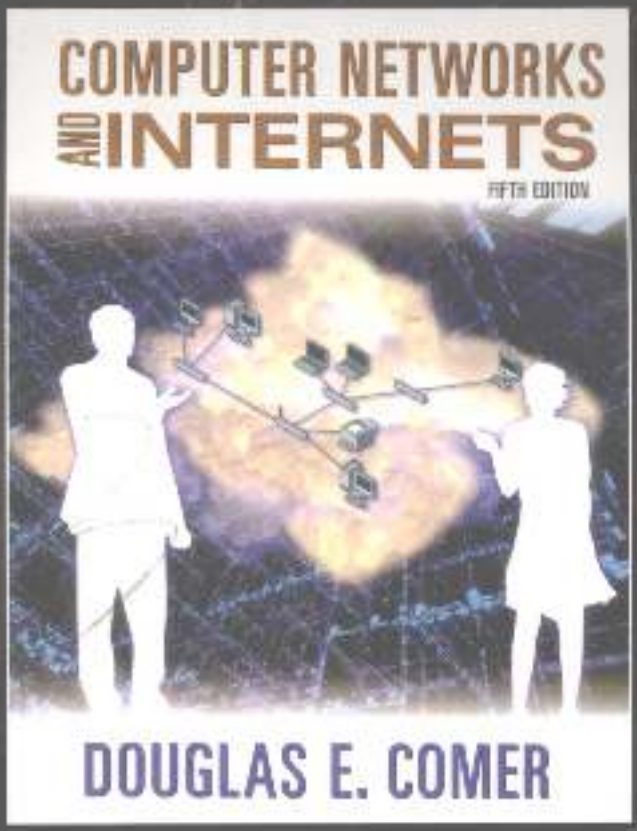
计 算 机 科 学 丛 书

原书第5版

计算机网络与因特网

(美) Douglas E. Comer 著 普渡大学
林 生 范冰冰 张奇支 黄兴平 译 华南师范大学
林 生 审校 华南师范大学



Computer Networks and Internets

Fifth Edition

机械工业出版社
China Machine Press

计算机网络与因特网（原书第5版）

"对初学者和专业人士来说，本书都是一本很好的书……写得好，综合而完，易于理解。"
——Lin Liu，英年实验室

"早……在我准备CCIE考试的时候，本书的出现解释解决了我的所有问题，使我终于搞懂了OSI模型和TCP/IP协议，它打开了计算机通信迷人的网络和TCP/IP世界的大门。"
Simon Tam，香港半导体公司

"拿到本书系统地学习关于计算机网络的理论知识，这事与愿违是我生命中……"
——Lan Y Jian，印度Rajarshi工程学院

国际公认的TCP/IP协议和因特网专家、互联网的先驱者之一Douglas Comer博士以独树一帜的方法把准确的技术知识和当前网络的研究热点完美结合，讲述了网络的底层技术和联网技术。作者在最广泛的意义上回答了"计算机网络和互联网是如何工作的"这个基本的问题。本书通过描述底层细节、网络技术、网络互联协议和应用软件等全面的联网知识，给读者提供了综合性的知识大观。本书第5版已经重新组织和全面修订，新增了无线网络协议、网络性能等一系列最新的热点技术话题。

本书的Web网站含有大量有助于教学和帮助学生理解的材料，包括课程资料、来自课文中的插图和帮助澄清概念的动画图片。网站也包含了书中没有的一些内容，比如网络布线和设备的照片，以及能用作学生作业题的数据文件。详情请登录 http://www.netbook.cs.purdue.edu。

本书特色

- 补充数据通信方面的材料，纳入书中的数据通信部分，介绍了信息源和信号、传输、调制、调制解调器、复用与解复用、接入与互连技术。
- 新增网络性能一章，包括QoS和区分服务。
- 扩充无线联网技术的篇幅，包括蓝牙、Wi-Fi、WiMax和802.11 2007协议。
- 新增网络技术及发展趋势两章内容，涵盖了虚拟化、社区网络应用等内容。
- 介绍蜂窝电话网络及其标准，帮助读者了解蜂窝移动通信如何采纳使用因特网协议。
- 介绍如何通过实时传输协议进行多媒体传输。

作者简介



Douglas E. Comer 美国普度大学教授，著名的网络技术专家，他每年都要向学生、专业人士等讲授计算机网络和Internet课程。他编写的《TCP/IP网络互联》（3卷本）、《计算机网络和因特网》等都是非常受欢迎的著作。他是对20世纪70年代末期和80年代因特网的形成有杰出贡献的研究人员之一，他还供职于因特网结构委员会（负责指导因特网发展的团体），是美国计算机学会的会员。

www.PearsonEd.com
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
ISBN 978-7-111-26831-4

定价：55.00元（附光盘）

计 算 机 科 学 丛 书

原书第5版

TP393/59=5D
2009

计算机网络与因特网

(美) Douglas E. Comer 著 林 生 范冰冰 张奇支 黄兴平 译 林 生 审校
普渡大学 华东师范大学 华东师范大学



Computer Networks and Internets

Fifth Edition

机械工业出版社
China Machine Press

本书系统介绍计算机网络各方面知识，全面翔实地诠释网络底层细节。本书在前一版的基础上增加了网络新技术内容。随书光盘包含大量相关代码和实例，方便读者实践练习。本书适合作为高等院校计算机、通信、电子等专业的教材或参考书。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: Computer Networks and Internets(ISBN 978-0-13-606127-4) by Douglas E. Comer , Copyright ©2009 .

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，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：01-2009-1351

图书在版编目（CIP）数据

计算机网络与因特网（原书第5版）/（美）科默（Comer, D. E.）著；林生等译. —北京：机械工业出版社，2009.7

（计算机科学丛书）

书名原文：Computer Networks and Internets, Fifth Edition

ISBN 978-7-111-26831-4

Ⅰ.计… Ⅱ.①科…②林… Ⅲ.①计算机网络—教材②互特网—教材 Ⅳ.TP393

中国版本图书馆CIP数据核字（2009）第068325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：王 霞

北京诚信伟业印刷有限公司印刷

2009年6月第1版第1次印刷

184mm×260mm · 23印张

标准书号：ISBN 978-7-26831-4

ISBN 978-7-89451-061-7（光盘）

定价：55.00元（附光盘）

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

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅奠定了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作，而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

华章教育

对本书的热情评价

"本书是我读过的最好的书之一。"

——Gokhan Mutlu，土耳其Ege大学

"拿到本书后我儿乎是爱不释手地读完的，这本书实在是太出色了！"

Lalit Y.Raju，印度Regional工程学院

"对初学者和专业人士来说，本书都是一本极好的书——写得好，综合面宽，易于理解。"

——John Lin，贝尔实验室

"本书内容涵盖之广真是令人……"

——George Verghese，美国加州大学圣地亚哥分校

"这真的是我看见过的网络书中最好的！"

——Chez Ciechanowicz，英国伦敦大学信息安全组

"附录的Web服务器小模型太绝妙了——读者看到这里都会激动不已。"

Dennis Brylow，美国马凯特大学

"哇！真是一本极好的教科书！"

——Jaffet A.Cordoba，本书作者

"这本书相当出色！"

——Peter Parry，英国南伯明翰学院

"哇！在我准备CCNA考试的时候，本书的清晰解释解答了我的所有问题，使我终于搞懂了OSI模型和TCP/IP传输。它打开了使我通向迷人的网络和TCP/IP世界的记忆之门。"

——Solomon Tang，香港电信公司

"一个非常宝贵的工具，特别是对于需要有更广泛地理解计算机网络的那些程序员和计算机科学工作者来说。"

——Peter Chuks Obiefuna，美国朱克罗莱的大学

"本书涵盖了大量的内容，而且作者把内容写得易读易懂，这就是我喜欢这本书的最大理由。它非常适合作为3学分课程的教科书。学生的正面反馈也无疑明，他们确实太喜欢使用这本教科书了。"

——Jie Hu，美国圣克劳得州立大学

"尽管网络技术中充斥着太多的缩写词，甚至多到了使乱人开目的地步，但本书却使人心明眼亮。Comer是一位出色的作者，他扩展并解释了很多术语。本书涵盖了从布线到整个Web网络范围的大量内容。这本书的确很出色。"

——Jennifer Seitzer，美国代顿大学

译者序

本书作者Douglas E. Comer博士是一位在TCP/IP协议、计算机联网和因特网方面国际上公认的资深网络专家，他在上世纪七八十年代因特网发展过程中作出过很大的贡献，当时他是因特网体系结构委员会的一个成员，负责指导因特网发展的工作组。他是美国普度（Purdure）大学的计算机科学教授，他除了在本校讲授课程并进行计算机联网、网络互联和操作系统方面的研究工作外，每年还要在世界各地讲授很多网络专业方面的选修课程。他编写出版了一系列畅销的技术书籍（已经被翻译成16种语言），尤其是网络课程的教科书在国际上颇具影响。

本书是作者的代表作之一，前面曾有过4个版本，这次出版的是最新的第5版，以前的几个版本都已经产生了很好的教学效果，除了有几百所美国学校使用它作为网络课程的教科书外，在美国之外的其他国家和地区也被翻译成多种语言作为高校的教科书使用，获得了很多赞誉。在当前网络书籍供过于求的市场中，能获得如此成功确实难能可贵。

本书能从众多网络书籍中脱颖而出，主要在于书中内容涵盖广泛，组织结构逻辑性强，概念解释清晰透彻，重点讲述因特网，兼顾到教师和学生的双重需求，正如美国贝尔实验室的一位教授评价的："对初学者和专业人士来说，本书都是一本很好的书——写得好，综合面宽，易于理解。"

基于目前网络领域的发展和变化，作者又一次对本书全面完成了新版本的组织、设计和更新。主要的内容更动包括：删减了对较老技术的阐述，对数据通信方面的基础内容进行了充实并纳入到本书的第二部分；在数据通信基础上再讲述有关联网方面的知识，而且对有线的和无线的联网技术都做了介绍和描述；强调了新的802.11无线联网标准，还引入了蜂窝电话技术，因为目前的蜂窝移动通信系统提供数据业务，并且很快会采纳使用因特网协议。

在内容结构方面，本书组合了"自底向上"和"自顶向下"这两种方法各自的优点，以讨论网络应用以及因特网提供的通信规范开始，让学生在学习网络设施的底层技术之前，先去理解因特网的应用设施，在讨论了应用之后再介绍连网知识，并且用富含逻辑的手法，介绍新技术是如何构筑在较低层技术基础之上的。从而在最广泛的意义上回答了"计算机网络和互联网是如何工作的？"这个最基本的问题。

本书的新版面世后，译者有幸再次受出版社之托翻译了本书最新版，并向广大读者（尤其是各高校教师和学生）推荐这本书，本书适合于高年级本科生或低年级研究生作为课堂教学使用，也可作为一般读者进修网络专业知识的自学和培训教材。

本书的全部内容由华南师范大学计算机学院的多位博士和教授共同翻译完成。为保证高质量的翻译水平，译者们在用词和语句连贯性上反复推敲，最后由一名具有多年丰富翻译经验的教授（也是前一版本的译者）进行全面审校和文字统稿。其中，范冰冰教授翻译了第1、2、3、4、30、32章以及附录；黄兴平博士翻译了第5、6、7、8、9、10、11、12、16、17、18、19、28、31章；张奇支博士翻译了第13、14、15、20、21、22、23、24、25、26、27、29章。林生教授翻译了前言和评价，并承担全书内容的审校和全面的统稿工作。尽管审、译者们对于本书的翻译质量保证方面有很强的自信心，也难免仍会出现少许瑕疵，如有不妥之处，敬请读者批评指正。

译者

于 广州华南师范大学计算机学院

2008年12月

前言

本书上一版我很惊喜地收到了很好的评价，在此特别要感谢花时间给我写信的那些读者，除了采用本书作为课本的学生外，还有联网方面的专家也写信肯定本书叙述透彻明了，还述说了本书如何帮助他们通过资格考试。我还收到了对本书的外文翻译版的许多热情洋溢的评价。本书能够在当前供过于求的网络书籍市场中获得如此成功，是件特别令人欣慰并且感到满足的事情。本书能脱颖而出的关键在于涵盖的内容广泛，组织结构的逻辑性强，概念的解释清晰透彻，重点讲解点特网，兼顾了师与生的共性需求。

为回应读者的建议，并鉴于目前网络领域的发展和变化，新版已经全部重新组织、设计和更新，删减了对陈旧技术的阐述。由于数据通信方面的材料越来越成为网络课程的重要基础内容，所以充实了这部分内容并编入本书的第二部分。在数据通信基础上再讲述联网方面的知识，而且对有线和无线联网技术都做了介绍和描述。此外，为突出新的802.11无线联网标准，新版对无线联网的讨论还包括了蜂窝电话技术，因为目前的蜂窝移动通信系统可提供数据业务，并且会很快兼容因特网协议。

现阶段关于网络课程安排的讨论，出现了两种方法上的争论，即“自底向上”的方法和“自顶向下”的方法。在自底向上的方法中，学生先学习最底层的细节，然后依次学习相邻的更高层如何利用较低层提供扩展的功能。而在自顶向下的方法中，学生先开始学习高层应用，并学习足够的较底层的知识以理解应用如何才能工作。本书组合了这两种方法的优点。从讨论网络应用以及因特网提供的通信规范开始，让学生在学习网络设施的底层技术之前，先理解因特网的应用设施。在讨论了应用之后再介绍联网知识，并且用富含逻辑的手法，使读者去理解在较低层技术基础之上如何构筑每一种新技术。

本书的读者对象是高年级本科生和低年级研究生，他们或许只有少许或没有联网方面的背景知识。本书既没有使用复杂的数学公式，也不需要操作系统方面的前导知识，而旨在清晰地阐述概念，采用实例并提供大量技术原理的示意图，分析并说明结论但不提供数学证明。

本书在最广泛的意义上回答了“计算机网络和互联网是如何工作的？”这个基本的问题。通过阐述底层细节（如数据传输和布线）、网络技术（如局域网和广域网）、网络互联协议和应用软件等各面的联网知识，奉献给读者以综合性的知识大观，还阐明了协议如何利用底层硬件，以及应用程序如何使用协议栈为用户提供各种服务功能。

本书分为五大部分，第一部分（第1~4章）集中介绍点特网的应用和网络应用开发，阐述协议分层、客户-服务器交互模式、套接字API，并列举了因特网中应用层协议的例子。

第二部分（第5~12章）介绍数据通信技术，底层硬件和调制、复用、信道编码等原理的背景知识。这几章中还讨论传输模式，并定义一些术语，如“带宽”、“波特”等。本部分最后一章里介绍因特网中使用的接入和互连技术，并阐述如何利用前面章节提到的概念来实现每一种技术。

第三部分（第13~19章）重点讲述分组交换技术。这部分先解释采用分组传输数据的动机和起因，介绍协议第2层的IEEE模型，然后再探求有线和无线联网技术。这部分内容也介绍网络的4个基本分类：局域网、城域网、个域网和广域网，并讨论广域网的路由技术。这部分的

最后一章介绍了互特网中已经应用的网络技术的例子。

第四部分（第20-27章）重点讨论因特网中协议。在讨论了网络互联的由来之后，这部分描述互特网结构、路由器、因特网编址、地址绑定和TCP/IP协议组。对其中的一些协议（如IP、TCP、UDP、ICMP和ARP）做了更详细的讲解，帮助学生更深入地理解这些概念是如何联系到实际中的。第26章是讲TCP协议的，它涵盖了传输协议中可靠性方面的重要而深层次的课题。

本书的最后一部分（第28-32章）内容涉及协议栈多层次交叉的一些课题，包括：网络性能、网络安全、网络管理、网络软件自举和多媒体支持等。这些课题也都是从前面的各个部分内容中提取出来的，安排到本书的最后这部分对它们的概念进行定义，但并不表示这些课题不重要。

本书很适合作为网络导论性的课程教材，可供初级至高年级学生一学期使用。本书按综合性课程的要求来设计，涵盖了从布线到应用的全部知识点。我鼓励教师要给学生布置一些课后作业，例如在美国普度大学的本科课程中，学生每周都有覆盖综合知识点的实践作业：网络测量、分组分析以及网络编程等。等到学生学完课程之后，期望每个学生能够达到以下目标：知道IP路由器如何利用路由表转发IP数据报，能描述数据报如何通过因特网传输，会解释以太网集线器与以太网交换机的区别；知道TCP如何标识连结，为什么一个并发Web服务器能在80号端口处理多个连结，会计算在千兆以太网上传输的单个码位的长度，能解释为什么TCP协议被归类为端到端协议；知道DSL为何在导线上发送数据的同时还能进行模拟电话通信。

一门课程的主要目标是知识的广度而不是深度——要涵盖所有主题，而非集中在几种技术或几点概念上。因此，授课结果的好坏取决于能否快速地讲好这门课。为了能使学生在一个学期内学到基本内容，可以把第二部分的较低层次内容压缩在1周学完；把有关网络和互联的部分备妥安排在4周内学完，余下2周留给应用和一些专题部分（如网络管理与安全）作介绍性讲解。至于套接字编程的细节问题，可以留在编程练习题中。

教师应该对学生强调概念和原理的重要性：有些技术可能在几年后就会过时，但原理却是永恒的。此外，教师也应该激发学生投身到联网技术中的热情。

尽管本书没有涵盖高难度的知识点，但学生们会发现书中资料的数量还是有点让人望而生畏的。特别是学生们要面对过多的新术语，书中的网络缩写字和术语也特别容易引起混淆，因此要求学生必须花费大量时间养成使用正确术语的习惯。在普度大学的课堂上，我们发现每周的词汇测验有助于加深学生对术语的理解。

因为编程和实验是帮助学生掌握网络知识的重要环节，所以实践环节是所有联网课程[⊖]的重要组成部分。我们普度大学的课程更强调分组分析和套接字编程。在学期初，我们先让学生构建客户软件去访问Web并提取数据（例如编写一个程序并打印出当前的温度）。附录可以帮助学生入手：这个附录阐述了一个简化的API，它可用在Web网站上，并允许学生在不用协议、地址和套接字API之前就能编写可执行的代码。当然，到了学期末，学生就学会了套接字编程。最后，他们还要编写一个并发Web服务器程序（支持服务器端脚本部分可选，大多数学生能够完成）。除了应用编程外，学生还可以利用实验室设施从正在运行的网络上捕获数据分组，并编写程序对分组（例如以太网帧、IP数据报、TCP段）的头部进行解码，并观察

⊖ 可以采用实验手册《Hands on Networking》，它给出了一些实验题目和作业题。这些实验涉及需要在各种各样的硬件（包括一台计算机或局域网上的一组计算机）上才能完成。

TCP连结的情况。假如不具备先进的实验室设施，可以让学生使用免费软件（例如Ethereal）来做实验。

让学生去接触真实网络能够激励其对实践的热情和信心——我们的经验表明：凡是接触过网络现场的学生都能更好地理解学习主题，有更强的辨别能力。所以，如果没有专门的分组分析器，可以在一台标准PC上安装合适的共享软件来创建一个分析器。

本书的配套网站含有大量有助于教学和帮助学生理解的材料。针对不具备接触联网设施条件的学生，本网站收录了一些分组跟踪的例子，学生可以编写程序去读取分组的传输踪迹并处理分组，就好像已经从网络上捕捉到这个分组似的。从教师的角度来说，该网站还包含的课程资料和来自课文中的插图，可用来制作演示文档，还有动画图片更可帮助澄清概念。网站也包含了书中没有的一些内容，比如网络布线和设备的照片，以及用于学生作业题输入的数据文件。网站地址是：

http://www.netbook.cs.purdue.edu

我要感谢所有为本书新版作出贡献的人们。Cisco公司的Fred Baker和Dave Oran提出了重要主题的建议。Lami Kaya对本书的整体构思和组织提出了宝贵意见，并帮助系统修订了数据通信各章的内容，审阅了全书内容，以及提出了具有很多极有价值的建议。Lami还负责管理本书的网站。我还要特别感谢我的妻子和合作者Christine，她细心的编辑和很多建设性意见使全书增色不少。

Douglas E. Comer

2008年3月

目　录

出版者的话
对本书的热情评价
译者序
前言

第一部分　引论及因特网应用

第1章　导论和概述 ……2
1.1　计算机网络的发展过程 ……2
1.2　联网为何变得复杂 ……2
1.3　联网的5个关键方面 ……3
1.4　因特网的公网和专网 ……5
1.5　网络，可互操作性和标准 ……6
1.6　协议组和分层模型 ……6
1.7　数据如何通过各个层次 ……7
1.8　头部和各层 ……8
1.9　ISO与OSI七层参考模型 ……9
1.10　关于模型的六个观点 ……9
1.11　本书内容简介 ……9
1.12　本章小结 ……10
练习题 ……10
第2章　因特网的发展趋势 ……11
2.1　引言 ……11
2.2　资源共享 ……11
2.3　因特网的成长 ……11
2.4　从资源共享到通信 ……13
2.5　从文本到多媒体 ……13
2.6　目前的发展趋势 ……13
2.7　本章小结 ……14
练习题 ……14
第3章　网络应用与网络编程 ……16
3.1　引言 ……16
3.2　因特网的基本通信模式 ……16
3.3　面向连接的通信 ……17
3.4　客户-服务器交互模式 ……17
3.5　客户和服务器特征 ……18
3.6　服务器程序和服务器类计算机 ……18
3.7　请求、响应和数据流方向 ……19
3.8　多客户与多服务器 ……19
3.9　服务器的标识与识别 ……20
3.10　并发服务器 ……20
3.11　服务器间的循环依赖 ……21
3.12　P2P交互 ……21
3.13　网络编程与套接字API ……21
3.14　套接字、描述符和网络I/O ……22
3.15　函数与套接字API ……22
3.16　客户和服务器中的套接字调用 ……22
3.17　客户和服务器共用的套接字函数 ……23
3.18　仅供客户使用的connect函数 ……24
3.19　仅供服务器使用的套接字函数 ……24
3.20　采用报文模式的套接字函数 ……26
3.21　其他套接字函数 ……27
3.22　套接字、线程和继承性 ……27
3.23　本章小结 ……28
练习题 ……28
第4章　传统的因特网应用 ……30
4.1　引言 ……30
4.2　应用层协议 ……30
4.3　表示与传输 ……30
4.4　Web协议 ……31
4.5　HTML文档表示法 ……31
4.6　统一资源定位符和超级链接 ……33
4.7　用HTTP传输Web文档 ……33
4.8　浏览器中的高速缓存 ……35
4.9　浏览器结构 ……36
4.10　文件传输协议 ……36
4.11　FTP通信模式 ……37
4.12　电子邮件 ……38
4.13　简单邮件传输协议 ……39
4.14　ISP、邮件服务器和邮件访问 ……40
4.15　邮件访问协议 ……41

4.16 电子邮件表示标准 ……41
4.17 域名系统 ……42
4.18 www开头的域名 ……43
4.19 DNS层次结构和服务器模型 ……44
4.20 域名解析 ……45
4.21 DNS服务器中的缓存处理 ……45
4.22 DNS记录项的类型 ……46
4.23 别名和CNAME资源记录 ……46
4.24 端号与DNS ……47
4.25 国际化域名 ……47
4.26 可扩展表示 ……48
4.27 本章小结 ……48
练习题 ……49

第二部分 数据传输

第5章 数据通信概述 ……52
5.1 引言 ……52
5.2 数据通信所涉及的学科 ……52
5.3 因通动机与范围 ……52
5.4 通信系统的构成 ……53
5.5 通信系统各子课题 ……54
5.6 本章小结 ……55
练习题 ……55
第6章 信息源和信号 ……56
6.1 引言 ……56
6.2 信息源 ……56
6.3 模拟与数字信号 ……56
6.4 周期信号与非周期信号 ……57
6.5 正弦波与信号特征 ……57
6.6 复合信号 ……58
6.7 复合信号和正弦函数的重要性 ……58
6.8 时域与频域表示法 ……58
6.9 模拟信号的带宽 ……59
6.10 数字信号与信号电平 ……59
6.11 波特率与比特率 ……60
6.12 数字—模拟信号转换 ……61
6.13 数字信号的带宽 ……61
6.14 信号的同步与协调 ……62
6.15 线路编码 ……62
6.16 单切原编码 ……63
6.17 模拟—数字信号转换 ……64
6.18 奈奎斯特定理与抽样率 ……65
6.19 奈奎斯特定理与电话系统传输 ……65
6.20 编码与数据压缩 ……65
6.21 本章小结 ……66
练习题 ……66
第7章 传输介质 ……68
7.1 引言 ……68
7.2 导向传输与非导向传输 ……68
7.3 按能量形式分类 ……68
7.4 背景噪声和电气噪声 ……69
7.5 双绞线 ……69
7.6 屏蔽：同轴电缆和屏蔽双绞线 ……70
7.7 双绞线分类 ……71
7.8 使用光能的介质及光纤 ……71
7.9 光纤类型及光传输 ……72
7.10 光纤与铜导线的比较 ……73
7.11 红外通信技术 ……73
7.12 点对点激光通信 ……73
7.13 电磁波（无线电）通信 ……74
7.14 信号传播 ……74
7.15 卫星类型 ……75
7.16 GEO通信卫星 ……75
7.17 GEO卫星的覆盖 ……76
7.18 LEO卫星与群集 ……76
7.19 介质类型之间的权衡 ……77
7.20 对传输介质的度量 ……77
7.21 噪声对通信的影响 ……77
7.22 信道容量的重要性 ……78
7.23 本章小结 ……78
练习题 ……79
第8章 可靠性与信道编码 ……80
8.1 引言 ……80
8.2 传输差错的3个主要源头 ……80
8.3 传输差错对数据的影响 ……81
8.4 处理信道差错的两种策略 ……81
8.5 分组码和卷积码 ……82
8.6 分组差错编码举例：单奇偶校验 ……82
8.7 分组码数学与 (n, k) 表示 ……83
8.8 汉明距离：编码强度的测量 ……83

8.9 码集中码字之间的汉明距离 ……84
8.10 差错检测与代价之间的权衡 ……84
8.11 采用纵横奇偶校验的纠错 ……84
8.12 用于因特网的16位校验和 ……85
8.13 循环冗余校验码 ……86
8.14 用硬件高效实现CRC ……87
8.15 自动重传请求（ARQ）机制 ……88
8.16 本章小结 ……88
练习题 ……88
第9章 传输模式 ……90
9.1 引言 ……90
9.2 传输模式分类 ……90
9.3 并行传输 ……90
9.4 串行传输 ……91
9.5 传输顺序：码元与字节 ……91
9.6 串行传输的定时 ……92
9.7 异步传输 ……92
9.8 RS-232异步字符传输 ……92
9.9 同步传输 ……93
9.10 字节、块和帧 ……93
9.11 等时传输 ……94
9.12 单工、半双工与全双工传输 ……94
9.13 DCE和DTE设备 ……95
9.14 本章小结 ……96
练习题 ……96
第10章 调制与调制解调器 ……97
10.1 引言 ……97
10.2 载波，频率和传播 ……97
10.3 模拟调制方案 ……97
10.4 振幅调制 ……97
10.5 频率调制 ……98
10.6 相位调制 ……98
10.7 调幅与香农定理 ……99
10.8 调制，数字输入和键控 ……99
10.9 移相键控 ……99
10.10 相移与星座图 ……100
10.11 正交调幅 ……101
10.12 调制解调器硬件 ……101
10.13 光纤和射频调制解调器 ……102
10.14 拨号调制解调器 ……102
10.15 应用于拨号的QAM ……103
10.16 V.32与V.32bis拨号modem ……103
10.17 本章小结 ……103
练习题 ……104
第11章 复用与解复用 ……105
11.1 引言 ……105
11.2 复用的概念 ……105
11.3 复用的基本类型 ……105
11.4 频分多路复用 ……106
11.5 每个信道使用一个频率范围 ……107
11.6 分级FDM ……108
11.7 波分多路复用 ……108
11.8 时分多路复用 ……109
11.9 同步TDM ……109
11.10 电话系统中TDM的成帧技术 ……109
11.11 分级TDM ……110
11.12 同步TDM的问题：空闲时隙 ……111
11.13 统计TDM ……111
11.14 旋转复用 ……112
11.15 码分多路复用 ……112
11.16 本章小结 ……113
练习题 ……114
第12章 接入与互连技术 ……115
12.1 引言 ……115
12.2 因特网接入技术：上行与下行 ……115
12.3 窄带与宽带接入技术 ……115
12.4 本地环路及ISDN ……116
12.5 数字用户线技术 ……117
12.6 本地环路特征及适配 ……117
12.7 ADSL的数据速率 ……118
12.8 ADSL安装和分离器 ……118
12.9 电缆调制解调器技术 ……119
12.10 电缆调制解调器的速率 ……119
12.11 电缆调制解调器的安装 ……120
12.12 光纤与同轴电缆混合使用 ……120
12.13 采用光纤的接入技术 ……120
12.14 头端与用户端调制解调器技术 ……121
12.15 无线接入技术 ……121
12.16 因特网核心区的高容量连接 ……122
12.17 线路终端、DSU/CSU及NIU ……122

12.18 数字线路的电话标准 ……123
12.19 DS术语及数据速率 ……123
12.20 最高容量线路 ……124
12.21 光载波标准 ……124
12.22 C后缀 ……124
12.23 同步光网络 ……124
12.24 本章小结 ……125
练习题 ……126

第三部分 分组交换及网络技术

第13章 局域网：分组、帧和拓扑 ……128
13.1 引言 ……128
13.2 线路交换 ……128
13.3 分组交换 ……129
13.4 局域的和广域的分组网络 ……130
13.5 分组标识及其格式标准 ……130
13.6 IEEE 802模型与标准 ……131
13.7 点对点与多址接入网络 ……132
13.8 LAN拓扑 ……132
13.9 分组标识、解复用、MAC地址 ……133
13.10 单播、广播和组播地址 ……134
13.11 广播、组播和高效的多点传送 ……134
13.12 帧与成帧 ……135
13.13 字节插入与位插入 ……136
13.14 本章小结 ……137
练习题 ……137
第14章 IEEE MAC子层 ……138
14.1 引言 ……138
14.2 多址接入机制的分类 ……138
14.3 静态与动态信道分配 ……138
14.4 信道分配协议 ……139
14.5 受控接入协议 ……140
14.6 随机接入协议 ……141
14.7 本章小结 ……145
练习题 ……145
第15章 有线局域网技术 ……146
15.1 引言 ……146
15.2 最早的以太网 ……146
15.3 以太网帧格式 ……146
15.4 以太网的类型域 ……147
15.5 以太网的IEEE版本 ……147
15.6 LAN连接和网络接口卡 ……148
15.7 粗缆为线的以太网 ……148
15.8 细缆布线的以太网 ……148
15.9 双绞线布线的以太网和集线器 ……149
15.10 以太网的物理和逻辑拓扑 ……149
15.11 办公大楼内的布线 ……150
15.12 双绞线以太网的变种及其速率 ……151
15.13 双绞线连接器与缆线 ……151
15.14 本章小结 ……152
练习题 ……152
第16章 无线局域网技术 ……153
16.1 引言 ……153
16.2 无线网络的分类 ……153
16.3 个域网 ……153
16.4 LAN和PAN使用的ISM无线频谱 ……154
16.5 无线LAN技术与Wi-Fi ……154
16.6 扩频技术 ……154
16.7 其他无线LAN标准 ……155
16.8 无线LAN体系结构 ……155
16.9 重叠、关联和802.11帧格式 ……156
16.10 接入点之间的协调 ……157
16.11 竞争与无竞争接入 ……157
16.12 无线MAN技术与WiMAX ……158
16.13 PAN技术与标准 ……159
16.14 其他短距离通信技术 ……160
16.15 无线WAN技术 ……161
16.16 基站集群和频率重用 ……162
16.17 蜂窝技术的更新换代 ……163
16.18 VSAT卫星技术 ……164
16.19 GPS卫星 ……165
16.20 软件无线电和无线电的未来 ……165
16.21 本章小结 ……166
练习题 ……167
第17章 局域网扩展技术 ……168
17.1 引言 ……168
17.2 距离限制与LAN设计 ……168
17.3 光纤调制解调器扩展 ……168
17.4 中继器 ……169
17.5 网桥与桥接 ……169

17.6 自学习网桥与帧过滤 ……170
17.7 为什么桥接能行 ……170
17.8 分布式生成树 ……171
17.9 交换与第二层交换机 ……172
17.10 虚拟局域网交换机 ……173
17.11 使用其他设备实现桥接 ……173
17.12 本章小结 ……174
练习题 ……174
第18章 广域网技术与动态路由 ……175
18.1 引言 ……175
18.2 大型广域网络 ……175
18.3 传统的广域网体系结构 ……175
18.4 广域网的构成 ……176
18.5 存储/转发模式 ……177
18.6 广域网的编址与寻址 ……177
18.7 下一跳转发 ……178
18.8 源点独立性 ……179
18.9 广域网动态路由更新 ……180
18.10 默认路径 ……181
18.11 转发表的计算 ……181
18.12 分布式路径计算 ……181
18.13 基于最短路径的计算 ……184
18.14 路由问题 ……185
18.15 本章小结 ……185
练习题 ……186
第19章 网络技术的过去与现在 ……187
19.1 引言 ……187
19.2 连接与接入技术 ……187
19.3 LAN技术 ……188
19.4 WAN技术 ……189
19.5 本章小结 ……191
练习题 ……191

第四部分 网络互联

第20章 网络互联：概念、结构与协议 ……194
20.1 引言 ……194
20.2 网络互联的动机 ……194
20.3 全局服务概念 ……194
20.4 异构网络中的全局服务 ……195
20.5 网络互联 ……195
20.6 用路由器互接物理网络 ……195
20.7 互联网体系结构 ……196
20.8 实现全局服务 ……196
20.9 虚拟网络 ……196
20.10 网络互联协议 ……197
20.11 TCP/IP分层结构综述 ……197
20.12 主机、路由器及协议层 ……198
20.13 本章小结 ……198
练习题 ……199
第21章 网际协议编址 ……200
21.1 引言 ……200
21.2 每个连接网络的地址 ……200
21.3 IP编址方案 ……200
21.4 IP地址的层次结构 ……201
21.5 IP地址的原分类 ……201
21.6 点分十进制数表示法 ……202
21.7 地址空间的划分 ……203
21.8 地址的授权 ……203
21.9 子网与无类编址 ……203
21.10 地址掩码 ……204
21.11 CIDR表示法 ……205
21.12 CIDR举例 ……206
21.13 CIDR主机地址 ……206
21.14 特殊的IP地址 ……207
21.15 小结特殊IP地址 ……208
21.16 为克利广播地址形式 ……208
21.17 路由器与IP寻址原理 ……208
21.18 多穴主机 ……209
21.19 本章小结 ……209
练习题 ……210
第22章 数据报转发 ……211
22.1 引言 ……211
22.2 无连接服务 ……211
22.3 虚拟分组 ……211
22.4 IP数据报 ……212
22.5 IP数据报头部格式 ……212
22.6 IP数据报转发 ……213
22.7 网络前缀提取与数据报转发 ……214
22.8 最长前缀匹配 ……214
22.9 目的址与下一跳地址 ……215

22.10 尽力传递 ……215
22.11 IP封装 ……216
22.12 通过因特网传输 ……216
22.13 MTU和数据报分片 ……217
22.14 分片数据的重装 ……218
22.15 分片数据报的收集 ……219
22.16 片丢失的后果 ……219
22.17 分片再分片 ……219
22.18 本章小结 ……219
练习题 ……220
第23章 支持协议与相关技术 ……221
23.1 引言 ……221
23.2 地址解析 ……221
23.3 地址解析协议 ……222
23.4 ARP报文格式 ……222
23.5 ARP封装 ……223
23.6 ARP缓存与报文处理 ……224
23.7 概念地址边界 ……225
23.8 互联网控制报文协议 ……225
23.9 ICMP报文格式与封装 ……226
23.10 协议软件、参数与配置 ……227
23.11 动态主机配置协议 ……227
23.12 DHCP协议操作与优化 ……228
23.13 DHCP报文格式 ……229
23.14 通过中继代理访问DHCP服务器 ……230
23.15 网络地址转换 ……230
23.16 NAT操作与私有地址 ……231
23.17 传输层NAT ……232
23.18 NAT与服务器 ……233
23.19 实用NAT软件和系统 ……233
23.20 本章小结 ……233
进一步的阅读资料 ……234
练习题 ……234
第24章 未来的IP：IPv6 ……236
24.1 引言 ……236
24.2 IP成功之处 ……236
24.3 改革的动机 ……236
24.4 沙漏模型与改革的难点 ……237
24.5 名称和版本号 ……237
24.6 IPv6的特性 ……238
24.7 IPv6数据报格式 ……238
24.8 IPv6基本头部的格式 ……239
24.9 地址和量式头部长度 ……240
24.10 分片、重装和通路MTU ……240
24.11 采用多重头部的目的 ……241
24.12 IPv6编址 ……241
24.13 IPv6冒分十六进制数表示法 ……242
24.14 本章小结 ……243
练习题 ……243
第25章 UDP：数据报传输服务 ……244
25.1 引言 ……244
25.2 传输协议与端到端通信 ……244
25.3 用户数据报协议 ……244
25.4 无连接的通信模式 ……245
25.5 面向报文的接口 ……245
25.6 UDP通信语义 ……246
25.7 交互模式和广播传送 ……246
25.8 用协议端口号标识端点 ……247
25.9 UDP数据报格式 ……247
25.10 UDP校验和伪头部 ……247
25.11 UDP封装 ……248
25.12 本章小结 ……248
练习题 ……249
第26章 TCP：可靠的传输服务 ……250
26.1 引言 ……250
26.2 传输控制协议 ……250
26.3 TCP为应用程序提供的服务 ……250
26.4 端到端服务与虚电路连接 ……251
26.5 传输协议所采用的技术 ……251
26.6 避免网络阻塞的技术 ……254
26.7 协议设计技巧 ……255
26.8 用来对付分组丢失的技术 ……255
26.9 自适应重传技术 ……256
26.10 重传时间的比较 ……257
26.11 缓冲、流控与窗口 ……257
26.12 TCP的三次握手 ……258
26.13 TCP拥塞控制 ……259
26.14 TCP段格式 ……260
26.15 本章小结 ……260
练习题 ……261

第27章 互特网路由与路由协议 ……262
27.1 引言 ……262
27.2 静态与动态路由 ……262
27.3 主机静态路由与默认路径 ……262
27.4 动态路由与路由器 ……263
27.5 全球因特网的路由技术 ……264
27.6 自治系统概念 ……264
27.7 两类因特网路由协议 ……264
27.8 路径与数据业务 ……266
27.9 边界网关协议 ……266
27.10 路由信息协议 ……267
27.11 RIP分组格式 ……268
27.12 开放最短路径优先协议 ……268
27.13 OSPF网的例子 ……269
27.14 OSPF区域 ……270
27.15 中间系统到中间系统协议 ……270
27.16 组播路由技术 ……270
27.17 本章小结 ……273
练习题 ……273

第五部分 其他网络概念与技术

第28章 网络性能 ……276
28.1 引言 ……276
28.2 性能度量 ……276
28.3 延迟 ……276
28.4 吞吐率、容量、实际吞吐量 ……277
28.5 理解吞吐率与延迟 ……278
28.6 抖动 ……279
28.7 延迟与吞吐率的关系 ……279
28.8 测量延迟、吞吐率与抖动 ……281
28.9 被动测量、小分组及网流监测 ……282
28.10 服务质量 ……282
28.11 细粒度与粗粒度QoS ……283
28.12 QoS的实现 ……284
28.13 因特网QoS技术 ……286
28.14 本章小结 ……286
练习题 ……287
第29章 多媒体与IP电话 ……289
29.1 引言 ……289
29.2 实时数据传输和尽力而为传送 ……289
29.3 延迟抖动与抖动缓冲 ……289
29.4 实时传输协议 ……290
29.5 RTP封装 ……291
29.6 IP电话 ……291
29.7 信令与VoIP信令标准 ……292
29.8 IP电话系统的组成部件 ……292
29.9 协议及所在层次归纳 ……295
29.10 H.323特性 ……295
29.11 H.323分层 ……295
29.12 SIP特性和方法 ……296
29.13 SIP会话举例 ……296
29.14 电话号码映射及路由 ……297
29.15 本章小结 ……297
进一步的阅读资料 ……298
练习题 ……298
第30章 网络安全 ……299
30.1 引言 ……299
30.2 网络犯罪与攻击 ……299
30.3 安全策略 ……301
30.4 安全责任与控制 ……302
30.5 安全技术 ……302
30.6 数列法：完整性与鉴别机制 ……302
30.7 访问控制与口令 ……303
30.8 加密：基本的安全技术 ……303
30.9 私有密钥加密 ……304
30.10 公开密钥加密 ……304
30.11 用数字签名的鉴别 ……305
30.12 密钥分发和数字证书 ……305
30.13 防火墙 ……306
30.14 包过滤防火墙的实现 ……307
30.15 入侵检测系统 ……308
30.16 内容扫描和深度包检查 ……309
30.17 虚拟专网 ……309
30.18 VPN技术应用于远程办公 ……310
30.19 数据包加密与隧道技术 ……311
30.20 安全技术 ……313
30.21 本章小结 ……313
练习题 ……314
第31章 网络管理 ……316
31.1 引言 ……316

31.2 管理内部网 ……316
31.3 FCAPS：行业标准模型 ……316
31.4 典型的网络元素 ……317
31.5 网络管理工具 ……318
31.6 网络管理应用 ……319
31.7 简单网络管理协议 ……319
31.8 SNMP的取/存操作模式 ……320
31.9 管理信息库和对象名 ……320
31.10 MIB变量的种类 ……321
31.11 对应于数组的MIB变量 ……321
31.12 本章小结 ……322
练习题 ……322
第32章 网络技术及应用发展趋势 ……324
32.1 引言 ……324
32.2 可扩展网络服务的需求 ……324
32.3 内容缓存加速 ……324
32.4 Web负载均衡器 ……325
32.5 服务器虚拟化 ……325
32.6 P2P通信 ……325
32.7 分布式数据中心 ……326
32.8 通用表示 ……326
32.9 社区网络 ……326
32.10 移动性及无线联网 ……326
32.11 数字视频 ……327
32.12 多播传送 ……327
32.13 高速接入与交换 ……327
32.14 光交换 ……328
32.15 网络商务应用 ……328
32.16 传感器件应用 ……328
32.17 Ad Hoc网络 ……328
32.18 多核CPU和网络处理器 ……328
32.19 IPv6 ……329
32.20 本章小结 ……329
练习题 ……329
附录 一种简化的应用编程接口 ……331

第一部分

引论及因特网应用

从应用以及编写通过因特网进行通信的程序开始

第1章　导论和概述

1.1　计算机网络的发展过程

计算机网络技术已得到迅猛的成长与发展。自20世纪70年代以来，计算机通信已从深奥的研究专题演变为社会基础结构的基本组成部分之一。网络已应用于各行各业，包括广告、生产过程、货运、计划、报价和会计等。结果，绝大多数公司一般都拥有多个网络。下至小学，上至研究生教育，所有层次的学校都在利用计算机网络为教师和学生提供实时的在线信息。从联邦（国家），州（省）到各级地方政府的办公室，都在使用网络，各军事单位同样如此。简言之，计算机网络已无处不在。

全球因特网[①]的发展与应用是网络领域最令人激动和富有意义的现象。1980年，因特网还只是一个只有几十个站点的研究项目。今天，因特网已发展成为一个覆盖世界上所有居民区的大规模通信系统，许多用户通过线缆Modem、DSL以及无线技术高速接入因特网。

网络的出现和使用促使经济产生了巨大的变化，数据网络使得个体之间进行远程通信成为可能，并改变了商业交互方式。此外，一个完整的从事于研发网络技术、产品和服务的产业已经形成。计算机联网的重要性使得各行各业都需要具有更多网络知识的人才。公司需要能从事规划、获取、安装、操作、管理计算机网络系统的工作人员。此外，计算机编程已不再局限于单台计算机，而是要求进行网络编程。因为所有程序员都期望设计出并实现能够与其他计算机上的应用进程进行通信的应用软件。

1.2　联网为何显得复杂

因为计算机联网是一个非常活跃且快速发展的新兴领域，所以这个主题似乎显得有点复杂。其中存在很多技术问题，并且每种技术都有各自的特点。许多公司陆续推出了各种新的、非常规技术的商用联网产品和服务。最终，由于这些技术可以用许多方法进行组合和互连，从而使得联网的问题变得复杂起来。

对于一个初学者来说，联网的问题尤其令人困惑，因为不存在单一的基础理论来解释网络各部分的相互关系。有多个组织机构已经开发了各自的联网标准，但这些标准相互间并不兼容。各种组织和研究团体也都已经尝试定义了各种概念模型，用来描述网络硬件和软件系统之间的差异性和相似性。可惜的是，由于各系统所涉及的技术各不相同，并且变化也非常快，这些概念模型要么过于简单以致于无法区分各系统间的细节，要么过于复杂而无助于对主题的简化。

在联网领域中由于缺乏一致性，从而对初学者还产生了另一个挑战：在联网概念方面没有统一的术语，多个组织机构试图推行各自定义的概念术语，而研究者又太讲究在科学上要用精确的术语。公司的销售人员也常常把其产品或服务与通用术语联系起来，或者为了区别与其他公司的竞争产品或服务而创造一些新的术语来，所以技术术语有时还会与流行产品的名称相混淆。专业人员有时会使用一种技术中的某个术语，去表达另一技术中一个类似的概

---

① 在整本书中我们都遵循这个书写惯例，即首字母为大写I的词汇Internet，都是指全球“因特网”。介则，就是泛指“互联网”。

念，从而进一步增加了混乱。结果，除了一大堆术语和包含很多同义语的缩写词外，网络行话里还包含了一些常被简称、被误用或与产品相关的术语。

1.3 联网的5个关键方面

为了把握联网的复杂性，重要的是要获取以下5个关键方面的内容：

- 网络应用和网络编程。
- 数据通信。
- 分组交换和联网技术。
- 使用TCP/IP的网络互联。
- 附加的网络概念和技术。

1.3.1 网络应用和网络编程

用户介入的网络服务和设施都是由应用软件提供的——某一计算机上的应用程序通过网络与另一计算机上的应用程序进行通信。网络应用服务的范围很宽，包括电子邮件、文件传输、Web浏览、电话、分布数据库，以及音频和视频会议等。尽管每个应用以其各自的用户界面提供特定的服务，但所有应用都可以在一个共享网络上通信。由于支撑所有应用的低层网络的可用性，使得程序员的工作变得更加容易。程序员只需要掌握一种网络接口和一个基本的函数集即可使通过网络通信的所有应用程序都使用相同的函数集。

我们将看到，即使用户不清楚应用程序通过网络传送数据的硬件和软件这一技术，但他同样可能理解网络应用，甚至有可能编写出实现网络通信的程序代码。这似乎表明程序员一旦掌握了接口技术，其他的网络知识都不需要掌握了。然而，网络编程与传统的编程方法是相似的。虽然一名传统的程序员开发应用程序可以不懂编译、操作系统或计算机体系结构，但这些基础知识能够帮助程序员开发更可靠、更正确和更高效的程序。类似地，有了关于底层网络系统方面的知识，则可以使程序员编写出更好的程序代码。

要点小结 懂得底层网络机制和技术的程序员，能够编写出更加可靠、正确和高效的网络应用程序。

1.3.2 数据通信

术语数据通信（data communication）是指对通过物理介质（如导线、电波和光纤）实现信息传送的低层机制和技术的研究。数据通信主要应用于电气工程领域，研究如何设计和构建一个广域的通信系统。重点是研究利用物理现象实现信息传输的各种方法。因此，它的许多基本思路来自于物理学家研究的物质和能量特性，例如，用于传输高速数据的光纤，就是利用了光及光在两种不同物质间边界上反射的特性。

由于涉及许多物理概念，数据通信似乎对我们理解联网有些不相关。特别是，由于涉及有关物理现象的许多术语和概念，这个主题也似乎只对设计低层传输功能的工程师有用。例如，利用物理能量形式（如电磁辐射）携带信息的调制技术，就与设计和使用协议无关。可是我们会看到，来自数据通信的几个关键概念将影响许多协议层的设计。在调制技术中，带宽概念就直接跟网络吞吐量有关。

数据通信中引入的多路复用技术可作为特例，它是指来自多个源的信息经过组合通过共享介质传输，然后再拆分传递到多个目的地。我们将看到，多路复用并不局限于物理传输，

大多数协议也都吸取了多路复用的某种形式[⊖]。类似地，引入到数据通信中的加密概念也形成了大多数网络安全的基础。因此，对数据通信的重要性归纳如下：

尽管数据通信涉及许多底层的细节，但却为构建网络的其他方面提供了概念基础。

1.3.3　分组交换和联网技术

在20世纪60年代，一个新的概念使得数据通信发生了一场革命：分组交换。早期的通信网络从电话和电报通信系统中演变而来，这两种原始的系统只是利用一对物理导线将通信双方连接起来形成一条通信线路。虽然这种导线的机械连接已经被电子交换所替代，但是底层的结构模式仍然维持原样，即形成一条通信线路，然后通过该线路传送信息。分组交换从根本上改变了联网方法，并奠定了现代因特网的基础——分组交换使得多个通信方通过一个共享的网络传送数据，而不是形成一条条专用的通信线路。尽管分组交换建立在与电话系统相同的最基本的通信原理上，但它却是以一种崭新的方式来利用底层的通信机制。分组交换把数据划分成许多小的数据块（称为“分组”），并在每个分组中加进目标接收方的标识信息。遍布网络的所有交换设备都保存有分组如何抵达所有可能目的地的有关信息。当一个分组到达任一个交换设备时，该设备就会选择一条路径，分组沿着这条路径被最终传送到正确的目的地。

理论上，分组交换原理是简单而直观的，但却可能存在很多设计上的考虑，这取决于对一些基本问题的解决。目的地址如何标识，发送方又怎样发现目的地址？分组长度是多大？网络如何辨认一个分组的尾部和另一个分组的头部？如果多台计算机同时发送数据分组，它们如何进行协调以确保它们得到公平的发送机会？分组交换如何适应于无线网络？如何设计联网技术以便满足在速率、传输距离和经济性等不同方面的要求？为此，人们提出过许多解决方案，并研发了许多种分组交换技术。其实，人们在研究分组交换网络的时候，就可以勾画出一个基本结论来：

因为开发原一种网络技术都是为了满足在速率、距离和经济成本等方面的不同要求，因而就存在着多种分组交换技术。各种技术在细节（如分组长度、寻址方法）上是有差别的。

1.3.4　使用TCP/IP的网络互联

20世纪70年代，计算机网络发生了另一场革命——因特网的构思。研究分组交换的许多专家一直努力寻求一种能满足所有需求的单一分组交换技术。1973年，Vinton Cerf和Robert Kahn注意到：没有任何一种单一的分组交换技术可以完全满足所有的需求，尤其是要以极低的成本为家庭或办公室建造小容量网络方面的需求。他们建议：停止对寻求单一的最佳解决方案的尝试，转而去探索把多种分组交换网络互联成一个有机整体的方法。经他们建议，出现了一套专门为这种互联而研发的标准，并最终产生了被大家所熟知的TCP/IP互联网协议簇（通常简称为TCP/IP）。现在称为网络互联（internetworking）的这个构思具有极其强大的影响力，它奠定了全球因特网（Internet）的基础，而且已成为计算机网络研究领域的重要组成部分。

TCP/IP标准获得成功的主要原因之一，就在于解决了异构兼容问题。TCP/IP不是试图去强制规定分组交换技术的细节（如分组长度、寻址方法等），而是采取一种虚拟化的手法，即定义一种与网络无关的分组格式和一种与网络无关的寻址方案，然后规定虚拟分组如何映射到每一种可能的低层网络上。

⊖ 例如，网络层和传输层协议中的连接复用，就是吸取了多路复用的某种形式。——译者注

有趣的是，TCP/IP能够容纳新的分组交换技术的能力，变成了分组交换技术持续发展的主要动力。随着因特网的发展壮大，计算机能力也变得更加强大，各种应用（特别是图形、图像和视频系统）将发送更多数据，为了适应这种应用发展，工程技术人员创造了在给定时间内传输更多数据和处理更多分组的许多新技术。除了继续延用和扩展已有技术外，这些新技术也随时被融合到因特网之中。也就是说，由于因特网能够兼容异构性，所以工程技术人员就可以不断试验各种新的联网技术，而不用担心破坏现有网络。

小结 因特网是通过互连多种分组交换网络而形成的。由于互联网允许随时纳入各种新技术而不必替换旧的技术，所以网络互联技术比单一联网技术具有更加强大的威力。

1.4 因特网的公网和专网

虽然因特网只是起到一个通信网络的单一功能，但它却是由很多个体或组织所拥有和运行的各部分网络组成的。为了分清拥有权和用途，网络界使用术语公网（public network）和专网（private network）来区分。

1.4.1 公网

公网是为签约用户提供服务的网络，任何支付签约费用的个体或团体都能够使用公网。提供通信服务的公司称为服务提供商（service provider）。服务提供商的含义很宽，可延伸到因特网服务提供商（ISP）。其实，这个术语起源于提供模拟电话服务的公司。

小结 公网由服务提供商所拥有，并且为任何支付签约费用的个体或组织提供服务。

所谓"公"（public）是指网络服务的公众可用性，而不是针对传输的数据而言，理解这点很重要。特别要强调的是，许多公网都必须遵守严格的政府规章，并要求服务提供商能保护通信不被窃听。

要点 "公"意味着对于一般公众用户的服务可用性，通过公网传输的数据不应对外泄露。

1.4.2 专网

专网是由某个特殊团体所控制的网络。这看起来虽然是很简单的事，但由于控制权并不总是意味着拥有权，所以因特网的公网和专网之间的差别可能是很微妙的。例如，如果某公司从运营商那里租用了一条数据线路，然后限制该线路只能用于该公司的通信业务，那么该线路就成为该公司专网的一部分了。

要点 如果网络只限于供某个团体使用，则称该网络是专用的。专网可以包含从运营商那里租用的线路。

网络设备供应商把专网划分为4类：

- 消费者网，
- 小型办公/家庭办公网（SOHO），
- 中小型商务网（SMB），
- 大型企业网。

由于以上分类关系到销售和市场，所以术语定义得比较宽松。虽然有可能给出对每个类型的定性描述，但很难找到准确的定义。因此，下面的几段内容只是给出这些类型在规模和用途方面的大致特征，而不是对它们进行详细的剖析。

消费者网：这是由个人拥有的LAN所构成的、花费最小的一种专网，即个人购买廉价LAN交换机，并将PC、打印机连接在一起，就构建了一个专网。类似地，消费者个人也可以购置和安装无线路由器来组建一个专网。

小型办公/家庭办公网（SOHO）：SOHO网络大于消费者网。一个典型的SOHO网由2~3台计算机、一台或多台打印机、一台连接因特网的路由器以及可能的其他设备（如收银机）所构成。大多数SOHO网都配置有后备电池供电和保证其不中断运行的其他机制。

中小型商务网（SMB）：SMB网可以连接建筑物中多个办公室里的许多计算机，也包括生产设施中（如在运输部门）的计算机。SMB网通常包括由多路由器互连的多个第二层交换机，并使用宽带因特网连接，也可能含有无线接入点。

大型企业网：大型企业网为大型公司提供所需的IT基础设施。典型的大型企业网往往连接着在地理上分开的几个基地（每个基地都有多个建筑物），需要使用大量的第二层交换机和路由器，并使用两条或更多的高速因特网连接。企业网通常兼有无线和有线技术设施。

小结　专网可以为个体消费者、小型办公室、中小型商业和大型企业提供网络服务。

1.5　网络、可互操作性和标准

通信至少要涉及两个实体，即发送信息的一方与接收信息的另一方。我们将看到，其实大多数分组交换通信系统还包含有中间实体（例如分组转发设备）。为保证通信成功，其中重要的一点是网络中所有参与通信的实体，必须在信息如何表示与沟通方面达成一致。通信协定包括许多细节。例如，当两个实体通过有线网络进行通信时，双方必须就所用的电压、使用电信号表示数据的正确方法、用于初始化与连接通信的规程以及消息的格式等达成一致。

我们使用术语可互操作性（interoperability）来表达两个实体进行通信的能力。并且说如果两个实体能够相互通信而不产生任何误解，那么它们就能正确地互操作。为了确保通信各方在通信细节上一致并遵从一组相应的规则，就必须制订一套精确的通信规范。

小结　通信涉及多个实体，它们必须就所用的电压及消息的格式与表示等诸多细节上取得一致。为确保所有通信实体能够正确地互操作，必须制订出涉及通信所有方面的规则。

套用外交上的词汇，对网络通信的规范通常使用通信协议、网络协议或协议（protocol）等术语。一个具体协议要规定低层通信的细节（例如无线网络中使用的无线电传输类型）或者要描述高层机制（例如两个应用程序所交换的消息细节），并定义在消息交换期间要遵从的规程。协议中最重要的方面之一就是对差错或意外情况的处理，因此协议通常都要对处理每个异常情况所采取的适当措施作出解释。

小结　通信协议规定了计算机通信某方面的细节，包括出现差错或意外情况时所采取的动作。一个具体协议可能规定低层的细节要求（如所采用的电压和信号形式），也可能描述高层方面的事项（如应用程序间交换的消息格式）。

1.6　协议组和分层模型

为确保所形成的通信系统是完整而有效的，必须认真构建一整套协议。为了避免重复，每个协议只需具备处理其他协议不处理的那部分通信功能。如何保证所有的协议都能很好地协调工作呢？这就需要有一个总体的设计规划——每个协议的设计不能是孤立的，而是应该

整体协调地设计所有协议，称为协议组或协议族。协议组中的每个协议只处理通信功能的一部分，而所有协议联合起来完成所有的通信功能，包括硬件故障和其他意外情况的处理。而且，还要使一个完整的协议组能高效协调地工作。

把各种协议集成为一个统一整体的抽象结构，被称为分层模型（layering model）。本质上，分层模型所描述的，就是如何把通信问题的所有方面划分成一个个协调工作的分块结构，每个分块就叫做一个层（layer）。因为协议组的这些协议被组织成一个线性序列，所以就产生了"层"这个术语，把协议划分到不同的层中，使它们各自在给定时间内专注于处理通信的某部分功能，有助于减少协议设计和实现的复杂性。

图1-1所示是采用因特网协议的分层模型概念示意图。在口头上，人们又把用来展现分层模型的直观图形说成是堆积起来的栈（stack），而协议组或协议簇也就被称为协议栈（protocol stack），这个术语就是指计算机中的协议软件，例如说："那台计算机运行TCP/IP协议栈吗？"



图1-1 使用因特网协议（TCP/IP）的分层模型

以后的几章将通过对协议的详细解释，来帮助我们理解分层，在此，我们只要领会每一层的用途以及如何利用协议来进行通信就足够了。后面几节将概括地描述各层所扮演的角色，以及计算机通信时数据是如何通过各个层次的。

第1层：物理层

物理（physical）层协议规定底层传输介质和相关硬件的细节。与电气特性、无线电频率和信号等有关所有的规范，都归属于第1层。

第2层：网络接口层

网络接口（network interface）层协议规定有关较高协议层（通常用软件实现）与底层网络（用硬件实现）之间进行通信的细节。有关网络地址、网络可支持的最大分组长度、用于接入底层介质的协议以及硬件编址等方面的规范，都归属于第2层。

第3层：网络互联层

网络互联（internet）层协议形成因特网最重要的基础，第3层协议规定两台计算机通过因特网（即通过多个互联网络）进行通信的细节。因特网的编址结构、因特网的分组格式、将大分组划分为小分组传输的方法以及差错报告机制等，都归属于第3层。

第4层：传输层

传输（transport）层协议为一台计算机上的应用程序和另一台计算机上的应用程序之间提供通信手段。控制接收端最大可接收数据的速率、避免网络拥塞的机制，确保所有数据以正确顺序接收的技术等方面的规范，都归属于第4层。

第5层：应用层

应用层是TCP/IP协议栈的最高层，该层协议规定一对应用进程在它们通信的时候如何交互，这层协议还规定有关应用进程所交换的消息含义和格式，以及通信过程中要遵循的规程等方面的细节。电子邮件交换、文件传输、Web浏览、电话服务和视频会议等方面的规范，都归属于第5层。

1.7 数据如何通过各个层次

分层的出现不只是为帮助人们理解协议这一抽象概念这么简单，其实协议的实现也将遵

络分层模型——将某层协议的输出传递到下一层协议的输入。而且，为了获得效率，相邻层上的一对协议只是传递数据分组的指针，而不是去复制完整的分组。这样在层之间就可以更加高效地传递数据。

为了更好地理解协议是如何操作的，假定有两台联网的计算机。图1-2是两台计算机上的分层协议示意图。正如图中所示，每台计算机上都含有一套分层协议。当一个应用进程要发送数据时，首先将数据放置到一个分组中，然后使该分组向下传递通过协议的每个层次。一旦该数据分组穿过了发送计算机的所有协议层，分组就离开计算机并通过底层物理网络进行传输[⊖]。当数据分组到达接收计算机时，该分组将向上传递通过各协议层。如果接收计算机上的应用进程收到数据后发出一个响应，则整个过程逆向进行。也就是说，这个响应报文以同样的方式向下传递通过各个层次，然后到达接收该响应的计算机，再向上传递通过各个层次。

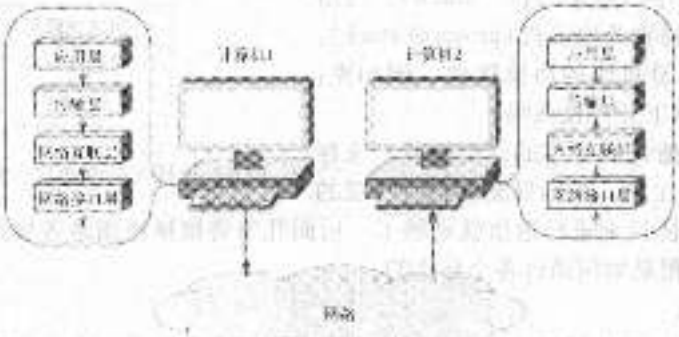


图1-2　当计算机通过网络进行通信时，显示数据如何在协议层间传递。每台计算机上都有一组分层协议，数据的传递要通过每一层

1.8　头部和各层

这里我们要弄清楚，协议软件的每一层都要完成一些计算，才能保证报文如期到达目的地。而为了完成这样的计算，两台计算机上的协议软件就必须交换一些信息。为此，发送计算机的每一层都要在数据分组中附加一些额外的信息；接收计算机的对应层协议则要取出并利用这些额外的信息。

由协议加进去的附加信息通常被称为头部（header）。为了理解头部是如何出现的，请再思考一下图1-2中通过网络的两台计算机间一个分组的传递过程。在发送计算机上，当分组向下传递通过每一层时，该层协议软件就加进去一个头部，即传输层附上一个头部，网络互联层附上一个头部，依此类推。这样，如果我们观察一个分组通过网络传送，则所有的头部都将按如图1-3所示的顺序出现。

虽然图1-3中所示的头部长度相同，但在实际中各个头部的长度是不统一的，且物理层头部可选的。当知道了头部的具体内容后，就可理解头部长度为什么会不相同。类似地，物理层通常只是规定如何使用信号去传输数据，所以人们不用期待一定能找到物理层头部。

⊖　图中只表示出一个网络。后面当我们学习因特网体系结构的时候，就会学到由路由器（router）的中间设备，而且还会知道当特网分层协议是如何工作的。

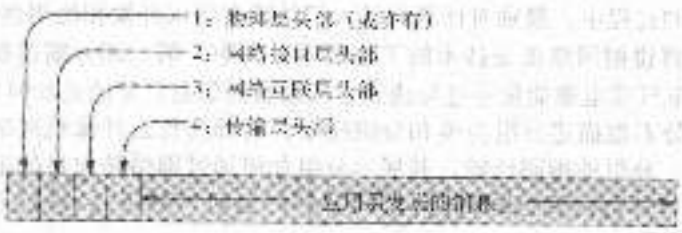


图1-3 分组在两台计算机间传递时，出现在分组上的联系式头部。图中分组起始位（通过底层网络传输的第1个位）在最左边

1.9 ISO与OSI七层参考模型

在互联网协议发展的同时，国际两大标准化组织联合推出了另一个参考模型，也创建了一套网络互联协议。这两大标准组织是：

- 国际标准化组织（ISO）。
- 国际电信联盟，电信标准化组（ITU-T）[1]。

ISO的分层模型就是后来为人熟知的开放系统互连七层参考模型（OSI）。由于协议参考模型的字首缩写（OSI）和国际标准化组织的字首缩写（ISO）很相似，所以二者的术语缩写经常引起混淆。因此，有时人们也可能会说成OSI七层模型或ISO七层模型。图1-4所示为该模型的7个层次。



图1-4 由ISO标准化的OSI七层模型

1.10 关于模型的内情点滴

像大多数标准化组织一样，ISO和ITU在创建一个标准的时候，在尽可能多的观点上经历了协调平衡。其结果，有些标准不是由工程师和科学家设计的，而是由相互妥协的标准化委员会设计完成的。七层参考模型实际上就是一个折中的产物，而且其模型和OSI协议也只是为了跟因特网协议相竞争而设计的。

ITU和ISO是处理全球电话系统和其他国际标准的大型标准化组织，而互联网协议和参考模型则是由十几个研究人员组成的小群体开发出来的。显而易见，这就是为什么这些标准化组织显得很自信，以为他们能够硬性推行一套协议并迫使大家从研究小组设计的协议那边转移过来的原因。当时，甚至连美国政府都相信，ITU和ISO制定的OSI协议将会取代TCP/IP。

最终，事情变得明朗起来，TCP/IP协议在技术上领先于OSI协议，而且在随后几年，OSI协议的研发和施行也被终止了。标准化组织遗留下一个七层参考模型，当然其中并没有包含网络互联层。后来，七层参考模型的倡导者为了和TCP/IP协议相吻合，试图将OSI定义延伸，考虑将七层模型中的第3层作为网络互联层，而且把一些支撑协议放入第5层和第6层。最滑稽的是，很多工程人员明明知道第5层和第6层是空洞的和没必要的，却仍然把应用层当做是第7层协议。

1.11 本书内容简介

本书划分为五大部分，在简单的导论以后，第一部分的其他章节将介绍网络应用和网络

[1] 标准首次建立之时，国际电信联盟（ITU）当时被称为国际电话电报咨询委员会（CCITT）。

编程。在学习本书的过程中，鼓励对计算机已入门的读者尝试开发和使用因特网的应用程序。本书的其他四部分将讲解网络低层技术的工作原理。其中，第二部分描述数据通信和信息传输，解释如何利用电气或电磁能量通过导线或空间来载送信息，并描述如何实现数据的传输。

本书的第三部分着重描述分组交换和分组技术，解释为什么计算机网络要采用分组，描述分组的一般格式、分组的编码传输，并展示分组如何通过网络传向目的地。这部分还将介绍计算机网络的基本分类，例如局域网（LAN）和广域网（WAN），表征每一类网络的特性并给出典型技术的例子。

本书的第四部分深入网络互联及其相关的TCP/IP互联协议组等内容，将介绍因特网的结构和TCP/IP协议，解释IP编址方案，以及因特网地址与底层硬件地址的映射，还会讨论因特网路由和路由协议。这部分内容还要阐述几个重要概念，包括封装、分片、报文和流量控制、虚拟连接、地址翻译、自举、IPv6和各种支撑协议。

本书的第五部分介绍了把网络作为一个整体而与之有关的其他课题。前面讲述网络性能，后面几章讲述新兴技术、网络安全和网络管理。

1.12　本章小结

大量的网络技术、产品和互连方法使得联网问题成为一个复杂的课题，这个课题有5个关键的方面：网络应用与网络编程，数据通信，分组交换和联网技术，使用TCP/IP的网络互联，以及跨层次应用的课题（如网络安全和网络管理）。

由于通信过程中有多个实体参与，所以它们必须在所有的细节问题上取得一致，包括电气特性（如电压）以及所有报文的格式和含义等。为了确保各实体间的可互操作性，每个实体必须遵从一系列通信协议。这些协议规定了为完成通信所需的所有细节。为了确保所有协议能够协同工作和处理好通信的各个方面，必须同时设计一整套协议组。为了构建完整的协议组而所做的一个最重要的抽象，称为分层模型（layering model）。分层有助于降低复杂性，可以使工程人员在某个时间内只集中考虑通信的一个方面，而不必顾虑其他方面。用于因特网的TCP/IP协议遵从于五层参考模型；而电话公司（作者可能是指ITU-T——译者注）和国际标准化组织（ISO）却提出了七层参考模型。

练习题

1.1　请说出近年来互特网发展的原因。
1.2　请列举出10个依赖于计算机网络的行业。
1.3　依照本章内容，在不了解因特网体系结构和技术的前提下，开发因特网应用是否可能？并对你的答案作出解释。
1.4　数据通信涉及联网的哪些方面？
1.5　什么是分组交换，为什么因特网与分组交换有关？
1.6　简述因特网的发展历史，说明因特网是何时和如何起源的。
1.7　什么是可互操作性，为什么它在因特网中特别重要？
1.8　什么是通信协议？在概念上，一个协议要对通信的哪两个方面作出规定？
1.9　什么是协议族（组），它具有哪些优点？
1.10　描述TCP/IP的分层模型，并解释该模型是如何推演出来的。
1.11　列出TCP/IP分层模型的各个层，并对每个层作简要说明。
1.12　请解释数据通过分层模型时，它的头部是怎样被加上和去除的。
1.13　请列举出为数据通信和计算机网络创立标准的主要标准化组织。
1.14　试简要说明ISO七层参考模型中的各个层次。

第2章 因特网的发展趋势

2.1 引言

本章内容论及自从数据网络和因特网开始以来，它们是如何发展演变的。本章首先介绍因特网的简要发展历史，突出反映了因特网的早期发展动机。然后，再描述因特网从共享集中化设施向完全分布式信息系统的重心转移过程。

这一部分的后续各章通过考察具体的网络应用来继续我们的讨论。这些章节除了描述因特网上采用的通信模式外，还将解释被网络应用程序利用来进行通信的编程接口。

2.2 资源共享

在进行计算机网络设计的早期，计算机都是昂贵的大型机，而且设计的主要目的是为了资源共享（resource sharing）。例如，网络只是被设计成将多用户终端（只有显示器和键盘）连接到一台中心的大型计算机。后来，网络又使得多用户终端可以共享外部设备（例如打印机等）。

要点 早期设计的计算机网络允许多台机器共享昂贵的、集中的各类资源。

20世纪60年代，美国国防部的高级研究计划署（ARPA）[⊖]对如何找到实现资源共享的方法特别感兴趣。他们的研究人员需要功能强大的计算机，而当时计算机又是出奇的昂贵。ARPA的财政预算根本无法采购许多计算机，于是ARPA就开始调研数据联网的问题——改变为每个研究项目购置计算机的做法。ARPA计划通过数据网络来实现所有计算机的互连，并开发相关的软件，以允许研究人员能够使用网络中任一台最适合完成给定任务的计算机。

为此，ARPA经过集思广益，最后决定集中到对计算机联网的研究，并由承包商把设计方案转化成为一个所谓的ARPANET运行系统。该项研究最终证明是革命性的，研究团队选择了遵循一种被称为分组交换（packet switching）的实现方法，它成为了数据通信和因特网的基础[○]。ARPA还继续资助互特网研究项目，在80年代期间，因特网随着研究人员的努力在不断扩展；到90年代，互特网已经成为一个商业上的成功范例。

2.3 因特网的成长

在不到30年的时间里，因特网已经由早期连接少量站点的研究原型成长为覆盖世界上所有国家的全球通信系统，其增长速度是非凡的。图2-1中，通过1981年到2008年间接入因特网的计算机数目的年函数曲线，展示了因特网的成长情况。

图2-1中的图形是采用线性比例绘制的，Y轴的数值范围从0～550百万台计算机，线性绘图方法会掩盖一些细节而不能反映出真实的情况。例如，图2-1中掩盖了关于互特网早期成长的真实数值详情，这就使人感觉好像因特网在1994年前几乎没有任何进展，而所有的增长都发生在最近的几年。事实上，在1998年新加入到因特网的计算机数的平均率已达到每秒钟超

⊖ 在许多时候，计划署还包含有国防（defense）这个词，并使用缩写DARPA。

○ 第13章将讨论分组交换。

过一台，这已经是在加速增长了，在2007年，连接到因特网的计算机数每秒钟已超过两台。为了理解早期的增长率，请看图2-2中采用对数比例尺绘制的图形。

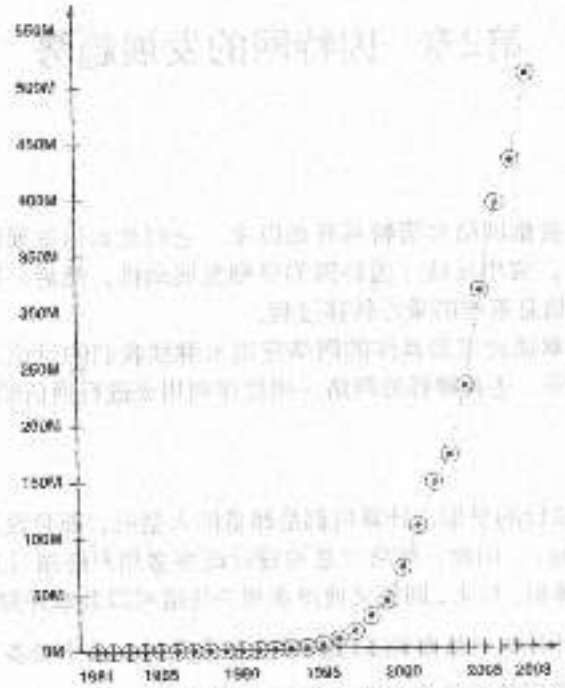


图2-1　因特网计算机数目的增长图

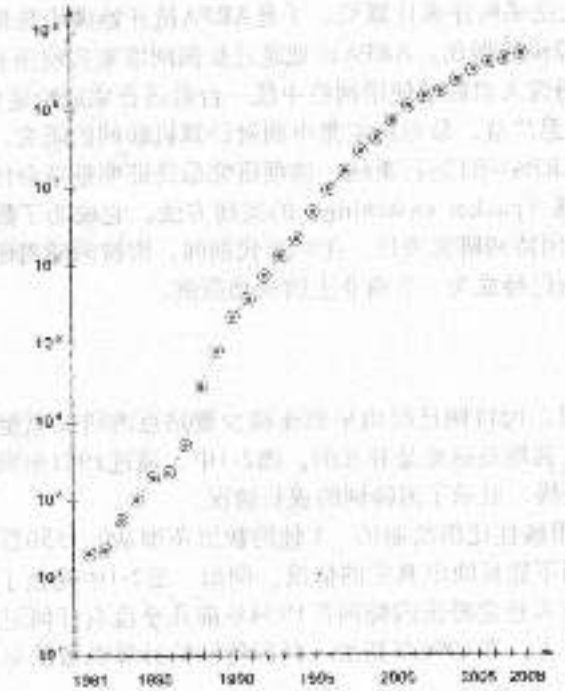


图2-2　采用对数比例尺绘制的因特网计算机数目的增长图

图2-2所示图形显示了因特网在过去近二十年中经历了指数的增长规律。也就是说，因特网的规模每过9～14个月就增长一倍。有趣的是，自20世纪90年代后期以来发达国家中相当大比例的人口接入因特网后，指数增长率开始出现下降趋势。

2.4 从资源共享到通信

在因特网的成长过程中，出现了两个方面的显著变化。第一，通信速率的急剧增长。目前因特网上干线路每秒传输的位数是原始因特网上干线路的10万倍。第二，已深入到社会方方面面的各种新应用的出现。十分明显，因特网不再是受科学家和工程师支配的网络，网络应用也不再只是接入到计算资源。

两个技术方面的变化加速了从资源共享转移到新网络应用的演化。一方面，更高的通信速率使应用系统可快速地传输大容量的数据；另一方面，强大而又廉价的个人计算机的出现提供了为复杂计算和图形化显示所需的计算能力，以往的资源共享需求已不再是最重要的目的了。

要点 高速计算和通信技术的可用性，使得因特网的关注点从提供资源共享能力转移到了提供通用的通信能力。

2.5 从文本到多媒体

通过因特网传输的数据类型也已经发生了十分明显的转移，图2-3所示为这种转移的某一个方面。

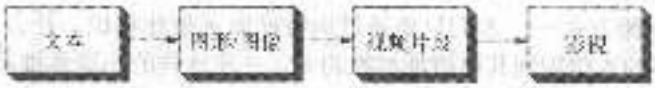


图2-3 用户通过因特网传输的数据类型的演进

如图所示，因特网通信起初只涉及文本类数据，特别是电子邮件消息还局限于固定大小字体显示的文本。到了20世纪90年代，计算机已配备可以显示图形的彩色屏幕，并且出现了可方便地允许用户传送图像的网络应用。到20世纪90年代后期，因特网用户开始发送视频片段，而且全活动影视数据的传送也已变得可行。图2-4所示为因特网传送音频类数据的转移过程。



图2-4 用户通过因特网传输音频类数据的演进

我们使用术语多媒体（multimedia）来表征含有文本、图形、音频和视频组合的数据。目前，因特网传送的大部分实用内容都是由多媒体文档组成的，而且随着更高的带宽及传输质量也不断得到改善，使得在网上进行高清晰度影视和高保真音乐的交流，都已成为可能。

要点 因特网应用已从传送静态的文本文档发展到可传送高质量的多媒体内容。

2.6 目前的发展趋势

令人惊奇的是，新的联网技术和新的因特网应用还在不断涌现。当传统的通信系统（如电话网、有线电视网）从模拟向数字转移并采纳因特网技术以后，发生了一些最显著的变化。

另外，因特网对移动用户的支持也在加速发展。图2-5列举了这些转变：

领　域	转　变
电话系统	从模拟电话转向IP电话（VoIP）
有线电视	从模拟传递转向IP传送
蜂窝移动通信	从模拟模式转向数字蜂窝服务（3G）
因特网接入	从有线接入转向无线接入（Wi-Fi）
数据访问	从集中式服务转向分布式服务（P2P）

图2-5　IP网技术引因特网发生转变的例子

因特网最有趣的方面之一是，虽然网络应用发生了或发生着许多的变化，但因特网的底层技术却未发生实质性的改变。例如，图2-6中所列举的已涌现出的几个应用类型。

应　用	适用领域
高质量远程会议	商务到商务（B2B）项目
导航系统	军事、航运业、旅游者
传感器网络	环保、安全、舰队跟踪
社区联网	消费者、志愿者组织

图2-6　当前流行的网络应用举例

商务活动中使用的高质量远程会议系统（如思科公司Telepresence）是非常有意义的，因为它可以节省差旅费。在许多商务活动中，减少差旅费能显著降低企业成本。

社区网络应用（例如Facebook、Second Life和YouTube）是很令人着迷的，因为它们创建了一个全新的社会交流方式——人们只要通过因特网即可彼此相识。社会学家暗示：这样的网络应用能够使更多的人结识到其他情趣相投的人，并使这样的小群体维系得更加紧密。

2.7　本章小结

美国国防部的高级研究计划署（ARPA）资助了早期的绝大部分联网的研究，为ARPA的研究人员找到了一条共享计算资源的途径。后来，ARPA把重心转移到网络互联上，并资助了对因特网的研究，在近几十年来因特网一直在按指数规律增长。

随着高速个人计算机和更高速网络技术的出现，因特网的焦点从起初的资源共享转移到了通用的通信交互。因特网上传送的数据类型也从文本演进为图形、视频片断和全活动视频。在音频方面也发生了类似的转变。因特网已能传送各种多媒体文档。

因特网技术在许多方面影响着现代社会。目前的显著变化是电话、有线电视和蜂窝移动服务等领域都在采用数字的因特网技术。此外，无线的因特网接入和对移动用户的支持也都已成为不可或缺的因特网服务。

虽然底层的因特网技术还基本维持不变，但各种新的应用层出不穷地涌现出来，为因特网用户提供了更强的网络应用体验。商务活动利用远程会议系统可以减少差旅花费，传感器网络、电子地图和导航系统等将使环境监测、安全和旅行变得更方便易行，社区联网将有助于形成新的社会群体和组织。

练习题

2.1　在20世纪60年代，为什么共享计算资源是最重要的？

2.2 图2-1中绘出的图形显示出1995年以前因特网几乎没有增长，为什么会出现这样的误导？

2.3 假设每年新增1亿台计算机连接到互特网，如果按均匀的速率增加，则每增加一台计算机要经历多少时间？

2.4 延伸绘制图2-2，试估算一下到2020年将会有多少台计算机连接到因特网。

2.5 当万维网首次出现时，因特网应用发生了什么转变？

2.6 试列举从早期因特网到目前因特网图形表达的演变步骤。

2.7 试描述因特网在音频方面所发生的演进过程。

2.8 因特网技术对有线电视产业有什么冲击？

2.9 电话系统在采用什么因特网技术？

2.10 为什么因特网从有线接入到无线接入的转变是意义重大的？

2.11 请列举4个新的因特网应用，并说出各自的重要应用领域。

2.12 请描述你经常使用的互特网应用，且该应用是你父母在你目前的年龄时所没有的。

第3章 因特网应用与网络编程

3.1 引言

因特网为用户提供了丰富而多样化的服务选择，包括Web浏览、电子邮件、文字短信和视频远程会议等。但令人惊讶的是，虽然没有一种服务属于底层通信基础结构部分，因特网只是为构建所有服务提供了通用的通信机制而已。而各种服务功能都是由连接到因特网上的计算机通过运行应用程序来提供的。其实，即使不对因特网进行任何改变，也可能设计出全新的应用服务。

本章涵盖了两个阐释因特网应用的关键概念。第一，在因特网通信时网络应用所要遵循的概念模式。第二，因特网应用所采用的套接字应用编程接口（socket API）。

本章将演示即使程序员不懂得数据通信或网络协议的详情，也能编写出一个有创意的应用程序，也就是说，只要程序员掌握了一些基本的概念，就可能构建一个通过因特网通信的应用系统。下一章通过剖析因特网应用系统的例子（如电子邮件），来继续我们的讨论。

虽然程序员可以在不了解网络如何运作的情况下，就能轻松起步去开发因特网应用，但是理解网络协议和技术却可以让程序员能够编写出高效的、可靠的，能在更大网络范围内应用的程序代码。

3.2 因特网的基本通信模式

因特网支持两种基本通信模式，即流模式和报文模式，图3-1概括了两种模式的差异。

流模式	报文模式
面向连接的	无连接的
一对一通信	多对多通信
一个字节流序列	一个个报文的序列
任意长度传输	每个报文最长最多64KB
被大多数应用使用	用于多媒体应用
构建在TCP协议之上	构建在UDP协议之上

图3-1 因特网支持的两种通信模式

3.2.1 因特网中的流传送模式

术语流（stream）是指字节序列从一个应用程序流到另一个应用程序的一种通信模式。事实上，因特网的通信机制在一对通信应用之间安排了两个方向的流。例如，浏览器就是采用流服务来与Web服务器通信的：浏览器发出一个请求，Web服务器以发送网页作为响应。网络从任一个应用接受输入数据，并将数据传递给另一个应用。

采用“流”机制来传送字节序列时，它不会对字节加入任何含义，也不会对序列插入任何边界。特别是，发送端程序可以选择每次只发送一个字节，也可以发送多个字节块。任何时候网络都是选择字节的编号来进行传送的。也就是说，网络可以选择把多个小字节块合并

成一个大字节块，或把大字节块分割为多个小字节块，这对字节流的传送都不会造成影响。

要点 虽然流模式是按顺序传递所有字节的，但它并不保证传递到接收端应用的字节块还会与发送端发出的字节块是相对应的。

3.2.2 因特网中的报文传送模式

因特网通信机制还遵循报文模式（message paradigm），按这种模式，网络所接收和传递的数据是报文形式的，而且传递到接收方的每个报文跟发送方发出的报文是相对应的，即网络不会只传递报文的一部分，也不会把多个报文合并在一起传递。所以，如果发送方发出的报文正好是n个字节，那么接收端输入的报文也一定会正好是n个字节。

报文模式允许单播、多播或广播传输，即一个报文可以从一台计算机上的某个应用直接发送给另一台计算机上的某个应用，该报文也可以广播到给定网络的所有计算机上，或者组播到某个网络的某些计算机上。而且，多台计算机上的多个应用也可发送报文到某一特定的计算机上，所以，报文模式可以提供一对一、一对多、多对一等方式的通信。

应注意的是，报文服务对报文被传送的顺序不做任何保证，也不能确保某个给定的报文一定能到达接收方。这种服务模式使得报文有如下问题：

- 丢失（即从未传递成功）。
- 重复传递（有多个报文副本到达）。
- 乱序传递。

采用报文模式的程序员一定要保证：即使在分组丢失或乱序的情况下，应用程序仍然能够正确运行[⊖]，由于大多数应用都要求保证数据的正确传递，所以除非有特殊情况（例如传输视频，它要求采用多播方式，而且应用软件应能支持对分组丢失和乱序的处理），一般程序员都倾向于采用流模式，因此，以下我们重点讨论流模式。

3.3 面向连接的通信

因特网流服务是面向连接的，这意味着其服务操作与电话呼叫过程相似：在双方开始通信之前，两端应用进程必须请求建立一个连接，一旦连接建立后，就允许应用进程在该连接上进行任一方向的数据发送，最后，当通信完成后，应用进程要请求终止该连接。算法3-1概括了面向连接的通信交互过程。

算法3-1

目的：

进行面向连接的通信

方法：

对应用进程请求建立连接

从对应用进程到用该连接交换数据

当对应用进程请求终止该连接

算法3-1 面向连接的通信过程与方法

3.4 客户—服务器交互模式

算法3-1中的第一步出现了一个问题：运行在两台独立计算机上的一对应用进程如何协调

⊖ 后续章节将解释为什么会发生这样的错误。

来确保在相同时间内请求连接呢？其答案就在于一种称为客户-服务器模式（client-server model）的交互形式。一个应用程序（称为服务器）首先启动运行并等待连接请求。而另一个应用程序（称为客户）随后运行并主动发起连接请求。图3-2概括了客户-服务器的交互过程。

服务器应用进程	客户应用进程
首先运行	随后运行
不需要知道哪个客户将连接它	必须知道想要连接的服务器
被动等待来自客户的连接请求，且等待时间任意长	在需要通信的任何时候，发起连接请求
通过发送和接收数据与客户进行通信	通过发送和接收数据与服务器进行通信
在实现又一个客户请求服务后，继续运行并等待另一个请求	在完成与服务器的交互后，可以终止运行

图3-2　客户-服务器模式概要

在随后的几节中，将描述一些特定的应用是如何使用客户-服务器模式的。现在，我们只要理解下面的意思就可以了：

> 虽然因特网提供了基本的通信服务，但它并不能发起与远端计算机的连接请求，也不能接受来自远端计算机的连接请求。这些服务由称为客户和服务器的应用程序来完成。

3.5　客户和服务器特征

虽然存在少量的变种，但大多数客户-服务器交互模式都具有相同的一般特征。一般情况下，客户软件具有如下特征：

- 它是一个任意的应用程序，仅在需要进行远程访问时才暂时成为客户，同时还要完成其他的计算任务。
- 直接受用户个人操作，并且只执行一个会话过程。
- 在用户的个人计算机上进行本地运行。
- 主动地发起与服务器的连接请求。
- 能访问所需的多种服务，但通常一次只与一个远地服务器请求连接。
- 不会特别地要求功能强大的计算机硬件。

而服务器软件的特征如下：

- 它是一个专门提供某种服务的专用特权程序，但同时可以处理多个远程客户的请求。
- 在系统启动时自动被调入执行，进行多次会话并持续不断地运行。
- 运行在大型、高性能计算机上。
- 被动地等待来自任意的远端客户的通信请求。
- 接收来自任何客户的通信请求，但只提供单一的服务。
- 要求功能强大的硬件和高级复杂的操作系统支持。

3.6　服务器程序和服务器类计算机

对于服务器（server）这个术语，有时会出现一些混淆。正式地说，这个术语是指一个被动地等待通信的程序，而不是指运行服务器程序的那台计算机。然而，当一台计算机专门用来运行一个或几个服务器程序时，这台计算机本身有时也称为服务器。硬件供应商更是促成了这种混淆，因为他们将那些具有快速CPU、大容量存储器和强大操作系统的计算机都归类

为服务器。图3-3所示为有关的定义。

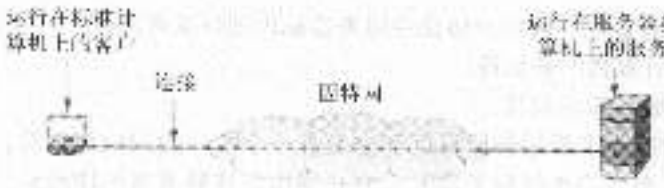


图3-3 客户与服务器示意图

3.7 请求、响应和数据流方向

术语客户和服务器是通过谁发起连接请求来区分的。一旦连接建立，就可以进行双向通信（即数据可以从客户流向服务器，或从服务器流向客户）。一般是客户向服务器发送一个请求，服务器向客户返回一个响应。在某些情况下，客户可以向服务器发送一系列请求，服务器则返回一系列响应（例如，一个数据库客户程序可能允许用户一次查询一个以上的记录）。

要点 信息可以在客户与服务器之间沿任一方向或双向流动。虽然很多服务都是设置为由客户发出一个或多个请求，服务器返回相应的响应，但其他交互方式也是可能的。

3.8 多客户与多服务器

客户或服务器都是由应用程序构成的。一台计算机可以同时运行多个应用程序，因此一台计算机能够运行：

- 单个客户。
- 单个服务器。
- 连接某服务器的多个客户副本。
- 各自连接特定服务器的多个客户。
- 各自提供特定服务的多个服务器。

允许一台计算机运行多个客户是很有用的，因为这样它可以同时访问多个服务。例如，一个用户同时打开3个窗口分别运行3个应用程序。第一个进程取回并显示电子邮件，第二个进程进行聊天，第三个进程进行Web浏览；每个应用进程都各自连接到彼此独立的特定服务器上。实际，一个用户也可以同时打开某个应用程序的两个副本（如Web浏览器的两个副本），而且两个副本可各自分别连接到一个服务器上。

允许一台计算机上运行多个服务器也是很有用的，因为这样就可以共享硬件设备，此外也可以降低系统的管理开销，更重要的是，经验表明对服务器的需求并不是经常性的——在一段很长的期间服务器是空闲的。在等待下一个请求到来期间，服务器空闲就意味着它没有使用CPU。因此，假如对服务的需求很低，就可以将多个服务器合并在单台计算机上运行，从而能够急剧减少成本但不会明显降低性能。

要点 一台功能强大的计算机可以同时提供多种服务，而每种服务都需要单独的服务器程序。

3.9　服务器的标识与识别

客户怎样识别服务器呢？因特网协议将服务器标识划分成两部分：

- 运行服务器的计算机的标识符，
- 计算机上特定服务的标识符。

标识计算机。因特网上的每台计算机都分配有一个唯一的32位识别符，称为网际协议地址（IP地址）[1]。当客户连接服务器时，它必须指定该服务器的IP地址。为了使人们更容易标识服务器，每台计算机也被分配了一个名字。第4章要描述的域名系统就是用来将计算机名字翻译成IP地址的。所以，用户对服务器是使用名字（如www.cisco.com）来指定的，而不是使用整数地址。

标识服务。因特网上每个可用的服务也分配有一个唯一的16位识别符，称为协议端口号（通常简称为端口号），例如：电子邮件（Email）的端口号是25，Web服务的端口号是80。当服务器开始运行时，服务器通过指定所提供服务的端口号来向本地系统注册。当一个客户连接远地服务器请求服务时，该请求中就包含了端口号。这样，当一个请求到达服务器时，服务器的软件就利用请求中包含的端口号来决定服务器计算机上的哪一个应用软件去处理这个请求。

图3-4通过列举客户和服务器通信所采取的步骤，对以上的讨论进行归纳。

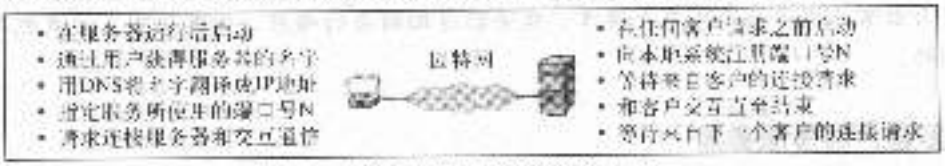


图3-4　客户与服务器进行通信的步骤

3.10　并发服务器

图3-4中的处理步骤暗示：一个服务器一次只能处理一个客户的请求。虽然在一些简单的情况中可采用串行排队的方法，但大多数服务器都是并发工作的，即服务器使用多个控制线程[2]，同时处理多个客户请求。

为了理解同时服务的重要性，考虑一下假如一个客户从服务器下载一部电影会发生什么样的情况。如果服务器一次只能处理一个请求，那么在服务器传送电影的过程中所有的客户必须等待，而并发服务器却不会要求客户等待。因此，假如第二个客户到达并请求下载一个短内容（例如一首歌），服务器会立即响应第二个请求，并在电影下载结束前很快完成下载。

有关并发执行的细节取决于所用的操作系统，但其思路是很简单的：并发的服务器程序被分为主程序（线程）和句柄两部分，主程序只接受来自客户的连接请求，并为该客户创建一个控制线程；每一个控制线程只与一个客户交互，并执行句柄程序。当处理完一个客户后，该线程终止。这期间，主程序（线程）仍然保持活跃状态——在为一个客户请求创建一个线程后，主程序等待另一个请求的到来。

注意：如果有N个客户在同时使用并发服务器，那么就有N+1个线程在运行。主线程在等待其他的请求到来，其他的N个线程则分别与N个客户进行交互通信，我们可以概括如下：

并发服务器使用执行线程来同时处理多个客户的请求。

---

[1] 第21章将详细阐述因特网地址。

[2] 有些操作系统使用术语"执行线程"、"进程"或表示控制线程。

3.11 服务器间的循环依赖

从技术角度看，向一个程序发起连接请求的任何程序，都是在扮演一个客户的角色，而接受来自另一个程序的连接请求的任何程序，都是在扮演一个服务器的角色。实际上，由于提供某种服务的服务器也可以起到是另一个服务器的客户的作用，所以客户与服务器的界限有时会变得模糊不清。例如，在填写完一个动态网页前，Web服务器可能需要变成数据库服务器的客户（向数据库发出请求），一个服务器也可能成为安全服务器的客户（例如，服务器需要以客户的身份请求安全服务器，以便认证原来的客户是否允许访问这项服务）。

当然，程序员必须精心规划以避免服务器间的循环依赖。考虑一下，假如服务X_1的服务器成为服务X_2的客户，且服务X_2的服务器成为服务X_3的客户，最后服务X_3的服务器又变成X_1的客户，这会出现什么现象呢？这些请求形成一个闭合的链，将会连续不断地循环请求，直到耗尽这三台服务器的资源为止。当各自独立地设计各个服务的时候，由于单个程序员不可能控制所有的服务器，所以潜在的循环依赖风险特别高。

3.12 P2P交互

如果单个服务器提供某一特定服务，那么服务器和因特网间的网络连接可能成为瓶颈。图3-5所示为这种架构。

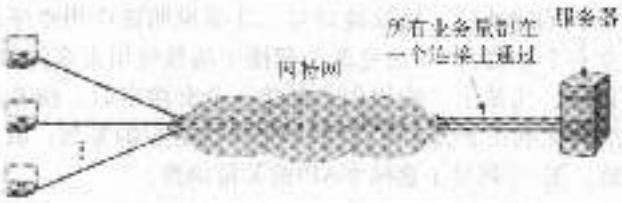


图3-5 使用单服务器设计的流量瓶颈

这里就产生了一个问题：因特网提供的服务能否不产生中心瓶颈呢？避免瓶颈的方法形成了一个文件共享应用的基础。有一种被称为P2P（Peer to Peer）的结构方案，它的思想就是避免将数据放置在单个中央服务器上。从概念上讲，就是将数据平均分布在N个服务器上，并将每个客户的请求发送到合适的服务器。由于一个服务器仅仅提供1/N的数据，所以服务器与因特网之间的业务流量只是单服务器结构中的1/N。因此，服务器软件可以运行在与一般客户机相同的计算机上。图3-6所示为P2P方案的架构。



图3-6 P2P系统的交互关系

3.13 网络编程与套接字API

被应用软件用来规范通信操作的接口，就称为应用编程接口API[⊖]。虽然API的真实情况

⊖ 附录给出了一个简化的API（只含有7个函数）和实现代码，通过这个示例代码，演示了如何使用这个API来开发因特网应用（包括一个实用的Web服务器）。

取决于操作系统，但特别的套接字API的出现，使它已成为开发因特网通信软件的事实标准。套接字API目前被许多操作系统采用，例如，微软的Windows、各类UNIX、Linux。

要点　套接字API是因特网通信的一个事实标准。

3.14　套接字、描述符和网络I/O

因为套接字API最初是作为UNIX操作系统的一部分而开发的，所以套接字API与系统的其他I/O设备集成在一起。特别是，当应用程序要为因特网通信而创建一个套接字（socket）时，操作系统就返回一个小整数作为描述符（descriptor）来标识这个套接字。然后，应用程序以该描述符作为传递参数，通过调用函数来完成某种操作（例如通过网络传送数据或接收输入的数据）。

在许多操作系统中，套接字描述符和其他I/O描述符是集成在一起的，所以应用程序可以对文件进行套接字I/O或I/O读/写操作。

要点　当应用程序要创建一个套接字时，操作系统就返回一个小整数作为描述符，应用程序则使用这个描述符来引用该套接字。

3.15　参数与套接字API

套接字编程与传统的输入/输出有所不同，因为一个应用程序要使用套接字必须指定许多细节。例如，远端计算机的IP地址、协议端口号，并要说明该应用程序是作为客户还是服务器（即是否要发起建立一个连接）。为避免单个套接字函数使用太多的参数，套接字API的设计者选择定义了很多函数。实质上，应用程序创建一个套接字后，接着就要调用函数来指定细节。这种套接字方法的优点在于大多数函数只有3个或更少的参数，缺点是编程者在使用套接字时要调用多个函数。图3-7列举了套接字API的关键函数。

名　称	使 用 者	含　义
accept	服务器	接受一个长到的连接请求
bind	服务器	指定IP地址和端口号
close	服务器、客户	终止通信
connect	客户	连接到远端应用进程
getpeername	服务器	获取客户IP地址
getsockopt	服务器	获取套接字的当前选项
listen	服务器	准备服务器使用的套接字
recv	服务器、客户	接收输入的数据或报文
recvmsg	服务器、客户	接收数据（报文模式）
recvfrom	服务器、客户	接收报文和发送者地址
send（write）	服务器、客户	发送输出数据或报文
sendmsg	服务器、客户	发送一个输出报文
sendto	服务器、客户	发送一个报文（sendmsg的变形）
setsockopt	服务器、客户	改变套接字选项
shutdown	服务器、客户	终止一个连接
socket	服务器、客户	创建一个套接字（用以上某种形式）

图3-7　套接字API的主要函数

3.16　客户和服务器中的套接字调用

采用流连接模式的客户和服务器进行套接字调用的顺序，如图3-8所示。图中，客户首先

向服务器发送数据，服务器等待接收数据。实际上，有些应用则是安排由服务器先向客户发送数据（即接收的顺序调用send和recv函数）。



图3-8 采用流连接模式的客户和服务器调用套接字函数的顺序

3.17 客户和服务器共用的套接字函数

3.17.1 Socket函数

调用一次Socket函数创建一个套接字，并返回一个整型描述符：

描述符 = socket (protofamily, type, protocol)

参数protofamily指定与套接字一起使用的协议族，在因特网中所使用的TCP/IP协议族是用标识符PF_INET来指定的。参数type指定套接字将要采用的通信类型：流传输模式的通信用值SOCK_STREAM来指定；无连结的报文传输模式的通信用值SOCK_DGRAM来指定。参数protocol指定套接字所使用的特定传输协议。

有了type参数再加上protocol参数，就可以允许单个协议组里包含有两个或多个协议，它们提供相同的服务。当然，protocol参数所能取的值取决于协议族。

3.17.2 Send函数

客户和服务器都使用send函数来发送数据，通常，客户发送的是一个请求，而服务器发送的则是一个响应。send有4个参数：

send (socket, data, length, flags)

参数socket是要用到的套接字描述符；参数data是待发送数据在内存中的地址；参数length是一个整数，指定数据的字节数；参数flags包含请求特殊选项的比特位[①]。

3.17.3 Recv函数

客户和服务器都使用Recv函数获取由另一方发送的数据。这个函数具有如下形式：

recv (socket, buffer, length, flags)

参数socket是要从中接收数据的套接字描述符；参数buffer指定用来存放接收数据的内存地址；参数length指定这个缓冲区的大小。最后，参数flags允许调用者控制一些细节（例如，允许应用程序在未将报文从套接字中删除之前提取它的一个副本）。平时Recv是被阻塞的，直

① 有许多选项是依系统设计的，不同的客户和服务器程序是不可用的。

至数据到达时才被开启，然后在缓存器的指定位置存放最多length字节的数据（从函数调用返回的值将指明被提取数据的字节数）。

3.17.4　调用Socket的读/写操作

在有些操作系统中（如Linux），可以使用操作系统函数read和write来代替recv和send函数。read函数的3个参数要与recv函数的前3个参数一致；同样，write函数的3个参数也要与send函数的前3个参数一致。

使用read和write的主要优点在于它们的通用性——开发一个传输数据的应用程序时，只需将数据发送到一个描述符（或者从一个描述符接收数据），而不必知道这个描述符对应的是一个文件还是一个套接字。因此，在程序员试图通过网络进行通信之前，可以使用本地盘上的文件来测试客户或服务器程序。使用read和write的主要缺点是，应用程序被用在其他操作系统环境中时，可能需要修改程序。

3.17.5　Close函数

close函数告诉操作系统要终止对一个套接字的使用[1]。它具有如下形式：

close (socket)

socket是要关闭的套接字的描述符。如果连接是打开的，则close就终止该连接（即通知另一方）。关闭一个套接字意味着立即终止对它的使用——描述符被释放，以阻止应用程序再发送和接收数据。

3.18　仅供客户使用的connect函数

客户调用connect函数与指定的服务器建立连接。它的形式如下：

connect (socket, saddress, saddresslen)

参数socket是为建立的连接所用的套接字描述符；参数saddress是一个sockaddr结构，它指定服务器地址和协议端口号[2]；参数saddresslen指定服务器地址的长度（字节数）。

对于采用流模式的套接字，connect发起一个到指定服务器的传输层连接，而这个服务器必须等待连接（参见下述的accept函数）。

3.19　仅供服务器使用的套接字函数

3.19.1　bind函数

创建套接字之后，它并不含有本地或远端地址，也没有协议端口号。服务器调用bind函数以提供一个协议端口号，并在这个端口上等待通信连接请求。bind有3个参数：

bind (socket, localaddr, addrlen)

其中，参数socket是所用套接字的描述符；参数localaddr是一个结构，指定将要赋给套接字的本地地址；参数addrlen是一个整数，指定地址的长度。

由于套接字可以被任何协议所使用，所以这个地址的格式取决于所使用的协议。套接字API定义了一个用来表示地址的通用格式，然后要求每个协议族说明它们的协议地址如何使用这个通用格式。这个表示地址的通用格式被定义为一个sockaddr结构。虽然已经公布了几种版

[1] 微软的Windows Sockets接口使用名称closesocket而不是close。

[2] IP地址和协议端口号的组合有时被称为端地址（endpoint address）。

本，但大多数系统定义的sockaddr结构包含3个域：

```
struct sockaddr {
        u_char  sa_len;         /* total length of the address */
        u_char  sa_family;      /* family of the address       */
        char    sa_data[14];    /* the address itself          */
};
```

sa_len域指定地址长度由8位字节组成，sa_family域指定地址所属的协议族（字符常量AF_INET表示TCP/IP地址）。最后，sa_data域包含该地址。

每个协议族要为sockaddr结构中的sa_data域定义准确的地址格式。例如，TCP/IP协议使用sockaddr_in结构来定义地址：

```
struct sockaddr_in {
        u_char  sin_len;        /* total length of the address */
        u_char  sin_family;     /* family of the address       */
        u_short sin_port;       /* protocol port number        */
        struct  in_addr sin_addr;/* IP address of computer     */
        char    sin_zero[8];    /* not used (set to zero)      */
};
```

sockaddr_in结构的前两个域正好对应于通用sockaddr结构的前两个域，后三个域定义了TCP/IP协议所希望的正确地址格式。有两点值得注意：第一，每个地址标识了一台计算机以及该计算机上的一个特定应用程序。sin_addr域包含这台计算机的IP地址，而sin_port域包含这个应用程序的协议端口号。第二，虽然TCP/IP只需要6个字节来存放整个地址，但通用sockaddr结构仍保留了14字节。于是，sockaddr_in结构的最后一个域包含了一个8个字节的零值域，将这个结构填充到与sockaddr相同的大小。

我们说过，服务器调用bind来指定它将要接受连接请求的协议端口号。然而，除了协议端口号外，sockadd_in结构还包含一个记录IP地址的域。虽然服务器在指定地址时也能选择填写IP地址，但当主机是多穴主机（即有多个网络连接）时，这样做将会产生问题，因为这意味着计算机有多个地址。为了使服务器能在多穴主机上运行，套接字API包含了一个特殊的符号常量INADDR_ANY，它允许服务器指定一个端口号，使该计算机可以在任一个IP地址上使用这个端口号来接受连接请求，格括如下：

