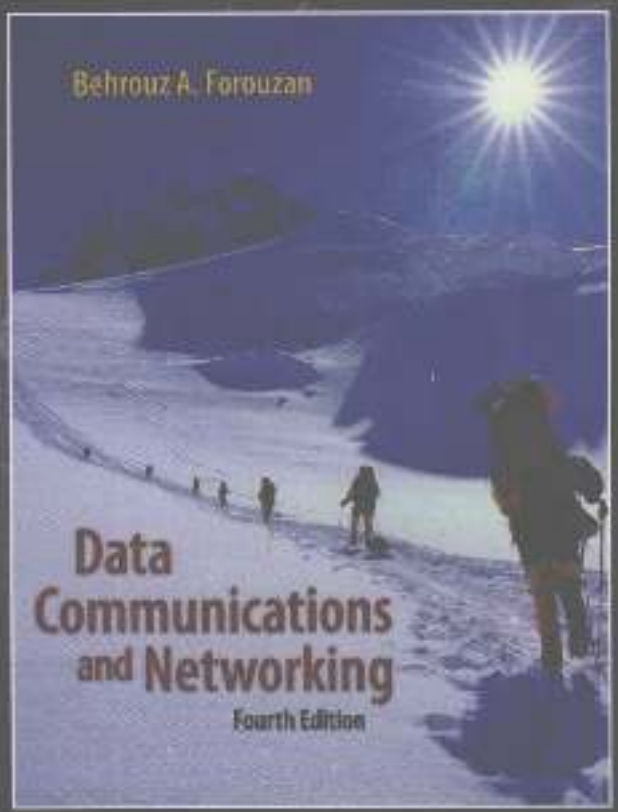
计 算 机 科 学 丛 书

原书第4版

数据通信与网络

(美) Behrouz A. Forouzan Sophia Chung Fegan 著
迪安学学院
吴时霖 吴永辉 吴之艳 魏霖 译
复旦大学



Data Communications and Networking
Fourth Edition

机械工业出版社
China Machine Press

数据通信与网络（原书第4版）

本书延续了Forouzan一贯的风格，以通俗易懂的方式全面阐述了网络技术及其应用。本书自第1版引进国内以来，对网络课程教学产生了较大影响，被多所名校采用为教材，成为清华大学和中科院博士入学考试网络课程的主要参考书。

本书以因特网五层模型为框架，以形象直观的描述手法，详细地介绍了数据通信和网络领域的基础知识、基本概念、基本原理和实践方法，堪称数据通信和网络方面的经典著作。

本书适合作为计算机专业有关数据通信和网络课程的本科生或研究生的教材或教学参考书，也可作为相关的研究人员和技术人员的参考书。

本书特点：

- 作者教学经验丰富，书中反映实际应用、工作实例和众多新近的实践问题。
- 图表众多，并且尽量少地使用数学公式，复杂的概念通过形象直观的图解加以清晰的说明。
- 较第3版增加了新的一章“网络管理：SNMP”（第28章），修订后的第30章、第31章和第32章涉及因特网和建网中最新的安全问题。

作者简介

Behrouz A. Forouzan 毕业于加州大学艾尔语分校，现在圣迭安哥学院教授，负责计算机信息系统专业的课程设置。此外，他还是多家公司的系统开发咨询顾问。除本书外，Forouzan还著有多部成功的编程与网络方面的书，包括《TCP/IP Protocol Suite》和《Local Area Networks》等。

Education

上架指导：计算机，通信与网络

ISBN 978-7-111-21354-3

9 787111 213543

ISBN 978-7-111-21354-3

定价：79.00 元

投稿热线：(010) 88379604

购书热线：(010) 68995259, 68995264

读者信箱：hzjsj@hzbook.com

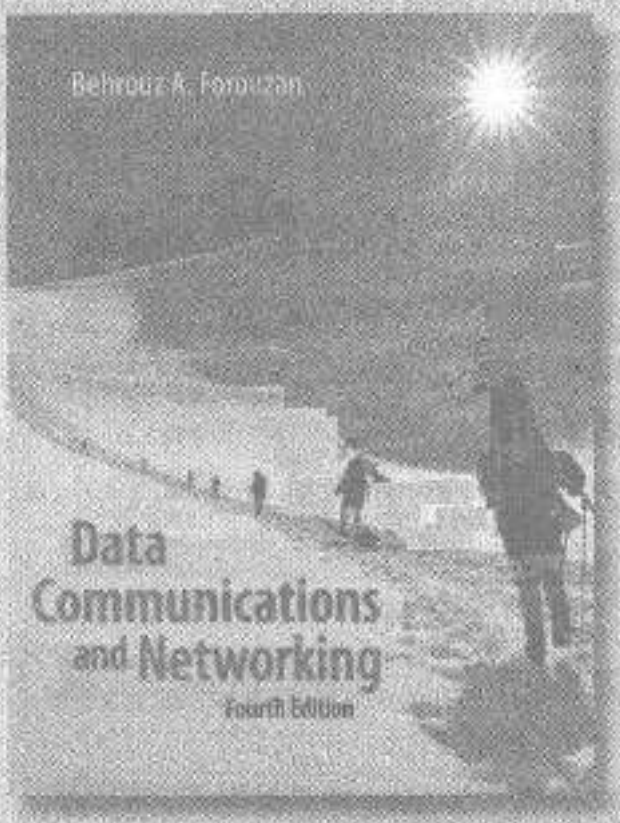
华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

计 算 机 科 学

数据通信与网络

(美) Behrouz A. Forouzan Sophia Chung Fegan 著 吴时霖 吴永军 吴之艳 魏霖 译
迪安萨学院 复旦大学



TN919
30(4)

Data Communications and Networking

Fourth Edition

机械工业出版社
China Machine Press

本书系统地介绍了数据通信与计算机网络领域的基本理论及应用。与前几版相比，第4版在内容和编排上都有较大更新，更加注重理论与实际应用相结合。主要内容包括：传输介质和数据编码、模拟信号和数字信号、建立连接、数据压缩、数据完整、数据安全、流量控制、局域网、网络互连、因特网协议和应用、选路策略、电路技术等。另外，每章后面均附有大量复习题与练习题。

本书适合作为高等院校计算机科学专业本科生的教材或参考书，也可供从事数据通信和网络工作的技术人员参考。

Behrouz A. Forouzan, Sophia Chung Fegan: Data Communications and Networking, Fourth Edition (ISBN 0-07-296775-7)

Copyright © 2007 by The McGraw-Hill Companies, Inc.

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-3132

图书在版编目（CIP）数据

数据通信与网络（原书第4版）/（美）佛罗赛（Forouzan, B. A.），（美）费根（Fegan, S. C.）著；吴时霖等译. 北京：机械工业出版社，2007.7

（计算机科学丛书）

原名原文：Data Communications and Networking, Fourth Edition

ISBN 978-7-111-21354-3

Ⅰ.数… Ⅱ.①佛…②费…③吴… Ⅲ.①数据通信…②计算机网络… Ⅳ.TN 919 TP393

中国版本图书馆CIP数据核字（2007）第056757号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李东霞

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007年7月第1版第1次印刷

184mm×260mm · 45.5印张

定价：79.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

译 者 序

当今，数据通信与网络技术飞速发展，需要学习这门课程的人不断增加。其中既包含在校学习的学生也包含从事相关工作的工程技术人员。因此，“数据通信和网络”课程的教材必须既要反映最新技术进展，又要方便没有这方面基础知识的学生学习。

Forouzan的《Data Communications and Networking》（中文译名为《数据通信与网络》）一书自1998年第1版到2006年第4版，一直按照这一原则组织教材并不断修改更新教材内容，以适应当前网络技术的发展。从第3版开始，本书不按照OSI模型七层模式而是根据因特网模型组织全书内容，并花更多的篇幅讨论TCP/IP协议族的内容。这是当今因特网发展的实际需要，而不是停留在标准理论模式讨论，因为七层模式从未直接实现过，而因特网模型是已经实现了的模型。该书第3版在2004年出版，两年不到，2006年1月又作了全面修订，重写一些章节出版了第4版，更新速度比前几版要快。在第4版中，还是采用由下而上的方式，从数据通信和网络技术的高度提出问题，图文并茂而不是使用复杂公式来讲述技术性很强的内容，作者充分利用他丰富的教学方法，采用700多幅直观图来辅助学生理解与掌握，同时尽可能结合实际的例子来阐明基本概念。该书第4版使初学者站在最新科技进展最前沿，学习网络有关问题（如多播路由选择协议、流控制传输协议、网络安全性以及其他新的网络技术）。特别要提到的是，书中讨论了下一代IP协议并扩充了第3版网络安全问题的讨论。

译者曾参加本书第1、2、4版的翻译工作，因此深切体会到作者对“数据通信与网络”这门课程的教学方法有着深入的研究和丰富的经验。全书以学生为中心，重点概念在书中多次重复，每章都有小结归纳该章所有内容，“自下而上”的方式允许教师把几门单独的数据通信和网络课程合并为一门课程。事实上，作者已将“TCP/IP协议族”作为另一本教科书出了第3版。

正如作者在前言中所说，对本书编写贡献最大的是来自同行们的审查，他们花费了大量心血和时间阅读原稿，并提出了许多有益的意见和想法，这也就是本书深受读者欢迎的原因之一。

本书前言、第一部分、第二部分、第四部分、第五部分以及附录由吴时霖、吴未辉翻译，第三部分由魏霖翻译，第六部分与第七部分由吴之竑、吴时霖翻译，最后由吴时霖审校，限于译者水平，翻译不妥之处敬请广大读者批评指正。

吴时霖于复旦大学

2006年11月

前　言

数据通信与网络是现代文明中发展最快的技术。一方面，这种发展带来了相关职业数量的迅猛增长，在这些职业中，成功的关键在于对这些技术的理解；另一方面，学习这些技术的学生数量与种类也在不断增长。

本书特点

为了使学生更容易地理解数据通信与网络技术，本书具有如下几个特点。

结构新颖

我们采用五特网的五层模型作为本书的框架。这不仅仅是因为对该模型的透彻理解对于学习最流行的网络理论是至关重要的，还因为该模型是一个互相依赖的结构：每一层都构建在其下层之上，同时对它的上层提供支持。同样，本书中引入的每一概念都构建在之前章节讨论的概念之上。另外，之所以选择因特网模型，是因为它是一个已经完全实现了的协议。

本书是为那些没有或者只有很少的电信或数据通信基础的学生编写的。因此，我们采用的是“自底向上”的讲述方式。通过这种讲述方式，学生在学习网络（高层协议）之前，可以先接触数据通信（低层协议）的相关知识。

形象直观

本书采用图文并茂的方式而没有使用复杂的公式来表述技术性很强的问题。书中约有700多幅插图与文字说明配合起来讲解，使内容直观易懂。在解释网络概念时，插图的作用尤其重要，因为这些网络概念是基于连接和传输的。这两种技术都易于通过插图理解。

突出重点

本书将重要概念用黑体表示，以便于快速查找和迅速引起读者的注意。

举例和应用

在适当的时候，我们会选用实例来反映现实的情况。例如，在第6章中，我们就给出了几个现今电话网络中电信方面的实例。

推荐读物

每章都有“推荐读物”小节，其中给出一系列图书和网站，用于读者进一步阅读和提高。

本章小结

每章末都有该章内容的小结。小结是该章全部重点内容的归纳。

习题

每章都包括习题，用来巩固和应用重要概念。它包含三部分：“复习题”、“练习题”和“探究题”（仅在某些章）。“复习题”旨在测试学生对该章内容的初步理解。“练习题”则需要学生对内容有更深入的理解。而“探究题”是为了进一步深入学习而设计的。

附录

附录的目的是提供快速的内容查阅，是对全书中需要加以理解的概念的回顾。

术语表和缩略词表

本书包含术语表和缩略词表。

第4版的变化

第4版在组织结构和内容两个方面对第3版有重大的变化。

组织结构方面

本书的组织结构方面的变化如下：

1. 第6章包含复用和分离；
2. 第8章专门讨论交换；
3. 原第3版的第12章的内容现移到第4版的第11章；
4. 第17章讨论SONET技术；
5. 第19章讨论IP寻址；
6. 第20章专门介绍虚特网协议；
7. 第21章讨论三个协议：ARP、ICMP和IGMP；
8. 第28章是新的，专门讨论因特网络管理；
9. 原第3版的第29章－第31章现在是第4版的第30章－第32章。

内容方面

我们已修订了许多章的内容，包括如下：

1. 第1章～第5章的内容已作了修订和增加。增加实例是为了使内容清晰易懂；
2. 第10章的内容作了修订和增加以包括检错和纠错的方法；
3. 第11章的内容已作了修订以包括一些控制链路协议的完整讨论；
4. 第22章中增加了数据报的传递、转发和路由选择；
5. 第23章中增加了新的传送协议SCTP；
6. 第30章～第32章的内容已作了修订和增加以包括有关虚特网和安全问题的补充讨论；
7. 为了更清楚地理解概念，增加了一些新的实例。

各章结尾资料

1. 每章末增加一节“推荐读物”，列出进一步阅读和学习的一些补充资料；
2. “复习题”已作了变动和更新；
3. “多项选择题”已移到本书的配套网站上，使学生能够自我测试对本章内容和知识的掌握情况并可以在线获得；
4. “练习题”已作了修订，在某些章中增加了新的习题；
5. 有些章后包括“探究题”小节。

辅导材料

学生和教师两者相应的辅导资料都已作了修订和增加。必要时，习题解答中会既包括解释又包括含有彩色图或表格的答案。幻灯片演示会更加全面和丰富，图文并茂。

内容安排

本书分为七部分。第一部分是概述，最后一部分是关于网络安全的内容，中间的五个部分用来阐述因特网模型的五个层次。下面是各部分内容的概要。

第一部分：概述

第一部分是数据通信与网络的总体概述，由两章组成。第1章列出了本书其余部分所需要的基本概念，第2章介绍了因特网模型。

第二部分：物理层和介质

第二部分是对因特网模型物理层的讨论。这一部分包括第3章～第9章。其中第3章－第6章讨论物理层的电信方面的问题。第7章介绍传输介质。虽然传输介质不是物理层的一部分，但物理层涉及传输介质问题。第8章专门讨论某些层中所用的交换技术。第9章给出如何使用两个公共网络：电话网和有线电视网来进行数据传输。

第三部分：数据链路层

第三部分专门讨论了因特网模型的数据链路层。这一部分包括第10章－第18章。其中第10章主要讨论的是检错和纠错。第11章和第12章讨论与数据链路控制有关的问题。第13章～第16章是关于局域网的内容。第17章和第18章是有关广域网的内容。局域网和广域网是因特网模型的前面两层的网络操作的实例。

第四部分：网络层

第四部分专门讨论了因特网模型的网络层。这一部分包括第19章～第22章的内容。其中第19章主要讲述IP地址。第20章和第21章专门讨论网络层协议，如IP、ARP、ICMP和IGMP。第22章讨论因特网中分组的传递、转发和路由选择。

第五部分：传输层

第五部分专门讨论了因特网模型的传输层。这一部分包括第23章和第24章。其中第23章是传输层概述，并讨论了这一层的服务和职责。第23章还介绍了三种因特网传输层协议，即UDP、TCP和SCTP。第24章讨论拥塞控制和服务质量。这两个问题与传输层及前两层有关。

第六部分：应用层

第六部分专门讨论了因特网模型的应用层。这一部分包括第25章～第29章。第25章是关于域名系统（DNS）的内容，该应用程序被其他应用程序用来将应用层地址映射到网络层地址。第26章－第29章讨论因特网一些公共的应用协议。

第七部分：网络安全

第七部分讨论网络安全问题。这部分包括第30章～第32章，它作为进一步学习网络安全的基础。其中第30章简短讨论密码学思想。第31章介绍安全的各个方面，第32章给出因特网模型中三个层如何应用安全的问题。

在线学习中心

McGraw-Hill在线学习中心有许多补充材料，网址是www.mhhe.com/forouzan，学生在阅读本书时，可以上网做一下自我测试，还可以浏览讲义内容（如幻灯片）并能够通过一些动

回来温习本书的内容。此外，在网站上还可以得到答案，提供给学生的是奇数号问题的答案，而教师可通过向McGraw-Hill公司申请密码来下载全部问题的答案[⊖]。

另外，为了使该课程独立网站更加方便，McGraw-Hill公司提供了一款优秀的产品，称为PageOut。它不需要预先学习HTML的知识，也不需要花太多时间和设计技巧去建立网站。它提供了一系列的模板。只要简简单单地填上课程信息，在16个设计模板中点击选择一个就可以了。整个过程用不了一个小时，就拥有了一个专业设计水准的网站。

尽管PageOut提供的只是一种"速成"式的开发，但是所设计出来的网站具有强大的功能。交互式的课程表允许教师发布讲授的课程内容，这样，当学生们访问PageOut网站时，课程讲授的要点可以指引他们访问Forouzan在线学习中心的内容或教师指定的内容。

如何使用本书

本书适用于大专院校的教学，也适用于专业人员的阅读。它可以作为对此感兴趣的专业人员的自学指导教材。作为教科书，可以安排一个学期或半个学期的课程。下面是学习本书的一些建议：

- 强烈推荐学习第一部分～第三部分。
- 如果没有讲授TCP/IP协议的后续课程，则应该学习第四部分～第六部分。
- 如果没有讲授网络安全的后续课程，则推荐学习第七部分。

致谢

很明显，写这样篇幅的书需要许多人的帮助。

同行评审

对本书编写最大贡献的是同行们对本书的评审。我们难以用语言来表达对这些评审者的感激之情，他们花费了大量心血和时间阅读原稿，并给我们提出了许多有益的意见和想法。在此特别答谢以下评审者对本书第3版和第4版作出的贡献：

Farid Ahmed, 天主教大学
Kaveh Ashenayi, 塔尔萨大学
Yoris Au, 得克萨斯大学圣安东尼奥分校
Essie Bakhtiar, 克雷顿学院与州立大学
Anthony Barnard, 阿拉巴马大学伯明翰分校
A.T.Burrell, 俄克拉荷马州立大学
Scott Campbell, 迈阿密大学
Teresa Carrigan, 布莱科本学院
Hwa Chang, 塔大茨大学
Edward Chlebus, 伊利诺伊理工学院
Peter Cooper, 州立萨姆休斯敦大学
Richard Coppins, 弗吉尼亚联邦大学
Harpal Dhillon, 西南俄克拉荷马州立大学
Hans-Peter Dommel, 圣塔克拉拉大学
M.Barry Dumas, 纽约市立大学柏鲁克分校
William Figg, 达科他州立大学
Dale Fox, 奎尼皮雅大学(Quinnipiac University)
Terrence Fries, 卡罗来纳海岸大学
Errin Fulp, 维克森林大学
Sandeep Gupta, 亚利桑那州立大学
George Hamer, 南达科他州立大学
James Henson, 加利福尼亚州立大学弗雷斯诺分校
Tom Hilton, 犹他州立大学
Allen Holliday, 加利福尼亚州立大学富勒顿分校
Seyed Hossein, 威斯康星大学密尔沃基

⊖ 需要本书教师辅导资料的教师可填写本书最后的教师服务咨询表，向McGraw-Hill公司申请账户名和密码下载相关资料。——编辑注

分校

Gerald Isaacs, 位于沃基夏的卡罗尔学院
Hrishikesh Joshi, 德福瑞大学
E.S.Khosravi, 南方大学
Bob Kinicki, 伍斯特理工大学
Kevin Kwiat, 哈密尔顿学院
Ten-Hwang Lai, 俄亥俄州立大学
Chung-Wei Lee, 奥本大学
Ka-Cheong Leung, 得克萨斯工科大学
Gertrude Levine, 菲尔莱狄更斯大学
Alvin Sek Soo Lim, 奥本大学
Charles Liu, 加利福尼亚大学洛杉矶分校
Wenhang Liu, 加利福尼亚大学洛杉矶分校
Mark Llewellyn, 中佛罗里达大学
Sanchita Mal-Sarkar, 克利夫兰州立大学
Louis Marseille, 夏福社区学院
Kevin McNeill, 亚利桑那大学
Arnold C.Meltzer, 乔治·华盛顿大学
Rayman Meservy, 杨百翰大学
Prasant Mohapatra, 加利福尼亚大学戴维斯分校
Hung Z Ngo, 纽约州立大学布法罗分校
Larry Owens, 加利福尼亚州立大学弗雷斯诺分校
Arnold Patton, 布莱德利大学
Dolly Samson, 夏威夷太平洋大学
Joseph Sherif, 加利福尼亚州立大学富勒顿分校
Robert Simon, 乔治梅森大学
Ronald J.Srodawa, 奥克兰大学
Daniel Tian, 加利福尼亚州立大学蒙特里港分校
Richard Tibbs, 雷德福大学
Christophe Veltsos, 明尼苏达州立大学孟卡多分校
Yang Wang, 马里兰大学帕克分校
Shenali Zeadally, 韦恩州立大学

McGraw-Hill公司的工作人员

我们还要感谢McGraw-Hill公司的工作人员。我们的出版人Alan Apt证明了一个出版专家可以将不可能的事情变成可能。只要需要帮助，开发编辑Rebecca Olson就会给我们帮助。我们的项目经理Sheila Frank在出版过程中一直以极大的热情指导我们。还要感谢David Hash在设计上和Kara Kudronowicz在生产过程中的贡献。也要感谢文字编辑Patti Scott。

目　录

译者序
前言

第一部分　概　述

第1章　绪论 ……2
1.1　数据通信 ……2
1.1.1　组成 ……3
1.1.2　数据表示 ……3
1.1.3　数据流 ……4
1.2　网络 ……5
1.2.1　分布式处理 ……5
1.2.2　网络类型 ……5
1.2.3　物理结构 ……5
1.2.4　网络模型 ……9
1.2.5　网络分类 ……9
1.2.6　网络互联：因特网 ……10
1.3　因特网 ……11
1.3.1　历史简介 ……11
1.3.2　因特网现状 ……12
1.4　协议和标准 ……13
1.4.1　协议 ……13
1.4.2　标准 ……13
1.4.3　标准化组织 ……14
1.4.4　因特网标准 ……14
推荐读物 ……15
本章小结 ……15
习题 ……16
第2章　网络模型 ……17
2.1　任务分层 ……17
2.1.1　发送方，接收方和载体 ……17
2.1.2　层次结构 ……18
2.2　OSI模型 ……18
2.2.1　层次化体系 ……19
2.2.2　对等过程 ……19
2.2.3　封装 ……20
2.3　OSI模型的各层功能 ……21
2.3.1　物理层 ……21
2.3.2　数据链路层 ……21
2.3.3　网络层 ……23
2.3.4　传输层 ……24
2.3.5　会话层 ……25
2.3.6　表示层 ……25
2.3.7　应用层 ……26
2.3.8　各层功能小结 ……27
2.4　TCP/IP协议族 ……27
2.4.1　物理层和数据链路层 ……28
2.4.2　网络层 ……28
2.4.3　传输层 ……29
2.4.4　应用层 ……29
2.5　寻址 ……29
2.5.1　物理地址 ……30
2.5.2　逻辑地址 ……31
2.5.3　端口地址 ……32
2.5.4　专用地址 ……33
推荐读物 ……33
本章小结 ……33
习题 ……34

第二部分　物理层和介质

第3章　数据和信号 ……37
3.1　模拟与数字 ……37
3.1.1　模拟数据与数字数据 ……37
3.1.2　模拟信号与数字信号 ……37
3.1.3　周期信号与非周期信号 ……38
3.2　周期模拟信号 ……38
3.2.1　正弦波 ……38
3.2.2　相位 ……41
3.2.3　波长 ……42
3.2.4　时域和频域 ……42
3.2.5　复合信号 ……43

3.2.6 带宽 ……45
3.3 数字信号 ……47
3.3.1 比特率 ……48
3.3.2 位长 ……48
3.3.3 数字信号是一种复合模拟信号 ……48
3.3.4 数字信号的传输 ……49
3.4 传输减损 ……54
3.4.1 衰减 ……54
3.4.2 失真 ……55
3.4.3 噪声 ……56
3.5 数据速率限制 ……57
3.5.1 无噪声通道：奈奎斯特比特率 ……57
3.5.2 噪声通道：香农容量定理 ……58
3.5.3 使用两种限制条件 ……59
3.6 性能 ……59
3.6.1 带宽 ……59
3.6.2 吞吐量 ……60
3.6.3 延迟 ……60
3.6.4 带宽与延迟的乘积 ……62
3.6.5 抖动 ……63
推荐读物 ……63
本章小结 ……63
习题 ……64
第4章 数字传输 ……67
4.1 数字到数字转换 ……67
4.1.1 线路编码 ……67
4.1.2 线路编码方案 ……70
4.1.3 块编码 ……76
4.1.4 扰动 ……78
4.2 模拟到数字转换 ……79
4.2.1 脉冲码调制（PCM） ……79
4.2.2 Delta调制（DM） ……85
4.3 传输模式 ……86
4.3.1 并行传输 ……87
4.3.2 串行传输 ……87
推荐读物 ……89
本章小结 ……89
习题 ……90
第5章 模拟传输 ……92
5.1 数字到模拟转换 ……92
5.1.1 数模转换问题 ……92
5.1.2 幅移键控 ……93
5.1.3 频移键控 ……95
5.1.4 相移键控 ……96
5.1.5 正交振幅调制 ……99
5.2 模拟信号调制 ……99
5.2.1 调幅 ……100
5.2.2 调频 ……100
5.2.3 调相 ……101
推荐读物 ……102
本章小结 ……102
习题 ……103
第6章 带宽利用 ……105
6.1 复用 ……105
6.1.1 频分复用 ……106
6.1.2 波分复用 ……109
6.1.3 同步时分复用 ……110
6.1.4 统计时分复用 ……117
6.2 扩频 ……118
6.2.1 跳频扩频（FHSS） ……118
6.2.2 直接序列扩频 ……120
推荐读物 ……120
本章小结 ……120
习题 ……121
第7章 传输介质 ……124
7.1 有向介质 ……125
7.1.1 双绞线 ……125
7.1.2 同轴电缆 ……127
7.1.3 光缆 ……128
7.2 无线通信 ……132
7.2.1 无线电波 ……133
7.2.2 微波 ……133
7.2.3 红外波 ……134
推荐读物 ……135
本章小结 ……135
习题 ……135
第8章 交换 ……138
8.1 电路交换网络 ……138
8.1.1 三个阶段 ……140
8.1.2 效率 ……140
8.1.3 延迟 ……141
8.1.4 电话网中电路交换技术 ……141

8.2 数据报网络 ……143
8.2.1 路由表 ……142
8.2.2 效率 ……142
8.2.3 延迟 ……143
8.2.4 因特网中的数据报网 ……143
8.3 虚电路网络 ……143
8.3.1 编址 ……144
8.3.2 三个阶段 ……144
8.3.3 效率 ……146
8.3.4 虚电路网络延迟 ……146
8.3.5 广域网中电路交换技术 ……147
8.4 交换机结构 ……147
8.4.1 电路交换机结构 ……147
8.4.2 分组交换机结构 ……150
推荐读物 ……152
本章小结 ……153
习题 ……153
第9章 使用电话网和有线电视进行数据传输 ……156
9.1 电话网络 ……156
9.1.1 主要组成 ……156
9.1.2 LATA ……157
9.1.3 信令 ……158
9.1.4 电话网络提供的业务 ……160
9.2 电话调制解调器 ……161
9.3 数字用户线路 ……163
9.3.1 ADSL ……163
9.3.2 ADSL Lite ……165
9.3.3 HDSL ……166
9.3.4 SDSL ……166
9.3.5 VDSL ……166
9.3.6 总结 ……166
9.4 有线电视网 ……166
9.4.1 传统有线网络 ……166
9.4.2 光纤同轴电缆混合（HFC）网络 ……167
9.5 有线电视网络用作数据传输 ……168
9.5.1 带宽 ……168
9.5.2 共享 ……169
9.5.3 CM与CMTS ……169
9.5.4 数据传输方案：DOCSIS ……170
推荐读物 ……170
本章小结 ……170
习题 ……171

第三部分 数据链路层

第10章 检错与纠错 ……174
10.1 引言 ……174
10.1.1 差错的类型 ……174
10.1.2 冗余 ……175
10.1.3 检错和纠错 ……175
10.1.4 前向纠错和重传 ……175
10.1.5 编码 ……175
10.1.6 模运算 ……176
10.2 块编码 ……176
10.2.1 差错检测 ……177
10.2.2 纠错 ……178
10.2.3 汉明距离 ……178
10.2.4 最小汉明距离 ……179
10.3 线性块编码 ……181
10.3.1 线性块编码的最小距离 ……181
10.3.2 一些线性块编码 ……181
10.4 循环编码 ……186
10.4.1 循环冗余校验 ……186
10.4.2 硬件实现 ……188
10.4.3 多项式 ……190
10.4.4 循环编码分析 ……192
10.4.5 循环编码的优点 ……195
10.4.6 其他循环编码 ……195
10.5 校验和 ……195
10.5.1 概念 ……195
10.5.2 反码 ……195
10.5.3 因特网校验和 ……196
推荐读物 ……198
本章小结 ……198
习题 ……198
第11章 数据链路控制 ……202
11.1 成帧 ……202
11.1.1 固定长度成帧 ……202
11.1.2 可变长度成帧 ……202
11.2 流量控制和差错控制 ……204
11.2.1 流量控制 ……204

11.2.2 差错控制 ……205
11.3 协议 ……205
11.4 无噪声通道 ……205
11.4.1 最简单的协议 ……205
11.4.2 停止等待协议 ……207
11.5 有噪声通道 ……209
11.5.1 停止等待自动重复请求 ……209
11.5.2 连续N帧自动重发请求 ……214
11.5.3 选择性重复ARQ ……220
11.5.4 捎带 ……225
11.6 高级数据链路控制 ……226
11.6.1 配置和传输方式 ……226
11.6.2 帧 ……226
11.6.3 控制字段 ……227
11.7 点到点协议 ……231
11.7.1 成帧 ……231
11.7.2 传输阶段 ……232
11.7.3 多路复用 ……232
11.7.4 多链路PPP ……236
推荐读物 ……237
本章小结 ……238
习题 ……238
第12章 多路访问 ……241
12.1 随机访问 ……241
12.1.1 ALOHA ……242
12.1.2 载波侦听多路访问（CSMA） ……246
12.1.3 带冲突检测的载波侦听多路访问（CSMA/CD） ……248
12.1.4 带冲突避免的载波侦听多路访问（CSMA/CA） ……251
12.2 受控访问 ……253
12.2.1 预约 ……253
12.2.2 轮询 ……253
12.2.3 令牌传递 ……254
12.3 通道化 ……255
12.3.1 频分多路访问（FDMA） ……255
12.3.2 时分多路访问（TDMA） ……256
12.3.3 码分多路访问（CDMA） ……257
推荐读物 ……261
本章小结 ……261
习题 ……262
第13章 有线局域网：以太网 ……264
13.1 IEEE标准 ……264
13.1.1 数据链路层 ……264
13.1.2 物理层 ……265
13.2 标准以太网 ……265
13.2.1 MAC子层 ……266
13.2.2 物理层 ……269
13.3 标准的变化 ……271
13.3.1 桥接以太网 ……271
13.3.2 交换式以太网 ……272
13.3.3 全双工以太网 ……273
13.4 快速以太网 ……273
13.4.1 MAC子层 ……273
13.4.2 物理层 ……274
13.5 千兆以太网 ……275
13.5.1 MAC子层 ……276
13.5.2 全双工模式 ……276
13.5.3 物理层 ……276
13.5.4 10千兆以太网 ……278
推荐读物 ……279
本章小结 ……279
习题 ……279
第14章 无线局域网 ……281
14.1 IEEE 802.11 ……281
14.1.1 体系结构 ……281
14.1.2 MAC层 ……282
14.1.3 寻址机制 ……286
14.1.4 物理层 ……288
14.2 蓝牙 ……290
14.2.1 体系结构 ……290
14.2.2 蓝牙层 ……291
14.2.3 无线电层 ……291
14.2.4 基带层 ……291
14.2.5 L2CAP ……294
14.2.6 其他上层 ……295
推荐读物 ……295
本章小结 ……295
习题 ……295
第15章 连接局域网、主干网和虚拟局域网 ……297
15.1 连接设备 ……297

15.1.1 无源集线器 ……297
15.1.2 中继器 ……297
15.1.3 有源集线器 ……298
15.1.4 网桥 ……299
15.1.5 两层交换机 ……303
15.1.6 路由器 ……304
15.1.7 三层交换机 ……304
15.1.8 网关 ……304
15.2 主干网 ……304
15.2.1 总线型主干网 ……304
15.2.2 星型主干网 ……305
15.2.3 连接远程LAN ……305
15.3 虚拟局域网 ……306
15.3.1 成员 ……307
15.3.2 配置 ……308
15.3.3 交换机间的通信 ……308
15.3.4 IEEE标准 ……309
推荐读物 ……309
本章小结 ……309
习题 ……310
第16章 无线WAN：移动电话和卫星网络 ……311
16.1 移动电话 ……311
16.1.1 频率复用原理 ……311
16.1.2 传输 ……312
16.1.3 接收 ……312
16.1.4 漫游 ……312
16.1.5 第一代 ……312
16.1.6 第二代 ……313
16.1.7 第三代 ……317
16.2 卫星网络 ……318
16.2.1 轨道 ……319
16.2.2 覆盖区域 ……319
16.2.3 三种类型的人造卫星 ……319
推荐读物 ……324
本章小结 ……324
习题 ……324
第17章 广域网SONET/SDH ……326
17.1 体系结构 ……326
17.1.1 信号 ……326
17.1.2 SONET设备 ……326
17.1.3 连接 ……327
17.2 SONET层 ……328
17.2.1 路径层 ……328
17.2.2 线路层 ……328
17.2.3 段层 ……328
17.2.4 光子层 ……328
17.2.5 设备-层之间的关系 ……328
17.3 SONET帧 ……329
17.3.1 帧、字节和位传输 ……329
17.3.2 STS-1帧格式 ……330
17.3.3 开销总结 ……332
17.3.4 封装 ……332
17.4 STS多路复用 ……333
17.4.1 字节交错 ……334
17.4.2 虚级信号 ……335
17.4.3 分插复用器 ……335
17.5 SONET网络 ……336
17.5.1 线状网络 ……336
17.5.2 环状网络 ……337
17.5.3 网状网络 ……339
17.6 虚拟支路 ……339
推荐读物 ……340
本章小结 ……340
习题 ……341
第18章 虚电路网络：帧中继和ATM ……342
18.1 帧中继 ……342
18.1.1 结构 ……343
18.1.2 帧中继层 ……344
18.1.3 扩展地址 ……345
18.1.4 FRAD ……345
18.1.5 VOFR ……345
18.1.6 LMI ……345
18.1.7 拥塞控制和服务质量 ……345
18.2 ATM ……346
18.2.1 设计目标 ……346
18.2.2 问题 ……346
18.2.3 结构 ……348
18.2.4 交换 ……349
18.2.5 ATM层 ……350
18.2.6 拥塞控制和服务质量 ……354
18.3 ATM局域网 ……354

18.3.1 ATM局域网体系结构 ……354
18.3.2 局域网仿真（LANE）……356
18.3.3 客户/服务器模型 ……356
18.3.4 具有客户/服务器的混合体系结构 ……357
推荐读物 ……357
本章小结 ……358
习题 ……358

第四部分 网 络 层

第19章 逻辑寻址 ……362
19.1 IPv4地址 ……362
19.1.1 地址空间 ……362
19.1.2 标记法 ……363
19.1.3 分类寻址 ……364
19.1.4 无类寻址 ……366
19.1.5 网络地址转换（NAT）……372
19.2 IPv6地址 ……374
19.2.1 结构 ……374
19.2.2 地址空间 ……375
推荐读物 ……378
本章小结 ……378
习题 ……379
第20章 IP协议 ……382
20.1 网际互联 ……382
20.1.1 网络层需求 ……382
20.1.2 作为数据报网络的因特网 ……384
20.1.3 作为无连接网络的因特网 ……384
20.2 IPv4 ……384
20.2.1 数据报 ……385
20.2.2 分段 ……389
20.2.3 校验和 ……392
20.2.4 选项 ……392
20.3 IPv6 ……393
20.3.1 优点 ……394
20.3.2 分组格式 ……394
20.3.3 扩展头部 ……397
20.4 IPv4到IPv6的过渡 ……398
20.4.1 双协议栈 ……399
20.4.2 隧道技术 ……399
20.4.3 头部转换 ……399
推荐读物 ……400
本章小结 ……400
习题 ……401
第21章 地址映射、差错报告和多播 ……403
21.1 地址映射 ……403
21.1.1 逻辑地址到物理地址的映射：ARP ……404
21.1.2 物理地址映射到逻辑地址：RARP、BOOTP和DHCP ……408
21.2 ICMP ……410
21.2.1 报文类型 ……410
21.2.2 报文格式 ……410
21.2.3 差错报告 ……411
21.2.4 查询 ……413
21.2.5 调试工具 ……414
21.3 IGMP ……416
21.3.1 组管理 ……417
21.3.2 IGMP报文 ……417
21.3.3 报文格式 ……417
21.3.4 IGMP操作 ……418
21.3.5 封装 ……420
21.3.6 Netstat应用程序 ……421
21.4 ICMPv6 ……422
21.4.1 差错报告 ……422
21.4.2 查询 ……423
推荐读物 ……423
本章小结 ……424
习题 ……425
第22章 传递、转发和路由选择 ……427
22.1 传递 ……427
22.2 转发 ……428
22.2.1 转发技术 ……428
22.2.2 转发过程 ……429
22.2.3 路由表 ……433
22.3 单播路由选择协议 ……435
22.3.1 优化原则 ……435
22.3.2 域内部和域间路由选择 ……435
22.3.3 距离向量路由选择 ……436
22.3.4 链路状态路由选择 ……440
22.3.5 路径向量路由选择 ……445
22.4 多播路由选择协议 ……448

22.4.1 单播、多播与广播 ……448
22.4.2 应用 ……450
22.4.3 多播路由选择 ……450
22.4.4 多播路由选择协议 ……452
推荐读物 ……459
本章小结 ……459
习题 ……460

第五部分 传 输 层

第23章 UDP、TCP和SCTP ……464
23.1 进程到进程的传递 ……464
23.1.1 客户服务器模式 ……464
23.1.2 复用与分离 ……466
23.1.3 无连接服务与面向连接的服务 ……466
23.1.4 可靠服务与不可靠服务 ……467
23.1.5 三种协议 ……467
23.2 用户数据报协议（UDP） ……468
23.2.1 熟知端口号 ……468
23.2.2 用户数据报 ……469
23.2.3 校验和 ……469
23.2.4 UDP的操作 ……471
23.2.5 UDP的使用 ……472
23.3 TCP ……472
23.3.1 TCP服务 ……472
23.3.2 TCP特点 ……475
23.3.3 段 ……476
23.3.4 TCP连接 ……478
23.3.5 流量控制 ……482
23.3.6 差错控制 ……483
23.3.7 拥塞控制 ……486
23.4 SCTP ……486
23.4.1 SCTP服务 ……486
23.4.2 SCTP特性 ……487
23.4.3 分组格式 ……490
23.4.4 SCTP关联 ……491
23.4.5 流量控制 ……494
23.4.6 差错控制 ……496
23.4.7 拥塞控制 ……498
推荐读物 ……498
本章小结 ……498
习题 ……499
第24章 拥塞控制和服务质量 ……502
24.1 数据通信量 ……502
24.1.1 通信量描述符 ……502
24.1.2 通信量特征值 ……503
24.2 拥塞 ……503
24.3 拥塞控制 ……505
24.3.1 开环拥塞控制 ……505
24.3.2 闭环拥塞控制 ……506
24.4 两个示例 ……507
24.4.1 TCP中的拥塞控制 ……507
24.4.2 帧中继中的拥塞控制 ……510
24.5 服务质量 ……511
24.5.1 数据流特性 ……512
24.5.2 数据流类型 ……512
24.6 改进QoS的技术 ……512
24.6.1 调度 ……512
24.6.2 通信量整形 ……513
24.6.3 资源预留 ……515
24.6.4 许可控制 ……515
24.7 综合业务 ……515
24.7.1 信令 ……516
24.7.2 数据流规范 ……516
24.7.3 许可 ……516
24.7.4 业务类型 ……516
24.7.5 RSVP ……516
24.7.6 综合业务所存在的问题 ……518
24.8 差分业务 ……518
24.9 交换网络中的QoS ……520
24.9.1 帧中继中的QoS ……520
24.9.2 ATM中的QoS ……521
推荐读物 ……522
本章小结 ……522
习题 ……523

第六部分 应 用 层

第25章 域名系统 ……526
25.1 名字空间 ……526
25.1.1 平面名字空间 ……527
25.1.2 层次名字空间 ……527
25.2 域名空间 ……527
25.2.1 标号 ……527

25.2.2 域名 ……528
25.2.3 域 ……528
25.3 名字空间的分布 ……529
25.3.1 名字服务器的层次结构 ……529
25.3.2 区域 ……529
25.3.3 根服务器 ……530
25.3.4 主服务器和辅助服务器 ……530
25.4 因特网中的DNS ……530
25.4.1 通用域 ……530
25.4.2 国家域 ……531
25.4.3 反向域 ……531
25.5 解析 ……532
25.5.1 解析程序 ……532
25.5.2 名字到地址的映射 ……532
25.5.3 地址到名字的映射 ……533
25.5.4 递归解析 ……533
25.5.5 迭代解析 ……533
25.5.6 高速缓存 ……533
25.6 DNS报文 ……534
25.7 记录的类型 ……535
25.7.1 询问记录 ……535
25.7.2 资源记录 ……535
25.8 注册机构 ……535
25.9 动态域名系统（DDNS） ……535
25.10 封装 ……536
推荐读物 ……536
本章小结 ……536
习题 ……537
第26章 远程登录、电子邮件与文件传输 ……539
26.1 远程登录 ……539
26.2 电子邮件 ……544
26.2.1 构象 ……544
26.2.2 用户代理 ……546
26.2.3 报文传输代理：SMTP ……550
26.2.4 报文访问代理：POP和IMAP ……552
26.2.5 基于Web的邮件 ……554
26.3 文件传输 ……554
26.3.1 文件传输协议(FTP) ……554
26.3.2 匿名FTP ……557
推荐读物 ……557
本章小结 ……558
习题 ……559
第27章 万维网与超文本传输协议 ……561
27.1 体系架构 ……561
27.1.1 客户（浏览器） ……561
27.1.2 服务器 ……561
27.1.3 统一资源定位符 ……562
27.1.4 Cookies ……562
27.2 Web文档 ……563
27.2.1 静态文档 ……563
27.2.2 动态文档 ……565
27.2.3 活动文档 ……566
27.3 超文本传输协议（HTTP） ……567
27.3.1 HTTP事务 ……567
27.3.2 持续与非持续连接 ……572
27.3.3 代理服务器 ……572
推荐读物 ……572
本章小结 ……573
习题 ……574
第28章 网络管理 ……576
28.1 网络管理系统 ……576
28.1.1 配置管理 ……576
28.1.2 故障管理 ……577
28.1.3 性能管理 ……578
28.1.4 安全管理 ……578
28.1.5 计费管理 ……578
28.2 简单网络管理协议（SNMP） ……578
28.2.1 概念 ……578
28.2.2 管理组件 ……579
28.2.3 管理信息结构 ……581
28.2.4 管理信息数据库（MIB） ……585
28.2.5 字典排序 ……587
28.2.6 SNMP ……588
28.2.7 报文 ……589
28.2.8 UDP端口 ……591
28.2.9 安全性 ……592
推荐读物 ……592
本章小结 ……593
习题 ……593
第29章 多媒体 ……595
29.1 音频与视频的数字化 ……595

29.1.1 音频数字化 ……595
29.1.2 视频数字化 ……596
29.2 音频和视频压缩 ……596
29.2.1 音频压缩 ……598
29.2.2 视频压缩 ……597
29.3 流式存储音频/视频 ……599
29.3.1 第一种方法：使用Web服务器…599
29.3.2 第二种方法：使用带有元文件的Web服务器 ……600
29.3.3 第三种方法：使用媒体服务器…600
29.3.4 第四种方法：使用媒体服务器和RTSP ……601
29.4 流式实时音频/视频 ……602
29.5 实时交互式音频/视频 ……602
29.6 实时传输协议 ……605
29.6.1 RTP分组格式 ……605
29.6.2 UDP端口 ……606
29.7 实时传输控制协议 ……607
29.7.1 发送方报告 ……607
29.7.2 接收方报告 ……607
29.7.3 源描述报文 ……607
29.7.4 关闭报文 ……607
29.7.5 特定应用报文 ……607
29.7.6 UDP端口 ……607
29.8 IP语音 ……608
29.8.1 SIP ……608
29.8.2 H.323 ……610
推荐读物 ……611
本章小结 ……611
习题 ……612

第七部分 网络安全

第30章 密码学 ……615
30.1 引言 ……615
30.1.1 定义 ……615
30.1.2 两类 ……616
30.2 对称密钥密码学 ……617
30.2.1 传统加密算法 ……617
30.2.2 简单的现代加密算法 ……619
30.2.3 现代迭代加密 ……621
30.2.4 运算模式 ……624
30.3 非对称密钥密码学 ……627
30.3.1 RSA ……627
30.3.2 Diffie-Hellman加密系统 ……630
推荐读物 ……632
本章小结 ……633
习题 ……634
第31章 网络安全服务 ……636
31.1 网络安全服务种类 ……636
31.1.1 报文保密性 ……636
31.1.2 报文完整性 ……636
31.1.3 报文鉴别性 ……636
31.1.4 不可否认性 ……636
31.1.5 实体鉴别 ……636
31.2 报文保密性 ……636
31.2.1 对称密钥密码学的保密性 ……637
31.2.2 非对称密钥密码学的保密性 …637
31.3 报文完整性 ……638
31.3.1 文档与指纹 ……638
31.3.2 报文与报文摘要 ……638
31.3.3 差异 ……638
31.3.4 创建和检验报文摘要 ……639
31.3.5 散列函数准则 ……639
31.3.6 散列算法：SHA-1 ……640
31.4 报文鉴别 ……641
31.5 数字签名 ……642
31.5.1 比较 ……642
31.5.2 需要密钥 ……643
31.5.3 过程 ……643
31.5.4 服务 ……644
31.5.5 签名方案 ……645
31.6 实体鉴别 ……645
31.6.1 口令 ……646
31.6.2 询问-应答 ……647
31.7 密钥管理 ……649
31.7.1 对称密钥的分发 ……649
31.7.2 公钥分发 ……652
推荐读物 ……655
本章小结 ……655
习题 ……656
第32章 因特网中的安全类协 ……658
32.1 IP层的安全：IPSec ……658

32.1.1 两种方式 ……659
32.1.2 两种安全协议 ……661
32.1.3 安全关联 ……662
32.1.4 因特网密钥交换（IKE） ……664
32.1.5 虚拟专用网 ……664
32.2 SSL/TLS ……667
32.2.1 SSL服务 ……667
32.2.2 安全参数 ……668
32.2.3 会话与连接 ……669
32.2.4 四个协议 ……670
32.2.5 传输层安全 ……671
32.3 PGP ……671
32.3.1 安全参数 ……672
32.3.2 服务 ……672
32.3.3 一种设想者号 ……672
32.3.4 PGP算法 ……673
32.3.5 密钥环 ……674
32.3.6 PGP证书 ……675
32.4 防火墙 ……676
32.4.1 分组过滤防火墙 ……677
32.4.2 代理防火墙 ……677
推荐读物 ……678
本章小结 ……678
习题 ……679
参考文献 ……681
缩略词 ……684
术语表 ……688
附录[1]

[1] 附录部分放到www.hzbook.com网站上，供读者自己下载。——编者注

第一部分 概 述

目标

第一部分综述本书总体思想，讨论四个主要概念：数据通信、网络、协议和标准、网络模型。

数据通过网络可以从一个地方传送到另外一个地方，这是数据通信的基本概念。要完全掌握这一领域的技术，就必须理解数据通信的构成，了解不同类型的数据是如何表示的，以及如何创建数据流。

远程实体之间的数据通信可以通过连网的过程完成，该过程包括计算机、介质和网络设备之间的连接。网络分为两大类：局域网（LAN）和广域网（WAN）。这两类网络具有不同的特性和不同的功能。本书重点讨论的因特网是由网络互连设备连接起来的LAN和WAN的集合。

协议和标准对数据通信与连网的实施至关重要。协议指的是规则，标准就是被销售商和制造商采纳的协议。

网络模型用来组织、协调和控制参与数据通信与连网的软硬件组件。虽然“网络模型”这个词表面上指的是网络结构，但实际上网络模型也包括数据通信的内容。

第1章 绪　论

数据通信与网络正在改变人们商务活动和生活方式。商务决策必须以非常快的速度做出，而决策者需要即时获得确切的信息。如果通过网络可以立刻得到来自德国的报告时，为什么还要通过邮递方式等上一个星期呢？今天的商务依赖于计算机网络和网络互联。但在同及如何更快地连接到网络之前，我们需要知道网络是如何运转的，网络使用哪些类型的技术，以及何种设计最能够满足什么样的需要。

个人计算机的发展带来了商业、工业、科学与教育的巨大变革。数据通信与网络领域也正发生着一场类似的革命。技术的进步使得通信链路能够承载更多更快的信号。因此，服务也向前发展，允许使用扩展的功能，包括对已有电话业务的扩展项目，比如电话会议、呼叫等待、语音邮件以及呼叫者ID。

现已用新的技术研究数据通信与网络。其目标是能够在世界的任何地点交换诸如文本、音频和图像之类的数据。希望在任何时间都能登录因特网快速而准确地下载和上传信息。

本章讨论当今问题：数据通信、网络、因特网以及协议和标准。首先给出数据通信广义的定义，然后将网络定义为数据传输的高速公路。互联网络（即网络的网络）的一个好的实例是因特网。最后，讨论不同类型的协议、协议和标准之间的区别以及制定这些标准的组织机构。

1.1 数据通信

当我们通信时，就共享了信息。这种共享可以是本地的，也可以是远程的，就个体之间而言，本地通信通常是面对面发生的，而远程通信发生在一定的距离之间。**电信**（telecommunication）这个词，包括电话、电报和电视，意味着在一定距离上的通信（tele是希腊语，即“远程的”）。**数据**（data）一词指的是以任何格式表示的信息，该格式需经由创建和使用数据的双方均已达成共识。

数据通信（data communication）是在两台设备之间通过诸如线缆的某种形式的传输介质进行的数据交换。因为数据通信的发生，通信设备必然成为由软（程序）硬（物理设备）件结合组成的通信系统的一部分。数据通信系统的效率取决于四个关键因素：传递性、准确性、及时性和抖动性。

1. **传递性**。系统必须将数据传递到正确的目的地。数据必须向而且只能由预定的设备或用户接收。

2. **准确性**。系统必须准确地传递数据。在传递过程中发生改变和不正确的数据都是不可用的。

3. **及时性**。系统必须以及时的方式传递数据。传递延误的数据是无用的。就视频和音频数据而言，及时传递意味着在数据产生时就传递数据。所传递数据的顺序和产生时的顺序相同，而且没有明显的延迟。这种传递称为实时传输。

4. **抖动性**。抖动是指分组到达时间的变化。音频或视频的分组在传递过程中延迟各不相同。比如，假定每30ms发送一个视频的分组，其中某些分组到达延时30ms，而另一些分组延时40ms引起视频不均匀后果。

1.1.1 组成

数据通信系统需要有五个组成部分（见图1.1）。

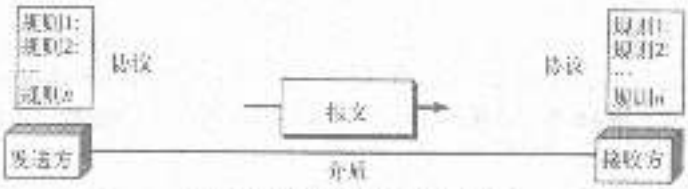


图1.1 数据通信的五个组成部分

1. **报文**（message）是进行通信的信息（数据），它可以是文本、数字、图片、声音、视频等信息形式。

2. **发送方**（sender）是指发送数据报文的设备，它可以是计算机、工作站、手机、摄像机等。

3. **接收方**（receiver）是指接收报文的设备，它可以是计算机、工作站、手机、电视等。

4. **传输介质**（transmission medium）是报文从发送方到接收方之间所经过的物理通路，它可以是双绞线、同轴电缆、光纤和无线电波。

5. **协议**（protocol）是管理数据通信的一组规则，它表示通信设备之间的一组约定。如果没有协议，即使两台设备之间可能是连接的，那也无法通信，就像一个说法语的人无法被一个只说日语的人理解一样。

1.1.2 数据表示

现在的信息有文本、数字、图像、音频和视频等多种形式。

文本

在数据通信中，文本表示为位模式，即位(0或1)序列。模式中位的数目取决于该文本中的符号数目。每一种位模式称为一组**编码**（code），表示符号的过程称为编码过程。当前流行的编码系统是**统一码**（Unicode），用32位表示一个符号或字符，它可表示世界上任何一种语言。几十年前，美国制定的**美国信息交换标准代码**（American Standard Code for Information Interchange，ASCII）现在是统一码前面的127个字符，也称为**基本拉丁字符集**(Basic Latin)。在附录A中可以找到统一码。

数字

数字也用位模式表示，但诸如ASCII的编码不是用来表示数字的。数字直接转换为二进制数。原因在于简化数字的数学运算。附录B列出几种不同的数制系统。

图像

现在，图像（image）也采用位模式表示。就最简单的形式而言，图像被划分成像素（图片的基本元素）矩阵，每个像素是一个小点。像素的大小取决于称为分辨率的因素。例如，一幅图像可以划分成1000像素或10 000像素，后一种有更清晰的图像表示（更高分辨率），但需要更多的内存来存储图像。

将图像划分成像素，对每个像素分配一个位模式。模式的大小和值取决于该图像，就一幅仅由黑白点（例如棋盘）构成的图像而言，1位模式就足以表示一个像素。

如果像素不是由纯白或纯黑的像素构成，则可通过增加位模式的大小来包含更多的灰度范围。例如，可以采用2位模式来表示4个级别的灰度。黑像素可以用00表示，深灰像素用01，浅灰像素用10，白像素用11来表示制。

有几种方法表示彩色图像：一种方法是每个颜色像素被分解为三原色：红、绿、蓝

(RGB)，然后测量每种色彩的浓度，并为其赋予一种位模式；另一种方法是每个颜色像素被分解为二原色：黄、紫、青（YCM）。

音频

音频（audio）是指录音带、声音广播或音乐等声音的表示。音频与文本、数字或图像有本质的不同，它是连续的、非离散的，即使用麦克风将声音或音乐转换成电信号，所产生的也是一个连续的信号。第4章和第5章将介绍如何将声音或音乐转换成数字或模拟信号。

视频

视频（video）是指录像带、图像或动画的合成既可以由一个连续实体（例如电视摄像机）产生，也可以由一组图像或动画合成。这些图像的每一幅都是一个离散的实体，组织起来即可表示运动的意思。正如第4章和第5章所述，同样可以将视频转换为数字或模拟信号。

1.1.3 数据流

两台设备之间的通信可以是单工、半双工或全双工的，如图1.2所示。

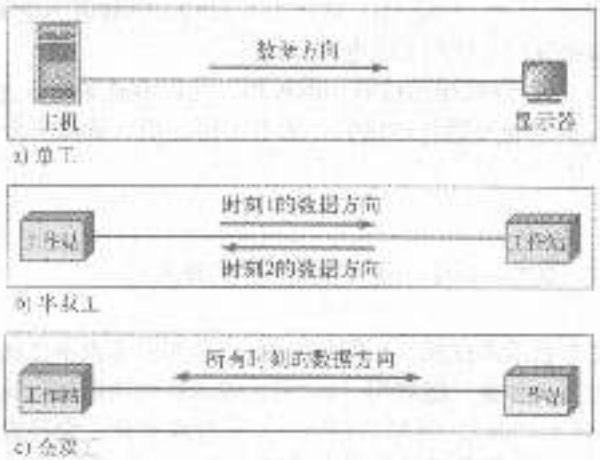


图1.2 数据流（单工、半双工和全双工）

单工

在单工模式（simplex mode）下，通信是单方向的，如同在单行道上，两台设备只有一台能够发送，另一台则只能接收（见图1.2a）。

键盘和传统的显示器都是单工通信设备，键盘只能用来输入，显示器只能接受输出。单工模式仅具有一个方向发送能力。

半双工

在半双工模式（half-duplex mode）下，每台主机均能发送和接收，但不能同时进行。当一台设备发送时，另一台只能接收，反之亦然（见图1.2b）。

半双工如同一条双向单车道的道路。当一辆轿汽车正朝一个方向行驶时，想朝另一个方向行驶的汽车就必须等待。在半双工模式下，通道的通信能力被两台设备中的发送方完全占用。对讲机和CB(民用波段）无线电设备都是半双工系统的例子。半双工模式用于不需要双方同时通信的情况，每一方可利用整个通道的能力。

全双工

在全双工模式（full-duplex mode）也称双工（duplex）下，双方主机都能同时发送和接收（见图1.2c）。

全双工模式如同一条双向大道，同时有驶向两个方向的车辆。在全双工模式下，两个方向的信号共享该链路的通信能力，共享可以采用两种方式：链路包含两条物理上独立的传输通路，其中一条用于发送，另一条用于接收；或者通道的通信能力在以不同方向传输的信号间进行划分。

常见的全双工的例子是电话网络。当两个人通过电话进行通信时，他们都能同时说和听。当要求双方同时通信时，采用全双工。然而通道的通信能力必须在两个方向之间进行划分。

1.2 网络

网络（network）是用通信链路连接起来的设备（通常称为节点）的集合。节点可能是一台计算机，也可能是打印机，还可能是其他任何能够发送或接收数据的设备。

1.2.1 分布式处理

大多数网络采用**分布式处理**（distributed processing），将任务划分给多台计算机。不是让一台大型机负责所有的处理，而是由多台独立的计算机（通常是个人计算机或工作站）各自处理一小部分。

1.2.2 网络准则

网络必须能够满足一定的准则，其中最为重要的是性能、可靠性和安全性。

性能

性能（performance）可以用多种方式来衡量，包括传输时间和响应时间。传输时间是指报文从一台设备传输到另一台设备所需的时间。响应时间是指请求和响应之间所花费的时间。网络性能取决于多种因素，包括用户的数量、传输介质类型、所连接的硬件的能力以及软件的效率。

性能通常由网络两个因素：吞吐量（throughput）和延迟（delay）来度量。需要更多吞吐量和较小的延迟时间，但是这两个准则时常是矛盾的。如果想要在网络上发送更多的数据用来增加吞吐量，但由于网络通信量的拥塞，增加了延迟时间。

可靠性

除了传输的准确性之外，网络的可靠性（reliability）用故障出现的频率、一条链路从故障恢复所花费的时间，在灾难中的健壮性来衡量。

安全性

网络的安全性（security）包括保护数据免受未授权的访问，保护数据在传输过程中免受攻击，和数据丢失与损伤的恢复策略和方法。

1.2.3 物理结构

在讨论网络之前，需要定义一些网络属性。

连接类型

网络是通过链路连接在一起的两台或两台以上的设备。链路是将数据由一台设备传输到另一台设备的通信通路。为了更直观地说明问题，最简单的情形，可以将链路想象为在两点之间连接的线。要想建立通信，两台设备必须以某种方式在同一时间连接到同一链路上。有两种可能的连接方式：点到点方式和多点方式。

点到点（point-to-point）连接提供两台设备之间专用的链路。链路全部的能力均为两台设备之间的传输所共用。大多数点到点连接采用实际长度的线或电缆来连接两端；但诸如微波或卫星链路的其他选择也是可行的（见图1.3a）。当使用红外线遥控器切换电视频道时，就在

遥控器和电视控制系统之间建立了一条点到点连接。

多点连接（multipoint connection）也称为**多站连接**（multidrop connection）是两台以上设备共享单一链路的情形（见图1.3b）。

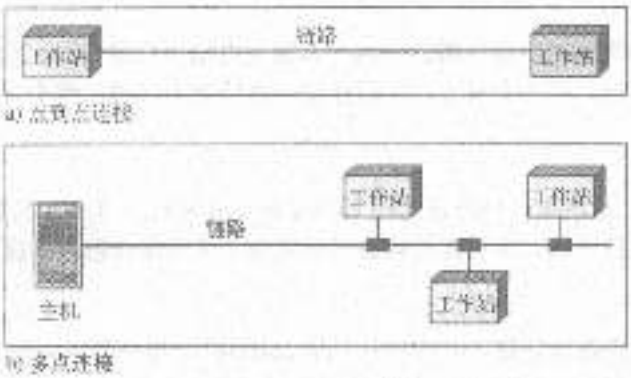


图1.3　连接类型：点到点连接与多点连接

在多点环境中，通道的能力在空间或时间上共享。如果同时可以有多台设备使用链路，这是空间上共享连接；如果用户必须轮流使用，则是时间上共享连接。

物理拓扑结构

物理拓扑结构（physical topology）一词指的是网络物理上分布的方式。两台或更多的设备连接到一条链路；两条或更多的链路组成拓扑结构。网络拓扑结构是所有链路及其互相连接的设备（通常称为节点）之间关系的几何表示。有四种可能的基本拓扑结构：网状、星型、总线和环状（见图1.4）。

网状　在网状拓扑结构（mesh topology）中，各台设备之间都有一条专用的点到点链路。“专用的”这个词意味着该链路只负责它所连接的两台设备间的通信量。为了求n个节点的一个全连接的网状网络中物理通道的个数，节点1必须与其他$n-1$个节点相连，节点2必须与其他$n-1$个节点相连，依此类推，最后节点n也必须与其他$n-1$个节点相连。因此，需要$n(n-1)$条物理链路。然而，如果每条链路允许双向通信（双工模式），则链路个数除以2。换言之，可以说在网状拓扑结构中需要

$$n(n-1)/2\text{条双工模式的链路}$$

为了提供这么多数量的链路，网络中的每台设备必须有$n-1$个输入/输出（I/O）端口（见图1.5）连接其他$n-1$个站点。



图1.4　拓扑结构分类

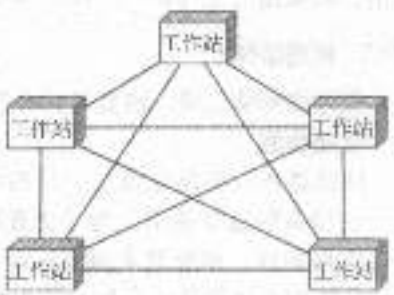


图1.5　全连接网状拓扑结构（5台设备）

与其他网络拓扑结构相比，网状结构具有诸多优点。首先，专用链路的使用保证了每条链路

都能传输自己的数据载荷，因此消除了当链路被多台设备共享时可能发生的通信量问题。其次，网状拓扑结构是健壮的。即使一条链路变得不可用，也不会使得整个系统不可用。另一个优点是机密性或安全性。每一个报文经过专用线路传输时，只有特定的接收方才能看到它。物理边界阻止其他用户获得对报文的访问权。最后，使用点到点链路，容易进行故障识别和故障隔离。对通信量进行路由时会尽量避免有问题的链路。这些设施使得网络管理员能够找到故障的确切位置，有助于寻找其原因和解决办法。

网状拓扑结构的主要不利因素与所需的线缆和I/O端口的数目有关。首先，因为每台设备都必须与其他所有设备相连，安装和重新连接十分困难。其次，单就线缆的数量而言，就可能超过可利用空间（在墙内、天花板或地板里）的容纳能力。最后，为连接每个链路所需的硬件（I/O端口和电缆）的价格也是可观的。由于以上原因，网状拓扑结构通常在有限的环境下使用，例如，作为混合网络（这些混合网络可能包括多种拓扑结构）主要计算机的主干连接。一个网状拓扑结构实例是电话区域交换局的连接，每个区域交换局必须与其他每个交换局连接。

星型 在星型拓扑结构（star topology）中，每台设备拥有一条仅与中央控制器连接的点到点专用链路。该中央控制器通常称为集线器（hub）。这些设备不是直接相互连接的。与网状拓扑结构不同，星型拓扑结构不允许设备之间有直接的通信量。集线器扮演交换的角色：如果一台设备想发送数据到另外一台，它需要将数据发到集线器，然后由集线器将数据转送给所连接的另一台设备（见图1.6）。

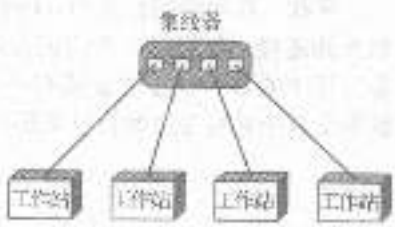


图1.6 连接4个工作站的星型拓扑结构

星型拓扑结构不如网状拓扑结构那样昂贵。在星型拓扑结构中，每台设备仅需要一条链路和一个I/O端口即可与任何数目的其他设备相连。这种因素也使得该拓扑结构易于安装和重新配置。需要安装的线缆更少，增加、移动和删除链路只涉及一条连接，即该设备和集线器之间的那条连接。

另一个优点是健壮性。如果一条链路失败，就只有该链路受影响，其他所有的链路仍然起作用。这种因素也易于进行故障识别和故障隔离。只要集线器正在工作，它就可以用来监控链路中出现的问题并将路出现问题的链路。

星型拓扑结构最大的缺点是依赖于集线器，如果集线器出了毛病，则整个系统就瘫痪。

然而，尽管星型链路需要比网状拓扑结构少得多的线缆，但每条链路仍需要连接到中央的集线器。因此，星型结构通常需要比其他拓扑结构（比如环状结构和总线结构）更多的线缆。

星型拓扑结构用于局域网（LAN）中，在第13章将会看到高速局域网常用星型拓扑结构。

总线 前面的几个例子描述的都是点到点连接，相反，总线拓扑结构（bus topology）是多点连接，由一条较长的线缆作为主干（backbone）来连接网络上所有的设备（见图1.7）。



图1.7 连接3个工作站的总线拓扑结构

节点由引出线和分接头连接到总线电缆，引出线是在设备和主线缆之间运行的连接。分接头是一个连接器，该连接器将入主线缆或者穿过线缆的外皮与金属芯接触。当信号经过主

干传输时，部分能量转化为热能。因此，当信号传输得越来越远时会变得越来越微弱。因为这种原因，总线所能支持的分接头数目以及这些分接头之间的距离是有限的。

总线拓扑结构的优势包括安装简易。主干电缆以最高效的路径来铺设，然后通过不同长度的引出线连接到节点。这样，总线结构使用的线缆要少于网状和星型结构。例如，星型结构中，同一个房间中的四台设备需要四条与集线器连接的电缆，而采用总线结构则不需要这么多电缆。只有主干电缆铺设于整个房间之中，每条引出线只需到达主干线上最近的节点。

不利因素包括难以重新连接和进行错误隔离。总线结构通常设计成在安装时达到最优，因此增加新设备是比较困难的。在分接头产生的信号反射会造成质量下降，可以通过限制连接到给定电缆的设备数目和距离来控制这种质量下降的问题。增加新设备可能因此需要改造或替换主干线。

另外，总线电缆上的错误或中断会使所有的传输中止。即使位于故障区域同一侧的设备之间也不能幸免。故障区域会将源方向的信号反射回去，从而在两个方向上产生噪音。

总线拓扑结构是早期局域网所用的结构之一，以太局域网就用总线拓扑结构，但现在不常用，我们将在第13章讨论其原因。

环状　在环状拓扑结构（ring topology）中，每台设备只与其两侧的设备有一条专用的点到点的连接。信号以一个方向在环中传输，从一台设备到达另一台设备，直到其到达目的设备。环中的每台设备中安装有一个中继器。当设备接收到发给另一台设备的信号时，它的中继器会再生并转发这些位（见图1.8）。

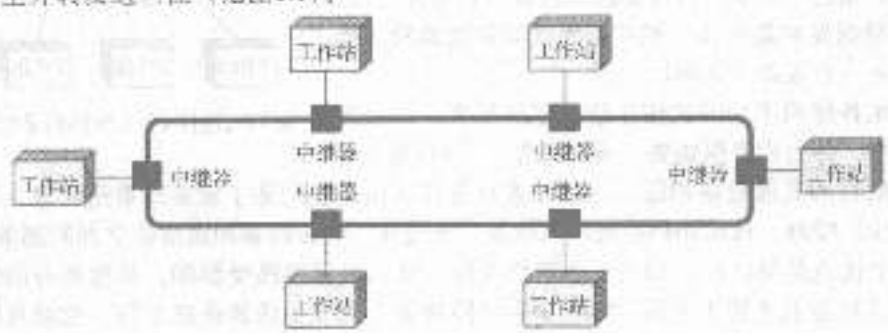


图1.8　连接6个工作站的环状拓扑结构

相对而言，环状结构易于安装和重新配置。每台设备只与其直接邻居（物理或者逻辑的）连接。要增加或删除设备只需要改变两条连接。唯一的约束在于介质和通信量方面的考虑（环最大长度和设备的最大数目）。另外，这种结构还简化了故障隔离。一般地，在环状结构中，信号始终是循环的。如果一台设备在指定的时间中没有接收到信号，它就会发出警报。该警报会向网络操作员告知问题及其位置。

然而，单向通信量可能是不利因素。在简单的环中，环中的故障（比如工作站的瘫痪）可能使整个网络瘫痪。这个缺陷可以通过引入双环或使用能够旁路故障的开关得到弥补。

IBM引入令牌环局域网时，环状拓扑结构曾盛行一时。目前要求高速局域网，这种拓扑结构就不流行了。

混合型　网络可以是混合型拓扑结构，例如，具有分支的星型拓扑结构，其中每一个分支是连接一些工作站的总线结构，如图1.9所示。

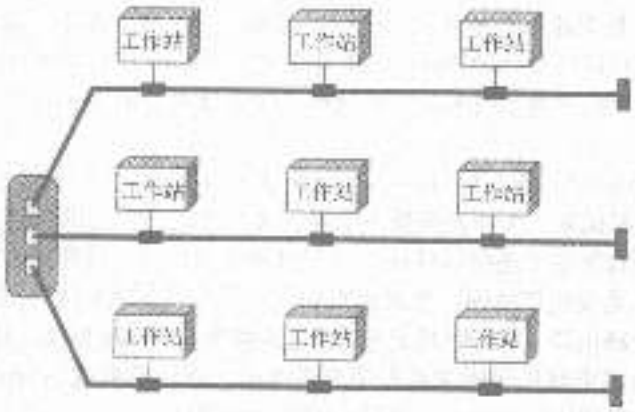


图1.9 混合型拓扑结构：一个星型结构为主干并具有三个总线结构的网络

1.2.4 网络模型

计算机网络是由许多实体组成，需要标准协调网络之间的通信。最著名的两个标准是OSI模型和因特网模型。在第2章中，将讨论这两个模型，OSI（开放系统互联）模型定义了网络7层模式，而因特网模型定义网络5层模式。本书是基于常常参照OSI模型的当特网模型。

1.2.5 网络分类

如今，当谈到网络时，通常指的是二种主要类型的网络：局域网和广域网。网络归于哪种类型主要取决于它的规模。局域网覆盖的范围小于2英里，广域网可以是世界范围。网络规模在这两者之间，常称为城域网，覆盖范围十几英里。

局域网

局域网（local area network，LAN）通常是专用的，连接的设备在一间简单的办公室、一幢建筑物或校园中（见图1.10），根据组织机构的需要和所采用技术的类型，局域网可能像办公室中只有两台PC机和一台打印机那样简单，也可能延伸到整个公司，并且包括音频和视频等外围设备。目前，LAN的规模限制在几公里的范围内。

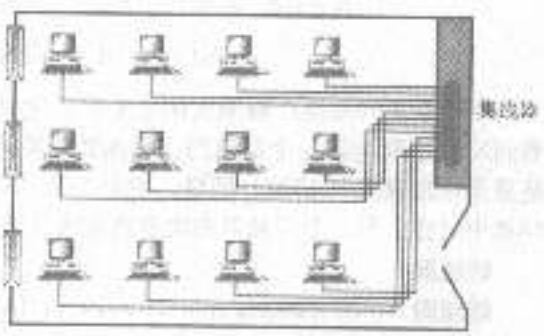


图1.10 在房间中12台计算机连接到一个集线器的独立局域网

LAN设计用来共享个人计算机或工作站之间的资源。共享的资源可能包括硬件（比如打印机）、软件（比如应用程序）或者数据。在商业环境中，LAN的常见的例子是用来连接与任务相关的工作组中的计算机，例如工程工作站或财务用PC，假定这些计算机中的某一台有大容量的磁盘驱动器，可以作为其他客户机的服务器。软件可以存储在这台中央服务器上，供整个工作组使用。在这种情况下，LAN的规模取决于每套软件拷贝的用户数目的许可限制，或者取决于操作系统允许访问的用户数目。

除了规模之外，LAN与其他类型的网络的区别还在于它的传输介质和拓扑结构。通常，给定的LAN仅使用一种类型的传输介质。最常见的LAN拓扑结构是总线结构、环状结构和星型结构。

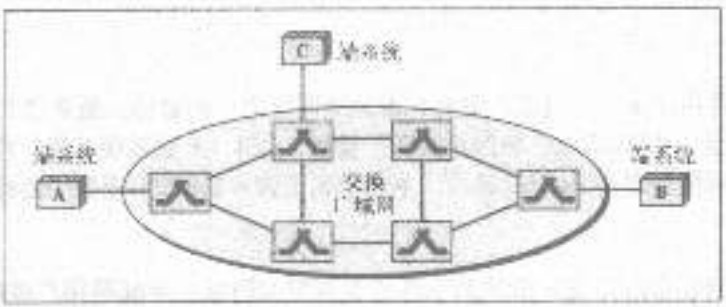
传统上，LAN的数据速率在每秒4－16兆位（Mbps）之间。现在，随着占位系统的开发，速度在不断增加以至可以达到100Mbps。LAN在第13、14和15章有详细讨论。

无线局域网是最新的局域网技术，在第14章中详细讨论无线局域网。

广域网

广域网（wide area network，WAN）提供远距离的数据、音频、图像和视频信息传输，传输跨越的地理区域包括国家、大陆甚至整个世界。将在第17章和第18章更详细地讨论广域网。

广域网可以复杂到用卡干连接因特网，也可以简单到用拨号线连接家用计算机到因特网。通常广域网首先指的是交换广域网，其次是点到点广域网（见图1.11），交换广域网连接端系统，该系统含有一个路由器（网际互联设备），它连接另一个局域网或广域网。点到点广域网通常是从电话局或有线电视电缆提供商租用专线连接家用计算机或小型局域网到因特网服务提供商（ISP）。这种类型的广域网时常用做对因特网的访问。



a) 交换广域网



b) 点到点广域网

图1.11　广域网：交换广域网和点到点广域网

一个最早的交换广域网实例是X.25，它是为端用户之间提供连接而设计的。在第18章将看到X.25逐渐地被一个高速的、更高效的称为帧中继网络所代替。交换广域网一个好的实例是异步传输模式（ATM）网络，它是一个具有固定大小的称为信元数据单元的网络，将在第18章中讨论。另一个广域网的实例是无线广域网，它越来越流行，我们将在第16章中讨论。

城域网

城域网（metropolitan area network，MAN）规模界于局域网与广域网之间，它通常的覆盖范围是一个乡镇或一个城市。它是为需要高速地连接因特网的用户而设计的，端点可分布在整个城市或城市的一部分。城域网的一个好的实例是可向用户提供高速DSL线路的电话网的一部分。另一个实例是有线电视网络，它原先是为有线电视而设计的，但是现在它也可用做到因特网的高速数据连接。在第9章中将讨论DSL线路和有线电视网。

1.2.6　网络互联：因特网

今天，很少看到单独的一个局域网、城域网或广域网。它们都是与另一个网络连接的。当两个或多个网络彼此连接，它们成为互联网络（internetwork）即互联网（internet）。

作为实例，假定有两个办公的机构组织，一个在东海沿，另一个在西海沿。在西海沿机

构已有一个总线拓扑结构的局域网，东海沿机构新建一个星型拓扑结构局域网。公司总裁住在中间某一地方，他需要在家中控制整个公司情况。为此创建连接三个实体（两个局域网和总裁的计算机）的主干广域网，租用一个已建立的交换广域网（由服务提供商所运行的，比如电信公司）。然而，为了连接两个局域网到这个交换广域网，要求三个点到点的广域网。这样的点到点广域网可以是由电话公司提供的高速DSL线路，或者是由有线电视商提供的电缆调制解调器线路，如图1.12所示。

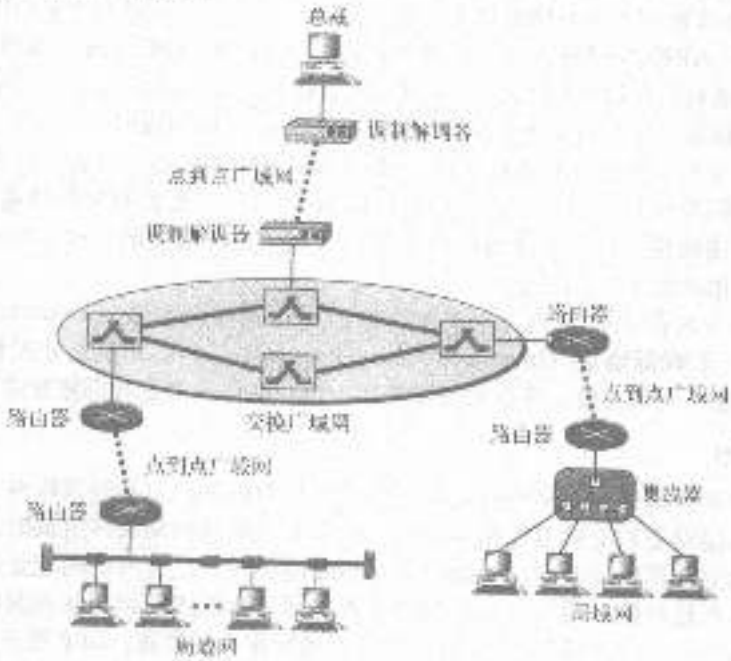


图1.12 由四个广域网和两个局域网组成的混合网络

1.3 因特网

因特网使我们日常生活的许多方面发生了革命性的变化，它不仅影响着我们生活娱乐的方式，还影响着我们商务活动的方式。考虑一下我们目前使用因特网的情况。你可能给业务伙伴发送电子邮件（E-mail），支付账单，阅读远方城市的报纸，查看电影预告——所有这些都是通过因特网实现的。或者，你可能研究医学课题，预订宾馆房间，与一个叫Trekkie的人聊天，选购汽车。因特网是一个通信系统，它把信息财富带到我们的指尖之上，将这些信息组织起来为我们所用。

因特网是一个结构化的、组织化的系统。首先需要介绍互特网的历史，然后描述因特网的现状。

1.3.1 历史简介

网络是由连接起来的通信设备（例如计算机和打印机）组成的集合。互联网络（internet，注意小写字母i）是两个或多个可以互相通信的网络。最为著名的互联网络称为因特网（Internet，大写字母I），一个由成千上万的互相连接的网络组成的集合。个人，以及100多个国家的政府部门、学校、研究机构、公司和图书馆等各种组织都在使用因特网。它拥有无数的用户，然而，这个非凡的通信系统直到1969年才刚刚诞生。

在20世纪60年代中期，各研究机构的大型机是相互独立的设备。来自不同制造商的计算

机之间不能互相通信。美国国防部（DoD）高级研究计划局（Advanced Research Project Agency，ARPA）想找到一种能将计算机连接起来使研究人员能共享他们的研究成果的方式，以达到降低成本、避免重复劳动的目的。

1967年，在美国计算机学会（ACM）会议上，ARPA提出了ARPANET的概念，即一个由互相连接的计算机组成的小型网络。它的主要思想是每台主机（不必是出自同一个制造商的产品）应该连接到一台称为接口报文处理器（IMP）的专用计算机上，IMP也相互之间进行连接。各个IMP不仅能够与各自所连接的主机进行通信，同时也能够与其他IMP进行通信。

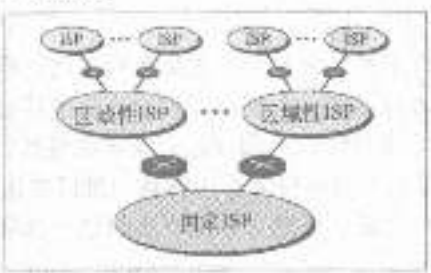
到了1969年，ARPANET最终实现。网络是由4个节点通过IMP连接而成的，这4个节点是加利福尼亚大学洛杉矶分校（UCLA）、加州大学圣塔芭芭拉分校（UCSB）、斯坦福研究所（SRI）和犹他大学。由称为网络控制协议（NCP）的软件提供主机间的通信。

1972年，Vint Cerf和Bob Kahn，两个人都是ARPANET核心组的成员，在一个他们称为网络互联项目的研究项目上进行合作。Cerf和Kahn在1973年发表的具有里程碑意义的论文，概要论述了端到端的信息分组发送协议的实现。这篇关于传输控制协议（TCP）的论文包含了封装、数据报和网关功能等概念。

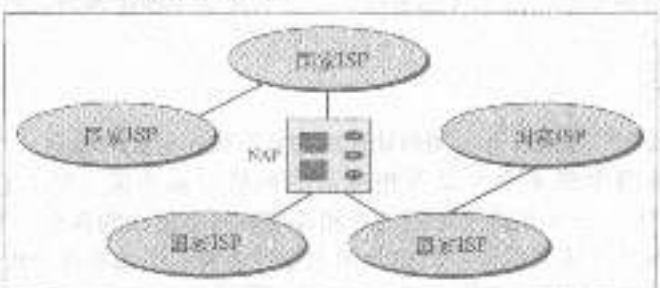
此后不久，专家们决定将TCP分成两个协议：**传输控制协议**（Transmission Control Protocol，TCP）和**网际协议**（Internetworking Protocol，IP）。IP负责处理数据报路由，而TCP负责高层功能，比如分段、重组和差错检测。该互联协议就是我们熟知的TCP/IP协议。

1.3.2 因特网现状

从20世纪60年代以来，因特网的发展经历了漫长的道路。当今的因特网不是简单的层次结构，而是由连接设备和交换工作站结合起来的众多广域网和局域网组成的。很难确切地表示因特网，因为它在不断地改变——新的网络不断地加入，已有网络的地址在不断增加，倒闭公司的网络不断地废除。现在大多数想建立因特网连接的终端用户使用**因特网服务提供商**（Internet Service Provider，ISP）提供的服务。有国际服务提供商、国家级服务提供商、区域性服务提供商和本地服务提供商。因特网现在是由私有公司而不是由政府运行的。因特网的示意图（而不是地理图）如图1.13所示。



a) 国家ISP的结构



b) 国家ISP的互联

图1.13　因特网层次结构组织

国际服务提供商

在层次结构的顶端是将国家连接起来的国际服务提供商。

国家服务提供商

国家服务提供商（National Service Provider，NSP）是由专门的公司创建和维护的主干网络。在北美运营着许多NSP，其中比较著名的是Sprint Link、PSINet、UUNet Technology、AGIS和Internet MCI。为了提供终端用户之间的连接，这些主干网络由复杂的交换工作站（通常由第三方运营）连接，称为网络访问点（network access point，NAP）。有些NSP网络也是由称为对等点的私有交换站连接起来的。NSP通常以很高的数据速率运行（高达600Mbps）。

区域性因特网服务提供商

区域性因特网服务提供商即区域性ISP（regional ISP）是连接到一个或多个NSP的小型ISP。区域性ISP以较低数据速率运行在第二层。

本地因特网服务提供商

本地因特网服务提供商（local Internet service provider）提供到终端用户的直接服务。本地ISP可以连接到区域性ISP或直接连接到NSP。大多数终端用户连接到本地ISP。注意，从这种意义上讲，本地ISP可以是一个仅仅提供因特网服务的公司，或者是一个为其自身员工提供网络服务的企业，或者是一个运行有自己网络的非盈利组织（比如学院或者大学）。这些本地ISP可以连接到区域性的或者国家服务提供商。

1.4 协议和标准

在本节中，定义两个广泛使用的词：协议和标准。首先定义协议，它是规则的同义词，然后讨论标准，它是经过协商达成一致的规则。

1.4.1 协议

在计算机网络中，通信发生在不同系统的实体之间。实体是能够发送和接收信息的任何事物。然而，两个实体间仅发送位流就指望能相互理解是不可能的。要实现通信，实体之间必须遵循协议。协议是用来管理数据通信的一组规则。协议规定了通信的内容、通信的方式和通信的时间。协议的核心要素是语法、语义和时序。

语法（syntax）指的是数据的结构或格式，即它们是以何种顺序表示的。例如，一个简单的协议可能将第一个8位作为发送方的地址，第二个8位作为接收方的地址，信息流的其余部分作为报文本身。

语义（semantics）指的是每一个位片断的含义：如何解释一个特别的位模式，基于该解释应该采取什么操作？例如，地址是否标示了路由，是否标示了报文的最终目的地址。

时序（timing）指的是两个特性：报文发送的时间和发送的速率。例如，如果发送方以100Mbps的速率发送而接收方只能以1Mbps的速率处理数据，那么传输中会使接收方过载而造成数据的大量丢失。

1.4.2 标准

在创立和维护对设备制造商开放并具有竞争力的市场方面，以及在保证国家及国际的数据及电信技术和过程的合作方面，标准都是至关重要的。在确保目前的市场及国际通信中的互连接必要性方面，标准为制造商、销售商、政府部门和其他服务提供商提供了一系列准则。数据通信标准分为两类：事实的（de facto，意思是依据事实或依据惯例）和法定的（de jure，意思是依据法规或依据规则）。

- **事实标准**。那些未经组织团体承认但已经在广泛使用中被接受的标准就是事实标准（de facto standard）。事实标准通常是由制造商在打算定义新的产品或技术的功能时最初建立的。
- **法定标准**。那些由官方认可的团体制定的标准称为法定标准（de jure standard）。

1.4.3　标准化组织

标准是通过标准创建委员会、论坛以及政府管理机构之间的合作开发出来的。

标准创建委员会

尽管致力于制定标准的组织很多，但数据电信方面的标准，就北美而言主要由以下几个组织发布：

- **国际标准化组织（ISO）** ISO是一个国际性组织，其成员主要来自世界各地的不同政府的标准化创建委员会。ISO的活动范围包括科学、技术和经济活动领域的开发合作。
- **国际电信联盟——电信标准分会（ITU-T）** 在20世纪70年代早期，许多国家都在制定各自的国家电信标准，但各国的标准之间是不兼容的。为此，联合国建立了一个称做国际电报电话顾问委员会（Consultative Committee for International Telegraphy and Telephony，CCITT）的协会，并将其作为国际电信联盟（ITU）的一部分。该协会致力于研究和建立通用的电信标准，以及专门用于电话和数据系统的电信标准。1993年3月1日，该协会的名称改为国际电信联盟——电信标准分会（ITU-T）。
- **美国国家标准协会（ANSI）** 尽管名字上是这样的，但美国国家标准协会是一个完全私有的、非营利性的、不附属于美国联邦政府的机构。然而，ANSI所有的活动都是在美国及其国民的扶持下进行的，具有非常重要的影响。
- **电气和电子工程师协会（IEEE）** 电气和电子工程师协会是世界上最大的国际性专业工程师社团，其目的是在电气工程、电子、无线电以及所有相关分支工程领域中，促进理论研究的深入、创新活动的开展和产品质量的提高。IEEE目标之一是监督计算机及通信的国际标准的制定和采纳。
- **电子工业协会（EIA）** 电子工业协会与ANSI结成联盟，它也是一个非营利组织，致力于提升电子制造的影响力。除了标准的制定以外，其活动包括公众意识教育和游说活动。在信息技术领域，EIA为数据通信定义物理连接接口和电子信号规程作出了巨大贡献。

论坛

电信技术的发展速度已经超出了标准委员会批准标准的能力。标准委员会都是按部就班、按照自然而缓慢的速度工作的，为了适应推动标准化进程的工作模式及规范的需要，许多特定利益团体发展了不同的论坛（forums），论坛代表来自于感兴趣的相关公司、论坛、院校以及用户，共同测试、评估和标准化新技术。由于他们专注于某一具体的技术，论坛能够在电信领域加速这些技术的采纳和使用。论坛将他们的结论提交给标准化团体。

管理机构

所有的通信技术都必须服从诸如美国联邦通信委员会（Federal Communications Commission，FCC）的政府机构的管辖。这些机构的目的是通过制定无线电、电视和有线通信的规范来保护公众利益。在通信相关的商务领域，FCC具有国内的和国际的权威性。

1.4.4　因特网标准

因特网标准（Internet standard）是经过完全测试的规范，它是对使用因特网的人们有用的并且被他们所拥护的标准。它是必须遵循的格式化规则。一份规范要想成为因特网标准，

需要经过严格的程序。规范最初以草案开始。因特网草案（Internet draft）是一份没有官方地位的工作文档，有6个月的生命期，经过因特网权威机构的推荐，草案能够以请求评论（Request for Comment，RFC）的形式发表。每份RFC文档经过编辑后赋予一个编号，对所有感兴趣的团体开放。RFC文档经过成熟度考察并根据其需求程度进行归类。

推荐读物

为了更详细地讨论与本章有关的主题，推荐下列读物与网站。在本书末列出了括号[……]内的参考资料。

读物

本章所涉及的资料可在[Sta04]和[PD03]中找到。[Tan03]的1.6节讨论标准。

网站

下列网站有与本章有关的主题。

- www.acm.org/sigcomm/sos.html 该网站给出各种网络标准。
- www.ietf.org/因特网工程任务组（IETF）的主页。

请求评论（RFC）

下列网站是所有RFC的列表，包括IP和TCP。在以后各章中引用RFC的有关材料。

- www.ietf.org/rfc.html

本章小结

- 数据通信是指通过某种介质将数据从一台设备传输到另一台设备。
- 数据通信必须将数据准确及时地传输到正确的目的设备。
- 数据通信系统的五个组成部分是报文、发送方、接收方、介质和协议。
- 文本、数字、图像、音频和视频是信息的不同形式。
- 两台设备之间的数据流可以是以下三种方式之一：单工、半双工和全双工。
- 网络是由介质链路连接起来的一组通信设备。
- 在点到点连接中，一条专用链路只能连接两台设备。在多点连接中，三台或三台以上的设备共享一条链路。
- 拓扑结构指的是网络物理上或逻辑上的分布。设备可能以网状、星型、总线或环状拓扑结构分布。
- 网络可分为局域网（LAN）和广域网(WAN)。
- LAN是指在建筑物内、工厂里、校园中或者相邻建筑物之间的数据通信系统。
- WAN是指跨越多个州或国家甚至整个世界的数据通信系统。
- 互联网是网络的网络。
- 因特网是由许多独立的网络组成的集合。
- 因特网服务提供商可分为本地的、区域性的、国家的和国际的。
- 协议是管理数据通信的一组规则，协议的核心要素是语法、语义和时序。
- 标准的重要性在于它能确保不同制造商的产品按照人们所期望的那样进行协作。
- ISO、ITU-T、ANSI、IEEE和EIA是一些参与标准制定的组织。
- 论坛是一些具有特殊兴趣的团体，它们能够很快地评估和标准化新的技术。
- 请求评论包含了很多思想或概念，用于产生因特网标准。

习题

复习题

1. 说出数据通信系统的五个组成部分？
2. 分布式处理的优点是什么？
3. 一个经济的、高效的网络的三个必要标准是什么？
4. 与点到点连接相比，多点连接有什么优点？
5. 线配置有哪两种类型？
6. 根据线配置，对四种网络拓扑结构进行分类。
7. 半双工和全双工传输模式的区别是什么？
8. 试说出四种网络拓扑结构的名称和各自的优点。
9. 在一个有n台设备的网络中，网状、环状、总线结构和星型拓扑结构各需要多少根的线缆？
10. 一个通信系统是LAN还是WAN，由哪些因素决定？
11. 什么是互联网？什么是因特网？
12. 为什么说协议是必需的？
13. 为什么说标准是必需的？

练习题

14. 用统一码表示的字符或符号的最大个数是多少？
15. 一个彩色图用16位表示一个颜色像素，它最大能表示多少种颜色？
16. 假设一种网状拓扑结构中有六台设备，需要多少根线缆？每台设备需要多少个端口？
17. 讨论下列四种网络中，如果一条连接失败会出现什么后果？
    a. 五台设备按照网状拓扑结构配置。
    b. 五台设备按照星型拓扑结构配置（不将集线器计算在内）。
    c. 五台设备按照总线拓扑结构配置。
    d. 五台设备按照环状拓扑结构配置。
18. 你在家中用因特网集线器连接两台计算机，这是局域网、城域网还是广域网？并说明理由。
19. 在图1.8的环状拓扑结构中，如果有一个工作站没有连接好，将会发生什么事情？
20. 在图1.7的总线拓扑结构中，如果有一个工作站没有连接好，将会发生什么事情？
21. 试画出一个混合拓扑结构图，该结构有一个星型的主干网和三个环状网。
22. 试画出一个混合拓扑结构图，该结构有一个环状主干网和两个总线网。
23. 当你使用因特网时，下列哪个应用更令人感到延迟的负面效应？
    a. 发送电子邮件。
    b. 复制文件。
    c. 访问因特网。
24. 当一方给另一方打本地电话时，这是一个点到点连接还是多点连接？并说明理由。
25. 比较电话网和因特网，它们有什么相似之处？有什么不同之处？

探究题

26. 利用网站www.cne.gmu.edu/modules/network/osi.html 讨论OSI模型。
27. 利用网站www.ansi.org 讨论ANSI活动。
28. 利用网站www.ieee.org 讨论IEEE活动。
29. 利用网站www.ietf.org/ 讨论各种类型RFC。

第2章 网络模型

网络利用硬件和软件的组合将数据从一个地点发送到另一地点。硬件由能够将信号从网络的一点传输到另一点的物理设备组成。软件是由指令集合组成的，它使我们所期待的来自网络的服务成为可能。

可以将网络的任务与运用计算机解决数学问题的任务进行比较。计算机解决数学问题时最基础的工作是由计算机硬件完成的。然而，如果只使用硬件，这将是一项十分冗长乏味的工作。我们需要对每个存储和处理数据的内存位置进行跟踪。如果软件能够掌握这些情况，那么就可以很容易完成这些任务，在最高层，程序能够指导问题解决的过程，而像实际的硬件该如何完成任务之类的细节问题，则留给软件层来完成。这些软件层是由更高的层来调用的。

计算机网络的情形与之相似。例如，将一份电子邮件从世界上的一个地点发送到另一个地点的任务可以划分成几个子任务，每个子任务都是由一个独立的软件包来完成。每一份软件都使用另一软件包提供的服务来完成自己的工作。在最底层，将一个信号或一组信号从源计算机发送到目的计算机。

本章给出网络层次的总体概念，并讨论每一层各自的功能。这些层在以后的各章中详细描述。

2.1 任务分层

在日常生活中，我们就使用层次的概念。举个例子，假设有两位朋友通过邮寄信件互相通信。如果没有邮局提供的服务，将信件发送给朋友的过程将是十分复杂的。图2.1显示了该任务是如何完成的。

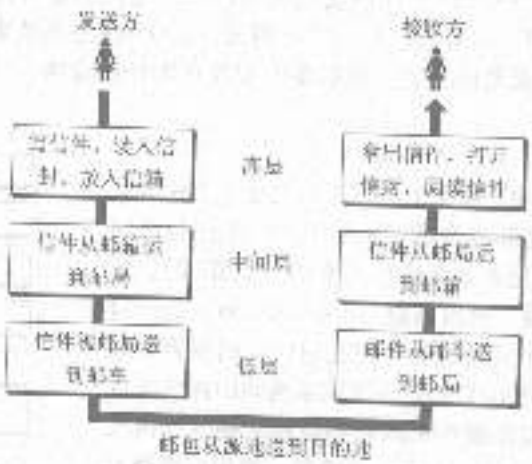


图2.1 发送信件所包含的任务

2.1.1 发送方，接收方和载体

在图2.1中有一个发送方，接收方和传输信件的载体。任务具有层次结构。

在发送方

让我们按顺序描述一下在发送方发生的活动。

- 高层　发送方写信，将信件装入信封，写下发送方和接收方的地址，将信件投入信箱。
- 中间层　信件被邮递员取出并送到邮局。
- 低层　邮局对信件进行分拣，邮车转运信件。

路途中

然后，邮件走上到达接收方的路途。在到达接收方的本地邮局的路途中，它实际上还要经过中央邮局。另外，信件可能是通过卡车、火车、飞机、船或者以上交通工具的组合进行转运的。

在接收方

- 低层　邮车将信件转运到邮局。
- 中间层　信件分拣后递送到接收方的信箱。
- 高层　接收方取出信件，打开信封，读信。

2.1.2 层次结构

根据分析，在发送方有三个不同的活动，在接收方也有另外三个活动。在发送方和接收方之间转运信件的任务是由邮车完成的。不能一眼看出来的是，这些任务必须按照给定的层次顺序完成。在发送方，信件必须是先写出、投进邮箱，然后才能让邮递员拣起再递送到邮局。在接收方，信件必须先放入接收方信箱，然后才能被接收方取出并被阅读。

服务

在接收方，每一层都使用其直接下层提供的服务。在高层的发送方使用中间层提供的服务，中间层使用低层提供的服务，低层使用邮车提供的服务。

1990年以前，在数据通信和网络文献资料中占有主要地位的分层结构模型是**开放系统互联**（Open System Interconnection, OSI）模型。大家都认为OSI模型会成为数据通信的最终标准，但这种情况没有出现。TCP/IP协议族成为占统治地位的业务体系结构，由于该协议族已在因特网中被大量采用，并为其已作过广泛测试，而OSI模型却从未完全实现过。

本章将对OSI模型作简要的讨论，然后集中关注TCP/IP协议族。

2.2 OSI模型

国际标准化组织（ISO）成立于1947年，它是一个致力于在世界范围内建立统一国际标准的多国组织。一个能包含网络通信各方面的ISO标准是开放系统互联模型（见图2.2）。它最初于20世纪70年代形成，**开放系统**（open system）是一组协议。它允许任何两个不同系统进行相互通信，而不管其底层结构如何。建立OSI模型的目的是使不同系统间的通信变得容易，而不要求对其底层的硬件和软件的逻辑结构做任何改动。OSI模型并不是一个协议，而是一个模型。利用该模型可以理解并设计一个灵活、健壮和可互操作的网络体系结构。



图2.2 OSI模型

> ISO是组织，OSI是模型。

OSI模型提供了一个用来进行网络系统设计的层次化框架。它由7个相互独立但又有互相

关联的层次组成。每一层都定义了通过网络传递信息的一部分功能（见图2.2）。深刻理解OSI模型的基本概念是探索数据通信领域知识的一个坚实基础。

2.2.1 层次化体系

OSI模型由7个有序的层组成：物理层（第一层）、数据链路层（第二层）、网络层（第三层）、传输层（第四层）、会话层（第五层）表示层（第六层）以及应用层（第七层）。图2.3表示了当一个报文从设备A到设备B传输时所涉及的层次。当报文从设备A到设备B时，它可能通过许多中间节点。这些中间节点通常只涉及OSI模型的前三层。



图2.3 OSI模型中层次间的交互

在建立模型的过程中，设计者将传输数据的过程归结为一些最基本的元素。他们确定了具有相关用途的网络功能，并将这些功能归类到一些独立的功能组。这些功能组就形成了各个层次。每一层都定义了一系列与其他层次不同的功能。通过这种方式进行功能定义和局部化，使设计者能建立一个既全面又有灵活性的体系结构。最重要的是，OSI模型允许在不兼容系统之间的完全透明化。

在单台机器中，每一层调用下一层提供的服务，例如，第三层使用第二层提供的服务，同时也向第四层提供服务。在两台机器之间，一台机器上的第x层与另一机器的第x层进行通信。这种通信由一系列称为协议的规则和约定来控制。每台机器上的特定层次的通信过程，因而被称为对等过程（peer-to-peer process）。因此，机器间的通信就是使用适用于给定层次的协议的对等过程。

2.2.2 对等过程

在物理层，通信是直接发生的。在图2.3中，设备A将位流传送到设备B（通过中间节点）。但是，在更高层次中，通信必须先在设备A中从上层传到下层，再传输到设备B，然后在设备B中再从下层传输到上层。在发送端，每一层都在从直接上层传来的报文加上自己的信息，并将整个分组传到它的直接下层。

在第一层，整个分组被转换成可以传输到接收端设备的形式。在接收端，报文被一层一层地打开，每一层接收并提取对它有意义的数据。例如，第二层将提取对它有意义的数据，

并将剩下的传输给第三层。第三层提取对它有意义的数据，并将剩下的传输给第四层，依此类推。

层间接口

发送设备将数据和网络信息从高层向低层传递，而接收设备将这些信息由低层向高层传递，这个过程是通过相邻两层的接口（interface）来实现的。每一层接口都定义了该层必须向上层提供信息和服务。良好定义的接口和层功能可使网络模块化。只要某一层仍能为上一层提供所期望的服务，该层功能的具体实现就可以在不改变周围层次的基础上进行修改和替换。

层次组织

7个层次可以为分属于3个子功能组。第一层、第二层和第三层（物理层、数据链路层和网络层）是网络支持层，它们处理从一设备到另一个设备数据传输的物理方面问题（比如，电气特性、物理连接、物理寻址、传输的时序和可靠性）。第五层、第六层和第七层（会话层、表示层和应用层）可认为是用户支持层，它们允许不相关的软件系统间的互操作。第四层即传输层，将这两个子功能组连接起来，并保证下层是以上层能够使用的形式传输的。OSI的高层常通过软件来实现，而低层则是由硬件和软件混合实现，但物理层几乎都是由硬件实现。

在图2.4中描述了OSI层次的整体结构，D7指第七层的数据，D6指第六层的数据，依此类推。整个过程从第七层（应用层）开始，然后一层层向下移动，在每一层，数据单元可能被加入头部（header）或者尾部（trailer），通常在第一层仅加上尾部。当格式化的数据单元经过物理层（第一层）时，它被转换成电磁信号，并在物理链路上传输。

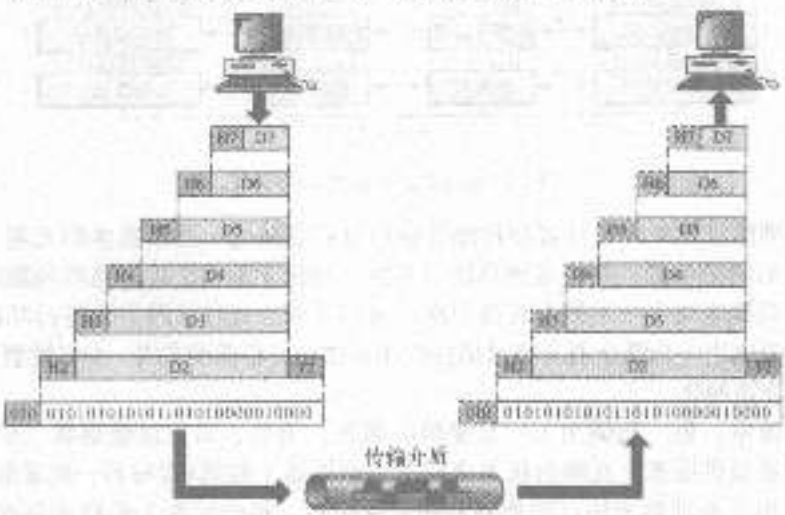


图2.4　使用OSI模型的传输过程

到达目的地后，信号传入第一层并被重新转换成数字形式。然后，数据单元由下向上传到OSI各层。当每一数据块到达下一个更高层时，在发送端对应层次附加的头部和尾部就被除去，并执行该层相应的动作。数据到达第七层时，报文又回到应用层的格式，并可以为接收者使用。

2.2.3　封装

图2.3也表示了OSI模型数据通信另一个问题：封装问题。在第七层的一个分组（头部和数据）封装在第六层的一个分组中，第六层的整个分组封装在第五层的一个分组中，依此类推。

换言之，第$N-1$层的分组中的数据部分是第N层的完整分组（数据、头部、也可能有尾部）。

这种概念称为封装，第$N-1$层不知道被封装分组中，那些部分是数据，那些部分是头部或尾部。对于第$N-1$层来说，来自第N层的分组是作为一个整体单元处理的。

2.3 OSI模型的各层功能

在本节中，简要描述OSI模型每一层的功能。

2.3.1 物理层

物理层（physical layer）包含那些在物理介质上传输位流所必需的功能。它定义了接口与传输介质的机械和电气特性，也定义了物理设备和接口为了传输而必须执行的过程和功能。图2.5表示了对应传输介质与数据链路层的物理层的位置。

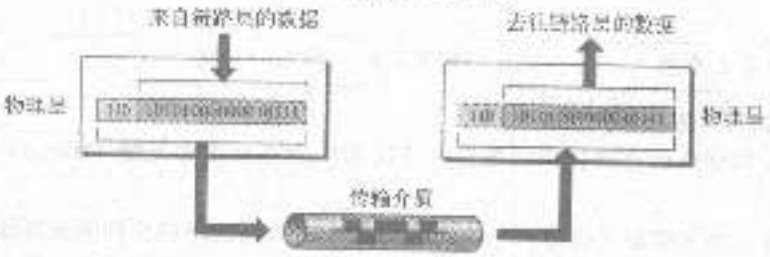


图2.5 物理层

物理层负责位从一个节点到另一个节点的传递。

物理层考虑以下一系列问题：

- **接口与介质的物理特性**。物理层定义了设备与传输介质之间的接口特性，也定了传输介质的类型。
- **位的表示**。物理层的数据是没有任何解释的位（bit）流（由0和1所组成的序列）。要进行传输，位必须编码成信号——电信号或光信号。物理层定义**编码**（encoding）的类型（如何将0和1转换成信号）。
- **数据速率**。**传输速率**（transmisson rate），即每秒发送的位数，也在物理层定义。换言之，物理层也定义一个位持续多长时间。
- **位同步**。发送方与接收方不仅使用相同的比特率，还必须位同步。换言之，发送方的时钟与接收方的时钟必须同步。
- **线路配置**。物理层涉及设备与介质的连接。在点到点配置中，两个设备通过一条专用链路连接。在多点配置中，许多设备共享一条链路。
- **物理拓扑结构**。物理拓扑结构定义如何将设备连接成网络。设备的连接方式可以是网状拓扑结构（每台设备均与其他设备连接）、星型拓扑结构（通过中心设备与其他设备连接）、环状拓扑结构（每台设备与下一台设备连接以组成一个环）和总线拓扑结构（每台设备都在一条公共链路上）。
- **传输方式**。物理层也定义两台设备之间的传输方向：单工、半双工和全双工。在单工方式中，只有一个设备能发送，另一个设备只能接收。单工方式是单向通信。在半双工方式中，两台设备都能发送和接收，但不能在同一时刻。在全双工（或双工）方式中，两个设备能在同一时刻发送与接收。

2.3.2 数据链路层

数据链路层（data link layer）将物理层一对数据不做任何改动的传输通道变成可靠的

链路，这样可以将物理层的数据无差错地传递给上层（网络层）。图2.6表明了数据链路层与网络层及物理层之间的关系。

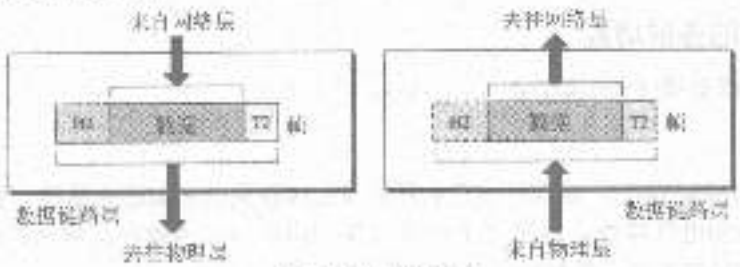


图2.6 数据链路层

> 数据链路层负责将从一跳（节点）到下一跳（节点）传递。

数据链路层的具体职责如下：

- **成帧**。数据链路层将接收到的来自网络层的位流划分成称为帧（frame）的易处理数据单元。
- **物理寻址**。如果帧是发送给网络中不同系统，则数据链路层在帧的头部添加发送方的物理地址与接收方的物理地址。如果帧要发往发送方网络之外的系统，那么接收方的地址就是连接一个网络到下一个网络的设备地址。
- **流量控制**。如果接收方接收数据的速率小于发送方发送的速率，那么数据链路层就会采用流量控制机制以防止接收方过载。
- **差错控制**。数据链路层增加一些机制用来检测与重发损坏帧或丢失帧，从而增加了物理层的可靠性，还用了一个机制防止重复帧。差错控制通常在一个帧的结束处增加一个尾部来处理。
- **访问控制**。当两台或多台的设备连接到同一条链路上时，数据链路协议必须能决定在任意时刻由哪一台设备来获取对链路的控制权。

图2.7说明数据链路层**跳到跳传递**（hop-to-hop delivery），或者称为**节点到节点传递**（node-to-node delivery）。

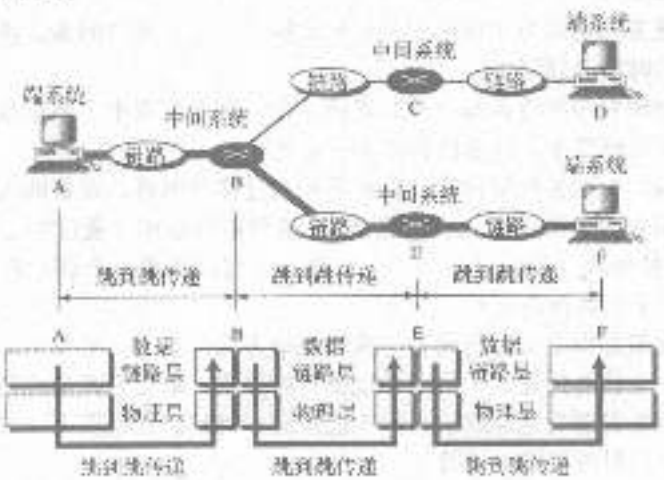


图2.7 跳到跳传递

如图2.7所示，两个相邻节点在数据链路层的通信。节点A发送数据到节点F要经过三次传递。首先，节点A的数据链路层发送一个帧到节点B（路由器）的数据链路层；其次节点B的数据链路层发送一个新帧到节点E的数据链路层；最后节点E数据链路层发送一个新帧到节点F的数据链路层。注意：三个节点之间交换的帧具有不同的头部值。从A到B，以B作为目的地址和以A作为源地址；从B到E，以E作为目的地址和以B作为源地址；从E到F，以F作为目的地址和以E作为源地址。如果差错检测包括帧的头部，则尾部的值也可以是不同。

2.3.3 网络层

网络层（network layer）负责将分组从源地址传送到目的地址，可能会通过多个网络（链路）。尽管数据链路层会监视同一个网络（链路）上两个系统之间分组的传递，但是还需要网络层确保每个分组能够从出发点到达目的地。

如果两个系统连接在同一条链路上，则通常不需要网络层。然而，如果两个系统在不同的网络（链路）上并通过网络（链路）之间的设备连接，通常就需要网络层以完成源端到目的端的传递。图2.8表示了网络层与数据链路层及传输层的关系。

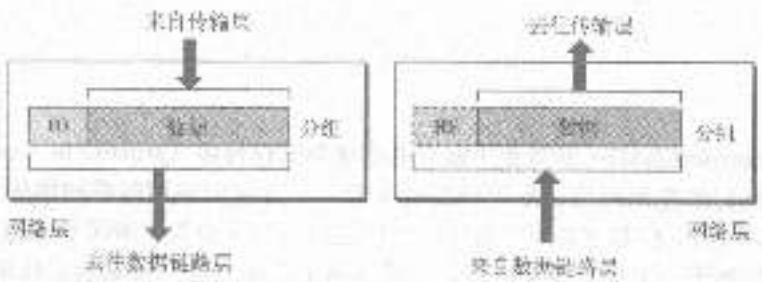


图2.8 网络层

网络层负责将各个分组从源地址传送到目的地址。

网络层的具体任务是：

- **逻辑寻址**。物理寻址负责处理本地网络寻址问题，它由数据链路层完成。如果一个分组通过网络边界，就需要另一个寻址系统来帮助区分源与目的系统。网络层给来自上一层的分组增加一个头部，它包括发送方与接收方的逻辑地址。在本章后面讨论逻辑地址。
- **路由选择**。当一些独立网络或链路连接在一起构成一个互联网（一个由网络连成的网络）或一个更大的网络时，连接设备（称为路由器或网关）把分组送到它们的最终目的地，网络层的功能之一是提供这种机制。

图2.9说明了网络层端到端的传递。

如图2.9所示，现在我们需要一个源到目的传递。节点A的网络层发送一个分组到节点B的网络层，当分组到达路由器B时，则该路由器按照该分组的最终目的地（节点F）做出决定。在以后几章中，将会看到路由器B利用它的路由表寻找下一跳的路由器，该路由器是E，所以，节点B的网络层发送分组到节点E的网络层，节点E的网络层再发送分组到节点F的网络层。

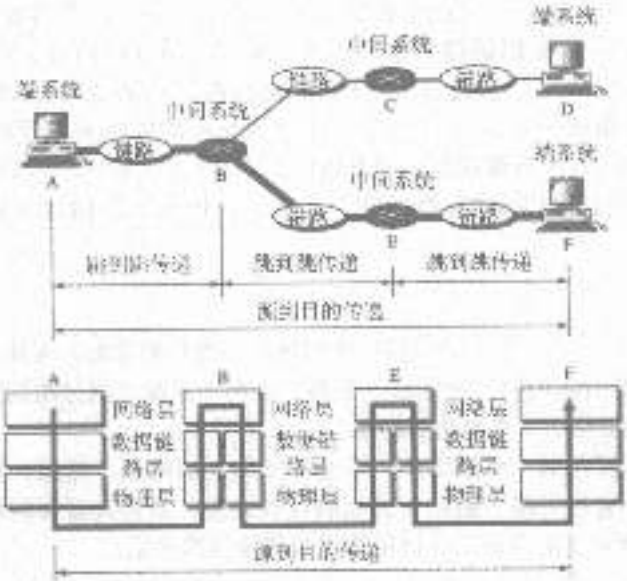


图2.9 端到目的传递

2.3.4 传输层

传输层（transport layer）负责整个报文的进程到进程传递（process-to-process delivery）。进程是主机上运行的应用程序。尽管网络层监管各个分组的源端到目的端传递（source-to-destination delivery），但是它并不了解这些分组之间的相互关系。网络层独立处理每个分组，就像每个分组都属于一个独立报文一样。不管实际上是否这样。传输层正好相反，它确保整个报文无差错并按顺序地到达目的地，并在源到目的层次进行差错控制和流量控制。图2.10表示了传输层与网络层及会话层关系。

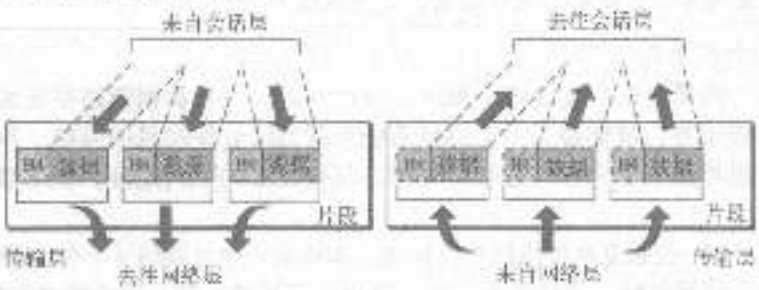


图2.10 传输层

> 传输层负责一个报文从一个进程到另一个进程的传递。

传输层的具体任务如下：

- **服务点寻址。**常常会有许多程序同时在一台计算机上运行，因此源端到目的端的传递就意味着不仅是从一台计算机传递到另一台计算机，而是从一台计算机上的一个特定进程（运行的程序）传递到另一台计算机上的一个特定进程（运行的程序）。因此，传输层信息的头部就必须包含服务点地址（也叫端口地址）。网络层将每个分组传送到指定的计算机上，而传输层则将整个报文传送给该计算机上的指定进程。

- **分段和组装**。将报文分解成可传输的片段，并且给这些片段编上序号。这些序号不仅使传输层可以在接收端将报文正确地组装，而且可以用来标识和替换传输中丢失的分组。
- **连接控制**。传输层可以是无连接的或面向连接的。无连接的传输层将每个片段作为一个独立的分组，并将它传递到目的机器的传输层。而面向连接的传输层，在传递分组之前，需要与目的机器的传输层建立一条连接。所有数据被传输后，该连接释放。
- **流量控制**。如同数据链路层一样，传输层担负流量控制（flow control）的责任。然而，这一层的流量控制是在端到端上，而不是在单条链路上。
- **差错控制**。如同数据链路层一样，传输层担负差错控制（error control）的责任。然而，这一层的差错控制是在进程到进程上，而不是在单条链路上。发送方的传输层确保整个报文无差错（损坏、丢失或重复）地到达接收方的传输层。通常通过重发来纠正差错。

图2.11说明了传输层进程到进程的传递。



图2.11 一个报文的进程到进程的可靠传递

2.3.5 会话层

前面3层（物理层、数据链路层和网络层）提供的服务对某些进程是不需要的。会话层（session layer）是网络的对话控制器，它建立和维护以及同步通信系统间的交互操作。

> 会话层负责对话控制和同步。

会话层的具体任务是：

- **对话控制**。会话层允许两个系统进入会话。它允许两个进程之间以半双工（一次一个方向）或全双工（一次两个方向）方式进行通信。
- **同步**。会话层允许一个进程在数据流中增加检查点或同步点（synchronization point）。

例如，假定一个系统要传输2000页的一个文件。为了确保每收到100页就独立地进行确认，每100页后应适当插入一个检查点。在这种情况下，如果传输到第532页发生崩溃，则在第501页处开始重传。而第1页到第500页不必重传。图2.12描述了会话层与传输层及表示层的关系。

2.3.6 表示层

表示层（presentation layer）是两个系统之间交换信息的语义和语法。图2.13描述了表示层与会话层及应用层的关系。

> 表示层负责翻译、加密和压缩数据。

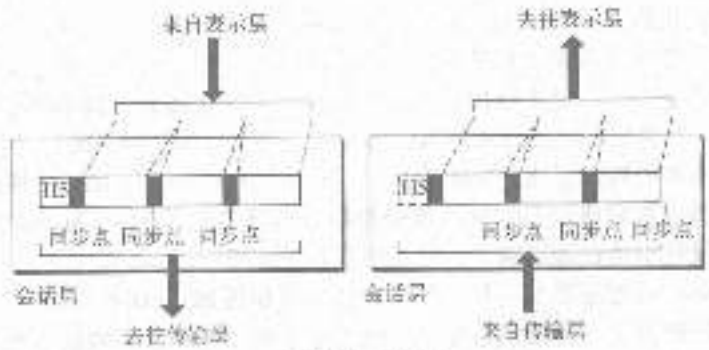


图2.12　会话层

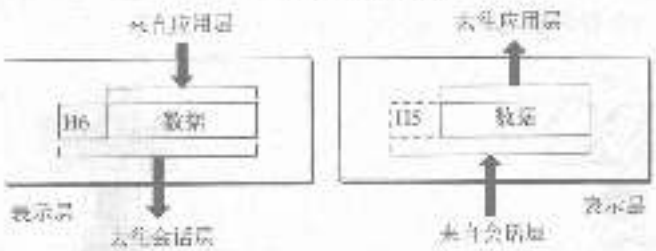


图2.13　表示层

表示层的具体任务是：

- 翻译。两个系统中的进程（运行程序）通常以字符串、数字等格式来交换信息。在传输前，信息必须变换成位流。因为不同计算机采用不同的编码系统，因此表示层负责不同编码方法之间的互操作。发送方的表示层将信息从发送方使用的格式转换成一个公共的格式。接收方机器的表示层将公共格式转换成接收方使用的格式。
- 加密。为了传输敏感信息，一个系统必须保证秘密。加密是指发送方将原来的信息转换成另一种形式，并在网络上发送转换后的形式。解密是将信息转换回它的原始形式。
- 压缩。数据压缩减少需要传输数据的位数。数据压缩对于多媒体（如文本、音频和视频）的传输显得特别重要。

2.3.7　应用层

应用层（application layer）使得用户（不管是人还是软件）可以访问网络。它提供了用户接口和服务支持。比如电子邮件、远程文件访问与传输、共享数据库管理以及其他分布式信息服务。

图2.14说明了应用层与用户和表示层的关系。在许多现有的应用服务中，图中只表示了三

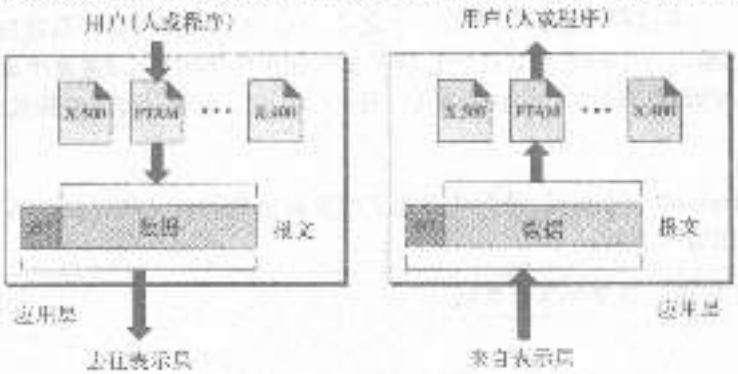


图2.14　应用层

种：X.400（报文处理服务）、X.500（目录服务）和文件传输，访问和管理（FTAM）。图中用户正在使用X.400发送电子邮件。

应用层负责向用户提供服务。

应用层的具体服务如下：

- **网络虚拟终端。**网络虚拟终端是物理终端的软件版本，允许用户登录到一台远程主机上。为此，应用程序在远程上机上创建一个软件模拟终端。用户计算机只与这个软件终端对话，而软件终端与主机对话，反之亦然。而远程上机认为自己是在与自己的一个终端通信，因而允许用户登录。
- **文件传输、访问和管理。**这项应用允许用户访问远程计算机上的文件（进行修改或读取数据），从远程计算机上检索文件，以及管理或控制远程计算机上的文件。
- **邮件服务。**提供电子邮件存储转发的基础。
- **目录服务。**提供分布式数据库资源以及对不同对象和服务的全球信息的访问。

2.3.8 各层功能小结

图2.15对每层的功能做了小结。

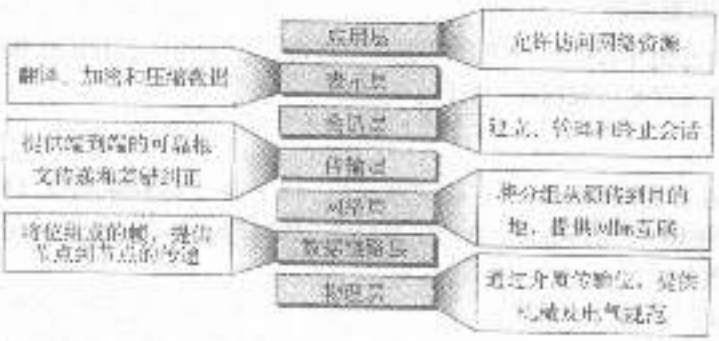


图2.15 各层功能小结

2.4 TCP/IP协议族

TCP/IP协议族（TCP/IP protocol suite）是在OSI模型之前开发的，所以TCP/IP协议族的层次与OSI模型的层次并不严格对应。TCP/IP协议族由4个层次组成：主机到网络层、互联网层、传输层和应用层。然而，当TCP/IP与OSI模型比较时，则可以说主机到网络层相当于物理层和数据链路层的组合。互联网层等价于网络层，应用层所做的工作大致上相当于会话层、表示层和应用层以及TCP/IP传输层处理会话层的部分功能。所以在本书中假定TCP/IP协议族由5层组成：物理层、数据链路层、网络层、传输层和应用层。前面4层提供物理标准、网络接口、网际互联和传输功能，它相应于OSI模型前的4层，OSI模型最上面的3层由TCP/IP一个称为*应用层*（参看图2.16）的单一层来表示。

TCP/IP是由相互作用的模块组成的分层协议，每一层提供一个特定的功能，但它们不需要相互依赖。尽管OSI模型指定了属于每一层的功能，但TCP/IP协议族的各层却包含一些相对独立的协议，这些协议依据系统的需要能混合与匹配。术语分层意指每个上一层协议是由一个或多个下层协议支持的。

在传输层，TCP/IP定义了3个协议：传输控制协议（TCP）、用户数据报协议（UDP）和流控制传输协议（SCTP）。在网络层，由TCP/IP定义的主要协议有网际协议（IP），尽管还有支持数据运行的其他协议。

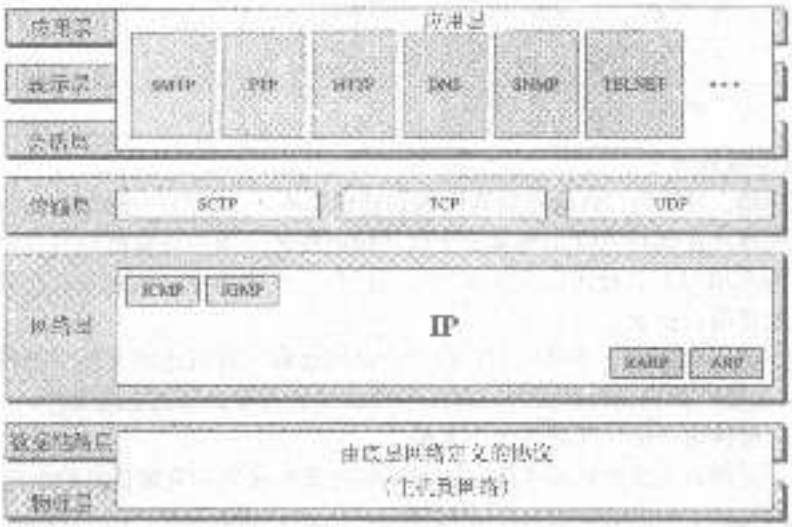


图2.16　TCP/IP和OSI模型

2.4.1　物理层和数据链路层

在物理层和数据链路层中，TCP/IP不定义任何协议。它支持所有标准和专门协议。采用TCP/IP互联的网络可以是局域网或广域网。

2.4.2　网络层

在网络层（或者更确切地说，互联网络层），TCP/IP支持网际协议（Internetworking Protocol, IP），IP使用4个支持协议：ARP，RARP，ICMP和IGMP。这些协议中每一个都将在后面几章更详细地介绍。

网际协议（IP）

网际协议是TCP/IP协议使用的传输机制。它是一个不可靠、无连接数据报协议，即尽力传递（best-effort delivery）服务。尽力一词的意思是IP提供的是没有差错检测和跟踪。IP假定了底层是不可靠的，因此尽最大努力传输到目的地，但没有保证。

IP传输的数据称为数据报，每一个数据报都独立地传输。数据报可以沿着不同的路由传输，有可能不按顺序到达，或者重复出现，IP没有路由跟踪，因此一旦这些数据报到达目的地，它们没有重新排序的功能。

然而，IP这一局限功能不认为是一个缺点，因为它提供这样简单的传输使用户自由添加已知应用所必需的那些功能，使得效率更高。IP将在第20章中讨论。

地址解析协议

地址解析协议（Address Resolution Protocol，ARP）将逻辑地址与物理地址相联系起来。在一个典型的物理网络中，比如说一个局域网，链路上的每一个设备都用物理地址或站点地址来标识。这个地址通常印记在网络接口卡（NIC）上。ARP用于已知节点的因特网地址寻找该节点的物理地址。将在第21章讨论ARP。

逆地址解析协议

逆地址解析协议（Reverse Address Resolution Protocol，RARP）允许主机在仅知道物理地址的情况下寻找因特网地址。如果主机是第一次连接到网络上或启动一台无盘计算机时，就要使用该协议。将在第21章讨论RARP。

因特网控制报文协议

因特网控制报文协议（Internet Control Message Protocol，ICMP）是由主机和网关所使用的一种机制，用来向发送方通知数据报所发生的问题。ICMP在第21章中讨论。

因特网组报文协议

因特网组报文协议（Internet Group Message Protocol，IGMP）用于将一个报文同时发送给一组接收者。我们将在第22章讨论IGMP。

2.4.3 传输层

传输层在TCP/IP中用两个协议表示：TCP和UDP。IP是主机到主机协议（host-to-host protol），这就是说它可以将分组从一个物理设备传递到另一个物理设备。UDP和TCP是负责将报文从一个进程（运行程序）传送到另一个进程的**传输层协议**（transport level protocol）。已设计出一个新的协议STCP满足某些较新的应用需要。

用户数据报协议

用户数据报协议（User Datagram Protocol，UDP）是两个标准TCP/IP传输层协议中较简单一个。它是一个进程到进程的协议，它仅将端口号、校验和差错控制及信息长度添加到来自上一层的数据中。UDP在第23章中讨论。

传输控制协议

传输控制协议（Transmission Control Protocol，TCP）为应用程序提供了完整的传输层服务。TCP是一个可靠的流传输协议，流这个词在这里表示面向连接的：在传输的两端可以传送数据以前必须先建立连接。

在每个传输的发送端，TCP划分数据流为称为片段的更小的数据单元，每个片段都包括一个用来在接收后重新排序的序号，以及对接收到片段的确认序号。片段将包含在IP数据报中，通过互联网传输。在接收端，TCP收集每个到来的数据报，然后基于序号对传输重新排列。TCP在第23章中讨论。

流控制传输协议

流控制传输协议（Stream Control Transmission Protocol，SCTP）对较新应用提供支持，比如因特网上的音频传输。它结合了UDP和TCP最好的特点，将在第23章中讨论。

2.4.4 应用层

在TCP/IP中的应用层相当于OSI模型中的会话层、表示层和应用层的组合。这一层定义了许多协议，在后面几章中，将讨论多个标准协议。

2.5 寻址

采用TCP/IP协议的互联网使用4层地址：**物理（链路）地址**（physical（link）address），**逻辑（IP）地址**（logical（IP）address），**端口地址**（port address）和**专用地址**（见图2.17）。



图2.17 TCP/IP的地址

每种地址都与TCP/IP层次结构中的某个特定层有关，如图2.18所示。

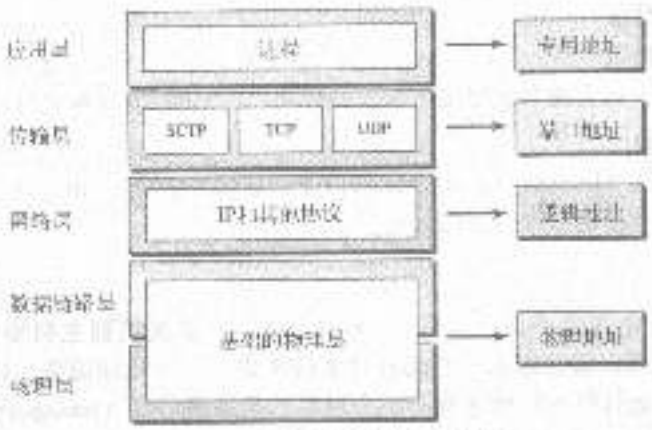


图2.18　TCP/IP的地址与层的关系

2.5.1　物理地址

物理地址，也称为链路地址，它是局域网或广域网定义的节点地址。它包含在数据链路层所用的帧中，是最低级的地址。

物理地址在网络（LAN或者WAN）中使用，这类地址的长度和格式是可变的，它们与网络有关。例如，以太网使用6个字节（48位）物理地址，在网络接口卡（NIC）上标明。然而，LocalTalk（Apple）使用1个字节的动态地址，它每次在站点开始到来时改变。

例2.1

在图2.19中，物理地址为10的节点发送一个帧到物理地址为87的节点。这两个节点通过链路连接（总线拓扑结构的LAN）。在数据链路层，该帧的头部含有物理（链路）地址，它们是需要的地址，头部还含有该层其他必要的信息。尾部通常是为了差错检测需要的附加位。如图2.19所示，具有物理地址为10的计算机是发送方，而具有物理地址为87的计算机是接收方。发送方的数据链路层接收来自上一层的数据，把它封装在一个帧中并添加头部和尾部。其头部包含接收方和发送方的物理（链路）地址，作为信息的一个部分。注意：大多数情况下，数据链路协议的目的地址置于源地址之前，在本例中地址87在地址10之前。已经说明总线拓扑结构为单一的局域网，帧在两个方向传播（左和右），向右传播的帧被送到网络上每一个站。凡是物理地址不是87的站都放弃这个帧。这是由于目的地址与它们的物理地址不相同，而具有目的地址的计算机发现帧中的目的地址与自己相同，就检验该帧，去掉头部和尾部，提取出数据部分向上层传递。



图2.19　物理地址

例2.2

将在第13章看到，大多数局域网使用48位（6个字节）的物理地址，该地址被写作12个十

六进制的数字，每个字节（2个十六进制数）用冒号分隔开，如下所示：

07：01：02：01：2C：4B

6个字节（12个十六进制）物理地址

2.5.2 逻辑地址

逻辑地址对于全世界通信是必需的，它与下面的物理网络无关。物理地址不适用于互联网的环境，其中不同网络可能有不同地址格式。需要有全世界的寻址系统，其中每台主机有唯一的标记，而不管下面的物理网络。

为此，设计了逻辑地址，因特网的逻辑地址现在流行的是32位地址，它唯一定义了连接到因特网的一台主机。因特网上没有两台主机具有相同的IP地址。

例2.3

图2.20表示了用两个路由器连接三个局域网所组成的互联网的一个部分。每台设备（路由器和计算机）都有一对地址（逻辑的和物理的）用作连接。此时，每台计算机仅连接到一条链路，因此它只有一对地址。然而，路由器连接3个网络（图中仅连接2个），因此每一个有3对地址，每个连接用一对。虽然，显然对每一个连接，每个路由器必须有一个独立的物理地址，为什么对每一连接，每个路由器并不需要一个明确的逻辑地址。当讨论路由选择时，这些问题将在第22章中讨论。

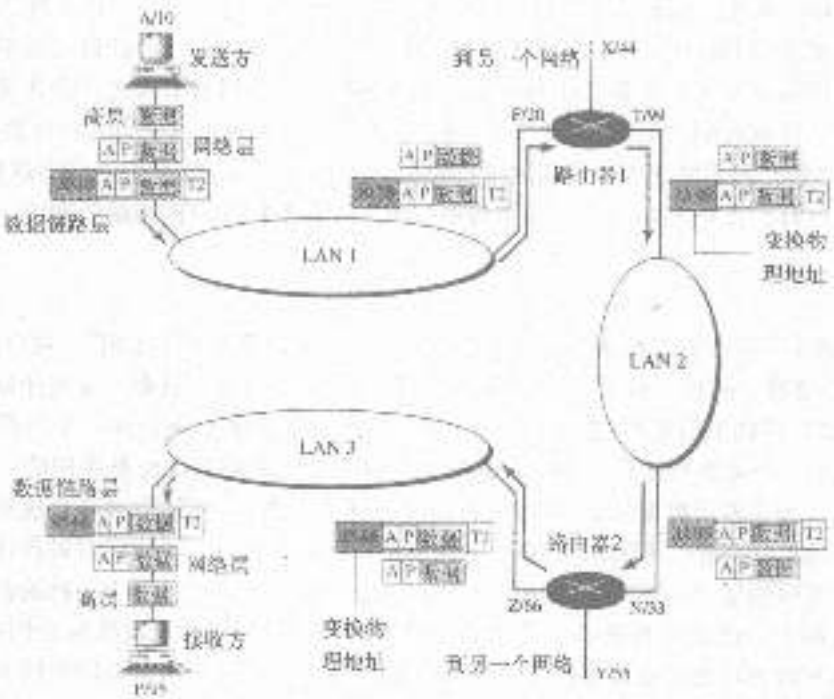


图2.20 IP地址

逻辑地址A和物理地址10的计算机想要发送一个分组到逻辑地址P和物理地址95的计算机，此处用文字表示逻辑地址，用数字表示物理地址，但是需要注意的是这两个地址实际上都是数字，这将在后面的章看到。

发送方在网络层封装这个分组并添加两个逻辑地址（A和P）。注意在多数协议中，逻辑源地址在逻辑目的地址之前（与物理地址刚好相反）。在该分组传递之前，网络层必须要寻找下

一跳的物理地址。网络层查阅它的路由表（见第22章），并找到下一跳的逻辑地址为F（路由器1），前面提到的ARP发现路由器1的物理地址是20，网络层将这个地址传给数据链路层，接着在该层用目的物理地址20和源物理地址10封装该分组为帧。

该帧被网络1中所有设备接收到，但除掉路由器1发现帧中的目的物理地址与它自己的物理地址相同外，其余的设备都废弃该帧。路由器1拆封该帧为分组，并读出逻辑地址P，由于目的逻辑地址与路由器的逻辑地址不相同，路由器知道该分组需要转发。路由器查询它的路由表并应用ARP发现下一跳目的物理地址（路由器2），创建封装该分组的一个新帧发送到路由器2。注意，该帧中的物理地址，源物理地址由20变为99，目的物理地址由20（路由器1的物理地址）变成33（路由器2的物理地址）。源和目的逻辑地址必须保持相同，否则该分组将会被丢失。

路由器2有相似的过程，物理地址将被改变并向目的计算机发送一个新帧。当帧到达目的地，被拆封为分组。目的逻辑地址P与计算机逻辑地址相同，由分组拆封的数据传送给上一层。注意：虽然跳到跳时物理地址改变了，但源到目的逻辑地址保持相同。在本书后面，也有例外的情况。

> 跳到跳时物理地址将改变，但逻辑地址保持不变。

2.5.3　端口地址

IP地址和物理地址对于从源主机到目的主机的数据传输是必需的，但是到达目的主机不是因特网数据通信的最终目标。只将数据从一台计算机传输到另一台计算机而不做其他事情是不完整的传输。今天，计算机能同时运行多个进程，因特网通信最终目的是进程到进程的通信。例如，计算机A使用TELNET与计算机C通信，同时计算机A使用文件传输协议（FTP）与计算机B通信。为了使这些进程同时接收数据，需要有一种方法标记不同的进程。换言之，需要地址标识符。在TCP/IP分层体系结构中，赋于进程的标识符称为端口地址，它的长度是16位。

例2.4

图2.21表示了两台计算机通过因特网进行通信。发送计算机此时使用3个端口地址a、b和c同时运行3个进程，接收计算机用2个端口地址j和k也同时运行2个进程。发送计算机中的进程a需要与接收计算机中的进程j进行通信。注意：虽然两台计算机运行同一个应用程序，比如FTP，但由于一个是客户进程，另一个是服务进程，它们的端口地址是不同的，这些将在第23章中讨论。为了表示数据从进程a传递到进程j而不是进程k，传输层对来自应用层的数据添加两个端口地址（a和j），即源和目的地址封装在分组中。然后，网络层对来自传输层的分组用源和目的逻辑地址（A和P）封装在为一个分组中。最后，该分组用下一跳的源和目的物理地址封装在帧中。此处没有表示物理地址，因为在因特网范围内从跳到跳它们是要改变的。注意：虽然从跳到跳物理地址要改变，但从源到目的逻辑地址和端口地址保持不变。在本书后面，也有一些例外的情况。

> 跳到跳时物理地址要改变，但逻辑地址保持不变。

例2.5

在第23章将会看到，16位端口地址用一个十进制数表示，如下所示：

753
16位端口地址用单个数字表示。

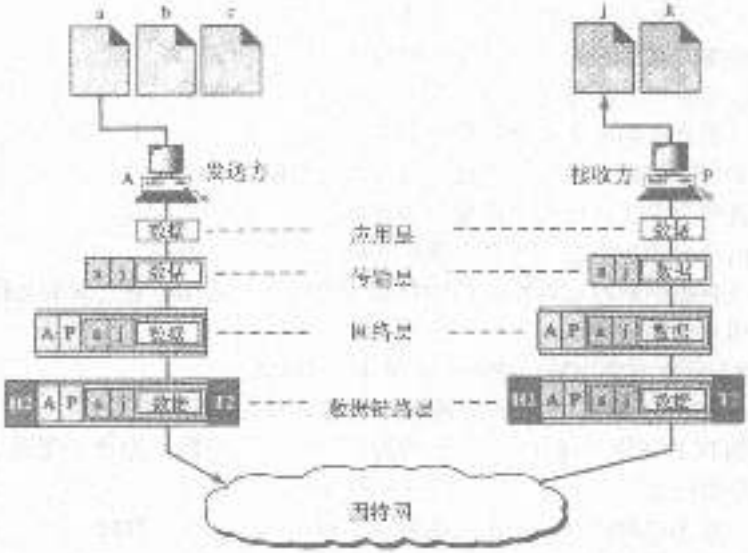


图2.21 端口地址

2.5.4 专用地址

有些面向用户的应用被设计为专用地址。例如，电子邮件的地址（例如forouzan@fhda.edu）和统一资源定位符（URL）（例如www.mhhe.com）。前一个地址定义邮件接发者（见第26章），后一个是在万维网中寻找一个文档（见第27章）。这些地址由发送计算机转换成对应的端口地址和逻辑地址，将在第25章讨论。

推荐读物

为了更详细讨论与本章有关的主题，推荐下列读物与网站，在本书末列出括号[……]内的参考资料。

读物

网络模型的讨论有[Tan03]的1.3节，[For06]的第2章，[Sta04]的第2章，[GW04]的2.2节及2.3节，[PD03]的1.3节和[KR05]的1.7节，有关地址讨论可在[Ste94]中找到。

网站

下面的网站与本章讨论主题有关。

- www.osi.org/ 有关OSI的信息。

请求评论

下面的网站列出有关IP和端口地址的所有RFC。

- www.ietf.org/rfc.html

本章小结

- 国际标准化组织（ISO）建立了一个称为开放系统互联（OSI）模型，允许各种各样的

系统相互通信。

- 七层的OSI模型为开发完全兼容的网络协议提供了指导原则。
- 物理层、数据链路层和网络层都是网络支持层。
- 会话层、表示层和应用层都是用户支持层。
- 传输层将网络支持层和用户支持层连接起来。
- 物理层在通过物理介质发送位流的各个功能之间进行协调。
- 数据链路层负责准确无差错地将数据单元从一个站点传递到下一个站点。
- 网络层负责将分组通过多个网络链路进行源到目的传递。
- 传输层负责整个报文从进程到进程的传递。
- 会话层在两个通信设备之间建立、维护和同步交互的过程。
- 表示层通过将数据转换成互相认同的格式，确保两个通信设备之间的互操作性。
- 应用层使用户能访问网络。
- TCP/IP是5个层次的协议族，它的开发早于OSI模型。
- TCP/IP应用层相当于OSI模型的会话层、表示层和应用层的组合。
- 互联网中的TCP/IP协议使用下列4层的地址：物理（链路）地址，逻辑（IP）地址，端口地址和专用地址。
- 物理地址也称为逻辑地址，是由局域网或广域网定义的节点地址。
- IP地址定义准一标识了互特网上的主机的地址。
- 端口地址是标识一台主机的进程。
- 专用地址是一个面向用户的地址。

习题

复习题

1. 列出因特网模型的各个层。
2. 因特网模型中哪些层是网络支持层？
3. 因特网模型中哪个层是用户支持层？
4. 网络层之间传递与传输层之间的传递有什么不同？
5. 什么是对等进程？
6. 在因特网模型中信息如何从一层传到下一层？
7. 什么是头部和尾部以及它们是如何被添加和删除？
8. 因特网模型的物理层主要与什么相关？
9. 因特网模型的数据链路层的职能是什么？
10. 因特网模型的网络层的职能是什么？
11. 因特网模型的传输层的职能是什么？
12. 端口地址、逻辑地址和物理地址之间有什么不同？
13. 试说出因特网模型中应用层提供哪些服务。
14. 因特网模型中各层是如何与OSI模型相对应？

练习题

15. OSI与ISO之间有什么关系？
16. 将下列各项与OSI模型的一层或多层对应起来。

a. 路由决策　　b. 流量控制

c. 传输介质接口　　d. 为终端用户提供网络访问

17. 将下列各项与OSI模型的一层或多层对应起来。

a. 可靠的进程到进程的报文传递　　b. 路径选择

c. 定义帧　　d. 提供诸如电子邮件和文件传输等用户服务

e. 通过物理介质传输位流

18. 将下列各项与OSI模型的一层或多层对应起来。

a. 直接与用户的应用程序通信　　b. 纠错与重传

c. 机械的，电气的和功能的接口　　d. 负责相邻两节点之间帧的传输

19. 将下列各项与OSI模型的一层或多层对应起来。

a. 格式和代码转换服务　　b. 建立、管理和终止会话

c. 确保可靠的数据传输　　d. 进入和退出系统

e. 提供不同数据表示的独立性

20. 在图2.22中，计算机A经过LAN1、路由器R1和LAN2，发送一个报文到计算机D，分别表明数据链路层和网络层每一跳接口的帧和分组的内容。

21. 在图2.22中，假定通信是在具有端口地址i的计算机A上的进程与具有端口地址j的计算机D上的进程之间进行的，分别表明数据链路层、网络层和传输层每一跳的帧和分组的内容。

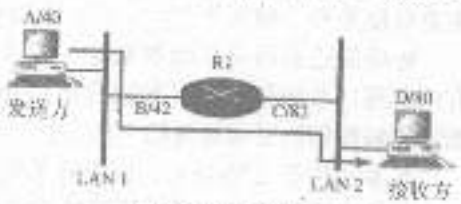


图2.22 练习题20

22. 假定一台计算机发送一个帧到总线拓扑结构局域网上的一台计算机，在传输过程中帧的目的物理地址受损，那么该帧将会发生什么？有关情况如何通告给发送方？

23. 假定一台计算机在网络层发送一个分组到因特网某处的另一台计算机，分组中的目的逻辑地址受损，那么该分组将会发生什么？有关情况如何通告给发送方？

24. 假定一台计算机在传输层发送一个分组到因特网某处的另一台计算机，在目的计算机中运行的目的端口地址没有进程，将会发生什么？

25. 如果在数据链路层各跳之间能直接检错，为什么你认为在传输层需要另一个检测机制？

探究题

26. 因特网模型将OSI模型的会话层、表示层和应用层结合在一起成为的应用层有什么优、缺点。

27. 对话和同步是OSI模型中会话层的两个任务，你认为在因特网模型中这些任务由什么层来负责？

28. 翻译、加密和数据压缩是OSI模型表示层的任务，你认为在因特网模型中这些任务由什么层来负责？

29. OSI模型提出了传输层的几个模式，找出这些模式并解释它们的不同之处。

30. OSI模型提出了网络层的几个模式，找出这些模式并解释它们的不同之处。

第二部分　物理层和介质

目标

我们以最底层——物理层开始因特网模型的讨论。实际上，物理层与传输介质交互，它属于网络中的物理部分，将各网络构件连接在一起。这一层涉及到物理上如何将信息从网络中的一个节点传送到下一个节点。

物理层要执行一些复杂的任务。其中一个主要任务是为数据链路层提供服务，数据链路层中的数据由组织成帧的0和1组成，这些帧准备通过传输介质发送。0和1组成的数据流，首先要转换为另一种实体，即信号，物理层提供的服务之一就是生成表示位流的信号。

物理层还必须管理物理网络，即传输介质。传输介质是一种被动的实体，不像其他层具有内部程序和控制逻辑。传输介质由物理层控制，物理层确定数据的流向，也确定来自不同源端传输数据的逻辑通道的个数。

在本书的第二部分中，讨论有关物理层和它所控制的传输介质的问题。这个部分的最后一章讨论电话网络和有线电视网络的物理层及它们的结构。

本书第二部分专注于物理层和传输介质。

第3章 数据和信号

物理层的一个主要功能是通过传输介质以电磁信号的形式传输数据。无论是从另一台计算机中收集统计数据，还是从一台设计工作站发送动画图片，或者是使远程控制中心响铃，都需要通过网络连接传输数据（data）。

一般情况下，对于个人或者应用程序有用的数据并非都是能够通过网络进行传输。例如，一张照片必须先转换为传输介质可以接受的形式。传输介质通过沿着物理路径传导能量的方式进行工作。

> 数据要进行传输，必须将其转换为电磁信号。

3.1 模拟与数字

数据以及表示数据的信号可以使用模拟（analog）或者数字（digital）的形式。

3.1.1 模拟数据与数字数据

数据可分为模拟数据和数字数据。**模拟数据**（analog data）是指连续状态的信息。**数字数据**（digital data）是指离散状态的信息。例如，一个模拟时钟有时针、分针和秒针，它们以连续的方式给出时间信息，针的移动是连续的。另一方面，报告小时和分钟的数字时钟会突然从8：05变成8：06，它们以离散方式给出时间信息。

模拟数据（比如人类语音产生的声音）采用连续值。当人说话时，会在空气中形成模拟形式的声波，能通过麦克风采集到这个声波并转换成模拟信号，或者对这个声波进行采样转换成数字信号。

数字数据采用离散值。例如，数据以0和1的形式存储在计算机内存中。它们可以转换成数字信号或者调制成模拟信号通过介质进行传输。

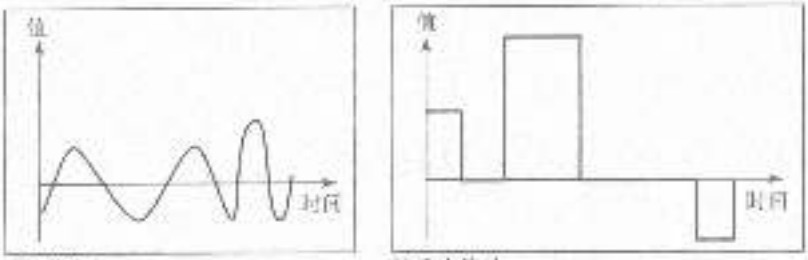
> 数据可以是模拟也可以是数字的。模拟数据是连续的，它采用连续值。数字数据有离散状态，它采用离散值。

3.1.2 模拟信号与数字信号

就像它们所表示的数据一样，信号（signal）可以是模拟信号也可以是数字信号。**模拟信号**（analog signal）在一段时间内有无穷多个强度等级。当波形振幅由值A变为值B时，沿着它变化的过程有无穷多个值。另一方面，**数字信号**（digital signal）只能有有限个已定义的数值，通常情况下是简单的0和1。

表示信号的最简单方法是将它们绘制在直角坐标系中。垂直坐标轴表示信号的值或者强度，水平坐标轴表示时间。图3.1中说明了一个模拟信号和一个数字信号。表示模拟信号的曲线通过了无限多个数值点。然而，数字信号的垂直线表示了信号值到值的突变。

> 信号可以是模拟的也可以是数字的。模拟信号在一个范围内可以有无穷多个数值，而数字信号只能有有限个数值。



a) 模拟信号　　b) 数字信号

图3.1　模拟与数字信号的比较

3.1.3　周期信号与非周期信号

模拟与数字信号都可能以两种形式之一出现：周期信号或者非周期信号。

周期信号（periodic signal）在一个可测量的时间范围内（称为一个周期（period））完成一种模式，并且在后续的相同时间范围内重复这一模式。一个完整模式的实现称为一个循环（cycle）。而**非周期信号**（nonperiodic signal）的变化则不会随着时间的变化出现重复的模式或循环。

模拟和数字信号可以是周期的或者非周期的。然而，在数据通信中，通常使用周期模拟信号和非周期数字信号，因为周期模拟信号需要更少的带宽（我们会在第5章看到），而非周期数字信号可以表示数据的变化（我们会在第6章看到）。

> 在数据通信中，通常使用周期模拟信号和非周期数字信号。

3.2　周期模拟信号

周期模拟信号可以分为简单类型或复合类型两种。简单类型模拟信号，即正弦波（sine wave），不能再分解为更简单的信号。而复合型模拟信号则是由多个正弦波信号组成的。

3.2.1　正弦波

正弦波是周期模拟信号的最基本形式。可以看做一条简单的振荡曲线。在一个周期内的变化是平滑、一致的、连续的、起伏的曲线。图3.2表示的就是一条正弦波形。每个循环由时间轴上方的单弧和后跟着的时间轴下方的单弧构成。

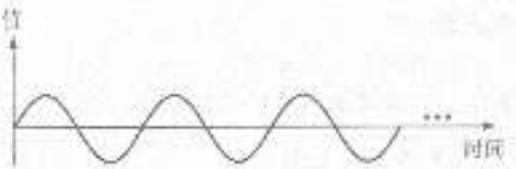


图3.2　正弦波

> 我们会在附录C中讨论正弦波的数学方法。

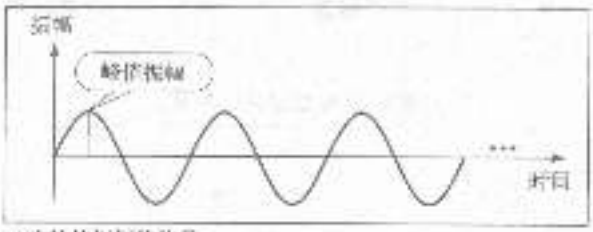
单个正弦波可用三个参数表示：峰值振幅、频率和相位。这三个参数完全决定正弦波。

峰值振幅

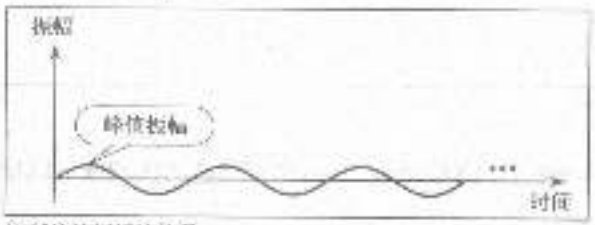
信号的**峰值振幅**（peak amplitude）是其最高强度的绝对值，与其携带的能量成比例。对于电信号，峰值振幅通常以伏特为单位来计量。图3.3表示了两个信号和它们的峰值振幅。

例3.1

家用电源的功率可以用峰值振幅为155V到170V的正弦波表示。但是，大家都知道美国家用的电压是110V到120V。不一样的原因在于这些均方根（rms）值。信号平方后再计算平均振幅。峰值等于$2^{1/2} \times$ rms。



a) 有峰值振幅的信号



b) 有峰值振幅的信号

图3.3 相位和频率相同但振幅不同的两个信号

例3.2

电池电压是恒定的，这个恒定值可以看做是一个正弦波，我们会在后面看到。例如AA电池的峰值通常是1.5V。

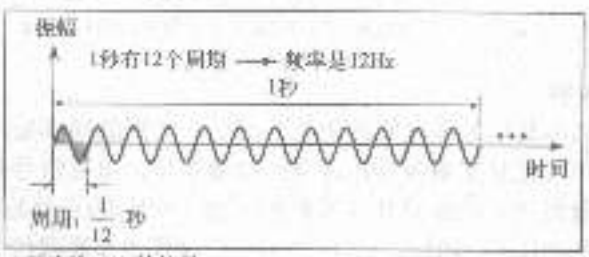
周期与频率

周期（Period）是信号完成一个循环所需要的时间，以秒为单位。**频率**（frequency）是指1秒内的周期数。注意周期与频率是按两种方式定义的同一特性。周期是频率的倒数，频率是周期的倒数，如下列公式所示。

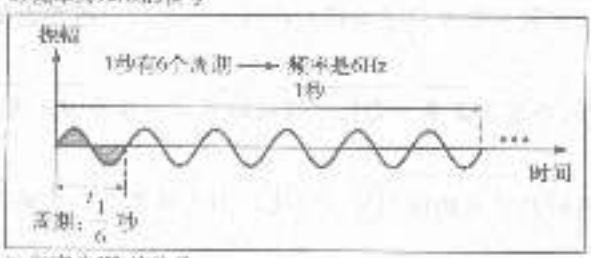
$$f = \frac{1}{T} \text{ 和 } T = \frac{1}{f}$$

频率与周期互为倒数。

图3.4显示了两个信号和它们的频率。



a) 频率为12Hz的信号



b) 频率为6Hz的信号

图3.4 振幅和相位相同但频率不同的两个信号

通常周期用秒（s）来表示，频率用赫兹（Hz）来表示，它是每秒的循环数。周期和频率的单位如表3.1所示。

表3.1　周期和频率的单位

单　位	等价值	单　位	等价值
秒（s）	1s	赫兹（Hz）	1Hz
毫秒（ms）	10^{-3} s	千赫（kHz）	10^{3}Hz
微秒（μs）	10^{-6} s	兆赫（MHz）	10^{6}Hz
纳秒（ns）	10^{-9} s	吉赫（GHz）	10^{9}Hz
皮秒（ps）	10^{-12} s	太赫（THz）	10^{12}Hz

例3.3

家用电的频率是60Hz（欧洲是50Hz）。这个正弦波的周期就可以确定如下：

$$T=\frac{1}{f}=\frac{1}{60}=0.0166\text{s}=0.0166\times10^{3}\text{ms}=16.6\text{ms}$$

这就是说，我们家用电的周期是0.0116s，或16.6ms。我们的眼睛不够敏感，觉察不出振幅的快速变化。

例3.4

用微秒表示周期为100ms。

解：

从表3.1中可以找到1ms的等价值（$1\text{ms}=10^{-3}\text{s}$）和1s的等价值（$1\text{s}=10^{6}\mu\text{s}$），做以下替换：

$$100\text{ms}=100\times10^{-3}\text{s}=100\times10^{-1}\times10^{6}\mu\text{s}=10^{2}\times10^{-1}\times10^{6}\mu\text{s}=10^{5}\mu\text{s}$$

例3.5

一个信号的周期是100ms，那么它以千赫为单位表示的频率是多少？

解：

先把100ms表示为秒，然后从周期计算出频率（$1\text{Hz}=10^{-3}\text{kHz}$）

$$100\text{ms}=100\times10^{-3}\text{s}=10^{-1}\text{s}$$

$$f=\frac{1}{T}=\frac{1}{10^{-1}}\text{Hz}=10\text{Hz}=10\times10^{-3}\text{kHz}=10^{-2}\text{kHz}$$

关于频率的其他内容

我们已经知道，频率是信号与时间的关系，而一个波形的频率是一秒钟内完成的周期的次数，但从另一种角度可把频率看做变化速率的计量单位。电磁信号是振荡的波形，也就是说，它们会在平均能量级上下持续而且可预测地波动。40Hz的信号是80Hz信号频率的一半，完成一个循环的时间是80Hz信号的两倍。所以每一个循环也需要两倍长的时间从最低电压变化到最高电压。所以，尽管将频率描述为每秒内的循环次数，通常也用于计量信号对于时间的变化速率。

> 频率是相应于时间的变化速率。变化占用的时间短意味着频率高；变化占用的时间长意味着频率低。

如果信号值在非常短的时间跨度内发生变化，则其频率高；如果在较长的时间跨度内发生变化，则其频率低。

两个极端

若信号持续不变化会怎样呢？若信号在整个存在的时间内始终保持稳定的电平值会怎样呢？在这种情况下，频率值为0。从概念上来讲，这个想法很简单。如果信号始终没有变化，则它永远不能完成一个循环，所以它的频率为0。

如果信号瞬间发生变化会怎么样呢？如果信号瞬间由一个电平立即跳变到另一电平情况又会怎么样呢？在这种情况下，其频率值是无穷大，换句话说，当信号瞬间发生变化时，它的周期是0，既然频率是周期的倒数，当面频率等于1/0，或者是无穷大（无边界）。

> 如果信号始终不变化，则其频率是0；如果信号瞬间发生变化，则其频率值是无穷大。

3.2.2 相位

术语相位（phase）是指波形相对于时间零的位置。如果我们将波形想象为能够沿着时间轴向后或者向前平移，则相位描述的就是这种偏移的数量。它指明了第一个循环的状态。

> 相位描述了波形相对于时间零的位置。

相位使用角度或者弧度进行计量[360°等于2π弧度；1°等于2π/360弧度，1弧度等于360/(2π)]。一个360°的相位偏移（相移）相当于一个完整周期的偏移，相移180°相当于半个周期的偏移，相移90°相当于四分之一周期的偏移（见图3.5）。

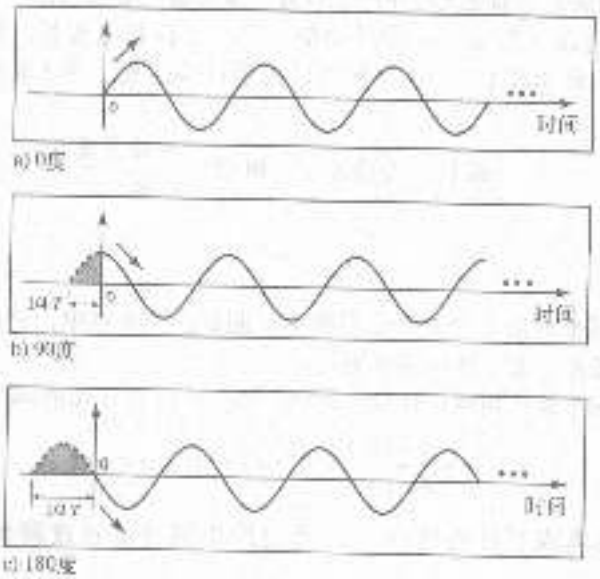


图3.5 振幅和频率相同但相位不同的三条正弦波

从图3.5中，我们可以看出：

1. 相位为0°的正弦波在0时刻以零振幅开始，然后振幅递增。
2. 相位为90°的正弦波在0时刻以峰值振幅开始，然后振幅递减。
3. 相位为180°的正弦波在0时刻以零振幅开始，然后振幅递减。

另一种看相位的方式是位移或偏移。我们可以看出：

1. 相位为0°的正弦波没有位移。
2. 相位为90°的正弦波向左位移1/4个周期。但是，注意在0时刻前信号并不真的存在。
3. 相位为180°的正弦波向左位移1/2个周期。但是，注意在0时刻前信号并不真的存在。

例3.6

正弦波相对于0时刻的偏移量是1/6周期，相位的角度值和弧度值是多少？

解：

一个完整的周期是360°，所以1/6周期是：

$$\frac{1}{6}\times 360 = 60° = 60\times\frac{2\pi}{360}\text{rad} = \frac{\pi}{3}\text{rad} = 1.046\text{rad}$$

3.2.3 波长

波长（wavelength）是信号通过传输介质传播的另一个特性，波长将简单正弦波的周期或频率与介质的**传播速度**（propagation speed）结合在一起（见图3.6）。

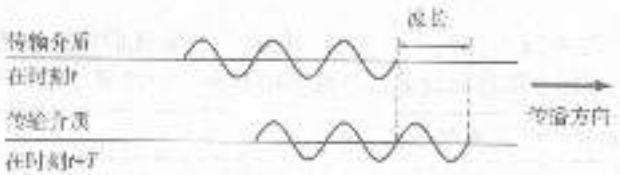


图3.6 波长和周期

信号的频率与介质无关，但波长取决于频率和介质，波长是任何信号的一个属性。在数据通信中，我们一般使用波长来描述光纤中的光传输。波长是简单信号在一个周期能传播的距离。

如果给定了传播速度（光速）和信号周期，就可以计算出波长。但是，因为周期和频率相关，如果我们用λ表示波长，用c（光速）表示传播速度，用f表示频率，那么我们可以得到：

$$\text{波长} = \text{传播速度}\times\text{周期} = \frac{\text{传播速度}}{\text{频率}}$$

$$\lambda = \frac{c}{f}$$

电磁信号的传播速度取决于介质和信号频率。例如，在真空中，光以3×10^8m/s的速度传播。在空气中速度会低些，而电缆中速度更低。

波长通常以微米而不是米为单位计量。例如，空气中红光（频率=4×10^{14}）的波长是：

$$\lambda = \frac{c}{f} = \frac{3\times10^8}{4\times10^{14}} = 0.75\times10^{-6}\text{m} = 0.75\mu\text{m}$$

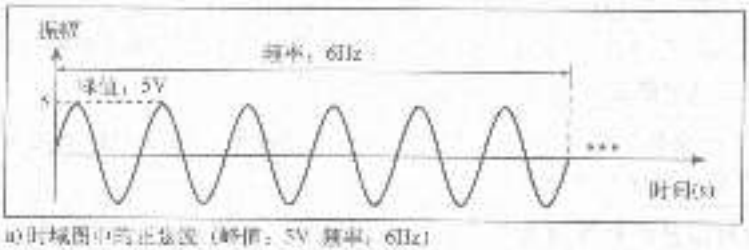
但是，在同轴电缆或光纤电缆中，因为电缆中的传播速度降低了，所以波长更短（0.5 μm）。

3.2.4 时域和频域

一个正弦波可以通过它的振幅、频率和相位得到完整的定义。前面已经使用时域图（time-domain plot）来表示正弦波。时域图显示了信号振幅随时间的变化情况（振幅的时间图）。在时域图中，相位并未显式地表示出来。

为了表示振幅与频率的关系，可以使用频域图（frequency-domain plot）。频域图只和峰值和频率相关，不显示一个周期中的振幅变化。图3.7显示了信号的时域图和频域图。

很明显频域图更容易画，并且传递了时域图中能找到的信息。频域的优点是我们能直接看到频率值和峰值振幅值。完整的正弦波通过一个尖峰表示，尖峰的位置表示了频率，尖峰的高度表示了峰值振幅。



a) 时域图中的正弦波（峰值：5V 频率：6Hz）



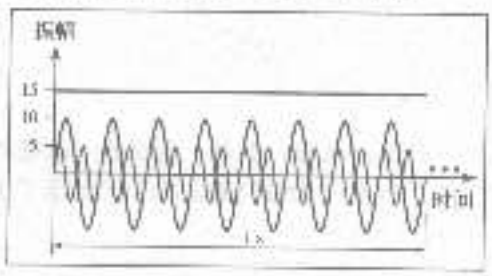
b) 频域图中的相同正弦波（峰值：5V 频率：6Hz）

图3.7 正弦波的时域图和频域图

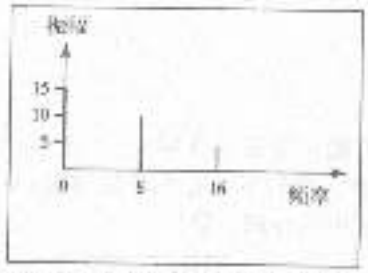
用频域图中单个峰值可表示时域图中一个完整正弦波。

例3.7

当我们处理多个正弦波时，频域更简洁更有用。例如，图3.8显示了三个不同振幅和频率的正弦波。它们通过频域中的三个尖峰表示。



a) 三个频率分别为0，8和16的正弦波的时域表示



b) 三个相同信号的正弦波的频域表示

图3.8 三个正弦波的时域和频域

3.2.5 复合信号

到目前为止，我们主要讨论的是简单正弦波。单个正弦波在日常生活中有很多应用。可以发送单一正弦波将电子能量从一个位置传送到另一个位置。例如，电力公司通过发送频率为60Hz的单一正弦波将电能分送到我们的住所和商业部门。另一个例子是在夜贼打开我们住所的房门或者窗户时，我们可以使用单一正弦波向安全中心发送警报。第一种情况下，正弦波是运送能量；第二种情况下，正弦波是信号。

如果使用单一正弦波来传输电话的谈话，就毫无意义，因为它不会携带任何信息，我们只会听到嘀嗒声。我们将在第4章和第5章中看到，我们需要复合信号来进行数据通信。复合信号（composite signal）由许多简单正弦波构成。

单一频率的正弦波在数据通信中没有用处，我们需要发送复合信号，复合信号由许多简单正弦波组成。

在20世纪早期，法国数学家Jean-Baptiste Fourier证明了任何复合信号都是由具有不同频率、相位和振幅的正弦波信号组合而成的。在附录C中讨论傅里叶分析（Fourier analysis），这里我们只给出这个概念。

> 按照傅里叶分析，任何复合信号是由具有不同频率、相位和振幅的简单正弦波的组合而成。在附录C讨论傅里叶分析。

复合信号可以是周期性的也可以是非周期性的。周期复合信号可以分解成一系列具有离散频率（取整数值1、2、3等的频率）的简单正弦波。非周期复合信号可以分解成具有连续频率（取实数值的频率）的无穷简单正弦波的组合。

> 如果复合信号是周期性的，分解得到的是一系列具有离散频率的信号。如果复合信号是非周期性的，分解得到的是具有连续频率的正弦波组合。

例3.8

图3.9显示了频率为f的周期复合信号。这种信号不是数据通信中的典型信号。我们可以把它看成三个告警系统，每一个有不同的频率。这个信号的分析可以让我们理解如何分解信号。

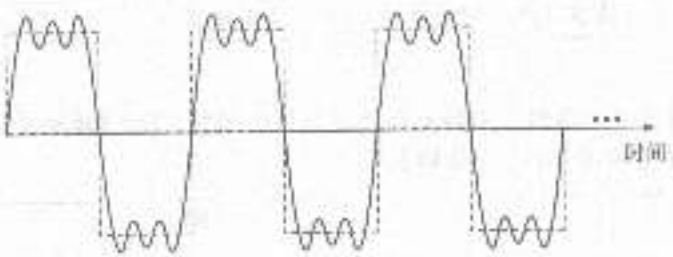
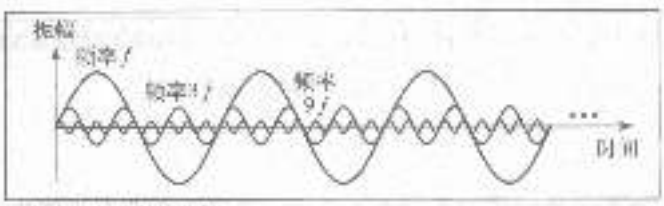
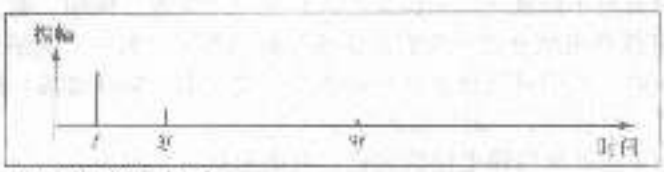


图3.9　复合周期信号

用人工把这个信号分解成一系列简单正弦波是困难的，但是，有工具（软硬件）可以帮助我们做到。我们不关心它是如何做到的，我们只对结果感兴趣。图3.10显示了上面信号在时域和频域中的分解结果。



a) 复合信号的时域分解



b) 复合信号的频率分解

图3.10　复合周期信号在时域和频域中的分解

频率为f的正弦波的振幅几乎和复合信号的峰值一样。频率为$3f$的正弦波的振幅是第一个的三分之一，频率为$9f$的正弦波的振幅是第一个的九分之一。频率为f的正弦波的频率和复合信号的频率一样，称为基础频率（fundamental frequency）或者第一谐波（harmonic）。频

率为3f的正弦波的频率是基础频率的3倍，称为第三谐波。频率为9f的正弦波的频率是基础频率的9倍，称为第九谐波。

注意：分解后信号的频率是离散的，频率为f、3f和9f。因为f是整数，所以9f也是整数，没有诸如1.2f或2.6f的频率。周期复合信号的频域总是由离散尖峰组成。

例3.9

图3.11表示了一个非周期复合信号。它可以在一个或两个词发音时由麦克风或电话机产生。这种情况下，复合信号不可能是周期的，因为我们不可能以完全相同的音调重复相同的词或语句。

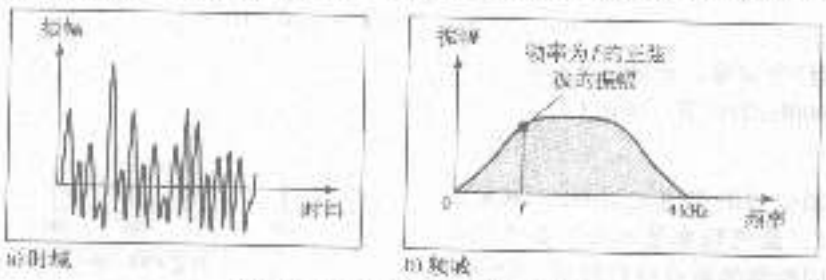


图3.11 非周期信号的时域和频域

在这个复合信号的时域表示中，有无数个简单正弦频率。虽然人类语音中的频率数量是无限的，但是范围是有限制的。普通人会产生0Hz到4kHz之间的频率。

注意：信号的频率分解后得到的是一条连续曲线。在实数区间[0.0，4000.0]内有无数个（实数值）频率。为了找到频率为f的振幅，我们在f处画一条垂直线与曲线相交。垂直线的高度就是相应频率的振幅。

3.2.6 带宽

复合信号包含的频率范围称为带宽（bandwidth）。带宽通常是两个数的差值。例如，如果一个复合信号包含1000到5000的频率，它的带宽就是5000 − 1000（4000）。

复合信号的带宽是信号最高频率与最低频率的差值。

图3.12说明了带宽的概念。图中描述了两个复合信号，一个是周期信号而另一个是非周期

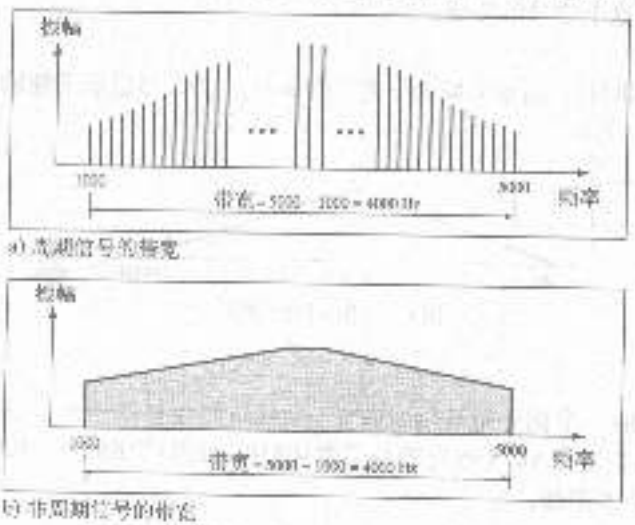


图3.12 周期复合信号和非周期复合信号的带宽

信号。周期信号的带宽包含1000到5000之间（1000、1001、1002、…）的所有整数频率。非周期信号的带宽有相同的范围，但频率是连续的。

例3.10

如果一个周期信号分解为5个正弦波信号，频率分别为100、300、500、700和900Hz，那么其带宽是多少？假定所有分量的最大振幅都为10V，试画出该信号的频谱。

解：

设f_h为最高频率，f_l为最低频率，B为带宽，则：

$$B = f_h - f_l = 900 - 100 = 800 \text{ (Hz)}$$

频谱只有5个尖峰，分别位于100、300、500、700和900Hz的位置（见图3.13）。

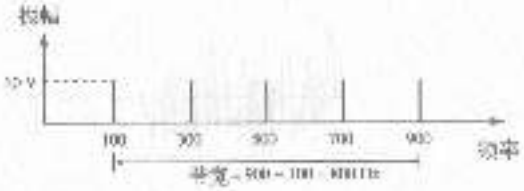


图3.13　例3.10的带宽

例3.11

一个周期信号的带宽是20Hz，其最高频率是60Hz，最低频率是多少？如果信号包含具有相同振幅的所有整数频率，试画出其频谱。

解：

设f_h为最高频率，f_l为最低频率，B为带宽，则：

$$B = f_h - f_l \rightarrow 20 \rightarrow 60 - f_l \Rightarrow f_l = 60 - 20 = 40\text{Hz}$$

频谱包含所有整数频率。用一系列尖峰表示（见图3.14）。



图3.14　例3.11的带宽

例3.12

一个非周期复合信号的带宽为200kHz，中间频率为140kHz，峰值振幅为20V。两个频率极值的振幅为0。画出这个信号的频域。

解：

最低频率一定是40kHz，而最高频率一定是240kHz。图3.15显示了频域和带宽。

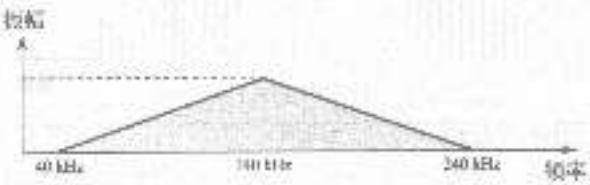


图3.15　例3.12的带宽

例3.13

非周期复合信号的一个例子是调幅（AM）无线电站传播的信号。在美国，分配给每个AM无线电站10kHz的带宽。AM无线电的总带宽从530kHz到1 700kHz。我们会在第5章说明这个10kHz带宽后面的基本原理。

例3.14

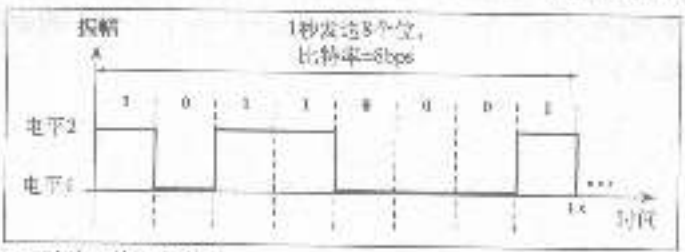
非周期复合信号的另一个例子是调频（FM）无线电站传播的信号。在美国，分配给每个FM无线电站200kHz的带宽。FM无线电的总带宽从88MHz到108MHz。我们会在第5章说明这个200kHz带宽后面的基本原理。

例3.15

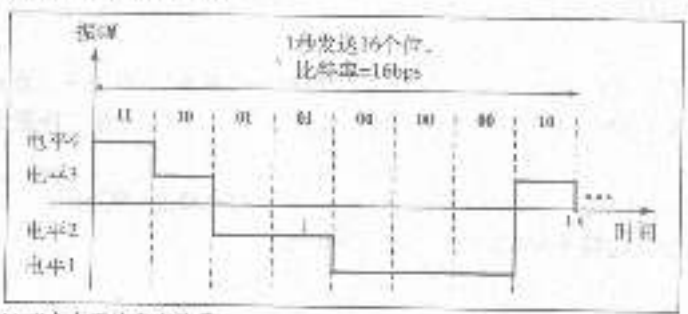
非周期复合信号的另一个例子是老式模拟黑白电视接收的信号。电视屏幕由像素（图像元素）组成，每个像素要么白要么黑。屏幕每秒扫描30次。（实际上每秒扫描60次，但是隔行扫描的，一个回合扫描奇数线，下一个回合扫描偶数线）。如果我们假定分辨率为525×700（525条垂直线，700条水平线），比率为3∶4，那么每个屏幕就会有367 500个像素。如果屏幕每秒扫描30次，每秒就是367 500×30=11 025 000个像素。最坏的情况是黑色像素和白色像素交替。这种情况下，我们需要用最低振幅表示一种颜色而用最高振幅表示另一种颜色。每个周期我们能发送两个像素。因此，每秒我们需要11 025 000/2=5 512 500个周期（或Hz）。所需的带宽是5.5124MHz。最坏情况发生的可能性很小，假定只需要这个带宽的70%，即3.85MHz。因为还需要音频和同步信号，为每个黑白电视频道留出了4MHz带宽。模拟彩色电视频道的带宽为6Mhz。

3.3 数字信号

数据除了可以用模拟信号表示以外，还可以使用数字信号表示。例如，1可以编码为正电平，0可以编码为0电平。一个数字信号可以多于两个电平。在这种情况，每个电平就可以发送多个位。图3.16表示了两个信号，一个信号有两个电平而另一个信号有四个电平。



a) 两个电平的数字信号



b) 四个电平的数字信号

图3.16 两个数字信号：一个有两个电平另一个有四个电平

图3.16a中，每个电平可以发送1位，而图3.16b中每个电平可以发送2位。一般，如果信号有L个电平，则每个电平需要$\log_2 L$个位。

附录C复习有关指数函数和对数函数

例3.16

一个数字信号有8个电平，那么每个电平需要多少个位？我们根据下面公式进行计算：

$$每个电平的位数=\log_2 8=3$$

每个信号电平用3位表示。

例3.17

一个数字信号有9个电平，问每个电平可表示多少位？我们根据公式计算，得出每个信号电平用3.17个位表示。但是，这个答案是不现实的，每个电平发送的位数必须是整数并且是2的幂。这个例子中，4位用来表示一个电平。

3.3.1 比特率

大多数数字信号都是非周期性的，这样周期和频率就不再适用。另一个术语——比特率（而不是频率）用来描述数字信号。比特率（bit rate）是1秒中发送的位数。以每秒位（bits per second，bps）表示。图3.16表示了两个信号的比特率。

例3.18

假定我们需要每分钟100页的速率下载文本文档，所需的通道比特率是多少？

解：

一页平均24行，每一行80个字符，如果我们假定每个字符需要8位，比特率

$$(100 \times 24 \times 80 \times 8)/60=(1\ 636\ 000)/60\text{bps} = 256\text{kbps}$$

例3.19

我们会在第4章中看到，数字化语音通道是通过数字化4kHz带宽的模拟语音信号形成的。我们需要以最高频率的2倍对信号进行采样（即每秒两个样本）。我们假定每个样本需要8位。那么所需的比特率是多少？

解：

比特率计算如下：

$$2 \times 4\ 000 \times 8 = 64\ 000\text{bps} = 64\text{kbps}$$

例3.20

高清晰电视（HDTV）的比特率是多少？

解：

HDTV使用数字信号广播高质量视频信号。HDTV屏幕通常为16:9（普通电视为4:3），表示屏幕宽度。每屏有1 920 × 1 080个像素，每秒刷新30次。24位代表一种颜色像素。我们计算比特率如下：

$$1\ 920 \times 1\ 080 \times 30 \times 24 = 1\ 492\ 992\ 000\text{或}1.5\text{Gbps}$$

电视台通过压缩把比特率降为20Mbps到40Mbps。

3.3.2 位长

我们已经讨论过模拟信号的波长概念：一个周期在传输介质上的距离。我们可以为数字信号定义相似的概念：位长。位长（bit length）是一个位在传输介质上的距离。

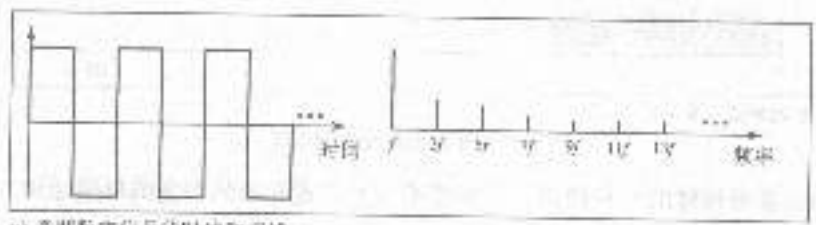
$$位长=传播速度 \times 位持续时间$$

3.3.3 数字信号是一种复合模拟信号

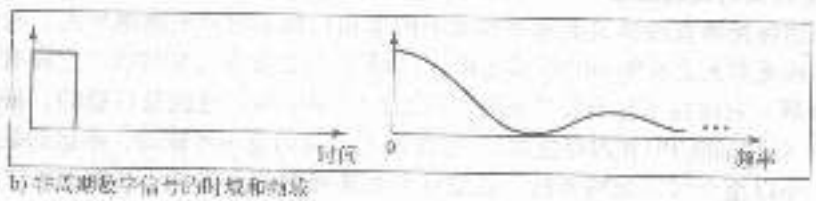
基于傅里叶分析，数字信号是复合模拟信号。正如你猜想的，带宽是无穷大的。当我们

考虑一个数字信号时，可以直观地提出这个概念。在时域中，数字信号由连接的垂直和水平线段组成。时域中的垂直线表示为无穷大的频率（随时间突变），而时域中的水平线表示为0的频率（不随时间变化）。从频率0变为频率无穷大（以及反之）暗示两者间的所有频率都是域的一部分。

傅里叶分析可以用来分解数字信号。如果数字信号是周期性的（在数据通信中较少见），分解后的信号可以表示为无穷大带宽和离散频率的频域。如果数字信号是非周期性的，分解后的信号仍然是无穷大带宽，但频率是连续的。图3.17显示了周期数字信号和非周期数字信号以及它们的带宽。



a) 周期数字信号的时域和频域



b) 非周期数字信号的时域和频域

图3.17 周期数字信号和非周期数字信号的时域和频域

注意：两个带宽都是无穷大的，但是周期信号有离散频率而非周期信号有连续频率。

3.3.4 数字信号的传输

前面的讨论断言数字信号（周期性或非周期性）是由零到无穷大范围内的频率组合成的复合模拟信号。对于下面的讨论，我们假定是非周期数字信号的情况，类似于我们在数据通信中碰到的信号。基本问题是我们如何从A点发送数字信号到B点？我们使用两种不同的方法传输数字信号：基带传输和宽带传输（使用调制）。

基带传输

基带传输就是通过通道发送数字信号，该信号不转换成模拟信号。图3.18表示了基带（baseband）传输。

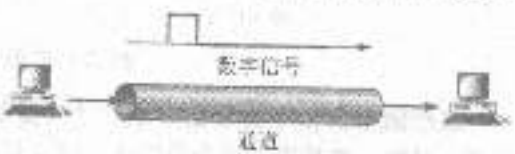
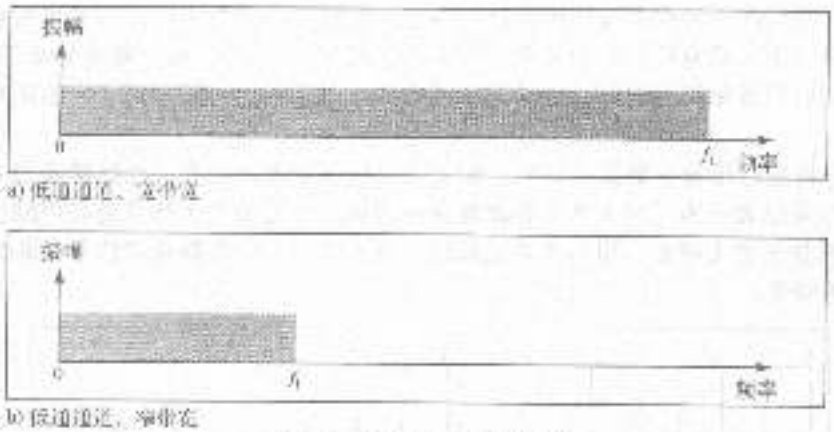


图3.18 基带传输

> 数字信号是无穷大带宽的复合模拟信号。

基带传输需要一个带宽下限频率为0的**低通通道**（low-pass channel），即带宽从0开始的通道。如果我们的带宽只组成一条通道的专用介质，就是这种情况。例如，连接两台计算机的电缆的整个带宽就是一条单通道。另一个例子，我们可以连接多台计算机到一条总线上，但是不允许同一时刻超过两个站进行通信，我们有低通通道，我们就可以用它来进行基带传输。图3.19显示了两条低通通道：一条是窄带宽而另一条是宽带宽。我们需要记住无穷大带宽的低通通道仅是理想情况，我们不可能在现实中有这样的通道。但是，我们可以接近。



a) 低通通道，宽带宽

b) 低通通道，窄带宽

图3.19　两条低通通道的带宽

让我们考察基带传输的两种情况：宽带宽的低通通道和有限带宽的低通通道。

情形1：宽带宽的低通通道

如果我们想保持垂直段垂直而水平段水平的非周期数字信号的准确形式，我们需要发送整个频谱（零和无穷大之间频率的连续范围）。如果我们在发送方和接收方之间有一条无穷大带宽的专用介质，它保持了复合信号的每个成分的准确振幅。这就是可能的。虽然这在计算机里是可能的（比如在CPU和内存之间），但在两个设备间是不可能的。幸运的是，带宽边界的频率振幅太小以至于可以忽略不计。这意味着如果我们的介质具有很宽的带宽（比如同轴电缆或光纤），两个站就可以使用具有非常好准确性的数字信号进行通信，如图3.20所示。注意f_1接近于0而f_2很大。

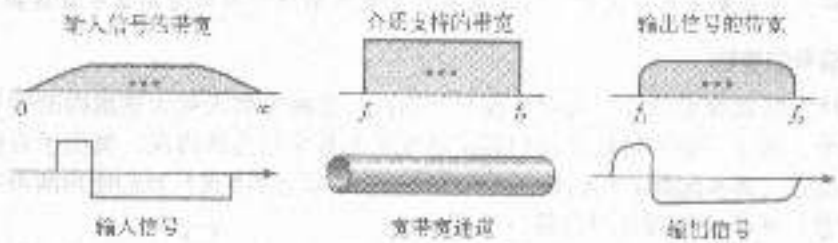


图5.20　使用专用介质的基带传输

虽然输出信号不是原始信号的严格复制品，但是仍然可以从接收到的信号中得出数据。注意，虽然一些频率会被介质阻止，但它们不是主要的。

只有我们有无穷大或非常大带宽的低通通道，保持数字信号形状的数字信号基带传输才是可能的。

例3.21

专用通道（介质的整个带宽用于一条单通道）的一个例子是LAN。几乎现在的每一个有线LAN使用专用通道用于两个站之间的互相通信。在多点连接的总线拓扑结构LAN中，每个时刻只有两个站可以互相通信（时间共享），而其他站限制发送数据。在星型拓扑结构LAN中，每个站和集线器之间的整个通道用于这两个实体间的通信。我们会在第14章学习LAN。

情形2：有限带宽的低通通道

在有限带宽的低通通道中，我们把数字信号近似成模拟信号。近似的程度取决于可用的

带宽。

大致近似 让我们假定我们有比特率为N的数字信号。如果我们想发送模拟信号来大致近似这个信号，我们需要考虑最坏情况，即数字信号中变化的最大数。这发生在信号携带了序列01010101…或者序列10101010…。为了模拟这两种情况，我们需要频率为$f = N/2$的模拟信号。1为正尖峰值，0为负尖峰值。我们每个周期发送2位，那么模拟信号的频率是这个比特率的一半，即$N/2$。但是，只有这个频率不能模拟出所有的模式，我们需要更多的成分。最大的频率是$N/2$。作为这个概念的一个例子，让我们先看看如何使用模拟信号模拟3位模式的数字信号。图3.21说明了这个概念。对两个相似情形（000和111），000用频率$f = 0$与相位为180°的信号模拟，111用频率$f = 0$与相位为0°的信号模拟。对于两个最坏情形（010和101），010用频率$f = N/2$与相位为180°的信号模拟，101用频率$f = N/2$与相位为0°的信号模拟。其他四种情形用$f = N/4$、相位分别为180°、270°、90°和0°的信号模拟。换言之，我们需要能处理频率0、$N/4$和$N/2$的通道。这个大致近似称为第一谐波（$N/2$）频率。所需的带宽是

$$\text{带宽} = \frac{N}{2} - 0 = \frac{N}{2}$$

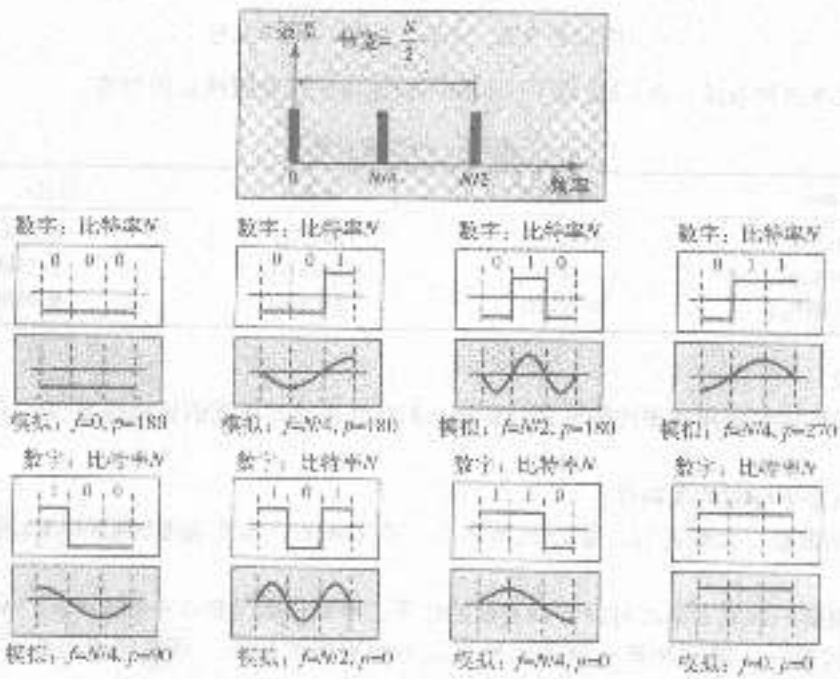


图3.21 对于最坏情形的数字信号使用第一谐波的大致近似

更好近似 为了使得模拟信号的形状更像数字信号，我们需要增加频率的更多谐波，我们需要增加带宽。我们可以增加带宽到$3N/2$、$5N/2$、$7N/2$等。图3.22说明了这种增加对于一种最坏情形（模式010）的影响。

注意我们显示的只是每个谐波的最高频率。我们使用第一、第三和第五谐波。所需的带宽现在是$5N/2$，最低频率0和最高频率$5N/2$之间的差值。正如我们前面强调的，我们需要记住所需带宽与比特率成正比。

> 在基带传输中，所需的带宽与比特率成正比；如果我们需要更快地发送位，我们就需要更大的带宽。

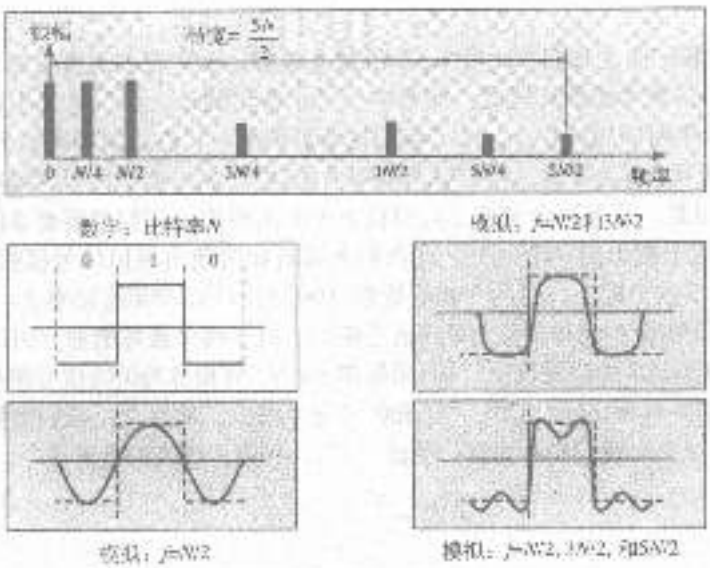


图3.22　使用三个第一谐波模拟数字信号

通过使用这种方法，表3.2显示了我们以不同速率发送数据所需的带宽。

表3.2　带宽需求量

比特率	谐波1	谐波1，3	谐波1，3，5
n=1kbps	B=500Hz	B=1.5kHz	B=2.5kHz
n=10kbps	B=5kHz	B=15kHz	B=25kHz
n=100kbps	B=50kHz	B=150kHz	B=250kHz

例3.22

如果我们需要使用基带传输发送1Mbps，那么低通通道所需的带宽是多少？

解：

答案取决于所需的准确性。

a. 最小带宽、大致近似，是B=比特率/2，即500kHz。我们需要0到500kHz范围频率的低通通道。

b. 使用第一和第二谐波可以得到更好的结果，所需带宽是$B=3\times500\text{kHz}=1.5\text{MHz}$。

c. 使用第一、第三和第五谐波仍然可以得到更好的结果，所需带宽是$B=5\times500\text{kHz}=2.5\text{MHz}$。

例3.23

我们有一条带宽为100kHz的低通通道，那么这条通道的最大比特率是多少？

解：

如果我们使用第一谐波可以得到最大比特率。比特率是2倍有效带宽，即200kbps。

宽带传输（使用调制）

宽带传输或调制就是把数字信号转换成模拟信号传输。调制允许我们使用带通通道（bandpass channel），即带宽不从0开始的通道。这种通道比低通通道更可用。图3.23显示了带通通道。



图3.23 带通通道的带宽

注意：低通通道可以看做是以低频率从0开始的带通通道。

图3.24说明了数字信号的调制过程。在该图中，数字信号转换成复合模拟信号。我们使用单频率模拟信号（称为载波），已经改变的载波振幅使其看起来像数字信号。但是，结果不是单频率信号，它是复合信号，正如我们将在第5章看到的。在接收方，接收到的模拟信号转换成数字信号，结果是被发送信号的复制品。

如果可用通道是带通通道，我们不能直接发送数字信号到通道；我们需要在传输前把数字信号转换成模拟信号。

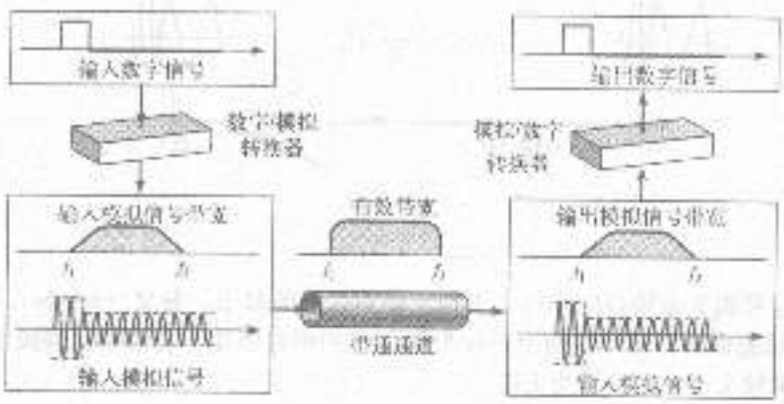


图3.24 数字信号在带通通道传输的调制过程

例3.24

使用调制的宽带传输的一个例子是通过电话用户线传输计算机数据。电话用户线连接住所到中心电话局。这些线路在许多年前就安装了，设计用有限带宽（0到4kHz间的频率）承载语音（模拟信号）。虽然这个通道能用做低通通道，但是通常被看做带通通道。一个原因是带宽太小（4kHz），以至于如果我们把它当做低通通道用于基带传输，那么最大的比特率只有8kbps。解决方案是把它看做带通通道，从计算机把数字信号转换成模拟信号，然后发送模拟信号。我们要安装两个转换器用来在发送端把数字信号转换成模拟信号以及在接收端把模拟信号转换回数字信号。在这种情况下，转换器称为调制解调器（调制器/解调器）。我们会在第5章更详细地讨论调制解调器。

例3.25

第二个例子是数字蜂窝电话。为了更好地接收，数字蜂窝电话把模拟信号转换成数字信号（见第16章）。虽然分配给公司提供数字蜂窝电话服务的带宽很广，但是我们不能不经转换发送数字信号，原因是，我们在主叫方和被叫方之间只有一条可用的带通通道。例如，如果有效带宽是W，并且我们允许1 000对用户同时交谈，这意味着有效带宽是W/1 000，只是整个带宽的一部分。我们需要在发送前把数字信号转换成复合模拟信号。数字蜂窝电话把模拟音频信号转换成数字信号，然后再把数字信号转换成模拟信号通过带通通道传输。

3.4　传输减损

信号通过介质进行传输，但是其传输并非是完美无缺的，不完美的地方导致了信号减损。这意味着信号在介质的开始一端和结束一端是不相同的。发送的信号并非就是接收到的信号。通常会发生三种类型的减损：衰减、失真和噪声（见图3.25）。



图3.25　减损的原因

3.4.1　衰减

衰减（attenuation）意味着能量的损失。当某种简单或者复合的信号通过某种介质传输时，它会失去一些能量用于克服介质的阻抗。这就是为什么传输电信号的电缆会变热的原因，如果不热，过一段时间也会发热。信号中的一些电能转换为热能。为了补偿能量损失，用放大器放大信号。图3.26说明了衰减与放大的效果。

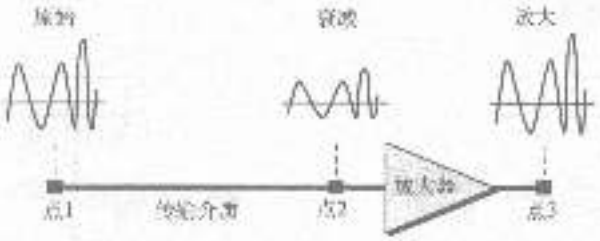


图3.26　衰减

分贝

为了说明信号损失或增益的强度，工程上使用分贝的概念。分贝（decibel，dB）用于计算两种信号之间或者同一信号在两个不同位置之间的相对强度。若信号被衰减了，则分贝为负值；若信号被放大了，则分贝为正值。

$$\mathrm{dB} = 10\log_{10}\frac{P_2}{P_1}$$

P_1和P_2分别是信号在位置1和位置2的功率。注意一些工程书籍中以电压而不是功率定义分贝。这种情况下，因为功率与电压的平方成正比，所以公式变为$\mathrm{dB}=20\log_{10}(V_2/V_1)$。本书中我们以功率表示分贝。

例3.26

假设信号通过一种传输介质传输后，它的功率降低了一半。这可以表示为$P_2 = 1/2\ P_1$，这种情况下衰减（损失的能量）可以计算为：

$$10\log_{10}\frac{P_2}{P_1} = 10\log_{10}\frac{0.5P_1}{P_1} = 10\log_{10}0.5 = 10(-0.3) = -3\mathrm{dB}$$

−3dB或者3dB衰减等价于功率损失了一半。

例3.27

假定信号通过一个放大器，它的功率增大为原来的10倍。这表示$P_2 = 10P_1$。在这种情况下，放大量（功率增益）可以计算为：

$$10\log_{10}\frac{P_2}{P_1} = 10\log_{10}\frac{10P_1}{P_1} = 10\log_{10}10 = 10(1) = 10\mathrm{dB}$$

例3.28

工程人员使用分贝来计量信号强度变化的原因之一是，当涉及计量多处（级联）而不仅仅是两处的信号强度时，分贝数可以相加（或相减）。在图3.27中，一个信号从位置1传输到位置4。信号在到达位置2时衰减了，在位置2与位置3之间，信号被放大了。在位置3与位置4之间信号再次衰减。只需要将各位置之间的分贝值相加，就可以得到信号最终的分贝数。

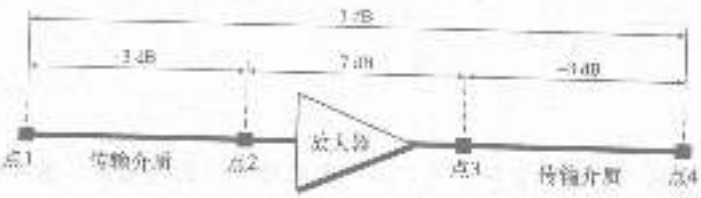


图3.27 例3.28的分贝

在这种情况下，分贝可以计算为：

$$dB = -3 + 7 - 5 = +1$$

表示信号的功率增大了。

例3.29

有时候分贝用来以毫瓦计量信号功率。这种情况下，它成为dB_m，计算为$dB_m=10\log_{10}P_m$，这里P_m是以毫瓦为单位的功率。如果信号的功率为$dB_m=-30$，计算它的功率。

解：

我们计算信号的功率如下：

$$dB_m=10\log_{10}P_m=-30$$
$$\log_{10}P_m=-3 \quad P_m=10^{-3}\text{mW}$$

例3.30

电缆中的损耗一般定义为每公里分贝数（dB/km）。如果电缆每公里分贝数为−0.3dB/km，信号在电缆开始端的功率为2mW，则在5km处信号的功率是多少？

解：

电缆中的损耗为$5\times(-0.3)=-1.5$dB。我们计算功率如下：

$$dB = 10\log_{10}\frac{P_2}{P_1} = -1.5$$

$$\frac{P_2}{P_1} = 10^{-1.5/10} = 0.71$$

$$P_2 = 0.71P_1 = 0.7\times 2 = 1.4\text{mW}$$

3.4.2 失真

失真（distortion）意味着信号改变了形态或形状。失真产生在由不同频率成分组成的复合信号当中。每一种信号成分在通过介质时有自己的传播速度（见3.5节），所以到达最终目的节点时有各自的延迟。如果延迟与周期时间不完全一致，那么延迟的差异就会产生相位的差异。换言之，接收方的信号成分与发送方的信号成分存在相位差异。因此复合信号的形状会不一样。图3.28说明了复合信号中失真的影响。

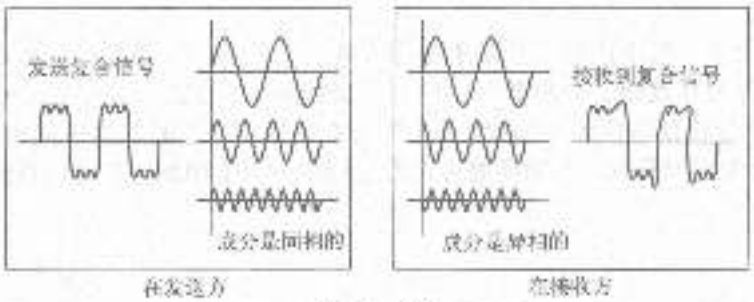


图3.28　失真

3.4.3　噪声

噪声（noise）是另一个减损原因。有几种类型的噪声，如热噪声、感应噪声、串扰和脉冲噪声，都会损害信号。热噪声是电缆中的电子随机移动而产生的额外信号，而不是信号发送装置发送的。感应噪声的来源是发动机和设备。这些设备工作时相当于发射天线，而传输介质成为接收天线。串扰则是电缆之间的互相影响，一根电缆作为发射天线而另一根电缆作为接收天线。脉冲噪声是一种尖峰信号（在非常短时间内有很高能量的一种信号），来自输电线、闪电等。图3.29说明了信号中噪声的影响。我们会在第10章讨论差错。

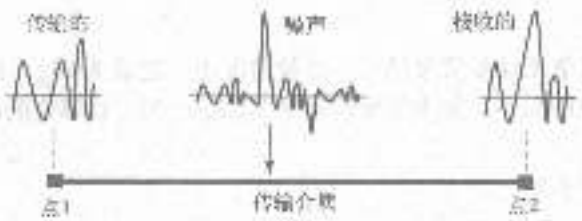


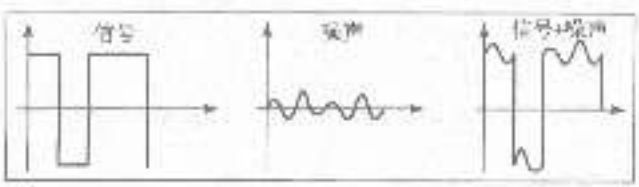
图3.29　噪声

信噪比（SNR）

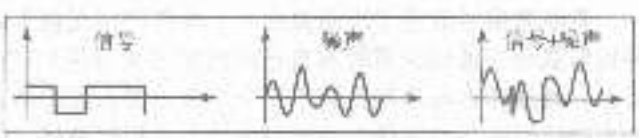
我们稍后会看到，为了找到理论上的比特率限制，我们需要知道信号功率与噪声功率的比率。**信噪比**（signal-to-noise ratio）定义如下：

$$\text{SNR}=\frac{\text{平均信号功率}}{\text{平均噪声功率}}$$

我们需要考虑平均信号功率和平均噪声功率，因为这些会随时间变化。图3.30说明了SNR的概念。



a) 大SNR



b) 小SNR

图3.30　SNR的两种情形：高SNR和低SNR

SNR实际上是我们需要的（信号）和我们不需要的（噪声）之间的比率。较高的SNR意味着信号较少地被噪声破坏，较低的SNR意味着信号被噪声更多地破坏。

因为SNR是两个功率的比率，所以一般用分贝单位描述，SNR_{dB}定义如下：

$$SNR_{dB} = 10\log_{10}SNR$$

例3.31

信号的功率是10mW，噪声的功率是1μW，那么SNR和SNR_{dB}是多少？

解：

SNR和SNR_{dB}可以计算如下：

$$SNR = \frac{10\ 000\mu W}{1mW} = 10\ 000$$

$$SNR_{dB} = 10\log_{10}10\ 000 = 10\log_{10}10^4 = 40$$

例3.32

对于无噪声通道的SNR和SNR_{dB}是：

$$SNR = \frac{信号功率}{0} = \infty$$

$$SNR_{dB} = 10\log_{10}\infty = \infty$$

我们在现实中不会得到这样的比率，它是理论上的。

3.5 数据速率限制

数据通信中一个非常重要的问题是：在一个通道中能够以多快的速率发送数据，即每秒钟的比特数。数据速率取决于三种因素：

1. 有效带宽；
2. 使用的信号电平数；
3. 通道的质量（噪声电平）。

有两个理论上的公式用于计算数据速率：一个是由奈奎斯特提出的，用于无噪声通道；另一个是由香农提出的，用于噪声通道。

3.5.1 无噪声通道：奈奎斯特比特率

对于无噪声通道，奈奎斯特比特率（Nyquist bit rate）公式定义了理论上的最大比特率：

$$比特率 = 2 \times 带宽 \times \log_2 L$$

公式中，带宽是指通道的带宽，L是用于表示数据的信号电平的数量，比特率是每秒中的位数。

根据这个公式，我们可以认为：给定特定的带宽可以通过增加信号电平数获得任何我们想要的比特率。虽然这个思想在理论上是正确的，但是实际上有局限。当我们增加信号电平数时，我们增加了接收方的负担。如果信号的电平数只是2，接收方可以轻易地区分0和1。如果信号的电平数是64，接收方必须很复杂才能区分64个不同的电平。换言之，增加信号电平数减弱了系统的可靠性。

> 增加信号电平数会减弱系统的可靠性。

例3.33

奈奎斯特公式是否适用于宽带传输中描述的音频上的比特率？

解：

当只有两个电平时它们匹配。我们说，在基带传输中，如果我们在最坏情况只使用第一谐波，那么比特率是2倍带宽。但是，奈奎斯特公式比我们直觉上得到的更通用。它可以应用到基带传输和调制，也可以应用到有两个或多个信号电平的情况。

例3.34

考虑带宽为3000Hz的无噪声通道，传输两种电平的信号。则最大比特率计算为：

$$比特率 = 2 \times 3\,000 \times \log_2 2 = 6\,000 bps$$

例3.35

考虑同样的无噪声通道，传输具有四种信号电平的信号（每一种电平发送两个位），最大比特率可以计算为：

$$比特率 = 2 \times 3\,000 \times \log_2 4 = 12\,000 bps$$

例3.36

我们需要通过带宽为20kHz的无噪声通道发送265kbps，我们需要多少信号电平？

解：

我们使用奈奎斯特公式如下：

$$265\,000 = 2 \times 20\,000 \times \log_2 L$$

$$\log_2 L = 6.625 \quad L = 2^{6.625} = 98.7 个电平$$

因为结果不是2的幂，所以我们需要增加电平数或者减小比特率。如果我们有128个电平，比特率是280kbps，如果我们有64个电平，比特率是240kbps。

3.5.2　噪声通道：香农容量定理

实际情况中，不可能有无噪声通道，通道总是有噪声的。1944年，Claude Shannon引进了一个公式，称为香农容量定理（Shannon Capacity），能够确定噪声通道理论上的最高数据速率：

$$通道容量 = 带宽 \times \log_2 (1 + SNR)$$

公式中，带宽是指通道的带宽，SNR是信噪比，通道容量是指通道的传输容量，即每秒的比特数。注意：香农公式中没有指出信号电平。这意味着无论使用多少个电平，都不可能获得比通道容量更高的数据速率。换句话说，公式定义了通道的特性，而不是传输方式。

例3.37

考虑一个极端的噪声通道，其信噪比的值接近于0。换句话说，噪声很强而信号很弱。对于该通道，其容量C计算为：

$$C = B \log_2 (1+SNR) = B \log_2 (1+0) = B \log_2 1 = B \times 0 = 0$$

这意味着通道的容量为0，与带宽无关。换句话说，通过该通道不能接收到任何数据。

例3.38

我们可以计算一条常规电话线路理论上的最高比特率。通常情况下，电话线路的带宽为3 000Hz（300Hz～3 300Hz）。信噪比通常为3 162。对于这一通道，其容量计算为：

$$C = B \log_2 (1+SNR) = 3\,000 \log_2 (1+3\,162) = 3\,000 \log_2 3\,163$$

$$= 3\,000 \times 11.62 = 34\,860 bps$$

这表示电话线路的最高比特率是34 860bps。如果要使数据发送速率比这更快，则可增大

线路的带宽或者提高信噪比。

例3.39

信噪比通常以分贝给定。假定$SNR_{dB}=36$，通道带宽是2MHz。理论上的通道容量计算为：

$$SNR_{dB}=10\log_{10}SNR \rightarrow SNR=10^{SNR_{dB}/10} \rightarrow SNR=10^{3.6}=3\,981$$

$$C=B\log_2(1+SNR)=2\times10^6\times\log_2 3\,982=24\text{Mbps}$$

例3.40

为了实际操作方便，当SNR很大时，假定SNR+1几乎与SNR相等，理论通道容量可以简化为：

$$C=B\times\frac{SNR_{dB}}{3}$$

例如，我们可以计算前一例子的理论容量为：

$$C=2\text{MHz}\times\frac{36}{3}=24\text{Mbps}$$

3.5.3 使用两种限制条件

在实际应用中，两种方法都要使用，以确定所需要的带宽以及信号电平数。下面通过一个实例说明这一点。

例3.41

有一个1MHz带宽的通道。通道的信噪比是63，合适的比特率以及信号电平是多少？

解：

首先，使用香农公式确定上限：

$$C=B\log_2(1+SNR)=10^6\log_2(1+63)=10^6\log_2 64=6\text{Mbps}$$

尽管香农公式计算结果是6Mbps，但这是上限。为了获得更好的性能，可选择低一些的值，如4Mbps。然后使用奈奎斯特公式计算信号电平的数量：

$$4\text{Mbps}=2\times1\text{MHz}\times\log_2 L \rightarrow L=4\ (\text{个})$$

> 香农容量定理给出数据速率的上限，奈奎斯特公式给出所需的信号电平数。

3.6 性能

到目前为止，我们已经讨论了在网络上传输数据（信号）的工具以及如何传输数据。网络中的一个重要问题是网络的性能——网络怎么好？我们会在第21章更详细地讨论服务质量、网络性能的整体衡量。这一节，我们介绍以后章节会用到的术语。

3.6.1 带宽

衡量网络性能的一个特性就是带宽。但是，这个术语在两种不同情况下将使用不同的物理量值：以赫兹衡量的带宽和以每秒比特数衡量的带宽。

以赫兹衡量的带宽

我们已经讨论过这个概念。以赫兹衡量的带宽是复合信号包含的频率范围或者通道能通过的频率范围。例如，我们可以说用户电话线的带宽是4kHz。

以每秒比特数衡量的带宽

术语带宽还可以指通道、链路或者甚至网络每秒能发送的位数。例如，我们可以说快速以太网（或者这种网络中的链路）的带宽最高达100Mbps。这意味着这种网络能发送100Mbps。

两者关系

在以赫兹衡量的带宽和以每秒比特数衡量的带宽之间有明显的关系。基本上，以赫兹衡量的带宽的增长意味着以每秒位数衡量的带宽的增长。它们之间的关系取决于是否有基带传输或者调制传输。我们会在第4章和第5章讨论这个关系。

> 在网络中，我们在两种情形下使用术语带宽：
> - 第一种，以赫兹衡量的带宽指复合信号包含的频率范围或者通道能通过的频率范围。
> - 第二种，以每秒比特数衡量的带宽指通道或链路中数传输的速率。

例3.42

用于语音或数据的用户线带宽是4kHz。使用复杂的调制解调器把数字信号转换成模拟信号，这种线路用于数据传输的带宽可以达到56 000bps。

例3.43

如果电话公司提高线路的质量，把带宽增加到8kHz。通过例3.42提到的相同技术，可以发送112 000bps。

3.6.2 吞吐量

吞吐量（throughput）用于衡量通过网络发送数据的快慢。虽然，第一感觉上，以每秒位数衡量的带宽和吞吐量看起来一样，但是它们是不同的。一条链路可以有B bps的带宽，但是我们只能通过这条链路发送T bps，T永远小于B。换言之，带宽是链路的潜在衡量值，而吞吐量是发送速度快慢的实际衡量值。例如，我们可以有一条带宽为1Mbps的链路，但是连接到链路末端的设备只能处理200kbps。这意味着我们不能通过这条链路发送数据高于200kbps。

想像一下，一条高速公路设计从一点到另一点每分钟运送1 000辆汽车。但是，如果在路上有拥塞，这个数字会降低到每分钟100辆汽车。带宽是每分钟1 000辆汽车，而吞吐量是每分钟100辆汽车。

例3.44

带宽为10Mbps的网络每分钟只能平均传输12 000个帧，每个帧平均携带10 000个位。那么这个网络的吞吐量是多少？

解：

我们可以计算吞吐量为：

$$\text{吞吐量} = \frac{12\,000 \times 10\,000}{60} = 2 \text{ Mbps}$$

这个例子中的吞吐量几乎是带宽的五分之一。

3.6.3 延迟

延迟定义了第一个位从源开始发出到整个报文完全到达目标所经历的时间。我们可以说延迟由四个部分组成：**传播时间**（propagation time）、**传输时间**（transmission time）、**排队时间**（queuing time）和**处理延迟**（processing delay）。

延迟=传播时间+传输时间+排队时间+处理延迟

传播时间

传播时间衡量一个位从源传输到目标所需的时间。传播时间通过距离除以传播速度计算而得。

传播时间=距离/传播速度

电磁信号的传播速度取决于介质和信号频率。例如，在真空中，光以速度3×10^8m/s传播。它在空气中速度较低，在电缆中更低。

例3.45

如果两点之间的距离是12 000km，传播时间是多少？假定电缆中的传播速度是2.4×10^8m/s。

解：

我们可以计算传播速度为：

$$\text{传播速度}=\frac{12\ 000\times1\ 000}{2.4\times10^8}=50\ \text{ms}$$

这个例子说明如果源和目标之间有点到点电缆，一个位经过大西洋只需要50ms。

传输时间

在数据通信中我们不只发送1个位，我们需要发送报文。第一个位可能会花费跟传播时间相等的时间到达目标，最后一个位也可能需要相同的时间。但是，在第一个位离开发送方到最后一个位到达接收方之间有个时间，第一个位较早离开到达也较早，最后一个位较晚离开也较晚到达。传输一个报文所需时间取决于报文的长度和通道的带宽。

传输时间=报文长度/带宽

例3.46

如果网络的带宽是1Gbps，那么2.5千字节的报文（一封电子邮件）的传播时间和传输时间是多少？假定发送方和接收方之间的距离是12 000km，光以速度2.4×10^8m/s传输。

解：

我们计算传播时间和传输时间为

$$\text{传播时间}=\frac{12\ 000\times1\ 000}{2.4\times10^8}=50\ \text{ms}$$

$$\text{传输时间}=\frac{2\ 500\times8}{10^9}=0.020\ \text{ms}$$

注意：在这个例子中，因为报文较短而带宽较高，主导因素是传播时间而不是传输时间。传输时间可以忽略不计。

例3.47

如果网络带宽是1Mbps，那么5M字节的报文（一张图像）的传播时间和传输时间是多少？假定发送方和接收方之间的距离是12 000km，光以速度2.4×10^8m/s传输。

解：

我们计算传播时间和传输时间为

$$\text{传播时间}=\frac{12\ 000\times1\ 000}{2.4\times10^8}=50\ \text{ms}$$

$$\text{传输时间}=\frac{5\ 000\ 000\times8}{10^6}=40\ \text{s}$$

注意这个例子中，因为报文较长而带宽不是很高，所以主导因素是传输时间而不是传播时间，传播时间可以忽略不计。

排队时间

延迟中的第三个成分是排队时间，即每个中间或端设备在处理报文前保持报文所需的时间。排队时间不是固定因素，它会根据网络负载改变。当网络上负载较重时，排队时间增长。一个中间设备（比如路由器）对到达报文进行排队然后一个接一个处理。如果有许多报文，每个报文就不得不等待。

3.6.4　带宽与延迟的乘积

带宽和延迟是链路的两个性能度量。但是，正如我们将在本章和以后章节中看到的，数据通信中十分重要的是两者的乘积，让我们使用两个假想情形作为例子详细阐述这个问题。

- 情形1：图3.31说明了情形1。

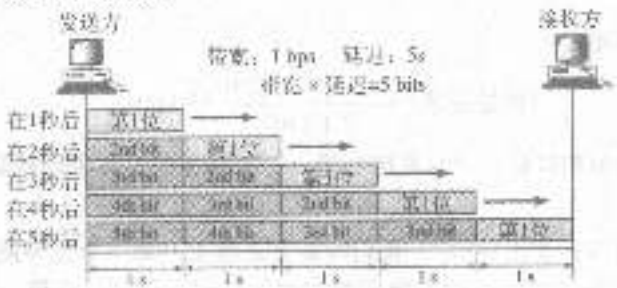


图3.31　在情形1中用位充满链路

假定我们有一条带宽是1bps（不现实，但用于说明目的）的链路。还假定链路的延迟是5s（也是不现实的）。我们想看这个情形中带宽延迟乘积意味着什么。从图中，我们可以说这个乘积1×5是能充满链路的最大位数。链路上在任何时都不可能多于5位。

- 情形2：现在假定带宽是4bps。图3.32说明了链路上最多可以有4×5=20位。原因是每一秒钟链路上有4位，每位的持续时间是0.25s。

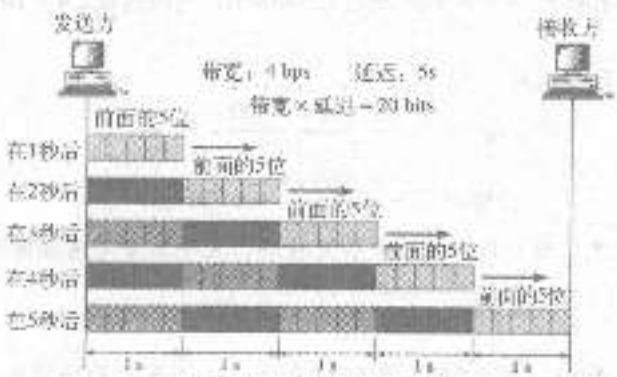


图3.32　在情形2下位充满链路

上述两种情形说明带宽和延迟的乘积是能充满链路的位的个数。如果我们需要以脉冲发送数据并在发送下一个脉冲前等待每个脉冲的确认，这个度量是重要的。为了使用链路的最大容量，我们需要脉冲长度是带宽和延迟乘积的两倍，我们需要充满全双工通道（两个方向）。发送者应该发送一个（2×带宽×延迟）位的数据脉冲。然后发送方在发送下一个脉冲前等待这个脉冲部分的接收方的确认。数量2×带宽×延迟是任何时刻转换的位数。

带宽延迟乘积定义了能充满链路的位数。

例3.48

我们把两点间的链路看做管道，管道的横截面表示带宽，管道的长度表示延迟。我们可以说管道的容量定义了带宽延迟乘积，如图3.33所示。

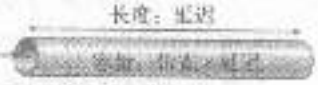


图3.33 带宽延迟乘积的概念

3.6.5 抖动

另一个与延迟有关的性能问题是抖动(jitter)。如果数据的不同分组碰到不同的延迟并且接收方使用数据的应用是时间敏感的（例如音频和视频数据），我们可以大致说抖动是个问题。如果第一个分组的延迟是20ms，第二个是45ms，第三个是40ms，那么使用这些分组的实时应用会遭遇抖动。我们会在第29章更详细地讨论抖动。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物。在本书末列出了括号[……]中的参考资料。

读物

数据和信号在[Pea92]的第1章到第6章中有详细的讨论。[Cou01]的第2章给出有关信号完整的讨论。[Ber96]给出进一步深入研究的资料。[Hsu03]给出信号严格的数学方法描述。[Spi74]讨论傅里叶分析。[Sta04]的第3章和[Tan03]的2.1节讨论数据和信号。

本章小结

- 数据在通过网络传输之前必须转换为电磁信号。
- 数据可以是模拟的也可以是数字的。模拟数据是连续状态，取连续的值。数字数据具有离散状态，取离散的值。
- 信号可以是模拟的也可以是数字的。模拟信号具有在某范围内无限个数值。数字信号仅具有有限个值。
- 在数据通信中，常用周期模拟信号和非周期数字信号。
- 频率和周期是互逆的。
- 频率是单位时间的变化率。
- 相位是波形相对于时间零的位置。
- 用频域图中的一个振幅峰值表示时域图中的一个完整正弦波。
- 在数据通信中，单一频率的正弦波是没有用的。我们需要发送由多个简单正弦波组成的复合信号。
- 按照傅里叶分析，任何复合信号是由具有不同频率、振幅和相位的正弦波叠加而成。
- 复合信号的带宽是该信号的最高频率与最低频率的差值。
- 数字信号是一个具有无穷带宽的复合信号。
- 仅当具有无穷大带宽或很大的带宽的低通通道时，基带传输才能保持数字信号的形状。
- 如果一个有效通道是带通通道，则它不能直接发送数字信号，在传输之前需将数字信号转换成模拟信号。

- 奈奎斯特公式能够确定无噪声通道理论上的最大比特率。香农容量原理能够确定噪声通道理论上的最大比特率。
- 衰减、失真和噪声会减损信号。
- 衰减是由于介质的阻抗所导致的信号能量的损失。
- 失真是由于构成信号的不同频率成分的传播速度不同而引起的信号改变。
- 噪声是损害信号的外部能量。
- 带宽与延迟的乘积定义了充满链路的位数。

习题

复习题

1. 周期与频率有什么关系？
2. 信号振幅的度量是什么？信号频率的度量是什么？信号周期的度量是什么？
3. 复合信号如何分解成单独的频率成分？
4. 试说出传输减损三种类型的名称。
5. 基带传输与宽带传输的区别。
6. 低通通道与带通通道的区别。
7. 奈奎斯特公式在通信中有什么作用？
8. 香农容量原理在通信中有什么作用？
9. 光缆中的光信号为什么是很短波长？
10. 是否可用频域观察一个信号是周期的还是非周期的？为什么？
11. 语音信号的频域图是离散的还是连续的？
12. 报警系统的频域图是离散的还是连续的？
13. 语音信号从麦克风到录音机的发送是基带传输还是宽带传输？
14. 在局域网中，一个数字信号从一个站到另一个站的发送是基带传输还是宽带传输？
15. 调制一些语音信号并通过空气发送它们，是基带传输还是宽带传输？

练习题

16. 给定下列频率，计算相应的周期。
    a. 24Hz　　b. 8MHz
    c. 140kHz
17. 给定下列周期计算相应的频率。
    a. 5s　　b. 12μs
    c. 220ns
18. 下列各题中相移是多少？
    a. 在时间0具有最大振幅的正弦波　　b. 经过1/4周期达到最大振幅的正弦波
    c. 经过3/4周期，上升段振幅达到0值的正弦波
19. 一个信号被分解成5个正弦波，它们频率分别为0、20、50、100和200Hz，该信号的带宽是多少？它们的峰值振幅都相同，试画出该带宽。
20. 一个带宽为2000Hz的复合信号由两个正弦波组成。其中一个频率为100Hz，最大振幅为20V，另一个最大振幅为5V，画出其带宽。
21. 一个频率为100Hz的正弦波，另一个频率为200Hz的正弦波。试问哪一个正弦波带宽较大？
22. 下列信号的比特率各是多少？

a. 一个位持续0.001s　　b. 一个位持续2ms

c. 一个位持续2μs

23. 一个设备以1 000bps发送数据

a. 发送10位需要多长时间？　　b. 发送一个字符（8位）需要多长时间？

c. 发送100 000个字符的文件需要多长时间？

24. 图3.34的信号比特率是多少？

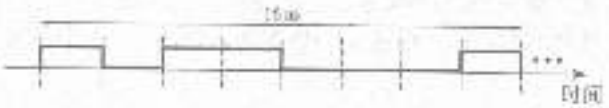


图3.34 练习题24

25. 图3.35的信号频率是多少？

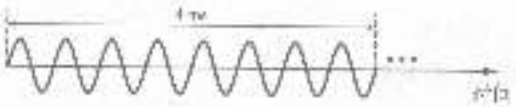


图3.35 练习题25

26. 图3.36的复合信号的带宽是多少？



图3.36 练习题26

27. 周期复合信号包含10kHz到30kHz的频率成分，每一个成分的振幅为10V，试画出其频谱。

28. 非周期复合信号包含10kHz到30kHz的频率成分，信号的最低频率成分和最高频率成分的振幅是10V，20kHz信号的振幅是30V。假定振幅由最小值到最大值是渐变的，试画出其频谱。

29. TV频道的带宽是6MHz，如果用一个通道发送数字信号，分别用第一谐波、第三谐波和第五谐波的数据速率是多少？

30. 信号从A点到达B点，在A点时功率是100W，在B点时功率是90W。问它衰减多少分贝？

31. 信号衰减是10dB，如果信号的初始功率是5W，问衰减后的功率是多少？

32. 信号通过三级放大器，每级的增益是4dB，问总增益是多少？信号放大了多少倍？

33. 如果通道的带宽是5kbps，那么从设备发送100 000位的帧需要多长时间？

34. 太阳光到达地球大约需要8分钟，问太阳与地球之间距离是多少？

35. 信号在空气中的波长是1μm，1 600个周期波的前端能传输多远？

36. 一条线路的信噪比是1 000，带宽是4 000kHz，该线路能支持的最大数据速率是多少？

37. 要测试电话线路（带宽4kHz）的性能，信号为10V，噪声是5mV，该电话线路支持的最大数据速率是多少？

38. 一个文件有2GB字节，用56kbps通道下载该文件需要多少时间？用1Mbps下载需多少时间？

39. 一台计算机显示器的分辨率是1 200×1 000个像素，如果每个像素使用1 024个颜色，发送一个完整屏幕需要多少位？

40. 具有200毫瓦功率的信号通过10个设备，每个设备平均噪声是2微瓦，问SNR是多少？SNR_{dB}是多少？

41. 如果信号峰值电压值是噪声峰值的20倍，问SNR是多少？SNR_{db}是多少？
42. 下列各情形的通道，理论上的容量是多少？
    a. 带宽:20kHz，$SNR_{db} = 40$　　b. 带宽:200kHz，$SNR_{db} = 4$
    c. 带宽:1MHz，$SNR_{db} = 20$
43. 需要提高通道的带宽，问题是：
    a. 如果是高带宽两倍，如何改进速率？　　b. 如果提高SNR两倍，如何改进速率？
44. 通道具有4kHz的带宽，如果想以100kbps发送数据，则最小的SNR_{db}是多少？SNR是多少？
45. 如果一个分组的长度是1兆字节，而通道的带宽是200kbps，一个站发送一个分组要多少传输时间？
46. 通道的传播速率是2×10^8 m/s，如果通道速率如下：
    a. 1Mbps　　b. 10Mbps
    c. 100Mbps
    问通道内的一位的长度是多少？
47. 如果链路具有2ms的延迟，带宽如下：
    a.1Mbps　　b.10Mbps
    c.100Mbps
    问在链路上填满有多少位？
48. 一条链路有10个路由器，每个路由器排队的时间是2μs，处理时间为1μs，在其上发送具有5兆字节的一个帧。问总的延迟是多少？链路长为200km，链路中内的光速是2×10^8m/s，带宽是5Mbps，决定总延迟的是哪个成分？哪些成分可忽略不计？

第4章 数字传输

计算机网络用于在网络节点之间传送信息。信息需要转换为数字信号或者模拟信号再传输。本章讨论数字信号的转换。在第5章中会讨论模拟信号的转换。

第3章讨论了数字传输相对于模拟传输的优点和缺点。本章说明以数字方式传输数据所用的机制和技术。首先，我们讨论**数字到数字转换**（digital-to-digital conversion）技术，这种技术将数字数据转换为数字信号。其次，我们讨论**模拟到数字转换**（analog-digital conversion）技术，这种技术将模拟数据转换为数字数据。最后，我们讨论**传输模式**（transmission mode）。

4.1 数字到数字转换

在第3章中，我们讨论了数据和信号。已说过数据可以是数字的也可以是模拟的，还说过表示数据的信号也可以是数字的或者模拟的。本节中，会看到如何使用数字信号表示数字数据。转换涉及三种技术：线路编码、块编码和扰动。线路编码总是需要的，块编码和扰动可能需要也可能不需要。

4.1.1 线路编码

线路编码（line coding）是将数字数据转换为数字信号的过程。假定以文本、数字、图形图像、音频或视频为形式的数据在计算机内存中都是以位序列形式保存的（见第1章）。线路编码将位序列转换为数字信号。在发送方，数字数据被编码成数字信号，在接收方，解码数字信号重新生成原数字数据。图4.1说明了这个过程。

特性

在讨论不同的线路编码方案之前，先讲述它们的共同特性。

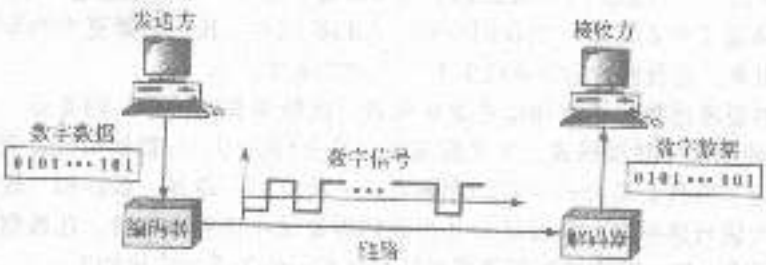


图4.1 线路编码和解码

信号元素与数据元素 先让我们区分**数据元素**（data element）和**信号元素**（signal element）。在数据通信中，我们的目标是发送数据元素。数据元素是表示一块信息的最小实体，即位。在数字数据通信中，信号元素承载数据单元。信号元素是数字信号的最小单元。换言之，数据元素是我们需要发送的，而信号元素是我们能发送的。数据元素是被承载，而信号元素是载体。

我们定义比率r为每个信号元素承载的数据元素的数量。图4.2说明了几种不同r值的情况。

在图4.2a中，一个数据元素被一个信号元素承载（$r = 1$）。在图4.2b中，我们需要两个信号元素（两个转换）来承载每个数据元素（$r = 1/2$）。我们会在后面看到需要额外的信号元素

来保证同步。在图4.2c中，一个信号元素承载两个数据元素（$r = 2$）。最后在图4.2d中，三个一组的信号元素承载四个一组的数据元素（$r = 4/3$）。对于我们讨论的每个线路编码，我们都会给出r值。

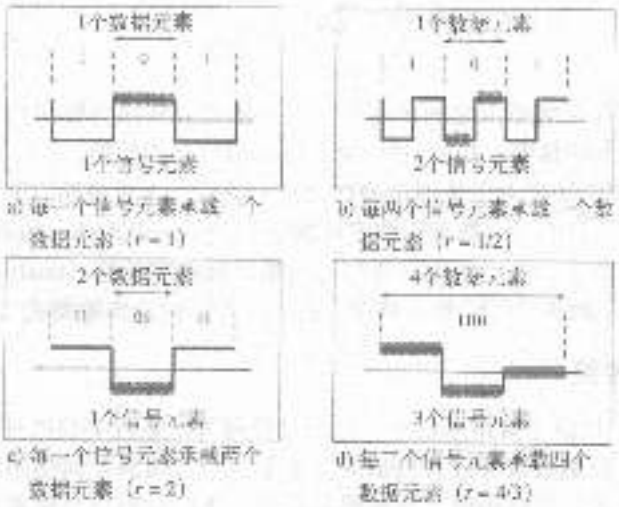


图4.2　信号元素与数据元素

借用比喻说明，假定每个数据元素是需要从一个地方运到另一个地方的人。我们可以把运送人的车辆看做是信号元素。当$r = 1$，意味着每个人驾驶一辆车。当$r > 1$时，意味着多于一个人乘坐一辆车（比如合伙使用车）。还可以有一个人驾驶一辆车和一辆拖车（$r = 1/2$）。

数据速率和信号速率　数据速率（data rate）定义了1秒发送的数据元素（位）的数量，单位是每秒位（bps）。信号速率（signal rate）是1秒发送的信号元素的数量，单位是波特（baud）。还有很多通用的术语。数据速率有时称为比特率（bit rate），信号速率有时称为脉冲速率（pulse rate）、调制速率（modulation rate）或波特率（baud rate）。

数据通信中的一个目标是增加数据速率而降低信号速率。增加数据速率增加了传输速度，降低信号速率降低了带宽需求。在我们车辆一人的类比中，我们需要更少的车辆运送更多的人来防止交通阻塞。在我们的运输系统中有一个限定带宽。

现在我们需要考虑数据速率和信号速率两者（比特率和波特率）的关系。当然这个关系取决于r值。它还取决于数据模式。如果数据模式是全1或全0，与数据模式是交替0和1信号速率是不同的。为了得到关系公式，我们需要定义三种情形：最坏、最好和一般。最坏情形是当我们需要最大信号速率时，最好情形是当我们需要最小信号速率时。在数据通信中，我们通常对一般情形感兴趣。我们把数据速率和信号速率间的关系公式化如下：

$$S = c \times N \times \frac{1}{r} \text{ baud}$$

这里N是数据速率（bps），c是情形因子，会根据每种情形改变，S是信号元素数量，而r是前面定义的因子。

例4.1

一个信号携带数据，一个数据元素编码成一个信号元素（$r = 1$）。如果比特率是100kbps，c在0和1之间，那么波特率的平均值是多少？

解：

假定c的平均值是1/2，那么波特率是：

$$S = c \times N \times \frac{1}{r} = \frac{1}{2} \times 100\ 000 \times \frac{1}{1} = 50\ 000 = 50\ \text{kbaud}$$

带宽 我们已经在第3章中讲过携带信息的数字信号是非周期性的，我们还说明了非周期性信号的带宽是无限连续的。但是，我们在实际中碰到的大多数数字信号的带宽是有限的。换言之，带宽理论上是无限的，但是许多成分的振幅很小可以忽略不计。有效的带宽是有限的。从现在开始，当我们说到数字信号的带宽时，我们需要记住我们讲的是这个有效带宽。

> 虽然数字信号的真实带宽是无限的，但有效带宽是有限的。

可以说是波特率而不是比特率决定了数字信号的带宽。如果我们使用运输作为类比，是车辆数量影响了交通，而不是运输的人数。信号中更多的变化意味着在信号中插入了更多的频率，带宽影响了我们所需的频率范围。波特率（信号速率）和带宽之间是有关系的。带宽是一个复杂的概念。当我们讲到带宽，我们通常定义为频率范围。我们需要知道这个范围位于哪里以及最低频率和最高频率的值。另外，每个成分的振幅（如果不是振幅就是相位）是个重要问题。换言之，我们需要有关带宽的更多信息而不仅仅是它的值，我们需要带宽的图表。我们会说明本章中讨论的大多数方案的带宽。这里我们说带宽（频率范围）与信号速率（波特率）成正比。最小带宽可以如下给出：

$$B_{min} = c \times N \times \frac{1}{r}$$

如果给出通道带宽，我们可以得出最大数据速率如下：

$$N_{max} = \frac{1}{c} \times B \times r$$

例4.2

通道的最大数据速率（见第3章）是$N_{max}=2 \times B \times \log_2 L$（由奈奎斯特公式定义）。这与前面的$N_{max}$公式一致吗？

解：

有L个电平的信号，每个电平可以携带$\log_2 L$个位。如果每个电平与一个信号元素对应，假定是一般情形（c =1/2），那么

$$N_{max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L$$

基线偏移 在解码数字信号时，接收方计算接收到信号功率的运行平均值。这个平均值称为基线（baseline）。输入信号的功率会与基线比较来确定数据元素的值。一个0或1的长字符串会引起基线偏移（baseline wandering），使得接收方不能正确地进行解码。一个好的线路编码方案需要防止基线偏移。

直流成分 当数字信号中的电平保持一段时间的恒定时，频谱会产生很低的频率（傅里叶分析的结果）。这些接近于零的频率称为DC（直流）成分，会给不允许通过低频率的系统或者使用电子耦合的系统（如变压器）带来问题。例如，电话线不能通过低于200Hz的频率。还有，长距离的链路可能使用一个或多个变压器来隔离线路的不同部分。对于这些系统，我们使用无直流成分（DC component）的方案。

自同步 为了正确地解释从发送方接收到的信号，接收方的位间隔必须与发送方的位间隔严格对应。如果接收方的时钟快了或者慢了，位间隔就不会匹配，接收方可能会错误地解

释信号。图4.3说明了接收方位间隔较短的一种情况。发送方发送10110001，而接收方收到的是1101110000011。

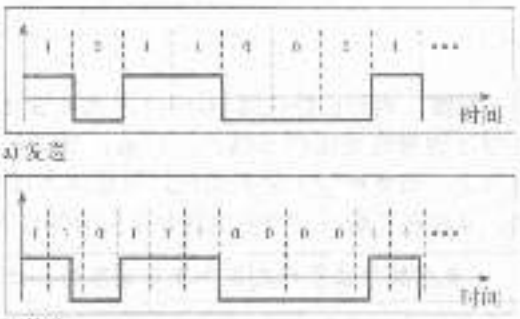


图4.3　缺乏同步的结果

自同步（self-synchronizing）数字信号在传输的数据中包含有定时信息。如果信号中包含有提示接收方起始、中间和结束位置的脉冲的跳变就可以完成自同步。如果接收方的时钟不同步，这些提示跳变就会将接收方的时钟复位。

例4.3

在数字传输中，接收方时钟比发送方时钟快0.1%，如果数据速率是1kbps，则接收方每秒钟可以接收到多少额外的位？如果数据速率是1Mbps呢？

解：

在1kbps时，接收方接收速度是1 001bps而不是1 000bps。

发送1 000位→接收到1 001位→多出1位

在1Mbps时，接收方接收速度是1 001 000bps而不是1 000 000bps。

发送1 000 000位→接收到1 001 000位→多出1 000位

内置差错检测　在产生的代码中拥有内置差错检测能力来检测传输过程中发生的一些差错是需要的。我们将讨论的一些编码方案中有一定程度的这种能力。

抗噪声和抗干扰的能力　另一个需要的编码特性是编码应具有抗噪声和抗干扰能力。我们将要讨论的一些编码方案有这种能力。

复杂性　复杂的方案比简单的方案实现起来成本更高。例如，使用4个信号电平的方案比只使用2个电平的方案更难。

4.1.2　线路编码方案

线路编码机制可以大致分为五大类，如图4.4所示。



图4.4　线路编码方案

每一大类中又有很多方案。我们需要熟悉本节所讨论的所有方案，以便于理解本书后面的内容。这一节作为后面所碰到方案的参考。

单极编码方案

在单级编码方案中，所有的信号电平都在时间轴的一边，或者时间轴的上面或者时间轴的下面。

不归零（NRZ）　一般，单级编码方案设计成不归零（non-return-to-zero，NRZ）方案，在此方案中正电平定义成位1而零电平定义成位0。之所以称为NRZ，是因为在位中间信号不会回到零。图4.5说明了单级NRZ方案。

与它的极性对应方案相比（见下节），这个方案成本很高。正如我们很快会看到的，标准功率（每个单元线路阻抗发送1位所需的功率）是极性NRZ方案的两倍。基于这个原因，现在这个方案不用于数据通信中。

极性编码方案

在极性编码方案中，电平在时间轴的两边。例如，0的电平可能是正的，而1的电平可能是负的。

不归零编码（NRZ）在极性NRZ（polar NRZ）编码中，信号的振幅使用两个电平。极性NRZ有两种常见形式：NRZ-L和NRZ-I，如图4.6所示。图中也说明了r值、平均波特率和带宽。第一种形式NRZ-L（NRZ电平编码，NRZ-Level），信号电平决定了位值。而第二种形式NRZ-I（NRZ反相编码，NRZ-Invert），信号电平是否反相转或跳变决定了位值。如果没有跳变，位值是0，如果有跳变，位值是1。

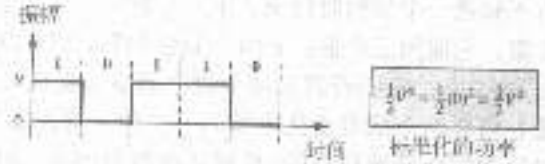


图4.5 单级NRZ方案

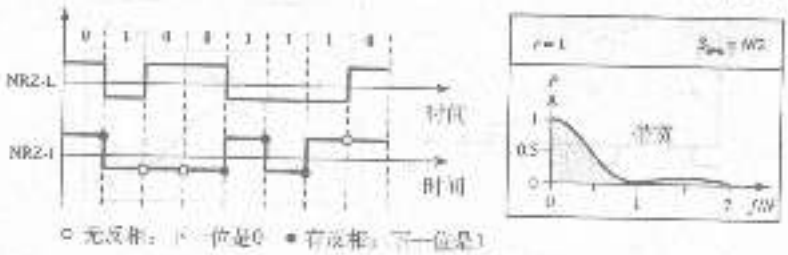


图4.6 极性NRZ-L方案和极性NRZ-I方案

在NRZ-L中，电平决定了位值。在NRZ-I中电平是否反相转决定了位值。

让我们基于前面规定的标准来比较这两个方案。虽然基线偏移对这两个方案都是个问题，但NRZ-L比NRZ-I更严重。如果在NRZ-L中有一个全0或全1的长序列，平均信号功率就会变斜，接收方辨别位值就会较困难。而在NRZ-I中这个问题只发生在全0的长序列中。如果我们能消除全0的长序列，我们就可以避免基线偏移。我们很快就会看到如何做到这一点。

同步问题（发送方和接收方的时钟不同步）也都存在于这两个方案中。同样，这个问题在NRZ-L中比NRZ-I中更严重。在这两个方案中，全0的长序列都会引起问题，而全1的长序列只在NRZ-L中引起问题。

当系统中的极性意外改变时，NRZ-L会发生另一个问题。例如，如果介质是双绞线，该双绞线意外改变引起将所有0解释为所有的1或所有的1解释为所有0。NRZ-I不会有这个问题。两个方案都有$N/2$波特的平均信号速率。

NRZ-L和NRZ-I都有$N/2$ Bd的平均信号速率。

接下来讨论带宽。图4.6中还表示了两个方案的标准带宽。垂直轴表示功率密度（每1Hz带宽的功率），水平轴表示频率。对这种类型编码，带宽隐藏了一个很严重的问题。在频率接近于零时，功率密度值会很高。这意味着有携带了高电平能量的DC成分。实际上，大多数能量集中在0到$N/2$之间的频率。这意味着虽然平均信号速率是$N/2$，但是能量被平均在这两个值之间的频率中。

NRZ-L和NRZ-I都有DC成分问题。

例4.4

系统使用NRZ-I传输10Mbps数据，试问平均信号速率和最小带宽是多少？

解：

平均信号速率是$S = N/2 = 500$ kBd，这个平均波特率的最小带宽是$B_{min} = S = 500$ kHz。

归零编码（RZ）　当发送方和接收方时钟不同步时，NRZ编码的主要问题就发生了。接收方不知道一个位何时结束，下一个位何时开始。一种解决方法是归零编码（return-to-zer，RZ）方案，它使用三个值：正值、负值和零。在RZ中，信号不在两个位之间变化而是在位中变化。在图4.7中，我们看到在每个位中间信号变成0。它保持到下一个位开始。RZ编码的主要缺点是它需要两个信号变化来编码一个位，因此占用更大的带宽。我们前面提到的问题——极性意外改变会导致所有的0被解释成所有的1，而所有的1被解释成所有的0的问题同样存在，但没有DC成分问题。另一个问题是复杂性：RZ使用三个电平，这样生成和辨别更加困难。这些缺陷使得今天不再使用这个方案。它由性能更好的曼彻斯特编码和差分曼彻斯特编码方案替代（会在后面部分讨论）。

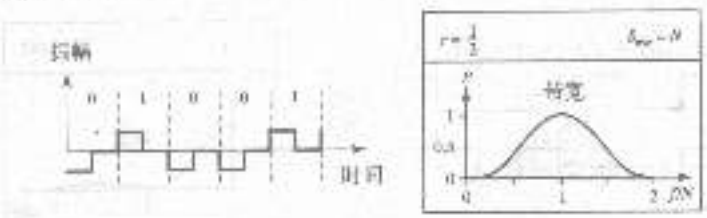


图4.7　极性RZ编码方案

双相编码：曼彻斯特编码和差分曼彻斯特编码　RZ的思想（位中间跳变）和NRZ-L的思想共同组成了曼彻斯特（Manchester）编码方案。在曼彻斯特编码中，位的持续时间被二等分。在前半部分电平保持一个水平，后半部分变成另一个水平。位中间的跳变提供了同步。另一方面，差分曼彻斯特（Differential Manchester）组合了RZ和NRZ-I的思想，在位中间总是有一个跳变，但是位值在位开始时确定。如果下一个位是0，就有一个跳变。如果下一个位是1，则没有跳变。图4.8显示了曼彻斯特编码和差分曼彻斯特编码。

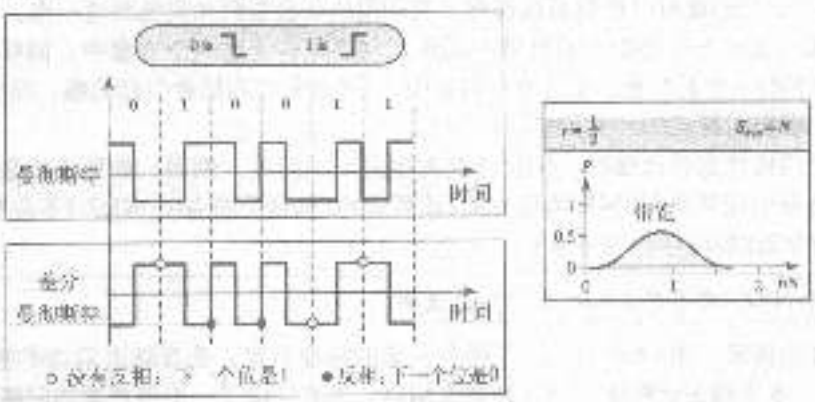


图4.8　极性双相：曼彻斯特编码和差分曼彻斯特编码方案

在曼彻斯特和差分曼彻斯特编码中，位中间的跳变用于同步。

曼彻斯特编码方案克服了NRZ-L编码的一些问题。差分曼彻斯特编码方案克服了NRZ-I编码的一些问题。首先，没有基线偏移，因为每个位是正负电平值，所以没有DC成分。唯一的缺点是信号速率。曼彻斯特编码和差分曼彻斯特编码的信号速率是NRZ的两倍。原因是每个位中间都有一个跳变，而在每个位结束可能有一个跳变。图4.8说明了曼彻斯特编码和差分曼彻斯特编码方案。注意曼彻斯特编码和差分曼彻斯特编码方案也称为两相（biphase）方案。

曼彻斯特编码和差分曼彻斯特编码的最小带宽是NRZ的两倍。

双极性方案

在双极（bipolar）编码（也称为多电平二进制（multilevel binary））中，有三个电平：正值、负值和零。一个数据元素的电平是0，而另一个数据元素的电平在正值、负值间交替。

在双极编码中，我们使用三种电平：正值、负值和零。

AMI和伪三元编码 图4.9显示了双极编码的两种编码：AMI和伪三元编码。一个常用的双极编码方案称为双极交替传号反转（alternate mark inversion，AMI）。在AMI中，传号这个名词来自电报，它指的是1。因此，AMI表示的含义是交替的1的反换。中值0电平表示二进制0，而二进制1由交替正负电平表示。AMI编码的一个变型是伪三元编码（pseudoternary），位1编码成0电平，而位0编码成交替正负电平。

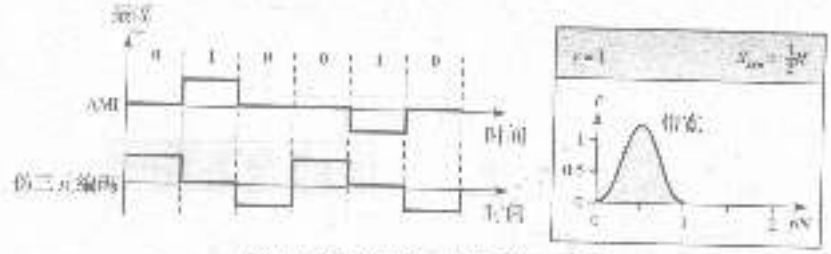


图4.9 双极方案：AMI和伪三元码

双极型方案是NRZ的替代方案，双极型方案的信号速率和NRZ一样，但是没有DC成分。NRZ方案中很大部分能量集中于频率0附近，如果通道这个频率附近性能较差，这类编码是不适合的。在双极编码中能量集中于频率$N/2$处。图4.9说明了双极方案的典型能量集中。

有人可能会问为什么在双极编码中没有DC成分？理论上，我们可以通过使用傅里叶变换来回答这个问题，但也可以直观地说明，如果有一串1的长序列，电平在正值和负值间交替，它不是常数。所以没有DC成分。对于一串0的长序列，电平保持常数，但是它的振幅是0，同样也没有DC成分。换言之，产生恒定0电平的序列不会有DC成分。

AMI通常用于长距离通信，但是当数据中存在0的长序列时就会有同步问题。在本章后面会看到用扰动技术解决这个问题。

多电平方案

增加数据速度或者降低所需带宽的需求导致了很多方案的产生。目标是通过把m个数据元素的模式编码成n个信号元素的模式，增加每波特的位数。我们只有两种数据元素（0和1），这表示m个数据元素组可以产生2^m个数据模式组合。不同信号元素可用不同的电平表示。如果有L个不同的电平，就能产生L^n个信号模式组合。如果$2^m = L^n$，那么每个数据模式编码成一个信号模式。如果$2^m < L^n$，那么数据模式只能占据一个信号模式的子集。应仔细设计这个子集来避免基线漂移、提供同步并检测数据传输中出现的差错。如果$2^m > L^n$，数据编码就不可能了，因为有一些数据模式就不能编码了。

编码设计者以$mBnL$区分这些编码类型，这里m是二进制模式的长度，B表示二进制数据，n是信号模式的长度，L是信号中的电平数。一般用字母L替换：$L = 2$是用B表示（二元），$L = 3$时用T（三元），$L = 4$时用Q（四元）。注意前两个字母定义了数据模式，后两个字母定义了信号模式。

在*mBnL*方案中，*m*个数据元素模式编码成*n*个信号元素模式，$2^m <= L^n$。

2B1Q　我们讨论的第一个*mBnL*方案，两个二元一个四元（two binary，one quaternary，2B1Q）使用长度为2的2位数据模式编码成一个4电平信号元素。在这个编码类型中，*m*=2，*n*=1，*L*=4。图4.10显示了2B1Q的一个例子。

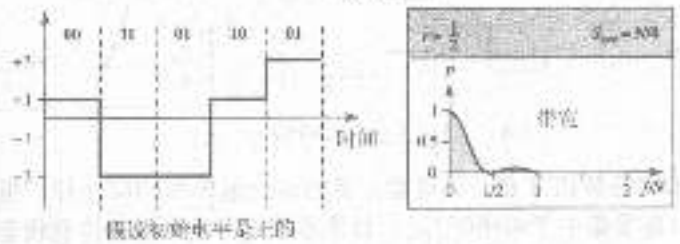
2B1Q的平均信号速率是*S*=*N*/4，这意味着使用2B1Q发送数据可以比NRZ-L快2倍。但是，2B1Q使用4个不同的信号电平，意味着接收方需要辨别4个不同的阈值。降低的带宽带来了成本的提高。因为$2^2 = 4^1$，所以这个方案中没有多余的信号模式。

正如我们会在第9章中看到，2B1Q用于DSL（数字用户线路）技术，通过使用用户电话线提供高速因特网连接。

前一个电平：正　前一个电平：负

下一个位	下一个电平	下一个电平
00	+1	-1
01	+3	-3
10	-1	+1
11	-3	+3

转换表



假设初始电平是正的

图4.10　多电平：2B1Q编码方案

8B6T　一个很有趣的方案是8个二元、6个三元（eight binary，six ternary，8B6T）。这个编码用于100BASE-4T电缆，我们会在第13章看到。这个方案是把8位模式编码成6个信号元素模式，每个信号有3个电平（三元的）。我们可以有$2^8 = 256$个不同的数据模式和$3^6 = 478$个不同的信号模式。附录D给出了映射表。有478−256=222个冗余的信号元素用来提供同步和差错检测。其中一部分还用来提供DC平衡。每个信号模式有0或+1的DC值的平衡量。这意味着没有平衡量为−1值的模式。为了DC整个流实现DC平衡，发送方必须关注平衡量。如果两个平衡量为+1的组紧接着，第一个组保持原状发送，而接着的第二个组全部取反使得平衡量为−1。

图4.11给出了3个数据模式编码成3个信号模式的例子，3个可能的信号电平表示为−、0和+。第一个8位模式00010001编码成平衡量为0的信号模式−0−0++，第二个8位数据模式01010011编码成平衡量为+1的信号模式−+−++0，第三个8位数据模式编码成平衡量为+1的信号模式+−−+0+。为了建立DC平衡，发送方反相实际的信号。接收方能轻易地辨认出这是反相过的模式，因为平衡量为−1。模式在解码前反相。

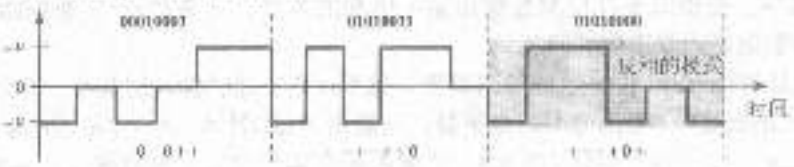


图4.11　多电平：8B6T方案

这个方案的平均信号速率理论上是 $S_{av}=\frac{1}{2}\times N\times\frac{6}{8}$，实际上最小带宽十分接近6N/8。

4D-PAM5 我们在这个类别中讨论的最后一个信号方案称为**4维5级脉冲振幅调制**（four-dimensional five-level pulse amplitude modulation，4D-PAM5）。4D表示数据同时通过4条线路发送。它使用5个电平：−2、−1、0、1和2。但是电平0只用于发送差错检测（在第10章讨论）。如果假定编码是一维的，4个电平产生类似于8B4Q方案的编码。换言之，一个8位码字转换成一个4电平的信号元素，这个假想一维版本的最差信号速率是$N\times 4/8$或$N/2$。

设计4D-PAM5技术用来通过4个通道（4条线路）发送数据。这意味着信号速率可以降低到$N/8$，大幅度的改进。所有8个位会同时进入一条线路通过使用1个信号元素发送。关键点在于4个信号元素组成1个信号组，并通过四维环境同时发送。图4.12给出了假想的一维和实际的四维实现。吉位LAN（见第13章）是用这个技术来通过4条铜线电缆（能处理125Mbd）发送1Gbps数据。这个方案的信号模式有很大的冗余。因为2^8个数据模式匹配$4^4=256$个信号模式。额外的信号模式可以用于诸如差错检测等其他目的。

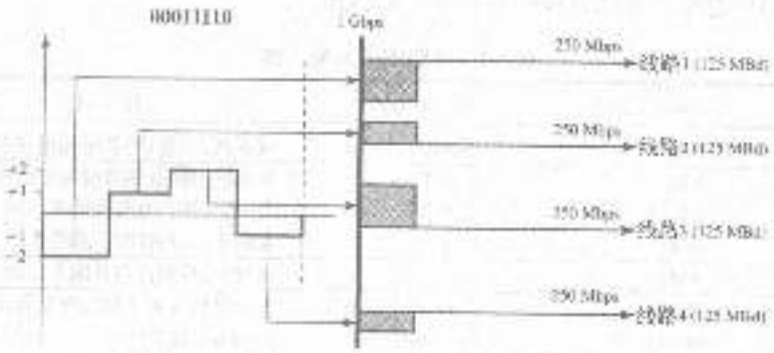


图4.12 多电平：4D-PAM5方案

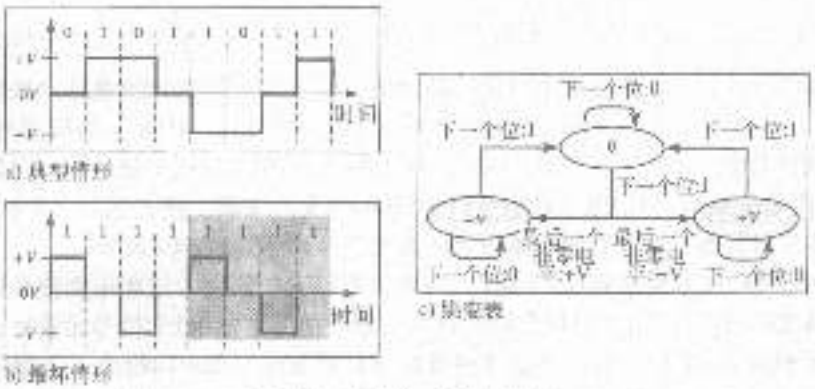
多线路传输：MLT-3

NRZ-I和差分曼彻斯特编码是差分编码，即使用两种不同的跳变规则来编码二进制数据（没有反相、有反相）。如果信号的电平多于两个，可以设计一种多于两个跳变规则的差分编码方案。MLT-3就是其中之一。**三电平多线路传输**（multiline transmission，three level，MLT-3）方案使用三个电平（+V、0、−V）和三个跳变规则在电平间变动。

1. 如果下一个位是0，没有跳变。
2. 如果下一个位是1而且当前电平不是0，下一个电平是0。
3. 如果下一个位是1而且当前电平是0，下一个电平是最后一个非零电平的相反值。

图4.13所示的状态图很好地描述了MLT-3。三个状态（用椭圆）表示三个电平（+V、0、−V）。连接线表示一个状态（电平）到另一个状态（电平）的跳变。图4.13还给出了MLT-3信号的两个例子。

有人可能想知道为什么需要使用MLT-3，一个位映射成一个信号元素的方案。信号速率和NRZ-I一样，但是更复杂（三个电平和复杂的跳变规则）。看起来这个方案中信号形状帮助降低了所需的带宽。让我们看一看最坏的情形，全1的序列。每4个位就重复信号元素模式+V0 −V0。非周期信号已经变成周期信号，周期等于4倍位持续时间。这个最坏情形可以模拟成频率为1/4比特率的模拟信号。当我们需要在铜线电缆（不能支持高于32MHz，高于这个电平的频率会产生电磁发射）上发送100Mbps时，MLT-3是个合适的选择。MLT-3和LAN会在第13章中讨论。



c) 状态表

图4.13　多信道：MLT-3方案

线路编码方案小结

我们在表4.1中总结了前面讨论的各种方案的特性。

表4.1　线路编码方案小结

类　别	方　案	带宽（平均）	特　性
单极	NRZ	$B=N/2$	基本的。0或1的长序列没有自同步，DC
单极	NRZ-I	$B=N/2$	全0或全1的长序列没有自同步，DC
	NRZ-I	$B=N/2$	全0的长序列没有自同步，DC
	双相	$B=N$	自同步，没有DC，高带宽
双极	AMI	$B=N/2$	全0的长序列没有自同步，DC
多电平	2B1Q	$B=N/4$	相同双位长序列没有自同步
	8B6T	$B=3N/4$	自同步，没有DC
	4D-PAM5	$B=N/8$	自同步，没有DC
多线路	MLT-3	$B=N/3$	全0的长序列没有自同步

