list 1 permit any

```

7. RTC 中 EIGRP 配置的 **distance** 语句是：

```

distance eigrp 90 90

```

第 14 章

1. RTA 的配置为：

```

interface Serial0
    ip address 172.16.14.6 255.255.255.252
    ip policy route-map Exercisel
!
interface Serial1
    ip address 172.16.14.10 255.255.255.252
    ip policy route-map Exercisel
!

```



```
access-list 1 permit 172.16.1.0 0.0.0.127
access-list 2 permit 172.16.1.128 0.0.0.127
!
route-map Exercise1 permit 10
  match ip address 1
  set ip next-hop 172.16.14.17
!
route-map Exercise1 permit 20
  match ip address 2
  set ip next-hop 172.16.14.13
```

2. RTA 的配置为:

```
interface Serial0
  ip address 172.16.14.6 255.255.255.252
  ip policy route-map Exercise2A
!
interface Serial1
  ip address 172.16.14.10 255.255.255.252
  ip policy route-map Exercise2B
!
access-list 1 permit 172.16.1.0 0.0.0.127
!
route-map Exercise2A permit 10
  match ip address 1
  set ip next-hop 172.16.14.13
!
route-map Exercise2B permit 10
  match ip address 1
  set ip next-hop 172.16.14.17
```

3. RTA 的配置为:

```
interface Serial2
  ip address 172.16.14.18 255.255.255.252
  ip access-group 101 in
  ip policy route-map Exercise3
!
interface Serial3
  ip address 172.16.14.14 255.255.255.252
  ip access-group 101 in
  ip policy route-map Exercise3
!
access-list 101 permit udp any 172.168.1.0 0.0.0.255
access-list 101 permit tcp any eq smtp 172.16.1.0 0.0.0.255
access-list 102 permit tcp any eq smtp 172.16.1.0 0.0.0.255
access-list 103 permit udp any 172.168.1.0 0.0.0.255
!
route-map Exercise3 permit 10
  match ip address 102
  set ip next-hop 172.16.14.9
!
```



```
route-map Exercise3 permit 20
  match ip address 103
  set ip next-hop 172.16.14.5
```

4. OSPF 配置为:

```
router ospf 1
  redistribute eigrp 1 route-map Exercise4
  network 192.168.1.0 0.0.0.255 area 16
!
access-list 1 deny 10.201.100.0
access-list 1 permit any
!
route-map Exercise4 permit 10
  match ip address 1
!
route-map Exercise4 permit 20
  match route-type internal
  set metric 10
  set metric-type type-1
!
route-map Exercise4 permit 30
  match route-type external
  set metric 50
  set metric-type type-2
```

5. EIGRP 配置为:

```
router eigrp 1
  redistribute ospf 1 route-map Exercise5
  network 192.168.100.0
!
access-list 1 permit 192.168.1.0
access-list 1 permit 192.168.2.0
access-list 1 permit 192.168.3.0
!
route-map Exercise5 permit 10
  match ip address 1
!
route-map Exercise5 permit 20
  match route-type internal
  set metric 10000 100 255 1 1500
!