> 虽然Sockaddr_in结构含有IP地址域，但套接字API仍然提供了一个符号常量，以允许服务器在该计算机的任一IP地址指定一个协议端口。

3.19.2 listen函数

在使用bind指定了协议端口后，服务器调用listen函数使这个套接字处于被动模式，以使它能等待来自客户的连接请求。listen过程有两个参数：

```
listen ( socket, queuesize )
```

参数socket是套接字的描述符，参数queuesize指定该套接字的请求队列的长度。操作系统为每个套接字各自创建一个请求队列。起初，队列是空的。当来自客户的请求到达时，它们都被置入队列中。当服务器要求从套接字中检出一个接收到的请求时，系统即从队列中取出下一个请求。队列长度是一个重要的参数，假如请求到达时队列已满，系统将会拒绝该请求。

3.19.3 accept函数

服务器调用accept来建立与一个客户的连接。如果一个请求出现在队列中，accept立即返回（接受请求并建立连接），如果没有请求到达，系统阻塞服务器直至有客户发起请求。一旦连接已被接受，服务器就能够使用该连接与客户进行交互通信。完成通信后，服务器关闭该连接。

accept函数具有如下形式：

newsock = accept (socket, caddress, caddresslen)

参数socket是服务器已经创建并绑定在指定协议端口上的套接字描述符，参数caddress是sockaddr结构类型的地址，而caddresslen则是一个指向一个整数的指针。accept在caddress参数域中填入已建立连接的客户地址，并给caddresslen参数设置地址长度。最后，accept为该连接创建一个新的套接字，并将这个新套接字的描述符返回给调用者。服务器使用这个新的套接字与客户进行通信，然后在结束后关闭该套接字。同时，服务器的原始套接字保持不变——在服务器结束与一个客户的通信后，它使用这个原始套接字来接收下一个客户的连结请求。因此，原始套接字只用于接受请求，而所有通信发生在由accept创建的newsock上。

3.20　采用报文模式的套接字函数

比起在流模式中使用的情况，用于发送和接收报文的套接字函数要复杂得多，因为有很多的选项可供使用。例如，发送方可以选择是将接收方地址存入套接字中并只是发送数据，还是指定接收方的地址每次发送一个报文。此外，一个函数允许发送方将地址和报文放置到结构中，并把结构的地址作为一个参数来传递；另一个函数则允许发送方将地址和报文作为分开的参数来传递。

3.20.1　Sendto和Sendmsg函数

sendto和sendmsg函数可以让客户或服务器使用未连接的套接字来发送报文，两者都要求调用者指定目的地址，sendto使用单独的参数来传送报文和目的地址：

sendto (socket, data, length, flags, destaddress, addresslen)

前4个参数对应于send函数的4个参数，后两个参数指定目的地址和该地址的长度。参数destaddress的类型为sockaddr结构（特别是当使用TCP/IP协议组时，就是sockaddr_in结构）。

sendmsg函数完成和sendto函数相同的操作，但它通过定义结构而缩短了参数表，从而使采用sendmsg的程序可读性更好：

sendmsg (socket, msgstruct, flags)

参数msgstruct是一个结构，它包含目的地址、该地址长度、待发送报文以及报文长度等信息：

```
struct msgstruct {                  /* structure used by sendmsg  */
        struct sockaddr *m_addr;    /* ptr to destination address */
        struct iovec    *m_iovec;   /* ptr to message (vector)    */
        int     m_iovlength;        /* num. of items in vector    */
        struct access   *m_rights;  /* ptr to access rights list  */
        int     m_rlength;          /* num. of items in list      */
};
```

报文结构的详情并不重要——它应当被看做是将多个参数组合成为一个结构的一种方法而已。大多数应用程序仅使用前3个域，以指定目的协议地址、一个构成报文的数据项列表和列表中的项目个数。

3.20.2　Recvfrom和Recvmsg函数

一个未连接的套接字可被用来接收来自任意客户组发送的报文。在这种情况下，系统返回每个接收到的报文和发送方地址（接收方将使用该地址来回送应答）。recvfrom函数的参数用于说明下一个接收报文的存放区域和发送方地址：

recvfrom (socket, buffer, length, flags, sndraddr, saddrlen)

前4个参数对应于recv函数的参数，另外两个参数（sndraddr和saddrlen）用来记录发送方的IP地址，参数sndraddr是sockaddr结构的指针，系统将发送方地址写入其中；参数addrlen是指向一个整数的指针，系统用它来记录地址的长度。

注意 recvfrom记录的发送方地址与sendto所期望的正好相同，这样就很容易回送给发送方一个应答。

函数recvmsg（它的对应发送函数是sendmsg）的操作类似于recvfrom，但要求较少的参数。它具有如下形式：

recvmsg (socket, msgstruct, flags)

其中参数msgstruct给出一个结构的地址，这个结构包含了接收报文的地址以及发送方的IP地址。由recvmsg所记录的msgstruct使用与sendmsg所要求的结构完全相同，这样就使回送应答十分容易。

3.21 其他套接字函数

套接字API还包含各种各样的支持函数。例如，服务器在调用accept函数接受连接请求之后，它可以调用getpeername函数以获取启动连接的远程客户的完整地址。客户或服务器也可以调用gethostname来获取运行该程序的计算机的信息。

有两个通用的函数可以被用来设置套接字选项，setsockopt函数用来设置套接字选项，getsockopt函数可获取当前选项值，选项主要用来处理特殊情况（例如增加内部缓冲区大小）。

有两个函数可被用来在IP地址和计算机名之间进行转换，gethostbyname函数通过给出计算机名字来返回该计算机的IP地址。客户经常使用gethostbyname将用户输入的名字转换成相应的协议软件所需的IP地址。gethostbyaddr函数则是提供一个反向映射，即给出一台计算机的IP地址，它将返回该计算机的名字。客户和服务器可以使用gethostbyaddr函数把地址转换为人们能理解的计算机名。

3.22 套接字、线程和继承性

套接字API要与并发服务器一起工作。虽然其细节要取决于底层操作系统，但套接字API的实现仍遵循下面的继承性原理：

每个新创建的线程，都从创建它的线程那里继承所有打开套接字的一个副本。

套接字的实现使用了一种引用计数（reference count）机制来控制。当一个套接字首次被创建时，系统将该套接字的引用计数置为1，只要引用计数保持为正值，该套接字就存在。当程序创建一个新的线程时，该线程对程序拥有的每个打开的套接字继承一个指针，并将每个套接字的引用计数值加1。当一个线程调用close时，系统将该套接字的引用计数减1，如果引用计数值减到零，则删除该套接字。

对并发服务器而言，主线程拥有用来接受连接请求的套接字。当一个连接请求到达时，系统为这个连接创建一个新套接字，同时主线程创建一个新线程去处理该连接。在创建一个线程后，两个线程就可去访问新的和旧的套接字，而且两个套接字的引用计数值都为2。主线程为新套接字调用close，而服务线程为旧套接字调用close，两者的引用计数值皆减为1。最后，在服务线程结束与客户通信时，它对新套接字调用close，将它的引用计数值减到零，以

号被该套接字被删除。因此，并发服务器中的套接字生存期可概括为：

只要主服务器线程在执行，它用来接受连接请求的initial套接字就一定存在；仅当处理连接请求的服务线程存在时，为特定连接所使用的套接字才会存在。

3.23 本章小结

在因特网中，所有服务都是由应用软件提供的。应用程序可才用流模式或报文模式进行通信。流模式通信可确保按顺序传递一系列字节，而且传送给接收方的每批数据可以选择不同的长度。报文模式通信则要求传递的数据要保留边界，而且所传送的报文可能丢失、重复和乱序。

网络应用所采用的基本通信模式被称为客户/服务器工作模式。被动等待通信的程序称为服务器，主动发起与服务器建立连接的程序称为客户。

每台计算机都分配有一个唯一的地址，每个服务（如Email、Web访问）也分配有一个称为协议端口号的唯一标识符。服务器启动时，会指明一个协议端口号，客户要连接服务器时，必须指定运行服务器的计算机IP地址，以及服务器所用的协议端口号。

一个客户可访问多个服务，也可访问多台计算机上的多个服务器；提供某种服务的服务器在访问其他服务时也可成为一个客户。在多服务器的环境中，设计者和编程人员应足够小心以避免循环依赖。

应用编程接口（API）规范了应用程序如何与协议软件进行交互的细节。尽管其细节取决于操作系统，但套接字API是一个事实上的标准。一个程序创建一个套接字，然后使用该套接字调用一系列函数。使用流模式的服务器可调用的套接字函数有socket、bind、listen、accept、recv、send和close；客户可调用的套接字函数有socket、connect、send、recv和close。

因为很多服务器都是并发式的，所以套接字API被设计成要与并发应用一起工作。当一个新的线程被创建时，这个新线程就继承了其创建线程所拥有的对所有套接字的访问权。

练习题

3.1　因特网中使用的两个基本通信模式是什么？

3.2　试给出因特网流模式通信的6个特征。

3.3　试给出因特网报文模式通信的6个特征。

3.4　如果发送方使用流模式，且每次总是发送1024B，那么因特网传递给接收方的数据块长度是多少？

3.5　假如发送方想把一个数据块的副本传递给3个接收者，应使用哪个通信模式？

3.6　因特网报文传递语义的3个意外方面是什么？

3.7　试给出面向连接系统所用的一般算法。

3.8　当两个应用软件在因特网上交互通信时，哪一个是服务器？

3.9　通过总结客户和服务器特征，试对两者进行比较。

3.10　服务器和服务器类计算机有什么不同？

3.11　数据能从客户流向服务器吗？请解释。

3.12　试列举出在给定计算机上运行的客户和服务器的可能组合。

3.13　所有的计算机都能高效地运行多个服务吗？为什么？

3.14　指定一个特定的服务器要用那两个标识符？

3.15 在用户输入服务器的域名后，列出客户请求连接服务器的步骤。
3.16 为了同时处理多个客户请求，并发服务器要用的基本操作系统特性是什么？
3.17 什么性能问题激发了P2P通信？
3.18 请给出两个提供套接字API的操作系统。
3.19 一旦创建套接字后，应用程序如何引用该套接字？
3.20 套接字API的主要函数有哪些？
3.21 请给出客户和服务器的典型函数调用流程。
3.22 与read和write相对应的函数是什么。
3.23 客户可用bind函数吗？请解释。
3.24 为什么要用符号常量INADDR_ANY？
3.25 函数sendto是用于流模式还是报文模式的？
3.26 假设套接字是打开的并创建了新线程，新线程可以使用该套接字吗？
3.27 测试附录中的Web服务器，并用套接字API建立一个等同的服务器。
3.28 用套接字函数实现附录中简化的API。

第4章 传统的因特网应用

4.1 引言

第3章主要介绍了因特网应用和网络编程的内容。本章要阐述利用应用程序来定义的因特网服务，并展示这些应用程序使用客户-服务器模式进行交互的特征。本章也涉及套接字API。

本章进一步剖析因特网应用和定义传输协议的概念，并解释应用程序是怎样实现传输协议的。最后，讨论一些标准的因特网应用，并描述它们各自所用的传输协议。

4.2 应用层协议

在开发网络通信两端应用程序时，程序员要详细说明一些细节，例如：

- 报文的语法和语义。
- 客户或服务器是否发生交互。
- 发生差错时所采取的动作。
- 网络通信两端怎样知道何时终止通信。

在指定通信过程的细节方面，程序员要定义出应用层协议。目前有两大类应用层协议，它们是：

- 专用协议。程序员开发的两端应用程序是在因特网上通信并具有专门目的的。大多数情况下，两端应用程序之间的这种交互过程很简单，即程序员不需要编写正式的协议说明书就可以直接编写程序代码。
- 标准化服务。由很多程序员参与开发服务器软件来提供该种服务，或由很多程序员参与开发客户软件来访问该种服务。在这种情况下，应用层协议的制定必须独立于任何实现，而且协议规范必须准确而无二义性，这样才能保证所有的客户和服务器正确地进行互操作。

协议规范书的长度取决于服务的复杂性，一个简单服务的说明书可以是单页文本。例如，因特网协议中包含一个称为日期时间（DAYTIME）的标准服务，它可使客户在服务器特定区域上得到当地日期和时间，该协议简单明了：客户形成到服务的连接，服务器发送ASCII码表示的日期和时间，然后服务器关闭连接。例如，服务器可能发送如下字符串：

Sat Sep 9 20:18:37 2008

客户端从连接上读取数据，直至收到EOF（文件结束）符为止。概括如下：

> 为了使应用软件满足标准化服务的互操作性，应用层协议标准的制定必须独立于任何实现。

4.3 表示与传输

应用层协议需要规范交互操作的两个方面，即表示和传输。图4-1解释了两者的区别。

方 面	描 述
数据表示（Data Representation）	报交换数据项的语法，传输期间采用的特定形式，计算机间整数、字符和文件的转换
数据传输（Data Transfer）	客户和服务器之间的交互，报文语法和语义，有效和无效交换的差错处理，交互过程的终止

图4-1 应用层协议的两个关键方面

对于基本服务，用单个协议标准就可以说清楚这两方面内容；对于更加复杂的服务，则要用单独的协议标准才能分别说清楚各个方面的内容。例如，上述的DATATIME协议用单个标准就可以说明如何将日期和时间表示为ASCII字符串，以及规定服务器如何发送该字符串，然后关闭连接等传输过程。在4.4节将解释Web是如何使用单独的协议分别去描述网页的语法和网页的传输过程，协议设计者要区分清楚：

作为习惯，应用层协议标题中的词汇“传输”（Transfer）意味着协议规范所说明的是通信数据传输方面的内容。

4.4 Web协议

万维网（Web Wide Web, WWW）是因特网上使用最广泛的服务之一，由于Web比较复杂，已经设计很多协议标准来规范Web的各个方面和具体细节。图4-2列出了3个关键的标准。

标 准	用 途
超文本标记语言（HTML）	用于规范网页内容和数据布局的表示标准
统一资源定位符（URL）	用于规范网页识别符格式和含义的表示标准
超文本传输协议（HTTP）	用于规范浏览器如何与Web服务器交互传输数据的传输协议

图4-2 WWW服务使用的3个关键标准

4.5 HTML文档表示法

超文本标记语言（HyperText Markup Language, HTML）是一种规范网页语法的表示标准，它具有以下特征：

- 使用文本表示。
- 描述包含多媒体的页面。
- 遵循说明性的而不是过程性的模式。
- 提供标记规范而不是格式化。
- 允许超级链接嵌入在任意对象上。
- 允许文档包含元数据。

虽然HTML文档是由文本组成的，但它允许程序员说明含有图形、音频、视频和文本的任意复杂的网页。事实上，由于HTML允许任意对象（如一幅图像）包含指向另一个网页的链接（有时称为超级链接），所以更准确地说，在名称上超文本（hypertext）更应该表达为超媒体（hypermedia）。

由于HTML只是允许人们去规定将要做什么，而不是如何做，所以HTML被归类为说明性（declarative）的语言。由于HTML仅仅给出了网页显示方面的一般指导，而没有包括详细的

格式编排指令，所以它又被归类为标记语言（markup language）。例如，HTML可以指定一个标题的重要性级别，但它不需要作者指定标题的准确字体、字形、磅值和间隔[1]。实质上，由浏览器来选择所有显示的细节，而标记语言的重要性在于它能使浏览器将网页适配到底层的显示硬件上。例如，一个网页可以根据显示器的分辨率，是大屏幕还是小型手持设备（如iPone或PDA）等情况，对网页进行相应的格式化。

概括如下　超文本标记语言（HTML）是Web网页的表示标准，为了能使网页在任意设备上显示，HTML对网页显示给出了一般性的指导，而显示的具体细节则允许由浏览器来选择。

HTML采用嵌在文本中的标签（tags）来规范标记的方法。标签是由小于号和大于号括起来的专用词构成的，它对文档提供结构化表示，并提供格式化提示。标签控制所有的显示，在HTML文档的任何位置可以插入空白（即额外的行和空隔字符），但对浏览器的显示格式不会造成影响。

例如，一个HTML文档以标签<HTML>开始，以标签</HTML>结束；一对标签<HEAD>和</HEAD>之间是文档的头部。同样，一对标签<BODY>和</BODY>之间是文档主体。在头部中，标签<TITLE>和</TITLE>之间所含的内容是文档的标题部分。图4-3所示为HTML文档的一般形式。[2]

```
<HTML>
  <HEAD>
      <TITLE>
            text that forms the document title
      </TITLE>
  </HEAD>
  <BODY>
    body of the document appears here
  </BODY>
</HTML>
```

图4-3　HTML文档的一般形式

HTML使用IMG标签给外部图形的引用进行编码。例如，标签：

```
<IMG SRC="house_icon.gif">
```

说明：house_icon.gif文件中含有一个浏览器将在文档中插入图形。还可以用IMG标签中的附加参数来说明图形和周围文本的排列关系，例如图4-4所示为以下HTML文档的输出，图中文本与图形的中部对齐显示。

```
Here is an icon of a house. <IMG SRC="house_icon.gif" ALIGN=middle>
```

浏览器垂直放置该图形，文本与图形中部对齐显示。

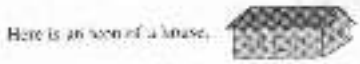


图4-4　HTML文档中图形对齐的示意图

---

[1] HTML扩充版本已经制定，允许说明准确字体、字形、磅值和编排格式。

[2] HTML不区分标签里的大小写字母，例子中用了大写字母只是为了强调而已。

4.6 统一资源定位符和超级链接

Web使用一种称为统一资源定位符（Uniform Resource Locator，URL）的句法形式来指定一个网页，它的一般形式是：

协议://计算机名字:端口/文档名?参数

其中：协议是访问文档所使用的协议名，计算机名字是文档所在计算机的域名，端口是可选的协议端口号，服务器以该端口号被动等待请求，文档名是在指定计算机上的可选文档名，?参数给出网页的可选参数。

例如，URL：

http://www.netbook.cs.purdue.edu/toc/toc01.htm

指明所用的协议是http，计算机域名是www.netbook.cs.purdue.edu，文件路径和名是toc/toc01.htm。

用户输入的典型URL中许多部分可省略。例如，URL：

www.netbook.cs.purdue.edu

省略了协议（默认为http）、端口号（默认为80）、文档名（默认为主页）和参数（默认为空缺）。

URL含有浏览器提取网页所需的相关信息，浏览器利用分隔符号（冒号、斜杠和百分号）把URL划分为4部分：协议、计算机域名、文档名和参数。浏览器运用计算机域名和端口号形成向网页所在服务器的连接，运用文件名和参数请求一个特定的网页。

HTML中的锚标(anchor)利用URL来提供超级链接的能力（即从一个Web文档链接到另一个文档的能力）。下面的例子展示了具有锚标的HTML源文档，其中锚标的引用涉及名称Prentice Hall：

```
This book is published by
<A HREF="http://www.prenhall.com">
Prentice Hall</A>, one of
the larger publishers of Computer Science textbooks.
```

上述锚标引用了URL http://www.prenhall.com。当显示在屏幕上时，这个HTML文档输入将产生：

This book is published by Prentice Hall, one of the larger publishers of Computer Science textbooks.

4.7 用HTTP传输Web文档

超文本传输协议（HyperText Transfer Protocol，HTTP）是浏览器用于与Web服务器交互的主要传输协议。根据客户/服务器模式，浏览器就是客户，它从URL中提取出服务器名字并请求连接服务器。大多数URL都含有显式的协议引用http://，或者省略该协议提示，此时默认就是HTTP。

HTTP可被表征如下：

- 使用文本控制报文。
- 传送二进制数据文件。
- 可以下载或上传数据。
- 一体化高速缓存。

一旦建立连接，浏览器就会向服务器发送一个HTTP请求。图4-5列举了4种主要的请求类型，

请　求	描　述
GET	请求一个文档，服务器响应：发送状态信息，紧接着发送该文档的一个副本
HEAD	请求状态信息，服务器响应：发送状态信息，但不发送文档副本
POST	发送数据给服务器，服务器将该数据添加到指定主页上（例如，将提交添加到一个列表中）
PUT	发送数据给服务器，服务器用该数据完全替代指定的项（即覆盖或重写原先的数据）

图4-5　4种主要的HTTP请求类型

当浏览器向服务器请求一个页面时，即开始了最普通的交互形式。浏览器在跟客户的连接上发送GET请求，然后服务器响应回送头部、一空行和被请求的文档。在HTTP中，请求和响应的头部都是由文本信息组成的。例如，一个GET请求具有如下的形式：

GET /item version CRLF

其中：item给出被请求项的URL，version指定HTTP的版本号（一般是HTTP 1.0或HTTP 1.1），CRLF表示回车和换行的ASCII字符，常用于表示文本行的结束。

由于对协议的改变要保持后向的兼容性，所以版本信息很重要。例如，使用HTTP 1.0协议的浏览器与具有更高版本的服务器交互时，服务器就重返到旧版本协议上，并构建出相应的响应文本来。概括如下：

使用HTTP协议时，浏览器要发送版本信息，以便让服务器选择使用双方都能理解的最高版本协议。

响应头部的第一行含有一个状态码，它告知浏览器其请求是否已被服务器处理过。如果请求形式不正确或请求项不可用，则状态码将指明问题所在。例如，如果请求的项不存在，旧服务器返回大家熟知的状态码404。若服务器接受了一个请求，就返回状态码200，并在头部中给出多行有关该请求项的进一步信息，例如该项的长度、最后被修改的时间、内容类型等。图4-6表示出一个基本响应头部各行的一般格式。

说明：status_code域是表示为字符串的一个十进制整数值，它指示出某个状态，status_string是供人阅读的相应的注释。图4-7列出了常用状态码和注释行的例子。域server_identification包含一个描述性的字符串，它是可供人阅读的对服务器的描述，很可能包含服务器的域名。在Content_Length头部中的datasize域指定了紧随的数据项的长度（以字节为单位）。document_type域含有一个字符串，通知浏览器有关文档内容方面的类型信息，该字符串由斜杠分隔的两个项组成：文档类型和它的表示。例如，当服务器返回一个HTML文档时，document_type是text/html；当服务器返回一个jpeg文件时，document_type是image/jpeg。

```
HTTP/1.1 status_code status_string CRLF
Server: server_identification CRLF
Last-Modified: date_document was_changed CRLF
Content-Length: datasize CRLF
Content-Type: document_type CRLF
CRLF
```

图4-6　一个基本响应头部的一般行格式

状态码	对应的状态字符串
200	OK
400	Bad Request
404	Not Found

图4-7　HTTP中使用的状态码例子

图4-8所示为由Apache Web服务器输出的样本，正被请求的项是含有16个字符的文本文件（即该文本是This is a test. 外加一个换行符）。虽然GET请求指定是HTTP 1.0，但服务器运行的是HTTP 1.1。服务器返回9行头部、一个空行和文件内容。

```
HTTP/1.1 200 OK
Date: Sat, 13 Mar 2008 07:35:25 GMT
Server: Apache/1.3.37 (Unix)
Last-Modified: Tue, 11 Jan 2008 12:05:17 GMT
Etag: "76595-0[illegible]-[illegible]"
Accept-Ranges: bytes
Content-Length: 16
Connection: close
Content-Type: text/plain

This is a test.
```

图4-8 由Apache Web服务器返回的HTTP响应样本

4.8 浏览器中的高速缓存

由于用户趋向重复地访问相同的网站，所以使用高速缓存能为Web访问提供重要的最优化处理。一个网站的大部分内容是由使用GIF或JPEG标准的大型图像组成的，这样的图像通常含有不经常变化的背景和标志。关键思想：

> 浏览器通过在用户硬盘缓存中保存每个图像的副本，以及利用被缓存的副本，就可以明显地减少下载的次数。

这里产生了一个问题：浏览器在缓存中保存了一个文档副本后，若服务器上的该文档改变了会怎样呢？即浏览器如何知道其缓存的副本是否已经过期？图4-8中的响应信息提供了一个提示：头部中的"最后更新"（Last-Modified）。浏览器无论何时从服务器获取一个文档，其头部中都会指明该文档最后变更的时间；而随着对文档副本的缓存保留，浏览器也随之保存了该文档最后变更的日期时间信息。在使用一个本地缓存的文档之前，浏览器向服务器发出一个HEAD请求，然后比较服务器副本与本地缓存副本的最后变更日期时间。假如缓存的文档版本是旧的，浏览器将下载新的版本。算法4-1归纳了缓存处理的过程。

算法4-1

```
Given:
   A URL for an item on a web page
Obtain:
   A copy of the page
Method:
  if (item is not in the local cache) {
        Issue GET request and place a copy in the cache;
  } else {
     Issue HEAD request to the server;
     if (cached item is up-to-date) {
        use cached item;
     } else {
        Issue GET request and place a copy in the cache;
     }
  }
```

算法4-1 浏览器中用于减少下载次数的缓存处理

上述算法省略了几个次要的细节。例如，HTTP允许Web站点发送包含不可缓存的头部，其头部规定客户端不可以缓存某个给定项。另外，浏览器也不缓存一些小长度的内容，因为使用GET请求来下载这些内容的时间和使用HEAD请求的时间几乎是相同的，以及在缓存中保存许多小长度的内容也会增加对缓存的查找次数。

4.9　浏览器结构

由于浏览器要提供通用服务和支持图形界面，所以浏览器是复杂的。当然，浏览器首先必须要理解HTTP，但同时还需提供对其他协议的支持。特别是，由于URL可以指定协议，所以浏览器必须包含有每个所用协议的客户代码。对于每种服务，浏览器必须知道如何与服务器交互和如何解释响应，例如浏览器要知道如何去访问FTP服务器（将在4.10节介绍）。图4-9所示为浏览器所含的功能组件。

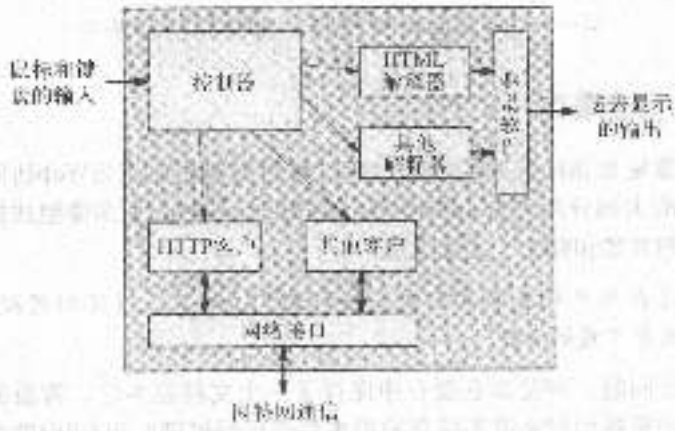


图4-9　能访问多种服务的浏览器结构

4.10　文件传输协议

文件（file）是基本的存储单元。因为文件可以保存为任意对象（例如，文档、电子表格、程序、图形图像或数据），所以将文件副本从一台计算机传送到另一台计算机的设施，为数据交换提供了强大的机制。对这样的一种服务，我们就使用文件传输（file transfer）这个术语。

由于计算机系统是异构的，这意味着每个计算机系统都各自定义文件表示方式、类型信息、文件命名和文件访问机制，所以通过因特网来传送文件也就变得复杂起来。一些计算机系统中，.jpg扩展名被作为JPEG图像，而在另一些计算机系统中则可能是.jpeg；一些系统中，文本中每行的结束使用单个换行符（LINEFEED），而其他系统中则需要回行符（CARRIAGE RETURN）再后随换行符。一些系统使用斜杠（/）作为文件名的分隔符，其他系统中则采用反斜杠（\）。而且，操作系统可以定义一组用户账户，每个账户都给以访问文件的一定权限。然而，在不同计算机之间的账户信息也不尽相同，因此某计算机上的用户X并不等同于另一计算机上的用户X。

因特网最广泛使用的文件传送服务是采用文件传输协议（File Transfer Protocol，FTP）。FTP具有如下特征：

- 任意文件内容。FTP可以传送任意类型的数据，包括各类文档、图像、音乐或存储的视频。
- 双向传送。FTP可以用于下载文件（从服务器到客户的传送）或上传文件（从客户到服务器的传送）。
- 支持验证和拥有权。FTP允许文件具有拥有权、访问限制和授权访问。
- 有能力浏览文件夹。FTP允许客户获得目录（即文件夹）的内容。
- 文本形式的控制报文。像许多其他的因特网应用服务，在FTP客户和服务器之间交换的

控制报文是ASCII文本。

- 容纳异构性。FTP隐藏各个计算机操作系统的细节，从而可以在任意两个计算机之间传送文件的副本。

由于很少用户专门去安装和启用FTP应用，所以通常见不到FTP协议。但是，当用户在浏览器中请求下载文件时，浏览器会自动调用FTP。

4.11 FTP通信模式

FTP最有趣的一点就是客户与服务器交互的方式。总的来说，交互方式好像很简单：客户向FTP服务器建立连接，并发送一系列的请求，对此服务器作出响应。但不像HTTP那样，服务器在客户发送请求的同一个连接上回送响应并交互数据，而是由FTP服务器在每次需要下载或上传文件时另开通一个并交互数据。为了区分两个不同的连接，由客户请求建立的原始连接称为控制连接（control connection），用于传送命令；后开通的新连接称为数据连接（data connection），专门用于传输文件数据。

感到意外的是，正是这个数据连接使得FTP反转了客户—服务器关系，即开通数据连接时，客户却起到像服务器（等待数据连接）那样的作用，而原来的服务器却起到像客户（发起数据连接）那样的作用。在用于完成一次文件传送以后，数据连接被关闭；如果客户发送另一个请求，则服务器再开通一个新连接。图4-10显示其交互过程。



图4-10 共享会话过程的FTP连接示意图

图4-10省略了几个重要细节。例如，建立控制连接后，客户必须登录服务器。FTP服务器要求用户提供USER（用户名）；客户发送并提交登录用户名后，FTP服务器又要求提供PASS（密码）；客户发送并提交自己的用户密码。服务器将在控制连接上回送一个数值状态响应，使客户知道是否登录成功。客户只有在成功登录后才能发送其他命令。[1]另一个重要细节是

[1] 当访问公共文件上，客户使用匿名登录，其用户名是anonymous，密码是guest。

关于数据连接使用的协议端口号，连接客户时，服务器应该指定什么协议端口号呢？FTP协议提供了有趣的答案：在向服务器发出请求前，客户在本地操作系统上分配一个协议端口号，并将之发送给服务器。也就是说，客户是绑定到这个端口上等待控制连接的，然后在控制连接上发送PORT命令，通知服务器已用的端口号。算法4-2归纳了这些步骤。

算法4-2

给定：
　一个FTP控制连接。
目的：
　在一个FTP数据连接上传输一个数据项。
方法：
　在控制连接上，客户发送对一特定文件的请求；
　服务器收到请求；
　客户分配一个本地协议端口号，称为X；
　客户绑定到端口X，并准备接受连接；
　在控制连接上，客户发送PORT X给服务器；
　服务器接收PORT命令和数据项请求；
　客户在端口X上等待数据连接和接受该连接；
　服务器向客户计算机自己端口X建立一个数据连接；
　在数据连接上，服务器发送所请求的文件；
　服务器关闭该数据连接。

算法4-2　FTP客户和服务器使用数据连接要采取的步骤

在一对应用程序间传输端口信息似乎无关紧要，但实际上却不是，在所有场合这种技术都难于很好地工作。特别是，如果两者有一端是安放在网络地址转换NAT设备之后，例如，用于小型办公室或住宅的无线路由器，那么协议端口号的传输就会出问题。第23章将会解释FTP是一个例外——为了支持FTP，NAT设备要识别FTP控制连接，检查该连接传送的内容，并重新填写PORT命令的数值。

4.12　电子邮件

尽管像即时通信这样的服务已经十分流行，但电子邮件（E-mail或Email）仍然是使用最广泛的因特网应用之一。由于电子邮件的构思出现在PC和手持PDA以前，所以电子邮件被设计成允许在一台计算机上的用户直接发送信息给另一台计算机的用户。图4-11所示为电子邮件结构。算法4-3列出了其采用的步骤。

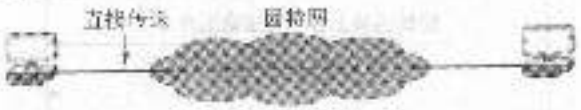


图4-11　从发方计算机到收方计算机直接发送的原始Email配置

正如算法4-3所示，即便是早期的Email软件，它也被划分成概念上独立的两个部分，即

- 电子邮件应用接口。
- 邮件传送程序。

算法4-3

给定：
从一个用户到另一个用户发Email消息。
提供：
向期望的接收者传输一个报文。
方法：
用户调用应用接口，并为用户x@destination .com产生一个Email报文；
用户的Email接口程序将该报文排队待送；
用户计算机上的邮件传送程序检查发送队列，并发现该邮件；
邮件传送程序向destination .com建立连接；
邮件传送程序使用SMTP传送该报文；
邮件传送程序关闭连接；
destination .com上的邮件服务器将收到邮件，并将一个副本放入用户X的邮箱；
destination .com上的用户X运行邮件接口程序来显示用户X的邮箱（含有新邮件）。

算法4-3 以原始模式发送Email所采取的步骤

用户可直接调用Email接口应用。该接口为用户提供以下功能：发送邮件的写作和编辑，以及接收邮件的阅读和处理。但Email接口应用不能起到客户或服务器那样的作用，也不能向其他用户传送邮件。Email接口应用只是从用户的邮箱（即用户计算机上的文件）读出报文，并将待发报文存放到发送邮件队列（通常是用户磁盘上的一个文件夹）中。负责处理传送邮件的是两个单独的程序，即邮件传送程序和邮件服务器程序。前者扮演客户的角色，向远地计算机上的邮件服务器发送邮件；后者扮演服务器角色，负责接收输入的报文，并将每个报文存放到相应的用户邮箱中。

用于因特网Email系统的规范可划分为3大类，如图4-12所示。

类 型	描 述
传送	将Email报文从一个计算机移动到另一个计算机的协议
访问	允许用户访问其邮箱并阅读或发送Email报文的协议
表示	Email报文本身的，规定其格式的协议

图4-12 用于Email系统的3类协议

4.13 简单邮件传输协议

简单邮件传输协议（Simple Mail Transfer Protocol, SMTP）是邮件传送程序所用的标准协议，用于通过因特网向服务器传送Email报文。SMTP可被表征如下：

- 遵循流模式。
- 使用文本控制报文。
- 仅传送文本报文。
- 允许发送者指定接收者的名字和核实每个名字。
- 发送给定报文的一个副本。

SMTP最不希望有的特征就是它对文本内容的限制。在4.4节中将阐述一个多用途因特网邮件扩展MIME标准，它允许Email包含如图形文件或二进制文件的附件，但底层的SMTP机制仍限于文本传送。

SMTP第一个受关注的方面是：将单个报文发送到一给定计算机上多个接收者的能力问题。

协议允许客户端一次一个地列出用户，然后为列表上的所有用户逐个发送一个报文副本。也就是说，客户发送报文"I have a mail massage for user A"，服务器要么回答"OK"，要么回答"No such user here"。其实，每个SMTP服务器报文都是以一个数值码开头，所以回答的形式是"250 OK"或"550 No such user here"。图4-13给出一个SMTP会话的例子，发生会话期间，example.edu计算机上的John_Q_Smith传送一个邮件报文给somewhere.com计算机上的两个用户。

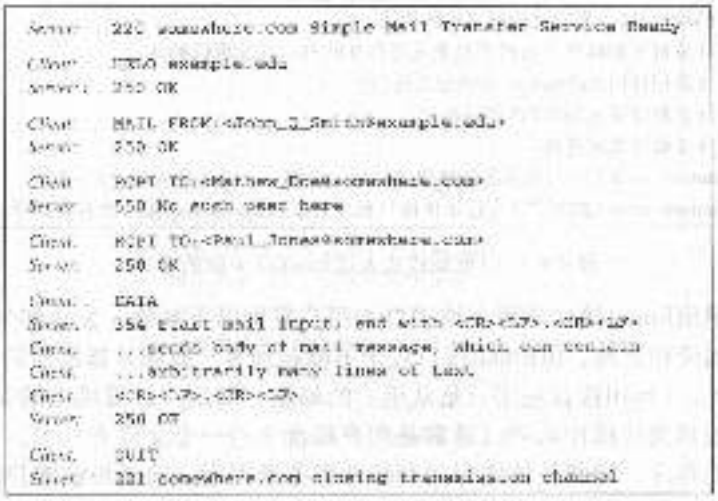


图4-13　一个SMTP会话实例

在图4-13中，每行被标记为Client或Server，指示该行是由客户还是服务器发送的。而协议并不包含这些标记。HELO命令允许客户通过自身域名鉴别自己。最后，符号<CR><LF>表示回车后跟换行符（即一个行的结束标志）。因此，通过由句号和空格（无其他文本字符）组成的一行来表示邮件报文主体的结束。

所谓简单（Simple）只是意味着SMTP是经简化的版本。因为SMTP的前面版本复杂得难以置信，所以设计者删除了不必要的特性，并浓缩到最基本的功能上。

4.14　ISP、邮件服务器和邮件访问

当因特网扩展到包含消费者的时候，Email系统出现了新模式。由于大多数消费用户并不能连续守候在运行的计算机旁，而且也不知道如何配置和管理邮件服务器，所以因特网服务提供商ISP开始提供Email服务。本质上，ISP运行Email服务器并为每个签约用户提供邮箱。ISP不是提供传统的Email软件，而是提供允许用户访问邮箱的接口软件。图4-14所示为这种Email系统的配置情况。



图4-14　一种Email系统配置（其中ISP运行Email服务器并向用户提供对邮箱的访问）

Email访问可采取以下两种途径之一：

- 使用专用的Email接口应用软件，
- 使用Web浏览器访问Email网页。

使用Web浏览器的方法是一种简捷途径：ISP提供显示用户邮箱报文的特殊网页，所以用户只要安装并启动Web浏览器并访问ISP即可。Web网页要求输入用户登录ID和密码(password)，Web服务器用它来识别用户邮箱；Web服务器从相应用户邮箱中提取邮件，并作为网页来显示邮件内容。使用Web网页的Email系统的主要优点是：可在任一台计算机上阅读Email——用户不需要运行特殊的邮件接口应用软件。

使用特殊的邮件接口应用软件的优点是：可以在本地计算机上下载完整的邮箱。这种下载能力对于使用掌上电脑的移动用户来说特别重要。当掌上电脑连接网络时，用户运行Email程序把整个邮箱下载到掌上电脑；在掌上电脑和因特网无法连接时（如在飞机上）用户可以处理邮件；一旦与因特网重新连接后，掌上电脑的软件与ISP的服务器交互通信，上传用户编写的邮件和下载已到达用户邮箱的新邮件。

4.15 邮件访问协议

目前，已经开发了多种提供电子邮件访问（access）的协议。访问协议不同于邮件传输协议，因为访问仅仅涉及到单个用户与单个邮箱的交互作用，而传输协议则允许一个用户向多个其他用户发送邮件。访问协议具有以下特征：

- 提供对用户邮箱的访问。
- 允许用户浏览邮件头部，下载、删除邮件，发送单个邮件报文。
- 客户运行在用户PC机上。
- 服务器运行在储存用户邮箱的计算机上。

在用户和邮件服务器之间采用低速连接的情况下，只浏览邮件列表而不下载邮件内容的做法很有用。例如，一个通过移动蜂窝电话接入进行浏览的用户，不用等待下载报文内容，就可以查阅邮件头部信息和删除垃圾邮件。

在Email访问方面，已经提出了多种机制，有些ISP向其签约用户提供免费的Email访问软件。此外，有两个标准的Email访问协议也已经开发出来。图4-15列出了标准协议名称。

缩　写	全　称
POP3	邮局协议版本3 Post Office Protocol version 3
IMAP	因特网邮件访问协议Internet Mail Access Protocol

图4-15 两种标准Email访问协议

虽然它们提供相同的基本服务，但这两个协议在许多细节上有所不同。特别是，每个协议使用各自的认证机制来进行用户自我识别。认证对确保用户不能访问其他用户的邮箱是必需的。

4.16 电子邮件表示标准

目前有两个重要的Email表示标准：

- RFC2822邮件报文格式。
- 多用途因特网邮件扩展（Multi-purpose Internet Mail Extension，MIME）。

RFC2822邮件报文格式。该邮件报文格式标准取自IETF标准文档RFC2822。这个格式简单明了：邮件报文被表示为一个文本文件，并由一个头部(header)、一空行和一个主体(body)组成。每个头部行的形式是：

关键词：信息

这里一系列关键词的定义包括：From:，To:，Subject:和Cc:等。另外，可以加入由大写字母X起始的头部行，不会影响邮件的处理。因此，一个邮件报文可能包括一个随机的头部行，如：

```
X-Worst-TV-Shows:  any reality show
```

多用途因特网邮件扩展MIME。前面说过，SMTP仅支持文本报文。MIME标准扩展了Email的功能，使其允许在报文中传输非文本数据。MIME规定了如何将二进制文件编码成为一系列可印刷的字符，包含在传统Email报文中，并在接收方解码。

显然MIME引入了早已十分流行的Base64编码标准，但它并不局限于只编码成一种特定的形式。而是允许发送和接收双方去选择各自方便的编码。为了规定所使用的编码，发送方在报文的头部中应包含一些额外的行。而且，MIME还允许发送方将报文分割成几个部分，并为每一部分各自规定编码形式。因此，使用MIME时，用户可以发送纯文本的报文，再附加图形图像、电子表格和音频剪辑等，各自采用各自的编码形式。接收方的Email系统则可以由自己来决定如何处理这些附件（例如，将其复制存盘或显示）。

事实上，MIME在Email头部加上两行内容：一行是声明已使用MIME来创建报文；另一行是指出在主体中如何包含MIME信息。例如，头部的这些行是：

```
MIME-Version: 1.0
Content-Type: Multipart/Mixed; Boundary=Mime_separator
```

以上两行指出：该报文是使用MIME 1.0版来创建的；在主体中报文的每个部分前将含有MIME分隔符（Mime_separator）的行。当MIME被用于发送一个标准的文本报文时，以上第二行就变成：

```
Content-Type: text/plain
```

MIME可以向后兼容那些不理解MIME标准或编码的Email系统，当然这样的系统是没有办法提取非文本附件的——它把主体当作单个文本块来处理。概括如下：

> 为了允许非文本的附件可以在一般的Email报文中传送，MIME标准插入了额外的头部行，附件被编码成可印刷的字母，并在每个附件前面出现一个分隔行。

4.17　域名系统

域名系统（Domain Name System，DNS）提供将人可阅读的符号名到计算机地址映射的服务。浏览器、邮件软件和其他大多数因特网应用都要使用DNS。由于映射不是通过单个服务器来完成的，DNS系统是一个有趣的客户-服务器交互的实例。换言之，域名信息分布在整个因特网上各地站点的大量服务器之中。应用程序需要解析域名的时候，该应用将成为域名系统的客户，并向域名服务器发送请求报文，域名服务器找到相应的IP地址并回送应答报文；如果不能回答请求，则服务器暂时变成另一个域名服务器的客户，直至发现可回答该请求的服务器。

在句法上，每个域名由一串通过句号分割的字母数字段组成。例如，普渡大学计算机科学系的一台计算机具有域名：

```
murdred.cs.purdue.edu
```

思科公司的一计算机域名：

anakin.cisco.com

域名是分等级的，其最重要部分是在右边，最左边的段（上面例子中的mordred和anakin）是一台个体计算机的名字；其他段用于识别拥有该计算机域名的组织。例如段purdue给出了大学名，Cisco给出了公司名。DNS并没有规定域名中段的数目，每个组织都可以选用多少个段来标识组织内部的计算机，并选择每个段所表达的含义。

域名系统只规定了最重要段（称为顶级域LTD）的值。顶级域由互特网名字和号码分配机构（Internet Corporation for Assigned Names and Numbers，ICANN）所控制，它委派了一个或多个域注册员来管理某一给定的顶级域，并负责核准特定的域名。有些LTD是一般性的，即意味着通常是可用的；另一些LTD被限制于特殊的组织或政府部门。图4-16列出了顶级DNS域示例。

域　名	分配给
aero	航空运输业
arpa	ARPA基础结构域
asia	针对亚有关业务地区
biz	商务
com	商业机构
coop	合作联盟
edu	教育机构
gov	美国政府机关
info	信息服务商
int	国际贸易组织
jobs	人力资源管理
mil	美国军事机构
mobi	移动内容提供商
museum	博物馆
name	个人
net	主要的网络支持中心
org	非营业机构
pro	专门（职）业认证机构
travel	旅行和观光业
国家代码	主权国家

图4-16 顶级DNS域示例和每个域分配的组织

一个组织可在已有的顶级域名下申请一个域名，例如大多数美国公司选择在com域下注册。这样，一个名为Foobar的公司可以在顶级com域下申请分配域名foobar，一旦请求被批准，Foobar公司将分配给域名：

foobar.com

一旦该域名被分配后，另一个名为Fuobar的组织可以申请foobar.biz或foobar.org，但不能是foobar.com了。进一步说，一旦分配了foobar.com域名，Foobar公司可以选择增加附加级并确定每级的含义。因此，如果Foobar公司有位于东海岸和西海岸的办公场所，可以选择如下域名：

computer1.east-coast.foobar.com

除熟悉的组织结构以外，DNS也允许组织使用地理位置注册，例如，国家研究与发展公司注册域名为：

cnri.reston.va.us

由于该公司位于美国Virginia州的Reston镇，因此公司的计算机名字结尾就以.us代替了.com。

其他一些国家也采用地理与组织相结合的域名。例如，英国大学注册的域名包括：

ac.uk

这里的ac是academic（学术机构）的缩写，而uk是英国的官方国家代码。

4.18 www开头的域名

有很多组织所指派的域名反映了计算机所提供的服务。例如，一台运行文件传输协议

(FTP) 的服务器可以命名为：

ftp.foobar.com

类似地，运行Web服务器的计算机可以命名为：

www.foobar.com

这样的域名易于记忆，但并不是一定要求的。特别是，用www来对运行Web服务器的计算机命名，也仅仅是一种习惯而已。任何计算机都可以运行Web服务器，即使其计算机的域名不含有www。而且，带有www开头域名的计算机也不必要求一定运行Web服务器。这里的要点：

域名中采用第一个标记来表示提供的服务（如www），仅仅是一种有助于人记忆的习惯。

4.19 DNS层次结构和服务器模型

DNS的主要特性之一就是它的自治性。DNS系统被设计成能允许每个组织给其所辖的计算机分配名字或更改这些名字，而无需通知中央权力机构。为了实现自治，允许每个组织运行DNS服务器来管理域名层次结构中的自身部分。因此，普度大学可运行服务器来管理以purdue.edu结尾的域名，IBM公司可运行服务器来管理以ibm.com结尾的域名。每个DNS服务器中必须含有连接到层次结构的上一层和下一层的其他域名服务器的有关信息。而且，一个给定的DNS服务器是可以被复制的，以致可能存在多个该服务器的物理拷贝。这种复制特性对于重负载运行的服务器（如提供顶级域信息的根服务器）来说总很有用的。在这种情况下，系统管理员必须确保所有复制的一致性，以准确地提供完全相同的信息。

每个组织可自由地选择其DNS服务器的构成细节。例如，一个只有少量计算机的小型组织可以与ISP签约委托运行DNS服务器；一个运行自己的DNS服务器的大型组织，可以把本组织内所有计算机的域名放在单台物理服务器上，也可以选择把域名划分到多台服务器上。又例如，图4-17所示为假想的Foorbar公司可能选择的DNS服务器构成，这里假设Foorbar下设一个肥皂子公司和一个糖果子公司。

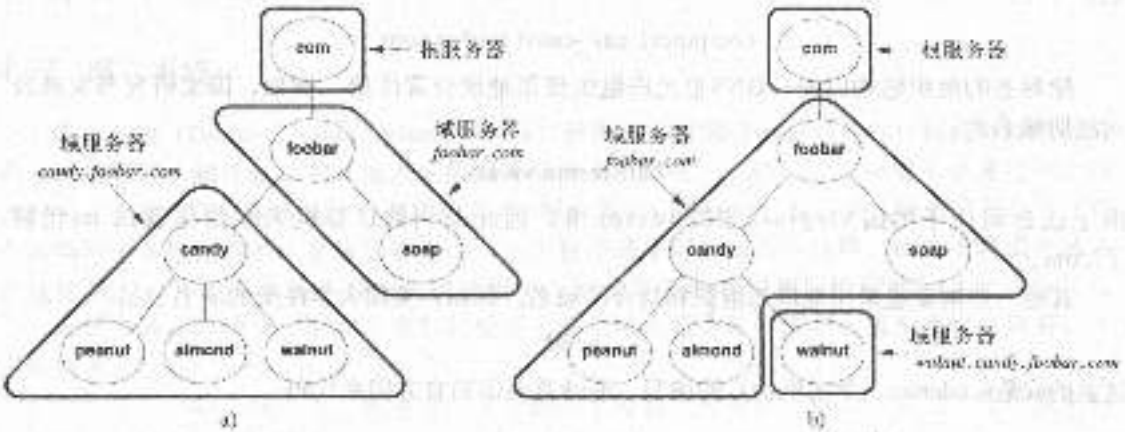


图4-17　一种假想的DNS层次结构和两种可能的服务器域名分配

4.20 域名解析

将域名翻译成相应IP地址的过程称为域名解析（name resolution），或者就说，把名字解析成地址，完成这项翻译工作的软件称为域名解析器（name resolver），或简称解析器。例如，在套接字API中，解析器是通过调用函数gethostbyname来完成的，此时解析器变成一个客户，与DNS服务器建立联系并返回一个应答给原请求者。

每个解析器配置一个或多个本地域名服务器的地址[⊖]。解析器形成一个DNS请求报文，并将报文发送给本地的域名服务器，接着等待服务器发回一个含有答案的DNS应答报文。解析器可以选择使用流模式或报文模式与DNS服务器通信，但解析器大多数被配置为使用报文模式，因为它对小的请求报文承担更小的开销。

作为域名解析的例子，参考图4-17a所示的服务器层次结构，并假设肥皂千公司的一台计算机发出对域名chocolate.candy.foobar.com的解析请求。解析器按配置将请求发给指定的本地DNS服务器（即foobar.com的DNS服务器），虽然它无法答复该请求，但它知道联系下一级candy.foobar.com的DNS服务器，而candy.foobar.com的DNS服务器却能够给出答案。

4.21 DNS服务器中的缓存处理

形成缓存处理基础的引用局部性（locality of reference）原则，以下面两种方式应用于域名系统：

- 空间上：用户更趋向于经常地查找本地计算机域名。
- 时间上：用户更趋向于重复地查找一些相同的计算机域名。

我们已经看到DNS如何利用空间局部性（即名字解析）首先联系本地DNS服务器。为了体现时间局部性，DNS服务器缓存了所有的查找记录。算法4-4归纳了这两种处理过程。

```
算法4-4
Given:
  A request message from a DNS name resolver
Provide:
  A response message that contains the address
Method:
  Extract the name N from the request
  if (server is an authority for N) {
    Form and send a response to the requestor
  else if (answer for N is in the cache) {
    Form and send a response to the requestor
  else {  /* Need to look up an answer */
    if (authority server for N is known) {
      Send request to authority server;
    } else {
      Send request to root server;
    }
    Receive response and place in cache;
    Form and send a response to the requestor;
  }
```

算法4-4 DNS服务器进行域名解析的步骤

⊖ 在讨论缓存处理时，我们会明显看到首先联系本地DNS服务器的重要性。

根据算法4-4，当到达的域名请求是权威域名服务器的解析权限范围以外时，就会产生进一步的客户—服务器交互。该服务器暂时变为另一域名服务器的客户。当其他的域名服务器返回结果时，原服务器就将结果缓存起来，并给发出请求的解析器回送一份该结果的副本。因此，除了要知道该层级以下的所有域名服务器的IP地址外，每个域名服务器必须知道根服务器的IP地址。

所有缓存处理的基本问题，都与被缓存的时间项长短有关系——如果某个项被缓存的时间太长，该项内容就会变得过时。DNS解决该问题的方法是：通过安置一台权威服务器来指定每项的缓存超时时间。因而，当本地服务器查找某个域名时，其响应由指定缓存超时的资源记录（resource record）和应答其同组成。在服务器缓存一个应答的任何时候，它都要尊重由资源记录所指定的超时值。这里的要点是：

> 因为每个由权威服务器产生的DNS资源记录都指定了一个缓存超时值，所以当一个缓存项失效时就可以将它从DNS缓存中删除。

DNS缓存处理不局限在服务器，解析器同样也可以缓存一些项。事实上，大多数计算机的解析器软件都会缓存来自一些由DNS查找到的结果。这也意味着对相同域名的连续请求不需要再使用网络处理，因为解析器可以从本地磁盘的缓存中得到相应的请求结果。

4.22　DNS记录项的类型

DNS数据库中的每个记录由3个项组成：域名、记录类型和值。记录类型指出该值如何解释（例如，该值是一个IP地址）。更重要的是，一个发给DNS的询问必须指明域名和类型，服务器只返回一个匹配该查询类型的一个绑定。

主要的类型是域名到IP地址的映射，DNS把这样的绑定归类为A类型。而且A类型查找常用于如FTP、ping或WWW浏览器等应用。DNS还支持一些其他的类型，包括指明邮件交换器（Mail eXchanger）的MX类型，当查找Email地址中的域名时，SMTP就使用MX类型。服务器返回的结果应与请求类型相匹配，因此Email系统只接收与MX类型相匹配的结果。重点归纳：

> DNS服务器中的每个记录都有一个类型。当解析器查找一个域名时，要指明它所需要的类型，而DNS服务器也仅返回与所指类型相匹配的记录。

由于返回的地址要取决于类型，所以DNS类型机制会产生意想不到的结果。例如，一个公司可能决定对Web和Email系统都使用corporation.com域名，那么在使用DNS过程中，该公司就有可能把负荷分流到不同的计算机上，通过匹配A类型去查找一台计算机和匹配MX类型去查找另一台计算机，但这种方案的缺点是，有点违背人的直觉——即使不可能访问Web服务器或ping不通corporation.com的计算机，但还可能发送Email给corporation.com的计算机。

4.23　别名和CNAME资源记录

DNS还提供一种CNAME类型，该类型类似于文件系统中的符号链接——该记录为另一个DNS记录提供一个别名。为了理解别名是如何使用的，假设Foobar公司有两台计算机，域名分别为hobbes.foobar.com和calvin.foobar.com。进一步假设Foobar决定在hobbes计算机上运行一个Web服务器，并希望运行公司Web服务器的计算机能脱离在域名中使用www名字的习惯。虽然该公司可以选择将hobbes计算机重命名，但还有更简单的方法是公司可以为www.foobar.com生成一个指向hobbes的CNAME项。当有解析器发出对www.foobar.com的请求

时，服务器就会给它返回计算机hobbes的地址。

使用别名特别的方便，因为它允许一个组织更改用于特殊服务的计算机而不必更改该计算机的域名或地址。例如，Foobar公司可以将Web服务从计算机hobbes移至calvin，方法是移动服务器并改变DNS服务器中的CNAME记录——两台计算机都还保留了它们原先的域名和IP地址。利用这种别名机制，也可以将多个别名跟一台计算机联系起来，因此，Foobar公司可以在同一台计算机上运行FTP服务器和Web服务器，并创建CNAME记录为：

www.foobar.com
ftp.foobar.com

4.24 缩写与DNS

DNS并不能处理缩写域名——服务器只对全名做出响应。然而，大多数解析器可以通过配置一系列后缀来允许用户缩写域名。例如，Foobar公司的每个解析器可以编程分两次查找一个域名：一次输入原名，另一次附加后缀foobar.com。假如用户输入一个完整的域名，则本地服务器执行处理，并将返回相应的地址。如果用户输入的是一个缩写名，解析器首先尝试解析该域名，但由于这样的域名不存在，所以会收到出错信息。然后，解析器就尝试附加一个后缀，再进行查找。因为解析器运行在用户PC机上，所以允许每个用户选择尝试后缀的顺序。

允许每个用户配置其解析器来处理缩写的做法，存在一个缺点：一个用户输入的特定域名可能不同于另一个用户输入的该域名，因此假如某些用户相互交流域名时（如通过在Email报文中发送域名），每个用户都必须非常小心地指明是全名而不是缩写。

4.25 国际化域名

由于DNS使用ASCII字符集，所以它不能存储用ASCII以外字母表示的域名。特别是，例如俄语、希腊语、汉语和日语，这些语言都存在ASCII不能表达的字符，许多欧洲语言还使用了不能用ASCII表示的读音符（diacritical mark）。

这些年，IETF一直在讨论修订和扩展DNS以容纳国际化域名。在考虑了许多种方案以后，IETF选择了一种称为国际化域名应用（Internationalizing Domain Names in Applications，IDNA）的方法。IDNA仍使用ASCII来存储所有的名字，而不去更改底层DNS——当给出的域名包含非ASCII码的字符时，IDNA把该域名转换成一个ASCII字符串，然后将其存储到DNS系统中。在用户查询域名时，使用同样的转换把该域名转换为ASCII字符串，并将其放到DNS询问中。实质上，IDNA依靠应用软件完成用户所见的国际字符集与DNS所用的内部ASCII形式之间的转换。

转换国际域名的规则是件复杂的事情，并使用统一字符编码标准Unicode（采用双字节对字符进行编码）。[6]大体上，转换要应用于域名中的每个标记，并形成如下的标记形式：

xn--α-β

这里，xn--是一个预留的4字符串，它指明该标记是一个国际名字；α是来自可用ASCII表示的原标记的字符子集；β是一个附加的ASCII字符串，它告诉IDNA应用程序如何将非ASCII字符插入到α中以形成该标记的可显示（或可打印）形式。

广泛运用的最新版浏览器Firefox和Explorer已经实现了IDNA，所以可以接纳和显示非

[6] 用于非ASCII标记的转换算法称为punycy算法，所形成的字串称为punycode（punycode）。

ASCII字符域名。如果应用程序没有实现IDNA，则会输出显示出一串乱码，使用户感到很奇怪。也就是说，当一个没有实现IDNA的应用软件要显示一个国际域名时，用户将会看到上述的内部形式，包括起始串xn--和随后的a和b部分。

概括如下：

> 国际域名的IDNA标准将每个标记编码成一个ASCII字符串，并依靠应用程序来完成在用户期望的字符集和存储于DNS的编码形式之间的转换。

4.26　可扩展表示

本章涉及的所有传统应用协议都采用固定的表示方法，也就是说，应用协议规定了客户与服务器进行交换的一组确定报文，以及伴随报文所传输数据的确定形式。这种固定方法的主要缺点是对表示方法要进行改变时具有一定的难度。例如，由于Email标准将报文内容限制为文本，所以如果要加入MIME扩展，就要对它进行重大的改变。

另一种表示方法是一种允许发送方规定数据格式的可扩展系统。有一种可扩展表示标准已经被广泛接受，即可扩展标记语言（Extensible Markup Language，XML）。从某种意义上来说，XML类似于HTML，两者都是将标签嵌入到文本文档。但不同于HTML的是，XML中的标签未规定优先顺序，也不与格式命令相对应。XML的做法是描述数据的结构并对每个域提供名字，XML中的标签是平衡对称的，即每个<X>标签出现后必须要有后面的</X>结束。而且，由于XML对标签并不赋予任何含义，所以标签名可以按需要来选定。特别是，可以选择更合适的标签名字，以使数据更容易分析或访问。例如，两个公司同意交换公司电话本，他们可以定义XML格式有如下数据项：职工姓名、电话号码和办公室；公司还可以选择进一步把姓名划分名和姓，如图4-18所示。



图4-18　一个公司电话本的XML例子

4.27　本章小结

为标准化服务所要求的应用层协议，要为通信交互的数据表示和数据传送这两个方面作出定义。应用于WWW中的表示协议包括超文本标记语言（HTML）和URL标准，Web传输协议就是超文本传输协议（HTTP），它规定了浏览器如何与Web服务器通信以实现下载或上传内容。为了加快下载速度，浏览器对网页内容进行高速缓存，并使用HTTP的HEAD命令来请求关于网页的状态信息。如果被缓存的网页版本是最新的，浏览器则使用已缓存的版本；否则，浏览器就发出GET请求下载一个更新的副本。

HTTP采用文本型报文，来自服务器的每个响应都以描述该响应的一个头部开始，头部中的行又是用ASCII表示的指示状态（如请求是否出错）的一个数值来开头。紧随在头部后面的数据可以含有任意的二进制位。

文件传输协议（FTP）经常被用于文件下载，FTP要求客户登录到服务器的系统，对于公

用的文件访问，FTP支持匿名（anonymous）登录并使用guest作为登录密码。FTP最有趣的方面，是它采用了不寻常的连接方式：一客户端创建用于发送一系列命令的控制连接，而当服务器需要发送数据时（如文件下载或列出目录），该服务器则扮作客户，原客户却扮作服务器，并由原服务器向原客户端发起一个新的数据连接。一旦一个文件传送完毕，就关闭数据连接。

电子邮件使用了3种类型的应用层协议：传输、表示和访问。简单邮件传输协议（SMTP）是关键的传输标准，但仅能传输文本报文。Email有两种表示标准：RFC2822定义的由空行分割头部和主体的邮件报文格式；多用途因特网邮件扩展（MIME）标准定义的作为一个邮件报文附件来发送二进制文件的机制。MIME插入额外的头部行来告诉接收者怎样解释报文。MIME要求发送者将文件编码成可打印的文本形式。

Email访问协议（如POP3和IMAP）允许用户去访问邮箱。因为用户可以让ISP来运行Email服务器和维护自己的邮箱，所以访问邮箱就变得十分流行。

域名系统（DNS）提供从人可阅读的名字到计算机地址的自动映射能力。域名系统由许多服务器组成，而每个服务器控制着名字空间的一个部分。服务器按层次结构来配置，并且服务器知道层次结构中各个服务器的特定位置。

DNS使用高速缓存来保持它的运行效率：当一个权威服务器提供一个应答时，传输该应答的每个服务器都在缓存中放置一个副本。为了防止缓存的副本过时失效，域名的权威提供者规定了一个域名可以被缓存的时间长度。

练习题

4.1 应用层协议要规定哪些细节？

4.2 为什么用于标准服务的协议文档描述要独立于实现？

4.3 应用协议有哪两个关键的方面，各自包含哪些内容？

4.4 试给出展示应用协议两个方面的Web协议实例。

4.5 请归纳HTML的特征。

4.6 URL的4个组成部分是什么？用什么符号来分隔各个部分？

4.7 HTTP的4个请求类型是什么？各类型什么时候使用？

4.8 浏览器如何知道HTTP请求语法上是否不正确或者参考项是否不存在？

4.9 浏览器缓存是什么？为什么要使用缓存？

4.10 试描述浏览器确定是否使用缓存中网页的步骤。

4.11 浏览器能使用HTTP以外的传输协议吗？并解释为什么。

4.12 当用户要请求一个FTP目录列表时，会形成几个TCP连接？试解释。

4.13 判断对或错：当用户运行FTP应用时，应用软件要起到客户和服务器的双重作用。请解释你的回答。

4.14 FTP服务器如何知道用于数据连接的端口号？

4.15 根据原来的Email模式，如果用户计算机不能运行Email服务器，能否接收邮件？请解释。

4.16 说出用于Email的3类协议，并分别给予描述。

4.17 SMTP的特征是什么？

4.18 SMTP能依靠自身传送一行中包含句号的Email报文吗？并解释为什么。

4.19 Email访问协议用在什么地方？

4.20 有哪两种主要的Email访问协议？

4.21　为什么要发明MIME?

4.22　域名系统的主要用途是什么?

4.23　假如ISO已经分配了N个国家的域编码，那么有多少个顶级域存在?

4.24　判断对或错：Web服务器必须要有一个以WWW开始的域名，试解释。

4.25　判断对或错：一跨国公司可以选择划分域名层次，这样公司就可以有一个欧洲域名服务器、亚洲域名服务器和北美域名服务器。

4.26　什么时候域名服务器要向权威域名服务器发送请求?什么时候域名服务器不需向权威域名服务器发送请求?

4.27　判断对或错：对于一个给定域名，域名服务器可能返回不同的IP地址，这取决于该查询指明的是Email服务还是Web服务，试解释。

4.28　IDNA标准要求改变DNS服务器或DNS客户吗?请解释。

4.29　搜索Web去找出迭代的DNS查找情况。在什么情况下使用迭代查找?

4.30　XML是如何允许由应用软件来指定域（如名字和地址）的?

第二部分

数据传输

传输介质、信号、码位、载体和调制解调器基础

第5章 数据通信概述

5.1 引言

本书前面讨论了网络编程方面的知识，并描述了因特网的应用。有关套接字编程的章节阐述了操作系统为网络应用软件开发所提供的API，从而表明：程序员只要通过调用相应套接字API，就可以开发出利用因特网的各种应用，而无需去理解网络的底层通信机制。下面将学习支持通信的复杂的协议和技术。可以看到，理解这些复杂的协议和技术，能够帮助程序员写出更好的程序代码。

本书这一部分的各章内容，介绍信息在物理介质（如导线、光纤和无线电波）上的传输原理。我们将看出，虽然在细节技术上有所不同，但有关信息和通信的基本思想都适用于所有的传输形式。我们还将理解到，数据通信为获得对通信系统工作机制的统一解释，提供了概念的和分析的工具。更重要的是，数据通信原理还将告诉我们理论上可能的传输能力，以及物理世界中的现实又是如何限制实际传输系统能力的。

本章对数据通信进行概述，并阐述如何由几个概念部分来构成一个完整的通信系统。后面的每一章将详述一个概念。

5.2 数据通信所涉及的学科

数据通信涉及那些学科范畴呢？如图5-1所示，这个课题是3个学科的有机结合。

因为信息的传输是通过物理介质进行的，所以数据通信需要接触物理学，在这方面该课题主要涉及电流、光以及其他形式的电磁辐射问题。因为信息要经过数字化转变成为数字数据才能传输出去，所以数据通信要用到数学以及各种形式的分析方法。最后，因为最终的目标是要找到实用的方法来设计和构建传输系统，所以数据通信更加注重开发出电子工程师可用的各种技术。因此，问题的关键点是：

> 虽然数据通信包含了物理学和数学方面的概念，但该课题不仅仅是两种学科的理论，而是更注重于提供用于构建实际通信系统的理论和技术基础。

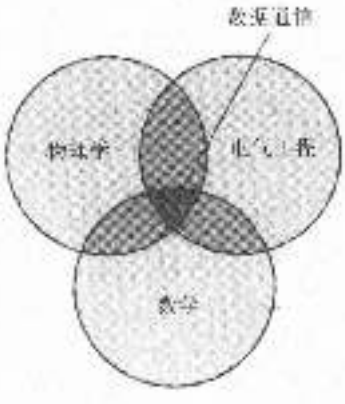


图5-1 数据通信课题属于物理学、数学和电气工程的交集

5.3 课题动机与范围

有3个考虑要点足以给出数据通信的动机，并有助于界定数据通信的研究范围。它们是：

- 信息源可以为任意类型。
- 传输系统采用物理系统。
- 多个信息源可以共享下层传输介质。

第一点是特别考虑了多媒体应用的普及：被传输的信息不仅仅局限于储存在电脑上的二

进制数据。还可以是来源于现实世界的各种信息，包括音频和视频信息。因此，数据通信系统必须能够理解各种可能的信息源、信息形式以及实现各种形式之间的转换。

第二点指出数据通信系统必须利用某种自然的物理现象（如电流和电磁辐射等）来进行信息传输。这样，理解可用传输介质的类型和属性就显得很重要。而且，还必须理解每种介质的哪些物理现象可以被利用来传输信息，并且解数据通信与下层传输介质之间的关系。最后，必须理解实际硬件限制、传输中可能出现的问题以及检测和解决相关问题的技术。

第三点指出“共享”是根本。我们将会看到，的确在大多数计算机网络系统中“共享”都扮演着根本的规则，也就是说，通常网络都能允许多对通信实体通过一条给定的（共享）物理介质来进行通信。因此，理解各种可能被共享的底层设施及其优缺点，以及相应的通信模式是很重要的。

5.4 通信系统的构成

为了更好地理解数据通信，我们可以想象一个正在工作的通信系统，该系统允许多个信息源分别独立地往不同的目的地址发送信息。这样的系统其通信过程可以看似相对简单：每个信息源独立地收集和准备要被传输的信息，并通过共享的物理介质进行传输。类似地，在终端则需要一种机制来提取被传输信息并分发到相应的目的地。图5-2为数据通信的简化视图。

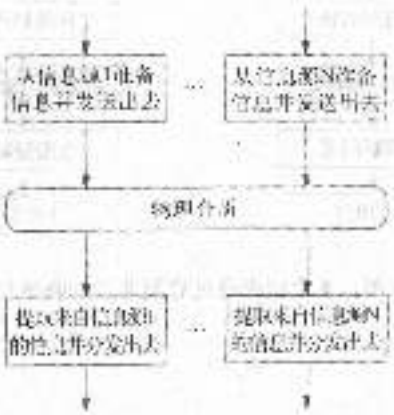


图5-2 数据通信简化视图：多个信息源通过共享信道往不同目的地发送信息

在实际中，数据通信系统要比图5-2所示的情况复杂得多。由于信息可以来源于很多不同类型的信息源，所以处理每种信息源的技术也各不相同。在发送前，信息必须经数字化，而且为了防止差错必须添加冗余信息。如果要考虑保密，还可能需要对信息进行加密。为了通过共享的通信机制发送多个信息流，必须要标识不同的源信息，然后将所有源信息混合在一起传输。因此，需要有相应机制来标识每个信息源，从而保证经过传输以后不同源的信息能够被有效地区分开来，以免它们被混淆。

为了说明数据通信的主要方面，工程师们提出了一个概念框架用于说明数据通信系统的各个子课题，以及各个子课题是如何结合在一起而形成一个完整的数据通信系统。其基本思想是：框架的各个部分可以独立研究，一旦充分理解各个部分之后，就可以完整地解整个课题了。如图5-3所示为这个概念框架，并表示了各个部分如何互相关联而组织起一个完整的通信系统。

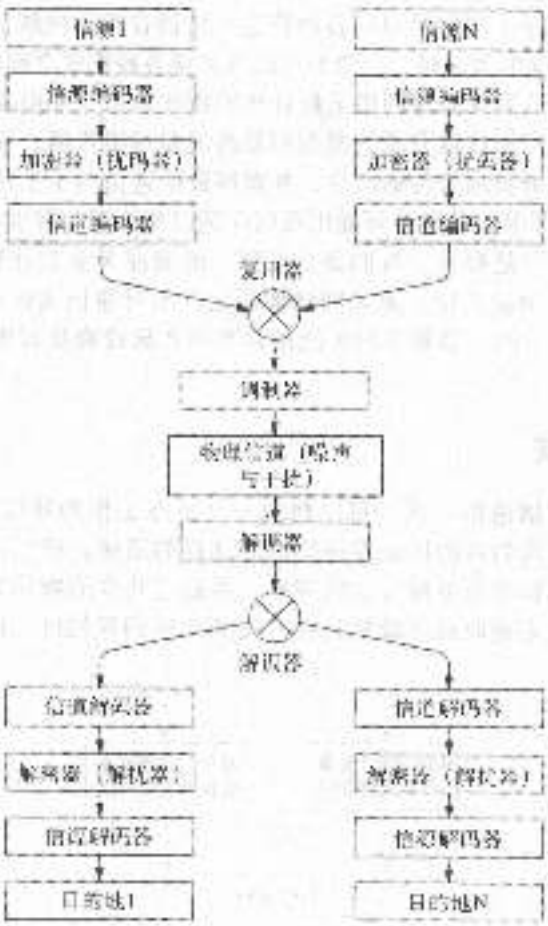


图5-3　数据通信系统概念框架，多个信源通过共享的物理信道往多个目的地发送信息

5.5　通信系统各子课题

图5-3中的每一个方框对应于数据通信中的一个子课题。下面段落内容解释相关术语，后面的每一章都将研究一个子课题。

- 信息源。信息源[⊖]既可以是模拟的也可以是数字的，其重要概念包括各种信号特征（如振幅、频率、相位以及是否具有周期性）。此外，信息的模拟与数字表示之间的转换也是这个子课题所关注的焦点。
- 信源编码器和解码器。一旦信息被数字化后，就可以对其数字表示进行转换或变换，其重要概念包括数据压缩和用于通信的后验性能。
- 加密器和解密器。为了保护信息不被窃密，信息可以在传输前先加密，在接收后再解密。其重要概念包括加密技术和算法。
- 信道编码器和解码器。信道编码是用来检测和纠正传输错误的，其重要的概念包括检测和限制差错的方法，以及实际中可采用的技术（如在计算机网络中使用的奇偶校验、校验和以及循环冗余码等）。

⊖　有时简称信源。——译者注

- 复用器和解复用器。复用指多个信源同时在共享介质上传输信息，其重要概念包括多信源轮流使用介质的同步共享技术等。
- 调制器和解调器。调制指信息发送时把需要传输的数据转化为相应的电磁信号，其概念包括模拟和数字调制的技术方案及其设备。实施调制和解调的设备称为调制解调器（MODEM，调制器和解调器英文字头的缩写）。
- 物理信道与传输。本子课题包括传输介质和传输模式，其重要概念包括带宽、电磁噪声、干扰和信道容量以及传输模式（如串行传输和并行传输）等。

5.6 本章小结

因为课题涉及数字信息及其在物理介质上的传输，所以数据通信与物理学和数学有着密切关系，其关注的焦点是能让电气工程师设计出实用的通信机制。

为了简化理解，工程师已经设计了一种数据通信系统概念框架，它把整个课题分解为一系列子课题来解决。后面的每一章将讨论一个子课题。

练习题

5.1 数据通信的研究遵从了那3个基本假设？

5.2 数据通信的动机是什么？

5.3 数据通信的概念模型包括哪些构成部件？

5.4 数据通信系统中由哪个部件处理模拟信号的输入？

5.5 在数据通信系统中由哪个部件来防止数据由于传输差错而被破坏？

第6章 信息源和信号

6.1 引言

从第5章我们开始学习数据通信。这是所有联网技术的基础。在第5章介绍了数据通信课题，给出了一个数据通信的概念框架，并介绍了这个课题的各个重要方面，解释了各个方面如何结合在一起，对各个概念部件也作了简单描述。

从本章开始对数据通信进行更为详细的讨论。本章先研究有关信息源和载送信息的信号特征方面的课题，后面章节接着讨论数据通信其他方面的课题。

6.2 信息源

回顾一下，通信系统从一个或多个源点（source）接收输入信息，并把它从源点传递到指定的宿点（destination），即目的地。对于一个网络（如个域网特网）来说，信息的源点和宿点就是产生和消化数据的一对网络应用程序。不过，数据通信理论专注于低层的通信系统，可以适用于任何信息源（以下简称为"信源"）。例如，除了传统的计算机外部设备（如键盘、鼠标）外，信源还可以包括传声器、传感器和测量设备（如温度计、计量仪等）。类似地，信宿点可以包括音频输出设备（如耳机、扬声器），以及诸如发光二极管等设备。这里的要点是：

> 通过数据通信的学习，重要的是记住，信源可以是任意的，可以包括各种设备，而不仅仅是计算机。

6.3 模拟与数字信号

数据通信涉及两种类型的信息：模拟的和数字的。模拟信号由连续变化的数学函数来表征——当输入从一个值到下一个值改变时，其输出也通过所有可能的中间值而改变。与之相反的是，数字信号则具有固定的一组有效电平值，它的每一次改变就是瞬间地从一个有效电平跃迁到另一个有效电平。图6-1给出了模拟源和数字源信号如何随时间变化的示例，说明了相应的概念。在图6-1中，模拟信号有可能是人们测量传声器输出所得到的，而数字信号则可能是人们测量计算机键盘的输出所获得的。

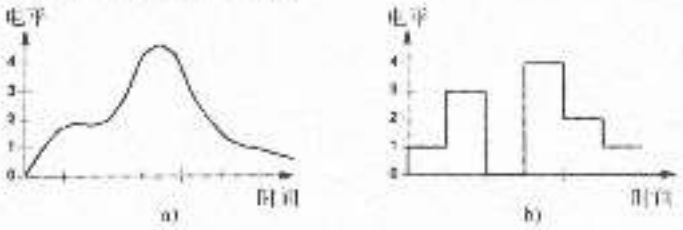


图6-1 示例图：a) 模拟信号；b) 数字信号

6.4 周期信号与非周期信号

依据它的重复性，信号可以大致分为周期信号和非周期信号。例如，图6-1a所示的都是非周期信号，因为在所示的时间间隔内信号没有重复。图6-2所示的信号是周期信号（即不断重复出现的信号）。

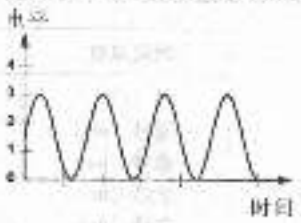


图6-2 周期信号的重复出现

6.5 正弦波与信号特征

我们将看到，在大多数数据通信的分析中都要用到正弦曲线三角函数，特别是正弦（sine，通常缩写为sin），正弦波在信源中显得特别重要，因为很多自然现象都会产生正弦波，例如，当传声器接收到人耳能听见的声音时，它的输出就是正弦波，类似地，电磁辐射也可以用正弦波表示。正弦波对应于在时间上进行振荡的信号，如图6-2所示的波形。这里的要点是：

> 因为许多自然现象产生与正弦波的时间变化函数相符合的信号，所以正弦波在信号的输入处理中起着根本作用。

与正弦波相关的信号具有以下4个重要特征：

- 频率：每单位时间（通常是s）的振荡次数。
- 振幅：最大与最小信号幅值之差。
- 相位：正弦波起点相对于参考时间的移位值。
- 波长：信号在介质中传播时一个周期的长度。

波长由信号在介质中传播的速度来决定（因它是底层介质的函数），其他3个特征可以用数学表示。振幅是最容易理解的，回想一下，正弦函数sin(ωt)的值在−1与+1之间变化，振幅为1，因此当值乘以A时，所产生的波形振幅就是A。在数学上，相位是正弦波沿着X轴向右或向左的偏移，那么sin(ωt+Φ)的相位就是Φ。信号的频率由每秒内正弦波的周期数目来测量，度量单位是赫兹（Hertz）[⊖]。一个完整的正弦波需要2π弧度，因此如果t以s为时间单位并且ω=2π，那么sin(ωt+Φ)的频率就是1Hz，图6-3表示了这3个数学特征。

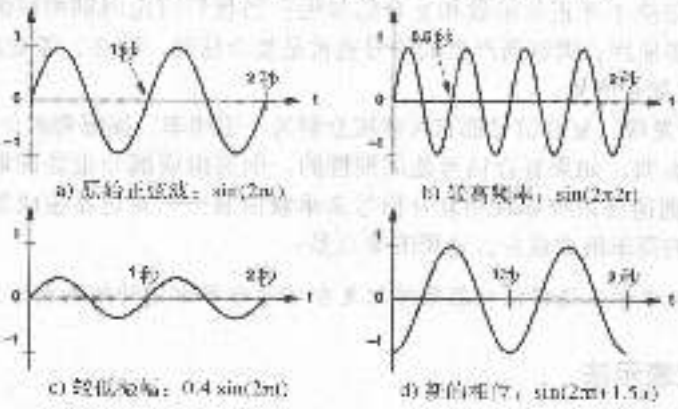


图6-3 频率、振幅与相位特征示意图

频率是周期的倒数。在图6-3a的例子中，正弦波的周期是T=1s，频率则是1/T=1Hz；在图6-3b的例子中，正弦波的周期是T=0.5s，因此它的频率是1/T=2Hz。这两个例子都是非常低的

⊖ 缩写为Hz。——译者注

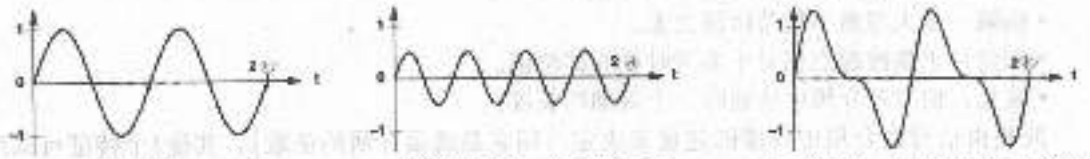
频率。典型的通信系统通常使用以每秒百万周期数计的高频率。为了清楚表达高频率，工程师使用以μs等非常小的时间单位表示时间，或者以MHz等为单位表示频率。图6-4列出了时间标度和通常使用的频率前缀。

时间单位	值	频率单位	值
秒（s）	10^{0}s	赫兹（Hz）	10^{0}Hz
毫秒（ms）	10^{-3}s	千赫兹（KHz）	10^{3}Hz
微秒（μs）	10^{-6}s	兆赫兹（MHz）	10^{6}Hz
纳秒（ns）	10^{-9}s	千兆赫兹（GHz）	10^{9}Hz
皮秒（ps）	10^{-12}s	太赫兹（THz）	10^{12}Hz

图6-4　时间和频率单位的前缀与缩写

6.6　复合信号

图6-3所示的各个信号都归类为简单信号，因为它们都是由单个正弦波构成的，不能再进行分解。事实上，大多数信号都归类为复合信号，因为这些信号可以分解为一组正弦波。例如，图6-5所示为由两个简单正弦波相加而形成的一个复合信号。



a）简单信号1：sin(2πt)　　b）简单信号2：0.5*sin(2π2t)　　c）复合信号：sin(2πt)+0.5*sin(2π2t)

图6-5　由两个简单信号形成一个复合信号

6.7　复合信号和正弦函数的重要性

为什么数据通信经常并且正弦函数和复合信号呢？当我们讨论调制和解调的时候，将会理解到其中的一个主要原因：调制所产生的信号通常是复合信号。现在，重要的是理解其动机：

- 调制通常形成复合信号。
- 数学家傅里叶发现：复合信号都有可能被分解为一组频率、振幅和相位不同的正弦函数。

傅里叶的分析表明，如果复合信号是周期性的，则其组成部分也是周期性的。因此我们将看到，大多数数据通信系统都利用复合信号来承载信息——发送器生成复合信号，接收器把信号分解为原始的简单组成成分。这里的要点是：

傅里叶发明的数学方法使得接收器能够把复合信号分解出它的组成成分。

6.8　时域与频域表示法

由于复合信号是周期性的东西，所以对它进行过广泛的研究，并发明了好几种表示它们的方法。我们已经在前面的图示中看到其中的一种表示方法，即作为时间函数的信号图形。工程师把这种表示图形称为在时域（time domain）上的信号表示。

相对于时域表示的另一种表示法称为频域（frequency domain）表示。频域图表示出构成一个复合函数的一组简单正弦波成分，Y轴给出振幅，X轴给出频率。这样，函数Asin(2πft)可

以用一个位于x=t，高为A的一根线段来表示。例如，图6-6频域图表示了图6-5c中的复合信号。

频域图表示出一组简单周期信号，但它也可以表示非周期信号，不过非周期信号的表示对于理解当前主题并不是必须的。

频域表示的优点之一是紧凑，与时域表示相比，频域表示既小又易读，因为每一个正弦波只占据X上的一个点。当一个复合信号由很多简单信号构成时，这个优点就更明显了。

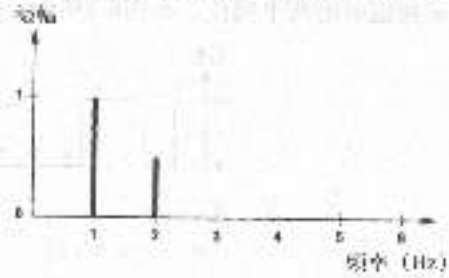


图6-6 sin(2πt)与0.5sin(2π2t)的频域表示法

6.9 模拟信号的带宽

几乎每个人都听说过“网络带宽”这个词，并且都懂得一个网络最好具有高的带宽。后面我们将讨论网络带宽的定义，现在先解释为一个相关的概念——模拟带宽（analog bandwidth）。

我们把模拟信号的带宽定义为：构成信号的最高频率与最低频率的差（最高频率与最低频率由傅里叶分析得到）。在图6-5c的第3个例子中，傅里叶分析得到频率为1Hz和2Hz的信号，意味着带宽是频率之差，为1Hz。当需要计算带宽时，频域图的优点就非常明显。因为在频域图中最高频率与最低频率是显而易见的。例如，在图6-6中带宽为1就非常明显。

图6-7表示出一个测量单位为千赫兹（KHz）的频域图，这种频率范围是人耳可以听见的频率范围。在图6-7中，信号的带宽就是最高频率与最低频率之差（5KHz−1KHz = 4KHz）。

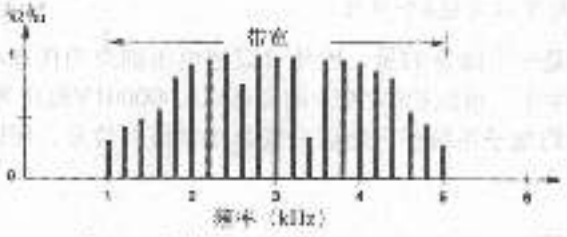


图6-7 带宽为4KHz的模拟信号频域图

概括如下：

> 模拟信号的带宽是其组成部分的最高频率与最低频率之差，如果信号由频域图表示，则带宽计算就非常简单。

6.10 数字信号与信号电平

我们曾经说过，信息除了可以用模拟信号表示外，也可以用数字信号表示。再进一步定义，所谓数字信号就是固定的一组有效电平值的集合，信号在任何时候只处于其中一个有效电平值上。一些系统采用电压来表示数字值，正电压对应于逻辑1，零电压对应于逻辑0，例如，可以用5V电压表示逻辑1，用0V电压表示逻辑0。

如果只使用两个电压值，那么每个电平可分别对应于一个数据位（0或1）[⊖]。不过，有些物理传输机制可以支持两个以上的电平值，这时每个电平可以表示多个数据位。例如，考

⊖ bit又译为“比特”、“码位”、“码元”或“位元”等，后文中也有会出现这几个词汇，可视为同义词。——译者注

这一个使用4个电平的系统。这些电平分别是：−5V、−2V、+2V、+5V，每个电平值分别表示数据中的两个码位，如图6-8所示。

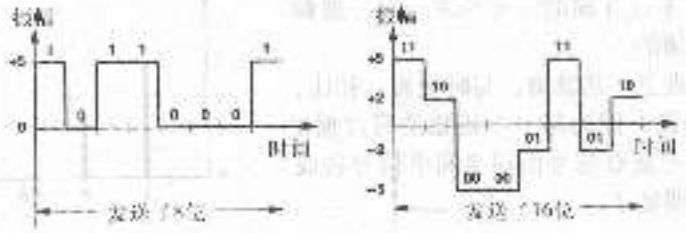


图6-8　a)使用两个电平的数字信号；b)使用4个电平的数字信号

如图6-8所示，使用多电平的主要优点是具有一次可以发送多个码位的能力。例如，在图6-8b中，−5V可以代表两个码位00，−2V可以代表01，+2V可以代表10，+5V可以代表11。因为使用了多电平，原来的每个码位间隔就可以传输两个码位，这意味着：在单位时间内，图6-8b中4电平表示所传输的信息是图6-8a中两个电平表示所传输信息的两倍。

所要求的电平数目与发送的位数之间的关系是非常直接的。每个可能的位组合都必须有一个对应的电平来表示。因为n个码位共有2^n个位组合，所以通信系统必须使用2^n个电平来表示所有的n位码。根据如下：

使用两个信号电平的通信系统在一个给定时间内仅可以发送一个码位；而支持2^n个电平的系统一次却可以发送n个码位。

人们假设认为电压是一个随意的量，由此可以把电压细分为任意小的增量值，从而获得任意多的电平数。在数学上，可以把0V到1V的电压以0.000001V电压为单位增量，划分出100万个电平。可惜，实际的电子系统不可能区分任意小差值的信号，所以实际系统都限制使用较少的电平个数。

6.11　波特率与比特率

在给定时间内能发送多少数据呢？答案取决于通信系统的两个方面。如前所述，数据能被发送的速率取决于信号电平的数目。另一个因素也同样重要，即系统维持在某一电平上的时间长短。例如，图6-8a中X轴表示时间，时间被分为8段，每一段间隔内发送1位数据。如果修改通信系统使得只用一半时间就能发送1位数据，那么在给定的时间内发送的位数就是原来的两倍。这里的要点是：

要增加在给定时间内的数据传输量，另一种途径就是减小一个信号电平的持续时间。

实际系统中的硬件都对信号电平的最短持续时间有限制——如果信号电平不能维持足够长的时间，接收端硬件就会检测不到信号。有趣的是，通信系统所接受的测量方法并不是规定时间的长度，而是去测量相反的东西，即每秒内信号能改变的次数。定义为波特（baud）。例如，如果一个系统要求信号在一个电平上维持0.001s，那我们就说这个系统工作于1000baud的速率。

这里很关键的一点是：波特率和信号的电平数两者共同决定了比特率[⊖]。如果一个系统

⊖　又称为"位速率"。——译者注

有两个信号电平并以1000比特工作，则系统正好每秒可以传输1000比特（位）。然而，如果一个按1000比特工作的系统具有4个信号电平，那么这个系统每秒可以传输2000比特（因为4个信号电平系统每次可以传输2比特）。公式（6.1）表示了比特、信号电平与比特率的关系。

$$\text{比特/秒} = \text{波特} \times \lceil \log_2(\text{电平数}) \rceil \tag{6.1}$$

6.12 数字—模拟信号转换

如何将数字信号转换为等效的模拟信号呢？回想一下，根据傅里叶变换原理，任意曲线可以表示为正弦波的组合，在此组合里的每个正弦波都具有特定的振幅、频率和相位。这为适用于任何曲线，所以傅里叶原理也适用于数字信号。从工程上的观点，傅里叶变换的结果对于数字信号是不实用的，因为要准确表示一个数字信号需要正弦波的一个无限集合。

工程上采取一个折中方案：从数字到模拟信号进行近似转换，也就是说，工程师们构建出来的设备只能产生近似地逼近于数字信号的模拟波形，其方法是只采用有限几个正弦波来构成复合信号。这些正弦波的频率要选择为数字信号频率（主频）的合适倍数，而且至少要使用3个正弦波才能近似地表示出数字信号。这其中的确切细节超出了本书的范围，但图6-9的几个图形可以说明这种近似表示的几种情况：图6-9a是一个数字信号，图6-9b是单个主频正弦波；图6-9c主频正弦波与一个3倍频正弦波的复合；图6-9d是在图6-9c复合波形基础上再加一个5倍频正弦波。

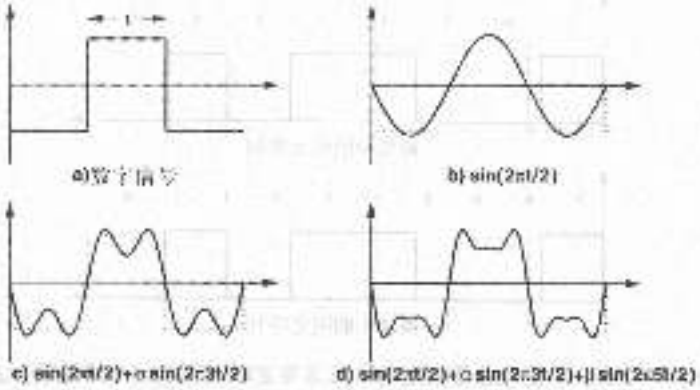


图6-9 用正弦波近似合成数字信号

6.13 数字信号的带宽

数字信号的带宽是什么？回顾一下，信号带宽就是构成信号的最高频率波与最低频率波之差。因此，计算带宽的一种方法就是应用傅里叶分析，先找出信号的正弦波成分，然后计算它们的频率范围。

数学上，对方波（如图6-9a所示的数字信号）进行傅里叶分析时，分析的结果会是正弦波的一个无限集，而且这个集合中的频率范围也是无限的。因此，在频域上绘图表示时，其频率范围沿X轴无限延伸，重要结论：

> 根据带宽的定义，数字信号具有无限大的带宽，因为对一个数字信号进行傅里叶分析会产生正弦波的一个无限集，集合中正弦波的频率也是无限多个。

6.14　信号的同步与协调

前面的例子都忽略了在设计可行的通信系统时要涉及到许多细节问题。例如，为了保证发送器和接收器的信号位元时间相一致，物理介质两端的电子设备必须具有能精确测量时间的电路。也就是说，如果一端以每秒10^8个位元发送信号的话，另一端就必须预期正好每秒接收10^8个位元。当低速传输时，使两端协调是很容易的，但要构建出在现代网络中使用高速电子设备却是极为困难的。

还有一个更基本的问题，要追溯于数据的信号表示方法是关于发送器和接收器之间的同步（synchronization）问题。例如，假定接收器没有接收到到达的第1个数据位，而从第2个位开始解释数据。或者考虑一下，如果接收器预期的数据到达速率高于发送器的发送速率，那会发生什么现象呢？图6-10说明了速率不匹配时会如何产生错误的情况。在图6-10中，虽然发送器和接收器是在相同的时间范围内开始和结束，但由于接收器都在略超前于每个位元起点上进行定位，因而造成接收器对信号作出错误解释，好像实际接收的位元数要多于已发送的数量。

在实际中，同步的偏差可能是极其微小的。例如，假定接收器的硬件对数据位1的定时误差是10^{-5}，这个误差只有在连续发送1千万位元后才可能表现出来，不过，因为高速通信系统以每秒传输千兆位的速率发送数据，所以即便如此小的误差，也会很快呈现同步差错而使问题变得严重起来。

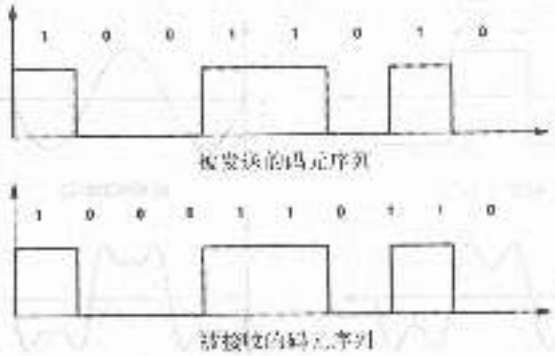


图6-10　接收器定时总是略超前于发送器定时而引起同步错误的示意图

6.15　线路编码

现在已经发明了多种技术能帮助避免同步差错。一般来说，有两种广泛采用的方法：一种方法是，在发送数据前，发送器先发送一段约定模式的位串（通常是一组0、1交替的位串），从而促使接收器实现同步；另一种方法是，采用不会混淆数据含义的合适信号来表示数据。我们使用术语线路编码（line coding）[⊖]来描述这种采用信号对数据进行编码的方法。

现在考虑如何采用支持3个离散信号电平的传输机制，以此作为消除模糊含义的线路编码例子。为了保证同步，保留其中一个信号电平以作为每个码位的开始。例如，如果3个可能的电平分别对应于−5、0、+5V电压，则保留−5V作为码位的开始电平。逻辑0可由（−5, 0）电平顺序来表示，而逻辑1可由（−5, +5）电平顺序来表示。如果我们定义其他电平组合顺序都是无效的，−5V电平的出现总是表示一个码位的开始，那么接收器可以利用−5V电平的出现

⊖　又称为“信号编码”。——译者注

来实现与发送器的准确同步，图6-11说明了上述表示方法。

当然，使用多个信号元素来表示单个码位意味着单位时间所传输的比特数减少。因此，设计者更愿意使用每个信号元素可以传输多个码位的方案，如图6-8b所示的方案。

图6-12列出了通常使用的线路编码技术名称及相关的分类。更详尽的细节超出了本书的讨论范围，我们只需知道根据通信系统的不同需要来选择不同的方案就够了。

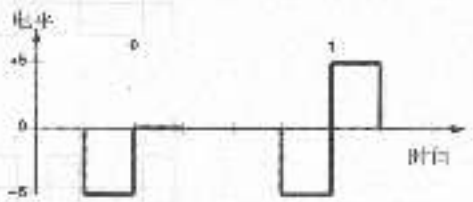


图6-11 用来表示数据位的两个信号元素示例

分 类	方 案	自同步特性
单极性	NRZ	全0或全1的长序列没有自同步
	NRZ-L	全0或全1的长序列没有自同步
	NRZ-I	全0或全1的长序列没有自同步
	Diphase	有自同步
双极性	AMI	全0的长序列没有自同步
多电平	2B1Q	相同电位长序列没有自同步
	8B6T	有自同步
	4D-PAM5	有自同步
多线路	MLT-3	全0的长序列没有自同步

图6-12 通常使用的线路编码技术名称

要点 有各种不同的线路编码技术可供采用，它们的区别在于如何处理同步以及其他特性（如所用带宽）方面有所不同。

6.16 曼彻斯特编码

除了图6-12所列举的线路编码技术外，还有一个特殊的线路编码标准对于计算机网络特别重要——用于以太网的曼彻斯特编码（Manchester Encoding）。

为了理解曼彻斯特编码，重要的是明白：检测信号的跳变比测量信号电平值更容易。为何曼彻斯特编码要使用电平的跳变而不是电平值的大小来定义码位，也只有用硬件的这种工作特点才能解释清楚。也就是说，曼彻斯特编码不是用某个电平值（例如+5V）来对应于逻辑1，而是定义电压由0V到一个正电压的跳变来对应于逻辑1，相应地，电压从某个正电压值到0V的跳变则对应于逻辑0，而且跳变发生在码位时限的“中间”，所以在数据中出现两个连续1或两个连续0的情况下，可以使得码位结束时信号总能回到原来的电平上。图6-13a说明了这种情况。

另有一种由此衍生的编码技术称为差分曼彻斯特编码（Conditional Dephase Encoding），也叫作条件反相编码，它使用相对跳变而不是绝对跳变的规则，即对一个码位的表示取决于前一个码位，每个码位时限包含1次或2次跳变，而且在码位的中间总要发生1次跳变。码位的逻辑值取决于该码位开始时刻是否发生跳变——发生跳变则代表逻辑0（这时该码位中有2次跳变），而没有发生跳变则代表逻辑1（这时该码位中只有1次跳变）。图6-13b说明了差分曼彻斯特编码原理。也许差分编码的最重要特性来源于实际的考虑——即使将收送信号的两根导线偶尔弄反了，这种编码规则还应该是正确的。

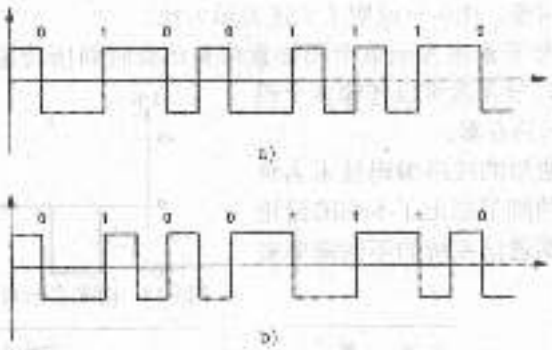


图6-13　a)曼彻斯特编码，b)差分曼彻斯特编码两种编码都假设了原点的前一码位以低电平结束

6.17　模拟—数字信号转换

有很多信源输出的是模拟信号，这意味着需对它们进行进一步处理（如进行加密）之前必须先转换成数字形式的信号。有两种基本的转换方法：

- 脉码调制。
- delta调制。

脉码调制（pulse code modulation，PCM[⊖]）是指对模拟信号的电平按固定时间间隔进行反复测量并转换为数字形式的一种技术。图6-14说明了它的转换步骤。



图6-14　脉码调制的3个步骤

每个测量值被称为抽样值，所以把第一个步骤叫抽样（sampling）。将抽样值记录下来后再将它转换成为一个小整数值，然后编码成为某种特定的格式，这一步叫做量化（quantized）。量化值并不是对信号的电压值或任何属性的测量，而是对信号最低电平与最高电平之间的范围被分割出来的一组间隔数，通常是2的指数。图6-15说明了把一个信号以8个间隔量化的情况。

在图6-15中，6个抽样点由垂直灰线表示。每一抽样点选择最接近的量化间隔数作为量化值。例如第3个抽样点，这个靠近波峰的点被赋给的量化值为6。

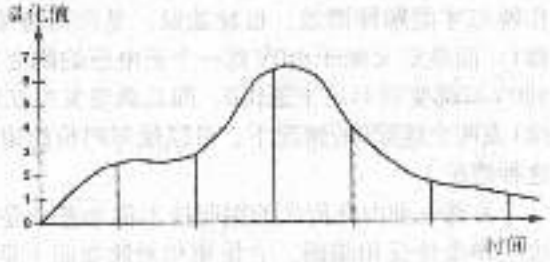


图6-15　脉码调制的抽样和量化示意图

在实际中，所采用的抽样方法都有少许的改进。例如，为了避免信号毛刺所产生的误差，可以使用取平均值的方法，就是说，每个抽样值不是单个抽样点的测量值，而是取相邻3个点的测量值经算术平均而计算出的抽样值。

脉码调制的主要替换技术是delta调制（又叫Δ调制）。delta调制也需要抽样，但它不是发送每个样点的量化值，而是发送前一个样点与当前样点之间差值的量化值，这种方法源自这个观点：只传输差异值比传输全部样点值所要求的比特数要少，特别是当信号不是快速变化的时候。delta调制值得权衡之处在于由差错

⊖　缩写词PCM具有二义性，因为它可以泛指一般的脉码调制技术，也可以特指电话系统中采用的脉码调制形式，下一小节将会讲到。

引起的后果问题——如果码位序列中的任何一位被丢失或毁坏了，所有后续比都会被错误解释。因此，在预期传输数据容易丢失或改变的通信系统中，通常都采用脉码调制。

6.18 奈奎斯特定理与抽样率

无论采用脉码调制还是采用delta调制，我们都必须对模拟信号进行抽样。那么，对模拟信号应该以多高的频度进行抽样呢？抽样太少，又称为“抽样不够”（undersampling），意味着数字值只是给出原始信号的粗略近似；抽样太多，又称为“抽样过头”（oversampling），意味着会产生太多数字数据，会消耗额外的带宽。

数学家奈奎斯特回答了应该进行多少次抽样的问题：

$$\text{抽样率} = 2 \times f_{\max} \tag{6.2}$$

其中，$f_{\max}$是复合信号中的最高频率。这就是奈奎斯特定理（Nyquist Theorem），它提供了对抽样问题的实际解决方案——对信号进行抽样的最低频率，至少必须是信号中所含的最高频率的两倍。

6.19 奈奎斯特定理与电话系统传输

作为奈奎斯特定理的一个特例，我们来考虑一下用来传送话音的电话系统。对人类的话音进行测试表明，信号中包含0～4000Hz频率成分即可提供可接受的声音质量。因此，根据奈奎斯特定理可以确定：把话音信号从模拟转换为数字信号的抽样率应该是每秒钟8000次。

为了进一步提供合理的声音重现质量，电话系统中的PCM标准使用8位值对话音信号进行量化，也就是说，输入信号的电平范围被划分为256个可能的等级，每个抽样点的可取值范围为0～255。因此，单个电话呼叫所产生的数字数据的速率就是：

$$\text{数字化话音速率} = 8000\frac{\text{样点数}}{\text{s}} \times 8\frac{\text{bit}}{\text{样点}} = 64\,000\text{bit/s} \tag{6.3}$$

在后面的章节中会看到，电话系统使用64 000bit/s（64Kbit/s）的速率作为数字通信的基础。我们还将进一步知道，因特网正是使用数字电话线路进行远距离连接的。

6.20 编码与数据压缩

我们用术语数据压缩（data compression）来指那种减少表示数据所要求的比特数的技术。数据压缩与通信系统的关系特别密切，因为减少了表示数据所需的比特数就可以减少传输所需的时间。

第29章将讨论数据压缩在多媒体中的应用。这里我们只需要理解两类压缩的基本定义即可：

- 有损压缩——在压缩的过程中会丢失一些信息。
- 无损压缩——在压缩后的数据中保留了所有信息。

有损压缩通常应用于人类使用的数据，如图像、视频或音频文件的片断等。这类压缩的主要原则是压缩数据所保留的细节只需达到人类感官可以接受的程度，也就是说，人类无法觉察到的变化都是可以接受的。我们将看到著名的压缩方案，如JPEG（应用于图像）或MPEG-3（简写成MP3，用于声音录制）都采用有损压缩。

无损压缩可以完好无缺地保存原始数据，因此它可以用于文档或任何必须准确保存数据的场合。当应用于通信时，发送器在传输前压缩数据，接收者接收后解压数据。因为压缩是

无损失的，所以发送器可以压缩任意数据，接收端解压缩恢复后可以得到与原始数据一样的拷贝。

大多数无损压缩使用字典法，压缩时找到数据中重复的字符串，形成一个字符串字典，每次相应的字符串出现时，则用字典中的索引替代，从而达到压缩数据的目的。发送器必须把字典和压缩数据一起传输。如果数据中包含重复很多次的字符串，压缩数据与字典的组合规模就会比原始数据的规模小。

6.21　本章小结

信源可能给出模拟数据或数字数据。模拟信号有周期性和非周期性的。周期信号具有振幅、频率和相位等属性。傅里叶发现任何曲线可以由一组正弦波相加而形成，单个正弦波形归类为简单信号，可以分解为多个正弦波的信号归类为复合信号。

工程上使用两种主要的复合信号表示方法。时域表示法表示信号随时间的变化，频域表示法表示复合信号中每个简单信号的振幅和频率。带宽是信号的最高频率与最低频率之差，在信号的频域图中能很清楚地看出来。

信号的波特率是信号每秒钟可以变化的次数。使用多电平的数字信号来表示数据时，信号的每次变化可以表示多个码位，这样就使得有效传输速率提高到波特率的几倍。虽然数字信号具有无限带宽，但仍然可以用3个正弦波对它作近似表示。

实际中存在不同的线路编码技术，其中在以太网中使用的曼彻斯特编码特别重要。曼彻斯特编码不是采用绝对的信号电平值来表示数据码位，而是采用信号电平的跳变。差分曼彻斯特编码使用相对跳变。使用差分曼彻斯特编码时即使两根导线弄反了，系统也一样能正常工作。

脉码调制和delta调制用来把模拟信号转换为数字信号。PCM方案应用于电话系统时，采用8位量化并进行每秒8 000次抽样，因此速率是64Kbit/s。

压缩分为有损的和无损的两类，有损压缩更适用于图像、音频及视频等被人类观赏的信息，丢损失而造成的信息变化可以被控制在低于人类感觉极限的范围内，无损压缩更适用于文件或者必须准确保存的数据。

练习题

6.1　举出除计算机以外的3个信息源。

6.2　举出一个产生非周期信号的家用设备。

6.3　为什么说正弦波是数据通信的基础？

6.4　说出并描述正弦波的4个基本特征。

6.5　当给出一个正弦波形图时，用什么方法能最快判断出它的相位是否为0？

6.6　何时一个波形可被归类为简单波形？

6.7　对一个复合波进行傅里叶分析会产生什么？

6.8　在频域图中，y轴代表什么？

6.9　信号的模拟带宽是什么？

6.10　对于时域图和频域图，利用哪一个表示图更容易计算带宽？为什么？

6.11　假设工程师把信号电平由2个增加为4个，在相同的时间内一次能多发送几个数据位？请解释。

6.12 波特的定义是什么?
6.13 为什么使用模拟信号来近似表示数字信号?
6.14 数字信号的带宽是什么?请解释。
6.15 什么是同步差错?
6.16 为什么在有些编码技术中使用多个信号元素来表示一个码位?
6.17 在曼彻斯特编码中使用信号的什么特征来表示码位?
6.18 差分曼彻斯特编码的主要优点是什么?
6.19 把模拟信号转换到数字信号的过程中,抽样之后接着是什么步骤?
6.20 如果人耳可听到的最高频率是20 000Hz,那么对麦克风产生的模拟信号抽样以进行数字化转换时,其抽样率必须是多少?
6.21 对于电话系统中所采用的PCM编码,抽样点之间的时间间隔是多少?
6.22 简述有损压缩和无损压缩之间的区别,并说明何时该应用哪种压缩技术?