4.1.3　块编码

我们需要某种冗余以确保同步，并提供一些内在的差错检测。块编码可以给出这种冗余并提高线路编码的性能。一般，块编码（block coding）把m位的块变成n位的块，这里n大于m。块编码称为mB/nB编码技术。

> 块编码通常称为mB/nB编码技术，它用n位组替换m位组。

块编码中的斜线（比如4B/5B）用来区别于多电平编码（比如8B6T），多电平编码中没有斜线。块编码一般涉及三个步骤：分组、置换和组合。在分组步骤中，位序列划分成许多个m位组。例如，在4B/5B编码中，原始的位序列划分成4位组。块编码的核心是置换步骤。在这个步骤，用n位组置换m位组。例如在4B/5B编码中，用5位组置换4位组。最后5位组组合形成流。新的流比原始的流有更多的位。图4.14说明了这个过程。

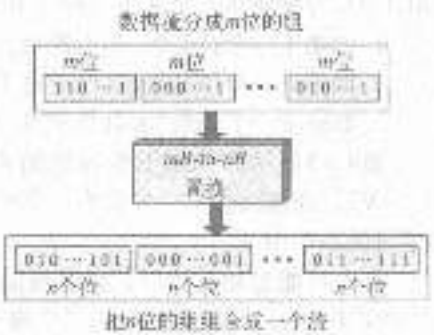


图4.14　块编码概念

4B/5B

4个二元/5个二元（four binary/five binary，4B/5B）编码方案设计出来和NRZ-I组合使用。

回忆一下，NRZ-I有很好的信号速率，它是双相编码的一半，但是它有同步问题。连续的0的长序列会让接收方失去同步，一种解决方法是在使用NRZ-I编码前改变位流。4B/5B方案实现了这个目的。我们后面会看到，经过块编码的流不会有多于三个连续的0。在接收方，NRZ-I编码后的数字信号先解码成位流，然后去掉冗余。图4.15说明了这个思想。

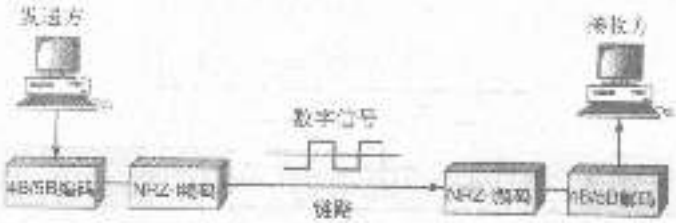


图4.15 使用块编码4B/5B和线路编码NRZ-I的组合方案

在4B/5B编码中，将4位输入置换为5位输出以包含不超过一个前导0（左边的位）和两个后缀0（右边的位）。因此当不同的组组合形成新的序列时，最多只会有二个连续的0。（注意NRZ-I中1的长序列没有问题。）表4.2给出了用于4B/5B编码的相应对。注意前两列是4位组和5位组。4位组只有16个不同的组合，而5位组有32个不同的组合。这意味着有16个组没有用于4B/5B编码。这些不用的组当中，一些用于控制，另一些仍然没有使用，可以提供一些差错检测。如果收到的5位组属于表中没有使用的部分，接收方知道传输中出现了差错。

表4.2 4B/5B映射码

数据序列	编码后的序列	控制序列	编码后的序列
0000	11110	Q（错误）	00000
0001	01001	I（空闲）	11111
0010	10100	H（停止）	00100
0011	10101	J（开始定界符）	11000
0100	01010	K（开始定界符）	10001
0101	01011	T（结束定界符）	01101
0110	01110	S（设置）	11001
0111	01111	R（重置）	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

图4.16给出了用4B/5B编码方案置换的一个例子。4B/5B编码方案解决了同步问题，克服了NRZ-I的缺陷，但是，我们需要记住它增加了NRZ-I的信号速率。冗余位增加20%的波特。结果仍然小于两相方案，它的信号速率是NRZ-I的两倍。但是，4B/5B编码方案没有解决NRZ-I的DC成分问题，如果DC成分是不可接受的，就需要使用双相或双极编码方案。

例4.5

我们需要以1Mbps速率发送数据。使用4B/5B和NRZ-I组合，或4B/5B和曼彻斯特编码组合，最小的带宽是多少？

解：

首先4B/5B块编码增加比特率到1.25Mbps。使用NRZ-I的最小带宽是N/2或625kHz。曼彻斯特方案需要最小带宽1MHz。第一个选择需要更低的带宽，但有DC成分问题；第二个选择需要更高的带宽，但没有DC成分问题。

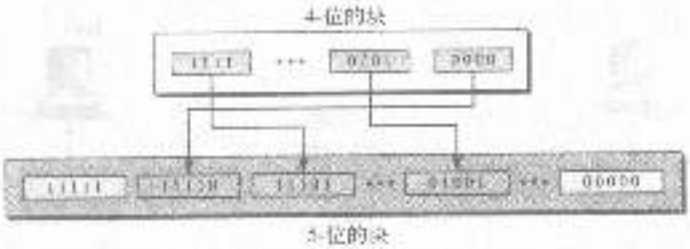


图4.16　使用4B/5B编码方案进行置换

8B/10B

8个二元/10个二元（eight binary/ten binary，8B/10B）编码方案类似于4B/5B编码方案，除了它是8位数据组被置换成10位编码。比起4B/5B，它提供了更强的差错检测能力。8B/10B块编码实际上是5B/6B编码和3B/4B编码的组合，如图4.17所示。

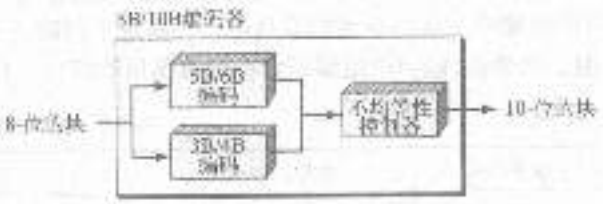


图4.17　8B/10B 编码

8位块的高5位送入5B/6B编码器，低3位送入3B/4B编码器。这样做是为了简化映射表。为了防止连续0或1的长串，编码使用不均等性控制器观察0比1多（或1比0多）。如果当前块中的位产生了不均等性，而该不均等性对上一块的不均性有影响（任一方向），那么当前块的所有位取反（0变成1而1变成0）。这种编码有$2^{10}-2^8=768$个冗余组，可以用于不均等性校验和差错检测。一般，这种技术由于有更好的内置差错检查能力来和同步能力而优于4B/5B。

4.1.4　扰动

两相编码方案适用于LAN中站间的专用链路，并不适用于长距离通信（由于宽带宽需求）。块编码和NRZ线路编码的组合由于DC成分也不适用于长距离编码。双极AMI编码有窄带宽而且不会产生DC成分。但是，连续0的长序列会不同步。如果能找到避免初始流中连续0长序列的方法，那么双极AMI可以用于长距离。我们正寻找一种不会增长位数并提供同步的技术，以及一种把0电平脉冲置换成其他电平组合来提供同步的解决方法。一种解决方法称为扰动（scrambling）。我们修改了部分AMI规则引入了扰动，如图4.18所示。注意：相对于块编码，扰动与编码同时完成。系统需要基于定义好的扰动规则插入所需脉冲。两种通用扰动技术是B8ZS和HDB3。

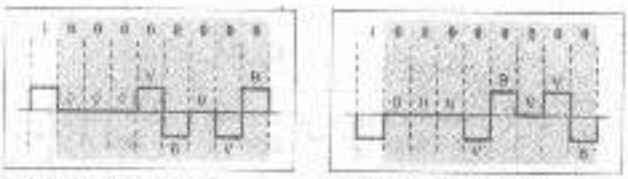


图4.18　使用扰动的AMI

B8ZS

8零置换的双极编码方案（bipolar with 8-zero substitution，B8ZS）通常用于北美地区。在这个技术中，8个连续0电平会被替换成

000VB0VB，这里的V表示违反。这个非零电平违反了AMI编码规则（前一极的相反极），这里的B表示双极，表示与AMI规则相一致的非零电平，有两种情形，如图4.19所示。



a) 前一个非零电平是正电平　　b) 前一个电平是负电平

图4.19 B8ZS扰动技术的两种情形

注意：这里的扰动不会改变比特率，而且这个技术平衡了正电平和负电平（两个正电平和两个负电平），这意味着维持了DC平衡。注意置换可能会改变1的极性，因为置换后AMI需要遵守它的规则。

> B8ZS把8个连续零置换成000VB0VB。

还要提到一点，这里的字母L（违反）或B（双极）是相对的，V表示与前一个非零脉冲极性相同的极性，B表示与与前一个非零脉冲极性相反的极性。

HDB3

高密度双极3零方案（high-density bipolar 3-zero，HDB3）通常用于北美以外地区。这种技术比B8ZS技术更加保守，四个连续零电平被置换成000V或B00V。两个不同置换的原因是为了维持每次置换后非零脉冲为偶数。这两个规则可以定义如下：

1. 如果最后一次置换后的非零脉冲数是奇数，置换模式是000V，这样使得非零脉冲总数为偶数。

2. 如果最后一次置换后的非零脉冲数是偶数，置换模式是B00V，这样使得非零脉冲总数为偶数。

图4.20给出了一个实例。

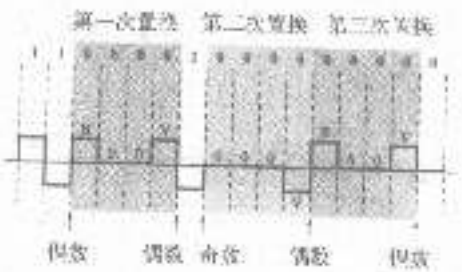


图4.20 HDB3扰动技术中的不同情况

这里还需要提到几点。第一，在第一次置换前，非零脉冲数是偶数，所以第一次置换是B00V。这次置换后，位1的极性改变，因为每次置换后必须遵循AMI方案自己的规则。在这个位后，我们需要另一次置换，这次置换是000V，因为在最后一次置换后只有一个非零脉冲（奇数），第三次置换是B00V，因为在第二次置换（偶数）后没有非零脉冲。

> HDB3技术根据最后一次置换后非零脉冲数把四个连续零置换成000V或B00V。

4.2 模拟到数字转换

4.1节描述的技术用于把数字数据转换成数字信号。然而，有时我们只有模拟信号，诸如麦克风或照相机产生的信号。在第3章，我们已经看到数字信号优于模拟信号。今天的趋势是把模拟信号转换成数字信号。本节将描述两种技术：脉冲码调制和delta调制。在数字数据产生（数字化）后，我们使用4.1节描述的一种技术把数字数据转换成数字信号。

4.2.1 脉冲码调制（PCM）

把模拟信号转换成数字信号（数字化）的最通用技术称为脉冲码调制（pulse code

modulation，PCM）。PCM编码器有三个过程，如图4.21所示。

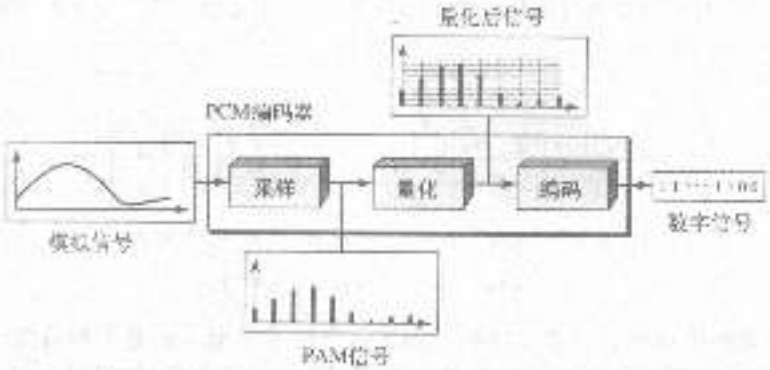


图4.21　PCM编码器组成

1. 对模拟信号进行采样；
2. 对采样后信号进行量化；
3. 量化后的值编码成位流。

采样

PCM的第一步是采样（sampling），每隔T_s秒对模拟信号进行采样，这里T_s是样本间隔或周期。采样间隔的倒数称为**采样率**（sampling rate）或**采样频率**（sampling frequency），定义成f_s，这里$f_s = 1/T_s$。有三种采样方法：理想采样，自然采样和方顶采样，如图4.22所示。

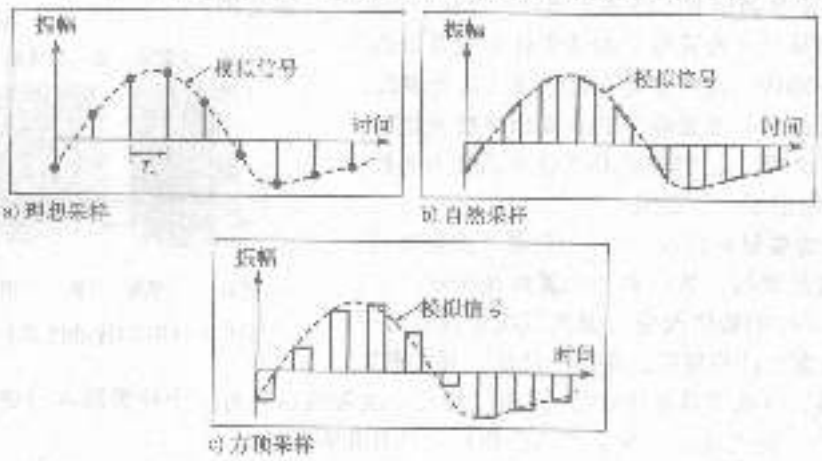


图4.22　PCM的三种不同采样方法

在理想采样中，对来自模拟信号的脉冲进行采样，这是理想采样方法，不能容易实现。在自然采样中，当采样发生时高速开关开启很短的时间，结果是样本序列保持了模拟信号的形状。最常用的采样方法称为**采样和保持技术**（sample and hold），然而它通过通用电路产生方顶样本。

采样过程有时称为**脉冲振幅调制**（pulse amplitude modulation，PAM），但是我们需要记住，结果仍然是非完整的模拟信号。

采样速率　一个重要考虑是采样速率或采样频率。对T_s的限制是什么？这个问题由奈奎斯特给出了完备的解决。根据**奈奎斯特定理**（Nyquist theorem），为了再生原始模拟信号，一个必要条件是**采样速率**（sampling rate）至少是原始信号中最高频率的两倍。

根据奈奎斯特定理，采样速率必须至少是信号所含最高频率的2倍。

我们需要在这里详细描述这个理论。首先，只有信号带宽是有限的时，我们才能对它进行采样。换言之，我们无法对无限带宽的信号进行采样。其次，采样率必须至少是最高频率而不是带宽的两倍。如果模拟信号是低通的，带宽和最高频率有相同的值。如果模拟信号是带通的，带宽值小于最大频率值。图4.23说明了两种类型信号的采样率值。

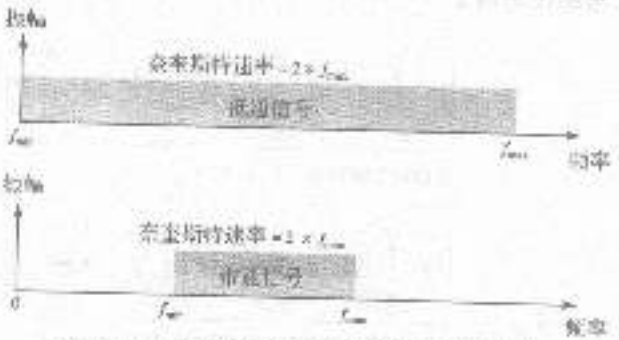


图4.23 低通和带通信号的奈奎斯特采样速率

例4.6

对于奈奎斯特定理的一个直观实例，让我们以三种采样率对简单正弦波进行采样：$f_s = 4f$（2倍奈奎斯特速率），$f_s = 2f$（奈奎斯特速率）和$f_s = f$（一半奈奎斯特速率）。图4.24给出了采样和后续的信号恢复。

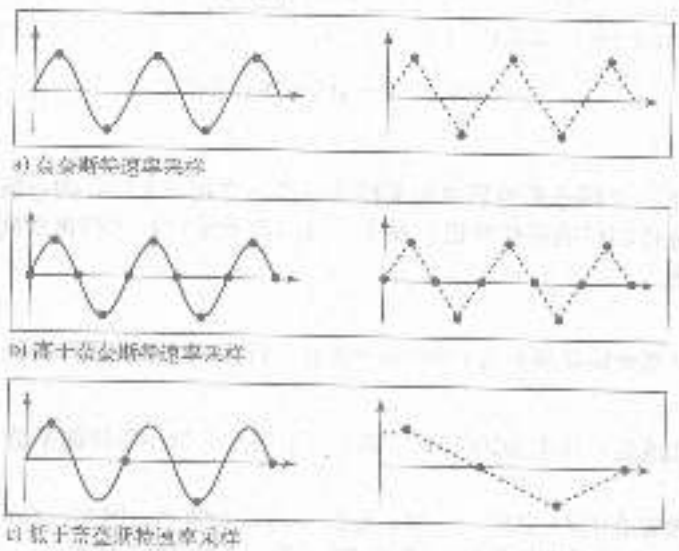


图4.24 不同采样率下采样正弦波的恢复

可以看到，以奈奎斯特速率进行采样可以得到与原始正弦波较好近似的信号（图4.24a部分），图4.24b部分过采样得到相同的近似，但它是冗余的，没必要。低于奈奎斯特率的采样（图4.24c部分）不能产生与原始正弦波相似的信号。

例4.7

一个有趣的例子，如果我们对读如时钟指针旋转的周期性事件进行采样，我们会看到什么？时钟的分针周期为60秒，根据奈奎斯特定理，我们需要每隔20秒（$T_s = \frac{1}{2}T$ 或 $f_s = 2f$）对分针进行采样（拍照并发送）。在图4.25a中，样本点依次为12、6、12、6、12和6，样本的接收方无法知道时钟是向前走还是向后走。在图4.24b部分，我们以两倍奈奎斯特率（每隔15秒）进行采样。样本点依次是12、3、6、9和12。时针在向前走。在图4.24c部分，我们以低于奈

奈斯特率的采样率 $\left(T_s = \frac{3}{4}T \text{ 或 } f_s = \frac{4}{3}f \right)$ 进行采样。样本点依次是12、9、6、3和12，虽然时钟在向前走，但接收方认为时钟在向后走。

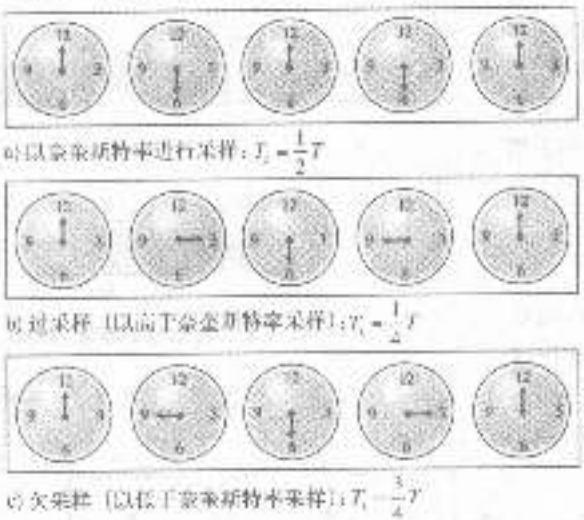


图4.25　对时钟分针进行采样

例4.8

与例4.7相似的一个例子是电影中向前移动的汽车看起来车轮在向后旋转。这可以用欠采样解释。电影以每秒24帧的速度拍摄。如果车轮以高于每秒12次的速度旋转，欠采样就会产生向后旋转的印象。

例4.9

电话公司假定语音最高频率为4 000Hz对语音进行采样。因此采样率是每秒8 000个样本。

例4.10

一个复杂的低通信号带宽为200kHz。那么这个信号的最小采样速率是多少？

解：

低通信号的带宽在0和f之间，这里f是信号的最大频率，因此，我们以两倍这个最高频率（200 kHz）对这个信号进行采样。采样速率是每秒400 000个样本。

例4.11

一个复杂的带通信号带宽为200kHz，那么这个信号的最小采样率是多少？

解：

因为我们不知道带宽从何开始和到何结束，所以我们无法找到这个例子中的最小采样率。我们不知道这个信号中的最大频率。

量化

采样后的结果是一系列振幅值介于信号最大振幅和最小振幅间的脉冲。振幅集可能是无限个介于两个限制值间的非整数值。这些值无法用于编码过程。以下是量化步骤：

1. 我们假定原始模拟信号有介于V_{min}和V_{max}的瞬时振幅。

2. 我们把范围分成L个区间，每个区间高度为Δ（delta）。

$$\Delta = \frac{V_{max} - V_{min}}{L}$$

3. 我们分配0到$L-1$的量化值给每个区间的中点。

4. 样本振幅值近似为量化值。

一个简单的例子，假定我们有一个采样信号，并且样本振幅在−20V到20V范围内。我们决定有8个电平（$L=8$）。这意味着$\Delta = 5$V。图4.26说明了这个例子。

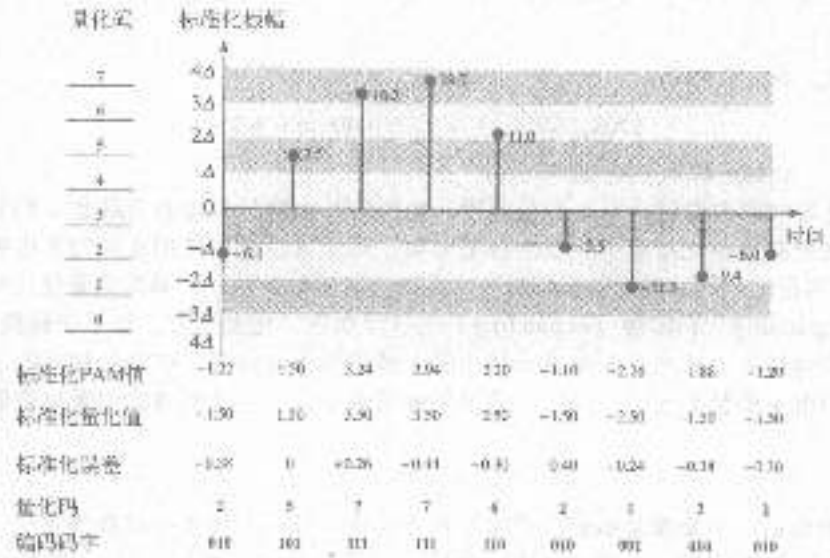


图4.26 采样信号的量化和编码

我们使用理想采样（为了简单）只给出了9个样本。图中每个样本顶端的值给出了实际振幅。在表中，第一行是每个样本标准化后的值（实际振幅/Δ）。量化过程从每个区间的中间选择量化值。这表示标准化量化值（第二行）不同于标准化振幅。两者之差称为标准化误差（第三行）。第四行是基于图左边量化电平的每个样本的量化码。编码码字（第五行）是转换的最后结果。

量化等级 在前面例子中，我们给出了八个量化等级。L（等级数）的选择取决于模拟信号振幅范围以及我们需要准确恢复信号的程度。如果一个信号的振幅只在两个值之间变动，我们只需要两个电平；如果信号（像语音）有许多振幅值，我们需要更多的量化等级。在音频数字化中，L通常为256，在视频中，L一般是几千。如果信号的幅度变化很大，较低的L值就会增加量化误差。

量化误差 一个重要的问题是量化过程产生的误差。（后面我们会看到这个问题如何影响高速调制解调器）。量化是一个近似过程。量化器的输入值是实际值，输出值是近似值。输出值为区间的中间值。如果输入值也在区间中间，就没有量化误差，否则就会有误差。在前面的例子中，第二个例子的标准化振幅是3.24，但标准化量化值是3.50。这表示误差是+0.26。任一样本的误差值都小于$\Delta/2$。换言之，$-\Delta/2 \leq$ 误差 $\leq \Delta/2$。

量化误差改变了信号的信噪比，根据香农理论这反过来减小了上限容量。

可以证明量化误差（quantization error）对信号SNR_{dB}的影响取决于量化级别L或每个样本位数n_b，如下面公式所示：

$$SNR_{dB} = 6.02n_b + 1.76 \text{ dB}$$

例4.12

图4.26例子中的SNR_{dB}是多少？

解：

我们可以使用公式来得出这个量化值。有8个级别，每个样本有3个位，所以$SNR_{dB} = 6.02(3) + 1.76 = 19.82$ dB。增加级别数就会增加SNR。

例4.13

电话用户线路必须有高于40的SNR_{dB}。那么每个样本的最小位数是多少？

解：

我们可以计算位数如下：

$$SNR_{dB} = 6.02n_b + 1.76 = 40 \quad n = 6.35$$

电话公司一般每个样本分配7或8个位。

均匀量化与非均匀量化　对于许多应用，模拟信号中瞬时振幅的贡献是不均匀的。振幅的变化一般在低频率中比高频率中发生得更频繁。对于这类应用使用非均匀量化更好。换言之，Δ的高度不是固定的，接近低振幅时较大而接近高振幅时较小。非均匀量化还可以使用称为压缩（companding）和扩展（expanding）的过程实现。在发送方，信号在转换前先压缩，在接收方，在转换前先扩展。压缩表示减小较大瞬时电平振幅值，扩展是相反的过程。压缩使得强信号的电平衡量更大，而弱信号的电平衡量更小。已经证明非均匀量化有效地减小了量化SNR_{dB}。

编码

PCM中的最后一个步骤是编码。在每个样本量化并且每个样本的位数确定后，每个样本可以转换成n_b个位的码字。在图4.26中，编码后的码字显示在最后一行。2的量化码编码成010，5编码成101等。注意每个样本的位数取决于量化等级数。如果量化等级数是L，位数是$n_b=\log_2 L$。在我们的例子中，L是8，所以n_b是3。比特率可以通过如下公式获得：

$$\text{比特率} = \text{采样速率} \times \text{每个样本位数} = f_s \times n_b$$

例4.14

要数字化人的语音。假定每个样本有8个位。那么比特率是多少？

解：

人的语音通常包含0到4 000Hz的频率。因此采样率和比特率可以计算如下：

$$\text{采样率} = 4\,000 \times 2 = 8\,000 \text{样本/秒}$$
$$\text{比特率} = 8\,000 \times 8 = 64\,000 \text{bps} = 64 \text{kbps}$$

原来信号恢复

原来信号的恢复需要PCM解码器。解码器先使用电路把码字转换成保持下一个脉冲前振幅的脉冲，当阶梯信号完成后，它经过一个低通过滤器把阶梯信号平滑成模拟信号。过滤器有同发送方原来信号一样的截断频率。如果信号是以奈奎斯特采样速率采样的，并且有足够的量化等级，那么就可以重新生成原来信号。注意：原来信号的最大值和最小值可以使用放大得到。图4.27显示了简化的过程。

PCM带宽

假设给定低通模拟信号的带宽。如果要数字化这个信号，那么通过这个数字化信号的通道的最小带宽是多少？已经说过线路编码的信号的最小带宽是$B_{min} = c \times N \times (1/r)$。在下面公式中置换$N$值：

$$B_{min} = c \times N \times \frac{1}{r} = c \times n_b \times f_s \times \frac{1}{r} = c \times n_b \times 2 \times B_{analog} \times \frac{1}{r}$$

当1/r=1（对于NRZ或双极信号）和c=（1/2）（平均情况）时，最小带宽是

$$B_{min} = n_s \times B_{analog}$$

这表示数字信号的最小带宽是模拟信号的带宽的n_s倍。这就是数字化付出的代价。

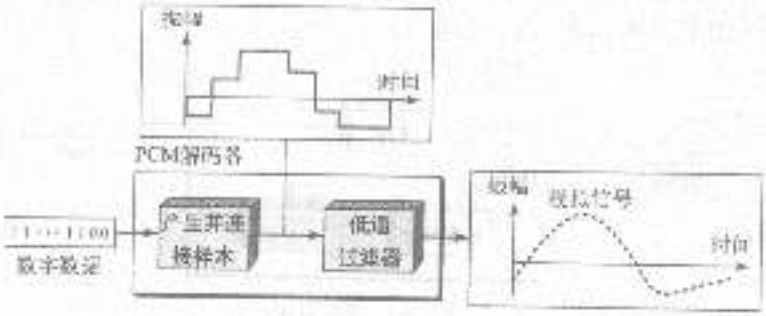


图4.27 PCM解码器的组成

例4.15

有一个4kHz的低通信号。如果发送这个模拟信号，需要最小带宽是4kHz的通道。如果数字化这个信号并且每个样本发送8个位，需要最小带宽是8×4kHz=32kHz的通道。

通道的最大数据率

在第3章中，讨论了奈奎斯特定理，它给出了通道的数据速率为$N_{max}=2\times B\times \log_2 L$。使用下面的论据可以从奈奎斯特采样定理得出这个速率：

1. 假定可用通道是带宽为B的低通通道。
2. 假定要发送的数字信号有L个电平，每个电平是一个信号元素，这表示$r=1/\log_2 L$。
3. 先把数字信号通过低通过滤器截断BHz以上的频率。
4. 把结果信号看作模拟信号，并且以每秒$2\times B$个样本对它进行采样，使用L个电平进行量化。因为信号初始有L个电平，所以额外的量化电平是无用的。
5. 结果比特率是$N=f_s\times n_s=2\times B\times \log_2 L$。这是最大带宽；如果情形五子c变大，数据速率就减小。

$$N_{max} = 2\times B\times \log_2 L \quad \text{bps}$$

最小所需带宽

如果数据速率和信号电平数都是固定的，前面的论据就给出了最小带宽。如下

$$B_{min} = \frac{N}{2\times \log_2 L} \quad \text{Hz}$$

4.2.2 Delta调制（DM）

PCM是十分复杂的技术。已经开发出其他技术来减少PCM的复杂性。最简单的一种是delta调制。PCM得到每个样本的信号振幅值，DM从前一个样本中得到变化。图4.28显示了这个过程，注意这里没有任何码字，位一个接一个被发送。

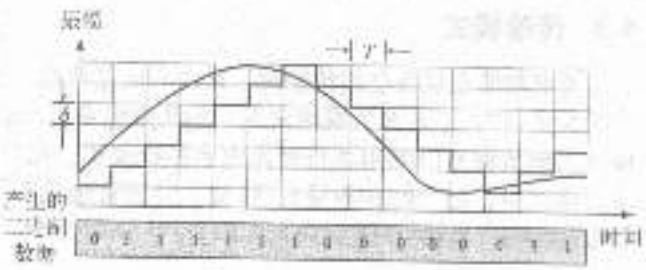


图4.28 delta调制过程

调制器

调制器用在发送方站点，用来从模拟信号中产生位流。这个处理记录了小的正/负改变，称为delta δ。如果delta是正的，就记录成1。如果是负的，就记录成0。但是，需要一个基准供模拟信号比较。调制器生成一个像梯形的第二个信号。发现变化后变为输入信号与逐步生成的梯形信号进行比较。图4.29给出了处理图。

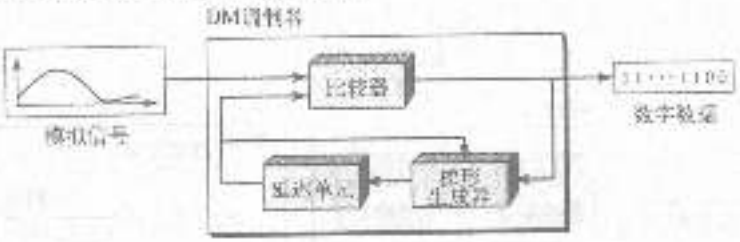


图4.29　delta调制组成

在每个采样区间，调制器将模拟信号的值与梯形信号的最后一个值相比较。如果模拟信号的振幅大，数字数据中的下一个位就是1；否则就是0。但是，比较器的输出能自己生成梯形信号。如果下一个位是1，梯形信号生成器就会把梯形信号的最后一个点向上移δ；如果下一个位是0就向下移δ。注意我们需要一个延迟单元，在两次比较间保持梯形函数。

解调器

解调器得到数字数据，用梯形生成器和延迟单元产生模拟信号。但是产生的模拟信号需要经过一个低通过滤器来平滑信号。图4.30给出了示意图。

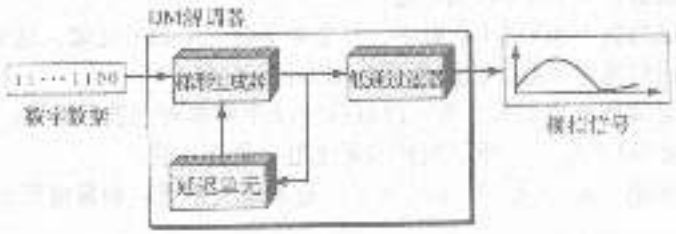


图4.30　delta解调器组成

适配DM

如果δ值不固定就可以得到更好的性能。在适配delta调制（adaptive delta modulation）中，δ根据模拟信号的振幅而改变。

量化误差

显然DM不是完美的，在处理中总是出现量化差错，但是DM的量化误差比PCM的小得多。

4.3　传输模式

考虑数据在设备之间传输时，最关心的是线路，而在考虑线路时，最关心的就是数据流。一次发送1位，还是多位成组发送。如果成组发送，那么如何成组？结果会如何？通过链路传输二进制数据可以采用并行模式或者串行模式。在并行模式中，每个时钟脉冲发送多位。在串行模式中，每个时钟脉冲发送1位。并行传输只有一种方式，串行传输则分为三类：异步、同步和等时（见图4.31）。

4.3.1 并行传输

由1和0构成的二进制数据，可以组成每组为n位的位组。计算机使用和生成以位组为单位的数据，这像在英语会话时用单词而不是一个一个字母来交流一样。通过位组，一次可以发送n位数据而不是1位。这种方式称为**并行传输**（parallel transmission）。

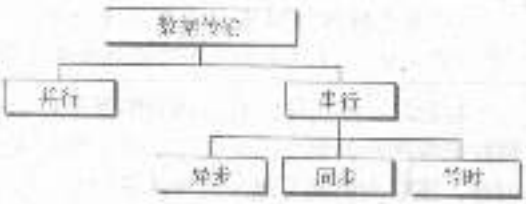


图4.31 数据传输模式

并行传输机制原理上很简单：每次使用n条线路传送n位。这种方式中，每位有自己的线路，一个位组中的所有n位能够在一个时钟脉冲从一个设备传输到另一个设备。图4.32说明了n=8时并行传输的工作情况。通常将8条电线捆成一根电缆，两端各有一个连接头。

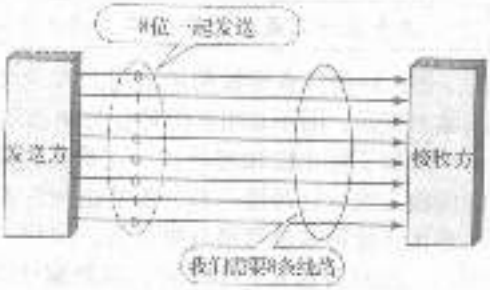


图4.32 并行传输

并行传输的优点是速度。当其他因素相同时，并行传输比串行传输可以提高n倍传输速度。但是也有一个严重缺点，那就是成本高。并行传输需要n条通信线路（在本例中是电线）传输数据流。因为比较昂贵，因此并行传输通常只用于短距离通信。

4.3.2 串行传输

在**串行传输**（serial transmission）中，位是一个一个依次传输的，因此，在两个通信设备之间传输数据只需要一条而不是n条传输通道（见图4.33）。

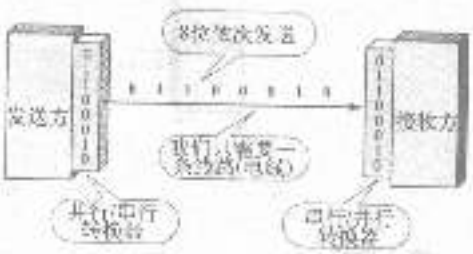


图4.33 串行传输

串行传输与并行传输相比，其优点是只需要一条通信通道，串行传输的成本大约只是并行传输的$1/n$，因为在设备内部是并行的，所以在发送方与线路之间（并行/串行转换）以及接收方与线路之间（串行/并行转换）的接口都需要转换设备。

串行传输可以使用三种方式之一：异步、同步和等时。

异步传输

如果在传输中信号的时序并不重要，将这种传输称为**异步传输**（asynchronous transmission）。信息的接收和转换通过约定的模式进行，只要遵循约定模式，接收设备就可以不管信息发送的节奏而正确地获得信息。约定模式基于将位流组成字节的方式建立的，每一个组（通常是8位）作为一个单位沿着链路传送。发送系统独立处理每个分组，每处理完一个组就将它转发到链路上而不提供时钟信号。

没有同步时钟，接收方就不能通过计时方式预测下一个新组何时到达。所以为了通知接收方有一个新的组到达，在每一个字节的开始处增加一个位，这一位（通常是0）称为**起始位**（start bit）。为了让接收方知道字节已经结束，在字节的结束位置会增加一个或者多个位，这些位（通常是1）称为**停止位**（stop bit）。通过这种方法，每一个字节至少要增加到10位，其中8位是信息，2位或者更多位是给接收方的提示信号。此外，每发送完一个字节，可能还要跟上

一段可变长的时间间隔。这段间隔或者通过空闲状态表示，或者通过附加的停止位流表示。

在异步传输过程中，需要在每个字节开始时发送1个起始位（0），结束时发送1个或者多个停止位（1）。在每个字节之间会有一个时间间隔。

起始位、停止位以及时间间隔用于提示接收方每一个字节的开始和结束，使接收方可以根据数据流进行同步。因为在字节这一级，发送方和接收方不需要进行同步。但是在每一个字节内部，接收方的要与输入的位流进行同步，这种机制称为异步。也就是说，一定程度上的同步还是存在的，但是只局限在单个字节内。每个新字节开始时，接收设备会重新进行同步。当接收方检测到起始位后，它会启动计时器，随着位的到来开始计数。在接收完8个位后，接收方就等待停止位到达。一旦检测到停止位，接收方在下一个起始位到达前忽视接收到的所有信号。

异步在这里是指在字节级上的异步，但是每位仍然是同步，它们的持续时间是相同的。

图4.34是异步传输的示意图，在这个例子中，起始位是0，停止位是1，间隔用线路的空闲状态表示，而不是附加的停止位来表示。

附加了停止位和起始位，以及在位流中插入间隙，使得异步传输要比没有附加控制信息的传输方式运行得慢一些，但是这种方式既便宜又有效，对于某些情况（如低速通信），它的两种优点使得其显得很有吸引力。例如，键盘到计算机的连接自然是异步传输的一种的应用。用户一次只会输入一个字符，从数据处理方面来说，输入是非常慢的，并且每个字符之间留有不可预计的时间间隙。

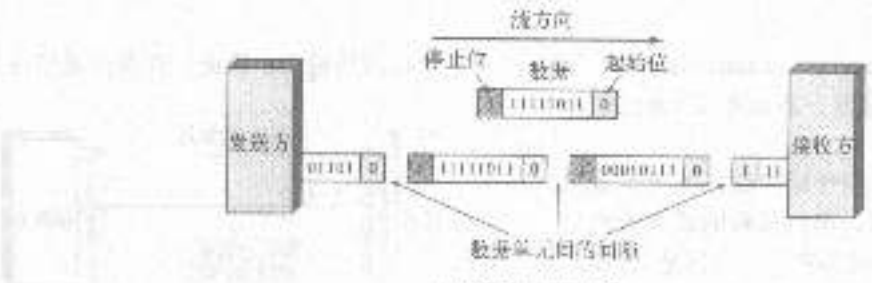


图4.34　异步传输

同步传输

在同步传输（synchronous transmission）中，位流被组合为更长的“帧”，一帧包含了多个字节。引入帧内的各字节之间没有间隙。接收方必须在解码时将位流分解为字节。换句话说，数据以一种不间断的1和0位串进行传输，接收方将位流分割成重构信息所需的一个个字节。

在同步传输模式中，依次发送位流而不含起始位、停止位和间隙。接收方负责将位进行分组。

图4.35给出了同步传输的示意图。在字节之间画出了分隔线，实际上，这些分隔线是不存在的，发送方将数据作为一个长的串发送到线路上。如果发送方希望在单独的一段一段地发送数据，则每段数据之间的间隙必须由意味着空闲状态的0和1序列填充。接收方将在位到达时计数，并将它们重组为字节形式。

因为没有时隙和起始位、停止位，就没有位流内部的同步机制来帮助接收设备在处理位流时调整位同步。由于接收到的信息的准确性完全取决于接收设备根据接收到的位的准确计数能力，所以计时变得非常重要。



图4.35 同步传输

同步传输的优点是速度快，因为发送方没有引入额外的位或者可除并且接收方也不需要将它们去掉，即通过链路传送的位更少，所以同步传输比异步传输要快。由于这个原因，它在高速应用当中更加有用，例如计算机之间的数据传输。字节同步在数据链路层完成。

我们需要在这着重说明一点，虽然在以同步串行传输的字符间没有间隔，但是帧间可能有不等的间隔。

等时

在实时音频和视频中，帧间的不等延迟是不可接受的，例如，TV图像以每秒30个图像的速率广播，它们必须看成一样的速率，如果一个图像是用一个或多个帧发送，帧间应该没有延迟。对于这种应用，字符间的同步是不够的，整个流必须同步。**等时传输**（isochronous transmission）保证了数据以固定速率到达。

推荐读物

为了更详细地讨论与本章有关的主题，推荐下列读物与网站，在本书末列出了括号[……]内的参考资料。

读物

数字信号到数字信号转换的讨论有[Pea92]的第7章，[Cou01]的第3章，[Sta04]的5.1节。采样讨论有[Pea92]的第15，16，17和18章，[Cou01]的第3章，[Sta04]的5.3节。[Hsu03]有调制和采样的数学方法。更深入的资料可参考[Ber96]。

本章小结

- 数字到数字转换有三种技术：线路编码、块编码和扰动。
- 线路编码是将数字数据转换成数字信号的处理过程。
- 线路编码大致可划分五个大类：单极性的、极性的、双极性的、多电平的和多跳变的。
- 块编码提供冗余以保证同步和内部差错检测，块编码一般是mB/nB编码，用n位组取代m位组。
- 扰动是在不增加位的个数时提供同步，两个最常用的扰动是B8ZS和HDB3。
- 将模拟信号变成数字数据（数字化）最常用的技术是脉码调制（PAM）。
- PAM的第一步是采样，每T_s秒采样一次模拟信号，此处T_s是采样间隔或周期。采样周期的倒数称为采样速率或采样频率记为f_s，此处$f_s=\frac{1}{T_s}$。采样有三种方法：理想的、自然的和方顶的。
- 按照奈奎斯特定理，要恢复原始模拟信号，采样速率一个必要条件至少是原始信号最高频率的两倍。
- 为化简PCM的复杂性，已开发出另一些采样技术，其中最简单的是delta调制。PCM是

寻找信号振幅值作为样本，DM是寻找当前样本与前一样本的改变值。

- 并行发送数据仅有一种方法，而串行发送数据有三种方法：异步的、同步的和等时的。
- 在串行异步传输中，发送每个字节加一个起始位（0）和一个或多个停止位（1）。
- 在同步传输中，位以连续的位流传输，没有起始位和停止位，也没有间隙，接收方负责将位重组为有意义的字节。
- 等时方式提供整个位流必须同步，也就是说保证数据以固定的速率到达接收方。

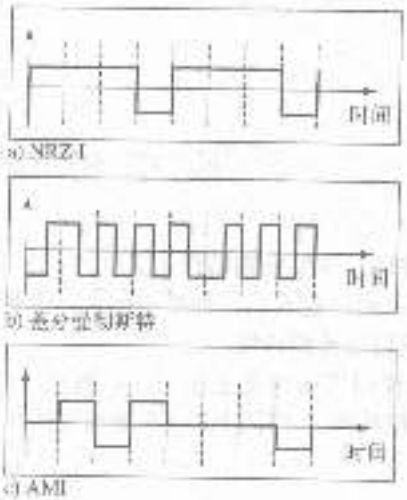
习题

复习题

1. 试列出数字到数字转换的三种技术。
2. 区别信号元素与数据元素。
3. 区别信号速率与数据速率。
4. 说明基线漂移及其对数字传输的影响。
5. 说明DC成分及其对数字传输的影响。
6. 说明具有自同步的信号的特征。
7. 试列出本书所讨论的线路编码方案。
8. 说明块编码及其目的。
9. 说明扰动及其目的。
10. 说明PCM和DM相同和不同之处。
11. 并行传输和串行传输有什么不同？
12. 试列出串行传输的三种方法并说明它们的不同之处。

练习题

13. 如果数据速率是1Mbps和c=1/2，对图4.2中每种情况计算信号速率的值。
14. 在数字传输中，发送方时钟比接收方时钟快0.2%。如果发送方以1Mbps速率发送，问发送方每秒发送多少额外的位？
15. 使用下列数据流的每个画出NRZ-L方案的图，假定最近一个信号电平是正的，从图中用信号电平改变的平均数猜测带宽，并与表4.1的相应的项目进行比较：
    a. 00000000　　b. 11111111
    c. 01010101　　d. 00110011
16. 对NRZ-I方案，重做练习15。
17. 对曼彻斯特方案，重做练习15。
18. 对差分曼彻斯特方案，重做练习15。
19. 对2B1Q方案，用下列数据流，重做练习15。
    a. 0000000000000000　　b. 1111111111111111
    c. 0101010101010101　　d. 0011001100110011
20. 对MLT-3方案，用下列数据流，重做练习15。
    a. 00000000　　b. 11111111
    c. 01010101　　d. 00011000
21. 对图4.36中的每种8位组的编码，求数据流。
22. 一个NRZ-I信号具有100kbps的速率，用图4.6计算在0Hz、50kHz和100kHz处标准化能量（P）。
23. 一个曼彻斯特信号具有100kbps速率，用图4.8计算在0Hz、50kHz和100kHz处标准化能量（P）。



a) NRZ-I

b) 差分曼彻斯特

c) AMI

图4.36 练习21

24. 到4B/5B编码器的输入数据流是0100 0000 0000 0000 0000 0001，试回答下列问题：
    a. 输出数据流是什么？
    b. 在输入中最长的连续0串的长度是多少？
    c. 在输入中最长的连续1串的长度是多少？
25. 在5B/6B编码方案中，有多少无效的（不使用）代码序列？而在3B/4B又有多少？
26. 用下列扰动技术对序列111000000000000进行扰动，引起什么结果？假定最近非零信号电平是正的：
    a.B8ZS
    b.HDB3（最近替换后非零脉冲的个数是奇数）
27. 对下列每个信号奈奎斯特采样速率是多少？
    a. 带宽为200kHz的低通信号；
    b. 带宽为200kHz，最低频率是100kHz带通信号。
28. 对带宽200kHz的低通信号使用1024级采样
    a. 计算数字化信号的速率；
    b. 计算这个信号的SNR_{dB}；
    c. 计算这个信号PCM的带宽。
29. 如果使用4级数字信号，问带宽为200kHz的通道最大速率是多少？
30. 一个模拟信号带宽为20kHz，如果采样该信号并在30kbps通道上发送，问SNR_{dB}是什么？
31. 带宽为1MHz的基带通道，按下列线路编码：
    a. NRZ-L
    b. 曼彻斯特
    c. MLT-3
    d. 2B1Q
    问该通道数据速率是多少？
32. 发送1 000个字符，每个字符编码为8位。
    a. 求同步传输需要多少位？
    b. 求异步传输需要多少位？
    c. 每一情况冗余的百分比？

第5章 模拟传输

第3章已经讨论了数字传输和模拟传输的优点和缺点。我们已经注意到，数字传输有很多好处的，但是需要低通通道。同时也看到，如果实际上只有带通通道，则只能选择模拟传输。第4章讨论了数字传输，本章讨论模拟传输。

将数字数据转换为带通模拟信号传统上称为数字到模拟转换。将低通模拟信号转换为带通信号传统上称为模拟到模拟转换。本章讨论这两种类型转换。

5.1 数字到模拟转换

数字到模拟转换（digital-to-analog conversion）是指根据数字数据中的信息而改变模拟信号的某种特性的过程。图5.1说明了数字信息、数模调制过程和最终模拟信号之间的关系。

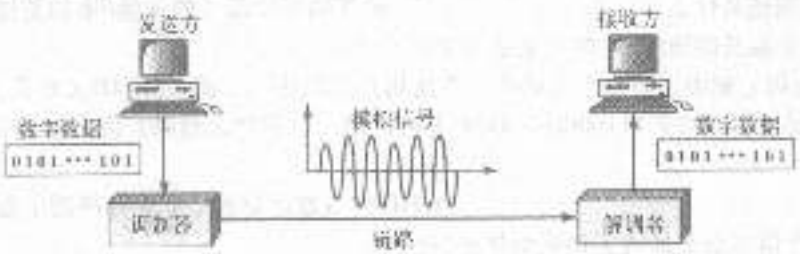


图5.1 数字到模拟转换

如第3章所述，一个正弦波可通过三个特性定义：振幅、频率和相位。当我们改变其中任意一个，就有了波的另一种形式。所以，改变简单正弦波的某一特性，就可以用它来表示数字数据。波的三个特性中的任意何一个都可以用这种方式改变。从而使我们至少有二种将数字数据调制到模拟信号的机制：**幅移键控**（amplitude shift keying，ASK）、**频移键控**（frequency shift keying，FSK）和**相移键控**（phase shift keying，PSK）。另外，还有第四种（更好的）将振幅和相位的变化结合起来的机制，称为**正交振幅调制**（quadrature amplitude modulation，QAM）。其中正交调幅是效率最高的，也是目前调制解调器普遍采用的技术（见图5.2）。



图5.2 数字到模拟转换类型

5.1.1 数模转换问题

在具体讨论数模调制方法之前，有两个基本问题要说明：比特率和波特率以及载波信号。

数据元素和信号元素

在第4章中已讲过数据元素和信号元素的概念。数据元素定义成一块信息的最小实体，即位。而信号元素定义成信号的最小单元，它们是固定的。虽然在本章中继续使用相同的术语，但是我们会看到信号元素与模拟传输中有点不一样。

比特率和波特率

我们可以像数字传输中一样定义数据速率（比特率）和信号速率（波特率）。两者的关系定义如下：

$$S = N \times \frac{1}{r} \text{波特}$$

这里N是数据速率（bps），r是一个信号元素携带的数据元素个数。模拟传输中r值为$r = \log_2 L$，这里L是信号元素类型，而不是电平个数。为了易于类比使用同一术语。

> 比特率是每秒发送的位数，波特率是每秒发送的信号元素数。在数字数据模拟传输中，波特率小于等于比特率。

这里使用第4章中用于比特率和波特率的相同比喻。在运输中，波特类似车辆，比特类似乘客，每辆车需要最大化乘车人数来减少交通量。

例5.1

模拟信号的每个信号单元运送4位。如果每秒发送1 000个信号单元，试求比特率。

解：

这里$r = 4$，$S = 1\,000$，利用

$$S = N \times \frac{1}{r}$$

可求得$N = S \times r = 1\,000 \times 4 = 4\,000$ bps。

例5.2

一个信号的比特率为8 000bps，波特率为1 000波特。问每个信号元素携带多少个数据元素？需要多少个信号元素？

解：

在这个例子中，$S = 1\,000$，$N = 8\,000$，r和L未知。先得到r值，再得到L值。

$$S = N \times \frac{1}{r} \Rightarrow r = \frac{N}{S} = \frac{8\,000}{1\,000} = 8 \text{位/波特}$$

$$r = \log_2 L \Rightarrow L = 2^r = 2^8 = 256$$

带宽

除FSK外，数字数据的模拟传输所需的带宽与信号速率成正比，而在FSK中还要加上载波信号频率间的差值。我们对每种技术都要讨论其带宽。

载波信号

在模拟传输中，发送设备产生一个高频率信号作为基波来承载信息信号。这个基波就称为**载波信号**（carrier signal）或者**载波频率**。接收设备要将自己的收听频率调整到与所期望的发送设备的载波信号的频率一致，然后数字信息通过改变载波信号的一个或者多个特性（振幅、频率和相位）来调制载波信号。这种形式的改变称为调制（或移动键控）。

5.1.2 幅移键控

在幅移键控方式中，通过改变载波信号的振幅来生成信号元素。只有振幅变化而频率和相位保持不变。

二进制ASK（BASK）

虽然有若干个种信号元素的电平，每一个有不同的振幅，但是ASK常用的信号元素只使用两个电平，这称为二进制幅移键控或开关键控（OOK）。一个信号电平的振幅等值是0，另

一个和载波频率振幅一样。图5.3给出了二进制ASK的概念。

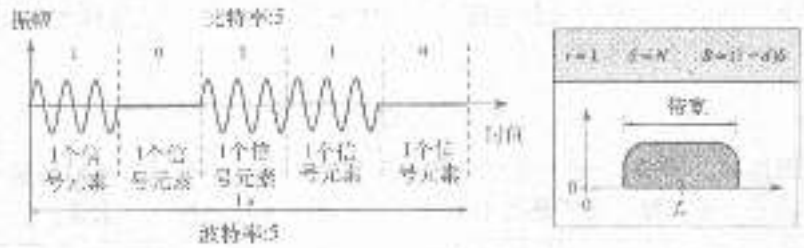


图5.3　二进制幅移键控

ASK带宽　图5.3也给出了ASK的带宽。虽然承载信号只是一个简单正弦波，但调制处理后产生一个非周期性复合信号，正如第3章讨论的，这个信号有一个连续频谱。正如我们期望的，带宽与信号速率（波特率）成正比。但是，通常涉及另一个因子称为d，它取决于调制和过滤处理。d值在0和1之间。这意味着能用如下公式表示带宽。

$$B = (1 + d) \times S$$

这里S是信号速率而B是带宽。

这个公式说明所需带宽最小值是S，最大值是$2S$，最重要的一点是带宽的位置。载波频率f_c位于带宽中间。这意味着如果有可用的带通通道，就可以选择f_c使得调制后的信号能占用该带宽。实际上这是数模转换的最重要优点，可以把发生的带宽平移到可用带宽。

实现　ASK完整实现的讨论超出了本书的范围。但是，实现的简单想法能帮助我们更好地理解概念本身。图5.4说明了如何实现二进制ASK。

如果数字数据表示为一个单极NRZ（见第4章），具有1V高电平和0V低电平的数字信号，那么实现可以通过NRZ数字信号乘以来自振荡器的载波信号获得。当NRZ信号的振幅是1，载波频率的振幅保持不变；如果NRZ信号的振幅是0，载波频率的振幅是零。

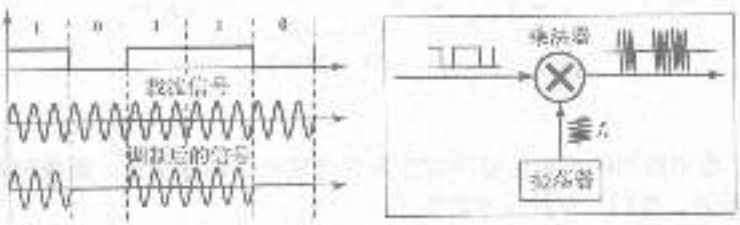


图5.4　二进制ASK实现

例5.3

有100kHz的可用带宽，范围从200到300kHz。如果通过使用$d = 1$的ASK调制数据，那么载波频率和比特率各是多少？

解：

带宽中点是250kHz，这意味着载波频率可以是$f_c = 250$kHz。可以使用带宽的公式得到比特率（$d = 1$，$r = 1$）

$$B = (1 + d) \times S = 2 \times N \times \frac{1}{r} = 2 \times N = 100\text{kHz} \rightarrow N = 50\text{kbps}$$

例5.4

在数据通信中，通常使用双向通信的全双工链路，需要把带宽分成有两部分，每部分各

有一个载波频率，如图5.5所示。图中给出了两个载波频率和带宽的位置。每个方向可用带宽现在是50kHz。因此每个方向的数据速率为25kbps。

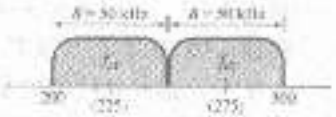


图5.5 例5.4所用的全双工ASK带宽

多电平ASK

上面的讨论只使用两个振幅电平，也可以有多于两个电平。信号可以使用4、8、16或更多的振幅，对应每次使用2、3、4或更多位来调制数据。在这些情况中，$r=2$、$r=3$、$r=4$等，这不是用单纯的ASK实现，而是使用QAM（后面会见到）实现。

5.1.3 频移键控

在频移键控中，通过改变载波信号的频率来表示数据。调制后信号的频率在一个信号元素持续期间是恒定的，但是如果数据元素改变则在下一个信号元素会改变，而所有信号元素振幅峰值和相位保持不变。

二进制FSK（BFSK）

理解二进制FSK（或BFSK）的一个方法是考虑两个载波频率。在图5.6中，已经选择了两个载波频率：f_1和f_2。如果数据元素是0，使用第一个载波，如果数据元素是1则使用第二个载波。但是，注意：这是只用于说明而不是实际的例子。通常这两个载波频率很高，它们的差很小。

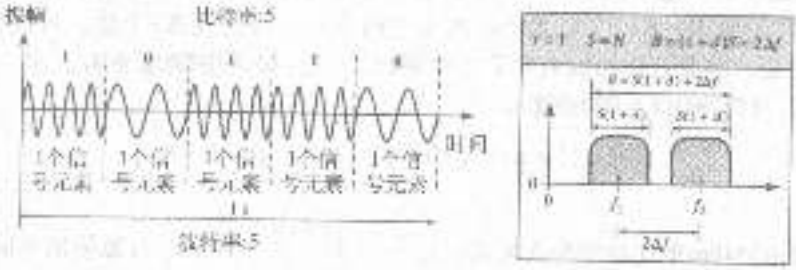


图5.6 二进制频移键控

正如图5.6所示，一个带宽中点的频率是f_c，而另一个带宽中点的频率是f_0。f_1和f_2离这两个带的中点距离都是Δf，则两个频率之差是$2\Delta f$。

BFSK的带宽

图5.6还给出了FSK的带宽。两个载波信号还是简单正弦波，但是调制产生一个具有连续频谱的非周期复合信号。我们可以把FSK看做两个ASK信号，每一个有自己的载波频率（f_1或f_2）。如果两个频率的差是$2\Delta f$，那么BFSK要求的带宽是

$$B=(1+d)\times S+2\Delta f$$

$2\Delta f$的最小值应该是多少？在图5.6中，我们已选择了一个大于$(1+d)S$的值。对于正确的调制解调操作，显然最小值至少应该是S。

例5.5

有一个100kHz的可用带宽，范围从200到300kHz。如果使用$d=1$的FSK调制数据，那么载波频率和比特率应该是多少？

解：

这个问题类似于例5.3，但是我们使用FSK进行调制。频带的中点是在250kHz。选择$2\Delta f$为50kHz，这意味着

$$B = (1+d) \times S + 2\Delta f = 100 \rightarrow 2S = 50\text{kHz} \quad S = 25\text{kbaud} \quad N = 25\text{kbps}$$

对比例5.3，可以看到ASK的比特率是50kbps，而FSK的比特率是25kbps。

实现　BFSK有两种实现方法：非相干和相干。在非相干BFSK中，当一个信号元素结束下一个信号元素开始时相位是不连续。在相干BFSK中，两个信号元素的边界处的相位是连续的。非相干BFSK可以通过把BFSK看作两个ASK调制并使用两个载波频率实现。相干BFSK可以通过使用一个压控振荡器（VCO）实现。压控振荡器是根据输入电平改变频率。图5.7简单说明了BFSK实现。振荡器的输入是单极NRZ信号。当NRZ振幅为零时，振荡器保持它的正常频率；如果NRZ振幅是正的时，增加频率。

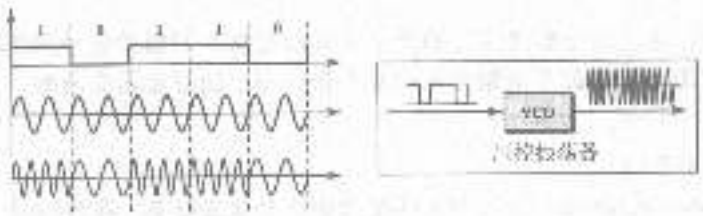


图5.7　BFSK实现

多电平FSK

在FSK方法中，多电平调制（MFSK）是常见的。我们可以使用多于两个频率。例如，可以使用4个不同的频率：f_1、f_2、f_3和f_4每次发送两个位。每次发送3个位，可以使用8个频率。依次类推。但是，需要记住相邻频率需要相隔$2\Delta f$。为了调制解调器正常工作，最小的$2\Delta f$需要是S。可以得出$d = 0$的带宽是

$$B = (1+d) \times S + (L-1)2\Delta f \Rightarrow B = L \times S$$

例5.6

我们需要用3Mbps的比特率每次发送3位。载波频率是10 MHz。计算使用不同频率的个数、波特率和带宽。

解：

使用不同频率的个数$L = 2^3 = 8$。波特率S是3 MHz/3=1 000 Mbaud。这意味着载波频率必须是相隔1MHz（$2\Delta f$ = 1MHz）。带宽是B=8 × 1 000 = 8 000MHz。图5.8给出了频率和带宽的分配。

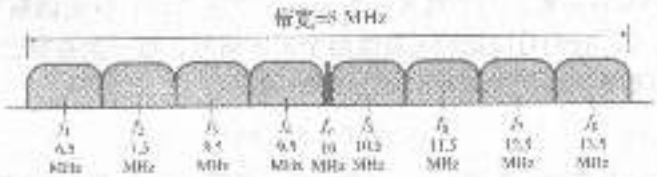


图5.8　用于例5.6的MFSK带宽

5.1.4　相移键控

在相移键控中，通过改变载波的相位来表示两个或更多个不同的信号元素。当相位改变时，峰值振幅和频率保持不变。今天，PSK比ASK或FSK更通用。很快会看到QAM（组合了ASK和FSK）是数模转换中占主导的方法。

二进制PSK（BPSK）

最简单的PSK是BPSK，只用两个信号元素，一个相位是0°，另一个相位是180°。图5.9给

出了PSK的概念描述。二进制PSK和二进制ASK一样简单，但是PSK比起ASK不易受噪声影响。在ASK中，位检测的标准是信号振幅，在PSK中则是相位。换句话说，PSK比ASK更不易受噪声影响。PSK优于FSK是因为不需要两个载波信号。

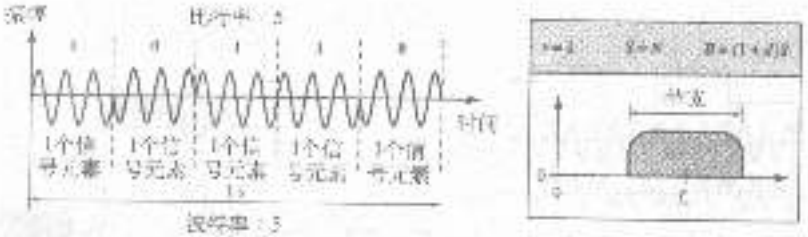


图5.9 二进制相移键控

带宽 图5.9还说明了BPSK的带宽。这个带宽和二进制ASK一样，但比BFSK少。没有浪费带宽用来分离成两个载波信号。

实现 BPSK的实现和ASK一样简单。原因是相位为180°的信号元素可以看成是相位为0°的信号元素的互补波形（基于时间轴对称）。这为如何实现BPSK提供了线索。我们使用用于ASK相同的思路，但这里是极性NRZ信号而不是单极NRZ信号，如图5.10所示。极性NRZ信号乘以载波频率。位1（正电平）表示成相位从0°开始，位0（负电平）表示成相位从180°开始。

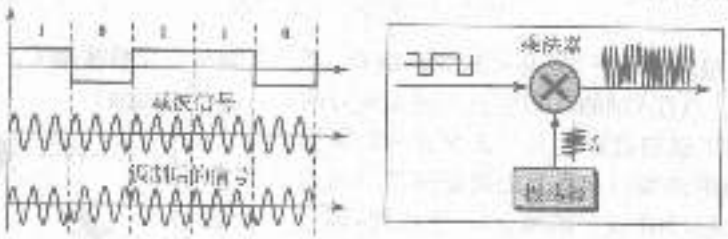


图5.10 BASK实现

正交PSK（QPSK）

BPSK的简单性促使设计者在每个信号元素中一次使用2位，因此减小了波特率和所需带宽。这个方案称为正交PSK或QPSK，因为它使用两个独立的BPSK调制，一个是同相的，另一个是正交的（异相）。进入的2个位先经过串行到并行的转换，它发送一个位给一个调制器，发送下一个位给另一个调制器。如果进入信号中的每个位的持续时间是T，发送相应BPSK信号的每个位持续时间就是$2T$，这意味着每个BPSK信号中位的频率是原始信号的一半。图5.11说明了这个概念。

由每个乘法器产生各自的复合信号是频率相同，但相位不同的正弦波。当这两个复合信号相加时，其结果是另一个正弦波，相位可能是45°、−45°、135°和−135°中之一。输出信号有4种信号元素（$L=4$），使得每个信号元素可以发送2位（$r=2$）。

例5.7

试求使用QPSK以速率12 Mbps传输信号的带宽，设$d=0$。

解：

对于QPSK，每个信号元素携带2位，即$r=2$。因此，信号速率（波特率）是$S=N\times(1/r)=6$M波特。已知$d=0$，有$B=S=6$ MHz。

星座图

星座图（constellation diagram）有助于定义信号元素的振幅和相位。尤其当我们使用类

个载波（一个同相，而另一个正交）时，当处理多电平ASK，PSK或QAM（见下一个节）时，星座图很有用。在星座图中，一个信号元素用一个点表示，它携带的位或位组合一般与在它的旁边。

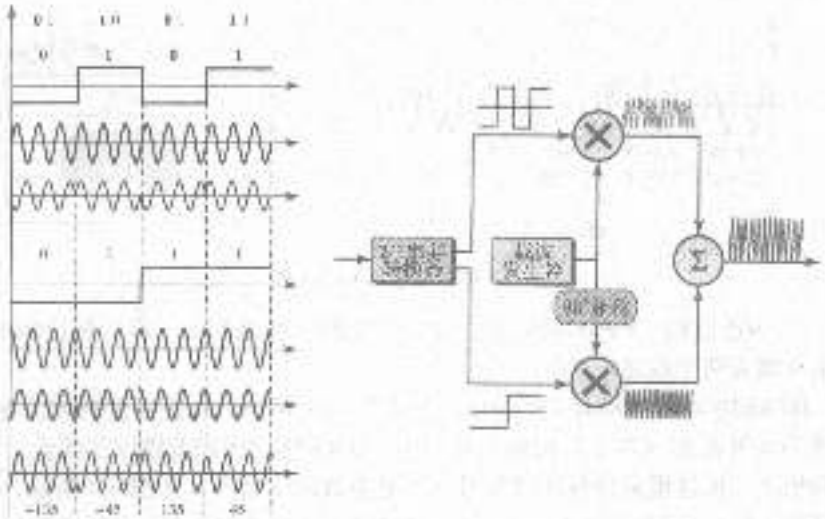


图5.11　QPSK及它的实现

星座图有两根轴，水平X轴与同相载波相关，垂直Y轴与正交载波相关。图中每个点，可以包含4条信息。点在X轴的投影定义了同相成分的峰值振幅，点在Y轴的投影定义了正交成分的峰值振幅。点到原点的连线（向量）长度是该信号元素的峰值振幅（X成分和Y成分的组合），连线和X轴之间的角度是信号元素的相位。所有需要的信息都可以从星座图中轻易得到。图5.12表示了星座图。

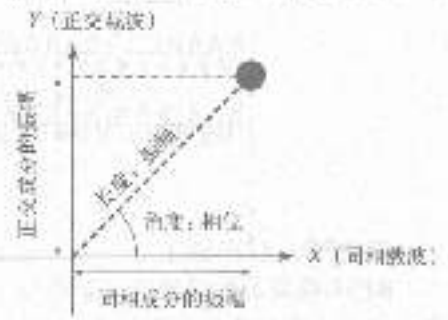


图5.12　星座图概念

例5.8

试分别画出ASK（OOK）、BPSK和QPSK信号的星座图。

解：

图5.13分别是这三个星座图。

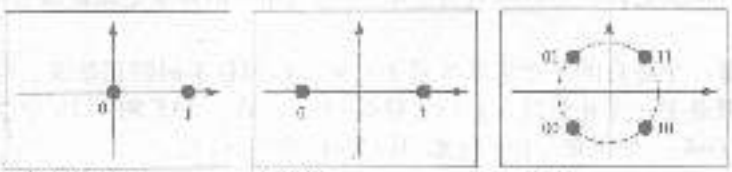


图5.13　三个星座图

让我们分别分析每一种情形：

a. 对于ASK，我们只需要同相载波。因此，两个点应该在X轴上。二进制0有0V的振幅，二进制1有（比如）1V的振幅。这两个点位于原点和单位1处。

b. BPSK也只使用同相载波。但是，我们使用极性NRZ信号用于调制。它产生两种类型的

信号元素：一种振幅是1，另一种振幅是−1。换句话说，BPSK创建两个不同的信号元素，一个振幅为1并同相，另一种振幅为1并有180°相移。

c. QPSK使用两种载波：一种同相而另一种正交。表示11的点由两个组合信号元素组成，两个都是1V的振幅，一个元素由同相载波表示，另一个元素由正交载波表示。发送这个2位数据元素的最后信号元素的振幅是$2^{1/2}$，相位是45°，其他三个点类似。所有信号元素的振幅都有$2^{1/2}$，但是它们的相位不同（45°，135°，−135°和−45°）。当然，可以选择载波振幅$1/(2^{1/2})$代替1V振幅。

5.1.5 正交振幅调制

由于PSK受到设备辨别相位细小差别能力的限制，所以它限制了其潜在的比特率。到目前为止，每次只是改变正弦波三种特性中的一种特性。那么，改变两种特性会怎样呢？带宽的限制使得FSK与其他调制方法无法结合使用。为什么不将ASK与PSK结合在一起呢？即使用两个载波，一个同相而另一个正交，而且每个载波都用不同的振幅。这就是正交振幅调制（quadrature amplitude modulation，QAM）的概念。

正交振幅调制是ASK和PSK的结合。

QAM可能的变化有无穷多个。图5.14给出了其中一些变化方案。图5.14a给出了最简单的4-QAM方案（4个不同信号元素类型），它使用单极NRZ信号来调制每个载波。这与ASK（OOK）的机制一样。图5.14b部分给出了另一个使用极性NRZ的4-QAM方案，但它完全跟QPSK一样。图5.14c部分给出了另一个QAM-4，这个方案使用带两个正电平的信号来调制两个载波。最后，图5.14d给出了带8个电平（4个正4个负）信号的16-QAM星座图。

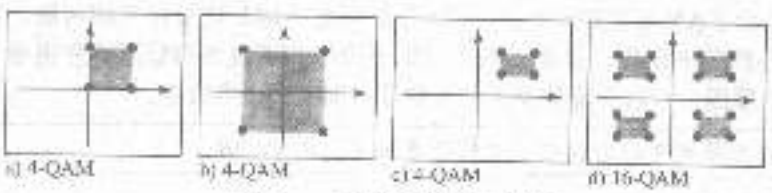


图5.14 一些QAM方案的星座图

QAM的带宽

QAM传输所需的最小带宽与ASK和PSK传输所需的最小带宽相同。相对于ASK来说，QAM与PSK具有同样的优点。

5.2 模拟信号调制

模拟信号调制是通过模拟信号来表示模拟信息的。人们可能会问，既然信号已经是模拟的了，为什么还要调制模拟信号呢？答案是，如果介质具有带通特性或者只有带通带宽可用，那么模拟信号就需要进行调制。这方面的一个实例就是无线电。政府为每一个无线电台分配基带带宽，每个电台生成的模拟信号都是低通信号，都在同一频率范围内。为了能够收听不同电台，需要将低通信号平移，使每一个信号对应不同的频率范围。



图5.15 模拟到模拟调制的类型

模拟到模拟转换（analog-to-analog conversion）可以通过三种方法实现：调幅（amplitude modulation，AM）、调频（frequency modulation，FM）和调相（phase modulation，PM）。FM和PM通常归在一起，见图5.15。

5.2.1 调幅

在AM传输中，对载波信号进行调制，使其振幅随着调制信号的振幅变化而被改变。载波的频率和相位保持不变，只有振幅随着信息改变。图5.16说明了其工作原理。调制信号变成了载波信号的一个包络线。

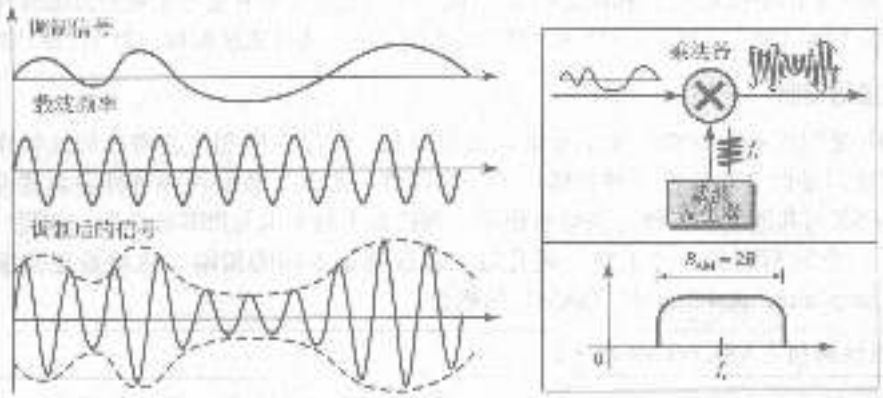


图5.16　调幅

如图5.16所示，AM通常使用简单乘法器实现，因为载波信号的振幅需要根据调制信号的振幅变化。

AM带宽

图5.16还表示了AM信号的带宽。AM信号的带宽是调制信号带宽的两倍，并且覆盖了以载波频率为中心的频率范围。但是，载波频率上边带和下边带的信号成分携带了完全相同的信息。基于这个原因，一些实现丢弃了一半信号，削减了一半带宽。

> AM所需的总带宽可以由音频信号的带宽确定：$B_{AM}=2B$。

AM无线电的标准带宽分配

音频信号（语音和音乐）的带宽通常是5kHz，所以一个调幅的无线电台需要的最小带宽是10kHz，实际上，联邦通信委员会（FCC）分配给每一个AM电台的带宽是10kHz。

AM电台可以使用在530kHz－1 700kHz（1.7MHz）之间的任何一个频率作为载波频率。但是，每一个调幅电台的载波频率与其他电台的载波频率间隔至少是10kHz（一个AM带宽），以避免干扰。如果一个电台使用的载波频率是1 100kHz，下一个电台的载波频率不能低于1 110kHz（见图5.17）。



图5.17　AM频带分配

5.2.2 调频

在FM传输中，载波信号的频率随着调制信号电平（振幅）的改变而调整。载波信号的峰值振幅和相位都保持不变，但是当调制信号的振幅改变时，载波信号的频率相应地改变。图5.18说明了调制信号、载波信号和最终得到的FM信号的关系。

如图5.18所示，FM通常使用用于FSK的压控振荡器实现。振荡器的频率根据输入电平（即调制信号的振幅）变化。

FM带宽

图5.18还表示了FM信号的带宽。实际带宽很难完全确定，但是可以从经验上得出是模拟信号的若干倍，或者$2(1+\beta)B$（这里β是取决于调制技术的因子，一般为4）。

FM所需的总带宽可以由音频信号的带宽确定：$B_{FM}=2(1+\beta)B$。

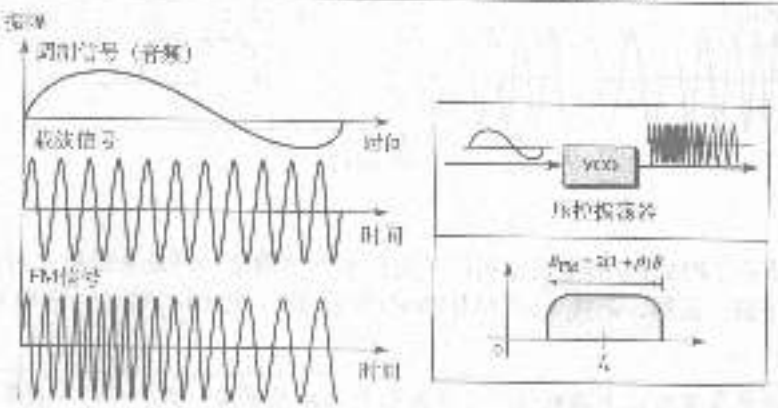


图5.18 调频

FM无线电的标准带宽分配

立体声广播用的音频信号（语音和音乐）的带宽接近于15kHz。FCC允许每一个调频电台使用200kHz（0.2MHz）的带宽，这表示具有$\beta=4$的额外的防护频带。FM电台可以采用88MHz～108MHz之间的任何频率作为载波频率。电台之间必须有至少200kHz的频率差，以保持电台之间的带宽不重叠。为了提供更强的专用性，FCC规定在一个给定的区域内只能交替使用分配的波段，其他波段都被保留，以防止任何两个电台之间的相互干扰。比如给定的范围是88MHz～108MHz，在这个范围共有100个可能的调频波段，但只能同时使用其中的50个。图5.19说明了这一概念。



图5.19 FM波段分配

5.2.3 调相

在PM传输中，载波信号的相位是根据调制信号的电平（振幅）改变而调整。载波信号的峰值振幅和频率都保持不变，但是当调制信号的振幅改变时，载波的相位也相应地变化。可以从数学上证明PM和FM除了有一点不同外都相同（见附录C）。在FM中，载波频率的瞬时变化与调制信号的振幅成正比；在PM中，载波频率的瞬时变化与调制信号振幅的导数成正比。图5.20说明了调制信号、载波信号和最后得到的PM信号之间的关系。

如图5.20所示，PM通常使用压控振荡器和导数发生器实现。振荡器的频率根据输入电平（调制信号的振幅）的导数变化。

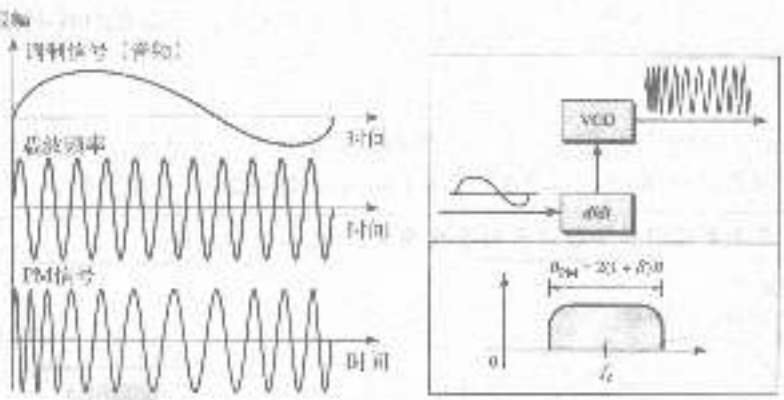


图5.20　调相

PM带宽

图5.20还显示了PM信号的带宽。实际带宽很难准确确定，但是从经验上可以得出是模拟信号带宽的若干倍。虽然，公式显示FM和PM的带宽相同，PM中β较小（窄带大约是1，宽带大约是3）。

PM所需的总带宽可以由调制信号的带宽和最大振幅确定：$B_{PM} = 2(1 + \beta)B$。

推荐读物

为了更详细地讨论与本章有关的主题，推荐下列读物，在本书末列出括号[……]内的参考资料。

读物

数字信号到模拟信号的讨论有[Pea92]的第14章，[Cou01]的第5章和[Sta04]的5.2节。模拟信号到模拟信号的讨论有[Pea92]的第8章到第13章，[Cou011]的第5章和[Sta04]的5.4节，[Hsu03]给出本章所有内容的数学方法，更深入的资料可参考[Ber96]。

本章小结

- 数字到模拟的转换是指根据数字数据中信息而改变模拟信号某种特性的过程。
- 数字到模拟转换可以使用下列方法完成：幅移键控（ASK），频移键控（FSK）和相移键控（PSK），以及ASK与PSK结合的正交振幅调制（QAM）。
- 在幅移键控调制中，载波信号的振幅改变生成信号元素。振幅改变，频率与相位保持不变。
- 在频移键控调制中，载波信号的频率随所表示数据而改变。调制信号的频率在一个信号元素持续期间内保持不变。但如果数据元素改变则下一个信号元素也改变。而所有信号元素峰值振幅和相位保持不变。
- 在相移键控调制中，载波信号的相位改变表示两个或多个不同信号元素。相位变化，峰值振幅和频率保持不变。
- 特别在用两个载波（一个是同相的，另一个是正交的）时，星座图表示信号元素的振幅和相位。
- 正交振幅调制（QAM）是ASK与PSK结合，QAM使用两个载波（一个是同相的，另一

个是正交），每个载波信号有不同的振幅值。

- 模拟到模拟转换是通过模拟信号来表示模拟信息。如果介质具有带通特性或者只有带通带宽可用，那么模拟信号就需要调制。
- 模拟到模拟转换可使用三种方法：调幅（AM），调频（FM）和调相（PM）。
- 在AM传输中，载波信号被调制使得其振幅随调制信号的振幅改变而改变。载波的频率和相位保持不变，只有振幅随信息不同而改变。
- 在FM传输中，载波信号的频率随调制信号（振幅）的电平改变化而调整。载波信号的峰值振幅和相位保持不变。但是当信号的振幅改变时，载波的频率相应地改变。
- 在PM传输中，载波信号的相位随着调制信号（振幅）电平改变调整。载波信号峰值振幅和频率保持不变，但信息信号的振幅改变化，载波的相位相应地改变。

习题

复习题

1. 解释模拟传输。
2. 解释载波信号及其在模拟传输中的作用。
3. 解释数字到模拟的转换。
4. 在下列每种数字到模拟的转换中，模拟信号对哪些特性的修改来表示数字信号？
   a. ASK　　b. FSK
   c. PSK　　d. QAM
5. 数字到模拟转换的四种技术（ASK，FSK，PSK及QAM），哪一种最易受噪声影响？
6. 解释星座图，并说明它在模拟传输中所起的作用？
7. 信号波表示为星座图，信号的两个分量是什么？水平轴代表什么分量？垂直轴代表什么分量？
8. 解释模拟到模拟的转换。
9. 在下列每种模拟到模拟转换中，模拟信号什么特性的改变可表示低通模拟信号？
   a. AM　　b. FM
   c. PM
10. 模拟到模拟转换的三种技术（AM，FM及PM），哪一种最易受噪声影响？

练习题

11. 已给定比特率和调制类型，试计算波特率：
    a. 2 000bps，FSK　　b. 4 000bps，ASK
    c. 6 000bps，QPSK　　d. 36 000bps，64-QAM
12. 已给定波特率和调制类型，试计算比特率：
    a. 1 000波特，FSK　　b. 1 000波特，ASK
    c. 1 000波特，BPSK　　d. 1 000波特，16-QAM
13. 下列技术每波特有多少位？
    a. 具有4个不同振幅的ASK　　b. 具有8个不同频率的FSK
    c. 具有4个不同相位的PSK　　d. 128点星座图的QAM
14. 试画出下列各情况的星座图：
    a. 峰值振幅为1和3的ASK调制
    b. 峰值振幅为2的BPSK调制
    c. 峰值振幅为3的QPSK调制

d. 具有两个分别为1和3的峰值振幅以及4个相位的8-QAM

15. 对下列情况试画出星座图，并对每种情况求出峰值振幅值及确定调制类型（ASK，FSK，PSK或QAM）。注意：括号内的坐标分别表示I和Q分量。

a. 两个数据点（2，0）和（3，0）的星座图

b. 两个数据点（3，0）和（-3，0）的星座图

c. 四个数据点（2，2），（-2，2），（-2，-2）和（2，-2）的星座图

d. 两个数据点（0，2）和（0，-2）的星座图

16. 如果星座图具有下列数据点的个数，试问每个波特发送多少位？

a. 点数为2　　b. 点数为4

c. 点数为16　　d. 点数为1 024

17. 如果发送速率为4 000bps，试问下列情况要求带宽是多少？设d=1。

a. ASK　　b. 具有$2\Delta f$=4kHz的FSK

c. QPSK　　d. 16-QAM

18. 电话线的带宽是4kHz，用下列调制技术能发送的最大比特率是多少？设d=0。

a. ASK　　b. QPSK

c. 16-QAM　　d. 64-QAM

19. 某公司拥有1MHz带宽（低通）的介质，需要建立10个独立的通道，每个通道至少能发送10Mbps的能力，决定用QAM调制技术。试问对每个通道，每波特最小位的个数是多少？每个通道星座图点数是多少？设d=0。

20. 有线电视公司用有线电视通道（带宽为6MHz）为每个用户提供数字通信服务。如果公司使用64-QAM技术，每个用户的有效速率是多少？

21. 如果需要调制带宽为5 kHz的语音信号，对下列情形试求带宽。

a. AM　　b. FM（置$\beta=5$）

c. PM（置$\beta=1$）

22. 用FCC规定的带宽试求下列通道的个数：

a. AM　　b. FM

第6章 带宽利用

实际生活中，链路都是有限带宽，合理运用带宽将是电路通信亟待解决的主要问题。然而，所谓合理是与应用有关。有时我们需要将一些低带宽通道组合在一起形成一条高带宽通道，有时为达到目的又需要扩展通道带宽，如保密、抗干扰等。本章将系统地讨论带宽合理使用的两个主要类型：复用与扩频。在复用中，目标是效率，将一些通道合并为一个通道；在扩频中，目标是保密、抗干扰。要插入冗余扩展通道的带宽，这是为达到保密与抗干扰要求所必需的。

为达到特殊的目的，带宽利用是可用带宽的合理使用。复用可获得效率；扩频可达到保密与干扰。

6.1 复用

只要连接两台设备的介质带宽比设备间传输所要求的带宽高时，该链路就可以被共享。**复用**（multiplexing）就是允许同时通过一条数据链路传输多个信号的一组技术。随着数据和电信应用的增加，通信量也不断增加。当然可以通过每需要一条新的通道就建立一条单独链路的方式来满足这种增长，也可以安装更高带宽的链路，并在这些链路上采用复用技术。如第7章所述，目前的技术包括诸如光纤、地面微波和卫星微波等高带宽介质。每一种介质都具有远超过平均传输需求的承载能力。如果一条链路的带宽比连接在它的上面的设备所需的带宽要大，那么多余的带宽就被浪费。一个高效的系统是可以最大限度地使用所有设备的资源，而带宽是在数据通信中最为昂贵的资源之一。

在复用的系统中，n条线路共享一条链路的带宽。图6.1说明了复用系统的基本形式。左侧的4条线路将它们的传输流量送到**复用器**（multiplexer，MUX）。复用器将这些流量组成一个单独的传输流（多对一）。在接收端，这个传输流量被**分离器**（demultiplexer，DEMUX）接收，并分解成原来几个独立的传输流（一对多），并直接发送到对应的线路上。图6.1中的**链路**（link）一词是指的是物理通路。**通道**（channel）一词是指在给定一对设备之间传送传输信号的链路的那部分。一条链路可能有多个（n）通道。



图6.1 链路划分为通道

信号可以通过三种基本技术进行多路复用：**频分复用**（frequency-division multiplexing，FDM），**波分复用**（wave-division multiplexing，WDM）和**时分复用**（time-division multiplexing，TDM）。前两种技术用于模拟信号，第二种技术用于数字信号（见图6.2）。

虽然有些教科书将码分复用接入（CDMA）看做是第四种技术，但我们还是把它作为接入方法讨论（见第12章）。

6.1.1　频分复用

频分复用（FDM）是一种模拟技术，在链路带宽（以Hz为单位）大于要传输的信号的带宽之和时采用。在FDM中，每个发送设备生成的信号用于调制不同的载波频率。调制后的信号再被合并为一个可以通过链路传输的复合信号。载波频率之间的频率差必须能够容纳调制信号的带宽。这些带宽的范围就是不同信号通过的通道。通道之间由狭长的未使用的带宽，即防护频带（guard band）将进行分隔，以防止信号重叠。另外，载波频率必须不会影响原始的数据频率。不符合以上任一条件就会导致原始信号的不可恢复。



图6.2　多路复用的类型

图6.3给出了FDM的概念描述。图中传输通路分为三部分，每一部分都是传输某一个流量的通道。

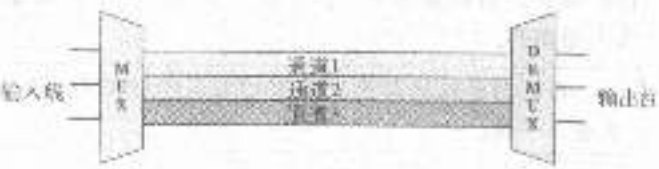


图6.3　频分复用

我们认为FDM是模拟复用技术，但这不是说FDM不能把源端的发送数字信号组合在一起。在FDM使用复用之前，数字信号可以转换成模拟信号（有关技术在第5章中已讨论）。

> FDM是用来组合模拟信号的模拟多路复用技术。

复用过程

图6.4是多路复用过程的示意图。FDM是一个模拟过程。每一个源端产生频率范围类似的信号。在复用器中，这些类似信号被调制到不同的载波频率上（f_1、f_2和f_3）。然后，将调制后的信号合成为一个复合信号，并通过具有足够带宽的介质链路发送出去。

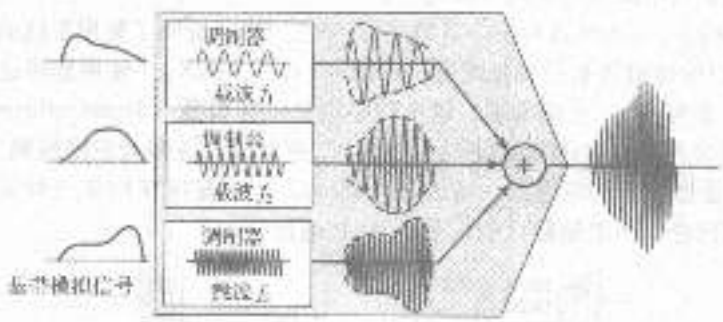


图6.4　FDM过程

复用分离过程

分离器使用一系列滤波器来将复用信号分解为组成它的各个信号。每个信号随后被传送到解调器，解调器将它们与载波分离并转发给等待的接收端。图6.5是FDM复用分离过程的示意图。

例6.1

假定一个语音通道占用的带宽是4kHz。要将三个语音通道合并到一条带宽为12kHz（20～32kHz）的链路，使用频域图表示这一复用过程。这里假定不使用防护频带。

解：

将三个语音通道平移（调制）到不同的带宽，如图6.6所示。第一个通道使用20～24kHz

的带宽。第二个通道使用24～28kHz的带宽，第三个通道使用28～32kHz的带宽。然后，如图6.6所示将它们合并。在接收端，每个通道接收到完整信号后，使用滤波器将自己的信号分离出来。第一个通道使用能够通过20～24kHz频率成分的滤波器，并丢弃任何其他频率成分。第二个通道使用能够通过24－28kHz频率成分的滤波器，而第三个通道使用能够通过28kHz－32kHz频率成分的滤波器。最后，每一个通道将频率平移到从0开始的起点。

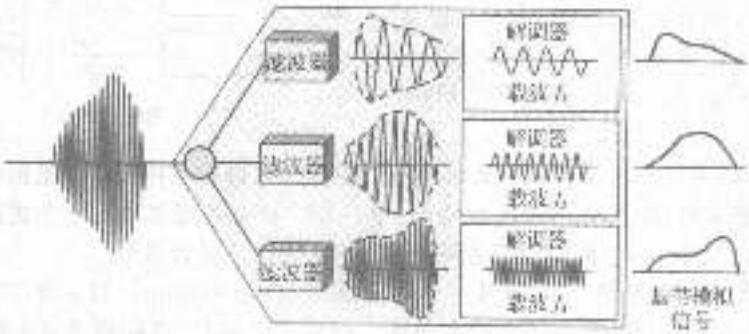


图6.5 FDM分复示例

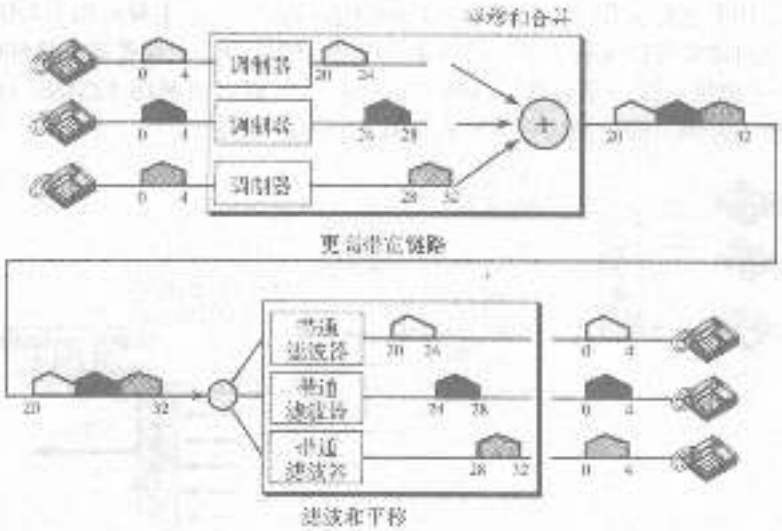


图6.6 例6.1

例6.2

有5个通道，每个通道的带宽是100 kHz，全部进行多路复用。如果通道之间需要10kHz的防护频带以防止干扰，则链路的最小带宽是多少？

解：对于5个通道，至少需要4个防护频带。这意味着至少需要带宽5 × 100 + 4 × 10 = 540kHz，如图6.7所示。

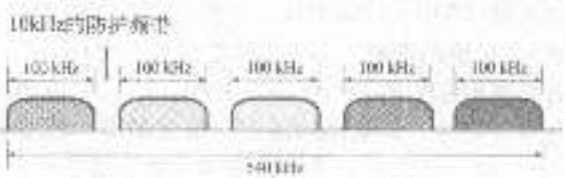


图6.7 例6.2

例6.3

有4个数据通道（数字的），每一个通道的传输速率是1Mbps，使用1MHz的卫星通道。使用FDM设计一种合理的配置。

解：

卫星通道是模拟的，将其划分为4个通道，每个通道的带宽是250kHz。对每个1Mbps的数

于通道进行调制，每4位调制为1Hz。一种解决方案是16-QAM调制。图6.8描述了一种可能的配置方案。

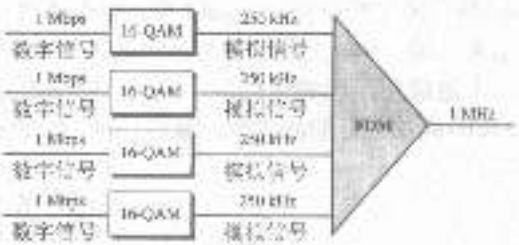


图6.8　例6.3

模拟载波系统

电话公司为了最大限度地是高基础设施的效率，传统方式是把来自低带宽线路的信号复用到更高带宽的线路上。通过这种方式，许多交换和专用线路可以合并为更少但是容量更大的通道。对于模拟线路来说，使用的是FDM技术。

其中一种AT&T采用层次结构的系统是由群、超群、主群和巨群组成（见图6.9）。

在这个**模拟层次结构**（analog hierarchy）中，12个语音通道复用到一个更高带宽的线路上构成一个群（group）。一个群的带宽为48kHz，并支持12个语音通道。

在下一层，最多可以复用5个群产生一个称为**超群**（supergroup）的复合信号。一个超群的带宽是240kHz，而且支持多达60个语音通道。超群可以由5个群构成或者由60个独立的语音通道构成。

在下一层，10个超群复用为一个**主群**（master group）。一个主群必须有2.40MHz的带宽。但是由于超群之间需要防护频带，使得总带宽上升到2.52MHz。主群最多支持600个语音通道。

最后，6个主群组合成一个**巨群**（jumbo group）。巨群必须有15.12MHz（6×2.52MHz）的带宽，但是为了容纳主群之间防护频带而调整为16.984MHz。

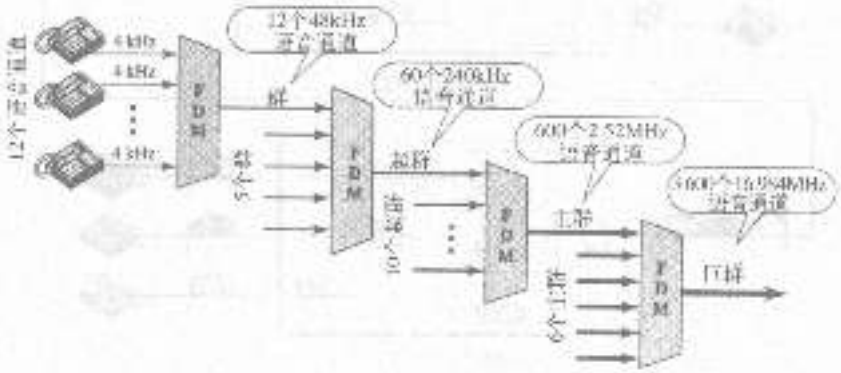


图6.9　模拟层次结构

FDM的其他应用

FDM的一个最常见的应用是AM或者FM广播。无线电使用空气作为传输介质。将一个特定波段（530～1700kHz）分配给了AM无线电广播，所有的无线电台需要共享这一波段。如第5章讨论的那样，每个AM电台都需要10kHz带宽，每个电台使用不同的载波频率，这意味着它要平移它的信号并进行复用。空气中传播的信号是所有信号的组合。接收机会接收到所有这些信号，但是能够滤波（通过调谐）想要接收的信号。如果没有复用，那么就只会有一个AM电台可以向公共链路即空气中广播信号。然而，我们需要知道这里有物理的复用器或分离器，在第12章将会看到复用是在数据链路层完成的。

这种情况与FM广播类似。但是FM使用的带宽更高，为88－108MHz，这是因为每个电台需要更高的带宽，即200kHz。

FDM的另一个常见应用是电视广播，每一个电视频道的带宽是6MHz。

第一代移动电话（仍然在运行）也使用FDM，为每一用户分配两个30kHz的通道，一个用于发送给台而另一个用于接收。语音信号的带宽是3kHz（300～3 300Hz），使用调频进行调制，要记住：FM信号的带宽是调制信号的10倍，因此每个通道有30kHz（10×3）的带宽。这样，每一个用户在呼叫时，由基站提供某个可用范围内的带宽是60kHz。

例6.4

高级移动电话系统（AMPS）使用两个波段。第一个波段是（824～849MHz），用于发送，而869～894MHz用于接收。每一个用户在每个方向上都有30kHz的带宽。3kHz语音使用FM调制，生成30kHz的调制信号。试问可以有多少人同时使用移动电话？

解：

每一个波段是25MHz。如果将25MHz按30kHz划分，可以得到833.33个。实际上，波段划分为832个通道。在这些通道中，42个通道用于控制，意味着只有790个通道可用于移动电话用户。第16章会更详细地讨论AMPS。

实现

FDM的实现非常容易。和无线电或电视广播一样，在多数情况下不需要物理的复用器和分离器。只要所有电台使用不同的载波频率向空气中广播，就可以实现多路复用。其他情况下，例如移动电话系统，基站需要为用户分配载波频率。在一个信元内，由于没有足够的可用带宽，所以不能为每一个电话用户分配永久的带宽。当某个用户挂断时，带宽就会分配给其他的呼叫者。

6.1.2 波分复用

波分复用（Wavelength division multiplexing，WDM）用于具有高数据速率传输能力的光缆。光缆的数据速率比金属传输介质的数据速率高，将光缆用作单一线路浪费了可用带宽。复用允许将多条线路连接为一条线路。

除了复用和多路分离包括通过光纤通道传输的光信号以外，波分复用在概念上与FDM相同。其原理是一样的，都是将不同频率的不同信号合并，但其差别是这些频率非常高。

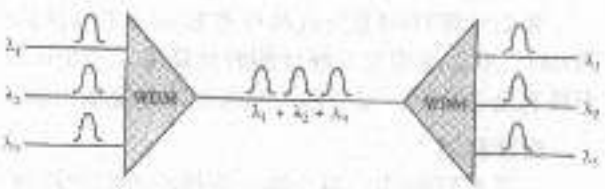


图6.10 波分复用

图6.10给出了波分复用的复用器和信号分离器的示意图。来自不同源端的窄波段的光合并生成一种波段更宽的光。在接收器端，信号通过信号分离器进行分离。

> WDM是合并多个光信号的模拟多路复用技术。

尽管WDM这种技术非常复杂，但其想法是非常简单的。在复用器上将多个光源组成单一光信号，而在分离器上做相反的处理。光源组合与分离由棱镜完成。普通物理知识告诉我们，棱镜可以根据入射角和频率将几束光合成一道光。利用这项技术，复用器可组合几束输入光（每束光包含窄带频率）为一束输出光（宽带频率）。分离器做相反的过程。其概念如图6.11所示。



图6.11 棱镜作为波分复用和分离

WDM的一种应用是同步光纤网络（SONET），其中多条光纤线路进行复用和信号分离。第17章会讨论SONET。

有一种新方法，称为密集波分复用（dense WDM，DWDM），这种方法通过使相邻通道之间的频率更接近的方法来复用大量的通道，从而获得非常高的效率。

6.1.3　同步时分复用

时分复用（time-division multiplexing，TDM）是一个数字化过程，它允许多个连接共享一条高带宽链路，与FDM共享一部分带宽不同的是，TDM是在时间上共享，每个连接占用链路的一个时间段。图6.12给出了TDM的示意图。注意：同一条链路都用于FDM，但是这里表示的链路分割是时间上而不是频率上的。在时分复用图中，信号1、2、3和4依次占用链路。

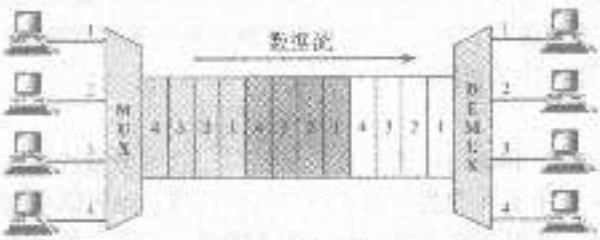


图6.12　TDM

注意：图6.12只涉及复用而不是交换，这就是说来自源1的报文中的所有数据总有一个特定的目的地，它可能是1、2、3或4，传送到固定目的地是不变的，这与交换是不同的。

从理论上讲，TDM是数字复用技术。它将不同源端的数字数据合并到一个时间共享的链路上。然而，这并不是说源端不能生成模拟数据，模拟数据可以被采样改变成数字数据，然后用TDM进行复用。

> TDM是组合多个低速的通道为一个高速通道数据的复用技术。

我们可将TDM划分成两种方案：同步的与统计的。首先讨论同步的时分复用（synchronous TDM），然后说明它与统计的时分复用（statistical TDM）有那些不同。在同步的TDM中，甚至没有发送数据，每个输入端也允许连接输出端。

时隙和帧

在同步TDM中，每个输入连接的数据流被划分为多个单元，其中每个输入占用一个输入时隙，一个单元可以是一位，一个字符或一个数据块。每个输入单元成为一个输出单元，占用一个输出时隙。然而，每个输出单元的持续时间是输入单元持续时间的n分之一。如果输入时隙是T秒，则输出时隙是T/n秒，其中n是连接数。换言之，输出连接单元具有较短的持续时间，输出速率更快。图6.13显示了同步TDM一个实例，其中n为3。

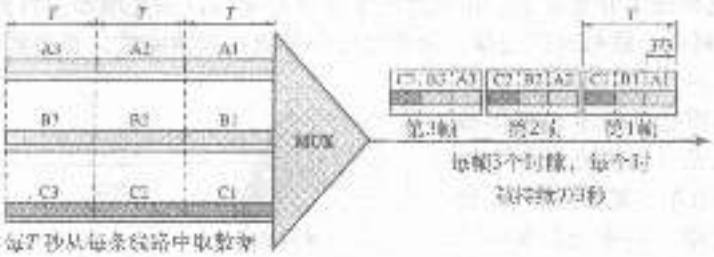


图6.13　同步时分复用

在同步TDM中，每个输入连接的全部数据单元组成一个帧（不久我们将会看到其理由）。如果有n条连接线，则一帧划分成n个时隙，一个时隙分配给每个单元。如果输入单元持续T秒，则每个时隙持续时间是T/n，而每个帧持续时间是T（除非帧携带其他信息，这不久我们将会看到）。

输出的链路的数据速率必须是单个连接的数据速率的n倍，以确保数据流动。在图6.13中，链路数据速率是一条连接的数据速率的3倍。同样，一条连接中的一个单元的持续时间是时隙（链路中一个单元的持续时间）的3倍。图中，将复用之前的数据表示为复用后数据大小的3倍。这只是表达一种思想，即每个单元在复用之前的持续时间是复用后的3倍。

> 在同步TDM中，链路速率是数据速率的n倍，并且比单元持续时间短n倍。

时隙分组为多个帧，帧由多个时隙组成的一个完整的循环构成。每个时隙专用于每一个发送设备。在具有n条输入线路的系统中，每个帧有n个时隙，分配每个时隙用于传送来自特定输入线路的数据。

例6.5

在图6.13中，每个输入连接的数据速率是33kbps，一个单元为1位。试确定：(a) 每个输入单元的时隙；(b) 每个输出单元的时隙；(c) 每个帧的时隙。

解：

这些问题回答如下：

a. 每个输入连接的数据速率是1kbps，这就是说位持续时间是1/1 000秒，即0.001s（1ms），输入时隙的持续时间是1ms（与位持续时间相同）；

b. 每个输出时隙的持续时间是输入时隙的三分之一，这说明输出时隙的持续时间是1/3ms。

c. 每个帧具有三个输出时隙，因此一个帧持续时间是3×1/3ms，即1ms。

例6.6

图6.14表示了4个输入数据流和1个输出数据流的同步TDM，数据单元是一位。试求(a) 输入位的持续时间；(b) 输出位的持续时间；(c) 输出比特率；(d) 输出帧的速率。

解：

这些问题回答如下：

a. 输入位的持续时间是速率的倒数：1/1Mbps=1μs；

b. 输出位持续时间是输入位持续时间的四分之一，即1/4μs；

c. 输出比特率是输出位持续时间1/4μs的倒数，即4Mbps。这也可从输出速率比输入速率快4倍这一事实推出，即输出速率=4×1Mbps=4Mbps；

d. 帧速率常与任一输入速率相同，因此帧速率是每秒1 000 000帧。因为我们发送每帧4位，所以我们可用复用证明前一结论，用每帧位的个数证明帧速率。

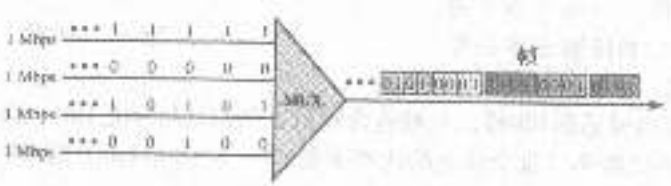


图6.14 例6.6

例6.7

将4个1kpbs的连接一起复用，每个单位为1位，试求：(a) 复用前一位持续的时间；(b) 链路传输速率；(c) 时隙持续时间；(d) 一帧持续时间。

解：

a. 复用前一位持续时间是1/1kbps，即0.001秒（1ms）；

b. 链路速率是连接速率的4倍，即4kbps；

c. 每个时隙持续时间是复用前每位持续时间的四分之一，即250μs。注意：我们也可以从链路数据速率4kbps来计算，位持续时间是数据速率1/4kbps的倒数，即250μs；

d. 帧持续时间总是与复用前一个单元持续时间相同，即1ms。我们也可以从另一个方法计算，此时每帧有4个时隙，所以一个帧持续时间是250μs的4倍，即1ms。

交替

TDM可以看作是两个快速旋转的开关，一个位于复用一侧，另一个位于信号分离一侧。两个开关是同步的，以相同的速度旋转，但是方向相反。在复用一侧，当开关在一个连接前打开时，连接即有机会向通路上发送一个单元。这个过程称为**交替**（interleaving）。在信号分离一侧，当开关在某个连接前打开时，连接即有机会从通路中接收一个单元。

图6.15回出了图6.13所示连接的交替过程。图中暂时不考虑切换，假定来自复用器一端的第一个连接的数据传输到信号分离器一端的第一个连接中。我们将第8章讨论交换。

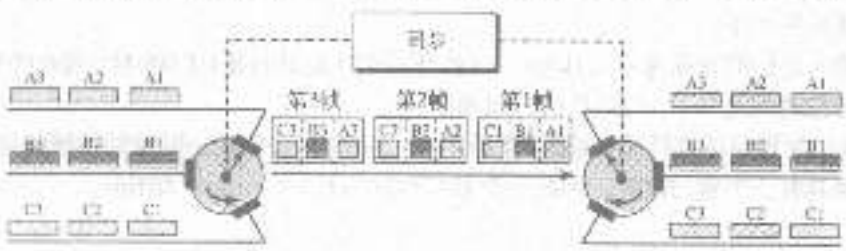


图6.15　交替

例6.8

4个通道使用TDM实现复用。如果每个通道的发送速度是100字节/秒，每个通道复用1字节，试画出链路中帧的传输情况，说明帧的大小、帧的持续时间、帧传输速率以及链路的比特率。

解：

复用器如图6.16所示。每帧从每个通道中传送1字节，所以每个帧的大小是4字节，即32位。因为每个通道发送速率是100字节/秒，一帧从每个通道中运送一个字节，帧传输速率必须是每秒100帧。所以，每帧的持续时间是1/100秒，链路每秒钟运送100帧，每帧包含32位，所以比特率是100×32，即3 200bps。这实际上是每个通道的比特率（每个通道的比特率是100×8=800 bps）的4倍。

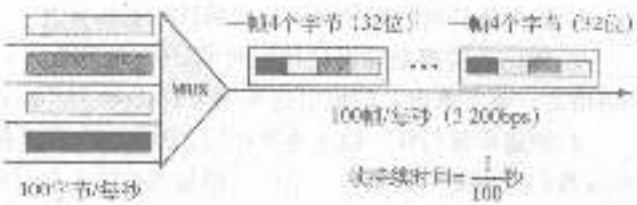


图6.16　例6.8

例6.9

复用器使用2位的时隙合并4个100kbps的通道，试画出任意4种输入的输出结果。帧速率

是多少？帧的持续时间是多少？比特率是多少？位持续时间是多少？

解：

图6.17画出了任意4种输入的输出结果。由于每帧包含每个通道的2位，所以链路每秒钟传输50 000帧。每帧的持续时间是1/50 000秒，即20ms。帧速率是50 000帧/秒，每帧运送8位。比特率是50 000 × 8 = 400 000位，即400kbps。位持续时间是1/400 000s，即2.5μs。注意：帧的持续时间是位持续时间的8倍，因为每帧运送8位。



图6.17 例6.9

空时隙

同步TDM的效率并不高。当某一源端没有数据发送时，输出帧中对应的时隙是空的。图6.18表示了其中一条输入线没有数据发送而另一输入线中一个时隙有不连续数据的情况。

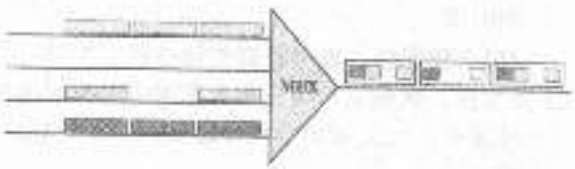


图6.18 空时隙

第一个输出帧就有三个时隙，第二个帧填有两个时隙而第三个帧填有三个时隙。没有一个帧是填满的。下一节讨论通过除去帧中空时隙来提高效率的统计TDM。

数据速率管理

TDM中的一个问题是如何处理输入数据速率的差异，直到目前为止，都假定所有输入数据的速率都是相同的。当数据速率不同时，可用三种策略或这三种策略的组合。它们是多级复用（multilevel multiplexing）、多时隙分配（multiple-slot allocation）和脉冲填充（pulse stuffing）。

多级复用 当一条输入数据线的数据速率是其他一些输入数据线数据速率的整倍数时，可采用多级复用技术。例如，在图6.19中，有两条20kbps的输入线和三条40kbps的输入线。前面两条输入线可一起复用提供的数据速率等于后面的三条，两级复用产生160kbps的输出。

多时隙分配 有时在一个帧中允许对一条输入线分配多个时隙更为有效率。例如，可能有一条输入线的数据速率是其他输入线数据速率的倍数。图6.20中，具有50kbps数据速率的输入线在输出中可分配两个时隙。这可通过插入串并行转换使原输入线分成两个输入线实现。

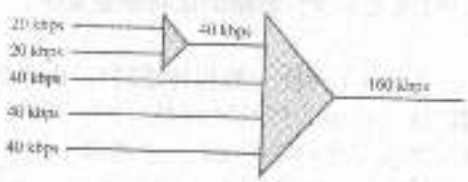


图6.19 多级复用

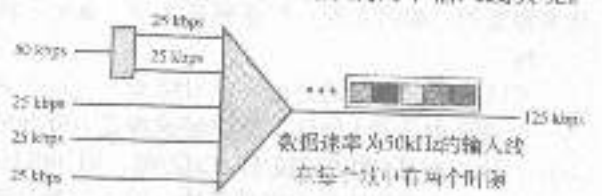


图6.20 多时隙复用

脉冲填充 有时源输入线的比特率不是其他每个输入线比特率的整数倍，上面两种技术都不能使用，此时解决办法是选取输入线最高速率作为主要速率，然后将所有低速率的输入线添加虚位。这种技术称为脉冲填充，或位填充。其想法如图6.21所示，具有46kbps的输入

线填充脉冲使其速率增加到50kbps，此时就可以复用了。

帧同步

TDM的实现不如FDM简单，复用器和分离器之间的同步是一个主要问题。如果复用器和分离器失去同步，则属于某一个通道的位可能会被错误的通道所接收。由于这个原因，通常在每帧的开始增加一个或者多个同步位。这些位称为帧指示位（framing bit）。帧指示位按照某种模式逐帧发生变化，使分离器与输入流同步，以准确地分离时隙。多数情况下，每帧的同步信息由1位构成，交替变化为0和1，如图6.22所示。

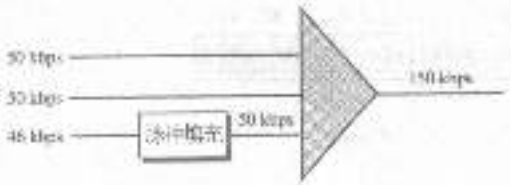


图6.21　脉冲填充

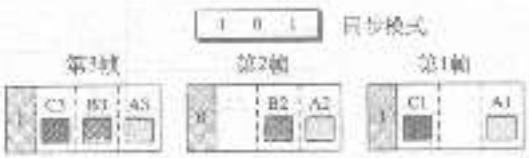


图6.22　帧指示位

例6.10

有4个数据源，每个数据源每秒种产生250个字符。如果交替的单元是1个字符，每帧增加1个同步位，试确定：(a) 每个数据源的数据速率；(b) 每个数据源中每个字符的持续时间；(c) 帧速率；(d) 每帧的持续时间；(e) 每帧的位数；(f) 链路的数据速率。

解：

问答问题如下：

a. 每个数据源的数据速率是250 × 8 = 2 000bps = 2kbps。

b. 每个数据源每秒发送250个字符，所以每个字符的持续时间是1/250s，即4ms。

c. 每帧含有每个数据源的1个字符，意味着链路需要每秒发送250帧，以保持每个数据源的传输率。

d. 每帧的持续时间是1/250s，即4ms。注意，每帧的持续时间与来自各个数据源的每个字符的持续时间相同。

e. 每帧传送4个字符和1个附加的同步位，这表示每帧是4 × 8 + 1 = 33位。

f. 链路每秒钟发送250帧，每帧包含33位，这表示链路的数据速率是250 × 33，即8 250bps。该链路的数据速率高于4个通道的比特率之和。若将4个通道的比特率相加，结果是8 000bps。由于每秒传输250帧，同时每一帧包含附加的1位用于同步，所以要在结果中再加上250，即得到8 250bps。

例6.11

两个通道，一个通道的比特率是100kbps，而另一个通道的比特率是200kbps，对它们实现多路复用。如何实现？帧速率是多少？每帧的持续时间是多少？链路的比特率是多少？

解：

可以为第一个通道分配一个时隙而为第二个通道分配两个时隙，每帧传送3位。由于它从第一个通道中传送1个位，所以帧速率是100 000帧/秒，每帧的持续时间是1/100 000s，即10ms。比特率是100 000帧/秒 × 3位/帧，即300kbps。注意，由于每帧从第一个通道中传送1个位，第一个通道的比特率得到保持。因为每帧从第二个通道中传送2个位，所以第二个通道的比特率也得到保持。

数字信号服务

电话公司通过一种数字信号的层次结构实现TDM，称为**数字信号**（digital signal，DS）**服务或数字层次结构**（digital hierarchy）。图6.23说明了每一级支持的数据速率。

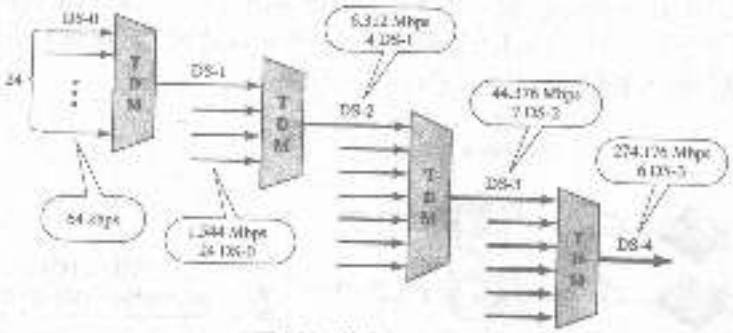


图6.23 数字层次结构

- DS-0服务是64kbps的单一数字通道。
- DS-1是一种1.544Mbps的服务。1.544Mbps是24×64kbps加上8kbps的开销得到的。可以用做单一的1.544Mbps传输服务，或者作为24个64kbps通道的多路复用，或者用户需要时，还可以用于适合于1.544Mbps容量的其他服务类型的组合形式。
- DS-2是一种6.312Mbps服务。6.312Mbps是96×64kbps加上168kbps的开销得到的。可以用做单一的6.312Mbps传输服务，或者用做4个DS-1通道的多路复用，或者用做96个DS-0通道，或者用做这些服务类型的组合形式。
- DS-3是一种44.376Mbps服务。44.376Mbps是672×64kbps加上1.368Mbps开销得到的。可以用做单一的44.376Mbps传输服务，或者用做7个DS-2通道的多路复用，或者用做28个DS-1通道，或者用做672个DS-0通道，或者用做这些服务类型的组合形式。
- DS-4是一种274.176Mbps服务。274.176是4 032×64kbps加上16.128Mbps开销得到的。可以用做6个DS-3通道的多路复用，或者用做42个DS-2通道，或者用做168个DS-1通道，或者用做4 032个DS-0通道，或者用做这些服务类型的组合形式。

T线路

DS-0、DS-1等是服务的名称。为了实现这些服务，电话公司使用T线路（T lines，T-1到T-4）。这些线路的容量与DS-1到DS-4的数据速率是精确匹配的（见表6.1）。

表6.1 DS和T线路速率

服 务	线 路	速率（Mbps）	语音通道
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

T-1线路用于实现DS-1，T-2线路用于实现DS-2等。从表6-1中可以看出，DS-0实际上没有作为服务提供，但是它被定义为用于参考的基础。

用于模拟传输的T线路

T线路是为数字数据、音频或者视频设计的数字线路，但它也可以用于实拟传输（常规的电话连接）。前提是要先对模拟信号采样，然后使用TDM。

能够使用T线路作为模拟信号载体为电话公司开启了新的服务时代。早期，如果某个组织机构需要24条单独的电话线路，需要从公司到中心交换机安装24条双绞线电缆（你是否还记得一些老电影片中，繁忙的行政长官的桌子上有10部电话，或者还能有粗电缆的老式办公室

电话？那些电缆中有很多根电话线）。现在，那些组织机构可以把24条线路合并为一条T-1线路，并且只租这一条T-1线路即可进行数据交换。图6.24说明了如何将24个语音通道多路复用为一条T-1线路（参考第5章的PCM编码）。

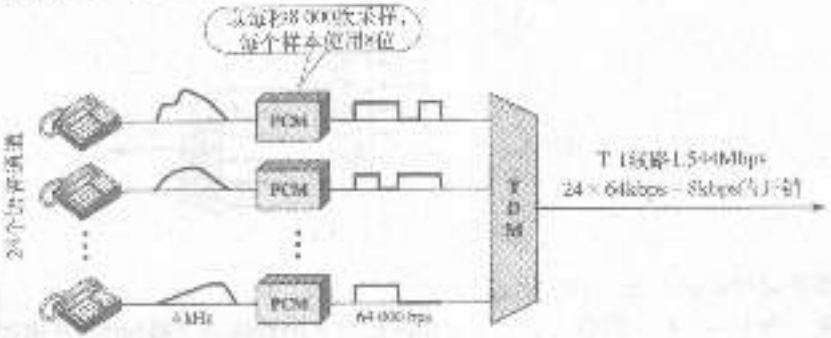


图6.24　用于电话线路多路复用的T-1线路

T-1帧　上面提到过，DS-1需要8kbps的开销。为了理解这种开销是如何计算出来的，必须考查24条语音通道的帧的格式。

用于T-1线路的帧通常是193位，分为24个8位的时隙，每帧附加1位用于同步（24×8+1=193），见图6.25。换句话说，每个时隙包含来自每个通道的信号片段。24个片段在帧中交替排列。如果一条T-1线路传送8 000帧，那么数据速率（即线路的容量）是1.544Mbps（193×8 000=1.544Mbps）。

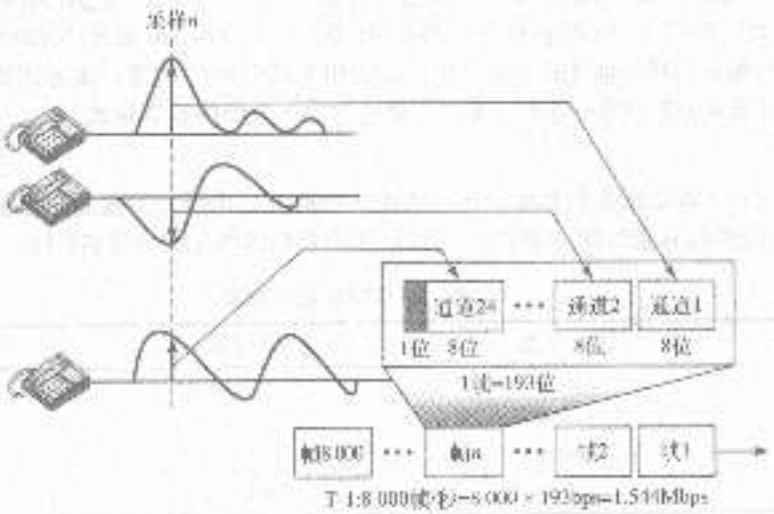


图6.25　T-1帧结构

E线路

欧洲使用的T线路版本称为E线路（E line），这两种系统在概念上是相同的，但是容量不同。表6.2列出了E线路及其容量。

表6.2　E线路的速率

线路	速率（Mbps）	语音通道
E-1	2.048	30
E-2	8.448	120
E-3	34.368	480
E-4	139.264	1 920

同步TDM的其他应用

一些第二代移动电话公司使用同步TDM。例如，前面讨论过的移动电话数字版本，仍然将可用带宽划

分为多个30kHz的波段。对于每一个波段，使用TDM使6个用户可以共享这一波段。这表示每个30kHz的波段现在由6个时隙构成。用户的数字化语音信号就插入到这些时隙中。使用TDM，每个区域的电话用户的数量现在增加到6倍以上。我们将在第16章讨论第二代移动电话。

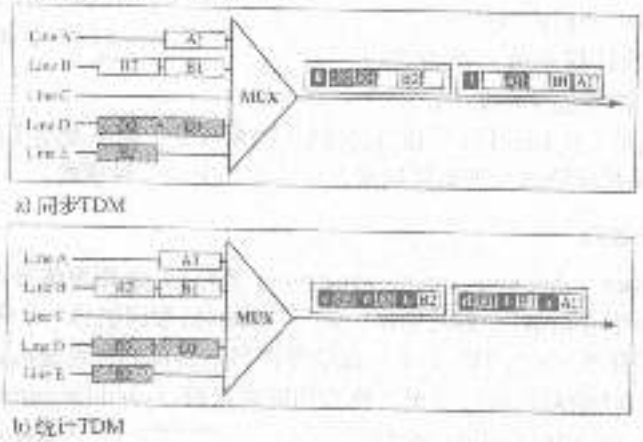
6.1.4 统计时分复用

在上一节看到，在同步TDM中每个输入在输出帧中都占有一个时隙，如果有些输入线没有数据发送，那么效率就不高。在统计时分复用中，动态地分配时隙以提高带宽的效率。仅当输入线有发送数据，时隙才有意义并在输出帧中给予一个时隙。在统计复用中，每个帧中时隙的个数小于输入线的条数。复用器循环顺序地检测每条输入线，如果输入线有数据发送，则对输入线分配一个时隙。否则跳过这条线检测下一条线。

图2.26表示了一个同步和一个统计TDM的例子。前一个例子由于对应的线没有数据发送，某些时隙是空的。然而在后一个例子中，只要有一条输入线有数据要发送，那么就没有空的时隙。

寻址

图6.26也表明了同步TDM和统计TDM时隙之间的主要差别。在同步TDM中，输出时隙全部由数据占用；而在统计TDM中，输出时隙需要携带数据和目的地址。在同步TDM中不需要寻址。作为输入和输出的地址之间关系是同步的和指定的。例如，我们知道输入线1后转向输入线2。如果复用器与分离器是同步的，这是可以保证的。在统计TDM中，由于输入和输出之间不存在指定或预定的时隙，它们之间没有固定的关系。需要在每个时隙中包含接收方地址以表明将要传送的地方。寻址最高效率形式可用定义N个不同输出线的n位表示，其中$n = \log_2 N$。例如，对于8条输出线，需要3位地址。



a) 同步TDM

b) 统计TDM

图6.26 TDM时隙比较

时隙大小

因为在统计TDM中时隙携带数据和地址，为了传输的效率，数据长度与地址长度之比率必须合理。例如，每个时隙发送1位数据而地址是3位，这样做效率很低。这就是说开销是百分之三百。在统计TDM中，通常一个数据块有许多字节而地址只有很少字节。

没有同步位

统计TDM与同步TDM还有一个不同，它没有同步位。这是由于此时，它是处于帧级，不需要同步。

带宽

在统计TDM中，链路的容量通常小于每个通道容量之和。统计TDM的设计者按照每条通道负载的统计确定链路的容量。如果输入时隙平均百分之x就认为满了，那么链路的容量反映这一情况。当然，峰峰时某些时隙需要等待。

6.2 扩频

复用把来自某些源端的信号组合在一起获得带宽的效率。链路的有用带宽在各个源端之间划分。在扩频（spread spectrum，SS）中，也把来自某些源端的信号组合在一起形成一个更宽的带宽，可是目的略有不同。扩频是为无线应用而设计的（LAN与WAN）。在这些类型应用中，我们所关注的比带宽效率更为重要。在无线应用中，所有站点都以空气（或真空）作为通信传输介质，共享这介质需要没有窃听者拦截，也需要没有恶意的入侵者的干扰（例如，军事行动）。

为达到此目的，扩频技术增加了冗余部分。扩展原始信号的频带满足每个站的需要。如果每个站要求的带宽是B，扩频将带宽扩展到B_{ss}，这里$B_{ss}>>B$。扩大的带宽允许源端用有防护的封装将它的报文进行更安全的传输，类似发送精美的礼品，把它放在特制礼盒中防止在运输过程中受到损坏，可用一种优级传递服务保证包装安全。

图6.27表示了扩频的思想，扩频通过两个原则达到它的目的：

1. 对每个站点需要分配的带宽显然要比它所需要的带宽更大；

2. 原来的带宽B扩大到B_{ss}必须由一个与原来的信号无关的过程来做。换言之，信号由源端生成后，扩频过程才发生。

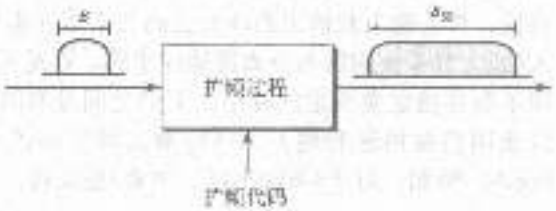


图6.27　扩频

信号由源端生成后，扩频过程利用扩频代码并扩大带宽。图中表示了原来的带宽B和扩大的带宽B_{ss}。扩频代码是伪随机二进制数据流，但它实际上是一种模式。

6.2.1 跳频扩频（FHSS）

跳频扩频（frequency hopping spread spectrum，FHSS）是用源信号调制M个不同的载波频率。在某一时刻用信号调制1个载波频率，在下一时刻信号调制另一个频率。虽然调制是一次使用一个频率，但在最终用了M个频率。源信号扩展后占用的带宽是$B_{FHSS}>>B$。

图6.28表示了FHSS的总的设计概况。称为**伪随机噪声**（pseudorandom noise，PN）的**伪随机代码生成器**（pseudorandom code generator）对每个**跳周期**T_h（hopping，T_h）生成一个k位模式。频率表使用这个模式查找频率作为这个跳周期的频率，而且通过它传送到频率合成器，频率合成器生成该频率的载波信号，同时源信号调制这个载波。

假定我们决定采用8个频率，这是实际应用最简单的例子，但恰好说明问题。此时，M是8，k是3，伪随机代码生成器将生

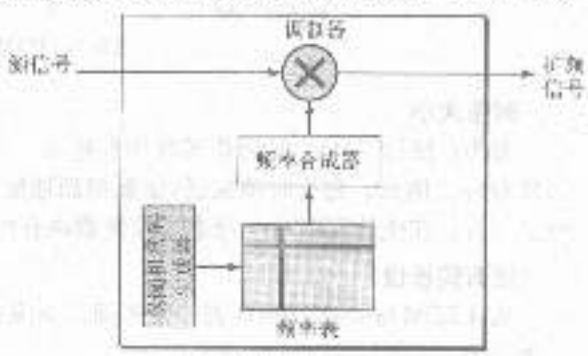


图6.28　跳频扩频（FHSS）

成8个不同的3位模式。这些映射到频率表中8个不同的频率（见图6.29）。

这个站点的模式是101，111，001，000，010，110，011，100。注意：模式是伪随机的，它在8次跳后重复循环。这就是说，在跳周期1，模式是101。所选的频率是700kHz，源信号调制这个载波频率。所选第2个模式是111，它选中900kHz的载波。第8个模式是100，所选的频率是600kHz。8次跳频后，模式重复，再从101开始。图6.30表示了信号如何从载波跳到载波的循环，并假定源信号要求的带宽是100kHz，

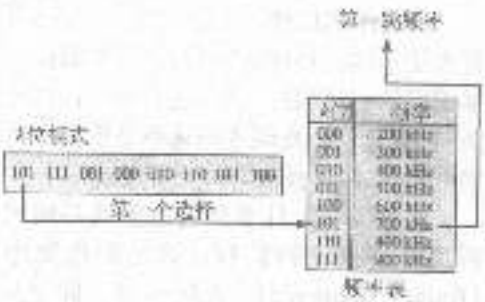


图6.29 FHSS中频率的选择

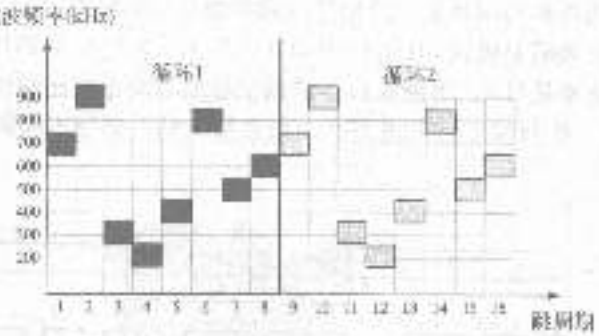


图6.30 FHSS循环

它可表示该方案能实现前面提到的目的。当有许多个k位组模式而跳周期是短的，那么发送方和接收方可具有保密性。如果入侵者企图窃听传输信号，他仅能获取很少一部分数据，因为他不知道扩频序列，不能很快地使他能适应下一跳变。一个怀恶意的发送者可能会将噪声发送到一个跳周期中（随机地）干扰信号，但不是全部周期。这个方案也有抗干扰的效果。

带宽共享

如果跳频数是M，那么可将M个通道复用为使用同一带宽B_c的一个通道。由于一个站点在每跳周期中只用一个频率，其他$M-1$个站点可使用另外的$M-1$个频率。换言之，如果使用适当的调制技术，比如多FSK（MFSK），M个不同站点可用同一带宽B_c。FHSS与FDM相似，如图6.31所示。

图6.31表示了使用FDM的4个通道与使用FHSS的4通道的例子。在FDM中，每个站点用带宽的$1/M$，但是固定的分配；在FHSS中，每个站点占用带宽的$1/M$，但跳到跳改变分配。

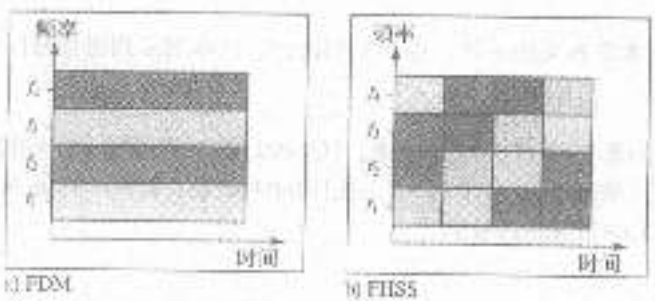


图6.31 带宽共享

6.2.2　直接序列扩频

直接序列扩频（direct sequence spread spectrum，DSSS）技术也是扩大源信号的带宽，但方法不同。在DSSS中，每个数据位用扩展编码的n位代替。换言之，每一位被编码为n个码片，此处码片的速率是数据比特率的n倍。图6.32表示了DSSS的概念。

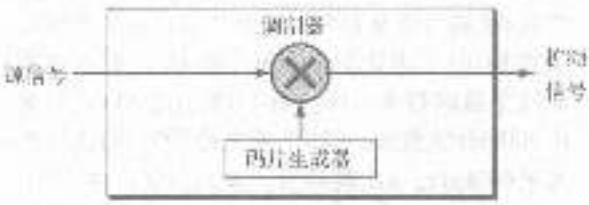


图6.32　DSSS

作为实例，让我们考察无线局域网中所用的直接序列扩频，著名的巴克序列（Barker sequence），此时n=11，假定源信号与码片生成器中的码片使用极性NRZ编码。图6.33表示了码片和带码片复用源信号得到扩频信号的结果。

在图6.33中，扩频码是模式10110111000（该例子中）的11位码片，如果源信号的速率是N，则扩频信号的速率是11N。这就是说，扩频信号要求的带宽比源信号带宽大11倍。如果入侵者不知道该编码，扩频信号可提供保密，如果每个站点使用不同的编码，它也可提供抗干扰的能力。

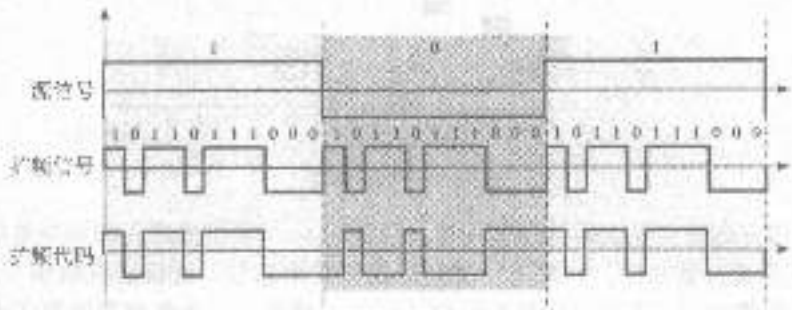


图6.33　DSSS的例子

带宽共享

在DSSS中能否像FHSS一样能共享带宽？回答不是，也可能是。如果我们使用一个扩频代码去扩频那不能组合和分离的信号（来自不同的站），则不能共享带宽。例如，将在第14章中看到，某些无线局域网使用DSSS，但扩频带宽不能被共享。然而，如果使用一个允许组合和分离的扩频信号的特殊类型的序列代码，则可共享带宽。将在第16章看到，一个特殊的扩频代码允许在移动电话中使用DSSS，某些用户可共享带宽。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物。在本书末列出括号[……]中的参考资料。

读物

复用在[Pea92]的第19章有详细的讨论，[Cou01]的3.9节到3.11节给出了完整的TDM和FDM的讨论，更深入资料可参考[Ber96]。在[Sta04]的第8章中，有复用的讨论。扩频可从[Cou01]的5.3节和[Sta04]的第9章找到。

本章小结

- 带宽的利用就是使用有用的带宽达到特殊的目的。复用可达到提高效率的目的；扩频可

达到保密和抗干扰的目的。

- 复用是一组技术，它是指多个信号通过一条单独的数据链路同时传输，n条线路共享该链路的带宽。名词链路是指物理通路；名词通道是指支持一个传输的链路一个部分。
- 基本复用技术有三种：频分复用、波分复用和时分复用。前两种技术是用于模拟信号，第三种是用于数字信号。
- 频分复用（FDM）是一种模拟技术，当一条链路的带宽（用赫兹表示）大于被传输信号的带宽总和时，可以采用该技术。
- 波分复用（WDM）是为具有高带宽的光纤电缆而设计的，WDM是组合光信号的一种模拟复用技术。
- 时分复用（TDM）是一种数字处理方法，它允许多个连接共享一条链路的高带宽。TDM是将一些低速率通道组合为一个高速率通道的数字复用技术。
- 可将TDM分为两种不同方案：同步的和统计的。在同步TDM中，每个输入连接即使不发送数据也分配输出。在统计TDM中，为了提高效率，时隙是动态分配的。
- 在扩频（SS）中，我们把来自不同源的信号组合在一起形成一个较大带宽。扩频是为无线应用而设计的，每个站点必须能共享没有窃听者拦截、也没有恶意入侵者干扰的介质。
- 跳频扩频（FHSS）是用源信号调制M个不同的载波频率。在某一时刻用该信号调制一个载波频率，在下一时刻，用该信号调制另一个频率。
- 直接序列扩频（DSSS）技术也扩大源信号的带宽，每个数据位用扩展编码的n位代替。换言之，每一位被编码为n个码片。

习题

复习题

1. 叙述复用的目的。
2. 列出本章所提出的主要复用技术。
3. 区别复用中的链路与通道。
4. 二种复用技术中哪些技术是用于组合模拟信号？哪些用于组合数字信号？
5. 解释电话公司使用的模拟结构层次并列出结构层次不同的级。
6. 解释电话公司使用的数字结构层次并列出结构层次不同的级。
7. 三种复用技术中哪一种技术经常用于光纤链路？并说明理由。
8. 区别多级TDM、多时隙TDM与脉冲填充TDM。
9. 区别同步TDM和统计TDM。
10. 解释扩频及其目的，列出本章所讨论的两种扩频技术。
11. 说明FHSS并解释它如何实现带宽扩大。
12. 说明DSSS并解释它如何实现带宽扩大。

练习题

13. 假定语音通道带宽为4kHz，需要用FDM复用10个语音通道，而防护频带为500Hz，试计算所要求的带宽。
14. 需要用一个20kHz带通通道传输100个数字语音通道。假定没有防护频带，每赫兹位的比率应该是多少？
15. 在图6.9中的模拟信号层次结构中，求每个层次（群、超群、主群和巨群）的开销（额外

的防护频带或控制的带宽）。

16. 20个数字信号源使用同步TDM实现多路复用，每个信号源的速率是100kbps，每个输出时隙携带来自每个数字信号源的一位，但帧需要增加一位用于同步。试回答下列问题：
    a. 以位为单位的输出帧的长度是多少？　　b. 输出帧速率是多少？
    c. 输出帧持续时间多长？　　d. 输出数据的速率是多少？
    e. 系统效率是多少（有用位与全体位之比）？
17. 如果每个输出时隙携带来自每个信号源的2位，重做习题16。
18. 14个数字信号源，每个信号源每秒生成500个字符（8位）。因为当前只有一些数字信号源是活跃的，所以使用字符交替的统计TDM实现复用。每帧每次携带6个时隙，但帧对每个时隙需要增加4位地址。试回答下列问题：
    a. 以位为单位的输出帧的长度是多少？　　b. 输出帧速率是多少？
    c. 输出帧持续时间多长？　　d. 输出数据的速率是多少？
19. 10个数字信号源，其中6个比特率为200kbps，另外4个比特率为400kbps，使用无同步位多级TDM实现复用，对复用的最后一级，回答下列问题：
    a. 以位为单位帧长度是多少？　　b. 帧速率是多少？
    c. 帧持续时间多长？　　d. 数据速率是多少？
20. 4个通道，其中2个比特率为200kbps，另外2个比特率为150kbps。使用无同步位多时隙TDM实现复用，试回答下列问题：
    a. 以位为单位帧长度是多少？　　b. 帧速率是多少？
    c. 帧持续时间多长？　　d. 数据速率是多少？
21. 2个通道，一个速率为190kbps，另一个速率为180kbps。使用无同步位脉冲填充TDM实现复用，试回答下列问题：
    a. 以位为单位帧长度是多少？　　b. 帧速率是多少？
    c. 帧持续时间多长？　　d. 数据速率是多少？
22. 关于T-1线回答下列问题：
    a. 一帧持续时间多长？　　b. 开销是多少（每秒额外位的个数）？
23. 使用同步TDM 实现复用，共4个信号源按下列发送字符（注意：字符按它们键入的顺序发送，第三个源端不发送数据）。试表示出5个输出帧的内容：
    a. 信号源1信息:HELLO；　　b. 信号源2信息:HI；
    c. 信号源3信息；　　d. 信号源4信息:BYE。
24. 图6.34表示了使用同步TDM的一个复用器。如果输出时隙长度只有10位（每个输入取3位加上1个帧指示位），那么输出流是什么？到达复用器的位的顺序如箭头所示。

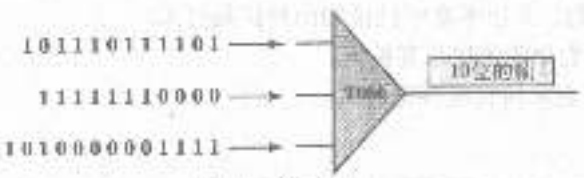


图6.34　练习24

25. 图6.35表示了使用同步TDM的一个分离器。如果输入时隙长度是16位（忽略帧指示位），那么每条输出线的位流是什么？到达分离器的位的顺序如箭头所示。

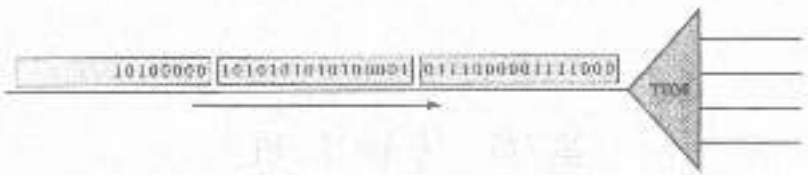


图6.35 练习25

26. 关于图6.23的数据层次结构，试回答下列问题：
    a. 在DS-1服务中，开销（附加位的个数）是多少？
    b. 在DS-2服务中，开销（附加位的个数）是多少？
    c. 在DS-3服务中，开销（附加位的个数）是多少？
    d. 在DS-4服务中，开销（附加位的个数）是多少？
27. 如果对于一个带宽B = 4kHz而B_{ss} = 100kHz的通道使用FHSS，试问PN序列最小位的个数是多少？
28. 某个FHSS系统使用4位PN序列。如果PN的比特率是64kbps，试回答下列问题：
    a. 可能的跳的总数是多少？　　b. 一个完整的PN的循环需要多少时间？
29. 一个伪随机数生成器用下面公式生成随机数序列：

$$N_{n+1} = (5 + 7N_n) \bmod 17 - 1$$

其中N_n是当前的随机数，N_{n+1}是下一次的随机数，mod意思是指用$(5 + 7N_n)$被17除的余数。

30. 一个数据速率为10Mbps的介质，如果使用巴克序列的DSSS，该介质能携带多少个64kbps的语音通道？

第7章 传输介质

在第3章到第6章，我们讨论了许多与物理层相关的问题。本章讨论传输介质。传输介质实际上位于物理层以下并直接由物理层控制。可以说，传输介质属于第0层。图7.1说明了传输介质相对于物理层的位置关系。



图7.1 传输介质与物理层

传输介质（transmission medium）可广义地定义为能从源端传送信息到目的端的任何东西。例如，用餐的二个人交谈时，传输介质是空气，空气也能用烟信号或旗语传送信息。对于一封书信，传输介质可能是邮递员、卡车和飞机。

在数据通信中，信息和传输介质的定义更加特定。传输介质通常是自由的空间、金属的电缆或光纤。信息通常是一种信号，是来自另一种形式的数据转换的结果。

利用电信号进行长途通信开始于19世纪，Morse发明的电报，电报通信的速度很慢并依赖于金属的电缆。

当电话在1869年被发明的时候，人类声音的通信成为了可能。那时电话通信需要一种金属的介质将人类声音的转变换成信号进行传送。然而，由于电线的质量不佳，通信是不可靠的。线路时常有噪声，并且技术也很简单的。

当赫兹（Hertz）能够发送高频信号时，1895年开始有了无线通信。稍后，马可尼（Marconi）设计了一个能跨越大西洋传输电报类型信息的方法。

现在已取得了很大进展，已发明了较好的金属介质（例如，双绞线、同轴电缆），光纤的使用已经极大地增加了数据速率，自由空间（空气、真空和水）被更有效地利用，有一部分运用前面章节中讨论的技术。

正如第3章所述，计算机和其他电信设备用信号表示数据。这些信号以电磁能的形式在设备之间传输。电磁能通过传输介质传播。

电磁能是电场和磁场相互振荡的组合形式，它包括电力、无线电波、红外线、可见光、紫外线、X射线、伽马射线和宇宙射线。每一种都对应于电磁频谱（electromagnetic spectrum）的一部分。但是目前，并非所有频谱都可用于电子通信。这些可利用频谱的介质也只限于有限的几类。

按照电信中的用途，传输介质可以分为两大类：有向的和无向的。有向介质包括双绞线、同轴电缆和光缆，无向介质通常是空气。图7.2说明了分类情况。

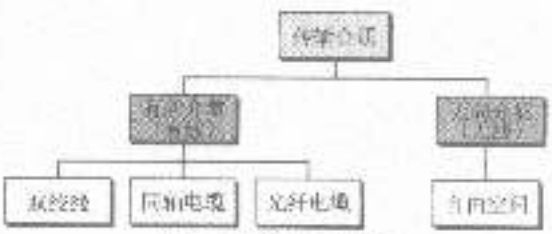


图7.2 传输介质的分类

7.1 有向介质

有向介质（guided media）是指那些在设备之间提供通路的介质，包括双绞线（twisted-pair cable）、同轴电缆（coaxial cable）和光缆（fiber-optic cable）。沿着这类介质传输的信号，其传输方向和传播范围受介质的物理边界限制。双绞线和同轴电缆使用金属（铜）导体接收和传输电流形式的信号。光纤（optical fiber）是一种玻璃线缆，接收和传输光波形式的信号。

7.1.1 双绞线

双绞线由两根导线（通常是铜线）构成，其中的每一根导线都有自己的塑料绝缘层，两者绞在一起，如图7.3所示。



图7.3 双绞线

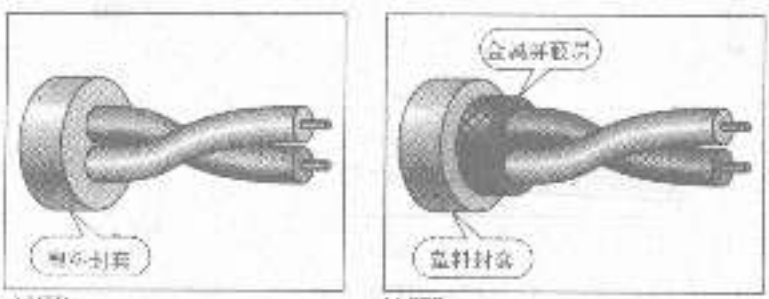
线路中的一条用于将信号传送到接收方，另一条仅用作接地参考点。接收方使用两条线路的电平之间的差值。

除了发送方在线路上发送的信号之外，干扰（噪声）和串扰也可能会影响两条线路并产生有害的信号。

如果两条导线是平行的，这些有害信号的影响在两条线路中就会不同，因为它们相对于噪声或者串扰所处的位置是不同的（例如，一条较近而另一条较远）。这会在接收方产生差值。通过将两条线绞合在一起，可以保持平衡。例如，假定在一个绞合段，一条导线距噪声源较近而另一条导线距噪声源较远，而在下一个绞合段，情况刚好相反。绞合可以使两条线路同等地受到外部影响（噪声或者串扰），这意味着，接收方不会通过计算两者之间的差值接收到有害信号。从上面的讨论中可以很明显地看出，单位长度（例如，英寸）的绞合数决定了电缆的质量，绞合数越多意味着质量更高。

非屏蔽双绞线与屏蔽双绞线

通信中使用的最常见的双绞线是指非屏蔽双绞线（unshielded twisted-pair，UTP）。IBM还生产了一种自己使用的双绞线类型，称为屏蔽双绞线（shielded twisted-pair，STP）。STP电缆有一层金属薄片或者织成网状的包覆材料将每一对带有绝缘层的导体包围起来。尽管金属包装层可以通过防止噪声或者串扰而提高电缆的质量，但是它更加笨重和昂贵。图7.4说明了UTP和STP之间的差别。我们讨论的中心主要是UTP，因为STP除了IBM以外很少使用。



a) UTP　　b) STP

图7.4 UTP和STP

分类

电子工业协会（EIA）已经制定了一个将非屏蔽双绞线分为七类的标准。分类由电缆的质量确定，第1类是最低的，第7类是最高的。每一种EIA分类适用于特定的用途。表7.1说明了这些分类。

表7.1　非屏蔽双绞线的分类

分　类	说　明	数据率（Mbps）	用　途
1	电话中所用的非屏蔽双绞线	<0.1	电话
2	本来用于T1线路的非屏蔽双绞线	2	T1线路
3	改进的CAT2，用于局域网	10	局域网
4	改进的CAT3，用于令牌环网	20	局域网
5	电缆线通常带有一个在下列外面护层的24AWG	100	局域网
5E	5类线的扩充包括最小带长和抗电磁干扰的额外标准	125	局域网
6	数据率超过200Mbps级类的一种新型电缆线	200	局域网
7	有时又称为SSTP（屏蔽屏屏双绞线）。每对双绞线被单独屏蔽在塑料外壳内，一个金属薄膜包裹着一个金属线编织品中，在外面再加屏蔽。屏蔽减少串扰和增加数据率	600	局域网

连接器

最常用的UTP连接器是RJ45（RJ表示注册插头），如图7.5所示。RJ45是一种有插槽的连接器，表示只能以一个方向插入。

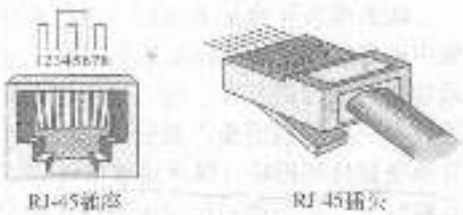


图7.5　UTP连接

性能

测试双绞线性能的一种方法是对比频率和距离的衰减曲线。双绞线允许通过的频率范围比较宽。但是，图7.6说明随着频率的升高，以每英里的分贝数进行测量，在频率超过100kHz时，衰减数值会突然升高。注意：规格（gauge）是电线粗细的标准计量单位。

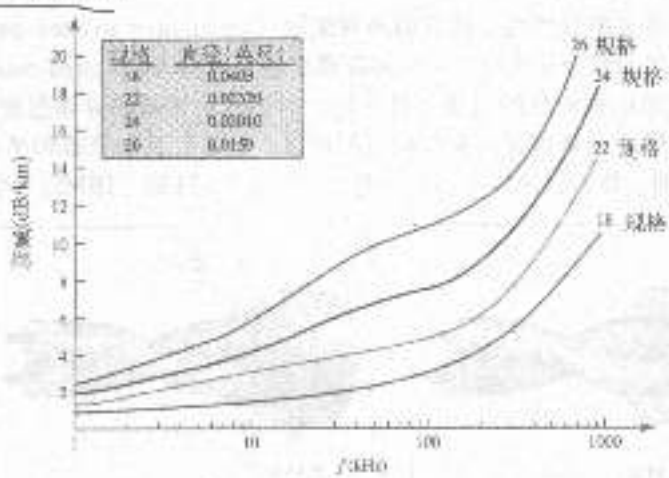


图7.6　UTP的性能

应用

双绞线用于电话线路，提供语音和数据的通道。本地环路，即连接用户到中心电话机房的线路，最常用的是非屏蔽双绞线。第9章讨论电话网络。

电话公司使用DSL线路提供高数据速率连接，这种线路也使用具有高带宽容量的非屏蔽双绞线。第9章讨论DSL技术。

局域网，如10Base-T和100Base-T，也使用双绞线。第13章讨论这些网络。

7.1.2 同轴电缆

同轴电缆（或者称为同轴）与双绞线相比，可以传输更高频率范围的信号，部分原因是这两种介质的构造有很大的不同。同轴电缆不使用两根电线，而是使用一根位于中央的实心或者多股绞合的核心金属丝导体（通常是铜的）。导体封装在绝缘护套中，然后再把它封装在金属箔、金属网或者两者的组合成的外部导体中。外部金属包装既可以屏蔽噪声，又可以作为第二导体，构成回路。外部导体的外面再由绝缘护套封装，最后整条电缆由一层塑料外套保护（见图7.7）。



图7.7 同轴电缆

同轴电缆标准

同轴电缆是按照它们的无线电管理（radio government，RG）等级进行分类的。每一个RG号表示单独的一组物理规范，包括内部导体的电线规格、内部绝缘体的粗细和类型、屏蔽层的结构以及外部封套的尺寸和类型。RG等级定义的每一种电缆适用于特定的功能，如表7.2所示。

表7.2 同轴电缆的分类

分 类	阻 抗	用 途
RG-59	75Ω	有线电视
RG-58	50Ω	粗缆以太网
RG-11	50Ω	粗缆以太网

同轴电缆连接器

为了将同轴电缆和设备相连，需要使用同轴电缆连接器。目前最常见的连接器类型是BNC（Bayonet-Neil-Concelman）连接器。图7.8表示了这些连接器中三种常用的类型：BNC连接器、BNC T型连接器和BNC终接器。

BNC连接器用于将电缆的一端连接到设备如电视机上。BNC T型连接器用于以太网（见第13章），从主线路引出分支以连接计算机或其他设备。BNC终接器用于电缆的末端以防止信号反射。

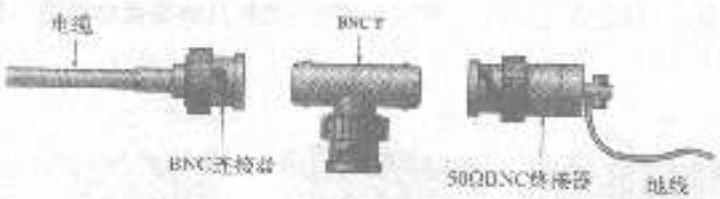


图7.8 BNC连接器类型

性能

同双绞线一样，我们也能测试同轴电缆的性能。从图7.9中可以看出，同轴电缆的衰减要

高于双绞线。换句话说，尽管同轴电缆有更高的带宽，但是信号减弱很快且面经常需要使用中继器。

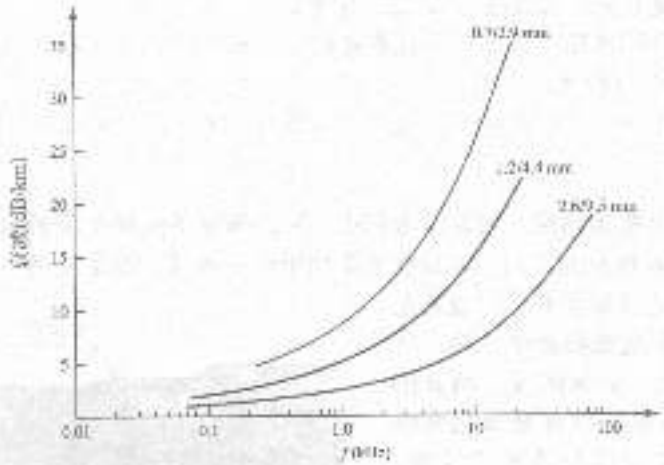


图7.9　同轴电缆的性能

应用

同轴电缆最初是在模拟电话网络中使用的，单个的同轴电缆网络能够传送10 000路语音信号。后来将它用于数字电话网络中，单个的同轴电缆能够以高达600Mbps的速率传送数据。但是，电话网络中的同轴电缆目前大部分已经被光缆取代了。

有线电视网络（见第9章）也使用同轴电缆。在传统的有线电视网络中，整个网络都使用同轴电缆。但是，后来有线电视提供商用光缆取代了网络中的大部分的同轴电缆，混合网络只是在网络边界、靠近消费者房屋的地方使用同轴电缆。有线电视使用RG-59同轴电缆。

另一种常见的应用是在传统的以太局域网中（见第13章）。由于它的带宽高以及由此产生的高速率，因此在早期的以太局域网中选用同轴电缆进行数字传输。10Base-2或者细缆以太网，使用RG-58同轴电缆和BNC连接器以10Mbps的速率传输数据，传输距离是185m。10Base5或者粗缆以太网，使用RG-11（粗同轴电缆）以10Mbps的速率传输数据，传输距离是5 000m。粗缆以太网有专用的连接器。

7.1.3　光缆

光缆由玻璃或者塑料构成，能够传输光信号。为了理解光纤，首先需要研究光的几个本质特性。光在通过单一物质时，是沿直线传播的。如果光线从一种物质突然进入另一种物质（密度更高或者更低），则会改变方向。图7.10说明了光线从高密度物质进入低密度物质时是如何改变方向的。



图7.10　光线的折射

如图7.10所示，如果入射角（angle of incidence）（入射光线与界面上入射点法线的夹角）小于临界角（critical angle），光线会沿着较接近于表面的方向发生折射（refract）；如果入射角等于临界角，光线则会沿着表面弯曲；如果入射角大于临界角，光线则会反射（reflect，发生转向）并再次在高密度物质中传输。注意，临界角是物质的一种特性，不同物质的临界角的值是不同的。

光纤使用反射来引导光通过通道。玻璃芯或塑料芯外面环绕着低密度的玻璃的或塑料的包层（cladding），两种材料的密度差值必须达到如下条件，即通过纤芯传播的一束光必须完全被包层反射，而不发生折射。见图7.11。

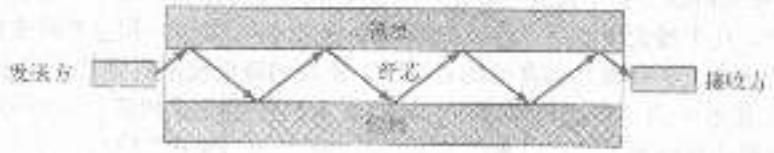


图7.11 光纤

传播模式

目前的技术支持两种模式（多模和单模），通过这两种模式光线沿着光通道传播，每一种模式需要光纤能具有不同的物理特性。多模传输有两种实现形式：阶跃折射率模式和渐变折射率模式（见图7.12）。

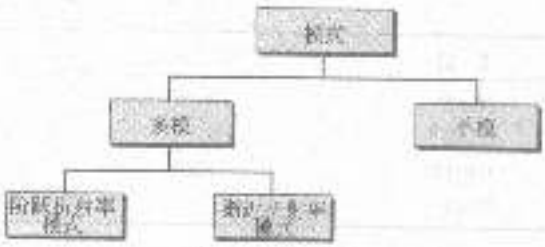


图7.12 传播模式

多模 之所以叫做多模，是因为来自光源的多个光束沿着不同的路径通过纤芯。这些光束如何在光缆内部通过取决于纤芯的结构，如图7.13所示。

在**多模阶跃折射率光纤**（multimode step-index fiber）中，纤芯的密度从中心到

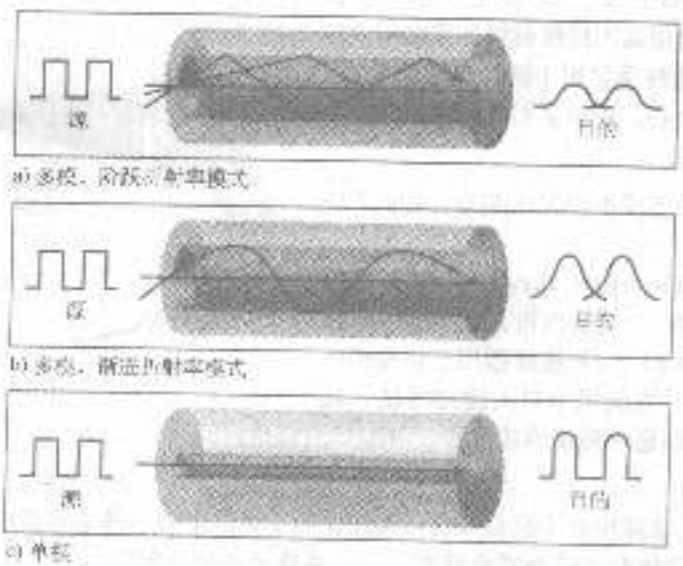


图7.13 传播模式

边缘保持不变。光束到达纤芯与包层交界处之前，以直线的形式通过等密度的纤芯。在交界处，突然遇到低密度的包层会引起光束传输角度的变化。阶跃折射率（step index）这个术语是指这种变化的突然性。

第二种类型的光纤，称为多模渐变折射率光纤（multimode graded-index fiber）。通过这种光缆的信号可以减少失真。指数（index）这个词是指折射率。

正如上面所看到的，折射率与密度有关。所以，渐变折射率光纤是一种具有不同密度的光纤。密度在纤芯的中心处最高，从中心到边缘逐渐降低，在边缘处最低。图7.13说明了密度的变化对光束传播的影响。

单模　单模使用阶跃折射率光纤并且使用聚焦效果更好的光源，这样可以把光束限制在更小的角度范围内，几乎接近于水平。单模光纤（single-mode fiber）在生产时使用直径比多模光纤要小得多且密度（折射率）非常低的光纤维。密度的降低使临界角几乎接近90°，这样光束的传播基本上是水平的。在这种情况下，不同光束的传播几乎相同，从而可以忽略延迟。所有的光束同时到达目的端，并可以重组为失真很小的信号（见图7.13）。

光纤规格

光纤是通过纤芯的直径与光纤包层的直径的比率定义的，两者都用微米（mm）表示。常见规格如表7.3所示。注意：最后一行列出的规格仅用于单模光纤。

表7.3　光纤类型

类　型	纤芯（μm）	包层（μm）	模　式
50/125	50.0	125	多模，渐变折射率模式
62.5/125	62.5	125	多模，渐变折射率模式
100/125	100.0	125	多模，渐变折射率模式
7/125	7.0	125	单模

光缆组成

图7.14说明了一种典型光缆的组成。外部封套由PVC或者特氟隆制成。封套里面是用于提高光缆强度的Kevlar纤维。Kevlar是一种用于制造防弹背心的强度很高的纤维材料。在Kevlar层的下面是另一层塑料薄层用于缓冲光纤的受力。光纤位于光缆的中心，由包层和纤芯构成。



图7.14　光纤组成

光缆连接器

光缆使用三种不同类型的连接器，如图7.15所示。

用户通道（subscriber channel，SC）连接器用于有线电视中，它采用推拉式固定方法。**直插式**（straight-tip，ST）连接器用于将光缆连接到网络设备上，它使用卡口式固定方法，比SC更可靠。MT-RJ是一种新的连接器，与RJ45的规格相同。

性能

在图7.16中，衰减相对于波长的变化曲线说明了光缆中的一种非常值得注意的现象，衰减曲线比双绞线和同轴电缆的衰减曲线要平坦。其性能如此之好，以至于在使用光纤时，只需要很少的中继器（实际上少于1/10）。

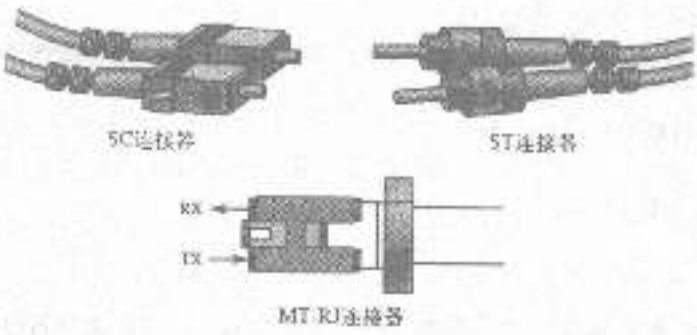


图7.15 光缆连接器

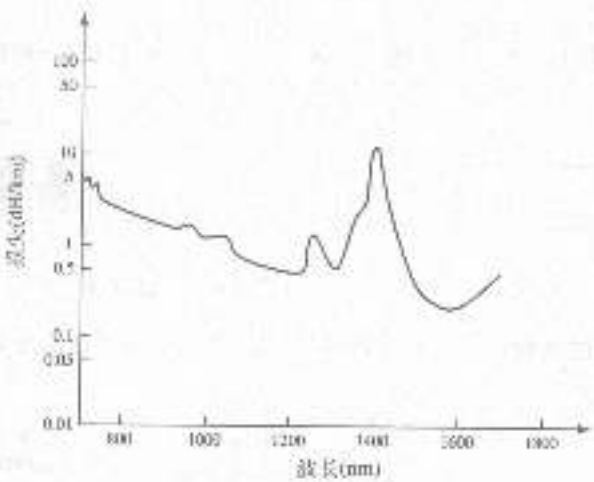


图7.16 光纤性能

应用

光纤通常用在主干网络中，因为它提供了很高的带宽，所以其性能成本是合算的。目前，对于WDM，能以1 600Gbps的速率传输数据。第17章讨论的SONET就提供了这样的主干网。

一些有线电视公司结合使用光纤和同轴电缆，可以构建混合网络。光纤提供主干结构，而同轴电缆则提供到用户住所的连接。这是一种性能成本合算的配置方案，因为用户端所需的带宽窄，使用光纤在经济上不合算。

局域网，如100Base-FX网（快速以太网）和1 000Base-X网也使用光缆。

光纤的优点和缺点

优点 与金属电缆（双绞线或者同轴电缆）相比，光缆具有以下优点：

- **高带宽** 光缆能够支持比双绞线或同轴电缆高得多的带宽（数据速率）。目前，光缆的数据速率和带宽利用所受到的不是介质的限制，而是受到信号生成和接收技术的限制。
- **信号衰减小** 光纤传输距离明显高于其他有向介质。信号在不需要再生的情况下可以传输50km。对于同轴电缆或者双绞线，每隔5km就需要一个中继器。
- **无电磁干扰** 电磁噪声不能影响光缆。
- **抗腐蚀材料** 玻璃比铜抗腐蚀能力更强。
- **重量轻** 光缆比铜缆要轻得多。
- **不易被窃听** 光纤明显比铜缆更不易被窃听。铜缆会形成天线，易于被窃听。

缺点 光纤在使用中也有一些缺点：

- **安装/维护** 光纤是一种较新的技术，安装和维护需要专门技能，这种技能并不是在任何地方都有的。
- **单向性** 光的传播是单向的，如果需要进行双向通信，则需要两根光纤。
- **成本** 光缆和接口比其他有向介质的昂贵一些。如果对带宽的需求不是很高，通常情况下使用光纤不合算。

7.2 无线通信

无向介质（unguided media）不使用物理导体传输电磁波。这种类型的通信通常是指**无线通信**（wireless communication）。信号通常通过空气广播，能够被任何人接收，只要他拥有一台接收信号的设备。

图7.17表示了用于无线通信的部分电磁频谱，范围为3kHz～900THz。

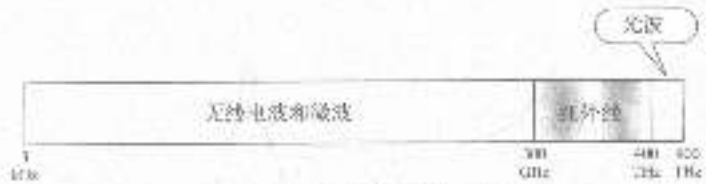


图7.17 无线通信使用的电磁频谱

无向信号能通过多种途径从信号源传输到目的地，其方法有地表传播、天空传播和视线传播，如图7.18所示。



图7.18 传播方法

在**地表传播**（ground propagation）中，无线电波通过地球的大气层的最低部分紧绕地球传播。这些低频率信号通过发射天线沿着地球表面向各个方向发射，传输距离取决于信号功率的大小，功率越强，传输的距离越远。在**天空传播**（sky propagation）中，较高频率的无线电波向上传播到电离层（大气中的一层，该层中粒子以离子的形式存在），从电离层再反射回地球。这种类型的传输可以使用较低的输出功率，传输更远的距离。在**视线传播**（line-of-sight propagation）中，非常高频率的信号沿着直线直接在天线之间传输。天线必须是有向的、互相正对的，而且必须足够高或距离足够近，以免受地球曲率的影响。视线传播是很难处理的，因为无线电传输不能完全聚焦。

电磁频谱中，定义为无线电波和微波的部分被分成八个区域，称为**波段**（band），每一个波段都受政府机构的管制。这些波段从甚低频率（very low frequency，VLF）到极高频率（extremely high frequency，EHF）依次排列。表7.4列出了这些波段及其频率范围、传播方法和应用。

表7.4 波段

波 段	范 围	传 播	应 用
VLF（甚低频）	3～30kHz	地面	远程无线电导航
LF（低频）	30～300kHz	地面	无线电导航台和广播定位信
MF（中频）	300～3MHz	大空	AM广播
HF（高频）	3～30MHz	天空	民用波段（CB）和航海航空通信
VHF（甚高频）	30～300MHz	天空和视线	VHF电视波段和FM广播
UHF（超高频）	300～3GHz	视线	UHF电视波段，移动电话，寻呼，卫星
SHF（极高频）	3～30GHz	视线	卫星通信
EHF（极高频）	30～300GHz	视线	雷达和卫星

可以把无线传输分为三大组：无线电波、微波和红外波，如图7.19所示。



图7.19 无线传输波

7.2.1 无线电波

尽管无线电波和微波之间没有明确的界限，但是通常将频率范围在3～1GHz之间的电磁波称为无线电波（radio wave），频率范围在1～300GHz的电磁波称为微波（microwave）。但是，我们使用波的特性而不是频率的特性作为分类的标准。

大部分无线电波是全方向的，在天线发射无线电波时，它们会向多个方向传播。这意味着发射天线和接收天线不必对准。发射天线发射的无线电波，可以被任何接收天线接收。全方向特性也有一个缺点：一只天线发射的无线电波，容易受到另一只使用相同频率或波段的天线所发射的信号的干扰。

无线电波，尤其是那些以天空模式传播的电波，可以传输很长的距离。这使得无线电波成为诸如AM广播等远距离广播的优选。

无线电波，特别是那些低、中频率的电波，可以穿透墙体。这种特性既有优点也有缺点。说它有优点，是因为如AM收音机可以在建筑物内接收信号。说它有缺点，是因为不能隔离建筑物内部和外部的通信。无线电波的波段与微波相比较窄，仅在1 GHz以下。当这一波段被划分为多个子波段时，子频带也很窄，导致在数据通信中数据速率很低。

几乎整个波段都受政府机构（比如，在美国是FCC）的管制。使用波段的任何部分必须得到这些机构的许可。

全向天线

无线电波使用全向天线（omnidirectional antennas）向所有方向发送信号。根据波长，强度和传输目的的不同，可以有多种类型的天线。图7.20说明了一种全向天线。


图7.20 全向天线

应用

无线电波的全向特性使得它对多播很有用，在多播过程中，有一个发送方和多个接收方。AM和FM收音机、电视、航海无线电、无绳电话和寻呼机都是多播的例子。

> 无线电波用于多播通信，如收音机、电视以及寻呼系统。

7.2.2 微波

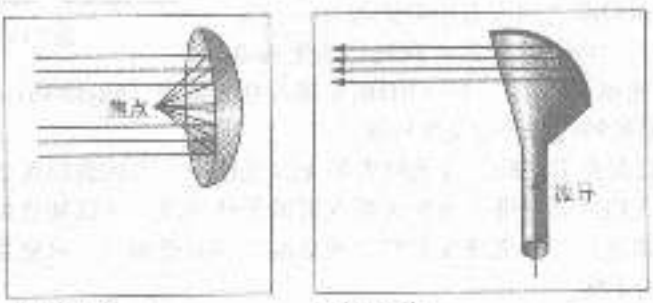
频率范围为1～300GHz的电磁波称为微波。

微波是单向的。在天线发射微波时，它们可以聚集在很窄的范围内。这意味着发射天线和接收天线需要对准。单向特性具有一个很明显的优势，即一对对准的天线不受另一对对准天线的干扰。

- 微波传播属于视线传播。安装天线的塔必须相互可见。如果相距很远，天线塔就必须很高。地球的曲率和其他障碍物不允许两座很低的天线塔使用微波进行通信。长距离通信时，通常需要中继站。
- 其高频微波不能穿透墙体。在接收机位于建筑物内部时，这种特性是一种缺点。
- 微波波段相对较宽，接近于299GHz。所以，它能够分配更高带宽的子波段而且可以获得更高的数据速率。
- 要使用波段中的某个部分，就必须得到政府机构的许可。

单向天线

微波需要单向天线（unidirectional antennas）在一个方向发射信号。微波通信使用两种类型的天线：抛物面碟形天线和喇叭天线（见图7.21）。



a）碟形天线　　b）喇叭天线

图7.21　单向天线

抛物面碟形天线（parabolic dish antenna）基于抛物面的几何形状：平行于对称线的每条光线（视线）在曲面上以不同角度反射回来，以使所有的光线相交于一个公共点，这个点称为焦点。抛物面碟形天线的工作原理如同一个漏斗，接收到较宽范围的微波后，将它们导向到一个公共点。与使用单点接收机相比，通过这种方式可以恢复更多的信号。

外发的传输是通过对准抛物面碟形天线的喇叭天线广播出去的。微波遇到抛物面碟形天线时，会以与接收路径相反的方向偏转出去。

喇叭天线（horn antenna）看起来像一个巨大的勺子。外发的传输沿着主干（像一个柄）广播，通过弯曲的头部以一系列范围较窄的平行波束的形式偏转出去。接收到的传输由勺形的喇叭天线采用与抛物面碟形天线类似的方式汇聚起来，偏转入主干中。

应用

由于微波具有单向性，因而在发送方和接收方之间需要单播（一对一）通信时非常有用。微波应用于移动电话（第16章）、卫星网络（第16章）和无线局域网（第14章）。

微波用于单播通信，如移动电话、卫星网络和无线局域网。

7.2.3　红外波

红外信号（infrared wave）的频率范围为300GHz～400THz（波长为1mm～770nm），可以用于短距离通信。红外信号的频率很高，不能穿透墙体。这种优点可以防止系统之间相互干扰，在一个房间内的短距离通信不会受到相邻房间的另一个系统的影响。在使用红外遥控

器时，不会干扰邻居红外遥控器的使用。但是，这种特性也使得红外信号无法进行长距离通信。另外，在建筑物外面不能使用红外波，因为太阳射线中包含可能干扰通信的红外波。

应用

红外波段，接近于400THz，对数据传输有非常大的潜力。这样一种高带宽能以高数据速率传输数字数据。红外数据联盟（Infrared Data Association，IrDA）负责红外波的使用，为了使用这些信号用于设备（如键盘、鼠标、个人计算机和打印机等）之间的通信，该联盟已经建立了一些相应的标准。例如，一些生产商提供一种特殊的接口称为IrDA接口，支持无线键盘与PC的通信。该标准最初确定最大传输距离为8 m时的数据速率是75kbps，最新标准确定的数据速率是4Mbps。

IrDA确定的红外信号通过视线传播，键盘上的红外接口需要指向PC才能进行传输。

红外信号可以在封闭区域用于短距离通信，使用视线传播。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物，在本书末列出括号[……]中的参考资料。

读物

在[GW04]3.8节，[Sta04]第4章和[Tan03]2.2节及2.3节讨论传输介质。[SSS05]给出传输介质完整讨论。

本章小结

- 传输介质位于物理层以下。
- 有向介质在设备之间提供一条物理通道。双绞线、同轴电缆和光纤是最常用的有向介质。
- 双绞线由两根绞合在一起的互相绝缘的铜线组成。双绞线电缆用于语音和数据通信。
- 同轴电缆含有中心导体和屏蔽层。同轴电缆比双绞线更能携带高频范围的信号，它用于有线电视网和传统的以太局域网。
- 光缆由玻璃或塑料纤芯和环绕纤芯的包层构成，纤芯和包层封装在封套中。光纤传送光形式的数据信号，信号通过反射沿着内部纤芯传播。光纤的传输变得日益普及，原因在于它抗干扰、低衰减和高带宽性能。光纤用于主干网、有线电视网和快速以太网。
- 无向介质（通常指空气）不使用物理导体传输电磁波。
- 无线数据传输的形式有地面传播、天空传播和视线传播，无线波可分为无线电波、微波和红外波，无线电波是全向的，微波是单向的。微波用于移动电话、卫星和无线局域网通信。
- 红外波用于短距离通信，如PC与外部设备之间的通信，也可用于室内局域网通信。

习题

复习题

1. 在OSI模型或因特网模型中，传输介质位于什么位置？
2. 试说出传输介质的两种主要类型。
3. 有向介质与无向介质有什么不同？
4. 有向介质主要有哪三类？

5. 双绞线缆中的绞合有什么作用？
6. 什么是折射？什么是反射？
7. 光纤中包层的用途是什么？
8. 试说出光纤与双绞线和同轴电缆相比的优点。
9. 天空传播与视线传播有什么不同？
10. 全向波和单向波有什么不同？

练习题

11. 依照图7.6，按指出的频率和距离对规格18非屏蔽双绞线（18-gauge UTP）的衰减填写下表。

表7.5　规格18-UTP的衰减

距　离	1kHz时衰减 dB	10kHz时衰减dB	100kHz时衰减dB
1km			
10km			
15km			
20km			

12. 使用练习11的结果，推论出UTP电缆的带宽随距离的增加而减少。
13. 如果使用规格18的UTP在1km开始处的功率是200mw，对于频率分别为1kHz、10kHz和100kHz的结束处的功率是多少？使用练习11的结论。
14. 依照图7.9，按指出的频率和距离对2.6/9.5mm同轴电缆的衰减填写下表。

表7.6　2.6/9.5mm同轴电缆的衰减

距　离	1kHz时衰减 dB	10kHz时衰减dB	100kHz时衰减dB
1km			
10km			
15km			
20km			

15. 使用练习14的结果，推论出同轴电缆的带宽随距离的增加而减少。
16. 如果使用2.6/9.5mm同轴电缆在1km开始处的功率是200mw，对于频率分别为1kHz、10kHz和100kHz的结束处的功率是多少？使用练习14的结论。
17. 对于下列波长范围，试计算光波的带宽（假定光速是2×10^8m）：

a. 1 000nm到1 200nm　　b. 1 000nm到1 400nm

18. 图7.6和图7.9的水平轴表示频率，而图7.16的水平轴表示波长。你能说明其理由吗？如果光纤传播速度是2×10^8m，你能改变频率在水平轴的单位吗？垂直轴应该也被改变吗？曲线应该也改变吗？
19. 依照图7.16，按指出的波长和距离对光缆的衰减（以dB表示）填写下表。

表7.7　光纤的衰减

距　离	波长800nm时dB	波长1 000nm时dB	波长1 200nm时dB
1km			
10km			
15km			
20km			

20. 光信号通过光纤传播，试问如果光缆长度分别是10m、100m和1km（假定光纤传播速度是2×10^8m），信号延迟是多少？

21. 一束光从一种介质到另一种密度较低的介质时，临界角是60°，当入射角是下列角度时，有没有折射或反射？并画出光线通过这两种介质时的路径。

a. 40°　　b. 60°

c. 80°

第8章 交　　换

网络是连接设备的一组集合。无论何时只要拥有了多台设备，就会产生这样一个问题，即如何将它们连接起来实现一对一的通信。一种解决方案是在每对设备之间建立点到点连接（网状拓扑），或者在一台中心设备与其他所有设备之间建立点到点的连接（星形拓扑）。但是，在应用于规模很大的网络时，这些方法不切实际且很浪费。链路的数量和长度将要太多的基础设施从而使成本效益很低，而多数链路在大多数时间内是空闲的。其他拓扑结构采用了多点连接，如总线拓扑，这种方案也被排除，原因是设备之间的距离和设备总数的增长会超出介质和设备的容量。

一种更好的解决方案是**交换**（switching）。交换网络由一系列互连的节点构成，这些节点称为**交换机**（switch）。交换机能够在链接到交换机的两台或多台设备之间建立临时连接。在交换网络中，一些节点连接到端系统（比如，计算机或电话系统），而另一些只是用于路由选择。图8.1表示了一个交换网络。

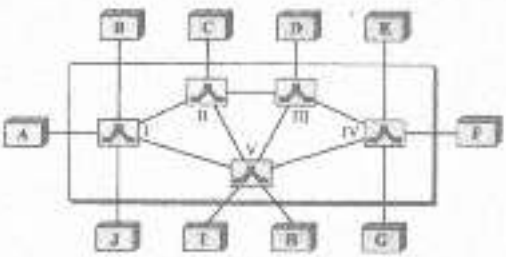


图8.1　交换网络

端系统（end sysyem）（通信设备）标以A、B、C、D等，而交换机标以I、II、III、IV和V，每个交换机连接多条链路。

传统上，有三种重要的交换方式：电路交换、分组交换和报文交换。前两种目前普遍被用，第三种在一般通信中已逐步淘汰，但是仍然存在于网络应用中。然后我们将当前网络分成二大类：电路交换网、分组交换网和报文交换网。分组交换网进一步划分二个子类：虚电路网和数据报网，如图8.2所示。



图8.2　交换网分类法

我们能说虚电路网具有与电路交换网与数据报网同样的一些特性。所以，先讨论电路交换网和数据报网，后再讨论虚电路网。

目前，分组交换网的趋向是结合数据报网和虚电路网，网络根据数据报寻址的思想对第一个分组进行路由选择，然后为同一个源端和目的端的所有其余的分组建立一个虚电路网，在后面几章将会看到一些这样的网络。

在报文交换中，每个交换机存储整个报文，并转发到下一个交换机。尽管在低层看不见报文的交换，但它还是用于某些应用中，如电子邮件（E-mail）。本书不讨论该主题。

8.1　电路交换网络

电路交换网络（circuit-switched network）是由物理链路连接的一组交换机组成的。两个

站点的连接是由一条或多条链路组成的专用路径来实现。然而，每次连接仅使用每条链路上的一专用通道。通常每条链路用FDM或TDM划分成n个通道，如在第6章中讨论的那样。

> 电路交换网络是由物理链路连接的一起交换机组成的，每条链路划分成n个通道。

图8.3表示了一个有4个交换机和4条链路的电路交换网络。每条链路用FDM或TDM划分成n（图中n=3）个通道。为了强调链路划分成通道我们采用明显复用记号，即使复用可隐含在交换机结构中。

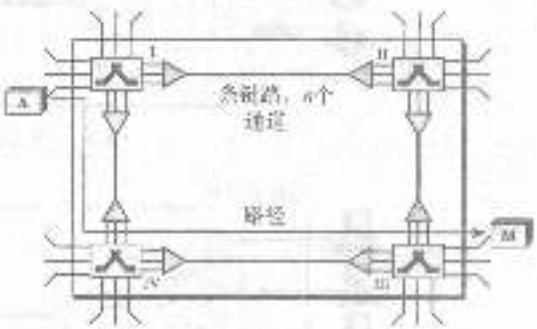


图8.3 一个简单的电路交换网

端系统（如计算机或电话）直接连到交换机。为了简明起见，我们仅表示了两个端系统。当端系统A需要与端系统M通信时，系统A需要向M发送一个连接请求，这个连接请求必须被所有交换机接收包括M本身的交换机。这称为**建立阶段**（setup phase）；在每条链路上预定一个电路（通道），并将这些电路或通道联合起来指定一条专用路径。当建立了这些连接的电路或通道的专用路径后，**数据传输**（data transfer）才可以进行。所有数据传送完以后，拆除这些电路。

这里需要强调几点：

- 电路交换是在物理层；
- 通信开始前，站点必须对通信时间所用的资源给以预留。这些资源如通道（FDM的带宽和TDM的时隙）、交换机的缓冲器、交换机处理时间和交换机输入/输出端口，在整个数据传输期间必须保留专用直到拆除阶段（teardown phase）为止；
- 两个站点之间数据传输不打包（物理层传输信号）。尽管可能存在有间隙，但源站点发送连续的数据流，由目的站接收这些数据流；
- 数据传输期间没有寻址，交换机基于它们占有频带（FDM）或时隙（TDM）发送数据。当然，在建立阶段，存在端到端的寻址，稍后将讨论这点。

> 在电路交换中，建立阶段必须预留资源，作为整个数据传输期间的专用资源直到拆除阶段。

例8.1

作为一个简单的例子，让我们考察在一个小范围内连接8台电话机的电路交换网络，其通信通过4kHz语音通道。假定每条链路用FDM连接最大语音通道是两个，每条链路带宽为8kHz。图8.4表示了电话机1连接到电话机7，电话机2连接到电话机5，电话机3连接到电话机8和电话机4连接到电话机6的情况。当然，当新连接发生时，情况会变化。交换机控制这些连接。

例8.2

作为另一个例子，考虑某个私人公司两个远程办公室计算机的连接。办公室从通信服务提供商租用T-1专用线连接这些计算机。在这个网络中，有两台4×8（4个输入8个输出）交换机。每台交换机中4个输出端与输入端重叠以允许同一办公室的计算机之间通信，另外4个输出端允许两个办公室之间通信。图8.5表示了这情况。

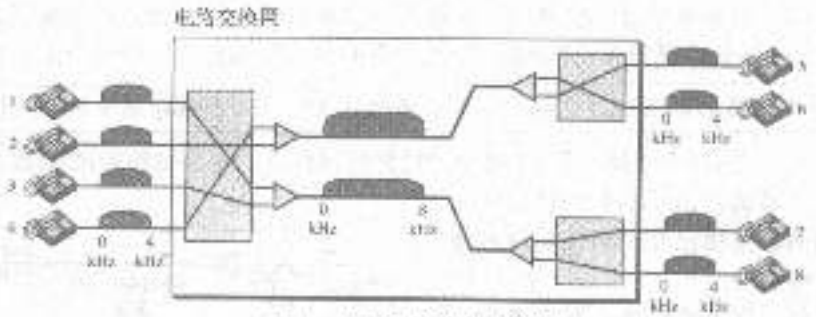


图8.4　例8.1所用的电路交换网

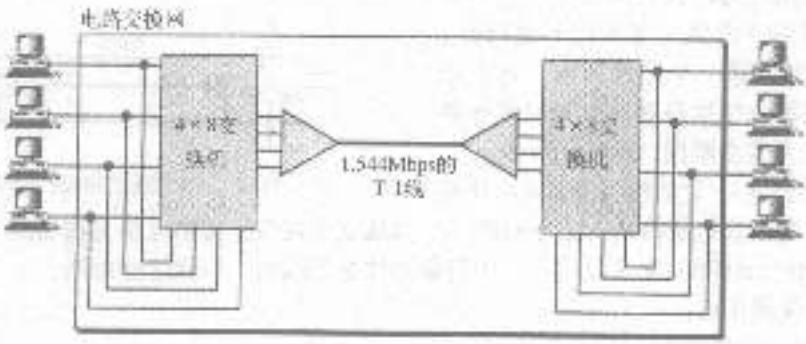


图8.5　例8.2所用的电路交换网

8.1.1　三个阶段

电路交换网实际通信有三个阶段：连接建立、数据传输和连接拆除。

连接建立阶段

在双方（或电话会议中多方）通信之前，需要建立一条专用电路（链路中的通道的组合）。

端系统通常通过专用线路连接到交换机，于是连接建立的意思就是交换机之间建立一些专用通道。例如，在图8.3中，当系统A要求与系统M连接时，它发送一个含有M地址的连接请求给交换机I，为此目的交换机I发现它与交换机IV之间有一条专用的通道。然后交换机I向交换机IV发送一个请求，交换机IV发现它与交换机III之间有一条专用通道。此时，交换机III把系统A的意图通知系统M。

下一步做出一次连接，系统M必须反方向发送一个确认给系统A。仅当系统A接收到这个确认之后，连接才建立。

注意：端到端系统的寻址要求建立两个端系统之间的连接。例如，计算机的地址是由FDM网络管理者给定的，或电话号码是向FDM网络管理者给定的。

数据传输阶段

专用电路（通道）建立后，双方可传输数据。

拆除阶段

当任何一方要撤销连接，就向每台交换机发送一个信号释放资源。

8.1.2　效率

因为资源在整个连接期间都被占用，这些资源不能被其他连接所用。所以电路交换网的效率不及其他两类网络类型的效率。在电话网中，当人们结束对话时，就终止通信，可是在

计算机网络中，即使长时间没有通信，一台计算机也可能与另一台计算机相连。此时，允许拥有专用资源的意思是不能用于其他连接。

8.1.3 延迟

尽管通常认为电路交换网的效率低，但这种网络的延迟最小。在连接期间，资源被分配给该连接，每个交换机没有延迟。图8.6表示了仅有两台交换机的电路交换网延迟的概念。

正如图8.6所示，每台交换机没有等待时间，整个延迟时间是建立连接需要的时间、数据传输时间和拆除电路时间。由建立所引起的延迟是4个部分和：源计算机请求的传播时间（第一个灰色框的斜度），请求信号的传输时间（第一个灰色框的高度），目的计算机确认的传播时间（第二个灰色框的斜度）和确认信号的传输时间（第二个灰色框的高度）。数据传输延迟是2个部分之和：传播时间（浅灰色框的斜率）和数据传输时间（浅灰色框的高度），它可能很长。第二框表示了拆除电路所需的时间，图中表示了接收方请求撤销连接，产生的最大延迟。

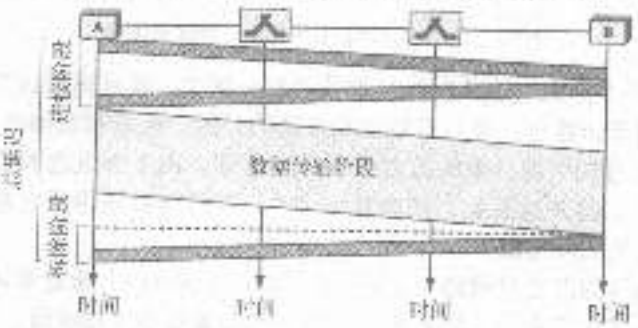


图8.6 电路交换网中延迟

8.1.4 电话网中电路交换技术

将在第9章中看到，电话公司以前选择电路交换方法在物理层中进行交换。目前趋向转用其他交换技术。例如，采用全球地址的电话号码，用7号信令系统（称为SS7）建立连接和撤销连接。

> 传统电话网物理层的交换采用电路交换方法。

8.2 数据报网络

在数据通信中，我们需要从一个端系统发送报文到另一个端系统。如果经过分组交换网传送报文，则报文必须划分为一些固定长的分组或可变长的分组，分组长度由网络和控制协议决定。

在分组交换中，对分组不存在资源分配。这就是说链路没有预留的带宽，以及对每个分组没有安排预定的处理时间。资源按需分配，基于先来先服务的原则，当交换机接收到一个分组时，不管它是源端还是目的端，如果还有其他分组正在处理，该分组必须等待。正如日常生活一样，没有预留可能会发生延迟。例如，没有预定去餐馆就餐，可能会等待。

> 在分组交换网中，不存在资源预留，资源按需分配。

在数据报网络（datagram network）中，每个分组独立处理，与其他分组无关。即使某个分组是多分组传输的一个部分，网络处理它好像它是单独存在一样，这种方法的分组称为数据报（datagram）。

数据报交换通常是在网络层。此处简要对数据报网、电路交换网和虚电路交换网进行比较，本书第4部分将详细讨论它。

图8.7表示了站点A如何传递4个分组到站点X的数据报方法。传统上称数据报网中的交换机为路由器，这就是我们为什么要在图中的交换机使用不同的符号。

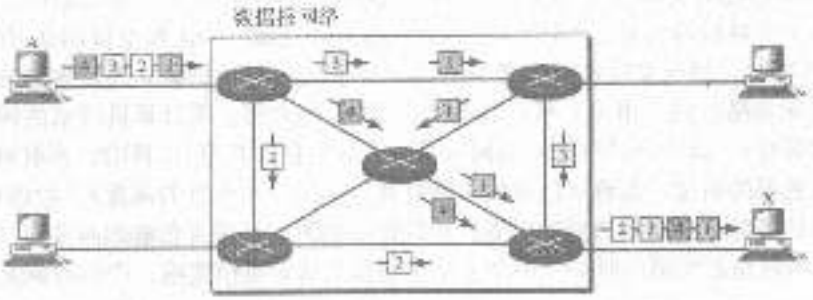


图8.7　有4个交换机（路由器）的数据报网

在本例中，所有4个分组（或数据报）都属于同一报文，但可能经过不同的路径传送到目的端。所以这样是由于链路包含来自其他源端发送的分组，而没有足够的带宽携带从A到X的所有分组。这种方法可能引起传输的数据报到达目的端，由于分组之间延迟不同而失序，也可能由于缺乏资源，引起分组丢失。在多数协议中，在传送到应用层之前，由上层负责数据报重新排序或寻找丢失的数据报。

数据报网有时也称为**无连接网络**（connectionless network）。此处术语无连接的意思是交换机（分组交换机）不保存有关连接状态的信息。不需要建立连接阶段，也不需要拆除阶段。每个分组不管源端和目的端由交换机同样处理。

8.2.1　路由表

如果没有建立和拆除阶段，在数据报网中如何将分组传送到它的目的端？在这种类型网络中，每个交换机（或分组交换机）都有一个基于目的地址的路由表。路由表是动态的，周期性地修改。表中记录目的地址和相应的转发端口，这与电路交换网在建立阶段完成表的项目和在拆除阶段结束时删除不同。图8.8表示了交换机的路由表。

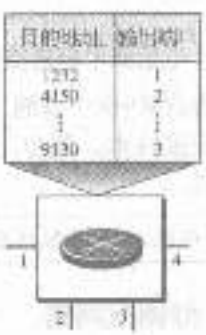


图8.8　数据报网络中路由表

数据报网中的交换机使用基于目的地址的路由表。

目的地址

数据报网络中每个分组有一个头部，它包含分组的目的地址。当交换机接收到分组时，检查目的地址，查阅路由表找到对应的输出端口，通过它将转发出去。不像虚电路交换网中的地址，这个地址在分组传送期间保持不变。

数据报网分组的头部中的目的地址在分组传送期间保持不变。

8.2.2　效率

数据报网的效率比电路交换网高。仅当有传输的分组时，才分配资源。如果源端发送一个分组，在另一个分组可发送前，存在很少时间延迟。对于来自其他源端的分组在这段时间，需重新分配资源。

8.2.3 延迟

数据报网的延迟比虚电路交换网长。虽然数据报网没有建立阶段与拆除阶段，但每个分组在转发前在交换机会有等待。另外，由于报文中所有分组不需要通过同一个交换机传送，因此，报文的分组延迟不一致。图8.9给出了单个分组在数据报网中延迟的例子。

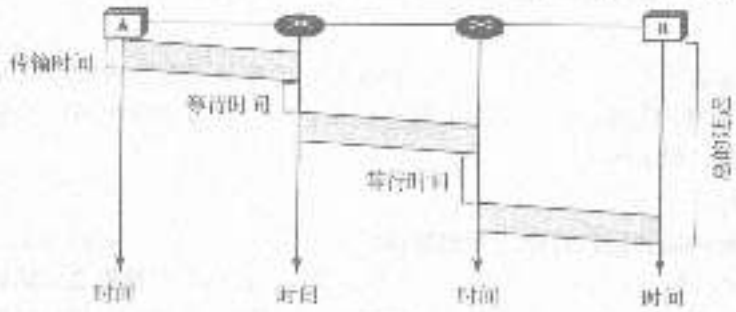


图8.9 数据报网中延迟

分组通过两个交换机传送，存在3个传输时间（3T），3个传播延迟（线的斜度3τ）和2个等待时间（$w_1 + w_2$），不计每个交换机处理时间，则

$$\text{总延迟时间} = 3T + 3\tau + w_1 + w_2$$

8.2.4 因特网中的数据报网

将在后面几章中看到，因特网在网络层选用数据报方法进行交换，采用网络层规定的统一地址寻找从源端到目的端的路径。

> 因特网在网络层用数据报方法对分组进行交换。

8.3 虚电路网络

虚电路网络（virtual-circuit network）是结合电路交换网络与数据报网络的产物，它具有两者的某些特征：

1. 在数据传输阶段，如同电路交换网络一样有建立阶段与拆除阶段；
2. 同电路交换网络或数据报网络一样，能够在建立阶段期间分配资源；
3. 同数据报网络一样，数据被划为分组，每一分组的头部含有地址，它具有本地的权限（它定义下一个交换机和传送分组的通道应该是什么），而不是端到端的权限。读者可能会问，如果分组没有携带最终目的地址，中间交换机如何知道应该将分组发送到何处。在下一节讨论虚电路标识符时，将明确回答这个问题；
4. 同电路交换网络一样，所有分组沿着连接期间建立的路径传送；
5. 虚电路网络通常在数据链路层实现，而电路交换网络是在物理层实现，数据报网络在网络层实现，但在将来这可能会变化。

图8.10是虚电路交换网络的例子。网

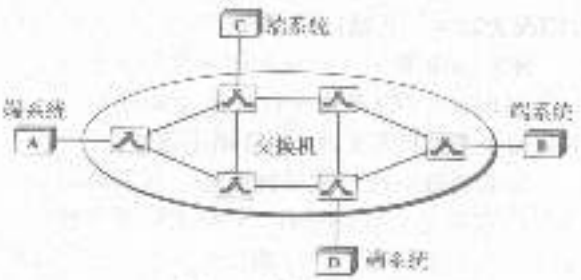


图8.10 虚电路网络

络的交换机执行从源端到目的端运信量交换。源端或目的端可以是计算机，分组交换机，网桥或连接网络的任何设备。

8.3.1 编址

在虚电路网络中，有两类编址：全局的和本地的（虚电路标识符）。

全局编址

源端或目的端需要有一个全局地址，该地址是一个广域网范围内或者国际范围内（如果这个广域网是一个国际网络的一个部分）唯一地址。然而，虚电路网络中的全局地址仅用于建立虚电路标识符，如下所述。

虚电路标识符

实际用于数据传输的标识符称为**虚电路标识符**（virtual circuit identifier，VCI）。与全局地址不同，一个VCI是一个仅在交换机范围内的小数字，让两个交换机之间的链来使用。当一帧到达一个交换机时，它有一个VCI，当它离开时，它有另一个VCI。图8.11说明了数据帧中的VCI从一个交换机到另一个交换机时是如何变化的。注意：既然每个交换机都可以使用自己唯一的VCI集，因此，VCI不必是一个大的数字。

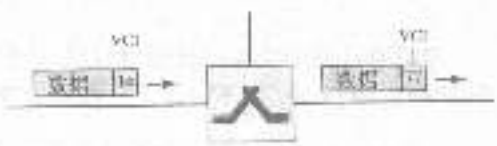


图8.11　虚电路标识符

8.3.2 三个阶段

如电路交换网一样，虚电路网络中的源端和目的端也需要经历三个阶段：建立连接、数据传输和拆除连接。在建立连接阶段，源端和目的端使用各自的全局地址来帮助交换机为连接建立表项。在拆除连接阶段，源端和目的端通知交换机删除相应的表项。数据传输发生在这两个阶段之间。先讨论数据传输阶段，这样更直接，然后再讨论建立连接阶段和拆除连接阶段。

数据传输阶段

为了从源端向目的端传输帧，所有交换机需要为这个虚电路建立表项。这个表，最简单的形式有4列。这意味着交换机需要知道已经建立的虚电路的4个信息。将在以后讨论交换机如何建立表项，现在假设每个交换机已经为每个活动的虚电路建立了表项。图8.12显示了一个这样的交换机和它相应的表。

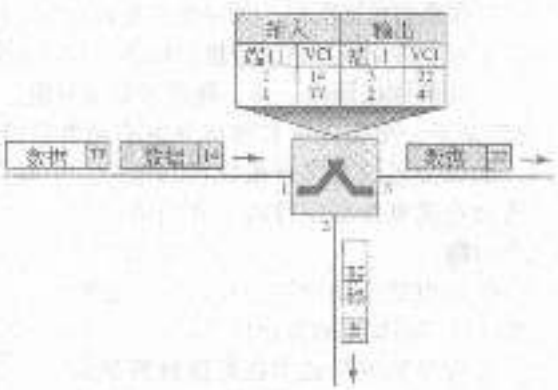


图8.12　虚电路网络中交换机和表

图8.12表示，一个VCI是14的帧到达端口1。当该帧到达时，交换机在表中查找端口1和VCI14。找到后，交换机知道应该将VCI改为22并且从端口3发送输出。

图8.13说明了一个帧如何从源端A到达目的端B以及VCI在这个过程中是如何变化的。每个交换机都更改VCI并选出该帧。

数据传输阶段一直保持活动，直到源端向目的端发送完所有的帧。对报文的所有帧来说，交换机的这个程序都是一样的。这个过程在源端和目的端之间建立虚电路，而不是实际电路。

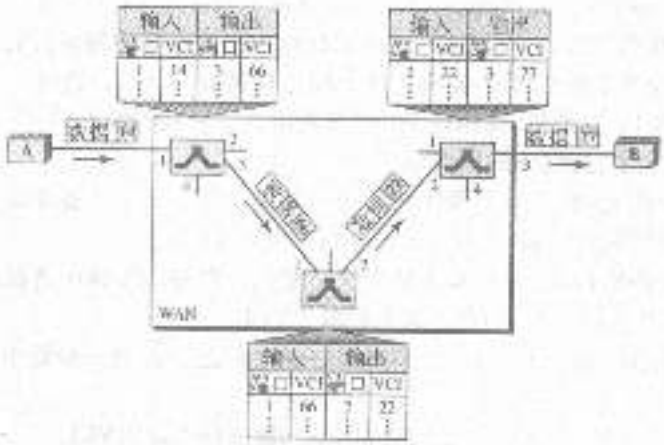


图8.13 源端到目的端的数据传输

建立阶段

在建立阶段，一个交换机为虚电路生成一个入口。例如，假定源A需要生成到B的虚电路，需要两个步骤：建立请求和确认。

连接请求 一个连接请求的帧从源端发送到目的端。图8.14表示了这一过程。

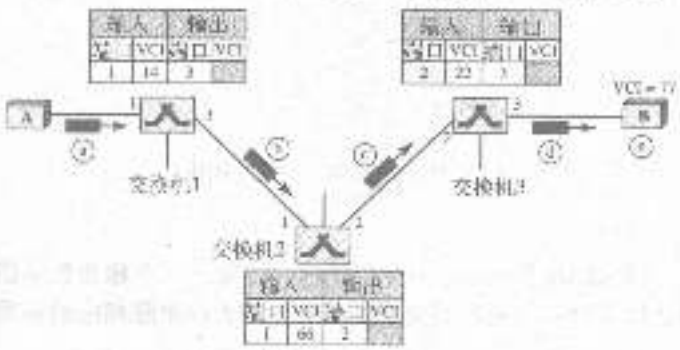


图8.14 虚电路交换网中连接请求

a. 源端A向交换机1发送一个建立连接帧。

b. 交换机1接收这个建立请求帧。它知道从A到B的帧要经过端口3。交换机如何知道这个信息是下一章讨论的一个问题。在建立连接阶段，交换机担当分组交换机的作用。它有一个和交换表不同的路由表。暂时假设它知道输出端口。交换机为这个虚电路在表中建立一个表项，但它只能填四列中的三列：输入端口（1）、输入VCI（14）和输出端口（3）。它仍然不知道输出VCI，这个VCI将在确认步骤中找到。然后，这个交换机通过端口3向交换机2转发该帧。

c. 交换机2接收这个建立请求帧。与在交换机1中发生的情况一样，完成了表的三列：在这个情形中是输入端口（1）、输入VCI（66）和输出端口（2）。

d. 交换机3接收这个建立请求帧。同样完成了三列：输入端口（2），输入VCI（22）和输出端口（3）。

e. 目的端B接收这个建立请求帧，如果它准备好接收从A发来的帧，它就分配一个VCI给从A来的输入帧，这里是77。这个VCI让目的端知道帧是从A发出的，而不是从其他源端发出的。

确认　一个称为确认帧的特殊帧可以完成交换表中的表项。图8.15说明了这个过程。

a. 目的端给交换机3发送一个确认。这个确认帧携带有全局源端地址和目的端地址，这此交换机知道该完成交换表中的那个表项。这个帧还携带有VCI77，这是由目的端选择的作为从A来的帧的输入VCI。交换机3使用这个VCI来完成这个表项的输出VCI列。注意，77是目的端B的输入VCI，但也是交换机3的输出VCI。

b. 交换机3向交换机2发送一个确认帧。这个帧包含有在前一步骤中选择的，自己表中的输入VCI。交换机2使用这个VCI作为自己表中的输出VCI。

c. 交换机2向交换机1发送一个确认帧。这个帧包含在前一步骤中选择的，自己表中的输入VCI。交换机1使用这个VCI作为自己表中的输出VCI。

d. 最后，交换机1向源端A发送一个确认帧。这个帧包含在前一步骤中选择的，自己表中的输入VCI。

e. 源端使用这个VCI作为准备发送给目的端B的数据帧的输出VCI。

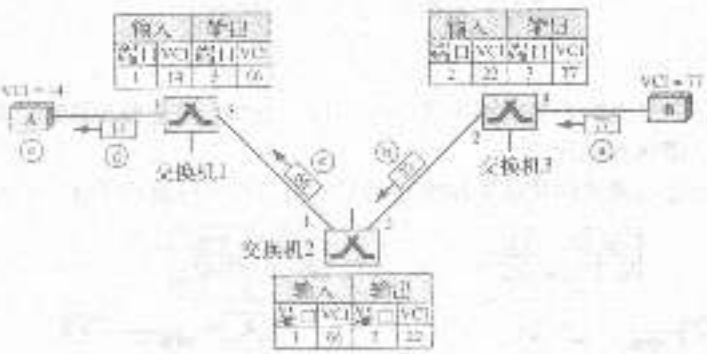


图8.15　虚电路交换网中建立确认

拆除连接阶段

在这个阶段中，当源端A向B发送完所有的帧后，再发送一个称为拆除请求的特殊帧。目的端B用一个拆除确认帧来响应。所有的交换机从各自的表中删除相应的表项。

8.3.3　效率

正如前述，在连接建立阶段虚电路网络要作资源预留，或在数据传输阶段按需分配资源。在第一种情况，每个分组延迟都相同；在第二种情况，各分组延迟可能不同。虚电路网络最大一个优点就是资源可按需分配。源端可检测资源可利用的程度，而不必真实预留。如同一个家庭去餐馆就餐，尽管餐馆不接受预订（席位按需分配）而家庭可打电话查明等待时间。这可节省时间与精力。

> 在虚电路交换中，属于相同源端与目的端的所有分组都按同一路径传送；但如果资源被实分配，分组到达目的端可能有不同延迟。

8.3.4　虚电路网络延迟

在虚电路网络中，建立阶段有一个时间延迟，拆除阶段有一个时间延迟。如果在建立阶段分配资源，那么个别分组没有等待时间。图8.16表示了虚电路网络中一个分组经过两个交换机传送的延迟。

分组经过两个交换机（路由器）传送，有三个传输时间（3τ），二个传播时间（3τ），效

据传输由斜线描述。建立阶段延迟（包括两个方向传输与传播）和拆除阶段延迟（包括一个方向传输与传播），我们不计每个交换机处理时间。总的延迟时间是

总延迟 = 3T + 3τ + 建立阶段延迟 + 拆除阶段延迟

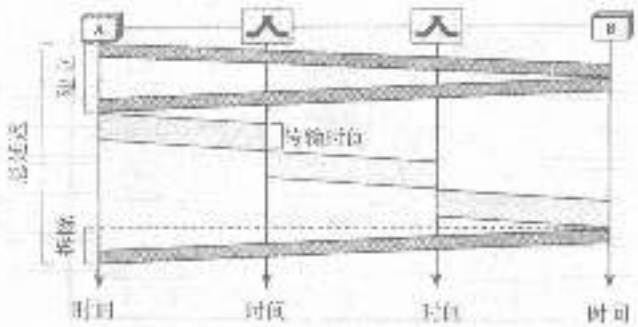


图8.16 虚电路网络延迟

8.3.5 广域网中电路交换技术

将在第18章中，讨论用于广域网中的虚电路网络，如帧中继网和ATM网，这种虚电路技术非常适用于链路层。

> 在交换广域网中，数据链路层通常采用虚电路技术实现。

8.4 交换机结构

在电路交换网和分组交换网中，我们使用交换机。本节讨论每种类型网络所用的交换机的结构。

8.4.1 电路交换机结构

目前电路交换机采用两个技术：空分交换机和时分交换机。

空分交换机

在空分交换（space division switching）中，电路中的各条通路在空间上是相互分离的。这种技术最初是用于模拟网络的，但是目前它被同时使用在模拟和数字网络中。它是由多种设计方案经过很长时间才逐步发展起来的。

纵横制交换机（crossbar switch）在栅格中将n个输入连接到m个输出，连接方法是在每个交叉点（crosspoint）使用微电子开关（晶体管）（见图8.17）。这种设计方法的主要限制是所需要的交叉点的数量。使用纵横制交换机将n个输入连接到m个输出，需要$n \times m$个交叉点。例如，将1 000个输入连接到1 000个输出，需要一个具有1 000 000个交叉点的纵横制交换机。一台具有如此多数量交叉点的纵横制交换机是不切实际的。这种交换机的效率也不高，因为统计表明，在任何时刻，只有至于25%的交叉点处于工作状态，而其他交叉点都处于闲置状态。

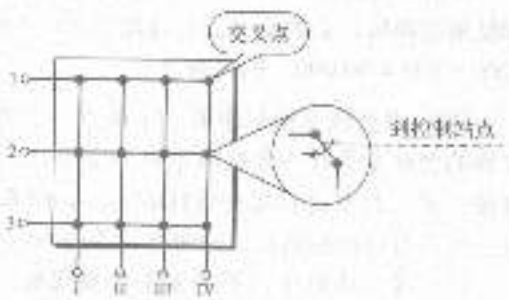


图8.17 3个输入和4个输出的纵横制交换机

多级交换机 解决纵横制交换机限制的一种方案是使用**多级交换机**（multistage switch），

它将多台纵横制交换机按多级（通常是3级）结构组合在一起，如图8.18所示。在单个纵横交换机中，任何连接仅有一行或一列（一条路径）是激活的，所以需要$N \times N$个交叉点。如果在交换机中允许多条路径，则可以降低交叉点数。中间级（第二级）每一个交叉点可被第一级或第三级交叉点访问。

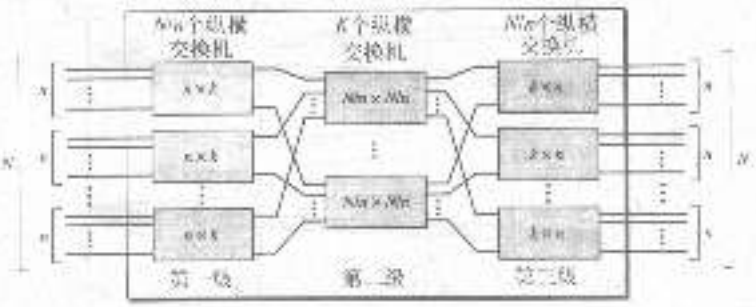


图8.18　多级交换机

设计三级交换机，按照下列步骤：

1. 将N条输线划分为组，每组为n条输入线。对每组用一个$n \times k$纵横交换机，其中k是中间级纵横交换机的个数。换言之，第一级有N/n个纵横交换机，每个具有$n \times k$个交叉点；

2. 在中间级，使用k个$(N/n) \times (N \times n)$纵横交换机；

3. 在第三级，使用N/n个$k \times n$纵横交换机。

计算总的交叉点的个数如下：

$$\frac{N}{n}(n \times k) + k\left(\frac{N}{n} \times \frac{N}{n}\right) + \frac{N}{n}(k \times n) = 2kN + k\left(\frac{N}{n}\right)^2$$

在一个三级交换机中，总的交叉点数是$2kN + k\left(\frac{N}{n}\right)^2$，它比单级交换机的交叉点的个数$N^2$小了许多。

例8.3

设计一个200×200交换机（$N = 200$），其中$k = 4$和$n = 20$。

解：

在第一级有$N/n = 10$个20×4纵横交换机。第二级有4个4×4纵横交换机，第三级有10个4×20纵横交换机。总的交叉点个数是$2kN + (N/n)^2 = 2\,000$。它是单级纵横交换机的交叉点的个数（$200 \times 200 = 40\,000$）的百分之五。

例8.3中多级交换机中有一个缺点，在高通信量时会引起阻塞（blocking）的现象。多级交换的想法是共享中间级纵横机的交叉点。如果资源是有限，并且同一时刻所有用户都请求连接，那么共享会引起有效性的短缺。阻塞是指输入无法连接到输出的情形。因为此时在两者之间没有可用的路径，即所有可用的中间交换机都被占用了。

在单级交换机中，不会发生拥塞现象。因为输入和输出的所有组合都有自己的交叉点，所以总是有路径存在（两个输入端试图连接到同一输出端的情况不计算在内。因为这时还路没有阻塞，只是输出端很忙）。然而，在上面例8.3中描述的多级交换机中，第一组20个输入中只有4个能够同时使用交换机；同样，第二组40个输入中只有4个能够同时使用交换机，依此类推。中间交换机个数少可能形成阻塞现象。

在大规模系统中，如那些有10 000个输入和输出的系统，级数的增加能够迅速降低所需交叉点的数量。但是，随着级数的增长，产生阻塞的可能性也随之增大。许多人经历过这样的情况：在一次自然灾害过后，公用电话系统经常阻塞，因为此时要求确认亲属情况的电话数量远远超过了系统的常规负荷。

Clos研究了多级交换机没有阻塞的条件，提出了下面的公式。在一个没有阻塞的交换机中，中间级交换机的个数必须大于等于$2n-1$，即$k \geq 2n-1$。

注意：交叉点个数仍少于单级交换机交换的个数。现在对固定N，需要用Clos准则确定交叉点的最小个数。对于n（仅有的可变量）可推导出公式，并发现其结果n是0。这个n必须大于或等于$(N/2)^{1/2}$。此时，总交叉点的个数大于或等于$4N[(N/2)^{1/2}-1]$。

换言之，按照Clos准则，最小交叉点个数与$(N/2)^{1/2}$成正比。

按照Clos准则：
$N = (N/2)^{1/2}$
$k > 2n-1$
总的交叉点个数$\geq 4N[(2N)^{1/2} - 1]$

例8.4

用Clos准则和最小交叉点个数重新设计前面三级200 × 200交换机。

解：

设$n = (200/2)^{1/2}$，即$n=10$。计算$k=2n-1=19$。第一级有200/10=20个10 × 19交叉点的纵横交换机。第二级有19个10 × 10交叉点的纵横交换机。第三级有20个19 × 10交叉点的纵横交换机。总的交叉点个数是$20(10 \times 19) + 19(10 \times 10) + 20(19 \times 10) = 9\,500$。如果使用单级交换机需要$200 \times 200 = 40\,000$个交叉点，这个三级交换机交叉点的个数是单级交换机交叉点个数的24%，多于例8.3（5%）中交叉点的个数，这些额外交叉点是防止阻塞所需的。

使用Clos准则和最小交叉点个数的多级交换仍要求很大的交叉点个数。例如，有100 000条输入/输出线的交换机，需要接近2亿个交叉点（而不是千万）。这就是说，如果在城市中电话公司需要一台交换机来提供连接10万台电话机，则它需要2亿个交叉点。如果允许阻塞，则这个数字可降低。目前电话公司使用时分交换或空分和时分相结合的交换机，接下来将马上讨论。

时分交换机

时分交换（time-division switching）采用时分复用（TDM）在交换机内部来实现交换。最常用方法是时隙互换（time-slot interchange）（TSI）。

时隙互换　图8.19说明了一个将4条输入线路连接到4条输出线路的系统。设想每一条输入线路按下列模式将数据发送到一条输出线路：

$$1 \rightarrow 3,\ 2 \rightarrow 4,\ 3 \rightarrow 1,\ 4 \rightarrow 2。$$

该图将一个TDM复用器、一个TDM分离器和一个TSI组合在一起，其中TSI是由具有多个存储单元的随机访问器（RAM）构成。每个存储单元的大小与单个时隙的大小相同。存储单元的个数与输入端的个数相同（多数情况，输入端与输出端个数相同）。RAM中存储的是顺序接收的时隙数据。然后，时隙按照控制单元的决策发送出去。

时分交换与空分交换机组合

在比较空分交换和时分交换时，会出现一些有趣的情况。空分交换的优势在于它是瞬态的。它的缺点在于，空分交换的有效性，即拥塞可以忍受是由交叉点数量决定的。

时分交换的优点在于它不需要交叉点，它的缺点是在使用TSI时，对每个连接的处理会产生延迟。每个时隙必须存储在RAM中，然后被检索和传递。

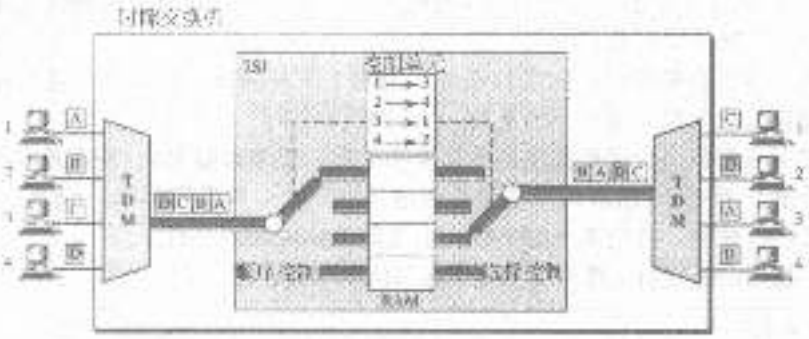


图8.19　时隙互换

在第二种可选方法中，将空分和时分技术组合起来，以充分利用两者的优点。将两种方法组合产生的交换，可以在物理上（交叉点的个数）和时间上（延迟时间）进行优化。这一类多级交换机可以设计为时间－空间－时间（time-space-time，TST）交换机。

图8.20说明了一种简单的TST交换机，包括两级时分交换和一级空分交换，具有12路输入和12路输出。它不是使用一台时分交换机，而是将所有输入分成3组（每组4个输入）并连接到3台时隙交换机中。这做的结果是，平均延迟只是使用单一时隙交换机处理12个输入所产生的延迟时间的1/3。

最后一级是第一级的镜像。中间级是一台空分交换机（纵横制），它连接到TSI交换机组，使所有可能的输入输出对之间建立连接（比如，将第一组中的输入3连接到第二组中的输出7）。

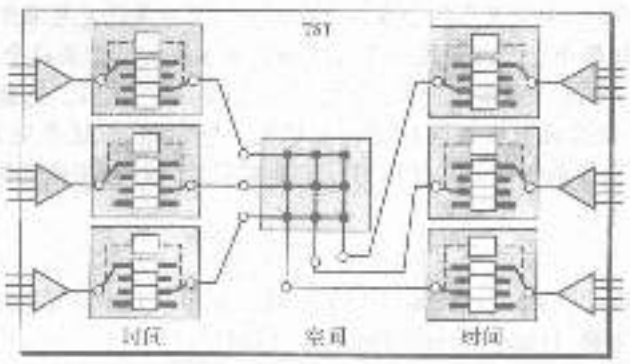


图8.20　时间-空间-时间交换机

8.4.2　分组交换机结构

分组交换网所用的交换机与电路交换网所用的交换机不同。分组交换机有三个部分：**输入端口**（input port）、**输出端口**（output port）、**路由处理器**（rounting processor）和**交换结构**（switching fabric），如图8.21所示。

输入端口

输入端口执行分组交换机的物理层和数据链路层的功能，将接收信号转换成位，将帧分解成分组，进行差错检测与纠错。准备由网络层发送分组，除物理层处理器和数据链路层处理器之外，在分组转向交换结构前，输入端口有缓冲器（队列）保存分组。图8.22表示了输

入端口的框图。

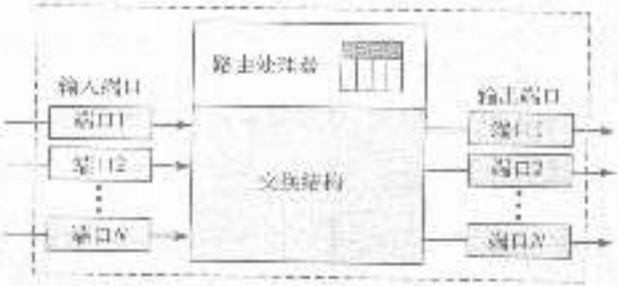


图8.21 分组交换机组成部分

输出端口

输出端口执行的功能与输入端口相同，但顺序相反。首先对输出分组进行排队，然后将分组组装成帧，最后，利用物理层功能将帧转换成信号在线路上发送。图8.23表示了输出端口的框图。



图8.22 输入端口



图8.23 输出端口

路由处理器

路由处理器执行网络层的功能。利用目的地址寻找下一跳的地址和输出端口号，通过它发送分组。因为路由处理器寻找路由表，有时这个动作称为**路由表查寻**（table lookup）。在最新的分组交换机中，为了加速与方便该过程，将路由处理器的功能移到输入端口。

交换结构

在交换机中最艰难的任务是从输入队列将分组传送到输出队列。该动作的速度影响输入/输出队列的长度和分组总传输延迟。在过去，分组交换机实际上是专用的计算机，它的存储器或总线用作交换结构。输入端口将分组保存在存储器中，输出端口从存储器中取回分组。目前，分组交换机是专门的机制，有各种各样的交换结构，此处简要讨论其中一些结构。

纵横交换机 交换机制最简单的类型是纵横交换机的结构，在前节已讨论过。

Banyan交换机 一个比纵横交换机更实际的方法是banyan交换机（banyan switch）（以格列命名）。banyan交换机是一个多级交换，每一级有微交换开关表示为二进制中的输出端口来发送分组。对于n个输入和n个输出，我们将有$\log_2 n$级，每级有$n/2$个微交换。第一级基于二进制中的最高位发送分组，第二级基于二进制中的次最高位发送分组，依此类推。图8.24表示了有8个输入和8个输出的banyan交换机，级数是$\log_2 8 = 3$。

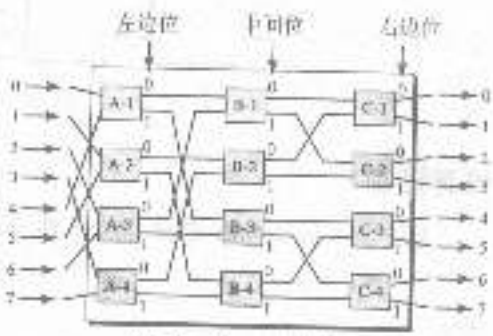
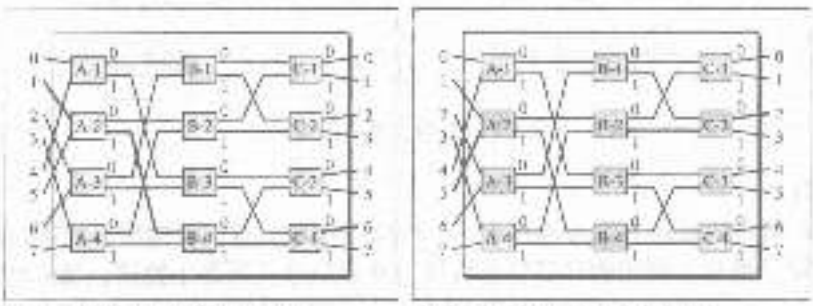


图8.24 Banyan交换机

图8.25表示了操作。在a部分中，一个分组到达输入端口1且应该去输出端口6（二进制110）。第一级微交换（A-2）基于第一位（1）发送分组，第二级微交换（B-4）基于第二位（1）发送分组，第三级微交换（C-4）基于第三位（0）发送分组。在b部分中，一个分组到达输入端口5且应该去输出端口2（二进制010）。第一级微交换（A-2）基于第一位（0）发送分组，第二级微交换（B-2）基于第二位（1）发送分组，第三级微交换（C-2）基于第三位（0）发送分组。



a) 输入1发送信元给输出6（110）　　b) 输入5发送信元给输出2（010）

图8.25　Banyan交换机路由示例

Batcher-banyan交换机　banyan交换机中存在问题是即使两个分组不是前往相同的输出端口，也可能存在内部冲突。这个问题可通过对目的端口对到达的分组进行排序来解决。

K.E.Batcher设计了一种交换机，放在banyan交换机之前，并根据输入分组的最终目的对分组进行排序。这种组合称为Batcher banyan交换机（Batcher-banyan switch）。排序交换机采用硬件合并技术，但此处不详细讨论。通常，将另一个称为陷阱（trap）的硬件模块加在Batcher交换机与banyan交换机之间（见图8.26）。陷阱模块防止重复的分组（具有相同输出目的地的分组）同时通过banyan交换机。在每个时间节拍，对每个目的地仅有一个分组，如果存在多于一个分组，它们就应该等待下一个节拍。

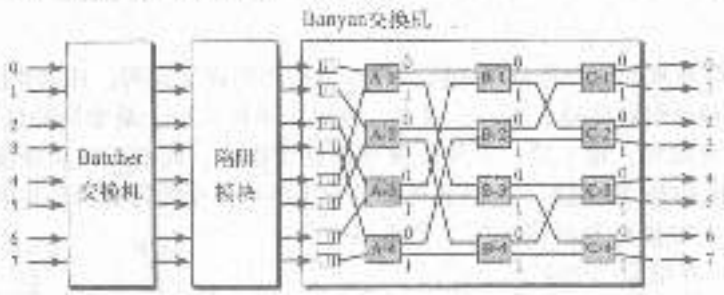


图8.26　Batcher-banyan交换机

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物，在本书末列出括号[……]中的参考资料。

读物

[STAN04]的第10章和[GW04]的第4章与第7章讨论交换。[BELL00]全部讨论电路交换。

本章小结

- 交换网络由一系列互连称为交换机的节点构成。传统上有三种重要的交换方法：电路交换、分组交换和报文交换。
- 目前的网络分为三大类：电路交换网、分组交换网和报文交换网。分组交换网可进一步分为两个子类：虚电路网和数据报网。
- 电路交换网是由物理链路连接一组交换机组成，每条链路划分成n个通道。电路交换在物理层进行。在电路交换中，建立阶段期间必须预留资源，整个数据传输期间资源保留专用直到拆除阶段。
- 在分组交换中，对分组不存在资源分配，这就是说链路没有预留的带宽，对每个分组没有按预定处理时间，资源按需分配。
- 在数据报网络中，每个分组独立处理与其他分组无关。这个方法的分组称为数据报。没有建立阶段和拆除阶段。
- 虚电路网络是结合电路交换网络与数据报网的产物。它具有两者一些特性。
- 电路交换使用两种技术：空分交换或时分交换。
- 分组交换网中的交换机与电路交换网中的交换机有不同的结构。分组交换机有四个组成部分：输入端口、输出端口、路由处理器和交换结构。

习题

复习题

1. 描述交换所需要的事物和交换机基本特征。
2. 列出传统的三种交换方法，目前最常用方法是什么？
3. 分组交换的两种方法是什么？
4. 比较电路交换网和分组交换网，并说明它们之间的差异。
5. 通过数据报网传送时，分组的地址字段起什么作用？
6. 通过虚电路网传送时，分组的地址字段起什么作用？
7. 比较时分交换机和空分交换机。
8. TSI是什么？它在时分交换中起什么作用？
9. 解释交换网中的阻塞现象。
10. 列出分组交换机的两个主要组成部分，并说明它们的功能。

练习题

11. 数字电路交换网有一条路径，数据速率为1Mbps。在建立与拆除阶段，要交换1 000位数据，双方距离为5 000km。如果传播速度为2×10^8m，试回答下列问题：
    a. 如果在数据传输阶段，交换1 000位数据，总的延迟是多少？
    b. 如果在数据传输阶段，交换100 000位数据，总的延迟是多少？
    c. 如果在数据传输阶段，交换1 000 000位数据，总的延迟是多少？
    d. 对上面每种情形，求每1 000位数据延迟是多少并比较它们，能推断出什么？
12. 属于同一报文中的5个同样长度的数据报从一个目的地到下一个目的地，但它们通过不同路径，如表8.1所示。

表8.1　练习12

数据报	路径长度	访问的交换机
1	2 200km	1，2，5
2	11 700km	1，3，5
3	12 200km	1，2，3，5
4	10 200km	1，4，5
5	10 700km	1，4，3，5

假设每台交换机（包括等待和处理）的延迟分别是3，10，20，7和20ms。传播速度为2×10^8m。求数据报到达目的端的顺序与每一个的延迟。不计传输中任何其他延迟。

13. 网络中信息传输牵涉到端到端的寻址，有时也牵涉本地寻址（如CVI）。表8.2表示了的每种类型网络和所用的寻址机制。

表8.2　练习13

网　络	建立阶段	数据传输阶段	拆除阶段
电路交换网	端到端		端到端
数据报网		端到端	
虚电路交换网	端到端	本地	端到端

试回答下列问题：

a. 为什么在建立阶段和拆除阶段，电路交换网需要端到端寻址？在数据传输阶段不需要地址？

b. 为什么数据报网仅需在数据传输阶段需要端到端寻址，而建立阶段和拆除阶段不需要寻址？

c. 为什么在虚电路交换网中所有三个阶段都需要地址？

14. 我们提到两种类型的网络：数据报网与虚电路网需要一张路由表或交换表，从该表中属于目的端信息中寻找输出端口，但在电路交换网不需要这样的表。说出这种不同的理由。

15. 虚电路网的交换表的表项通常是在建立阶段和删除拆除阶段时生成的。换言之，这种类型网络的这些表项反映了当前网络动态连接状况。与此相反，数据报网的路由表中的这些表项与当前连接状况无关，它们表示网络的构架和分组应该如何发送到最终目的端。即使网络没有激活，这些表项仍保持不变。如果网络改变，要修改路由表。解释这两种不同特性的理由？由于上述特性说明虚电路是面向连接网络而数据报网是无连接网络？

16. 在数据报网络中路由表列的最小个数是2，而在虚电路网中路由表列的最小个数是4，解释其理由？每种网络的分组中的有关地址类型是否不同？

17. 图8.27表示了数据报网中一个交换机（路由器），按照下列目的地址求分组输出端口：

分组1：7176

分组2：1233

分组3：8766

分组4：9144

18. 图8.28表示了虚电路网络中的一个交换机，按照下列输入端口和VCI地址求输出端口和输出VCI地址：

分组1：3，78

分组2：2，92

分组3：4，56

分组4：2，71

目的地址	输出端口
1235	3
1456	2
3255	1
4470	4
7156	2
8356	1
9141	2

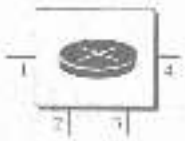


图8.27 练习17

输入		输出	
端口	VCI	端口	VCI
1	14	3	22
2	71	4	41
2	92	1	35
3	58	2	43
3	78	2	59
4	56	3	11



图8.28 练习18

19. 试回答下列问题：
    a. 数据报网络中的路由表是否可有相同目的地址两个表项？并说明之。
    b. 虚电路网中的交换表是否可有相同输入端口的两个表项？是否可有相同输出端口的两个表项？是否可有相同输入VCI的两个表项？是否可有相同输出VCI的两个表项？是否可有相同输入值（端口，VCI）的两个表项？是否可有相同输出值（端口，VCI）的两个表项？
20. 显然，路由器或交换机需要做的搜索是在对应表中寻找信息。对于数据报网，搜索路由表是基于目的地址；对于虚电路网络，搜索交换表是基于输入端口和输入VCI。试解释其理由。并定义如何基于对表进行排序（分类）。
21. 考察有n个输入k个输出的纵横交换机。
    a. 如果$n>k$，能否可说交换机如同复用器？
    b. 如果$n<k$，能否可说交换机如同分离器？
22. 我们需要N = 100的三级空分交换，第一级和第三级使用10个纵横交换机，中间级使用4个纵横交换机。
    a. 画出结构图；
    b. 计算总的交叉点个数；
    c. 求同时可能连接的个数；
    d. 如果用单级纵横交换机（100 × 100），求同时可能连接的个数；
    e. 求阻塞因子，即在c与在d中连接的个数之比。
23. 如果中间级用6个纵横交换机，重做练习22。
24. 利用Clos准则，重新设计练习22的结构。
25. 我们需要1 000个输入和输出的空分交换机，在下列各情况求交叉点个数是多少？
    a. 采用单个纵横交换机；　　b. 基于Clos准则，采用多级交换。
26. 我们需要N=100的三级时—空—时交换，在第一级和第三级采用10个TSI，中间级用4个纵横交换机。
    a. 画出结构图；　　b. 计算总的交叉点个数；
    c. 计算所需TSI的总存储器个数。

第9章 使用电话网和有线电视网进行数据传输

电话网原来是为语音通信而设计的，由于数据通信的需要导致拨号调制解调器的发明。因特网的出现带来数据高速的下载和上传的需要，但调制解调器速度还是太慢。电话公司增加一种数字用户线路（DSL）新技术。尽管拨号调制解调器现还仍旧存在全世界许多地方，但DSL已通过电话网提供快速接入因特网。本章首先讨论电话网的基本结构，然后考察如何使用拨号调制解调器和DSL技术将这些网络接入因特网。

有线网络本来是为了那些由于天然障碍（如山区）的用户不能接收到电视节目而设计的。后来，有线网络成为深受想要接收到较好信号的人们的欢迎。另外，有线网络可通过微波连接接入远程广播站。有线电视网也可用来为视频设计的某些通道提供接入因特网的良好市场。在讨论有线网络基本结构后，讨论电缆调制解调器如何提供高速接入因特网的连接。

9.1 电话网络

电话网络使用电路交换技术。电话网络起始于19世纪末。当时的整个网络，也就是简易普通传统电话系统（plain old telephone system，POTS），最初是一个使用模拟信号传输语音的模拟系统。随着计算机时代的到来，在20世纪80年代，电话网络除了传输语音以外，还传输数据。在接下来的十年里，电话网络经历了多次技术改进。目前电话网络是一种数字和模拟共存的网络。

9.1.1 主要组成

如图9.1所示，电话网络主要由三部分构成：本地回路，干线和交换局。电话网络具有多级交换局，如**本地局**（end office）、**中继局**（tandem office）和**地区局**（regional office）。

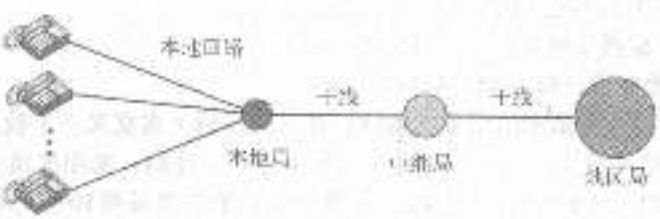


图9.1 电话系统

本地回路

本地回路（local loop）是电话网络的组成部分之一，也就是将用户电话机连接到最近本地局或者本地中心局的双绞线。用于语音传送时，本地回路的带宽是4 000Hz（4kHz）。研究一下与每一个本地回路相关联的电话号码是非常有意思的。一个本地电话号码的前二位确定其所属局，其他4位确定了本地回路编号。

干线

干线（trunk）是处理局间通信的传输介质。一条干线通过多路复用通常可以处理成百上千个连接。传输通常是通过光纤和卫星链路实现的。

交换局

为了避免在任何两个用户之间存在一条永久的物理链路，电话公司在**交换局**（switching office）内安装有交换机。一台交换机可以连接多条本地回路或者干线，并且允许在不同用户间建立连接。

9.1.2 LATA

AT&T在1984年“解体”之后（见附录E），整个美国划分为200多个**本地接入和传送区域**（local access and transport area，LATA）。从那以后，LATA的数量也有所增加。一个LATA可以是一个小城市地区或者大城市地区。小的州可能有一个单独的LATA，而大的州可能有多个LATA。LATA的边界可能与州的边界交错，即LATA的一部分可能位于一个州，另一部分也可能位于另一个州。

LATA内部服务

LATA内部的**公用电信公司**（common carrier，或称电话公司）提供的服务称为LATA内部服务。处理这些服务的运营商称为**本地交换电信公司**（local exchange carrier，LEC）。在1996年的电信法实施之前，LATA内部服务是授权给一个单独的运营商经营的，这是一种垄断经营。1996年以后，多个运营商可以提供LATA内部的服务。1996年以前提供服务的运营商占有电缆铺设系统（本地回路），称为**责任本地交换电信公司**（incumbent local exchange carrier，ILEC）。提供服务的新运营商称为**竞争本地交换电信公司**（competitive local exchange carrier，CLEC）。为了避免增加新的电缆铺设费用，一般同意由ILEC继续提供主要服务，而CLEC在LATA内部可以提供其他服务，如移动电话服务、收费长途电话等。图9.2说明了LATA及其交换局。

> LATA内部服务由本地交换电信公司提供。1996年以后，有两种类型的LEC：责任本地交换电信公司和竞争本地交换电信公司。

LATA内部的通信由本地局和中继局处理。只使用一个交换局就能完成的呼叫是免费的。必须通过一个中继局（LATA内部的长途电话局）的呼叫是收费的。

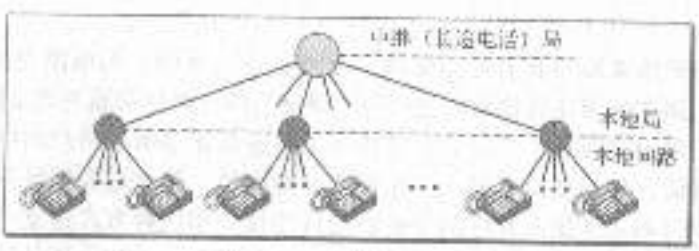


图9.2 LATA内部的交换局

LATA间的服务

LATA之间的服务由**跨区交换电信公司**（interexchange carrier，IXC）处理。这些公司有时称为**长途电话公司**（long-distance company），为不同LATA的两个客户之间提供通信服务。1996年电信法实施以后（见附录E），这些服务可以由任何电信公司提供，包括那些在LATA内部提供服务的电信公司。这个领域是完全开放的。提供LATA间服务的电信公司包括AT&T、MCI、WorldCom、Sprint和Verizon。

IXC是长途电话公司，它们提供包括电话服务在内的通用数据通信服务。通过IXC的电话

呼叫通常被数字化处理，电信公司使用多种类型的网络提供这一服务。

服务接入点

前面讨论过，LATA内部服务可以由多个LEC（一个ILEC和多个CLEC）提供。还提到过，LATA之间的服务可以由多个IXC提供。这些电信公司之间如何交互呢？答案是由称为**服务接入点**（Points of Presence，POP）的交换局来完成交互。所有希望在LATA中提供LATA间服务的IXC，都必须在该LATA内有一个POP。在LATA内部提供服务的LEC，必须提供连接以使每个用户都能访问到所有的POP。图9.3说明了这一概念。

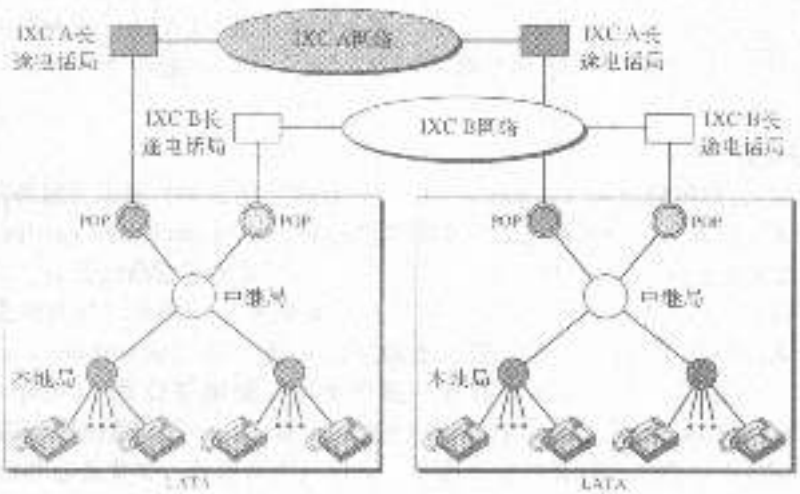


图9.3　服务接入点（POP）

当一个用户需要与另一个用户建立连接时，首先被连接到本地交换机，然后，直接连接到一个POP或者通过一台中继交换机连接到POP。这时，呼叫从源LATA中的IXC（用户选择的）的POP到达目的LATA中的同一IXC的POP。呼叫将会通过IXC的长途电话局，并且经过IXC提供的网络进行传送。

9.1.3　信令

在早期，电话网络采取用具有专用链路（复用还没有发明）的电路交换方式传输语音。正如第8章所见，电路交换网在通信双方之间为了确定与终止路径必须有建立阶段与拆除阶段。在初期这项交换工作由接线员人工完成，接线员工作室是连接所有用户的中心。一个用户想要与另一个用户通话，拿起话筒（摘机）振铃通知接线员。接线员听到呼叫方并知道被呼叫方后，由接线员利用两条线插入对应两个插座进行连接，用这种方式建立一条专用电路。双方中某一方在通话结束后，通知接线员拆除电路。由于用于信令和通话是同一电路，因此这种类型的信令称为**带内信令**（in-band signaling）。

后来，信令系统成为自动系统。发明了旋转拨号电话，发送数字信号表示多位数字电话号上每个数字，电话公司的交换机用该数字信号建立呼叫方与被呼叫方之间的连接，同时使用带内信令和**带外信令**（out-of-signaling）。在带内信令中，4-kHz语音通道也用于提供信令。在带外信令中，部分语音通道用作信令。语音通道的带宽与信令通道的带宽是分离的。

当电话网络发展成一个复杂的网络时，信令系统的功能也有所增加。要求信令系统执行其他一些任务，比如：

1. 提供拨号音、振铃和忙音；

2. 办公室电话号码转移；
3. 维护和监控呼叫；
4. 保留账单信息；
5. 维护和监控电话网络设备；
6. 提供其他功能，如呼叫方ID和语音信箱等。

这些复杂的任务造成了为信令提供一个单独的网络。这就是说，目前电话网络可认为是两个网络：信令网络和数据传输网络。

> 数据传输和信令任务在现代电话网中是分开的：一个网络用来执行数据传输，另一个用来执行信令任务。

可是，在此处我们需要强调一点，虽然两个网络是分开的，但不是每处都存在独立的物理链路，两个网络可用系统部分中同一链路的不同通道。

数据传输网络

目前可携带多媒体信息的数据传输网络最重要的部分是电路交换网，尽管它也可能是分组交换网。这种网络如同本书讨论其他网络一样遵照同一模型和协议。

信令网络

本节主要关心的信令网络是包含层次的分组交换网，层次与第2章讨论的OSI模型和因特网模型的层次相似。信令的特性更适合具有不同层次的分组交换网。例如，需要传送电话地址的信息可封装在一个具有所有差错控制和寻址信息的分组中。图9.4表示了电话网一个简化情形，其中两个网络是分开的。

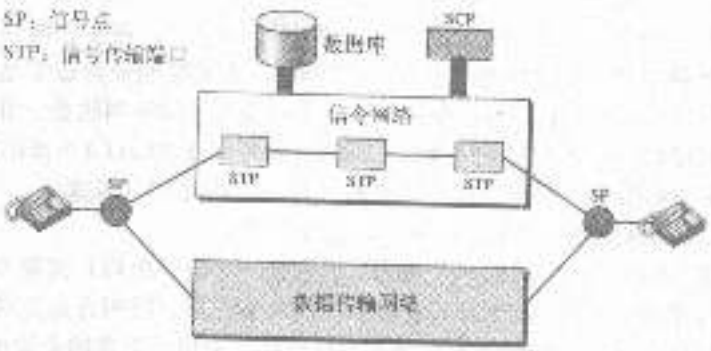


图9.4 数据传输网络和信令网络

用户电话或计算机被连接到**信号点**（signal point，SP）。电话机与信号点之间链路是两个网络公用的。信令网络利用称为**信号传输端口**（signal transfer port，STP）接收或转发信令报文。信令网络也包括**服务控制点**（service control point，SCP）控制所有的网络操作。其他系统如数据库中心可包括有关整个信令网络提供的存储信息。

7号信令系统（SS7）

信令系统所用的协议称为**7号信令系统**（Signaling System Seven，SS7），它与第2章中5层因特网模型很相似，但每层有不同名字，如图9.5所示。

物理层：MTP1 称为**报文传输部分层次1**（message transfer part level1），使用物理层规范，例如像T-1线（1.544Mbps）和DC0（64kbps）。

数据链路层：MTP2 报文传输部分层次2提供典型的数据链路层服务，如用分组头部的

源和目的地址以及差错检测的CRC打包。

网络层：MTP3　报文传输部分层次3提供端到端连接，使用数据报方法进行交换。

传输层：SCCP　信令连接控制点（signaling connection control point，SSCP）用于特殊业务，如800呼叫处理。

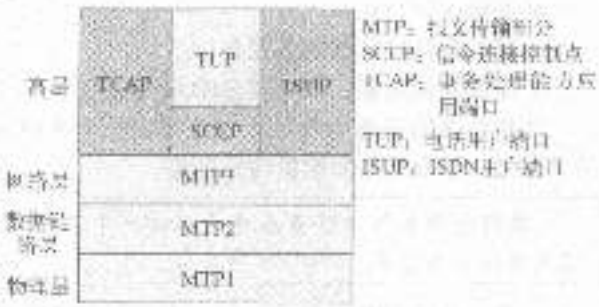


图9.5　SS7的层次

高层：TUP，TCAP和ISUP　在高层有3个协议。电话用户端口协议（telephone user port，TUP）负责建立语音呼叫，它接收拨号数字和被指定路线传送呼叫。事务处理能力应用端口协议（Transaction capabilities application port，TCAP）提供远程呼叫，让一台计算机上的应用程序调用另一台计算机的过程。ISDN用户端口协议（ISDN user port，ISUP）代替TUP提供类似ISDN网络的服务。

9.1.4　电话网络提供的业务

电话公司提供两类业务：模拟业务和数字业务。

模拟业务

从很早开始，电话公司就为它们的用户提供模拟业务。这些业务一直延续到今天。这些业务可以分为两类：模拟交换业务（analog switched service）和模拟租用业务（analog leased service）。

模拟交换业务　模拟交换业务是一种常见的拨号呼叫业务，在使用电话时经常遇到。本地回路中的信号是模拟信号，带宽通常为0～4 000Hz。本地呼叫业务通常是按包月制收费形式提供的，但在有一些LATA中，运营商会按每次呼叫或者一组呼叫收费。非包费制的合理性在于为那些拨打电话较少的客户提供更廉价的服务。服务可能是LATA内部的或LATA之间的。如果LATA的地理分布很广，那么一次呼叫可能经过一个中继局（长途中心），用户就需要为这次呼叫付费。LATA间的呼叫是长途呼叫，也需要付费。

另一种业务称为800业务。如果一个用户（通常是一个组织机构）需要为其他用户（通常是客户）提供免费连接，它可以申请800业务。在这种情况下，呼叫方是免费的，但必须由被呼叫方付费。组织机构使用这种服务来鼓励客户打电话。费用比正常的长途电话要便宜一些。

广域电话业务（wide-area telephone service，WATS）是一种与800相反的业务。后者是一种由组织机构付费的入站呼叫，而前者是由组织机构付费的出站呼叫。这种服务价格较低，可以替代常规的长途电话，收费是依据呼叫次数确定的。这种业务可以指定为出站呼叫到同一个州、到多个州或者到整个国家并收取相应的费用。

900业务（900 service）与800业务类似，都是用户入站呼叫。但是，与800业务不同的是，呼叫是由呼叫方付费的而且通常比正常的长途电话要贵。原因是运营商收取两种费用：第一种是长途电话费，第二种是付给被呼叫方的费用。

模拟租用业务（analog leased service）为客户提供租用一条线路的机会，这种线路有时称为专线，是永久性地连接到另一个客户的。尽管连接仍然通过了电话网络中的交换机，但用户感觉它是条单一的线路，因为交换机总是闭合的，不需要拨号。

数字业务

近年来，电话公司开始为它们的用户提供数字业务（digital service）。数字业务比模拟业

务对噪声和其他形式的干扰的敏感度要低得多。两种最常见的数字业务是交换式/56业务和数字数据业务（digital data service，DDS）。第6章已经讨论了高速数字服务，即T线路。本章讨论其他服务。

交换式/56业务 交换式/56业务（switched/56 service）是模拟交换线路的数字版本。它是一种交换式数字业务，允许数据速率最高到56kbps。为了通过这一业务进行通信，通信双方必须先预定这种业务。使用正常电话业务的呼叫方不能使用交换式/56业务连接到一部电话或者计算机，即使使用了一台调制解调器也不行。总之，对电话公司而言，模拟业务和数字业务表示两种完全不同的领域。由于交换式/56业务中使用的线路已经是数字化的，所以用户不需要用调制解调器来传输数字数据。然而，用户需要另一种称为**数字业务单元**（digital service unit，DSU）的设备。

数字数据业务 数字数据业务（digital data service，DDS）是模拟租用线路的数字版本。它是一条最高数据速率为64kbps的数字租用线路。

9.2 电话调制解调器

传统的电话线路可以传送频率范围是300～3 300Hz的信号，提供给信号的带宽是3 000Hz。整个频率范围用于传输语音，在不丢失可识别性的情况下，有一些干扰和失真是可以接受的。但是，已经看到，数据信号需要更高等级的准确度以确保完整性。所以出于安全性原因，这一范围的边缘不能用于数据通信。通常情况下，可以说信号带宽必须小于电缆带宽。用于数据传输的电话线路的有效带宽是2 400Hz，覆盖的频率范围是600～3 000Hz。注意：目前一些电话线路能够处理比传统线路更高的带宽。但是，调制解调器的设计仍然是基于传统线路容量的（见图9.6）。



图9.6 电话线带宽

术语调制解调器（modem）是一个复合词，源于组成设备的两个功能实体：信号调制器和信号解调器。**调制器**（modulator）从二进制数据生成带通模拟信号。**解调器**（demodulator）将调制信号中恢复为二进制数据。

> 调制解调器代表调制器/解调器。

图9.7说明了调制解调器与通信链路的关系。左侧的计算机发送数据到调制解调器的调制器部分，数据作为模拟信号在电话线路上发送。右侧的调制解调器接收到模拟信号，通过解调器解调，将数据分发给右侧的计算机。通信可以是双向的，表示右侧的计算机可以使用同一调制/解调过程向左侧的计算机发送数据。

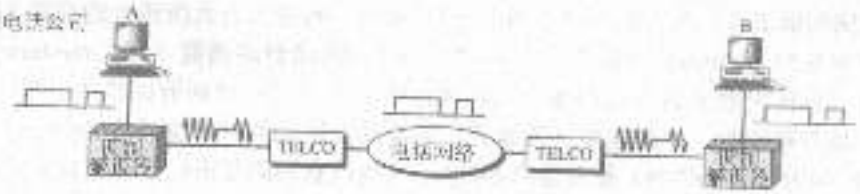


图9.7 调制器/解调器

调制解调器标准

目前，许多最流行的调制解调器都是基于ITU-T发布的V系列（V-series）标准。我们只讨

论最新的一系列标准。

V.32和V32bis

V.32调制解调器结合使用调制和编码技术，称为格码调制（trellis-coded modulation）。格式结构本质上是QAM加上一位冗余位。数据流划分为多个4位部分，但是，传输的是五位组（pentable）而不是四位组（quadbit）。额外的位值从数据位值中计算得到，它用于差错检测。

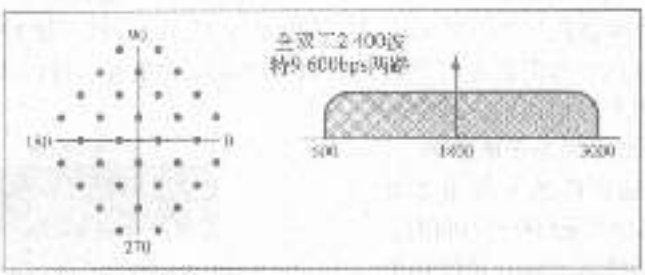
V.32需要波特率为2 400的32-QAM。因为每个五位组只有4位表示数据，所以最后所得到的速度为4×2 400＝9 600bps。星座图和带宽如图9.8所示。

V.32bis调制解调器是第一个支持14 400bps传输的ITU-T标准。V.32bis使用的是在波特率为2 400波特（2 400×6＝14 400bps）下的128-QAM传输（7位/波特，1位用于差错控制）。

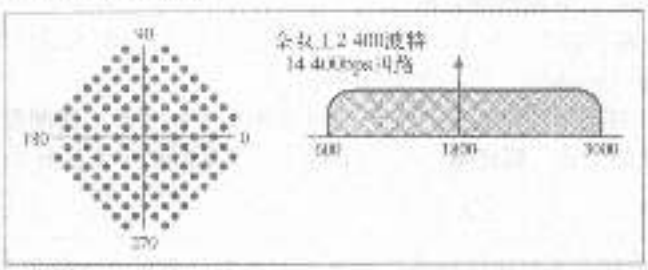
V.32bis提供的附加增强功能包含了传输速度自动降低和提高等特性，使得调制解调器能够依据线路或者信号的质量调高或者调低速度。星座图和带宽如图9.8所示。

V.34 bis

V.34bis调制解调器为28 800的比特率提供了960点的星座图，为33 600bps的比特率提供了1 664点的星座图。



a) V.32最终调制带宽



b) V.32bis最终调制带宽

图9.8　V.32和V.32bis星座图和带宽

V.90

传统的调制解调器的最大数据速率为最大33.6kbps，如香农公式所确定的那样（见第3章）。但是，比特率是56 000bps的V.90调制解调器，称为56K调制解调器（56K modem），已经可用了。只有一方使用数字信号（如通过因特网提供商）时这些调制解调器才可用，下载速率与上载速率是不对称的，下载速率（数据流从因特网提供商到PC）最大是56kbps，上载速率（数据流从PC到因特网提供商）最大是33.6kbps。这些调制解调器违反了香农容量原理吗？不，在下行方向，由于没有量化误差，所以SNR很高（见图9.9）。

在上传（uploading）过程中，模拟信号仍然在交换站进行采样，在这个方向，量化噪声（见第4章）引入信号降低了信噪比（SNR），将上载的数据速率限制为33.6kbps。

然而，在下载（downloading）过程中没有采样。信号不受量化噪声的影响，不受香农极

限容量原理的限制。上载方向的最大数据速率仍然是33.6kbps，但是下载方向的数据速率现在是56kbps。

人们可能想知道为什么是56kbps。电话公司每秒采样8 000次，每个采样使用8位，每一个采样的一位用于控制目的，意味着每个采样只有7位，所以速率是8 000×7，或者是56 000bps，即56kbps。

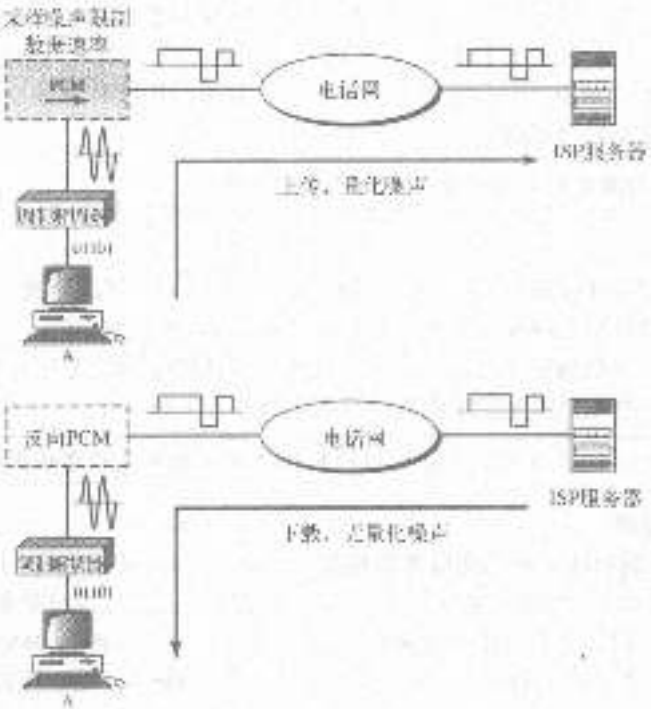


图9.9 56K调制解调器

V.92

V.90之后标准是V.92。这些调制解调器能自己调整速度，如果噪声允许，上传数据速率可以是48kbps。下载速率仍然是56kbps。这种调制解调器具有一些附加特性。例如，如果线路有呼叫等待服务，在有电话拨入时，调制解调器可以中断因特网连接。

9.3 数字用户线路

在传统调制解调器到达其最高数据速率以后，电话公司开发出了另一种技术——**数字用户线路**（digital subscriber line，DSL）技术来提供对因特网的高速访问。DSL技术是在现有本地回路上支持高速数字通信的最有前途的技术之一。DSL技术是一组技术，每一种只是首字母不同（ADSL、VDSL、HDSL和SDSL）。这组技术通常称为xDSL，这里的x可用A、V、H或S来代替。

9.3.1 ADSL

xDSL技术中的第一种技术是**非对称数字用户线路**（asymmetrical DSL，ADSL）。与56K调制解调器相似之处是，ADSL的下行（从因特网到住户）速率（比特率）比上行（从住户到因特网）速率要高。这就是称为“非对称”的原因。与56K调制解调器的非对称性的不同之处是，ADSL是专门针对住宅客户设计的，将本地回路的可用带宽进行了不均衡的划分。这种业务不

适合于那些在两个传输方向都需要较高带宽的商业客户。

ADSL是专门针对住宅用户而设计的非对称通信技术，它不适合于商业应用。

使用现有本地回路

一个值得注意的方面是ADSL使用现有的本地回路。但是，ADSL是如何达到传统的调制解调器永远不能实现的数据速率呢？答案是，双绞线本地回路能够处理的带宽实际上高达1.1MHz，但是电话公司安装在本地交换局的滤波器将带宽限制到4kHz（对语音通信已经足够）。这样做允许对大量语音通道进行多路复用。然而，如果将滤波器拆除，那么整个1.1MHz行宽就可以全部用于数据和语音通信了。

现有的本地回路能够处理的带宽可以高达1.1MHz。

自适应技术

1.1MHz只是本地回路的理论带宽。诸如住宅与交换局之间的距离、电缆的尺寸以及所用的传输信号等因素都会影响带宽。ADSL技术的设计者们意识到了这一问题，便采用一种自适应技术：在设定某种数据速率之前，会运用该技术测试线路的状况和可用带宽。ADSL的数据速率不是固定的，它根据本地回路电缆的状况和类型而有所变化。

ADSL是一种自适应技术，系统使用的数据速率依据本地回路的线路状况而定。

离散多音频技术

ADSL的标准调制技术称为离散多音频技术（discrete multitone technique，DMT），它将正交振幅调制（QAM）和频分多路复用（FDM）结合在一起。系统带宽的划分没有固定的方法，每个系统都可以决定自己的带宽如何划分。一般情况下，将1.104MHz的可用带宽划分为256个通道，每个通道使用的带宽是4.312kHz，如图9.10所示。图9.11表示了带宽是如何进行划分如下：

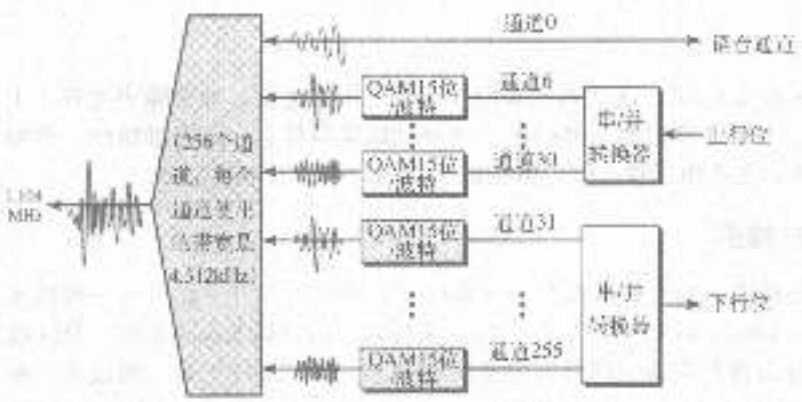


图9.10　离散多音频技术

- 语音　保留通道0用于语音通信。
- 空闲　通道1～5不使用，作为语音和数据通信之间的间隙。
- 上行数据与控制　通道6～30（25个通道）用于上行数据传输和控制。1个通道用于控制，24个通道用于数据传输。如果有24个通道，每个使用4kHz（在可用带宽4.132kHz之内）带宽用QAM调制，那么在上行方向可有24×4 000×15，即1.44Mbps的传输速率，然而，由于某些载波在噪声大处被剔除，通常数据速率在500kbps以下。换言之，有些通道可