route-map Exercise5 permit 30
  match route-type external
  set metric 10000 10000 255 1 1500
```


附录 F

故障排除练习答案

第 2 章

1. 子网: 10.14.64.0

主机地址: 10.14.64.1 - 10.14.95.254

广播地址: 10.14.95.255

子网: 172.25.0.224

主机地址: 172.25.0.225 - 175.25.0.254

广播地址: 172.25.0.255

子网: 172.25.16.0

主机地址: 172.25.16.1 - 172.25.16.126

广播地址: 172.25.16.127

2. 192.168.13.175/28 是子网 192.168.13.160/28 的广播地址。

第 3 章

1. 从 Piglet 到子网 192.168.1.64/27 不再可达, 从 Piglet 到子网 10.4.6.0/24 和 10.4.7.0/24 也不再可达。

2. 错误发生在:

RTA, 第 2 个表项。

RTB, 第 3 个表项。

RTC, 第 2 个表项。

RTC, 第 5 个表项。

3. 错误是:

RTC: 到 10.5.8.0/24 的路由指向了错误的下一跳地址。

RTC: 到 10.1.1.0/24 的路由应该是 10.5.1.0/24。

RTC: 到 10.5.4.0/24 没有路由。

RTD: 到 10.4.5.0/24 的路由应该是 10.5.4.0/24。

第 5 章

1. 新的访问列表给每一条路由 (除了 10.33.32.0) 添加了两跳。

2. RTB 根据其错误配置的掩码解释 172.16.0.0 的所有子网。结果, 正如它路由表中的 4 个表项所示:

- 表项 1: 正确。因为 172.16.24.0 掩码既可以是 22 位也可以是 23 位。
- 表项 2: RTC 宣告子网 172.16.26.0。因为该地址的第 23 位是 1 且 RTB 使用 22 位的掩码, 从 RTB 的角度来看这个 1 出现在主机地址部分。所以, RTB 把 172.16.26.0 的宣告看作是一条主机路由, 并且在路由表里将其掩码设为 32 位。
- 表项 3: RTB 将其接口地址 172.16.22.5 解释为子网 172.16.20.0/22 的一部分, 而不是 172.16.22.0/23 的一部分。当 RTB 收到 RTA 关于子网 172.16.20.0/23 的宣告时, 由于 RTB 认为自己与该子网直接相连, 于是忽略了该通告。注意, RTA 和 RTC 的路由表里没有子网 172.16.22.0, 这是由于同一个原因: RTB 宣告它为 172.16.20.0。
- RTB 将其接口地址 172.16.18.4 解释为子网 172.16.16.0/22 的一个成员, 而不是 172.16.18.0/23。

3. 答案在图 5.24 中 RTC 的路由表里。注意到 172.16.26.0/23 的路由在 2min42s 内没有更新。RTC 的无效计时器在它听到来自 RTD 的一条新的更新之前就超时了, 于是它宣布到 172.16.26.0/23 的路由无效。因为没有从 RTA 或 RTB 来的路由被宣布无效, 所以问题出在 RTD 的 update 计时器上一更新周期太长。当 RTD 最终送出一条更新, 它再次出现在 RTC 的路由表里, 并且保留到 RTC 的无效计时器再次超时。

第 6 章

1. RTB 接收 RTD 关于 192.168.3.0/24 的路由更新, 但是不包括它向 RTA 发出的更新中的子网。允许这条路由在 RTB 上的接收, 但不允许该路由的宣告。可能是一个错误配置的路由过滤。

2. 答案在图 6-40 里。RTC 已被配置为 **router igrp 51**, 而不是 **router igrp 15**。

第 7 章

1. RTA 和 RTB 只发送和接收 RIPv2 消息。RTC 接受 RIPv1 和 RIPv2 消息, 但只发送

RIPv1 消息。结果，RTA 和 RTB 的路由表里没有子网 192.168.13.75/27。虽然 RTC 接收来自 RTA 和 RTB 的更新，但它的路由表里没有子网 192.168.13.90/28 和 192.168.13.86/29，因为 RTC 把它们解释成 192.168.13.64/27，这是与 RTC 的 E0 接口直接相连的。

2. 是的。RTA 和 RTB 的路由表中将会加入子网 192.168.13.64/27，因为它们现在能接收来自 RTC 的 RIPv1 更新。

第 8 章

1. EIGRP 和 IGRP 进程的自主系统号是不同的。
2. 在到子网 A 的路径上 RTG 是 RTF 的后继。
3. RTC 到子网 A 的可行距离是 309760。
4. RTG 到子网 A 的可行距离是 2214319。
5. RTG 的拓扑表显示，RTD 和 RTE 是它到子网 A 的可行后继。
6. RTA 到子网 B 的可行距离是 2198016。

第 9 章

1. 在一个邻居接口上，IP 地址或者其掩码配置出错。
2. 一个路由器被配置为末梢区域路由器，另一个不是。
3. 接收端路由器被配置为 MD5 (type 2) 验证，其邻居没有配置认证 (type 0)。
4. 两个路由器上配置的密码不一致。
5. 邻居接口没有被配置相同的 area ID。
6. RTA 的 **network** 语句次序错误。第一个语句匹配 IP 地址 192.168.50.242，并把它放入区域 192.168.50.0。