第7章 传输介质

7.1 引言

第5章给出了数据通信的概述，第6章讲述了信息源、模拟和数字信息以及编码技术等方面的内容。

本章从传输介质（包括有线、无线和光学介质）方面继续讨论数据通信的内容。本章对介质类型进行分类，介绍电磁传播的基本概念，并解释如何采用屏蔽的方法来阻止或减少干扰和噪声。最后，阐释信道容量的概念。后面章节将继续讨论数据通信其他方面的内容。

7.2 导向传输与非导向传输

该如何对传输介质进行分类？有两种相分类方法：

- 按路径类型：通信可以沿着确切的路径（如导线）传输信号，也可以没有明确的路径（如无线电传输）。
- 按能量形式：电气能量应用于有线传输，无线电波应用于无线传输，光应用于光纤。

我们使用术语导向（guided）传输和非导向（unguided）传输，以便区分物理介质传输和无线电波传输。物理介质（如铜导线或者光纤）提供明确的传输路径，而无线电波则通过自由空间在所有方向上传播。非正式地，工程人员也经常使用有线（wired）和无线（wireless）这两个术语。应该注意到这两个非正式术语的使用有点混乱，因为即使当物理介质是光纤的时候，也好像听到有人称它是有线的（wired）。

7.3 按能量形式分类

图7-1说明如何根据用于传输数据的能量形式，对物理介质进行分类。后面几节将分别对每种介质进行描述。

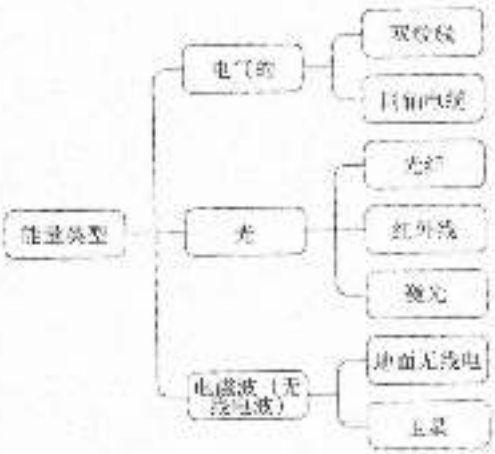


图7-1 按使用的能量对介质进行分类

与大多数分类方法一样，这种分类方法也不是完美的，也存在例外情况。例如，环绕地球轨道的空间站就可能采用不涉及卫星的非地面通信。尽管如此，我们的分类方法都覆盖了大多数通信方式。

7.4 背景辐射和电气噪声

回顾基础物理学中所述，电流要通过一个完整的电路才能流动。所以，所有电能的传输都要求有两根导线以便形成一个电路——其中一根连到接收端，另一根则返到发送端。最简单的连线传输形式是使用一根含有两根铜导线的电缆即可，电缆中的每根导线都被电绝缘材料包裹，以使导线之间互相绝缘，并用外围的被覆材料将这两根导线包裹在一起，这样就使电缆使用起来更方便。

计算机网络使用另一种形式的导线，要理解为什么这样，必须知道如下3个事实。

(1) 随机电磁辐射，又称为噪声（noise），弥漫于环境之中。事实上，通信系统本身就会产生少量的电气噪声，这是系统正常操作下的一种副作用。

(2) 当电磁辐射碰到金属的时候，会感应出小的信号，这意味着随机噪声会干扰用于通信的信号。

(3) 因为金属可以吸收辐射，所以它可以起到屏蔽（shield）的作用，因此，如果在噪声源与通信介质之间放置足够（厚的）金属，即可防止噪声干扰通信。

前两个事实概括了使用电气或电磁能量的通信介质所固有的根本问题。这个问题在产生随机辐射源的附近表现得尤其严重。例如，白炽灯泡和电机都会产生辐射，特别是具有较大功率的电机（如用于操作电梯、空调以及泵等的电动机）则问题更为严重。令人吃惊的是，诸如碎纸机或者电动工具这样的小设备，也会产生足够的电磁辐射来干扰通信。这里的要点是：

> 诸如电动机这样的设备所产生的电磁辐射，会干扰那些使用无线电波传输或通过导线传送电气能量的通信系统。

7.5 双绞线

第7.4节的第3个事实解释了通信中使用导体的抗干扰作用，有3种形式的导线能帮助减小电气噪声的干扰。

- 无屏蔽双绞线（Unshielded Twisted Pair，UTP）。
- 同轴电缆（Coaxial Cable）。
- 屏蔽双绞线（Shielded Twisted Pair，STP）。

第一种形式通常称为双绞线或者无屏蔽双绞线，在通信中广泛使用。双绞线由两根绞在一起的导线构成。当然，每一根导线都有一层起绝缘作用的塑料包裹，以防止导线之间电流短路。

令人吃惊的是，绞在一起的两根导线其抗噪声干扰的能力明显强于两根平行放置的导线。图7-2说明了产生这种现象的原因。

如图7-2所示，当两根导线平行时，很有可能其中一根导线距离电磁辐射源更近，那这根导线会试图多吸收一些电磁辐射，从而起到对另一根导线的屏蔽作用。因此，处在它后面的另一根导线吸收到的电磁能量会比较少。在图7-2中，总共32个单元的辐射碰到这两根导线。在图7-2a中，上面的导线吸收了20个单元，下面的导线吸收了12个单元，因此上下两根导线吸收的辐射单元差值是8。在图7-2b中，每根导线处在上下位置的部分各为一半，这意味着每

根导线所吸收的电磁辐射数目是相同的（即差值为0）。

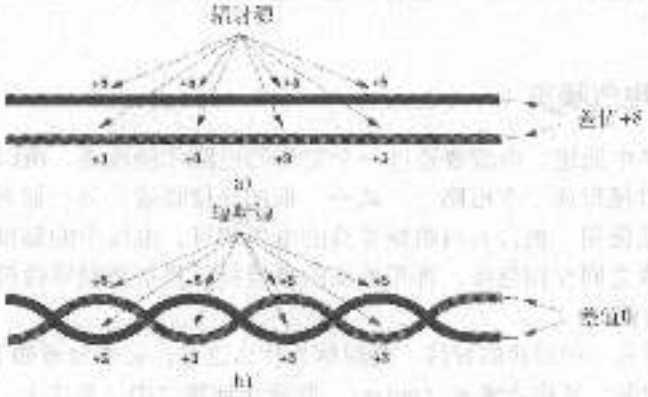


图7-2　不希望的电磁辐射干扰：a）两根平行导线；b）双绞线

为什么吸收等量辐射这么重要呢？答案是如果每根导线受到干扰所感应的电能正好相等，就不会有额外的电流流通，因而原始信号不会受到干扰。这里的要点是：

为了减少随机电磁辐射所产生的干扰，通信系统使用双绞导线而不是平行导线。

7.6　屏蔽：同轴电缆和屏蔽双绞线

虽然双绞线可以抵御大多数背景辐射的影响，但是它并不能解决所有的问题。当碰到下列情况时，双绞线仍然存在问题：

- 特别强的电子噪声。
- 与噪声源之间的物理距离非常近。
- 通信使用高频率。

如果辐射强度很高（如在使用电弧焊设备的工厂），或者通信电缆需要在电气噪声源附近经过，这样的情况下使用双绞线也可能不足以屏蔽噪声。因此，如果双绞线要经过办公楼安装有白炽灯的天花板上方时，就会形成干扰。况且，要设计出能区分有效的高频信号与噪声的设备是很困难的，这意味着：当采用高频率通信时，即使少量的噪声也会引起干扰。

为了应对双绞线抗干扰性不充分的情况，可以采用具有额外金属屏蔽层的导线。我们最熟悉的一种，就是用于有线电视的称为同轴电缆（coaxial cable）的导线。这种导线的外面是厚实的编织导线所形成的屏蔽层，其所包裹的芯导线负责载送信号。同轴电缆的结构如图7-3所示。

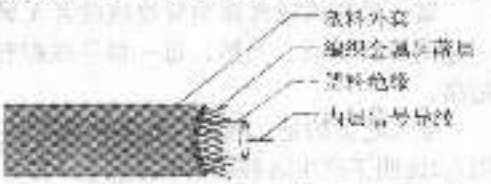


图7-3　屏蔽层包围信号导线的同轴电缆示意图

同轴电缆中的屏蔽层形成一个柔韧性好的金属圆柱体围绕着内层导线，从而为内层导线提供了可以阻挡任何方向电磁辐射的屏障。这种屏障同时也能防止内层导线在传输信号时辐射电磁能量干扰其他导线。因此，同轴电缆可以放置在电气噪声源和其他电缆的附近，而且可应用于高频率通信。这里的要点是：

导线的屏蔽层及整体的对称性使得同轴电缆可以免受噪声的干扰，能够承载高频信号，并能防止内部的信号辐射噪声去干扰周围的电缆线。

使用编织导线来代替坚实的金属作为屏蔽层可以保持同轴电缆的柔韧性，而且方便灵活。但是，这种厚实的屏蔽层还是使同轴电缆的柔韧性不及双绞线。有几种不同的屏蔽形式提供了折中方案：电缆越柔韧，其抗电磁噪声干扰能力就越差。有一种流行的叫做屏蔽双绞线（STP）的电缆线，它使用更细、更柔韧的金属屏蔽层包裹一对或多对双绞线。大多数STP类型的电缆其屏蔽层由金属薄片构成，类似于厨房里（用于烘烤肉品）的铝薄片。STP的优点是：其柔韧性比同轴电缆更好，抗电磁噪声干扰的能力要比UTP更强。

7.7 双绞线分类

电话公司率先制定了在电话网络中使用的双绞线标准。最近，3个标准化组织联合制定了在计算机网络中使用的双绞线标准。美国国家标准化组织（ANSI）、电信工业协会（TIA）和电子工业联盟（EIA）创建了一个双绞线的导线分类列表，并分别为每一类别制定了严格的规范。图7-4归纳了主要的分类。

分　类	描　述	数据速率/Mbit/s
1	用于电话的无屏蔽双绞线	<0.1
2	用于T1数据的无屏蔽双绞线	2
3	用于计算机网络的改进型CAT2	10
4	用于令牌环网的改进型CAT3	20
5	用于网络的无屏蔽双绞线	100
5E	具有更高抗噪声能力的扩展型CAT5	125
6	用于200Mbit/s速率测试的无屏蔽双绞线	200
7	屏蔽双绞线（每对屏蔽双绞线使用金属薄片屏蔽整个电缆线）	600

图7-4 双绞线分类

7.8 使用光能的介质及光纤

根据图7-1的分类法，有3种介质使用光能来载送信息：

- 光纤。
- 红外传输。
- 点对点激光。

使用光的最重要介质类型是光纤（optical fiber）。每根光纤由封装在塑料封套中的一束细玻璃或透明塑料丝构成。一根典型的光纤用于在单个方向上的通信——一端使用激光器或发光二极管LED作为发射器，将光脉冲发送到光纤里面；另一端则使用光敏元件来检测光脉冲。若要实现双向通信，需要使用两根光纤，每根分别承载一个方向的信息。因此，光纤通常被集中封装在塑料封套内形成光缆，每个光缆至少包含两根光纤。如果要连接两个拥有许多网络设备的大站点，则所用光缆通常要包含多根光纤。

虽然光纤不能打直角弯曲，但是它的柔韧性也足以形成一个直径小于2ft[①]的圆圈而不会折断。问题是，光是如何沿着弯曲的光纤传播的呢？答案来自光物理学原理：当光到达两种物质的边界时，它的传播途径由两种物质的密度和入射光线与边界之间的角度共同决定。对于给定的两种物质，存在一个临界角（critical angle），记为θ，即入射线与入射点法线（在入射点上与边界垂直的线）之间所形成的一个锐角。当入射角小于0度时，光线在边界处会发生

① 1ft=0.3048m。——编者注

折射；当入射角大于0度时，光线在边界处则发生镜面反射。上述概念如图7-5所示。

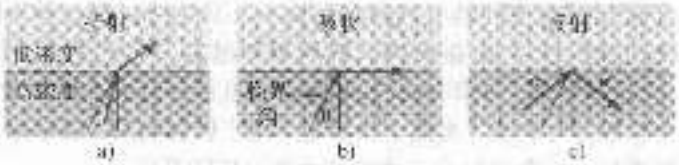


图7-5　不同入射角对光线在密度边界上发生的不同现象

a)入射角小于临界角，b)入射角等于临界角，c)入射角大于临界角

图7-5c解释了为什么光线能保留在光纤里面的原因——光纤被一层覆层物质包裹从而形成一个边界，当光线射进光纤内时就在边界上不断反射。如此始终使光线保留在光纤内传播。

遗憾的是，光纤内的反射并不是完美的，它会吸收一小部分能量。另外，当光子沿着锯齿形路径通过多次反射传播时，光子传播的路径会比直线传播的稍长，从而导致从光纤一端发送的光脉冲到达另一端时，能量变少，并且出现色散（dispersed），如图7-6所示。

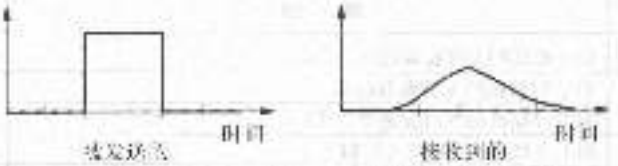


图7-6　光脉冲通过一根光纤被发送和接收的情况

7.9　光纤类型及光传输

使用光纤来连接计算机与临近设备是毫无问题的。但是如果要使用很长的光纤（如连接两个城市或者要通过大西洋的海底）的话，色散就成为一个非常严重的问题。为此，工程人员发明了以下3种光纤，供人们根据需要在性能和代价之间权衡选择。

- 多模突变（Multimode，Step Index）光纤是最便宜的，用于对光纤性能要求不高的场合。纤芯的密度从中心到边缘不变，但与覆层之间的边界密度是突变的，这使得光线频繁反射，因此造成色散程度高。
- 多模渐变（Multimode，Graded Index）光纤比多模突变光纤clear贵。在多模渐变光纤中，纤芯中心处的密度最高，从中心到边缘逐渐降低，这样可以减少反射，降低色散程度。
- 单模（Single Mode）光纤是最贵的，色散最少。这种光纤具有较小的直径以及其他有助于减少反射的特性。单模光纤用于长距离及较高比特率传输的场合。

单模光纤及其端设备的设计都是为了聚焦光束，因此光脉冲可以传输几千公里而不发生色散。最小化色散有助于提高比特传输速率，因为对应于一个比特的光脉冲不会因为色散而与下一个比特的光脉冲相串扰。

在光纤中如何发送和接收光束呢？关键是用于传输的设备必须与光纤相匹配。可用的机制包括：

- 传输：发光二极管（LED）或注入激光二极管（ILD）。
- 接收：光敏元件或光敏二极管。

通常，LED和光敏元件用于较短距离较低速率传输，一般与多模光纤匹配使用。用于长距离和高速率传输的单模光纤则通常要求使用ILD和光敏二极管。

7.10 光纤与铜导线的比较

光纤的几个特性使其在数据传输中比铜导线更受欢迎。光纤可免受电气噪声干扰，具有更高的带宽，并且光信号在光纤中的传输衰减也小于电信号在铜导线中的传输衰减。不过，铜导线比较便宜，而且由于光纤在使用之前必须将端面打磨得平整光滑，所以安装时需要用到专门设备，而铜导线的安装更不需要此类特殊设备和专业人员。最后，铜导线比光纤坚固，不容易因为意外牵拉或弯折而断裂。图7-7总结了这两种介质的优点。



图7-7 光纤和铜导线的优点

7.11 红外通信技术

红外（InfraRed，IR）通信技术与平常使用的电视机遥控器一样，都使用同样的能量类型，即与可见光行为特性相似但却在人眼可见范围以外的一种电磁辐射形式。与可见光一样，红外线扩散非常快，红外信号可以在光滑且坚硬的表面反射，然而即使是薄如纸片的不透明物体也能阻挡红外信号的传播，即使空气中的水蒸气也一样能阻挡红外信号。

这里的要点是：

> 红外通信技术非常适合于室内环境中，其在发送器和接收器之间直接视距没有任何物体的场合下使用。

红外技术最常用于连接计算机和邻近的外部设备（如打印机）。计算机和打印机上各有一个接口分别发射出覆盖弧度为30°的红外线信号。假若两个设备开排放着，彼此都能接收到对方的信号。红外线的无线传输方式对于便携式电脑特别有吸引力，因为用户可以在房间内随意移动，而且还能接入到打印机上。图7-8所示为3种常用的红外技术及其分别所支持的数据速率。

简　称	全　称	速率/Mbit/s
IrDA-SIR	低速红外线	0.115
IrDA-MIR	中速红外线	1.150
IrDA-FIR	高速红外线	4.000

图7-8 3种常用红外线技术及其数据速率

7.12 点对点激光通信

因为沿视线传播的光束只能连接一对设备，所以上述红外技术可以归类为点对点（point-to-point）通信技术。除了红外，还有其他点对点通信技术，其中一种点对点通信的形式就是使用由激光器产生的相干光（coherent light）。

与红外光类似，激光通信沿视线传播，并且要求通信站点之间有一条清晰且畅通无阻的路径。与红外发送器不同的是，激光束覆盖的区域不能太宽阔，只有几厘米宽。这样，发送器和接收器必须精确校准，以确保发送器的光束能准确到达接收设备的感应器。在典型的通信系统中，必须具有双向通信能力，因此每一边都必须有一个发送器和一个接收器，并且两边的发送器都必须经过仔细校准。因为校准非常关键，所以点对点激光设备通常都是固定安装的。

激光束具有适合户外使用的优点，并且与红外相比可以传播更长的距离，因此激光技术特别适用于城市楼宇之间的信息传输。例如，假设有一个大公司同时在两座相邻的大楼里拥有办公室，但是一般不允许公司在建筑物之间跨街道布线。不过，公司可以购买激光通信设备并且固定安装在两座大楼上，根据需要既可以分别安装在两座大楼的旁边，也可以安装在屋顶。一旦设备采购安装后，日常的运营费用是相对较低的。

概括如下：

> 激光技术可以用于研发点对点通信系统，因为激光器发送的光束很窄，发送器和接收器必须精确校准，因此典型的安装方法是把设备附着在固定设施上（如大楼的屋顶）。

7.13　电磁波（无线电）通信

回顾前面提到的术语非定向（unguided），它是用于描述不需要通信介质（如导线或光纤）传播能量的通信技术。最常用的非定向通信机制就是在射频（Radio Frequency，RF）范围内使用电磁辐射能量的无线通信技术，RF传输相比光线具有明显的优势，因为RF能量可以长距离传播，并能穿透诸如建筑物墙壁这样的物体。

电磁辐射能量的确切属性取决于频率。我们使用术语频谱（spectrum）来指频率的可能范围。世界各国政府都会根据特定目的分配使用频率。在美国，是由联邦通信委员会（Federal Communications Commission）制定频率的分配规则，并对通信设备在每个频率上允许的发射功率设置限度。图7-9说明了电磁辐射整体频谱的范围和每一频段的大体特征。如图7-9中所示，频谱的一部分对应前面描述过的红外线；无线通信所使用的频谱范围大致是3KHz到300GHz，包括了分配给无线电和电视广播以及卫星和微波通信的使用频率。

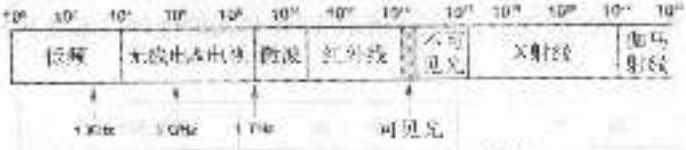


图7-9　电磁辐射频谱的主要频段范围

7.14　信号传播

第6章解释了电磁波所能表示的信息量取决于它的频率，而电磁波的频率决定了它的传播途径。图7-10描述了3种主要的传播方式。

分　类	频率范围(MoHz)	传播方式
低频段	<2	波沿着地球表面传播，会被不平坦的地形阻碍
中频段	2～30	波一般被大气层特别是电离层反射
高频段	>30	波沿直线传播并且会被障碍物吸收

图7-10　不同频率段电磁波的传播方式

依图7-10所述，低频率段的电磁辐射是沿着地球表面传播的，这意味着如果地表相对平坦，则可以将接收器放置在距离传输器超过视野范围的地方。对于中频段，发送器和接收器可以分开放置得更远，因为信号可以在电离层和地面之间不断反弹从而在两者之间传播。最

后，高频段的无线电与光线的行为特性相似——信号在发送器和接收器之间沿直线传播，且两者之间的路径必须畅通无阻。这里的要点是：

无线网络技术的频率不能随意选择，因为政府管制了频谱的使用。每个频率都有自己的特征，例如波的传播方式、功率要求以及对噪声的敏感性。

无线技术可以归类为如下两大类：

- 陆地的（Terrestrial）。通信所使用设备（如无线电接收装置或微波传输器）与地表相对较近，天线或其他设备通常位于山顶、人造塔或高楼。
- 非陆地的（Nonterrestrial）。通信所使用的一些设备位于地球大气层外（如在环绕地球轨道的卫星上）。

第16章将介绍特定的无线技术，并描述每种技术的特征。现在，理解如下方面就足够了，即使用的频率和功率会影响可传输数据的速率、可实现的最大通信距离以及其他特征（如信号是否能穿透固态物体等）。

7.15 卫星类型

物理学定律（具体是指开普勒定律）决定了沿地球轨道环绕物体（如卫星）的运动规律。具体来说，周期（围绕轨道绕行一周所需的时间）取决于与地球的距离。因此，根据与地球的距离不同，卫星通信可以大致划分为3种类型。图7-11列出了分类和对每种类型的描述。

轨道类型	描　述
低地球轨道（LEO）	优点：低时延 缺点：相对于地球上的观测者，卫星在天空中不断移动
中地球轨道（MEO）	特征为椭圆形（而非圆形），主要用于提供有北向限的通信
地球静止轨道（GEO）	优点：相对于地球表面某一位置，卫星保持在固定方位 缺点：时延长

图7-11 通信卫星的3种基本类型

7.16 GEO通信卫星

正如图7-11所描述的，通信卫星的主要权衡考虑在于高度和周期。地球静止轨道（Geostationary Earth Orbit，GEO）卫星的主要优点来自于轨道周期与地球自转周期完全一致。如果是位于赤道上空，GEO卫星始终保持在地球表面上空的同一位置。固定的卫星位置意味着一日地面站对准卫星，设备永远都不需要再移动。图7-12说明了这个概念。

遗憾的是，地球静止轨道的距离要求是35 785km（或22 236mile），大约相当于地球与月球之间的距离的十分之一。为了理解这么长的距离对通信的影响，我们来计算一下无线电波往返于GEO和地球所需的时间。以光速3×10^8m/s计算，往返所需时间是：

$$\frac{2\times35.8\times10^6\text{m}}{3\times10^8\text{m/s}}=0.238\text{s} \tag{7.1}$$

虽然这个数值看似并不重要，但对于一些应用来说，近似0.2s的延迟却可能会显得很重要。在电话交谈或者视频电话会议中，人能感觉到0.2s的延迟。对于电子商务（如股票交易）会出一个受限制的证券单价，延迟0.2s出价有可能意味着交易成功与否的分水岭。总结如下：

即使以光速传播，当信号从地面站传到GEO卫星并返回另一个地面站，所需时间

超过0.2s。

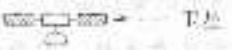




图7-12　GEO卫星和永久固定的地面站

7.17　GEO对地球的覆盖

可能有多少颗GEO通信卫星呢？有趣的是，由于使用某一给定频率的通信卫星之间必须隔开一定的距离以避免干扰，所以分布在赤道上空的地球同步轨道上只能存在有限的"间隔"可用。最小的间隔取决于发射器功率的大小，但通常需要4～8°的角度间隔，这样，赤道上空的360°圆形轨道上就只能容纳45～90颗卫星了。

覆盖整个地球最少需要几颗卫星呢？3颗。为了明白其中道理，考虑图7-13。这个图表显示了3颗GEO卫星间隔120°被定位在赤道上空，以及来自3颗卫星的信号覆盖地球一圈的情况。在图7-13中，地球的大小及地球与卫星的距离是按相应比例画出来的。

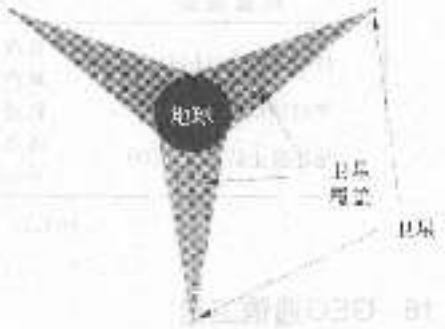


图7-13　3颗GEO卫星间信号足以覆盖地球

7.18　LEO卫星与群集

对于卫星通信，亦可选择使用低地球轨道（Low Earth Orbit，LEO），这个轨道的定义高度是在地球上空2 000km以下。实际问题是，卫星的位置必须高于大气层以避免运动时与大气的摩擦，因此LEO卫星通常处于与地球相距500km或者更高的位置。LEO的优点是低时延（通常是1～4ms）。缺点是卫星的环绕速度与地球自转速度不匹配。因此，地球上的观察者通过望远镜可以看到这种卫星在天空中移动，这意味着地面站必须具有能旋转和跟踪卫星的天线。然而跟踪是非常困难的，因为卫星运动很快，高度最低的LEO卫星沿整个轨道环绕一圈大约需要90分钟，更高的卫星则需要几个小时。

为了利用低地球轨道卫星进行连续通信，通常采用的一种技术是众所周知的集群（clustering）或阵列（array）配置。一大群卫星被设计成一个集群一起工作。除了与地面站通信，群集中的LEO卫星还可以与群集中的其他卫星保持相互通信，而且在需要的时候还可以

转发消息。例如，考虑当一个欧洲的用户想发送一条消息给一个北美的用户时，会发生什么呢？首先欧洲的地面站传送消息给正在欧洲上空的卫星，群集中的卫星相互通信以把消息转发给此时正在北美上空的卫星。最后，此时正在北美上空的卫星就把消息传送给北美的地面站。概括如下：

> LEO卫星群集一起工作以实现消息转发。群集中的成员必须知道哪颗卫星在当时位于地球某个特定区域的上空，从而把消息转发给合适的成员以便发送到对应的地面站。

7.19 介质类型之间的权衡

介质的选择并不简单，包括对多种因素的评估。选择介质时必须考虑的项目包括：

- 费用：材料、安装、运营以及维护。
- 数据速率：每秒可以发送的比特数。
- 时延：信号传播或处理所需要的时间。
- 对信号的影响：衰减和失真。
- 环境：对干扰和电气噪声的敏感性。
- 安全：对窃听的敏感性。

7.20 对传输介质的度量

前面我们已经提到过用来评估传输介质的两个最重要的性能指度：

- 传播时延：信号在介质中生成一次所需的时间。
- 信道容量：介质可以支持的最大数据速率。

我们在第6章提到过，20世纪20年代有一位研究者，发现了传输系统的带宽与通过该系统每秒能传输的最大速率之间的基本关系，这就是著名的奈奎斯特定理（Nyquist Theorem）。这个定理给出了在不考虑噪声影响情况下最大数据传输速率的理论上限。如果传输系统使用K个信号电平，模拟带宽是B，奈奎斯特定理所阐明的以每秒位元数计算的最大数据速率D是：

$$D=2B\log_2 K \tag{7.2}$$

7.21 噪声对通信的影响

奈奎斯特定理给出的是一个实际无法达到的绝对最大值。实际上，工程师们已经观察到实际通信系统总是受到一些称为噪声（noise）的背景干扰所限制，而这些噪声使得通信系统不可能达到这种理论极限传输速率。1948年，香农（Claude Shannon）扩展了奈奎斯特的结果，推导出在噪声影响下传输系统所能达到的最大数据传输速率。这个结果即香农定理（Shannon's Theorem）[⊖]，可以描述为：

$$C=B\log_2(1+S/N) \tag{7.3}$$

其中：C是用每秒位数表示的对信道容量的有效限制，B是硬件带宽，而S/N是信噪比（signal-to-noise ratio），即平均信号功率与平均噪声功率的比值。

作为香农定理的例子，假设某一具有1 KHz带宽的传输介质，它的平均信号功率是70单位，

⊖ 也称为香农-哈特雷定律（Shannon-Hartley law）。

平均噪声功率是10单位，因此信道容量是：

$$C=10^3\times\log_2(147)=10^3\times 3=3\,000\ \text{bit/s}$$

信噪比通常以分贝（decibels），缩写为dB表示，而其中的分贝被定义为对信号与噪声功率电平之差的度量。图7-14说明了功率电平的测量方法。

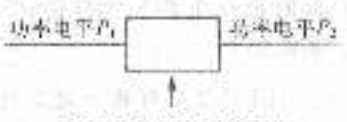


图7-14　系统两边所测量的功率电平

一旦两个功率电平测量出来，它们的差值由分贝表示。定义如下：

$$\text{dB}=10\log_{10}\left(\frac{P_2}{P_1}\right) \tag{7.4}$$

使用dB作为信噪比的测量看似平常，其实有两个优点。首先，dB值是负数时意味着信号被衰减（attenuated），即减弱；而dB值是正数则意味着信号被放大（amplified）。其次，如果一个通信系统由多个部分串行组合，整体系统的信噪比量值可以由各个部分测量到的dB值简单相加即可。

音频电话系统的信噪比大约为30dB，模拟带宽大约为3 000Hz。为了把信噪比dB转换为一个简单的分数，除以10并把结果作为10的幂（即30/10 = 3，而10^3=1 000，因此信噪比为1 000）。根据香农定理，通过电话网络所能达到的最大数据传输速率为：

$$C=3\,000\times\log_2(1+1\,000)$$

或者近似为30 000 bit/s。工程人员公认这是一个根本性的限制——只有提高信噪比才有可能达到更高的数据传输速率。

7.22　信道容量的重要性

对数据通信网络的设计工程师而言，上述的奈奎斯特定理和香农定理是很重要的。奈奎斯特的工作提供了一种动机和目标，它激励人们去探索对位串进行信号编码的复杂方法：

> 奈奎斯特定理鼓励工程人员去探索对位串进行信号编码的方法，因为巧妙的编码方法能实现在单位时间内传输更多的数据位。

从某种意义上来说，香农定理才更具根本性，因为它通过物理定律得到了一个数据传输速率的绝对极限。传输线路上的大多数噪声都可以归因于由于“大爆炸”而残留在宇宙中的背景辐射所造成的。

> 香农定理告诉工程人员：不存在任何巧妙的编码方法能够突破物理定律对实际通信系统中实现的最大每秒传输位数的根本限制。

7.23　本章小结

在现实中存在各种各样的传输介质，可以划分为导向的和非导向的两种形式，也可以根据使用的（电气的、光的或无线电传输的）能量形式进行分类。电气能量用在导线上。为了防止电气干扰，可以将铜导线成对绞合在一起或者封装在屏蔽物里面。

光能量可用在光纤上，或者使用红外线或激光实现点对点通信。光在光纤与包层之间的边界处反射。在入射角小于临界角的情况下，光可以保持在光纤中传播。当光通过光纤时，光脉冲会出现色散。在多模光纤中，色散最为严重，而在单模光纤中，色散最少。单模光纤

比多模光纤价格更贵。

无线通信使用电磁能量。在无线通信中，带宽和传播的行为特性都取决于所采用的频率。低频段信号在地球表面传播，较高频段信号则在地球与电离层之间反射，最高频段的行为特性与可见光相似。其发送器和接收器之间的路径要求是畅通无阻的直线。

卫星是主要的非地面通信技术。GEO卫星绕地球轨道而行的速度与地球自转相匹配，但是这种轨道与地球距离较远，会导致消息传播时延达到零点几秒。LEO卫星时延较低。在天空中迅速移动，LEO卫星使用群集或阵列技术来实现消息的中继转发。

奈奎斯特定理给出了在无噪声情况下传输介质的信道容量理论上限。香农定理给出了现实中存在噪声情况下的信道容量。香农定理中的信噪比通常用分贝测量。

练习题

7.1 导向与非导向传输的区别是什么？

7.2 当依据所用能量的类型来划分物理介质类别时，是指哪3种能量类型？

7.3 当噪声遇到金属物体时会发生什么现象？

7.4 请说出用于降低噪声干扰的3种导线类型。

7.5 请解释双绞缆线是如何降低噪声影响的。

7.6 请绘出同轴电缆横截面的示意图。

7.7 如果你正在为你的新房子安装计算机网络布线，你会选择哪一类双绞线？为什么？

7.8 请解释：当光纤弯曲成弧形时，光线为什么不会离开光纤？

7.9 什么叫色散？

7.10 请列举3种类型的光纤，并分别说明每一种光纤的一般特性。

7.11 在光纤中使用什么光源和感应器？

7.12 与铜导线相比，光纤的主要缺点是什么？

7.13 在红外技术中，用于发射光线的圆锥角大约是多少？

7.14 在运动车辆中可以使用激光通信吗？请解释。

7.15 为什么低频电磁辐射可用于通信？请解释。

7.16 无线通信的两个大类是什么？

7.17 请列举3种类型的通信卫星，并分别说明每一种类型的特征。

7.18 如果使用GEO卫星传送从欧洲发往美国的消息，从发送消息到接收到应答消息需要多长时间？

7.19 需要多少颗GEO卫星才能覆盖地球上所有的人类居住地区？

7.20 什么叫传播时延？

7.21 带宽、信号电平和数据速率之间的关系是什么？

7.22 如果使用了两个信号电平，在一条模拟带宽为6.2MHz的同轴电缆上能实现的数据传输速率是多少？

7.23 如果一个系统的平均功率电平是100，平均噪声电平是33.33，带宽是100MHz，那么信道容量的有效极限是多少？

7.24 如果系统的输入功率电平是9 000，输出功率电平是3 000，那么用dB表示的差值是多少？

7.25 如果某电话系统可以按40dB的信噪比运行，模拟带宽是3 000Hz，每秒可能传输多少位元？

第8章 可靠性与信道编码

8.1 引言

本书数据通信这一部分的每一章都介绍数据通信的一个方面，这是所有计算机联网技术的基础。第7章我们讨论了传输介质，并指出电磁噪声的问题。本章将继续讨论如何检测发现传输中出现的差错以及可用于差错控制的技术。

本章介绍的概念是计算机网络的基础，在协议栈的很多层上的通信协议中都要用到这些知识。特别是，差错控制的方法和技术贯穿于本书第4部分所讨论的各个因特网协议中。

8.2 传输差错的3个主要源头

所有数据通信系统对差错都是敏感的。有些问题来源于广义物理学的内在原因，而有些问题则是由仪器故障或设备没有达到工程标准要求而引起的。广泛测试可以排除由于不合格的工程而导致的许多问题，而仔细的监控可以确认设备的故障。不过，在传输中出现的往往是难以检测的小差错，并不是完全的损坏。因此许多计算机网络专注于控制差错发生以及从这种差错中恢复过来的方法。传输差错（transmission error）主要有3类：

- 干扰（Interference）。正如第7章所述，设备的电磁辐射（如电动机和背景宇宙辐射）所导致的噪声会扰乱无线电波的传输和导线的信号传输。
- 失真（Distortion）。所有物理系统都会导致信号失真。光脉冲在光纤中传播，会出现色散。金属导线具有电容和电感特性，这些特性会阻碍信号的某些频率成分而允许其他频率的信号通过。只是把金属导线靠近一个大的金属物体放置，也会改变此导线允许通过的信号频率范围。类似地，金属物体可以阻挡一部分频率的无线电波，而允许其他部分通过。
- 衰减（Attenuation）。信号通过介质传输后，信号强度会变弱。工程师就说信号已经被衰减了。因此，通过导线和光纤经长距离传输的信号会变得越来越弱，无线电信号也是这样，经过长距离传播后也会变得越来越弱。

香农定理表示了一种减少差错的方法：增加信噪比（通过增加信号或者降低噪声）。即使诸如屏蔽导线这样的机制有助于降低噪声，但物理传输系统对差错总是敏感的，也不太可能改变传输系统的信噪比。

虽然传输差错不能被完全消除，但是很多都是可以被检测到的。在一些情况下，系统可以自动纠正差错。我们将看到有时候差错检测被过分强调了。因此，系统设计者必须仔细考虑差错是否可能发生，以及发生差错后的后果是什么（如在银行转账中单个位差错有可能意味着超过一百万的差别，而在图像传输中单个位差错根本不重要）。因而所有对差错的处理实际上是设计者要做出的权衡措施，这里的要点是：

> 虽然传输差错是不可避免的，但差错检测机制却要增加开销，因此，设计者必须根据需要正确地选择采用哪种差错检测和补偿机制。

8.3 传输差错对数据的影响

数据通信只关注于传输差错对数据的影响，而不是去检查引起传输差错的确切原因。图8-1列举了传输差错影响数据的3种主要途径。

虽然任何传输差错都可能引起每种可能的数据差错，图8-1指出了潜在的传输差错通常会通过特定的数据差错表现出来。例如，一种称为毛刺（spike）的干扰，持续时间特别短，这种干扰往往是出现单个位错误的原因。持续时间稍长的干扰或失真则会产生突发性错误。有时信号难以清晰区分是0还是1，而是落在一个模糊区域，这种差错被称为擦除（erasure）。

差错类型	描　述
单个差错	在一串码位中仅有单个码位被改变（通常是由于受到非常短时间的干扰导致）
突发差错	在一串码位中有多个码位被改变（通常是由于受到稍长时间的干扰导致）
擦除（模糊）	到达接收方的信号非常模糊（不能清晰对应到逻辑1或逻辑0）（有可能是失真或干扰导致）

图8-1 数据通信系统中的3种数据差错类型

对于突发性差错，突发尺寸（burst size）或突发长度（burst length）被定义为从第一个错误码位到最后一个错误码位的总位数。图8-2表示出这个定义。



图8-2 用阴影标识的突发性差错示意图

8.4 处理信道差错的两种策略

为了克服数据差错从而增加可靠性，人们开发了多种数学技术。这些技术被统称为信道编码（channel coding），可以分为两种主要类型：

- 前向纠错（Forward Error Correction，FEC）机制。
- 自动重传请求（Automatic Repeat-reQuest，ARQ）机制。

前向纠错的基本思想非常直接：发送时在数据中增加冗余信息，接收方收到数据后可以根据冗余信息校验所接收数据是否正确，甚至在有可能的时候进行纠错。图8-3说明了前向纠错机制的概念性流程。

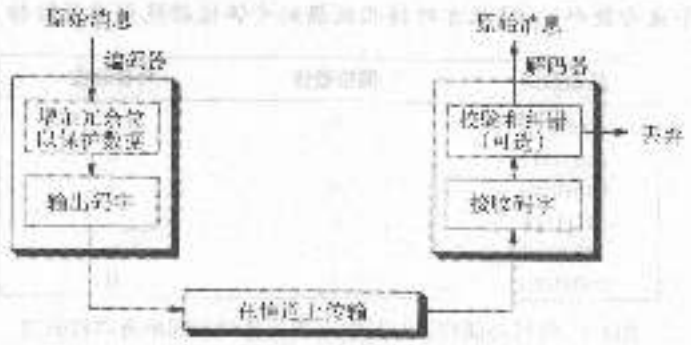


图8-3 前向纠错机制的概念性流程

基本的差错检测机制（error detection mechanisms）能使接收方及时检测到差错的发生；前向纠错机制则能使接收方准确判定那个位被改变并计算出正确的值。

信道编码的第二种方法，就是大家熟知的ARQ[⊖]，需要发送方的合作——发送方和接收方之间交换信息，以确保接收方所接收的所有数据都是正确的。

8.5　分组码和卷积码

有两类前向纠错技术可以满足不同的需求：

- 分组码（Block Error Codes），分组码先把要发送的数据划分成一系列分组，然后给每个分组添加额外信息，叫做冗余（redundancy）。对一个分组的编码仅仅依赖于本组的位元，而与前面已发送的位元无关。因为这种编码机制不用将一个分组数据的状态信息携带给下一个分组，所以从这个意义上来说分组码是无记忆的（memoryless）。
- 卷积码（Convolutional Error Codes），卷积码是把数据作为一个位元序列来处理，并根据这个连续的位元序列计算编码。因此，要计算出一组位元编码既要依赖于当前的输入，又要依赖于此前输入的部分位流，因而卷积码被称为有记忆的编码。

当用软件来实现编码时，通常卷积码要比分组码要求更多的计算量。不过，卷积码也往往具有更高的检错概率。

8.6　分组差错编码举例：单奇偶校验

为了理解如何使用冗余信息来实现差错检测，可以考虑单奇偶校验（single parity check，SPC）机制。SPC先定义每8位数据单元（即一个字节）为一个分组。在发送方传输每个字节之前，编码器在该字节中增加一个额外的位，称为奇偶位（parity bit），接收方收到分组后，去除奇偶位并用它校验接收字节中的数据是否正确。

在使用奇偶校验前，发送方和接收方必须设定究竟是使用偶（even）校验规则还是奇（odd）校验规则。当使用偶校验规则时，如果需要编码的字节中已含有偶数个1，则发送方选择的奇偶校验位为0；而如果需要编码的字节已含有奇数个1，则奇偶校验位为1，记作这个定义的方法是：通过信道发送的9位码元中含有偶数个1，就是偶校验；通过信道发送的9位码元中含有奇数个1，就是奇校验。图8-4给出示例列举了数据字节以及使用奇偶校验时添加的相应奇偶校验位的值。

概括如下：

> 单奇偶校验是信道编码的基本形式，发送方给每个字节增加一个冗余位，使得每组含有偶数个或奇数个1，接收方对接收数据的奇偶校验规则进行检验，看是否正确。

原始数据	偶校验位	奇校验位
00000000	0	1
01011011	1	0
01010101	0	1
11111111	0	1
10000000	1	0
01001001	1	0

图8-4　使用奇偶校验时数据字节及其对应的单奇偶校验位

⊖　第8.15与将会介绍ARQ。

单奇偶校验是信道编码中能力较弱的一种形式，它只能检测差错，不能纠正差错。此外，奇偶机制只能检测到奇数个位元被改变的情况。如果9个位元（包括奇偶位）中的一个在传输期间被改变，接收方会判定接收字节无效。然而，如果在传输期间由于突发差错引起2、4、6或8个位元被改变，接收方则会误将接收字节归类为有效的字节。

8.7 分组码数学与 (n, k) 表示

前向纠错机制是把一组消息作为输入，然后对它插入一些附加位，从而产生一个已编码消息。数学上，我们定义所有可能的消息集合为一个数据字（datawords）集合，并定义所有可能的已编码消息为一个码字（codewords）集合。如果一个数据字包含k个数据位和r个附加位形成一个码字，我们就说这种编码结果是：

(n, k) 编码方案

其中：$n = k + r$。这种方案能实现成功检错的关键，在于从2^n个可能的组合中选择有效码字的子集。这个有效子集被称为码簿（codebook）。

作为示例，这里考虑单奇偶校验。数据字分组包含8位的任意可能组合。因此，$k = 8$，有2^8（或256）个可能的数据字。发送数据含有$n = 9$位，因此有2^9（或512）种可能。然而，在512个值中仅有一半是有效码字。

考虑n位的所有可能值集合以及形成码簿的有效子集。如果传输中出现差错，在码字中会有1位或多位被改变，这种变化有可能生成另一个有效码字，也有可能生成一个无效的组合。例如，在上面讨论过的单奇偶校验方案中，有效码字中单个位元被改变会生成一个无效组合，但是改变两个位元则会产生另一个有效码字。显然，我们想要的是出现差错而产生无效组合的编码方案。一般来说：

理想的信道编码方案是：改变一个有效码字中的任意位元，就产生一个无效组合。

8.8 汉明距离：编码强度的测量

没有哪种信道编码方案是完全理想的——一个码字中有足够多的位元被改变之后，它总有可能转变为另一个有效码字。因此，对于实际方案，就存在这样的问题：由一个有效码字转变为另一个有效码字，最少要改变多少个位元？

为了回答这个问题，工程人员使用一种称为汉明距离（Hamming distance）的测量方法。这是以贝尔实验室一位理论家的名字命名的，他是信息理论和信道编码领域的先驱。给定两个n位的位串，汉明距离定义为两个位串中对应位不同的数量（即要把一个位串转变为另一个位串最少需要改变的位数）。图8-5说明了这个概念。

d(000, 001) = 1	d(000, 101) = 2
d(101, 100) = 1	d(001, 010) = 2
d(110, 001) = 3	d(111, 000) = 3

图8-5 多对3位串的汉明距离示例

计算汉明距离的一种方法，就是对两个位串进行异或（xor）运算，并计算出异或运算结果中1的个数。例如110和011这两个位串，对它们进行异或运算，其结果是：

$$110 \oplus 011 = 101$$

异或结果中含有两个1，因此110和011之间的汉明距离就等于2。

8.9　码簿中码字之间的汉明距离

如前所述，我们的兴趣在于差错能否将一个有效码字转换成为另一个有效码字。为了测量这样的转换，我们来计算一下给定码簿中所有码字之间的汉明距离。作为一个小例子，考虑对两位长的数据字应用偶校验。图8-6列举了4个可能的数据字、4个添加奇偶位后的可能码字以及码字之间的汉明距离。

数据字	编码字
00	001
01	010
10	100
11	111

a)

d(001,010) = 2
d(010,100) = 2
d(001,100) = 2
d(010,111) = 2
d(001,111) = 2
d(100,111) = 2

b)

图8-6　a) 两位长的数据字及添加单奇校验编码后的码字；b) 所有码字之间的汉明距离

码字的全体集合称为码簿（codebook）。我们使用d_{min}表示码簿中所有码字之间的最小汉明距离（minimum Hamming distance），到底要多少位差错才会引起从一个有效码字转换为另一个有效码字呢？最小汉明距离就是对这个问题的精确回答。在图8-6的单奇校验的例子中，集合中包含所有码字之间的汉明距离，并且$d_{min}=2$。这意味着，如果在传输中出现两个位元错误，至少有一个有效码字可能被转换为另一个有效码字。这里的要点是：

把一个有效码字转换为另一个有效码字，最少需要改变多少位？找到这个答案的方法就是计算码簿中所有码字之间的最小汉明距离。

8.10　差错检测与代价之间的权衡

对于一个编码字集合，最好具有较大的d_{min}值，因为编码对于位元差错具有较好的免疫性——如果位差错小于d_{min}，该编码就可以检测到差错的发生。公式（8.1）确定了最小汉明距离d_{min}与可以检测到的最大的出错位数e之间的关系：

$$e = d_{min} - 1 \tag{8.1}$$

差错编码的选择需要仔细权衡　虽然d_{min}值越大可以检测到的差错越多，但是d_{min}值越大的编码所需要发送的冗余信息也就越多。为了测量编码开销的大小，工程上定义编码效率（code rate）来表示数据字长度与编码字长度的比值。公式（8.2）定义了(n, k)差错编码方案中的编码效率R。

$$R = \frac{k}{n} \tag{8.2}$$

8.11　采用纵横奇偶校验的纠错

我们已经知道信道编码方案可以实现检测差错的原理。为了理解利用编码实现纠错的原理，先来看一个例子。假设数据字含有k=12位，但不要把它看作是一个位串，而是想象把这个数据字安排成为一个具有3行4列的矩阵，而且在每行和每列分别增加一个奇偶位。图8-7说明了这种排列，这就是众所周知的纵横奇偶校验（Row And Column，RAC）码。这个RAC编码例子的n=20，意味着这是一个（20，12）编码方案。



图8-7 纵横奇偶校验编码实例：数据位安排成3×4矩阵，并对每一行和列进行偶校验

为了弄清纠错的原理，假定图8-7中的一个数据位在传输中被改变。接收方把接收到的位串安排成为矩阵并重新计算奇偶位，显然有两个计算结果将与原定的偶校验规则不相符，如图8-8所示。

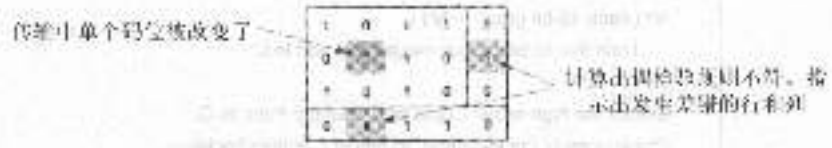


图8-8 纵横奇偶校验编码纠正单比特差错的原理

如图8-8所示，单个位元差错将导致纵向奇偶位和横向奇偶位的计算结果与原定的奇偶检验规则不相符，分别对应了出错码位所在的行和列，从而可以唯一地确定出错码位的位置。接收方使用计算的奇偶位判定哪个确切的数据位有差错，然后纠正数据位。因此，RAC可以纠正所有单个数据位被改变的差错情况。

如果使用RAC编码的数据块中差错位多于一个，会怎样呢？RAC只能纠正单个位差错，当有2或3个位被改变时，使用RAC编码将能检测到奇数个码位的差错。

概括如下：

> 纵横奇偶校验编码技术使接收方能纠正所有单个位元的差错，并能检测到2或3个位元被改变的情况。

8.12 用于因特网的16位校验和

在因特网中，有一种特殊的信道编码技术起到关键作用，那就是众所周知的因特网校验和（Internet checksum），这个编码由一个16位的反码校验和构成。它不强求数据字要固定长度。相反，这种算法允许报文为任意长度，对整个报文计算校验和。实质上，因特网校验和是把报文中的数据当作一系列16位的整数来处理，如图8-9所示。

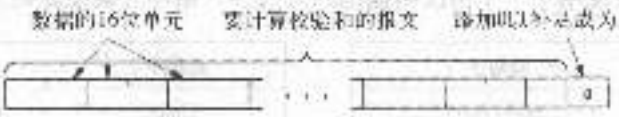


图8-9 因特网校验和把数据划分为16位单元，如果不足16位单元的整数倍则填充零

发送方把逐个16位整数值相加，最后对和值计算其反码作为校验和[⊖]，并将此结果发送出去。为了验证报文是否出错，接收方执行相同的计算过程即可，算法8-1给出了计算的细节。

理解这个算法的关键是要认识到：这个校验和是被计算成反码形式的，而不是在大多数计算机上所采用的补码计算，而且计算是使用16位整数而不是32或64位整数。这样的话，在编写这个算法时，要使用32位补码算术来完成反码计算。在for循环中，加法运算可能会溢出，

⊖ 原书这里没有对校验和值计算反码的步骤，疑是笔误，特此纠正。——译者注

因而在循环过程中，算法要把溢出（高阶进位）部分加回到和值中去。

```
                        算法8-1
Given:
   A message, M, of arbitrary length
Compute:
   A 16-bit 1s complement checksum, C, using 32-bit arithmetic
Method:
   Pad M with zero bits to make an exact multiple of 16 bits
   Set a 32-bit checksum integer, C, to 0;
   for (each 16-bit group in M) {
      Treat the 16 bits as an integer and add to C
   }
   Extract the high-order 16 bits of C and add them to C;
   The inverse of the low-order 16 bits of C is the checksum;
   If the checksum is zero, substitute the all 1s form of zero
```

算法8-1　用在因特网协议中的16位校验和算法

为什么校验和是计算总和的算术反码而不直接使用和值呢？答案是为了效率——接收方可以使用与发送方相同的算法，相加的整数中还包括校验和本身。因为校验和包含和值的算术反码，所以把校验和与和值相加的结果将为0。因此，接收方在计算中包括校验和，就可以判断计算结果是否为零来检测被接收数据是否存在传输差错。

8.13　循环冗余校验码

在高速数据网络中使用的一种信道编码称为循环冗余检验码（Cyclic Redundancy Code，CRC）。CRC码的3个关键特性使其在数据网络中显得非常重要，图8-10对这3个特性作了总结。

任意长度报文	与网检验和一样，数据字长度不固定，这意味着CRC可用于对任意长度报文的编码
出色的检错能力	因为检验和的计算取决于报文中位元顺序，所以CRC可生成很出色的检错能力
快速硬件实现	虽然具有复杂的数学基础，但CRC计算步骤可以用硬件快速完成

图8-10　使CRC码在数据网络中显得重要的3个关键方面

这里所谓的循环（cyclic），是来自编码字的一个属性：对一个编码字进行循环移位产生另一个编码字。图8-11说明了一个由汉明提出的（7，4）循环冗余检验码例子。

数据字	编码字
0000	0000 000
0001	0001 011
0010	0010 110
0011	0011 101
0100	0100 111
0101	0101 100
0110	0110 001
0111	0111 010

数据字	编码字
1000	1000 101
1001	1001 110
1010	1010 011
1011	1011 000
1100	1100 010
1101	1101 001
1110	1110 100
1111	1111 111

图8-11　（7，4）循环冗余检验码示例

CRC码已经被广泛研究，并产生了诸多数学解释和计算技术。对这些解释和技术的各种描述看上去非常不一样，以致让人难以理解它们为何能反映出同一个概念来。主要的观点包括：

- 数学家把CRC计算解释为两个具有二进制系数的多项式相除所得的余数，其中一个多项式表示报文，而另一个多项式则表示一个固定的除数。
- 计算机理论科学家把CRC计算解释为两个二进制数相除所得的余数，其中一个数表示报文，而另一个数则表示一个固定的除数。
- 密码学家把CRC计算解释为在二阶伽罗域（写成GF(2)）中的一种运算。
- 计算机程序员把CRC计算解释为对报文进行迭代并在每一步查表获得参加值的一个算法。
- 硬件架构师把CRC计算解释为一个小的硬件流水线（pipeline）单元，它以报文中的位元系列作为输入，不使用除法或迭代操作就可以生成CRC。

作为上述观点的一个例子，考虑无进位的二进制数除法。图8-12说明了计算的过程，其中被除数1010代表报文数据，除以一个为产生特定的3CRC而选择好的一个常数1011。

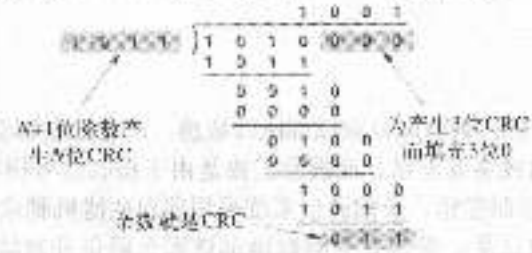


图8-12 视为无进位二进制数相除的余数即是CRC的计算示意图

为了理解为什么数学家把上述过程看作两个多项式相除，可以把二进制数中的每一位看成多项式的系数。例如，我们可以考虑图8-12中的除数1011，作为下面多项式的系数：

$$1\times x^3+0\times x^2+1\times x+1\times x^0=x^3+x+1$$

类似地，图8-12中的被除数1010000，表示为下面的多项式：

$$x^6+x^4$$

我们使用术语生成多项式（generator polynomial）来描述对应除数的多项式。生成多项式的选择是能够产生具有良好检错能力的CRC的关键，因此，在生成多项式的推导上已经有很多数学分析。例如，我们知道一个理想的多项式是不可约的（即只能被自己和1整除），并且具有一个以上非零系数的多项式就能检出所有单个码位的差错。

8.14 用硬件高效实现CRC

计算CRC所需的硬件简单得令人惊奇，它只是一个带有异或（exclusive or或xor）门的移位寄存器（shift register）而已，其中在某些位之间插入的异或门用来执行异或运算，移位寄存器每次操作一个输入位元。在每一级寄存器上，要么是接受从前一级移位来的输入位，要么进行一次异或运算并接受运算的结果。当全部输入都已经完成移位并进入寄存器后，寄存器里保存的值就是CRC。

图8-13图示说明了图8-12中3位CRC计算所需要的硬件。因为异或运算或者移位操作都可以高速进行，所以这种硬件配置可以用于高速计算机网络。

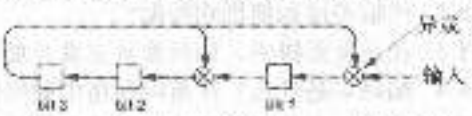


图8-13 使用生成多项式x^3+x+1计算3位CRC的硬件单元

8.15 自动重传请求（ARQ）机制

ARQ方法实现纠错要求发送方和接收方之间连续通信以维持信息沟通。也就是说，无论何时一方发送报文给另一方，而接收方都能立即回送一条短的确认（acknowledgement）报文。例如，如果A发送一个报文给B，B接收校验后送回一条确认报文给A。一旦接收到确认报文，A就知道该报文已经被对方正确接收。如果过了T时间单元仍然没有收到确认报文，A就假定该报文已在传输中丢失并重传该报文的一份副本给对方。

ARQ特别适用于下层系统提供差错检测但不提供差错纠正的情况。例如，许多计算机网络使用CRC检测传输差错，这时在传输方案中增加ARQ机制即可确保报文能正确传送到目的地——如果某报文发生了传输差错，接收方将其丢弃，要求发送方重传该报文的一份副本。

第26章将讨论采用ARQ方法的肯特网协议的细节。除了说明实践中如何使用超时重传机制外，本章也会解释发送方和接收方如何识别已经被确认过的报文，并讨论发送方在重传之前应该等待多长时间。

8.16 本章小结

物理传输系统对干扰、失真以及衰减都比较敏感，所有这些都会引起传输差错。传输差错有可能会导致单个差错或突发差错，而接收差错是由于接收信号模糊（即不能清晰判断是0还是1）所引起的。为了控制差错，数据通信系统采用前向纠错机制或使用自动重传请求技术。

前向纠错机制的做法是：发送方先对数据加进冗余码位并对结果进行编码，然后才通过信道传输。接收方对接收数据解码，然后对解码数据进行校验。如果数据字包含k个码位，编码字包含n个码位，这样的编码方案叫（n, k）码。

人们对某种编码性能的测量，就是对由于差错而引起一个有效编码字变成另一个编码字的可能性进行评估。最小汉明距离正是提供了一个精确的测量方法。

分组码以其简单性而著称。例如对每个字节加进单个奇偶位，就可以检测到奇数个位的差错，但不能检测到偶数个位的差错。纵横奇偶校验（RAC）码可以纠正单个码位差错，并检测一个数据分组中3个以内的差错，还能检测任意奇数个码位被改变的差错。

在因特网中使用的16位校验和算法适用于任意长度的报文。校验和算法是把报文划分为16位长的块，计算各块数据的算术和，然后取其反码即是校验和。计算各块算术和的过程中如果有溢出（即进位），要把溢出位也加回到校验和中。

循环冗余检验码（CRC）用于高速数据网络中，CRC接受任意长度的报文，提供极好的检错能力，并且可以用高效的硬件实现。CRC技术具有很好的数学基础，已经被广泛研究过。CRC计算可被视为：计算二进制数相除的余数，或计算多项式相除的余数，或是利用伽罗域理论的一种运算。执行CRC计算的硬件是采用移位寄存器和异或运算。

练习题

8.1 列举并说明3个主要的传输差错的来源。

8.2 传输差错如何影响数据？

8.3 在突发差错中，如何度量突发长度？

8.4 编码字是什么？在前向纠错中如何使用编码字？

8.5 举出一个针对字符数据所采用的分组差错编码的例子？

8.6 理想的信道编码方案能达到什么效果？

8.7 请定义出汉明距离的概念。

8.8 计算下列数据配对的汉明距离：(0000,0001)，(0101,0001)，(1111,1001)和(0001,1110)。

8.9 如何计算把一个有效编码字转换为另一个有效编码字所需改变的最少位数？

8.10 请解释编码效率的概念？并说明是高编码效率好还是低编码效率好？

8.11 对数据字100011011111进行(20，12)进行RAC编码，请你生成一个RAC奇偶矩阵。

8.12 请问RAC方案能完成单奇偶位方案却不能完成的是什么？

8.13 请编写一段能计算出16位因特网校验和的计算机程序。

8.14 CRC的特征是什么？

8.15 请做出10010101010除以10101的除法计算。

8.16 请将上题中的两个值表示成多项式。

8.17 请编写一段计算机程序，实现图8-11中(7,4)循环汉会码。

8.18 列出并解释用于实现CRC计算的两种硬件构件。

第9章 传输模式

9.1 引言

本书这部分的各章涵盖了数据通信组成部分的基本概念。这一章将继续讨论关于数据传输的方式，先介绍通用的术语，解释并行模式的优缺点，再讨论同步和异步通信的重要概念。后续章节将会表现出本章所介绍的概念是如何应用在整个因特网的各种网络中的。

9.2 传输模式分类

我们使用术语传输模式（transmission mode）来说明数据在底层介质中传输的方式。传输模式可以分成两个基本类型：

- 串行传输——一次发送一个码位。
- 并行传输——同时发送多个码位。

我们将会看到，串行传输又可以根据传输的定时形式的不同，做进一步的分类。图9-1给出了本章讨论的传输模式的整体分类方法。

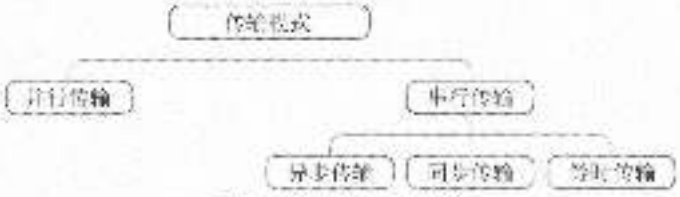


图9-1 传输模式分类法

9.3 并行传输

术语并行传输（parallel transmission）是指在分离的媒体上同时传输多个数据位的传输机制。通常来说，并行传输模式用于具有多根独立导线的布线介质。此外，所有导线上的信号都必须进行同步，这样，码元可以准确无误地同时通过每根导线。图9-2说明了这个概念，并表示出工程上为什么要使用术语并行（parallel）来表征这种连线方法。



图9-2 使用8根导线同时传输8个码位的并行传输示意图

这个图省略了两个重要细节。首先，除了承载数据的并行导线，并行接口上通常还包括

其他导线以使发送器和接收器能够彼此协调。其次，为了使安装和故障维修简单易行，并行传输系统的所有导线都捆扎成一根单独的物理线缆。因此，我们通常看到的是一根单独的粗电缆连接着发送器和接收器，而不是一组分离的物理导线。

并行传输模式有两个主要的优点：

- 高速（high speed）。因为并行传输可以同时发送N位，所以并行接口的传送速度是同等串行接口的N倍。
- 与下层硬件相匹配（match to underlying hardware）。在机器内部，计算机和通信硬件都使用并行电路，因此并行接口与内部硬件能较好地匹配。

9.4 串行传输

与并行传输对应的另一种传输模式，称为串行传输（serial transmission），一次发送一个码位。如只是以追求速度为主，可能设计数据通信系统的任何人都会选择并行传输。然而，大多数通信系统都使用串行模式，主要有两个原因。首先，串行网络长距离扩展的费用要低得多，因为需要的物理导线更少，并且一般中等的电子元件价格更便宜。其次，仅使用一根物理导线意味着不会有由于多根导线不等长而引起的定时问题（在高速通信系统中，导线之间几毫米的长度差别都可能引起严重影响）。

为了采用串行传输，发送器和接收器必须具备少量硬件，这些硬件能把设备中使用的并行数据转换为导线使用的串行数据。图9-3所示为这种配置。



图9.3 串行传输模式示意图

将数据在内部并行形式与串行形式之间转换时所需的硬件可以非常简单，也可以很复杂，这取决于串行通信机制的类型。在最简单的情况下，用一块称为通用异步收发器（UART）的芯片即可完成数据转换功能；用一块称为通用同步/异步收发器（USART）的芯片，即可处理同步网络的数据转换。

9.5 传输顺序：码元与字节

串行传输模式引入了一个有趣的问题，即当要发送码元序列时，应该先发送哪个码元通过介质呢？例如，考虑一个整数，发送器应该先传输最高位（MSB）还是最低位（LSB）呢？

工程上使用术语逆序（little-endian）来表述先发送LSB的系统，使用术语正序（big-endian）来表述先发送MSB的系统。两种形式都可以使用，不过在发送前发送器和接收器必须对使用何种形式达成一致。

有趣的是，码元传输次序的确定并没有解决传输顺序的所有问题。在计算机中数据划分为字节，每个字节再进一步划分为码元（通常是每字节包含8个码元）。因此，字节的传输顺序和码元的传输顺序可能要分别选择。例如，以太网技术规定数据的传输顺序是：字节按正序传输，码元按逆序传输。图9-4表示了在以太网中发送32位数据量时码元的传输顺序。



图9.4　字节正序、码元逆序的传输次序示意图，其中最高有效字节中的最低有效位最先发送

9.6　串行传输的定时

依据传输在时间上的间隔，串行传输机制可以分为3种主要类型：

- 异步（asynchronous）传输：数据项的传输可以在任意时间开始，两组数据项之间的间隔时长也可以是任意的。
- 同步（synchronous）传输：数据项连续不断地传输，数据项之间没有间隔。
- 等时（isochronous）传输：数据项在规则的时间区间上进行传输，两数据项之间的间隔是固定的。

9.7　异步传输

如果系统允许物理介质在两次传输之间空闲任意长时间，这种传输系统则归类为异步（asynchronous）传输系统。异步通信非常适合于随机产生数据的应用（例如用户敲打键盘或用户点击链接打开一个网页，读一会儿，再点击链接打开另一个网页）。

异步传输的缺点是发送器和接收器之间缺少协调——当介质空闲时，接收器不知道在新的数据到来前，介质还会空闲多长时间。因此，异步传输技术在每个数据项发送前，通常安排发送器发送一些冗余位以通知接收器开始传输数据。冗余位使得接收器硬件能与进入的信号同步。在一些异步系统中，冗余位被称为前导（preamble），在其他系统中，冗余位被称为开始位（start bits）。概括如下：

> 因为异步传输机制允许发送器在传输之间空闲任意长时间，所以其在每次传输之前要发送一些额外信息，以使接收器能与信号同步。

9.8　RS-232异步字符传输

作为异步通信的例子，考虑在计算机和设备（例如键盘）之间通过铜导线传输字符的情况。电子工业联盟（Electronic Industries Alliance，EIA）标准化的异步通信技术——称为RS-232-C（一般缩写为RS-232[⊖]），已经成为了最广泛接受的字符通信技术。EIA标准规定了物理连接的细节（例如，连接线长度必须小于50ft）、电气特性（例如，电压范围为−15~+15V）以及线路编码（例如，负电压对应逻辑1，正电压对应逻辑0）等。

因为RS-232技术被设计用于设备（例如键盘）之间的通信传输，所以该标准规定每个数据项对应一个字符。通过一定的配置，硬件可以控制每秒传输的确切码位数，可以发送7位元字符或者8位元字符。虽然发送器在发送前可以延迟任意长时间，但是一旦传输开始，发送器一位接一位地发送字符的所有位，码元之间没有间隔。一个字符发送结束后，发送器保持导线处于负电压（对应逻辑1），直到准备发送下一个字符。

⊖　虽然后来的RS-449标准提供了很多新功能，但大多数工程师仍然使用原来的名字。

接收器如何知道新的字符从哪里开始呢？RS-232规定发送器在发送一个字符的码元之前要先发送一个额外的0位，称为开始位（start bit）。而且，RS-232规定在发送的字符之间发送器必须使线路至少保留一个码元的空闲时间。因此，我们可以想象成每个字符后面都有一个附加码元1。在RS-232的术语中，这个附加位被称为停止位（stop bit）。图9.5为当发送一个开始位、一个字符的8个码位以及一个停止位时，电压如何变化的示意图。



图9.5 使用RS-232传输8位元字符时的电压变化示意图

概括 RS-232标准用于短距离的异步串行通信，它以一个开始位作为每个字符的开始，之后发送字符中的每个码元，在每个字符的后面至少跟着一个码元时间的空闲期（停止位）。

9.9 同步传输

与异步传输不同的另一个主要的模式，就是同步传输。同步机制要求连续地传输数据码元。码元之间没有空闲时间。也就是说，发送器在发送完数据字节的最后一位后，立即发送下一个数据字节的码元。

同步机制的最大优点在于发送器和接收器始终保持同步，这意味着同步额外开销较少。为了理解这种额外开销，我们可以比较8位元字符在如图9.5所示的异步传输系统中的传输和在图9-6所示的同步传输系统中的传输。使用RS-232标准传输每个字符需要一个额外的开始位和停止位，这意味着即使不插入空闲时间，每个8位元字符的传输也至少需要10位码元的传输时间。而在同步系统中，每个字符的传输不需要开始位和停止位。

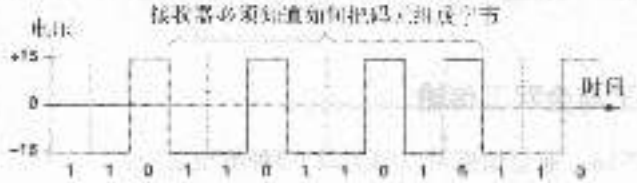


图9.6 同步传输示意图，字节第一位紧跟前一字节最后一位

要点 与同步传输机制相比，异步的RS-232机制传输每个字符要增加25%的开销。

9.10 字节、块和帧

如果底层的同步机制要求必须连续发送数据码元，但是发送器却没有随时准备好的数据可以发送，那该怎么办呢？办法就是采用一种被称为成帧（framing）的技术——在同步机制中增加一种接口，它负责接收和传递字节块（称为帧）。为了确保发送器与接收器同步，这种帧要以一种特殊的码元系列作为前导，而且，当发送器没有数据可发送时，大多数同步系统都要求传输特殊的空闲序列（idle sequence）或空闲字节（idle byte）。图9-7说明了这个概念。



图9.7　同步传输系统中的成帧示意图

成帧的重要性可以概括如下：

> 虽然底层同步传输机制要求连续发送数据码元，但如果利用空闲序列和成帧技术的话，也可以让同步传输机制提供面向字节的接口，因而就允许数据块之间存在空闲间隔。

9.11　等时传输

第三种类型的串行传输系统不提供新的底层传输机制，但可以把它看成采用同步传输机制的一种重要方式。这一类串行传输系统称为等时传输（isochronous transmission），它是为包含声音和视频的多媒体应用提供稳定比特流而设计的。以稳定的速率来发送音视频信息数据是非常有必要的，因为延迟的变化（称为抖动jitter）会破坏相应信息的接收效果（即会引起音频的颤抖或使得视频停顿一小会儿）。

等时网络不是利用数据的出现来驱动传输，而是有意将它设计成以固定速率R去接收和发送数据。事实上，等时网络接口本身就决定了数据被发送到网络进行传输的速率必须正好是Rbit/s。例如，设计用于传输声音的等时机制以64 000bit/s的速率运作，发送器必须连续产生数字音频，而接收者必须能接收和播放相应的音频流。

底层网络可以采用成帧技术，并可以选择额外信息跟随数据一起传输。然而，为了实现等时，系统必须经过仔细设计以使得发送器和接收器能看到连续的数据流，并且在帧的起始没有额外的延迟。因此，数据速率为Rbit/s的等时网络通常都具有其操作速率略大于Rbit/s的底层同步机制。

9.12　单工、半双工与全双工传输

根据传输方向的不同，通信信道可以归为以下3种类型之一：

- 单工信道。
- 全双工信道。
- 半双工信道。

单工（simplex）。单工机制是最容易理解的。顾名思义，单工机制只能在单一方向传输数据，例如，单条光纤可以实现单工传输，因为光纤的一端是发送设备（即发光二极管或激光器），而另一端是接收设备（即光敏接收器）。单工传输类似于无线电或电视广播。图9-8a对单工传输进行了说明。

全双工（full-duplex）。全双工机制也简单易懂：底层系统允许两个方向的传输同时进行。典型的全双工机制由两个单工机制组成，每个负责一个方向的信息发送，正如图9-8b所示。例如，将一对光纤并行排放并且安排它们分别朝相反的方向发送数据，这对光纤就可用于提供全双工通信。全双工通信很类似于电话会话时的情况，参与方在与对方讲话的同时，甚至

还能听到对方的背景音乐。

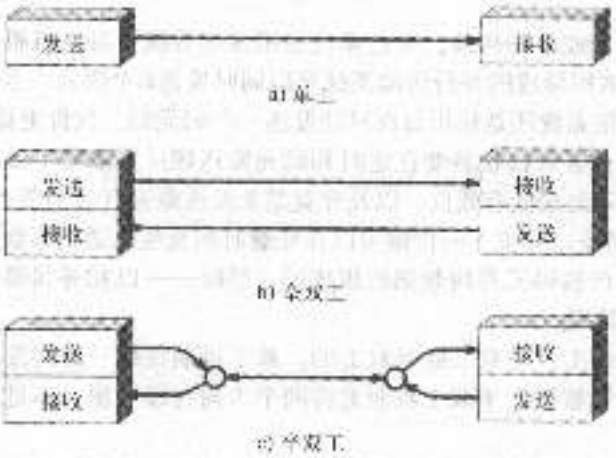


图9-8 3种操作模式示意图

半双工（half-duplex）。半双工机制要求有一个共享的传输介质，这个共享介质可用于每个方向的通信，但不同方向的通信不能同时进行。因此，半双工通信与使用对讲机相似，同一时间只能有一个方向的传输，在半双工通信中，介质的两端需要增加额外的机制来协调通信，以确保在给定的时间上只有一端发送信息。图9-8c说明了半双工通信。

9.13 DCE和DTE设备

术语数据线路设备[⊖]（Data Circuit Equipment，DCE）和数据终端设备（Data Terminal Equipment，DTE）是由AT&T最先提出的，以此来区分电话公司拥有的通信设备和用户所拥有的终端设备。

从术语来理解，应该是：如果一家企业从电话公司租赁了数据线路，那么电话公司就要在企业安装DCE设备，而企业则要购置DTE设备连接到电话公司的DCE设备上。

从学术的观点看，DCE-DTE之间重要的差别并不在于设备所有权问题，而在于为用户定义任意接口的能力问题。例如，如果底层网络使用同步传输，DCE设备既可以给用户提供同步接口，也可以提供等时接口。图9-9所示为这种概念性的组织关系。



图9-9 在两位置之间提供通信服务的DCE和DTE示意图

已经有几个标准可用于指定DCE和DTE之间可能的接口。例如，本章已描述的RS-232标准及其作为替代的建议标准RS-449，都是可用的。另外，还有X.21标准也可供使用。

⊖ 原书错写为“数据通信设备”（Data Communication Equipment），经为笔误。——译者注

9.14　本章小结

通信系统使用并行或串行传输。并行系统使用多根导线，每次每根导线只载送一个码元的信号。因此，具有K根导线的并行传输系统可以同时发送K个码元。虽然并行通信提供更高的速率，但大多数通信系统还是使用每次只能发送一个码元的、代价更低的串行机制。

串行通信要求发送器和接收器要在定时和码元发送顺序上取得一致。传输顺序是指究竟是先发送最高有效位还是最低有效位，以及究竟是先发送最高有效字节还是最低有效字节。

3种定时类型分别是：异步——传输可以在任意时间发生，通信系统可以在传输间隙保持空闲；同步——连续传输码元并将数据组织成帧，等时——以相等间隔传输数据，并且在帧边界处不带有额外的延迟。

通信系统可以是单工、全双工或半双工的。单工机制在单一方向发送数据；全双工机制同时在两个方向上传输数据；半双工机制允许两个方向传输数据，不过在给定时间只允许一个方向的传输。

提出数据线路设备和数据终端设备的区别，最初是为了指明设备究竟是由提供商拥有还是由用户拥有。而其关键的意义却是出自于它为用户定义接口的能力，凭此能力可以提供与底层通信系统不一样的服务。

练习题

9.1　描述串行传输与并行传输之间的区别。

9.2　并行传输的优点是什么？主要缺点呢？

9.3　以正序传输32位的补码整数时，何时发送符号位？

9.4　异步传输的主要特征是什么？

9.5　哪种类型的串行传输适用于视频传输？对于键盘与计算机之间的连接，用哪种类型呢？

9.6　开始位是什么？其用于哪种类型的串行传输？

9.7　使用同步传输方案时，发送器没有数据可供发送时会发生什么情况？

9.8　当两个人进行对话时，他们是使用单工、半双工还是全双工传输？

9.9　Modem属于DTE还是DCE？

9.10　请上网查找DB-25连接器中使用的DCE和DTE针脚的定义。提示：针脚2和3用于发送或者接收。在DCE型连接器中，针脚2是用于发送还是接收？

第10章 调制与调制解调器

10.1 引言

本书中数据通信这一部分的每一章都介绍数据通信的一个方面。上一章我们讨论了信息源，解释了信号如何表示信息，并描述了用于各种传输介质的能量形式。

这一章将继续讨论数据通信，关注于使用高频信号承载信息。本章讨论如何利用信息来改变高频电磁波，解释为什么这个技术非常重要，并说明如何使用模拟和数字输入。后续章节将扩展相关讨论，目的是解释如何将这项技术用于设计实现一种通信系统，这种通信系统可同时在共享介质上传输多路独立的数据流。

10.2 载波、频率和传播

许多远距离通信系统都使用连续振荡的电磁波，称为载波（carrier）。系统对载波实施小的改变，以此来表达正被发送的信息内容。为了理解载波的重要性，回顾一下在第7章中介绍的内容，电磁能量的频率确定了电磁能量的传播方式。为什么要使用载波？其动机是：人们希望能够选择一种传播特性良好且不依赖于数据发送速率的频率。

10.3 模拟调制方案

我们使用术语调制（modulation）表示系统根据正在发送的信息对载波所做的改变。从概念上讲，调制具有两个输入：载波和信号，生成被调制后的载波作为输出，正如图10-1所示。



图10-1 具有两个输入的调制概念图

其实，发送器只需改变电磁波的一个基本特性即可。因此，可以利用信号来调制电磁载波的方法有3种：

- 振幅调制。
- 频率调制。
- 移相调制。

前两种调制方法大家最为熟悉，已经得到广泛使用。事实上，它们并不是起源于计算机网络——早期它们被发明用于无线电广播，后来也用于电视广播系统。

10.4 振幅调制

一种称为振幅调制（amplitude modulation，简称调幅）的技术可以根据发送信息（即根据信号）成正比地改变载波的振幅。这种载波以固定频率连续振荡，而波的振幅随信号不断

变化。图10-2说明了一个未经调制的载波、一个模拟信息信号以及这两个信号实施振幅调制所得到的载波。

振幅调制非常容易理解，因为调制只改变了正弦波的振幅（即幅度）。而且，调制后载波的时域图形状与所使用的信号相似。例如，我们想象一下将图10-2c中的所有正弦波的波峰连接起来，就形成一个包络（envelope）曲线，显然这个曲线的形状与图10-2b中信号的形状是相同的。

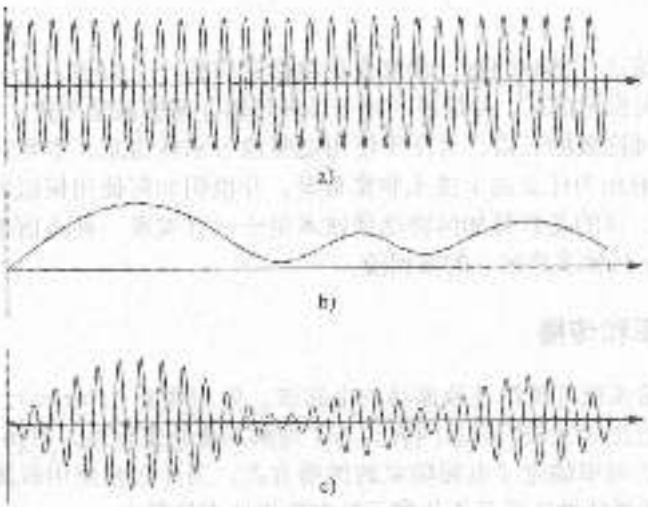


图10-2　a) 未经调制的载波，b) 模拟信息信号，c) 已调制载波

10.5　频率调制

可替代调幅的另一种调制技术称为频率调制（frequency modulation，简称调频）。采用调频技术时，载波的幅度保持不变，而频率却根据信号的改变不断发生变化。当信号增强时，载波频率略微提高；当信号减弱时，载波频率则略微下降。图10-3表示出根据图10-2b的信号进行频率调制后的载波图。

正如图中所示，频率调制较难可视化，因为轻微的频率改变无法清晰可见。不过，我们可以注意到，用于调制的信号较强时，被调制波则具有较高频率。



图10-3　以图10-2b所示信号进行频率调制后的载波波形图

10.6　相位调制

正弦波的第三个特性是相位，即正弦波起点相对于参考时间的移位值。利用相位的变化来表示信号是可能的，我们使用术语移相（phase shift）来表征这种变化。

虽然对相位进行调制在理论上是可能的，但这个技术却很少应用于模拟信号。为了理解

这是为什么，我们可以观察如果一个正弦波经过k个周期后发生相位变化，下一个正弦波的开始时间将相晚于k个周期完成的时间。波形的精确延迟与频率的变化非常类似，因此，对于模拟输入，移相调制可以被认为是一种特殊形式的频率调制。然而，我们将看到，在使用数字信号进行载波调制时，移相是非常重要的。

10.7 调幅与香农定理

图10-2c所示的幅度变化范围从最大值到接近于零。虽然这种图示有助于人们的理解，但却有一定的误导作用。实际上，调制仅仅会轻微改变载波的振幅，这种改变取决于一个称为调制指数（modulation index）的常数。

为了理解实际系统为什么不允许调制信号接近于零，可以考虑香农定理。假设噪声的总量是常数，当信号接近于零时，信噪比也会接近于零。因此，保持载波波形接近于最大值可以确保信噪比保持尽可能大，这样就允许每秒传输更多的码元数。

10.8 调制、数字输入和键控

以上对调制的描述说明了模拟信息信号是如何用于载波调制的。问题出现了："如何用数字输入进行调制？"答案是对上面所描述的调制方案的直接修改：数字方案使用离散值，而不是对连续信号成比例地调制。此外，为了区分模拟和数字调制，我们使用术语键控（shift keying）而不是调制来指后者。

本质上，键控操作类似于模拟调制，但它不采用连续的任意值对载波进行调制，而是只使用一组固定值的集合。例如，调幅允许载波幅度在一个小范围内任意变化，以此对应所使用信号的变化。与此不同，移幅键控使用一组可能幅度的固定集合。在最简单的例子中，全幅对应于逻辑1，而一个明显较小的幅度可以对应于逻辑0。与此相似，移频键控则使用两组基本频率。图10-4所示为一个载波，一个数字输入信号以及分别使用移幅键控（amplitude-shift keying）和移频键控（frequence-shift keying）所产生的波形。

10.9 移相键控

虽然利用幅度或相位的改变对于传播音频类信息能得到较好的效果，但除非采用特殊编码方案（例如信号的正负部分独立变化），否则这两种机制都要求至少一个载波周期才能发送一个码元的信息。第6章所述的奈奎斯特定理提示我们：如果编码方案允许每个载波周期编码多个位，那么每单位时间发送的码元数就会相应增加。因此，数据通信系统通常使用可以发送更多码位的技术，尤其是移相键控（phase shift keying）技术，它通过突然改变载波波形的相位来实现数据编码。每一次的这种改变称为相移（phase shift）。相移后，载波继续振荡，然后突然跳到正弦波周期中的一个新点。图10-5说明了相移是如何影响一个正弦波的。

相移以它改变的角度来进行测量。例如，图10-5最左边的相移变化是π/2弧度（或180°）。图中的第二个相位变化也对应180°的相移。第三个相位变化对应相移为-90°（也等于270°）。

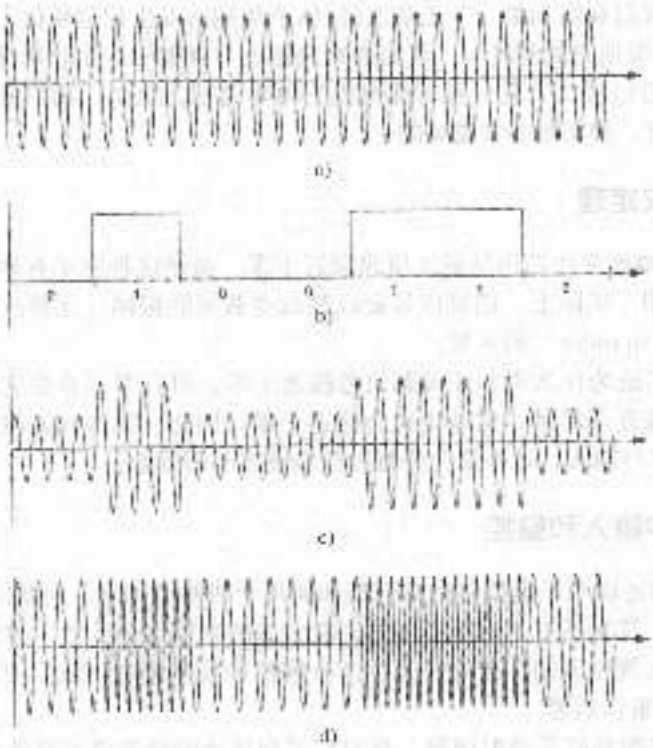


图10-4　a)载波，b)数字输入信号，c)移幅键控，d)移相键控

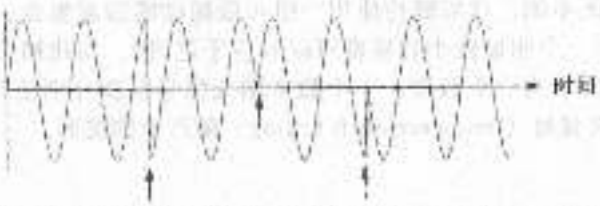


图10-5　移相调制示意图，箭头指示出载波突然跳变到正弦波后期一个新点的时间

10.10　相移与星座图

如何利用相移对数据进行编码呢？在最简单的情况下，发送器和接收器要对每秒发送的码位数达成一致，并规定不发生相移表示逻辑0，出现相移表示逻辑1。例如，系统可能采用180° 的相位偏移。星座图（constellation diagram）用于表达确切分配给数据位的具体的相位变化。图10-6对这个概念进行了说明。

硬件可以检测到相位偏移的出现，接收器还可以测量载波相位变化的偏移值，因此，可以设计一个能识别一组相位偏移值的通信系统，使得每个特定相移值分别代表一个具体的数值。通常，系统设计的相移数为2的幂值，这意味着发送器可以用不同相移来区分多码元组合的数据。

图10-7说明了一个使用4（即2^2）个相移值的系统星座图，在传输的每个阶段，发送器使用两位元数据在4个可能的相移值之间选择。

概括如下：

键移相键控那样的调制机制其主要优点是，具有每次变化可以表示多于一个数据位的能力。图10-6表示了数据位与相位变化的对应关系。

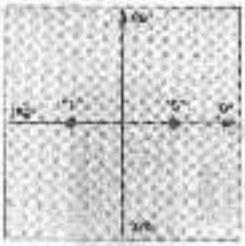


图10-6 0°相移代表逻辑0，180°相移代表逻辑1的星座图

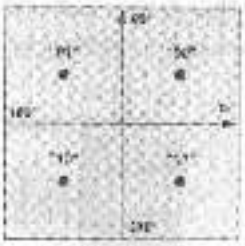


图10-7 使用4个相移，每个相移代表两位元数据的星座图

移相键控有多个变种技术。例如，允许发送器一次传输一个码位的移相键控机制，就如图10-6所示的移相键控，它们被归类为二相移相键控（Binary Phase Shift Keying，BPSK）机制。使用记法"2-PSK"来表示两个可能的相移值。类似地，如图10-7所示的形式称为4-PSK机制。

在理论上，可以通过增加相移的范围从而增加数据传输速率。因此，16-PSK机制每秒可以发送的数据码位数是4-PSK机制的两倍。然而，在实际应用中，噪声和失真限制了硬件辨别相移间微小差别的能力。

要点 虽然存在许多变种的移相键控机制，噪声和失真却限制了实际系统区分任意小相位变化的能力。

10.11 正交调幅

如果硬件不能检测任意的相位变化，那如何才能进一步提高数据速率呢？答案在于调制技术的结合——同时改变载波的两个特征。最巧妙的技术就是结合调幅和移相键控，这种技术称为正交调幅[⊖]（Quadrature Amplitude Modulation，QAM），使用同时包含相位和振幅的改变来表示数据[⊜]。

为了在星座图中表示QAM，我们使用与原点之间的距离作为振幅的量度。例如，图10-8表示了一个称为16QAM的变种机制的星座图，其中灰暗区域表示振幅。

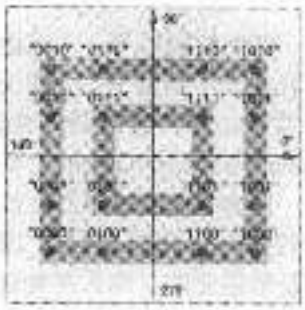


图10-8 16QAM的星座点，其中离原点的距离反映幅度

10.12 调制解调器硬件

接受数据码元系列并根据码元将信息调制到一个载波波形上，实现这种机制的硬件被称为调制器（modulator）；接受调制载波波形并将被调制到载波上的数据码元序列重新生成出来，实现这种机制的硬件被称为解调器（demodulator）。因此，数据传输要求传输介质的一端具有调制器，而另一端具有解调器。实际上，大多数通信系统都是全双工通信，这意味着每

⊖ 文献通常使用术语正交调幅（quadrature amplitude modulation）而不是正交移幅键控（quadrature amplitude shift keying）。

⊜ 原文中使用values，疑为误用，应为data才对。——译者注

一端都必须有一个调制器和解调器，其中调制器用于发送数据，而解调器用于接收数据。为了降低成本，并使得这一对设备容易安装和操作，生产商把调制和解调机制合并到一个单独的设备中，这种设备被称为调制解调器（modulator and demodulator，缩写成modem）。图10-9说明了一对modem如何使用4线制连接来进行通信的。

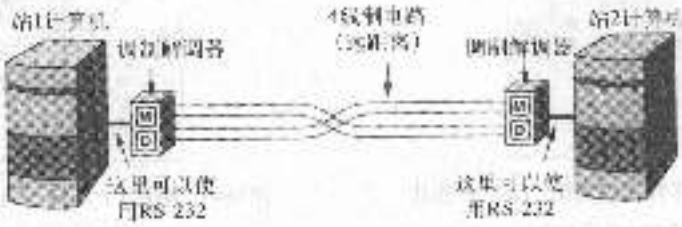


图10-9　采用4线制连接的两个调制解调器示意图

正如图中所示，调制解调器的设计用于提供远距离通信。连接两个调制解调器的4线制线路可以延伸到楼宇内，以及跨越楼宇之间或者城市之间的公司区号[⊖]。

10.13　光纤和射频调制解调器

除了专用导线外，调制解调器也可用在其他介质上，包括射频传输和光纤。例如，一对射频（Radio Frequency，RF）调制解调器可用于通过无线电波发送数据，而一对光学调制解调器（optical modems）可用于通过一对光纤发送数据。虽然与使用专用导线的调制解调器相比，这类调制解调器使用完全不同的介质，不过原理还是相同的，即在发送端，modem将数据调制到载波上，在接收端，modem从已调载波上提取出数据来。

10.14　拨号调制解调器

调制解调器另一个有趣的应用涉及音频电话系统。拨号调制解调器（dialup modem）使用音调而不是电子信号作为载波。当它作为传统的调制解调器来使用时，在发送端对载波进行调制，在接收端则对载波施行解调。因此，除了发起和接收电话呼叫的能力外，拨号modem与传统modem的主要区别，就在于可听音调具有较低的带宽。

在拨号调制解调器被首先设计出来的时候，这一方法完全合乎常理——拨号modem把数据转换为已调模拟载波，因为电话系统传输模拟信号。具有讽刺意味的是，现代电话系统内部是数字化的，那么在发送端拨号modem使用数据去调制可听音载波，这些载波被传输到电话系统。电话系统又要把输入的音频信息数字化，内部以数字形式传输，并把数字形式再转换回模拟音频信息，最后分发到接收端。图10-10说明了拨号modem对模拟和数字信号这种讽刺性的应用。

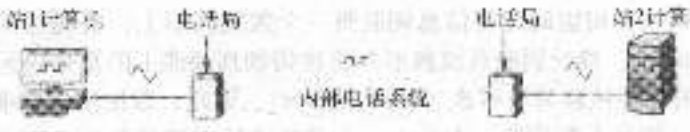


图10-10　使用拨号modem在计算机之间发送数据的示意图
（图中数字和模拟信号分别用方波和正弦波表示）

⊖　跨越公共场所的线路必须从服务提供商租用，这类服务提供商通常是电话公司。

如图10-10所示，拨号modem经常被内嵌于计算机内部。我们使用术语内置调制解调器（internal modem）意指嵌入的设备，而使用术语外置调制解调器（external modem）意指分立的物理设备。

10.15 应用于拨号的QAM

正交调幅也应用于拨号调制解调器，以作为最大化数据发送速率的一种方法。为了理解其中的原因，考虑图10-11，这个图表示出在拨号连接上可获得的带宽。正如图10-11中所示，大多数电话连接传输的频率范围为300~3 000Hz。但一个给定连接往往达不到对它的最大利用。因此，为了保证较好的再生能力和较低的噪声，拨号调制解调器使用600~3 000Hz的频率范围，这意味着可用带宽为2 400Hz。采用QAM方案可以大幅提高数据速率。



图10-11 拨号电话连接上语音和数据带宽的示意图

10.16 V.32与V.32bis拨号modem

作为使用QAM的拨号modem的例子，我们来看看V.32和V.32bis标准。图10-12所示说明了V.32 modem的QAM星座图，在这个方案中使用了32种幅移与相移的组合，由此可以获得双向9600bit/s的数据速率。

一个使用128种幅移与相移组合的V.32 modem，可以获得双向14 400bit/s的数据速率。图10-13所示为这个modem的星座图。在这个方案中，需要有更加精确的信号分析技术，才能检测到从星座图中的一个点到相邻的另一个点的很小的信号变化。

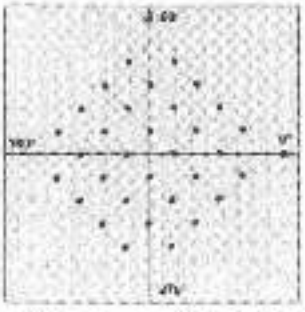


图10-12 V.32拨号modem的QAM星座示意图

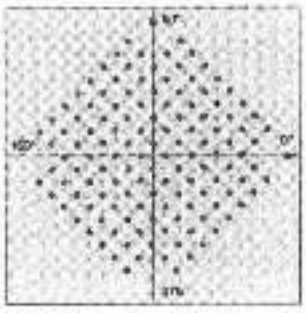


图10-13 V.32拨号modem的QAM星座示意图

10.17 本章小结

远距离通信系统使用已调载波的波形来传输信息。对载波的调制是通过改变振幅、频率或相位来实现的。调幅和调频是最常用于模拟输入的调制形式。

当以数字信号作为输入时，调制被称为键控。与模拟调制一样，键控也是改变载波，但

这时改变的值只允许在一个固定的可能值集合范围内。星座图用于表示移相键控的可能值。如果系统允许的可能值是2的幂，则每次可用多个输入码元从星座图的点中选择一个可能值。正交调幅把移幅键控和移相键控结合起来，产生了更多的可能值。

调制解调器是硬件设备，其中包含既可执行调制功能又可执行解调功能的电路。一对调制解调器可用于全双工通信。调制解调器也用在光、射频和拨号电话线路上。由于拨号电话线路带宽有限，所以拨号modem采用正交调幅方案来提高数据速率。V.32调制解调器使用相移和振幅改变的32个可能组合；V.32bis调制解调器使用128个可能组合。

练习题

10.1　列举模拟调制的3种基本类型。

10.2　当采用调幅时，用2Hz的正弦波去调制1Hz的载波是否有意义？为什么？

10.3　使用香农定理，解释为什么实际的调幅系统要使载波保持接近于最大限度？

10.4　键控与调制之间的区别是什么？

10.5　在移相键控中，相移有没有可能为90°，270°或者360°？请画出一个实例来解释你的答案。

10.6　使用互联网搜索一张32QAM的图片，在每个象限区域中定义了多少个点？

10.7　图10-9表示使用4根导线实现全双工配置的方案，每两根用于一个方向的传输，请讨论一下是否有可能改为使用3根导线来实现。

10.8　上题中，为什么使用4根导线更好些？

10.9　假设信噪比为30dB，使用图10-11所示的拨号带宽，可获得的最大数据速率是多少？

第11章 复用与解复用

11.1 引言

本书这部分的各章涵盖了数据通信的基础知识。上一章讨论了调制的概念，并解释了如何对载波实施调制以承载模拟或数字信息。

这一章继续讨论数据通信，介绍有关多路复用的相关知识。本章先阐述产生复用技术的起因，然后定义应用于计算机网络和因特网的基本复用类型。本章对已调载波如何为许多复用机制提供基础也进行了解释。

11.2 复用的概念

术语多路复用（multiplexing，简称复用）是指来自多个信源的信息流组合在一条共享介质上传输，并用复用器（multiplexor）来实现这种组合的机制。类似地，我们使用术语解复用（demultiplexing）来说明从组合信息中分离并还原出各自的信息流，并用解复用器（demultiplexor）来实现这种分离的机制。复用与解复用并不局限于硬件或者单个的位流——组合和分离通信的思想形成了在计算机联网的很多场合都要用到的基础技术。图11-1说明了复用与解复用的概念。

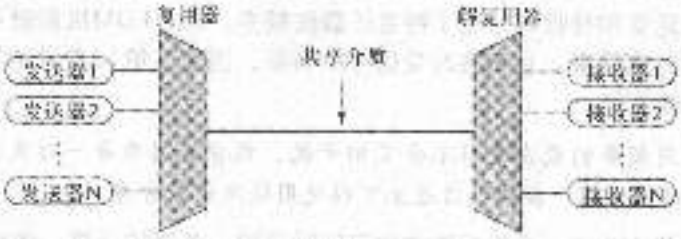


图11-1 复用的概念：多对发送器与接收器共享一条传输介质

图中，每个发送器与单个接收器通信。虽然每一对发送器与接收器之间的通信是各自独立进行的，但所有通信对却共享着一个传输介质。复用器将各发送器传来的信息进行组合传输，解复用器根据这样的组合方式再把各路信息分离出来并提交给接收器。

11.3 复用的基本类型

有4种实现复用的基本方法，并且每一种都有它自己的一套形式和实现方案。

- 频分多路复用（FDM）。
- 波分多路复用（WDM）。
- 时分多路复用（TDM）。
- 码分多路复用（CDM）。

时分和频分复用是广泛使用的多路复用方法；波分复用是用于光纤介质的一种频分复用方式；码分复用是用于蜂窝电话机制中的一种数学方法。

11.4　频分多路复用

频分多路复用（Frequency Division Multiplexing，FDM）非常容易理解，因为它形成了无线电广播的基础。这种技术的底层原理来自于传输的物理学原理——一组广播电台如果它们各自使用不同的频道（即载波频率），就可以同时发生电磁信号而不会互相干扰。数据通信系统应用同样的原理，在单根铜导线上同时发送多路载波，或者利用波分复用在一根光纤上同时发送多个频率的光信号。在接收端，解复用器应用一组过滤器，每个过滤器提取载波频率附近一个小的频率范围。图11-2所示为这种组织结构。

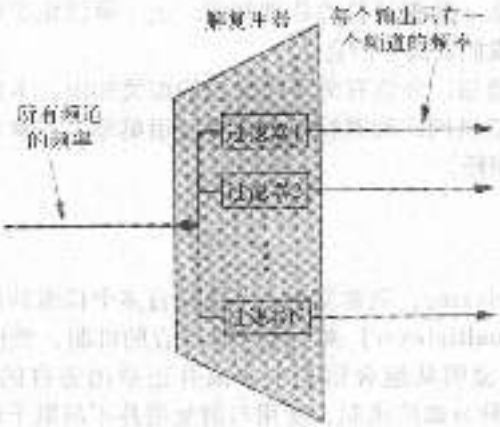


图11-2　基本FDM的概念示意图，每个过滤器选择一个频道而拒绝其他频率

这里有一个关键点：FDM中所用的过滤器仅仅检查频率，不影响信号的其他特征。如果一对相互通信的发送器和接收器分配了特定的载波频率，那么FDM机制就可以把这个频率的载波从其他载波中分离出来，且不会改变信号的特征。因此，第10章讨论的所有调制技术都可以用于任一个载波信号。

要点　因为不同频率的载波波形不会互相干扰，频分复用为每一对发送器和接收器就供了专有的通信信道，在这个信道上可以使用任何调制方案。

FDM最重要的优点在于：多对通信实体可同时使用一条通信介质。我们可以想象成FDM为每一对通信实体提供了专用传输通路，这个通路就像一个个单独的物理传输介质似的。图11-3说明了这个概念。

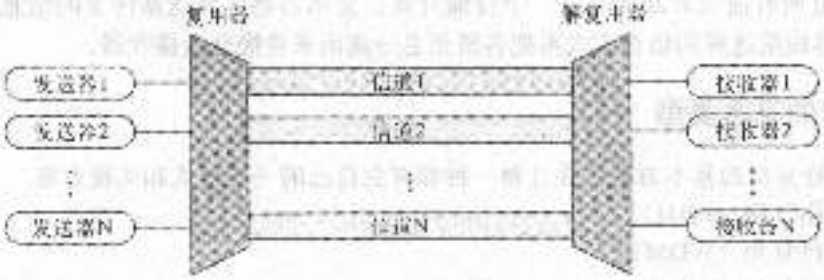


图11-3　当做提供一组独立信道的频分复用概念视图

当然，任何实际FDM系统都对可用于信道的频率范围有所限制。如果两个信道的频率任意接近，就会出现干扰。而且，接收复合信号的解复用硬件必须能够把信号分离成单独的载

波。就美国的无线电广播来说，由联邦通信委员会（Federal Communications Commission，FCC）管制着电台使用的频率，以确保载波频率之间有足够间隔。对于数据通信系统，设计者遵循同样的方法，选择一组有间隔的载波频率，频率之间的间隔被称为防护带（guard band）。

作为信道分配的一个例子，考虑图11-4所示的分配方案。图中的6个信道每个都分配了200kHz的带宽，信道之间的防护带具有20kHz带宽。

信 道	所用频率范围
1	100~300kHz
2	320~520kHz
3	540~740kHz
4	760~960kHz
5	980~1180kHz
6	1 200~1 400kHz

图11-4 相邻信道之间具有防护带的信道频率分配举例

在频率域上绘图表示时，防护带清晰可见。图11-5所示的就是图11-4分配方案的频域图表示。

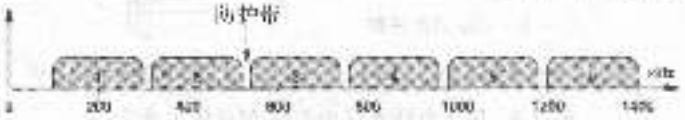


图11-5 图11-4所示信道分配的频域图，信道之间的防护带清晰可见

11.5 每个信道使用一个频率范围

既然一个载波只使用单一频率，那为什么上面的例子要分配一块块的频率范围呢？为了理解其动机，考虑FDM的一般特征。

- 历史悠久。FDM的提出早于数据通信——把电磁频段分成不同信道的思想源于早期的无线电实验。
- 广泛使用。FDM用于无线电和电视广播、有线电视以及AMPS蜂窝电话系统。
- 模拟。FDM复用与解复用硬件只接受和分发模拟信号。即使载波被调制成仅包含数字信息，FDM硬件也依然可以把载波作为模拟波来处理。
- 通用。由于FDM只过滤一定范围的频率而不检查信号的其他方面，因此它是通用的。