能不可用。

- 下行数据与控制：通道31～255（225个通道）用于下行数据传输和控制。1个通道用于控制，而其他224个通道用于数据传输。如果有224个通道，则可获得224×4 000×15，即13.4Mbps的传输速率。然而，由于某些载波在噪声大处被删除，通常数据速率在8Mbps以下。换言之，有些通道可能不可用。

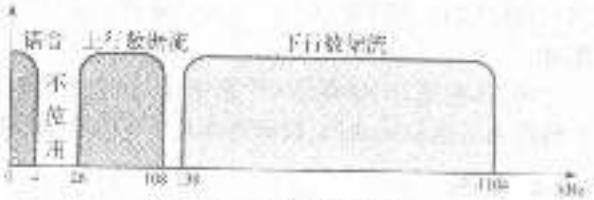


图9.11 ADSL中带宽划分

客户端：ADSL调制解调器

图9.12表示了安装在客户端的ADSL调制解调器。本地回路连接到分离语音和数据通信的滤波器上。ADSL调制解调器使用DMT调制数据并创建上行和下行通道。

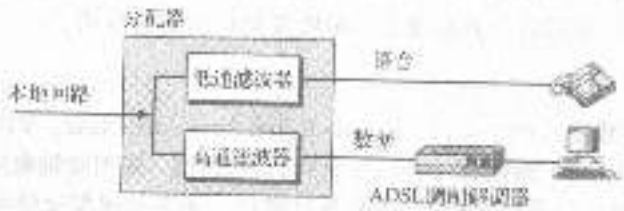


图9.12 ADSL调制解调器

注意：一般分配器由电信公司技术员安装在客户端，语音线可用在房间中已有的电话线，但数据线需要由专业人员来安装。这些使ADSL的代价高昂，我们将看到有另一种运用ADSL（或ADSL-Lite）可供选择。

电话公司端：DSLAM

在电话公司一端，情况有所不同。这一端安装了一种称为**数字用户线路接入复用器**（digital subscriber line access multiplexer，DSLAM）的设备来取代ADSL调制解调器。这种设备与ADSL调制解调器的功能相似。另外，它还负责将发送到因特网（ISP服务器）的数据进行分组。图9.13说明了这种配置。

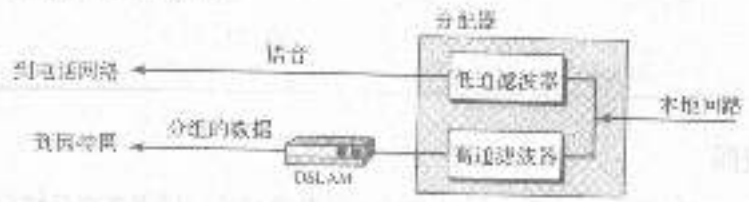


图9.13 DSLAM

9.3.2 ADSL Lite

在建筑物的边界处安装分配器和新的数据线的价格昂贵，多数用户难以接受。一种新型号的ADSL称为ADSL-Lite（通用的ADSL或无分配器ADSL）的技术适用这些用户。这种技术允许ADSL-Lite调制解调器直接插入电话机插座而连接计算机。分配工作由电话公司完成，ADSL-Lite使用8位（代替15位）调制256DMT的载波。由于语音信号产生的错误可能与他们混合，所以有些载波可能不适用。它提供最大下行数据速率为1.5Mbps，上行数据速率为512kbps。

9.3.3 HDSL

高速数字用户线路（high-bit-rate digital subscriber line，HDSL）是为了替代T-1线路（1.544Mbps）而设计的。T-1线路使用AMI编码，因为AMI在高频频段对衰减非常敏感，所以将T-1线路的长度限制为1km，如果要传输更长的距离，就必须有中继器，这也意味着费用的增加。

HDSL使用2B1Q编码（见第4章），对衰减的敏感程度要低得多。在不使用中继器的情况下，传输距离可达3.6km时，数据速率几乎可以达到2Mbps。HDSL使用双绞线实现全双工传输。

9.3.4 SDSL

对称数字用户线路（symmetric digital subscriber line，SDSL）是HDSL一种双绞线型，它提供全双工对称通信，在每个方向支持速率达768kpbs。提供对称通信的SDSL可取代ADSL。ADSL提供非对称通信，下行比特率远高于上行比特率，尽管这种特性满足大多数住宅用户的需要，但是它不适用于那些双向都需要大容量收发数据的商业应用。

9.3.5 VDSL

甚高速数字用户线路（very-high-bit-rate digital subscriber line，VDSL）是与ADSL类似的一种替代实现技术，适于短距离（300～1 800m）的通信，使用同轴电缆、光纤或者双绞线，采用DMT调制。它对上行通信在距离3 000英尺到10 000英尺提供比特率的范围是25Mbps到55Mbps，下行比特率通常是3.2Mbps。

9.3.6 总结

表9.1显示了DSL技术的总结。注意：数据速率和距离是近似的，从一种实现方法到另一种实现方法是可变的。

表9.1　DSL总结

技　术	下行速率	上行速率	距离（ft）	双绞线对	线路代码
ADSL	1.5～6.1Mbps	16～640kbps	12 000	1	DMT
ADSL Lite	1.5Mbps	500kbps	18 000	1	DMT
HDSL	1.5～2.0Mbps	1.5～2.0Mbps	12 000	2	2B1Q
SDSL	768kbps	768kbps	12 000	1	2B1Q
VDSL	25～55Mbps	3.2Mbps	3000～10 000	1	DMT

9.4 有线电视网

有线电视网（cable TV network）开始是作为视频提供者，但现在它已转为因特网接入商务服务。在这一节里，我们讨论有线电视网本身的内容，在9.5节中，讨论有线电视网如何用来提供高速接入因特网。

9.4.1 传统有线网络

20世纪40年代后期，**有线电视**（cable TV）开始将视频信号分散传输到没有接收能力或者接收能力很弱的地区，原来称为公共天线电视（community antenna TV，CATV），是因为位于高山顶或者建筑物顶端的天线从电视台接收信号，通过同轴电缆将这些信号分配给社区公用。图9.14说明了传统的有线电视网络的示意图。

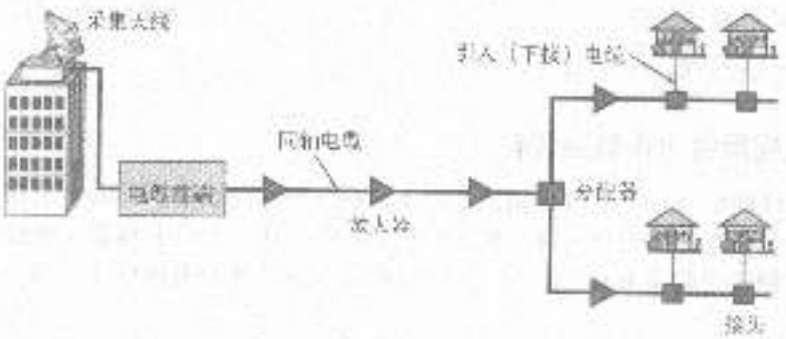


图9.14 传统的有线电视网络

有线电视局，称为**前端**（head end），从广播电台接收视频信号并将信号馈送给同轴电缆。由于信号会变得越来越弱，所以在网络的通路上安装放大器对信号进行放大。在前端与用户住宅之间会有多达35个放大器。另一端，分配器分离电缆信号，通过分接头和分接电缆与用户住宅建立连接。

传统的有线电视系统使用同轴电缆进行端与端的连接。由于信号的衰减和需要使用大量的放大器，传统网络中的通信是单向的（一个方向）。视频信号是下行传输的，即向前端到用户住宅。

> 传统有线电视网中的通信是单向的。

9.4.2 光纤同轴电缆混合（HFC）网络

第二代有线网络称为**光纤同轴电缆混合**（hybrid fiber coaxial，HFC）**网络**。这一网络混合使用光纤和同轴电缆。从有线电视局到称为**光纤节点**（fiber node）的分线盒之间的传输介质是光纤，而从光纤节点穿过相邻住户并到达房屋内部的线路仍然使用同轴电缆。图9.15说明了HFC网络的示意图。



图9.15 光纤同轴电缆混合（HFC）网络

地区有线前端（regional cable head，RCH）通常可以为多达400 000个用户服务。RCH将信号馈送到**分配集线器**（distribution hub），每一台集线器可以为40 000个用户服务。分配集线器在这一新的基础设施中发挥着重要作用。信号的调制和分配都在这里完成，经过处理的信号再通过光纤馈送到光纤节点。光纤节点将模拟信号进行分配，将相同的信号发送到每一条同轴电缆。每一条同轴电缆为1 000个用户服务。光纤的使用可以将所需放大器的数量减少到8个或者更少。

从传统基础设施转换到混合基础设施的一种原因是使有线网络成为双向的（两个方向）。

HFC有线电视网络中的通信可以是双向的。

9.5 有线电视网络用作数据传输

目前，有线网络公司与电信公司正在竞争那些需要高速数据传输的住宅用户。DSL技术为住宅用户通过本地回路提供高速连接，但是，DSL使用已有的非屏蔽双绞线，对干扰非常敏感，这对数据速率的上限有了限制。另一种解决方案是使用有线电视网络。本节简要讨论这种技术。

9.5.1 带宽

即使是在HFC系统中，在网络的最后一部分，即从光纤节点到用户住宅之间仍然是同轴电缆。同轴电缆的带宽位于5－750MHz的范围（近似数）。电缆公司将这一带宽划分为三个波段：视频、下行数据和上行数据，如图9.16所示。



图9.16　有线电视划分同轴电缆波段

下行视频波段

下行视频波段（downstream video band）占用的频率范围是54－550MHz，由于每个电视频道占用6MHz，这可以满足80多个频道的需要。

下行数据波段

下行数据（从因特网到用户住宅）占用上端波段，频率范围是550－750MHz。这一波段也是按6MHz划分为多个通道。

调制　下行数据波段（downstream data band）使用64-QAM（也可能是256-QAM）调制技术进行调制。

下行数据使用64-QAM调制技术进行调制。

数据速率　在64-QAM调制技术中，每一个波特使用6位。每个波特中，1个位用于前向纠错，5个位用于数据。该标准规定每个波特为1Hz。这意味着从理论上讲，下行数据能以30Mbps（5bit/Hz×6MHz）的速率接收。该标准仅指定27Mbps。然而，由于有线调制解调器是通过10Base-T（见第13章）电缆连接到计算机上的，因此将数据速率限制为10Mbps。

理论上的下行数据速率是30Mbps。

上行数据波段

上行数据（从用户住宅到因特网）占用低端波段，频率范围是5－42MHz。这一波段也是按6MHz划分为多个通道。

调制　上行数据波段（upstream data band）使用较低的频率，对噪声和干扰更为敏感。由于这种原因，QAM不适用于这个波段。更好的解决方案是使用QPSK。

上行数据使用QPSK调制技术进行调制。

数据速率　在QPSK调制技术中，每波特使用2个位。该标准规定每波特是1Hz。这意味着从理论上讲，下行数据能以12Mbps（2bit/Hz×6MHz）的速率发送。然而，数据速率通常小

于12Mbps。

> 理论上的上行数据速率是12Mbps。

9.5.2 共享

上行和下行波段可以由用户共享。

上行波段共享

上行数据带宽只有37MHz，这表示在上行方向只有6条6MHz的通道可用，用户需要使用一条通道在上行方向发送数据。问题是，6条通道如何在含有1 000，2 000甚至100 000个用户的区域内共享？解决方案是分时共享。这一波段使用FDM划分为多条通道，这些通道必须在具有相邻关系的用户间共享。有线提供商以静态或者动态方式，将一条通道分配给一组用户。如果某个用户要发送数据，他需要与要访问这一通道的其他用户竞争，即用户必须等待直到通道可用。

下行波段共享

下行方向的情况与上行方向类似。下行波段有33个6MHz的通道。有线提供商的用户数可能多于33个，所以每条通道必须在一组用户之间共享。然而，对下行方向来说，情况就有所不同，这里存在多播的情况。只要存在用户组中任何一个用户的数据，这些数据就会发送到该通道。数据会发送到该用户组中的每一个用户。但是，由于每个用户都具有提供商提供的注册地址，用户组的有线调制器就会将传输数据中包含的地址与提供商提供的地址进行匹配。如果地址匹配，数据就保留下来，否则就被丢弃。

9.5.3 CM与CMTS

为了使用有线网络进行数据传输，需要两种关键设备：**有线调制解调器**（cable modem，CM）和**有线调制解调器传输系统**（cable modem transmission system，CMTS）。

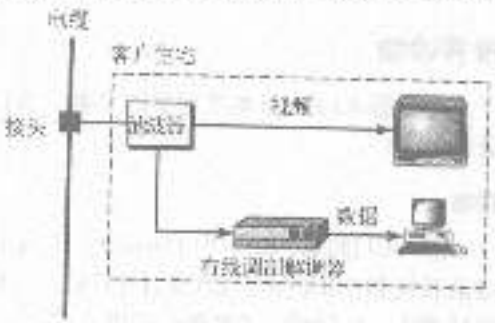


图9.17 有线调制解调器（CM）

CM

有线调制解调器（cable modem）安装在用户的住宅内，与ADSL调制解调器类似。图9.17说明了它的位置。

CMTS

有线调制解调器传输系统（cable modem transmission system，CMTS）由有线公司安装在分配集线器中。它从因特网接收数据，并传送给混合器，由混合器发送给用户。CMTS也接收来自用户的数据并将它们传送到因特网。图9.18说明了CMTS的位置。

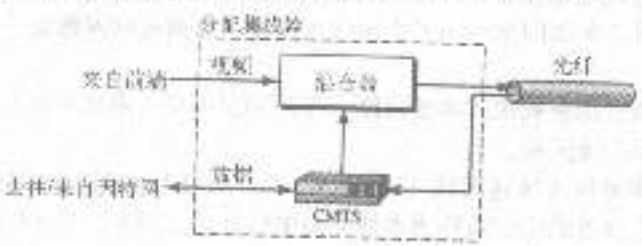


图9.18 有线调制解调器传输系统（CMTS）

9.5.4 数据传输方案：DOCSIS

在最近几十年，已经设计了多种方案用于创建通过HFC网络进行数据传输的标准。最流行的一种方案是由多媒体有线网络系统（MCNS）设计的，称为有线数据系统接口规范（Data Over Cable System Interface Specification，DOCSIS）。DOCSIS定义了将数据从CMTS传输到CM所必需的全部协议。

上行通信

DOCSIS定义了一种用于上行通信的协议。下面是这种协议高度简化的版本。它描述了CM必须遵循的步骤：

1. CM检查CMTS定期发送的特定分组的下行通道。该分组要求有任何一个新的CM声明自己位于一条特定的上行通道。

2. CMTS向CM发送一个分组，规定分配给它的下行通道和上行通道。

3. CM随后启动一个过程，称为测距（ranging），以确定CM与CMTS之间的距离。这一过程是必需的，用于所有CM与CMTS之间的同步以获取上行通道分时共享时所使用的微时隙。在第12章讨论竞争协议时，读者会了解到关于分时共享的知识。

4. CM向ISP发送一个分组，请求连特网地址。

5. CM与CMTS随后交换一些分组以建立安全参数。公用网络（比如有线电视网络）需要使用这些参数。

6. CM向CMTS发送它的唯一标识符。

7. 上行通信在所分配的上行通道中开始，CM会竞争微时隙来发送数据。

下行通信

在下行方向，通信非常简单，因为只有一个发送方，所以没有竞争。CMTS使用所分配的下行通道，发送包含接收CM地址的分组。

推荐读物

为了深入讨论与本章有关的主题，我们推荐下列读物，在本书末列出括号[……]中的参考资料。

读物

[Cou01]的第8章给出电话系统、DSL技术和CATV引起关注的讨论。[Tan03]的2.5节讨论电话系统和DSL技术，2.7节讨论CATV。[GW04]1.1.1节讨论电话系统、3.7.3节讨论标准的调制解调器。[Max99]可找到全部用户宽带（DSL和CATV）的讨论。

本章小结

- 电话是指普通传统电话系统（POTS），最初是一个使用模拟信号传输语音的模拟系统。在最近几十年里，电话网络经历了多次技术改进，目前电话网络是一种数字和模拟共存的网络。
- 电话网络主要由三部分构成：本地回路、干线和交换局。电话网络具有多级交换局，如本地局、中继局和地区局。
- 美国划分多个本地接入传送区域（LATA）。一个LATA内部提供的服务称为LATA内部服务，处理这些服务的运营商称为本地交换电信公司（LEC）。LATA之间的服务由跨区交换电信公司（IXC）处理。

- 在带内信令中，信令和电话两者用同一电路。在带外信令中，带宽的一部分用于信令，另一部分用于数据传输。电话网络中信令所用的协议是7号信令系统（SS7）。
- 电话公司提供两类业务：模拟业务和数字业务。这些业务可以分为两类：模拟交换业务和模拟租用业务。两种最常见的数字业务是交换式/56业务和数字数据业务（DDS）。
- 用电话本地回路进行数据传输传统上是用拨号调制解调器完成的。术语调制解调器是一个复合词，它是指由两个功能实体组成的设备：信号调制器和信号解调器。
- 许多流行的调制解调器是基于V系列标准。V.32调制解调器的数据速率为9 600bps。V.32bis调制解调器支持14 400bps传输。V.90调制解调器称为56K调制解调器，下载数据速率为56kpbs，上传数据速率是普通的33.6kbps。V.90之后标准是V.92。这些调制解调器能自己调整速度。如果噪声允许，上传数据速率可以是48kbps。
- 电话公司开发出了另一种技术，称为数字用户线路（DSL）技术来提供对因特网的高速访问。DSL技术是一组技术，每一种只是首字母不同（ADSL、VDSL、HDSL和SDSL）。ADSL是供下行的速率高于上行速率；高速数字用户线路（HDSL）是为了替代T-1线路（1.544Mbps）而设计的；对称数字用户线路（SDSL）是HDSL一样双绞线型；甚高速数字用户线路（VDSL）是与ADSL类似的一种替代实现技术。
- 公用天线TV（CATV）本来为社区提供视频业务而设计的。传统的有线TV系统使用同轴电缆实现端到端的传输。第二代有线网络称为光纤同轴电缆混合（HFC）网络。这一网络混合使用光纤和同轴电缆。
- 目前，有线网络公司与电信公司正在竞争那些需要高速接入因特网的住宅用户。为了使用有线网络进行数据传输，需要两种关键设备：有线调制解调器（CM）和有线调制解调器传输系统（CMTS）。

习题

复习题

1. 电话网络二个主要部分是什么？
2. 给出电话网络分等级交换层次。
3. LATA是什么？LATA内部服务和LATA间服务是什么？
4. 叙述SS7服务并说明它与电话网络关系。
5. 美国电话公司提供两种主要业务是什么？
6. 有线调制解调技术是什么？列出本章所讨论的常用调制解调器标准并给出它们的数据速率。
7. DSL技术是什么？DSL电话公司提供什么服务？
8. 将传统有线网络与混合光纤同轴电缆网络进行比较，说出它们的不同之处。
9. 使用CATV通道如何进行数据传输？
10. 分清有线调制解调器（CM）和有线调制解调器系统（CMTS）。

练习题

11. 用第8章中的电路交换，解释电话网络为何选择这种类型的交换？
12. 用第8章中的电路交换网所说的三个通信阶段，将这些阶段与电话呼叫双方所建立的阶段相比较。
13. 在第8章中，已知道电路交换网在建立阶段和拆除阶段期间需要端到端的寻址。解释电话网络通信双方端到端的寻址。

14. 当我们打越洋电话时，有时会感觉到延迟。能说明其原因吗？
15. 画一个条形统计图比较用常用调制解调器的不同下载数据速率。
16. 画一个条形统计图比较用常用的DSL技术实现的不同下载数据速率（使用最小数据速率）。
17. 使用下列技术计算，下载1 000 000字节所需的最小时间：
    a. V.32调制解调器　　b. V.32bis调制解调器
    c. V.90调制解调器
18. 使用不同的DSL实现（考虑最小速率）重做练习17。
19. 使用有线调制解调器（考虑最小速率）重做练习17。
20. 在某一区域中客户为了数据传输使用DSL调制解调器，所用的拓扑结构是什么类型？并说明之。
21. 在某一区域中客户为了数据传输使用有线调制解调器，所用的拓扑结构是什么类型？并说明之。

第三部分　数据链路层

目标

数据链路层把物理层的原始传输设施转换成一条负责点到点（逐跳）通信的链路。数据链路层的具体任务包括成帧、寻址、流量控制、差错控制和介质访问控制。数据链路层把从网络层接收到的位流划分为帧，以便易于处理。数据链路层增加头部用来规定发送方和接收方地址。如果数据在接收方的速率低于发送方的发送速率，则数据链路层会实行流量控制机制来防止数据大量涌入接收方。数据链路层还增加了一些机制用来检测和重传被损坏帧、防止重复或丢失帧。当两个或更多的设备连接到一条链路上，就需要数据链路层协议来确定在任一给定时刻具体哪个设备控制这条链路。

在本书的第三部分，我们先讨论数据链路层提供的服务，然后讨论这些服务在局域网（LAN）中的实现，最后我们讨论广域网（WAN）如何应用这些服务。

第三部分重点介绍数据链路层以及该层提供的服务。

第10章 检错与纠错

网络必须能够以可接受的准确率将数据从一个设备传送到另一个设备。对于大多数应用，系统必须保证接收到的数据与发送的数据是一致的。将数据从一个节点传送到下一个节点的过程中，数据有可能遭到破坏。有许多因素影响或改变一个数据报文的一位或多位。有些应用要求有一个机制用以检测或纠正这些差错（error）。

> 数据在传输的过程中可能遭到破坏，一些应用需要进行检错和纠错。

有些应用能忍受一些小差错，比如，可以忍受音频或视频传输中的随机差错，但是传输文本时，期望得到很高的准确率。

10.1 引言

首先直接地或间接地讨论有关差错检测和差错纠正问题。

10.1.1 差错的类型

无论何时，在位流从一点流动到另一点时，由于**干扰**（interference）的存在，都可能经受到不可预测的变化，这些干扰可能会改变信号的波形。在单个位差错中，就是由0变成1或是由1变成0。在突发性差错中，可能有多个位的改变。例如，在一个具有1 200bps速率的传输中，一个0.01s的脉冲噪声突发可能要改变12位信息中一些位或所有位。

单个位差错

单个位差错（single-bit error）一词的意思是在给定的数据单元（例如一个字节、字符或分组）中仅有一位发生从1到0或从0到1的变化。

> 在一个单个位差错中，数据单元中仅有一位发生变化。

图10.1说明在一个数据单元中单个位差错的影响。为了理解这一变化的影响，设想每一个8位组是一个左边加一个0位的ASCII码字符。在图10.1中，发送00000010（ASCII的STX），表示文本的开始，但是接收方接收的是00001010（ASCII的LF），表示换行（有关ASCII码更详细的信息，可以参考附录A）。

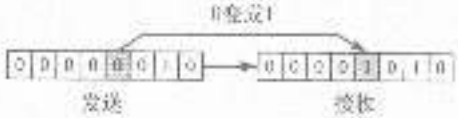


图10.1 单个位差错

单个位差错在串行数据传输中很少出现。要了解为什么，可以想像以1Mbps发送数据的情况，这意味着每一位仅持续1/1 000 000s即1μs。要出现单个位差错，噪音信号必须只有1μs的持续时间，而这是非常罕见的情况。一般的噪音信号都比这个时间要长。

突发性差错

突发性差错（burst error）是指在数据单元中有两位或更多位发生1到0或0到1的变化。

> 一个突发性差错意味着在数据单元中有两位或多位发生变化。

图10.2说明在数据单元中的一个突发性差错的影响。在这种情况下，发送的是0100010001000011，但是接收的却是0101110101000011。注意：突发性差错并不意味着差错在连续位上出现，突

发性差错的长度是从第一个差错位到最后一个差错位的长度计算。其间的某些位也可能未产生差错。

突发性差错比单个位差错更容易发生。噪声的持续时间要大于一个位的持续时间，这就意味着在噪声影响数据时，它要影响许多位。受影响的位的数量取决于数据速率和噪声的持续时间。例如，如果以1Kbps的速率发送数据，一个1/100s的噪声就会影响10位；而如果我们以1Mbps的速率发送数据，同样的噪声就会影响10 000位。

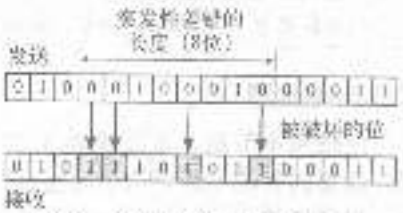


图10.2 长度为8位的突发性差错

10.1.2 冗余

检错或纠错的核心概念是冗余（redundancy）。为了能检测或纠正差错，我们除了数据外还需要发送一些额外的位。这些冗余位由发送方添加并由接收方除去。它们的存在允许接收方检测或纠正被破坏的位。

> 为了检测或纠正差错，我们需要发送除了数据外的额外（冗余）位。

10.1.3 检错和纠错

差错的纠正比检测更难。在检错（error detection）中，只看是否发生差错。回答是简单的是或否。甚至对差错个数都不感兴趣。对我们来说，单个位差错和突发性差错是一样的。

在纠错（error correction）中，需要知道被破坏的位的个数，更重要的是知道它们在报文中的位置。差错个数和报文长度是重要的因素。如果需要纠正8位数据单元中的单个位差错，则需要考虑8个可能的差错位置。如果需要纠正相同长度数据单元中的两个差错，则需要考虑28种可能。可以想像，要在1 000位数据单元中查找10个差错，是多么难了。

10.1.4 前向纠错和重传

有两种主要的纠错方法。**向前纠错**（forward error correction）是接收方通过使用冗余位尝试推测报文的方法。正如在后面看到的那样，如果差错的个数很小，这是可能的。通过重传（retransmission）纠错是接收方检测出有差错发生并要求发送方重新发送报文的技术。重复重发，直到接收方认为到达的报文无差错（一般，不是所有的差错都可以被检测到）。

10.1.5 编码

通过各种编码方案实现冗余。发送方增加冗余位，通过某种方法建立冗余位和真实数据位之间的某种关系。接收方检测这两者的关系来检错或纠错。冗余位和数据位的比率以及方法的健壮性是任何编码方案的重要因素。图10.3说明了编码的一般概念。

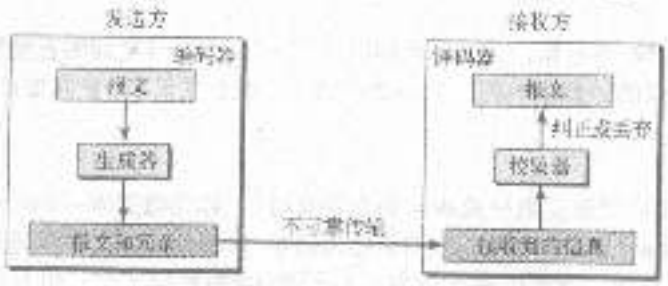


图10.3 编码器和译码器的结构

可以把编码方案分成两大类：块编码（block coding）和卷积编码（convolution coding）。本书中只专注于块编码，而卷积编码更加复杂超出了本书的范围。

> 在本书中，介绍块编码。会在更高级的书中介绍卷积编码。

10.1.6 模运算

在结束本节前，首先简要地讨论一般的计算机科学尤其是检错和纠错的基本概念：模运算。这里的目的不是对这个问题进行深入的数学讨论，而只是给出本章所需的内容。

在模运算（modular arithmetic）中，只使用有限范围中的正整数。它的上限称为**模数**（modulus）N。然后使用0到$N-1$的正整数。这是以N为模数的运算。例如，如果模数是12，我们只使用0到11的整数。我们的时钟系统是模运算的一个实例。它基于以12为模数的运算，12由0代替。在模N系统中，如果数字大于N，则除以N并取余数。如果它是负数，需要加上N的倍数使其变为正整数。再次考虑时钟系统。如果我们在早上11点开始工作，并且工作需要5个小时，如果在军队中我们会说在16点结束工作，或者我们会说在下午4点结束（16/12的余数是4）。

> 在以N为模数的运算中，我们只使用0到$N-1$的正整数。

模运算中的加法和减法是简单的。在同一列上两个数字相加没有进位，在同一列上两个数字相减没有借位。

模2运算

特别感兴趣的是模2运算。在这个运算法中，模数N为2。我们只能使用0和1。这个运算法的操作很简单。如下给出了我们如何加减2个位。

加：0+0=0 0+1=1 1+0=1 1+1=0

减：0−0=0 0−1=1 1−0=1 1−1=0

特别注意，加法和减法给出了相同的结果。在这个运算中，对加法和减法，使用XOR（异或）运算。如果两个位相同，XOR运算的结果是0；如果不同，则结果是1。图10.4给出了这个运算。

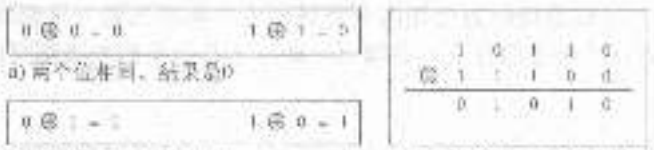


图10.4　两个单个位或两个字的异或运算

其他模数运算

在本书中也使用模$-N$运算。其原理是相同的，使用0到$N-1$之间的正整数。如果模数不是2，加法和减法是不同的。如果得到一个为负的结果，就加上倍数N使其变成正整数。

10.2 块编码

在块编码中，我们把报文划分成块，每个块有k位，称为**数据字**（dataword）。并增加r个冗余位使其长度变为$n=k+r$，形成n位的块称为**码字**（codeword）。如何选择或计算这额外的r位将在后面讨论。目前，重要的是知道有一组长度为k的数据字和一组长度为n的码字。k位

就有2^k个数据字组合，n位就有2^n个码字组合。既然$n>k$，可能的码字数大于可能的数据字数。块编码处理是一对一，相同的数据字总是编码成相同的码字。这意味着$2^n - 2^k$个码字没有使用。我们称这些码字为无效或非法码。图10.5说明了这个情况。



图10.5 块编码中的数据字和码字

例10.1

第4章中讨论过的4B/5B块编码是这种编码的一个好示例。在这个编码方案中，$k=4$，$n=5$。正如所看的那样，有$2^4=16$个数据字以及$2^5=32$个码字。32个码字中只有16个用于报文传输。其余的16个或者用于其他目的或者不用。

10.2.1 差错检测

如何使用块编码检测差错呢？如果满足以下两个条件，接收方就能检测出原来码字的一个差错。

1. 接收方有（或能够找到）有效码字的列表。
2. 原来码字已改变成无效的码字。

图10.6表示了差错检测中块编码的作用。

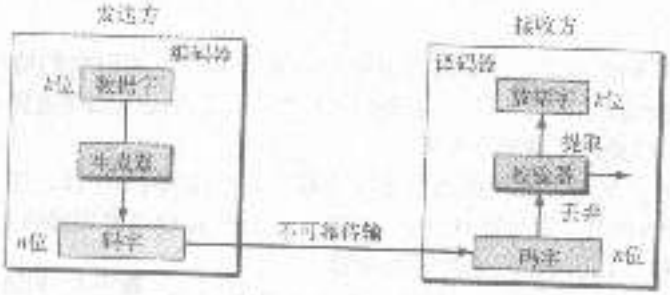


图10.6 使用块编码的差错检测过程

通过使用编码规则和过程的生成器（后面讨论）发送方根据数据字生成码字。每个发送给接收方的码字在传输中可能会出错。如果接收到的码字是有效码字中的一个，就接收这个码字，提取出对应的数据字。如果接收到的码字不是有效的，就将它丢弃。但是，如果码字在传输中被破坏，但接收到的码字仍然是一个有效的码字，差错就无法被检测到。这种类型的编码只能检测出简单的差错。两个或更多的差错可能就无法被检测到。

例10.2

假设$k=2$，$n=3$。表10.1给出了数据字和码字列表。后面将会看到如何从数据字生成码字。

表10.1 差错检测码（例10.2）

数据字	码字
00	000
01	011
10	101
11	110

假设发送方把数据字01编码成011并发送给接收方。考虑以下的情况：

1. 接收方接收到011，它是有效的码字。接收方从它提取出数据字01。

2. 码字在传输中被破坏，接收方接收到111（最左边的位被破坏）。这是无效的码字被丢弃。

3. 码字在传输中被破坏，接收方接收到000（右边两位被破坏）。这是个有效码字，接收

方错误地提取出数据字00。这两个破坏位形成的差错无法被检测到。

> 差错检测码是根据某些类型的差错而设计的，因此只能检测到这些类型的差错，其他类型的差错无法被检测到。

10.2.2 纠错

如前述，纠错比检错更复杂。在检错中，接收方只需要知道接收到的码字是无效的，而在纠错中，接收方需要知道（或推测）发送的原来码字。可以说纠错比检错需要更多的冗余位。这个概念与检错一样，只是校验可能更加复杂。图10.7表示了纠错中块编码的任务。

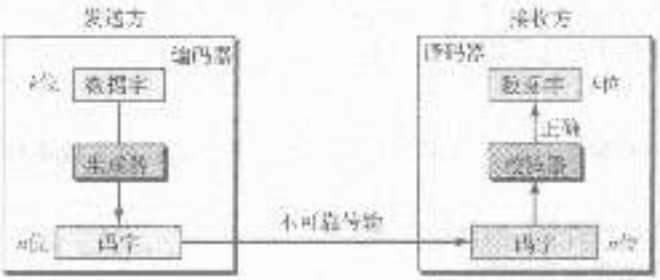


图10.7　差错纠正中编码器和译码器的结构

例10.3

给例10.2加入更多的冗余位，看接收方在不知道实际发送的码字的情况下是否能纠正差错。我们给2位数据字加入3个冗余位，形成5位码字。后面会说明如何选择冗余位，现在集中说明纠错思想。表10.2给出了数据字和码字。

假设数据字是01，发送方参照该表（或使用算法）生成码字01011。这个码字在传输中被破坏，接收方接收到01001（右边第2位发生差错）。首先，接收方发现接收到的码字不在表中。这意味着发生了差错。（必须在纠错前进行检错）。假定只有1位被破坏，接收方使用以下策略来推测正确的码字。

表10.2　纠错码（例10.2）

数据字	码字
00	00000
01	01011
10	10101
11	11110

1. 用表中第一个码字与接收到的码字进行比较（01001和00000），因为有两位不同所以接收方确定第一个码字不是发送的那个码字。

2. 相同的原因可知表中的第三个和第四个码字不可能是原来的码字。

3. 原来码字一定是表中的第二个码字，因为与接收到的码字只有一位不同。接收方用01011代替01001，并参照表找到原来数据字01。

10.2.3 汉明距离

用于差错控制编码的一个核心概念是汉明距离。两个（相同长度）字的汉明距离（Hamming distance）是对应位不同的数量。我们以$d(x, y)$表示两个字x和y之间的汉明距离。如果我们对两个字进行异或操作并计算1的个数，就可以很容易地得出汉明距离。注意：汉明距离大于零。

> 两个字的汉明距离是对应位不同的数量。

例10.4

让我们来导出下列两对字的汉明距离。

1. 汉明距离d（000，011）是2，因为000⊕011是011（两个1）。

2. 汉明距离d（10101，11110）是3，因为10101⊕11110是01011（三个1）。

10.2.4 最小汉明距离

虽然汉明距离概念是处理检错和纠错的核心概念，但是用于设计编码的度量是最小汉明距离。在一组字中，最小汉明距离（minimum Hamming distance）是所有可能对中的最小汉明距离。我们以d_{min}定义编码方案中的最小汉明距离。为了得出这个值，我们导出所有字的汉明距离并选择最小值。

> 最小汉明距离是一组字中所有可能对的最小汉明距离。

例10.5

求表10.1中编码方案的最小汉明距离。

解：

我们先求出所有的汉明距离：

d（000，011）=2　　d（000，101）=2　　d（000，110）=2

d（011，101）=2　　d（011，110）=2　　d（101，110）=2

得出d_{min}是2。

例10.6

求表10.2中编码方案的最小汉明距离。

解：

我们先求出所有的汉明距离：

d（00000，01011）=3　　d（00000，10101）=3　　d（00000，11110）=4

d（01011，10101）=4　　d（01011，11110）=3　　d（10101，11110）=3

得出d_{min}是3。

三个参数

在继续讨论前，我们需要注意任何编码方案需要至少三个参数：码字长度n，数据字长度k，以及最小汉明距离d_{min}。编码方案C写成$C(n, k)$和一个单独的d_{min}表达式。例如，我们可以把第一个编码方案写作$C(3, 2)$和$d_{min}=2$，把第二个写作$C(5, 2)$和$d_{min}=3$。

汉明距离和差错

在我们探究检错和纠错的标准前，让我们讨论汉明距离与传输中发生的差错之间的联系。当一个码字在传输中被破坏时，发送的码字和接收到的码字之间的汉明距离是差错影响的位数。换言之，接收到的码字和发送的码字之间的汉明距离是传输中破坏的位数。例如，如果发送的码字是00000而接收到的码字是01101，3个位发生差错，这两个码字间的汉明距离是d(00000，01101)=3。

检错的最小距离

如果我们能够检测出最多s个差错，那么让我们导出编码中的最小汉明距离。如果s个差错发生在传输中，发送的码字和接收到的码字间的汉明距离是s。如果我们的编码能检测出最多s个差错，那么两个有效编码间的最小汉明距离必须是$s+1$，这样接收到的码字才不会与有效码字匹配。换言之，如果所有有效码字间的最小距离是$s+1$，那么接收到的码字不会被错误地

认为是另一个正确的码字。距离小于（$s+1$）码字接收方不会认为是有效码字。差错可以被检测到。我们需要在这里澄清一点：虽然$d_{min}=s+1$的编码可能能够在某些特殊情况下检测出多于s个差错，但是只有s个或更少的差错可以保证被检测到。

为了保证检测出所有情况下最多s个差错，块编码中最小汉明距离一定是$d_{min}=s+1$。

例10.7

我们第一个编码方案（表10.1）的最小汉明距离是2。这个编码方案保证检测到单个差错。例如，如果发送第二个码字（101），发生了一个差错，那么接收到的码字就不能与任何一个有效码字匹配。但是如果发生了两个差错，接收到的码字可能与其一个有效码字匹配，因此差错无法被检测到。

例10.8

第二个编码方案（表10.2）有$d_{min}=3$。这个编码能检测到最多2个差错。可以看到，当发送任一个有效码字时，发生两个差错得到的码字不在有效码字表中。接收方不会被欺骗。但是，三个差错的一些组合会把一个有效码字改变成另一个有效码字。接收方会接收接收到的码字，而无法检测到差错。

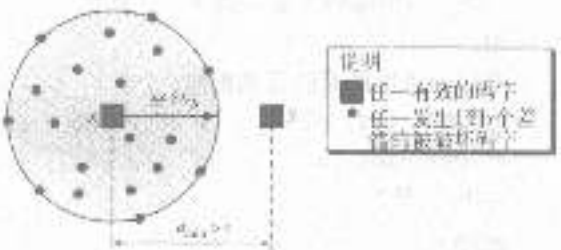


图10.8　每出差错检测中d_{min}的几何意义

我们以几何意义看待这个问题。假设发送的码字x在以s为半径的圆中心。所有其他接收到的发生1到s个差错的码字位于圆内或者圆周上。所有其余有效码字一定位于圆外，如图10.8所示。

在图10.8中，d_{min}一定是大于s的整数，即$d_{min}=s+1$。

纠错的最小距离

纠错比检错更复杂。它涉及一个决策。当接收到的码字不是有效的码字时，接收方需要确定实际发送的是哪一个码字。这个决策基于区域概念，围绕码字的独占区域。每个有效码字都有自己的区域。

我们使用几何方法定义每个区域。假设每个有效码字有以t为半径的圆形区域并且该有效码字在圆心。例如，假设码字x被破坏的位小于或等于t位，那么这个被破坏码字位于圆内或圆周上。如果接收方接收到属于该区域的一个码字，就确定原来的码字就是位于圆心的码字。注意：假设最多只能发生t个差错，否则决策是错误的。图10.9说明了这个几何解释。一些书使用球体来说明所有有效块编码间的距离。

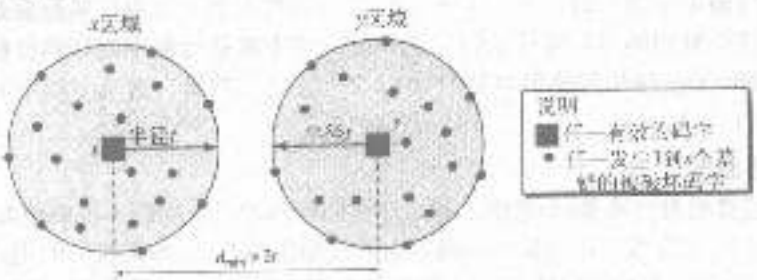


图10.9　差错纠正中d_{min}的几何意义

在图10.9中，$d_{min}>2t$；因为下一个整数增量是1，所以可以说$d_{min}=2t+1$。

> 为了保证纠正所有情况下最多t个差错，块编码中的最小汉明距离是$d_{min}=2t+1$。

例10.9

一个编码方案有汉明距离$d_{min}=4$。这个方案的检错和纠错能力是多少？

解：

这个方案保证检测到最多3个差错（s=3），但是它能纠正1个差错，换言之，如果这个编码用于纠错，它的部分能力被浪费了。纠错编码需要的最小距离是奇数（3、5、7、…）。

10.3 线性块编码

目前，几乎所有使用的块编码都属于一个称为**线性块编码**（linear block code）的子集。用于检错和纠错的非线性块编码的应用不是很广泛，因为它们的结构使得理论分析和实现都很困难；因此，我们只介绍线性块编码。

线性块编码的正式定义需要抽象代数（尤其是Galois域）理论，这超出了本书范围。所以我们给出一个非正式定义。为此目的，线性块编码是一种由任何两个有效码字的异或（模2加法）产生的另一个有效码字。

> 在线性块编码中，任两个有效码字的异或（XOR）生成另一个有效码字。

例10.10

让我们观察定义在表10.1和表10.2中的两个编码是否属于线性块编码。

1. 表10.1中的方案是线性块编码，因为任何一个码字和任何另一个码字XOR的结果是一个有效码字。例如，第2个和第3个码字的XOR得到第4个码字。

2. 表10.2中的方案也是线性块编码。我们可以通过异或两个其他码字得到所有的4个码字。

10.3.1 线性块编码的最小距离

可以简单地求出线性块编码的最小汉明距离。最小汉明距离是具有最小1的个数的非0有效码字中1的个数。

例10.11

在第一个编码方案（表10.1）中，非零码字的1的个数是2、2和2。因此，最小汉明距离是d_{min}=2。在第二个编码方案（表10.2）中，非零码字的1个数是3、3和4。因此d_{min}=3。

10.3.2 一些线性块编码

现在给出一些线性块编码。这些代码是线性块编码不证自明，因为我们可以轻易地得到代码和译码算法并检验它们的性能。

简单奇偶校验编码

可能最熟悉的差错检测编码是**简单奇偶校验编码**（simple parity-check code）。在这个编码中，k位数据字变成n位码字，这里$n=k+1$。选择称为奇偶位的额外位，使得码字中1的总个数为偶数。虽然另一些实现规定1的个数为奇数，但是我们讨论偶数情况。这种编码的最小汉明距离是$d_{min}=2$，这意味着编码是单个位检错编码，它不能纠正任何差错。

> 简单奇偶校验编码是$n=k+1$、$d_{min}=2$的单个位检错编码。

我们的第一个编码（表10.1）是k=2，n=3的奇偶校验编码。表10.3中的编码也是奇偶校验编码，这里k=4，n=5。

图10.10说明（在发送方）编码器和（在接收方）译码器的可能结构。编码器使用一个生成器，该生成器得到一个4位数据字的副本（a_0、a_1、a_2和a_3）并产生一个奇偶位r_0。数据字的各个位和奇偶位（parity bit）组成一个5位码字。增加的奇偶位使得码字中的1的个数为偶数。

表10.3　简单奇偶校验码C (5, 4)

数据字	码字	数据字	码字
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

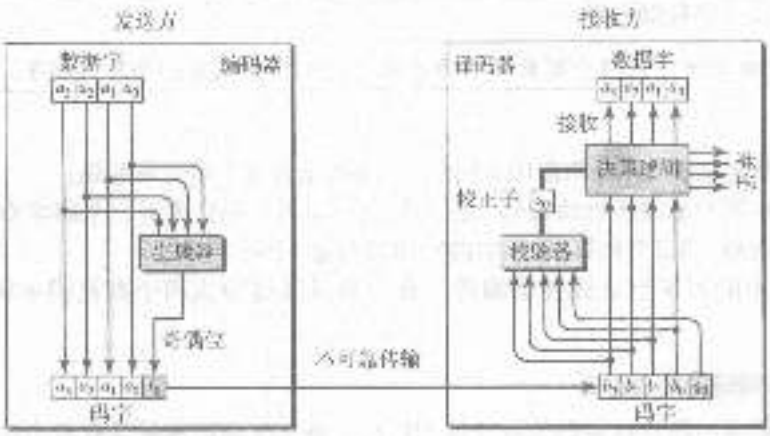


图10.10　简单奇偶校验编码的编码器和译码器解码器

通常这是通过数据字的4位做加法（模2运算）实现，结果是奇偶位的值。换言之，

$$r_0 = a_3 + a_2 + a_1 + a_0 \quad \text{（模2运算）}$$

如果1的个数是偶数，结果是0；如果1的个数是奇数，结果是1。在这两种情况中，码字中1的总个数是偶数。

发送方发送一个码字，该码字可能在传输中被破坏。接收方接收到一个5位码字。接收方的校验器将做与发送方的生成器相同的工作，不同的是所有的5位都要做加法。称为校正子（syndrome），只有1位。当接收到的码字中1的个数是偶数时，校正子是0，否则它是1。

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \quad \text{（模2运算）}$$

将校正子传送给决策逻辑分析器。如果校正子是0，接收到的码字中没有差错；其中的数据部分作为数据字接收。如果校正子是1，接收到的码字中的数据部分被丢弃。没有数据字生成。

例10.12

让我们看一些传输实况。假设发送方发送数据字1011。从这个数据字生成的码字是10111，

它发送给接收方。我们检查五种情况：

1. 没有差错发生。接收到的码字是10111。校正子是0，生成数据字1011。

2. 一个单个位差错改变了a_1。接收到的码字是10011，校正子是1，没有数据字生成。

3. 一个单个位差错改变了r_3。接收到的码字是10110，校正子是1，没有数据字生成。注意：虽然没有数据字位被破坏，但是因为编码不够复杂不能说明破坏位的位置，因此仍然没有数据字生成。

4. 一个差错改变了r_2，第二个差错改变了a_3，接收到的码字是00110。校正子是0，在接收方生成数据字0011。注意：这里因为校正子生成错误的数据字，简单奇偶校验译码器不能检测出偶数个差错。差错互相抵消使得校正子为0。

5. 三个位a_1，a_2和a_3被差错改变。接收到的码字是01011。校正子是1，没有数据字生成。这说明简单奇偶校验除了保证检测出一个单个位差错外，还能发现任何奇数个差错。

> 简单奇偶校验编码能检测出奇数个差错。

更好的办法是**两维奇偶校验**（two-dimensional parity check）。在这一方法中，数据字以表格形式（行和列）组织。在图10.11中，要发送的数据（5个7位字节）分别放在各自单独的行中。对于每一行和每一列，计算出一个奇偶校验位，然后将整个表发送给接收方。接收方将分别得出每一行和每一列的校正子。正如图10.11所示，两维奇偶校验能检测出表中任何位置发生的最多三个差错（箭头指向生成的非零校正子位置）。但是，4位的差错无法检测到。

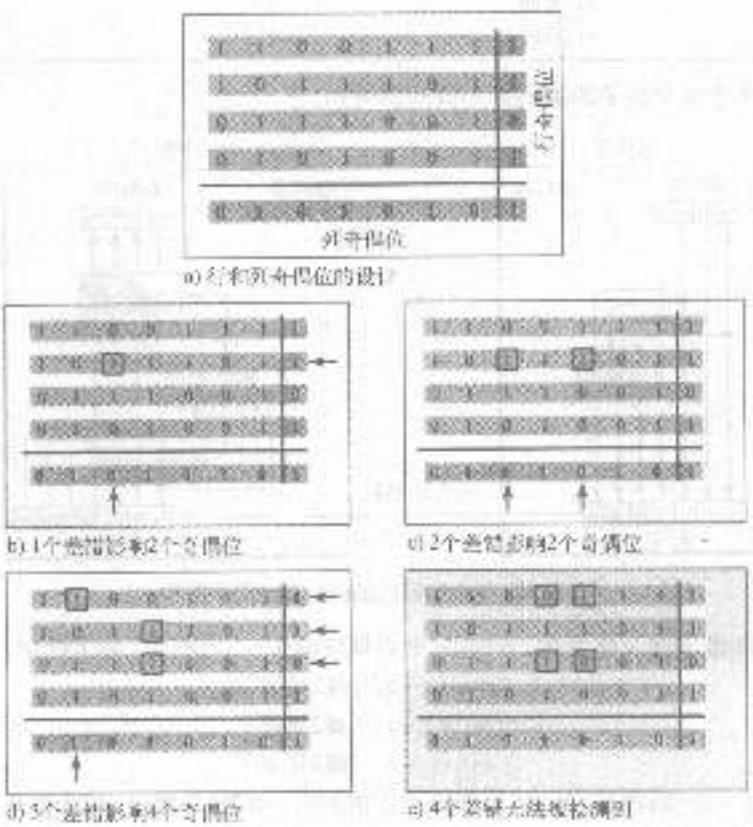


图10.11 两维奇偶校验编码

汉明编码

现在让我们讨论称为汉明编码（Hamming code）的纠错编码。这些编码最初设计是$d_{min}=3$，这意味着它们能做最多检测出2个差错和最多纠正1个差错。虽然有一些汉明编码能纠正多于1个差错，但是我们的讨论只限于单个位纠错编码。

让我们先找出汉明编码中n和k的关系。我们需要选择$m \geqslant 3$的整数。然后从m计算出n和k，分别为$n=2^m-1$和$k=n-m$。校验位个数$r=m$。

> 本书讨论的所有汉明编码都有$d_{min}=3$，这些编码中m和n之间的关系是$n=2^m-1$。

例如，如果$m=3$，那么$n=7$，$k=4$，这就是$d_{min}=3$的汉明编码$C(7,4)$。表10.4给出了这个编码的数据字和码字。

表10.4　汉明编码C (7, 4)

数据字	码字	数据字	码字
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

图10.12给出了这个例子的编码器和译码器结构。

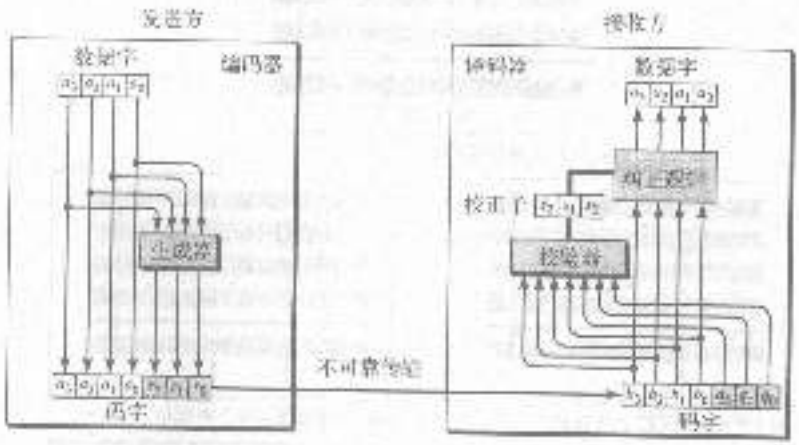


图10.12　汉明编码的编码器和译码器结构

4位数据字的副本进入生成器，产生三个奇偶校验位r_0、r_1和r_2，如下所示：

$$r_0=a_2+a_1+a_0 \quad (\text{模2运算})$$

$$r_1=a_3+a_2+a_1 \quad (\text{模2运算})$$

$$r_2=a_2+a_0+a_3 \quad (\text{模2运算})$$

换言之，每个奇偶校验位处理4位数据字中的3位。每个4位组合（3个数据字位和1个奇偶位）中的1的个数一定是偶数。三个等式不是唯一的。只要数据字4位中取3位生成独立等式（两个等式组合不能生成第三个等式）的任何三个等式是有效的。

$$c_0=b_3+b_2+b_1+q_0 \quad (\text{模2运算})$$
$$c_1=b_3+b_2+b_0+q_1 \quad (\text{模2运算})$$
$$c_2=b_3+b_1+b_0+q_2 \quad (\text{模2运算})$$

校验器使用的等式与生成器使用的一样，奇偶校验位加到等式的右边。3位校正子产生8个不同的位模式（000到111），这能表示8种不同的情况。这些情况定义了接收到的码字中无差错或者7位中有1位有差错。如表10.5所示。

表10.5 由译码器的纠错逻辑分析器做出的逻辑决定

校正子	000	001	010	011	100	101	110	111
差错	无	q_0	q_1	b_3	q_2	b_1	b_2	b_0

注意：生成器不关心表10.5阴影中的四种情况（无差错或奇偶位的差错）。在其他四种情况中，4位中相应的位必须反转（0变成1或1变成0）得出正确的码字。

表10.5中的校正子的几个值是每个校正子值的组合。例如，如果q_0发生差错，s_0是唯一受到影响的位，因此校正子是001。如果b_2发生差错，s_1和s_0都受到影响，因此校正子是011。类似地，如果b_1发生差错，所有三个校正子位都受到影响，因此校正子是111。

有两点需要在这里强调一下。第一点，如果在传输中发生两个差错，生成的数据字可能不是正确的数据字。第二点，如果想用上面的编码用于纠错，还需要独特的设计。

例10.13

让我们跟踪3个数据字从发送方到目的地的路径。

1. 数据字0100变成码字0100011。接收到码字0100011。校正子是000（无差错），最后的数据字是0100。

2. 数据字0111变成码字0111001。接收到码字0011001。校正子是011。根据表10.5，b_2发生差错。反转b_2后（0变成1），最后的数据字是0111。

3. 数据字1101变成码字1101000。接收到码字0001000（两个差错）。校正子是101，这意味着b_0发生差错。反转b_0后，我们得到0000。这是错误的数据字。这说明我们的编码检测不到两个差错。

例10.14

我们需要至少7位的数据字。计算满足这个条件的k值和n值。

解：

我们要求$k=n-m$大于或等于7，或者$2^m-1-m\geq 7$。

1. 如果我们设$m=3$，结果是$n=2^3-1$，$k=7-3=4$，这是不能接受的。

2. 如果我们设$m=4$，那么$2^4-1=15$，$k=15-4=11$，这满足了条件。因此编码是$C(15,11)$。有一些方法使得数据字为特定长度，但是这个问题的讨论和实现超出了本书的范围。

性能

汉明编码只能纠正单个差错或检测出两个差错。但是，有方法使它能检测出突发差错，如图10.13所示。

关键思想是分离若干个码字间的突发性差错为每个码字只有一个差错。在数据通信中，我们通常发送数据分组或数据帧。为了让汉明编码能对长度为N的突发差错起作用，需要从帧生成N个码字。然后，不是每次发送一个码字，而是以表格形式组织码字，并每次发送表中一列的位。在图10.13中，以逐列发送位（从左边开始）。在每一列中，从底到顶发送位。在这种方式中，一个帧由4个码字组成，并发送给接收方，图10.13说明长度为4的突发差错破坏了

这个帧，每个码字只有1位被破坏。每个码字被破坏的位在接收方可以轻易地被纠正。

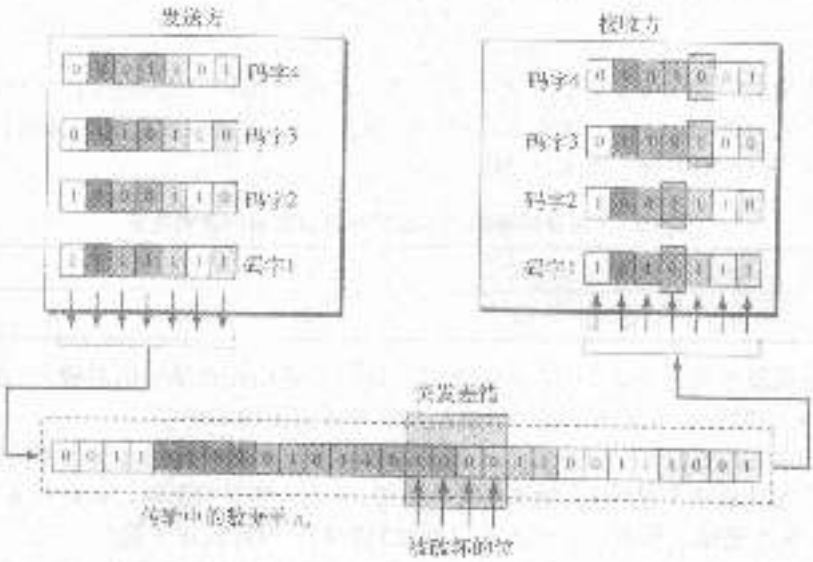


图10.13　使用汉明码的突发差错纠正

10.4　循环编码

循环编码是有一个附加性质的特殊线性块编码。在**循环编码**（cyclic code）中，如果码字是循环移位（旋转），则其结果是另一个码字。例如，如果1011000是一个码字，循环左移，那么0110001也是一个码字。在这种情况中，如果称第一个码字中的位为a_0到a_6，那么称第二个码字中的位为b_0到b_6，我们可以通过如下方式进行移位：

$$b_1 = a_0 \quad b_2 = a_1 \quad b_3 = a_2 \quad b_4 = a_3 \quad b_5 = a_4 \quad b_6 = a_5 \quad b_0 = a_6$$

在最右的等式中，第一个码字中的最后一位变成了第二个码字中的第一位。

10.4.1　循环冗余校验

我们能够生成循环编码来纠错。但是，所需的理论背景超出了本书的范围。在本节中，我们简单地讨论一种称为**循环冗余校验**（cyclic redundancy check，CRC）的循环编码，它用于诸如LAN和WAN的网络中。图10.14说明了编码器和译码器的一种可能设计。

表10.6给出了CRC编码的一个例子，可以看到这个编码具有的线性特性和循环特性。

表10.6　C (7, 4)的CRC编码

数据字	码字	数据字	码字
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

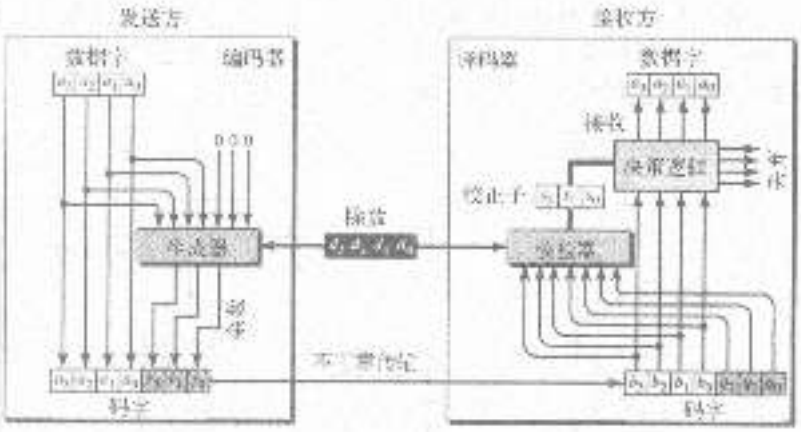


图10.14 说明了编码器和译码器的一种可能设计

左编码器中，数据字有k位（这里k是4），码字有n位（这里n是7）。在数据字的右边加上n−k（这里是3）个位，长度加3，n位结果传给生成器。生成器使用长度为n−k+1（这里是4）的除数，这个除数是预定义并经双方同意的。生成器用除数除增加后的数据字（模2除法）。除法的商被丢弃，余数（$r_2r_1r_0$）加到数据字上生成码字。

译码器接收到可能被破坏的码字，全n位的副本传送给校验器，它是生成器的复制品。校验器产生的余数是n−k（这里是3）位的校正子，它被传递给决策逻辑分析器。分析器是一个简单的函数。如果校正子全0，码字最左边的4位被接收为数据字（被认为无差错）；否则，这4位被丢弃（有差错）。

编码器

让我们仔细考察编码器。编码器将得到的数据字加上n−k个0，然后它被除数去除，如图10.15所示。

模2二进制除法过程与我们熟悉的用于十进制数字的除法过程是一样的，但是，正如在本章开始所提到的，此时加法和减法是一样的。我们使用异或操作来做加法和减法。

像在十进制除法中一样，处理过程逐步完成。在每一步，除数的副本与被除数的4位进行异或操作。异或运算的结果（余数）是3位（在这种情况中），1个额外位移下来加上去组成4位进行下一步。在这种除法中有重要的一点需要记住：如果被除数（用于每一步的部分）的最左边的位是0，这一步就不能使用正常的除数，需要使用全0的除数。

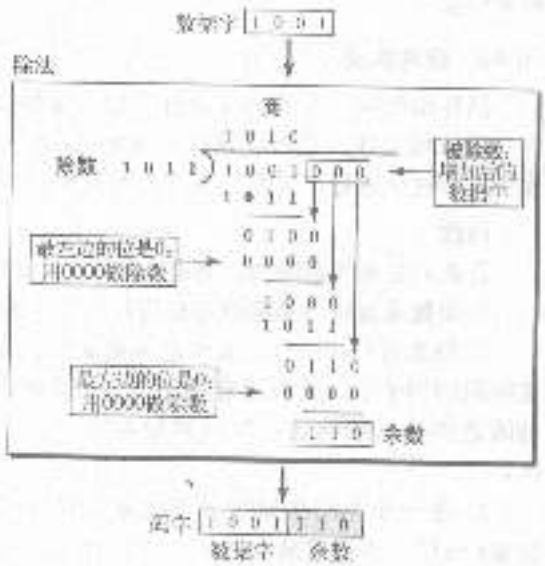


图10.15 CRC编码器中的除法

当没有可移下来的位时，就得到了结果。3位余数形成了校验位（r_2、r_1和r_0），它们附加在数据字上生成码字。

译码器

码字可能会在传输中改变。译码器执行与编码器相同的除法处理，除法的余数是校正子。

如果校正子是全0，就没有差错；数据字从接收到的码字中分离出来并被接收。否则，整个码字被丢弃。图10.16说明了两种情形：左边的图说明当没有差错发生时的校正子，其为000。右边的图说明有单个差错发生的情形，校正子不是全0（它是011）。

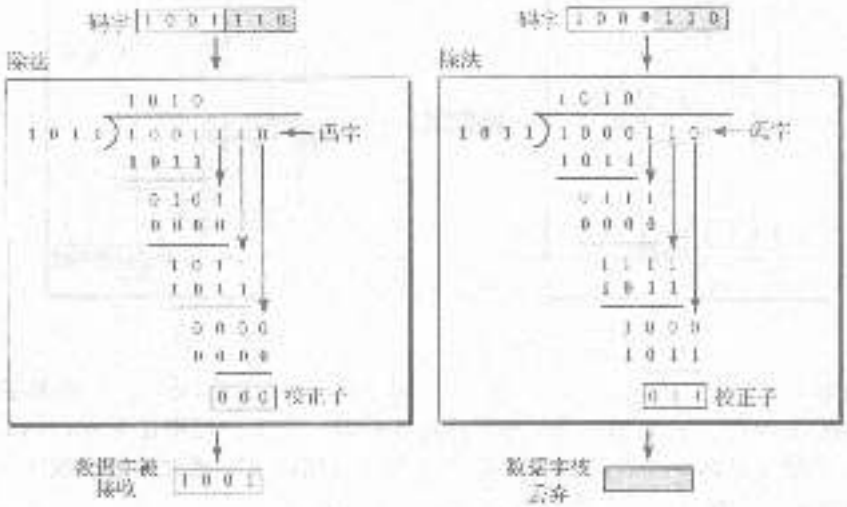


图10.16　两种情形下CRC译码器中的除法

除数

读者可能想知道如何选择除数1011。在本章后面我们会给出一些标准，但是通常它涉及抽象代数。

10.4.2　硬件实现

循环编码的一个优点是通过使用少量的电子设备，编码器和译码器就可以用硬件轻易地并且便宜地实现。还有，硬件实现增加了校验位和校正子位计算的比率。在本节，我们尝试逐步说明这个过程。但是，本节是可选的，不会影响对本书其他部分的理解。

除数

让我们先来考虑除数。需要注意如下几点：

1. 除数重复地与被除数部分进行异或运算；

2. 除数有$n-k+1$位，或者是预定义的，或者全为0。换言之，这些位不会随数据字改变。在前面的例子中，除数或者是1011或者是0000。具体选哪一个取决于异或运算中起作用的增加的数据位（被除数）部分的最左边一位；

3. 进一步说明在异或运算中除数只需要$n-k$位。不需要最左边的一位，因为无论这个位的值是什么运算，其结果永远是0。原因是异或操作的输入或者都是0或者都是1。在前面的例子中，只有3位（而不是4位）实际用于异或运算中。

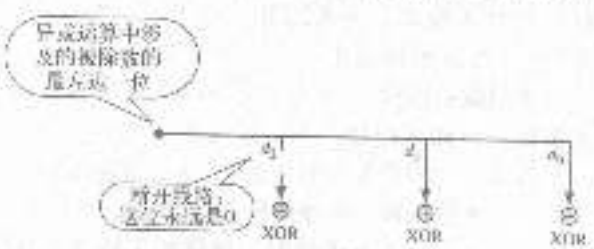


图10.17　CRC中除数的硬件设计

根据这几点，如果我们知道除数的模式，就能生成可用于循环编码的固定（硬件线路的）除数。图10.17给出了这样一个用于前面例子的设计，还显示了用于运算的异或设备。

注意：如果用于这一步的被除数部分的最左边一位是1，除数位（$d_2 d_1 d_0$）是011，如果最

左边一位是0，则除数仍是000，这个设计根据最左边位做出正确的选择。

增加后的数据字

在图10.15用纸笔进行的除法运算过程，我们表示增加后的数据字的位置不变，而将除数向右位移，每一步移1位。除数各位与增加后的数据字各位正好对齐。虽然除数是固定的，那么我们需要把增加后的数据字向左位移（反方向），使得除数的各位正好对齐。不需要保存增加后的数据字的各位。

余数

在前面的例子中，余数的长度是3位（一般n-1位）。可以使用3个寄存器（单个位存储设备）来保存这些位。为了得到除法的最终余数，需要修改除法过程。下面是以硬件（或者甚至以软件）模拟除法的过程。

1. 假设余数初始为全0（我们的例子中是000）。
2. 在每个时间节拍（增加后的数据字每到来1位），重复以下两个处理过程：
   a. 我们使用最左边的位来决定除数（011或000）。
   b. 余数的其他2位和增加后数据字的下一位（一共3位）与3位除数进行异或运算得出下一个余数。

图10.18表示了这个模拟过程。但是注意，这不是最终设计，最终设计将会有更大的改进。

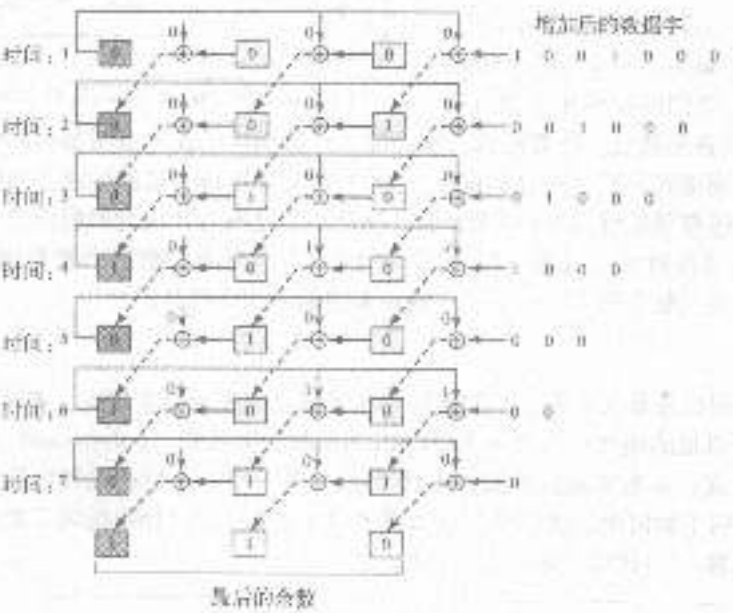


图10.18 CRC编码器中除法模拟

每个时间节拍用不同的时间表示，增加后数据字的一位用于异或运算。如果仔细观察这个设计有7个步骤，而用纸笔运算时只有4个步骤。增加了前3个步骤是为了使得每个步骤一样以及每个步骤的设计一样。步骤1、2和3把前3位送进余数寄存器；步骤4、5、6和7与纸笔运算相匹配，注意步骤4到7的余数寄存器中的值正好与纸笔运算中的值相匹配。最后的余数也一样。

上面的设计只是为了演示，它需要简化才能用于实际。首先，不需要保持余数各位的中间值，只需要最后的值。因此，只需要3个寄存器而不是24个。在异或运算后，不需要前一个

余数的各位值。还有，不需要21个异或设备，两个就足够了，因为如果一个位是0，异或运算的输出就是另一个位的值。另一个位就可以无干输出。使用这两个修改，设计变得极其简单也更便宜，如图10.19所示。

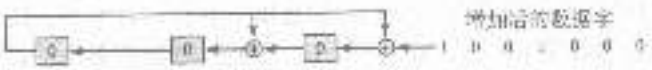


图10.19　使用移位寄存器的CRC编码器设计

但是，需要把寄存器改成移位寄存器，在一个时钟节拍期间，1位移位寄存器保存1位。在每个时间节拍点，移位寄存器从输入端口接收一个位，存储新的位并显示在输出端口。直到下一个输入到来其内容和输出保持一致。当把若干个1位移位寄存器连接在一起，它看起来好像寄存器的内容在移位。

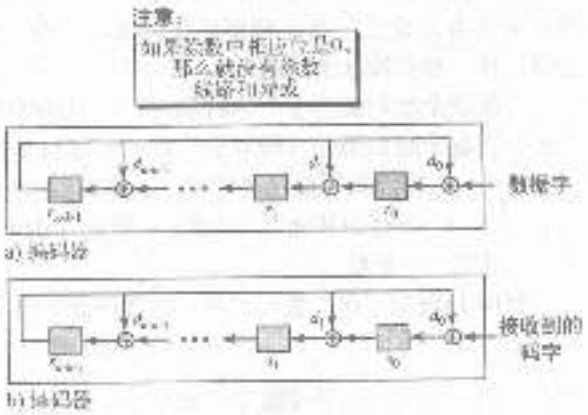


图10.20　CRC编码中编码器和译码器的一般设计

一般设计

一个编码器和译码器的一般设计如图10.20所示。

注意：在编码器和译码器中有$n-k$个1位移位寄存器，最多有$n-k$个异或设备，但是除数通常在它们的模式中有若干个0。这可以减少设备的数量。还要注意：作为输入是数据字，而不是增加后的数据字，因为数据字中的位输入到编码器后，附加的位（全为0）不会影响最右边的异或。当然，在生成校验位前，这个处理还需要继续另外$n-k$个步骤。这个事实是对这个设计的批评之一，更好的方案设计能消除这个等待时间（校验位在k步后就生成），但是我们把这个留作读者的研究课题。然而，在译码器中，整个码字在校正子生成前必须输入到译码器。

10.4.3　多项式

理解循环编码以及如何分析它们的更好办法是将它们表示为多项式，本节也是可选的。

若干个0和1组成的模式可以表示为以0和1为系数的多项式（polynomial）。每一项的幂次表示位所在的位置，系数表示位的值。图10.21给出了一个二进制模式和它的多项式表示。在图10.21a中，说明了如何把二进制模式翻译成多项式；在图10.21b中说明了如何通过除去系数为0的项并用x代替x^1，1代替x^0来简化多项式。

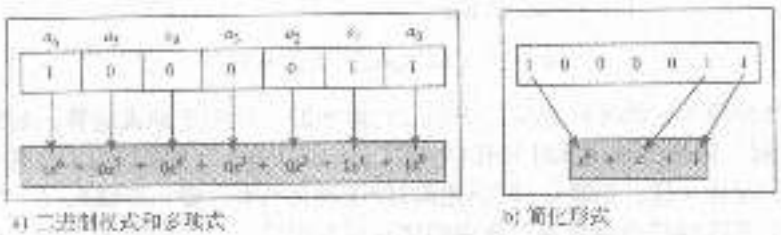


图10.21　用多项式表示二进制串

图10.21说明了一个显而易见的好处，7位模式可以用3项表示。当遇到诸如$x^{23}+x^3+1$的多项式时，好处更加突出。这里模式是24位长（3个1和21个0），而用多项式表示时只有3项。

多项式的次数

多项式的次数是多项式中的最高幂次。例如，多项式x^6+x^2+1的次数是6。注意多项式的次数比二进制模式中位数少1。这种情况下位模式有7位。

多项式加减法

数学上的多项式加减是通过加减相同幂次项的系数。此时系数只有0和1，加是模2加法。有两个结论。第一，加和减相同。第二，加减是删除相同项把不同项合在一起。例如$x^5+x^2+x^1$和$x^4+x^3+x^2$的结果是$x^5+x^4+x^3$。x^2和x^2被删除。然而，注意：如果3个多项式相加，有3个项x^2，就删除其中2个保留1个。

项乘法和除法

在这个算术中，一个项乘以另一个项很简单，只是幂次相加。例如$x^2\times x^3$是x^5。对于除法，只需第一个幂次减去第二个幂次即可。例如x^7/x^4是x^3。

两个多项式相乘

一个多项式乘以另一个多项式通过项乘项完成。第一个多项式的每一个项必须乘以第二个多项式的所有项。然后相同项对被删除化。简其结果。一个实例如下：

$$(x^7+x^5+x^2+x^1)(x^2+x+1)=x^9+x^8+x^7+x^7+x^6+x^3+x^4+x^3+x^2+x^3+x^2+x=x^9+x^6+x^3+x$$

一个多项式除另一个多项式

从概念上来说，多项式除法与编码器中讨论的二进制除法相同。被除数位的第一项除以除数的第一项得到商的第一项。然后，将这一项乘以除数的结果从被除数中减去，重复这个过程直到被除数次数小于除数为止。我们会在本章稍后给出一个除法例子。

移位

二进制模式时常左移或右移若干位。左移表示在右边加上若干个额外0；右移表示删除右边若干位。左移通过每一项乘以x^m实现，这里m是移动的位数；右移通过每一项除以x^m实现。下面说明了左移和右移。注意在多项式中没有负幂次。

左移3位：10011变成10011000　x^4+x+1变成$x^7+x^6+x^3$

右移3位：10011变成10　x^4+x+1变成x

在图10.15编码器中的增加后数据字实际上是左移了若干位。还要注意，连接两个位模式就是左移第一个多项式再加上第二个多项式。

使用多项式的循环编码编码器

已经讨论过了多项式运算，现在说明如何从数据字生成码字。图10.22是对图10.15的多项式的表示形式。可以看到这个过程更短。数据字1001表示为x^3+1。除数1011表示为x^3+x+1。为了得到增加后的数据字，先把数据字左移3位（乘以x^3），其结果是x^6+x^3。除法是直接的，用被除数的第一项x^6先除以除数的第一项x^3，商的第一项是$x^6/x^3=x^3$。然后，被除数减去x^3乘以除数（根据前面的减法定义）得到的结果。结果的第一项是x^5，它的次数大于除数的次数，继续除直到余数的次数小于除数的次数。

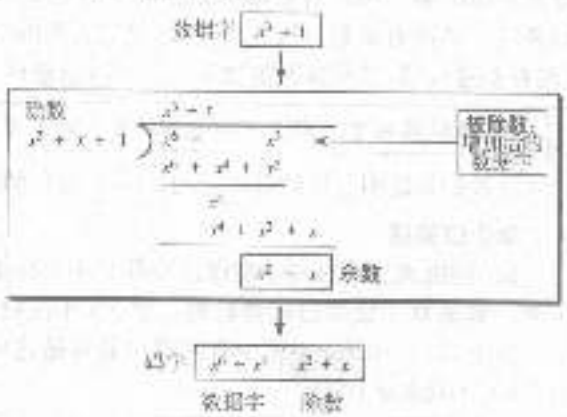


图10.22 使用二项式的CRC除法

可以看到在这个例子中，多项式表示简化了除法运算，因为这里不再需要涉及全0除数的两个步骤。（当然，人们可以将二进制除法中全0除数的步骤也去掉。）在多项式表示中，除数一般指生成多项式d(x)。

循环编码中的除数通常称为生成多项式或简称为生成器。

10.4.4 循环编码分析

我们可以使用多项式来分析循环编码的能力。我们定义下面多项式，这里f(x)是二进制系数的多项式。

数据字：$d(x)$　　码字：$c(x)$　　生成多项式：$g(x)$

校正子：$s(x)$　　差错：$e(x)$

如果$s(x)$不为0，那么一位或多位被破坏。但是，如果$s(x)$为0，或者没有位被破坏或者译码器无法检测到任何差错。

在循环编码中，

1. 如果$s(x) \neq 0$，一位或多为被破坏。
2. 如果$s(x) = 0$，
   a. 没有位被破坏，或者
   b. 有一些位被破坏，但是译码器无法检测到。

在分析中，要找到生成多项式$g(x)$使得能检测出那些想检测出的差错类型的标准。让我们先找到发送的码字、差错、接收到的码字和生成多项式之间的关系。可以说

$$\text{接收到的码字} = c(x) + e(x)$$

换言之，接收到的码字是发送的码字和差错之和。接收方用$g(x)$除接收到的码字得到校正子。可以写成

$$\frac{\text{接收到的码字}}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

等式右边的第一项没有余数（根据码字的定义）。因此，校正子实际上是右边第二项的余数。如果这一项没有余数（校正子=0），即$e(x)$为0或者$e(x)$可以被$g(x)$整除。不必担心第一种情况（没有差错），第二种情况很重要。那些可以被$g(x)$整除的差错无法被捕捉到。

在循环编码中，那些可以被$g(x)$整除的差错无法被捕捉到。

让我们先说明一些特殊的差错，看看它们如何被设计好的$g(x)$捕捉到。

单个位差错

$g(x)$的组成应该怎样才能保证检测到单个位差错？单个位差错是$e(x)=x^i$，这里i是差错位的位置。如果单个位差错被捕捉到，那么x^i不能被$g(x)$整除。（注意：当我们说不能整除时，表示它有余数。）如果$g(x)$至少有两项（通常是这样的）并且x^0的系数不为0（最右边的位为1），那么$e(x)$不能被$g(x)$整除。

如果生成多项式至少有两项并且x^0的系数为1，那么所有单个位差错都可以被捕捉到。

例10.15

如下那个$g(x)$可以保证捕捉到单个位差错？对于每种情况，不能捕捉的差错是什么？

a. $x+1$

b. x^3

c. 1

解：

a. 没有一个x^i可以被$x+1$整除。换言之，$x^i/(x+1)$总有余数。因此，校正子不是零。任何单个位差错都可以被捕捉到。

b. 如果i大于或等于3，x^i可以被$g(x)$整除。x^i/x^3的余数是0，接收方便会被蒙蔽而认为没有差错，虽然实际是一个差错。注意：这种情况下，被破坏的位必须位于第4位或更高位。能捕捉到位1到位3的单个差错。

c. i的所有值使得x^i可以被$g(x)$整除。没有单个位差错可以被捕捉到。另外，$g(x)$是无用的，因为它表示的只是增加了$n-k$个0后的数据字的码字。

两个独立的单个位差错

现在想像有两个独立的单个位差错。什么情况下这种类型的差错可以被捕捉到？可以把这种差错表示为x^j+x^i。i和j的值只表示差错的位置，差$j-i$表示两个差错的距离，如图10.23所示。

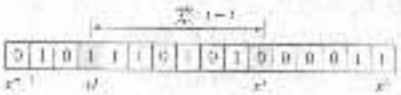


图10.23 使用多项式表示的两个独立的单个位差错

可以写成$e(x)=x^i(x^{j-i}+1)$。如果$g(x)$多于一项并且其中一项为x^0，那么正如在前面看到的，它不能整除x^i。因此，如果$g(x)$能整除$e(x)$，那么它一定能整除$x^{j-i}+1$。换言之，$g(x)$一定不能整除x^t+1，这里t在0和$n-1$之间。t是$t=0$是无意义的，并且稍后会看到需要$t=1$。这就表示t应该在2和$n-1$之间。

> 如果生成多项式不能整除x^t+1（t在0和$n-1$之间），那么所有独立的双差错都能被检测到。

例10.16

求出以下与两个独立的单个位差错相关的生成多项式的情况。

a. $x+1$

b. x^4+1

c. x^5+x^0+1

d. $x^{15}+x^{14}+1$

解：

a. 对生成多项式来说，这是一个很差的选择。任何两个相邻的差错都不能被检测到。

b. 这个生成多项式不能检测到相隔4个位置的两个差错。这两个差错能位于任何位置，但是如果它们的距离是4仍然无法被检测到。

c. 这是个好的选择。

d. 如果t小于32 768，这个多项式不能整除类型为x^t+1的任何差错。这表示一个码字中两个独立的差错相邻或者最多离开32 768位都能被这个生成多项式检测到。

奇数个差错

有因子$x+1$的生成多项式可以捕捉所有奇数个差错。这表示需要把$x+1$作为任何生成多项式的因子。注意：不是说生成多项式本身应该是$x+1$，而是说它应该有因子$x+1$。如果它只是$x+1$，那么它不能捕捉到两个相邻的独立差错（见前一节）。例如x^4+x^3+1能捕捉到所有奇数个差错，因为它可写作两个多项式$x+1$和x^3+x^2+1的乘积。

包含因子x+1的生成多项式能检测到所有奇数个差错。

突发性差错

现在把分析扩展到突发性差错，它是所有差错中最重要的。一个突发性差错的形式是$e(x)=(x^i+\cdots+x^j)$。注意：突发性差错和两个独立的单个位差错的不同。第一种有两项或更多项；而第二种只有两项。我们可以分解出因子x^j把差错$e(x)$写成$x^j(x^{i-j}+\cdots+1)$。如果生成多项式能检测到单个差错（对于生成多项式的最小条件），那么它不能整除x^j。我们应该关注的是能整除$x^{i-j}+\cdots+1$的那些生成多项式。换言之，$(x^{i-j}+\cdots+1)/(x^r+\cdots+1)$的余数一定不是0。注意分母是生成多项式，可能有二种情况：

1. 如果$i-j<r$，余数永远不会是0。我们可以写成$i-j=L-1$，这里L是差错的长度。因此$L-1<r$，$L<r+1$或$L\leq r$。这表示所有长度小于或等于校验位数r的突发性差错都会被检测到。

2. 在一些少见的情况中，如果$i-j=r$或$L=r+1$，校正子是0，差错无法被检测到。这可以在这些情况中得到验证，长度是$r+1$的突发性差错无法被检测到的概率是$(1/2)^{r-1}$。例如，如果我们的生成多项式是$x^{14}+x^3+1$，这里r是14，那么长度$L=15$的突发性差错概率是$(1/2)^{14-1}$，即几乎10 000中有1个无法被检测到。

3. 在一些少见的情况中，如果$i-j<r$或$L>r+1$，校正子是0，差错无法被检测到。这可以在这些情况中得到验证，长度大于$r+1$的突发性差错无法被检测到的概率是$(1/2)^r$。例如，如果我们的生成多项式是$x^{14}+x^3+1$，这里r是14，那么长度$L=15$的突发性差错概率是$(1/2)^{14}$，即几乎16 000中有1个无法被检测到。

- 所有$L<r$的突发性差错都会被检测到。
- 所有$L=r+1$的突发性差错有$1-(1/2)^r$的概率会被检测到。
- 所有$L>r+1$的突发性差错有$1-(1/2)^r$的概率会被检测到。

例10.17

分析出下面与不同长度突发性差错相关的生成多项式的适用性。

a. x^8+1

b. $x^{18}+x^3+x+1$

c. $x^{32}+x^{23}+x^7+1$

解：

a. 这个生成多项式可以检测出所有长度小于或等于6位的突发性差错；长度为7的突发性差错无法被检测到概率是3%；长度大于或等于8的突发性差错无法被检测到概率是16%。

b. 这个生成多项式可以检测出所有长度小于或等于18位的突发性差错；长度为19的突发性差错无法被检测到概率是百万分之8；长度大于或等于20的突发性差错无法被检测到概率是百万分之4。

c. 这个生成多项式可以检测出所有长度小于或等于32位的突发性差错；长度为33的突发性差错无法被检测到概率是千万分之5；长度大于或等于34突发性差错无法被检测到概率是千万分之3。

总结

可以总结一下令人满意的生成多项式的标准：

高性能生成多项式需要有以下特性：

1. 它应该至少有两项。
2. x^0项的系数应该是1。
3. 它应该不能整除x^t+1（t是2到$n-1$之间）。
4. 它应该有因子$x+1$。

标准多项式

常用协议中为CRC产生的标准多项式如表10.7所示。

表10.7 标准生成多项式

名 称	多 项 式	应 用
CRC-8	x^8+x^2+x+1	ATM头部
CRC-10	$x^{10}+x^9+x^5+x^4+x^1+1$	ATM AAL
CRC-16	$x^{16}+x^{15}+x^2+1$	HDLC
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	LANs

10.4.5 循环编码的优点

我们已经看到循环编码在检测单个位差错、双差错、奇数个差错和突发性差错中有非常令人满意的性能。它们可以很容易地以软硬件实现。当以硬件实现时，速度尤其快。这使得循环编码成为许多网络的好选择。

10.4.6 其他循环编码

本节中讨论的循环编码很简单，校验位和校正子可以通过简单的代数计算得出。但是还有基于抽象代数的Galois域上更强功能的多项式，它们超出了本书的范围。这些编码中最令人感兴趣的是目前用于检错和纠错的Reed-Solomon编码（Reed-Solomon code）。

10.5 校验和

这里讨论的最后一个检错方法称为校验和（checksum），校验和在因特网中被许多不属于数据链路层的协议使用。但是，在这里简单地讨论它以使关于检错的讨论更加完整。像线性和循环编码一样，校验和也是基于冗余的概念。将会在下面的章节中看到，许多协议仍然使用校验和进行检错，虽然趋势是它将被CRC替代。这表示CRC还用于除了数据链路层之外的其他层。

10.5.1 概念

校验和的概念并不难。让我们用几个例子来解释它。

例10.18

假设我们要发送到目的地的数据是5个4位数字。除了发送这些数字外，我们还发送这些数字的和。例如，如果这组数字是（7，11，12，0，6），那么我们发送（7，11，12，0，6，36），这里36是原来数字的和。接收方将这5个数字相加并将相加的结果与这和比较。如果两者相同，接收方就认为没差错，接收这5个数字并丢弃和。否则，就认为有差错，不接收这个数据。

例10.19

如果我们发送和的负值（补数）——称为校验和，就会使接收方的工作更容易。在这个例子中，我们发送（7，11，12，0，6，−36）。接收方将所有接收到的数字（包括校验和）相加。如果结果是0，就认为没差错。否则就认为有差错。

10.5.2 反码

前面的例子中有一个主要的缺点，除了校验和以外，所有的数据都写做4位字（它们小于15）。一个解决办法是使用反码（one's complement）算法。在这个算法中，只使用n位表示0

到2^n-1的无符号数字[⊖]。如果这个数字多于n位，那么最左边的额外位要加到最右边的n位。在反码算法中，一个数的负数可用该数所有位取反（0变成1，1变成0）表示。这与从2^n-1减去这个数字一样。

例10.20

在反码运算中如何只用4位表示数字21？

解：

数字21的二进制表示是10101（需要5位）。可以把最左边的位加到最右边的4位上。0101+1=0110或6。

例10.21

在反码运算中如何只使用4位表示数字−6？

解：

在反码算法中，一个数的负数是将所有位取反。正6是0110，负6是1001。如果只考虑无符号数字，它是9。换言之，6的补数是9。在反码运算中，另一种求一个数的反码的方法是2^n-1减去这个数（这里是16−1）。

例10.22

用反码运算重做练习10.19。图10.24给出了发送方和接收方的处理过程。发送方先把校验和初始化为0，并将所有数据项与校验和（校验和被看做是一个数据项并用颜色显示）相加。其和是36，但是，36不能用4位表示。多出的2位与和相加得到约束的校验和是6。在图中，以二进制表示其详细过程。然后对和求反，结果是校验和为9（15−6）。现在，发送方发送6个数据项（包括校验和9）给接收方。接收方执行与发送方相同的处理。它先把所有数据项相加（包括校验和），结果是45。然后和变成15。对约束和求反变成0。因为校验和的值是0，这表示数据没有被破坏。接收方去掉校验和保存其他数据项。如果校验和不是0，整个分组就被丢弃。

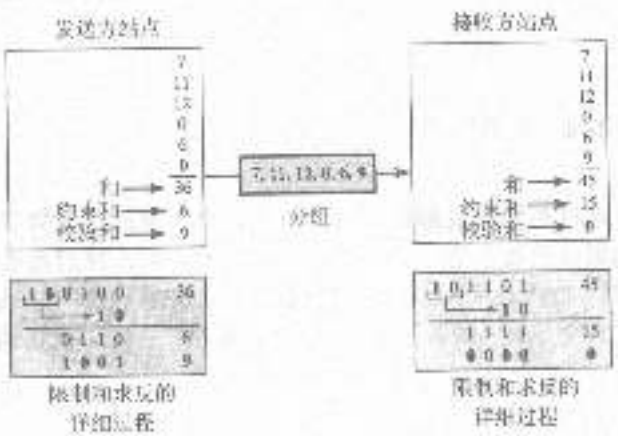


图 10.24

10.5.3 因特网校验和

传统上，因特网使用16位校验和。发送方通过下面的步骤计算校验和。

⊖ 虽然，一个数的反码可以同时代表正数和负数，但这里我们只关心无符号数。

发送方站点：

1. 报文被划分为16位字。
2. 校验和字的值设为0。
3. 所有字包括校验和使用反码运算相加。
4. 对这个和求反变成校验和。
5. 校验和随数据一起发送。

接收方使用如下步骤用于差错检测。

接收方站点：

1. 报文（包括校验和）被划分成16位字。
2. 用反码加法将所有字相加。
3. 对该和求反生成新的校验和。
4. 如果校验和的值是0，接收报文；否则就丢弃报文。

校验和的特性（把字看做是数字并相加后求反）完全适用于软件实现。可以写小程序在接收方站点计算校验和或检查报文的有效性。

例10.23

让我们计算8个字符（Forouzan）文本的校验和。该文本需要划分成2个字节（16位）的字。使用ASCII（见附录A）把每个字符变成2位十六进制数。例如，F表示成0x46，o表示成0x6F。图10.25说明了如何在发送方站点和接收方站点计算校验和。在图10.25a中，第一列部分的和是0x36，我们保持最右边的数字（6）并将最左边的数字（3）作为进位插入到第二列中。每一列重复这个过程。十六进制数见附录B。

注意：如果受到任何破坏，接收方重新计算的校验和将不全是0。把这个留到练习中。



a) 发送站点校验和　　b) 接收站点校验和

图 10.25

性能

传统的校验和使用较少的（16）位来检测任何长度（有时是数千位）报文中的差错。但是，在差错检测能力上它没有CRC强。例如，增加一个字的值而另一个字的以相同值减小，因为和以及校验和不变，所以这两个差错无法被检测到。还有，如果多个字的值增加，但是总的变化量是65535的倍数，则和以及校验和也不变，这表示差错也无法被检测到。Fletcher和Adler建议使用一些带权重的校验和，每个字乘以一个与它在文本中位置相关的一个数（称为它的权重）。这就消除了提到的第一个问题。但是当特网的趋势，尤其在新协议的设计中，是由CRC代替校验和。

推荐读物

为了深入讨论与本章有关的主题，我们推荐下列读物。在本书末列出括号[…]中的参考资料。

读物

许多优秀的书籍致力于差错编码。在这些书籍中我们推荐[Ham80]、[Zar02]、[Ror96]和[SWE04]。

请求评论

因特网中校验和应用的讨论可以在RFC1141中找到。

本章小结

- 传输期间数据可能被损坏，某些应用要求检错与纠错。
- 单个位差错是数据单元中仅有一位改变，突发性差错是数据单元中有两个或多个位改变。
- 为了检错和纠错，须与数据一起发送附加（冗余）位。
- 有两种纠错方法：向前纠错与重发纠错。
- 可将编码方案分为两大类：块编码与卷积编码。
- 在编码中，须采用模-2运算，这个运算非常简单：加法与减法是相同的，对加法与减法可用XOR（异或）运算。
- 在块编码中，将报文划分为块，每块k位，称为数据字。每块增加r个冗余位，使长度为$n=k+r$，形成n位的块称为码字。
- 在块编码中，用下列两个条件进行检错：
  a. 接收方有（或可找到）一张有效码字表；
  b. 原来的码字变成无效码字。
- 两个字的汉明距离是对应位不同的个数。一组字的最小汉明距离是该组中所有字对的最小汉明距离。
- 在任何情况下，为了保证能检测到s个差错，在块编码中的最小汉明距离必须是$d_{min}=s+1$。在任何情况下，为了保证能纠正t个差错，在块编码中的最小汉明距离必须是$d_{min}=2t+1$。
- 在线性块编码中，任何两个有效代码的异或（XOR）生成另一个有效代码。
- 在简单奇偶校验编码中$n=k+1$与$d_{min}=2$，它能检测代码的单个位差错。
- 本书讨论的汉明编码都是$d_{min}=3$。在这些编码中m与n之间的关系是$n=2m-1$。
- 循环编码是具有附加性质的特殊的线性块编码。在循环编码中，一个码字循环移位必是另一码字。
- 一类称为循环冗余校验（CRC）的循环编码用于局域网（LAN）和广域网（WAN）。
- 0和1的位组可表示系数为0和1的代数多项式。
- 传统上因特网一直使用16-位校验和，并使用反码运算。这种运算仅用n位表示0到2^n-1之间的无符号数。

习题

复习题

1. 单个位差错与突发性差错有什么不同？
2. 叙述检错与纠错中冗余的概念。

3. 区别向前纠错与重发纠错。
4. 线性块编码的定义是什么？循环编码的定义是什么？
5. 什么是汉明距离？什么是最小汉明距离？
6. 简单的奇偶校验与两维奇偶校验有什么关系？
7. 在CRC中，表示下列实体之间关系（长度是指位的个数）：
   a. 数据字长度与码字长度　　b. 除数长度与余数长度
   c. 生成多项式的次数与除数长度　　d. 生成多项式的次数与余数长度
8. 在校验和计算中，数据项相加采用哪一种算术运算？
9. 什么类型的差错用校验和不能检测出来？
10. 校验和的值是否可为全0（二进制）？说出理由。校验和的值是否可为全1（二进制）？说出理由。

练习题

11. 一个2ms的突发性噪声在下列数据传输速率时造成的最大影响是什么？
   a. 1 500bps　　b. 12kbps
   c. 100kbps　　d. 100Mbps
12. 进行异或运算（符号⊕表示异或）：
   a. （10001）⊕（10000）
   b. （10001）⊕（10001）（由结果推断出什么？）
   c. （11100）⊕（00000）（由结果推断出什么？）
   d. （10011）⊕（11111）（由结果推断出什么？）
13. 按表10.1，发送方发送数据字10。一个3位突发性差错损坏了码字，接收方能否检测出差错？说出理由。
14. 按表10.2，发送方发送数据字10。如果一个3位突发性差错损坏了码字的前3位，接收方能否检测出差错？说出理由。
15. 下列代码字的汉明距离是多少？
   a. d（10000，00000）　　b. d（10101，10000）
   c. d（11111，11111）　　d. d（000，000）
16. 对下列情况，求最小汉明距离：
   a. 检测2个差错　　b. 纠正2个差错
   c. 检测3个差错与纠正2个差错　　d. 检测6个差错与纠正2个差错
17. 按表10.2，如果接收到下列码字，试问数据字是什么？
   a. 01011　　b. 11111
   c. 00000　　d. 11011
18. 证明表10.8所示的编码不是线性编码，只须找到一种情形违反线性的性质。
19. 虽然数学上能证明简单的奇偶校验编码是线性编码，对表10.3中5对代码用手工方法检测其线性性。
20. 说明表10.4中的汉明编码C（7，4），通过检测下列各情况的编码可检测出2位差错但不一定能检测出3位差错。在突发性差错中，字符“∀”是指没有差错，字符“H”是指有差错。
   a. 数据字：0100　突发性差错：∀H H∀∀∀∀

表10.8 练习10.18

数据字	码字
00	00000
01	01011
10	10111
11	11111

b. 数据字：0111　突发性差错：E ∇ ∇ ∇ ∇ ∇E
c. 数据字：1111　突发性差错：E ∇E ∇ ∇ ∇E
d. 数据字：0000　突发性差错：EE E ∇ ∇ ∇ ∇

21. 说明表10.4中的汉明编码C (7, 4)。在下列各情况通过检验代码可检测出1位差错但不一定能检测出多位差错。在突发性差错中，字符“∇”是指没有差错。字符“E”是指有差错。
a. 数据字：0100　突发性差错：E∇ ∇ ∇ ∇ ∇ ∇
b. 数据字：0111　突发性差错：∇E∇ ∇ ∇ ∇ ∇
c. 数据字：1111　突发性差错：E∇ ∇ ∇ ∇ ∇E
d. 数据字：0000　突发性差错：EE∇ ∇ ∇ ∇E

22. 虽然可证明表10.6中代码既是线性的又是循环的，仅用两次测试证明部分事实：
a. 测试码字0101100具有循环性质；
*b. 测试码字0010110和1111111具有线性性质。

23. 我们需要至少11位的数据字，求具有$d_{min}=3$的汉明编码$C(n,k)$的n与k值。

24. 对相应的多项式进行下列运算：
a. $(x^7+x^5+x+1)+(x^4+x^3+x+1)$　　b. $(x^4+x^3+x+1)-(x^4+x^3+x+1)$
c. $(x^3+x^2)\times(x^2+x^2+x+1)$　　d. $(x^4+x^3+x+1)/(x^2+1)$

25. 试回答下列问题：
a. 101110代表的多项式是什么？　　b. 101110左移3位其结果是什么？
c. 用多项式重做b。　　d. 101110右移4位其结果是什么？
e. 用多项式重做d。

26. 下列哪个生成多项式能检测出单个位差错？
a. x^3+x+1　　b. x^5+x^2
c. 1　　d. x^2+1

27. 用表10.7提到的CRC-8多项式，回答下列问题：
a. 它能检测出单个位差错吗？说出其理由。
b. 它能检测出长度为6的突发性差错吗？说出其理由。
c. 检测出长度为9的突发性差错的概率是多少？
d. 检测出长度为15的突发性差错的概率是多少？

28. 用表10.7提到的CRC-32多项式，回答下列问题：
a. 它能检测出单个位差错吗？说出其理由。
b. 它能检测出长度为16的突发性差错吗？说出其理由。
c. 检测出长度为33的突发性差错的概率是多少？
d. 检测出长度为55的突发性差错的概率是多少？

29. 假定为偶校验，求下列数据单元的校验位：
a. 1001011　　b. 0001100
c. 1000000　　d. 1110111

30. 已知数据字1010011110和除数10111，
a. 指出发送站点产生的码字（用二进制除法）。
b. 指出接收站点校验的码字（无差错）。

31. 用多项式重做练习30。

32. 发送方须发送4个数据项：0x3456、0xABCC、0x02BC和0xEEEE，回答下列问题：
    a. 求发送站点的校验和；
    b. 如果不存在差错，求接收站点的校验和；
    c. 如果第二个数据项变成0xABCF，求接收站点的校验和；
    d. 如果第二个数据项变成0xABCE和第三个数据项变成0x02BA，求接收站点的校验和。
33. 指出用校验和处理一个特殊情况的问题。发送方有两个数据项0x4567和0Xba98要发送。试问校验和的值是什么？

第11章 数据链路控制

数据链路层的两个主要功能是数据链路控制和介质访问控制。前者，即数据链路控制的功能是用来处理两个邻近节点之间通信，即节点到节点通信的设计和程序。我们将在此章节中阐述这一功能。而后者，数据链路层的第二个功能是介质访问控制，或者说如何共享链路，将在第12章中阐述。

数据链路控制（data link control）的功能包括成帧、流量控制和差错控制，以及能提供帧在节点之间流畅且可靠传输的软件实现协议。在这一章，首先阐述成帧，或者说如何组织物理层所承载的位，然后再阐述流量控制和差错控制。关于差错的检测和纠正的技术已在第10章阐述过了。

我们需要协议来实现数据链路控制。每一个协议都是一组以软件实施的规则，并由在数据链路层进行数据交换涉及的两个节点来运行。我们将阐述五个协议：两个用于无噪声（理想的）通道，三个用于有噪声（现实的）通道。第一类的协议并不能真正地实施，而是为理解第二类协议提供了基础。

在阐述了这五个协议后，我们将用一个实例，说明面向位的协议如何通过使用高级数据链路控制协议（HDLC）来实现。并阐述一个流行的面向字节的协议，即点对点协议（PPP）。

11.1 成帧

位于物理层的数据传递是指以信号的形式从源端传输到目的端。物理层提供位同步以保证发送方和接收方能使用相同的位周期和时序。

另一方面，数据链路层需将位组合成帧，并使帧与帧之间是可识别的。邮局系统就实践了一种成帧形式，将一份信插入一个信封的简单动作就将一段信息与另一段信息分离开来，信封就是一种分隔符。另外，既然邮局系统是一种多对多的传输设施，所以每个信封都标明了发送方和接收方的地址。

在数据链路层中，通过添加发送方地址和接收方地址，**成帧**（framing）将一条从源端到目的端的报文分离开来，或者将到不同目的端的报文分离开来。目的地址定义了分组要去的地方，而发送方地址帮助接收方确认接收。

尽管整条报文都应该分组为一个帧，但通常却不这么做。一个原因是一个帧会很大，将使得流量和差错控制变得十分低效。当一个巨大的帧承载一条报文时，即使一个字节的错误也需要重新传递整条报文。

11.1.1 固定长度成帧

帧的大小可以是固定的，也可以是可变的。**固定大小成帧**（fixed-size framing），不需要定义帧的边界，长度本身可以用作分隔符。这种成帧类型的一个例子是ATM广域网，它用称为信元的固定长度帧。我们将在第18章讨论ATM。

11.1.2 可变长度成帧

此章中我们主要讨论在局域网中需行的**可变长度成帧**（variable-size framing）。在可变长度成帧中，需要一种方法来规定一个帧的结束和下一个帧的开始。过去，主要使用两种方法：

面向字符的方式和面向位的方式。

面向字符协议

在面向字符协议（character-oriented protocol）中，数据用来自诸如ASCII（参见附录A）编码系统的8位字符传输。头部通常携带源端地址和目的端地址以及其他控制信息，而尾部携带检错或纠错冗余位，也是8位的倍数。为了将一个帧与下一个分离开来，在帧的开始和结束增加了8位（1字节）的标记（flag）。标记由与协议相关的特殊字符组合而成，标明一个帧的开始和结束。图11.1说明了在面向字符协议中帧的格式。



图11.1 面向字符协议中的帧

当数据链路层只交换文本时，流行面向字符成帧。标记可以选择文本通信中不使用的任何字符。然而现在，我们传输其他如图片、音频、视频等的信息，标记所使用的任何模式可能是这些信息的一部分。如果接收方恰巧在数据的中段接收到这一模式，它会认为已经到了帧的结束处了。为了解决这个问题，在面向字符成帧中使用了**字节填充**（byte-stuffing）策略。字节填充（或字符填充）是指如果存在与标记相同模式的字符的话，便在帧的数据部分插入一个特殊的字节。数据段被填入一个额外的字节，这个字节通常称为**换义字符**（escape character，ESC），它有预定义的位模式。当接收方遇到换义字符时，便会将它从数据段移走，并仍将下一字符作为数据处理，而不是当作分界的标记。

通过换义字符来进行字节填充允许标记出现在帧的数据段中，但也产生了另一问题。如果紧随标记，文本包含了一个或多个换义字符该怎么办？接收方移走了换义字符，但保留了标记，会被错译为帧的末端。为了解决这一问题，作为文本一部分的换义字符必须能被另一个换义字符标注。换言之，如果换义字符是文本的一部分，那么要增加额外的换义字符来表明第二个换义字符是文本的一部分。图11.2说明了这一情况。

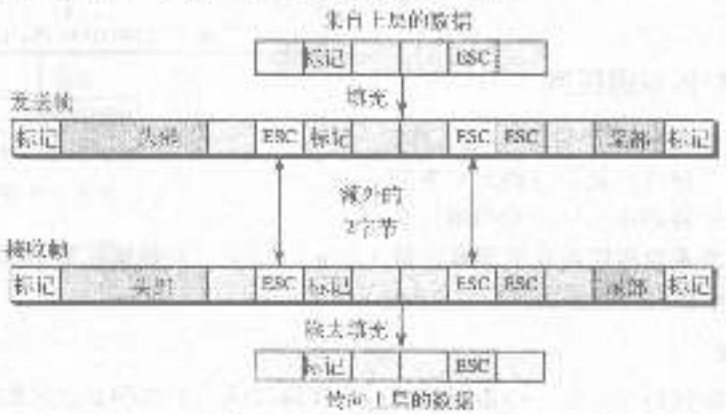


图11.2 字节填充和移除

> 字节填充是当文本中存在一个标记或换义字符时，添加一个额外字节的过程。

面向字符协议在数据通信中产生了另一个问题。如今广泛应用的编码系统，例如Unicode有16位和32位字符，就会与8位字符产生冲突。所以我们可以说，以下将讨论的面向位协议将是大势所趋。

面向位协议

在面向位协议（bit-oriented protocol）中，帧的数据段是一个位的序列，将由上层翻译成文本、图片、音频、视频等等。然而，除了头部（也可能是尾部）之外，我们仍然需要一个分隔符将帧与帧区分开来。多数协议使用一个特殊的8位模式的标记01111110作为分隔符来标明帧的开始和结束，如图11.3所示。

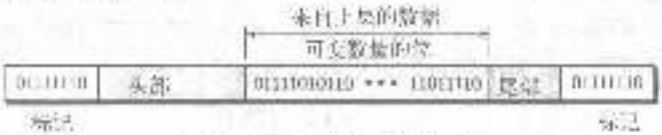


图11.3　面向位协议中的帧

此标记也会产生与面向字节协议相同的问题。也就是说，当标记的模式出现在数据中时，我们需要一定的信息告知接收方这并不是帧的结束。我们通过填充一个简单的位（而非一个字节）使得这个模式区别于标记，这个策略就是位填充（bit stuffing）。在位填充中，当遇到1个0和5个连续的1，便增加一个0。这个额外填充的位最终将被接收方移走。注意在1个0和5个连续的1后面填加1个额外的位而不管接下来那个位是什么，这就保证了标记不会出现在帧中。

> 位填充是指在遇到1个0后面紧跟着5个1时便添加一个0的过程，使得接收方不会误认01111110是一个标记。

图11.4说明了发送方填充位和接收方移走位的过程。注意，当碰到在5个1以后如果是0，我们仍然填充0，这个0将会被接收方移走。

这就意味着如果有类似标记的模式01111110出现在数据中时，它会变成011111010（填充后），而不会被接收方误认为是一个标记。而真正的标记01111110是不会被发送方填充的，故而能被接收方正确地确认。

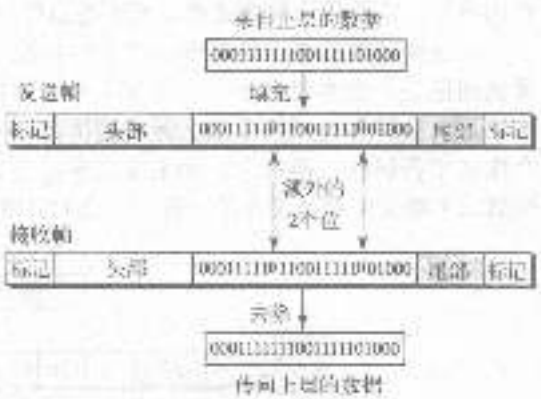


图11.4　位填充和移除

11.2　流量控制和差错控制

数据通信至少需要两个设备一起工作，一个发送而另一个接收。就算是如此基本的过程也需要大量的协调来实现互相理解的交换。数据链路层最重要的职能就是流量控制（flow control）和差错控制（error control）。确切地说，这些功能被视为数据链路控制（data link control）。

11.2.1　流量控制

流量控制是数据链路层的一项重要职能，它在接收确认前协调发送的数据数量。在大多数协议中，流量控制是一系列程序，告知发送方在等到来自接收方的确认之前它能传输多少数据，任何接收设备都有处理进入数据的速度限制和存储进入数据的容量限制。接收设备必须能够在达到这些限制前提示发送设备，并要求传输设备发送的帧少一些或者暂时停止发送。进入的数据必须经过校验和处理才能使用，这些过程通常比传输速度慢。因此，每个接收设备都有一块存储区，称之为缓冲区，存储进入的数据直到它们被处理。如果缓冲区也装满了，接收方必须能告知发送方停止传输直到它能够再次接收为止。

流量控制就是一系列程序，用来限制发送方在等到确认之前发送的数据数量。

11.2.2 差错控制

差错控制包括差错检测和纠正。它使得接收方能够提示发送方在传输过程中有帧的丢失或破坏，并协调发送方重新传输这些帧。在数据链路层，术语差错控制通常指差错检测和重传的方法。在数据链路层中，差错控制一般容易实现：在交换的任何时刻检测到一个差错，就要重传这个出错的帧。这个过程被称之为**自动重复请求**（automatic repeat request，ARQ）。

数据链路层的差错控制基于自动重复请求，即重传数据。

11.3 协议

现在我们来看看数据链路层如何将成帧、流量控制和差错控制结合起来，从而实现节点对节点的数据传输。在软件中，通常使用一种通用程序语言实现协议。为了使我们的讨论不受程序语言的影响，我们对每一协议用伪码进行编写，从而主要关注程序本身而非研究程序语言的规则细节。

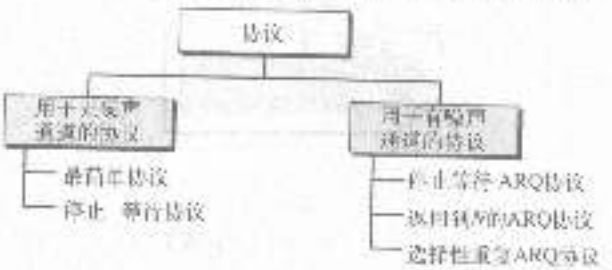


图11.5 本章所讨论的协议分类

我们在此讨论的协议与在真实网络中使用的协议是有区别的。此处我们讨论的协议是单向的，数据帧从一个节点，称之为发送方，传输到另一个节点，称之为接收方。虽然有些特殊的帧，我们称之为**确认**（acknowledgment，ACK）和**否定确认**（negative acknowledgment，NAK），但为了实现流量控制和差错控制，这些帧可以反方向流动，但数据只能单方向流动。

在现实生活的网络中，数据链路协议是双向实现的：数据可以双向流动。在这些协议中，流量控制和差错控制的信息，诸如ACK和NAK通过**携带**（piggybacking）技术被包含在数据帧中。因为双向协议要比单向协议复杂的多，我们只讨论后者。只要理解了它们，就可以扩展到双向协议了。我们把扩展留做练习。

11.4 无噪声通道

首先，我们假设有一种不会丢失帧、复制帧或损坏帧的理想通道。然后介绍此类通道的两种协议。第一种协议不使用流量控制，而第二种使用流量控制。当然，两者都不使用差错控制，因为，我们假设这种通道是完美的无噪声通道。

11.4.1 最简单的协议

我们讨论的第一个协议就称为**最简单的协议**（simplest protocol）而没有其他特定的命名，这个协议没有流量控制和差错控制。与其他我们将要在本章中讨论的协议一样，此协议是单向的，数据帧从发送方到接收方单方向传输。我们假定接收方可以即时处理它所接收的任何帧，而所花费的时间少到可以忽略不计。数据链路层中的接收方可以即时去除帧的头部并将数据分组传递到它的网络层，而网络层会即时收到分组。换言之，接收方永远不会因进入的帧而超载。

设计

在这种方案中无需流量控制。发送方站点的数据链路层从网络层获取数据，然后将数据成帧后发送。接收方站点的数据链路层从物理层接收到帧，从帧中取出数据，然后将数据传输到网络层。数据链路层的发送方和接收方为其网络层提供传输服务。数据链路层使用其物理层提供的服务（例如发信号、多路复用等）进行位的物理传输，图11.6给出了一个设计。

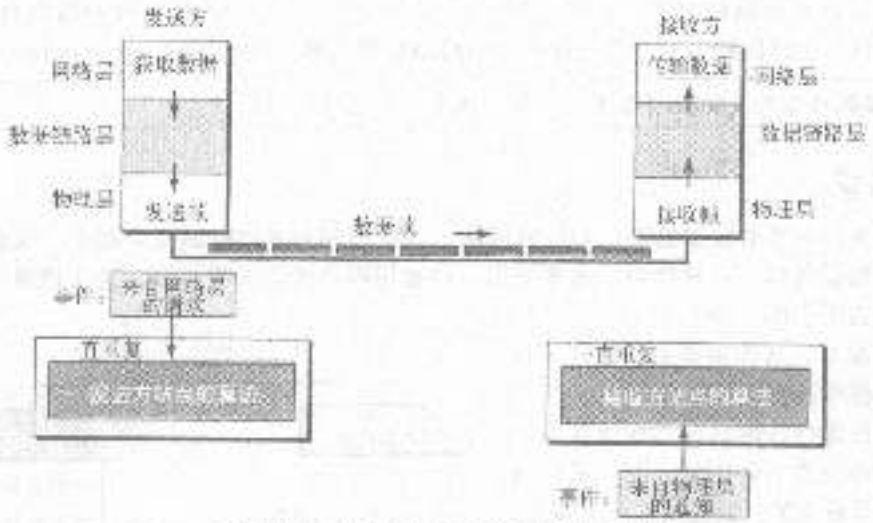


图11.6　没有流量控制和差错控制的最简单协议的设计

我们需要精心设计数据链路层所使用的程序，发送方站点直到其网络层有数据包要发送时才发送帧，接收方站点直到有帧到达才将数据分组传递给它的网络层。当协议用程序实现时，我们就要介绍协议中的**事件**（event）概念。发送方的程序不断地运行着，直到有来自网络层的请求才会有行为。接收方的程序也不断地运行着，直到有来自物理层的通知才会有行为。两端的程序都不断地运行，因为它们不知道相关事件何时发生。

算法

算法11.1给出了发送方的程序。

算法11.1　在最简协议中发送方站点的算法

```
1  while(true)                    //不断重复
2  {
3    WaitForEvent();              //当一个事件出现时，处理它
4    if(Event(RequestToSend))     //有一个要发送的分组
5    {
6      GetData();
7      MakeFrame();
8      SendFrame();               //发送帧
9    }
10 }
```

分析　该算法有一个无限的循环，也就是说当程序一旦启动，第3行到第9行就会一直重复。该算法是事件驱动型的，也就是说它休眠（第3行）直到一个事件唤醒它（第4行）。这意味着第3行与第4行之间的执行存在不确定的时间段：它们之间存在缺口。当事件，即网络层发出的请求发生时，第6行到第8行被执行。然后，程序开始重复循环，它又在第3行休眠直到下一个事件发生时。我们为主要过程编写了伪代码，而并没有说明GetData、MakeFrame和

SendFrame这些模块的细节。GetData（）若从网络层获取数据包，MakeFrame（）在数据包中添加头部和分界的标记形成帧，SendFrame（）则将帧传输到物理层。

算法11.2给出了接收方站点的程序。

算法11.2 在最简协议中接收方站点的算法

```
1 while(true)                    //不断重复
2 {
3   WaitForEvent();              //当一个事件出现时，唤醒它
4   if(Event(ArrivalNotification)) //数据帧到达
5   {
6     ReceiveFrame();
7     ExtractData();
8     DeliverData();             //将数据发送给网络层
9   }
10 }
```

分析 除了帧和数据的传输方向是向上的以外，该算法和算法11.1的格式相同。此处的事件是数据帧的到达，该事件发生后，通过使用ReceiveFrame()语句，数据链路层从物理层获取帧，通过使用ExtractDAta()语句从帧中取出数据，并通过使用DeliverData()语句将数据传输到网络层。这里，我们也使用事件驱动型算法，因为该算法无法知晓帧何时会到达。

例11.1

图11.7例举了一个使用最简协议通信的例子。这个例子非常简单。发送方发送一个帧序列而不用考虑接收方。为了发送三个帧，在发送方站点和接收方站点各发生了三个事件。注意图中倾斜的长方块表示数据帧，方块的高度定义为帧中第一位和最后一位传输的时间差。

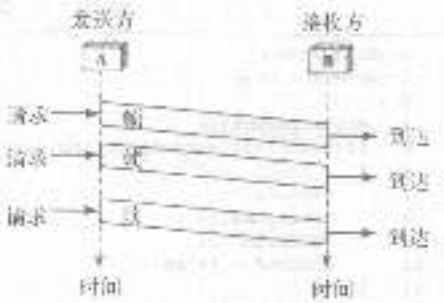


图11.7 例11.1的流量图

11.4.2 停止等待协议

如果到达接收方的数据帧的速度大于它们能被处理的速度，帧在使用前必须被存储。通常，接收方没有足够的存储空间，特别是它将接收来自多个源的数据时。这样将导致帧被丢掉或者服务被拒绝。为了避免接收方出现帧超负荷，我们要想办法让发送方减缓速度，也就是说接收方必须给发送方反馈信息。

我们现在讨论的协议称为停止等待协议（stop-and-wait protocol），因为当发送方发出一个帧后必须停下来直到接收到来自接收方的确认（可以继续了），然后再发送一个帧。这里，数据帧仍是单向通信的，除了辅助的ACK帧（确认的简单标记）从另一端传输过来。我们在前面的协议中加入流量控制。

设计

图11.8描述了该过程。将该图与图11.6对比，我们可以看到前向通道（从发送方到接收方）和反向通道的数据流。在任何时刻，不是在前向通道有一个数据帧，就是在反向通道中有一个ACK帧。为此，我们需要一个半双工的链路。

算法

算法11.3是发送方站点的算法。

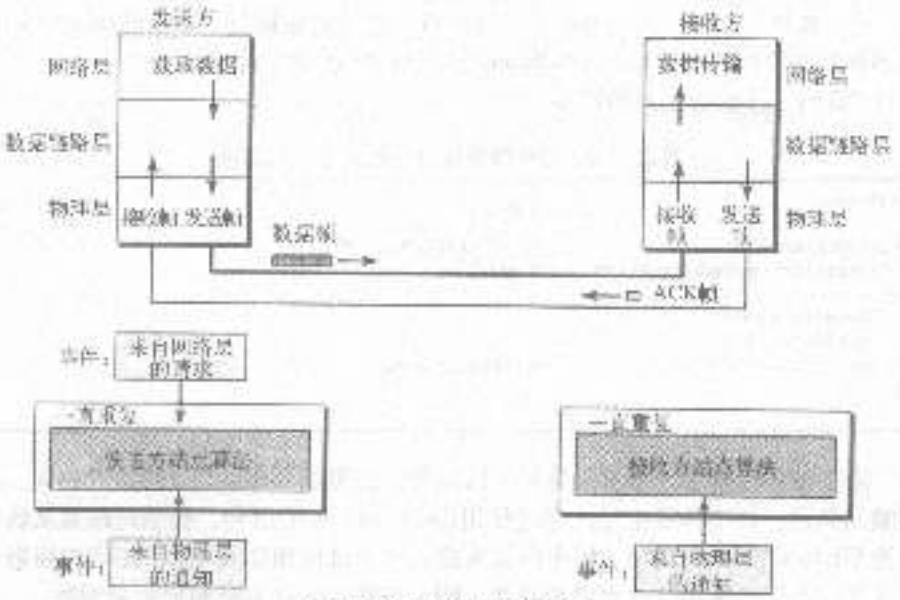


图11.8 停止等待协议的设计

算法11.3　停止-等待协议中发送方算法

```
1  while(true)                          //不断重复
2  canSend = true                       //允许发送第一个帧
3  {
4    WaitForEvent();                    //当一个事件出现时，唤醒它
5    if(Event(RequestToSend) AND canSend)
6    {
7       GetData();
8       MakeFrame();
9       SendFrame();                    //发送数据帧
10      canSend = false;                //不能发送直到ACK到达
11   }
12   WaitForEvent();                    //当一个事件出现时，唤醒它
13   if(Event(ArrivalNotification)      //ACK到达
14    {
15      ReceiveFrame();                 //接收ACK帧
16      canSend = true;
17    }
18 }
```

分析　这里会发生两个事件：来自网络层的请求或来自物理层的通知。对这些事件的响应必须是交替的。换言之，当发送一个帧后，算法必须忽略另一个网络层的请求直到该帧被确认。我们知道两个到达事件不会一个紧接着一个发生，因为该通道是无差错的且不能复制帧。但是，网络层发送的请求可以一个紧接着一个发生而中间无需有到达事件的发生。所以，我们要在一定程度上阻止数据帧的即时发送。虽然有很多方法，但我们使用简单的变量*canSend*，该变量既可以是正确（true）也可以是错误（false）。当一个帧被发送，该变量被设置为错误，以表示不能发送新的网络请求直到*canSend*变为正确。当一个ACK帧被接收，*canSend*便被设置为正确，以允许下一个帧的发送。

算法11.4描述了接收方站点的程序。

分析　除了一处不同外，这个算法与算法11.2十分相似。当数据帧到达后，接收方发送一个ACK帧（第9行）以确认接收到并允许发送方发送下一个帧。

算法11.4 停止等待协议中接收方站点算法

```
1  while(true)                          //不断重复
2  {
3    WaitForEvent();                    //当一个事件发生时，唤醒它
4    if(Event(ArrivalNotification))     //告诉帧到达
5    {
6      ReceiveFrame();
7      ExtractData();
8      Deliver(data);                   //将数据发送给网络层
9      SendFrame();                     //发送一个ACK帧
10   }
11 }
```

例11.2

图11.9 例举了一个使用停止-等待协议的例子。这个例子仍然很简单。发送方发送一个帧后要等待接收方的反馈。当ACK帧到达后，发送方发送下一个帧。注意，该协议中发送两个帧涉及了发送方的四个事件和接收方的两个事件。

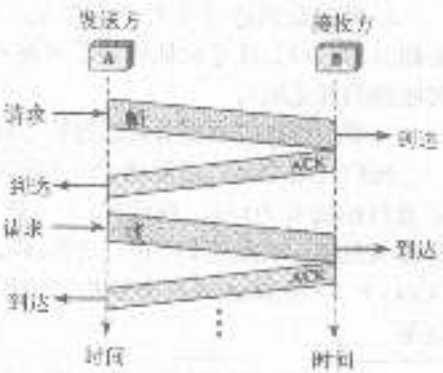


图11.9 例11.2的流量图

11.5 有噪声通道

虽然停止-等待协议给出了我们如何在共程序中增加流量控制的概念，但是无噪声通道是不存在的。我们能忽略差错（就想我们时常做的那样），或者我们需要为我们的协议增加差错控制。这部分我们讨论三个使用差错控制的协议。

11.5.1 停止等待自动重复请求

第一个协议称为**停止等待自动重复请求**（Stop-and-Wait ARQ），是在停止等待协议中增加了一个简单的差错控制机理。让我们来看看这个协议如何检测并纠正差错。

为了检测和纠正被损坏的帧，我们需要给数据帧加上冗余的位（见第10章）。当帧到达接收方站点时，它被检测，如果它被损坏了，它会被无声地丢弃。此协议中差错检测由接收方的静音来证明。

对丢失帧的处理远比对损坏帧的处理要难的多。在我们先前的协议中，我们没有办法确认一个帧。接收到的帧可以是正确的，或者复制的，也可以是无序的。解决的途径是为帧编号。当接收方接收到一个无序的数据帧时，就意味着这些帧或者是丢失的或者是被复制的。

在此协议中，要重发被损坏的帧和丢失的帧。当有差错时，如果接收方没有响应，那么发送方如何知道要重发哪个帧呢？为了解决这个问题，发送方保留已发送帧的副本。与此同时，它启动一个定时器。如果定时器到时且没有收到已发送帧的ACK，就要重发该帧，保留副本并重启定时器。既然该协议使用停止等待机理，那么只有一个特定帧需要ACK，尽管网络中会存在同一个帧的多个副本。

> 在停止等待自动重复请求协议中，差错的检测由保留已发送帧的副本并当定时器到时对重传这个帧来实现。

既然一个ACK帧同样会被损坏或丢失，它也需要冗余位和一个序列号。此协议中的ACK帧有一个序列号字段。此协议中，发送方只要简单的丢弃一个损坏的ACK帧或者忽略一个无序的帧。

序列号

正如我们讨论的那样，此协议要求必须对帧进行编号。这通过使用序列号（sequence number）来完成。数据帧要增加一个字段来容纳帧的序列号。

序列号的排列是一个值得考虑的重要问题。既然我们希望帧的长度越小越好，我们寻找能提供无歧义通信的最小范围。序列号当然是可以循环的，例如，我们决定字段m位长，序列号从0开始，到2^m-1，再开始重复。

让我们来推出所需序列号的范围。假定我们使用x作为一个序列号，接下来我们只需用$x+1$，而没有必要用$x+2$。为了加以演示，假定发送方已经发送了序列号为x的帧，那么将会发生三种情况。

1. 帧安全到达接收方，接收方发送确认。确认到达发送方，促使发送方发送下一个序列号是$x+1$的帧。

2. 帧安全到达接收方，接收方发送确认，但是确认被损坏或丢失了。在超时后发送方重发帧（编号x），注意此处的帧是重复帧。接收方可以确认这一事实，因为它预期帧$x+1$到达但接收到的却是帧x。

3. 帧被损坏或者没有到达接收方时，在超时后发送方重发帧（编号x）。

我们可以看到有必要使用序列号x和$x+1$，因为接收方需要区分情况1和情况2。但是没有必要将帧编号为$x+2$。在情况1中，帧可以再编号为x，因为帧x和$x+1$被确认而且在任何一端都没有歧义。在情况2和3中，新的帧是$x+1$，不是$x+2$。如果只需要x和$x+1$，我们可以使$x=0$及$x+1=1$。这就意味着序号是01010等。这个模式熟悉吗？这就是我们在第10章看到的模2运算。

> 在停止等待ARQ中，我们使用序列号给帧编号。该序列号基于模2运算。

确认号

既然序列号必须既适用于数据帧，也要适用于ACK帧，那么我们使用这个约定：确认号总是告知接收方期待的下一个帧的序列号。例如，如果帧0安全到达，接收方发送一个带有确认1的ACK帧（表示下一个期待收到的是帧1）。如果帧1安全到达，接收方则发送一个带有确认0的ACK帧（表示下一个期待收到的是帧0）。

> 在停止等待ARQ中，确认号总是以模2运算宣布期待收到的下一个帧的编号。

设计

图11.10描述了停止等待ARQ协议的设计，发送设备保留最后传输的帧的副本直到接收到该帧的确认。一个数据帧使用一个序列号，一个ACK帧使用一个确认号。发送方有一个控制变量，我们称为S_n（发送方，下一个要发送的帧），它保存下一个将被发送的帧的序号（0或1）。

接收方有一个控制变量，我们称之为R_n（接收方，期待收到的下一个帧），它保存下一个期待收到帧的编号。当一个帧被发送，S_n值递增（模2），也就是说如果它是0，它会变成1；如果是1则会变成0。当一个帧被接收时，R_n值递增（模2），也就是说如果它是0，它会变成1；如果是1则会变成0。发送方站点将有三个事件发生，接收方站点发生一个事件。变量S_n指向的时隙与已经发送的帧的序列号相匹配，但没有被确认；R_n指向的时隙与期待收到的帧的序列号相匹配。

算法

算法11.5是发送方站点的算法。

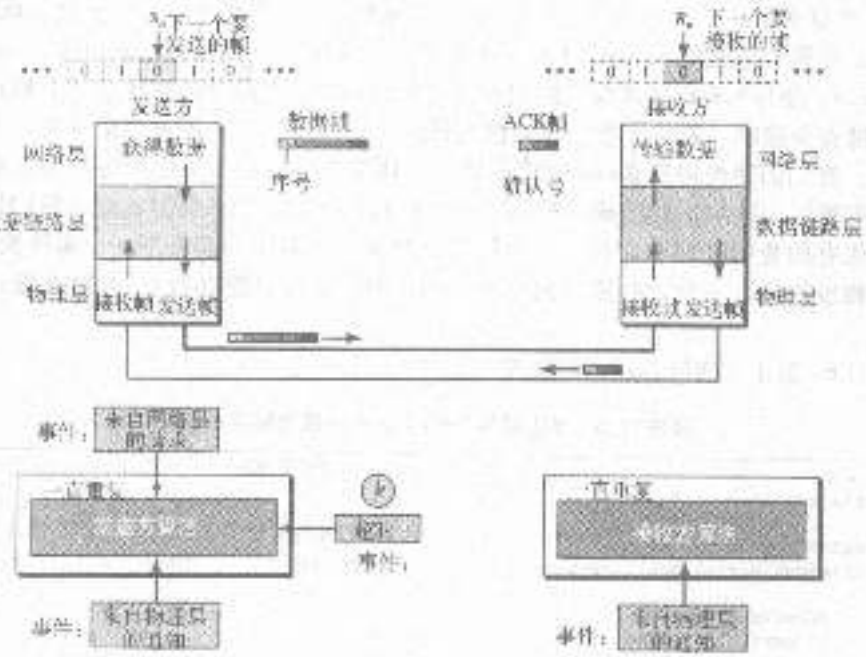


图11.16 停止等待ARQ协议设计

算法11.5 停止等待ARQ协议中发送方站点的算法

```
1  S0 = 0;                                  //第0帧最先发出
2  canSend = true;                          //允许最初请求
3  while(true)                              //无休止重复
4  {
5    WaitForEvent();                        //当一个事件发生时，唤醒它
6    if(Event(RequestToSend) AND canSend)
7    {
8      GetData();
9      MakeFrame(S0);                       //加序号为S0
10     StoreFrame(S0);                      //保留副本
11     SendFrame(S0);
12     StartTimer();
13     S0 = S0 + 1;
14     canSend = false;
15   }
16   WaitForEvent();                        //休眠
17   if(Event(ArrivalNotification)          //ACK或NAK到达
18   {
19     ReceiveFrame(ackNo);                 //接收ACK帧
20     if(not corrupted AND ackNo == S0)    //有效ACK帧
21       {
22         StopTimer();
23         PurgeFrame(S0-1);                //不再需要的副本
24         canSend = true;
25       }
26     }
27
28   if(Event(TimeOut)                      //超时
29   {
30     StartTimer();
31     ResendFrame(S0-1);                   //发送副本检查
32   }
33 }
```

分析 我们首先注意到S_0，它是下一个将要发送的帧的序列号，该变量只初始化一次

（第1行），但每次发送一个帧后（第13行），它便递增为下一个帧数准备。然而，既然这是模2运算，那么序列号就是0、1、0、1等。注意第一个事件的进程（SendFrame，SotreFrame，PurgeFrame），使用S_n来定义被发送的帧。我们至少需要一个缓冲区来保存这个帧直到我们能确认它已被安全接收。第10行指出在帧被发送前，它被保存了。副本用来重发一个损坏或者丢失的帧。我们仍然使用变量canSend在前一个帧安全接收之前防止网络层提出请求。如果这个帧没有损坏，且ACK帧的确认号与下一个将要发送的帧的序列号匹配，我们便停掉定时器并清除保存的数据帧副本。否则，我们只要忽略这个事件并等待下一个事件发生就行了。每一个帧被发送后，一个定时器开始计时。当定时器超时（第28行），帧被重发，且定时器被重置。

算法11.6　显示了接收方站点的程序。

算法11.6　停止等待ARQ协议中接收方站点的算法

```
1  Rn = 0;                                  //试0有望首先到达
2  while(true)
3  {
4    WaitforEvent();                        //当一个事件发生时，唤醒它
5    if(Event(ArrivalNotification))         //数据试到达
6    {
7       ReceiveFrame();
8       if(corrupted(frame));
9          sleep();
10      if(seqNo == Rn)                     //有效数据试
11      {
12       ExtractData();
13        DeliverData();                    //传送数据
14        Rn = Rn + 1;
15      }
16       SendFrame(Rn);                     //发送ACK帧
17   }
18 }
```

分析　这显然与算法11.4不同。首先，所有已到达数据帧的损坏被忽略了。当帧的序列号是期待收到的那个（R_n）时，帧才被接收，数据才被传输到网络层，R_n值增加。然而，这里有一个微小的差别，即使数据帧的序列号与下一个期待的帧的序列号不匹配，也会发送一个ACK帧给发送方。然而这个ACK只是再确认前一个ACK，而不是确认接收到的帧。这样做是因为接收方认为先前的ACK有可能已经丢失了，接收方便发送一个副本。重发的ACK可能在时钟超时之前解决问题。

例11.3

图11.11给出了一个停止等待ARQ协议的例子，帧0被发送并被确认，帧1丢失了并在超时之后被重发。重发的帧1被确认，定时器便停止计时。帧0被发送并确认，但是确认丢失了。发送方无法判断是帧被丢失了还是确认被丢失了，因此超时后，它重发帧0，虽然帧0已被确认了。

效率

如果我们的通道粗且长的话，那么前部分讨论的停止等待ARQ协议是十分低效的。粗指的是通道有很大的带宽，长指的是往返延迟较长，这两者的乘积称为带宽延迟乘积（bandwidth-delay product），正如我们在第3章中讨论的。我们可以把这种通道看作一根管道，那么带宽延迟乘积就是以位为单位的管道容量。管道总是在那里，如果我们不使用它，那么就是无效的。带宽延迟乘积是一种尺度，用来度量我们在等待来自接收方的消息的同时所能发送到系统外的位数。

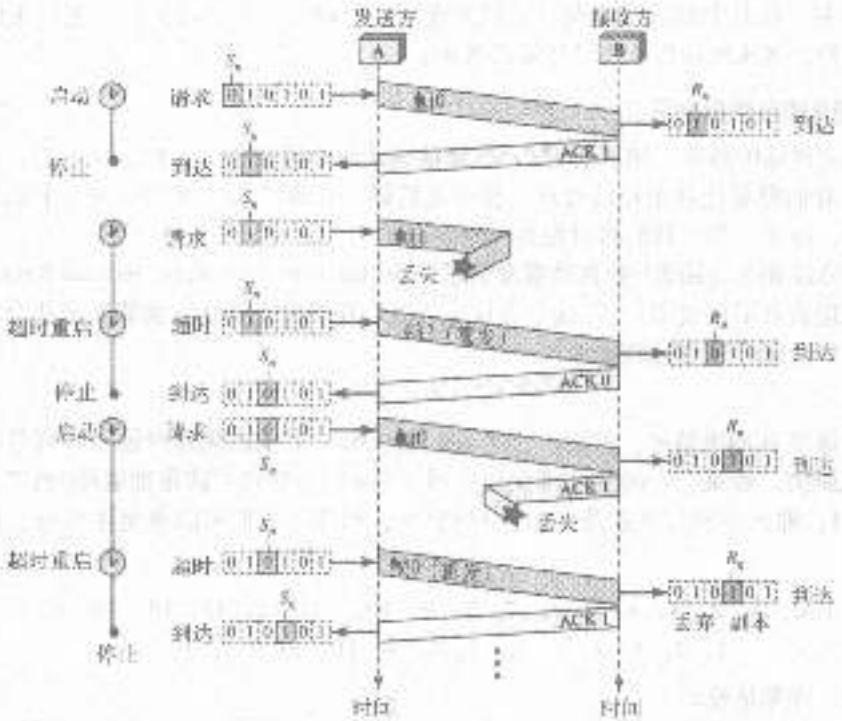


图11.11 例11.3的流量图

例11.4

假定，在一个停止等待ARQ系统中，带宽是1Mbps，且一个位往返一次是20微秒。那么带宽延迟乘积是多少？如果系统的数据帧的长度是1000位，那么链路的利用率是多少？

解：

带宽延迟乘积是

$$(1 \times 10^6) \times (20 \times 10^{-3}) = 20\,000 \text{位}$$

在数据从发送方到接收方再返回的期间，系统能够发送20 000位，然而，系统只能发送1 000位。我们因此说链路的利用率仅有1 000/20 000，即5%。正因为此，如果链路有高带宽或长延迟的话，使用停止等待ARQ协议是对链路能效的一个浪费。

例11.5

在例11.4中，如果我们拥有一种协议能在停止和担心确认之前发送15个帧，那么链路层的利用率是多少？

解：

带宽延迟乘积仍然是20 000位。在一个往返过程中，系统能发送15个帧或者15 000位。这意味着利用率是15 000/20 000，即75%。当然，如果存在损坏的帧，那么利用率就要低很多，因为这些帧要被重发。

流水线操作

在网络或者其他领域中，一项任务通常在前一项完成后开始，这就是流水线操作(pipelining)。在停止等待ARQ协议中不存在流水线操作，因为我们必须等待一个帧到达目的地并被确认后才能发送下一个帧。然而，流水线操作应用到了我们下面的两个协议中，因为

在我们得到关于前几个帧的消息前，就能发送后几个帧了。如果传输的位是巨大的，考虑到带宽延迟乘积，流水线操作改进了传输的效率。

11.5.2　回退N帧自动重发请求

为了提高传输的效率（填满管道），在等待确认时必须传输多个帧。换言之，在发送方等待确认时，我们需要让多个帧待处理来使通道忙碌。在这一节，我们讨论一个能够达到这个目的的协议，在下一节，我们将讨论另一个。

第一个协议称之为回退N帧自动重发请求（Go-Back-N Automatic Repeat Request）（该名称的基本原理会在后面说明）。在这个协议中，我们在接收到确认之前能发送几个帧，在确认到达之前我们会保存这些帧的副本。

序列号

帧在发送站点按序编号。然而，因为我们要让每一个帧的头部中包含序列号，我们需要设置一个限制值。如果一个帧的头部允许序列号有m位，序列号的范围就是0到2^m-1。例如，如果m等于4，那么序列号只能是0到15中的数字。当然，我们可以重复序列号。因此，序列号是

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0,$$
$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\cdots\cdots$$

换言之，序号是模2^m

> 在回退N帧协议中，序列号是模2^m，这里m是以位为单位的序列号字段长度。

滑动窗口

在此协议（和下一个协议）中，**滑动窗口**（sliding window）是一个抽象的概念，用来定义发送方和接收方关心的序列号范围。换言之，发送方和接收方只需处理可能的序列号部分。发送方关心的序列号范围称为**发送滑动窗口**（send sliding window），接收方关心的则称为**接收滑动窗口**（receive sliding window），我们将对两者进行讨论。

发送窗口是一个假想的盒子，包含了能传输的数据帧的序列号。在每一个窗口中，一部分序列号定义了已经发送的帧，另一部分定义了能够被发送的帧。窗口的最大尺度是2^m-1，原因我们以后阐述。在这章中，我们固定窗口的大小，使之为最大值。但是在以后的章节中我们会看到一些协议有可变的窗口大小。图11.12给出了一个大小为15（$m=4$）的滑动窗口。

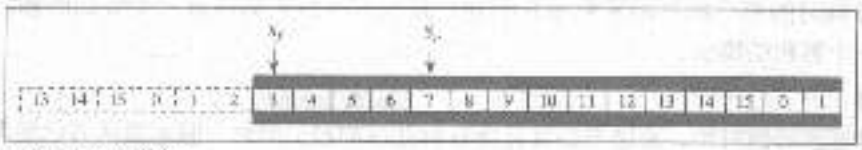
在任何时刻窗口将可能适用的序号分成三部分。第一部分，从最左端到窗口的左墙，定义属于已经被确认的帧的序号。发送方不用担心这些帧，也不用保留它们的副本。第二部分，图11.12a的着色部分，定义了属于已发送但处于未知状态的帧的序号。发送方必须等待直到知道这些帧被接收或者被丢失，我们称这些帧为待处理的帧。第三部分，图中的空白部分，显示了那些能够被发送的帧的序号范围，但还没有从网络层接收到相对应的数据包。最后，第四部分显示了那些直到窗口滑动才能使用的序号，正如我们接下来看到的那样。

窗口本身是一个抽象体，在任何时刻用三个变量界定了它的大小和位置，我们称这些变量为S_f（发送窗口，第一个待处理的帧），S_n（发送窗口，下一个要发送的帧），和S_{size}（发送窗口，大小）。变量S_f定义了第一个（最先）待处理帧的序号，变量S_n定义了分配给下一个将发送帧的序号。最后，变量S_{size}定义了窗口的大小，该变量在此协议中是固定的。

> 发送窗口是一个抽象概念，通过三个变量S_f，S_n和S_{size}定义的一个大小为2^m-1的假想盒子。



a) 滑动前的发送窗口



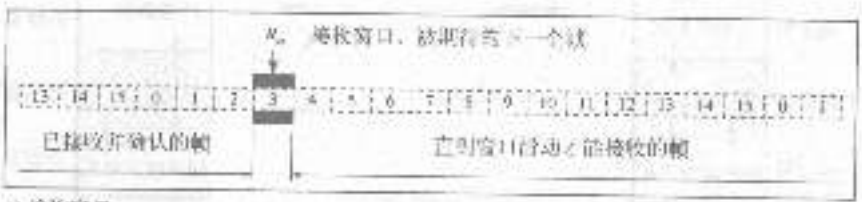
b) 滑动后的发送窗口

图11.12 回退N帧自动重发请求协议中的发送窗口

图11.12b描述了当另一端发来的确认到达时，一个发送窗口可以向右滑动一个或者多个时隙。正如我们不久可以看到的那样，在此协议中，确认是累加的。这意味着一个ACK帧可以确认一个以上的帧。在图11.12b中，帧0，1和2已被确认，因此窗口向右滑动了三个时隙。注意S_f的值是3，因为帧3现在是待处理的帧。

有效确认到达时，发送窗口能滑动一个或者多个时隙。

接收窗口确定所接收的数据帧时，发送正确的确认。接收窗口的大小总是1。接收方总是在等待一个指定帧的到达。任何到达的失序帧都会被丢掉并要求重发。图11.13显示了接收窗口。



a) 接收窗口



b) 滑动后的窗口

图11.13 回退N帧自动重发请求协议中的接收窗口

接收窗口是一个抽象概念。通过唯一变量R_n定义了一个大小为1的滑动窗口，当一个正确的帧到达后，窗口滑动，但每次只滑动一个时隙。

注意：我们只需要一个变量R_n（接收窗口，期待的下一个帧）来定义这个抽象概念。窗口左边的序号属于那些已接收并确认的帧，右边的序号显示了那些不能被接收的帧。任何已收到帧的序号属于这两部分都会被丢弃。只有当一个帧的序号与R_n匹配时，才被接收和确认。

接收窗口同样会滑动，但是每次只滑动一个时隙。当收到一个正确的帧（且每次只能收到一个帧）时，窗口才滑动。

定时器

虽然每一个已发送的帧都能有一个定时器，但在我们的协议中我们只用一个。原因是第一个待处理帧的定时器总是先到时，当这个定时器到时时，我们发送所有的待处理帧。

确认

当一个帧安全有序到达时，接收方发送一个肯定的确认。如果一个帧被损坏了或者收到时次序颠倒了，接收方便不响应并将丢弃所有后来的帧直到它收到一个它所期待的帧。接收方的不响应使得不被确认的帧的定时器在发送方站点到时。这样，接下来会使发送方返回，并从定时器到时的那个帧开始重发所有的帧。接收方不需要确认每一个收到的帧。它能为多个帧发送一个累积的确认。

重发帧

当定时器到时的时候，发送方会重发所有待处理的帧。例如，假定发送方已经发送了帧6，但是帧3的定时器到时了。这就意味着帧3没有被确认，发送方就要返回并重发帧3，4，5和6。这就是这个协议称之为回退N帧自动重发请求协议的原因。

设计

图11.14描述了该协议的设计。正如我们可以看到的，多个帧能在前向方向传输，在反向传输多个确认。这个理念与停止等待ARQ协议类似，区别在于发送窗口允许我们传输与其时隙一样的帧。

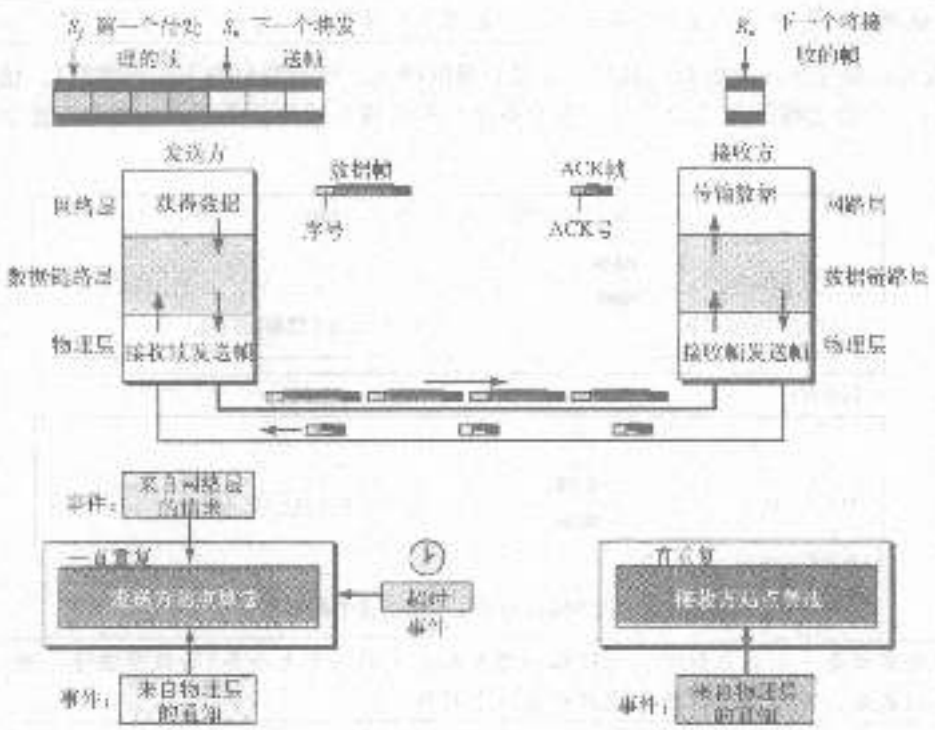


图11.14　到N步返回的自动重复请求协议的设计

发送窗口大小

我们现在可以阐述为什么发送窗口的大小必须小于2^m了。作为一个示例，我们使$m=2$，这就意味着窗口的大小是2^m-1，即3。图11.15将一个大小为3的窗口和一个大小为4的窗口进

行了比较。如果窗口的大小为3（小于2^2）且二个确认被丢失了，帧0的定时器到时，那么所有三个帧都要被重发。接收方现在期待的是帧3，而不是帧0，于是副本帧被正确地舍弃了。另一方面，如果窗口的大小是4（等于2^2）且所有确认被丢失了，发送方就要重发帧0的副本。然而，这次接收方的窗口期待收到帧0，于是它接收了帧0，是下一周期中的第一个帧，而不是一个丢失帧的副本，这是一个差错。

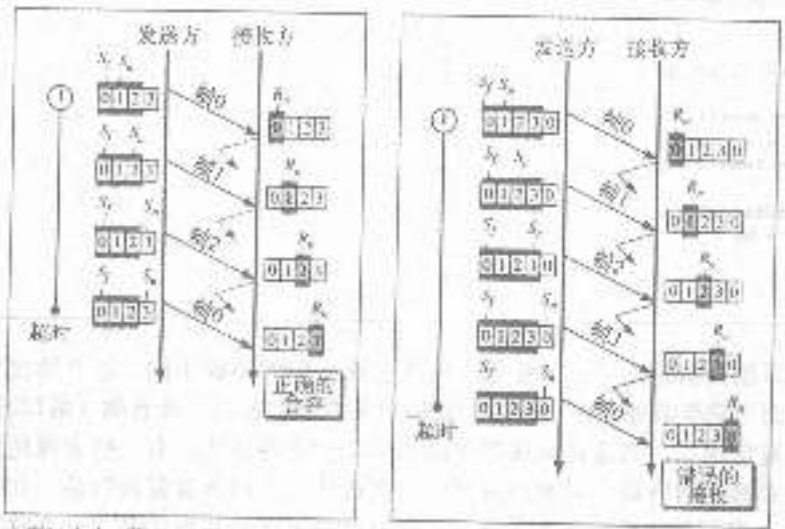


图11.15 在回退N帧请求重发协议中窗口的大小

在回退N帧自动重发请求协议中，发送窗口的大小必须小于2^n，而接收窗口的大小始终是1。

算法

算法11.7是在回退N帧自动重发请求协议中发送方算法。

算法11.7 回退N帧自动重发请求的发送方算法

```
1  Sw = 2^n - 1;
2  Sf = 0;
3  Sn = 0;
4
5  while (true)                          //不断重复
6  {
7   WaitForEvent();
8    if(Event(RequestToSend))            //发送帧
9    {
10      if(Sn-Sf >= Sw)                  //如果窗口满
11          Sleep();
12      GetData();
13      MakeFrame(Sn);
14      StoreFrame(Sn);
15      SendFrame(Sn);
16      Sn = Sn + 1;
17      if(tiMer not running)
18          StartTimer();
19   }
20
21   if(Event(ArrivalNotification)) //ACK帧到达
22   {
23      Receive(ACK);
```

```
24      if(corrupted(ACK))
25            Sleep();
26      if((ackNo>S_f)&&(ackNo<=S_n))  //如果为有效
27      while(S_f <= ackNo)
28       {
29        PurgeFrame(S_f);
30        S_f = S_f + 1;
31       }
32       StopTimer();
33    }
34
35    if(Event(TimeOut))                //计时器超时
36    {
37     StartTimer();
38     Temp = S_f;
39     while(Temp < S_n)
40      {
41       SendFrame(S_f);
42       S_f = S_f + 1;
43      }
44    }
45  }
```

分析　该算法先初始化了三个变量，与停止等待ARQ协议不同，这个协议允许从网络层得到多个请求而不需要其他事件的发生。我们只需要确定窗口并没有满（第12行）。在我们的方式中，如果窗口满了，请求就会被忽视而网络层就要再尝试。有一些实现使用了其他的方法比如激活或者抑制网络层。对到达事件的处理远比之前的协议复杂得多。如果我们收到一个损坏的ACK，我们就忽略它；如果这个ackNo（ACK序号）属于其中一个待处理的帧，我们就用一个循环来清除缓冲区并将窗口的左墙移到右边。超时事件同样更复杂。我们首先启动一个新的定时器，然后我们重发所有的待处理帧。

算法11.8是在回退N帧自动重发请求协议中接收方算法。

算法11.8　回退N帧自动重发请求接收方算法

```
1  R_n = 0;
2
3  while (true)                      //不断重复
4  {
5    WaitForEvent();
6
7    if(Event(ArrivalNotification)) //数据帧到达
8    {
9      Receive(Frame);
10     if(corrupted(Frame))
11           Sleep();
12     if(seqNo == R_n)              //如果为期帧到
13     {
14       DeliverData();              //传送数据
15       R_n = R_n + 1;              //滑动窗口
16       SendACK(R_n);
17     }
18   }
19 }
```

分析　这个算法是简单的，我们忽略了损坏的帧或者失序的帧。如果一个帧与一个被期待的序列号一起到达，我们就传送数据，更新R_n的值，并发送一个ACK，其ackNo指明了下一个被期待的帧。

例11.6

图11.6描述了一个回退N帧的例子。这个例子中前向通道是可靠的，而反向通道则不可靠。没有数据帧丢失，但是有一些ACK被延迟了并且有一个被丢失了。这个例子同样指出了再确

认被延迟或丢失的情况下，累积的确认能有所帮助。

在初始化后，有7个发送方事件。请求事件由来自网络层的数据触发，到达事件由来自物理层的确认触发。这里没有超时事件，因为所有待处理的帧在超时前都被确认了。注意，虽然ACK2丢失了，ACK3能起到ACK2和ACK3两者的作用。

这里有4个接收方事件，所有的事件由来自物理层的帧的到达触发。

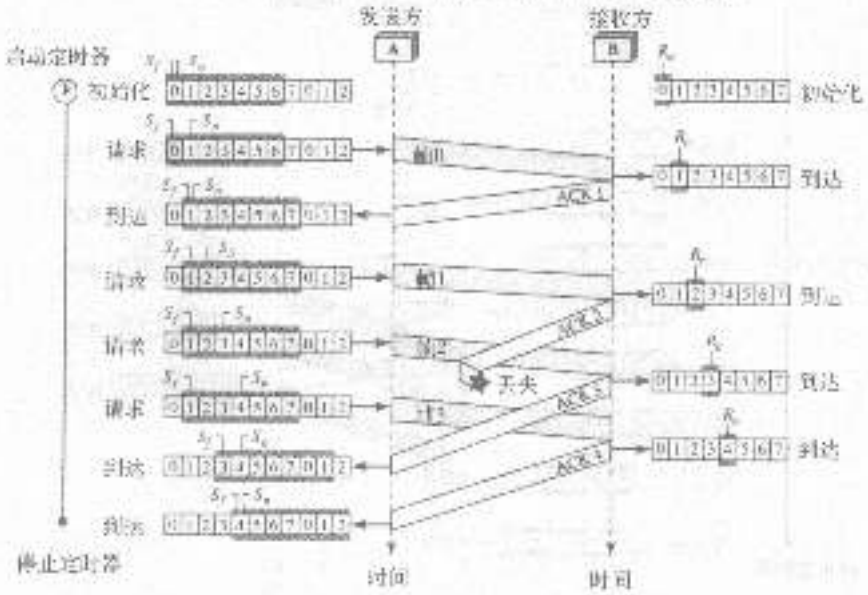


图11.16 例11.6的流量图

例11.7

图11.7说明了如果一个帧丢失了，情况会怎样。帧0、1、2和3被发送了，而帧1丢失了。接收方收到了帧2和3，但是它们被舍弃了因为接收到时它们失序了（被期待的是帧1）。发送方没有收到任何与帧1、2和3相关的确认。定时器最终到时，发送方再发送所有待处理的帧（1、2和3），因为它不知道哪里出错了。注意，帧1、2和3的重发是对一个单一事件的响应。当发送方对这一事件响应时，它不能再接受其他事件的触发。这就意味着，当ACK2到达时，发送方仍然在忙着发送帧3，物理层必须等待直到这一事件被完成，而数据链路层重新回到休眠状态。我们画了一条垂线来显示延迟，ACK3发生了同样的状况，但当ACK3到达时，发送方正忙着应答ACK2。当ACK4到达时，同样的状况又发生了，注意在第二个定时器到时前，所有待处理的帧已经都被发送了且定时器停止了。

回退N帧ARQ与停止等待ARQ的比较

读者能发现回退N帧的ARQ与停止等待ARQ有相同之处，我们可以认为停止等待ARQ协议实际上是回退N帧ARQ协议的一种，即当只有2个序列号而窗口的大小为1时。换言之，就是$m=1$，$2^m-1=1$。在回退N帧的ARQ中，条件是$>2^m$，而在停止等待ARQ中，条件是2，即$m=1$时，与2^m一样。

> 停止等待ARQ是一种窗口大小为1的特殊的回退N步帧ARQ协议。

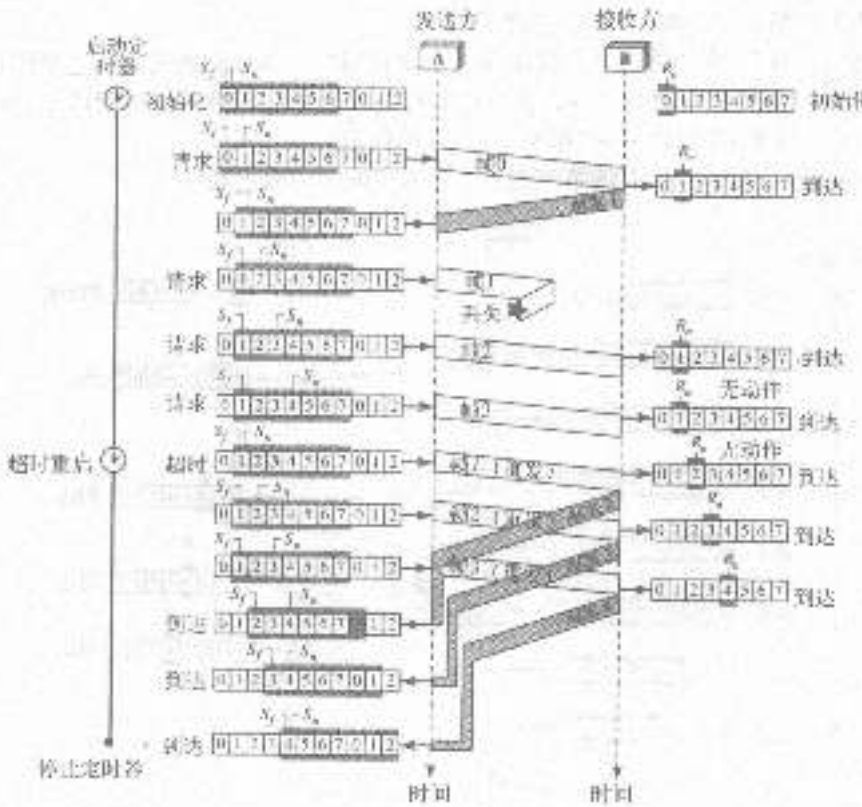


图11.17　例11.7的流量图

11.5.3　选择性重复ARQ

回退N帧ARQ协议简化了接收方的程序。接收方只持有一个变量，且没有必要缓冲失序的帧，它们只是被简单地丢弃。然而，这个协议在有噪声链路中十分低效。在有噪声链路中，一个帧被损坏的可能性更大，这就意味着要重发多个帧。重发占用了带宽并减慢了传输的速度。对于有噪声链路而言，有另一种机制可以在一个帧被损坏时不必重发N个帧。这个机制就称为选择性重复ARQ（Selective Repeat ARQ）。它对有噪声链路更为有效，但是接收方的程序也更复杂。

窗口

选择性重复协议同样使用两个窗口：发送窗口和接收窗口。但是，这些窗口与回退N帧协议中的窗口有所不同，首先，窗口的尺度更小，为2^{m-1}。原因我们在后面解释。第二，接收窗口与发送窗口的大小一样。

发送窗口最大只能是2^{m-1}。例如，如果$m=4$，序号为0到15，但是窗口的大小只能是8（而在回退N帧协议中，它是15）。窗口尺度变小意味着对管道填充的有效性减少，但事实上副本帧会少很多，能使这点得到补偿。该协议使用了与回退N帧协议相同的变量。图11.18显示了选择性重复协议中的发送窗口来强调窗口的大小。将其与图11.12进行比较。

在选择性重复协议中的接收窗口与回退N帧的接收窗口完全不同。首先，接收窗口的大小和发送窗口的一样（2^{m-1}）。选择性重复协议允许与接收窗口大小一样多的帧乱序到达，并被保留直到有一组有序帧能交付给网络层。因为发送窗口的大小和接收窗口的一样，所有被发送的帧能乱序到达并被保存直到它们被传输。然而，我们需要指出的是接收方从不会将无序

的数据分组交付给网络层。图11.19显示了该协议中的发送窗口。窗口中的青色时隙指的是那些乱序到达的帧，这些帧被交付给网络层前正在等待其相邻帧的到达。



图11.18 选择性重复ARQ的发送窗口



图11.19 选择性重复ARQ的接收窗口

设计

该协议的设计与我们描述的回退N帧协议的设计有一定程度的相似，但是正如图11.20所示，它要复杂的多。

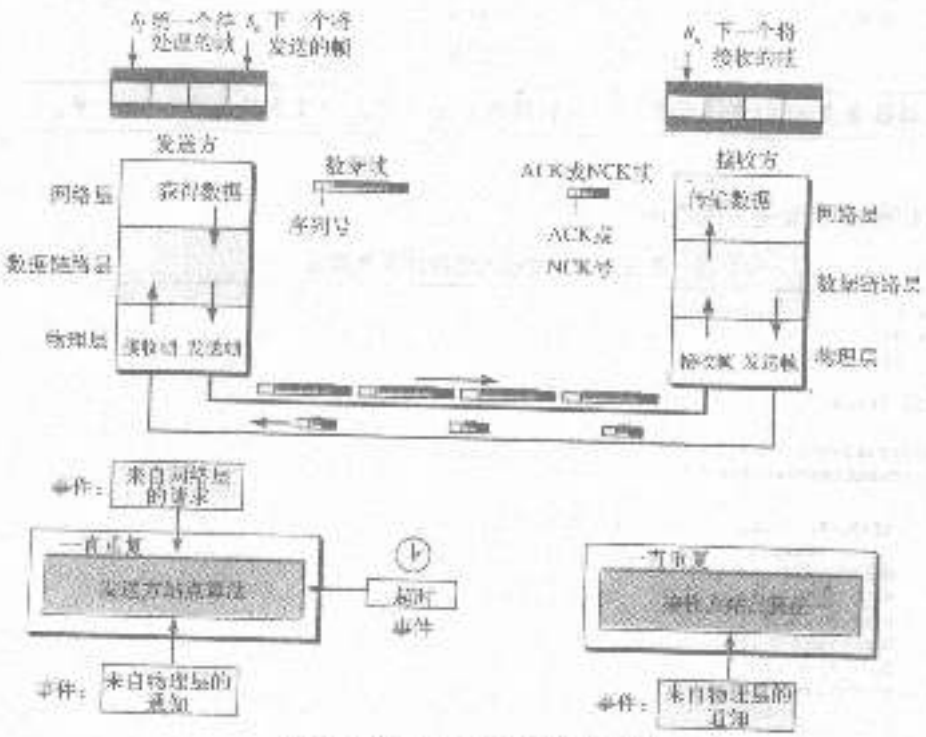
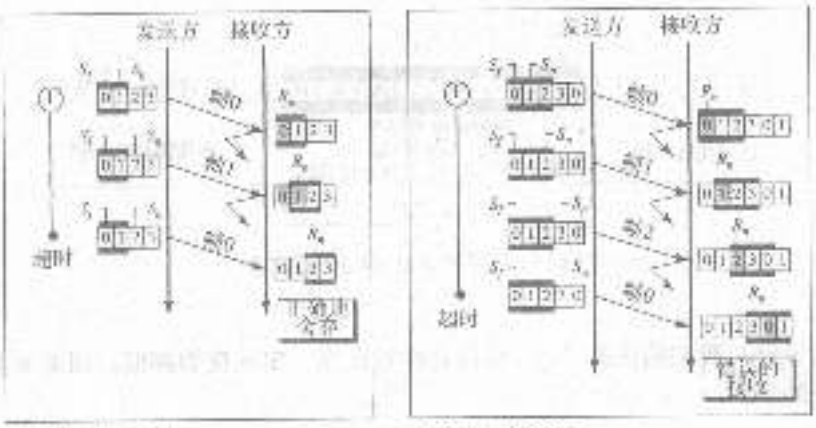


图11.20 选择性重复ARQ协议的设计

窗口大小

我们现在能解释为什么发送窗口和接收窗口的大小最多是2^m的一半了。举例来说，我们使$m=2$，意味着窗口的大小为$2^m/2$，即2。图11.21将大小为2的窗口与大小为3的窗口做了一个比较。

如果窗口的大小为2且所有的确认都丢失了，那么当帧0的定时器到时时，帧0就被重发。然而，接收方的窗口现在期待的是帧2而不是帧0，那么帧0的副本就会被正确地丢弃。当窗口大小为3且所有的确认丢失了时，发送方发送了一个帧0的副本。而这次，接收方窗口期待接收帧0（0是窗口的一部分），所以它接收帧0，但它是下一个周期的第一个帧，而不是一个副本。这明显是一个错误。



a) 窗口大小=2^{m-1}　　b) 窗口大小>2^{m-1}

图11.21　选择性重复ARQ中的窗口大小

> 在选择性重复ARQ中，发送方窗口和接收方窗口的大小最多只能是2^m的一半。

算法

算法11.9描述了发送方的程序。

算法11.9　发送方选择性重复算法

```
1  S_w = 2^(m-1);
2  S_f = 0;
3  S_n = 0;
4
5  while (true)                          //不断重复
6  {
7     WaitForEvent();
8     if(Event(RequestToSend))           //有机器发送
9     {
10       if(S_n-S_f >= S_w)              //如窗口满
11          Sleep();
12       GetData();
13       MakeFrame(S_n);
14       StoreFrame(S_n);
15       SendFrame(S_n);
16       S_n = S_n + 1;
17       StartTimer(S_n);
18    }
19
20    if(Event(ArrivalNotification))     //帧到达
21    {
22       Receive(frame);                 //接收ACK或NAK
23       if(corrupted(frame))
```

```
            Sleep();
        if (FrameType == NAK)
           if (ackNo between Sf and Sn)
           {
            resend(ackNo);
            StartTimer(ackNo);
           }
        if (FrameType == ACK)
           if (ackNo between Sf and Sn)
           {
            while(Sf < ackNo)
            {
             Purge(Sf);
             StopTimer(Sf);
             Sf = Sf + 1;
            }
           }
    }

    if(Event(TimeOut(t)))          //计时器超时
    {
     StartTimer(t);
     SendFrame(t);
    }
}
```

分析　请求事件的处理与前一协议相类似，除了每一个帧的发送都启动一个定时器外。这里到达的事件要复杂得多，到达的可能是一个ACK，也可能是一个NAK。如果一个有效的NAK帧到达了，我们只需重发相应的帧。如果一个有效的ACK到达了，我们使用一个循环来清除缓冲区，停止相应的定时器，并移动窗口的左墙。此处的超时事件比较简单，只有超时的帧才重发。

算法11.10描述了接收方的程序。

算法11.10　接收方选择性重复算法

```
1  Rn = 0;
2  NakSent = false;
3  AckNeeded = false;
4  Repeat(for all slots)
5      Marked(slot) = false;
6
7  while (true)                              //不断重复
8  {
9    WaitForEvent();
10
11   if(Event(ArrivalNotification))          //数据帧到达
12   {
13      Receive(Frame);
14      if(corrupted(Frame))&& (NOT NakSent)
15      {
16       SendNAK(Rn);
17       NakSent = true;
18       Sleep();
19      }
20      if(seqNo <> Rn)&& (NOT NakSent)
21      {
22       SendNAK(Rn);
23       NakSent = true;
24       if ((seqNo in window)&&(!Marked[seqNo]))
25       {
26        StoreFrame(seqNo)
27        Marked(seqNo)= true;
28        while(Marked(Rn))
29        {
30         DeliverData(Rn);
```

```
31          Purge(R,I)
32          R, = R, + 1;
33          AckNeeded = true;
34          }
35          if(AckNeeded)
36          {
37          SendAck(R,I);
38          AckNeeded = false;
39          NakSent = false;
40          }
41        }
42      }
43    }
44 }
```

分析　此处我们要有更多的初始化。为了不使另一端因为NAK而超负荷，我们使用了一个称为NakSent的变量。为了明确何时我们需要发送一个ACK，我们使用一个名为AckNeeded的变量。这两个变量都初始化为错误状态（false）。一旦相应的帧到达并被保存了，我们会用一系列变量来标记接收窗口中的时隙。如果我们收到一个损坏的帧而NAK帧还未被发送，我们就发送一个NAK去告知另一方我们还未收到我们期待的帧。如果帧没被损坏且序列号在窗口中，我们保存这个帧并在时隙上做标记。如果从R_n开始相邻的帧都被标记了，我们将它们的数据交付给网络层并滑动窗口。图11.22描述了这一情形。



图11.22　在选择性ARQ中传输数据

例11.8

这个例子与例11.3相似，都是一个帧丢失了。它描述了选择性重复在这个情况下如何工作。图11.23描述了这一情形。

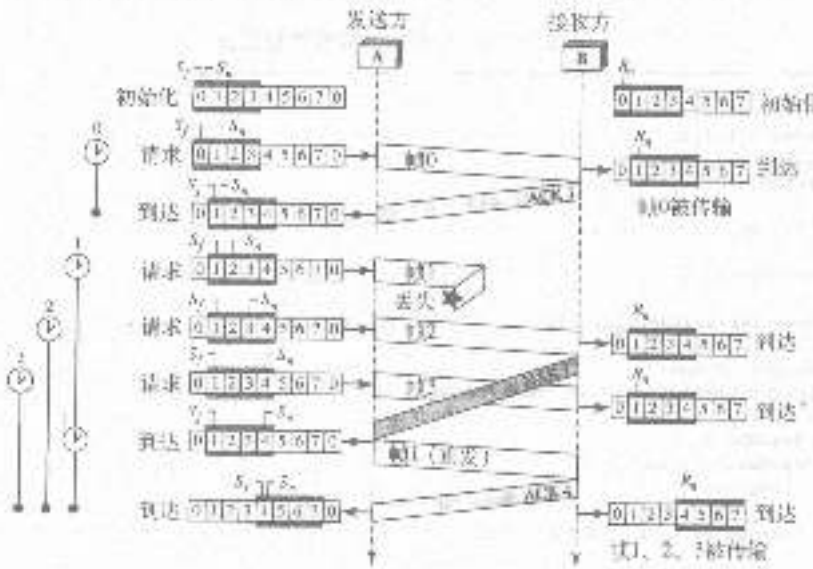


图11.23　例11.8的时序图

一个主要的区别是定时器的数量。这里，每发送或重发一个帧都需要一个定时器，这就意味着定时器也要编号（0、1、2和3）。在第一个请求时，帧0的定时器启动，并在这个帧的ACK到达时停止。第二个请求时，帧1的定时器启动，当NAK到达时再重启，直到最后的ACK到达时才最终停止。其他两个定时器在相应的帧发送时启动而在最后的到达事件发生时

停止。在接收方站点我们需要分辨帧的接收和向网络层交付帧。在第二次到达发生时，帧2到达了并被保存和标记了（着色时指），但是它不能被交付因为帧1丢失了。下一个到达事件，帧3到达了并被保存和标记了，但是仍然没有帧能被交付。只有当最后一个到达事件发生，即帧1的副本到达了，帧1，2和3才能被交付给网络层。将帧交付给网络层有两个条件：第一，一组连续帧到达；第二，这组帧从窗口的起点开始。在第一个到达事件后，只有一个帧到达，它是从窗口的起点开始的。到最后一个到达事件后，有三个帧且第一个是从窗口的起点开始的。

另一个重要之处是在第二个到达事件后，一个NAK帧被发送了，但在第三个到达事件后没有发送NAK。虽然情况看上去一样。原因是这个协议不想让网络充斥不必要的NAK帧和不必要的重发帧而拥堵。第二个NAK帧仍然会是NAK1去提醒发送者再重发帧1，这已经由第一个NAK做过了，第一个NAK被发送是被记忆的（使用nakSent变量），不应该再发送直到帧滑动。每一个窗口位置只发送一个NAK并指明窗口中的第一个时隙。

接下来这点是关于ACK的。注意这里只发送了两个ACK。第一个ACK只确认了第一个帧，而第二个确认了三个帧。在选择性重复协议中，只有当数据被交付到网络层时才发送ACK。就算属于一个时隙的n个帧的数据被传输，也只发送一个ACK。

11.5.4 捎带

在这部分我们讨论的三个协议都是单向的：数据帧只朝一个方向流动而诸如ACK和NAK等的控制信息则能反方向传输。在现实生活中，数据帧通常可以双向流动：从节点A到节点B，或者从节点B到节点A。这就是说控制信息也要能双向传输。一项称之为捎带技术的运用提高了双向协议的效率。当一个帧从A到B运载数据时，它也能从B运载关于帧到达或丢失的控制信息。当一个帧从B到A运载数据时，它也能从A运载关于帧到达或丢失的控制信息。

我们在图11.24中描述了使用捎带的回退N帧ARQ的设计。注意每一个节点都有两个窗口：一个发送窗口和一个接收窗口。两者都需要使用定时器。两者都涉及三个类型的事件：请求，

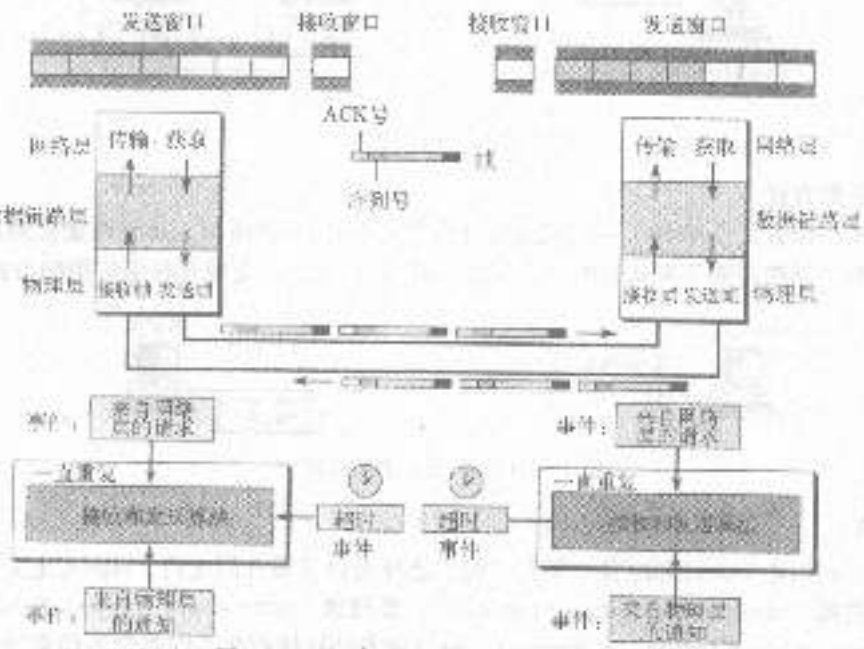


图11.24 回退N返回ARQ的捎带的设计

到达和超时。然而，此处的到达事件比较复杂，当一个帧到达时，每一端都要同时处理帧和控制信息。所以必须小心处理到达事件。请求事件只使用每一方的发送窗口，而到达事件既要用到发送窗口也要用到接收窗口。

关于捎带，另一重要之处是发送方和接收方要使用同一种算法。该算法是复杂的，因为它要将两个到达事件合而为一。我们将它留为一个练习。

11.6　高级数据链路控制

高级数据链路控制（HDLC）是一种面向位的点到点和多点链路进行通信的协议。它实现了在本章中讨论的ARQ机制。

11.6.1　配置和传输方式

HDLC提供了在不同的配置中都能使用的两种通用的传输模式：**正常响应方式**（normal response mode，NRM）和**异步平衡方式**（asynchronous balanced mode，ABM）。

正常响应方式

在正常响应方式中，站点配置是不平衡的。有一个主站（primary station）和多个从站（secondary station）。主站能发送指令，而从站只能响应。如图11.25所示，NRM能在点到点和多点链路中使用。

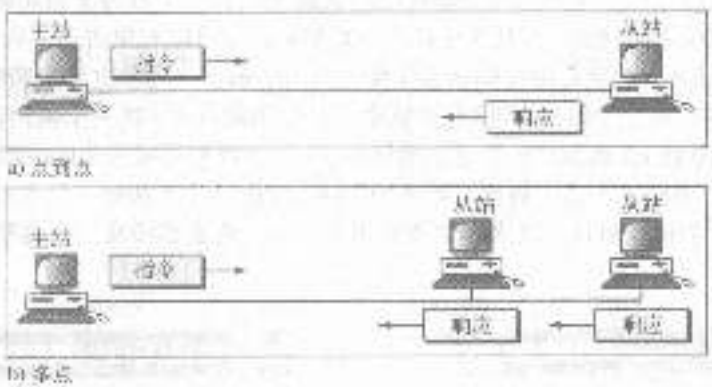


图11.25　正常响应方式

异步平衡方式

在异步平衡方式（ABM）中，配置是平衡的。如图11.26所示，该链路是点到点的，且每个站点既有主站的功能又有从站的功能（作为对等站点）。这是现在普遍应用的方式。



图11.26　异步平衡方式

11.6.2　帧

为了对刚刚述的方式和配置中所有可能的选择提供灵活性的支持，HDLC定义了三种类型的帧：**信息帧**（information frames，I-frames）、**管理帧**（supervisory frames，S-frames）和**无编号帧**（unnumbered frames，U frames）。每一种类型的帧都在不同类型的信息传输中扮演着

封的角色。信息帧用来传输用户数据及与用户数据相关的控制信息（捎带）。管理帧只用来传输控制信息。保留无编号帧作为系统管理之用。无编号帧携带的信息用来管理链路自身。

帧的格式

在HDLC中，每个帧都最多可包含六个字段：如图11.27所示：开始标记字段，地址字段，控制字段，信息字段，帧校验序列字段，结束标记字段。在多帧传输中，一个帧的结束标记可以作为下一个帧的开始标记。



图11.27 HDLC帧

字段

现在我们来讨论各个字段及它们在不同类型的帧中的作用。

- **标记字段** 一个HDLC帧的标记字段是8位序列，其模式是01111110，既可作为一个帧的开始也可以作为一个帧的结束，为接收方提供同步模式。
- **地址字段** 一个HDLC帧的第二个字段包含了从站的地址。如果主站产生帧，它包含着一个"去往"地址。如果从站产生帧，它则包含一个"来自"地址。一个地址字段可能是一个字节也可能是多个字节，这取决于网络的需求。1字节可以最多定义128个站点（1个位用于其他目的了）。更大的网络需要多字节地址字段。如果地址字段只有一个字节，最后的1位通常是1。如果地址字段多于一个字节，所有的字节除了最后一个字节外都以0结束，只有最后一个字节以1结束。中间的字节都以0结束是为了提示接收方会有更多的地址字节到来。
- **控制字段** 控制字段是帧的1个字节或2个字节段，用于流量控制和差错控制。这个字段中位的解释取决于帧的类型，我们将以后讨论这种字段及它在每种类型的帧中的格式。
- **信息字段** 信息字段包含了来自网络层的用户数据或者管理信息。不同的网络该字段的长度不同。
- **帧校验序列字段** 帧校验序列字段（FCS）是HDLC的差错校验字段。它可以包含2到4个字节的ITU-T CRC。

11.6.3 控制字段

控制字段决定帧的类型及其功能。因此，让我们更为详细地阐述这种字段的格式。其格式对于一种类型的帧是特定的。如图11.28所示：

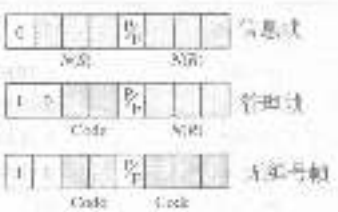


图11.28 不同类型帧的控制字段

信息帧的控制字段

信息帧被设计用来运载来自网络层的用户数据。另外，它们也能包含流量控制和差错控制信息（捎带）。控制字段的子字段用来定义这些功能。第一个位决定类型，如果控制字段的第一个位是0，这就意味着这是一个信息帧。接下来的三个位，

称为N(S)，定义帧的序号。注意通过这一个字节，我们能定义的序号是从0到7；但是在控制字段为2字节的扩展格式中，这部分也会大得多。最后的三个位，称为N(R)，在使用捎带时与确认号相对应。在N(S)和N(R)之间的单个位称之为P/F位。P/F虽是一个单个位，但是有双重用途。它只有被设定时（位=1）才有意义，意味着轮询或终止。当帧由主站发送到从站时（即地址字段包含接收方的地址），它意味着轮询。当帧由从站向主站发送时，它意味着（即地址字段包含发送方的地址）终止。

管理帧的控制字段

无论当捎带是不可能的还是不恰当时（例如，当站点没有自身的数据要发送时，或者站点需要发送一个指令或响应而不是确认时），管理帧都用来进行流量控制和差错控制。管理帧没有信息字段。如果控制字段的前两个位是10，就说明这是一个管理帧。最后三个位，称之为N(R)，与确认（ACK）号或否定确认（NAK）号相对应，它取决于管理帧的类型。称为编码的2个位用来决定管理帧自身的类型。使用2个位，我们能产生4种类型的管理帧，如下所述：

- 准备接收（RR）。如果编码的子字段是00，这就是一个RR管理帧。这种帧对一个或一群安全接收的帧加以确认。在这种情况下，N(R)的值定义了确认号。
- 不准备接收（RNR）。如果编码的子字段是10，这就是一个RNR管理帧。这是一类有附加功能的RR帧。它对一个或一群帧的接收加以确认，并宣布接收方正忙不能接收更多的帧。它通过要求发送方减速来实现拥塞控制机能。N(R)的值是确认号。
- 拒收（REJ）。如果编码的子字段是01，这就是一个REJ管理帧。这是一个NAK帧，但与适用于选择性重复ARQ协议中的不同，它是一个能在回退N帧ARQ协议中使用的NAK，它能在发送方计时到时前，告知发送方最后一个帧丢失或者损坏了，以此来提高程序的效率。N(R)的值是否定确认号。
- 选择性拒收（SREJ）。如果编码的子字段是11，这就是一个SREJ的管理帧。这是一个适用于选择性重复ARQ协议的NAK帧。注意，在HDLC协议中使用术语选择性拒收而不是选择性重复。N(R)的值是否定确认序号。

无编号帧的控制字段

无编号帧负责在连接的设备之间交换会话管理和控制信息。与管理帧不同，无编号帧包含一个信息字段，但是这个信息字段是用作系统管理的，而不是用户数据。虽然与管理帧有相似之处，但无编号帧承载的多数信息包含在含有控制字段中的编码中。无编号帧编码被分成两部分：在P/F位前有一个2位的前缀，在P/F位后有一个3位的后缀，总的而言，这两部分（5位）共能产生32种不同类型的无编号帧。其中常用的一些无编号帧如表11.1所示。

表11.1　无编号帧的指令和响应

编　码	命　令	响　应	含　义
00 001	SNRM		设为正常响应方式
11 011	SNRME		设为正常响应方式，扩展的
11 100	SABM	DM	设为异步平衡方式或启动连接方式
11 110	SABME		设为异步平衡方式，扩展的
00 000	UI	UI	无编号信息
00 110		UA	无编号确认
00 010	DISC	RD	断开连接或请求断开连接
10 000	SIM	RIM	设为初始化方式或请求信息方式
00 100	UP		无编号轮询

（续）

编 码	指 令	响 应	含 义
11 001	RSET		重置
11 101	XID	XID	交换ID
10 001	FRMR	FRMR	坏帧

例11.9 连接/断开

图11.29描述了如何使用无编号帧来建立连接和断开连接。节点A通过一组异步平衡方式（SABM）帧来请求连接；节点B通过一个无编号确认（UA）响给出一个肯定的响应。通过这两个交换，数据能在两个节点之间传输（图中并未显示）。数据传输完毕，节点A发送一个DISC（断开）帧请求断开连接；节点B通过一个UA（无编号确认）帧响应加以确定。

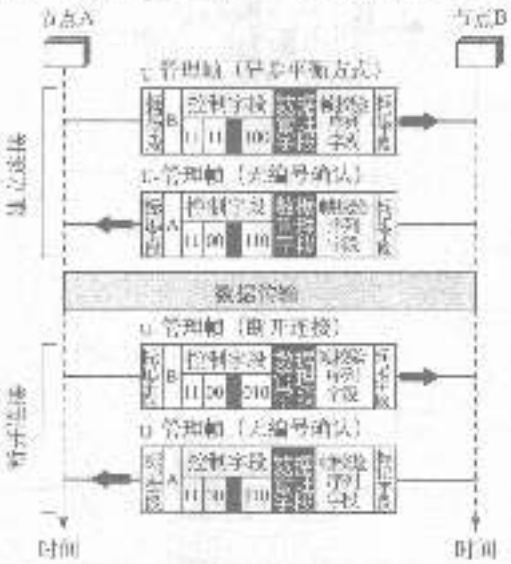


图11.29 建立连接和断开的先决

例11.10 无差错捎带

图11.30描述了一个使用捎带的交换。节点A通过一个编号为0及紧随其后的编号为1的信息帧来开始交换。节点B发送自身的一个信息帧并捎带以上两个帧的确认。节点B的第一个信息帧同样编号为0[$N(S)$字段]并在其$N(R)$字段有一个2，以此来确认对来自节点A的帧0和1的接收，并表示接下来它等待帧2的到达。在从节点A接收更多的帧之前，节点B传输其第二个和第三个信息帧（编号1和2），它的$N(R)$信息并没有因此变化：B的帧1和2提示节点B仍然等待节点A的帧2的到达。节点A发送了它所有的数据，因此，它不能在一个信息帧上捎带上确认，故而发送一个管理帧以代之。编码RR指明节点A仍然处于准备接收状态，$N(R)$字段的数字3告知B帧0、1和2已经被接收且A现在正等待的是帧3。

例11.11 有差错的捎带

图11.31描述了丢失了一个帧的交换。节点B发送了数据帧（0，1，2），但是帧1被丢失了。当节点A收到帧2，它丢弃了帧2并为帧1发送了一个拒收（REJ）帧。注意这是一个使用回退N帧的协议，因此一个REJ帧有着如同NAK帧的特殊用途。在这里NAK要做两件事：它对帧0的接收加以确定并宣布帧1及其后的任何帧都必须重发。收到REJ帧后，节点B重发了帧1和2，

节点A通过发送一个确认号为3的RR帧（ACK帧）对接收加以确认。

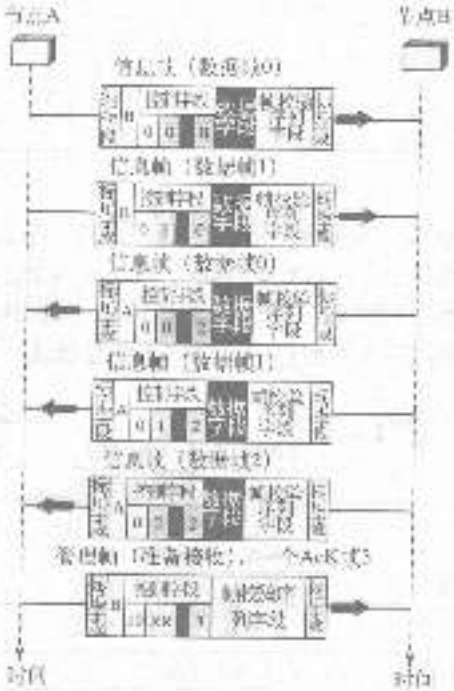


图11.30　无差错捎带的示例

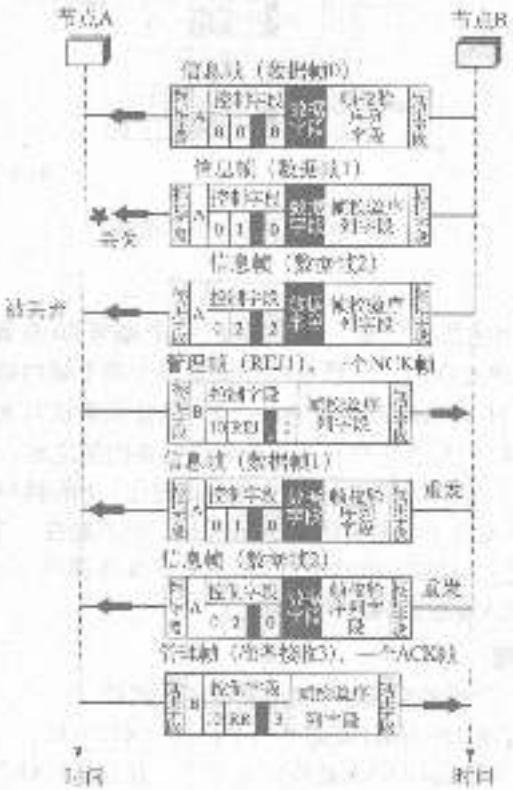


图11.31　有差错捎带的示例

11.7 点到点协议

虽然高级数据链路控制协议是点到点和多点配置都能使用的一个通用协议，但点到点访问最通用的协议还是点到点协议（Point-to-Point Protocol，PPP）。如今，成千上万的因特网用户需要通过PPP来将他们的家庭电脑与因特网服务提供商的服务器连接起来。这些用户中的大多数有一个额外的调制解调器，通过提供物理层服务，它们通过电话线与因特网连接。但是，为了数据传输的控制和管理，在数据链路层也需要一个点到点协议。到目前为止，PPP是最通用的协议。

PPP提供以下服务：

1. PPP决定了设备间交换的帧的格式。
2. PPP决定了两台设备如何为链路的建立和数据的交换达成一致。
3. PPP决定了网络层数据如何封装到数据链路帧中。
4. PPP决定了两台设备如何互相鉴别。
5. PPP通过支持一系列不同的协议来提供多个网络层服务。
6. PPP提供多链路连接。
7. PPP提供网络地址配置。当一个家庭用户连与因特网连接需要一个临时地址时尤其有用。

另一方面，为了使PPP尽量简化，有些功能被删除了：

1. PPP不提供流量控制。发送方能一个接着一个地发送帧，而不关心接收方是否超负荷。
2. PPP对差错控制有一个很简单的机制。一个CRC字段用来检测差错。如果帧被损坏了，它便被无响应地丢弃；上层协议必须解决这一问题。缺乏差错控制和顺序编号，这可能会导致数据包被乱序接收。
3. 在多点配置中，PPP无法提供高级的地址分配机制来处理帧。

11.7.1 成帧

PPP是一个面向字节的协议。根据本章开始部分讨论的面向字节协议实现成帧。

帧的格式

图11.32显示了一个PPP帧的格式。每个字段的描述如下：



图11.32 PPP帧的格式

- 标记。一个PPP帧分别以位模式为01111110的一个字节标记开始和结束。虽然这个位模式与在HDLC中使用的相同，但是仍然存在一个很大的区别。即，PPP是一个面向字节的协议，而HDLC是一个面向位的协议。这个标记作为字节处理，我们随后会加以解释。
- 地址。在这个协议中地址字段是常量11111111（广播地址），协商时双方可同意省略这个字节。
- 控制。这个字段被设定为常量11000000（效仿HDLC中的无编号帧）。正如我们随后会阐述的那样，PPP不提供任何流量控制。关于差错检测方面的差错控制也是有限的。这就意味着这个字段是根本不需要的，而且在协商时，双方能达成一致忽略这个字节。
- 协议。协议字段指明了数据字段承载着什么：用户数据或者其他信息。我们将简短但详细地阐述这个字段。这个字段被默认为2个字节，但是双方可以同意只用一个字节。
- 有效载荷字段。这个字段承载着用户数据或其他信息，我们将简短的加以讨论。数据字

段中有一系列的字节，最大数量的默认值是1 500字节，但是通过协商这是可以变化的。如果标记字节模式出现在数据字段的话，数据字段就是字节填充（转义）的。因为没有字段来界定数据字段的大小，如果数据字段的大小小于默认的最大值或者协商的最大值的话，就需要填充了。

- 帧校验序列。帧校验序列是简单的2到4字节标准CRC。

字节填充

正如我们之前讨论的那样，PPP和HDLC的帧格式的结尾有相似之处，但是前者是面向字节的协议，这点完全区别于后者。作为一个面向字节的协议，PPP的标记是一个字节，而且只要它出现在帧的数据部分，就需要被转义。转义字节是01111101，表明每次类似标记模式的字节出现在数据部分，都要填充这个额外的字节来告知接收方接下来的字节并不是标记。

> PPP是一个面向字节的协议，通过转义字节01111101来进行字节填充。

11.7.2 传输阶段

一个PPP连接可以通过转换阶段（transition phase）图来说明（见图11.33）。

- 闲置。在闲置阶段，链路没有被使用，没有活动的载体（在物理层）且线路是静默的。
- 建立。当一个节点开始通信，连接就进入这个阶段，在这个阶段，双方就选项进行协商。如果协商成功，系统就进入鉴别阶段（如果鉴别是需要的话）或者直接进入联网阶段。链路控制协议信息分组的应用就是以此为目的的，此处交换了一些信息包。
- 鉴别。鉴别阶段是可选的，两个节点在建立阶段可以决定是否跳过该阶段。然而，如果它们决定使用鉴别，它们会发送一些鉴别信息分组，我们以后会加以讨论。如果结果是成功的，连接进入联网阶段，否则，它将进入终止阶段。
- 联网。在联网阶段，将进行网络层协议的协商，因为PPP在网络层是支持多协议的，故PPP规定在网络层的数据进行交换之前，节点双方必须达成一个网络协定。如果一个节点在网络层同时运行多个协议的话，接收方需要知道哪个协议用来接收数据。
- 打开。在打开阶段，将进行数据传输。当一个连接到了这个阶段，数据分组的交换就可以开始了。在这个阶段，连接将持续直到其中一个终端希望终止这次连接为止。
- 终止。在终止阶段，终止连接。两端交换一些分组以消除和关闭链路。

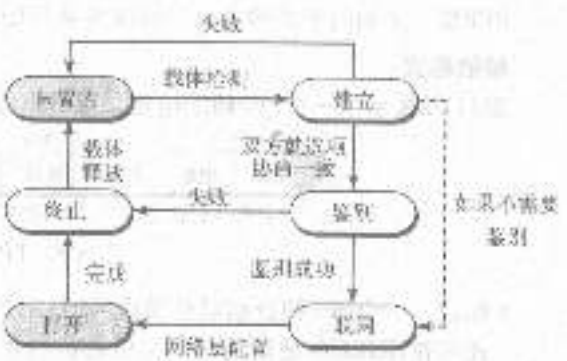


图11.33　转换阶段

11.7.3 多路复用

虽然PPP是一项数据链路层协议，但PPP也使用其他一组协议来建立链路，对涉及的参与者加以鉴别，并承载网络层数据。定义了三组协议使得PPP更为强大：链路控制协议（LCP）、两个鉴别协议（AP）和多个网络控制协议（NCP）。任何时刻，一个PPP信息分组都能在它的数据字段中携带来自这些协议的数据，如图11.34所示。注意，其中有一个链路控制协议、两个鉴别协议和多个网络控制协议。数据也可能来自不同的网络层。

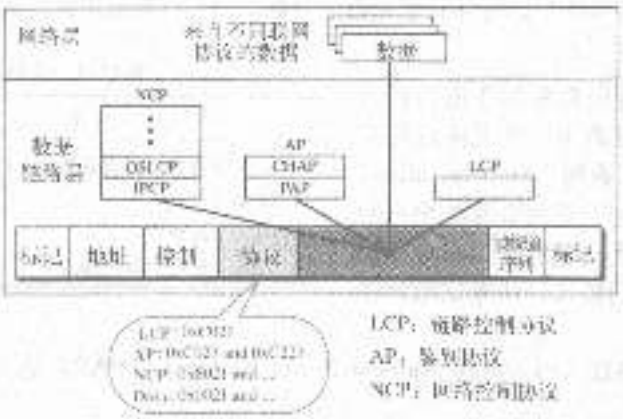


图11.34 PPP中的多路复用

链路控制协议

链路控制协议（Link Control Protocol，LCP）的职责是建立、维系、确认并终止链路。它还提供协商机制来设置两个端点间的选项。在链路建立之前，链路的两个端点必须就选项达成一个协定。见图11.35。



图11.35 封装在一个帧中的LCP分组

所有的LCP分组都位于PPP帧的有效载荷字段中，该帧的协议字段被设定为十六进制的C021。

编码字段决定了LCP分组的类型，如表11.2所示，共有11种类型的信息分组。

表11.2 LCP分组

编 码	分组类型	说 明
0x01	配置请求	包含建议选项及其值的列表
0x02	配置确认	接受所有建议的选项
0x03	配置不确认	告知某些选项不能被接受
0x04	配置拒绝	告知某些选项不可识别
0x05	终止请求	请求关闭线路
0x06	终止确认	接受关闭请求
0x07	编码拒绝	告知一个未知编码
0x08	协议拒绝	告知一个未知协议
0x09	回声请求	检测另一端是否活动的一种可选报文
0x0A	回声应答	对回声请求报文的响应
0x0B	丢弃请求	丢弃分组的请求

分组可以分为三类。第一类，包括前四种分组类型，用于建立阶段的链路配置。第二类，包括类型5和6，在终止阶段用于链路终止。后五种分组用于链路管控和调试。

ID字段持有一个将请求和应答相匹配的值。一个端点将一个值插入这个字段，这个值便被复制到应答分组中。长度字段决定了整个LCP分组的长度。为了满足某些LCP分组的需求，信息字段包含诸如选项等信息。

在两个端点之间有许多能被协商的选项。选项插入配置分组的信息字段。这种情况下，信

息字段被分为三段：选项类型，选项长度和选项数据。表11.3列示了其中最常用的一些选项。

表11.3　常用选项

选　项	默认值
最大接收单位（有效载荷字段大小）	1 500
鉴别协议	无
协议字段压缩	关闭
地址和控制字段压缩	关闭

鉴别协议

在PPP中鉴别具有很重要的作用，这为PPP被设计用于拨号链路中，这就使得用户身份的鉴别十分必要。鉴别（Authentication）意味着使一个需要访问一系列资源的用户的身份合法化。PPP有两种鉴别协议：口令鉴别协议和查询握手鉴别协议。注意这两种协议均用于鉴别阶段。

PAP　口令鉴别协议（Password Authentication Protocol，PAP）是一个两步骤的简单鉴别过程。

1. 需要访问系统的用户发送一个鉴别身份（通常是用户名）和一个口令。

2. 系统检测身份和口令的合法性，并选择接收连接或拒绝连接。

图11.36显示了PAP使用的三种类型的分组以及它们如何交换。当一个PPP帧承载PAP分组时，协议字段的值是0xC023。这三个PAP分组分别是鉴别-请求，鉴别-确认和鉴别-不确认。第一个分组被用户用来发送用户名和口令。系统用第二个分组来表示准许访问。系统用第三个分组来表示拒绝访问。



图11.36　封装入一个PPP帧中的PAP分组

CHAP　查询握手鉴别协议（Challenge Handshake Authentication Protocol，CHAP）是一个三步握手鉴别协议，比PAP更具安全性。在这种方式下，口令是保密的，无需上线发送。

1. 系统发送给用户一个包含查询值（通常是一些字节）的查询分组。

2. 用户应用一项预定义功能，根据查询值和用户自身的口令产生一个结果。用户将结果放入一个响应分组发送给系统。

3. 系统同样操作。它应用同样的功能，使用户的口令（系统已知的）和查询值产生一个结果。如果这个结果与响应分组中发送的结果相同，访问被允许；否则，访问被拒绝。CHAP比PAP安全得多，特别是当系统不断地更换查询值时，甚至当入侵者得知查询值和结果时，口令始终是保密的。图11.37显示了分组及其如何被使用。

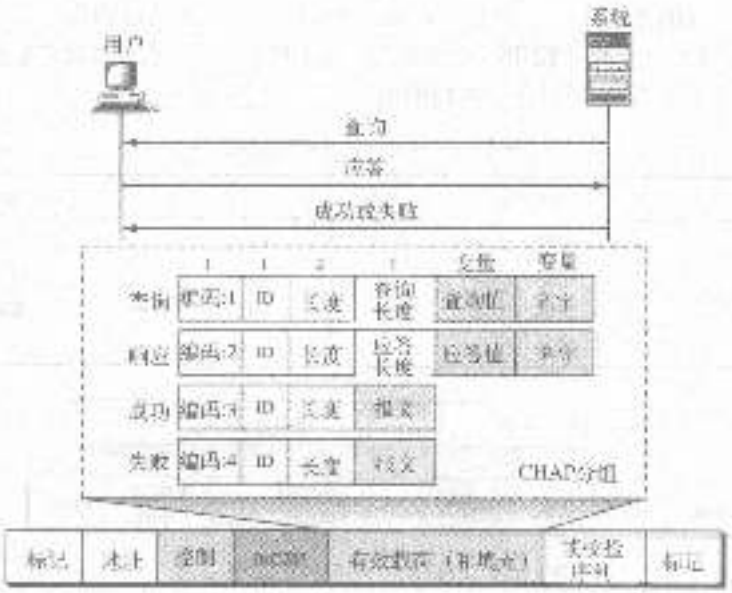


图11.37 封装入一个PPP帧中的CHAP分组

CHAP分组以十六进制的协议值C223封装在一个PPP帧中。有四种CHAP分组：查询、响应、成功或失败。第一种分组被系统用来发送查询值。第二种分组被用户用来返回计算结果。系统用第三种分组来表示准许访问，而用第四种分组来表示拒绝访问。

网络控制协议

PPP是一个多网络层的协议。它能使用两特网限定的协议，如OSI、Xerox、DECnet、AppleTalk、Novel等来运载网络层数据分组。为此，PPP为每一个网络协议都界定了一个特定的网络控制协议。例如，IPCP（互联网络协议控制协议）配置用来承载IP数据分组的链路。Xerox CP配置用来承载Xerox协议数据分组的链路等。注意，没有一个网络控制协议承载网络层数据，它们仅仅是在网络层为进入的数据配置链路。

IPCP　互联网络协议控制协议（Internet Protocol Control Protocol，IPCP）是一种网络控制协议。这个协议配置用于承载点特网中IP分组的链路。IPCP对我们而言有特殊的重要性。图11.38显示了一个IPCP分组的格式。注意协议字段的值是十六进制的8021。



图11.38 封装入一个PPP帧的IPCP分组

IPCP定义了七个分组，通过不同编码值加以区分，如表11.4所示。

其他协议　还有针对其他网络层协议的NCP协议。OSI的网络层控制协议的协议字段值是8023；Xerox NS IDP 控制协议的协议字段值是8025，诸如此类等。其他协议的编码值和分组的格式与表11.4所示的相同。

来自网络层的数据

当由一个NCP协议完成网络层确认后，用户可以交换来自网络层的数据分组。再次重申，

不同的网络层有不同的协议字段。例如，如果PPP运载来自IP网络层的数据，字段值就是0021（注意这个值最右边的三个数字跟IPCP的相同），如果PPP运载来自OSI网络层的数据，协议字段值就是0023，等等。图11.39显示了IP帧的格式。

表11.4　IPCP分组的编码值

编　码	IPCP分组	编　码	IPCP分组
0x01	配置－请求	0x05	终止－请求
0x02	配置－确认	0x06	终止－确认
0x03	配置－不确认	0x07	编码－拒绝
0x04	配置－拒绝		

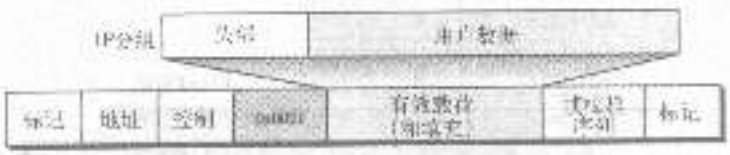


图11.39　封装在一个PPP帧的IP数据报

11.7.4　多链路PPP

PPP最初是为了单通道点到点的物理链路设计的。在单个点到点链路中多通道的实用性促使了多链路PPP的发展。在这种情况下，一个PPP逻辑帧被分割成多个实际的PPP帧。逻辑帧的一段由一个实际PPP帧的有效载荷字段所承载，如图11.40所示。为了显示一个实际PPP帧正承载一个PPP逻辑帧的一段，协议字段被设置为0x003d。这个新发展带来了复杂性。举例来说，必须在实际PPP帧上增加一系列数字以说明这部分在逻辑帧中的位置。

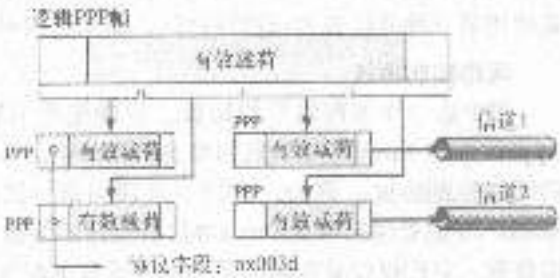


图11.40　多链路PPP

例11.12

让我们来看一下一个网络层分组通过PPP连接传输的各个阶段。图11.41描述了这些步骤。为了简化，我们假定数据从用户端向系统端单向运动（例如通过ISP发送一封邮件）。

前两个帧用作链路的建立。我们选择了两个选项（并没有显示在图中）：使用PAP来鉴别并隐匿地址控制字段。帧3和帧4用来鉴别。帧5和帧6使用IPCP建立了网络层连接。

接下来的几个帧显示了封装在PPP帧的一些IP分组。系统（接收方）已经运行了一些网络层协议，但是它知道进入的数据必须交付给IP协议，因为在数据传输前使用的NCP协议是IPCP。

数据传输之后，用户终止了数据链路连接，并且这被系统所确认。当然如果想要运行另一个NCP协议的话，用户和系统也可以选择终止网络层协议IPCP，而维持数据链路层的运行。

这个例子是微不足道的，但是它指出了LCP、AP和NCP分组的相似之处。它同样描述了协议字段值和特定协议的编码数字。

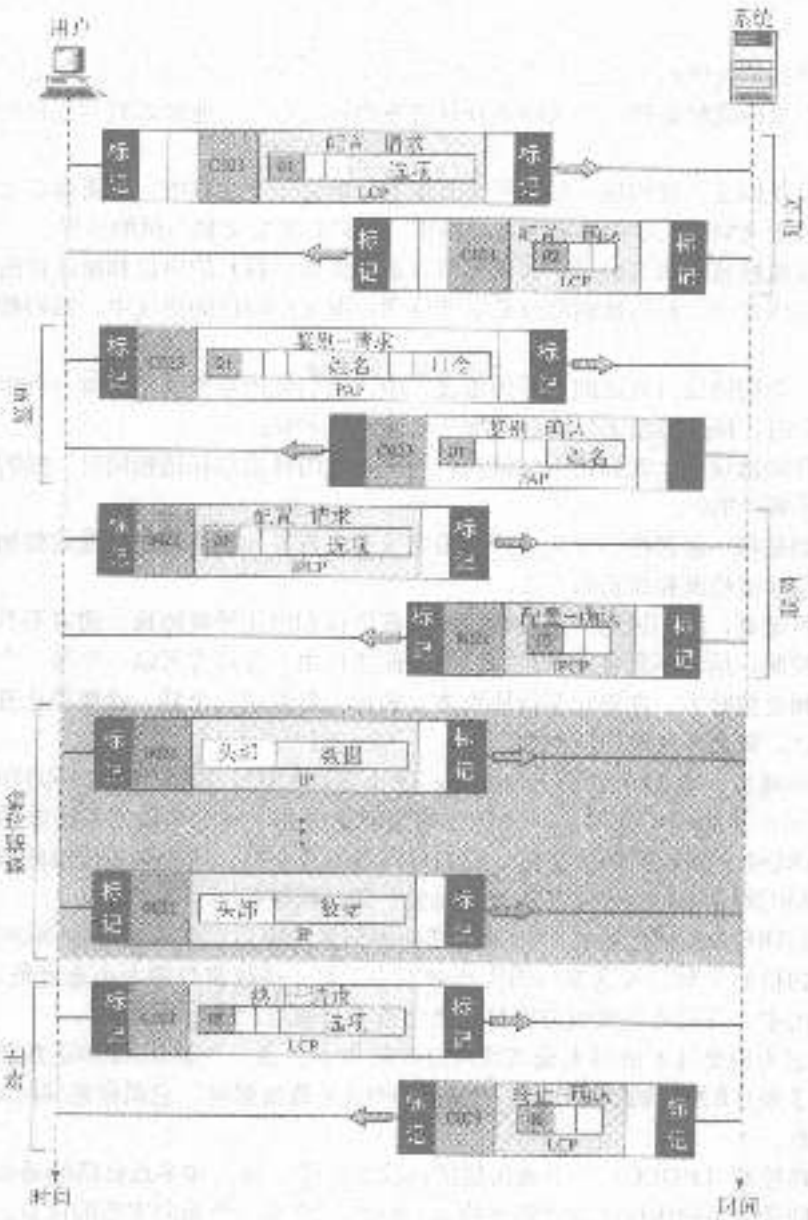


图11.41 一个例子

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物。在本书末列出括号[……]中的参考资料。

读物

数据链路控制的讨论可以在[GW04]、[Fan03]的第3章、[Sta04]的第7章、[Kes97]的第12章与[PD03]的第2章中找到。更进一步的材料可以在[KMK04]找到。

本章小结

- 数据链路控制处理两个邻近节点之间的通信（即节点到节点的通信）的设计和程序。
- 数据链路层的成帧是将一个从源方到目的方的报文与从其他源方到其他目的方的报文分离开来。
- 帧既可以是固定长度的也可以是可变长度的。固定长度成帧中，不需要定义帧的分界；在可变长度成帧中，我们需要一个分隔符（标记）来定义帧与帧的边界。
- 可变长度成帧使用两类协议：面向字节（或者面向字符）的协议和面向位的协议。在面向字节的协议中，帧的数据部分是字节序列；而在面向位的协议中，帧的数据部分是位序列。
- 在面向字节的协议（或面向字符的协议）中，我们使用字节填充，当一个字符的模式与标记相同时，帧的数据部分就会增加一个特定的字节。
- 在面向位的协议中，我们使用位填充，当一串位的模式与标记相同时，帧的数据部分就增加一个额外的0。
- 流量控制是指一组程序，它用来严格限定发送方在等到确认前能发送的数据量。差错控制指的是差错检测和纠正的方法。
- 对无噪声通道，我们讨论了两种协议：最简协议和停止等待协议。前者不具备差错控制和流量控制；后者不具备差错控制。在最简协议中，发送方可以一个接一个地发送帧而不需要顾及接收方。在停止等待协议中，发送方每发送一个帧，就要停止直到收到接收方的确认，随后再发送下一个帧。
- 对有噪声通道，我们介绍了三种协议：停止等待ARQ，回退N帧的ARQ和选择性重复ARQ。停止等待ARQ在停止等待协议的基础上增加了一个简单的差错控制机制。在回退N帧ARQ中，在收到确认之前，我们可以发送几个帧，这提高了传输的效率。在选择性重复ARQ中，我们避免了不必要的传输，除非帧被损坏了。
- 回退N帧ARQ和选择性重复ARQ两者都使用到滑动窗口。在回退N帧ARQ中，如果m是序列号的位的个数，发送窗口的大小必须小于2^m；接收窗口的大小永远是1。在选择性重复ARQ中，发送方和接收方窗口的大小最多只能是2^m的一半。
- 一项称之为捎带技术被用来提高双向协议的效率。当一个帧从A到B运载数据时，它同时运载了来自B的控制信息；当一个帧从B到A运载数据时，它同样能同时运载来自A的控制信息。
- 高级链路控制（HDLC）一个面向位的协议，用于点到点和多点链路的通信。然而，点到点访问最通用的协议还是点到点协议（PPP），它是一个面向字节的协议。

习题

复习题

1. 简要描述数据链路层所提供的服务。
2. 定义成帧及其原因。
3. 比较和对比面向字节的协议和面向位的协议。过去哪一种更为流行（解释原因）？而现在哪一个更流行（解释原因）？
4. 比较和对比字节填充和位填充。在面向字节的协议中使用哪一种技术？而面向位的协议又使用哪一种技术？
5. 比较和对比流量控制和差错控制。

6. 在本章中，我们介绍了无噪声通道中的哪两种协议？
7. 在本章中，我们介绍了有噪声通道中的哪三种协议？
8. 解释从停止等待ARQ协议发展到回退N帧ARQ协议的原因。
9. 比较和对比回退N帧ARQ协议和选择性重复ARQ协议。
10. 比较和对比HDLC和PPP，那一个是面向字节的，那一个面向位的？
11. 定义捎带技术及其用途。
12. 本章中利用流水线操作的协议是哪一个？

练习题

13. 对图11.42中的数据进行字节填充。



图11.42 练习题13

14. 对图11.43中的数据进行位填充。

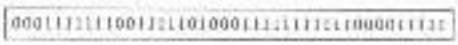


图11.43 练习题14

15. 为字节填充设计两个简单的算法。发送方使用前者添加字节，接收方使用后者去除字节。
16. 为位填充设计两个简单的算法。发送方使用前者添加位，接收方使用后者去除位。
17. 一个发送方发送一系列分组到同一个目的地，使用一个5位长度的序列号。如果序列号以0开始，发送一百个分组后序号是多少？
18. 使用一个5位序号，在下列协议中，发送和接收窗口的最大尺寸分别是多少？
    a. 停止等待ARQ　　b. 回退N帧ARQ
    c. 选择性重复ARQ
19. 为使用捎带的最简协议设计一个双向算法，注意双方都使用同一个算法。
20. 为使用捎带的停止等待协议设计一个双向算法，注意双方都使用同一个算法。
21. 为使用捎带的停止等待ARQ设计一个双向算法。注意双方都使用同一个算法。
22. 为使用捎带的回退N帧ARQ设计一个双向算法。注意双方都使用同一个算法。
23. 为使用捎带选择性等待ARQ设计一个双向算法。注意双方都使用同一个算法。
24. 图11.44显示了一个状态示意图来模拟停止等待ARQ中发送方站点的行为。该状态中S_n有一个值（0或1）。箭头表示传输。请解释标记为A和B的两个传输事件。
25. 图11.45显示了一个状态示意图来模拟停止等待ARQ中接收方站点的行为。该状态中R_n有一个值（0或1）。箭头表示传输。请解释标记为A和B的两个传输事件。

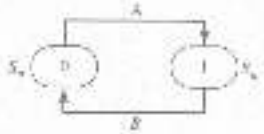


图11.44 练习题24

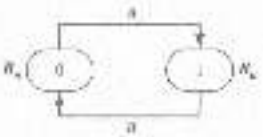


图11.45 练习题25

26. 在停止等待ARQ中，我们可以将练习题24和练习题25中发送方和接收方的算法综合起来。一个状态来界定R_n和S_n的值。这意味着我们有四种状态，由（x，y）决定，x指的是S_n的值，而y指的是R_n的值。换言之，图11.46中会有四种状态，请解释标记为A、B、C和D四个传

输事件。

27. 使用停止等待ARQ协议的系统的一个定时器有6微秒超时。为4个帧画一张类似图11.11的流量图，假定往返延迟是4微秒，且没有数据帧或者控制帧被丢失或损坏。

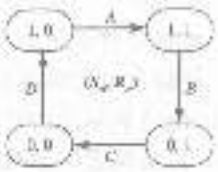


图11.46　练习题26

28. 如果练习题27中超时时间为4微秒，而往返延迟为6微秒，请画图。
29. 如果练习题27中第一个帧（帧0）被丢失了，请画图。
30. 一个系统使用停止等待ARQ协议。如果每个分组承载1 000位数据，发送方和接收方之间的距离是5 000公里，传输速度是2×10^8米，那么传输1百万位需要多长时间？忽略传输、等待和处理延迟，且我们假定没有数据帧和控制帧丢失或损坏。
31. 练习题30中，如果使用回退N帧ARQ协议，且窗口大小是7，需要多长时间？忽略用于头部和尾部的开销。
32. 练习题30如果使用选择性重复ARQ协议，且窗口大小是4，需要多长时间？忽略用于头部和尾部的开销。

第12章 多路访问

在第11章我们讨论了数据链路控制，它是为链路提供可靠通信的一种机制。在我们描述的协议中，我们假定在发送方和接收方之间存在可用的专用链路（或通道）。这个假定可能正确也可能不正确。事实上，如果我们有一条专用链路，就像我们连接因特网时使用PPP作为数据链路控制协议时，那么这个假定是正确的，我们就无需其他任何东西了。

另一方面，如果我们使用移动电话去与其他移动电话连接，通道（分配给售方公司的波段）并不是专用的。几步以外在用移动电话和他的朋友聊天的一个人就可能和我们使用同一个通道。

我们认为数据链路层有两个子层，上子层负责数据链路控制，下子层负责解决对共享介质的访问。如果通道是专用的，我们并不需要下子层。图12.1显示了在数据链路层中的两个子层。

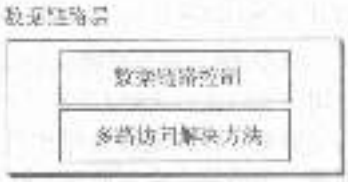


图12.1 数据链路层被分为两个功能定位的子层

在第13章中，我们将会看到IEEE确实为LAN做了如此的分割。上子层负责流量控制和差错控制，被称为逻辑链路控制层（LLC），下子层主要负责多路访问的解决方法，被称为介质访问控制层（MAC）。

当节点或站点被连接并使用一个称为多点链路或广播链路的公共链路时，我们就需要一个多路访问协议去协调链路的访问。介质访问的控制问题与在集会上发言的规则相似。有程序来为发言授权并保证没有两个人会在同一时间发言，发言者不会打扰到彼此，不会独占发言权等。

多点网络的情况与此相似。许多正式的协议被设计用来处理共享链路的访问，我们将这些协议分为三类，如图12.2所示。



图12.2 本章讨论的多路访问协议的分类

12.1 随机访问

在随机访问（random access）或竞争（contention）访问方法中，没有任何站点是优于其他站点的，也没有任何站点能控制其他站点。一个站点没有权利允许或者不允许另一个站点的发送。每一次，当一个站点要发送数据时，都要用协议定义的程序来决定是否要发送。这个决定取决于介质的状态（空闲或忙碌）。换言之，只要遵循预定义的程序，包括介质状态的检测，满足条件的每一个站点都能传输数据。

该方法之所以如此命名因为两个特性。第一，每一个站点的传输没有特定的时间表。站点的传输是随机的。这就是该方法称为随机访问的原因。第二，没有规则来规定下一个将要发送的站点是哪一个。为了访问介质，站点展开竞争。这就是该方法称为竞争访问的原因。

在随机访问中，每个站点都有权访问介质且不受控于其他站点。然而，如果当一个以上的站点想要发送时，便会产生冲突（collision），那么便会有帧被损坏或修改。为了避免访问

冲突或在访问冲突发生时解决这个问题，每个站点遵循一个程序。该程序回答了以下问题：

- 站点何时能访问介质？
- 如果介质忙碌，站点能做什么？
- 站点如何确定传输的成功或者失败？
- 如果发生访问冲突，站点能做什么？

本章中我们学习的随机访问方法是由一个十分有趣的称之为ALOHA的协议发展而来的，ALOHA使用一个十分简单的程序称为多路访问（multiple access，MA），该方法是外加一种程序（就是在站点传送之前要对介质进行侦听）改进而来。它称为载波侦听多路访问（CSMA）。该方法后来又演化成两个并行的方法：带冲突检测的载波侦听多路访问（carrier sense multiple access with collision detection，CSMA/CD）和带避免冲突的载波侦听多路访问（carrier sense multiple access with collision avoidance，CSMA/CA）。CSMA/CD告诉站点一旦检测到冲突，下一步该做什么，而CSMA/CA尝试如何避免冲突。

12.1.1　ALOHA

ALOHA是最早的随机访问方法，它由夏威夷大学在20世纪70年代初期开发出来。它被设计用于无线电（无线）LAN，但也可以使用在任何共享介质上。

很明显，在这种安排中有一些潜在的冲突。介质为各个站点所共享，当一个站点发送数据时，其他站点也可能在同时尝试发送，两个站点的数据相互冲突并且相互混淆。

纯ALOHA

最早的ALOHA协议称为纯ALOHA（pure ALOHA），这是一个简单的但优秀的协议。它的概念是只要每个站点有帧要发送，它就发送帧。然而，既然只有一条共享通道，不同站点之间的帧就有可能发生冲突。图12.3描述了一个在纯ALOHA中帧冲突的例子。

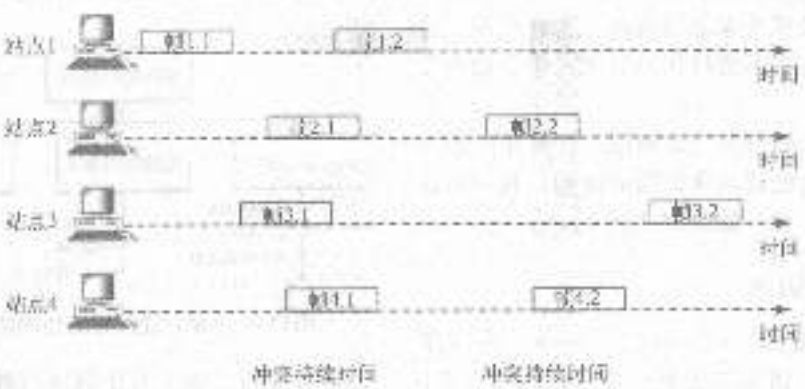


图12.3　纯ALOHA网络中的帧

图中有4个站点（非现实的假设）为访问共享通道彼此竞争。每个站点发送两个帧，共享介质中共有8个帧。在为共享通道展开的多帧竞争中，某些帧冲突了。图12.3中只有两个帧是成功的：来自站点1的帧1.1和来自站点3的帧3.2。必须要提醒的是，只要一个帧的一个位与另一个帧的一个位同时存在于通道中，就会产生冲突且两个帧都会被损坏。

很明显，我们需要重发在传输过程中损坏的帧。纯ALOHA协议依赖于接收方的确认。当一个站点发送一个帧时，它就等待接收方发送一个确认。如果超时之后确认还未达到，站点就认为帧（或者确认）被损坏了，于是重发帧。

一个冲突涉及两个或更多的站点，如果这些站点在超时后都尝试重发帧，那么这些帧可能再次发生冲突。纯ALOHA指定超时后，每个站点在重发帧之前都会随机等待一段时间。这个随机事件有助于避免更多的冲突。我们称这个时间为补偿时间T_b。

纯ALOHA还需要第二个方法来阻止重新传输的帧造成通道拥塞。一个站点在经过最大次数（K_{max}）的重发尝试后，它必须放弃并在以后再试。图12.4描述了基于上述策略的纯ALOHA进程。

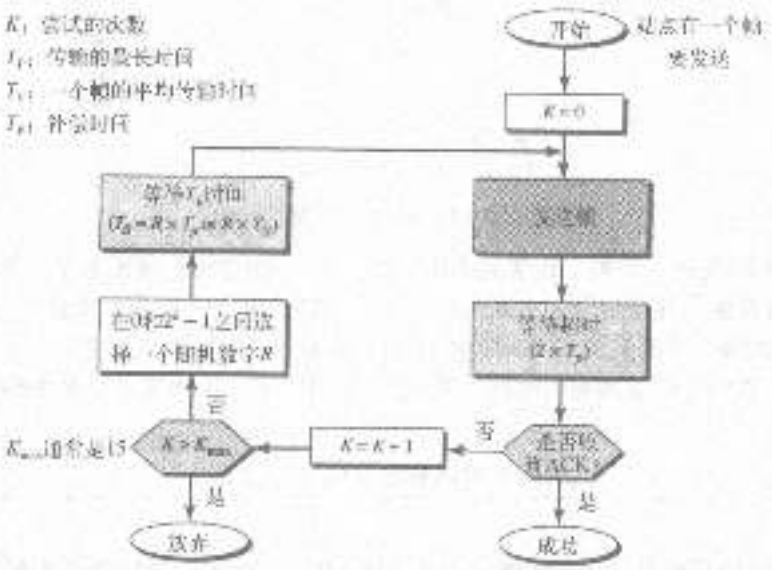


图12.4 纯粹ALOHA协议的进程

超时周期等于往返传输延迟的最大可能值，也就是两个最远距离站点之间发送一个帧所需时间的两倍（$2\times T_p$）。补偿时间T_b是一个随机值，主要取决于K（未成功的传输尝试次数）的值。T_b的公式取决于实现。一个通用公式是二进制指数回退（binary exponential back-off）。这种方法中，每一次传输随机选择0到2^K-1范围内的一个数。该数乘以T_p（传输的最长时间）或乘以T_{fr}（一个帧的平均传输时间）以得到T_b的值。注意在这个进程中，每一次冲突后随机数的范围便扩大了。K_{max}的常用值是15。

例12.1

一个无线ALOHA网络中的站点之间的最远距离是600公里。如果我们假设信号的传播速度是3×10^8米/秒，那么$T_p=600\times10^3/(3\times10^8)=2$毫秒。于是，对于不同的$K$值，我们可以得到不同的$T_b$值。

a. 如果$K=1$，取值范围是{0，1}。站点会产生一个随机变量。变量值为0或1。这就意味着T_b是0（0×2）毫秒或2（1×2）毫秒，取决于产生的随机变量。

b. 如果$K=2$，取值范围是{0，1，2，3}。这就意味着T_b是0，2，4或6毫秒，取决于产生的随机变量。

c. 如果$K=3$，取值范围是{0，1，2，3，4，5，6，7}。这就意味着T_b是0，2，4，……14毫秒，取决于产生的随机变量。

d. 要提醒的一点是，如果$K>10$，它通常被设定为10。

脆弱时间 我们来计算可能发生冲突时间的长短，即脆弱时间（vulnerable time）的长短。假设站点发送固定长度的帧，且每个帧的发送时间是T_{fr}秒。图12.5显示了站点A的脆弱时间。

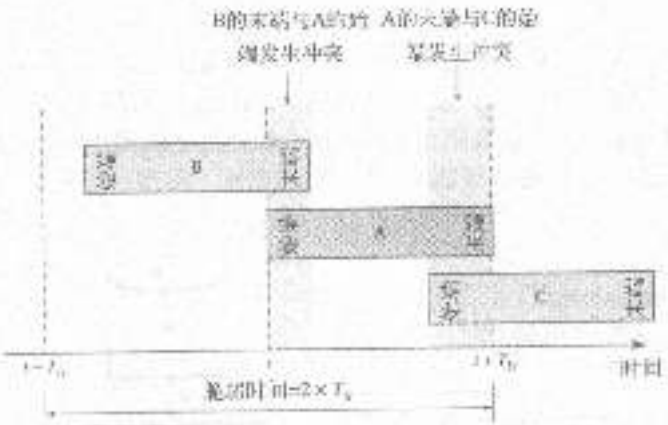


图12.5　纯ALOHA协议的脆弱时间

站点A在时刻t发送一个帧。设想站点B在时刻$t-T_{fr}$和t之间已经发送了一个帧，B帧的末端与A帧的始端发生了冲突。另一方面，假定站点C在时刻t和$t+T_{fr}$之间发送了一个帧。此时，站点A和站点C的帧之间发生了冲突。C帧的始端与A帧的末端发生了冲突。

图12.5中，我们可以看到脆弱时间，即纯粹ALOHA中一个冲突可能发生的时间是帧传输时间的2倍。

纯ALOHA脆弱时间 $= 2 \times T_{fr}$

例12.2

一个纯ALOHA网络在一个传输速度为200kbps的共享通道上传输200位的帧。使这个帧在传输过程中避免冲突要达到什么样的条件？

解：

帧的平均传输时间T_{fr}是200比特/200kbps，即1毫秒。脆弱时间是2×1毫秒$=2$毫秒。这就意味着在该站点开始传输前1毫秒，没有站点会开始传输，并且当该站点正在传输的1毫秒期间也没有站点会开始传输。

吞吐量　我们称G为一个帧传输时间内系统产生的帧的平均数量。能证明纯ALOHA成功传输帧的平均数量是$S=G \times e^{-2G}$。当$G=1/2$，得到最大吞吐量S_{max}是0.184，换言之，在一个帧的传输时间内产生二分之一个帧（即两个帧的传输时间产生一个帧），这些帧的18.4%能成功到达目的地。因为脆弱时间是帧传输时间的2倍，故这是一个可预期的结果。因此，如果一个站点在脆弱时间内只产生一个帧（且在这段时间内没有其他站点产生帧），这个帧便能成功地到达它的目的地。

纯ALOHA的吞吐量是$S=G \times e^{-2G}$，当$G=(1/2)$时，最大吞吐量$S_{max}=0.184$。

例12.3

一个纯ALOHA网络在一个传输速度为200kbps的共享通道上传输200位的帧。如果系统（包括所有站点）产生以下数量的帧，吞吐量将是多少？

a. 每秒1 000个帧

b. 每秒500个帧

c. 每秒250个帧

解：

帧的传输时间是200比特/200kbps，即1毫秒。

a. 如果系统每秒产生1 000个帧，也就是每毫秒产生一个帧，负载为1。在这种情况下，即$S=0.135$ (13.5%)。也就是说，吞吐量是$1\ 000\times0.135=135$帧。1 000个帧中只有135个可能成功。

b. 如果系统每秒产生500个帧，也就是每毫秒产生1/2个帧，负载为1/2。在这种情况下，即$S=0.184$ (18.4%)。也就是说，吞吐量是$500\times0.184=92$帧。500个帧中只有92个可能成功。注意这就是最大吞吐量百分比的例子。

c. 如果系统每秒产生250个帧，也就是每毫秒产生1/4个帧，负载为1/4。在这种情况下，即$S=0.152$ (18.4%)。也就是说，吞吐量是$250\times0.152=38$帧。250个帧中只有38个可能成功。

时隙ALOHA

纯ALOHA的脆弱时间是$2\times T_t$。这是因为没有规则来定义站点何时能发送。一个站点可能在另一个站点已经开始发送之后马上发送，也可能在另一个站点完成发送之前马上发送。时隙ALOHA的发明发送了纯ALOHA的效率。

在时隙ALOHA（slotted ALOHA）中，我们把时间分割成T_t秒的时隙，并强制站点只有在时隙开始之时才能发送。图12.6描述了一个在时隙ALOHA中帧冲突的例子。

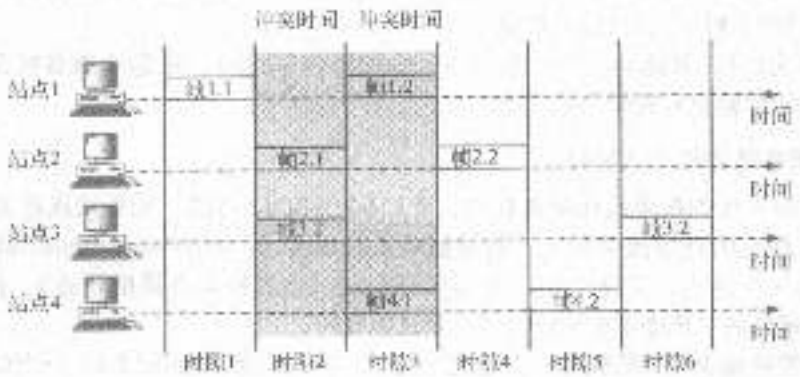


图12.6 时隙ALOHA中的帧

因为只允许一个站点在同步时隙的开始时发送帧，如果一个站点错过了这个时刻，就要等到下一个时隙的开始之时。这就意味着在时隙开始之时，发送帧的站点在该时隙就已经完成了发送。当然，如果两个站点试图在同一个时隙的开始之时发送帧，那么仍然存在冲突的可能。无论如何，现在脆弱时间减少了一半，如图12.7所示，它等于T_t。

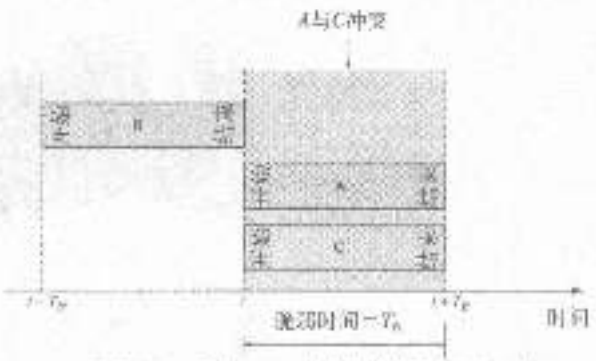


图12.7 时隙ALOHA协议中的易受攻击时间

图12.7说明了时隙ALOHA的脆弱时间只是纯ALOHA的一半。

时隙ALOHA的脆弱时间 = T_t

吞吐量 能证明时隙ALOHA成功传输帧的平均数量是$S=G\times e^{-G}$。当$G=1$时，得到最大吞吐量S_{max}是0.368。换言之，当一个帧传输时间内有一个帧产生的话，这些帧的36.8%会成功达到它们的目的地。这个结果是能被

预知的，因为脆弱时间等于帧传输时间。因此，如果一个站点在脆弱时间内只产生一个帧（且在这个时段内没有其他站点产生帧），这个帧便能成功地到达它的目的地。

时隙ALOHA的吞吐量是$S = G \times e^{-G}$。当$G = 1$时，最大吞吐量$S_{max} = 0.368$

例12.4

一个时隙ALOHA网络使用一个带宽为200kbps的共享通道发送200比特的帧。如果系统（包括所有站点）产生以下数量的帧，吞吐量将是多少？

a. 每秒1 000个帧

b. 每秒500个帧

c. 每秒250个帧

解：

此情况与前一个练习相似，除了网络使用时隙ALOHA来代替纯ALOHA之外。帧传输时间是200/200kbps，即1微秒。

a. 此例中G为1，因此$S = G \times e^{-G}$，即$S = 0.368$（36.8%）。这意味着吞吐量是1 000 × 0.368 = 368帧。1 000个帧中只有368个能成功。注意这就是最大吞吐量百分比的例子。

b. 此例中G为1/2，因此$S = G \times e^{-G}$，即$S = 0.303$（30.3%）。这意味着吞吐量是500 × 0.303 = 151帧。500个帧中只有151个能成功。

c. 此例中G为1/4，因此$S = G \times e^{-G}$，即$S = 0.195$（19.5%）。这意味着吞吐量是250 × 0.195 = 49帧。250个帧中只有49个能成功。

12.1.2 载波侦听多路访问（CSMA）

为了减少冲突发生的概率从而提高性能，我们采用CSMA方法，如果站点在发送前先侦听一下介质，其冲突的概率就会减少。载波侦听多路访问（Carrier sense multiple access，CSMA）要求每一个站点在发送之前首先要监听介质（或者检查介质的状态）。换句话说，CSMA基于的原理就是“传输前先侦听”或“说话前先听”。

虽然CSMA能够减少冲突的概率，但是它不能消除冲突。原因如图12.8，一个CSMA网络的时空模型中所示，站点与一个共享通道连接（通常是一个专用介质）。

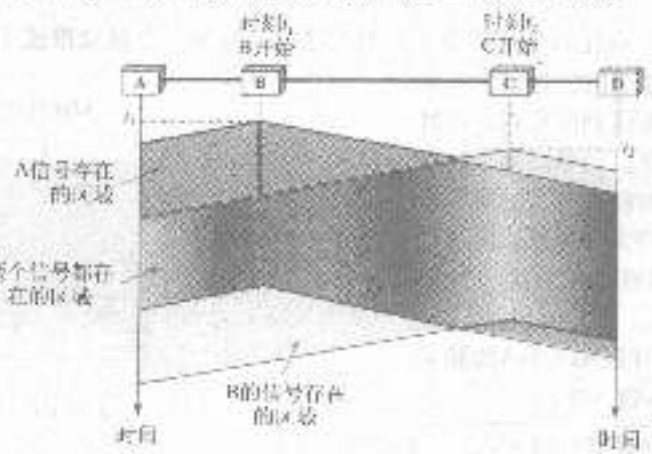


图12.8　CSMA中冲突的时空模型

冲突的概率依然存在的原因是因为传播的延迟。当站点发送一帧时，它的第一个位到达每一个站点和每一个站点检查到它的信号都需要一定的时间（尽管这一时间非常之短）。换句

话说，一个站点之所以能够侦听到介质并发现它处于空闲状态，仅仅是因为其他站点的传播信号还没有到达该站点。

在时刻t_1，站点B侦听介质并发现介质处于空闲状态，所以它发送一个帧。在时刻t_2($t_2>t_1$)，站点C侦听介质并发现它处于空闲状态，因为这时从站点B发来的第一个位还没有到达站点C。站点C也发送一个帧。于是这两个信号冲突且两个帧同时被损坏了。

脆弱时间

CMSA的脆弱时间是传播时间T_p。这是一个信号从介质的末端到另一端的传播时间。当一个站点发送一个帧，且其他站点也试图在这个时间段内发送帧时，就会发生冲突。但是，如果帧的第一个位已经到达了介质的末端，其他每个站点都能"听到"这个位，并停止发送。图12.9显示了最差的状况。最左边的站点A在时刻t_1发送一个帧，它在时刻t_1+T_p到达最右边的站点D。灰色区域显示了时间上和空间上的脆弱区域。

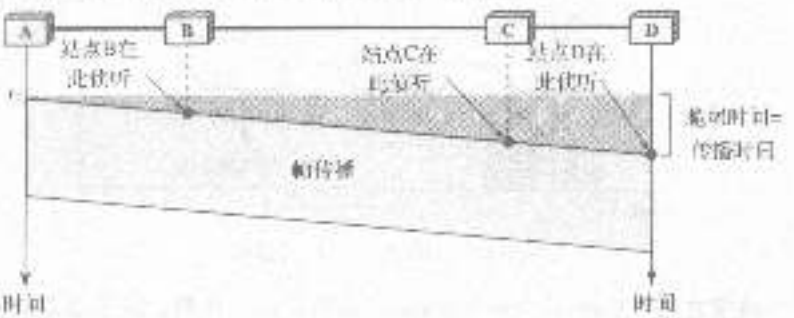


图12.9 CMSA的脆弱时间

持续方法

如果通道繁忙，站点将如何做？如果通道空闲，站点又将做什么？有三种方法被设计用来回答这些问题：1-持续方法，非持续方法和p-持续方法，图12.10描述了当一个站点发现通道繁忙时三种持续方法的表现。

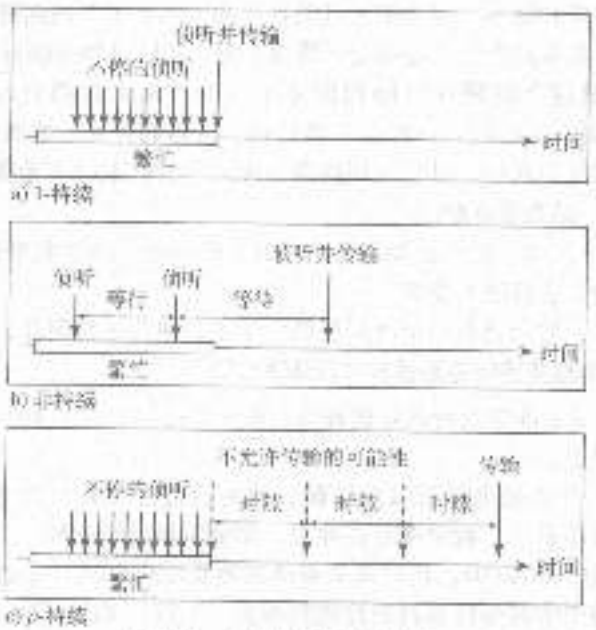


图12.10 三种持续方法的表现

图12.11描述了这些方法的流量图。

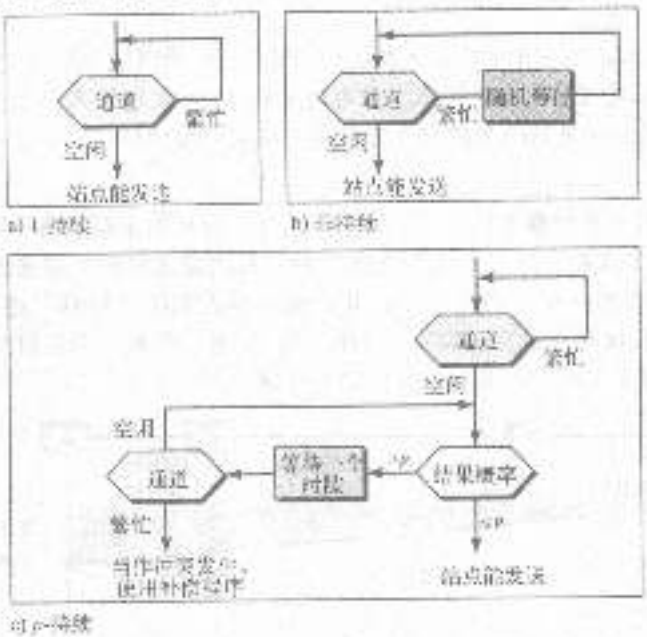


图12.11　三种持续方法的流量图

1-持续　1-持续方法（1-persistent method）是最简单直接的，该方法是，如果站点发现线路空闲，那么它就立即发送帧（概率是1）。这种方法发生冲突的机率最高，因为在发现了线路空闲后，两个或更多的站点都可以发送它们的帧。在第13章中，我们将看到以太网使用该方法。

非持续　在非持续方法（nonpersistent method）中，要发送帧的站点侦听线路，如果线路空闲，它就立即发送。如果线路不空闲，站点就等待一个随机时间，然后再次侦听线路。非持续策略可以减少冲突的概率，因为不太可能存在两个或更多个站点等待相同的时间并且再次立即重新尝试。因为当有站点要发送帧而介质却空闲，所以这种方法降低了网络的效率。

p-持续　如果通道有时隙且时隙周期等于或大于最大传播时间时，使用p-持续方法（*p*-persistent method），*p*-持续方法综合了其他两个策略的优点。它既减少了冲突的机会，又提高了网络效率。在此方法中，站点发现线路空闲后它会采取以下步骤：

1. 如果概率是*p*，站点发送帧。
2. 如果概率是$q=1-p$，站点会等待下一个时隙的开始并再次检查线路。
   a. 如果线路空闲，它将进入步骤1。
   b. 如果线路繁忙，它会当做一个冲突已经发生了并使用补偿程序。

12.1.3　带冲突检测的载波侦听多路访问（CSMA/CD）

CSMA方法没有规定冲突之后的处理程序。带冲突检测的载波侦听多路访问（CSMA/CD）提出了处理冲突的算法。

在这一方法中，任意站点都可以发送帧，然后就监控介质看传送是否成功。如果成功，站点完成发送。如果不成功，就说明存在冲突，需要重新发送此帧。

为了更好地理解CMSA/CD，我们来看看冲突涉及的两个站点发送的第一个位的情况。每个站点都继续发送帧中的其他位直到它检测到冲突，图12.12描述了第一个位发生冲突的情况。冲突的双方是站点A和站点C。

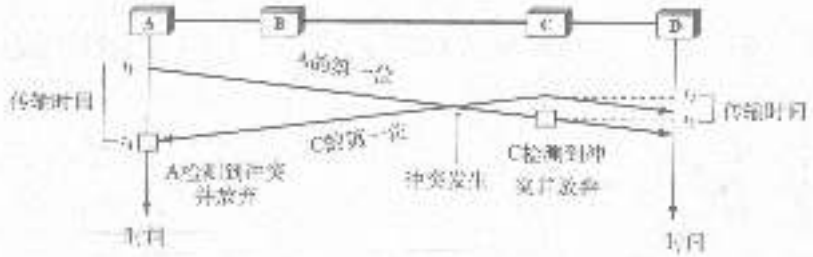


图12.12 在CMSA/CD中第一位冲突的情况

在时刻t_1，站点A执行它的持续程序并开始发送帧。在时刻t_2，站点C还未侦听到A发送的第一个位。站点C执行它的持续程序并开始发送帧。都是从左向右传输的，在时刻t_2后的某一时刻，冲突发生了。在时刻t_3，当站点C收到A的第一个位时，它检测到冲突。站点C马上（或者一小段时间后，但我们假设为马上）放弃了传输。在时刻t_4，当站点A收到C的第一个位时，它检测到冲突。它也同样马上放弃了传输。在图12.12中，我们可以看到站点A进行传输的持续时间是t_4-t_1；站点C进行传输的持续时间是t_3-t_2。随后我们可以看到，因为这个协议，在该协议中任何被比特率所分割的帧的长度都必须大于这些持续的时间段。在时刻t_4，虽然A帧的传输还未完成，但却被放弃了；在时刻t_3，虽然C帧的传输还未完成，但却也被放弃了。

现在我们明白了这两个传输的持续时间，图12.13显示了一张更为完整的图。

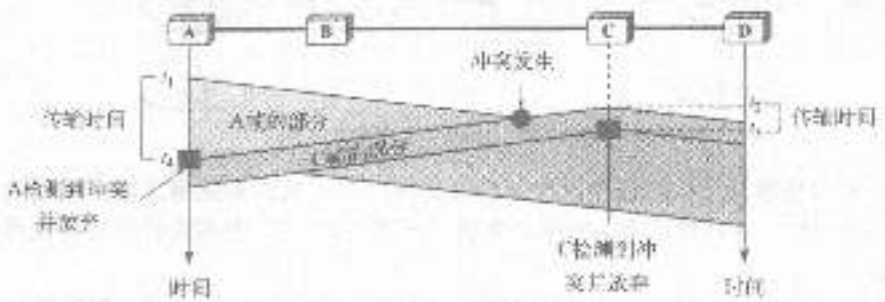


图12.13 在CSMA/CD中的冲突和放弃传输

帧的最小长度

为了实现CSMA/CD，需要限定帧的长度。在发送帧的最后一位前，发送站点必须检测冲突，如有任何冲突要放弃传输。这是因为一旦整个帧被发送了，站点就无法保留帧的副本并无法掌控线路的冲突检测。因此，帧传输时间T_f必须至少是最大传播时间T_p的两倍。为了理解这个原由，让我们来考虑最差的情况。如果两个站点卷入一个冲突中，且这两个站点距离最远，信号从第一个站点到达第二个站点要花费时间T_p。冲突的作用使得到达第一个站点还需要花费时间T_p。因此，便要求第一个站点在$2T_p$时间后仍在进行传输。

例12.5

使用CMSA/CD的一个网络，带宽是10Mbps。如果最大传播时间（包括设备的延迟并忽略发送一个干扰信号所需的时间）是25.6μs，那么帧的最小长度是多少？

解：

帧的传输时间是$T_f=2\times T_p=51.2\mu s$。这就是说，最差的情况下，一个站点需要传输51.2μs后才能检测到冲突。帧的最小长度是10Mbps×51.2μs=512比特或64字节。这是标准以太网中帧的最小尺寸，在第13章中会有介绍。

程序

现在让我们看一下图12.14中的CMSA/CD流量图。它与ALOHA协议的流量图相似，但有在着不同。

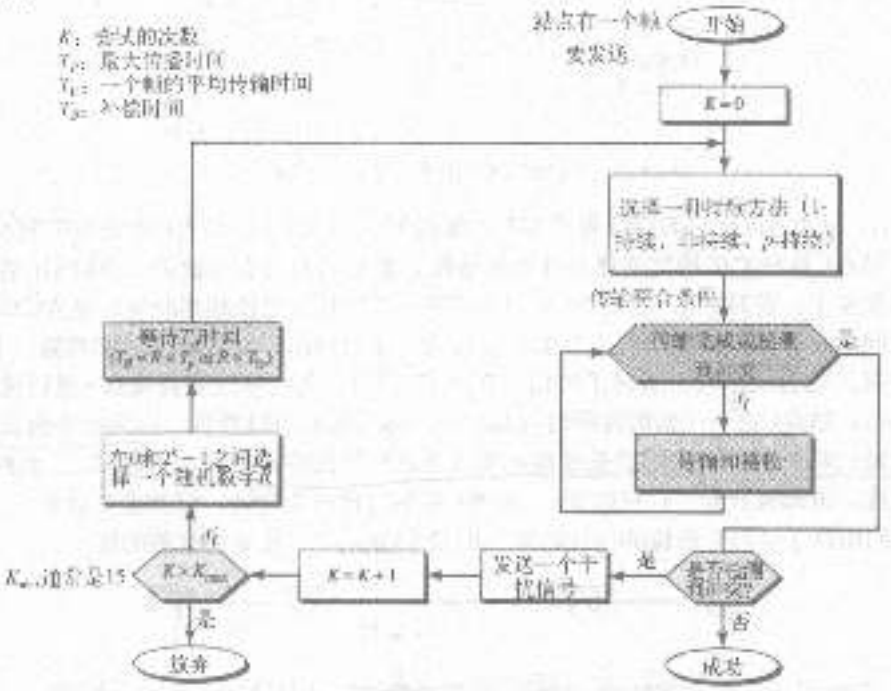


图12.14　CSMA/CD的流量图

第一个不同是增加了持续程序。在开始发送帧之前，我们要使用之前讨论过的持续程序中的一种（1-持续，非持续或者p-持续）来侦听通道。图12.11中相关的型形将被持续程序中的一种所代替。

第二个不同是帧的传输。在ALOHA中，我们先传输一个完整的帧，然后等待确认。在CMSA/CD中，传输和冲突检测是一个连续的过程。我们并不是先发送一个完整的帧，然后才检测冲突。站点的传输和接收是连续的和同步的（使用两个不同的端口）。我们用一个回路来显示传输的一个连续的过程，我们不断地进行监控以检测出以下两种情况中的一种：传输完成了或者检测到一个冲突。这两个事件都能使传输停止。如果从回路出来，还是没有检测到冲突，这就意味着传输已经完成了，整个帧被传输了。否则，便有冲突发生了。

第三个不同是万一其他站点仍未检查到冲突，便发送一个短小的干扰信号（jamming signal）使冲突加强。

能量级别

通道中的能量级别有三个值：0，正常和不正常。在级别0，通道是空闲的。在正常级别，一个站点成功地占用了通道并正在发送一个帧。在非正常级别，发生了冲突且能量的级别是正常级别的两倍。将要发送或者正在发送帧的站点必须监控能力级别来测定通道是处于空闲，繁忙还是冲突状态。图12.15描述了这个情形。

吞吐量

CSMA/CD的吞吐量大于纯ALOHA或时隙ALOHA，基于不同的持续方法及在p-持续方法中不同的p值，不同的G值将有不同的最大吞吐量。1-持续方法中，当$G=1$时，得到最大吞吐

量约为50%。非持续方法中，当G值在3到8之间时，能得到高达90%的最大吞吐量。

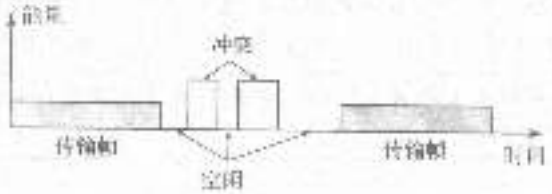


图12.15 传输、空闲或冲突时的能力级别

12.1.4 带冲突避免的载波侦听多路访问（CSMA/CA）

CMSA/CD的基本理念是一个站点在传输过程中检测到冲突时，要有能力去接收信号。当没有冲突时，站点接收到一种信号：它自身的信号。当冲突发生时，站点接收到两种信号：它自身的信号和第二个站点传输的信号。为了区分以上两种情况，不同情况中接收到的信号应该明显不同。换言之，来自第二个站点的信号需要为第一个站点产生的信号增加大量的能量。

在一个有线网络中，接收信号的能量与发送信号的能量基本一样，或者因为线路长度较短，或者因为存在中继器放大了发送方和接收方之间的能量。这就是说在冲突中，被检测到的能量几乎翻倍。

然而，在一个无线网络中，大量发送信号的能量在传输中丢失。接收信号的能量也很小。因此，一个冲突可能只增加5%～10%的额外能量。这对有效的冲突检测起不了什么作用。

因为无线网络的冲突无法检测到，故而我们要避免冲突。带冲突避免的载波侦听多路访问（CSMA/CA）就是为这种网络发明的。通过使用CSMA/CA三种方法来避免冲突：帧间间隔，竞争窗口和确认，如图12.16所示。

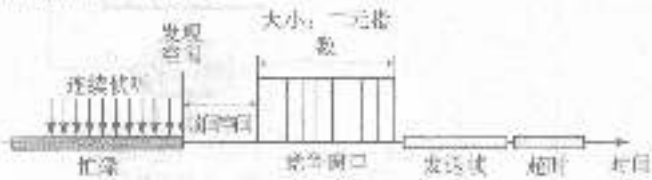


图12.16 CSMA/CA中的时间安排

帧间间隔（IFS）

首先，通过推迟传输来避免冲突，即使通道是空闲的。当发现通道是空闲的时，站点并不马上发送。它会等待一段时间，这段时间称之为帧间间隔（interframe space，IFS）。当然当通道被侦听时，它看上去是空闲的，但是一个远端站点可能早已开始传输了，只是远端站点的信号还未到达这个站点而已。IFS时间使得远端站点传输的信号能到达该站点。如果在IFS时间后，通道仍然是空闲的，那么站点就能发送了，但是它仍然要等待一段时间，这段时间等于竞争时间（以后解释）。IFS变量也能用来区分站点或帧的优先次序。例如，一个站点被赋值的IFS越短，它的优先权越高。

> 在CSMA/CA中，IFS也能用来定义一个站点或一个帧的优先权。

竞争窗口

竞争窗口将时间分割成时隙。预备发送的站点选择一个随机时隙作为等待的时间。窗口中时隙的数量是根据补偿策略中的二元指数而变化的。这就是说第一次设定一个时隙，以后每次站点在IFS时间后无法检测到空闲通道时，时隙的数量都要翻番。这与p-持续方法很相似，

除了在后者中等待站点根据随机结果来决定时隙的数量以外。竞争窗口中有趣的一点是，每次时隙后站点都要侦听通道。然而，如果站点发现通道繁忙，它并不重启程序，而是停止定时器并当侦听到通道空闲时再重启定时器。这给了站点最长等待时间的优先权。

> 在CSMA/CA中，如果站点发现通道繁忙，它并不重启竞争窗口的时钟，而是停止定时器并当通道空闲时再重启定时器。

确认

虽然用了所有这些防范措施，但是仍然有可能产生冲突而导致帧的损坏。另外，数据也可能在传输中被损坏。肯定的确认和定时器的超时将有助于确保接收方收到帧。

程序

图12.17说明了程序。注意通道在IFS之前和之后都要被侦听。在竞争时间内，通道也要被侦听。在竞争窗口中的每一个时隙内，通道都要被侦听。如果发现通道空闲，定时器继续；如果通道繁忙，定时器停止并在通道空闲后再继续。

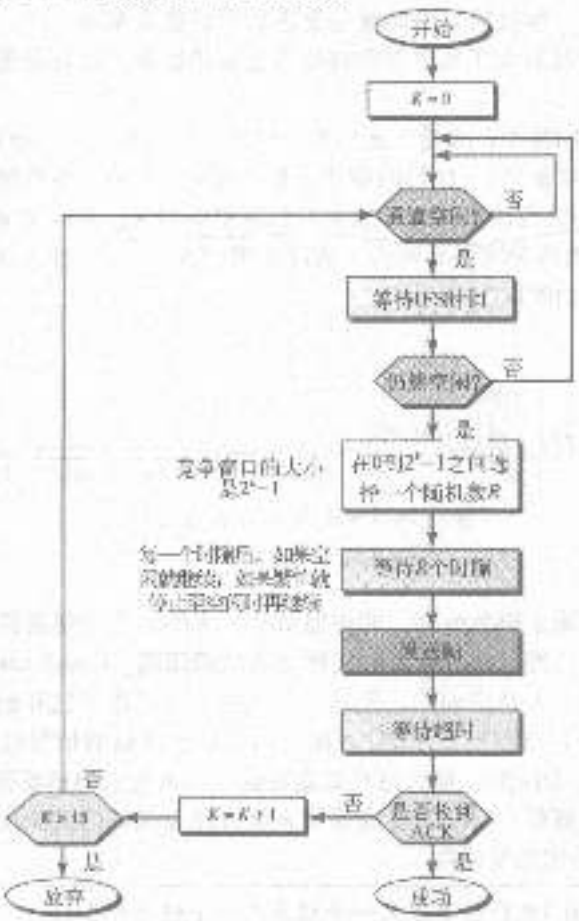


图12.17　CSMA/CA的流量表

CSMA/CA和无线网络

CSMA/CA被广泛地使用在无线网络中。然而，上述的程序还不能处理一些与无线网络相关的特殊问题，比如隐藏的终端或暴露的终端。我们将看到如何通过为上述协议添加握手帧

件来解决这些问题。CSMA/CA在无线网络中的应用会在第14章中讨论。

12.2 受控访问

在受控访问（controlled access）中，站点之间相互协商以确定哪一个站点有权发送。某一个站点没有得到其他站点的认可就无权发送。下面我们讨论三种流行的受控访问方法。

12.2.1 预约

在预约（reservation）方法中，站点在发送数据之前需要预约。将时间划分为时隙。在每一个时隙内，在数据帧之前先发送一个预约帧。

如果系统中有N个站点，在预约帧中就给好会有N个预约子时隙。每个子时隙均属于一个站点。当一个站点想要发送一个数据帧时，它在自己的子时隙中做一个预约。已经预约完的站点可以在预约帧之后发送数据帧。

图12.18描述了一个有五个站点和一个有五个子时隙的预约帧的情形。在第一个间隙，只有站点1，3和4做了预约。在第二个间隙，只有站点1做了预约。

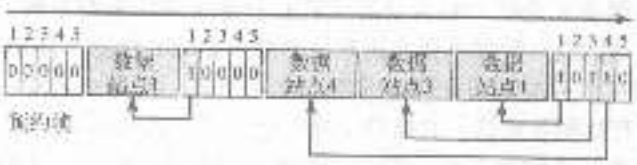


图12.18 预约访问方法

12.2.2 轮询

轮询（polling）方法的拓扑结构是一个设备作为主站（primary station），而另一些设备作为从站（secondary station）。所有的数据交换都要通过主站点进行，即使最终的目的站点是从站点时也是这样。主设备控制链路，而从设备只是跟随着它的指令工作。主设备决定在给定的时间内哪个设备被容许使用通道。所以，主设备始终是一个会话的发起者。（见图12.19）

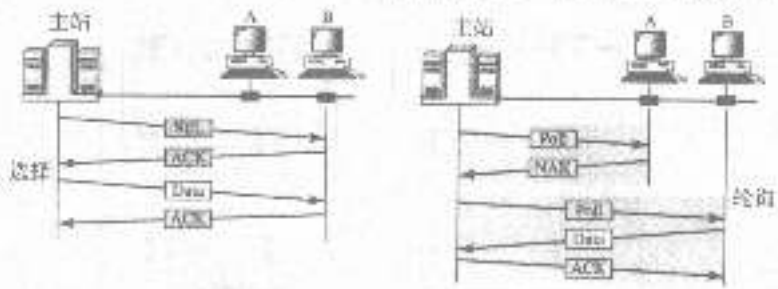


图12.19 在轮询访问方法中的选择和轮询功能

如果主设备希望接收数据，它就询问从设备是否有数据要发送，这一功能称做轮询。如果主设备希望发送数据，它就通知从设备准备接收，这一功能称做选择。

选择

在主设备要发送数据时就使用选择功能。请记住，主设备控制链路，如果主设备既不发送数据也不接收数据，那么它就知道链路是可用的。这时，如果它有数据要发送，主设备就可以发送，但是它不清楚的是，目标设备是否准备接收。所以，主设备必须通知从设备这即将到来的传输，并等待从设备发送它已经处于准备好状态的确认。在发送数据之前，主设备要创建并发送一个选择（SEL）帧，其中的一个字段包含目标从设备的地址。

轮询

主设备使用轮询功能向从设备请求传输。当主设备准备接收数据时，它必须逐个询问每一个设备是否要发送数据。当询问到第一个设备时，如果它没有数据要发送，就用一个NAK帧来响应，如果它要发送数据，就用数据（用数据帧格式）来响应。如果响应的是否认帧（NAK），主设备就用同样的方法轮询下一个从站点，直到它找到一个要发送数据的站点为止。如果响应是数据帧，主设备就读取该帧并返回一个确认（ACK帧），表示已经接收。

12.2.3 令牌传递

在令牌传递（token-passing）方法中，网络中的站点被组织在一个逻辑环中。换言之，对每一个站点来说，都有它的前驱和后继。前驱是指逻辑环中处于该站点之前的站点，而后继是指在环中处于该站点之后的站点。该站点是目前正在进行访问的站点。访问的权限由该站点的前驱站点传递给它。当该站点没有数据再要发送时，便将权限传送给它的后继站点。

但是访问通道的权限如何一个站点一个站点地进行传递呢？在此方法中，一个特殊的信息分组被称为令牌（token）在环中循环。令牌的持有赋予站点访问通道并发送数据的权限。当一个站点有一些数据要发送时，它要等待从它的前驱站点传来的令牌。然后，它持有令牌并发送数据。当这个站点没有数据再要发送时，它释放令牌并传递给环中的下一个逻辑站点。站点只有在下一个循环中再次收到令牌时才能再发送数据。在这个过程中，如果一个站点收到了令牌，但是它没有数据要发送，它便将令牌传递给下一个站点。

在这种访问方法中，令牌管理是必需的。站点占有令牌的时间必须得到限制。令牌必须得到监控以确定它没有丢失或损坏。例如，如果一个站点持有令牌失败了，令牌就会在网络中消失。令牌管理的另一个功能是给站点和要传输的数据类型分配优先权。最后，令牌管理能够让低优先权站点将令牌释放给高优先权的站点。

逻辑环

在令牌传递网络中，站点不必要真的连接在一个环中，这个环只是逻辑的。图12.20显示了能创造逻辑环的四种不同的物理拓扑结构。

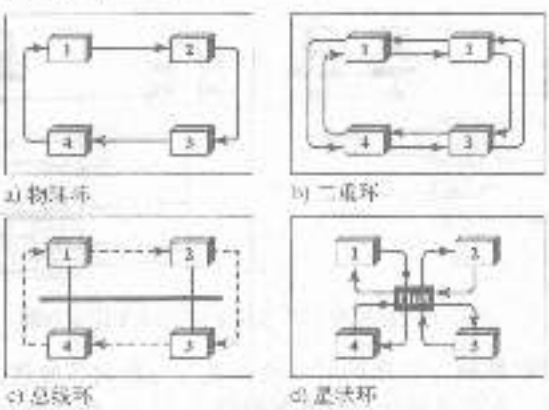


图12.20　在令牌传递访问方法中的逻辑环和物理拓扑结构

在物理环拓扑结构中，当一个站点把令牌传给它的后继站点时，其他站点无法看到令牌，后继站点是线路中的下一个，这意味着令牌不必知道下一个后继站点的地址。这种结构的问题是如果链路中的一个链路（在两个邻近站点之间的介质）出现了故障，整个系统就失败了。