7. Link ID 10.8.5.1 最可疑，因为它的序列号比其他链路的序列号要高很多。

第 10 章

1. 虽然 IS-IS 形成了邻接，但是 IP 地址并不在同一个子网里，所以接口不通过流量。
2. 路由器只发送 L1 Hello，表明它是一个 L1 路由器。它只从 0000.3090.c7df 接收 L2 Hello，表明它是一个 L2 路由器。

第 11 章

1. 水平分割将阻止 192.168.10.0/24 从 IGRP 域到 RIP 域的宣告。
2. 不。因为不是所有 10.0.0.0 的子网都在 RIP 端与 Mantle 直接相连。

3. Robinson 的 EIGRP1 配置中的 **network** 语句匹配接口 192.168.3.32, 即使没有 EIGRP 的 Hello 在这个接口被转发出去。因此, 这个子网在 EIGRP1 域内被声明为内部的。

4. EIGRP 自动加入这条汇总路由, 因为子网 192.168.3.0 是从 OSPF 重新分配到 EIGRP 中的。

5. 192.168.1.0/24 在 OSPF 域中, OSPF 不被重新分配到 EIGRP2 中。不像 OSPF, EIGRP 使用水平分割。因此, 虽然 192.168.1.0/24 被宣告至 EIGRP1, 它将不被从 EIGRP1 通过 Robinson 重新分配至 EIGRP2。

第 13 章

1. 访问列表的第一行中的反码错误。由于此反码, 所有路由将被匹配。这一行应该是 **access-list 1 deny 0.0.0.0 0.0.0.0**。

2. 所有没有被访问列表匹配的路由将给予距离值 255 (不可到达)。如果一个优先路由失败, Grimwig 将不使用备用路径。

3. OSPF 通过它从 LSA 学来的信息计算路由。路由过滤不影响 LSA, 所以过滤仅仅影响配置它的路由器。

4. 这两个路由过滤涉及错误的接口。**distribute-list 1** 应该对应 E1, **distribute-list 2** 应该对应 E0。

第 14 章

1. 两个错误都是顺序错误。首先, access list 101 中的 *telnet* 关键字与目标端口关联; 正确的应该是源端口。其次, 路由图语句顺序错误。从 192.168.10.5 来的 telnet 报文匹配第一个语句, 并且被转发至 192.168.16.254。

索引

access-list access-list-number {deny|permit}, 554
access-list access-list-number{deny|permit} source[source-wildcard], 552
AllDRouters, 262
AllL1ISs, 385
AllL2ISs, 385
AllSPFRouters, 261
Area ID (区域 ID), 266
ASBR 汇总 LSA (ASBR Summary LSA), 297
ASBR 汇总 LSA, 319
ATT, 377
A 类 IP 地址, 29
BGP 协议, 142
Boolean AND, 553
B 类 IP 地址, 29
CDP, 474
CIDR 块 (CIDR block), 235
Cisco 发现协议, 474
Cisco 访问列表编号, 551
Cisco 缺省管理距离, 442
clear ip accounting, 563
CLNP, 373
CLV 代码, 392
CLV 字段, 391
Cost (代价), 267
CSNP, 405
C 类 IP 地址, 29

- Database Summary List (数据库摘要列表), 273
- DD Sequence Number (数据库描述序列号), 273
- default-information originate, 481
- default-information-originate, 340
- Dijkstra 算法, 102
- distribute-list, 493
- DRothers 路由器, 265
- DUAL 算法, 210
- DUAL 有限状态机, 215
- DUAL 有限状态机的输入事件, 218
- EIGRP 协议, 205
- EPHOS, 374
- ES, 375
- ES-IS, 375
- Exchange 状态, 277
- Forwarding process, 383
- GOSIP, 374
- Hello 报文, 311
- Hello 报文协议, 260
- Hello 时间间隔 (HelloInterval), 260
- IAB, 374
- ICMP 报文类型字段和代码字段, 44
- ICMP 访问列表, 557
- ICMP 路由器发现协议 (IRDP), 46
- IGRP 进程域, 142
- IGRP 协议, 141
- Internet 工程任务组 (Internet Engineering Task Force, IETF), 258
- Internet 消息控制协议 (ICMP), 44
- ip access-group access-list-number {in|out}**, 558
- ip access-list {standard|extended} name**, 560
- ip accounting access-violation, 563
- IPX SAP, 551
- IP 版本号, 21
- IP 报文头, 21