频分复用的模拟特征使得其具有一个缺点：对噪声和失真非常敏感[⊖]；同时又具有一个优点：拥有很好的灵活性。特别是，大多数FDM系统给每对发送器和接收器分配一段频率范围，以及选择如何使用相应频率范围的能力。系统使用一段频率范围上要有两个目的。

- 增加数据速率，
- 增加抗干扰能力。

为了增加整体数据速率，发送器把信道的频率范围划分为K个载波，并在每个载波上发送1/K数据。本质上，发送器是在已经分配好的信道上实施频分复用。一些系统使用术语子信道分配（subchannel allocation）就是指这种子划分。

为了增强抗干扰的能力，发送器使用一种称为扩频（spread spectrum）的技术。扩频技术具有多种形式可供使用，然而其基本思想就是把信道的频率范围划分为K个载波，在K个不同的信道上传输相同的数据，并允许接收器选择使用具有最少差错的到达数据的副本。在噪声有可能在特定时间里只干扰了某些频率的情况下[⊖]，这种方案工作得非常良好。

⊖ 使用FDM的数据通信系统经常要求使用同轴电缆以具有更好的抗噪声能力。

⊖ 即K个载波大部被干扰。——译者注

11.6　分级FDM

FDM中的一些灵活性来源于硬件的频率搬移能力。如果一组输入信号都使用0～4kHz的频率范围，复用硬件可以把第一组信号保持不变，把第二组信号映射到4～8kHz，把第三组映射到8～12kHz，依此类推。这种技术形成了FDM复用器分级的基础，每个复用器可以把输入映射到一个更大的连续的频率波段中。图11-6说明了分级FDM[⊖]的概念。

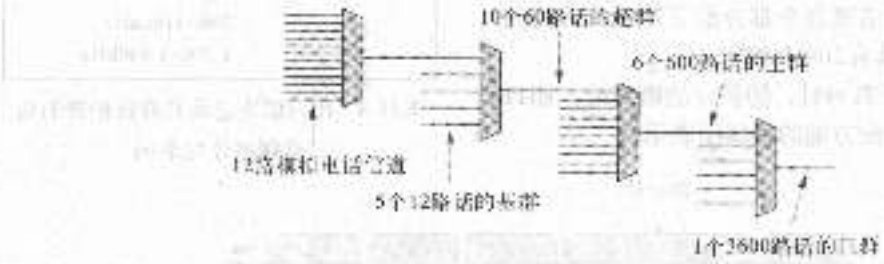


图11-6　用于电话系统中的FDM分级示意图

如图所示，基本输入由12路模拟电话信号组成，其中每个都占有频率段0～4kHz。在第一级上，这12路基频信号被复用成单路复合信号，称为一个群（group，俗称为基群）。使用的频率范围为0～48kHz。在下一级上，5个基群被复用成一个超群（super group），使用的频率范围为0～240kHz。依此类推，在最后一级上，由3600路电话信号被复用成一个巨群信号。

要点　频分复用系统可以按分级方式构建，其中的每一级接收来自上一级的若干个复合输出作为本级的多路输入。

11.7　波分多路复用

术语波分多路复用（Wavelength Division Multiplexing，WDM）是指应用于光纤[⊜]中的频分复用技术。这种复用的输入和输出是各种波长的光，使用希腊字母λ表示，非正式地称为色（colors）。为了理解复用和解复用技术是如何应用于光的，我们要回忆一下基础物理知识。当白光通过三棱镜时，它会将有色光谱展开。棱镜也可以以相反的模式操作，即如果一组有色光束中的每条光线都以恰当的角度直接进入棱镜，棱镜则会把光束复合成一束白光。最后回忆一下，人们所观察到的每一道有色光实际上是在一定波长范围内的复合光。

棱镜形成了光复用和解复用的基础，复用器接受不同波长的光束，用棱镜复合出一个单独的光束，解复用器则使用三棱镜分离出不同波长的光。图11-7说明了这个概念。



图11-7　波分复用技术中使用棱镜复合与分离不同波长的光示意图

要点　在光纤中采用频分复用时，使用棱镜来实现复合与分离各路不同波长的光。

⊖　实际上，还需要有额外的带宽来承载成帧码位。

⊜　有些资料中使用术语密集波分复用（Dense Wavelength Division Multiplexing，DWDM）是强调可以支持光的很多波长。

这种技术就称为波分多路复用。

11.8 时分多路复用

与FDM对应的另一种主要复用方法是时分多路复用（Time Division Multiplexing，TDM）。TDM没有FDM那么深奥，也不依赖于电磁能量的特殊性质，只不过是对信号时间域的操作。在时间上复用就意味着在一个时隙（time slot）内传输来自一个源的一个数据项，接着在下一个时隙内传输来自另一个源的一个数据项，依此类推。图11-8说明了这个概念。



图11-8 时分复用的概念示意图：来自多个源的数据项在共享传输介质上传输

11.9 同步TDM

时分多路复用是以多种形式呈现的一个宽泛概念，并被广泛应用于整个互特网中。因此，图11-8也仅仅是时分复用的一个概念视图而已，其细节可能因不同的形式而有所变化。例如，图中所显示的数据项是以轮流（round-robin）方式发送（即先发送从发送器1过来的数据项，接着是从发送器2过来的项，如此循环往复）。虽然有些TDM系统采用轮流的顺序，但有些则不用这种方式。

图11-8中的另一个细节也没有应用于所有类型的TDM，即图中表示出数据项之间存在微小的间隔。回忆一下第9章，如果通信系统使用同步传输，码元之间是没有间隔出现的。当TDM应用于同步网络时，数据项之间也不出现间隔，这样的系统才能被称为同步时分多路复用（synchronous time division multiplexing）。同步TDM系统采用轮流顺序去选择数据项。图11-9说明了具有4个发送器的同步TDM系统的工作情况。

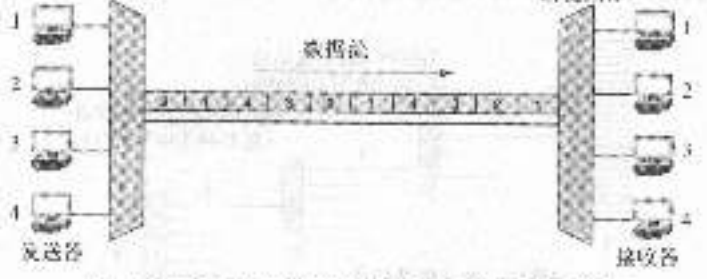


图11-9 具有4个发送器的同步时分复用系统示意图

11.10 电话系统中TDM的成帧技术

电话系统使用同步TDM实现在单个介质上复用来自多个电话呼叫的数字流。事实上，电话公司使用缩写名称TDM，是指用于多路数字电话业务的TDM特定形式。

电话系统标准的TDM包括一项有趣的技术，这个技术确保解复用器与复用器保持同步。为了理解需要同步的原因，我们来观察一下这样系统的工作情况：系统按一个时隙接着一个时隙地发送信息，中间没有任何输出指示来表明当前输出是来自于哪一个时隙。因为解复用器无法断定每个循环时隙从哪里开始，所以解复用器中用于码元定时的时钟稍微出现一点偏差，就会导致解复用器曲解码元流信息。

为了防止对码元流的曲解，电话系统中使用的TDM版本在输入中包含一个额外的成帧信道（framing channel）。成帧信道每一循环在流中插入一个码元，而不是插入一个完整的时隙。与其他信道一样，解复用器从成帧信道中提取数据，同时检查数据码位是否按0和1交替。它的基本思想是：如果发生差错导致解复用器丢失一个码位，成帧校验就有极大可能检测到这个差错并允许传输重新开始。图11-10说明了成帧位的使用情况。

概括如下：

数字电话系统中使用的同步TDM机制在每一循环的开始包含一个成帧位。成帧位序列保持1和0交替出现，这样解复用器就能保持同步又能检测到差错。

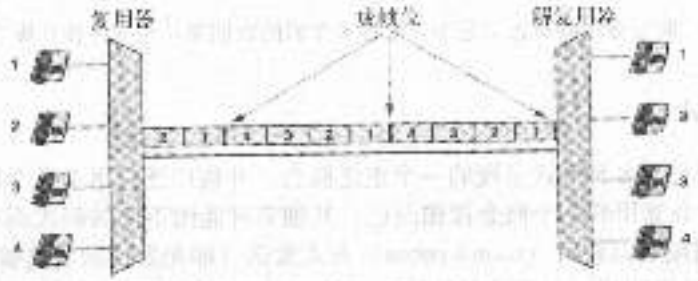


图11-10　电话系统中使用的同步TDM系统示意图，成帧位与字节每次的时隙循环

11.11　分级TDM

与FDM相似，TDM也可以安排成层级式结构。不同之处是TDM分级系统的每个后一级具有N倍码元速率，而FDM分级系统的每个后一级具有N倍频率。在数据中增加了额外的成帧位，这意味着层级系统的每个后一级的码元速率稍大于总合话音通信量。请比较一下图11-11中的分级TDM系统和图11-6中的分级FDM系统的例子。

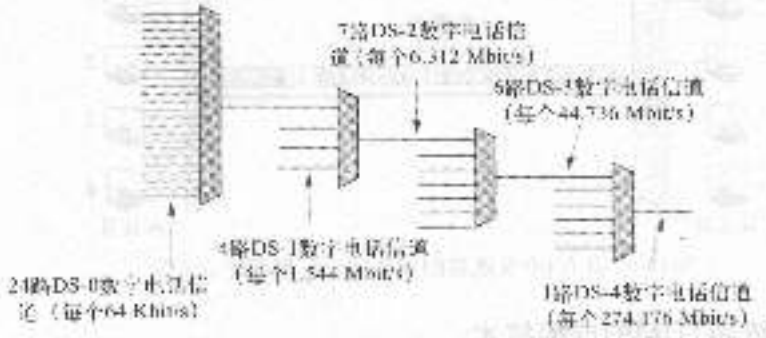


图11-11　电话系统中使用的分级TDM示意图

11.12 同步TDM的问题：空闲时隙

如果每个源产生统一形式的数据，则同步TDM工作得很好，以等于共享介质容量1/N的固定速率运行。例如，如果每个源对应于一路数字电话业务，则数据将以统一速率64Kbit/s到达。然而，正如第9章所指出的，有很多源是以突发形式产生数据的，而在前后突发之间则是空闲时间，这种情况下同步TDM系统就不能很好地工作。为了理解其中的原因，考虑图11-12的例子。

图中，左边的源各自随机地产生数据项。这样，如果某个源在轮到它发送信息的对应时隙却未产生数据项，则同步复用器就产生一个未填充的时隙。当然，在实际应用中，时隙不可能空闲因为下层系统必须连续传输数据，因此，时隙会被硬性地填充为一个值（例如0），并且会被设置一个额外的位以指示这个值是无效的。

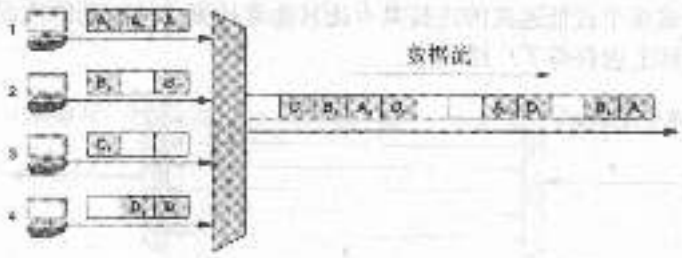


图11-12 信源未及时准备好发送数据项时，同步TDM系统保留时隙空闲的示意图

11.13 统计TDM

复用系统如何能更好地利用共享传输介质呢？一种被用来增加整体数据速率的技术称为统计时分多路复用（statistical time division multiplexing）或简称统计复用（statistical multiplexing）[⊖]。术语有点笨拙，但是这种技术却简单实用：以轮流方式选择数据项传输，跳过未准备好发送数据的源，而不去保留未填充的空闲时隙。通过消除这种未填充的时隙，统计TDM就可以使用更少的时间来发送等量数据。例如，图11-13说明了统计TDM系统如何用8个时隙发送了图11-12中用12个时隙发送的数据的示例。

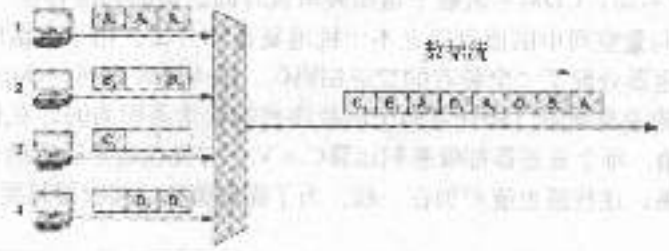


图11-13 统计TDM舍弃未填充时隙并以较少的时间发送等量数据的示意图

虽然应用统计复用避免了未填充时隙，但是统计复用却会招致额外的开销。为了理解原因，看看解复用的情况。在同步TDM系统中，解复用器每一次都知道第N个时隙对应于一个给定接收器。在统计复用系统中，在给定时隙中的数据则可以对应于任一个接收器。因此，除了数据，每个时隙必须包含数据发送给哪个接收器的标识。后续章节将讨论在分组交换网和因特网中用于统计复用的标识机制。

⊖ 有些文献使用术语异步时分多路复用（asynchronous time division multiplexing）。

11.14　逆转复用

有些情况下会出现有趣的复用逆转的做法：两点之间的连接由多条传输介质组成，但没有一条传输介质具有足够的码元速率。例如，在因特网的核心区域，服务提供商需要比可获得的传输介质更高的码元速率。为了解决这个问题，可以采用逆转方式来利用复用技术，即把高速数字输入展开在多条较低速度的线路上分路传输，在接收端再将分路传输的输出合并在一起。图11-14说明了这个概念。

实际上，逆转复用器不能仅通过反向连接几个传统复用器就可构建成功。相反，硬件的设计必须使得发送器和接收器就如何将到达的输入数据分配到较低速度的线路方面要达成一致。更重要的是，为了确保所有数据分发的顺序与到达的顺序一致，系统必须经过工程设计，以处理好其中一个或多个较低速度的连接具有比其他连接更长的时延的情况。尽管较为复杂，但逆转复用在因特网上也获得了广泛应用。



图11-14　逆转复用示意图：单个高速数字输入被分配到多个低速连接上，传输后再重新复合而形成输入的副本

11.15　码分多路复用

最后一种复用形式称为码分多路复用（Code Division Multiplexing，CDM），它被应用于部分蜂窝电话系统和一些卫星通信系统中，用于手机电话的具体的CDM版本，又被称为码分多址接入（Code Division Multi-Access，CDMA）技术。

与FDM和TDM不同，CDM不依赖于诸如频率或时间之类的物理特性，CDM依赖于有趣的数学思想：正交向量空间中的值可以互不干扰地复合和分离。用于电话网络的具体形式最容易理解。每个发送器分配了一个特有的二进制码C，称为码片序列（chip sequence）。码片序列都是经过选择的正交向量（即任意两个码片序列的标量乘积为0）。在任意时间点，发送器有一个值V_i要传输，每个发送器都做乘积运算$C_i \times V_i$，并发送这个运算结果。本质上，发送器同时传输这些结果，这些值也就相加在一起。为了提取值V_i，接收器只要用C_i乘以和值就可以了。

为了阐明这个概念，考虑一个例子。为了保持例子容易理解，我们将使用仅有两个码元长度的码片序列和4个码元长度的数据值。我们将码片序列看做向量，图11-15列举了这些值。

发送器	码片序列	数据值
A	1 0	1 0 1 0
B	1 1	0 1 1 0

图11-15　使用码分复用的简单例

第一步：用−1表示0，把二进制值转换为向量

$$C_1=(1,-1)\quad V_1=(1,-1,1,-1)\quad C_2=(1,1)\quad V_2=(-1,1,1,-1)$$

相乘运算$C_1 \times V_1$和$C_2 \times V_2$，得到：

$$((1,-1),(-1,1),(1,-1),(-1,1))\qquad((-1,-1),(1,1),(1,1),(-1,-1))$$

如果我们把结果值看做是要同时发送的信号强度序列，那么所得到的信号将是这两个信号之和，即

```
  1  -1  -1  1  1  -1   1  1
+ -1  -1   1  1  1   1  -1 -1
-------------------------------
  0   2   0  2  2   0  -2  0
```

接收器把接收序列当做向量处理，计算该向量与码片序列的乘积，把结果当做序列并把正值解释为二进制1，负值解释为二进制0，从而完成把结果序列转换成为二进制数。这样，1号接收器的计算是：

$$(1,-1)\cdot((0,-2),(0,2),(2,0),(-2,0))$$

得到：

$$((0+2),(0-2),(2+0),(-2+0))$$

把结果解释为序列乘积：

$$2\ -2\ 2\ -2$$

转换成二进制值：

$$1\ 0\ 1\ 0$$

显然，1010正是V_1的正确值。与此同时，2号接收器也会从同一个传输序列中提取到V_2的正确值。

CDM看上去几乎没有胜过TDM的优点。事实上，CDM甚至还有些不足，因为CDM要求一个大的码片序列，即使在一给定时间区间内只有少数发送器发送信息，这一情况也不例外。因此，如果利用率低的话，CDM还不如统计TDM工作得好。

CDM的优点在于它的可伸缩能力，以及它在高利用率网络中能提供较低的时延。为了理解低时延的重要性，考虑一个统计TDM系统。一旦一个发送器进行发送，在下一次轮到这个发送器发送前，TDM复用器允许其他N－1个发送器发送。因此，如果所有发送器都是活动的，那么对于一个特定的发送器来说，它的后续发送时延可能就较大。然而，在CDM系统中，一个发送器可以与其他发送器同时发送，这意味着时延较低。CDM对于电话业务特别有吸引力，因为传输的低时延对于高质量的语音传递是非常重要的要求。概括如下：

当网络处于高利用率时，CDM具有比统计TDM更低的时延。

11.16 本章小结

多路复用是数据通信中的基本概念。复用机制允许多对发送器和接收器在共享介质上通信。复用器在共享介质上发送来自于多个发送器的输入，解复用器分离并分发数据项到各个接收器。

多路复用有4种基本方法：频分、时分、波分以及码分。频分多路复用（FDM）允许在多个信道上同时通信，每个信道对应于电磁辐射的一个单独频段。波分多路复用（WDM）是一种特殊形式的频分复用——在光纤上发送不同频率（又叫波长）的光。

时分多路复用（TDM）在共享介质上一次发送一个数据项。同步TDM系统发送数据项时，各项之间没有间隔，通常采用轮流选择方式。在轮到该发送的发送器未准备好发送数据项时，统计TDM系统会跳过此发送器，从而避免空闲时隙。

码分多路复用（CDM）采用编码的数学组合，这允许多个发送器同时发送而不会互相干

扰。CDM的主要优点在于具有较低时延的可伸缩能力。

练习题

11.1　给出在非电子通信系统中采用复用技术的一个例子。
11.2　多路复用的4种基本类型是什么？
11.3　FDM如何使用电磁辐射？
11.4　什么是防护带？
11.5　FDM系统可能给每一信道分配一段频率范围。请问：为每个数据使用何种类型的调制时，使用一段频率范围是必要的吗？
11.6　说明如何使用一段频率范围才能增加数据速率。
11.7　请说明：在分级FDM系统中，如何把高容量信道划分为子信道。
11.8　在WDM系统中用来复合和分离不同波长光的关键机制是什么？
11.9　TDM系统要求采用轮流服务方式吗？
11.10　请解释：为什么在TDM系统中，成帧和同步是重要的？
11.11　在分级TDM系统中，给定等级的输出必须以什么位速率操作？（按照输入数目和输入位速率的方式来表达答案。）
11.12　假设N个用户竞争使用统计TDM系统，并且假设下层物理传输每秒可以发送K位，一个单独用户可以体验到的最高数据速率和最低数据速率分别是什么？
11.13　假设OC-12线路的费用是OC-48的1/20。ISP可以使用何种复用技术来降低以OC-48的速率发送数据的费用？请解释。
11.14　在因特网上进行搜索，找到在CDMA电话系统中使用的码片序列长度。
11.15　在4种基本的多路复用技术中，CDM总是最好的吗？请解释。

第12章 接入与互连技术

12.1 引言

前面的每一章都分别研究了数据通信的一个基本方面。上一章我们讨论了复用和分级复用的概念，并描述了电话公司用于数字电话的时分和频分复用方案。

这一章将研究用于因特网的两种工具，以此来结束对数据通信的讨论。首先，本章讨论接入技术（例如拨号、DSL以及电缆调制解调器），应用这些技术可以把个人住户和商业用户连接到因特网。接着，本章要讨论涉及应用于因特网核心部分的高容量数字线路。本章还将进一步讨论电话系统分级复用技术，并给出提供给商业用户和因特网服务提供商使用的公共承载线路的例子。本章的讨论重点在于这些技术的数据通信方面，主要涉及多路复用和数据速率这两方面的问题。

12.2 因特网接入技术：上行与下行

因特网接入技术（Internet access technology）是指连接因特网用户（subscriber）（一般是私人住户和商业机构）和因特网服务提供商（Internet Service Provider，ISP）（例如电话公司或电缆公司）的数据通信系统。为了弄懂如何设计接入技术，我们必须清楚其中的一点，那就是大多数因特网用户都是按非对称（asymmetric）模式使用网络的，即典型的居民用户从因特网接收的数据要多于他们发送出去的数据。例如，为了浏览一个网页，浏览器发送一个只包含几个字节的URL，而Web服务器响应回来的内容可能是包含几千字节的文本或者是一个包含好几万字节的图片。运行Web服务器的商业用户可能具有相反的流量模式——商业用户发送的数据要多于接收的数据。

要点 因为典型的居民用户接收的信息远多于发送的信息，所以就把因特网接入技术设计成一个方向上的传输量要远大于另一个方向上的量。

网络行业使用术语下行（downstream）表示数据从因特网中的业务提供商传输到用户，并用术语上行（upstream）表示数据从用户传输到服务提供商。图12-1所示为这两个术语的定义。

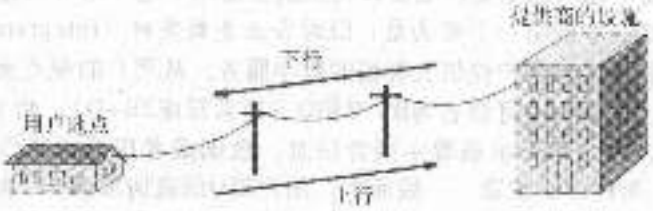


图12-1 接入技术中上行与下行方向的定义

12.3 窄带与宽带接入技术

有多种技术可用于提供因特网接入服务。根据这些技术提供的数据速率，接入技术可以

划分为两种大类：

- 窄带（narrowband）。
- 宽带（broadband）。

虽然第6章已解释了传输介质的带宽与数据速率的区别，但在接入网中使用的术语并不关心这种区别。相反，联网行业通常使用术语网络带宽（network bandwidth）来指代数据速率。因此，术语窄带（narrowband）和宽带（broadband）的叫法只是反映了一种行业习惯。

12.3.1 窄带技术

窄带（narrowband）通常是指传递数据的速率低于128Kbit/s的技术。例如，使用拨号连接时，即使使用最智能的modem技术并且电话线路的噪声最低，它能获得的最大数据速率也只有56Kbit/s。因此，拨号连接被归类为窄带技术。类似地，使用modem的模拟线路、较低速率的数字线路以及电话公司提供的某些数据服务（例如ISDN）都是窄带的。图12-2归纳了主要的窄带接入技术。

窄　带
拨号电话连接
使用供音频调制的线路半双工
2B(+D)数字电路
ISDN及其他电信数据服务

图12-2　主要的四种网窄带接入技术

12.3.2 宽带技术

术语宽带（broadband）一般是指提供高速数据速率的技术，不过宽带与窄带之间的确切界限比较模糊。许多专业人员建议宽带技术的传输速率应该大于1Mbit/s。不过，像电话公司这样的提供商在接入服务的速率高于拨号连接的速率时，他们就会使用术语宽带来宣传他们的此项服务。因此，虽然ISDN提供的接入速率只有128Kbit/s，有时电话公司也会声称他们的ISDN服务是宽带的。图12-3归纳了主要的宽带接入技术。

宽　带
DSL技术
电缆调制解调器技术
无线接入技术
T1速率或更高速率的数据线路

图12-3　主要的四种网宽带接入技术

12.4 本地环路及ISDN

术语本地用户线路（local subscriber line）或本地环路（local loop）是指电话公司的交换局（Central Office，CO）与用户驻地之间的物理连接。为了理解如何使用本地环路，重要的是要把本地环路与电话系统的其余部分独立开来看待。虽然整个电话系统的工程设计为每个拨号业务提供了4kHz的带宽，但本地环路部分由双绞线组成并通常具有高得多的带宽。尤其是接近CO的用户，本地环路能够应付1MHz以上的频率范围。

随着数据联网变得越来越重要，电话公司探索了使用本地环路可以提供更高速的数据通信的各种方法。电话公司的第一个努力是：以综合业务数字网（Integrated Services Digital Network，ISDN）的名义为用户提供大规模的数字服务。从用户的观点来看，ISDN提供了3个独立的数字信道，这些信道可命名为B、B和D（通常写成2B+D）。两个B信道，每个都以64Kbit/s速度运作，设计用于承载数字话音信息、数据或者压缩视频信息；一个D信道以16Kbit/s速度运作，用作控制信道。一般而言，用户用D信道请求服务，随后此服务通过B信道去提供（例如使用数字话音的电话业务）。两个B信道可以联合或合并（bonded）提供有效数据速率为128Kbit/s的单信道服务。当ISDN被首次提出时，128Kbit/s看似远快于拨号调制解调器，但是现在，更新的本地环路技术以更低的代价却提供了更高的数据速率，使得ISDN被论落为很少应用的特殊技术。

12.5 数字用户线技术

数字用户线（Digital Subscriber Line，DSL）是在本地环路中用来提供高速数据通信服务的主要技术之一。表12-4列举了DSL技术的各种类型。由于这些技术名称的差别仅限于第一个字母，因此可以用缩写xDSL来统称这个技术的集合。

名　字	全　称	一般用途
ADSL	不对称DSL	居民用户
ADSL2	第2类不对称DSL	大约比普通的ADSL快3倍
SDSL	对称DSL	输出数字的商业用户
HDSL	高位率DSL	距离大约5英里（1mile=1609.344m）的商业用户
VDSL	甚高位率的DSL	速率52Mbit/s的建设类型

表12.4 被统称为xDSL的DSL主要类型

ADSL是最广泛使用也是居民使用最多的类型。ADSL使用频分复用把本地环路的带宽划分为3个区域。其中一个区域对应到传统的模拟电话服务，这种服务在工业界被称为普通老式电话业务（Plain Old Telephone Service，POTS），其他两个区域提供数据通信服务。

要点 因为ADSL使用频分复用技术，所以ADSL和传统的模拟电话服务（POTS）可以同时使用同一根导线。

图12-5所示为ADSL的带宽划分情况。

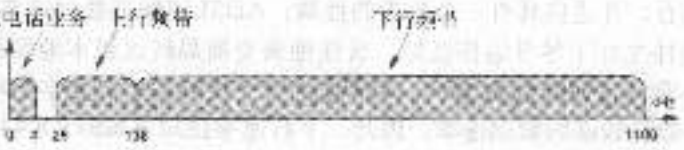


图12-5 本地环路带宽被划分的示意图

在图中，x轴是非线性的，如果是线性的，则为POTS保留的4kHz区域将会看不见，POTS与上行区域之间的22kHz防护带也不会显示出来。

12.6 本地环路特征及适配

ADSL技术是复杂的，因为没有任何两个本地环路具有完全相同的电气特性。事实上，本地环路承载信号的能力取决于距离、所使用导线的直径以及电磁干扰的水平。例如，考虑分别居住在一个城镇内不同区域的两个用户。如果通往第一个用户的电话线经过一个商业广播电台附近，那么电台的信号就会在电台使用的频率点上引起对线路的干扰。如果第二个用户不居住在同一个电台的附近，那么电台使用的频率对于这个用户线上的数据传输就没有什么影响。然而，第二个用户也可能在另一个频率上遭受干扰。因此，ADSL设计者不可能找到一个特别的载波频率集合或调制技术，能使所有本地环路中都不会收到任何干扰。

为了适应本地环路特征中的差别，ADSL是可以自动适应的（adaptive），也就是说，当一对ADSL调制解调器通电后，它们探测彼此之间的线路以发现线路特征，接着协商一致使用对于当前线路最优的技术进行通信。尤其是ADSL采用了一个称为离散多音频调制（Discrete Multi Tone modulation，DMT）的方案，这个方案结合了频分复用和逆转复用技术。

在DMT中采用的频分复用是这样实现的：将线路带宽划分为286个分离频率段，称为子信

道[⊖]（subchannels），其中255个子信道分配用于下行数据传输，31个分配用于上行数据传输；两个上行信道保留用于控制信息。在概念上，每个子信道都有各自的“调制解调器”，而且它们都有各自的调制载波，载波之间间隔4.1325kHz以保持信号不会互相干扰。此外，为了保证子信道中的信号传输不会干扰模拟电话信号，ADSL避免使用频率低于26kHz的带宽。ADSL启动后，两端的调制解调器探测可用频率，从而确定哪些频率可以工作良好，哪些频率会受到干扰。除了选择频率，两端也评测每个频率上的信号质量，并根据信号质量来选择调制方案。如果某个特定频率具有高信噪比，ADSL就选择每波特可以编码多个码元的调制方案；如果在给定频率上的信号质量较低，ADSL就选择每波特编码较少码元的调制方案。我们可以概括如下：

> 因为本地环路的电气特性变化各异，ADSL采用了自适应技术，即一对调制解调器先探测彼此之间连接线路上的许多频率，然后选择在此线路上能产生最优传输质量的频率和相应的调制技术。

12.7　ADSL的数据速率

ADSL如何能实现较快的数据传输速度呢？ADSL在短距离的本地环路上可以达到8.448Mbit/s的下行速率，而上行速率为640Kbit/s。因为强制的网络控制信道要求64Kbit/s，用户数据可获得的有效上行速率是576Kbit/s。在最好的条件下，ADSL2可以以接近20Mbit/s的速度下载。

从用户观点上看，自适应具有一个有趣的性质：ADSL不保证数据速率。事实上，ADSL只能保证在线路条件允许下尽量运作良好。居住地离交换局较远或本地环路要通过干扰源附近的用户，往往体验到较低的数据速率；而居住地离交换局较近以及本地环路不经过干扰源附近的用户，则体验到较高的数据速率。因此，下行速率在32 Kbit/s～8.448Mbit/s之间变化，而上行速率在32 Kbit/s～640Kbit/s之间变化。

ADSL的速率仅仅应用于用户到电话交换局之间的本地环路连接，理解这一点非常重要。有许多其他因素还会影响用户体验到的整体数据速率。例如，当一个用户连接到Web服务器时，可以限制有效数据速率的因素有服务器的速度或当前负荷，用于连接服务器站点到因特网的接入技术，或者用户的交换局与处理服务器的提供商之间的即时网络状况。

12.8　ADSL安装和分离器

虽然传统的模拟电话在低于4kHz的频段上运作，但是举起一个话筒就可能产生干扰DSL信号的噪声。为了提供完全的隔离，ADSL使用一个称为分离器（splitter）的FDM设备，这个设备对线路的带宽进行划分，将低频部分的通过作为一个输出，而将高频部分的通过作为另一个输出。有趣的是，分离器是无源（passive）的，这意味着它不需要电源。分离器通常安装在本地环路进入居民用户或商业用户的入口，它的一端连接POTS线路而另一端连接ADSL调制解调器。图12-6所示为这种连接关系。

目前已经流行着一种有趣的ADSL配线变型，有时称它为DSL lite。这种变型方法不要求在连到话机的入线上安装分离器，而是将房屋内现有的电话配线直接用于DSL，并且在每个

⊖ 术语子信道（subchannels）源于一些版本的DSL，它把带宽划分成1.544Mbit/s的信道，这种信道带宽与本章后面描述的T1线路相对应。

话机连接与入户配线之间安装一个分离器。这种方法的优点是：用户把分离器插到墙上的插座，再把电话插在分离器上，就可以完成DSL的安装。

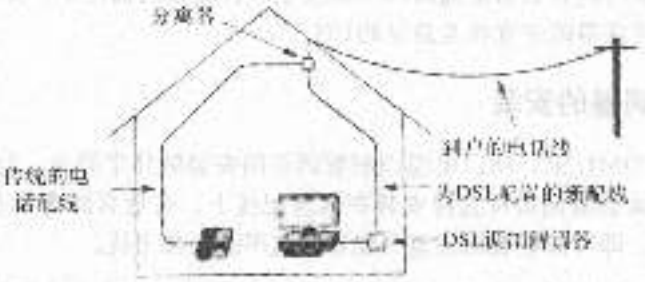


图12-6 用于ADSL的分离器及其配线示意图

12.9 电缆调制解调器技术

虽然像ADSL这样的技术提供了以前从不敢想象的高速率，但电话本地环路配线具有内在的局限性，其主要问题还是在于双绞线的电磁特性。缺少屏蔽使得导线对干扰非常敏感，从而导致一些用户的数据传输性能大大降低。随着对更高速率的需求增长，寻找可替换的配线方案变得非常重要，因此，多种用于本地环路的无线和有线技术陆续被开发出来。

另一种接入技术就是使用已经广泛部署的有线电视[1]（cable television）配线，这种技术一经推出就极具吸引力。在缆线系统中使用的介质是同轴电缆，它具有高的带宽，抗电磁干扰能力比双绞线强。而且，有线电视系统采用频分复用(FDM)技术，实现同时传递很多的娱乐频道。

人们可能会做这样的假设：利用同轴电缆具有很多频道可供使用，所以CATV提供商就可以使用一个个单独的信道给每一个用户传递数字信息。也就是说，CATV提供商配置一对电缆调制解调器（cable modems），一个位于CATV中心，而另一个位于用户端，通过这对电缆调制解调器使用电缆线的一个特定信道（即载波频率）进行通信，并将电视信号一起复用到电缆线上。

尽管CATV系统有很大的带宽可供使用，但这些带宽仍然不能满足使用频分复用方案为每个用户提供一个独立信道的需求。为了理解其中的原因，试想一下高密度的大都市区域，单个CATV提供商就可能拥有几百万用户，因此，每个用户使用单独信道的这个方案并不具有可适应能力。

为了解决这个问题，CATV系统组合了FDM和统计复用技术，为一组用户（典型的，一群邻居）的数据通信分配一个信道。每个用户分配一个唯一的地址，信道上发送的每条消息都包含所发往的地址信息。用户的调制解调器监听被分配的频率，然而在接收消息前，调制解调器验证消息中的地址与分配给用户的地址是否相匹配。

12.10 电缆调制解调器的速率

电缆调制解调器能实现多快的传输速度呢？理论上，电缆系统可以支持下行52Mbit/s以及上行512Kbit/s的数据速率，但实际的速率远远低于理论值。首先，电缆调制解调器的数据速

[1] 有线电视，正式全称是公用天线电视（Community Antenna Television，CATV），它某个同轴电缆并使用FDM向用户提供广播电视信号。

率仅属于本地电缆中心与用户端之间的通信。其次，N个用户形成的集体共享一个带宽，其中集体的大小由电缆提供商控制，从用户的观点来看，与其他用户共享带宽可能是一个不利的方面，因为每个用户可获得的有效带宽随时间不断改变。在最差的情况下，如果N个用户共享单个频率，则单个用户可获得的带宽将是总量的1/N。

12.11　电缆调制解调器的安装

由为电缆系统使用FDM技术，所以电缆调制解调器的安装就非常简单，与xDSL技术要求使用分离器不同，电缆调制解调器可直接安装在电缆配线上，在现有的有线电视盒和电缆调制解调器中的FDM硬件，即可保证数据信道与娱乐频道不会互相干扰。

要点　因为电缆系统使用频分复用，所以电缆调制解调器不需要分离器就可以直接安装在现有的电缆配线上。

12.12　光纤与同轴电缆混合使用

提供高速数据通信的最有前途的技术之一，就是光纤同轴电缆混用（Hybrid Fiber Coaxial，HFC）。顾名思义，光纤同轴电缆混合系统就是组合运用光纤与同轴电缆技术，其中光纤用于中心设施，而同轴电缆则用于连接单个用户。本质上，HFC系统是分级的，要求最高带宽的网络部分使用光纤，但是可容忍较低速率的网络部分使用同轴电缆。为了实现这样一个系统，在每个小区，提供商都放置了可以在光纤和同轴电缆之间进行转换的设备。每个设备与提供商中心机房之间通过光纤连接，与小区中各户之间则通过同轴电缆连接。图12-7说明了这种架构。

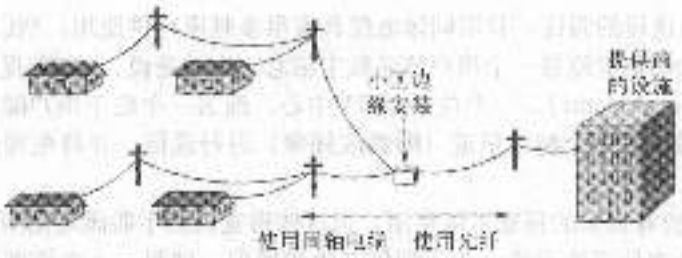


图12-7　光纤同轴电缆混合系统的示意图

电缆业界使用术语干线（trunk）来指电缆中心与邻居区域间高容量的连接，使用术语馈送线（feeder circuit）表示到个体用户的连接。干线连接可以远至15km，而馈送线通常少于1km。

12.13　采用光纤的接入技术

电缆公司已经提出多种技术，有的技术在混合系统中采用光纤，而有的技术在通往用户的全程部署光纤。图12-8归纳了其中关键技术的名字。

名字	全　称
FTTC	光纤到区边
FTTB	光纤到楼宇
FTTH	光纤到家庭
FTTP	光纤到户端

图12-8　使用光纤的各种接入技术名称

光纤到区边（Fiber To The Curb，FTTC）。顾名思义，FTTC与HFC类似，都是在高容量的干线上使用光纤。这个技术的思想是先将光纤接近用户端（进入

小区的边界外），然后在馈送线上使用铜导线。FTTC与HFC也有不同，因为FTTC在每条馈送线上使用两种介质以允许电缆系统提供附加的服务（例如话音）。这种技术正在一些区域开始部署，特别是美国和加拿大。

光纤到楼宇（Fiber To The Building，FTTB）。一个基本问题涉及商业用户需要的带宽和使用铜导线（甚至是同轴电缆）的接入技术是否能够满足他们的需要。FTTB是一种使用光纤以允许高上行数据速率的技术。

光纤到家庭（Fiber To The Home，FTTH）。这是与FTTB功能相当的另一种技术。FTTH是一种使用光纤为用户提供更高下行数据速率的接入技术。虽然它也可提供更高的上行数据速率，但其重点在于提供更多的娱乐和视频信道。

光纤到户端（Fiber To The Premises，FTTP）。这是一个通称，它包含FTTB和FTTH两种情况。

12.14 头端与尾端调制解调器技术

接入技术要求使用一对调制解调器，一个用于用户，而另一个用于提供商。业界使用术语头端调制解调器（head-end modem）来指用在交换局的调制解调器，使用术语尾端调制解调器（tail-end modem）来指用在用户处所的调制解调器。

头端调制解调器不是独立设备，而是由大量调制解调器集合构建起来的一个单元，可以对它进行统一的配置、监视和控制。由电缆提供商使用的一组头端调制解调器称为电缆调制解调器终端系统（Cable Modem Termination System，CMTS）。称为有线电视数据服务接口规范（Data Over Cable System Interface Specifications，DOCSIS）的一组工业标准，规定了被发送数据的格式以及用于点播服务（例如预付费电影）的报文格式。

12.15 无线接入技术

虽然诸如ADSL和HFC这样的技术可以为大多数用户提供数据服务，但是它们并不足以应付所有的情况。主要的问题来自于边远地区，例如，距最近的城市都有很远距离的农场或遥远的乡村。如果在这样的区域使用双绞线提供电话服务，所需双绞线的长度将超过类似于ADSL这样的技术所能提供的最大距离。此外，边远地区几乎没有有线电视服务。

即使在市郊，类似ADSL这样的技术在可使用的线路类型上也可能存在技术限制。例如，在包含加感线圈（loading coils）、桥接抽头（bridge taps）或中继器（repeaters）的电话线路上，使用高频率的可能性就很小。因此，即使在本地环路技术可为大多数用户提供服务的区域，本地环路技术也不一定在所有线路上都可以使用。

为了应对这些特殊情况，工程人员已经探索了多种无线接入技术。图12-9列举了几个例子，第16章将讨论这几种技术。

技　术	描　述
3G业务	基于数据的第三代蜂窝电话服务（例如EVDO）
WiMAX	使用无线射频高达155Mbit/s速率的无线接入技术
卫星	一些商业供应商使用卫星提供数据服务

图12-9 无线接入技术举例

12.16　因特网核心区的高容量连接

联网专家说，接入技术解决的是最后一英里问题（last mile problem），此处所谓“最后一英里”，即是指到典型的居民用户或小型商业用户的连接。接入技术为居民用户或小型商业用户（业界使用术语小型办公、家庭办公（Small Office Home Office，SOHO））提供了充足的带宽容量，而到大型商业机构的连接或提供商之间的连接则要求更多的带宽。为了与互联网边缘的连接相区分，专家使用术语核心（core）来表示这种连接，同时使用术语核心技术（core technologies）来表示用于核心区域连接的高速技术。

为了理解核心区所需要的数据速率，考虑一个具有5 000客户的提供商。假设提供商使用可为每个客户提供高达2Mbit/s带宽的接入技术。如果所有用户都试图同时下载数据会发生什么？图12-10表示从因特网到提供商的汇聚流量。

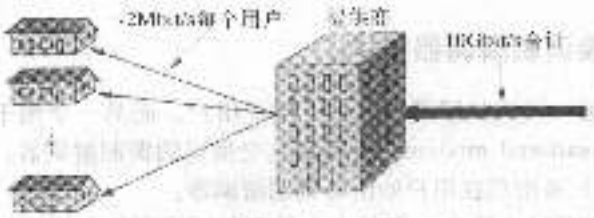


图12-10　从因特网到提供商的汇聚流量：假设提供商有5 000用户同时以2Mbit/s速率下载

问题出现了：提供商可以使用什么技术以10Gbit/s的速率远距离传输数据？答案是从电话公司租用点对点数字线路（point-to-point digital circuit）。虽然它的最初设计用于电话系统内部，但高容量数据线路可以通过月结租赁获得，而且可用于传输数据。因为电话公司有权穿越市政街道安装电线，线路也可以在城市之间延伸，越过城市，或者从一城市的某处到另一城市的某处，收取的费用取决于线路的数据速率和所跨越的距离。

概括如下：

> 从公用电话服务商那里租用数字线路，形成远距离数据通信的基本的建模块，费用取决于线路容量和距离。

12.17　线路终端、DSU/CSU及NIU

为了使用租用的数字线路，提供商必须同意遵守电话系统的规则，包括遵守设计用于传输数字话音的相关标准。遵守数字信息的标准看似微不足道，因为计算机也同样是数字化的，但是由于计算机工业和电话工业是独立发展的，所以电话系统数字线路的标准不同于计算机工业中使用的标准。因此，需要有一个特殊的硬件模块使计算机能与电话公司提供的数字线路进行接口，这个接口设备称为数据服务单元/信道服务单元（Data Service Unit/Channel Service Unit，DSU/CSU），它包括两个功能部分，通常将它们组合在单个机盒里面。DSU/CSU设备的CSU部分处理线路终接和诊断，例如，CSU包含可以测试线路是否已经断开的诊断电路，也包含环路测试设施，可以允许CSU把从线路接收到的所有数据进行复制，并传回给发送器而不作进一步的处理。

CSU提供一个出乎计算机工程师意料的服务——禁止过多的连续“1”码元。需要阻止过多连续“1”的这个做法，是出于对所用电信号方面的考虑。因为电话公司起初设计他们的数字线路是在铜导线上工作的，工程人员担心太多连续“1”码元会导致在导线上出现过多的电

流。为了避免这个问题，CSU可以采用能保证电流平衡的编码技术（例如差分编码），或者使用位插入（bit stuffing）技术。

DSU/CSU设备的DSU部分处理数据，它对载波线路上所用的数字格式与客户计算机设备要求的数字格式进行转换。用于计算机侧的接口标准取决于线路操作的数据速率。如果数据速率低于56Kbit/s，计算机可以使用RS-232。对高于56Kbit/s的速率，则必须使用支持更高速度的接口硬件（例如支持RS-449或V.35标准的硬件）。

电话公司提供一个附加的设备模块，称为网络接口单元（Network Interface Unit，NIU），它在电话公司拥有的设备与用户提供的设备之间形成一个边界。电话公司将这个边界称为分界（demarc）。

数字线路两端都需要有一个称为DSU/CSU的设备。DSU/CSU设备实现电话公司所使用的数字表示与计算机界所使用的数字表示之间的转换。

12.18 数字线路的电话标准

从电话公司租用的数字线路遵循电话公司用于传输数字电话业务的数字传输标准。在美国，数字电话线路标准的命名包含一个字母T，其后是一个数字，工程师统称这些标准为T系列标准（T-series standards）。其中最流行的一个标准称为T1，许多小型商业机构都使用T1线路来发送数据。

遗憾的是，T标准并不是通用的。日本采用了T系列标准的修改版本，而欧洲选择了另一种稍微不同的方案。因为欧洲标准使用字母E开头，所以很容易被区分开。图12-11列举几种数字线路标准的数据速率。

名 字	位速率	话音线路数	归属地
基本速率	0.064 Mbit/s	1	
T1	1.544 Mbit/s	24	北美
T2	6.312 Mbit/s	96	北美
T3	44.736 Mbit/s	672	北美
E1	2.048 Mbit/s	30	欧洲
E2	8.448 Mbit/s	120	欧洲
E3	34.368 Mbit/s	480	欧洲

图12-11 数字线路及其容量举例

12.19 DS术语及数据速率

回顾第11章所述，电话公司采用分级复用技术把多路话音业务组合到单条数字线路上传输。电话公司选择了T标准的数据速率，以使每条线路可以处理多路话音业务。重要的是要注意到，线路的容量并不是随标准系列编号数字的增长而呈线性增长的。例如，T3标准定义的线路具有远高于T1标准3倍的容量。最后，应该注意到电话公司也租用容量低于图中所列标准的线路，此类线路被称为部分T1电路（fractional T1 circuits）。

为了技术上的准确性，我们必须区分开这两个标准，即用于定义下层载波系统的T系列标准和用于规定如何在单个连接上复用多路电话业务的标准。后者被称为数字信号等级标准（digital signal level standards）或称DS标准（DS standards）。名字是以DS开头的，后面跟随数字，与T系列标准相似。例如，DS1是指可在单条线路上复用24路电话业务的服务，而T1也

是以这样的方式表示了一个特定的标准。因为DS1定义了有效的数据速率，因此在技术上说"线路以DS1速度运行"要比说"T1速度"更为准确。实际上，很少有人会用心去区分T1和DS1，所以你可能还会听到一些人说"T1速度"。

12.20　最高容量线路

电话公司使用术语干线（trunk）来指高容量线路，并为数字干线线路制定了一系列标准。这些标准称为同步传输信号（Synchronous Transport Signal，STS）标准，规定了高速连接的细节。图12-12归纳了与不同STS标准相关的数据速率。在表中所有数据速率的单位都以Mbit/s表示，这样便于相互比较。我们应该注意到，STS-24及其以上标准的数据速率都大于1Gbit/s。

铜导线名	光纤名	位速率	语音线路数
STS-1	OC-1	51.840 Mbit/s	810
STS-3	OC-3	155.520 Mbit/s	2 430
STS-12	OC-12	622.080 Mbit/s	9 720
STS-24	OC-24	1 244.160 Mbit/s	19 440
STS-48	OC-48	2 488.320 Mbit/s	38 880
STS-192	OC-192	9 953.280 Mbit/s	155 520

图12-12　符合STS分级标准的数字线路的数据速率

12.21　光载波标准

除了STS标准，电话公司还定义了同等的光载波（Optical Carrier，OC）标准集合。图12-12给出了铜标准以及对应的光标准。为了准确，我们应该观察STS与OC术语的区别，即STS标准针对用于数字线路接口（即基于铜导线）的电信号，而OC标准针对通过光纤传播的光信号。当与其他网络术语一起使用时，很少有专业人士会去区分这种差别。因此，人们常常听到网络专家用OC-3来指以155Mbit/s速率操作的数字线路，而不管线路使用的是铜导线还是光纤。

12.22　C后缀

上述的同步传输信号和光载波术语具有一个图12-12未说明的附带特点：可选的后缀字母C代表级联（concatenated）的意思。后缀的存在表示了线路不采用逆转复用。也就是说，一条OC-3线路可以由3条OC-1线路组成，其中每条OC-1都以51.840Mbit/s的速率运作。或者是由单条以155.520Mbit/s速率运作的OC-3C（STS-3C）线路组成。

全速运作的单条线路会比多条较低速率运作的线路更好吗？答案取决于如何使用线路。一般而言，全容量运作的单条线路提供更多的灵活性并且不需要逆转复用设备。更重要的是，数据网络与话音网络不同。在话音系统中，高容量线路是一种用来聚合更小话音流的方法。然而，在数据网络中，存在单个数据通信流。因此，如果可以选择，大多数网络设计者更愿意选择OC-3C线路而不是OC-3线路。

12.23　同步光网络

除了上述的STS和OC标准外，电话公司还定义了一大类数字传输标准。在北美，这类标准使用术语同步光网络（Synchronous Optical NETwork，SONET）来称呼，而在欧洲则称它

们为同步数字体系（Synchronous Digital Hierarchy，SDH）。SONET规定传输标准的细节，例如数据如何成帧，如何将较低容量的线路复用成为一个高容量线路，以及同步时钟信息如何与数据一起发送等。因为运营商广泛使用SONET，所以当有人要租用STS-1线路时，运营商可能会要求他们在线路上使用SONET编码。例如，图12-13表示出在STS-1线路上使用的SONET帧格式。

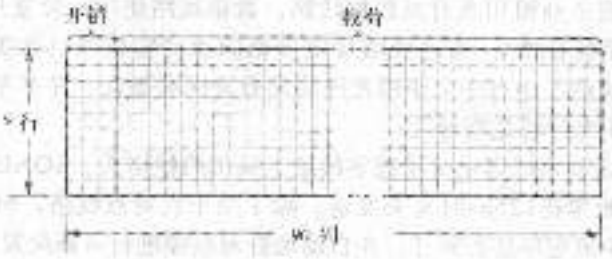


图12-13 在STS-1线路上使用的SONET帧格式示意图

每个帧的长度是810个字节，根据SONET的有关规范，帧中的字节被划分为9行，每行具有90列。有趣的是，SONET帧长取决于下层线路的位速率。然而，当用于STS-3线路时，SONET帧包含2430字节。这个数目是如何增大起来的呢？为了理解这个差别，回顾一下数字电话实行每秒8 000次PCM采样，这意味着每125μs采样一次。SONET使用这个时间来定义帧长度。以STS-151.840Mbit/s的传输速率，125μs确切传输6480码位，这意味着一个帧包含810字节。类似地，以STS-3的速率，125μs可以传输2430字节。帧长度由线路位速率决定的主要优点是同步复用变得非常容易——当把3个STS-1 SONET流合并成一个STS-3 SONET流时，保持同步非常简单易行。

虽然大多数数据网络都是在单条点对点线路上使用SONET作为编码方案，但该标准却提供了更多的可能性。特别是，使用SONET技术有可能构建出高容量的计数器旋转环网络。这种网络可以应付单节点故障。在环上的每一站点都使用一个称为分插复用器（add/drop mux）的设备。除了沿着环传递接收数据，分插复用器还可配置成从本地线路接收额外数据，并将其插入到环上传递的帧中或者提取数据并分发到本地计算机。如果环断裂了，硬件就会检测到丢失了帧信息并使计数器旋转环重新连接起来。

概括如下：

> 虽然SONET标准定义了一种能用来构建高容量环网的技术，并且在构成环的光纤上能复用多条数据线路，但是大多数据网络只是使用SONET来定义在租用电路上的成帧与编码。

12.24 本章小结

接入技术为个体住户或小型商业用户提供到因特网的连接。有多种接入技术可供选择，包括拨号电话连接、无线（使用射频或卫星）和有线技术。目前两种流行的接入技术是数字用户线（DSL）和电缆调制解调器。DSL所用介质为连接电话公司交换局与用户的本地环路，它采用FDM技术，允许同时处理本地环路上的数字通信和传统模拟话音业务。电缆调制解调器采用FDM技术在承载娱乐频道的同轴电缆系统上复用数字通信。使用电缆调制解调器技术时，邻近的电缆调制解调器采用统计复用来共享单个数据通信信道。

像光纤同轴电缆混合网（HFC）和光纤到区边（FTTC）这样的技术，都使用光纤来传输数据到每个社区，而用同轴电缆到达每个个体用户。使用光纤提供更高数据速率联网服务到个体住所的未来技术，也已经被提出来了。

虽然接入技术对于个体住户或小型商业用户具有足够带宽，但是这些技术并不能为因特网核心的使用提供足够容量。为了能跨越远距离获得最高的数据速率，服务提供商和大型的商业用户从公用电话服务商租用点对点数据线路。数据线路使用时分复用标准(在美国是T系列标准，在欧洲是E系统标准)。高速线路规定采用同步传输信号（北美）或同步数字体系(欧洲)。对应于高速线路，还有一个使用光纤的光载波标准集合，许多专家使用OC标准来命名，而不管线路是使用光纤还是铜导线。

称为SONET的电话公司标准定义了数字线路上采用的帧格式。SONET帧的大小取决于线路的位速率；一个帧通常花125μs时间来发送。除了用于点对点线路，SONET也可配置成一个环，允许环上的硬件测定环是否断开，并自动地针对故障进行重新配置。

练习题

12.1　什么是接入技术？

12.2　服务提供商为什么要区分上行与下行通信？

12.3　举出几个窄带与宽带接入技术的例子。

12.4　电话公司曾经提倡ISDN作为高速接入技术。为什么ISDN的应用会被否定？

12.5　如果客户试图传输更多他们发送的数据，那种形式的DSL更合适？

12.6　ADSL使用哪种类型的复用技术？

12.7　住在同一条街上的两个邻居，都使用ADSL服务，不过测量结果表明一个用户可以以大约1.5Mbit/s的速率下载，而另一个却可以以2Mbit/s的速率下载，请解释原因。

12.8　为什么分离器需要与DSL一起使用？

12.9　如果你可以在DSL和电缆调制解调器之间选择，哪个可以提供最高的潜在数据速率？

12.10　为什么服务提供商选择光纤同轴电缆混合网而不是光纤到户？

12.11　头端调制解调器应该安置在何处？尾端调制解调器呢？

12.12　与卫星相比，WiMax接入技术有什么优点？卫星的优点有哪些？

12.13　如果你租用了一条T1线路，在线路与你处所的计算机之间应该安装什么设备？

12.14　使用Web寻找DVD电影的大约容量。使用T1线路下载需要花费多长时间？使用T3线路呢（不计额外开销）？

12.15　如果有人给你出示铜电缆并声称是“OC-12”线路，他们犯了什么错误？他们应该使用什么名字才是正确的？

12.16　为什么同步数字体系的设计者选择不常用的数值（不正好是10的幂）作为数据速率？

12.17　请解释SONET帧的大小是如何计算的？

第三部分

分组交换及网络技术

分组交换和基于有线、无线介质的分组技术

第13章 局域网：分组、帧和拓扑

13.1 引言

本书第一部分的内容涵盖了因特网应用和网络编程技术。第二部分探讨了数据通信方面的话题，每一章涵盖一个基本概念（例如复用），它形成所有计算机网络的基础。

本章作为第三部分的开始，要仔细考查有关分组交换和计算机网络技术方面的内容。在进行简短的评述后，将接着解释IEEE标准模型，并专注于硬件编址和帧标识的概念。

这部分的后续章节将扩展这里讨论的内容，考虑广域网中分组技术的应用。此外，后续章节还会涵盖各种能接受和传递分组的有线和无线的联网技术。

13.2 线路交换

术语线路交换（circuit switching）[⊖]指的是一种在发送方与接收方之间建立一条通路的通信机制，它能保证该通路与其他发送方和接收方之间使用的通路是分开的。线路交换通常与电话技术紧密相关，因为电话系统在两台电话之间提供了专门的连接。事实上，这个术语起源于早期那种使用机械式交换设备来形成物理电路的电话网络。图13-1说明了通信是如何在线路交换网络上进行的。

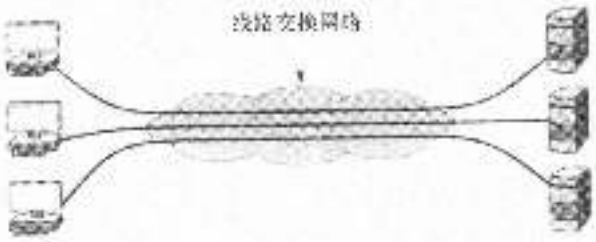


图13-1 能在每对通信实体间提供直接连接的线路交换网络

当前，线路交换网络使用电子设备[⊜]来建立线路，并且它也不是让每条线路对应一条物理通路，而是让多条线路复用在共享介质上。所形成的线路称为虚线路（virtual circuit）。因此，线路交换与其他网络交换形式之间的区别并不在于是否有独立的物理通路存在。3个一般特性定义了线路交换模式：

- 点对点通信。
- 线路建立、使用和终止由分开的步骤完成。
- 性能上等价于一条独立的物理通路。

第一种属性意味着一条线路恰好在两个端点之间形成，而第二种属性用于将交换式线路

⊖ 请看注释2的内容。

⊜ 从目前的现状和将来发展来看，交换设备已不再限于是电子设备了，可以是光学设备甚至是计算机软件实现的交换运营。因此，circuit一词并不是只有“电路”这一个意思，所以传统的“电路交换”应改称为“线路交换”才是合适的。——译者注

（即在需要的时候才建立）与永久式线路（即总保持随时可用状态）区分开来。线路交换使用类似于拨打一个电话的三阶段处理过程。在第一阶段，建立线路；在第二阶段，通信双方使用线路进行通信；在第三阶段，双方终止线路的使用。

第三种属性提供了线路交换网络与其他网络类型之间一个关键性的不同。线路交换意味着双方的通信不会受到其他通信方之间通信的任何影响，哪怕所有的通信都复用在一个公共的介质上。特别地，线路交换必须为每一对通信实体提供一种使用独立通路的幻象。因此，诸如频分多路复用或同步时分多路复用之类的技术，必须用在共享介质上实现线路复用。

要点 线路交换在一对通信实体间提供一种独立物理通路的幻象。通路在需要时就建立，使用完成后就断开。

13.3 分组交换

除了线路交换外，另一种主要的交换方式是分组交换（packet switching），它是形成因特网的基础。分组交换系统使用统计多路复用技术，来自多个信源的通信竞争使用共享介质。图13-2所示为这种概念。

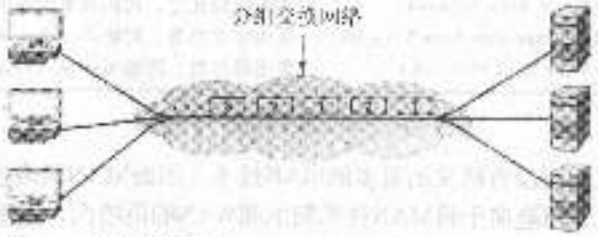


图13-2 在共享介质上每次只发送一个分组的分组交换网络

产生分组交换与统计多路复用的其他形式之间主要差别的原因，是由于分组交换系统要求发送方将每个报文分割成称为分组（packet）的数据块。每个分组的长度是可变的，每种分组交换技术都会定义一个最大的分组长度[⊖]。

3个一般特性定义了分组交换模式：

- 随意的、异步的通信。
- 通信开始前无须建立连接。
- 因分组间的统计多路复用致使性能可变。

第一个特性意味着分组交换允许一个发送方与一个或多个接收方通信，并且一个接收方可以接收来自于一个或多个发送方的报文，而且，通信可发生在任何时刻，且在相继的通信之间发送方可以延迟任意长的时间。第二个特性意味着分组交换系统与线路交换系统不同，它总是保持在准备状态，随时可向任何目的地发送分组。因此，发送方在通信前不需要执行初始化操作，也不需要在通信终止时通知底层系统。

第三个特性意味着多路复用发生在分组之间，而不是发生在码位或字节之间。也就是说，一旦发送方得到了访问底层信道的机会，它即可发送一个完整的分组，然后允许其他发送方发送一个分组。在其他发送方都不准备发送分组的时候，则允许单个发送方不断地发送分组。然而，如果N个发送方每个都有一个分组要发送，那么一个发送方只能发送大约是总分组量的1/N。

⊖ 分组不生太大，常通常最大分组长度是1500B。

概述如下：

> 分组交换是形成因特网的基础，它是统计时分多路复用的一种形式，允许多对多方式的通信。发送方必须将报文分割成一系列的分组。发送一个分组后，发送方在发送后续分组之前允许其他发送方发送分组。

分组交换的主要优点之一是成本较低，这是由于共享特性而带来的。为了提供N台计算机之间的通信，线路交换网络必须与每台计算机有一条连接，外加至少N/2条独立的通路。如果采用分组交换的话，虽然网络必须与每台计算机有一个连接，但却只需要一条共享的通路即可。

13.4　局域的和广域的分组网络

分组交换技术通常根据它们所跨越的距离来进行分类。费用比较便宜的网络使用的技术只能跨越较短的距离（例如，在一幢建筑物内部），而费用最贵的网络却能跨越很长的距离（例如，跨越几个城市）。图13-3概括了所使用的术语。

名　称	全　称	描　述
LAN	局域网（Local Area Network）	费用比较便宜，操作在单个房间或单个建筑物范围内
MAN	城域网（Metropolitan Area Network）	费用中等昂贵，操作在一个大城市范围内
WAN	广域网（Wide Area Network）	费用最昂贵，跨越多个城市或站点

图13-3　分组交换网络的3个类别

在实际使用中，人们还没有研发出很多的MAN技术，因而MAN网络也就没有在商业上取得成功。因此，网络专业人员趋向于将MAN技术归纳到WAN的范畴内，只使用LAN和WAN术语。

13.5　分组标识及其格式标准

由于分组交换系统依赖于共享机制，所以在这种网络中发送的分组必须要包含对应接收方的标识。此外，为确保不会产生歧义，所有发送方必须在如何标识接收方和将标识放在分组的什么位置等具体细节上达成一致意见。标准化组织创建了规定所有这些细节的协议文档。最为广泛使用的一套LAN标准由电子电气工程师协会（Institute for Electrical and Electronic Engineers，IEEE）创建。

1980年，IEEE组建了802项目LAN/MAN标准委员会（Project 802 LAN/MAN Standards Committee）来制订网络标准。这个标准化组织由那些专注于协议栈底下两层的工程专家所组成，知道这一事实对于理解IEEE的标准非常重要。事实上，如果我们阅读了IEEE文档，那么网络的其他方面看起来就不那么重要了。然而，也存在着其他的标准化组织，而且每个组织都会强调协议栈的某个特定层次。图13-4给出了各个标准化组织如何看待某种协议的一种较为幽默的说明。

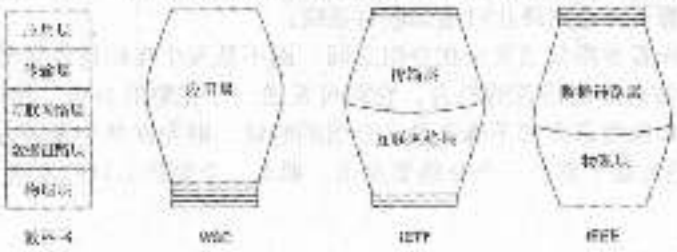


图13-4　各种标准组织如何描述协议栈的一种幽默示意图

这样看来，人们就不应该下结论说，由某特定组织制定的标准就是全面的，或者认为标准出版的质量就会与特定层次的重要性成比例。概述如下：

每个标准化组织都专注于协议栈的特定层次。IEEE标准专注于针对协议栈最底下两层和LAN技术方面的规范。

13.6 IEEE 802模型与标准

为了有助于描述标准的特征，IEEE将协议栈的第二层划分为两个概念子层（sublayer），如图13-5所示。

子 层	全 称	用 途
LLC	逻辑链路控制（Logical Link Control）	寻址和解复用
MAC	介质接入控制（Media Access Control）	接入共享介质

图13-5 根据IEEE模型将第二层划分为两个概念子层

逻辑链路控制子层规定编址以及解复用技术中地址的使用（本章后面将介绍）。介质接入控制子层规定多台计算机如何共享底层的介质。

IEEE宁可不使用文字名称来标识那些从事标准工作的工作组或最后的标准文档名称，而是分配一种形如XXX.YYY.ZZZ的、由多部分组成的标识符。数值XXX表示标准的大类，后缀YYY表示子类。如果子类还很大，可以增加第三级来区分特定的标准。例如，LAN规范已经被分配了类别802。因此，每个设计LAN标准的工作组被分配给一个ID，例如802.1、802.2，等等。注意，值802和独立的后缀并不传达任何技术含义——它们只不过是对标准进行标识而已。图13-6列出了IEEE分配数值的例子。

ID	主 题
802.1	较高层LAN协议
802.2	逻辑链路控制
802.3	以太网
802.4	令牌总线（已解散）
802.5	令牌环
802.6	城域网（已解散）
802.7	使用同轴电缆的宽带局域网（已解散）
802.9	综合业务局域网（已解散）
802.10	可互操作的局域网安全（已解散）
802.11	无线局域网（Wi-Fi）
802.12	需求优先级
802.13	6类10Gbit/s规程
802.14	电缆调制解调器（已解散）
802.15	无线PAN（个域网）
	802.15.1（蓝牙，Bluetooth）
	802.15.4（紫蜂，ZigBee）
802.16	宽带无线接入
	802.16e（移动）宽带无线
802.17	弹性分组环
802.18	无线管制TAG
802.19	共存TAG
802.20	移动宽带无线接入
802.21	介质无关切换
802.22	无线地域网（WRAN）

图13-6 IEEE已经分配给各种LAN标准的标识符例子

如图所示，IEEE已经开设了很多工作组，每个工作组致力于标准化一种网络技术类型。工作组由工业公司和学术团体的代表组成，他们经常有规律地碰在一起讨论解决方法和设计标准。在工作组取得进展并且研究的技术依然被认为重要的前提下，IEEE允许一个工作组一直保持活动状态。如果工作组判定正在研究的技术不再与工作组的主题相关，它可以决定解散工作组，例如，可能会发现一种更好的技术，从而再作进一步的标准化变得毫无意义。另外，另一个标准组织可能先产生了一个类似的标准，这会使IEEE的努力有点多余。因此，图13-6包含了那些曾经很重要但现在已经解散了的主题。

13.7　点对点与多址接入网络

回顾术语点对点（point-to-point），它是指恰好连接两个通信实体的一种通信机制。LAN技术允许多台计算机按这样一种方式共享介质，即LAN上的任一台计算机可与其余的任一台计算机相互通信。为了描述这样的组织形式，我们使用术语多址接入（multi-access），而且把LAN说成是一种多址接入式网络。

LAN技术一般都提供通信实体之间的直接连接。专业人员虽说LAN是连接着计算机的，但其实他们也知道，诸如打印机之类的设备也是可以连接到多址接入LAN的。

13.8　LAN拓扑

由于已经发明了很多种局域网技术，那么了解各种具体技术的类似之处和不同之处是很重要的。为了帮助了解各种技术类似的地方，每一种网络都根据它的拓扑（topology）或一般形状进行了分类。这一节主要介绍构成LAN的4种基本拓扑，后面将讨论具体的技术。图13-7所示为几种拓扑。

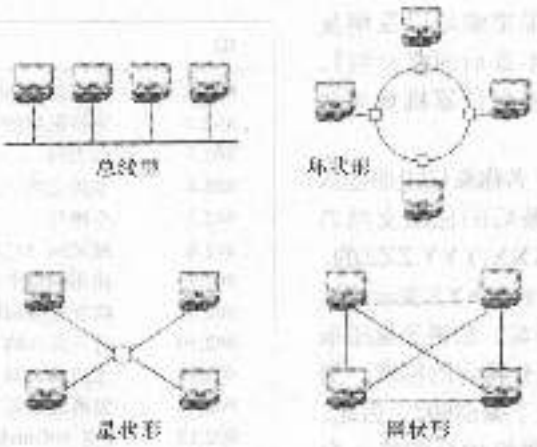


图13-7　LAN中使用的4种拓扑

13.8.1　总线拓扑

使用总线拓扑（bus topology）的网络通常采用单根电缆将所有计算机连接起来[○]。任何连接到总线上的计算机都能发送信号到总线上，并且所有计算机都能接收到这个信号。由于所有计算机直接连接在电缆上，因此任何一台计算机都能向其他计算机发送数据。当然，连接在总线网络上的计算机必须相互协调，以保证在任何时候只有一台计算机发送信号。

13.8.2　环形拓扑

使用环形拓扑（ring topology）的网络把计算机连接成一个封闭的回环——一根电缆连接第一台计算机与第二台计算机，另一根电缆连接第二台计算机与第三台，依此类推，直到一根电缆连接最后一台计算机与第一台计算机。使用环形拓扑的有些技术要求计算机连接到一个小型的环接设备上，这样做的优点是：即使某些计算机断开了连接，而环路本身仍然有能力继续运行。因为我们能想象计算机和连接计算机的电缆被安排成一个如图13-7所示的圆圈的样子，所以才产生了环这个名字。在实际中，环形网络中的电缆并不形成一个回圈，而是

○　在实际使用中，总线电缆的端点必须终接匹配，以防止电子信号沿着总线反射回来。

可以顺着过道或垂直地从大楼的一层到另一层，可以有任意的走向。

13.8.3 网状拓扑

使用网状拓扑（mesh topology）的网络在每一对计算机之间提供直接连接。网状拓扑的主要缺点在于费用问题——连接n台计算机的网状网络要求：

$$\text{网状网络中的连接数}=\frac{n!}{(n-2)!2!}=\frac{n^2-n}{2} \tag{13.1}$$

最重要的是：网状网络所需的连接数的增长速度远快于计算机数目的增长速度。由于连接费用昂贵，很少有LAN会采用网状拓扑结构。

13.8.4 星形拓扑

当所有的计算机都连接到一个中心节点时，网络即形成了星形拓扑（star topology）。因为星形网络很像车轮子的轮毂（英文词汇是hub），所以它的中心节点就通常被称为集线器（hub）。典型的集线器其实就是这样一种电子设备：它从发送计算机接收数据，然后再把数据转发到合适的目标计算机。

实际上，星形网络很少会使集线器位于与所有计算机相同距离的地方而呈对称形状的。相反，集线器通常安放在与所连计算机相分离的地方，例如，计算机在各自的办公室里，而集线器却安放在单位网络管理员容易接近的地方。

13.8.5 多种拓扑可用的理由

每种拓扑都有各自的优点与缺点。环形拓扑使计算机容易协调访问以及容易检测网络是否正确运行。然而，如果其中一根电缆断掉，整个环形网络都要失效。星形拓扑能保护网络不受任一根电缆损坏的影响，因为每根电缆只连接一台机器。总线型拓扑所需的布线比星形拓扑的少，但是存在与环形拓扑同样的缺点，即如果有人偶然切断主电缆，网络就要失效。后续章节在描述具体的网络技术时，会提供这些差异的额外细节。目前，了解下面的内容就够了：

> 网络按照它们的一般形状被分为几个大类。虽然网状拓扑也有可能采用，但是局域网采用的主要拓扑是星形、环形和总线型，每种拓扑都有各自的优点和缺点。

13.9 分组标识、解复用、MAC地址

除了规定各种LAN技术细节的标准外，IEEE还为寻址（addressing）问题制定了标准。为了理解寻址问题，考虑一个如第13.3节的图13-2所示那样穿越共享介质的分组。在最简单的情况下，在共享介质上传播的每个分组都要发送给特定的接收方，并且只有期望的接收方才应该处理这个分组。在分组交换系统中，解复用技术[○]使用一个叫做地址的标识符。每台计算机都会分配一个唯一的地址，同时每个分组也包含着期望接收方的地址。

在IEEE的编址方案中，每个地址由48位组成。IEEE使用术语介质接入控制地址（Media Access Control address，简称为MAC地址）。由于48位的地址来源于以太网技术，因此网络专业人员也经常使用术语以太网地址。为了保证每个地址的唯一性，IEEE为每块网络接口硬件

○ 在本章13.3节中曾提过，分组交换技术其实就是一种共享介质上采用的统计时分多路复用技术（multiplexing），那么，它的逆过程（将分组从复用分组流中分离出来寻址到它的目的地的过程）自然地使用解复用（demultiplexing）。——译者注

分配一个地址。因此，如果消费者为PC购买了一块网络接口卡（Network Interface Card，NIC），那么NIC就包含了一个唯一的IEEE地址，该地址是在制造这个设备的时候分配的。

IEEE不分配单个的地址，而是给每个设备制造商分配一个地址块，并允许制造商为他们制造的每个设备分配一个唯一的地址值。因而，48位的地址被划分成3字节的机构唯一标识符（Organizationally Unique ID，OUI）用以标识设备制造商和3字节块用以标识一个特定的网络接口控制器（Network Interface Controller，NIC）。图13-8所示为这种划分方法。

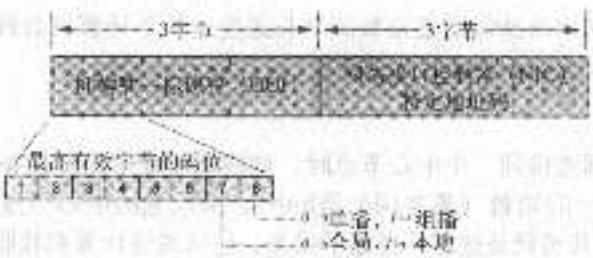


图13-8　48位IEEE MAC地址的划分方法

有意思的是，OUI中最高有效字节的两个低阶位（按上图的指示）已经分配了特殊的含义。其中的最低位是一个组播位，用来规定它是一个单播地址（0）还是一个组播地址（1）。另一位规定OUI是全局唯一的（0）还是本地分配的（1）。下一节将解释组播技术。全局唯一地址由IEEE分配。而本地分配地址只对实验性工作或那些想创建自己的地址空间的机构有效。

13.10　单播、广播和组播地址

IEEE的编址方案支持3种地址类型，它们正好对应3种分组传递的类型。图13-9做了归纳。

地址类型	地址的含义以及分组传递的方式
单播	唯一标识一台计算机，并规定只有被标识的那台计算机才能接收分组的副本
广播	对应所有的计算机，并规定网络上每台计算机都应该接收分组的副本
组播	标识指定网络上所有计算机的一个子集，并规定该子集中的每台计算机都应该接收分组的副本

图13-9　3种MAC地址类型以及相应的含义

IEEE的做法看起来有些奇怪，因为IEEE地址格式中保留了一位用于区分组播和单播却不提供指明广播地址的方式。实际上，标准规定广播地址由48个全"1"的位组成，因此，广播地址已经将组播位给置位了。从概念上讲，广播可以看成是一种特殊形式的组播，也就是说，每个组播地址对应于一组计算机，而广播地址对应于一个包含网络上所有计算机的组。

13.11　广播、组播和高效的多点传递

广播和组播在LAN中特别有用，因为它们准许向很多计算机进行高效的传递。为了理解这种高效性，回顾一下LAN在共享介质上传输分组的情景。在一个典型的LAN中，LAN中的每个计算机监视着共享介质，提取每个分组的副本，然后检查分组中的地址以决定是否应该处理这个分组还是忽略它。算法13-1给出了计算机处理分组的算法。

从上面的算法来看，它的效率应该是很清楚的。在广播或组播情况下，分组只有单份副本在共享介质上传输，而所有（或部分）计算机都会接收并处理这个副本。例如，考虑广播情况，发送方无须进行N次单独的发送（即给每台计算机都要发送分组的一个副本），只须发

送分组的一个包含广播地址的副本即可让所有计算机收到这个副本。

```
算法13-1
目的：
  处理LAN上已经到达的分组
方法：
  从分组中提取目的地址ID；
  If (ID匹配"我自己") {
    接受并处理分组；
  } else if (ID匹配广播地址) {
    接受并处理分组；
  } else if (ID是在一个我是其成员的组播组对应的组播地址)
    接受并处理分组；
  } else {
    忽略分组；
  }
```

算法13-1 LAN中使用的分组处理算法

13.12 帧与成帧

第9章在讲述同步通信系统时介绍了成帧的概念，它作为一种机制允许接收方知晓报文在哪里开始和结束。在更一般的意义下，我们使用术语成帧（framing）来指被添加到一连串的位或字节上的结构，这种结构允许发送方和接收方就报文的正确格式达成一致。在分组交换网络中，每个帧都会对应一个分组。一个帧由两个概念部分组成：

- 包含元数据（如地址）的头部。
- 包含发送数据的载荷。

帧的头部包含一些用来处理帧的信息。特别地，头部通常包含一个地址来指定期望的接收方。载荷域包含要发送的报文，并且它通常比帧头部大很多。在大多数网络技术中，网络只检查帧头部，在这个意义下，报文是不透明的。因此，载荷区可以包含任何只对发送方和接收方有意义的字节序列。

通常安排帧的头部在前，载荷在后传输，因为这样可以使接收方在帧头部码位到达时就开始处理帧。有一些技术通过在帧前面发送一小段前导码（prelude），在帧后面发送一小段后接码（postlude）来勾画帧的完整性。图13-10说明了这种概念。



图13-10 分组交换网络中典型的帧结构

为了理解成帧的原理，不妨考虑一个使用字节构成帧的例子，即假设一种数据通信机制能将任意8位字节从发送方传送到接收方，并设想这种机制用于发送分组，再假设分组头部由6个字节构成而载荷部分由任意个字节构成。我们将使用单个字节来标记帧的开始，并用单个字节来标识帧的结束。在ASCII字符集中，头部起始字符（Start Of Header，SOH）标记帧的开始，传输结束字符（End Of Transmission，EOT）标记帧的结束。图13-11说明了这种格式。

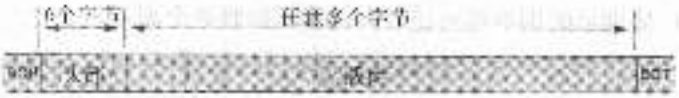


图13-11　帧格式示例，它使用SOH和EOT字符来与固定的完整单

上面示例的帧格式看起来似乎有一些不必要的开销，为了了解其原因，我们考虑当发送方无延迟地发送两个帧时会发生什么现象。当第一个帧发送结束时，发送方发送EOT，然后无延迟地发送SOH以开始第二帧的发送。在这种情况下，只需一个字符就可分开这两块数据——这种标记帧开始和结束的成帧方案看起来似乎在帧之间多发送了一个多余的、无用的字符。

当我们考虑分组传输采用异步方式且可能会发生错误的时候，这种在帧结束时发送字符的方案的优点就显而易见了。对异步通信而言，使用EOT来标记帧的结束可使发送方无须等到下一帧的开始就能处理当前帧。在出现错误的情况下，使用SOH和EOT来包裹帧有助于错误恢复和同步——如果发送方在传输一个帧的期间崩溃，接收方就有能力判定出不完整帧的到达。

13.13　字节插入与位插入

在ASCII字符集中，SOH的十六进制值是0x01，EOT的值是0x04，现在的问题是：如果帧的载荷中也含有一个或多个其值为0x01或0x04的字节，就会产生对帧错误定界的问题。对这个问题的解决，是采用一种称为字节插入（byte stuffing）的技术，它能允许传输任意数据而不会发生与SOH或EOT帧定界符相混淆的问题。

通常，为了区分数据与控制信息（例如帧的定界符），发送方采用插入某种特殊的位元序列（或字符）来替换数据中的每个控制字节，然后接收方再将这些特殊序列替换回原来的值。这样，帧就能传输任意的数据而底层系统却不会把它们与控制信息相混淆。这种技术被称为字节插入（byte stuffing）。有时候也使用术语数据插入（data stuffing）和字符插入（character stuffing）。在传输位元流的系统中，与此相关的技术则称为位插入（bit stuffing）。

作为字节插入的一个例子，我们考虑图13-11所示的帧。因为字符SOH与EOT用来标记帧的定界，所以这两个字节就不能出现在载荷中。字节插入技术能解决这个问题，它使用第三个字符来标记数据中保留字符的出现。例如，假设选择ASCII字符ESC（十六进制值为1B）作为第三字符。当这三个特殊字符中的任何一个在数据中出现时，发送方就用一个2字符序列来取代这个字符。图13-12列出一种可能的映射关系。

载荷中的字节	发送序列
SOH	ESC A
EOT	ESC B
ESC	ESC C

图13-12　字节插入的一个例子，它将每个特殊字符映射到一个2字符的序列

如采用上图的规定，发送方用ESC和A两个字符去替换每次出现的SOH，用ESC和B替换每次出现的EOT，用ESC与C代替每次出现的ESC。接收方进行反向映射，它寻找ESC的出现以及随后的A或B或C，用对应的单个字符去替换2字符组合码。图13-13所示为一个原来的载荷域和经过字节插入后的载荷域的例子，应注意到，一旦完成了字节插入，SOH和EOT就不会再出现在载荷域中了。

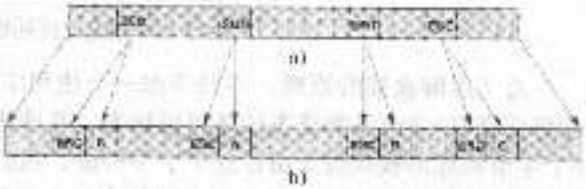


图13-13　字节插入示意图，其中：a)原始数据的版本；b)经字节插入后的版本

13.14 本章小结

数据网络可以分为线路交换和分组交换两类。分组交换构成了因特网的基础，它是统计多路复用的一种形式。分组交换网络中发送方会将报文划分为小的分组，分组交换网络技术又分为局域网（LAN）、广域网（WAN）和城域网（MAN），其中LAN和WAN最流行。

一个名为IEEE的机构已经为数据网络制定了标准，IEEE标准主要规定LAN的细节并专注于协议栈的最低两层。

人们使用4种基本形状或拓扑来描述LAN的结构：总线型、星形、环形和网状形。网状形拓扑很少使用，因为它要求的建网费用较高。

在LAN上发送的每个分组都含有一个用来标识期望接收方的MAC地址。IEEE的MAC地址标准规定了一个48位的数值，它划分为两个域：一个域标识分配此地址的机构，另一个域指派一个唯一的值给特定的一块硬件（即分配该地址的那块硬件）。一个地址可以指定为单播（单台计算机）、广播（指定LAN上的所有计算机）或组播（LAN上所有计算机的一个子集）。

术语帧用于指定特定网络上的分组格式。帧由两个概念部分组成：包含元信息的头部和包含发送数据的载荷域。对于传输字符的网络而言，帧可以这样构成：使用一字节的值指示帧的开始，再用另一字节的值指示帧的结束。

为了用字节（位）标记帧的开始和结束，字节（位）插入技术准许保留某些字节（位序列）。为了确保载荷中不包含保留的字节（位串），发送方在发送前会替换出现的保留值，而接收方会对这种改变实施反操作以恢复原来的数据。

练习题

13.1 什么是线路交换，它的主要特征是什么？

13.2 在线路交换网络中，多条线路能否共享单条光纤？试解释。

13.3 在分组交换系统中，发送方如何发送一个大的文件？

13.4 如果某人想广播一个视频表示的副本，线路交换与分组交换哪个更可取？为什么？

13.5 LAN、MAN和WAN的特征是什么？

13.6 说出IEEE定义的第二层协议的两个子层名称，并指出它们的用途。

13.7 什么是点对点网络？

13.8 4种基本的LAN拓扑是什么？

13.9 令牌环网中的缆线能否排列成一条直线（例如，沿着过道）？试解释。

13.10 在一个网状网络中，20台计算机之间需要多少个连接？

13.11 给定一个IEEE的MAC地址，我们如何才能判定它是不是一个单播地址？

13.12 试定义单播、组播和广播地址，并解释它们的含义。

13.13 连接到共享LAN上的计算机如何判定是否要接受一个分组？

13.14 使用什么术语来描述伴随分组的元数据？

13.15 试给出术语帧的定义。

13.16 为什么需要字节插入？

13.17 编写一对计算机程序：一个用于接受数据文件作为输入并根据图13-12中的映射关系产生文件的字节插入版本；另一个用于去除字节插入。请说明你的程序能与其他人编写的程序进行互操作。

第14章 IEEE MAC子层

14.1 引言

本书这一部分的各章涵盖采用分组交换技术的数据通信网络。前一章介绍了分组交换的概念并定义了两种基本的分组交换网络类型：WAN和LAN，还介绍了协议标准的IEEE模型，并解释了IEEE为何将第二层划分为两个子层。

本章继续前面的讨论，考查IEEE的MAC子层。本章将解释多址接入协议并考虑静态和动态的信道分配机制。这一部分的后续各章将讨论具体的网络技术，它们要用到这里所讲述的接入机制。

14.2 多址接入机制的分类

多台相互独立的计算机如何进行协调才能接入到一个共享的介质呢？有3种主要的方法，它们可以采用某种复用技术的改进形式，也可以加入某种分布式算法以进行受控接入或使用一种随机接入策略。图14-1说明了这种分类方法，其中包括每一种方法的具体形式。

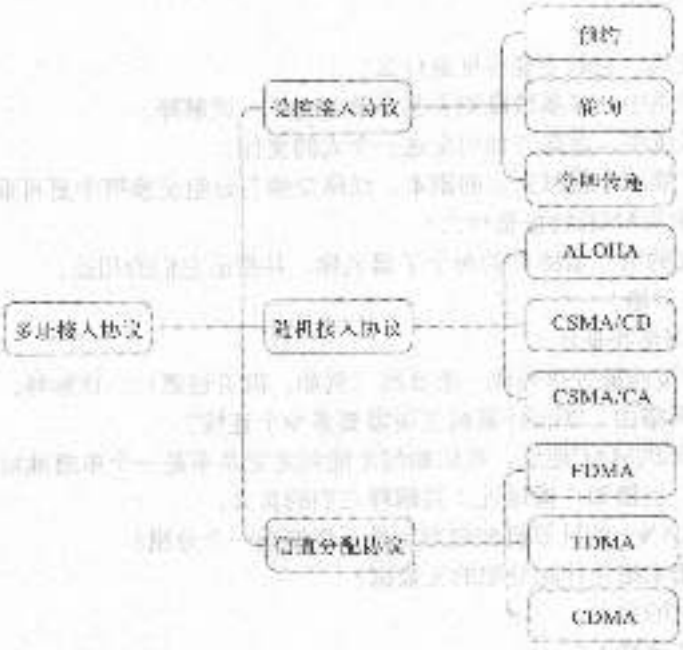


图14-1 控制接入共享介质的协议分类

14.3 静态与动态信道分配

我们使用术语信道分配（channelization）来指某给定的通信过程与底层传输系统中的一

个信道之间的映射关系。信道分配与第11章中讨论的复用技术相关。例如，考虑频分多路复用（FDM）机制。大多数FDM系统给每一对通信实体分配一个唯一的载波频率。即每一对通信实体被分配了一个唯一的信道。此外，一对通信实体与载波频率之间的映射一直保持不变。在这种情况下，我们将通信实体与信道之间的映射描述为一对一的关系而且是静态的。

在事先知道通信实体集合且集合保持不变的情况下，静态信道分配方案工作得很好。然而，在很多网络中，使用网络的实体集合会随时间而变化。例如，考虑一个城市中的蜂窝电话系统。用户会移动，并且他们会随时打开或关闭蜂窝电话，因此运行在指定蜂窝基站覆盖范围内的蜂窝电话集合会不断地改变。在这种情况下，我们需要一种动态信道分配方案——当一个新站点（例如蜂窝电话）出现时即可建立一个映射，而当一个站点消失时即可移除这个映射。

概括如下：

在事先知道通信实体集合且集合保持不变时，静态信道映射方式就能满足我们的需求，但大多数网络要求某种动态信道分配的形式。

14.4 信道分配协议

信道分配协议扩展了第11章中涵盖的复用技术。图14-2列出了主要的信道分配技术。

协　议	全　称
FDMA	频分多址接入（Frequency Division Multi-Access）
TDMA	时分多址接入（Time Division Multi-Access）
CDMA	码分多址接入（Code Division Multi-Access）

图14-2 3种主要的信道分配协议类型

14.4.1 FDMA

正如上图所示，信道分配技术采用了频分、时分和码分多路复用技术，例如，频分多址接入（Frequency Division Multiple Access，FDMA）就扩展了频分多路复用技术。本质上，这种扩展由这样一种机制构成，即允许独立的站点选择一个不会与其他站点相冲突的载波频率。FDMA如何分配载波呢？在某些系统中，一个中心控制器提供了一种动态的分配方式。只要一个新站点出现，它就会使用一个保留的控制信道与控制器通信，这个站点发出一个请求，控制器选择一个当前没有使用的频率然后通知站点。经过初始的交换后，这个站点即可使用所分配的载波频率（即分配的信道）进行所有通信。

14.4.2 TDMA

时分多路复用的扩展叫做时分多址接入（Time Division Multiple Access，TDMA），它类似于频分多路复用的扩展。在最简单的情况下，每个处于活动状态的参与者都被分配一个1～N之间的顺序号，站点即按1，2，3，…，N的顺序发送，与FDMA中的做法一样，有些TDMA系统提供动态分配机制，即当一个站点第一次出现在网络中时，就分配一个时隙给它使用，而当它从网络中消失时，就收回那个时隙。

14.4.3 CDMA

码分多路复用采用数学方法对传输信号进行编码，允许多个站点同时发送。正如第11章中解释的那样，码分多址接入（Code Division Multiple Access，CDMA）是码分多路复用技

术的一种主要应用。

14.5 受控接入协议

受控接入协议提供了统计复用的一种分布式版本。图14-3列出了3种主要的形式。

类　型	说　明
轮询	中心控制器不断重复查询每个站点并允许每个站点发送一个分组
预约	站点为下一轮的数据发送提交一个请求
令牌传递	站点自循环传递一个令牌；站点每次收到一个令牌时即可发送一个分组

图14-3　受控接入协议的主要类型

14.5.1 轮询

采用轮询（polling）技术的网络要用到一个中心控制器。中心控制器会循环扫描网络上的站点，给每个站点一次发送分组的机会。算法14-1给出了控制器遵循的步骤。其中的选择步骤很重要，因为它意味着在某一指定时刻控制器可以选择查询哪个站点。这里有两种通用的轮询策略：

- 按循环顺序。
- 按优先级顺序。

按循环顺序意味着每个站点有均等的机会发送分组；按优先级顺序则意味着某些站点将有更多的发送机会。例如，按优先级顺序的做法可以为一台IP电话分配比个人计算机更高的优先级。

算法14-1

```
目的：
    通过轮询来控制分组的发送
方法：
    控制器不断重复{
        选择站点S，发送一个查询报文给S；
        等候S发送一个分组来进行响应或跳过；
    }
```

算法14-1　通过轮询来控制接入

14.5.2 预约

预约（reservation）系统经常与卫星传输一起使用，它采用一个具有两个步骤的过程，其中每一轮分组的发送都是事先安排好的。典型地，预约系统都有一个中心控制器，它按算法14-2进行控制。

算法14-2

```
目的：
    通过预约来控制分组的发送
方法：
    控制器不断重复{
        形成一个需要发送分组的站点列表；
        允许列表中的站点发送分组；
    }
```

算法14-2　通过预约来控制接入

第一步，每个潜在的发送者都会指出在下一轮的发送过程中，它们是否有分组要发送，然后控制器发送一个将要发送分组的站点的列表。第二步，站点利用这个列表即可知道它们应该在什么时候发送分组，也有不同的做法，即在主信道上进行当前这一轮传输的期间，控制器利用另一个信道收到下一轮的预约信息。

14.5.3 令牌传递

令牌传递（token passing）已经在好几种局域网技术中使用，并且与环形拓扑紧密相关[⊖]。为了了解令牌传递，可以想象一些计算机连接成一个环，并设想在任意时刻，恰好仅有一台计算机接到一种叫做令牌（token）的特殊控制报文。为了控制接入过程，每台计算机都遵循算法14-3。

```
算法14-3
目的：
    通过令牌传递来控制分组的发送
方法：
    网络上的每台计算机重复执行{
        等待令牌的到达；
        如果本计算机有分组正在等待发送，则发送一个分组；
        将令牌发送到下一个站；
    }
```

算法14-3 通过令牌传递来控制接入

在令牌传递系统中，当不再有站点发送分组的时候，令牌在所有的站点间不断地循环传递。对于环形拓扑，循环的顺序由环来规定，也就是说，如果环按顺时针方式发送报文，那么上述算法中的下一个站指的就是按顺时针顺序的下一个物理站点。当令牌传递方式应用到其他拓扑结构（例如总线结构）的时候，每个站点按逻辑顺序分配一个位置，令牌根据站点分配的顺序来传递。

14.6 随机接入协议

很多网络，特别是局域网，都没有采用受控接入机制。相反，连接到共享介质上的一些计算机却试图不经过协调就去接入介质。这里用到了随机（random）这个术语，因为当一个指定的站点有分组需要发送的时候接入才会出现，并且它会采用随机选择方式，以防止局域网中所有计算机试图同时使用介质。下面描述的具体方法将阐明随机选择的使用含义。

图14-4列出3种将要讨论的随机接入方法。

类　型	说　明
ALOHA	历史上著名的一种协议，在无线网的早期无线网络中使用过，它在教科书中比较流行并且容易分析，但是没有在实际网络中使用
CSMA/CD	带冲突检测的载波侦听多址接入。它是以太网的基础，并且也是最为广泛使用的随机接入协议
CSMA/CA	冲突避免的载波侦听多址接入。它是Wi-Fi无线网络的基础

图14-4 3种随机接入协议

⊖ 虽然老的LAN有使用令牌传递环技术的，但是其流行性已经降低，几乎没有令牌传递网络还流行下来。

14.6.1　ALOHA

夏威夷有一种早期网络叫做ALOHA网（见图14-5），它开辟了随机接入概念的先河。虽然这种网络已不再使用，但它的思想却得到了发展。网络由单台位于中心地理位置的功能强大的发射机，以及围绕在其周围的一组站点（每个对应一台计算机）组成。每个站点也都有一个发射机，其辐射范围能到达中心发射机（但不足以到达所有其他的站点）。ALOHA网络使用两种载波频率：一种工作在413.475MHz，它用于向下传递，即由中心发射机发送给所有站点的广播业务；另一种工作在407.305MHz，它用于向上传递，即由各站点向中心发射机发送的业务。

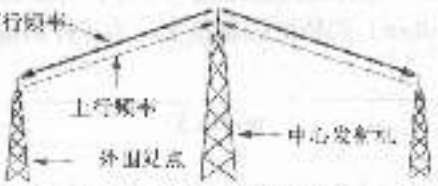


图14-5　ALOHA网的下行与上行频率示意图

ALOHA协议的原理很简单：当一个站点有分组要发送的时候，它在上行频率上发送这个分组。中心发射机在下行频率上重发这个分组（因而所有站点都能接收到）。为了确保发送的成功，发送站点会侦听下行信道。如果下行信道上有分组的一个副本到达，发送站点会转移到下一个分组的发送阶段；如果没有副本到达，发送站点会等待一小段时间然后重新尝试。

为什么会有分组不能到达呢？答案就是干扰——如果两个站点试图同时在上行频率上发送分组，那么信号就会相互干扰而这两个传输也会出现混乱。我们使用术语冲突（collision）来描述它，并说这两个被发送的分组在介质上出现冲突。协议通过请求发送方重新发送每个损坏的分组来处理冲突问题，这种思想很普遍，并且出现在很多网络协议中。

重传之间等待的时间长度必须进行仔细地选择。否则，两个站点会在重发之前又正好等待相同的时间，从而再一次相互干扰。因此，如果加入随机选择的话（即每个站点选择一个随机延迟），那么干扰的概率就会很低。分析表明，当ALOHA网变得繁忙时，它会发生很多冲突。即使采用了随机选择方式，这种冲突现象也会使得ALOHA网数据传输的成功率降低到信道容量的大约18%（即信道的利用率最高大约18%）。

14.6.2　CSMA/CD

1973年，施乐PARC中心的研究人员开发了一种极其成功的网络技术，它采用随机接入协议。1978年，数字设备公司（Digital Equipment Corporation），英特尔公司和施乐公司共同制定了一种标准（正式叫法为DIX标准），它就是以太网（Ethernet）。原始的以太网由一根很长的同轴电缆以及连接在电缆上的计算机组成[⊖]。电缆充当共享介质——以太网并不借助大气传输来广播无线频率，而是在电缆上传输信号。此外，以太网并不使用两种频率和中心发射机，而是允许所有通信都在共享电缆上进行。虽然以太网和ALOHA网有一些差异，但是它们都必须解决同样的基本问题，即如果两个站点试图同时发送，那么信号就会干扰并且出现冲突。

以太网提供了3种新方法来处理冲突问题：

- 载波侦听（carrier sense）。
- 冲突检测（collision detection）。
- 二进制指数退避（binary exponential backoff）。

载波侦听，以太网并不允许一个站点只要有分组做好了准备就可以发送，而是要求每个

⊖　下一章将考虑现代以太网的布线方案。

站点监视电缆，检测是否有另一个传输正在处理之中。这种机制就叫做载波侦听，它阻止了最明显的冲突问题，并且能充分提高网络的利用率。

冲突检测。尽管采用了载波侦听技术，但是如果两个站点正在等待某一传输的结束，发现电缆空闲后同时启动发送过程的话，还是可能发生冲突。问题还有另一个方面，即信号即使以光速传输，它沿整条电缆传递还是需要一些时间的。因此，电缆一端的站点无法立刻知道另一端的站点什么时候开始发送。

为了处理冲突问题，每个站点在发送过程中都会监视电缆。如果发现电缆上的信号与本站发出去的信号不符，那就意味着出现了冲突。这种技术称为冲突检测。只要检测到冲突，发送站点就会立即终止发送。

许多技术细节使得以太网的传输变得更加复杂。例如，在出现冲突后，传输过程不会马上终止，直到已经发送了足够多的位，这样做可以保证冲突信号到达所有的站点。另外，在一次传输之后，站点必须等待一个分组间间隙（interpacket gap）（10Mbit/s以太网中是9.6μs），以确保所有站点都感觉到网络已经空闲并且有信道可用于传输。这些细节说明，以太网技术被设计得多么地仔细。

二进制指数退避算法。以太网不仅要检测冲突，还要从冲突中恢复过来。在一个冲突发生之后，计算机必须等待电缆再次空闲后才能发送帧。与ALOHA网中的做法类似，以太网使用随机选择的办法来避免电缆一出现空闲就会有多个站点同时进行发送。做法是：标准规定一个最大延迟值d，要求每个站点在冲突发生后选择一个小于d的随机延迟。在大多数情况下，当两个站点每个都选择一个随机值时，选择了较小延迟值的站点将先开始发送帧，于是网络就恢复正常运行了。

如果有两台或多台计算机恰好选择了几乎相同的延迟，那么它们将几乎同时开始发送，导致第二次冲突。为了防止一连串的冲突，以太网要求每台计算机在每次冲突后把选择延迟的范围加倍。这样的话，计算机在第一次冲突后在0～d之间选择一个随机延迟，第二次冲突后在0～2d之间选择，第三次冲突后在0～4d之间选择，依此类推。在几次冲突后，选择随机值的范围就会变得很大。这样，一些计算机会选择比其他计算机短的随机延迟，从而避免了再次发生冲突的可能[⊖]。

每次冲突后随机延迟的范围加倍就是所谓的二进制指数退避法。本质上，指数退避法意味着以太网能在冲突后迅速恢复，因为当电缆繁忙时每台计算机都同意在两次尝试之间等待更长时间。即使两台或多台计算机选择几乎相等的延迟（这只是极少数的偶发事件），这时指数退避法也能保证经几次冲突后对电缆的竞争性将大大降低。

以上描述的这几种技术组合到一起，其名称就是带冲突检测的载波侦听多址接入（Carrier Sense Multiple Access with Collision Detect，CSMA/CD）。算法14-4对CSMA/CD的过程进行了归纳。

算法14-4

目的：

使用CSMA/CD发送一个分组

方法：

等待分组做好发送准备；

算法14-4 使用CSMA/CD发送分组

⊖ 这段话的意思是：通过对退避时间的随机化，将可以发生的事件分离开，以避免在次同时出现而发生冲突。这种过程通常叫做“冲突分解”。——译者注

```
等待介质出现空闲（载波侦听）；
延迟一个分组传输时间；
将变量x设置为从集合{0,1,2,3,4,5,6,7}
    中随机选择的一个值；
试发送分组（冲突检测）；
当（在前一次传输中发生冲突）{
    从{0,1,2,3,4,5,6,7}中随机选择一个值并赋给q；
    延迟q个时间片；
    将x加倍，直到下一轮之前；
    试发重传分组（冲突检测）；
}
```

算法14.4　（续）

14.6.3　CSMA/CA

虽然CSMA/CD在电缆介质上工作得很好，但是它在无线LAN中却不会工作得如此出色。这是因为无线LAN中所用的发射机有一个受限的发射范围δ，就是说，离发射机的距离超过δ的接收方将无法收到信号，因而无法检测载波。为了理解距离限制为什么会造成CSMA/CD出现这种问题的原因，不妨考虑一下安装有无线LAN硬件的3台计算机，它们的位置如图14-6所示。



图14-6　以最大距离δ配置的3台无线LAN计算机

在图中，计算机1能与计算机2通信，但不能接收计算机3的信号。因此，如果计算机3向计算机2发送一个分组，计算机1的载波侦听机制将无法检测到这个传输过程。类似地，如果计算机1和计算机3同时发送分组，只有计算机2才能检测到冲突。这种问题有时称为站点隐藏问题（hidden station problem），是指有些站点对其他站点来说是不可见的。

为了保证所有站点能正确地共享介质，无线LAN使用一种改进的接入协议，叫做避免冲突的载波侦听多址接入（Carrier Sense Multiple Access with Collision Avoidance，CSMA/CA）。无线LAN中使用的CSMA/CA并不依赖于所有计算机都能接收全部传输，而是在发送一个分组之前先从预期的接收方触发一个很短的传输过程。其思想是，如果发送方和接收方都发送一个报文，那么处在这两台计算机任何一台范围内的所有其他计算机都将知道一个分组的传输即将开始。图14-7说明了这个顺序过程。

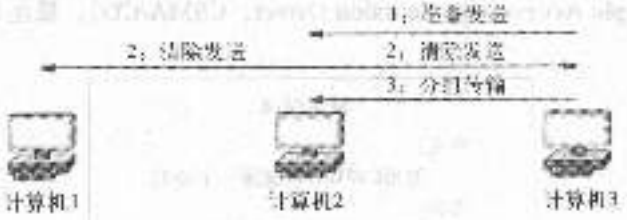


图14-7　计算机3向计算机2发送分组时发出的一连串报文

在图中，计算机3发送一个短的报文宣告它准备向计算机2发送一个分组，而计算机2也发送一个短的报文作为响应，宣告它已经做好接收分组的准备。在计算机3范围内的所有计算机将收到初始宣告，而计算机2范围内的所有计算机会接收响应报文。这样，即使计算机1不能

收到信号或侦听到载波，它也知道一个分组传输过程即将发生。

使用CSMA/CA的时候控制报文间也会发生冲突，但却很容易处理这种问题。举例来说，在上图中，如果计算机1和计算机3试图同时向计算机2发送一个分组，它们的控制报文将会出现冲突。计算机2将检测到这种冲突，并不做出回应。当这种冲突发生时，发送站点应用随机退避算法，然后重发控制报文。因为控制报文比分组要短得多，所以发生第二次冲突的概率也小了很多。最终，两个控制报文中总有一个能正确到达，接着计算机2发送一个响应报文。

概括如下：

> 由于无线LAN中计算机的距离跨度太于信号的传播范围，因此无线LAN采用了CSMA/CA技术。发送计算机与接收计算机在分组传输开始前都会发送一个控制报文。

14.7 本章小结

IEEE MAC层包括控制接入共享介质的协议。信道分配协议由时分、频分和码分多路复用技术扩展而来，这些扩展技术分别称为时分、频分和码分多址接入。静态或动态的信道分配方案都是可能的。

受控接入协议允许独立的站点加入到统计多路复用中去。轮询技术使用一个中心控制器来反复检查是否有站点准备发送分组，预约系统经常与卫星通信一起使用，它要求站点宣布它们是否准备在下一轮传输中发送分组。令牌传递经常与环形拓扑一起使用，它在站点间传递一个控制报文，收到令牌的站点可以发送一个分组。

随机接入协议允许站点竞争接入权。历史上著名的ALOHA协议使用两种频率，一个用于上行传输，另一个用于下行传输。如果一个站点没有收到它之前发送出去的分组的副本，它就会重传这个分组。以太网使用带冲突检测的载波侦听多址接入（CSMA/CD）技术来接入共享介质。该协议能防止某个站点在已正在传输进行过程中发送分组，此外它还利用二进制指数退避算法来从冲突中恢复传输。