二重环拓扑结构使用了一个二级（辅助的）环，二级环的运行方向与主环的正好相反。二级环只在紧急情况才使用（就像汽车的备用胎一样），如果在主环链路中的一个链路发生了故障，系统会自动地将两个环合并为一个临时的环。当发生故障的链路修复好后，辅助环又进入空闲状态。注意要运行这种拓扑结构，每个站点都要有两个传输端口和两个接收端口。被称为光纤分布式数据接口（Fiber Distributed Data Interface，FDDI）和铜线分布式数据接口（Copper Distributed Data Interface，CDDI）的高速令牌环网络就使用这个拓扑结构。

总线环拓扑结构，也称为令牌总线。在此拓扑结构中，站点与称为总线的单根光纤连接。然而，它们也组成了一个逻辑环，因为每个站点都知道它后继站点的地址（为了令牌管理的目的，它们同样也知道前驱站点的地址）。当一个站点完成数据发送后，它释放令牌并将后继站点的地址插入令牌中。只有地址与令牌的目的地址相匹配的站点才能访问共享介质。IEEE标准化的令牌总线局域网，就使用这种拓扑结构。

在星状环拓扑结构中，物理拓扑结构是星状的。其中有一个网络集线器充当连接器。网络集线器中的电线形成了环，站点通过两个电线与环相连。这种拓扑结构使得网络不容易发生故障，因为如果一个链路失败了，网络集线器可以忽略它，而其他的站点还是能够运行的。而且从环中增加和减少站点比较方便。IBM设计的令牌环局域网仍在使用这种拓扑结构。

12.3 通道化

通道化（channelization）也是一种多路访问方法。在这一方法中，不同的站点间在时间上、频率上或通过编码来共享链路的可用带宽。在本节我们讨论三种信道化协议：FDMA、TDMA和CDMA。

> 在第16章中，当我们讨论移动电话系统时，将看到所有这些方法的应用。

12.3.1 频分多路访问（FDMA）

在频分多路访问（frequency-division multiple access，FDMA）中，可用的带宽被划分成频带。每一个站点都使用分配给它的频带发送数据。换言之，每一个频带都预留给特定的站点使用，该频带永远属于一个站点。每个站点也使用带通滤波器来限定传输器的频率。为了防止站点间的冲突，通过小的防护频带将指定的波段彼此分开。图12.21说明了FDMA的概念。

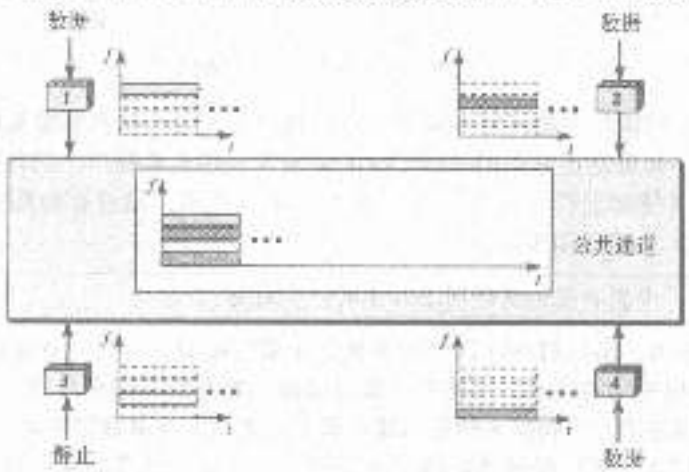


图12.21 频分多路访问（FDMA）

在FDMA中，公共通道的可用带宽被分割成频带并由防护频带将其分离。

FDMA为整个通信期间指定一个预定频率的频带。这就意味着FDMA能方便地使用流数据（无法分组的连续的数据流）。在第16章中，我们将看到移动电话系统是如何利用这一特性的。

我们必须强调的是，虽然FMDA与FDM概念上看似相似，但是两者之间存在区别。第6章介绍的FMD是一种物理层的技术，能组合低带宽通道的负载并使用一个高带宽通道来传输它们。这些被组合的通道是低速。多路复用器调制信号，将它们组合并产生一个带通信号。每个通道的带宽都被多路复用器转换了。

另一方面，FDMA是数据链路层的一种访问方法。每个站点的数据链路层都告知它的物理层把传递给它的数据转换成带通信号，信号必须在被指定的频带中产生。在物理层并没有实际的多路复用器存在。每个站点产生的信号是自动带通过滤的，当它们被发送到公共通道时，它们是混合在一起的。

12.3.2　时分多路访问（TDMA）

在时分多路访问（time-division multiple access，TDMA）中，所有的站点在时间上共享通道的带宽。每一个站点被分配一个时隙，只有在这一时隙内它才能够发送数据。每个站点在指定的时隙中传输数据。图12.22说明了TDMA的概念。

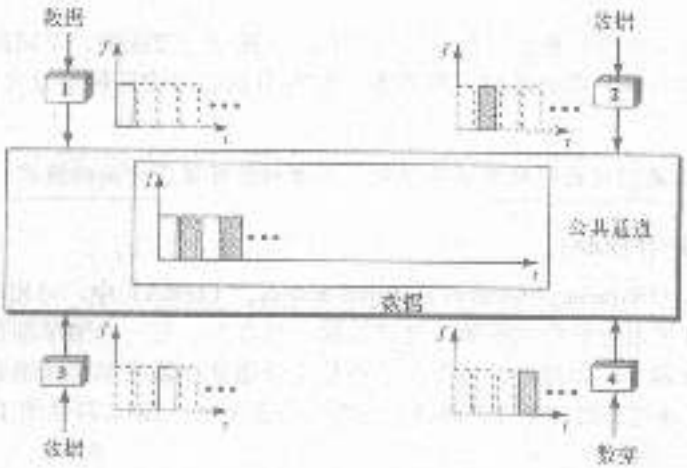


图12.22　时分多路访问（TDMA）

TDMA的主要问题在于如何使不同的站点达到同步。每个站点都要知晓自身时隙的开始点和所处的位置。如果站点分布在一个广泛的区域内，那么系统产生的传播延迟会给前述问题带来困难。为了使延迟得到补偿，我们插入了保护时间。通过在时隙开始点插入同步位（通常称为前导位）来完成同步化。

在TDMA中，带宽被视为是时间上共享的一个通道。

我们同样要强调，虽然TDMA与TDM在概念上看似相似，但它们也存在不同。第6章中介绍的TDM是一种物理层的技术，它将来自低速通道的数据组合起来并使用一个高速通道来传输它们。这个过程使用一个物理多路复用器从每个通道交叉存取数据单元。

另一方面，TDMA是数据链路层的一种访问方法。每个站点的数据链路层都告知它的物理层使用指定的时隙。在物理层并没有实际的多路复用器存在。

12.3.3 码分多路访问（CDMA）

码分多路访问（code-division multiple access，CDMA）的概念在几十年前就已经提出来了。电子技术的新发展使得它的实现成为可能。与FDMA不同，CDMA仅有一个通道占据链路整个的带宽；而与TDMA不同的是，所有的站点都可以同时发送数据。没有时间的共享问题。

> 在CDMA中，一个通道同时承载所有的传输。

类比

我们先来给出一个类比。CDMA简指使用不同的编码来通信。例如，在一个大房间里有许多人，有两个人能用英语通话而其他的人不懂英语。另外两个人能用中文通话，也只有他们两人懂中文，依此类推。换言之，公共通道即该例中的房间空间，几对人很容易进行沟通，却要使用不同的语言（编码）。

思想

我们假设有四个站点1，2，3和4连接在同一通道上。来自站点1的数据是d_1，站点2的是d_2，依此类推。分配给第一个站点的编码是c_1，第二个的是c_2，依此类推。我们假定被分配的编码有两个特性。

1. 两个编码相乘得0。
2. 编码自身相乘得4（站点的数量）。

通过这两个假想的特性，我们来看看以上四个站点如何使用同一公共通道发送数据，如图12.23所示。

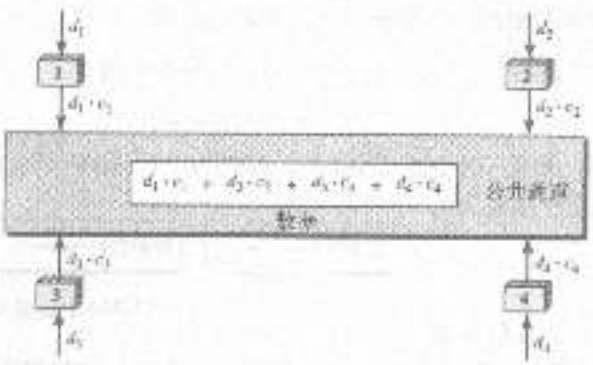


图12.23 用编码进行的简单想象

站点1将它的数据与编码相乘（正如我们将看到的那样，这是一种特殊的乘法）得到$d_1 \times c_1$。站点2将它的数据与编码相乘得到$d_2 \times c_2$，依此类推。通道上的数据是所有这些结果的总和，如图中方块部分所示。任何站点想要接收的数据是发送方的编码和数据的乘积。例如，假定站点1和站点2正在通话。站点2想知道站点1在说什么。它在通道中将数据c_1和站点1的编码d_1相乘。

因为$(c_1 \times c_1) = 4$，但$(c_2 \times c_1)$，$(c_3 \times c_1)$和$(c_4 \times c_1)$都为0，站点2为了获取从站点1传来的数据，将结果除以了4。

> 数据$=(d_1 \cdot c_1+d_2 \cdot c_2+d_3 \cdot c_3+d_4 \cdot c_4) \cdot c_1=d_1 \cdot c_1 \cdot c_1+d_2 \cdot c_2 \cdot c_1+d_3 \cdot c_3 \cdot c_1+d_4 \cdot c_4 \cdot c_1=4 \times d_1$

芯片

CDMA基于编码理论。每个站点被指定一个编码，编码是一个被称之为芯片数字序列，如图12.24所示。图中的编码是上一个例子中的。

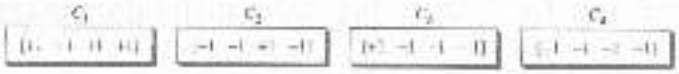


图12.24　芯片序列

本章后面的内容还会介绍如何选择序列。现在，我们需要知道的是：序列的选择并非随机的，它们是被谨慎选择的。它们被称为正交序列（orthogonal sequences），有以下的特性：

1. 每个序列有N个元素，而N正是站点的数量。

2. 如果将序列乘以一个数，序列中的每个元素都乘以这个数。这称为序列和标量的乘积。例如，

$$2 \times [+1+1 \quad -1 \quad -1] = [+2+2 \quad -2 \quad -2]$$

3. 如果将两个相同序列的元素逐个相乘，并将结果相加，结果为N，而N是每个序列中元素的数量。这就是两个相同序列的内积。例如，

$$[+1+1-1-1] \times [+1+1-1-1] = 1+1+1+1 = 4$$

4. 如果将两个不同序列的元素逐个相乘，并将结果相加，结果为0。这就是两个不同序列的内积。例如，

$$[+1+1 \quad -1 \quad -1] \times [+1+1+1+1] = 1+1-1-1 = 0$$

5. 将两个序列相应的元素相加，结果是另一个序列。例如，

$$[+1+1-1-1] + [+1+1+1+1] = [+2+2 \quad 0 \quad 0]$$

数据的表示

我们遵循的编码规则：如果一个站点发送0位，编码为－1；如果发送1位，编码为+1。当站点空闲时，它不发送信号，编码为0。

如图12.25所示。

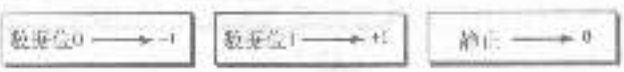


图12.25　CDMA中数据的表示法

编码和解码

作为一个简单的例子，我们来看看四个站点如何共享一个时间间隔为1位的链路。程序容易为附加的时间间隔重复。我们假定站点1和站点2正在发送0位，站点4正发送1位，而站点3是静止的。发送方的数据被依次编码为－1，－1，0和+1。每个站点将相应的数与它的芯片序列（正交序列）相乘。这对每个站点来说都是唯一的。结果是产生一个新的序列并被发送到通道上。为了简化，我们假定所有的站点在同一时点发送结果序列。通道上的序列是所有四个序列的总和。图12.26说明了该种情况。

现在假定静止的站点3正在“听”站点2。站点3将通道上的数据乘以站点2的编码[+1 −1 +1−1]，得到：

$$[-1 \quad -1 \quad -3 \quad +1] \times [+1 \quad -1 \quad +1 \quad -1] = -4/4 = -1 \Rightarrow \text{位1}$$

信号水平

如果我们说明了每个站点产生的数字信号及目的地所恢复的数据（见图12.27），便能更

好的理解这个过程，图中显示了每个站点相应的信号（为简化使用了NRZ－L）及在公共通道中的信号。

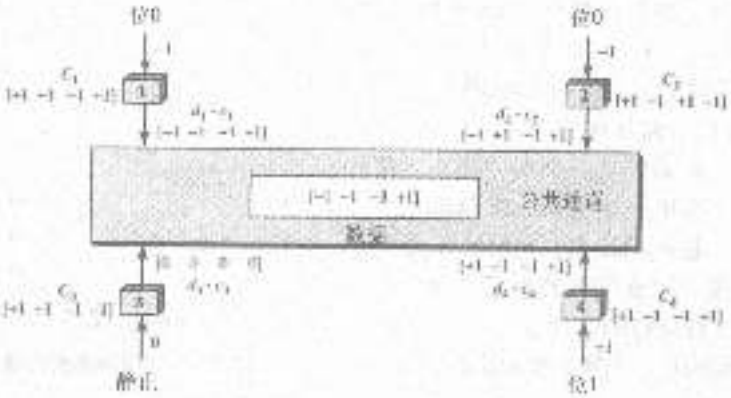


图12.26 CDMA的共享通道

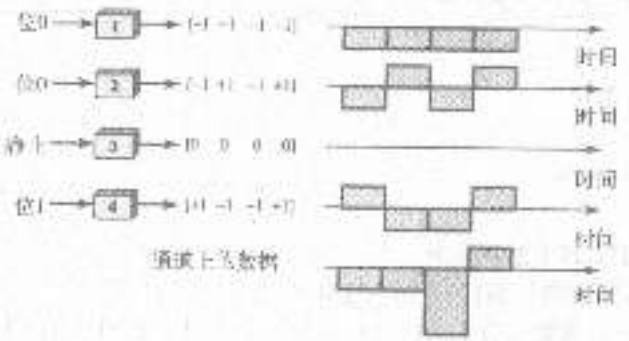


图12.27 在CDMA中四个站点产生的数字信号

图12.28显示了通过使用对站点2的编码，站点3可以检测到来自站点2的数据。通道中的总数据与代表站点2的芯片编码的信号相乘（内积运算）产生一个新的信号。然后，站点将信号下的领域整合并相加，得到的值是－4，－4除以4的结果－1代表位0。

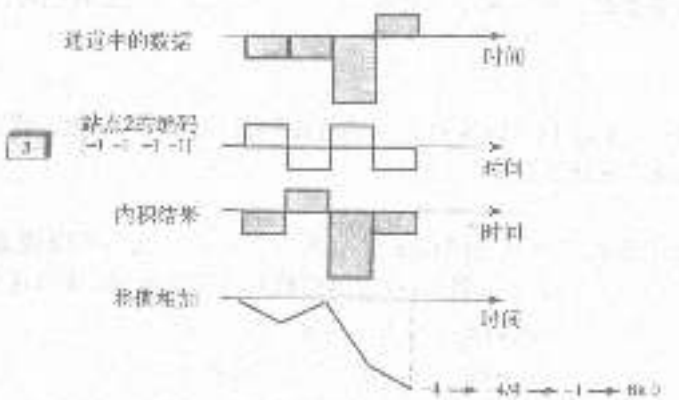


图12.28 CDMA中合成信号的解码

序列的产生

为产生芯片序列，我们使用一张行列数相同的Walsh表，这是一张二维表见图12.29。

在Walsh表中，每一行是一个芯片序列。对于一芯片序列的W_1有一行和一列，我们可以为W_1选+1或−1，这里选+1。根据Walsh，如果我们知道N序列W_N的表，我们就能产生$2N$序列W_{2N}的表。如图12.29所示。带上标线的$\overline{W}_N$代表W_N的反码，这里+1的反码是−1，而−1的反码是+1。图12.29也说明了我们如何从W_1产生W_2和W_4。在我们选择了W_1之后，W_2就能从四个W_1中得出，最后一个是W_1的反码。在W_2产生之后，W_4就能从四个W_2得出，最后一个是W_2的反码。当然，W_8由四个W_4组成，依此类推。注意在W_N产生后，会分配给每一个站点与其一行对应的芯片。

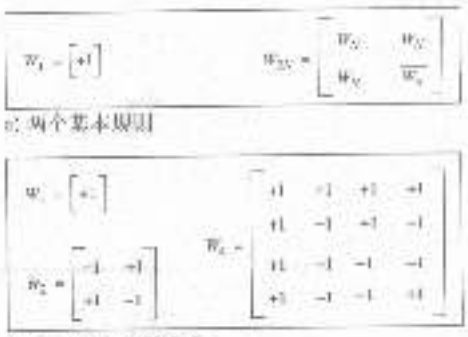


图12.29　产生Walsh表的通用规则和示例

我们需要强调的是，序列个数N需要是2的幂。换言之，$N=2^n$。

> Walsh表中的序列数需要是$N=2^n$。

例12.6

找出以下网络的芯片：

a. 两个站点

b. 四个站点

解：

我们可以使用图12.29中的W_2和W_4。

a. 对于两个站点的网络，我们有[+1 +1]和[+1 −1]。

b. 对于四个站点的网络，我们有[+1 +1 +1 +1]，[+1 −1 +1 −1]，[+1 +1 −1 −1]和[+1 −1 −1 +1]

例12.7

如果在网络中有90个站点，序列个数是多少？

解：

序列个数需要是2^n。我们可以选择$n=7$，$N=2^7$或128。那么我们可以使用90个序列作为芯片。

例12.8

如果用通道中总的数据乘以发送方的芯片编码再除以站点个数，那么可以证明接收站点能够接收特定发送方发送的数据。

解：

让我们使用前面的四个站点的例子证明第一个站点。我们可以说通道上的数据$D=d_1 \cdot c_1+d_2 \cdot c_2+d_3 \cdot c_3+d_4 \cdot c_4$。需要得到站点1发送的数据的接收方用$c_1$乘以这个数据。

$$\begin{aligned} D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\ &= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\ &= d_1 \times N \end{aligned}$$

当我们用N除以结果，我们就得到d_1。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物。在本节末列出括号[……]中的参考资料。

读物

多路访问在[Tan03]的第4章、[Sta04]的第16章、[GW04]的第6章和[For03]的第8章中有讨论。更深一步的材料可以在[KMK04]中找到。

本章小结

- 我们认为数据链路层可分为两个子层，上子层负责数据链路控制，而下子层负责解决对共享介质的访问。
- 为了解决共享链路的访问并发了许多协议，我们将这些协议归为三类：随机访问协议，受控访问协议和通道化协议。
- 在随机访问方法即竞争方法中，没有站点优先于其他站点，也没有站点受控于其他站点。
- ALOHA允许多路访问以共享介质。在此协议中存在着潜在的冲突，当一个站点发送数据时，另一个站点可能也在试图同一时间发送数据。那么来自两个站点的数据冲突并混淆了。
- 为了减小冲突的可能性并提高效率，发明了CSMA方法。如果站点在发送前先侦听介质，那么冲突的机率就减少了。有三种载波侦听的方法：1-持续、非持续和p-持续。
- 带冲突检测的载波侦听多路访问（CSMA/CD）在CSMA基础上增加了算法以解决冲突问题。在这个方法中，在发送一个帧后，站点监控介质以了解传输是否成功。如果成功了，站点就完成了传输。如果存在冲突，帧就会被重发。
- 为了避免无线网络中的冲突，发明了带冲突避免的载波侦听多路访问（CSMA/CA）方法。通过使用三种策略来避免冲突：帧间间隔，竞争窗口和确认。
- 在受控访问中，站点间彼此商议以决定那个站点有权发送。一个站点只有被其他站点赋权后才能发送。我们介绍了三种著名的受控访问方法：预约，轮询和令牌传递。
- 在预约访问方法中，站点在发送数据前要先预约。时间被分割成时间间隙。在每个时间间隙，预约帧都先于数据帧发送。
- 在轮询访问方法中，所有的数据交换都必须通过主设备，即使最终目的地是从设备也不例外。主设备控制整个链路，从设备遵循它的指令。
- 在令牌传递访问方法中，网络中的站点被组成一个逻辑环。每个站点都有一个前驱和一个后继。被称为令牌的特殊信息分组在环中循环。
- 通道化也是一种多路访问方法。在这一方法中，不同的站点在时间上、频率上或通过编码来共享链路的可用带宽。我们讨论了三种通道化协议：FDMA，TDMA和CDMA。
- 在频分多路访问（FDMA）中，可用的带宽由所有的站点共享，每一个站点都使用分配给它的频带发送数据。换言之，每一个频带都预留给特定的站点使用。该频带永远属于一个站点。
- 在时分多路访问（TDMA）中，所有的站点在时间上共享通道的带宽。每一个站点被分配一个时隙，只有在这一时隙内它才能够发送数据。每个站点在指定的时隙中传输数据。
- 在码分多路访问（CDMA）中，站点使用不同的编码以达到多路访问。CDMA基于编码理论并使用芯片序列，通过使用例如Walsh表的正交编码方法来产生序列。

习题

复习题

1. 列出本章中介绍的三类多路访问协议。
2. 定义随机访问方法并列出其中的三种协议。
3. 定义受控访问方法并列出其中的三种协议。
4. 定义通道化方法并列出其中的三种协议。
5. 解释为什么冲突在随机访问协议中是一个问题，而在受控访问和通道化协议中不是。
6. 比较和对照随机访问协议和受控访问协议。
7. 比较和对照随机访问协议和通道化协议。
8. 比较和对照受控访问协议和通道化协议。
9. 当我们使用本地电话公司的线路访问因特网时，需要一个多路访问协议吗？为什么？
10. 我们使用有线电视的通道访问因特网时，需要一个多路访问协议吗？为什么？

练习题

11. 我们有一个有100个站点的纯ALOHA网络。如果T_p=1μs，每个站点发送多少个帧能达到最大效率。
12. 重复第11题，这次为时隙ALOHA网络。
13. 一个纯ALHOA网络中100个站共享1Mbps通道。如果帧是1 000比特长并且每个站每秒发送10个帧，那么吞吐量是多大？
14. 重复第13题，这次是时隙ALHOA。
15. 在数据速率为10Mbps的CDMA/CD网络中，为了冲突检测过程的正确操作，最小帧长度是512比特。如果我们把数据速率提高到100Mbps，那么最小帧长度应该是多少？那么提高到1Gbps、10Gbps分别是多少呢？
16. 在数据速率为10Mbps的CDMA/CD网络，为了冲突检测过程的正确操作，最小帧长度是512比特。如果我们把数据速率提高到100Mbps，那么最大距离应该是多少？那么提高到1Gbps、10Gbps分别是多少呢？
17. 在图12.12中，数据速率是10Mbps，站点A和站点C之间的距离是2 000米，传播速度是2×10^8m/s。站点A开始在$t_1=0$时刻开始发送一个长帧；站点C在$t_2=3$μs时刻开始发送一个长帧。帧足够长以保证两个站点的冲突检测。得出：
    a. 站点C听到冲突的时刻（t_3）　　b. 站点A听到冲突的时刻（t_4）
    c. 站点A在检测到冲突前发送的比特数　　d. 站点C在检测到冲突前发送的比特数
18. 重复第17题，如果数据速率是100Mbps。
19. 根据图12.29中的W_1计算出W_8。
20. 使用$W_1=[-1]$重新得出图12.29中的W_2表和W_4表。用得出的表与图12.29中的表相比较。
21. 证明图12.29中W_4的Walsh芯片的第三个和第四个正交属性。
22. 证明第20题中W_4的Walsh芯片的第三个和第四个正交属性。
23. 如果站点1和站点3静止，重复图12.27中描述的情节。
24. 有1个主站和4个从站的网络使用轮询。数据帧的长度是1 000个字节。轮询帧、ACK帧和NAK帧长度是32个字节。每个站点有5个帧发送，如果站点响应轮询能发送的帧数没有限

制，那么交换的总字节数是多少？

25. 如果每个站点响应轮询时只能发送一个帧，重复第24题。

探究题

26. 你能解释为什么ALOHA中的旋回时间依靠T_b，而CSMA中依靠T_p？

27. 在分析ALOHA中，我们只使用一个系数：时间。分析CSMA中，我们使用两个系数：间隔和时间。你能解释原因吗？

第13章 有线局域网：以太网

在第1章中我们知道，局域网（LAN）是用于有限的地理范围，如一幢建筑物或一个学校的计算机网络。虽然为了共享资源这一目的可以将LAN作为孤立的网络，用来连接一个组织内的计算机，但是，当今很多的LAN也连接到广域网（WAN）或因特网上。

局域网市场有多种技术，例如以太网，令牌环，令牌总线，FDDI和ATM LAN，其中的某些技术只生存了一段时间，而如今最普遍应用的技术是以太网。

在本章中，我们将先简要地介绍IEEE标准项目802，它被设计来规范制造LAN以及不同LAN之间的互连。然后我们主要介绍以太局域网。

虽然在过去的几十年以太网经历了四代的发展，但是主要理念还是相同的。已经对以太网进行了改进以满足市场的需求并对新技术加以应用。

13.1 IEEE标准

1985年，计算机团体IEEE（电气与电子工程师学会）开始了一个项目，称为项目802（Project 802），以设定标准使得不同制造商生产的设备之间能相互通信。项目802并不旨在代替OSI或因特网模式，相反，它说明了大多数局域网协议的物理层和数据链路层的功能。

标准被美国国家标准协会（ANSI）所采用。1987年，国际标准化组织（OSI）也批准它成为一个国际标准，命名为ISO 8802。

图13.1显示了802标准和传统的OSI模型之间的关系。IEEE将数据链路层分为两个子层：**逻辑链路控制层**（logical link control，LLC）和**介质访问控制层**（media access control，MAC）。IEEE也为不同的局域网协议制定了一些物理层标准。

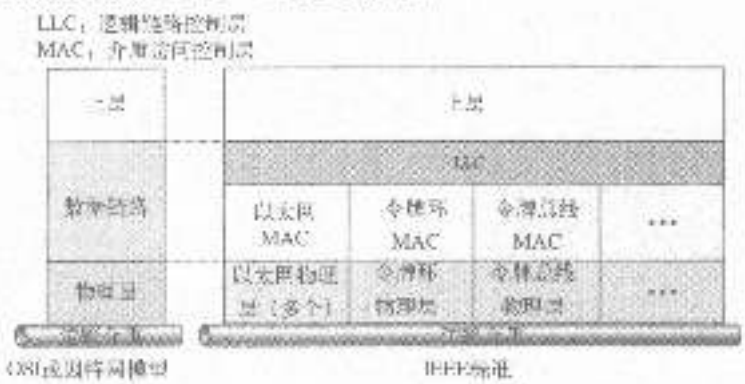


图13.1 局域网的IEEE标准

13.1.1 数据链路层

正如我们之前提到的那样，在IEEE标准中，数据链路层被分成两个子层：LLC和MAC。

逻辑链路控制层（LLC）

在第11章中，我们介绍了数据链路控制。数据链路层是用来处理成帧、流量控制和差错控制的。在IEEE项目802中，流量控制、差错控制和部分成帧的职能都被集中到称为逻辑链路

控制层的子层中。LLC和MAC两者都进行成帧处理。

LLC为所有的IEEE局域网提供一个单一的数据链路控制协议。在这方面，LLC不同于介质访问控制子层，后者为不同的局域网提供不同的协议。一个单一的LLC协议能使不同的局域网之间进行相互交换。因为它使MAC子层变得透明。图13.1显示了为多个MAC协议服务的一个单一LLC协议。

成帧 LLC定义了一个协议数据单元（PDU），它在一定程度上与HDLC相似。头部包含一个控制字段就像HDLC中的一样，这个字段用以流量控制和差错控制。另两个头部字段界定了使用LLC的源和目的地的上层协议。这些字段称为**目的业务接入点**（destination service access point，DSAP）和**源业务接入点**（source service access point，SSAP）。在典型的数据链路控制协议中定义的其他协议（如HDLC）被移入到了MAC子层中。换言之，HDLC中定义的帧被分为LLC子层的PDU和MAC子层的一个帧。如图13.2所示。

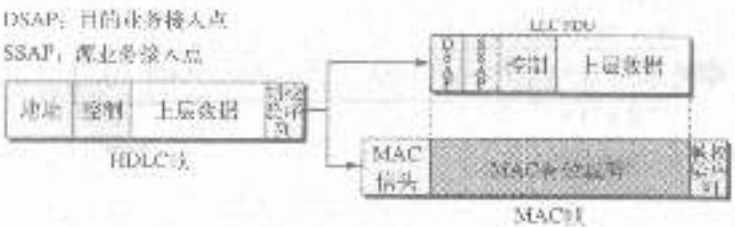


图13.2 HDLC帧与LLC及MAC帧的比较

LLC的需求 LLC的目的是为需要流量控制和差错控制的上层协议提供这些服务。例如，如果一个或几个局域网被应用于一个孤立的系统中，就需要LLC为应用层协议提供流量控制和差错控制。然而，多数上层协议诸如IP（第20章会介绍）并不需要LLC的服务。因此，我们就此结束对LLC的介绍。

介质访问控制层（MAC）

在第12章，我们介绍了包括随机访问、受控访问和通道化等的多种访问方法。IEEE项目802创造了一个称为介质访问控制层的子层来为每个局域网定义特定的访问方法。例如，它定义CSMA/CD作为以太网的介质访问方法，令牌传递方法为令牌环和令牌总线局域网的介质访问方法。正如我们之前讨论的那样，MAC子层同样也有部分成帧的功能。

与LLC子层相比较，MAC子层包含了一些独特的模块，每个模块都为相应的局域网协议定义了访问方法和成帧的格式。

13.1.2 物理层

物理层取决于网络的实现和使用的物理介质。IEEE为每个局域网的实现定义了详细的规范。例如，在标准以太网中虽然只有一个MAC子层，但是每种以太网的实现都有不同的物理层规范，我们稍后会作介绍。

13.2 标准以太网

最初的以太网是由施乐公司的Palo Alto研究中心（PARC）于1976年创建的。自此之后，它经历了四代发展：标准以太网（Standard Ethernet）（10[⊖] Mbps），快速以太网（Fast Ethernet）（100Mbps），千兆以太网（Gigabit Ethernet）（1Gbps）和10千兆以太网（Ten-

⊖ 以太网定义了一些1 Mbps协议，但它们没有发展起来。

Gigabit Ethernet）（10Gbps），如图13.3所示。我们将简要地介绍这四代以太网，先介绍标准（或传统）以太网。

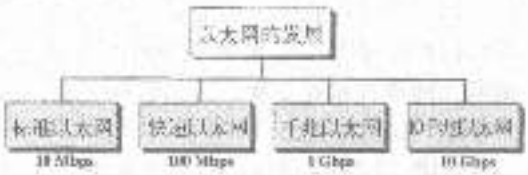


图13.3　以太网的四代发展

13.2.1　MAC子层

在标准以太网中，MAC子层控制访问方法。它也负责将来自上层的数据成帧并传输给物理层。

帧的格式

以太网的帧包括7个字段：前导符、SFD、DA、SA、协议数据单元（PDU）的长度/类型、上层数据和CRC。对接收到的帧，以太网不提供确认的任何机制，所以将它称做不可靠的介质。确认必须在其高层完成。MAC的帧格式如图13.4所示。



图13.4　802.3MAC帧

- **前导符**。802.3帧的第一个字段包含7个字节（56位），其中的1和0交替出现，以此通知接收系统有帧的到来并使其与输入的时钟同步。该模式仅提供一个通知和时钟脉冲。56位模式容许站点在帧的开始可以丢弃一些位。实际上，**前导符**（preamble）是在物理层上加进去的，并不是帧的一个部分。
- **起始帧分界符**（SFD），第二个字段（1个字节：10101011）说明帧的开始。SFD通知站点它们这是最后一个同步的机会，最后的两位是11，并且通知接收方下一个字段是目的地址。
- **目的地址**（DA）。DA字段为6个字节，包含目的站点或接收该分组的站点的物理地址。后面我们会简要地介绍目的地址。
- **源地址**（SA）。SA字段也是6个字节，包含分组的发送方的物理地址。后面我们会简要地介绍源地址。
- **长度/类型**。这一字段定义为长度或类型字段。最初的以太网使用这个字段作为类型字段来定义使用MAC帧的上层协议。IEEE标准使用它作为长度字段来定义数据字段的字节数。在今天两者都被普遍使用。
- **数据**。这一字段的内容是由上一层协议封装的数据。它最少是46字节，最大是1 500字节。如后面所述。
- CRC。最后字段为差错检测信息，在此情况下为CRC-32（见第10章）。

帧的长度

以太网对帧长度的最小值和最大值都有严格的规定。如图13.5所示。

长度最小值的限定是为了满足CSMA/CD正确运行的需要，我们简要地来看一下。以太网的帧必须有一个长度最小值：512位或64字节。长度的一部分是头部和尾部。如果我们将头部和尾部算为18字节（源地址6字节，目的地址6字节，长度或类型2字节及CRC 4字节），那么来自上层的数据的最小长度是64－18－46字节。如果上层分组小于46字节，相差部分将会被填充满。



图13.5 最小长度和最大长度

该标准定义帧长度的最大值（不包括前导符和SFD字段）为1 518字节。如果我们减去头部和尾部的18字节，有效载荷的最大长度是1 500字节。长度最大值的限定有两个历史原因。第一，当以太网被设计出来时，存储器很贵，最大长度的限制有助于减小缓冲器的大小。第二，长度最大值的限定防止了一个站点独占共享介质，而阻碍了其他要发送数据的站点。

帧的长度：最小值：64字节（512位）
最大值：1518字节（12 144位）

寻址

以太网中的每个站点（例如一台电脑，一个工作站或一台打印机）都有自己的网络接口卡（network interface card，NIC），简称为网卡。网卡安在站点内部并给该站点提供一个6字节的物理地址。如图13.6所示，以太网地址是6字节（48位）的，通常用十六进制表示法（hexadecimal notation）来表示，字节间以冒号断开。

06 : 01 : 02 : 01 : 2C : 4B

6字节=12十六进制数字=48位

图13.6 用十六进制表示法表示的以太网地址的一个示例

单播地址、多播地址和广播地址 源地址永远是一个单播地址——帧只来自一个站点。然而目的地址既可以是单播地址，也可以是多播地址，还可以是广播地址。图13.7说明了如何区分单播地址和多播地址。如果目的地址字段第一个字节的最低位是0，那么这个地址是单播地址。否则便是多播地址。

第一个字节的最低位决定了地址的类型。如果该位是0，地址是单播地址，否则便是多播地址。

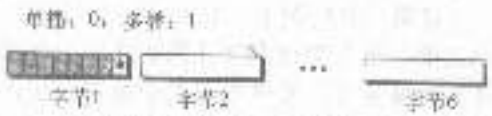


图13.7 单播地址和多播地址

单播目的地址限定了只有一个接收方，发送方和接收方之间的关系是一对一的。多播目的地址定义了一组地址，发送方和接收方之间的关系是一对多的。

广播地址是多播地址的一个特例，接收方是整个局域网中的所有站点。一个广播目的地址是48位1。

广播目的地址是多播地址的一个特例。它的所有位都是1。

例13.1

定义下列目的地址的类型：

a. 4A：30：10：21：10：1A

b. 47：20：1B：2E：08：EE

c. FF：FF：FF：FF：FF：FF

解：

为了知道地址的类型，我们必须看左边第二个十六进制数字。如果是偶数，那么地址是单播地址。如果是奇数，那么地址是多播地址。如果所有的数字都是F，那么地址是广播地址。因此，我们得到以下答案：

a. 这是一个单播地址，因为A在二进制中是1010（偶数）。

b. 这是一个多播地址，因为7在二进制中是0111（奇数）。

c. 这是一个广播地址，因为所有的数字都是F。

在线发送地址的方法与将地址用十六进制表示法表示的方法不同。传输是从左到右的，故对每个字节而言，最低位被最先发送而最高位被最后发送。这就意味着界定地址是单播地址还是多播地址的那一位最先到达接收方。

例13.2

请说明地址47：20：1B：2E：08：EE是如何在线发送的。

解：

地址被一个字节一个字节地从左向右发送，每个字节是一位一位地从右向左发送的。如下所示：

⇐11100010 00000100 11011000 01110100 00010000 01110111

访问方法：CSMA/CD

标准以太网使用1-持续的CSMA/CD方法（见第12章）。

时隙　在一个以太网络中，一个帧从最大长度网络的一个端到另一端之间往返所需的时间加上发送干扰序列所需的时间称为时隙。

时隙＝往返时间＋发送干扰序列所需时间

以太网的时隙以位来定义。它是一个站点发送512位所需的时间。这就意味着实际的时隙取决于数据速率；对于传统的10Mbps以太网来说是51.2μs。

时隙和冲突　选择512位的时隙并非偶然的。这个选择使得CSMA/CD能发挥正确的功能。为了理解这点，让我们来看两种情况。

在第一种情况中，我们假定发送方发送一个512位的最小分组。在发送方能将整个分组发出之前，信号穿过网络并到达网络的终端。如果在网络的终端有另一个信号（最坏的情况），冲突就发生了。发送方有机会放弃发送帧并发送一个干扰序列以提醒其他的站点。往返时间加上发送干扰序列所需的时间必须小于发送方发送最小的512位帧所需的时间。发送方必须在它发送整个帧之前意识到冲突的存在。否则，就太迟了。

在第二种情况中，发送方发送一个大于最小长度的帧（在512位和1 518位之间）。这种情况下，如果发送方已经发送了第一个512位并没有听到冲突，它便被保证在这个帧的传输过程中不会发生冲突。原因是信号会在小于1/2时隙的时间内到达网络的终端。如果所有的站点都遵循CSMA/CD协议，它们都已经检测到在线上这个信号的存在并制止了自身的发送。如果它们在1/2时隙时间期满之前发送了信号，冲突便发生了，而发送方也检测到了冲突。换言之，冲突只可能在时隙的前半段发生，如果冲突真的发生了，发送方在时隙期间能检测到。这就是说发送方发送第一个512位之后，它便被保证在这个帧的传输过程中不会有冲突发生。介质属于该发送方，而其他的站点不能使用介质。换言之，发送方只需要在发送第一个512位的时间内“听”是否有冲突发生就可以了。

当然，如果有一个站点不遵循CSMA/CD协议的话，所有这些假设都是不成立的。在这种情况下，不是发生了冲突，而是一个站点被损坏了。

时隙和最大网络长度 时隙和最大网络长度（冲突域）之间有关联，这取决于在一个特定的介质中信号的传播速度。在大多数传输介质中，信号以2×10^8m/s传播（是在空气中传播速度的2/3）。对传统的以太网而言，计算如下：

$$\text{最大长度}=\text{传播速度}\times(\text{时隙}/2)$$

$$\text{最大长度}=(2\times10^8)\times(51.2\times10^{-6}/2)=5\,120\text{m}$$

当然，我们需要考虑中继器和接口的延迟，及发送干扰序列所需的时间。这些将传统以太网的最大长度减少为2 500米，仅仅是理论上的48%。

$$\text{最大长度}=2\,500\text{米}$$

13.2.2 物理层

标准以太网定义了几种物理层的实现，如图13.8显示了其中最通用的四种。

编码和解码

所有标准的实现都使用10Mbps的数字信号（基带）。在发送方，使用曼彻斯特方案将数据转换成数字信号；在接收方，信号又被转译成曼彻斯特码并被解码成数据。如我们在第4章中看到的，曼彻斯特编码是自我同步的，在每一位的间隙提供一次转换。图13.9显示了标准以太网的编码图解。



图13.8 标准以太网的分类

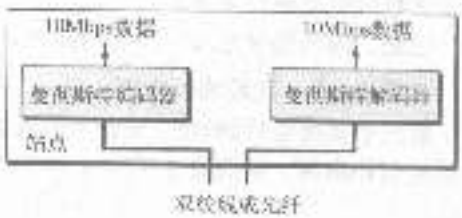


图13.9 标准以太网实现中的编码

10Base5：粗缆以太网

第一个实现被称为10Base5，**粗缆以太网**（thick Ethernet）或**粗缆网**（Thicknet）。这个别称是因电缆的粗细发展而来的，实际上粗略地说是橡胶软管的尺寸，并且非常坚硬不能用于弯曲。10Base5是第一个以太网规范，它使用一个有外部**收发器**（transceiver）（发送器/接收器）的总线拓扑，并通过一个外接口与粗缆电缆相连接。图13.10显示了一个10Base5实现的示意图。



图13.10 10Base5的实现

收发器负责传输，接收和检测冲突。收发器通过收发器电缆与站点连接，收发器电缆能为发送和接收提供独立的路径。这就意味着冲突只会发生在粗缆电缆中。

粗缆电缆的长度最大不能超过500米，否则，便会出现信号的过分衰减。如果长度必须超过500米，最多分五个分段，每个分段的最大长度是500米，可以使用中继器连接。中继器将

在第15章中介绍。

10Base2：细缆以太网

第二个实现是10Base2，细缆以太网（thin Ethernet），或便宜网络（Cheapernet）。10Base2使用总线拓扑，电缆细多了但很有弹性。电缆可以被弯曲以离站点很近，这种情况下，收发器通常是网卡（NIC）的一部分，被安装在站点内部。图13.11显示了一个10Base2实现的理论流程图。

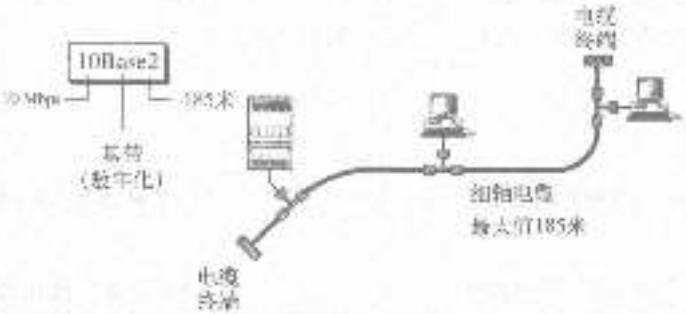


图13.11　10Base2的实现

注意：这种情况下冲突发生在细轴电缆中。这个实现比10Base5的成本效益高。因为细轴电缆比粗轴电缆便宜多了，且T型接口比分接头便宜。安装也因为细轴电缆的弹性变得简单多了。但是，因为细轴电缆中信号的高度衰减，每个分段的长度不能超过185米（接近200米）。

10Base-T：双绞线以太网

第三个实现是10Base-T或者叫做双绞线以太网（twisted-pair Ethernet）。10Base-T使用物理星型拓扑结构。站点通过双绞线连接到一个网络集线器上，如图13.12所示。

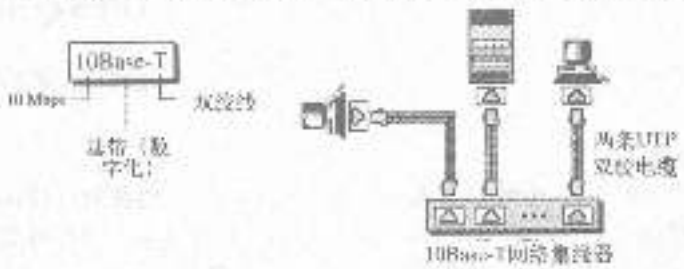


图13.12　10Base-T的实现

注意：双绞线在站点和网络集线器之间形成了两条路径（一条发送一条接收）。在这种情况下冲突发生在网络集线器中。与10Base5或10Base2相比较，可以看到就冲突而言，网络集线器实际上替代了同轴电缆的位置。双绞线的最大长度是100米，以使得双绞电缆的信号衰减最小。

10Base-F：光纤以太网

虽然10兆光纤以太网有很多种类型，但其中最普遍的还是10Base-F。10Base-F使用一种星型拓扑将站点与网络集线器相连接。如图13.13所示，站点使用两条光纤与网络集线器相连接。

总结

表13.1将标准以太网的实现做了一个总结。

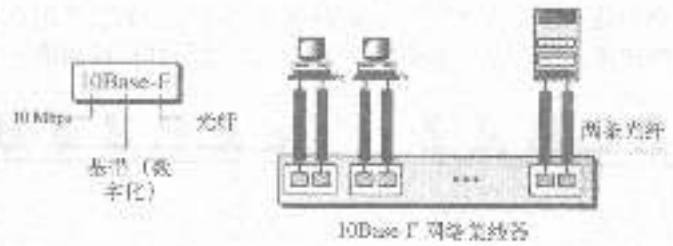


图13.13 10Base-F的实现

表13.1 标准以太网实现的总结

特 性	10Base5	10Base2	10Base-T	10Base-F
介质	粗同轴电缆	细同轴电缆	2条无屏蔽双绞线	2条光纤
最大长度	500米	185米	100米	2 000米
线路编码	曼彻斯特编码	曼彻斯特编码	曼彻斯特编码	曼彻斯特编码

13.3 标准的变化

10 Mbps标准以太网在运行更高的传输速率之前已经经历了一些变化。这些变化事实上为以太网的发展开辟了一条新路，使得与其他高传输速率的局域网相比，以太网变得更有竞争性。在这部分，我们介绍其中的一些变化。

13.3.1 桥接以太网

以太网发展的第一步是将局域网用网桥（bridge）分割。在以太局域网中，网桥有两个作用：提高带宽和分割冲突域。我们将在第15章中介绍网桥。

提高带宽

在非桥接以太局域网中，总能力（10 Mbps）被所有要发送帧的站点共享，这些站点共享网络的带宽。如果只有一个站点要发送帧，它受益于总能力（10 Mbps）。但是如果一个以上的站点要使用网络，能力就被共享了。例如，如果有两个站点要发送大量的帧，它们可能会轮流使用，当一个站点发送时，另一个站点停止发送。在这种情况下，我们可以认为平均每个站点发送的速度是5 Mbps。图13.14说明了这种情况。

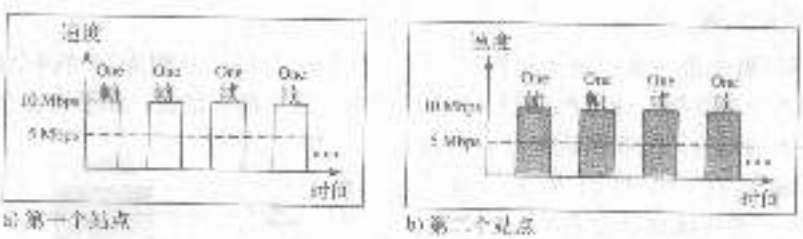
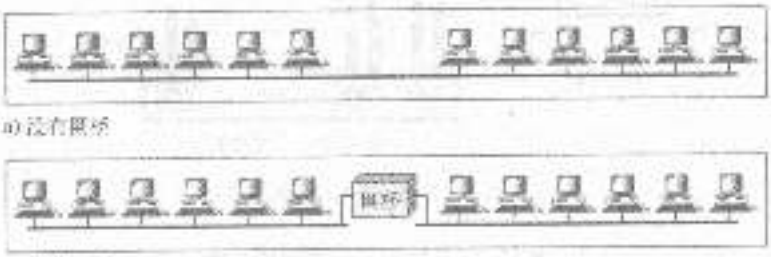


图13.14 带宽共享

我们将在第15章中学到的网桥能在这里起作用。一个网桥将网络分成两个或更多的网络。基于带宽，每个网络都是独立的。例如，在图13.15中，一个有12个站点的网络被分成两个网络，每个网络有6个站点，每个网络的能力都是10 Mbps。10 Mbps的能力在每个网络中被6个站点（实际上是7个，因为网桥在每个网络也算一个站点）所共享，而并非12个。在一个重负荷的网络中，假定流量并不经过网桥，每个站点理论上能达到10/6 Mbps而不是10/12 Mbps。

显而易见，如果我们进一步分割网络，我们能为每个部分得到更多的带宽。例如，如果我们使用一个四端口的网桥，每个站点能使用10/3 Mbps，这是非桥接网络的4倍。



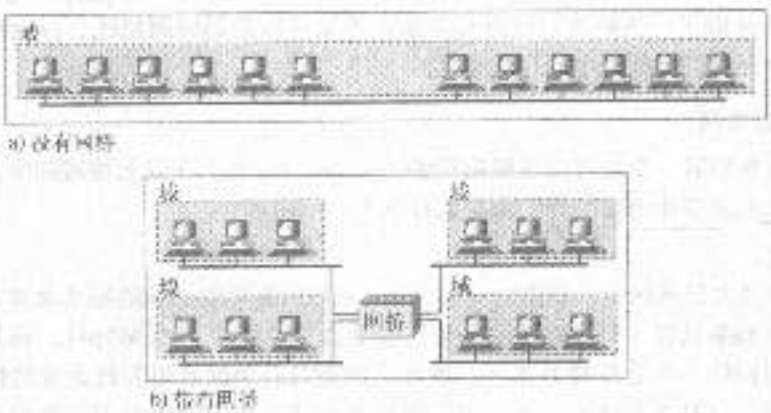
a) 没有网桥

b) 带有网桥

图13.15　一个带有网桥的网络和不带网桥的网络

分割冲突域

网桥的另一个优势就是分割**冲突域**（collision domain）。图13.16表示对于一个桥接网络和一个非桥接网络的冲突域的情况。从图中可以看到，在有桥接的情况下，冲突域会变得更小而且冲突的概率大幅度减少。如果没有桥接，则有12个站点竞争访问介质。有了桥接之后，就只有3个站点竞争访问。



a) 没有网桥

b) 带有网桥

图13.16　一个带有网桥和不带有网桥网络的冲突域

13.3.2　交换式以太网

桥接LAN的概念进一步扩展为交换LAN。与其将一个网络分割为2个到4个网络，为什么不将其分割成N个网络呢？这里的N是LAN上站点的个数。换句话说，如果有多个端口的网桥，为什么不能有N个端口的交换机呢？按这种说法，带宽仅由站点和交换机共享（每个5Mbps）。另外，冲突域也就分为N个。

一个2层**交换机**（switch）就是一个N个端口的网桥，该网桥具有允许快速处理分组的附加功能。从桥接以太网到**交换式以太网**（switched Ethernet）的发展对于并发更快的以太网是一个大的进步，正如我们将会看到的那样。图13.17表示一个交换LAN。

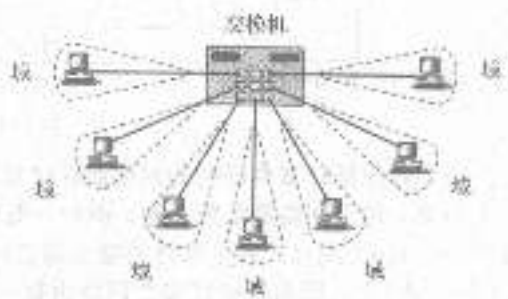


图13.17　交换式以太网

13.3.3 全双工以太网

10Base5和10Base2的一个局限就是它们的通信是半双工的（10Base-T始终是全双工的）。一个站点要么发送数据要么接收数据，但是不能同时进行。下一步就是从交换式以太网到全双工交换式以太网（full-duplex switched Ethernet）发展，全双工模式将每一个端的能力从10Mbps增加到20Mbps。图13.18表示了在全双工模式下的交换式以太网。注意，在全双工模式的交换以太网中，在站点与交换机之间不是使用一条链路，而是使用两条链路：一个用于传送，另一个则用于接收。

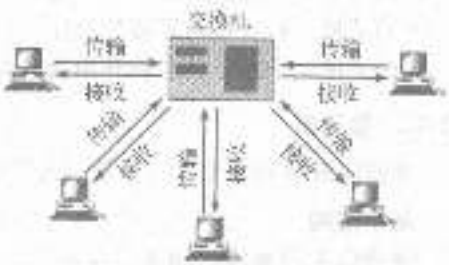


图13.18 全双工交换式以太网

不需要CSMA/CD

在全双工交换式以太网中，不需要CSMA/CD方法。在一个全双工交换式以太网中，每个站点都是通过两条分离的链路连接到交换机的。每个站点或交换机都能够独立地发送或接收而不必考虑冲突。在站点和交换机之间的每一条链路都是点到点的专用链路。它们不再需要载波检测，也不再需要冲突检测。因此MAC层的工作将变得更加容易。MAC子层的载波检测和冲突检测的功能可以不要了。

MAC控制层

标准以太网被设计为在MAC子层的无连接的协议。没有显式的流量控制和差错控制通知发送方帧已经无误地到达目的地。在接收方接收到帧后，它不必发送肯定或否定的确认。

为了在全双工交换式以太网中提供流量控制和差错控制，在LLC子层和MAC子层之间又增加了一个新的子层叫做MAC控制。

13.4 快速以太网

设计快速以太网是为了与诸如FDDI或光纤通道等的局域网协议相竞争。IEEE在名为802.3u下创造了快速以太网。快速以太网是标准以太网的向后兼容，但是它的传输速度是100 Mbps，快了10倍。快速以太网的目标是：

1. 将数据速率升级为100Mbps。
2. 使它能与标准以太网兼容。
3. 保留48位地址。
4. 保留相同的帧格式。
5. 保留帧长度的最大值和最小值。

13.4.1 MAC子层

使以太网从10Mbps向100Mbps过渡的总的思想是不触及MAC子层。所做出的决定是放弃总线拓扑而只保留星型拓扑。在星型拓扑中，正如我们之前介绍的，有两种选择：半双工和全双工。在半双工方法中，站点通过集线器连接；在全双工方法中，通过每个端口都带有缓冲区的交换机来进行连接。

对于半双工方法来说，访问方法是相同的（CSMA/CD）。对于全双工快速以太网而言，CSMA/CD是不必要的。然而，在实现时还是保留了CSMA/CD，以使于和标准以太网向后兼容。

自动协商

快速以太网上增加的一个新特性就是自动协商（autonegotiation）。它允许一个站点或一

个集线器有一定的能力范围。自动协商允许两个设备协商它们的运行模式和传输速率。它的设计更是处于以下的考虑：

- 为了使不兼容的设备间相互连接。例如，一个带有10Mbps的设备可以和带有100Mbps的设备（在低速率下工作的）通信。
- 为了使一种设备有更多的功能。
- 为了使一个站点具有检测集线器的能力。

13.4.2 物理层

快速以太网的物理层比标准以太网复杂得多。我们随后将简要地加以介绍。

拓扑结构

快速以太网被设计用来连接两个或两个以上的站点。如果只有两个站点，它们可以是点到点的连接。三个或三个以上的站点就需要中间有一个集线器或交换机来连接，星型拓扑结构，如图13.19所示。

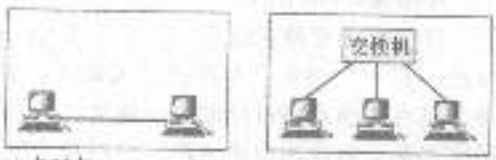


图13.19　快速以太网拓扑

实现

快速以太网在物理层的实现方式可分为两线或四线的。两线的实现可以是5类UTP（100Base-TX）或者光纤（100Base-FX）。而四线的实现则是仅有3类UTP（100Base-T4）。如图13.20所示。



图13.20　快速以太网的实现

编码

为了100Mbps的传输速率曼彻斯特编码需要200兆波特的带宽，这使得诸如双绞线这样的介质并不合适。因此，快速以太网的设计者寻求了其他可替代的编码/解码方案。然而，发现对于三种实现而言，一种编码方案的表现并不是同样的好。出此，选择了二种不同的编码方案（见图13.21）。

100Base-TX　使用两条双绞线（5类UTP或STP）。对于这种实现而言，基于它有很好的带宽性能（见第4章），选择MLT-3编码方案。然而，MLT-3却不是一个自我同步的线编码方案，故使用4B/5B块编码来实现位的同步并阻止了一长串0和1的发生（见第4章）。插入MLT-3编码产生了125Mbps的传输速率。

100Base-FX　使用两条双光纤。通过使用简单的编码方案，光纤很容易就能满足高带宽的需求。100Base-FX的设计者为这个实现选用了NRZ-I编码方案（见第4章）。然而，NRZ-I对于一长串的0（或者1，取决于编码）存在位同步的问题。正如我们第4章讨论的那样。为解决这一问题，设计者使用了与100Base-TX 中相同的4B/5B块编码。块编码将比特率从100Mbps提高到125Mps，但对于光纤而言这很容易解决。

100Base-TX　网络能提供100Mbps的传输速率，但需要使用5类UTP或者STP电缆。这对于已经安装了音级双绞线（3类）的建筑物成本效益较低。一个称为100Base-T4的新标准是为了使用3类或更高的UTP而设计的。为达到传输速度100Mbps这个实现要用四条UTP。100Base-T4的编码/解码也复杂得多。因为该实现使用3类UTP，每条双绞线都不能简单应对超过25M波特的带宽。在这个设计中，在发送和接收之间有一对交换机。然而，三条3类UTP也

只能处理75M波特（每条25M波特）。我们需要使用一个编码方案将100Mbps转换成75M波特的信号。正如我们在第4章中所介绍的，8B/6T满足了这一需求。在8B/6T中，8个数据元素被编码成6个信号元素。这就意味着100Mbps只需要（6/8）×100Mbps，即75M波特。

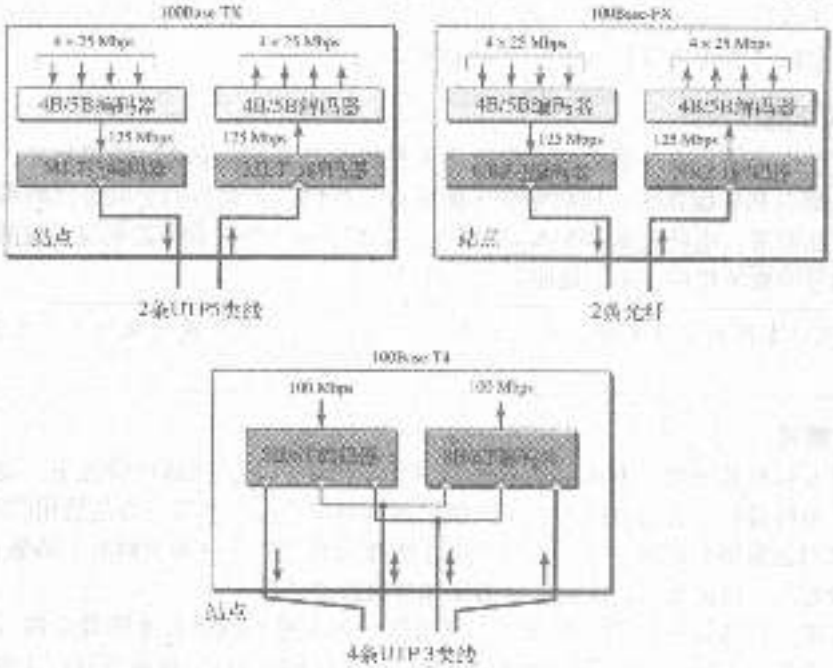


图13.21 快速以太网实现的编码

总结

表13.2是对快速以太网实现的一个总结。

表13.2 快速以太网实现的总结

特 性	100Base-TX	100Base-FX	100Base-T4
介质	5类 UTP或STP	光纤	4类 UTP
电线数量	2	2	4
最大长度	100米	100米	100米
块编码	4B/5B	4B/5B	
线编码	MLT-3	NRZ-I	8B/6T

13.5 千兆以太网

对传输速度更高的需求使得千兆以太网（1 000Mbps）应运而生。IEEE委员会称之为标准802.3z。千兆以太网设计的目标可总结如下：

1. 将数据速率升级到1千兆。
2. 使其与标准以太网或快速以太网相兼容。
3. 使用相同的48位地址。
4. 使用相同的帧格式。
5. 保留帧长度的最大值和最小值。
6. 支持快速以太网中定义的自动协商。

13.5.1 MAC子层

有关以太网发展的总的思想是尽量不触及到MAC子层，但在实现1Gbps的速率发送时，这种想法就不现实了。千兆以太网在介质访问方面有两个独特的方法：半双工或全双工方法。几乎所有的千兆以太网的实现都采用了全双工方法。然而，我们会简要地介绍半双工方法以说明千兆以太网与前几代以太网是可以兼容的。

13.5.2 全双工模式

在全双工模式中，有一个中心交换机将所有的电脑或其他交换机连接起来。每个交换机的每个进入端口都有缓存区，使数据在传输前得以存储。正如我们之前讨论的那样，在这种模式中不存在冲突。也就是说CSMA/CD是不必要的。缺少冲突意味着电缆长度的最大值取决于电缆中信号的衰减程度，而不是冲突检测过程。

> 在千兆以太网的全双工模式中，不存在冲突；电缆长度的最大值取决于电缆中信号的衰减程度。

半双工模式

千兆以太网也能使用半双工模式，虽然事实上很少用到。在这种情况下，交换机被集线器所替代。集线器作为普通电缆的一部分便可能有冲突产生。半双工方法使用CSMA/CD。然而，正如我们之前所介绍的，这种方法下网络的最大长度完全取决于帧大小的最小值。于是，定义了三种方法：传统方法，载波扩展方法和帧突发方法。

传统方法　在传统方法中，保留了与传统以太网中相同的帧长度的最小值（512位）。然而，因为在千兆以太网中一位的长度是10Mbps以太网中的1/100，因此千兆以太网的时隙时间是512位×1/1 000μs，即为0.512μs。时隙时间的减短意味着能快100倍的时间检测到冲突。这就是说网络的最大长度是25米。这个长度也许对在一个房间内的所有站点是合适的，但是它也许还不足长到可以连接一个办公室的所有电脑。

载波扩展方法　为了适应更长的网络，我们增加了帧长度的最小值。载波扩展方法定义的帧长度的最小值是512字节（4 096位）。这意味着长度最小值是以前的8倍。这个方法使得站点给不满4 096位的帧增加了扩展位（填充）。在此方法中，网络长度的最大值也扩大了7倍，即200米。这就允许从集线器到站点的长度可以是100米。

帧突发方法　如果我们要发送的帧都比较短，那么载波扩展方法是非常低效的，因为每个帧都承载了无余的数据。为了提高效率，提出了帧突发方法。在此方法中，发送成倍的帧而不是给每个帧增加扩展。然而，为了使成倍的帧看上去像一个帧，要在帧间加以填充（这与载波扩展方法中的一样）而使通道不会空闲下来。换言之，这个方法使得其他的站点认为有一个非常大的帧正在传输。

13.5.3 物理层

千兆以太网的物理层比标准以太网或者快速以太网的物理层复杂得多。在此我们仅简要地讨论其中的某些特性。

拓扑结构

千兆以太网被设计用来连接两个或两个以上的站点。如果只有两个站点，它们可以是点到点的连接。三个或三个以上的站点就需要有一个集线器或交换机来连接，呈星型拓扑结构。另一种可以的配置是将几个星型拓扑连接起来或者使得一个星型拓扑是另一个星型拓扑结构的一部分。如图13.22所示。

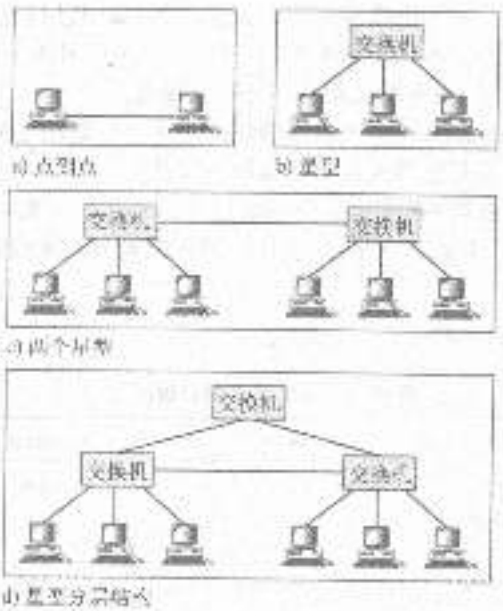


图13.22 千兆以太网的拓扑结构

实现

千兆以太网可以分为两线或四线的实现。两线的实现使用光纤（1000Base-SX，短波或1000Base-LX，长波）或STP（1000Base-CX）。四线的实现使用5类双绞线电缆（1000Base-T）。换言之，可以有四种实现，如图13.23所示。1000Base-T的设计是为了满足某一类用户，这类用户已经为其他诸如快速以太网或电话服务等目的进行了线路安装。



图13.23 千兆以太网的实现

编码

图13.24描述了这四种实现的编码/解码方案。

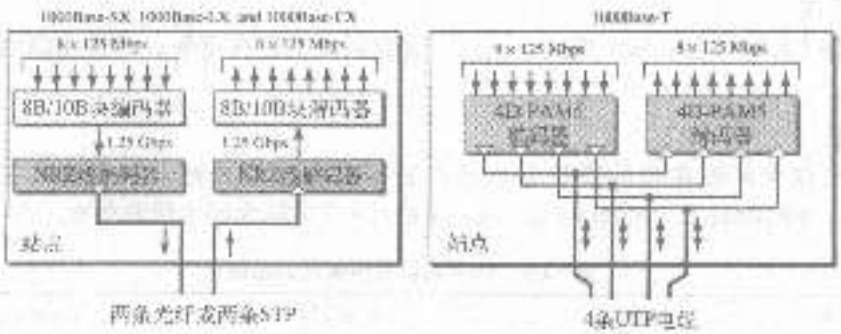


图13.24 千兆以太网实现中的编码

千兆以太网由于涉及的带宽太大（2G波特），故不能使用曼彻斯特编码方案。两线实现使用NRZ方案，但是NRZ无法正确地进行自我同步。为了实现位的自我同步，特别是为了适应

如此高的传输速率，使用了8B/10B块编码，该方法在第4章中已经介绍过了。

块编码避免了在数据流中出现长串的0或1，但产生的数据流是1.25Gbps。注意在此实现中，一条线路（光纤或STP）用来发送而另一条用来接收。

在四线实现中，设为每条线路要承载500Mbps，这已经超出了5类UTP的承载能力，故更不可能让两条线路专门发送而让另两条专门接收。作为一个解决方案，使用了4D-PAM5编码方法来减小带宽，该方法在第4章中也已经介绍过了。该实现中的4条线路既用来输出又可用来输入，每条线路承载250Mbps，这是在5类UTP的承载能力范围内的。

总结

表13.3对千兆以太网的实现作了一个总结。

表13.3　千兆以太网实现的总结

特　性	1000Base-SX	1000Base-LX	1000Base-CX	1000Base-T
介质	光纤 短波	光纤 长波	STP	5类UTP
线路数目	2	2	2	4
长度最大值	550米	5 000米	25米	100米
块编码	8B/10B	8B/10B	8B/10B	
线编码	NRZ	NRZ	NRZ	4D-PAM5

13.5.4　10千兆以太网

IEEE委员会创造了10千兆以太网并称之为标准802.3ae。10千兆以太网的设计目标可总结如下：

1. 将数据速率提升为10Gbps。
2. 使其与标准以太网、快速以太网和千兆以太网相兼容。
3. 使用相同的48位地址。
4. 使用相同的帧格式。
5. 保留帧长度的最大值和最小值。
6. 允许将现有的局域网与城域网（MAN）或广域网（WAN）相互连接。
7. 使得以太网与诸如帧中继和ATM（见第18章）等的技术相兼容。

MAC子层

10千兆以太网只在全双工模式下运行，这意味着不存在竞争；10千兆以太网也不使用CSMA/CD。

物理层

10千兆以太网物理层的设计目标是在长距离内使用光纤。最常见的三种实现是：10GBase-S，10GBase-L，10GBase-E。表13.4是对10千兆以太网实现的总结。

表13.4　10千兆以太网实现的总结

特　性	10GBase-S	10GBase-L	10GBase-E
介质	短波 850-nm 多模式	长波 1310-nm 单一模式	超长波 1550-nm 单一模式
长度最大值	300米	10公里	40公里

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物。在本书末列出括号[……]中的参考资料。

读物

以太网在[For03]的第10章、第11章和第12章，[Ken02]的第5章，[Tan03]的第4.3节，[Sta04]的第15章、第16章中有讨论。[Spu00]是关于以太网的书。千兆位以太网的完整讨论可以在[KCK98]和[Sau98]中找到。[Izz00]的第2章对不同代的以太网有很好的比较。

本章小结

- 以太网是使用最广泛的局域网协议。
- IEEE802.3标准定义了1-持续CSMA/CD是第一代10兆以太网的访问方法。
- 以太网的数据链路层由LLC子层和MAC子层组成。
- MAC子层负责CSMA/CD访问方法的运行和成帧。
- 以太网中的每个站点都有烙印在其网卡（NIC）上的唯一的48位地址。
- 10兆以太网中帧长度的最小值是64字节，最大值是1 518字节。
- 10兆以太网中常见的实现是10Base5（粗缆以太网），10Base2（细缆以太网），10Base-T（双绞线以太网）和10Base-F（光纤以太网）。
- 10Base5以太网实现使用粗同轴电缆，10Base2使用细同轴电缆。10Base-T使用四根双绞线将每个站点与一个公共集线器连接。10Base-F使用光纤。
- 网桥可以提升以太网局域网的带宽并分离冲突域。
- 交换机使得以太网局域网中的每个站点都可以让自己获得整个网络的传输能力。
- 全双工模式将每个域的传输能力翻倍并取消了对CSMA/CD访问方法的需求。
- 快速以太网的数据速率是100Mbps。
- 在快速以太网中，自动协商使得两个设备能协商运行的模式和数据速率。
- 常见的快速以太网的实现是100Base-TX（双线双绞线），100Base-FX（双光纤）和100Base-T4（音级或更高级别的双绞线）。
- 千兆以太网的数据速率是1 000Mbps。
- 千兆以太网的访问方法包括使用传统CSMA/CD方法的半双工模式（不常见）和全双工模式（最盛行的方法）。
- 常见的千兆以太网的实现是1 000Base-SX（双光纤和短波激光源），1 000Base-LX（双光纤和长波激光源）和1 000Base-T（四线双绞线）。
- 最新的以太网标准是10千兆以太网，其运行速率是10Gbps。常见的三种实现是：10GBase-S，10GBase-L和10GBase-E。这些实现都使用全双工模式下的光纤。

习题

复习题

1. 如何区别前导符字段和SFD字段？
2. 使用网卡的目的是什么？
3. 单播地址、多播地址和广播地址之间的区别是什么？
4. 用网桥分割以太网局域网的优势是什么？
5. 交换机和网桥之间的关系是什么？

6. 全双工以太网总域网为何不需要CSMA/CD?
7. 比较标准以太网、快速以太网、千兆以太网和10千兆以太网的数据速率。
8. 标准以太网最常见的实现是什么?
9. 快速以太网最常见的实现是什么?
10. 千兆以太网最常见的实现是什么?
11. 10千兆以太网最常见的实现是什么?

练习题

12. 下列以太网地址用十六进制如何表示?

01011010 00010001 01010101 00011000 10101010 00001111

13. 以太网地址1A：2B：3C：4D：5E：6F的二进制是什么?
14. 如果一个以太网地址是07：01：02：03：04：05，这个地址的类型是什么（单播、多播还是广播）?
15. 地址43：7B：6C：DE：10：00是一个用以太网中的帧表示的源地址，但是接收方舍弃了这个帧，为什么?
16. 一个以太网MAC子层从上层接收到42个字节的数据。这个数据需要填充多少字节?
17. 一个以太网MAC子层从上层接收到1 510个字节的数据。这些字节能压缩在一个帧内吗?如果不可以的话，那么要发送多少帧?每个帧中的数据大小是多少?
18. 最小的以太网帧中有用数据与整个分组的比例是多少?最大的以太网帧中，该比例又是多少?
19. 假定10Base5电缆的长度是2 500米。如果粗缆电缆的传播速率是200 000 000米/秒，一个位从网络的一端到另一端需要多长时间?假定设备有10μs的延迟。
20. 10Base5的数据速率是10Mbps。产生最小的帧需要多少时间?列出计算过程。

第14章 无线局域网

无线通信是发展最快的技术之一。无论在什么地方，不用电缆来连接设备的需求都在与日俱增。**无线局域网**（wireless LAN）在校园、办公楼内和公共场合随处可见。

本章集中讨论两种非常有前途的无线LAN技术：IEEE 802.11无线局域网（有时也称做无线以太网）和蓝牙技术——一个用于小型无线局域网的技术。虽然这两种协议需要在几个层中运行，但本书集中讨论物理层和数据链路层。

14.1 IEEE 802.11

由IEEE定义的无线局域网规范称做IEEE 802.11，该规范涵盖了物理层和数据链路层。

14.1.1 体系结构

标准定义了两种类型的服务：基本服务集（BSS）和扩展服务集（ESS）。

基本服务集

IEEE 802.11将**基本服务集**（basic service set，BSS）定义为无线局域网的积木块。一个基本服务集由固定的或移动的无线站点和一个选的、称为**访问点**（access point，AP）的中央基站组成，图14.1说明了这一标准的两个集合。

不带AP的BSS是一个独立网络，它不能向其他BSS发送数据，这就是它被称做特别的体系结构（*ad hoc architecture*）的原因。在这种体系结构中，站点能够构成网络而不需要AP，它们能够相互定位并允许是BSS的一部分。带AP的BSS有时称为基础设施（*infrastructure*）网络。

> 不带AP的BSS称为特别网络，带AP的BSS称为基础网络。



图14.1 基础服务集（BSS）

扩展服务集

扩展服务集（extended service set，ESS）由两个或更多个带有AP的BSS组成。在这种情况下，通过一个分布式系统（通常是一个有线局域网）将各个BSS连接在一起。分布式系统连接BSS中的AP。IEEE 802.11对分布式系统并没有严格的限制。它可以是任何的IEEE局域网，如以太网等。注意，扩展服务集使用两种类型的站点：移动的和固定的。移动站点BSS内的一个普通站点。而固定站点则是一个属于有线局域网一部分的AP站点。图14.2显示一个ESS。

在连接BSS时，互相可达的站点之间可以相互通信而不必使用AP。但是，位于两个不同BSS的两个站点间的通信一般要跨过两个AP。这一概念与移动电话网络中的通信类似。这时

可以将每一个BSS看做是一个信元而将每一个AP看做是一个基站。注意，一个移动站点可以同时属于多个BSS。

ESS：扩展服务集
BSS：基本服务集
AP：协同点

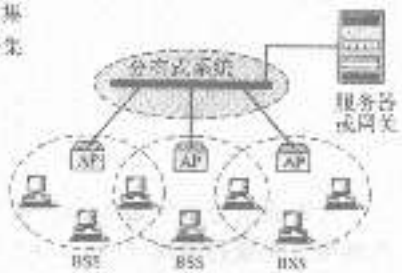


图14.2　扩展服务集（ESS）

站点类型

根据站点在无线局域网中的移动性，IEEE 802.11将站点定义为三种类型，即不迁移（no-transition）、BSS迁移（BSS-transition）和ESS迁移（ESS-transition）。具有不迁移移动性的站点或者是一个固定（不移动）的，或者是仅在BSS内部移动的站点。具有BSS迁移移动性的站点可以从一个BSS移动到另一个BSS，但是其移动被限制在一个ESS之内。具有ESS迁移移动性的站点可以从一个ESS移动到另一个ESS，但是IEEE 802.11不能保证通信在移动中是连续的。

14.1.2　MAC层

IEEE 802.11定义了两个MAC子层：分布式协调功能（DCF）和点协调功能（PCF），见图14.3显示了两个MAC子层、LLC子层和物理层之间的关系。我们现在集中讨论MAC子层，在本章后面会讨论物理层实现。



图14.3　IEEE 802.11标准中的MAC层

分布式协调功能

IEEE在MAC子层定义的两种协议之一称为分布式协调功能（distributed coordination function，DCF），DCF使用CSMA/CA（已在第12章中定义）作为访问方式。无线LAN因为下面三个原因不能实现CSMA/CD。

1. 为了冲突检测，站点必须能够同时发送数据和接收冲突信号，这就意味着建立站点的费用很高，并且增加了带宽需求。

2. 因为隐藏的站点问题，可能检测不到冲突。我们会在本章稍后讨论这个问题。

3. 站点间的距离可以很大。信号衰减会使得一端的站点无法侦听到另一端的冲突。

处理流程图　图14.4说明了用于无线LAN的CSMA/CA的处理流程图，我们会简要解释这些步骤。

帧交换时序　图14.5表示按时间进行的数据交换和控制帧。

1. 在发送一个帧之前，源站点通过检测载波频率的能量级来侦听介质。

a. 在通道空闲之前，通道使用带有补偿的持续策略。

b. 站点发现通道空闲之后，它等待一个称做分布式帧间间隔（distributed interframe space，DIFS）的时间周期，然后就发送一个叫做请求发送（RTS）的控制帧。

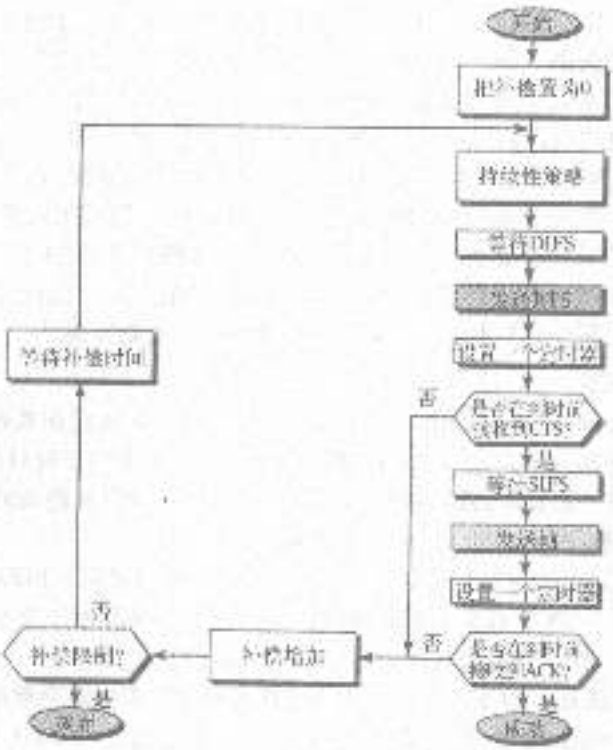


图14.4 CSMA/CA流程图

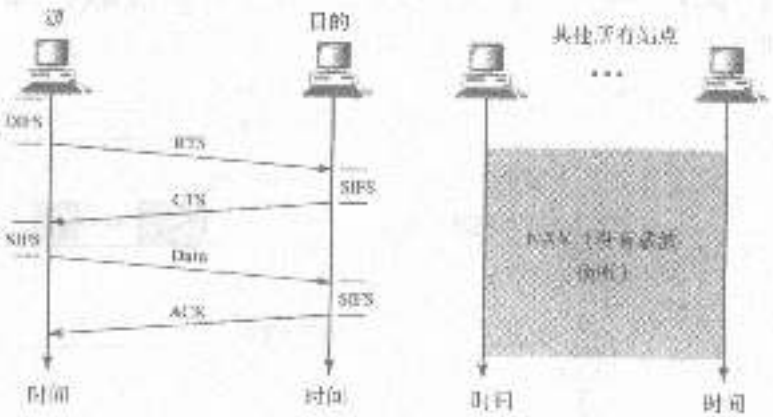


图14.5 CSMA/CA与NAV

2. 接收到RTS并等待一个称做短帧间间隔（short interframe space，SIFS）的短暂时间后，目的站点就向源站点发送一个叫做清除发送（CTS）的控制帧，这一控制帧表示目的站点准备接收数据。

3. 在等待一个与SIFS相等的时间后，源站点就发送数据。

4. 在等待一个与SIFS相等的时间后，目的站点发送确认说明已经接收到帧。在这个协议中，由于没有任何方法检查它的数据是否已经成功地到达目的站点，所以确认还是需要的。另一方面，在CSMA/CD中，没有冲突就是向源站点表示数据已经到达。

网络分配矢量 如果一个站点获得访问，那么其他站点如何推迟发送它们数据的时间呢？换句话说，这一协议的冲突避免是如何实现的？其关键是一个叫做NAV的特性。

当一个站点发送RTS帧时，它包含了需要占据通道的时间。受这一传送所影响的站点建立一个叫做网络分配矢量（network allocation vector，NAV）的定时器，该定时器指出在允许这些站点检测通道是否空闲之前还将需要经过多长时间。每当一个站点访问系统并且发送一个RTS帧时，其他站点就启动它们的NAV。换句话说，对任何一个站点，在检查物理介质是否空闲之前，首先要检查它的NAV是否过期。图14.5说明了NAV的概念。

握手时的冲突　在RTS或CTS控制帧正在发送时，这一期间叫做握手周期（handshaking period），如果产生了冲突会发生什么情况？两个或更多的站点会在同一时刻试图发送RTS帧。这些控制帧可能会相互冲突。但是，由于没有检测冲突的机制，因此发送方在它没有从接收方收到CTS时就会以为产生了冲突，于是采取补救策略并重新发送。

点协调功能（PCF）

点协调功能（point coordination function，PCF）是一种可以在基础设施网络中（不在特别网络中）实现的可选访问方式。它在DCF的上面实现，主要用于对时间敏感的传输。

PCF是集中式的、无竞争的轮询访问方式。AP对那些可以被轮询的站点进行轮询。站点依次被轮询，将数据发送给AP。

为了给予PCF高于DCF的优先级，已定义了另一套帧间间隔：PIFS和SIFS。SIFS与DCF中的一样，但是PIFS（PCF IFS）比DIFS短。这意味着如果同时一个站点想只使用DCF，而一个AP想使用PCF，那么AP有优先级。

由于PCF的优先级高于DCF，只使用DCF的站点可能得不到对介质的访问。为了避免这种情况，设计了重复间隔来覆盖无竞争（PCF）和基于竞争（DCF）的通信。**重复间隔**（repetition interval），会持续地重复，并始于一个称为**信号帧**（beacon frame）的特殊控制帧。当站点听到信号帧时，在重复间隔的无竞争周期内它们并始它们的NAV。图14.6给出了重复间隔的一个例子。

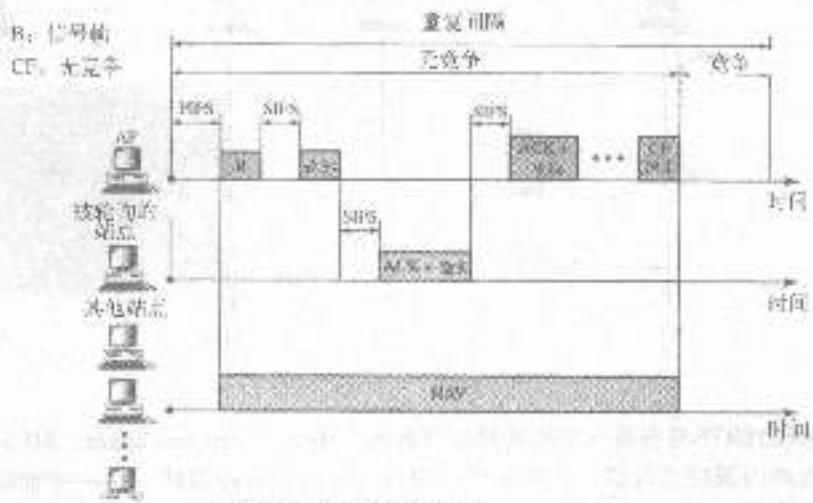


图14.6　重复间隔的例子

在重复间隔中，PC（点控制方）可以发送轮询帧、接收数据、发送ACK、接收ACK或者做任何这些动作（802.11使用捎带）的组合。在无竞争周期结束时，PC发送CF结束（无竞争）帧允许基于竞争的站点能使用介质。

分段

无线环境噪声很多，被损坏的帧必须进行重传。因此，协议建议要分段，就是将一个大

的帧分割为多个更小的帧，这种用更小的帧代替大的帧的方法在传输时更加有效。

帧格式

MAC层包含9个字段，见图14.7。

- **帧控制（FC）**。FC字段为2字节，它定义了帧的类型和一些控制信息。表14.1描述了它的子字段。在本章的后面我们要讨论每一个帧类型。



图14.7 帧格式

表14.1 FC字段中的子字段

字 段	说 明
协议版本	当前版本是0
类型	信息类型：管理（00）、控制（01）或数据（10）
子类型	定义每一个类型中的子类型（见表14.2）
To DS	后面有定义
From DS	后面有定义
更多标记（More flag）	置1表示有更多段
重传（Retry）	置1表示重发帧
Pwr mgt	置1，表示站点处于电源管理模式
更多数据（More data）	置1，表示站点有更多的数据要发送
WEP	有线等效保密（实现加密）
Rsvd	保留

- **D**。在除了一个类型以外的所有帧类型中，本字段定义了用于设置NAV值的传输间隔时间。在一个控制帧中，这个字段用来定义帧的ID。
- **地址**。有4个地址字段，每个有6字节。每一个地址字段的含义取决于*to DS*和*from DS*子字段的值，后面要讨论。
- **序列控制**。本字段定义帧的序列号，以用于流量控制。
- **帧主体**。长度可以在0～2 312字节之间。包含了根据FC字段中定义的类型和子类型的信息。
- **FCS**。长度为4字节，包含一个CRC-32的差错检测序列。

帧类型

由IEEE 802.11定义的无线局域网有三种类型的帧，它们是管理帧、控制帧和数据帧。

管理帧 管理帧用于在站点和接入点之间初始化通信。

控制帧 控制帧用于访问通道和对帧的确认。图15.10说明了它的格式。

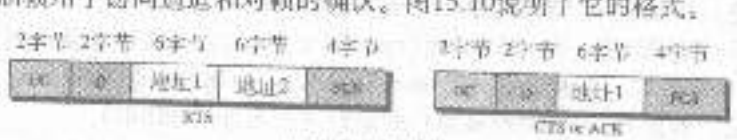


图14.8 控制帧

对于控制帧来说，类型字段的值是01，帧的子类型字段的值如表14.2所示。

数据帧　数据帧用于携带数据与控制信息。

表14.2　控制帧中子类型字段的值

子 类 型	含　义
1011	请求发送（RTS）
1100	清除发送（CTS）
1101	确认（ACK）

14.1.3　寻址机制

IEEE 802.11的寻址机制说明了四种情况，由FC字段的两个标记（*From DS*和*To DS*）定义，每一个标记可能是0或1，于是就定义了四种不同的情况。MAC帧的四种地址（地址1～地址4）的解释取决于这些标记的值，见表14.3。

表14.3　地址

To DS	From DS	地址1	地址2	地址3	地址4
0	0	目的地址	源地址	BSS ID	N/A
0	1	目的地址	发送AP	源地址	N/A
1	0	接收AP	源地址	目的地址	N/A
1	1	接收AP	发送AP	目的地址	源地址

注意：地址1总是下一个设备的地址，而地址2总是前一个设备的地址，如果最后的目的站点没有被地址1定义的话，地址3就是最后目的站点的地址。如果起始源站点的地址与地址2不同的话，那么地址4就是起始信源站点的地址。

- 情况1：00 在情况1下，*To DS* = 0并且*From DS* = 0，这表明该帧既不是发往一个分布式系统（*To DS* = 0）也不是来自一个分布式系统中（*From DS* = 0）。该帧是从BSS中的一个站点到另一个站点，而不经过分布式系统，需要给原始的发送方站点发送ACK帧。地址见图14.9。

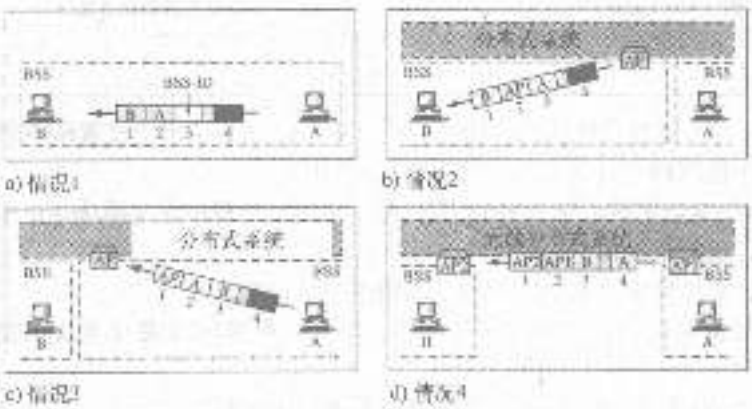


图14.9　寻址机制

- 情况2：01 在情况2下，*To DS* = 0而*From DS* =1，这表示该帧从一个分布式系统中来（*From DS* = 1），该帧从一个AP来，到一个站点去。ACK帧将被发送到AP。地址见图14.9。注意，地址3包含该帧的原始发送方地址（在另外的BSS中）。
- 情况3：10 在情况3中，*To DS* = 1而*From DS* = 0，这表示该帧要发送到一个分布式系统中去（*To DS* = 1）。该帧是从一个站点发往AP。ACK将被发送到原始站点。地址见图14.9。注意，地址3包含该帧的最终目的地址（在另外的BSS中）。
- 情况4：11 在情况4中，*To DS* = 1并且*From DS* = 1。这种情况说明分布式系统也是无线

的。帧在一个无线分布式系统中从一个AP发送到另一个AP，如果分布式系统是一个有线局域网，这里就不需要定义地址，因为在这些情况下帧的格式就是有线局域网帧（如以太网）的格式。这里需要四个地址定义原始发送方、最终的目的地和两个中间AP。图14.9说明了这一情况。

隐藏站点问题和暴露站点问题

我们在前面已经提到了隐藏站点问题和暴露站点问题，现在对这些问题以及它们的影响进行讨论。

隐藏站点问题 图14.10给出了隐藏站点问题的一个例子。站点B有一个传输范围，由左边椭圆示出；这个范围中的每一个站点都能听到站点B传送的信号。站点C有一个传输范围，由右边椭圆示出；这个范围中的每一个站点都能听到站点C传送的信号。站点C在站点B的传输范围之外，站点B在站点C的传输范围之外，但是，站点A既在站点B的范围之内，也在站点C的范围之内，它能听到B或C传送的任何信号。

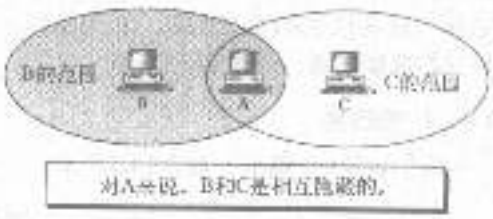


图4.10 隐藏站点问题

假定站点B发送数据给站点A。在这个传输中，站点C也有数据要发送给站点A。但是，站点C在B的范围之外，来自B的传输不能到达C。因此，C认为介质是空闲的。站点C发送它的数据给A，这就导致了冲突，因为这个站点同时收到来自B和C的数据。在这种情况下，我们说站点B和C对于A来说是互相隐藏的。因为冲突的可能性，隐藏站点降低了网络的能力。

隐藏站点问题的解决方法是使用我们前面提到的握手帧（RTS和CTS）。图14.11显示来自B的RTS消息到达A，而没有到达C。但是，因为B和C都在A的范围内，CTC消息（包含从B到A的数据传输期间）到达了C，站点C知道一些隐藏站点正使用通道，就限制传输直到这个期间结束。

> CSMA/CA握手中的CTS帧可以避免来自隐藏站点的冲突。

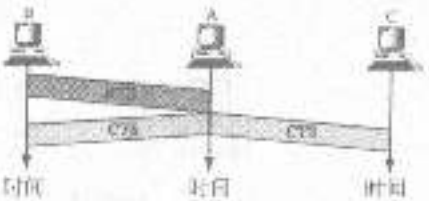


图14.11 握手的使用来避免隐藏站点问题

暴露站点问题 现在考虑前一种情况的相反情况：暴露站点问题。在这个问题中，站点被限制使用通道，而这个通道是可用的。在图14.12中，站点A向站点B传送。站点C有一些数据要发送给站点D，这应该可以发送而不会干扰到从A到B的传输。但是，站点C暴露给了A的传送。它听到A在发送，这样被限制了发送。换句话说，C太保守，浪费了通道的能力。

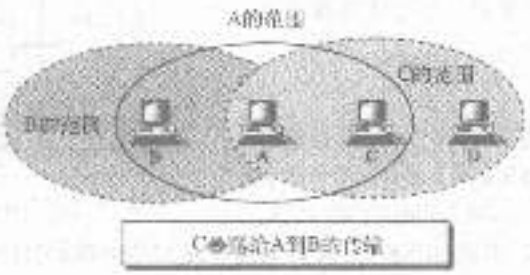


图14.12 暴露站点问题

这种情况握手消息RTS和CTS不能帮上忙。站点C听到来自A的RTS，但是没有听到来自B的CTS。站点C在听到来自A的RTS后，能等一段时间以获来自B的CTS到达A，然后它发送RTS给D，说明它需要和D通信。站点B和A可能都听到这个RTS，但是站点A在发送状态，而不是在接收状态。但是站点B以CTS响应。问题就在这儿，如果站点A开始发送它的数据，站点C由于冲突不能听到来自站点D的CTS，它就不能发送它的数据给D。它会保持暴露直到A结束它的数据发送，如图14.13所示。

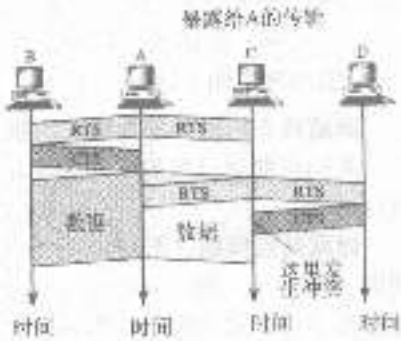


图14.13　暴露站点问题中握手的使用

14.1.4　物理层

我们讨论6个规范，如表14.4所示。

表14.4　物理层

IEEE	技　术	频　带	调　制	速率（Mbps）
802.11	FHSS	2.4 GHz	FSK	1和2
	DSSS	2.4 GHz	PSK	1和2
		红外线	PPM	1和2
802.11a	OFDM	5.725 GHz	PSK或QAM	6到54
802.11b	DSSS	2.4 GHz	PSK	5.5到11
802.11g	OFDM	2.4 GHz	不同	22到54

除了红外线以外，所有实现都工作在工业的、科学的、和医学的（ISM）频带，定义了一个范围（902-928MHz，2.400-4.835GHz，和5.725-5.850GHz）中的三个许可的频带，如图14.14所示。

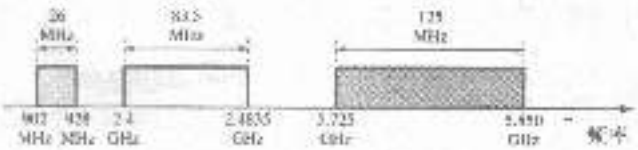


图14.14　工业的、科学的和医学的（ISM）频带

IEEE 802.11 FHSS

IEEE 802.11 FHSS使用了第6章中讨论的跳频扩频（FHSS）方法。FHSS使用2.4GHz ISM频带。这个频带被分成79个1MHz的子频带（以及一些防护频带）。由伪随机数字生成器选择调频序列。这个规范中的调制技术可以是二电平FSK或四电平FSK（1或2位/波特），这使得数据速率是1或2Mbps，如图14.15所示。

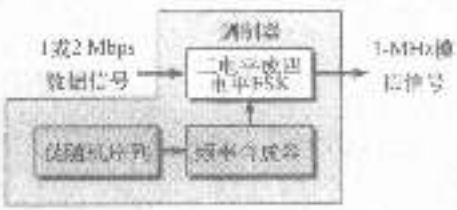


图14.15　IEEE 802.11 FHSS的物理层

IEEE 802.11 DSSS

IEEE 802.11 DSSS使用第6章中讨论的直接序列扩频（DSSS）方法。DSSS使用2.4GHz的ISM频带。这一规范的调制技术是在1M baud/s下的PSK。系统允许1位或2位/波特（BPSK或QPSK）。其数据速率是1Mbps或2Mbps，如图14.16所示。



图14.16 IEEE 802.11 DSSSS的物理层

IEEE 802.11 红外线

IEEE802.11红外线使用800nm到950nm范围中的红外线。调制技术称为脉冲位置调制（pulse position modulation，PPM）。为了1Mbps的数据速率，将4位序列先映射成16位序列，在这个16位序列中只有1位置成1其余位置成0。为了2Mbps的数据速率，将2位序列先映射成4位序列，在这个4位序列中只有1位置成1其余位置成0。然后将映射后的序列转换成光纤信号，光存在则为1，光不存在则为0。见图14.17。



图14.17 IEEE 802.11红外线的物理层

IEEE 802.11a OFDM

IEEE 802.11a OFDM描述了在一个5GHzISM波段上的正交频分多路复用（orthogonal frequency-division multiplexing，OFDM）的信号产生方法。OFDM和第6章中讨论的FDM类似，其主要区别是，在给定的时间内，所有的子波段都由一个源使用。各个源在数据链路层访问时相互竞争。频带被划分为52个子频带，其中有48个子频带每次发送48个位组，另外4个子频带用于控制信息。方案与ADSL类似，如第9章讨论的那样，将频带划分为子频带可以减少干扰的影响。如果随机地使用子频带，则还可以增加安全性。OFDM使用PSK和QAM进行调制。适用的数据速率是18Mbps（PSK）和54Mbps（QAM）。

IEEE 802.11b DSSS

IEEE 802.11b DSSS描述了在一个2.4GHz ISM波段上产生信号的高速率直接序列扩频（high-rate direct sequence spread spectrum，high-rate DSSS，HR-DSSS）方法。HR-DSSS和DSSS类似，区别在于它们的编码方法。HR-DSSS的编码方法叫做补码键控（complementary code keying，CCK）。CCK编码将4位或8位编码成一个CCK符号。为了与DSSS向后兼容，HR-DSSS定义了四种数据速率：1，2，5.5和11（Mbps）。前两种使用与DSSS相同的调制技术。5.5Mbps版本使用BPSK，其传输速率在1.375M baud/s，带有4位CCK编码。而11Mbps版本使用QPSK，其传输速率在1.375M baud/s，带有8位CCK编码。图14.18显示了这种标准的调制技术。

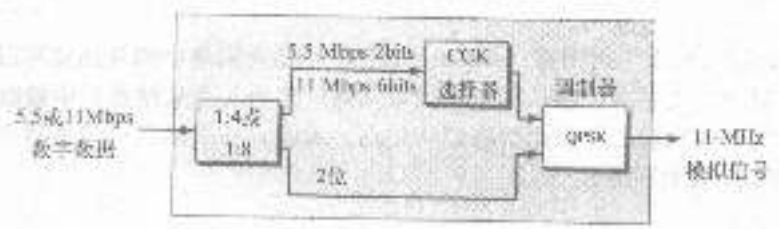


图14.18 IEEE 802.11b 的物理层

IEEE 802.11g

这个较新的规范定义了前向纠错和使用带有2.4GHz ISM波段的OFDM，调制技术可达到22或54Mbps的数据速率。它向后兼容802.11b，但调制技术是OFDM。

14.2 蓝牙

蓝牙（Bluetooth）是一种无线局域网技术，被设计用来连接具有不同功能的设备，如电话、笔记本、计算机（台式机和膝上型电脑）、照相机、打印机、咖啡壶等。一个蓝牙局域网是一个特别的网络，意思是该网络是自然形成的。所用的设备有时叫做小设备（gadgets），它们之间相互发现并构成网络，该网络叫微微网络（piconet）。如果每一个小设备具有这一功能的话，一个蓝牙局域网甚至可以和因特网相连。自然，一个蓝牙局域网的规模不能很大，因为如果要连接很多小设备的话，可能会造成混乱。

蓝牙技术在以下各个方面得到应用。计算机的外部设备可以通过这一技术与计算机通信（无线鼠标或键盘）；在规模不大的卫生保健中心，监控设备也能与传感器设备通信；家庭安全设备可以使用这一技术将不同的传感器连接到主安全控制器；参加会议的人员可以在会议上同步安排他们的膝上型电脑。

蓝牙开始于爱立信（Ericsson）公司的一个项目，最初的名字叫做Harald Blaatand。Harald Blaatand是丹麦的一个国王（940－981年），他统一了丹麦和挪威。Blaatand翻译成英语的蓝牙（Bluetooth）。

目前，蓝牙技术由IEEE 802.15定义的标准实现。标准定义了一个无线个人域网络（personal-area network，PAN），可以在一个房间或大厅范围内运行。

14.2.1 体系结构

蓝牙定义了两种类型的网络：微微网络（piconet）和散射网络（scatternet）。

微微网络

蓝牙网称做微微网络（piconet）或小网络。一个微微网络可以包含有8个站点，其中一个叫做主站（primary），其余的叫从站（secondary）。所有的从站点的时钟与跳频序列都要与主站点同步。注意，一个微微网络仅能有一个主站点。在主站点和从站点之间的通信可以是一对一的，也可以是一对多的。图14.19显示一个微微网络。

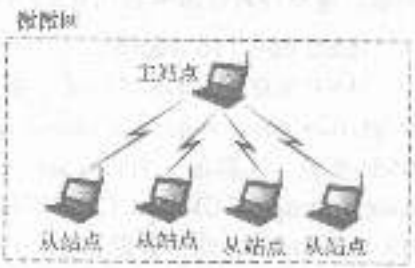


图14.19 微微网

虽然一个微微网络最多可以有7个从站点，但是可以附加处于休眠状态的8个从站点。一个处于休眠状态的从站点和主站点同步，但是在它脱离休眠状态之前不能参加通信。这里因为在一个微微网络中仅能有8个站点处于活动状态，从休眠状态激活一个站点就意味着必须有一个站点进入休眠状态。

散射网络

微微网络可以组合形成散射网络（scatternet）。一个微微网络中的从站点可以变成另一个微微网络中的主站点。该站点能够从第一个微微网络（作为一个从站点）中接收报文，并且作为主站点将报文发送到第二个微微网络的从站点。所以，一个站点可能是两个微微网络中的成员。图14.20表示散射网络。

蓝牙设备

蓝牙设备有一个内置的短程无线电发射机。目前的数据速率是在2.4GHz带宽下的1Mbps。这就意味着在IEEE 802.11b无线局域网和蓝牙局域网间有可能产生干扰。

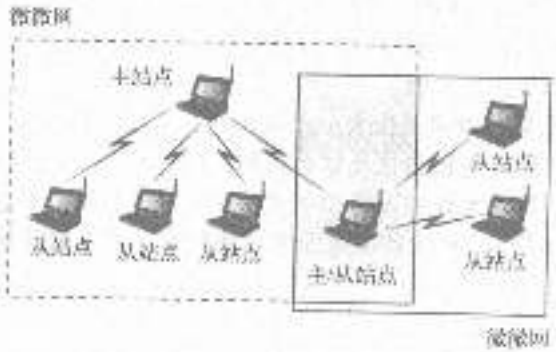


图14.20 散射网络

14.2.2 蓝牙层

蓝牙使用与本书所定义的因特网模型不完全匹配的几层。图14.21显示了这几层。

14.2.3 无线电层

无线电层大体上与因特网模型中的物理层相当。蓝牙设备是低功率的，其范围是10m。



图14.21 蓝牙层

频带

蓝牙使用2.4GHz ISM频带，并将其分为79个1MHz的通道。

FHSS

蓝牙在物理层使用跳频扩频（frequency-hopping spread spectrum，FHSS）方法以避免与其他设备或网络发生干扰。蓝牙每秒跳频1 600次，即每个设备每秒改变它的调制频率达1 600次。在一个设备跳频到另一个频率之前，它使用一个频率的时间仅为625μs（1/1600s），驻留时间是625μs。

调制

为了将位转变为信号，蓝牙使用一个复杂的FSK版本，叫做GFSK（带有Gaussion 带宽的滤波，该问题的讨论已经超出了本书的范围）。FSK有一个载波频率，位1由一个在载波频率之上的偏移来表示，位0由一个在载波频率之下的偏移来表示。在兆赫兹的载波频率，其每一通道都根据以下的公式定义：

$$f_c = 2\,402 + n \quad n = 0,\ 1,\ 2,\ 3,\ \cdots,\ 78$$

例如，如果第一个通道使用的载波频率为2 402MHz（即2.402GHz），那么第二个通道使用的载波频率就是2 403MHz（即2.403GHz）。

14.2.4 基带层

基带层大体上相当于局域网中的MAC子层。其访问方法是TDMA（见第12章）。主设备和从设备之间的通信使用时隙，时隙的长度恰好与驻留时间相等，为625μs。即在这一时间间隔，一个发送方发送一帧到从设备或者从设备发送一帧到主设备都使用一个频率。注意，通信仅仅在主设备和从设备之间进行，从设备之间不能直接相互通信。

TDMA

蓝牙使用的TDMA（见第13章）的格式称做时分双工TDMA（time-division duplexing

TDMA，TDD-TDMA)，TDD-TDMA是半双工通信的一种类型，在这种类型中，从设备和接收方发送和接收数据，但不在同一时间（半双工），并且每一方向的通信使用不同的跳频。这与使用不同的载波频率的步话机类似。

单一从设备通信　如果微微网络仅有一个从设备，那么TDMA运行是非常简单的，时间被划分为多个625ms的时隙。主设备使用偶数时隙（0，2，4，…），而从设备使用奇数时隙（1，3，5，…），TDD-TDMA允许主设备和从设备在半双工的模式下通信。在时隙0，主设备发送而从设备接收。在时隙1，从设备发送而主设备接收，并且按此规律循环。图14.22说明了这一思想。

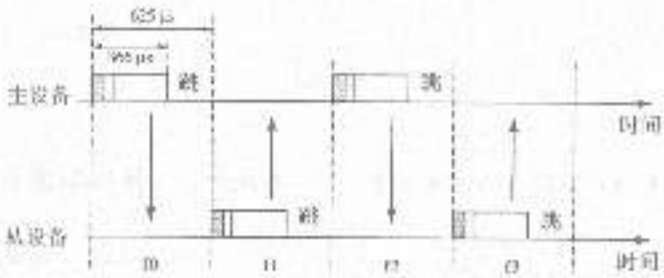


图14.22　单个从设备通信

多个从设备通信　如果在一个微微网络中有多个从设备，那么通信过程要稍微复杂一些，和前面一样，主设备使用偶数时隙发送，但只有在前面时隙的分组寻址到的那个从设备才在下一个奇数时隙内发送。网络内所有的从设备都在偶数时隙期间侦听，但只有一个从设备在下一奇数时隙内发送。图14.23显示了这一情况。

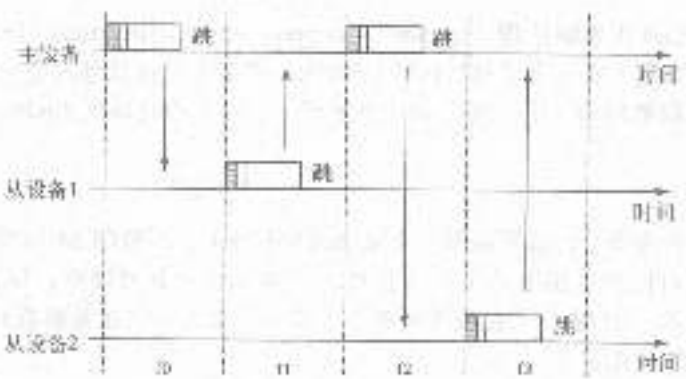


图14.23　多个从设备通信

下面对图14.23做详细说明：

1. 在时隙0，主设备向从设备1发送一帧。

2. 在时隙1，只有从设备1向主设备发送，因为前一个帧是向从设备1寻址的，所以其他从设备处于静止状态。

3. 在时隙2，主设备向从设备2发送一帧。

4. 在时隙3，只有从设备2向主设备发送，因为前一个帧是向从设备2寻址的，所以其他从设备处于静止状态。

5. 循环继续。

可以说，这一访问方法类似于有保留的轮询选择操作。当主设备选择一个从设备时，也

就轮询到该从设备，保留下一个时隙为被轮询到的站点发送它的帧。如果被轮询到的从设备没有要发送的帧，那么通道就保持静止状态。

物理链路

在主设备和从设备之间可以建立两种类型的链路：SCO链路和ACL链路。

- SCO　当避免延迟（在数据传递中的时间延迟）比保证完整性（无错传递）更重要的情况下，使用同步面向连接（synchronous connection-oriented，SCO）的链路。在SCO中，主设备和从设备之间通过在正常的时间间隔内保留特定的时隙来建立链路。连接的基本单元是两个时隙，每个方向一个。在SCO中，如果一个分组遭到破坏，它就不会进行重传。SCO用于实时的视频传输，因为在这种场合避免延迟是非常重要的。一个从设备可以与主设备之间建立三个SCO链路，每一个链路发送64kbps的数字化的视频信号（PCM）。
- ACL　当保证数据的完整性比避免延迟更重要的情况下，使用异步无连接链路（asynchronous connectionless link，ACL）。在这种链路中，如果封装在帧中的有效载荷遭到破坏，就要重传。当且仅当前面的时隙已经寻址到它，从设备才会在可用的奇数时隙内返回一个ACL帧。ACL可以使用一个、三个或更多个时隙并能达到721Kbps的数据速率。

帧格式

在基带层的帧可以是以下三种类型之一：一个时隙、三个时隙或五个时隙。如我们以前所说的那样，一个时隙是625μs。但是在一个时隙的帧交换中，有259μs用于跳频和控制机制。这就意味着一个时隙的帧仅能持续（625-259）μs，即366μs。对于1MHz带宽和1位/Hz来说，一个时隙的帧是366位。

一个三时隙的帧占据三个时隙的时间。但是由于有259μs用于跳频，所以帧的长度是（3×625-259）μs，即1 616μs或1 616位。一个使用三个时隙帧的设备在同一跳频中（在相同的载波频率下）保持三个时隙。即使仅使用一个跳频，也要消耗三个跳频数。也就是说，每一帧的跳频数等于帧的第一个时隙。

一个五个时隙的帧也使用259位实现跳频，也就是说帧的长度是5×259μs，即2 866μs或2 866位。

图14.24显示了三种帧类型的格式。

下面对每个字段进行说明：

- 访问代码。72位的字段通常包括同步位和主设备的标识符，用于将一个微微网络的帧与其他微微网络的帧区别开来。

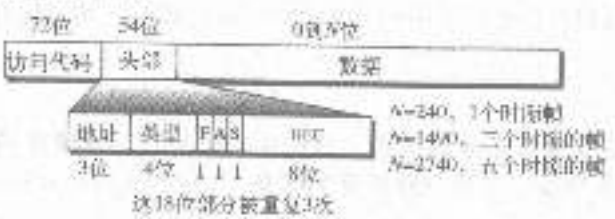


图14.24　帧格式类型

- 头部，54位的字段3次重复18位的模式。每个模式有下面三个子字段：

1. 地址，3位的地址子字段最多可以定义7个从设备（1－7）。如果该值为0，则用于从主设备到所有从设备的广播通信。

2. 类型。4位的类型子字段定义了来自上层的数据类型。我们会在后面讨论这些类型。

3. F。1位的子字段用来进行流量控制。当设置为1时，表示设备不能接收更多的帧（缓存区满）。

4. A。1位的子字段用来进行确认。蓝牙使用停-等ARQ，用1位就能满足确认的要求。

5. S。1位的子字段用来保持序列号。蓝牙使用停-等ARQ，用1位就能满足序列编码的要求。

6. HEC。8位的头部纠错子字段是一个校验和，用于检查每一个18位头部分段的错误。

头部有三个18位的分段，发送方在创建时这三个分段是一样的，接收方逐位地比较这三个分段。如果三个分段的每一个对应位相等，该位就被接收；否则就接收两个相同的位。这是前向差错校正（只对于头部）方式。由于使用空气作为通信的介质，噪音大，所以需要双重的差错控制。注意，在这一子层中没有重传机制。

- **有效载荷**。该子层的长度是0～2 740位。它包含数据或从上层发来的控制信息。

14.2.5 L2CAP

逻辑链路控制和自适应协议（Logical Link Control and Adaptation Protocol，L2CAP，这里的L2是L1的意思）基本上与局域网中的LLC子层类似。用于在ACL链路中的数据交换。SCO通道不使用L2CAP。图14.25显示了这一级的数据分组格式。

16位的长度字段按字节定义从上层传来的数据的大小，最大可达65 535字节。通道ID（CID）定义了在本级创建的虚拟通道的单一标识符（后面有叙述）。

L2CAP有几个特殊的任务：多路复用、分段和重装、服务质量以及组管理。



图14.25　L2CAP数据分组格式

多路复用

L2CAP可以实现多路复用，在发送站点，它接收一个上层协议的数据，组织成帧，将帧传递到基带层以备发送。在接收方，从基带层接收帧，提取数据并将它们递交给相应的协议层。它创建一种类型的虚通道，这将在以后有关高层协议的章节中讨论。

分段和重组

基带层的有效负载字段的最大长度是2 774位或343字节，其中包含4字节用来定义分组和分组的长度。因此，能够从上层到达的分组的长度仅为339字节。但是，应用层有时需要发送的数据分组可达65 535字节（例如一个因特网分组）。L2CAP就将这些大的分组分割成段并附加一些信息以定义该段在原始分组中的位置。L2CAP在源站点将分组分成段并在目的站点再将这些段重组。

QoS

蓝牙允许站点定义服务质量的等级。我们将在第23章讨论服务质量。暂时可以这样理解，如果不定义服务质量，蓝牙就默认尽力服务，即在所处的环境下，它将尽力提供好的服务质量。

组管理

L2CAP的另一功能就是允许在设备之间创建一种逻辑寻址类型。这与多播类似。例如，2个或3个从设备可以是一个从上设备接收数据的多播组中的一部分。

14.2.6 其他上层

蓝牙为使用L2CAP服务的上层的定义了多个协议，这些协议因不同目标而专用。

推荐读物

为了详细地讨论与本章有关的主题，我们推荐下列读物，在本书末列出了括号[……]中的参考资料。

读物

无线LAN在包括[Sch03]和[Gas02]的许多书中有讨论。无线LAN在[For03]的第15章，[Kei02]的第8章，[Sta02]的第13、14章，[Sta04]的第17章中有讨论。蓝牙在[Sta02]的第15章，[For03]的第15章中有讨论。

本章小结

- 无线LAN的IEEE 802.11标准定义了两类服务：基础服务集（BSS）和扩展服务集（ESS）。
- 用于分布式协调功能（DCF）MAC子层的访问方式是CSMA/CA。
- 用于点协调功能（PCF）MAC子层的访问方式是轮询。
- 网络分配矢量（NAV）是用于避免冲突的定时器。
- MAC层帧有9个字段。寻址机制最多能包括4个地址。
- 无线LAN使用管理帧、控制帧和数据帧。
- IEEE 802.11定义了多个物理层，使用不同的数据速率和调制技术。
- 蓝牙是连接小区域内的设备（称为小设备）的无线LAN技术。
- 蓝牙网络称为微微网。多个微微网形成称为散射网的网络。
- 蓝牙网络由一个主设备和最多7个从设备组成。

习题

复习题

1. BSS和ESS的区别是什么？
2. 讨论无线LAN中的三种移动性。
3. OFDM如何不同于FDM？
4. 无线LAN使用的访问方式是什么？
5. NAV的目的是什么？
6. 比较微微网和散射网。
7. 将蓝牙和因特网模型的各层相比较。
8. 蓝牙主设备和蓝牙从设备之间的两种链路是什么？
9. 在多个从设备通信中，哪个设备使用偶数时隙，哪个设备使用奇数时隙？
10. 蓝牙一个时隙帧中多少时间用于调频机制？三个时隙帧和五个时隙帧呢？

练习题

11. 比较CSMA/CD和CSMA/CA。
12. 使用表14.5比较IEEE 802.3和802.11中的字段

表14.5　练习12

字　段	IEEE 802.3字段长度	IEEE 802.11字段长度
三角地址		
源地址		
地址1		
地址2		
地址3		
地址4		
FC		
D/ID		
SC		
PDU长度		
数据和填充		
或主体		
FCS（CRC）		

第15章 连接局域网、主干网和虚拟局域网

局域网通常不是孤立运行，一般互相连接或者连接到因特网。为了连接局域网或局域网的各个分段，我们需要使用连接设备。连接设备能够工作在因特网模型的不同层。本章将只讨论工作在物理层和数据链路层的设备，在19章将讨论工作于前三层的设备。

在讨论完这些连接设备之后，我们将说明如何使用它们建立主干网，最后再讨论虚拟局域网。

15.1 连接设备

本节中，我们根据在网络中所工作的层把连接设备（connecting devices）分成五类，如图15.1所示。



图15.1 五类连接设备

我们把设备定义成五类：

1. 工作在物理层以下的设备，如无源集线器。
2. 工作在物理层的设备（中继器或有源集线器）。
3. 工作在物理层和数据链路层的设备（网桥或两层交换机）。
4. 工作在物理层、数据链路层和网络层的设备（路由器或三层交换机）。
5. 工作在所有五层的设备（网关）。

15.1.1 无源集线器

无源集线器只是个连接器，它连接来自不同分支的线路。在星型拓扑以太网中，无源集线器只是一个来自不同站点的信号冲突点，集线器是冲突点。这种集线器是介质的一部分，它在因特网模型中的位置在物理层以下。

15.1.2 中继器

中继器（repeater）是仅工作在物理层的设备。网络内携带信息的信号在衰减到危及数据完整性之前，可以传输一段固定的距离。中继器接收信号，并且在信号变得很弱或者被破坏之前，重新生成原始的位模式，然后中继器发送新生成的信号。如图15.2所示，中继器能够扩展局域网的物理长度。

中继器实际并不能连接两个局域网，它连接的是同一局域网的两个分段。所连接的两段仍然是一个局域网的一部分。中继器不能连接采用不同协议的两个局域网。

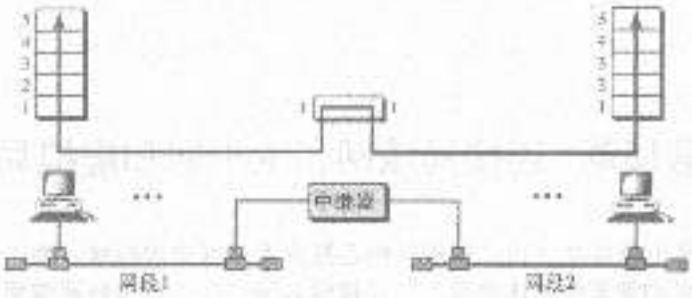


图15.2　连接两个局域网网段的中继器

中继器连接一个局域网的各个网段。

中继器可以克服10Base5以太网标准的长度限制。在这个标准下，电缆的长度限制在500m以内。为了扩展长度，我们将电缆分割成几段，在分段间安装中继器。注意，整个网络仍然是一个局域网，但是被中继器分割的部分称为网段（segment）。中继器担当两端口的节点，但是中继器仅工作在物理层。当从任何一个端口接收到一帧时，它就重新生成并转发给另一个端口。

中继器转发每一帧，它没有过滤能力。

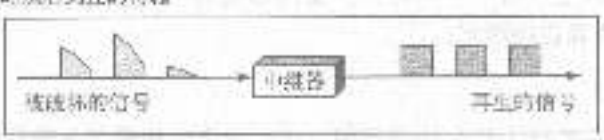
有人将中继器比做放大器，但这种比照是不准确的。放大器（amplifier）不能区分有效信号和噪声，它放大所有输入的信号。中继器不放大信号，它重新生成信号。当中继器接收到微弱的或被破坏的信号时，就逐位地以原始的信号强度生成一个拷贝。

中继器是再生器，不是放大器。

中继器在链路中安装的位置是很关键的，它必须放置在信号的每个位被噪声改变之前能够到达的位置。小的噪声可能改变位电压的准确值，但不破坏它的可识别性（见图15.3）。然而，如果被破坏的位传输的距离太远，积聚的噪声就可能完全改变它的含义。在这种情况下，原始电压就不能复原，必须纠错。在信号的可读性消失之前，放置在该处的中继器可以读出信号，足以确定原始电压，并按原始格式复制信号。



a) 从右到左的传输



b) 从左到右的传输

图15.3　中继器的功能

15.1.3　有源集线器

实际上，有源集线器（hub）是一个多端口的中继器。它通常用于星型拓扑结构中，在站点间建立连接。我们已经见过集线器在一些以太网配置中的应用实例（例如10Base-T）。实际上，集线器也可以像图15.4所示的那样用来创建多级连接。集线器的级连使用克服了10Base-

T标准的距离限制（100m）。

15.1.4 网桥

网桥（bridge）工作在物理层和数据链路层。用做物理层设备时，它重新生成接收到的信号。用做数据链路层设备时，它可以检查帧所包含的物理（MAC）地址（源地址和目标地址）。



图15.4 集线器级连

过滤

有人会问，网桥和中继器在功能上有什么差别？网桥具有过滤（filtering）能力。它检查帧的目的地址，并决定该帧是被转发或丢弃。转发帧时必须指定端口。每个网桥都有一个端口地址映射表。

> 网桥有一个用做过滤决策的表。

举个例子。如图16.5所示，两个局域网通过网桥相连。如果一个目的地址是712B13456142的帧到达端口1，网桥就查阅它的表，查找该帧离开的端口。根据映射表，目的地址是712B13456142的帧从端口1离开，因此不需要转发，该帧被丢弃。另一方面，如果一个目的地址是712B13456141的帧到达端口2，它离开的端口是端口1，该帧被转发。在第一种情况下，局域网2没有流量；第二种情况下，两个局域网都有通信量。我们的例子中给出了一个两端口的网桥。实际上，网桥通常有更多的端口。

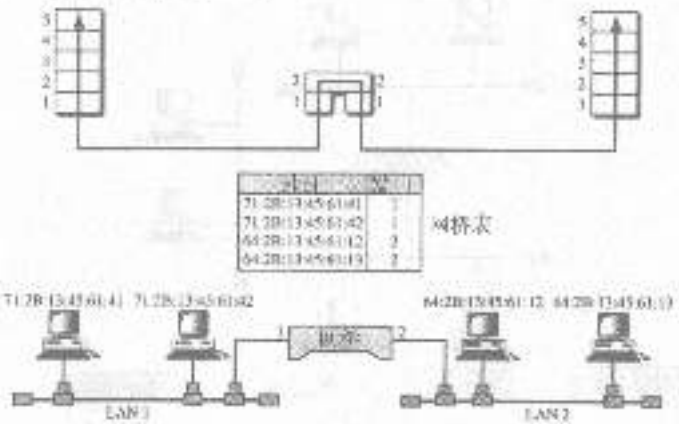


图15.5 连接两个局域网的网桥

还要注意：网桥不改变帧中所包含的物理地址。

> 网桥不改变帧中所包含的物理（MAC）地址。

透明网桥

透明网桥（transparent bridge）是一个它所连接的站点完全意识不到其存在的网桥。如果在系统中增加或移除一个透明网桥，则不需要重新配置。参考IEEE 802.1d规范，安装有透明网桥的系统必须符合以下三个标准：

1. 帧必须能从一个站点转发到另一个站点。
2. 通过学习网络中帧的传输，自动建立转发表。

3. 必须避免系统内循环。

转发　像前一节讨论的那样，透明网桥必须正确地转发帧。

学习　最早的网桥的转发表是静态的，配置网桥时，系统管理员需要手工输入每个表的条目。尽管过程简单，却并不实用。如果增加或移除了一个站点，转发表就必须手工修改。如果站点的MAC地址改变了（这并不少见），也是如此。例如，安装一个新网卡就意味着一个新的MAC地址。

比静态表更好的解决方法是使用自动映射地址到端口的动态表。为了动态生成转发表，需要网桥从帧传输中逐渐学习。要做到这一点，网桥对源地址和目的地址都要检查。目的地址用来做转发决定（表查找）；源地址用做添加表条目和更新的目的。我们用图16.6来详细阐述这个过程。

1. 当站点A向站点D发送帧时，网桥还没有关于D或者A的表条目。帧将从所有二个端口转发出去，从而在网络中泛滥。然而，通过查看源地址，网桥知道了站点A位于连接在端口1的LAN中。这意味着将来目的地是A的帧必须通过端口1转发出去。网桥就添加这个条目到表中。至此，转发表有了第一个条目。

2. 当站点E向站点A发送帧时，网桥有一个A的条目，所以就仅向端口1转发该帧。这样就不会出现帧泛滥。另外，还把该帧的源地址E添加到表的第二个条目。

3. 当站点B向站点C发送帧时，网桥没有关于C的表条目，所以又一次出现帧泛滥，再向表中添加新的条目。

4. 当网桥转发帧时，将继续这个学习过程。

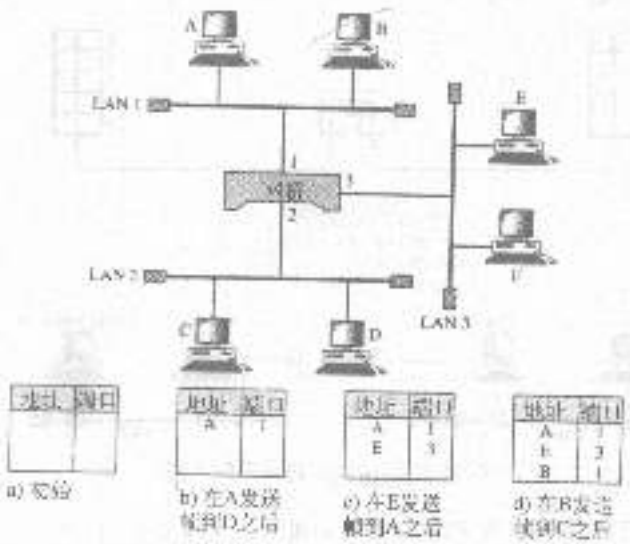


图15.6　学习型网桥和学习过程

循环问题　只要系统中没有冗余的网桥，透明网桥将工作得很好。然而，系统管理员喜欢使用冗余网桥（在两个LAN间使用多个网桥）来使系统更可靠。如果一个网桥失效，另一个网桥将接管直到失效的网桥被修好或被替换。不幸的是，冗余会在系统中产生循环。图16.7显示了一个很简单的例子，由于在系统中使用两个网桥连接两个LAN而产生了循环。

1. 站点A向站点D发送一帧。两个网桥的转发表都是空的，所以都转发帧，并且根据源地址A更新各自的表。

2. 现在LAN 2中有该帧的两份拷贝，网桥1发出的拷贝由网桥2接收，但是网桥2没有任何

关于目的地址D的信息，此时出现喷泛滥。网桥2发出的拷贝被网桥1接收到，由于没有关于D的信息，它也向所有端口发出，注意，网桥作为网络共享介质的两个节点，每个帧都是被独立处理的，因为网桥使用了像CSMA/CD这样的访问方法。网桥的转发表被更新，但仍然没有目的地址D的信息。

3. 现在LAN 1中有该帧的两份拷贝。又重复上一步骤的过程，每个帧都在网络中泛滥。

4. 这个过程一直持续，注意，网桥也是中继器，并且重新生成帧。所以在每次重复中，都有该帧重新生成的新拷贝。

为了解决循环问题，IEEE规范需要网桥使用生成树算法来建立无循环拓扑结构。

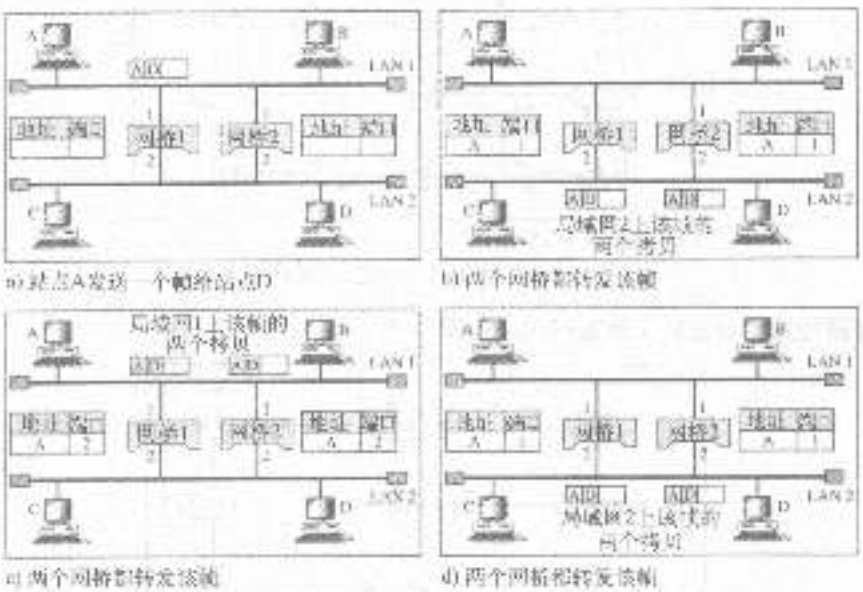


图15.7 循环问题

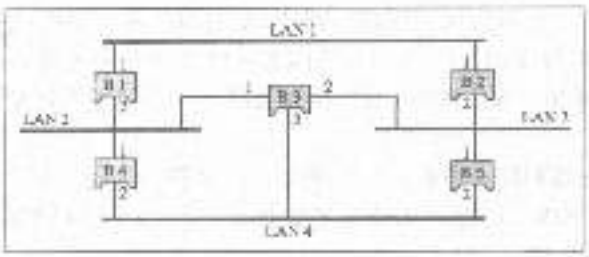
生成树

在图论中，生成树（spanning tree）是一个没有循环路径的图。在用网桥连接的LAN中，这就意味着要建立每个局域网都能通过唯一路径（没有循环）到达其他任何LAN的拓扑结构。因为电缆和网桥之间的物理连接，我们不能改变系统的物理拓扑结构，但是我们可以创建一个覆盖物理拓扑的逻辑拓扑结构。图15.8说明了一个有4个LAN和5个网桥的系统。我们已经说明了物理系统和它在图论中的表示。虽然一些教科书把LAN表示成节点而网桥表示成连接线，但是我们在这里把LAN和网桥都显示成节点，连接线表示LAN和网桥的连接。为了找出生成树，我们需要给连接线分配一个成本（度量）。成本的解释规则留给系统管理员。它可以是最小跳数（节点）的路径，最小延迟的路径，或者最大带宽的路径。如果两个端口有相同的最小值，系统管理员只需选择一个。我们在这里选择最小跳数作为成本。但是我们会在22章看到通常从网桥到LAN的跳数为1，而从LAN到网桥的跳数为0。

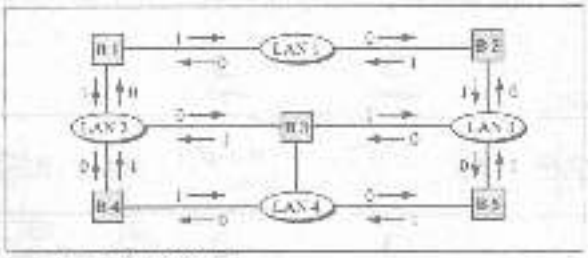
这个过程包括以下三个步骤：

1. 每个网桥有一个内置的ID（通常是唯一的序列号）。每个网桥广播它的ID。这样所有网桥都知道哪个是最小的ID。选择ID最小的网桥作为根网桥（作为树的根）。我们假定B1有最小ID。所以它被选为根网桥。

2. 这个算法尝试找出从根网桥到其他每个网桥或LAN的最短路径（最小成本的路径）。可以通过检查从根网桥到目的地的整个成本来找出最短路径。图15.9给出了最短路径。



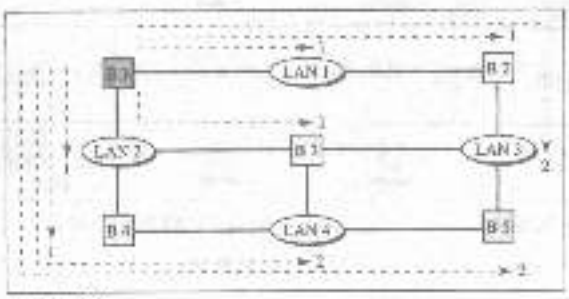
a) 实际系统



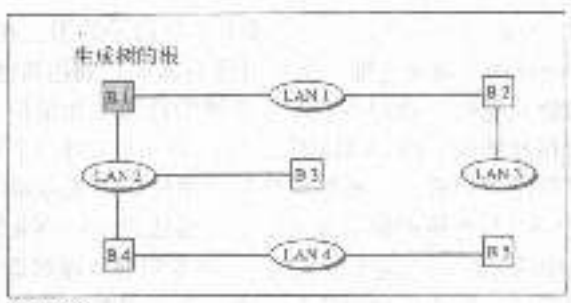
b) 每条线有成本值的图示

图15.8　连接的LAN系统和它的图示

3. 最短路径的组合生成了最短的树，如图15.9所示。



a) 最短路径



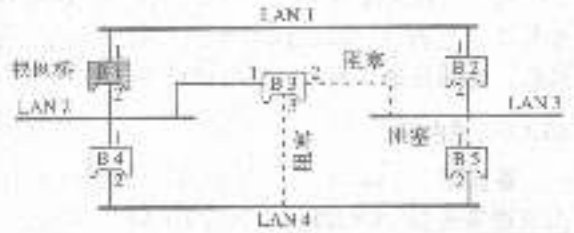
b) 生成树

图15.9　找出网桥系统的最短路径和生成树

4. 基于生成树，我们标记属于生成树部分的端口为转发端口（forwarding port），它转发网桥接收到的帧，还标记不属于生成树部分的端口为阻塞端口（blocking port）它阻塞网桥接收到的帧。图15.10显示了有转发端口（实线）和阻塞端口（虚线）的LAN物理系统。

注意生成树中从任一LAN到其他LAN只有一条路径。这意味着从一个LAN到另一个LAN只有一条路径。你可以自己证明从LAN 1 到LAN 2、LAN 3、或LAN 4只有一条路径。对LAN3和LAN 4同样正确。

动态算法 我们已经通过手工描述了生成树算法，但这不是实际情况。每个网桥都配置有一个软件包来动态地完成这个过程。网桥互相发送称为网桥协议数据单元（BPDU）的特定消息，用来更新生成树。当系统有变化时，如某个网桥出现故障或者增加或移除网桥，生成树都需要更新。



网桥B3的端口2和端口3是阻塞端口（没有帧从这些端口发出）。
网桥B5的端口1也是阻塞端口（没有帧从这些端口发出）。

图15.10 使用生成树算法后的转发端口和阻塞端口

源路由网桥

另一个在有冗余网桥的系统中防止循环的方法是使用**源路由网桥**（source routing bridge）。透明网桥的职责包括过滤帧、转发和阻塞。在有源路由网桥的系统中，这些职责由源站点来执行，某种程度由目的站点来执行。

在源路由算法中，发送站点指定帧必须经过的网桥，这些网桥的地址包含在帧中。换句话说，帧中不仅包含了源地址和目的地址，还包含了所有要经过的网桥地址。

在发送数据帧之前，源站点通过和目的站点交换特定的帧来得到这些网桥地址。

源路由网桥是IEEE设计用于令牌环局域网的，现在这些局域网并不普及。

用网桥连接不同的局域网

理论上，网桥在数据链路层应该能够连接使用不同协议的局域网，如以太局域网连接到无线局域网。然而，有很多问题需要考虑：

- **帧格式**。每种局域网类型都有自己的帧格式（比较以太网帧和无线局域网帧）。
- **最大数据长度**。如果接收的帧长度对目的局域网来说太大了，那么数据必须分割成几个帧，然后在目的站点重新组装。然而，没有数据链路层的协议支持帧的分割和重新组装。我们将在第19章看到，这在网络层是允许的。因此，网桥必须丢弃任何对系统来说太大的帧。
- **数据速率**。每种局域网类型都有自己的数据速率（例如以太网的数据速率是10Mbps，而无线局域网的是1 Mbps）。网桥必须缓存帧以补偿这个差别。
- **位顺序**。每种局域网类型都有自己的位发送策略，有些先发送字节中高位，而有些先发送低位。
- **安全**。有些局域网，如无线局域网，在数据链路层执行安全措施。而有些则没有，如以太局域网。安全通常包括加密（见第30章）。当接收到来自无线局域网的帧时，在转发给以太局域网之前，网桥需要解密这些信息。
- **多媒体支持**。有些局域网支持多媒体和这种通信所需要的服务质量，而有些则不支持。

15.1.5 两层交换机

当使用“交换机”这个术语时，必须要谨慎，因为交换机可能代表两种不同的东西。必须要加上设备所工作的层次来加以说明。可以有两层交换机或者三层交换机。**三层交换机**（three-layer switch）工作在网络层，它是路由器的一种。**两层交换机**（two-layer switch）工作在物理层和数据链路层。

两层交换机是一个有许多端口并且有更好（更快）性能的网桥。有少量端口的网桥只可以连接几个局域网。有更多端口的网桥可以给每个站点分配唯一的端口。每个站点都作为独

点的实体。这意味着没有通信量竞争（没有在以太网中见到的冲突）。

两层交换机像网桥一样，基于接收到帧的MAC地址做出过滤决策。但是，两层交换机更加复杂。它有缓存区来保存帧进行处理，有交换组件使得更快地转发帧。一些新型的两层交换机，称为快速交换机，已经设计成一检查到帧头部的MAC地址就转发帧。

15.1.6 路由器

路由器（router）是三层设备，它基于分组的逻辑地址（主机到主机寻址）路由分组。路由器通常连接LAN和路由器中的WAN。它有一张表用来决策路由。路由表通常是动态的，使用路由协议更新。我们将在第19章和第21章中更详细地讨论路由器和路由选择。图15.11给出了使用路由器来连接LAN和WAN的因特网部分。

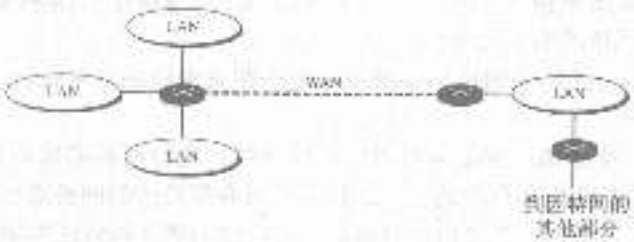


图15.11　连接独立LAN和WAN的路由器

15.1.7 三层交换机

三层交换机是路由器，但更快更复杂。三层交换机中的交换光纤允许更快的表查询和转发。本书中，我们是互换地使用路由器和三层交换机。

15.1.8 网关

虽然一些教科书互换地使用术语网关和路由器，但是大多数书还是对两者有区别的。网关通常是工作在因特网全部五层或者OSI模型中全部七层的计算机。网关拿到一条应用消息、读取消息并解释消息。这意味着它能作为两个使用不同模型的互联网的连接设备。例如，设计成使用OSI模型的网络可以连接到使用因特网模型的另一个网络。连接两个系统的网关接收到来自第一个系统的帧时，把它交付给OSI应用层，然后删除这条消息。

网关可以提供安全性。在第32章中，我们将学到网关用来过滤不需要的应用层消息。

15.2 主干网

本章所讨论的一些连接设备可以用来在主干网中连接局域网。主干网允许连接多个局域网。在主干网中，站点不直接连接到主干上；站点是局域网的一部分，由主干来连接这些局域网。主干网本身也是使用局域网协议的局域网，比如以太网，每个到主干网的连接本身是另一个局域网。

尽管有很多不同的结构可以使用在主干网中，但我们只讨论两种最常见的结构：总线型和星型结构。

15.2.1 总线型主干网

在总线型主干网（bus backbone）中，主干网的拓扑结构是总线型。主干本身可以使用支持总线型拓扑结构的协议的其中一种，如10Base5或者10Base2。

在总线型主干网中，主干网的拓扑结构是总线型。

总线型主干网通常作为分布式主干网来连接一个组织的不同建筑物。每个建筑物可以包含单独的局域网，或者是另一个主干网（通常是星型主干网）。使用总线型主干网的一个典型例子是在校园内连接单层或多层建筑物。每个单层建筑物通常有一个单独的局域网，每个多层建筑物有一个主干网（通常是星型主干网）连接每层的局域网。总线型主干网可以互连这些局域网和主干网。图15.12展示了基于网桥的连接有四个局域网的总线型主干网。

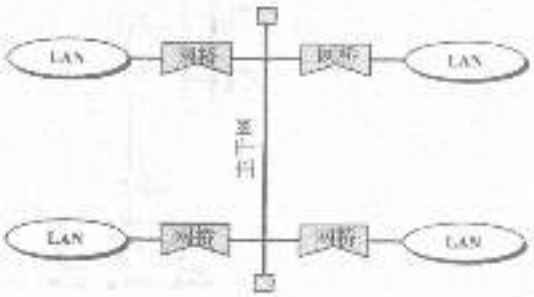


图15.12 总线型主干网

在图15.12中，如果一个站点需要向同一LAN中的另一个站点发送帧，相应的网桥将阻塞该帧，这个帧永远也到达不了主干。但是，如果一个站点需要向另一个LAN中的站点发送帧，网桥将传输该帧到主干，由适当的网桥接收并发送到目的LAN。每个和主干相连的网桥都有一个该网桥的LAN一侧的所有站点列表。基于这个表的内容来决定是阻塞还是转发帧。

15.2.2 星型主干网

星型主干网（star backbone），有时称为折叠式或交换式主干网，其拓扑结构是星型。在这个配置中，主干网仅是一台连接局域网的交换机（这就是为什么称为折叠式主干网的原因）。

> 在星型主干网中，拓扑结构是星型，主干网仅是一台交换机。

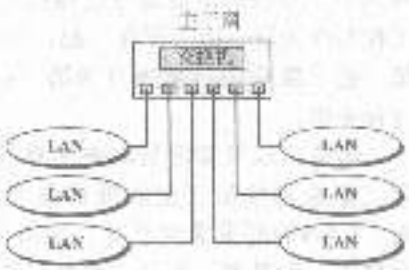


图15.13 星型主干网

图15.13展示了一个星型主干网。注意，在这个配置中，交换机担当主干的工作，并且同时连接LAN。

星型主干网最常用做一个建筑物内的分布式主干。在一个多层的建筑物内，通常有服务特定楼层的LAN。星型主干网连接这些LAN。只有一个交换机的主干网可以安装在地下室或第一层，用单独的电缆从交换机连接到每个LAN。如果单个的LAN也有物理星型拓扑结构，则所使用的集线器（或交换机）可以安装在相应楼层的机柜中，或者所有的都可以安装在最靠近交换机的地方。我们在安装有主干网交换机和集线器或交换机的地下室中经常可以看到机架或机柜。

15.2.3 连接远程LAN

主干网的另一个常见的应用是连接远程LAN。当公司有几个拥有局域网的办公室并且需要连接起来时，这种类型的主干网就很有用处，可以通过网桥，有时称为远程网桥（remote bridges）来连接。这些网桥充当连接局域网和点对点网络（如租用的电话线路或ADSL线路）的连接设备。这个情况中的点对点网络称为没有站点的LAN。点对点链路可以使用类似PPP这样的协议。图15.14展示了一个连接远程LAN的主干网。

> 在用远程网桥连接的远程主干网中，点对点链路被认为是一个局域网。

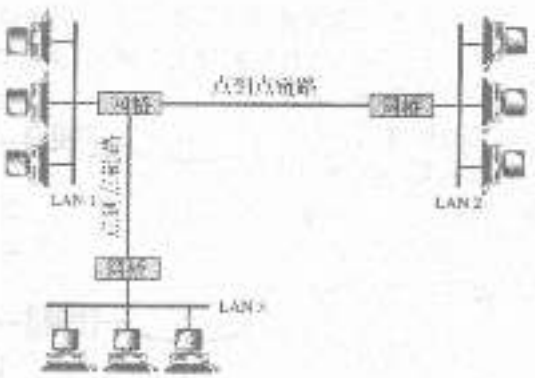


图15.14　连接远程LAN

15.3　虚拟局域网

如果站点物理上属于某个LAN，那它就被当做该LAN的一部分，成员关系的标准是地理位置。如果需要在属于不同物理LAN的两个站点之间建立虚拟连接，那该怎么办呢？我们可以像局域网一样将路由器定义虚拟局域网（virtual local area network，VLAN），不过是通过软件，而不是通过物理线路来配置。

我们通过一个例子来详细说明这个定义。图15.15展示了一个工程公司中的交换式局域网，有10个站点被分组成通过交换机连接的3个LAN。前4个工程师作为第一组工作在一起，接下来的3个工程师作为第二组，最后3个工程师作为第三组。LAN被设置成支持这种安排。

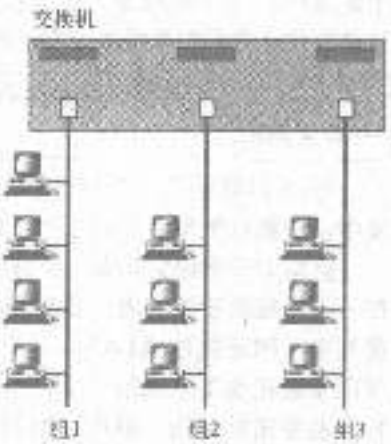


图15.15　连接三个LAN的交换机

但是，如果管理员需要将两个工程师从第一组调到第三组来加快第三组的项目进度，将会带来什么影响呢？LAN的配置需要改变，网络技术员必须重新布线。如果下一个星期，两个工程师又要回到原来的组，相同的问题会再次出现。在交换式局域网中，工作组的改变意味着网络配置的物理改变。

图15.16展示了被划分成三个VLAN的一个交换式局域网。VLAN技术的全部思想是将LAN划分成逻辑的，而不是物理的网段。一个LAN可以划分成称为VLAN的多个逻辑LAN。每个VLAN是组织的一个工作组。如果某人从一个组移动到另一个组，就没有必要改变物理配置。在VLAN中，组的成员是由软件而不是硬件来定义的。任何站点都可以逻辑地移动到另一个VLAN中，VLAN的所有成员都可以接收发送到特定VLAN的广播信息。这意味着，如果某个站点从VLAN1移动到VLAN2，它将接收发送到VLAN2的广播信息，而不再接收发送到VLAN1的广播信息。

很明显，先前例子中的问题可以通过使用VLAN很容易地得到解决。通过软件把工程师从一个工作组移动到另一个工作组，比改变物理网络的配置要容易得多。

VLAN技术甚至允许连接在不同交换机上的站点组成一个VLAN。图15.17展示了一个有两台交换机和三个VLAN的主干LAN，交换机A和B的站点属于不同VLAN。

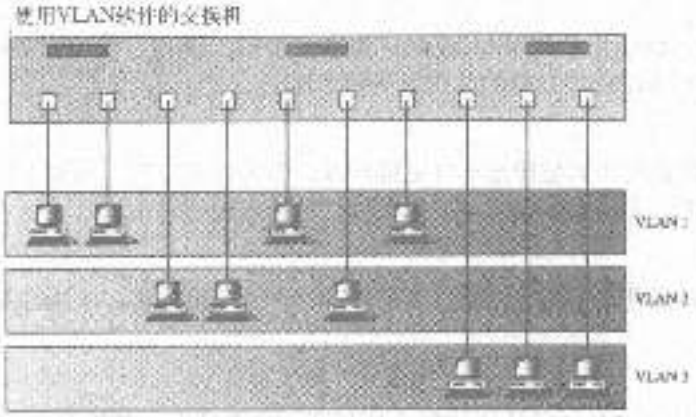


图15.16 使用VLAN软件的交换机

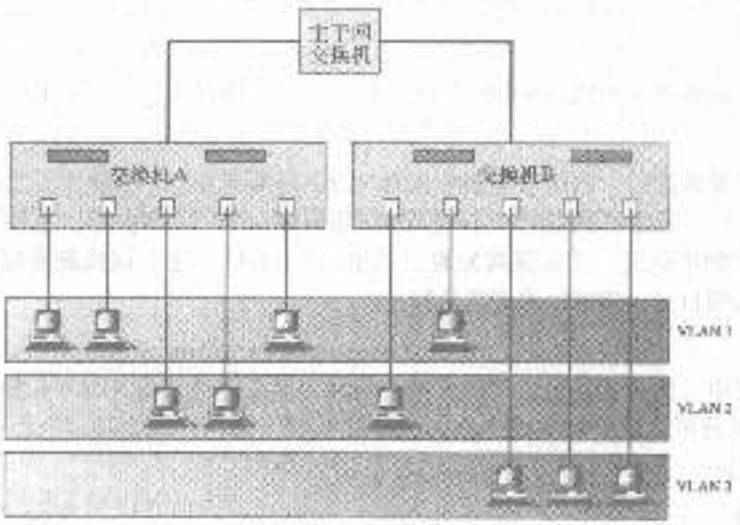


图15.17 主干网中使用VLAN软件的两个交换机

对有两个分离的建筑物的公司来说，这是一个好的配置，每个建筑物都可以拥有自己的交换式LAN，这些LAN由主干网连接。第一个和第二个建筑物内的人可以处于同一个工作组，即使他们由不同的物理LAN来连接。

从这三个例子中，我们可以定义VLAN的一个特征：

VLAN创建广播域。

VLAN将属于一个或多个物理LAN的站点分组到广播域中。VLAN中的站点和其他站点通信时就像它们属于一个物理网段一样。

15.3.1 成员

采用什么特征来分组VLAN中的站点呢？厂商使用不同的特征，例如端口号、MAC地址、IP地址、IP多播地址或者以上两种或多种联合使用。

端口号

有些VLAN厂商使用交换机的端口号作为分组依据。例如，管理员可以定义连接在端口1，2，3和7的站点属于VLAN1，连接在端口4，10和12的站点属于VLAN2，依此类推。

MAC地址

有些VLAN厂商使用48位MAC地址作为分组依据。例如，管理员可以定义MAC地址为E2342A12334和F2A123BCD341的站点属于VLAN1。

IP地址

有些VLAN厂商使用32位IP地址（见第19章）作为分组依据。例如，管理员可以定义IP地址为181.34.23.67，181.34.23.98和181.34.23.112的站点属于VLAN1。

多播IP地址

有些VLAN厂商使用多播IP地址（见第19章）作为分组依据。IP层的多播现在转换为数据链路层的多播。

联合使用

最近，有些厂商开发的软件支持联合使用所有这些特征。管理员在安装软件时可以选择一个或多个特征。另外，软件可以重新配置来改变这些设置。

15.3.2　配置

站点怎样分组到不同的VLAN中呢？可以通过以下三种方式之一：手工、半自动和自动。

手工配置

采用手工配置方式时，网络管理员在安装VLAN时要使用VLAN软件手工地将站点分配到不同的VLAN中。以后要将站点从一个VLAN迁移到另一个VLAN中时，也要手工地操作。注意，这不是一个物理配置，而是逻辑配置。这里，"手工地"这个词代表管理员使用VLAN软件时，需要键入端口号、IP地址或者其他特征。

自动配置

在自动配置中，使用管理员定义的标准，自动地将站点连接到VLAN或断开与VLAN的连接。例如，管理员可以定义项目编号作为分组的标准。当用户改变项目时，他将自动地迁移到新的VLAN中。

半自动配置

半自动配置在某种程度上介于手工配置和自动配置之间。通常，初始化时是手工操作，迁移时是自动完成。

15.3.3　交换机间的通信

在多交换式主干网中，每个交换机不仅必须知道哪个站点属于哪个VLAN，而且必须知道连接在其他交换机上的站点的成员。例如，在图15.17中，交换机A必须知道交换机B上连接的站点的成员状况。交换机B也必须知道交换机A同样的信息。有三个方法可以达到这个目的：表维护、帧标记和时分多路复用。

表维护

在这种方法中，当一个站点向它的组成员发送广播帧时，交换机将在一个表中增加一个条目，并记录站点的成员关系。交换机互相发送自己的表以定期更新。

帧标记

在这种方法中，当帧在交换机之间传输时，将会在MAC帧上加一个额外的头部来定义目标VLAN。接收帧的交换机用帧标记来决定接收广播消息的VLAN。

时分多路复用（TDM）

在这种方法中，交换机之间的连接（主干）被分成时分共享通道（见第6章的TDM）。例如，如果在一个主干网中VLAN的总数是5个，每个主干划分成5个通道，去往VLAN1的通信量在通道1中传输，去往VLAN2的通信量在通道2中传输，依此类推。通过检查帧到达时的通道，接收的交换机就可以确定目的VLAN。

15.3.4 IEEE标准

1996年，IEEE802.1小组委员会通过了称为802.1Q的标准，定义了帧标记的格式。这个标准也定义了在多交换式主干网中使用的格式，并且支持在VLAN中使用多家供应商的设备。IEEE802.1Q已经开放了与VLAN有关的其他问题进一步标准化的方法。多数的厂商已经接受这个标准。

优势

使用VLAN有以下优势：

降低费用和节省时间

VLAN可以降低站点从一个组到另一个组的迁移费用。物理上重新配置费时又昂贵。相对于把站点物理地移动到另一个网段或另一个交换机上，使用软件移动要容易和快捷得多。

建立虚拟工作组

VLAN可以用来建立虚拟工作组。例如，在大学校园环境中，工作于同一个项目中的教授可以互相发送广播报文，而没有必要属于同一个系。与以往使用的IP多播功能相比，这种方式降低了通信量。

安全

VLAN提供了附加的安全措施。属于同一个工作组的人们被授权允许发送广播报文，而其他工作组的人将接收不到这些报文。

推荐读物

为了更详细地了解与本章有关的主题，推荐阅读下面的读物和网站。本书末列出了方括号[……]中的参考资料。

读物

专门介绍连接设备的书是[Per00]。连接设备和VLAN在[Tan03]的4.7节中有讨论。交换机、网桥和集线器在[Stu03]和[Stu04]有讨论。

网站

- IEEE 802 LAN/MAN 标准委员会

本章小结

- 中继器是工作在因特网模型中物理层的连接设备。中继器再生信号，连接LAN的网段，但并没有过滤功能。
- 网桥是工作在因特网模型中物理层和数据链路层的连接设备。
- 透明网桥能转发和过滤帧并自动建立它的转发表。
- 网桥能使用生成树算法建立无回路拓扑。
- 主干LAN允许连接多个LAN。

- 主干网一般是总线型或星型。
- 虚拟局域网（VLAN）用软件而不是物理线路配置。
- VLAN中的成员关系基于端口号、MAC地址、IP地址、IP多播地址，或这些特性的组合。
- VLAN成本效益和时间效率都较高，能减少网络通信量，并提供额外的安全性措施。

习题

复习题

1. 中继器和放大器有何不同？
2. 当我们说网桥能过滤通信量是什么意思？为什么过滤很重要？
3. 什么是透明网桥？
4. 中继器怎样扩展LAN的长度？
5. 集线器和中继器有何不同？
6. 转发端口和阻塞端口的差异是什么？
7. 总线型主干网和星型主干网的差异是什么？
8. VLAN如何帮助公司节省时间和金钱？
9. VLAN如何为网络提供额外的安全性？
10. VLAN如何减小网络通信量？
11. VLAN中成员关系的基础是什么？

练习题

12. 在每个站发送一个分组给另一个站后完成图15.6中的表。
13. 得出图15.7中系统的生成树。
14. 如果移除网桥B5，得出到15.8中系统的生成树。
15. 如果移除网桥B2，得出图15.8中系统的生成树。
16. 如果网桥B5选为根网桥，得出图15.8中系统的生成树。
17. 在图15.6中，我们使用一个网桥。我们能用路由器代替网桥吗？请解释原因。
18. 网桥使用过滤表，路由器使用路由选择表。你能解释差别吗？
19. 建立有三个LAN和四个网桥的系统。网桥（B1到B4）如下连接LAN：
    a. B1连接LAN1和LAN2　　b. B2连接LAN1和LAN3
    c. B3连接LAN2和LAN3　　d. B4连接LAN1、LAN2和LAN3。
    B1选为根网桥。在应用生成树程序后，标出转发端口和阻塞端口。
20. 网桥或路由器哪一个开销更大？解释你的结论。
21. 中继器或网桥哪一个开销更大？解释你的结论。
22. 路由器或网关哪一个开销更大？解释你的结论。

第16章 无线WAN：移动电话和卫星网络

我们在第14章中讨论了无线局域网，无线技术也应用于移动电话和卫星网络中。本章讨论移动电话以及通道化访问方法（见12章）的实例。我们还要简要地讨论卫星网络，这是一种移动电话直接访问因特网的技术。

16.1 移动电话

移动电话（cellular telephony）被设计用来在两个被称为移动站点（MS）的移动单元之间，或者在一个移动单元和一个固定单元（通常称为地面信元）之间提供通信。服务提供商必须能够定位并追踪到主叫方，给呼叫分配通道，并且当主叫方移动到范围之外时，应该将通道从一个基站转移到另一个基站。

为了使这种追踪成为可能，每个移动电话服务区域都被分割成小的地区，称为信元（cell）。每个信元包括一个天线，并且被一个小的称为基站（BS）的由太阳能或AC供能的网络站点所控制。同样，每个基站被一个称为**移动交换中心**（mobile switching center，MSC）的交换局所控制。MSC协调所有基站和电话中心局之间的通信，它是一个计算中心，负责呼叫连接、记录呼叫信息和计费（见图16.1）。

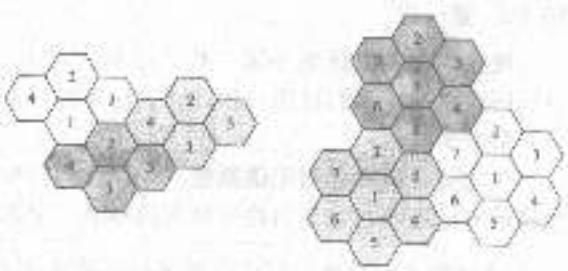


图16.1 移动系统

信元的大小并不固定，依据该地区人口的多少可以增大或缩小。信元的典型半径是1－12英里。与低人口密度的地区相比，高人口密度的地区需要更多的地理范围更小的信元来满足通信量的需求。信元大小一旦确定下来，应该是阻止邻近信元信号相互干扰的最佳大小。每个信元的传输能量都保持在较低水平以防止信号被其他信元干扰。

16.1.1 频率复用原理

通常，相邻的信元不能使用同一套频率来通信，因为当用户位于信元边界时可能产生干扰。然而，可用的频率集是有限制的，所以频率需要复用。频率复用的模式是N个信元的配置，N是复用因子（reuse factor），代表每个信元使用的唯一频率集。当这个模式重复时，频率就可以复用。有几种不同的模式，图16.2显示了其中的两种。



a) 复用因子为4　　b) 复用因子为7

图16.2 频率复用模式

信元中的数字定义了模式。在一个模式中有相同因子的信元可以使用相同的频率集，我们称这些信元为复用信元（reusing cell）。如图16.2所示，在复用因子是4的模式中，仅用一个信元隔开使用相同频率集的信元。在复用因子是7的模式中，用两个信元来隔开复用信元。

16.1.2 传输

为了从移动电话发出呼叫，主叫方要输入7或10位数字（电话号码），然后按发送按钮。移动站点扫描频带，寻找一个信号强的通道，然后就使用这个通道向最近的基站发送数据（电话号码）。基站向MSC转发数据，MSC接着向电话中心发送数据。如果被叫方有空闲，就建立连接并将结果返回给MSC。MSC就给呼叫分配一个没有使用的话音通道，建立连接。移动站点根据通道自动调整频率，然后就可以开始通信了。

16.1.3 接收

当某个移动电话被呼叫时，电话中心便将这个电话号码发送到MSC。MSC通过向每个信元发送查询信号来寻找该移动站点的位置。这个过程称为寻呼（paging）。一旦找到移动站点，MSC就发送一个铃声信号。当移动站点回答后，就给此次呼叫分配一个话音通道，此时便可以开始话音通信了。

切换

在会话中，可能发生移动电话从一个信元移动到另一个信元的情况。这时信号可能变弱。为解决这个问题，MSC每隔几秒钟就监测一次信号强度。当信号强度变小时，MSC就寻找一个新的可以更好提供通信的信元来接替上一个信元，然后MSC改变承载该呼叫的通道（将信号从旧的通道交换到新的通道）。

硬切换 早期的系统使用硬切换（handoff）。在硬切换中，移动站点只和一个基站通信。如果MS从一个信元移动到另一个信元，则必须先切断与当前基站的通信，再和新的基站建立通信。这可能建立一个粗转换。

软切换 新的系统使用软切换。在这种情况下，移动站点可以同时和两个基站通信。这意味着，在切换过程中，移动站点可能在切断和旧基站的联系之前，就可以使用新的基站继续通话。

16.1.4 漫游

移动电话的一个特征，称为漫游（roaming）。漫游意味着原则上只要在覆盖范围以内，用户都能够发出呼叫或者被呼叫到。服务提供商的覆盖范围通常是有限的，通过漫游联系，邻近的服务提供商可以提供扩展的覆盖范围，这种情况类似于在两个国家间传送信件。两个国家间分发信件的费用可以由他们在协商同意的情况下分组。

16.1.5 第一代

现在的移动电话处于第二代，而第三代也已浮出水面。第一代设计使用模拟信号，用来进行话音通信。我们讨论一种在北美使用的移动系统，即AMPS。

AMPS

在北美，高级移动电话系统（Advanced Mobile Phone System，AMPS）是一种领先的模拟移动电话系统。它在链路中使用FDMA（见第12章）来分隔通道。

> AMPS是一种使用FDMA的模拟移动电话系统。

频段 AMPS工作在ISM 800MHz频段。系统使用两个单独的模拟通道，一个用于转发

（从基站到移动电话）通信，一个用于接收（从移动电话到基站）通信。824MHz－849MHz的频段用于接收通信，869MHz－894MHz的频段用于转发通信，见图17.3所示。



图16.3 AMPS的移动频段

每个频段分为832个通道。然而，如果两个提供商共享一个地区，这就意味着每个提供商的信元只有416个通道。其中21个通道用于控制，这就还剩下395个通道。AMPS的频率复用因子是7，这意味着在一个信元中仅有395个通道的1/7可以使用。

传输 AMPS使用FM和FSK来调制信号。图17.4显示了反向的传输。话音通道使用FM调制，而控制通道使用FSK调制以生成30kHz的模拟信号。AMPS使用FDMA来将每个25MHz的频段分离成30kHz的通道。

16.1.6 第二代

为提供更高质量（更少的噪声）的移动话音通信，第二代移动电话网络被开发出来。相对于第一代设计用于模拟话音通信而言，第二代主要设计用于数字话音通信。如图16.5所示，第二代有三种主要的系统，我们将逐一讨论。

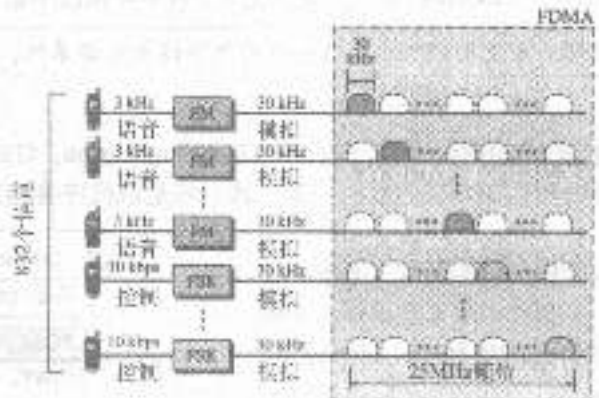


图16.4 AMPS反向通信频带

D-AMPS

模拟AMPS演化成数字系统的一个产品是数字AMPS（digital AMPS，D-AMPS）。D-AMPS设计为向后兼容AMPS，这意味着，在一个信元里，一个电话可以使用AMPS，而另一个可以使用D-AMPS。D-AMPS最初由IS-54（过渡标准54）定义，后来在IS-136中修订。



图16.5 第二代移动电话系统

频段 D-AMPS使用和AMPS同样的频段和通道。

传输 使用一种非常复杂的PCM和压缩技术来对每个话音通道进行数字化。将每个话音

通道数字化为7.95Kbps，使用TDMA合成3个7.95Kbps的数字话音通道，结果是48.6Kbps的数字化数据。这里有很大的额外开销。如图16.6所示，系统每秒发送25帧，每帧1 944位，每帧持续40ms（1/25），并且分为6个时隙，被3个数字通道共享，每个通道分配2个时隙。

每时隙容纳324位。然而，只有159位来自数字化的语音，64比特用于控制，101比特用于纠错。换句话说，在分配给它的两个通道中，每个通道都将携带159位的数据，系统加上64个控制位和101个纠错位。

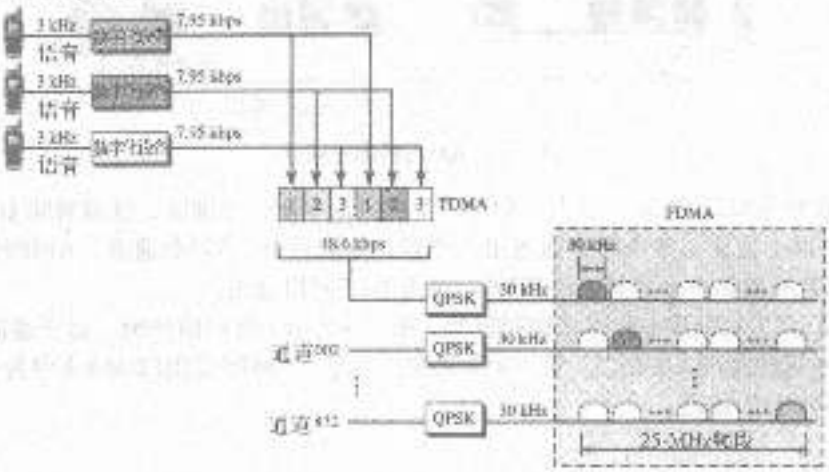


图16.6　D-AMPS

使用QPSK将48.6Kbps的数字信号调制成载波信号，结果是30kHz的模拟信号。最后，多个30kHz的模拟信号共享一个25MHz的频段。D-AMPS的频率复用因子是7。

> D-AMPS，即IS-136，是使用TDMA和FDMA的数字移动电话系统。

GSM

全球移动通信系统（Global System for Mobile Communication，GSM）是为所有欧洲国家提供通用的第二代移动通信而开发的欧洲标准。其目标是取代许多相互不兼容的第一代移动通信技术。

频段　GSM使用两个频段以实现双工通信。每个频段宽度为25MHz，向900MHz方向推移。如图16.7所示。每个频段划分成200kHz的124个通道，这些通道被防护频段隔开。



图16.7　GSM频段

传输　图17.8显示了一个GSM系统。每个话音通道被数字化并压缩为13Kbps的数字信号。每个时隙携带156.25位。8个时隙混合一起形成一个TDM帧。26个帧组合成一个多帧（TDMA）。我们可以按如下方法计算出每个通道的比特率：

$$\text{通道数据速率} = (1/120\text{ms}) \times 26 \times 8 \times 156.25 = 270.8\text{ kbps}$$

每个270.8Kbps的数字通道使用GMSK（主要在欧洲系统中使用的一种FSK形式）调制成载波信号，结果是200kHz的模拟信号。最后，124个200kHz的模拟通道使用FDMA合成在一起，结果是25MHz波段。图16.9显示了在一个多帧中的用户数据和开销。

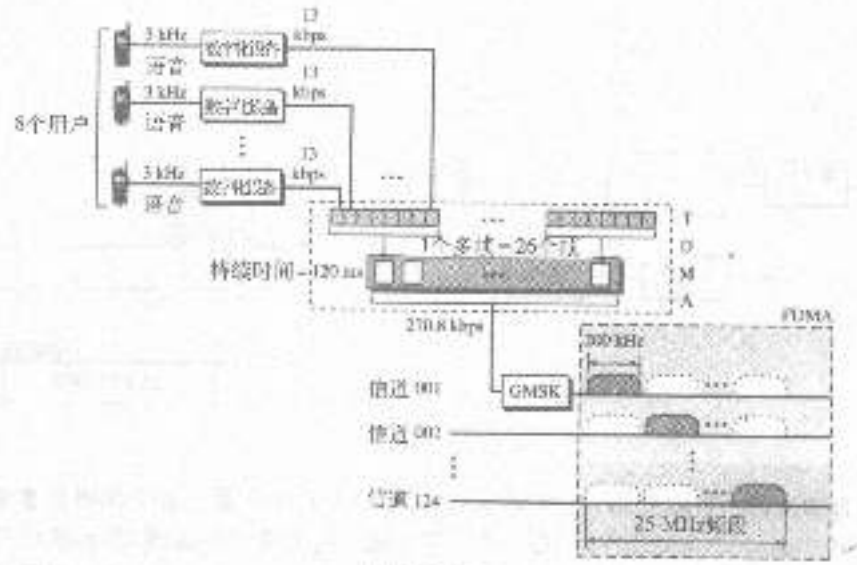


图16.8 GSM

读者可能已经注意到，在TDMA中有大量的开销，每个时隙的用户数据仅有65位。系统加上用于纠错的附加位使得每个时隙达到114位。就这样，加上控制位使每个时隙达到156.25位。8个时隙封装成一帧。24个通信帧和2个附加的控制帧形成一个多帧。一个多帧的持续时间为120ms。然而，这个结构确实定义了不附加任何开销的超帧和超大帧，我们这里不讨论它们。

复用因子　由于复杂的纠错机制，GSM支持的复用因子可以低到3。

> GSM是一个使用TDMA和FDMA的数字移动电话系统。

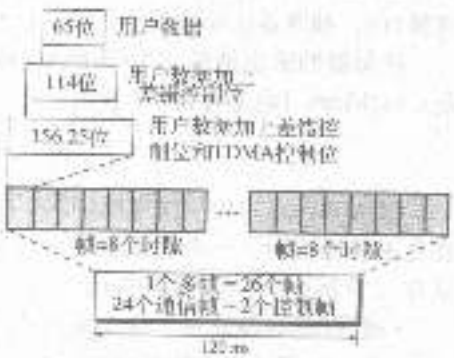


图16.9 多帧组成

IS-95

在北美，一种占统治地位的第二代标准是过渡标准95（Interim Standard 95，IS-95）。它基于CDMA和DSSS。

频段和通道　IS-95为实现全双工通信使用了两个频段，频段可以是传统的ISM-800MHz，或者是ISM-1 900MHz。每个频段划分成20个1.228MHz并被防护频段隔开的通道。给每个服务提供商分配10个通道。IS-95可以用来和AMPS并行工作。每个IS-95通道相当于41个AMPS通道（41 × 30kHz = 1.23MHz）。

同步　所有使用CDMA的基本通道都需要同步。为了提供同步，基站使用GPS（全球定位系统）的服务。GPS是一个卫星系统，我们将在下节讨论。

前向传输　IS-95有两种不同的传输技术：一个用在前向（从基站到移动站）方向，另一个用在反向（从移动站到基站）。在前向方向上，基站和所有移动站的通信都要同步，由基站发送同步数据给所有的移动站。图16.10是一个前向方向上的简化图示。

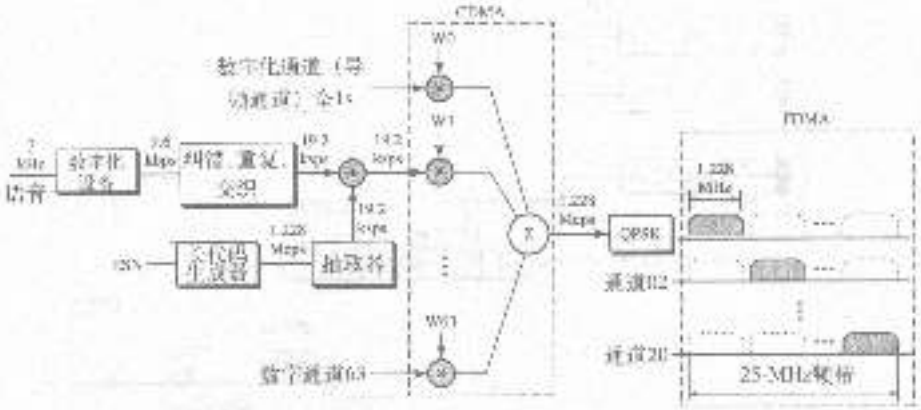


图16.10　IS-95前向传输

每个话音通道都是数字化的，以9.6Kbps的基本速率产生数据，加上纠错和重复位和交织后，结果是19.2ksps（千信号每秒）的信号。现在输出结果就使用这19.2ksps的信号来进行扰频处理。由一个长代码生成器来产生这些扰频的信号。这个长代码生成器使用移动站的电子序列号（ESN）生成242个伪随机芯片。每个芯片有42位。注意，这些芯片是伪随机生成的，而不是随机的，因为样本重复使用。长代码生成器的输出供应给抽取器，抽取器从每64位中选择1位。抽取器的输出用于扰频，而扰频用于保密，因为每个移动站的ESN是唯一的。

扰频器的输出结果使用CDMA合成。每个通信通道选择一个Walsh 64×64行芯片，结果是1.228Mcps（兆芯片每秒）的信号：

$$19.2\text{ksps} \times 64\text{cps} = 1.288\text{Mcps}$$

CDMA合成的信号供应给QPSK调制器，生成1.288MHz的信号。使用FDMA，产生的带宽也正确地移动。一个模拟通道可以生成64个数字通道，其中55个是通信通道（携带数字化话音），9个通道用于控制和同步：

- 通道0是导频通道，这个通道不间断地向移动站发送1s的数据流。这个数据流提供位同步，也可以作为解调的相位参考，并且也允许移动站与邻近基站的信号强度进行比较以做出切换决定。
- 通道32向移动站给出关于系统的信息。
- 通道1～7用于调度，以向一个或多个移动站发送信息。
- 通道8～31和33～63是携带从基站到相应移动站的数字化话音的通信通道。

反向传输　在前向方向上使用CDMA是可行的，因为导频通道不间断地发送1s的序列来同步传输。在反向传输中不能使用同步，因为我们需要一个实体来做这些，但这是不可行的。反向通道不是使用CDMA，而是使用DSSS（直接序列扩频）。这在第15章中已经讨论过。图16.11是反向传输的简化图。

每个话音通道都数字化的，以9.6kbps的速率产生数据。然而，加上纠错和重复位后，再加上交织，结果是28.8ksps的信号。这个输出经过一个6/64符号调制器，这些符号被分成六符号的大块，每个大块用一个二进制数（0～63）来说明。这个二进制数用做选择芯片行的64×64 Walsh矩阵的索引。注意，这个过程不是CDMA，每位没有被行中的芯片相乘。每个六符号的块被一个64芯片代码替代。这样做是为了提供一种正交性，它与从不同移动站发出的芯片流有区别。结果是生成了307.2kcps或者（28.8/6）×64的信号。

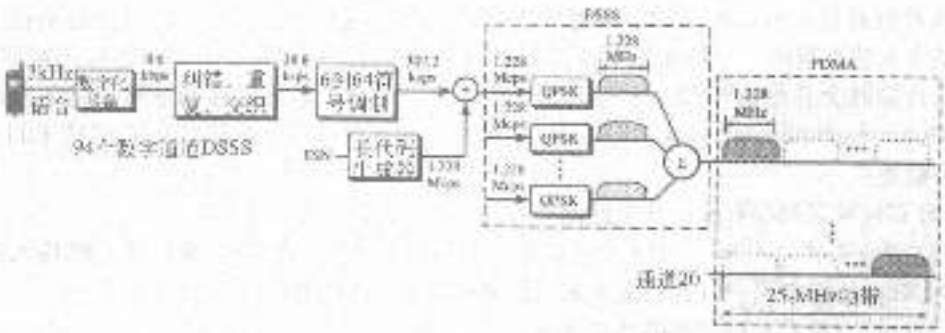


图16.11 IS-95反向传输简化图

下一步是扩频，每个芯片被扩充为4个。另外，移动站的ESN以1.228Mcpc（307.2Kcps的4倍）的速率生成一个42位的长代码。扩频后，每个信号使用QPSK来调制，这里的QPSK和在前向方向中使用的有细微差别，我们这里不讨论细节。注意，这里没有多路访问机制，所有的反向通道都向空气中发送模拟信号。但是由于扩频，基站只接收正确的芯片。

尽管在反向上可以生成$2^{42}-1$个数字通道（因为有长代码生成器），但通常只使用94个通道，其中62个用于通信通道，32个通道用于和基站接入。

IS-95是一个使用CDMA/DSSS和FDMA的数字移动电话系统。

两套数据速率 IS-95定义了两套数据速率，每套中有4个不同的速率。第一套定义了9 600、4 800、2 400和1 200bps。例如，如果选定的速率是1 200 bps，每位重复8次以提供9 600 bps的速率。第二套定义了14 400、7 200、3 600和1 800 bps。通过减少用于纠错的位数，这个速率是可能的。比特率与通道的活动状态有关。如果通道是静止的，则只能传输1 200位。通过每位重复8次来改善扩频。

频率复用因子 在IS-95系统中，频率复用因子通常是1，因为邻近信元的干扰不会影响CDMA和DSSS传输。

软切换 每个基站使用它的导频通道不间断地广播信号。这意味着移动站可以探测到从所在信元和邻近信元的导频信号。相对硬切换而言，这可以允许移动站做软切换。

PCS

在结束第二代移动电话的讨论之前，让我们讨论一个经常听到的关于第二代的术语：PCS。**个人通信系统**（Personal communications system，PCS）不是指类似GSM、IS-136或者IS-95等的单独技术，它是一个提供多种通信服务的业务系统的总称。这些系统的一般特征可以概括如下：

1. 它们可以使用任何第二代技术（GSM、IS-136或者IS-95）。
2. 它们使用1 900MHz频段。这意味着移动站需要更多的能量，因为高频比低频有更短的传输距离。然而，因为基站的功率被FCC限制，所以基站和移动站之间的距离要更近一些（更小的信元）。
3. 它们提供诸如短消息服务和有限因特网接入这样的通信服务。

16.1.7 第三代

第三代移动电话技术是指提供各种各样服务的技术的综合。理想情况下，当技术成熟时，第三代系统可以提供数字数据通信，也可以提供话音通信，使用一个小的手持设备，应该可以和世界上其他任何人通话，并且通话质量和现有的固定电话网络相似。可以下载并观看电影，

可以下载并听音乐，可以网上冲浪或者玩游戏，可以开视频会议……可以做的还有很多。第三代系统令人感兴趣的一个特征是手持设备可以一直连接，不需要拨号即可连接因特网。

第三代的概念开始于1992年，当时ITU颁布了一个称为国际移动通信-2000（International Mobile Communication for year 2000，IMT-2000）的蓝图。这个蓝图定义了3G技术的一些准则，如下概述：

- 话音质量可以和现有的公共电话网络相比。
- 移动交通工具（汽车）的接入速率为144Kbps，当用户行走（步行者）时接入速率为384 Kbps。固定用户（办公室或家里）的接入速率为2Mbps。
- 支持分组交换和电路交换的数据服务。
- 一个2GHz的频段。
- 2MHz的带宽。
- 到因特网的接口。

> 第三代移动电话技术的主要目标是提供普遍的个人通信。

IMT-2000无线电接口

图16.12展示了IMT-2000采用的无线电接口（无线标准）。所有这五个接口都是从第二代技术发展而来。前两个是从CDMA技术发展而来，第三个是从CDMA和TDMA的联合使用技术发展而来，第四个从TDMA发展而来，最后一个从FDMA和TDMA发展而来。

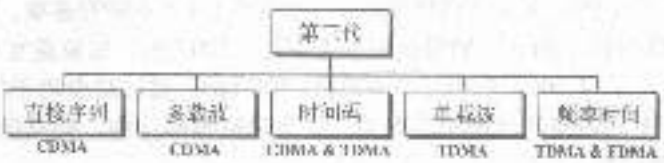


图16.12　IMT-2000无线电接口

IMT-DS　这种方法使用CDMA的一个称为宽带CDMA或W-CDMA的版本。W-CDMA使用5MHz的带宽，它在欧洲开发，并和在IS-95中使用的CDMA兼容。

IMT-MC　这种方法在北美开发，并以CDMA 2000而闻名。它是对在IS-95通道中使用的CDMA技术的一次革命。它联合使用新的宽带（15MHz）扩频和IS-95中的窄带（1.25MHz）CDMA。它向后兼容IS-95，支持多路1.25-MHz的通道（1，3，6，9，12倍），最高达15MHz。更宽通道的使用使其可以达到第三代标准定义的2-Mbps的数据速率。

IMT-TC　这个标准联合使用W-CDMA和TDMA，通过将TDMA多路技术加到W-CDMA上而努力达到IMT-2000的目标。

IMT-SC　这个标准仅使用TDMA。

IMT-FT　这个标准联合使用FDMA和TDMA。

16.2　卫星网络

卫星网络（satellite network）是多个节点的联合体，一些节点是卫星，提供从地球上某点到另一点的通信。网络中的节点可以是卫星、地面站、用户终端或者电话。尽管真实的卫星，例如月球，可以作为网络中的中继节点，但是倾向于使用人造卫星，因为可以在卫星上安装电子设备以重新生成在传输中损失能量的信号。另一个使用自然卫星的限制是它们距离地球太远，这将在通信中产生很长的延迟。

卫星网络和移动电话网络的相同点是，它们都将地球划分成信元。卫星可以提供到达或来自地球上任意地区的传输能力，而不管有多远。这种优势使得不需要大量的地面基础设施的投资而提供与世界上不发达地区的高质量通信成为可能。

16.2.1 轨道

人造卫星需要有一个轨道（orbit），即绕地球旋转的路径。轨道可以是赤道轨道、倾斜轨道或者是极地轨道，如图16.13所示。

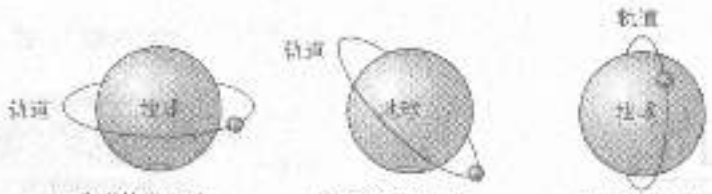


图16.13 卫星轨道

卫星的周期，即绕地球旋转一圈所需的时间，由开普勒定律决定，该定律定义了周期是一个卫星到地球中心的距离的函数。

例16.1

根据开普勒定律，月球的周期是多少？

$$周期=C\times 距离^{3/2}$$

这里，C是一个常数，大约等于1/100，周期的单位是s，距离的单位是km。

解：

月球距离地面大约384 000km，地球的半径是6 378km，应用上述公式，我们得到：

$$周期=\frac{1}{100}(384\,000+6\,378)^{3/2}=2\,439\,090=1个月$$

例16.2

根据开普勒定律，一个定位在距离地球35 786km的轨道的卫星的周期是多少？

解：应用公式，我们得到：

$$周期=\frac{1}{100}(35\,786+6\,378)^{3/2}=86\,579=24小时$$

这意味着一个定位在距离地球35 786km的轨道的卫星的周期是24小时，这和地球的自转周期相同，这样的卫星相对地球是静止的，这样的轨道，称为地球同步轨道。

16.2.2 覆盖区域

卫星使用双向的天线（在视线范围内）进行微波通信。因此，从卫星上发出的信号只瞄准一个特定的区域，该区域称为覆盖区域（footprint）。在覆盖区域的中心，信号能量最强，当远离覆盖区域中心，信号能量变弱。覆盖区域的边界是那些信号能量达到预先定义的极限值的地区。

16.2.3 三种类型的人造卫星

根据轨道的位置，卫星可以分为三类：地球同步轨道（geostationary earch orbit，GEO）卫星、近地轨道（low-Earth-orbit，LEO）卫星和中地轨道（middle-Earth-orbit，MEO）卫星。

图16.14显示了这种分类法。

图16.15显示了卫星距离地球表面的高度。对GEO来说，只有一个高度在35 786km的轨道。MEO定位在高度介于5 000km和15 000km之间。LEO通常低于2 000km的高度。

拥有不同轨道的一个原因，是由于有两个范艾伦（Van Allen）辐射带的存在。Van Allen辐射带是包含带电粒子的层。在这两个带中环绕的卫星将被充满能量的带电粒子完全破坏。MEO的轨道定位在这两个带之间。

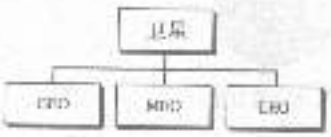


图16.14　卫星分类



图16.15　卫星轨道高度

卫星通信的频段

保留的用于卫星微波通信的频率处于千兆赫兹（GHz）范围内。每颗卫星使用两个以上不同的频段发送和接收。从地球到卫星的传输称为卫星上行链路（uplink）。从卫星到地球的传输称为卫星下行链路（downlink）。表16.1给出了频段名称和每个范围的频率。

表16.1　卫星频段

频段	下行链路/GHz	上行链路/GHz	带宽/MHz
L	1.5	1.6	15
S	1.9	2.2	70
C	4.0	6.0	500
Ku	11.0	14.0	500
Ka	20.0	30.0	3 500

GEO卫星

视线传播要求发送和接收天线能够随时锁定彼此的位置（一个天线必须在另一个天线的视线之内）。因为这个原因，比地球自转快或慢的卫星只有在较短的时间可以使用。为了确保稳定的通信，卫星必须以与地球同样的速度移动，这样看起来好像在一个特定点保持固定一样。这种类型的卫星称为地球同步轨道（GEO）卫星。

因为轨道速度与距地球的距离有关，因此只有一个轨道可以是同步的。这个轨道在地球赤道平面上，距地球表面大约22 000英里。

但是，一个同步卫星不能覆盖整个地球。尽管一颗在轨道上的卫星可以和视线之内很大数量的地面站保持联系，但是地面的曲率仍然使地球的很多部分处于视线之外。至少需要在地球同步轨道（geosynchronous Earth orbit，GEO）上等距的三颗卫星来实现全球传输。图16.16显示了这三颗卫星，在围绕赤道的同步轨道上，每个卫星相隔120°，视点在北极。

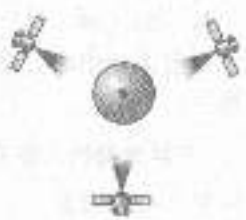


图16.16　在地球同步轨道上的卫星

MEO卫星

中地轨道（Medium-Earth orbit，MEO）卫星定位在两个Van Allen辐射带之间。这个轨道上的卫星大约需要6～8小时环绕地球一圈。

GPS

MEO卫星系统的一个实例是全球定位系统（Global Positioning System，GPS），轨道位于地球上空海拔约18 000km（11 000英里）的高度。尽管GPS是美国国防部建立的，但现在是一个公用系统。这个系统包括24颗卫星，用于地面和航海，为车辆和轮船提供时间和定位服务。GPS使用6个轨道上的24颗卫星，如图16.17所示。在每个轨道上的卫星，其轨道和位置基于如下原则来设计，即在任何时间，在地球上的任意点，都可以同时看到4颗卫星。每个GPS接收器都有一个天文年历来得知卫星的当前位置。



图16.17 全球定位系统（GPS）卫星的轨道

三边测量 GPS基于称为三边测量[⊖]（trilateration）的原理。在平面上，如果我们知道与三个点的距离，就可以准确地知道我们在哪里。假设我们距A点10英里，距B点12英里，距C点15英里。如果分别以A、B和C为中心画三个圆，我们必然处在圆A、圆B和圆C上的某点。这三个圆相交于一点（如果我们的距离准确的话），这个点就是我们的位置。图16.18a解释了这个概念。

在三维空间上，情况就不一样了，三个球相交到两个点，如图16.18b所示。我们至少需要四个球体才能知道我们在空间中准确的位置（经度、纬度和高度）。但是，如果我们有与位置相关的其他信息（比如，我们知道我们不在海洋里或是空间的其他地方），三个球体就够了。因为球体相交的两个点中，如果选了另一个，这一个就毫无疑问是不可能的了。

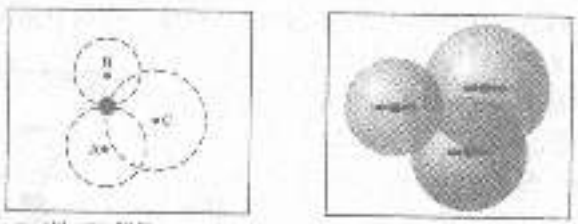

a）二维三边测量　　b）三维三边测量

图16.18 二维平面的三边测量

测距 如果我们知道距三个卫星的距离以及每个卫星的位置，那么就可以使用三边测量原理找出我们在地球上的位置。每个卫星的位置可以由GPS接收器计算得出（使用卫星预先确定的路径）。然后GPS接收器需要获得至少距三个卫星的距离（球心）。测距是通过称为单向测距的原理实现的。这时，我们假定所有GPS卫星和地球上的接收器是同步的。24个卫星中的每一个同步地传输一个复杂信号，每个信号有一个唯一的模式。接收器上的计算机测量来自卫星的信号和信号副本之间的延迟，以确定到各个卫星的距离。

同步 前面的讨论基于一个假设，即卫星的时钟和接收器的时钟是互相同步的。卫星使用精确的原子钟，能互相同步地工作。但是，接收器的时钟是一般的石英钟（原子钟比石英

⊖ 术语三边测量（trilateration）和三角测量（triangulation）通常是互用的。这里使用三边测量，指的是三个距离，而不是三个角度。

钟贵50 000美元），没有办法与卫星时钟达到同步。卫星时钟和接收器时钟间有个未知的偏移量，使得测距时引入了相应的偏移量。当为这个偏移量，测得的距离称为伪距。

GPS使用一种优秀的解决方案来解决时钟偏移问题。通过分析得知偏移值对所有卫星都是一样的。位置的计算变成求四个未知量：x_r、y_r、z_r（接收器的坐标）以及通用时钟偏移dt_r。为了求这四个未知量，我们需要至少四个方程。这意味着我们需要测量到四个卫星而不是三个卫星的伪距。如果测得的四个伪距是PR1、PR2、PR3和PR4，每个卫星的坐标是x_i、y_i和z_i（i=1到4），我们可以使用如下四个方程求出前面提到的未知量（四个未知量用黑体标出）。

$$PR_1=[(x_1-\mathbf{x_r})^2+(y_1-\mathbf{y_r})^2+(z_1-\mathbf{z_r})^2]^{1/2}+c\times dt$$
$$PR_2=[(x_2-\mathbf{x_r})^2+(y_2-\mathbf{y_r})^2+(z_2-\mathbf{z_r})^2]^{1/2}+c\times dt$$
$$PR_3=[(x_3-\mathbf{x_r})^2+(y_3-\mathbf{y_r})^2+(z_3-\mathbf{z_r})^2]^{1/2}+c\times dt$$
$$PR_4=[(x_4-\mathbf{x_r})^2+(y_4-\mathbf{y_r})^2+(z_4-\mathbf{z_r})^2]^{1/2}+c\times dt$$

上面公式使用的坐标在地球中心地球固定（ECEF）参考帧中。这表示坐标空间的原点是地球球心，坐标空间绕地球旋转。这说明地球表面固定点的ECEF坐标不会改变。

应用　军队也使用GPS。例如，在海湾战争中，步兵、车辆和直升飞机使用了数以千计的便携式GPS接收器。GPS的另一个应用是导航。汽车驾驶员可以找出汽车的位置，然后查阅汽车上存储的数据库，以指引到目的地。换句话说，GPS给出汽车的位置，数据库使用这些信息来找出一条到达目的地的路径。正如我们先前提到过的，IS-95移动电话系统就是使用GPS在基站间实现同步。

LEO卫星

近地轨道（LEO）卫星有极地轨道，高度在500－2 000km之间，自转周期为90－120分钟，速度为20 000－25 000km/h。LEO系统通常有移动类型的访问方式，这和移动电话系统类似。覆盖范围通常是直径8 000km的区域。因为LEO卫星离地球近，来回传播的时间延迟通常少于20ms，这对于视频通信来说是可以接受的。

LEO卫星系统由像网络一样协同工作的一群卫星构成，每颗卫星担当交换机的角色。相互邻近的卫星通过卫星间链路（ISL）连接起来。移动系统通过用户移动链路（UML）和卫星通信。卫星也可以通过网关链路（GWL）和地面站（网关）通信。图17-19显示了一个典型的LEO卫星网络。

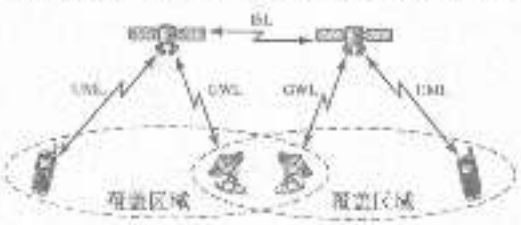


图16.19　LEO卫星系统

LEO卫星可以分为三类：小型LEO、大型LEO和宽带LEO。小型LEO卫星工作频率低于1GHz，通常用于低数据速率通信。大型LEO卫星工作于1GHz－3GHz之间。全球星（Globalstar）和铱星（Iridium）系统是大型LEO卫星系统的实例。宽带LEO卫星提供与光纤网络类似的通信。第一个宽带LEO卫星系统是Teledesic。

铱星系统　是一个77颗卫星的系统。其概念是在1990年由Motorola公司提出的。这个项目花了8年时间来实现。在这期间，卫星的数目减少了。最后，在1998年，拥有66颗卫星的系统开始提供服务。最初的名称铱，来自于第77号化学元素，其实更适当的名称应该是镝（第66号元素的名称）。

铱星系统经历了很多波折。1999年，由于财政问题，该系统停建，随后被卖掉。2001年，该系统重新启动。

这个系统有66颗卫星，分成6个轨道，每个轨道有11颗卫星，轨道位于750km的高度。每个轨道上的卫星大约每隔纬度32°相互隔开。图16.20是这群卫星的示意图。

铱星系统在6个LEO轨道上有66颗卫星，每个轨道的高度是750km。

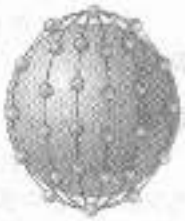


图16.20 铱星示意图

由于每个卫星有48个点波束，因此系统可以有多达3 168个波束。然而，当卫星到达极地时，一些波束被关掉。任意时刻活动的点波束的数目大约是2 000个。每个点波束覆盖地球的一个信元，这也意味着地球被分成大约2 000个（重叠）信元。

在铱星系统中，两个用户之间的通信是通过卫星来进行的。当一个用户呼叫另一个用户时，这个呼叫在到达目的地之前可能通过好几个卫星，这意味着做了适当的中继。并且每颗卫星都需要改进以适合中继的需要。这种策略排除了需要许多地面站的限制。

铱星系统的整个目标是通过手持终端（和移动电话同样的概念）直接提供全球范围的通信。该系统可以用于话音、数据、调度、传真甚至导航。该系统可以在不能使用其他通信类型的地区的用户之间提供连通性。系统在便携式电话之间提供2.4～4.8kbps的话音和数据传输。传输使用1.616－1.6126GHz波段，卫星间通信使用23.18－23.38GHz波段。

铱星系统设计用来提供使用手持终端进行直接的全球范围的话音和数据通信服务。类似于移动电话的服务，但是在全球范围内。

全球星系统 是另一个近地轨道卫星系统。该系统使用在6个轨道上的48颗卫星，每个轨道8颗卫星。轨道定位于接近1 400km的高度。全球星系统类似于铱星系统，主要的区别在于中继机制。在铱星系统中，两个远距离用户之间的通信需要在几颗卫星之间传递，而全球星系统的通信既需要卫星，也需要地面站，这意味着地面站可以产生能量更高的信号。

Teledesic 是一个提供类似光纤（宽带通道、低差错率和低延迟）通信的卫星系统。其主要目的是为全世界的用户提供宽带因特网接入服务，有时也称之为"空中互联网"。这个项目由Craig McCaw和Bill Gates开始于1990年，随后有其他投资者加入。该项目计划2005年实现全部功能。

卫星群 Teledesic提供288颗卫星，分布在12个轨道上，每个轨道24颗卫星。轨道在1 350km的高度上，如图16.21所示。

Teledesic有288颗卫星，分布在12个LEO轨道上，每个轨道的高度为1 350km。

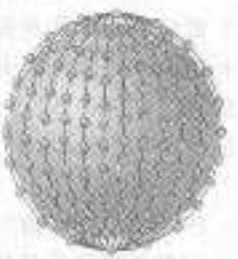


图16.21 Teledesic

通信 系统提供二种通信方式。卫星间通信允许8个相邻的卫星相互通信。在卫星和地面网关站之间通信也是可能的。用户使用终端可以直接和网络通信。地面被划分为数以万计的信元，每个信元分配一个时隙，卫星在相应的时隙将波束集中到该信元。终端在自己的时隙里可以发送数据。终端接收所有目的地是该信元的分组，但是只选择目的地是自己地址的分组。

波段 传输使用Ka波段。

数据速率 卫星上行链路的数据速率可以达到155Mbps，卫星下行链路的数据速率可以达到1.2Gbps。

推荐读物

为了详细地讨论与本章有关的主题，我们推荐下列读物。在本书末列出方括号[……]中的参考资料。

读物

无线WAN在[Sta02]、[Jam03]、[AZ03]和[Sch03]中有完整讨论。通信卫星在[Tan03]的第2.4节和[Cou01]的第8.5节有讨论。移动电话系统在[Tan03]的第2.6节和[Cou01]的第8.8节有讨论。

本章小结

- 移动电话提供两个设备间的通信，一个设备或者两个设备是移动的。
- 将移动服务区域划分成信元。
- 高级移动电话系统（AMPS）是第一代移动电话系统。
- 数字AMPS（D-AMPS）是第二代移动电话系统，它是AMPS的数字版本。
- 全球移动通信系统（GSM）是欧洲使用的第二代移动电话系统。
- 过渡标准95（IS-95）是基于CDMA和DSSS的第二代移动电话系统。
- 第三代移动电话系统提供普遍个人通信。
- 卫星网络使用卫星来提供地球上任何点之间的通信
- 同步地球轨道（GEO）在赤道平面上，绕地球旋转。
- 全球定位系统（GPS）卫星是中地轨道（MEO）卫星，能提供车辆和船只的时间和位置信息。
- 铱星卫星是近地轨道（LEO）卫星，提供手持终端的直接普遍语音和数据通信。
- Teledesic卫星是近地轨道卫星，提供普遍宽带因特网访问。

习题

复习题

1. 基站和移动交换中心之间的关系是什么？
2. 移动交换中心的功能是什么？
3. 低复用因子和高复用因子那一个更好？解释你的结论。
4. 硬切换和软切换有什么不同？
5. 什么是AMPS？
6. D-AMPS和AMPS之间的关系是什么？
7. 什么是GSM？
8. IS-95中CDMA的功能是什么？
9. 三种类型轨道是什么？
10. GEO卫星有几种轨道？解释你的答案。
11. 什么是覆盖范围？
12. Van Allen辐射带和卫星之间的关系是什么？
13. 比较上行链路和下行链路。
14. GPS的目的是什么？
15. 铱星系统和全球星系统的主要区别是什么？

练习题

16. 画出频率复用因子为3的信元模式。
17. AMPS中每个信元的最大主叫数是多少？
18. 假设没有模拟控制通道，IS-136（D-AMPS）系统中每个信元的最大同时呼叫数是多少？
19. 假设没有模拟控制通道，GSM系统中每个信元的最大同时呼叫数是多少？
20. IS-95中每个信元的最大主叫数是多少？
21. 求出每兆赫兹带宽同时呼叫的AMPS的效率。换言之，求用于1MHz带宽分配的呼叫数。
22. 改为D-AMPS，重复练习21。
23. 改为GMS，重复练习21。
24. 改为IS-95，重复练习21。
25. 说出使用AMPS的系统中，3kHz语音通道和30kHz调制通道之间的关系。
26. 使用D-AMPS的系统中每秒发送多少个时隙？每个用户在1s内发送多少个时隙？
27. 使用开普勒公式校验GPS卫星给定周期和高度的正确性。
28. 使用开普勒公式校验铱星系统卫星给定周期和高度的正确性。
29. 使用开普勒公式校验全球星系统卫星给定周期和高度的正确性。

第17章 广域网SONET/SDH

在本章中，我们介绍一种广域网SONET，它能够承载来自其他WAN数据的传输网络。我们先把SONET作为一种协议进行讨论，然后说明如何从协议定义的标准构建SONET网络。

光纤的高带宽适用于今天的高数据速率技术（比如视频会议）和低速率下同时承载大量的数据。基于这个原因，光纤和要求高数据速率或者高带宽传输的技术共同发展，继而有了标准化的需要。因此，美国组织（ANSI）和欧洲组织（ITU-T）定义了相应的标准，虽然两种标准是独立的，但是基本功能相似并且最终是兼容的。ANSI标准成为同步光纤网络（Synchronou Optical Network，SONET），ITU-T标准成为同步数字体系（Synchronous Digital Hierarchy，SDH）。

SONET由ANSI定义；SDH由ITU-T定义。

SONET/SDH是使用同步TDM多路复用的同步网络。系统中的所有时钟都锁定于主时钟。

17.1 体系结构

让我们先介绍SONET系统的体系结构：信号，设备和连接。

17.1.1 信号

SONET定义了称为同步传输信号（synchronous transport signal，STS）的电子信号等级体系。每个STS等级（STS-1到STS-192）支持特定的数据速率（以每秒兆位规定），见表17.1。相应的光信号称为光载波（optical carrier，OC）。SDH也规定了一个相似的系统，称为同步传输模块（synchronous transport module，STM），STM用于与现存欧洲体系（比如E线路）和STS等级兼容。最后，最低的STM等级STM-1定义为155.520Mbps，正好等同于STS-3。

表17.1 SONET/SDH 速率

STS	OC	速率(Mbps)	STM	STS	OC	速率(Mbps)	STM
STS-1	OC-1	51.840		STS-24	OC-24	1244.160	STM-8
STS-3	OC-3	155.520	STM-1	STS-36	OC-36	1866.230	STM-12
STS-9	OC-9	466.560	STM-3	STS-48	OC-48	2488.320	STM-16
STS-12	OC-12	622.080	STM-4	STS-96	OC-96	4976.640	STM-32
STS-18	OC-18	933.120	STM-6	STS-192	OC-192	9953.280	STM-64

粗略一看表17.1，就能看出一些有趣的地方。首先，这个体系中最低等级的速率是51.840Mbps，它比DS-3服务（44.736Mbps）大。实际上，STS-1设计用于等同于DS-3的速率，提供容量上的差异来处理光系统中的开销需要。

其次，STS-3的速率正好是STS-1速率的三倍，STS-9的速率正好是STS-18速率的一半。这些关系意味着18个STS-1通道能复用成一个STS-18通道，六个STS-3通道能复用成一个STS-18通道，依此类推。

17.1.2 SONET设备

图17.1显示了一个使用SONET设备的简单链路。SONET传输依靠三个基础设备：STS多

路复用器/分离器、再生器、分插复用器和终端。

STS多路复用器/分离器

STS多路复用器/分离器标记SONET链路的开始点和结束点。它们提供了电子支路网络和光网络之间的接口。STS多路复用器（STS multiplexer）复用来自多个电子源的信号，并产生相应的OC信号。STS分离器（STS demultiplexer）将光OC信号分解成相应的电子信号。

再生器

再生器扩展了链路的长度。再生器（regenerator）是个中继器（见第15章），它把接收到的光信号（OC-n）解调成相应的电信号（STS-n），再生成电信号，最后把电信号调制成相应的OC-n信号。SONET再生器用新的信息替换一些现有的开销信息（头部信息）。

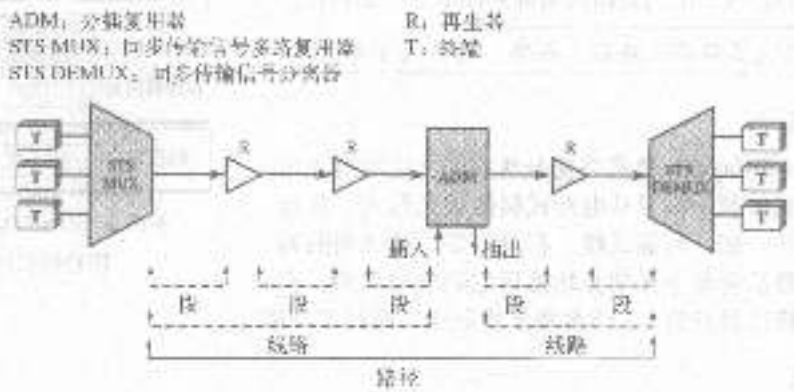


图17.1 使用SONET设备的简单网络

分插复用器

分插复用器允许信号的插入和抽取。分插复用器（add/drop multiplexer，ADM）可以把来自多个源的STS加到给定路径中或是从某条路径中除去并重定向所需的信号，而无需多路分解整个信号。分插复用器使用诸如地址和指针（会在后面描述）的头部信息来确定单个流，而不是依赖时间和位置。

在图17.1所示的简单配置中，多个进入的电子信号输入一个STS多路复用器中，这些信号在多路复用器里组合成一个单一光信号。光信号传输给再生器，在这里重新生成不带有传输过程中加入的噪声的信号。从多个源再生的信号输入到一个分插复用器。分插复用器重新组织这些信号，如果需要，用数据帧中的信息将它们直接发送出去。这些考次多路复用的信号发送给另一个再生器，然后再发送给接收STS分离器，在这里以接收链路可以接受的格式返回。

终端

终端（terminal）是使用SONET网络服务的设备。例如，在因特网中，终端可以是一个路由器，该路由器需要将发送分组给SONET网络另一端的另一个路由器上。

17.1.3 连接

前一节定义的设备使用段、线路和路径进行连接。

段

段（section）是连接两个相邻设备（多路复用器到多路复用器、多路复用器到再生器，或再生器到再生器）的光链路。

线路

线路（line）是两个多路复用器之间（STS多路复用器到分插复用器、两个分插复用器、或两个STS多路复用器）的网络部分。

路径

路径（path）是两个STS多路复用器之间的端到端网络部分。在两个STS多路复用器直接互连的简单SONET中，段、线路和路径相同。

17.2 SONET 层

SONET标准包括四个功能层：光子层、段层、线路层和路径层。它们对应于物理层和数据链路层（见图17.2）。各层中加到帧的头部会在本章后面讨论。

SONET定义了四层：路径、线路、段和光子层。



图17.2　SONET层与OSI或因特网层比较

17.2.1 路径层

路径层（path layer）负责将信号从它的光信源到光信宿的移动。在光信源，信号从电形式转换成光形式，再与其他信号复用在一起，封装成帧。在光信宿，接收到的帧被多路分解，然后将单个光信号转换回它们的电形式。在这一层增加了路径层开销。STS多路复用器提供路径层功能。

17.2.2 线路层

线路层（line layer）负责信号在物理线路中的移动。这一层给帧增加线路层开销。STS多路复用器和分插复用器提供线路层功能。

17.2.3 段层

段层（section layer）负责信号在物理段中的移动。它处理成帧、申扰和差错控制。在这一层给帧增加段层开销。

17.2.4 光子层

光子层（photonic layer）对应于OSI模型中的物理层。它包括光纤通道、接收器敏感度、多路复用功能等等的物理规范说明。SONET使用NRZ进行编码，光存在为1，不存在为0。

17.2.5 设备-层之间的关系

图17.3显示了用于SONET传输的设备和四层标准之间的关系。正如看到的，STS多路复用

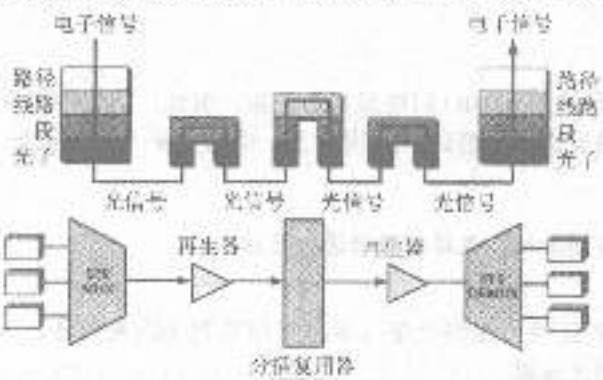


图17.3　SONET中的设备-层关系

器是四层设备。分插复用器是三层设备，再生器是两层设备。

17.3 SONET帧

每个同步传输信号STS-n由8 000个帧组成。每个帧是个$90 \times n$列9行的两维矩阵。例如，STS-1帧有90列、9行（810字节），STS-3有270列、9行（2430字节）。图17.4显示STS-1和STS-n的一般格式。

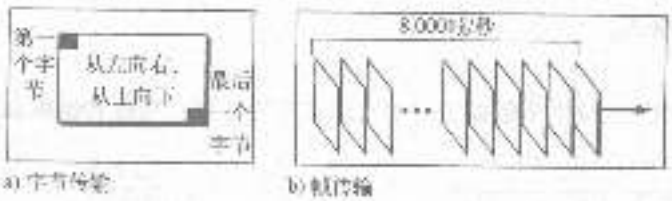


a) STS-1帧　　b) STS-n帧

图17.4 STS-1和STS-n帧

17.3.1 帧、字节和位传输

SONET的一个有趣的地方是每个STS-n信号以每秒8 000个帧的固定速率传输。这是数字化语音的速率（见第4章）。对于每个帧，从左向右、从上到下传输字节。对于每个字节，从高位向低位（从左向右）传输位。图17.5显示了帧传输和字节传输的顺序。



a) 字节传输　　b) 帧传输

图17.5 传输中的STS-1帧

> SONET STS-n信号以每秒8 000个帧传输。

如果我们对语音信号进行采样，每个样本8位（1个字节），我们可以说SONET帧中的每个字节可以携带来自数字化语音通道的信息。换言之，STS-1信号能同时携带774个语音通道（810减去用于开销所需的字节）。

> SONET帧中的每一个字节能承载一个数字化语音通道。

例17.1

求STS-1信号的数据速率。

解：

STS-1像其他STS信号一样每秒发送8 000个帧。每个STS-1帧由9乘（1×90）个字节组成。每个字节由8位组成。数据速率是：

STS-1数据速率 $= 8\,000 \times 9 \times (1 \times 90) \times 8 = 51.849$Mbps

例17.2

求STS-3信号的数据速率。

解：

STS-1像其他STS信号一样每秒发送8 000个帧。每个STS-1帧由9乘（3×90）个字节组成。

每个字节由8位组成。数据速率是：

STS-3数据速率＝8 000×9×(3×90)×8=155.52Mbps

注意：在SONET中，不同STS信号之间的数据速率是有成切关系的，我们可以由STS-1的数据速率来得STS-3的数据速率（前者乘以3）。

> 在SONET中，STS-n信号的数据速率是STS-1信号数据速率的n倍。

例17.3

STS-1帧的周期是多少？STS-3帧呢？STS-n帧呢？

解：

在SONET中，每秒发送8 000个帧。这表示STS-1、STS-3或STS-n帧的周期相同，等于1/8 000s或125us。

17.3.2 STS-1帧格式

STS-1帧的基本格式如图17.6所示。正如我们前面所讲到的，SONET帧是9行，每行90个字节（8位字节），一共810个字节的矩阵。

该帧的前三列用于段和线路开销。前三列的上面三行用于段开销（section overhead，SOH）。下面六行用于线路开销（line overhead，LOH）。该帧的其他字节成为同步有效载荷封装（SPE），它包括用户数据和用户数据级别所需的路径开销（path overhead，POH）。我们很快就会介绍SPE的格式。

段开销

每个段开销由9个八位字节组成，这些八位字节的标签、功能和组织如图17.7所示。

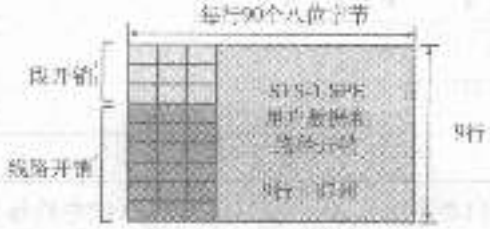


图17.6　STS-1帧开销

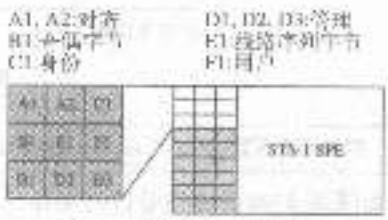


图17.7　STS-1帧：段开销

- **对齐字节（A1和A2）**。字节A1和A2用于实现成帧和同步，称为对齐字节。这些字节向接收器告警有一个帧即将到达，并给接收器一个预先确定的用于同步的位模式。这两个字节的位模式的十六进制为0xF628。这些字节充当标记。
- **奇偶校验字节（B1）**。字节B1用于位交织奇偶（BIP-8）校验。它的值由前一个帧所有字节计算而得。换言之，这个字节的第i位是前一个STS-n帧的所有第i位计算而得的奇偶位。这个字节的值只放在STS-n帧中的第一个STS-1中。换言之，虽然一个STS-n帧有n个B1字节（正如我们会在后面看到的），但是只有第一个字节有这个值，其余用0填充。
- **标识字节（C1）**。字节C1携带了STS-1帧的标识。当多个STS-1帧复用生成更高速率STS帧（STS-3，STS-9，STS-12等）的时候需要这个字节。这个字节中的信息允许在分解时能轻易辨认出各个信号。例如，在一个STS-3信号中，对应第一个STS-1的C1字节的值是1，第二个是2，第三个是3。
- **管理字节（D1、D2、和D3）**。D1、D2和D3字节一起形成一条称为数据通信通道的192kbps的通道（3×8 000×8）。这个通道用于操作、管理、和维护（OA&M）信令。

- **线路序列字节（E1）**。字节E1是线路序列字节，在连续帧中的E1字节形成了一条64kbps通道（每秒的8 000个帧乘以每个帧的8位），这个通道用于再生器之间，或终端和再生器之间的通信。
- **用户字节（F1）**。在连续帧中的F1字节形成了一条64kbps的通道，保留用于在段层的用户需求。

为每个SONET设备（再生器和多路复用器）重新计算段开销。

线路开销

线路开销由18个字节组成，这些字节的标签、功能和安排如图17.8所示。

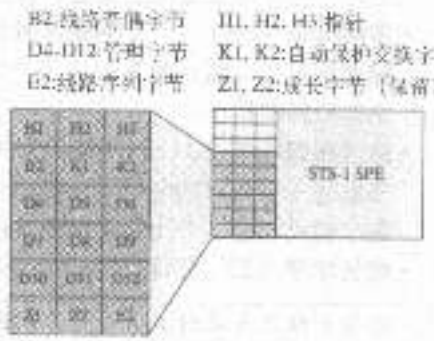


图17.8 STS-1帧：线路开销

- **线路奇偶字节（B2）**。字节B2用于位交织奇偶校验，它用于线路上（两个多路复用器间）帧的差错检测。在一个STS-*n*帧中，为前一个STS-1帧中的所有字节计算B2，并插入到那个帧的B2字节中。换言之，在一个STS-3帧中，有三个B2字节，每个都是为一个STS-1帧计算的。这个字节与段开销中的B1字节相对应。
- **数据通信通道字节（D4到D12）**。连续帧中的线路开销D字节（D4到D12）形成了一条576kbps的通道，该通道提供与D1～D3字节（OA&M）一样的服务，但是在线路级别而不是在段级别（两个多路复用器之间）。
- **线路序列字节（E2）**。连续帧中的E2字节形成了一条64kbps的通道，该通道提供与E1线路序列字节一样的功能，只是E2在线路级别。
- **指针字节（H1、H2和H3）**。字节H1、H2和H3是指针。前两个字节用于说明帧中SPE的偏移量，第三个字节用于证明。我们会在后面说明这三个字节的用法。
- **自动保护交换字节（K1和K2）**。连续帧中的K1和K2字节形成了一条128kbps的通道，用于在线路终止设备中的问题自动检测。我们会在本章后面讨论自动保护交换（APS）。
- **成长字节（Z1和Z2）**。Z1和Z2字节保留用于将来的使用。



图17.9 STS-1帧：路径开销

同步有效载荷封装

同步有效载荷封装（synchronous payload envelope，SPE）包含用户数据和与用户数据相关的开销（路径开销）。并不需要将一个SPE恰好放入一个STS-1帧中，我们很快会看到它可以分成两个帧。这意味着路径开销（SPE的最左列）不需要与段开销或线路开销对齐。必须先把路径开销加到用户数据中生成SPE，然后将SPE插入到一个或两个帧中。路径开销由9个字节组成。这些字节的标签、功能和安排如图17.9所示。

- 路径奇偶字节（B3）。字节B3用于位交织奇偶校验。像B1和B2一样，但是对SPE位进行计算。它实际上是对流中前一个SPE进行计算。
- 路径信号标签字节（C2）。字节C2是路径标识字节。它用于确认SPE中所携带数据在更高级（诸如IP或ATM）所使用的不同协议。
- 路径用户通道字节（F2）。连续帧中的F2字节，像F1字节一样，形成一条64kbps的通道，它保留用于用户需求，但在路径级别。
- 路径状态字节（G1）。字节G1由接收方发送给发送方来告知它的状态。当通信是多路复用时，它在保留通道上发送。我们会在本章后面看到它用于线性或环网络。
- 多帧指示（H4）。字节H4是多帧指示。它说明无法放入单个帧的有效载荷。例如，虚拟辅助指示可以组合起来形成一个帧，它比SPE帧更大，需要分成几个帧。会在下一节讨论虚拟辅助指示。
- 路径跟踪字节（J1）。连续帧中的J1字节形成一条用于跟踪路径的64kbps的通道。J1字节发送一个连续的64字节字符串来验证连接。字符串的选择留给应用程序。接收方会将每个模式与前一个比较来保证路径层上的通信没有错误。
- 成长字节（Z3、Z4和Z5）。字节Z3、Z4和Z5保留用于将来的使用。

> 路径开销只为端到端（在STS多路复用器之间）计算。

17.3.3　开销总结

表17.2比较并总结了段、线路和路径中使用的开销。

表17.2　SONET/SDH速率

字节功能	段	线　路	路　径
对齐	A1, A2		
奇偶	B1	B2	B3
标识符	C1		C2
DA&M	D1-D3	D4-D12	
线路序列	E1		
用户	F1		F2
状态			G1
指针		H1-H3	H4
跟踪			J1
容错		K1, K2	
成长（保留用于将来使用）		Z1, Z2	Z3-Z5

例17.4

STS-1帧的用户数据速率是多少（不考虑开销）？

解：

一个STS-1帧中的用户数据由9行86列组成，因此

STS-1用户数据速率 = 8 000 × 9 × (1 × 86) × 8 = 49.536Mbps

17.3.4　封装

前面的讨论说明了SPE需要封装在STS-1帧中，封装会产生两个问题，但SONET可以使用指针（H1到H3）很好地处理它们。我们将在本节讨论这些字节的使用。

偏移

SONET允许一个SPE扩展到两个帧，SPE的一部分在第一个帧中，另一部分在第二个帧中。当要封装的一个SPE与经过的同步帧在时间上不对齐时可能会发生这种情况。图17.10显示了这种情况。SPE字节分在两个帧中。字节的第一组封装在第一个帧中，第二组封装在第二个帧中。图中还显示了路径开销，它与任一个帧中的段/线路开销对齐。问题是SONET多路复用器如何知道在帧中哪里是SPE的开始？哪里是结束？解决方法是使用指针H1和H2来定义SPE的开始，因为每个SPE有固定个字节数所以结束可以确定。SONET允许SPE相对于STS-1帧的偏移。

为了找到帧中每个SPE的开始，我们需要线路开销中的两个指针H1和H2。注意这些指针位于线路开销，因为封装发生在多路复用器中。图17.11显示这两个字节如何指向SPE的开始。注意我们需要2个字节来定义在帧中一个字节的位置，一个帧有810个字节，因此不能用1个字节来定义。

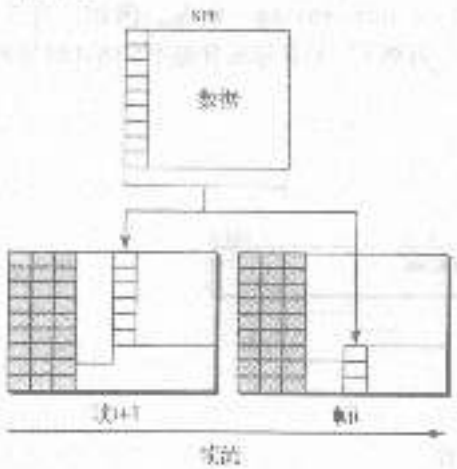


图17.10 SPE相对于帧边界的偏移

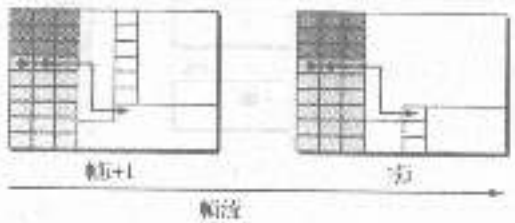


图17.11 使用H1和H2指针来指示SPE在帧中的开始

例17.5

如果SPE开始于第650个字节，那么H1和H2的值是多少？

解：

数字650可以表示成4个十六进制数字0x028A。这表示H1的值是0x02，H2的值是0x8A。

证明

现在假定有效载荷的传输速率与SONET的传输速率有很小的不同。首先，假设有效载荷的速率较高。这表示有时有1个额外的字节无法放入一个帧中。这种情况下，SONET允许这个额外的字节插入到H3字节中。现在，假设有效载荷的速率较低。这表示有时帧中需要留空1个字节。SONET允许这个字节放在H3字节的后面。

17.4 STS多路复用

在SONET中，较低速率的帧会同步地时分多路复用到一个更高速率的帧中，例如，三个STS-1信号（通道）可以组合成一个STS-3信号（通道），四个STS-3信号可以多路复用成一个STS-12信号，等等，如图17.12所示。

多路复用是同步TDM，网络中的所有时钟都锁定一个主时钟来实现同步。

> 在SONET中，网络中的所有时钟都锁定一个主时钟。



图17.12　STS多路复用/多路分解

我们要提醒多路复用也可以发生在更高的数据速率。例如，四个STS-3信号可以多路复用成一个STS-12信号。但是，STS-3信号需要先分离成12个STS-1信号，然后这十二个信号再多路复用成一个STS-12信号。这个额外工作的原因会在我们讨论字节交织后明了。

17.4.1　字节交替

在SONET中同步TDM多路复用通过字节交替（byte interleaving）实现。例如，当三个STS-1信号多路复用成一个STS-3信号时，STS-3信号中每组3个字节与来自每个STS-1信号的1个字节相关。图17.13显示了交替。

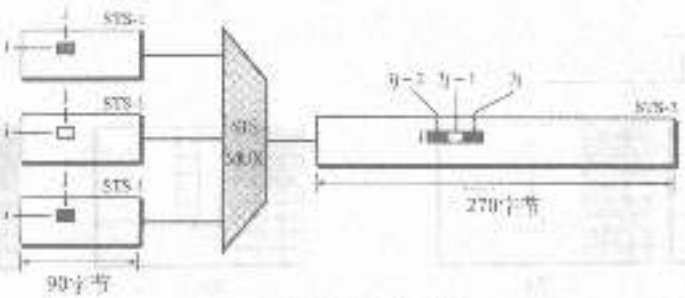


图17.13　字节交替

注意，STS-1帧的字节保持它的行位置，但是它移到不同的列。原因是所有信号帧都有相同的行数（9），而列数改变了。STS-n信号帧中的列数是STS-1帧中列数的n倍。因此，STS-n帧的一行能够适应STS-1帧的n行。

如图17.14所示，字节交替还保留了相应的段开销和线路开销。来自三个STS-1帧的段开销交织在一起生成STS-1帧的段开销。线路开销同理。但是，每个通道保留用来控制那个通道的相应字节。换言之，段和线路为每个多路复用通道保留它们自己的控制字节。这个有趣的特性允许分插复用器的使用，这已简单讨论过，如图所示，有三个A1字节，每个A1字节属于三个被复用信号的每一个。同样有三个A2字节，三个A3字节。

这里的多路复用比第6章描述过的统计TDM容易，因为分离器除去第一个A1并把它分给第一个STS-1、除去第二个A1并把它分给第二个STS-1、除去第三个A1并把它分给第三个STS-1，与字节的函数无关。换言之，分离器仅仅处理字节的位置，而不是它们的函数。

我们已经讲过的段开销和线路开销并不适用于路径开销。这是因为路径开销是SPE的一部分，SPE可能已经分到两个STS-1帧了。但是，字节交替与SPE的数据部分相同。

字节交替过程使得可以更高的数据速率多路复用，但也更加复杂。我们如何把四个STS-3信号多路复用成一个STS-12信号呢？这可以通过两步做到：首先，必须先分解STS-3信号生成12个STS-1信号。然后这12个STS-1信号多路复用成一个STS-12信号。

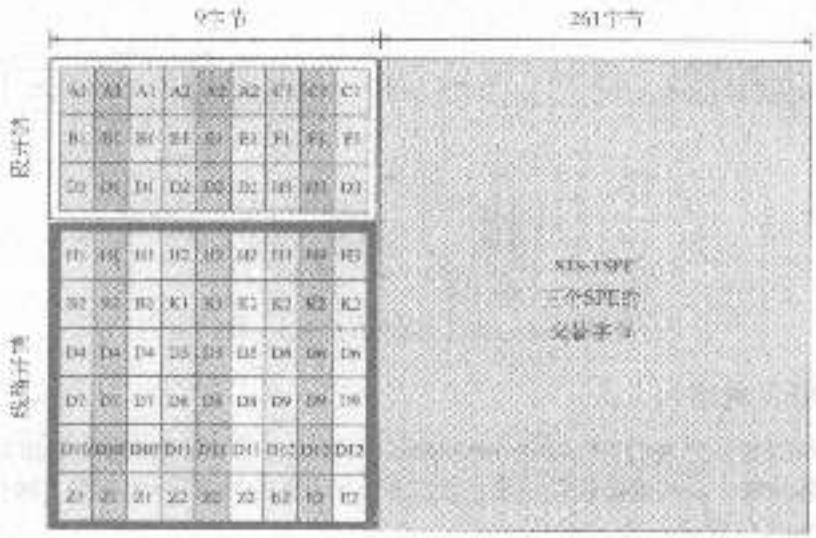


图17.14 STS-3帧

17.4.2 重组信号

在SONET的正常工作模式中，STS-n信号由n个多路复用的STS-1信号组成。有时，我们有比STS-1数据速率高的信号。这种情况下，SONET允许生成一个STS-n信号，它不被看做是n个STS-1信号，它是不可以分解成n个STS-1信号的一个STS-n信号（通道）。为了说明如何分解这个信号，后缀c（用于说明重组）加到信号名后面。例如，STS-3c是不能分解成三个STS-1信号的信号。但是，我们需要知道一个STS-3c信号中的整个有效载荷是一个SPE，这意味着我们只有一列路径开销（9字节）。这种情况使用的数据占用260列，如图17.15所示。

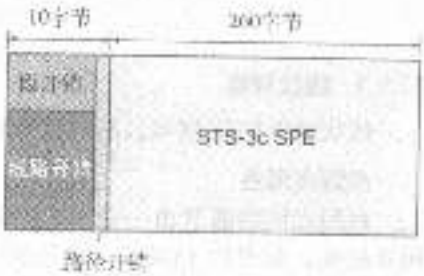


图17.15 重组STS-3c信号

携带ATM信元的重组信号

我们将在第18章中讨论ATM和ATM信元。ATM网络是信元网络，在ATM网络中每个信元有固定长度53个字节。一个STS-3c信号的SPE可以是ATM信元的承载者。一个STS-3c信号的SPE能携带$9 \times 260 = 2\,340$个字节，这大约于44个ATM信元（53个字节）。

> 一个STS-3c信号可以携带44个ATM信元作为它的SPE。

17.4.3 分插复用器

将多个STS-1信号多路复用成一个STS-n信号，是在STS多路复用器中完成（在路径层）。STS-n信号分离成STS-1信号，是在STS分离器中完成。但是，在两者之间，SONET使用分插复用器，它可以用一个信号替换另一个信号。我们需要知道这不是传统意义上的分离/复用。分插复用器工作在线路层。分插复用器不生成段、线路或路径开销。它几乎作为交换机，它除去一个STS-1信号，加上另一个STS-1信号。分插复用器的输入和输出的信号类型相同（比如两个STS-3或两个STS-12），分插复用器（ADM）只除去相应的字节，然后用新的字节替换它们（包括段开销和线路开销中的字节）。图17.16显示ADM的工作机制。

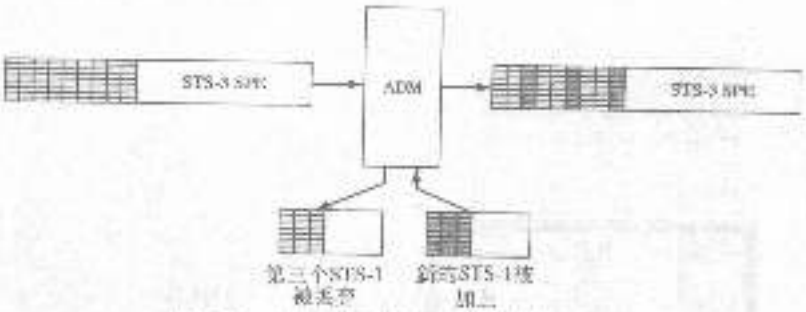


图17.16　在分插复用器中丢弃并加上STS-1帧

17.5　SONET 网络

使用SONET设备，我们可以建立一个SONET网络，它可以用做承载来自诸如ATM（第18章）或IP（第20章）负载的高速骨干网。我们可以大致把SONET分为三类：线状、环状和网状网络，如图17.17所示。

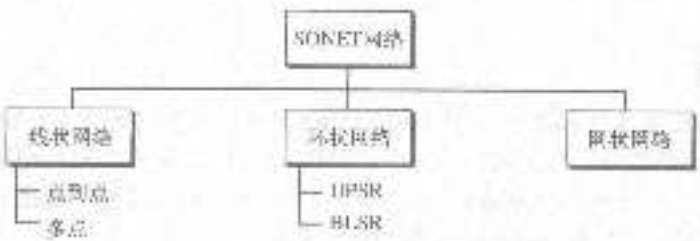


图17.17　SONET网络分类

17.5.1　线状网络

线状SONET网络可以是点到点的，也可以是多点的。

点到点网络

点到点网络通常由一个STS多路复用器、一个STS分离器、0个或更多再生器、无分插复用器组成，如图17.18所示。信号流可以是单向的也可以是双向的，虽然图17.18为了简单只显示了单向。

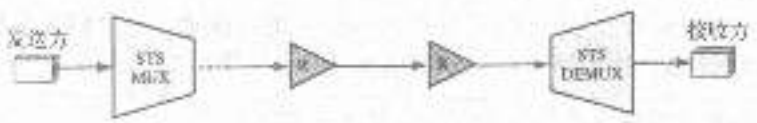


图17.18　点到点SONET网络

多点网络

多点网络使用ADM允许多个终端间的通信。ADM除去属于还到它的终端的信号，并加上来自另一个终端的信号。每个终端可以发送数据给一个或多个下游终端。图17.19显示了一个单向方案，在这个方案中每个终端可以只发送数据给下游终端，但多点网络也可以是双向的。

在图17.19中，T1同时发送数据给T2和T3，但是T2只能发送数据给T3。该图显示了一个简单的配置，在正常情况下我们可以有多个ADM以及更多的终端。

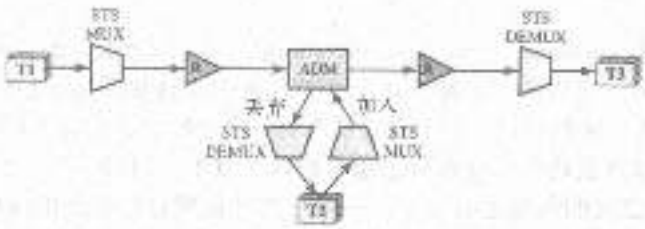


图17.19 多点SONET网络

自动保护交换

为了建立对线状网络中故障的保护，SONET定义了**自动保护交换**（automatic protection switching，APS）。线状网络中的APS定义在线路层，这意味着保护是在两个ADM或一对STS多路复用器/分离器之间。这个主意提供了冗余性。在发生故障的情况下可以使用一条冗余线路（光纤）。主线路称为工作线路，冗余线路称为保护线路。在线状通道中通常有三种保护方案：一加一、一到一和一到多。图17.20显示这三种方案。

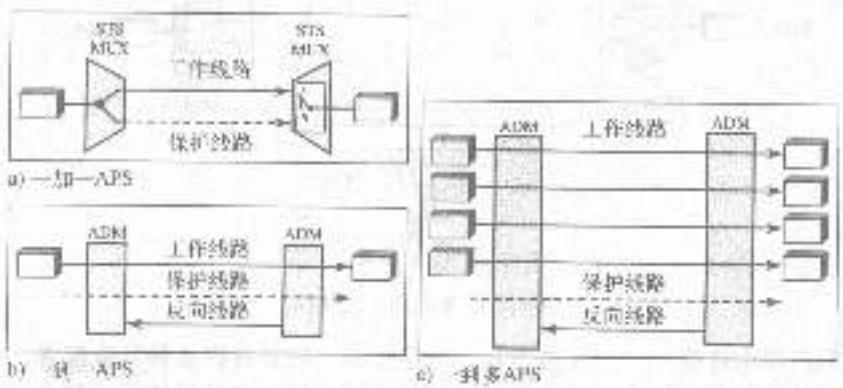


图17.20 线状网络中自动保护交换

一加一APS 在这个方案中，通常有两条线路：一条工作线路和一条保护线路。两条线路总是活跃的。发送方多路复用器在两条线路上都发送相同的数据，接收方多路复用器监控线路并选择质量更好的那条线路。如果其中一条线路发生故障了，它丢失了信号，当然在接收方会选择另一条线路。虽然这种方案中会立即恢复故障，但是因为需要两倍带宽，所以这个方案的效率不高。注意一加一交换在路径层实现。

一到一APS 在这个方案中，看起来像一加一方案，也有一条工作线路和一条保护线路。但是，数据通常只在工作线路上发送直到它发生故障。这时，接收方使用反向线路通知发送方使用保护线路。很明显，故障恢复比一加一方案慢，但是这个方案效率更高，因为保护线路只有在工作线路发生故障时才会用来传输数据。注意一到一交换在线路层实现。

一到多APS 这个方案与一到一方案相似，不同的是对于许多条工作线路，它只有一条保护线路。当其中一条工作线路发生故障时，保护线路就会接管控制直到发生故障的线路被修复。它没有一到一方案安全，因为如果多于一条工作线路同时发生故障，保护线路只能替换其中一条。注意一到多APS在线路层实现。

17.5.2 环状网络

ADM使得SONET环状网络成为可能。SONET环可以用于单向或双向配置。在每种情况下，我们可以增加一个额外环使得网络能自我修复线路故障。

单向路径交换环

单向路径交换环（unidirectional path switching ring，UPSR）是有两个环的单网络：一个环用做工作环而另一个用做保护环。这个主意与我们在线状网络中讨论过的一加一APS类似。相同的信号从两个环都经过，一个顺时针而另一个逆时针。它称为UPSR，是因为监控在路径层实现。一个节点在路径层接收到电信号的两个复本，比较它们，并选择质量更好的一个。如果两个ADM之间的环发生故障，另一个环仍然能保证数据流的继续。UPSR像一加一方案一样有快速的故障恢复，但是因为需要两个环来完成工作所以效率不高，一半的带宽被浪费掉了。图17.21显示了一个UPSR网络。

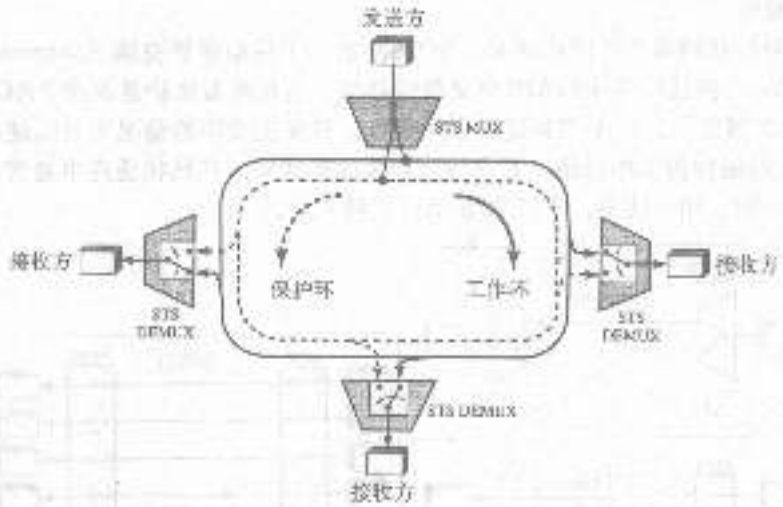


图17.21　单向路径交换网

虽然我们在图中选择了一个发送方和二个接收方，但是有许多种其他配置。发送方使用两路连接来同时发送数据给两个环，接收方使用选择交换机来选择信号质量更好的环。我们使用一个STS多路复用器和三个STS分离器来强调工作在路径层的节点。

双向线路交换环

SONET环状网络的另一个方案是**双向线路交换环**（bidirectional line switching ring，BLSR）。这种情况下，通信是双向的，这意味着我们需要两个环来作为工作线路。我们还需要两个环作为保护线路。这意味着BLSR使用四个环。但是工作机制与一到一APS方案类似。如果两个节点间一个方向的工作环发生故障，接收方节点可以使用反向环通知故障方向的上游节点使用保护环。网络能恢复许多种不同的故障，我们不在这讨论了。注意BLSR中故障恢复是在线路层而不是在路径层。ADM发现故障并通知邻近节点使用保护环。图17.22显示了一个BLSR环。

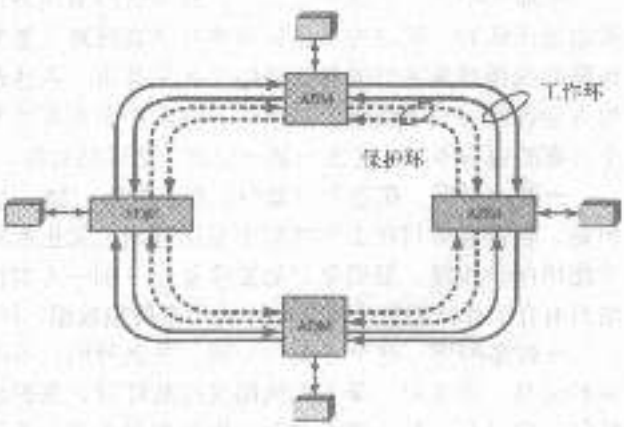


图17.22　单向线路交换环

环的组合

今天SONET网络使用互连环的组合来提供广大区域的服务。例如，SONET网络可以有一个区域环、多个本地环和许多站点环来为广大区域提供服务。这些环可以是UPSR、BLSR、或者两者的组合。图17.23显示了这样一个广域环网络的概念。

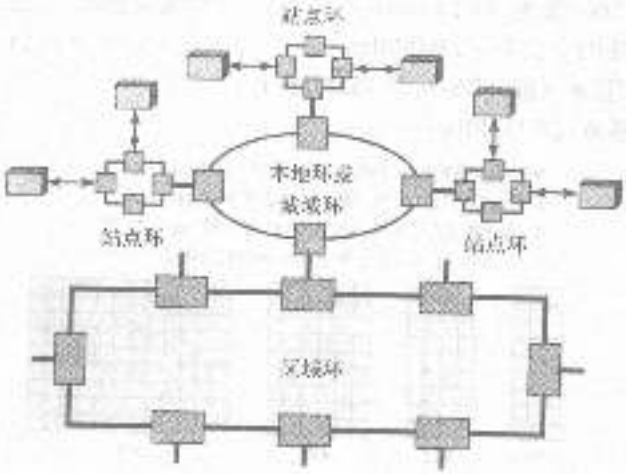


图17.23 SONET网络中环的组合

17.5.3 网状网络

环网络的一个问题是缺乏扩展性，当环中的流量上升时，我们不仅需要升级线路也需要升级ADM。这种情况下，有多个交换机的网状网络有可能提供更好的性能。网状网络中的交换机称为交叉连接。交叉连接像我们看到的其他交换机一样，有输入端口和输出端口。在输入端口，交换机接到一个OC-*n*信号，把它转换成一个STS-*n*信号，把它分解成相应的STS-1信号，并把每个STS-1信号发送给正确的输出端口。输出端口接收到来自不同输入端口的STS-1信号，把它们多路复用成一个STS-*n*信号，并生成一个OC-*n*信号传输。图17.24显示了一个网状SONET网络以及交换机的结构。

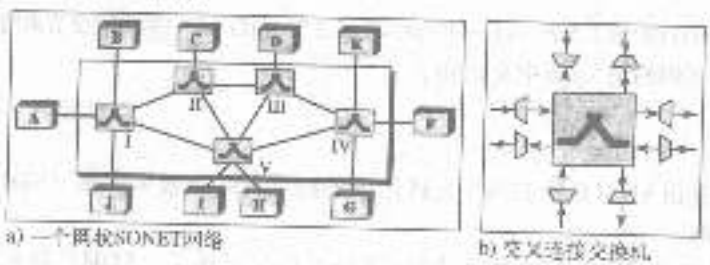


图17.24 一个网状SONET网络

17.6 虚拟支路

SONET设计用来承载宽带载荷，现在的数字体系的数据速率（DS-1到DS-3）比STS-1低。为了使SONET向后兼容现在的体系，它的帧设计包括了虚拟支路（virtual tributaries，VT）系统（见图17.25）。虚拟支路是能插入到STS-1的部分载荷，并与其他部分载荷组合装入一个帧。对于来自同一个源的数据，我们把SPE分割开来并称每一部分为VT，而不是使用STS-1帧的所有86个有效载荷列。

VT类型

已经定义了四种VT类型来适应现在的数字体系（见图17.26），注意允许每种VT的列数由倍乘类型标识号确定（VT1.5得到3列，VT2得到4列等等）。

- VT1.5适应U.S.DS-1服务（1.544Mbps）
- VT2适应欧洲CEPT-1服务（2.048Mbps）
- VT3适应DS-1C服务（部分DS-1，3.152Mbps）
- VT6适应DS-2服务（6.312Mbps）

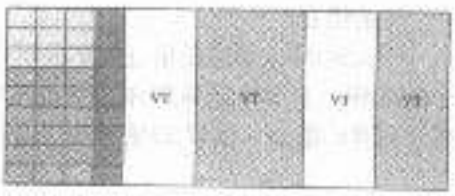


图17.25　虚拟支路

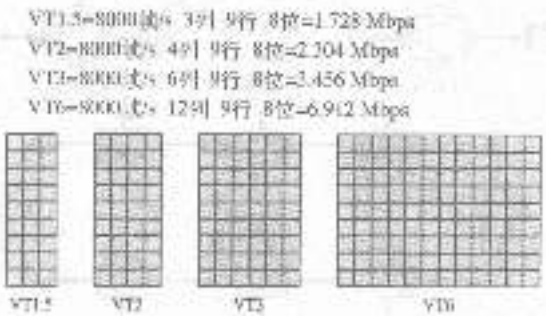


图17.26　虚拟支路类型

当两个或更多支路被插入到单个STS-1帧中时，它们逐列交织。SONET提供了确定每个VT并把它们分开而无需分解整个流的机制。这些机制的讨论以及后面的控制问题超出了本书的范围。

推荐读物

为了深入讨论与本章有关的主题，我们推荐下列读物。本书末的方括号[……]中给出了参考资料。

读物

SONET在[Tan03]的第2.5节、[Kes97]的第15.2节、[GW04]的第4.2节和第4.3节、[Sta04]的第8.2节、和[WV00]的第5.2节中有讨论。

本章小结

- 同步光网络是由ANSI开发的用于光纤网络的标准；同步数字体系（SDH）是由ITU-T开发的类似标准。
- SONET已经定义了称为同步传输信号（STS）的信号体系。SDH已经定义了称为同步传输模块（STM）的类似的信号体系。
- OC-n信号是STS-n（或STM-n）信号的光调制。
- SONET定义了四层：路径、线路、段和光子。
- SONET是同步TDM系统，所有时钟都锁定一个主时钟。
- SONET系统能使用以下设备：

1. STS多路复用器
2. STS分离器
3. 再生器

4. 分插复用器
5. 终端

- SONET每秒发送8 000帧，每个帧持续125us。
- STS-1帧由9行、90列组成；STS-n帧由9行、$n \times 90$列组成。
- STS能多路复用为到一个更高数据速率的新STS。
- SONET网络拓扑可以是线状的、环状的、或是网状的。
- 线状SONET网络可以是点到点或是多点的。
- 环状SONET网络可以是单向的或是双向的。
- 为了让SONET向后兼容现在的体系，它的帧设计包括虚拟支路（VT）系统。

习题

复习题

1. SONET和SDH之间的关系是什么？
2. STS和STM之间的关系是什么？
3. STS多路复用器与分插复用器有什么不同，既然两者都能把信号加在一起。
4. STS信号和OC信号之间的关系是什么？
5. 线路开销中的指针的作用是什么？
6. 为什么SONET称为同步网络？
7. SONET再生器的功能是什么？
8. SONET四层是什么？
9. 讨论SONET每层的功能。
10. 虚拟支路是什么？

练习题

11. STS-3、STS-9和STS-12的用户数据速率是多少？
12. 说明STS-9如何多路复用生成STS-36。这种类型的多路复用涉及到额外开销吗？
13. 数据流由STS-1帧携带。如果流的数据速率是49.540Mbps，那么每秒多少个STS-1帧必须让它们的H3字节携带数据？
14. 数据流由STS-1帧携带。如果流的数据速率是49.530Mbps，那么每秒多少个帧必须在H3字节后留一个空字节？
15. 表17.2显示开销字节可以分类成A、B、C、D、E、F、G、H、J、K和Z字节。
    a. 在LOH或POH中为什么没有A字节？
    b. 在LOH中为什么没有C字节？
    c. 在POH中为什么没有D字节？
    d. 在LOH或POH中为什么没有E字节？
    e. 在LOH或POH中为什么没有F字节？
    f. 在SOH或LOH中为什么没有G字节？
    g. 在SOH中为什么没有H字节？
    h. 在SOH或LOH中为什么没有J字节？
    i. 在SOH或POH中为什么没有K字节？
    j. 在SOH中为什么没有Z字节？
16. 为什么在所有三个头部中都存在B字节？

第18章 虚电路网络：帧中继和ATM

在第8章中，我们已讨论了交换技术。我们说过有三种交换技术：电路交换、分组交换和报文交换。我们也提到分组交换可以使用两种方法：虚电路方法和数据报方法。

在本章中，我们讨论虚电路方法如何用于广域网。两种常用的WAN技术使用虚电路交换。帧中继是一种相对高速协议，它能提供一些在诸如DSL、有线TV和T线路等的其他WAN技术中没有的服务。ATM作为一种高速协议，当它部署诸如SONET的物理层承载时，可以成为通信高速公路。

我们先讨论帧中继，然后更详细地讨论ATM。最后，我们说明最初设计作为WAN技术的ATM技术如何用于LAN技术，即ATM LAN。

18.1 帧中继

帧中继（Frame Relay）是一种虚电路广域网，设计用来满足在20世纪80年代末期和20世纪90年代早期对新型广域网的需求。

1. 在帧中继之前，一些组织使用一种称为X.25的虚电路交换网，它在网络层中进行交换。例如，因特网需要将自己的分组从一个地方传输到另一个地方，就曾使用X.25。X.25仍然在因特网中使用，但是正在被其他广域网取代。X.25有几个缺陷：

a. X.25具有64Kbps的低数据速率。到20世纪90年代，出现了对更高数据速率广域网的需求。

b. X.25在数据链路层和网络层都提供了广泛的流量和差错控制。这是因为X.25设计于20世纪70年代，那时的传输介质更容易出错。在这两个层中的流量和差错控制产生了大量的额外开销，并降低了传输率。对于在节点之间、源和目的地址之间传输的数据链路层的帧和网络层的分组，X.25都需要确认。

c. X.25最初设计用于个人应用，而不是因特网。X.25有自己的网络层。这意味着用户数据将被封装成X.25的网络层的分组。然而，因特网有自己的网络层，这就意味着，如果想要使用X.25，因特网必须将自己的称为数据报的网络层分组交付给X.25封装成X.25的分组。这是双重的额外开销。

2. 由于对X.25的失望，一些组织开始从公共服务提供商那里租用T-1或T-3线路组建自己的私有广域网。这种方法也有一些缺陷。

a. 如果一个组织有n个部门分布在一个地区，就需要$n(n-1)/2$条T-1或T-3线路，尽管可能使用这些线路只有10%的时间，但组织也必须为所有线路付费。这可能非常昂贵。

b. T-1和T-3线路提供的服务假定用户在所有时间都有固定的数据速率。例如，T-1线路是为那些以固定的1.544Mbps使用线路的用户而设计的。这种类型的服务并不适合现在的需要发送突发性数据（bursty data）的许多用户。例如，一个用户可能想要以6Mbps的速度发送数据2秒，7秒不发送，以3.44Mbps发送数据1秒，总共在10秒内发送15.44M比特。尽管平均速度仍然是1.544Mbps，但T-1线路不能接受这种类型的请求，因为它是为固定速率的数据而不是为突发性数据而设计的。突发性数据需要按需分配带宽（bandwidth on demand），用户在不同的时间里可能需要不同的带宽分配。

为解决以上缺陷，设计了帧中继。帧中继是广域网，它有以下特征：

1. 帧中继以较高的速率（1.544Mbps以及最近的44.376Mbps）工作，这就是说，它可以容易地代替网状的T-1或T-3线路。

2. 帧中继只工作在物理层和数据链路层。这就是说，它可容易地用作主干网，为已经有一个网络层协议的协议提供服务，例如因特网。

3. 帧中继允许突发性数据。

4. 帧中继允许的帧大小为9 000字节，这适合于所有的局域网帧。

5. 帧中继比其他传统的广域网花费少。

6. 帧中继仅在数据链路层有错误检测，没有流量或错误控制。当一个帧被破坏时，它甚至没有一个重发策略，该帧只是简单地被丢弃。使用这种方式设计帧中继为更可靠的介质和在更高层上有流量和差错控制的协议提供更快的传输能力。

18.1.1 结构

帧中继提供永久虚电路和交换虚电路。图18.1展示了一个连接到因特网的帧中继网络的例子。使用路由器（将在第22章中见到）来连接局域网和因特网中的广域网。在图中，帧中继广域网作为全球互特网的一个链路。

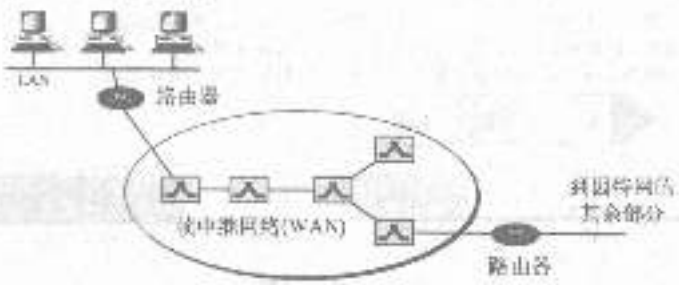


图18.1 帧中继网络

虚电路

帧中继是一种虚电路网络。帧中继的虚电路是用称为**数据链路连接标识符**（data link connection identifier，DLCI）的一个数字来定义的。

> 帧中继中的VCI称为DLCI。

永久虚电路和交换虚拟电路

源和目的地址可能选择拥有一条**永久虚电路**（permanent virtual circuit，PVC）。这种情况下，连接建立就简单了。相应的表条目由管理员为所有的交换机建立（当然是远程地建立并且是电子化地建立）。为源地址分配一个出的VCI，为目的地址分配一个入的VCI。

PVC连接有两个缺点。首先，花费大，因为即使没有使用，连接的两方也需要为这个连接一直付费。其次，只在一个源地址和一个目的地址之间建立连接。如果源地址需要和多个目的地址连接，那么每个连接都需要一个PVC，替代的方法是**交换虚电路**（switched virtual circuit，SVC）。SVC建立一个临时的、短的连接，该连接只存在于源地址和目的地址的数据传输过程中。像第8章讨论的一样，SVC需要一个连接建立阶段和连接终止阶段。

交换机

帧中继网络中的每个交换机都有一个用来路由帧的表。这个表将一个输入端口（DLCI组

合）与一个输出端口（DLCI组合）相匹配，如我们在一般虚电路网络中描述的一样。区别仅是VCI被DLCI代替。

18.1.2　帧中继层

图18.2显示了帧中继层。帧中继仅有物理层和数据链路层。

帧中继仅工作在物理层和数据链路层。



图18.2　帧中继层

物理层

帧中继中的物理层没有定义一个具体的协议。相反，它允许实现者使用可用的任何协议。帧中继支持任何可以被ANSI所识别的协议。

数据链路层

在数据链路层，帧中继使用一个简单协议，它不支持流量控制和差错控制。它只有一个差错检测机制。图18.3给出了帧中继的帧的格式。地址字段定义了DLCI和一些用于控制拥塞的位。

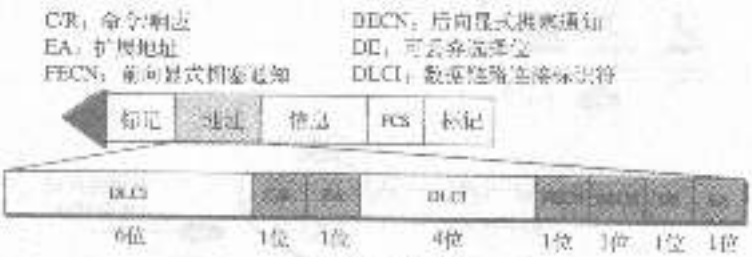


图18.3　帧中继帧

各个字段的描述如下：

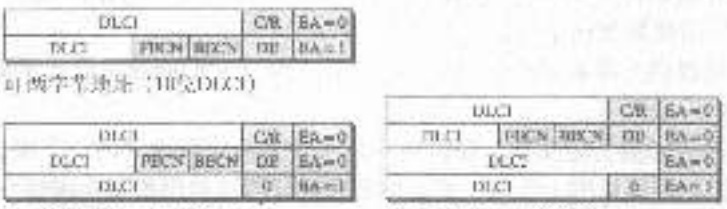
- **地址（DLCI）字段。**第一个字节的前6位构成DLCI的第一部分。DLCI的第二部分使用第二个字节的前4位。这些位是标准所定义的10位数据链路连接标识符的一部分。DLCI的功能前面已经讨论过。在本节的最后，我们将讨论扩展地址。
- **命令/响应（C/R）。**命令/响应（C/R）位允许上层识别帧是命令还是响应。帧中继协议不用这一位。
- **扩展地址（EA）。**扩展地址（EA）位表明当前字节是否是地址的最后一个字节。EA为0表示后面还跟着另外一个地址字节，EA为1则意味着当前字节是地址的最后一个字节。
- **前向显式拥塞通知（FECN）。**前向显式拥塞通知（forward explicit congestion notification，FECN）位可以由所经过路径中的任何一个交换机来设置，用来表示在帧传输方向上出现了拥塞。它通知目的站点发生了拥塞。用这种方式，目的站点知道可能将出现延迟或丢失分组。我们将在第24章讨论拥塞控制时讨论这一位的用法。
- **后向显式拥塞通知（BECN）。**后向显式拥塞通知（backward explicit congestion notification，BECN）位用于表示在帧传输相反的方向上出现了拥塞。它通知发送方发生了拥塞。用这种方式，发送方知道它应该放慢发送速度以防止丢失分组。我们将在第24章讨论拥塞控制时讨论这一位的用法。
- **可丢弃选择位（DE）。**可丢弃选择位（discard eligibility，DE）指明帧的优先级。在紧急情况下，交换机可能需要丢弃一些帧来缓和拥塞并防止网络由于过载而崩溃。当设置

DE为1时，这个位通知网络当发生拥塞时就丢弃该帧。该位可以由帧的发送方（用户）设置，也可以由网络中任何一个交换机设置。

帧中继不提供流量和差错控制，这些必须由上层协议提供。

18.1.3 扩展地址

为了增加DLCI的范围，帧中继地址从原先的2字节扩展到3字节或4字节。图18.4显示了不同的地址。注意，EA字段定义了字节数，在地址的最后一字节EA为1，在其他字节EA为0。注意，在3字节或4字节格式中，最后一位之前的位都设为0。



a) 两字节地址（10位DLCI）

b) 三字节地址（16位DLCI）

c) 四字节地址（23位DLCI）

图18.4 二种地址格式

18.1.4 FRAD

为了处理从其他协议到达的帧，帧中继使用一种称为**帧中继组装器/分解器**（Frame Relay assembler/disassembler，FRAD）的设备。FRAD对来自其他协议的帧进行组装和分解，使它们成为帧中继的帧。一个FRAD可以作为独立的设备或交换机的一部分来实现。图18.5显示了连接到帧中继网络的两个FRAD。



图18.5 FRAD

18.1.5 VOFR

帧中继网络提供一个称为**帧中继语音**（Voice Over Frame Relay，VOFR）的选项，它通过网络发送语音。语音用PCM数字化，然后压缩，其结果在网上作为数据帧发送。这个特性允许长距离廉价发送语音。然而请注意，语音的质量没有在电路交换网（如电话网）上好。另外，有时上面提到的可变延时也会破坏实时语音。

18.1.6 LMI

帧中继最初是为PVC连接而设计的，所以没有提供控制与管理接口。**本地管理信息**（Local management information，LMI）是为了提供更多的管理特性才在最近加到帧中继协议中的一个协议。LMI能特别提供：

- 一个保活机制以检测是否有数据在传输。
- 一个多播机制，允许一个本地终端系统向多个远程终端系统发送帧。
- 一个允许一个终端系统检测一个交换机状态（例如，查看交换机是否拥塞）的机制。

18.1.7 拥塞控制和服务质量

帧中继一个很好的特性是提供了**拥塞控制**（congestion control）和**服务质量**（quality of service，QoS）。我们还没有讨论这些特性。在第24章，我们将介绍网络的这两个重要方面，并且讨论在帧中继和其他网络中如何实现它们。

18.2 ATM

异步传输模式（Asynchronous Transfer Mode，ATM）是由ATM论坛设计的信元中继（cell relay）协议，并被ITU-T采纳。ATM和SONET的结合将允许世界上的网络之间高速互连。实际上，ATM可以被设想为信息超高速公路上的“高速公路”。

18.2.1 设计目标

在ATM设计者所面临的所有挑战中，有六个方面非常突出。

1. 首先需要有一个传输系统来优化高速传输介质特别是光纤的使用。除了提供大的带宽外，新的传输介质和设备受噪声的影响显著降低。因此需要一种新的技术来同时利用这两个方面的优点，从而使数据速率达到最大。

2. 系统必须提供与现有系统的接口，并且在不降低它们的效率或者不更换它们的情况下，在它们之间提供广域互连。

3. 设计必须能够廉价地实现，使价格不再成为采纳这种设计的一个障碍。如果ATM如预期的那样成为国际性通信的主干网，那么它必须以低价格提供给每个需要它的用户。

4. 新系统必须能够支持现有的电信体系（本地环路、本地提供商和长途电信公司等）并和它们协同工作。

5. 新系统必须是面向连接的，以确保准确和可预测的传输。

6. 最后但并非最不重要的是，新系统的一个目标是将尽可能多的功能转移到硬件上（为了提高速度），同时删除尽可能多的软件功能（同样为了提高速度）。

18.2.2 问题

在我们讨论这些设计需求的解决方法之前，分析现有系统存在的问题是很有必要的。

帧网络

在ATM之前，数据链路层的数据通信是基于帧交换和帧网络的。不同协议使用大小和复杂性不同的帧。当网络变得更加复杂时，必须在帧头部携带的信息变得更加广泛。结果是相对于数据单元而言，头部越来越大。相应地，一些协议已经增大了数据单元头部的大小，使头部的使用更有效率（使用相同大小的头部而传输更多的数据）。但是，大的数据字段导致浪费，如果没有很多的信息要传输，这个字段的大部分将未被使用。为了提高利用率，一些协议给用户提供了可变长的帧。

混合网络通信

可以想像，帧大小的变化导致通信量不可预测。交换机、多路复用器和路由器必须融合复杂的软件系统来管理不同大小的帧，必须阅读大量的帧头信息，并对每个位计数和赋值来确保每个帧的完整性。在最好的情况下不同的帧网络之间的互连是慢而昂贵的，在最坏的情况下是不能实现的。

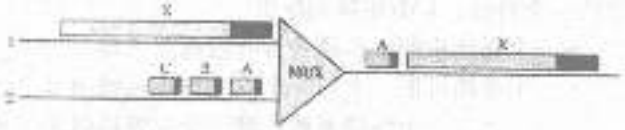


图18.6 使用不同帧大小的复用

另外一个问题是，在帧大小不可预测且变化很大的情况下，如何提供稳定速率的传输。为了从宽带技术中获得最大好处，通信量必须被时分复用到共享的通路上。想像一下，将来自两个有不同需求（和帧设计）的网络中的帧复用到同一条链路上（如图18-13）。会出现什么结果？当线路1使用大的帧（通常是数据帧），而线路2使用非常小的帧（音频和视频信息标准）时，将会发生什么情况呢？

如果线路1的巨大帧X只比线路2的帧早一点到达多路复用器，多路复用器也将先把帧X放到新路径上。毕竟，即使线路2的帧有更高的优先级，多路复用器也无法知道应该等待它们的到来，而是处理已经到达的帧。因此帧A必须等待整个X的位流进入路径后才能跟随进入。帧X的绝对大小导致了帧A的不正常延时。同样的不平衡可能影响从线路2来的所有帧。

因为语音和视频帧通常是很小的，将它们和传统的数据帧混合传输，往往导致这种类型帧的不可接受的延时，使得共享帧链路无法为语音和视频信息所使用。通信量必须经过不同的路径，这在很大程度上类似于汽车和火车的交通。但是，为了最大程度地利用宽带链路，我们需要在相同的链路上传输各种各样的通信量。

信元网络

和帧互连网络有关的许多问题可以通过采用称为信元网络（cell network）的概念来解决。一个信元是一个固定大小的小数据单元。在信元网络（cell network）中，使用信元（cell）作为数据交换的基本单位。所有的数据都装载入相同的信元中。这些信元可以按照完全可预测和统一的方式进行传输。当不同大小和格式的帧从分支网络到达信元网络时，它们被分割成相同大小的多个小数据单元，并装载入信元中。这些信元和其他信元多路复用并路由通过整个信元网络。由于每个信元大小相同并且都很小，因此避免了由于多路复用不同大小的帧所带来的问题。

> 信元网络使用信元作为数据交换的基本单位。信元定义为一个小的、固定大小的信息块。

图18.7显示了图18.6中的有两条链路的多路复用器发送信元而不是帧的情形。帧X被分割成三个信元：X、Y和Z。链路1中只有第一个信元在链路2的第一个信元前发送。两条链路的信元将交织在一起，这样没有一个信元必受漫长的延时。