- IP 地址, 28
- IP 接口地址 CLV, 398
- IP 内部可达性信息 CLV, 402
- IP 外部可达性信息 CLV, 403
- IS-IS Hello PDU 报文, 392
- IS-IS, 373
- IS-IS 区域, 376
- IS-IS 网络类型, 380
- IS-IS 协议链路状态 PDU 报文, 399
- IS-IS 协议序列号 PDU 报文, 405
- keepalive 报文, 259
- L1/L2 路由器, 376
- L1 路由器, 376
- L2 路由器, 376
- LAN ID, 382
- Link State Request List (链路状态请求列表), 273
- Link State Retransmission List (链路状态重传列表), 273
- loopback 地址, 260
- LSA 格式, 315
- LSA 类型, 293
- LSA 头部, 315
- LSP, 376
- LSP 条目 CLV, 406
- MAC 地址, 7
- MOSPF 协议, 299
- Neighbor ID (邻居路由器 ID), 272
- Neighbor Priority (邻居优先级), 272
- NET, 378
- NLPID, 398
- NSAP 选择符 (SEL), 378
- NSSA 外部 LSA (NSSA External LSA), 299
- NSSA 外部 LSA, 320
- OL 位, 385
- On-Demand Routing, ODR, 474
- Opaque LSA, 300
- OSPF TOS, 318
- OSPF 报文, 308
- OSPF 报文类型, 310
- OSPF 接口, 266
- OSPF 接口数据结构, 266

- OSPF 接口状态机, 269
- OSPF 邻居, 271
- OSPF 认证类型, 310
- PDU, 375
- PDU 报文类型, 390
- Process ID (进程 ID), 266
- Pseudonode ID, 381
- PSNP, 384
- PUP 协议, 114
- Receive process, 383
- RIP_JITTER, 115
- RIPE, 235
- RIPv1, 113
- RIPv2 协议, 175
- RIP 协议, 113
- SEL, 378
- show ip accounting access-violations, 563
- SNP, 382
- SNPA, 375
- SPF 算法, 102
- SPF 算法树, 259
- stuck-in-active, SIA, 217
- TCP 报头格式, 48
- TCP 访问列表, 556
- TLV, 228
- UDP 报头格式, 50
- UDP 访问列表, 557
- Xerox 网络系统(XNS), 114
- XNS RIP, 114
- 安全过滤, 547
- 按需路由, 474
- 棒棒糖形序列号空间, 99
- 报头校验和 (Header Checksum), 25
- 备份指定路由器 (Backup Designated Router, BDR), 264
- 被动接口 (Passive Interfaces), 124
- 本地电路 ID (Local Circuit.ID), 395
- 本地计算 (local computation), 216
- 避免路由环路, 446
- 边界汇总 (Boundary Summarization), 118
- 编辑访问列表, 552
- 标记 (Flag), 49
- 标记字段 (Flag), 23
- 标准 IP, 551
- 标准 IPX, 551
- 标准访问列表, 552
- 不关心位, 553
- 布尔或, 553
- 部分序列号报文, 384, 405
- 策略路由, 513
- 层 1 路由器 (Level 1 Router), 376
- 层 2 路由器 (Level 2 Router), 376
- 查询始发标记 (Query origin flag (O)), 217
- 差异变量 (Variance), 154
- 超时计时器(Timeout Timer), 115
- 超网 (supernet), 232
- 超网, 180
- 超载 (Overload, OL), 385
- 出站 (outgoing) 路由更新, 131
- 出站过滤器, 561
- 初始位 (Initial bit), 277, 313
- 触发更新(Triggered Update), 94
- 传输控制协议 (TCP), 47
- 传送网络 (Transit Network), 262
- 窗口大小 (WindowSize), 49
- 窗口机制, 48
- 答复状态标记 (reply status flag(r)), 216
- 大型互联网络, 29
- 代价 (Cost), 86
- 代理 ARP, 42
- 带宽 (Bandwidth), 146
- 带宽 (Bandwidth) 度量, 86
- 单播更新(Unicast update), 125
- 等价均分负载 (Equal-Cost Load Sharing), 65
- 地址簇标识 (Address Family Identifier, AFI), 117
- 地址解析协议 (ARP), 38
- 地址聚合, 180
- 地址聚合, 232
- 地址掩码 (Adresse Mask), 30

- 递归表查询, 67
- 点到点网络 (Point-to-Point), 261