由于无线LAN中存在隐蔽站点的问题，所以它要采用避免冲突的载波侦听多址接入（CSMA/CA）技术。在一台计算机向另外一台计算机发送分组之前，这两台计算机都要发送一个短的控制报文，从而使处于这两台计算机范围内的所有计算机都知道有一个传输即将发生，这样就可避免冲突。

练习题

14.1 试解释用于任意接入共享介质的3种基本方法。

14.2 试举出一个使用动态信道分配方案的网络例子。

14.3 列出信道分配协议的3种主要类型，并说出各类型的特点。

14.4 试解释轮询的含义并说出两种通用的轮询策略。

14.5 在一个预约系统中，控制器如何形成一个将在指定轮次中发送分组的站点列表？

14.6 什么是令牌？它是如何控制网络接入的？

14.7 在Aloha协议中，如果两个站点尝试同时在上行频率上发送分组，会发生什么现象？如何解决这个问题？

14.8 写出缩写CSMA/CD的全称，并解释每一部分的含义。

14.9 什么是二进制指数退避算法？

14.10 CSMA/CD为什么要使用一个随机延迟（提示：想象网络上有很多同类的计算机）？

14.11 为什么在无线网络上需要采用CSMA/CA？

第15章 有线局域网技术

15.1 引言

本书这一部分的章节介绍分组交换网络技术。第13章介绍了LAN中使用的IEEE 802模型，以及如何将第二层划分为逻辑链路子层和MAC子层，还讨论了48位的编址方案，它是构成逻辑链路子层的一个重要部分。第14章（原书"13"有错）专门介绍了MAC子层，考虑了包括CSMA/CD在内的介质访问协议。

本章继续讨论局域网，重点关注有线LAN技术。这一章将说明如何根据前一章的概念形成以太网的基础，而以太网则是在所有网络中占支配地位的一种有线LAN技术。

15.2 最早的以太网

回顾第14章所讲的，以太网是一种LAN技术，最早在施乐公司的PARC发明出来，后来由数字设备公司、因特尔公司和施乐公司进行标准化。这种最早的以太网[⊖]已经存在了30年。虽然以太网中使用的硬件设备、电缆和介质已经发生了显著的变化，但很多基本原理却一直保持不变。以太网在演进过程中最引人注目的地方，是关于新版本保持向后兼容的方式问题——新版本能识别老版本，并能自动适应老的技术。

15.3 以太网帧格式

术语帧格式（frame format）指的是对分组进行组装的方式，包括帧长度和每个域的含义这些细节。以太网的老版本之所以能与新版本保持兼容的主要原因在于帧格式。它自从20世纪70年代DIX标准被创建后就一直保持不变。图15-1所示为帧的基本格式和帧头部的细节。

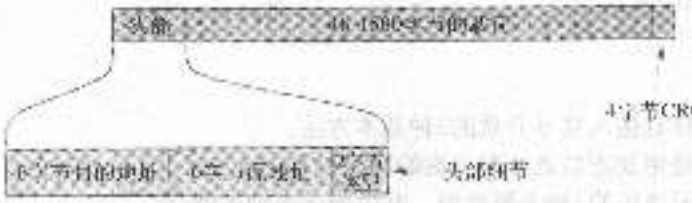


图15-1 以太网帧格式和头部细节的示意图

如图所示，以太网帧由固定长度的头部、可变长度的载荷以及固定长度的循环冗余验校码（cyclic redundancy check）[⊖]构成。头部包含3个域：48位的目的地址域（它指出期望接收方的地址）、48位的源地址域（它包含发送该帧的计算机的地址），以及一个16位的类型域。

⊖ 在后文中也称"传统以太网"、"原始以太网"或"第一代以太网"——译者注。

⊖ 当以太网帧在网络上发送时，使用第6章描述的曼彻斯特编码方式对码位进行编码，并且可能在该前面加上由交替的1和0组成的64位前导序列。

15.4 以太网的类型域

以太网帧中的类型域提供了复用与分用功能，以使允许在一台指定的计算机上同时运行多种协议。例如，后面的章节将会讲到，在因特网上使用的协议要通过以太网来发送IP数据报和ARP报文。这两种报文都被分配了一个唯一的以太网类型值（IP数据报的十六进制值为0800，ARP报文的十六进制值为0806）。在以太网帧中传输一个数据报的时候，发送方为类型域分配值0800。当帧到达它的目的地时，接收方检查该类型域，并使用这个值来确定哪个软件模块应该处理这个帧。图15-2说明了分用的过程。

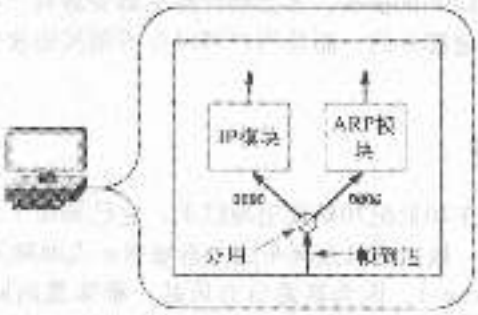


图15-2 利用帧类型域进行分用的示意图

15.5 以太网的IEEE版本

有趣的是，在1983年IEEE开发了一个以太网标准并试图重新定义以太网的帧格式[1]。制订这个标准的IEEE工作组编号是802.3，为了将这个IEEE标准与其他标准区分开来，专业人员通常将它称为802.3以太网。

传统以太网与802.3以太网的主要差异在于对类型域的解释。802.3标准将原来的类型域解释为分组长度，并增加了一个额外的8字节头部，其中包含了分组的类型。这个额外的头部称为逻辑链路控制/子网附着点（Logical Link Control/Sub Network Attachment Point，LLC/SNAP）头部，大多数专业人员将其简称为SNAP头部。图15-3说明了它的格式。

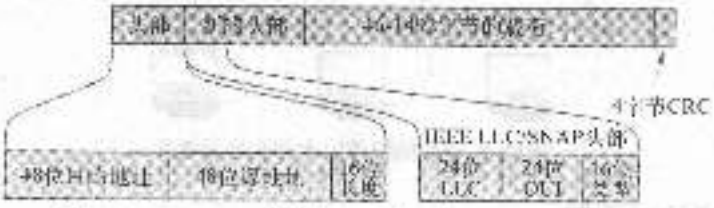


图15-3 IEEE 802.3帧格式，它有一个LLC/SNAP头部

如图所示，802.3以太网中整个帧的长度保持着和传统以太网同样的长度：1514字节。因此，IEEE将最大载荷长度从1500字节缩短到了1492字节，我们可以认为一个SNAP头部占用了载荷区域的前8字节。

为了保持这两个以太网版本的兼容性，我们使用如下约定：

> 如果以太网帧中的第13、14字节包含的数值小于1500，那么这个域可以解释为分组的长度并适用802.3标准；否则，该域被解释为一个类型域并适用原始以太网标准。

[1] 以太网的IEEE版本并不总是成功。大多数的网络实例依然使用并兼容原始的帧格式。

15.6　LAN连接和网络接口卡

从计算机体系结构上来说，LAN看起来像是一种I/O设备，并以与磁盘或视频设备相同的方式连接到计算机。也就是说，它是通过一块网络接口卡[⊖]（Network Interface Card，NIC）插到计算机的总线上的。从逻辑上讲，网络接口卡处理地址识别、CRC计算和帧的识别（例如，NIC检查帧中的目的地址，并忽略那些不是发给本机的帧）。此外，网络接口卡将计算机连接到网络，并处理数据通信的细节问题（即发送和接收帧）。从物理上讲，一块网络接口卡由一块电路板构成，电路板的一侧有一个插头，它正好与计算机的总线相配；另一侧有一个连接器，能适配于某种指定LAN的插头。大多数计算机都安装有一块网络接口卡，但网络接口卡又是独立于计算机的其他部分的，而且用户可以在不做其他改变的情况下选择替换这块网络接口卡。

15.7　粗缆布线的以太网

自从以太网最初的版本在20世纪70年代出现以来，它已历经了几种大的变化，最明显的变化体现在介质和布线上面。最初的以太网布线方案被非正式地称为粗缆以太网（thick wire Ethernet）或粗缆网（thicknet），因为其通信介质是一根笨重的同轴电缆，其正式术语是10Base5。粗网使用的硬件分成两个主要的部分。NIC处理通信的数字方面。一种叫做收发器（transceiver）的独立电子设备连接在以太网电缆上，并处理载波信号检测，把位串转换成适合传输的相应的电平，把传入的信号转换成位串。

一种称为附属单元接口（Attachment Unit Interface，AUI）的物理电缆把收发器连接到计算机的NIC上。收发器通常离计算机比较远，例如，在一间办公大楼内，收发器可能会连接在走廊天花板上的以太网上。图15-4说明了最初的粗缆布线是如何用一根AUI电缆将计算机连接到收发器上的。

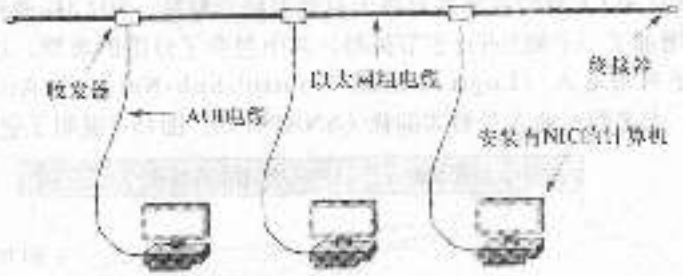


图15-4　原始的粗缆以太网布线示意图

15.8　细缆布线的以太网

第二代以太网布线系统采用比粗网更柔软的细同轴电缆。其正式名称为10Base2，非正式名称为细缆以太网（thin wire Ethernet）或细网（thinnet），这种布线方案与粗网有明显的不同。细网不必使用AUI电缆来连接计算机与收发器，而是将收发器直接集成到了网络接口卡中，并用一条同轴电缆从一台计算机连接到另一台计算机。图15-5说明了细网的布线。

细网既有优点也有缺点。其主要优点是较低的总费用和容易安装。它不需要外部收发器，并且细网电缆可以安装在方便的通道上（例如，通过计算机之间的桌面上、地板下面或管道

⊖　严格来讲，该设备是一种网络接口控制器（network interface controller）。

里面）。主要缺点在于整个网络容易出故障——如果用户拔掉网络的一段进行重新布线或移走一台计算机，那么整个网络就会停止工作[⊖]。

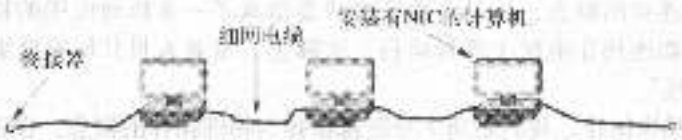


图15.5 被称为细同的第二代以太网布线示意图

15.9 双绞线布线的以太网和集线器

第三代以太网布线系统在两个方面做了很大的改变。

- 第三代系统使用一个中央电子设备来取代同轴电缆，它将连接到网络上的计算机分隔开来。
- 第三代系统采用双绞线来取代容量的哥蒂缆线。

由于不使用同轴电缆，第三代技术俗称为双绞线以太网（twisted pair Ethernet），并且已经取代了其他的版本。因此，现在某人说以太网的时候，都是指双绞线以太网。

对于双绞线以太网的最初版本而言，那个作为中心互连设备使用的电子设备被称为集线器（hub）。集线器有多种型号，其费用也与其型号的规模大小成正比。小的集线器有4个端口或8个端口，每个端口可连接一台计算机或其他设备（例如打印机）。较大的集线器可以容纳几百个连接。图15-6说明了这种布线方案。

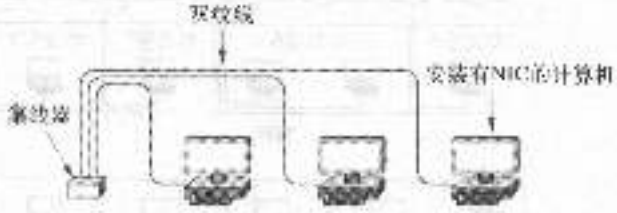


图15-6 使用双绞线的第三代以太网示意图

集线器中的电子部件仿真了物理电缆的特性，使整个系统能像传统以太网那样运行。例如，连接在集线器上的计算机采用CSMA/CD协议来接入网络，接收每个帧的副本，并利用帧中的地址来决定是否处理还是忽略收到的帧。而且，双绞线以太网保留了与以前版本同样的帧格式。实际上，计算机上的软件并不区分粗缆以太网、细缆以太网和双绞线以太网——都由计算机上的网络接口去处理所有细节和隐蔽任何差异。

要点 双绞线以太网布线方案使用一种叫做集线器的电子设备来取代共享电缆。

15.10 以太网的物理和逻辑拓扑

回顾一下，LAN是根据它们的拓扑（即整体形状）来分类的。图13-7归纳了一些主要的拓扑形式[⊙]。问题是："以太网采用的是什么拓扑呢？"出人意料的是，问题的答案有些复杂。

⊖ 原书这句话表达不太准确，移走一台计算机并不会使网络出故障，应该表述为"如果用户拔掉网络的任一段布线，或者任何一个终结连接器接触不好，那么整个网络就会停止工作。"——译者注

⊙ 见13-7小在第13章的13.8节找到。

很明显，以太网最初的粗网版本是一种总线拓扑。的确，最初的以太网作为总线拓扑的一个经典例子经常被引用。双绞线以太网看起来是一种星形拓扑，实际上，术语集线器的出现是要澄清中心互连点的概念。但是，由于集线器仿真了一条物理电缆的特性，使系统看起来好像所有计算机都连接在电缆上那样运行。实际上，专业人员开玩笑地讲集线器其实提供了一条"盒中的总线"。

为了理解以太网的拓扑，我们必须区分逻辑拓扑与物理拓扑的概念。在逻辑上，双绞线以太网采用了总线型拓扑，但在物理上，双绞线以太网形成了一种星形化（star-shaped）拓扑。

要点　正确地区分逻辑拓扑与物理拓扑使我们领会到：双绞线以太网采用的是一种星形物理拓扑，但在逻辑上却是起到总线的作用。

15.11　办公大楼内的布线

LAN中采用的布线类型在机房或实验室里没有太多差异。但是，当用在办公大楼时，根据所需要的线缆类型、数量、所跨的距离和费用，布线类型会有明显的差异。以太网布线的3个版本说明了LAN采用的3种主要形式。图15-7描绘出一幢办公大楼的层楼布线情况。

在图中，注意双绞线以太网需要在办公室和中心节点之间走很多单独的线，我们称中心节点为配线盒（wiring closet）。因此，双绞线以太网需要仔细地标注每一根线。

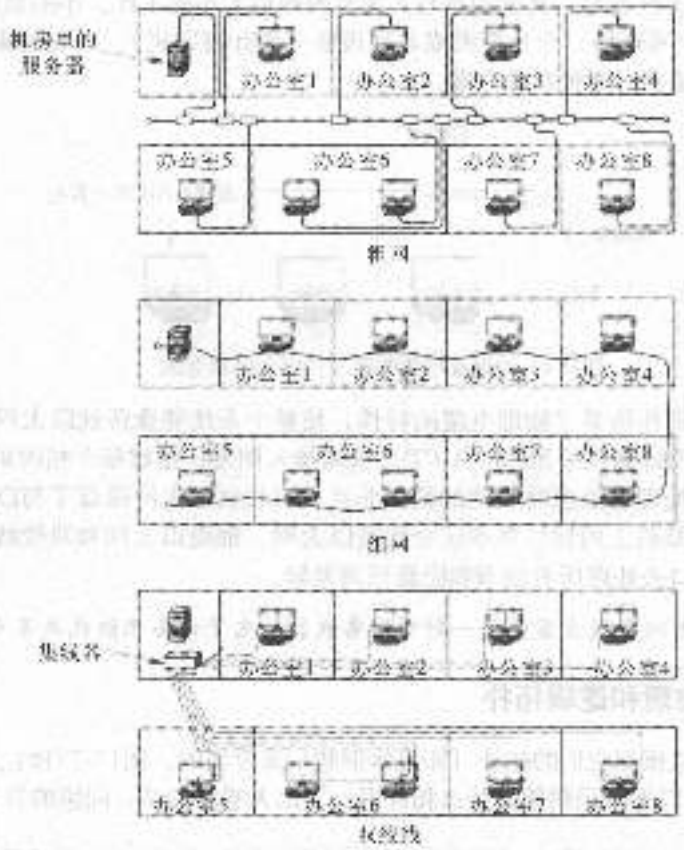


图15-7　一幢办公楼内采用的各种LAN布线方案示意图

15.12 双绞线以太网的变种及其速率

自从双绞线以太网首次出现以来，人们在双绞线的质量和导线性能方面就做了很多明显的改进。因此，双绞线上的数据速率大大提高。图15-8归纳了3种类型的双绞线以太网以及它们所使用的电缆。

称 呼	名 字	数据速率	使用的电缆
10BaseT	双绞线以太网	10Mbit/s	5类线
100BaseT	快速以太网	100Mbit/s	超5类线
1000BaseT	千兆以太网	1Gbit/s	6类线

图15-8 3种类型的双绞线以太网及各自使用的电缆

如图所示，双绞线以太网的第一版被给予了一个正式称呼10BaseT，其中数值10指明速率是10Mbit/s。后一个版本引入的时候起名为快速以太网（Fast Ethernet），运行在100Mbit/s的速率上，也被给予了一个正式称呼100BaseT。第三个版本叫做千兆以太网（Gigabit Ethernet），运行在1Gbit/s（即1000M bit/s）的速率上，专业人员经常将其名称简写为Gig-E。第17章将解释更高速率的以太网技术，它采用一种叫做交换机（switch）的电子设备，而不再使用集线器。另外，为了保持向后兼容性，更高速率版本的以太网标准规定接口必须能自动检测到能使这个连接正常工作的速率，也能降低速率以适应旧的接口设备。因此，如果我们将一根以太网电缆插在一个采用10BaseT的旧设备和一个采用1000BaseT的新设备之间，这个新设备会自动感知到这种差异，并把速率降为10Mbit/s。

15.13 双绞线连接器与缆线

双绞线以太网使用RJ45连接头，它是用来连接电话的RJ11连接头的更大号版本。一个RJ45连接头仅能以一种方式插入到插座中，一个物理小片会将连接头卡在正确的位置。因此，当连接头在插入的方法不对时是插不进去的，而一旦插好了，它就不会掉出来。

人们可以购买各种长度的缆线，并且每一端都安装了RJ45连接头。这也意味着大多数用户并不需要自己去制作缆线。然而，由于有两种类型的缆线：直连的（straight）和交叉的（crossed），因此造成了一些混淆。交叉线用来连接两台交换机，它将缆线一端的连接头上的针连接到另一端连接头的不同针上；直连线用来连接计算机与交换机，它将缆线一端的连接头上的针连接到另一端连接头对应的针上。因此，针1连接着针1，针2连接着针2，依此类推。尽管非常精密的接口硬件能检测出错误的缆线并能自动适应，但是大多数硬件在需要直连线的情况下使用了交叉线时是不能正常运行的。

为了帮助技术人员进行正确的连接，5类线或6类线中的单根电线都用有颜色的塑料包裹。图15-9列出了直连线使用的颜色码[⊖]。

RJ45针	线使用的颜色	功能
1	绿/白	TX_D1+
2	绿	TX_D1-
3	橙/白	RX_D2+
4	蓝	BI_D3+
5	蓝/白	BI_D3-
6	橙	RX_D2-
7	棕/白	BI_D4+
8	棕	BI_D4-

图15-9 RJ45连接头中使用的颜色码列表

⊖ 图中第3列的缩写字母把每根针分别标记为被用于发送还是接收，或者是作为双向通信的4个可能方案之一。

15.14 本章小结

以太网技术首创于20世纪70年代，已经成为有线局域网的事实标准。一个以太网帧以一个14字节的头部开始，其中包含48位的目的地址、48（原书“8”有误）位的源地址和16位的类型域。尽管IEEE 802.3标准尝试定义一种新的有额外8字节的帧格式，但是IEEE版本的帧格式很少使用。

以太网帧到达它的目的地后，接收方利用类型域的值来确定上层协议（即分用功能）。发送方在生成一个帧时会指定帧的类型，接收方利用帧类型来确定哪个模块应该处理这个帧。

虽然以太网的帧格式从它的第一个标准开始就一直保持不变，但是以太网使用的缆线类型以及布线方案却产生了显著的变化。目前已有3种主要的版本用于以太网布线。粗网使用大直径的同轴电缆，它的收发器与计算机相分离。细网使用柔软的细同轴电缆，它从一台计算机蜿到另一台计算机，并且每台计算机的网络接口上包含有一个收发器。双绞线以太网用一种叫做集线器或交换机的电子设备取代了同轴电缆，且在计算机与集线器（交换机）之间使用双绞线。所形成的系统拥有物理星形拓扑和逻辑总线拓扑这两种构型特征。

像以太网的早期版本那样，最初的双绞线技术运行在10Mbit/s的速率上，并被称为10BaseT，一个正式称为100BaseT的以太网版本运行在100Mbit/s的速率上，它在商业上叫做快速以太网（Fast Ethernet）。以太网的第二个版本叫做千兆以太网（Gigabit Ethernet或Gig-E），运行在1000Mbit/s（等于1Gbit/s）的速率上。当低速设备连接到以太网时，高速以太网上的硬件会自动感知到，并相应地降低速率。

练习题

15.1 包括CRC在内，以太网帧最长有多长？

15.2 如何使用以太网头部中的类型域？

15.3 在一个802.3以太网帧中，最大的载荷长度是多少？

15.4 接收方如何知道一个以太网帧使用的是802.3标准？

15.5 在使用LLC/SNAP头部的时候，它被放置在哪里？

15.6 一台计算机如何连接到粗缆以太网？

15.7 计算机如何连接到细缆以太网？

15.8 以太网集线器是什么？集线器中使用什么样的缆线？

15.9 在网站上查阅有关交换机和集线器的资料，如果给你提供一个运行在同样的位速率且价钱相同的交换机或集线器，你会选择哪一个？为什么？

15.10 试举一个物理拓扑和逻辑拓扑不同的网络的例子。

15.11 在一幢办公大楼内，哪一种以太网布线方案需要更多的物理连线？

15.12 10Mbit/s网络需要哪一类双绞线？100Mbit/s呢？1000Mbit/s呢？

第16章 无线联网技术

16.1 引言

本书的这一部分重点讲述联网技术及其在分组交换系统中的应用。各章分别介绍分组交换技术并给出IEEE模型。前一章已经阐述了在局域网中使用的有线联网技术。

本章介绍无线联网技术。要阐述的内容包括：当前已经提出的多种无线联网技术；无线通信可被应用于各种距离范围内；目前已存在多种商业化的无线通信系统。因此，与有线联网只有单一技术占主导地位的情形不同，无线联网出现了多样化的技术，其中很多技术具有类似的特征。

16.2 无线网络的分类

无线通信可应用于多种网络类型和规模。政府规定特定范围的电磁频谱可用于通信，这是无线通信出现多样性的部分诱因。在部分频谱中操作传输设备需要许可证，而在其他部分则不需要。人们已经开发出多种无线技术，同时新的变种也不断出现。无线技术大体上可以根据网络的类型进行分类，正如图16-1所示。



图16-1 无线联网技术的分类

16.3 个域网

除了在第13章中介绍的3种网络类型（LAN、MAN和WAN）外，无线网络还包括个人区域网（Personal Area Networks，PAN），或简称“个域网”。PAN技术提供短距离的通信，设计用于单个用户拥有和操作的设备。例如，PAN技术可以提供无线耳机与手机之间通信，也可以在计算机与邻近的无线鼠标或键盘之间使用。

PAN技术可以分成三大类，图16-2列出了这些分类，并给出每一类的简要说明。后续几节将更详细地介绍PAN通信，并列出PAN的标准。

类 型	用 途
蓝牙	小型外围设备（例如耳机或鼠标）与系统（例如手机或计算机）之间的短距离通信
红外	小型设备（经常关于特控制器）与邻近系统（例如计算机或娱乐中心）之间的视线通信
ISM无线	使用为工业、科学和医疗设备预留的频率进行短通信，此频率易被可能有干扰

图16-2 无线个域网技术的3种基本类型

16.4 LAN和PAN使用的ISM无线频带

政府已经为工业（industrial）、科学（scientific）和医疗（medical）组预留了3个电磁频谱区域，称为ISM无线（ISM wireless）。其频率没有授权给特定的运营商，各种产品大多都可使用这些频率段，主要用于LAN和PAN。图16-3表示了ISM的频段范围。

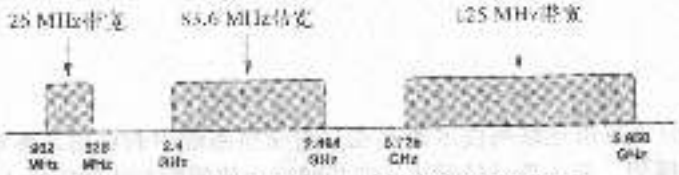


图16-3　构成ISM频带及其带宽的各个频率段

16.5 无线LAN技术与Wi-Fi

目前有各种各样的无线LAN技术，它们使用各种不同的频率，调制技术和数据传输速率。IEEE提供了大部分的标准。这些标准归类为IEEE802.11标准。1999年，一群制造无线设备的供应商形成了Wi-Fi联盟，这是一个非营利性组织，该组织使用802.11标准对无线设备进行测试和认证。由于该联盟已经占领了广泛的市场，所以大多数客户把无线LAN和术语Wi-Fi[⊖]联系起来。图16-4列出了被归入Wi-Fi联盟的主要IEEE标准。

IEEE标准	频　段	数据速率	调制技术	复用技术
原始802.11	2.4 GHz	1或2 Mbit/s	FSK	DSSS
	2.4 GHz	1或2 Mbit/s	FSK	FHSS
	红外线	1或2 Mbit/s	PPM	无
802.11a	5.725 GHz	6～54 Mbit/s	PSK或QAM	OFDM
802.11b	2.4 GHz	5.5和11 Mbit/s	PSK	DSSS
802.11g	2.4 GHz	22和54 Mbit/s	可变	OFDM

图16-4　Wi-Fi联盟认证的主要无线标准

16.6 扩频技术

第11章介绍了扩频（spread spectrum）这个术语，并解释扩频传输使用多种频率来发送数据。也就是说，发送者通过多种频率传输数据，接收者把从多个频率上接收到的信息组合起来，从而再生出原始数据。

通常，扩频可以用来实现下列两个目标之一：

- 提高总体性能。
- 使传输更能抵抗噪声干扰。

在图16-5的表中，总结了在Wi-Fi无线网络中使用的3种关键的复用技术。

每种技术都有其优点。OFDM技术提供了最大的灵活性；DSSS技术具有良好的性能；而FHSS技术使得传输能更好的抗噪声干扰。因此，定义一种无线技术时，设计者会选用一种合适的多路复用技术。例如，为了适应DSSS技术和FHSS技术，人们制定了两个版本的原始802.11标准。概括如下：

⊖　虽然最初出现在联盟广告中的是短语"无线保真"（wireless fidelity），但后来该联盟放弃了使用这个短语，并且不对Wi-Fi这个名字进行任何解释。

扩频技术有助于无线LAN在嘈杂的环境中发挥它的性能。

名 字	扩 展	描 述
DSSS	直接序列扩频	与CDMA类似，发送器把输出数据具以一个序列从而形成多个频率，接收器聚以相同序列实现解码
FHSS	跳频	发送器使用一序列频率传输数据，接收器使用同样序列的频率提取数据
OFDM	正交频分复用	传输波段被复分为多个载波，以使得载波之间互不干扰的一种频分复用方案

图16.5 用于Wi-Fi的主要多路复用技术

16.7 其他无线LAN标准

IEEE已经制定了许多无线联网标准，可以处理各种类型的通信。每种标准规定了频率范围、使用的调制、复用技术及数据传输率。图16-6列出了已经制定或提议的主要标准，并对每个标准作了简要说明。

标 准	用 途
802.11e	提高服务质量，例如，保证信息抖动
802.11h	与802.11a类似，不过增加了对频谱和功率的控制（主要设计在欧洲使用）
802.11i	提高安全性，包含高级加密标准，其完整版称为WPA2
802.11k	将提供无线电资源管理，包括传输功率
802.11n	数据速率超过100Mbit/s以处理多媒体（视频）应用（可以高达500Mbit/s）
802.11p	高速公路的车辆之间以及车辆与路边的专用短程通信（Dedicated Short-Range Communication，DSRC）
802.11r	改进漫游能力，使得在接入点之间切换而不会丢失连接
802.11s	一种建议的网状形网络，其中的一组节点能自动形成网络并在其中传递分组

图16-6 主要的802.11标准及其各自的用途

2007年，IEEE将许多已有的802.11标准整理成一个单一的文档即802.11-2007。该文档描述了整套标准的基础性内容，并为每个变体标准提供了一个附录。

要点 802.11标准的很多变体已经被制定或提议，每种变体都提供了各自的某些优点。

16.8 无线LAN体系结构

无线LAN的3个构件是：接入点（非正式也叫基站）、互连机构（例如用于连接接入点的交换机或路由器）、一组无线主机（也叫无线节点或无线站点）。从原理上，有可能实现两种类型的无线LAN：

- 专门构建型（Ad hoc）——无线主机之间相互通信，不使用基站。
- 基础结构型（infrastructure）——一台无线主机只与一个接入点通信，由接入点转发所有分组。

实际中，很少有Ad hoc网络，更多的是基础结构型网络，即一个组织或服务提供商部署一组接入点作为网络基础结构，然后每台无线主机通过其中的一个接入点进行通信。例如，一家私营公司或一所大学可能在其所有的建筑物中都部署接入点。图16-7所示为这种体系结构。



图16-7　无线LAN的基础体系结构示意图

从互连设备延伸到各个接入点的有线连接，通常由双绞线以太网构成。在某一给定接入点范围内的计算机集合，被叫做基本服务组[⊖]（Basic Service Set，BSS）。在上图中，存在3个基本服务组，每个组对应一个接入点。

概括如下：

大部分无线LAN使用基础结构型体系结构。在这种体系结构中，无线计算机通过一个接入点（基站）进行通信。

16.9　重叠、关联和802.11帧格式

在实际中，有很多细节问题使得基础结构型的体系结构变得很复杂。一方面，如果有一对接入点相距太远，这两者之间将会出现一个无信号的盲区（dead zone，即没有任何无线连接的物理位置）。另一方面，如果有一对接入点相距太近，将会出现一个重叠区，在这个区域中，一台无线主机可以同时触及到两个接入点。此外，大部分无线LAN都连接到Internet，因此，互连机构往往有一个附加连接到路由器，然后由它再连接到Internet。图16-8说明了这个体系结构。

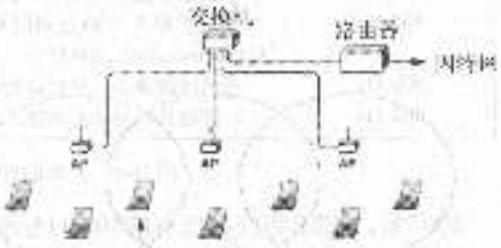


图16-8　有重叠区域的基础结构示意图

为了解决重叠的问题，802.11网络要求一台无线主机只与单一接入点相关联（associate），也就是说，一台无线主机发送帧到一个特定的接入点，然后由该接入点通过网络转发这些帧。图16-9给出了802.11帧格式。如图所示，当使用基础结构型体系结构时，帧同时携带了接入点的MAC地址和Internet路由器的MAC地址。

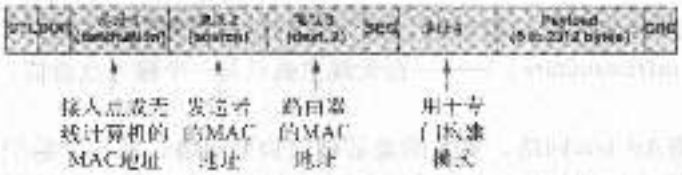


图16-9　802.11无线LAN中使用的帧格式

⊖　类似于蜂窝电话系统，一个给定接入点可达到区域被非正式地叫做“蜂窝”（cell）。

16.10 接入点之间的协调

这里出现了一个有趣的问题：接入点之间需要协调至什么程度？有许多早期的接入点其设计都比较复杂。接入点要互相协调，才能提供类似蜂窝电话系统那样的无缝移动性。也就是说，接入点之间相互通信，以确保无线计算机从一个接入点的区域移动到另外一个接入点区域时能够平滑切换。例如，在一些设计中测量信号的强度，当一个无线节点从新的接入点处收到的信号比从原来接入点处收到的信号强时，就把这个无线节点切换到新的接入点上。

作为一种替代的方法，一些供应商开始提供较低成本、较低复杂性的接入点，这些接入点之间没有协调。供应商们提出一些理由：信号强度不能为移动性提供一个有效的度量标准；移动计算机自身能够处理从一个接入点到另一个接入点的变化；而且，连接接入点的有线基础设施具有足够的容量，从而允许做更集中化的协调。较低复杂性的接入点设计，对于安装只包含单一接入点的情况，尤为合适。

概括如下：

> 存在两种基本方法：复杂的接入点之间通过协调来确保平滑切换；低成本的接入点则独立地运行（即互相不进行协调），并依赖无线计算机自身去完成从与一个接入点关联切换到与另外一个接入点关联。

16.11 竞争与无竞争接入

原始的802.11标准为信道接入定义了两个常规的方法，它们的特征分别是：

- 无竞争服务的点协调功能（Point Coordinated Function，PCF）。
- 基于竞争服务的分布协调功能（Distributed Coordinate Function，DCF）。

点协调服务意味着接入点对基本服务组内的站点实施控制，以确保传输不会相互干扰。例如，接入点可以为每个站点分配一个独立的频率。实际上，这种PCF从未被采用。

分布协调服务则是支持BSS中的每个站点各自运行随机接入协议。回顾一下，在第14章中我们提到，无线网络有隐藏站问题（hidden station problem），即两个站点之间可以通信，但第三个站点只能收到其中一个站点的信号。为了解决这个问题，802.11网络使用载波侦听多址接入/碰撞避免（Carrier Sense Multi-Access with Collision Avoidance，CSMA/CA）协议，该协议要求一对站点在传输分组之前交换一对准备发送（RTS）与清除发送（CTS）报文。802.11标准则将若干在第14章中遗漏的一些细节包括了进去。例如，该标准定义了如下3个定时参数：

- SIFS——10μs的短帧间间隔（Short Inter-Frame Space，SIFS）。
- DIFS——50μs的分布帧间间隔（Distributed Inter-Frame Space，DIFS）。
- 20μs的时隙。

直观地，SIFS参数定义接收站点在发送一个ACK或其他响应之前的等待时间；DIFS参数，等于SIFS加上两倍的时隙，定义一个站点在试图传输之前信道必须持续处于空闲状态的时间长度。图16.10说明了这些参数在分组传输中是如何使用的。

> 要点 在Wi-Fi网络中使用的CSMA/CA技术包含定时参数，这些参数规定了一个站点在发送一个初始分组之前要等待多长时间，以及一个站点在发送一个应答之前要等待多长时间。

站点之间的物理间距和电气噪声使得硬件难以区分微弱信号、干扰和碰撞。因此，Wi-Fi

网络没有采用碰撞检测。也就是说，硬件在传输期间并不去侦测干扰，而是由发送者等待一个确认（ACK）报文。如果没有收到ACK，发送者就假定该传输丢失了，并采用退避（backoff）策略，该策略与有线以太网使用的退避策略类似。在实际中，由于802.11网络的用户很少，而且不会遭受电气干扰，所以很少需要重传。然而，其他的802.11网络则会出现频繁丢失分组的情况，这时就要依赖重传了。

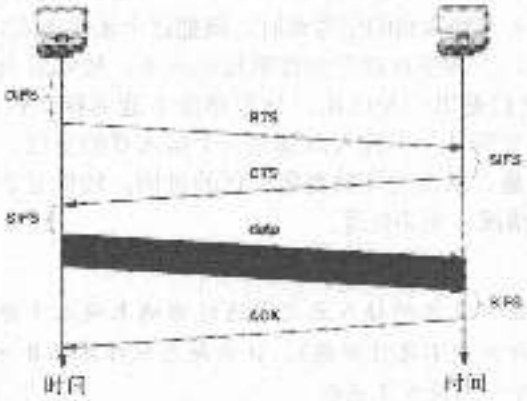


图16-10　使用了SIFS和DIFS规则的CSMA/CA示意图

16.12　无线MAN技术与WiMAX

总的来说，MAN技术还没有获得成功的商业化应用。有一种无线MAN技术脱颖而出，并有可能获得成功。该技术是由IEEE标准化，列在802.16大类之下。有一批公司联合为这项技术创造了一个术语WiMAX，解释为全球微波接入互操作性（World-wide Interoperability for Microwave Access，WiMAX），并成立了WiMAX论坛，以促进该技术的推广应用。

正在开发的WiMAX有两个主要版本，两者的整体方案并不相同。这两个方案普遍地被称为：

- 固定WiMAX，
- 移动WiMAX。

固定WiMAX是指基于IEEE 802.16-2004标准所建立的系统。非正式地，该标准也称为802.16d。词汇固定（fixed）是指该技术不提供接入点之间的切换。因此它被设计成为服务提供商和固定位置（诸如住宅或办公楼）的设备之间提供连接，而不是为供应商和诸如手机这类位置变化较为频繁的设备之间提供连接。

移动WiMAX是指基于802.16e-2005标准所建立的系统，非正式地，该标准也可缩写为802.16e。正如词汇移动（mobile）所指，该技术可以为移动设备提供接入点之间的切换，这意味着移动WiMAX系统可以与移动较为频繁的便携式装置（诸如笔记本电脑或手机）一起使用，为便携式装置提供灵活的接入服务。

WiMAX提供了可以以各种方式使用的宽带通信。一些服务提供商计划把WiMAX作为一种“跨越最后一英里的Internet接入”技术。其他服务提供商则看到了WiMAX在物理站点之间（特别是在一个城市范围内的站点之间）提供通用互连的潜力。另外还有一种互连类型，被称为回程连接（backhaul），它能在服务提供商的中央网络设施与远地站点（例如蜂窝基站塔）之间实现连接。图16-11列出了一些WiMAX已被提议的用途。

接 入
——作为替代DSL或电缆modem的最后一英里接入技术 ——移动用户的高速互连 ——对一台数据和电信接入 ——站点到网络间的备份连接
互 连
——从Wi-Fi接入点到服务者的回程连接 ——公司站点之间的专用连接 ——小型ISP与大型ISP之间的连接

图16-11 WiMAX技术的潜在用途

一般来说，用于回程链接的WiMAX部署将具有最高的数据速率，并将使用要求在两个通信站点之间有一条清晰视线（Line-Of-Sight，LOS）的频率。LOS站点通常安装在塔楼或者建筑物的顶端。尽管用于固特网接入的部署可以用固定或移动WiMAX，这样的部署通常使用不要求LOS的频率。因此，它们被归类为非视线（Non-Line-Of-Sight，NLOS）部署。图16-12表示了这两种部署方式。

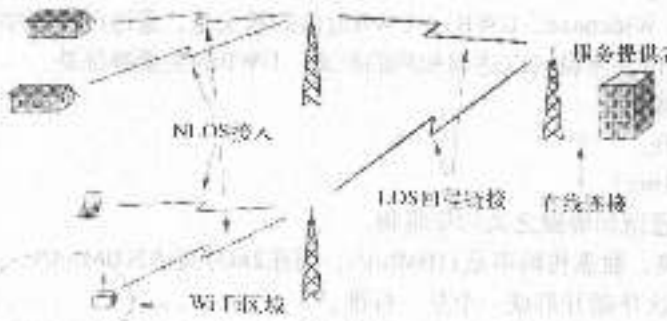


图16-12 用于接入和回程链接的WiMAX示意图

WiMAX的关键特点可归纳如下：

- 使用授权的频谱（即由运营商提供）。
- 每个基站可覆盖半径3～10km。
- 采用可伸缩的正交频分复用。
- 保证服务质量（对话音或视频）。
- 在短距离内每个方向可以传输70Mbit/s。
- 在长距离（10km）上提供10Mbit/s。

概括如下：

> WiMAX是一种无线LAN技术，可用于回程链接、固定或移动接入，用于接入的基站部署不要求有清晰的视线。

16.13 PAN技术与标准

IEEE把编号802.15分配给PAN标准，针对每个关键的PAN技术，已经成立了几个工作组和产业联盟。图16-13列出了主要的PAN标准。

标　准	用　途
802.15.1a	蓝牙技术（1Mbit/s，2.4GHz）
802.15.2	PAN之间的共存性（不互相干扰）
802.15.3	高速PAN（55Mbit/s，2.4GHz）
802.15.3a	超宽带（UWB）高速PAN（110Mbit/s，2.4GHz）
802.15.4	Zigbee技术——用于远程控制的低速率PAN
802.15.4a	小功率低速PAN的替代技术

图16.13　IEEE PAN标准

蓝牙（Bluetooth）。在供应商创造了蓝牙技术作为短距离的无线连接技术之后，这项技术演进为IEEE 802.15.1a标准。蓝牙技术的特征是：

- 无线代替电缆（例如，耳机或鼠标）。
- 使用2.4GHz频段。
- 短距离（最长5m，其变种的范围可扩大到10m或50m）。
- 有主设备和从设备。
- 主设备为从设备授予许可。
- 数据传输率可达721Kbit/s。

超宽带（Ultra Wideband，UWB）。UWB通信的深义是，通过很多频率来散播（spreading）数据，它要求较少的功率就可以达到相同的距离。UWB的主要特征是：

- 使用很宽的频谱。
- 非常低的功耗。
- 短距离（2~10m）。
- 信号可以穿透诸如墙壁之类的障碍物。
- 在10m的距离，数据传输率是110Mbit/s，而在2m时高达500Mbit/s。
- IEEE未能解决争端并形成一个单一标准。

Zigbee[⊖]，Zigbee标准（802.15.4）的出现是为了标准化无线远程控制技术，尤其是对工业设备的控制。因为远程控制单元只发送短命令，所以不要求高的数据传输率。Zigbee的主要特征是：

- 远程控制（不是数据）的无线标准。
- 目标是工业以及家庭自动化。
- 使用3个频段（868MHz、915MHz和2.4GHz）。
- 数据传输率有20、40或250Kbit/s，具体速率取决于频段。
- 低功耗。
- 要定义3个安全级别。

16.14　其他短距离通信技术

还有其他两种无线技术（即红外线技术和RFID技术），尽管它们通常没有被归入到无线PAN的类别中，但它们也可以提供短距离的通信。其中，红外技术提供控制和低速率数据通信，而RFID技术则被应用于传感器通信。

红外线（InfraRed）。红外线技术通常用于远程控制，并可作为绳线的替代物（例如，用

⊖ 目前对术语Zigbee尚无统一的中文译名，但偶见一些开发商使用"紫蜂"这一名字。——译者注

于无线鼠标与主机之间的通信)。红外数据协会(Infrared Data Association，IrDA)制定了一套被广泛接受的标准，IrDA技术的主要特征如下：

- 适于各种速度和用途的标准系列。
- 实际系统具有1米至几米的范围。
- 覆盖30°的圆锥体定向传输。
- 数据传输率在2.4Kbit/s(控制)至16Mbit/s(数据)之间。
- 使用低功率等级的电源，非常低的功率损耗。
- 信号可以从固体表面反射，但不能穿透它。

射频识别(Radio Frequency Identification，RFID)。RFID技术是一种有趣的无线通信形式，它创建一个含有标识信息的小标签(tag)的机制，接收器可以从标签中"找出"标识信息。

- 为了各种应用，当前已存在超过140个RFID标准。
- 无源的RFID从阅读器发送的信号中提取能量。
- 有源的RFID包含一个电池，可使用长达10年。
- 虽然有源RFID的范围比无源RFID扩展得更远，但距离还是很有限。
- 可用频率范围从低于100MHz至868-954MHz。
- 可用于库存控制、传感器、护照及其他应用领域。

16.15 无线WAN技术

无线WAN技术可以分成两大类：

- 蜂窝通信系统。
- 卫星通信系统。

蜂窝通信系统

蜂窝系统最初是设计用于为移动用户提供话音服务的，因此该系统设计成基站(其信号覆盖形成一个单元区)与公用电话网互连。目前，蜂窝通信系统越来越多地被用于提供数据服务和提供因特网连接。

就体系结构而言，每个单元区(cell)内含有一个基站塔，一群单元区(通常彼此相邻)的各个基站都被连接到一个移动交换中心(mobile switching center)，该中心跟踪移动用户，并负责管理用户从一个单元区向另一个单元区移动时的切换。图16-14说明了单元区(及其基站塔)在高速公路上的部署情况。

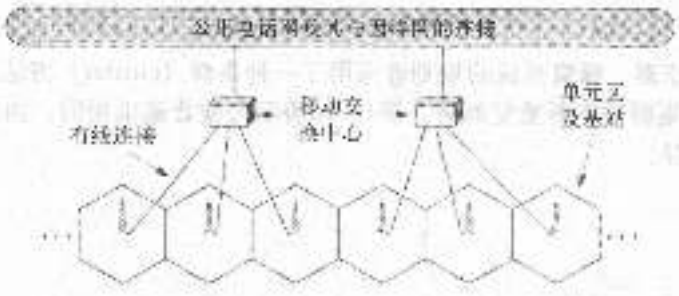


图16-14 蜂窝体系结构的示意性

当一个用户在连接到同一个移动交换中心的两个单元区之间移动时，则由这个移动交换中心来处理由此产生的接入切换。当一个用户通过一个地理区域到达另一个地理区域时，则

要由两个移动交换中心参与处理其接入切换。

在理论上，因为单元区可以被安排为一个蜂巢形状，所以如果每个单元区都形成等边六角形，那么就可以构成完美的蜂窝覆盖。在实际应用中，蜂窝系统的信号覆盖并不完美。大部分基站塔使用按圆形方向图发送信号的全向（omnidirectional）天线。而障碍物和电气干扰会使信号衰减，或造成信号覆盖形状不规则。因此，在有些情况下，基站的信号覆盖之间会彼此重叠，而在另一些情况下则存在没有信号覆盖的缝隙。图16-15说明了理想的和实际的信号覆盖情况。

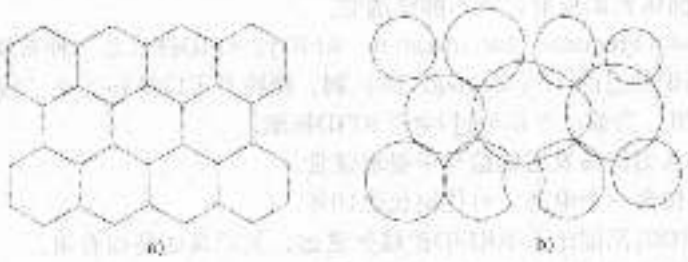


图16-15　a)理想化的蜂窝覆盖；b)实际情况中的信号覆盖重叠和间隙

蜂窝技术的另一个实际问题来源于基站密度的可变性。在农村地区，可预期的手机密度较低，而基站的信号覆盖可达范围较大，即单个基站就足以处理一个大的地理区域的需求。然而，在市区环境中，很多手机都集中在某一特定区域，例如，在一个位于大都市中的城区，除了行人和车辆上的人，还包括办公大楼或有许多住户的公寓大楼。为了处理更多的手机通信需要，设计师把这样一个区域划分成很多单元区。因此，与图16-15a中全部单元区都一样大的理想化结构不同，实际要部署的单元区是大小不同的，其中在大都市区域要使用覆盖范围较小单元区。

要点　虽然很容易把部署单元区可视化成为统一的蜂巢形状，但实际的系统却需要根据手机的密度来改变单元区的大小，而且障碍物也会导致信号覆盖不规则，因而造成单元区的重叠和间隙。

16.16　基站集群和频率重用

蜂窝通信遵循一个重要的原理：

如果相邻的一对基站不使用相同的频率，则可以使相互干扰达到最小化。

为了实现这个原理，蜂窝系统的规划者采用了一种集群（cluster）方法。根据这个方法，一个小格局的基站集群被不断重复部署。图16-16所示为被普遍采用的、由3、4、7和12个单元区构成的几种集群。

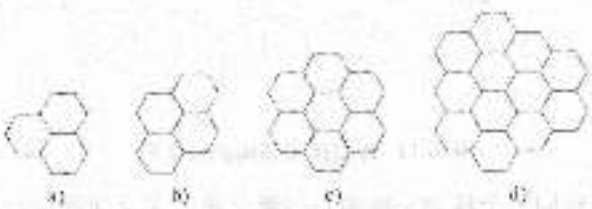


图16-16　典型的单元区集群示意图

根据几何原理，图中的每个形状都可以用来铺成一个平面。也就是说，通过复制相同的形状，有可能覆盖整个区域而不留任何间隙。此外，如果在给定基站集群中的每个单元区分配一个唯一的频率，那么重复相同的格局将不会使相同的频率分配给任何一对相邻的单元区。例如，图16-17展示了一个7-基站集群的复制，每个单元区用一个字母来表示其所分配到的频率。

在图中，每个字母对应一个特定的频率，集群中的每个基站都被分配一个频率，如图所示。当集群格局被复制时，就不会出现相邻基站共用同一个频率的情况。

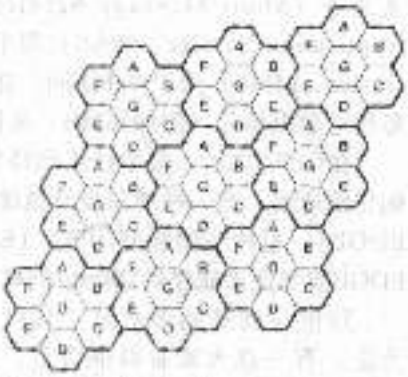


图16-17 当一个7-基站集群被复制时，其频率分配的示意图

16.17 蜂窝技术的更新换代

电信业界把蜂窝技术划分为4代，分别标记为1G、2G、3G和4G，中间版本标记为2.5G和3.5G，这几代技术可被表征如下：

- 1G，第一代蜂窝技术始于20世纪70年代后期，并延续至20世纪80年代末。该系统最初被称为蜂窝移动无线电话（cellular mobile radio telephones），使用模拟信号来承载话音。
- 2G和2.5G，第二代蜂窝技术始于20世纪90年代初，并延用至今。1G和2G之间的最主要区别是2G使用数字信号来传输话音。2.5G的标记则是所使用系统扩展2G系统使其包含了3G业务的一些特点。
- 3G和3.5G，第三代蜂窝技术始于21世纪，其专注于增加提供高速的数据服务。3G系统可以提供400Kbit/s－2Mbit/s的下载速率，目的是支持一些应用（如网页浏览和图片共享）。3G允许单个话机在北美、日本和欧洲之间漫游。
- 4G，第四代蜂窝技术始于2008年左右，其专注于对实时多媒体业务的支持（例如电视节目或高速视频下载）。此外，4G电话包含了多种连接技术（如Wi-Fi和卫星），在任何时候，话机都是自动地选择可用的最佳连接技术。

各种各样的蜂窝技术和标准也已经过不断地发展和演进。当2G出现时，很多团体试图各自选择一种技术并制定一个标准。欧洲邮电管理局会议（European Conference Of Postal and Telecommunications Administrators）选用了TDMA技术，称之为全球移动通信系统（Global System for Mobile Communications，GSM），并开发了一个有意作为全球标准的系统。在美国，每个运营商利用各自的技术建立一个网络。摩托罗拉公司发明了一种TDMA系统，称之为iDEN。大多数美国和亚洲的运营商采用一种被标准化为IS-95A的CDMA技术。日本创造了另一种TDMA技术，称为PDC。图16-18归纳了主要的2G标准及其演化出来的一些2.5G标准，图中未列举出来的其他各种技术则在发展过程中只起到次要作用。

方法	标 准	代 际
GSM	GSM	2G
	GPRS	2.5G
	EDGE(EGPRS)	2.5G
	EDGE Evolution	2.5G
	HSCSD	2.5G
CDMA	IS-95A	2G
	IS-95B	2.5G
TDMA	iDEN	2G
	IS-136	2G
	PDC	2G

图16-18 主要的第二代蜂窝技术

图中列举的每个标准都分别提供了一个基本的通信机制，通过这些机制很多服务都能运行。例如，通用分组无线业务（General Packet Radio Service，GPRS）可供有GSM或IS-136

接入的用户使用。一旦他（她）订阅GPRS，用户可以选择激活运行在GPRS上的业务。短消息业务（Short Message Service，SMS）用于简单文本业务；无线应用业务（Wireless Application Service，WAS）用于接入因特网；而多媒体消息业务（Multimedia Messaging service，MMS）用于Web访问。在通常情况下，服务供应商要对GPRS服务另行收费，通常按每传送数据单元（例如每MB）来计费。

在GPRS之后，数字技术已经发展得非常成熟，使用更加先进的调制和复用技术以增加数据传输速率。有一种称为数据速率GSM演进技术（Enhanced Data rate for GSM Evolution，EDGE），也称为增强型GPRS（EGPRS），提供了高达473.6Kbit/s的传输速率。后来出现的EDGE演变技术能提供1Mbit/s的峰值数据速率。

到供应商开始考虑第三代技术为止，有一点大家看得很清楚：客户需要能在全球范围工作的手机服务。因此，供应商推动技术的互操作性，而业界合并了2G中的很多技术从而形成几个关键的标准，IS-136、PDC、IS-95A和EDGE都影响了UMTS的设计（UMTS是一种使用宽频带的CDMA（WCDMA）技术）。与此同时，IS-95B被扩充成CDMA2000，如图16-19所示。

方　法	标　准	继 承 自
WCDMA	UMTS	IS-136、IS-95A、EDGE、PDC
	HSDPA	UMTS
CDMA2000	1xRTT	IS-95B
	EVDO	1xRTT
	EVDV	1xRTT

图16-19　第三代蜂窝技术

几个针对第三代数据服务的互相竞争的标准应运而生，EVDO和EVDV几乎同时出现。它们都结合了CDMA和频分复用技术，以提高峰值性能。EVDO既可以扩充为数据优化演进（Evolution Data Optimized），也可以扩充为数据演进（Evolution Data Only），成为最广泛部署的标准。根据数据传输速率的不同，EVDO可以分为两个版本：2.4Mbit/s和3.1Mbit/s，相对于EVDO的另一种可选标准，即高速下行链路分组接入（High-Speed Downlink Packet Access，HSDPA）提供了14Mbit/s的下载速度。当然，对于提供服务的数据传输率越高，运营商收取的费用也就越高。

16.18　VSAT卫星技术

第7章介绍了3种类型的通信卫星（即LEO、MEO和GEO），第14章讨论了信道接入机制，包括用来提供跨卫星的TDMA的预约机制。这一节通过描述具体的卫星技术来结束关于卫星方面的讨论。

卫星通信的关键是抛物面天线（俗称碟dish）的设计。抛物线的形状意味着来自远方卫星的电磁能量将被反射至一个焦点。通过瞄准一个卫星的碟面并在其焦点放一个探测器，设计者可以确保收到强的信号。图16-20所示为这个设计思想，它表示出入射能量是如何通过碟的表面反射而到达接收器的。

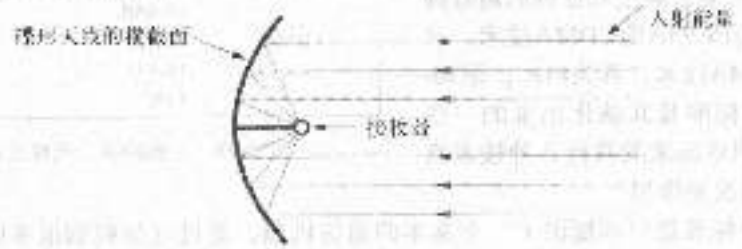


图16-20　抛物面碟形天线的反射示意图

为了最大化接收信号，早期的卫星通信使用地面站，这些地面站都装备了直径超过3m的大碟形天线。虽然这些大碟形天线在一些情况下是合适的，如电话公司所使用的跨大西洋连接，但是用户或者小型企业不可能在他们的场所放置这么大的地面站。因此，随着一种称为甚小口径终端（Very Small Aperture Terminal，VSAT）技术的出现，一个重大的变化随之而至——碟的半径缩小至3m以下。典型的VSAT天线的直径小于1m。

很多企业使用VSAT技术来连接他们所有的商店。例如，Walgreens和CVS等药房部署了VSAT通信，Pizza Hut和Taco Bell等快餐连锁店以及Wal Mart等零售商也都部署了VSAT通信。此外，VSAT服务可向用户提供娱乐和因特网接入。

VSAT卫星使用3个频率范围，它们在传输信号的强度、对雨以及其他大气条件的敏感度、在地球表面（称为卫星的足迹）的覆盖区域等诸多方面有所不同。图16-21描述了每个频带的特征。

频带	范围	足迹	信号强度	雨雪影响
C/Ku	3-7GHz	大	低	中等
Ku	10-18GHz	中等	中等	小
Ka	18-31GHz	小	高	严重

图16-21 VSAT技术使用的频带及其特征

16.19 GPS卫星

全球定位系统（Global Positioning System，GPS）中的卫星提供准确的时间和位置信息。尽管它不是计算机通信的一部分，但位置信息却越来越多地用于移动网络。其关键特性包括：

- 精确度在2～20m（军用版本具有更高的精确度）。
- 具有24颗卫星绕行地球轨道。
- 卫星被安排在6个轨道平面中。
- 提供用于一些通信网络的时间同步。

在某种意义上来说，获取位置信息的技术是直截了当的：因为所有的GPS卫星盘旋在众所周知的位置，一个接收器通过找出与3颗卫星的距离就可以确定在地球表面的确切位置。要理解其中的原因，可考虑与卫星1距离D_1的点集，这个集合定义了一个球面。同样地，与卫星2距离D_2的点集则定义了另一个球面。如果一个GPS系统同时与卫星1相距D_1，与卫星2相距D_2，则此GPS系统处在一个由上述两个球面相交组成的圆中。如果GPS系统也与卫星3相距D_3，该系统将会处在第三个球面与那个圆的交点中。这将出现两个可能的点。适当地安排卫星的位置，使得这两点只有其中一点落在地球表面上，另一点则处于宇宙空间中，因此很容易选出正确的点。

为了计算距离，GPS系统采用了牛顿物理学中的公式，该公式指明了距离等于速率乘以时间。速率是常量（即光速，3×10^8m/s）。时间是通过这样的计算得来的：安排每个GPS系统计算本地时间，并且每颗卫星都有一个精确的时钟，使得卫星可以在发送信息中包含时间戳（timestamp）。接收器以本地时间减去时间戳，就可确定信息传输所用的时间。

16.20 软件无线电和无线电的未来

本章中描述的各种无线技术都分别使用了专用的无线电硬件。在一个特定设备中的天线、发射器和接收器，都被设计成使用特定形式的调制和复用技术来操作预定的频率。可以使用

GSM、Wi-Fi和CDMA网络的电话，必须有3个完全独立的无线电系统，而且必须选择其中之一。

传统的无线电正在被具有可编程模式的无线电所取代，这种无线电的特征由运行在处理器上的软件控制。图16-22列出了可由软件可编程无线电控制的主要无线电特性。

特　性	描　述
频率	在特定时间所使用的确切频率集合
功率	发送器所发射的功率数量
调制	信号和信道编码及调制技术
复用	CDMA、TDMA、FDMA及其他复用方法的任意组合
信号方向	可调谐天线以接收特定方向的信号
MAC协议	有关或帧和MAC寻址的所有方面

图16-22　可编程无线电软件控制下的特性

使软件无线电变得可行的关键技术是：可调谐模拟滤波器和多天线管理。模拟芯片目前可提供可调谐模拟滤波器。因此，人们有可能选择频率和控制功率。数字信号处理器（Digital Signal Processors，DSP）可用于处理信号编码和调制。软件无线电更有趣的方面是涉及多天线的使用。与特定时间仅仅可以选择一个天线不同，软件无线电可以同时使用多天线来提供空间复用（spatial multiplexing）技术，这种技术允许在特定方向发射和接收信号。我们使用术语多输入多输出（Multiple-Input Multiple-Output，MIMO）来表示一个采用多个天线进行发射和接收（即可以瞄准发射或接收）的系统。

软件可编程无线电来自实验室研究，目前正准备被美国军方部署。此外，通用软件无线电外围设备（Universal Software Radio Peripheral，USRP）和GNU无线电（GNU Radio）目前也正处于实验研究之中。在可编程无线电大规模商业化之前，仍需要解决一些细节问题。首先，现行成本太高（接近1000美元）。其次，频谱的使用需要建立相关政策。事实上，当前传输电磁能量的设备都经过认证以确保不会和其他通信相互干扰（例如，手机不能干扰警方或紧急通信）。如果软件无线电可以被重编程，用户可能会在无意中下载病毒，这可能导致无线电阻塞紧急事务处理信道。因此，正在研究的一些技术，可以用来控制软件无线电在某些频率上能够产生的功率总量。

16.21　本章小结

目前已经有很多无线通信技术，并被用于构建无线LAN、PAN、MAN和WAN。IEEE已经对几种LAN和MAN技术进行了标准化。Wi-Fi使用IEEE 802.11标准，其变体则分别加上一个后缀，如802.11b或者802.11g。无线LAN可以是ad hoc型的，也可以是使用有接入点的基础结构型的，其帧格式包含了接入点的MAC地址以及此接入点上一级路由器的MAC地址。

除LAN以外，无线技术还用于MAN和PAN。主要的MAN技术称为WiMAX，可用于回程链接或接入。还有各种PAN技术，包括蓝牙、超宽带、Zigbee和IrDA。RFID标签提供了另一种形式的无线通信，其主要用于仓储和航运。

无线WAN使用蜂窝和卫星技术。根据提供的业务形式，蜂窝技术可分为1G（模拟话音）、2G（数字话音）、3G（数字话音和数据）和4G（高速数字话音和数据）。每一代蜂窝技术都存在很多相互竞争的实现技术。VSAT技术使得企业和用户有可能在他们的场所放置碟形天线来使用卫星通信。

新兴的无线系统使用软件可编程无线电，允许软件控制无线电传输的各个方面。可编程

无线电很昂贵，当前主要用于军事和特殊用途。

练习题

16.1 举出3种无线PAN技术的名称，并简要描述每种技术。
16.2 无线LAN和PAN使用哪3种频率段？
16.3 什么是Wi-Fi联盟？
16.4 给出用于Wi-Fi网络的IEEE标准的数值前缀。
16.5 列举3种扩频技术，并全面描述每种技术。
16.6 上网查找OFDM，并用自己的话给出一段描述。
16.7 列出无线LAN中已被提议或制定的IEEE标准。
16.8 为什么大部分无线LAN采用基础结构型方案而不采用ad hoc方案？
16.9 为什么一台无线计算机必须与一个具体的基站相关联？
16.10 802.11帧首部包含两个目的地址。解释它们各自的用途。
16.11 什么是SIFS和DIFS，为什么需要它们？
16.12 说出两种WiMAX技术的名称，并描述各自的用途。
16.13 什么是Zigbee，其用于何处？
16.14 举出UWB技术的特征。
16.15 在文件传输等应用中使用IrDA是否明智？为什么？
16.16 什么是RFID，其用于何处？
16.17 单元区的基站塔连接到何处？
16.18 什么是基站集群，设计师怎样使用集群？
16.19 说出四代蜂窝技术的名称，并描述每一代蜂窝技术。
16.20 什么是GSM，它包含哪些标准？
16.21 在第二代蜂窝技术中，使用码分复用的技术是什么？
16.22 什么是VSAT卫星？
16.23 为什么一个卫星碟形天线的形状是抛物线？
16.24 说出通信卫星使用的3个主要的频带，并说明天气对各自的影响。
16.25 GPS中使用多少颗卫星，一个GPS系统有多精确？
16.26 除了定位，GPS还提供了什么功能？
16.27 软件无线电中哪些特性是可以控制的？

第17章 局域网扩展技术

17.1 引言

前面几章介绍了局域网（LAN）的拓扑结构和布线方案。典型的LAN一般的跨距为几百米。这意味着在一栋大楼或在校园范围内，LAN技术能运作良好。

本章讨论两个重要的概念：对LAN进行扩展以便跨越更远距离的机制和LAN交换技术。本章介绍中继器、网桥，以及用于防止循环转发的生成树算法。

17.2 距离限制与LAN设计

距离限制是局域网设计要考虑的一个基本因素。在设计一种局域网技术时，工程人员要在给定费用的情况下，选择一种网络容量、最大延迟和连接距离等诸因素的组合。造成距离限制的原因是由于硬件被设计成只能发出一定的能量——如果布线超过了这个设计极限，站点将接收不到足够强的信号，差错也将随之产生。

要点 最大长度规格是LAN技术的基本部分；在超过此长度界限的线路上，LAN硬件将不能正常地工作。

17.3 光纤调制解调器扩展

工程人员已经开发了多种方式来扩展LAN的连通性。作为一般的规则，扩展机制既不是要增强发送信号的强度，也不仅仅只是延长线缆。因此，大部分扩展机制都是在标准的接口上插入附加硬件，使得信号能够实现更远距离的中继。

最简单的LAN扩展机制由一条光纤和一对光纤调制解调器（fiber modems）组成，这些设备可以将一台计算机连接到一个远程以太网。图17-1表示了这种互连方法。

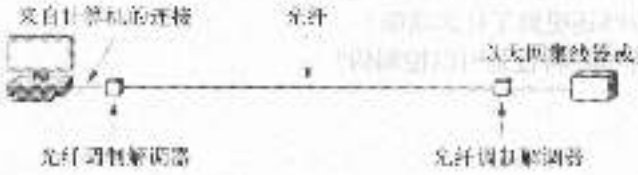


图17-1 实现一台计算机与远程以太网互连的光纤调制解调器示意图

每一个光纤调制解调器都含有专门执行以下两个任务的硬件：通过以太网接口接收分组并通过光纤发送这些分组，以及通过光纤接收分组并通过以太网接口发送这些分组[1]。如果调制解调器在各自的端口提供一个LAN接口，那么计算机和LAN设备就可以使用标准的LAN接口。

概括如下：

---

(1) 实际中，常使用一对光纤来实现，以允许双向同时传输。

使用光纤和一对光纤调制解调器，可以实现计算机与远程LAN（例如以太网）之间的连接。

17.4 中继器

中继器（repeater）是用于远距离传播LAN信号的模拟设备。中继器并不理解分组或信号编码，它只是放大接收到的信号，并将放大的信号再转发。

中继器曾广泛地应用于早期的以太网中，如今已经与其他LAN技术一起应用于实现网络的扩展。最近，中继器引入了红外线接收器，它允许接收器放置在距离计算机更远的地方。例如，可以考虑这样一种情况：用于有线电视控制器的红外线接收器必须与控制器放置在不同的房间。中继器可用于扩展这种连接，正如图17-2所示。



图17-2 使用中继器进行扩展的红外线传感器示意图

概括如下：

中继器是用来扩展LAN的模拟硬件设备。中继器可放大所有从输入一侧接收到的信号，然后再发送到另一侧。

17.5 网桥与桥接

网桥（bridge）是一种连接两个LAN网段（如两个集线器）并在网段之间传输分组的机制。网桥以混杂模式（promiscuous mode）监听每个网段（即接收所有被发送到网段上的分组）。当网桥从一网段接收到一个完整帧时，它会将此帧的一个副本转发到另一个网段上。因此，连接到一个网桥的两个LAN网段，其行为表现与单一LAN相似，即连接于任一网段的计算机可以给连在两个网段的任一计算机发送帧。此外，广播帧也会转发给两网段内的所有计算机。因此，计算机并不知道它们是连接在单一LAN网段上还是在被桥接的网段上。

起初，网桥是被作为具有两个网络连接接口的独立硬件设备销售的。如今，网桥技术已合并在其他设备（如缆线调制解调器）中。图17-3示意了这个概念结构。



图17-3 6台计算机连接到一对桥接LAN网段上的示意图

概括如下：

网桥是用来连接两个LAN网段并在网段之间转发分组的一种机制；计算机不能分辨它们自己是连接在单一网段上还是在被桥接的网段上。

17.6 自学习网桥与帧过滤

网桥并不是将每个帧的副本盲目地从一个LAN转发到另一个LAN，而是要利用MAC地址来执行过滤（filtering）操作。也就是说，网桥检查帧中的目的地址，并只在必要的时候才将帧转发到另一个LAN网段。当然，如果LAN支持广播或多播，网桥就必须转发每个广播帧或多播帧的副本，从而使得桥接LAN像单一LAN那样工作。

网桥如何知道哪台计算机连接到哪个网段呢？大多数网桥被称为自适应的（adaptive）或自学习的（learning）网桥，因为它们能自动地获悉计算机的位置。为此，网桥要利用帧中的源地址。当一个帧从某个网段到达时，网桥从帧头部提取源地址，并将该地址加入到连接于此网段的计算机列表中。当然，网桥还须从帧中提取目的MAC地址，并利用此地址来确定是否转发此帧。这样，只要计算机发送一个帧，网桥即可获悉此计算机处于哪个网段中。

为了理解网桥是如何在帧传输时获知计算机所处网段的位置信息，参考图17-3中的网桥。图17-4列出了传输的序列分组，网桥在每一步所积累的位置信息以及分组的去向（即分组所要发往的网段）。

事　件	网段1	网段2	帧发往的网段
网桥启动	–	–	–
A发往B	A	–	两个网段
B发往A	A，B		仅网段1
X广播	A，B	X	两个网段
Y发往A	A，B	X，Y	两个网段
Y发往X	A，B	X，Y	仅网段2
X发往Z	A，B	X，Y	两个网段
Z发往X	A，B，C	X，Y，Z	仅网段2

图17-4 自学习网桥的一个例子，计算机A、B和C在一个网段，计算机X、Y和Z在另一网段

概括如下：

自适应网桥利用分组中的源MAC地址来记录发送者的位置，并利用目的MAC地址来决定是否转发该帧。

17.7 为什么桥接能行

网桥一旦获悉了所有计算机的位置信息，桥接的网络就能够比单一LAN表现出更高的整体性能，了解这一点很重要。要理解其中的原因，重要的是要明白：网桥允许各网段同时传输。例如，在图17-3中，计算机A可以给计算机B发送一个分组，而计算机X也可以同时给计算机Y发送一个分组。尽管网桥会收到每个分组的副本，但它并不转发其中任何一个，因为这两个分组发送的目的地址与源地址都位于各自的同一网段。因此，网桥只是丢弃这两个帧而不转发。

概括如下：

因为网桥允许桥接的各网段可同时进行传输，所以在一个网段中的两台计算机进行通信的同时，另一网段的两台计算机也可以通信。

这种通信本地化的能力使得桥接校园内的不同建筑物成为可能。大多数通信是本地化的（例如，计算机与在同一栋楼房内的打印机通信就远多于与在另一栋楼房中的打印机的通信），

而当需要时，网桥也可提供不同楼宇间的通信，但不发送无须转发的分组。DSL和电缆调制解调器也利用了桥接的概念——在用户的本地网络与因特网服务提供商的网络之间，调制解调器也是起到网桥的作用。

17.8 分布式生成树

如图17-5所示，在图中的4个LAN网段由3个网桥连接，并计划在其中插入第4个网桥。我们假定计算机（图中未显示）也被接入到各个集线器中。



图17-5 将要插入第4个网桥的桥接网络

在插入第4个网桥之前，网络正常运行——任一计算机可发送单播帧到另一台计算机，或者发送广播帧或多播帧给所有计算机。对广播和多播的支持是因为网桥总是转发目的地址为广播或多播的帧，如果插入第4个网桥，则存在一个环路，问题也随之产生，除非至少有一个网桥禁止转发广播帧，否则广播帧副本将在环内永远循环，同时连接到各集线器的计算机将会接收到无数的广播帧副本。

为了禁止环路引起无限循环，网桥执行一个计算分布式生成树（Distributed Spanning Tree，DST）的算法。也就是说，这个算法把网桥当做一个图中的节点，并在图上生成一棵树（树是不含有环的图）。最初的方法，是由数字设备公司于1985年提出的，它是为以太网设计的，称之为生成树协议（Spanning Tree Protocol，STP）。STP包括3个步骤：

- 推选出根节点；
- 计算最短路径；
- 转发。

为了使用STP，以太网网桥利用了一个为生成树保留的多播地址在节点之间相互通信。这个多播地址是[⊖]：

01:80:C2:00:00.00

第一步是推选出一个根节点，推选机制很简单：各网桥多播一个包含各自网桥标识符（bridge ID）的分组，拥有最小ID的网桥即被选为根节点。为了使管理员可以控制选择，网桥标识符由两部分组成：16位的可配置优先数（priority number）和48位的MAC地址。比较标识符时，网桥先比较优先数，再利用MAC地址部分决定大小。因此，管理员可以通过赋给比其他网桥更低优先数的方法来保证一个特定的网桥成为根节点。

第二步是计算最短路径。每个网桥都可以计算出一条到达根网桥的最短路径。于是，所有网桥与根网桥的最短路径连接起来，即构成了生成树。

一旦生成树被计算出来，网桥便开始转发分组。连接在最短路径上的接口可以转发分组，而没有连接在最短路径上的接口则被阻塞，这意味着用户分组不可以通过此接口进行发送。

⊖ 通常，以太网地址使用一六进制数表示，每一对十六进制数之间使用冒号分隔。

多种生成树的变种已被设计出来并进行了标准化。在1990年，IEEE制定了一个名为802.1d的标准，此标准在1998年进行过更新。IEEE 802.1q标准提供了一种在一组逻辑独立的网络上运行生成树的方法，这组逻辑网络共享物理介质且不会有任何混乱和干扰。思科开发了一个生成树专有版本（Per-VLAN Spanning Tree，PVST），用于VLAN交换机[⊖]，随后此协议更新为PVST+，使之与802.1q兼容。1998年，IEEE 802.1w标准引入快速生成树协议（Rapid Spanning Tree Protocol），以减少网络拓扑发生变化后网桥进行收敛所需要的时间。快速生成树已被纳入801.1d-2004，现在在网桥类设备上已经取代了STP。另两个版本，多实例生成树协议（Multiple Instance Spanning Tree Protocol，MISTP）和多快速生成树协议（Multiple Spanning Tree Protocol，MSTP）都被指定用于处理更复杂的VLAN交换机。其中，MSTP已被纳入IEEE 802.1q-2003标准。

17.9　交换与第二层交换机

桥接的概念有助于解释形成现代以太网基础的一个机制：交换（switching）。以太网交换机（Ethernet switching），有时又称为第二层交换机（Layer 2 switching），是类似集线器的电子设备。与集线器的相同之处在于，它们都提供多个端口（ports），每个端口连接一台计算机，并允许计算机之间发送帧。与集线器的不同之处在于设备运行的方式：集线器是在计算机间转发信号的模拟设备，而交换机是转发分组的数字设备[⊖]。我们可将集线器看成是仿真共享传输介质，而将交换机看成是仿真每个网段只有一台计算机的桥接网络。图17-6所示为在交换机中对网桥概念的运用。

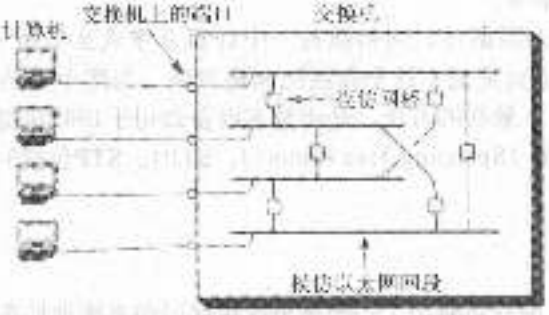


图17-6　交换式LAN的概念式组织结构

上图只是提供了一个概念视图，其实交换机中并不含有单独的网桥。在实际的交换机中，包含一个连接各个端口的智能接口（intelligent interface）和一个可为各对接口间提供同时传输的中央结构单元（fabric）。智能接口包含一个处理器、存储器以及实现其功能所必要的其他硬件。接口所实现的功能包括：接收进入的分组、查询转发表，以及接收来自中央结构单元的分组，并将其发往对应端口。最重要的是，由于接口拥有存储器，它能在输出端口繁忙时缓存到来的分组。因此，如果计算机1和计算机2同时向计算机3发送分组，那么在其他接口繁忙期间，接口1或接口2会将分组保存起来。图17-7所示为这种结构。

在物理上看，交换机有多种大小型号。最小型的交换机是较廉价、提供4个连接接口的独立设备，它足以连接一台计算机、一台打印机以及另外两台诸如扫描仪之类的设备。公司使

⊖ 下一节介绍交换与VLAN（虚拟局域网）交换机。

⊖ 原书这里以“模拟”或“数字”来区分设备的不同，并未对问题出中要害。其实，集线器是属于物理层桥接设备，而网桥是属于数据链路层（即第2层）桥接设备，这才是关键。——译者注

用最大型的交换机来连接整个公司数以万计的计算机和其他设备。

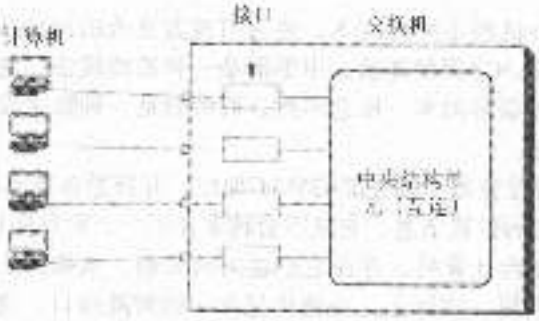


图17-7 交换机结构示意图

使用交换式LAN代替集线器的主要优点在于其并行性。集线器在某一时刻只能支持一个传输，而交换机允许在同一时刻进行多个传输（假设传输是独立发生的，即在特定时刻只有一个分组正被发送到一个特定的端口）。因此，如果一台交换机有N个端口连接N台计算机，同一时间可进行N/2个传输。

要点 因为交换机处理的是分组而不是信号，而且使用了中央结构单元提供内部并行路径，所以一台有N个端口的交换机能最多同时传输N/2个分组。

17.10 虚拟局域网交换机

通过加入虚拟化措施，可以对交换机进行扩展，扩展后的交换机被称为虚拟局域网交换机（Virtual Local Area Network switch，VLAN switch）。扩展的概念简单易懂，即允许管理员对一台交换机进行配置，使其可以仿真多台独立的交换机。也就是说，管理员划定这个交换机上的一组端口并将它们指派给VLAN 1，再把另一组端口指派给VLAN 2，依此类推。当VLAN 2的一台计算机广播一个分组时，只有在同一个VLAN的计算机才会收到该广播分组的副本（即一旦完成配置，一台VLAN交换机看起来就像是多台交换机）。

只有在考虑到大型公司或服务提供商时，将诸多计算机划分为多个分隔的广播域（broadcast domains）才显得尤为重要。在这种情况下，保证一组计算机在它们进行通信时不被外部计算机接收和不接收外部的分组，都是非常重要的。例如，一家公司可能会选择在CEO办公室的计算机与公司的其他计算机之间设置一个防火墙[6]，那你就要把CEO的那些计算机配置成一个单独的VLAN，这样就允许安装防火墙了。

17.11 使用其他设备实现桥接

虽然我们是将网桥作为一个独立的设备来进行描述的，但桥接作为一个基本的概念已被吸收进许多的设备之中。例如，DSL或电缆调制解调器就可提供一种桥接方案：调制解调器在用户驻地提供一个以太网连接，并在用户驻地与提供商网络之间传输以太网分组。有些无线技术也会利用某种形式的桥接，在移动设备与提供商的网络之间传输帧。因此：

> 尽管厂家不再销售独立的网桥设备，但桥接的概念已经被吸收进各种网络设备之中，例如，接入技术中采用的调制解调器。

[6] 第30章介绍防火墙。

17.12 本章小结

人们已经发明了多种机制来扩展LAN，使之可覆盖更大的地域范围。一对光纤调制解调器可用来延伸计算机与LAN之间的连接。中继器是一种模拟设备，它放大来自LAN网段的电子信号并向另一网段传输信号副本。反之亦然。网桥则是一种数字设备，它连接两个LAN网段并在网段之间转发分组。

为了优化转发，网桥检查每个帧头部的MAC地址，并记录各计算机所属网段的位置信息。一旦网桥掌握了各计算机的位置信息，它就不会转发在同一个网段内计算机之间发送的分组。

以太网交换机连接多台计算机，并在它们之间转发帧。从概念上讲，一台交换机相当于一组用网桥互连的LAN网段。实际上，交换机包含一组智能接口，这些接口采用了一种称为"中央结构单元"的高速硬件来实现互连机制。与集线器相比，交换机的主要优点是，假如只有一个分组被引向一个特定输出端口，则交换机可以同时转发多个分组。VLAN交换机允许管理员将一台交换机配置成起到一组多台独立交换机的作用。

练习题

17.1　当用光纤来扩展LAN的连接时，还需要哪些附加设备？

17.2　如果一台电视机能为远程红外传感器提供有线扩展，其最有可能运用什么技术？

17.3　如果两台计算机连接在一个桥接网络上，地址或应用需要改变吗？请解释。

17.4　试给出对自适应网桥转发一个分组的条件的准确描述。

17.5　考虑发往一个桥接LAN的分组，其目的地址并不存在。请问：网桥转发此分组将会经过多少个网段？

17.6　假设一个网络包括3个速率为100Mbit/s的网段，它们由两个网桥连接，且每个网段有一台计算机。如果两台计算机向第三台计算机发送数据，发送方可达到的最高速率和最低速率是多少？

17.7　在网上搜索生成树算法的描述，然后编写一个模拟网桥形成生成树的计算机程序。

17.8　在桥接以太网中的计算机会不会收到生成树分组？请解释。

17.9　用网络分析器观测桥接式以太网上的通信量。网桥重新启动后你会观察到什么？

17.10　当桥接应用于卫星连接时，通常需要两个网桥，一端一个。请解释为什么。

17.11　根据图17-6，连接到一个交换式LAN中的两台计算机是否能同时传输分组？请解释。

17.12　扩展图17-6，使之含有5个端口。

17.13　针对上题中的情况，请写出一个等式，以表达出被模仿的网桥数作为端口数的函数关系。

17.14　编写一个计算机程序来模仿网桥的功能，以两个数据文件来模仿通过网桥连接的两个局域网网段间的帧传输。假定每个模仿帧包含一个源地址和一个目的地址。为了进行模仿，先从第一个文件中读入一帧，然后从第二个文件中读入一帧，如此下去。对于每一帧，都显示网桥是否会把帧转发到另一个网段。

17.15　扩充上题中的程序使之可模仿VLAN交换机。程序从读入说明一组主机和一组它们所连的虚拟网段的配置信息开始。创建一份帧文件，文件指定发送每一帧的计算机（即帧应到达交换机的哪个端口）以及帧的目的地址。显示每一帧是如何转发的。

17.16　网桥能将Wi-Fi网络连接到以太网吗？用交换机可以吗？说明理由。

第18章 广域网技术与动态路由

18.1 引言

本书中这部分的各章介绍各种有线与无线分组交换技术。上一章介绍了LAN扩展技术，这一章再介绍一种可覆盖任意大小区域的网络结构。在本章中，我们要描述构建一个分组交换系统要用到的各个基本组成部分，并解释路由的基本概念；还要阐述两种基本的路由算法，并解释这两种算法各自的优点。下一章再进一步讨论因特网的路由问题，并介绍运用本章所述算法实现的路由协议。

18.2 大型广域网络

我们曾经提到，网络技术可以根据覆盖范围的大小进行分类：

- PAN——覆盖一个个体附近的区域。
- LAN——覆盖一座建筑物或一个校园。
- MAN——覆盖一个大都市区域。
- WAN——覆盖多个城市或国家。

一个公司使用卫星网桥连接两个不同站点的局域网，那么这个网络应该属于WAN还是扩展LAN呢？如果这个公司在这两个站点中分别只有一台PC和一台打印机，答案是否就会改变呢？回答是肯定的。区别WAN技术和LAN技术的关键是网络的规模（scalability）——WAN能按需要扩展连接地理距离较远的许多站点，而且每个站点内有多台计算机。例如，WAN应能连接一个大公司散布于数千平方公里范围内几十个不同地点的办公室或工厂的所有计算机。而且，还必须使大规模网络的性能达到相当水平，否则不能称之为WAN。也就是说，WAN不仅仅只是连接许多站点中的多台计算机，它还必须能提供充足的网络容量，以使大量计算机之间能彼此通信。因此，一个连接一对PC和打印机的卫星网桥只不过是一个扩展LAN。

18.3 传统的广域网体系结构

在局域网出现和个人计算机应用之前，而且是在因特网诞生之前，传统的广域网技术就已经发展了[⊖]。迄今，传统广域网的体系结构被设计成可以连接大量的站点，每个站点都有少量大型计算机的形式。

由于当时没有出现局域网技术，因此WAN设计者选择研制了一个具有特殊用途且能放置在每个站点的硬件设备，该设备称为分组交换机（packet switch）。分组交换机既可以提供站内计算机之间的连接，也可以提供通向另一个站点的数据线路的连接。

从概念上讲，一个分组交换机是由一个小型的计算机系统组成的，该系统有处理器和存储器以及用来收发分组的I/O设备。早期的分组交换机是由普通计算机构成的，而现代高速广域网中所用的分组交换机则需要具有特殊用途的硬件。图18-1说明了分组交换机的内部体系

⊖ 早期，WAN被称为远程网（long haul network）。

结构，

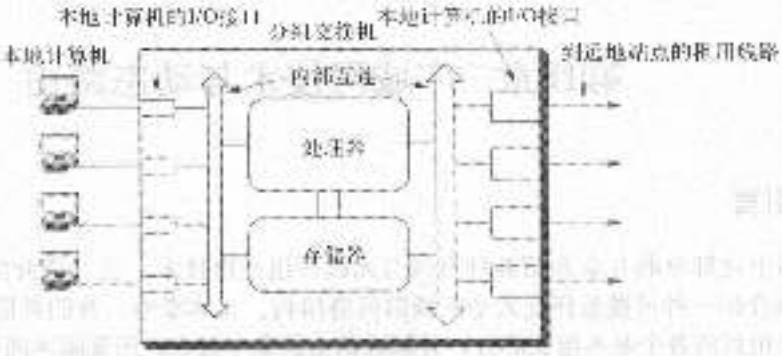


图18-1　传统分组交换机的系统结构

如图所示，一台分组交换机包含两种I/O设备，第一种I/O设备可以高速运行，用于连接本地交换机和通向另一台分组交换机的数字线路，第二种I/O设备以较低的速度运行，用于连接本地交换机和个人计算机。

自从LAN技术出现以后，大多数WAN把一个分组交换机分为两部分：一台连接本地计算机的第2层交换机和一个连接其他站点的路由器。本书的第四部分将会详细讨论因特网路由器，并解释如何将此处涉及的概念应用到因特网。现在，我们只要理解这一点就足够了，即本地计算机的通信可以与通过WAN的传输分开考虑。图18-2所示为这种分离后的连接关系。

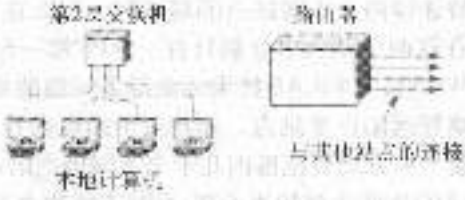


图18-2　由分离LAN交换和本地通信的现代WAN站点示意图

18.4　广域网的构成

从概念上讲，一个WAN可通过互连一系列站点构成，而这种互连的确切细节要取决于所需的数据速率，覆盖距离，以及所能接受的延时。许多WAN都采用租用数据线路的形式（例如，一条T3线路或一条OC-12线路）。然而，其他形式的连接技术也是可行的，例如微波和卫星信道。除了选择一种连接技术外，设计者还必须选择一种拓扑结构。对于给定的一组站点，可以有多种可能的拓扑结构。例如，图18-3给出了一种4台分组交换机和8台计算机互连所构成WAN的可能方式。

如图所示，WAN的互连布局不必是对称的。交换机之间的互连和每条连接的容量都是根据预测的通信量来确定的，还要提供冗余量以应故障发生时所需。在图中，站点1的分组交换机和其他的网络只有一个连接，而其他站点的分组交换机至少有两个外部连接。

要点　传统的WAN由一组分组交换机互连而成，每个站点的分组交换机与计算机相连。连接的拓扑结构和容量都要根据预测的通信量来确定，还需要留有一定的冗余量。

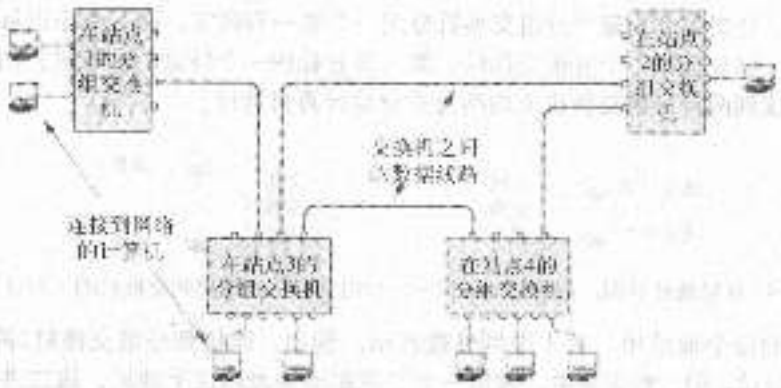


图18.3 由分组交换机互连而构成WAN的例子

18.5 存储/转发模式

WAN的目标是允许尽可能多的计算机能同时发送分组，而实现这种同时传输的基本模式称为存储/转发（store and forward）。为完成存储与转发的处理过程，分组交换机在存储器中对分组进行缓冲。存储（store）操作在分组到达时执行，即分组交换机的I/O硬件把分组的一个副本放在存储器中并通知处理器。转发（forward）操作在分组已到达并在存储器中等待时执行。处理器检查分组，确定其目的地址，然后通过连接到目的地的I/O接口发送该分组。

采用存储/转发模式的系统能保持每个数据连接都在使用，因而能提高整体性能。更重要的是，如果有多个分组发送到同一个输出设备，分组交换机能接收并将这些分组保存到存储器中，直到该输出设备准备好发送。例如，考虑图18-3所示网络中的分组传输情况。假设站点1中的两台计算机几乎同时产生一个分组发送到站点3中一台计算机，这两台计算机可以同时将它们的分组发送到分组交换机中。每个分组到达时，分组交换机中的I/O硬件把分组放在存储器中并通知分组交换机的处理器。处理器检查每个分组的目的地址，确定分组都要发往站点3。当一个分组到达时，如果连接到站点3的输出接口是空闲的，则立即开始传输。如果输出接口处于忙碌中，处理器就把将要发出的分组放在该设备相关的队列中。一旦发送完一个分组，该输出设备就从队列中取出下一个分组并开始发送。

这个概念可以被概括如下：

> 广域分组交换系统采用存储/转发技术，将到达分组交换机的分组先放进队列排队，直到交换机能将它们朝目的地转发出去。这项技术使分组交换机能对同时到达的短时突发分组进行缓冲。

18.6 广域网的编址与寻址

从一台联网计算机的角度看，传统的WAN网络运行起来与LAN相似。每种WAN技术都准确定义了计算机在收发数据时所使用的帧格式，而且连到WAN上的每台计算机都分配有一个地址。当把帧传输到另外一台计算机时，发送方必须提供目的地址。

虽然具体细节有所不同，但WAN的地址还是遵循因特网中采用的一个关键概念：分层编址（hierarchical addressing）。从概念上讲，分层编址把每个地址分成两部分：

（站点，站点中的计算机）

在实际中，分层编址为每个分组交换机分配一个唯一的数字，而不是标识每个站点。这意味着地址的第一部分标识一个分组交换机，第二部分标识一个特定的计算机。例如，图18-4表示出分配给连接到两台分组交换机上的两段式分层计算机地址。

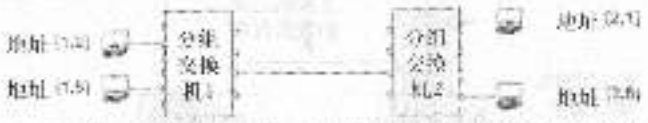


图18-4　分层地址举例，每个地址标识一个分组交换机和连接到交换机的一台计算机。

图中所示的每个地址用一对十进制整数表示。例如，连接到分组交换机2的6号端口的计算机被赋予地址[2，6]。在实际中，常用一个二进制值来表示这个地址。该二进制值的某些位用来标识一个分组交换机，其他位则用来标识一台计算机。在本书第四部分中，我们将看到因特网采用同样的方案，即每个因特网地址由一个二进制数字组成，该数字前几位标识因特网中某一个特定的网络，其他位则标识连接到该网络的一台计算机。

18.7　下一跳转发

当我们考虑到分组处理时，分层编址与寻址的重要性就显而易见了。当一个分组到达时，分组交换机必须选择一条输出路径来转发每个分组。如果分组的目的地是一台本地计算机，交换机就直接将分组发往该计算机。否则，分组必须通过连接到另外一个交换机的一条链接进行转发。为了作出选择，分组交换机检查分组中的目的地址，并提取分组交换机的号码。如果目的地址中的分组交换机号码与本分组交换机的ID相同，则该分组的目的地址是与本地交换机相连的一台计算机。否则，该分组的目的地址就是另一个分组交换机上的计算机。算法18-1解释了这种寻址方式的计算过程。

这种寻址方式的重要理念是，分组交换机无须保存有关如何到达所有可能目的计算机的完整信息，也不用计算一个分组在整个网络中的路由，而是由一个交换机基于分组交换机的ID来决定转发行为。这意味着，一个交换机只需要知道使用哪一条输出链路就可以到达一个指定的交换机。

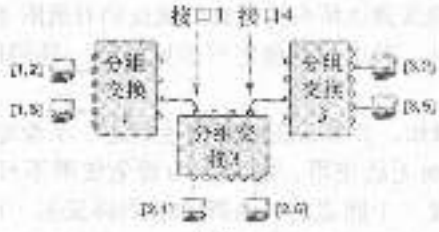
算法18-1

```
Given:
   A packet that has arrived at packet switch O
Perform:
   The next-hop forwarding step
Method:
   Extract the destination address from the packet;
   Divide the address into a packet switch number, P, and a
      computer identification, C;
   if (P == O) {        /* the destination is local */
      Forward the packet to local computer C;
   } else {
      Select a link that leads to another packet switch, and for-
ward
         the packet over the link;
   }
```

算法18-1　分组交换机转发分组的两步算法

我们提到，每个交换机只需计算分组的下一跳（next hop）地址，这个过程称为下一跳转发（next hop forwarding），类似于飞机航班表。假定一个从旧金山飞往迈阿密的旅客发现仅有的一个航班包含3个航程：第一航程从旧金山到达拉斯，第二航程从达拉斯到亚特兰大，第三航程从亚特兰大到迈阿密。虽然整个旅行的最终目的地仍然是迈阿密，但每个机场的下一站都不一样。当这个旅客离开旧金山时，下一站是达拉斯；当旅客在达拉斯时，下一站是亚特兰大；当旅客在亚特兰大时，下一站是迈阿密。

为了使这个计算更有效，分组交换机采用查表法。也就是说，每个分组交换机包含一个转发表[⊖]（forwarding table），这个表列出了所有可能的分组交换机，并为每个分组交换机给出一个下一跳地址。图18-5通过一个小例子说明下一跳转发的原理。



a)

目的地	发往
交换机1	接口1
交换机2	本地分发
交换机3	接口4

b)

图18-5 每个交换机都有不同的下一跳转发信息：a) 由3个分组交换机构成的网络；b) 交换机2的下一跳转发表

使用转发表时，交换机提取分组的目的地址，并将地址中标识分组交换机的部分作为转发表中的一个索引。例如，考虑图18-5b，如果一个分组的目的地址为[3,5]，那么交换机抽取出3，通过查找转发表，交换机把此分组发往接口4，这个接口连接了交换机3。

仅利用两段式分层地址的一部分来转发分组，有两个重要的实际意义。第一，转发分组所需的计算时间减少了，因为转发表可组织成一个数组，可以利用索引来代替搜索。第二，在这个转发表中，每个分组交换机（而不是每台目的计算机）包含一个表项。这样，转发表的规模可以大大缩小，尤其是对于每个分组交换机连接很多计算机的大型WAN而言。

实质上，利用两段式分层地址方案转发分组，除了分组到达的最后的交换机（即目的计算机所连接的交换机）外，所有其他分组交换机都只需利用目的地址的第一部分。当分组到达最后的一台交换机时，该交换机利用目的地址的第二部分选择一台指定的计算机，该过程正如算法18-1所描述。

概括如下：

> 在WAN中转发一个分组时，只需利用分组目的地址的第一部分。当分组到达目的计算机所连的交换机时，交换机利用目的地址的第二部分来把分组转发给正确的本地计算机。

18.8 源点独立性

值得注意的是，下一跳转发的过程既不取决于分组的源地址，也不取决于分组到达某一特定的分组交换机之前所经过的路径，而是仅取决于分组的目的地址。这个概念称为源点独立性（source independence），它是网络中的一个基本概念。这个概念将隐含于本章以及后续

⊖ 虽然纯粹主义者坚持要使用转发表这个名字，但其实这种表现实是称为路由表（routing tables）的，并且这个术语在网络界仍广泛使用。

章节中有关因特网转发分组的讨论中。

源点独立性能使计算机网络中的转发机制更简捷和更有效。因为所有分组沿相同的路径发送，仅需一张路径表。转发不需要利用分组源点的任何信息，因此只需要从分组中提取出目的地址即可。此外，这种单一的机制足以来应付统一的转发处理过程——来自本地计算机的分组和来自其他分组交换机的分组都采用相同的转发机制。

18.9　广域网动态路由更新

为使广域网能正确运行，每个交换机都必须拥有一个转发表，并且都必须能转发分组。此外，转发表中的数据必须符合以下条件：

- 全局通信——每个交换机的转发表必须包含到达所有可能目的地址的有效的下一跳路径。
- 最优路径——在每个交换机的转发表中，对于一个给定目的地的下一跳的值，必须是指向目的地的最短路径。

网络故障会使转发过程进一步复杂化。例如，如果存在两条路径到达一个给定的目的地，其中一条由于硬件故障（如线路断开连接）而无法使用，那么，为避免使用不可用的路径，就应该改变转发表。因此，管理员不能仅配置一个固定的具有静态值的转发表，而是应该在分组交换机上运行软件以便不停地测试故障，并自动重新配置转发表。我们使用术语路由软件（route software）来描述自动重新配置转发表的这种软件。

理解WAN中路由计算的最容易的方法，就是考虑用一个图来模仿网络模型，并设想软件使用该图来计算到达所有可能目的地的最短路径。图中的每个节点（node）代表网络中的一个交换机（个人计算机不属于图中部分）。如果网络中包含一对交换机之间的直接连接，则在图中的相应节点间有一条边（edge）或链路（link）[⊖]。例如，图18-6说明了一个WAN的实例及其对应的图。

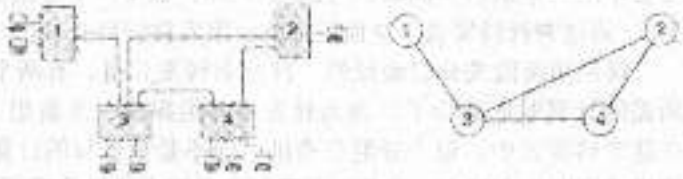


图18-6　一个WAN及其对应的图

如图所示，图中各个节点及其对应交换机都配有相同的标号。网络的图形表示在计算下一跳转发中很有用，因为图论已经得到很好的研究，而且已经开发出一些有效的算法。此外，图将细节抽象开来，允许路由软件能够处理问题的本质。

当路由算法计算一幅图的下一跳转发时，它必须标识一条链路。我们的例子将采用符号（k，j）来表示一条从节点k到节点j的链路。因此，当为图18-6b所示的网络节点抽象图运行路由算法计算下一跳转发时，该算法产生的输出结果如图18-7所示。

目的地	下一跳
1	—
2	(1，3)
3	(1，3)
4	(1，3)

节点1

目的地	下一跳
1	(2，3)
2	—
3	(2，3)
4	(2，4)

节点2

目的地	下一跳
1	(3，1)
2	(3，2)
3	
4	(3，4)

节点3

目的地	下一跳
1	(4，3)
2	(4，2)
3	(4，3)
4	

节点4

图18-7　图18-6b所示的图中每个节点的转发表

⊖ 因为图论和计算机网络的关系密切，所以我们经常可以看到分组交换机被称为网络节点（network node），而连接两个站点的数据线路被称为链路（link）。

18.10 默认路径

图18-7中节点1的转发表说明了一个重要思想：一个转发表可能包含很多指向相同下一跳的表项。检查图18-6a中WAN，可以揭示所有不同表项都包含相同下一跳的原因，就是此分组交换机只有一条连接到网络的链路。因此，所有向外的传输都必须通过这条唯一的链路发送出去。这样，除了对应于节点自身的表项外，节点1的转发表中所有表项都包含有相同的下一跳值，即指向节点1到节点3的那条链路。

在小型网络的例子中，转发表中重复表项的列表很短。然而，一个大的WAN可能包含数百个重复表项。大多数WAN系统都包含有消除这种重复路由的机制，即称为默认路径（default route）。这种机制允许转发表使用单个表项来代替一长串具有相同下一跳值的表项。在一个转发表中，只允许存在一个默认表项，与其他表项相比，这个表项具有较低的优先级。如果转发机制找不到一个针对某一指定目的地址的明确表项，转发机制就使用默认项。图18-8表示出如何将图18-7中的转发表修改成使用默认路径的情况。

目的地	下一跳
1	—
*	(1, 3)

节点1

目的地	下一跳
2	—
3	(2, 4)
*	(2, 3)

节点2

目的地	下一跳
1	(3, 1)
2	(3, 2)
3	
4	(3, 4)

节点3

目的地	下一跳
2	(4, 2)
3	—
*	(4, 3)

节点4

图18-8 图18-7中的转发表使用星号来表示默认路径

默认路径是可选的——仅当超过一个以上的目的地址有相同的下一跳时，才会存在一个默认项。例如，节点3的转发表没有默认路径，因为每个表项都有唯一的下一跳。然而，节点1的转发表则受益于默认路径，因为所有的目的地址都有相同的下一跳。

18.11 转发表的计算

如何构造一个转发表呢？有两种基本的方法：

- 静态路由（static routing）。当分组交换机启动时，由一个程序来计算并设置好路径，这些路径不再改变。
- 动态路由（dynamic routing）。当分组交换机启动时，由一个程序建立起初始的转发表，随着网络情况的变化，该程序不断更改转发表。

每种方法都有其优缺点。静态路由的主要优点是简单以及开销小。主要缺点是缺乏灵活性——当通信中断时，静态路径不能被改变。因为大型网络都设计有冗余连接，以便应对偶然的硬件故障，所以大多数WAN都采用某种形式的动态路由。

18.12 分布式路径计算

算法18-2表示了当网络信息被编码成图之后如何计算转发表的过程。实际上，WAN需要完成分布式路径计算（distributed route computation），也就是说，这种计算方法不是通过一个集中的程序来计算所有的最短路径，而是通过每个分组交换机在本地计算自己的转发表。

所有的分组交换机都必须参与分布式路径计算，有两种常用的形式：

- 链路状态路由（LSR），采用Dijkstra算法。

- 距离向量路由（DVR），采用另一种方法。

下面将分别介绍上述两种方法。第27章将解释如何利用这两种方法来控制因特网中的路径。

18.12.1 链路状态路由

这种方法的正式名称是链路状态路由（link-state routing或link-status routing），但后来也非正式地称为最短路径优先（Shortest Path First，SPF），这个术语的出现是因为Dijkstra用它来描述这个算法的工作方式，但这个术语会产生一些误导，因为所有的路由算法都是为了寻找最短路径。

为了使用LSR路由方法，每个分组交换机周期性地向网络发送携带两个分组交换机之间的链路状态的报文。例如，测试分组交换机5和9之间的链路，并发送一个诸如“交换机5和9之间的链路畅通”之类的状态报文。每个状态报文广播给网络中所有的交换机。每个交换机通过运行软件来收集不断进来的状态报文，并利用它们来构建一个网络图。然后，每个交换机使用算法18-2，并以自己为源节点生成一个转发表。

LSR算法能适应硬件故障。如果分组交换机之间的一条链路中断，与此链路相连的分组交换机将检测到这个故障，并广播一份表明该链路中断的状态报文。所有分组交换机收到广播后，更新它们网络图的备份，以反映链路状态的改变，并重新计算最短路径。同理，当一个链路恢复使用时，与该链路相连的分组交换机检测到此链路处于正常状态，就开始发送报告该链路可用的状态报文。

算法18-2

```
Given:
   A graph with a nonnegative weight assigned to each edge
   and a designated source node
Compute:
   The shortest distance from the source node to each other
   node and a next-hop routing table
Method:
   initialize set S to contain all nodes except the source node;
   initialize array D so that D[v] is the weight of the edge from the
      source to v if such an edge exists, and infinity otherwise
   initialize entries of R so that R[v] is assigned v if an
      edge exists from the source to v, and zero otherwise;

   while (set S is not empty) {
      choose a node u from S such that D[u] is minimum
      if (D[u] is infinity) {
         error: no path exists to nodes in S; quit;
      }
      delete u from set S;
      for each node v such that (u,v) is an edge {
         if (v is still in S) {
            c = D[u] + weight[u,v];
            if (c < D[v]) {
               R[v] = R[u];
               D[v] = c;
            }
         }
      }
   }
```