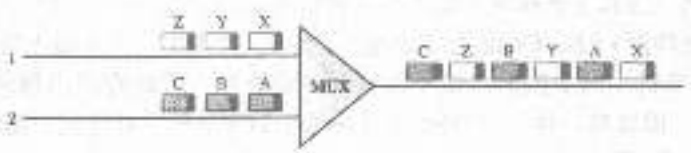


图18.7 使用信元的多路复用

这种情形的第二个优点是链路的高速率和小信元相结合，这意味着如果不考虑交织，从每条链路的信元将以近似连续流的方式到达各自的目的地（就像一部电影在大脑中看上去是连续的画面，而实际上是由一系列静止的图像所组成的）。在这种方法中，信元网络可以在双方完全没有意识到分段或复用的情况下，处理如电话呼叫等实时传输。

异步TDM

ATM使用异步时分复用处理来自不同通道的信元——这就是为什么称为异步传输模式的理由。它使用固定大小的时隙（一个信元的大小）。ATM复用器使用来自任何输入通道的一个信元填充一个时隙，如果通道没有发送的信元，则时隙为空。

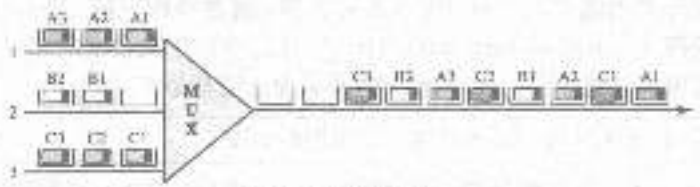


图18.8 ATM复用

图18.8显示了来自三个输入的信元是如何进行多路复用的。在第一个时钟节拍，通道2没有信元（空的输入时隙），因此多路复用器使用来自第三个通道的信元填充时隙。当所有通道中的所有时隙都被复用后，输出时隙为空。

18.2.3 结构

ATM是信元交换网络，用户访问设备，称为端点（endpoint），通过用户到网络接口（user-to-network interface，UNI）连接到网络内部的交换机上，交换机通过网络到网络接口（network-to-network interface，NNI）彼此连接。图18.9显示了一个ATM网络的例子。

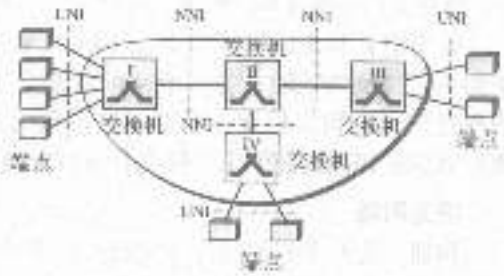


图18.9　ATM网络的体系结构

虚连接

两个端点之间的连接是通过传输路径（TP），虚路径（VP）和虚电路（VC）完成的。**传输路径**（transmission path，TP）是一个端点与一个交换机或者两个交换机之间的物理连接（电线、电缆、卫星等）。可将两个交换机看做两个城市，传输路径是直接连接两个城市的所有高速公路的集合。

一个传输路径划分成几个虚路径。一个**虚路径**（virtual path，VP）提供两个交换机之间的一条连接或连接的集合。可将一条虚路径看做连接两个城市的一条高速公路。每条高速公路是一个虚路径，所有高速公路的集合是传输路径。

信元网络基于**虚电路**（virtual circuit，VC）。属于同一报文的所有信元沿着同一条虚电路传输，同时保持它们的原始次序直到到达目的地。可将虚电路看做高速公路的车道。图18.17显示了传输路径（物理连接），虚路径（捆在一起的虚电路的集合，因为它们的部分路径是相同的）和虚电路（逻辑上连接两个端点）之间的关系。

为了更好地理解VP和VC的概念，请看图18.11。在图中，八个端点用四个VC通信。然而，前两个VC从交换机I到交换机II似乎共享同一虚路径，因此有理由捆绑这两个VC形成一条VP。另一方面，很显然，其余两个VC从交换机I到交换机IV共享同一虚路径，因此也有理由捆绑它们形成一条VP。

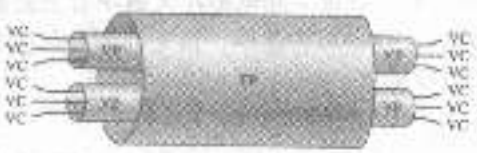


图18.10　TP，VP和VC

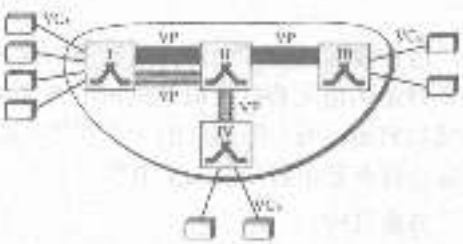


图18.11　VP和VC的例子

标识符　虚电路网络中，为了从一个端点路由数据到另一个端点，需要标识该虚连接。为此，ATM设计者创建了一个两级的层次标识符：**虚路径标识符**（virtual path identifier，VPI）和**虚电路标识符**（virtual circuit identifier，VCI）。VPI定义特定的VP，而VCI定义VP中特定的VC。VPI对捆绑（逻辑上）成一个VP的所有虚连接都是相同的。

> 注意：一个虚连接由一对数字定义：VPI和VCI。

图18.12显示了一条传输路径的VPI和VCI。当我们讨论ATM网络的路由时，将一个标识

特划分成两部分的基本原理就很清楚了。

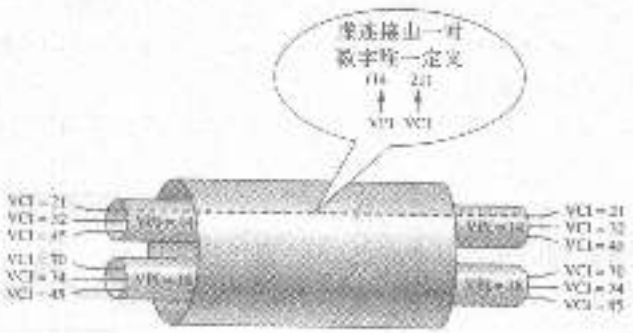


图18.12 连接标识符

在UNI和NNI中，VPI的长度是不同的。在UNI中，VPI的长度是8位，而在NNI中是12位。VCI的长度在两种接口中是相同的（都是16位）。因此，我们可以说，在UNI中，虚连接由24位确定，而在NNI中由28位确定（见图18.13）。

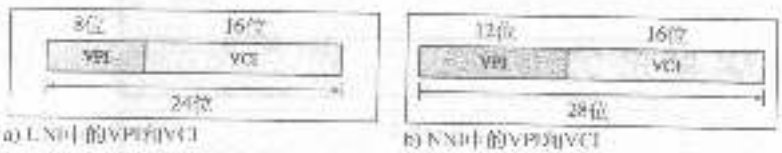


图18.13 UNI和NNI中的虚连接标识符

将虚连接标识符分割成两部分，其背后的整个思想是要允许层次路由。在一个典型的ATM网络中，大多数的交换机使用VPI来路由。而处于网络边界的那些直接和端点设备交互的交换机，使用VPI和VCI来路由。

信元

ATM网络中的基本数据单元称为信元。一个信元只有53字节长度，其中5个字节为头部，48个字节为有效载荷（用户数据可能少于48个字节）。我们将详细讨论信元的各个字段，但当前只能说，信元头部的大部分被VPI和VCI占用，它们定义了虚连接，信元通过虚连接可以从一个端点传送到一个交换机，或者从一个交换机传送到另一个交换机。图18.14显示了信元的结构。

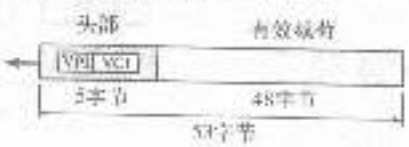


图18.14 ATM信元

连接建立与释放

像帧中继一样，ATM使用两种类型的连接：PVC和SVC。

PVC 永久虚电路连接是由网络提供而建立的两个端点之间的连接。VPI和VCI是为了永久连接而定义的，它们的值输入到每个交换机的交换表中。

SVC 在交换虚电路连接中，当一个端点想要和另一个端点连接时，必须建立一条新的虚电路。ATM本身无法做到这件事，需要网络层地址和另一个协议（如IP）的服务。该协议的信令机制利用两个端点的网络层地址做出一个连接请求。实际的机制依赖于网络层协议。

18.2.4 交换

ATM使用交换机将信元从源端点路由到目的端点。交换机使用VPI和VCI来路由信元。路由要求整个标识符。图18.15显示了一个VPC交换机如何路由信元。VPI为153，VCI为67的信

元到达交换机接口（端口）1，交换机检查它的交换表。该交换表每行存储六项信息：到达接口号、输入VPI、输入VCI以及对应的输出接口号、新的VPI和新的VCI。交换机在表中找到接口1和VPI为153及VCI为67的表项，并发现对应的输出接口为3、VPI为140和VCI为92。改变信元头部的VPI和VCI为140和92，并通过接口3发送信元。

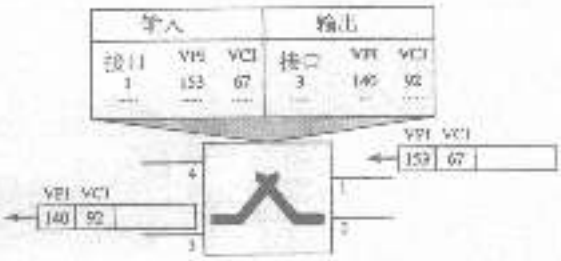


图18.15　交换机的路由

交换结构

交换技术创立了许多令人感兴趣的特性，以提高交换机处理数据的速度。我们已在第8章讨论了交换结构。

18.2.5　ATM层

ATM标准定义了三个层，从上到下依次为：应用适配层、ATM层和物理层（见图18.16）。



图18.16　ATM层

端点使用所有这三层，而交换机只使用底下的两层（见图18.17）。

物理层

像以太网和无线局域网一样，ATM的信元可以在任何物理层介质中传输。

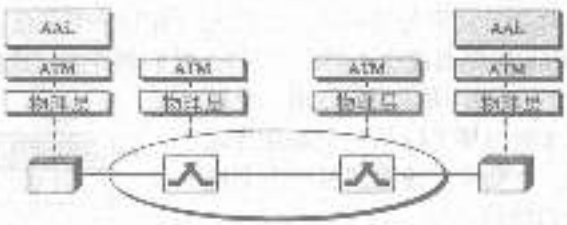


图18.17　端设备和交换机中的ATM层

SONET　ATM最初的设计是基于SONET（见第17章）作为物理层介质。SONET更受偏爱有两个原因。首先，SONET介质的高速率反映了ATM的设计和基本原理。其次，使用SONET可以清晰地定义信元的边界。如我们在第17章中所看到的，SONET详细说明了指针的用法以便于定义有效载荷的起始点。如果定义了第一个ATM信元的起始点，那么同一有效载荷的其他信元就可以很容易地确定，因为信元间没有间隙。只需要向前数53个字节就可以找到下一个信元。

其他物理技术　ATM并没有限制物理层只能使用SONET，其他技术，甚至无线技术，都可能被使用。然而，信元边界的问题必须解决。一个解决方法是，让接收方猜测信元的结尾，并在5字节的头部中应用CRC（循环冗余码校验）。如果没有错误，信元的结尾就找到了，这

种方法有很高的可能性和准确性。向后数52个字节就找到了信元的起始点。

ATM层

ATM层（ATM layer）提供了路由、通信量管理、交换和复用服务。它按照以下方式处理输出通信：从AAL子层接收48字节的分段，然后通过添加5字节的头部将它们转换成53字节的信元（见图18.18）。

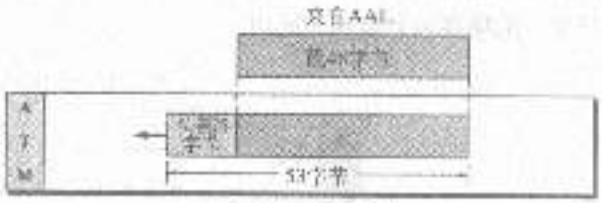


图18.18 ATM层

头部格式 ATM使用两种格式的头部：一种用于用户到网络接口（UNI）信元，另一种用于网络到网络接口（NNI）信元。图18.19以逐字节的形式显示了ITU-T所建议的头部格式（每一行代表一个字节）。

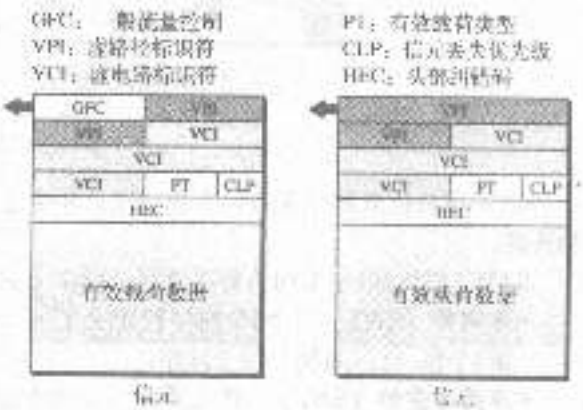


图18.19 ATM头部

- **一般流量控制（GFC）**。4位的GFC字段在UNI层提供了流量控制。ITU-T认为UNI级的流量控制在NNI级中是不必要的，因此，在NNI的头部中，这些位被添加到VPI中。较长的VPI允许在NNI级中定义更多的虚路径。这种附加的VPI的格式尚未定义。
- **虚路径标识符（VPI）**。VPI在UNI信元中是8位的字段，而在NNI信元中是12位的字段（见上文）。
- **虚电路标识符（VCI）**。VCI在两种帧中都是16位的字段。
- **有效载荷类型（PT）**。在3位的PT字段中，第一个位定义了有效载荷是用户数据还是管理信息。最后2个位的解释取决于第一个位。
- **信元丢失优先级（CLP）**。1位的CLP字段是用来提供拥塞控制的。只要还有CLP为0的信元存在，CLP设置为1的信元就必须保留。我们将在第24章讨论ATM网络的拥塞控制和服务质量。
- **头部纠错码（HEC）**。HEC是对头部前4个字节计算出来的校验码，它是系数为x^8+x^2+x+1的CRC码，用来纠正单个位错误和大部分的多个位错误。

应用适配层

应用适配层（application adaptation layer，AAL）设计用来支持两个ATM概念。首先，ATM必须接受任何类型的有效载荷，不论是数据帧还是位流。一个数据帧可能来自上层协议，这些协议发送给承载网络（如ATM）的帧都有清晰的定义。一个很好的实例是因特网。ATM也必须携带多媒体有效载荷。它能接受连续位流，把它们分成块，然后在ATM层再封装成帧。AAL使用两个子层来完成这些任务。

不论数据是数据帧还是位流，有效载荷都必须分割成48字节的分段以被信元携带。在目的端，这些分段需要重组以重新生成初始的有效载荷。AAL定义了一个称为分割与重组（segmentation and reassembly，SAR）的子层来做这个工作。在源端拆分，在目的端重组。

在SAR将数据拆分之前，必须保证数据的完整性。这个工作由一个称为会聚子层

(convergence sublayer，CS)的子层来完成。

ATM定义了四个版本的AAL：AAL1、AAL2、AAL3/4和AAL5。

AAL1　AAL1支持按照恒定比特率传输数据的应用，如视频和音频。它允许ATM和现有的数字电信网络（如语音通道和T线路）相连接。图18.20显示了一个数据位流如何被拆分成47字节的块并被封装到信元中。



图18.20　AAL1

CS子层将位流划分成47字节的分段，并传递给下面的SAR子层。注意，CS子层并没有添加头部。

SAR子层添加1字节的头部并将这个48字节的分段传递给ATM层。这个头部有两个字段：

- **序列号（SN）**。这个4位的字段定义了一个序列号来为位排序。第一个位用于定时，其余3个位（以8为模）用于排序。
- **序列号保护（SNP）**。第二个4位字段保护第一个字段。前3个位自动纠正SN字段。最后一个位是奇偶位，用来检测所有8位的错误。

AAL2　AAL2最初打算用来支持可变数据率的位流，但它已经被重新设计了。它现在用于低比特率通信和短帧通信，如音频（压缩或未压缩）、视频或者传真。AAL2应用的一个很好的例子是应用在移动电话中。AAL2允许复用的短帧封装到一个信元中。

图18.21显示了将来自一个源端（一个移动电话用户）或几个源端（几个移动电话用户）的短帧封装成一个信元的过程。

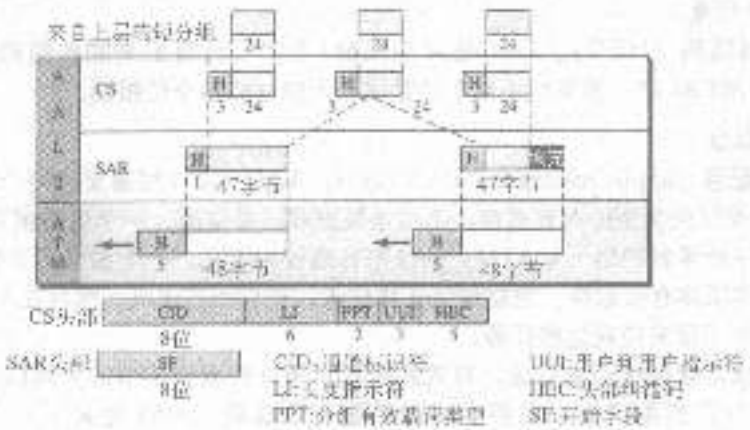


图18.21　AAL2

CS层的开销包括五个字段：

- 通道标识符（CID）。8位的CID字段定义了短分组的通道（用户）。
- 长度指示符（LI）。6位的LI字段指明了最后多少个分组是数据。
- 分组有效载荷类型（PPT）。PPT字段定义了分组的类型。
- 用户到用户指示符（UUI）。UUI字段可以用于端到端的用户。
- 头部纠错码（HEC）。最后5位用于头部的纠错。

SAR层唯一的开销是开始字段（SF），它定义了从分组起点的偏移量。

AAL3/4 最初，AAL3准备用来支持面向连接的数据服务，而AAL4用来支持无连接的数据服务。然而，当它们进一步发展后，人们发现这两个协议的基本问题是相同的，因此它们被合并成一个格式，称为AAL3/4。图18.22显示了AAL3/4子层。

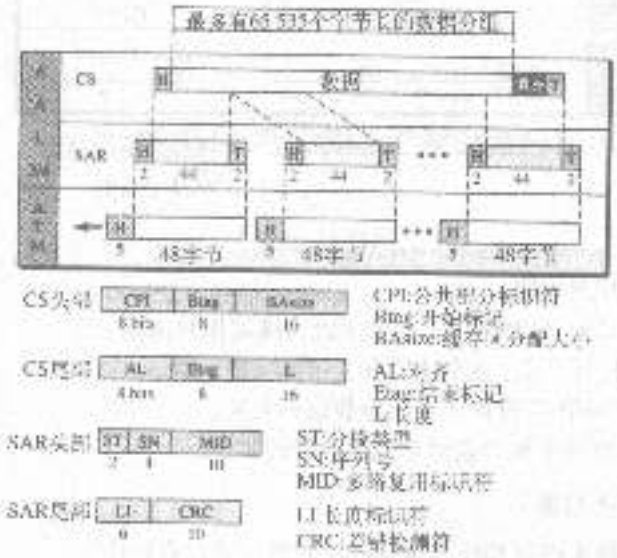


图18.22 AAL3/4

CS层的头部和尾部包括六个字段：

- 公共部分标识符（CPI）。CPI定义了后跟的字段是如何解释的。目前的值是0。
- 开始标记（Btag）。这个字段的值在每个信元中都重复，用来确定所有信元在同一分组中的起始位置。这个值和Btag（见下面）一样。
- 缓存区分配大小（BAsize）。2字节的BA字段告诉接收者即将到来的数据需要多大的缓存区。
- 对齐（AL）。1字节的AL字段是用来使尾部的剩余部分保持4字节长。
- 结束标记（Etag）。1字节的ET字段用做一个结束标记。它的值和开始标记的一样。
- 长度（L）。2字节的L字段指明了数据单元的长度。

SAR层的头部和尾部包括五个字段：

- 分段类型（ST）。2位的分段类型标识符说明了分段在报文中的位置：开头（00）、中间（01）或者结尾（10）。一个单分段的报文的ST是11。
- 序列号（SN）。这个字段和先前定义的一样。
- 多路复用标识符（MID）。10位的MID字段识别来自不同数据流并在同一个虚电路上复用的信元。

- 长度标识符（LI）。这个字段定义了分组中多少是数据，而不是系统填充的。
- CRC。尾部的最后10位是整个数据单元的CRC。

AAL5　AAL3/4提供了综合的排序和差错控制机制。这并不是在每个应用中都是必须的。对于这些应用，ATM的设计者提供了第五种AAL子层，称为简单有效适配层（simple and efficient adaptation layer，SEAL），AAL5假设所有属于单个报文的信元顺序发送，并且控制功能已经包含在发送应用的上层中了。图18.23显示了AAL5子层。

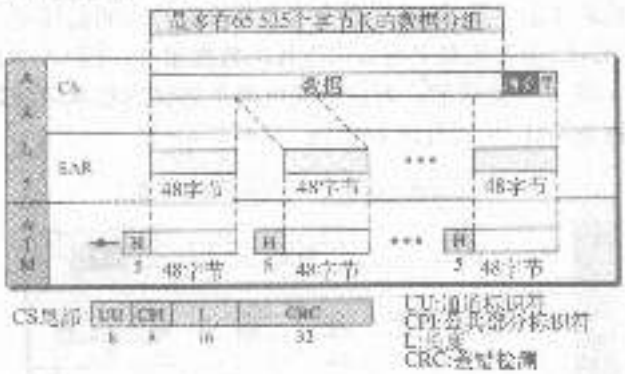


图18.23　AAL5

在CS层中有四个尾部字段：

- 用户到用户（UU）。如前所述，这个字段由终端用户使用。
- 公共部分标识符（CPI）。这个字段和先前定义的一样。
- 长度（L）。2字节的L字段指明了原始数据的长度。
- CRC。最后4字节用于整个数据单元的差错控制。

18.2.6　拥塞控制和服务质量

ATM有很先进的拥塞控制和服务质量，我们将在第24章讨论。

18.3　ATM局域网

ATM主要用于广域网（WAN ATM），然而，该技术也适用于局域网（ATM LAN）。高速数据速率技术（155 Mbps和622 Mbps）吸引了众多寻求在局域网中实现更高速数据传输的设计者的注意力。另外，ATM技术有几大优势使其成为构建高速局域网的理想选择：

- ATM技术支持两个终端用户间的不同类型的连接。而且它支持永久的和暂时的连接。
- ATM技术支持不同应用的多种带宽条件下的多媒体通信。它能保证用于实时视频的每秒几M比特的带宽。而且它还能给非高峰时间内的文件传输提供支持。
- ATM局域网很容易适应机构扩充的需要。

18.3.1　ATM局域网体系结构

如今，我们有两种办法将ATM技术集成到一个局域网体系结构中：创建一个纯ATM局域网（pure ATM LAN），或继承ATM局域网（legacy ATM LAN）。图18.24说明了这种分类法。

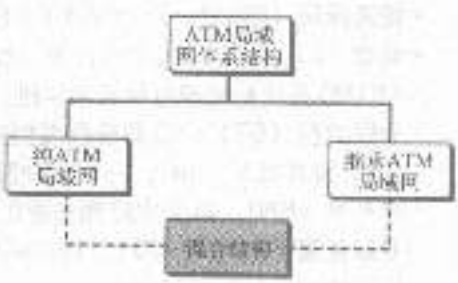


图18.24　ATM局域网

纯ATM体系结构

在纯ATM体系结构中，用一台ATM交换机

来连接一个局域网中的多个工作站，其连接方式与多个工作站连接到一台以太网交换机的方式完全一样。图18.25说明了这种情况。

按这种方式，工作站以ATM技术（155 Mbps和652 Mbps）中的两个标准速率之一进行数据交换。然而，工作站使用了虚路径标识符（virtual path identifier，VPI）和虚连接标识符（virtual connection identifier，VCI），而不是源和目的地址。

这种方法有一个严重的缺陷。该系统需要完全从最基础的水平开始构建，现存的局域网不能升级为纯ATM局域网。

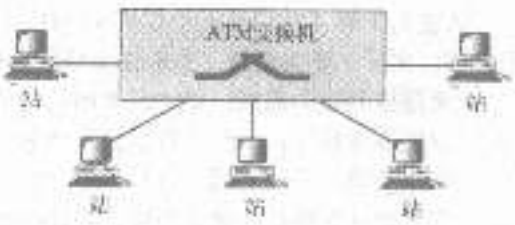


图18.25 纯ATM LAN

继承局域网体系结构

第二种方法是将ATM技术用做连接传统LAN的主干网，图18.26说明了这种继承ATM局域网的体系结构。

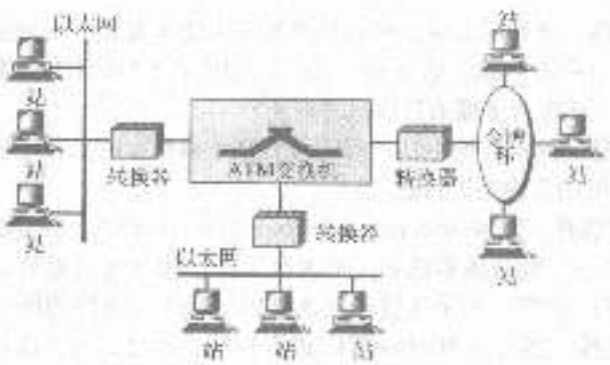


图18.26 继承ATM LAN

按这种方式，同一局域网上的工作站能以传统局域网（以太网、令牌环等）的数据速率和数据格式来交换数据。但是，当位于两个不同局域网上的两个工作站需要交换数据时，它们却需要一种转换装置来变换帧格式。此处的优势是来自多个局域网的输出可以多路复用到一起，以给ATM交换机创建一条具有高数据速率的输入数据流。但首先必须解决其中的几个问题。

混合体系结构

最好的解决方案可能是将前面两个体系结构结合起来。这意味着既可保留现存的局域网，同时又可允许新的工作站直接连接到ATM交换机上。**混合体系结构局域网**（mixed architecture LAN）通过增加更多的与交换机直接相连的工作站的方式，允许将继承局域网逐步移植到ATM局域网上。图18.27说明了这种体系结构。

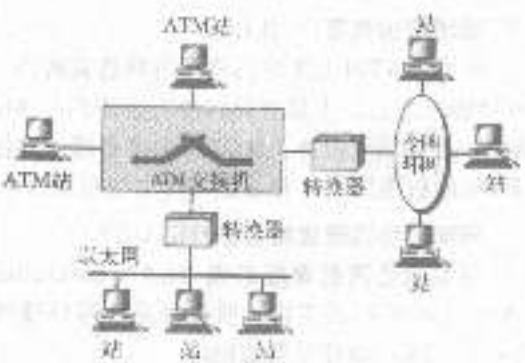


图18.27 混合体系结构ATM局域网

另外，一个特定局域网上的工作站能够运用该特殊局域网中的数据格式和数据速率进行数据交换，与ATM交换机直接相连的多个工作站能运用ATM帧交换数据。然而，问题是：

传统局域网中的工作站如何与直接连接到ATM交换机上的工作站进行通信呢？反过来情况又如何呢？下面可以看到该问题是如何得到解决的。

18.3.2 局域网仿真（LANE）

从表面上看，在局域网中应用ATM技术似乎是非常自然的事情。然而，这种相似性只限于表面，实际上要实现这一点有许多问题需要解决，现将其归纳如下：

- **无连接与面向连接**。诸如以太网这样的传统局域网是应用无连接协议的，无论何时只要分组准备好了，一台工作站就可将其发送到另一台工作站。这里就没有连接建立和连接终止阶段。另一方面，ATM是一种面向连接的协议。一台要给另一台工作站发送信元的工作站应该首先建立连接，并且在所有的信元发送完毕之后终止该连接。
- **物理地址与虚电路标识符**。与第一个问题紧密联系的另一个问题是两者寻址间的差异。诸如以太网这样的无连接协议通过源地址和目的地址来确定分组的路由，而对于诸如ATM这样的面向连接的协议，则是通过虚连接标识符（VPI和VCI）来确定信元的路由的。
- **多播和广播传递**。诸如以太网这样的传统局域网能多播和广播分组。一个工作站能给一组工作站或所有的工作站发送分组。然而，即使在ATM网络中能实现一点对多点的连接，但要在它上面进行多播或广播也不容易。
- **互操作性**。在混合体系结构中，一台连接到继承局域网的工作站应该能够与一台直接连接到ATM交换机的工作站进行通信。

一种称为局域网仿真（local area network emulation，LANE）的方法可以解决上面所提到的问题。该方法允许一个混合体系结构中的多个工作站相互进行通信。该方法运用了仿真的思想。多个工作站可以使用一种无连接服务来模拟一种面向连接的服务。工作站将源地址和目的地址用于初始连接，然后运用VPI和VCI进行寻址。而且，该方法还允许各工作站使用单播、多播和广播地址。最后，在帧通过交换机发送之前，该方法使用继承格式将帧转换成ATM信元。

18.3.3 客户/服务器模型

LANE是设计作为客户/服务器模型（client/server model）来处理前面所讨论的四个问题。该协议使用一种类型的客户机和三种类型的服务器，如图18.28所示。

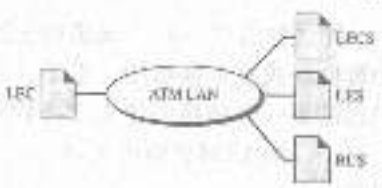


图18.28　LANE中的客户和服务器

局域网仿真客户（LEC）

所有的ATM工作站都将局域网仿真客户（LAN emulation client，LEC）软件安装在三个ATM协议之上。上层协议并没有意识到ATM技术的存在。这些协议将其请求发送到LEC，以请求一种局域网服务，如运用MAC单播、多播或广播地址进行的无连接数据传输服务。然而，LEC仅解析该请求，并将结果传递到服务器中。

局域网仿真配置服务器（LECS）

局域网仿真配置服务器（LAN emulation configuration server，LECS）用于客户机和LANE之间的初始连接。服务器总是等待接收初始连接。它有一个公开的ATM地址，并且系统中每个客户端都知道该地址。

局域网仿真服务器（LES）

局域网仿真服务器（LAN emulation server，LES）软件安装在LES上。当一台工作站根据物理地址接收到一个将要发送到另一台工作站的帧时，LEC就给LES发送一个特殊的帧。服

务器就在源和目的工作站之间创建一条虚电路。此时源工作站就能使用这条虚电路（和相应的标识符）给目的工作站发送一个或多个帧。

广播/未知服务器（BUS）

多播和广播要求使用另一台称为**广播/未知服务器**（broadcast/unknown server，BUS）的服务器。如果一台工作站需要给一个工作站组或每台工作站发送一个帧时，它首先将该帧发送到BUS。该服务器具有到每台工作站的永久虚连接。此时该服务器就创建所接收帧的多个拷贝，并将其中一个拷贝发送到一组工作站或所有的工作站，来模拟多播或广播过程。该服务器还能通过将帧发送到每台工作站的方式来传递单播帧。在此情况下，目的地址是未知的。有时这种做法比从LES获取连接标识符更为高效。

18.3.4 具有客户/服务器的混合体系结构

图18.29说明了一个混合体系结构ATM局域网中的客户机和服务器。图18.29中，有二种类型的服务器连接到了ATM交换机上（实际上，它们可以是交换机的一部分），而且图中还有两种类型的客户机。设计用于发送和接收LANE通信的工作站A和工作站B直接与ATM交换机相连。传统的继承局域网中的工作站C，D，E，F，G和H都通过一个转换器连接到交换机上。这些转换器充当LEC客户机，并代表它们所连接的工作站进行通信。

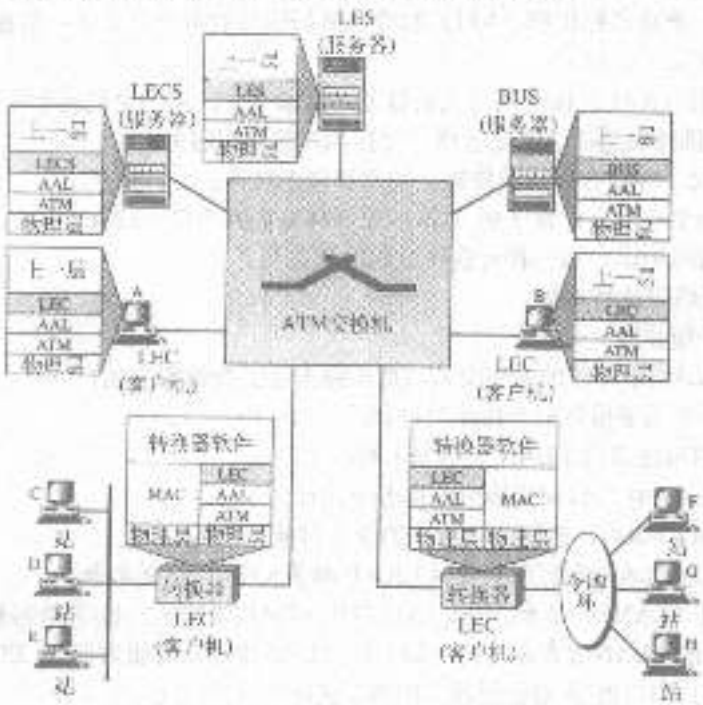


图18.29 使用LANE的混合体系结构ATM局域网

推荐读物

为了深入地讨论与本章有关的主题，我们推荐下列读物。本书末的方括号[……]中给出了参考资料。

读物

帧中继和ATM在[Sta98]的第3章中有讨论。ATM LAN在[For03]的第14章中有讨论。在

[Kei02]的第7章中也有关于ATM LAN的讨论。

本章小结

- 虚电路交换是共享链路的数据链路层技术。
- 虚电路标识符（VCI）标识了两个交换机间的帧。
- 帧中继是能处理突发性数据的相对高速、廉价的技术。
- PVC和SVC连接都应用于帧中继。
- 数据链路连接标识符（DLCI）标识了帧中继中的虚电路。
- 异步传输模式（ATM）是信元中继协议和SONET的组合，允许高速连接。
- 信元是较小的固定长的信息块。
- ATM数据分组由53个字节（其中5个字节头部和48个字节有效载荷）组成。
- ATM消除了与不定长分组相关的可变延迟时间问题。
- ATM能处理实时传输。
- 用户到网络接口（UNI）是用户和ATM交换机之间的接口。
- 网络到网络接口（NNI）是两个ATM交换机之间的接口。
- 在ATM中，两个端点间的连接通过传输路径（TP）、虚路径（VP）和虚电路（VC）完成。
- 在ATM中，虚路径标识符（VPI）和虚电路标识符的组合定义了一条虚连接。
- ATM标准定义了三层：

a. 应用适配层（AAL）接受来自上层服务的传输信息，并把它们映射成信元。

b. ATM层提供路由选择、通信管理、交换和多路复用服务。

c. 物理层定义了传输介质、位传输、编码和电光转换。

- AAL分成两个子层：汇聚子层（CS）和分割与重组子层（SAR）。
- 有四种不同的AAL，每一种对应特定的数据类型：

a. AAL1用于恒定比特率流

b. AAL2用于短分组

c. AAL3/AAL4用于传统的分组交换（虚电路方法或数据报方法）

d. AAL5用于不需要排序和差错控制机制的分组

- 可以采用ATM技术用于LAN（ATM LAN）。
- 在纯ATM LAN中，ATM交换机连接各个站点。
- 在继承ATM LAN中，连接传统LAN的骨干网使用ATM技术。
- 混合结构ATM LAN组合了纯ATM LAN和继承ATM LAN的特性。
- 局域网仿真（LANE）是允许在LAN中使用ATM技术的客户机/服务器模型。
- LANE软件包括LAN仿真客户机（LECS）、LAN仿真配置服务器（LECS）、LAN仿真服务器（LES）和广播/未知服务器（BUS）模块。

习题

复习题

1. 在帧中继中没有序列号。为什么？
2. 两个设备能使用相同的DLCI连接到相同的帧中继网络吗？
3. 为什么帧中继连接LAN比起T-1线路是更好的方案？
4. 比较SVC和PVC。

5. 讨论帧中继物理层。
6. 如果所有数据单元都是一样长的，为什么多路复用效率会更高？
7. NNI与UNI的不同在什么地方？
8. TP、VP和VC之间是什么关系？
9. 如何标识ATM虚连接？
10. 写出ATM各层的名称和作用。
11. 在UNI中能定义多少条虚连接？在NNI中能定义多少条虚连接？
12. 简要描述LAN中使用ATM技术所涉及的问题。

练习题

13. 帧中继帧的地址字段是10110000000010111，那么DLCI是什么（十进制）？
14. 帧中继帧的地址字段是10110000010100l，它有效吗？
15. 如果接收到的前三个字节是十六进制的7C 74 E1，请找出DLCI。
16. 如果DLCI是178，请写出2字节地址字段的值（十六进制），假设没有拥塞。
17. 在图18.30中，在A和B之间建立一条虚连接。给出每条链路的DLCI。

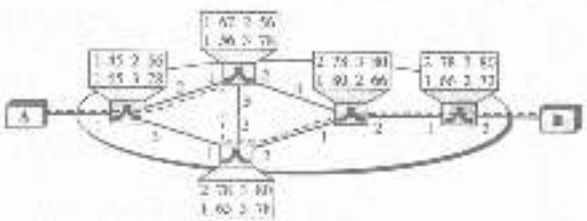


图18.30 练习17

18. 在图18.31中，在A和B之间建立一条虚连接。给出每个交换机表中相应的表条目。

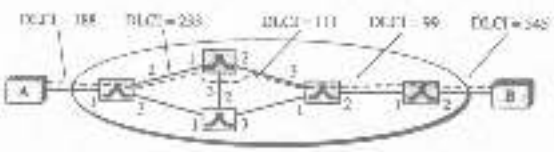


图18.31 练习18

19. AAL1层以2Mbps接收数据。那么每秒由ATM层产生多少个信元。
20. 使用AAL1，ATM的总效率（接收位与发送位的比率）是多少？
21. 如果应用程序使用AAL3/4并且有47 787字节数据进入CS，那么需要多少填充字节，多少数据单元从SAR传递给ATM层？产生多少个信元？
22. 假设没有填充，使用AAL3/4的ATM效率取决于分组长度吗？解释你的答案。
23. 在AAL3/4中，从一个输入分组得到的最少信元数是多少？从一个输入分组得到的最多信元数是多少？
24. 在AAL5中，从一个输入分组得到的最少信元数是多少？从一个输入分组得到的最多信元数是多少？
25. 解释为什么在AAL1中不需要填充而在其他AAL中需要。
26. 使用AAL3/4，这种情况我们需要_____个填充。

a. 0个字节（没有填充）　　b. 40个字节

c. 43个字节

27. 使用AAL5，这种情况我们需要_____填充。
    a. 0个字节（没有填充）　　b. 40个字节
    c. 43个字节
28. 在一个53字节的信元中，以下情况下多少个字节属于用户（假设没有填充）？
    a. AAL1
    b. AAL2
    c. AAL3/4（不是第一个信元也不是最后一个信元）
    d. AAL5（不是第一个信元也不是最后一个信元）

探究题

29. 学习理解为UNI接口提供一般流量控制的I.150协议。
30. ATM在头部的前四个字节（32位）中使用8位HEC（头部纠错码）字段来控制差错。生成二项式是x^8+x^2+x+1。说明这是如何实现的。
31. 找出LANE帧的格式，并与以太网帧进行比较。
32. 给出LANE工作所包含的各个步骤。

第四部分 网 络 层

目标

网络层负责在计算机间传输分组。可能经过多个网络（链路），而数据链路层负责在同一网络（链路）上的两个系统之间传输分组。网络层保证每一分组从源到它的最后目的地。

网络层负责从源主机到目的主机之间单独分组的传输。

网络层增加一个头部，它包含来自上一层分组的发送方与接收方的逻辑地址。如果分组经过因特网传输，还需要寻址系统来识别源与目的地址。

当若干个独立的网络（链路）连接在一起组成一个互联网时，路由器或交换机对分组进行路由选择到达它的最终目的地。网络层一个功能是提供一种路由机制。

在本书的第四部分中，首先讨论逻辑寻址（参照当特网中的IP寻址），然后讨论控制分组从源到目的地传输的主要部分和一些附加协议。

第四部分专门介绍网络层及其提供的服务。

第19章 逻辑寻址

如第2章所述，网络层上的通信是主机到主机（计算机到计算机），世界上任一地方的一台计算机需要与世界上任何另一台计算机通信，通常是通过因特网实现。发送计算机所要传输的分组在到达目的计算机前可能经过若干个局域网或广域网。

为了这一层的通信，我们需要有一个全球寻址的方案，在第2章中这称为逻辑寻址。目前我们用术语IP地址（IP address）表示，它是指TCP/IP协议栈的网络层中的逻辑地址。

因特网地址是32位长，它给出最大地址个数是2^{32}。如没有混淆，这些地址称作为IPv4（IP版本4）地址或简称为IP地址。

除了对IP层其他有关问题外，我们需要更多的地址，这激发了称为IP新的一代或IPv6（IP版本6）的IP层的一个新的设计。在这个版本中，因特网使用128位地址，给地址分配更多的灵活性。这些地址称为IPv6（IP版本6）地址。

本章首先讨论当前因特网使用的IPv4地址，然后讨论可能不久将来成为主要的IPv6地址。

19.1 IPv4地址

IPv4地址（IPv4 address）是一个32位地址，它唯一地与通用地定义了一个连接在因特网上的设备（例如，主机或路由器）。

> IPv4地址是32位长。

IPv4地址是唯一的，唯一的是指每一个地址定义了在因特网上的一个且仅有一个设备。在因特网上的两个设备永远不会具有同样的地址，但是稍后我们将会看到，有一些策略可将一个地址暂时指定给一个设备，然后它被取消又将该地址指定给另一设备。此外，如果在因特网上一个设备连接m个物理网络，则它需要拥有m个地址。如稍后看到的路由器就是这样的一个设备。

IPv4地址是通用的，因为这个地址系统必须被任何一个愿意连接到因特网上的主机所接受。

> IPv4地址是唯一的与通用的。

19.1.1 地址空间

IPv4协议定义了一个地址空间（address space）。地址空间是该协议能用地址的总个数，如果协议使用N位定义地址，由于每位可能有两个不同值（0和1），因而N位有2^N个地址，所以地址空间是2^N。

IPv4使用32位地址，这就是说地址空间是2^{32}或4 294 967 296（多于四百万）。在理论上，如果没有其他限制，就可有超过四百万个设备连接到因特网上。我们不久将会看到，因为存在限制，实际可使用地址的个数要少一些。

> IPv4的地址空间是2^{32}或4 294 967 296。

19.1.2 标记法

IPv4地址有两种常用的标记法：二进制标记法（binary notation）和点分十进制标记法（dotted-decimal notation）。

二进制标记法

在二进制标记法中，IPv4地址用32位表示。为了使这种地址的可读性更强，通常在每8位之间插入一个或多个空格。每8位通常称为一个字节。因此，常可听到将IP地址称为32位地址、4个8位地址或一个4字节地址等说法。下面是一个用二进制标记法表示的IPv4地址的实例。

01110101 10010101 00011101 00000010

点分十进制标记法

为了使IPv4地址更加简洁和更易阅读，这种两地址通常用十进制形式来书写，并用十进制点来分隔这些字节。下面是上述地址的点分十进制标记法（dotted-decimal notation）。

117.149.29.2

图19.1表示了IPv4地址的二进制标记法和点分十进制标记法。注意：由于每个字节（8位组）是8位，每个点分十进制数的取值范围是0到255。

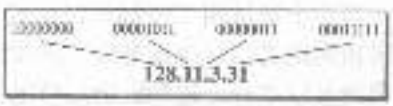


图19.1 IPv4地址二进制标记法和十进制标记法

> 数制系统在附录B中复习。

例19.1

将下列IPv4地址由二进制标记法转换成点分十进制标记法：

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

解：

我们可将8位组用一个等价的十进制数代替（见附录B）并添加点来分隔。

a. 129.11.11.239

b. 193.131.27.255

例19.2

将下列IPv4地址由点分十进制标记法转换成二进制标记法：

a. 111.56.45.78

b. 221.34.7.82

解：

将十进制数用等价的二进制数代替（见附录B）。

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

例19.3

寻找下列IPv4地址的标记法中的差错：

a. 111.56.045.78

b. 221.34.7.8.20

c. 75.45.301.14

d. 11100010.23.14.67

解：

a. 首位必需非0(045)。

b. IPv4地址不能有多于4位。

c. 每个数必须小于等于255（301在范围外）。

d. 二进制标记法与点分十进制标记法不能混用。

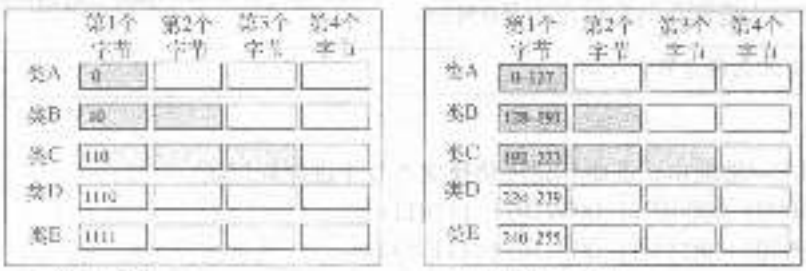
19.1.3　分类寻址

IPv4寻址在开始时使用了类的概念，这种体系结构称为**分类寻址**（classful addressing）。虽然这个方案现正在逐步被取代，但这里还是简短地讨论它以说明分类寻址的基本原理。

在分类寻址中，地址空间被划分为5类：A、B、C、D和E。每一类拥有地址空间的某些部分。

> 在分类寻址中，地址空间被划分为5类：A、B、C、D和E。

当一个地址用二进制标记法或点分十进制标记法表示时，我们可从开头的几位求出这个地址所属的类。如果地址以二进制标记法给出，那么前面的几个位可立即告诉我们所属的类。如果地址以点分十进制标记法给出，那么第1字节就定义了类。这两种方法如图19.2所示。



a) 二进制标记法　　b) 点分十进制标记法

图19.2　以二进制和点分十进制标记法求类

例19.4

求每个地址的类

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

解：

a. 第1位是0，这个地址是A类（class A address）。

b. 前2位是11，第3位是0，这个地址是C类（class C address）。

c. 第1个字节是14（在0到127之间），这个地址是A类。

d. 第1个字节是252（在240到255之间），这个地址是E类。

地址类与地址块

分类寻址存在的一个问题是，每个地址类被分成一个固定数量的地址块，并且每个地址块都具有固定的长度。如表19.1所示。

让我们分析该表。当一个组织机构申请一个地址块时，被允许A类、B类或C类中的一个块。A类地址（Class A address）是为那些具有大量的主机或路由器的大型组织机构所设计的。B类

地址（Class B address）是为那些可能具有数万台主机或路由器的中型组织机构所设计的。C类地址（Class C address）是为那些具有少量主机或路由器的小型组织机构所设计的。我们可以看到这种设计的缺点，A类地址中块个数可能比几乎所有的组织机构的需要都要大，这就是说该类中许多地址被浪费掉了。B类地址块中的地址个数比大多数中型组织机构的需要量也太大，因而该类中的许多地址也被浪费掉了。而C类地址块的个数比大多数组织机构的需要又太少。在下一章中将会看到D类地址（Class D address）是为多播而设计的，这类中的每一地址用来定义连接因特网上的一组主机。因特网管理机构错误地预测需要268 435 356组，这是永远不会发生的，这里浪费了太多的地址。最后，E类地址（Class E address）是为将来用而保留的，仅有少数几个被使用，这也引起了地址的浪费。

表19.1 IPv4分类寻址块的个数与每块的长度

类	块的个数	块的长度	应用
A	128	16 777 216	单播
B	16 384	65 536	单播
C	2 097 152	256	单播
D	1	268 435 456	多播
E	1	268 435 356	保留

在分类寻址中，大部分可用的地址被浪费了。

网络号与主机号

在分类寻址中，A类、B类和C类中一个IP地址被分成网络号（netid）与主机号（hostid）。依照地址类的不同，它们具有不同的长度。图19.2表示了网络号与主机号的字节。网络号用彩色表示，主机号用白色表示。注意：D类和E类地址并没有分成网络号与主机号。

在A类中，一个字节定义网络号而三个字节定义主机号。在B类中，二个字节定义网络号，二个字节定义主机号。在C类中，三个字节定义网络号而一个字节定义主机号。

掩码

虽然分类寻址中网络号与主机号长度是预先规定的，但我们也可用连续1的串后跟连续0的串组成的一个32位掩码（mask）（也称为默认掩码（default mark））。A类、B类和C类的掩码如表19.2所示。D类和E类没有掩码的概念。

表19.2 分类寻址的默认掩码

类	二进制格式	点分十进制格式	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

掩码能帮助我们找到网络号和主机号。例如，对A类地址，掩码有8个1，就是指A类任何地址前8位定义网络号，后面24位定义主机号。

表19.2中最后的列用/n的形式表示掩码，在分类寻址中这里n是8、16或24。这种标记法称为斜杠标记格式或无类域间路由选择（Classless Interdomain Rounting，CIDR）。这种标记法用于无类寻址中，我们将在后面讨论。这里介绍是否为它也可用于分类寻址中。后面还将看到分类寻址是无类寻址的一种特殊情况。

子网化

在分类寻址中，已引入了子网化（subnetting）概念。如果一个组织机构指派A类或B类中一大块地址，它可将这些地址划分为几个类组，并赋予每一组为较小的网络（称为子网，subnet）。特殊情况下，一个子网与邻近子网间有共享地址的部分。子网化是在掩码中增加1的

个数。稍后在我们讨论无类寻址时会看到它。

超网化

虽然A类地址和B类地址已几乎用完，但仍有大量要求中等规模的地址块。然而，C类地址块空间只能容纳最多256个地址，无法满足多数组织机构的需要，甚至一个中等规模的组织机构也会需要更多的地址。一种解决问题的方法是**超网化**（supernetting）。在超网化中，一个组织机构能将几个C类块地址构成更大范围的地址空间。换言之，将几个网络联合起来构成一个**超网**（supernet）。一个组织机构要申请一组C类地址块而不是仅仅一个，例如，一个需要1 000个地址的组织机构可以申请4个C类地址块。该组织机构就可以使用这些地址创建一个超网。超网化是在掩码中减小1的个数。例如，一个组织机构需要4个C类地址块，掩码从/24变成/22。我们将会看到在无类寻址中没有超网化的需要。

地址耗尽

分类寻址方案的缺点是与导致有效地址耗尽的因特网快速发展密切相关。因特网设备个数还是小于2^{32}地址空间，我们已耗尽A类和B类地址。而C类地址块对多数中等规模的组织机构来说太小了。减轻这问题的一个解决办法是无类寻址思想。

> 过时的分类寻址被无类寻址所取代。

19.1.4　无类寻址

为了克服地址耗尽并使更多组织机构接入因特网，设计并开发了**无类寻址**（classless addressing）。在这个方案中，没有类但仍提供地址块。

地址块

在无类寻址中，当一个小的或大的实体需要连接因特网时，给它分配一个适合的地址块。块的大小（地址的个数）按实体大小规模与性质来决定。例如，可以给一个家庭仅分配一个由2个地址组成的地址块，给一个大型的组织机构分配由几千个地址组成的地址块。因特网服务提供商（ISP）可根据其提供服务的顾客数分配几十万个或几万个地址组成的地址块。

限制条件　为了简化对地址处理，因特网管理机构对无类地址块强加了3个限制条件：

1. 块中的地址必须是一个接着一个连续的；
2. 一个块中地址的个数是2的整数次次幂（1，2，4，8，…）；
3. 块的起始地址必须能被块的个数整除。

例19.5

图19.3表示了给一个小型商业公司分配一个由16个地址组成的地址块，用二进制标记法和点分十进制标记法表示。

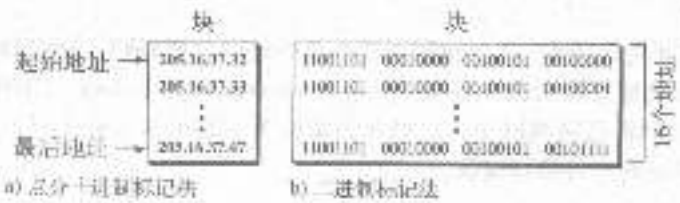


图19.3　分配给一个小型组织机构16个地址的块

我们可以看见对这个块实施了限制。地址都是连续的，地址的个数是2的次幂（$16=2^4$），而起始地址可用16整除。起始地址转换成十进制数是3 440 387 360，当它被16除，商是

215.024.210。在附录B中，我们表示了如何求一个IP地址的十进制值。

掩码

定义块地址的一个较好的方法是选择块中任一地址和掩码。如前所述，一个掩码是32位的数，其中最左边n位都是1，而最右边$32-n$位都是0。然而，在无类寻址中，一个地址块所用的掩码可取0到32中任一个值。用斜杠（CIDR标记法）很方便给出这个n的值。

> 在IPv4寻址中，块地址可用x.y.z.t/n来定义，其中x，y，z，t定义了一个地址，而/n定义掩码。

地址和/n可以完全定义整个地址块（起始地址，最后地址和地址个数）。

起始地址　块的起始地址的二进制标记法可通过设置最右边32－n位都是0求得。

> 用二进制标记法的起始地址可设置最右边的32－n位都是0求得。

例19.6

分配给某一小型组织机构一个地址块。我们已知块中一个地址是205.16.37.39/28，求该块的起始地址？

解：

已知地址的二进制表示是11001010 00010000 00100101 00100111，如果置最右边32－28位都为0，则得到块的起始地址11001010 00010000 00100101 00100000或205.16.37.32。这个块的实际情况，如图19.3所示。

最后地址　块的最后地址的二进制标记法可通过设置最右边32－n位都是1求得。

> 用二进制标记法的最后地址可设置最右边的32－n位都是1求得。

例19.7

求例19.6中的最后地址。

解：

已知地址的二进制表示是11001010 00010000 00100101 00100111，如果置最右边32－28位都为1，则得到块的最后地址11001010 00010000 00100101 00101111或205.16.37.47。这个块的实际情况，如图19.3所示。

地址个数　块中地址个数是最后地址与起始地址的差加1，可简单地用公式2^{32-n}表示。

> 块中地址个数可简单用公式2^{32-n}求得。

例19.8

求例19.6中的地址个数。

解：

n的值是28，这就是说地址个数是2^{32-28}或16。

例19.9

求起始地址，最后地址和地址个数另一种方法是用32位二进制（或8位十六进制）数的掩码表示。当我们编制一个程序去求这些信息时，这特别有用。在例19.5中，/28可表示为

11111111 11111111 11111111 11110000（28个1和4个0），求

a. 起始地址；

b. 最后地址；

c. 地址个数。

解：

a. 起始地址可用给定地址与掩码进行与运算求得。这里与运算是对应位逐位与操作，如果两位都是1，则与操作结果为1，否则为0。

地址：11001010 00010000 00100101 00100111

掩码：11111111 11111111 11111111 11110000

起始地址：11001010 00010000 00100101 00100000

b. 最后地址可用给定地址与掩码的反码进行或运算求得。这里或运算是对应位逐位或操作，如果两位都是0，则或操作结果为0，否则为1。

地址：11001010 00010000 00100101 00100111

掩码取反：00000000 00000000 00000000 00001111

最后地址：11001010 00010000 00100101 00101111

c. 地址的个数可由掩码求反，它的十进制数加1求得。

掩码取反：00000000 00000000 00000000 00001111

地址的个数：15+1=16。

网络地址

IP寻址中一个很重要的概念是网络地址（network address）。当一个组织机构已有一个地址块时，它可自由地给连接到因特网的每一个设备分配地址。然而，该块的起始地址通常（不是永远）被指作一个特殊地址，它称为网络地址，定义该组织机构的网络，也就是对于全球其他部分来说它是该组织机构本身的网络。在后面一章，将会看到起始地址是路由器用来将报文发送到该组织机构外部的地址。图19.4表示了给某组织机构分配了一个16个地址的块。

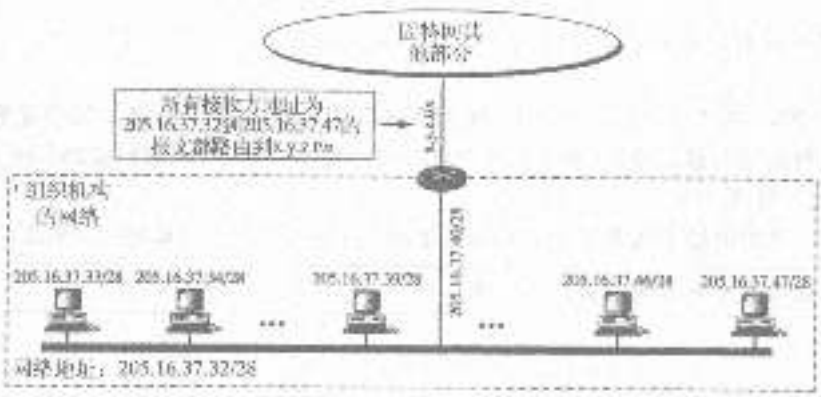


图19.4　有关地址块205.16.37.32/28的网络配置

该组织机构的网络通过路由器连接到互特网。路由器有两个地址，一个是属于该块的块，另一个属于路由器另一端所在的网络。由于我们不知道连接在另一端网络的有关信息，我们称第二个地址为x.y.z.t/n。在组织机构块中地址块中的（205.16.37.32到205.16.37.47）的所有指定报文都直接地或间接地发送到x.y.z.t/n，我们之所以说直接或间接，是因为不知道路由器连接的另一端网络的结构。

> 块中的起始地址通常不分配给任何设备；它用做向世界之其他部分表示该组织的网络地址。

层次结构

像我们碰到的其他地址或标识符一样，IP地址也有层次结构。例如，北美电话网络的层次结构有3级。最左的3个数字定义地区代码，下一个3个数字定义交换局代码，最后4个数字定义到中心局本地回路的连接。图19.5表示了电话号码的层次结构。

二级层次结构：没有子网化

当没有划分子网时，IP地址仅能定义二级层次结构。地址x.y.z.t/n的最左边的n位定义网络（组织机构的网络），而最右边的32−n位定义连接到网络的特定主机（计算机或路由器）。这两个常用术语就是前缀与后缀，即定义网络部分的地址称为前缀（prefix），定义主机部分的地址称为后缀（suffix），图19.6表示了IPv4地址的层次结构。



图19.5 北美电话网络层次结构

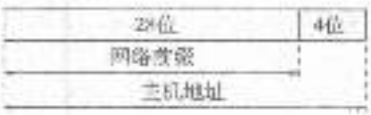


图19.6 IPv4地址中二级层次结构

前缀是网络中所有地址公用的，后缀对于不同的设备是不同的。

块中的每一个地址可看做二级层次结构：最左的n位（前缀）定义网络，最右的32−n位定义主机。

三级层次结构：子网化

一个组织机构被指派一大块地址，它想要分成几个网络（称为子网）。划分不同子网的地址。世界的其余部分仍将该组织机构看做是一个实体，但其内部有几个子网。将所有报文发送到将组织机构连接到因特网其余部分的路由器地址上；路由器将报文发送到合适的子网。然而，该组织机构需要创建多个较小地址子块，每个子块指向特殊的子网。组织机构有它自己的掩码，每个子网也必须有它自己的掩码。

作为一个例子，假定已给一个组织机构分配的地址块为17.12.40.0/26，它有64个地址。该组织机构有3个部门，需要该块地址划分为32、16和16个地址的子块。我们可用下列论点求出新的掩码：

1. 假设第一个子网的掩码是/n1，则2^{32-n1}必定是32，也就是说n1=27；

2. 假定第二个子网的掩码是/n2，则2^{32-n2}必定是16，也就是说n2=28；

3. 假定第三个子网的掩码是/n3，则2^{32-n3}必定是16，也就是说n3=28。

这是指具有掩码为/26的该组织机构有三个掩码/27、/28和/28。图19.7表示上述情况的一个配置。

让我们验证一下是否可以从子网的任一地址中求出子网地址。

a. 在子网1中，从地址17.12.14.29/27可求出子网的地址，如果我们使用该子网的掩码/27的话，这是因为

主机：00010001 00001100 00001110 00011101

掩码：/27

子网：00010001 00001100 00001110 00000000 ⇒ (17.12.14.0)

b. 在子网2中，从地址17.12.14.45/28可求出子网的地址，如果我们使用该子网的掩码/28的话，这是因为

主机：00010001　00001100　00001110　00101101

掩码：/28

子网：00010001　00001100　00001110　00100000 ⇒ (17.12.14.32)

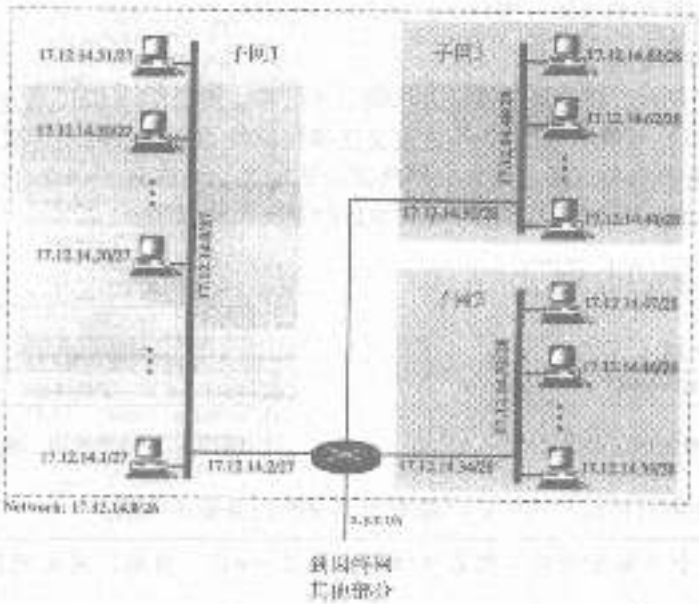


图19.7　一个子网化网络中的配置和地址

c. 在子网3中，从地址17.12.14.50/28可求出子网的地址。如果我们使用该子网的掩码/28的话，这是因为

主机：00010001　00001100　00001110　00110010

掩码：/28

子网：00010001　00001100　00001110　00101000 → (17.12.14.48)

注意：应用网络的掩码/26，可以从地址中的任一个地址求出网络地址17.12.14.0/26。这留给读者证明。

我们可以说，通过划分子网，可以得到三级层次结构。注意，在该例子中，子网前缀的长度对于不同的子网是不同的，如图19.8所示。

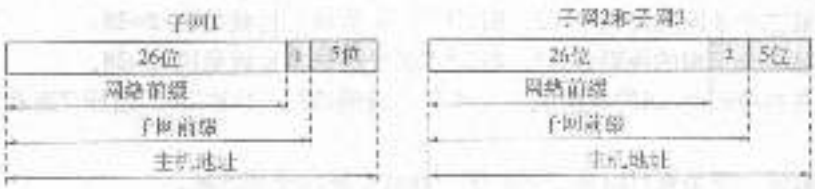


图19.8　IPv4中三级层次结构

多级层次结构

分类寻址结构没有限制层次结构的级数。一个组织机构可以将已给予的地址块划分为子块。每个子块可依次再划分为更小的子块等。在ISP中可见到这样的例子。国家级的ISP可将已给予的大块地址划分为较小的子块，将它们分配给区域级的ISP。而每个区域级的ISP将从国家级的ISP接收到地址块再划分为更小子块，并将它们分配给本地级的ISP。每个本地级的

ISP将从区域级的ISP接收到地址再划分为更小的子块，并将每一个子块分配给不同的组织机构。最后，组织机构划分接收到的块，并由它组成多个子网。

地址分配

分类寻址的下一个问题是地址分配，地址块如何分配？最终负责地址分配的是称为因特网名称和编号分配组织（ICANN）的全球权威机构。但是，ICANN通常不向个人组织机构分配地址。它分配一大块地址给ISP，每个ISP依次将已给予它的块划分为较小子块，并将各个子块分配给它的用户。换言之，将一个ISP接收到的一大块地址分配给它的用户。这称为地址聚合（address aggregation）：将多个地址块组成一个块并分配给一个ISP。

例19.10

给一个ISP分配了起始地址为190.100.0.0/16（65，536个地址）的地址块。ISP需要按如下给3组客户分发这些地址：

1. 第一组有64个客户，每个需要256个地址；
2. 第二组有128个客户，每个需要128个地址；
3. 第三组有128个客户，每个需要64个地址。

设计这些子块，并求出分配后还有多少可用的地址？

解：

图19.9表示了该情况。

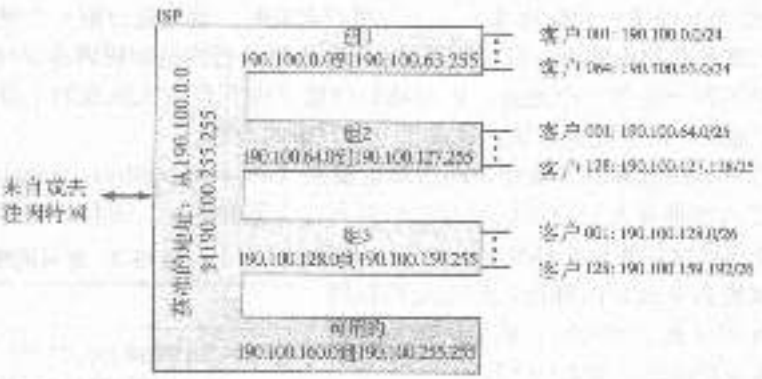


图19.9 由ISP分配和分发地址的例子

1. 组1

对于这组，每个客户需要256个地址，就是说需要用8位（$\log_2 256$）定义每一台主机，那么前缀的长度是32−8=24。地址是

第1个客户：190.100.0.0/24　190.100.0.255/24

第2个客户：190.100.1.0/24　190.100.1.255/24

…

第64个客户：190.100.63.0/24　190.100.63.255/24

总计=64 × 256=16 384

2. 组2

对于这组，每个客户需要128个地址，就是说需要用7位（$\log_2 128$）定义每一台主机，那么前缀的长度是32−7=25。地址是

第1个客户：190.100.64.0/25　190.100.64.127/25

第2个客户：190.100.64.128/25　190.100.64.255/25

...

第128个客户：190.100.127.128/25　190.100.127.255/25

总计＝128×128＝16 384

3. 组3

对于这组，每个客户需要64个地址，就是说需要用6位（$\log_2 64$）定义每一台主机，那么前缀的长度是32−6=26。地址是

第1个客户：190.100.128.0/26　190.100.128.63/26

第2个客户：190.100.128.64/26　190.100.128.127/26

...

第128个客户：190.100.159.192/26　190.100.159.1925/26

总计＝128×64＝8 192

分配给该ISP的地址个数：65 536

由ISP分配的地址个数：40 960

可用地址个数：24 576

19.1.5　网络地址转换（NAT）

越使用因特网的家庭用户和小型业务公司的数量在不断增加。最初，这些用户通过拨号线路连接到因特网，这就意味着他们在一段特定的时间内是连接在因特网上的。一个具有地址块的ISP动态地为这些用户分配地址。当一个用户需要时，就给他分配一个地址。但现在情况已不同了，家庭用户和小型商业公司能通过ADSL线路或有线调制解调器连接到因特网。另外，许多用户并不仅只使用一个地址，因为他们创建了由几台主机组成的小型网络，每台主机都需要一个IP地址。由于地址短缺，这成了一个严重的问题。

对该问题的一个快速解决方案称为网络地址转换（network address translation，NAT）。NAT能使用户在内部拥有大量的地址，而在外部只有少量的地址，而且，内部通信能使用内部的地址，而外部通信能使用外部地址。

为了将家庭或商业公司内部使用的地址与因特网上使用的地址相分离，因特网权威组织机构预留了三组地址作为专用地址，如表19.3所示。

表19.3　专用网络地址

范　围	总　数
10.0.0.0到10.255.255.255	2^{24}
172.16.0.0到172.31.255.255	2^{20}
192.168.0.0到192.168.255.255	2^{16}

任何组织机构都能使用这组地址以外的地址而不必得到因特网权威组织机构的许可。大家都知道这些预留地址是留给专用网络的，在组织机构内部它们是唯一的，但在全世界范围内它们并不唯一。路由器不能转发将这些地址作为目的地址的分组。

通过一台运行NAT软件的路由器，站点必须只有一条与全局因特网相连接的链路。图19.25显示了一个利用NAT实现的简单例子。

如图19.10所示，专用网络使用专用地址，将一个网络与全球地址连接起来的路由器使用了一个专用地址和一个全球地址。专用网络相对于因特网的其他部分是透明的，并且因特网的其他部分只能看见一台地址为200.24.5.8的NAT路由器。

地址转换

所有外发的分组都通过NAT路由器发送出来，该路由器用全球NAT地址来替代分组中的源地址。所有输入的分组也要通过NAT路由器，该路由器用相应的专用地址来替代分组中的目的地址（NAT路由器的全球地址）。图19.11说明了地址转换的一个例子。



图19.10 NAT实现

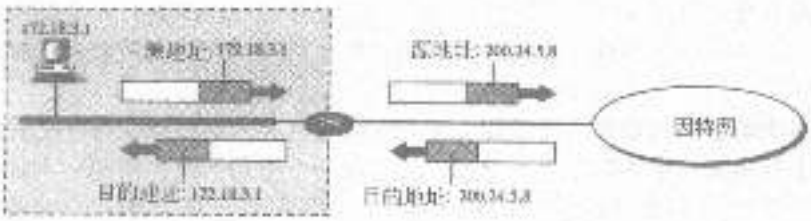


图19.11 NAT中的地址

转换表

读者可能注意到了，将源地址转换成外发地址是一件简单的事情，但是，NAT路由器如何知道来自因特网的分组的目的地址的呢？可能有数十个或数千个专用IP地址，每个专用IP地址各自属于某台特定的主机。如果NAT路由器有一个转换表，问题就可以得到解决。

使用一个IP地址 在最简单的情况下，一个转换表只有两列：专用地址和外部地址（分组的目的地址）。当路由器转换外发分组的源地址时，它同样分析其目的地址（它标志了分组的去向）。当响应从目的地址返回时，路由器利用分组的目的地址（作为外部地址）来找出分组的专用地址。图19.12说明了这种思想。注意：已改变（转换）的地址用彩色表示。

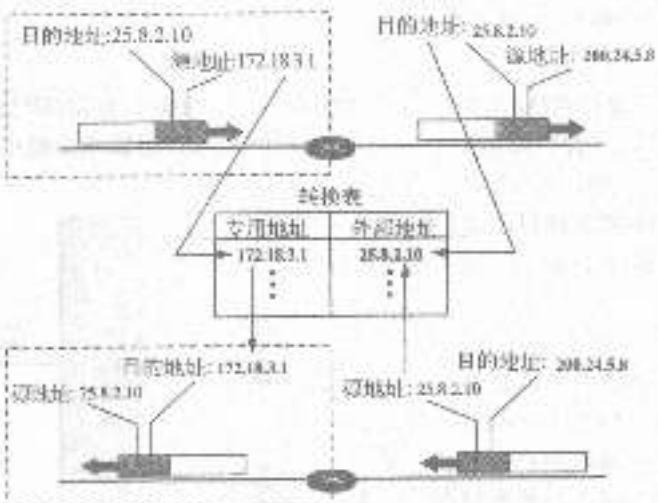


图19.12 NAT地址转换

在该策略中，通信必须由专用网络来发起。前面所描述的NAT机制要求专用网络发动通信。正如我们所看到的那样，NAT主要由ISP使用，ISP为每个用户分配一个单一的地址。然而，用户可能是一个具有许多专用地址的专用网络的一员。在这种情况下，与因特网的通信通常总是由用户站点发起的，并使用诸如HTTP、TELNET或FTP这类的客户端程序来访问远

应的服务器程序。例如，当ISP的电子邮件服务器接收到由一个非用户站点发送的电子邮件时，电子邮件就存储在用户邮箱中直到被检索。如果运用NAT技术，专用网络就不可能为它的网络之外的客户端运行服务器程序。

运用IP地址池　因为NAT路由器只有一个全球地址，因而只有一台专用网络主机能访问同样的外部主机。为了打破这种限制，NAT路由器使用了一种全球地址池技术。例如，不是只使用一个全球地址（200.24.5.8），NAT路由器能使用四个地址（200.24.5.8，200.24.5.9，200.24.5.10和200.24.5.11）。在这种情况下，有四台专用网络主机能够同时与同样的外部主机通信，因为每对地址都定义了一条连接。然而，还是存在一些不足：在本例中，不能与同样的目的端建立四条以上的连接；没有一台专用网络主机能同时访问两个外部服务器程序（如HTTP和FTP）。

同时运用IP地址和端口号　为了能在专用网络主机和外部服务器程序之间建立多对多关系，需要获取转换表中更多的信息。例如，假定在一个专用网络中有两台主机，地址分别为172.18.3.1和172.18.3.2，需要访问一台地址为25.8.3.2的外部主机上的HTTP服务器。如果转换表有5列，而不是2列，其中包括传输层协议的源和目的端的端口号，这样就可以消除二义性。在第23章中将讨论端口号。表19.4是这种表的一个示例。

表19.4　5列转换表

专用地址	专用端口	外部地址	外部端口	传输协议
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
.....	.....	.....	.....	.....

注意：当来自HTTP服务器的响应返回时，目的地址（25.8.3.2）和目的端口号（1400）的组合就确定了应该接收这一响应的内部网络主机。还须注意，要使这种转换能正常运作，临时端口号（1400和1401）必须是唯一的。

NAT和ISP

一个服务于成千客户的ISP可用NAT技术保存地址。例如，假定ISP获准1 000个地址，但有100 000个客户。每个客户被指定一个专用网络地址。ISP将输出分组中的100 000源地址的每一个地址转换为1 000个全球地址中某一个；它将输入分组中的全球目的地址转换为相应的专用地址。图19.13表示了这个概念。

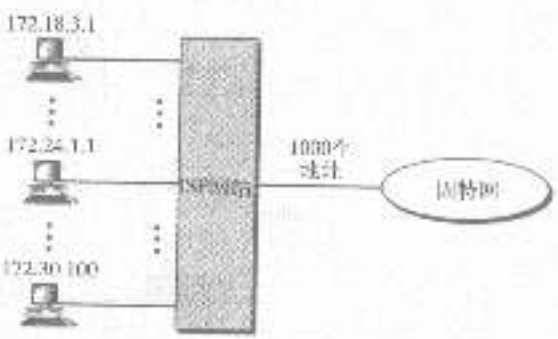


图19.13　ISP和NAT

19.2　IPv6地址

虽然诸如无类寻址、在第21章中讨论的动态主机配置协议（DCHP）和NAT等能暂时解决问题，但是因特网地址耗尽依然是长期存在的问题。这个问题与其他诸如没有对实时音频和视频传输提供资源以及某些应用不提供数据加密与鉴别等问题一起，促使IPv6协议的提出。本节中，我们比较IPv6地址与IPv4地址的结构。

19.2.1　结构

IPv6地址（IPv6 address）由16个字节（8位组）组成，它的长度是128位。

一个IPv6地址的长度是128位。

十六进制冒号标记法

为了使地址的可读性更好，IPv6协议指明了**十六进制冒号标记法**（hexadecimal colon notation）。在这种标记法中，将128位划分为8个部分，每个部分为2字节。在十六进制标记法中，2个字节包括4个十六进制数字。每4个数字用一个冒号分隔开，如图19.14所示。

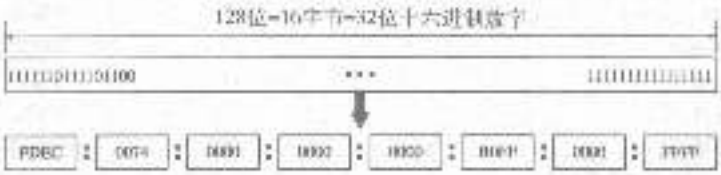


图19.14 IPv4地址用二进制与十六进制冒号标记法

缩短

即使是用十六进制格式表示，IP地址也非常长，但它的许多数字是0。在这种情况下，我们可以将地址缩短。一个部分（即两个冒号之间的4个数字）中开始的一些0可以省略。只有开始的一些0才可以省略，而在末尾的0不能被省略（见图19.15）。

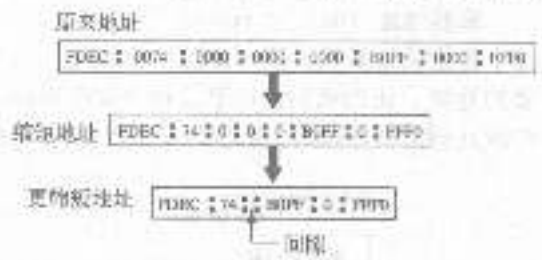


图19.15 缩短的IPv6地址

使用这种**缩短**（abbreviation）形式，0074可写为74，000F可写为F，而0000可写为0。注意：3210就不能缩短。如果有几个连续的部分仅包含0，则还可再进行缩短。我们可以将所有的0移去，而用两个冒号来代替0。注意：这种类型的缩短对一个地址仅能使用一次。如果有两串0的部分，则只能有其中的一部分进行缩短。将缩短地址再扩展并是很简单的，将未缩短的部分对齐，将一些0插入进来就可得到原来扩展的地址。

例19.11

将地址0：15：：1：12：1213扩展成原来的地址。

解：

首先需要将两倍冒号左边与原来模式左边对齐，然后将两倍冒号右边与原来模式右边插入我们所需要的多个0。

```
XXXX：XXXX：XXXX：：XXXX：XXXX：XXXX：XXXX：XXXX
  0：  15：                    1：  12：  1213
```

这就是说，原来地址是

0000：0015：0000：0000：0000：：0001：0012：1213

19.2.2 地址空间

IPv6有非常大的地址空间，有2^{128}个可用的地址。IPv6地址设计者将地址空间划分成多个类，其中少数几个位称为类型前缀。用类型前缀定义类中的每个地址。这个类型前缀是可变长的，代码设计使得第一部分的代码都没有相同的。这样在给出一个地址时就不会有歧义，而类型前缀就能很容易确定。表19.5表示了每一种类型地址的前缀，表中第3列表示每一种类

型相对于整个地址空间占有多少份额。

表19.5　IPv6地址的类型前缀

类型前缀	类　型	份　额	类型前缀	类　型	份　额
0000 0000	保留	1/256	101	未指定	1/8
0000 0001	未指定	1/256	110	未指定	1/8
0000 001	ISO网络地址	1/128	1110	未指定	1/16
0000 010	IPX（Novell）网络地址	1/128	1111 0	未指定	1/32
0000 011	未指定	1/128	1111 10	未指定	1/64
0000 1	未指定	1/32	1111 110	未指定	1/128
0001	保留	1/16	1111 1110 0	未指定	1/512
001	保留	1/8	1111 1110 10	本地链路地址	1/1024
010	基于提供者的单播地址	1/8	1111 1110 11	站点本地地址	1/1024
011	未指定	1/8	1111 1111	多播地址	1/256
100	基于地理的单播地址	1/8			

单播地址

单播地址（unicast address）定义一个单独的计算机，发送到单播地址的分组必须传递给这个指定的计算机。IPv6地址协议定义了两种类型的单播地址：基于地理的地址和基于提供者的地址。此处我们讨论第二种类型的地址，第一种类型的地址留给将来定义。基于提供者的地址通常由普通的主机作为一个单播地址使用，其地址格式如图19.1所示。

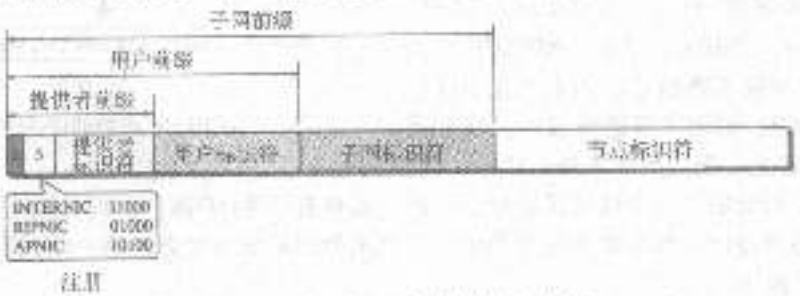


图19.16　基于提供者的单播地址前缀

基于提供者的地址具有如下的一些字段：

- 类型标识符。这个3位字段定义一个基于提供者的地址。
- 注册标识符。这个5位字段指出注册地址的机构，目前已经定义了3个注册中心。INTERNIC（代码11000）是北美的注册中心；RIPNIC（代码01000）是欧洲的注册中心；APNIC（代码10100）是亚洲和太平洋国家的注册中心。
- 提供者标识符。这个可变长字段标识因特网接入提供者（如ISP）。对这个字段的推荐长度是16位。
- 用户标识符。当一个组织机构通过一个提供者接入因特网时，就给它分配一个用户标识符。对这个字段的推荐长度是24位。
- 子网标识符。每一个用户可以有多个不同的子网，而每一个子网可以有不同的标识符。子网标识符在用户的地区范围内定义一个特定的网络。对这个字段的推荐长度是32位。
- 节点标识符。最后一个字段定义连接到子网的节点的标识。对这个字段的推荐长度是48位。使它和以太网使用的48位的链路（物理）地址相兼容。将来这个链路地址可能会和节点的物理地址相同。

多播地址

多播地址用于定义一组主机而不仅是一个主机。发送给多播地址（multicast address）的分组必须传递到该组中的每一个成员。图19.17表示了多播地址的格式。



图19.17 IPv6中多播地址

第二个字段是定义组地址的一个标志：永久的或暂时的。永久的组地址由因特网管理机构定义，并可在任何时刻对它进行访问。另一方面，暂时的组地址只是临时使用。例如，参加电话会议的系统就可使用暂时的组地址。第三个字段定义组地址的范围，各种范围如图19.17所示。

任播地址

IPv6也定义任播地址。任播地址（anycast address）与多播地址一样也定义一组计算机，然而，发送到任播地址的分组必须传递给该组成员中的一个且仅有一个，即最近的一个（具有最短路径的一个）。虽然任播地址的定义还存在争议，但因特网中对网络服务商（ISP）在大范围逻辑区域内分布的所有路由器赋予一个任播地址是可以接受的。ISP以外的路由器传递给ISP的一个发送分组到最近的ISP路由器。没有给块不分配任播地址。

保留地址

在地址空间中，另一类是保留地址（reserved address）。这类地址以8个0开始（前缀类型是00000000），这类定义的一些子类，如图19.18所示。

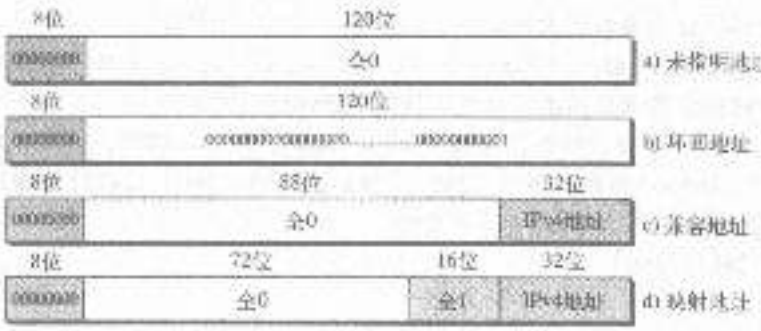


图19.18 IPv6中保留地址

未指明地址（unspecified address）是用于当主机不知道它自己的地址时，它发送查询以便找出其地址。环回地址（loopback address）是主机用来测试它自己，而不需要连接到网络上。兼容地址（compatible address）用于从IPv4到IPv6的转换（见第20章）。当一台使用IPv6协议的计算机要发送报文到另一台使用IPv6协议的计算机，但报文需要通过网络某一部分还仍旧使用IPv4协议操作时，就需要该地址。映射地址（mapped address）也是用于从IPv4到IPv6的转换，它用于一台计算机已安装了IPv6但要发送分组到一台仍使用IPv4协议的计算机。

本地地址

一个组织机构想要使用IPv6协议，但还没有连接到全球因特网，就要用这种地址。换言之，它们提供专用网络地址。在该组织机构网络以外没有人能将报文发送给使用这类地址的节点。为此目的，定义了两种类型的地址，如图19.19所示。



图19.19　IPv6中本地地址

本地链路地址（link local address）是用于一个孤立的子网；而**本地站点地址**（site local address）是用于具有几个子网的站点。

推荐读物

为了深入讨论与本章有关的主题，推荐阅读下列读物和网站，在本书末列出方括号[……]中参考资料。

读物

IPv4地址在[For06]的第4、5章、[Sta94]的第3章、[PD03]的4.1节、[Sta04]的第18章和[Tan03]的5.6节中有讨论。IPv6地址在[For06]的27.1节和[Los04]的第8章中有讨论。在[Dur01]可以找到NAT有益的讨论。

网站

- www.ietf.org/rfc.html 有请求评论的信息。

请求评论

有关IPv4协议的多数请求评论中可找到对IPv4地址的讨论：

760，781，791，815，1025，1063，1071，1141，1190，1191，1624，2113

有关IPv6协议的多数请求评论中可找到对IPv6地址的讨论：

1365，1550，1678，1680，1682，1683，1686，1688，1726，1752，1826，1883，1884，1886，1887，1955，2080，2373，2452，2463，2465，2466，2472，2492，2545，2590

NAT的讨论可在下列请求评论中找到：

1361，2663，2694

本章小结

- 在网络层，对网络间的分组传递必须有一个能唯一识别每个主机和路由器的全球地址系统。
- IPv4地址是32位长的地址，它能唯一地与通用地在因特网上定义一台主机或路由器。
- 在分类寻址中，IP地址中能识别网络的那一部分称为网络号。
- 在分类寻址中，IP地址中能识别主机的那一部分称为主机号。
- IP地址定义了一个连接到网络的设备。
- IPv4地址有五类，A类、B类和C类在每个网络上所允许的主机数是不同的。D类是多播

地址，而E类是保留地址。

- 通过检查IP地址的第一个字节，可以很容易地确定地址属于的类。
- A类、B类和C类的地址多数用于单播通信。
- D类地址用于多播通信。
- 子网化是将一个规模较大的网络划分成几个规模较小的网络，增加IP地址层次结构中的级。
- 超网化是将几个网络合并组成一个规模较大的网络。
- 在分类寻址中，我们可将地址空间划分成可变长的块。
- 无类寻址有三个限制：
  a. 地址的个数必须是2的幂次；
  b. 掩码必须包含在地址之中定义地址块；
  c. 块的起始地址必须能被块的个数整除。
- 在无类寻址中，掩码可用CIDR标记法的前缀长度（/n）来表示。
- 为了求块的起始地址，我们置最右的32−n位为0。
- 为了求块中地址个数，我们计算2^{32-n}，此处n是前缀长度。
- 为了求块的最后地址，我们置最右的32−n位为1。
- 子网化增加了n值。
- 全球地址分配的权威机构是ICANN，通常ICANN将大块地址分配给ISP，然后ISP将小块依次分配给单个客户。
- IPv6地址使用十六进制冒号标记法，还可使用缩短方法。
- IPv6地址有三种类型：单播、任播与多播。
- 在IPv6地址中，可变的类型前缀字段定义了地址类型或目的。

习题

复习题

1. IPv4地址有多少位？IPv6地址有多少位？
2. 什么是IPv4地址的点分十进制标记法？用点分十进制标记法表示IPv4地址的字节个数是多少？用十六进制标记法表示IPv6地址的数字个数是多少？
3. 在IPv4协议中，分类寻址与无类寻址有什么差别？
4. 试列出分类寻址中的类并定义每一类的功能（单播、多播、广播或保留）。
5. 解释为什么A类地址有大量的浪费，解释为什么中型或大型企业不想要C类地址块？
6. 在IPv4地址中，掩码是什么？在IPv6地址中，默认掩码是什么？
7. 在地址块中，网络地址是什么？如果有一个在给定块中的地址，如何求网络地址？
8. 简短定义了网化和超网化，在分类寻址中，子网掩码和超网掩码与一个默认掩码有什么差异？
9. 在IPv4地址中，如何识别一个多播地址？在IPv4地址中，如何识别一个多播地址？
10. 什么是NAT？NAT为什么能有助于地址耗尽？

练习题

11. 下列每个系统的地址空间是什么？
    a. 具有8位地址的系统。　　b. 具有16位地址的系统。
    c. 具有64位地址的系统。
12. 一个地址空间的全部地址是1 024个。试问需用多少位来表示？

13. 一个地址空间用三个记号0、1和2来表示地址，如果每个地址由10个记号组成，这个系统有多少个可用地址？

14. 试将下列IP地址从点分十进制标记法转换成二进制标记法：

a. 114.34.2.8　　b. 129.14.6.8

c. 208.34.54.12　　d. 238.34.2.1

15. 试将下列IP地址从二进制标记法转换成点分十进制标记法：

a. 01111111 11110000 01100111 01111101　b. 10101111 11000000 11111000 00011101

c. 11011111 10110000 00011111 01011101　d. 11101111 11110111 11000111 00011101

16. 试找出下列IP地址的类：

a. 208.34.54.12　　b. 238.34.2.1

c. 114.34.2.8　　d. 129.14.6.8

17. 试找出下列IP地址的类：

a. 11110111 11110011 10000111 11011101　b. 10101111 11000000 11110000 00011101

c. 11011111 10110000 00011111 01011101　d. 11101111 11110111 11000111 00011101

18. 试找出下列IP地址的网络号和主机号：

a. 114.34.2.8　　b. 132.56.8.6

c. 208.34.54.12

19. 在一个地址块中，我们已知一主机的IP地址是25.34.12.56/16，试问该块的起始地址（网络地址）和最后地址（有限广播地址）是什么？

20. 在一个地址块中，我们已知一主机的IP地址是182.44.82.16/26，试问该块的起始地址（网络地址）和最后地址（有限广播地址）是什么？

21. 某一组织机构被指派地址块16.0.0.0/8，网络管理员想建立500个固定长的子网。

a. 求子网的掩码；　　b. 求每个子网地址的个数；

c. 求第1个子网起始地址与最后地址；　　d. 求第500个子网起始地址与最后地址。

22. 某一组织机构被指派的地址块是130.56.0.0/16，网络管理员想建立1 024个子网。

a. 求子网的掩码；　　b. 求每个子网地址的个数；

c. 求第1个子网的起始地址与最后地址；　　d. 求第1 024个子网的起始地址与最后地址。

23. 某一组织机构被指派的地址块211.17.180.0/24，网络管理员想建立32个固定长度的子网。

a. 求子网的掩码；　　b. 求每个子网地址的个数；

c. 求第1个子网的起始地址与最后地址；　　d. 求第32个子网的起始地址与最后地址。

24. 用斜杠标记法（/n）写出下列掩码：

a. 255.255.255.0；　　b. 255.0.0.0；

c. 255.255.224.0　　d. 255.255.240.0

25. 求下列地址块的地址范围：

a. 123.56.77.32/29，　　b. 200.17.21.128/27，

c. 17.34.16.0/23，　　d. 180.34.64.64/30，

26. 某一ISP被指派以150.80.0.0/16为起始地址的块，该ISP想要按下列规则给2 600个客户分配块：

a. 第一组是200个中型的企业，每一个需要128个地址；

b. 第二组是400个小型的企业，每一个需要16个地址；

c. 第三组是2 000个家庭用户，每一个需要4个地址。

设计这些子块并对每一个子块用斜杠标记法表示。
求这些块分配后还有多少可用地址?

27. 某一ISP被指派以120.60.4.0/22为起始地址的一个块。该ISP想要给100个组织机构分配块，每个组织机构仅仅接收8个地址。设计这些子块并对每一个子块用斜杠标记法表示。
求这些块分配后还有多少可用地址?

28. 某一ISP有一个1 024个地址的块，需要分给1 024个客户。试问是否需要划分子网?并说明其理由。

29. 试给出下列地址的缩短形式：
a.2340:1ABC:119A:A000:0000:0000:0000:0000;
b.0000:00AA:0000:0000:0000:0000:119A:A123;
c.2340:0000:0000:0000:0000:119A:A001:0000;
d.0000:0000:0000:2340:0000:0000:0000:0000。

30. 试给出下列地址的原始（非缩短）形式：
a. 0::0;　　b. 0::AA::0;
c. 0::1234::3;　　d. 123::1:2。

31. 下列地址中的每一个地址是何种类型?
a. FE80::12;　　b. FEC0::24A2;
c. FF02::0;　　d. 0::01。

32. 下列地址中的每一个地址是何种类型?
a. 0::0;　　b. 0::FFFF:0:0;
c. 582F:1234::2222;　　d. 4821::14:22;
e. 54EF::A234:2。

33. 试给出一个地址的提供者前缀（十六进制冒号标记法），将该地址分配给一个在美国注册的用户，其提供者标识是ABC1。

34. 在IPv6地址中，试用十六进制冒号标记法给出：
a. 与IPv4地址129.6.12.34兼容的地址;　　b. 映射到129.6.12.34的地址。

35. 试用十六进制冒号标记法给出：
a.本地链路地址，其中节点标识符是0::123/48;
b.本地站点地址，其中节点标识符是0::123/48。

36. 试用十六进制冒号标记法给出用于本地链路范围的永久多播地址。

37. 一个主机地址为581E:1456:2314:ABCD::1211，如果节点标识是48位，试求出该主机所连接的子网地址。

38. 一个具有200个子网的网站使用B类地址132.45.0.0，该网站最近过渡到IPv6，其用户前缀为581E:1456:2314::ABCD/80，试设计子网并定义子网地址，使用32位子网标识符。

探究题

39. 试找出分配给你的单位或协会的地址块。

40. 如果你是用ISP把你的家庭连接到因特网，试问ISP的名称及分配给它的地址块是什么?

41. 一些人士张我们应考虑将整个地址空间作为一个单独的块，其中这些地址的每个范围是这个单独块的子块。详述这一思想。如果我们接受这一观念，将会对子网化产生什么后果?

42. 你所在学校或组织机构是否使用分类寻址?如果是，找出所用的地址类。

第20章 IP协议

在因特网模型中，主要的网络协议是网际协议（InternetProtocol，IP）。在这一章中，我们首先讨论互联网和网络层协议的一般问题。

然后讨论网际协议当前版本4，即IPv4。进一步使我们提出这个协议的下一代，即IPv6。不久的将来它可能成为主要的协议。

最后，我们讨论从IPv4过渡到IPv6的策略。有些读者可能注意到没有IPv5，因为IPv5是基于从未实现过的OSI模型的实验性协议。

20.1 网际互联

网络的物理层和数据链路层在本地运行，这两层共同负责网络相邻节点间的数据传递，如图20.1所示。

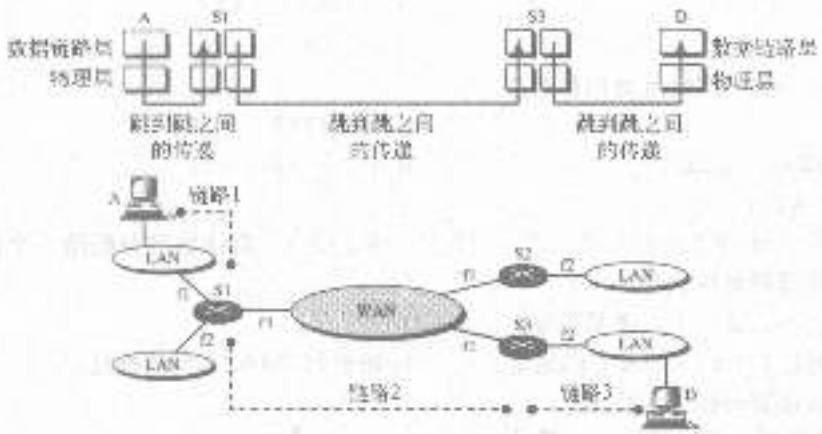


图20.1 两台主机之间的链路

这个互联网是由五个网络组成的：四个局域网和一个广域网。如果主机A需要发送一个数据分组到主机D，该分组首先需要从A到达R1（一个交换机或路由器），然后从R1到达R3，最后从R3到达主机D。我们说数据分组经过三个链路进行传递。在每个链路中，包括两个物理层和数据链路层。

然而，这里有一个很大的问题。当数据到达R1的接口f1时，R1如何知道应该从f3接口将其发送出去？在数据链路层（或物理层）并没有规则来帮助R1做出正确的决策。而且，帧也没有携带任何路由选择信息，帧中只包含主机A的MAC地址（源地址）和R1的MAC地址（目的地址）。对于一个局域网或广域网，传递意味着通过一条而不是多条链路传送帧。

20.1.1 网络层需求

为了解决通过多条链路进行传递的问题，就设计了网络层（有时称之为互联网络层）。网络层不仅负责主机间的传递，而且还负责通过路由器或交换机对分组进行路由选择。图20.2说明增加了一个网络层的同一互联网络。

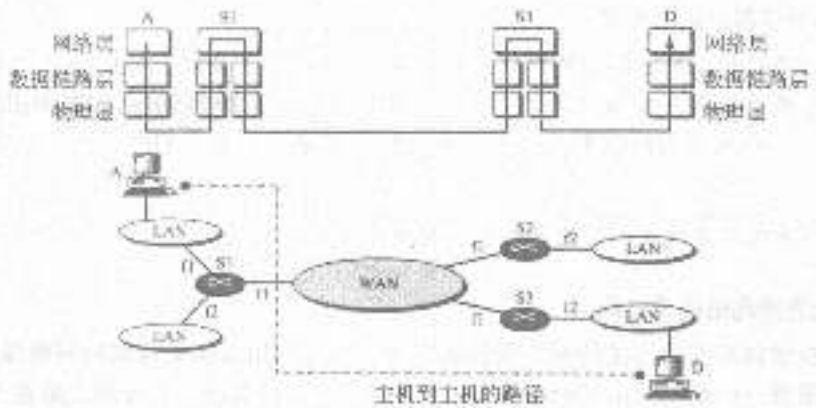
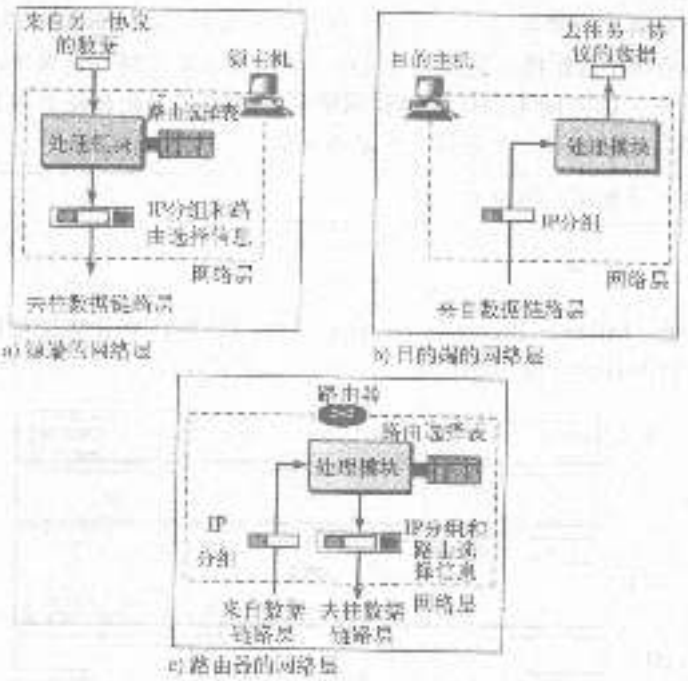


图20.2 互联网中的网络层

图20.3说明了在源端、路由器端和目的端的网络层功能总的思想。在源端网络层负责将来自另一个协议（如一个传输层协议或一个路由协议）的输入数据生成一个分组，分组的头部包含源和目的逻辑地址以及其他信息。网络层负责检验路由表寻找路由选择信息（如分组出去的接口或下一节点的物理地址）。如果分组太大，那么就得对其进行分段（在本章的后面讨论分段）。



a) 源端的网络层　b) 目的端的网络层

c) 路由器的网络层

图20.3 源端、路由器端和目的端的网络层

交换机或路由器中的网络层负责对分组进行路由选择。当一个分组到达时，路由器或交换机就从它的路由选择表中为该分组找到一个必须将其发送出去的接口。改变头部的某些内容后的分组按路由选择信息再传送给数据链路层。

目的端的网络层负责地址验证。它确保分组中的目的地址与主机地址是相同的。如果分组是一个分段，网络层就等待所有的分段到达后再对其进行重组，然后再将重组后的分组交给传输层。

20.1.2 作为数据报网络的因特网

因特网的网络层是一种分组交换网络。我们在第8章讨论了交换，我们说，交换可以分为三大类：电路交换、分组交换和报文交换。分组交换使用的是虚电路方法或数据报的方法。

在网络层，因特网使用数据报交换方法。它利用网络层所定义的全球地址从源端到目的端来路由分组。

> 因特网中的网络层交换是利用数据报分组交换方法来实现的。

20.1.3 作为无连接网络的因特网

分组传递能够利用面向连接的网络服务来实现，也可利用面向无连接的网络服务来实现。**在面向连接的服务**（connection oriented service）中，源端在发送一个分组之前首先与目的端建立一个连接。在连接建立后，分组就可以按顺序依次从相同的源端发送到相同的目的端。在这种情况下，分组间就存在一种关系，它们沿相同的路径按顺序发送，一个分组与其前后的分组在逻辑上连接在一起。当一个报文的所有分组都传递完毕后，连接就会终止。

在一个面向连接的协议中，当一个连接建立之后，对那些具有相同源和目的地址的分组序列，只会进行一次路由策略，交换机不会为每个单独的分组重复计算路由。这种类型的服务用在类似帧中继和ATM等虚电路分组交换方法中。

在无连接的服务（connectionless service）中，网络层协议独立地对待每个分组，而且每个分组与任何其他分组没有联系。一个报文中分组可能会也可能不会沿同样的路径到达其目的地。这种类型的服务用于数据报分组交换方法中。因特网在其网络层就采用这种类型的服务。

这样做的原因是：因特网是由许多异构网络所组成的，因此在没有提前弄清网络特性的前提下，在源端和目的端建立一个连接几乎是不可能的。

> 因特网的网络层通信是无连接的。

20.2 IPv4

网际协议第4版（Internet Protocol version4，IPv4）是TCP/IP协议使用的传输机制。图20.4表示了IPv4在TCP/IP协议族中的位置。

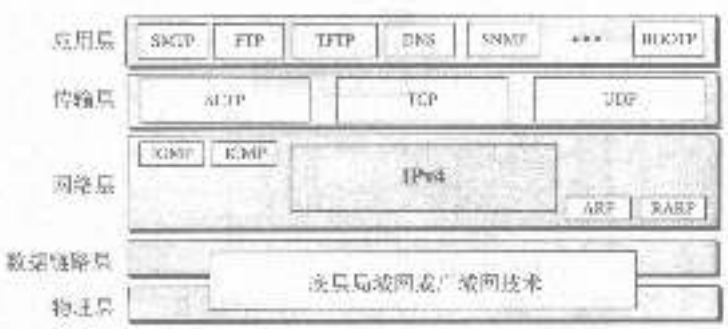


图20.4 IPv4在TCP/IP协议族中的位置

IPv4是一种不可靠的无连接数据报协议——**尽力传递**（best-effort delivery）。尽力传递这一词是指IPv4协议不提供差错控制或流量控制（除在头部差错检测外）。IPv4假定底层是不可靠的，因此尽力传输到目的地，但没有保证。

当可靠性很重要时，IPv4必须与一个可靠协议（如TCP）配合起来使用。理解尽力传递的

常见例子是邮局。邮局尽力传送邮件，但不是永远成功。如果一封未挂号的信遗失，那只能由发信人或可能的收信人来发现遗失或损坏的情况。

IPv4也是使用数据报的分组交换网的无连接协议（见第8章），这就是说每一分组独立进行处理，而每一分组使用不同的路由传送到目的端。这表明了如果一个源端向同一目的端发送许多数据报，那么这些数据报有可能不按顺序到达。有一些数据报也可能遗失，而有些在传输过程中可能受损坏。此时，IPv4要依靠高层的协议解决这些问题。

20.2.1 数据报

IPv4层的分组称为**数据报**（datagram），图20.5表示了IPv4数据报的格式。

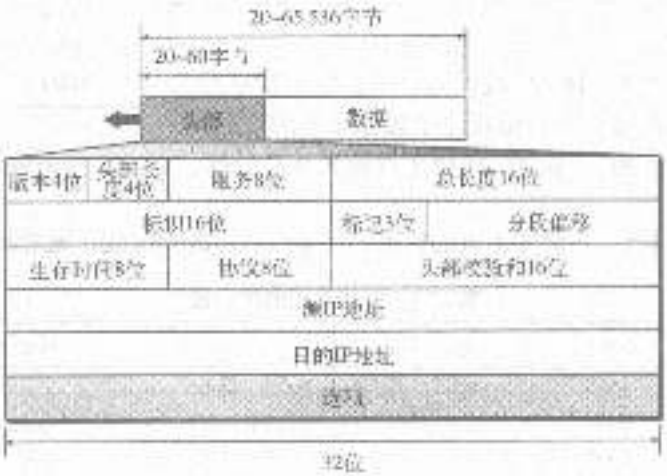


图20.5 IPv4数据报格式

数据报是一个可变长分组，由两个部分组成：头部和数据。头部长度可由20到60个字节组成，包含有与路由选择和传输有关的重要信息。习惯上在TCP/IP中都以4个字节表示头部，按顺序简要地介绍每个字段。

- 版本（VER）。这4位字段定义IPv4版本，目前版本是4，但在不久版本6（IPng）将取代版本4。这个字段向处理机所运行的IP软件指明该IPv4数据报是版本4格式。所有字段都要按版本4的协议来解释。如果计算机使用其他版本，则丢弃数据报而不是错误地解释。
- 头部长度（HLEN）。这4位字段以4字节字定义数据报头部的总长度。这个字段是必需的，因为头部长度是可变的（在20到60个字节之间）。当没有选项时，头部长度是20个字节，而这个字段的值是5（$5\times4=20$）。当选项字段为最大值时，这个字段的值是15（$15\times4=60$）。
- 服务。IETF已经改变了这个8位字段的解释与名称，这个字段以前称为服务类型（service type）现称为差分服务（differentiated service）。图20.6显示了这两种解释。

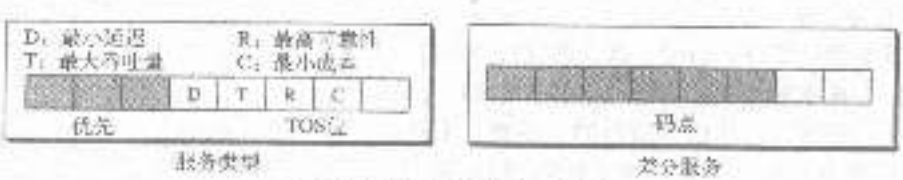


图20.6 服务类型或差分服务

1. 服务类型

在这种解释中，前面的3位称为优先位，而后面的4位称为服务类型（type of service，TOS），最后1位未使用。

a. 优先（Precedence）是3位的子字段，其值由0（二进制000）到7（二进制111）。优先子字段定义了在出现如拥塞等问题时，数据报的优先级。当路由器出现拥塞并必须丢弃一些数据报时，最低优先值的数据报将首先被丢弃。在因特网中，有一些数据报比另一些更为重要。例如，用于网络管理的数据报要比向一组人发送的可选信息的一些数据报紧迫得多和重要得多。

在版本4中未使用优先子字段。

表20.1　服务类型

TOS位	意　义
0000	正常（默认）
0001	最小成本
0010	最高可靠性
0100	最大吞吐量
1000	最小延迟

b. TOS位是一个4位的子字段，每一位都有其特殊的意义。虽然每一位可以是1或0，但在一个数据报中的这4个位中，只能有一位的值为1。表20.1给出了位模式及其解释。每次仅当一位置1，有5种不同的服务。

应用程序可以请求一个特定类型的服务，但某些应用的默认值，如表20.2所示。

表20.2　服务类型的默认值

协　议	TOS位	说　明	协　议	TOS位	说　明
ICMP	0000	正常	FTP（控制）	1000	最小延迟
BOOTP	0000	正常	TFTP	1000	最小延迟
NNTP	0001	最小成本	SMTP（命令）	1000	最小延迟
IGP	0010	最高可靠性	SMTP（数据）	0100	最大吞吐量
SNMP	0010	最高可靠性	DNS（UDP查询）	1000	最小延迟
TELNET	1000	最小延迟	DNS（TCP查询）	0000	正常
FTP（数据）	0100	最大吞吐量	DNS（区域）	0100	最大吞吐量

从表20.2中可清楚地看到交互式的活动、需要立即引起注意的活动与需要立即响应的活动都要求最小的延迟。发送块数据的活动要求最大吞吐量。管理活动要求最高的可靠性，后台活动要求最小成本。

2. 差分服务

在差分服务情况下，前6位组成码点（codepoint）子字段，而后2位不用。码点子字段可有两种使用方法：

a. 当最右边3位都是0时，最左边3位与服务类型解释中的优先位相同。换言之，它与原来的解释相同；

b. 当最右边3位不全为0时，则6位由因特网或本地机构按表20.3赋予的优先级定义64种服务。第一类包含32种服务类型，第二与第三类分别包含16种服务。第一类（0，2，4，…，62）由因特网工程任务组（IETF）分配。第二类（3，7，11，15，…，63）由本地组织机构使用。第三类（1，5，9，…，61）是临时的用作实验目的。注意：有些数不出现在类中，它们是第一类中的0到31，第二类中的32到47，第三类中的48到63。由于XXX000（包括0，8，16，24，32，40，48，56）

表20.3　码点的值

类	码　点	分配相关
1	XXXXX0	因特网
2	XXXX11	本地的
3	XXXX01	临时的或实验的

属于所有这3大类，这与TOS解释不相容，相反的，在这种分配方法中这些服务属于第一类。注意：这些分配还未最后定下来。

- 总长度。这个16位字段定义了一个以字节计的IPv4数据报的总长度（头部加上数据）。为了找到从上层来的数据长度，可以将总长度减去头部长度。将HLEN字段的值乘以4就可得到头部长度。

$$\text{数据长度}=\text{总长度}-\text{头部长度}$$

因为这个字段是16位长，因此IPv4数据报长度限制在65 535（$2^{16}-1$）字节，其中头部占20到60字节，剩下的是从上层来的数据。

总长度定义了包括头部在内的数据报的总长度。

一个长度为65 535字节的数据报在今天的技术看来很长，但在不久的将来数据报的长度可能会增大，因为底层技术允许具有更大的吞吐量（更高带宽的网络）。

当我们在下一节讨论分段时，将会看到某些网络不能将65 535字节的数据报封装成它们的帧。要通过这些网络就必须将这些数据报进行分段。

人们可能会问为什么需要这个字段。当一台机器（路由器或主机）接收到一个帧时，它除去头部和尾部并转发此数据报。那么为什么要使用一个并不需要的字段？回答是：在许多情况下，我们的确不需要这字段的值，但有些情况下，封装在一个帧中的并不仅仅是数据报。例如，以太网协议对能够封装在一个帧中的数据有最小值和最大值的限制（46到1 500个字节）。当数据报的长度小于46字节时，它必须增加一些填充字节才能满足这个要求。在这种情况下，当机器将数据分离出来时，它必须检查数据报的总长度，以便确定实际的数据长度和填充字节的长度（见图20.7）。

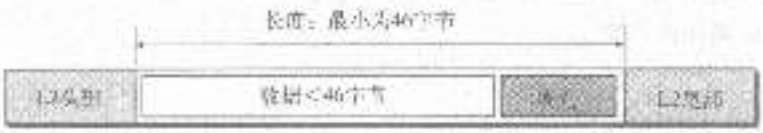


图20.7 一个小的数据报封装在以太网帧中

- 标识。这个字段用于分段中（在下一节讨论）。
- 标记。这个字段用于分段中（在下一节讨论）。
- 分段偏移。这个字段用于分段中（在下一节讨论）。
- 生存时间。一个数据报在它通过互联网时必须具有有限的寿命。这个字段最初设计时是打算保持一个时间戳，由每一个经过的路由器将此值减1。当时间戳的值变为0时，即丢弃该数据报。但是，对于这种方案，所有机器必须是同步的，并且还必须知道一个数据报从一个机器到另一个机器所花费的时间。现在，这个字段基本上是用来控制一个数据报所通过路由器的最大跳数（路由器数）。当源端发送数据报时，它在此字段存入一个数。这个数值约为任何两个主机之间的路由器数量的两倍。处理数据报的每一个路由器将此数值减1。当路由器接收到一个数据报时，它先将此字段的值减1，如果减1之后此字段的值为0，路由器就丢弃该数据报。

这个字段是必需的，因为因特网中的一些路由表可能会出故障。如果发生故障，一个数据报就可能在两个或更多的路由器之间传输很长时间，但永远传输不到目的端。这个字段限制了数据报的寿命。

这个字段的另一个用途是故意限制分组的行程。例如，如果源端想将分组限制在局域范

围内，就可将此字段置为1。当分组到达第一个路由器时，这个值就减为0。因而数据报就被丢弃了。

- 协议。这个8位字段定义了使用此IPv4层服务的高层协议，来自高层一些协议如TCP、UDP、ICMP和IGMP等的数据能够封装到IPv4数据报中。这个字段指明IPv4数据报必须传递到的最终的目的协议。换言之，由于IPv4协议携带来自高层不同协议的数据，这个字段的值对接收方的网络层了解数据属于哪个协议很有帮助（见图20.8）。

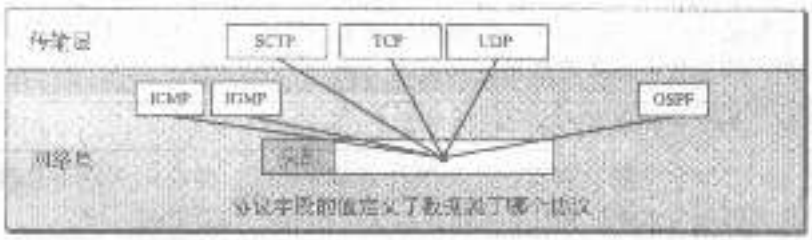


图20.8　协议字段和封装的数据

对于不同的高层协议，这个字段的取值如表20.4所示。

表20.4　协议值

值	协议
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

- 校验和。校验和的概念及其计算将在本章的后面讨论。
- 源地址。这个32位的字段定义了源端的IPv4地址，在IPv4数据报从源主机到目的主机传输期间，这个字段必须保持不变。
- 目的地址。这个32位的字段定义了目的端的IPv4地址，在IPv4数据报从源主机到目的主机传输期间，这个字段必须保持不变。

例20.1

到达的一个IPv4分组的前8位如下：

01000010

接收方是否应丢弃该分组？为什么？

解：

这个分组有错误，其中最左的4位是版本，它是正确的。下一个4位是1000，则表明它是一个无效长度（2×4=8）。而头部的最小字节数是20，因此，这个分组在传输过程中被损坏了。

例20.2

在一个IPv4分组中，头部长度字段的值用二进制表示为1000，试问这个分组携带的选项是几个字节？

解：

头部长度值是8，说明头部的总字节数为8×4=32，前面的20个字节是基本头部，后面的12个字节是选项。

例20.3

在一个IPv4分组中，头部长度字段的值是5，而总长度字段的值是0x0028，试问这个分组携带的数据是多少字节？

解：

头部长度字段值为5，就是说头部的总字节数为5×4=20（无选项）。总长度是40字节，也就是说，这个分组携带40−20=20个字节的数据。

例20.4

一个IPv4分组已到达，最前面几个十六进制数字如下：

0x45000028000100000102...

在丢弃这分组之前，它经历了多少跳？数据是属于上层的哪一个协议？

解：

为了求生存时间字段，我们跳过8个字节（16个十六进制数字）。生存时间字段是第9个字节，它的值是01，这就是说分组仅能跳一次。协议字段是下一个字节02，也就是说上层协议是IGMP（见表20.4）。

20.2.2 分段

一个数据报可以通过几个不同的网络进行传输。每一个路由器将它所接收的帧拆封成IPv4数据报，对它进行处理，然后再将它封装成另一个帧。接收到的帧的格式和长度取决于此帧刚刚经过的物理网络所使用的协议，发出去的帧的格式和长度则取决于此帧将要经过的物理网络所使用的协议。例如，如果一个路由器将一个局域网连接到一个广域网，那么它接收到的帧是一个局域网格式的帧，发出去的帧是广域网的格式。

最大传输单元（MTU）

每一个数据链路层协议都有其自己的帧格式，这格式中定义的一个字段是数据字段的最大长度。换言之，当数据报封装成帧时，该数据报的总长度必须小于这个最大数据长度。这是由网络所使用的硬件和软件给出的限制所定义的（见图20.9）。

MTU的值取决于物理网络协议。表20.5显示了不同协议的MTU值。

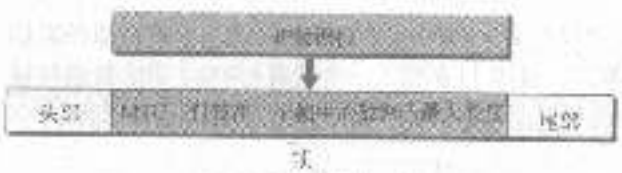


图20.9 最大传输单元（MTU）

表20.5 某些网络的MTU值

协 议	MTU
超级通道（Hyperchannel）	65 535
令牌环（16Mbps）	17 914
令牌环（4Mbps）	4 464
FDDI	4 352
以太网	1 500
X.25	576
PPP	296

为了使IPv4协议与物理网络无关，协议设计者决定使IPv4数据报的最大长度等于最大传输单元，其目前定义值是65 535字节。如果使用具有此值的MTU协议，那么可使传输更加有效。然而，对其他一些物理网络，我们就要将数据报进行分割，使其能够通过这些网络。这个过程称为分段（fragmentation）。

源端通常不对IPv4分组进行分段，传输层将数据分割成能适合在数据链路层所用IPv4的长度。

当对一个数据报进行分段时，每个分段都有其自己的头部，其中大部分的字段是重复的，但有些发生了变化。如果一个已分段的数据报遇到一个更小MTU的网络，那么已分段的数据报还可再进行分段。换句话说，数据报在到达最终的目的端之前，可能经过多次分段。

在IPv4中，数据报可能被主机或其路径中的任何路由器进行分段。然而，数据报的重组只能在目的主机上进行，因为每个分段都是一个独立的数据报，由于被分段的数据报可以沿着不同的路径传输，因此我们不能控制或保证一个分段的数据报会走哪一条路径，而属于同一数据报的所有分段最后都必须到达目的主机。所以，从逻辑上说应当在最终的目的端进行重组。一个更强的目的是在传输期间重组分组将会带来效率的降低。

当对数据报进行分段时，头部中的某些部分需要被复制到所有的分段中，如下一节中所见，选项字段可以被复制，也可以不被复制。对一个数据报进行分段的主机或路由器必须改变三个字段的值：标志、分段偏移和总长度。当然，不管是否进行分段，校验和的值总是重新计算的。

与分段相关的字段

与IPv4数据报的分段和重组相关的字段是：标识、标记和分段偏移。

- 标识。这个16位字段标识一个从源主机发出的数据报。当数据报离开源主机时，这个标识与源IPv4地址必须唯一地定义这个数据报。为了保证唯一性，IPv4协议使用一个计数器来标识数据报。当IPv4协议发送数据报时，就将该计数器的当前值复制到标识字段中，并将此计数器的值加1。只要此计数器保存在主存储器中，唯一性就得到保证。当数据报分段时，标识字段的值就复制到所有的分段中。换言之，所有的分段都有与原始的数据报相同的标识号。该标识号有助于在目的端重组数据报。目的端知道将所有具有相同标识值的分段必须重组成一个数据报。
- 标记。这是一个3位的字段。第一位保留为以后用。第二位称为“不分段位”，如果其值是1，则机器就不能将该数据报进行分段。如果无法将此数据报通过任何可用的物理网络对数据报进行传递，则它就丢弃该数据报，并向源主机发送一个ICMP差错报文（见第21章），如果其值为0，则根据需要对数据报进行分段。第三位称为“多分段位”，如果其值为1，则表示此数据报不是最后的分段，在该分段后还有更多的分段；如果其值为0，则表示它是最后一个或唯一的分段（见图20.10）。

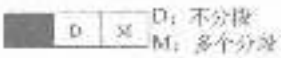


图20.10　标记字段

- 分段偏移。这个13位的字段表示这个分段在整个数据报中的相对位置。它是在原始数据报中的数据偏移量，以8字节为度量单位。图20.11说明了一个具有4 000字节的数据报被划分为三个分段。

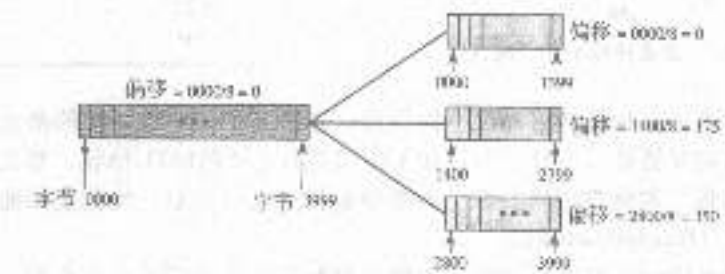


图20.11　分段示例

原始数据报的字节编号从0～3 999，第一个分段携带的数据是字节0～1 399，对于这个数据报，分段的偏移量为0/8=0；第二分段携带的数据是字节1 400～2 799，对于这个数据报其偏移量为1 400/8=175。最后，第三个分段携带的数据是字节2 800～3 999，其偏移量为2 800/8=350。

请记住，偏移量是以8字节为单位的。这样做是因为偏移字段的长度只有13位，不能用来表示一个大于8 191的字节数，因此，将数据报进行分段的主机或路由器必须这样选择每个分段的长度，即第一个字节编号能被8整除。

图20.12表示了图20.11中分段的扩展图。注意：所有分段的标识字段的值都相同。还要注意，除最后一个分段外，所有分段的标记中的值都是设置为多个分段。每一个分段的偏移字段的值也在图中表示。

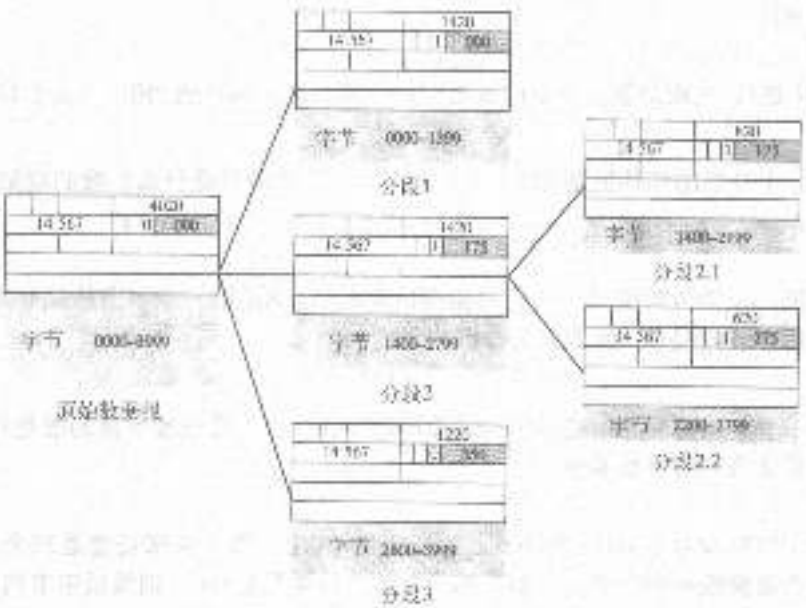


图20.12 分段的细节示例

图20.12还表示了一个分段本身再进行分段时会发生什么。在这种情况下，分段偏移值永远是相对于原始的数据报。例如，图20.12中的第二个分段又划分为两个长度分别为800字节和600字节的分段，但这些分段的偏移量所表示的位置都相对于原始的数据。

显然，即使每一个分段走不同的路径，并在到达时失序，最终目的主机也能用收到的这些分段（假定没有丢失）重组成原始数据报。所使用的策略如下：

1. 第一个分段的偏移字段的值为0；
2. 将第一个分段长度除以8，其结果就是第二个分段的偏移值；
3. 将第一个和第二个分段的总长度除以8，其结果为第三个分段的偏移值；
4. 继续以上过程，最后一个分段的多个分段位的值为0。

例20.5

到达的一个分组的*M*位的值是0，试问这是第一个分段还是中间的分段，还是最后的分段？我们是否知道这个分组已被分段？

解：

如果*M*位是0，这就是说不存在更多的分段，该分段是最后的一个分段。但是我们不能说原来的分组是否已被分段。没有分段的分组被认为是最后一个分段。

例20.6

到达的一个分组的*M*位的值是1，试问这是第一个分段还是中间的分段，还是最后的分段？我们是否知道这个分组已被分段？

解：

如果M位是1，这就意说至小还有一个分段，这个分段是第一个或中间的分段，而不是最后的分段。但我们不知道它是第一个分段还是中间分段，还需要有更多的信息（分段偏移值），见例20.7。

例20.7

到达的一个分组的M位的值是1，而偏移值是0，试问这是第一个分段还是最后的分段，或是中间的分段？

解：

因为M位是1，它或是第一个分段或是中间分段。由于偏移值为0，因此它是第一个分段。

例20.8

到达的一个分组的偏移值是100，试问第一个字节的编号是什么？我们能知道最后一个分段的编号吗？

解：

为了求第一个字节的编号，我们将偏移值乘8，这就说第一个字节的编号是800。但我们不能确定最后字节的编号，除非知道数据的长度。

例20.9

到达的一个分组的偏移值是100，而HLEN字段值为5，总长度字段的值是100，试问第一个字节和最后字节的编号是多少？

解：

第一个字节的编号是100×8=800，总长度是100个字节，头部长度是20个字节（5×4），这就是说这个数据报有80个字节。如果第一个字节的编号是800，则最后字节的编号是879。

20.2.3 校验和

在第10章中，我们已讨论过校验和总的思想及其计算方法。在IPv4分组中的校验和的计算方法与之相同。首先，将校验和字段置为0。然后，将整个头部划分为16位的部分，并将各部分相加。将计算结果（和）取反码，插入到校验和字段中。

IPv4分组中的校验和只对头部进行，而不在数据部分进行。这有两点很充足的理由。首先，所有将数据封装在IPv4数据报中的高层协议中，都有覆盖整个分组的校验和。因此，IPv4数据报的校验和就不必校验所封装的数据部分。其次，每经过一个路由器，IPv4数据报的头部就要改变一次，但数据部分不改变。这样，校验和只对发生变化的部分进行校验，如果包含数据部分，则每一路由器必须重复计算整个分组的校验和，这就表示每一个路由器要花费更多的处理时间。

例20.10

图20.13表示了一个不没有选项的IPv4首部计算校验和。首部分成多个16位的部分。将所有这些部分相加，再将得到的和取反码。将其结果插入到校验和字段中。



图20.13 IPv4中校验和的计算

20.2.4 选项

IPv4数据报的头部由两部分组成：固定部分与可变

部分。固定部分的长度是20个字节，已在前一节讨论过。可变部分由若干选项组成，它最长可达40个字节。

选项正如其名字所表含的，它对每个数据报来说并不是必需的。这些选项用于网络测试和调试。虽然选项并非IPv4数据报必需的部分，但选项处理却是IPv4软件的必需部分，这表示所有的标准必须能够处理选项，如果这些选项出现在头部中。

每个选项详细的讨论已超出本书的范围，在图20.14中给出选项的分类，并对每项进行了简要地说明。

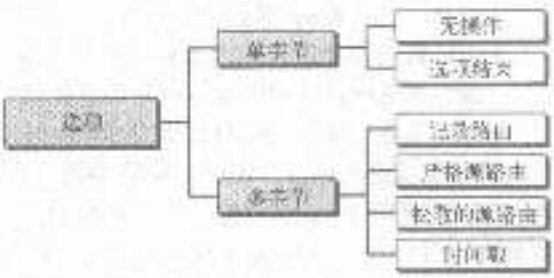


图20.14 IPv4中选项的种类

无操作

无操作选项（no-operation option）是一个1字节选项，用做选项之间的填充符。

选项结束

选项结束选项（end-of-option option）也是一个1字节，用于选项字段结束时的填充。但是，它只能用做最后一个选项。

记录路由

一个记录路由选项（record route option）是用来记录处理数据报的因特网路由器。它可列出多达9个路由器地址，它用于调试和管理目的。

严格源路由

源端使用严格源路由选项（strict source route option），用来预先确定数据报在因特网中传送时的路由。由于几个原因，源端确定路由是很有用的。发送方可以选择一个具有特定服务类型的路由，如最小延迟或最大吞吐量。此外，出于发送端的目的，它还可选择一条更加安全或更加可靠的路由。例如，发送方可以选择某条路由使得数据报不经过对手的网络。

如果一个数据报指明了一个严格的源路由，该数据报就必须经过选项中定义的所有路由器。若路由器的IPv4地址未列入到数据报中，则该数据报就一定不能通过此路由器。若一个数据报通过一个未列入的路由器，则该路由器将数据报丢弃并发出差错报文。

松散的源路由

松散的源路由选项（loose source route option）与严格源路由选项相似，但更加宽松，必须访问表中的路由器，但数据报还可以访问其他的路由器。

时间戳

时间戳选项（timestap option）用来记录路由器处理数据报的时间。时间是世界时，从午夜开始以毫秒计。知道一个数据报的处理时间将有助于用户和管理者对因特网上的各个路由器的行为进行跟踪。我们能够估计一个数据报从一个路由器到另一个路由器所需的时间。我们说估计，是因为虽然所有的路由器都使用世界时，但它们的本地时钟可能没有进行同步。

20.3 IPv6

TCP/IP协议族中的网络层协议现在是IPv4（因特网协议，版本4）。IPv4提供在因特网中的系统之间的主机到主机的通信。虽然IPv4设计得很好，但自从20世纪70年代IPv4问世以来，数据通信已经有了很大的发展，IPv4有一些缺点（列出如下），这使得它对飞速发展的因特网

有些不适应。

- 虽然有许多近期的解决方案，如子网化、无类寻址和NAT，但在因特网中地址耗尽还依旧是一个长期的问题。
- 因特网必须能适应实时音频和视频传输。这种类型的传输要求最小延迟的策略和预留资源，而这些在IPv4的设计中并没有提供。
- 对于某些应用，因特网必须能够对数据进行加密和鉴别。IPv4不提供数据的加密和鉴别。

为了克服这些缺点，IPv6（网际协议，版本6，Internetworking Protocol，version 6），也称为下一代网际协议（IPng，Internetworking Protocol，next generation）被提出，并已成为标准。在IPv6中，网际协议修改了很多，以便适应因特网不可预见的增长。IP地址格式和长度以及分组的格式都改变了。相关的一些协议，如ICMP也修改了。网络层的其他一些协议，如ARP、RARP和IGMP，则被取消或包含在ICMPv6协议之中（见第21章）。路由选择协议，如RIP和OSPF（见第22章）也进行了少量的修改以适应变化。通信专家预计，IPv6及其相关协议将很快地取代当前的IP版本。在本节中，我先讨论IPv6，然后研究从IPv4过渡到IPv6的一些策略。

对IPv6的采用进展很慢，理由是最初的动机为它的发展，IPv4地址消耗尽，已经被短期策略修改，如无类寻址和NAT，但是，因特网快速应用的发展和新的服务的出现，如移动IP，IP电话和IP移动电话最终要求用IPv6全部代替IPv4。

20.3.1　优点

与IPv4相比，下一代IP或IPv6具有如下的优点：

- 更大的地址空间。正如第19章中讨论的那样，IPv6地址是128位长。与32位地址相比，其地址空间增加了很多（2^{96}）。
- 更好的头部格式。IPv6使用了新的头部格式，其选项与基本头部分开，如果需要，可将选项插入到基本头部与上层数据之间。这就简化和加速了路由选择过程，因为大多数的选项不需要由路由器检查。
- 新的选项。IPv6有一些新的选项来实现附加的功能。
- 允许扩充。如果新的技术或应用需要时，IPv6允许协议进行扩充。
- 支持资源分配。在IPv6中，服务类型字段被取消了，但增加了一种机制（称为流标号）使得源端可以请求对分组进行特殊的处理。这种机制可用来支持像实时音频和视频的通信量。
- 支持更多的安全性。在IPv6中的加密和鉴别选项提供了分组的保密性和完整性。

20.3.2　分组格式

IPv6分组格式如图20.15所示，每个分组由必须要有的基本头部和紧接其后的有效载荷所组成。有效载荷由两部分组成：可选的扩展头部和来自上层的数据。基本头部占用40字节，而扩展头部和来自上层的数据可以包含多达65 535字节的信息。



图20.15　IPv6数据报头部和有效载荷

基本头部

图20.16说明了具有8个字段的基本头部（base header），这些字段如下：

- 版本。这个4位字段定义了IPv6的版本号。对IPv6，其值为6。
- 优先级。这个4位字段定义了当发生通信量拥塞时的分组的优先级。我们将在后面讨论这个字段。

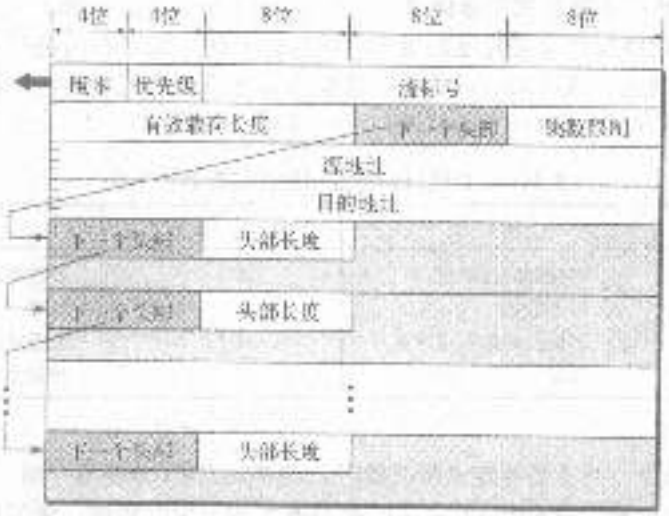


图20.16 IPv6数据报的格式

- 流标号。流标号（flow lab）是一个3字节（24位）的字段，它用来对特殊的数据流提供专门处理。我们将在后面讨论这个字段。
- 有效载荷长度。这2字节的有效载荷长度字段定义了包括基本头部在内的IP数据报的总长度。
- 下一个头部。下一个头部是一个8位的字段，它定义了数据报中跟随在基本头部之后的头部。下一个头部或者是IP所使用的可选的扩展头部，或者是上层协议的（如UDP或TCP）头部。每一个扩展头部也包含这个字段。表20.6给出了下一个头部的值。注意：在版本4中这个字段称为协议。
- 跳数限制。这个8位的跳数限制（hop limit）字段与IPv4中的TTL（生存时间）字段所起的作用是一样的。
- 源地址。该字段是一个16字节（128位）的因特网地址，它是用来识别数据报的原始源端。
- 目的地址。该字段是一个16字节（128位）的因特网地址，通常用来识别数据报的最终目的地。然而，如果使用了源路由选择，那么该字段就包含了下一个路由器的地址。

表20.6 IPv6下一个头部的代码

代 码	下一个头部	代 码	下一个头部
0	逐跳选项	44	分段
2	ICMP	50	加密的安全有效载荷
6	TCP	51	鉴别
17	UDP	59	空（没有下一个头部）
43	选路由选项	60	目的站选项

优先级

IPv6分组的优先级字段定义从相同源端发出的每一个分组相对于其他分组的优先级。例如，如果由于拥塞原因两个连续的数据报中必须丢弃一个，则具有较低的分组优先级（packet

priority）的数据报将被丢弃。IPv6将通信量划分为两大类：可进行拥塞控制的和不可进行拥塞控制的。

可进行拥塞控制的通信量　如果源端在出现拥塞时能够适应这种情况使其通信量下降，则此通信量称为可进行拥塞控制的通信量（congestion-controlled traffic）。例如，使用滑动窗口协议的TCP协议能够很容易地对通信量进行响应。众所周知，在可进行拥塞控制的通信量中，分组可以延迟到达，或丢失，或不按序接收。可进行拥塞控制的数据被指定为从0到7的优先级，如表20.7所示。优先级0是最低的，优先级7是最高的。

表20.7　可进行拥塞控制的通信量的优先级

优先级	意义	优先级	意义
0	未指明的通信量	4	关注的批量数据通信量
1	后台数据	5	保留
2	不关注的数据通信量	6	交互式通信量
3	保留	7	控制通信量

优先级描述如下：

- 未指明的通信量。当不需要定义优先级时，就将优先级0分配给分组。
- 后台数据。这个组（优先级1）定义的数据通常是在后台传送，新闻的传递是一个很好的例子。
- 不关注的数据通信量。如果用户并不等待（关注）要接收的数据，则分组将被指定优先级2。电子邮件就属于这类。一个用户将电子邮件发送给另一个用户，但接收者并不知道电子邮件马上到达。此外，电子邮件通常在转发之前要经过存储，有少量延迟并不要紧。
- 关注的批量数据通信量。当传送批量数据而用户正在等待（关注）接收这个数据（可能会有些延迟时），则这种传送数据协议就使用优先级4。FTP和HTTP属于这组。
- 交互式通信量。在这一组中，需要与用户交互的协议，如TELNET被指定为第二高的优先级（6）。
- 控制通信量。控制通信量的优先级最高（7）。路由选择协议（如OSPF和RIP）以及管理协议（如SNMP）都使用这个优先级。

不可进行拥塞控制的通信量　这指的是期望最小延迟的通信量类型，丢弃分组是不希望出现的。在大多数情况下重传也是不可能的。换言之，源端不能使自己适应拥塞。实时音频和视频是这类通信量很好的例子。

优先级号8到15被指定给不可进行拥塞控制的通信量（noncongestion-controlled traffic）。虽然还没有特殊的标准指定给这类数据，但优先级的指定是基于当丢弃一些分组时接收到的数据的质量会有多大的影响来确定的。包含较小的冗余度的数据（如低保真音频和和视频）给予较高的优先级（15），包含较小冗余度的数据（如高保真音频或视频）可以给予较低的优先级（8），见表20.8。

表20.8　不可进行拥塞控制通信量的优先级

优先级	意义
8	具有最大冗余度的数据
…	……
15	具有最小冗余度的数据

流标号

从特定源端向特定目的端发送的分组序列，如果需要路由器的特殊处理，则称为分组流。源地址与流标号的值组合唯一地定义了一个分组流。

对路由器来说，一个流是共享某些特性的分组序列。例如，经过相同路径，使用相同的资源，具有相同安全性，等等。支持流标号处理的路由器有一个流标号表，这个表为每一个活动的流标号设置一个项目。每一个项目定义相应的流标号所需的服务。当路由器收到一个分组时，它就从其流标号表中找出在分组中定义的流标号值所对应的项目。但注意，流标号本身并不给流标号表项目提供信息，信息是由其他方法提供的，如逐跳选项或其他协议。

在最简单形式中，流标号可用来加速路由器对分组的处理。当路由器收到一个分组时，它不用查找路由表并用路由选择算法确定下一跳的地址，而是可以很容易地在流标号表中找到下一跳的地址。

在更复杂的形式中，流标号可用来支持实时音频和视频的传输，特别是数字形式的实时音频或视频，需要高带宽、大缓存、长处理时间等资源。进程可以事先对这些资源进行预留，以保证实时数据不会因资源不够而被延迟。使用实时数据和预留这些资源需要IPv6外的一些其他协议，如实时协议（RTP）和资源预留协议（RSVP）。

为了有效地使用流标号，已定义了三个原则：

1. 流标号由源主机指定给分组，这个标号是在1到$2^{24}-1$之间的随机数。当已存在的流仍处于活跃状态时，源端一定不能给新的流重复使用一个用过的流标号；

2. 如果主机不支持流标号，它就将这个字段置为0。如果路由器支持流标号，它就简单地忽略它；

3. 所有属于同一个流的分组必须具有相同的源地址、相同的目的地址、相同的优先级和相同的选项。

IPv4头部和IPv6头部的比较

表20.9比较了IPv4和IPv6分组的头部。

表20.9 IPv4和IPv6分组头部的比较

比 较
1. IPv6取消了头部长度字段，因为在此版本中头部长度是固定的
2. IPv6取消了服务类型字段，优先级和流标号字段合并一起取代服务类型字段的功能
3. IPv6取消了总长度字段，替代的是有效载荷长度字段
4. 在IPv6中的基本头部中取消了标识、标记和偏移字段，这些都包含在分段扩展头部中
5. 在IPv6中，将TTL字段称为跳数限制字段
6. 协议字段被称为下一个头部字段
7. 头部校验和取消了，因为校验和由上层协议提供，因此这一过不需要了
8. IPv4的选项字段在IPv6中被实现为扩展头部

20.3.3 扩展头部

基本头部的长度是固定的40字节，但是要使IP数据报有更多的功能，在基本头部的后面还可增加多达6个扩展头部（extension header）。这些头部中的许多都是IPv4的选项。图20.17给出了扩展头部的6种类型。



图20.17 扩展头部的类型

逐跳选项

当源端需要将信息传送给数据报经过

的所有路由器时，就需要使用逐跳选项（hop-by-hop option）。到目前为止，只定义了三种选项：Pad1、PadN和特大有效载荷（jumbo payload）。Pad1选项是1字节长，其作用是为了对齐。PadN在概念上与Pad1相似，其差别是当两个或更多字节需要对齐时，就要使用PadN。特大有效载荷选项用来定义比65 535字节更大的有效载荷。

源路由选择

源路由选择扩展头部将IPv4严格的源路由和松散的源路由的概念结合并在一起。

分段

分段（fragmentation）的概念与IPv4中的一样。但分段发生的地方却不同。在IPv4中，如果数据报长度超过数据报所要经过的网络的MTU时，源端或路由器就需要将其进行分段。在IPv6中，只有原始的源端才能进行分段。源端必须使用路径MUT发现技术（path MTU discovery technique）来找出在其路径上的任何网络所支持的最小MTU，然后源端利用得到的这个知识进行分段。

鉴别

鉴别（authentication）扩展头部有双重目的：它证实报文发送者并保证数据的完整性。当我们在第31章中讨论网络安全性时，再讨论这个扩展头部。

加密的安全有效载荷

加密的安全有效载荷（encrypted security payload，ESP）是一个扩展，它要保密性并能防止窃听。我们将在第31章中讨论这个扩展头部。

目的端选项

目的端选项（destination option）用于当源端需要将信息仅传递给目的端。中间的各路由器则不允许读取这些信息。

IPv4选项和IPv6扩展头部的比较

表20.10　比较了IPv4中的选项和IPv6中的扩展头部。

表20.10　IPv4选项和IPv6选项的比较

比　较
1. 在IPv4中为必要作者选项结束选项被替换成IPv6中的Pad1和PadN选项
2. 在IPv6中，没有记录路由选项，因为它未被使用
3. 在IPv6中，时间戳选项没有实现，因为它未被使用
4. 在IPv6中，源路由选项被称为路由扩展头部
5. 在IPv4中的基本头部的分段字段已经移到IPv6中的分段扩展头部
6. 在IPv6中的鉴别扩展头部是新的
7. 在IPv6中，加密的安全有效载荷扩展头部是新的

20.4　IPv4到IPv6的过渡

因为因特网上的系统非常多，所以从IPv4过渡到IPv6不能突然发生。要使每一个在因特网中的系统从IPv4过渡到IPv6，需要花费相当长的时间。这种过渡必须平滑的，以防止IPv4和IPv6系统间出现任何问题。IETF已经设计了三种策略来使这一过渡时期更加平滑（见图20.18）。



图20.18　三种过渡策略

20.4.1 双协议栈

IETF推荐所有的主机在完全过渡到第6版之前，使用一个双协议栈（dual stack），换言之，一个站点同时运行IPv4和IPv6，直到整个因特网使用IPv6。图20.19给出了双协议栈的配置示意图。

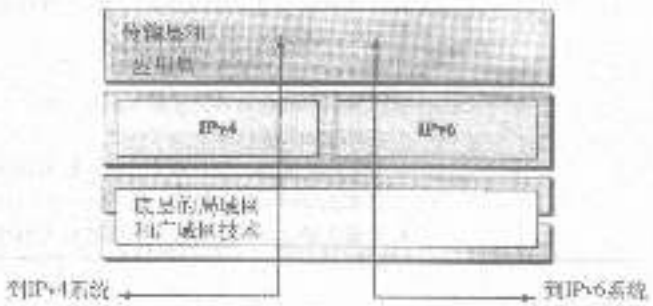


图20.19 双协议栈

当把分组发送到目的端时，为了确定使用哪个版本，主机要向DNS进行查询。如果DNS返回一个IPv4地址，那么源主机就发送一个IPv4分组；如果返回一个IPv6地址，就发送一个IPv6分组。

20.4.2 隧道技术

当两台使用IPv6的计算机要进行相互通信，但其分组要通过使用IPv4的区域时，就要使用隧道技术（tunneling）这种策略。要经过该区域，该分组必须具有IPv4地址。因此，当进入这种区域时，IPv6分组要封装成IPv4分组，而当分组离开该区域时，再去掉这个封装。这就好像IPv6分组进入隧道一端，而在另一端流出来。为了更清楚地说明利用IPv4分组携带IPv6分组，其协议的值设置成41。隧道技术如图20.20所示。



图20.20 隧道策略

20.4.3 头部转换

当因特网中绝大部分已经过渡到IPv6，但一些系统仍然使用IPv4时，需要使用头部转换（header translation）。发送方想使用IPv6，但接收方不能识别IPv6。这种情况下使用隧道技术无法工作，因为分组必须是IPv4的格式才能被接收方识别。在此情况下，头部格式必须通过头部转换彻底改变，IPv6的头部就转换成IPv4的头部（见图20.21）。



图20.21 头部转换

头部转换使用被映射的地址将IPv6地址转换成IPv4地址。表20.11列出了将IPv6分组头部转换为IPv4分组头部所使用的一些规则。

表20.11　头部转换

头部转换过程
1. 用掩取最右边32位的方法，将IPv6被映射的地址改变为IPv4地址
2. 将IPv6优先级字段的值丢弃
3. 将IPv4的服务类型置为0
4. 计算IPv4的校验和，并插入到相应的字段中
5. 忽略IPv6的流标号
6. 兼容的扩展头部要转换成选项，插入到IPv4的头部中
7. 计算出IPv4头部的长度，将其插入到相应的字段中
8. 计算出IPv4分组的总长度，将其插入到相应的字段中

推荐读物

作为深入讨论与本章有关的主题，推荐下列读物和网站。在本书末列出的方括号[……]中给出参考资料。

读物

[For06]的第8章、[Ste94]的第3章、[PD03]的4.1节，[Sta04]的第18章和[Tan03]的5.6节有关于IPv4的讨论。[For06]的第27章和[Los04]有关于IPv6的讨论。

网站

- www.ietf.org/rfc.html 有关请求评论（RFC）的信息。

请求评论

IPv4的讨论可从下列请求评论找到：

760，781，791，815，1025，1063，1071，1141，1190，1191，1624，2113

IPv6的讨论可以下列请求评论找到：

1365，1550，1678，1680，1682，1683，1686，1688，1726，1752，1826，1883，1884，1886，1887，1955，2080，2373，2452，2463，2465，2466，2472，2492，2545，2590

本章小结

- IPv4是一个不可靠的无连接协议，负责源端到目的端的传递。
- 在IPv4层的分组称为数据报。一个数据报包括一个头部（20字节到60字节）和数据。数据报的最大长度是65 535字节。
- MTU是数据链路协议能够封装的最大字节数。MTU随着协议的不同而不同。
- 分段就是将一个数据报划分成若干个更小的单元，从而适应数据链路层协议的MTU。
- IPv4数据报的头部包括一个固定的20个字节的部分与一个可变的选项部分。其最大值为40个字节。
- IPv4头部中的选项部分是用于测试和调试。
- IPv4的6个选项各有其特定的功能。它们是：用于选项之间为了对齐而使用的填充符、用于选项字段结束时的填充、记录数据报所经过的路由、由发送方选择的强制性路由、选择必须经过某些路由器的路由以及记录在各路由器的处理时间。
- IPv6是网际协议的最新版本。它有128位的地址空间、修改过的头部格式、新的选项、允许扩展、支持资源分配以及增加安全措施。

- IPv6数据报由基本头部和有效载荷组成。
- 扩展头部给IPv6数据报增加了功能。
- 从IPv4过渡到IPv6所使用的三个策略：双协议栈、隧道技术和头部转换。

习题

复习题

1. 数据链路层帧传递和网络层分组传递有什么不同？
2. 无连接服务和面向连接服务有什么不同？
3. 定义分段并解释IPv4和IPv6协议为什么需要对有些分组要进行分段？在这个意义上这两个协议有没有差别？
4. 解释在IPv4协议中校验和的计算与验证。校验和计算是对IPv4分组的哪些部分进行的？为什么？如果有选项出现，是否在校验和中包括选项？
5. 解释在IPv4中需要选项，并列出本章所提到的所有选项。简要说明每个选项。
6. 比较与对照IPv4和IPv6基本头部的字段，并用一个表来说明每个字段出现或不出现。
7. IPv4和IPv6两者假定一些分组具有不同优先级，解释每个协议如何处理这个问题。
8. 比较与对照IPv4和IPv6扩展头部的选项，并做一个表来说明每个选项出现或不出现。
9. 解释在IPv6头部取消校验和的理由。
10. 列出从IPv4到IPv6过渡的三种策略。解释过渡过程中隧道策略和双协议栈策略有什么不同，每种策略应在何种情况下使用？

练习题

11. IPv4头部中的哪一个字段在经过每一个路由器时都不一样？
12. 如果总长度为1 200字节而其中的1 176字节是来自高层的数据，试计算HLEN的值。
13. 表20.5列出了不同协议的MTU，MTU的范围从296到65 535。使用大的MTU有什么好处？使用小的MTU有什么好处？
14. 已知一个被分段的数据报的分段偏移是120，如何确定第一个和最后一个字节的编号？
15. 一个分组头部长度的值能否小于5？什么时候它正好等于5？
16. 一个IPv4数据报中的HLEN值是7，试问有多少选项字节？
17. 一个IPv4数据报的选项字段长度的值是20字节，HLEN的值是多少？其二进制的值是多少？
18. 一个IPv4数据报的总长度字段的值是3G，而头部长度字段的值是5，试问该分组携带的数据是多少字节？
19. 一个IPv4数据报携带的数据共1 024字节，如果没有选项信息，则头部长度字段的值是多少？
20. 一台主机发送100个数据报给其他主机，如果第一个数据报的标识号为1 024，则最后一个数据报的标识号是多少（IPv4）？
21. 一个IPv4数据报在到达时，其分段偏移为0而M位（多个分段）是0，它是一个分段吗？
22. 一个IPv4数据报在到达时，其分段偏移为100，在该分段的数据前，源端已发送了多少个分段？
23. 一个IPv4数据报在到达时，其头部有如下的信息（十六进制）：
    0x45 00 00 54 00 03 58 50 20 06 00 00 7C 4E 03 02 B4 0E 0F 02
    a. 分组是否损坏？　　b. 有无任何选项？
    c. 分组是否分段？　　d. 数据的长度是多少？

e. 分组能够经过多少路由器？ f. 分组的标识号是多少？

g. 它的服务类型是什么？

24. 一个IPv4数据报M位为0，而HLEN是5，总的长度值是200，而偏移值是200。在该数据报中第一个字节和最后一个字节的编号是多少？这是第一个分段还是最后一个分段，或中间分段？

探究题

25. 寻找为何IPv6中有两个安全协议（AH和ESP）。

第21章 地址映射、差错报告和多播

在第20章中，我们讨论了网络层的主要协议，网际协议（IP）。IP被设计为尽力传递协议，而没有如流控制和差错控制等一些特性。它是使用逻辑寻址的主机到主机的协议。为了使IP协议满足今天的网际互联的更多需求，还需要其他协议的帮助。

我们仍需要一些协议来创建物理地址和逻辑地址之间的映射。IP分组使用逻辑（主机到主机）地址，而这些分组需要封装成帧，前需要物理地址（节点到节点）。我们将看到为此而设计的一个称为地址解析协议（Address Resolution Protocol，ARP）的协议，有时我们还需要逆映射（物理地址到逻辑地址）。例如，为引导一个无盘网络或给主机租用IP地址的目的，设计了三个协议：RARP、BOOTP和DHCP。

网际协议缺少流控制和差错控制，从而产生另一个协议，ICMP，它提供差错告警。报告在网络上或目的端发生拥塞和差错的类型。

IP最初是为单播传送而设计的，即一个源端和一个目的端。由于因特网对多播传送的极大需求，也就是一个源端和多个目的端。因此，IGMP提供了IP的一种多播能力。

本章将在某些方面详细讨论ARP、RARP、BOOTP、DHCP和IGMP等协议。也讨论ICMPv6，当IPv6是运行时它将也运行。IMCPv6将ARP、ICMP和IGMP合并在一个协议中。

21.1 地址映射

互联网是由一些物理网络和一些网际互联设备（如路由器）所组成。从源主机发送的分组在到达目的主机之前可能要经过许多不同的物理网络。在网络级上，主机和路由器用它们的逻辑（IP）地址进行标识。

但是，分组都要通过物理网络到达这些主机和路由器。在物理级上，主机和路由器用它们物理地址标识。**物理地址**（physical address）就是本地地址，它的管辖范围是本地网络。物理地址在本地范围内是唯一的，但在全局上并不必如此，所以称为物理地址，是由于物理地址通常（并非永远）是用硬件实现的。物理地址的例子是以太网协议中48位的MAC地址，它被写入在主机或路由器的网卡（NIC）上。

物理地址和逻辑地址是两种不同的标识符。这两个我们都需要，因为一个物理网络，如以太网可以在同一时间在网络层使用两个不同的协议，如IP和IPX（Novell）。同样的，在网络层的分组，如IP，也可以通过不同的物理网络，如以太网和LocalTalk（Apple）传输。

这就是说将分组传递到一台主机或路由器需要两级地址：逻辑地址和物理地址。我们需要能将一个逻辑地址映射成为它相应的物理地址，反过来的映射也需要的。这可使用静态和动态映射。

静态映射（static mapping）是创建一个表，它将一个逻辑地址与物理地址联系起来。这个表存储在网络上的每个机器上。例如，每一个机器知道其他机器的IP地址，但却不知道其物理地址。可通过参考表得知物理地址。这样做有其些局限性，因为物理地址可能因以下原因而发生变化：

1. 一个机器可能会更换网卡，结果得到了一个新的物理地址；
2. 在某些局域网中，如LocalTalk，每当计算机加电时，其物理地址都要改变一次。

3. 移动的计算机可以从一个物理网络转移到另一个物理网络，这就引起物理地址的改变。要完成这些变化，静态映射必须周期性地改变，这给网络增加了非常大的开销。

在动态映射（dynamic mapping）中，每当一个机器知道两个地址（逻辑地址和物理地址）中的一个时，就可使用协议将另一个求出。

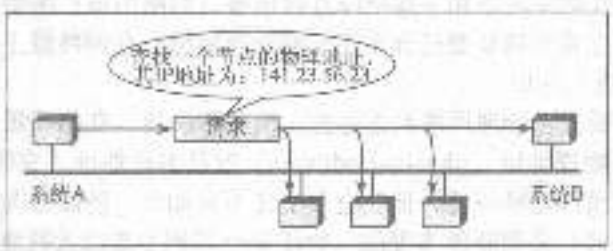
21.1.1　逻辑地址到物理地址的映射：ARP

任何时候，当主机或路由器有数据报要发送到另一个主机或路由器时，它必须有接收方的逻辑（IP）地址。如果发送方是主机，它可从DNS求得逻辑（IP）地址（见第25章）；如果发送方是路由器，它可从路由选择表求得（见第22章）。但是，IP数据报必须封装成帧才能通过物理网络，这就是说，发送方必须有接收方的物理地址。主机或路由器发送一个ARP查询分组，该分组包括发送方的物理地址和IP地址以及接收方的IP地址。因为发送方不知道接收方的物理地址，查询就在网络上广播（见图21.1）。

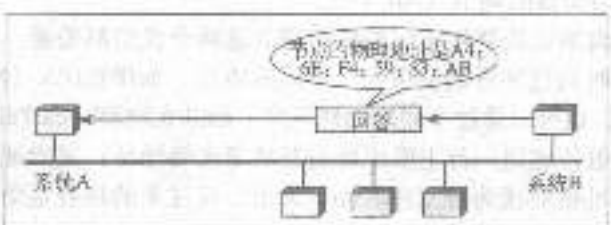
网络上的每一个主机或路由器都接收和处理这个ARP查询分组，但只有预期的接收者才能识别它的IP地址，并发回ARP响应分组。这个分组使用接收到的查询分组中的物理地址直接用单播发送给查询者。

在图21.1a中，左边系统（A）有一个分组要传递给IP地址为141.23.56.23的另一个系统（B）。系统A需要将分组传送给它的数据链路层进行实际的传递，但它不知道接收方的物理地址。它使用ARP的服务，请求ARP协议发送一个广播ARP请求分组，以查询IP地址为141.23.56.23的系统的物理地址。

在该物理网络上每一个系统都接收到此分组，但只有系统B才回答，如图21.1b所示。系统B发送一个包含它的物理地址的ARP回答分组。现在系统A可使用接收到的物理地址来发送到该目的地的所有分组。



a) ARP请求用广播发送



b) ARP回答用单播发送

图21.1　ARP操作

高速缓存器

如果系统A对发送到系统B的每一个分组都需要广播一个ARP请求，那么对ARP协议的使用就是低效率的，它可能有对自己IP分组的广播。为了解决这一问题，ARP协议可使用高速缓存器。因为一个系统通常发送多个分组到同一目的地，接收到ARP回答的系统将它的映射

存储在高速缓存器中，保持20分钟到30分钟（除非高速缓存已满）。在以后发送ARP请求之前，系统先在它的高速缓存器中检查是否可找到它的映射。

分组格式

图21.2表示了ARP分组格式。



图21.2 ARP分组

ARP分组具有下列字段：

- **硬件类型**。这是一个16位字段，用来定义运行ARP的网络类型。每一个局域网基于其类型被指定一个整数。例如，以太网是类型1，ARP可使用在任何物理网络上。
- **协议类型**。这是一个16位字段，用来定义协议的类型。例如，对于IPv4协议，这个字段的值是0800_{16}。ARP可用于任何高层协议。
- **硬件长度**。这是一个8位字段，用来定义以字节为单位的物理地址的长度。例如，对于以太网，这个值是6。
- **协议长度**。这是一个8位字段，用来定义以字节为单位的逻辑地址的长度。例如，对于IPv4协议，这个值是4。
- **操作**。这是一个16位字段，用来定义分组的类型。已定义了两种类型：ARP请求（1）和ARP回答（2）。
- **发送方硬件地址**。这是一个可变长的字段，用来定义发送方的物理地址的长度。例如，以太网这个字段是6字节。
- **发送方协议地址**。这是一个可变长的字段，用来定义发送方的逻辑（例如，IP）地址的长度。对于IP协议，这个字段是4字节。
- **目标硬件地址**。这是一个可变长的字段，用来定义目标的物理地址的长度。例如，对以太网，这个字段是6字节。对ARP请求报文，这个字段是全0，因为发送方不知道目标的物理地址。
- **目标协议地址**。这是一个可变长的字段，用来定义目标的逻辑地址（例如，IP地址）的长度。对于IPv4协议，这个字段是4字节。

封装

ARP分组是直接封装在数据链路帧中。例如，在图21.3中，ARP分组封装在以太网的帧中。

注意：类型字段指出了此帧所携带的数据是ARP分组。

操作

让我们讨论在一个典型的互联网上，ARP是如何工作的。首先，我们描述所包含的一些步骤，然后讨论主机或路由器需要使用ARP的四种情况。在ARP过程中，需要进行如下几个步骤：



图21.3　ARP分组的封装

1. 发送方知道目标的IP地址。我们将简要地考察发送方如何得到这个地址；

2. IP请求ARP协议产生一个ARP请求报文，填入发送方的物理地址、发送方的IP地址以及目标的IP地址。目标的物理地址字段则填入0；

3. 将这个报文发送给数据链路层，在这个层它被封装成帧，使用发送方的物理地址为源地址，而将物理广播地址作为目的地址；

4. 每一个主机和路由器都接收到这个帧。因为这个帧包含了广播目的地址，所以所有的站点都将此报文送交ARP。除了目标机器外，所有的机器都丢弃该分组。目标机器识别这个IP地址；

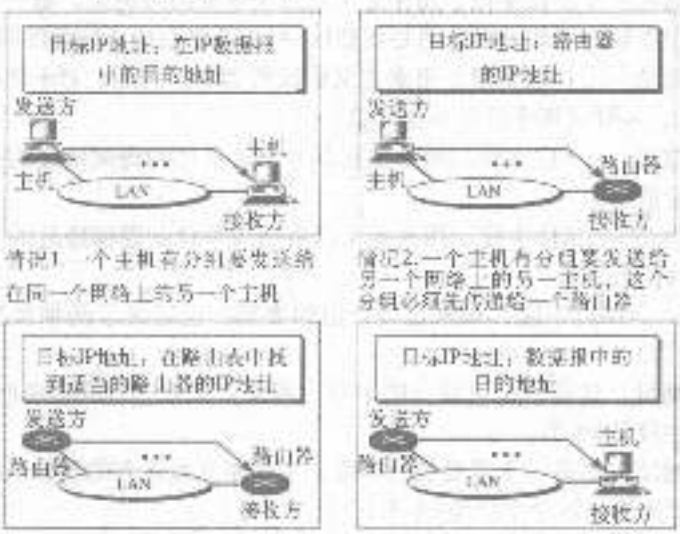
5. 目标机器用ARP回答报文进行应答。此回答报文包含它的物理地址。报文使用单播；

6. 发送方接收到这个回答报文，它现在知道了目标机器的物理地址；

7. 携带发送给目标机器数据的IP数据报现在封装成帧，用单播发送给目的端。

四种不同的情况

下列是可使用ARP服务的四种不同的情况（见图21.4）：



情况1　一个主机有分组要发送给在同一个网络上另一个主机

情况2　一个主机有分组要发送给另一个网络上的另一主机，这个分组必须先传送给一个路由器

情况3　一个路由器接收到一个分组，要将该分组发送给在另一个网络上的主机。这个分组必须先传送给适当的路由器

情况4　一个路由器接收到一个分组，要将该分组发送给在同一个网络上的主机

图21.4　使用ARP的四种情况

1. 发送方是一个主机，它希望将分组发送给同一个网络上的另一个主机。在这种情况下，必须将物理地址映射为逻辑地址，并将该逻辑地址作为数据报头部的目的IP地址；

2. 发送方是一个主机，它希望将分组发送给另一个网络上的另一主机。在这种情况下，

该主机查找它的路由表，找出这个目的地下一个跳（路由器）的IP地址。如果该主机没有路由表，它就要查找默认路由表的IP地址。这个路由器的IP地址就是必须映射为一个物理地址的那个逻辑地址；

3. 发送方是一个路由器，它已经接收到了一个数据报，要将该数据报发送给另一个网络上的一个主机。它先检查它的路由表，找出下一个路由器的IP地址。这个下一路由器的IP地址就是必须映射为物理地址的那个逻辑地址；

4. 发送方是一个路由器，它已经收到了一个数据报，要将该数据报发送给同一网络上的一个主机。数据报的目的IP地址就是必须映射为物理地址的那个逻辑地址。

ARP请求报文是广播发送；ARP回答报文是单播发送。

例21.1

一个主机的IP地址为130.23.43.20，物理地址为B2：34：55：10：22：10。它有一个分组想要发送给另一个主机，其IP地址为130.23.43.25，物理地址为A4：6E：F4：59：83：AB（第一个主机并不知道该物理地址），两个主机在同一个网络上。试说明ARP请求与回答分组如何封装在以太网帧中。

解：

图21.5说明了ARP请求与回答分组。注意：此时ARP数据字段是28个字节，而单个地址不适合用4字节表示界限，这就是我们为什么不以4字节界限表示这些地址。

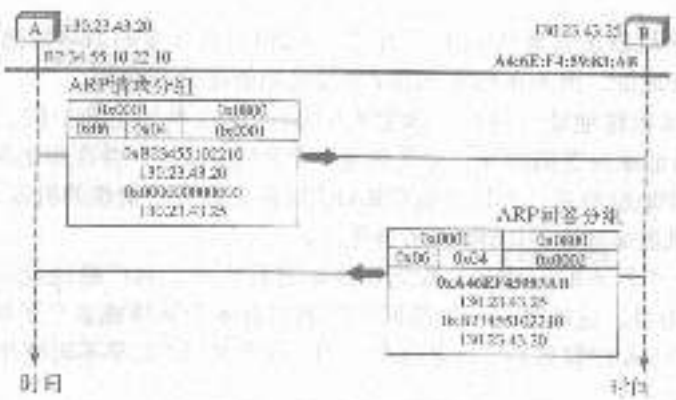


图21.5 例21.1的ARP请求与回答分组

代理ARP

有一种称为代理ARP的技术可用来产生子网化的效果。代理ARP（proxy ARP）是可以代表一组主机的ARP，每当运行代理ARP的路由器接收到一个寻找这些主机中的一个主机的IP地址的ARP请求时，路由器就发送一个ARP回答，宣布它自己的硬件（物理）地址。当这个路由器收到真正的IP分组后，它就将这个分组发送给相应的主机或路由器。

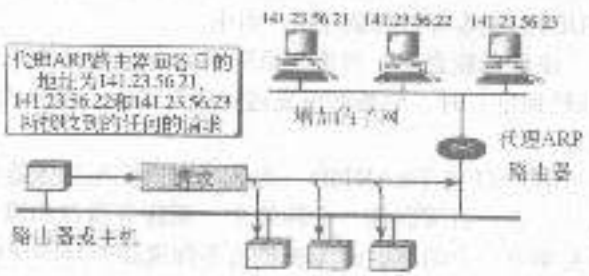


图21.6 代理ARP

让我们给出一个实例。在图21.6

中，安装在右边的主机只回答目标IP地址为141.23.56.23的ARP请求。

但是，管理员可能需要创建一个子网而不改变整个系统来识别出子网的地址。一个解决问题的方法是增加一个运行代理ARP的路由器。在这种情况下，这个路由器代表所有安装在子网上的主机。当路由器接收到一个ARP请求时，如果其目标IP地址与它所代表的主机之一的地址相匹配（141.23.56.21，141.23.56.22和141.23.56.23），则它发送一个ARP回答，并宣布其硬件地址作为目标硬件地址。当该路由器收到IP分组，它就将该分组发送给相应的主机。

21.1.2 物理地址映射到逻辑地址：RARP、BOOTP和DHCP

有两种可能的场合，当一个主机知道它的物理地址但不知道其逻辑地址：

1. 无盘站点正在被引导，站点可以通过检查其接口得到它的物理地址，但不知道它的逻辑地址；

2. 一个组织机构没有足够的IP地址分配给每一个站点，只能按需分配。站点可发送它的物理地址并请求延续一个短暂的时间。

RARP

反向地址解析协议（Reverse Address Resolution Protocol，RARP）是为仅知道物理地址的机器寻找它的逻辑地址而设计的。每一个主机或路由器都被指定一个或多个逻辑（IP）地址，这些地址是唯一的，并与机器的物理（硬件）地址无关。要创建一个IP数据报，主机或路由器就要知道它自己的IP地址。一个机器的IP地址通常可从存储在磁盘文件中的配置文件中读出。

但是，一个无盘机器通常是从ROM引导的，ROM只有少量的引导信息。ROM是由厂商安装的。它不包括IP地址，因为在网络上的IP地址是由管理员分配的。

机器可以得到其物理地址（例如，读它的NIC），这在本地是唯一的。然后就可以使用RARP协议从物理地址求得逻辑地址。先是创建一个RARP请求，并在本地网络上广播。在本地网络上知道所有IP地址的另一个机器就用RARP回答来响应。请求的机器必须运行RARP客户程序，而响应的机器必须运行RARP服务器程序。

RARP协议有一个严重的问题：在数据链路层进行广播，其广播地址在以太网中是全1，而且不能通过网络边界。这就是说，如果网络管理员有多个网络或多个子网，它需要为每个网络或子网指定一个RARP服务器。正是这个原因，使得RARP几乎不再使用，而被BOOTP和DHCP两个协议取代。

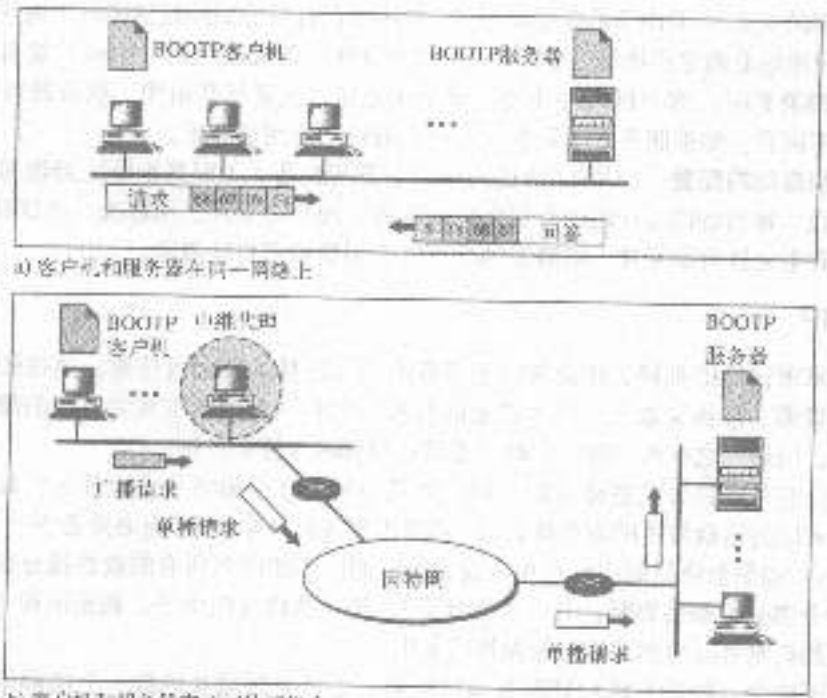
BOOTP

引导程序协议（Bootrap Protocol，BOOTP）是一种客户机/服务器协议，设计这个协议是提供物理地址到逻辑地址的映射。BOOTP是一个应用层协议，网络管理员可以将客户机和服务器放置在同一网络上或不同的网络上，如图21.7所示。BOOTP报文被封装在UDP分组中，而UDP分组本身被封装在IP分组中。

读者可能会问，当客户机既不知道它自己的IP地址（源地址）也不知道服务器的IP地址（目的地址）时，它是如何发送IP数据报的。客户机简单地使用全0作为源地址，全1作为目的地址。

BOOTP优于RARP的一点是，客户机与服务器都是应用层的进程，当在其他应用层进程中时，客户机可以在一个网络上，而服务器在相隔多个其他网络的另一个网络上。但是，必须要解决一个问题，因为客户机不知道服务器的IP地址，所以BOOTP请求是广播的。而广播的数据报不可能通过任何路由器。解决这个问题必须要有一个中间媒介，可以用一个主机

（或者配置为在应用层操作的一个路由器）作为中继，在这种情况下，该主机称为中继代理（relay agent）。中继代理知道BOOTP服务器的单播地址，当它接收到这种类型的分组时，它在单播的数据报中封装报文并向BOOTP服务器发送请求。任何一台路由器都可以路由携带单播目的地址的分组并传送到BOOTP服务器。BOOTP服务器知道来自中继代理的报文，因为在请求报文中有一个字段定义了中继代理的IP地址。在接收到回答之后，中继代理向BOOTP客户机转发它。



a) 客户机和服务器在同一网络上

b) 客户机和服务器在不同的网络上

图21.7 在同一网络上和不同网络上的BOOTP客户机和服务器

DHCP

BOOTP不是一个动态配置协议（dynamic configuration protocol）。当客户机请求其IP地址时，BOOTP服务器查找一个表，寻找与客户机物理地址相匹配的IP地址。这意味着客户机的物理地址与IP地址的绑定必须已经存在，绑定是预先确定的。

但是，如果一个主机从一个物理网络移动到另一个物理网络时，情况又如何？如果主机想要一个临时的IP地址又如何？BOOTP不能处理这些问题，因为表中的物理地址和IP地址的绑定在网络管理员改动之前，是静态的和固定的。BOOTP是一个静态配置协议。

动态主机配置协议（Dynamic Host Configuration Protocol，DHCP）已设计出来，提供静态的和动态的地址配置，可以是人工的或自动的。

> DHCP提供可以是人工的或自动的静态或动态的地址配置。

静态地址配置 这种DHCP的性能与BOOTP相同，它是与BOOTP向后兼容的，这就表示一个运行BOOTP客户机的主机可以向一个DHCP服务器请求一个静态地址。DHCP服务器有一个数据库静态地绑定物理地址和IP地址。

动态地址配置 DHCP有第二个数据库，它拥有一个可用的IP地址池。这个第二个数据库使DHCP成为动态的。当DHCP客户机请求一个临时的IP地址时，DHCP服务器就查找可用

(即未使用的) IP地址池，然后指定一个在可协商的期间内有效的IP地址。

当DHCP客户机向DHCP服务器发送请求时，服务器首先检查它的静态数据库，如果数据库中存在所请求的物理地址的项目，则返回该客户的永久IP地址，反之，如果静态数据库中没有该项目，服务器就从可用的IP地址池中选择一个IP地址，并将这个地址指定给该用户，然后将该项目加到动态数据库中。

当主机从一个网络移到另一个网络，或连续到一个网络后又断开连接时（如同一个用户和服务提供者的关系），DHCP是需要动态的，DHCP在有限的期间提供临时IP地址。

从可用IP地址池指定的地址是临时地址，DHCP服务器发出租用（lease）是对特定期间而言的。但租用期到时，客户机或停止使用这个IP地址，或更新其租用。服务器对这个更新可选择同意或不同意。如果服务器不同意，客户机就停止使用该地址。

人工的和自动的配置 BOOTP协议的一个主要问题是人工配置IP地址到物理地址的映射表。这就是说，每当物理或IP地址有变化时，网络管理员必须人工地修改。而DHCP协议则既允许人工配置也允许自动配置。即静态地址可人工创建也可自动创建。

21.2 ICMP

如第20章所讨论的那样，IP提供了不可靠的和无连接的数据报传递。这样的设计是为了有效地利用资源。IP协议是一个尽力传递的服务。它将一个数据报从它原始的源端传递到最终的目的端。但是，它有两个缺点：缺少差错控制和缺少辅助机制。

IP协议没有差错报告或差错纠正机制，如果出现某些差错将会发生什么？如果路由器因找不到一个可以到达最终目的端的路由器，或者因生存时间字段是0而必须丢弃一个数据报时，将会发生什么？如果最终目的主机在预先设定的时间内不能收到所有的数据报分段，因此将一个数据报的全部分段都丢弃时，又会发生什么？这是一些情况的例子，都是出现了差错而IP协议没有内在的机制可以通知发送该数据报的主机。

IP协议还缺少一种为主机和管理查询的机制，主机有时需要确定一个路由器或另一个主机是否是活跃的，有时网络管理员需要从另一个主机或路由器得到信息。

因特网控制报文协议（Internet Control Message Procotol，ICMP）就是为了弥补上述两个缺点而设计的，它是配合IP协议使用。

21.2.1 报文类型

ICMP报文分为两大类：**差错报告报文**（error-reporting message）和**查询报文**（query message）。

差错报告报文向路由器或主机（目的端）报告在处理一个IP数据报时可能碰到的一些问题。

查询报文是成对出现的，它帮助主机或网络管理员从一个路由器或另一个主机得到特定的信息。例如，节点能够发现它们的邻站。此外，主机能够发现和知道在它们的网络上的一些路由器的情况，而一些路由器能帮助一个节点改变报文的路由。

21.2.2 报文格式

ICMP报文有一个8字节的头部和一个可变长的数据部分。虽然每一种报文类型的头部的格式都是不同的，但最前面的4个字节对所有类型都是相同的，如图21.8所示。第一个字段是ICMP的类型，它定义报文的类型。代码字段指定了发送此特定报文类型的原因。最后一个共同的字段是校验和字段（在本章后面讨论）。头部其余部分对每种报文类型都是特定的。

在差错报文的数据部分所携带的信息可找出引起差错的原始分组。在查询报文的数据部分携带了基于查询类型的额外信息。

21.2.3 差错报告



图21.8 ICMP报文一般格式

ICMP的主要责任之一就是差错报告。虽然现代技术已经制造出很可靠的传输介质，但差错还是存在的，因而必须进行处理。如在第20章中讨论的那样，IP是不可靠的协议。这就是说，IP是不考虑差错校验和差错控制的。ICMP就是为弥补这个缺点而设计的。然而，ICMP不能纠正差错，它只是报告差错。差错纠正留给高层协议去完成。差错报文总是发送给原始的源端，因为在数据报中关于路由唯一可用的信息就是源IP地址和目的IP地址。ICMP使用源IP地址将差错报文发送给数据报的源端（发送方）。

> ICMP总是向原始的发方报告差错报文。

共有5种差错可处理：目的端不可达、源端抑制、时间超时、参数问题及重定向，见图21.9。



图21.9 差错报告报文

关于ICMP差错报文有下列要点：

- 对于携带ICMP差错报文的数据报，不再产生ICMP差错报文；
- 对于分段的数据报文，如果不是第一个分段则不产生ICMP差错报文；
- 对于多播地址的数据报文，不产生ICMP差错报文；
- 具有特殊地址，如127.0.0.0或0.0.0.0，不产生ICMP差错报文。

注意：所有差错报文都包括一个数据部分。而这个数据部分包括原始数据报的头部加上数据报中的前8个字节的数据。加上原始数据报中的头部就可给出原始的源端，它接收差错报文，这就是关于数据报本身的信息。要包括8个字节是因为这前8个字节提供了关于端口号（UDP和TCP）和序列号（TCP）的信息。这些我们将在第23章讨论UDP和TCP协议时看到。这个信息是需要的，这样源端可以将差错情况通知给这些协议（TCP或UDP）。ICMP形成差错分组，然后再封装成IP数据报（见图21.10）。



图21.10 差错报文的数据字段的内容

目的端不可达

当路由器不能够给数据报找到路由或主机不能传递数据报时，就丢弃这个数据报。然后，这个路由器或主机就发回目的端不可达报文（destination-unreachable message）给发出该数据报的源主机。注意，目的端不可达报文或者由路由器或者由目的主机创建。

源端抑制

IP协议是一个无连接协议，在产生数据报的源端、转发数据报的路由器以及处理数据报的目的主机之间，并没有通信关系。这种由于缺乏通信引起的一个问题就是缺乏流量控制。IP在协议中没有嵌入流量控制机制。在运行IP时，缺乏流量控制会产生的一个主要问题是拥塞。源主机从来不知道某些路由器或目的主机是否已经被过多的数据报造成超载。源主机也从来不知道它产生的数据报是否太快，以致路由器来不及转发或目的主机来不及处理。

缺乏流量控制可能会在路由器或目的主机中产生拥塞。路由器或主机中的队列长度（缓存）是有限的，这种队列是为到来的数据报等待转发（对于路由器）或等待处理（对于主机）。如果数据报的接收速率比它们被转发或处理的速率快得多，则队列就会溢出。在这个情况下，路由器或主机别无选择，只能将某些数据报丢弃。在ICMP的源端抑制报文（source-quench message）就是为了给IP增加一种流量控制而设计的。当路由器或主机因拥塞而丢弃数据报时，它就向数据报的发送方发送源端抑制报文。这个报文有两个目的。第一，它通知源端数据报已被丢弃；第二，它警告源端，在路径中的某处出现拥塞，因而源端必须放慢（抑制）发送过程。

时间超时

时间超时报文（time-exceeded message）是在二种情况下产生的。其一，如我们将在第22章中看到的，路由器使用路由表找出必须接收此分组的下一跳（下一个路由器）。如果一个或多个路由表中出现差错，那么某一个分组就可能在一个回路或循环中从一个路由器到下一个路由器，或无休止地通过一系列的路由器。正如第20章中见到的，每一数据报都有生存时间的字段控制这种情况。当数据报通过路由器时，这个字段的值减1。收到数据报的路由器在发现这个字段的值为0时，就丢弃这个数据报。但是，当丢弃这样数据报时，就要由路由器向源端发送一个时间超时报文。其二，当组成一个报文的所有分段未能在某一时限内到达主机时，也要产生时间超时报文。

参数问题

当数据报在因特网上传输时，在其头部中出现的任何二义性都可能会产生严重的问题。如果路由器或目的主机发现了这种二义性或在数据报的某个字段中缺少某个值，它就丢弃这个数据报，并向源端发送参数问题报文（parameter-problem message）。

重定向

当路由器要将分组转发到另一个网络时，它必须知道下一个适当的路由器的IP地址。如果发送方是主机，情况也是这样的。路由器和主机都必须有一个路由表，以便找出路由器或下一个路由器的地址。正如我们将在第22章看到的，路由器还要参与路由选择更新过程，并且其更新被认为是周期完成的。路由选择是动态的。

但是，为了提高效率，主机都不参与路由选择更新过程，因为在因特网上主机的数量比路由器要多出很多。动态地更新主机的路由表会产生不可接受的通信量。主机通常使用静态路由选择。当主机开始联网工作时，其路由表中项目的个数是很有限的。它通常只知道默认路由器这一个路由器的IP地址。因此，当主机向另一个网络发送数据报时，就将此数据报发给了这个错误的路由器。在这种情况下，收到此数据报的路由器会将该数据报转发给正确的路由器。但是，要更新主机中的路由表，它就要向主机发送重定向报文。重定向的概念如图21.11所示。主机A想要向主机B发送数据报。路由器R2显然是最有效的路由选择，但是主机A没有选择路由器R2。数据报发送到路由器R1而不是R2。R1在查找路由表后发现分组应当走R2。它把分组发送到R2，并同时向主机A发送重定向报文。这时主机A的路由表就被更新了。

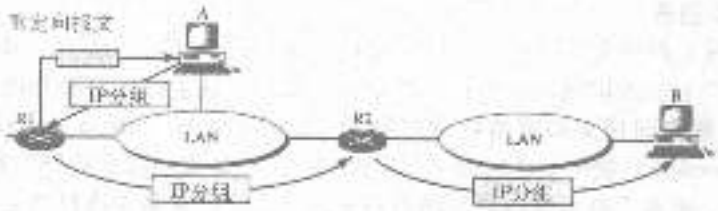


图21.11 重定向概念

21.2.4 查询

除了差错报告外，ICMP还能对某些网络问题进行诊断。这是通过使用由4对不同报文组成的查询报文来完成的，如图21.12所示。在这种类型的ICMP报文中，一个节点发送出报文，然后由目的节点用特定的格式进行回答。查询报文封装在IP分组中，然后再将IP分组封装在数据链路层的帧中。然而，在这种情况下，原始IP字节没有包含在报文中，如图21.13所示。



图21.12 查询报文

回送请求和回答

回送请求报文（echo-request message）和回送回答报文（echo-reply message）是为诊断目的而设计的。网络管理员和用户都可以使用这对报文发现网络问题，回送请求和回送回答组合起来确定了两个系统（主机或路由器）是否彼此能通信。回送请求报文和回送回答报文可用来确定是否在IP级能够通信。因为ICMP报文被封装成IP数据报，发送回送请求的机器在收到回送回答报文时，就证明了在发送方和接收方之间能够使用IP数据报进行通信。此外，还证明了中间的一些路由器能够接收、处理和转发IP数据报。今天，大多数系统都提供*ping*命令，它可以创建一系列（不仅仅是一个）回送请求或回送回答报文，提供统计信息。我们在本章的末尾将看到这个程序的用法。



图21.13 封装ICMP查询报文

时间戳请求和回答

两个机器（主机或路由器）可使用时间戳请求和时间戳回答报文（timestamp request and timestamp reply messages）来确定IP数据报在两个机器之间往返所需的时间。它也可用做两个机器之间的时钟同步。

地址掩码请求和回答

主机可能知道它的IP地址，但可能不知道相应的掩码。例如，主机知道它的IP地址是159.31.17.24，但可能不知道相应的掩码/24。为了求得它的掩码，主机应向局域网上的路由器发送地址掩码请求报文（address-mask request message）。如果主机知道该路由器的地址，它就直接向路由器发送该请求，如果它不知道地址，它就广播该报文。路由器接收到地址掩码请求报文后，用地址掩码回答报文（address-mask reply message）进行响应。向主机提供所需要的掩码。将这个掩码应用到完整的IP地址上，就可求得子网的掩码。

路由器询问和通告

如同我们在重定向报文的讨论中那样，主机如果想将数据发送给另一网络上的主机，就需要知道连接到它自己的网络上的路由器的地址。另外，该主机还需要知道这些路由器是否正常工作。**路由器询问报文和路由器通知报文**（router-solicitation message and router-advertisement message）就用于这种情况。主机可将路由器询问报文进行广播（或多播），收到询问报文的一个或多个路由器就使用路由器通告报文广播其路由选择信息。即使在没有主机询问时，路由器也可周期性地发送路由器通告报文。注意：路由器在发送通告报文时，它不仅通告自己的存在，而通告了它所知道的所有在这个网络上的路由器。

校验和

在第10章中，我们知道了校验和的概念与思想。在ICMP中，校验和的计算包括了整个报文（头部和数据）。

例21.2

图21.14表示了对一个简单的回送请求报文的校验和计算的实例。我们随机选定标识符为1，而序列号为9。将报文划分成16位（2字节）的字，将这些字相加，然后将其和取反。现在发送方就可将此值置在校验和字段。

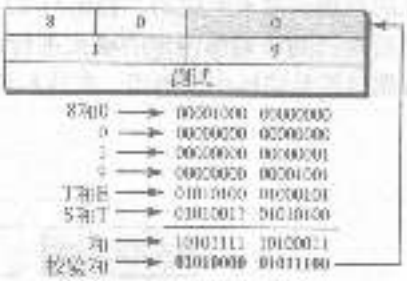


图 21.14

21.2.5 调试工具

有许多工具可用于调试因特网。我们可以确定主机或路由器是否正常工作，还可以跟踪分组的路由。我们介绍ICMP所采用两个调试工具：*ping*和*traceroute*。在对多个协议讨论之后，我们将在后面几章中介绍更多的调试工具。

Ping程序

我们可用*ping*程序来确定一个主机是否正常工作与做出响应。此处观察*ping*程序如何用于ICMP分组。

源主机发送ICMP回送请求报文（类型：8，代码：0），如果目标机器是正常工作，则它用ICMP回送回答报文响应。*ping*程序在回送请求报文和回送回答报文中设置标识符字段，并以序列号0开始，每发送一个新的报文该序列号加1。注意：*ping*程序能计算出往返时间，*ping*程序在报文数据部分中插入发送时间。往返时间（RRT）的值是收到回答报文的时间减去发送请求报文时间的值。

例21.3

我们用*ping*程序测试服务器fhda.edu，其结果如下：

```
$ ping fhda.edu
PING fhda.edu (153.18.8.1)  56 (84)  bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0   ttl=62   time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1   ttl=62   time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2   ttl=62   time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3   ttl=62   time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4   ttl=62   time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5   ttl=62   time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6   ttl=62   time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7   ttl=62   time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8   ttl=62   time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9   ttl=62   time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10  ttl=62   time=1.98 ms

--- fhda.edu ping 统计 ---
已发送11个分组，11个分组都经过接收，丢弃分组为0%，共计时间为10 103ms
往返最小时间/平均时间/最大时间=1.899/1.955/2.041ms
```

*ping*程序发送报文的序列号从0开始，每次探测给出这次的往返时间。封装了ICMP报文的IP数据报的TTL（生存时间）字段已被置为62，这就是说该分组的传输不能超过62跳。一开始*ping*程序定义数据部分为56个字节，IP数据报总长度为84字节。这是显然的，这是因为我们要增加ICMP头部8字节和IP头部20字节，所以其结果是84字节。但是注意：每次探测*ping*程序定义的字节个数为64，这是ICMP分组的总长度（56+8）。如果不用中断键（例如，ctrl+c）停止*ping*程序，则它一直继续下去，当程序被中断后，它打印出检查的统计结果，它告诉我们发送的分组数以及接收到的分组数，总的时间和往返最小、最大平均时间。有些系统还可以打印出更多的信息。

Traceroute程序

UNIX中的*traceroute*程序或Windows中的*tracert*程序可用于追踪一个分组从源端到目的端所经过的路由。*Traceroute*程序的应用与第20章中的IP数据报的松散的路由选项和严格路由选项相类似。本章我们将这个程序和ICMP数据报一道使用，程序巧妙地利用时间超时与目的端不可达两种ICMP报文找出分组经过的路由。这是一个UDP提供服务的应用程序（见第23章）。让我们用图21.15表示*traceroute*程序的思想。

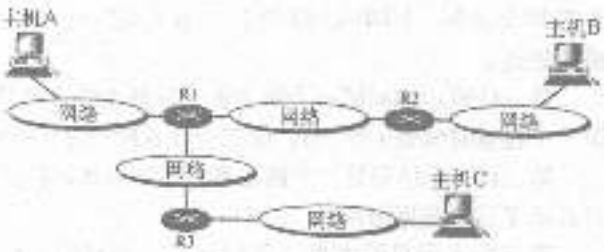


图21.15 *traceroute*程序的操作

给定图中的网络拓扑结构，我们知道主机A向主机B发送一个分组要经过路由器R1和路由器R2。但是，多数情况下我们并不知道网络拓扑结构，主机A到主机B可能存在多条路由，*traceroute*程序利用ICMP报文和IP分组的TTL（生存时间）字段找出路由。

1. *traceroute*程序用下列步骤求得路由器R1的地址和主机A到路由器R1的往返的时间。

a. 主机A上的*traceroute*程序利用UDP协议发送一个分组到目的主机B，该报文被封装在TTL字段值为1的IP分组中。程序记住发送分组的时间。

b. 路由器R1接收到分组并将TTL的值减1成为0，然后路由器R1丢弃该分组（由于TTL字段的值为0），但是路由器R1发送一个时间超时ICMP报文（类型：11，代码：0）表示TTL的值为0而该分组已被丢弃。

c. *traceroute*程序接收到该ICMP报文，并利用封装了ICMP报文的IP分组的目的地址求得路由器R1的IP地址。程序也记住分组到达的时间，这个时间与a步记下的时间差就是往返时间。

*traceroute*程序重复步骤a到c三次求得一个较好的平均往返时间。第一次往返时间可能会比第二次或第三次时间长，因为第一次需用ARP程序去查找R1的物理地址，而第二次和第三次往返时，ARP在它的高速缓存中有该地址。

2. *traceroute*程序重复上面的步骤求得R2的地址以及主机A到R2的往返时间。但是，在这一步，TTL字段设置为2。因此，R1转发这个报文，而R2丢弃它并发送时间超时ICMP报文。

3. *traceroute*程序重复步骤2求得主机B的地址以及主机A到主机B往返时间。当主机B接收到分组时，TTL值减1。但主机B不丢弃该报文，因为它已到达了它的最终的目的地。ICMP报文如何发送回到主机A？*traceroute*程序在此处采用一种不寻常的策略。UDP分组的目的端口被设置UDP协议不支持的端口。当主机B接收到该分组，它找不到接收该传递的应用程序，它

丢弃该分组并发送一个ICMP目的端不可达报文（类型：3，代码：3）到主机A。注意：这种情形在路由器R1和R2不会发生，因为路由器不检查UDP的头部。*traceroute*程序记录到达的IP数据报的目的地址以及记住往返时间。接收到代码为3的目的端不可达的ICMP报文就表示已求得全部路由，此时不需要再发送分组。

例21.4

我们用*traceroute*程序求计算机voyager.deanza.edu到服务器fhda.edu的路由，下面表示其结果：

```
$ traceroute fhda.edu
traceroute to fhda.edu    (153.18.8.1), 30 hops max, 38 byte packets
 1  Dcore.fhda.edu    (153.18.31.254)    0.995 ms    0.899 ms    0.878 ms
 2  Dbackup.fhda.edu  (153.18.251.4)     1.039 ms    1.064 ms    1.083 ms
 3  tiptoe.fhda.edu   (153.18.8.1)       1.797 ms    1.642 ms    1.757 ms
```

命令后无序列的行表示了目的端是153.18.8.1，TTL值是30跳，该分组包含38个字节：IP头部20个字节、UDP头部8个字节和应用数据10个字节。*traceroute*程序使用应用数据存储分组的轨迹。

第一行表示访问第一个路由器，该路由器名称是Dcore.fhda.edu，其IP地址为135.18.31.254，第一个往返时间是0.995ms，第二个往返时间是0.899ms，第三个往返时间是0.878ms。

第二行表示访问第二个路由器，该路由器名称是Dbackup.fhda.edu，其IP地址为153.18.251.4，也显示了三个往返时间。

第三行表示目的主机。我们知道这是目的主机，因为没有更多的行。目的主机是服务器fhda.edu，但它的名称是tiptoe.fhda.edu，其IP地址为153.18.8.1，也显示了三个往返时间。

例21.5

这个例我们追踪了一个较长的路由，到xerox.com的路由。

```
$ traceroute xerox.com
traceroute to xerox.com (13.1.64.93), 30 hops max, 38 byte packets
 1  Dcore.fhda.edu    (153.18.31.254)    0.622 ms    0.891 ms    0.875 ms
 2  Ddmz.fhda.edu     (153.18.251.40)    2.132 ms    2.266 ms    2.094 ms
 3  Cmic.fhda.edu     (153.18.253.126)   2.110 ms    2.145 ms    1.753 ms
 4  cemic.net         (137.164.32.140)   3.069 ms    2.875 ms    2.930 ms
 5  cemic.net         (137.164.22.31)    4.205 ms    4.870 ms    4.197 ms
    ...                ...                ...         ...         ...
14  attc21.phi.net    (151.164.191.49)   7.656 ms    7.129 ms    6.866 ms
15  stcglobal.net     (151.164.243.58)   7.844 ms    7.545 ms    7.353 ms
16  pacbell.net       (209.232.138.114)  9.857 ms    9.535 ms    9.603 ms
17  209.233.48.223    (209.233.48.223)   10.634 ms   10.771 ms   10.592 ms
18  alpha.Xerox.COM   (13.1.64.93)       11.172 ms   11.048 ms   10.922 ms
```

此处在源端到目的端有17跳。注意：有些往返时间看起来非常异常，这可能是路由器太忙不能立即处理分组。

21.3 IGMP

IP协议可用到两种类型的通信：单播和多播。单播是一个发送方和一个接收方之间的通信。它是一对一的通信。但是，有些过程有时需要将同一个报文同时发送给许多的接收方，这称为**多播**（multicasting），即一到多的通信。多播有许多应用。例如，股票价格的变动要同时通知许多的证券经纪人，或者旅行的取消也会同时通知许多旅行代理，其他一些应用包括远程学习和视频点播等。

因特网组管理协议（Internet Group Management Protocol，IGMP）是其中一个必要的，但不是充分的协议（正如我们将会看到），多播也包含其他的协议。在IP协议中，IGMP是一

个辅助协议。

21.3.1 组管理

对于因特网上的多播，我们需要具有路由多播分组能力的路由器。这些路由器的路由表必须由某些多播路由协议更新，这些多播路由协议将在第22章中讨论。

IGMP不是一个多播路由协议，而是一个管理组成员（group membership）的协议。在任何网络中，都存在有一个或多个多播路由器把多播分组分发给主机或其他的路由器，IGMP协议为多播路由器（multicast router）提供关于连接到网络上的主机（路由器）成员状态的信息。

一个多播路由器每天可以接收到数以千计的属于不同组的多播分组。如果一个路由器不了解主机的成员状态，它就必须广播所有的分组。这造成通信量大量的增加并浪费带宽。一个较好的解决办法是在网络中保存一份组列表，该列表中至少有一个忠实的成员，IGMP帮助多播路由器创建和更新这个表。

> IGMP是一个组管理协议，它帮助多播路由器创建和更新与每个路由器接口有关的忠实成员的列表。

21.3.2 IGMP报文

IGMP有两种版本，我们讨论现在的版本是IGMPv2。IGMPv2有三种类型的报文（message）：查询（query）、成员报告（membership report）和离开报告（leave report），其中查询报文（query message）有两种类型：普通的（general）和特殊的（special）（见图21.16）。



图21.16 IGMP报文类型

21.3.3 报文格式

图21.17显示了IGMP（版本2）报文的格式。



图21.17 IGMP报文格式

- 类型。这个8位字段定义报文类型，如表21.1所示。类型的值用十六进制或二进制标记法表示。
- 最大响应时间。这个8位字段定义回答一个查询所需的时间，该值以十分之一秒为单位。例如，如果其值为100，即是10秒。在查询报文中，该值是非0的；在其余两个类型的报文其值为0。稍后我们会看到它的应用。
- 校验和。这个16位字段是校验和，校验和以8字节的报文计算。

表21.1 IGMP类型字段

类 型	值
普通的或特殊的查询	0x11或00010001
成员报告	0x16或00010110
离开报告	0x17或00010111

- 组地址。对于普通的查询报文，这个字段的值为0。在特殊的查询、组成员报告和离开报告报文中，这个值定义为组标识符（groupid）（组的多播地址）。

21.3.4　IGMP操作

IGMP是本地运行的。连接到网络的多播路由器有一个组多播地址表，该表内至少有一个忠实的成员（见图21.18）。

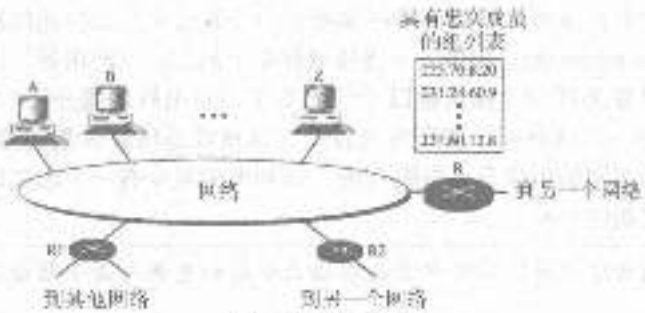


图21.18　IGMP操作

对每一组，都有一个路由器负责将多播分组分发给指定的那个组。这就是说如果与网络相连的多播路由器有三个，它们的组标识符（groupid）都是相互排斥的。例如，在图21.18中只有路由器R分发其多播地址为225.70.8.20的分组。

主机或多播路由器在一个组中可能存在成员关系。当一个主机有成员关系时，就意味着它的一个进程（一个应用程序）接收到来自某一组的多播分组。当一个路由器具有成员关系时，就意味着一个连接到它的其他接口的网络接收到这些多播分组。我们说该主机或路由器对该组有加入请求。在这两种情况下，主机和路由器保存一个组标识符表，并将它们的关系中继给分发路由器。

例如，在图21.18中，路由器R是分发路由器。其余两个多播路由器（R1和R2）依赖于路由器R维护的组列表，它们可能是这个网络中路由器R的接收者。对其他网络的这些组来说，R1和R2可能是分发路由器，而不在本网络上。

加入一组

一个主机或路由器可加入一个组。每一个主机维护一个组内成员进程表。当一个进程要加入到一个新组时，它就向主机发送请求。该主机就在它的表中增加该进程的名字和所请求的组的名字。如果这是在该特殊组中的第一个成员关系的请求，则该主机就向多播路由器发送一个成员关系报告报文。如果这不是第一个成员关系，则不需要发送成员关系报告，因为主机已是组的成员，它已经接收了这个组的多播分组。

协议要求发送两次成员关系报告，在几分钟内一个接着一个地发送。这样，如果第一个丢失或损坏了，第二个可代替它。

> 在IGMP中，成员关系报告一个接着一个地发送两次。

离开一个组

当主机发现没有进程对一个特殊的组有加入请求时，它就发送一个离开报告。同样的，当路由器发现没有一个与网络连接的接口对一个特殊的组有加入请求，它就发送一个关于该组的离开报告。

但是，当一个多播路由器接收到一个离开报告时，因为报告只是来自一个主机或路由器，

可能还有其他主机或路由器依然存在对该组有加入请求，所以它不可能立即从它的多播地址表中清除该组。为了确保这样做，路由器发送一个特殊查询报文，并插入有关该组的组标识符或多播地址（multicast address），对任意主机或路由器，路由器允许有一个规定的时间对主机或路由器做出响应。如果在这规定时间内没有接收到加入请求（成员关系报告），那么路由器就假定网络中没有忠实的成员，并从它的列表中清除该组。

监视成员关系

主机或路由器可通过发送成员关系报告报文加入到一个组中，也可通过发送一个离开报告离开一个组。但是，发送这两种类型的报告是不够的。考虑这样的情形，只有一个主机对一个组中有加入请求，但是该主机关闭或从系统中删除，多播路由器将永远接收不到离开报告，这个如何处理？多播路由器负责监视局域网上的所有主机或路由器，以发现它们是否愿意继续它们在一个组中的成员关系。

这个路由器周期性地（默认值为125秒）发送一个普通的查询报文。在这个报文中，组地址字段被设置为0.0.0.0。这就是说查询成员关系的延续需要考虑一个主机所加入的所有的组，而不是仅一个组。

> 普通查询报文没有定义一个特殊的组。

路由器期待从多播地址表中的每个组得到一个回答，即使一个新组也不例外。查询报文的最大响应时间为10秒（字段中的值为100，而这个是以十分之一秒为单位的）。当主机或路由器接收到普通查询报文时，如果它对一个组有加入请求，它就用一个成员关系报告来响应。但是如果这是一个公用的加入请求（例如，两个主机对同一组有加入请求）时，只有一个响应发送到那个组，以防止不必要的通信量，这称为一个延迟响应。注意：必须仅有一个路由器发送查询报文（通常称为查询路由器）。这也是为了防止不必要的通信量。我们在后面讨论这个问题。

延迟响应

为了防止不必要的通信，IGMP使延迟响应策略（delayed response strategy）。当一个主机或路由器接收到查询报文后，它并不立即响应而是经过一定的时间间隔后才发出响应，每一个主机或路由器使用一个随机数建立一个计时器，计时器在1秒到10秒之间，截止时间可以以1秒或更小的时间为步长。对多播组地址表中的每一个组设置一个计时器。例如，计时器对第一个组可用2秒截止，但对第三个组可用5秒截止。每个主机或路由器在它的计时器截止之前一直等待，直到它的计时器截止时间到了才发送一个成员关系报告报文。在等待过程中，如果另一个主机或路由器的计时器对同一个组截止得较早，该主机或路由器就发送一个成员关系报告。因为我们不久就会见到，报告是广播的，正等待的主机或路由器接收到报告而且知道不需要为同一个组发送一个重复报告，这样等待站点就取消它的相应的计时器。

例21.6

设想网络有三个主机，如图21.19所示。

在时刻0，接收到查询报文，每一个组的随机延迟时间（以十分之一秒为单位）在该组地址后面表出，如图所示。试说明其报告序列。

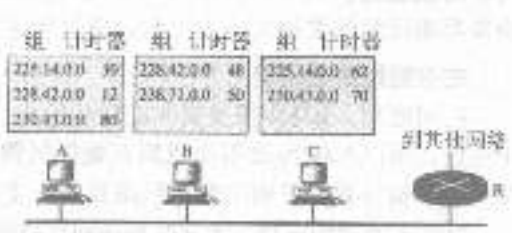


图21.19 例21.6

解：

事件发生的序列如下：

a. 时刻12：此时主机A中的组地址228.42.0.0的计时器截止，发送一个成员关系报告。路由器和每个主机（包括主机B）接收到这个报告，这样主机B删除组地址228.42.0.0的计时器；

b. 时刻30：此时主机A中的组地址225.14.0.0的计时器截止，发送一个成员关系报告。路由器和每个主机（包括主机C）接收到这个报告，这样主机C删除组地址225.14.0.0的计时器；

c. 时刻50：此时主机B中的组地址238.71.0.0的计时器截止，发送一个成员关系报告。路由器和每个主机接收到这个报告。

d. 时刻70：此时主机C中组地址230.43.0.0的计时器截止，发送一个成员关系报告。路由器和每个主机（包括主机A）接收到这个报告，主机A删除组地址230.43.0.0的计时器。

注意：如果每个主机为多播地址表中的每个组地址都发送一个报告，则将是7个报告，而用这种策略只发送4个报告。

查询路由器

询查报文可以生成许多响应。为了防止不必要的通信，IGMP为每一个网络设计了一个路由器作为查询路由器（query router）。只有这个路由器发送查询报文，其他路由器都是被动的（它们接收响应并更新它们的列表）。

21.3.5　封装

IGMP报文封装在IP数据报中，而IP数据报本身又封装在帧中，见图21.20所示。



图21.20　IGMP分组的封装

在网络层封装

对于IGMP协议，IP分组的协议字段的值是2。在协议字段中携带这个值的每一个IP分组都有需要给IGMP协议传送数据。当这个报文被封装在IP数据报中时，TTL的值必须为1。这是必需的，因为IGMP的域是局域网，IGMP报文不能传送到局域网之外。这个TTL的值为1保证了报文不能离开这个局域网，因为第一个路由器就将此值减小为0，因而就将其丢弃。表21.2显示了每一种类型报文的目的IP地址。

携带IGMP分组的IP分组的TTL字段的值为1。

表21.2　目的IP地址

类　型	IP目的地址
查询	224.0.0.1这个子网上的所有系统
成员关系报告	组的多播地址
离开报告	244.0.0.2这个子网上的所有路由器

查询报文是用多播地址224.0.0.1进行广播，所有主机和路由器都将接收到该报文。成员关系使用与被报告的多播地址（组标识符）相同的目的地址进行广播。接收到该分组的每个站点（主机或路由器）可立即确定（从头部）哪一个组报告已经被发送。由于前面已讨论过，相应的未发送报告的计时器可以被删除。站点不需要打开该分组去寻找组标识符，这个地址被复制在一个分组中，它是报文本身的一部分，也是IP头部一个字段。复制可以防止差错。离开报告报文用多播地址224.0.0.2广播（这个子网上所有的路由器），因此路由器都接收到这类报文，主机也都接收到这类报文，但不理它。

在数据链路层封装

在网络层，IGMP报文封装在IP分组中，作为IP分组处理。但是，因为IP分组有一个多播IP地址，所以ARP协议不能找到在数据链路层转发该分组的对应的MAC（物理）地址。接着发生的事情依赖于下面的数据链路层是否支持物理多播地址。

物理多播地址支持　大多数局域网支持物理多播寻址，以太网就是其中的一种，以太网的物理地址（MAC地址）是6字节（48位）。如果以太网地址的前25位是00000001000000000010111100，

它定义TCP/IP协议的物理多播地址。余下的23位可用来定一个组。要将IP多播地址转换为以太网地址，多播路由器就要提取D类IP地址中的最低的23位，并将其插入到以太网物理地址中（如图21.21所示）。



图21.21 将D类地址映射到以太网物理地址

但是，D类IP地址的组标识符是28位长，这就是说有5位未使用。这就表示了32（2^5）个IP级的多播地址映射到一个的多播地址。换言之，映射是多对一，而不是一对一。如果D类地址中的组标识符的最左5位不全为0，主机就可能接收到并不是真正属于它所涉及到的组的分组。为此，主机必须检查IP地址，并丢弃任何不属于它的分组。

其他局域网也支持同样的概念，但映射的方法可以不同的。

以太网的多播物理地址是在01：00：5E：00：00：00到01：00：5E：7F：FF：FF范围中。

例21.7

将多播IP地址230.43.14.7转换成以太网多播物理地址。

解：

这可用2个步骤来完成：

a. 我们用十六进制写出IP地址的最右23位。这可将最右边3个字节变换成十六进制。然后如果最左边的数字大于或等于8，则将该数减去8。在本例中，其结果是2B：0E：07。

b. 将a步所得的结果加到开始的以太网多播地址01：00：5E：00：00：00，其结果是01：00：5E：2B：0E：07。

例21.8

将多播IP地址238.212.24.9转换成以太网多播地址。

解：

a. 用十六进制表示的最右边3个字节是D4：18：09。我们需将最左边的数字减去8，其结果是54：18：09。

b. 将a步所得的结果加到以太网多播的开始地址，其结果是

01：00：5E：54：18：09。



图21.22 隧道技术

无物理多播地址支持 大多数广域网不支持物理多播地址。要通过这样的网络发送多播分组，就要使用称为隧道技术的过程。在隧道技术（tunneling）中，多播分组封装成单播分组并通过网络发送，然后在另一端，这个分组又转换成多播分组（如图21.22所示）。

21.3.6 Netstat应用程序

*netstat*应用程序可以用来得到接口所支持的多播地址。

例21.9

我们使用带有三个选项的*netstat*命令。选项-n是以数字形式显示的IP地址。选项-r显示

路由表。选项-a显示所有的地址（单播和多播地址）。注意：这里仅给出与我们讨论有关的项目。“Destination”定义目的地址，“Gateway”定义路由器，“Iface”定义接口，“Mask”定义掩码，“Flag”定义标记。标记U表该路由可使用，标志G表示该路由是一个网关（路由器）。

```
$ netstat -nra
Kernel IP routing table
Destination     Gateway         Mask            Flags   Iface
153.18.16.0     0.0.0.0         255.255.240.0   U       eth0
169.254.0.0     0.0.0.0         255.255.0.0     U       eth0
127.0.0.0       0.0.0.0         255.0.0.0       U       lo
224.0.0.0       0.0.0.0         224.0.0.0       U       eth0
0.0.0.0         153.18.31.254   0.0.0.0         UG      eth0
```

注意：多播地址用灰色表示。具有从240.0.0.0到239.255.255.255多播地址的任何分组都被屏蔽，并传递给以太网接口。

21.4 ICMPv6

我们在第20章中讨论过IPv6。在TCP/IP协议族的版本6中被修改的另一个协议是ICMP（IMCPv6）。这个新版本与版本4的策略和目的一样，修改ICMPv4使得它更加适合IPv6。此外，在版本4中的一些独立的协议现在成为**网际控制报文协议**（Internet working Control Message Protocol，ICMPv6）的一部分。图21.23比较了版本4和版本6的网络层。



图21.23　版本4和版本6的网络层比较

版本4中的ARP和ICMP协议合并到ICMPv6。RARP协议从这个协议族中取消了，因为它很少使用。而且BOOTP具有与RARP相同的功能。

正如ICMPv4那样，我们将ICMP报文分成两大类。但是，每一类都比前面版本有更多的报文类型。

21.4.1 差错报告

正如我们在讨论版本4时所看到的那样，ICMP的一个主要职责就是对差错进行报告。有5种差错类型：目的端不可达、分组太大、时间超时、参数问题和重定向。ICMPv6组成差错分组，再封装成IP数据报，然后传递给产生差错的数据报的源端。表21.3对ICMPv4和ICMPv6的**差错报告报文**（error-reporting message）进行了比较。在版本6中源端抑制报文取消了，因为优先级和流标号字段允许路由器控制拥塞，并将最不重要报文丢弃。在这个版本中，不需要通知发送方将速率放慢。增加了分组太大报文，因为在IPv6中分段是发送方的责任。如果发送方没有对分组长度做出正确的判断，路由器就没有任何选择，而只能丢弃分组，并发送差错报告报文给发送方。

表21.3　ICMPv4和ICMPv6的差错报告的比较

报文类型	版本4	版本6
目的端不可达	是	是
源端抑制	是	否
分组太大	否	是
时间超时	是	是
参数问题	是	是
重定向	是	是

目的端不可达

目的端不可达报文（destination-unreachable message）的概念和ICMPv4完全相同。

分组太大

这是增加到版本6的一个新的报文类型。如果路由器接收到一个数据报，它大于数据报能够通过网络的最大传输单元（MTU），那么就要发生两件事。第一，路由器要丢弃该数据报；第二，要发送差错分组，即**分组太大报文**（packet-too-big message）给源端。

时间超时

这个报文与版本4相似。

参数问题

这个报文与版本4相应的报文相似。

重定向

重定向报文（redirection message）的目的与版本4的相同。

21.4.2 查询

除差错报告外，ICMP还能诊断某些网络问题。这是通过**查询报文**（query message）来完成的。已定义了4组报文：回送请求和回答、路由器询问和通知、邻站询问和通知以及组成员关系。表21.4显示了版本4与版本6的查询报文的比较。ICMPv6取消了两组查询报文：时间戳请求和回答以及地址掩码请求和回答。取消时间戳请求和回答是因为它们在其他协议中，如TCP已实现了。同时也因为在过去没有使用过它们；在IPv6中取消地址掩码请求和回答报文是因为地址的子网部分允许用户使用多达$2^{32}-1$个子网。因此，定义在IPv4中的子网掩码在这里是不需要的。

表21.4 ICMPv4和ICMPv6中查询报文的比较

报文类型	版本4	版本6
回送请求和回答	是	是
时间戳请求和回答	是	否
地址掩码请求和回答	是	否
路由器询问和通知	是	是
邻站询问和回答	ARP	是
组成员关系	IGMP	是

回送请求和回答

回送请求和回答报文的格式和思路与版本4一样。

路由器询问和通告

路由器询问和通告报文的格式和思路与版本4一样。

邻站询问和通告

如前所述，版本4的网络层包含一个独立的协议称为ARP。在版本6中，这个协议取消了，而它的任务并入到ICMPv6中。思路完全相同，但报文格式改变了。

组成员关系

如前所述，版本4的网络层包含一个独立的协议称为IGMP。在版本6中，这个协议取消了，而它的任务并入到ICMPv6中。其目的完全相同。

推荐读物

作为深入讨论与本章有关的主题，推荐下列读物和网站，在本书末列出方括号[…]中的参考资料。

读物

ARP和RARP在[For06]的第7章和[Ste94]的第4、5章中有讨论。ICMP在[For06]的第9章和[Ste04]的第6章中有讨论。IGMP在[For06]的第10章和[Ste94]的第13章中有讨论。BOOTP和DHCP可参见[For06]的第16章和[Ste94]的第16章。ICMPv6的讨论见[For06]的第27章。

网站

- www.ietf.org/rfc.html 有关请求评论（RCF）信息。

请求评论

ARP和RARP的讨论可在下列请求评论中找到：
826、903、925、1027、1293、1329、1433、1868、1931、2390
ICMP的讨论可在下列请求评论中找到：
777、792、1016、1018、1256、1788、2521
IGMP的讨论可在下列请求评论中找到：
966、988、1054、1112、1301、1458、1469、1768、2236、2357、2365、2502、2588
BOOTP和DHCP的讨论可在下列请求评论中找到：
951、1048、1084、1395、1497、1531、1532、1533、1534、1541、1542、2131、2132

本章小结

- 将分组传递给主机或路由器需要两级地址：逻辑地址和物理地址。物理地址在物理级标识一个主机或路由器。
- 逻辑地址到物理地址的映射可以是静态的或动态的。
- 静态映射要使用逻辑地址与物理地址的对照表，维护该表需要很大开销。
- 地址解析协议（ARP）是一种动态映射方法，在给出逻辑地址时可找出其物理地址。
- 在代理ARP中，一个路由器代表一组主机。当ARP请求寻找该组中任何一个主机时，该路由器就发送它自己的物理地址。这就产生了一个子网化的效应。
- 反向地址解析协议（RARP）是一种动态映射，它将一个给定物理地址和一个逻辑地址联系起来。
- ICMP发送4对查询报文：回送请求和回送回答、时间戳请求和回答、地址掩码请求和回答以及路由器询问和回答。
- 使用ICMP报文的头部和数据字段计算ICMP的校验和。
- 分组因特网搜寻器（*ping*）是一个应用程序，它使用ICMP的服务来测试主机的可达性。
- 多播就是将相同的报文同时发送给一个以上的接收方。
- 因特网组管理协议（IGMP）帮助多播路由器创建和更新多播地址表。
- 3种类型的IGMP的报文是查询报文、成员关系报告和离开报告。
- 延迟响应策略是防止局域网上不必要的通信。
- IGMP报文封装在IP数据报中。
- 包括以太网在内的大多数局域网支持物理多播寻址。
- 不支持物理多播寻址的广域网可使用称为隧道技术的过程来发送多播分组。
- BOOTP和动态主机配置协议（DHCP）都是客户机/服务器应用程序，用于将重要的网络信息传递给无盘计算机或第一次引导计算机。
- BOOTP请求被封装成UDP用户数据报。

- BOOTP是静态配置协议，它使用一个表将IP地址映射到物理地址。
- 中继代理是一个路由器，它帮助将本地的BOOTP请求发送到远程服务器。
- DHCP是动态配置协议，它有两个数据库：一个与BOOTP相似，另一个是IP地址池，用于临时的分配。
- DHCP服务器将IP地址的租用发送给客户使用一个特定的时间。
- DHCPv6和版本4一样，报告差错、处理组成员关系、更新特定的路由器和主机的表以及检查主机的生存能力。

习题

复习题

1. ARP分组的长度是固定的吗？试加以解释。
2. 当协议是IPv4而硬件是以太网时，一个ARP分组的长度是多少？
3. 在问题2中，携带ARP分组的以太网的帧长度是多少？
4. 以太网的广播地址是什么？
5. 为什么对一个失效的ICMPv4差错报文要限制再产生一个ICMPv4报文？
6. 在差错报告IMCPv4报文中包括IPv4的头部和数据报数据前8个字节的目的是什么？
7. 试给出一个例子，说明一个主机可能会永远收不到一个重定向报文。
8. ICMPv4分组的最小长度是多少？ICMPv4分组的最大长度是多少？
9. 携带ICMPv4分组的IPv4分组的最小长度是多少？最大长度是多少？
10. 我们如何能够确定是否IPv4分组携带了IPCMv4分组？
11. 携带IPv4分组的以太网帧的最小长度是多少？最大长度是多少？
12. 为什么没有必要将ICMPv4报文传送到它自己所在的网络之外？

练习题

13. 一个路由器的IPv4地址是125.45.23.12，而所在的以太网物理地址是：23：45：AB：4F：67：CD，接收到一个分组，其目的IPv4地址是125.11.78.10。试给出此路由器发出的ARP请求分组中的各项，假定子网化。
14. 试给出响应练习13而发送ARP分组中的各项目。
15. 将练习13的结果封装成数据链路层的帧，试填充所有字段。
16. 将练习14的结果封装成数据链路层的帧，试填充所有字段。
17. 主机A向主机B发送数据报。主机B从未接收到该数据报，而主机A也未收到出问题的通知。试给出可能发生的情况的两种不同的解释。
18. 试计算以下ICMP分组的校验和。

    类型：回送请求　标识符：123　序号：25　报文：Hello
19. 一个路由器接收到一个IPv4分组，其源IP地址是130.45.3.3，而目的IP地址是201.23.4.6，该路由器不能在其路由表中找到其目的地址，试问发送出去的ICMP报文各字段应该是什么？
20. TCP接收到一个目的端口为234的报文分段，但TCP经过检查无法打开这个目的端口，试问发出去的ICMP报文各字段应该是什么？
21. 一个组多播地址是231.24.60.9。对于局域网在使用TCP/IP时，其48位的以太网地址是什么？
22. 如果一个路由器的组表中有20个项目，它应当周期性地发送20个不同的查询还是只发送一个？试解释之。

23. 如果一个主机愿意在5个组中继续其成员关系，它应该发送5个不同的成员关系报告报文还是只发送一个？
24. 一个以太网上的路由器接收到多播IP分组，其组标识符为226.17.18.4。当主机检查其多播组表时，它找到这个地址。试说明路由器如何将这个IP分组封装成以太网帧，并将它发送给各接收端。进一步给出以太网帧的所有项目。此路由器出去的IP地址是185.23.5.6，而物理地址是4A：22：45：12：B1：E2，这个路由器需要ARP服务吗？
25. 一个IPv4地址为114.45.7.9的主机接收到一个IGMP查询。当它检查表组时发现没有项目，该主机应采取什么动作？它应该发送什么报文？
26. 一个IPv4地址为222.5.7.19的主机接收到一个IGMP查询，当它检查表组时发现有两个项目：227.4.3.7和229.45.6.23。该主机应该采取什么动作？它应该发送报文吗？如果是的话，什么类型报文和发送几个？
27. 在以太网中的IPv4协议能够支持多少个多播地址？IPv4协议能够支持多少个多播地址？当将多播IPv4地址转换为以太网多播地址时，地址空间损失多少？
28. 试将下列的IPv4多播地址转换成以太网多播地址，它们中有哪些指明了相同的以太网地址？

a. 224.18.72.8　　b. 235.18.72.8
c. 237.18.6.88　　d. 224.88.12.8

探究题

29. 用*ping*程序测试你自己的计算机（环回）。
30. 用*ping*程序测试美国国内的一个主机。
31. 用*ping*程序测试美国国外的一个主机。
32. 用*traceroute*（或*tracert*）程序测试从你的计算机到学院或大学中的一个主机的路由。
33. 用*traceroute*程序寻找你的服务器是否支持多播寻址。
34. DHCP使用诸如DHCPREQUEST、DHCPDECLINE、DHCPACK、DHCPNAK以及DHCPELEASE的报文，试说明这些报文的目的。

第22章 传递、转发和路由选择

这一章讨论IP分组到它们最终目的端的传递、转发和路由选择。所谓传递（Delivery）是指在网络层控制下用底层的网络对一个分组进行处理的方法。所谓转发（Forwarding）是指将一个分组传递到下一个站点的方法。所谓路由选择（Routing）是指在转发过程中创建路由表的方法。

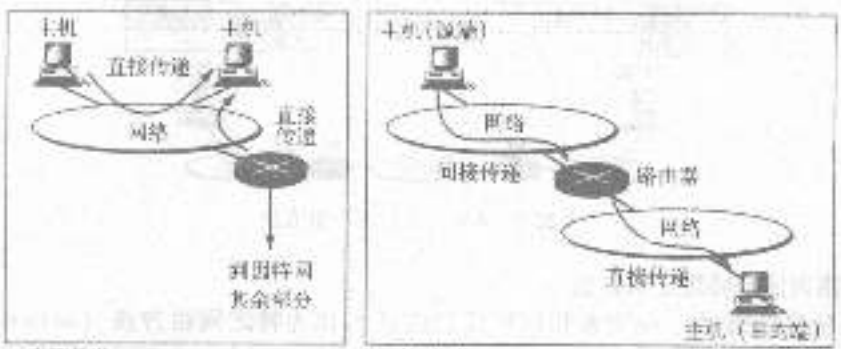
路由选择协议（routing protocol）用于不断地更新在转发和路由选择中要查找的路由表。在本章中，我们也讨论常用的单播和多播路由（multicast routing）协议。

22.1 传递

网络层负责用底层物理网络处理分组，我们定义这种处理为分组的传递。

直接传递和间接传递

分组传递到最终目的端可使用两种不同的方法：直接传递和间接传递，如图22.1所示。



a) 直接传递　　b) 间接传递

图22.1 直接传递和间接传递

直接传递

直接传递（direct delivery）就是分组的最终目的端的主机是与发送方都连接在同一个物理网络上。当分组的源端和目的端都在同一个物理网络上，或传递是在最后一个路由器与目的主机之间进行时，就出现直接传递。

发送方很容易确定传递是否是直接传递，它只要提取出分组目的端的网络地址（用掩码），并与它所连接的网络地址进行比较，如果相同，则传递就是直接的。

间接传递

如果目的主机与发送方不在同一个网络上，分组就是间接传递。在间接传递（indirect delivery）时，分组从一个路由器传送到另一个路由器，直到它到达与最终目的端连接在同一个物理网络上的路由器为止。注意：一个传递永远包含一个直接传递和0个或多个间接传递。还要注意：最后的传递总是直接传递。

22.2 转发

转发是指将分组路由到它的目的端。转发要求主机或路由器有一个路由表。当主机有分组要发送时，或路由器已收到一个分组要进行转发时，就要查找路由表以便求得到达最终目的端的路由。但是，这种简单方法对于今天的互联网（如因特网）已是不可能的。因为路由表中的项目数已使得路由表的查找效率很低。

22.2.1 转发技术

有些转发技术可以使路由表的大小成为可管理的，同时也可以处理一些问题（如安全性问题）。此处，我们简要讨论这些方法。

下一跳方法与路由方法

第一种减少路由表中内容的技术，称为下一跳方法（next-hop method）。在这种方法中，在路由表中只保留下一跳地址，而不保留完整的路由信息。在路由表中保留完整路由信息的技术，称为路由方法（route method）。路由表各项目必须彼此协调的。图22.2表示使用这种技术可以简化路由表。

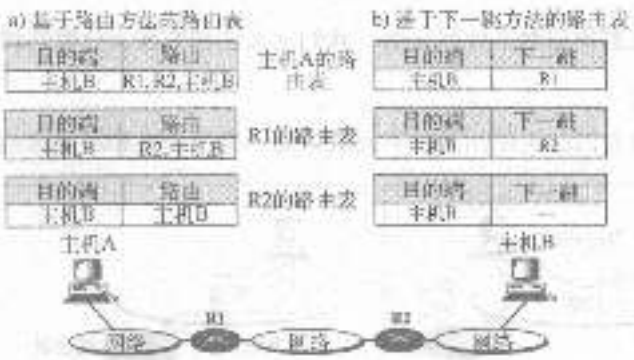


图22.2　路由方法与下一跳方法

特定网络方法与特定主机方法

第二种使路由表减小和使查找过程简化的技术称为特定网络方法（network-specific method）。此处不是对连接在同一个物理网络上的所有主机都有一个项目（这种技术称为特定主机方法（host-specific method）），而是仅用一个项目来定义这个目的网络本身的地址。换言之，我们将连接在同一网络上的所有主机看作是一个项目。例如，有1 000个主机连接在同一个网络上，则在路由表中仅用一个项目而不是1 000个项目，图22.3表示了这个概念。

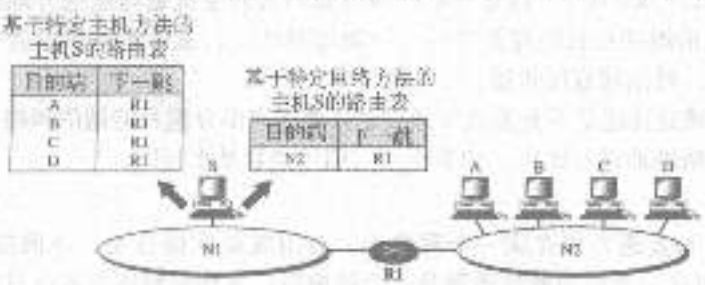


图22.3　特定主机方法与特定网络方法

特定主机路由选择是用于如检查路由或提供安全措施等目的。

默认方法

另一种简化路由表的技术是默认方法（default method）。在图22.4中，主机A连接到只有两个路由器的网络上。路由器R1用来将分组转发到连接网络N2的主机，但是，对因特网的其余部分，则使用路由器R2。因此，可不必将整个互特网中的所有网络都列出，主机A可以仅使用一个称为默认的项目（通常定义网络地址为0.0.0.0）。

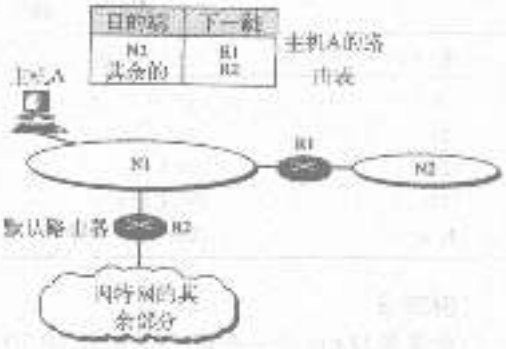
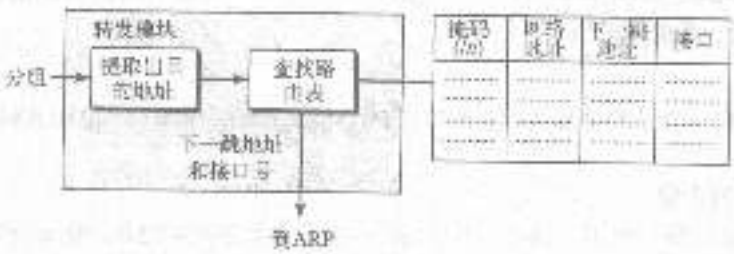


图22.4 默认方法

22.2.2 转发过程

让我们讨论转发过程。假定主机和路由器使用无类寻址，这是因为分类寻址可以被作为无类地址的一种特殊情况处理。在无类寻址中，路由表对涉及到的每一个地址块都需要有一行信息。路由表需要根据网络地址（地址块的第一个地址）进行查询，但很遗憾，在分组中只有目的地址而没有网络地址。为了解决这个问题，在路由表中需要包含掩码（/n），因此，对相应的地址块，我们需要有包含该掩码的附加列。图22.5表示了无类寻址的一个简单转发模块。


转发模块
分组
提取目的地址
查找路由表
下一跳地址和接口号
掩码(/n)
目的地址
下一跳地址
接口
……
……
……

解：

表22.1是相应的路由表。

表22.1　图22.6中路由器R1的路由表

掩码	网络地址	下一跳	接口
/26	180.70.65.192	——	m2
/25	180.70.65.128	——	m0
/24	201.4.22.0	——	m3
/22	201.4.16.0	……	m1
其余	其余	180.70.65.200	m2

例22.2

如果图22.6中的一个目的地址为180.70.65.140的分组到达路由器R1，说明其转发过程。

解：

路由器执行下列步骤：

1. 第一个掩码（/26）作用于这个目的地址，其结果是180.70.65.128，它与对应的网络地址不匹配；

2. 第二个掩码（/25）作用于这个目的地址，其结果是180.70.65.128，它与对应的网络地址匹配。将下一跳地址（nex-hop address）（在这种情况下，它是分组的目的地址）和接口号m0传送到ARP做进一步处理。

例22.3

如果图22.6中的一个目的地址为210.4.22.35的分组到达路由器R1，说明其转发过程。

解：

路由器执行下列步骤：

1. 第一个掩码（/26）作用于这个目的地址，其结果是210.4.22.0，它与对应的网络地址不匹配（第一行）；

2. 第二个掩码（/25）作用于这个目的地址，其结果是210.4.22.0，它与对应的网络地址不匹配（第二行）；

3. 第三个掩码（/24）作用于这个目的地址，其结果是201.4.22.0，它与对应的网络地址匹配。分组的目的地址和接口号m3传送到ARP。

例22.4

如果图22.6中的一个目的地址为18.24.32.78的分组到达路由器R1，说明其转发过程。

解：

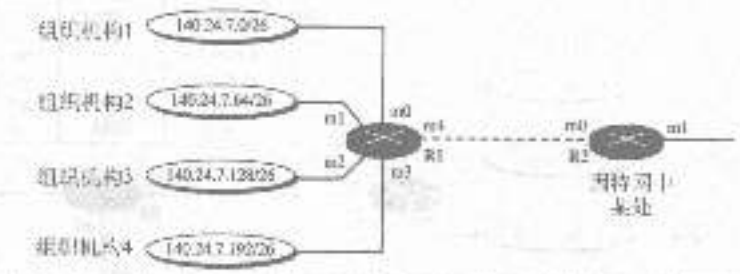
这时所有掩码作用于这个目的地址都与网络地址不匹配。当它到达表的尾部时，模块将下一跳地址180.70.65.200和接口号m2传送到ARP。这就是需要通过路由器转发到因特网某处的输出包。

地址聚合

当我们使用无类寻址时，很可能路由表的项目数量将会增多。这是因为无类寻址的意图是将整个地址空间划分成可管理的地址块。而路由表增大将增加查找时间。为了缓解这个问题，我们设计了地址聚合（address aggregation）思想。在图22.7中有两个路由器，路由器R1连接到4个组织机构的网络，每个组织机构的网络都使用64个地址。而路由器R2在远离R1的某个地方。路由器R1的路由表较长，这是因为分组必须要正确地路由到适当的组织机构网络。

而路由器R2的路由表很短。对于路由器R2来说，不管到那一个组织机构，任一目的地址为140.24.7.0到140.24.7.255的分组都是从接口m0发送。因为4个组织机构的地址块被聚合为一个较大的地址块，所以这个称为地址聚合。如果每个组织机构的地址没有聚合成一块地址，那么路由器R2将要有较长的路由表。

注意：尽管地址聚合的思想与子网化的思想类似，但这里没有一个共同的站点，每个组织机构的网络都是独立的。另外，我们还可有若干级的聚合。



掩码	网络地址	下一跳地址	接口
/26	140.24.7.0	———	m0
/26	140.24.7.64	———	m1
/26	140.24.7.128	———	m2
/26	140.24.7.192	———	m3
/0	0.0.0.0	默认	m4

R1路由表

掩码	网络地址	下一跳地址	端口
/24	140.24.7.0	———	m0
/0	0.0.0.0	默认	m1

R2路由表

图22.7 地址聚合

最长掩码匹配

如果在图22.7中，其中一个组织机构在地理位置上与其他三个不邻近，将会发生什么？例如，如果组织机构4由于某种原因没有连接到路由器R1。我们能否还能用地址聚合的思想以及还能对组织机构4指派地址块140.24.7.192/26吗？

由于无类寻址的路由选择使用另一个**最长掩码匹配**（longest mask matching）原则，回答是行的。这个原则指明在路由表中掩码存放是按最长的到最短的次序。换言之，如果有三个掩码/27、/26和/24，则/27必定是第一个项目，而/24必定是最后项目。如果用这一原则，我们看看如果组织机构4与另外三个组织机构分离，情形会是怎样，如图22.8所示。

假定一个分组到达目的地址是140.24.200的组织机构4，将路由器R2中第一个掩码作用于它，得到网络地址140.24.7.192。该分组被正确通过接口m1而到达组织机构4。但是，如果路由表不按最长的前缀存储，作用掩码/24将给出分组到路由器R1的不正确路由选择。

分层路由选择

为了解决巨大的路由表问题，我们可在路由表中建立层次结构。在第1章中，我们提到过今天的因特网具有层次结构，就是说因特网被划分为国际的和国内的ISP，国内的ISP又被划分为地区ISP，而地区ISP再进一步划分为本地ISP。如果路由表也像因特网那样具有层次结构，那么可降低路由表的长度。

让我们考虑本地ISP情况。本地ISP可被分配一个单地址块，而该块是具有某一前缀长度的大块。本地ISP将这个块划分为不同大小的一些小块。如果分配给本地ISP的地址块以a.b.c.d/*n*开始，那么这个ISP可建立以e.f.g.h/*m*开始的一些块，其中*m*大于*n*，而对每个客户，*m*是可变的。

这如何降低路由表的长度呢？因特网的其余部分不必知道这样的划分。对因特网的其余

部分来说，本地ISP的所有客户都被定义为a.b.c.d/n。在这个大块中的每一个分组的目的地址都路由到本地ISP。对所有这些客户来说，在世界上的每个路由器中都只有一项，它们都属于同一组。当然，在本地ISP内部，路由器必须识别出了块并路由到目的客户。如果某一客户是一较大的组织机构，那么它也可以通过子网化和划分它的子块为更小的子块（子块的子块）建立另一个层次。在无类路由选择中，只要我们遵循无类寻址规则，层次的级是没有限制的。

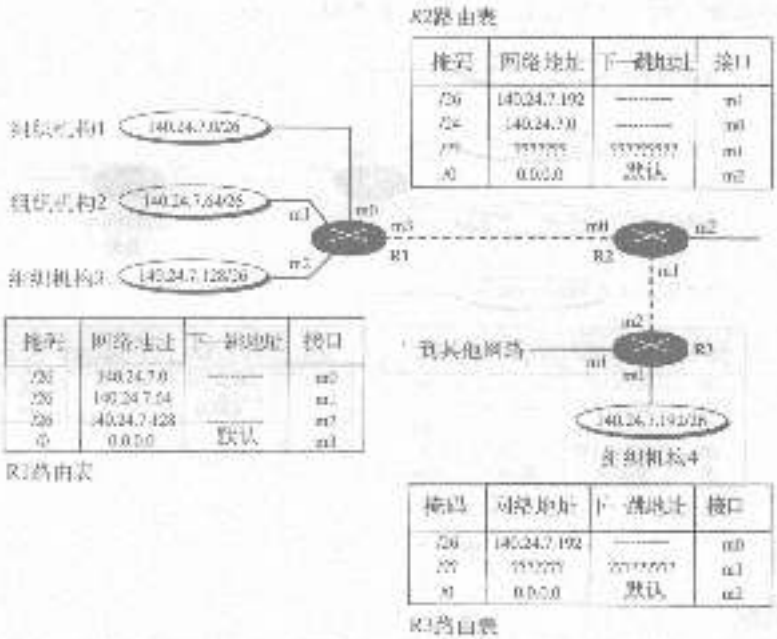


图22.8　最长掩码匹配

例22.5

作为分层路由选择（hierarchical routing），让我们考虑图22.9。区域ISP被授予以地址120.14.64.0开始的16 384个地址。区域ISP决定将这个块划分成4个子块，每块4 096个地址。其中的3个子块分别指派给3个本地ISP，而第2子块保留作将来使用。注意：由于原始块的掩码是/18，因此每块的掩码是/20。

第一个本地ISP将已指派给它的子块划分成8个较小的子块，每一个指派给一个小的ISP，每个小的ISP对128个家庭（H001到H128）提供服务，每个家庭使用4个地址。注意：现在由于进一步划分成8个子块，因此每个小ISP的掩码是/23，因为一个家庭只有4个地址（2^{32-30}等于4），所以每个家庭的掩码是/30。

第二个本地ISP将它的块划分成4个子块，并将地址指派给4个大型的组织机构（LOrg01到LOrg04）。注意：每个大型组织机构有1 024个地址。

第三个本地ISP将它的块划分成16个子块，并将每个子块指派给一个小型的组织机构（SOrg01到SOrg16），每一个小型的组织机构有256个地址，掩码是/24。

在这种配置中，存在层次结构。在因特网上的所有路由器将带有目的地址从120.13.64.0到120.14.127.255的分组发送到区域ISP。

区域ISP向本地ISP1发送目的地址为120.14.64.0到120.14.64.255的分组。本地ISP1向H001发送目的地址为120.14.64.0到120.14.64.3的分组。



图22.9 ISP的层次结构路由选择

地理路由选择

为了更进一步地减少路由表的大小，我们需要将分层路由选择扩充为扩充包括地理的路由。我们必须将整个地址空间划分为数量不多的几个大块，我们指派一个大块给北美，一个大块给欧洲，一个给亚洲，一个给非洲等。欧洲以外ISP的路由器在它们的路由表中将只有到欧洲的一个项目，在北美以外的ISP路由器在它们的路由表只有到北美的一个项目等。

22.2.3 路由表

现在让我们讨论路由表。主机或路由器要转发IP分组就要有一个路由表，并给每一个目的端设置一个项目。路由表可以静态的，也可以是动态的。

静态路由表

静态路由表（static routing table）包含有人工输入的信息。网络管理人员将每一个目的地址的路由输入到路由表中。当路由表生成后，因特网中的变化无法自动在路由表中进行自动更新。路由表必须由网络管理员手工改变。

静态路由表用在不会经常改动的小型互联网中，或用于故障查找的试验互联网中。对于像因特网这样的大型互联网，静态路由表是极差的方法。

动态路由表

动态路由表（dynamic routing table）使用一个动态路由选择协议，如RIP，OSPF或BGP，表面可以周期性地进行更新。当因特网中发生变化时，例如当某个路由器关闭或某条链路中断，动态路由选择协议就自动更新所有路由器（最后也将在所有主机中）的路由表。

为了有效地传递IP分组，一个大的互联网如因特网需要动态地更新其路由表。我们将在本章的后面详细讨论二种动态路由选择协议。

路由表格式

正如前面所述的那样，对于无类寻址的路由表的最小列数是4，但是今天的路由表甚至有更多的列数。我们应该明白列数与厂商有关，并不是所有列在不同的路由器中都有。

目前路由器的路由表中的一些常有字段，如图22.10所示。

掩码	网络地址	下一跳地址	接口	标志	引用计数	使用
.....	.....	.....	.....	.....	.....	.....