- 点到多点网络 (Point-to-Multipoint), 262
- 点分十进制法, 28
- 电路 ID (Circuit ID), 381
- 调用访问列表 in, 558
- 调用访问列表 out, 558
- 调用访问列表, 558
- 定期更新 (Periodic Updates), 89
- 动态路由选择协议, 83
- 毒性逆转水平分隔, 92
- 度量, 441
- 度量, 58
- 端口号, 48
- 端系统, 375
- 队列计数 (Q Count), 209
- 二进制 (binary), 542
- 反码, 553
- 反向 ARP (RARP), 43
- 反向 DNS 查找, 327
- 泛洪 (Flooding), 281
- 泛洪扩散 (Flooding), 96
- 访问列表, 548
- 访问列表的监视和计费, 563
- 访问列表类型, 550
- 非纯末梢区域 (Not-so-stubby-area, NSSA), 302
- 非等价均分负载 (Unequal-Cost Load Sharing), 65
- 非广播多路访问 (NBMA), 209
- 非广播型多路访问网络 (NBMA), 262
- 分段区域 (Partitioned Area), 288
- 分片偏移 (Fragment Offset), 24
- 服务类型 (TOS), 22
- 浮动静态路由 (Floating Static Route), 63
- 辅助 IP 地址 (Secondary IP Address), 128
- 负载 (Load), 148
- 负载 (Load) 度量, 86
- 更新处理过程 (Update Process), 383
- 更新处理过程, 383
- 更新计时器 (Update Timer), 115
- 孤立区域 (Isolated Area), 288
- 骨干路由器 (Backbone Routers), 288
- 骨干区域 (Backbone Area), 287
- 关键字 any, 555
- 关键字 established, 556
- 关键字 host, 555
- 关联路由器 (Attached Router), 319
- 管理关闭 (administratively shutdown), 327
- 管理距离, 442
- 管理距离, 64
- 广播, 12
- 广播更新 (Broadcast Updates), 89
- 广播型网络 (Broadcast), 261
- 过程交换 (Process Switching), 67
- 过滤规则的放置顺序, 549
- 后继路由器 (Successor), 211
- 后继位 (More bit), 277, 313
- 候选对象数据库, 103
- 呼叫协议 (Hello Protocol), 96
- 互连网络层, 20
- 汇总路由 (Summary Route), 61
- 活动计时器 (active timer), 216
- 基于报文的均分负载, 67
- 基于目标网络的负载均衡, 66
- 集成 IS-IS 协议, 374
- 记录路由选项 (Record Route), 25
- 加密校验和 (cryptographic checksum), 184
- 加权公平队列 (Weighted Fair Queuing, WFQ), 309
- 加权随机预先检测 (Weighted Random Early Detection, WRED), 309
- 简单静态路由, 58
- 接口 MTU (Interface MTU), 312
- 接口状态机的输入事件, 271
- 紧急指针 (Urgent Point), 49
- 进程 ID 号 (process ID), 153
- 进程域 (Process Domain), 142
- 静态路由, 58
- 局域网, 6

- 拒绝, 548
- 距离矢量 (Distance Vector), 88
- 决策处理过程, 386
- 均分负载 (Load Sharing), 65
- 开放最短路径优先协议 (Open Shortest Path First, OSPF), 258
- 可变长子网掩码 (VLSM), 180
- 可靠传输协议 (Reliable Transport Protocol), 208
- 可靠性 (Reliability), 148
- 可靠性 (Reliability), 86
- 可靠性度量, 86
- 可靠组播 (reliable multicast), 208
- 可行后继路由器 (Feasible successor), 211
- 可行距离 (Feasible distance), 210
- 可行性条件 (Feasibility Condition, FC), 211
- 空接口 (NULL Interface), 343
- 快速交换 (Fast Switching), 66
- 扩散更新算法 (Diffusing Update Algorithm), 210
- 扩散计算 (diffusing computations), 206
- 扩展 IP, 551
- 扩展 IPX, 551
- 扩展 IP 访问列表, 554
- 垃圾收集 (Garbage Collection), 115
- 老化 (Aging), 100
- 类型 1 的外部路径 (Type 1 external path, E1), 305
- 类型 2 的外部路径 (Type 2 external path, E2), 306
- 链路 ID (Link ID), 317