算法18-2　Dijkstra算法的一个版本：计算下一跳转发表R和一个指定源节点到每个节点的距离D

18.12.2 距离向量路由

与LSR算法对应的另一种主要的路由算法被称为距离向量路由（Distance-Vector Routing，DVR）。与LSR算法一样，网络中的每条链路都被赋予一个权值（weight）。在两个分组交换机之间到达目的地的距离（distance）被定义为沿着两个分组交换机之间路径的权值之和。类似于LSR，距离向量路由法安排分组交换机周期性地交换报文。然而，与LSR不同的是，距离向量路由法是安排分组交换机发送一份包含所有目的地和到达每个目的地的当前代价的完整列表。本质上，当分组交换机发送一份DVR报文时，它正在发送的就是一系列如下形式的单个语句：

"我能到达目的地X，目前它与我之间的距离是Y。"

DVR报文不是广播的，而是由每个分组交换机周期性地向它的邻机发送一个DVR报文。每个报文都包括一对（目的，距离）值。因此，每个分组交换机必须保存一份包含所有可能目的地的列表，表中内容还包括到达每个目的地的当前距离和分别使用的下一跳。目的地列表和针对每个地址的下一跳都可以在转发表中找到。我们可以把DVR软件看做为维护转发表的一个扩展，该表保存了到达每个目的地的距离。

当一份发自相邻节点N的报文到达分组交换机时，该交换机检查报文中的每一项。如果相邻节点到某目的地址有比现有路径更短的路径，则修改本机中的转发表。例如，如果相邻节点N发布一条到目的地D的权值为5的路径，而本交换机转发表中当前的路径是通过相邻节点K到目的地D的权值为100的路径，那么，就用N取代K作为到达目的地D的下一跳，而到达D的权值更新为5加上该交换机到达N的权值。算法18-3说明了采用距离向量的方法时是如何更新路径的。

算法18-3

```
Given:
    A local forwarding table with a distance for each entry, a
    distance to reach each neighbor, and an incoming DV
    message from a neighbor
Compute:
    An updated forwarding table
Method:
    Maintain a distance field in each forwarding table entry.
    Initialize forwarding table with a single entry that has the
      destination equal to the local packet switch, the
      next-hop unused, and the distance set to zero;

    Repeat forever {
      Wait for a routing message to arrive over the network
        from a neighbor; let the sender be switch N;
      for each entry in the message {
        Let V be the destination in the entry and let D
          be the distance;
        Compute C as D plus the weight assigned to the
          link over which the message arrived;
        Examine and update the local routing table:
        if no route exists to V {
          add an entry to the local routing table for destination
            V with next-hop N and distance C;
        } else if a route exists that has next-hop N {
          replace the distance in existing route with C;
        } else if a route exists with distance greater than C {
          change the next-hop to N and distance to C;
        }
      }
    }
```

算法18.3 距离向量算法的路径计算

18.13　图中最短路径的计算

当对应某个网络的图构造出来之后，计算路由表项的软件利用一种称为Dijkstra[⊖]算法的方法进行计算。利用这个算法分别找出图中各源节点与其他节点之间的最短路径，在最短路径的计算过程中，下一跳转发表也就构造出来了。该算法必须针对图中的每个节点分别运行一次。也就是说，为了计算分组交换机P的转发表，就将对应于P的节点指定为源节点并运行Dijkstra算法。

Dijkstra算法由于能用来计算各种不同定义的最短路径（shortest path）而得到广泛应用。特别是该算法不要求图中的边代表地理距离，甚至允许给每条边赋予一个非负值，称之为权值（weight），并将两节点之间的距离定义为沿该两点间通路的权值之和。这里的重点是：

> 由于Dijkstra算法在计算最短路径时使用链路上的权值，所以它字可使用权值的大小也不会使其地理距离来测量。

图18-9用一个示例来说明权值这个概念，该图的每条边都赋予一个整数权值，图中还标出了两个节点之间的一条最小权值路径。

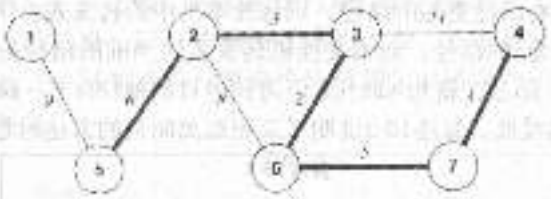


图18-9　为每条边赋予权值的示例图。图中用粗线标出了节点1到节点5之间的一条最短路径

Dijkstra算法首先建立一个节点的集合S，此时尚未计算它的最小距离和下一跳。初始化后的集合S包含除了源节点外的所有其他节点，接着算法进行迭代计算直到S为空。在每一次迭代计算中，算法都将删除集合S中与源节点距离最小的一个节点。在删除节点u的时候，算法要检查源节点与每个跟u相邻的其他节点之间的当前距离（这个检查仅限于尚保留在集合中的节点）。如果从源节点经过u到达邻节点的路径具有比当前路径更小的权值，算法则更新到邻节点的距离。待所有节点都从集合S中删除完毕后，算法也将计算出到每个节点的最小距离和一个包括所有可能路径的正确的下一跳转发表。

Dijkstra算法的实现很简单，除了用于存储查表信息的数据结构外，算法还需要分别存储3个数据结构：源节点到每个节点的当前距离，最短路径的下一跳，以及集合中剩余节点的信息。如图18-9中所示，可以用数字1到n对节点编号，这样可以使算法的实现比较高效，因为节点编号可用来作为数据结构的一个索引。尤其是，该算法可以使用两个数组D和R，每个数组都可以用节点编号来索引，数组D中第i表项存储从源节点到节点i的当前最短距离，数组R的第i项则存储用于沿所计算的路径到达节点i的下一跳值。集合S可用节点编号的双链表来实现，这样便于搜索整个集合或删除集合中的某一项。

算法18-2规定了如何在图中计算最短路径。算法使用weight（i，j）函数，它返回从节点i到节点j的边的权值。如果从节点i到j没有边，则假设weight函数返回一个保留值infinity，实际上，任何大于沿图中任何路径上权值之和的数值，都可以用来表示无穷大（infinity）。产生无穷大值的一个方法就是把所有边的权值相加后再加1。

⊖　此算法是以它的发明者E.Dijkstra的名字命名的。

允许将任意权值赋给图中的边，这意味着算法可以使用不同的量度来表示距离。例如，有些广域网技术通过计算路径上分组交换机的数目来度量距离。这时，如要使用这个算法，则图中的每条边都要赋给权值为1。在其他广域网技术中，可用反映基础连接的传输容量来表示权值。作为一种可选的方法，管理员可以给链路赋于不同的权值来实现路由策略的管理。例如，考虑这样一个案例，在两个分组交换机中存在两条独立的路径，其中一条设计为主路径，另一条设计为备份路径。为了执行这样的策略，管理员可以给主路径分配一个低权值，给另一条路径分配一个高权值。路由选择软件将会配置转发表使用低权值的路径，如果该路径不可用，路由选择软件就会选择另一条路径。

18.14 路由问题

理论上，LSR或DVR路由选择都能计算最短路径。此外，每种方法最终都将会收敛(converge)，这意味着，所有分组交换机的转发表都能达成一致。然而，问题出现了。例如，如果LSR报文丢失了，两个分组交换机分别认同的最短路径可能将不一致。DVR的问题更严重，当为一条链路的故障会导致两个或更多分组交换机产生路由循环(routing loop)，此时每个分组交换机都认为集合中下一个分组交换机就是到达特定目的地的最短路径。结果，一个分组就会在几个分组交换机之间无限地循环。

DVR协议出现问题的一个主要原因来自回流(backwash)(即一个分组交换机收到它自己发出的信息)。例如，假设一个分组交换机告诉它的邻居"我能到达目的地D，权值为3"。如果通向目的地D_i的连接失败，那么交换机就从转发表中删除D_i这一表项(或把该项标记为无效)。但该交换机已经告诉邻居有一条链路存在。假设刚好在链路失效后，其中一个邻居发送一条DVR报文，其称为"我能到达D_i，权值为4"。遗憾的是，该报文将会被其他交换机所信任，这样，一个路由循环就产生了。

大多数实际的路由机制都包含一些约束和方法来阻止出现类似路由循环这样的问题。例如，DVR方案采取分割范围(split horizon)，这个技术的细节规定可以保证一个交换机不会把信息发送回它原来的发送者，从而有效阻止回流。此外，多数实际的路由系统还引入滞后机制，以防止软件在短时间内对转发表做出太多改变。然而，在一个有很多链路频繁地发生故障而后又恢复正常的大规模网络中，在路由方面出现一些问题是不可避免的。

18.15 本章小结

广域网(WAN)技术可用来构建跨越任意远距离、连接任意多台计算机的网络。一个典型的广域网由通过通信线路互连起来的分组交换机所构成。一个分组交换机由一个处理器、存储器和多个I/O接口组成，此接口既可以连接本地计算机，也可以连接其他分组交换机。

分组交换网采用存储/转发的方法，它要先把到达的分组存放在交换机的存储器中，直到处理器能够将分组转发到它的目的地。转发操作要依靠一种称为转发表的数据结构，对于每个目的地在表中都有一个项，并指定了到达该目的地的下一跳。为节省空间，转发表通常是以交换机而不是以计算机作为目的地的。

广域网可以表示为一个图，图中每个节点对应于一台分组交换机，每条边对应于一条通信线路。这样的表示很有用，因为它忽略了细节，允许分析网络，并使于用来计算路由表。路由软件中使用的两种基本方法是：链路状态路由(LSR)和距离向量路由(DVR)。LSR让每个分组交换机广播与其直接相连的链路状态，并用Dijkstra最短路径算法计算最短路径。DVR让每个分组交换机向它的所有邻居发送一个含有本机到达每个目的地以及相应权值的列

表。它的邻居检查所收到的DVR报文中的这张列表，如果存在更低代价的路径，就替换自身转发表中相应的表项。

练习题

18.1　一个传统的分组交换机由哪几个概念部分组成？分组交换机连接什么设备？

18.2　一个现代的分组交换机分为哪两个概念部分？

18.3　一台计算机能否使用以太网接口与WAN通信？请解释。

18.4　如果一个WAN连接N个站点，至少需要多少条数字线路？最多可以使用多少条？

18.5　解释存储转发模式。

18.6　WAN地址的两个组成部分是什么？

18.7　图18.4表示了如何给连接到一个分组交换机上的计算机分配地址。假定交换机上有一个硬件接口发生了故障，网络管理员把计算机接到另一个未用的接口上。这种新的配置还能正常工作吗？说明原因。

18.8　请编写一个计算机程序，其输入是一个转发表和一系列的分组，其输出是每个分组应该如何转发的说明。记住，要能够处理含有不正确地址的分组。

18.9　考虑一个有两个分组交换机的WAN。假设每个交换机的转发表中对于每个本地地址（即和本交换机直接相连的每台计算机的地址）都有对应的一项，并有一个默认项指向另一台交换机。在什么情况下该方案可以正常工作？在什么情况下该方案将会失败？

18.10　动态路由有什么好处？

18.11　编写一个计算机程序来实现Dijkstra算法，求出一个图的最短路径。

18.12　分布式路由计算的两种基本方法是什么？它们各自是如何工作的？

18.13　当运行在两个分组交换机上的计算机程序交换距离向量信息时，该程序必须有统一的报文格式。请设计一个统一的报文格式的规范。提示：要考虑到不同计算机表示信息方式的不同。

18.14　采用指定的报文格式来完成上题中的程序，对上一题练习进行扩充。看看是否还有其他同学也用该格式来实现程序，这些程序能互相操作吗？

18.15　当一个分组交换机从它的一个邻居收到一个距离向量报文时，该交换机的转发表总要进行改变吗？请解释。

18.16　什么是路由循环问题。

第19章 网络技术的过去与现在

19.1 引言

本书这一部分的各章通过典型的LAN、MAN和WAN，分别介绍了各种有线和无线网络。前几章围绕基本分类介绍了数据通信和数据网络。在这几章之前的内容主要是考虑用于因特网接入技术与用于因特网核心区的技术之间的区别。

多年来，很多网络技术都已经定义了各自的基本类型，一些曾经占据主流的技术现在已经为无人问津的冷门技术，而其他一些技术却继续占据着合适它的地位。这简短的一章，突出了一些主要的技术，并介绍了每种技术的显著特征和特点。文中的例子列举了多种技术，并显示了技术的发展更新是多么的迅速。

19.2 连接与接入技术

前几章介绍了当前最重要的接入与连接技术（DSL和电缆调制解调器），也定义了多种其他的技术，包括在电力线上传送数据的技术和无线接入机制。这些技术可归纳如下。

19.2.1 同步光纤网或同步数字体系

同步光纤网（SONET）和相关的TDM系列最初是设计为一个用来承载数字语音电话业务的系统，该技术当前已成为用于整个因特网的数字线路标准。SONET允许构建一个物理环路来提供冗余。硬件可以自动检测并纠正问题——即使环的一部分损坏了，数据仍然可以通过冗余线路正常传输。有一种称为分插复用器（Add-Drop Multiplexor）的设备，用于将站点连接到SONET环上。之所以使用分插复用器这个名称，是因为它可以插入或者终止已连接到环中另外一个插拔复用器上的一组数据线路。SONET使用时分复用技术将线路复用到底层光纤上。同步数字体系（SDH）则是为配置各种规格的线路提供大家熟悉的标准，例如可以在SONET环上配置出一条T3线路。

19.2.2 光载波

光载波（OC）标准规定了用在光纤SONET环上的信号规格，它能提供比SDH所提供的T系列标准更高的数据传输速率。一家私营公司可能会选择租用一条OC线路来连接公司的两个站点，第一层级的ISP可以使用OC-192（10 Mbit/s）线路，而在因特网的骨干网中可以使用OC-768（40 Mbit/s）线路。

19.2.3 数字用户线路与电缆调制解调器

这两种技术是当前为个人住户和小型企业提供宽带因特网接入的主要手段。数字用户线路（DSL）是利用现有的电话地面线路，而电缆调制解调器技术则是利用现有的有线电视基础设施。DSL提供了1～6Mbit/s的数据传输速率，具体速率取决于交换局和用户的距离。电缆调制解调器提供了高达52Mbit/s的数据传输速率，但其带宽是由一组用户共享的。在光纤到小区或光纤到户技术达到可用水平之前，DSL和电缆调制解调器这两种技术都被看做是过渡性技术。

19.2.4 WiMAX与Wi-Fi

Wi-Fi包括一组无线技术，这些技术已广泛用于为家庭、网吧、机场、旅馆及其他地点提供互特网接入。连续几代的Wi-Fi技术已经增加了整体的数据传输速率。

WiMAX是一种新兴的无线技术，可用来构成MAN。WiMAX提供了接入或者回程[⊖]的能力，并定义了两个版本来支持固定和移动终端。

19.2.5 甚小口径卫星

使用尺寸小于3m的碟形天线的甚小口径卫星（VSAT）技术，由于其占用空间不大，使得卫星通信为个人用户或小型商户提供因特网接入具有可行性。虽然VSAT能提供很高的数据速率，但它却会导致很长的延时。

19.2.6 电力线通信

电力线通信（PLC）使用高频率沿着电力线传输数据。它的出发点还是想利用现有的基础设施来提供因特网接入。虽然此项技术已经作了很多研究，但该技术仍未能获得广泛的部署。

19.3 LAN技术

LAN被发明后，很多组织提出了各种设计方案或建造出各种实验原型。LAN新技术的发展持续了二十年。其中有几种LAN技术深受欢迎并获得了商业上的成功。有趣的是，很多LAN技术已经逐渐聚拢到极少数成熟的技术上，而新的LAN却出人意料。

19.3.1 IBM令牌环

在LAN早期的一些工作中，人们探索将令牌传递作为接入控制机制。IBM选择并发了一种令牌传递的LAN技术，这项技术称为IBM令牌环。IBM令牌环最初版本的运行速率是4Mbit/s。而其竞争对手以太网的运行速率是10Mbit/s。之后，IBM推出了16Mbit/s版本的令牌环。尽管其数据传输速率较低而成本较高，但IBM令牌环仍受到公司信息技术部门的广泛接受，并在多年来一直是一种主要的LAN技术。

19.3.2 光纤和铜导线分布式数据互连

在20世纪80年代末，两种主要的LAN技术（10Mbit/s的以太网和16Mbit/s的IBM令牌环）所提供的数据传输速率，已明显变得不能满足日益增长的需求。光纤分布式数据互连（FDDI）标准正是为了把LAN的数据传输速率提高到100Mbit/s而开发的。当时，设计师们认为更高的数据传输速率需要使用光纤代替铜导线，并建议办公室重新布线，以实现光纤到桌面。此外，FDDI使用一对反向旋转环（counter-rotating rings）来提供冗余度，即如果一个FDDI环被切断，硬件自动环接数据通路而绕开故障点，从而形成一个新的环路，使得数据能够在新的环路中继续传输，保持环可以继续正常工作。最后，FDDI还引入了最早的LAN交换机，使得每台计算机直接连接到一个中心的FDDI装置中。类比，FDDI具有物理的星形拓扑结构和逻辑的环形拓扑结构。

因为它提供了可达到的最高的数据传输速率，并有冗余机制，所以FDDI在数据中心的计算机中逐渐成为一种受欢迎的高速互连设施。然而，高成本以及需要特别的专家来安装光纤，使得很多组织打消了用它来代替铜导线布线的念头。随着快速以太网的研究工作不断获得新的进展，FDDI的支持者又开发了一种可以运行在铜导线布线中的FDDI版本，称为铜导线分布

⊖　从远程供客或接入点返回到提供商中心设施之间的连接。

式数据互连（CDDI），可以为LAN提供100Mbit/s的数据传输速率。最终，以太网技术由于成本较低，从而在LAN的部署中占据了主流地位，而FDDI技术则逐渐消亡。

19.3.3 以太网

从某种意义上说，以太网赢得了竞争并完全主宰了LAN市场。的确，以太网比任何其他类型的LAN都获得了更广泛的应用。从另一个意义上讲，以太网已经完全消失，并被新的技术所代替，而这种新技术仍然称为以太网。我们可以看到，例如，早期以太网中所使用的沉重的同轴电缆及射频信号与千兆以太网所使用的导线及信号之间，几乎没有任何相似性。除了数据传输速率的变化之外，物理的和逻辑的拓扑也改变了：集线器代替了电缆，以太网交换机代替了集线器，而VLAN交换机又代替了交换机。

19.4 WAN技术

人们已经开发了很多的技术用于广域网的实验和生产。本节列举几个例子以说明广域网技术的多样性。

19.4.1 ARPANET

分组交换广域网的历史还不到50年。在20世纪60年代后期，高级研究计划署（Advanced Research Projects Agency，ARPA）为美国国防部提供联网技术研究的资助。ARPA的一个主要研究项目是开发广域网，以确定分组交换技术在军事上是否有价值。该网络称为ARPANET，是最早的分组交换广域网之一。ARPANET把学术界与工业界的研究人员连接起来，虽然从现在的标准看来ARPANET的速度很慢（连接到分组交换机的租用串行数据线其运行速度只有56 Kbit/s），但是从该项目流传下来的概念、算法和术语，直到现在仍然在使用着。

在因特网项目开始时，ARPANET用作骨干网络，研究人员利用它来进行学术交流和实验。在1983年1月，ARPA规定连接到ARPANET的每个用户停止使用原来的协议，开始使用网络间协议。类比，ARPANET就成了第一个因特网骨干网。

19.4.2 X.25

制定国际电话标准的国际电信联盟（International Telecommunication Union，ITU）为广域网技术开发出一种早期的标准，并在公共运营商中深受欢迎。当时，ITU称为国际电报电话咨询委员会（Consultative Committee for International Telephone and Telegraph，CCITT），因此该标准也称为CCITT X.25标准。X.25网络在欧洲比在美国更受欢迎。

X.25使用传统的广域网设计——一个X.25网络由两个或两个以上用租用线路互连的X.25分组交换机构成。计算机直接连接到分组交换机。X.25采用类似电话呼叫的面向连接的模式，即一台计算机在传输数据之前要先打通一个连接。

因为X.25是在个人计算机流行之前发明的，很多早期的X.25网络设计都用来连接ASCII终端和远程分时计算机。当用户通过键盘输入数据时，X.25网络接口捕捉按键，并放在X.25分组中，然后通过网络传输分组。类似地，当运行在远程计算机的一个程序显示输出时，计算机把输出传给X.25网络接口，再把信息放在X.25分组中，并传回到用户屏幕上。虽然电话公司推出了X.25的服务，但从它所提供的性能上来说，这种技术的代价非常昂贵。目前它已经被其他广域网技术所取代。

19.4.3 帧中继

长途运营商已经开发了一系列传输数据的广域网技术，其中之一就是一种称为帧中继（Frame Relay）的技术，它被设计用于接收和发送数据块，其中每个数据块可最多包含8KB的数据。采用这种大数据（以及使用这个名称）的部分动机，是因为发明者设想使用帧中继服务来桥接LAN网段。在两个城市都有办事处的一个组织，可以为每个办事处办理获取帧中继服务，然后就可以使用帧中继服务在办事处站点的LAN网段之间转发分组。设计者选择了面向连接的模式，使得帧中继才能为拥有多个办事处的公司所接受。因此，在出现可替换的更低成本的技术之前，帧中继是深受欢迎的。

由于帧中继的设计是用于处理来自LAN网段数据的，所以设计者设想帧中继的运行速度应该在4～100Mbit/s（这是当时帧中继出现时的局域网速度）之间。但是实际上，帧中继服务的高费用却导致很多用户宁可选择运行在1.5Mbit/s或56Kbit/s速度上的连接手段。

19.4.4 交换式多兆位数据服务

与帧中继相似，交换式多兆位数据服务（SMDS）也是长途运营商提供的高速广域数据服务。SMDS建立在IEEE标准802.6 DQDB基础上，并被看做是ATM的先导。SMDS的设计用于承载数据而非语音流。更重要的是，SMDS经过优化后能以最高速度运行，例如，由于分组的头部信息可能会占据相当数量的可用带宽，所以为了尽量减少头部开销，SMDS采用一个短的头部，并限制每个分组的数据不能超过9188字节。SMDS还定义了一个特殊的硬件接口，该接口可将计算机连接到网络上，而且能使传输数据的速度与计算机移动数据到内存的速度一样快。

顾名思义，SMDS网络的运行速度通常超过1 Mbit/s（即比典型的帧中继连接还要快）。这两种服务的不同之处在于它们可运行的方式，即SMDS是无连接的，这使得其灵活性好。但是，大部分电话公司更愿意接受面向连接的技术，这意味着SMDS不再受欢迎，并已经被取代。

19.4.5 异步传输模式

电信业设计了异步传输模式（ATM）以作为因特网的替代手段，并大力宣传其成果。ATM技术在20世纪90年代刚刚出现的时候，具有宏伟的目标——设计者宣称ATM将取代所有的WAN和LAN技术，从而产生一个完全统一的全球通信系统。除了数据，ATM还可以处理视频传输和传统的语音电话传输。此外，设计者宣称，ATM将扩展到具有远高于其他分组交换技术的数据速率。

ATM引入了一个关键的新思想，称为标记交换（label switching）。ATM属于面向连接的技术，然而，与普通面向连接的技术不同的是，其分组没有地址，相反，每个分组携带一个小ID称为标记（label）。此外，分组每次通过一台交换机时这个标记都可以改变。连接建立后，该连接路径中的每个链接都有一个独一无二的标记，这些标记都被存放在交换机的表格中。当一个分组到达时，交换机查找当前的标记，并替换交换标记。理论上，标记交换在硬件上比传统的转发具有更高的速度。

为了适应所有可能的用途，设计者为ATM增加了很多功能，包括提供端到端可靠服务的机制（例如，有保证的带宽和时延范围）。当开始实现ATM时，工程人员发现，过多的功能意味着硬件构造复杂且成本高昂。此外，用来建立标记交换路径的机制非常麻烦，这一点导致其最终没有被采用。因此，ATM也没有被人们接受，而且事实上已经消失了[⊖]。

⊖ 原书的这句话很容易得有点过头，因为目前ATM与以太网交换技术之间的较量尚未结束，所以还不能过早就认为ATM技术将被别的技术完全取代。——译者注

19.4.6 多协议标记交换

虽然MPLS不是一个网络系统，但它是ATM努力的一个最著成果——工程师在互特网路由器[⊖]中采用了标记交换。与ATM试图完全取代底层硬件不同，多协议标记交换（MPLS）可以作为一个额外的特性在软件中实现。MPLS路由器接收因特网分组，把每个分组分别放入特定的包装，用标记交换在MPLS路径中传输分组，然后解封分组，并继续正常地转发。MPLS专门用在因特网的中心区域，第一层级ISP可以使用MPLS以允许一些分组沿着特殊路径传输（例如，一个付费更高的大客户可以让分组沿着一条更短的路径传输，而这条路径对那些付费较低的客户是不可用的）。

19.4.7 综合业务数字网

在第12章中已经详细介绍了综合业务数字网（ISDN），本章只给出简短的概要。电话公司开发了ISDN以提供比拨号调制解调器的数据速率更高的网络服务。当ISDN首次提出时，128 Kbit/s的速度似乎已经很快了。但在实际应用中，对比其价格，这种技术提供的速度显得很慢。在世界上大部分地区，ISDN已经被能够提供远高于其数据速率的DSL、电缆调制解调器或3G蜂窝系统所取代。

19.5 本章小结

目前，已经有很多网络技术被开发出来了，不过，有些技术太复杂，有些则过于昂贵，而还有一些则缺少实质性的特性。尽管很多技术在商业上取得一些成功，但仍然被其他技术所取代。具有讽刺意味的是，虽然以太网技术已存活了超过30年，但只有它的名称和帧格式被保留下来，而其底层所采用的技术却已完全改变了。

练习题

19.1 SONET是什么？
19.2 用户可以通过什么名称知道DOCSIS技术？
19.3 你认为哪种技术具有更小的时延，是VSAT技术还是WiMAX技术？为什么？
19.4 哪个公司因其令牌环技术而出名？
19.5 哪种技术使得FDDI前景黯淡，并最终取代了FDDI？
19.6 是什么技术已取代了以太网集线器？
19.7 请说出在1983年就采纳因特网协议的一种广域网技术。
19.8 在1980年银行采用了什么广域网技术？
19.9 在网络领域，ATM代表了什么？
19.10 请说出一种起源于ATM且目前还在流行的技术。
19.11 为什么ISDN不能占领大的市场？

⊖ 第20章讲述因特网交换与路由。

第四部分

网络互联

互联网体系结构；编址，绑定，封装以及TCP/IP协议组

第20章 网络互联：概念、结构与协议

20.1 引言

前几章讲述了基本的组网知识，包括在局域网和广域网中使用的硬件设备，以及诸如编址和路由的一般概念。本章开始探讨计算机通信领域中一个新的基本思想——一种能用来将多个物理网络连成一个大型、统一的通信系统的网络互联技术。本章还将讨论网络互联的动机，介绍要用到的硬件设备，描述这些设备连在一起的体系结构，并讨论这个概念的重要意义。这一部分的其余章节要扩展网络互联概念，并提供有关技术方面的更多细节。这里还要介绍各种协议，并解释每个协议如何使用前面章节所述的技术来获得可靠而无差错的通信。

20.2 网络互联的动机

设计每一种网络技术，都必须满足特定的一组约束条件。例如，局域网技术只是设计用在短距离内提供高速通信，而广域网技术则是设计用在广大地区范围内提供通信。因此，

没有任何一种单一的网络技术对于所有的需求都是最好的。

一个有分散联网需求的大单位往往需要多个物理网络。更重要的是，如果该单位为每种用途都选择最适合的网络类型，那么这个单位就会存在多种类型的网络。例如，像以太网这样的局域网技术对于连接同一场所内的计算机，可能是最佳解决方案；但是如果一个城市中某个场所的计算机与另一城市中某个场所的计算机要实现互连的话，就可能要租用数据线路服务。

20.3 全局服务概念

使用多个网络所造成的主要问题是，连接到给定网络的计算机只能与连接到同一网络的其他计算机通信。在20世纪70年代，当一些大单位开始需要多个网络时，这个问题变得越来越突出。这样，单位内的每一个网络就形成一个个孤岛，许多较早安装的系统中，每台计算机都连在单个网络上，职员不得不为每个任务选择一台适当的计算机，这就是说，一个职员被迫从一台计算机跑到另一台计算机去操作多个屏幕和键盘，才能在合适的网络上传递消息。

由于用户必须在每种网络上使用各自所属的计算机，所以工作起来用户既不满意又很低效。因此，大多数现代计算机通信系统都允许任意两台计算机之间能够相互通信，类似电话系统那样任意两部电话机间都能通信。这个被称为全局服务（universal service）的概念是联网的基础。有了全局服务，一个用户在单位里的任何一台计算机上都可以发送消息或数据给其他任何用户，而且，当任务改变时，用户不需要更改计算机系统——所有的信息对所有的计算机都可用。这样，用户的生产率得以提高。概括如下：

支持全局服务的通信系统允许任意两台计算机间进行通信。

20.4 异构网络中的全局服务

全局服务意味着一个单位内只能采用单一的网络技术吗？在使用多种网络技术的多个网络中，有可能提供全局服务吗？由于电气指标的不兼容，我们不可能仅仅通过互连网络之间的导线来形成一个大网络。而且，由于不同的网络技术使用互不兼容的分组格式和编址方案，比如“桥接”这类扩展技术就不能用于异构网络技术。因此，由某种网络技术产生的帧，不能在使用不同技术的网络中传输。要点概括如下：

> 虽然全局服务非常必需，但由于网络硬件、帧结构和物理编址的不兼容性，妨碍了在一个单位内构建包括任意网络技术的桥接网络。

20.5 网络互联

尽管网络技术互不兼容，研究人员仍然设计出一种方案，能在异构网络间提供全局服务。这种被称为网络互联（internetworking）的方案既要使用硬件，也要使用软件。附加的硬件系统用于将一组物理网络互连起来，然后在所有相连的计算机中运行的附加软件，即可提供全局服务。连接物理网络所形成的网络系统被称为互联网络（internetwork）或简称为互联网（internet）。

网络互联相当普遍。特别是互联网没有规模上的限制——既有包含几个网络的互联网，也有包含几千个网络的互联网。同样，互联网中连接到每个网络的计算机数目也是可变的——有些网络没有连接任何计算机，而有些网络则连接了几百台计算机。

20.6 用路由器连接物理网络

用于连接异构网络的基本硬件设备是路由器（router）。在物理上，路由器是专门用来完成网络互联任务的一种专用硬件系统。像桥接器那样，路由器含有处理器和内存，以及用于连接每个网络的单独的输入/输出接口。网络对待路由器的连接与对待任何其他计算机的连接一样。图20-1简单明了地表示出使用路由器实现网络的物理连接。



图20-1 用路由器连接的两个物理网络。对每个网络连接，路由器都有一个单独的接口，计算机连接到各个网络上

由于路由器连接并不限于某种网络技术，图中使用一朵云而不是一条线或一个圆来描绘每个网络。一个路由器可以连接两个局域网、局域网与广域网或者两个广域网。而且，当路由器连接同一基本类型的两个网络时，这两个网络不必使用同样的技术。例如，一个路由器可将一个以太网连接到一个Wi-Fi网[⊖]。这里，每朵云都代表任意的一种网络技术。

概括如下：

> 路由器是一台专门完成网络互联任务的专用硬件系统。路由器可以将多个使用不同技术（包括不同的传输介质、物理编址方案或帧格式）的网络互相连接（互联）起来。

⊖ Wi-Fi（Wireless Fidelity）网，即无线保真网，属于在办公室和家庭中使用的短距离无线技术，是计算机（特别是笔记本电脑）无线上网的主要技术。该技术使用2.4GHz附近的频段，传输速率可达11Mbit/s。目前这种网络可使用的标准有两个，分别是IEEE802.11a和IEEE802.11b。多数使用后者。——译者注

20.7　互联网体系结构

一个单位如果要根据需要来选择适当的网络技术，使用路由器就可能做到。路由器可以把所有不同的网络连接起来而形成单个互联网。例如，图20-2说明了如何使用3个路由器把4个任意的物理网络连接成为一个互联网。

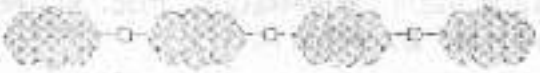


图20-2　用3个路由器连接4个物理网络形成一个互联网

虽然图中表示的每个路由器只有两个连接，但商用的路由器可以连两个以上的网络。因此，用单个路由器就可以把我们上例中的所有4个网络连接起来。然而，一个单位只使用单个路由器连接所有的网络的情况很少，这有两个原因：

- 由于路由器的CPU和内存要用来处理每个被传递的分组，而单个路由器的处理器还不足以处理太多的网络之间要传递的通信量。
- 冗余度能改善互联网的可靠性。为了避开网络中单个点上的故障，协议软件一直监视着互联网的连接情况，当某个网络或路由器发生故障时，就指令路由器沿另一条迂路传输通信业务。

因此，在规划一个互联网时，一个单位必须选择一个能满足该单位对可靠性、容量和价格方面要求的设计。特别是，互联网拓扑结构的具体细节常常要取决于物理网的带宽、预期的通信量、单位的可靠性要求，以及路由器硬件设备的费用和性能。

概括如下：

> 互联网由通过路由器连接起来的一组网络所构成。在规划互联网方案的时候，允许使用单位按使用要求来选择网络的数量和类型、用于互连的路由器数量以及具体的互连拓扑结构。

20.8　实现全局服务

网络互联的目标是实现通过异构网络提供全局服务，为了在互联网中的所有计算机之间提供全局服务，路由器必须能将某个网络中源端发出的信息转发到另一网络中的目的端。这一任务很复杂，因为不同网络所使用的帧格式和编址方案不同。因此，在计算机和路由器上都需要有协议软件，才有可能实现全局服务。

后续章节将详细描述因特网（Internet）[⊖]协议软件，阐述因特网协议如何克服帧格式和物理地址的不同，使在不同技术的网络之间实现通信成为可能。在考虑因特网协议如何工作之前有一点很重要，就是要理解互联系统对连接的计算机所带来的影响。

20.9　虚拟网络

总的来说，互联网软件提供了一个连接着许多计算机的单一而无缝的通信系统。这种系统提供全局服务，即每台计算机只分配一个地址，任何计算机都能发送分组到其他计算机。而且，互联网协议软件隐藏了物理网络连接、物理地址及路由信息等方面的细节——用户和应用程序都无须知道基础物理网络以及连接它们的路由器的情况。

⊖　正如前面已述，大写的术语Internet特指全球互联网及相关的协议（而internet是指通常的互联网）。

我们说互联网是一种虚拟网络（virtual network）系统，是因为这个通信系统只是一种抽象而已，也就是说，虽然硬件和软件联合提供了一个单一网络的错觉，但这一网络并不存在。图20-3解释了虚拟网的概念和相应的物理结构。

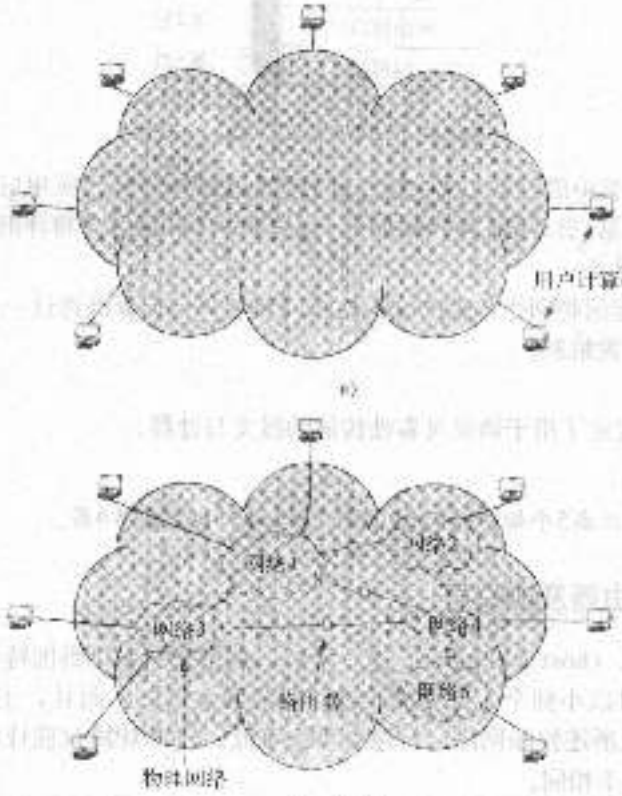


图20-3 互联网概念：a）提供给用户和应用程序的单一网络错觉（抽象），b）通过路由器连接而形成的底层物理网络结构

20.10 网络互联协议

虽然人们提议了多种协议来实现网络互联，但是只有一组协议标准最终胜出并被广泛应用。这一组协议的正式名称是TCP/IP互联网协议（The TCP/IP Internet Protocol），多数网络专业人员将其简称其为TCP/IP[⊖]。

TCP/IP的开发与全球因特网的发展同时起步。事实上，设计TCP/IP的研究人员也提出过前面所述的互联网体系结构。20世纪70年代，几乎就在开发局域网的同时，开发TCP/IP的工作就开始了，并一直持续到20世纪90年代的早期，那时，因特网已经开始商业化运作了。

20.11 TCP/IP分层结构综述

回顾第1章，因特网协议使用了一个五层的参考模型，如图20-4所示。

⊖ TCP和IP是协议组中两个最重要的协议名称的首字母缩写，各位读者直接读T-C-P-I-P即可。



图20-4　TCP/IP参考模型的5个层次

我们已经讨论了其中的三层。本书第一部分的相关章节讨论了应用层，第二部分和第三分中的相关章节讨论了第1层和第2层中的协议。这一部分的相关章节将详细地讨论剩下的两层。

第3层：互联网络层

第3层（IP）规定因特网中传输的分组格式以及从一台计算机通过一个或多个路由器到达最终目的地的分组转发机制。

第4层：传输层

第4层（TCP）规定了用于确保可靠性传输的报文与过程。

概括如下：

因特网协议被组织成5个概念层，其中IP在第3层，TCP在第4层。

20.12　主机、路由器及协议层

TCP/IP定义主机（host computer）这一术语，是指任何连接到因特网并运行应用程序的计算机系统。主机可以小到个人计算机，也可以大到大型机。而且，主机的CPU可快可慢，内存可大可小，主机所连接的网络运行速度可高可低。TCP/IP协议能让任何一对主机互相通信，即使它们的硬件不相同。

主机和路由器都需要TCP/IP协议软件。但是，路由器不必使用所有层的协议。特别是路由器软件不需要第5层对应于应用程序（如文件传送）的协议，因为路由器不必运行常规的应用程序[⊖]。下面的章节将更详细地讨论TCP/IP协议软件，并解释因特网是如何分层工作的。

20.13　本章小结

逻辑上，互联网呈现为一个单一的无缝的通信系统。互联网上的任一对计算机可以互相进行通信，如同它们连接在单个网络上一样，也就是说，一台计算机可以发送一个分组给连接到互联网的任何其他计算机。物理上，互联网是由被称为路由器的设备互连起来的多个网络的集合。每个路由器都是一台连接两个或多个网络的专用设备，专门用于在它连接的各个网络之间传输互联网报文。

连接到互联网上的计算机称为主机。主机可以是大型的计算机（如超级计算机），也可以是小型的计算机（如个人计算机）。每个主机连接到互联网中的一个物理网上。

由于互联网运行协议软件，使人产生认为它是一个单一通信系统的错觉，互联网中的每个主机和路由器都必须运行这一软件。该协议软件隐藏了底层物理连接的细节，而着重于将每个分组转送到目的地。

为网络互联而开发的最重要协议，就是人所共知的TCP/IP互联网协议，通常被简称为

⊖ 在一些路由器的确运行了特殊的应用软件，允许管理员远程控制它。

TCP/IP。除了有很多专用网络使用它外，TCP/IP在全球因特网上也使用了多年。

练习题

20.1 互联网会被一种单一的联网技术所取代吗？为什么会或为什么不会？

20.2 提供全局服务的主要困难是什么？

20.3 一个单位不使用单个路由器来连接所有网络的两个理由是什么？

20.4 如果一个给定路由器可最多连接到K个网络，那么要连接N个网络需要多少个路由器R？请用K和N做参数，写出R的表达式。

20.5 用户视互联网为一单个网络。真实情况是怎样的呢？用户计算机实际连接的是什么？

20.6 在采用TCP/IP互联协议的因特网五层参考模型中，每一层的目的是什么？

第21章 网际协议编址

21.1 引言

前一章介绍了用路由器连接多个物理网络的互联网物理结构。本章开始讲述协议软件，该软件使因特网[1]呈现成为一个单一的无缝的通信系统。这里主要介绍网际协议（IPv4）采用的编址方案，并讨论无类地址中地址掩码以及子网编址问题[2]。

后续章节再进一步讨论IP，每章分别详细讨论协议的一个方面。可将这几章视为一个整体，它们定义了IP协议并解释IP软件怎样使计算机能通过因特网来交换数据。

21.2 虚拟因特网的地址

回顾第20章所讲，网络互联的目标是要提供一个无缝的通信系统。为达到这个目标，因特网协议必须屏蔽物理网络的具体细节，并提供一个单一的、大型的虚拟网络设施。从应用程序的角度来看，虚拟因特网像任何网络一样操作，允许计算机发送和接收数据分组。因特网和物理网的主要区别是：因特网仅仅是设计者想象出来的抽象物，完全由软件产生。因此，设计者可以自由地选择独立于底层硬件细节的地址、分组格式和传递技术。

编址是因特网抽象的一个关键组成部分。为了呈现出一个单一的系统，所有主机必须使用统一的编址方案，而且每个地址必须是唯一的。虽然每台计算机都有一个MAC地址，但是这种网络地址并不能满足这个要求，因为一个因特网可包括多种物理网络技术，而每种技术都定义了自己的MAC地址格式。

为保证统一编址，IP协议定义了一种与底层MAC地址无关的编址方案。IP地址作为因特网的目的地址使用，类似于在一个局域网内把MAC地址作为目的地址来使用那样。为了在因特网上发送分组，发送方把目的地的IP地址放在分组中，将它提交给IP协议软件去发送。IP协议软件使用目的IP地址将分组转发到目的地计算机。

IP编址方案的优势在于它的统一性：任意一对应用程序不需要知道对方的网络硬件或所采用的MAC地址就能相互通信。这种想象太完美了，以至于有些用户发现协议地址是由软件提供而不是计算机系统的一部分时感到很吃惊。有意思的是，我们还将知道协议软件的许多层也要使用IP地址。概括如下：

> 为了在因特网中提供统一编址，IP协议定义了一个抽象的编址方案，给每台主机分配一个唯一的地址。应用程序间使用IP地址进行通信。

21.3 IP编址方案

IP标准规定：每台主机分配一个唯一的32位二进制数作为该主机的网际协议地址

---

[1] 因特网可以认为是一种企业性的互联网特例。本书的后续内容都是针对因特网来阐述的。——译者注

[2] 除非特别声明，本书中的网际协议和IP均是指IP协议的第4版。

(Internet Protocol address)，常简称为IP地址或因特网地址[5]。在因特网上发送一个分组时，发送方必须指定它自己的32位IP地址（源地址）和接收方的IP地址（目的地址）。

概括如下：

> 网际协议地址（IP地址）是分配给主机并用于该主机进行所有通信活动的一个唯一的32位二进制数。

21.4 IP地址的层次结构

类似于广域网中使用的分层地址结构，每个32位IP地址是分成两部分：前缀部分和后缀部分。与广域网中地址前缀标识一个分组交换机不同，IP地址前缀部分标识计算机从属的物理网络，IP后缀部分标识该网络上的一台计算机。也就是说，因特网的每一物理网络都分配一个唯一的网络号（network number），而网络号在从属于该网络的每台计算机的IP地址中作为前缀出现。同一物理网络上每台计算机则分配一个唯一的地址后缀。

为了保证唯一性，因特网上的两个网络不能分配同一个网络号，同一网络上的两台计算机也不能分配同一个后缀。假设一个互联网由3个网络组成，它们可分配网络号1，2，3。从属于网络1的3台计算机可分配后缀1、3、5；同时，从属于网络2的3台计算机可分配后缀1、2、3。分配的后缀值不要求保持连续。

IP地址的编址方案保证了两个重要性质：

- 每台计算机只分配给一个唯一地址（即一个地址从不分配给多台计算机）；
- 虽然网络号分配必须全球一致，但后缀可由本地分配，不需全球一致。

因为整个地址包括前缀和后缀，它们分配时保证唯一性，所以第一个性质得到保证；如果两台计算机连接在不同的物理网络，它们的地址会有不同的前缀；如果两台计算机连接在同一个物理网络，它们的地址会有不同的后缀。因此，分配给一台计算机的地址始终是唯一的。

21.5 IP地址的原分类

IP的设计人员确定了IP地址的长度并决定将它分为前缀和后缀两部分之后，接着就必须决定每部分要包含多少位。前缀部分需要足够的位数，才足以分配唯一的网络号给因特网上的每一个物理网络；后缀部分也需要足够位数，才能对连接于某一网络的每一台计算机都能分配一个唯一的后缀。简单的选择是不行的，因为在某一部分增加一位就意味着在另一部分减少一位。选择大的前缀可容纳大量网络，但限制了每个网络的规模；选择大的后缀意味着每个物理网络能包含大量计算机，但却限制了网络的总数。

由于因特网包括任意的网络技术、构成因特网的网络中可能包含少量的大型物理网络和很多的小型网络。因此，设计人员选择了一个能满足大网和小网组合情况的折中编址方案。这个原始的编址方案我们称之为有类IP编址（classful IP addressing）。这个方案将IP地址空间划分为3个基本类，每类分别有不同长度的前缀和后缀。

地址的前4位确定地址所属的类别，并确定如何将地址的其余部分划分成前缀和后缀两部分。图21-1所示为5个地址类，用于决定类别的前导位以及前缀和后缀的划分。该图按照TCP/IP协议惯例，数字位从左到右位顺序，且以0作为第一位。

---

[5] 这3个术语在本书中作为同义词使用。



图21-1　IP地址原分类方案的5个类

尽管有类编址方案已经被其他方案所取代，但是D类地址依然用于组播。它允许传递给一组计算机，即每个组播地址对应于一组计算机。一旦建立起组播组，发送到该组播地址的任何分组都要被传递到该组中的每一台主机。在实际应用中，因特网组播从未在全球规模内实现，这也意味着组播只能局限于一定的场合范围内。

概括如下：

IP地址的原分类方案将主机地址分为几个类。D类地址依然用于组播，但组播并非全球规模内有效。

21.6　点分十进制数表示法

虽然IP地址是32位二进制数，但用户很少以二进制方式去输入或读出它的值。相反，当用户跟地址打交道时，软件使用一种更易于理解的表示法，称为点分十进制数表示法（dotted decimal notation）。其做法是将32位二进制数中的每8位为一组，用它的十进制数表示，利用句点分割各组数字。图21-2表示出几个二进制数及其等价的点分十进制数形式的例子。

32位二进制数	等价的点分十进制数
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

图21-2　32位二进制数及其等价的点分十进制数表示的例子

点分十进制数表示法把每个8位组作为一个无符号整数来处理[⊖]。如同图中最后一例所示，当8位组内所有位都是0时，最小可能值为0；当8位组内所有位都是1时，最大可能值为255。这样，点分十进制地址的范围是从0.0.0.0到255.255.255.255，其中D类组播地址的范围则是从224.0.0.0到239.255.255.255。

概括如下：

点分十进制数表示法是一种语法表达形式。当人跟IP地址打交道时，IP软件用它来表示32位的二进制数值。点分十进制数表示法将每个8位组用等价的十进制数表示，并用句点隔开每个8位组的十进制数。

⊖　IP协议使用术语8位组（octet）而不使用字节（byte）是因为一个字节的大小依赖于所使用的计算机。虽然8比特字节已经成为事实上的标准，但是术语8位组的含义却是十分明确的。

21.7 地址空间的划分

原来的有类编址方案出现在人类发明PC、局域网得到广泛使用和很多公司建立自己的计算机网络之前。它将地址空间划分成大小不等的类，设计者通过这种不等量的划分方案来适应各种情况。例如，虽然A类地址最多只能有128个网络，但它却包含了所有地址的一半。这样做的目的是为了使一些主要的ISP能构建大型网络，使得这些网络能连接数以百万计的计算机。与此类似，C类地址的目的是允许一个单位能将少量的计算机连接到局域网。图21-3总结了每类地址中的最大网络数和每一个网络中的最大主机数。

地址类别	前缀位数	最大网络数	后缀位数	每个网络可有的最大主机数
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

图21-3 3个基本IP地址类中所含的网络数和每个网络所含的主机数

21.8 地址的授权

在整个互特网中，分配给网络的网络前缀必须唯一。因此，一个叫因特网名字与号码分配公司（Internet Corporation for Assigned Names and Numbers，ICANN）的中心组织成立了，它负责分配地址和协调争议。ICANN并不负责分配具体的网络前缀，相反，它授权注册商（registrar）来做这件事情。注册商向ISP（因特网服务提供商）提供有效的地址块，再由ISP向用户提供地址。因此，为了获得网络前缀，公司通常都需要联系一家ISP[⊖]。

21.9 子网与无类编址

随着因特网的发展，原来的有类编址方案逐渐受到了限制。每个组织都要求A类或B类地址，这样他们将有足够多的地址来应对未来的发展，这导致了其中很多地址并未得到利用。虽然还有很多C类地址未分配，但只有少量的组织愿意使用它们。

现在已经发明了两种机制来克服这种限制：

- 子网编址。
- 无类编址。

这两种机制的相互关系非常密切，可认为它们都是一种抽象的两个部分。它们不再采用3个不同的地址分类，而是直接利用前缀和后缀在地址的任意位置上进行分界。子网编址最初用在那些连接到全球因特网的大机构中，无类编址则把这种方法扩展到了整个因特网。

为了更好地理解为什么允许在任意位置上进行分界，我们考虑一个需要处理网络前缀的ISP。假设该ISP的一个客户需要一个能容纳35台主机的网络前缀。使用有类编址方案，ISP需要向它分配一个C类地址前缀。而实际上，它只需要6位的主机后缀即可表示所有的主机地址[⊖]，这意味着254个主机后缀中的219个不会分配给主机[⊜]。换句话说，C类地址空间中的大部分地址被浪费了。无类编址方案则提供了一个比较好的解决办法，它允许ISP向客户分配一

⊖ 23章解释了一台计算机如何获取唯一的网络前缀。

⊜ 原书为4位，正确的应为6位。——译者注

⊜ 一个C类地址有256种可能的后缀，但全0和全1的后缀保留用于表示子网和子网广播。因此得到254。本章后续章节类同。

个26位长度的网络前缀，因此后缀长度是6位，这样就只有27个地址未被使用。

换一个角度来看待子网和无类编址的问题。假设ISP拥有一个C类地址前缀，有类编址中需要将整个前缀分配给一个使用单位，而无类编址方案中，ISP可以将这个前缀划分为几个更长的前缀，然后再将它们分配给客户。图21-4展示出无类编址方案中ISP如何将一个C类前缀划分为4个更长的前缀，使每个前缀对应一个能容纳62个主机的网络。

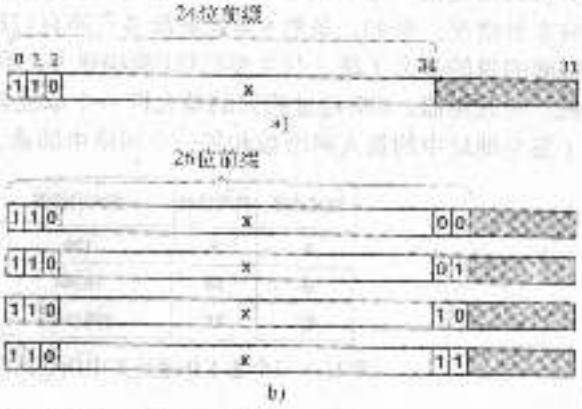


图21-4　a) C类地址的前缀；b) 同一个前缀划分为4个无类前缀

在上图中，每个前缀的主机部分用灰色表示。原始的C类地址有8位后缀，而每个无类地址中有6位后缀。假设原始的C类前缀是唯一的，那么每个无类地址前缀也将是唯一的。因此，在不再浪费地址空间的情况下，ISP可将4个无类前缀分别分配给不同的客户。

21.10　地址掩码

IP地址怎样才能在任意的位置上进行分界呢？无类和子网编址方案要求处理地址的主机和路由器存储一点附加的信息，以便指定网络前缀与主机后缀之间的准确界线。为了标记精确的边界，IP使用了一个叫做地址掩码（address mask）的32位长的值。地址掩码最早叫做子网掩码（subnet mask）。地址掩码中值为1的位标识网络前缀，值为0的位标识主机部分。

为什么要把分界值作为一个位掩码来保存呢？这样做主要是为了提高地址计算的效率。特别是我们将会看到，当主机和路由器处理一个IP分组的时候，需要比较路由表中地址的网络前缀部分的值，而利用掩码位可以使得这个比较操作的速度加快。为了理解这一点，假设路由器上给出了一个目的地址D、一个32位值的网络前缀N和一个32位的地址掩码M。又假设N的前面部分包含网络前缀，剩余位为0，为了检测目的地是否位于指定的网络，路由需要测试条件：

$$N == (D \& M)$$

这表明，路由器利用掩码M对地址D执行“逻辑与”操作，以便将地址D中的主机后缀对应的码位置成零，然后再用这个结果和网络前缀N进行比较。

这里以下面的32位网络前缀作为例子：

10000000 00001010 00000000 00000000

用点分十进制数表示为128.10.0.0。再考虑一个前16位为1后16位为0的32位掩码，用点分十进制数表示为255.255.0.0：

11111111 11111111 00000000 00000000

今有一个32位目的地址128.10.2.3，相应的二进制数是：

10000000 00001010 00000010 00000011

将目的地址与地址掩码进行“逻辑与”运算，提取出高位的16位，它产生的二进制结果是：

10000000 00001010 00000000 00000000

它正好等于网络前缀128.10.0.0。

21.11 CIDR表示法

无类编址方案的正式名称为无类域间路由（Classless Inter-Domain Routing，CIDR）。这个名字不是特别适合它，因为CIDR仅仅规定了编址和转发过程。设计者在设计CIDR编址方案时，希望人们能很容易地指定一个掩码。为了理解指定掩码的难度，考虑图21-4b中例子所需要的掩码。该掩码有26个1，后面紧跟6个0。以点分十进制数表示的话，该掩码为：

255.255.255.192

为了易于指定和解释掩码的值，人们修正了点分十进制数表示法。修正后的版本称为CIDR表示法（CIDR notation）。地址和掩码通过在一个点分十进制地址的后面附加一个斜杠符和一个十进制数来表示。该十进制数指明了掩码中从地址最左边开始的连续的1的位数。也就是说，通用的格式为：

ddd.ddd.ddd.ddd/m

其中，ddd是地址中与8位组等价的十进制数值，m是掩码中1的个数。因此，我们可以这样写：

192.5.48.69/26

它指定了一个26位的掩码。图21-5列出了用CIDR表示的地址掩码和对应的用点分十进制数表示的掩码。注意，一些CIDR地址掩码正好对应于原来的有类编址方案。

长度（CIDR）	连正掩码	备注
/0	0 . 0 . 0 . 0	全0（等效于无掩码）
/1	128 . 0 . 0 . 0	
/2	192 . 0 . 0 . 0	
/3	224 . 0 . 0 . 0	
/4	240 . 0 . 0 . 0	
/5	248 . 0 . 0 . 0	
/6	252 . 0 . 0 . 0	
/7	254 . 0 . 0 . 0	
/8	255 . 0 . 0 . 0	缺A类掩码
/9	255 . 128 . 0 . 0	
/10	255 . 192 . 0 . 0	
/11	255 . 224 . 0 . 0	
/12	255 . 240 . 0 . 0	
/13	255 . 248 . 0 . 0	
/14	255 . 252 . 0 . 0	
/15	255 . 254 . 0 . 0	
/16	255 . 255 . 0 . 0	缺B类掩码
/17	255 . 255 . 128 . 0	
/18	255 . 255 . 192 . 0	
/19	255 . 255 . 224 . 0	
/20	255 . 255 . 240 . 0	
/21	255 . 255 . 248 . 0	
/22	255 . 255 . 252 . 0	
/23	255 . 255 . 254 . 0	
/24	255 . 255 . 255 . 0	缺C类掩码
/25	255 . 255 . 255 . 128	
/26	255 . 255 . 255 . 192	
/27	255 . 255 . 255 . 224	
/28	255 . 255 . 255 . 240	
/29	255 . 255 . 255 . 248	
/30	255 . 255 . 255 . 252	
/31	255 . 255 . 255 . 254	
/32	255 . 255 . 255 . 255	All to host specific mask

图21-5 用CIDR和点分十进制数表示的地址掩码列表

21.12　CIDR举例

作为一个CIDR的例子，我们假定一个ISP有以下的地址块可供分配：

128.211.0.0/16

再进一步假设它有两个客户，其中一个客户需要12个IP地址而另一个需要9个，它可以向一个客户分配CIDR网络前缀：

128.211.0.16/28

同时向另一个客户分配CIDR网络前缀：

128.211.0.32/28

虽然两个客户都有相同大小的掩码（28位），但它们的网络前缀不同，分配给前一位客户的网络前缀的二进制值是：

10000000 11010011 00000000 0001 0000

分配给后一位客户的网络前缀的二进制值是：

10000000 11010011 00000000 0010 0000

显而不会产生混淆——每位客户都有唯一的网络前缀。更重要的是，ISP保留了大部分原始地址块，可以分配给其他客户。

21.13　CIDR主机地址

下面考虑如何计算一个CIDR地址块中地址的范围。一旦客户从ISP那里得到一个CIDR网络前缀，他就可以进行主机地址的分配。例如，假设某单位被分配得到之前所述的网络前缀128.211.0.16/28。图21-6说明该单位将拥有4个二进制位作为主机地址域，并用二进制数和点分十进制数表示了最高和最低的地址。本例中排除了主机位全1和全0的主机地址。



图21-6　对一个/28前缀进行CIDR地址的图示

图21-6也表明了无类编址的缺点——因为主机后缀可以在任意的分界上开始，所以它的值不容易用点分十进制数的形式读出来。例如，它跟网络前缀组合在一起时，14种可能的主机后缀所形成的点分十进制值就是从128.211.0.17到128.211.0.30。

21.14 特殊的IP地址

除了给每台计算机分配一个地址外，用地址来表示网络或一组计算机也很方便。IP定义了一组特殊地址形式，称为保留（reserved）地址。也就是说，特殊地址从不分配给主机使用。本节阐述每个特殊地址形式的语法和语义。

21.14.1 网络地址

其实从图21-6中就可以看出定义特殊地址的一个动机——用一个地址来表示分配给某个特定网络的前缀是很方便的。IP保留主机地址为0的地址，用它来表示一个网络。因而，地址128.211.0.16/28表示一个网络。其原因就是它的第28位之后都是0。网络地址绝不应该作为目的地址出现在分组中[⊖]。

21.14.2 直接广播地址

有时候，需要将一个分组副本发送给某个物理网络中的所有主机。为了更容易实施这种广播，IP为每个物理网络定义了一个直接广播地址（directed broadcast address）。当一个分组要发送到一个网络的直接广播地址时，就只有一个分组副本会通过互特网到达该网络，然后被传送到该网络中的每一台主机。

在网络前缀后面增加一个全1后缀，便形成了网络的直接广播地址。因而，主机后缀为全1的地址被保留了——如果管理员不小心将全1后缀地址分配给了一台计算机，计算机上的软件就可能出现故障。

广播是如何工作的呢？如果网络硬件支持广播，那么直接广播就利用硬件广播能力来实现。如果某个特定的网络不支持硬件广播，则软件必须分别为网络上的每台主机发送一个该分组的副本。

21.14.3 有限广播地址

有限广播（limited broadcast）这一术语，是指直接在本地物理网内的广播，不太严格地说，广播被限制在“单根导线”上。由一台还不知道网络号的计算机去启动系统时，可以采用有限广播。

IP保留了由32位的1所组成的地址用于有限广播。因此，IP对所有要发送给全1地址的分组将通过本地网络进行广播。

21.14.4 本机地址

由于每个因特网分组都包含源地址和目的地址，因此计算机在发送或接收互特网分组前需要知道它自己的IP地址。在第23章中，我们将学习TCP/IP协议组中这样一个协议，即当计算机启动时，借助于该协议，它就可以自动获得IP地址。有趣的是，启动协议也要使用IP来通信，而在使用这个启动协议时，计算机还没有一个正确的IP源地址。为了处理这一情况，IP保留了全0地址来表示本台计算机（this computer）。

21.14.5 回送地址

IP定义了一个回送地址（loopback address）用于测试网络应用程序。在编写完成一个网络应用程序后，程序员常常要采用回送测试的方法对程序进行预调试。为了实现回送测试，程序员必须准备好要通过网络进行通信的两个应用程序。每个应用程序都包括与TCP/IP协议软件进行交互所需要的代码。程序员不是在不同的计算机上执行每个程序，而是在同一台计

⊖ 21.16节讨论了定义到广播地址，它是一种非标准的例外形式。

算机上运行两个程序的进程，并指令它们在通信时使用回送IP地址。当一个应用进程发送数据给另一个应用进程时，数据向下穿过协议栈到达IP软件，IP软件再把数据向上通过协议栈返回给另一个应用进程。因此，程序员可很快地测试程序逻辑而无须使用两台计算机，也无须通过网络传输数据。

IP保留A类网络前缀"127/8"以供回送测试时使用，带有"127"的主机地址无关紧要，所有的主机地址都做同样处理。根据习惯，程序员经常使用主机号1，形成最普遍使用的回送地址形式，即127.0.0.1。

在回送测试时，没有数据会离开计算机。IP软件将数据从一个应用进程转发到另一个应用进程，因此，回送地址永远不会出现在网络中所传输的任何数据单元中。

21.15　小结特殊IP地址

图21-7中的表概括了特殊IP地址的各个形式。

前　缀	后　缀	地址类型	用　途
全0	全0	本计算机	自举期间使用
网络	全0	网络	标识网络
网络	全1	直接广播	在指定网络上广播
全1	全1	有限广播	在本地网络上广播
127	任意	回送	测试

图21-7　特殊IP地址形式汇总表

我们说过，特殊地址是被保留的，决不可分配给任何主机使用。而且，每个特殊地址只限于某一种用途，例如广播地址永远不能作为源地址出现，全0地址在主机完成了启动过程并获得IP地址后，就不能再使用了。

21.16　伯克利广播地址形式

美国加州大学伯克利（Berkeley）分校开发并发行了TCP/IP协议的一个早期实现版本，作为BSD UNIX的一部分[⊖]。BSD版本包含非标准特点，影响了许多随后的实现。这个实现版本改为采用全0主机后缀（即与网络地址一致）来代表直接广播地址，而不是采用全1后缀。这一地址形式被非正式地称为伯克利广播（Berkeley broadcast）。

不幸的是，许多计算机制造商的早期TCP/IP软件都继承了伯克利版本，有些站点仍然还在使用伯克利广播。有些TCP/IP实现则包含有一个配置参数，可供选择使用TCP/IP标准形式还是伯克利形式，也有很多实现版本既采纳标准地址形式又采纳伯克利广播地址形式。因此，网络管理员必须为每个网络选择所要采用的地址形式（如果允许直接广播的话）。

21.17　路由器与IP寻址原理

除了给每个主机分配一个无特网地址外，IP规定路由器也应分配IP地址。事实上，每个路由器分配了两个或更多的IP地址，路由器连接的每个网络都需要一个。要理解为什么这样，回想如下两个事实：

- 一个路由器与多个物理网络相连接。

⊖　BSD代表"伯克利软件发行"。

• 每个IP地址只包含一个特定物理网络的前缀。

因此，对一个路由器而言，单用一个IP地址并不够，因为每个路由器连接到多个网络而每个网络都有唯一的前缀，IP方案可通过一个基本原理进行解释：

> 一个IP地址并不标识一台特定的计算机，而是标识一台计算机与一个网络之间的连接。一台连接多个网络的计算机（例如，路由器）必须为每个网络连接分配一个IP地址。

图21.8用一个例子来说明这一思想，图中表示出给连接着3个网络的两个路由器分配IP地址的情况。

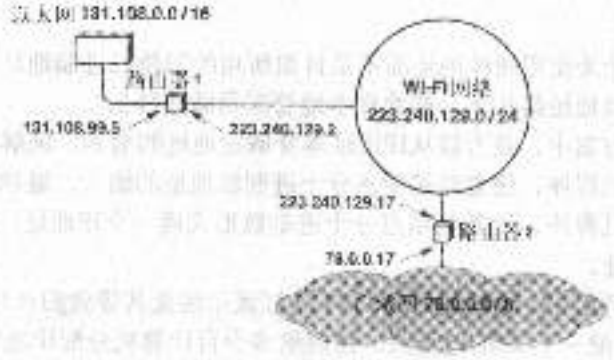


图21-8 给两个路由器分配IP地址的情况

IP并不要求给路由器的所有接口分配同样的后缀。例如上图中连接到以太网和Wi-Fi网的路由器，具有后缀“99.5”（连接到以太网）和“2”（连接到Wi-Fi网）。然而，IP并不反对为所有的连接使用同样的后缀。因此，例子中表示出管理员为从Wi-Fi网连到广域网去的路由器的两个接口，选用了相同的主机号“17”。实际上，使用相同的主机号对管理人员有利，因为单个数字较容易记忆。

21.18 多穴主机

一个主机能有多个网络连接吗？能。一台主机被连接到多个网络的时候，称这台主机是多穴的（multi-homed）。多穴主机有时用来增加可靠性——如果一个网络发生故障，主机仍能通过第二个连接到达因特网。多穴主机也可用来提高性能——连接到多个网络使它能维持随时发送信息并避开有时会拥塞的路由器。像路由器一样，多穴主机也要有多个协议地址，即每个网络连接需有一个IP地址。

21.19 本章小结

为呈现出一个大型无缝的网络，因特网使用统一的编址方案。每台计算机分配给一个协议地址，用户、应用程序及大多数协议在通信时都要使用协议地址。

网际协议规定了编址方案。IP把每个五特网地址划分成两层：地址的前缀表示计算机所连接的网络，后缀标识这个网络中的一台特定计算机。为了确保这一地址在整个特定因特网中的唯一性，必须由一个中心组织来分配网络前缀。一旦分配了一个前缀，本地网络管理员便能给该网络中的每个主机分配一个唯一的后缀。

IP地址是一个32位长的二进制数。原编址方案将IP地址分成几类，其中组播类仍在使用中。无类和子网编址可以实现在任何位值上划分前缀与后缀的分界。为实现这种子网和无类

编址（CIDR）方案，随同每个地址还必须保存一个32位的掩码。掩码的前缀部分为全1，后缀部分为全0。

IP标准定义了一系列有特殊意义的保留地址。特殊地址可被用于回送测试、在本地网络内广播和在远地网络内广播。

虽然认为用一个IP地址来指定一台计算机很方便，但要清楚，每个IP地址所标识的应该是一台计算机与一个网络的连接。路由器和多穴主机连接到多个物理网络上，一定要有多个IP地址。

练习题

21.1 能将IP重新设计为使用硬件地址而不是目前所用的32位二进制地址吗？为什么？

21.2 在因特网的分层地址结构下，能允许本地管理员做些什么？

21.3 在原有类编址方案中，是否能从IP地址本身确定地址的类别？试解释。

21.4 编写一个计算机程序，使它能接受点分十进制数地址的输入，显示32位二进制数输出。

21.5 编写一个计算机程序，使其能用点分十进制数形式读一个IP地址，并能判定该地址是否是一个组播地址。

21.6 编写一个计算机程序，使其能翻译CIDR斜杠表示法及其等效的点分十进制值。

21.7 如果ISP给你分配一个/28的地址块，你能给多少台计算机分配IP地址？

21.8 如果ISP提供一个/17的地址块每月收费N美元，而一个/16地址块每月收费1.5N美元，对每台计算机来说，哪一种方案更便宜？

21.9 形式为1.2.3.4/29的CIDR前缀是否有效？为什么？

21.10 假设你是一个拥有/24地址块的ISP。试解释你能否满足你客户的请求，他需要为255台计算机申请地址（提示：考虑特殊地址）。

21.11 假设你是一个拥有/22地址块的ISP，需要向4个客户分配地址块，每个都需要为60台计算机申请地址。请给出你的CIDR分配方案。

21.12 假设你是一个拥有/22地址块的ISP。你能否满足以下6个客户的地址申请请求，他们分别需要为9、15、20、41、128和260台计算机申请地址。如果可以，如何分配地址？如果不可以，那又是为什么？

21.13 编写一个计算机程序，它能用CIDR斜杠表示法读出一个地址，并以二进制形式打印出所读出的地址和掩码。

21.14 编写一个计算机程序，它能读取用CIDR斜杠表示法输入的网络前缀和希望产生的主机数。假设这个请求已经递交给了拥有此前缀的ISP，ISP能借助于这个程序产生一个新的CIDR前缀，以此来满足这个请求而不会造成地址浪费。

21.15 编写一个计算机程序，它能用CIDR斜杠表示法读出32位的主机地址和掩码，并能判断它是否是一个特殊地址。

21.16 什么是伯克利广播地址？

21.17 一个路由器需要分配多少个IP地址？试解释。

21.18 主机可以拥有一个以上的IP地址吗？试解释。

第22章 数据报转发

22.1 引言

前面几章介绍了因特网的体系结构和因特网编址。本章将讨论因特网中基本的通信服务，介绍通过因特网传输的分组格式，并讨论数据报封装、转发、分片和重装等重要概念。后续章节将进一步讨论构成完整服务所需的其他协议。

22.2 无连接服务

网络互联的目的是为了提供一种分组通信系统，在这种系统中，运行在某台计算机上的程序能够向运行在另一台计算机上的程序传送数据。在一个设计完善的互联网中，底层物理网络对应用程序来说是透明的——这些应用程序能够收发数据而又无须了解很多细节，比如本计算机所在的局域网情况、目的计算机所在的远程网络情况以及两者之间的互连情况。

设计者在设计互联网时要考虑一个最基本的问题，那就是网络需要提供什么样的服务，特别是，设计者还必须决定是为程序提供面向连接（connection-oriented）的服务还是无连接（connectionless）的服务，或是两者都提供。

TCP/IP的设计者作出了包括无连接服务和面向连接服务的选择。他们选择了无连接的基本传送服务，并在这种无连接的下层服务之上增加了可靠的面向连接的服务。这样的设计取得了成功，并为所有的因特网通信奠定了基础。

22.3 虚拟分组

无连结服务其实是分组交换的一种扩展——这种服务允许发送方通过互特网传输单独的分组数据，每一个分组都独立地在网上传输，它本身包含了标识接收方的信息。

分组是如何在因特网上传输的呢？一般的回答是，由路由器将每个分组从一个网转发到另一个网。源主机产生一个分组，将目的地址放入分组的头部，然后将分组送往邻近的路由器。当路由器收到一个分组时，它利用分组中的目的地址来选择下一个路由器并将分组转发给它。最终，分组到达了能直接将分组传递给最终目的地的那个路由器。

互特网分组采用什么格式呢？由于因特网是由帧格式彼此不兼容的异构网络构成的，所以因特网不能简单地直接采用其中任何一种硬件帧格式。另外，路由器也不能简单地重新格式化帧的头部，因为两个网络可能使用不兼容的地址格式（例如，输入帧中的地址对于另一个网络来说可能是没有意义的）。

为了克服异构性，网际协议软件定义了一种独立于底层硬件的分组格式，它是一种能够完整无损地通过底层硬件传输的通用的（universal）、虚拟的（virtual）分组，正如“虚拟”一词所蕴涵的意思，因特网分组格式不直接与任何底层硬件挂钩。实际上，底层硬件根本不理解和不认识因特网分组。又正如“通用”一词所蕴涵的意思，因特网上的每一台主机或路由器中都含有能理解这种分组的协议软件。

概括如下：

因为因特网包含不兼容的网络技术，所以它不能直接采用任何特定的硬件帧格式。为了克服异构性，网际协议定义了一种与硬件无关的分组格式。

22.4 IP数据报

TCP/IP协议使用IP数据报（IP datagram）这个名字来指因特网分组。令人奇怪的是，IP数据报竟然与硬件帧有同样的基本格式：IP数据报也是以一个头部开始，后跟数据区（载荷）。图22-1表示了这种数据报格式。

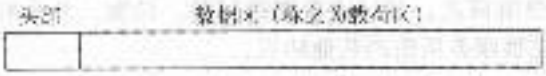


图22-1　由头部和后续的数据区构成的IP数据报的通用格式

概括如下：

在TCP/IP网络上发送的分组叫做IP数据报。每个数据报由一个头部和紧跟在其后的数据区组成。其中数据区也称之为载荷区。

数据报中所携带的数据量是不固定的，由发送方根据特定的用途选择合适的数据量。例如，传输键盘输入信息的应用程序可以将每个键入字符放进单独的数据报中发送；而传输大文件的应用程序则可能要发送装载有大数据量的数据报。

要点　数据报的大小由发送数据的应用程序来确定。数据报的大小可变这一特点，使得IP可以适应各种应用。

在目前的IP版本（版本4）中，一个数据报少则包含1个字节的数据，多则包含64K字节的数据（包括头部）。在大多数数据报中，头部比数据区要小得多，头部代表着额外的开销。这就是说，由于数据报头部的长度固定，发送大数据报意味着在单位时间内能传输更多的用户数据（即更高的吞吐率）。

22.5 IP数据报头部格式

数据报头部包含什么内容呢？与帧的头部类似，数据报头部中包含了在网络中转发它时所需的信息，特别是，头部包含源地址（最初的发送方）、目的地址（最终的接收方）和指示数据区中数据类型的一个域（字段）。数据报头部中的地址是IP地址，而发送方与接收方的MAC地址并不出现在数据报中。

IP数据报头部中的每个域都有固定的长度，这样做可以提高头部处理过程的效率。图22-2所示为IP数据报头部所包含的各个域。随后的文字给出了对每个域的详细解释。

版本号：数据报以4位长的无区版本号开始（上图标出版本4的头部）。

头部长度：4位长，它指定以32位为度量单位的头部长度。如果报文头部中没有可选项和填充，它的值就是5。

服务类型：8位长，携带该数据报提供服务的类型（实际上很少使用）。第28章在讲述区分服务（DiffServ）时，将解释服务类型域的含义。

总长度：16位长，它指定以字节数计量的数据报总长度，包括头部长度和数据长度。

标识：每个数据报都被分配一个互不相同的16位长的标识。通常这些标识在数值上都是

前后连续的。在重装数据报的时候，通过标识域来识别属于同一个数据报的所有片段。

标志：3位长，它的各个值用于指示数据报已分片，未分片，以及如果进行了分片，则这一片是否对应于原始报文中的最后一片，等等。

片偏移：13位长，指示本片中的数据在所属的原始报文中的位置。该域的值乘以8就是偏移量的大小。

生存时间：由发送方初始指定的8位长整数值，每个路由器处理数据报时，会将它的值减1。该值被减为0时，路由器会丢弃该数据报，同时给源主机发回一个出错报文。

类型：8位长的域，指定载荷数据的类型。

头部校验和：16位长，所有头部域的补码校验和，它根据第8章第8.12节的算法8.1计算得到。

源IP地址：源发端的32位因特网地址（中间路由器的地址不会出现在数据报头部）。

目的IP地址：最终目的端的32位因特网地址（中间路由器的地址不会出现在数据报头部）。

IP可选项："可选项"头部域用于控制路由和数据报的处理。大多数数据报都不含可选项，这意味着头部中忽略了"IP可选项"域。

填充：如果可选项达不到32位的整数倍，就加入全0位来填充，以保证头部长度为32位的倍数。



图22-2 IPv4数据报头部的各个域

22.6 IP数据报转发

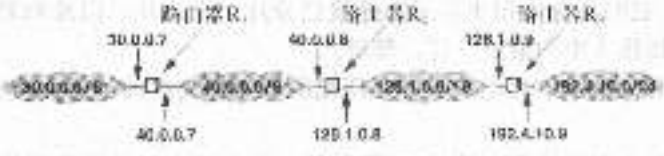
数据报沿着从源地址到最终目的地址的一条路径通过因特网传输。中间会经过很多路由器。因特网采用"下一站转发"（next-hop forwarding）的方式来转发IP数据报，即路径上的每个路由器收到一个数据报时，先从头部取出目的地址，根据这个地址决定数据报该发往的下一站，然后路由器就将此数据报转发给这个下一站，它可能就是最终目的地，也可能是另一个路由器。

为了能高效地选择下一站，每个IP路由器都会使用一张转发表（forwarding table）。路由器启动时，需对转发表进行初始化，而当网络的拓扑发生变化或某些硬件发生故障时，它必须更新转发表。

概念上，转发表中的每一项都指定了一个目的地以及为到达这个目的地所要经过的下一站。图22-3表示出一个互联网络的例子，这个网络有3个路由器，给出了其中一个路由器的转发表内容。

图中，每个路由器都有两个IP地址，每个接口对应一个。路由器R_2与40.0.0.0/8网络和128.1.0.0/16网络直接相连，因此分配了地址40.0.0.8和128.1.0.8。回忆一下，路由器所有接口中的IP地址并不要求有相同的后缀。这里，网络管理员为路由器的每个接口选择了相同的后

缀，这样就方便了管理网络的人。



a)

目的地	掩码	下一站
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	直接传送
128.1.0.0	255.255.0.0	直接传送
192.4.10.0	255.255.255.0	128.1.0.9

b)

图22-3　a)互连4个物理网络形成互联网的例子；b)路由器R_2中的转发表

最值得关注的一点是转发表的大小，它对全球互特网来说是很重要的：

> 由于转发表中的每个目的地对应于一个网络，所以转发表中的项数正比于互特网中的网络个数，而不是主机个数。

22.7　网络前缀提取与数据报转发

利用转发表为数据报选择下一站的过程叫做转发（forwarding）。回顾在第21章所讲述的，查表时要使用掩码以便从一个IP地址中取出它的网络前缀部分。假设路由器遇到一个包含目的地址D的数据报，转发功能必须在转发表中找到指明去往D的下一站的那一项。为了做到这一点，软件检测转发表中的每一项，利用掩码提取地址D的前缀，并把结果与目的地址进行比较，如果相同，数据报就转发到该表项中所指的下一站去。

位掩码表示法使得提取网络前缀的工作非常有效——软件将掩码与数据报目的地址D进行逻辑"与"（and）运算。于是，检测表中第i项的计算过程可以表述如下：

如果 (掩码[i] & D)==目的地[i]就转发到下一站[i]

举一个例子，考虑一个目的地址为192.4.10.3的数据报。假设该数据报到达了一个包含图22-3所示转发表的路由器R_2上，并假设软件按顺序搜索表中每一项。对于第一项，因为255.0.0.0 & 192.4.10.3不等于30.0.0.0，故匹配失败。同样，第二、三项都不符合，路由软件最终选择了地址为128.1.0.9的下一站，因为

255.255.255.0 & 192.4.10.3 == 192.4.10.0

22.8　最长前缀匹配

图22-3中所示只是一个小小的例子，而在实际中，因特网的转发表可能十分庞大，转发算法也十分复杂。例如，因特网转发表中可以包含一个默认表项，为所有没有被指定的目的地提供一条路径，这与第18章中所描述的广域网转发过程类似。此外，因特网转发表也允许管理员指定一个特定主机路由（host-specific route），让去往某一特定主机的流量沿着一条不同于去往同一网络的其他主机的路径前进（即指定一个拥有32位掩码的转发表项，这个表项要求整个主机地址与之匹配）。

因为地址掩码可以重叠，这就导致了因特网转发过程的一个重要特性。例如，假设某路

由器的转发表中包含下面两个网络前缀表项：

128.10.0.0/16
120.10.2.0/24

考虑一下，如果一个到达的数据报要去往128.10.2.3，会发生什么情况呢？令人惊奇的是，以上给出的两个表项的匹配过程都会成功。也就是说，一个16位掩码的逻辑"与"运算会得到128.10.0.0，而一个24位掩码的逻辑"与"运算会得到128.10.2.0。这种情况下，应该使用哪一个表项呢？

为了解决重叠地址掩码产生的歧义，因特网转发过程使用最长前缀匹配（longest prefix match）规则，也就是说，替代原先的按任意顺序检查表项的做法，转发软件首先安排拥有最长前缀的表项去进行检查。在上述例子中，因特网转发表将选择128.10.2.0/24对应的表项。

要点 为了解决多个表项匹配某个目的地而导致的歧义，因特网转发过程首先对最长前缀表项进行检查。

22.9 目的地与下一站地址

数据报头部中的目的地址与其被转发的下一站地址之间，到底有什么关系呢？数据报中的"目的地址"域所含的地址是最终目的地址，它不会随数据报在因特网上的传递而改变。当路由器收到一个数据报时，会利用这个最终目的地址D来计算数据报将发往的下一个路由器的地址N。虽然这个数据报被路由器转发到下一站地址N，但其头部中仍保持着目的地址D。也就是说：

数据报头部中的目的地址总是指最终目的地址。在每一个中间节点上，路由器都要计算下一站的地址，但下一站的地址并不出现在数据报头部。

22.10 尽力传递

除了定义因特网数据报格式外，IP还定义了通信的语义，并使用尽力而为（best effort）这个词汇来描述所提供的服务。从本质上讲，这个标准显然规定IP会尽力地尝试传递每个数据报，但并不保证它能处理好所有问题。特别是，IP标准承认会出现以下问题：

- 数据报重复；
- 延迟或乱序传递；
- 数据损坏；
- 数据报丢失。

IP竟然还会出现这些问题也许看起来令人很奇怪，然而这有一个重要的原因：我们设计IP的目的是让它能在任何类型的网络上运行。从前面的章节可知，网络设备会受到噪声干扰，从而导致数据的损坏或是丢失。在路由发生改变的系统中，沿着某条路径前进的分组可能会比沿另一条路径前进的分组花费更多的时间，从而导致乱序传递。

要点 由于IP是针对各种类型的网络硬件而设计的，其中包括那些可能会产生问题的硬件，因此IP数据报就有可能会发生丢失、重复、延迟、乱序或损坏等问题。

值得庆幸的是，我们将看到TCP/IP协议栈已经包含了额外的协议来处理其中的很多问题。我们也将学习到，比起那些具有检测和纠正能力的服务来，有些应用程序则更愿意采用"尽力而为"的服务。

22.11　IP封装

在物理网络不了解数据报格式的情况下，数据报怎样才能在该网络中传输呢？答案就在于要采用一种所谓的封装（encapsulation）技术。当要把一个IP数据报封装进一个帧中时，就将整个数据报放进帧的数据区（载荷）。网络硬件会像对待普通帧一样对待这种已包含数据报的帧。事实上，硬件不会去检查或改变帧的数据区内容。图22-4说明了这种封装情况。

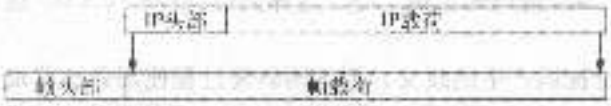


图22-4　被封装在硬件帧中的IP数据报

接收方如何知道输入帧的数据区中是否含有IP数据报还是其他数据呢？为此，发送方和接收方必须就“帧类型”域中的值达成一致。当发送方将一个数据报放入帧里时，发送方机器上的软件必须在“帧类型”域内置入保留给IP数据报的特定值。当这样一个帧到达接收方后，根据它的“帧类型”域值就可知道帧中含有一个IP数据报。例如，以太网标准规定携带IP数据报的以太网帧中的“帧类型”域的值是0x0800。

携带有IP数据报的帧同样要有一个目的地址，因此封装过程除了将数据报放入帧数据区外，还要求发送方提供数据报送往的下一站MAC地址。为了得到这个MAC地址，发送方机器上的IP软件必须由下一站的IP地址找到它（下一站）的等效MAC地址，然后将这个MAC地址作为帧头部中的目的地址。下一章将讲述用于完成此过程[3]的ARP协议。

概括如下：

> 为了便于在物理网络中传输，数据报被封装在一个帧里，帧中的目的地址是数据报应该送往的下一站的MAC地址；这个地址是把下一站的IP地址翻译成等效的MAC地址而得到的。

22.12　通过因特网传输

每当数据报要作一次传输时，需进行一次封装处理。发送方在选好下一站之后，将数据报封装到一个帧里，并通过物理网络传给下一站。当帧到达下一站时，接收软件从帧中取出数据报，然后丢弃这一帧。如果数据报必须通过另一个网络转发时，就要生成一个新的帧。图22-5表示出一个数据报从源主机出发，在通过3个网络和两个路由器到达目的主机的过程中，是如何进行封装和解封装的。每个网络都可能使用不同的网络硬件技术，这就意味着它们的帧格式和帧的头部长度也可能各不相同。

如图所示，主机和路由器只将数据报保存在内存中，而不必保存附加的帧头信息。当数据报要通过一个物理网络时，才会被封装进一个适于该网络传输的帧里，帧头部的长度取决于相应的网络技术。例如，如果网络1是以太网，帧1的头部就是以太网头部。类似地，如果网络2是Wi-Fi网络，则帧2的头部就是Wi-Fi网络的头部。

有一点很重要，即在通过因特网的传输过程中，帧的头部并没有累积起来。帧到达的时候，数据报从输入帧中取出，然后被重新封装成另一个输出帧。因而，当数据报到达它的最终目的地时，帧头部就是数据报到达的最后那个网络所封装的头部。去掉头部后，剩下的就是原始的数据报。

---

[3] 即所谓的“地址绑定”（binding）过程。——译者注

要点 当数据报以一个网络帧的形式到达时，接收方将其从帧的数据区中提取出来，并将该帧的头部丢弃。

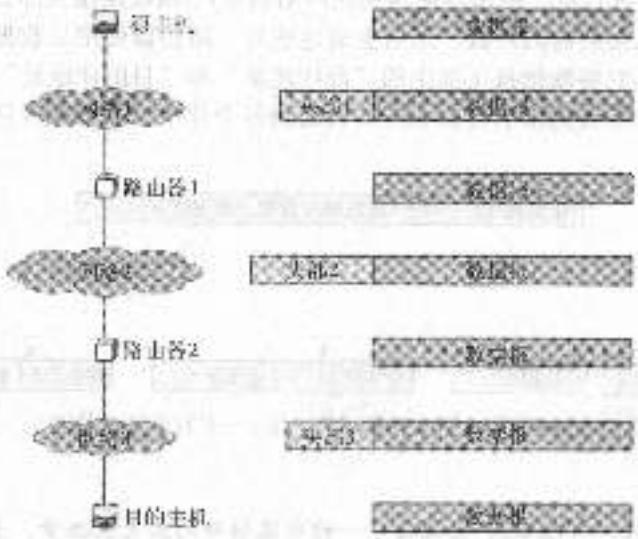


图22-5 在因特网上传送的IP数据报

22.13 MTU和数据报分片

每一种物理网络都规定了一帧所能携带的最大数据量。这一限制被称为最大传输单元（Maximum Transmission Unit，MTU）。对于MTU这一限制，不存在什么例外——网络硬件在设计上都不允许接受或传输数据量大于MTU的帧，因而一个数据报必须小于或等于物理网络的MTU，否则就不能被封装传输。

在包括各种异构网络的互联网中，MTU大小的限制可能会引起一个问题。具体来说，由于一个路由器可以连接多个分别具有不同MTU值的网络，从一个网络上接收到的数据报可能因为尺寸太大而无法在另一个网络上发送。例如，如图22-6所示，一个路由器互连了两个网络，这两个网络的MTU值分别为1500和1000。



图22-6 一个路由器连接两个具有不同MTU值的网络

图中主机H_1连在MTU值为1500的网络1上，它只能发送最多1500字节的数据报。主机H_2连在MTU值为1000的网络2上，因此它不能发送大于1000字节的数据报。如果H_1将一个1500字节的数据报发给H_2，路由器R将无法封装这个数据报使它在网络2上传送。

为了解决由于MTU不同而产生的问题，IP路由器采用了一种叫分片（fragmentation）的技术。当一个数据报长度大于前方网络的MTU值时，路由器会将数据报分成若干较小的部分，称为片（fragment），然后再将每片独立地进行封装并发送出去。

令人惊奇的是，每个片具有与数据报一样的格式——只是它头部的"标志"域中的值标识了它是一个片还是一个完整的数据报[⊖]。在片的头部中，还包含有给最终目的主机用于重

⊖ 数据报头部格式参见本章的图22-2。

装这些片以便重新生成原数据报的有关信息。另外，头部的“片偏移”域指出了该片在它所属的原数据报中的位置。

在对一个数据报分片时，路由器使用相应网络的MTU和数据报头部长度来计算每片所能携带的最大数据量以及所需的片数，然后生成这些片。路由器使用原数据报头部中的域来生成片的头部。例如，它将数据报头部中的“源IP地址”和“目的IP地址”域复制到片的头部中。最后，路由器从原数据报中复制相应的数据到每个片中，并开始传送。图22-7表示了这一处理过程。

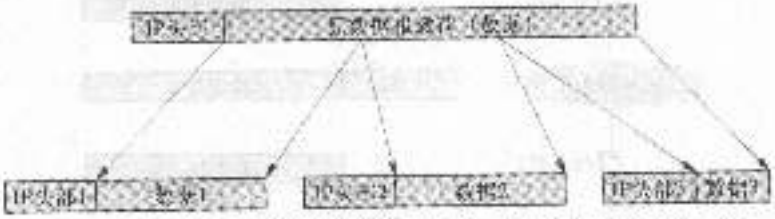


图22-7　被分成3个片的IP数据报，最后一片比其他两片要小一些

据括如下：

任何网络都有一个MTU，它规定了一帧所能携带的最大数据量。当路由器收到一个数据报，但这个数据报长度比前方网络的MTU大时，路由器将数据报分成若干较小的片（fragment），每一片都使用IP数据报格式，但只携带了原数据报的一部分数据。

22.14　分片数据的重装

在所有片的基础上重新产生原数据报的过程，叫做重装（reassembly）。由于每个片都以原数据报头部的一个副本作为开始[1]，因此都有与原数据报相同的目的地址。另外，在含有最后一块数据的片的头部中，设置有一个特别的码位作为标志，这样执行重装的接收方就能知道是否所有的片都成功地到达了。

有趣的是，IP规定只有最终目的主机才能对片进行重装。例如，请看图22-8中的例子：

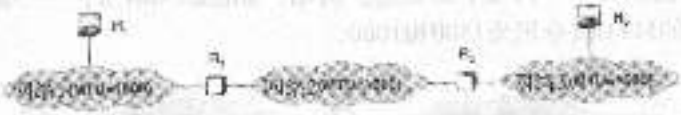


图22-8　由两个路由器连接起来的3个网络示意图

在图中，如果主机H要发送一个1500字节长的数据报给H_2，路由器R_1会把数据报分为两片转发给路由器R_2，R_2不进行片的重装，只是利用片中的目的地址像以往那样转发这些片，最终目的主机H_2搜集了这些片之后，将它们重装起来以生成原来的数据报。

要求在最终目的地重装数据报的做法有两大好处：

- 首先，这样做可以减少路由器中的状态信息数量，当转发一个数据报时，路由器不需要知道它是不是一个片。
- 其次，这样做允许动态改变路由。如果一个中间路由器要进行重装操作，那就要求所有的片都须到达这个路由器才行。

因此，通过将重装操作推后到目的地进行，IP就可以自由地将数据报的不同片沿不同的路

[1] 原数据报与片数据的头部中不同的域仅是指示分片的那个域。

经传输。也就是说，因特网可以在任何时候改变路由（例如，绕开一个出现硬件故障的节点）。

22.15 分片数据报的收集

前面说过，IP并不保证传送成功，所以与其他数据报的转发一样，个别的片可能会丢失或不按次序到达目的地。更重要的是，如果一个源主机将多个数据报发给同一个目的地，那这些数据报的多个片就可能以任意的次序到达。

IP软件如何重装这些乱序的片呢？发送方将一个唯一的标识号放进每个输出数据报的"标识"域中。当一个路由器对一个数据报分片时，就会将这一标识号复制到每一片中。接收方即可利用收到片的标识号和IP源地址来确定该片属于哪个数据报。另外，"片偏移"域也可以告诉接收方片中数据在原数据报中的位置。

22.16 片丢失的后果

由于IP并不保证片的成功传送，所以如果底层网络遗失了分组，则封装在其中的数据报（或片）也就遗失了。当一个数据报的所有片都到达以后，目的主机才能重装该数据报。然而可能产生的问题是：当一个数据报的一个或多个片到达的时候，很可能仍有一些片被延迟或丢失了。虽然这时数据报还不能被重装，但接收方仍须保留所有已收到的片，以防未到达的片可能只是被延迟了而已。

接收方不能将一些片保留任意长的时间，因为它们会占用大量的内存资源。为了避免耗尽内存，IP规定了保留片的最大时间。当数据报的某一片第一个到达时，接收方启动一个重装计时器（reassembly timer）。如果数据报的所有片在规定时间内到达，接收方就取消计时，重装数据报。否则，到了时间而所有片还未到齐，接收方会丢弃已到达的片。

IP重装计数器的结果只是全有/全无（all-or-nothing）这两种情况：要么所有的片都到达且IP重装数据报，要么IP丢弃不完整的数据报。实际上，没有任何机制让接收方去告知发送方哪些片已经送达目的地，因为发送方本身并不知道有关分片的事情，这是IP的设计使然。而且，如果发送方重发该数据报，而路由却可能不同，这意味着每次传输不会总是通过同样的路由器。因此，无法保证被重发的数据报还会跟上次一样地被分片。

22.17 分片再分片

分片之后，路由器将每一片转发给它的目的地。如果某个片偶然遇到一个MTU值更小的网络时，那怎么办呢？分片方案本身设计得很周到，它允许片本身还能再被分片。路径上的另一个路由器会将片分成更小的一些片。如果互联的网络设计得很糟糕，每个网络按MTU从大到小依次连接，则路径上的每个路由器都必须对片进行再分片。当然，设计者小心一点的话可以防止因特网上发生这样的情况。

无论如何，IP都不会去区分是原来的片还是再分的子片，接收方也并不知道收到的是一个由数据报分出来的片，还是一个已经被多个路由器经多次分出的再分片，它都同等地对待所有片。这样做的优点是：接收方不需要先重装子片以后才能重装原数据报，这样就可以节省CPU时间，减少了每一片头部中所需的信息量。

22.18 本章小结

网际协议定义了因特网上传输的基本单元——IP数据报。每个数据报类似于一个硬件帧，

都是由头部与其后的数据区构成，就像硬件帧那样。其头部包含了将数据报传输到特定目的地去所需的信息。与硬件帧不同的是，数据报头部所含的地址是IP地址而不是MAC地址。

路由器中的IP软件利用转发表来决定数据报发送的下一站。转发表中的每一项对应于一个目的地网络，这就使得转发表的规模与因特网中的网络数目成正比。要选择一条路径的时候，IP软件只要将目的地址的网络前缀与表中的每一项进行比较即可。为了避免歧义，IP规定如果转发表中存在两个匹配指定目的地的表项，则应该根据最长前缀匹配原则来转发。

虽然IP软件要为数据报选择发往的下一站，但这个下一站地址并不出现在数据报头部中。相反，头部中总是放着最终目的地的地址。

IP数据报封装在帧中传送。每一种网络技术都定义了一个分组所能携带的最大数据量——MTU（最大传输单元）。当数据报超过网络规定的MTU时，IP将数据报分割成片。在必要情况下，片可能还会进行再分片。最终目的地使用一个计时器重装这些片。如果一个或多个片在计时器超时之前还未到齐，接收方丢弃已经到达的片。

练习题

22.1 设计者在设计网络时要考虑的两种基本通信服务形式是什么？

22.2 因特网能容纳各种拥有不同分组格式的异构网络，它是如何设计实现的呢？

22.3 编写一个计算机程序，从IP数据报中取出源地址和目的地址，以点分十进制数表示法打印输出。

22.4 编写一个计算机程序，从IP数据报头部中提取所有的域，以十六进制数的形式或点分十进制数的形式打印出它们的值。

22.5 IP数据报的最大长度是多少？

22.6 编写一个计算机程序，输入图22-3b中的IP转发表以及一系列的目的地址，对于每一个目的地址，该程序顺序搜索转发表，找到正确的下一站，然后输出结果。

22.7 如果一个数据报包含一个8位长的数据值且没有可选项，则头部里的“头部长度”域和“总长度”域的值分别是多少？

22.8 如果转发表中的两个前缀同时匹配一个指定的目的地址，转发算法会使用哪一项？

22.9 IP数据报中的目的地址会指向一个中间路由器的地址吗？试解释。

22.10 假设两个路由器被错误地配置，以至对某个目的地D产生了闭合的路径环。解释一下为什么目的地为D的数据报不会永远地在这个路径环中传送。

22.11 当IP数据报通过因特网传输时，会发生什么问题？

22.12 IP数据报被放在帧的什么位置进行传输？

22.13 如果我们捕获一个正在因特网中间的某个物理网络中传递的IP数据报，请问该数据报的前面会出现多少个帧头呢？

22.14 网络的MTU是指什么？

22.15 如果一个数据报的数据长度为1480字节，它必须在一个MTU等于500的网络中发送，那会发送多少个片呢？试解释。

22.16 在因特网中，分片在哪里进行重装？

22.17 重装分片的时候，主机如何确定输入的片属于同一个数据报？

22.18 如果一个分片丢失了，接收方是否会请求一个新的副本？试解释。

22.19 阅读RFC 1149和1217，它们是严格的网络标准吗（提示：注意日期）？

22.20 构建一个能进行随机丢弃、复制和延迟分组等操作的因特网仿真网关。

第23章　支持协议与相关技术

23.1　引言

本书这一部分的几章讨论因特网及相关技术。在讲述了网络互联的基本概念和因特网的体系结构后，接下来讲述了IP编址和无类编址方案、IP数据报的格式，以及IP数据报的转发过程。前一章介绍了数据报封装、分片和重装的知识。

本章继续讨论网络互联的问题，介绍4种关键的支撑技术，即地址绑定、差错报告、协议自举和地址绕接。每一种技术都试图解决一个小问题，结合其他协议的功能，每种技术对完善因特网的整体功能都作出重要的贡献。后面的章节将针对传输层协议和因特网路由协议，继续延伸对网络互联问题的讨论。

23.2　地址解析

回顾第22章所述，一个数据报在因特网上传输时，初始发送方和所经路径上的每个路由器都会利用数据报中的目的IP地址来选择下一站地址，将数据报封装在硬件帧中，然后就在这个网络上传输这种帧。转发过程中最关键的一步是地址翻译，即转发过程要用到IP地址，而在物理网络上传输时却必须包含下一站的MAC地址。因此，IP软件必须将下一站的IP地址翻译成等效的MAC地址。原理是：

> IP地址是由协议软件提供的一种抽象地址，因为物理网络的硬件设备不知道如何根据IP地址来定位一台计算机，所以一个帧在被发送前必须先将下一站的IP地址翻译成等效的MAC地址。

将计算机的IP地址翻译成等效的硬件地址的过程，叫做地址解析（address resolution），即将IP地址解析为正确的MAC地址。地址解析是一个网络内的本地行为，只有两台计算机都连在同一物理网络时，一台计算机才能解析另一台计算机的地址——一台计算机无法解析远地网络[1]上的MAC地址，即地址解析总是限于单一网络内，例如，考虑图23-1所示的简单互联网的情况。



图23-1　由3个物理网络组成的互联网例子，每个网络都连接有计算机

在图中，如果路由器R1要向路由器R2转发数据报，R1需要将R2的IP地址解析为对应的MAC地址。与此类似，连接到同一物理网络上的主机A和B也需要这一过程。如果主机A上的应用进程需要向主机B上的应用进程发送数据，A上的协议软件必须将B的IP地址解析为B的

[1] 即另一个物理网络。——译者注

MAC地址，并直接利用此MAC地址发送数据帧。

但是，如果主机A上的应用进程要向连接在远地网络的主机F上的应用进程发送报文，主机A上的协议软件将不会去解析F的地址。主机A上的协议软件首先确定分组必须经过路由器R_1，然后解析R_1的地址，R_1上的软件经计算得知R_2是下一站路由器，于而解析R_2的地址，类似地，最后由R_2去解析F的地址。

概括如下：

> 协议地址与硬件地址之间的映射过程叫做“地址解析”（address resolution）。当主机或路由器需要向同一物理网络内的另一台计算机发送分组时，它必须进行地址解析，一台计算机永远不会去解析连接在远地网络上的计算机地址。

23.3　地址解析协议

软件使用什么算法来将高层协议地址翻译成硬件能理解的地址呢？答案取决于协议和硬件的编址方案。在因特网中，我们只涉及IP地址的解析问题，而几乎大多数硬件都已经采用了48位的以太网编址方案，所以也就只有一种地址解析方案占据主导地位。这就是起初设计用于以太网的地址解析协议（Address Resolution Protocol，ARP）。

ARP的设计思想很直观。假设计算机B想解析计算机C的IP地址，计算机B广播一个请求说“我想查找拥有IP地址C的计算机所对应的MAC地址”。这个广播报文只在本网络上传输。当计算机C收到请求报文的一个副本时，直接向B回复一个报文说“我就是拥有IP地址C的计算机。我的MAC地址是M”。图23-2说明了这种报文的交换过程。

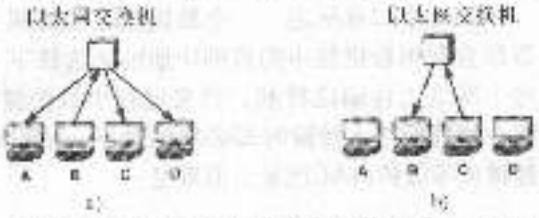


图23-2　计算机B要解析计算机C的地址时，ARP报文的交换过程

图中表明，虽然ARP请求报文会发送给网络上的所有计算机，但不是所有的计算机都会作出回答。我们将会看到，发送者在广播请求中提供了一些信息，所有的计算机在处理这个请求时都会收到这些信息。

23.4　ARP报文格式

ARP并不局限于针对IP和以太网，设计者还创建了一种通用的ARP报文格式。这样，与规定一个固定报文格式的做法不同，ARP标准描述了一种ARP报文的通用格式，并规定了这种格式如何去适应每一种协议地址和每一种网络硬件类型。让ARP报文适应硬件的原因是，ARP的设计者意识到他们无法为硬件地址域选择一个固定的长度，因为新的网络技术不断涌现，它们的地址长度会超过所设计的长度。因此，设计者在ARP报文的开始处引入一个固定长度的域，用于指定所使用的硬件地址长度。例如，当在以太网中使用ARP时，硬件地址长度定为6个字节，因为一个以太网地址是48位长。为了提高ARP的通用性，设计者为协议地址和硬件地址都引入了一个地址长度域。

需要指出的一点是：ARP协议并不局限于IP地址或指定的硬件地址——从理论上讲，该协议也可用于将任意高层地址与任意硬件地址进行绑定。实际上，ARP的通用性并没有充分发挥出来，大多数ARP实现的都是针对IP地址和以太网地址的绑定。

概括如下：

虽然ARP报文格式对任何协议和硬件地址都足充分通用的，但ARP几乎总是应用于IP地址与48位以太网地址的绑定。

图23.3给出了一个应用于版本4的IP地址（4字节）和以太网硬件地址（6字节）的ARP报文格式。图中的每一行对应ARP报文中的32位。后面几段文字解释了每个域的含义。



图23.3 用于绑定IPv4地址与以太网硬件地址的ARP报文格式

硬件地址类型——16位域，指定所使用的硬件地址类型，以太网对应的值为1。

协议地址类型——16位域，指定所使用的协议地址类型，IPv4对应的值为0x0800。

硬件地址长度——8位长整数，指定以字节计量的硬件地址长度。

协议地址长度——8位长整数，指定以字节计量的协议地址长度。

操作——16位域，指定报文是一个请求报文（值为1）还是一个应答报文（值为2）。

发送方硬件地址——包含长度由“硬件地址长度”域指定的发送方硬件地址。

发送方协议地址——包含长度由“协议地址长度”域指定的发送方协议地址。

目标方硬件地址——包含长度由“硬件地址长度”域指定的目标方硬件地址。

目标方协议地址——包含长度由“协议地址长度”域指定的目标方协议地址。

如图所示，每个ARP报文中包含两个地址绑定的域。一个绑定对应于发送方，另一个绑定对应于接收方，接收方在ARP中可做目标（target）。当一个请求发出后，发送方并不知道目标的硬件地址（这就是所要请求获得的信息）。因此，ARP请求报文中的“目标硬件地址”域可以用零填充，因为这个内容尚未获得。在响应报文中，目标就是针对发送请求的计算机来说的，所以在响应报文中的目标地址配对毫无用处——这些域只是从早期的协议版本中继承下来的。

23.5 ARP封装

ARP报文在物理网络上传输时，它被封装在一个硬件帧内，即把它当做正在传输的数据来处理，就像传输IP数据报那样——底层网络不会去分析ARP报文或是解释其中域的内容。图23.4说明了ARP在以太网帧中的封装过程。



图23.4 封装在以太网帧中的ARP报文

帧头中的类型域（type field）指明帧中含有一个ARP报文。发送方在发送前必须为该域指定相应的值，接收方必须检查每个输入帧中的类型域。以太网用类型值0x806来表示ARP报文。ARP请求与ARP响应使用同样的类型值。因此，帧类型并不能区分ARP报文本身的各种类型——接收方必须检查报文中的“操作”域来确定它是一个请求还是一个响应。