图22.10 路由表中常用的字段

- 掩码。这个字段定义应用于该表项的掩码。

- 网络地址。这个字段定义分组必须传递的最终的网络地址。在特定主机路由选择的情况下，该字段定义目的主机的地址。
- 下一跳地址。这个字段定义分组必须传递到下一跳路由器的地址。
- 接口。这个字段显示接口的名称。
- 标记。这个字段定义多达5个标记。所谓标记就是一个通/断开关，它表示或者存在或者不存在。这5个标记是：U（工作）、G（网关）、H（特定主机）、D（由于重定向而增加的）和M（由于重定向而修改的）。

a. U（工作）标记U指出路由器正在工作。如果没有这个标记就表示路由器已不工作，这时分组无法转发因而将被丢弃。

b. G（网关）标记G指出目的端是另一个网络。分组必须传递到下一个跳路由器以便传递（间接传递）。如果没有这个标记，就表示目的端在这个网络中（直接传递）。

c. H（特定主机）标记H指出在地址字段的项目是一个特定主机地址。如果没有这个标志，就表示这个地址只是目的端的网络地址。

d. D（由于重定向而增加的）标记D指出由于ICMP报文中的重定向报文而在主机路由表中增加了到这个目的端的路由选择信息。我们已在第21章讨论过重定向与ICMP协议。

e. M（由于重定向而修改的）标记M指出由于ICMP报文中的重定向报文而修改了到这个目的端的路由选择信息。我们已在第21章讨论过重定向与ICMP协议。

- 引用计数。这个字段给出在任何时候使用本路由的用户个数。例如，如果有5个人在同样的时间从本路由器连接到这个相同的主机，则这一列的值是5。
- 使用。这个字段指出经过本路由器发送到相应的目的端的分组数。

实用程序

有几个用于查找路由信息和路由表的内容的实用程序，我们讨论两个：netstat和ifconfig。

例22.6

在UNIX或者LINX操作系统下，可用于查找路由信息和路由表的内容一个命令是netstat。下面显示了一个默认服务器的内容列表。我们在命令中使用了两个选项*r*和*n*，其中选项*r*表示我们对路由表感兴趣，而选项*n*是查找地址的数字形式。注意：这是一个主机的路由表，而不是路由器的。

```
$ netstat -rn
Kernel IP routing table
Destination     Gateway         Mask            Flags   Iface
153.18.16.0     0.0.0.0         255.255.240.0   U       eth0
127.0.0.0       0.0.0.0         255.0.0.0       U       lo
0.0.0.0         153.18.31.254   0.0.0.0         UG      eth0
```

注意：表中列的次序与我们表示的是不同的。此处目的端列定义网络地址，UNIX中的网关（gateway）与路由器同义，该列定义下一跳地址。值0.0.0.0表示传递是直接的。最后一行中的G表示了通过一个路由器（默认路由器）到达目的端。*Iface*列定义接口，主机只有一个实际的接口eth0，它是指连接到以太网的接口0。第二个接口lo，实际上是虚拟回送接口，它包含主机用回送地址127.0.0.0接收分组。

通过在给定接口（eth0）上使用*ifconfig*命令可以查找到有关服务器IP地址和物理地址更多的信息。

```
$ ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:B0:D0:D6:09:5D
inet addr:153.18.17.11  Bcast:153.18.31.255  Mask:255.255.240.0
```

从上述信息中，我们可推断出服务器的配置，如图22.11所示。

注意：*ifconfig*命令给出IP地址和接口的物理（硬件）地址。



图22.11 例22.6的服务器的配置

22.3 单播路由选择协议

路由表可以是静态的也可以是动态的。静态路由表是由人工输入项目，而动态路由表在互联网中某处有变化时就自动地更新。当今，互联网需要动态路由表。只要在互联网中有些变化，路由表就应当尽快地更新。例如，当一个路由器不通了，路由表就必须更新，而当发现有一条较好的路由时，路由表也需要更新。

由于需要有动态路由表，因此产生了多种路由选择协议。路由选择协议是一些规则和过程的组合，使得在互联网中的各路由器能够彼此互相通知这些变化。路由选择协议使得旧金山的路由器能够知道在得克萨斯的网络故障。路由选择协议还包括一些过程，用来合并从其他路由器接收到的信息。

22.3.1 优化原则

路由器从网络接收到分组，并将其转发到另一个网络。一个路由器通常连接到多个网络，当它接收到分组时，它应当将分组转发到哪一个网络呢？这个决定是基于最优化原则而做出哪一个可用路径是最佳路径？优化这一名词的定义是什么？

一种方法是给通过网络指定代价。我们称这个代价为度量（metric）。然而，给每一个网络指定的度量取决于协议的类型。有些简单协议，如路由信息选择协议（RIP），同等对待每一个网络，即通过每一个网络的代价都是一样，其跳数为1。因此，如果分组经过10个网络才到达目的端，则总的代价是10跳。

其他协议，如开放最短路径优先（OSPF），允许网络管理员基于所需服务类型指定通过网络的代价，即通过一个网络的路由可以有不同的代价。例如，如果最大吞吐量是所期望的服务类型，则一条卫星链路就比一条光纤线路具有更小的度量；另一方面，如果最小延迟是所期望的服务类型，则光纤线路比卫星链路具有更小的度量。OSPF允许每一个路由器有几个基于服务类型的路由表。

还有一些协议以完全不同的方式定义度量。在边界网关协议（BGP）中，准则就是可由网络管理员设置的策略。策略定义应当选择什么路径。

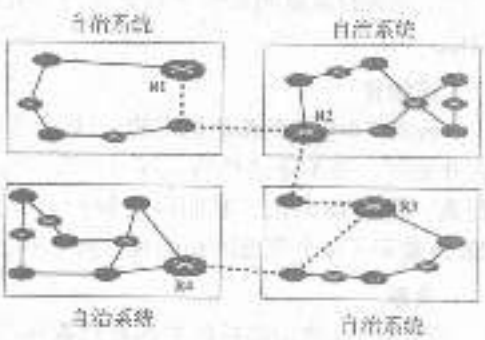


图22.12 自治系统

22.3.2 域内部和域间路由选择

今天的互联网是非常大的。以致于仅使用一个路由选择协议就无法处理更新所有路由器路由表的任务。为此，需要将互联网划分为多个自治系统。**自治系统**（autonomous system）是一个单一的管理机构管辖下的一组网络和路由器。在自治系统内部的路由选择称为**域内路由选择**（interadomain routing）。在自治系统之间的路由选择称为**域间路由选择**（interdomain routing）。每一个自治系统可以选择一种或多种域内路由选择协议处理自治系统内部的路由选择。但是，处理自治系统之间的路由选择，通常只能使用一种域间路由选择协议（见图22.12）。

现有几种域内和域间路由协议在使用。本章只讨论几个最流行的协议。两个域内路由协议：RIP和OSPF，以及一个域间路由协议BGP（见图22.13）。



图22.13　流行的路由选择协议

路由选择信息协议（Routing Information Protocol，RIP）是一种距离向量协议。**开放最短路径优先**（Open Shortest Path First，OSPF）是一种链路状态协议。**边界网关协议**（Border Gate Protocol，BCP）是一种路径向量协议。

22.3.3　距离向量路由选择

在距离向量路由选择中，任何两个节点之间路由最低代价是最小距离的路径。正如其名字的含义，每个节点都保留一张到其他每个节点的最小向量距离（表）。每个节点也用这张表中所表示的路由中的下一个节点（下一跳路由选择）指导分组流向目的节点。

我们可将节点看做区域中的城市，而将线看做连接城市的道路。一张表可表示城市之间路程最小距离的表。

图22.14表示具有相应表的5个节点的系统。

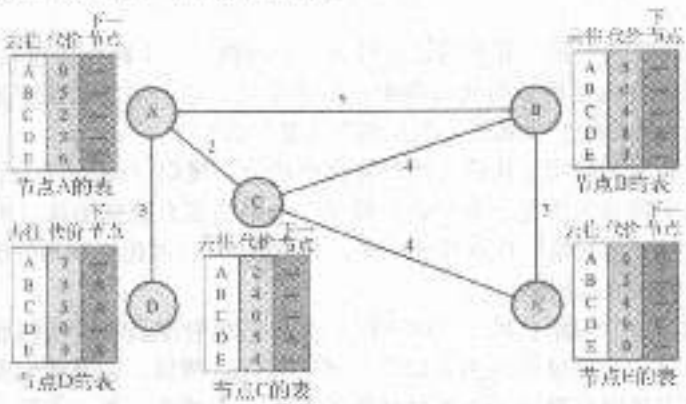


图22.14　距离向量路由选择表

节点A的表说明从该节点如何到达其他节点。例如，到达节点E的代价是6，其路由是经过C。

初始化

图22.14中的表都是稳定的，每个节点都知道如何到达其余任一个节点及其代价。但是，在开始时，并不是这样的。每一个节点仅知道与它直接连接的邻站（immediate neighbor）的距离。到目前为止，我们假定每一个节点能向邻站发送报文并得到与这些邻站的距离。图22.15表示了每个节点的初始表，其中不是邻站的距离标为无穷大（不可达）。

共享

距离向量路由选择的总的思想是与邻站共享路由信息。尽管节点A不知道节点E，但节点C知道节点E，如果节点A共享节点C的路由表，那么节点A也知道如何到达节点E。另外，节点C不知道如何到达节点D，但节点A知道。如果节点C共享节点A的路由表，那么节点C也知道如何到达节点D。换言之，相邻站两个节点A和C如果彼此帮助，那么就可以改进它们的路由表。

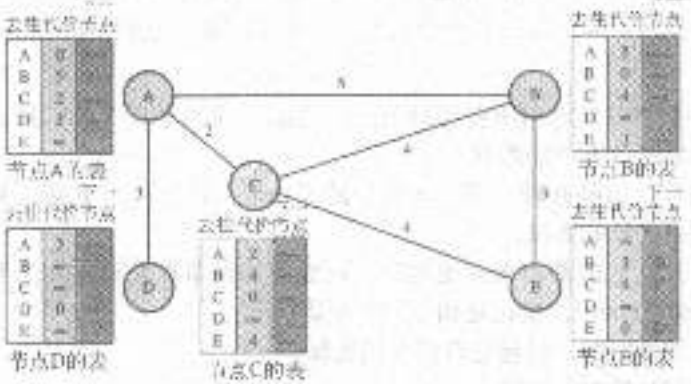


图22.15 距离向量路由选择的初始表

只还有一个问题，与每个邻站需共享表中多少项？一个节点并不知道邻站的表，解决的最好方法是每个节点与邻站发送它的完整表，让邻站决定哪些项目是有用的，哪些项目应丢弃的。但是表中第三列（下一节点）对邻站是没有用处的，因此，当邻站接收到一个表时，第三列用发送方的节点名替代。如果有一行可以用，则下一节点是表的发送节点。所以一个节点仅向任一邻站发送它的表中前二列。换言之，这里的共享是指仅共享前二列。

> 在距离向量路由选择中，每个节点与它的邻站周期性地或有变化时共享它的路由表。

更新算法

当一个节点从它的邻站接收到二列的表时，它需要更新它的路由表。更新用三个步骤：

1. 接收节点在表的第二列中的每一个值加上它与发送节点之间代价的值。这是显然的，因为节点C声称它到目的距离是x英里，而节点A和C的距离是y英里，则A与目的地的距离是x+y英里。

2. 如果接收节点使用来自任一行的信息，接收节点需要把发送节点名加入作为第三列，也就是发送节点作为路由的下一个节点。

3. 接收节点将修改过的接收到的表与它的旧表的相应行进行逐行比较。

a. 如果下一个节点项目不同，则接收节点选取具有最小代价的行，如果最小代价相同，则保持旧的。

b. 如果下一个节点项目相同，则接收节点选取新行。例如，假定节点C在前面已经告知到节点X的距离是3。现在假如节点C到节点X没有路径，即节点C现在告知这条路径距离为无穷大。节点A不能不理睬这值，即使它的旧项目较小。旧的路径不再存在，新的路径距离为无限大。

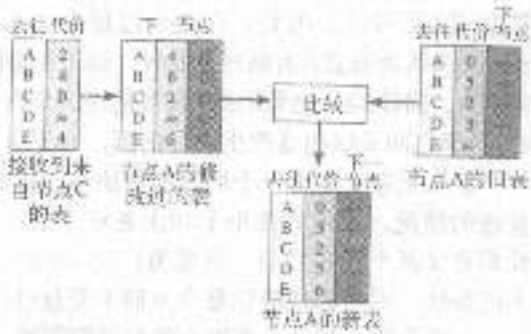


图22.16 距离向量更新

图22.16表示了节点A在接收到来自节点C的部分表后，如何更新它的路由表。

这里需要强调几点。第一，任何数加无穷大仍为无穷大。第二，修改过的表说明了如何从节点A经过节点C到达节点A。如果节点A需要通过节点C到达它自己，则必须去往节点C然后返回，其距离为4。第三，修改节点A的唯一好处是最后的项目，原先节点A不知道如何到达节点E（距离是无穷大），而现在知道通过

节点C代价是6。每一节点可用来自其他节点的表更新它的表。在短时间内，如果网络本身没有变化，如链路故障等，那么每一个节点达到一个稳定状态，它的表的内容保持不变。

何时共享

现在问题是一个节点何时向所有邻站节点发送部分的路由表（仅二列）？周期性地和当表有改变的时，就发送该表中的两列。

周期更新　通常，每隔30秒，节点发送它的路由表一次，定期更新。其更新时间依赖于所用的距离向量路由选择的协议。

触发更新　节点在它的路由表有变化时，向它的邻站节点发送它的二列路由表，这称为**触发更新**（triggered update）。变化是由下列的原因引起：

1. 节点接收到邻站的表，引起它自己表的更新；
2. 节点检测到邻站链路有故障。

两个节点循环不稳定性

距离向量路由选择的一个问题就是不稳定性，就是说使用这种协议的网络可能成为不稳定的。要理解这个问题，让我们考察图22.17的情况。

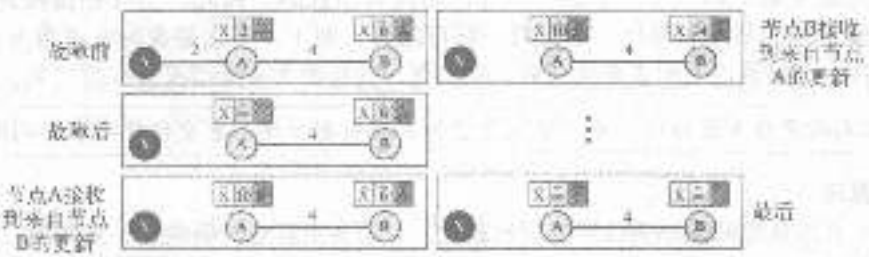


图22.17　两个节点不稳定性

图22.17表示一个有三个节点的系统。我们仅表示了与讨论需要有关的路由表部分。开始时，两个节点A与B都知道如何到达节点X。但突然A与X断开，节点A改变它的路由表。如果A立即发送它的表给B，一切都正常。但是，在B接收A的路由表之前，B已发送它的路由表。该系统变成不稳定。节点A接收B发来的路由表，它假定B已找到了到达X的路径，立即更新它的路由表。基于触发更新策略，A向B发送它的新更新的路由表。现在B认为A的周围环境已改变并更新它的路由表，于是到达X的代价逐渐地增加直到无穷大。在这时刻，A与B两者都知道X不可达。但是，在这个过程中，系统是不稳定的。节点A认为通过B有路径到达X，而节点B认为通过A有路径到达X。如果A接收到目的地为X的一个分组，它向B发送，然后返回到A。同样的，如果B接收到目的地为X的一个分组，它向A发送，然后返回到B。这样分组在A与B之间来回生存产生循环问题。对于这类不稳定性已经提出几个解决办法。

定义无穷大　第一个明显的解决办法是将一个较小的数值定义为无穷大，例如100。对于前述的情况，系统将在小于20次更新下是不稳定的。事实上，大多数距离向量路由协议的实现都定义两个相邻的节点距离为1，并认为16为无穷大。这就是说距离向量路由协议不能用于大的系统，网络的规模在每个方向不能超过15次跳。

分割范围　另一个解决办法是分割范围（split horizon）。在这种策略中，不是通过每一个接口发送整个表，而是每一个节点通过每一个接口仅发送它的表的一部分。按照它的表，如果节点B认为通过A到达X是最佳路径，则节点B不需要向A通知这一部分信息，因为这一信息来自A（A已知道），把来自节点A的信息修改后，发送回节点A可能会产生混淆。在前面图22.17情况下，节点B在向节点A发送它的表之前，就删除它的路由表中最后一行，节点A仍保

持到X的距离为无穷大。稍后，当节点A发送它的路由表到节点B时，节点B也修改它的路由表。在第一次更新后，系统成为稳定，两个节点A和B都知道X是不可达的。

分割范围与毒性逆转 使用分割范围策略有一个缺点，距离向量路由协议经常使用定时器。如果没有关于一条路径的新信息，那么节点就在它的表中删除该路径。在前面情况下，当节点B删除它向节点A的通知中到X的路径时，节点A不能推断这是由于分割范围策略（信息源是A），还是因为节点B最近没有接收到关于X的消息。分割范围策略可与毒性逆转（poison reverse）策略结合起来。节点B依旧可通知X的值，但如果信息源是节点A，则它用无穷大表示距离作为警告"不要使用这个值，我知道的这个路径来自你。"

三个节点不稳定性

三个节点不稳定性可用分割范围策略与毒性逆转策略结合起来避免。但是，如果不稳定性存在于二个节点间，则稳定性不能保证。如图22.18所示。

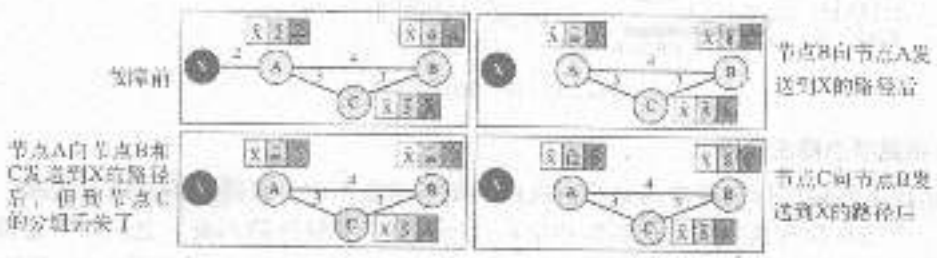


图22.18 三个节点不稳定性

假设发现节点X不可达后，节点A发送一个分组给节点B和C通知它们路径的情况。节点B立即更新它的表，可是到C的分组在网络中丢失，因此永远到达不了C。节点C根本不知道，仍认为有一条通过A到达X的距离为5的路径。稍后，节点C向B发送它的路由表，其中包含有到X路径，节点B此时完全糊涂了。它接收到从C到X的路径信息并按照算法更新它的路由表，其中从C到达X的代价为8。这个信息来自C而不是A，稍后节点B向节点A通知这条路径。节点A糊涂了并更新它的路由表，指出A通过B可达到X，其代价为12。当然，这个循环在继续。现在节点A对C而不对B通知到达X的路径增加了代价，然后节点C用增加的代价向节点B通知该路径，节点B向节点A重复相同的操作等，当每个节点的代价到达无穷大时，循环停止。

RIP

路由选择信息协议（Routing Information Protocol）是一个在自治系统内部使用的域内路由选择协议。它是基于距离向量路由选择的一个非常简单的协议。RIP基于下列考虑直接地实现距离向量路由选择：

1. 在一个自治系统中，包括了路由器和网络（链路）。路由器有路由表，而网络没有路由表；

2. 路由表中的目的端这一列是网络，这表示它的第一列定义了目的网络地址；

3. RIP所用的度量很简单，距离定义为到达目的端的链路（网络）个数。因此，RIP的度量称为**跳数**（hop count）；

4. 16就定义为无穷大，就是说在使用RIP的任何自治系统中，任何路径不能大于15跳；

5. 下一个节点这一列定义为被发送分组所要到达的目的路由器的地址。

图22.19表示了一个具有7个网络和4个路由器的自治系统，也表示了每个路由器的路由表。让我们考察R1的路由表。该表有7个项目，表示了在这个自治系统中如何到达每个网络。路由器R1直接连接到网络130.10.0.0和130.11.0.0，这就是说对于这两个网络没有下一跳项目，向

左边远方3个网络之一发送一个分组，路由器R1需要R2传递该分组。对这3个网络的下一跳项目是IP地址为130.10.0.1的R2接口。向右边远方2个网络发送一个分组，路由器R1需要IP地址为130.11.0.1的路由器R4。其他表可相似地解释。



图22.19　使用RIP的区域例子

22.3.4　链路状态路由选择

链路状态路由选择与距离向量路由选择的基本原理不同。在链路状态路由选择中，区域中的每一个节点拥有该区域的全部拓扑结构（所有节点和链路的列表，它们如何连接包含类型、代价也就是度量和链路的接通或断开的情形），根据这个拓扑结构，一个节点可用Dijkstra的算法（Dijkstra's algorithm）建立一个路由表。图22.20表示了这一概念。

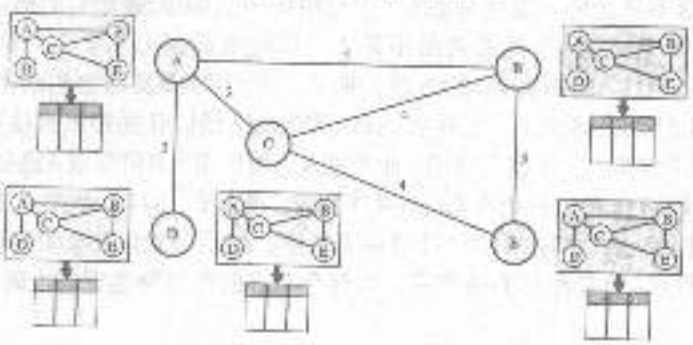


图22.20　链路状态路由选择的概念

该图显示了具有5个节点的简单区域，每个节点使用相同的拓扑建立各自的路由表，但是由于计算是基于该拓扑的不同解释上，所以每个节点的路由表是唯一的。这类似于一个城市的地图，每个人都可有同样的地图，但到达他特殊目的地却有不同的路径。

拓扑必须动态地表示每一节点最近的状态和链路。如果网络中的任一点有改变（例如，某一链路断开连接），则必须更新每个节点的拓扑。

如何使公共拓扑成为动态的以及如何存储在每个节点中？没有一个节点在开始时或在网络上的某处发生变化后就能知道该拓扑。链路状态路由选择是基于下面的假设，虽然不知道拓扑的全局知识，但每个节点知道自己这一部分的链路状态（如类型、状态和代价）。换言之，整个拓扑可由每一个节点的部分知识复合而成。图22.21的区域与图22.20的区域相同，但它包含了每个节点部分的知识。

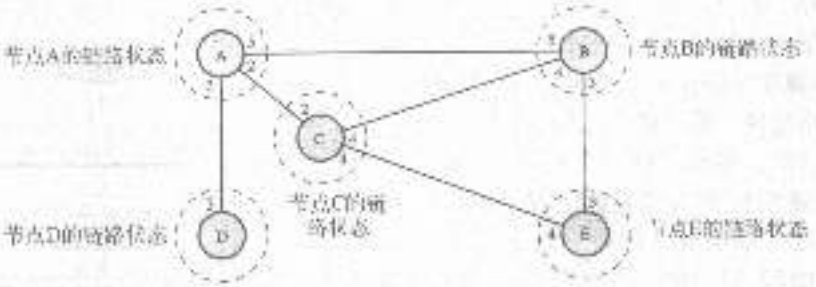


图22.21 链路状态范围

节点A知道它与节点B连接的度量为5，与节点C连接的度量为2，与节点D连接的度量为3。节点C知道它与节点A连接的度量为2，与节点B连接的度量为4，与节点E连接的度量为4。节点D知道它与节点A连接的度量为3，等等。虽然知识有重叠部分，但这个重叠部分保证了公共拓扑的建立（对于每一个节点的整个区域图）。

建立路由表

在链路状态路由选择（link state routing）中，为了保证每个节点对于其余每一节点都具有最小成本的路由表，需要做4件事：

1. 按每个节点建立称为链路状态分组（LSP）的链路状态；
2. 用一种有效而可靠的方法向其他每个路由器扩散LSP，这称为洪泛（flooding）；
3. 为每个节点构成一个最短路径树；
4. 基于最短路径树计算路由表。

链路状态分组（LSP）的生成 链路状态分组可携带大量信息。但是，我们假定它携带最小的数据量：节点的标识、链路的清单、序列号和寿命。前两个是节点标识和链路清单。它们是生成拓扑所需要的。第3个是序列号，它用于洪泛和区别新与旧的LSP。第4个是寿命，它防止旧的LSP在区域中长期保留。在两种情况下产生LSP：

1. 当区域的拓扑有变化时，触发LSP扩散是快速通知区域中任一节点更新它的拓扑主要方法；
2. 基于周期性产生。在这种情况下，其周期比距离向量路由选择的周期更长些。事实上，对于这种类型的LSP扩散没有实际的需要。它是保证旧的信息从区域中除去。因为周期的扩散以实现为基础，定时器设定的范围通常是 60 分或 2小时。一个比较长的周期保证洪泛在网络上不会产生太多的通信量。

LSP洪泛法 一个节点已准备好LSP之后，它必须向其余所有的节点扩散，不仅只是邻站。这个过程称为洪泛法，而洪泛方法按下列几步进行：

1. 创建节点的LSP，并从每个接口发送LSP的副本；
2. 接收到LSP的每个节点与已有的副本比较。如果新到达的LSP是比较旧的（用列序号检验），则丢弃该LSP。如果是比较新的，则该节点执行下面的工作：

a. 它丢弃旧的，保留新的；

b. 节点通过它的每个接口，但除掉接收分组的接口，再发送分组的副本。这将保证了洪泛在区域的某处停止（此时仅有一个接口）。

最短路径树的构成：Dijkstra算法 接收到所有的LSP后，每个节点就有了完整的拓扑副本，但是对到其余每个节点找出最短距离还是没有完成，需要有**最短路径树**（shortest path tree）。

树是由节点与链路组成的图，其中有一个节点称为根，所有其他节点都可从根开始通过

唯一的一条路径到达。最短路径树是根与其他节点之间路径最短的一个树。对每个节点，我们需要有该节点为根的最短路径树。

Dijkstra算法（Dijkstra algorithm）是从图创建一个最短路径树。算法将节点划分成两个集合：试验的和永久的。算法寻找一个现行节点的一些邻站节点，将它们作为试验的，检查它们。如果它们能通过一些准则，则将它们变为永久的。我们可以用图22.22中的流程图形式化地定义该算法。

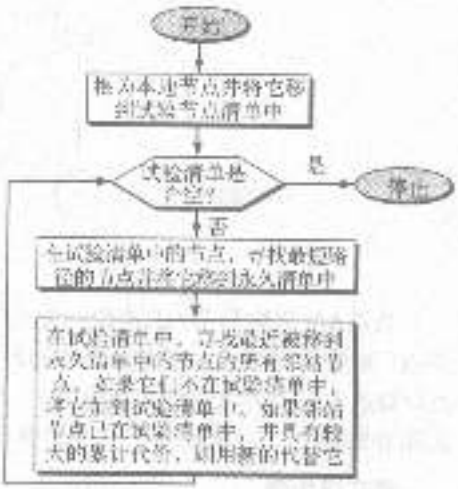


图22.22　Dijkstra算法

让我们用该算法应用于节点A，如图22.23所示。在每步中寻找最短路径，我们需要计算从根到每个节点的累计代价，已显示在节点旁边。

下面表示这些步骤，在每步结束，我们表示了的永久（已决定的）节点和试验（未决定的）节点以及累计代价。

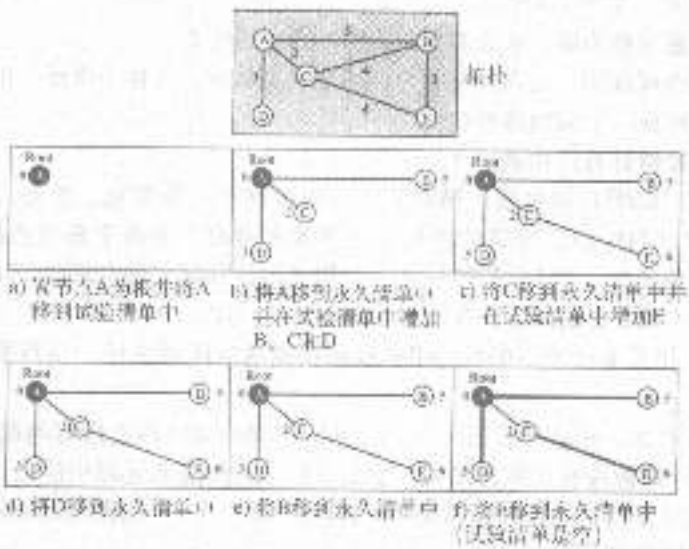


图22.23　最短路径树构成的范例

1. 我们设置节点A为树的根，并将它移到试验清单中。两个清单是：

永久清单：空　　试验清单：A（0）

2. 节点A与试验清单中的所有节点有最短的累计代价，我们将A移到永久清单中，并在试验清单中增加A的所有邻站，两个新的清单是：

永久清单：A（0）　　试验清单：B（5），C（2），D（3）

3. 在试验清单中的所有节点，节点C有最短累计代价，我们将C移到永久清单中，节点C有3个邻站，但节点A已处理，那些没有处理的是B和E，但是B已在试验清单中而累计代价为5，节点A也可经过C到达节点B，其代价为6，因为5小于6，我们在试验清单仍保留代价为5的节点B而不被代替，两个新的清单是：

永久清单：A（0），C（2）　　试验清单：B（5），D（3），E（6）

4. 在试验清单中的所有节点，节点D有最短累计代价，我们将D移到永久清单中，节点D

没有未处理的邻站增加到试验清单中，两个新的清单是：

永久清单：A (0)，C (2)，D (3)

试验清单：B (5)，E (6)

5. 在试验清单中的所有节点，节点B有最短累计代价，我们将B移到永久清单中。我们需要增加B的邻站的所有未处理的节点到试验清单中（这个节点正是E），但是，E (6) 已在该清单中并具有最小的累计代价，作为B的邻站E，它的累计代价是8，所以我们仍保留节点E (6) 在试验清单中。两个新的清单是：

永久清单：A (0)，B (5)，C (2)，D (3)

试验清单：E (6)

表22.2 节点A的路由表

节点	代价	下一个路由器
A	0	
B	5	—
C	2	
D	3	—
E	6	C

6. 在试验清单中的所有节点，节点E有最短累计代价，我们将E移到永久清单中，节点E没有邻站。现在试验清单为空，我们停止处理，最短路径树已构成，最后的清单是：

永久清单：A (0)，B (5)，C (2)，D (3)，E (6) 试验清单：空

从最短路径树计算路由表　每个节点用最短路径协议构造它的路由表，路由表说明了从根到每个节点的代价。表22.2表示了节点A的路由表。

将表22.2与图22.14中节点A的路由表相比较，对于节点A，距离向量路由选择与链路状态路由选择的最终结果相同。

OSPF

开放最短路径优先或OSPF协议（Open Shortest Path Frist protocol）是基于链路状态路由选择的一个域内路由选择协议。它的域也是一个自治系统。

区域　为了有效地和及时地处理路由选择，OSPF将自治系统划分为一些区域。一个区域（area）是包含在自治系统中的一些网络、主机和路由器的集合。自治系统又可划分为多个不同的区域，在一个区域里所有网络必须是互相连接的。

在一个区域内的路由器都使用洪泛法传送路由选择信息。在一个区域的边界，称为**区域边界路由器**（area border router）的一些特殊的路由器，将有关本区域的信息概括起来发送给其他区域。在自治系统中有一个特殊区域称为主干，在自治系统中的所有区域必须连接到主干上。换言之，主干相当于主区域，而且其他区域相当于从区域。但是，这并不表示在各区域内的路由器不能相互连接。在主干中的路由器称为**主干路由器**（backbone router）。注意：一个主干路由器也可以同时是一个区域的边界路由器。

如果由于某些问题，在主干和区域之间的连通性被破坏了，那么网络管理员就必须在路由器之间建立一条**虚链路**（virtual link），以保持作为主区域的主干的各种功能的连续性。

每一个区域有一个区域标识。主干的区域标识是0。图22.24显示了一个自治系统及其他的区域。

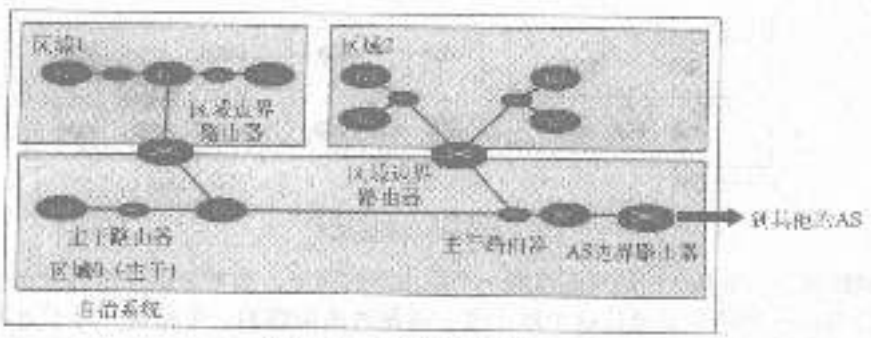


图22.24 自治系统中的区域

度量　OSPF协议允许网络管理员给每一条路由指定一个代价，称为度量（metric）。度量可以基于服务类型（最小延迟、最大吞吐量等）。事实上，一个路由器可以有多个路由表，而每一个路由表基于不同的服务类型。

链路类型　在OSPF术语中，一个连接称为链路。已定义了4种类型的链路：点对点链路、过渡链路、残桩链路和虚链路（见图22.25）。



图22.25　链路的类型

点到点链路（point-to-point link）连接两个路由器，而中间没有任何其他的主机或路由器。换言之，该链路（网络）的目的就是为了连接两个路由器。这种链路类型的例子就是两个路由器用一条电话线（或一条T线）连接起来。没有必要给这种类型的链路指定一个网络地址。用图表示时，路由器用节点表示，而链路用一条连接两个节点的双向边来表示。度量表示在两端，各表示每一个方向的度量，通常它们是一样的。即每一个路由器只有一个邻站在链路的另一端（见图22.26）。

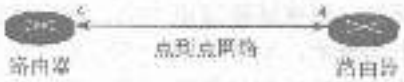


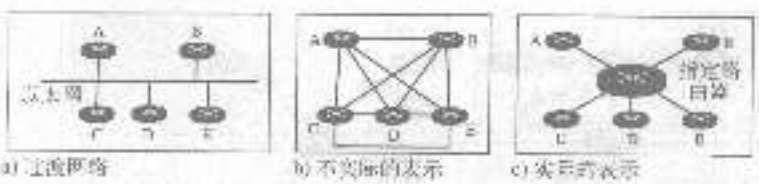
图22.26　点到点链路

过渡链路（transient link）是一种连接多个路由器的网络。数据可以从任何一个路由器进入网络，并从任何一个路由器离开网络。所有局域网和某些具有两个或更多的路由器的广域网都属于这种类型。在这种情况下，每一个路由器都有好几个邻站。例如，考虑图22.27a中的以太网，路由器B的邻站是路由器A、C、D和E。如果我们要表示这种情况下的邻站关系，就使用图22.27b。

这种链路既低效又不实际，低效是因为每一个路由器需要通知其他4个邻站的路由器，总共有20个通知；不实际是因为在每一对路由器之间没有一个单独的网络（链路），只有一个网络用做所有这5个路由器之间的交叉路口。

为了表示出每一个路由器是通过一个单独网络连接到其他的每一个路由器，我们将这个网络用一个节点来表示。但是，网络并不是一个路由器，它不能像路由器那样工作，因此，在网络中某一个路由器就要负起这个责任。这个路由器有双重作用，它是一个真的路由器，又是一个指定路由器，用图22.27c的拓扑结构可以表示过渡网络的连接。

一方面现在每一个路由器只有一个邻站，即指定路由器（网络），另一方面这个指定路由器（即网络）有5个邻站。我们可看到，向邻站通知的数目由20减少到10。另外，链路用连接节点的双向边来表示。但是，虽然从每一个节点到指定路由器都有一个度量，但从指定路由器到其他任何节点都没有度量，原因就是指定路由器代表了网络，我们不能为它计算两次。当一个分组进入网络时，我们指定一个代价，当分组离开网络到其他路由器时，就没有代价。



a) 过渡网络　b) 不实际的表示　c) 实际的表示

图22.27　过渡链路

残桩链路（stub link）是只连接到一个路由器的网络。数据分组通过这个单一的路由器进入网络，而离开网络也是通过这个路由器。这是过渡网络的一个特例。对于这种情况，可以将路由器表示为一个节点，而用指定路由器表示这个网络。但是，链路是单向的，即从路由

器到网络（见图22.28）。

当两个路由器之间的链路断开时，网络管理员就在它们之间使用一条更长的路径，它可能经过好几个路由器，创建一条虚链路（virtual link）。

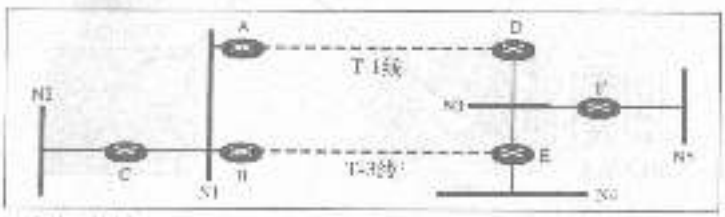


a) 残桩网络　　b) 表示法

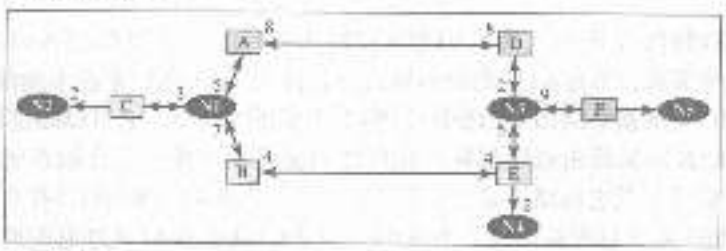
图22.28 残桩链路

图形表示 现在让我们考虑一个自治系统如何用图形表示。图22.29表示了具有7个网络和6个路由器的一个小型自治系统。有两个网络是点到点网络，我们使用记号如N1和N2表示过渡网络和残桩网络。没有必要给点到点网络指定一个标识符。该图也表示OSPF所见到的自治系统的图形。

我们用正方形节点表示路由器，用椭圆节点表示网络（用指定路由器表示的网络）。但是，OSPF将两者都看做节点。注意，我们共有3个残桩网络。



a) 自治系统例子



b) 图形表示

图22.29 一个自治系统的例子及其OSPF图形表示

22.3.5 路径向量路由选择

距离向量路由选择和链路状态路由选择都是域内路由选择协议，它们用于一个自治系统内部，而不是自治系统之间。因为可量测性，这两个协议都不适合域间路由选择。当操作区域变大时，这两个路由选择协议变得很难处理。如果在操作区域中存在多个跳数，距离向量路由选择协议是不稳定的，而链路状态路由选择协议需要有巨大的资源来计算路由表，由于洪泛也会产生严重的通信拥塞，因此，需要第三个路由选择协议，我们称它为路径向量路由选择（path vector routing）。

已经证明路径向量路由选择对区域之间的路由选择是有用的。它的原理与距离向量路由选择相似。在路径向量路由选择中，我们假定每个自治系统有一个节点的行为代表了整个自治系统（也可以多个，但对于我们概念性的讨论一个已足够）。该节点称为代言节点（speaker node）。

一个自治系统的代言节点生成一个路由表并通知相邻自治系统中的代言节点。除了每个自治系统仅有一个代言节点可以彼此通信外，它与距离向量路由选择的思想相同。但通知的内容是不同的，一个代言节点通知是在它的自治系统或其他自治系统中的路径，而不是节点跳数的度量。

初始化

在开始时，每个代言节点仅能知道它的自治系统内部节点的可达性。图22.30表示了4个自治系统组成的互联网络中每个代言节点的初始路由表。

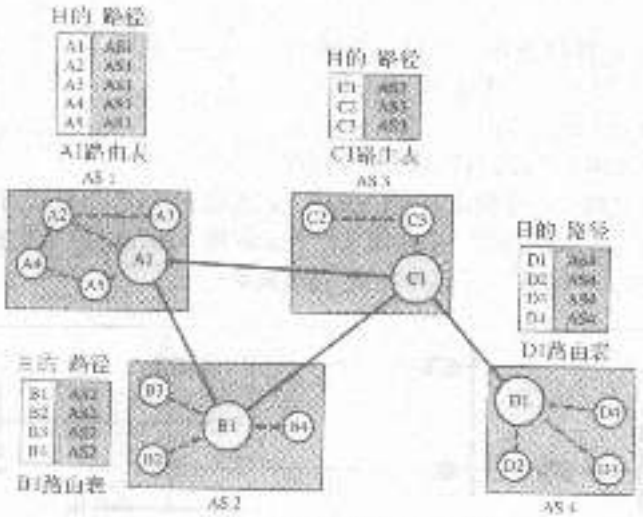


图22.30　路径向量路由选择的初始路由表

节点A1是AS1的代言节点，节点B1是AS2的代言节点，节点C1是AS3的代言节点，节点D1是AS4的代言节点。节点A1生成的初始路由表说明了A1到A5是在本地的AS1中，通过它可以到达。节点B1生成初始路由表通知B1到B4在本地的AS2中，它可以经过B1到达等。

共享　正如距离向量路由选择那样，在路径向量路由选择中，自治系统的一个代言节点与它邻站的代言节点共享它的路由表。在图22.30中，节点A1与B1及C1共享它的路由表，节点C1与节点D1及B1共享它的路由表，节点B1与节点C1及节点A1共享它的路由表，以及节点D1与C1共享D它的路由表。

更新　当一个代言节点从一个邻站接收到一个表列的路由表时，它更新它自己的路由表。更新内容包括增加不在表中的节点以及其自治系统与发送方的自治系统之间的路径。此后，每一个代言节点都有一个如何到达其他自治系统各个节点的路由表。图22.31表示系统稳定后，每一个代言节点的路由表。

按照图22.31，如果路由器R1接收一个到节点A3的分组，它知道这条路径是在AS1中（分组在本自治系统内）。但如果它接收一个到D1的分组，它知道这分组应该从AS1到AS2然后再到AS3。路由表完全表示了该路径。另外，如果AS4中的节点D1接收一个到节点A2的分组，它知道应该经过AS4，AS3和AS1。

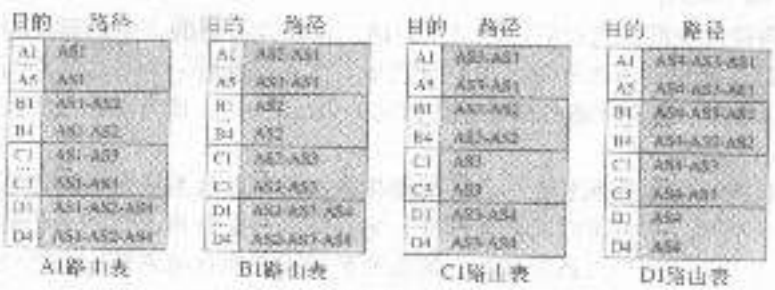


图22.31　三个独立系统的稳定表

- 预防回路。在路径向量路由选择中可以避免距离向量路由选择的不稳定性和产生回路。当路由器接收到报文时，它要检查它所在的自治系统是否在到目的网络的路由表中。如果是的话，就会产生回路，这个报文就被忽略。
- 策略路由选择。通过路径向量路由选择可以很易地实现策略路由选择。当路由器接收到报文时，它就检查其路径。如果在路径中所列出的某个自治系统不符合其策略，它就忽略这条路径和目的端。它不用这条路径更新其路由表，也不将这个报文发送给它的邻站节点。
- 优化路径。什么是在路径向量选择中最佳的路径？我们在自治系统中寻找一条到达目端的最佳的路径。因为包含路径的每个自治系统中，它们有不同度量准则。某个自治系统内部使用RIP，它是以跳数为度量，而另一个自治系统使用OSPF，它是以最小延迟为度量。所以在这个路由中不可能明确地含有这些度量。最佳路径是符合组织机构标准的路径。在图22.30中，每个自治系统到目的端可能有多条路径。例如，AS4到AS1可以是AS4-AS3-AS2-AS1，或者AS4-AS3-AS1。对于这些路径，我们选取经过自治系统数目最小的一条。但也不总是这种情况，还有其他准则，如保密、安全和可靠性。

BGP

边界网关协议（BGP）是使用路径向量路由选择的域间路由选择协议。它是1989年问世，现已有4个版本。

自治系统类型　正如以前我们说过，因特网被划分成多层次的域，称为自治系统。例如，一个大型的企业有它自己的网络并有完整的控制系统，它是一个自治系统。为本地客户提供服务的本地ISP也是一个自治系统。自治系统可分三大类：残桩的、多接口的和转送的。

- 残桩自治系统。它与其他自治系统只有一个连接。在该自治系统中，域间数据流量起始或终止于该自治系统，即其源IP地址或目的IP地址所指定的主机位于该自治系统中。但是，数据流量不能通过残桩自治系统。残桩自治系统或者是一个源端或者是一个接收器。好的例子是小型企业或小的本地ISP。
- 多接口自治系统。它与其他自治系统有多个连接，但它仍旧还是数据流量的源端或接收器。它能接收来自多个自治系统的数据流量，也能发送数据流量到多个自治系统，但没有过渡的数据流量。它不允许来自一个自治系统的数据经它转发到另一个自治系统。一个多接口自治系统的例子是一个连接到多个区域或国家自治系统的大型企业，不允许转送数据。
- 转送自治系统。它是一个多接口自治系统，但它允许过渡数据流量。过渡自治系统的例子是国家或区际ISP（因特网主干网）。

路径属性　在前面的例子中，我们讨论到目的网络的路径。这个路径可以用自治系统的列表来表示。但是，实际上是用属性的列表来表示的，每一个属性给出关于路径的一些信息。当使用它的策略时，属性列表可帮助接收信息的路由器做出更好的决定。

属性可分为两大类：熟知的和可选的。熟知属性（well-known attribute）是每一个BGP路由器必须知道的，而可选属性（optional attribute）则不需要被每一个路由器都知道。

熟知属性本身又可划分为两类：强制的和自选的。熟知强制属性是在一条路由的描述中必须出现的属性，熟知自选属性是每一个路由器必须知道的，但不一定需要包括在每一个更新报文中。一个熟知强制属性是ORIGIN，它定义路由选择信息的源端（RIP、OSPF等等）。另一个熟知强制属性是AS_PATH，它定义自治系统的列表，表中记录了通过那个自治系统可以到达目的端。还有一个熟知强制属性是NEXT-HOP，它定义分组必须发送到的下一个路由器。

可选属性也可以划分为两类：传递的和非传递的。可选传递属性是没有实现这个属性的路由器必须传递给下一个路由器的属性，可选非传递属性是如果接收路由器还没有实现的这个属性，则必须丢弃的属性。

BGP会话　使用BGP的两个路由器之间的路由信息的交换产生一次会话。会话是两个BGP系统仅为交换路由选择信息而建立的一次连接。为了可靠性，BGP使用TCP作为其传输层协议。换言之，作为一个应用程序，BGP级的会话是TCP级的一条连接。但是，对于BGP和其他应用程序所做的TCP连接略有不同。对于BGP建立的TCP连接可持续一段较长的时间直到某一不寻常的事件发生。因此，BGP会话有时称为半永久连接。

外部与内部BGP　再明确些，有两种类型的BGP会话：外部BGP（E-BGP）会话和内部BGP（I-BGP）会话。E-BGP会话用于属于两个不同自治系统的两个代言节点之间交换信息，而I-BGP会话用于一个自治系统内部的两个路由器之间交换路由信息。图22.32表示了这一思想。

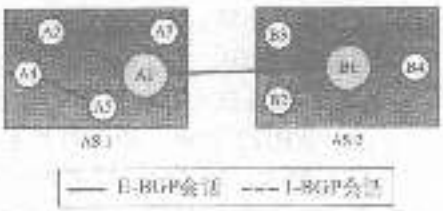


图22.32　内部和外部BGP会话

AS1与AS2之间建立的会话是E-BGP会话。两个代言路由器交换它们所知道的因特网中有关网络的信息。但是，这两个路由器需要从自治系统中的其他路由器收集信息。这是由I-BGP会话完成。

22.4　多播路由选择协议

在这一节，我们讨论多播与多播路由选择协议。首先定义术语多播（multicasting），并将它与单播和广播相比较，还简要介绍多播的应用。最后，我们转向讨论多播路由选择以及它的一般思想和相关目的。我们还会讨论当今因特网常用的多播路由选择协议。

22.4.1　单播、多播与广播

一个报文可以单播、多播或广播。当这些术语涉及因特网时，让我们清晰地说明这些术语。

单播

在单播通信中，只有一个源端和一个目的端，源端和目的端的关系是一对一的。在这种类型的通信中，IP数据报中的源地址和目的地址都是分配给主机（更确切地说是主机端口）的单播地址。如图22.33所示，一个单播分组从源端S1开始，经过路由器到达目的端D1。为了简化该图，可将网络视为路由器之间的一条链路。

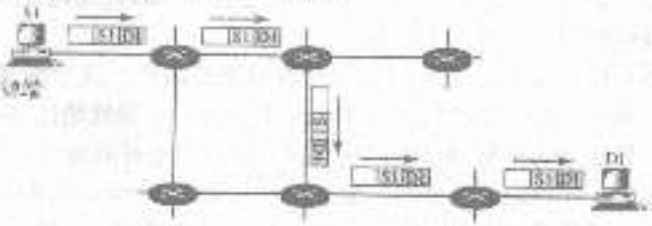


图22.33　单播

注意：在单播（unicast）中，当路由器接收到一个分组时，它仅从路由选择表所定义的端口中的一个（属于最优路径上的一个）将分组转发出去。如果路由器不能在路由选择表中找到目的地址，它就可能将分组丢弃。

在单播中，路由器将接收到的分组仅从其端口中的一个转发出去。

多播

在多播通信中，存在一个源端和一组目的端，其关系是一对多的。在这类通信中，源地址是一个单播地址，而目的地址是一个组地址。组地址定义了一个或多个目的地址，该组地址标识了该组的成员。图22.34表示了多播的思想。

一个多播分组从源端S1发出传向属于组G1的所有目的端。在多播传输中，当路由器接收到一个分组时，它可能通过它的多个端口将其转发出去。

> 在多播中，路由器可能通过它的多个端口将其所接收的分组转发出去。

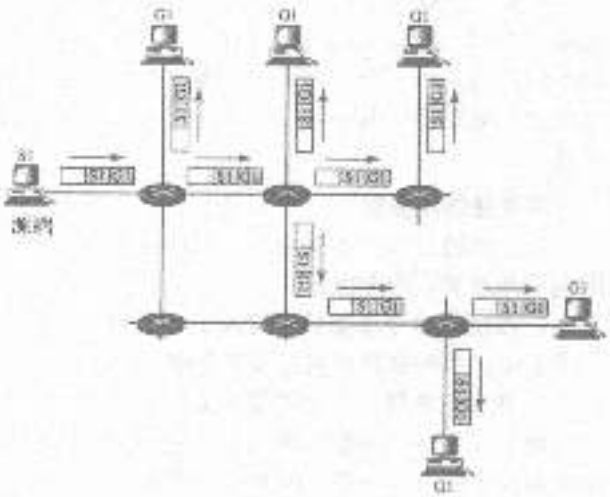


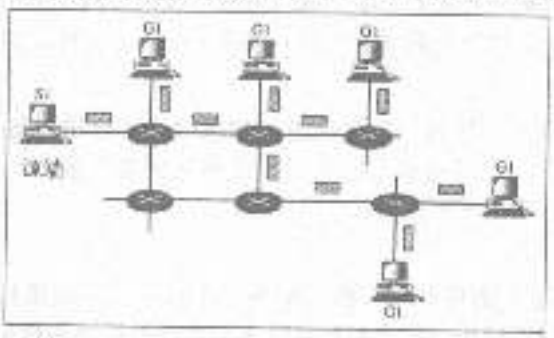
图22.34 多播

广播

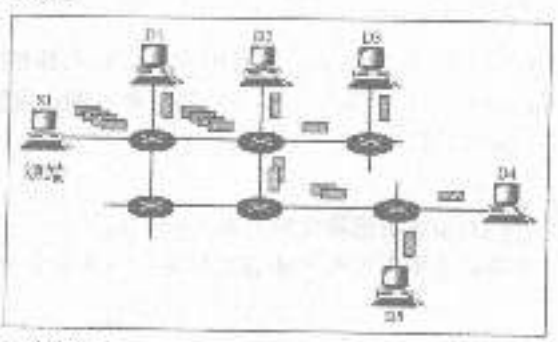
在广播通信中，源端与目的端的关系是一对所有的，源端只有一个，而目的端是其余所有的主机。因特网没有明确地支持**广播**（broadcasting），因为它会产生大量的通信量，并需要高带宽。设想一下，如果一个人想给与因特网相连的其他每个人都发送一条报文，那么在因特网上会产生多大的通信量。

多播与多个单播

本节结束之前，我们必须区别多播与多个单播，这两个概念用图22.35说明。



a) 多播



b) 多个单播

图22.35 多播与多个单播

多播是从源地址以一个单个分组发出，它被路由器复制。每个分组中的目的地址对所有的副本是相同的。注意：在任何两个路由器之间只有分组的一个副本。

在多个单播（multiple unicasting）中，从源端发出多个分组。例如，如果有5个目的地址，源端发送5个分组，每个分组具有不同的单播目的地址。注意：在两个路由器之间可能有多个副本在传送。当一个人发送一封电子邮件给一组人，这就是多个单播。电子邮件软件创建报文的多个复制件，其中每一个具有不同的目的地址并逐一发送它们。这不是多播，而是多个单播。

用单播仿真多播

人们可能会问当多播可用单播仿真时，为什么对多播还要用一个单独的机制。对于这个问题有两点明显的理由：

1. 多播比多个单播效率更高。在图22.35中，我们看到多播比多个单播要求带宽小，在多个单播中，某些链路必须处理多个副本。

2. 在多个单播中，由源端生成的一些分组，这些分组之间有相对的延迟。如果有1 000个目的地址，第一个分组与最后一个分组之间的延迟可能是不可接受的。如果在多播中，因为由源端仅生成一个分组，因此没有延迟。

> 用单播仿真多播效率不高并可能产生长的延迟，特别是对于大的组。

22.4.2 应用

当今多播有许多应用，如访问分布式数据库（distributed database）、信息发布、电话会议和远程学习。

访问分布式数据库

当前大多数数据库都是分布式的，即信息通常在生成时存储在多个地方。需要访问数据库的用户不知道信息的地址。用户的请求是向所有数据库的地方多播，而有该信息的地方响应。

信息发布

商业机构时常需要向它们的客户发送信息。如果信息的性质对每个客户都相同，则它可以多播。采用这种方式，一个商业机构可发送一个报文到多个客户。例如，可向购买某个特殊软件包的所有客户发送一个更新软件。

新闻传播

同样的，新闻可通过多播简单地传播。单独一个报文可发送给特殊感兴趣的那些人员。例如，中学篮球冠军巡回赛的统计结果可被送给多份报纸的运动编辑记者。

电话会议

电话会议（Teleconferencing）包含多播。所有出席会议的人都在同一时间接收到相同的信息。为此，可构成临时组或永久组。例如，每个星期一举行例行会议的工程组可编为永久组，而计划休假人员的组可编为临时组。

远程学习

多播使用中一个正在成长的领域是远程学习（distance learning）。某一教授讲的课可被一个特定组的学生接收到。这特别适用于那些不能到大学课堂听课的学生。

22.4.3 多播路由选择

在这一节，我们首先讨论所有多播协议的共同部分——最佳路由选择。然后，我们给出

多播路由选择协议的概况。

优化路由选择：最短路径树

最佳域内路由选择过程最后形成的结果是寻找最短路径树。树的根是源端，而叶子都是可能的目的端。从根到每个目的端的路径是最短路径。但是，在单播中树的数目和树的构成与多播路由选择是不同的。让我们分别加以讨论。

单播路由选择 在单播路由选择中，当路由器接收到一个要转发的分组时，它需要寻找到分组的目的端的最短路径。路由器在路由表中查询那个特定的目的地址，对应于该目的地址的下一跳是最短路径的起始点。路由器知道到达每个目的端的最短路径。这说明路由器有一个最佳地到达所有目的地址的最短路径树。换言之，路由表的每一行是一条最短路径，而整个路由选择表是一个最短路径树。在单播路由选择中，每个路由器需要有唯一的最短路径树转发分组，然而每个路由器有它自己的最短路径树。图22.36显示了这样的情形。

图22.36显示了路由器R1路由表和最短路径树。路由表中的每一行对应于从根到相应网络的一条路径。整个路由表代表了最短路径树。

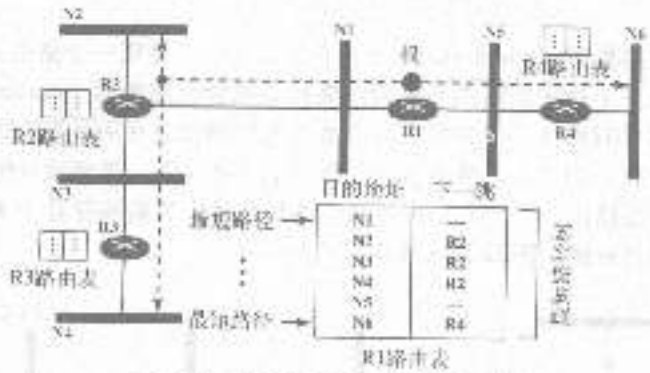


图22.36 单播路由选择中的最短路径树

> 在单播路由选择中，互域中的每一个路由器都有一张表，该表定义了到可能目的地址的一棵最短路径树。

多播路由选择 当路由器接收到一个多播分组时，情形与接收到一个单播分组时有所不同。一个多播分组可能有多个网络中的目的地址。向组中的成员转发单个分组要求一个最短路径树。如果有n个组，那么我们需要n个最短路径树。可以想像多播路由选择的复杂性。为了解决这个问题，已使用两种方法：基于源树和组共享树。

- 基于源树。在基于源树（source-based tree）方法中，每个路由器要求每组都要有一个最短路径树。一个组的最短路径树定义了包含那个组的忠实成员的每个网络的下一跳。在图22.37中，我们假定在互域中只有5个组：G1、G2、G3、G4和G5。此时，G1在4个网络上有忠实成员，G2在3个网络上有忠实成员，G3在2个网络上有忠实成员，G4在2个网络上有忠实成员以及G5在2个网络上有忠实成员。我们已在每个网络上表示了忠实成员所在组的名。图22.37也表示了路由器R1的多播路由表。每个组有一个最短路径树，所以5个组有5个最短路径树。如果路由器R1接收到目的地址为G1的一个分组，那么它需要将分组的一个副本发送给与它有连接的网络，向路由器R2发送一个副本，向路由器R4发送一个副本。这样G1的所有成员都能接收到一个副本。在这种方法中，如果组的个数是m，那么每个路由器需要有m个最短路径树，每组一个。我们可以想像，如果有

成百或成千的组，路由表将是多么复杂。然而，我们将说明不同的协议将如何处理以减小其复杂性。

在基于源树方法中，对每个组每个路由器都需要有一个最短路径树。

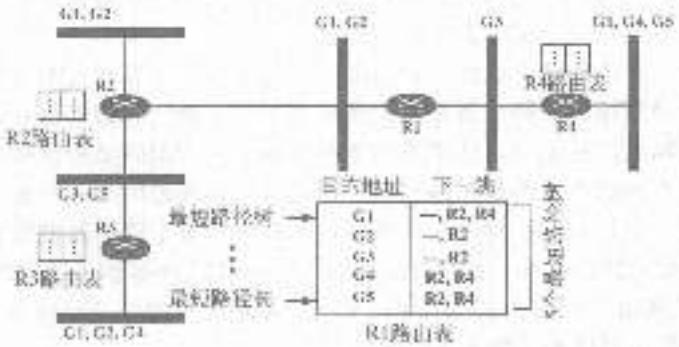


图22.37　基于源树方法

- 组共享树。在组共享树（group-shared tree）方法中，不是每一个路由器都具有m个最短路径树，而只是一个指定的路由器称为中央核心或会合路由器（rendezvous router）担任分配多播通信量的责任。中央核心路由器在它的路由表中有m条最短路径。区域中其余的路由器都没有。如果一个路由器接收到一个多播分组，它将该分组封装在单播分组中，并发送给核心路由器。核心路由器从它的封装体中重新恢复这个多播分组，并查询它的路由表路由该分组。图22.38表示这个思路。

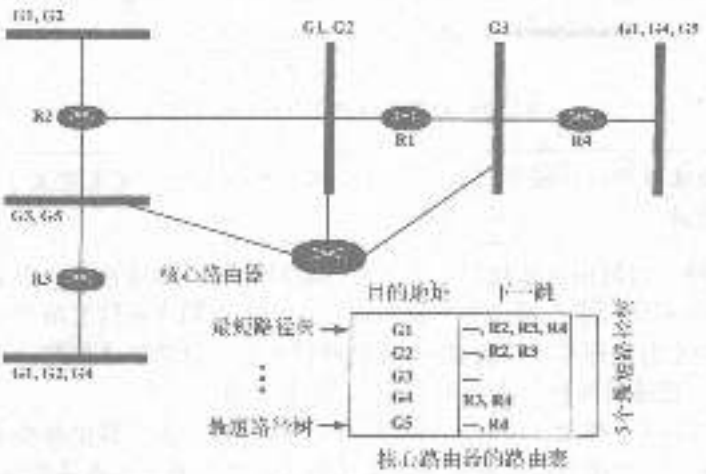


图22.38　组共享树

在组共享树方法中，只有一个核心路由器，它对多播所涉及的每一个组有一个最短路径树。

22.4.4　多播路由选择协议

在最近十几年间，已经出现一些多播路由选择协议。其中有些是单播路由选择协议的扩展，另一些全是新开发的。

在本章的余下部分就讨论这些多播路由选择协议。图22.39显示了这些协议的分类。

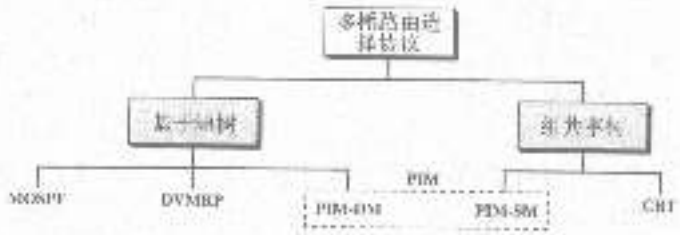


图22.39 常用的多播路由选择协议的分类

多播链路状态路由选择：MOSPF

这一节简要讨论多播链路状态路由选择及其在因特网中的实现，MOSPF。

多播链路状态路由选择 我们在22.3节中讨论了单播链路状态路由选择，我们说每个路由器用Dijkstra算法创建一个最短路径树，路由表是最短路径树的一种转换形式。多播链路状态路由选择是单播路由选择的直接扩充，它使用一个基于源树方法。尽管它完全包含单播路由选择，但多播路由选择的扩充部分是很简单的和直接的。

多播链路状态路由选择使用基于源树方法。

记住在单播路由选择中，每个节点需要通告它的链路状态。对于多播路由选择来说，每个节点要修正状态的解释，一个节点通知链路上有忠实成员的每个组。此时，状态的意思是"在链路上哪些组是活动的"。有关组的信息来自IGMP（见第21章），每个运行IGMP的路由器请求链路上主机寻找成员状态。

当路由器接收到所有LSP时，它用Dijkstra算法创建n个最短路径树，从中创建n个拓扑结构（n是组的个数）。于是，每个路由器有一个路由表，该表代表与组一样多的最短路径树。这种协议要解决的唯一问题是创建和保存多个最短路径树所需的时间和空间。解决办法是仅当需要时才创建树。当路由器接收到具有多播目的地址的分组时，它运行Dijkstra算法计算那个组的最短路径树。其结果被高速缓存，以防万一到那个目的地址还有其他的分组出现。

MOSPF 多播开放最短路径优先（Multicast Open Shortest Path First，MOSPF）协议是对OSPF协议的一个扩充，它运用多播链路状态路由选择创建基于源树策略的树。该协议增加了一个新的链路状态更新分组，将一个主机的单播地址与组地址或该主机提供的地址关联起来。这个分组被称为组成员关系LSA，用这种方法，能够在树中只包括属于一个特定组的主机（用它们的单播地址）。换句话说，可以生成一个包含属于一个组所有主机的树，但在计算时使用主机的单播地址。为了提高效率，路由器按需求计算最短路径树（当它接收到第一个多播分组时）。此外，能将该树存储在高速缓存中，以备将来相同的源/组对使用。MOSPF是一种数据驱动（data-driven）协议。当MOSPF路由器第一次发现一个具有给定源和组地址的数据报时，该路由器就计算Dijkstra最短路径树。

多播距离向量：DVMRP

在这节中，我们简要讨论多播距离向量路由选择及其在因特网中的实现，DVMRP。

多播距离向量路由选择 单播距离向量路由选择是很简单的，扩充它支持多播路由选择是复杂的。多播路由选择不允许一个路由器向它的邻站发送它的路由表。它使用单播距离向量路由表的信息从头创建一张路由表。

多播距离向量路由选择使用基于源树，但路由器从不真正创建路由表。当路由器接收到多播分组时，它转发该分组好像它正在查询一张路由表一样。我们可以说最短路径树是短暂的，它使用后（转发分组后），该表就消失了。

为了达到这个目的，多播距离向量算法是基于4个决策的一个过程。每个策略建立在前一个策略之上，并改进前一个存在的缺点。我们依次说明并观察每个策略是如何改进的。

- 洪泛。洪泛是第一个策略。一个路由器接收到一个分组，除该接收接口外它甚至不查找目的组地址就向它的所有接口发送该分组。洪泛实现多播目的：具有活动成员的每个网络都接收到该分组，而没有活动成员的网络也接收到该分组。这是广播而不是多播。这就出现另一个问题：它会形成回路。一个分组离开路由器，它可能从另一个接口或同一接口重新返回，再次被转发。有些洪泛协议为了避免回路暂时保存分组副本一段时间，然后就丢弃。下一个策略是逆路径转发，纠正这个缺点。

洪泛策略广播分组，但在系统中形成回路。

- 逆路径转发（RPF）。逆路径转发是洪泛策略的一个修改版。为了防止回路的形成，只转发一个副本而将其他副本都丢弃。在RPF中，路由器只转发那个已经通过从源端到路由器的最短路径进行传输的副本。为了找到这个副本，RPF使用了单播路由选择表。路由器接收到一个分组，并从该分组中提取出源地址（一个单播地址）。路由器查询它的单播路由表，就好像它要发送一个分组到这个源地址。路由表会告诉路由器下一跳地址。如果到达的多播分组正好来自于路由表中所定义的下一跳地址，那么该分组是沿着从源端到路由器的最短路径传输的。这是因为在单播距离向量路由选择协议中，最短路径的反方向的。如果A到B的路径是最短的路径，那么B到A的路径也是最短的。如果分组是沿着最短路径传输，则路由器转发它。否则丢弃它。

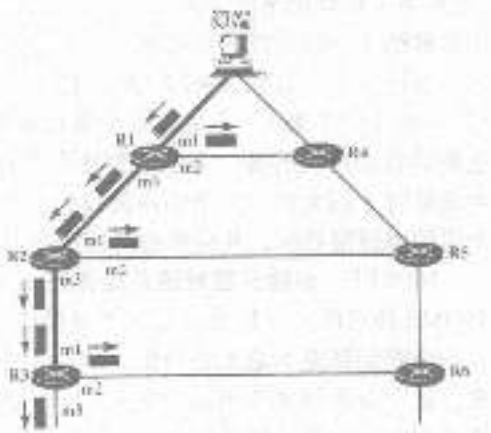


图22.40　逆路径转发（RPF）

这种策略防止回路形成，因为从源到路由器总是沿着最短路径传输。如果一个分组离开后又返回，显然这不是最短路径。图22.40清楚地说明了这一点。

图22.40表示了部分区域和源端。计算路由器R1、R2和R3的最短路径，用粗线表示。当R1通过接口m1接收到来自源端的一个分组时，它查询它的路由表并找到从R1到源端的最短路径是通过接口m1，转发该分组。但是，如果通过接口m2到达的分组一个副本，则丢弃。因为m2不是R1到源端的最短路径。对R2和R3也一样。读者可能会问，如果一个分组副本到达R3的m1接口，经过R6、R5、R2传输，然后通过接口m1进入R3将会发生什么？这个接口的确是R3的接口，该分组的副本是否被转发？回答是在这种情形下什么也不会发生，因为当分组从R5到R2传输时，它将被R2丢弃，永远不会到达R3。源端的上行路由器总是丢弃那些不是通过最短路径传输的分组，因此避免了下行路由器的混淆。

RPF消除了洪泛过程的回路。

- 逆路径广播（RPB）。RPF能确保每个网络接收一个多播分组的副本而不构成回路。然而，它不能保证每个网络只接收一个副本（一个网络可能接收两个或多个副本）。其原因是，RPF并不是基于目的地址（组地址）的，而转发是基于源地址的。为了直观地了解这个问题，让我们考察图22.41。

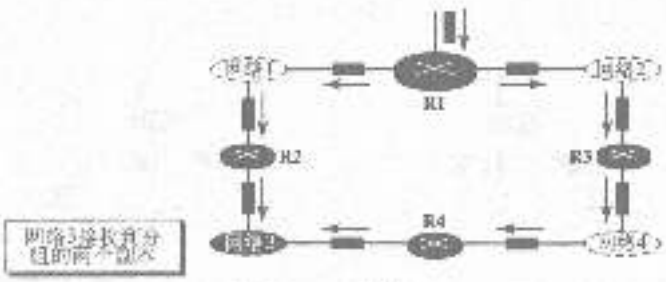
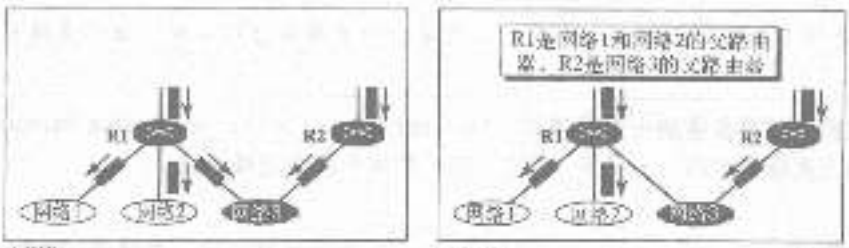


图22.41 RPF的问题

在图22.41中，即使每个路由器从每个接口只发送出分组的一个副本，网络3还是接收到分组的二个副本。有重复的原因是没有生成树，而是用洪。网络3有二个父路由器：R2和R4。为了消除重复情况的发生，对每个网络必须只定义一个父路由器，而且必须施加一种限制，即：一个网络只能通过一个指定的父路由器（designated parent router），接收一个来自特定源端的多播分组。

现在，这种策略就很清楚了。对每个源端，路由器只将分组从指定的父路由器的端口发送出去。这种策略称为逆路径广播（RPB）。RPB不仅能确保分组到达每个网络，而且能保证每个网络只接收一个副本。图22.42说明了RPF与RPB之间的差异。



a) RPF　　b) RPB

图22.42 RPF与RPB的比较

读者可能会问父路由器是如何确定的。用到源端具有最短路径的路由器作为指定的父路由器。因为路由器周期地向每个其他的路由器发送更新分组（在RIP），它们很容易确定在邻近路由器中哪一个路由器到源端有最短路径（在源端作为目的端解释时）。如果存在多个路由器相等时，选取具有最小IP地址的路由器。

> RPB创建从源端到每一个目的端的一个最短路径广播树，它保证每个目的端接收到分组的一个且仅有一个副本。

- 逆路径多播（RPM）。也许读者已经注意到了，RPB并不能对分组进行多播，而只能对它进行广播。这样做是没有效率的。为了使其更有效率，对特定的组，多播分组只能到达那些具有活动成员的网络。该策略称为逆路径多播（reverse path multicasting，RPM）。为了把广播转变成多播，协议使用两种方法：修剪和嫁接。图22.43说明了修剪和嫁接的思想。

每个网络的指定的父路由器负责保存成员关系信息。这通过第21章所讨论的IGMP协议来完成。当连接网络的路由器发现没有参加多播分组时，修剪过程就开始了。路由器开始发送一个**修剪报文**（prune message）到上行路由器，这样它可拒绝相应的接口。即上行路由器可停止通过那个接口发送这个组的多播报文。现在，如果这个路由器接收到来自所有下行路由

器的修剪报文，它依次地发送一个修剪报文到上行路由器。

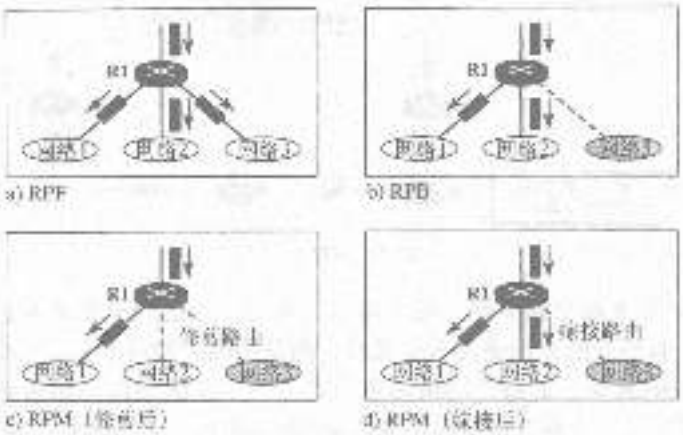


图22.43　RPF、RPB和RPM

如果一个叶子路由器（在树底端的路由器）已发送一个修剪报文，但通过IGMP突然了解到其中一个网络现在希望接收该多播分组。将怎样做？它可发送嫁接报文（graft message），该报文强制上行路由器恢复发送多播报文。

> RPM将修剪和嫁接添加到RPB中，以创建一个多播最短路径树，该树支持动态成员关系的改变。

DVMR距离向量多播路由选择协议（Distance Vector Multicast Routing Protocol）是多播距离向量路由选择的实现。它是基于RIP的基于源树的路由选择协议。

CBT

基于核的树（core-Based Tree，CBT）协议是一种组共享协议，它使用核心路由器作为树的根。自治系统（AS）被分成多个区域，而每个区域选择一个核心路由器（中心路由器或会合路由器）。

树的生成　在选择了一个会合路由器之后，就要将所选择路由器的单播地址通知给每个路由器，然后每个路由器就发送一条单播加入报文来表明它想加入该组。这条报文通过位于发送方和会合路由器之间的所有路由器。每个中间路由器就从这条报文中取出所需的信息，如发送方的单播地址和分组到达所经过的端口等。然后它就将该报文转发到路径中的下一个路由器，当会合路由器已经收到了来自组中每个成员的所有加入报文时，树就生成完毕。此时，每个路由器都知道了它的上行路由器（通向根的路由器）和下行路由器（通向叶子的路由器）。

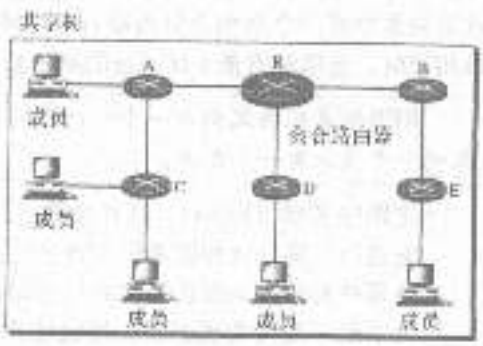


图22.44　具有会合路由器的组共享树

如果一个路由器想离开该组，它就给它的上行路由器发送一条离开报文。上行路由器就从树中除去与那台路由器的链路，并将该报文转发到它的上行路由器，依此类推。图22.44说明了一个具有会合路由器的组共享树。

读者可能已注意到了DVMRP、MOSPF及CBT之间的差异。首先，前面两种树是从根生成的，而CBT的树是从叶子生成的。其次，在DVMRP中，首先生成树（广播），然后对其进行修剪，而在CBT中，开始并没有生成树，树是通过逐步加入（嫁接）节点而形成的。

发送多播分组　在树生成之后，任何（属于组或不属于组的）源端都会给组中所有的成员发送一个多播分组。它仅使用会合路由器的单播地址将该分组发送到这个路由器。然后这个会合路由器将该分组分发到该组的所有成员。图22.45说明了一个主机如何能将一个多播分组发送到该组的所有成员。注意，源主机可以是共享树内（或外）的任何主机。在图22.45中，说明了位于共享树外的源主机。

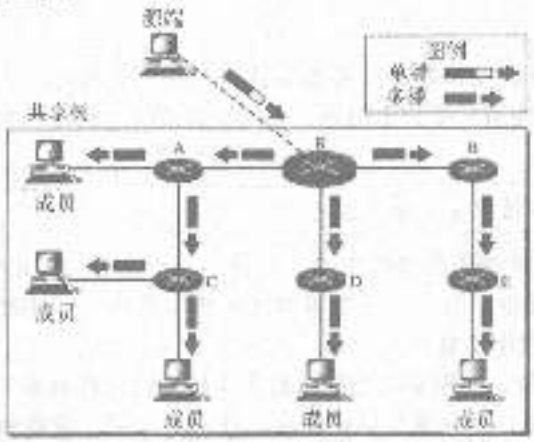


图22.45　发送一个多播分组到会合路由器

选择会合路由器　除了一点之外，这个方法是简单的。应该如何选择会合路由器以优化处理过程和多播呢？已经有了多种实现方法，但这个问题已超出了本书的讨论范围，留给更高级的书讨论。

总之，基于核的树（CBT）是一个组共享树，一种每组使用一棵树的基于中心的协议。树中的一个路由器称为核心路由器。遵循如下过程将一个分组从源端发送到组的成员：

1. 可能是（也可能不是）树的一部分的源端将多播分组封装到一个具有核心路由器单播目的地址的单播分组中，并将其发送给这个核心路由器。这一阶段的传递是使用一个单播地址来实现的，并且唯一的接收方是核心路由器。

2. 核心路由器将单播分组分解出来，并将其转发到所有加入请求到达的端口。

3. 每个接收到多播分组的路由器接着又将其转发给所有有加入请求的端口。

> 在CBT中，源端给核心路由器发送多播分组（封装在一个单播分组中），核心路由器分解该分组，并将其转发到所有有加入请求的主机。

PIM

协议独立多播（Protocol Independent Multicast，PIM）是两个独立多播路由选择协议的合称，它们是PIM密集模式（PIM Dense Mode，PIM-DM）和PIM稀疏模式（PIM Sparse Mode，PIM-SM）。这两个协议都是依赖于单播协议，在这里就不讨论其相似性了，下面分别对其进行讨论。

PIM-DM　当每个路由器存在参与多播路由选择的可能性时，就使用PIM-DM。在此情况下，使用一种广播分组的协议是合理的，因为几乎所有的路由器都参与到该过程中。

PIM-DM用于密集多播环境，如一个局域网中。

PIM-DM是一种以基于源树的路由选择协议，它使用RPF和修剪/嫁接的多播路由选择策略。其操作与DVMRP相似。然而，与DVMRP不同的是，它不依赖于一种特定的单播路由选择协议，它假定自治系统使用一种单播协议，并且每个路由器有一个表，利用该表就能找到通向目的端的最优路径的外发端口。这种单播协议可能是一种距离矢量协议（RIP）或链路状态协议（OSPF）。

PIM-DM使用RPF和修剪/嫁接策略来处理多播路由选择，然而，它还是独立于底层的单播协议。

PIM-SM　当每个路由器存在参与多播路由选择的可能性很小时（稀疏模式），就使用PIM-SM。在此情况下，使用广播分组的协议是不合理的，但使用一种诸如CBT之类的组共享树的协议就更加合理了。

PIM-SM用于稀疏多播环境，如广域网。

PIM-SM是一种组共享树路由选择协议。它使用一个会合点（RP）作为树的源端。其操作与CBT相似，但它更简单。因为它不要求对加入报文的确认。此外，它还为每个区域创建一个RP备份集，用于RP故障恢复。

PIM-SM的一个特点是，在需要时它能从组共享树策略转换到基于源树策略。如果存在一个远离RP的密集活动区域，就会发生这种情况。在此情况下，就能利用基于源树策略，而不是组共享树策略对该区域进行更高效地处理。

PIM-SM与CBT类似，但其实现过程更简洁。

MBONE

多媒体和实时通信使因特网上对多播路由选择的需求增大了。然而，只有一小部分因特网路由器是多播路由器。换句话说，一个多播路由器（multicast router）可能在其附近找不到另一个多播路由器来转发多播分组。虽然在近几年内可通过增加越来越多的多播路由器解决此问题，但可以采用另一种方法，这种方法就是隧道技术（tunneling），多播路由器可视为紧接着单播路由器的一组路由器。多播路由器在物理上可能没有连接在一起，但它们在逻辑上是连在一起的，图22.46说明了这种思想。在图22.46中，只有阴影圆圈内的路由器能进行多播路由选择，如果没有隧道技术，这些路由器是一些孤立的"岛屿"。为了使它们能进行多播路由选择，可运用隧道技术的概念将这些孤立的路由器组成一个多播主干网（multicast backbone，MBONE）。

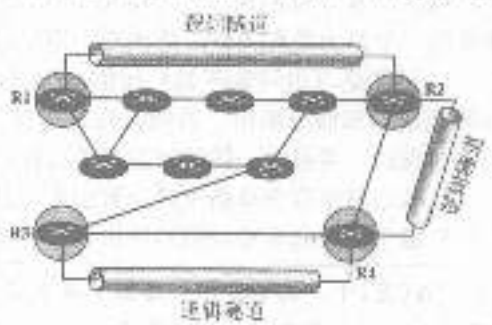


图22.46　逻辑隧道

通过将多播分组封装于一个单播分组中可以创建一个逻辑隧道（logical tunnel），此时，多播分组就成为单播分组的有效载荷（数据）。中间（非多播）路由器作为单播路由器来路由分组，并将其从一个"岛屿"传递到另一个。这样做好像单播路由器不存在，而两个多播路由器是相邻的一样。目前唯一支持MBONE和隧道技术的协议是DVMRP。图22.47说明了这种概念。

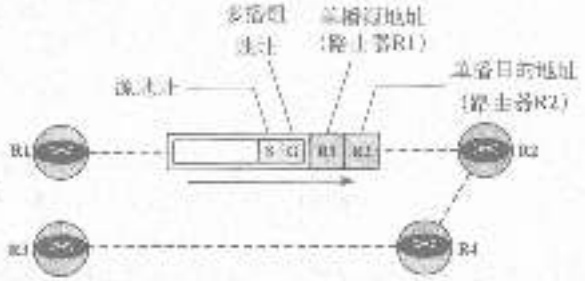


图22.47 MBONE

推荐读物

作为深入讨论与本章有关的主题，推荐阅读下列读物和网站，在本书末列出方括号[…]中的参考资料。

读物

传递和转发在[For06]的第6章中有讨论。单播路由选择协议可参看[For06]的第14章，而多播与多播路由选择可参看同书的第15章。[WZ01]有完整多播讨论。路由选择协议可参见[Hui00]。OSPF在[Moy98]中有讨论。

网站

• www.ietf.org/rfc.html 有关请求评论。

请求评论

RIP的讨论可从下列请求评论中找到：

1131，1245，1246，1247，1370，1583，1584，1585，1586，1587，1722，1723，2082，2453

OSPF的讨论可从下列请求评论中找到：

1131，1245，1246，1247，1370，1583，1584，1585，1586，1587，2178，2328，2329，2370

BGP的讨论可从下列请求评论中找到：

1092，1105，1163，1265，1266，1267，1364，1392，1403，1565，1654，1655，1665，1771，1772，1745，1774，2283

本章小结

• 如果一个分组的发送方（主机或路由器）与目的端都连接在同一网络上，则分组的传递就称为直接传递；如果分组的发送方（主机或路由器）和目的端连接在不同的网络上，则分组的传递就称为间接传递。
• 在下一跳方法中，不需要将分组所经过的每个路由器全部列在路由表中，而只需要将下一跳地址写在路由表中。在特定网络方法中，在一个网络上的所有主机都共享路由表中的一个项目。
• 在特定主机方法中，路由表中包含了一个主机的完整地址。
• 在默认方法中，指定一个路由器接收所有在路由表中无匹配的分组。
• 对于无类寻址，路由表至少需要4列。
• 在无类寻址中，地址聚合简化了转发过程。
• 在无类寻址中，要求最长的掩码匹配。

- 为了防止极大的路由表，无类寻址需要分层结构的和地域的路由选择。
- 静态路由表中的项目是由网络管理人员手工更新，而动态路由表中的项目是由路由选择协议自动更新的。
- 度量是指通过网络传输分组的代价。
- 自治系统（AS）是在一个单独的管理机构管辖之下的一组网络和路由器。
- RIP是距离向量路由选择协议，每个路由器与它的邻站在固定时间间隔内共享整个关于AS的知识。
- RIP协议的两个缺点是慢收敛和不稳定性。解决RIP不稳定性的方法有触发更新、分割范围和毒性逆转。
- OSPF将自治系统划分为一些区域，这些区域定义为一些网络、主机和路由器的集合。
- OSPF是链路状态路由选择协议，每一个路由器向区域内的每个其他路由器发送其邻站的状态，只有当邻站有变化时才发送分组。
- OSPF路由表是使用Dijkstra算法计算出来的。
- BGP是用在自治系统之间的路由选择协议，用于更新路由表。
- BGP是基于称为路径向量路由选择一种路由选择协议。在这种协议中，经过自治系统的分组必须显式地列出。
- 在多播路由选择的基于源树方法中，源端/组对决定了树的结构；在多播路由选择的组共享树方法中，组决定了树的结构。
- MOSPF是一种多播路由选择协议，它使用多播链路状态路由选择来创建一个基于源树方法的最小代价树。
- 在逆路径转发（RPF）中，路由器只转发那些从源端沿着最短路径到达它的分组。
- 逆路径广播（RPB）创建一个从源端到每个目的端的最短路径广播树。它能保证每个目的端只接收到一个而只有一个分组副本。
- DVMRP是一种多播路由选择协议，它用距离向量路由选择协议创建一个基于源的树。
- 基于核树（CBT）协议是一种利用一个路由器作为树根的多播路由选择协议。
- PIM-DM是一种基于源树的路由选择协议，它使用RPF和修剪以及嫁接策略来处理多播路由选择。
- PIM-SM是一种与CBT类似的组共享树路由选择协议，它使用会合路由器作为树的根。
- 对于两个不邻的多播路由器之间的多播，我们建立一个多播主干网（MBONE）以实现隧道技术。

习题

复习题

1. 直接传递与间接传递有什么不同？
2. 列出本章所讨论的二种转发技术，对每种技术给出简要说明。
3. 比较本章讨论的两种不同的路由表。
4. RIP的目的是什么？
5. RIP报文的功能是什么？
6. 为什么截止计时器的值是周期计时器的值6倍？
7. 跳数限制如何缓解RIP问题？
8. 列举RIP的缺点及其相应的补救办法。

9. OSPF定义的4种类型链路的分类基础是什么？
10. 为什么OSPF报文比RIP报文传播速度更快？
11. BGP的目的是什么？
12. 试给出本章所讨论的两组多播路由选择协议的简要说明。

练习题

13. 试表示一个完全被隔离的主机的路由表。
14. 试表示一个连接局域网但未连接互联网的主机的路由表。
15. 如果路由器R1的路由表是如下表22.3，试找出网络的拓扑结构。

表22.3 练习15的路由表

掩 码	网络地址	下一跳地址	接 口
/27	202.14.17.224	—	m1
/18	145.23.192.0	—	m0
默认	默认	130.56.12.4	m2

16. 在图22.8中的路由器R1能否接收到目的地址为140.24.7.194的分组？并说明其理由。
17. 在图22.8中的路由器R1能否接收到目的地址为140.24.7.42的分组？并说明其理由。
18. 试表示图22.9中区域ISP的路由表。
19. 试表示图22.9中本地ISP1的路由表。
20. 试表示图22.9中本地ISP2的路由表。
21. 试表示图22.9中本地ISP3的路由表。
22. 试表示图22.9中本地ISP3的路由表。
23. 试比较距离向量路由选择和链路状态路由选择。
24. 一个路由器具有如下的路由表

    Net1 4 B
    Net2 2 C
    Net3 1 F
    Net4 5 G

    如果该路由器从路由器C接收到下列的RIP报文，那么路由表中的内容将会是怎样？

    Net1 2
    Net2 1
    Net3 3
    Net4 7

25. 通知N个网络的RIP报文中有多少个字节是空的？
26. 一个路由器具有如下的RIP路由表

    Net1 4 B
    Net2 2 C
    Net3 1 F
    Net4 5 G

    试给出这个路由器发送的响应报文。
27. 试表示具有下列说明的自治系统：

    a. 具有8个网络（N1到N8）；　　b. 具有8个路由器（R1到R8）；

c. N1、N2、N3、N4、N5和N6是以太网；　d. N7和N8是点到点的广域网；
e. R1连接到N1和N2；　f. R2连接到N1和N7；
g. R3连接到N2和N8；　h. R4连接到N7和N8；
i. R5连接到N6和N3；　j. R6连接到N6和N4；
k. R7连接到N6和N5；　l. R8连接到N8和N5。

28. 试给出由OSPF看到的练习27中的自治系统的图形表示。
29. 练习27中的哪一个是过渡网络？哪一个是残桩网络？
30. 一个路由器使用DVMRP从接口2接收到源地址为10.14.17.2的一个分组。如果路由器转发该分组，在单播路由表中有关这个地址项目的内容是什么？
31. RPF实际上创建最短路径树吗？试说明之。
32. RPB实际上创建最短路径树吗？试说明之。树的叶子是什么？
33. RPM实际上创建最短路径树吗？试说明之。树的叶子是什么？

探究题

34. 如果你已访问过UNIX（或LINUX），试用*netsat*和*ifconfig*程序寻找你所连接的服务器的路由表。
35. 查明你的ISP如何使用地址聚合和最长掩码匹配原则。
36. 查明你的IP地址是否也是域地址分配的部分。
37. 如果你使用路由器，求出路由表中列个数与列名。

第五部分 传 输 层

目标

传输层负责整个报文的进程到进程的传递。进程就是运行在主机上的应用程序。虽然网络层监管独立分组从源端到目的端的传递，但它不辨认这些分组之间的关系。它独立地对待每个分组，无论什么情况，好像每块都属于一个单独的报文。另一方面，传输层在从源端到目的端的层次中，监管差错控制和流量控制以确保全部报文完整地、按序地到达。

传输层负责报文从一个进程到另一个进程的传递。

计算机时常在同一个时间运行多个程序。因此，从源端到目的端的传递不仅是一个计算机到下一个计算机的传递，而且还是计算机上的一个特定进程到另一计算机上的一个特定进程的传递。所以，在OSI模型中，传输层头部必须包含一个称为服务点地址的地址类型，或在因特网和TCP/IP协议族中的端口号或端口地址。

传输层协议可以是无连接的，也可以是面向连接的。无连接传输层将每段作为一个独立的分组处理，将它传递到目的机器的传输层。面向连接传输层在传递分组前要与目的机器的传输层建立一个连接。当所有数据传输完成后，要终止这个连接。

在传输层，一个报文通常被划分成可传输的段。一个无连接协议（如UDP）对每一段独立处理。一个面向连接协议（如TCP和STCP）还要用序号生成这些段之间的关系。

像数据链路层一样，传输层也负责流量控制和差错控制。但是，在这一层的流量控制和差错控制是端到端的，而不是在单一一条链路上。我们将在本书的这部分讨论这些协议。UDP不包含流量控制和差错控制。另一方面，其他两个协议，TCP和SCTP使用滑动窗口进行流量控制，使用确认系统进行差错控制。

本书的第五部分专门讨论传输层以及该层提供的服务。

第23章 UDP、TCP和SCTP

本章开始给出传输层（transport layer）存在的基本理由，即进程到进程传递的必要性。主要讨论这类传递产生的问题，并讨论解决这些问题的方法。

在因特网模型中，传输层有三种协议：UDP、TCP和SCTP。首先讨论UDP协议，它是这三种中比较简单的一种协议。我们将看到如何使用缺乏另外两种协议的某些特性的这种简单的传输层协议。

然后讨论TCP协议，它是一种复杂的传输层协议。我们将了解到如何将以前提到的概念应用于TCP中。并将有关TCP协议中拥塞控制和服务质量的讨论放在第24章。因为这两个问题同样也适用于数据链路层和网络层。

最后，我们讨论新的传输层协议SCTP，它是为像多媒体这样的多接口与多流应用而设计的。

23.1 进程到进程的传递

数据链路层负责链路上的两个相邻节点之间的帧传递，这称为节点到节点的传递。网络层负责两台主机之间的数据报传递，这称为主机到主机的传递。因特网中的通信并不是定义为两个节点或两个主机之间的数据交换。实际的通信是发生在两个进程（应用程序）之间的。我们需要进程到进程的传递（process-to-process delivery）。但是，在任何时刻，在源主机上可能运行着多个进程，并且在目的主机上也运行着多个进程。为了完成传递过程，需要一种机制将源主机上运行的某个进程的数据发送到目的主机上运行的对应进程上。

传输层负责进程到进程的传递，即进程之间的分组传递以及部分消息传递。后面将会看到两个进程以客户/服务器的方式通信。图23.1表示了二种传送类型以及它们的应用范围。

传输层负责进程到进程的传递。



图23.1 数据传送类型

23.1.1 客户/服务器模式

虽然有多种方法可以实现进程到进程的通信，但最常用的一个方法是通过客户/服务器模式（client-server paradigm）。本地主机上的进程称为客户（client），它通常需要来自远程主机上的进程提供的服务。这个远程主机称为服务器（server）。

这两个进程（客户和服务器）有相同的名字。例如，如果要从远程机器上获得日期和时间，我们需要在本地主机上运行Daytime客户进程和在远程机器上运行Daytime服务器进程。

目前的操作系统支持多用户和多程序运行的环境。一个远程计算机在同一时间可以运行多个服务器程序，就像许多本地计算机可在同一时间运行一个或多个客户应用程序一样。对通信来说，我们必须确定以：

1. 本地主机
2. 本地进程
3. 远程主机
4. 远程进程

寻址

每当需要传送信息到多个目的地之中的某一特定目的地时，就需要一个地址。在数据链路层，如果连接不是点到点的，则需要一个MAC地址从多个节点中选择一个节点。数据链路层的帧需要一个目的端MAC地址用于传送数据和一个源地址用于下一节点的回答。

在网络层，需要一个IP地址来选择数百万主机之中的一个主机。网络层中的数据报需要目的IP地址用于传送数据，需要源IP地址用于接收目的主机的回答。

在传输层，需要一个称为**端口号**（port number）的传输层地址，利用这一地址从目的主机上运行的多个进程中选择相应的进程。目的端口号用于传送，而源端口号用于接收回答。

在因特网模型中，端口号是在0～65 535之间的16位整数。客户程序用端口号定义它自己。这个端口号由运行在客户主机上的传输层软件随机选择的，这是**临时端口号**（ephemeral port number）。

服务器进程也必须用一个端口号定义它自己，但这个端口号不能随机选择。如果服务器站点的计算机运行一个服务器进程，并随机分配一个数字作为端口号，那么在客户站点的进程想访问该服务器并使用其服务时，将不知道此端口号。当然，一个解决方法是发送一个特殊分组，并请求一个特定服务器的端口号，但是这需要更多的开销。因特网决定给服务器使用全局端口号，它们称为**熟知端口号**（well-known port number）。但这条规则也有例外，例如，有一些客户端也被分配了熟知端口号。每一个客户进程都知道相应服务器进程的熟知端口号。例如，前面所讨论的Daytime客户进程可以使用临时（暂时）端口号52 000来表示自己，但服务器Daytime进程就必须使用熟知（永久的）端口号13。图23.2表示了这一概念。

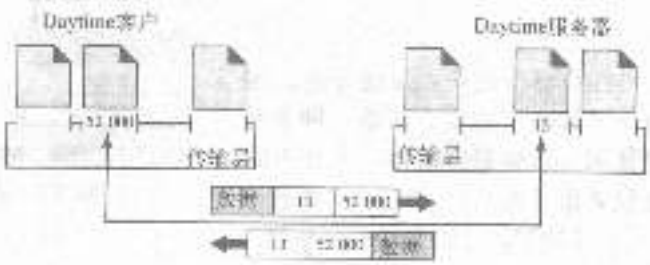


图23.2 端口号

现在必须搞清楚，在选择数据的最终目的端时，IP地址和端口号起着不同的作用。目的IP地址定义了在世界范围的不同主机中确定一个主机。但主机被选定后，端口号定义了在该特定主机上的多个进程中的一个进程（见图23.3）。

IANA范围

因特网号码分配管理局（Internet Assigned Number Authority，IANA）已经把端口编号划分为三种范围：熟知的、注册的和动态的（或私有的），如图23.4所示。

- 熟知端口。端口号的范围是0～1 023，由IANA分配和控制。这些是熟知端口。
- 注册端口。端口号的范围是1 024～49 151。IANA不分配或也不控制。它们可在IANA注册以防重复。
- 动态端口。端口号的范围是49 152～65 535。这一范围内的端口号既不受控制也不需要注册，可以由任何进程使用。它们是临时端口。

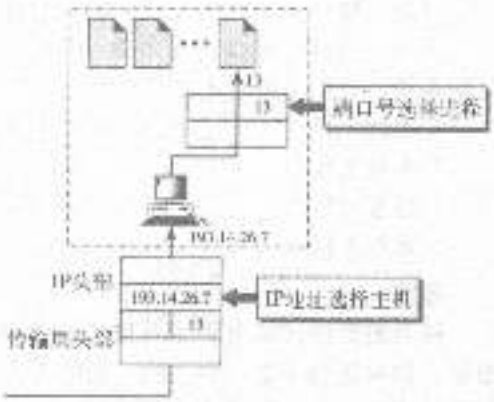


图23.3　IP地址与端口号

套接字地址

进程到进程的传递需要有两个标识符，IP地址和端口号，它们各用在一端以建立一条连接。一个IP地址和一个端口号结合起来称为套接字地址（socket address）。客户套接字地址唯一定义了客户机进程，而服务器套接字地址唯一地定义了服务器进程（见图23.5所示）。

传输层协议需要一对套接字地址：客户套接字地址和服务器套接字地址。这四条信息是IP头部和传输层协议头部的组成部分。IP头部包含IP地址，而UDP或TCP头部包含端口号。

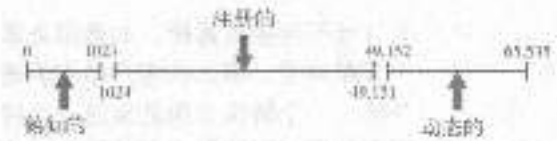


图23.4　IANA定义的范围

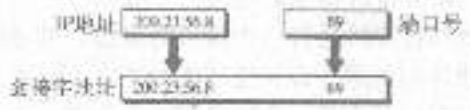


图23.5　套接字地址

23.1.2　复用与分离

寻址机制允许通过传输层进行复用和分离，如图23.6所示。

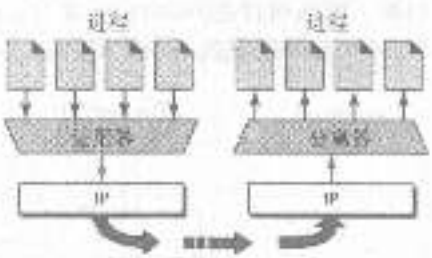


图23.6　复用与多路分离

复用

在发送方站点，可能有多个进程要发送分组。但是，在任何时候只有一个传输层协议。这是一种多对一的关系，因而需要复用。传输层协议接收来自不同进程的报文，这些进程是由分配给它们的端口号进行区分。添加了头部以后，传输层把分组发送给网络层。

分离

在接收方站点，也只有一个传输层协议，但有多个进程可能要接收分组。这是一对多的关系，因而需要分用。传输层接收来自网络层的数据报，经过纠错和去除头部以后，传输层根据端口号将每个报文传递到适当的进程。

23.1.3　无连接服务与面向连接的服务

传输层协议可以是无连接的或面向连接的服务。

无连接服务

在无连接服务（connectionless service）中，分组从一方发送给另一方，不需要建立连接和释放连接。分组没有编号，它们可能被延迟、丢失或无序到达，也没有确认过程。下面我们会看到，在因特网模型中，其中的一个传输层协议UDP就是无连接的。

面向连接的服务

在面向连接的服务（connection-oriented service）中，首先在发送方和接收方之间建立一个连接，然后传送数据，最后释放连接。我们将看到TCP和SCTP都是一种面向连接的协议。

23.1.4 可靠服务与不可靠服务

传输层服务可以是可靠的或不可靠的。如果应用层程序需要可靠性，我们使用可靠的传输层协议，通过在传输层实现流量控制和差错控制来获得这种可靠性。这意味着一种较慢和更复杂的服务。另一方面，如果应用程序不需要可靠性，因为它使用自己的流量和差错控制机制或者它需要快速服务或者服务的本质特性不要求流量和差错控制（如实时应用），那么就可以使用不可靠服务。

在因特网中，有三种不同的传输层协议，前面已经提到过。UDP是无连接的和不可靠的，而TCP和SCTP则是面向连接的和可靠的。这三种协议都能与应用层程序的需要相对应。

肯会想到一个疑问：如果数据链路层是可靠的，并且已经有了流量控制和差错控制，那么还需要在传输层进行流量控制和差错控制吗？回答是需要的。因为在数据链路层的可靠性存在于两个节点之间，而我们需要端到端的可靠性，并且因为因特网的网络层是不可靠的（尽力传递），所以必须在传输层实现可靠性。要知道数据链路层的差错控制并不能保证传输层的差错控制，让我们看一下图23.7。

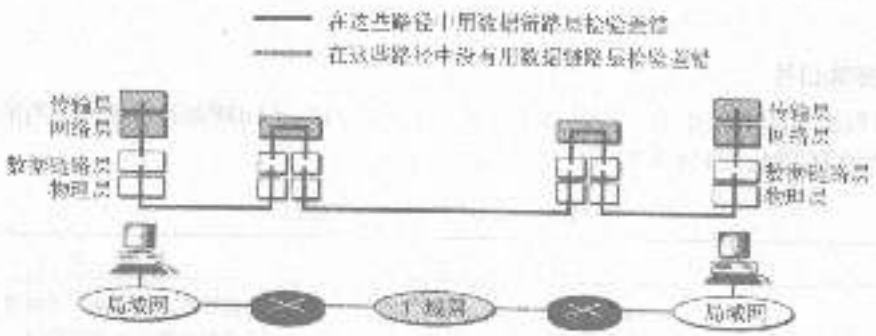


图23.7 差错控制

下面将看到TCP中的流量和差错控制是通过滑动窗口协议实现的，就像我们在第11章中已经讨论过的那样。但是，窗口是面向字符而不是面向帧的。

23.1.5 三种协议

原始的TCP/IP协议族在传输层只规定了两个协议：UDP和TCP。在讨论TCP以前，我们首先关注这两个协议中较简单的UDP。然后在本章中也讨论新设计的传输层协议SCTP。图23.8表示该协议在TCP/IP协议族中的位置。

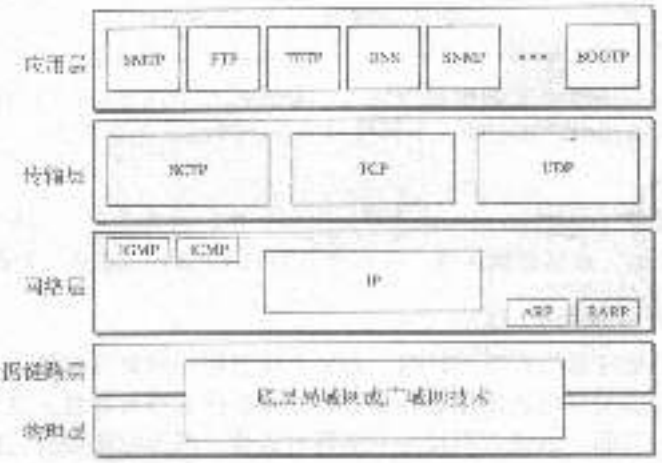


图23.8　在TCP/IP协议族中UDP、TCP和SCTP的位置

23.2　用户数据报协议（UDP）

用户数据报协议（User Datagram Protocol，UDP）称为**无连接不可靠传输层协议**（connectionless，unreliable transport protocol），它除了提供进程到进程通信而不是主机到主机通信外，就没有给IP服务增加任何东西。此外，它还完成非常有限的差错检验，如果UDP功能是如此差，那么为什么进程还要使用它？它有缺点但也有优点。UDP是一个非常简单的协议，开销最小。如果一个进程想发送很短的报文，而且不在意可靠性，就可以使用UDP。使用UDP发送一个很短的报文，在发送方和接收方之间的交互要比使用TCP或SCTP时少得多。

23.2.1　熟知端口号

表23.1列出了UDP使用的一些熟知端口号。有些端口号是UDP和TCP都可使用的。在本章稍后我们讨论TCP时，将讨论它们。

表23.1　UDP使用的熟知端口

端　口	协　议	说　明
7	Echo	将接收到的数据报回送到发送方
9	Discard	丢弃接收到的任何数据报
11	Users	活跃的用户
13	Daytime	返回日期和时间
17	Quote	返回日期名引用
19	Chargen	返回一个字符串
53	Nameserver	域名服务
67	BOOTPs	下载引导程序信息的服务器端口
68	BOOTPc	下载引导程序信息的客户端端口
69	TFTP	简单文件传送协议
111	RPC	远程过程调用
123	NTP	网络时间协议
161	SNMP	简单网络管理协议
162	SNMP	简单网络管理协议（中断）

例23.1

在UNIX中，熟知端口号存储在/etc/services文件中。这个文件的每行给出服务器名和熟知端口号。我们可用*grep*命令提取该行所对应的应用。下面表示了FTP的端口。注意：FTP可使用UDP或TCP的端口21。

```
$ grep  ftp  /etc/services
ftp         21/tcp
ftp         21/udp
```

SNMP使用两个端口号（161和162），我们将在第28章中看到每一个用于不同目的。

```
$ grep  snmp /etc/services
snmp        161/tcp     #Simple Net Mgmt Proto
snmp        161/udp     #Simple Net Mgmt Proto
snmptrap    162/udp     #Traps for SNMP
```

23.2.2 用户数据报

UDP分组称为**用户数据报**（user datagram），有8字节的固定头部。图23.9说明了用户数据报的格式。



图23.9 用户数据报格式

用户数据报共有下列字段：

- 源端口号。这是在源主机上运行的进程使用的端口号。它有16位长，这就表示端口号可以从0到65 535。如果源主机是客户端（当客户进程发送请求时），那么大多数情况下这个端口号就是暂时端口号，它由该进程请求，由源主机上运行的UDP软件进行选择。如果源主机是服务器（当服务器进程发送响应时），那么在大多数情况下这个端口号是熟知端口号。
- 目的端口号。这是在目的主机上运行的进程使用的端口号。它有16位长。如果目的主机是服务器（当客户进程发送请求时），那么在大多数情况下这个端口号是熟知端口号。如果目的主机是客户端（当服务器进程发送响应时），那么大多数情况下这个端口号是暂时端口号。在这种情况下，服务器将它接收到的请求分组中的暂时端口号复制下来。
- 长度。这是一个16位字段，它定义了用户数据报的总长度，头部加上数据。16位可定义的总长度是从0到65 535字节。但是，总长度必须比这个小，因为UDP数据报存放在具有总长度为65 535字节的IP数据报中。

UDP用户数据报的长度字段实际上是没有必要的。用户数据报封装成IP数据报。在IP数据报中有一个字段定义总长度，IP数据报还有一个字段定义头部长度。所以如果我们将上述第一个字段的值减去第二个字段的值，我们就能求得封装在IP数据报中的UDP数据报的长度。

UDP长度=IP长度−IP头部长度。

但是，UDP协议设计者认为，要让IP软件提供这一信息更有效。我们应该当记得，当IP软件把UDP用户数据报提交给UDP层时，它已经除去了IP头部。

- 校验和。这个字段用来校验整个用户数据报（头部加数据）出现的差错。校验和将在下一节讨论。

23.2.3 校验和

我们已在第10章讨论过校验和的概念与计算方法。我们还讨论过如何计算IP和ICMP分组

的校验和。现在我们讨论如何计算UDP的校验和。

UDP校验和的计算与IP和ICMP校验和的计算不同。这里校验包括三个部分：伪头部，UDP头部和从应用层来的数据。

伪头部（pseudoheader）是IP分组的头部的一部分，其中有些字段要填入0，用户数据报封装在IP分组中（见图23.10）。



图23.10　用于校验和计算的伪头部

如果校验和不包括伪头部，用户数据报也可能是安全的和正确地到达。但是，如果IP头部受到损坏，那么它可能被提交到错误的主机。

增加协议字段可确保这个分组是属于UDP，而不是属于其他传输层协议。我们在后面将会看到，如果一个进程既可用UDP又可用TCP，则端口号可以是相同的。UDP的协议字段值是17。如果在传输过程中这个值改变了，在接收端计算校验和时就可检测出来，UDP就可丢弃这个分组，这样就不会传递给错误的协议。

注意伪头部字段和IP头部最后12字节的相似性。

例23.2

图23.11给出了只有7个字节数据的很小的用户数据报的校验和计算。因为数据的字节数是奇数，因此为了计算校验和就要加上填充。当用户数据报要传递给IP时，就将伪头部和填充丢弃。

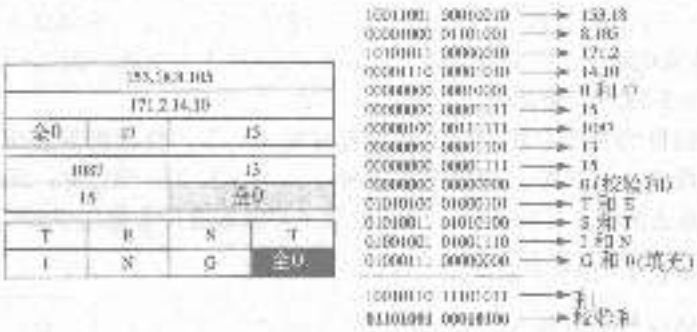


图23.11　简单UDP用户数据报的校验和计算

可选的校验和

校验和的计算以及在用户数据报中包括的校验和都是可选的。如果不进行校验和的计算，则这个字段就填入全1。注意：计算得到的校验和永远不会是全1，因为这就是说和是全0，这是不可能的。由于它要求字段的值为全0，这是不可能的。

23.2.4 UDP的操作

UDP使用了传输层共同的一些概念。将简要地讨论这些概念，然后在下一节介绍TCP协议时展开讨论。

无连接服务

如前所述，UDP提供无连接服务。这就是表示UDP发送出去的每一个用户数据报都是一个独立的数据报。不同的用户数据报之间没有关系，即使它们都来自相同的源进程并发送到相同的目的程序。用户数据报不进行编号。此外，也没有像TCP协议那样的连接建立和连续终止，这就表示每一个用户数据报可以沿着不同的路径传递。

无连接的一个结果就是使用UDP的进程不能向UDP发送数据流，也不能期望UDP将这个数据流分割成许多不同的相关联的用户数据报。相反，每一个请求必须足够小，使其能够装入到用户数据报中，只有那些发送短报文的进程才应当使用UDP。

流量控制和差错控制

UDP是一个简单的不可靠的传输协议，它没有流量控制，因而也没有窗口机制。如果到来的报文太多时，接收方可能会溢出。

除校验和外，UDP也没有差错控制机制。这就表示发送方不知道报文是否丢失还是重复传递。当接收方使用校验和检测出差错时，就悄悄地将此用户数据报丢弃。

缺少流量控制（flow control）和差错控制（error control）就表示使用UDP的进程必须要提供这些机制。

封装和拆封

要将报文从一个进程发送到另一个进程时，UDP协议就要将报文在IP数据报中进行封装和拆封。

排队

我们已经提到过端口，但没讨论端口的实际实现。在UDP中，队列是与端口联系在一起的（见图23.12所示）。

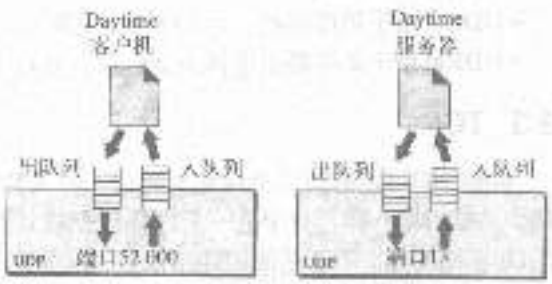


图23.12 UDP中的队列

在客户机端，当进程启动时，它从操作系统请求一个端口号。有些实现是创建一个入队列和一个出队列与每一个进程相关联，而有些实现只创建与每一个进程相关的入队列。

注意：即使当一个进程想与多个进程通信时，它也只得到一个端口号，而最后也只有一个出队列和一个入队列（queue）。在多数情况下，由客户打开的队列由暂时端口号来标识，只要进程在运行，这些队列就起作用。当进程终止时，队列就撤销。

客户进程使用在请求中指明的源端口号将报文发送到出队列。UDP逐个将报文取出，加上UDP头部递交给IP。出队列可能发生溢出。如果发生溢出，操作系统就要求客户进程在继续发送报文之前要等待。

当报文到达客户端时，UDP要检查一下以确认对应于该用户数据报中目的端口号字段指明的端口号是否创建了入队列，如果有这样的入队列，UDP就将接收到的用户数据报放在该队列的末尾。如果没有这样的队列，UDP就丢弃该用户数据报，并请求ICMP协议向服务器发送端口不可达报文。所有发送给一个特定客户程序的入报文，不管是来自相同的服务器还是

不同的服务器，都被放在同一个队列中。入队列可能会溢出，如果发生溢出，UDP就丢弃此用户数据报，并请求向服务器发送端口不可达报文。

在服务器端，创建队列的机制是不同的，用最简单形式，服务器在它开始运行时就用它的熟知端口请求入队列和出队列。只要服务器运行，队列就一直是打开的。

当报文到达服务器进程时，UDP要检查一下以确认对应于该用户数据报中目的端口号字段指明的端口号是否创建了入队列。如果有这样的队列，UDP就将接收到的用户数据报放在该队列的末尾。如果没有这样的队列，UDP就丢弃该用户数据报，并请求ICMP协议向客户端发送一个端口不可达报文。一个特定服务器程序的所有入报文，不管是来自相同的客户还是不同的客户机，都放入同一队列。入队列可能溢出。如果发生溢出，UDP就丢弃这个用户数据报，并请求向客户发送端口不可达报文。

当服务器想要回答客户时，它就使用在请求报文中指明的源端口号将报文发送到出队列。UDP逐个将报文取出，加上UDP的头部，递交给IP。出队列可能溢出，如果发生溢出，操作系统就要求服务器进程在继续发送报文之前要等待。

23.2.5 UDP的使用

下列是UDP协议的某些应用：

- UDP适用于这样的进程，它需要简单要的请求–响应通信，而较少考虑流量控制和差错控制。对于需要传送成块数据的进程，如FTP（见第26章），则通常不使用它；
- UDP适用于具有有内部流量控制和差错控制机制的进程。例如，简单文件传输协议（TFTP）的进程就包括流量控制和差错控制，它能够很容易地使用UDP；
- 对多播来说，UDP是一个合适的传输协议。多播能力已被嵌入在UDP软件中，但没有嵌入在TCP软件中；
- UDP可用于管理进程，如SNMP（见第28章）；
- UDP可用于某些路由选择更新协议，如路由选择信息协议（RIP）（见第22章）。

23.3 TCP

本章我们讨论的第二个传输层协议，称为传输控制协议（Transmission Control Protocol，TCP）。与UDP一样，TCP是一个进程到进程（程序到程序）的协议，所以它使用端口号；但与UDP不同的是，TCP是面向连接协议。为发送数据它在两个TCP之间建立一个虚拟连接。另外，TCP在传输层使用流量控制和差错控制机制。

简单地说，TCP被称为面向连接的、可靠的传输协议。它为IP服务增加了面向连接和可靠性的特性。

23.3.1 TCP服务

在详细讨论TCP之前，让我们解释TCP向应用层进程提供的服务。

进程到进程的通信

像UDP一样，TCP用端口号提供进程到进程的通信。表23.2列出了TCP使用的一些熟知端口号。

流传递服务

与UDP不同，TCP是一个面向流的协议。在UDP中，进程（应用程序）发送一些具有预先规定界限的报文给UDP进行传递，UDP将它自己的头部添加到这些报文中每一个报文上。来自进程的每一个报文称为一个用户数据报，最后变成一个IP数据报。IP和UDP都不认识这

些数据报之间的关系。

表23.2 TCP使用的熟知端口

端 口	协 议	说 明
7	Echo	将接收到的数据报返回发送方
9	Discard	丢弃任何接收到的数据报
11	Users	活跃的用户
13	Daytime	返回日期和时间
17	Quote	返回日期的引用
19	Chargen	返回一字符串
20	FTP，数据	文件传输协议（数据连接）
21	FTP，控制	文件传输协议（控制连接）
23	TELNET	终端网络
25	SMTP	简单邮件传输协议
53	DNS	域名服务器
67	BOOTP	引导程序协议
79	Finger	Finger
80	HTTP	超文本传输协议
111	RPC	远程过程调用

另一方面，TCP允许发送进程以字节流形式传递数据，并且接收进程也以字节流形式接收数据。TCP建立一种环境，在这种环境中，两个进程好像由一个假想的“管道”连接，这个管道通过因特网传送这些进程的数据。这种假想的环境，如图23.13所示。发送进程产生（写入）字节流，而接收进程消费（读出）这些字节流。



图23.13 字节流传递

发送和接收缓冲区 因为发送和接收进程可能以不同的速度写入和读出数据，所以TCP需要用于存储的缓冲区。每一个方向都存在一个缓冲区：发送缓冲区和接收缓冲区（稍后我们会看到这些缓冲区也用于TCP流量和差错控制机制）。实现缓冲的一种方法是使用一字节存储单元的循环数组，如图23.14所示。为了简化，我们只画出了两个缓冲区，每个缓冲区20个

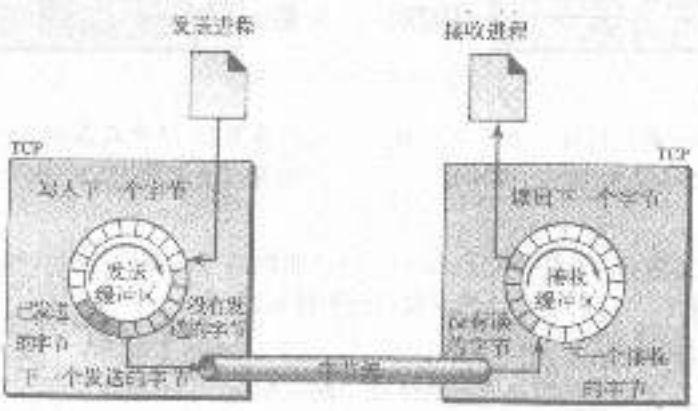


图23.14 发送与接收缓冲区

字节。通常情况下，缓冲区是成百或上千个字节，这取决于实现方法。这里表示的缓冲区大小是相同的，实际上并非总是如此。

图23.14中表示了在一个方向上数据的移动。在发送端，缓冲区有三种类型的存储单元。白色的部分是空存储单元，可以由发送进程（生产者）填充。灰色的部分用于保存已经发送但还没有得到确认的字节。TCP在缓冲区中保留这些字节，直到收到确认为止。灰色缓冲区是将要由TCP发送的字节。但是，在本章的后面将会看到，TCP可能只发送灰色部分。这可能是由于接收进程缓慢或者网络中可能发生的拥塞而造成的。还要注意，灰色存储单元的字节被确认后，这些存储单元可以回收并且对发送进程可用。这就是我们表示一个环形缓冲区的原因。

接收端的缓冲区操作比较简单。环形缓冲区分成两个区域（表示为白色和灰色）。白色区域包含空存储单元，可以由从网络上接收的字节进行填充。灰色区域表示接收到的字节，可以由接收进程读出。当某个字节被接收进程读出以后，这个存储单元可被回收，并加入到空存储单元池中。

段　尽管缓冲能够处理生产进程速度和消费进程速度之间的不相称问题，但在发送数据之前，还需要多个步骤。IP层作为TCP服务的提供者，需要以分组的方式而不是字节流的方式发送数据。在传输层，TCP将多个字节分组合在一起成为一个分组，这个分组称为段（segment）。TCP给每个段添加头部（为了控制的目的），并将该段传递给IP层。段被封装到IP数据报中，然后再进行传输。整个操作对接收进程是透明的。稍后，我们会看到这些段可能被无序接收、丢失、或者损坏和重发。所有这些均由TCP处理，接收进程不会觉察到任何操作。图23.15表示了在缓冲区中如何从字节生成段。

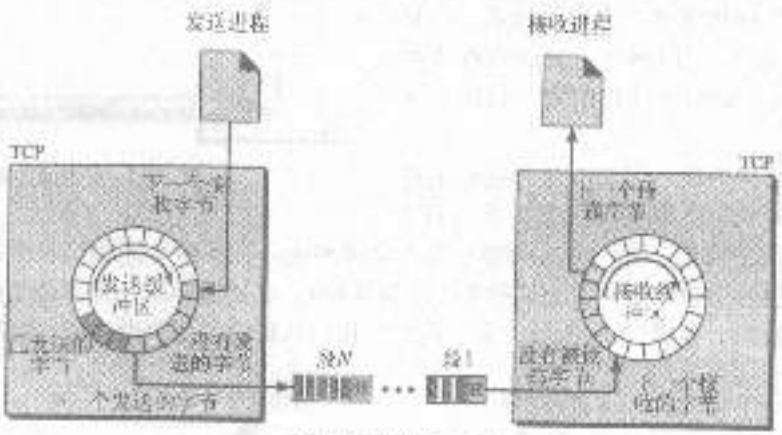


图23.15　TCP中的段

注意：段的大小不必相同。为了简单起见，我们在图23.15中只表示了一个包含3个字节的段和另一个包含5个字节的段。实际的段可能包含数百（或者数千）个字节。

全双工通信

TCP提供全双工服务（full-duplex service），即数据可以在同一时间双向流动。每一方TCP都有发送和接收缓冲区，它们能在双向发送和接收段。

面向连接的服务

与UDP不同，TCP是一种面向连接的协议。站点A的一个进程要发送和接收来自站点B的另外一个进程的数据时，步骤如下：

1. 在两个TCP之间建立一个连接；
2. 在两个方向交换数据；
3. 连接终止。

注意：这是一个虚连接，而不是一个物理连接。TCP段封装成IP数据报，并且可能被无序地发送、或丢失或被破坏，然后重发。每个段都可以通过不同的路径到达目的端。虽然不存在物理连接，但TCP会建立一种面向字节流的环境，在这种环境中，TCP能承担按顺序传递这些字节到其他站点的任务。这就好像建立了横跨多个岛屿的一座桥，以单一的连接从一个岛屿到另一个岛屿传送所有字节。

可靠的服务

TCP是一种可靠的传输协议，它使用确认机制来检查数据是否安全和完整地到达。我们将在本章的后面进一步讨论差错控制的特点。

23.3.2 TCP特点

前节已提到TCP提供服务本节将简要概述TCP的其他一些特点，稍后将进行详细的讨论。

序号系统

虽然TCP软件能够记录发送或接收的段，但是在段的头部没有段序号字段。TCP在段的头部采用称为**序号**（sequence number）和**确认号**（acknowledgment number）的两个字段。这两个字段指的是字节序号，而不是段序号。

字节序号　TCP为在一个连接中传输的所有数据字节编号。在每个方向上序号都是独立的。当TCP接收来自进程的一些数据字节时，将它们存储在发送缓冲区中时并给它们编号。不必从0开始编码，TCP在0到$2^{32}-1$之间生成一个随机数作为第一个字节的序号，例如，如果随机数是1 057，并且发送的全部字节个数是6 000，那么这些字节序号是1 057－7 056。下面将会看到字节序号是用于流量和差错控制。

> 在每个连接中传送的字节都由TCP编号，序号开始于一个随机产生的数。

序号　字节被编号后，TCP对发送的每一个段分配一个序号，每个段的序号是这个段中的第一个字节的序号。

例23.3

假设一个TCP连接正在传送一个5 000字节的文件，第一个字节序号是10 001。如果数据被分成5个段，每一个数据段携带1 000字节，试问每个段的序列号是什么？

解：

每个段的序号如下所示：

段 1→序号：10 001　（范围：10 001－11 000）
段 2→序号：11 001　（范围：11 001～12 000）
段 3→序号：12 001　（范围：12 001－13 000）
段 4→序号：13 001　（范围：13 001～14 000）
段 5→序号：14 001　（范围：14 001～16 000）

> 一个段的序号字段的值定义为该段包含的第一个字节的序号。

当一个段携带数据和控制信息（捎带）时，它使用一个序号。如果一个段没有携带用户数据，那么它逻辑上不定义序号。虽然字段存在，但值是无效的。然而，有些段当仅携带控制信息时也需要有一个序号用于接收方的确认。这些段用做连接建立、连接终止或连接废弃。这些段中的每一个好像携带一个字节那样使用一个序号，但都没有实际的数据。如果随机产

生的序号是x，则第一个数据字节的编号是x+1。字节x被认为是一个控制段打开一个连接的一个伪字节。我们不久将会见到。

确认号　正如前面所讨论过的那样，TCP中的通信是全双工的，当建立一个连接时，双方同时都能发送和接收数据。每一方为字节编号，经常使用不同的起始字节号。每一方向的序号表明了该段所携带的第一个字节的序号。每方也使用确认号来确认它已收到的字节。但是，确认号定义了该方预期接收的下一个字节的序号。另外，确认号是累积的，这意味接收方记下它已安全而且正确地接收到最后一个字节的序号，然后将它加1，并通告这个结果作为确认号。在这里，术语"累积"指的是，如果一方使用5 643作为确认号，则表示它已经接收了所有从开始到序号为5 642的字节。但要注意，这并不是指接收方已经接收了5 642个字节，因为第一个字节的编号通常并不是从0开始的。

段中确认字段的值定义了通告一方预期接收的下一个字节的编号，确认号是累加的。

流量控制

与UDP不同的是，TCP提供流量控制。数据的接收方控制发送方发送数据的数量，这样做是为了防止接收方数据溢出。序号系统允许使用面向字节的流量控制。

差错控制

为了提供可靠的服务，TCP有一个差错控制机制。虽然差错控制以段作为差错检测（丢失或损坏段）的数据单元，稍后我们将会看到面向字节的差错控制。

拥塞控制

与UDP不同的是，TCP考虑网络中的拥塞。发送方发送的数据量不仅由接收方控制（流量控制），而且还要由网络中的拥塞程度决定。

23.3.3　段

在我们更详细地讨论TCP之前，让我们讨论TCP分组本身。在TCP中的分组称为段（segment）。

格式

段的格式，如图23.16所示。



图23.16　TCP段格式

段包括20字节到60字节的头部，接着是来自应用程序的数据。如果没有选项，那么头部

是20字节，如果有选项，最多是60字节。在本节中，我们将讨论其他头部字段。通过本章的学习可以对这些字段的意思和目的更加清楚。

- 源端口地址。这是一个16位的字段，它定义了在主机中发送该段的应用程序的端口号。这与UDP头部的源端口地址的作用一样。
- 目的端口地址。这是一个16位的字段，它定义了在主机中接收该段的应用程序的端口号。这与UDP头部的目的端口地址的作用一样。
- 序列号。这个32位的字段定义了一个数，它分配给段中数据的第一个字节。如前所述，TCP是一种字节流传输协议，为了确保连通性，对要发送的每一个字节都进行编号。序列号告诉目的端，在这个序列中哪一个字节是该段的第一个字节。在连接建立时，每一方都使用随机数生成器产生一个初始序列号（initial sequence number，ISN），通常每一个方向的ISN都不同。
- 确认号。这个32位的字段定义了段的接收方期望从对方接收的字节号。如果段的接收方成功地接收了对方发来的字节号x，它就将确认号定义为$x+1$。确认和数据可捎带一起发送。
- 头部长度。这个4位的字段指明了TCP头部中共有多少个4字节长的字。头部的长度可以在20字节到60字节之间。因此，这个字段的值在5（5×4=20）到15（15×4=60）之间。
- 保留。这个6位的字段保留为将来使用。
- 控制。这个字段定义了6种不同的控制位或标记，如图23.17所示。在同一时间可设置一位或多位。

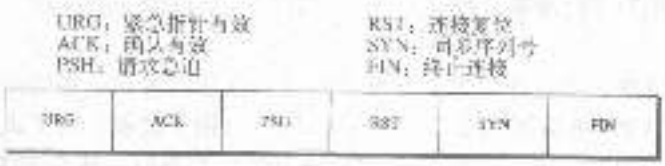


图23.17 控制字段

这些位用在TCP的流量控制、连接建立和终止、连接失败和数据传送方式等方面。表23.3显示了每位的说明。本章中，当我们讨论TCP的详细操作时，我们将对它们做深入的讨论。

表23.3 控制字段的标志描述

标 记	说 明
URG	紧急指示符字段包含有效
ACK	确认字段的值有效
PSH	请求数据推向前
RST	连接复位
SYN	在连接时对序列号进行同步
FIN	终止连接

- 窗口大小。这个字段定义对方必须维持的窗口的大小（以字节为单位）。注意：这个字段的长度是16位，这意味着窗口的最大长度是65 535字节。这个值通常称为接收窗口（rwnd），它由接收方确定。此时，发送方必须服从接收端的支配。
- 校验和。这16位的字段包含了校验和。TCP校验和的计算过程与前面描述的UDP所采用的计算过程相同。但是，在UDP数据报中校验和是可选的。然而，对TCP来说，将校验和包括进去是强制的。起同样作用的伪头部被加到段上。对TCP伪头部，协议字段的值是6。
- 紧急指示符。这个16位的字段只有当紧急标志置位时才有效。这个段包含了紧急数据。它定义了一个数，将此数加到序列号上就得出此段数据部分中最后一个紧急字节。在本章稍后将会讨论它。

• 选项。在TCP头部中可以有多达40字节的可选信息。我们这里不讨论这些选项，更多信息请参见参考文献。

23.3.4 TCP连接

TCP是一种面向连接的协议，面向连接的传输协议在源端和目的端之间建立一条虚路径。然后，属于一个报文的所有段都沿着这条虚路径发送。整个报文使用单一的虚路径有利于确认处理以及对损坏或丢失帧的重发。读者可能想知道TCP如何使用IP服务，一个无连接协议如何能面向连接。这就是TCP的连接是虚连接，而不是物理连接。TCP在一个较高层次上操作。TCP使用IP服务向接收方传递独立的段，但它控制连接本身。如果一个段丢失或损坏了，则重新发送它。与TCP不同，IP不知道这个重新发送过程。如果一个段失序到达，则TCP保存它直到缺少的段到达。IP是不知道这个重新排序过程的。

在TCP中，面向连接的传输需要三个过程：连接建立、数据传输和连接终止。

连接建立

TCP以全双工方式传输数据。当两个机器中的两个TCP建立连接后，它们就能够同时向对方发送段。这就表示，在传输数据之前，每一个方都必须对通信进行初始化，并得到对方的认可。

三次握手　在TCP中连接建立称为三次握手（three-way handshaking）。在我们的例子中，称为客户的应用程序想要与另一个称为服务器的应用程序使用TCP作为传输层协议建立连接。

该过程从服务器开始。服务器程序告诉它的TCP，它已准备好接受一个连接。这就称为请求被动打开。虽然TCP已准备好接受从世界上任何一个机器来的连接，但它自己并不能完成这个连接。

客户程序发出请求主动打开。想要与服务器进行连接的客户告诉它的TCP，它需要连接到特定的服务器。TCP现在就开始如图23.18所示的三次握手过程。为了表示该过程，我们使用了两个时间序列。每端一个。每个段头部的所有字段都有值，或许它的某些选项字也有。但是，我们每次仅表示少数几个必须要知道的字段。如果序列号、确认号、控制标记（仅是其中被置位的）和窗口大小等有值，我们就表示它们。这个阶段的三个步骤如下：

1. 客户发送的第一个段是SYN段。这个段仅有SYN标志被置位。它用于序列号同步。它占用一个序列号。当数据传输开始时，序号加1。我们说SYN段不携带实际数据，但我们可以认为它是一个字节。

> SYN段不携带数据，但它占用一个序列号。

2. 服务器发送第二个段，两个标志位SYN和ACK置位的段，即SYN和ACK段。这个段有两个目的。它是另一方向通信的SYN段，并用ACK标志作为对第一个SYN段的确认。它占用一个序列号。

> SYN + ACK段不携带数据，但它占用一个序列号。

3. 客户发送第三个段。这个段仅仅是一个ACK段，它使用ACK标志和确认序号字段来确认收到了第二个段。注意，这个段的序列号与SYN段中的序列号相同，ACK段没有占用任何序列号。

> ACK段，如果不携带数据，则它不占用序列号。

同时打开　两个进程都发出主动打开的极少发生的情况，我们称这种情况为同时打开

(simultaneous open)。在这种情况下，两个TCP彼此发送SYN + ACK段，在它们之间建立一条单独的连接。

SYN洪泛攻击 在TCP中，连接建立过程易遭受到称为SYN洪泛攻击(SYN flooding attack)的严重安全问题。在当一个恶意的攻击者将大量的SYN段发送到一个服务器，在数据报中通过伪装源IP地址假装这些段来自不同的客户端时就发生了这种情况。假定客户机发出主动打开，服务器分配必要的资源，如生成通信表和设置计时器等。然后服务器送SYN + ACK段给这些假装客户，但这些段都丢失了。然而，在这段期间，许多资源被占用但没有被使用。如果在短时间内，SYN段的数量很大，服务器最终耗尽它的资源而崩溃。这种SYN洪泛攻击属于一种称为**拒绝服务攻击**(denial-of-service attack)的安全攻击类型。此时一个攻击者独占系统如此多的服务请求使得系统崩溃，拒绝对每个请求提供服务。

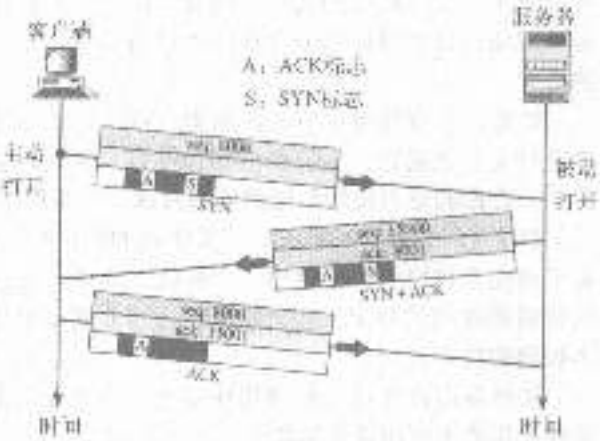


图23.18 使用三次握手建立连接

TCP的某些实现有减轻SYN攻击影响的策略。有些实现在特定时间周期内对连接请求进行限制，另一些实现过滤掉来自不需要的源地址的数据报。一个新的策略使用cookie推迟资源分配直到一个完整的连接建立。新的传输层协议SCTP使用这种策略，我们将在下一节中讨论。

数据传输

连接建立后，可进行双向数据传输(data transfer)，客户端与服务器双方都可发送数据和确认。在本章稍后我们将学习确认的规则。目前，知道在同一段内携带确认时，在同一方向上也可以传输数据就行了。这就是数据捎带确认。一个例子如图23.19所示，在连接建立后(图中没有表示出)，客户端用两个段发送2 000个字节的数据。然后，服务器用一个段发送2 000个字节的数据。客户端发送另一个段。前面三个段携带数据与确认，但是最后一个段仅携带确认，这是因为已没有数据发送了。注意序列号与确认号的数值。客户端发送的数据段有PSH(急迫)标志，所以服务器TCP知道当收到数据后立刻传递给服务器进程。稍后我们讨论这个标志的用法。另一方面，来自服务器的段没有设置急迫标志。大多数TCP的实现都有可选标志，可设置或不设置。

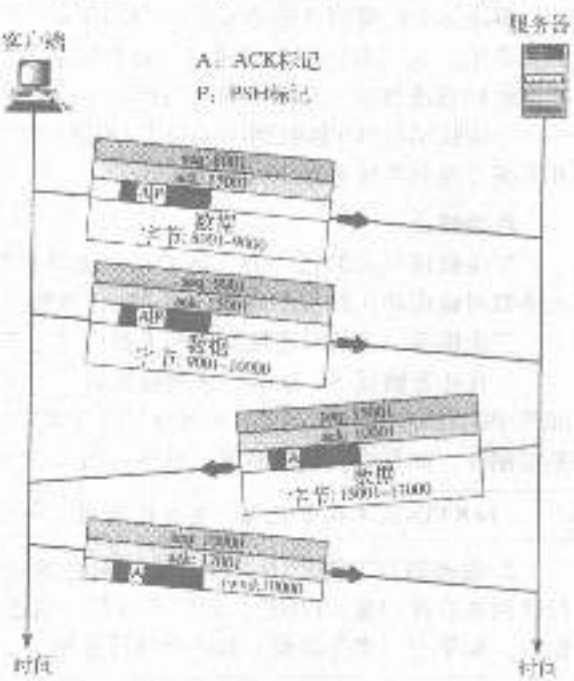


图23.19 数据传输

急迫数据　我们知道发送方的TCP使用缓冲区存储来自发送方应用程序的数据流。发送方的TCP可以选择段的大小。接收方的TCP在数据到达时也将数据进行缓存，并当应用程序准备就绪时或当接收端TCP认为方便时将这些数据传递给应用程序。这种灵活性增加了TCP的效率。

但是，在有些情况下，应用程序并不需要这种灵活性。例如，应用程序与另一方应用程序进行交互式通信。一方的应用程序打算将其击键发送给对方应用程序，并希望接收到立即响应。数据的延迟传输和延迟传递对这个应用程序来说是不可接受的。

TCP可以处理这种情况。在发送端的应用程序可请求急迫操作。这就表示发送端的TCP不必等待窗口被填满。它创建一个段就立即将其发送。发送端的TCP还必须设置急迫位（PSH）以告诉接收端的TCP，这个段所包括的数据必须尽快地传递给接收应用程序，而不要等待更多数据的到来。

虽然急迫操作可由应用程序提出，但是现在的大多数实现却忽视了这些请求。TCP可以选择使用或不使用这个操作。

紧迫数据　TCP是面向字节流的协议。这就是说，从应用程序到TCP的数据被表示成一串字节流。数据的每一个字节在流中占有一位置。但是，在某些情况下，应用程序需要发送紧急字节。这表示发送应用程序希望某一块数据由接收应用程序不按序读出。作为一个例子，假定发送应用程序正在发送被接收应用程序处理的数据。当处理结果返回时，发送应用程序出现了某些差错，它希望中止此过程，但它已发送了大量的数据。如果它发送出异常中止命令（control＋C），这两个字符将被存放在接收端TCP缓冲区的末尾。当所有的数据被处理完毕时，这两个字符才被传递给接收应用程序。

解决这个问题的方法是发送将URG位置1的段。发送应用程序告诉发送端的TCP这块数据是紧急的。发送端的TCP创建段，并将紧急数据放在段的开始，段的其余部分可以包括来自缓冲区的普通数据。头部中的紧急指针字段定义了紧急数据的结束和普通数据的开始。

当接收端的TCP接收到URG位置1的段时，它就利用紧急指针的值从段中提取出紧急数据，并不按序地将其传递给接收应用程序。

连接终止

交换数据双方的任一方（客户或服务器）都可关闭连接，通常它是由客户端发起。当前大多数对连接终止的实现有两个方法：三次握手和带有半关闭选项的四次握手。

三次握手　当前对连接终止的多数实现是三次握手，如图23.20所示。

1. 在正常情况下，从客户进程接收到一个关闭命令后，客户的TCP发送第一个段：FIN段，即其中的FIN位置位。注意：FIN段可包含客户机要发送的最后数据块，或如图23.20所示的只是控制段。如果它只是控制段，它仅占用一个序列号。

> 如果FIN段不携带数据，则该段占用一个序列号。

2. 服务器TCP接收到FIN段后，通知它的进程，并发送第二个段：FIN＋ACK段，证实它接收到来自客户端的FIN段，同时通告另一端连接关闭。这个段还可以包含来自服务器最后数据块。如果它不携带数据，则这个段仅占用一个序列号。

> 如果FIN＋ACK段没有携带数据，则该段仅占用一个序列号。

3. 客户端的TCP发送最后一段：ACK段，证实它接收到来自服务器的FIN段。这个段包含确认号，它是接收到来自服务器FIN段的序号加1。这个段不携带数据也不占用序列号。

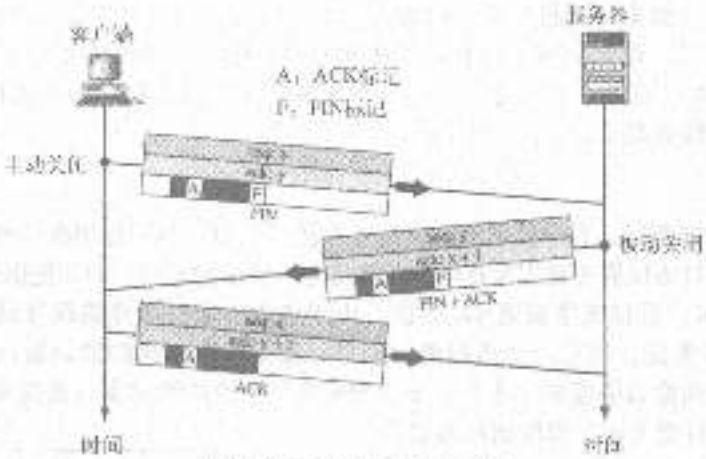


图23.20 一次握手的连接终止

半关闭 在TCP中，一端可以停止发送数据后，还可以继续接收数据。这就是所谓的半关闭（half-close）。虽然任一端都可发出半关闭，但通常都是由客户端发起的。当服务器在开始处理之前需要所有数据时，就会出现半关闭。例如，排序是一个很好的例子。当客户端发送数据给服务器进行排序时，在开始排序之前，服务器需要接收到全部数据。这就是说客户端发送完全部数据之后，它在向外方向可连接可关闭。但在向内方向，为了接收排序数据必须保持打开。服务器在接收到数据后还需要时间进行排序，它的向外方向必须保持打开。

图23.21表示了半关闭的例子。客户端发送FIN段半关闭连接，服务器通过发送ACK段确认半关闭。从客户端到服务器的数据传输结束，但是服务器还可以发送数据。当服务器已经发送完被处理的数据时，它发送一个FIN段，该FIN段由客户端的ACK来确认。

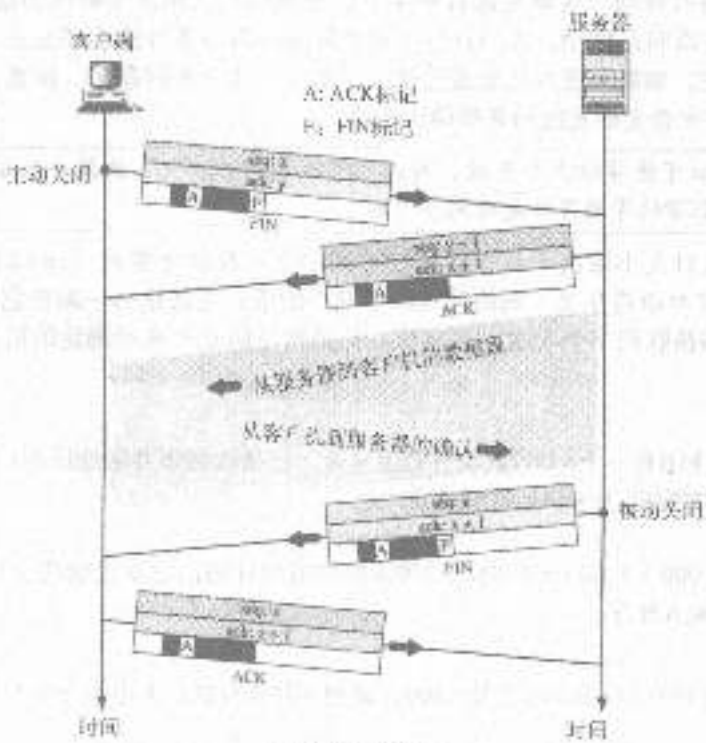


图23.21 半关闭

连接半关闭后，数据可从服务器传送给客户端，而确认可从客户端传送给服务器。注意所用的序列号，第二个段（ACK）没有占用序列号。虽然客户端已接收了序列号$y-1$，正在期望的是y，但服务器的序列号还是$y-1$。当连接最后关闭时，最后ACK段的序列号还是x。因为在那个方向传输数据时，不占用序列号。

23.3.5 流量控制

如第11章讨论的那样，TCP使用滑动窗口处理流量控制。TCP使用滑动窗口协议，但是，TCP使用的滑动窗口协议界于回退N帧与选择重发之间的滑动窗口。TCP提供的滑动窗口协议由于它不使用NAK，看起来像回退N帧协议；由于接收方保存失序的段直到丢失的段到达，它看起来又像选择重发。可是，这里的滑动窗口与数据链路所用的滑动窗口有两大点不同。第一点，TCP的滑动窗口是面向字节的，而数据链路层讨论的滑动窗口是面向帧的；第二点，TCP的滑动窗口是可变大小，而数据链路层讨论的滑动窗口是固定大小。

图23.22表示了TCP中的滑动窗口，窗口覆盖了存储着从进程接收到的字节的一部分缓冲区。窗口内的字节都是在传输中的字节，可以发送它们而不必考虑来自另一端的确认。这个设想的窗口有两个边沿：左边沿和右边沿。



图23.22　滑动窗口

窗口是张开的、合拢的或收缩的。正如我们将看到的，这三种活动是由接收方控制（取决于网络中的拥塞）而不是发送方。在这个问题上，发送方必须服从接收方的命令。

窗口张开是指右边沿向右移动。它将允许缓存区中存储符合发送条件更新的字节。窗口合拢是指左边沿向右移动，这就是说有些字节已被确认，发送方不必再为这些字节而担忧。窗口收缩是指右边沿向左移动。因为这种活动废除某些符合条件的字节发送，因此强烈建议在实现中不使用它。如果发送方已发送过这些字节了，这就有问题了。注意：左边沿不能向左移动，因为这将宣告先前发送的某些确认无效。

> 使用滑动窗口可使传输更加有效，同时也可以控制数据流，使得目的端不致因数据来得过多而瘫痪。TCP的滑动窗口是面向字节的。

一端的TCP窗口大小取决于两个值中较小的一个：接收方窗口（rwnd）值和拥塞窗口（cwnd）。接收方窗口值是在另一端的确认段中宣布的值。它就是另一端在它的缓存溢出和数据被丢弃之前能够接收的字节个数。拥塞窗口值是网络防止拥塞所确定的值。我们将在本章后面讨论拥塞。

例23.4

如果接收方主机B有一个5 000个字节的缓冲区，已接收到但并未处理的1 000个字节数据。试问主机A的接收方窗口（rwnd）的值是多少？

解：

*rwnd*的值＝5 000－1 000－4 000，主机B在它的缓冲溢出之前仅能接收4 000个字节。主机B在下一段向主机A宣布。

例23.5

如果*rwnd*值是3 000，而*cwnd*值是3 500，试问主机A的窗口大小是多少？

解：

窗口大小是*rwnd*和*cwnd*中较小的一个，它应是3 000个字节。

例23.6

图23.23表示了一个不切实际的滑动窗口，发送方已发送了202个字节，我们假定*cwnd*是20（实际上，这个值是上千字节），接收方已发一个*rwnd*为9（实际上，这个值也是上千字节）而确认号为200的ACK段。发送窗口的大小是*rwnd*和*cwnd*的最小值，即9个字节。字节号200到202都已发送但未被确认。字节号203到208可以发送而不必考虑从另一端来的确认。字节号209和它以上各字节不可能发送。



图23.23 例23.6

TCP滑动窗口的要点如下：

- 窗口大小是*rwnd*和*cwnd*中的最小值。
- 发送方不必发送一个全窗口大小的数据；
- 接收方可张开或合拢窗口，但不能收缩窗口；
- 只要不引起窗口收缩，接方可随时发送一个确认；
- 接收方可暂时关闭窗口，但在窗口关闭后发送方总是发送一个1字节的段。

23.3.6 差错控制

TCP是一个可靠的传输层协议。这意味着将数据流传递给TCP的应用程序依靠TCP将整个数据流传递给另一端的应用程序，并且是按序的、无差错的、没有任何一部分丢失或重复的。

TCP使用差错控制提供可靠性。差错控制包括一些机制：检测受到损坏的段、丢失的段、失序的段和重复的段。差错控制还包括检测出差错后纠正差错的机制。TCP中的差错检测和纠正是通过三种简单工具来完成：校验和、确认和超时。

校验和

每个段都包括校验和字段，用来检查受到损坏的段。如果段被损坏，它将被目的端TCP丢弃，并认为是丢失了。TCP在每段中强制使用一个16位的校验和。在第24章中，我们将会看到这16位的校验和对于新的传输层协议SCTP是不适合的。因为SCTP将整个头部格式重新配置，而TCP不可能改变它。

确认

TCP使用确认方法来证实收到了数据段。不携带数据但占用序列号的一些控制段也要确认，但ACK段是不要确认的。

> ACK段不占用序列号，它不需要确认。

重传

差错控制机制的核心是段的重传。当一个段损坏、丢失或延迟时，就重传这个段。在当前实现中，有两种情况要重传段：重传计时器（retransmission timer）到时或当发送方收到三个重复的ACK时。

> 在当前的实现中，如果重传计时器到时或当发送方收到三个重复的ACK时，则发生重传。

注意：不占用序列号的段不重传，尤其是，ACK段不重传。

> 对ACK段不设置重传计时器。

RTO后重传 TCP发所有重要的段（已发送，但还未被确认）的最新实现使用了一个重传

超时（retransmission time-out，RTO）计时器。当计时器到时，即使可能是由于段被延迟、ACK被延迟或丢失确认等未接收到ACK时，重发一个最早的未要的段。注意：对仅携带确认的段没有设置超时计时器。这说明这样的段不需要重发。在TCP中，RTO的值是动态的，根据段的往返时间（round-trip time，RTT）为进行更新。RTT是一个段到达目的端并接收到一个确认所需要的时间。它使用往返策略与第12章中讨论的相似。

三个重复ACK段之后的重发　如果RTT的值不是很大，段重发前的规则是可用的。但有时某一段丢失了，而接收端收到了许多失序的段，它们不可能都被存储（有限的存储器大小）。为了缓解这种情况，当今大多数实现遵循三次重复ACK规则，立即重发缺少的段。这一特点称为**快速重传**（fast retransmission），在稍后的第中我们将会看到。

失序的段

当一个段被延迟、丢失或废弃时，后面一些段的到达就失序了。TCP最初的设计是丢弃所有失序的段，并重传那个缺少的段以及其后的一些段。当今大多数的实现不丢弃失序的那些段，暂时将它们存储，并标记它们为失序的段直到缺少的那个段到达。但是注意：这些失序的段不传递给进程，TCP确保数据按序传递给进程。

> 数据可以失序到达，并被接收的TCP暂时存储，但是TCP确保传递给进程的段是无失序的。

某些情况

在本节中，我们给出TCP操作时某些情况的例子。在这些情况中，用长方形表示段，如果段携带数据，我们显示字节序列号的范围和确认字段的值。如果段仅是一个确认，我们仅用小方框中的确认号表示。

正常操作　第一种情况表示了两个系统之间的双向数据传输，如图23.24所示。客户TCP发送一个段，而服务器TCP发送二个段。图23.24表示了应用于每个确认的规则。有数据要发送，所以该段显示期望接收的下一个字节序号。当客户端接收到来自服务器的第一个确认时，它没有更多的数据要发送，它仅发送一个ACK段。但是，该确认段需要延迟500ms以观察是否还有更多的段到达。当计时器到时，客户端触发一个确认。这样做是因为客户端不知道是

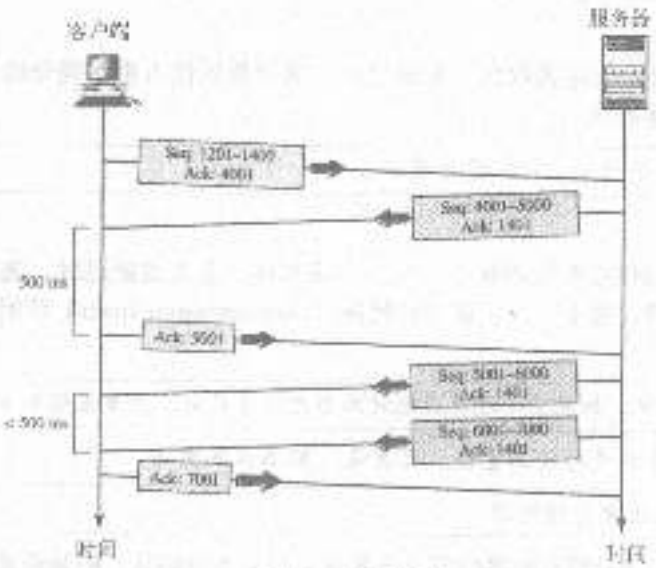


图23.24　正常操作

否还有其他的段到来，它不再延迟这个确认。当下一个段到达时，启动另一个确认计时器。但是，在它到时之前，第二个段到达，第二个段的到达触发另一个确认。

丢失的段 在这种情况下，我们说明了当一个段丢失或损坏时，将会发生什么。接收方对丢失的段和损坏的段用同样的方法处理。丢失的段是在网络中的某处被丢失的。损坏的段是被接收方本身丢弃的。图23.25表示了一种情况，一个段被丢失，被在网络中的某个路由器丢弃掉，或许由于拥塞引起。

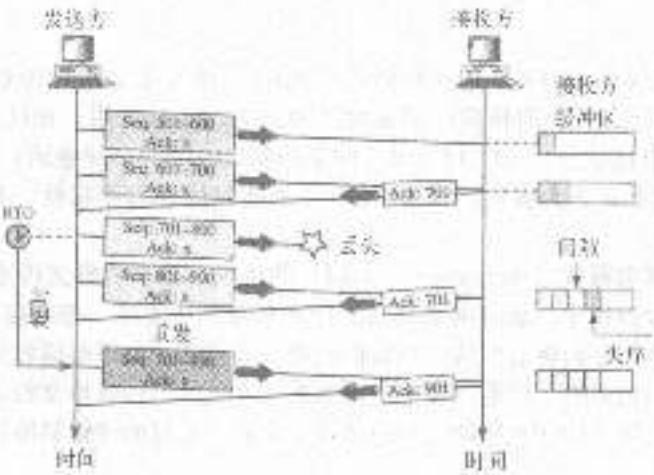


图23.25 丢失的段

我们假定数据传输是单向的，一方发送，另一方接收。在这个特定情况下，发送方发送段1和段2，这两个段立即被一个ACK确认。但是，段3被丢失了，接收方接收到段4，发生了失序。接收方将数据在储存在它的缓冲器的段中但留出一个间隙指明数据中存在不连续性。接收方立即给发送方发送一个确认，表示了它期望的下一个字节。注意：接收方存储从801到900的字节，但在这个间隙被填充之前不将这些字节传送给应用程序。

> 接收方TCP仅将有序的数据传送给进程。

我们已表示了最早的重要段的计时器，因为接收方永远不会给丢失或失序段发送确认。因此这个定时器肯定会超时。当该计时器到时时，发送方TCP重发段3，这个到达时才真正地被确认。注意：按照相应的规则，第二个确认和第二个确认的值是不同的。

快速重传 在这个情况中，我们要表示快速重传的思想。除了RTO具有较大的值外，该情况与第一种情况相同(见图23.26)。

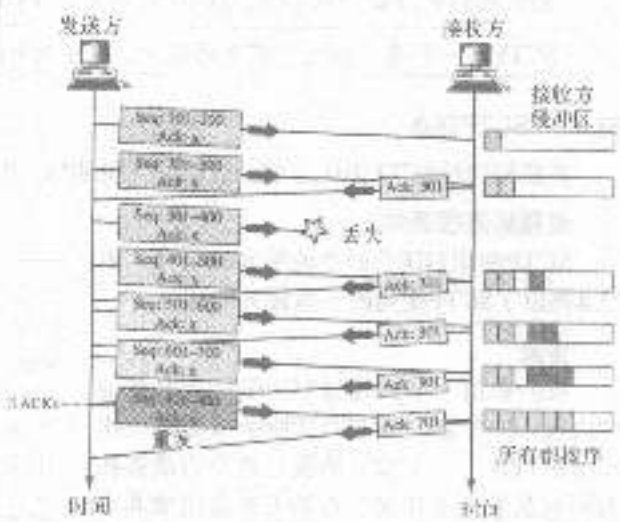


图23.26 快速重传

当接收方接收到第四个、第五个和第六个段时，它触发一个确认。发送方接收到4个具有相同值的确认（3个是重复的）。虽然段3的计时器还没有到时，但快速传输要

求立即重发段3，该段是所有这些确认所期望的。注意：虽然4个段都没有被确认，但仅仅重发一个段。当发送方接收到这个重发的ACK时，因为确认是累计的，因此它知道这4个段是安全的和可靠的。

23.3.7 拥塞控制

我们将在第24章讨论TCP的拥塞控制。

23.4 SCTP

流控制传输协议（SCTP）是一种新的可靠的、面向报文的传输层控制协议。然而，SCTP主要是为最近引入的因特网应用而设计的。这些新的应用，如IUA（IP上的ISDN）、M2UA和M3UA（电话信令）、H.248（媒体网关控制）、H.323（IP电话）和SIP（IP电话）等都需要提供比TCP更复杂的服务。SCTP提供了更高的性能和可靠性。我们简要比较UDP、TCP和STCP的性能。

- UDP是一个面向报文（message-oriented）的协议，进程将报文传送给UDP，将它封装在一个用户数据报中，通过网络发送。UDP保留报文边界，每个报文与其他报文无关。正如我们在本书后面将看到的，当我们处理一些应用，如IP电话和实时数据传输时，这个特性是我们想要的。但是UDP是不可靠的，发送方不知道报文的命运，报文可能会丢失、重复或失序。UDP也还缺乏其他特性，如缺少友好的传输层协议所需要的拥塞控制和流量控制。
- TCP是一个面向字节（byte-oriented）的协议。它从进程中接收一个报文或多个报文，并以一个字节流的方式存储它们，并以段的方式发送它们。没有保存报文的边界。但是，TCP是一个可靠的协议。重复段会被删除，丢失的段会重发，并且字节按序传递到端进程。TCP还有拥塞控制和流量控制机制。
- SCTP兼有UDP和TCP的特性。SCTP是一个可靠的面向报文的协议，它保存报文的边界，同时它检测丢失的数据、重复的数据和失序的数据。它还有拥塞控制和流量控制机制。稍后我们将会看到SCTP还有UDP和TCP中没有的一些新特性。

> SCTP是一个面向报文的可靠的协议，它兼有UDP和TCP好的特性。

23.4.1 SCTP服务

在我们讨论SCTP操作之前，让我们先说明SCTP向应用层进程提供的服务。

进程到进程通信

SCTP使用TCP空间中的熟知端口号，表23.4列出了SCTP使用的一些额外的端口。

表23.4　某些SCTP应用

协　议	端　口　号	说　明
IUA	9990	IP上的ISDN
M2PA	2904	SS7电话信令
M2UA	2905	SS7电话信令
H.248	2945	媒体网关控制
H.323	1718、1719、1720、11720	IP电话
SIP	5060	IP电话

多流

我们在前一节讨论过TCP是一种面向流的协议，TCP客户端与TCP服务器的每一次连接都包含了一个单一的流。这个方法存在的问题是在流中任何一点的丢失会阻塞其余数据的传递。当传输文本时，这还可以接受，但当传输实时数据（音频或视频）时，这就不能接受了。SCTP在每一次连接中提供多流服务（multistream service），用SCTP专门名词称为关联。如果多流中某一个被阻塞，其余的

流仍然可传送它们的数据。这种思想类似于高速公路上的多车道，每一个车道可用于不同类型的交通量。例如，一条车道用于正常的交通，另一条车道用于合伙车。如果正常的交通受到阻塞，合伙车辆依旧可达目的地。图23.27表示了多流传递的思想。

SCTP的一次关联能包含多个流。

多接口

一个TCP连接包括了一个源IP地址和一个目的IP地址。这就是说，即使发送方或接收方是一个多接口主机（即连接到带有多个IP地址的多个物理地址上），在连接期间每端的这些IP地址中仅有一个是可用的。另一方面，SCTP关联支持多接口服务（multihoming service）。发送主机和接收主机对每一次关联在每一端可定义多个IP地址。在这个容错方法中，当一条通路发生故障时，可以用另一个接口不中断地传递数据。当发送和接收实时有效载荷时，如因特网上的电话，这个特性是很有用的。图23.28表示了多接口的思想。

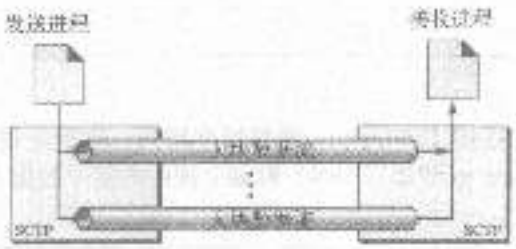


图23.27 多流的概念



图23.28 多接口概念

在图23.28中，客户端用两个IP地址分别连接两个局域网络，服务器也用两个IP地址分别连接两个网络。客户端和服务器可用四个不同的IP地址对做出一个关联，但是应注意，在目前SCTP的实现中，仅有一对IP地址可用于正常的通信，当选用的主路径有故障时，可使用替换路径。换言之，目前SCTP不允许不同路径共用。

SCTP关联允许每个端有多个IP地址。

全双工通信

像TCP一样，SCTP提供全双工服务，在全双工服务中，数据可同时在两个方向流动。每一个SCTP都有一个发送和接收缓存区，在两个方向都可以发送分组。

面向连接服务

像TCP一样，SCTP是面向连接的协议。但是，在SCTP中，连接被称为关联。站点A的进程要向站点B的另一个进程发送或接收数据时，其步骤如下：

1. 两个SCTP彼此建立关联；
2. 双方向交换数据；
3. 终止关联。

可靠的服务

像TCP一样，SCTP是可靠的传输协议。它利用确认机制检测数据是否安全和可靠地到达。我们将在差错控制这一节进一步讨论这个特性。

23.4.2 SCTP特性

首先让我们讨论SCTP的一般特性，然后将这些特性与TCP特性相比较。

传输序列号

TCP的数据单元是字节，在TCP中用序列号对字节编号控制数据传输。另一方面，SCTP的数据单元是**数据大块**（DATA chunk）。由于分段原因，数据大块与来自进程的报文可能有也可能没有一一对应的关系（以后讨论）。在SCTP中用数据大块的编号控制数据传输，SCTP使用**传输序列号**（transmission sequence number，TSN）对数据大块进行编号。换言之，SCTP中的TSN所起的作用类似于TCP中的序列号，TSN是32位长，初始值可取0到$2^{32}-1$之间的随机数。每个数据大块的头部必须有相应的TSN。

> 在SCTP中，数据大块按TSN编号。

流标识符

在TCP中，每一次连接仅有一个流。而在SCTP中，每次关联可以有多个流。因此，每个流需要用**流标识符**（SI）进行标识。每个数据大块的头部必须有SI，这样当它到达目的端时，它可正确地放入它的流中。SI是从0开始的16位数字。

> 为了区别不同的流，SCTP使用SI。

流序列号

当一个数据大块到达目的SCTP时，它被传递到适当的流中，并按顺序排列。这就是说，除了SI之外，SCTP还用**流序列号**（stream sequence number，SSN）对每个流中的每个数据大块进行定义。

> 为了区别属于同一个流中的不同数据大块，SCTP使用SSN。

分组

在TCP中，段携带数据和控制信息。数据作为一组字节被携带，而控制信息是用它头部的6个控制标记来定义，而SCTP的设计完全不同：数据作为数据大块被携带，控制信息作为控制大块被携带。多个控制大块与数据大块可合并成一个分组。SCTP中一个分组起的作用与TCP中的段相同。图23.29比较了TCP的段与SCTP的分组。让我们扼要列出SCTP的分组和TCP的段不同：

> TCP有段；SCTP有分组。

1. TCP中的控制信息在头部，而SCTP中的控制信息是在控制大块中。有几种类型的控制大块，每个类型用于不同的目的。

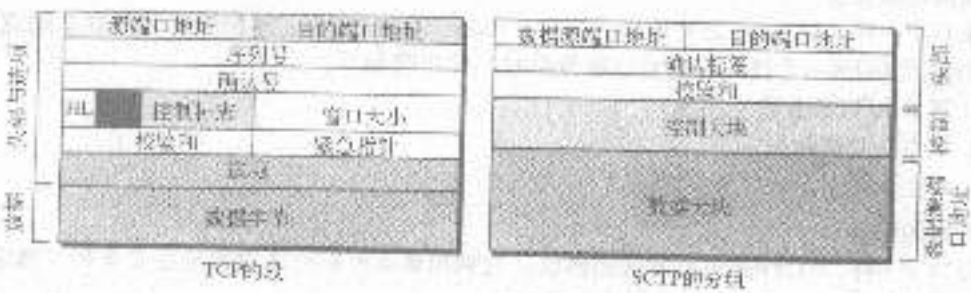


图23.29 TCP段与SCTP分组的比较

2. 一个TCP段中的数据作为一个实体处理，而一个SCTP分组可以携带多个数据大块。每

个可能属于不同的流。

3. 选项可以是TCP段的一部分，而SCTP分组不存在选项。SCTP中的选项是由定义新的大块字节来处理的。

4. TCP头部的强制性部分是20个字节，而SCTP头部仅是12个字节。SCTP头部较短是由于下列因素：

a. SCTP序列号（TSN）属于每个数据大块，而非它位于数据大块的头部；

b. 确认号和窗口大小都是每个控制大块的一部分；

c. SCTP头部的长度是固定的（12个字节），因此不需要头长度字段（TCP段中用HL表示），不存在用选项生成头部的长度；

d. 在SCTP中没有紧急指针。

5. TCP中的校验和是16位，而SCTP中的校验和是32位。

6. SCTP中确认（verification）标签是关联标识符；而在TCP中是没有的。TCP中的IP地址和端口地址一起定义了一个连接，而在SCTP中，用不同的IP地址实现多接口，定义每个关联都需要唯一的确认标签。

7. TCP段的头部有一个序列号，它定义数据部分中的第一个字节编号；而SCTP分组可包含多个数据大块，用TSN、SI和SSN定义每个数据大块。

8. TCP中有些段携带控制信息（如SYN和FIN），它要占用一个序列号；在SCTP中，控制大块不使用TSN、SI和SSN。这三个标识符仅属于数据大块，而不是属于整个大块。

> 在SCTP中，控制信息和数据信息在分开的大块中携带。

在SCTP中，有数据大块、流和分组。一个关联可发送多个分组，一个分组可包含多个大块，而各个大块可属于不同的流。为了使这些术语定义得更清晰，让我们假定进程A用3个流发送11个报文给进程B。前面的4个报文用第一个流，其次的3个报文用第二个流，最后的4个报文用第三个流。

虽然一个长的报文可能被多个数据大块携带，但是我们还是假定每个报文装载在一个数据大块中。因此，我们在3个流中有11个数据大块。

应用进程传递11个报文给STCP，这里对每个报文标上适合的流记号。虽然进程可传递第一个流的一个报文，然后再传递第二个流的另一个报文，但是，我们假定首先传递属于第一个流的所有报文，然后再传递属于第二个流的所有报文，最后传递属于第三个流的所有报文。

我们还假定网络仅允许每个组有三个数据大块，这就是说，我们需要4个分组如图23.30所示。第一个分组和一部分第二个分组流0中的数据大块。第二个分组和第三个分组携带流1中的数据大块，第三个分组和第四个分组携带流2中的数据大块。

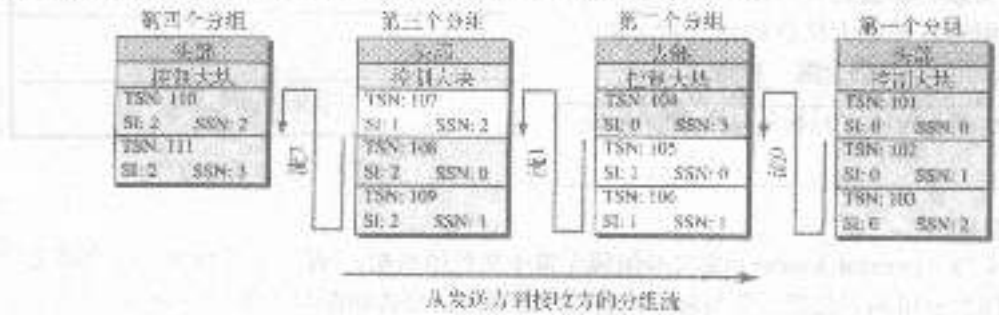


图23.30 分组、数据大块和流

注意每个数据大块需要三个标识符：TSN、SI和SSN。TSN是累计数，将会看到它也用于流量控制和差错控制。SI定义这些块所属的流。SSN定义了在特定流中数据大块的次序。在我们的例子中，对每个流，SSN都从0开始编号计数。

数据大块由三个标识符来标识：TSN、SI和SSN。TSN是用来标识关联的累计数；SI定义流；SSN定义流中的块。

确认号

TCP确认号是面向字节的，它指的是序列号。而SCTP确认号是面向大块的，它指的是TSN。TCP与SCTP确认号的第二个不同是控制信息。在TCP中控制信息是段头部的一部分。为了让确认段仅携带控制信息，TCP使用序列号和确认号（例如，SYN段需要用ACK段确认）。但在SCTP中，控制大块携带控制信息，它不需要TSN，而这些控制大块都由另一个适当类型的控制大块来确认（有些不需要确认），例如，INIT控制大块是由INIT ACK大块来确认，它不需要序列号或确认号。

在SCTP中，确认号仅用于对数据大块的确认，如果需要的话，控制大块用另一个控制大块来确认。

流量控制

像TCP一样，SCTP实现流量控制以避免接收方崩溃。我们将在本章后面讨论SCTP的流量控制。

差错控制

像TCP一样，SCTP实现差错控制以提供可靠性。TSN号和确认号用于差错控制。我们将在本章后面讨论差错控制。

拥塞控制

像TCP一样，SCTP实现拥塞控制以决定多少个数据大块可置于网络中。我们将在第24章讨论拥塞控制。

23.4.3 分组格式

在这一节，我们说明了分组格式和大块的各种类型。在本节中所列出的大多数信息将在后面变得更加清楚。第一次阅读可以跳过本节或仅作为参考。SCTP分组有一个强制性的通用头部和一个称为大块的块的集合。大块有两种类型：数据大块和控制大块。控制大块控制和维护关联，而数据大块携带用户数据。在一个分组中，控制大块在数据大块之前。图23.31表示了SCTP的分组一般格式。



图23.31 SCTP的分组格式

在SCTP分组中，控制大块在数据大块之前。

通用头部

通用头部（general header）定义分组属于每个关联的两端，保证了分组属于一个特定的关联，并保护分组内容包括本身头部的完整性。通用头部格式如图23.32所示。

通用头部有4个字段：

- 源端口地址。这是一个16位字段，它定义进程发送分组的端口号；
- 目的端口地址。这是一个16位字段，它定义进程接收分组的端口号；
- 确认标签。这是一个数，它将一个分组与一个关联相匹配。它防止将以前关联的一个分组误认为是这个关联中的一个分组，它作为关联的一个标识符。在关联期间，它在每个分组中都是相同的。在关联中，每个方向有各自的确认。
- 校验和。这是一个32位字段，它包含CRC-32校验和。注意校验和的长度从16位（UDP、TCP和IP）增加到32位，并允许用CRC-32校验和。

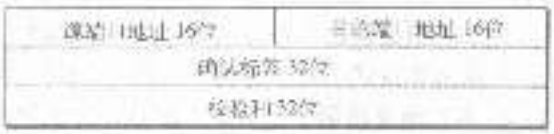


图23.32 通用头部

大块

大块携带控制信息或用户数据。每块的详细格式超出了本书的范围，详细内容可阅读[For06]。所有大块的前三个字段是相同的，信息字段取决于大块类型。最重要的是，SCTP要求信息部分是4字节的倍数；如果不是，在信息部分的末尾增加填充字节（8个0）。一些大块和它们的说明见表23.5。

表23.5 大块

类 型	大 块	说 明
0	DATA	用户数据
1	INIT	建立关联
2	INIT ACK	INIT大块的确认
3	SACK	选择性的确认
4	HEARTBEAT	探测对等方当前可达性
5	HEARTBEAT ACK	确认HEARTBEAT大块
6	ABORT	异常中止关联
7	SHUTDOWN	关闭关联
8	SHUTDOWN ACK	确认SHUTDOWN大块
9	ERROR	报告没有致命错误
10	COOKIE ECHO	关联建立中的第三个分组
11	COOKIE ACK	确认COOKIE ECHO大块
14	SHUTDOWN COMPLETE	关闭关联中的第三个分组
192	FORWARD TSN	调整累计TSN

23.4.4 SCTP关联

像TCP一样，SCTP是一个面向连接的协议。但是，为了强调其多接口性，SCTP中的连接被称为关联。

> SCTP中的连接被称为关联。

关联建立

SCTP中的关联建立（association establishment）要求四次握手（four-way handshake）。在这个过程中，一个进程（通常是客户）想要与另一个进程（通常是服务器）建立关联。使用SCTP作为传输层协议。与TCP相似，SCTP服务器需要准备接收一个关联（被动打开）。

而由客户发起建立关联（主动打开）。SCTP关联建立，如图23.33所示。在正常情形下，步骤如下：

1. 客户端发送第一个分组，它是一个INIT大块（INIT chunk）；

2. 服务器发送第二个分组，它是一个INIT ACK大块（INIT ACK chunk）；

3. 客户端发送第三个分组，它包含一个COOKIE ECHO大块（COOKIE ECHO chunk），这是一个很简单的大块。它毫无改变地回应由服务器发送的cookie。SCTP允许在这个分组中包含数据大块；

4. 服务器发送第四个分组，它包含对接收到COOKIE ECHO大块进行确认的COOKIE ACK大块（COOKIE ACK chunk）。SCTP允许在这个分组中包含数据大块。

携带INIT或INITACK大块的分组中不允许有其他大块；而COOKIE ECHO或COOKIE ACK大块可携带数据大块。

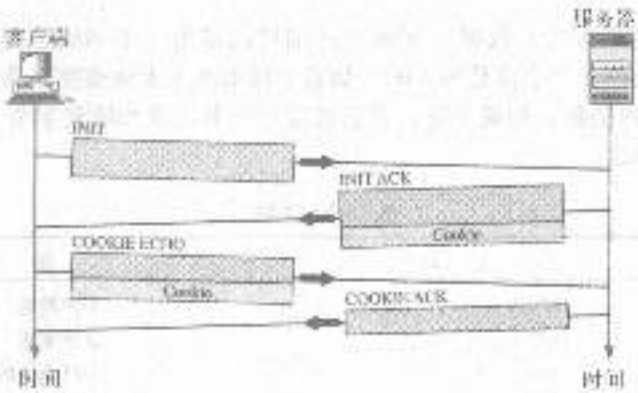


图23.33　四次握手

Cookie　在前几节中，我们讨论过SYN洪泛攻击。利用TCP，一个恶意的攻击者可以向一个TCP服务器发送大量的具有不同假IP地址的伪造的SYN段。每次服务器接收到一个SYN段时，它建立一个状态表并分配更多的资源等待下一个段的到达。然而，不久服务器由于资源耗尽而崩溃。

SCTP设计者使用一种策略防止这种类型的攻击。当发送方IP地址被确认时，这个策略是推迟资源分配直到接收到第三个分组。第一个分组中接收到的信息必须以某种方式保存到第二个分组到达。但是，如果服务器保存该信息，那就需要资源（存储器）的分配。这是一个困难。解决办法是包装该信息，并把它送回到客户。这称为生成一个cookie，这个cookie与第二个分组一起发送给接收第一个分组的地址。有两种可能的情形：

1. 如果第一个分组的发送方是一个攻击者，那么服务器永远不会接收到第三个分组，cookie丢失了，但没有分配资源，且服务器的唯一的工作是“生成”（“baking”）cookie。

2. 如果第一个分组的发送方是需要建立连接的一位诚实的客户，那么它接收到带有cookie的第二个分组。它毫无改变地用该cookie发送一个分组（在序列中是第三个分组）。服务器接收到第三个分组而且知道它来自一位诚实的客户，当为发送方发送的cookie已经在那里。现在服务器可以分配资源了。

如果没有实体可以“吃掉”由服务器“生成”的cookie，那么上述策略就可以起作用。为了保证这一点，服务器用它自己的密钥产生该信息的一个摘要（见第30章）。信息和摘要一起生成一个cookie，它在第二个分组中被发送给客户。当cookie在第三个分组中返回时，服务器

计算信息的摘要。如果该摘要与发送的摘要相匹配，那么该cookie就没有被其他的实体修改。

数据传输

关联的目的是为了在两端之间传输数据。关联建立后，就可以进行双方向的数据传输。客户端与服务器都可发送数据。像TCP一样，SCTP支持捎带。

然而TCP与SCTP在数据传输方面存在着很大的不同。TCP从进程读取报文。这些报文作为字节流在其间没有可识别的边界。为了它的对等方的使用，进程可插入一些边界，但TCP将那些标记作为文本的一部分对待。换言之，TCP把每个报文附加到它的缓存中，一个段可携带两个不同的报文。TCP使用的唯一的排序系统是字节编号。

另一方面，SCTP识别并维护边界，来自进程的每个报文作为一个单元处理，并插入到一个DATA大块中除非这个报文被分段（后面讨论）。在这种情况下，SCTP与UDP一样具有一个很大优点是数据大块彼此有关。

从进程接收到的一个报文成为一个DATA大块，如果要分段，则成为几个DATA大块。把DATA大块的头部加到报文上。一个报文或一个报文的分段形成的每个DATA大块都有一个TSN。我们要记住，只有DATA大块使用TNS，也只有DATA大块使用SACK大块进行确认。

> 在SCTP中，只有DATA大块占用TNS，也只有这种大块要进行确认。

图23.34表示了一个简单数据传输的情形，客户端发送4个DATA大块，从服务器接收两个DATA大块。稍后，我们将讨论SCTP中的流量和差错控制，此时假定在这种情形下一切都顺利。

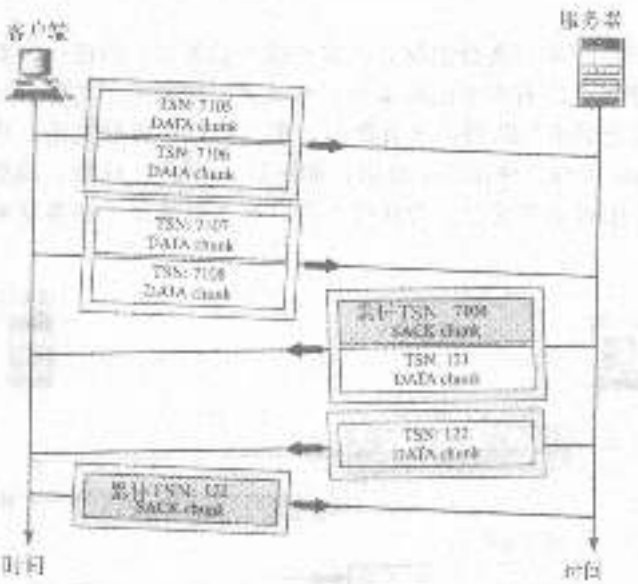


图23.34 简单数据传输

1. 客户端发送第一个分组，它携带TSN分别为7 105和7 106的两个DATA大块。

2. 客户端发送第二个分组，它携带TSN分别为7 107和7 108的两个DATA大块。

3. 第三个分组来自服务器，它包括了对收到来自客户端的DATA大块进行确认所需要的SACK大块。与TCP相反，SCTP只有按顺序收到的最后一个TSN进行确认，而不是下一个所期望的。第三个分组还包括了TSN为121的来自服务器的第一个DATA大块。

4. 不久，服务器发送另一个捎带TSN为122的最后一个DATA大块的分组，但由于接收来

自客户端的最后的DATA大块已被确认，所以在这个分组中不包括SACK大块。

5. 最后，客户端发送一个分组，该分组包含了一个SACK大块，该SACK大块是对接收到来自服务器最后两个DATA大块的确认。

在SCTP中的确认定义了累计TSN，最后数据大块的TSN是按序接收的。

多接口数据传输 我们讨论SCTP多接口的性能，它与UDP和TCP不同的特性。多接口允许双方为了通信定义多个IP地址。但是，这些地址中只有一个地址能被定义为主地址（primary address），其余的地址都是可选地址。在关联建立期间，定义主地址，主要的点是一个端的主地址是由另一端来确定的，换言之，源端定义目的端的主地址。

多流传递 SCTP的一个重要特性是数据传输和数据传递是不同的。SCTP使用TSN号处理数据传输，在源端与目的端之间移动数据大块。数据大块的传递是用SI和SSN来控制。SCTP支持多流，这就是说发送进程可定义不同的流，一个报文可属于这些流中某一个。每个流被赋于一个流标识符（SI），它唯一地定义这个流。

分段 在数据传输中的另一个问题是分段（fragmentation）。虽然SCTP也用IP中分段这个术语，但IP中的分段与SCTP中的分段属于不同的级别：前者是在网络层，而后者是在传输层。

如果一个报文（封装在IP数据报中）的长度不超过路径的MTU，当这个报文创建一个DATA大块的时，SCTP保存进程到进程的报文的边界。携带一个报文的IP数据报的长度等于报文的长度（以字节为单位）加上4个开销：数据大块的头部、必需的SACK大块、SCTP通用头部和IP头部。如果总长度超过MTU，则报文需要分段。

关联终止

像TCP一样，在SCTP中交换数据双方（客户端与服务器）的任一方都可关闭连接。但与TCP不同的是，SCTP不允许有半关闭的情形。如果其一端关闭了关联，另一端必须停止发送新的数据。如果在终止请求的队列中还有数据，那么将这些数据发送，并关闭关联。**关联终止**（association termination）使用三个分组，如图23.35所示。注意：虽然在图中终止是由客户端发起的，但也可由服务器发起。关联终止可以有多种情况，在本章末提到的参考文献中有讨论。

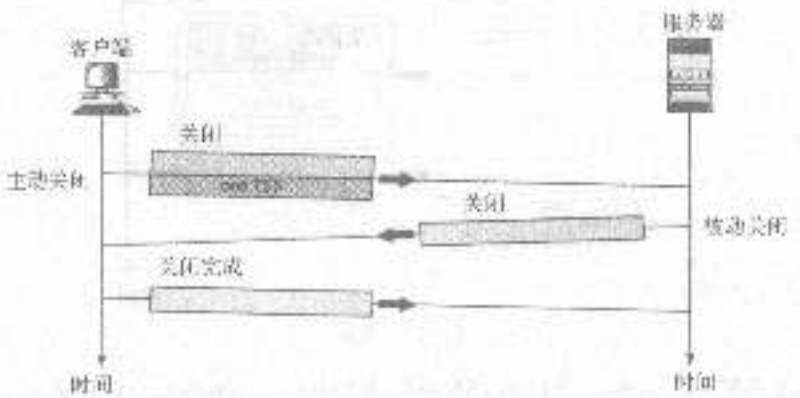


图23.35 关联终止

23.4.5 流量控制

SCTP的流量控制与TCP的流量控制相类似。在TCP中，我们需要处理仅一个数据单元，字节。在SCTP，我们需要处理两个数据单元的数据：字节和大块。*rwnd*和*cwnd*的值用字节表

示，TSN的值和确认用大块表示。为了说明这个概念，我们做某些不实际的假定，假设网络永远不会拥塞和永远没有差错，换言之，我们假定*cwnd*是无穷大，分组不丢失、不延迟、不失序到达，还假定数据传输是单向的。在后面几节我们将改进这些不实际的假定。关于流量控制，现在SCTP的实现依旧是使用一个面向字节的窗口。然而，我们为了使概念变成更加容易了解，用大块表示缓冲区。

接收方站点

接收方有一个缓冲区（队列）和三个变量。该队列保存接收到但还未被进程读出的大块。第一个变量保存最后收到的TSN值，即*cumTSN*。第二个变量保存缓冲区大小，即*winSize*。第三个变量保存最后的累计确认值，即*lastACK*。图23.36表示接收方站点的队列和变量。

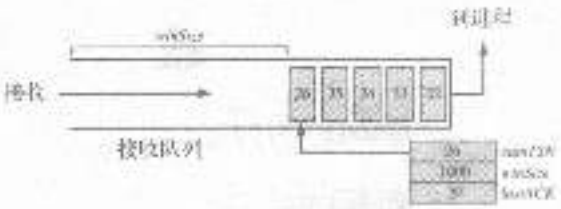


图23.36 接收方站点的流量控制

1. 当站点接收到一个数据大块时，将它存储在缓冲区（队列）的末端并从*winSize*减去该数据大块的大小。大块的TSN号存储在*cumTSN*变量中。

2. 当进程读出一个大块时，该块从队列中移去，并在*winSize*中加上移去大块的大小（循环使用）。

3. 当接收方决定发送一个SACK时，它检验*lastACK*的值，如果它小于*cumTSN*，接收方就发送一个具有累计TSN号等于*cumTSN*的SACK，同时该SACK也包含*wizeSize*的值作为该窗口的大小。

发送方站点

发送方有一个缓冲区（队列）和三个变量：*curTSN*、*rwnd*和*inTransit*。如图23.37所示。我们假定每个大块的长度为100个字节。

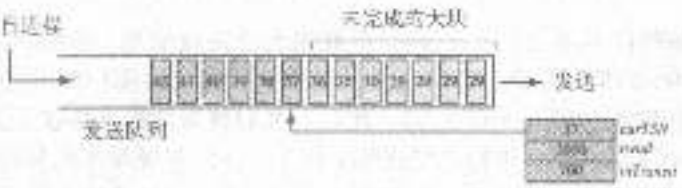


图23.37 发送方站点的流量控制

缓冲区保存由进程产生的大块，这些大块或者已被发送或者准备发送。第一个变量*curTSN*指的是下一个要发送的大块，队列中所有小于这个值的大块都已被发送，但还没有被确认，它们是未完成的。第二个变量*rwnd*保存接收方通告窗口的最新值（以字节为单位）。第三个变量*inTrasit*保存正在传输中的字节个数。即字节已发送但还未被确认。下面是发送方所用的过程。

1. 如果*curTSN*指向的大块的数据大小等于或小于*rwnd* − *inTransit*时，该大块可以发送。发送后，*curTSN*值增1并指向下一个要发送的大块，*inTransit*的值加上已被发送大块的数据大小。

2. 当接收到一个SACK，队列中具有TSN值小于或等于SACK中的TSN值的那些大块从队列中移去并丢弃。发送方不再关心它们。*inTrasit*值减去被丢弃块总的大小。用SACK中通告窗口的值更新*rwnd*的值。

一个实例

让我们给出如图23.38所示的一个简单实例。开始时，发送方站点的*rwnd*和接收方站点的*winSize*都是2 000（在关联建立期间通告）。发送方队列初始有4个报文，它发送一个数据大块并在*inTransit*变量中加上字节数（1 000）。之后，发送方检查*rwnd*与*inTransit*的差，它是1 000，所以发送方可发送另一个数据大块。现在两个变量的差为0，不能再发送数据大块。过一会儿，确认数据大块1和2的SACK到达。这两个大块从队列中移去，此时*inTransit*的值是0，但是，SACK通告了接收窗口的值是0，这使得发送方将*rwnd*值更新为0。现在发送方被阻塞，它不能发送任何数据大块（有一个例外稍后说明）。

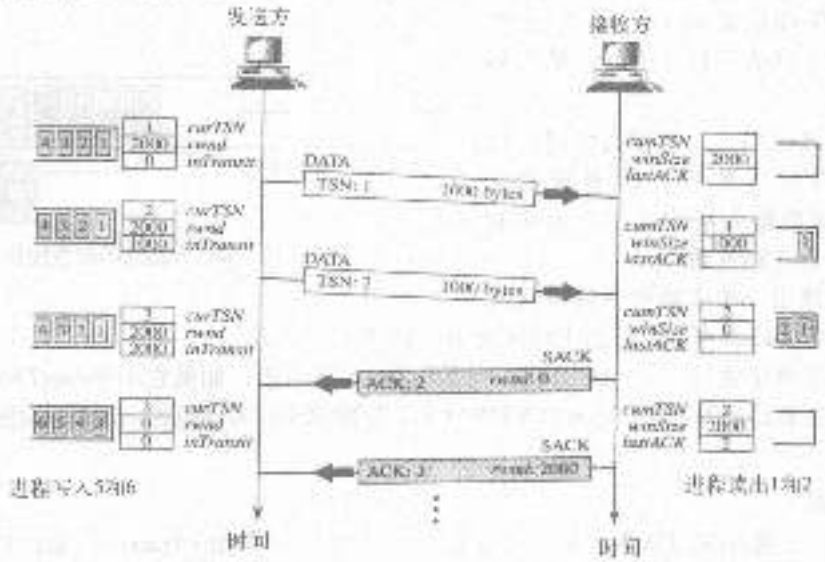


图23.38　流控制情况实例

在接收方，开始时队列是空的。当第一个数据大块接收到后，队列中有一个报文并且*cumTSN*值为1。*winSize*的值减小到1 000，这是因为第一个报文占用1 000个字节。接收到第二个报文后，窗口大小的值为0而*cumTSN*为2，此时，我们将看到要求接收方发送累计TSN为2的SACK。第一个SACK发送后，进程读出这两个报文，这就是说现在队列有空间；接收方用SACK通告这种情况允许发送方发送更多的数据大块。余下的事件在图中没有表示。

23.4.6　差错控制

像TCP一样，SCTP是一个可靠的传输层协议。它使用一个SACK大块向发送方报告接收方缓冲区的状态。在接收方站点与发送方站点每个实现都使用不同的实体和计时器集合，我们用一个最简单设计告诉读者这个概念。

接收方站点

在我们的设计中，接收方将所有到达的大块存储在队列中，包括失序大块。但是，它给有序的大块保留空间。它丢弃那些重复的报文，但追踪它们并向发送方报告。图23.39表示了一个接收方站点的典型设计和在一个特定时间接收队列的状态。

发送的最后确认是对数据大块20，有效窗口大小是1 000个字节。大块21到23已按序收到。第一个失序的块包含大块26到28，第二个失序块包含大块31到34。一个变量保存*cumTSN*的值。一组变量保存失序的那些块的首块TSN和末块TSN。一组变量保存接收到那些重复的大

块。注意：在队列中，不需要对重复的大块进行排序，它们被丢弃。图23.39中也显示了将要发送给发送方的接收方状态的SACK大块。那些失序大块的TSN号是相对于（偏移量）累计TSN的。

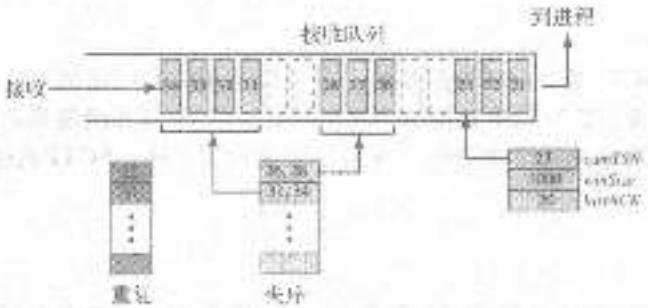


图23.39 接收方站点的差错控制

发送方站点

在发送方站点，我们的设计要求两个缓冲区（队列）：一个发送队列和一个重传队列。也使用前一节描述的三个变量*rwnd*、*inTransit*和*curTSN*。图23.40表示了一个典型的设计。

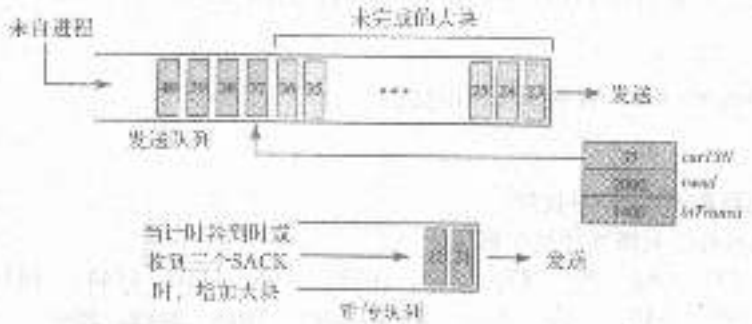


图23.40 发送方站点差错控制

发送队列保存大块23到40。其中大块23到36已发送，但还未被确认，它们是未完成的大块。*curTSN*指向下一个要发送的大块（37）。假定每个大块是100个字节，这就是说数据中的1 400个字节（大块23到36）是正在传输中的。这时发送方有一个重传队列，当发送一个分组时，那个分组（那个分组中的所有大块）的重传计时器开始启动。有些实现对整个关联使用一个计时器，但我们为了简单起见继续按惯例对每个分组使用一个计时器。当一个分组的重传计时器到时时，或声称一个分组丢失的4个重复的SACK（在第12章中讨论过快速重传）到达时，那个分组中的所有大块都移到重传队列中被重新发送。在重传队列中的大块具有优先权。换言之，下一次发送方发送的大块将来自重传队列的大块21。

发送数据大块

每当在发送队列中有那些具有TSN的值大于或等于*curTSN*的数据大块时，或者在重传队列中有数据大块时，一个端可发送一个数据大块。重传队列中的大块具有优先发送权。但是，一个分组所包含的数据大块的大小或多个数据大块总的大小必须小于或等于*rwnd* − *inTransit*。在前几节我们已讨论过，帧的总的大小必须小于或等于MTU。

重发 像TCP一样，为了控制丢失的或丢弃的大块，SCTP采用两种策略：使用重发计时器和接收到同一丢失大块的4个SACK。

生成SACK大块

差错控制中的另一个问题是生成SACK大块。生成SCTP SACK大块的规则与TCP中使用ACK标记进行确认所用规则相似。

23.4.7 拥塞控制

像TCP一样，SCTP是一个传输层协议，其分组会遭受到网络拥塞。SCTP设计者也使用相同的策略，我们将在第24章中讨论TCP拥塞控制。SCTP具有慢速启动（指数增加）、拥塞避免（渐加）和拥塞检测（倍增降低）等阶段。像TCP一样，SCTP也使用快速重传和快速恢复。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物和网站，在本书末列出方括号[……]中的参考资料。

读物

UDP在[For06]的第11章，[Ste94]的第11章和[Com00]的第12章中有讨论。TCP可参看[For06]的第12章，[Ste94]的第17到第24章和[Com00]的第13章。SCTP在[For06]的第13章和[SX02]中有讨论。UDP和TCP在[Tan03]的第6章中都有讨论。

网站

- www.ietf.org/rfc.html 有关请求评论信息。

请求评论

UDP的讨论可在RFC769中找到。

TCP的讨论可在下列请求评论中找到：

675，700，721，761，793，879，896，1078，1106，1110，1144，1145，1146，1263，1323，1337，1379，1644，1693，1901，1905，2001，2018，2488，2580

SCTP的讨论可在下列请求评论中找到：

2960，3257，3284，3285，3286，3309，3436，3554，3708，3758

本章小结

- 在客户服务器模型中，在本地主机上的称为客户机的应用程序，需要来自远程主机上的称为服务器的应用程序上的服务。
- 每个应用程序有一个端口号，可以将在同一个机器同时运行的程序区分开来。
- 给客户程序指定一个随机端口号，称为临时端口号。给服务器程序指定一个通用端口号，称为熟知端口号。
- ICANN已经指明了不同类型端口号的范围。
- IP地址与端口号的组合称为套接字地址，它定义了一个进程和一个主机。
- UDP是一个无连接不可靠的传输层协议，除了采用校验和进行差错检测外，它没有流量和差错控制机制。
- UDP分组称为用户数据报，一个用户数据报封装在IP数据报的数据字段中。
- 传输控制协议（TCP）是TCP/IP协议族中的一个传输层协议。
- TCP提供进程到进程、全双工和面向连接的服务。

- 采用TCP软件的两个设备之间传输的数据单元称为段。它有20到60个字节的头部，其后是来自应用程序的数据。
- 一个TCP连接通常包括三个阶段：连接建立、数据传输和连接终止。
- 连接建立要求三次握手，连接终止要求三次或四次握手。
- TCP使用滑动窗口机制进行流量控制避免接收方由于数据过多而崩溃。
- TCP的窗口大小由接收方通告窗口大小（*rwnd*）或拥塞窗口大小（*cwnd*）来确定，看哪一个较小。窗口可由接收方张开或合拢，但不能收缩。
- TCP在每一次连接中对传输的数据字节进行编号，初始号是随机产生的。
- TCP使用差错控制提供可靠服务。差错控制是由校验和、确认和超时来处理，受损坏的和丢失的段要重传，而重复的段应丢弃。数据可能失序到达，接收方的TCP将它们临时存储起来，但TCP保证传送给进程的是没有失序的段。
- 在现代实现中，如果重传计时器到时或三个重复ACK段已到达，则开始重发。
- SCTP是一个面向报文的、可靠的协议，它兼有UDP和TCP的特性。
- SCTP提供了UDP和TCP没有提供的服务，如多流和多接口服务。
- SCTP是一个面向连接的协议。一个SCTP连接被称为一个关联。
- SCTP使用的传输单元称为分组。
- 在SCTP中，控制信息与数据信息在不同的大块中携带。
- SCTP分组可以包含控制大块和数据大块，而控制大块是在数据大块之前。
- 在SCTP中，每个数据大块用传输序列号（TSN）来编号。
- 为了区别不同的流，SCTP使用序列标识符（SI）。
- 为了区别属于同一流的不同数据大块，SCTP使用流序列号（SSN）。
- 数据大块是用三个标识符来标识：TSN、SI和SSN。TSN是由整个关联识别的累计号，而SSN在每个流中以0开始。
- SCTP的确认号仅用来确认数据大块，如需要确认控制大块，则使用另一个控制大块。
- 通常用4个分组（四次握手）建立一个，SCTP关联使用3个分组（三次握手）终止一个关联。
- 一个SCTP关联使用cookie防止无法控制的洪泛攻击，使用确认标签防止插入攻击。
- SCTP提供流量控制、差错控制和拥塞控制。
- SCTP的确认SACK报告累计TSN、按序接收到的最后数据大块的TSN以及已接收到那些TSN。

习题

复习题

1. 在可靠性不是最重要的情况下，UDP可能是一个好的传输协议。试给出这种情况的一些例子。
2. UDP和IP的不可靠程度是否相同？为什么是或为什么不是？
3. 端口地址是否是唯一的？是或不是，为什么？为什么端口地址比IP地址短？
4. 在字典上*ephemeral*的定义是什么？它如何应用在临时端口号这个概念上？
5. UDP数据报的最小长度是多少？
6. UDP数据报的最大长度是多少？
7. 封装在UDP数据报中的进程数据的最小长度是多少？

8. 封装在UDP数据报中的进程数据的最大长度是多少？
9. 比较UDP与TCP的头部，试列出TCP头部中那些UDP头部中没有的字段，并说出它们没有的原因。
10. UDP是面向报文的协议，而TCP是面向字节的协议。如果应用需要保护它的报文边界，应该选用哪个协议，UDP还是TCP？
11. 当控制字段的值为下列的一个时，这个TCP段的含义是什么？

a. 000000　　b. 000001

c. 010001

12. TCP头部的最大长度是多少？TCP头部的最小长度是多少？

练习题

13. 一个UDP用户数据报携带了从TFTP客户到TFTP服务器的报文，试给出其头部的各项。试选择适当的临时端口号和正确的熟知端口号。数据的长度是40个字节。并用图23.9的格式表示UDP分组。
14. 驻留在IP地址为122.45.12.7主机中一个SNMP客户进程发送一个报文给驻留在IP地址为200.112.45.90主机中一个SNMP服务器进程。在这次通信中所使用的套接字地址是什么？
15. 驻留在IP地址为130.45.12.7主机中一个TFTP服务器进程发送一个报文给驻留在IP地址为14.90.90.33主机中一个TFTP客户进程。在这次通信中所使用的套接字地址是什么？
16. 一个客户有一个68 000字节的分组，试说明怎样使用UDP用户数据报将这个分组进行传输？
17. 一个客户使用UDP将数据发送给一个服务器，数据具有16个字节。试计算在UDP级的传输效率（有用字节与总字节之比）。
18. 重做练习17，但在IP级计算传输效率。假定IP头部无选项。
19. 重做练习18，但在数据链路级计算传输效率。假定IP头部无选项，数据链路层使用以太网。
20. 下面是打印出来的十六进制格式的UDP头部。

06 32 00 0D 00 1C E2 17

a. 源端口号是什么？　　b. 目的端口号是什么？

c. 用户数据报的总长度是多少？　　d. 数据的长度是多少？

e. 该分组是直接从客户发送到服务器还是从服务器发送到客户？　　f. 客户或进程是什么？

21. 一个IP数据报携带目的地址为130.14.16.17/16的TCP段。目的端口地址受到损坏，它到达的目的地址为130.14.16.19/16。接收方的TCP对这个差错如何反应？
22. 在TCP中，如果HLEN值是0111，试问这个段包含的选项有多少个字节？
23. 一个TCP报文携带了从FTP客户到FTP服务器的报文。试给出其头部的各项。将校验和字段填入0。试选择适当的临时端口号和正确的熟知端口号。数据长度为40个字节。
24. 下面是打印出来的十六进制格式的TCP头部。

05320017 00000001 00000000 500207FF 00000000

a. 源端口号是什么？　　b. 目的端口号是什么？

c. 序列号是什么？　　d. 确认号是什么？

e. 头部长度是多少？　　f. 段的类型是什么？

g. 窗口大小是多少？

25. 为了使初始序号是成为一个随机数，大多数的系统计数器从1开始每隔0.5秒自动引导增加64 000。试为计数器要环绕多久？

26. 在一个连接中，*cwnd*的值是3 000而rwnd的值是5 000。主机已发送2 000个字节，但它们还未确认。试问还有多少个字节可以发送？
27. TCP用初始序列号（ISN）14 534打开一个连接，而另一方用ISN为21 732打开该连接。试表示连接建立期间的这三个TCP段。
28. 一个客户使用TCP将数据发送给一个服务器，数据共有16个字节。试计算在TCP级的传输效率（有用字节与总字节之比）。计算在IP级的传输效率，假定IP头部无选项。计算在数据链路层传输的效率，假定IP头部无选项且在数据链路层使用以太网。
29. TCP发送数据的速率是每秒1M字节。如果序列号从7 000开始，试问经过多长时间序列号又回到0？
30. 一个TCP连接使用10 000字节的窗口大小，而上一次的确认号是22 011。它接收到一个确认号是22 001的段并且通告窗口大小为12 000。试用图说明窗口情况前后的变化情况。
31. 一个窗口保存了字节2 001到5 000。下一个要发送的字节是3 001。试用图说明下列两个事件之后窗口的情况：
    a. 接收到带有确认号为2 500的ACK段并且窗口通告大小为4 000；
    b. 携带1 000个字节的段被发送。
32. 在SCTP中，SACK的*cumTSN*值是23，SACK上次的*cumTSN*值是29，试问这是什么问题？
33. 在SCTP中，接收方的状态如下：
    a. 接收队列中有大块1到8、11到14以及16到20；b. 在队列中有1 800个字节为空；
    c. *lastACK*的值是4；d. 没有接收到重复的大块；
    e. *cumTSN*的值是5。
    试说明接收队列的内容及其变量。
34. 在SCTP中，发送方的状态如下：
    a. 发送队列中有大块18到23；b. *cumTSN*的值是20；
    c. 窗口大小值是2 000个字节；d. *inTransit*的值是200。
    如果每一个数据大块包含100个字节的数据，现在能有多少个DATA大块可发送？下一个DATA大块是什么？

探究题

35. 寻找有关ICANN更多的信息。在这名称之前它称做什么？
36. TCP用一个状态转换图处理段的发送与接收。画出该图并说明它如何处理流量与控制。
37. STCP用一个状态转换图处理段的发送与接收。画出该图并说明它如何处理流量与控制。
38. 在TCP中，半打开情形是怎样？
39. 在TCP中，半双工关闭情形是怎样？
40. 在UNIX或LINUX中，*tcpdump*命令可用来打印出一个网络接口的一些分组的头部信息。用*tcpdump*命令还可以查看已发送和接收的段。
41. 在STCP中，如果一个SACK大块被延迟或遗失，将会发生什么事？
42. 查明TCP中所用的计时器的名称和功能。
43. 查明STCP中所用的计时器的名称和功能。
44. 在SCTP中查找有关ECN更多的信息。查明这两个大块的格式。
45. 有些应用程序，如HTTP，当使用TCP时需要多个连接。查明SCTP多流服务如何帮助这些应用程序用多个流仅建立一个关联。

第24章 拥塞控制和服务质量

拥塞控制和服务质量是紧密联系在一起的两个问题：改进了其中的一个问题，则另一个问题也会有所改善；忽视了其中的一个问题，则通常意味着另一个也被忽视。在一个网络中，大多数防止或消除拥塞的技术也能改进网络的服务质量。

之所以推迟到现在才讨论这些问题，是因为这些问题不止涉及到一层，而是涉及到三层：数据链路层、网络层和传输层。而且现在才讨论它们也是为了能一次性地对这些问题做集中的探讨，而不需要多次讨论同一个主题。本章也给出了一些不同层中的拥塞控制和服务质量的示例。

24.1 数据通信量

对拥塞控制和服务质量关注的重要方面是数据通信量。在拥塞控制中，要设法避免通信量拥塞。在服务质量中，要设法为通信量创造合理的环境。因此，在讨论拥塞控制和服务质量之前，首先讨论数据通信量本身。

24.1.1 通信量描述符

通信量描述符是描述数据流的定性值。图24.1用这些值描述了通信量。

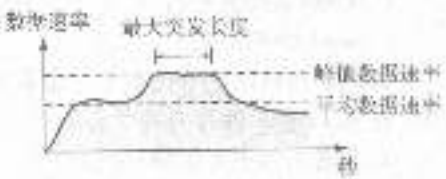


图24.1 通信量描述符

平均数据速率

平均数据速率（average data rate）是在一段时间内发送的数据位数除以该时段所含的秒数。我们使用如下的等式：

$$\text{平均数据速率} = \frac{\text{数据量}}{\text{时间}}$$

平均数据速率是通信量的一个非常有用的特性，因为它表明通信量所需要的平均带宽。

峰值数据速率

峰值数据速率（peak data rate）定义了通信量的最大数据速率，如图24.1所示，它是y轴方向上的最大值。峰值数据速率是一个很重要的概念，因为它表明让通信量通过网络而且无须改变数据流的情况下，网络所需的峰值带宽。

最大突发长度

虽然峰值数据速率是很关键的，但是如果峰值持续时间很短，那么它通常可忽略不计。例如，如果数据正以1Mbps的速率稳定地传送，而突然以2Mbps的峰值数据率传送了1ms，网络也许可以处理这种情况。但是如果峰值速率持续60ms，那么网络可能会出问题。最大突发长度（maximum burst size）一般是指以峰值速率传输通信量所持续时间的最大值。

有效带宽

有效带宽（effective bandwidth）是指网络需要分配给通信流的带宽。有效带宽是三个值的函数：平均数据速率、峰值数据速率和最大突发长度。有效带宽的计算过程非常复杂。

24.1.2 通信量特征值

根据需要，数据流可以用以下通信量特征值之一来表述：恒定比特率、可变比特率或突发性。如图24.2所示。

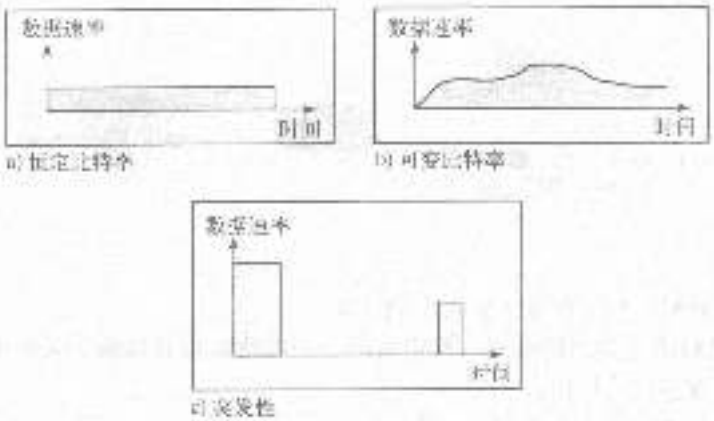


图24.2 三个通信量特征值

恒定比特率

恒定比特率（constant-bit-rate，CBR）或固定速率的通信量模型具有恒定不变的数据速率。在这种类型的流中，平均数据速率和峰值数据速率是相同的，最大突发长度是无效的。对于一个网络而言，这种类型的通信量是可预知的，也而很容易处理。网络可以提前知道分配多少带宽给这种类型的流。

可变比特率

在可变比特率（variable-bit-rate，VBR）类型中，数据流的速率随时间发生平滑的、而不是突然的或急剧的变化。在这种类型的流中，平均数据速率和峰值数据速率是不同的。最大突发长度通常是一个很小的值。与恒定比特率的通信量相比，这种类型的通信量更难处理，但是它一般不需要重新整形，这将在后面可以看到。

突发性数据

突发性数据（bursty data）是指在很短的时间内数据速率突然发生了变化。例如，它可能在几微秒内从零猛增到1Mbps，或从1Mbps猛降为零。它也可能在这个值上保持一会儿。在这种类型的数据流中，平均比特率和峰值比特率非常不同。最大突发长度非常重要。对于网络，这是最难处理的一种通信量类型，因为其特征值是不可预知的。为了处理这种类型的通信量，网络通常需要使用重新整形技术对其进行重新整形。这种技术稍后将会看到。突发性通信量是网络中发生拥塞的主要原因之一。

24.2 拥塞

拥塞（congestion）是分组交换网络中的一个重要问题。如果网络中的**载荷**（load），即发送到网络中的分组数量，超过了网络的容量，即网络中能处理的分组数量，那么在网络中就可能发生拥塞。**拥塞控制**（congestion control）指的是控制拥塞和使载荷低于网络容量的机制和技术。

读者可能会问，为什么在网络中会发生拥塞。拥塞可能发生在任何涉及等待机制的系统之中，例如，发生在高速公路上的拥塞是因为车流的任何异常，如在高峰期的事故，都可能

造成交通阻塞。

在网络或者互联网络中也会发生拥塞，这是因为路由器和交换机中存在队列——在处理前后保存分组的缓冲区。例如，路由器的每个接口中都有输入队列和输出队列。当分组到达输入接口时，在将其转发前要经历三个步骤，如图24.3所示。

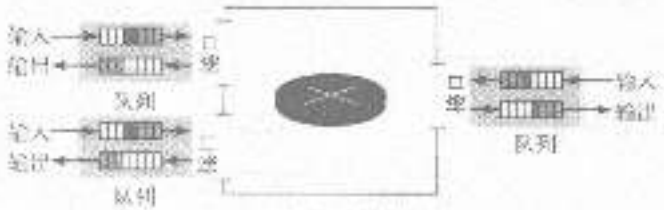


图24.3 路由器中的队列

1. 在等待检验时，将分组置于输入队列的尾部。

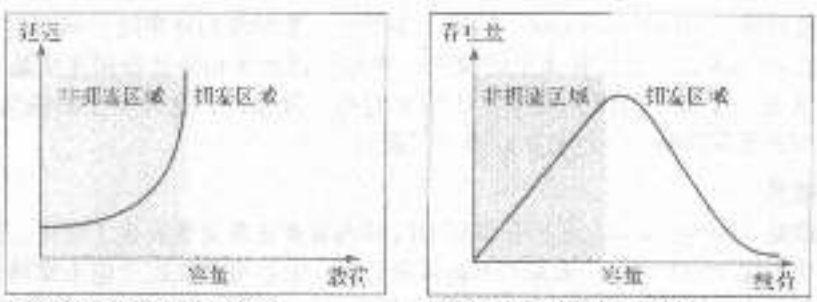
2. 一旦分组到达了队列的前端，路由器的处理模块就将其从输入队列中移除，并运用路由表和目的地址来查找其路由。

3. 将分组置于适当的输出队列中，等待发送。

这里需要注意两个问题。第一，如果分组的到达速率高于分组的处理速率，那么输入队列会变得越来越长。第二，如果分组的转发速率低于分组的处理速率，输出队列也会变得越来越长。

网络的性能

拥塞控制包括了两个测试网络性能的要素：延迟和吞吐量。图24.4说明了作为载荷函数的这两个性能度量。



a) 分组延迟是网络载荷的函数　　b) 吞吐量是网络载荷的函数

图24.4 分组延迟和吞吐量是网络载荷的函数

延迟和载荷

注意，当载荷比网络容量小得多时，延迟（delay）最小。最小延迟是由传播延迟和处理延迟所组成的，并且它们都可以忽略不计。然而，当载荷达到网络容量时，延迟就会急剧增加。因为现在需要将在队列（对路径中所有的路由器）中等待的时间加到总的延迟中。注意，当载荷大于网络容量时，延迟会变为无穷大。如果这还不能说明问题，那么就请考虑一下几乎没有分组到达目的端或经过无穷大的延迟到达目的端时队列的长度，此时队列变得越来越长。延迟会加重载荷，并因此会导致拥塞现象发生。当分组被延迟时，源端就会在没有收到确认的情况下重发分组，这个过程则会加重延迟和拥塞。

吞吐量和载荷

在第3章中，我们将吞吐量定义为在1秒内通过一个节点的位数。可以将位换成分组，将

节点换成网络，来扩展这个定义，从而将网络的吞吐量（throughput）定义为单位时间内通过网络的分组数量。注意，当载荷小于网络容量时，吞吐量随载荷的增加成比例地增长。当载荷达到网络容量后，期望吞吐量保持恒定，但是实际情况恰好相反，吞吐量会急剧下降。原因是路由器丢弃了分组。当载荷超过网络容量时，队列饱和了，此时路由器不得不丢弃一些分组。丢弃分组并不能减少网络中分组的数量，因为当分组没有到达目的端时，源端就运用超时机制重发它们。

24.3 拥塞控制

拥塞控制是指在拥塞发生之前预防拥塞，或在拥塞发生之后消除拥塞的技术和机制。通常，我们把拥塞控制分为两大类：开环拥塞控制（预防）和闭环拥塞控制（消除），如图24.5所示。

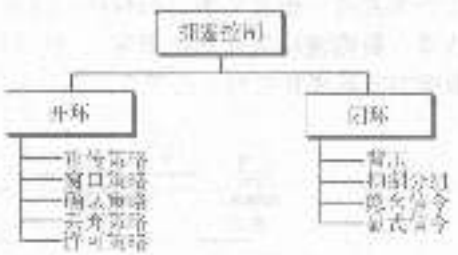


图24.5 拥塞控制分类

24.3.1 开环拥塞控制

在**开环拥塞控制**（open-loop congestion control）中，在拥塞发生之前，应用某种策略来预防拥塞现象的发生。在这些机制中，源端或目的端都可以处理拥塞控制。下面简要列出能预防拥塞的几种策略。

重传策略

有时重发是不可避免的。如果发送方认为一个发送分组丢失或损坏，则该分组就需要重发。重发一般在网络上会增加拥塞现象。然而一个好的重传策略能预防拥塞。必须优化设计重传策略和重传定时器，使之具有高效率，并用来预防拥塞。例如，TCP所用的重传策略（稍后说明）的设计是防止和减轻拥塞。

窗口策略

发送方窗口的类型也会影响拥塞。对拥塞控制而言，选择性重复窗口要优于回退N帧窗口。在回退N帧窗口中，当一个分组的计时器到时时，可以重发多个分组。虽然有一些可安全与可靠地到达接收方，但这种重复会使拥塞更加严重。而选择性重复窗口试图发送那些被丢失或损坏的特定分组。

确认策略

接收方使用的确认策略也可能影响拥塞。如果接收方并不对它所接收的每一个分组进行确认，则它会使发送方放慢发送速度，从而有助于预防拥塞。有些方法是采用这种办法。如果接收方有一个要发送的分组或一个特定的计时器到时时，接收方才发送一个确认。接收端可以决定每次对N个分组仅发送一个确认。我们要知道确认也是网络中负载的一个部分，发送确认越少意味着网络的负载越轻。

丢弃策略

路由器使用好的丢弃策略可以预防拥塞，同时不破坏传输的完整性。例如，在声音传输中，如果在可能发生拥塞时丢弃那些不敏感的分组，则仍可保证声音的质量，并且还能预防或减轻拥塞现象的发生。

许可策略

在虚电路网络中，许可策略是一种服务质量机制，也能预防拥塞。在允许数据流进入网

络之前，与数据流通路相连的交换机首先检查其资源需求。如果网络有拥塞或可能出现拥塞，路由器将拒绝建立虚电路。

24.3.2　闭环拥塞控制

在拥塞发生之后，采用闭环拥塞控制（closed-loop congestion control）可以缓解拥塞状况。不同的协议采用了多种机制，下面对这几种机制进行介绍。

背压

背压技术是一种拥塞控制机制，在这种技术中，一个拥塞点停止接收来自直接上行节点或一些近邻节点的数据。这会引起上行节点或一些近邻节点发生拥塞，它们依此拒绝它们的上行节点或一些近邻节点的数据，依此类推。背压是点到点拥塞控制，它从一点开始，然后传播，数据流反方向到达源端。背压技术仅用于虚电路网络，在虚电路网络中，每个节知道数据的流是来自它的上行节点，图24.6表示了背压的思想。

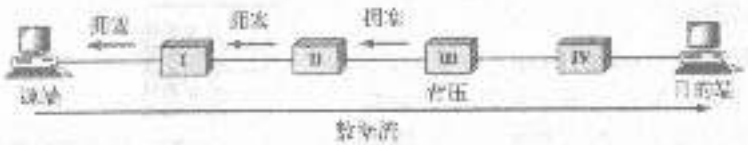


图24.6　逐跳拥塞背压方法

在图24.6中，节点III有了超出它处理能力的输入数据，它停止将某些分组放入输入缓存区中，并通知节点II降低传输速率。依次，节点II由于降低输出数据流的速率可能会拥塞，如果节点II拥塞，则它通知节点I降低传输速率，它也可能会形成拥塞。如果是如此，则节点I通知源端降低数据传输速率。这样可以及时减轻拥塞。注意：为了消除阻塞，对节点III的压力反向移动到源端。

本书讨论的一些虚电路网络不使用背压策略，但是，在第一个虚电路网络X.25，使用过这策略。这种技术在数据报网络不可能实现，因为在这种类型的网络中，一个节点（路由器）没有关于上行节点的知识。

抑制分组

抑制分组（choke packet）是一个分组，该分组由节点发送给源端，通知它发生拥塞的情况。注意背压和抑制分组方法的不同，在背压方法中，警告从一个节点到它的上行节点，虽然警告可能最后到达源端，而在抑制分组方法中，警告从已经发生拥塞的路由器直接传到源端，该分组经过的那些中间的节点没被警告，我们在ICMP中已看到过这种控制类型。当因特网中一个路由器被大量的IP数据报流浸时，它可能丢弃一些数据报。但它使用一个ICMP源站抑制ICMP报文通告源主机，警告报文直接发送给源站点，中间的路由器不进行任何处理，图24.7表示抑制分组的思想。

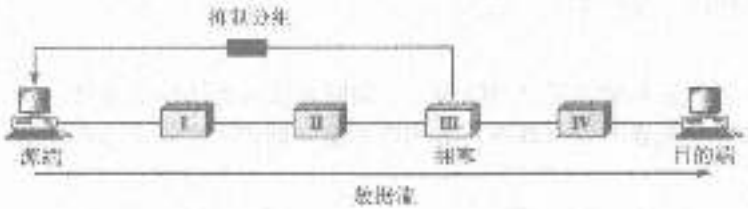


图24.7　抑制分组

隐含信令

在隐含信令中，拥塞节点或节点与源端之间没有通信。源端能从其他有关征兆中察觉出在网络某处有拥塞。例如，当源端发送多个分组，暂时还没有收到确认时，一种设想是网络发生了拥塞。在接收确认过程中发生的延迟现象就可以认为网络发生了拥塞，源端应该降低发送速率。在本章后面讨论TCP拥塞控制时，将看到这种类型的信令。

显式信令

发生拥塞的节点能发送一种显式信令通知源端或目的端发生了拥塞。但是，显式信令方法与抑制分组方法是不同的。在抑制分组方法中，有一个单独的分组用于此目的；而在显式信令方法中，信号包含在携带数据的分组中。就像在帧中继拥塞控制中所看到的那样，显式信令也具有前向或后向特征。

后向信令：将一个位设置在分组中，并使之向与拥塞发生方向相反的方向移动。该位提示源端网络发生了拥塞，并提示它需要放慢发送速度，以避免分组的丢失。

前向信令：将一个位设置在分组中，并使之向发生拥塞的方向移动。该位提示目的端网络发生了拥塞。在此情况下，接收端能使用诸如放慢确认发送速度的策略来减轻拥塞。

24.4 两个示例

为了更好地理解拥塞控制的概念，我们举两个例子：一个是TCP中的拥塞控制，另一个是帧中继中的拥塞控制。

24.4.1 TCP中的拥塞控制

我们在第23章中已讨论过TCP，现在我们说明TCP如何利用拥塞控制避免或减轻网络中的拥塞。

拥塞窗口

在第23章中，我们曾提到过流量控制，并试图讨论应如何解决接收方由于大量数据的到来而陷入瘫痪的情况。我们讲过，发送方窗口大小取决于接收方可用的缓冲空间（*rwnd*）。换言之，我们假设只有接收方能够规定发送方的发送方窗口大小。在这里我们完全忽视了另一个实体，即网络。如果网络不能够像发送方产生数据那样快传递数据，它就应当告诉发送方减慢发送速率。也就是说，除了接收方之外，网络应当是确定发送方窗口大小的第二个实体。

现在，发送方窗口大小不仅取决于接收方，而且还取决于网络拥塞的情况。

发送方有两种信息：接收方通告的窗口大小和拥塞窗口大小。实际的窗口大小是这两者中的最小者。

```
实际的窗口大小 = min (rwnd, cwnd)。
```

我们简要说明如何确定拥塞窗口大小（*cwnd*）。

拥塞策略

TCP处理拥塞的一般策略基于三个阶段：慢速启动，拥塞避免和拥塞检测。在慢速启动阶段，发送方用很慢的传输速率启动，但迅速地增加到阈值。在达到阈值时，为了避免拥塞而降低数据速率。最后，如果检测到拥塞，则发送方基于如何检测拥塞而返回到慢速启动或拥塞避免阶段。

慢速启动：指数增长 TCP拥塞控制所使用的一种算法称为慢速启动（slow start）。这种算法是基于这样的想法，它在开始时设置拥塞窗口大小（*cwnd*）为一个最大段长度（MSS）。

这个MSS是连接建立期间由最大段长度选项所决定的。每次接收到一个确认时，窗口大小增加一个MSS值。正如其名的含义，窗口是慢速启动，但是按指数规则增长的。为了说明这个思想，让我们观察图24.8。注意：为了使讨论更清晰，我们进行了三个简化。使用段的个数而不是字节的个数（好像每段仅有一个字节），假定*rwnd*比*cwnd*大得多，这样发送方窗口大小永远等于*cwnd*。还假定每段都是单独进行确认。

发送方以*cwnd*＝1MSS开始。这意味着发送方仅能发送一个段，接收到段1的确认后，拥塞窗口增加1，*cwnd*是2。这意味着现在可发送另外两个段。当接收到每一个确认时，窗口大小都增加1MSS。当所有7个段都被确认时，*cwnd*＝8。

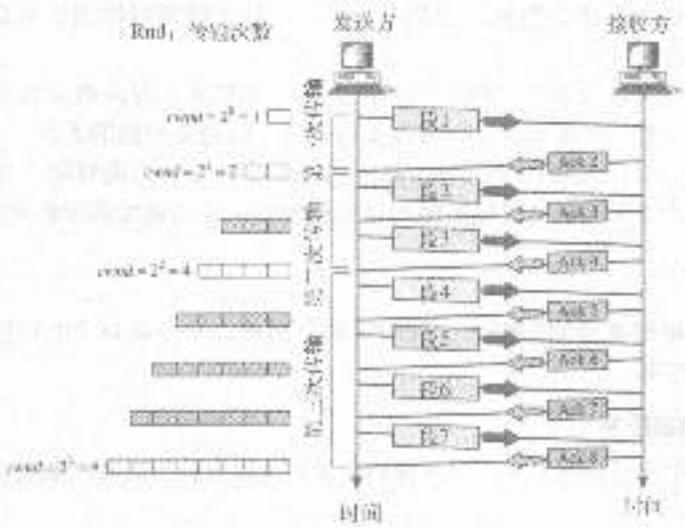


图24.8　慢速启动：指数增加

如果我们按照传输次数观察*cwnd*的大小（整个窗口中的段被确认），则发现其速率是按指数规律增长如下所示：

开始　⇒　$cwnd = 1$
第一次传输后　⇒　$cwnd = 2^1 = 2$
第二次传输后　⇒　$cwnd = 2^2 = 4$
第三次传输后　⇒　$cwnd = 2^3 = 8$

我们需要指出的是，如果有被延迟的ACK，则窗口大小的增长是小于2的幂。

慢速启动不能一直继续下去，到达阈值必须停止该阶段。发送方保存一个称为*ssthresh*（慢速启动阈值）变量。当拥塞窗口中的字节达到这个阈值时，慢速启动阶段结束而下一个阶段开始。在大多数实现中，*ssthresh*值是65 535字节。

> 在慢速启动算法中，拥塞窗口大小按指数规律增长直到达到阈值。

拥塞避免：加性增加　如果我们以慢速启动算法开始，则拥塞窗口大小按指数规律增长，为了在拥塞发生之前避免拥塞，必须降低指数增长的速度。TCP定义了另一个算法，称为**拥塞避免**（congestion avoidance），这个算法是加性增加（additive increase），而不是指数增加。当拥塞窗口的大小达到慢速启动的阈值时，慢速启动阶段停止，加性增加阶段开始。在这个算法中，每次将个窗口所有段都被确认（一次传输）时，拥塞窗口才增加1。为了说明这个概念，我们将这个算法应用到与慢速启动相同的情况中，不过我们将会见到的拥塞避免算法通

常开始时窗口大小是大于1的。图24.9表示这个思想。

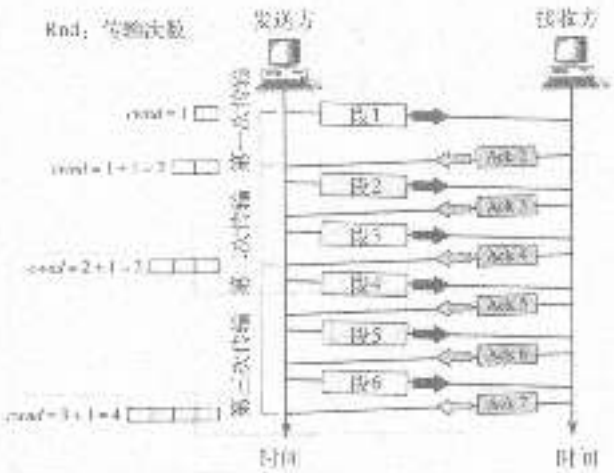


图24.9 拥塞避免：加性增加

在这种情况下，发送方接收到对一个完整窗口大小段的那些确认后，窗口大小才增加一个段。

如果我们按照传输次数观察*cwnd*的大小，则发现其速率是按加性规律增长，如下所示：

开始 ⇒ $cwnd = 1$

第一次传输后 ⇒ $cwnd = 1 + 1 = 2$

第二次传输后 ⇒ $cwnd = 2 + 1 = 3$

第三次传输后 ⇒ $cwnd = 3 + 1 = 4$

> 在拥塞避免算法中，拥塞窗口大小是加性增加的直到检测到拥塞。

拥塞检测：乘性减少 如果发生拥塞，拥塞窗口的大小必须减小。发送方能推测出发生拥塞现象的唯一方法是通过重传段的要求。可是重传是在两种情况下发生：重传计时器到时或接收到了三个ACK。在这两种情况下，阈值就下降一半，即乘性减少（multiplicative decrease）。大多数TCP实现包含两个反应：

1. 如果计时器到时，那么存在着非常严重的拥塞的可能性，一个段可能已在网络中丢失并且没有关于该发送段的消息。

在这种情况下，TCP做出强烈的反应：

a. 设置阈值为当前拥塞窗口大小的一半；

b. 设置*cwnd*为一个段的大小；

c. 启动慢速启动阶段。

2. 如果接收到三个ACK，那么存在着轻度拥塞的可能性，一个段可能已丢失，但自从接收到三个ACK后，有一些段可能已安全到达。这称为快速传送和快速恢复。

在这种情况下，TCP做出轻度的反应：

a. 设置阈值为当前拥塞窗口大小的一半；

b. 设置*cwnd*为阈值（有些实现是阈值加上三个段）；

c. 启动拥塞避免阶段。

用下列方法之一对拥塞检测做出反应：
- 如果检测出计时器到时，那么一个新的慢速启动阶段开始。
- 如果检测出三个ACK，那么一个新的拥塞避免阶段开始。

总结　在图24.10中，概括了TCP的拥塞策略与三个阶段之间的关系。

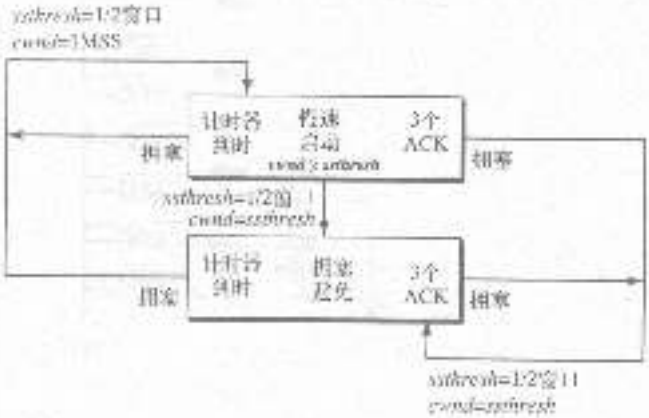


图24.10　TCP拥塞策略总结

我们在图24.11中给出了一个例子。假定最大窗口大小是32个段，阈值设置为16个段（最大窗口的一半），在慢速启动阶段，窗口大小从1开始按指数规律增长直到它达到阈值。当它达到阈值后，拥塞避免（加性增加）过程允许窗口大小线性增长直到计时器到时或到达最大窗口大小。在图24.11中，当窗口大小为20时，计时器到时，此时，进入乘性减少过程，将阈值设置为当前窗口大小的一半。当计时器到时时，当前窗口大小是20，因此现在阈值是10。

TCP再次进入慢速启动，并置窗口大小为1。当到达新的阈值时，TCP进入加性增加阶段。当窗口大小为12时，三个ACK事件发生。再次进入乘性减少过程，阈值设置为6，这时TCP进入加性增加阶段，该阶段一直维持到另一个计时器到时或者另外三个ACK事件发生为止。

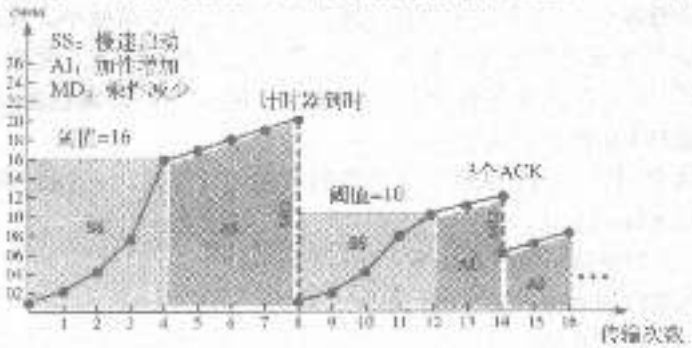


图24.11　拥塞例子

24.4.2　帧中继中的拥塞控制

帧中继网络中的拥塞会导致吞吐量降低和延迟的增加。高吞吐量和低延迟是帧中继协议的主要目标。帧中继协议没有流量控制机制。另外，帧中继允许用户传送突发性数据。这意味着帧中继网络有可能因为通信量过大而发生拥塞现象，因此需要有拥塞控制机制。

避免拥塞

为了避免拥塞，帧中继协议在帧中利用2个位来明确地提示源端和目的端拥塞的发生。

BECN　后向显式拥塞通知（backward explicit congestion notification，BECN）位提示发

送方网络中的拥塞情况。因为帧是从发送方传送出来的，有人可能问这是如何实现的。实际上有两种方法：交换机利用来自接收方（全双工模式）的响应帧，或者为了特定的目的，交换机可使用一个预定义连接（DLCI=1023）来发送专门的帧。发送方仅通过减小数据速率对该提示信息做出响应。图24.12说明了运用BECN的情况。

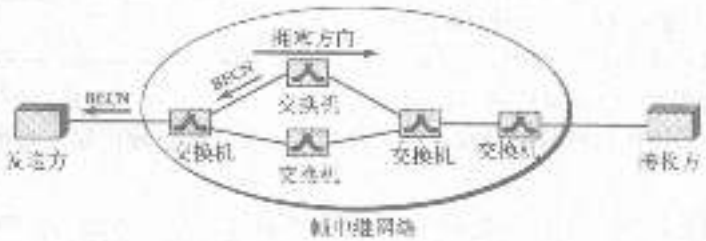


图24.12 BECN

FECN 前向显式拥塞通知（forward explicit congestion notification，FECN）位是用来提示接收方网络中拥塞的情况。有可能会出现接收方不能采取任何措施来减轻拥塞的情况。但是，帧中继协议假定发送方和接收方能相互通信并在较高层使用某种类型的流量控制机制。例如，如果在较高层有确认机制，接收方就能延迟发送确认。这样就可以迫使发送方放慢发送速度。图24.13说明了运用FECN的情况。

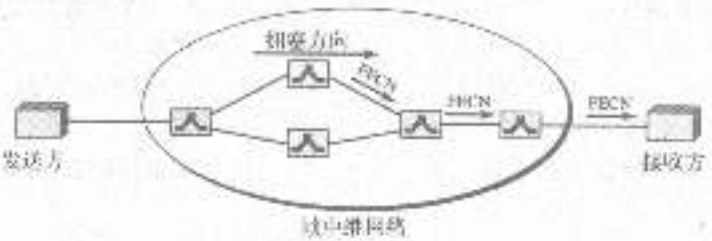


图24.13 FECN

当两个端点运用帧中继网络进行通信时，可能会出现四种与拥塞有关的情况。图24.14说明了这四种情况以及FECN和BECN的值。

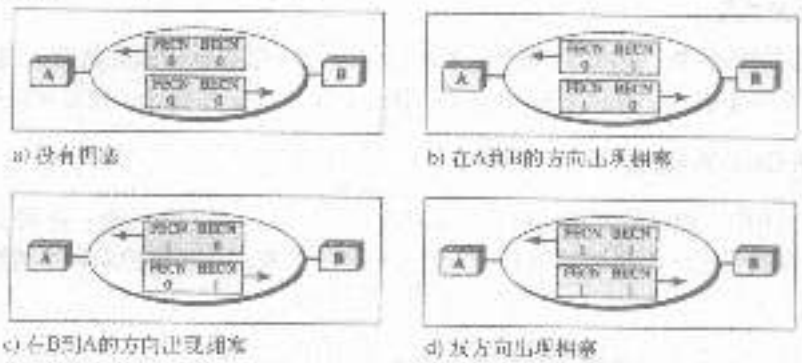


图24.14 拥塞的四种情况

24.5 服务质量

服务质量（quality of service，QoS）是一个网络互联问题，对该问题的讨论已经远远超出对它的定义。我们可以非形式地将服务质量定义为数据流所追求的某种目标。

24.5.1 数据流特性

按传统的观点，数据流具有四种特性：可靠性、延迟、抖动和带宽，如图24.15所示。

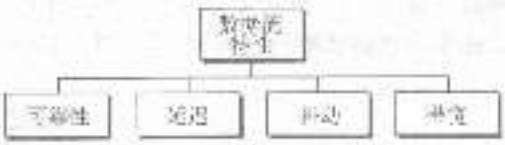


图24.15　数据流特性

可靠性

可靠性（reliability）是数据流所需要的一个特性。缺乏可靠性意味着分组或确认的丢失，这些都必须通过重传来弥补。然而，应用层程序对可靠性的敏感程度是不同的，例如，电子邮件、文件传输和因特网访问与电话或音频会议相比，保证其可靠传输更为重要。

延迟

源端到目的端的延迟（delay）是另一个数据流特性。此外，不同的应用对延迟的容许程度是不同的。在此情况下，电话、音频会议、视频会议和远程登录都需要最小的延迟，但是在文件传输或电子邮件中，延迟并不那么重要。

抖动

抖动（jitter）是属于同一数据流的分组延迟的变化。例如，如果4个分组离开的时间分别是0，1，2，3，而到达的时间分别是20，21，22，23，则它们都有相同的延迟，为20。另一方面，如果上述4个分组的到达时间分别是21，23，21，28，则它们延迟是不同的：21，22，19，24。对于音频和视频的应用，前者是完全可接受的，而后者是不能接受的。只要延迟对所有的分组都是相同的，分组以短或者长的延迟到达，无关紧要。对于这个应用，后者的情形是不可接受的。抖动被定义为分组延迟的变化。较高的抖动意味着在延迟之间的差异较大，而低的抖动意味着差异很小。

在第29章中，我们将会看到多媒体通信如何处理抖动。如果抖动高，那么为了使用接收到的数据，就需要执行一些动作。

带宽

不同的应用需要不同的带宽。在视频会议中需要发送数百万比特率来刷新彩色屏幕，但是在一封电子邮件中的全部位的个数甚至都不到一百万。

24.5.2 数据流类型

基于数据流的特性，将数据流分成不同的组，每个组都有相似的特性。这种分类方法既不够规范也不够普遍。一些诸如ATM之类的协议定义了类型的概念，稍后可以看到这一点。

24.6 改进QoS的技术

在第24.5节中，根据数据流的特性对QoS进行了定义。本节将讨论一些用于改进服务质量的方法。这里简要地讨论两种常用方法：调度、通信量修形、许可控制和资源预留。

24.6.1 调度

来自不同数据流的分组到达交换机或路由器，并由它进行处理。一种好的调度技术会以公平合理的方式来对待不同的数据流。已经设计了多种调度技术用来改进服务质量，在这里讨论其中的三种技术：FIFO队列、优先权队列和加权公平队列。

先进先出队列

在先进先出（first-in，first-out，FIFO）队列中，分组在缓冲区（队列）中等待，直到节

点（路由器或交换机）准备处理它们为止。如果平均到达速率高于平均处理速率，那么队列将被填满，新的分组将被丢弃。FIFO队列和那些必须在公共汽车站等待汽车的人类似。图24.16说明了FIFO队列的概念。

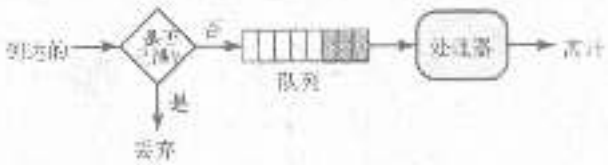


图24.16 FIFO队列

优先权队列

在优先权队列（priority queuing）中，首先给不同的分组分配一个不同的优先权类。每个优先权类都有自己的队列。在最高优先权队列中的分组首先得到处理。而最后才对最低优先权队列中的分组进行处理。注意：系统不停地为一个队列提供服务，直到它变空为止。图24.17说明了具有两个优先权级别的优先权队列（为了简单起见）。

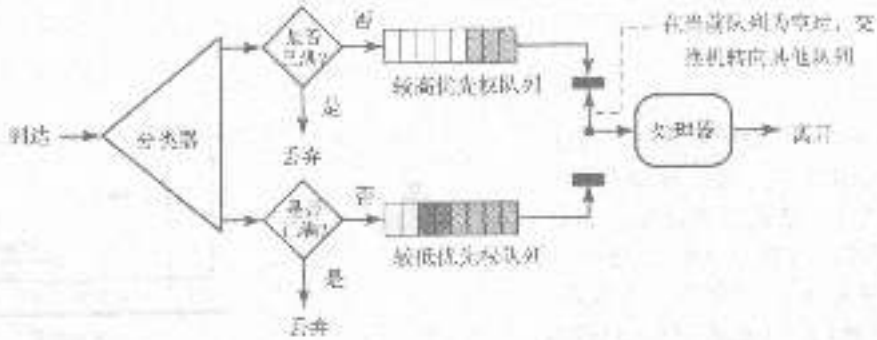


图24.17 优先权队列

与FIFO队列相比，优先权队列能提供更好的QoS，因为具有较高优先权的通信量，如多媒体，能用较少的延迟到达目的端。但是，优先权队列也有潜在的缺点，如果在高优先权队列中有持续的通信量，那么处于较低优先权队列中的分组将永远得不到处理。这种情形称为"饥饿"。

加权公平队列

较好的调度算法是加权公平队列（weighted fair queuing）。在这种技术中，分组仍然被分成不同的类，并且属于不同的队列。然而，队列是基于队列的优先权来分配权重的，较高的优先权就意味着具有较高的权重，系统以轮换的方式来处理每个队列中的分组，所处理的分组的数量等于相应队列的权重。例如，如果这些权重是3、2和1，那么就对第一个队列中的3个分组、第二个队列中的2个分组和第三个队列中的1个分组进行处理。如果系统并没有给不同类指定不同的权重，那么所有的权重就是相等的。这就是加权公平队列的原理。图24.18以三个类为例说明了这种技术。

24.6.2 通信量整形

通信量整形（traffic shaping）是一种控制发送到网络中的通信量和速率的机制。通信量整形有两种技术：漏桶和令牌桶。

漏桶

如果桶在底部有一个小洞，只要桶中有水，水便从桶中以不变的速率漏下。如果桶中有

水，水漏的速率并不依赖于将水倒入桶中的速率。输入速率可以发生变化，但是输出速率保持恒定。同样，在网络中一种被称为漏桶（leaky bucket）的技术能消除突发性通信量。将突发性大块数据存储到桶中，然后以平均速率发送出去。图24.19说明了漏桶和其效果。

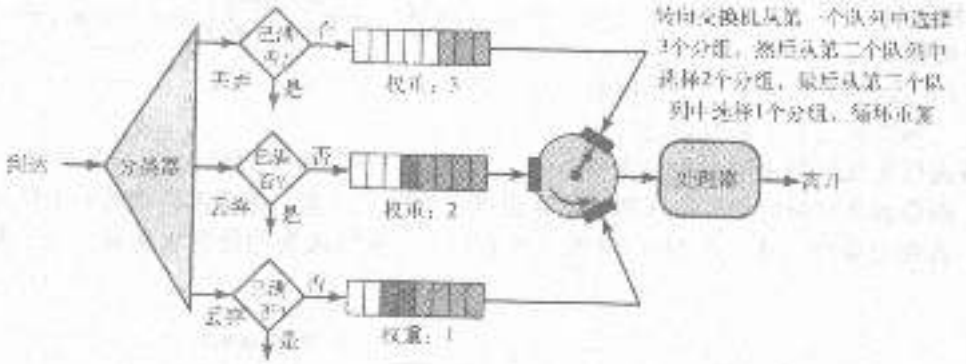


图24.18　加权公平队列

在图24.19中，假设网络为每台主机都设定了3Mbps的带宽。使用漏桶对输入通信量进行整形，是为了使通信量符合这个设定的带宽。在图24.19中，主机以12Mbps的速率发送了2秒的突发性数据，数据一共是24兆位，主机在之后的5秒内没有任何反应，然后以2Mbps的速率发送数据，持续时间为3秒，此时的全部数据为6兆位，主机在10秒内总共发送了30兆位的数据。在这同样的10秒内，漏桶以3Mbps的速率发送数据来使通信量保持平稳。如果没有漏桶，因为该主机占用了比分配给它的更大的带宽，它开始所发送的突发性数据可能已经导致网络拥塞。而且，漏桶还可以避免拥塞。做个类比，以高峰期间（突发性通信量）的高速公路为例，如果往返者们能错开他们的工作时间，那么高速公路上的拥塞状况就可以得到避免。

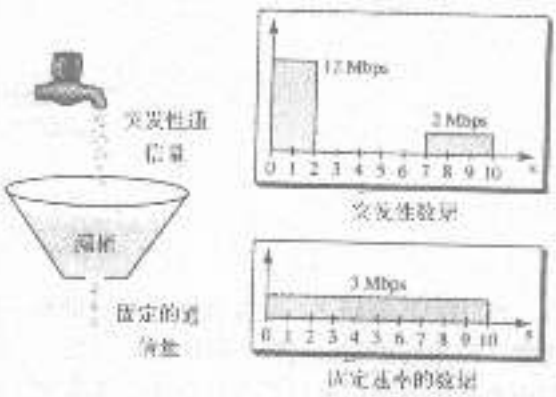


图24.19　漏桶

简单的漏桶实现过程如图24.20所示。FIFO队列保存了分组。如果这个通信量由固定大小的分组（例如，ATM网中的信元）组成，那么在每个时钟单位时间内，进程从队列中移除固定数量的分组。如果通信量是由变长分组组成的，那么固定输出速率必须是基于字节个数或位个数。

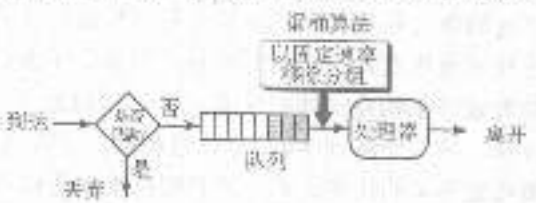


图24.20　漏桶的实现过程

以下是变长分组的算法：

1. 在计时开始时，将计数器初始化为n；

2. 如果n比分组的长度大，就发送分组，并将计数器的值减去分组的长度，重复该步骤，直到n值小于分组的长度；

3. 重新设置计数器，并返回到步骤1。

漏桶算法通过均分数据速率将突发性流信量整形为固定速率的通信量，如果桶已满，就可能丢失分组。

令牌桶

漏桶算法有很大的局限性，它不能给空闲的主机提供信用。例如，如果一台主机在一段时间内没有发送数据，其漏桶就变为空桶了。现在即使主机中有突发性数据，漏桶也只能允许平均速率的数据。此时并没有考虑主机空闲时间。另一方面，令牌桶（token bucket）算法允许空闲主机以令牌的形式为未来累积信用。在每个时钟单位时间内，系统发送n个令牌到桶中。在每个数据信元（或字节）发送后，系统就从中去掉一个令牌。例如，假设n是100，主机空闲了100个时钟单位时间，那么令牌桶便收集了10 000个令牌。主机可以在一个时钟单位时间消费所有这些令牌来发送10 000个信元，或者花去1 000个时钟单位时间，每个时钟单位时间内发送10个信元。换句话说，只要令牌桶不是空的，主机就能发送突发性数据。图24.21说明了该算法的原理。

使用计数器可以很容易地实现令牌桶算法。令牌初始化为0，每次增加1个令牌，计数器就加1，每次发送一个数据单元，计数器就减1。当计数器为0时，主机不能发送数据。

令牌桶允许规定的最大速率发送突发性通信量。

令牌桶和漏桶的结合使用

可将两种技术结合起来给空闲主机提供信用，同时调整通信量。在令牌桶之后应用漏桶，并且漏桶的速率应比将令牌放入桶中的速率高。

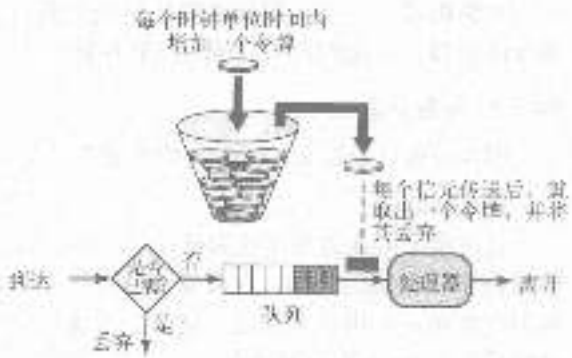


图24.21 令牌桶算法的原理

24.6.3 资源预留

数据流需要诸如缓冲区、带宽、CPU时间等资源。如果提前对这些资源进行预留，将会提高服务质量。我们在这一节将讨论一个称为综合业务的QoS模型，该模型极大地依赖资源预留来提高服务质量。

24.6.4 许可控制

许可控制指的是由路由器或交换机使用的一种机制，它基于一个称为数据流规范的预定义参数来接收或拒绝数据流。在路由器接收数据流并对其进行处理之前，它检查数据流规范，查看其性能（根据带宽、缓冲区大小及CPU速度等）和它与其他数据流先前的约定能否处理新的数据流。

24.7 综合业务

根据第24.5节和第24.6节的讨论，在互特网中已经设计出了两种用于提供服务质量的模型：综合业务和差分业务。两个模型都强调在网络层（IP）使用服务质量，尽管这些模型也可以用于诸如数据链路层的其他层。本节讨论综合业务，第24.8节讨论差分业务方面的内容。正如在第20章中所了解到的，IP最初是为尽力传递而设计的。这意味着每个用户接受同样级别的服务。这种类型的传递不能保证一项业务的最低要求，例如实时音频和视频应用所需的带宽。即使这样的应用偶然获得了额外的带宽，也可能对其他应用造成不利影响，并导致拥塞

现象发生。

综合业务（integrated service）有时称为IntServ，它是基于数据流的QoS模型，这意味着用户需要创建一条从源端到目的端的流，即一种虚电路，并将资源需求通知给所有的路由器。

综合业务是设计用于IP的基于数据流的QoS模型。

24.7.1 信令

读者可能记得，IP是一种无连接的、数据报、分组交换协议。如何在一种无连接的协议上实现一个基于数据流的模型呢？解决办法是在IP上使用信令协议，它为资源预留提供信令机制。这个协议称为资源预留协议（Resource Reservation Protocol，RSVP），稍后会讨论到它。

24.7.2 数据流规范

当源端做资源预留时，需要定义数据流规范。数据流规范包括两个部分：Rspec（资源规范）和Tspec（通信量规范）。Rspec定义数据流需要预留的资源（如缓冲区、带宽等）。Tspec定义数据流的通信量特征。

24.7.3 许可

在路由器从一个应用接收到数据流规范后，它决定许可或拒绝这个业务。所做出的决定基于路由器的先前约定和资源的当前有效性。

24.7.4 业务类型

综合业务已经定义了两种类型的业务：保证型业务和受控载荷型业务。

保证型业务

这种类型的业务用于实时传输，这种传输需要保证端到端延迟最小。端到端延迟是指在各路由器中的延迟之和、在介质中的传播延迟和设置机制。只有第一项，即在各路由器中的延迟之和可由路由器来保证。这类业务保证分组在一定的传递时间内到达，并且只要数据流通信量在Tspec边界范围之内，即可保证不丢弃分组。可以说保证型业务就是定量业务。在这种业务中，端到端延迟量和数据速率必须由应用来定义。

受控载荷型业务

这种类型的业务用于可以接受一定延迟，但对超载网络和丢失分组的危险很敏感的应用。这些类型应用的较好实例有文件传输、电子邮件和因特网访问。受控载荷型业务是一种定性业务，因为应用要求较低的分组丢失率或零分组丢失率。

24.7.5 RSVP

在综合业务模型中，应用程序需要资源预留。正如在IntServ模型中所讨论的那样，为数据流进行资源预留。这意味着如果要在IP层使用IntServ，则需要在IP层之外创建一条数据流通路，它是一种虚电路网络，最初是用于数据报分组交换网络。在数据传输可以启动之前，虚电路网络需要信令系统来建立虚电路。资源预留协议（RSVP）是一种信令协议，它帮助IP创建一条数据流通路，从而实现资源预留。在讨论RSVP之前，需要指出的是，它是一种与综合业务模型分开的独立协议。它可以用于其他模型中。

多播树

RSVP不同于以前看到的一些其他信令系统，因为它是一种用于多播的信令系统。但是，

RSVP也能用于单播，因为单播仅仅是在多播组中只有一个成员的多播的一个特例。这样设计的原因是为了使RSVP能为包括经常使用多播的多媒体在内的各种通信量提供资源预留。

基于接收方的预留

在RSVP中，由接收方（而不是发送方）建立预留。这种策略和其他多播协议相反。例如，根据多播路由选择协议，由接收方（而非发送方）来决定加入或离开多播组。

RSVP报文

RSVP有多种类型的报文，然而根据需要，下面只讨论其中的两种：路径（Path）和预留（Resv）。

路径（Path）报文 回想一下在RSVP中数据流中的接收方预留资源。但是在预留资源之前，接收方并不知道分组经过的路径，而预留资源是需要知道路径的。为了解决这个问题，RSVP使用路径报文。路径报文从发送方传送出来，到达多播路径中的所有接收方。在整个路径中，路径报文为接收方存储必要的信息。在多播环境中将路径报文发送出去，并且在路径分叉时会产生新的报文。图24.22说明了路径报文。

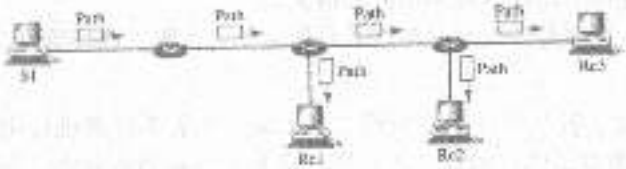


图24.22 路径报文

预留（Resv）报文 在接收方已经收到路径报文之后，它发送预留报文。预留报文朝着发送方传送（上行），并且在支持RSVP的路由器上预留资源。如果路由器不支持路径中的RSVP，它将使用以前所讨论的尽力传递方法对分组进行路由。图24.23说明了预留报文。

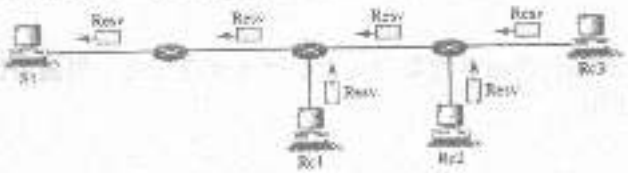


图24.23 预留报文

预留合并

在RSVP中，并没有为数据流中的每个接收方预留资源，而是将资源预留进行了合并。在图24.24中，Rc3请求2Mbps带宽，而Rc2请求1Mbps的带宽需要建立带宽预留的路由器R3将这两个请求合并。只对2Mbps带宽的那个请求（即两个中较大的那个请求）做预留，这是因为2Mbps的输入预留就能处理这两个请求。这种情况对R2也是适用的。读者可能问，为什么Rc2和Rc3（两者都属于一个单一数据流）要请求不同的带宽值呢？答案是，在多媒体环境中不同的接收方可以处理不同的质量等级。例如，Rc2或许只能以1Mbps（较低质量）的速率接收视频信息，而Rc3或许可以以2Mbps（较高质量）的速率接收视频信息。

预留方式

当存在多条数据流通路时，路由器需要预留资源来接收所有的数据流。RSVP定义了三种预留方式，如图24.25所示。

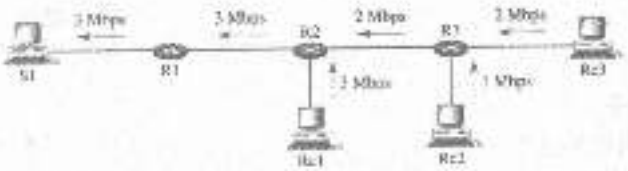


图24.24　预留合并

通配过滤器方式　在这种方式中，路由器为所有发送方创建一个预留。这个预留是基于最大的请求。这种方式用于来自不同发送方的数据流不同时出现的情况。

固定过滤器方式　在这种方式中，路由器为每个数据流创建一个不同的预留，这意味着如果有n个数据流，就要建立n个不同的预留。这种方式用于来自不同发送方的数据流同时出现的概率很高的情况。



图24.25　预留方式

共享显式方式　在这种方式中，路由器只创建一个预留，该预留可以被一组数据流共享。

软状态

为数据流而存储在每个节点中的预留信息（状态）需要定期进行刷新。与用于其他诸如ATM或帧中继这类虚电路协议的硬状态相比，这种状态称为软状态。在硬状态中，有关数据流的信息被一直保留着，直到这些信息被撤除为止。目前默认的刷新间隔是30秒。

24.7.6　综合业务所存在的问题

在综合业务中至少存在两个问题，这可能会妨碍了它在因特网中的实现。它们是可伸缩性和服务类型限制。

可伸缩性

综合业务模型要求每台路由器都要为每个数据流保存信息。但随着因特网规模的日益增长，这是一个严重的问题。

业务类型限制

综合业务只提供了两种类型的业务，即保证型业务和受控载荷型业务。那些反对该模型的人指出，应用可能需要比这两种业务类型更多的业务类型。

24.8　差分业务

差分业务（Differentiated Services，DS或Diffserv）是由因特网工程任务组（Internet Engineering Task Force，IETF）提出的用于弥补综合业务缺陷的一种业务。与综合业务相比，它有两个基本变化：

1. 主要处理过程从网络核心转移到网络边缘。这就解决了可伸缩性的问题。路由器不需要存储有关数据流的信息，每次发送分组时，应用程序或主机定义它们所需要的业务类型。

2. 将针对每条数据流的业务改变为针对每种类型的业务。路由器根据在分组中定义的业务类型，而不是根据数据流来发送分组。这就解决了业务类型限制的问题，可根据应用程序的需要定义不同的业务类型。

> 差分业务是基于IP的基于类的QoS模型。

DS字段

在Diffserv中，每个分组都包含一个称为DS的字段。该字段的值是在网络边界由主机或用作边界路由器的第一台路由器来设置的。IETF提议用DS字段替换现存的IPv4中的TOS（服务类型）字段或IPV6中的类字段，如图24.26所示。

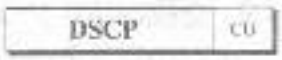


图24.26 DS字段

DS字段包含两个子字段：DSCP和CU。差分业务编码点（Differentiated Services Code Point，DSCP）是一个6位的子字段，它定义了逐跳行为（per-hop behavior，PHB）。长度为2位的当前未用（currently unused，CU）子字段当前还未使用。

Diffserv可用的节点（路由器）使用DSCP的6位作为定义分组处理机制表的索引，并且该定义是针对当前正在处理的分组的。

逐跳行为

Diffserv模型为接收到分组的每个节点定义了逐跳行为（PHB）。到目前为止已定义了三种PHB：默认（DE）PHB、加速转发（EF）PHB和保证转发（AF）PHB。

DE PHB　默认PHB和尽力传递是相同的，它与TOS相兼容。

EF PHB　加速转发PHB提供如下的服务：

- 低丢失率
- 低等待时间
- 可保证的带宽

这与源端和目的端之间使用的虚连接是相同的。

AF PHB　只要类通信量不超过节点的通信量规范，AF PHB就能以较高可靠性传送分组。网络用户要意识到一些分组可能会丢失。

通信量调节器

为了实现Diffserv，DS节点使用了几种调节器，如测量器、标记器、整形器和丢弃器。如图24.27所示。

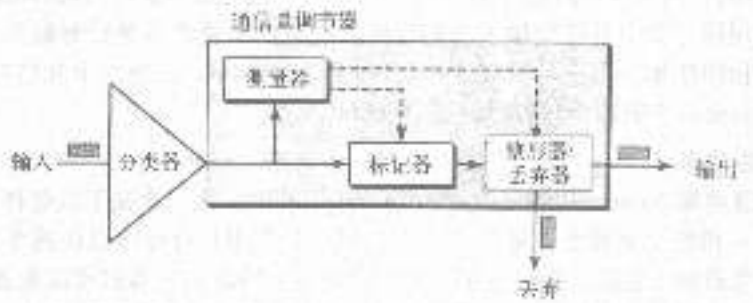


图24.27 通信量调节器

测量器　测量器检查输入数据流和协商的通信量规范是否匹配，并将这个结果发送给其他组件。测量器也能利用诸如令牌桶之类的工具来检查配置文件。

标记器　标记器能重新标记正使用尽力传递的分组（DSCP：000000）或依据接收自测量器的信息给分组作降标记。如果数据流与其对应的规范不匹配，就会对其作降标记（降低数据流的等级），标记器不对分组作升标记（提升等级）。

整形器　当通信量与协商的规范不匹配时，整形器使用接收自测量器的信息对该通信量进行重新整形。

丢弃器　丢弃器的运作过程就像没有缓冲区的整形器。如果数据流严重违反协商的规范，那么丢弃器就要丢弃这些分组。

24.9　交换网络中的QoS

讨论了IP协议中提出的QoS模型之后，现在来讨论一下用于两个交换网络中的QoS：帧中继和ATM。这两个网络都是虚电路网络，它们需要一个诸如RSVP的信令协议。

24.9.1　帧中继中的QoS

在帧中继中已经设计了控制通信量的四个不同的属性指标：访问速率，提交突发业务量B_c，提交信息速率（CID）和超突发长度B_e。它们是在用户和网络进行协商期间设置的。对于PVC连接而言，它们需协商一次，而对于SVC连接而言，在连接建立期间要为每个连接进行协商。图24.28说明了这四个属性指标间的关系。

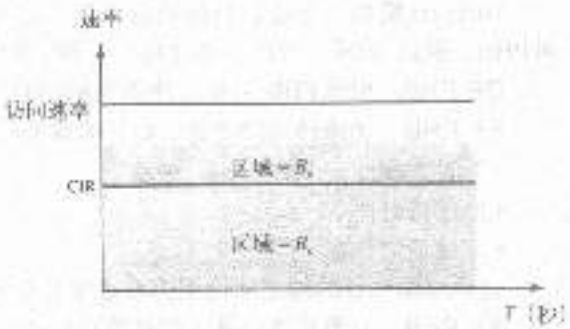


图24.28　通信量控制属性指标间的关系

访问速率

每个连接都要定义**访问速率**（access rate）（以每秒位为单位）。访问速率实际上是由连接用户与网络的通道的带宽来决定的。用户永远不能超过这个速率。例如，如果用T-1线路将用户连接到帧中继网络，则访问速率是1.544Mbps，并且用户永远无法超过这个速率。

提交突发业务量

对于每个连接而言，帧中继定义了**提交突发业务量**（committed burst size）B_c。该值是在一段预定义的时间内，在没有丢弃任何帧或设置DE位的条件下，网络所传输的最大比特数。例如，在4秒的时间段内B_c的值为400kbit是可以的，因为用户能在4秒的间隔内传送400千位的数据，而不用担心会出现任何帧丢失的情况。注意这个量并不是针对每个1秒而定义的速率，而是一个累积的度量。用户可以在第一个1秒发送300kbit，在第二个和第三个1秒期间不发送任何数据，而最后在第四个1秒期间发送100kbit。

提交信息速率

提交信息速率（committed information rate，CIR）除了定义了以每秒比特为单位的平均速率之外，它和提交突发业务量在概念上类似。如果用户持续地以该速率发送数据，那么网络就能一直发送帧。但是，因为它是一个平均度量，所以用户有时可以发送高于CIR的数据而在其他时间里发送低于CIR的数据。只要预定义期间的平均速率符合要求，就会发送这些帧。在预定义期间，发送的累积比特数不能够超过B_c。注意CIR不是一个独立的度量，可使用如下的公式对它进行计算：

$$\mathrm{CIR}=\frac{B_c}{T}\ (\mathrm{bps})$$

例如，如果在5秒内，B_c是5kbit，那么CIR就是5 000/5，即1kbps。

超突发长度

对于每个连接而言，帧中继都定义了**超突发长度**（excess burst size）B_e。该值是在一段预定义的时间内，用户发送的超过B_c的最大位数。如果网络中没有拥塞，网络就保证传输这些

比特。注意，这种约定实现的可能性比在B_c情况下要低，因为网络本身实现这种约定是有条件的。

用户速率

图24.29说明了用户如何才能发送突发性数据。如果用户发送的数据没超过B_c，那么网络就能保证传送帧而不会出现丢弃帧的情况。如果用户发送的数据超过了B_c而小于B_e（也就是全部比特数小于$B_c + B_e$），那么在网络中没有拥塞的条件下，它就能保证传送所有的帧。如果出现拥塞情况，将会丢失一些帧。从用户接收帧的第一台交换机有一个计数器，它可为超过B_c的帧设置DE位。如果网络出现拥塞情况，那么其余的交换机将会丢弃这些帧。注意，需要较快发送数据的用户可能超过B_c，只要该指标不超过$B_c + B_e$，那么这些帧就有可能到达目的端而不丢失。然而，要记住的是，当在某一时刻用户发送的数据超过了$B_c + B_e$，那么第一台交换机就会丢弃此后所发送的所有帧。

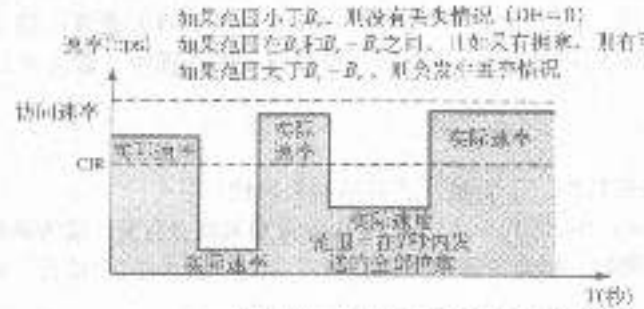


图24.29 用户送率和B_c与$B_c + B_e$的关系

24.9.2 ATM中的QoS

ATM中的QoS是基于类、与用户相关的属性和与网络相关的属性。

类

ATM论坛定义四种服务类型：恒定比特率（CBR）、可变比特率（VBR）、可用比特率（ABR）和未指定比特率（UBR）（见图24.30所示）。



图24.30 服务类型

CBR 恒定比特率（constant-bit-rate，CBR）类用于需要实时音频或视频服务的客户。这个服务与诸如T线之类的专线提供的服务类似。

VBR 可变比特率（variable-bit-rate，VBR）类分为两个子类：实时（VBR-RT）和非实时（VBR-NRT）。VBR-RT用于需要实时服务（如声音和视频传输）和使用压缩技术来生成可变比特率的用户；VBR-NRT用于不需要实时服务，但是使用压缩技术生成可变比特率的那些用户。

ABR 可用比特率（available-bit-rate，ABR）类以最小的速率发送信元。如果有更多的网络资源可用的话，那么就能突破这个最小比特率。ABR特别适用于突发性的应用。

UBR 未指定比特率（unspecified-bit-rate，UBR）类是一种尽力传递但没有任何保证的服务。

图24.31说明了不同类的服务与网络总容量的关系。

用户相关属性

ATM定义了两组属性。用户相关属性是定义用户要以多快的速率发送数据的属性，是在用户和网络之间约定时协商而定的。下面是一些用户相关属性：

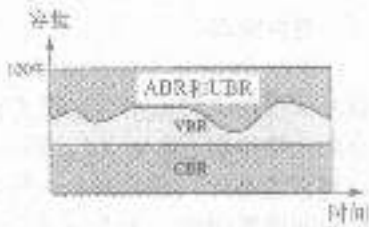


图24.31　不同类型服务和网络容量间的关系

SCR　持续信元速率（sustained cell rate, SCR）是一段较长时间间隔的平均信元速率。实际信元速率可能低于或高于这个值，但是平均值应该等于或小于SCR。

PCR　峰值信元速率（peak cell rate, PCR）定义了发送方的最大信元速率。只要能维持SCR，用户的信元速率有时就能达到该峰值。

MCR　最小信元速率（minimum cell rate, MCR）定义了发送方能接受的最小信元速率。例如，如果MCR是50 000，那么网络必须保证发送方至少每秒发送50 000个信元。

CVDT　信元可变延迟极值（cell variation delay tolerance, CVDT）是表示信元传输时间变化量的一个量。例如，如果CVDT是5ns，则表示在传递信元的过程中，最大延迟和最小延迟之差不能超过5ns。

网络相关属性

网络相关属性定义了网络的特性。下面是一些网络相关属性：

CLR　信元丢失率（cell loss ratio, CLR）定义的是在传输期间信元丢失（或传递得太迟以致于它们被视为丢失）的比率。例如，如果发送方发送了100个信元，丢失了1个信元，则CLR是

$$\mathrm{CLR} = \frac{1}{100} = 10^{-2}$$

CTD　信元传送延迟（cell transfer delay, CTD）是一个信元从源端传输到目的端所需要的平均时间。最大CTD和最小CTD也属于网络相关属性。

CDV　信元延迟变化量（cell delay variation, CDV）是CTD的最大值和最小值之差。

CER　信元差错率（cell error ratio, CER）定义错误传递信元的比率。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物和网站，在本节末列出方括号[……]中的参考资料。

读物

拥塞控制和QoS在[Tan03]中的5.3节和5.5节以及[Sta98]的第3章中有讨论。关于QoS入门读物可参看[FH98]。在[Bla00]中有关于QoS的完整讨论。

本章小结

- 平均数据速率、峰值数据速率、最大突发长度和有效带宽是用来描述数据流的定性值。
- 数据流可以具有恒定比特率、可变比特率或者突发性通信量。
- 拥塞控制是指控制拥塞和保持载荷低于容量的机制和方法。
- 延迟和吞吐量可衡量网络的性能。
- 开环拥塞控制是预防拥塞；闭环拥塞控制是消除拥塞。
- TCP通过使用两个策略来避免拥塞：慢速启动与加性增加相结合和乘性减少。

- 帧中继通过使用两个策略来避免拥塞：后向显式拥塞通知（BECN）和前向显式拥塞通知（FECN）。
- 数据流可以由可靠性、延迟、抖动和带宽这些特性来描述。
- 调度、通信量整形、资源预留和许可控制都是提高服务质量（QoS）的方法。
- FIFO队列、优先权队列和加权公平队列都是调度方法。
- 漏桶和令牌桶是通信量整形方法。
- 综合业务是用于IP的基于数据流的QoS模型。
- 资源预留协议（RSVP）是一种信令协议，它帮助IP生成数据流，并建立资源预留。
- 差分业务是用于IP的基于类的QoS模型。
- 在帧中继中，访问速率、提交突发业务量、提交信息速率和超突发长度都是控制通信量的属性。
- ATM中的服务质量是基于类、用户相关属性和网络相关属性的。

习题

复习题

1. 拥塞控制和服务质量的关系如何？
2. 什么是通信量描述符？
3. 平均数据速率和峰值数据速率之间关系是什么？
4. 突发性数据的定义是什么？
5. 开环和闭环拥塞控制之间的区别是什么？
6. 试列出防止拥塞的一些策略的名称。
7. 试列出减轻拥塞的一些机制。
8. 在TCP中，发送窗口大小由什么来决定？
9. 帧中继如何控制拥塞？
10. 哪些特性可以用来描述数据流？
11. 提高服务质量的四个常用方法是什么？
12. 什么是通信量整形？试列出通信量整形的两种方法的名称。
13. 综合业务和差分业务的主要区别是什么？
14. 资源预留协议和综合业务有什么关系？
15. 在帧中继中用于通信量控制的属性是什么？
16. 在ATM中，就服务质量来说，网络相关性与用户相关性的差别是什么？

练习题

17. 帧中继的地址字段是10110000000010111。试问在前向有拥塞吗？在后向有拥塞吗？
18. 一个帧从A到B，在两个方向都有拥塞。试问FECN位是否置位？BECN位是否置位？
19. 使用漏桶控制液体的流量，如果输出速率是5gal/min。输入突发性速率是100gal/min，持续12s，在48s内没有任何输入。试问桶中还剩下多少加仑(gal)的液体？
20. 交换机的输出接口使用漏桶算法以8000字节/s（节拍）的速率发送数据，如果接下列的帧顺序接收，试说明每秒发送了哪些帧。

    帧1，2，3，4：每帧4000字节

    帧5，6，7：每帧3200字节

    帧8，9：每帧400字节

帧10，11，12：每帧2000字节

21. 一个用户通过T-1线连接到帧中继网络，允许的CIR是1Mbps，B_c是5Mb/5s而B_e是1Mb/s。试回答下列问题：
    a. 访问速率是多少？
    b. 用户能以1.6Mbps发送数据吗？
    c. 用户能始终以1Mbps发送数据吗？在这种情况下，它能保证不丢失帧吗？
    d. 用户能始终以1.2Mbps发送数据吗？在这种情况下，它能保证不丢失数据吗？如不是，它是否能保证只有在网络发生拥塞时，才会丢失帧？
    e. 以恒定比特率1.4Mbps重复d的问题。
    f. 用户能始终使用而无需担心帧丢失的最大数据速率是多少？
    g. 如果用户想冒险，那么在网络没有拥塞，没有丢弃帧的情况下，能使用的最大数据速率是多少？
22. 在练习21中，用户以1.4Mbps发送了2s的数据，接下来的3s内没有发送任何数据。如果网络没有拥塞，试问有丢弃帧的危险吗？如果网络中有拥塞，有丢弃帧的危险吗？
23. 如果每个信元都经过10μs到达目的端，CTD的值是多少？
24. 在10 000个信元中，网络已丢失了5个信元，其中2个是错误信元，那么CLR的值是多少？CER的值是多少？

第六部分 应 用 层

目标

应用层使用户能够访问网络，用户可以是人或软件。它为用户提供服务的接口和支持，如电子邮件、文件访问和传输、访问系统资源、浏览万维网，及网络管理。

应用层负责向用户提供服务。

本部分简要地讨论为因特网中的客户机/服务器模式设计的一些应用程序。客户机向服务器发送一个服务请求，而服务器对客户机做响应。

本书第六部分重点介绍应用层及其提供的服务。

第25章 域名系统

在因特网模型的应用层中，许多应用都遵循客户机/服务器模式。客户端/服务器应用程序可以分成两类：一种直接被用户使用，例如电子邮件，另一种是支持其他应用程序的应用程序。域名系统（DNS）就是一种支持程序，它被其他的应用程序使用，例如电子邮件。

图25.1举例说明了DNS客户端/服务器程序如何支持电子邮件程序找到电子邮件接收者的IP地址。电子邮件程序的用户可以知道接收者的电子邮件地址，但是IP协议需要知道IP地址。DNS客户端程序就向DNS服务器发送查询请求，获取与电子邮件地址相对应的IP地址。

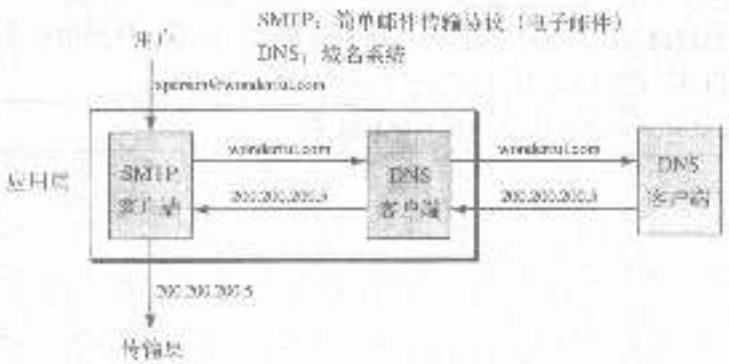


图25.1 使用DNS服务的例子

为了识别一个实体，TCP/IP协议使用IP地址唯一地确定一台主机到因特网的连接。然而，人们更喜欢使用名字而不是数字地址。所以，需要一种能够完成名字到地址或者地址到名字的映射的系统。

在因特网发展初期，使用一个**主机文件**（host file）实现映射。主机文件只有两列：名字和地址。每台主机可以将主机文件存储在主机磁盘中，并以一个标准主机文件为依据定期地进行更新。当程序或者用户需要将某一名字映射到一个地址时，主机就查询主机文件并找到相应的映射。

但目前，不再可能用一个单独的主机文件将每一个地址与名字关联起来，反过来也是一样。主机文件会因为太大而无法存储在每一台主机中。另外，也不能每次在发生变化时，更新世界上所有的主机文件。

一种解决方案是将全部主机文件存储在一台单独的计算机中，允许每一台需要映射的计算机访问这个集中化的信息。但是，这样会在因特网上产生非常大的通信量。

另一种解决方案，也是目前所使用的，是将这种巨大的信息数据分割为许多更小的部分，并将每一部分存储在不同的计算机中。采用这种方式，需要映射的计算机可寻找到最近一台持有所需信息的计算机。**域名系统**（Domain Name System，DNS）就使用这种方法。本章先讨论DNS的概念和思想，然后描述DNS协议本身。

25.1 名字空间

为实现无二义性，分配给机器的名字必须从**名字空间**（name space）中仔细地选择，该名

字空间完全控制对名字和IP地址的绑定。换句话说，因为地址是唯一的，所以名字也必须唯一的。名字空间将每一个地址映射到一个唯一的名字，它可以按两种方式进行组织：平面的和层次的。

25.1.1 平面名字空间

在平面名字空间（flat name space）中，一个名字分配给一个地址。空间中的名字是一个无结构的字符序列。名字之间可能有也可能没有公共部分；即使有公共部分，也没有实际含义。平面名字空间的主要缺点是它必须集中控制才能避免二义性和重复。因而不能用于如因特网这样的大规模系统中。

25.1.2 层次名字空间

在层次名字空间（hierarchical name space）中，每一个名字由几个部分组成。第一部分可以定义组织的性质，第二部分可以定义一个组织的名字，第三部分可以定义组织的部门，等等。在这种情况下，分配和控制名字空间的机构就可以分散化。中央管理机构可以分配名字的一部分，这部分定义组织的性质和组织的名字，名字其他部分的分配可交给这个组织自身。这个组织可以给名字加上后缀（或前缀）来定义主机或者其他资源。这个组织机构的管理机构不必担心为一个主机选择的后缀会被其他组织机构所采用，因为即使地址的某一部分相同的，整个地址也是不同的。例如，假定两所学院和一个公司都将它们的一台计算机命为challenger。第一所学院由中央管理机构分配的名字是fhda.edu，第二所学院获得的名字是berkeley.edu，而公司获得的名字是smart.com。当这些组织的每一个在已有的名字加上名字challenger后，得到了三个不同的名字：challenger.fhda.edu、challenger.berkeley.edu和challenger.smart.com。这些名字是唯一的，而不需要由中央管理机构来分配。中央管理机构只控制名字的一部分，而不是整个名字。

25.2 域名空间

为了获得层次结构的名字空间，设计了**域名空间**（domain name space）。在这种设计方式中，所有的名字由根在顶部的倒置树结构定义。该树最多有128级：0级（根节点）~127级（见图25.2所示）。

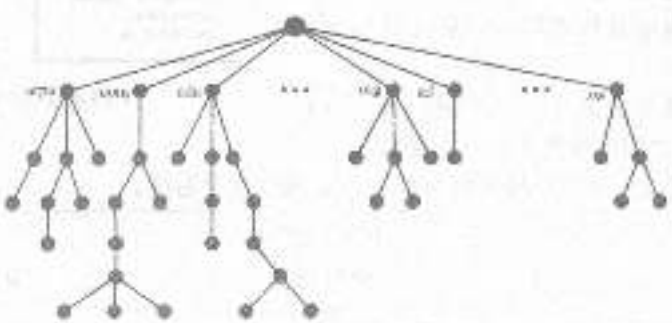


图25.2 域名空间

25.2.1 标号

树上的每一个节点有一个**标号**（label）。标号是一个最多为63个字符的字符串。根节点的标号是空字符串（空串）。DNS要求每一个节点的子节点（从同一节点分支出来的节点）有不

同的标号。这样就确保了域名的唯一性。

25.2.2 域名

树上的每一个节点都有一个域名。一个完全的域名（domain name）是用点（.）分隔的标号序列。域名总是从节点向上读到根节点。最后一个标号是根节点的标号（空）。这表示一个完全的域名总是以一个空标号结束，也意味着最后一个字符是一个点（.），表为空字符串表示什么也没有。图25.3给出了某些域名。

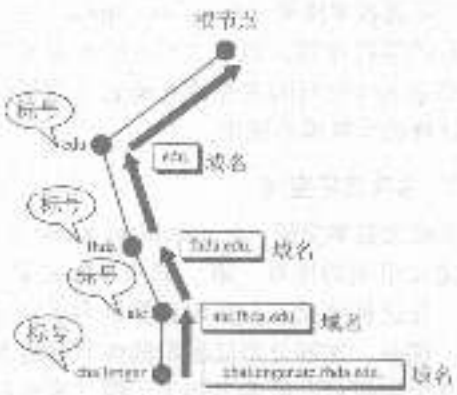


图25.3　域名和标号

全称域名

如果一个标号以一个空字符串结束，则它就称全称域名（fully qualified domain name，FQDN）。全称域名是包含一台主机所有名字的域名。它包含所有的标号，从最具体的到最一般的标号，并能唯一地定义一台主机的名字。例如，域名

```
challenger.atc.fhda.edu.
```

是一个全称域名，代表安装在De Anza大学的高级技术中心（ATC）的一台名字为*challenger*的计算机。DNS服务器只能和FQDN的一个地址相匹配。注意：名字必须以空标号结束。但是由于空标号表示没有东西，所以这种标号以一个点（.）结束。

部分域名

如果一个域名不是以空字符串结束，则称为部分域名（partially qualified domain name，PQDN）。部分域名起始于一个节点，但没有到达根节点。当这个需要解析的名字属于和客户机相同的站点时使用部分域名。这种情况下，解析程序能够提供省略的部分，称为后缀（suffix），以创建FQDN。例如，如果在fhda.edu.站点的一个用户想要获取challenger计算机的IP地址，用户就可定义这个部分域名

```
challenger
```

DNS客户在将地址传送到DNS服务器后就加上后缀atc.fhda.edu.。



图25.4　FQDN和PQDN

DNS客户端通常会保存一个后缀列表。下面这些是De Anza大学的后缀列表。空后缀表示什么也没有，当用户定义一个全称域名的时候，就会加上这个后缀。

```
atc.fhda.edu
fhda.edu
null
```

图25.4表示了一些FQDN和PQDN。

25.2.3 域

域（domain）是域名空间的一棵子树。这个域的名字是子树顶部节点的域名。图25.5列出了一些域。注意：一个域本身也可以再分划为多个域。（也就是平时所说的子域（subdomain））。

25.3 名字空间的分布

必须将域名空间所包含的信息存储起来。然而，只使用一台计算机存储如此大容量的信息，效率非常低和不安全的。效率低的主要是因为响应来自世界各地的请求会给系统造成非常大的负荷，不安全的原因是因为任何故障将使整个数据库无法使用。

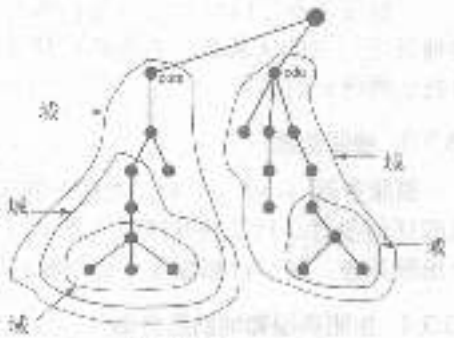


图25.5 域

25.3.1 名字服务器的层次结构

解决这些问题的办法是将信息分布在多台称为DNS服务器（DNS server）的计算机中。一种方法是将整个空间划分为多个基于第一级的域。换句话说，让根节点保持不动，但创建许多与第一级节点一样多的域（子树）。这样创建的域会很大，DNS允许将域进一步分划为更小的域（子域）。每一台服务器对一个大的域或者较小的域是负责的（授权的）。换句话说，与建立名字的层次结构一样，也建立了服务器的层次结构（见图25.6）。