- 链路类型 (Link Type), 317
- 链路数据 (Link Data), 317
- 链路状态 (Link State), 88
- 链路状态 ID (Link State ID), 313
- 链路状态 PDU (LSP), 376
- 链路状态更新报文, 314
- 链路状态计时器 (LSRefreshTime), 101
- 链路状态路由选择协议, 95
- 链路状态请求报文, 313
- 链路状态确认报文, 315
- 链路状态数据库, 101
- 链路状态数据库, 291
- 链路状态通告 (Link State Advertisement, LSA), 259
- 邻接 (Adjacency), 210
- 邻接关系 (Adjacency), 258
- 邻居 (Neighbours), 89
- 邻居表 (neighbor table), 209
- 邻居数据结构, 272
- 邻居状态机, 273
- 零老化生存时间 (ZeroAgeLifetime), 384
- 流量过滤, 548
- 路径决策, 84
- 路径类型 (Path Type), 305
- 路由标记, 529
- 路由标志 (Route Tag), 177
- 路由环路 (Routing Loop), 87
- 路由回馈, 489
- 路由器, 12
- 路由器 ID, 259
- 路由器 LSA (Router LSA), 294
- 路由器 LSA, 316
- 路由器的优先级 (Router Priority), 264
- 路由器条目 (Router Entries), 305
- 路由图, 513
- 路由选择表, 56
- 路由选择表查询, 57
- 路由选择域 (Routing Domain), 142
- 命令 (Command), 117
- 命名访问, 552
- 命名访问列表, 560
- 末梢 (Stub) 区域, 300
- 末梢区域 (Stub Area), 287
- 末梢网络 (Stub Network), 262
- 目标子网, 58
- 目的地址, 56
- 目的端口 (Destination Port), 49
- 内部路由 (Interior Routes), 143
- 内部路由器 (Internal Routers), 288

- 内部网关协议 (IGP), 107
- 逆向路由 (Reverse Route), 91
- 偏移列表 (offset list), 129
- 平均回程时间 (smooth round-trip time, SRTT), 209
- 请求消息 (Request messages), 114
- 请求消息类型 (Request Message Type), 118
- 区域 (Area), 105
- 区域 (Area), 285
- 区域边界路由器 (Area Border Router, ABR), 106
- 区域边界路由器 (Area Border Routers, ABRs), 288
- 区域地址 CLV, 395
- 区域关联位 (Attached, ATT), 377
- 区域间路径 (Inter-area path), 305
- 区域间路由汇总 (Inter-area summarization), 342
- 区域内路径 (Intra-area path), 305
- 全 0 子网, 33
- 全 1 子网, 33
- 缺省路由, 472
- 缺省网络, 478
- 确认报文 (Acknowledgments, ACKs), 208
- 确认号 (Acknowledgment Number), 49
- 认证, 182
- 认证信息 CLV, 397
- 进站 (incoming) 路由更新, 131
- 进站过滤器, 561
- 生存时间 (Time To Live, TTL), 24
- 十进制 (decimal), 541
- 十六进制 (hexadecimal), 543
- 十六进制, 33
- 时间戳选项 (Timestamp), 25
- 时延 (Delay), 146
- 时延 (Delay), 86
- 收敛 (Convergence), 87
- 收敛时间, 87
- 首个 8bit 字节规则, 29
- 输入事件 (input event), 216
- 树数据库, 103
- 数据封装, 7
- 数据库描述报文, 312
- 数据库描述序列号 (DD Sequence Number), 313
- 数据库同步 (Database Synchronization), 271
- 数据链路 PDU (DLPDU), 375
- 刷新计时器 (flush timer), 115
- 双向通信 (two-way communication), 261
- 水平分隔 (Split Horizon), 91
- 松散源选路选项 (Loose Source Routing), 25
- 算术校验和 (arithmetic checksum), 184
- 套接字 (Socket), 49
- 填充 CLV, 397
- 跳数 (Hop Count) 度量, 86
- 通告路由器 (Advertising Router), 313
- 透明桥接 (厂商代码), 551
- 透明桥接 (协议类型), 551
- 透明网桥, 10
- 拓扑结构表 (topological table), 211
- 外部路由 (Exterior Routes), 143
- 外部路由标志 (External Route Tag), 320
- 外部路由汇总 (External route summarization), 342