23.6 ARP缓存与报文处理

尽管ARP用来实现地址绑定，但为每个数据报都发送一次ARP请求的做法却是非常低效的——发送一个数据报会导致3个帧在网络上传输（ARP请求、ARP响应、数据报自身）。更重要的是，大多数计算机通信都会涉及一系列的分组传送过程，发送方就可能要重复多次进行ARP报文的交换。

为了减少网络通信量，ARP软件从响应报文中提取并保存有关信息，以便能应用于后续的分组传输。ARP软件不会永久地保存这些绑定信息，它的实际做法是在内存中维护一个小的绑定表。ARP把这个表当做高速缓存（cache）来管理——即当一个响应来到时，就要替换表中的某个项。当表已满或某一项长期（例如20分钟）未被更新时，就删除表中最老的那个项。当ARP要执行地址绑定时，它首先在高速缓存中搜索，如果需要的绑定信息存在于缓存中，ARP就直接使用这个绑定信息而无须再发送一个请求。如果要求的绑定不存在于高速缓存中，ARP就广播一个请求并等待响应，然后更新高速缓存，并用所得绑定信息继续工作。

注意，与很多缓存方案不同，ARP在进行查找的时候（即引用一个表项的时候），高速缓存并不实行更新。相反，只有当一个ARP报文（请求或响应报文）经由网络到达的时候，高速缓存才会被更新。算法23-1列出了对一个输入的ARP报文的处理过程。

```
算法23-1
假设：
    一个输入的ARP报文（请求或响应）
执行：
    处理报文并更新ARP高速缓存
方法：
    提取发送方的IP地址I和MAC地址M
    if（地址I已经存在于高速缓存中）{
        用M替代高速缓存中的MAC地址
    }
    if（报文是一个请求报文且目标方是自身）{
        在ARP缓存中添加一个新的表项，因为表中没有提供发送方的绑定信息；
        生成并发送一个响应；
    }
```

算法23-1　ARP处理一个输入报文时所采取的步骤

正如上述算法规定，ARP在处理一个报文时，必须执行两个基本步骤：第一步，接收方从ARP报文中提取出发送方的地址绑定信息，并检查高速缓存中是否存在发送方的地址。若已有，则更新高速缓存中的信息。这种更新操作正对应于发送方硬件地址发生变化的情况。第二步，接收方检查ARP报文中的“操作”域以确认是一个请求报文还是一个响应报文。若是一个响应报文，接收方以前一定发送过一个请求，而且正在等待对方的绑定信息（即高速缓存中已经含有发送方的一个绑定表项，它是在发送请求操作时填进去的一个空表项）。若是一个请求报文，接收方就要比较“目标协议地址”域与本地协议地址，如果两者一致，则本计算机就是请求的目标，必须发出一个ARP响应。为了生成一个ARP响应报文，计算机利用接收到的报文，将其中的发送方绑定信息与目标绑定信息进行互换，然后在“发送方硬件地址”域中插入自己的硬件地址，并将“操作”域的值改为2，以示这是一个响应报文。

ARP还包括进一步的优化措施：当某台计算机遇到一个必须回答的请求时，它从请求报文中抽取出发送方地址绑定信息加入自己的高速缓存中，以便往后加以利用。为了理解这一

优化措施，我们需要知道两个事实：

- 大多数计算机通信总是涉及双向通信业务——如果一个报文从A传往B，则从B向A返回一个响应报文的概率非常高。
- 由于每一个地址绑定都需要内存，因此一台计算机不可能存储任意数量的地址绑定信息。

第一个事实解释了为什么取出发送方地址绑定信息会优化ARP的性能。仅当计算机A要向计算机B发送一个分组时，它才会发送针对B的ARP请求。因此，当B发现自己是A的请求目标时，一种可能的情况就是，当A发送的分组到达B后，B也会向A发送一个分组。ARP协议安排B从接收到的ARP请求中提取出A的绑定关系就省去了后面B向A发送ARP请求的必要。

第二个事实解释了为什么只有被ARP请求的目标计算机才向它的ARP高速缓存添加一个新的表项，而收到了该请求的其他计算机却不这样做。如果所有的计算机都添加这种信息，它们的高速缓存很快就会充满。而事实上，其中很多计算机从来都不会与网络上的另一些计算机通信。因此，ARP仅记录看起来很有必要的那些地址绑定信息。

23.7 概念地址边界

回顾第1章中所讲述的TCP/IP，它使用了一个五层参考模型。地址解析就是与网络接口层（即第2层）相关的一个具体功能。ARP在MAC地址和IP地址之间提供了一种重要的概念边界：ARP隐藏了物理编址的细节，允许高层软件直接使用IP地址。这样，则在网络接口层和所有更高层之间设置了一个非常重要的概念边界——应用程序及更高层软件都要利用协议地址来构建。图23-5说明了地址边界问题。

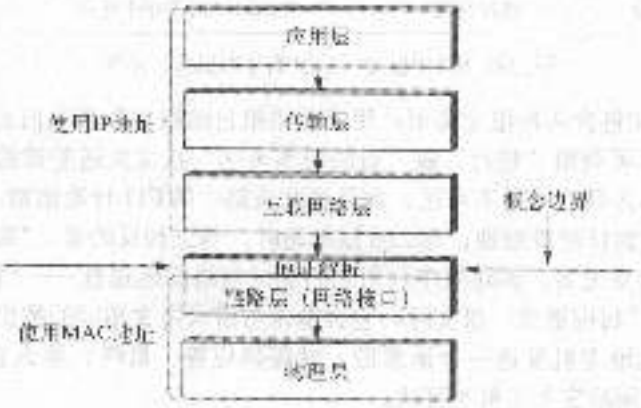


图23-5 使用IP地址与使用MAC地址之间的概念边界

要点 ARP在协议栈上形成了一个概念边界——ARP以上的层使用IP地址，ARP以下的层使用MAC地址。

23.8 因特网控制报文协议

我们说过，IP定义了一种尽力而为的通信服务，它的数据报可能会在传输过程中发生丢失、重复、延迟或乱序。这样看来，似乎尽力而为的服务并不需要任何差错检测机制。然而，要认识到尽力服务并不是意味着漠不关心——IP试图避免差错的发生，并在发生差错时报告这种消息。事实上，我们已经看到了IP中有关于差错检测的一个例子，即用于检测传输错误的头部校验和。当一台主机产生了一个IP数据报时，该主机会产生一个对整个头部进行计算

的校验和。无论何时收到一个数据报，校验和都用于验证头部是否无损地到达。类似地，IP头部还包含一个"生存时间"域，当路由器中的路由表因为某种原因产生环路时，"生存时间"域可以防止数据报在环路上不断循环传送。

对于校验和出错的反应非常简单：立即丢弃该数据报，不作进一步的处理。接收者无法相信数据报头部中的任何域，因为接收者不知道哪一位被改变了，甚至也无法给发送者发出任何差错报文，因为接收者也不能相信头部中的源地址。因此，接收者除了将被损坏的数据报丢弃外，别无选择。

IP利用一个辅助协议来处理后果的危害程度甚于校验和错误的其他问题。这个协议叫做因特网控制报文协议（Internet Control Message Protocol，ICMP），它主要用来向源发端（即最初发送数据报的计算机）报告错误。有意思的是，IP和ICMP是相互依赖的：IP要依赖ICMP来报告错误，而ICMP又要利用IP来传输差错报文。

尽管目前已经定义了20多种ICMP报文，但是只有少量在使用。表23-6列出了关键的ICMP报文以及它们的用途。

编　号	类　型	用　途
0	回应应答	在ping程序中使用
3	目的地不可达	数据报无法传递
5	重定向	主机要求改变路由
8	回应请求	在ping程序中使用
11	超时	TTL域中或是分片重装计时器超时
12	参数问题	IP头部不正确
30	路径跟踪	在tracemute程序中使用

图23-6　ICMP报文（给出编号和用途）示例

如图所示，ICMP包含两种报文类型：用于差错报告的报文和获取信息的报文。例如，数据报不能成功投递时可利用"超时"或"目的地不可达"报文发送差错报告。如果去往目的地的路由不存在，那么目的地址不可达；如果报文头部中的TTL计数值减为0或属于同一报文的所有分片到达前重装计时器到期，那么数据报超时。与之相反的是，"回应请求"与"回应应答"报文与错误信息无关，ping程序利用它们测试网络的连通性——当主机或路由器上的ICMP软件收到一个"回应请求"报文时，它会携带与请求报文相同的数据发回"回应应答"。因此，ping程序向远地主机发送一个请求后，就等待应答。最终，要么它从远地主机可达，要么在超过适当的时间后宣布主机不可达。

23.9　ICMP报文格式与封装

ICMP利用IP来传输每一个差错报文。当路由器有一个ICMP报文需要传输时，它就产生一个IP数据报并将ICMP报文封装在其中，也就是说，ICMP报文是放置在IP数据报的数据区里被传输出去的。然后，这个数据报像通常那样进行转发，并封装成帧进行传输。图23-7说明了封装的两个层次。

携带ICMP报文的数据报并没有特别优先权，它们像其他数据报一样被转发，但有一个很小的例外：如果携带ICMP差错报文的数据报又出了错，就不再发送差错报文了。理由很简单，设计者是想避免由于传输过多的差错报文而造成互联网拥塞。

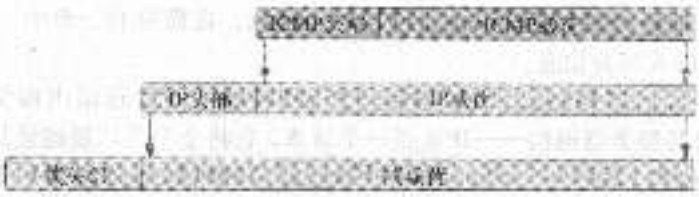


图23-7 发送ICMP报文时进行到某处的两个层次

概括如下：

因特网控制报文协议包括差错报文和信息报文两种。ICMP将报文封装在IP数据报中传输，而IP则利用ICMP来报告数据传输中出现的问题。

23.10 协议软件、参数与配置

我们在讨论因特网协议的时候已经介绍了它们的操作过程，即主机或路由器加电后，操作系统开始启动，然后协议软件进行初始化。问题是：主机和路由器中的协议软件是如何开始运作的呢？对于路由器，这个问题的答案比较平常：管理员必须为路由器指定一些初始配置值，比如每个网络连接的IP地址、运行的协议软件、路由表的初始值。这些配置值保存在磁盘上，路由器在启动过程中加载这些值。

主机配置则相对比较复杂，通常需要两个处理步骤，即所谓的自举（bootstrapping）[⊖]过程。第一步发生在计算机启动的时候，操作系统载入一些基本的配置参数，使协议软件能在本地网络上进行通信。第二步，协议软件填入额外的信息，比如计算机的IP地址、地址掩码和本地DNS服务器的地址。本质上，协议软件由参数化二进制镜像文件组成，初始化时填入一些参数即可工作。因此，同样的二进制镜像文件可以运行在很多计算机上，即使计算机的网络连接发生变化，镜像也无须改变。我们称协议软件针对具体情况是可配置的（configured），概括如下：

为了使协议软件可以在各种网络环境下的多台计算机上运行被设好的二进制镜像文件而无须更改，对它进行了参数化设计。当要在给定的计算机上启动该软件的版本时，必须将有关这台计算机和所连接网络的信息，通过一组参数设置对该软件进行配置。

23.11 动态主机配置协议

目前，协议设计者创建了很多种方法来让主机获得协议参数。一种早期的叫做反向地址解析协议（Reverse Address Resolution Protocol，RARP）的机制，允许一台计算机从服务器获得IP地址。前面讲述的ICMP协议包含地址掩码请求（Address Mask Request）和路由器发现（Router Discovery）报文，借助这两种报文，主机可以获得指定网络上使用的地址掩码和路由器地址。这几种早期的机制在使用上相互独立，以广播方式发送请求报文，典型的做法是主机从低到高配置协议层。

在因特网协议的演进过程中，设计者发明了一种功能独立的协议，它允许主机通过一个请求获得多个协议参数，这就是自举协议（Bootstrap Protocol，BOOTP）。自举协议可以为计

⊖ 这个术语来源于词组"pulling oneself up by one's bootstraps"，意即"让你自己的靴带把自己拉起来"。

算机提供IP地址、地址掩码和默认路由器地址[7]。因此，在简单的一步中，主机就能获得配置IP协议栈所需要的大部分信息。

与其他获取配置信息的协议一样，BOOTP采用广播方式发送请求报文，但不同的是，BOOTP使用IP协议与服务器通信——IP发送一个请求，它将全"1"广播地址用作"目的地址"，而将全"0"地址用作"源地址"。BOOTP利用接收帧中的MAC地址通过单播（unicast）方式发送响应。因此，一台不知道其IP地址的主机也能够与BOOTP服务器通信。

BOOTP协议的最初版本采用固定地址分配方案。其中，服务器包含一个数据库，该数据库保存着分配给每台主机的IP地址。主机发送的请求中包含一个唯一的ID（通常使用主机的MAC地址），服务器利用这个ID来找到主机对应的IP地址。它的问题在于：BOOTP需要管理员手工参与管理——某台计算机在使用BOOTP获得地址前，网络管理员必须在BOOTP服务器上配置该计算机对应的IP地址。

虽然当计算机的设置维持固定的时候这样做还是不错的，但是如果计算机的设置经常改变的话，靠人工来做这样的准备工作就很麻烦了。例如，考虑在一个餐馆里的Wi-Fi接入点，它要提供对任何客户的访问能力。为了解决这种情况，IETF扩展了BOOTP的功能，将它的名字改为动态主机配置协议（Dynamic Host Configuration Protocol，DHCP）。

DHCP提供了一种新的机制，它允许任意一台计算机加入到新的网络中并自动获得IP地址。这个概念术语称为即插即用网络（plug-and-play networking）。要点概括如下：

> DHCP允许计算机移动到新的网络并获得配置信息，不需要管理员人工去修改数据库条目。

事实上，DHCP也继承了与BOOTP同样的实现方法，即当计算机启动时，就广播一个DHCP请求，服务器则发送一个DHCP应答[8]。管理员可以将DHCP服务器配置成能提供两种地址配置方式的形式：与BOOTP类似的固定地址分配方式和根据需要从动态地址池中获取地址的方式。一般来说，固定地址分配给服务器，而动态地址分配给其他主机。事实上，按需分配的地址不是无限期可用的，而是由DHCP产生一个时间有限的地址租用期来决定[9]。通过设定租用期，DHCP服务器在必要时可以收回地址。当租期届满时，服务器就要收回地址放回到地址池中，以便分配给其他计算机使用。这种租借方式是服务器连续工作的基础，因为这可以使服务器控制资源和收回地址。这样，即使持有地址的那台主机崩溃了，也能将它的地址收回再用。

当租期届满的时候，主机可以选择是释放地址还是与DHCP服务器重新协商延长租期。这种协商过程与计算机的其他活动并行发生。一般情况下，DHCP允许每个租期都延长，计算机可以继续工作无须中断运行应用程序或网络通信。然而，出于管理或技术缘由，服务器也可以配置成拒绝延长租期。例如，考虑某大学教室里网络的情况，服务器可以配置成在一堂课结束时租期届满（以允许下一堂再次使用同一地址）。DHCP授予服务器在租期上绝对的控制权——如果服务器拒绝一个延长请求，主机必须停止使用该地址。

23.12　DHCP协议操作与优化

虽然DHCP协议的工作原理很简单，但它还是包括了一些具体的措施来优化性能，最有意

---

[7] 即默认网关，一译者注。

[8] DHCP使用术语提供（offer）来表示服务器发送的报文，可以说是服务器向客户提供一个地址。

[9] 管理员在建立地址池时为每个地址指定租期。

文的3个措施是：

- 分组出现丢失或重复时的恢复过程。
- 对服务器地址进行高速缓存。
- 避免因同时出现大量请求而发生阻塞。

第一条意味着DHCP在设计过程中要确保分组出现丢失或重复时不会产生错误的配置。即主机如果没有收到响应，则必须重新发送它的请求报文；如果收到重复的响应，则忽略这个多余的副本。第二条意味着主机一旦使用DHCP发现（DHCP Discover）报文找到了一个DHCP服务器，就将服务器的地址存入缓存中以备后用，这提高了租期续约过程的效率。

第三条意味着DHCP要采取一些措施来防止大量主机同时发送请求。例如，断电后恢复来电时，网络中所有的计算机几乎同时启动，很可能就会出现所有计算机同时发送请求的现象。为了防止网络中的大量主机同时发送请求涌入DHCP服务器而导致阻塞，DHCP要求每个主机在发送（或重发）请求之前要等待一个随机时间。

23.13 DHCP报文格式

由于DHCP是当作对BOOTP的扩充而设计的，因此它的报文格式是由BOOTP格式稍加修改而成的。图23-8说明了DHCP的报文格式。

除了“选项”域，DHCP报文中的其他域都有固定的大小。前7个域包含处理报文的信息。OP域指明该报文是请求还是响应。为了区分不同的报文，例如是客户用来发现服务器还是请求地址的报文，或者是服务器用来确认或拒绝请求的报文，DHCP包含有一个“选项”用于指定报文类型（message type），也就是说，OP域告知了报文是从客户端发往服务器端的还是从服务器端发往客户端的，而其中一个“选项”给出了准确的报文类型。

HTYPE和HLEN域指明网络硬件类型和硬件地址的长度。客户使用“标志”域来指明它能接收广播式应答还是直接应答。HOPS域指明有多少服务器转发了请求。“事务标识符”域提供给客户一个值，用来确定到来的响应是否与其请求相匹配。“逝去时间”域指明从主机[1]开始启动至当前已过去了多少秒。最后，如果主机知道它的IP地址（例如，通过DHCP以外的其他方式获取的地址），就将它填入请求中的“客户IP地址”域。



图23-8 DHCP报文格式

[1] 这里及后面的“主机”都是指发送DHCP请求的客户机。——译者注

报文中后面的一些域则用于携带响应信息给发送请求的主机。如果主机不知道它的地址，服务器用“你的IP地址”域来提供该值。此外，服务器使用“服务器IP地址”域和“服务器主机名”向该主机提供有关服务器的位置信息。“路由器IP地址”域含有默认路由器的IP地址。

除了协议配置功能外，DHCP还允许计算机协商寻找自举镜像文件。为了做到这一点，主机可以在“自举文件名”域中填进一个请求（例如主机可以请求Linux操作系统）。DHCP服务器不发送镜像文件，只是确定哪一个文件包含被请求的镜像文件，并用“自举文件名”域送回该文件名。一旦DHCP响应到达，主机就必须使用另外的协议（如TFTP协议）去下载该镜像文件。

23.14　通过中继间接访问DHCP服务器

虽然DHCP通过在本地网络上发送广播报文来寻找服务器，但协议并不要求每个网络里都存在一个服务器，也可以使用中继代理（relay agent）来转发客户与服务器之间的请求和响应。在每个网络中必须至少有一个中继代理，而且必须使用适当的DHCP服务器地址对代理进行配置。当服务器发出响应时，中继代理将响应转发给客户。

似乎采用多个中继代理的做法并不比使用多个服务器的做法好多少。然而，网络管理员却喜欢管理多个中继代理。这有两个原因：第一，在有一个DHCP服务器和多个中继代理的网络里，地址管理被集中在单个设备，这样管理员就不必与多个设备交互以改变租借策略或确定当前状态。第二，很多商用路由器都包含有一种机制能对它所连接的所有网络提供DHCP中继服务，而且路由器中的中继代理设施都很容易配置（包括打开转发开关和指定DHCP服务器地址），且不一定要经常改变配置。

23.15　网络地址转换

随着因特网的发展，地址逐渐变得不能满足需求。于是人们引入了子网和无类编址（CIDR）方案[○]以帮助节省地址的使用。此外，人们还发明了另外一种叫做网络地址转换（Network Address Translation，NAT）的技术，使同一站点的多台计算机能共享一个全球有效的IP地址。在NAT技术中，站点内的主机看似有一个正常的因特网连接，而因特网上的主机看似总是从单台计算机接收通信信息，而不是从站点内的某一台计算机那里接收信息。也就是说，站点内的多台主机都在运行常规的TCP/IP协议和应用，它们就和普通情况一样在因特网上进行通信。

NAT要按事核（in-line）配置来工作，就是说，运行NAT的设备要放置在因特网与站点之间的连接上。虽然NAT在概念上与其他设施和服务是分开的，但大多数实现都是将NAT内嵌在另一种设备中，比如在Wi-Fi无线接入点或者在因特网路由器中。图23-9表示出一个站点使用NAT的典型配置结构。

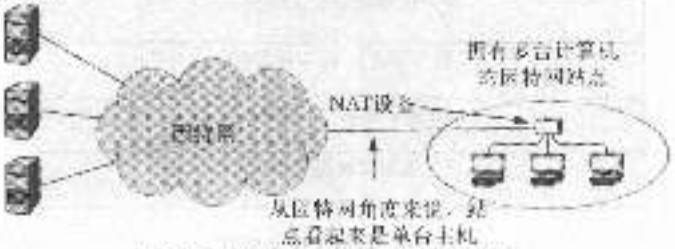


图23-9　使用NAT的概念架构示意图

○　关于子网与无类编址方案的介绍可在第21章中找到。

23.16 NAT操作与私有地址

NAT的目的是要提供一种虚拟的寻址机制。从因特网的角度看，站点看似只由分配了有效IP地址的单台主机构成——从站点所有计算机发送出来的数据报好像都是发源于一台主机，而发送给站点内所有计算机的数据报好像只是发送给一台主机。从站点内的某一台计算机的角度看，因特网似乎能接收和路由私有地址[⊖]。

当然，一个IP地址不能分配给多台计算机，两台或多台计算机使用同一个地址会引起地址冲突，因为这些计算机都会对同一ARP请求进行响应。因此，为了保证准确性，指定网络上的每台计算机都必须分配一个唯一的IP地址。NAT利用两种类型的地址来解决这个问题。NAT设备自身分配一个全球有效的IP地址，这样NAT设备看起来好像是因特网上的一台主机。而站点内的每台计算机却被分配一个唯一的私有地址（private address），即不可路由地址（nonroutable address）。图23-10列出了IETF指定作为私有地址的地址块。

地址块	描　述
10.0.0.0/8	A类私有地址块
169.254.0.0/16	B类私有地址块
172.16.0.0/12	16个连续的B类地址块
192.168.0.0/16	256个连续的C类地址块

图23-10 NAT使用的私有（不可路由）地址块

举例来说，假设某个NAT设备使用192.168.0.0地址块给站点内的主机分配私有地址。为了确保站点内的每个地址是唯一的（即防止地址冲突），主机分配得到的地址可以是192.168.0.1，192.168.0.2，等等。

可惜的是，私有地址在全球因特网上是无效的，因特网上的路由器已经被配置为拒绝路由含有不可路由地址的数据报。因此，私有地址只能在站点内部使用——数据报在从站点内部进入因特网之前，NAT设备必须将私有IP地址转换成全球有效IP地址。类似地，NAT设备在将进入的数据报转发到站点内的某台主机前，必须将分组中的全球有效IP地址转换成私有IP地址。

NAT的最基本操作，就是当数据报从站点进入因特网时进行源地址转换，而当数据报从因特网进入站点时进行目的地址转换。例如，假设NAT设备已经分配了一个全球有效IP地址128.210.24.6，如果一个地址为私有地址192.168.0.1的主机向因特网上另一个地址为198.133.219.25的主机发送数据报和接收返回的应答，就会发生这种转换。图23-11说明了每个方向上将要发生的转换过程。



图23-11 NAT基本的地址转换操作示意图，它既对送出数据报的源地址进行转换，也要对进入数据报的目的地址进行转换

概括如下：

NAT的最基本操作就是，当数据报从站点进入因特网时替换其中的源IP地址，而当数据报从因特网进入站点时替换其中的目的IP地址。

⊖ 因为站点内的所有计算机都使用私有地址。——译者注

NAT通过使用一张地址转换表（translation table）存储重写地址时所需的信息来实现地址转换。例如，图23-12给出了对应于图23-11中地址映射关系的转换表。

方向	域	旧　值	新　值
出去	源IP地址	192.168.0.1	128.210.21.6
	目的IP地址	198.133.219.25	不变
进入	源IP地址	198.133.219.25	不变
	目的IP地址	128.210.21.6	192.168.0.1

表23-12　NAT地址转换表示例（图23-11中的地址映射关系）

那些要改变的值是如何放进转换表的呢？虽然这些值可以由系统管理员手工配置，但大多数NAT系统是自动完成的。每当站点内的计算机向因特网发送分组时，NAT就在转换表中放置一个记录项。例如，当计算机192.168.0.1第一次给目的地址198.133.219.25发送数据报时，NAT就在表中增加一个记录项。此后，当NAT从198.133.219.25收到一个应答报文时，它就找出表中相应的项，将报文目的地址转换成192.168.0.1。

23.17　传输层NAT

前面所述的NAT基本版本只能处理站点内的每台主机与因特网上唯一的服务器进行通信的情况。如果站点内有两台主机都试图与远程服务器X进行通信，那么转换表中就会含有匹配X的多个记录项，这时NAT将无法路由进入的数据报。另外，当站点内某台主机上运行的两个或多个应用进程都试图同时与因特网上不同的目的地进行通信时，基本的NAT转换方法也行不通。

当前最广泛使用的NAT改进版处理了两个问题：它允许站点内任一台主机上运行任意多的应用进程；所有的应用进程都可以与因特网上的任一个目的地进行通信。虽然这种改进的NAT版本在技术上称为网络地址与端口转换（Network Address and Port Translation, NAPT）技术，但是由于它太流行了，以至于很多的网络专业人员直接将术语NAT等同于术语NAPT。

理解NAPT工作原理的关键点在于，明白应用进程利用了协议端口号（protocol port number）来区分同一主机上的不同服务（第25章和第26章将介绍使用端口号的UDP和TCP传输层协议），除了维护含有源地址和目的地址的转换表外，NAPT还利用端口号将每个数据报与TCP或UDP流进行关联。也就是说，NAT仅涉及IP层，而NAPT却是在传输层头部的基础上进行操作。其结果就是，NAPT使用的地址转换表中的记录项包括4个部分，即源IP地址、目的IP地址、源端口号和目的端口号。

例如，如果计算机192.168.0.1和计算机192.168.0.2上的浏览器都使用本地端口号30000，而且它们都通过地址为128.10.24.6的NAPT设备与在80号端口监听的Web服务器建立了TCP连接，那么这时的转换表可能会发生什么情况呢？为了避免冲突，NAPT必须为这两个连接另外选择不同的TCP源端口。图23-13所示为一种可能的情况。

方向	域	旧　值	新　值
出去	IP SRC:TCP SRC	192.168.0.1:30000	128.10.24.6:40001
出去	IP SRC:TCP SRC	192.168.0.2:30000	128.10.24.6:40002
进入	IP DEST:TCP DEST	128.10.19.20:40001	192.168.0.1:30000
进入	IP DEST:TCP DEST	128.10.19.20:40002	192.168.0.2:30000

表23-13　NAPT转换表的例子，包含两台同一Web服务器的两个TCP连接

在图中，两台本地上机上的应用进程都使用本地端口号30000。由于操作系统循环使用端口号，不同主机上的应用进程使用相同的端号口是不太可能的，然而，NAPT在处理这种极端情况时也不会发生混淆。本例中，NAPT为一个连接选择了40001端口，为另一个连接选择了40002端口。

23.18 NAT与服务器

我们说过，NAT系统通过查看发送出去的通信业务，每当站点内的一个应用进程发起对外的通信过程时就建立一个新的映射关系，从而自动地创建一个转换表。可惜的是，从因特网上发起到站点内部的通信过程中，自动地建转换表的这种组成结构却无法正常工作。例如，如果站点内的多台计算机运行Web服务器，NAT就无法知道哪台计算机应该接收进入的连接请求。有一种叫做双NAT（Twice NAT）的变种技术，它允许站点内运行多个服务器。这种技术要求NAT系统与站点的域名系统服务器进行交互才能正常工作。当因特网上的应用进程要查找站点内某台计算机的域名时，站点的域名服务器就返回已分配给NAT设备的有效IP地址，并在NAT转换表中创建一个新的记录项[⊖]。这样，在第一个分组到达之前，转换表就被初始化了。虽然这种双NAT不是十分优良，但在大多数情况下还算可行。然而，如果客户应用进程直接使用IP地址而不进行域名查找，或是客户使用DNS代理去解析域名，那么这种双NAT技术就不起作用了。

23.19 家用NAT软件和系统

在安装有宽带网络的居民区或小型商业区，NAT特别有用，因为它允许一组计算机共享连接而不要求用户从ISP购买额外的IP地址。从市场上除了能买到允许一台PC为其他PC充当NAT设备的软件外，也可以买到廉价的专用NAT硬件系统。这样的系统通常叫做无线路由器（wireless router）[⊖]，例如，Linksys公司出售一种能提供NAT功能的专用硬件系统，其中包括4个以太网接口和Wi-Fi无线接入功能。图23-14所示为这种路由器的连接方式。

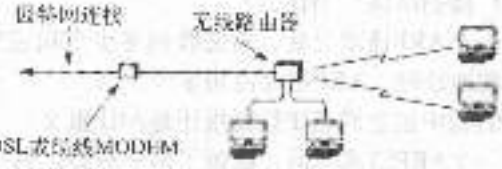


图23-14 无线路由器连接示意图

23.20 本章小结

IP利用地址解析协议（ARP）将下一站IP地址绑定到一个等效的MAC地址。ARP定义了计算机在解析地址时交换的报文格式、数据封装方法和处理规则。由于各种网络的硬件编址方案各不相同，因此ARP只规定了报文格式的通用模式，但允许根据具体的硬件编址方案来确定其中的细节。ARP规定计算机发送请求报文时应该采用广播方式，发送响应报文时则采用直接应答方式。此外，ARP还利用高速缓存技术来避免为每个分组都发送一个请求。

⊖ 新的记录项将该域名对应的计算机内私有IP地址与NAT设备的IP地址关联起来。——译者注

⊖ 无线路由器这个术语容易产生误导，因为这种路由器也能（而且是主要地）为主机提供有线连接。

网际协议包含一种辅助的差错报告机制，即因特网控制报文协议（ICMP）。到达路由器的数据报头部域中如果有不正确的值或是数据报无法投递，路由器会利用ICMP发送差错报告。ICMP报文总是发回给数据报的源发端，从不发送给中间路由器。除了差错报告报文，ICMP还包括诸如在ping程序中使用的“回应请求”和“回应应答”之类的信息报文。每一种类型的ICMP报文都有其独特的格式。头部中的“类型”域使得接收方能将接收到的报文划分为正确的域。ICMP报文被封装在IP数据报中传输。

起初，在系统启动时要用单独的协议来分别获取每个协议配置参数。动态主机配置协议（DHCP）扩充了自举协议（BOOTP），使主机通过单个请求就能获得所有必须的信息。DHCP响应能提供IP地址、默认路由器的地址以及域名服务器的地址。当DHCP向主机自动分配一个IP地址的时候，DHCP服务器即向主机提供了一个租用期，在此期间该地址都可以使用。租期届满时，主机必须协商延长租期，否则就要停止使用该地址。

NAT机制允许一个站点内的多台计算机通过单一的IP地址去访问因特网。NAT设备要对去往因特网和进入站点的每个数据报头部域进行重写。对于客户应用，当NAT设备发现第一个通信分组时，就会自动地建立NAT转换表。目前已有几种NAT的变种技术，但是最流行的是NAPT。它在传输层头部的基础上进行操作，能转换协议端口号和IP地址。NAPT允许站点内多台计算机上的多个应用进程同时与因特网上的任意目的地进行通信。

进一步的阅读资料

有关NAPT的详细内容，请参考RFC2663和RFC2766。

练习题

23.1　路由器使用转发表查找下一站地址时，得到的结果是一个IP地址。路由器在转发数据报前，还必须做哪些事情？

23.2　我们用什么术语来描述协议地址与硬件地址之间的映射过程？

23.3　ARP能否用于不支持广播的网络？为什么？

23.4　当某台计算机广播了一个ARP请求之后，希望收到多少个响应？为什么？

23.5　将IP地址解析为以太网地址时，ARP报文占用多少字节？

23.6　计算机如何知道到达的帧中包含的是IP数据报还是ARP报文？

23.7　假设一台计算机发出一个ARP请求之后，收到了两个应答。第一个应答声明它的MAC地址是M_1，而第二个应答声明它的MAC地址是M_2。ARP软件该如何处理这样的应答呢？

23.8　ARP只允许在同一网络内进行地址解析。利用IP数据报向远地服务器发送一个ARP请求是否有意义？为什么？

23.9　算法23-1什么时候会在ARP高速缓存中创建一个新的记录项？

23.10　ARP协议以下的层使用那种类型的地址？

23.11　如果数据报头部域中的某个值不正确，发送端将接收到那种ICMP差错报文？

23.12　如果出现路径环路，路由器将会发送那种ICMP差错报文？试解释这个过程。

23.13　假定用户在执行ping程序时指定一个直接广播地址作为目的地址，会出现什么样的结果？并解释。

23.14　有些版本的traceroute程序发送ICMP报文，而另一些版本的这种程序则发送UDP报文。用你计算机上的traceroute程序版本做一个实验，确定它发送的是那种报文。

23.15　给定一个以太网帧，接收主机需要检查其中的那些域才能确定它是否包含有一个ICMP

报文?

23.16 请列出计算机启动时可能配置的关键网络信息。

23.17 BOOTP与DHCP的主要不同点是什么?

23.18 有些网络应用程序把配置推迟到有服务需要的时候才进行。例如，一台计算机可以一直等到用户试图打印文档时才启用软件去查找可用的打印机。请问，推迟配置的主要优缺点是什么?

23.19 DHCP允许一台服务器设置在远地网络上。那么计算机怎样才能向另一网络的服务器发送DHCP报文呢?

23.20 设计一个分布式算法作为DHCP的一个变种技术，用它来实现一种投标（bidding）方案。假设在每一台计算机上都运行该算法的一个副本，利用这个算法给每一台主机分配一个唯一的地址。

23.21 NAT的主要用途是什么?

23.22 有很多NAT设备选择图23-10所示的10.0.0.0/8地址块，因为它提供了最大的通用性，试解释为什么。

23.23 在图23-11中，ISP向站点分配了一个IP地址。请问哪一个是它分配的地址?

23.24 在图23-13对应的示例中，如果第三个应用进程也试图访问同一个Web服务器，请扩充这个表以反映出它的映射关系。

23.25 假设在一个站点内有3台计算机分别与因特网上的3个不同的Web服务器有TCP连接，请你创建一个NAPT转换表。

23.26 NAPT用到的什么关键信息是在大多数IP数据片中找不到的?

23.27 为了优化重装过程，有些版本的Linux操作系统首先发送IP数据报的最后一片，然后才按顺序发送其他片。请解释为什么这种做法对NAT却是行不通的?

23.28 当你使用无线路由器的时候，可能分配给主机的IP地址是什么?

第24章 未来的IP：IPv6

24.1 引言

前几章讨论了网际协议的当前版本——IPv4，其中讲到IP数据报是由一个头部和其后的数据组成。头部包含了各种信息，例如IP软件在传递数据报时要用到的目的地址；头部中的每个域都是固定大小，以使处理起来更为有效。第22章还讲述了如何将IP数据报封装到网络帧中，然后在物理网络中传输。

本章集中讨论网际协议的未来。我们先对当前IP版本的健壮性和局限性进行评估，然后再考虑一个由IETF开发的IP新版本，本章将介绍新版本的特征，并说明设计者是如何克服当前版本中的一些局限性的。

24.2 IP成功之处

当前版本的IP被认为成功，IP已经使因特网能够对付异构网络、硬件技术的不断变化以及网络规模的急剧增长，网际协议提供了一种网络抽象，使应用程序在无须了解因特网的体系结构或是底层网络硬件的情况下彼此之间就能通信。为了适应网络的异构性，IP定义了与底层网络无关的编址方案、数据报格式、封装方法和分片策略。

从IP技术的应用和全球因特网的规模，充分体现出IP的通用性和可伸缩性。更为重要的是，IP能适应硬件技术的不断变化。虽然这一协议在局域网技术流行之前就已制定下，但原先的这些设计在几代硬件技术的演进中仍能很好地工作。除了能实现更高的数据传输速率外，IP也能适应帧长度的不断增加。

概括如下：

> 当前版本的IP取得的成功是令人难以置信的——该协议已经能适应硬件技术、异构网络以及网络规模等方面的变化。

24.3 改革的动机

既然IP工作得很成功，为什么还要改革呢？当初设计者在定义IP的时候，只有少量计算机网络存在，所以他们才决定使用32位的IP地址，因为即使这样也能允许因特网包含百万以上的网络。然而，全球因特网正在以指数规律增长，不到一年其规模就翻了一倍，以当前的增长率，每种可能的网络前缀最终都会被分配完，网络未来的增长就不太可能了，因此，设计新版本IP的主要动机就是由于地址空间限制——持续增长的因特网需要更大的地址空间来适应。

IP改革的第二个动机产生于人们对某些应用需要特殊设施的认识。结果，经各个技术研究组商讨后认为：如果IP要被取代，则取代它的新版本应该具有更多的特性，例如，不妨考虑一下发送实时音频和视频的应用，它们要求数据能以等时间隔传递，且要求保证小的抖动。可惜的是，路由的变化经常导致端对端时延的变化，这意味着抖动也在增加。虽然当前的IP

数据报头部已经包含了用于请求某类服务的域，但协议并没有定义实时服务。因此，人们商讨后认为新的IP版本应该提供新的机制，允许携带实时流量的数据报以避免因路由改变而产生的抖动问题。

另一个技术研究组则认为新版本IP应该具有更复杂的寻址和路由能力。特别地，它应该能配置IP寻址和路由以处理那些重复性服务。例如，Google在世界各地维护着很多数据中心，该研究组认为，用户在浏览器中输入google.com后，IP若能将数据报送到最近的Google数据中心，那么用户和系统都将十分受益。此外，当前很多应用允许用户间的协同工作（collaborate）。为了使协同更高效，因特网需要有一种机制允许创建和改变这种协同组，并且能提供一种方法以便将每个分组的副本传送给指定组中的每位成员。

24.4 沙漏模型与改革的难点

虽然促使人们在1993年就开始着手进行IP新版本工作的重要原因，是认为由于明显缺乏剩余的可分配地址，但也还没有出现因为地址不足而产生严重后果的事件，因此IP也就一直在沿用着而没有被改变。为了理解其中的原因，我们不妨思考一下IP的重要性和改革产生的费用。从IP的重要性来说，它处于互联网通信的中心位置——所有的应用都要使用IP，而IP又运行在所有底层网络技术之上。网络专家认为因特网通信遵循沙漏模型（hourglass model），IP处于沙漏最细的位置。图24-1表示了这一概念。

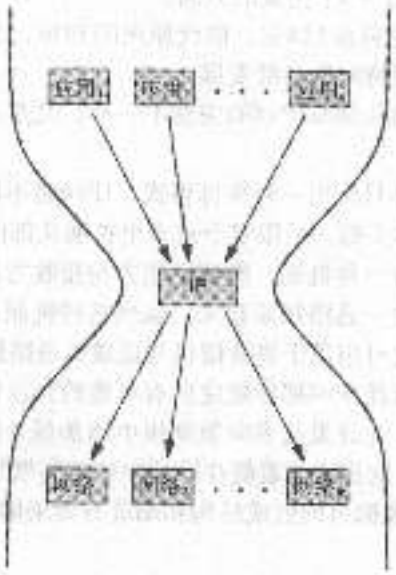


图24-1 因特网通信的沙漏模型，IP处于沙漏中心位置

由于对IP的依赖性以及IP带来的后续惰性，因而产生了一个重要的观点：

由于IP处于所有因特网通信的中心位置，因此要改变IP就要求改变整个因特网。

24.5 名称和版本号

当研究人员着手进行新版本IP的工作时，需要为这项工作起一个名称。借用当时流行的一个电视节目中的说法，他们选用了下一代IP（IP-The Next Generation）这一名称，所以很

多早期的报告中都将新的IP协议称为IPng。可惜的是，当时还有很多关于IPng方面的争论性的提议，这样就使名称问题变得含糊起来了。

在定义一个特定协议的时候，设计者必须将此协议与所有其他的提议区分开来。他们决定在最终被标准化的协议头上加入一个正式的版本号，被选择的那个版本号令人感到奇怪，因为当前的IP版本号是4，网络界期待下一个正式的IP版本号是5。然而，版本号5早已被分配给一个叫ST的实验性协议。因此，IP的新版本就将"6"作为它的正式版本号，这一协议也就被称为IPv6。为了与IPv6区分开来，当前的IP版本就被称为IPv4。

24.6　IPv6的特性

IPv6保留了IPv4的很多非常成功的设计特性。像IPv4那样，IPv6也是无连接的——每个数据报都含有目的地址，并且可以独立地确定传输路径。也像IPv4那样，IPv6的数据报头部含有最大跳数，即数据报被丢弃之前允许经过的最大跳程数。另外，IPv6还保留了IPv4可选项中提供的大多数适用性设施。

尽管IPv6保留了当前版本的基本概念，但仍修改了所有的细节。例如，IPv6使用更大的地址空间和一个全新的数据报头部格式。而且，IPv6将头部信息划分为一系列定长的头部。因此，它不像IPv4那样把关键信息都放置在头部的固定域而只将次要信息添加在可变长的可选项中，IPv6的头部则总是可变长的。

IPv6的新特性可以归纳为以下5个主要的方面：

- 地址空间。每个IPv6地址包含128位，取代原来的32位，从而所形成的地址空间大得足以适应好几十年全世界因特网的持续发展。
- 头部格式。IPv6的数据报头部与IPv4的完全不一样，几乎每个域都做了改变，或被替换掉了。
- 扩展头部。不像IPv4那样只使用一种头部格式，IPv6将不同的信息编码到各个不同的头部中。IPv6数据报由基本头部、后跟零个或多个扩展头部以及数据等所构成。
- 支持实时业务。IPv6含有一种机制，能使发送方与接收方通过底层网络建立一条高质量的通路，并将数据报与这一通路联系起来。虽然这种机制是提供给要求较高性能保证的音频和视频应用的，但也可应用于将数据报与低成本通路联系起来。
- 可扩充的协议。IPv6没有像IPv4那样规定所有可能的协议特征，取而代之的是，设计者们提供了一种新的方案，允许发送者向数据报中添加额外的附加信息。这种扩充方案使得IPv6比IPv4更加灵活，这也意味着能在设计中按需要增加新的特性。

下面几节将通过介绍IPv6数据报的组成结构和编址方案来解释以上新特性的实现原理。

24.7　IPv6数据报格式

一个IPv6数据报包含一连串的头部。如图24-2所示，IPv6数据报由一个基本头部（base header）开始，后跟零个或多个扩展头部（extension header），再后面跟着载荷数据。

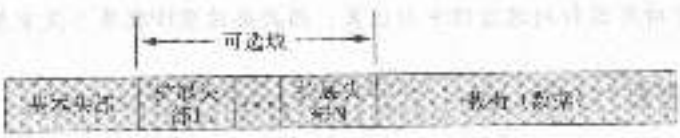


图24-2　IPv6数据报的一般格式

虽然这个图只是表示出数据报的一般格式，但图中的各个域并没有按比例来绘制。实际

上，有一些扩展头部可能比基本头部要长，而其他扩展头部可能较小。在很多数据报中，载荷长度比头部长度大得多。

24.8 IPv6基本头部的格式

虽然IPv6的基本头部是IPv4头部的两倍，但包含的信息却比IPv4的少，图24-3所示为它的格式。

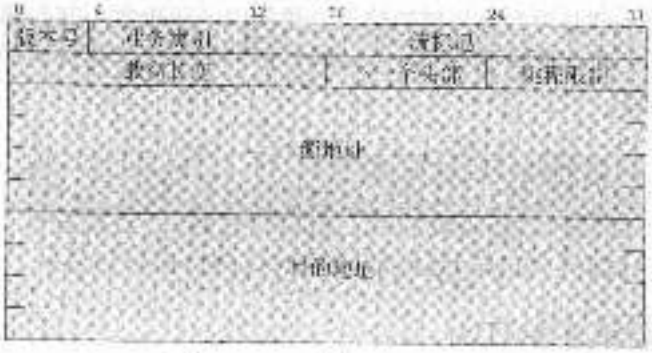


图24-3 IPv6基本头部的格式

如图所示，头部里的大多数空间都用于标识“源地址”和“目的地址”两个域，每个地址占用16个字节，是IPv4地址长度的4倍。像IPv4那样，“源地址”域标识发送方，“目的地址”域标识接收方。

除了源地址和目的地址，基本头部还包含6个域。“版本号”域指明协议是第6版本。“业务类别”域指明业务的类别，它使用一个称为区分服务（differentiated services）的业务类型定义来指定数据报所需的一般特征。例如，为了发送交互业务（如击键和鼠标移动信息），人们可能会规定具有低延时特征的一类业务；而为了在因特网上发送实时音频信息，发送者会请求一条低抖动的通路。“载荷长度”域对应于IPv4中的“数据报长度”域，但与IPv4不同的是，载荷长度规定的只是数据报所携带的数据（即载荷）的长度[⊖]，而基本头部的长度除外。“跳程限制”域对应于IPv4中的“生存时间”域。IPv6对跳程限制做了非常严格的解释——如果在数据报到达其目的地之前跳程限制计数器被减小到零，则该数据报将被丢弃。“流标记”域最初的目的是将数据报与一个特定的底层网络路径联系起来。IPv6被定义之后，端到端流标记的使用就不太受重视了，因此“流标记”域也就不那么重要了。

“下一个头部”域用于指定跟在当前头部后面的信息的类型。例如，如果数据报含有一个扩展头部，则“下一个头部”域就指明该扩展头部的类型；如果没有其他扩展头部，则该域就指明载荷中所携带的数据的类型。图24-4说明了这两种情况。

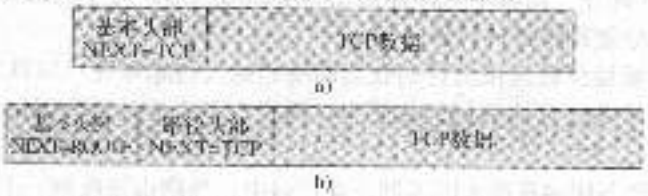


图24-4 a）只有基本头部和TCP载荷的情况；b）包含基本头部、路径头部和TCP载荷的情况

⊖ IPv6中所有的扩展头部和数据合起来叫做数据报的有效载荷，因此载荷长度等于整个数据报长度减去40个字节基本头部的长度。——译者注

24.9　隐式和显式头部长度

因为IPv6标准为每种可能的头部类型都规定了一个唯一的值，所以对“下一个头部”域的解释不存在二义性。接收方按头部出现的顺序依次处理每一个头部，利用每个头部中的“下一个头部”域来确定其后面跟着的内容。

有些头部类型具有固定的长度，例如基本头部正好是40字节的固定长度。为了指向紧跟基本头部后面的那一项，IPv6软件只需在基本头部的首地址上增加40即可。有些扩展头部没有固定的长度，这时头部必须含有足够的信息以使接收IPv6能确定头部在哪儿结束。例如，图24-5表示出一个IPv6可选项头部（options header）的一般形式，这种头部携带的信息与IPv4数据报中的可选项类似。

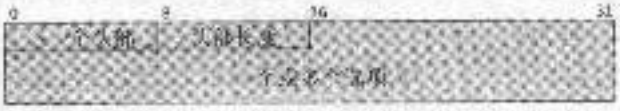


图24-5　具有显式长度指示的IPv6可选项扩展头部示意图

24.10　分片、重装和通路MTU

虽然IPv6的分片处理也类似IPv4的情况，但其细节有所不同。像IPv4那样，要将原始数据报的前缀复制到每个片中，并将载荷长度修改为片长度。与IPv4不同的是，IPv6不将包含分片信息的域放在基本头部，而是放在一个单独的扩展头部中，该头部的存在就表示该数据报是一个片。图24-6说明了IPv6的分片情况。

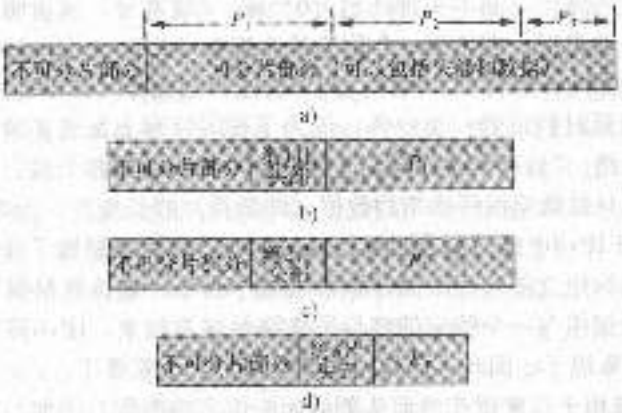


图24-6　IPv6的分片处理，数据报a)被分割成片b)，c)，d)

图中，“不可分片部分”指基本头部加上控制路由的头部。为了确保所有片按同一路径传输，要将不可分片部分复制到每一片中。

就像IPv4一样，要按照数据报去往的底层网络的最大传输单元（MTU）长度来选择片的长度。当然，最后一个片往往比前面的片要小，因为它包含的是原数据报按MTU尺寸分割后剩余的部分。

IPv6中的分片处理与IPv4有很大的不同。在IPv4中，当路由器收到一个长度大于数据报要去往的网络的MTU时，就由该路由器负责执行分片任务。但在IPv6中，由发送数据报的主机负责分片，即希望主机能选择一个数据报长度，期望以后不需要再分片。路径上的路由器收到长度大于网络MTU的数据报时，就向发送方发送一个差错报文并丢弃该数据报。

一台主机怎样选择数据报的长度才能避免再分片呢？主机必须了解去往目的地路径上的每一个网络的MTU，以便挑选一个合适的数据报长度去适应最小的那个MTU。从源端到目的地的路径上，其最小的MTU叫做路径MTU（path MTU），获悉通路MTU的过程叫通路MTU发现（path MTU discovery）。通常，通路MTU发现是一个迭代过程：一台主机发送一系列不同长度的数据报，看看它们能否无错到达目的地。如果需要分片，发送方主机将收到一个ICMP差错报文[○]。一旦某个数据报足够小到能通过网络而不被分片，则主机就找到了一个等于通路MTU的数据报长度。

概述如下：

> 在IPv6中，由发送方主机执行分片过程，路由器不参与。如果需要分片，发送方主机将收到中间路由器发送的ICMP差错报文。发送方会不断减小分片长度，一直到分片能最终送达目的地为止。

24.11 采用多重头部的目的

为什么IPv6要使用多个单独的扩展头部呢？有两个理由：

- 经济性。
- 可扩展性。

经济性很容易理解：将数据报的功能划成单独的多个头部的做法很经济，因为它可以节省空间。尽管IPv6包含很多特性，但是设计者还是希望一个数据报只使用其一个小的子集即可。有了多个单独的扩展头部后，IPv6就有可能定义出很多的功能特性，而无须要求每一个数据报头部都为每一特性至少保留一个域。例如，虽然很多IPv4数据报没有被分片，但它们的头部中仍有用于保存分片信息的域。相反，除非数据报被分片，否则IPv6的头部就不必为标识分片信息而浪费多余的空间。由于大多数数据报只需要少量的头部域，所以避免不必要的头部域就能节省可观的头部空间。另外，较短的数据报在传递过程中所花的时间也少，因此减小数据报长度也就等于减少了带宽消耗。

为了理解可扩展性，考虑为协议增加一个新特性的情况。象IPv4那样使用固定长头部格式的协议，要增加新特性就要对它做完全的改变——必须重新设计头部格式以便安排进支持新特性的域。然而，在IPv6中，现有的协议头部可以维持不变，只要定义一个新的"下一个头部"类型和一个新的扩展头部格式即可。

将新功能放入一个新扩展头部的做法，其主要优点在于：在改变因特网中所有计算机的功能之前，协议就有能力对这一功能特性进行实验。例如，假设两台计算机的所有者希望测试一种新的数据报加密技术，那么这两台计算机必须在一种实验性加密头部的细节方面达成一致。发送方将新的头部加入到数据报中，接收方解释接收数据报中的这种新头部。只要新头部出现在路由头部之后，因特网上发送方与接收方之间的路由器就能让该数据报传输通过，而不必去理解这个实验头部的含义[○]。一旦一个实验性的功能特性被证明可使用，即可将它列入到标准里面。

24.12 IPv6编址

类似IPv4，IPv6也为计算机与物理网络之间的每条连接分配了一个唯一的地址。因而，

○ IPv6包括新版本的ICMP。

○ 如果实验头部被错误地放置在路由信息头之前，路由器会将其丢弃。

如果一台计算机（例如一个路由器）连接着3个物理网络，该计算机就被分配给3个IPv6地址，与IPv4一样，IPv6将每一个这样的地址划分成一个前缀和一个后缀，前缀标识一个网络，后缀标识网上的某台特定的计算机。

尽管这两种协议采用了同样的方法给计算机分配地址，但IPv6的编址方案与IPv4仍存在明显的不同。第一，所有地址的细节都完全不同。类似于CIDR地址，前缀与后缀之间的边界可能出现在地址范围内的任何地方。与IPv4地址不同，IPv6地址具有多个层次结构。虽然地址分配不是固定的，但我们可以假设最高层次对应于ISP，下一层次对应于组织机构（例如公司），再下一层次则是站点，依此类推。第二，IPv6定义了一组特殊地址，它与IPv4的特殊地址截然不同。特别是，IPv6不包括针对特定远地网络进行广播的特殊地址。取而代之的是，任一个IPv6地址只归属于图24-7所示的3种基本类型之一。

如图所示，IPv6保留了单播和组播编址方案，但取消了直接广播，其原因是它会导致安全问题。为了处理有限广播（即本地网络上的广播），IPv6定义了一个特殊的组播组，这个组播组对应于本地网络上所有的主机和路由器。

任意播编址原先被称为簇（cluster）编址。这种编址的动机源自于一种对重复性服务的需求。例如，在网络上提供某种服务的公司，可以为几台提供这种服务的计算机分配一个任意播地址。当用户给这个任意播地址发送一个数据报时，IPv6将该数据报路由到组（即簇）中的某一台计算机。如果用户从另一个地点也给该任意播地址发送一个数据报，则IPv6可选择将该数据报路由到组中的另一台计算机，以允许有两台计算机同时来处理这些请求。

类型	用　途
单播	该地址对应于单台计算机，送往这种地址的数据报将沿着一条最短路径被路上到该计算机
组播	该地址对应于一组计算机，这些计算机可能在不同的地点，组内的成员关系在任何时刻都能改变。IPv6向组内的每个成员传送该数据报的一个副本
任意播	该地址对应于共享相同地址前缀的一组计算机，送往这种地址的数据报只会传递给它们中的一台计算机（例如最靠近发送方的那台计算机）

图24-7　IPv6地址的3种类型

24.13　IPv6冒分十六进制数表示法

IPv6地址占用128位，要写出这么长的数字是很不方便的。例如，用IPv4中采用的点分十进制数表示法写一个128位的数字：

105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255

为了减少写一个地址所用的字符个数，IPv6的设计者建议使用一种更紧凑的语法形式，叫冒分十六进制数表示法（colon hexadecimal notation，简写为colon hex）。其中，每16位为一组写成十六进制数，用冒号分隔每个组。例如，上面的地址用冒分十六进制数表示法时，变为：

69DC:8864:FFFF:FFFF:0:1280:8C0A:FFFF

从上例可见，表示同一个地址时，采用冒分十六进制数表示法所需的字符数少得多。另外，还有一种优化表示法叫零压缩（zero compression），又进一步减少了字符个数。零压缩用两个冒号代替连续的零。例如，地址：

FF0C:0:0:0:0:0:0:B1

可写成：

FF0C::B1

对于IPv6的大地址空间和所建议的地址分配方案，零压缩表示法特别重要，因为设计者认为很多IPv6地址都会包含零字串。特别是，为了便于从IPv4过渡到新的协议，设计者将IPv4现存的地址映射到IPv6的地址空间。任何以96个0位开头的IPv6地址，它的低32位都含有一个IPv4地址。

24.14 本章小结

虽然当前版本的IP多年以来工作得都很好，但因特网规模的指数增长意味着32位的地址空间最终会耗尽。IETF已经设计了IP的一个新版本，使用128位来表示每一地址。为了区别IP的新版本与当前版本，两个协议的命名都使用了它们的版本号：当前版本的IP是IPv4，新版本的IP是IPv6。

IPv6保留了IPv4中的很多概念，但在所有的具体细节上都做了改变。例如，像IPv4那样，IPv6是供无连接服务，两台计算机交换的报文叫数据报。然而，不像IPv4数据报那样在头部中为每一功能提供相应的域，IPv6为每一功能定义了单独的（扩展）头部。每个IPv6数据报这样构成：先是基本头部，然后跟着零个或多个扩展头部，最后是数据。

像IPv4那样，IPv6为每个网络连接定义了一个地址，因此连接到多个物理网络的一台计算机（如路由器）拥有多个地址。然而，在IPv6中重新定义了特殊地址。它定义了组播和任意播（簇）地址来取代IPv4的网络广播表示，这两种地址表示都对应于一组计算机。组播地址对应处在不同地点的一组计算机，把这些计算机当作单个实体来对待——组内的每台计算机都将收到发往该组的任何数据报的一个副本。任意播地址支持提供重复型服务——发往一个任意播地址的数据报只会传递给任意播组中的一个成员（例如，离发送方最近的成员）。

为使IPv6的地址易于为人们使用，设计者创建了冒分十六进制数表示法。这种方法将每16位作为一组写成十六进制数，并用冒号分隔开。零压缩消除了长串零的冗长表示。这种方法表示的结果比IPv4使用的点分十进制数形式更加紧凑。

练习题

24.1 从IPv4变成IPv6的主要动机是什么？

24.2 因特网通信的沙漏模型指的是什么？

24.3 列出IPv6主要的特性，并给出简短的描述。

24.4 最小的IPv6数据报头部有多大？

24.5 IPv6数据报头部中的“下一个头部”域指示什么内容？

24.6 IPv6数据报中的可分片部分指的是什么？

24.7 为什么IPv6要使用分开的扩展头部来代替单个的固定格式头部中的相关域？

24.8 列出3种IPv6地址类型，并对每种类型给出简短的说明。

24.9 编写一个计算机程序，读入一个128位的二进制数，然后用冒分十六进制数表示法将它打印输出。

24.10 将上一练习题的程序加以扩充，以实现零压缩表示。

第25章 UDP：数据报传输服务

25.1 引言

前一章讲述了由IP及实用于报告差错的辅助协议所提供的无连接分组传递服务。本章考虑互特网上广泛使用的两种传输层协议之一的UDP协议，它也是传输层上能单一提供无连接服务的协议。本章讨论UDP分组的格式以及使用UDP的方法。我们将会看到，虽然UDP既简单又灵活，但它也具有由于采用了尽力传递机制而带来的一些令人惊奇的特性。除了讨论UDP外，本章还涵盖了协议端口号这一重要概念。

下一章继续讨论另一个主要的传输层协议——TCP，再后面的章节则讨论因特网路由和网络管理，它们都要用到传输协议。

25.2 传输协议与端到端通信

正如前几章所述，网际协议提供跨越因特网的分组传送服务（即数据报可以从发送主机通过一个或多个物理网络到达接收主机）。尽管它有通过因特网传输业务的能力，但IP还缺乏一个最基本的特性：IP无法区分指定主机上运行的多个应用程序。如果用户在主机上同时运行一个邮件应用程序和一个Web浏览器或是运行指定应用程序的多个副本，这些程序必须有能力进行各自的通信。

IP是不能支持多个应用同时通信的，因为数据报头部中的域仅能标识计算机本身，也就是说，从IP的角度看，数据报中源和目的地址域只能标识计算机主机，它并没有包含更多的码位来标识主机上的应用程序。所以说，IP只是将一台计算机当做通信的一个端点（endpoint）。相比之下，传输层协议之所以被称为端到端协议（end-to-end protocol），是因为传输协议允许将单个应用程序看做通信的端点。TCP/IP协议的设计者不是采取在IP基础上增加附加特性的方法来标识应用，而是将端到端协议放置到另一个单独的层（第4层）里。

25.3 用户数据报协议

我们将看到，TCP/IP协议组包含两个传输协议，即用户数据报协议（User Datagram Protocol，UDP）和传输控制协议（Transmission Control Protocol，TCP）。两者最大的不同在于它们向应用提供的服务方面。UDP没有TCP那么复杂，而且最容易理解。这种简单性和易懂性也伴随着其所付出的代价——UDP并不是供典型应用所期待的服务类型。

UDP可以被表征为：

- 端到端。UDP是一个传输协议，它能区分运行在给定计算机上的多个应用程序。
- 无连接。UDP提供给应用的接口遵从无连接模式。
- 面向报文。使用UDP的应用进程所发送和接收的数据是单个报文。
- 尽力而为。UDP提供给应用的是与IP一样的尽力传递机制。
- 任意交互。UDP允许应用进程给很多其他应用进程发送数据，也允许从很多其他应用进程那里接收数据，或者只跟一个其他应用进程相互通信。

• 操作系统无关。UDP所提供的标识应用程序的方法，不取决于本地操作系统所使用的标识符。

UDP之所以呈现“尽力而为”这一最重要的特征，是因为UDP使用IP来直接传输它的数据报。其实，UDP有时被描述成一个细薄的协议层，它只是为应用进程提供了发送和接收IP数据报的能力而已。概括如下：

> UDP提供端到端服务，它允许应用进程发送和接收单个报文，每个报文都被装进单个数据报中进行传输。应用进程既可选择只限于与一个其他应用程序通信，也可选择与多个应用程序通信。

25.4 无连接的通信模式

UDP采用无连接（connectionless）通信模式，也就是说，使用UDP的应用进程在发送数据之前不需要预先建立通信连接，在通信结束后也无须通知网络。实际上，应用进程可以在任何时候生成和发送数据，而且UDP还允许在两个报文传输之间延迟任意长的时间。UDP不需要维护通信状态，也不使用控制报文，通信业务仅由这些数据报文本身构成。更为重要的是，如果双方应用进程都停止发送数据，它们之间就不再交换任何其他分组。因此，UDP的传输开销极其低。概述如下：

> UDP是无连接方式的，这意味着一个应用进程可以在任何时候发送数据，而且除了传输携带用户数据的分组外，它不再传输任何其他分组。

25.5 面向报文的接口

UDP提供给应用程序的是面向报文的（message-oriented）接口。应用进程每次向UDP请求发送一块数据时，UDP会将数据放到一个单独的报文中来传输。UDP不会将报文分割成多个分组进行传递，也不会将短的报文组合在一起进行传递——应用进程发送的每个报文都（原样地）通过本特网传输并最终传递给接收方。

面向报文的接口对于程序员具有几个重要的影响。在正面影响方面，使用UDP的应用程序能依赖这个协议来保留数据边界。即UDP传递给接收应用进程的每个报文与发送方发出的报文完全一样；在负面影响方面，每个UDP报文必须要适合于单个IP数据报的长度要求。因而，IP数据报的大小就形成了对UDP报文大小的绝对限制。更重要的是，UDP报文的大小可能会导致底层网络的利用率下降。如果应用进程发送太小的报文，就会造成数据报头部与数据部分的比率变大；如果应用进程发送太大的报文，则造成数据报长度大于网络MTU而要求分片处理。

如果允许UDP发送大报文的话，将会产生一种有趣的意外情况。通常，一个应用程序员可以通过采取大批量传输的方法来实现高效率的通信。例如，程序员往往被鼓励定义大的I/O缓冲区，并且规定按缓冲区大小进行批量传输。然而，采用UDP来发送大报文将会导致效率下降，因为大的报文会造成分片处理。甚至更加令人意外的是，在发送计算机上也会发生分片操作，即应用进程要UDP发送一个大报文，UDP将整个报文放进用户数据报中，并将该用户数据报封装成IP数据报，而IP在发送该数据报之前必须完成分片处理。

要点 虽然程序员的直觉提示使用大的报文会提高通信效率，但是如果一个UDP报文的大小超过了网络的MTU，IP就会对数据报进行分片，从而导致效率的降低。

因此，很多使用UDP的程序员都会适当选择报文的长度，使产生的数据报能适合标准的MTU大小。特别地，由于因特网的绝大多数底层网络规范都支持1500B的MTU，程序员经常会选择1400B或1450B大小的报文长度，以留下足够的空间来容纳IP和UDP的头部信息。

25.6 UDP通信语义

UDP的所有通信业务都是利用IP来传递，而且它提供给应用的也是与IP完全一样的尽力传递语义。这就意味着报文可能会：

- 丢失，
- 重复，
- 延迟，
- 乱序，
- 损坏。

当然，UDP并不是有意要引入这些传递方面的问题，只不过是因为利用了IP来发送报文，而且又没有去检测或纠正这些方面的问题。UDP的尽力传递语义对应用有重要的影响——要么应用程序对以上问题天生具有免疫能力，要么程序员采取附加的措施来检测和纠正这些问题。作为一个容许分组出错的应用例子，我们考虑一种音频传输的情况。如果发送者将少量的音频数据放进每个报文里传输，那么只要丢失一个分组就会在重放声音的时候产生一个小的间隙，使人听到“扑”或“嗒”的一声。尽管这不是我们所想要的，但也是有点烦人。作为与之相反的另一个极端，考虑一个在线购物应用系统。这种应用程序不能使用UDP来编程，因为分组的出错可能会造成灾难性的后果（例如，如果载送分类定单的报文发生了重复，将会产生两份定单，就会从购物者的信用卡中扣去双份的货款）。

概括如下：

> 由于UDP就像与IP一样的尽力传递语义，所以UDP报文在传输中也可能会出现丢失、重复、延迟、乱序或者被损坏。UDP只适用于那些能够容许分组传递出错的应用，比如语音或视频之类的业务。

25.7 交互模式和广播传递

UDP允许采用4种交互通信方式：

- 一对一，
- 一对多，
- 多对一，
- 多对多，

这就是说，使用UDP的应用程序有了选择余地，它可以选择“一对一”交互方式与另一个应用程序交换报文，选择“一对多”交互方式向多个接收者发送报文，或者选择“多对一”交互方式从多个发送者那里接收报文。最后，一组应用程序可以建立“多对多”交互方式来实现彼此之间的报文交换。

虽然可以通过向每个接收者发送一个报文的单个副本的方法来实现“一对多”交互方式，但是UDP允许采用更有效的交换方式，它不要求应用进程重复地给多个接收者发送报文，而是利用IP的组播或广播机制来传输报文。为此，发送者采用IP广播地址作为目的IP地址，例如，利用IP的受限广播地址255.255.255.255可发送本地广播。类似地，UDP也允许应用进程发送

组播报文。利用广播或组播方式传递报文的做法在以太网上特别有用，因为它的底层硬件能有效地支持这两种传输类型。

25.8 用协议端口号标识端点

UDP应该如何正确地将应用程序标识为端口呢？看起来好像UDP也可以采用与操作系统同样的机制，其实不然，因为UDP必须跨越异构的计算机，它们不存在共同的机制。例如，有些操作系统采用进程标识符，有些则采用作业名称，而另一些则采用任务标识符。因此，在一个系统上有意义的标识符，到了另一个系统上可能就没有意义了。

为了避免含糊性，UDP定义了一个叫做协议端口号（protocol port number）的标识符抽象集，它独立于底层的操作系统。每一台实现了UDP的计算机，都必须提供协议端口号与操作系统所用的程序标识符之间的映射关系。例如，UDP标准定义了协议端口号7作为回应（echo）服务的端口，而端口号37作为时间服务器（timeserver）服务的端口。所有运行UDP的计算机都能识别这种标准的、独立于底层的操作系统的协议端口号。因此，当有一个UDP报文到达端口号7的时候，UDP协议软件就必然知道在本地计算机中哪一个程序实现了回应服务，从而把收到的报文传递给该程序。

应用程序为套接字填充地址和协议端口号所给出的规定，决定了通信的模式。例如，为了参与“一对一”通信，应用程序就要指定本地端口号、远地IP地址和远地端口号；UDP只把从指定了地址和端口号的发送者那里发出的报文传递给本地应用进程。如果要参与“多对一”通信的话[⊖]，应用进程要指定本地端口号，但通知UDP：远地端点可以是任何系统（即有多个远地端点），然后UDP就会把到达指定本地端口的所有报文传递给本地应用进程[⊖]。

25.9 UDP数据报格式

每个UDP报文被称为用户数据报（user datagram），它由两部分构成：一个用来指定发送端和接收端应用程序的短头部，一个携带着发送数据的载荷部分。图25-1表示了用户数据报的格式。



图25-1 带8位字节头部的UDP用户数据报的格式

UDP头部的前两个域包含16位长度的协议端口号，其中“UDP源端口”域含有发送应用进程的端口号，“UDP目的端口”域含有报文欲到达的应用进程的端口号。“UDP报文长度”域指定了以字节单位计量的UDP报文总长度。

25.10 UDP校验和伪头部

虽然UDP头部中含有一个16位长的“UDP校验和”域，但它是可选的。发送者既可选择计算校验和，也可将校验和域的所有位置成全0。当一个报文到达目的地时，UDP软件要检查

⊖ 这里假设本地端是指“多对一”中的“一”方，而远地端是指“多”方。——译者注

⊖ 对指定的端口，只能有一个应用进程请求接收发往它的报文。

校验和域，只有当该域的值非零时才进行验证[⊖]。

注意，UDP头部中除了协议端口号外，并不包含发送方和接收方的其他任何标识。特别是，UDP假设源和目的地址是包含在携带该UDP报文的IP数据报里的，所以在UDP头部中就不再含有IP地址了。

在UDP头部中省略源和目的IP地址，会使UDP报文更短小有效，但也存在引入差错的可能。特别是，如果IP出现功能性故障而将UDP报文传递到了错误的目的地，UDP将无法利用它的头部域来确定是否发生了差错。

为了使UDP能够验证报文是否到达了正确的目的地而又不增加额外的头部域开销，UDP扩充了校验和，在计算校验和的时候，UDP软件还要包含一个伪头部（pseudo header），其中含有从IP数据报那里获得的源地址、目的地址和类型域，还含有UDP的数据报长度。就是说，发送方在计算校验和时要假设UDP头部好像还包含了额外的域。类似地，为了验证校验和，接收方必须获知UDP的长度，还要从IP数据报那里获得源地址、目的地址和类型域。在验证校验和之前，接收方会将这些内容追加到UDP报文中。图25-2所示为伪头部中的域。

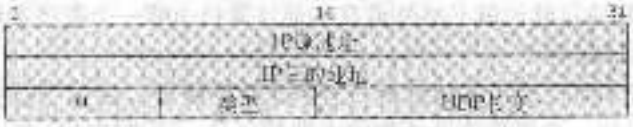


图25-2　用于计算UDP校验和的伪头部示意图

25.11　UDP封装

像ICMP那样，每个UDP数据报被封装在一个IP数据报中，然后发送到因特网上传输。图25-3所示为这种封装过程。

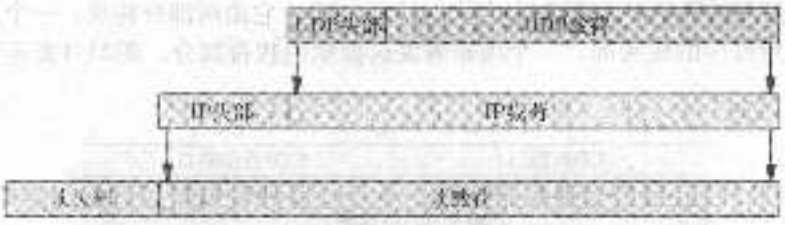


图25-3　发送UDP报文时的封装过程

25.12　本章小结

用户数据报协议提供了端到端报文传递的手段，它可以从一台计算机上运行的应用进程传递到另一台计算机上运行的应用进程。UDP提供与IP一样的尽力传递语义，这就意味着报文会出现丢失、重复或乱序等现象。UDP的无连接通信方式具有一些优点。这主要体现在它拥有能在多个应用程序间实现“一对一”、“一对多”和“多对一”交互的能力。

为了保持对底层操作系统的独立性，UDP采用短整数型协议端口号来区分应用程序。在运行UDP协议软件的计算机上，必须将每个协议端口号映射到该计算机所采用的相应标识机制（例如进程标识符）上。

⊖　像IP那样，UDP使用反码校和。如果计算得到的校验和的值为零，发送方就要校验和为全1。

UDP的校验和是可选的——如果发送方将校验和域填为零，接收方就不必验证校验和了。为了能够验证UDP数据报是否到达了正确的位置，又在数据报基础上外加一个伪头部，再计算得到UDP校验和。

UDP需要两级封装。每个UDP报文封装在IP数据报中以便在因特网上传输，而该数据报又被封装在帧中以便在某个物理网络上传输。

练习题

25.1 IP协议和端到端协议之间在概念上存在什么样的差异？

25.2 试罗列出UDP的特点。

25.3 在交换数据前，使用UDP的应用程序需要交换UDP控制报文吗？试解释。

25.4 请计算出UDP报文的最大可能长度（提示：整个UDP报文必须能被装进一个IP数据报中）。

25.5 如果一个包含1500B数据的UDP报文在以太网上发送，会发生什么现象？

25.6 如果一个应用程序要利用UDP在以太网上发送一个8KB的报文，那么会有多少帧在网络上传输？

25.7 UDP具有什么样的语义？

25.8 在“一对一”通信方式中，应用程序必须指定什么端点值？“一对多”呢？“多对一”呢？

25.9 UDP的伪头部指的是什么？什么时候用到它？

25.10 给定一个以太网帧，要确定帧中是否携带了一个UDP报文，需要检查什么域？

第26章 TCP：可靠的传输服务

26.1 引言

前面讲述了由IP以及运行在IP之上的UDP所提供的无连接分组传递服务。本章考虑一般性的传输协议，主要讲解在互特网上使用的主要传输协议——TCP，然后讲解它是如何提供可靠传递服务的。

TCP完成了一个看似不太可能完成的任务：在因特网上发送数据时，它利用IP提供的不可靠数据报服务，为应用程序提供可靠的数据传递服务。TCP必须为数据的丢失、延迟、重复、乱序传递作出补偿，而且它这样做时，还必须不能让底层的网络和路由器过载。在介绍TCP为应用提供的服务之后，本章重点讲解TCP为实现可靠性所采用的技术。

26.2 传输控制协议

程序员总认为可靠性是计算机系统的基础。例如，在编写一个向其I/O设备（如打印机）发送数据的应用程序时，程序员总是假定数据能准确地到达设备或是在出现错误时，操作系统能通知应用程序。也就是说，程序员总是假定底层系统能保证数据被可靠地传递。

为了让程序员在创建使用因特网进行通信的应用程序时，仍可以遵循常规的编程技术，协议软件必须提供与常规计算机系统一样的语义：软件必须保证迅速而又可靠的通信。数据必须按发送的顺序传递，而且不能出现丢失或重复现象。

在TCP/IP协议组中，传输控制协议（Transmission Control Protocol，TCP）提供可靠的传输服务。TCP很好地解决了一个难题——虽然其他协议早已出现，但还没有哪个通用的传输协议被证明工作得更好。因此，大多数因特网应用都建立在TCP的基础之上。

概括如下：

> 在因特网中，传输控制协议（TCP）是一个提供可靠性的传输层协议。

26.3 TCP为应用提供的服务

TCP提供的服务有7个主要特点：

- 面向连接。TCP提供面向连接的服务，应用程序必须首先请求建立一个到目的地的连接，然后使用这个连接来传输数据。
- 点对点通信。每个TCP连接上只有两个端点。
- 完全的可靠性。TCP能保证在一个连接上发送的数据被正确地传递，且保证数据的完整和按序到达。
- 全双工通信。TCP连接允许数据在任何一个方向上流动，并允许任何应用程序在任何时刻发送数据。
- 流接口。TCP提供一个流接口，利用它应用进程可以在一个连接上发送连续的字节流。TCP不必将数据组合成记录或是报文，也不要求传递给接收应用进程的数据段大小一定

要与发送端所送出的数据段大小相同。

- 可靠的连接建立。TCP允许两个应用进程可靠地开始通信。
- 友好的连接关闭。在关闭一个连接之前，TCP必须保证所有数据已经传送完毕，并且通信双方都要同意关闭这个连接。

概括如下：

> TCP提供可靠的、面向连接的、全双工的流传输服务，允许两个应用程序建立一个连接，并在任何一个方向上发送数据，然后终止连接，每个TCP连接都被可靠地建立和友好地终止。

26.4 端到端服务与虚拟连接

像UDP那样，TCP也被归类为端到端（end-to-end）协议，因为它提供在一台计算机上的应用进程与另一台计算机上的应用进程之间的通信能力。TCP是面向连接（connection oriented）的协议，因为应用进程在传输数据前，必须先请求TCP建立一个连接，并且在传输完成后要关闭这个连接。

由TCP提供的连接叫做虚拟连接（virtual connection），因为它由软件实现。事实上，底层的因特网系统并不对此连接提供硬件或软件支持，只是由两台机器上的TCP软件模块通过报文交换来实现一个连接的幻象。

每个TCP报文被封装在一个IP数据报内并通过因特网传输。当数据报到达目的主机时，IP将数据报的内容传递给TCP。需要注意的是，虽然TCP使用IP来发送报文，但IP并不阅读或解释这些报文。实际上，IP只是把每个TCP报文当作要传输的数据来处理，而TCP也只是把IP当成一个分组通信系统，在连接两端的TCP模块之间提供通信服务。如图26-1所示为TCP是如何看待底层因特网的。

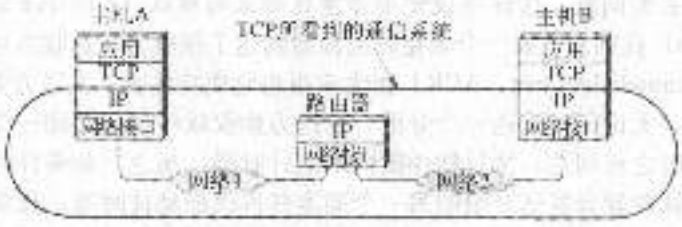


图26-1 TCP是如何看待底层因特网的示意图

如图26-1所示，在一个虚拟连接上的两端都需要有TCP软件，但中间的路由器却不需要。从TCP的角度来看，整个因特网是一个通信系统，它能接受和传递报文，但对报文的内容不作任何改变或解释。

26.5 传输协议所采用的技术

端到端传输协议必须经过缜密的设计才能实现高效可靠的传输。否则，会出现以下一些问题：

- 不可靠的通信。因特网上发送的报文可能会出现丢失、重复、损坏、延迟或者乱序传递。
- 端系统重启。在通信的任何时刻，通信两端的端系统都有可能崩溃和重启。不同的会话之间不能出现混淆，有些嵌入系统完成重启所需要的时间可能会小于一个分组在因特网

上传输所花的时间。

- 异构的硬系统。处理能力强的发送方产生数据的速度可能会很快，使得速度慢的接收方过载。
- 因特网上的拥塞。如果所有发送者都拼命发送数据，那么中间交换机和路由器就会因太多的分组到达而发生过载，这很类似于拥堵的高速公路。

我们已经看到数据通信系统为解决某些问题而使用的一些基本技术。例如，为了弥补传输过程中码位改变的差错，协议可能包括了奇偶校验位（parity bit）、校验和（checksum）或是循环冗余校验（cyclic redundancy check, CRC）等检测差错的功能。传输协议除检测错误外，还要做更多的事情——它们会采用各种能修复或是避开问题的技术去解决问题。特别是，传输协议使用了各种工具来处理一些最复杂的通信问题。后面几节将讨论这些基本技术方面的内容。

26.5.1　对付分组重复和乱序传递的排序技术

为了对付分组重复和乱序传递，传输协议采用了排序（sequencing）技术。发送端为每个分组附加一个序号；接收端保存当前按顺序收到的最后一个分组的序号，同时保存一个乱序到达的分组列表。当有一个分组到达时，接收方通过检查它的序号来决定如何处理：如果该分组是期待的下一个分组（即按顺序到达的分组），协议软件就将它递交给上一层，并检查它的列表，看是否还有其他分组也可以向上递交；如果分组是乱序到达，协议软件就将它添加到列表中。

排序同时也解决分组重复的问题——接收方在检查一个到达分组的序号时，要验证分组的重复性。如果这个序号的分组已经向上递交过，或是与列表中正在等候的某个分组的序号相匹配，那么协议软件就会丢弃这个新的副本。

26.5.2　对付分组丢失的重传技术

为了对付分组丢失问题，传输协议使用带重传的正向确认（positive acknowledgement with retransmission）机制。只要一个帧完好无损地到达了接收方，接收方协议软件就要发送一个短的确认（acknowledgement，ACK）报文来报告它成功接收。发送方要负责确保每一个分组都能成功传输。无论何时发送一个分组，发送方协议软件都会启动一个计时器。如果一个确认在计时器超时之前到达，协议软件就会取消计时器；反之，如果计时器在确认到达之前已经超时，协议软件就会发送分组的另一个副本并再次启动计时器。通常把发送第二个副本的行为叫做重传（retransmitting），所发送的副本称为重传数据（retransmission）。

当然，如果硬件故障导致网络永久性断开，或者接收方计算机崩溃，重传也不可能取得成功。因此，重传数据的协议通常要限定一个最大的重传次数。当达到这个最大重传次数时，协议就停止重传并宣布不可能通信。

注意，如果一个分组被延迟了，那么重传可能会导致分组的重复。因此，加入了重传机制的传输协议通常被设计也能处理分组重复的问题。

26.5.3　避免分组重复的技术

超长的延迟会导致重放错误（replay error）。在这种错误中，一个延迟很久的分组会影响后续的通信。例如，考虑下列事件的顺序：

（1）两台计算机同意在下午1点开始通信。

（2）一台计算机向另一台计算机发送10个分组的序列。

（3）硬件故障使使第三个分组异常延迟。

(4) 改变路径，以避开这个硬件故障。

(5) 发送计算机上的协议软件重传第三个分组，剩余分组被无差错地传送。

(6) 在下午1:05，两台计算机同意再次进行通信。

(7) 在第二个分组到达后，前一次会话中被延迟的第三个分组的副本到达。

(8) 第二次会话中发送的第三个分组也到达。

除非很小心地设计传输协议来避免这类问题，否则在较早会话期间发送的分组有可能会在后来的会话期间被接受，而正确的分组却被当成重复分组被丢弃。

重放现象也可能出现在传输控制分组（即建立或中止通信的分组）的时候。为了理解这一问题所涉及的范围，考虑这样一种情况：两个应用程序建立一个TCP连接、通信、关闭这个连接，然后再建立一个新的连接。由于关闭前一个连接的报文会出现重复，它的一个副本可能被拖延很长时间，一直到第二个连接建立时，还须设计一种协议来防止这个重复的报文将第二个连接关闭。

为了防止重放错误，协议软件用一个唯一的ID（例如，建立会话的时间）标记每一次会话，并要求这个ID出现在每一个分组中。协议软件会丢弃任何到达但含有不正确ID的分组。为了避免重放错误，在相当长的一段时间（如几小时）内都不能再重复使用同一个ID。

26.5.4 防止数据过荷的流量控制技术

有几种技术可用来防止速度快的计算机因发送太多数据而造成速度慢的计算机过荷问题。我们使用术语流量控制（flow control）来指处理这类问题的技术。流量控制最简单的形式是一种停—走（stop-and-go）系统，在该系统中，发送方在发送完一个分组后就要等待接收方的回答。当接收方准备好接收另一个分组时，就会发送一个控制报文，通常就是某种形式的确认。

虽然停—走协议能防止过载，但它导致了极低的吞吐率。要理解一点，考虑一个分组大小为1 000字节、吞吐量为2Mbit/s、延迟时间为50ms的网络，看看将会发生什么情况。网络硬件设备以2Mbit/s的速率把数据从一台计算机传送到另一台计算机，但是每发送一个分组后，发送方必须等待100ms，才能发送另一个分组（即分组到达接收方需50ms，传回确认也需50ms）。因此，使用停—走协议发送数据的最大速率为每100ms一个分组。若以位传输速率来表示，停—走协议的最大速率只达到80 000bit/s，这只是硬设备容量的4%。

为了获得高的吞吐率，传输协议使用称为滑动窗口（sliding window）的流量控制技术。发送方和接收方都被编程使用固定的窗口大小（window size），它是发送方收到确认前可以发送的最大数据量。例如，发送方和接收方同意窗口大小为4个分组。发送方在开始发送数据时，提取数据填充这4个分组（即第1个窗口），并发送每个分组的副本。在大多数传输协议中，发送方保留一份副本以备万一需要时重传。接收方必须预先分配好缓冲区空间以接收整个窗口。当分组顺序到达时，接收方把分组传给接收应用进程并发回一个确认给发送方。当确认到达后，发送方丢弃已被确认的副本并发送下一个分组。图26-2表示了这种机制为什么称为滑动窗口的原因。

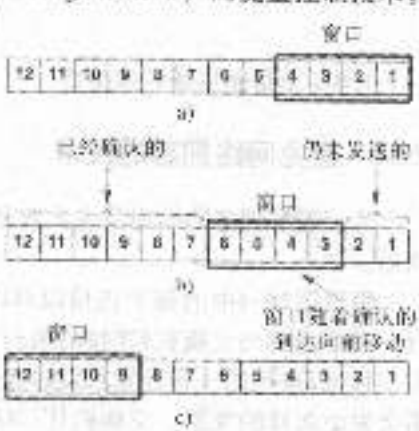


图26-2 滑动窗口示意图，a) 初始阶段，b) 中间阶段，c) 最后位置

滑动窗口能显著地提高吞吐率，我们来比较一下使用停—走方案和滑动窗口方案的传输顺序图，图26-3

包含了传输4个分组过程的比较。

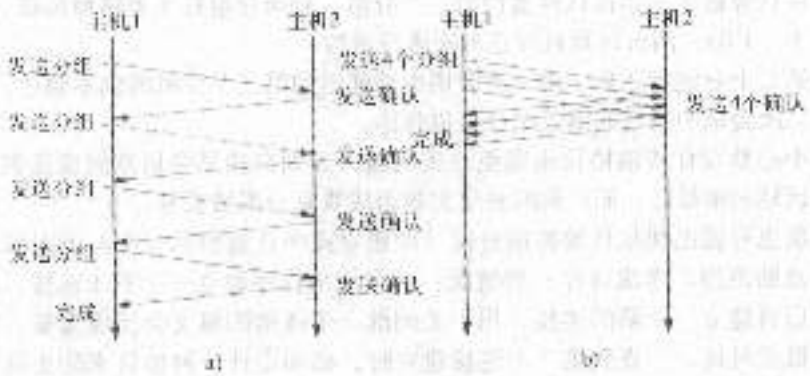


图26-3　两种传输机制的比较a)等一走方案；b)滑动窗口方案

在图26-3a中，发送方发送了4个分组，在每次在发送后续分组之前，都要等待前一分组的确认。如果沿着网络的一条路径发送一个分组所要求的延迟时间为N，那么发送4个分组所要求的总时间就是8N。在图26-3b中，发送方在等待确认到达前就可发送窗口中的所有分组。图中显示出在后续发送的分组之间都有一段小的延迟，这是因为传输并不是立即发生的——需要一段很短的时间（通常是若干微秒）让硬件完成分组的发送，然后开始下一个分组的发送。因此，发送4个分组所需要的总时间是2N + ε，这里ε是指很小的时延。

为了理解滑动窗口的意义，想象一个涉及许多分组的扩展通信。在这种情况下，传输所要求的总时间很长，以至ε可被忽略不计。对于这种网络，滑动窗口协议能充分提高性能。潜在的性能提高是：

$$T_w = T_0 \times W \tag{26.1}$$

这里：T_w是使用滑动窗口协议所能获得的吞吐率，T_0是使用停一走协议所能获得的吞吐率，W是窗口大小。式（26.1）说明了为什么在图26-3b中的滑动窗口协议的吞吐率大约是图26-3a中的停一走协议的4倍。当然，吞吐率不能仅靠增加窗口的大小来任意增大。底层网络的带宽设定了上限，即码位发送的速度不能快于承载它们的硬件的带宽。因此，式（26.1）可以被重写为：

$$T_w = \min(B, T_0 \times W) \tag{26.2}$$

其中，B是底层硬件的带宽。

26.6　避免网络拥塞的技术

为了理解拥塞的出现是多么容易，考虑一个由两台交换机连接4台主机的网络，如图26-4所示。



图26-4　由两台交换机连接的4台主机

假设图26-4中的每个连接以1Gbit/s的速率运行，我们考虑与交换机1连接的两台计算机试图向与交换机2连接的一台计算机发送数据的情况，看看会发生怎样的现象。交换机1以2Gbit/s的汇聚速率接收数据，但却只能以1Gbit/s的速率向交换机2转发。这种局面就是所谓的拥塞（congestion）。即使交换机将数据暂时保存在内存中，拥塞

仍然会增加延迟。如果拥塞现象持续下去，交换机将会发生内存溢出，并开始丢弃分组。虽然重传能恢复丢失的分组，但是重传将向网络发送更多的分组。因此，如果这种情况一直持续的话，整个网络会变得极其不稳定。这种情况就称之为拥塞崩溃（congestion collapse）。在因特网上，拥塞经常出现在路由器中。传输协议通过监视网络，一旦发现拥塞就迅速做出反应，力图避免拥塞崩溃。这里有两个基本方法：

- 拥塞出现时，安排中间系统（即路由器）去通知发送方。
- 利用延迟增加量或分组丢失率作为对拥塞程度的评估。

要实现前一个方法，可以在拥塞出现的时候，让路由器发送一个特殊报文给分组的源发端，或者让路由器给每个由于拥塞而产生延迟的分组头部设置一个码位。采用第二种方法时，接收这种分组的计算机就会在确认报文中包含相关信息去通知源发端。[⊖]

根据延迟量和分组丢失率来评估因特网上的拥塞程度是合理的，因为：

> 现代网络的硬件设备的工作性能都很好，大多数的延迟和分组丢失是由于拥塞而非硬件故障所引起的。

对于网络拥塞而作出恰当的反应，就是降低正在发送分组的速率。滑动窗口协议通过暂时减小窗口的大小，也能达到降低速率的效果。

26.7 协议设计技巧

虽然用于解决特定问题所需要的技术众所周知，但有两个理由说明协议的设计过程并不简单。第一，为了实现有效的通信，必须仔细考虑好各个具体环节——很小的设计错误都可能会导致错误的操作和不需要的分组或延迟。例如，如果使用序号，则每个分组都必须在头部中包含一个序号域。这个域必须足够大，才能使序号不会被频繁地重用，但也应尽量小，以免浪费不必要的带宽。第二，协议机制可能以预料不到的方式相互作用。例如，考虑流量控制机制与拥塞控制机制之间的相互作用。滑动窗口方案倾向于使用更多的底层网络带宽以提高吞吐率；而拥塞机制则恰好相反，它要通过减少一次注入的分组数目来避免网络崩溃。要在滑动窗口与拥塞控制之间取得平衡，可能是很棘手的，所以很难设计得使两方面都兼顾得很好。也就是说，太激进的流量控制可能造成网络拥塞，而太保守的拥塞控制又会降低所需的吞吐率。试图在出现拥塞时从激进行为切换到保守行为的设计，又可能会发生振荡，即逐渐增加带宽的使用直到网络开始出现拥塞，再逐渐减少带宽的使用直到网络变得稳定，然后又开始重复上述过程。

计算机系统的重启问题对传输协议设计又提出了一个严峻的挑战。想象这样一种情况：两个应用程序建立了一个连接，并始发送数据，然后接收数据的计算机被重新启动。虽然重启的计算机上的协议软件对此连接一无所知，但是发送计算机上的协议软件却认为该连接仍然有效。如果协议设计得不仔细，重复的分组也会造成计算机错误地创建一个连接并在半途中开始接收数据。

26.8 用来对付分组丢失的技术

TCP采用前面讲述的那一种技术来实现可靠性呢？答案挺复杂，因为TCP采用了多种方案并将它们进行了新的组合。与我们想的一样，TCP采用重传（retransmission）来弥补分组的

⊖ 从出现拥塞到源发端收到通知的这段延迟可能会比较长。

丢失。由于TCP提供双向的数据流，因此通信的双方都要参与重传。当TCP收到数据时，它就给发送方发回一个确认（acknowledgement）。发送方只要发送数据，TCP就会启动一个计时器，如果计时器超时，则重传数据。因此，基本的TCP重传操作如图26.5所示。



图26-5　一个分组丢失后TCP的重传示意图

TCP的重传方案是它获得成功的关键，因为它能处理跨越因特网上任意路径的通信。例如，一个应用程序能够通过卫星信道向另一个国家的某台计算机发送数据，与此同时，另一个应用程序正通过局域网向隔壁房间的某台计算机发送数据。TCP必须准备为任何一次连接中出现的丢失报文进行重传。问题是：TCP在重传之前应该等待多长时间呢？从局域网上某台计算机发回的确认一般预计在几毫秒内就能到达，而从卫星连接中发回确认则需要好几百毫秒。一方面，为这种确认等待过长的时间会使网络处于空闲而无法使吞吐率最大，那么在局域网中TCP就不应该在重传之前延迟太长时间。另一方面，在一个卫星连接上过快地重传分组并不会工作得很好，因为那些不必要的流量消耗了网络的带宽并降低了吞吐率。

TCP还面临着一个比区分本地和远地目的地更为困难的挑战，即数据报的突发可能导致网络拥塞，而网络拥塞又会导致给定通路上的传输延迟急剧变化。事实上，从发送一个报文到接收一个确认所需的总时间，可能就在几个毫秒数量级内增加或减少，概括如下：

> 数据到达目的地和返回确认所要求的延迟，取决于因特网中的业务量以及到目的地的距离。由于TCP允许多个应用程序与多个目的地并发通信，而且又由于网络业务量情况影响到延迟，因此TCP必须对付可能急剧变化的各种延迟。

26.9　自适应重传技术

在TCP出现之前，传输协议的重传延迟时间都规定为一个固定的值，即协议设计者或管理者为期望的延迟选择一个足够大的值。设计者在设计TCP时意识到，在因特网中采用一个固定的超时值并不会运行得很好。因而他们又选择了制定自适应的（adaptive）TCP重传方案。也就是说，TCP监视着每一个连接上当前的延迟，并调整（即改变）重传计时器以便适应条件的变化。

TCP如何监视因特网延迟呢？事实上，TCP并不知道因特网各个部位在任何时刻的精确延迟，但TCP可以通过测量收到一个响应报文所需的时间来估计每个活动连接的一个往返延迟（round-trip delay）。当发送一个期望响应的报文时，TCP记录报文发送的时间；当响应到来时，TCP从当前时间减去报文发送时的时间，从而得到该连接上新的往返延迟估计值。在多次发送数据报和接收确认后，TCP就产生了一系列的往返延迟估值，并利用统计函数产生出一个加权平均值。除了加权平均值外，TCP还保留了一个方差估值，利用所估计的均值和方差的

线性组合，即可计算出重传所需的等待时间。

经验表明，TCP自适应重传的效果很好。当延迟因分组流量突发而增加时，利用方差估值有助于TCP快速地作出反应。如果在经过短暂的突发后使延迟恢复到一个较低的值，利用加权平均值则有助于TCP重新设置重传计时器。当延迟保持常量时，TCP把重传超时值调整到比平均往返延迟稍大一点；当延迟开始变化时，TCP把重传超时值调整到比平均值更大的值，以适应峰值情况的发生。

26.10 重传时间的比较

为了理解为什么自适应重传能有助于TCP在每个连接上达到最大的吞吐率，考虑两个有不同往返延迟的连接上分组出现丢失的情况。例如，图26-6表示了两个这种连接上的业务量。

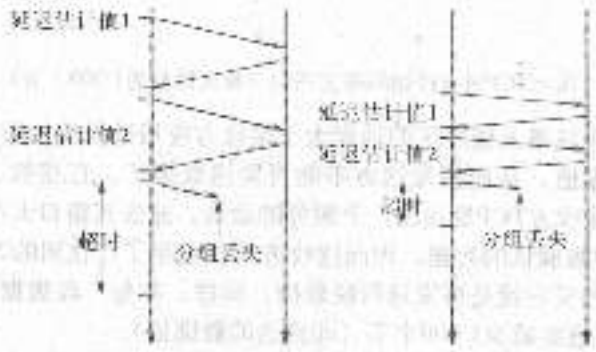


图26-6 具有不同往返延迟的两个连接上所发生的超时与重传

如图26-6所示，TCP将重传超时值设置到比平均往返延迟值稍大一点。如果延迟大，TCP使用一个大的重传超时值；如果延迟小，TCP就使用一个小的超时值，目的就是为了等待足够长的时间以确定一个分组是否被丢失，但又不至于等待太长的时间。

26.11 缓冲、流控与窗口

TCP使用一种窗口（window）机制来控制数据流，与前面讲述的那种简化的基于分组窗口方案不一样的是：TCP的窗口是以字节计量的。当一个连接建立时，连接的每一端都分配一个缓冲区来保存输入的数据，并将缓冲区的大小发送给另一端。当数据到达时，接收方TCP发送一个确认报文，指明自己剩余的缓冲区大小。TCP用术语窗口来指任意时刻可用的缓冲空间的大小，指出窗口大小的通知称为窗口通告（window advertisement）。接收方在发送的每个确认中都包含一个窗口通告。

如果接收方应用进程读取数据的速度与数据到达的速度一样快，接收方就在每个确认中发送一个正的窗口通告。但是，如果发送方操作的速度快于接收方（如由于CPU速度更快），那么输入的数据最终将填满接收方的缓冲区，导致接收方通告一个零窗口（zero window）。发送方在收到一个零窗口通告时必须停止发送，直到接收方重新通告一个正的窗口。图26-7说明了窗口通告过程。

在图26-7中，发送方使用的最大段长是1 000字节，传输开始于接收方通告一个2 500字节的初始窗口。发送方立刻传输3个数据段，其中两段含1 000字节的数据，一段含500字节。在每段数据到达时，接收方产生一个确认，其中的窗口值已经减去了已到达的数据量。

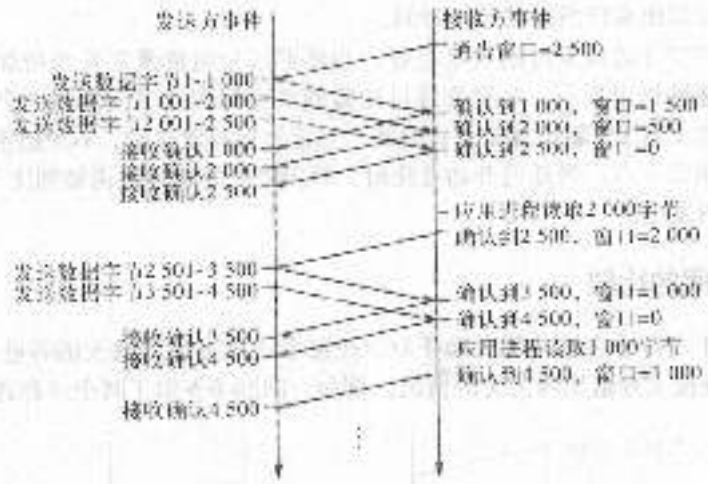


图26-7　演示TCP流量控制的报文序列（最大段长为1 000字节）

在这个例子中，前3个段填入缓冲区的速度大于接收方应用进程消化数据的速度，因而使通告的窗口大小达到了零值，从而使发送方不能再发送数据了。在接收方应用进程消化掉2 000字节的数据之后，接收方TCP发送出一个额外的确认，通告其窗口大小为2 000字节。在计算窗口大小时，要去掉被确认的数据，因而接收方通告它除了已收到的2 500字节之外还能接收2 000字节。发送方的反应就是再发送两段数据。同样，在每一段数据到达时，接收方发送一个确认，其中的窗口值要减少1 000字节（即到达的数据量）。

窗口大小又一次减到零值，迫使发送方停止传输数据。最终，接收方应用进程又消化掉了一些数据，因而接收方TCP又发送一个窗口大小为正的确认。如果发送方有更多数据等待发送，就可以继续发送另一段数据了。

26.12　TCP的三次握手

为确保连接的建立和终止都可靠，TCP采用三次握手（3-way handshake）的方式，其中要交换3个报文。在采用三次握手建立一个连接的过程中，连接的每一方都要发送一个控制报文，为流量控制指明初始的缓冲区大小以及初始的序号。科学家们已证实，不管是否出现分组丢失、重复、延迟或重传事件[⊖]，三次握手是确保非模糊一致性的充分必要条件。此外，握手过程还可保证TCP只有在连接的两端彼此都同意时才会打开或者关闭一个连接。

TCP使用术语同步段（synchronization segment，SYN segment）来描述三次握手中用于建立连接的控制报文，使用术语终止段（finish segment，FIN segment）来描述二次握手中用于关闭连接的控制报文。图26-8说明了用于建立连接的三次握手过程。

用于建立连接的三次握手过程的一个关键是涉及序号的选择问题。TCP要求每一端产生一个随机的32位序号作为数据发送的初始序号。在计算机重新启动之后，如果某个应用进程试图建立一个新的TCP连接，TCP会选择一个新的随机数。因为所选择的随机值与前一连接所使用的序号相匹配的概率很低，所以TCP避免了"重放"问题。也就是说，一对应用程序使用TCP进行通信和关闭连接，然后建立一个新的连接。新连接上使用的序号将不同于旧连接上使用的序号，这可使TCP拒绝任何延迟到达的分组。

⊖　像其他TCP分组一样，用于三次握手的报文也可能会丢失。

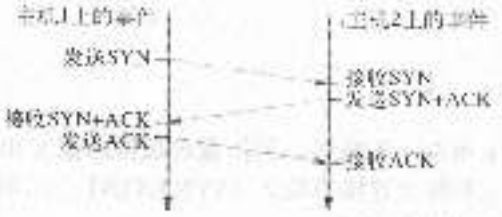


图26-8 用于建立TCP连接的三次握手过程

用于关闭一个连接的三次握手过程要用到FIN段。每个方向上发送一个带有FIN标志位的确认报文，以确保终止连接前所有数据都已到达接收端。图26-9是交换报文的示意图。

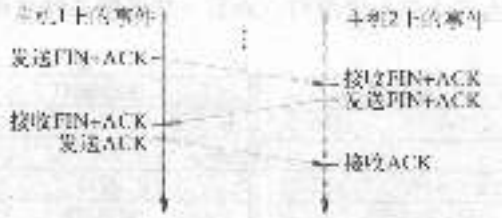


图26-9 用于关闭一个连接的三次握手过程

26.13 TCP拥塞控制

TCP最有趣的一个方面是拥塞控制（congestion control）机制。回顾一下，在因特网中，分组出现延迟或是丢失好像更多地是由于网络拥塞而非硬件故障造成的，而重传过程由于向网络注入了额外的分组副本反而加重了拥塞问题。为了避免发生拥塞崩溃，TCP利用分组延迟的变化来作为对网络拥塞的测量，通过减小重传数据的速率来应对网络拥塞。

尽管我们想到了减小传输速率，但是TCP并不去计算传输速率，而是基于缓冲区的大小来传输。也就是说，接收方通告一个窗口大小，发送方在接收到ACK之前可以发送数据填满接收窗口。为了控制数据速率，TCP可以强行限制窗口大小——通过暂时减小窗口大小，发送方TCP即可有效地降低数据速率。重要的概念是：

> 在概念上，当出现拥塞时，传输协议应该减少数据传输的速率，由于采用了可变窗口大小，TCP通过暂时减少窗口大小即可减小数据传输速率。在发生分组丢失的极端情况下，TCP会暂时将窗口大小减为当前值的一半。

在开始一个新连接或是一个报文出现丢失时，TCP会采用一种特殊的拥塞控制机制。TCP不是传输足够的数据去填充接收方的缓冲区（即接收方的窗口大小），而是以发送一个包含数据的单个报文作为开始。如果一个确认到达而没有丢失，TCP就将发送的数据量加倍，即发送两个报文。如果对应的两个确认也到达了，TCP就再发送4个报文，依此类推。这种指数增长方式一直继续下去，直到TCP正在发送的数据量等于接收方通告窗口的一半为止。当达到窗口大小一半的时候，TCP降低增长率，改为按线性增长窗口大小（只要不发生拥塞）。这种方法称为慢开始（slow start）。

TCP的拥塞控制机制对于业务量的增长能做出良好的反应。通过迅速后退的办法，TCP能够缓和拥塞。从本质上来讲，当因特网变得拥塞时，TCP是会避免增加重传的。更重要的是，如果所有TCP都遵循标准来开发，那么拥塞控制方案就意味着在发生拥塞的时候所有发送方

都会速率后退，从而避免了拥塞崩溃。

26.14　TCP段格式

TCP对所有的报文都采用单一的格式，包括载送数据的报文和载送确认信息的报文，以及载送三次握手中用于建立和终止连接的报文（SYN和FIN）。TCP使用术语段（segment）来指一个报文。如图26-10所示为段的格式。

为了理解段的格式，有必要记住：一个TCP连接上包含两个数据流，每个方向上各有一个。如果每一端的应用进程同时发送数据，TCP就可在它所发送的单个段中携带输出数据和对输入数据的确认以及窗口通告。该窗口通告指出仍可用于接收数据的缓冲区数量。因此，段中的某些域是针对前进方向上的数据流的，而另一些域则是针对相反方向上的数据流的。