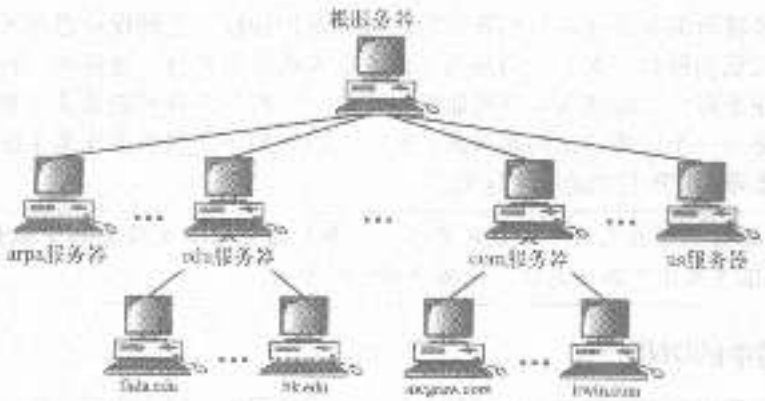


图25.6 名字服务器的层次结构

25.3.2 区域

既然完整的域名层次结构不能保存在单一的服务器上，那么它被分在多个服务器上。一个服务器负责或者授权的范围，称为区域（zone）。我们可以将一个区域定义为整个树中的一个连续的部分。如果服务器负责一个域，而且这个域并没有进一步被划分为更小的子域，此时域和区域是相同的。服务器有一个数据库，称为区域文件，它保存这个域里所有节点的信息。然而，如果服务器将它的域划分为多个子域，并将其部分授权委托给其他服务器，那么域与区域就不同了。在子域节点的信息会存放在较低层次的服务器中，原来的服务器则保存到这些较低层次服务器的某种参照。当然，原来的服务器并不是完全不负责任，它仍然拥有一个区域，只是将详细的信息保存在较低层次的服务器上。（见图25.7所示）。

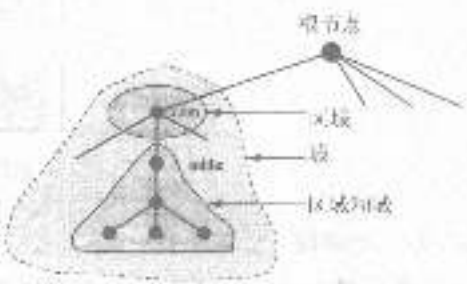


图25.7 区域和域

一台服务器可以将自己域进行划分并委托责任，但是仍然为它自己保存了一部分，在这种情况下，它的区域是由具有详细信息的那部分域所组成，它没有被委托权利，但它有到被委托那些部分的参照。

25.3.3 根服务器

根服务器（root server）是指它的区域由整棵树组成的服务器。根服务器通常不保存关于域的任何信息，只是将其授权托给其他服务器，并保存与这些服务器的参照关系。目前有多个根服务器，每一台都覆盖了整个域名空间。这些服务器分布在世界各地。

25.3.4 主服务器和辅助服务器

DNS定义了两种类型的服务器：主服务器和辅助服务器。**主服务器**（primary server）是指存储了授权区域有关文件的服务器。它负责创建、维护和更新区域文件，并将区域文件存储在本地磁盘中。

辅助服务器（secondary server）负责从另一个服务器（主服务器或者辅助服务器）传输一个区域的全部信息，并将文件存储在它的本地磁盘中。辅助服务器既不创建也不更新区域文件。如果需要更新，则必须由主服务器来完成，由主服务器发送更新的版本到辅助服务器中。

主服务器和辅助服务器对它们所服务的区域都有控制权。这种设计思想并不是把辅助服务器置于一个较低的授权层次上，而是为了建立数据的冗余备份，这样当一台服务器出现故障时，另一台服务器可以继续为客户机服务。注意：一台服务器可能是某个特定区域的主服务器，同时也是另一个区域的辅助服务器。所以，当提到一个服务器作为主服务器或者辅助服务器时，需要弄清楚所指的是哪个区域。

> 主服务器能够从磁盘文件中获取所有信息，辅助服务器从主服务器中获取信息。当辅助服务器从主服务器中下载信息时，这称为区域的传送。

25.4 因特网中的DNS

DNS是一种可以在不同平台上使用的协议。在因特网中，域名空间（树）被划分为三个部分：通用域、国家域和反向域（见图25.8所示）。

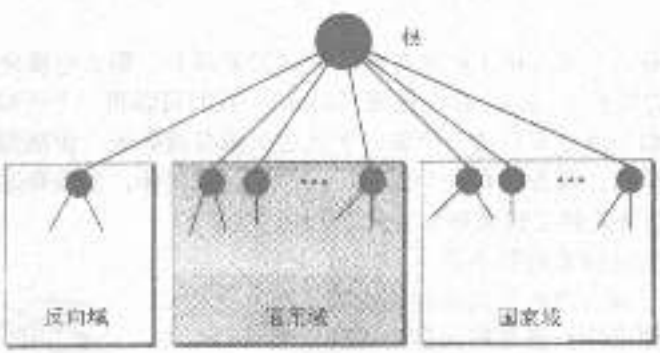


图25.8 因特网中的DNS

25.4.1 通用域

通用域（generic domain）按照已经注册主机的一般行为对主机进行定义。树中的每一个

节点定义一个域，它是到域名空间数据库的一个索引（见图25.9所示）。

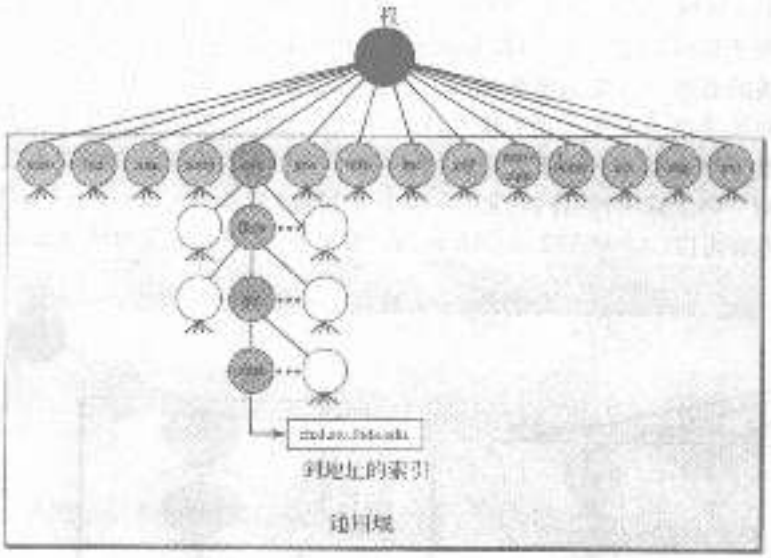


图25.9 通用域

我们从这棵树中可以看出，在通用域的第一层允许有14个可能的标号。这些标号描述了表25.1中列出的组织机构类型。

表25.1 通用域标号

标 号	描 述	标 号	描 述
aero	航空航天公司	int	国际机构
biz	商业或者公司（与com类似）	mil	军事机构
com	商业机构	museum	博物馆和其他非盈利性组织
coop	合作商业组织	name	私人名字（个人名字）
edu	教育机构	net	网络支持中心
gov	政府机构	org	非营利性组织
info	信息服务提供商	pro	专业个人组织

25.4.2 国家域

国家域（country domain）部分使用两个字母的国家缩写（例如us代表美国）。第二级标号可以是组织机构，或者更具体一些，由各个国家自己指定。例如，美国使用州的缩写作为国家域的子域划分（例如ca.us.）。

图25.10列出了国家域部分，地址anza.cup.ca.us可以理解为美国加州的Cupertino的De Anza学院。

25.4.3 反向域

反向域（inverse domain）用于将地址映射为名字。例如，当服务器接收到客户机请求完成某项任务的请求时，就会发生这种情况。尽管服务器中有一个包含着授权客户的列表文件，但文件中只列出了客户机的IP地址（从接收到的IP分组中提取出来）。为了确定该客户端是否在授权的列表中，服务器就用它的解析程序向DNS服务器发送一个查询，并请求将地址映射为名字。

这种类型的查询，称为反向或者指针（PTR）查询。为了处理一个指针查询，在域名空间中要增加一个反向域，并且其第一级节点称为*arpa*（由于历史原因）。第二级仍然是单节点，称为*in-addr*（用于反向地址）。域的其他部分定义IP地址。

处理反向域的服务器也是层次结构的。这意味着地址的网络号比子网号的层次更高，而子网号所处的层次要高于主机号所处的层次。同样，为整个网站服务的服务器应该比为子网服务的服务器处于更高层次。与通用域或者国家域相比，这种配置方法使得反向域看起来是倒置的。按照由下向上读取标号的约定，一个IP地址如132.34.45.121（一个B类地址，网络号是132.34）会读取为121.45.34.132.in-addr.arpa。参见图25.11关于反向域配置说明的实例。

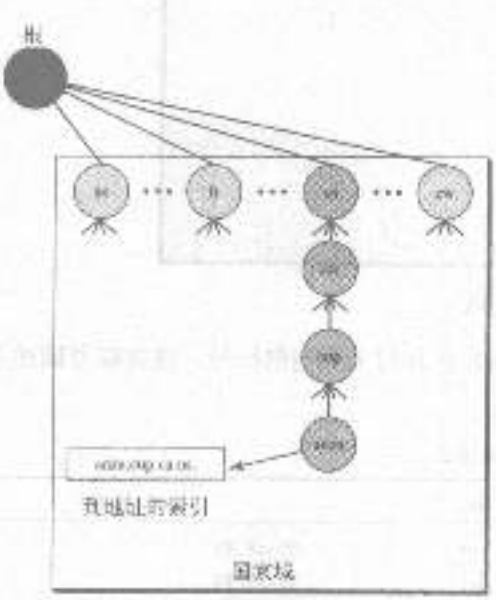


图25.10　国家域

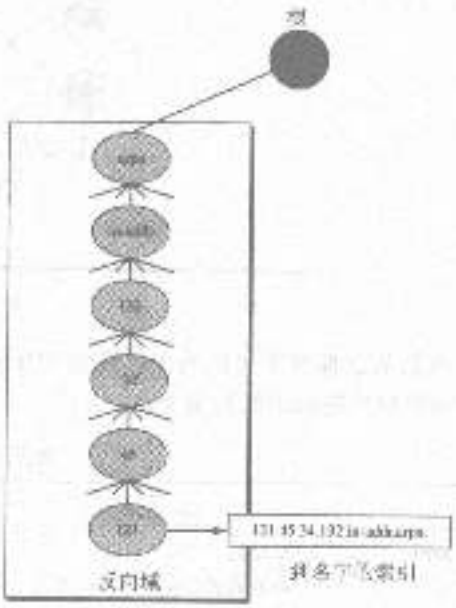


图25.11　反向域

25.5　解析

将名字映射为地址或者将地址映射为名字的过程，称为名字-地址解析。

25.5.1　解析程序

DNS是一个客户机/服务器应用程序。需要将地址映射为名字或者将名字映射为地址时，主机要调用一个称为解析程序（resolver）的DNS客户程序。解析程序用一个映射请求访问最近的一个DNS服务器。如果服务器含有该信息，它就满足解析程序的请求；否则，它将解析程序交付给其他的服务器，或者查询其他的服务器来提供这种信息。

当解析程序接收到映射后，它解释这一响应，以确定它是一个真正的解析还是一个差错，最后将结果传递给发出这一请求的进程。

25.5.2　名字到地址的映射

多数情况下，解析程序将域名提交给服务器，请求给出对应的地址。这种情况下，服务器检查通用域或者国家域以查找相应的映射。

如果域名来自于通用域部分，解析程序就会收到一个域名，如*chal.atc.fhda.edu.*。由解析程序将这个查询发送到本地DNS服务器进行解析。如果本地服务器不能解析这一查询，它或

者把解析程序提交给其他的服务器，或者直接询问其他的服务器。

如果域名来自于区家或部分，那么解析程序会接收到类似于*ch.fhda.cu.ca.us.*形式的域名。处理过程与上相同的。

25.5.3 地址到名字的映射

客户机向服务器发送IP地址，请求映射为域名。如前所述，这被称为一个PTR查询。要回复这种查询，DNS使用反向域。然而，在这种请求中，IP地址必须反过来，并且将*in-addr*和*arpa*两个标号附加在最后，以创建能够被反向域部分接受的域。例如，如果解析程序接收到的IP地址为*132.34.45.121*，那么解析程序会首先将地址反过来，然后在发送之前附加两个标号。发送的域名是*121.45.34.132.in-addr.arpa.*，它由本地DNS接收并解析。

25.5.4 递归解析

客户端（解析程序）可以从名字服务器中请求递归应答。这意味着解析程序期望服务器提供最终的答案。如果服务器是这一域名的授权服务器，它会检查它的数据库并做出响应。如果服务器不是授权服务器，它会把请求发送给另一个服务器（通常是父服务器），并等待响应。如果父服务器是授权服务器，它就做出响应。否则，它仍然把这个查询发送给另一个服务器。当这个查询最终得到解析后，响应就后向传送，直到最终到达发出请求的客户机。这就是**递归解析**（recursive resolution），如图25.12所示。

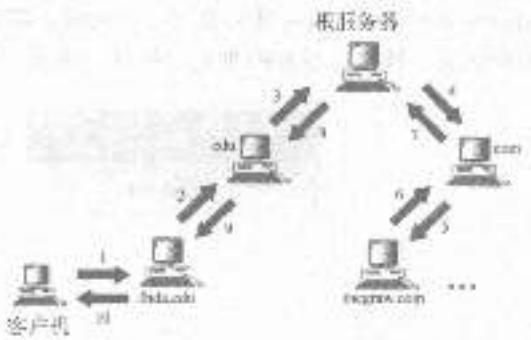


图25.12 递归解析

25.5.5 迭代解析

如果客户端没有请求递归应答，那么映射可以迭代的形式进行。如果服务器是该名字的授权服务器，那么它发送应答。如果它不是，它就返回它认为可以解析该这个查询的服务器的IP地址（给客户端），由客户端负责向第二台服务器重复发送请求。如果新的地址解析服务器能够解析这一问题，那么它就用IP地址响应这一请求；否则，它向客户端返回新服务器的IP地址。这时，客户端必须向第三台服务器重复该请求。这一过程称为**迭代解析**（iterative resolution），因为客户端向多台服务器重复同样的请求。在图25.13中，客户端在从mcgraw.com服务器得到应答之前，向四台服务器发送了查询请求。

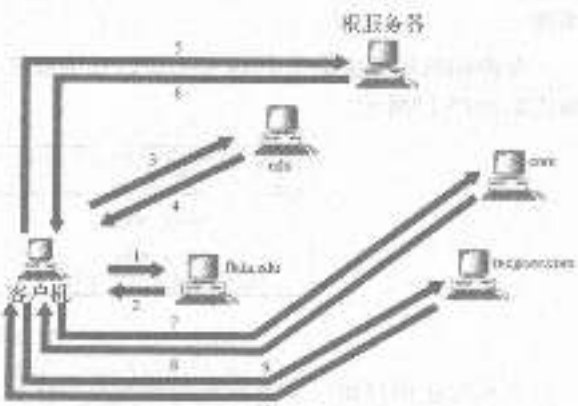


图25.13 迭代解析

25.5.6 高速缓存

每当服务器接收到查询一个不属于自己域的名字时，它需要搜索自己的数据库以查找

台服务器的IP地址。缩短这一查询时间能提高效率。DNS使用一种称为高速缓存（caching）的机制处理这一问题。当一个服务器向另一个服务器请求映射并得到回应时，它在将该回应发送给客户端之前，先将这一信息存储在高速缓存中。如果同一客户端或者另一个客户端请求同一映射时，它会检查其高速缓存并解决这一问题。然而，要通知客户这一响应来自于高速缓存而不是来自于授权的信息源，该服务器会将这一响应标志为非授权性的。

高速缓存能够加快解析过程，但仍存在问题，如果一台服务器长时间缓存一个映射时，可能会发送给客户端一个过期的映射。为了防止这种情况，使用了两种技术。第一种，授权服务器总是将称为生存时间（TTL）的信息添加在映射上。生存时间定义了接收服务器可以将信息放入高速缓存的时间（以秒计）。超过这一时间，该映射就变为无效，而任何查询必须再次发送到授权服务器。第二种技术，DNS要求每一台服务器对每一个映射保留一个TTL计数器。高速缓存会定期检查，并清除掉TTL已经过期的那些映射。

25.6 DNS报文

DNS有两种类型的报文：查询和响应。这两种类型的报文具有相同的格式。**查询报文**（query message）由头部和查询记录构成；**响应报文**（response message）由头部、查询记录、响应记录、授权记录和附加记录组成（见图25.14所示）。

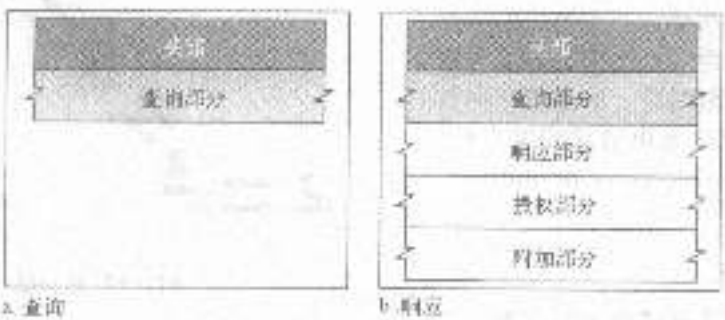


图25.14 查询和响应报文

头部

查询和响应报文的头部格式相同，查询报文头部的某些字段设置为0。头部为12个字节，格式如图25.15所示。

标识	标记
查询记录数	应答记录数
授权记录数 （在查询报文中为全0）	附加记录数 （在查询报文中为全0）

图25.15 头部格式

客户端使用标识字段来匹配对查询的响应。客户端每次发送查询时会使用不同的标识号。服务器在对应的响应中会重复这一编号。标记是子字段的集合，这些子字段定义了报文的类型、应答的类型、期望的解析类型（递归或迭代）等。查询记录数是指该报文中询问部分所含请求的数量。应答记录数是指响应报文中应答部分所含应答记录的数量。在查询报文中，它的值为0。授权记录数是指响应报文中授权部分所含授权记录的数量。在查询报文中，它的值为0。最后一项，附加记录数是指响应报文中的附加部分所含附加记录的数量。在查询

报文中，它的值为0。

询问部分

它由一条或者多条询问记录构成。查询和响应报文都含有这一部分。在下一节中，我们将讨论询问记录。

应答部分

它由一条或者多条资源记录组成，它仅存在于响应报文中。这一部分包括从服务器到客户端（解析器）的应答。在下一中，我们将讨论资源记录。

授权部分

它由一条或者多条资源记录构成，它仅存在于响应报文中。该部分给出了用于查询的一台或者多台授权服务器的信息（域名）。

附加消息部分

它由一条或者多条资源记录构成，它仅存在于响应报文中。该部分给出了有助于解析程序的附加信息。例如，服务器可以在授权部分为解析程序提供授权服务器的域名，并且把同一授权服务器的IP地址包含在附加信息部分中。

25.7 记录的类型

从25.6节中可以看出，DNS使用了两种类型的记录。在查询和响应报文的询问部分使用了询问记录；在响应报文中的应答、授权、附加信息部分使用了资源记录。

25.7.1 询问记录

客户机使用询问记录（question record）从服务器获取信息，它包含了域名。

25.7.2 资源记录

每一个域名（在树中的每一个节点）都与一个资源记录（resource record）相关联。服务器数据库包含了所有的资源记录，服务器还返回资源记录给客户机。

25.8 注册机构

新的域名是怎样加入到DNS中呢？这是通过注册机构（registrar）来完成的。一个熟知的商业实体是ICANN（因特网名字和编号分配组织）。注册机构首先确认询问的域名是唯一的，然后将它输入到DNS数据库中，这是需要收费的。

现在，有很多的注册机构，他们的名字和地址可以在网站http://www.intenic.net中找到。

为了能够注册，组织机构需要给出域名服务器的主机名和IP地址，例如，一个名字为*wonderful*的商业机构的域名服务器主机名为*ws*，IP地址为200.200.200.5，就需要给出以下的信息给注册机构：

```
域名：ws.wonderful.com
IP地址：200.200.200.5
```

25.9 动态域名系统（DDNS）

在设计DNS时，没有人预料到地址会有如此多的变化。在DNS中，当只有一个变化时，例如增加一台新主机，移除一台主机或改变一个IP地址时，那么这种更改就必然使DNS的主文件发生变化。这些类型的变化涉及许多手工的更新。因特网今天的规模已经不允许使用这

种手工操作。

DNS主文件必须能动态更新。动态域名系统（Dynamic Domain Name System（DDNS））就是为满足这种需求而设计的。在DDNS中，当名字和地址之间的绑定确定时，通常是由DHCP（见第21章）给上DNS服务器发送这种信息。主服务器更新这一区域。通知辅助服务器的方法可以以主动方式或者以被动方式。在主动通知方式中，主服务器向辅助服务器发送关于区域变化的报文；而在被动通知方式中，辅助服务器定期地检查是否有任何变化。无论使用哪一种方式，当得到变化的通知时，辅助服务器会请求整个区域的信息（区域传输）。

为了提供安全性以防止对DNS记录的非授权更改，DDNS可以使用鉴别机制。

25.10 封装

DNS可以使用UDP或者TCP协议。在这两种情况下，服务器使用的熟知端口是53。当响应报文的长度小于512字节时，就使用UDP。这为大多数UDP分组有512字节分组大小的限制。如果响应报文的长度大于512字节，则必须使用TCP连接。在这种情况下，可能会发生以下两种情况：

- 如果解析程序事先知道响应报文超过512字节，那么它必须使用TCP连接。例如，如果辅助名字服务器（作为客户端）需要从主服务器进行区域传输，那么必须使用TCP连接。因为被传输的信息通常是超过512字节的。
- 如果解析程序不知道响应报文的大小，那么可以使用UDP端口。但是，如果响应报文超过512字节，那么服务器会截断这一报文，此时解析程序会开启TCP连接，并重复该请求，从服务器中获得完整的响应。

DNS可以在熟知端口53使用UDP或者TCP服务。

推荐读物

为了深入讨论本章有关的主题，推荐下面的读物和网站。在本章末列出方括号[…]中的参考资料。

读物

DNS分别在[AL98]、[For06]的17章、[PD03]的9.1节、[Tan03]的7.1节中有讨论。

网站

下面的网站与本章中讨论的主题有关：

www.intenic.net关于注册机构的信息

www.ietf.org/rfc.html关于RFC的信息

请求评论

下面是与DNS相关的RFC：

799，811，819，830，881，882，883，897，920，921，1034，1035，1386，1480，1535，1536，1537，1591，1637，1664，1706，1712，1713，1982，2065，2137，2317，2535，2671

本章小结

- 域名系统（DNS）是一个客户端/服务器应用程序，它用一个唯一的用户友好的名字标

识因特网上的每一台主机。

- DNS以层次结构来组织名字空间，以使涉及名字的各种任务分散化。
- DNS可以描述为一棵倒过来的层次结构的树，根节点位于最顶层，最多有128层。
- 树上的每一个节点都有一个域名。
- 一个域定义为域名空间的任何一棵子树。
- 名字空间信息分布在DNS的各个服务器中，每一台服务器对它的区域都具有管辖权。
- 根服务器的区域是整个DNS树。
- 主服务器创建、维护和更新它的区域信息。
- 辅助服务器从主服务器中获取信息。
- 因特网中的域名空间划分为三个部分：通用域、国家域和反向域。
- 共有14种通用域，每个域指明了一个组织类型。
- 每一个国家域指明了一个国家。
- 反向域根据一个给定的IP地址查找一个域名，这称为由地址到名字的解析。
- 名字服务器，即运行DNS服务器程序的计算机，是按层次结构组织的。
- DNS客户端称为解析程序，它将名字映射为地址或者将地址映射为名字。
- 在递归解析中，客户端将它的请求发送到服务器，服务器最终返回一个应答。
- 在迭代解析中，客户端在获得一个应答之前可能会把请求发送给多台服务器。
- 高速缓存是一种将查询的应答存放在存储器中（存放有限的时间）以便以后的请求方便使用的方法。
- 全称域名（FQDN）是一个域名，该域名由主机标号以及向上到达根节点所经过的各级节点的标号组成。
- 部分域名（PQDN）是一种域名，它不全部包括主机到根节点之间所有层次的域名。
- DNS报文有两种类型：查询和响应。
- DNS记录有两种类型：询问记录和资源记录。
- 动态DNS（DDNS）自动更新DNS主文件。
- DNS在报文长度小于512字节时，使用UDP服务；否则，使用TCP。

习题

复习题

1. 对于因特网这样规模的系统，层次结构的名字空间比平面结构的名字空间有哪些优势？
2. 主服务器与辅助服务器之间有什么区别？
3. 域名空间有那三个域？
4. 反向域的用途是什么？
5. 递归解析与迭代解析有什么不同？
6. 什么是全称域名（FQDN）？
7. 什么是部分域名（PQDN）？
8. 什么是区域？
9. 高速缓存是如何提高名字解析的效率？
10. DNS报文主要分为那两种类型？
11. 为什么需要DNS？

练习题

12. 下列哪些是全称域名（FQDN），哪些是部分域名（PQDN）？

a. xxx　　b. xxx.yyy.

c. xxx.yyy.net　　d. zzz.yyy.xxx.edu

13. 下列哪一个是全称域名（FQDN），哪一个是部分域名（PQDN）？

a. mil　　b. edu.

c. xxx.yyy.net　　d. zzz.yyy.xxx.edu

14. 你的系统使用的是哪一个域，通用域还是国家域？

15. 为什么需要DNS系统？什么情况下我们可以直接使用IP地址？

16. 为了查找到目的IP地址，需要DNS服务。DNS需要UDP或者TCP服务，UDP或者TCP需要IP服务，IP需要目的地址。这是一个恶性循环吗？

17. 如果一个DNS域名是*voyager.fhda.edu*，那么它使用了多少个标号？有多少个层次？

18. 一个部分域名（PQDN）必须比对应的全称域名（FQDN）短吗？

19. 一个域名是*hello.customer.info*，这是一个通用域还是一个国家域？

20. 你认为一个递归解析通常会比迭代解析快吗？试解释之。

21. 一条查询报文只有一个询问部分，而对应的响应报文却有几个应答部分，可能出现这种情况吗？

第26章 远程登录、电子邮件与文件传输

因特网的主要任务是向用户提供服务。最流行的应用是远程登录、电子邮件和文件传输。本章讨论这三种应用。我们将在第27章讨论因特网的另一个普遍的应用，访问万维网。

26.1 远程登录

在因特网中，用户希望能够在一个远程网站上运行多个应用程序，而产生的结果能够传送到本地网站。例如，学生们希望能够从他们的家里连接到校园计算机实验室，以便访问应用程序来做家庭作业或完成项目。满足这种需求的一种方法是对每一种需要的服务创建不同的客户/服务器应用程序。现在可用的应用程序有文件传送程序（FTP）和电子邮件等。但是，要对每一个需求编写一个特定客户/服务器程序是不可能的。

更好的方法是使用通用的客户/服务器程序，它让用户能够访问远程计算机上的任何应用程序。换言之，允许用户在远程计算机上登录，在登录后，用户可使用远程计算机提供的服务，并将结果返回到本地计算机上。

TELNET

在这一节中，我们讨论这样一个客户/服务器应用程序TELNET。TELNET是终端网络（Terminal NETwork）的缩写。它是国际标准化组织（ISO）建议的虚终端服务的标准TCP/IP协议。TELNET使用户能够建立一个到远程系统的连接，就好象本地终端连接到远程系统上一样。

> TELNET是一个通用的客户/服务器应用程序。

分时环境

在TELNET设计出来的时候，多数操作系统（如UNIX）工作在分时（timesharing）的环境下。在这样的环境中，一个大型计算机支持许多个用户。用户计算机通过一个终端产生交互，这种终端通常是由键盘、监视器和鼠标组成，甚至一个微机也能够用一个终端仿真程序来模拟一个终端。

登录

在分时环境下，用户是系统的一部分，并具有使用资源的某些权利。每一被授权的用户都有一个标识符，也可能还有一个口令。用户标识符定义用户是系统一部分。要访问系统，用户要使用用户标识符或用户登录名登录到系统。系统还要进行口令检查以防止非授权用户使用资源。图26.1表示了登录过程。

当用户登录到本地的分时系统时，它称为本地登录（local log-in）。用户在终端上键入时，或在工作站上运行终端仿真程序时，他的击键就被终端驱动程序接受。终端驱动程序将字符传递给操作系统。操作系统解释字符的组合，并调用所需的应用程序或实用程序。

当用户想访问远程机器上一个应用程序或实用程序时，他就需要进行远程登录（remote log-in），此时就需要使用TELNET客户程序和服务器程序。用户将其击键发送给终端驱动程序，同时本地操作系统接受这些字符，但并不解释它们。这些字符被送到TELNET客户机，

它将这些字符转换成称为网络虚拟终端（NVT）字符的通用字符集。然后将其传送给本地TCP/IP协议堆栈。

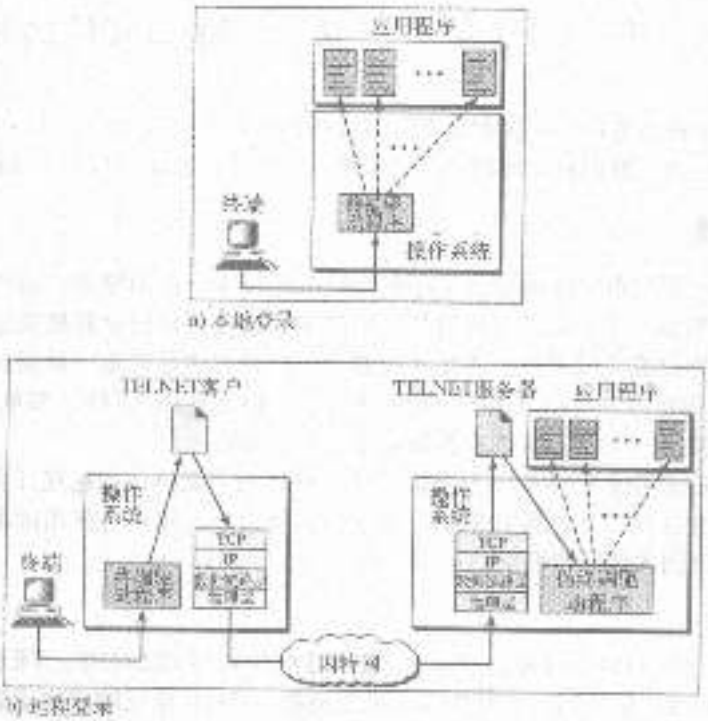


图26.1　本地和远程的登录过程

采用网络虚拟终端（NVT）形式的命令或文本通过因特网传送到远程机器的TCP/IP堆栈，在那里字符传递给操作系统，然后传送给TELNET服务器，TELNET服务器将这些字符转换成远程计算机可以理解的字符。但是，这些字符不能直接传送给操作系统，因为远程的操作系统不能接收来自TELNET服务器的字符，它只能接收来自终端驱动程序的字符。解决方法是增加一个称为伪终端驱动程序的软件块，它将这些字符伪装成好像是从一个终端发来的，然后操作系统将这些字符传送给适当的应用程序。

网络虚拟终端

访问一个远程计算机的过程是很复杂的，这是因为每一个计算机以及其操作系统都接受一个特殊的字符组合作为一个记号。例如，在运行DOS操作系统的计算机中，文件结束标记是Ctrl+z，但在UNIX操作系统中，则是Ctrl+d。

我们现在是和异构系统打交道。如果我们想要访问世界上的任何远程计算机，那么我们必须先知道将要连接的计算机是什么类型，我们还必须安装那个计算机所用的终端仿真程序。TELNET解决这个问题的方法是定义一个通用的接口，称为**网络虚拟终端**（network virtual terminal，NVT）字符集。通过这个接口，客户TELNET将来自本地终端的字符（数据或命令）转换成NVT形式，然后传送给网络。而服务器TELNET将来自NVT形式的数据或命令转换成远程计算机可接受的形式。图26.2表示了这一概念。

NVT字符集　NVT使用二个字符集：一个是数据字符，另一个是控制字符。两者都是8位的字符集。对于数据，它的7个最低位与ASCII是一样的，而最高位是0。在计算机之间发送（从服务器到客户机或从客户到服务器）**控制字符**（control character），NVT使用一个8位的字

符集，其最高位是1。

表26.1列出某些控制字符及其意义。

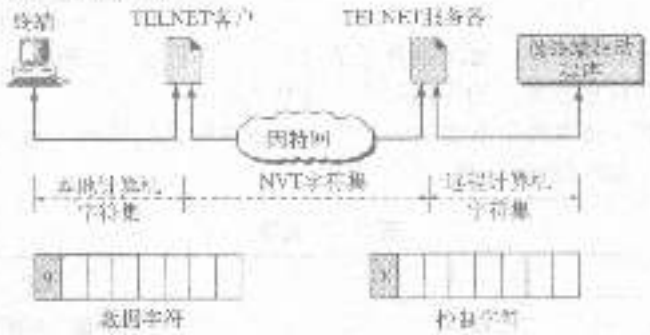


图26.2 NVT的概念

表26.1 一些NVT控制字符

字 符	十进制	二进制	意 义
EOF	236	11101100	文件结束
EOR	239	11101111	记录结束
SE	240	11110000	子选项结束
NOP	241	11110001	无操作
DM	242	11110010	数据标记
BRK	243	11110011	断开
IP	244	11110100	中断过程
AO	245	11110101	异常中止输出
AYT	246	11110110	对方是否还在运行?
EC	247	11110111	擦除字符
EL	248	11111000	擦除行
GA	249	11111001	前进
SB	250	11111010	子选项开始
WILL	251	11111011	同意激活选项
WONT	252	11111100	拒绝激活选项
DO	253	11111101	认可请求请求
DONT	254	11111110	拒绝请求请求
IAC	255	11111111	作为控制解释（下一个字节）

嵌入

TELNET仅使用一个TCP连接，服务器使用熟知端口23，而客户使用一个临时端口。使用同一个连接发送数据和控制字符，TELNET做到这点是将控制字符放入到数据流中。但是，要将数据与控制字符区别开来，在每个控制字符序列的前面要加上一个特殊的控制字符，它称为作为控制解释（IAC）。例如，假定一个用户想要一个服务器在一个远程服务器上显示一个文件（file1），他键入：

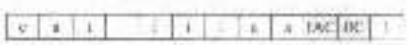
cat file1

然而，假定文件名输入错（如键入filea而不是file1），那么用户可用退格键进行改正：

cat filea<backspace>1

但是，在TELNET的默认实现中，用户不能在本地进行编辑。编辑工作必须在远程服务

器上进行。退格字符被转换成两个远程字符（IAC EC），它嵌入在数据中发送到远程服务器。图26.3表示了发送给服务器的内容。



在远程终端键入

图26.3　嵌入的例

选项

TELNET让客户与服务器在使用服务之前或期间可以协商选项。对具有更加复杂的终端用户，这些选项还提供额外的特性。使用较简单终端的用户只能使用默认的特性。前面讨论的某些控制字符可用来定义选项，表26.2给出一些常用的选项。

表26.2　选项

代　号	选　项	意　义
0	二进制	解释为8个二进制传输
1	回显	将在一端接到的数据回显到另一端
3	抑制继进	抑制继续后面的继进信号
5	状态	请求TELNET的状态
6	计时标记	定义计时标记
24	终端类型	设置终端类型
32	终端速率	设置终端速率
34	行方式	改变线行方式

选项协商　为了使用前面提到的任何选项，首先需要在客户与服务器之间进行选项协商（option negotiation）。为此要使用4个控制字符，这些字符如表26.3所示。

表26.3　NVT选项协商字符集

字　符	十进制	二进制	意　义
WILL	251	11111011	1. 提供激活选项 2. 接受请求激活选项
WONT	252	11111100	1. 拒绝请求激活选项 2. 提供禁止选项 3. 接受禁止请求选项
DO	253	11111101	1. 同意提供激活选项 2. 请求激活选项
DONT	254	11111110	1. 不同意提供激活选项 2. 可接受禁止选项 3. 请求禁止选项

一方可以提供激活或禁止一个选项，如果它有权这样做。对方可以同意或不同意这个选项。为了提供激活选项，提供方发送WILL命令，就是说“我能激活这个选项吗？”另一方或者发送DO命令，就是说“同意”，或者发送DONT命令，就是说“我不同意”。为了提供禁止选项，提供方发送WONT命令，就是说“我不再使用这个选项”。回答必须是DONT命令，就是说“我不要再使用这个选项”。

一方可以请求另一方激活或禁止一个选项。为了请求激活，请求方发送DO命令，就是说“请激活这个选项”。另一方或者可发送WILL命令，就是说“同意”，或者发送WONT命令，就是说“我不同意”。为了请求禁止，请求方发送DONT命令，就是说“请不要再使用这个选项”。回答必须是WONT命令，就是说“我不再使用它”。

例26.1

图26.4表示了一个选项协商的例。在这个例子中，客户希望服务器将发送给服务器的每一个字符回显，换言之，当在用户终端键盘上键入一个字符时，它传送给服务器，并在被处理之前发送回到用户的屏幕上。回显选项必须被服务器激活，因为正是服务器将这些字符发回到用户的终端。因此，客户应当请求服务器使用DO来激活这个选项。这个请求包括3个字符：IAC、DO和ECHO。服务器接受这个请求，并激活该选项。它通过发送3个字符的认可：IAC、WILL和ECHO通知客户机。



图26.4 例26.1回显选项

表26.4 子选项协商的NVT字符集

字 符	十进制	二进制	意 义
SE	240	11110000	子选项结束
SB	250	11111010	子选项开始

子选项协商 某些选项需要附加的信息。例如，要定义一个终端的类型或速率，协商就要包括一个字符串或数字来定义类型或速率。不论是哪种情况，表26.4所显示的二个子选项字符都是进行**子选项协商**（suboption negotiation）所需要的。

例26.2

图26.5表示了一个子选项协商的例子。在这个例子中，客户希望协商终端的类型。



图26.5 子选项协商

操作方式

大多数TELNET实现3种工作方式中的一种：默认方式、字符方式和行方式。

默认方式 如果没有通过选项协商调用其他方式，则使用**默认方式**（defualt mode）。在这种方式中，回显由客户完成。用户键入一个字符，客户机将字符回显到屏幕（或打印机）上，但到整个一行完成之前并不发送它。

字符方式 在**字符方式**（character mode）中，每一个键入的字符从客户机发送给服务器，服务器通常将这个字符回显在客户的屏幕上。如果传输时间较长时（如在一个卫星连接中）这种方式的字符回显可能被延迟时间较长（如在一个卫星连接中）。它还产生网络开销（通信量），因为每一个数据字符必须发送3个TCP段。

行方式 一种新的方式被提出来补偿默认方式的不足。在这种称为**行方式**（line mode）中，即行编辑（回显、字符擦除和行擦除等等）由客户完成，然后客户将整个行发送给服务器。

26.2 电子邮件

最常用的因特网服务之一是电子邮件（e-mail）。因特网的设计者们可能从未预料到该应用程序会如此流行。本章讨论电子邮件的几个构件。

早期因特网时期，用电子邮件发送的报文仅是短小的文本文件，它让人们快速地交换便笺。而当今的电子邮件更加复杂，它允许包括文本、音频及视频的报文，同时还允许发送给一个或多个接收者。本章首先讨论电子邮件一般的构架，它包括三个主要的组件：用户代理、报文传输代理和报文访问代理。然后讨论实现这些组件的协议。

26.2.1 构架

为了说明电子邮件的构架，我们给出四种情况，从最简单情况开始逐步增加其复杂性。其中第四种情况是最常用的电子邮件交换。

第一种情况

在第一种情况中，电子邮件的发送方与接收方是在同一个系统内的用户（或相同的应用程序），它们直接地连接到一个共享一个系统中。管理员为每个用户创建一个邮箱，在邮箱中存储接收到的报文。邮箱是本地硬盘的一部分，即批准某定的一个特殊文件。只有邮箱的拥有者可以访问它。如果一个用户Alice要发一个报文给另一个用户Bob时，Alice运行用户代理（UA）程序准备报文并在Bob的邮箱中存储该报文。报文中有发送方与接收方的邮箱地址（文件名）。Bob在他方便时，用用户代理检索他的邮箱并读取其内容。图26.6表明了这个概念。

这类似于在办公室中职员之间交换传统的便笺。有一个邮件空间，在其中每个职员都有一个用他自己名字命名的邮箱。当Alice要发送一个便笺给Bob时，她写好便笺并将它插入到Bob的邮箱。当Bob检查他的邮箱时，他发现Alice的便笺并读取它。

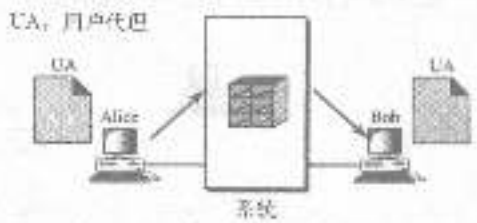


图26.6　电子邮件的第一种情况

> 当电子邮件的发送方和接收方都在同一个系统上，我们仅需要两个用户代理。

第二种情况

在第二种情况中，电子邮件的发送方和接收方是不同系统上两个用户（或应用程序）。此时，我们需要用户代理（user agent，UA）和**报文传输代理**（message transfer agent，MTA）。如图26.7所示。

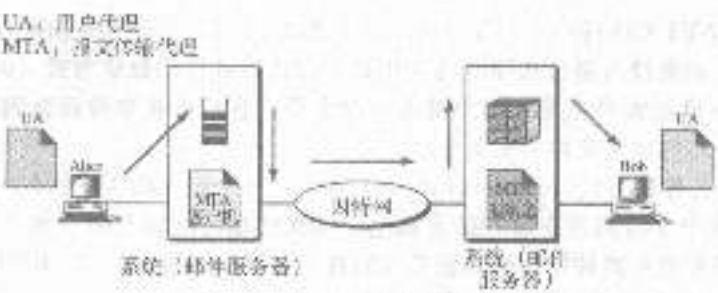


图26.7　电子邮件第二种情况

Alice需要使用用户代理程序发送她的报文到她所在网站的系统。她所在网站的系统（有时称为邮件服务器）使用一个队列存储报文等待发送。Bob也需要一个用户代理程序检索存储

在他的网站邮箱系统中的报文。但是，报文需要通过因特网从Alice的网站发送到Bob的网站。此处需要两个报文传输代理：一个是客户而另一个是服务器。像大多数因特网上的客户/服务器程序一样，服务器程序始终运行，因为它不知道何时用户将会请求一个连接。另一方面，当队列中有发送报文时，可由系统激活客户程序。

> 当电子邮件的发送方和接收方在不同的系统中时，我们需要两个UA和一对MTA（客户机和服务器）。

第三种情况

在第三种情况中，如同第二种情况一样，Bob直接连接到他的系统，但是Alice没有与她的系统直接连接，Alice或是通过点到点广域网连接到系统，譬如用拨号调制解调器、DSL或有线调制解调器，或者她连接到机构中的一个局域网，该机构用一个邮件服务器处理电子邮件，即所有用户都要将它们的邮件发送到这个邮件服务器上。图26.8表示了这种情况。

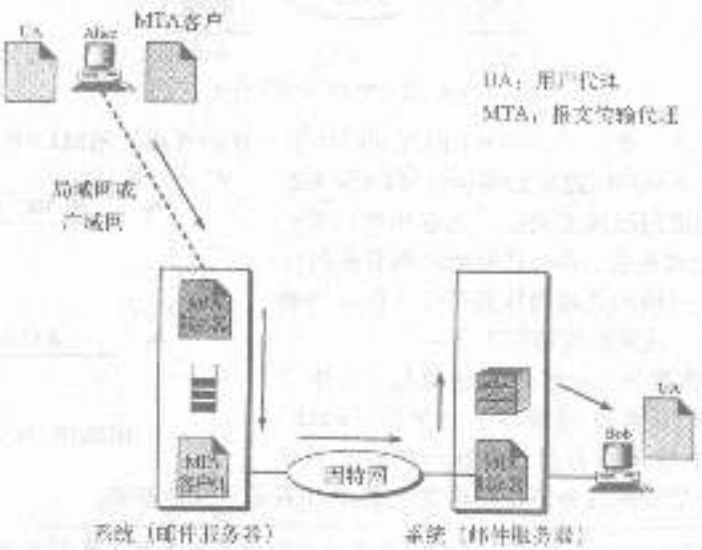


图26.8 电子邮件第三种情况

Alice仍旧需要一个用户代理准备她的报文，然后她需要通过局域网或广域网发送报文。这是通过一对报文传输代理（客户和服务器）完成的。每当Alice有报文要发送时，她调用用户代理，并由它调用MTA客户。MTA客户与系统上一直运行的MTA服务器建立一个连接。Alice网站的系统对接收到的所有报文进行排队。然后，它使用MTA客户向Bob网站的系统发送报文，该系统接收报文并存入Bob的邮箱。Bob在他方便时使用他的用户代理进行检索和读取它。注意：我们需要两对MTA客户机/服务器程序。

> 当发送方通过LAN或WAN连接邮件服务器时，我们需要两对MTA（客户和服务器）。

第四种情况

在第四种和最常用的情况中，Bob也是通过广域网或局域网连接到他的邮件服务器。报文到达Bob的邮件服务器后，Bob需要检索它。此时，需要另一组客户机/服务器代理，我们称它们为报文访问代理（message access agent）。Bob使用MAA客户检索他的报文。客户向始终运行的MAA服务器发送一个请求，并请求报文的传输。这情况如图26.9所示。

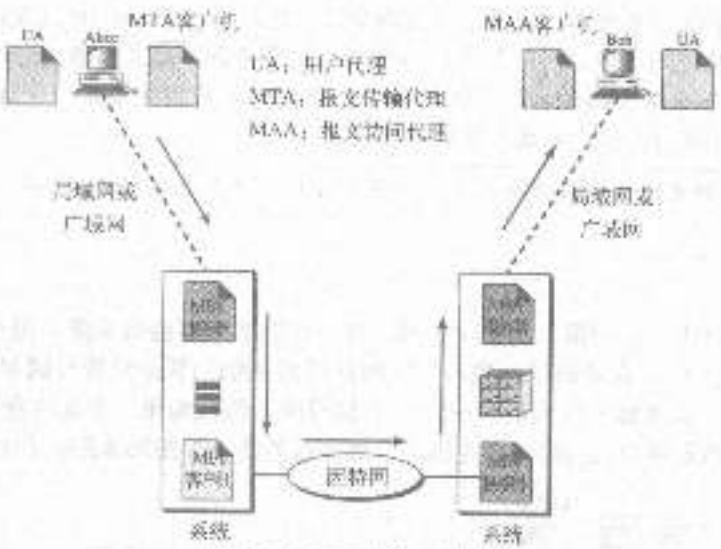


图26.9　电子邮件第四种情况

此处有两个要点。第一点，Bob不能绕开邮件服务器而直接使用MTA服务器。为了直接使用MTA服务器，Bob应该需要始终运行MTA服务器程序，因为它不知道何时报文到达。这意味着如果他通过局域网连接他的系统，Bob必须始终拥有他的计算机。如果他通过广域网连接到他的系统，他必须始终保持连接。今天，这两种情形都不适用。

第二点，Bob需要另一对客户/服务器程序，报文访问程序。因为MTA客户/服务器是一个推（push）程序，客户将报文推入服务器。而Bob需要一个拉（pull）程序。客户需要从服务器拉出报文。图26.10表示了这种差别。



图26.10　电子邮件推和拉

当发送方和接收方通过局域网与广域网连接到邮件服务器时，我们需要两个UA、两对MTA（客户机与服务器）和一对MAA（客户与服务器）。这就是当前最常见的情形。

26.2.2　用户代理

电子邮件系统的第一个组件是用户代理（UA）。它向用户提供服务，使发送和接收一个报文变得更容易。

用户代理提供的服务

用户代理是一个软件包（程序），它由组成、读取、回答和转发报文组成，它也处理邮箱。图26.11表示了一个典型的用户代理的服务。



图26.11　用户代理的服务

组成报文 用户代理帮助用户组成发送出去的电子邮件。大多数用户代理在屏幕上提供一个标准的格式由用户来填写，有些甚至还安装了编辑器，可完成拼写检查、文法检查以及其他想要的复杂的字处理任务。当然，用户可以选择使用他喜欢的文本编辑器或字处理器创建报文和输入报文，或剪切和粘贴报文到用户代理模板上。

读取报文 用户代理第二个功能是读取到来的报文。当用户调用用户代理时，它首先在输入邮箱中检查邮件。大多数用户代理都显示每个接收到邮件的一行摘要，每封电子邮件包含下列字段：

1. 编号字段；
2. 表示邮件的状态如新的、已读而没回复或已读并已回复等的标记字段；
3. 报文的长度；
4. 发送方；
5. 可选择的主题字段。

回答报文 读取报文后，用户可用用户代理对报文做出回答。通常用户代理允许对原始的发送方做回复或对该报文的所有接收方做回复。回答的报文可包含原始的报文（用做快速参考）和新的报文。

转发报文 回答是指对发送方或报文副本的接收者发送报文。而转发报文是指向第三方发送报文。用户代理允许接收方附带或不附带额外的注释将报文转发给第三方。

处理邮箱

通常用户代理创建两个邮箱：收件箱和发件箱。每个邮箱是具有一个特殊格式的文件，它能被用户代理所处理。收件箱保留所有接收到的电子邮件直到被用户删除为止。发件箱保留所有已发送的电子邮件直到被用户删除为止。今天，大多数的用户代理有能力创建个性化的邮箱。

用户代理类型

有两种类型的用户代理：命令驱动型和基于GUI型。

命令驱动型 命令驱动型的用户代理属于早期的电子邮件。在服务器中仍然存在这种类型的用户代理。命令驱动型的用户代理通常是从键盘接收单个字符的命令以执行某项任务。例如，用户可以在命令提示行输入字符r回答报文的发送方，或输入R回答发送方和所有的接收者。命令驱动型用户代理的例有：*mail*、*pine*和*elm*。

命令驱动型用户代理的例有mail、pine和elm。

基于GUI型 现在的用户代理都是基于GUI型。它们包含图形用户接口（GUI）组件，该组件允许用户使用键盘和鼠标与软件进行交互。它们还有一些图形组件，如图标、菜单条和使服务更容易访问的窗口。基于GUI的用户代理有Eudora、微软的Outlook和Netscape。

基于GUI的用户代理有Eudora、Outlook和Netscape。

发送邮件

为了发送邮件，用户通过UA创建邮件，邮件看起来很像邮政邮件，它有一个信封和一个报文（见图26.12所示）。

信封 **信封**（envelope）通常包含发信人的地址和收件人的地址。

报文 报文包含**头部**（hearer）和**主体**（body）。报文头部定义了发信人、收件人、报文的主题及其他信息（如编码类型）。报文的主体包含了由收信人读取的真正信息。

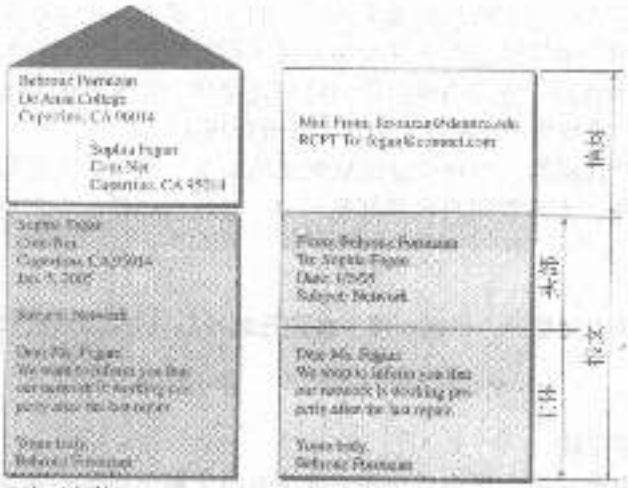


图26.12　电子邮件的格式

接收邮件

用户（或定时器）触发用户代理检查邮箱。如果用户有邮件，UA就用一个通知来告诉用户。如果用户准备读取邮件，则会显示一个列表，其中列表的每一行包含了邮箱中关于一个特定报文的信息概要。这个概要通常包括发信人的邮件地址、主题以及发送和接收此邮件的时间。用户可以选择任何一个报文，在屏幕上显示它的内容。

地址

为了传递邮件，邮件处理系统必须使用具有唯一地址的寻址系统。在因特网中，地址由两部分构成：本地部分（local part）和域名（domain name），并且用符号@分隔开（见图26.13所示）。



图26.13　电子邮件地址

本地部分　本地部分定义一个特定文件的名字，它称为用户邮箱，在用户邮箱中存储了用户所有接收到的邮件，以便用户代理进行读取。

域名　地址的第二部分是域名。一个组织机构通常选择一个或者多个主机用于接收和发送邮件，这些主机有时称为邮件服务器或交换器。分配给每一个邮件交换器的域名或者是来自DNS数据库，或者是一个逻辑名字（例如，该组织机构的名字）。

邮件列表

电子邮件允许用一个名字，即别名（alias），来表示多个不同的电子邮件地址，这称为邮件列表。每当发送一个报文时，系统就将收信人的名字与别名数据库进行比较；如果所定义的别名有一个邮件列表，那么就将报文分开，每一个对应列表中的一项，然后交给MTA处理。如果别名没有邮件列表，那么该名字就是接收地址，并将一个单独的报文传递给邮件传输实体。

MIME

电子邮件有一个简单的结构。但是，它的简单是有代价的。它只能发送使用NVT 7位ASCII格式的报文。换言之，它有一些限制，例如，它不能使用7位ASCII字符支持的语言（如法文、德文、希伯利文、俄文、中文以及日文）。另外，它不能用来发送二进制文件或各

频或视频数据。

多用途因特网邮件扩充（Multipurpose Internet Mail Extension，MIME）是一个辅助协议，它允许非ASCII数据能够通过电子邮件传送。MINE在发送方将非ASCII数据转换成NVT ASCII数据，并将其传递给MTA客户机通过因特网发送出去。在接收方再转换到原来的数据。

可以将MIME想像为一组软件功能，它能够将非ASCII数据转换成ASCII数据，以及进行相反的转换。如图26.14所示。

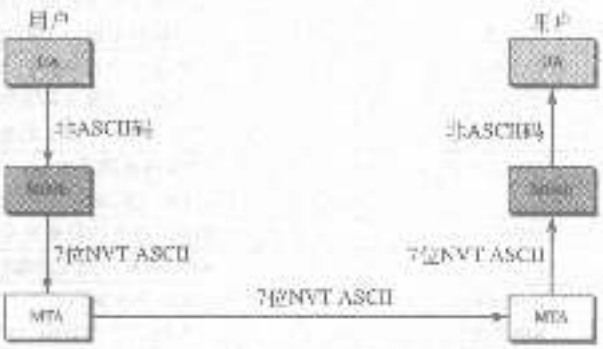


图26.14 MIME

MIME定义了5种头部，将MIME的头加在原来电子邮件的头部以定义转换参数：

1. MIME版本；
2. 内容-类型；
3. 内容-传送-编码；
4. 内容-标识符；
5. 内容-描述。

图26.15表示了MIME的头部，我们将详细讨论每一个头部。

MIME版本　这个头部定义MIME使用的版本，当前版本是1.1。

MIME-版本：1-1

内容-类型　这个头部定义报文主体使用的数据类型，内容类型和内容子类型用一个斜杠分隔开，根据子类型的不同，头部还可包含其他一些参数。

内容-类型：<类型/子类型；参数>

MIME允许7种不同的数据类型，这些类型在表26.5中列出。



图26.15 MIME头部

内容-传送-编码　为了传送的目的这个头部定义将报文编码为一些0和一些1的方法：

Content-Transfer-Encoding:（type）内容-传送-编码：<类型>

表26.6列出了5种类型的编码。

表26.5 MIME中的数据类型和子类型

类型	子类型	说明
正文	普通	无格式
	HTML	HTML格式（见第27章）
多部分	混合	主体包含不同数据类型的有序部分序列
	并行	组无序
	摘要	与混合子类型相似，但默认的是报文/RFC822
	可供选择的	几个部分是同一个报文的不同版本
报文	RFC822	主体是一个封装的报文
	部分	主体是一个较大报文的一部分
	外部主体	主体是另一个报文的参照
图象	JPEG	JPEG格式的图像
	GIF	GIF格式的图像
视频	MPEG	MPEG格式的视频信号
音频	基本	8kHz的单声道语音编码
应用	PostScript	Adobe PostScript
	8位组流	一般的二进制数据（8位字节）

表26.6 内容-传送-编码

类型	说明
7位	NVT ASCII字符和短行
8位	非ASCII字符和短行
二进制	长度不限行的ASCII字符
Base64	6位数据块被编码成8位ASCII字符
引用可打印的	非ASCII字符被编码成等号后面跟随一个ASCII码

内容-标识符 这个头部在多报文环境中唯一地标识整个报文。

内容标识符：id=<内容标识符>

内容-描述 这个头部定义主体是否为图像、音频或视频。

内容-描述：<描述>

26.2.3 报文传输代理：SMTP

实际的邮件传输是由报文传输代理（MTA）完成的。为了发送邮件，系统必须有客户MTA，而为了接收邮件，系统必须有服务器MTA。在因特网中，定义MTA客户机和服务器的形式化协议称为**简单邮件传输协议**（Simple Mail Transfer Protocol，SMTP）。如以前所述，在大多数情况下（第四种情况），使用两对客户/服务器MTA程序。图26.16表示了这种情况中，SMTP管辖的范围。

SMTP在发送方与接收方邮件服务器之间以及两个邮件服务器之间使用两次。稍后我们将看到，在邮件服务器与接收方之间还需要另一个协议。

SMTP只定义了如何来回发送命令和响应。每一个网络有权选择一种实现的软件包。在本节剩下的部分，我们讨论SMTP的邮件传输机制。

命令和响应

SMTP使用一些命令和响应在MTA客户和MTA服务器之间传输报文（见图26.17所示）。

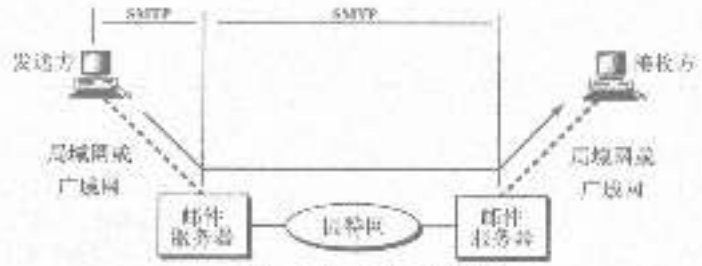


图26.16 SMTP管辖的范围

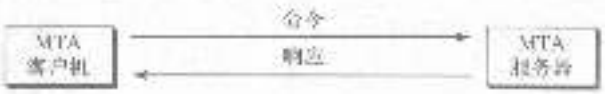


图26.17 命令和响应

每一命令或响应都以一个二字符（回车和换行）的行结束标记来终止。

命令 命令是从客户发送给服务器，命令的格式如图26.18所示。它包括一个关键词，后跟着0个或多个变量。SMTP定义了14个命令，前5个是强制性的，每一种实现都必须支持这5个命令。后面的3个是常用的，并且是着重推荐的。最后的6个很少使用。

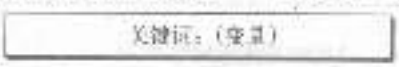


图26.18 命令格式

在表26.7中列出各种命令。

表26.7 命令

关键词	变量	关键词	变量
HELO	发送方的主机名	NOOP	
MAIL FROM	发信人	TURN	
RCPT TO	预期的收信人	EXPN	需要扩展的邮件列表
DATA	邮件主体	HELP	命令名
QUIT		SEND FROM	预期的收信人
RSET		SMOL FROM	预期的收信人
VRFY	需要验证收信人的名字	SMAL FROM	预期的收信人

响应 响应是服务器发送给客户。响应是一个3位数字码，后面可以跟着附加的文本信息。表26.8中列出了各种响应。

如表所示，响应划分为四大类，代码的最高位数字（2，3，4和5）定义大类。

邮件传输阶段

传输一个邮件共有三个阶段：连接建立、邮件传输和连接终止。

例26.3

让我们观察如何直接使用SMTP发送电子邮件和如何模拟本节所描述命令与响应。我们使用TELNET登录到端口25（SMTP的熟知端口号），然后用命令直接发送电子邮件。在这个例子中，forouzanb@adelphia.net发送一封电子邮件给他自己。前面的三行表示TELNET试图与Adelphia邮箱服务器建立连接。

连接建立后，我们输入SMTP命令，然后接收响应，如下所示。用黑色表示命令，而用彩色表示响应。注意：为了更清晰起见，我们增加了用一些"="符号指定的注释行。这些行

不是电子邮件过程的部分。

表26.8 响应

代 码	说 明	代 码	说 明
	肯定的完成答复	452	命令异常中止，存储空间不足
211	系统状态或帮助回答		成功生要定的完成答复
214	帮助报文	500	语法错误，命令无法识别
220	服务就绪	501	参数或变量中有语法错误
221	服务关闭传输通道	502	命令未执行
250	请求命令完成	503	命令序列入不对
251	用户不是本地的，报文将被转发	504	命令暂时未执行
	肯定的中间答复	550	命令未执行，邮箱不可用
354	开始邮件输入	551	用户不是本地的
	瞬态否定的完成答复	552	所请求的动作异常停止，超出存储位置
421	服务不可用	553	所请求动作未发生，不允许使用邮箱名
450	邮箱不可用	554	事务失败
451	命令异常中止，本地错误		

```
$ telnet mail.adelphia.net 25
Trying 68.168.78.100 . . .
Connected to mail.adelphia.net (68.168.78.100).

================ Connection Establishment ================
  220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
  250 mta13.adelphia.net
=================== Mail Transfer ====================
MAIL FROM: forouzanb@adelphia.net
  250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
  250 Recipient <forouzanb@adelphia.net> Ok
DATA
  354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.
================ Connection Termination ================
  250 Message received: adelphia.net@mail.adelphia.net
QUIT
  221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

26.2.4 报文访问代理：POP和IMAP

邮件传递的第一阶段和第二阶段使用SMTP。但是，因为SMTP是一个推（push）协议，第三阶段不使用SMTP。它将报文从客户推入服务器。换言之，大量数据（报文）的方向是从客户到服务器。另一方面，第三阶段需要一个拉（pull）协议，客户机必须从服务器拉出报文。大量数据的方向是从服务器到客户。因此，第三阶段使用报文访问代理协议。

目前有两种报文访问代理协议：邮局协议版本3（POP3）和因特网邮件访问协议版本4（IMAP4）。图26.19显示了在大多数情况下（第四种情况），这两种协议的位置。

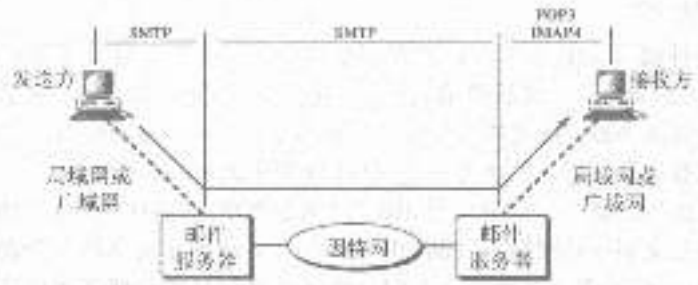


图26.19 POP3和IMAP

POP3

邮局协议版本3（Post Office Protocol version3，POP3）比较简单，但是在功能上受到一定的限制。客户端POP3软件安装在收信人的计算机中，服务器POP3软件安装在邮件服务器中。

用户需要从邮件服务器的邮箱中下载邮件时，客户端发起邮件访问操作。客户端（用户代理）开启与使用服务器110端口之间的TCP连接。然后发送用户名和口令来访问邮箱。用户就可以逐条列出和读取邮件信息了。图26.20说明了一个使用POP3下载邮件的例子。

POP3有两种模式：删除模式和保存模式。在删除模式中，当邮件从邮箱中读取以后就会从邮箱中删除该邮件。在保存模式中，邮件经过读取以后仍然保存在邮箱中。当用户在固定的计算机上工作时，能够在阅读和回复邮件后保存并组织接收到的邮件，此时通常使用删除模式。当用户远离他的主计算机（如在便携机上）访问邮件时，通常应该使用保存模式。此时可以阅读邮件，但是邮件通常仍然会保存在系统中以备日后读取和组织。

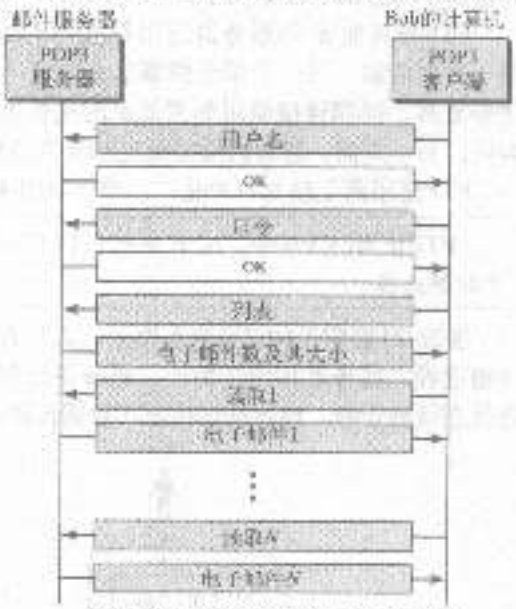


图26.20 POP3中命令与响应的交换

IMAP4

另外一个邮件访问协议是因特网邮件访问协议版本4（Internet Mail Access Protocol version 4，IMAP4）。IMAP4与POP3相似，但有更多特点，功能更强并更复杂。

POP3在某些方面存在缺点，它不允许用户在服务器上组织邮件，用户在服务器上不能有不同的文件夹。（当然，用户能够在自己的计算机上建立文件夹。）同时，POP3不允许用户在下载邮件之前部分地查看邮件的内容。

IMAP4提供下列额外的功能：

- 用户在下载电子邮件之前，可以检查电子邮件头部。
- 用户在下载电子邮件之前，可以读取电子邮件内容中的特定字符串。
- 用户可以部分地下载电子邮件。这在带宽受限制而电子邮件中包含了需要高带宽的多媒体信息时特别有用。
- 用户可以在邮件服务器上创建或删除邮箱，也可以改变邮箱的名字。
- 用户可以在文件夹中创建邮箱的层次结构以用于邮件存储。

26.2.5 基于Web的邮件

电子邮件是一种很常见的应用。目前有一些Web网站对任何访问者都提供服务。两个常用的网站是Hotmail和Yahoo。其思想很简单，Alice的邮件通过HTTP（见第27章）从她的浏览器传输到她的邮件服务器，报文从邮件发送服务器通过SMTP发送到邮件接收服务器。最后，该报文从邮件接收服务器（Web服务器）通过HTTP发送到Bob的浏览器。

最后一个阶段值得注意一下，通常使用HTTP代替POP3或者IMAP4。当Bob需要读取他的邮件时，他会发送报文到Web网站上（例如Hotmail）。Web网站会返回一个表单，由Bob填写，包括登录名和口令。如果登录名和口令匹配，电子邮件会从Web服务器以HTML格式传送到Bob的浏览器。

26.3 文件传输

从一个计算机向另一个计算机传输文件是在联网或互联网环境中最常见的任务之一。事实上，今天在因特网上数据交换的最大量应属于文件传输。在本节中，我们讨论传送文件最常用的协议：文件传输协议（FTP）。

26.3.1 文件传输协议（FTP）

文件传输协议（File Transfer Protocol，FTP）是TCP/IP提供的标准机制，用于从一个主机将文件复制到另一个主机。虽然从一个系统到另一系统传送文件看起来是很简单而且直观，但首先还要解决一些问题。例如，两个系统可能使用不同的文件名约定，两个系统使用不同的方式来表示文本和数据，两个系统具有不同的目录结构。所有这些问题都已经由FTP以一种非常简单巧妙的方法解决了。

FTP与其他客户/服务器应用程序的不同之处在于它在主机之间建立两个连接。一个连接用于数据传输，另一个用于控制信息传输（命令和响应）。将命令和数据传输分开使得FTP的效率更高。控制连接使用非常简单的通信规则。我们需要传输的只是一次一行命令或者一行响应。另一方面，数据传输需要更加复杂的规则，因为传输的数据类型种类多。

FTP使用两个熟知TCP端口：端口21用于控制连接，端口20用于数据连接。

> FTP使用TCP服务。它需要两个TCP连接，熟知端口21用于控制连接，而熟知端口20用于数据连接。

图26.21说明了FTP的基本模型。客户有三个组件：用户接口、客户控制进程和客户数据传输进程。服务器由两个组件：服务器控制进程和服务器数据传输进程。控制连接是在控制进程之间进行的，而数据连接是在数据传输进程之间进行的。

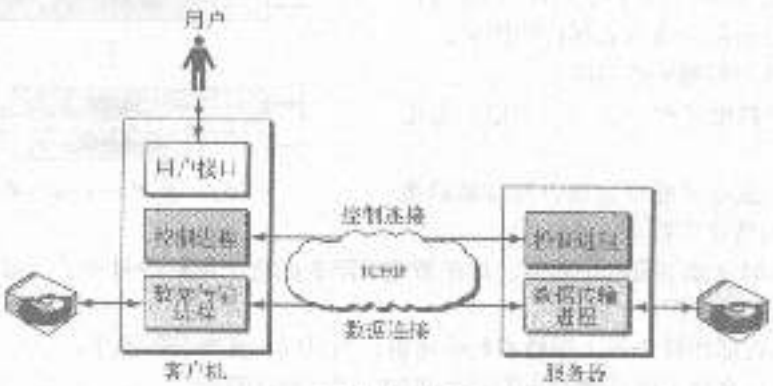


图26.21　FTP

在整个交互的FTP会话期间，**控制连接**（controlconnection）始终处于连接状态。**数据连接**（data control）则在每次传输文件时开启然后关闭。每当牵涉到文件传输的命令被使用时，数据连接就被打开。而当文件传输完毕时连接就关闭。换言之，当用户开始FTP会话时，控制连接就被打开。在控制连接处于打开状态期间，如果传输多个文件，那么数据连接可以打开和关闭多次。

通过控制连接的通信

FTP使用如STMP那样的方法通过控制连接进行通信。它使用7位ASCII字符集（见图26.22所示）。通信是通过命令和响应来完成的。这种简单方法对控制连接是合适的，因为我们一次发送一条命令（或响应）。每一条命令或响应都是一个短行，因此不必担心它的文件格式或文件结构。每一行结束处是两个字符（回车和换行）的行结束记号。

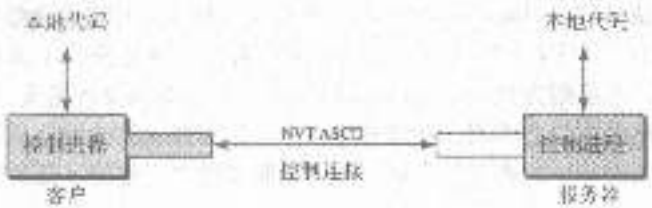


图26.22 使用控制连接

通过数据连接的通信

数据连接的目的和实现与控制连接是不同的。我们通过数据连接来传输文件。在控制连接上发送的命令控制下，在数据连接上进行文件传输。但是，我们要记住，FTP的文件传输表示三件事之一：

- 从服务器将一个文件复制到客户。这称为读取文件，这是在RETR命令监管下完成的；
- 从客户将一个文件复制到服务器。这称为存储文件，这是在STOP命令监管下完成的；
- 从服务器向客户发送目录或文件名列表。这是在LIST命令监管下完成的。应注意是FTP将目录或文件名列表当作一个文件，它在数据连接上发送。

客户必须定义要传送的文件类型、数据结构和传输方式。在通过数据连接传送数据之前，我们通过控制连接准备传输。异构性问题可以通过定义三个通信属性来解决：文件类型、数据结构和传输方式（见图26.23所示）。

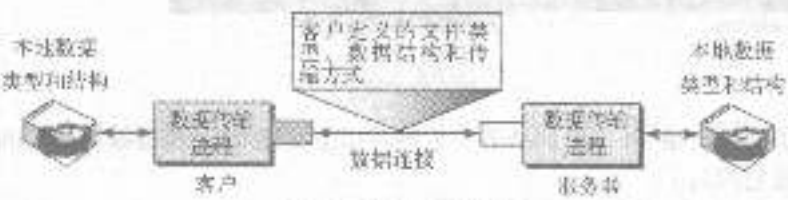


图26.23 使用数据连接

文件类型 FTP能够在数据连接上传送下列文件中的一种：ASCII文件、EBCDIC文件和图像文件。**ASCII文件**（ASCII file）是传送文本文件的默认格式。每个字符使用7位ASCII进行编码。发送方将文件从它自己的表示方式转换成ASCII字符，而接收方从ASCII字符转换成它自己的表示方式。如果连接的一方或两方使用EBCDIC编码（IBM使用的传输格式），则可用EBCDIC编码传送文件。**图像文件**（image file）是传送二进制文件的默认格式。这种文件是作为连续的位流传送而没有解释或编码。它在大多情况下是用来传送二进制文件，如已编

译的程序。

数据结构　FTP可以使用下列数据结构中的一种在数据连接上传送文件：文件结构、记录结构和页面结构。在**文件结构**（file structure）中，文件是连续的字节流。在**记录结构**（record structure）中，文件划分成一些记录，这只能用于文本文件。在**页面结构**（page structure）中，文件划分成一些页面，每一个页面有一个页面号和页面头部。页面可以随机地或顺序地存储或访问。

传输方式　FTP可以使用下列三种传输方式之一在数据连接上传送文件：流方式、块方式和压缩方式。**流方式**（stream mode）是默认方式。数据作为连续的字节流从FTP传递给TCP，TCP负责将数据划分成适当大小的段。如果数据是简单的字节流（文件结构），就不需要文件结束符。在这种情况下，发送方使用文件结束符关闭数据连接。如果数据划分为记录（记录结构），则每一个记录将有一个1字节的记录结束（EOR）字符，而在文件的结束处有一个文件结束（EOF）字符。在**块方式**（block mode）中，数据可以按块从FTP传递给TCP。在这种情况下，每一个块的前面有一个3字节的头部。第一个字节称为块描述符，下面两个字节定义块的大小，以字节为单位。在**压缩方式**（compressed mode）中，如果文件很大，数据可进行压缩。通常使用的压缩方法是游程长度编码。在这种方法中，数据单元的连续出现可用一个出现和重复数来代替。在文本文件中，这通常是空格。在二进制文件中，空格字符常常被压缩。

例26.4

下面给出了使FTP会话读出目录中的项目清单。彩色的行表示来自服务器控制连接的响应。黑色行表示用户发送的命令。黑色背景中带白色的行表示数据传输。

```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls reports
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x    2 3027     411          4096 Sep 24  2002 business
drwxr-xr-x    2 3027     411          4096 Sep 24  2002 personal
drwxr-xr-x    2 3027     411          4096 Sep 24  2002 school
226 Directory send OK.
ftp> quit
221 Goodbye.
```

1. 在创建了控制连接后，FTP服务器在控制连接上发送220（服务器就绪）响应。
2. 客户发送它的名字。
3. 服务器用331（用户名正确，需要口令）响应。
4. 客户发送口令（不显示出来）。
5. 服务器用230（用户登录正确）响应。
6. 客户发送列表命令（ls报告）去寻找报告名目录中的列表。
7. 现在服务器用150响应，并打开数据连接。
8. 然后服务器在数据连接上发送文件列表或目录（作为一个文件）。当整个列表（文件）发送出后，服务器在控制连接上用226（关闭连接）响应。

9. 客户现在有两个选择。它可以使用QUIT命令请求关闭这个控制连接，或者它可以发送另一个命令并始另一个活动（甚至打开另一个数据连接）。在我们的例子中，客户发送一个QUIT命令。

10. 接收到QUIT命令后，服务器用221（服务关闭）响应，然后关闭控制连接。

26.3.2 匿名FTP

要使用FTP，用户就需要在远程服务器上有一个账号（用户名）和口令。有些网站有一组可供公众使用的文件可激活匿名FTP（anonymous FTP）。要使用这些文件，用户不需要有账号和口令。因此可用*anonymous*这个字作为用户名和使用*guest*这个字作为口令。

用户访问这样的系统是受限制的。有些网站只允许用户使用命令的一个子集。例如，大多数网站允许用户复制某些文件，但不允许用户任意查找目录。

例26.5

我们给出了匿名FTP的例子，假定在internic.net上有些公众数据是可用的。

```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
bin
...
...
ftp > close
221 Goodbye
ftp > quit
```

推荐读物

为了深入讨论与本章有关的主题，推荐阅读下列读物和网站。在本书末列出方括号[…]中的参考资料。

读物

远程登录在[For06]的第18章和[Ste94]的第26章中有讨论。电子邮件在[For06]第20章、[PD03]的9.2节、[Com04]的第32章、[Tan03]的7.2节以及[Ste94]的第28章中有讨论。FTP在[For06]的第19章、[Ste94]的第27章和[Com04]的第34章中有讨论。

网站

下面网站与本章的主题有关。

* www.ietf.org/rfc.html有关请求评论的信息。

请求评论

下列请求评论是有关于TELNET：

137，340，393，426，435，452，466，495，513，529，562，595，596，599，669，679，701，702，703，728，764，782，818，854，855，1184，1205，2355

下列请求评论是有关于SMTP，POP和IMAP：

196，221，224，278，524，539，753，772，780，806，821，934，974，1047，1081，1082，1225，1460，1496，1426，1427，1652，1653，1711，1725，1734，1740，1741，1767，1869，1870，2045，2046，2047，2048，2177，2180，2192，2193，2221，2342，2359，2449，2683，2503

下列请求评论是有关于FTP：

114，133，141，163，171，172，238，242，250，256，264，269，281，291，354，385，412，414，418，430，438，448，463，468，478，486，505，506，542，553，624，630，640，691，765，913，959，1635，1785，2228，2577

DNS是在[Al98]，[For06]的第17章，[PD03]的9.1节和[Tan03]的7.1中有讨论。

本章小结

- TELNET是一种客户/服务器应用程序，它使用户能够登录到远程机器上，使该用户能够访问到远程系统。
- TELNET使用网络虚拟终端（NVT）系统在本地系统上对字符进行编码，在服务器机器上，NVT将字符解码为远程机器可接受的形式。
- NVT使用数据字符集和控制字符集。
- 在TELNET中，控制字符嵌入在数据流中，前面加上一个解释为控制（IAC）的控制字符。
- 选项的特点是可以增强TELNET进程。
- TELNET在使用服务之前或之中可以用协商方法在客户和服务器之间设置传输条件。
- TELNET的实现可以工作在默认方式，字符方式和行方式。

  1.在默认方式中，客户向服务器一次发送一行；

  2.在字符方式中，客户向服务器一次发送一个字符；

  3.在行方式中，客户向服务器一次发送一行。
- SMTP，POP3和IMAP4等多个程序支持互特网上的电子邮件服务。
- 在电子邮件中，UA准备报文，创建信封并将报文放入信封中。
- 在电子邮件中，邮件地址包括两部分：本地部分（用户邮箱）和域名，其形式是localpart@domainname。
- 在电子邮件中，多用途因特网邮件扩充（MIME）允许传送多个报文。
- 在电子邮件中，MTA在因特网，局域网或广域网上传输邮件。
- SMTP使用命令和响应在MTA客户和MTA服务器之间传输报文。
- 传输邮件报文的步骤是：

  1. 连接建立；

  2. 邮件传输；

  3. 连接终止。
- 邮局协议版本3（POP3）和互特网邮件访问协议版本4（IMAP4）都是用于从邮件服务器拉出报文的协议。
- 文件传输协议（FTP）是因特网中用于文件传输的一种程序。
- FTP需要两个用于数据传输的连接：一个控制连接和一个数据连接。
- 在两个不同的系统之间FTP使用NVT ASCII进行通信。

- 在真正传输文件之前，客户通过控制连接定义文件类型、数据结构和传输方式。
- 响应是在连接建立期间，由服务器向客户发送的。
- 文件传输的类型有三种：
  1. 从服务器将一个文件复制到客户；
  2. 从客户将一个文件复制到服务器；
  3. 从服务器向客户机发送目录列表或文件名。
- 匿名FTP提供了一个方法，使一般公众能使用远程网站的文件。

习题

复习题

1. 在TELNET中，本地登录和远程登录有什么不同？
2. 在NVT中，控制字符与数据字符如何区别？
3. 在TELNET中，如何进行选项协商？
4. 试描述SMTP所使用的寻址系统。
5. 在电子邮件中，用户代理的任务是什么？
6. 在电子邮件中，MIME是什么？
7. 为什么电子邮件需要POP3或IMAP4？
8. FTP的目的是什么？
9. 试描述FTP两种连接的功能。
10. FTP可以传输的文件类型有哪几种？
11. 三种FTP传输方式是什么？
12. 存储文件与检索文件有什么不同？
13. 什么是匿名FTP？

练习题

14. 试给出从客户TELNET对11110011 00111100 11111111进行二进制传输时所发送的位序列。
15. 如果TELNET使用字符方式，使用命令cp file1 file2将一个名为file1的文件复制成名为file2的文件，在客户和服务器之间要往返多少个字符？
16. 要完成练习15的任务，在TCP级需要发送的最少的位个数是多少？
17. 要完成练习15的任务，在数据链路层这一级（使用以太网）需要发送的最少的位个数是多少？
18. 在练习17中，有用的位与总的位之比是多少？
19. 试解释一个TELNET客户或服务器接收到的下列字符序列（以十六进制表示）：
    a. FF FB 01　　b. FF FE 01
    c. FF F4　　d. FF F9
20. 一个发信方发送无格式文本，试给出MIME头部。
21. 一个发信方发送JPEG文本，试给出MIME头部。
22. 如果TCP已建立一个连接，为什么传送邮件还要建立一个连接？
23. 为什么对匿名FTP会有限制？一个不讲道德的用户可能会怎样做？
24. 试解释为什么FTP没有一个报文格式。

探究题

25. 试给出从默认方式切换到字符方式时，在TELNET客户和服务器之间交换的字符序列。
26. 试给出从默认方式切换到行方式时，在TELNET客户和服务器之间交换的字符序列。
27. 试给出在SMTP中，从aaa@xxx.com到bbb@yyy.com的连接建立阶段。
28. 试给出在SMTP中，从aaa@xxx.com到bbb@yyy.com的报文传送阶段。报文是“Good morning my friend.”
29. 试给出在SMTP中，从aaa@xxx.com到bbb@yyy.com的连接终止阶段。
30. 如果正在传输数据时突然其控制连接出现严重故障，你认为会发生什么问题？
31. 试进行一些研究，找出TELNET提供的一些扩展选项。
32. 另一个登录协议是Rlogin，试比较TELNET与Rlogin。
33. UNIX中有一个称为安全壳（SSH）的更安全的登录协议，试对该协议进行一些研究。

第27章 万维网与超文本传输协议

万维网（World Wide Web，WWW）是分布在世界各地互相链接在一起的信息仓库。WWW具有独特的灵活性、可移植性以及友好的用户界面，它与因特网上提供的其他服务都不同。WWW项目最初是由CERN（欧洲粒子物理研究所）发起，用来创建一个能处理为科学研究用的分布式资源的系统。在本章中，我们首先讨论与Web相关的问题，然后讨论用来从Web中检索信息的协议HTTP。

27.1 体系架构

现在，WWW是一个分布式的客户/服务器服务，在这种方式下，客户机用浏览器能够使用服务器提供的服务。然而，提供的服务是分布在许多称为站点的位置上，如图27.1所示。

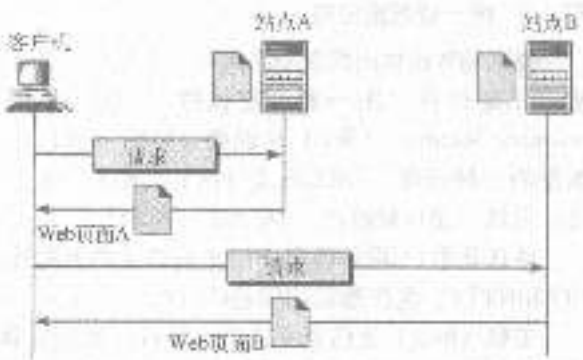


图27.1 WWW的体系架构

每一个站点拥有一个或多个文档，也就是所谓的Web页面，每一个Web页面可以包含一个本地站点或其他站点的页面链接。这些页面可以通过浏览器找到并浏览。让我们详细观察图27.1所示的情形。客户端需要查找它知道在站点A上的一些信息，它用浏览器（一个用来获取Web文档的应用程序）发送请求，请求中还包含了站点的地址和称为URL的Web页面信息，关于URL稍后会对它进行讨论。在站点A的服务器找到文档并将它发送给客户机。当用户浏览文档时，她发现有关其他文档的一些参考，包括在站点B上的一个Web页面。这个参考有新站点的URL，用户也有兴趣看这个文档。客户向新的站点发送另一个请求，这样就得到了新的页面。

27.1.1 客户（浏览器）

许多开发商提供了商用的浏览器来解释和显示Web文档，而所有这些浏览器几乎都使用相同的体系架构。每一种浏览器（browser）通常由三部分构成：一个控制程序，客户协议和一些解释程序。控制程序从键盘或者鼠标接收输入，并使用客户端程序访问文档。获取文档以后，控制程序使用解释程序将文档显示在屏幕上。客户机协议可以是前面所述的协议之一，如FTP或者HTTP（在本章后面会进行讨论）。解释程序可以是HTML，Java或者JavaScript，这取决于文档的类型。在本章后部分，我们将基于文档的类型讨论如何使用这些解释程序（见图27.2所示）。

27.1.2 服务器

Web页面存储在服务器中。每次当客户机请求到达时，相应的文档就发送给客户。为了提高效率，服务器通常把请求的文件存储在内存的高速缓存中；访问内存比访问硬盘要快。服务器也可以通过多线程、多处理器提高效率。在这种情况下，服务器可以一次应答更多的请求。

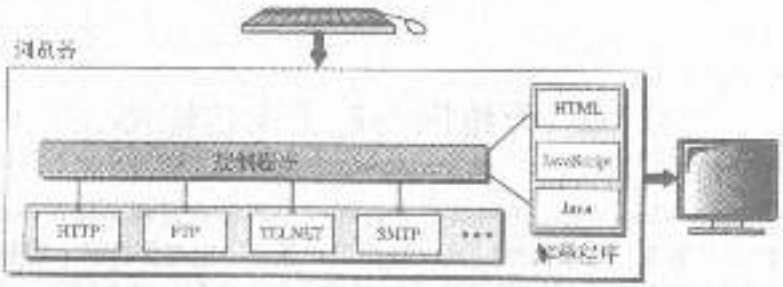


图27.2　浏览器

27.1.3　统一资源定位符

要访问Web页面的客户必须有一个地址，为了更方便地访问分布在世界各地的文档，HTTP使用了定位符。统一资源定位符（uniform resource locator，URL）是说明因特网上信息类型的一种标准。URL格式包括4个部分：协议、主机、端口和路径。（见图27.3所示）。



图27.3　URL

协议是客户/服务器程序用来检索文档的程序。许多不同的协议都能够检索文档，其中有FTP和HTTP。现在最常用的是HTTP。

主机（host）是信息所在的计算机，尽管计算机的名字可能是一个别名。Web页面通常存储在计算机中，计算机通常会赋予一个以字符“www”开头的别名。这并不是强制性的，因为主机可能是驻留Web页面的计算机的任何名字。

URL可以选择是否包含服务器的端口号。如果包含了端口，则将它插入在主机和路径之间，用冒号将它与主机分开。

路径（path）是信息所存储的文件的路径名。注意：路径可以包含斜杠。在UNIX操作系统中，斜杠用来分隔目录和子目录及文件。

27.1.4　Cookies

万维网最初被设计成一个无国界的实体。客户端发送请求，服务器响应。它们之间的关系就结束了。最初设计的WWW，公开地检索可用的文档，完全符合这个目标。现在Web还有其他的功能，如下所示：

1. 有些网站只允许注册客户才能访问；
2. 网站作为电子商店，允许客户在商店内浏览，选择需要的商品，把它们放入电子购物车内，最后使用信用卡付费；
3. 有些网站是门户网站，用户可以选择他想看的Web页面；
4. 有些网站仅作为广告。

为了实现这些目的，cookie机制就应运而生。我们在第23章中的传输层讨论了cookies的使用，现在我们讨论它在Web页面中的使用。

Cookies的创建和存储

Cookies的创建和存储与实现有关，然而，它的原理是相同的。

1. 当服务器从客户端接收到请求后，它将有关客户端的信息存储在文件或字符串中。这些信息可能包含客户端的域名、cookie的内容（服务器收集到的关于客户端的信息，如主机名、注册号等）、时间戳、以及与实现有关的其他信息。

2. 服务器在响应中包含了它发送给客户机的cookie；

3. 当客户端接收到响应后，浏览器在cookies目录中存储cookie，并根据域服务器名进行分类。

Cookies的使用

当客户向服务器发送请求时，浏览器在cookie目录中查询是否有从那个服务器发送过来的cookie，如果有，则在请求中包含这个cookie。当服务器接收到这个请求后，它就知道了这是一个老的客户，而不是新的。注意：cookie的内容从来不让浏览器读取或者透露给用户，cookie只由服务器创建并改回。现在让我们分析如何使用cookie来实现上面提到的4个功能。

1. 当客户端第一次注册时，网站就向客户端发送一个cookie，网站通过这种方式限制注册用户的访问。只有那些能够发送正确cookie的客户才能被允许今后重复的访问。

2. 网上电子商店（电子商务）可以为客户端的购物者使用cookie。当客户端选择商品，并放入到购物车中后，包含了这些商品信息的（如它的数量、单价）cookie就被发送到浏览器。如果客户端选择第二个商品，cookie就被新的选择信息更新，依次类推。当客户端结束购物并准备付账离开时，就检索最终的cookie，然后计算出总的费用。

3. Web门户以同样的方式使用cookie。当用户选择她最喜爱的页面时，就生成一个cookie并发送到浏览器。当这个网站再次被访问时，这个cookie就发送给服务器说明这个客户端要查找什么页面。

4. cookie也用来作为广告代理。广告代理能够将大字标题广告放置在用户经常访问的网站的主页面上。广告代理仅提供指出大字标题广告地址的URL，而不是大字标题广告本身。当用户访问网站主页并点击广告公司的图标时，一个请求就发送给了广告代理；广告代理就发送一个大字标题广告，如GIF文件，同时也包含了一个含有用户ID的cookie。这个大字标题广告将来的任何使用都会加入到一个分析用户Web行为的数据库中。广告代理已经收集了用户的爱好，并能够将这些信息卖给其他的组织。这种cookie的使用方法引起很多争议，但愿今后能引入一些新的法规来保护用户的隐私信息。

27.2 Web文档

WWW中的文档可以分为三大类：静态文档、动态文档和活动文档。基于文档内容的时间确定分类。

27.2.1 静态文档

静态文档（static document）是固定内容的文档，它由服务器创建和并存储在服务器中。客户端只能获取文档的一份副本。换句话说，文件的内容在创建时就已经确定了，而不是在它被使用时。当然，服务器中的内容也是可以改变的，但用户不能改变它。当客户端访问静态文档时，服务器会发送一份文档的副本，用户可随后使用浏览器程序来显示这个文档（见图27.4所示）。

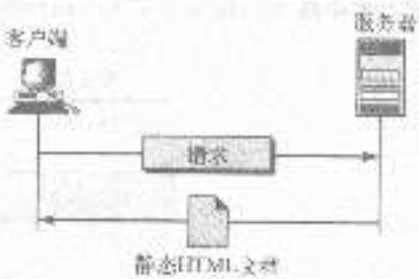


图27.4 静态文档

超文本标记语言

超文本标记语言（Hypertext Markup Language，HTML）是用于创建Web页面的语言。术语标记语言，来自于书籍出版业。一本书在排版和印刷之前，文字编辑阅读手稿，在上面做

一些标记。这些标记告诉具体的工作人员如何处理文本。例如，文字编辑想要一行中的某一部分加粗字体，则会在这一部分上划一条波浪线。同样道理，Web页面的数据也需要进行格式化，以便浏览器进行解释。

我们用一个例子来阐述这一想法。要使一个文本的部分文本使用HTML显示为粗体，必须在文本的开始和结尾处放置粗体标签（标记），如图27.5所示。

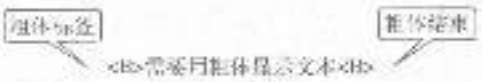


图27.5　粗体标签

两个标签<B>和</B>是浏览器指令。当浏览器检查到这两个标记时，就知道两者之间的文本需要加粗显示（见图27.6所示）。



图27.6　粗体标签的效果

如HTML这样的标记语言允许我们在文件中嵌入格式化的指令，这些指令存储在文本中。在这种方式下，任何浏览器能够读取这些指令，并且根据特定的工作站对文本进行格式化处理。读者可能要问，为什么不使用字处理软件的格式功能来创建和存储格式化的文本？答案是不同的字处理软件使用不同的技术和过程。例如，假设一个用户使用Macintosh计算机创建了格式化的文本并存储在Web页面中。另一个使用IBM计算机的用户则不能接收这一Web页，因为两台计算机使用不同的格式化处理过程。

HTML在正文和格式化指令中都只使用ASCII字符。通过这种方式，每一台计算机能够按ASCII文档的形式接收整个文档，正文是数据，而格式化指令由浏览器用来对数据进行格式化处理。

Web页面由两部分组成：头部和主体。头部是Web页面的第一部分，它包含页面标题和浏览器要用到的其他参数。页面的真正内容在主体内，包含了正文和标签。然而，正文是页面中包含的真正信息，而标签定义了文档的外观。每个HTML标签是一个名字，后面跟着可选的属性列表。所有的内容都嵌入在小于符号和大于符号之间（<和>）。

如果有属性的话，则属性后面跟着一个等于符号及属性的值，一些标签可以单独使用，有些必须成对使用。成对使用的标签称为起始标签和结束标签。起始标签由标签名开始，可以有属性和值；结束标签不可以具有属性和值，但必须在标签名前加上一个斜杠。浏览器根据文本中嵌入的标签对文本的结构做出判断。图27.7说明了标签的格式。



图27.7　起始标签和结束标签

通常使用的标签类型是文本格式标签，例如<B>和</B>，这使文本为粗体；<I>和</I>，这使文本为斜体；<U>和</U>，这使文本加下划线。

另外一个感兴趣的标签类型是图像标签。例如数字化照片或图像这样的非文本信息，这些并不是HTML文档的物理组成部分，但我们可以用图像标签来指明照片和图像文件。图像

标签定义了要检索图像的地址（URL），并且也指明了在找到图像之后如何插入到文本中。我们可以从许多属性中选择。最常用的是SRC（源），它定义源（地址），以及ALIGN，它定义了图像的对齐。属性SRC是必须有的。大多数的浏览器接受GIF或者JPEG格式的图像。例如，下面的标签可以在目录/bin/images里找到存储为image1.gif的图像：

<IMG SRC="/bin/images/image1.gif" ALIGN=MIDDLE>

第三类感兴趣的类型是超链接标签。它将文档链接起来。通过一个称为锚的机制，任何一个项目（文字、短句、段落、或者图像）都可以指向另一个文档。锚用标签<A...>和</A>来定义。而锚定的项目通过URL指向另一个文档。当文档显示时，锚定的项目出现下划线、闪烁、或者用粗体表示。用户可以点击锚定项目来访问另一个文档，这个文档不一定与原先的文档存储在同一个服务器中。参照短句嵌入在起始标签和结束标签之间，起始标签可以有多个属性，但其中的一个必须是HREF（超链接参考），它定义了链接文档的地址（URL）。例如，对一本书的作者的链接可以是：

<A HREF="http://www.deanza.edu/forouzan"> Author </A>

在文本中显示的是单词"Author"，但用户可以通过点击它访问该作者的Web主页。

27.2.2 动态文档

动态文档（dynamic document）是在有浏览器请求该文档时，才由Web服务器创建的。当请求到达时，Web服务器运行一个应用程序或者脚本来创建动态文档。服务器返回程序或者脚本的输出作为对浏览器文档请求的响应。因为一个新的文档是根据每一个请求而创建的，因此动态文档的内容是根据请求的不同而发生变化。动态文档非常简单的例子是获取服务器的时间和日期。时间和日期是一种每时每刻都在变化的动态信息。客户机可以请求服务器运行一个程序，如UNIX中的*date*程序，并将程序的结果发送给客户机。

公共网关接口（CGI）

公共网关接口（Common Gateway Interface，CGI）是创建和处理动态文档的一种技术。CGI是一组标准，它定义了如何编写动态文档，如何将数据传递给应用程序，以及如何使用输出结果。

CGI不是一种新的语言，它允许程序员使用多种语言中任何一种，如C，C++，Bourne 外壳、Korn 外壳、C外壳、TCL或者Perl。CGI只是定义了一组程序员应该遵循的规则和术语。

在CGI中，术语"公共"是指这个标准定义的规则集对任何语言或者平台都是通用的。术语"网关"在这里表示CGI程序可以用来访问其他资源，如数据库和图形包等。术语"接口"在这里是指有一组预先定义的术语、变量、调用等等都可以在任何CGI程序中使用。最简单的CGI程序的形式是用支持CGI的一种语言书写的代码。任何程序员，只要有程序设计的思想，并且知道上面提到的任何一种语言的语法，就能够编写出简单的CGI程序。图27.8说明了使用CGI技术创建动态程序的步骤。

输入（Input）在传统的编程设计中，当程序执行时，可以将参数传递给程序。参数传递机制允许编程设计者编写出在不同情况下能够使用的通用程序。例如，一个通用的复制程序可以编写为将任何文件复制到另一个文件中。用户可以使用这个程序将名为x的文件复制为另一个名为y的文件时，需要传递x和y作为参数。

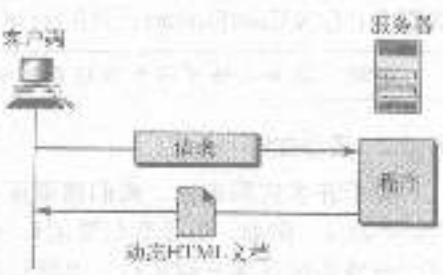


图27.8 使用CGI技术的动态文档

从浏览器到服务器的输入是通过使用一个表单（form）发送的。如果在表单中的信息比较短（如一个字），那么它可以通过一个问号附加在URL后面。例如，下面的URL携带了表单的信息（一个数值23）：

http://www.deanza/cgi-bin/prog.pl?23

当服务器接收到这个URL时，它利用URL问号前面的部分访问将要执行的程序，然后解析问号后面的部分（23）作为客户端发送的输入信息。服务器将这个字符串存储一个变量中，当CGI程序执行时，它可以读取这个值。

如果从浏览器来的输入信息太长而无法放入查询字符串中时，浏览器可以请求服务器发送一个表单，浏览器能够用输入数据填入这个表单中，并发送给服务器。表单中的信息可以作为CGI程序的输入。

输出（output）CGI总体的思想是在服务器站点执行CGI程序，并把输出结果发送给客户（浏览器）。输出的结果通常是普通文本或者是带有HTML结构的文本；然而，输出结果也可以是其他形式的对象。它可以是图像或者二进制数据、状态码、通知浏览器用来对结果高速缓存的指令、或者通知服务器用来发送现有的文档而不是真正输出的指令。

为了让客户端知道发送文档的类型，CGI程序必须创建头部。实际上，CGI程序的输出总是包括两个部分：头部和主体。头部与主体之间用空行隔开，这意味着任何一个CGI程序首先创建头部，然后是一个空行，最后是主体。尽管在浏览器屏幕上不显示头部和空行，但浏览器要使用头部对主体进行解释。

动态文档的脚本技术

CGI技术的问题是低效率，如果要创建的动态文档部分的内容是固定的，并且不随请求的不同而发生变化，这时候低效率就能体现出来。例如，假设我们需要检索一个特定汽车品牌的备用零件列表，它们的有效性及价格。尽管有效性和价格经常会发生变化，但是名字、描述以及零件的图片是固定不变的。如果我们使用CGI，每次一个请求生成后，程序必须创建一个完整的文档。创建一个包含文档固定部分的文件的解决方法是使用HTML，并嵌入脚本和源码，服务器运行的是提供可变化的有效性及价格部分。图27.9说明了这个概念。

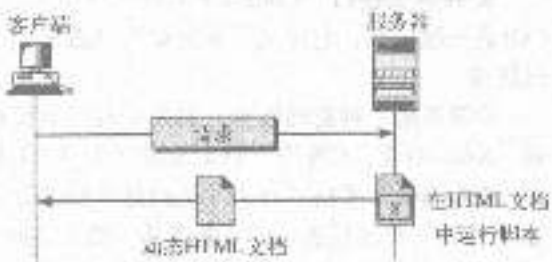


图27.9　使用服务器端脚本的动态文档

一些技术已经被引入到使用脚本来创建动态文档。其中最常见的是：超文本预处理器（Hypertext Preprocessor，PHP），它使用Perl语言；Java服务器网页（Java Server Pages，JSP）技术，它使用Java语言作为脚本；活动服务器网页（Active Server Pages，ASP）技术，这是微软的产品，它使用VB语言作为脚本；以及ColdFusion，它在HTML文档中嵌入了SQL数据库查询。

> 有时，动态文档可以称为服务器端的动态文档。

27.2.3　活动文档

对于许多应用来说，我们需要程序或者脚本在客户端运行。这些称为**活动文档**（active document）。例如，假设我们要运行在屏幕上创建动画图像与用户进行交互的程序。很明显，这个程序需要在客户端运行，也就是在动画和交互发生的地方运行。当浏览器请求活动文档时，服务器发送一份文档或者一个脚本的副本。然后文档就在客户端（浏览器）运行。

Java小应用程序

使用Java小应用程序（applets）是创建活动文档的一种方式。Java是一种高级编程语言，一个实时运行环境和一个类库的组合。程序员可以使用Java编写活动文档（一个小应用程序），并在浏览器中运行。它也可以是不使用浏览器的独立运行的应用程序。

applet是在服务器端用Java编写的应用程序。它是已经编译的，并且是可运行的程序。文档是字节码（二进制）格式的。客户端处理器（浏览器）创建了这个applet的实例并开始运行。用浏览器运行Java applet有两种方式。第一种方式：浏览器可以直接地在URL中请求java applet程序，并以二进制格式接收applet。第二种方式：浏览器可以找到并运行以applet地址嵌入为标签的HTML文件。图27.10说明了在第一种方式下，如何使用java applet，第二种方式是类似的，但需要两个事务。

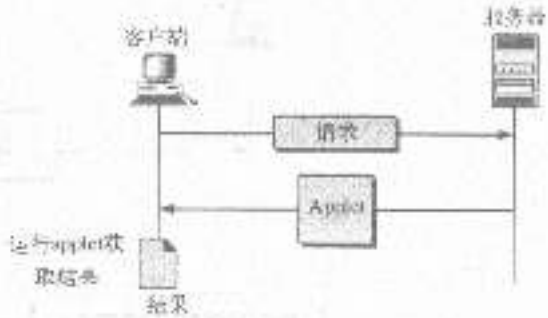


图27.10 用Java applet活动文档

JavaScript

在动态文档中的脚本思想也可以用在活动文档中。如果文档中的活动部分很少，那么它可以用脚本语言编写，然后在客户端运行并同时解释。脚本是源码形式（文本），而不是二进制格式。在这种方式下使用的脚本技术一般来说就是JavaScript。JavaScript（Java脚本）是与Java完全不同的语言，它是为这个目的而发展起来的高级脚本语言。图27.11说明了如何用JavaScript创建活动文档。

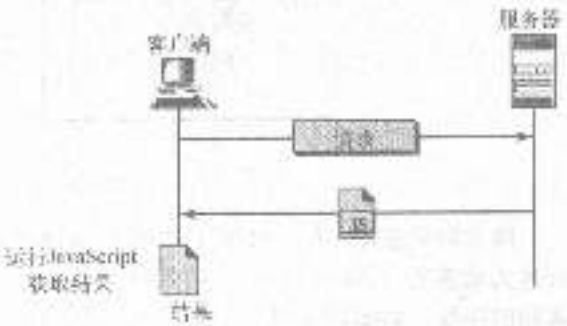


图27.11 使用客户端脚本的活动文档

> 活动文档有时候可以认为是客户端的动态文档。

27.3 超文本传输协议（HTTP）

超文本传输协议（Hypertext Transfer Protocol，HTTP）主要用于万维网上存取数据的协议。HTTP在功能像是FTP和SMTP的组合。它与FTP相似，是因为它能够传输文件并使用TCP服务。但是，它比FTP更简单，因为它只使用了一个TCP连接。它没有单独的控制连接，在客户机和服务器之间只传输数据。

HTTP也与SMTP相似，是因为在客户端和服务器之间传输的数据与SMTP报文很相似。此外，报文格式由类似MIME的头部控制。HTTP与SMTP不同的是，HTTP报文不是供人阅读的，它们是由HTTP服务器和HTTP客户端（浏览器）读取并解释。SMTP报文是存储转发的，而HTTP报文需要即时发送。从客户端发送到服务器的命令嵌入在请求报文中。而请求文件的内容或其他信息嵌入在响应报文中。

> HTTP在熟知端口80上使用TCP服务。

27.3.1 HTTP事务

图27.12说明了客户端和服务器之间的HTTP事务。尽管HTTP使用TCP服务，但是HTTP本身

是一种无状态协议。客户端通过发送请求报文初始化这个事务，服务器通过发送响应进行回复。

报文

请求和响应报文的格式是相似的，图27.13对两者都做了说明。请求报文由一个请求行、一个头部构成，有时还可能包括主体。响应报文由状态行、头部构成，有时还可能包含主体。

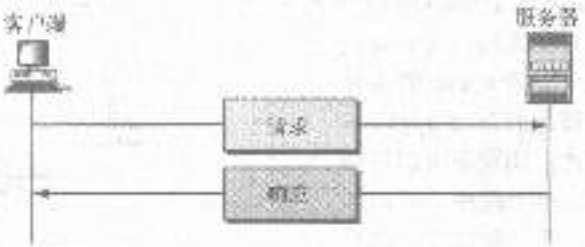


图27.12　HTTP事务

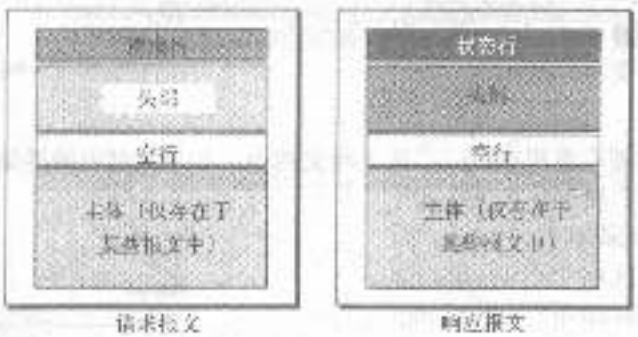


图27.13　请求与响应报文

请求和状态行　在请求报文中的第一行称为**请求行**（request line），在响应报文中的第一行称为**状态行**（status line）。两者有一个共同的字段，如图27.14所示。

- **请求类型**。这个字段在请求报文中使用。在HTTP 1.1版本中，定义了几种请求类型。**请求类型**（request type）分类成几种方法，如表27.1所示。

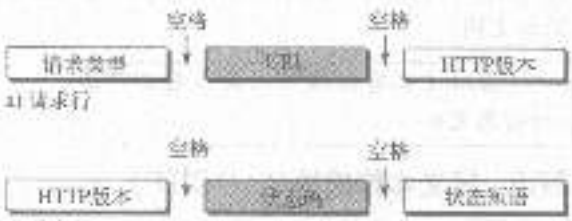


图27.14　请求和状态行

表27.1　方法

方　法	动　作	方　法	动　作
GET	向服务器请求文档	TRACK	对最后的请求
HEAD	请求关于文档的信息，而不是文档本身	CONNECT	预留
POST	从客户端向服务器发送信息	OPTION	询问有关可用的选项
PUT	从服务器向客户端发送文档		

- **URL**。在本章的前面部分，我们已经讨论了统一资源定位符（URL）。
- **版本**。HTTP目前最常用的版本是1.1。
- **状态码**。这个字段在响应报文中使用。**状态码字段**（status field）与在FTP、SMTP协议中的字段相似，它由三个数字组成。在100系列的代码只代表一个报告；在200系列的代

码表示这是一个成功的请求；在300系列的代码表示把客户端重定向到另一个URL；在400系列的代码表示在客户端发生错误；在500系列的代码表示错误发生在服务器端。表27.2列出了常用的代码。

- 状态短语。这个字段在响应报文中使用。它用文本格式解释了状态码。表27.2也给出了状态短语。

表27.2 状态码

代 码	短 语	描 述
		提供信息
100	Continue	请求的初始部分已经收到，客户端可以继续它的请求
101	Switching	服务器回答客户的请求，切换到更新头部所定义的协议成功
200	OK	请求成功
201	Created	创建了一个新的URL
202	Accepted	请求已经接受，但它不能立即响应
204	No content	主体中没有内容
		重定向
301	Moved permanently	服务器已不再使用所请求的URL
302	Moved temporarily	请求的URL已经暂时移开
304	Not modified	文档还没有发生改
		客户端错误
400	Bad request	在请求中有语法错误
401	Unauthorized	请求缺乏合适的授权
403	Forbidden	服务被拒绝
404	Not found	没有找到文档
405	Method not allowed	URL不支持该方法
406	Not acceptable	不接受这种格式的请求
		服务器错误
500	Internalserver error	服务器有错误，例如系统崩溃
501	Not implemented	请求的动作不能完成
503	Service unavailable	服务暂时不可用，但将来可以接再次请求

头部（header） 头部用于客户端和服务器之间交换附加的信息。例如，客户端可以请求以某种特殊格式发送文件，或者服务器发送与文档有关的附加信息。头部由一行或者多行组成，每一行由一个头名字、一个冒号、一个空格和头值组成（见图27.15所示）。在本节的后面，我们将举一些例子说明头部的各行。头部的各行属于下列四类之一：**通用头部**（general header）、**请求头部**（request header）、**响应头部**（response header）和**实体头部**（entity header）。请求报文只能包含通用、请求和实体头部。响应报文只能包含通用、响应和实体头部。

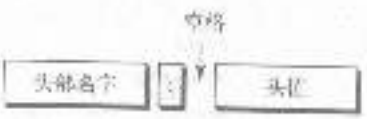


图27.15 头部格式

通用头部给出关于报文的通用信息，并可以在请求和响应报文中都存在。表27.3列出了一些通用头部及其说明。

表27.3　通用头部

头　部	说　明
Cache-control	定义了关于高速缓存的信息
Connection	说明连接是否应关闭
Date	给出现在的日期
MIME-version	说明MIME的使用版本
Upgrade	规定首选的通信协议

请求头部只能在请求报文中存在。它确定客户端配置和客户端首选的文档格式。表27.4列出了一些请求头部及其说明。

表27.4　请求头部

头　部	说　明
Accept	给出客户端能够接受的媒体格式
Accept-charset	给出客户端可以处理的字符集
Accept-encoding	给出客户端可以处理的编码方案
Accept-language	给出客户端可以接受的语言
Authorization	给出客户端有权访问
From	给出用户的电子邮件地址
Host	给出主机及服务器的端口号
If-modified-since	只有在指定的日期更新时，才发送文档
If-match	只有与给定的标记匹配时，才发送文档
If-non-match	只有与给定的标记不匹配时，才发送文档
If-range	只发送丢失的部分文档
If-unmodified-since	如在指定日期之后未改变，则发送文档
Referrer	给出被链接文档的URL
User-agent	标识客户端程序

响应头部只能出现在响应报文中。它确定了服务器的配置和关于请求的特定信息。表27.5列出了一些响应头部及其说明。

表27.5　响应头部

头　部	说　明
Accept-range	说明服务器是否接受客户端请求的范围
Age	给出文档使用年限
Public	给出支持的方法列表
Retry-after	规定一个日期，在这个日期之后服务器是可用
Server	给出服务器名及版本号

实体头部提供了关于文档主体的有关信息。尽管多数情况下它只出现在响应报文中，但在有些请求报文中，如POST和PUT方法，也会包含一个主体且使用这种类型的头部。表27.6列出了一些实体头部及其说明。

表27.6 实体头部

头 部	说 明
Allow	列出URL可以使用的方法方法
Content-encoding	指定编码方案
Content-language	指定语言
Content-length	给出文档的长度
Content-range	指定文档的范围
Content-type	指定媒体类型
Etag	给出实体的标记
Expires	给出内容改变的日期和时间
Last-modified	给出上次改变的日期和时间
Location	告诉新建或移动后文档的位置

主体（body）可以出现在请求和响应报文中。通常，它包含要发送或接收到的文档。

例27.1

这个例子是检索文档。我们用GET方法检索路径为/usr/bin/image1的图像。请求行给出方法（GET）、URL和HTTP版本（1.1）。其头部有两行，以表明客户端可以接受GIF和JPEG格式的图像。请求报文没有主体。响应报文包含状态行和四行的头部。这些头部行定义了日期、服务器、MIME版本和文档的长度。文档的主体位于头部之后（图27.16所示）。

例27.2

这个例子是客户端要向服务器发送数据，我们使用了POST方法。请求行说明了方法（POST）、URL和HTTP版本（1.1）。其头部有4行，请求主体中包含了输入信息。响应报文包含了状态行和四行的头部。被创建的文档是一个CGI文档，它包含在响应报文的主体中（如图27.17所示）。

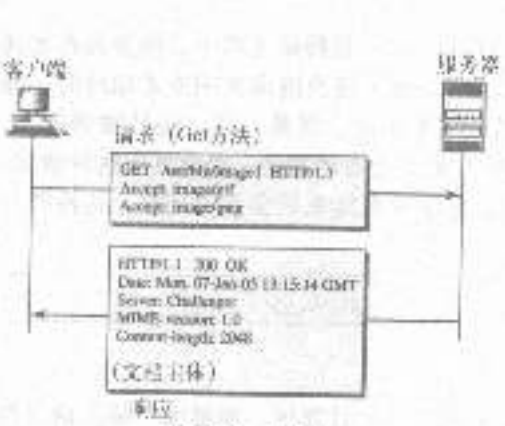


图27.16 例27.1

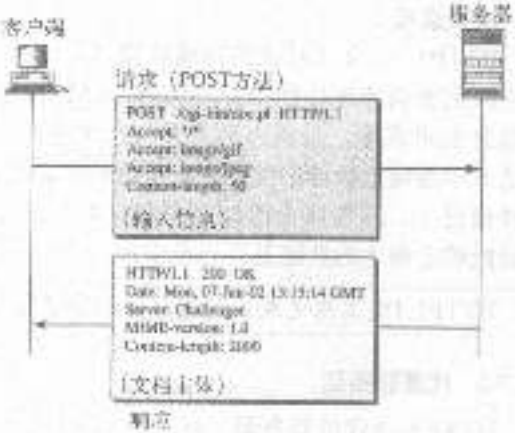


图27.17 例27.2

例27.3

HTTP使用ASCII字符，客户端使用TELNET直接连接到服务器上，登录端口号是80。接下来的二行说明了这个连接是成功的。

然后我们输入三行，第一行给出请求行（GET方法），第二行给出头部（定义了主机），第三行是空行，结束请求。

服务器响应是以状态行开始的7行信息。最后的空行作为服务器响应的结束。在空行之后是接收到的14 230行的文件（在这里没有显示）。最后一行是客户端的输出。

```
$ telnet www.mhhe.com 80
Trying 198.45.24.104...
Connected to www.mhhe.com (198.45.24.104).
Escape character is '^]'
GET /engcs/compsci/forouzan HTTP/1.1
From: forouzanbehrouz@fhda.edu

HTTP/1.1 200 OK
Date: Thu, 28 Oct 2004 16:27:46 GMT
Server: Apache/1.3.9 (Unix) Apache/Serv/1.1.2 PHP/4.1.2 PHP/3.0.18
MIME-version:1.0
Content-Type: text/html

Last-modified: Friday, 15-Oct-04 02:11:31 GMT
Content-length: 14230

Connection closed by foreign host.
```

27.3.2 持续与非持续连接

HTTP在版本1.1之前定义了非持续连接，而在版本1.1中默认的是持续连接。

非持续连接

在非持续连接（nonpersistent connection）中，每一次请求/响应都要建立TCP连接。下面是实现这一策略的步骤：

1. 客户端建立TCP连接，并发送请求；
2. 服务器发送响应，并关闭连接；
3. 客户端读取数据，直到遇到文件结束标志，客户端随后关闭连接。

在这种策略中，对于不同文件中的N个不同的图片，连接必须建立和关闭N次。因为服务器需要N个不同的缓冲区，并且每次建立一个连接时，需要一个较慢的启动过程，所以非持续策略在服务器端增加了很大的开销。

持续连接

HTTP1.1定义了默认的**持续连接**（persistent connection）。在持续连接中，服务器在发送响应以后会保持连接处于开启状态，以等待更多的请求。如果客户请求关闭或者超时时，服务器会关闭连接。发送方通常在每次响应时会发送数据的长度。但是，在一些特殊情况下，发送方不能确定数据的长度。如果文档是动态创建的或者属于活动文件，就会发生这种情况。这种情况下，服务器会通知客户端，文件长度未知，发送数据结束后会关闭连接。这样客户端就能确定到达的数据是否已经结束。

> HTTP1.1版本指定默认的连接是持续连接。

27.3.3 代理服务器

HTTP支持**代理服务器**（proxy server）。代理服务器是一台计算机，能够保存最近请求的响应的副本。HTTP客户端向代理服务器发送请求。代理服务器检查本机的高速缓存。如果高速缓存中不存在响应报文，代理服务器就向相应的服务器发送请求。返回的响应会发送到代理服务器中，并且进行存储，以用于其他客户端将来的请求。

代理服务器降低了原服务器的负载，减少了通信量并降低了延迟。但是，使用代理服务器，客户端必须配置为访问代理服务器而不是目标服务器。

推荐读物

为了深入讨论与本章有关的主题，推荐下列读物和网站。在本章末列出方括号[…]中的参

考资料。

读物

HTTP在[Ste96]的第13章和第14章、[PD03]的9.3节、[Com04]的第35章、[Tan03]的7.3节中讨论。

网站

下面的网站与本章中讨论的主题有关：

www.ietf.org/rfc.html关于RFC的信息

请求评论

下面是与WWW相关的RFC：

1614，1630，1737，1738

下面是与HTTP相关的RFC：

2068，2109

本章小结

- 万维网（WWW）是分布在世界各地互相链接在一起的信息仓库。
- 超文本是通过指针互相链接在一起的文档。
- 浏览器解释并显示文档。
- 浏览器由控制程序、客户端程序及解释程序组成。
- Web文档可以分为：静态文档、动态文档和活动文档。
- 静态文档是存储在服务器中的内容固定的文档，客户机不能改变服务器上的文档。
- 超文本标记语言（HTML）是用于创建静态Web页面的语言。
- 任何浏览器都能够读取嵌入在HTML文档中的格式化指令（标签）。
- 标签提供了文档的结构，定义了标题和头部，格式化文本，控制数据流，插入图，将不同的文档链接在一起，并且定义了可执行码。
- 只有在浏览器请求后，服务器才能创建动态Web文档。
- 公共网关接口（CGI）是创建、处理动态Web文档的标准。
- CGI程序和嵌入的CGI接口标签可以用多种语言编写，如C、C++、shell脚本语言或者Perl。
- 活动文档是客户端检索的程序的副本，在客户端运行。
- Java是一种高级程序设计语言，一种运行时环境及允许程序员编写活动文档的类库的组合。生成的活动文档可以在浏览器中运行。
- Java用来创建applet（小应用程序）。
- 超文本传输协议（HTTP）是万维网（WWW）存取数据的主要协议。
- HTTP使用TCP连接传输文件。
- HTTP报文在形式上与SMTP报文相似。
- HTTP请求行由请求类型、URL、以及HTTP版本号组成。
- 资源定位符（URL）是由方法、主机、可选的端口号以及在WWW中定位信息的路径名组成。
- HTTP请求类型或方法是由客户端向服务器发出的实际命令或请求。
- 状态行是由HTTP版本号、状态码和状态短句组成。

- HTTP状态码代表了通用的信息，这些信息与成功请求、重定向信息、错误信息相关。
- HTTP头部代表了在客户端与服务器之间的额外信息。
- HTTP头部由头部名字和头值组成。
- HTTP通用头部给出了关于请求和响应报文的通用信息。
- HTTP请求头部说明了客户端的配置和最佳的文档格式。
- HTTP响应头部说明了服务器的配置和关于请求的特殊信息。
- HTTP实体头部提供了关于文档主体的信息。
- HTTP1.1版本定义了持续连接。
- 代理服务器保存了最近请求的响应副本。

习题

复习题

1. HTTP与WWW之间有什么关系？
2. HTTP与SMTP的相似之处是什么？
3. HTTP与FTP的相似之处是什么？
4. 什么是URL？它由哪些部分组成？
5. 什么是代理服务器？它与HTTP有什么关系？
6. 说出浏览器共有的三个组成部分的名字。
7. Web文档有哪三种类型？
8. HTML的含义是什么？它的功能有哪些？
9. 活动文档与动态文档的区别是什么？
10. CGI的含义是什么？有什么功能？
11. 描述Java与活动文档之间的关系。

练习题

12. 试说出下面每一个图将会显示在屏幕上的什么位置？

```
Look at the following picture:
Then tell me what you feel:
<IMG SRC="Pictures/Funny1.gif" ALIGN=middle>
<IMG SRC="Pictures/Funny2.gif" ALIGN=bottom>
<B> What is your feeling? </B>
```

13. 说明下列HTML段落的效果：

```
The publisher of this book is <A HREF="www.mhhe">
McGraw-Hill publisher</A>
```

14. 写出检索文档/usr/users/doc/doc1的请求。至少使用2个通用头部、2个请求头部和一个实体头部。
15. 写出练习14成功请求的响应。
16. 写出练习14的响应。若文档已经永远地移动到/usr/deads/doc1。
17. 写出练习14的响应，如果在请求中有语法错误。
18. 写出练习14的响应。如果客户端未被授权访问这个文档。
19. 写出一个请求，用来查询在/bin/users/file的文档信息。请至少使用2个通用头部和一个请求头部。

20. 写出练习19成功请求的响应。
21. 写出将位于/bin/usr/file1的文件复制到/bin/file1的请求。
22. 写出练习21的响应。
23. 写出位于/bin/file1的文件删除的请求。
24. 写出练习23的响应。
25. 写出检索位于/bin/etc/file1的文件的请求。仅当这个文档在1999年1月23日之后修改过，客户端才需要这份文档。
26. 写出练习25的响应。
27. 写出检索位于/bin/etc/file1的文件的请求。客户端需要标识别自己。
28. 写出练习27的响应。
29. 写出在/bin/letter存储文件的请求。客户端标识它可以接受的文档的类型。
30. 写出练习29的响应。响应必须说明了文档的使用时限，以及内容可能改变的日期和时间。

第28章 网络管理

我们将网络管理（network management）定义为监视、检测、配置和故障发现及修理的网络组件，以满足组织机构所规定的一组要求。这些要求能使网络顺利的、高效的运行，为用户提供预定的服务质量。为了达到这个任务，网络管理系统使用软件、硬件以及人工参干来实现。本章首先简要讨论网络管理系统的功能，然后介绍最常用的网络管理系统：简单网络管理协议（SNMP）。

28.1 网络管理系统

我们说网络管理系统的功能可划分5大类：配置管理、故障管理、性能管理、安全管理和计费管理，如图28.1所示。



图28.1 网络管理系统的功能

28.1.1 配置管理

大型网络通常是由上千个物理地或逻辑地相互连接在一起的实体组成。在网络建立时，这些实体有一个初始的配置，但随着时间可能会改变。台式计算机可能由其他计算机取代，应用软件可能更新为新的版本，用户也可能由一组移动到另一组。任何时候，配置管理（configuration management）系统必须知道每个实体的状态及其与其他实体的关系。配置管理可划分成两个子系统：重新配置和文档资料。

重新配置

重新配置，是指调整网络组件与特性。在大型网络可能每天都会发生。重新配置有3种类型：硬件重新配置、软件重新配置和用户账号重新配置。

硬件重新配置包括硬件的所有变化。例如，台式计算机可能被其他计算机取代，一个路由器可能需要从网络的一个部分移动到另一个部分，某一个子网可能增加到网络上或从网络上删除。所有这些都需要网络管理及时关注。在大型网络中，必须有专职技术人员快速和高效地进行重新配置。令人遗憾的是，这种类型的重新配置不是自动的，而需按情况手工处理。

软件重新配置包括软件的所有变化。例如，新软件可能需要安装在服务器或客户机上，操作系统需要更新。幸好，大多数的软件重新配置都是自动的。例如，更新某些或所有客户机上的一个应用程序可以从服务器上下载电子版本。

用户账号重新配置不是简单地在系统上增加或删除用户，还必须考虑用户作为个人和作为一组的成员用户权限。例如，某个用户可以对有些文件有读和写的权限，但对另些文件仅有读

的权限。某种程度上，用户账号重新配置是自动的。例如，在学院或大学中，每一季度或每一学期开始将新生增加到系统中。根据新生所选择的课程或所从事的专业，通常将他们分成组。

文档资料

原始网络配置和每次变化都必须有详细的记录。这是说要有硬件、软件和用户账号文档。

硬件文档（Hardware documentation）通常包含两组文档：映照和说明。映照追踪硬件的每一部件及其与网络连接，也可有一个总的映照说明每个子网之间的逻辑关系。还可有第二个总映照，它说明了每个子网的物理位置。然后，对于每个子网有一个或多个说明了所有设备的映照。这些映照使用的标准能被当前的或以后的人们易于阅读和理解。仅有映照还是不够的，硬件的每一部件也需要有文档。连接网络的每一部件必需有一组说明。这些说明包含部件类型、序列号、制造厂商（地址和电话号码）、购买日期和保修卡等信息。

所有软件也需要有记录，软件文档包含的信息有软件型号、版本、安装日期和许可证。

多数操作系统有一个允许定义用户账号和权限的文件的程序，管理人员必须确保对这些文件进行更新与维护。有些操作系统在一个文档中记录访问权限：一个表示所有文件和每一个用户的访问类型；另一个表示访问一个具体文件的用户列表。

28.1.2 故障管理

当前复杂的网络是由数百个有时是数千个组件组成的，网络的正确运行取决于各个组件的正确运行以及相互之间的正确运行。**故障管理**（Fault management）是处理这个问题的网络管理。

一个有效的故障管理系统有两个子系统：再生的故障管理和主动故障管理。

再生的故障管理

再生的故障管理系统负责故障的检测、隔离、纠正和记录。它处理对故障的短期解决办法：

故障定义为系统的一种不正常的工作情况。再生的故障管理系统所采取的第一步是检测故障准确的位置。当一种故障发生的时候，系统或完全停止工作或产生过多的错误。故障的好例子是通信介质被损坏。故障可能中断通信或产生过多的差错。

再生的故障管理系统所采取的第二步是隔离故障。如果故障被隔离，则通常仅有少数用户受影响。隔离后，立即通知受影响的用户并给出一个纠正错误的大致时间。

第三步是纠正故障，这可包含替换和修理故障的部分。

故障纠正后，必须被记录在文档中。这个记录必须表明故障的准确位置，可能引起的原因或修复故障的措施、成本和每步花费的时间。文档有其重要有下列几点理由：

- 故障可能重复发生，文档有助于网络管理人员或技术人员解决同样的类似问题；
- 同一类故障的频率是系统中较重要问题的指示。如果某一故障在一个组件中经常发生，就应该用另一个组件去更换它，或者改变整个系统不要使用那种组件；
- 统计是对网络管理和性能管理有帮助的另一个部分。

主动的故障管理

主动的故障管理是防止故障发生。尽管通常这是不可能的，但有些故障还是可预知的和可制止的。例如，制造商指定一个部件或一个部件中某部分的有效期，在到达有效期之前，更换它是一种好的策略。再例如，如果在网络某一特定点经常有故障发生，那么在重新配置网络时注意防止故障再一次发生也是明智的。

28.1.3 性能管理

与故障管理有着密切关系的是性能管理（Performance management）。它试图监视和控制网络，确保网络尽可能有效地运行。性能管理试图通过一些可测量的量（如能力，通信量，吞吐量和响应时间等）来量化性能。

能力

性能管理系统必须监视的一个要素是网络的能力，每个网络的能力都是有限的，性能管理系统必须确保使用不能超过这个能力。例如，如果一个局域网设计为100个站点的平均数据速率是2Mbps，则连接到这个网络200个站点就不能正确地运行，数据速率将会下降甚至还可能发生拥塞。

通信量

通信量按两种方法测量：内部的和外部的。内部通信量是用网络内部传送分组（或字节）的个数来度量，外部通信量是用网络与外部交换分组（或字节）的个数来度量。系统在负载重的峰值期间，如果出现过多的通信量，拥塞就会发生。

吞吐量

我们可测量单独一台设备（如路由器）或部分网络的吞吐量。性能管理监视吞吐量确保它不降低于不可接受的水平。

响应时间

通常将从用户请求服务到服务被准许的时间称为响应时间。其他一些因素如能力和通信量会影响响应时间。性能管理监视平均响应时间和峰值时刻响应时间。

响应时间的增加是一个非常严重的情况，它表示网络正朝着超过它的能力方向工作。

28.1.4 安全管理

安全管理（Security management）是根据预定的负责策略监视对网络的访问。在第31章和第32章将讨论安全问题，尤其是网络安全问题。

28.1.5 计费管理

计费管理（Accounting management）是通过费用控制用户访问网络的资源。根据计费管理，个人用户、部门、公司，甚至工程等都需要对从网络上得到服务的付费。付费不必现金支付，为预算目的它可以记入部门或公司的账单上。今天，组织机构使用计费系统有下列原因：

- 它防止用户独占有限的网络资源；
- 它防止用户无效地使用系统；
- 网络管理员可以按需对网络使用作出短期和长期的计划。

28.2 简单网络管理协议（SNMP）

简单网络管理协议（SNMP）是使用TCP/IP协议族对互联网上的设备进行管理的一个框架，它提供一组基本的操作来监控和维护互联网。

28.2.1 概念

SNMP使用管理器和代理的概念。这就是说，管理器，通常是一台主机，控制和监视一组代理，代理通常是路由器（如图28.2所示）。

SNMP是应用级协议，它用少数几个管理器控制一组代理。该协议设计在应用级，因此它能监控不同厂商制造的设备，而这些设备可安装在不同的物理网络上。换言之，SNMP使管理任务与被管理设备的物理特性和联网技术没有关系。它可以用于由不同厂商制造的路由器相互连接在一起的不同局域网和广域网组成的异构互联网中。

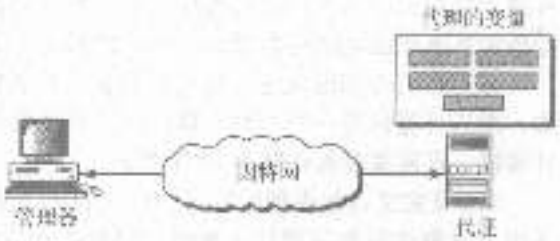


图28.2 SNMP概念

管理器和代理

称为**管理器**（manager）的管理站是运行SNMP客户程序的主机。称为**代理**（agent）的被管理站是运行SNMP服务器程序的路由器（或主机）。管理是通过管理站与代理之间的简单交互来实现的。

代理在数据库中保存性能信息。管理器使用该数据库中的值。例如，路由器可将接收到的和转发的分组数存储成适当的变量。管理器可读取和比较这两个变量，以便确定路由器是否拥塞。

管理器也可使路由器完成某些动作。例如，路由器定期检查一个重新启动计数器的值，看它何时应当重新启动自己。例如，当计数器的值为0时就应该引导自己。管理器可使用这个特性在任何时候从远程重新引导这个代理。它只要发送一个分组，就迫使这个计数器的值为0。

代理也可参加到管理过程中。在代理上运行的服务器程序可检查环境，如果发现有异常现象可发送一个称为陷阱（trap）的告警报文给管理器。

换言之，SNMP管理是基于3个基本思想：

1. 管理器检查代理的方法是发出请求得到能反映代理行为的信息；
2. 管理器可用重新设置在代理数据库中的某些值来强迫代理完成一个任务；
3. 代理管理过程的方法是向管理器发送异常情况的告警。

28.2.2 管理组件

为了完成管理任务，SNMP还使用另外两个协议：**管理信息结构**（Structure of Management，SMI）和**管理信息库**（Management Information Base，MIB）。换言之，在因特网上的管理是通过SNMP，SMI和MIB三个协议共同协作完成的，如图28.3所示。

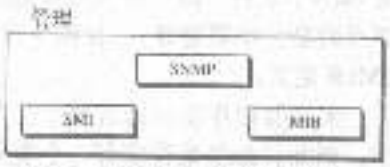


图28.3 因特网上网络管理组件

让我们详细阐述三个协议之间的交互作用。

SNMP的作用

SNMP在网络管理中起着一些很特殊的作用。它定义从管理器到代理和从代理到管理器发送分组的格式。还解释产生的结果并创建统计表（时常借助其他的管理软件）。交换的分组包括对象（变量）名及它们的状态（值）。SNMP负责读取和改变这些值。

> SNMP定义管理器与代理之间交换分组的格式，读取和改变SNMP分组中对象（变量）的状态（值）。

SMI的作用

要使用SNMP，我们需要有一些规则，需要命名对象的规则。本为在SNMP中的对象是有

层次结构的（对象有一个父对象和多个子对象），所以命名规则特别重要。对象名字的一部分可继承父对象的名字。还须有规则定义对象的类型。SNMP能处理什么样的类型？SNMP能否处理简单类型还是结构类型？这些类型的定义域是什么？这些类型如何编码？

由为我们不知道发送、接收或存储这些值的计算机的结构，我们必须有适用的规则。例如，发送计算机是一个大型计算机，能将整数存储为8字节数据，而接收计算机是一个小型的计算机，仅能将整数存储为4字节数据。

SMI是定义这些规则的协议。但我们必须明白SMI仅是定义规则。不能定义一个实体管理多少个对象或对象使用什么类型。SMI是给对象命名和定义对象数据类型的规则集合。对象与数据类型的关联不由SMI来做出。

> SMI定义对象命名、数据类型（包括长度和定义域）以及编码方法和取值的一般规则。
> SMI不定义实体应管理多少个对象或被管理对象的命名，或对象与值之间的关联。

MIB的作用

我们期望清晰地了解另一个所需协议。对每个被管理的实体，这个协议必须按照SMI规定的规则定义对象的个数和给对象命名，以及每个对象与类型的关联。这就是MIB协议，与数据库（在数据库中，大多数元数据未命名和没有类型）类似。MIB对每个实体创建一组规定的对象。

> MIB创建一个被管理的实体中所有对象命名和类型，以及它们之间相互关系。

类比

在较详细讨论这三个协议的之前，我们先给出一个类比。这三个网络管理组件类似于为求解某个问题用计算机语言所编写的程序。

在编写程序前，必须预先定义程序设计语言的语法（如C语言或Java语言）。语言还定义变量的结构（简单的、结构的、指针等）以及如何给变量命名。例如，一个变量名字长度需要1到N个字符，以一个字母开始后跟数字、字母字符。语言还定义所用的数据类型（整数型、浮点数型、字符型等）。在程序设计中，规则是由语言来定义的。而在网络管理中，规则是由SMI来定义。

大多数程序设计语言要求在每个具体的程序中声明变量，其内容包括变量的名字及其类型。例如，如果某程序有二个变量：一个名字为*counter*，数据类型是整数型；另一个名字为*grades*，数据类型是字符型，那么在程序开始时这二个变量必须声明：

```
int counter;
char grades [40];
```

注意：声明给变量命名（*counter*和*grades*）和定义每个变量的数据类型。由于在程序设计语言中，数据类型是预先定义的，所以程序知道每个变量的定义域与大小。

MIB在网络管理中负责类似的任务，MIB给每个对象命名并定义其类型。因为SMI定义类型，所以SNMP知道其定义域与大小。

在编制程序的声明后，程序要编写语句存储变量的值，如有需要，还要更新其值。SNMP在网络管理中负责类似的任务，SNMP按SMI定义规则对MIB已说明的对象值进行存储、更新和解释。

> 我们比较网络管理任务与编写程序的任务。
> • 两个任务都需要规则。在网络管理中，它由SMI处理。

- 两个任务都需要变量声明。在网络管理中，它由MIB处理。
- 两个任务都需要用语句来执行动作。在网络管理中，它由SNMP处理。

概要

在详细地讨论每个组件之前，我们用一个简单的情况说明每个组件涉及到的内容。这仅是概要的内容，在本章末将展开详述。管理器站点（SNMP客户）想要发送一个报文给代理站点（SNMP服务器）查找代理接收到UDP用户数据报的个数。图28.4表示每步的概要。

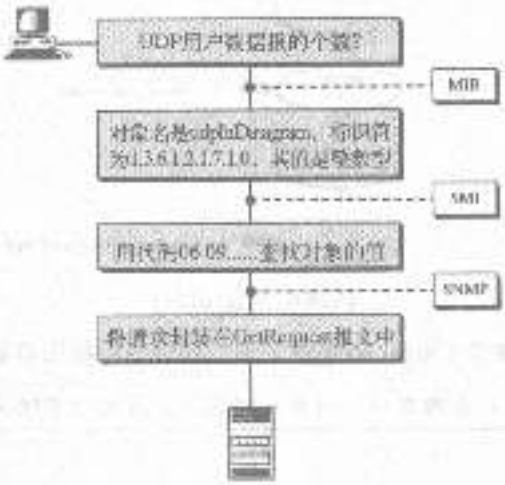


图28.4 管理概要

MIB负责查找保存接收到的UDP用户数据报个数的对象。在另一个嵌入协议的帮助下，SMI负责给对象名编码。SNMP负责创建一个称为GetRequest的报文，并封装已编码的报文。实际上，事情比概述复杂得多，但是，我们必须对每个协议有更详细的叙述。

28.2.3 管理信息结构

管理信息结构第二版（SMIv2）是网络管理中的一个组件，它的功能是：

1. 给对象命名；
2. 定义可在对象中存储的数据类型；
3. 给出如何对在网络上传输的数据进行编码的方法。

SMI是SNMP的一个指南。它强调了处理对象的三个属性：名字、数据类型和编码方法（如图28.5所示）。



图28.5 对象属性

名字

SMI要求每一个被管理对象（如一个路由器，一个路由器中的变量，一个值）具有一个唯一的名字。为了在全局给对象命名，SMI使用对象标识符（object identifier），它是基于树结构的分层次标识符（如图28.6所示）。

树结构从一个未命名的根开始，每一个对象可用一个点分隔开的整数序列来定义，树结构也可用一个点分隔开名字的文本序列来定义。整数-点的表示法是SNMP使用的一种方法，人们使用名字-点表示法，例如，下面是同一对象的两种不同的表示法：

iso.org.dod.internet.mgmt.mib-2→1.3.6.1.2.1

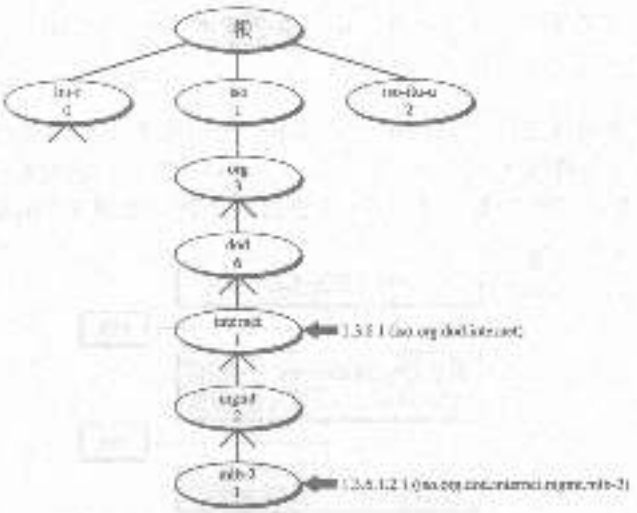


图28.6　对象标识符

在SNMP中使用的对象位于mib-2对象的下面，所以它们标识符都从1.3.6.1.2.1开始。

所有被SNMP管理的对象都有一个对象标识符，对象标识符都是从1.3.6.1.2.1开始。

类型

对象的第二个属性是在这个对象中存储的数据类型。为了定义数据类型，SMI使用抽象语法表示法1（Abstract Syntax Notation1，ASN.1）的一些基本定义，但还增加了几个新的定义。换言之，SMI既是ASN.1的一个子集，又是ASN.1的一个超集。

SMI使用两大类数据类型：简单的和结构化的。我们先定义简单类型，然后讨论如何从简单类型构造出结构化类型（如图28.7所示）。



图28.7　数据类型

简单数据类型（simple data type）是原子数据类型。这些类型中一些是直接取自ASN.1，另一些是SMI增加的。表28.1给出了最重要的一些类型。前5个取自ASN.1，后7个是SMI定义的。

表28.1　数据类型

类　型	大　小	说　明
INTEGER	4字节	在-2^{31}到$2^{31}-1$之间的一个整数
Integer32	4字节	与INTEGER类型相同
Unsigned32	4字节	在0到$2^{32}-1$之间的一个无符号整数
OCTET STRING	可变	长度最多达65 535字节字串
OBJECT IDENTIFIER	可变	对象标识符
IPAddress	4字节	由4个整数组成的IP地址
Counter32	4字节	一个整数，其值由0增加到2^{32}，当它达到最大值时就返回0
Counter64	8字节	64位计数器

（续）

类 型	大 小	说 明
Gauge32	4字节	与Counter32同，但当它达到最大值时不返回，而保持最大值直到复位为止
TimeTicks	4字节	记录时间的计数值，以1/100s为单位
BITS		比特
Opaque	可变	未解释的串

结构化类型 将简单的和结构化的数据类型结合起来，就可构成新的结构化数据类型。SMI定义了两种结构化数据类型（structured data type）：sequence和sequence of。

- Sequence。一个sequence数据类型是一些简单数据类型的组合，不必都是相同类型。它与C编程语言中使用的struct或record的概念相似。
- Sequence of。一个sequence of数据类型是所有相同类型的简单数据类型的组合，或所有相同类型的sequence数据类型的组合。它与C编程语言中使用的array的概念相似。

图28.8说明了数据类型。

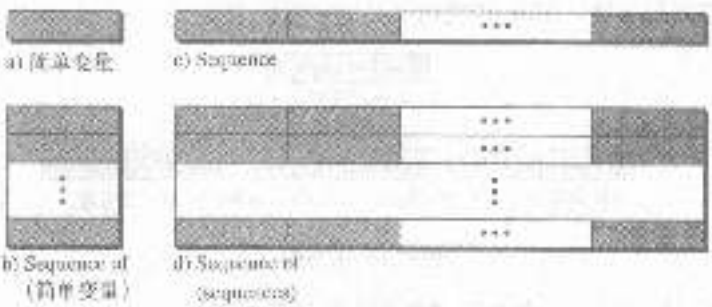


图28.8 概念性的数据类型

编码方法

SMI使用另一个标准，即基本编码规则（Basic Encoding Rule，BER），对数据进行编码并将编码后的数据在网络上传输。BER规定每一块数据都要被编码成三元组的格式：标签、长度和值，如图28.9所示。

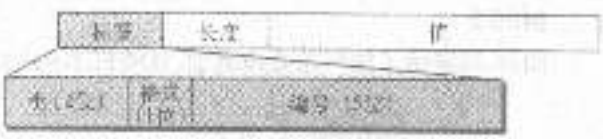


图28.9 编码格式

- 标签是一个字节的字段，它定义了数据类型。它由3个子字段组成：类（2位）、格式（1位）和编号（5位）。类子字段定义数据的作用域。其定义4类：通用类（00）、应用类（01）、特定上下文类（10）和专用类（11）。通用数据类型来自ASN.1（INTEGER、OCTET STRING和ObjectIdentifier）。应用数据类型是由SMI增加的（IPAddress、Counter、Gauge和TimeTicks）。有5种特定上下文数据类型，其意义随着协议的不同而不同，专用数据类型是特定厂商使用的。

格式子字段指出数据是简单的（0）还是结构化的（1）。编号子字段将简单的或结构化的数据进一步划分成一些子组。例如，在通用类中，对于简单格式，INTEGER的值是2，OCTET STRING的值是4等。表28.2给出本章所使用的数据类型，以及其标签的二进制和十六进制数。

表28.2　数据类型的代码

数据类型	类	格式	编号	标签（二进制）	标签（十六进制）
INTEGER	00	0	00010	00000010	02
OCTET STRING	00	0	00100	00000100	04
OBJECT IDENTIFIER	00	0	00110	00000110	06
NULL	00	0	00101	00000101	05
Sequence，sequence of	00	1	10000	00110000	30
IPAddress	01	0	00000	01000000	40
Counter	01	0	00001	01000001	41
Gauge	01	0	00010	01000010	42
TimeTicks	01	0	00011	01000011	43
Opaque	01	0	00100	01000100	44

- 长度字段是一个或多个字节。如果是一个字节，则最高位必定是0，其余7位定义数据长度。如果大于一个字节，则第一个字节的最高位必定是1，其他7位则定义在定义这个长度时所需的字节个数。图28.10说明了长度字段的意义：

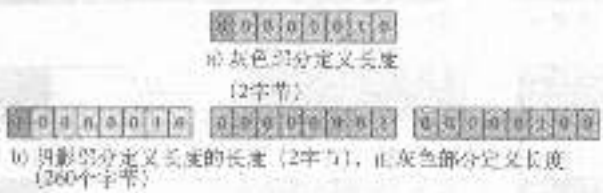


图28.10　长度格式

- 值字段按照在BER中定义的规则将数据的值进行编码。

我们给出一些例子来说明这三个字段（标签、长度和值）是如何定义对象的。

例28.1

图28.11表示了如何定义整数（INTEGER）14。

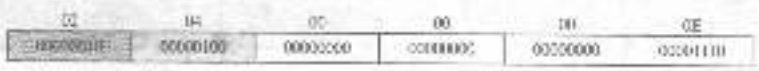


图28.11　例28.1 整数14

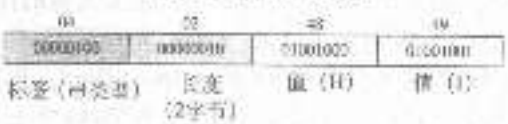


图28.12　例28.2 串"HI"

例28.3

图28.13表示了如何定义对象标识符1.3.6.1（iso.org.dod.internet）。

例28.4

图28.14表示了如何定义IP地址131.21.14.8。

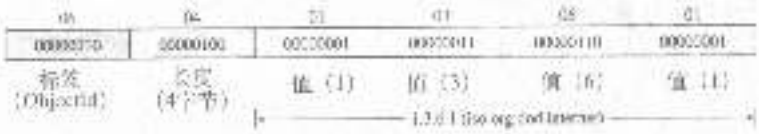


图28.13 例28.3对象标识符1.3.6.1

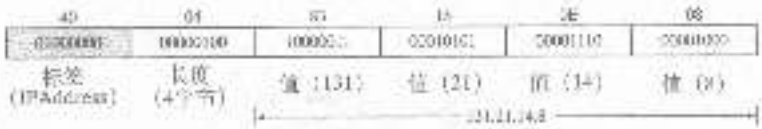


图28.14 例28.4 IP地址131.21.14.8

28.2.4 管理信息数据库（MIB）

管理信息数据库版本2（MIB2）是网络管理中的第二个组件。每个代理都有它自己的MIB2，这是管理器能够管理的所有对象的集合。在MIB中的对象分成10个不同的组：system、interface、address translation、ip、icmp、tcp、udp、egp、transmission和snmp。这些组都在对象标识符树中的mib-2对象的下面（如图28.15所示），每一个组定义一些变量和（或）表。

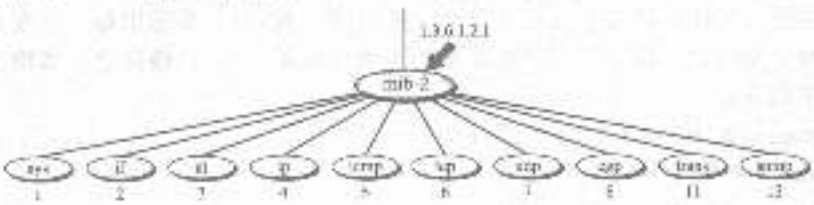


图28.15 mib-2

下列是一些对象的简要说明：

- sys 这个对象（system）定义了关于整个节点（系统）的通用信息，如名字、位置和生命周期。
- if 这个对象（interface）定义了关于整个节点所有接口的信息，如接口数、物理地址和IP地址。
- at 这个对象（address translation）定义了关于ARP表的信息。
- ip 这个对象定义了有关IP的信息，如路由表和IP地址。
- icmp 这个对象定义了有关ICMP的信息，如已发送和已接收到的分组个数和产生总差错数。
- tcp 这个对象定义了有关TCP的信息，如连接表、超时值、端口数及已发送和已接收到的分组个数。
- udp 这个对象定义了有关UDP的信息，如端口数及已发送和已接收到分组的个数。
- snmp 这个对象定义了有关SNMP本身的信息。

访问MIB变量

我们用udp组作为例子，说明如何访问不同的变量。在udp组中有4个简单变量和一个记录（表）序列，图28.16给出这些变量和表。

我们将说明如何访问每一个实体。

简单变量。要访问任何简单变量，我们用组的id（1.3.6.2.1.7）后面跟着该变量的id。下

面表示如何访问每一个变量：

udpInDatagrams ⇒ 1.3.6.1.2.1.7.1

udpNoPorts ⇒ 1.3.6.1.2.1.7.2

udpInErrors ⇒ 1.3.6.1.2.1.7.3

udpOutDatagrams ⇒ 1.3.6.1.2.1.7.4

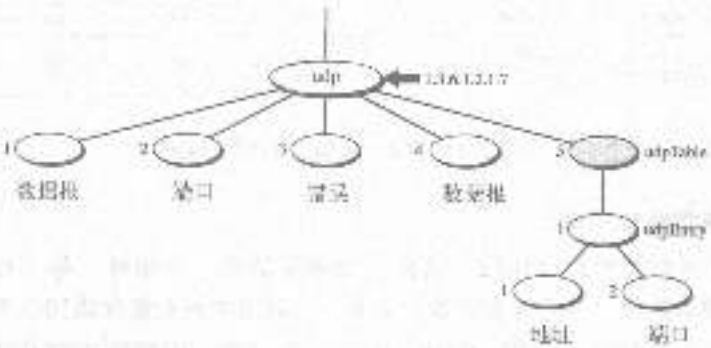


图28.16　udp组

但是，这些对象标识符定义的是变量而不是实例（内容）。要给出每一个变量的实例或内容，必须增加一个实例后缀，一个简单变量的实例后缀是一个0。换言之，要给出以上变量的实例，使用下列方法：

udpInDatagrams."0" ⇒ 1.3.6.1.2.1.7.1.0

udpNoPorts."0" ⇒ 1.3.6.1.2.1.7.2.0

udpInErrors."0" ⇒ 1.3.6.1.2.1.7.3.0

udpOutDatagrams."0" ⇒ 1.3.6.1.2.1.7.4.0

表　要标识一个表，首先使用表id。如图28.17所示，udp组只有一个表（id为5）。

因此，要访问这个表，使用下面形式：

udpTable ⇒ 1.3.6.1.2.1.7.5

但是，这个表不是树结构的叶子级，我们不能访问这个表，因此如下定义在这个表（id为1）中的项目（序列）：

udpEntry ⇒ 1.3.6.1.2.1.7.5.1

这个项目也不是一个叶子级，我们不能访问它，因此需要定义这个项目中的每一个实体（字段）：

udpLocalAddress ⇒ 1.3.6.1.2.1.7.5.1.1

udpLocalPort ⇒ 1.3.6.1.2.1.7.5.1.2

这两个变量在树的叶子上。虽然能访问它们的实例，但需要定义是哪一个实例。在任何时候，这个表对每一个本地地址/本地端口对可以有多个值。要访问表中的一个特定实例（行），应当给以上id加上索引。在MIB中，数组的索引不是整数（像大多数的编程语言那样）。这些索引是基于在这些项目中的一个或多个字段的值。在我们的例子中，udpTable的索引是基于本地地址和本地端口号，例如，图28.18显示一个具有4行的表和每一个字段的值。每一行索引是这两个值的组合。

要访问第一行本地地址的实例，使用标识符加上实例的索引：

udpLocalAddress.181.23.45.14.23 ⇒ 1.3.6.1.2.7.5.1.1.181.23.45.14.23

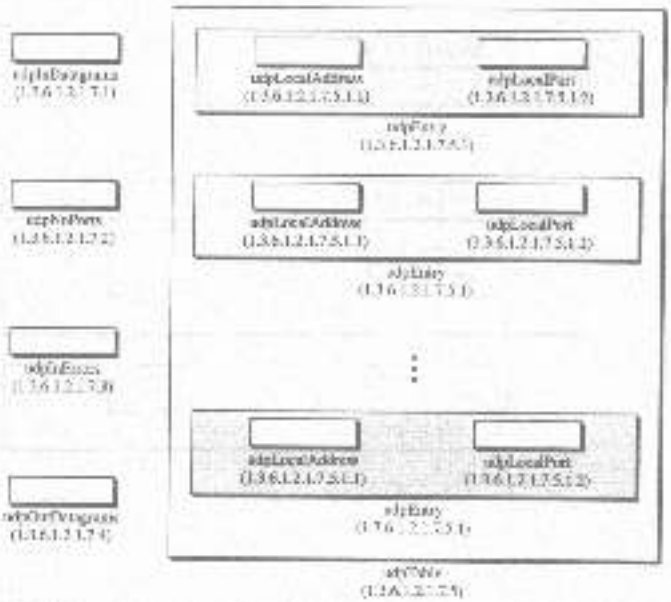


图28.17 udp的变量和表

注意：并非所有表的索引都是这样，有些索引使用一个字段的值，有些使用两个字段的值，等等。

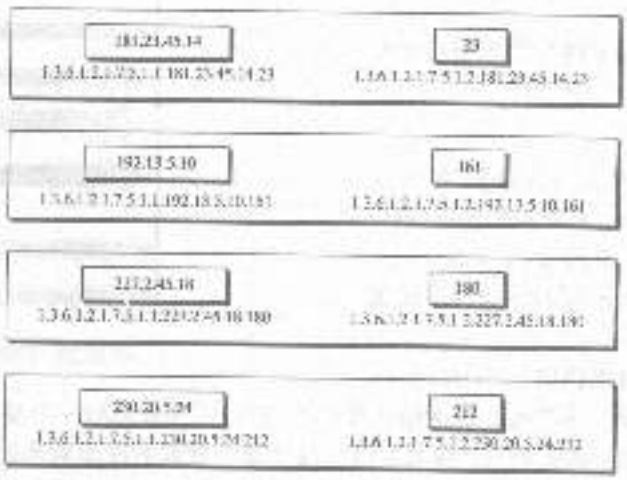


图28.18 udpTable的索引

28.2.5 字典排序

关于MIB变量的一个有趣问题就是对象标识符（包括实例标识符）是按字典的顺序排列的，表顺序是按照列-行规则，这就是说，要按表列的顺序。而在每一列中，必须从顶到底，如图28.19所示。

字典排序（lexicographic ordering）能使一个管理器在定义了第一个变量后，可一个一个接着地访问一组变量，在下一节我们将会在GetNextRequest命令中看到这一点。

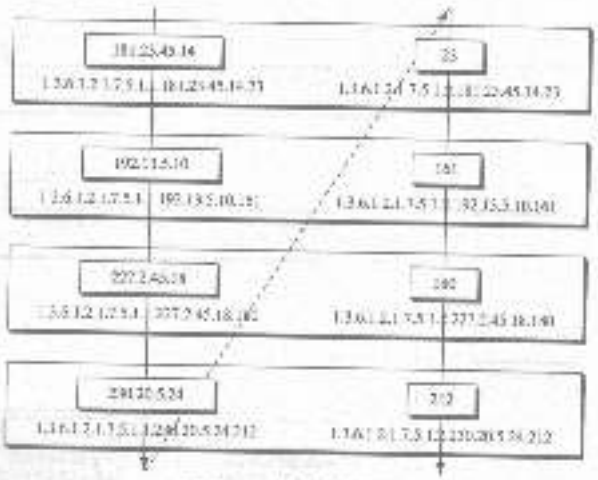


图28.19　字典排序

28.2.6　SNMP

SNMP在因特网的网络管理中使用SMI和MIB。SNMP是一个应用程序，它允许：

1. 管理器读取代理定义的一个对象的值；

2. 管理器将一个值存储在代理定义的一个对象中；

3. 代理将关于异常情况的告警报文发送给管理器。

PDU

SNMP版本3定义了8种报文（或PUD），GetRequest、GetNextRequest、GetBulkRequest、SetRequest、Response、Trap、InformRequest和Report（如图28.20所示）。

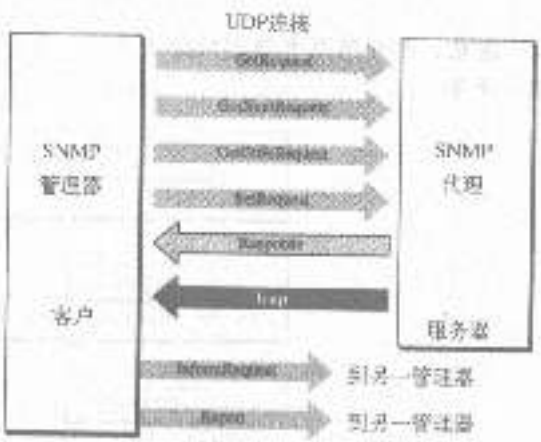


图28.20　SNMP PDC

GetRequest（请求读取）GetRequest PDU报文是由管理器（客户机）发送给代理（服务器），用来读取一个变量或一组变量。

GetNextRequest（读取下一个请求）PDU报文是由管理器发送给代理，用来读取一个变量的值。所要读取的值是PDU中定义的ObjectId后面对象的值，它主要是用来读取一个表中的项目的值。如果管理器不知道该项目的索引，它就不能读这个值。但是，它可以使用GetNextRequest并定义表的ObjectId。因为第一个项目的ObjectId是紧跟在表的ObjectId后面，因此第一个项目的值立即返回了。管理器可使用这个ObjectId来得到下一个项目的值，等等。

GetBulkRequest（读取大量请求）PDU报文是由管理器发送给代理，用来读取大量数据。它可用来取代多个GetRequest和GetNextRequest PDU。

SetRequest（设置请求）PDU报文是由管理器发送给代理，用来设置（存储）一个变量的值。

Response（响应）PDU报文是由代理发送给管理器以响应GetResponse和GetNextRequest，它包含管理器所请求的变量的值。

Trap（陷阱）（也称为SNMP版本2的Trap，以区别SNMP版本1的Trap）PDU报文是由代理发送给管理器，用来报告一个事件。例如，如果代理重新启动，它通知管理器并报告最新启动的时间。

InformRequest（读取信息）PDU报文是由一个管理器发送给另一个远程管理器，用来获得远程管理器所控制的代理的某些变量的值。

Report（报告）PDU用来报告管理器之间错误类型。

格式

8种SNMP PDU的格式如图28.21所示。在图28.21中，除GetBulkRequest有两处不同外，其他PDU都相同。

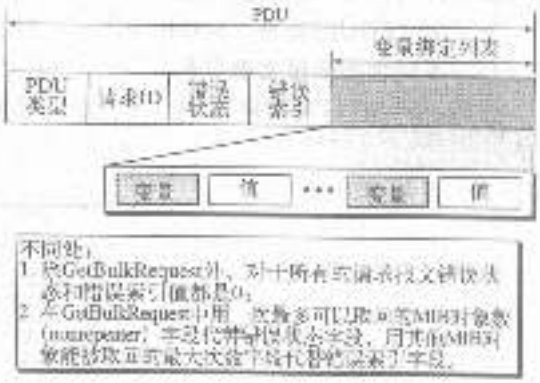


图28.21 SNMP PDU格式

各字段列出如下：

- PDU类型（PDU type）。这个字段定义PDU类型（如表28.4所示）。
- 请求标识（Request ID）。这个字段是序列号，由管理器在Request PDU中使用，而代理在响应中重复它。它用来使响应和请求相匹配。
- 错误状态（Error status）。这是只用于Response PDU中的一个整数，它给出代理报告的错误类型。在Request PDU中它的值是0，表28.3列出了可能出现的错误类型。

表28.3 错误类型

状 态	名 字	说 明
0	noError	无错误
1	tooBig	响应太大无法放入一个报文中
2	noSuchName	变量不存在
3	badValue	要存储的值无效
4	readOnly	这个值不能修改
5	genErr	其他错误

- 一次最多可以取回的MIB对象数（Nonrepeater）。这个字段只存在于GetBulkRequest PDU中，代替错误状态字段，在请求PDU中它的值是空。
- 错误索引（Error index）是一个偏移量，它告诉管理器哪一个变量引起这个错误。
- 其他MIB对象能被取回的最大次数（Max-repitetion）。这个字段只存在于GetBulkRequestPDU中，代替错误索引字段，在请求PDU中它是空的。
- 变量绑定列表（VarBindList）。这是管理器希望读取或设置的一组具有相应值的变量。在GetRequest和GetNextRequest中它的值是空。在TrapPDU中，它给出了与一个特定PDU有关的变量和值。

28.2.7 报文

SNMP不发送单个PDU，而是将它嵌入到一个报文中发送。SNMPv3中的报文是由四个元素组成：版本、头部、安全参数和数据（包含已编码的PDU），如图28.22所示。

因为不同报文中的这些元素的长度是不同的，所以SNMP使用BER标准对每个元素进行编

码。记住BER使用标签和长度来定义一个值。版本定义当前版本（3）。头部包含报文标识符、报文最大长度（回答的最大长度）、报文标记（OCTET STRING数据类型的一个8位组，其中每一位定义安全类型，如保密、鉴别或其他信息）以及报文安全性模型（定义安全协议）。报文安全参数用于创建报文摘要（见第31章）。数据包括PDU。如果数据是加密的，则存在有关加密引擎的信息（执行加密的管理器程序）和加密前后关系（加密类型）后跟被加密的PDU。如果数据没有加密，则数据就是PDU。

为了定义PDU的类型，SNMP使用一个标签。该类是上下文敏感的（10），格式是结构化的（1）。不同类型的报文编号是0，1，2，3，5，6.7和8（如表28.4所示）。

注意：SNMPv1定义标签A4作为陷阱，目前已不用了。



图28.22 SNMP报文

表28.4 SNMP报文的代码

代码	类	格式	编号	整个标签（二进制）	整个标签（十六进制）
GetRequest	10	1	00000	10100000	A0
GetNextRequest	10	1	00001	10100001	A1
Response	10	1	00010	10100010	A2
SetRequest	10	1	00011	10100011	A3
GetBulkRequest	10	1	00101	10100101	A5
InformRequest	10	1	00110	10100110	A6
Trap（SNMPv2）	10	1	00111	10100111	A7
Report	10	1	01000	10101000	A8

例28.5

在本例中，管理器站点（SNMP客户机）使用GetRequest报文读取路由器接收到的一些UDP数据报。

仅有一个变量绑定表项，与这个信息相应MIB变量是udpInDatagram，其对象标识符是1.3.6.1.2.1.7.1.0。管理器打算读取一个值（而不是存储一个值），因此值定义为0的空实体。

图28.23说明了分组与sequence的分级性质的概念。我们用白色与最里面的框表示sequence，而用其他灰色框表示UDP。

变量绑定表只有一个绑定变量。变量类型06，长度为09。变量类型05，长度为00。整个变量绑定是sequence，长度为$0D_{16}$（13）。变量绑定表也是sequence，长度为$0F_{16}$（15）。GetRequest PDU是长度为$1D_{16}$（29）。

此处，有3个OCTET STRING是关于安全参数、安全模型和标记的，还有2个interger定义最大长度（1024）和报文ID（64）。头部是sequence，长度为12，为简化用空白省略。整个报文是一个sequence，长度为52个字节。

图28.24表示了管理器站点（客户机）向代理（服务器）发送实际的报文。

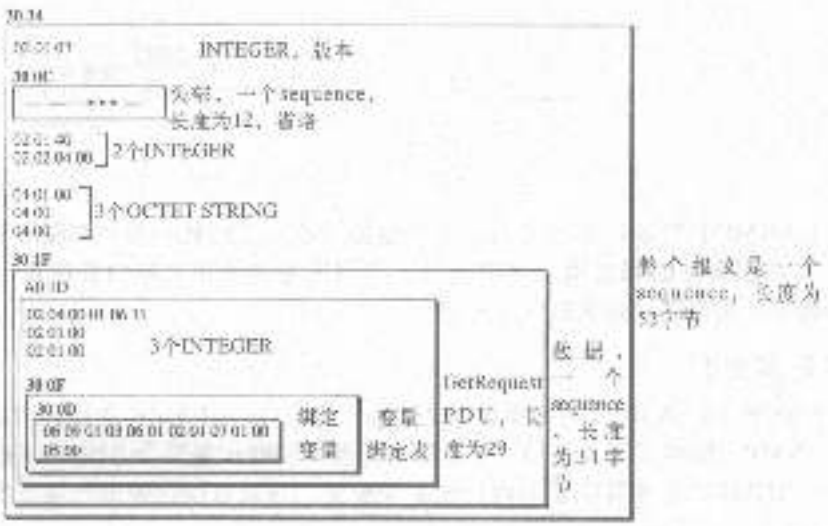


图28.23 例28.5

28.2.8 UDP端口

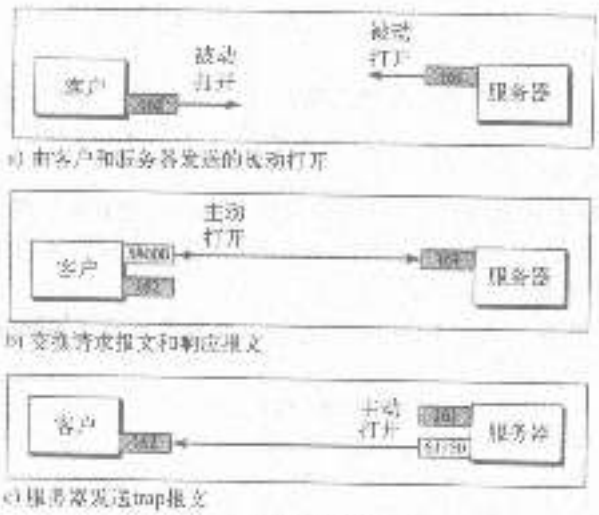
SNMP在两个熟知端口161和162上使用UDP的服务。熟知端口161由服务器（代理）使用，而熟知162端口由客户（管理器）使用。

代理（服务器）在端口161发出一个被动打开。然后它就等待从管理器（客户）来的连接。管理器（客户）使用临时端口发出一个主动打开。客户向服务器发送请求报文，使用临时端口作为源端口而熟知端口161作为目的端口。服务器向客户发送响应报文，使熟知端口161作为源端口而临时端口作为目的端口。

管理器（客户）在端口162发送一个被动打开，然后它就等待从代理（服务器）来连接。代理（服务器）只要有一个Trap报文要发送，就使用临时端口发出一个主动打开。这个连接是由向的，从服务器到客户（如图28.25所示）。



图28.24 GetRequest报文



a) 由客户和服务器发送的被动打开

b) 交换请求报文和响应报文

c) 服务器发送trap报文

图28.25 SNMP的端口号

在SNMP中的客户/服务器机制与其他协议不同。此处的客户和服务器使用熟知端口。此外，客户和服务器都必须不停地运行。其原因是请求报文是由管理器（客户）发出的，但Trap报文是由代理（服务器）发出的。

28.2.9 安全性

SNMPv3与SNMPv2的主要差别是提高了安全性。SNMPv3提供两类安全：通用的和专用的。SNMPv3提供了报文鉴别、安全性与管理器授权。这三个方面的问题将在第31章中讨论。另外，SNMPv3允许管理器远程修改安全配置。这就是说在管理器站点的管理器可以没有物理设备。

推荐读物

为了深入讨论与本章有关的主题，推荐阅读下列读物和网站。在本书末列出方括号[…]中的参考资料。

读物

[MS01]、[Ste94]第25章、[Stu04]的22.3节和[Com04]第39章中有SNMP的讨论。网络管理在[Sub01]中有讨论。

网站

下面网站有与本章有关题目的讨论：

- www.ietf.org/rfc.html有请求评论的信息

请求评论

下列是有关SNMP、MIB和SMI有关的请求评论：

1065，1067，1098，1155，1157，1212，1213，1229，1231，1243，1284，1351，1352，1354，1389，1398，1414，1441，1442，1443，1444，1445，1446，1447，1448，1449，1450，1451，1452，1461，1472，1474，1537，1623，1643，1650，1657，1665，1666，1696，1697，1724，1742，1743，1748，1749

本章小结

- 组成网络管理5个部分是配置管理、故障管理、性能管理、计费管理和安全管理。
- 配置管理涉及到网络实体的物理与逻辑的改变，它包括重新配置、软硬件文档和用户账号。
- 故障管理涉及到每个网络部件的正确操作，它可以是主动的或再生的。
- 性能管理涉及到监视和控制网络，确保网络尽可能有效地运行。用能力、通信量、吞吐量和响应时间等来度量。
- 安全管理涉及控制对网络的访问。
- 计费管理涉及通过计费控制用户对网络资源的访问。
- 简单网络管理协议（SNMP）是使用TCP/IP协议族对互联网中的设备进行管理。
- 管理器通常是主机，它控制和监视一组代理，而代理通常是路由器。
- 管理器是运行SNMP客户程序的主机。
- 代理是运行SNMP服务器程序的路由器或主机。
- SNMP使管理任务与被管理设备的物理特性和下层的联网技术都没关系。
- SNMP使用另外两个协议服务：管理信息结构（SMI）和管理信息库（MIB）。
- SMI给对象命名，定义可在对象中存储的数据类型，以及对数据进行编码。
- SMI的对象按照分层次的树结构命名。
- SMI的数据类型按照抽象语法记法1（ASN.1）来定义。
- SMI使用基本编码规则（BER）对数据进行编码。
- MIB是能够被SNMP管理的对象组的集合。
- MIB使用字典排序来管理其变量。
- SNMP的功能有3种：

1. 管理器读取代理定义的对象的值。
2. 管理器将一个值存储在代理定义的对象中。
3. 代理将告警报文发送给管理器。

- SNMP定义了5种报文：GetRequest、GetNextRequest、SetRequest、GetRespone和Trap。
- SNMP在两个熟知的端口161和162上使用UDP服务。
- SNMPv3是在前面版本上增加了安全机制。

习题

复习题

1. 定义网络管理。
2. 列出网络管理的5个功能。
3. 定义配置管理及其目的。
4. 列出配置管理的两个子功能。
5. 定义故障管理及其目的。
6. 列出故障管理的两个子功能。
7. 定义性能管理及其目的。
8. 列出性能管理的4个可测量的量。
9. 定义安全管理及其目的。
10. 定义记费管理及其目的。

练习题

11. 试给出INTEGER（整数）1 456的编码。
12. 试给出OCTET STRING（串）"Hello World" 的编码。
13. 试给出长度为100的任一OCTET STRING（串）的编码。
14. 试说明下列的记录（sequence）是如何编码的。

INTEGER	OCTET STRING	IP Address
2345	"COMPUTER"	185.32.1.5

15. 试说明下列的记录（sequence）是如何编码的。

Time Tick	INTEGER	Object Id
12000	14564	1.3.6.1.2.1.7

16. 试说明下列的数组（sequence of）是如何编码的，其中每个元素是一个整数。

2345
1236
122
1236

17. 试说明下列的记录数组（sequence of sequence）是如何编码的。

INTEGER	OCTET STRING	Counter
2345	"COMPUTER"	345
1123	"DISK"	1430
3456	"MONITOR"	2313

18. 试对下列的代码进行解码：

a. 02 04 01 02 14 32
b. 30 06 02 01 11 02 01 14
c. 30 09 04 03 41 43 42 02 02 14 14
d. 30 0A 40 04 23 51 62 71 02 02 14 12

第29章 多 媒 体

近年来，技术的进步已经改变了人们使用音频和视频的方式。过去，我们通过收音机收听广播，通过电视收看视频节目，通过电话网与另一方交互式通信。但是时代已经变了，人们使用因特网不仅限于文字和图像通信，而且用于音频和视频服务。本章将集中介绍因特网在音频和视频方面的应用。

我们可以把音频和视频服务分为三大类：**流式存储音频/视频**（streaming stored audio/video）、**流式实时音频/视频**（streaming live audio/video）和**交互式音频/视频**（interactive audio/video），如图29.1所示。流式意味着用户在文件下载的同时就可以收听（或者观看）。



图29.1 因特网音频/视频

第一类，流式存储音频/视频是一些经过压缩并存储在服务器中的文件。客户机通过因特网下载这些文件，这就是我们有时所说的**音频/视频点播**（on-demand audio/video）。存储音频文件的例子有歌曲、交响乐、盗版书籍和一些精品讲座。存储视频文件的例子有：电影、电视节目和音乐视频剪辑等。

> 流式存储音频/视频是指对压缩的音频/视频文件的点播请求。

第二类，流式实时音频/视频是指用户通过因特网收听收看音频和视频节目。这方面比较好的一个应用实例是因特网广播。有一些广播电台仅仅在因特网上播出它们的节目，而多数电台用因特网和无线电波播放节目。因特网电视还没有普及，但是人们相信电视台将来会在因特网上播放它们的节目。

> 流式实时音频/视频是指电台和电视台通过因特网播放节目。

第三类，交互式音频/视频，人们使用因特网进行相互之间的交互式通信。这方面应用的一个较好实例是因特网电话和因特网远程会议。

> 交互式音频/视频是指使用因特网实现交互式音频/视频应用。

本章讨论这三种应用。首先要讨论与音频/视频有关的其他一些问题，如音频/视频数字化和音频/视频压缩。

29.1 音频与视频的数字化

在音频和视频信号通过因特网发送之前，需要进行数字化处理。下面将音频和视频分开讨论。

29.1.1 音频数字化

麦克风接收到声音时，会产生模拟的电子信号，表示为时间函数的声音振幅。这种信号称为模拟音频信号。模拟信号，如音频信号，可以数字化而产生数字信号。根据奈奎斯特定理，如果信号的最高频率是f，则需要每秒对信号采样$2f$次。有许多用于将音频信号进行数字

化的方法，但原理是一样的。

声音每秒采样8 000次，每次采样值是8位，其结果是64Kbps的数字信号。音乐则每秒采样44 100次，每个采样值是16位，这样会产生单声道705.6Kbps和立体声1.411Mbps的数字信号。

29.1.2 视频数字化

视频由一系列的帧构成。如果帧在屏幕上显示足够快，我们就会有运动的感觉。原因是人的眼睛不能够分辨出快速闪过的单帧图像。每秒的帧数没有统一的标准。在北美，常见的是每秒25帧。但是为了避免闪烁的情况出现，需要不断重复刷新帧。电视需要每一帧刷新两次，也就是说需要发送50帧。换言之，如果在发送方有存储器的话，发送的25帧中每一帧都要从存储器中提取重画两次。

每一帧会被分为更小的栅格，称为图像元素或者像素（pixel）。对于黑白电视，每个8位的像素可以表示为256个不同的灰度等级。对于彩色电视，每个像素是24位，每一种三原色（红、绿和蓝）占用8位。

可以计算出在特定分辨率下每秒所用的位数。在最低分辨率下，一个彩色帧由1 024×768个像素构成。这表示我们需要：

$$2 \times 25 \times 1\,024 \times 768 \times 24 = 944 \text{ (Mbps)}$$

这种数据速率需要一种高数据速率技术，如同步光纤网络（SONET）技术。如果使用低速率技术发送视频，就需要对视频进行压缩。

在因特网上传送视频需要进行压缩。

29.2 音频和视频压缩

通过因特网传送音频和视频需要进行压缩。本节先讨论音频压缩，然后再讨论视频压缩。

29.2.1 音频压缩

音频压缩可以用于语音和音乐。对于语音，需要压缩为64-kHz的数字化信号；对于音乐，需要压缩为1.411-MHz的信号。用于音频压缩的两类技术是：预测编码和感知编码。

预测编码

在预测编码（predictive encoding）技术中，使用对采样间的差异进行编码取代对所有的采样值进行编码。这种类型的压缩技术常用于语音。已经定义了几种标准，如GSM（13kbps）、G.729（8kbps）和G.723.3（6.4或5.3kbps）。对这些技术的详细讨论已经超出了本书的范围。

感知编码：MP3

用于创建CD质量音频的最常用的压缩技术是基于感知编码（perceptual encoding）的技术。如前所述，这种类型的音频至少需要1.411Mbps，如果不进行压缩是无法通过因特网进行发送的。MP3（MPEG音频3层），作为MPEG标准的一部分（在视频压缩的小节中讨论）就使用这种技术。

感知编码应用了声音心理学的科学知识。声音心理学是研究人们如何感知声音的。这种思想是基于人类听觉系统的某些缺陷。一些声音可以遮掩其他声音，遮掩现象在频率和时间上都可以出现。对于**频率遮掩**（frequency masking），某一频率范围内的较强声音能够部分或者全部遮掩另一个频率范围内较弱的声音。例如，如果在一个房间里，一个重金属乐队正在演奏，我们很难听到邻伴在说什么。对于**暂时遮掩**（temporal masking），很强的声音可以使

耳朵在它停止后的短时间内失去听觉。

MP3利用频率遮掩和暂时遮掩这两种现象来压缩音频信号。这种技术对频谱进行分析，并分为不同的组。完全被掩盖的频率分配零比特，部分遮掩的频率分配的比特数较小，无遮掩频率分配的比特数值较大。

MP3产生三种数据速率：96Kbps、128Kbps和160Kbps，速率是基于原始模拟声音频率范围内的。

29.2.2 视频压缩

前面已经讲过，视频是由多个帧构成的。每一帧是一幅图像。可以通过先压缩图像的方法来压缩视频。市场上目前有两种常用的标准。**联合图像专家组**（Joint Photographic Experts Group，JPEG）用于压缩图像，而**运动图像专家组**（Moving Picture Experts Group，MPEG）用于压缩视频。我们首先简单讨论JPEG，然后再讨论MPEG。

图像压缩：JPEG

如前所述，如果图片不是彩色的（灰度），那么每个像素可以用8位的整数（256级）来表示。如果图片是彩色的，那么每个像素可以用24位（3×8位）表示，红、绿、蓝（RGB）分别用8位表示。为了简化讨论，我们只关注灰度图片。

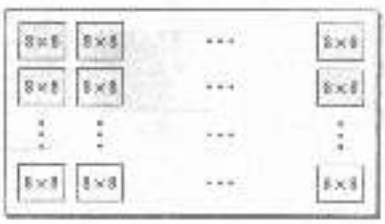


图29.2 JPEG灰度

在JPEG中，灰度图片被分割为8×8的像素块（如图29.2所示）。

将图片分割为块的目的是为了减少计算量。下面很快就会看到，每一张图片的数据计算次数是单元数目的平方。

JPEG的总体思想是将图片转换成揭示冗余的线性（矢量）数字集合。这样，冗余（不变化）部分就会通过使用文本压缩的方法来移除。图29.3是一个简化版本的处理过程。



图29.3 JPEG处理

离散余弦变换 这一步将每一个64像素块进行一种变换，称为离散余弦变换（discrete cosine transform，DCT），变换可以改变64个值，因而像素之间的相对关系得到保留，但却揭示了冗余部分。这里不给出公式，只给出三种情况下的变换结果。

第一种情况 在这种情况下，我们有一个单一灰度的块，每个像素的值是20。做变换时，我们得到第一个元素的值是非0值（左上角），其余像素的值是0。$T(0, 0)$的值是$P(x, y)$值的平均值（乘以一个常数），称为dc值（dc即直流，从电子工程中借用）。其余的值称为ac值。$T(m, n)$表示像素值的变化。由于没有变化，所以其余的值都是0（如图29.4所示）。

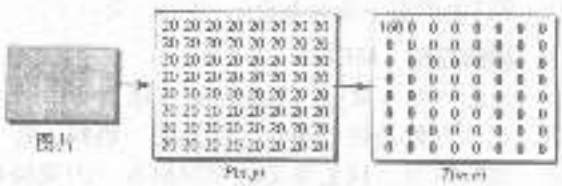


图29.4 第一种情况：单一灰度

第二种情况 在第二种情况下，我们有一个带有两种不同灰度的块。在像素值之间有一

个突变（从20跳到50）。做变换时，得到一个dc值与一系列非0的ac值。但是，只有少数非0值穿插在dc值的周围，多数值是0（如图29.5所示）。

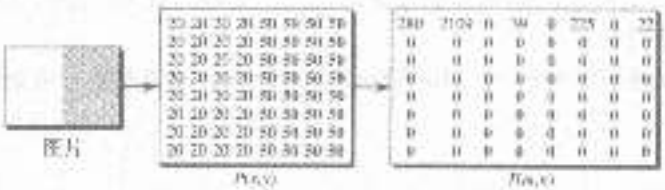


图29.5　第二种情况：两个部分

第三种情况　在第三种情况下，我们有一个块，它是逐步变化的。也就是说，相邻像素间的值没有突变。做变换时，得到一个dc值，也得到了很多非0的ac值（如图29.6所示）。

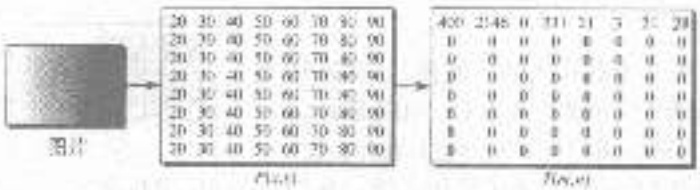


图29.6　渐变灰度等级

通过图29.4、图29.5和图29.6，我们可以得出下面的结论：

- 变换从表P中建立表T。
- dc值是所有像素的平均值（乘以一个常数）。
- ac值是变化的。
- 相邻像素之间没有变化时，生成0值。

量化　T表建立以后，将数值进行量化以减少编码所需要的位数。以前在量化（quantization）时，先要舍掉每一个值的小数部分，只保留整数部分。这里，我们用一个常数除每一个数，然后再舍掉小数部分，这样会更进一步减少所需要的位数。在大多数实现方法中，量化表（8乘8）定义了如何量化每一个数值。除数取决于值在T表中的位置。这样做是为了优化每一个特定应用中位的数量和0的数量。处理过程中唯一不可逆的阶段是量化阶段。在这里会丢掉某些信息，而且是不可恢复的。事实上，JPEG称为有损压缩的唯一原因是因为存在该量化阶段。

压缩　量化以后，从表中读取所有的值，并把冗余的0去掉。但是，为了把0组成串置于一起，表是按对角呈"之"字形读取的，而不是按行或者按列读取。原因是如果图片没有细小的变化，则T表的右下角会是全0。图29.7说明了这一过程。

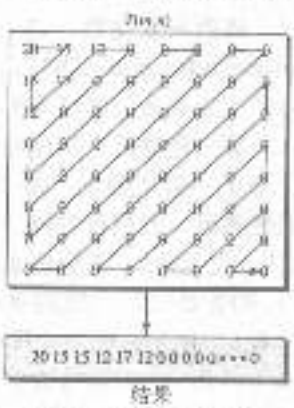


图29.7　表的读取方式

视频压缩：MPEG

运动图像专家组（MPEG）方法用于压缩视频。从原理上讲，运动图像是按顺序快速移动的一组帧。每一帧都是一幅图像。换句话说，帧是像素的空间组合，而视频是顺序发送的帧的时间组合。压缩视频，其含义是在空间上压缩每一帧，而在时间上压缩一组帧。

空间压缩　每一帧的空间压缩（spatial compression）通过JPEG完成（或者JPEG的修改版）。每一帧是一幅图像，可以单独进行压缩。

时间压缩 在时间压缩（temporal compression）中会去掉冗余帧。我们看电视时，每秒接收30帧，但是，大多数的连续帧几乎是相同的。例如，假设某个人在说话，除了帧中嘴唇部分外，大多数帧与前一帧是相同的，而嘴唇这一部分在每一帧中都可能会变化。

在时间域上压缩数据，MPEG方法首先将帧分为三类：I-帧、P-帧和B-帧，如图29.8所示。

- I-帧，内部编码帧（intracoded frame，I-帧）是一个独立帧，与其他任何帧没有关系（与相邻前后发送的帧都没有关系）。它们在规定的时间间隔内出现（例如，每一个第9帧是一个I-帧）。I-帧必须定期出现，用于处理帧中的突然变化。由于这种突然变化，使得前后帧无法显示。另外，在视频播出时，观看者可能会随时调整它的接收机。如果只在广播的开始时有一个I-帧，那么观众随后调谐其接收机时，可能会接收不到完整的画面。I-帧独立于其他帧，不能从其他帧中构造。
- P-帧，预测帧（predicted frame，P-帧），与前一个I-帧或P-帧有关，换言之，每一个P-帧只包含前一帧中变化的部分。这些变化的部分当然不能覆盖大的段，例如，一个快速移动的物体，这些新变化不会记录在P-帧中。P-帧只能够从前一个I-帧或者P-帧来构造。P-帧比其他类型的帧携带的信息要少得多，压缩后占用的位数很少。
- B-帧，双向帧（bidirectional frame，B-帧）与前一个或者后一个I-帧或P-帧有关。换句话说，每一个B-帧与过去和将来都有关联。注意，一个B-帧与另一个B-帧没有任何关系。

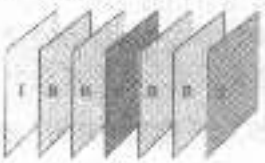


图29.8 MPEG3帧

图29.9显示了I-帧、P-帧和B-帧是如何从连续的七个帧中构造的。

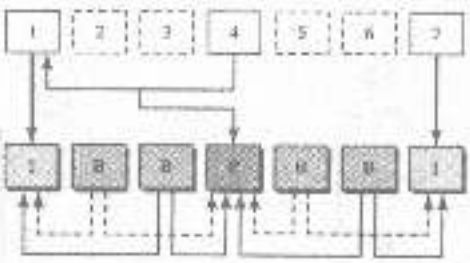


图29.9 MPEG帧的构造

MPEG已经发展了两个版本。MPEG1用于数据速率是1.5Mbps的CD-ROM。MPEG2用于数据速率是3Mbps－6Mbps的高质量DVD。

29.3 流式存储音频/视频

既然已经讨论过了音频/视频的数字化和压缩，那么现在就可以把注意力转向一些特定的应用了。首先是流式存储的音频和视频。从Web服务器中下载这些类型的文件可能与下载其他类型的文件有所不同。要理解这一概念，我们先使用三种方法，每一种都有不同的复杂度。

29.3.1 第一种方法：使用Web服务器

压缩的音频/视频文件可以像文本文件一样下载。客户端（浏览器）使用HTTP服务，发送GET报文下载文件。Web服务器可以将压缩的文件发送到浏览器中。然后浏览器借助于一个通常称为**媒体播放器**（media player）的应用程序播放文件。图29.10说明了这一方法。

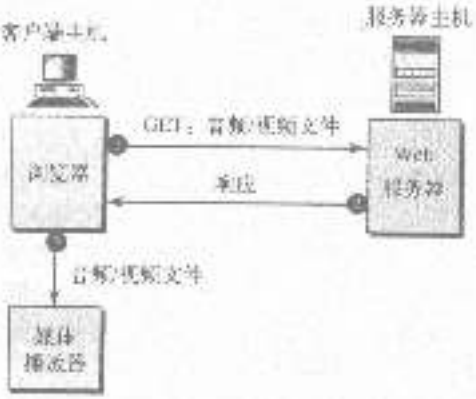


图29.10　使用Web服务器

这种方法很简单，还没有涉及到流。但是，它存在缺陷。一个音频/视频文件即使压缩以后通常仍然很大，音频文件可能包含几十兆字节，而视频文件可能包含几百兆字节。在这种实现方法中，文件在播放之前需要完整地下载下来。使用目前的数据速率，用户在文件播放之前需要几秒或者几十秒的时间来下载文件。

29.3.2　第二种方法：使用带有元文件的Web服务器

在这种方法中，媒体播放器直接连接到Web服务器并下载音频/视频文件。Web服务器存储两种文件：实际的音频/视频文件和一个元文件（metafile）。元文件中包含了音频/视频文件的有关信息。图29.11说明了这种方法的步骤。

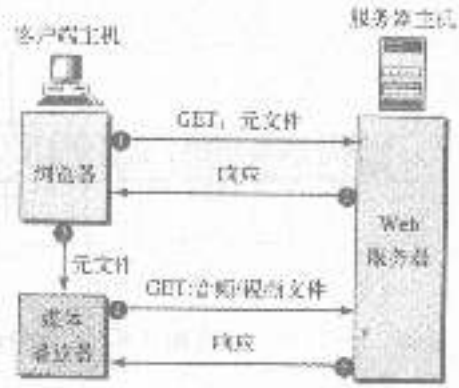


图29.11　使用带有元文件的Web服务器

1. HTTP客户端使用GET报文访问Web服务器。
2. 响应中包含元文件的有关信息。
3. 元文件传输到媒体播放器。
4. 媒体播放器使用元文件中指定的URL来访问音频/视频文件。
5. Web服务器做出响应。

29.3.3　第三种方法：使用媒体服务器

第二种方法中存在的问题是，浏览器和媒体播放器都使用HTTP服务。HTTP是运行在TCP之上的。它适合于检索元文件，但是不适合检索音频/视频文件，原因在于TCP会重发丢失或者损坏的段，这刚好与流的原理相反，所以我们不需要使用TCP和它的纠错控制，而是

应该使用UDP。HTTP用于访问Web服务器，而Web服务器则是为TCP设计的，所以需要另外一种服务器，即媒体服务器（media server）。图29.12说明了这一思想。

1. HTTP客户端使用GET报文访问Web服务器。

2. 响应中含有元文件的有关信息。

3. 元文件传送到媒体播放器。

4. 媒体播放器使用元文件中的URL访问媒体服务器以下载文件。通过任何使用UDP的协议都可以下载。

5. 媒体服务器做出响应。

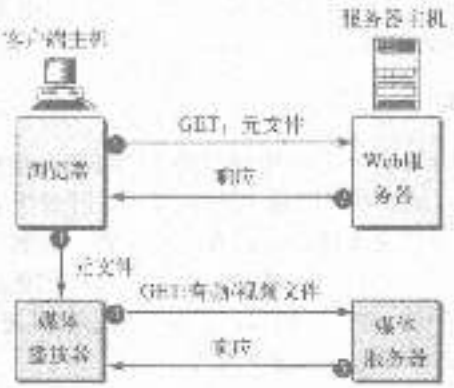


图29.12 使用媒体服务器

29.3.4 第四种方法：使用媒体服务器和RTSP

实时流协议（Real-Time Streaming Protocol，RTSP）是一种控制协议，被用来为流的处理增加更多的功能。使用RTSP，可以控制音频/视频的播放。RTSP是一种带外控制协议，与FTP中的第二种连接很相似。图29.13说明了媒体服务器和RTSP。

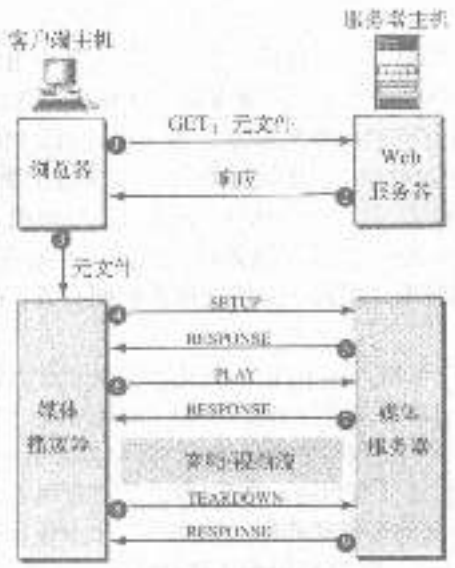


图29.13 使用媒体服务器和RSTP

1. HTTP客户使用GET报文访问Web服务器。

2. 响应中含有元文件的有关信息。
3. 元文件传送到媒体播放器。
4. 媒体播放器发送一条SETUP报文，建立与媒体服务器的连接。
5. 媒体服务器响应。
6. 媒体播放器发送一条PLAY报文以启动播放（下载）。
7. 使用一种运行在UDP之上的协议下载音频/视频文件。
8. 使用TEARDOWN报文断开连接。
9. 媒体服务器响应。

媒体播放器能够发送其他类型的报文。例如，一条PAUSE报文，能够暂时中止下载过程，通过PLAY报文，可以重新开始下载。

29.4 流式实时音频/视频

流式实时音频/视频类似于广播电台和电视台播放的音频和视频节目。区别在于电台通过无特网而不是无线电波播出。流式存储音频/视频与流式实时音频/视频之间有一些相似性，它们都对延迟很敏感，并且都不接受重传。但也有一些区别，在第一种应用情况下，通信是单播和点播的，在第二种情况下，通信是多播和实时的。实时流更适合于IP的多播服务和使用诸如UDP和RTP（后面将要讨论）这样的协议。但是目前，实时流仍然使用TCP和多个单播代替多播。在这一领域，仍然有许多需要改进的地方。

29.5 实时交互式音频/视频

在实时交互式音频/视频中，人们相互之间进行实时通信。因特网电话或者IP语音是这种应用的实例。视频会议是另一个例子，人们可以进行可视化通信和交谈。

特性

在讨论这类应用使用的协议之前，先讨论一下实时音频/视频通信的一些特性。

时间关系

分组交换网络中的实时数据，需要保留一次会话期内各分组之间的时间关系。例如，我们假定有一台实时视频服务器创建了实时视频图像，并在线发送。视频被数字化并打包分组。只有三个分组，每个分组保存10秒的视频信息。第一个分组从00:00:00开始。第二个分组从00:00:10开始，第三个分组从00:00:20开始，假定每个分组花费1秒的时间到达目的地（为了简化，夸张了一些）。接收方可以在00:00:01播放第一个分组，在00:00:11播放第二个分组，在00:00:21播放第三个分组。尽管在服务器发送和客户端在屏幕上所看到的之间有1秒的时间差，但动作基本是实时的，保留了分组之间的时间关系，而1秒的时间延迟并不重要。图29.14说明了这种思想。

但是，如果这些分组有不同的时间延迟，到达后会发生什么情况呢？例如，第一个分组在00:00:01到达（1秒的延迟），第二个分组在00:00:15时刻到达（5秒的延迟），第三个分组在00:00:27到达（7秒的延迟）。如果接收方在00:00:01时刻播放第一个分组，则会在00:00:11时刻结束播放。但是下一个分组还没有到达，要在4秒之后才会到达。远方站点观看这一视频时，在第一个分组和第二个分组之间会出现间断，同样第二个分组和第三个分组之间也有时间间断。这种现象称为抖动（jitter）。图29.15说明了这种情况。

> 由于分组之间的延迟，实时数据中会有抖动现象。

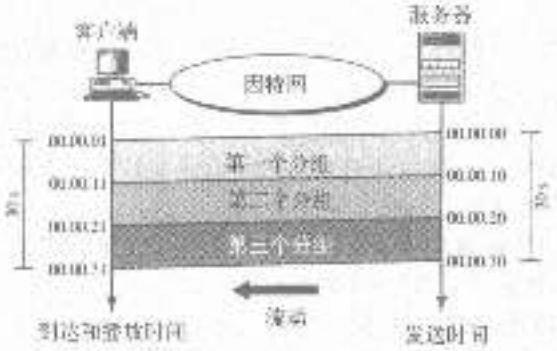


图29.14 时间关系

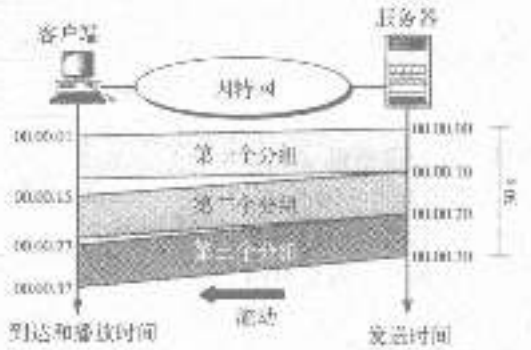


图29.15 抖动

时间戳

解决抖动的一种方案是使用时间戳（timestamp）。如果每一个分组都有一个时间戳，标明它的生成时间与第一个（或者前一个）分组的时间间隔，那么接收方就能够将它的播放开始时间加上这一段时间。换句话说，接收方能够确定每一个分组的播放起始时间。在上面的例子中，假设第一个分组的时间戳是0，第二分组的时间戳是10，第三个分组的时间戳是20。如果接收方在00:00:08时刻播放第一个分组，则第二个分组会在00:00:18时刻开始播放，第三个在00:00:28时刻开始播放。这样在分组之间就没有间断了。图29.16说明了这种情况。

为了防止抖动，可以为分组加上时间戳，将播放时间与到达时间进行分离。

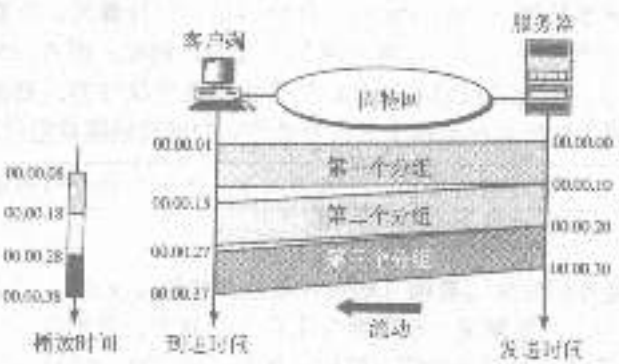


图29.16 时间戳

回放缓冲区

为了将到达时间与播放时间分离开，需要一个缓冲区存储数据直到回放开始。缓冲区是指回放缓冲区（playback buffer）。当会话开始（第一个分组的第一位到达）时，接收方会延迟播放数据，直到到达一个阈值。在上一个例子中，第一个分组的第一位在00:00:01到达，阈值是7秒，回放时刻是00:00:08。阈值通过数据的时间单元计算。重新播放会一直等到数据的时间单元数等于阈值时才会开始。

数据可能会以变化的传输速率存储在缓冲区内，但是它们是以固定速率抽取和回放的。注意，缓冲区内数据的数量可以缩小或扩大，但是只要延迟小于播放阈值时间，就不会产生抖动。图29.17说明了该例在不同时刻的缓冲区的情况。

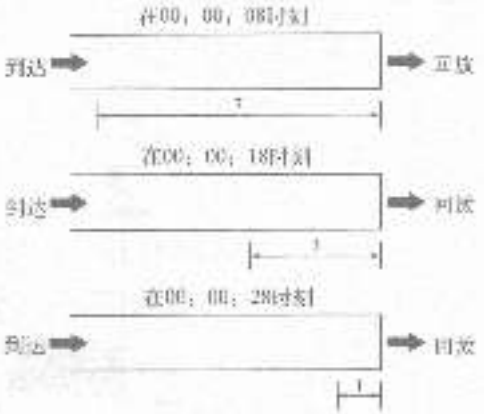


图29.17　回放缓冲区

实时通信量需要有回放缓冲区。

排序

除了时间关系信息和用于实时通信量（real-time traffic）的时间戳以外，还需要更多的特性。每一个分组需要一个序列号。只靠时间戳是不能通知接收方是否有分组丢失的。例如，假定时间戳是0、10和20。如果第二个分组丢失了，那么接收方只收到两个分组，时间戳是0和20。接收方假定时间戳为20的分组是第二个分组，也就是第一个分组20秒钟以后产生的。接收方没有办法确定第二个分组已经丢失。要处理这种情况，需要一个序列号对分组进行排序。

实时通信量需要每一个分组都有一个序列号。

多播

多媒体播放在音频和视频会议中发挥着重要作用，通信量负载可能很重。数据通过使用多播（multicasting）的方式进行分发。会议在接收方和发送方之间需要双向通信。

实时通信量需要多播的支持。

转换

有时实时通信量需要转换（translation）。转换器是一台计算机，它能够将高带宽视频信号的格式转换为低质量的窄带信号。这种转换是必要的。例如，源方以5Mbps的速率创建了一种高质量的视频信号，而是通过低于1Mbps的带宽发送给接收方。要接收到这种信号，就需要转换器将信号解码，然后再次编码成只需要低带宽的低质量视频信号。

转换的含义是改变有效载荷的编码，降低其质量以匹配接收方的网络带宽。

混合

如果有多个源方能够同时发送数据（如在视频或者音频会议中），通信量由多个数据流组成。要将通信量减少为一个数据流，就需要将来自多个源方的数据混合为一个数据流。混合器（mixer）通过数学算法将来自多个源方的信号相加，建立单一信号。

混合意味着将通信量中的多个数据流组合为一个数据流。

传输层协议的支持

前面各小节提到的各个过程在应用层都能够实现。但由于它们在实时应用中有共同的特性，因而在传输层实现应该是更可取的。现在看一下哪一种现有的传输层协议适合于这种类型的通信量。

TCP不适合交互式通信量。它不提供时间戳，也不支持多播。但是它能够提供排序（序列号）。TCP的一个使它特别不适合于交互式通信量的特性就是纠错控制机制。在交互式通信量中，不允许重新发送一个丢失或者损坏的分组。如果交互式通信量中丢失或者被破坏可分组，就必须忽略它。重发会破坏时间戳和回放的整体思想。目前在音频和视频信号中有很多的冗余（即使进行了压缩），所以可以简单地忽略一个丢失的分组。远方的收听者和观看者甚至不会察觉到这种情况。

> TCP，尽管有很多高级的特性，但它不适合于交互式多媒体通信量，原因是在交互式多媒体应用中不允许重发分组。

UDP更适合于交互式多媒体通信量。UDP支持多播并且没有重发策略。但是，UDP不支持时间戳、排序和混合。一个新的传输协议，实时传输协议（RTP）提供了这些混合的特性。

> UDP比TCP更适合于交互式通信量。但是我们需要RTP服务（另一种传输层协议）来弥补UDP的不足。

29.6 实时传输协议

实时传输协议（Real-Time Transport Protocol，RTP）是用来处理网络上实时通信的协议。RTP没有分发机制（多播、端口号等），它必须与UDP一起使用。RTP位于UDP和应用程序之间。RTP的主要作用是它的时间戳、排序和混合功能。图29.18说明了RTP在协议族中的位置。

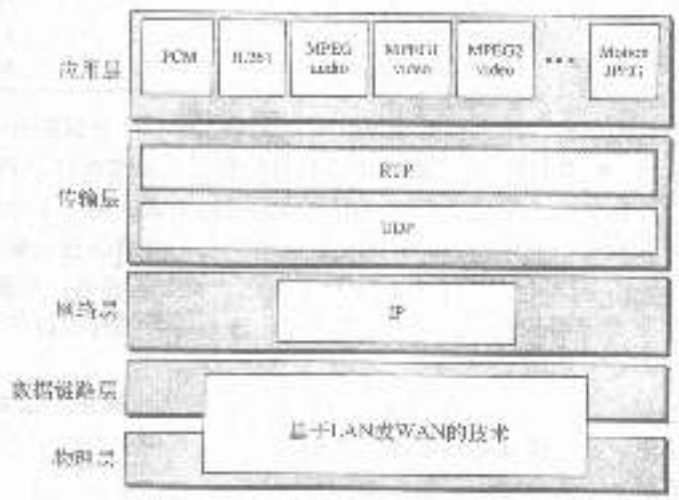


图29.18 RTP

29.6.1 RTP分组格式

图29.19说明了RTP分组头部的格式。这个格式非常简单而且通用，适合于所有的实时应用。如果应用需要更多的信息，则可以加入到头部开始的负载中。下面是每个字段的描述说明。

- 版本。这是一个2位的字段，定义了版本号。现在的版本是2。
- 填充位。这是一个1位的字段。如果值为1，分组末有填充，此时填充中最后一个字节的值为填充的长度。如果分组是加密的，则填充是标准的。如果值为0，则没有填充。
- 扩展标记位。这是一个1位的字段。如果值为1，则说明在基础头部与数据之间有额外的扩展头部；如果值为0，则没有额外的扩展头部。
- 贡献源计数（Contributor count）。这是一个4位的字段，说明了参与者的个数。注意：4位字段只能允许的最大数为15，所以最多参与者为15。
- 标志位。这1位的字段是由应用指示的标志，例如，数据的结束。
- 有效载荷类型。这是一个7位的字段，说明了有效载荷的类型。目前为止已经定义了许多有效载荷类型。在表29.1中列出了一些常用的应用。对这些类型的讨论超出了本书的范围。



图29.19　RTP分组头部格式

表29.1　有效载荷类型

类　型	应　用	类　型	应　用	类　型	应　用
0	PCMu音频	7	LPC音频	15	G728音频
1	1016	8	PCMA音频	26	Motion JPEG
2	G721音频	9	G722音频	31	H.261
3	GSM音频	10-11	L16音频	32	MPEG1视频
5,6	DVI4音频	14	MPEG音频	33	MPEG2视频

- 序列号。这是16位长的字段，用来给RTP分组进行编号。第一个分组的序列号是随机选择的；对后续的每一个分组则加1。接收者用序列号检测丢失或者次序颠倒的分组。
- 时间戳。这是32位长的字段，用来说明各分组之间的时间关系。第一个分组的时间戳是一个随机数；对随后的每一个分组，它的时间戳的值为前一分组的值与第一个字节产生的时间之和（取样值）。时钟计时单元的值依赖于应用，例如，音频应用通常产生160字节的大块，那么这个应用的时钟计时单元就是160。这个应用的时间戳的值就是每一个RTP分组累加160。
- 同步源标识符。如果只有一个源，那么这个32位就定义了源。然而，如果存在多个源，那么混合器是同步源，其他源都是贡献源。源标识符的值是由源选取的一个随机数。协议提供冲突情况下的一个策略（两个源开始具有相同的序列号）。
- 贡献源标识符。每个32位的标识符（最大值是15）定义一个源。当会话中有多个源时，混合器是同步源而其余的源都是贡献源。

29.6.2　UDP端口

尽管RTP本身是一种传输层协议，但RTP分组并非直接封装为IP数据报。相反，RTP被当

作应用程序对待，被封装成UDP用户数据报。但是，与其他应用程序不同，没有为RTP分配熟知端口。端口可以按需要选择，这里只有一个限制：端口号必须是偶数。下一个端口号（奇数）由RTP的伴随协议——实时传输控制协议（RTCP）——使用。

RTP使用的临时端口号是偶数的UDP端口。

29.7 实时传输控制协议

RTP只支持一种类型的报文，该报文负责将数据从源方传送到目的方。在多数情况下，在会话中还需要其他报文。这些报文用于控制流量和数据质量，允许接收方向一个或者多个源方发送反馈。实时传输控制协议（Real-Time Transport Control Protocol，RTCP）就是为这种目的而设计的。RTCP有五种类型的报文，如图29.20所示。紧跟在每个方格后面的数字定义了报文的类型。

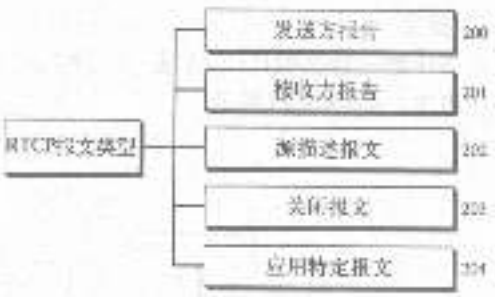


图29.20 RTCP报文类型

29.7.1 发送方报告

发送方报告由会议中的主动发送方定期发送，用来报告在一定的时间间隔内所有RTP分组的发送和接收的统计数据。发送方报告包括一个绝对时间戳，是从1970年1月1日午夜算起的秒数。绝对时间戳允许接收方同步不同的RTP报文。当音频和视频同时传输时（音频和视频传输使用单独的相对时间戳），这显得特别重要。

29.7.2 接收方报告

接收方报告是给被动参与者的。这些被动参与者不发送RTP分组。报告通知发送方和接收方有关服务质量的信息。

29.7.3 源描述报文

源方周期性发送源描述报文，提供关于自身的附加信息。这种信息可能是名字、电子邮件地址、电话号码、源控制者或者所有者的地址。

29.7.4 关闭报文

源方发送关闭报文来关闭流。它允许源方宣布要离开会议。尽管其他源方能够检测到一个源方的缺席，但是这条报文可以直接宣布。它对混合器也是非常有用的。

29.7.5 特定应用报文

特定应用报文是希望使用新的应用（没有在标准中定义）的一个应用分组。它允许定义新的报文类型。

29.7.6 UDP端口

与RTP一样，RTCP没有使用熟知的UDP端口。它使用临时端口。选择的UDP端口号必须是紧随UDP为RTP选定的端口号的那个数。它必须是一个编号为奇数的端口。

RTCP使用奇数的UDP端口号，该端口号必须是RTP选定的端口号的下一个数。

29.8 IP语音

下面主要讨论一种实时交互式音频/视频应用：IP语音（voice over IP），即互特网电话。这种思想是将因特网作为电话网络使用，并且提供一些附加功能。这种应用允许通过分组交换网络实现双方通信，从而取代了线路交换网络中的通信。SIP和H.323两种协议用来处理这种类型的通信。下面简要介绍一下这两种协议。

29.8.1 SIP

会话初始化协议（Session Initiation Protocol，SIP）由IETF设计，它是一种应用层协议，能够建立、管理和终止多媒体会话（呼叫）。它可以用来创建双方、多方或者多播会话。SIP设计上是独立于下面的传输层的，它能够运行在UDP、TCP或者SCTP之上。

报文

SIP是一种类似HTTP的基于文本的协议。与HTTP类似，SIP也使用报文。已定义的六种SIP报文，如图29.21所示。

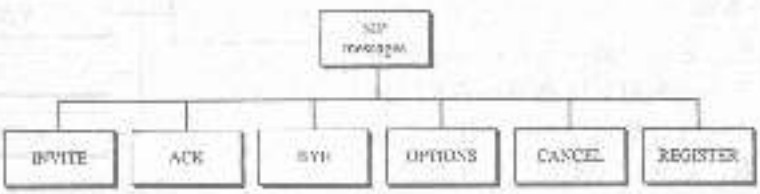


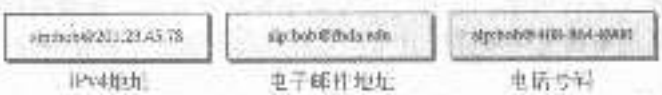
图29.21　SIP报文

每一种报文都有头部和主体。头部由多行构成，描述了报文的结构、呼叫方的功能、媒体类型等。下面给出每一种报文的简单描述，然后在简单会话中说明它们的应用。

呼叫方使用INVITE报文初始化会话。被呼叫方应答呼叫，呼叫方发送ACK报文用于确认。BYE报文可以终止会话。OPTIONS报文查询一台机器的功能。CANCEL报文取消已经启动的初始化进程。REGISTER报文会在被呼叫方不可用时建立连接。

地址

在常规的电话通信中，一个电话号码可用于识别发送方，而另一个电话号码用于识别接收方。SIP非常灵活。在SIP中，电子邮件地址、IP地址、电话号码和其他类型的地址都能用于识别发送方和接收方。但是，地址必须使用SIP格式（也就是呼叫方要）。图29.22列出了一些常用的格式。



IPv4地址　　电子邮件地址　　电话号码

图29.22　SIP格式

简单会话

使用SIP的简单会话由三个模块构成：建立、通信和终止。图29.23说明一个使用SIP的简单会话。

建立会话　在SIP中建立一次会话需要三次握手。呼叫方使用UDP、TCP或者SCTP发送INVITE报文开始通信。如果被呼叫方愿意启动会话，她会发送一条响应报文。为了确认已经接收到的回复代码，呼叫方会发送一条ACK报文进行确认。

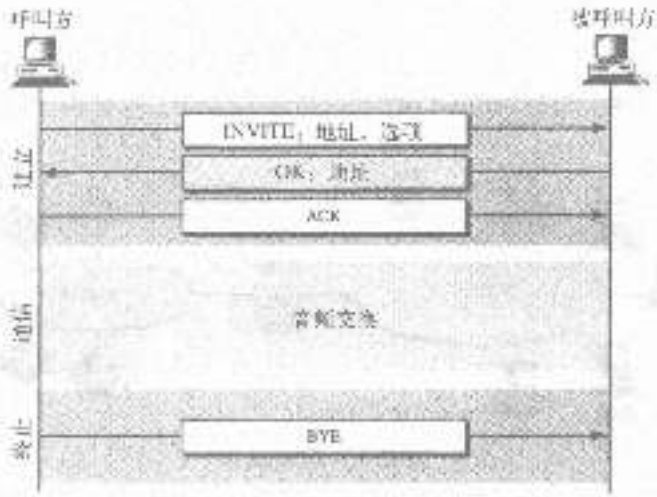


图29.23 SIP简单会话

通信 会话建立以后，呼叫方和被呼叫方能够通过两个临时端口进行通信。

终止会话 任何一方发送一条BYE报文，会话就会结束。

跟踪被呼叫方

如果被呼叫方没有坐在终端前面，会发生什么情况呢？她可能离开了自己的系统，或者在另一台终端前面。如果使用DHCP，他甚至不会有固定的IP地址。SIP使用一种机制（与DNS中的一种类似），能够查找到被呼叫方所处的终端的IP地址。要执行跟踪，SIP使用注册的概念。SIP规定了一些服务器作为**注册服务器**（register server）。任何时刻，用户至少注册到一台服务器上，这台服务器就能知道被呼叫方的IP地址。

当呼叫方需要与被呼叫方通信时，呼叫方可以用电子邮件地址代替在INVITE报文中的IP地址。报文会传送到代理服务器。代理服务器发送一条查询报文（不属于SIP的一部分）到某台被呼叫方注册的服务器。当代理服务器从注册服务器接收到回复报文时，代理服务器取出呼叫方的INVITE报文，插入新发现的被呼叫方的IP地址，然后将这条报文发送到被呼叫方。图29.24说明了这一过程。

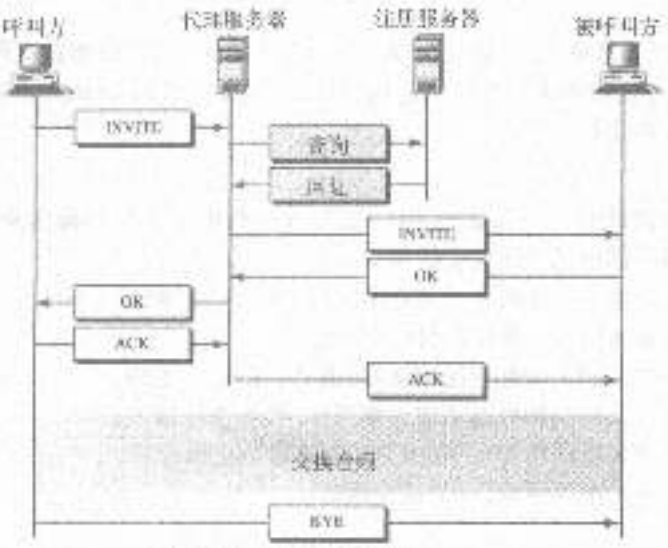


图29.24 跟踪被呼叫方

29.8.2 H.323

H.323是ITU制定的一个标准，它允许公共电话网络上的电话与连接到因特网上的计算机（在H.323中，称为终端）进行通话。图29.25说明了H.323的总体结构。

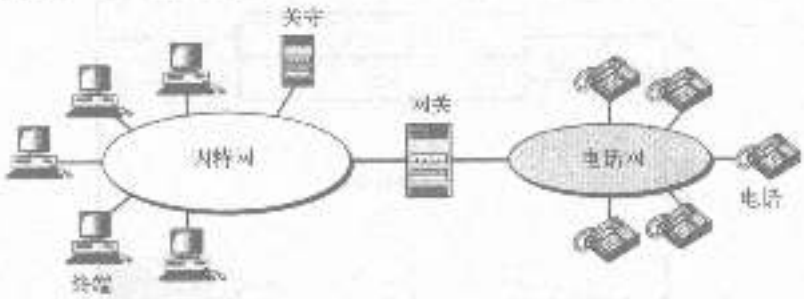


图29.25 H.323结构

网关（gateway）将因特网连接到电话网络中。通常，网关是一种五层设备，能够把报文从一个协议栈转换到另一个协议栈。此处的网关实际上做同样的事情，它将电话网络报文转换为因特网报文。本地局域网的关守（gatekeeper），承担着注册服务器的角色，正如在SIP协议中讨论的一样。

协议

H.323使用多种协议建立和维护音频（或者视频）通信。图29.26列出了这些协议。



图29.26 H.263协议

H.323使用G.71或G.723.1进行压缩。使用一个名为H.245的协议实现通信方之间压缩方法的协商。Q.931协议用于建立和终止连接。H.225或RAS（Registration/Administration/Status）协议用于与关守的注册。

操作

下面用一个简单的例子说明使用H.323协议进行电话通信的操作步骤。图29.27说明了一台计算机终端与一部电话之间通信的步骤。

1. 终端向关守发送广播报文。关守用它自己的IP地址响应。
2. 终端与关守通信，使用H.225协商带宽。
3. 终端、关守、网关和电话使用Q.931通信，以建立连接。
4. 终端、关守、网关和电话通信，使用H.245协商压缩方法。
5. 终端、网关和电话使用RTP在RTCP的管理下交换音频。
6. 终端、网关和电话使用Q.931终止通信。

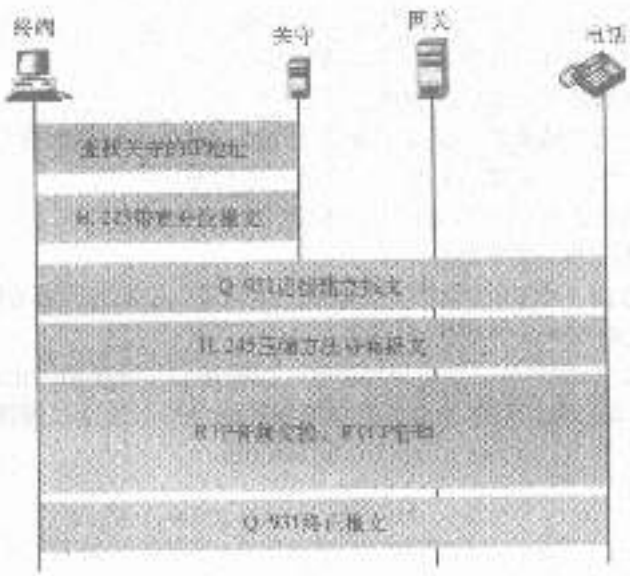


图29.27 H.323实例

推荐读物

为了深入讨论与本章有关的主题，推荐下面的读物和网站。在本书末列出方括号[……]中的参考资料。

读物

多媒体通信在[Hal01]和[Tan03]的7.4节中有讨论；图像和视频压缩在[Dro02]中讨论。

网站

www.ietf.org/rfc.html关于RFC的信息

本章小结

- 可以下载音频/视频文件以后使用（流式存储音频/视频），或者通过因特网向客户端播放（流式实时音频/视频）。因特网还可用于实时音频/视频交互。
- 音频和视频在通过因特网发送之前需要进行数字化处理。
- 音频文件通过预测编码或感知编码的方法进行压缩。
- JPEG是压缩图片和图像的一种方法。
- JPEG的处理过程包括：数据分块、离散余弦变换、量化和无损压缩。
- MPEG是一种压缩视频的方法。
- MPEG包括空间压缩和时间压缩。前者与JPEG类似，而后者去除冗余帧。
- 下载流式音频/视频文件可以使用Web服务器，或者带有元文件的Web服务器，或者媒体服务器，或者媒体服务器和RTSP。
- 分组交换网络中的实时数据需要保留一次会话期内的各个分组之间的时间关系。
- 在接收方，连续分组之间的间断会引起一种称为抖动的现象。
- 抖动可以通过使用时间戳和灵活选择回放时间来进行控制。
- 回放缓冲区会一直保存数据，直到回放开始。

- 接收方会延迟回放在回放缓冲区内保存的实时数据，直到达到设定的阈值。
- 实时数据分组中的序列号提供了一种纠错控制的方式。
- 实时数据是以多播的形式发送给接收方的。
- 实时通信量有时需要转换器，将高带宽信号转换为低质量窄带宽信号。
- 混合器可用于将来自不同源方的信号组合为一个信号。
- 实时多媒体通信量需要UDP和实时传输协议（RTP）。
- RTP负责处理时间线、排序和混合。
- 实时传输控制协议（RTCP）提供了流控制，数据质量控制并为源方提供反馈。
- IP语音是一种实时交互式音频/视频应用。
- 会话初始化协议（SIP）是一种应用层协议，能够建立、管理和终止多媒体会话。
- H.232是一种ITU标准，允许连接在公众电话网的电话与连接在因特网上的计算机之间进行通话。

习题

复习题

1. 流式实时音频/视频与流式存储音频/视频有何不同？
2. 频率遮掩与时间遮掩有什么不同？
3. 在流式存储音频/视频中，元文件有何作用？
4. 在流式存储音频/视频中，RTSP有何用途？
5. 抖动是如何影响实时音频/视频的？
6. 讨论SIP是如何用于多媒体传输的。
7. 什么情况下使用JPEG？什么情况下使用MPEG？
8. 在JPEG处理过程中，数据分块的作用是什么？
9. 在JPEG处理过程中，为什么需要DCT？
10. 与时间压缩相比，空间压缩指的是什么？

练习题

11. 在图29.17中，在下列时刻，回放缓冲区中有多少数据？
    a. 00:00:17
    b. 00:00:20
    c. 00:00:25
    d. 00:00:30
12. 试比较TCP与RTP，它们的功能相同吗？
13. 可以说UDP加上RTP就是TCP吗？
14. 为什么RTP需要另一种协议，RTCP的服务，而TCP不需要？
15. 在图29.12中，Web服务器和媒体服务器可以运行在不同的机器上吗？
16. 本章中，我们只讨论了SIP用于音频的情况，它有什么缺点阻碍它用于视频吗？
17. 你认为H.323与SIP实际上是相同的吗？不同点是什么？试在两者之间做一下比较？
18. IP语音的完整实现存在哪些问题？你认为不久的将来我们会不再使用电话网吗？
19. H.323能够用于视频吗？

探究题

20. 找出RTCP发送方报告的格式，特别注意分组的长度以及对每一个源方重复的部分。试对每一个字段进行描述。
21. 找出RTCP接收方报告的格式，特别注意分组的长度以及对每一个源方重复的部分。试对每一个字段进行描述。
22. 找出RTCP源方描述报文的格式，特别注意分组的长度以及对每一个源方重复的部分。试对每一个字段进行描述。
23. 找出RTCP源方描述分组中的源方描述项的含义，特别是找出CNAME、NAME、EMAIL、PHONE、LOC、NOTE和PRIV。
24. 找出RTCP关闭报文的格式，特别注意分组的长度以及对每一个源方重复的部分。试对每一个字段进行描述。

第七部分 网络安全

目标

在数据通信与网络中，没有人会否认安全的重要性。互联网安全的基础是密码学，它是将报文变成安全的和免于攻击的科学与技术。密码学提供了有关通过网络交换报文的一些安全问题。这些问题是：保密性、完整性、鉴别性和不可否认性。

密码学提供保密性、完整性、鉴别性和不可否认性。

密码学也可用于报文的发送方与接收方彼此鉴别。例如，需要访问系统资源的用户首先应被授权，我们称这个问题为实体鉴别。

密码学也提供实体鉴别。

本书这部分，首先介绍密码学，但对它不深入研究数学基础。然后简要探讨网络应用安全性问题。最后，我们讨论因特网模型上三层的某些协议安全实现问题。

本书第七部分介绍不同的安全问题。

第30章 密 码 学

通过使用密码学（基于抽象代数的一门科学），网络安全的大部分能够得到实现。在本章中，我们简单讨论了适合本书范围的密码学，我们尽可能地减少有关抽象代数的讨论，我们的目标是给出密码学足够多的资料使得网络安全更加易懂。本章是第31章和第32章研究的基础知识。

30.1 引言

我们先介绍一些与密码学相关的问题。首先，需要定义一些术语，然后我们给出一些分类。

30.1.1 定义

这里，我们定义本章使用的一些术语。

密码学

密码学（Cryptography）这个词在希腊语中的意思是"秘密的文字"。然而，现在这个词指的是将报文转换成安全的、免受攻击的科学和艺术。图30.1显示的是密码学的组成。

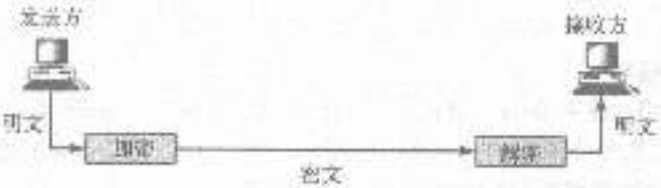


图30.1 密码学的组成

明文与密文

在进行转换之前，原始的报文称为明文（plaintext）。转换后的报文称为密文（ciphertext）。加密算法（encryption algorithm）将明文转换为密文；解密算法（decryption algorithm）将密文转换回明文。发送方使用加密算法，而接收方使用解密算法。

密码算法

我们把加密与解密算法合称为密码算法。术语密码算法（ciphers）有时也用来指密码学中不同类型的算法。这并不是说发送方和接收方在安全通信时都需要拥有自己唯一的加密算法，相反，一个密码算法可以为数百万对通信者服务。

密钥

密钥（key）是密码算法中参与运算的数值（或数值集）。对报文进行加密，我们需要一个加密算法、一个加密密钥以及明文，并由此产生密文。对报文进行解密，我们需要一个解密算法、一个解密密钥以及密文，并由此复原原始的明文。

Alice，Bob，and Eve

密码学中，在信息交换的情节里面，通常引入三个人。我们采用Alice、Bob和Eve，Alice是需要发送安全数据的人，Bob是数据接收方，Eve是个捣乱Alice和Bob通信的人，她窥探报

文或者发送旅自己伪造的报文。这三个名字代表的是实际发送或接收数据，或拦截或更改数据的计算机或者进程。

30.1.2　两类

我们可以将所有的加密算法（密码算法）分为两类：对称密钥（也称为秘密密钥）加密算法和非对称（公钥）加密算法。图30.2给出了分类。



图30.2　密码学的分类

对称密钥密码学

在对称密钥密码学中，双方使用相同的密钥。发送方使用此密钥和加密算法对明文数据进行加密，接收方使用相同的密钥和相应的解密算法对密文进行解密（如图30.3所示）。



图30.3　对称密钥密码学

> 在对称密钥密码学中，发送方（加密）和接收方（解密）使用相同的密钥。密钥是共享的。

非对称密钥密码学

在非对称或公钥加密中有两个密钥：私钥和公钥。私钥（private key）由接收方保存，而公钥（public key）是公开发布的，如图30.4所示，假设Alice想给Bob发送报文，Alice使用公钥对报文加密，Bob接收到报文后，再用私钥进行解密。

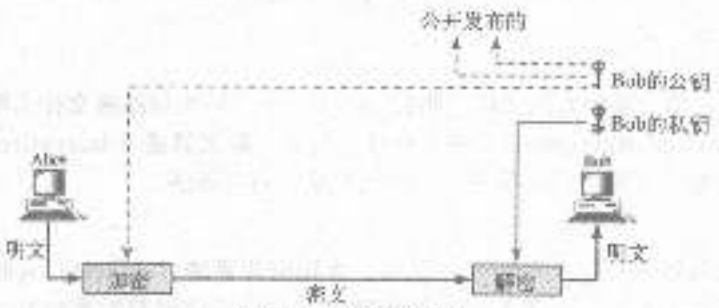


图30.4　非对称密码学

在公钥加密/解密中，用于加密的公钥与用于解密的私钥是不同的。公钥对于公众是可得到的，而私钥只有个人才能使用。

三种类型的密钥

读者应该已经注意到，在密码学中，我们要处理三种类型的密钥：秘密密钥、公钥及私钥。第一种秘密密钥是在对称密钥密码学中共享使用的；第二、三种分别是在非对称密钥密码学中使用的公钥和私钥。在本书中，我们使用三种不同的图标代表这二类密钥，将它们相互区分，如图30.5。

比较

我们比较一下对称与非对称密钥加密。加密可以想象为电子关锁，解密为电子开锁。发送方把报文放入盒子中，并使用密钥锁住盒子；接收方使用密钥打开盒子并取出报文。区别在于关锁与开锁的机制以及使用密钥的类型。

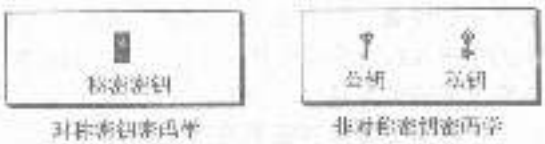


图30.5 在密码学中使用的密钥

在对称密钥密码学中，使用了相同的密钥对盒子关锁和开锁，在非对称密钥密码学中，用一个密钥锁住盒子，而用另外一个密钥打开盒子。如图30.6说明了它们之间的区别。



a) 对称密钥密码学



b) 非对称密钥密码学

图30.6 两类密码学的比较

30.2 对称密钥密码学

几千年前，当人们需要交换秘密的时候（例如在战争中），对称密钥密码学就开始诞生了。在我们的网络安全中，我们仍主要使用对称密钥密码学，然而，现代的加密算法变得更为复杂。我们首先讨论传统的算法，它是面向字符的；然后我们讨论现代的算法，它是面向位的。

30.2.1 传统加密算法

我们简单地介绍一些传统的加密算法，这是面向字符的。尽管现在看来，这些传统加密算法已经过时，但我们主要的目的是为了说明现代的加密算法是怎样从传统加密算法中演变过来。我们可以把传统的对称密钥加密算法分成两大类：替换加密算法和换位加密算法，如图30.7所示。



图30.7 传统加密算法

替换加密算法

替换加密算法（substitution cipher）使用一个符号代替另一符号。如果明文中的符号是字母，就用一个字母代替另一个。例如，可以用字母D代替A，Z代替T。如果符号是数字（0～9），则可以用7代替3，6代替2。替换加密算法可以分类为单字母替换和多字母替换加密算法。

> 替换加密算法使用一个符号代替另一个符号。

在**单字母替换加密算法**（monoalphabetic cipher）中，明文中的字符（或符号）不管其在文本中的位置，在密文中总是替换为某个固定的字符（或符号）。例如，如果算法要求在明文中的字符A必须替换为D，则每一个字符A都必须替换为字符D。换句话说，明文和密文中的字符之间的关系是一对一的。

在**多字母替换加密算法**（polyalphabetic cipher）中，明文中一个字符（或符号）的每次出现可以采用不同的字符（或符号）替换。明文中字符与密文中的字符之间的关系是一对多的关系。例如，字符A在文本的开始可以改为字符D，但在中间可以改为字符N。很明显，如果明文字符和密文字符之间的关系是一对多的话，那么密钥必须告诉我们许多可能的字符中哪些可以用来加密。为了达到这个目的，我们需要把文本分成多个字符组，并使用一套密码集。例如，我们可以把文本"THISISANEASYTASK"按每3个字符分为一组，并使用3位密码集对其进行加密。然后我们对下3个字符重复这个加密过程。

例30.1

下面所示为明文和其相应的密文，请问这是单字母替换加密算法吗？

Plaintext：HELLO

Ciphertext：KHOOR

解：

该加密算法可能是单字母替换加密。因为字符L出现的地方都加密为字符O。

例30.2

下面所示为明文和其相应的密文，请问这是单字母替换加密算法吗？

Plaintext：HELLO

Ciphertext：ABNZF

解：

该加密算法不是单字母替换加密。因为每次字符L出现的地方都加密为不同的字符。第一个L加密为N，第二个为Z。

移位加密 最简单的单字母替换加密算法可能就是**移位加密**（shift cipher）了。假设明文和密文中仅包含大写字母（A到Z），在这个加密过程中，加密算法是"后移*key*位字符"，*key*等于某一个数值；解密算法就是"前移*key*位字符"。例如，如果*key*为5，加密算法就是"后移5位字符"（向字母结束的方向），解密算法就是"前移5位字符"（向字母开始的方向）。当然，如果到了字母的结束或开始的地方，则进行环绕。

恺撒大帝使用移位加密与他的军官进行通信。正因为这个原因，移位加密有时也称为**恺撒密码**（Caesar cipher）。恺撒在他的通信中将*key*设为3。

> 移位加密有时也称为恺撒密码。

例30.3

使用移位加密算法，*key*=15，加密报文"HELLO"。

解：

我们一次加密一位字母，每一位字母后移15位，字母H加密为W，字母E加密为T，第一个字母L加密为A，第二个字母L也加密为A，字母O加密为D，因此密文为"WTAAD"。

例30.4

使用移位加密算法，key=15，解密报文“WTAAD”。

解：

我们一次解密一位字母，每一位字母前移15位。字母W解密为H，字母T解密为E，第一个字母A解密为L，第二个字母A也解密为A，最后字母D解密为O，因此明文为“HELLO”。

换位加密算法

在换位加密算法（transposition cipher）中，字符不发生替换，取而代之的是改变字符的位置，在明文中处于首位的字符可能出现在密文的第十位上，而明文中第八位字符则可能出现在密文的首位。换句话说，换位加密算法对符号分组中的符号进行重新排序。

换位加密算法对符号分组中的符号进行重新排序（改变次序）。

密钥 在换位加密算法中，密钥就是符号在明文和密文中位置的映射关系。例如，下面说明了使用了4字符分组的密钥。

Plaintext: 2 4 1 3

Ciphertext: 1 2 3 4

经过加密，我们把处于第二位的字符移位到第一位，第四位的字符移位到第二位，以此类推。解密时则相反，为了更有效率，密钥需要更长，这意味着可以对长数据分组进行加密和解密。图30.8说明了使用上面密钥的4字符分组的加密和解密。该图同时也说明了加密和解密使用了相同的密钥。加密采用了从上到下的移位，而解密则采用从下往上的移位。

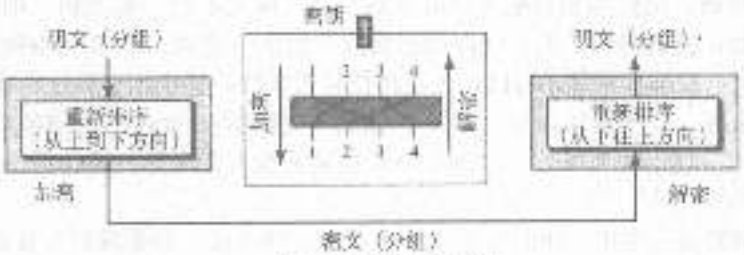


图30.8 移位加密算法

例30.5

使用上面的密钥对报文“HELLO MY DEAR”加密。

解：

我们首先把空格从报文中移除，然后把文本按照4字符为一组进行分块；在第三个分块的末尾处加上一个伪字符Z，结果就是HELL OMYD EARZ。这三个分组的密文是ELHLMDOYAZER。

例30.6

利用例30.5，解密报文“ELHLMDOYAZER”。

解：

结果是HELL OMYD EARZ，去除伪字符并加入空字符后，我们得到原始的报文“HELLO MY DEAR”。

30.2.2 简单的现代加密算法

到目前为止，我们所学的传统加密算法都是面向字符的，随着电脑的出现，加密算法必须要面向位。这是因为需要加密的信息不仅仅是文本，而且可能包含数字、图片、音频以及

视频数据，把这些类型的数据转换成位流是很方便的，对流进行加密，然后发送加密之后的位流。除此之外，当文本在位级进行处理的时候，每一位字符就被8位（或16位）替代，这意味着符号的数量变成了8个（或16个），杂乱的位比杂乱的字符能够提供更多的安全。现代加密算法与传统算法相比，使用了不同的策略。现代的对称加密算法是简单加密算法的混合。换句话说，现代的加密算法使用了许多简单的加密算法来达到加密的目的。我们首先讨论这些简单的加密算法。

XOR加密算法

如今，现代的加密算法通常是由一系列的简单加密算法（simple ciphers）组成。这些都是在数学或计算机科学中预定义的简单函数。在这里第一个讨论的是异或加密算法（XOR cipher），因为它使用了计算机科学中定义的“异或”操作。图30.9说明了异或加密算法。

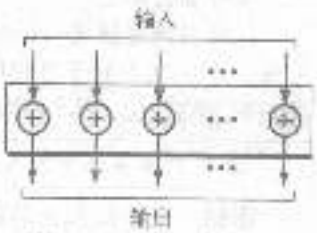


图30.9　异或加密算法

XOR操作需要两个数据输入，明文作为第一个数据，密钥作为第二个数据。换句话说，其中一个输入是需要加密的数据块，另一个输入就是密钥，运算后的结果就是加密后的数据块。注意：在XOR加密算法中，你可以发现密钥、明文及密文的长度是相同的，而且它还有一个非常有趣的属性：加密和解密是相同的。

旋转加密算法

另一个常用的加密算法是旋转加密算法（rotation cipher）。在这算法中，输入的位被向左或向右进行旋转移动。旋转加密算法可以有密钥，也可以无密钥。有密钥，则密钥的值定义了旋转移动的次数；没有密钥，则旋转移动的次数是固定不变的。图30.10举例说明了一个旋转加密算法。注意：旋转加密算法可以认为是用位替代字符的替换加密算法的特殊实例。

旋转加密算法有一个有趣的属性。如果原始流的长度为N，经过N次旋转移动后，我们得到了原始的输入流。这意味着多于$N-1$次的旋转移动是没有意义的，换句话说，旋转移动的次数必须介于1和$N-1$之间。

旋转加密的解密算法使用了相同的密钥及相反的旋转方向。如果我们在加密中向右旋转，解密时则向左旋转，反之亦然。

替换加密算法：S-盒

S-盒（替换盒）与传统的面向字符的替换加密算法是类似的。S-盒的输入是长度为N的位流；输出的结果是另一串长度为M的位流。N与M并不一定需要相等。图30.11所示为S-盒。

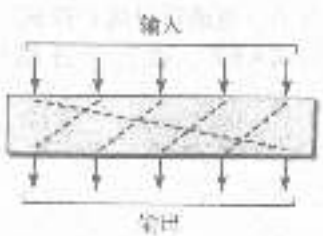


图30.10　旋转加密算法



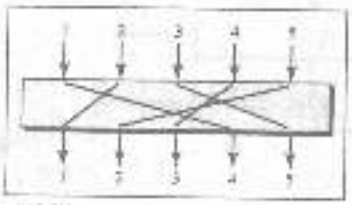
图30.11　S-盒

一般情况下，S-盒是没有密钥的，它作为加密或解密的一个中间过程。将输入与输出相匹配的函数可能是预定义的一个数学公式或者是一张表。

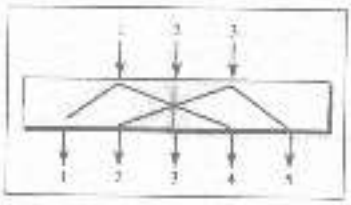
换位加密算法：P-盒

P-盒（置换盒）是位级的，与传统的面向字符的换位加密算法类似，它执行位级的换

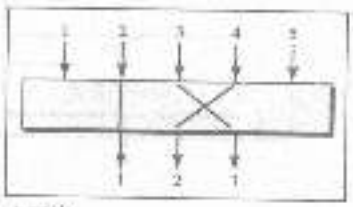
位，对位进行换位。它可以通过软件或硬件的方式实现换位，但硬件实现比较快。与S-盒类似，P-盒通常也是没有密钥的。在P-盒中，我们通常有3种类型的置换：直接置换（straight permutation）、扩充置换（expansion permutation）、以及压缩置换（compression permutation），如图30.12所示。



a) 直接



b) 扩充



c) 压缩

图30.12 P-盒：直接、扩充及压缩置换

在直接置换算法或直接置换的P-盒中，输入与输出的长度是相同的。换句话说，输入的长度是N，则输出的长度也是N。在扩充置换算法中，输出端口的数量要大于输入端口；在压缩置换算法中，输出端口的数量要小于输入端口。

30.2.3 现代迭代加密

如今的加密算法被称为迭代加密（round ciphers），因为它包含了多次迭代（rounds），每一次迭代都是由我们前面介绍的简单加密算法组成的复杂加密算法。在每次迭代中使用的密钥是一个子集或者是由称为迭代密钥的通用密钥衍生而成。如果加密算法有N次迭代，则密钥生成器产生N个密钥K_1、K_2、……、K_N，K_1在第1次迭代中使用，K_2在第2次迭代中使用，以此类推。

在本节中，我们介绍两种现代对称密钥加密算法：DES与AES。这些加密算法被认为是**分组加密**（block ciphers），因为它们把明文分成多个分组，然后使用相同的密钥对各分组进行加密与解密。到现在为止，DES已经成为事实上的标准，AES是现在正式的标准。

数据加密标准（DES）

复杂的分组加密算法的一个例子是**数据加密标准**（Data Encryption Standard，DES）。DES是由IBM设计并被美国政府接受作为非军事和非机密用途的加密标准。该算法采用64位的密钥对64位的明文进行加密，如图30.13所示。

DES有2个换位模块（P-盒）和16个复杂的迭代加密模块（它们一直在重复）。尽管16个迭代算法概念上是相同的，但每个模块使用的是由源密钥衍生的不同密钥。

在最初与最终的置换加密是无密钥的、相互之间进行反转的直接置换。这个置换加密根据预定义的值对64位的输入进行置换。

DES的每次迭代都是一个复杂的迭代加密算法，如图30.14所示，需要注意的是：加密迭代算法的结构与解密的结构是不相同的。

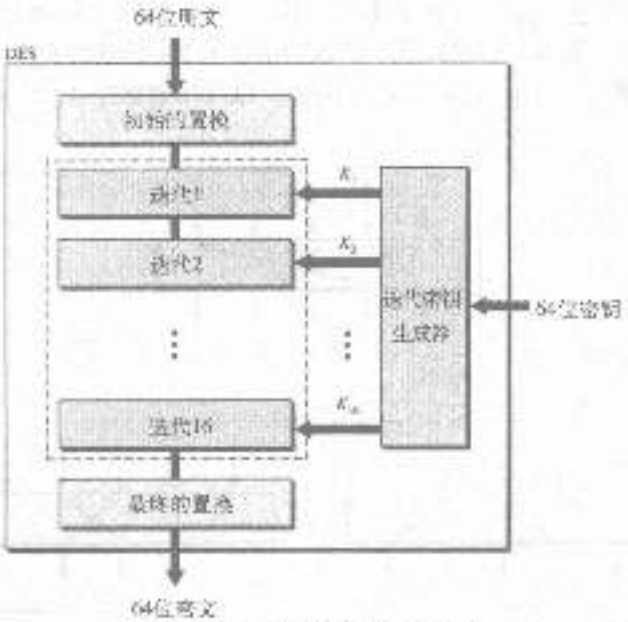


图30.13　数据加密标准（DES）

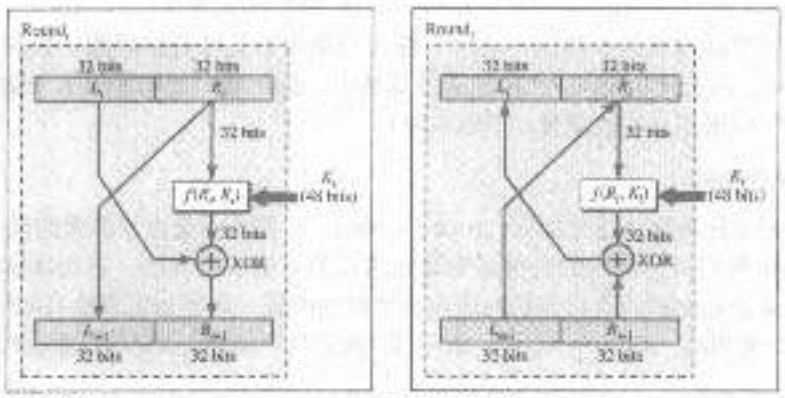


图30.14　在DES加密中的一次迭代

DES 函数

DES的核心是DES函数（DES function）。DES函数采用了48位的密钥，根据最右面32位R_i产生一个32位的输出。这个函数由4个操作组成：XOR、扩充置换、一组S-盒、以及一个直接置换。如图30.15所示。

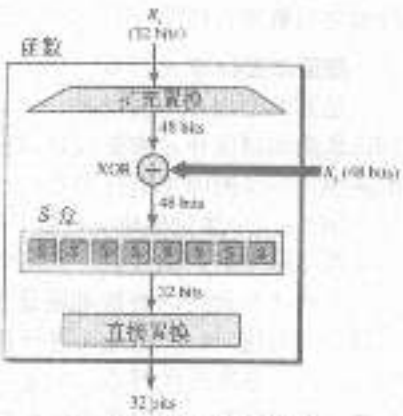


图30.15　DES函数

三重DES

对DES的批评是密钥太短了。为了加长密钥，提出并实现了三重DES或者3DES，它使用三个DES模块，如图30.16所示。注意，加密模块使用的是DES加密-解密-加密的组合，而解密模块使用的是解密-加密-解密的组合。两个不同版本的3DES分别使用：2

个密钥的3DES和3个密钥的3DES。为了使密钥长度达到112位，并且同时避免DES遭受类似中间人（man-in-middle）的攻击，设计出使用2个密钥的3DES。在这个版本中，第一个和第三个密钥是相同的（密钥1=密钥3），这有一个优势：通过一个单一的DES模块加密的文本可以用一个新的3DES模块解密，我们只是设置所有的密钥等于密钥1。许多算法使用了3个密钥的3DES加密算法，这使密钥的长度增加到168位。

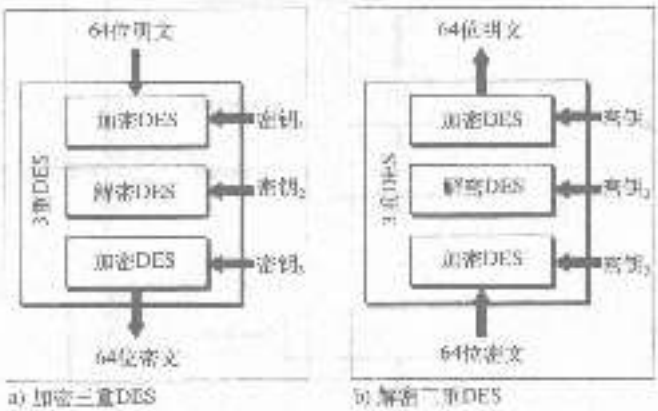


图30.16 三重DES

高级加密标准

因为DES的密钥太短，高级加密标准（Advanced Encryption Standard，AES）应运而生。尽管三重DES（3DES）增加了密钥的长度，但处理的过程太慢。国家标准与技术协会（National Institute of Standards and Technology，NIST）选择Rijndael算法（Rijndael algorithm，这算法是根据两个比利时发明人Vincent Rijmen与Joan Daemen命名的）作为AES的基础理论。AES是一个复杂的迭代加密算法，它有3种密钥长度：128位、192位及256位。表30.1所示为数据块、迭代次数及密钥长度之间的相互关系。

表30.1 AES配置

数据块的长度	迭代次数	密钥长度
128位	10	128位
	12	192位
	14	256位

AES有三种不同的配置，具有不同的迭代次数和密钥长度。

在本节中，我们仅讨论10次迭代、128位密钥的配置。其他配置的结构和操作是类似的，它们的区别就在于密钥的生成方式。

通用的结构如图30.17所示，在初始的异或操作后面是10次迭代加密，最后一次迭代与前面的迭代稍微有点区别，它缺少一步操作。

尽管10次反复的模块几乎是相同的，但每一个模块使用的是从原始密钥衍生出来的不同的密钥。

每次迭代的结构 AES的每次迭代，除了最后一个之外，都是由4个可倒转的操作组成的加密算法。最后一次迭代仅有3个操作。图30.18所示为每次迭代中操作的流程图。每次迭代中，4个操作中的每个操作都使用了一个复杂的加密算法，这个课题超出了本书的范围。

其他的加密算法

在过去的二十年里，设计并使用了许多其他的对称分组加密算法。大部分的这些加密算法有与我们本章中讨论的两种加密算法（DES和AES）相似的特征。通常情况下，它们之间

的区别在于分组和密钥的长度、迭代的次数及所用的函数，但基本原理是一致的。为了在这些加密算法的细节方面不给读者带来负担，我们对每一个加密算法给出了简单的描述。

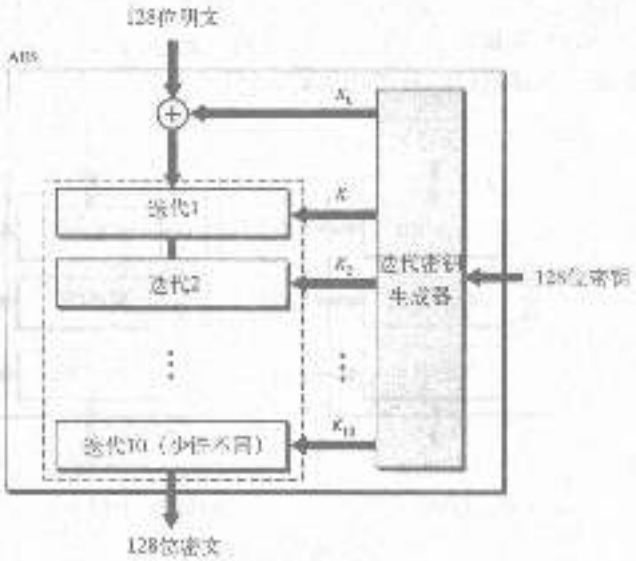


图30.17　高级加密标准

IEDA　国际数据加密算法（IEDA）是由XuejiaLai和James Massey发明的，分组长度为64位，密钥长度为128位，可以通过硬件和软件的方法实现。

Blowfish　Blowfish算法是由Bruce Schneier发明，分组长度为64位，密钥长度介于32位与448位之间。

CAST-128　CAST-128算法是由Carlisle Adams与Stafford Tavares共同发明的，它是由16次迭代的Feistel加密算法组成的，分组长度为64位，密钥长度为128位。

RC5　RC5算法是由Ron Rivest发明，这是由不同的分组长度、密钥长度、迭代次数组成的加密算法组合。

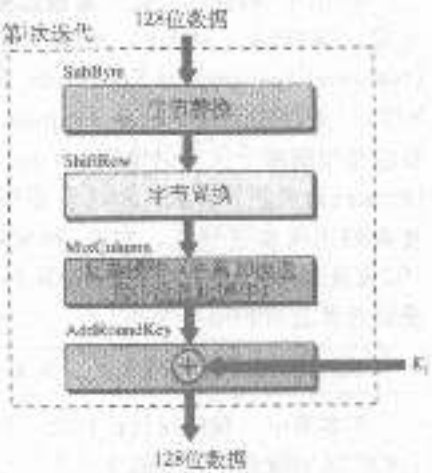


图30.18　第i次迭代的结构

30.2.4　运算模式

运算模式（mode of operation）是采用诸如我们前面讨论的DES、AES之类的现代分组加密算法的技术（如图30.19）。

电子代码本

电子代码本模式（electronic code block mode，ECB）是纯分组加密技术。明文被划分成N位长度的分组，密文也是由N位长度的分组组成。值N依赖于使用的加密算法的类型。方法如图30.20所示。

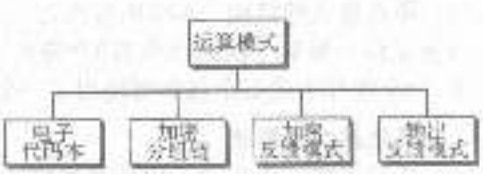


图30.19　分组加密算法的运算模式



图30.20 电子代码本模式

这个模式有4个特征：

1. 因为密钥，以及加密/解密的算法是相同的，因此在明文中相同的一些分组，在密文中对应的分组也是相同的。例如，在明文中分组1、5、9是相同的，则密文中分组1、5、9也是相同的。这就会导致了安全隐患，敌人会因为对应的密文分组相同，而猜测推断明文分组是相同的。

2. 如果我们对明文分组重新排序，则密文分组也会重新排序。

3. 每一个分组之间都是独立的。每一个分组都可以独立地进行加密与解密。其中一个分组在加密与解密时发生问题不影响其他的分组。

4. 在一个分组中发生错误不会传播到其他的分组。在传输过程中，如果一个或多个位被破坏，在加密后，这只影响这些位所对应的明文，其他的明文并没有受到影响。如果通道没有噪声，那么这将成为一个真正的优点。

加密分组链

加密分组链模式（cipher block chaining mode，CBC）通过将先前的密文分组应用到现在的分组加密中，以此来减少ECB模式中发生的问题。如果现在的分组是i，前面的密文分组C_{i-1}则应用到分组i的加密中。换句话说，当一个分组完全被加密后，分组被发送，但它的一份拷贝仍保存在寄存器（数据存放的地方）中，用于对下一个分组的加密。读者可能对最初的分组有些疑惑，在第一个分组前面是没有密文分组的。在这种情况下，使用了称为**初始化向量**（initiation vector，IV）的假冒分组。发送方与接收方都认可这个预先定义的IV。换句话说，这个IV替代了不存在的C_0。CBC模式，如图30.21所示。

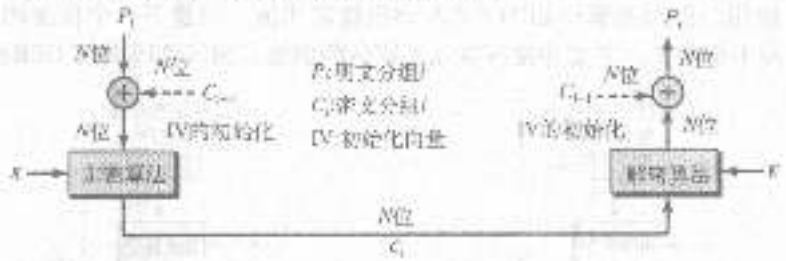


图30.21 加密分组链模式

下面是加密分组链模式的一些特征：

1. 即使密钥和加密/解密算法都相同，在明文中的相同分组对应的密文分组也不相同。例如，在明文中，分组1、5、9相同，但在密文中的分组1，5，9并不相同。敌人就不能从密文中猜测出那两个分组是相同的。

2. 分组之间是相互依赖的，每一个分组的加密与解密都是基于前一个分组。当一个分组在加密与解密中发生问题会影响到其他的分组。

3. 在一个分组中发生错误会传播到其他的分组。如果在传输过程中，一个或多个位被破坏，那么在加密后，它会影响到所有的在明文中下一分组的所有位。

加密反馈

加密反馈模式（cipher feedback mode，CFM）是专门为这个情况而设计的：我们需要发送或接收r位的数据，这里r是一个数值，不同于下面要采用的加密算法的分组长度。r的值可以是1、4、8或其他任意的位。因为所有的分组加密算法都是一次性地对数据分组进行加密。现在的问题就是如何对仅仅只有r位的数据进行加密。解决的方法是：用加密算法对分组的位进行加密，仅仅使用前面的r位作为新的密钥（密钥流），对r位的用户数据进行加密。如图30.22所示。



图30.22　加密反馈模式

下面是加密反馈模式的一些特征：

1. 对相同的明文进行加密，如果改变初始化向量，则密文是不相同的。
2. 密文C_i依赖于P_i及先前的密文分组。
3. 在密文分组中一位或多位发生错误，则会影响到后面的密文分组。

输出反馈

输出反馈模式（output feedback mode，OFB）与CFB非常相似，只有一个不同点。在密文中的每一位与前面的1位或多位都无关，这就避免了错误的传播。如果在传输的过程中发生了错误，它不会影响到后面的位。注意：在CFB中，发送方与接收方都使用了加密算法。而在OFB中，只使用分组加密算法如DES或AES创建密钥流。创建下一个位流的反馈来源于前面的密钥流，而不是密文，密文并没有参加密钥流的创建。图30.23说明了OFB模式。

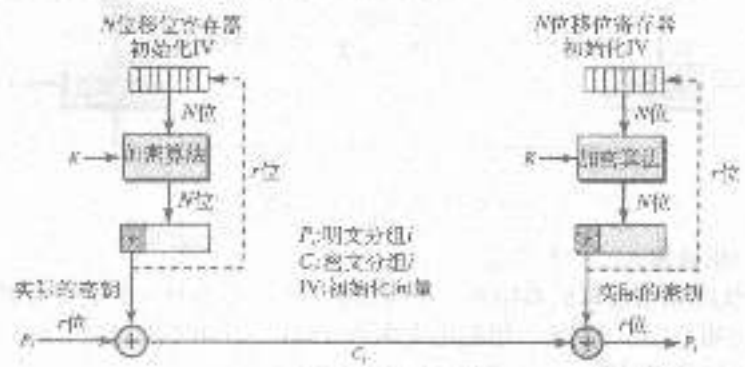


图30.23　输出反馈模式

下面是加密反馈模式的一些特征：

1. 对相同的明文进行加密，如果改变初始化向量，则密文是不相同的。
2. 密文C_i依赖于明文P_i。

3. 在密文分组中，一位或多位发生错误，并不会影响到后面的密文分组。

30.3 非对称密钥密码学

在前面几节中，我们讨论了对称密钥密码学。在本节中，我们介绍非对称密钥（公共密钥密码学）。正如我们前面所提到的，一个非对称密钥（或者公共密钥）加密算法使用2个密钥：一个是公钥，一个是私钥。我们讨论两种加密算法：RSA和Diffie-Hellman。

30.3.1 RSA

最常见的公共密钥加密算法是RSA，RSA是根据它的发明者Rivest，Shamir和Adleman（RSA）而命名的，它使用两个数字：e和d，作为公钥与密钥。如图30.24所示。

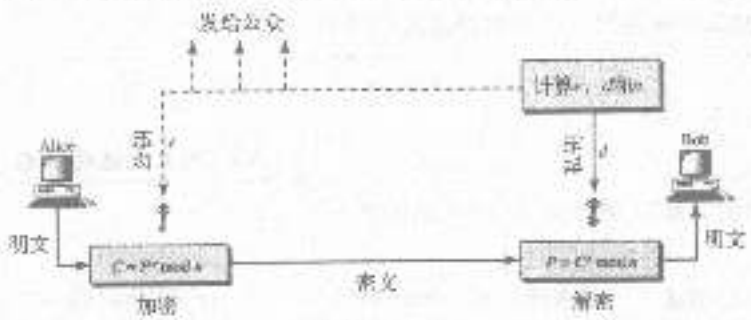


图30.24 RSA

两个密钥e和d之间有一种特殊的关系，讨论这个关系超出了本书的范围。我们说明一下怎样计算出这两个密钥而不去证明。

选择密钥

Bob使用以下的步骤选择私钥与公钥：

1. Bob选择两个非常大的素数p和q。记住素数只能被1和它本身整除。
2. Bob将上面两个素数相乘得到n，作为加密与解密的模，即$n=p\times q$。
3. Bob计算另一个整数$\phi=(p-1)\times(q-1)$。
4. Bob选择一个随机的正数e，然后计算出的d，使得$d\times e=1 \bmod \phi$。
5. Bob对外公布e和n，保留ϕ和d作为私钥。

> 在RSA中，e和n是对外公布的；而d和ϕ保留作为私钥。

加密

任何一个想发送信息给Bob的人都可以使用e和n。例如，如果Alice需要发送信息给Bob，她可以把信息转为整数（一般情况下，选择较小的整数），这就是明文。然后通过使用e和n计算出密文。

$$C=P^e \pmod n$$

Alice将密文C发送给Bob。

解密

Bob保留ϕ和d作为私钥。当接收到密文后，他使用私钥d对信息进行解密。

$$P=C^d \pmod n$$

限制

为了RSA能够正常工作，P的值必须要小于n的值。如果P是一个很大的整数，则明文需要分成多个分组，使得P小于n。

例30.7

Bob选择p=7，q=11，并计算出n=7×11==77，ϕ=(7−1)×(11−1)=60，他选择两个密钥e和d，如果e=13，则d=37。现在假设Alice发送明文5给Bob，她用公钥13对明文5进行加密。

明文：5

$C=5^{13} \pmod{77} = 26$

密文：26

Bob接收密文26，并使用私钥37对密文进行解密：

密文：26

$P=26^{37} \pmod{77} = 5$

明文：5　　Alice想要发送的信息。

Alice发送的明文是5，被Bob接收仍旧为明文5。

例30.8

Jennifer为自己创建了一对密钥。她选择p=397，q=401，计算出n=159 197，ϕ=396×400 = 158 400。然后她选择e=343，d=12 007。说明如果Ted知道了e和n，他如何发送信息给Jennifer。

解：

假设Ted想发送报文"NO"给Jennifer，他根据相应的字符码，把每一个字符转换为两位数（从00到25）。然后连接这两个字符码，得到一个四位的整数，即明文为1314。Ted用e和n对报文进行加密，密文为1314^{343}=33 677 mod 159 197。Jennifer接收到报文33 677后，使用解密密钥d对密文进行解密，如$33\,677^{12\,007}$=1 314 mod 159 197。然后Jennifer解码1314为报文"NO"。图30.25说明了过程。

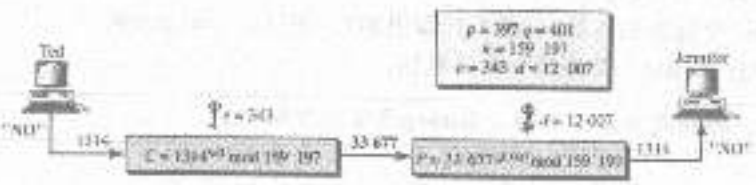


图30.25　例30.8

例30.9

我们给出一个现实的例子。我们选择512位的p和q，并计算出n与ϕ，然后选择e，并用$\phi(n)$测试相应的素数，计算出d，最后显示加密与解密的结果。我们写了一个Java的程序来做这件事情，该类型的计算是计算器不能完成的。

我们随即选择一个512位的整数，整数p是一个159位的数。

p = 9613034531358350457419158128061542790930984559499621582258315087964294045505647063849125716018034750312098666606492420191808780667421096606335421992666120 9

整数q是160位的数。

q = 12066191957231446918276794204450896001555925054637033936061798321731482
14848376465921538945320917522527322683010712069560460251388714552496900
0359660045617

我们计算n，它是309位的整数。

n = 11593504173967614968892509864615887523771457375454144775485526137614788
54083263508172768788159683251684688493006254857641112501624145523391829
27162507656772727460097082714127730434960500556347274566628060099924037
10299142447229221577279853172703383938133469268413732762200096667667183
18310883734208234443769953

我们计算ϕ，它也是309位的整数。

ϕ = 11593504173967614968892509864615887523771457375454144775485526137614788
54083263508172768788159683251684688493006254857641112501624145523391829
27162507656751054233608492916752034482627988117554787657013923444405716
98958172819609822636107546721186461217135910735864061400888517026537727
72644673410662438576864128

我们选择e=35 535，然后找到d。

e = 35535

d = 58008302860037763936093661289677917594669062089650962180422866111380593852
82235873170628691003002171085904433840217072986908760061153062025249598844
48047568240966247081485817130463240644077704833134010850947385295645071936
77406119732655742423721761767462077637164207600337085333288532144708859551
36670294831

Alice要发送报文“THIS IS A TEST”，这个报文可以通过00-26编码方案转换为数值（26为空字符）。

P = 1907081826081826002619041819

Alice计算密文为：$C=P^e$。

C = 47530912364622682720636555061054518094237179607049171652323924305445290
60613199328566617843418359114151197411252005682979794571736036101278211
88478927415660904800235071907152771859149751884658886321011483541023611
65789846796838676373376577746562507928052114814184404814184430812773050
90046928742485591664621086656

Bob通过$P=C^d$，将密文解密，P为：

P = 1907081826081826002619041819

解密后的明文解码后为：“THIS IS A TEST”。

应用

尽管RSA可以对实际的报文进行加密与解密，但如果报文很长，则加解密的过程非常慢。因而，RSA更适合那些短报文，如短的报文摘要或在对称密钥加密系统中的对称密钥。实际上，我们将看到RSA使用在数字签名和那些经常需要加密短报文而不访问对称密钥的加密系统中。在后面我们也可以看到，RSA也可以用于鉴别。

30.3.2 Diffie-Hellman 加密系统

RSA是一个公钥加密系统，并经常用于对称密钥的加密与解密。另一方面，Diffie-Hellman最初设计是用于密钥的交换。在Diffie-Hellman加密系统（Diffie-Hellman cryptosystem）中，两方产生一个对称的会话密钥（session key）用来交换数据，而且不需要为了将来的使用而必须记住或存储密钥。他们不需要面对面地对密钥达成共识，这个可以在因特网上进行。让我们看一下：当Alice与Bob需要一个对称密钥通信时，协议是如何进行工作的。在建立对称密钥之前，两方需要选择两个数p和g。第一个数p是非常大的素数，大概为300位的十进制数（1024位），第二个数是一个随机数。这两个数并不需要保密，可以在因特网上传送，并且可以是公开的。

过程

图30.26说明了整个过程，步骤如下：

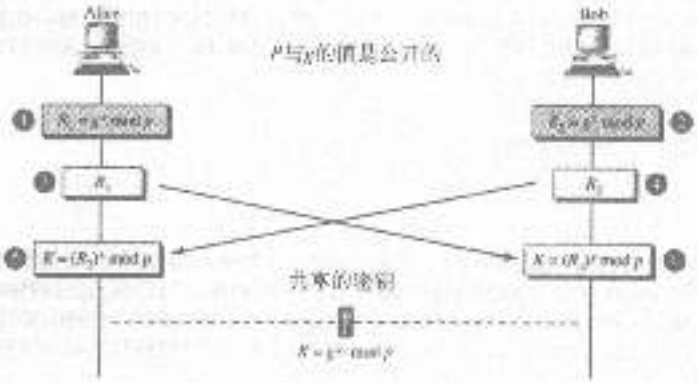


图 30.26 Diffie-Hellman方法

- 步骤1：Alice选择一个大的随机数x，并计算出$R_1=g^x \bmod p$。
- 步骤2：Bob选择另一个大的随机数y，并计算出$R_2=g^y \bmod p$。
- 步骤3：Alice发送R_1给Bob，注意Alice并不发送x的值，它仅发送R_1。
- 步骤4：Bob发送R_2给Alice，注意Bob并不发送y的值，它仅发送R_2。
- 步骤5：Alice计算出$K=(R_2)^x \bmod p$。
- 步骤6：Bob也计算出$K=(R_1)^y \bmod p$。

这个会话的对称密钥就是K。

$$(g^x \bmod p)^y \bmod p = (g^y \bmod p)^x \bmod p = g^{xy} \bmod p$$

Bob已经计算出$K=(R_1)^y \bmod p=(g^x \bmod p)^y \bmod p=g^{xy} \bmod p$，Alice也计算出$K=(R_2)^x \bmod p=(g^y \bmod p)^x \bmod p=g^{xy} \bmod p$，两个人都得到了相同的值，并且Bob不知道$x$的值，而Alice也不知道$y$的值。

> 在Diffie-Hellman协议中的对称（共享）密钥是$K=g^{xy} \bmod p$。

例30.10

我们举一个普通的例子来说清楚这个过程。例子中使用了小的数，但必须注意到现实中，数是非常大的。假设$g=7$、$p=23$。步骤如下：

1. Alice选择$x=3$，并计算出$R_1=7^3 \bmod 23=21$。
2. Bob选择$y=6$，并计算出$R_2=7^6 \bmod 23=4$。

3. Alice发送数值21给Bob。

4. Bob发送数值4给Alice。

5. Alice计算出对称密钥$K=4^3 \bmod 23=18$。

6. Bob也计算出对称密钥$K=21^6 \bmod 23=18$。

对Alice与Bob来说，K值是相同的，$g^{xy} \bmod p=7^{18} \bmod 23=18$。

Diffie-Hellman的思想

如图30.27所示，Diffie-Hellman的概念是简单但是一流的。我们可以认为在Alice与Bob之间的密钥是由三部分组成：g，x和y。第一部分是公开的。每个人都知道密钥的1/3，g是公开的值。另外两部分必须由Alice和Bob加入。每一个人加入一部分，Alice为Bob加入第二部分x，Bob为Alice加入第二部分y。当Alice从Bob接收到了全部密钥的2/3后，她在最后部分加入她的x，完成了整个密钥；当Bob从Alice接收到了全部密钥的2/3后，他在最后部分加入他的y，完成了整个密钥。注意到，尽管在Alice手中密钥由$g^y x$组成，在Bob手里密钥由$g^x y$组成，但这两个密钥是相同的，因为$g^{yx}=g^{xy}$。

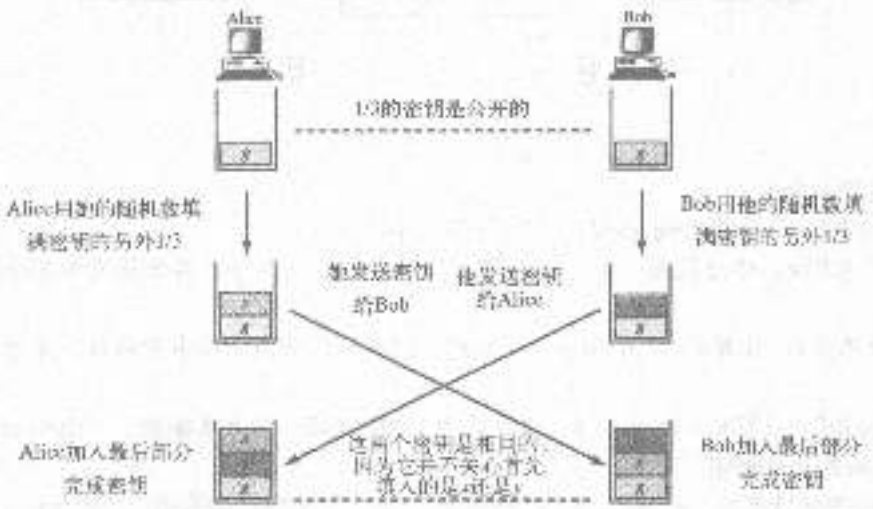


图30.27 Diffie-Hellman的概念

注意：尽管这两个密钥是相同的，但是Alice还是不能发现Bob使用的y值，因为计算是通过模p生成的；Alice从Bob接收到的是$g^y \bmod p$，而不是g^y。

中间人攻击（Man-in-the-Middle Attack）

Diffie-Hellman是一个非常复杂的创建对称密钥的算法，如果x、y是一个非常大的值，那么对Eve来说，只知道p与g的情况下来破解这个密钥是极其困难的。入侵者中途截获R_1和R_2，还需要确定x和y，但从R_1和R_2中找到x与y是两件非常困难的工作。即使一个高端的计算机也有可能花上几年的时间通过尝试各个不同的数值来找到密钥。另外，Alice和Bob在下次需要通信的时候可能已经更换了密钥。

然而，协议本身的确有自身的弱点。Eve并不需要一定要找到x与y的值来攻击通信协议。她可以通过产生两个密钥来愚弄Alice与Bob：一个密钥用于Alice与她自己之间；另一个密钥用于Bob与她自己之间。图30.28说明了这个情况。

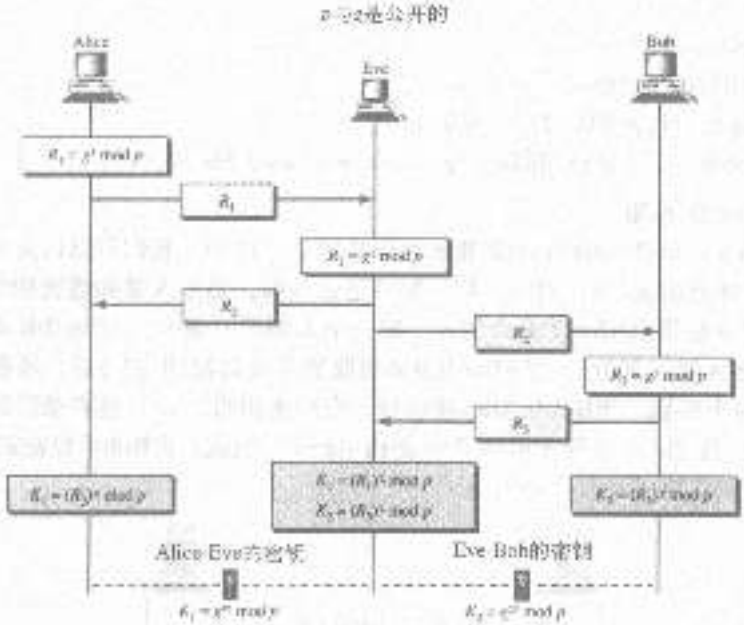


图30.28　中间人攻击

下面可能会发生：

1. Alice选择x，计算$R_1=g^x \bmod p$，并发送R_1给Bob。

2. 入侵者Eve，中途截获了R_1，她选择z，计算$R_2=g^z \bmod p$，并发送R_2给Alice和Bob两个人。

3. Bob选择y，计算$R_3=g^y \bmod p$，并发送R_3给Alice。R_3被Eve中途截获，永远不能到达Alice。

4. Alice和Eve计算$K_1=g^{xz} \bmod p$，这就成为了Alice与Eve的共享密钥。然而Alice认为这是她与Bob之间的共享密钥。

5. Eve和Bob计算$K_2=g^{yz} \bmod p$，这就成为了Eve与Bob的共享密钥。然而Bob认为这是他与Alice之间的共享密钥。

换句话说，产生两个密钥替代了原来的一个密钥：Alice与Eve之间一个，Eve与Bob之间一个。当Alice发送用K_1（Alice与Eve共享的）加密的数据给Bob时，它可以被Eve解密并读取。Eve可以发送用K_2（Eve与Bob共享的）加密的报文给Bob；或者她甚至可以更改报文并发送一个完全新的报文，Bob被愚弄并充分相信这个报文来自Alice。在另一个方向上，相同的场景也会发生在Alice方。

这种情况称为中间人攻击（man-in-middle attack）。因为Eve位于中间，他中途截取Alice发送给Bob的R_1和Bob发送给Alice的R_3。因为它类似于志愿救火队列中人与人传递水桶，所以也称为水桶队列攻击（bucket brigade attack）。

鉴别

如果Bob与Alice在第一次时相互进行鉴别，就可以避免中间人攻击。换句话说，密钥交换的过程可以结合鉴别方案，来避免中间人攻击。我们将在第31章讨论鉴别。

推荐读物

为了深入讨论与本章有关的主题，推荐下面的读物。在本节末列出方括号[ ]中的参考

资料。

读物

密码学可以在许多相关主题的书中找到，如[Bar02]、[Gar01]、[Sti02]、[Mao04]、[MOV97]、和[Sch96]。

本章小结

- 密码学是将报文进行转换以使其安全、免于遭受攻击的科学与艺术。
- 明文是变换前的原始报文；密文是经过变换后的报文。
- 加密算法是将明文转换为密文，而解密算法是将密文转换为明文。
- 加密算法与解密算法合在一起称为加密算法。
- 密钥是过程的一个数字或多个数字。
- 我们可以将加密算法分成两大类：对称密钥加密算法与非对称密钥加密算法。
- 在对称密钥加密算法中，发送方和接收方都使用相同的密钥。这个密钥就被称为秘密密钥。
- 在非对称密钥加密算法中，使用了一对密钥。发送方使用公钥；接收方使用私钥。
- 替换加密算法用另一个字符替代另一个字符。
- 替换加密算法可以分为两大类：单字母替换加密算法和多字母替换加密算法。
- 移位加密算法是最简单的单字符替换加密算法。它使用模26的模运算。恺撒加密算法就是密钥为3的移位加密算法。
- 换位加密算法对明文字符重新排序，从而生成密文。
- 异或加密算法是自身不可逆转的最简单的加密算法。
- 旋转加密算法是可逆转的加密算法。
- S-盒是输入为N位，输出为M位的无密钥的替换加密算法，并使用了公式定义了输入流与输出流之间的关系。
- P-盒是输入为N位，输出为M位的无密钥的换位加密算法，并使用表定义输入流与输出流之间的关系。P-盒只有在输入位与输出位相同的情况下是可逆转的；P-盒可以使用直接置换、压缩置换和扩充置换。
- 现代加密算法通常是迭代加密算法，每一次迭代都是由不同的简单加密算法组成的复杂加密算法。
- DES是美国政府采用的对称密钥方法，DES有初始和最终的置换模块及16次迭代。
- DES的核心是DES函数，DES函数由四部分组成：扩充置换、异或操作、S-盒及直接置换。
- DES使用密钥生成器产生16个48位的迭代密钥。
- 三重DES用来增加DES密钥的长度（有效位达到56位），提供更好的保密性能。
- AES是基于128位数据分组的Rijndael算法的迭代加密算法。AES有三种不同的配置：128位密钥的10次迭代；192位密钥的12次迭代和256位密钥的14次迭代。
- 操作模式是指部署如DES或AES等加密算法的技巧。通常的四种加密模式是ECB、CBC、CBF和OFB，ECB和CBC是分组加密算法；CBF和OFB是流加密算法。
- 通常使用的公共密钥加密方法是RSA算法，由Rivest、Shamir及Adleman发明。
- RSA选择n作为两个素数p和q的产物。
- Diffie-Hellman方法为两个组织提供了一次性的会话密钥。
- 如果两方相互之间并没有进行鉴别，那么中间人攻击可能会危及Diffie-Hellman方法的安全。

习题

复习题

1. 在对称密钥密码学中，如果Alice与Bob之间进行通信，需要多少个密钥？
2. 在对称密钥密码学中，Alice能够用相同的密钥与Bob和John两者都进行通信吗？并解释你的答案。
3. 在对称密钥密码学中，在一个10人组中的每一个人都需要对另外一个10人组中的每个人进行通信，试问需要多少密钥？
4. 在对称密钥加密学中，在一个10人组中的每一个人需要与这个组中的其他人进行通信，试问需要多少密钥？
5. 在非对称密钥密码学中，重新回答问题1。
6. 在非对称密钥密码学中，重新回答问题2。
7. 在非对称密钥密码学中，重新回答问题3。
8. 在非对称密钥密码学中，重新回答问题4。

练习题

9. 在对称密钥密码学中，你认为两个人如何在他们之间建立密钥？
10. 在非对称密钥密码学中，你认为两个人如何在他们之间建立两对密钥？
11. 用密钥为20的移位加密算法对报文“THIS IS AN EXERCISE”进行加密。忽略单词之间的空格。并对密文进行解密，得到原始的明文。
12. 如果符号仅是0和1，我们是否可以使用单字母替换加密算法？这是一个好的方法吗？
13. 如果符号仅是0和1，我们是否可以使用多字母替换加密算法？这是一个好的方法吗？
14. 使用下面的密钥，采用置换加密算法对“INTERNET”进行加密。

    3 5 2 1 4

    1 2 3 4 5
15. 将111001向右循环移3位。
16. 将100111向左循环移3位。
17. 一个S-盒有6位输入、2位输出，假定输入最低位是1，它将奇数位的输入（1，3，5，…）相加成为输出的低位，将偶数位的输入相加（2，4，6，…）相加成为输出的高位。如果输入是101101，试问输出是什么？如果输入是101101，试问输出是什么？假定最低位是1。
18. 一个S-盒有6位输入、2位输出，所有可能的输入的组合值是什么？所有可能的输出值是什么？
19. 一个S-盒有4位输入、3位输出，输入的最高位确定其余3位的循环。如果最高位是0，则其余3位向左循环移1位；如果最高位是1，则其余3位向右循环移1位。如果输入是1011，试问输出是什么？如果输入是0110，试问输出是什么？
20. P-盒使用下面的表进行加密，说明P-盒，并连接输入到输出。这个P-盒是直接置换、压缩置换还是扩充置换？

    4 2 3 1

    1 2
21. 在RSA中，给出两个素数p=19，q=23，计算n和ϕ。选择e=5，计算d，使得e和d能够符合要求。
22. 为了进一步理解RSA算法的安全性，如果知道了e=17，n=187，计算d。这个习题说明了如

果n非常小的话，对Eve来说破解这个算法还是非常简单的。

23. 在RSA算法中选择了一个非常大的n。解释：为什么Bob可以从n计算得出d，而Eve不能。
24. 在RSA算法中，e=13，d=37和n=77，采用00-25对应A-Z的编码方案，对报文"FINE"进行加密。为简单起见，逐个字符进行加密与解密。
25. 在RSA中，Bob为什么不能选择1作为公钥e?
26. 在RSA中，如果选择2作为公钥e会有什么风险?
27. Eve用Bob的公钥采用RSA发送一个报文给Bob。稍后在酒会上，Eve看到Bob并问Bob是否收到报文，Bob回答已收到。在饮酒后，Eve又问："密文内容是什么？"Bob告诉Eve密文内容。这是否危及Bob私钥的安全？试说明其理由。
28. 在Diffie-Hellman协议中，如果g=7、p=23、x=2及y=5，对称密钥的值是多少?
29. 在Diffie-Hellman协议中，如果x=y会发生什么？这就是说：Alice和Bob偶然地选择了同一个数，那么R_1和R_2的值也相等？Alice和Bob计算出来的会话密钥的值也相等？举例说明你的论点。

探究题

30. 另一个非对称密钥加密算法是ElGamal，请研究并查找关于此算法的资料，并请说明RSA与ElGamal的区别。
31. 另一个非对称密钥加密算法是基于椭圆曲线的，如果你熟悉椭圆曲线，请研究并找出基于椭圆曲线的算法。
32. 为了使Diffie-Helman算法更加便捷，有人使用了cookies。请研究并找出在Diffie-Helman算法中，是如何使用cookies的。

第31章 网络安全服务

第30章介绍了密码学，本章讨论密码学在网络安全中的一些应用。首先，我们介绍网络中典型的安全服务，然后说明如何用密码学提供这些服务。在本章末，我们还讨论对称与非对称密钥的分发问题。本章为第32章提供必要的基础，在第32章讨论因特网中的安全问题。

31.1 网络安全服务种类

网络安全能提供如图31.1所示的五种服务。其中的四种服务与使用网络的报文交换有关：报文保密性、完整性、鉴别和不可否认。第五种服务是提供实体的鉴别或认证。

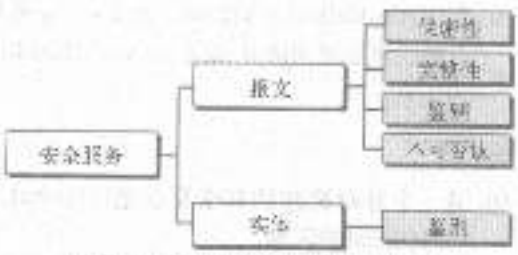


图31.1 报文或实体有关的安全服务

31.1.1 报文保密性

报文保密性（message confidentiality or privacy）是指发送方和接收方期望保持机密。传输的报文必须只对预定的接收方有意义。对所有其他人，报文都是毫无意义的。当顾客与银行通信时，他期望通信是保密的。

31.1.2 报文完整性

完整性（message integrity）是指数据到达接收方时必须和它们发送时保持一致。在传输过程中不能发生意外的或者恶意的改变。随着越来越多的货币交易在因特网上进行，完整性变得至关重要。例如，如果请求转账$100变成请求$10 000或$100 000，这将是灾难性的。在安全通信中必须保证报文的完整性。

31.1.3 报文鉴别

除了报文完整性之外，**报文鉴别**（message authentication）是指接收方需要确认发送者的身份，而不是未发送该报文的冒充者。

31.1.4 不可否认性

不可否认性（nonrepudiation）是指接收方必须能够证明报文来自特定的发送方。发送方无法否认他事实上已经发送过的报文。证明的责任落在接收方身上。例如，当顾客发送一份报文要求将资金从一个账户转到另一个账户时，银行必须拥有该顾客确实请求转账的证据。

31.1.5 实体鉴别

在实体鉴别（或用户认证）中，实体或用户在访问系统资源（例如，文件）之前核实身份。例如，需要访问学校资源的学生在登录过程中必须被核实身份，这是为保护学校与学生的利益。

31.2 报文保密性

如何实现报文保密性的观念数千年来都没有改变：报文必须在发送方加密而在接收方解

密。也就是说，报文必须处理成未授权人员无法理解的形式。良好的保密技术能够在一定程度上保证入侵者（窃听者）无法理解报文的内容。正如第30章讨论的那样，这可用对称密钥加密或非对称密钥加密。我们复习这两种技术。

31.2.1 对称密钥密码学的保密性

尽管现代的对称密钥算法比长期历史以来所用的秘密写法复杂得多，但基本原理是一致的。为了用对称密钥密码学中的密钥提供保密性，发送方与接收方必须共享密钥。在过去，两个特定人员（例如，两个朋友或一个统帅和军队的首领）可能当面交换个人的密钥。今天的通信不可能提供这样的机会。如一个居住在美国的人不可能与居住在中国的人碰面交换密钥。更进一步，上亿人之间的通信不是一个小数目。

为了能使用对称密钥密加密，我们需要解决密钥共享问题，这可用**会话密钥**（session key）来完成。这个会话密钥仅在一次会话期间使用，正如我们将在后面将看到的，会话密钥本身要使用非对称密钥加密方法进行交换。图31.2表示了Alice和Bob之间双方发送保密报文所用的会话对称密钥。注意：虽然今天我们不推荐采用对称密钥实施双方通信，但对称密钥的性质允许双方实施通信；使用两个不同的密钥更安全，因为如果一个密钥被泄露，那么在另一方向的通信保密性还依旧存在。



a) Alice到Bob通信所用的共享密钥



b) Bob到Alice通信所用的另一个不同的共享密钥

图31.2 双方采用对称密钥报文的保密性

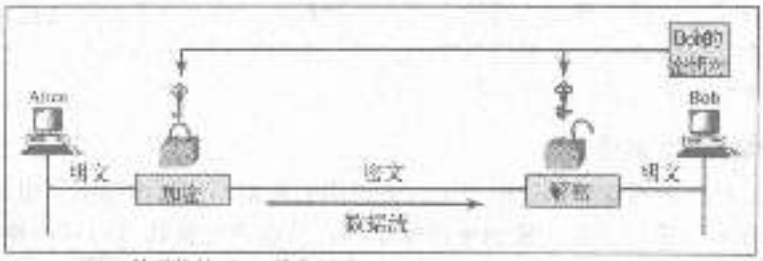
对称密钥加密仍是报文保密性的主要方法，其原因是它的效率。特别是对于长报文，对称密钥加密比非对称密钥加密的效率更高。

31.2.2 非对称密钥密码学的保密性

在对称密钥加密中，我们提到密钥交换是通过非对称密钥加密的保密性来完成。在非对称密钥加密过程中，没有共享密钥，而公钥是公开分发的。Bob创建两个密钥：一个是私钥，另一个是公钥。他保留私钥用于解密，向所有人公布他的公钥。公钥仅用于加密，而私钥仅用于解密。公钥锁定报文，私钥解锁报文。

Alice与Bob之间双向通信需要二对密钥。当Alice向Bob发送报文时，她使用Bob的公钥。当Bob向Alice发送报文时，他使用Alice的公钥。如图31.3所示。

非对称密钥加密的保密性存在两个问题：第一，该方法基于用长密钥进行复杂的数学计算，就是说，对于长报文，系统效率较低，它只适用于短报文的加密。第二，报文的发送还需要接收方的公钥。例如，Alice到Bob的通信，Alice需要确信Bob的公钥是真实的，因为Eve可用Bob的名义发布她自己的公钥。在本章后面将看到，系统需要信任。



a) Alice到Bob的通信使用Bob的密钥对



b) Bob到Alice的通信使用Alice的密钥对

图31.3　使用非对称密钥对的报文保密性

31.3　报文完整性

加密与解密提供保密性，但没有提供完整性（integrity）。但在有些场合下，我们甚至不要求保密性而要求完整性。例如，Alice立下关于她死后财产分配的遗嘱。遗嘱不要求保密性，死后任何人都可查阅遗嘱。但必须要保持遗嘱的完整性，Alice不希望遗嘱的内容有任何篡改。另一个例子，假定Alice发送一个报文通知银行工作人员Bob支付给Eve咨询费，该报文不需要对Eve保密，因为她知道要付钱给她。但是，报文需要安全不能被篡改，特别不能被Eve篡改。

31.3.1　文档与指纹

保持文档完整性的一个方法是使用指纹（fingerprint）。如果Alice需要确保文档的内容不被非法篡改，她可在文档末端按下她的指纹。Eve不能篡改该文档的内容，或者伪造文档，因为她不可能伪造Alice的指纹。要确信文档没有被篡改可比较文档上的Alice的指纹与文件上的Alice的指纹。如果它们是不同的，则该文档不是Alice的。

> 为了保持文档的完整性，需要文档与指纹两者。

31.3.2　报文与报文摘要

电子文档与指纹等价于报文（message）和报文摘要（message digest）。为了保持报文的完整性需要使用称为散列函数（hash function）的算法，创建报文的压缩映像用作指纹。图31.4表示了报文、散列函数与报文摘要。



图31.4　报文与报文摘要

31.3.3　差异

文档/指纹对与报文/报文摘要对有一些相似之处，但也有一些差异。文档与指纹在物理上是连在一起的，而且两个都不需要保密，而报文与报文摘要是彼此独立的（发送）。最重要的是报文摘要保密。如果需要通过通信通道发送报文摘要，那么报文摘要需要置于安全的地方加以保密或加密。

报文摘要需要保密。

31.3.4 创建和检验报文摘要

报文摘要在发送方站点创建，它与报文一起发送到接收方。为了检验报文或文档的完整性，接收方对报文使用相同的散列函数算法再次产生报文摘要，与接收到的摘要比较。如果两份摘要完全相同，则接收方确信原有报文没有被改变。当然，我们假定摘要是加密发送。图31.5表示了这个思想。

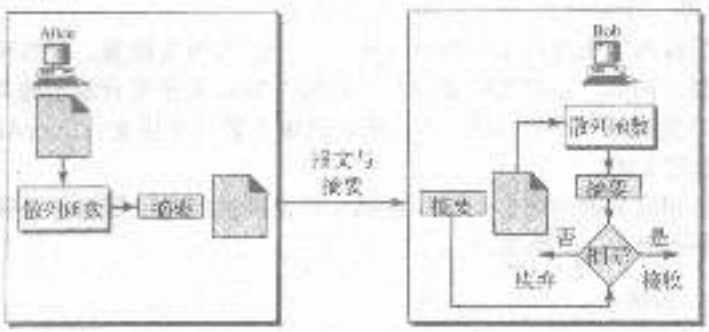


图31.5 检验完整性

31.3.5 散列函数准则

满足安全要求的散列函数必须有三个条件：单向性、抗弱碰撞攻击和抗强碰撞攻击。如图31.6所示。



图31.6 散列函数准则

单向性

散列函数必须具有单向性（one-wayness），用单向散列函数创建的报文摘要，不能从报文摘要中重新求出原来的报文。要百分之百做出单向散列函数有时是困难的。准则表明如果已知报文摘要，则求出原来的报文极其困难或不可能。这种情况与文档/指纹类似，没有一个人能从指纹求出文档。

例31.1

能否将传统的无损压缩方法作为一个散列函数？

解：

不可以。因为无损压缩报文是可逆的，可对压缩报文进行解压缩求出原来的报文。

例31.2

能否用校验和方法作为一个散列函数？

解：

可以，校验和函数是不可逆的。它满足第一个条件，但是不能满足其余的条件。

抗弱碰撞攻击

散列函数第二个准则是抗弱碰撞攻击（weak collision），确保一个报文不容易被伪装。如果Alice创建报文和摘要，并向Bob发送这两者，Eve不容易创建另一个报文，对它进行散列产生完全相同的报文摘要。换言之，任意给定一个报文以及该报文创建的摘要，不可能找到另一个报文，它的摘要与前一摘要相同，这在计算上是不可行的或很困难的。

当两个不同的报文能创建相同的摘要时，我们称为碰撞。在弱碰撞中，已知一个报文摘要，某人想创建一个报文具有完全相同摘要是极其不可能的，散列函数必须具有抗弱碰撞攻击性质。

抗强碰撞攻击

散列函数第三个准则是抗强碰撞攻击（strong collision），确保任意二个不同报文，不会有相同的报文摘要。需要这个准则是为了保证报文发送方Alice不能伪造她自己的报文引起问题。如果Alice可创建具有相同摘要的两个不同报文，她可否认发给Bob的第一个报文，声明她仅向Bob发送了第二个报文。

因为这种类型碰撞的概率比前一种情况更高，所以称为强碰撞。对方可创建两个不同报文具有相同的摘要，例如，当报文摘要位长较短时，Alice完全有可能创建具有相同摘要的两个不同的报文。她发送第一个报文给Bob，而自己保存第二个报文，以后Alice可以说第二个报文是她原来同意的文档。

假定创建具有相同文摘的两份不同的遗嘱，当遗嘱执行时，第二份遗嘱呈给继承人。因为两份遗嘱的摘要是相同，偷换成功。

31.3.6　散列算法：SHA-1

虽然已设计出许多散列算法，但常用的是SHA-1。**安全散列算法1**（Secure Hash Algorithm，SHA-1）是美国国家标准与技术协会（NIST）修订版，并由联邦信息处理标准（FIPS）发布。

该算法与其他算法一样，其要点都遵循同一个概念。每一个算法都创建N位长的摘要。一个报文被分成多个块，每块512位。如果报文长不是512位的倍数，填充0扩充成512位的整数倍，如图31.7所示。

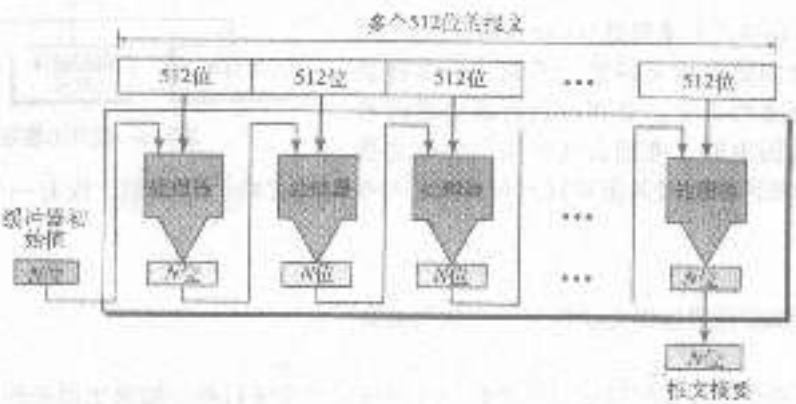


图31.7　创建报文摘要

在N位的缓冲器预置一个初始值。算法将该初始值与报文的第一个512位通过杂凑运算生成第一个N位的中间报文摘要。然后该中间报文摘要与第二个512位的块通过杂凑运算生成第二个N位的中间报文摘要，第$n-1$个N位中间报文摘要与第n个512位的块通过杂凑运算生成第n个N位的中间报文摘要。最后一个块处理所生成的摘要就是整个报文的摘要。SHA-1算法的摘要长度是160位（5个字，每字32位）。

> SHA-1算法是对超过512位的报文创建N位报文摘要。
> SHA-1算法的报文摘要是160位（5个字，每字32位）。

字扩展

在处理前，需要对块进行扩展，一个块是由512位或16个32位字组成，但在处理阶段要求80个字。因此16个字块需要扩展到80个字，即字0到字79。

每块的处理过程

图31.8表示了每块处理过程总的概要。处理过程共计80步。每一步将扩展块中的一个字与一个32位常数进行杂凑运算创建一个新的摘要。开始时，摘要字的值（A、B、C、D和E）保存在5个临时变量中。处理结束（第79步）时，这些变量与第79步生成的值相加。每步详细情况超出了本书讨论的范围。我们只需了解每一步中一个数据字与一个常数通过杂凑运算创建一个结果送到下一步。

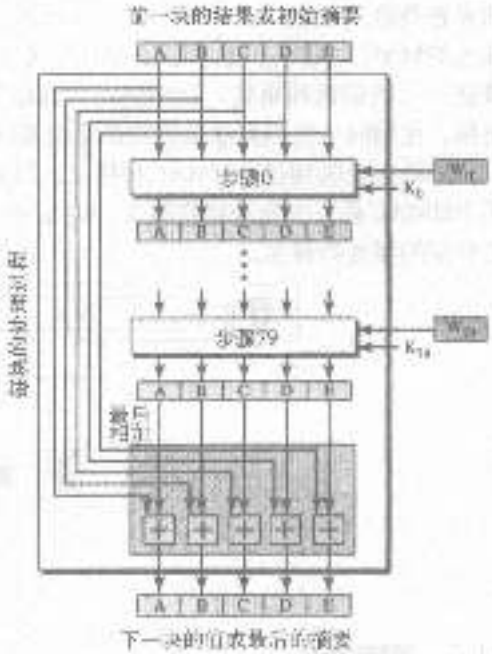


图31.8 SHA-1算法中一个块处理过程

31.4 报文鉴别

散列函数保证报文的完整性，它保证报文不被篡改，但是散列函数不能鉴别报文发送者的身份。当Alice与Bob发送报文时，Bob需要知道是否真的来自Alice，还是Eve伪造的。为了提供报文鉴别，Alice需要提供证据证明该报文确实是Alice发送的，而不是其他人假冒的。散列函数本身不能提供这样的证明，由散列函数创建的摘要称为**修改检测码**（modification detection code，MDC）。这个码可检测报文中的任何修改。

MAC

为了提供报文鉴别，我们需要将修改检测码改变成报文鉴别码（message authentication code，MAC）。MDC使用无密钥散列函数，MAC使用密钥散列函数。密钥散列函数包括当创建摘要时在发送方与接收方之间的对称密钥。图31.9表示了Alice如何使用密钥散列函数鉴别她的报文以及Bob如何核实报文的真实性。

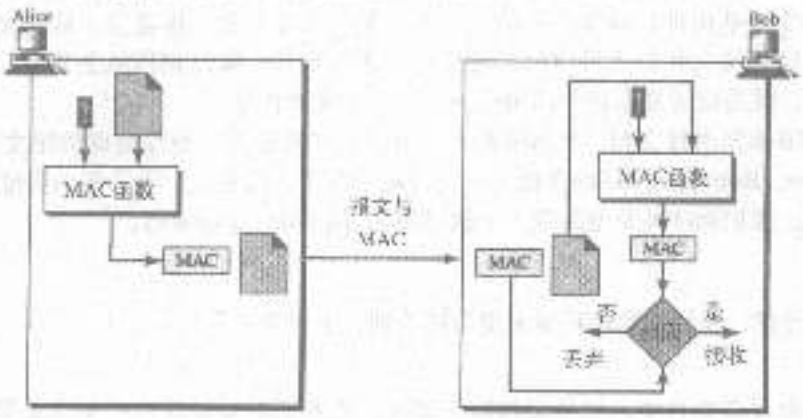


图31.9 Alice创建MAC和Bob检验MAC

Alice使用她与Bob之间的对称密钥（K_{ab}）和密钥散列函数生成MAC，然后她将原来的报文与这个MAC一起发送给Bob。Bob接收到报文与MAC后，他将报文与MAC分离。他利用对称密钥对报文应用同一密钥散列函数获得新的MAC，并将它与接收到的MAC比较。如果相同，报文没有被修改，而发送方确实是Alice。

HMAC

当今所用的MAC有多种实现方法，但是近几年来已设计出一些基于无密钥散列函数的MAC，如SHA-1。这个思想是散列MAC（hashed MAC），称为HMAC，它直接使用任何标准的无密钥散列函数，如SHA-1。HMAC用无密钥散列函数对报文与对称密钥串进行运行创建嵌入的MAC。图31.10表示了该思想。将预先约定的对称密钥的副本链接到报文，对这样的链接使用无密钥散列函数，如SHA-1。其结果是一个中间HMAC。再用密钥（同一密钥）与它链接，使用同一散列算法获得的最后结果就是HMAC。

接收方接收到这个HMAC和报文。接收方从接收到的报文创建自己的HMAC，并比较这两个HMAC是否一致来核实报文的完整性并鉴别原有数据。注意：HMAC详细情况比我们此处介绍的要复杂得多。

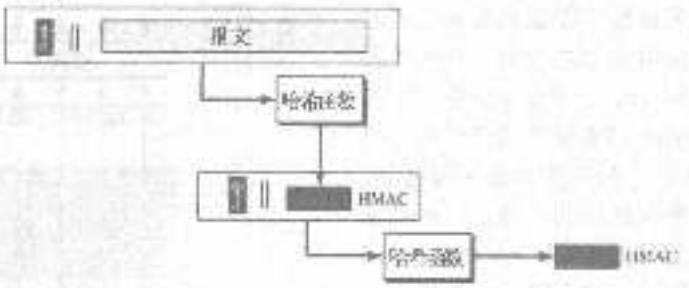


图31.10 HMAC

31.5 数字签名

虽然MAC能提供报文完整性与报文鉴别，但它有一个缺点，它必须要在发送方与接收方之间建立对称密钥。此外，数字签名可使用一对非对称密钥（一个公钥与一个私钥）。

我们都熟知签名的概念，在文档上签名表示该文档是我们发出的，或是被我们认可的。签名向接收方证实文档来自正确的实体。当一个顾客在支票上签上他的名字时，银行需要通过检验确信该支票是由那位顾客签发的，而不是其他人签发的。换言之，核实文档上的签名是鉴别符号，即核实文档是否可信的。考虑一个艺术家在一幅绘画作品上签名，如果作品上签名是可信的，就是说这幅绘画作品确实是这个艺术家的作品。

当Alice向Bob发送报文时，Bob需要通过检验核实发送方。他需要确信报文来自Alice而不是窃听者Eve。Bob可请求Alice在报文上进行电子签名。换言之，电子签名可证明Alice发送报文的真实性。我们称这种类型的签名为**数字签名**（digital signature）。

31.5.1 比较

在继续讨论前，我们考虑两种签名类型的差别：手书签名和数字签名。

包含性

手书签名包含在文档中，它是文档的一部分。当我们填写支票并在支票上签名时，签名不是独立部分。另一方面，当我们数字地签署文档时，我们将签名作为独立文档发送。发送方发送两个文档：报文与签名。接收方接收到两个文档，并核实签名是否属于假定的发送方。

如果是真的，保存报文；否则拒绝。

验证方法

两种不同签名类型的第二个不同点是验证方法。在手书签名中，当接收方接收到文档时，她将文档的签名与文件中的签名比较。如果相同，则文档是可信的。接收方需要有文件上这个签名的副本用于比较。在数字签名中，接收方接收到报文和签名。没有什么地方保存签名副本。接收方需要对报文与签名的组合用验证技术去核实其真实性。

关系

在手书签名中，签名与文档之间构成一对多的关系。例如，某人有一个签名，可用于签发多张支票、多个文档等等。在数字签名中，签名与文档之间是一对一的关系。每个文档有它自己的签名，一个文档的签名不能用于另一个文档。如果Bob一个接着一个地接收到来自Alice的二个文档，他不能用第一个报文的签名去验证第二个报文。每一个报文需要新的签名。

二重性

两种不同签名类型的另一个不同是称为二重性的性质。在手书签名中，被签文档的签名可用文件上原来的签名来识别。在数字签名中，除非在文档上有时间因素（如时间戳），否则没有这样的特性。例如，假定Alice发送一个文档指示Bob付款给Eve。如果Eve截取文档与签名，稍后她重发自从Bob处获得付款。

31.5.2 需要密钥

在手书签名中，签名好像一个属于文档签名者的专用“私钥”。签名者用它签名文档，任何其他人都不具有这样的签名。签名的副本好像是一个公钥，任何人都可以用它与原有签名进行比较来验证文档。

在数字签名中，签名者用她的私钥（即应用签名算法）签名文档。另一方面，验证者用签名者的公钥（即应用验证算法）验证文档。

我们能否用秘密（对称）密钥来实现签名与验证签名呢？回答是否定的，有几点理由。第一，秘密密钥只有通信两个实体（例如，Alice与Bob）知道，于是如果Alice需要签名另一个文档发送给Ted，又需要另一个秘密密钥。第二，正如我们将要看到的，为一次会话创建的秘密密钥包括了鉴别，鉴别通常是用数字签名实现，这样就形成了恶性循环。第三，Bob可用他与Alice之间秘密密钥，伪装Alice发送文档给Ted。

> 数字签名需要一个公钥系统。

31.5.3 过程

数字签名有两种，一种是对整个文档签名，另一种是对文档摘要签名。

签名整个文档

签名整个文档可能比较容易，但效率低。签名文档是用发送方的私钥加密，而验证一个文档则用发送方的公钥进行解密。签名与验证，如图31.11所示。

我们应该区分数字签名中所用的私钥和公钥与保密性加密中所用的私钥与公钥。在保密性过程中，使用接收方的私钥与公钥。发送方用接收方的公钥加密，接收方用自己的私钥解密。在数字签名中，使用发送方的私钥与公钥。发送方用她的私钥，而接收方用发送方的公钥。

> 在加密系统中，我们使用接收方的私钥与公钥。
> 在数字签名中，我们使用发送方的私钥与公钥。



图31.11　在数字签名中，签名报文本身

签名报文摘要

在前面加密系统中提到，报文很长时，使用公钥的效率很低。在数字签名系统中，报文通常是很长的，但我们还必须使用公钥。解决办法不是签名整个报文，而是签名报文摘要。我们已知道，精心计算的摘要与报文是一对一的关系，发送方签名报文摘要，接收方验证报文摘要的签名。图31.12表示了数字签名系统中的签名摘要。

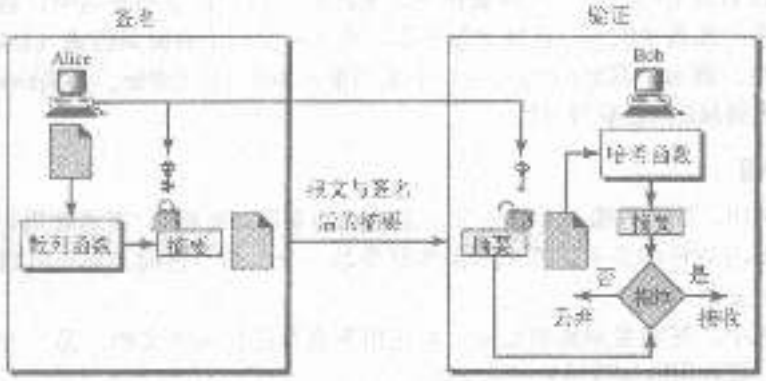


图31.12　数字签名系统中的签名摘要

在Alice站点从报文中创建报文摘要，然后用Alice的私钥签名摘要，再向Bob发送报文和签名后的摘要。在本章后面将会看到这个过程与系统有关，不同系统有一些差异。例如，在创建摘要前可能还有一些附加的计算，或者使用其他的密钥。在某些系统中，签名是一组值。

在Bob站点，首先用同一个散列函数从接收到的报文中创建摘要，然后对鉴别和摘要进行计算。验证过程还将标准应用于计算结果以确定签名的真实性。如果是可信的，接收报文，否则拒绝。

31.5.4　服务

数字签名提供安全系统五种服务中三种：报文的完整性、报文鉴别和不可否认性。注意：数字签名方案没有提供保密的通信。如果要求保密，那么报文和签名必须用秘密密钥加密或公钥加密系统进行加密。

报文完整性

如果我们签名整个报文，那么就保持了报文的完整性。因为如果报文被修改了，那么就不能得到相同的签名。当今的数字签名方案在签名与验证算法中使用散列函数，保持报文的完整性。

> 当今的数字签名提供报文完整性。

报文鉴别

一个安全的签名方案如手书签名一样（任何人都不易模仿）能提供报文鉴别。Bob因为可用Alice的公钥验证Alice发送的报文，Alice的公钥不可能创建出用Eve的私钥相同的签名。

> 数字签名提供报文鉴别。

报文不可否认性

如果Alice签名一个报文，事后又否认它，Bob是否可证明Alice的确签名过？例如，Alice发送一个报文给银行工作人员Bob请求从她的账号转$10 000到Ted的账号，事后Alice是否可赖掉她发送过该报文？到目前为止我们这个方案，Bob可能有一个问题，Bob必须保留文件上的签名和以后要用的Alice的公钥，他当它们可以创建原始报文以证明文件中的报文与新创建的报文是相同的。因为Alices可在这个期间更换她的私钥/公钥，还可声称文件包含的签名是不可信的，所以这是不可行的。

一种解决办法是要有一个可信任的第三方，人们在他们活动中形成一个可信任方。在第32章中，我们将看到，可信任方可解决许多问题如安全服务和密钥交换。图31.13表示了可信任方如何防止Alices否认她发送过的报文。

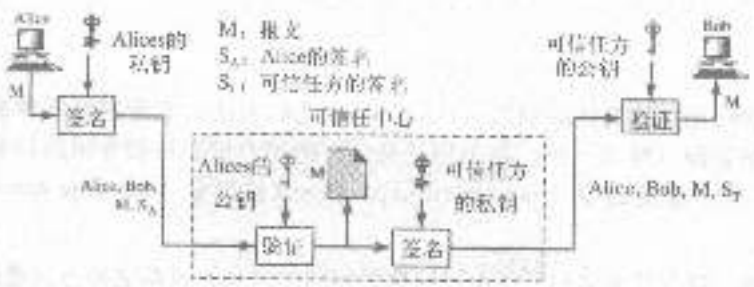


图31.13 使用可信任中心防否认

Alices对报文签名（S_A）并向可信任中心发送报文、Alices和Bob标识符以及签名。可信任中心验证Alices的公钥有效后，通过Alices的公钥验证报文是来自Alices。然后可信任中心保存报文副本，发送方标识符，接收方标识和时间戳在档案中。可信任中心使用它的私钥创建报文的另一个签名（S_T）。然后，中心向Bob发送该报文、新的签名、Alices的标识符和Bob标识符。Bob用可信任中心的公钥验证报文。

如果将来Alices否认她发送过该报文，可信任中心可出示一份保存的报文副本。如果Bob的报文是保存在信任中心的报文副本，则Alices将无法争辩。为了使每一报文都是可信任的，在下一节对方案增加加密/解密。

> 用可信任的第三方提供不可否认性。

31.5.5 签名方案

最近几十年来，逐渐地设计出一些签名方案。其中有些已实现，如RSA与DSS（数字签名标准）方案。后者可能成为今后的标准，而这些方案的讨论超出了本书范围。

31.6 实体鉴别

实体鉴别是让一方证明另一方的身份的一种技术。一个实体可以是人、进程、客户机或服务器，需要证明身份的实体称为**申请者**（claimant），试图证明申请者身份的一方称为验证

者（verifier）。当Bob试图证明Alices身份时，Alices是申请者，Bob是验证者。

报文鉴别与实体鉴别（entity authentication）有两点不同。第一，报文鉴别不提供实时性，实体鉴别是实时性。在前者，Alices向Bob发送报文，Bob鉴别报文时，在通信过程中Alices可以出现也可以不出现。而Alices请求实体鉴别时，在Bob鉴别Alices之前，没有实际的报文通信。在鉴别过程中，Alices需要在线并参与其中。只有Alices被鉴别后，Alices与Bob之间才可进行报文通信。当一份电子邮件从Alices发送到Bob时，需要报文鉴别，而当Alices从一部自动取款机提取现金时，需要实体鉴别。第二，报文鉴别只简单鉴别一个报文，对每一个新报文需要重复鉴别，而实体鉴别在一次会话期间证实申请者身份。

在实体鉴别中，验证者必须精确验证申请者的身份。鉴别实体依靠三种基本途径之一实现：所知的，所有的和固有的特征。

- 所知的　这是只有申请者所知的或所掌握的，用于验证者检验申请者所有的"物"，例如，口令、个人识别号（PIN）、密钥和私钥。
- 所有的　这是所具有的东西，可证明申请者身份的，例如，护照、驾驶证、身份证、信用卡和智能卡。
- 固有的特征　这是申请者的固有特征，例如，手写签名、指纹、语音、检型、视网膜图样和笔迹。

31.6.1 口令

口令（password）是实体鉴别最简单的与最古老的方法。它是申请者拥有的东西，当用户需要访问系统资源（登录）时，每个用户有公开的用户标识符和专用的口令。我们将这种鉴别方案分成二组：固定口令（fixed password）和一次性口令（one-time password）。

固定口令

在这个组中，口令是固定的。每次访问都用相同的口令，可是这种方会遭受到各种攻击。

- 窃听，Eve可注视Alices输入口令。多数系统不显示用户输入的口令字符作为安全措施，窃听有更加多样的形式。例如，Eve可使用在线窃听，拦截报文，由此捕获口令为她所用。
- 窃取口令。Eve直接偷取口令是第二种攻击。Alices不写下口令，而记住她自己的口令，这样可防止窃取口令。所以口令应是简单的，或是Alices易记住熟悉的对象，但它易受到其他类型的攻击。
- 访问文件。Eve像程序员那样操作系统，获取访问存储口令的文件。Eve读取该文件并搜索Alices口令，甚至篡改它。为了防止这类攻击，采用文件读/写保护。但是，多数系统需要这种类型的文件有公共的可读性。
- 猜测。Eve可登录系统，用字符的各种不同组合猜测Alices的口令。如果用户使短口令（少数几个字符），特别容易遭受攻击。如果口令选用某些平凡的记号，如她的生日、孩子的名字或她喜欢的明星，也易遭受攻击。为了防止猜测，推荐使用长的随机口令，这样的口令就不明显。但这样的随机口令也存在问题：Alices要在某处存储该口令，这样使她不会忘记它，这又会受到窃取口令攻击。

一个更安全的办法是将口令的散列存储在口令文件中（不用口令明文存储）。任何用户都可读取文件内容。但由于散列函数是单向的，几乎不可能猜测出口令值。即使Eve窃取口令文件，散列函数阻止她访问系统。但是，会遭受到另一种类型称为字典攻击（dictionary attack）的攻击。在这种攻击中，Eve不考虑用户ID而穷举搜索用户口令。例如，口令是6位数字，Eve可建立一张6位数字表（000000到999999），然后对每个数字应用散列函数得到一散列值，

形成一张一百万个散列值的表。她再用读取到的加密口令文件第二列的项与她自己形成的散列值表比较，看哪些密文重合。这可在Eve专用的计算机上脱机编程完成。找到匹配后，Eve联机使用口令访问系统。我们将会看到这种攻击在第三种方法中更加困难。

另一种方法称为掺杂（salting）口令。当创建口令时，一种称为掺杂的随机数与口令链接，然后对掺杂口令应用散列运算。用户ID、掺杂与散列存储在文件中。现在，当用户请求访问时，系统获取掺杂，并将它与接收到的口令链接，用该链接做出散列，然后与存储在文件中的散列比较。如果匹配，准予访问；否则拒绝。掺杂使字典攻击更加困难。如果原有的口令是6位数字，而掺杂是4位数字，那么散列是在10位数字上运算。这就是说，Eve要做出一张10兆项的表。做出比较长。如果掺杂是一个很长的随机数，掺杂口令是很有效的。UNIX操作系统使用这种方法的变种。

另一种方法是两种认证技术的组合。这种类型鉴别的一个很好的实例是ATM卡与PIN（个人识别号）结合使用，ATM卡属于“所有的”类，而PIN属于“所知的”类。PIN实际上是口令，它增强了卡的安全性。如果卡被偷窃，除非知道PIN，否则卡不能使用。但是习惯上PIN为自己易于记忆通常较短，这又易受到猜测攻击类型的攻击。

一次性口令

这个方案的口令只使用一次，称为**一次性口令**（one-time password）。一次性口令使窃听与窃取失效，但是这种方案很复杂，留在某些专著中讨论。

31.6.2 询问-应答

在口令鉴别中，申请者通过提供她的密码（口令）证明她的身份。但是，由于申请者出示这个密码，密码可能被入侵者截取。在询问-应答鉴别（challenge-response authentication）中，申请者可不泄露密码证明自己身份。换言之，申请者不用向验证者出示密码，验证者或者可得到它或者可找到它。

> 在询问-应答鉴别中，申请者不用泄露密码就可证明自己的身份。

询问是一个随时间变化的值，它是由验证者发送的一个随机数或时间戳。申请者对询问使用一个特定的函数，并向验证者发送其结果。这称为应答。应答表示了申请者已经知道密码。

> 询问是验证者发送的一个随时间变化的值。响应是对询问使用一个特定的函数。

使用对称密钥加密

第一类中，使用对称密钥加密实现询问-应答鉴别。在这一类中，申请者与验证者共享一个秘密密钥。函数是应用于询问的加密算法，图31.14表示了这种方法。图31.14中的第一个报文不属于询问-应答部分，它仅是通知验证者申请者想要询问。第二个报文是询问，R_B是验证者为询问申请者随机选取的一个瞬时值。申请者使用只有申请者和验证者知道的共享密钥加密瞬时值（nonce），并向验证者发送其结果。验证者解密该报文，如果解密得到的瞬时值与验证者发送的瞬时值相同，则同意Alices访问。

注意：在这个过程中，申请者与验证者必须保存保密所用的对称密钥。验证者必须保留用于鉴别申请者的瞬时值直到应答已经返回。

读者可能已注意到了使用瞬时值可以防止Eve重发第二个报文，Eve不可能重发第三个报文，伪装成这是一个新的鉴别请求，因为一旦Bob接收到应答，R_B的值就不再有效。下次要用新的值。

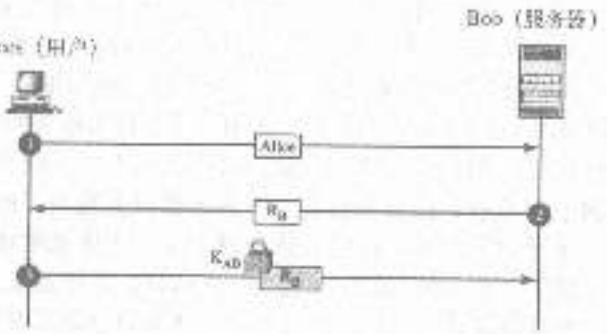


图31.14　使用时间值的询问-应答鉴别

这一类中的第二种方法，随时间变化的值是时间戳，因为时间戳显然随时间变化。在这种方法中，询问报文是验证者向申请者发送的当前时间。但是，这是假定客户机与服务器的时钟是同步的，申请者知道当前时间，这就是说，不需要询问报文。第一个报文与第三个报文可结合，其结果是用一个报文、对隐含询问的应答和当前时间完成鉴别。图31.15表示了这种方法。

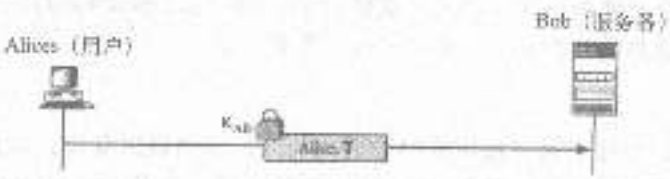


图31.15　使用时间戳的询问-应答鉴别

使用密钥散列函数

实体鉴别不用加密/解密方法，而是使用密钥散列函数（MAC）。这个方法有两个优点。第一，加密/解密算法不能进入某些国家。第二，使用密钥散列算法可保持询问与应答报文的完整性，并同时使用密钥。让我们观察使用密钥散列函数如何创建带有时间戳的询问报文，图31.16表示了该方案。

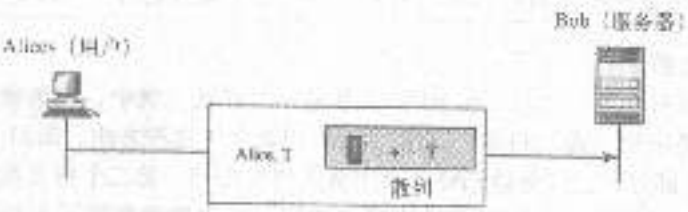


图31.16　使用密钥散列函数的询问-应答鉴别

注意在这种情况中，时间戳同时作为明文和用密钥散列函数杂凑的文本发送。当Bob接收到该报文时，他提取出明文T，对它使用密钥散列函数，然后将计算结果与接收到的散列比较来确定Alices的真实性。

使用非对称密钥加密

不用对称密钥加密，我们使用非对称密钥加密对实体进行鉴别。此时的密码必需是申请者的私钥，申请者要向每个人公开与她自己私钥有关的可用的公钥。这就是说，验证者用申请者的公钥加密询问，然后申请者用她的私钥解密。对询问的应答是解密的询问。我们说明

了两种方法：一种用于单向鉴别，另一种用于双向鉴别。在第一种方法中，Bob用Alices的公钥加密询问，Alices用她的私钥解密报文，并向Bob发送瞬时值。图31.17表示了这种方法。

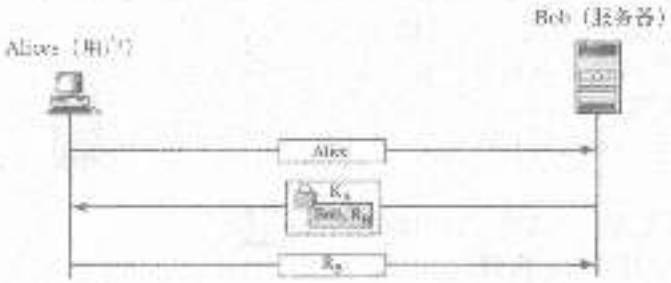


图31.17 使用非对称密钥鉴别

使用数字签名

我们可用数字签名进行实体鉴别。在这种方法中，我们让申请者使用她的私钥签名，而不是用私钥解密。图31.18表示了这种方法。Bob用明文询问，Alices签名应答。

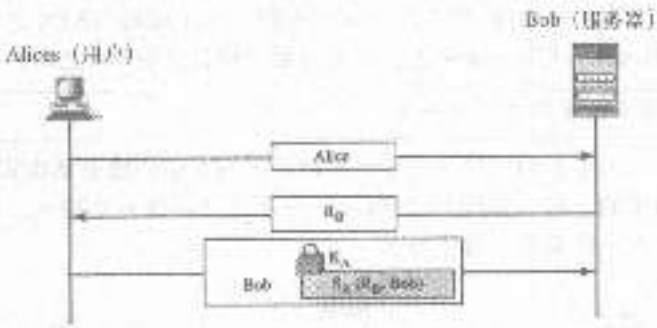


图31.18 用数字签名鉴别

31.7 密钥管理

我们已经在本章讨论了对称密钥密码学和非对称密钥密码学的应用，但未讨论对称密钥密码学中的秘密密钥和非对称密钥密码学中的公钥是如何分发和维护的，本节我们讨论这两个问题。首先讨论对称密钥的分发，然后讨论非对称密钥的分发。

31.7.1 对称密钥的分发

我们已知道当加密与解密大的报文时，对称密钥密码学比非对称密钥密码学更有效，但对称密钥密码学在双方之间需要一个共享的秘密密钥。

如果Alice需要与N个人交换保密报文，那么她需要N个不同的密钥。如果N个人需要彼此通信，要多少个密钥？总的密钥数共计$N(N-1)/2$。因为每个人需要与其他$N-1$个人通信，但由于密钥可共享，所以仅需要$N(N-1)/2$个密钥。这就是说，如果1百万人需要彼此通信，则每个人需要约50万个不同的密钥。因为N个实体要求的密钥数近似于N^2，这通常认为是N^2问题。

密钥的个数不是唯一存在的问题，还有密钥的分发问题。如果Alice与Bob想要通信，则他们需要以某种方法交换密钥。如果Alice要与1百万个人通信，她如何与1百万个人交换1百万个密钥？使用因特网没有一个明确的安全方法。

显然，我们需要一个维护与分发密钥的有效方法。

密钥分发中心：KDC

一个实际的解决方案是使用一个称为密钥分发中心（key distribution center，KDC）的可信任方。为了减少密钥的个数，每个人建立一个带有KDC的共享密钥，如图31.19所示。

在KDC与每一个成员之间建立一个私钥。Alice与KDC有一个私钥，我们称为K_{Alice}。Bob与KDC有一个私钥，称为K_{Bob}等等。现在的问题Alice如何向Bob发送保密的报文？其过程如下：

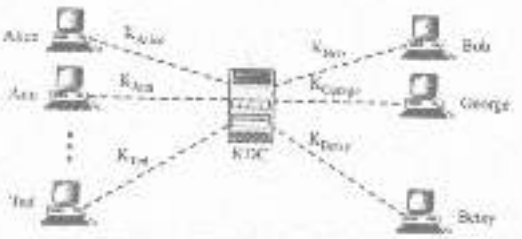


图31.19　KDC

1. Alice向KDC发送一个请求，表明她需要和Bob之间的会话（暂时的）密钥；
2. KDC向Bob通告Alice的请求；
3. 如果Bob同意，则在两者之间创建一个会话密钥。

用KDC在Alice与Bob之间创建的密钥用于KDC鉴别Alice和Bob，并防止Eve假装他们中的任一个。在本章的后面，我们将讨论在Alice和Bob之间创建一个会话密钥。

会话密钥

KDC为每个成员创建一个秘密密钥，该秘密密钥仅用于成员与KDC之间，而不是用在两个成员之间。如果Alice需要与Bob保密通信，她需要在她自己与Bob之间的一个秘密密钥。

> 双方之间的对称会话密钥仅使用一次。

使用前面讨论过的实体鉴别思想可以提出多种不同的方法创建会话密钥。

让我们讨论最简单的一种，如图31.20所示。尽管这个系统有些缺点，但它表明了思想实质。更复杂的方法可在一些安全书籍中找到。

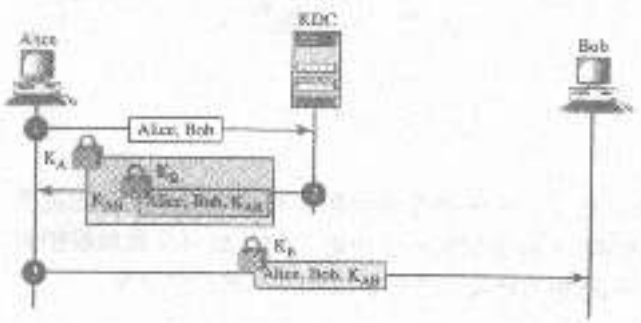


图31.20　使用KDC在Alice与Bob之间创建会话密钥

- 步骤1，Alice向KDC发送明文报文以获得她和Bob之间的对称会话密钥。报文包含了她的注册身份（图中的Alice）和Bob的身份（图中的Bob）。该报文没有加密，是公开的。KDC并不关心这些。
- 步骤2，KDC接收到报文，生成了所谓的票证（ticket）。该票证是用Bob的密钥（K_B）加密的。票证包含了Alice和Bob的身份，以及会话密钥（K_{AB}）。带有会话密钥副本的票证发送给Alice。Alice接收到该报文，对它进行解密，得到会话密钥。她无法解密Bob的票证，因为票证是给Bob的，不是给Alice的。注意：在该报文中采用了两次加密。票证是加密的，整个报文也是加密的。在第一个报文中，Alice被KDC鉴别，因为只有Alice能用她的带有KDC的密钥打开整个报文。

- 步骤3。Alice将票证发送给Bob。Bob打开票证，知道Alice需要使用K_{AB}作为会话密钥来向他发送报文。注意：在这个报文中，Bob被KDC鉴别，因为只有Bob能打开票证，因为Bob被KDC鉴别，他也被信任KDC的Alice鉴别。同样的方法，Alice也能被Bob鉴别，因为Bob信任KDC。KDC向Bob发送票证，该票证包括了Alice的身份。

Kerberos

Kerberos是一个鉴别协议，同时又是一个KDC，它已经变得十分流行。包括Windows 2000在内的几个操作系统都使用Kerberos。Kerberos是根据希腊神话中为冥王哈得斯守门的三头狗命名的。这个协议最初麻省理工学院设计出来的，目前已经经历了好几个版本，这里只讨论最常用的第4版。

服务器 Kerberos协议中包括二个服务器：一台鉴别服务器（AS），一台票证授予服务器（TGS）和一台用来向他人提供服务的实际的（数据）服务器。在我们所给出的一些实例及其插图中，Bob是实际的服务器，Alice是一个请求服务的用户。图31.21显示了这三台服务器之间的关系。

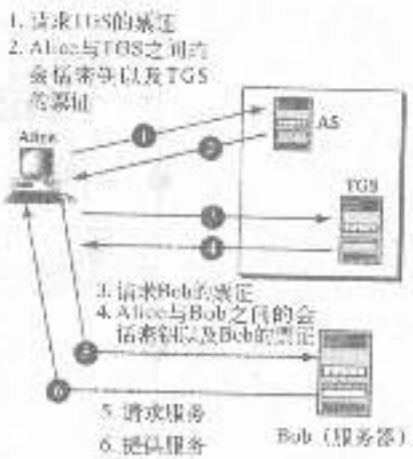


图31.21 Kerberos服务器

- 鉴别服务器（AS）是Kerberos协议中的KDC。每个用户在AS中进行注册，并被分配一个用户身份和口令。AS拥有一个包含这些用户身份和相应口令的数据库。AS验证用户，发布Alice和TGS之间使用的会话密钥，并发送一张给TGS使用的票证。
- 票证授予服务器（TGS）为实际服务器（Bob）发布票证。它还提供Alice和Bob之间的会话密钥（K_{AB}）。Kerberos将用户验证与票证发布分离开来。通过这种方式，尽管Alice只与AS验证了一次她的ID，但是她可以与TGS联系多次以获得不同的实际服务器的票证。
- 实际服务器（Bob）为用户（Alice）提供服务。Kerberos设计成一个诸如FTP的客户/服务器的程序，用户可以采用客户端进程访问服务器进程。Kerberos不是用来进行人对人的鉴别的。

操作 客户端进程（Alice）可以通过六个步骤，访问运行在实际服务器上的进程，如图31.22所示。

- 步骤1。Alice用明文向AS发送她的请求，使用的是她注册的身份。
- 步骤2。AS发送一份用Alice的对称密钥K_A加密过的报文。该报文包括两项：一份Alice用来与TGS联系时所使用的会话密钥K_S和一张TGS的票证，该票证是用TGS的对称密钥K_{TG}加密的。Alice不知道K_A，但是当报文到达时，她输入她的口令。如果口令正确，口令和相应的算法一起会生成K_A，口令立即被销毁。它不发送给网络，也不驻留在终端，它只是暂时用来生成K_A。现在，进程使用K_A来解密发送的报文，从而得到K_S和票证。
- 步骤3。现在，Alice向TGS发送三项内容。第一项是从AS接收的票证。第二项是实际服务器（Bob）的名称，第三项是采用K_S加密的时间戳，时间戳用来制止Eve的重放攻击。
- 步骤4，现在，TGS发送两张票证，每张都包括Alice和Bob之间的会话密钥K_{AB}。给Alice的票证采用K_A加密，给Bob的票证采用Bob的密钥K_B进行加密，注意：Eve无法得到K_{AB}，因为她不知道K_A或K_B。她不能重放步骤3，因为她无法用新的时间戳（她不知道K_S）来

替换真正的时间戳。即使她很快，即在时间戳超时之前发送了步骤3的报文，她得到的仍然是相同的她无法解密的两张票证。

- 步骤5，Alice发送Bob的带有用K_{AB}加密的时间戳的票证。
- 步骤6，Bob通过将时间戳加1来确认接收。报文用K_{AB}加密后发送给Alice。

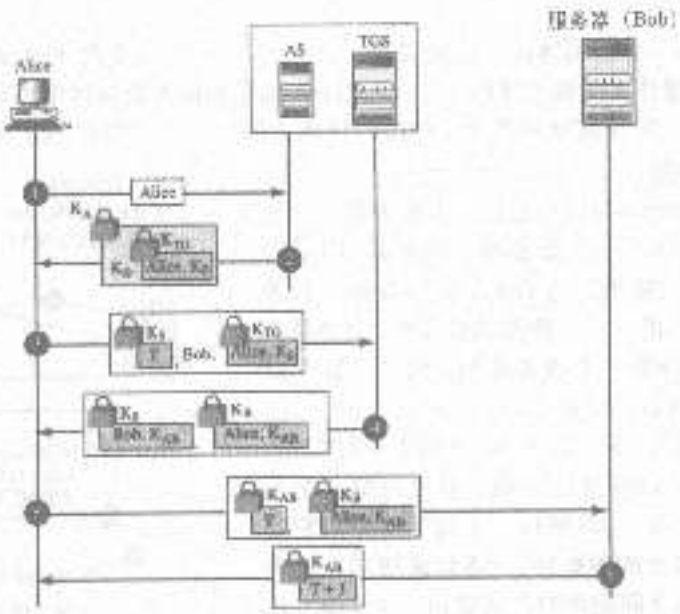


图31.22 Kerberos实例

使用不同的服务器 注意：如果Alice需要从不同的服务器接收服务，那么她只需要重复最后的四步（步骤3～步骤6）。最初的两步已经验证了Alice的身份，不需要重复。Alice通过重复步骤3～步骤6，就可以让TGS发布多个服务器的票证。

领域 Kerberos允许AS和TGS的全球发布，每个系统称为一个领域（realm）。用户可以获得本地服务器或远程服务器的票证。例如，在第二种情况下，Alice可能要求她的本地TGS发行一张由远程TGS接收的票证。如果该远程TGS在本地TGS中进行了注册，那么本地TGS可以发布这张票证，Alice从而可以使用该远程TGS来访问远程的实际服务器。

31.7.2 公钥分发

在非对称密钥密码学中，人们不需要知道对称共享密钥。如果Alice想给Bob发送报文，她只需要知道Bob的公钥，该公钥是公开的，任何人都能得到。如果Bob想给Alice发送报文，他只需要知道Alice的公钥，她的公钥也是任何人都知道的。在公钥加密中，每个人都隐藏私钥，并将公钥公开发布。

> 在公钥密码学中，任何人都可以访问其他人的公钥，公钥是公开发布的。

公钥像秘密密钥一样为应用需要而分发，让我们简要讨论公钥分发的方法。

公钥发布

最简单的方法是公开发布公钥。Bob可把他的公钥放在他的Web网站或在本地的与全国的报纸上公布。当Alice向Bob发送保密报文时，她从Bob的网站或报纸上得到他的公钥，甚至她可以发报文询问。图31.23表示这种情形。

但是，这种方法是不安全的，会遭受到假冒。例如，Eve可将她自己的公钥作为Bob公钥的一个公开的发布。在Bob做出反应前，破坏可能发生。Eve可混淆Alices发给Bob的报文，Eve也可用一个相应的私钥签名一个文档，使每个人相信文档是Bob签名，如果Alices直接向Bob询问公钥，也易受到攻击，Eve可截取Bob的应答，用她的公钥取代Bob的公钥。

可信任中心

一个更安全方法要有一个可信任中心保存公钥的目录，这个目录像电话簿一样是动态更新的。每个用户可选取一个私钥/公钥对，保留私钥并将公钥插入目录。可信任中心要求每个用户在中心注册并检验他或她的身份。目录可由可信任中心公开发布，中心也应答有关公钥的查询。图31.24表示了这个概念。



图31.23 发布公钥

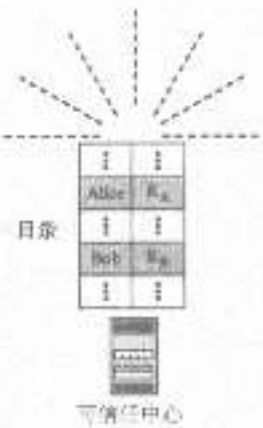


图31.24 可信任中心

受控制的可信任中心

如果对公钥的分发增加一些控制，那么可以得到更高级的安全。公钥发布可包括时间戳和权威机构的签名从而可防止应答的截取与修改。如果Alices需要知道Bob的公钥，她可以向中心发送一个包括Bob名和时间戳的一个请求。中心用Bob公钥应答原有的请求而时间戳用中心的私钥签名。Alices使用大家都知道的中心公钥解密报文并获得Bob公钥。图31.25表示了这种情况。

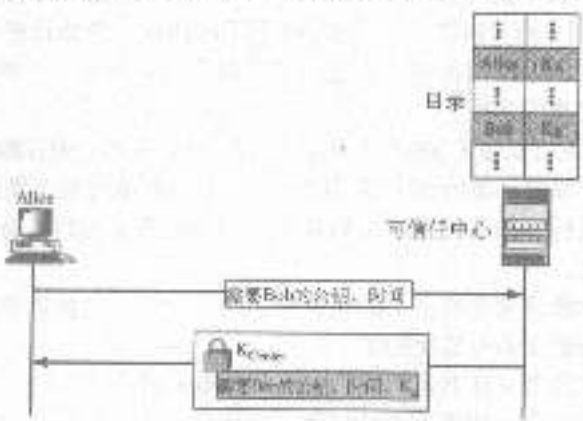


图31.25 受控制的可信任中心

认证中心

如果请求的个数很多，前述的方法在中心形成沉重的负担，另一种选择创建公钥的认证。

Bob需要做两件事情：他需要人们知道他的公钥，他还需要确保任何人无法获得以Bob名义伪造的公钥。Bob可以去认证中心（certification authority，CA），认证中心是一个联邦或州级组织，用来将公钥与实体绑定，并且发布一份证书。该CA本身有一个众所周知的无法伪造的公钥。CA核实Bob的身份（identication）（采用驾照ID和其他证据），然后向Bob索要公钥并将其写在证书上。为防止证书被伪造，CA根据证书生成一份摘要并用它的私钥对摘要进行加密。现在Bob可以上传明文形式的证书和加密后的报文摘要。任何人只要想得到Bob的公钥，都可以下载证书和加密后的摘要。然后根据证书生成摘要，并使用CA的公钥对加密的摘要进行解密。对两份摘要进行比较，如果它们相等，就说明证书是有效的，没有冒充者假装Bob。图31.26表示这个概念。

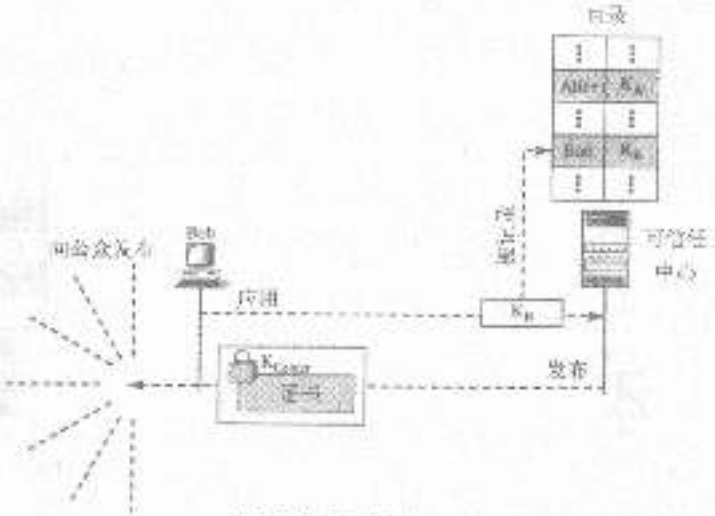


图31.26　认证中心

X.509　尽管采用CA解决了伪造公钥的问题，但是它出现了副作用。每份证书可能有不同的格式。如果Alice想使用程序自动下载属于不同人的不同的证书和摘要，程序可能无法做到这一点，一份证书中的公钥可能是一种格式，另一份可能又是另一种格式。公钥可能在一份证书中位于第一行，在另一份中就有可能位于第三行。任何要普遍使用的事物都必须有统一的格式。

要消除这种副作用，ITU制定了一个称为X.509的协议，该协议经过一些修改已经被因特网所采纳。X.509协议采用结构化方式描述证书的规范。它使用一个熟知的称为ASN.1的协议来定义C程序员非常熟悉的字段。

- 版本。这个字段定义认证X.509的版本号。版本号从0开始，现在的版本号是2（第3版本）。
- 系列号。这个字段定义赋给每个证书的一个号码，其值对每个发出的证书是唯一的。
- 签名。这个字段标识签名证书所用的算法，名字是不适当的。为这个签名所需的参数也在这个字段定义。
- 签发者。这个字段定义发布证书的认证中心的名称。该名称通常是一个由国家、州、组织机构和部门等定义的有层次的串。
- 有效期。这个字段定义证书有效期的起始和结束时间。
- 主体。这个字段定义公钥所有者的实体。它也是一个层次串。字段部分定义了称为公用名的字段。它是密钥的观看者真实名。
- 主体的公钥。这个字段定义主体的公钥。证书的核心。这个字段也定义主体的算法（例如，RSA）和它的参数。

- 签发者唯一的标识符。这是一个可选字段。它允许两个发布者具有相同的发布者字段的值，如果发布者唯一的标识符是不同的。
- 主体唯一的标识符。这个可选字段允许两个主体具有相同的主体字段的值，如果主体唯一的标识符是不同的。
- 扩展项。这个字段允许发布者为证书增加更多的私有信息。
- 加密。这个字段包含算法标识符、其他字段的安全散列和该散列的数字签名。

公钥基础设施（PKI）

当人们想统一地使用公钥时，会遇到一个类似于密钥分发的问题。人们发现不能仅有一个KDC来应答请求，而是需要很多服务器。另外，人们发现最佳方案是将服务器配置为层次关系。类似地，公钥查询的一种解决方案是采用称为**公钥基础设施**（public key infrastructure，PKI）的层次化结构。图31.27表示了这种层次结构的例子。

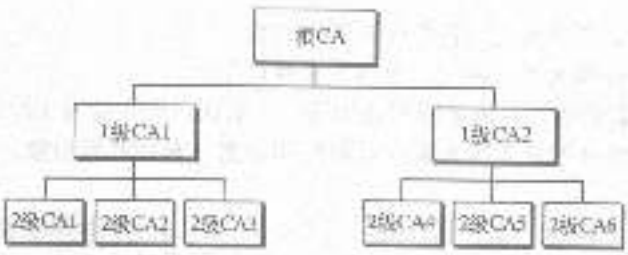


图31.27 PKI层次结构

在第一级中，拥有一个根CA来验证第二级CA。这些1级的CA可以在一个很大的地理区域或逻辑区域的范围内运作。2级的CA可以在一个相对较小的地理区域内运作。

在这个层次结构中，每个人都信任根节点。但人们可以信任也可以不信任中间的CA。如果Alice需要得到Bob的证书，她可以找发布该证书的某个地方的CA，但Alice不一定信任该CA。在层次结构中，Alice可以通过上一级CA来验证原始的CA。该查询可以一直进行到根CA。

推荐读物

为了深入讨论与本章有关的主题，推荐阅读下列读物。在本书末列出方括号[…]中的参考资料。

读物

有关网络安全的专著如[PHS02]、[Bis03]和[Sal03]。

本章小结

- 密码技术提供五种服务，其中四种是关于Alices与Bob之间的报文交换，第五种是关于试图访问系统并使用系统资源的实体的。
- 报文的保密性是指发送方与接收方期待保密。
- 报文的完整性是指数据必须正确地到达接收方。
- 报文的鉴别是指接收方确信报文来自假定的发送方而不是冒名顶替者。
- 不可否认性是指发送方不能否认已发送的数据。
- 实体鉴别是指证明试图访问系统资源的实体的身份。
- 报文摘要用于保护文档或报文的完整性，散列函数从报文创建报文摘要。

- 散列函数必须满足三个条件：单向性、抗弱碰撞性与抗强碰撞性。
- 无密钥报文摘要用作修改检测码（MDC），它保证报文完整性。要鉴别数据源，需要报文鉴别码（MAC）。
- MAC是密钥散列函数，它用报文和密钥创建压缩的摘要。其方法与加密算法的基本原理相同。
- 数字签名方案提供了与手书签名相同的服务。手书签名包含在文档中，而数字签名是独立的实体。
- 数字签名提供报文完整性、鉴别和不可否认性。但数字签名不能提供报文保密性。如果需要保密性，必须在方案上应用加密系统。
- 数字签名需要一个非对称密钥系统。
- 在实体鉴别中，申请者用三种基本途径：所知的物、所有的物和固有特征向验证者证明自己的身份。
- 在口令鉴别中，申请者用字符串作为所知的物。
- 口令鉴别可划分两大类：固定口令与一次性口令。
- 在询问-应答鉴别中，申请者证明她知道一个密钥，但并没有实际发送它。
- 询问-应答鉴别可划分为四大类：对称密钥加密、密钥散列函数、非对称密钥加密和数字签名。
- 密钥分发中心（KDC）是可信任的第三方，它向双方分配对称密钥。
- KDC仅在成员与中心之间创建密钥。当两个成员联系KDC时，成员之间的密钥需要作为会话密钥创建。
- Kerberos是一个流行的会话密钥创建协议。它需要一个鉴别服务器和一个票证服务器。
- 公钥基础设施（PKI）是一个应答有关密钥认证询问的层次结构的系统。

习题

复习题

1. 瞬时值是什么？
2. N^2问题是什么？
3. 列出一个使用KDC作为用户鉴别协议的名称？
4. Kerberos鉴别服务器的目的是什么？
5. Kerberos票证服务器的目的是什么？
6. X.509目的是什么？
7. 什么是认证中心？
8. 使用长的口令有哪些优点与缺点？
9. 讨论作为二种极端的固定和一次性的口令，口令改变频率是多少？你认为这个方案如何实现？优点与缺点是什么？
10. 一个系统如何防止口令猜测攻击？如果已发现银行卡被偷窃并试图使用它，银行能否防止PIN猜测？

练习题

11. 一个报文由00到99之中的10个数字组成，一个散列算法是这些数相加的和模100运算从这个报文中创建一个摘要，产生的摘要是00到99中的一个数。这个算法是否满足散列算法第一个条件？是否满足第二个条件？是否满足第三个条件？

12. 一个报文是由100个字符组成。散列算法是挑选1，11，21，……到91字符创建该报文的摘要。其结果的摘要有10个字符。这个算法是否满足散列算法第一个条件？是否满足第二个条件？是否满足第三个条件？
13. 一个散列算法创建N位摘要，这个算法能够有多少个不同的摘要？
14. 在一方，一个人的生日是特定的一天或两个（多个）人有相同生日，哪一个可能性更大？
15. 如何解决习题14有关散列函数第二个与第三个条件？
16. 固定长度的摘要和可变长度的摘要，哪一个更切实可行？试说明其理由。
17. 一个报文是20 000个字符，使用SHA-1创建这个报文的摘要，然后决定改变为最后10个字符，是否可说摘要中有多少位将被改变？
18. 创建一个MAC过程与签名一个散列是否相同？
19. 某人使用自动出纳机提取现金，这是报文鉴别还是实体鉴别，还是两者兼有？
20. 将图31.14改为提供双向鉴别（Alices鉴别Bob与Bob鉴别Alices）。
21. 将图31.16改为提供双向鉴别（Alices鉴别Bob与Bob鉴别Alices）。
22. 将图31.17改为提供双向鉴别（Alices鉴别Bob与Bob鉴别Alices）。
23. 将图31.18改为提供双向鉴别（Alices鉴别Bob与Bob鉴别Alices）。
24. 在大学中，学生在登录发送口令前需要加密口令，这加密是保护学校还是保护学生，试说明理由。
25. 在习题24中，如果在加密之前，学生附加时间戳，这有帮助吗？试说明理由。
26. 在习题24中，如果学生有一个口令表并每次使用一个不同口令，这有帮助吗？试说明理由。
27. 在图31.20中，如果KDC发生故障，将会发生什么？
28. 在图31.21中，如果AS发生故障，将会发生什么？如果TGS发生故障，将会发生什么？如果主服务器发生故障，将会发生什么？
29. 在图31.26中，如果可信任中心发生故障，将会发生什么？
30. 为了提供保密性，试在图31.11中增加对称密钥加密/解密。
31. 为了提供保密性，试在图31.11中增加非对称密钥加密/解密。

探究题

32. 有一个MD5散列算法，找出这个算法与SHA-1的不同处。
33. 有一个RIPEMD-160散列算法，找出这个算法与SHA-1的不同处。
34. 比较MD5，SHA-1和RIPEMD-160。
35. 讨论有关RSA数字签名一些信息。
36. 讨论有关DSS数字签名一些信息。

第32章 因特网中的安全措施

在本章中，我们要说明某些安全问题，特别是保密性和报文鉴别是如何应用于因特网模型的网络层、传输层和应用层的。我们简要说明了IPSec协议如何在IP协议中增加鉴别和安全，和SSL（或TLS）如何针对TCP协议完成相同的任务以及PGP如何针对SMTP协议（电子邮件）完成相同的任务。

在所有这些协议中，我们需要考虑一些共同的问题。首先，需要生成一个MAC；其次要加密报文，或许要加密MAC。这就是说，尽管有某些细小的差异，但本章所讨论的三个协议都会从适当的层得到一个分组并创建一个加密的新分组。如图32.1所示。



图32.1 三个安全协议的共同结构

注意，在加密过程中，可以包括也可以不包括安全协议的头部和尾部。还应注意有些协议还需要在被加密的分组中有更多的信息。图32.1仅表示了一般的思想。

在所有这些协议中还有一个共同的问题是安全参数。即使在图32.1的简化结构中也间接地表明Alice与Bob彼此在发送加密的分组之前，需要知道有关一些安全参数的信息。他们尤其需要知道用于鉴别和加密/解密分组的算法，即使这些算法为全球用户预先设计的，正如我们所见，他们是不知道的。Bob与Alice至少还需要两个密钥：一个是MAC的而另一个是加密/解密的。换句话说，这些协议的复杂性不存在于计算MAC数据和执行加密的方法。事实上，它存在于计算MAC数据和实施加密之前，我们需要在Alice与Bob之间创建一组安全参数。

先粗略看一下，这些协议中的任一个都好像必须有无限多的步骤，因为要发送安全的数据，必须要用一组安全参数，而安全参数的安全交换又需要另一个第二组安全参数，而第二组安全参数的交换又需要第三组安全参数，等等，直至无穷。

为了减少步骤，我们可使用公开密钥加密方法。如果每一方都有一对私钥和公钥，则步骤的数可减到一步或二步。在一步情况下，我们可用会话密钥生成该MAC并加密数据和MAC。会话密钥和算法列表可用分组发送，但用公开密钥加密算法加密。在二步的情况下，首先用公开密钥加密算法建立安全参数。然后用安全参数去安全地发送真正的数据。这三个协议中，PGP使用第一种方法，而其余两个IPSec和SSL/TLS使用第二种方法。

32.1 IP层的安全：IPSec

IP层安全（IP Security，IPSec）是由因特网工程任务组（IETF）设计的用来为IP层的分组提供安全的一组协议。IPSec帮助生成经过鉴别的与安全的IP层的分组，如图32.2所示。

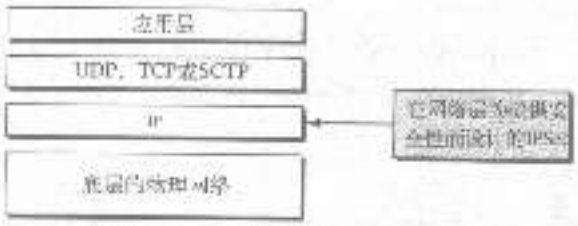
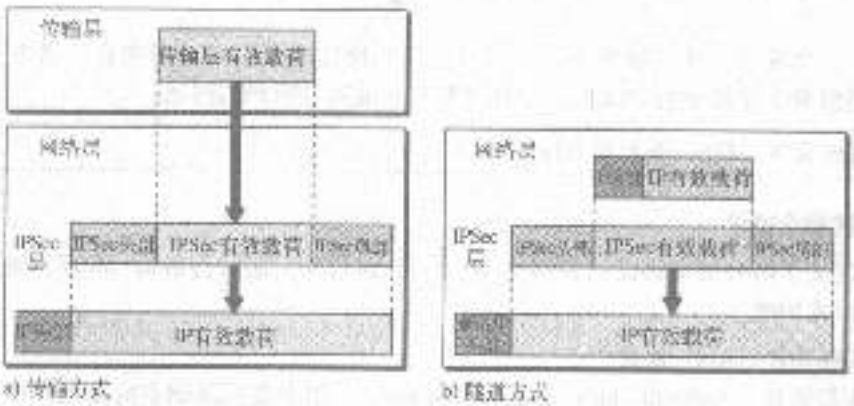


图32.2 TCP/IP协议族与IPSec

32.1.1 两种方式

IPSec以两种不同的方式运行：传输方式和隧道方式，如图32.3所示。



a) 传输方式　　b) 隧道方式

图32.3 IPSec协议的传输方式和隧道方式

传输方式

在**传输方式**（transport mode）下，IPSec保护传输层到网络层传递的内容。换言之，传输方式保护网络层的有效载荷，在网络层中封装有效载荷。

注意：传输方式不保护IP头部。换言之，传输方式不保护整个IP分组，而仅保护来自传输层的分组（IP层有效载荷）。在这种方式下，对来自传输层的信息增加IPSec的头部与尾部，然后再增加IP头部。

> 在传输方式下，IPSec不保护IP头部，仅保护来自传输层的信息。

传输方式通常用于主机到主机（端到端）的数据保护。发送主机使用IPSec鉴别和/或加密来自传输层的有效载荷，接收主机使用IPSec检验鉴别和/或解密IP分组并将它传送给传输层。图32.4说明了这个概念。

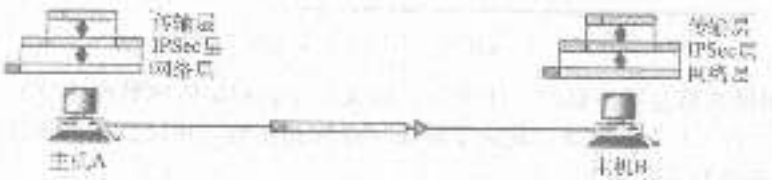


图32.4 传输方式

隧道方式

在**隧道方式**（tunnel mode）下，IPSec保护整个IP分组，包括头部。将IPSec加密方法用

于整个分组，然后增加一个新的头部，如图32.5所示。

不久将会看到，这个新的IP头部与原来的IP头部有一些不同的信息。隧道方式通常用于两个路由器之间，或一个主机与一个路由器之间以及一个路由器与一个主机之间。如图32.5所示。

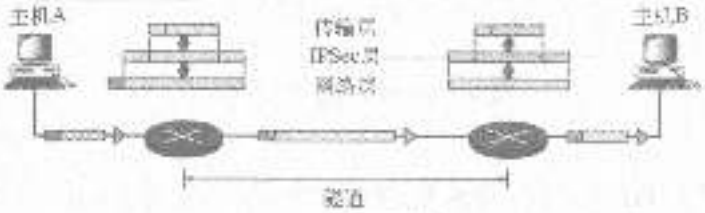


图32.5　隧道方式

换言之，当发送方或接收方不是主机时，我们使用隧道方式。对来自发送方与接收方之间的侵入提供整个原始分组的保护，它好像整个分组通过假想的隧道。

在隧道方式下，IPSec保护原始IP头部。

32.1.2　两种安全协议

IPSec定义了两种安全协议：鉴别头部（AH）协议和封装安全载荷（ESP）协议。在IP层提供鉴别和/或加密。

鉴别头部协议（AH）协议

鉴别头部协议（Authentication Header Protocol）用于鉴别源端主机以确保IP分组所携带载荷的完整性。协议使用散列函数和一个对称密钥生成报文摘要，并将摘要插入鉴别头部中。根据其运行方式（传输方式或隧道方式），将AH插入到相应的位置。图32.6表示了在传输方式下各个字段以及鉴别头部的位置。



图32.6　传输方式中的鉴别头部（AH）协议

当由IP数据报携带鉴别头部时，IP头部中协议字段的初始值就替换为值51。在鉴别头内部的一个字段（下一个头部字段）规定了协议字段的初始值（由IP数据报携带的有效载荷类型）。加入鉴别头部的步骤如下：

1. 鉴别头部加入到有效载荷中，将鉴别数据字段的值置为零。
2. 有可能加入填充项以使总长度适合于特定的散列算法。
3. 散列处理是根据总的分组计算的。但是，IP头部中只有在传输过程中不发生改变的那

些字段才能包含在报文摘要（鉴别数据）的计算过程中。

4. 将鉴别数据加入到鉴别头部中。

5. 协议字段的值改为51后，将IP头部加入。

各个字段简要描述如下：

- 下一个头部。这个8位字段定义了IP数据报所携带的有效载荷类型（TCP、UDP、ICMP、OSPF等）。它与封装前的IP头部的协议字段具有相同的功能。换言之，进程将IP数据报中的协议字段的值更改为51，以表明该分组携带有鉴别头部。
- 有效载荷长度。这个8位字段的名称容易使人误解。它定义的并不是有效载荷的长度，而是定义了以4字节的倍数计算的鉴别头部的长度，但它不包括最前面的8个字节。
- 安全参数索引（SPI）。32位的安全参数索引（SPI）字段的作用类似于虚电路的标志符。对于在安全关联（后面讨论）的连接期间发送的所有分组，SPI都是相同的。
- 序列号。32位的序列号为数据报提供排序信息。序列号不允许重复。注意，即使分组是重传的，序列号也不重复。当达到2^{32}时，序列号不再循环计数，这时必须建立新的连接。
- 鉴别数据。最后一项，鉴别数据字段是采用散列函数对整个IP数据报进行计算的结果，但是计算中不包括在传输（比如生存时间）过程中发生改变的字段。

> AH协议提供源端鉴别和数据完整性，但不提供保密性。

封装安全载荷

AH协议并不提供保密性。只提供源端鉴别和数据完整性，IPSec后来定义了一个可选的称为封装安全载荷（Encapsulating Security Payload，ESP）的协议，它提供源端鉴别、完整性以及保密性。ESP增加了一个头部和一个尾部。注意，ESP的鉴别数据加在分组的末端，这样使计算更容易。图32.7表示了ESP的头部和尾部的位置。

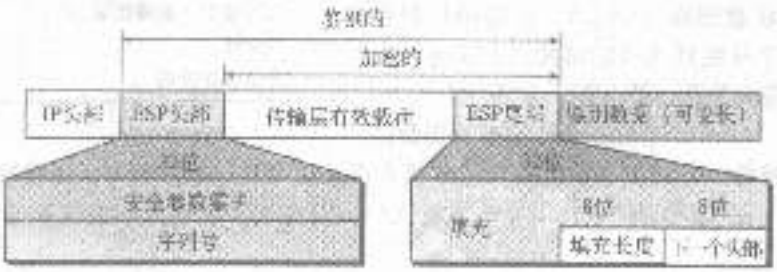


图32.7 传输方式中的封装安全载荷（ESP）协议

当IP数据报携带ESP的头部和尾部时，IP头部中协议字段的值更改为50。在ESP尾部内有一个字段（下一个头部字段）保留着协议字段的初始值（IP数据报所携带的载荷类型，比如TCP或UDP）。ESP过程遵循下列步骤：

1. 将ESP尾部加入到有效载荷中。
2. 对有效载荷和尾部进行加密。
3. 加入ESP头部。
4. 使用ESP头部、有效载荷以及ESP尾部来生成鉴别数据。
5. 将鉴别数据加入到ESP尾部的末端。
6. 将协议值改为50后，再加入IP头部。

头部和尾部的字段如下：

- 安全参数索引。这个32位的安全参数索引字段的规定与AH协议类似。
- 序列号。这个32位的序号字段的规定与AH协议类似。
- 填充。这个可变长（0～255字节）的全0字段作为填充使用。
- 填充长度。这个8位的填充长度字段定义了填充的字节数。取值范围为0～255，最大值很少出现。
- 下一个头部。这个8位的下一个头部字段的定义与AH协议类似。它与封装前IP头部中的协议字段有相同的目的。
- 鉴别数据。最后，鉴别数据字段是对数据报部分中使用鉴别方案进行计算所得出的结果。注意：AH和ESP中鉴别数据的区别。在AH中，IP头部的一部分包含在鉴别数据的计算中，而在ESP中，不包含IP头部数据。

ESP提供源端鉴别和数据完整性以及保密性。

IPv4和IPv6

IPSec同时支持IPv4和IPv6。但是在IPv6中，AH和ESP是作为扩展头部的一部分。

AH与ESP

ESP协议是在AH协议已经使用后设计的。ESP除了能做AH能做的任何事情外，还能提供额外的功能（保密性）。问题是，为什么我们需要AH？答案是我们并不需要。但是，AH的实施已经包含在一些商业产品中，这就是说AH将仍然是因特网的一部分，直至这些产品被淘汰。

IPSec提供的服务

AH和ESP两个协议可在网络层对分组提供几种安全服务。表32.1显示了每一个协议可用的服务列表。

表32.1　IPSec服务

服　务	AH	ESP
访问控制	有	有
报文鉴别（报文完整性）	有	有
实体鉴别（数据源鉴别）	有	有
保密性	无	有
避免重放攻击	有	有

访问控制　在下一节我们将会看到IPSec通过使用安全关联数据库（SADB）间接地提供访问控制。当一个分组到达目的端时，如果还没有为该分组建立安全关联，那么就丢弃该分组。

报文鉴别　AH和ESP两者通过使用鉴别数据提供报文的完整性。生成一个数据摘要，该摘要由发送方发送给接收方，由接收方进行检验。

实体鉴别　在AH和ESP中，用发送方发送的安全关联和密钥散列数据摘要鉴别数据的发送方。

保密性　在ESP中报文加密提供保密性，但是AH没有提供保密性。如果需要保密性，我们应使用ESP而不是AH。

避免重放攻击　在两个协议中，用序列号和一个滑动的接收窗口防止重放攻击（replay attack）。当建立安全关联时，每个IPSec头部都有一个唯一的序列号。序列号从0开始增加直到$2^{32}-1$（序列号字段的大小是32位）。当序列号到达最大值时，它复位为0。同时删除旧的安全关联（见下一节）并建立新的安全关联。为了防止处理重复的分组，IPSec要求接收方使用固定大小的窗口，接收方固定窗口大小的默认值是64。

32.1.3　安全关联

在本章引言中提到，我们讨论三个协议（IPSec、SSL/TLS和PGP）中的每一个在它运行前必须要一组安全参数。IPSec需要使用一个称为安全关联（Security Association，SA）来创建一组安全参数。

因为我们已知道IP是一个无连接协议，每个数据报彼此独立，对这种类型的通信，可用下列三种方法之一建立安全关联：

1. 每个数据报的安全参数可包含在该数据报中，可是IPSec的设计者没有可能选用这种方法，因为在每个数据报中都增加安全参数会引起大量的开销，特别是当数据报在传输过程中多次被分段时；

2. 对每个数据报建立一组安全参数。这就是说，为了建立安全参数，在传输每个数据报之前发送方与接收方之间需要交换一组分组。这种方法可能比第一种方法效率更低，IPSec设计者也不会使用；

3. IPSec设计者使用第三种方法。当发送方第一次有一个数据报要发送给某一特定的接收方时，在发送方与该特定的接收方之间建立一组安全参数。该组安全参数可保存用作以后向同一个接收方发送IP分组传输。

安全关联是IPSec一个很重要的方面。IPSec用安全关联将IP无连接协议改进为面向连接的协议。我们可以将关联认为是连续。我们可以说，当Alice与Bob一致同意他们之间用一组安全参数，则他们之间已建立了一个逻辑连接（称为关联）。但是，他们可以不是全部时间使用这个连接。连接建立后，Alice可以今天发送一个数据报，过几天后再发送另一个数据报等等，逻辑连接已存在，可以发送一个安全的数据报了。当然，他们可以中断连接，或者过一会建立一个新的连接（那是更安全通信方法）。

一个简单的例子

一个安全关联是一个很复杂的信息组，然而我们用一个最简单情况说明。Alice想要利用关联与Bob进行双向通信，Alice要有一个出关联（用于到Bob的数据报）和一个入关联（用于来自Bob的数据报），同时Bob也要有相同的关联。在这种情况下，安全关联对于Alice和Bob来说简化为两个小表，如图32.8所示。

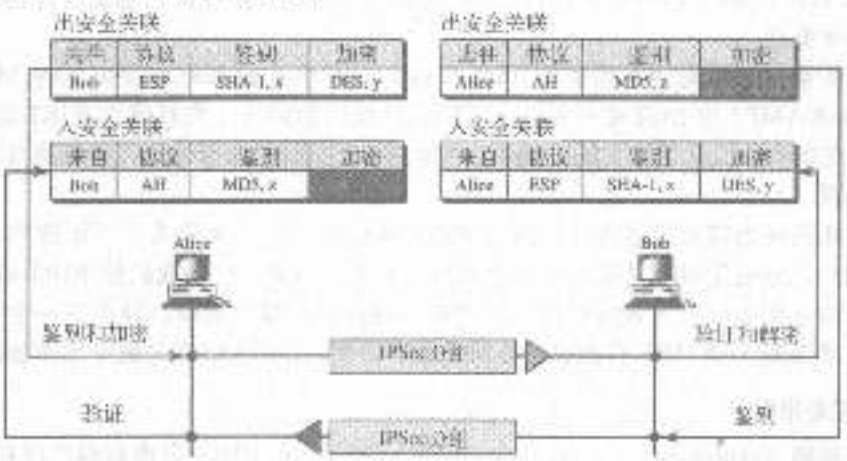


图32.8 简单的入和出安全关联

图32.8表示了当Alice需要向Bob发送数据报时，她使用IPSec的ESP协议，用密钥为x的SHA-1执行鉴别，而用密钥为y的DES执行解密。当Bob需要向Alice发送数据报时，他使用IPSec的AH协议，用密钥为z的MD5执行鉴别。注意：Bob的入关联与Alice的出关联是相同的，其逆亦是。

安全关联数据库（SADB）

安全关联数据库可能是很复杂的。如果Alice要发送多份报文给多个人，Bob需要接收来

自多个人的报文时，情况就相当复杂。另外，每一方都需要有入关联和出关联才能允许双向通信。换言之，我们要有一组关联，它被聚合成为一个数据库。这个数据库称为**安全关联数据库**（security association database，SADB），该库可认为是二维的表，其中每行定义一个关联。通常有两个SADB：一个入SADB，另一个出SADB。

安全参数索引

为了区别一个关联与另一个关联，用一个称为**安全参数索引**（security parameter index，SPI）标识的参数标识每个关联。这个参数与目的地址（出去）或源地址（进入）以及协议（AH或ESP）结合起来唯一地定义一个关联。

32.1.4　因特网密钥交换（IKE）

现在我们来解决最后的部分，如何创建SADB。**因特网密钥交换**（Internet key exchange，IKE）是用于建立SADB中入关联与出关联的一个协议。

IKE创建IPSec中的关联。

IKE是在其他二个协议（Oakley、SKEME和ISAKMP）之上的一个更复杂的协议，如图32.9所示。



因特网密钥交换（IKE）

图32.9　IKE的组件

Oakley协议是由Hilarie Orman开发的一个协议，它是在Diffie-Hellman密钥交换方法基础上的一个密钥生成协议，但做了一些改进。Oakley是一个无格式的协议，就是说没有规定交换的报文的格式。

SKEME是由Hugo Krawcyzk设计的，它是另一个密钥交换协议。它使用公钥密码的密钥交换作为实体鉴别。

因特网安全关联和密钥管理协议（Internet Security Association and Key Management Protocol，ISKAMP）是由国家安全机构（NSA）设计的协议，它具体实现IKE定义的交换。在生成标准化的和格式化的报文的过程中，它允许IKE交换的多个分组、协议及其参数，从而生成安全关联。

读者可能会问如何从发送方到接收方携带ISKAMP。这个协议为了下面的协议可使用而设计的。例如，分组在网络层或传输层中被当作有效载荷使用。当我们使用IPSec的时候，这个分组被IP协议作为一个有效载荷是很自然的，并在数据报中携带。现在下一个问题是，如何安全地交换携带ISAKMP的数据报？答案是没有必要，在ISAKMP分组中无需加密。

32.1.5　虚拟专用网

虚拟专用网（virtual private network，VPN）是一个在大型公司中获得广泛应用的技术，为了实现公司内部的和公司之间的通信，它们需要使用全球因特网，但在内部通信中又必须进行保密。这里我们讨论VPN，是因为它将IPSec协议用到IP数据报的安全性上。

专用网

专用网络是为组织机构内部使用而设计的。它允许对共享资源的访问，同时提供保密性。在讨论这些网络之前，先定义两个常用的有关术语：内联网和外联网。

内联网（intranet）是使用因特网模型的专用网（LAN）。然而，对网络的访问局限于组织机构内部的用户。网络使用针对全球因特网开发的应用程序，如HTTP，并且可能有Web服务

器、打印服务器、文件服务器等。

外部网（extranet）与内联网是相同的，但有一个主要差别：在网络管理员的监控下，组织机构外部的特定群体的用户可以访问某些资源。例如，组织机构可以允许授权的客户查看产品说明书、可用性以及在线订购，大学或学院可以让远程教育的学生在进行口令检查后接入计算机实验室。

寻址 使用因特网模型的专用网必须使用IP地址，有以下三种选择：

1. 可以向因特网管理机构申请一组地址，并在该网络不与因特网连接的情况下使用这些地址。这种策略的好处是：如果将来该组织机构决定连接到因特网，它就能相对容易地做到这一点。然而，这种策略也有一个坏处：地址空间被浪费掉了。

2. 网络可以在不向因特网管理机构申请的情况下使用任何地址。因为网络是独立的，地址无须保持唯一性。然而，这种策略有一个严重的缺陷：用户可能错误地将这些地址混淆为全球因特网的一部分。

3. 为了克服第一种和第二种策略中的问题，因特网管理机构已经保留了三组地址，如表32.2所示。

表32.2 专用网寻址

前 缀	范 围	总 数
10/8	10.0.0.0 – 10.255.255.255	2^{24}
172.16/12	172.16.0.0 – 172.31.255.255	2^{20}
192.168/16	192.168.0.0 – 192.168.255.255	2^{16}

任何组织机构都能在没有因特网管理机构允许的情况下使用这些地址。任何人都知道这些保留地址是为专用网使用的，它们在组织机构内部是唯一的，但在全球范围内是不唯一的，路由器不会将分组中的这些地址作为目的地址转发。

实现保密性

要实现保密性，组织机构可以使用以下三种策略之一：专用网络、混合网络和虚拟专用网。

专用网络 当路由信息在组织机构的内部时，如果组织机构需要实现保密性，则可使用前面所讨论的专用网络（private network）。在单个网点范围内的小型组织机构可以使用单独的LAN。组织机构内部的人员可以相互发送数据，对外部而言，内部是安全的。拥有多个网点的大型组织机构可以建立专用的互联网，不同网点的LAN可以使用路由器和租用线路相互连接起来。换言之，互联网可以由专用LAN和专用WAN组成。图32.10表示了一个有两个网点的组织机构的情况。LAN之间使用路由器和一条租用线路相互连接起来。



图32.10 专用网络

在这种情况下，组织机构建立了一个完全与因特网独立的专用互联网。对于不同网点的主机之间的端到端的通信，组织结构可以采用因特网模型。但是，组织机构没有必要向因特网管理机构申请IP地址，它可以使用专用IP地址。该组织机构可以使用任何类的IP地址，可以在内部进行网络和主机地址分配，因为互联网是专用，因此它与因特网中的另一个组织机构的地址重复不会产生任何问题。

混合网络　如今，大多数组织机构都需要在其内部的数据交换中实现保密性。但是，同时它们需要连接到因特网与其他组织机构进行数据交换。一种解决方案是采用混合网络（hybrid network）。混合网络允许组织机构拥有自身的专用互联网，同时还能访问因特网。组织机构内部的数据通过专用互联网路由，组织机构间的数据通过因特网路由。图32.11表示了这种情况下的一个例子。

拥有两个网点的组织机构通过一条租用线路，使用路由器R1和R2将两个网点进行专用连接。它使用路由器R3和R4将这两个网点连接到互特网。组织机构在两种类型的通信中都使用全局IP地址。但是目的地址为内部的分组通过路由器R1和R2路由，而目的地址为外部的分组通过路由器R3和R4路由。

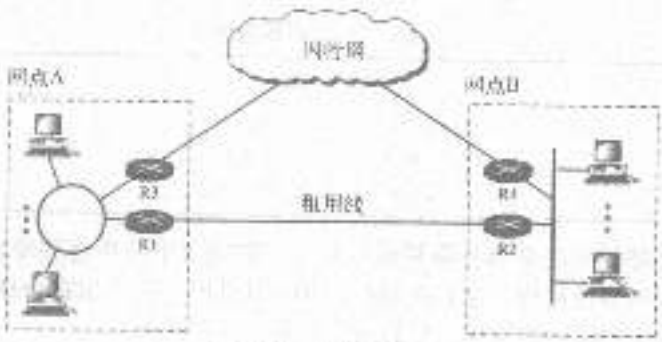


图32.11　混合网络

虚拟专用网　专用网络和混合网络都有一个主要的缺陷：费用问题。专用广域网（WAN）费用昂贵。连接多个网点，就需要多条租用线路，这意味着要付出高昂的月租费。一种解决方案是使用因特网来同时解决专用的和公开的通信。一种称为虚拟专用网（VPN）的技术就允许组织机构使用因特网同时达到以上两个目的。

VPN建立一个专用但是虚拟的网络，说它是专用的，是因为它可以保证在组织机构内部的保密性。说它是虚拟的，是因为它不使用实际的专用WAN，这种网络在物理上是公开的，但虚拟为专用的。

图32.12表示了虚拟专用网的思想。路由器R1和R2使用VPN技术保证组织机构的保密性。



图32.12　虚拟专用网

VPN技术

VPN技术在隧道方式下使用IPSec来提供鉴别、完整性以及保密性。

隧道技术 为了保证组织机构的保密性和其他安全措施，VPN可以在隧道方式下使用IPSec。在这种方式下，组织机构中每个专用的IP数据报都封装到另一个数据报中。为了在隧道技术（tunneling）中使用IPSec，VPN需要使用两组寻址方式，如图32.13所示。

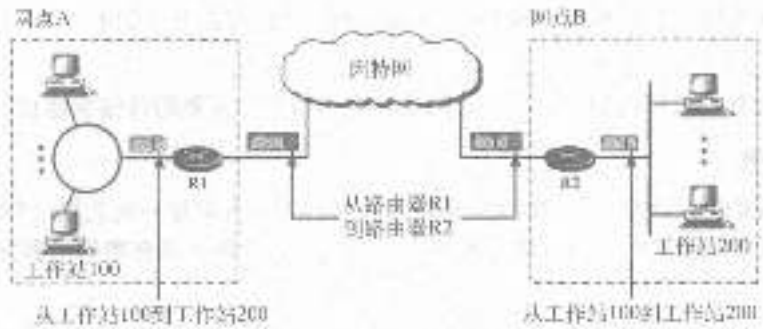


图32.13 VPN中的寻址

公共网络（因特网）负责将分组从R1传送到R2。外部人员无法解密分组的内容，也不能解密源地址与目的地址。解密过程发生在R2，它找到分组的目的地址并转发该分组。

32.2 SSL/TLS

传输层安全提供端到端的应用安全服务，该应用使用可靠的传输层协议，如TCP。其思想是为互特网上的事务提供安全服务。例如，当一位顾客在线购物时，要求下列安全服务：

1. 顾客需要确保服务器属于真正的销售商，而不是属于冒充者。顾客不希望将她自己的信用卡号交给一位冒充者（实体鉴别）。同样的，销售商也需要对顾客进行鉴别。

2. 顾客与销售商需要确保报文的内容在传输过程中没有被更改（报文完整性）。

3. 顾客与销售商需要确保冒充者截获不到诸如信用卡号之类的敏感信息（保密性）。

当今，在传输层提供安全性最主要的两个协议是安全套接字层（SSL）协议和传输层安全（TLS）协议。TSL实际上是SSL的IETF版本。我们首先讨论SSL，然后需要提及SSL与TLS的主要差别。图32.14表示了SSL和TLS在因特网模型中的位置。

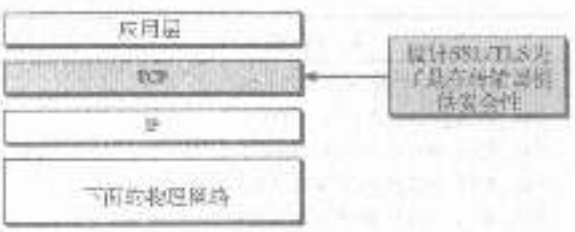


图32.14 因特网模型中SSL和TLS的位置

32.2.1 SSL服务

设计安全套接层（Secure Socket Layer，SSL）是为了对来自应用层的数据提供安全和压缩服务。SSL可接收到来自应用层任何协议的数据，一般典型的是HTTP协议。来自应用层的数据被压缩（可选的）、被签记和被加密，然后将它们传给可靠的传输层协议如TCP。Netscape在1994年开发了SSL，1995年发布第2版和第3版。本章讨论SSLv3。它对接收到的应用层数据提供了多个服务。

分段

SSL首先将数据划分为等于或小于2^{14}个字节的块。

压缩

在客户和服务器之间协商某一无损压缩方法对每个段进行压缩。这个服务是可选的。

报文完整性

要保持数据完整性，SSL使用密钥散列函数创建MAC。

保密性

为了提供保密性，原始数据和MAC使用对称密钥加密方法进行加密。

成帧

给被加密的有效载荷增加一个头部，然后将它传给一个可靠的传输层协议。

32.2.2 安全参数

在前一节我们讨论IPSec时，提到交换数据双方的任一方对每一次关联（SA）都需要一组参数。SSL具有相似的目的，但方法是不同的，没有安全关联，但有密码组和加密的密码一起组成安全关联。

密码组

对每次SSL会话，密钥交换、散列和加密算法一起定义密码组（cipher suite）。每个密码组以SSL开头后跟密钥交换算法并用WITH将密钥交换算法与加密算法和散列算法分开。例如，