- 外部通信量 (External Traffic), 287
- 外部网关协议 (EGP), 107
- 外部属性 LSA (External Attributes LSA), 300
- 完全邻接 (full adjacency), 271
- 完全末梢区域 (Totally Stubby Area), 301
- 完全序列号报文, 405
- 网关信息协议 (GWINFO), 114
- 网络 LSA (Network LSA), 295
- 网络 LSA, 318
- 网络层协议标识符, 398
- 网络层协议数据单元 (NPDU), 375
- 网络号, 28
- 网络汇总 LSA (Network Summary LSA), 295
- 网络汇总 LSA, 319
- 网络时钟协议 (Network Time Protocol, NTP), 193

- 网络实体标题 (Network Entity Title, NET), 378
- 网络条目 (Network Entries), 304
- 网桥, 11
- 伪节点, 263
- 伪节点 ID (Pseudonode ID), 381
- 位计数, 33
- 无故 ARP, 43
- 无类别路由查找, 179
- 无类别路由选择协议, 179
- 无类别域间路由选择 (CIDR), 235
- 无效计时器 (Invalidation Timer), 115
- 无效时间间隔 (RouterDeadInterval), 260
- 物理层, 20
- 系统标识 (System ID), 378
- 系统路由 (System Routes), 143
- 下一跳 (next hop), 56
- 下游路由器 (Downstream Router), 156
- 显式确认 (Explicit Acknowledgment), 278
- 线性序列号空间, 97
- 限时计时器 (Expiration Timer), 115
- 响应消息 (Response messages), 114
- 消息摘要 (message digest) 算法, 184
- 小型互连网络, 29
- 协议号, 25
- 协议数据单元, 375
- 虚链路 (Virtual Link), 288
- 虚链路 (Virtual Links), 262
- 虚拟标志 (Virtual Flag), 402
- 许可, 548
- 序列号 (Sequence Number), 49
- 选择路由, 61
- 选择项 (Options), 49
- 循环序列号空间, 98
- 严格源选路选项 (Strict Source Routing), 25
- 以太网地址, 551
- 以太网类型代码, 551
- 异步更新, 94
- 抑制计时器 (Holddown Timer), 116
- 抑制计时器 (Holddown Timer), 94
- 隐式确认 (Implicit Acknowledgment), 278
- 应用层, 20
- 硬件类型码, 41
- 拥塞, 10
- 用户数据报协议 (UDP), 49
- 有类别路由查询, 119
- 有类别路由选择 (Classful Routing), 118
- 有类别路由选择, 120
- 域间路由选择协议信息 CLV, 404
- 域间通信量 (Inter-Area Traffic), 286
- 域间信息类型 (Inter-Domain Information Type), 404
- 域内通信量 (Intra-Area Traffic), 286
- 源端口 (Source Port), 49
- 钥匙管理 (Key management), 192
- 直连网络, 57
- 指定路由器, 263
- 指定路由器, 381
- 中继器, 10
- 中间系统, 375
- 中间系统邻居 CLV, 396, 401
- 中型互连网络, 29
- 重传超时 (retransmission timeout, RTO), 209
- 重定向 (Redirection), 46
- 主/从位 (Master/Slave bit), 277, 313
- 主机到主机层, 20
- 主机号, 28
- 主网络号, 33
- 子网 (subnet), 33
- 子网的子网 (sub-subnets), 181
- 子网独立子层 (Subnetwork Independent Sublayer), 380
- 子网规划, 34
- 子网位, 32
- 子网掩码, 33
- 子网依赖子层 (Subnetwork Dependent Sublayer), 380
- 自动路由汇总, 239
- 自主系统 (Autonomous system), 106
- 自主系统边界路由器 (Autonomous System

- Boundary Routers, ASBR), 288
- 自主系统外部 LSA (Autonomous System External LSA), 298
- 自主系统外部 LSA, 319
- 组播地址, 56
- 组播流计时器 (multicast flow timer), 209
- 组步调 (group-pacing), 293
- 组成员 LSA (Group Membership LSA), 298
- 最长匹配, 306
- 最大路径条数 (maximum-paths), 156
- 最大年龄差距 (MaxAgeDiff), 100
- 最大年龄值 (MaxAge), 101
- 最短路径树, 104
- 最短路径优先算法 (SPF), 258
- 最优匹配, 57