```
SSL-DHE-RSA-WITH-DES-CBC-SHA
```

定义DHE-RSA（临时的Diffie-Hellman RSA数字签名）以DES-CBC作为加密算法进行密钥交换而SHA作为散列算法。注意：DH是固定的Diffie-Hellman，DHE是临时的Diffie-Hellman，而DH-anon是匿名的Diffie-Hellman。表32.3表示了在美国所用的密码组。我们没有包括那些用于出口的。注意：不是所有的密钥交换算法（建立报文鉴别和加密的密钥）、加密算法和鉴别算法组合包含在密码组列表中，在表中有些算法我们没有定义或讨论过，但为了完整说明使得读者对密码组有一个总的思想。

表32.3　SSL密码组表

密码组	密钥交换算法	加密算法	散列算法
SSL_NULL_*WITH*_NULL_NULL	NULL	NULL	NULL
SSL_RSA_*WITH*_NULL_MD5	RSA	NULL	MD5
SSL_RSA_*WITH*_NULL_SHA	RSA	NULL	SHA
SSL_RSA_*WITH*_RC4_128_MD5	RSA	RC4_128	MD5
SSL_RSA_*WITH*_RC4_128_SHA	RSA	RC4_128	SHA
SSL_RSA_*WITH*_IDEA_CBC_SHA	RSA	IDEA_CBC	SHA
SSL_RSA_*WITH*_DES_CBC_SHA	RSA	DES_CBC	SHA
SSL_RSA_*WITH*_3DES_EDE_CBC_SHA	RSA	3DES_EDE_CBC	SHA
SSL_DH_anon_*WITH*_RC4_128_MD5	DH_anon	RC4_128	MD5
SSL_DH_anon_*WITH*_DES_CBC_SHA	DH_anon	DES_CBC	SHA
SSL_DH_anon_*WITH*_3DES_EDE_CBC_SHA	DH_anon	3DES_EDE_CBC	SHA
SSL_DHE_RSA_*WITH*_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
SSL_DHE_RSA_*WITH*_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
SSL_DHE_DSS_*WITH*_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
SSL_DHE_DSS_*WITH*_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
SSL_DH_RSA_*WITH*_DES_CBC_SHA	DH_RSA	DES_CBC	SHA

（续）

密 码 组	密钥交换算法	加密算法	散列算法
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
SSL_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
SSL_FORTEZZA_DMS_WITH_NULL_SHA	FORTEZZA_DMS	NULL	SHA
SSL_FORTEZZA_DMS_WITH_FORTEZZA_CBC_SHA	FORTEZZA_DMS	FORTEZZA_CBC	SHA
SSL_FORTEZZA_DMS_WITH_RC4_128_SHA	FORTEZZA_DMS	RC4_128	SHA

加密的密码

安全参数的第二部分时常称为加密的密码。为了报文的完整性和保密性，SLL需要6个密码加密的密码，4个密钥和2个IV。

> 客户与服务器需要有6个不同的加密的密码。

生成这些密码的加密的过程，如图32.15所示。客户机需要一个用于报文鉴别的密钥，一个用于加密的密钥和一个用于块加密的密钥，服务器的需要也相同。SSL要求一个方向上的这些密钥与另一个方向上的密钥不同。这样，如果一方遭到攻击，另一方不受影响。下面将看到通过使用一个协商协议生成这些参数。

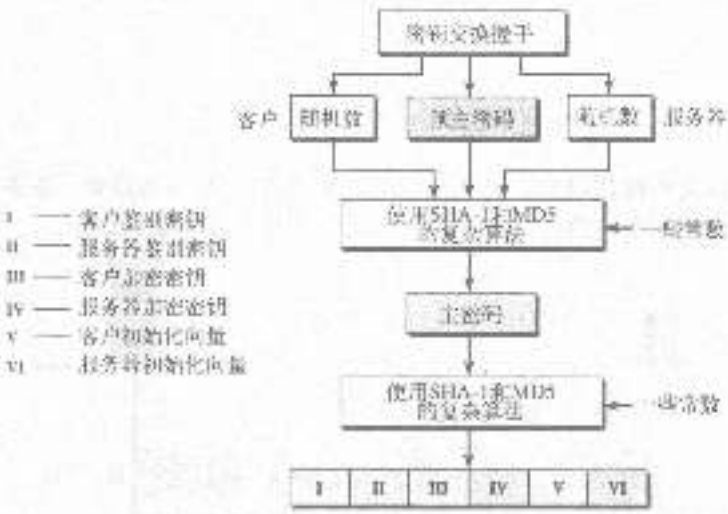


图32.15 SLL中生成密码的加密

1. 客户和服务器交换两个随机数，一个是客户生成的，另一个是服务器生成的。
2. 客户和服务器用前面讨论过的某一密钥交换算法交换预主密码（premaster secret）。
3. 对预主密码应用两个散列函数（SHA-1和MD5）生成48位的主密码（master secret）。
4. 对主密码应用预先规定的不同常数和同一组散列函数生成可变长的密码。

32.2.3 会话与连接

IP和TCP协议的特性是不同的。IP是无连接协议，而TCP是面向连接的协议。IPSec中的关联将无连接IP转换成面向连接的安全协议，而TCP已是面向连接的。然而SSL设计者采用二级连接性的办法：会话（session）和连接（connection）。两个系统之间的会话是一次关联。它能持续有很长的一段时间。在一次会话期间，连接可建立和中断若干次。

在会话建立期间生成的一些安全参数，一直有效直到会话终止（例如，加密组和主密钥）。有些安全参数对每个连接必须重新生成（或偶尔是可恢复的）（例如，6个密码）。

32.2.4　四个协议

我们已讨论过SSL的思想，但没有说明SSL如何实施它的任务。SSL在两层上定义了四个协议，如图32.16所示。**记录协议**（Record Protocol）是承载者，它携带来自其他三个协议的报文，也携带来自应用层的数据。对传输层，通常是TCP来说，来自记录协议的报文是它的有效载荷。**握手协议**（Handshake Protocol）向记录协议提供安全参数，它建立一个密码组并提供密钥和安全参数。它也实施用户对服务器的鉴别，以及如需要的话服务器对用户的鉴别。**改变密码规格协议**（ChangeCipherSpec Protocol）用于发出加密密码准备就绪的通知。**警告协议**（Alert Protocol）用于报告不正常情况。本节将简要讨论这些协议。

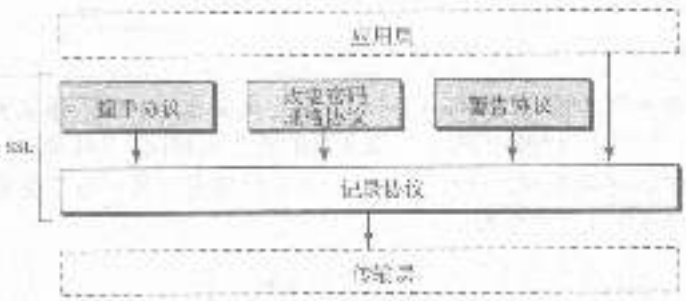


图32.16　四个SSL协议

握手协议

握手协议使用报文协商密码组、实现客户对服务器的鉴别、（如需要）服务器对客户的鉴别，并通过建立加密密码交换信息。握手用四个阶段完成。如图32.17所示。

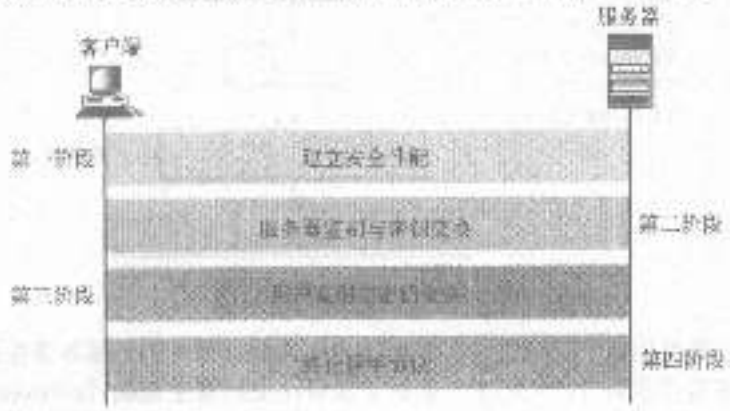


图32.17　握手协议

改变密码规格协议

我们已看到密码组的协商和加密密码的生成都是在握手协议期间逐步生成的。现在的问题是，双方何时使用这些安全参数？SSL规定双方在发送或接收到一个特殊报文（改变密码规格报文）之前不使用这些参数或密码。这个特殊的报文是在握手协议中交换的，在改变密码规格协议中定义的。在改变密码规格报文交换前，有价值的仅是悬而未用的列。

警告协议

SSL使用警告协议报告差错和不正常情况。它仅有一种类型的报文，即警告报文。它说明警告问题以及其强度（缓和的或致命的）。

记录协议

记录协议携带来自上层（握手协议、改变密码规格协议和警告协议或应用层）的报文。报文被分段和选择性地压缩。通过使用协商的散列算法将MAC增加到压缩报文中。被压缩的段和MAC用协商的加密算法加密。最后，将SSL头部加到加密的报文中。图32.18表示了在发送方这个过程以及在接收方的相反过程。

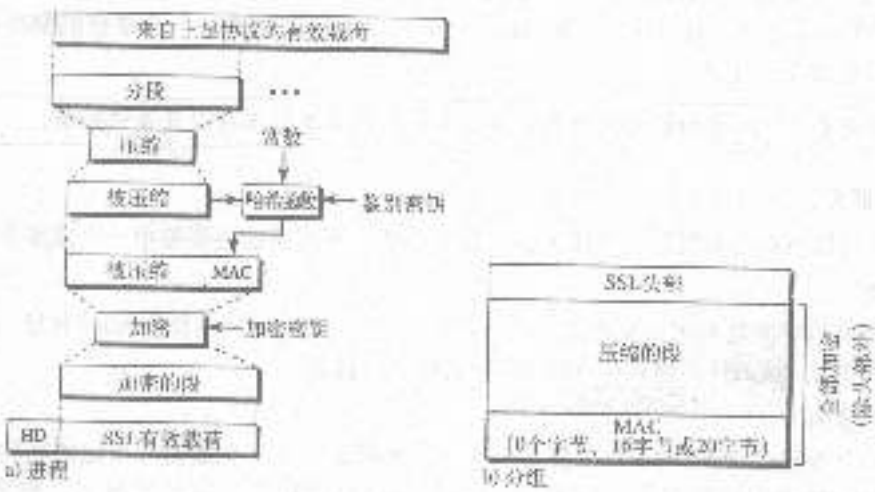


图32.18 记录协议实施过程

32.2.5 传输层安全

传输层安全（Transport Layer Security，TLS）是IETF的SSL版本。两者很相似，但略有不同。我们在下面强调不同处：

- 版本。本节讨论的SSLv3.0是与TLSv1.0一致的；
- 密码组。TLS密码组不支持Fortezza；
- 加密密码。加密密码生成有几种不同的方法。TSL用伪随机数函数（pseudorandom function，PRF）生成主密钥以及密钥资料；
- 警告协议。TLS删除某些警告报文并增加某些新的警告；
- 握手协议。某些报文的细节在TSL中已改变；
- 记录协议。TLS不用MAC而用在第31章定义的HMAC。

32.3 PGP

在应用层提供安全性的协议中一个是**相当好的保密性**（Pretty Good Privacy，PGP）协议。PGP用来生成鉴别的与安全的电子邮件。图32.19表示了在TCP/IP协议族中的PGP的位置。

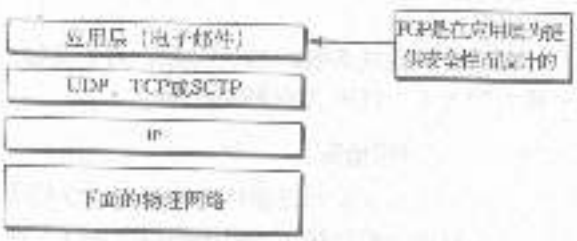


图32.19 在TCP/IP协议族中的PGP的位置

发送电子邮件是一次活动，活动的实质与前面两节所看到是不同的。在IPSec

或SSL中，我们假定双方在他们之间建立一个会话，并在两个方向上交换数据。而在电子邮件中不存在会话，Alice与Bob不可能生成会话。Alice发送电子邮件给Bob，而在以后的某一时间Bob才阅读该邮件，他可以发送回答或不回答。我们只要讨论单向报文的安全性就可以，因为Alice向Bob发送报文与Bob向Alice发送报文完全是独立的。

32.3.1　安全参数

如果电子邮件是一种一次性的活动，那么发送方和接收方如何能确定安全参数为电子邮件安全使用呢？如果没有会话和没有握手协商加密和鉴别的算法，接收方如何能知道发送方为每次目的选取哪一种算法？接收方如何能知道所用的加密和鉴别密钥的值呢？

PGP的设计者和开发者Phil Zimmerman提供了上述问题的一个非常好的解决办法。要求安全参数与报文一起送。

> 在PGP中，发送方的报文需要包括该报文所用的算法标识符以及密钥的值。

32.3.2　服务

PGP可按用户要求提供多个服务，一封电子邮件可使用这些服务中一个或多个。

明文

最简单的情况是用明文发送电子邮件的报文（没有服务）。发信人Alice书写一个封邮件发给接收人Bob。该邮件存储在Bob的邮箱中直到被他检索。

报文鉴别

下一个改进或许是让Alice在报文上签名，她创建一个报文摘要并用她的私钥在摘要上签名。当Bob接收到该报文后，他用Alice的公钥验证该报文。这种情况下，需要两个密钥，Alice需要知道她的私钥，而Bob需要知道Alice的公钥。

压缩

进一步改进是压缩报文和摘要使分组更紧凑。这种改进对安全没有贡献而对减少通信量带来好处。

一次性会话密钥的保密性

正如我们前面讨论的那样，电子邮件系统中的保密性可用一次性会话密钥的传统加密方法实现。Alice可生成一个会话密钥，使用会话密钥加密报文和摘要，并随报文一起发送密钥本身。但是，为了保护会话密钥，Alice用Bob的公钥加密它。

代码变换

PGP提供的另一种服务是代码变换。大多数电子邮件系统只允许使用ASCII码的文本。为了对其他不是ASCII字符提供转换方案，PGP使用基数为64的变换。每个发送的字符（加密后）被变成基数为64的代码。

分段

在已被转换成基数64的代码后，为了使每个被传输的单元能适应下面电子邮件协议的统一格式的大小，PGP允许报文分段。

32.3.3　一种设想情形

让我们说明鉴别和保密性两种服务组合的情况。PGP总的思想是假定基于一组人需要彼此信任地交换电子邮件报文，组中的每个人以某种方式知道（具有信任）其他任何人的公钥。按照这个单一的假定，图32.20表示了Alice发送一个已鉴别的和已加密的报文到Bob的简单情形。

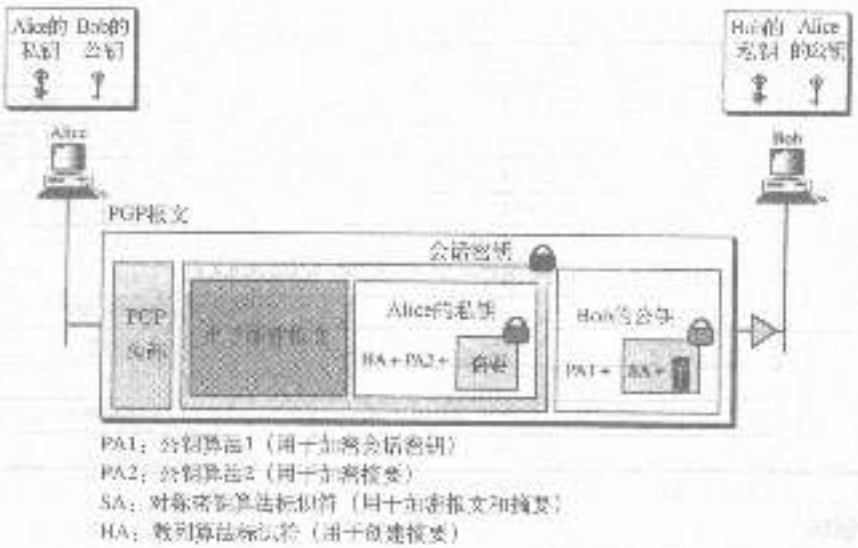


图32.20 一个已签别的和已加密的电子邮件报文的情形

发送方

在这种设想情形下，Alice的步骤如下：

1. Alice生成一个会话密钥（用于对称的加密/解密），并把它与它将要使用的加密钥算法串接在一起，用Bob的公钥进行加密。然后Alice在加密的结果前面添加公钥加密算法的标识符。报文的这个部分包含三个信息：会话密钥、将要使用的对称加密/解密算法和对这个部分已使用的不对称加密/解密算法。

2. a. Alice用公钥签名（散列）算法签别报文（电子邮件），并用她的私钥加密它。所得到的结果称为数字签名。Alice在签名前附加公钥标识符（用于加密的）和签名散列算法标识符（用于签别）。报文的这部分包含签名和二个附加信息：加密算法和散列算法。

b. Alice将上面生成的三个信息部分与报文（电子邮件）串接在一起，用步骤1生成的会话密钥加密串接的全部信息。

3. Alice合并步骤1和2的结果，（添加适当的PGP头部）向Bob发送。

接收方

在这种设想情形下Bob在接收到PGP分组后，其步骤如下：

1. Bob使用他的私钥解密会话密钥和对称密钥算法标识符。

2. Bob使用步骤1所得到的会话密钥和算法解密PGP报文的其余部分。现在Bob有了报文的内容、用于生成和加密签名的公钥算法标识符以及用于生成散列报文的散列算法标识符。

3. Bob使用Alice的公钥和PA2定义的算法解密摘要。

4. Bob使用第2步获得的HA的散列算法生成报文的散列代码。

5. Bob比较第3步解密摘要与第4步散列代码。如果相同，他接收报文；否则，丢弃报文。

32.3.4 PGP算法

表32.4表示了PGP所用的算法。该表是不完全的，因为新的算法不断在增加。

表 32.4

算　法	ID	说　明
公钥	1	RSA（加密或签名）
	2	RSA（仅用于加密）
	3	RSA（仅用于签名）
	17	DSS（用于签名）
散列算法	1	MD5
	2	SHA-1
	3	RIPE-MD
加密	0	不加密
	1	IDEA
	2	三重DES
	9	AES

32.3.5 密钥环

在前面的设想中，我们假定Alice仅仅只向Bob发送报文，但通常并不是如此。Alice可能需要给多个人发送报文。此时，Alice需要一个公钥的密钥环（key ring），环中的每个密钥属于Alice需要通信的每个人。另外，PGP设计者还指定了一个私钥/公钥对的环。一个理由是Alice想要经常更换她的密钥对，另一个理由是Alice可能需要对不同的人分组（朋友，同事等等），对每组使用不同的密钥对。所以，每个用户需要两个环：一个私钥/公钥对环和其他人的公钥环。图32.21表示了一个4个人的团体。每个人的私钥/公钥对环以及其他人的公钥环。图32.21表示了每个公钥环的7个公钥。在该环中的每个人保存其他人的公钥。

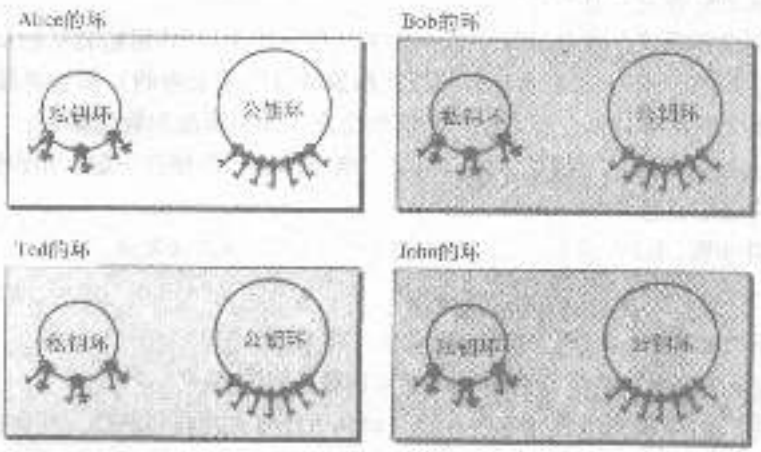


图32.21 环

例如，Alice有多个属于她的和其他人的私钥/公钥对，注意：每个人可能有多个公钥。两种情况可出现。

1. Alice发送一个报文给团体中某一个人，
   a. 她用她的私钥对摘要签名；
   b. 她用接收方的公钥对一个新生成的会话密钥进行加密；
   c. 她用会话密钥对报文和摘要进行加密。
2. Alice接收到来自团体中某个人的报文。

a. 她用她的私钥解密会话密钥；

b. 她用会话密钥解密报文和摘要；

c. 她用她的公钥认证摘要。

32.3.6 PGP证书

为了信任公钥的持有者，在PGP组中的每个使用者都需要隐式地或显式地拥有公钥持有者的证书副本。虽然证书可来自认证中心（CA），但是，在PGP中不要求这个规定。PGP有它自己的认证系统。

使用X509证书的协议基于信任链的层次结构，预先定义一个从根到任何证书的信任链。每个用户完全信任在根级（前提）的CA机构，根对在第二级的一些CA发出证书，一个第二级CA对在第三级发出证书，等等。需要被信任的每一方呈现来自树的某一CA的一份证书。如Alice不信任Bob发出的认证书，她可求助高一级机构直到根级（它对系统运行必须信任），换言之，存在着从完全信任的CA到一个证书的一条单一路径。

在PGP中，不需要有CA。环中任何一个人都可对环中其他任何人签署一个证书，表示他担保公钥正确。例如Bob可以为Ted、Jhon和Anne等签署证书。在PGP中信任没有层次结构，也没有树，正当为没有层次结构，Ted可从Bob获得一个证书和从Liz获得另一个证书。如果Alice想要沿着Ted证书的线路，则有二条路径：一条由Bob出发，另一条由Liz出发。一个引起关注的问题是Alice完全信赖Bob，但仅部分信赖Liz。为了获得一个证书，存在完全信任或部分信任的多条路线。在PGP中，证书的介绍者通常称为介绍人（introducer）。

> 在PGP中，对于任何对象可以有完全信任或部分信任的多条路径。

信任和合法性

PGP的全部操作是基于介绍人的信任、证书的信任和公钥的合法性。

介绍人信任级 由于没有中央权威机构，显然，如果用户PGP环中每一个用户必须完全信任其他每一个，则环不能很大。（在实际生活中，我们不可能对认识的每一个人都完全信任）为了解决这个问题，PGP允许信任有不同级别。级别数大多与实现有关，但为了简单起见，让我们对介绍人指定三级：无信任、部分信任和完全信任。介绍人信任（introducer trust）级是环中其他介绍人指定的，例如，Alice可完全信任Bob和部分信任赖Anne，对John完全不信任。PGP没有机制去确定介绍人的信任程度，它按用户的想法做出决定。

证书信任级 当Alice接收到介绍人的一个证书时，她将该证书存储在对应名（已签定的实体）下。证书信任（certificate trust）级通常与介绍证书的介绍人的信任级相同。假定Alice完全信任Bob，部分信任Anne和Janette，不信任John，则会发生下列设想的情况。

1. Bob介绍两个证书：一个是Linda的证书（用公钥K1），另一个是Lesley的证书（用公钥K2）。Alice存储Linda名下的公钥和Linder的证书，并指定该证书为完全信任级。Alice也存储Lesley名下的证书和公钥，并指定该证书为完全信任级。

2. Anne发出一个John的证书（用公钥K3）。Alice存储这个证书和John名下的公钥，但她对这个证书指定部分信任。

3. Janette发出两个证书，一个是John的证书（用公钥K3），另一个是Lee的证书（用公钥K4）。Alice存储John名下的证书和Lee名下的证书，每个证书指定部分信任级。注意，现在有两份John的证书，每一份都是部分信任级。

4. John发出Liz的证书，Alice可丢弃或保存该证书为不信任级。

密钥合法性 使用介绍人与证书信任级的目的是决定公钥的合法性。Alice需要了解Bob、

John、Liz、Anne等的公钥是否合法，PGP用一个很简单的办法定义密钥合法性（key legitimacy）。一个用户密钥合法性的级是该用户的权重信任级。例如，假设我们对证书指定下列的权：

1. 对不信任的证书，权为0；

2. 对部分信任的证书，权为1/2；

3. 对完全信任的证书，权为1。

然后，要完全信任一个实体，Alice需要有一个完全信任的证书或两个部分信任的证书。例如，在前面假想情况下，由于Anne和Janette已介绍过John的证书，每个证书的信任级为1/2，因此Alice可使用John的公钥。注意：属于一个实体的公钥的合法性与那个人的信任级别无关。虽然Bob可用John的公钥发送一个报文给他，但是Alice不接受John介绍的证书，因为对Alice来说，John是不可信任的。

启动环

读者可能会问上面讨论问题如何实现。如果没有人为一个完全信任或部分信任的实体发送证书，将会怎样？例如，如果没有人发送Bob的证书，那么如何确定Bob公钥的合法性？PGP中，一个信任或部分信任的实体的公钥合法性也可用其他方法确定：

1. Alice直接获得Bob的公钥。例如，Alice与Bob直接碰面交换写在纸上或台子上的公钥；

2. 如果Alice能识别Bob的语音，则Alice可打电话获得他的公钥；

3. PGP提出一个较好的解决办法是Bob通过电子邮件发送他的公钥给Alice。Alice与Bob两人各自都将该公钥生成一个16字节的MD5（或一个20字节SHA-1）摘要，这个摘要用8组表示，每组是4个十六进制数，这称为指纹（fingerprint）。然后Alice可通过电话验证指纹，如果在邮件传输期间，公钥被改动或改变，则两个指纹就不匹配。甚至为了更方便些，PGP创建一个字表。表中每一个字代表一个4个数的组合。当Alice打电话给Bob时，Bob可发出8个字（或10个字）的声音给Alice。PGP精心挑选这些字避免重复。例如，如果sword在表中，则word就不在表中；

4. PGP不阻止Alice在单一过程中从CA获得Bob的公钥，然后她将该公钥插入公钥环中。

信任网

最后PGP在一组人之间形成信任网（web of trust）。如果每个实体向其他实体介绍多个实体时，那么每个实体的公钥环越来越大。环中的实体彼此可发送它安全的电子邮件。

密钥废除

一个实体需要在自己的公钥环中回收公钥。如果密钥持有者怀疑该密钥不安全（例如，被窃）或有效期将到，为了废除一个公钥，持有者可以发送一个她自己签名的废除证书。废除证书必须签用旧的公钥签名。并传播给环中所有使用该公钥的人。

32.4　防火墙

上述所有安全措施都无法阻止Eve向系统发送有害的报文。使用防火墙可以达到对系统访问进行控制的目的。防火墙（firewall）是一种安装在组织机构的内部网络与因特网之间的设备（通常是路由器或者计算机）。它用于转发某些分组而过滤掉（不转发）其他分组。图32.22表示一个防火墙。

例如，防火墙可以过滤掉所有目的地址为特定主机或特定服务器（比如HTTP）的分组。防火墙在组织机构中用来拒绝对特定主机或特定服务的访问。

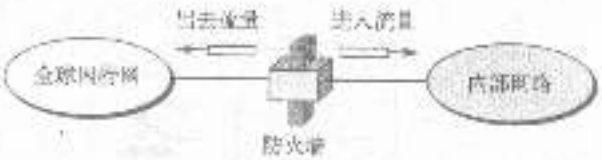
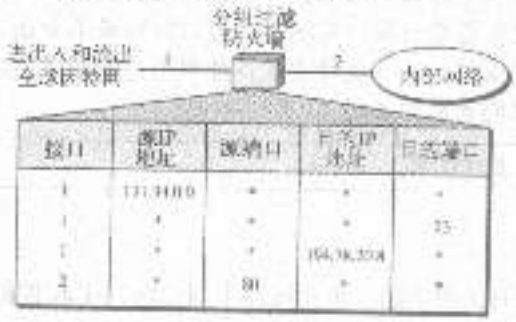


图32.22 防火墙

防火墙通常划分为分组过滤防火墙和基于代理的防火墙。

32.4.1 分组过滤防火墙

防火墙可以用作分组过滤器。它可以根据网络层和传输层的头部信息来转发或阻止分组，这些头部信息是指：源和目的IP地址、源端口和目的端口地址以及协议类型（TCP或者UDP）。**分组过滤防火墙**（packet-fileter firewall）是使用过滤表来确定抛弃（不转发）哪些分组的路由器。图32.23表示了这种类型防火墙过滤表的一个例子。



接口	源IP地址	源端口	目的IP地址	目的端口
1	131.34.0.0	*	*	*
1	*	*	*	23
1	*	*	194.78.20.8	*
2	*	80	*	*

图32.23 分组过滤防火墙

根据图32.23，对下列分组进行过滤：

1. 对来自网络131.34.0.0的分组要进行阻止（安全防范），注意*（星号）意思是“任何一个”。

2. 对目标是任何内部TELNET服务器（端口为23）的进入分组要进行阻止。

3. 对目标是内部主机194.78.20.8的进入分组要进行阻止。组织机构只希望该主机作内部使用。

4. 对目标是HTTP服务器（端口为80）的流出分组进行阻止。组织机构不希望雇员浏览因特网。

> 分组过滤防火墙在网络层或传输层进行过滤。

32.4.2 代理防火墙

分组过滤防火墙是基于在网络层和传输层头部（IP和TCP/UDP）中可利用的信息进行工作的。然而，有时我们需要基于报文自身（在应用层）中可利用的信息进行报文过滤。例如，假设组织机构需要实现以下关于其Web页面的策略：只有那些先前已与公司建立商业联系的因特网用户才拥有访问权，同时阻止其他用户的访问。在这种情况下，分组过滤防火墙是行不通的，因为它无法辨别到达TCP端口80（HTTP）的不同的分组。检查必须在应用层（使用URL）完成。

一种解决方案是安装一台代理计算机（有时称为应用网关），它位于顾客（用户客户端）计算机和公司计算机之间，如图32.24所示。

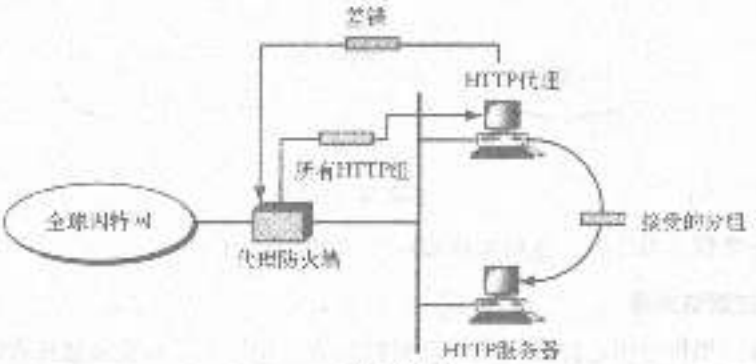


图32.24　代理防火墙

当用户客户端进程发送一份报文时，代理防火墙（proxy firewall）运行服务器进程来接收该请求。服务器在应用层打开该分组，查看该请求是否合法。如果请求合法，服务器便以客户端进程的身份将报文发送给公司中实际的服务器。如果不合法，报文就被抛弃，同时向外部的用户发送一份错误报文。通过这种方式，外部用户的请求在应用层进行了基于内容的过滤。图32.24表示了代理防火墙的实现。

代理防火墙在应用层进行过滤。

推荐读物

为了深入讨论与本章有关的主题，推荐阅读下列读物，在本书末列出方括号[…]中的参考资料。

读物

IPSec在[Rhe03]中的第7章、[PHS03]的18.1节以及[KPS02]的第17章和18章中有讨论。IPSec完整的讨论可在[DH03]中找到。SSL/TSL在[Rhe03]中的第8章和[KPS02]第9章有讨论。完整SSL和TSL的讨论可在[Res01]和[Tho00]找到。PGP在[Rhe03]的第9章和[KPS02]的第22章有讨论。防火墙在[Rhe03]的第10章和KPS02]的第23章有讨论。防火墙完整的讨论见[CBR03]。虚拟专用网在[YS01]和[SWE99]中有完整的讨论。

本章小结

- IP层安全（IPSec）是由IETF（因特网工程任务组）设计用来提供网络分组安全。
- IPSec以传输方式或隧道方式操作。
- 在传输方式中，IPSec保护传输层到网络层的信息，不保护IP头部。当我们需要主机到主机（端到端）的数据保护时，通常使用传输方式。
- 在隧道方式中，ISPec保护全部IP分组，包括原IP头部。
- IPSec定义两种协议：鉴别头部（AH）协议和封装安全载荷协议（ESP）。在IP级对分组提供鉴别和/或加密。
- IPSec要求两个主机之间称为安全关联（SA）的逻辑关系。IPSec使用称为安全关联数据库或SADB的一组SA。
- 因特网密钥交换（IKE）是为创建流入和流出的安全关联而设计的协议。IKE为IPSec创建SA。

- IKE是基于另外三个协议：Oakley，SKEME和ISAKMP之上的一个复杂协议。
- 专用网络是一个组织机构内部用的网络。
- 内联网是使用因特网模型的专用网，而外部网是一个允许外部特定群体的用户可以访问的内联网。
- 因特网机构为专用网已保留地址。
- 虚拟专用网（VPN）为必须经过因特网通信的局域网提供保密性。
- 传输层安全协议为可靠的传输层协议如TCP的应用提供端到端的安全服务。
- 当前为传输层提供安全的两个主要协议是安全套接层（SSL）和传输层安全（TLS），第二个协议实际上是第一个协议的IETF版本。
- SSL是为来自应用层数据提供安全和压缩服务而设计的，典型的SSL可接收任何应用层协议的应用数据，但是协议通常是HTTP。
- SSL对接收到来自应用层的数据提供诸如分段、压缩、报文完整性、保密性和成帧等服务。
- 密钥交换、散列处理和加密算法的组合定义了每次SSL会话的密码组，每组的名字是组合的描述。
- 在电子邮件中，加密算法与密码附加在报文中发送。
- 电子邮件的一个安全协议是相当好的保密性（PGP），PGP是Phi Zimmerman开发的，提供邮件的保密性、完整性和鉴别性。
- 要交换电子邮件报文，用户需要公钥环，每次电子邮件通信需要一个公钥。
- PGP也指定私钥/公钥对的环允许用户时时改变密钥对。PGP也允许用户有不同的ID（邮件地址）用于不同的人群。
- PGP证书与X.509证书是不同的。在X.509中从完全信任的机构到任何证书只有单一的一条路径，而在PGP中，从完全信任或部分信任中心有多条路径。
- PGP使用证书信任级的思想。
- 当一个用户接收到介绍人发来的一个证书时，存储对象（已鉴别的实体）名的证书并指定该证书的信任级。

习题

复习题

1. 为什么IPSec需要安全关联？
2. IPSec如何生成一组安全参数？
3. IPSec定义的两个协议是什么？
4. AH在IP分组中加入什么信息？
5. ESP在IP分组中加入什么信息？
6. IP安全是否同时需要AH和ESP？为什么？
7. 在传输层本章所提供安全性的两个协议是什么？
8. IKE是什么？
9. 在SSL中，一次会话与一次连接有什么不同？
10. SSL如何生成一组安全参数？
11. 本章提供电子邮件安全性的协议叫什么？
12. PGP如何生成一组安全参数？
13. SSL中握手协议的目的是什么？

14. SSL中记录协议的目的是什么？
15. 防火墙的目的是什么？
16. 防火墙类型有哪两种？
17. VPN是什么？为什么要它？
18. 在一个完全专用的互联网上的局域网之间如何通信？

练习题

19. 试计算图32.6中AH字段中的值，假定鉴别数据是128位。
20. 试计算图32.7中ESP头部与尾部各字段的值。
21. 在隧道方式下使用AH协议，重新画出图32.6。
22. 在隧道方式下使用ESP协议，重新画出图32.7。
23. 试用图说明AH在IPv6中位置。
24. 试用图说明ESP在IPv6中位置。
25. IPSec协议需要KDC服务吗？试解释之。
26. IPSec协议需要CA服务吗？试解释之。
27. SSL协议需要KDC服务吗？试解释之。
28. SSL协议需要CA服务吗？试解释之。
29. PGP协议需要KDC服务吗？试解释之。
30. PGP协议需要CAQ服务吗？试解释之。
31. 在IPSec中，是否有密码组？试解释之。
32. 在PGP中，是否有密码组？试解释之。

参考文献

[AL98] Albitz, P. and Liu, C. *DNS and BIND*. Sebastopol, CA: O'Reilly, 1998.

[AZ03] Agrawal D. and Zeng, Q. *Introduction to Wireless and Mobile Systems*. Pacific Grove, CA, NJ: Brooks/Cole Thomson Learning, 2003.

[Bar02] Barr, T. *Invitation to Cryptology*. Upper Saddle River, NJ: Prentice Hall, 2002.

[BEL00] Bellamy, J. *Digital Telephony*. New York, NY: Wiley, 2000.

[Ber96] Bergman, J. *Digital Baseband Transmission and Recording*. Boston, MA: Kluwer, 1996.

[Bis03] Bishop, M. *Computer Security*. Reading, MA: Addison-Wesley, 2003.

[Bla00] Black, U. *QoS In Wide Area Network*. Upper Saddle River, NJ: Prentice Hall, 2000.

[Bla03] Blahut, R. *Algebraic Codes for Data Transmission*. Cambridge, UK: Cambridge University Press, 2003.

[CRR03] Chesvick, W., Bellovin, S., and Rubin, A. *Firewalls and Internet Security*. Reading, MA: Addison-Wesley, 2003.

[Com00] Comer, D. *Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture*. Upper Saddle River, NJ: Prentice Hall, 2000.

[Com04] Comer, D. *Computer Networks*. Upper Saddle River, NJ: Prentice Hall, 2004.

[Cou01] Couch, L. *Digital and Analog Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2000.

[DH03] Doraswamy, H. and Harkins, D. *IPSec*. Upper Saddle River, NJ: Prentice Hall, 2003.

[Dro02] Drozdek, A. *Elements of Data Compression*. Brooks/Cole Thomson Learning, 2003.

[Dut01] Dutcher, D. *The NAT Handbook*. New York, NW: Wiley, 2001.

[FH98] Ferguson, P. and Huston, G. *Quality of Service*. New York, NW: Wiley, 1996.

[For03] Forouzan, B. *Local Area Networks*. New York, NY: McGraw-Hill, 2003.

[For06] Forouzan, B. *TCP/IP Protocol Suite*. New York, NY: McGraw-Hill, 2006.

[FRE96] Freeman, R. *Telecommunication System Engineering*. New York, NW: Wiley, 1996.

[Gar01] Garret, P. *Making, Breaking Codes*. Upper Saddle River, NJ: Prentice Hall, 2001.

[Gas02] Gast, M. *802.11 Wireless Network*. Sebastopol, CA: O'Reilly, 2000.

[GW04] Garcia, A. and Widjaja, I. *Communication Networks*. New York, NY: McGraw-Hill, 2003.

[Hal01] Halsall, F. *Multimedia Communication*. Reading, MA: Addison-Wesley, 2001.

[Ham80] Hamming, R. *Coding and Information Theory.* Upper Saddle River, NJ: Prentice Hall, 1980.

[Hsu03] Hsu, H. *Analog and Digital Communications.* New York, NY: McGraw-Hill, 2003.

[Hui00] Huitema, C. *Routing in the Internet.* Upper Saddle River, NJ: Prentice Hall, 2000.

[Izz00] Izzo, P. *Gigabit Networks.* New York, NY: Wiley, 2000.

[Jam03] Jamalipour, A. *Wireless Mobile Internet.* New York, NY: Wiley, 2003.

[KCK98] Kadambi, j., Crayford, I., and Kalkunte, M. *Gigabit Ethernet.* Upper Saddle River, NJ: Prentice Hall, 1998.

[Kei02] Keiser, G. *Local Area Networks.* New York, NY: McGraw-Hill, 2002.

[Kes97] Keshav, S. *An Engineering Approach to Computer Networking.* Reading, MA: Addison-Wesley, 1997.

[KMK04] Kumar A., Manjunath, D., and Kuri, J. *Communication Networking.* San Francisco, CA: Morgan, Kaufmans, 2004.

[KPS02] Kaufman, C., Perlmann, R., and Speciner, M. *Network Security.* Upper Saddle River, NJ: Prentice Hall, 2000.

[KR05] Kurose, J. and Ross, K. *Computer Networking.* Reading, MA: Addison-Wesley, 2005.

[Los04] Loshin, P. *IPv6: Theory, Protocol, and Practice.* San Francisco, CA: Morgan, Kaufmans, 2001.

[Mao04] Mao, W. *Modern Cryptography.* Upper Saddle River, NJ: Prentice Hall, 2004.

[Max99] Maxwell, K. *Residential Broadband.* New York, NY: Wiley, 2003.

[MOV97] Menezes, A., Oorschot, P., and Vanstone, S. *Handbook of Applied Cryptography.* New York, NY: CRC Press, 1997.

[Moy98] Moy, J. *OSPF: Anatomy of an Internet Routing Protocol.* Reading, MA: Addison-Wesley, 1998.

[MS01] Mauro D. and Schmidt K. *Essential SNMP.* Sebastopol, CA: O'Reilly, 2001.

[PD03] Peterson, L., and Davie B. *Computer Networks: A Systems Approach.* San Francisco, CA: Morgan, Kaufmans, 2000.

[Pea92] Pearson, J. *Basic Communication Theory.* Upper Saddle River, NJ: Prentice Hall, 1992.

[Per00] Perlman, R. *Interconnection: Bridges, Routers, Switches, and Internetworking Protocols.* Reading, MA: Addison-Wesley, 2000.

[PHS03] Pieprzyk, J., Hardjono, T. and Seberry, J., *Fundamentals of Computer Security.* Berlin, Germany: Springer, 2003.

[Res01] Rescorla, E. *SSL and TSL.* Upper Saddle River, NJ: Prentice Hall, 2000.

[Rhe03] Rhee, M, *Internet Security.* New York, NY: Wiley, 2003.

[Ror96] Rorabaugh, C. *Error Coding Cookbook.* New York, NY: McGraw-Hill, 1996.

[Sal03] Solomon, D. *Data Privacy and Security.* Berlin, Germany: Springer, 2003.

[Sau98] Sauders, S. *Gigabit Ethernet Handbook*. New York, NY: McGraw-Hill, 1998.

[Sch96] Schneier, B. *Applied Cryptography*. Reading, MA: Addison-Wesley, 1996.

[Sch03] Schiller, B. *Mobile Communications*. Reading, MA: Addison-Wesley, 2003.

[Spi74] Spiegel, M. *Fourier Analysis*. New York, NY: McGraw-Hill, 1974.

[Spu00] Spurgeon, C. *Ethernet*. Sebastopol, CA: O'Reilly, 2000.

[SSS05] Shimonski, R., Steuer, R. Sheedy, S. *Network Cabling Illustrated*. Sudbury, MA: Jones and Bartlette, 2005.

[Sta02] Stallings, W. *Wireless Communications and Networks*. Upper Saddle River, NJ: Prentice Hall, 2002.

[Sta03] Stallings, W. *Cryptography and Network Security*. Upper Saddle River, NJ: Prentice Hall, 2003.

[Sta04] Stallings, W. *Data And Computer Communications*. Upper Saddle River, NJ: Prentice Hall, 2004.

[Sta98] Stallings, W. *High Speed Networks*. Upper Saddle River, NJ: Prentice Hall, 1998.

[Ste94] Stevens, W. *TCP/IP Illustrated, Volume 1*. Upper Saddle River, NJ: Prentice Hall, 2000.

[Ste96] Stevens, W. *TCP/IP Illustrated, Volume 3*. Upper Saddle River, NJ: Prentice Hall, 2000.

[Ste99] Stewart III, J. *BGP4: Inter-Domain Routing in the Internet*. Reading, MA: Addison-Wesley, 1999.

[Sti02] Stinson, D. *Cryptography*. New York, NY: Chapman & Hall/CRC, 2002.

[Sub01] Subramanian, M. *Network Management*. Reading, MA: Addison-Wesley, 2000.

[SWF99] Scott, C., Wolfe, P. and Erwin, M. *Virtual Private Networks*. Sebastopol, CA: O'Reilly, 1998.

[SX02] Stewart, R. and Xie, Q. *Stream Control Transmission Protocol (SCTP)*. Reading, MA: Addison-Wesley, 2002.

[Tan03] Tanenbaum, A. *Computer Networks*. Upper Saddle River, NJ: Prentice Hall, 2003.

[Tho00] Thomas, S. *SSL and TLS Essentials*. New York, NY: Wiley, 2000.

[WV00] Warland, J. and Varaiya, P. *High Performance Communication Networks*. San Francisco, CA: Morgan, Kaufmans, 2000.

[WZ01] Wittmann, R. and Zitterbart, M. *Multicast Communication*. San Francisco, CA: Morgan, Kaufmans, 2001.

[YS01] Yuan R. and Strayer, W. *Virtual Private Network*. Reading, MA: Addison-Wesley, 2001.

[Zar02] Zaragoza, R. *The Art of Error Correcting Coding*. Reading, MA: Addison-Wesley, 2002.

缩 略 词

AAL application adaptation layer 应用适配层
ABM asynchronous balanced mode 异步平衡模式
ACK acknowledgment 确认
ACL asynchronous connectionless link 异步无连接链路
ADSL asymmetric digital subscriber line 非对称数字用户线路
AES Advanced Encryption Standard 高级加密标准
AH Authentication Header 鉴别头部
AM amplitude modulation 调幅
AMI alternate mark inversion 交替传号反转
AMPS Advanced Mobile Phone System 高级移动电话系统
ANSI American National Standards Institute 美国国家标准协会
AP access point 访问点
ARP address resolution protocol 地址解析协议
ARPA Advanced Research Projects Agency （美国国防部）高级研究计划局
ARPANET Advanced Research Projects Agency Network （美国国防部）高级研究计划局网络
ARQ automatic repeat request 自动重发请求
AS autonomous system or authentication server 自治系统或鉴别服务器
ASCII American Standard Code for Information Interchange 美国信息交换标准代码
ASK amplitude shift keying 幅移键控
ASN.1 Abstract Syntax Notation 1 抽象语法表示法1
ATM Asynchronous Transfer Mode 异步传输模式
AUI attachment unit interface 附属装置接口
B8ZS bipolar with 8 zero substitution 8零置换的双极编码
Be committed burst size 提交突发业务量
Be excess burst size 超突发长度
BECN backward explicit congestion notification 后向显式拥塞通知
BER Basic Encoding Rules 基本编码规则
BGP Border Gateway Protocol 边界网关协议
BNC Bayonet-Neill-Concelman BNC连接器
BOOTP Bootstrap Protocol 引导协议
BSS basic service set 基本服务集
BUS broadcast/unknown server 广播和未知服务器
CA Certification Authority 认证中心
CATV community antenna TV 公共天线电视
CBC cipher block chaining mode 加密分组链模式
CBR constant bit rate 恒定比特率
CBT Core Based Tree 基于核的树
CCITT Consultative Committee for International Telegraphy and Telephony 国际电话电报咨询委员会
CCK complementary code keying 补码键控
CDMA code division multiple access 码分多路访问
CFB cipher feedback mode 加密反馈模式
CGI Common Gateway Interface 公共网关接口
CHAP Challenge Handshake Authentication Protocol 查询握手鉴别协议
CIDR Classless Inter-Domain Routing 无类域间路由选择
CIR committed information rate 提交信息速率
CLEC competitive local exchange carrier 竞争本地交换电信公司
CMTS cable modem transmission system 有线调制解调器传输系统
CRC cyclic redundancy check 循环冗余检验
CS convergence sublayer 会聚子层
CSM cipher stream mode 加密流模式
CSMA carrier sense multiple access 载波检测多路访问
CSMA/CA carrier sense multiple access with collision avoidance 带冲突避免的载波检测多路访问
CSMA/CD carrier sense multiple access with collision detection 带冲突检测的载波检测多路访问
D-AMPS digital AMPS 高级数字移动电话系统
DARPA Defense Advanced Research Projects Agency （美国）国防部高级研究计划局
dB decibel 分贝
DC direct current 直流
DCF distributed coordination function 分布式协调功能
DCT discrete cosine transform 离散余弦变换
DDNS Dynamic Domain Name System 动态域名系统
DDS digital data service 数字数据业务
DE discard eligibility 可丢弃选择位
DEMUX demultiplexer 分离器
DES data encryption standard 数据加密标准

DHCP Dynamic Host Configuration Protocol 动态主机配置协议
DIFS distributed interframe space 分布式帧间间隔
DLCI data link connection identifier 数据链路连接标识符
DMT discrete multitone technique 离散多音频技术
DNS Domain Name System 域名系统
DOCSIS Data Over Cable System Interface Specifications 有线数据系统接口规范
DS digital signal 数字信号
DS Differentiated Services 区分业务
DSL digital subscriber line 数字用户线路
DSLAM digital subscriber line access multiplexer 数字用户线路接入复用器
DSSS direct sequence spread spectrum 直接序列扩频
DVMRP Distance Vector Multicast Routing Protocol 距离向量多播路由选择协议
DWDM dense wave division multiplexing 密集波分复用
EBCDIC extended binary coded decimal interchange code 扩充的二-十进制交换码
EIA Electronics Industries Association 电子工业协会
ESP Encapsulating Security Payload 封装安全载荷
ESS Extended Service Set 扩展服务集
FCC Federal Communications Commission （美国）联邦通信委员会
FDM frequency-division multiplexing 频分调制
FDMA frequency division multiple access 频分多路访问
FECN forward explicit congestion notification 前向显式拥塞通知
FHSS frequency hopping spread spectrum 跳频扩频
FIFO first-in, first-out 先进先出
FM frequency modulation 调频
FQDN fully qualified domain name 全称域名
FRAD Frame Relay assembler/disassembler 帧中继组装器/分解器
FSK frequency shift keying 频移键控
FTP File Transfer Protocol 文件传输协议
GPS Global Positioning System 全球定位系统
GSM Global System for Mobile Communication 全球移动通信系统
HDLC High-level Data Link Control 高级数据链路控制
HDSL high bit rate digital subscriber line 高速数字用户线路
HFC hybrid-fiber-coaxial network 光纤同轴电缆混合网络
HMAC hashed-message authentication code 散列报文鉴别码
HR-DSSS High Rate Direct Sequence Spread Spectrum 高速率直接序列扩频
HTML Hypertext Markup Language 超文本标记语言
HTTP HyperText Transfer Protocol 超文本传输协议
Hz hertz 赫兹
IAB Internet Architecture Board 因特网体系结构委员会
IANA Internet Assigned Numbers Authority 因特网号码分配管理局
ICANN Internet Corporation for Assigned Names and Numbers 因特网名称和数字分配组织
ICMP Internet Control Message Protocol 因特网控制报文协议
ICMPv6 Internet Control Message Protocol, version 6 因特网控制报文协议版本6
IEEE Institute of Electrical and Electronics Engineers 电气和电子工程师协会
IESG Internet Engineering Steering Group 因特网工程指导组
IETF Internet Engineering Task Force 因特网工程任务组
IFS interframe space 帧间间隔
IGMP Internet Group Management Protocol 因特网组管理协议
IKE Internet Key Exchange 因特网密钥交换
ILEC incumbent local exchange carrier 原本地交换电信公司
IMAP4 Internet Mail Access Protocol, version 4 因特网邮件访问协议版本4
INTERNIC Internet Network Information Center 因特网网络信息中心
IntServ Integrated Services 综合业务
IP Internet Protocol 因特网协议
IPCP Internetwork Protocol Control Protocol 互联网络协议控制协议
IPng Internet Protocol next generation 下一代因特网协议
IPSec IP Security IP层安全
IPv4 Internet Protocol, version 4 因特网协议版本4
IPv6 Internet Protocol, version 6 因特网协议版本6
IRTF Internet Research Task Force 因特网研究任务组
IS-95 Interim Standard 95 过渡标准95
ISAKMP Internet Security Association and Key Management Protocol 因特网安全联盟和密钥管理协议

ISO International Organization of Standardization 国际标准化组织
ISOC Internet Society 因特网协会
ISP Internet service provider 因特网服务提供商
ISUP ISDN user port ISDN用户端口
ITM-2000 International Mobile Communication 2000年国际移动通信
ITU-T International Telecommunications Union-Telecommunication Standardization Sector 国际电信联盟—电信标准分会
IXC interexchange carrier 跨区交换电信公司（长途电话公司）
JPEG Joint Photographic Experts Group 联合图像专家组
KDC key distribution center 密钥分发中心
L2CAP Logical Link Control and Adaptation Protocol 逻辑链路控制和适配协议
LAN local area network 局域网
LANE LAN emulation 局域网仿真
LANE local area network emulation
LATA local access and transport area 本地接入和传送区域
LCP Link Control Protocol 链路控制协议
LEC LAN emulation client 局域网仿真客户机
LEC local exchange carrier 本地交换电信公司
LEO low earth orbit 近地轨道
LES LAN emulation server 局域网仿真服务器
LLC logical link control 逻辑链路控制
LMI local management information 本地管理信息
LSA link state advertisement 链路状态通告
LSP link state packet 链路状态分组
MA multiple access 多路访问
MAA message access agent 报文接访问代理
MAC medium access control sublayer 介质访问控制子层
MAC message authentication code 报文鉴别码
MAN metropolitan area network 城域网
MBONE multicast backbone 多播主干
MDC modification detection code 修改检测代码
MEO medium Earth orbit 中地轨道
MIB Management Information Base 管理信息数据库
MII medium independent interface 独立介质接口
MIME Multipurpose Internet Mail Extension 多用途因特网邮件扩展
MLT-3 multitime transmission, 3-level 3电平多线路传输编码
MOSPT Multicast Open Shortest Path First 多播开放最短路径优先
MPEG motion picture experts group 运动图像专家组
MSC mobile switching center 移动交换中心
MTA mail transfer agent 邮件传输代理
MTA message transfer agent 报文传输代理
MTSO mobile telephone switching office 移动电话交换局
MTU maximum transfer unit 最大传输单元
MUX multiplexer 复用器
NAP network access point 网络访问点
NAT network address translation 网络地址转换
NAV network allocation vector 网络分配向量
NCP Network Control Protocol 网络控制协议
NIC network interface card 网络接口卡
NNI network-to-network interface 网络到网络接口
NRM normal response mode 正常响应方式
NRZ nonreturn to zero 不归零编码
NRZ-I nonreturn to zero, invert 不归零反向编码
NRZ-L nonreturn to zero, level 不归零电平编码
NVT Network Virtual Terminal 网络虚拟终端
OC optical carrier 光载波
OFB output feedback 输出反馈
OFDM orthogonal frequency division multiplexing 正交频分多路复用
OSI Open System Interconnection 开放系统互联
OSPF open shortest path first 开放最短路径优先
PAM pulse amplitude modulation 脉冲振幅调制
PAP Password Authentication Protocol 口令鉴别协议
PCF point coordination function 点协调功能
PCM pulse code modulation 脉冲码调制
PCS personal communication system 个人通信系统
PGP Pretty Good Privacy 相当好的保密性（PGP协议）
PHB per hop behavior 逐跳行为
PIM Protocol Independent Multicast 协议独立多播
PIM-DM Protocol Independent Multicast, Dense Mode PIM密集模式
PIM-SM Protocol Independent Multicast, Sparse Mode PIM稀疏模式
PKI public key infrastructure 公钥基础设施
PM phase modulation 相位调制
PN pseudorandom noise 伪随机噪声
POP point of presence 服务接入点
POP3 Post Office Protocol, version 3 邮局协议版本3
POTS plain old telephone system 普通旧式电话系统
PPP Point-to-Point Protocol 点到点协议
PQDN partially qualified domain name 部分域名

PSK phase shift keying 相移键控
PVC permanent virtual circuit 永久虚电路
QAM quadrature amplitude modulation 正交调幅
QoS quality of service 服务质量
RADSL rate adaptive asymmetrical digital subscriber line 速率自适应非对称数字用户线路
RARP Reverse Address Resolution Protocol 反向地址解析协议
RFC Request for Comment 请求评论
RIP Routing Information Protocol 路由选择信息协议
ROM read only memory 只读存储器
RPB reverse path broadcasting 逆路径广播
RPF reverse path forwarding 逆路径转发
RPM reverse path multicasting 逆路径多播
RSA Rivest, Shamir, Adleman RSA算法
RSVP Resource Reservation Protocol 资源预留协议
RTCP Real-time Transport Control Protocol 实时传输控制协议
RTP Real-time Transport Protocol 实时传输协议
RTSP Real-Time Streaming Protocol 实时流协议
RTT round-trip time 往返时间
RZ return to zero 归零
SA Security Association 安全关联
SADB security association database 安全关联数据库
SAR segmentation and reassembly 分割与重组
SCCP signaling connection control point 信令连接控制点
SCO synchronous connection oriented 同步面向连接
SCP server control point 业务控制点
SCTP Stream Control Transmission Protocol 流控制传输协议
SDH Synchronous Digital Hierarchy 同步数字系列
SDSL symmetric digital subscriber line 对称数字用户线路
SEAL simple and efficient adaptation layer 简单有效适配层
SHA-1 Secure Hash Algorithm 1 安全散列算法1
SIFS short interframe space 短帧间间隔
SIP Session Initiation Protocol 会话发起协议
SMI Structure of Management Information 管理信息结构
SMTP Simple Mail Transfer Protocol 简单邮件传输协议
SNMP Simple Network Management Protocol 简单网络管理协议
SNR signal-to-noise ratio 信噪比
SONET Synchronous Optical Network 同步光纤网络
SP signal point 信号点
SPE synchronous payload envelope 同步有效载荷净荷
SPI security parameter index 安全参数索引
SS7 Signaling System Seven 7号信令系统
SSL Secure Socket Layer 安全套接字层
STM synchronous transport module 同步传输模块
STP shielded twisted pair 屏蔽双绞线
STP signal transport port 信号传输端口
STS synchronous transport signal 同步传输信号
SVC switched virtual circuit 交换虚电路
TCAP transaction capabilities application port 事务处理能力应用端口
TCP Transmission Control Protocol 传输控制协议
TCP/IP Transmission Control Protocol/Internetworking Protocol 传输控制协议/网际协议
TDD-TDMA time division duplexing TDMA 时分双工TDMA
TDM time-division multiplexing 时分复用
TDMA time division multiple access 时分多路访问
TELNET Terminal Network 终端网络
TFTP Trivial File Transfer Protocol 简单文件传输协议
TGS ticket granting server 票证授予服务器
TLS Transport Layer Security 传输层安全
TOS type of service 服务类型
TSI time-slot interchange 时隙交换
TTL time to live 生存时间
TUP telephone user port 电话用户端口
UA user agent 用户代理
UBR unspecified bit rate 未指定比特率
UDP User Datagram Protocol 用户数据报协议
UNI user network interface 用户网络接口
UNI user-to-network interface
URL Uniform Resource Locator 统一资源定位符
UTP unshielded twisted pair 无屏蔽双绞线
VBR variable bit rate 可变比特率
VC virtual circuit 虚电路
VDSL very high bit rate digital subscriber line 甚高速数字用户线路
VLAN virtual local area network 虚拟局域网
VOFR Voice Over Frame Relay 帧中继上传音
VPN virtual private network 虚拟专用网
VT virtual tributary 虚拟支流
WAN wide area network 广域网
WATS wide area telephone service 广域电话业务
WDM wave-division multiplexing 波分复用
WWW World Wide Web 万维网

术 语 表

1000Base-CX　千兆位以太网双线的实现，该实现使用屏蔽双绞线电缆。

1000Base-LX　千兆位以太网双线的实现，该实现使用传输长波激光信号的光缆。

1000Base-SX　千兆位以太网双线的实现，该实现使用传输短波激光信号的光缆。

1000Base-T　千兆位以太网四线的实现，该实现使用无屏蔽双绞线电缆。

100Base-FX　快速以太网双线的实现，该实现使用光缆。

100Base-T4　快速以太网四线的实现，该实现使用无屏蔽双绞线电缆。

100Base-TX　快速以太网双线的实现，该实现使用无屏蔽双绞线电缆。

10Base 2　细缆以太网标准。

10Base 5　粗缆以太网标准。

10Base-F　光纤以太网标准。

10Base-T　双绞线以太网标准。

10GBase-E　10千兆位以太网扩展的实现。

10GBase-L　10千兆位以太网的实现，该实现使用传输长波激光的光缆。

10GBase-S　10千兆位以太网的实现，该实现使用传输短波激光的光缆。

1-persistent strategy（1-持续策略）CSMA持续性策略规定，如果线路空闲就立即发送帧。

2B1Q encoding（2B1Q编码）一种线路编码技术，每一个脉冲表示2位。

4B/5B encoding（4B/5B编码）一种分组编码技术，4位被编码为5位代码。

8B/10B encoding（8B/10B编码）一种分组编码技术，8位被编码为10位代码。

8B6T encoding（8B/6T编码）一种三级线路编码策略，8位被编码为6个三级脉冲代码。

4-dimensional, 5-level pulse amplitude modulation（4D-PAM5，4维5级脉冲振幅调制）1000Base-T使用的编码策略。

56K modem（56K调制解调器）使用两种不同速率的调制解调技术的调制解调器；一种速率用于上行到因特网，另一种速率用于从因特网上下载。

A

Abstract Syntax Notation 1（ASN.1，抽象语法记法1）它是基于简单结构化数据的一种标准。

access point（AP，访问点）BSS中的中央基站点。

acknowledgment（ACK，确认）由接收方发送的响应，表明数据已经成功收到。

active document（活动文档）在万维网中，使用Java语言在本地运行的文档。

adaptive delta modulation（自适应delta调制）一种调制技术，其中delta值随模拟信号的振幅而变化。

add/drop multiplexer（分插复用器）一种SONET设备，它将不同源的信号进行多路复用或者将一个信号分离到多个目的地。

additive increase（加性增加）带有慢速启动的避免拥塞策略，在这一策略中，窗口的大小每次增加一段而不是按指数形式增加。

address aggregation（地址聚合）将几个地址块组成较大地址块的机制。

Address Resolution Protocol（ARP，地址解析协议）在TCP/IP协议中，从一个节点的已知因特网的地址，得出其物理地址的协议。

address space（地址空间）由一个协议使用的地址总数。

ADSL Lite（ADSL Lite）一种不需要话音分离器的ADSL的技术，这种技术允许电话机的插头直接插入ADSL调制解调器直连接计算机，话音分离工作在电话局中完成。

Advanced Encryption Standard（AES，高级加密标准）NIST采纳的代替DES的秘密密钥加密系统。

Advanced Mobile Phone System（AMPS，高级移动电话系统）北美使用的FDMA模拟蜂窝电话系统。

Advanced Research Projects Agency（ARPA，美国国防部高级研究计划局）由资助建造ARPANET的政府机构。

Advanced Research Projects Agency Network（ARPANET，美国国防部高级研究计划局网络）由ARPA出资的分组交换网络。

ALOHA（ALOHA）最初的超机多路访问方法，一个站点只要它有要发送的数据，就可以在任何时刻发送。

alternate mark inversion（AMI，交替信号反转）一种数字到数字的双极性编码方法，在这种方法中，代表1的编码策略在正负电平之间交替。

American National Standards Institute（ANSI，美国国家标准协会）在美国定义标准的国家标准化组织。

American Standard Code for Information Interchange（ASCII，美国信息交换标准代码）由美国国家标准协会开发的一种字符代码，广泛用于数据通信。

amplitude（振幅）信号的强度，通常用伏特度量。

amplitude modulation（AM，调幅）一种模拟到模拟的转换方法，载波信号的振幅随着调制信号的振幅变化。

amplitude shift keying（ASK，幅移键控）一种调制方法，改变载波信号的振幅来表示二进制的0或1。

analog data（模拟数据）连续平滑的数据，对值的数量没有限制。

analog signal（模拟信号）随时间平滑变化的连续信号波形。

analog-to-analog conversion（模拟-模拟转换）用模拟信号表示模拟信息。

analog-to-digital conversion（模拟-数字转换）用数字信号表示模拟信息。

angle of incidence（入射角）在光学中，光线通过两种介质界面时，入射光线与界面的法线所形成的角。

anycast address（任播地址）一种地址，它定义一组计算机具有相同前缀的地址。

aperiodic signal（非周期性信号）一种没有重复模式的信号。

applet（Java小应用程序）用来创建万维网活动文档的应用程序，通常用Java语言编写。

application adaptation layer（AAL，应用适配层）ATM协议的一层，它将用户数据分割成48字节的有效载荷。

application layer（应用层）因特网模型中的第五层，该层提供对网络资源的访问。

area（区域）包含在自治系统内部的网络、主机和路由器的集合。

association（关联）流控制传输协议（SCTP）中的连接。

asymmetric digital subscriber line（ADSL，非对称数字用户线路）一种下行数据速率高于上行数据速率的通信技术。

asynchronous balanced mode（ABM，异步平衡模式）在HDLC中的一种通信模式，此模式中所有的站点都是平等的。

asynchronous connectionless link（ACL，异步无连接链路）蓝牙主站点和从站点之间的一种链路，此链路中，被破坏的有效载荷需要重传。

Asynchronous Transfer Mode（ATM，异步传输模式）一种广域协议，其特点是将数据速率和等长的分组（信元）。ATM适合于传送文本、音频和视频数据。

asynchronous transmission（异步传输）一种数据传输类型，它使用起始位和停止位，并且在数据单元之间有可变的时间间隔。

ATM LAN（ATM局域网）使用ATM技术的局域网。

ATM layer（ATM层）ATM参考模型中的一层，该层提供路由、流量管理、交换和复用服务。

attachment unit interface（AUI，附属装置接口）一种10Base5的电缆，在站点与收发器之间执行物理接口功能。

attenuation（衰减）由于介质的阻抗而使得信号能量的丢失。

audio（音频）声音或音乐的记录或传送。

authentication（鉴别）对报文发送方的验证。

Authentication Header（AH）Protocol（鉴别头部协议）IPSec在网络层定义的协议，通过中数列函数创建的数字签名来提供信息的完整性。

authentication server（AS，鉴别服务器）即Kerberos协议中的KDC。

automatic repeat request（ARQ，自动重复请求）一种差错控制方法，用重传数据来纠错。

autonegotiation（自动协商）快速以太网的特性，容许两种设备协商它们的通信模式或速率。

autonomous system（AS，自治系统）一个单独的行政单位管辖下的一组网络和路由器。

B

backward explicit congestion notification（BECN，后向显式拥塞通知）帧中继分组的一位，用于通知拥塞的发送方。

band-pass channel（带通通道）可以通过某一频率范围的通道。

bandwidth（带宽）复合信号中最高频率和最低频率的差值。也可以用来度量一条线路或网络的信息承载能力。

bandwidth on demand（按需分配带宽）一种数字服务，容许用户通过使用多条线路达到更高的速度。

bandwidth-delay product（带宽延迟乘积）在等待来自发送方的信息时，能够发送的位数的度量。

banyan swith（banyan交换机）一种多级交换机，每一级都有许多微交换机，它基于输出端口路由以二进制串表示的分组。

Barker sequence（巴克序列）11位二进制扩展序列。

baseband transmission（基带传输）利用非调制通道传输没有调制的数字或模拟信号。

base header（基本头部）IPv6中数据报的主要头部。

baseline wandering（基线漂移）在数字信号编码中，接收器计算接收到的信号功率的平均值。这个平均值称为基线。由0和1组成的序列可能会偏移基线（基线漂移）给接收器正确译码带来问题。

Basic Encoding Rule（BER，基本编码规则）表示网络传输编码数据的标准。

Basic Latin（基本拉丁字符集）ASCII字符集。

basic service set（BSS，基本服务集）由IEEE 802.11标准定义的无线局域网的构建部分。

Batcher-banyan switch（巴切尔—榕树交换机）根据目的端口对到达的分组进行排序的榕树交换机。

baud rate（波特率）每秒传输的信号元素个数。每个信号元素包括一个或多个位。

Bayone-Neill-Concelman（BNC）connector（BNC连接器）一种常用的同轴电缆连接器。

bidirectional authentication（双向鉴别）一种鉴方法，包括从发送方到接收方的询问和应答，反之亦然。

bidirectional frame（B-frame，双向帧）一种MPEG帧，它与前后的I帧和P帧有关。

binary exponential backup（二进制指数回退）在竞争存取方法中，系统对延迟访问所用的一种试退重传算法策略。

binary notation（二进制标记法）IP地址的二进制表示。

biphase（双相编码）双极型编码的一种类型，信号在每个位中间的跳变。Manchester编码和差分Manchester是双相编码的例子。

bipolar encoding（双极型编码）一种数字到数字的编码方法，振幅0表示二进制的0，正的和负的振幅表示交替出现的1。

bipolar with 8 zero substitution（B8ZS，8零置换的双极型编码）一种启动的技术，为了改善位同步，它对8个0的比特串，根据前导1的极性用已定义的模式替代。

bit（位，比特）一个二进制数字，是信息的最小单元：1或0。

bit rate（比特率）每秒钟传输的位的个数。

bit stuffing（位填充）在面向位的协议中，在帧的数据部分中插入额外的位以防止接收方发生将数据误认为是标志的错误。

bit-oriented protocol（面向位的协议）在这个协议中，数据帧被解释为位的序列。

block cipher（分组加密）一种加密和解密算法，将分组作为它的基本单元。

block code（块代码）将数据划分为数据字，每个数据字的加冗余位形成码字和纠错编码。

block coding（块编码）一种保证同步和检错的编码方法。

blocking（阻塞）当交换网络工作在它的最大容量不能接受更多的输入而发生的事件。

Bluetooth（蓝牙）一种无线局域网技术，它使用双在一个（室内房间）小范围内连接不同功能的设备，这些设备可以是电话或笔记本电脑等。

Bootstrap Protocol（BOOTP，引导程序协议）通过一个表（文件）来失配置信息的协议。

Border Gateway Protocol（BGP，边界网关协议）基于路径向量路由选择的自治系统间的路由选择协议。

bridge（网桥）在网络模型前两层运行的一个网络设备，它具有过滤和转发的能力。

broadband transmission（宽带传输）用高频信号调制的传输信号，它将来自各种源的数据组合在一起。

broadcast address（广播地址）为了定义发送到网络中所有节点的地址。

broadcast/unknown server（BUS，广播和未知服务器）连接ATM交换机与多播和广播功能的服务器。

broadcasting（广播）向网络中所有节点发送报文。

browser（浏览器）显示WWW文档的应用程序。浏览器通常使用其他因特网服务来访问文档。

BSS-transition mobility（BSS传输移动性）在无线局域网中，能够从一个BSS移动到另一个BSS。但标准定在一个ESS内部的站点间。

bucket brigade attack（水桶队列攻击）参看中间人攻击。

burst error（突发性差错）在数据单元中发生差错，其中有两个比特或更多比特被改变。

bursty data（突发性数据）具有可变的瞬时传输速率的数据。

bus topology（总线拓扑结构）一种网络拓扑结构，其中所有计算机都连接到一种介质上。

byte stuffing（字节填充）在面向字节的协议中，在域的数据部分插入额外的字节来防止接收方发生将数据误认为是标志的错误。

byte oriented protocol（面向字节协议）在这个协议中，将帧的数据解释为字节（字对）的序列。

C

cable modem（有线调制解调器）有线电视电缆提供互联网访问的一种技术。

cable modem transmission system（CMTS，有线调制解调器传输系统）安装在公布式集线器内部的一种设

备，从图像网络收数据并将数据传送到组合器。

cable TV network（有线电视网络）使用同轴电缆或光缆的系统，可以将多路的视频节目信号传送到用户家中。

caching（高速缓存）信息存储在小的高速的存储器中。

Caesar cipher（凯撒加密算法）Julius Caesar所用的移位加密算法，密钥是3。

carrier extension（载波伸延）千兆以太网的一种技术，增加最小帧的长度从而使一次传输的帧长度的时间超过传播时间。

carrier sense multiple access（CSMA，载波侦听多路访问）一种竞争访问方法，在这种方法中，每一个站点在发送数据之前先侦听线路。

carrier sense multiple access with collision avoidance（CSMA/CA，带冲突避免的载波侦听多路访问）一种访问方法，在这种方法中会避免冲突。

carrier sense multiple access with collision detection（CSMA/CD，带冲突检测的载波侦听多路访问）一种访问方法，在这种方法中，只要介质空闲时，站点就发送数据，并且当冲突出现时就进行重传。

carrier signal（载波信号）一种高频信号，用于数字到模拟或模拟到数字的调制，即载波信号的某个特征（振幅、频率或相位）随着调制数据变化而变化。

cell（信元）一种固定长度的小数据单元，在移动电话中，它指的是由一个交换局所服务的地理区域。

cell network（信元网络）一种将信元作为其基本数据单元的网络。

cellular telephony（移动电话）一种无线通信技术，一个地理区域被划分成一些小的单元，一个小的单元由一个传输器提供服务。

certification Authority（CA，认证中心）一个联邦政府级组织，该机构将公钥与实体绑定并发布证书。

challenge handshake Authentication Protocol（CHAP，查询握手鉴别协议）在PPP中，用于鉴别的三次握手协议。

channel（通道）一条通信通路。

channelization（通道化）一种多路访问方法，链路的可用带宽在时间上共享。

character oriented protocol（面向字符协议）参见面向字节协议。

checksum（校验和）用于差错检测的一个值，该值是使用反码运算将数据相加然后将结果取反。

chip（芯片）CDMA中，赋予站点的代码中的一个数字。

choke point（抑制点）由路由器发送到源晶站点的分组，通知拥塞的发生。

chunk（大块）SCTP中传输的单元。

cipher（密码）加密/解密算法。

cipher block chaining（CBC）mode（加密分组链模式）一种DES和三重DES操作模式，其中一个分组的加密（或解密）依赖于所有先前的分组。

cipher feedback mode（CFB，加密反馈模式）一种DES和三重DES操作模式，其中每一次只发送和接收一位数据，并且每一位都独立于前面的位。

cipher stream mode（CSM，加密流模式）一种DES和三重DES操作模式，其中一次发送和接收一个字节。

cipher suite（密码组）一组可能的密码。

ciphertext（密文）加密过的数据。

circuit switching（电路交换）一种交换技术，站点之间使用专用通路建立连接。

cladding（包层）纯线光缆的纤芯的玻璃或塑料，包层的光密度必须小于纤芯的光密度。

classful addressing（分类寻址）IPv4寻址机制，在这里，IP地址空间被分为五类：A、B、C、D和E，每一类占据IP地址空间的一部分。

classless addressing（无类寻址）一种寻址机制，在这里，IP地址空间不分类。

Classless InterDomain Routing（CIDR，无类域间路由选择）在使用超级网络时，一种减少路由表记录数量的技术。

client process（客户端进程）本地运行的应用程序，它向在远程站点上运行的应用程序请求服务。

client-server model（客户-服务器模型）在两个应用程序间的交互模型，即在一端的一个程序（客户）向在另一端的一个称为服务器的程序请求服务。

closed-loop congestion control（闭环拥塞控制）在拥塞发生后，一种减轻拥塞的方法。

coaxial cable（同轴电缆）一种传输介质，包含导电芯、绝缘材料和另一个导电包层。

code division multiple access（CDMA，码分多路访问）一种多路访问方法，在这种方法中，一个通道同时容纳所有的传输。

codeword（码字）已编码的数据字。

ColdFusion 一种动态Web技术，它允许传统数据库的数据项合并。

collision（冲突）两个发送方在一个通道中同时发送时所出现的事件，并且该通道同时只能容许有一个发送方发送，这时数据将会被破坏。

collision domain（冲突域）受冲突影响的介质长度。

committed burst size（Bc，提交突发业务量）在预定时间段内，城中新网络没有丢弃任何帧能传输的最大位数。

committed information rate（CIR，提交信息速率）提交突发业务量除以时间。

common carrier（公用电信公司）一个可以为公众使用的传输机构，服从于公用事业管理。

Common Gateway Interface（CGI，公共网关接口）一种在HTTP服务器和执行程序（文件）之间的通信标准。CGI用于创建动态文档。

community antenna TV（CATV，公共天线电视）一种有线网络服务，它将视频信号广播到并以接收或不能接收的地区。

compatible address（兼容地址）由96个0后接IPv4的32位地址组成的IPv6地址。

competitive local exchange carrier（CLEC，竞争本地交换电信公司）一种电话公司，它不能提供主要电话服务，而只能提供诸如移动电话服务以及在一个LATA内提供分费电话服务。

complementary code keying（CCK，补码键控）HR-DSSS的一种编码方法，将4位或8位的数编码成一个符号。

composite signal（复合信号）由多个正弦波信号组成的信号。

congestion（阻塞）网络或网络间出现过多通信量，因而引起服务质量的全面下降。

connecting device（连接设备）连接计算机或网络的工具。

connection establishment（连接建立）在真正的数据传输之前，建立一条逻辑连接建立所必须要先做的工作。

connectionless service（无连接服务）不建立连接也不终止连接的数据传送服务。

constant bit rate（CBR，恒定比特率）ATM服务类的数据速率，为需要实时传输视频或音频服务的用户设计。

constellation diagram（星座图）在数据转换调制中，不同位组合的相位和振幅的图形表示。

Consultative Committee for International Telegraphy and Telephony（CCITT，国际电话电报咨询委员会）一个ITU-T的国际标准化组织，现改名为ITU-T。

contention（竞争）一种访问方法，在这种方法中，在同一通道上有两个或多个设备在同一时刻都试图传输数据。

controlled access（受控访问）一种多路访问方法，在这种方法中，站点与其他站点协商以决定是否可以发送。

convergence sublayer（CS，会聚子层）在ATM协议中，AAL子层中的高层，它在用户数据中增加一个头部和尾部。

cookie 一个字符串，用来返回给用户服务器的一段信息。

core（纤芯）光纤的玻璃或塑料中心。

Core-Based Tree（CBT，基于核的树）在多播中的一个组共享协议，它使用一个中央路由器作为树的根。

country domain（国家域）规范系统中的一个子域，使用两个字符作为后缀。

crosshar switch（纵横开关）一个包括格状水平和垂直路径的开关，在每一个纵横路径的交点，就有一个将输入连接到输出的交叉点。

crosspoint（交叉点）在纵横开关中，一个输入与一个输出的连接点。

crosstalk（串扰）在一条线路上由信号产生的噪声传到另一条线路上。

cryptography（密码学）将报文进行转换，使其安全免受攻击的科学和艺术。

cyclic code（循环码）一种线性编码，每一码字循环移位产生另一码字。

cyclic redundancy check（CRC，循环冗余校验）一种高精度的检错方法，它基于将一个由位组成的模式解释为一个多项式。

D

data element（数据单元）表示信息的最小单元，如一个位。

data encryption standard（DES，数据加密标准）美国国家标准加密方法，用于非军事和非机密领域。

data link connection identifier（DLCI，数据链路连接标识符）在城中继中标志虚电路的编号。

data link control（数据链路控制）数据链路层的职责，负责流量控制与差错控制。

data link layer（数据链路层）因特网模型的第二层，负责节点到节点的传输。

Data Over Cable System Interface Specifications（DOCSIS，有线数据系统接口规范）在HFC网络上数据传输的标准。

data rate（数据速率）一秒发送数据单元的个数。

data transfer phase（数据传输阶段）电路交换网或虚电路网中数据传输过程的中间阶段。

data transparency（数据透明性）能够发送任何位模式的数据能力，而不会误认为是控制位。

datagram（数据报）在分组交换中的一个独立的数据单元。

datagram network（数据报网络）分组交换网，其中分组是彼此独立的。

dataword（数据字）分组编码中的最小数据块。

de facto standard（事实标准）一个协议，虽然并没有被有关组织的团体批准，但却通过广泛使用被采纳的标准。

de jure standard（法定标准）一个协议已被官方认可的团体批准。

deadlock（死锁）一个任务不能继续进行，由于它要等待另一个永远不会发生事件的情形。

decibel（dB）（分贝）一个信号相对于另一个信号的强度的一种量度。

decryption（解密）从已加密的数据恢复成原始报文。

default mask（默认抱轩）不再进行子网划分的网络的掩码。

default routing（默认路由选择）一种路由选择方法，路由器接收所有的与路由表不匹配的分组。

Defense Advanced Research Projects Agency（DARPA，美国国防高级研究计划局）一个政府组织，曾用ARPA的名义资助ARPANET和其他。

delta modulation（delta调制）一种模拟到数字转换技术，其中数字信号值是本次信号与前次信号采样的差。

demodulation（解调）将模拟信号从承载信息信号中分离的过程。

demultiplexer（DEMUX，分离器）将复用信号分离成原始各组成信号的设备。

denial of service attack（拒绝服务攻击）一种攻击形式，站点有许多假冒者请求服务这使最终取消服务。

dense wave division multiplexing（DWDM，密集波分复用）一种WDM方法，该方法通过减少通道间的距离来实现多路复用大量的通道。

differential Manchester encoding（差分Manchester编码）一种数字到数字的极性编码方法，其特点是在每一位间隔的中间有一个跳变，以及在每一个1开始有一个反转。

Differentiated Services（DS 或Diffserv，差分业务）为IP设计的基于分类地址的QoS。

Diffie-Hellman protocol（Diffie Hellman协议）一种密钥管理协议，为双方提供一次会话密钥。

digest（摘要）文档的压缩版本。

digital AMPS（D AMPS，高级数字移动电话系统）第二代蜂窝电话系统，即高级移动电话系统的数字化版本。

digital data（数字数据）通过离散值或多离散条件代表的数据。

digital data service（DDS，数字数据业务）速率为64Kbps的模拟租用线的数字形式。

digital signal（数字信号）由有限数值组成的离散信号。

digital signal (DS) service（DSS，数字信号服务）一种电信业务，其特点是多种数字信号具有层次结构。

digital signature（数字签名）鉴别报文发送方的一种方法。

digital subscriber line（DSL，数字用户线路）一种利用现有的电信网络实现数据、语音、视频和多媒体高速传送的技术。

digital subscriber line access multiplexer（DSLAM，数字用户线路接入复用器）电话公司的站点设备，功能与ADSL调制解调器类似。

digital-to-analog conversion（数模转换）使用模拟信号表示数字信息。

digital-to-digital conversion（数字到数字转换）使用数字信号表示数字信息。

digitization（数字化）模拟信息到数字信息的转换。

Dijkstra's algorithm（Dijkstra算法）在链路状态路由选择中，用来确定到达其他路由器最短路径的一种算法。

direct current（DC，直流）具有固定数幅并且频率为0的信号。

direct delivery（直接传送）分组到达最终目标的主机与发送方在同一个物理网络中的传送。

direct sequence spread spectrum（DSSS，直接序列扩频）一种无线传输方法，在这种方法中发送方发送的每一位都用称为芯片代码的位序列来代替。

discard eligibility（DE，可丢弃选择位）一个位，该位确定一个分组在网络存在拥塞时是否可以丢弃。

discrete cosine transform（DCT，离散余弦变换）JPEG处理的一个阶段，该变换改变了64个值从而像素之间的相对关系得到保持，但却暴露出了冗余。

discrete multitone technique（DMT，离散多音频技术）一种结合了正交振幅调制（QAM）和频分多路复用基本功能的调制方法。

Distance Vector Multicast Routing Protocol（DVMRP，距离向量多播路由选择协议）一种基于距离向量路由选择的协议，它与IGMP一起处理多播路由选择。

distance vector routing（距离向量路由选择）一种路由

选择方法，每一个路由器向其各邻站发送它所能到达的网络以及到这些网络的距离表。

distortion（失真）一个信号由于噪声、衰减或者其他影响而产生的任何信号变化。

distributed coordination function（DCF，分布式协调功能）无线局域网的基本访问方法，站点与其他站点竞争访问通道。

distributed database（分布式数据库）信息存在多个地方。

distributed interframe space（DIFS，分布式帧间间隔）在无线局域网中，站点在发送控制帧之前的等待时间。

distributed processing（分布式处理）一种算路，即为网络提供的服务驻留在多个地点。

DNS server（DNS服务器）保存名字空间信息的计算机。

domain name（域名）在DNS中，用点（.）分隔的一系列标号。

domain name space（域名空间）组织命名空间的一种结构，在这种结构中，名字定义为一棵的置树，根节点位于顶部。

Domain Name System（DNS，域名系统）一种TCP/IP应用服务，它以用户友好方式将名字转换为IP地址。

dotted-decimal notation（点分十进制标记法）一种为了使IP地址更易于阅读而设计的标记法，每个字节转换为等值的十进制数，十进制数之间用小数点分隔。

downlink（下行链路）从卫星到地球基站的传输。

downloading（下载）从远程站点检索文件或者数据。

dynamic document（动态文档）一种Web文档，它上服务器站点上运行的CGI程序生成的文档。

Dynamic Domain Name System（DDNS，动态域名系统）一种动态更新DNS主控文件的方法。

Dynamic Host Configuration Protocol（DHCP，动态主机配置协议）对BOOTP的扩展，它可以动态地指派配置信息。

dynamic mapping（动态映射）协议中用于地址解析的一种技术。

dynamic routing（动态路由选择）路由选择表记录由路由协议自动更新的路由选择方式。

E

E lines（E线路）与T线路相当的欧洲标准。

electromagnetic spectrum（电磁频谱）电磁能量占用的频率范围。

electronic code block（ECB）mode（电子编码分组模式）一种DES和三重DES操作模式，在这种模式中将长报文划分成64位长度的分组，并将每个分组独立加密。

Electronics Industries Association（EIA，电子工业协会）推广与电子有关产品的组织。它制定了一些端口标准，比如EIA 232、EIA 449和EIA 530等。

Encapsulating Security Payload（ESP，封装安全载荷）IPSec定义的一种协议，它提供保密性、完整性和报文鉴别。

encapsulation（封装）一种技术，将一个协议的数据单元置于另一种协议的数据单元中的数据字段部分中。

encryption（加密）将报文转换为一种看不懂的形式，如果不进行解密是无法阅读的。

end office（本地交换局）本地环路的终节点所在的交换局。

end system（端系统）数据发送方和接收方。

ephemeral port number（临时端口号）客户机使用的一个端口号。

error control（差错控制）数据传输中差错的检测和处理。

Ethernet（以太网）使用CSMA/CD协同方法的一种局域网。

excess burst size（Be，超突发长度）在帧中继中，在预先定义的特定时间间隔内，用户可以发送的超过Be的最大位数。

Extended Service Set（ESS，扩展服务集）由两个或者多个BSS和AP组成的无线局域网服务，AP间IEEE 802.11标准中所定义的一样。

exterior routing（外部路由选择）两个自治系统之间的路由选择。

extranet（外部网）一种使用TCP/IP协议族的专用网络，允许外部用户进行授权访问。

F

Fast Ethernet（快速以太网）数据速率为100Mbps的以太网。

fast retransmission（快速重传）在TCP中，发送方接收到三个完全相同的确认，隐含着数据段丢失或损失，要重传该数据段。

Federal Communications Commission（FCC，美国联邦通信委员会）制定无线电、电视和有线通信规范的政府机构。

fiber-optic cable（光缆）一种高智商的传输介质，以光冰冲形式承载数据信号。由两部分构成，中心是纤细的玻璃或者塑料纤芯，称为核，包围着用的玻璃或者塑料制成的圆柱体称为包层。

File Transfer Protocol（FTP，文件传输协议）在TCP/IP

中，在两个站点之间传输文件的应用层协议。

filtering（过滤）网桥架出转发决策的处理过程。

finite state machine（有限状态机）只在有两个状态之间转换的机器。

firewall（防火墙）安装在组织机构内部网络与因特网之间的一种设备（通常是一台路由器），用来提供安全性。

first-in, first-out（FIFO）queue（先进先出队列）队列中第一个进入的项也是最先出队列的项。

flag（标记）为了区别不同的域，用一个二进制的模式或一个字符添加在域的开始和结束处。

flat name space（平面命名空间）没有层次结构的命名空间。

flooding（洪泛）网络报文的饱和。

flow control（流量控制）一种控制域（分组或报文）流速率的技术。

footprint（覆盖区）卫星在特定时间内能够覆盖的地球上的某个区域。

forward error correction（前向纠错）在接收方纠错。

forward explicit congestion notification（FECN，前向显式拥塞通知位）域中部分组的一个位，用于通知拥塞的目的端。

forwarding（转发）把分组送到目的地路由。

Fourier analysis（傅里叶分析）在已知时域表示的情况下，获得非周期性信号频谱的数学方法。

fragmentation（分段）根据协议的最大传输单元（MTU）将一个分组划分为一些更小的单元。

frame（帧）表示一块数据的一组位。

frame bursting（帧突发）千兆以太网CSMA/CD的一种技术，它把多个小帧逻辑地连接在一起组成一个较长的帧。

Frame Relay（帧中继）因特网插图的前两层定义的一种分组交换规范。它没有网络层，差错检验是在端到端的基础上完成的，而不是每一条链路上。

Frame Relay assembler/disassembler（FRAD，帧中继组装器/分解器）帧中继中使用的一种设备，用来处理支自其他协议的帧。

frequency（频率）周期信号每秒重复的次数。

frequency division multiple access（FDMA，频分多路访问）一种多路访问方法，在这种方法中，带宽被划分为多个通道。

frequency hopping spread spectrum（FHSS，跳频扩频）一种无线传输方法。在这种方法中，发送方在一段较短的时间内以一种载波频率传输，然后跳转到另一种载波频率上，再以相同的时间跳转，一直继续直到N次跳转后重复。

frequency modulation（FM，调频）一种模拟到模拟的调制方法，在这种方法中，载波信号的频率随着调制信号的振幅变化而变化。

frequency shift keying（FSK，频移键控）一种数字到模拟编码方法，该方法中载波信号的频率随二进制数字0和1变化。

frequency-division multiple access（FDMA，频分多路访问）一种访问方法的技术，其中多个源站使用数据通信频带中指定的带宽。

frequency-division multiplexing（FDM，频分多路复用）一组模拟信号组成单一的信号。

frequency-domain plot（频域图）某种信号的频率成分的图形表示。

full-duplex mode（全双工模式）通信可以双向同时进行的传输模式。

full-duplex switched Ethernet（全双工交换式以太网）在这种以太网中，每个站点与交换机连接使用两条分离的链路，可以同时发送和接收。

fully qualified domain name（FQDN，全称域名）从主机标号开始口调到根节点，包含所经过的每一层的标号构成的域名。

fundamental frequency（基础频率）复合信号中主要正弦波的频率。

G

gatekeeper（关守）在H.323标准中，局域网中作为注册用的一个服务器。

gateway（网关）一种设备，用于连接两个使用不同通信协议的网络。

generic domain（通用域）域名系统中使用通用后缀的一个子域。

geographical routing（地理路由选择）一种路由选择技术，在这种技术中，整个地址空间按照实际的地理分布划分成多个块。

Gigabit Ethernet（千兆以太网）具有1000Mbps数据速率的以太网。

Global Positioning System（GPS，全球定位系统）一个中地轨道（MEO）公用卫星系统，由24颗卫星构成，用于陆地和海洋导航。GPS不用于通信。

Global System for Mobile Communication（GSM，全球移动通信系统）欧洲使用的第二代移动电话系统。

Globalstar（全球星）一种近地轨道（LEO）卫星系统，由位于6个极轨道的48颗卫星构成，每个轨道容纳

8颗卫星。

Go-Back-N ARQ（回退N帧自动重发请求）一种差错控制方法，从出现差错以后，该帧以及所有后续该必须重发。

grafting（嫁接）多带消息的恢复。

ground propagation（地表传播）无线电波通过大气的最低部分（环绕地球表面）传播。

group-shared tree（组共享树）多播路由选择的一种特性，系统中每一个组都共享同一棵树。

guard band（防护频带）分隔两个信号的特宽。

guided media（有向介质）具有物理边界的传输介质。

H

H.323 国际电信联盟（ITU）设计的一个标准，允许位于公用电话网中的电话与连接到因特网的计算机（在H.323中称为终端）进行会话。

half-duplex mode（半双工模式）可以双向通信但两个方向不能同时进行的传输模式。

Hamming code（汉明编码）在数据单元附加冗余位用于检测和纠正位差错的方法。

Hamming distance（汉明距离）两个码字相应位不同的个数。

handoff（切换）当移动设备从一个蜂窝移动到另一个蜂窝时，切换到新的通道。

harmonics（谐波）数字信号的多个成分，每一种成分都具有不同的振幅、频率和相位。

hash function（散列函数）从可变长的报文中创建固定大小摘要的算法。

hashed-message authentication code（HMAC，散列报文鉴别码）一个无密钥散列函数的MAC，如SHA-1。

header（头部）附加在数据分组开始部分的控制信息。

hertz（Hz，赫兹）频率的度量单位。

hexadecimal color notation（十六进制冒号标记法）IPv6的一种地址标注法，由32位十六进制数构成，每四位数字之间用冒号分隔。

hierarchical routing（分层路由选择）一种路由选择技术，它将整个地址空间按照特定标准划分为多个层次。

high bit rate digital subscriber line（HDSL，高速数字用户线路）与T1线路类似的服务，工作距离可达3.6 km。

High Rate Direct Sequence Spread Spectrum（HR-DSSS，高速率直接序列扩频）一种与DSSS类似的信号生成方法，但编码方法不同（CCK）。

High-level Data Link Control（HDLC，高级数据链路控制）ISO定义的面向位数据链路协议。

hop count（跳数）一条路由上节点的数量，在路由选择算法中它是距离的一种度量。

hop-to-hop delivery（跳到跳传递）从一个节点到下一节点的帧传输。

horn antenna（喇叭天线）一种勺子形状的天线，用于陆地微波通信。

host（主机）网络中的一个站点或者节点。

hostid（主机号）IP地址的一部分用于识别一台主机。

host-specific routing（特定主机路由选择）一种路由选择方法，在路由表中给出主机的完整IP地址。

hub（复线器）星形拓扑结构的中心设备，在多个节点之间提供共享连接。

Huffman encoding（Huffman编码）一种统计压缩编码方法，它使用可变长编码对一组符号进行编码。

hybrid network（混合网络）具有专用的互联网络并且能够访问全球因特网的网络。

hybrid-fiber-coaxial（HFC）network（光纤同轴电缆混合网络）第二代线缆网络，使用光纤和同轴电缆。

hypertext（超文本）包含文本的信息，并且能够通过链接指示符连接到其他文档。

HyperText Markup Language（HTML，超文本标记语言）用于创建Web文档的内容和格式的计算机语言。它允许附加一些文本用于包含定义字体、布局、嵌入图形和超文本连接的代码。

HyperText Transfer Protocol（HTTP，超文本传输协议）检索Web文档的应用服务。

I

inband signaling（带内信令）控制数据和用户数据使用一个通道进行传输。

incumbent local exchange carrier（ILEC，责任本地交换电信公司）1996年以前提供服务的电话公司，拥有线缆系统。

indirect delivery（间接传递）源端的分组到目的端的传递是在不同的网络上进行。

infrared wave（红外波）频率范围在300GHz～400THz的波，通常用于短距离通信。

inner product（内积）两个序列对应元素相乘，然后将积求和。

Institute of Electrical and Electronics Engineers（IEEE，电气和电子工程师协会）由专业工程师组成的团体，下设许多专业的社团，并在各自专业领域内起草各种标准。

Integrated Services（IntServ，综合业务）为IP设计的基于流量的QoS模型。

interactive audio/video（交互式音频/视频）具有声音和图像的实时通信。

interautonomous system routing protocol（自治系统之间的路由选择协议）用于处理自治系统之间传输的协议。

interdomain routing（域间路由选择）自治系统之间的路由选择。

interexchange carrier（IXC，跨区交换电信公司）长途电话公司，在1996年电信法之前，在不同LATA（本地接入和传送区域）的客户之间提供通信服务。

interface（接口）两台设备之间的边界，它规定了连接的机械的、电子的和功能的特性。

interference（干扰）妨碍清晰连接收有效信号。

interframe space（IFS，帧间间隔）无线局域网中，为了控制通道访问两个帧之间的时间间隔。

Interim Standard 95（IS-95，过渡标准95）北美第二代蜂窝电话主要标准之一。

interior routing（内部路由选择）自治系统内部的路由选择。

interleaving（交替）在多路复用中，按规定顺序从每一台设备中获取特定数量的数据。

International Organization of Standardization（ISO，国际标准化组织）一个国际性组织，它制定许多主题的标准。

International Telecommunications Union-Telecommunication Standardization Sector（ITU-T，国际电信联盟—电信标准分会）一个标准化组织，以前称为CCITT（国际电报电话咨询委员会）。

internet（互联网）一些网络的集合，这些网络由一些路由器或网关的网际互联设备连接起来。

Internet（因特网）使用TCP/IP协议族的全球互联网络。

Internet address（因特网地址）在一个使用TCP/IP协议的互联网络上，定义一个主机的32位或者128位的网络层地址。

Internet Architecture Board（IAB，因特网体系结构委员会）因特网协会（ISOC）的技术咨询机构，管理TCP/IP协议族的继续开发。

Internet Assigned Numbers Authority（IANA，因特网号码分配管理局）美国政府支持的一个机构，在1998年10月以前负责因特网域名和地址的管理。

Internet Control Message Protocol（ICMP，因特网控制报文协议）TCP/IP协议族中用于处理差错和控制报文的协议。

Internet Control Message Protocol，version 6（ICMPv6，因特网控制报文协议版本6）IPv6中处理差错和控制报文的协议。

Internet Corporation for Assigned Names and Numbers（ICANN，因特网名称和编号分配组织）由国际委员会管理的私有的非盈利性组织，是按IANA的方式运作的。

Internet draft（因特网草案）一份正处在研究阶段的因特网文档（正在进行的工作），没有获得官方承认而且有六个月的生命期。

Internet Engineering Steering Group（IESG，因特网工程指导组）管理因特网工程任务组（IETF）活动的机构。

Internet Engineering Task Force（IETF，因特网工程任务组部）进行设计与开发TCP/IP协议族和因特网一个机构。

Internet Group Management Protocol（IGMP，因特网组管理协议）TCP/IP协议族中处理多播的协议。

Internet Key Exchange（IKE，因特网密钥交换）在SADB中为建立安全关联而设计的协议。

Internet Mail Access Protocol，version 4（IMAP4，因特网邮件访问协议版本4）一种复杂但功能强的处理电子邮件传输的协议。

Internet Mobile Communication（ITM-2000，因特网移动通信）ITU发布的定义了第三代移动电话标准的数据。

Internet model（因特网模型）一种目前的数据通信与网络的五层协议栈。

Internet Network Information Center（INTERNIC，因特网网络信息中心）负责收集和发布有关TCP/IP协议信息的机构。

Internet Protocol（IP，因特网协议）在TCP/IP协议族中的网络层协议，负责跨越一些分组交换网的无连接传输。

Internet Protocol next generation（IPng，下一代因特网协议）参看因特网协议版本6（IPv6）。

Internet Protocol version 4（IPv4，因特网协议版本4）因特网协议目前的版本。

Internet Protocol，version 6（IPv6，因特网协议版本6）因特网协议的第6个版本。

Internet Research Task Force（IRTF，因特网研究任务组）多个工作组的论坛，集中讨论关于因特网远期研究课题。

Internet Security Association and Key Management Protocol（ISAKMP，因特网安全关联和密钥管理

协议）由国家安全局设计的协议，它实现IKE规定的交换。

Internet service provider（ISP，因特网服务提供商）通常是指连接到因特网各种服务的公司。

Internet Society（ISOC，因特网协会）宣传因特网的一个非盈利组织。

Internet standard（因特网标准）一个完整的测试规范，对想要研究因特网的人员来说是有用的并且要遵守的。这是一种正式规定，必须遵循。

internetwork（internet，互联网络）多个网络连接而成的网络。

Internetwork Protocol Control Protocol（IPCP，互联网络协议控制协议）在PPP中，为IP分组建立和终止网络层连接的一对协议。

internetworking（网际互联）使用诸如路由器和网关等网络互联设备将许多网络连接起来形成一个互联网。

intranet（内联网）使用TCP/IP协议栈的专用网络。

inverse domain（反向域）DNS中的一个子域，为给定的IP地址查找域名。

inverse multiplexing（反向多路复用）来自一个源端的数据被分成多个部分，可以通过多条低速线路发送。

IP datagram（IP数据报）Internet互联协议的数据单元。

IP Security（IPSec，IP层安全）IETF设计的一组协议，为在因特网传送的分组提供安全性。

IrDA port（红外端口）允许无线键盘与PC进行通信的端口。

iridium（铱星系统）由66颗卫星构成的网络，为地球上任何两个点之间提供通信。

ISDN user part（ISUP，ISDN用户部分）SS7中提供ISDN网络的服务。

isochronous transmission（等时传输）一种传输类型，整个位流在发送时保持同步。

iterative resolution（迭代解析）IP地址的解析，客户机在得到应答之前会将它的请求发送到多个服务器。

J

jamming signal（干扰信号）在CSMA/CD中，当发送站检测到冲突的时候发出的告知所有站发生碰撞的信号。

Java 用来创建活动Web文档的编程语言。

jitter（抖动）实时流量中的一种现象，是由于接收方连续分组之间的延时不同造成的。

Joint Photographic Experts Group（JPEG，联合图像专家组）一种压缩连续音调图像的标准。

K

Kerberos（Kerberos）Windows2000所用的鉴别协议。

key distribution center（KDC，密钥分发中心）在秘密密钥的密码学中，与每个用户共享密钥的可信任的第三方。

L

LAN emulation（LANE，局域网仿真）用ATM交换机局域网仿真软件。

LAN emulation client（LEC，局域网仿真客户机）在ATM局域网中，从LES接收业务的用户软件。

LAN emulation server（LES，局域网仿真服务器）在ATM局域网中，在源端与目的之间创建虚电路的服务的软件。

leaky bucket algorithm（漏桶算法）对突发流量进行整形的算法。

least-cost tree（最小代价树）一种MOSPF的特性，最小费用树是基于选定的度量而不是最短路径。

legacy ATM LAN（传统异步传输模式局域网）使用ATM技术作为主干连接多个传统局域网的局域网。

line coding（线路编码）将二进制数据转换成信号。

linear block code（线性块编码）在码字中每个码字是其他码字中任何两个码字按2作加是该码字中另一个码字。

line of sight propagation（视线传播）两个间的信号直接在天线之间直线传输。

link（链路）在设备之间传输数据和控制信息的通信通路。

Link Control Protocol（LCP，链路控制协议）一个PPP协议，它负责建立、维持、配置和终止链路。

link local address（本地链路地址）专用局域网所用IPv6地址。

link state advertisement（LSA，链路状态通告）在OSPF中，分发信息的一种方法。

link state database（链路状态数据库）在链路状态路由选择中，从LSP信息中构建的所有路由器共用的数据库。

link state packet（LSP，链路状态分组）在链路状态路由选择中，由一个路由器向所有相邻路由器发送的含有路由选择信息的一个小分组。

link state routing（链路状态路由选择）一种路由选择方法，它使每一个路由器与其他所有路由器共享有关路由器的连接的变化信息。

local access and transport area（LATA，本地接入和传送区域）由一个或多个电话公司服务的区域。

local area network（LAN，局域网）将一座建筑或者邻近建筑内部距离很近的一些设备连接在一起的网络。

local area network emulation（LANE，局域网仿真）使ATM交换机像局域网交换机一样工作的软件。

local exchange carrier（LEC，本地交换电信公司）在一

个LATA内部处理业务的电话公司。

local loop（本地回路）连接用户到电话交换局的链路。

local management information（LMI，本地管理信息）帧中继使用的一个协议，用于提供管理特性。

logical address（逻辑地址）网络层中定义的一种地址。

logical link control（LLC，逻辑链路控制）IEEE项目802.2定义的数据链路层的较高一个子层。

Logical Link Control and Adaptation Protocol（L2CAP，逻辑链路控制和自适应协议）在ACL链路中用于数据交换的蓝牙层。

logical tunnel（逻辑隧道）将一个多播分组封装在一个单播分组中，通过非多播路由器点用多播路由选择。

longest mask matching（最长掩码匹配）CIDR中的一种技术，当搜索路由表时，首先处理最长的前缀。

low earth orbit（LEO，近地轨道）一种极地卫星轨道，高度在500～2000km之间，位于此轨道的卫星，其环绕周期是90～120分钟。

low-pass channel（低通通道）通过频率为0到f的通道。

M

mail transfer agent（MTA，邮件传输代理）一个SMTP组件，能够在因特网上传输邮件。

Management Inforrmation Base（MIB，管理信息数据库）SNMP所用的数据库，它维持网络所必须的信息。

Manchester encoding（曼彻斯特编码）一种数字到数字的极性编码方法，在每个位的间隔中间都有一个跳变以便同步。

man-in the middle struck（中间人攻击）一种密钥管理问题，入侵者可能在特定的发送方和接收方之间截取和发送报文。

mapped address（映射地址）使用IPv6的计算机型发送分组到使用IPv4的计算机，所用的IPv6地址。

mask（掩码）对于IPv4来说，它是一个32位的二进制数，当它和地址块中的地址非进行逻辑与操作可以得到地址块的第一个地址（网络地址）。

maximum transfer unit（MTU，最大传输单元）一个特定网络能够处理的最大数据单元的长度。

medium access control（MAC）sublayer（介质访问控制子层）IEEE项目802.2定义的数据链路层的低子层，它确定了不同局域网络技中的访问方法和访问控制。

medium Earth orbit（MEO，中地轨道）在两个Van Allen辐射带之间的卫星轨道，位于此轨道中的卫星环绕地球需用6小时。

mesh topology（网状拓扑结构）一种网络配置，每一个设备与其他各个设备都有直接的点到点连接。

message access agent（MAA，报文接传达代理）获取存储电子邮件的客户机-服务器的程序。

message authentication（报文鉴别）一种安全措施，接收方对每一发送的报文验证确实是发送方的报文。

message authentication code（MAC，报文鉴别码）一个密钥散列函数。

message transfer agent（MTA，报文传输代理）一个SMTP组件，在因特网上传输报文。

metric（度量）通过某个网络的分配的特定费用。

metropolitan area network（MAN，城域网）可以跨越一个城市地理区域的网络。

microwave（微波）频率范围在2GHz～40GHz的电磁波。

minimum Hamming distance（最短汉明距离）在一组码字集合中，其中所有的码字对中最小的汉明距离。

mobile host（移动主机）一台能从一个网络移动到另一个网络的主机。

mobile switching center（MSC，移动交换中心）在蜂窝电话中，协调所有基站与电话控制局通信的交换局。

mobile telephone switching office（MTSO，移动电话交换局）控制和协调所有蜂窝局和电话控制局通信的局。

modem（调制解调器）由一台调制器和一台解调器构成的设备，它将数字信号转换为模拟信号（调制），将模拟信号转换为数字信号（解调）。

modification detection code（MDC，修改检测代码）散列函数创建的摘要。

modular arithmetic（模运算）运算在有限的整数范围（0到n-1）内进行。

modulation（调制）用信息承载信号改变载波信号的一个或者几个特性。

modulus（模）模运算的上限（n）。

monoalphabetic substitution（单字母替换）一种加密方法，对于出现的每个字符都用集合中的另一字符来代替。

motion picture experts group（MPEG，运动图像专家组）压缩视频的一种方法。

multicast address（多播地址）用于多播的地址。

multicast backbone（MBONE，多播主干）通过使用通道技术支持多播的互联网络路由器的集合。

Multicast Open Shortest Path First（MOSPF，多播开放最短路径优先）一种多播协议，使用多播链路路由选择生成基于源端的最小代价树。

multicast router（多播路由器）具有患实成员列表的路

由器，告诉成员与每台路由器分发多播分组的接口有关。

multicast routing

multicasting（多播）一种传输方式，可将一个单独分组的副本发送给选定的一组接收方。

multihoming service（多接口服务）SCTP提供一种服务，它可以将一台计算机连接到不同的网络。

multiline transmission, 3-level (MLT-3) encoding（3电平多线路传输编码）一种线路编码机制，特点是信号有3种电平。在位1的开始处有跳变。

multimode graded-index fiber（多模渐变折射率光纤）一种纤芯，其纤芯的折射系数是渐变的。

multimode step-index fiber（多模阶跃折射率光纤）一种纤芯，其纤芯的折射系数是均匀的，在纤芯与包层的交界面折射系数发生突变。

multiple access（MA，多路协问）一种线路访问方法，每一个站点可以自由协问线路。

multiplexing unicasting（多个单播）发送一个报文多个副本，每一个都具有不同单播地址。

multiplexer（MUX，复用器）用于多路复用的设备。

multiplexing（多路复用）为了在一条数据链路上传输，将多个信号的信号进行合并的过程。

multiplicative decrease（乘性减少）一种避免拥塞的技术，将门限设置为最近拥塞窗口大小的一半，拥塞窗口大小再次从1开始。

Multipurpose Internet Mail Extension（MIME，多用途网特网邮件扩展）SMTP的一种补充，可以通过SMTP发送非ASCII数据。

multistage switch（多级开关）一种开关阵列，设计的目的是减少交叉点的数。

multistream service（多流服务）SCTP提供的一种服务，用不同数字流的数据块传输。

N

name space（命名空间）在互联网上分配给机器的所有名称。

name-address resolution（名称-地址解析）指一个名称映射为地址或者将一个地址映射为名称。

Needham-Schroeder protocol（Needham-Schroeder协议）一种密钥管理协议，两个实体之间使用多次符问应答交互。

netid（网络号）IP地址中标志网络的部分。

network（网络）由若干连接的节点构成的系统，目的是共享数据、硬件和软件。

network access point（NAP网络访问点）连接十个网的复杂交换站点。

network address（网络地址）第一个网络与因特网的其他网络互联于的地址，它是地址块中的第一个地址。

network address translation（NAT，网络地址转换）一种技术，它允许专用网络使用一组专用地址用于内部通信，而使用一组因特网地址用于外部通信的技术。

network allocation vector（NAV，网络分配矢量）在CSMA/CA中，站点可以检查媒体是否空闲之前必须等待的时间。

Network Control Protocol（NCP，网络控制协议）PPP中的一组控制协议，用来封装来自网络层协议的数据。

network interface card（NIC，网络接口卡）一种电子设备，可以在站点的内部或者外部。它包含可以使该站点连接到网络的电路。

network layer（网络层）因特网模型的第三层，负责分组到达最终目标的传送。

Network Virtual Terminal（NVT，网络虚拟终端）允许远程登录的一种TCP/IP应用协议。

network-specific routing（特定网络路由选择）网络中所有主机共享路由选择表中一条记录的路由选择。

network-to-network interface（NNI，网络到网络的接口）在ATM中两个网络之间的接口。

next-hop routing（下一跳路由选择）一种路由选择方法，只有下一站的地址在路由表中列出，而不是分组到目的所有决定的完整列表。

node（节点）在网络上一个可寻址的通信设备（例如一台计算机或者路由器）。

node-to-node delivery（节点到节点的传送）将一个数据单元从一个节点传送到另一个节点。

noise（噪声）可以被传输介质接收的随机电信号，导致数据受到损伤或者丢失。

noiseless channel（无噪声通道）无差错通道。

noisy channel（噪声通道）在数据传输中可能会产生差错的通道。

nonce（临时值）一个很大的随机数，只用一次。用来从已经使用过的鉴别请求中辨别出一个新的鉴别请求。

nonpersistent strategy（非持续策略）一种随机访问方法。在这种方法中，站点在发现冲突以后会等待一段随机时间。

nonrepudiation（不可否认性）一种安全问题，它指接收方必须能够证明接收到的报文来自于某一个特定的发送方。

nonreturn to zero（NRZ，不归零编码）一种数字到数字的极性编码方法，在该方法中，信号电平是正值或者负值。

nonreturn to zero，invert（NRZ-I，不归零反向编码）一种不归零编码方法，信号电平每遇到1就翻转一次。

nonreturn to zero，level（NRZ-L，不归零电平编码）一种不归零编码方法，信号电平直接与位值相关。

normal response mode（NRM，正常响应模式）HDLC中的一种通信模式，从站在传输进行之前必须获得主站的许可。

Nyquist bit rate（奈奎斯特速率）基于奈奎斯特定理的数据速率。

Nyquist theorem（奈奎斯特定理）该定理说明了表示一种模拟信号所必需的采样数量，即等于原始信号最高频率的2倍。

O

Oakley 一个密钥生成协议，它由Hilaric Orman研发，也是IKE协议三个组成部分之一。

omnidirectional antenna（全向天线）在各个方向都可以发射和接收信号的天线。

one's complement（反码）一种二进制数的表示方法，即将一个数所有二进制位取反或得出该数的反码。

open shortest path first（OSPF，开放最短路径优先）基于链路状态路由选择的一种内部路由选择协议。

Open Systems Interconnection（OSI）model（开放系统互连模型）ISO定义的数据通信的七层模型。

open-loop congestion control（开环拥塞控制）用于防止拥塞的策略。

optical carrier（OC，光载波）同步光纤网络（SONET）中定义的光线波层次。

optical fiber（光纤）一个细的玻璃导管或透明材料用来散传光束。

orbit（轨道）卫星环绕地球运行的路径。

Orthogonal Frequency Division Multiplexing（OFDM，正交频分多路复用）一种与频分复用类似的多路复用方法，所有的子波段在指定的时间内只由一个源端使用。

orthogonal sequence（正交序列）元素之间具有特殊性质的序列。

Otway Rees protocol（Otway Rees协议）一种密钥管理协议，其所需步骤比Needham Schroeder方法要少。

out-of-band signaling（带外信令）用户数据和控制数据的通道彼此独立。

output feedback（OFB）mode（输出反馈模式）与CFB模式有一个差别，密文中每一位都与前一位或前面的位无关。

P

packet switching（分组交换）使用分组交换网络的数据传输。

packet-filter firewall（分组过滤防火墙）一种防火墙，根据网络层和传输层的头部信息决定是转发分组还是阻止分组。

packet-switched network（分组交换网）一种网络，数据以称为分组的独立单元进行传输。

parabolic dish antenna（抛物面碟形天线）形状类似抛物面的天线，用于地面微波通信。

parallel transmission（并行传输）在传输中一组中的位同时传输，各使用单独一条的链路。

parity check（奇偶校检）使用奇偶位的一种差错检测方法。

partially qualified domain name（PQDN，部分域名）没有包含从主机到根节点所有层次的一个域名。

Password Authentication Protocol（PAP，口令鉴别协议）在PPP中使用的两步鉴别协议。

path vector routing（路径向量路由选择）BGP的基本路由选择方法，分组经过的自治系统必须被明列出。

P-box（P-盒）加密使用的一种硬件电路，通过它将输入产生输出。

peer-to-peer process（对等进程）在两个相同层进行发送方机器和接收方机器上通信的进程。

per hop behavior（PHB，逐期行为）在Diffserv模型中，一个6位字段定义了分组的处理机制。

period（周期）完成一个完整的循环所需要的时间。

periodic signal（周期性信号）以重复模式出现的信号。

permanent virtual circuit（PVC，永久虚电路）一种虚电路传输方法，源端和目的端在连续的基础上使用同一条虚电路。

persistent connection（持续连接）一种连接方式，服务器在发送响应以后会使连接保持开启以响应更多的请求。

Personal Communication System（PCS，个人通信系统）提供多种通信服务的商业蜂窝电话系统的通用术语。

phase（相位）信号在时间上的相对位置。

phase modulation（PM，相位调制）一种根据到根据的调制方法，载波信号的相位随着调制信号的变化而变化。

phase shift keying（PSK，相移键控）一种数模调制方法，载波信号的相位随着要表示的特定位模式

变化的变化。

PHY sublayer（PHY子层）实现以太网中的收发器。

physical address（物理地址）数据链路层使用的设备的地址（MAC地址）。

physical layer（物理层）互联网模型的第一层，负责处理介质的机械和电气规范。

piconet（微微网）一种蓝牙网络。

piggybacking（式样）在数据流中嵌入确认信息。

pipelining（流水线）在Go-Back-N ARQ方法中，在未接到前面帧的有关情况之前，就发送后面的多个帧。

pixel（像素）图像的基本元素。

Plain old telephone system（POTS，普通传统电话系统）使用语音通信的电话网。

plaintext（明文）加密/解密技术中的原始报文。

playback buffer（回放缓冲区）用来存储准备播放数据的缓冲区。

point coordination function（PCF，点协调功能）在无线局域网中，用于基础结构网络中的一种可选的和复杂的访问方法。

point of presence（POP，服务接入点）电话公司可以与其他公司进行联系的交换点。

point to point connection（点到点连接）两个设备间的专用传输链路。

Point to Point Protocol（PPP，点到点协议）在四种线路上传输数据的一种协议。

poison reverse（毒性逆转）水平分割策略中增加一个特性，该特性是对来自接口更新路由表的一个记录，设置为无穷大。

polar encoding（极性编码）用两个振幅电平（正或负）表示一种数字到模拟的编码方式。

policy routing（策略路由选择）一种路径向量路由特征，路由表是基于网络管理员制定规则，而不是基于度量值。

poll（轮询）在主站/从站访问方式中，主站询问从站是否有数据需要传输。

poll/final（P/F）bit（轮询/终止位）HDLC帧控制字段的一个位，如果主站正在发送数据，则该位可为轮询位，如果从站正在发送数据，则该位可为终止位。

poll/select（轮询/选择）使用轮询和选择过程的访问方法协议。参见poll（轮询）和select（选择）。

polling（轮询）一种访问方法，其中一个指定为主站与其余的站为从站，由主站控制访问。

polyalphabetic substitution（多字母替换）一个字符的每一次出现可用不同字符替换的加密方法。

polynomial（多项式）作为CRC除数的一个代数表达。

port address（端口地址）在TCP/IP中一个整数，用来标识一个进程。

port number（端口号）参看port address（端口地址）。

Post Office Protocol，version 3（POP3，邮局协议版本3）一个流行而简单的邮件访问协议。

p-persistent（p-持续策略）一种CSMA持续性策略，当发现线路空闲时，站点以概率p发送分组。

preamble（前导符）IEEE 802.3帧中一个由交替的0和1组成的7字节的字段，对接收方提供警告和同步。

predictive encoding（预测编码）在快照压缩中，仅对采样间的差进行的编码。

prefix（前缀）地址范围中的公共部分。

presentation layer（表示层）OSI模型的第6层，负责数据的转换、加密、监视和数据压缩。

Pretty Good Privacy（PGP，相当好的保密性）提供了发送电子邮件过程中安全的所有四个方面的一种协议。

primary station（主站）在主站/从站访问方式中，向从站发出命令的站点。

priority queuing（优先权队列）给各个分组分配各自优先权的一种队列技术。

privacy（保密性）安全的一个方面，分组仅对特定的接收方有意义。

private key（私钥）在传统加密方法中，仅在一对设备（一个发送器和一个接收器）之间共享的一个密钥。在公开密钥加密技术中，如私钥只由接收方掌握。

private network（专用网）和互联网隔离的网络。

process（进程）一个正在执行的应用程序。

process-to-process delivery（进程到进程的传递）分组从发送进程到目的进程的传递。

Project 802（802项目）IEEE进行的一个项目，试图解决LAN的不兼容性。参见IEEE Project 802。

propagation speed（传播速度）信号或位传输的速率，用每秒距离来度量。

propagation time（传播时间）信号从一点到另一点所要求的时间。

protocol（协议）管控的规则。

Protocol Independent Multicast（PIM，协议独立多播）一个多播协议族，由两个协议组成：PIM-DM和PIM-SM。每个协议都依赖单播协议。

Protocol Independent Multicast，Dense Mode（PIM-DM，PIM密集模式）一种使用RPF和修剪/嫁接策略来处理多播的稠密路由协议。

Protocol Independent Multicast，Sparse Mode（PIM-SM，

PIM稀疏模式）类似于CBT，它是一种组共享路由选择协议，使用一个会合点（RP）作为树的源。

protocol suite（协议族）为一个复杂通信系统定义的协议栈或协议族。

proxy ARP（代理ARP）一种技术，它产生从分子网的效应，用一个服务器为多个主机回答ARP的请求。

proxy firewall（代理防火墙）基于报文自身的有效信息（在应用层中）来过滤报文的一种防火墙。

proxy server（代理服务器）保存对最近请求的响应副本的一台计算机。

pruning（裁剪）从一个接口停止多播报文的发送。

Pseudoheader（伪报头）IP头部的一些信息，它仅为了计算UDP和TCP分组的校验和。

pseudorandom noise（PN，伪随机噪声）FHHS中所用的伪随机码发生器。

public key infrastructure（PKI，公钥基础设施）CA服务器的层次结构。

public-key cryptography（公开密钥加密学）基于不可逆的算法的一种加密方法。该方法使用两种密钥：一种是大家都知道的公开密钥；另一种仅是为接收方才能知道的私钥。

pulse amplitude modulation（PAM，脉冲振幅调制）模拟信号进行采样的一种技术，对于采样的数据生成一串脉冲的序列。

pulse code modulation（PCM，脉冲码调制）修改PAM脉冲以创建数字信号的技术。

pulse stuffing（脉冲填充）在TDM中，对低速的输入线路添加虚位的一种技术。

pure ALOHA（纯ALOHA）最早的ALOHA。

Q

quadrature amplitude modulation（QAM，正交调幅）同时改变载波信号的振幅和相位来表示数字信号的一种数字-模拟调制方式。

quality of service（QoS，服务质量）有关连接性能的一组特性。

quantization（量化）分配给信号振幅值的一个特定范围。

quantization error（量化差错）来就在量化过程中（模拟到数特性）引入的差错。

queue（队列）一个等待的列表。

R

radio wave（无线电波）频率在3kHz～300GHz范围内的电磁能量。

random access（随机访问）一种介质访问策略，每个站点访问介质不受任何其他站点的控制。

ranging（测距）在HFC网络中，决定CM和CMTS之间的距离的一个进程。

rate adaptive asymmetrical digital subscriber line（RADSL，速率自适应非对称数字用户线路）基于DSL的技术，按照通信的类型给以不同的数据速率。

read-only memory（ROM，只读存储器）存储内容永久不改变的存储器。

Real-time Streaming Protocol（RTSP，实时流协议）一种带外控制协议，为音频/视频流增加更多的功能。

Real-time Transport Control Protocol（RTCP，实时传输控制协议）它和RTP报文一起使用，用来控制数据流量和欧量，并且支持接收方向源端或源端组发送反馈报文。

Real-time Transport Protocol（RTP，实时传输协议）一种实时通信量协议，和UDP一起使用。

reconciliation sublayer（协调子层）快速以太网的一个子层，数据以4位的格式传输到MII。

recursive resolution（递归解析）IP地址的分解方法，客户机和服务器发送自己的请求并由服务器最终返回一个响应。

redundancy（冗余）为了差错控制在报文中增加的额外的位。

Reed-Solomon（Reed-Solomon码）一种复杂但有效的循环码。

reflection（反射）光线在两种介质的界面产生的一种反射可来范现象。

refraction（折射）当光线通过一个介质到另一个介质时，产生方向改变的现象。

regional ISP（区域ISP）连接到一个或多个NSP的小型ISP。

registrar（注册机构）登记新域名的机构。

relay agent（延迟代理）对于HOOTP，能发送本地请求到远端服务器的路由器。

reliability（可靠性）QoS流的一个特征，传输的可靠程度。

remote bridge（远程网桥）连接LAN和点对点网络的设备，通常用在主干网中。

rendezvous router（会合路由器）作为每个多播组的核心或中心的路由器，它是展集树的根。

rendezvous point tree（会合点树）一个组共享树的方法，每个组都有一个树。

repeater（中继器）用信号再生方法使其传输距离得到伸展的设备。

replay attack（重放攻击）入侵者截取报文后再重新发送

的攻击方法。

Request for Comment（RFC，请求评论）关于证特网其个问题的因特网正式文档。

resolver（解析程序）需要转射地址到名字或名字到地址的主机使用的DNS客户机。

Resource Reservation Protocol（RSVP，资源预留协议）一个信令协议，使IP创建预留资源以提高QoS。

retransmission time-out（重传超时）控制分组重发的定时器超时。

return to zero（RZ，归零）一种数字到数字的编码技术，在每个位间隔中段后信号电平为零。

reuse factor（复用因子）在移动电话技术中，拥有不同频率集的单元的数目。

Reverse Address Resolution Protocol（RARP，反向地址解析协议）一种TCP/IP协议，它可以使一个主机根据其物理地址找出相应的IP地址。

reverse path broadcasting（RPB，逆路径广播）一种技术，路由器仅转发从源到路由器具有最短路径的分组。

reverse path forwarding（RPF，逆路径转发）一种技术，路由器仅转发从源到路由器具有最短路径的分组。

reverse path multicasting（RPM，逆路径多播）一种技术，该技术将修剪和嫁接添加到RPB中，以创建一棵多播最短路径树，该树支持动态成员关系的改变。

Rijndael algorithm（Rijndael算法）该算法按两个创建的比利时人Vincent Rijmen和Jean Daemen的名字命名，它是AES的基础。

ring topology（环状拓扑）所有设备都连接成一个环的拓扑结构，环上每个站从前一个站接收数据单元，将其再生，然后转发到下一个站。

Rivest，Shamir，Adleman（RSA）encryption（RSA算法）encryption（RSA加密）表示RSA encryption。

RJ45 一种双绞线电缆连接器。

roaming（漫游）在移动电话技术中，用户在自己服务提供商的区域之外还能够通信的能力。

root server（根服务器）在DNS中，包含有整个树的服务器，根服务器通常不存储关于域的任何信息。

rotary dialing（脉转信号）通过电话向终端交换局发送数字信号来访问交换式站点。

rotation cipher（循环加密算法）有密钥或无密钥的加密，由输入位向左或向右循环产生输出位。

round trip time（RTT，往返时间）报文从源端到目的再返回所需的时间。

route（路由）分组经过的路径。

router（路由器）工作在长三层的网际互连设备。一个路由器连接两个或多个网络。它将一个分组从一个网络转发到另一个网络。

routing（路由选择）路由器所完成的过程，用来发现数据报的下一跳。

Routing Information Protocol（RIP，路由选择信息协议）基于距离向量路由算法的路由选择协议。

routing table（路由选择表）含有路由器转发分组所需信息的表，这些信息包含网络地址、费用、下一跳的地址等。

RSA cryptosystem（RSA加密）由Rivest、Shamir和Adleman开发的一种流行的公钥加密方法。

S

sampling（采样）以固定的时间间隔获取信号振幅的过程。

sampling rate（采样率）在采样过程中每秒获得的样本数量。

satellite network（卫星网络）提供从地面上一点到另一点通信的节点的联合。

S-box（S-盒）由解码器、P-盒及编码器组成的加密设备。

scatternet（载射网络）微微网络的联合。

scrambling（扰动）在数字到数字的转换中，修改线路编码块的部分规则以生成位同步。

secondary station（从站）在轮询/选择访问方法中，对主站发出的命令做回响应的站。

secret-key encryption（秘密密钥加密）一种加密方法，其加密和解密的密钥相同，发送方和接收方使用相同的密钥。

Secure Hash Algorithm 1（SHA 1，安全散列算法-1）由国家标准与技术协会（NIST）设计的一种散列算法，作为联邦信息处理标准。

Secure Socket Layer（SSL，安全套接字层）一种安全协议，用来对应用层产生的数据提供安全和压缩服务。

Security Association（SA，安全关联）一种IPSec安全协议，用来在两个主机间创建逻辑连接。

security association database（SADB，安全关联数据库）定义一组安全关联的数据库。

security parameter index（SPI，安全参数索引）区别一个安全关联与另一个安全关联的参数。

segment（报文段）TCP层的分组，也指设备共享的传输介质的长度。

segmentation（分段）将一个报文分成多个分组，通常在传输层完成。

segmentation and reassembly（SAR，分割与重组）ATM协议中较低的一个AAL子层，在此子层加上头部和尾部来组成一个48字节的单元。

select（选择）在轮询/选择访问方法中一个规程，即主站询问从站是否已准备接收数据。

selective-repeat ARQ（选择性重复ARQ）一种差错控制方法，只重发有差错的帧。

self-synchronization（自同步）通过编码方法对长四的1或0提供同步。

semantics（语义）每个位组的意义。

sequence number（序列号）一个数用来表明帧或分组在一个报文中的位置。

serial transmission（串行传输）只用一条链路，一次一个的传输方式。

server（服务器）一个可以为别的程序（称为客户机）提供服务的程序。

server control point（SCP，服务控制点）SS7中的术语，该节点控制网络全部操作。

Session Initiation Protocol（SIP，会话初始化协议）在IP的语音中，一种应用协议，用来建立、管理和终止一个多媒体会话。

session layer（会话层）OSI模型的第5层，负责建立、管理和终止两个终端用户间的逻辑连接。

setup phase（建立连接阶段）在虚电路交换中，源端项目的使用各自的全局地址通过交换建立连接表记录的阶段。

Shannon capacity（香农容量定理）理论上最高的速度数据速率。

shielded twisted pair（STP，屏蔽双绞线）双绞线也在金属薄片或网状屏蔽层中，可防止电磁干扰。

shift cipher（移位加密）最简单的单字母加密，在由字母组成的密文和明文中，它的加密算法是字符按字符表后移，解密算法是前移。

shift register（移位寄存器）寄存器中每一位在时钟脉冲触发下接收输入端口对应的位，存储该位，并在输出端口展示它。

short interframe space（SIFS，短帧间间隔）在CAMA/CA中，目的方接收到RTS后等待的一段时间。

shortest path tree（最短路径树）利用Dijikstra算法得到的路由表。

signaling connection control point（SCCP，信令连接控制点）在SS7中，该控制点用作特殊服务，如800呼叫。

signal element（信号单元）信号的最小单元（时间状态），它代表一个数据单位。

signal point（SP，信号点）SS7中的术语，连接用户电话或计算机的信号点。

signal rate（信号速率）一秒发送信号单元的个数。

signal transport port（STP，信号传输端口）SS7中的术语，信令网络所用的节点。

Signaling System Seven（SS7，7号信令系统）信令网络中使用的一个协议。

signal-to-noise ratio（SNR，信噪比）信号强度除以噪声强度。

silly window syndrome（疯涂窗口综合症）一种情况，即接收方通知的是一个小窗口，而发送方也发送一个小报文段的情形。

simple and efficient adaptation layer（SEAL，简单有效适配层）为因特网设计的AAL层（ALL5）。

simple bridge（简单网桥）一种连接两个网段的设备，它需要人工维护和更新。

Simple Mail Transfer Protocol（SMTP，简单邮件传输协议）定义因特网上电子邮件服务的TCP/IP协议。

Simple Network Management Protocol（SNMP，简单网络管理协议）规定因特网的管理过程的TCP/IP协议。

Simple Protocol（简单协议）简单协议是一种无流量控制和差错控制的协同方法。

simplex mode（单工模式）通信是单向进行的一种传输模式。

sine wave（正弦波）一个振幅随时间周期变化的旋转向量。

single-bit error（单个位差错）在一个数据单元中，仅有一个位被改变的差错。

single-mode fiber（单模光纤）直径非常小的一种光纤，它将光束限制在很小角度内，结果得到几乎是水平的光束。

site local address（本地站点地址）具有数个网络但没有连接到因特网的站点IPv6地址。

SKEME 由Hugo Krawczyk设计的密钥交换协议，它是形成IKE基础的三个协议之一。

sky propagation（天空传播）无线电波传播入电离层然后再返回地面的传播过程。

slash notation（斜杠标记法）一种在掩码中指明1的数目的建记方法。

sliding window（滑动窗口）一种协议，允许在收到确认

之前，可以发送多个数据单元。

sliding window ARQ（滑动窗口ARQ）使用滑动窗口概念的一个差错控制协议。

slotted ALOHA（时隙ALOHA）修改过的ALOHA访问方法，其中时间被划分成时隙，每个站点保存分配的时隙开始时才能发送数据。

slow convergence（慢收敛）一种RIP缺陷，当互联网中的某一改变非常慢地传播到互联网的其余部分时出现慢收敛。

slow start（慢速启动）一种阻塞控制的方法，拥塞窗口大小最初以指数形式增大。

socket address（套接字地址）包含一个IP地址和一个端口号的结构。

source quench（源端抑制）ICMP使用的流量控制方法，却不发生拥塞时，通知源端减慢或停止发送。

source routing（源路由选择）分组的发送方指明分组传输的路径。

source routing bridge（源路由网桥）一个源端或目的站点，执行阻止循环的透明网桥的职责。

source-based tree（基于源树）该树是多播协议使用的多播树，其中每一模树由源端和组组成。

source-to-destination delivery（源端到目的端传递）一个报文从源发送方到预期接收方的传递。

space propagation（空间传播）一种能够穿透电离层的传播类型。

space-division switching（空分交换）路径在空间上相互独立地交换。

spanning tree（生成树）领袖作为根，组成员作为叶子的树，连接所有节点的一棵树。

spatial compression（空间压缩）通过去除冗余部分来压缩图像。

spectrum（频谱）信号的频率范围。

split horizon（水平分割）提高RIP稳定性的一种方法，路由器选择发送更新信息所用的端口。

spread spectrum（扩频）一种无线传输技术，该技术需要源带宽的几倍的带宽。

Standard Ethernet（标准以太网）速率为10Mbps的传统以太网。

star topology（星型拓扑）所有站点都连接到中心设备（集线器）的一种拓扑结构。

start bit（起始位）异步传输中指明传输开始的位。

state transition diagram（状态转换图）一个用来说明有限状态机的状态变化的图。

static document（静态文档）在万维网服务器中创建并存储的固定内容的文档。

static mapping（静态映射）使用逻辑地址和相应的物理地址的列表来进行地址解析的技术。

static routing（静态路由选择）一种类型的路由选择，其中路由表保持不变。

stationary host（固定主机）一直和网络保持联系的主机。

statistical TDM（统计TDM）一种TDM技术，高效率动态分配时隙。

status line（状态行）HTTP响应报文中的一行，它由HTTP版本号、空格、状态代码、空格和状态语句构成。

stop bit（停止位）在异步传输中，指明传输结束的一个或多个位。

stop and wait ARQ（停止等待ARQ）一种使用停止等待流量控制的差错控制协议。

Stop and Wait Protocol（停止等待协议）一种协议，发送方发送一帧必须被确认后，才能继续发送下一帧。

store and forward switch（存储转发交换）直到完整的分组到达给入缓冲区存储后才能交换。

straight tip connector（直插式连接器）一种使用卡口锁定系统的光纤连接器。

Stream Control Transmission Protocol（SCTP，流控制传输协议）为网络电话及其应用而设计的传输层协议。

streaming live audio/video（流式实时音频/视频）用户可以听或看来自因特网的广播数据。

streaming stored audio/video（流式存储音频/视频）用户可以听或看从因特网上作为文件下载的数据。

strong collision（强碰撞）创建两个具有相同摘要的报文。

Structure of Management Information（SMI，管理信息结构）在SNMP中，网络管理所用的组件。

STS multiplexer/demultiplexer（STS复用器/分离器）一种SONET设备，用来多路复用或分离信号。

stub link（残桩链路）仅用一个路由器连接的网络。

subnet（子网）网络的一个部分。

subnet address（子网地址）子网的网络地址。

subnet mask（子网掩码）子网的掩码。

subnetwork（子网）网络的一部分。

subscriber channel connector（用户通道连接器）一种使用滑/按锁定机制的光缆连接器。

substitution cipher（替换加密算法）由P-盒、编码器和解码器定义的。用n位替换另外n位的一种位级加密方法。

suffix（后缀）对于网络，指地址变化的部分（和主机号类似）；在DNS中，指组织机构用来定义主机或资

到的一个字符串。

summary link to AS boundary router LSA（到AS边界路由器的汇总链路LSA）一个LSA分组，它使一个区域内的路由器知道一个自治边界路由器的路由。

summary link to network LSA（到网络的汇总链路LSA）一个LSA分组，确定到达区域外部网络的费用。

supergroup（超群）一个由5个组构成的单一信号。

supernet（超网）由几个或更多个较小的网络组成的网络。

supernet mask（超网掩码）超网掩码。

switch（交换机）将多条通信链路连接在一起的设备。

switched Ethernet（交换以太网）使交换机代替集线器的以太网，可以引导传输到达其目的端。

switched virtual circuit（SVC，交换虚电路）一种虚电路传输方法，仅在交换持续期间建立和存在的虚电路。

switched/56（交换式/56业务）两个用户间的临时56 Kbps数字连接。

switching office（交换局）进行电话交换的地方。

symmetric digital subscriber line（SDSL，对称数字用户线路）一种DSL类技术，与HDSL类似，但只使用一根双绞线电缆。

symmetric-key cryptography（对称密钥密码学）一种加密方法，加密和解密采用同一个密钥。

synchronization points（同步点）会话层为流控和差错控制在数据中引入的点。

synchronous connection oriented（SCO）link（面向同步连接的链路）在蓝牙网中，在主链路和从链路间所创建的一条用十核对定为经间隔保持特定时间片的物理链路。

Synchronous Digital Hierarchy（SDH，同步数字系列）与SONET基本相同的ITU-T标准。

Synchronous Optical Network（SONET，同步光纤网络）ANSI开发的使用光纤技术传输高速数据的标准。该标准可用于传输文本、音频和视频信息。

synchronous payload envelope（SPE，同步有效载荷封装）包含用户数据和传输开销的SONET帧的一部分。

synchronous TDM（同步TDM）一种TDM技术，其中不发送数据，每个输入端有对输出端分配的行为。

synchronous transmission（同步传输）一种传输方法，在发送方和接收方之间需要有预定的定时关系。

synchronous transport module（STM，同步传输模块）种用于SDH层次结构的信号。

synchronous transport signal（STS，同步传输信号）一种用于SONET层次结构的信号。

syndrome（校正子）对码字使用类似检测函数生成一个位序列。

syntax（语法）数据的结构或格式，即表示它们的规则。

T

Tlines（T线路）为传送为数字形式的语音和其他信号而设计的数字线路体系，这个体系定义了T-1、T-2、T-3和T-4线路。

tandem office（中继局）电话网中区连局。

TCP/IP protocol suite（TCP/IP协议族）定义为能用特别进行传输交换的5层协议族。

teardown phase（拆除阶段）在虚电路交换中，源端和目的端通知交换机删除它们的表项的阶段。

telecommunications（电信）使用电子设备进行远程信息交换。

teleconferencing（电话会议）远程用户间的音频和视频通信。

Telnet一种提供光纤通信（宽带通讯，低差错率和低延迟）的卫星系统。

telephone user port（TUP，电话用户端口）在SS7中的一个三层协议，它处理所有呼叫建立的响应。

temporal compression（时间压缩）一种除去冗余帧的MPEG压缩方法。

Ten-Gigabit Ethernet（10千兆以太网）一种具有速率为10Gbps的新型以太网。

Terminal Network（TELNET，终端网络）一种允许远程终端的通用的客户机/服务器程序。

three-way handshaking（三次握手）建立连接或终止连接的一个事件序列，包括请求，然后是对请求确认，最后是确认的证实。

throughput（吞吐量）在1秒内能通过一个节点的位数。

ticket granting server（TGS，票证授权服务器）一种发布票证的Kerberos服务器。

time division duplexing TDMA（TDD-TDMA，时分双工复用）在蓝牙网络中，在从站和接收站点之间发送和接收数据所使用的一种双工（而不是半双工）通信。

time division multiple access（TDMA，时分多路访问）种带宽共享一条分时通道的多路访问方法。

time to live（TTL，生存时间）分组的生存时间。

time division multiplexing（TDM，时分复用）将多个低速通道合并成共享高速通道上时间的一种技术。

time-division switching（时分交换）一种电路交换技术。

使用了时分复用来完成交换。

time-domain plot（时域图）信号的振幅通过时间表示的图。

time-slot interchange（TSI，时隙交换）由RAM和控制单元组成的时分交换。

token（令牌）在令牌传递的访问方法中，所使用的一个小分组。

token bucket（令牌桶）一种算法允许空闲主机以令牌的形式累积信任。

token passing（令牌传递）一种访问方法，该方法使令牌在网络中循环，假获令牌的站点就能发送数据。

Token Ring（令牌环）使用环形拓扑结构和令牌传递访问方法的一种局域网。

topology（拓扑）包括设备的物理排列的网络结构。

traffic control（通信量控制）用于整形和控制广域网上通信量的一种方法。

traffic shaping（通信量整形）为改善QoS而控制发送给网络的通信量和数据速率的一种机制。

trailer（尾部）附加于数据单元之后的控制信息。

transaction capabilities application port（TCAP，事务处理能力应用端口）SS7中上层协议，它提供远端进程的序列，可使在一台计算机应用程序实现另一台计算机上的进程。

transceiver（收发器）一种既能发送又能接收的设备。

transient link（过渡链路）与多个路由器相连的网络。

Transmission Control Protocol（TCP，传输控制协议）TCP/IP协议族中的传输协议。

Transmission Control Protocol/Internetworking Protocol（TCP/IP，传输控制协议/网际协议）定义跨越因特网进行传输交换的5层协议族。

transmission medium（传输介质）连接两台通信设备的物理路径。

transmission rate（数据传输率）每秒发送的位数量。

transparency（透明）能够发送任何位模式数据的能力，而不会误认为是控制位。

transparent bridge（透明网桥）学习网桥的另一种名称。

transparent data（透明数据）包含控制位模式的数据，但不会被解析为控制。

transport layer（传输层）因特网和OSI模型中的第4层，负责可靠的端到端传输和差错恢复。

Transport Layer Security（TLS，传输层安全）SSL的IETF标准版本，它们之间只有微小的差别。

transposition cipher（换位加密算法）通过变换字符位置的字符级加密方法。

trellis-coded modulation（格码调制）一种包括纠错的调制技术。

trilateration（三边测量）已知到三个不同地点的距离，寻找该位置一种2-段方法。

triple DES（三重DES）一种与DES兼容的算法，它使用三个DES块和两个56位密钥。

Trivial File Transfer Protocol（TFTP，简单文件传输协议）一种不可靠文件传输的TCP/IP协议，它不需要客户机和服务器之间进行复杂的交互。

trunk（干线）处理两个局间通信的传输介质。

tunneling（隧道技术）在多播中，它指将多播分组封装成单播分组，并通过网络进行发送的过程。在VPN中，它指加密的IP数据报封装在另一个外部数据报中。对IPv6而言，它指两台使用IPv6的计算机相互间要进行通信，而其传送的分组要经过一个使用IPv4的区域时所采用的一种策略。

twisted pair cable（双绞线电缆）由两根绞合起来的绝缘线构成的传输介质。

two-dimensional parity check（二维奇偶校验）一种2维禁锢检测方法。

type of service（TOS，服务类型）规定数据报处理的类型或数值。

U

unbalanced configuration（非平衡配置）一种HDLC配置，其中一台设备是主设备，而其他设备是从设备。

unguided medium（无向介质）一种没有物理边界的传输介质。

unicast address（单播地址）只属于一个目的地的地址。

unicast routing（单播路由选择）分组只向一个目的地发送的过程。

unicasting（单播）只有一个目的地的分组发送。

Unicode（统一码）计算机科学中定义的国际字符集。

unidirectional antenna（单向天线）只从一个方向发送或接收信号的天线。

Uniform Resource Locator（URL，统一资源定位符）标识万维网的一个页面的字符串（地址）。

unipolar encoding（单极编码）一种数字到数字的编码方法，其中非零值表示为1或0，而其他的位由零值来表示。

unshielded twisted-pair（UTP，无屏蔽双绞线）一种电缆，其中线对是相互合在一起以减少噪音和串扰，参见twisted-pair cable和shielded twisted-pair。

unspecified bit rate（UBR，未指定比特率）一种ATM服务类型的数据速率，它只规定了尽力传递。

uplink（上行链路）从地球站到卫星的信号传输。

uploading（上传）发送本地文件或数据到远程站点。

user agent（UA，用户代理）一个SMTP的组件，它作备报文，创建信封，并将该报文放入该信封中。

user authentication（用户鉴别）一种安全措施，在通信开始之前就对发送方的身份进行验证。

user datagram（用户数据报）UDP协议中的分组名称。

User Datagram Protocol（UDP，用户数据报协议）一种无连接的TCP/IP传输层协议。

user network interface（UNI，用户网络接口）用户和ATM网络的接口。

user support layers（用户支持层）指会话、表示和应用层。

user-to-network interface（UNI，用户网络接口）在ATM中，一台终端（用户）和一台ATM交换机之间的接口。

V

V series（V系列）一种ITU-T标准，定义了电话线上的数据传输。如标准V.32、V.32b、V.90和V.92。

variable bit rate（VBR，可变比特率）为需要变化比特率的用户所提供的一种ATM服务类型的数字速率。

very high bit rate digital subscriber line（VDSL，甚高速数字用户线路）一种短距离的基于DSL的技术。

video（视频）图像的记录或发送，表动画。

Vigenere cipher（Vigenere密码）一种多字母替换方案。该方案利用了字母在明文中以及在字母表中的位置。

virtual circuit（VC，虚电路）在发送和接收计算机之间的一条逻辑电路。

virtual circuit switching（虚电路交换）在交换广域网中使用的交换技术。

virtual link（虚链路）在物理链路断开时，在两台路由器间所创建的一条OSPF连接。这两台路由器间的连接使用了一条可能通过几台路由器的路径。

virtual local area network（VLAN，虚拟局域网）通过软件方法将一个物理局域网分成多个虚拟工作组的技术。

virtual private network（VPN，虚拟专用网）一种创建网络的技术。该网络在物理上公开但虚拟专用的。

virtual tributary（VT，虚拟支流）可插入到一个SONET帧的局部有效载荷，它与另一局部有效载荷合在一起填满这个帧。

Voice Over Frame Relay（VOFR，帧中继语音）一种能处理语音数据的帧中继可选项。

voice over IP（IP语音）一种技术，它能将因特网用作电话网。

W

Walsh table（Walsh表）在CDMA中，用于产生正交序列的二维表。

wave division multiplexing（WDM，波分复用）将多个光信号调制为一个信号的组合过程。

wavelength（波长）简单信号在一个周期内传播的距离。

weak collision（弱碰撞）已给定摘要，建立具有相同摘要的第二个报文。

web page（网页）在Web上可得到的超文本或超媒体单元。

weighted fair queueing（加权公平队列）一种用来改善QoS的分组排列技术。应用该技术，基于给定的优先权编号对分组进行排队。

well-known port number（熟知端口号）在服务器上标志一个进程的端口编号。

wide area network（WAN，广域网）一种网络，使用可跨越很大的地理距离的技术。

wide area telephone service（WATS，广域电话服务）一种按所拨打的电话次数进行收费的电话服务。

World Wide Web（WWW，万维网）一种多媒体因特网服务。该服务允许用户通过因特网，从一个文档经过链接它们的链路精彩到另一个文档。

X

X.25 一种ITU-T标准，它定义了在数据交换终端设备和分组交换网络之间的接口。

X.509 一种用于公钥基础设施的ITU-T标准。

Y

Yahoo

Z

zone（区域）在DNS中，一个服务器所管辖的区域，或对其拥有授权的区域。