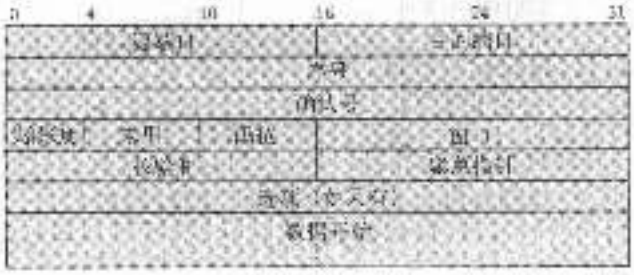


图26-10　用于数据和控制报文的TCP段的格式

当计算机发送一个段的时候，“确认号”域和“窗口”域是针对接收数据的，即“确认号”指出下一次期待接收的数据序号，“窗口”指出去掉已经确认的数据外还剩多少可用的缓冲区空间。“确认号”总是指向缺失数据（即期待接收的数据）的开始位置，如果数据段不按顺序到达，那么接收方TCP就会多次生成同样的确认，一直到缺失数据到达为止。“序号”域是针对发送数据的，它给出的是段中所携带数据的第一字节的序号，接收方利用这一序号对乱序到达的段进行排序，并利用这一序号计算确认号。“目的端口”域标识接收方计算机上的哪一个应用进程负责接收数据，“源端口”域标识发送数据的应用进程。最后，“校验和”域含有对这个TCP段头部和数据的校验和。

有关序号和确认号的关键思想是：

> TCP段中的“序号”域给出的是前进方向上段中所携带数据的第一个字节的序号，“确认号”域给出的则是相反方向上缺失数据开始位置的序号。

26.15　本章小结

传输控制协议（TCP）是TCP/IP协议组中最主要的传输协议。TCP为应用程序提供可靠的、可流控的、全双工的流传输服务。在请求TCP建立一个连接之后，应用程序即可利用这一连接发送和接收数据，TCP能确保数据按序传递而无重复。最后，当两个应用进程完成使用一个连接时，它们要请求终止该连接。

一台计算机上的TCP与另一台计算机上的TCP通过交换报文来进行通信。从一个TCP传递给另一个TCP的所有报文都采用TCP段格式，包括载送数据、确认和窗口通告的报文，以及用于建立和终止连接的报文。每个TCP段都被封装成IP数据报进行传输。

一般来讲，传输协议会采用各种机制来确保可靠的服务，TCP则采用了一种特别复杂的

技术组合，这种组合已经被证明是极其成功的。除了在每个段中提供校验和外，TCP会重传任何被丢失的报文。为了适应因特网中随时间变化的延迟，TCP的重传超时是自适应的——TCP为每个连接分别测量它当前的往返时间，然后利用往返时间的加权平均值去为重传选择一个合适的超时值。

练习题

26.1 假设两个程序之间发送的报文可能出现丢失、重复、延迟或乱序传递，请设计一个协议，能够可靠地允许两个程序间意通信。将你的设计给别人看看，看他们能否找出一个报文丢失、重复和延迟的序列来使你的协议失败。

26.2 请列出TCP的所有特点。

26.3 路由器用到协议栈中的哪些层协议？主机呢？

26.4 传输协议为实现可靠传输必须解决的主要问题是什么？

26.5 传输协议用到哪些技术？

26.6 当使用窗口大小为N的滑动窗口技术时，在不要求收到一个ACK确认的前提下可以发送多少个分组？

26.7 为什么停—走协议在2Mbit/s速率的GEO卫星信道上工作时其吞吐率特别低？

26.8 扩充图26-3中的图示，表示出16个连续分组的传输过程。

26.9 互特网上造成分组延迟或丢失的主要原因是什么？

26.10 TCP如何对付分组丢失的问题？

26.11 如果协议等候重传的时间太长，吞吐率将会怎样？如果协议等候重传的时间不够长，又会怎样？

26.12 TCP如何计算重传的超时值？

26.13 TCP窗口大小控制的是什么？

26.14 SYN指的是什么？FIN呢？

26.15 假设两个程序使用TCP来建立连接、通信、终止连接，然后又打开一个新连接。再假设用于终止第一个连接的FIN报文被重复且延迟直到第二个连接的建立，如果旧FIN报文副本这时传来了，TCP会终止这个新连接吗？为什么？

26.16 网络上出现什么问题时会促使TCP暂时减小它的窗口大小？

26.17 编写一个计算机程序，提取并打印出一个TCP段头中的域。

26.18 TCP校验和是必要的吗？它能否依靠IP校验和来确保数据的完整性？试解释。

第27章 因特网路由与路由协议

27.1 引言

前几章讲述了数据报转发的基本概念，并解释了IP如何利用转发表来为每个数据报选择下一站地址。本章要探讨网络互联技术中的一个重要课题：用于创建和更新转发表中路由信息的传播问题。这里要讨论转发表如何构建，并解释路由软件如何按需要更新转发表。

本章讨论的焦点是因特网中路由信息的传播问题，讲述所采用的几个路由更新协议，并解释内部与外部路由协议的区别。

27.2 静态与动态路由

IP路由技术可以分为两大类：

- 静态路由。
- 动态路由。

术语静态路由（static routing）描述了这样一种方法：系统在启动时即创建一个转发表，此后记录项内容保持不变（除非管理员手工更改这些记录项）。相反，术语动态路由（dynamic routing）描述的是这样一种方法：在系统中运行路由传播软件（route propagation software），并不断地更新转发表，以确保每个数据报都能沿最优路径转发。也就是说，路由软件与其他系统通信，以便“学习”去往每个目的地的最佳路径，并不断检查那些会使路径发生改变的网络故障。当系统启动时，动态路由的运行开始与静态路由极其相似，也要加载一组初始的路径信息到转发表中。

27.3 主机静态路由与默认路径

静态路由简单明了，易于指定，而且不需要额外的路由软件，它不消耗带宽，也无须花费CPU资源用于传播路由信息。但是，相对而言，静态路由不太灵活，无法适应网络故障或拓扑变化。

静态路由用在哪里呢？多数主机都使用静态路由，特别是在主机只有一个网络连接且只有一个路由器将该网络连接到因特网的情况下。例如，图27-1所示的网络结构，有4台主机接在以太网上，通过路由器R连接到因特网。

如图27-1所示，对一台典型的主机来说，静态转发表中有两个记录项就够了，其中一项指定直接连接的网络地址，另一项指定由路由器R提供的去往所有其他目的地的默认路径（default route）。当应用进程产生一个去往本地网络一台计算机（如本地打印机）的数据报时，转发表的第一项指示IP将数据报直接传递给它的目的地。当数据报发往因特网中的其他目的地时，转发表的第二项就指示IP将它发送到路由器R_1。

要点 多数因特网上的主机都采用静态路由，主机的转发表包含两项：一项指定该主机所在网络的直接路由，另一项是默认项，它将所有去往其他目的地的业务引到特定的路由器。



a)

网络	掩码	下一站
128.10.0.0	255.255.0.0	直接
默认	0.0.0.0	128.10.0.100

b)

图27-1 a)去往因特网的典型连接；b)每台主机中使用的静态转发表

27.4 动态路由与路由器

因特网中的路由器能否采用与主机同样的方法使用静态路由呢？虽然也存在路由器采用静态路由的情况，但大多数路由器都采用动态路由。为了更好地理解路由器仅采用静态路由即可工作这样一种特例，我们来看看图27-1。可以想象这个图对应的是一个小单位，它是某ISP的一个用户，通过路由器R_1离开用户站点的所有业务都必须传送到ISP（例如通过DSL连接）。这为路径总不会改变，所以路由器R_1的转发表可以是静态，而且可以使用默认路径，正如主机中的转发表那样。

尽管有少数例外，但对大多数路由器来说，使用静态路由和默认路径是不够的，这种用法仅限于图27-1所述的特定网络配置情况。当两个ISP互连时，双方需要动态地交换路由信息。为了理解这一点，考虑图27-2中所示的用两个路由器互连3个网络的情况。

每个路由器都了解它的直连网络的情况。因此，路由器R_1了解网络1和网络3，R_2了解网络2和网络3。但是，路由器R_1并不了解网络2，R_2也不了解网络1，因为它们之间没有直连关系。对于这个小例子来说，静态路由看起来似乎就已足够。但是，静态路由方法并不能扩展到处理有几千个网络的情况。特别是每次ISP要增加一个新的用户网络时，相关信息必须传遍整个因特网。更重要的是，人工处理过程太慢，不能适应因特网上的网络故障或拥塞情况。因此，为了保证所有路由器都能获得如何到达每一个可能目的地的有关信息，每个路由器都要运行路由软件，通过路径传播协议与其他路由器交换信息。当它了解到路径变化情况时，路由软件就要更新本地的转发表，而且由于路由器周期地交换信息，所以本地转发表就不断地被更新。



图27-2 一种需要动态路由的网络结构示例

举例来说，在图27-2中，路由器R_1和R_2通过网络3交换路由信息。这样，在R_2上的路由软件就要配置去往网络1的路径；在R_1上的路由软件就要配置去往网络2的路径。如果路由器R_2崩溃了，R_1中的路径传播软件就会检测到网络2不再可达，并将对应路径从它的转发表中移除。随后，当R_2恢复正常工作时，R_1的路由软件将确定网络2为重新可达，并重新配置好路径。

概括如下：

> 每个路由器都运行路由软件以获得其他路由器可到达的目的地的情况，并通知其他路由器本身可到达目的地的情况。路由软件利用收到的信息来不断地更新本地的转发表。

27.5　全球因特网的路由技术

迄今为止，我们已经讲述了最简单连通性情况下的路由问题（即仅涉及少量路由器的情况）。本节介绍更宽范围的话题：全球因特网的路由技术。本节先考虑一般原理，后续几节要解释专门的路径传播协议。

我们说过，路径传播协议能让一个路由器与别的路由器交换路由信息。但是，这种方案并不能应用于整个因特网——如果因特网上的一个路由器试图与所有其他路由器交换路由信息，那么所形成的业务量可能会压垮整个因特网。为了限制路由业务量，因特网采用了分级的路由层次结构，将因特网中的路由器和网络划分为各种群组，每个群组内的所有路由器交换信息，然后，在每个群组内至少要有一个（也可能更多）路由器总结有关的信息并发送给其他群组。

一个群组有多大？群组内的路由器采用什么路由协议？如何表示路由信息？群组之间的路由器采用什么协议？因特网路由系统的设计者既没有规定群组的准确规模大小，也没有指定正确的数据表示方法或协议，而是有目的地保持路由体系结构足够灵活，以便能对付更加广泛多样的组织结构。例如，为了适应各种不同规模的组织，设计者回避指定群组的最小或最大规模；为了适应任意的路由协议，设计者则允许每个组织可以独立地选择路由协议。

27.6　自治系统概念

为了掌握路由器群组的概念，我们使用术语自治系统（Autonomous System，AS）。直观上，人们可以把一个自治系统想象为在某个管理权限（administrative authority）控制之下的所有网络和路由器的一个集合（contiguous set）。管理权限在这里并没有准确的含义——这个术语十分灵活，可以适应很多的可能情况。例如，一个自治系统可对应到一个ISP、整个公司或一所大学的范围。另外，对于拥有多个站点的大型组织，则可以选择将每个站点定义一个自治系统。特别是，每个ISP一般都是单个自治系统，但也有可能会将一个大型ISP划分为多个自治系统。

自治系统规模的选择可能会影响到经济、技术或管理等诸多方面。例如，考虑一个跨国公司，如果将公司网络划分成多个自治系统，每个系统都连接到对应国家的一个ISP，那么这样做比起作为单个自治系统连接到因特网的做法，可能会更经济些。选择特定的自治系统规模的另一个原因来源于所使用的路由协议——一种路由协议在很多路由器上使用时，可能会产生过多的路由业务量，即路由业务量可能会随路由器数量的平方增长。

概括如下：

> 因特网被划分成一个个自治系统，一个自治系统内的路由器彼此交换路由信息，然后收集起来再传送给另一个群组（自治系统）。

27.7　两类因特网路由协议

既然我们理解了自治系统的概念，那么就可以更加确切地定义因特网路由。所有因特网路由协议归为两大类：

- 内部网关协议（Interior Gateway Protocol，IGP）。
- 外部网关协议（Exterior Gateway Protocol，EGP）。

我们先定义这两个大类，然后通过考查一组路由协议的具体例子来说明每个大类的情况。

27.7.1 内部网关协议

自治系统范围内的路由器群采用内部网关协议（IGP）来交换路由信息。有几种IGP可供使用，每个自治系统可以自由选择它自己的IGP。通常，IGP很容易安装和操作，但是每一种IGP都可能会限制自治系统的规模或路由复杂性。

27.7.2 外部网关协议

在一个自治系统内有一个路由器使用外部网关协议（EGP）与另一个自治系统内的一个路由器交换路由信息。EGP的安装和操作一般要比IGP复杂，但它提供了更大的灵活性和较小的开销（即较小的通信量）。为了节约通信量，EGP完从自治系统汇总路由信息，然后才发送给另一个自治系统。更重要的是，EGP实现了政策性约束（policy constraint），以便允许系统管理员能正确地决定哪些信息可以被传播到组织之外。

27.7.3 如何使用EGP和IGP

图27-3通过显示两个自治系统中的两个路由器来说明因特网所采用的两级路由层次结构。

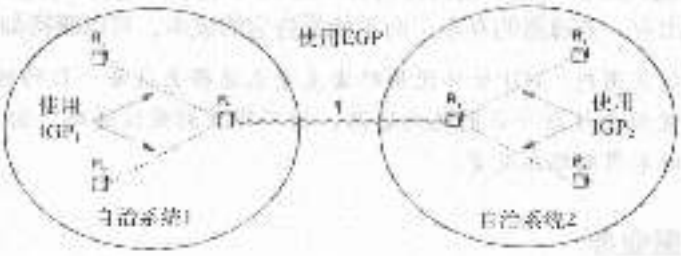


图27-3 因特网路由示意图，自治系统内部使用IGP，自治系统之间使用EGP

在图27-3中，自治系统1（AS_1）已选择了内部使用的IGP，自治系统2（AS_2）选择了IGP_2。AS_1中的所有路由器使用IGP_1进行通信，而AS_2中的所有路由器使用IGP_2进行通信。路由器R_1和R_2使用EGP在两个自治系统之间进行通信。也就是说，R_1必须汇总本自治系统内的信息并将它发送给R_4。此外，R_1还要接收来自R_4的汇总信息，并利用IGP将信息传播给AS_1中的路由器。R_4则要对AS_2完成同样的服务。

27.7.4 最佳路径，路由度量和IGP

路由软件对每个目的地似乎并不只是发现一条通路，而是找到所有可能的通路，然后选择其中最佳的一条。虽然因特网的任何一对源/目的端之间通常存在多条通路，但并不存在通用的准则来衡量哪条通路是最佳的。为了理解这是为什么，不妨考虑一下不同应用的要求。对于远地桌面应用系统来说，有最小延迟的通路是最佳的；对于浏览器下载大的图形文件来说，有最大吞吐率的路径是最佳的；而对于要接收实时话音的网站播音应用来说，有最小延迟抖动的路径则是最佳的。

我们使用术语路由度量（routing metric）来指路由软件在选择路径时对所用通路的测量。虽然吞吐率、延迟或抖动等都可能作为路由度量，但多数因特网路由软件并不这样做。典型的因特网路由采用两种度量的组合：管理成本（administrative cost）与跳程计数（hop count）。在因特网路由中，一跳程对应于一个中间网络（或路由器），所以对目的地的跳程计数就给出了去往目的地的通路上中间网络的个数。管理成本要由手工来赋值，常常用它来控制通路通信量的使用。例如，假设在某公司里财务部门与工资部门之间有两条通路连接：2跳通路上包括了指定给客户业务应用的网络，3跳通路上则包括了公司内部业务的网络。这表明，最短路

经违背了公司政策，因为它要通过指定服务于客户的网络。在这种情况下，网络管理员可以推翻这个2跳程通路的实际成本，而给该通路赋给4跳程的管理成本，即管理员要用能产生所需效果的管理成本去替换实际成本。路由软件将选择较低成本的通路，即具有3跳程度量的通路，这样才能符合公司的管理策略。

要点　虽然很多因特网路由软件被设计成使用跳程计数度量，但是网络管理员也能通过这种度量限制去实施某些策略。

在路由度量方面，IGP与EGP有一个重要的区别：IGP采用路由度量，而EGP却不采用。也就是说，每个自治系统选择好一种路由度量，并安排内部路由软件对每条路径发送度量信息。接收软件则利用度量信息来选择最佳通路。但是，在自治系统之外，EGP并不试图选择最佳通路，只是发现通路。这样做的理由很简单：因为每个自治系统可以自由选择路由度量，而EGP对此却无法进行有意义的比较。例如，假设某个自治系统报告到目的地D的通路跳程数，而另一个自治系统却是报告到D去的另一条通路的吞吐率。接收到两个报告的EGP无法选择两条通路中的那一条通路成本更低，因为没有什么方法可以将跳程度量转换为吞吐率度量。因此，EGP只能报告出有一条通路的存在，而不能报告它的成本。可以概括如下：

在自治系统范围内，IGP软件使用路由度量来选择去往每个目的地的最佳通路，EGP软件只是发现去往每个目的地的通路，但不能找到最佳通路，因为它无法比较多个自治系统所采用的路由度量。

27.8　路径与数据业务

在联网中有句俗语这样暗示：对路由通告的响应就是数据。亦即，去往指定目的地的数据业务流正好是在路由业务流的相反方向上流动。例如，假设ISP拥有的自治系统中包含网络N，在业务流去往N之前，ISP必须通告一条去往N的路径。这表明，当路由通告流出去的时候，数据即将开始流进来。如图27-4所示为与路由通告对应的数据流。



图27-4　ISP中的路由器通告一条路径后流入的数据流

27.9　边界网关协议

在因特网中已经出现一种叫做边界网关协议（Border Gateway Protocol，BGP）的特殊协议，它成为最为广泛使用的外部网关协议。它目前存在3个主要的修订版本。版本4是目前的标准版本，官方缩写为BGP-4。在实际使用过程中，这一版本号保持了很长时间都不变，因此网络专业人士用术语BGP来直接指版本4的BGP。

BGP具有如下特性：

- 自治系统之间路由。因为BGP要作为外部网关协议，所以它在自治系统层次上提供路由信息，亦即所有路径都可作为自治系统的通路。例如，到某一给定目的地的通路可能由

自治系统17、2、56和12构成。这里不使用路由度量，也不对BGP提供通路上每个自治系统中路由器的任何有关详情。

- 规定政策性条款。BGP允许发送者和接收者强加一些政策性约束，特别是，管理员可以通过配置BGP来限制哪些路径可以通告到外部去。
- 中转路由设施。如果一个自治系统同意某个业务流通过本系统转送到另一个自治系统，BGP就将本系统归类为中转系统（transit system）；或者如果不允许通过，则归类为残存系统（stub system）。类似地，通过本系统被转送到另一个自治系统去的业务流，就归类为中转业务，这种归类方法使得BGP便于区分ISP和其他自治系统。更重要的是，BGP可以让某个公司把自己的自治系统归类为残存[1]系统，即使它是多宿主的（multi-homed）系统，即具有多个外部连接的公司网络可以设置自己以拒绝接受中转业务。
- 可靠的传输。BGP的所有通信都要利用TCP。也就是说，在一个自治系统中的路由器上的BGP程序，与另一个自治系统中的路由器上的BGP程序之间形成一个TCP连接，然后通过这个连接发送数据。TCP确保数据能按正确的顺序到达且不会丢失。

BGP起到黏合剂的效果，它将因特网上的路由粘合到一起——在因特网的核心区域[2]。第1梯级（Tier-1）的ISP利用BGP来交换路由信息，并学习其他每个ISP的用户信息。概括如下：

> 边界网关协议（BGP）是外部网关协议，第1梯级的ISP利用BGP在因特网核心区域的自治系统之间交换路由信息。它的当前版本是BGP-4。

27.10 路由信息协议

路由信息协议（Routing Information Protocol，RIP）是因特网上使用的第一个内部网关协议。RIP具有如下特性：

- 自治系统范围内的路由。RIP作为一种内部网关协议而设计，用于在一个自治系统范围内的路由器之间传递信息。
- 采用跳计数度量。RIP采用网络跳程（hop）来测量距离，源与目的地之间的每个网络都计数为一个跳程。RIP将直接连接的网络计为一跳。
- 不可靠的传输。RIP使用UDP在路由器之间传输报文。
- 采用广播或组播传递。RIP本打算在局域网上应用，这种网络支持广播或组播（例如以太网）。版本1的RIP利用广播来传递报文，版本2则允许通过组播来传递报文。
- 支持CIDR和子网划分。版本2的RIP包含每个目的地址及其对应的地址掩码。
- 支持默认路径传播。除了指定明确目的地的路径外，RIP也允许路由器通告一条默认路径（default route）。
- 距离矢量算法。RIP采用算法18-3[3]定义的距离—矢量的方法来寻找路径。
- 主机被动模式。虽然只有路由器能够传播路由信息，但RIP也允许主机被动地听取和更新它的转发表。在主机可以在多个路由器中选择网络连接的网络上，被动模式的RIP很有用。

---

[1] 这里的stub本意是“残余下来的部分”，所以译为“残存”和“残桩”都是意思相当，但前者的字义却更易理解和确切。——译者注

[2] 这里所说的“核心区域”，是指因特网的主体网络部分，包括地区性的ISP网络和中转性的核心网络，而处在最外层与用户网络连接的是地区性网络，即第1梯级的ISP网络。——译者注

[3] 算法18-3可在第18章18.12.2节中找到。

为了理解RIP是怎样传播路径的，再回顾一下距离—矢量路由的工作原理。每个发出去的报文含有一个通告，其中列出了发送方能够到达的网络以及去往该网络的距离。当接收路由器收到一个通告的时候，RIP软件就利用目的地列表去更新本地转发表。RIP通告中的每个记录项由一对参数组成：

（目的网络，距离）

其中的距离是到目的地的跳数。当报文到达时，如果接收方没有路径去往所通告的目的地，或者所通告的距离短于目前使用路径的距离，接收方就利用去往发送方的路径来替换原来的路径。

RIP的主要优点是它的简单性，RIP只要求很少量的配置——管理员只需在本单位的每个路由器上启动运行RIP并允许每个路由器向其他路由器广播报文即可，一段短时间之后，本单位的所有路由器就会有去往所有目的地的路径。

RIP也处理默认路径的传播。一个单位仅需对它的一个路由器配置默认路径即可（典型作法是，选择连接到ISP去的那个路由器），RIP将默认路径传播给单位内的所有其他路由器。这样，任何发送到本单位之外的目的地的数据报都会转发给ISP。

27.11　RIP分组格式

RIP报文格式有助于说明距离—矢量路由协议是如何工作的，如图27-5所示为一个RIP更新报文。



图27-5　RIP版本2更新报文的格式

如图27-5所示，每个记录项包含目的地的IP地址及其距离。此外，为了允许RIP和CIDR或子网寻址一起使用，一个记录项还包含一个32位的子网掩码。每个记录项还包含一个下一站地址以及两个16位长的域，用于标识记录项是IP地址和提供记录项聚集时用的标签。每个记录项总共包含20字节，概括如下：

RIP是一种内部网关协议，它采用距离—矢量算法传播路由信息。

27.12　开放最短路径优先协议

RIP报文格式说明了距离—矢量协议的一些缺点：报文的大小与能到达的网络个数成正比，

发送RIP报文会引入延迟，且处理RIP报文也会消耗很多CPU周期。这种延迟意味着路径变化情况的传播速度会逐个路由器地降低。因此，虽然RIP在少数路由器的情况下工作良好，但它的扩展性却不是很好。

为了满足路由协议能扩展到大型组织范围的这一需求，IETF设计出一个称为开放最短路径优先协议（Open Shortest Path First Protocol，OSPF）的内部网关协议。OSPF这个名字得于它使用了Dijkstras的SPF算法，而这个算法是用来计算最短路径的。OSPF具有以下特性：

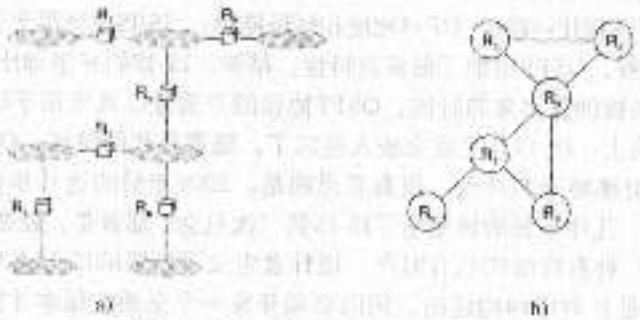
- 自治系统内的路由。OSPF是一个内部网关协议，用在自治系统内部。
- 支持CIDR。为了适应CIDR编址，OSPF包含了32位的地址及其对应的地址掩码。
- 安全的报文交换。使用OSPF的一对路由器可以验证每个交换的报文。
- 支持路径导入。OSPF允许路由器引入通过其他手段（如由BGP）学习到的路径。
- 链路—状态算法。OSPF采用了在第18章中所述的链路—状态路由（Link-State routing）算法。
- 支持度量。OSPF允许管理员对每条路径赋于成本参数值。
- 支持多址接入式网络。传统的链路—状态路由在多址接入式网络（如以太网）上效率很差，因为连接在网络上的所有路由器都会广播链路状态。OSPF通过只指定一个路由器在网络上广播的做法来优化路由算法。

概括如下：

> OSPF是一种内部网关协议，它采用链路—状态路由算法来传播路由信息，路由器利用Dijkstra的SPF算法来计算最短路径。

27.13 OSPF图的例子

回顾第18章所述，链路—状态路由使用了图论的抽象表示。虽然OSPF允许网络和图之间有复杂的关系，但用一个简单的例子有助于解释清楚它的基本概念[1]，如图27-6所示的网络和相关的图。



a) b)

图27-6 a)指示示意图，b)对应的OSPF图

图27-6表示出一个典型的OSPF图，每个节点对应于一个路由器，图中的一条边对应于一对路由器之间的一个连接（即一个网络）。为了遵循链路—状态算法，由网络连接的每对路由器要周期性地相互探测对方的状态，然后向其他路由器广播链路—状态报文。所有路由器都要接收广播报文，都要利用报文去更新它所保存的图的本地副本，当状态发生改变时还要重新计算最短路径。

[1] 实际中的OSPF比本节给出的例子更加复杂。

27.14　OSPF区域

有一个特性使OSPF变得比其他路由协议更加复杂，也使它更加强有力，即分层路由特性。为了实现层次化，OSPF允许一个自治系统为了路由目的而被分割，亦即管理员可以将自治系统内的路由器和网络划分成子集，OSPF称这种子集为区域（areas）。每个路由器经配置后会知道区域边界（即可准确地知道哪些路由器在它的区域内）。当OSPF运行起来后，给定区域内的路由器会周期性地交换链路—状态报文。

除了在区域内交换信息外，OSPF还允许区域之间的通信。每个区域有一个路由器被配置成与其他一个或多个区域内的一个路由器进行通信。这两个路由器会总结它们在各自区域内从其他路由器那里学到的路由信息，然后交换这些总结信息。因此，OSPF并不对自治系统内的所有路由器广播这些信息，而是只限于将链路—状态信息广播给一个区域内的路由器。分层处理的结果是相比其他路由协议，OSPF可应付更大的自治系统。

要点　由于OSPF允许管理员将自治系统内的路由器和网络划分成多个区域，所以相比其他IGP来说，它能够应付更多数量的路由器。

27.15　中间系统到中间系统协议

中间系统到中间系统协议[⊖]（Intermediate System to Intermediate System，IS-IS）是一种内部网关协议（IGP），最初是由数据设备公司设计的，作为DECNET V的一部分。IS-IS和OSPF是在差不多相同的时间创建的，并且两者在很多方面都类似。它们都采用链路—状态方法，且都使用Dijkstra算法来计算最短路径。此外，这两种协议都要求相邻的两个路由器周期性地检测它们之间的链路并广播状态报文。

OSPF与原始的IS-IS之间的主要差别可概述如下：

- IS-IS是一种私有协议（由数据设备公司拥有），而OSPF被作为一种开放的标准创建。所有厂商都可用。
- OSPF设计运行在IP协议上，IS-IS设计运行在CLNS上（已被淘汰的OSI协议栈的一部分）。
- OSPF设计用于传播IPv4路径（IPv4地址和地址掩码），IS-IS设计用于为OSI协议传播路径。
- 随着时间的推移，OSPF增加了很多新特性。结果，IS-IS的开销却比OSPF更小。

在这些协议最初被创建出来的时候，OSPF协议的开放性以及专用于IP的特点使它比IS-IS更加受人欢迎。事实上，IS-IS几乎完全被人遗忘了。随着年代的发展，OSPF受欢迎的程度鼓励着IETF为它增加更多额外的特性。很有意思的是，2000年后的这几年来（即这两个协议被设计出来后的10年），几件事情的转变给了IS-IS第二次机会，那就是，数据设备公司被解散了，IS-IS不再被认为是一种有价值的私有财产。设计者定义了新版的IS-IS并将它集成到了IP和因特网中。由于OSPF是针对IPv4构建的，所以必须开发一个全新的版本才能应付更多的IPv6地址。因特网上那些最大的ISP已经规模扩得很大，在这种规模中OSPF的额外开销使IS-IS具有了更大的吸引力。结果，IS-IS开始复出了。

27.16　组播路由技术

27.16.1　IP组播语义

迄今，我们已经讨论了单播路由的有关技术。也就是说，我们所涉及到的路由协议所传

⊖　它的名字源于数据设备公司的术语习惯，其中路由器叫做中间系统，主机叫做端系统。

播的路由信息，都是针对具有静态地址且位置不改变的那些目的地的。单播路径信息传播的一个设计目的就是稳定性——路径的不断改变是不希望有的，因为这将导致较严重的抖动及数据报乱序到达。因此，一旦单播路由协议发现了最短路径，通常都会保持这个路径直到由于故障而使得它不可用为止。

组播路由（multicast routing）信息的传播与单播路由信息的传播有很大的差别。存在这种差别的原因是由于因特网组播允许有动态组播成员关系以及存在匿名发送者。动态的组播成员关系意味着某个应用进程可以随时参与到一个群组中，并保持参与者角色任意长时间。也就是说，IP组播机制要允许在任何计算机上运行的应用进程能够：

- 随时加入到组播群组中并开始接收发给这个群组的所有分组的副本。为了加入一个群组，主机需要通知邻近的路由器。如果同一台主机上的多个应用进程决定加入一个群组，主机要接收发送给群组的每个数据报的副本，并由该主机给每个进程产生一个本地副本。
- 随时离开组播群组。主机周期性地发送群组成员关系报文给本地路由器。一旦该主机上的最后一个进程离开了这个组播群组，该主机要通知本地路由器不再参与到群组中了。

一个IP组播群组以两种方式表明它是匿名的。第一，发送方和接收方都不知道（或者找不到）组成员的身份或数目。第二，路由器和主机都不知道哪个应用进程要发送数据报给组播群组，因为任意一个进程随时都可以发送数据报给任意一个组播群组。这表明，组播群组的成员关系只定义了一组接收者。而发送者在发送报文给群组之前不需要加入到组播群组中。

概括如下：

> IP组播群组的成员关系构成是动态的，即计算机可以随时加入或离开一个群组。群组成员关系只是定义了一组接收者；任意一个应用进程（即使它不是一个群组成员）都能发送数据报给组播群组。

27.16.2 因特网群组管理协议

一台主机如何加入或离开组播群组呢？有一个现存的标准协议负责这件事。每当主机需要加入或离开某个特定的组播群组时，协议允许该主机去通知邻近的路由器。这个协议就是因特网群组管理协议（Internet Group Management Protocol，IGMP），它只用在主机与路由器之间的网络上，而且协议只把主机（不是应用进程）定义为群组成员，对应用进程没有做任何规定。如果在一台指定主机上有多个应用进程加入一个组播群组，主机必须把接收到的每个数据报复制成多个副本传给每个本地应用进程。当主机上最后一个应用进程离开群组时，主机会利用IGMP通知本地的路由器它不再是群组的成员了。

27.16.3 转发与发现技术

当路由器得知它连接的一个网络中的某台主机已经加入到一个组播群组时，它必须建立一条通往该群组的通路，并把从该群组接收到的数据报传送给主机。因此，路由器（而不是主机）有责任去传播组播路由信息。

动态的群组成员关系以及对匿名发送者的支持，使实现通用的组播路由变得极为困难。而且，群组规模和拓扑结构在不同应用中也有相当大的变化。例如，远程会议通常创建一个小的群组（如2—5个成员），成员可能分散在不同的地理位置，或者在同一个单位内，而网络广播应用则可能要创建一个拥有几百万成员且成员分布全球的大型群组。

为了适应动态的成员关系，组播路由协议必须能迅速地、不断地改变路径。例如，如果在法国的一个用户要加入到在美国和日本也有成员的组播群组中，组播路由软件必须首先找到群组的其他成员，然后创建一个最佳的转发路径结构。更重要的是，当为任意一个用户都

可能发送数据报给群组，所以路径信息必须扩展到群组成员以外。在实际使用中，组播协议采用下列3种不同的方法来转发数据报：

- 扩散与剪枝（flood-and-prune），
- 配置与隧道（configuration-and-tunnels），
- 基于核心的发现。

扩散与剪枝。在群组较小且所有成员都连接在邻近的局域网（如在一个公司内的群组）的情况下，采用扩散与剪枝方法比较理想。在起始阶段，路由器把每个数据报转发到所有网络，也就是说，当一个组播数据报到达时，路由器利用硬件组播能力将它发送给所有直连的LAN。为了避免形成路径环，扩散与剪枝协议采用一种称为逆向通路广播（Reverse Path Broadcasting，RPB）的技术来防止循环。在扩散期间，路由器之间交换有关群组成员关系的信息。如果某个路由器获悉在指定网络中没有主机是群组的成员，它就停止向该网络转发组播数据报（即将该网络从群组中"剪去"）。

配置与隧道。在群组成员的地理位置很分散（即每个站点内很少成员而且站点之间相距很远）的情况下，采用配置与隧道的方法比较理想。每个站点的路由器通过配置知道其他站点的情况。当一个组播数据报到达时，每个站点内的路由器利用硬件组播能力将它发送给所有直连的LAN，然后再查阅它的配置表以确定那些远地站点应该接收这个数据报的副本。路由器采用"IP-in-IP"隧道方法将组播数据报的副本传送给每一个远地站点。

基于核心的发现。虽然扩散与剪枝和配置与隧道各自能较好地处理两种极端的情况，但还是需要一种新的组播技术，它允许组播处理的群组范围能从那些成员分布在一个区域的小群组扩展到那些成员遍及四处的大群组。为了提供平滑增长的机制，有些组播路由协议为每个组播群组指定一个核心（core）单播地址。每当路由器R_1收到一个必须传输给某个群组的组播数据报时，它就将这个组播数据报封装在一个单播数据报中，并将它转发到群组的核心单播地址。在单播数据报通过因特网传输的过程中，每个路由器都要检查数据报的内容。当数据报到达另一个已参与该群组的路由器R_2时，它会剥除外面的封装并对里面的组播报文进行处理。R_2使用组播路由将数据报转发给本群组内的成员。加入群组的请求报文遵循同样的处理模式——如果R_3收到一个想加入群组的请求，它会添加一条新的路径到其组播转发表中并开始向R_1转发每个组播数据报的副本。因此，接收特定组播群组报文的路由器集合就由核心向外进行增长。用图论的术语来说，这些路由器形成了一棵树（tree）。

27.16.4　组播协议

虽然已经提出了许多组播路由协议，但目前还不存在因特网范围内的组播路由。所提出的一些协议有：

距离矢量组播路由协议（Distance Vector Multicast Routing Protocol，DVMRP）。这是UNIX程序mrouted以及因特网组播骨干网（Multicast Backbone，MBONE）所采用的协议。DVMRP执行本地组播，并采用"IP in IP"封装方法将组播数据报从因特网的一个站点发送到另一个站点。有关MBONE更多的信息可在以下网址找到：

http://www.lbl.gov/web/Computers-and-Networks.html#MBONE

基于核心的树型算法（Core Based Trees，CBT）。这是一种特殊的协议，在这种协议中路由器从中心点出发为每个群组构建一棵传递树，CBT依靠单播路由到达中心点。

协议无关的组播—稀疏模式（Protocol Independent Multicast-Sparse Mode，PIM-SM）。这个协议采用与CBT一样的方法来形成组播路由树。设计者选择术语"协议无关"是要强调：

虽然在建立组播转发路径时采用单播数据报来联系远端目的地，但PIM-SM并不取决于任何特殊的单播路由协议。

协议无关的组播—密集模式（Protocol Independent Multicast-Dense Mode，PIM-DM）。这是设计用于一个组织内的协议，采用PIM-DM广播（即扩散）的路由器将组播分组发送给组织内的所有位置，而没有特定群组成员的路由器则要回送一个报文（即停止发送分组流的请求），以便剪去它这条组播路由树枝。这个方案对于短期（如几分钟）的组播会话很有效，因为它不要求在传输开始之前建立路由树。

对开放最短通路优先协议的组播扩展（Multicast Extensions to the Open Shortest Path First Protocol，MOSPF）。MOSPF并非是一个用于通用目的的组播路由协议，而是设计用于在一个组织内的路由器之间传送组播路径。因此，MOSPF没有采用通用目的的组播做法，而是构建在OSPF的基础上且使用了LSR设施。

图27-7总结了前面讲述的几种组播路由协议。

协　议	类　型
DVMRP	扩散与剪枝
CBT	基于核心的发现
PIM-SM	基于核心的发现
PIM-DM	扩散与剪枝
MOSPF	链路状态（在一个组织范围内）

图27-7 组播路由协议及每种协议使用的方法

尽管人们进行了20年的研究且做了很多实验，但通用的因特网组播还是没有取得成功。即使是（多种协议的）协同应用也没有起到足够的刺激作用，其结果可概括如下：

> 因特网组播的动态特性使它的组播路径传播问题变得很困难，虽然已经提出了很多协议，但是目前因特网还没有全网范围内的组播路由设施。

27.17 本章小结

大多数主机采用静态路由，在系统启动时就对转发表进行初始化；路由器则采用动态路由，路径传递软件不断地更新转发表。根据使用的路由技术，因特网被划分成一个个的自治系统。用于在自治系统之间传递路径信息的协议叫做外部网关协议（EGP），用于在自治系统内部传递路径信息的协议叫做内部网关协议（IGP）。

边界网关协议（BGP）是因特网上主要的EGP，第一梯级ISP利用BGP彼此告知其用户的信息。IGP包括RIP，OSPF和IS-IS。

因为因特网组播允许存在动态的群组成员关系，并允许任意一个非群组成员的源端向群组发送分组，所以组播路径传播问题变得很困难。虽然已经提出了几个组播路由协议，但目前尚无全因特网范围内的组播技术。

练习题

27.1 列出因特网路由技术的两个大类，并分别解释。

27.2 一台典型的主机的转发表中需要哪两种记录项？

27.3 假设因特网中的所有路由器都含有一个默认路径。请表示出：必然存在一个路径环。

27.4 什么是自治系统？

27.5 列出两类因特网路由协议并解释。

27.6 假设一个组织内的一台路由器使用某种路由协议宣告一个指定的目的地有10跳远，而这个目的地实际只有3跳远。这种声明是否一定有错？试解释。

27.7 当路由器向一个指定的目的地通告路径时，期望得到的结果是什么？

27.8 列出并解释BGP的特征。

27.9 BGP用在什么地方？

27.10 RIP采用哪种路由算法？它用在什么地方？

27.11 列出RIP的特征。

27.12 当路由器收到一个RIP报文时，路由器如何将每一个IP地址分割为前缀和后缀呢？

27.13 编写一个计算机程序，读出一个RIP更新报文并打印出每个域的内容。

27.14 RIP限制距离最大为16跳程。请策划一个公司内部网的例子，它包括16个以上的路由器和网络，但还能够使用RIP。

27.15 列出OSPF的特征。

27.16 OSPF中的"开放"指的是什么意思？

27.17 为什么OSPF有多个区域？

27.18 OSPF和IS-IS协议中，哪一个有较小的开销？哪一个有更多的特性？

27.19 IGMP的主要用途是什么？它用在什么地方？

27.20 转发组播数据报的3种主要方法是什么？

27.21 假设你和两个朋友在相距较远的3个学院，想利用IP组播参与到一次三方会议中。采用哪个组播路由协议最合适？为什么？

27.22 虽然每个IP组播群组都需要一个唯一的IP组播地址，如果采用一个中心服务器来分配唯一地址的话，会造成中心瓶颈。请设计一种方案，能允许一组计算机随机选择组播地址，并且能解决可能出现的地址冲突问题。

27.23 由扩散与剪枝法所产生的业务量限制了采用这种协议的网络区域的规模。如果G个组播群组每个按每秒产生P个分组的速率产生业务量，每个分组包含B比特，内部网由N个网络组成，每个网络至少有一个对每个群组的倾听者。请估计一下这个内部网的总业务量。

27.24 因特网上是否已广泛地施行了组播？试解释。

27.25 哪种组播协议允许在它建立路径前发送组播报文？

第五部分

其他网络概念与技术

网络性能、安全、管理与新技术

第28章 网络性能

28.1 引言

前面章节研究了数据通信系统的基本特性，并讨论了信号、频率、带宽、信道编码和数据传输之间的关系，并解释了底层数据传输系统的度量，讨论了数据网络的规模，以及各种联网技术可以被归类为PAN、LAN、MAN或者WAN。

本章接着讨论网络性能这个话题，讨论网络的定量度量，并解释协议和分组转发技术如何实现为某些通信流提供优先权的机制。

28.2 性能度量

我们可以非正式地使用术语速度（speed）来描述网络性能，并分为低速（low-speed）或高速（high-speed）网络。然而，这种定义是不恰当的，因为网络技术的变化如此之快，以至于称为"高速"的网络，在很短的时间（如3或4年）就可能被视为中速或低速的网络。因此，科学家和工程师使用正式的、定量的量度来精确地描述网络的性能，而不是只停留在定性的描述。在回顾了基本的量度后，我们将解释它们是如何用于实现分层服务的。尽管初学者通常倾向于非正式的描述，但定量量度还是很重要的，因为定量量度使人们能够确切地比较两个网络的特征，并建立为某些传输提供更高优先权的机制。图28-1列出了网络性能的主要量度，在后面章节中将对每种量度进行说明。

量　度	描　述
延迟（时延）	通过网络传输数据所需的时间
吞吐率（容量）	每单位时间可以传输的数据量
抖动（变化量）	网络延迟的变化量及持续的时间

图28-1 数据网络性能的关键量度

28.3 延迟

能够定量量度网络的第一个特性是延迟（latency），也称作时延（delay）。网络的延迟指定了通过网络在计算机之间传送一位数据需要花费多少时间，它用几分之一秒来量度，跨越国际网的延迟取决于底层基础设施以及参与通信的两台计算机的位置。虽然用户只关心网络的总延迟，但工程人员仍然作出更加精确的测量，因此，工程人员通常指明最大延迟和平均延迟，并把延迟分为几个部分，如图28-2所示。

传播延迟（propagation delay）。网络中有些延迟是由于信号通过传输介质传输时需要少量时间。一般情况下，传播迟延与所传播的距离成正比，即便使用长电缆运行，用于一栋大楼内的典型局域网也只有不到1ms的传播延迟。虽然这点延迟对人类而言看似无关紧要，但现代计算机在1ms内却可执行10万条指令。因此当一组计算机之间要进行协同时，1ms的时延就很重要了（例如，在金融业，投票命令到达的确切时间就决定了接受命令的顺序）。使用地球

同步轨道卫星的网络具有更高的迟延——即使以光速传输，一个码位从地球传到卫星再传回地球，也需要几百毫秒的时间。

类 型	解 释
传播延迟	信号在介质中传播所需要的时间
接入延迟	接入传输介质（如电缆）所需要的时间
交换延迟	转发分组所需要的时间
排队延迟	分组在交换机或路由器的存储器中等待选择传输所花的时间
服务器延迟	服务器从接收请求到发送响应所需的时间

图28-2 各类延迟及其说明

接入延迟（access delay）。很多网络都使用共享介质，一组共享相同介质的计算机必须竞争才能接入介质（例如，Wi-Fi无线网络使用CSMA/CA方法接入介质），这种因接入介质而引入的延迟被称为接入延迟。接入延迟取决于竞争接入的站点数量以及每个站点发送的通信量，一般较小而且是固定的，除非介质超载了。

交换延迟（switching delay）。网络中的电子设备（如二层交换机或路由器）在通过输出接口发送分组前必须为每个分组计算下一跳路径，往往要涉及查表，这意味着要访问存储器。在有些设备中，还需要额外的时间花在内部的通信机制（如总线或交换矩阵）中发送分组。计算下一跳路径并开始传输分组所需要的时间，被称为交换延迟。使用快速的CPU和专用硬件，会使得交换延迟很小而成为计算机网络中最无关紧要的延迟。

排队延迟（queuing delay）。在分组交换中使用存储/转发模式，这意味着网络设备（如路由器）要收集分组的全部位元并放在存储器中，选择下一跳路径，然后等待，直到该分组可以发送时才能开始传输。这一段时延被称为排队延迟。在最简单的例子中，分组是存放在一个FIFO输出队列中的，然后该分组只需进行等待，直到先前到达的分组已经发送完；更复杂的系统则要实现某种选择算法，以便为一些分组指定优先权。排队延迟是变化的——队列的长短完全取决于最近到达的通信量。排队延迟占据因特网中的大部分延迟。当排队延迟变得很大时，我们就说网络出现拥塞了。

服务器延迟（server delay）。虽然服务器不是网络的一部分，但对大部分通信来说它是很重要的。服务器检查请求、进行计算并发送一个响应，这个过程所花费的时间构成总延迟的重要部分。服务器对传入的请求进行排队，这意味着服务器延迟是变化的，并取决于当前的负荷。在很多情况下，用户所感觉到的因特网延迟主要是来自服务器延迟而不是网络延迟。

28.4 吞吐率、容量、实际吞吐量

能够定量度量的网络第二个基本特性是网络的容量，通常表示为网络可以支持的最大吞吐率（throughput）。吞吐率是对数据能够通过网络传输的速率衡量，单位以位/秒（bits per second，bit/s）表示。大部分数据通信网络提供超过1 Mbit/s的吞吐率，而最高速度的网络以超过1 Gbit/s的速率运行。然而，正如我们所看到的，网络中会出现吞吐率少于1 Kbit/s的特殊情况。

由于吞吐率可用几种方法进行测量，所以要小心地指明到底是测量什么。下面是几种可能：

- 单条信道的容量。
- 所有信道的总容量。
- 底层硬件的理论容量。

- 一个应用所达到的有效的数据率（实际吞吐量）。

供应商通常是宣传他们设备的理论容量和在最佳条件下所达到的吞吐率。硬件容量通常看作是潜在吞吐率的近似值，因为该容量是关于性能方面的一个上界——用户发送数据的速率不可能比该硬件传输比特的速率还快。

用户并不关心底层硬件的能力，他们感兴趣的是通过网络能够传输数据的速率。用户通常通过测量单位时间传输的数据量来评估应用达到的有效数据速率（effective data rate）。术语实际吞吐量（goodput）有时就是用来描述这种测量的。实际吞吐量比硬件的容量要小，因为协议会导致额外开销——由于协议存在如下的开销，有些网络容量是不给用户使用的：

- 发送分组的头部、尾部以及控制信息。
- 限制窗口的大小（接收缓冲区）。
- 用于解释名称和地址的协议。
- 使用握手协议来启动和结束通信。
- 当检测到拥塞时要降低传输速率。
- 重传丢失的分组。

使用实际吞吐量作为网络容量量度的缺点，在于额外开销的量是不确定的，而且取决于所使用的协议栈。除了传输协议、因特网协议和第二层协议外，实际吞吐量还取决于应用协议。例如，考虑使用文件传输协议（FTP）来测量以太网的实际吞吐量。FTP使用TCP，而TCP使用IP。此外，FTP在传输前没有压缩数据。事实上，FTP把用户数据放在TCP段中，TCP把每个段封装到IP数据报中，而IP把每个数据报又封装到以太网的帧中。因此，每个帧包含一个以太网的头部信息和CRC域、一个IP数据报的头部信息以及一个TCP头部信息。如果用户选择另一种可选的文件传输应用，或者使用另一种可替代的协议栈，那么实际吞吐量会有所改变。

要点　尽管实际吞吐量提供了测量一个数据可以在网络中传输的有效速率的方法，但是它最终还要取决于应用。

28.5　理解吞吐率与延迟

在实际中，网络专业人员用于描述网络吞吐率或网络容量的术语往往会混淆。例如，数据通信的备章定义了信道的带宽，并解释了硬件带宽与最大数据速率之间的关系。遗憾的是，网络专业人员常常使用术语带宽（bandwidth）和速度（speed）作为吞吐率的同义词。因此，我们可能会听到某人说，某个网络具有“1 Gbit/s的速度”。另外一种说法是，一些广告中使用短语“1 Gbit/s的带宽”，工程人员在试图区分“带宽”的两种用法时，约定带宽表示模拟带宽（analog bandwidth），而把术语数字带宽（digital bandwidth）表示吞吐率（throughput）的同义词，尽管这种说法是常见的，但它们仍可能会造成混淆，因为吞吐率、时延和带宽具有各自不同的特性。

事实上，吞吐率是对网络容量的度量，而非速度。为理解这种关系，可以想象一个网络就好比连接两个地点的一条公路，沿网络传送的分组就好比沿公路行驶的汽车。吞吐率决定了每秒会有多少辆汽车驶进公路，而传播延迟决定了每辆车从一个地方到另一个地方所花的时间。例如，公路每5s可容纳一辆车，即吞吐率为每秒0.2辆，如果这辆车行驶完全程要30s，则公路的延迟为30s。现在考虑如果两地之间有了第二条通道（即容量加倍），那么每5s就可允许两辆车通行，即吞吐率为每秒0.4辆。然而延迟不变，仍为30s，因为每辆车仍需行驶完全

量。因此，当我们考虑对网络的度量时，要记住：

> 网络传播延迟用秒来度量，它表示一个码位在网络中被传输需要多长时间。网络吞吐率用位/秒来度量，它表示单位时间内有多少码位的数据可进入网络。吞吐率是对网络容量的度量。

网络专业人员有个有趣的格言：

> 你总可以买到更高的吞吐率，却不能买到更低的延迟。

将网络比喻成公路，有助于理解上面的格言：给公路增加更多的车道，可以增加单位时间内进入公路的汽车数量，但不能减少行驶完全程所需要的时间。网络遵循同样的模式：给网络增加更多的并行传输路径，可以增加网络的吞吐率，但不能减少取决于所跨越区域距离的传播延迟。

28.6 抖动

当网络用于传输实时的话音和视频的时候，网络的第三种度量指标变得日益重要，该度量度指标被称为网络的抖动（jitter），它用于评估延迟的变化量。两个网络可以有相同的平均延迟，却有不同的抖动值。特别地，如果通过一个特定网络的所有分组具有完全相同的时延D，则该网络没有抖动。然而，如果分组的延迟交替地在$D+\varepsilon$与$D-\varepsilon$之间变化，那么网络具有相同的平均延迟，但有一个非零值的抖动。

为了理解抖动的重要性，我们考虑在网络上传送话音的情况。在发送端，对模拟信号进行采样和数字化，每125μs发送一个8bit的数字值。这些采样码被装配成分组或信元，然后通过网络传送出去。在接收端，抽取出数字值并转换回原来的模拟信号输出。如果网络延迟没有抖动（即每个分组或信元都以完全相同的时间通过网络），那么输出的音频信号就与原来的信号完全相符。否则，输出就会失真。处理抖动有两种通用的方法：

- 设计一个无抖动的等时网络。
- 采用能补偿抖动的协议。

传统的电话系统使用第一种方法：电话系统实现一个等时网络，以确保数字数据沿着所有路径传送的时延都是相同的。因此，如果从话机发送出来的数字数据要在两条通路上传输，就要适当地配置好相应的硬件设备使这两条通路的延迟完全相同。

在因特网上传输话音或视频，使用第二种方法：虽然基础网络可能有明显的抖动，话音和视频应用则要依靠实时协议（Real-Time Protocols，RTP）来补偿抖动[⊖]。因为使用RTP协议的费用比构造一个等时网络的费用要低得多，所以电话公司正在放宽话音与视频业务中对等时性的严格要求。当然，协议不能补偿任意的抖动——如果时延的变化量过多，输出将会受到影响。因此，即使使用第二种方法，服务提供商仍试图使网络的抖动最小化。

28.7 延迟与吞吐率的关系

理论上，网络的延迟与吞吐率是独立的，但实际上它们之间也可能会有关。为了理解这一点，还是考虑上述用公路做比拟的情况。如果汽车以等时间隔驶进公路，则同速的汽车在路上是等间距的。如果某辆汽车速度变慢，它后面的汽车也会随之变慢，会引起暂时的交通拥塞。在拥塞的公路上行驶的汽车的延迟，显然要比在不拥塞的公路上行驶的延迟大。在网

⊖ 下一章讨论在因特网上的实时数据传输。

络中也会发生与此相似的情况，如果路由器中有一个分组等待队列，当新的分组传来的时候，就被放置在队尾并等候交换机发送完它前面的分组。与严重的交通拥塞相类似，网络中过多流量也会导致拥塞。显然，进入拥塞网络中的数据会比在空闲网络中所经受的延迟要长。

28.7.1　以利用率作为延迟估值

计算机专家已经研究了延迟与拥塞的关系，发现在许多情况下，期望的延迟可由当前所用网络容量的百分数来估计。若用D_0表示网络空闲时的延迟，U是0到1之间的一个数值，表示当前的网络使用率（utilization）。那么，有效延迟D可由下面的简单公式给出：

$$D = \frac{D_0}{(1-U)} \tag{28.1}$$

当网络完全空闲时，U等于零，有效延迟即等于D_0；当网络工作在1/2容量时，有效延迟将加倍。随着网络流量接近于网络容量时（即U趋于1时），延迟将趋于无穷大。虽然这个公式只估计了变化的趋势，但我们可以得出结论：

> 吞吐率与延迟并不完全独立。随着计算机网络流量的增加，延迟将随之增加；当网络流量接近于吞吐容量的100%时，网络将会经受严重的延迟。

在实际中，网络管理员也能理解到太高的使用率会产生灾难性的延迟，所以多数管理员都会使网络使用率保持在较低水平，使网络中测量到的流量保持稳定，当平均或峰值使用率开始上升接近预设的门限值时，管理员就增加网络的容量。例如，在100Mbit/s以太网上如果使用率增加时，管理员可能会选用吉比特的以太网来代替它。管理员也可能会选择另一个办法，把网络分成两部分：将半数的计算机连接在一个网络上，将另一半数计算机连接在另一个网络上（这样的划分用VLAN交换机很容易做到）。

使用率的门限应该设置多高呢？没有单一的答案，很多管理员都选择一个保守的值。例如，运行大型骨干网的某些重要ISP会把所有数字线路的使用率保持在50%以下，而其他ISP则会将门限值设置为80%，以便节省经费。在任何情况下，管理员一般都同意网络工作容量不应该在90%以上。

28.7.2　迟延—吞吐率乘积

一旦知道了网络的延迟与吞吐率，就可以计算另一个有趣的量值：延迟—吞吐率乘积（delay-throughput product）[⊖]。为了理解延迟—吞吐率乘积的含义，再想想用公路做比拟的例子：如果汽车以每秒T辆的固定速率驶入公路，并且每辆车驶完全程需D秒，那么当第一辆车行驶完全程时，已有$T \times D$辆汽车驶进公路，因此任何时候在公路上都有$T \times D$辆车。把这种情况应用于网络，在任意时刻，正在网络中传输的码位数是：

$$\text{网络中存在的码位数} = D \times T \tag{28.2}$$

其中：D是以s为单位的延迟，T是以bit/s为单位的吞吐率。

概括：

> 延迟与吞吐率的乘积表示网络中可容纳的数据量。任何时候，在吞吐率为T，延迟为D的网络中都有$T \times D$个码位的数据正在传输。

对于延迟特别长或者吞吐率特别大的任何网络，延迟—吞吐率乘积这个指标很重要，它意味着：一台计算机向网络发出的第一个码位到达目的地之前，会产生非常大的数据量。

⊖ 当做为流量硬件的量度时，延迟—吞吐率乘积通常称为延迟—带宽乘积（delay-bandwidth product）。

28.8 测量延迟、吞吐率与抖动

用于测量吞吐率和抖动的技术是相对简单的。为了评估吞吐率，发送者传输大量数据。接收者记录从数据开始部分到所有数据都到达所用的时间，然后计算单位时间发送的数据量，作为吞吐率。用于测量抖动的技术称为分包队列（packet train）：发送者发送一系列的分组，每个分组之间保持一个小的且固定的延迟间隔。通常，系列中的分组是背靠背发送的。接收者记录每个分组到达的时间，然后用该时间系列来计算延迟中的差异。

与吞吐率或抖动的测量不同，精确测量从主机A到主机B路径上的延迟，要求两台主机都有同步时钟。此外，要测量短距离（例如，一个LAN）的延迟，时钟必须非常精确。很多网络测量工具不是使用同步时钟，而是选择一个更简单的方法：测量往返的时间，然后除以2。例如，可以使用“ping”命令。

测量网络性能是非常困难的，原因有以下4个：

- 路径可能不对称。
- 网络条件急剧改变。
- 测量可能影响性能。
- 业务是突发的。

第一点解释了为什么可能无法使用往返时间来作为延迟的近似值。非对称的路径意味着沿着从B到A的路径传输的延迟与沿着从A到B路径传输的延迟有很大的不同。因此，往返时间的一半可能不是延迟的准确测量值。

第二点解释了为什么网络性能的准确测量是很难得到的：网络条件迅速改变。例如，考虑一个共享的网络。如果只有一个主机发送数据，该主机将享有低延迟、高吞吐量和低抖动。随着其他主机开始使用网络，网络的利用率提高了，同时也将会导致延迟和抖动增加，以及吞吐量降低。此外，由于网络条件改变得很快，延迟在一秒钟之内都可能变化很大。因此，即使每十秒测量一次，测量结果也可能错过性能中的一个重大转变。

第三点指出，发送用来测量网络的测试通信量本身又可能会影响网络的性能。例如，在互联网计划（PlanetLab）研究试验中，有太多的研究人员使用“ping”来测量网络性能，导致“ping”的通信量完全支配了其他的通信量。这种情况变得非常严重，以至于管理人员专门制定策略来阻止“ping”的使用。

第四点是根本的：数据网络表现为突发（bursty）行为，这意味着流量是不均匀的。如果我们考虑某个特定主机发送的流量，突发模式是显而易见的——大多数主机保持静止，直到有用户运行一个通过互特网进行通信的应用。当用户在Web浏览器中输入一个URL，浏览器获取网页的各部分内容，然后停止通信，直到该用户请求另一个网页。类似地，如果用户下载电子邮件，主机与电子邮件系统通信，并下载用户邮箱的一个副本，然后等待用户的操作。

有趣的是，汇聚的数据通信业务也是突发性的。有人可能预期突发性是一个局部现象，当来自数百万互特网用户的通信业务汇聚在一起时，其结果将会是一个平滑的使用模式。毕竟，不可能所有的用户在完全相同的时刻阅读电子邮件；因此，当一个用户在下载时，另一个用户可能正在阅读之前下载的电子邮件。事实上，电话网络的测量表明，来自数百万用户的电话通信业务能够平滑地汇聚在一起。然而，当来自数百万互特网用户的通信业务结合在一起时，结果却不是平滑的汇聚。相反，汇聚的业务是突发性的，从这个意义上来说，总的流量有高峰和低点。事实上，统计学家说，数据业务是自相似（self similar）的。这意味着总体流量与部分（fractal）流量的情况是相似的，相同的统计资料在任何粒度中都是明显的。因此，如果一个企业检测一个LAN，本地主机的业务将会显示出突发性。如果一个中型的ISP测

量来自一千个用户的流量或者一个大的ISP测量来自一千万个用户的流量，这种流量将有大的绝对数量，但该流量将展示出与LAN中统计的流量同样的整体统计模式。

我们可以概括如下：

> 与语音电话业务不同，数据业务是突发性的。数据业务又被称为是自相似的，因为汇聚在一起的数据流量表现出与个体或部分流量同样的突发性模式。

28.9 被动测量、小分组及网流监测

测量网络的网络管理员区分两种形式的测量方法：

- 主动的，
- 被动的。

我们已经讨论过主动（active）测量技术的缺点：通过把业务注入网络，这种测量的流量会改变网络的性能。另一种可替代的方法是被动（passive）测量，该方法监控网络并计算分组的数量，但并不注入额外的流量。例如，ISP可以计算在给定的时间段中通过链路传输的字节数，从而产生该链路利用率的一个估计值。也就是说，ISP安排一个被动的监控站，在一个时间间隔内观测网络，并累计所有分组中的总字节数。

有趣的是，ISP可能会选择测量所发送的分组的数量以及数据码位的数量。为了理解其中的原因，通过观察发现，链路的利用率用容量的百分比量度，而容量是用每秒的码位数量度，ISP需要测量单位时间发送的总的数据码位。然而，交换机和路由器的容量是用每秒的分组数来量度的，这是因为路由器或交换机对每个分组执行一次下一跳转发，所花费的计算工作量与处理的分组数量成正比，而不是与一个分组的码位数量成正比。如果数据流以1 Gbit/s的速度到达，而且如果该数据流是分成几个大的分组而不是分成很多个小的分组，那么交换机或路由器执行转发的工作量将会明显减少。联网设备提供商理解这个原理，一些提供商用数据速率而不是分组速率来宣传他们产品的性能（即用大的分组来量度他们的产品性能）。

我们可以概括如下：

> 为了评估链路的利用率，ISP测量单位时间内通过链路传输的数据总量；为了评估在路由器或交换机中的影响，ISP测量单位时间内传输的分组数量。

其中，一个最广泛使用的被动测量技术，最初由思科发明而现在作为IETF标准的是NetFlow（网流监测）。实现NetFlow的路由器根据网络管理员建立的参数（例如，每一千个分组采样一个）统计地采样分组。信息从每个采样分组的头部提取出来，然后进行归纳，并把生成的概要发送到网络管理系统进行处理（通常，数据是保存在磁盘中，以便在后面进行分析）。通常，NetFlow提取源IP地址和目的IP地址、数据报的类型以及协议端口号。为了确保是被动的，运行NetFlow的路由器必须通过一个特殊的管理端口发送NetFlow概要信息，而不是将该信息路由到处理用户数据的网络上传输。

28.10 服务质量

与网络测量相对应的是网络务备（network provisioning）——设计一个网络来提供特定等级的服务。本章后面讨论可以用来实现服务质量保证的机制。概括地说，其主题就是服务质量（Quality of Service，QoS）。

为了理解QoS，考虑服务提供商与用户之间的契约。最简单的契约就是要定义出由提供商

保证数据速率（例如，提供商保证提供到因特网的DSL连接要达到2.2 Mbit/s的数据速率）的服务。更复杂的契约则要定义层级服务（tiered service），即所接受的服务等级取决于支付的金额。例如，提供商可能会选择一种具有优先权（priority）的方法，这可以确保来自订购白金级服务的用户分组比来自订购银级服务的用户分组具有更高的优先权。

大企业用户通常需要更严格的服务保证（service guarantees）。金融业通常建立的服务契约中，包括对具体地点之间延迟范围的规定。例如，经纪公司可能需要一个服务契约，规定分组从公司的总办事处传输到纽约证券交易所的时间不能超过10ms；某个公司每晚都要备份整个数据中心，它可能需要一个服务契约来保证有一条吞吐率不少于1 Gbit/s的TCP连接用于数据备份。

28.11 细粒度与粗粒度QoS

提供商怎样规定QoS保证，并使用什么技术来实施QoS？图28-3列出了为服务规范而提出的两种一般方法。正如图28-3中所示，两种方法的区别在于其粒度粗细以及是否可由提供商或客户选择参数。

方　法	描　述
细粒度	提供商允许客户为特定的通信实例声明具体的QoS需求，客户在每次创建一个流（例如，每一次建立TCP连接）的时候做一次请求
粗粒度	提供商把服务规定为几种大的类别，每一类适合一种业务流类型，客户必须让所有的业务流都适合某种类型的服务

图28-3 为QoS服务规范提出的两种方法

28.11.1 细粒度QoS与流

有关QoS的很多早期工作都由电话公司来做。设计师们设想在电话系统之后建立一个面向连接的数据网络模型，即当客户要与远端站点（例如，一个Web服务器）通信时，该客户就会创建一个连接。此外，设计师们还假定客户对每条连接都会提出QoS要求，而提供商就会根据跨越的距离及所用的QoS计算出费用。

电话公司在异步传输模式（ATM）的设计中加入了很多QoS特性。虽然ATM没能生存下来，且提供商通常没有对每条连接都收费，然而ATM为细粒度QoS建立的一些术语却经过稍许的修改后仍然保留了下来。我们现在使用术语流（flow）来确定QoS，而不是针对每条连接指定QoS。流通常是指传输层的通信（例如，一条TCP连接；一对应用程序之间传输的一组UDP报文，或者一个VoIP电话呼叫）。图28-4列出了4种曾经在ATM中使用的主要服务类别，并解释了它们如何与流相关联。

缩　写	全　称	含　义
CBR	恒定位速率	数据以固定速率进入流，例如数字话音业务中的数据以精确的64Kbit/s速率进入流
VBR	可变位速率	数据在指定的统计范围内以可变速率进入流
ABR	可用位速率	流保证在给定时间内可使用任意的数据速率
UBR	未指定位速率	没有为流指定位速率，以尽力而为的服务来满足应用的需求

图28-4 4种主要的QoS服务类别

正如图28-4中所示，CBR服务适合以固定速率传输的数据流，数字化话音就是典型的例子。VBR服务适合使用可变速率编码的流，例如，一些视频编解码器发送的差分编码，一帧

中发送的数据量与前一帧和当前帧之间的差成正比。在这种情况下，客户可以指定预测的平均数据速率和最大的数据速率，以及最大速率出现的时间长度。VBR要求用户指定：

- 持续的位速率（Sustained Bit Rate，SBR）。
- 峰值位速率（Peak Bit Rate，PBR）。
- 持续的突发长度（Sustained Burst Size，SBS）。
- 峰值的突发长度（Peak Burst Size，PBS）。

ABR服务意味着共享——客户愿意支付任意数量的可用服务。如果其他客户发送数据，可用服务的数量将降低（这时提供商可能会收取更少的费用）。最后，UBR服务意味着客户不需要支付更高的费用，从而对尽力而为的服务感到满意。

当首次考虑因特网QoS的时候，电话公司认为：在分组网络中电话业务的质量被人们接受之前，是需要细粒度服务的。因此，除了在ATM上研究外，研究团体开始探索因特网上的细粒度QoS。这项研究被称为综合服务（Integrated Services，IntServ）。

28.11.2　粗粒度QoS与服务类别

替代细粒度QoS的另一种方法就是粗粒度方法。在这种方法中，业务被划分成几个类（classes），QoS参数被分配到类，而不是分配到单独的流。为了理解粗粒度方法的动机，有必要考虑在核心路由器上实现QoS的情况。与路由器的每个连接的速度都可能达到10 Gbit/s，这意味着分组以非常高的速率到达路由器；由于传统处理器的速度太慢，所以需要特殊的硬件来执行分组的转发。此外，由于核心路由器在主要的ISP之间承载通信业务，所以它能够处理数百万并发的流。QoS需要很多额外的资源。路由器必须为数百万的流维护状态，且必须为每个分组执行复杂的计算。内存访问会降低处理的速度。另外，路由器必须在流开始时分配资源，并在流结束时释放资源。

在经过对综合服务多年的研究并开发出几个协议之后，研究团体和IETF得出了结论：细粒度的方法一般是不切实际且不必要的。一方面，普通用户在选择参数时对QoS没有充分的理解。对于连接到一个典型网站的一个连接，人们究竟要指定怎样的吞吐率要求呢？另一方面，核心路由器没有足够的处理能力在每个流上实现QoS。因此，大部分关于QoS的工作都集中在确定几个广泛类别的服务上，而不是试图为每个独立的流提供端到端的QoS。我们可以概括如下：

> 尽管做了很多年的研究和标准化工作，然而QoS的细粒度方法还是被降格而限制于几种特殊的情况下使用。

28.12　QoS的实现

图28-5表示了使用交换机或路由器实现QoS的4个步骤。

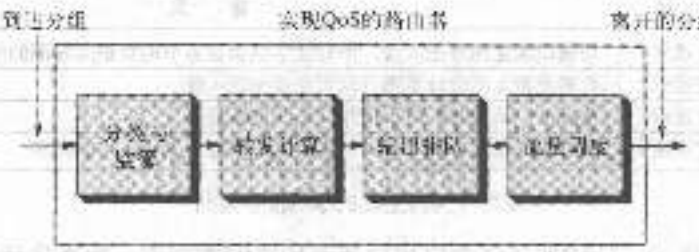


图28-5　实现QoS的4个关键步骤

分类与监管（classification and policing）。当分组到达时，路由器通过为该分组分配一个流标识符来对该分组进行分类（classifies）。对于细粒度系统，该标识符指定一个独立的连接；对于粗粒度系统，该标识符指定一个业务类别。一旦分配了标识符，路由器就执行监管（policing），这意味着该路由器要校验分组，确保分组没有违背流的参数。特别是，如果一个客户发送数据的速率大于他支付的最大速率，监管器就开始丢弃分组。有一项用于监管的技术被称为随机早期丢弃（Random Early Discard，RED），这项技术根据概率论的方法丢弃特定流中的分组。为流建立一个队列，队列的当前大小可用于决定丢弃的概率。当队列小于其总长的一半时，概率设置为0。当队列完全满时，概率设置为1。当队列的大小处在这两者之间时，概率与队列中分组的数量成线性正比。使用RED技术有助于避免由于队尾丢弃（tail drop）而导致的循环问题。如果是队尾丢弃，当队列填满时，所有传入的分组都被丢弃，很多TCP会话被取消并重新缓慢开始，通信量不断上升，直到队列再次填满，循环会不断重复。

转发计算（forwarding computation）。在计算下一跳路径时，路由器或交换机可以使用流标识符。在某些情况下，流标识符确定了要遵循的路径（例如，所有的话音业务从端口54发送到一个话音交换机中）。在其他情况下，流标识符被忽略，每个分组中的目的地址用于选择下一跳。转发的具体细节取决于特定交换机或路由器的用途以及管理员的QoS策略。

输出排队（Output Queuing）。大部分QoS的实现都为每个输出端口建立一组队列。一旦转发计算为分组选择好一个输出端口，输出队列机制使用流标识符就把该分组放到与端口相关联的一个队列中。粗粒度系统通常是每一类业务使用一个队列。因此，如果管理员建立8个QoS类别，每个输出端口就会有8个队列。细粒度系统通常是每条连接有一个队列，且队列被安排成一个分层系统。例如，一个网络处理器芯片提供了256 000个队列，这些队列被安排在一个多级别的分层系统里。

流量调度（traffic scheduling）。当一个端口空闲时，由流量调度器（traffic scheduler）选择一个分组发送，从而实现QoS策略。例如，管理员可能规定3个客户各自接收容量的25%，而其余客户则共享剩下的容量。为了实现这样一个策略，流量调度器可以使用4个队列及轮询（round-robin）的方法来选择分组。因此，如果所有客户同时都在发送数据，则有3个被指定的客户各自将得到总容量的1/4（25%），正如策略所规定的那样。

更多精巧的分组选择算法可用于实现复杂的按比例共享。这种复杂性是由于流量调度器必须要维持长期策略，即使分组是突发性到达的。因此，流量调度器必须适应这种情况：某个队列临时超过了它分配到的数据速率，但却符合规定界限内的长期平均水平。类似地，流量调度器也必须适应这样的情况：由其他队列共同划分未使用的容量，因而就有可能使一个或多个队列暂时变为空。

已经提出并分析过很多流量调度算法。要创造一个完善的实用算法是不可能的：每个算法都是在公平性与计算开销之间权衡折中。如图28-6所示为当前已经提出和研究过的流量管理算法。

算　法	描　述
漏桶	允许队列按固定速率发送分组，周期性地递增分组计数器，并使用计数器来控制传输
令牌桶	允许队列按固定速率发送数据，周期性地递增分组计数器，并使用计数器来控制传输
加权轮询	根据所占据容量的百分比分配一组权重，再根据权重从各队列选择分组（假定分组大小一致）
差额轮询	轮询方法的变种，它按被发送的字节（而不是按传输的分组）来计数，并允许由大分组所造成的临时差额

图28-6 流量调度算法举例

28.13　因特网QoS技术

IETF已经设计了一系列有关QoS的技术和协议，3个重要的努力成果是：

- RSVP与COPS，
- 区分服务，
- MPLS。

RSVP与COPS。在探讨综合服务时，IETF开发了两种协议来提供QoS：资源预留协议（Resource ReSerVation Protocol，RSVP）和公共开放策略服务（Common Open Policy Service，COPS）协议。RSVP是QoS的细粒度版本，因此每个TCP或UDP会话都需要RSVP。为了使用RSVP，应用程序要发送一个指定所需QoS的请求。沿着从源点到目的地路径上的每个路由器都要保存所请求的资源，并把该请求传递给下一个路由器。最后，目的主机必须同意该请求。当路径中的每一跳都同意该请求时，就产生一个流标识符并返回。这样应用业务就可以沿着预留路径传送。COPS是RSVP的伙伴协议，用于规定和执行策略。执行策略的路由器使用COPS与策略服务器通信，并获取关于流参数的信息。由于RSVP是设计来提供细粒度的（即每个流的QoS），因此它很少被采用。

区分服务。在放弃综合服务和细粒度QoS后，IETF又创造了区分服务（Differentiated Services，DiffServ），即定义了一个粗粒度的QoS机制。区分服务定义了如何规定服务等级，以及怎样用IPv4或IPv6头部中的服务类型（type of service）域来规定数据报的服务等级。虽然很多ISP都对区分服务进行了实验，不过该技术还没有被广泛接受。

MPLS。第19章描述了多协议标记交换（MultiProtocol Label Switching，MPLS），这是一种建立在IP之上的面向连接的通信机制。为了使用MPLS，管理员通过一组具有MPLS能力的路由器配置转发路径。在路径的一端，每个数据报文封装MPLS的头部并注入到MPLS路径中；在另一端，数据报文被提取出来，去掉MPLS头部，然后把数据报文转发到它的目的地。在许多情况下，流量调度策略被分配到MPLS路径中，这意味着当一个数据报文被插入到特定的路径时，该数据报文设置了QoS参数。因此，ISP可能会为语音数据建立一条MPLS路径，以便区分其他数据所使用的MPLS路径。

28.14　本章小结

网络性能的两种主要度量是延迟（即从一台计算机到另一台计算机之间发送一个比特所需要的时间）和吞吐率（即每秒钟通过网络传输的比特数量）。虽然吞吐率通常也被叫做速度，但吞吐率才是对网络容量的度量。延迟—吞吐率乘积用于测量在某一时刻网络正在传输中的数据量，延迟与吞吐率之间不是独立无关的——当吞吐率接近容量的100%时，延迟急剧上升。

抖动是对延迟差异性的度量，在数据网络中变得日益重要。等时网络或者具有处理实时音频与视频传输协议的网络都可以实现低的抖动，因特网是使用协议方法来实现低抖动性能的。

网络性能的测量是困难的。不对称的路由意味着需要使用同步时钟来测量延迟；突发性业务意味着性能可能会快速改变。由于来自测量的额外流量可能会改变网络条件，因此很多管理员选用被动的测量技术，例如NetFlow（回流监测）。

细粒度QoS与粗粒度QoS都已经被广泛研究过了，细粒度的研究成果已被普遍放弃。ATM定义了服务的类别，它的缩写词仍然在使用：CBR（恒定比特率），VBR（可变比特率），ABR（可用比特率）以及UBR（未指定比特率）。

为了实现QoS，交换机或路由器对输入的数据进行分类和监管，转发分组并把每个分组放到一个输出队列中，当有输出端口空闲时，就使用流量调度器选择一个分组发送。有几个流量调度算法已经被提出和分析，每个算法都是在最佳公平性与计算开销之间权衡折中。

IETF定义了RSVP与COPS作为综合服务的部分研究成果，当重心从细粒度QoS转移并后，IETF定义了区分服务。IETF也定义了MPLS作为流量工程的技术。QoS参数可以与每个MPLS隧道相关联，这意味着一旦一个数据报已经分好类，与它关联的MPLS即定义出它的QoS参数。

练习题

28.1 列举并描述3个主要的网络性能量度。

28.2 给出5种类型的延迟，并对每种延迟做出解释。

28.3 你认为LAN或WAN中的接入延迟是长还是短？排队延迟呢？为什么？

28.4 怎样才能测量出吞吐率？

28.5 什么名称可用于形容吞吐率，且对用户来说是最有意义的？

28.6 给出使实际吞吐量小于信道容量的数据处理的例子。

28.7 根据正在传输中的码位，解释延迟与吞吐率的含义。

28.8 延迟或吞吐率，哪一个在性能方面提供了最根本的限制？为什么？

28.9 使用ping命令测量到本地与远地站点的网络延迟。你可能得到的因特网延迟的最小值和最大值是多少？

28.10 如果你ping IP地址127.0.0.1，等待时间将非常短。请解释原因。

28.11 下载一份ttcp程序，用它来测量本地以太网的吞吐率。实际吞吐率是多少？估计一下链路能达到的利用率。

28.12 比较一下100 Mbit/s网络与1 Gbit/s网络的吞吐率。

28.13 什么是抖动？用于克服抖动的两种方法是什么？

28.14 专业人员有时候在时延曲线中提到“拐点”。要理解他们的意思，为0～0.95之间的利用率的值，绘制有效时延的曲线。在该曲线急剧上升时，你能够为其找出一个利用率的值吗？

28.15 有多少数据可以在地面站、卫星和接收站之间“飞行”？要找出答案，请计算运行在3 Mbit/s的GEO卫星网络的迟延-吞吐率乘积。假设卫星运行在地球上面20 000英里的轨道上，而无线电传输以光速传播。

28.16 为什么网络性能的量度是困难的？

28.17 数据业务流与话音业务流有什么不同？

28.18 为什么ISP是计算单位时间内接收到的分组数量，而不是仅仅计算单位时间内接收到的字节数？请解释。

28.19 QoS的两种类型是什么？

28.20 估算一下在核心因特网中实现细粒度QoS所需要的计算能力：假设一个10 Gbit/s链路传送1 000字节的分组，且每个分组执行N次算术运算，然后计算一个处理器每秒需要执行的运算次数。

28.21 列出4种起源于ATM的主要QoS类别，并给出各自的含义。

28.22 考虑一个Web浏览器，在浏览器下载一个网页的典型的流中，采用哪种类型的QoS比较合适？为什么？

28.23　如果两个用户通过因特网创建一个聊天会话，他们会使用哪种类别的QoS？

28.24　哪4种参数是用于表征VBR流的？

28.25　解释用于实现QoS的4个步骤。

28.26　如果你的ISP使用漏桶算法来调度分组传输，那么是使用大分组的吞吐率高，还是使用小分组的吞吐率高？请解释。

28.27　什么是区分服务？

28.28　MPLS转发与传统IP转发有什么区别？

第29章　多媒体与IP电话

29.1　引言

本书这一部分的几章主要考虑各种网络技术和它们的用途。第28章讨论了网络性能和QoS，本章将指出网络设计的两种基本方法，以便将网络设计成能向实时应用（例如话音）提供服务。这两种方法是：等时服务基础结构和抖动补偿协议的使用。

本章继续前面的讨论，考查多媒体在因特网上的传送，重点考查它如何利用尽力而为的通信机制发送数据，然后讲述一种针对实时业务的通用协议，并详细介绍话音电话业务的传输过程。

29.2　实时数据传输和尽力而为传递

我们使用术语多媒体（multimedia）来指含有音频和视频的数据，也可以含有文本。短语实时多媒体（real-time multimedia）指的是那些还原重现速率必须与捕获速率完全一样的多媒体数据（例如，包含真实事件的音频和视频的电视新闻节目）。

由此产生了一个问题：如何使用因特网传输实时多媒体呢？为了理解问题的难度，回顾一下，因特网提供的是尽力而为的递送服务。因此，分组会丢失、延迟或是乱序到达。音频或视频信号数字化后，如果不采取特别措施就在因特网上发送，且到达后马上展现出来，那么这种输出结果是无法让人接受的。早期的多媒体系统解决这个问题的办法是构造一些特别设计的通信网络去处理音频和视频。模拟电话网采用等时网络来提供高质量音频的重现，而模拟有线电视系统则被设计成能传递多个广播视频频道，不会出现信号中断或是丢失。

因特网并不要求底层网络来处理实时业务的传输问题，而是利用附加协议的支持。有意思的是，需要处理的最大问题是抖动，而非分组丢失。为了理解其中的原因，考虑一个正在直播的网络广播（webcast），如果协议采用超时重传的机制来重新发送分组，那么重传的分组会因为到达得太晚而失去作用——接收端可能已经播放了从连续分组中提取的视频和音频，再把已经丢失了的小片段插播进来是毫无意义的。

要点　与常规的传输协议不同，传输实时数据的协议只需要处理抖动问题，无须重传丢失的分组。

29.3　延迟重播与抖动缓冲

为了克服抖动，实现实时数据的平滑重放（playback），人们采用了两种主要的技术：

- 时间戳：发送方为数据的每个小片段都提供一个时间戳，接收方利用时间戳来处理乱序的分组并按正确的时间顺序来展现数据。
- 抖动缓冲：为了适应抖动（即很小的延迟变化），接收方会缓冲收到的数据并推迟重放。

抖动缓冲的实现比较简单，只需接收方维护一个数据项列表，并利用时间戳来排序列表。接收方在开始重放之前，会延迟d个时间单位，即正在播放的数据比刚到达的数据落后d个时

间单位。因此，如果一个给定的分组延迟的时间小于d，那么在需要对它做重放处理之前，分组的内容就会放置在缓冲区中，换句话说，数据项以稍有些变化的速率被插入到抖动缓冲区中，但是重放处理过程则要以一个固定的速率从抖动缓冲区中提取数据。图29-1说明了实时重放系统的构成。

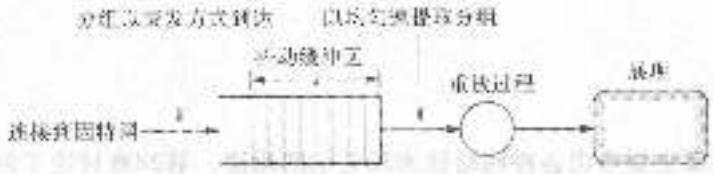


图29-1　延迟量为d的抖动缓冲区示意图

29.4　实时传输协议

在因特网协议簇中，实时传输协议（Real-time Transport Protocol，RTP）提供跨因特网传输实时数据的机制。使用传输（transport）这个词汇在此不恰当，因为RTP位于传输层协议之上，因此，不管其名称如何，人们都应该把RTP看做是一个传送协议。

RTP并不保证数据的及时传递，也不包含抖动缓冲或重放机制，而是在每个分组中提供了3项内容，能使接收方实现一个抖动缓冲区：

- 一个序列号，允许接收方将进入的分组按正确的顺序放置，并检测丢失的分组。
- 一个时间戳，允许接收方按多媒体流中正确的时间播放分组中的数据。
- 一系列源标识符，能让接收方知道数据的来源。

图29-2说明了序列号、时间戳和源标识符是如何出现在RTP分组头部中的。

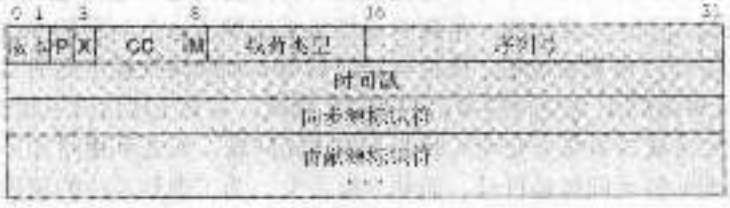


图29-2　出现在每个RTP分组开始部位的基本头部

“版本”域给出RTP的版本号，当前是2。“P”域指出载荷部分是否是零填充（有些编码要求固定的块长度）。“X”域指出是否存在头部扩展，而“CC”则给出源的个数，这些源按下面描述的方式组合到一起产生媒体流。M是一个标志位，它被用来标记某些帧。特别是，一些视频编码发送一个全帧之后会跟着一系列的增量变化帧；在这种情况下，只有RTP分组携带一个全帧的时候，M位才会被设置。“载荷类型”域指定载荷的类型，接收方利用“载荷类型”的值来解释分组的剩余部分。

每个分组包含一个“序列号”，每个分组的序列号依次增加1，与TCP的做法类似，发送方选择一个随机的开始序号，以帮助避免重放问题。“时间戳”域与序列号独立，它向接收者提供有关重放定时的信息，在时间与分组顺序非线性相关的情况下（例如，可变长度视频编码方案，它们在图片不发生快速改变时，只发送很少的分组），保持时间戳与序列号的独立性是十分重要的。

一个RTP“时间戳”不会对数据和时间进行编码，而是选择一个随机的初始时间戳，然后

相对这个初始值来制定每个后续的时间戳。此外，RTP并没有规定时间是否以秒、毫秒或是其他单位来衡量——由载荷类型来决定时间戳的粒度（granularity）。无论采用什么样的粒度，发送方必须连续地增加时间，即使没有分组发送的时候（例如，编解码器可能会在音频流的静默期间抑制传输）。

另两个域即"同步源标识符"和"贡献源标识符"用来标识数据源。数据源必须标识的原因出自于组播传递机制：一个主机可能会从多个源接收数据，可能会收到一个指定分组的多个副本。标识多个数据源的原因则出自于一种称之为混频（mixing）的技术，中间系统将来自多个实时流的数据进行组合从而产生一个新的流。例如，一个混频器可以把某部电影中分开的视频流和音频流组合起来，然后以组播方式传递组合后的流。

29.5 RTP封装

RTP使用UDP来传输报文，因此每个RTP报文被封装在一个UDP数据报中在因特网上传输。图29-3说明了一个RTP报文在单个网络上传送时所经历的3级封装过程。

因为RTP采用UDP封装，所以所形成的报文可以通过广播或是组播来发送。组播对于传递那些会吸引大量观众的娱乐节目尤其有用。例如，如果一个有线电视提供商提供一个电视节目或是体育赛事，多个客户可以同时观看。在这种情况下，提供商无须发送报文的副本到每个订阅者，RTP允许他在每个本地子网上通过组播来发送RTP报文的副本到达客户。如果指定的组播平均到达N个客户，那么业务总量将减少N倍。

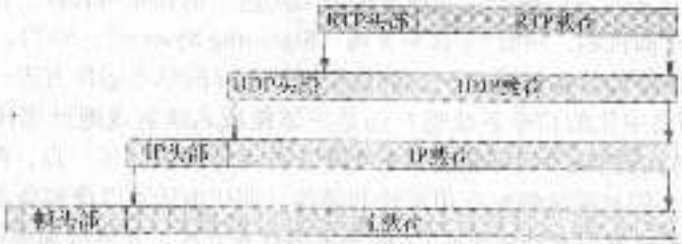


图29-3 RTP报文的3级封装过程

29.6 IP电话

术语IP电话（IP telephony[⊖]）或IP语音（Voice over IP，VoIP）用于描述一种最普遍的多媒体应用。全世界的电话公司都在用IP路由器取代传统的电话交换机。其动机来源于经济方面：路由器的花费远低于传统的电话交换机。企业出于经济方面的原因也开始使用IP电话——通过IP数据报来发送数据和话音可降低费用，这是因为底层网络基础设施是共享的，它为包括电话业务在内的所有通信提供服务，只要有一套用户设备、连线和网络连接就够了。

隐含在IP电话之后的基本思想很简单：连续地对音频信号进行采样，将每个采样点转换成数字形式，通过IP网络以分组形式发送所形成的数字化流，然后又将数字化流转换还原成模拟信号进行播放。然而，很多的细节使任务复杂化。发送者不能等着去填满一个大的分组，因为这样做会造成几秒的传输延迟。系统必须处理呼叫建立过程：当主叫方拨出一个呼叫时，系统必须将电话号码翻译成IP地址，然后定位指定的被叫方。当一次呼叫开始时，被叫方必须接受并回答呼叫。类似地，当一次呼叫结束时，双方必须在如何来终结通信方面达成一致。

⊖ 读作：I-P tel-ef-oh nee。

最显著的复杂性在于IP电话必须要尽力做到与已有的公共交换电话网络（Public Switched Telephone Network，PSTN）向后兼容。也就是说，它的机制不能仅限于对IP电话的呼叫，而是要允许主叫方和被叫方在PSTN的任何地方（包括国际线路和蜂窝连接）都能使用电话通信。因此，IP电话系统必须准备处理从PSTN到IP电话的呼叫，或者反方向的呼叫。用户希望IP电话系统能提供现有的各种电话业务，例如呼叫转移、呼叫等待、语音邮件、会议电话以及主叫身份识别等。此外，对于目前还使用用户专用交换机（Private Branch Exchange，PBX）的商务应用，也可能会要求IP电话系统提供等效于PBX的业务。

29.7　信令与VoIP信令标准

目前已经有两个组织为IP电话制定了标准：国际电信联盟（International Telecommunications Union，ITU）和因特网工程任务组（Internet engineering task force）。前者主持制定电话方面的标准，后者主持制定TCP/IP方面的标准。在介绍了IP电话系统的概念性部分之后，我们将回顾一下每个组织已经选定好的协议。

还好，这两个组织在音频编码和传输的基础技术方面所见略同：

- 音频采用脉冲编码调制（Pulse Code Modulation，PCM）进行编码。
- 数字化音频采用RTP进行传输。

IP电话的主要复杂之处（以及要提出多个标准的理由）在于呼叫建立和呼叫管理方面。在电话术语里，建立和终止呼叫的过程被称作信令（signaling）过程，包括：电话号码与位置的映射，找出到达被叫方的路径并处理其他细节问题，例如呼叫转移。在传统电话系统中处理呼叫管理所采用的机制，叫做7号信令系统（Signaling System7，SS7）。

IP电话中心所面临的基本问题之一，就是采用什么样的信令运作方式——应该像目前的电话系统那样采用集中化的信令系统呢？还是应该像域名映射成地址那样采用分布式系统呢？分布方式的支持者们认为有可能这样实现：在因特网上的任一点，两个IP电话可以彼此找到对方并进行类似目前万特网应用那样的通信（即IP电话可以像服务器那样接受进来的呼叫，也可像客户那样发出对外的呼叫）。如果采用分布方式，无须增加另外的基础设施，在当前用于数据通信的DNS和IP转发服务的基础上即可实现。对于局域范围内的IP电话系统（例如，在单个公司范围内允许两个IP电话间进行呼叫的系统），这种分布方式就特别适合。集中方式的支持者们则认为传统的电话运作模式是最好的，因为由电话公司来控制呼叫的建立，才能提供服务保证。

为了与已有的电话系统兼容，新的协议必须能与SS7进行交互，使两个系统都能发出呼叫和接受呼入。随着关于基本运作方式争论的不断向前推进，人们已经提出了4套信令协议与IP电话一起使用：IETF提出了会话初始协议（Session Initiation Protocol，SIP）和媒体网关控制协议（Media Gateway Control Protocol，MGCP），ITU在H.323的总体框架下提出了一个大型的综合协议组。这两个机构还联合提出了Megaco（H.248）。

要点　呼叫的建立与终止过程被称作信令过程，人们已经提出了多种信令协议与IP电话一起使用。

29.8　IP电话系统的组成部件

图29-4列出了IP电话系统的4个主要部件，图29-5说明了它们是如何用于互连网络的。

部 件	说 明
IP话机	操作上与传统话机相似，但是使用IP来发送数字化语音
媒体网关控制器	为IP话机间的业务（例如，呼叫建立、呼叫终止和呼叫转移）提供和管理呼叫功能
媒体网关	在两个使用不同编码格式的网络间提供转换功能，其实际呼叫通过对其进行格式转换
信令网关	连接两个使用不同信令机制的网络，其实际呼叫管理信息和响应进行转换

图29.4 IP电话系统的4个主要构件

IP话机（IP telephone）连接两网络，利用IP来进行所有通信。它提供一个传统的电话接口允许用户拨出或接受电话呼叫。IP话机可以是一个单独的硬件单元（即传统电话），或者也可以由带有麦克风、扬声器和IP电话软件的计算机构成。IP话机与世界其他地区的连接可以由有线或无线网络（例如，以太网或802.11b无线局域网）构成。

媒体网关控制器（media gateway controller），也叫关守（gatekeeper）或软交换机（softswitch），提供对IP电话之间的整体控制和协调，允许主叫方定位被叫方或是接入像呼叫转移之类的服务。

媒体网关（media gateway）在呼叫通过IP网络与PSTN之间的边界或两个采用不同编码格式的IP网络之间的边界时，对话音数据提供转换功能。例如，在PSTN与因特网之间边界上的媒体网关，就要实现传统话音线路上采用的TDM编码与因特网上采用的分组编码之间数字化话音的来回转换。

信令网关（signaling gateway）也跨在一对异构网络之间的边界上，它提供信令操作的转换，允许任何一侧的用户发起呼叫（例如，允许连特网上的一部IP话机向PSTN上的一部电话发起呼叫）。媒体网关控制器负责协调媒体网关和信令网关之间的操作。图29-5说明了如何利用这些部件对因特网与PSTN进行互连。



图29-5 IP电话不同部件间的连接

上面定义的概念和术语表现出对于IP电话的最直接、相对简化的观点，它来自于IETF和ITU在Megaco和媒体网关控制协议（MGCP）方面所做的工作。IP电话业务的实际实现要更复杂一些。下面几节将给出一些例子。

29.8.1 SIP术语与概念

只要可能，会话初启协议（SIP）会尽量利用现有协议，从而尽量少用附加的协议。例如，SIP利用域名系统将电话号码映射成IP地址。其结果是，SIP只定义了构成信令系统的3个新元素：

- 用户代理。
- 定位服务器。
- 支持服务器（代理服务器、重定向服务器、注册服务器）。

用户代理（user agent）。SIP文档特指那种能建立和终止电话呼叫过程的设备为用户代理。SIP用户代理可以在IP话机、便携式电脑中实现，或者在允许IP话机向PSTN呼叫的PSTN网关中实现。用户代理包含两个部分：产生呼出的用户代理客户（user agent client）和处理呼入的

用户代理服务器（user agent server）。

定位服务器（location server）。SIP定位服务器管理有关每个用户信息（例如，IP地址集合，用户订购的业务以及用户的偏好）的数据库，在呼叫建立期间，需要联系定位服务器以获取被叫用户的位置信息。

代理服务器（proxy server）。SIP使用了代理的概念，是指代理服务器可以将请求从用户代理转发到另一个位置。代理服务器处理优化路由和实施策略（例如，确保主叫方有权进行呼叫）。

重定向服务器（redirect server）。SIP利用重定向服务器处理诸如呼叫转移和800号电话之类的任务。它接收来自用户代理的请求，并返回替代的位置供用户代理联系。

注册服务器（registrar server）。SIP使用注册服务器来接收注册请求并更新被位置服务器访问的数据库。注册服务器要负责验证注册请求并确保底层数据库保持一致性。

29.8.2　H.323术语与概念

由ITU制定的H.323标准定义了另外一些术语和附加概念，重点是有关与PSTN交互的过程。虽然它的内容极其广泛且涉及到很多细节，但还是可以将它概述如下：

终端。H.323终端提供IP话机功能，也可能包括视频和数据传输的设施。

关守。H.323关守提供定位和信令功能，并协调与PSTN连接的网关的操作。

网关。H.323使用单个网关将IP电话系统与PSTN进行互连，它还处理信令和媒体转换。

多点控制单元（Multipoint Control Unit，MCU）。MCU提供诸如多点会议之类的业务。

29.8.3　ISC术语与概念

由于ITU和IETF对术语和概念的说法不一致，提供商成立了国际软交换联盟（International Softswitch Consortium，ISC）以便创建一个统一的综合功能模型，将所有IP电话模型合并到一个单一框架内。为此，ISC定义了可能需要用到的功能，包括：不同系统间的信令、编码转换，对诸如呼叫转移之类业务的支持以及诸如计费和付费之类的管理功能。然后，ISC定义出适合各种场合下使用的功能列表：

媒体网关控制功能（MGC-F）。MGC-F负责维护各个端点的状态信息，提供呼叫逻辑和呼叫控制。

呼叫代理功能（CA-F）。CA-F是MGC-F的一个子集，它维护呼叫状态。SIP、H.323和Q.931都是CA-F的例子。

互通功能（IW-F）。IW-F也是MGC-F的一个子集，它处理异构网络（如SS7与SIP）之间的信令。

路由功能和计费功能（R-F/A-F）。R-F为MGC-F处理呼叫路由；A-F收集有关用于计费和付费的信息。

信令网关功能（SG-F）。SG-F处理IP网络与PSTN之间的信令。

接入网关信令功能（AGS-F）。AGS-F处理IP网络与线路交换接入网（如ISDN）之间的信令。

应用服务器功能（AS-F）。AS-F处理一系列应用方面的业务（如语音邮件）。

服务控制功能（SC-F）。当AS-F必须要控制（即改变）业务逻辑（如安装一个新的映射关系）时，它要调用SC-F。

媒体网关功能（MG-F）。MG-F处理两种不同格式的数字化语音的转换，也可能包括对一些诸如是否摘机之类的事件的检测和对双音多频（Dual Tone Multi-Frequency，DTMF）信令的识别。DTMF是一种称为触音式（touch tone）编码的音频信令标准。

媒体服务器功能（MS-F）。MS-F在AS-F应用方面对媒体分组数据流实施操作。

29.9 协议及所在层次归纳

由于已经有很多团体提出了有关IP电话的协议，因此在协议栈的大多数层次上出现了有竞争关系的协议。图29-6列出了被提议的一些协议以及它们在因特网5层参考模型中的位置。

层	呼叫处理	用户多媒体	用户电路	支持协议	路由协议	信令传输
5	H.323 Megaco MGCP SIP	RTP	T.120	RTCP RTSP NTP SDP	ENUM TRIP	SIGTRAN ⊖
4	TCP UDP	UDP	TCP	TCP UDP		SCTP
3	IP, RSVP, IGMP					

图29-6 IP电话协议汇总

29.10 H.323特性

由ITU制定的H.323标准不是单个协议，而是由一组协议组成，它们一起工作共同处理电话通信的所有方面。H.323的要点如下：

- 处理数字电话呼叫的各个方面。
- 包含建立和管理呼叫的信令。
- 在通话过程中允许传输视频和数据。
- 发送由ASN.1定义的并经基本编码规则（Basic Encoding Rules，BER）编码的二进制报文。
- 合并了安全方面的一些协议。
- 使用一种称为多点控制单元的特殊硬件来支持会议电话业务。
- 定义各种服务器以处理各种任务，例如地址解析（即将被叫方号码映射为IP地址）、认证、权限（即确定一个用户是否有权访问指定服务）、计费以及特色服务（例如呼叫转移）等。

29.11 H.323分层

H.323协议利用TCP和UDP两者作为传输协议——在使用TCP来传输数据的过程中，可以利用UDP来传送音频。图29-7列出了H.323标准中的基本分层情况。

层	信令	注册	音频	视频	数据	安全
5	H.225.0 Q.931 H.250 Annex G H.245 H.250	H.225.0-RAS	G.711 H.263 G.722 G.723 G.728	H.261 H.323	T.120	H.235
			RTP, RTCP			
4	TCP, UDP	UDP			TCP	TCP, UDP
3	IP, RSVP, IGMP					

图29-7 H.323标准中主要协议的分层情况

⊖ SIGTRAN支持PSTN信令（即SS7，DTMF）通过IP网络传输，SCTP将多个输入流复用到单个传输层流上。

29.12　SIP特性和方法

IETF定义的会话初启协议（SIP）的要点如下：

- 在应用层上运行。
- 包含信令的所有方面，包括：被叫方的定位、通知与会话建立（即振铃），可用性确认（即被叫方是否接受呼叫），终结。
- 提供诸如呼叫转移之类的服务。
- 会议业务要靠组播来实现。
- 允许双方对能力的协商，并允许选择所用的媒体和参数。[1]

SIP URI包含一个能找到用户的用户名和域名。例如，一个名为Smith在Somecompany公司工作的用户，可能被分派的SIP URI是：

sip:smith @ somecompany.com

SIP定义了6个基本的报文类型和7种扩展。基本的消息类型称为方法（method）。图29-8列出了基本的SIP方法。

方　法	用　途
INVITE	建立会话：请某个端点参加会话
ACK	对INVITE的响应确认
BYE	终止会话：呼叫结束
CANCEL	取消未完成的请求（如果请求已经完成则无效）
REGISTER	注册用户位置（即能到达用户的一个URL）
OPTIONS	查询以便确定被叫方的能力

图29-8　SIP使用的6个基本方法

29.13　SIP会话举例

通过一个在SIP会话期间发送报文的例子，即可解释SIP协议的一些细节。图29-9列出一系列报文的发送过程：用户代理A先联系DNS服务器，然后与代理服务器通信，而代理服务器又要请求定位服务器。[2]一旦呼叫建立，两个IP话机即可直接通信。最后，再使用SIP来终止这次通信。

通常，要给一个用户代理配置一个或多个DNS服务器的IP地址（用于将SIP URI中的域名映射到IP地址上）以及一个或多个代理服务器。类似地，也要给每个代理服务器配置一个或多个位置服务器的地址。因此，在给定的服务器无效时，SIP可以很



图29-9　SIP通过交换报文来管理一次电话呼叫的例子

[1] SIP使用会话描述协议（Session Description Protocol，SDP）来描述能力和参数。

[2] 实际中，SIP支持呼叫数字（null forking），允许位置服务器送回一个用户的多个位置（如家里、办公室），同时也允许用户代理同时尝试联系这些位置。

快找到另一个服务器。

29.14 电话号码映射及路由

应该如何给IP用户命名和定位呢？PSTN遵循ITU标准E.164规定的电话号码格式，而SIP则使用IP地址。定位一个用户的问题比较复杂，因为可能要涉及多种网络类型。例如，考虑一个由两个PSTN网络通过IP网络互连而成的综合型网络。设计者定义了两个子问题：如何在综合型网络中定位一个用户，如何找到一条通往用户的有效路径。对应于这两个子问题的映射需求，IETF已经提出了两个协议：

ENUM——将电话号码转换成URI。

TRIP——找到综合网络中的用户。

ENUM。IETF的ENUM协议（E.164 NUMbers的缩写）解决从E.164电话号码转换成统一资源标识符（Uniform Resource Identifier，URI）的问题。实质上，ENUM要利用域名系统来存储映射关系，电话号码被转换成下面这个域中的一个特殊的域名：

e164.arpa

这种转换过程包括要处理电话号码串、逆转这个串、把单个数字写作为域名的段。例如，电话号码1-800-555-1234，会产生如下的域名：

4.3.2.1.5.5.5.0.0.8.1.e164.arpa

ENUM映射可以像传统电话号码方案那样是一对一的关系，也可以是一对多的关系。这也意味着可以分配同一个电话号码给用户的桌面电话和移动电话。当一个号码对应多个主机的时候，DNS服务器会返回一个主机的列表以及到达每个主机所用的协议。用户代理会一直联系列表中的主机，直到其中一台主机有反应为止。DNS返回一个对应于电话号码的主机列表以及到达每个主机所要使用的协议。

TRIP。这是IETF的基于IP的电话路由协议（Telephone Routing over IP，TRIP），它解决在综合型网络中找到一个用户的问题。位置服务器或其他网络元素可以使用TRIP来通告路径。这样，两个位置服务器可利用TRIP来互相通知它们彼此知道的外部路径。因为TRIP独立于信令协议，所以它可以与SIP或其他信令机制一起使用。

TRIP把世界划分成一组IP电话管理域（IP Telephone Administrative Domain，ITAD）。实质上，TRIP通告标识了一个出口点——一个位置服务器向另一个位置服务器通告一条通路，该通路能到达互连到另一个ITAD去的信令网关。由于IP电话还是新的事物，路由信息在将来可能还会变化，所以TRIP被设计成可扩充的协议。

29.15 本章小结

实时传输协议能提供实时多媒体在因特网上传送的能力。RTP报文包含序列号、单独的时间戳以及数据源的标识。接收方会在回放（playback）前利用时间戳将数据放入抖动缓冲区。RTP封装在UDP中传输，允许采用组播或广播方式发送。它不采用重传机制，因为那些在回放窗口之后接收的分组不能再被播放。

术语IP电话和VoIP指的是在因特网上传输的数字化电话业务。构建IP电话系统的最大挑战之一是它的向后兼容性——必须发明能连接IP电话系统到传统PSTN的网关设备。这种网关必须提供媒体转换（即不同语音编码之间的转换）和信令（即呼叫建立机制的转换）功能。

ITU和IETF都已经制定了有关IP电话的标准。ITU标准H.323包含了许多协议，提供呼叫

建立与管理、认证与计费、用户服务（例如呼叫转移），以及在电话连接上对话音、视频和数据的传输。IETF标准SIP提供信令能力，包括用户定位、呼叫建立以及为双方指定能力特性。SIP利用一组服务器来处理信令的各个方面：域名服务器、代理服务器和位置服务器。因特网软交换联盟（ISC）已经定义了一个附加的框架，试图包罗所有的IP电话系统模型。

还有另外两个IETF协议提供支持功能。ENUM协议利用域名系统将E.164电话号码映射成为统一资源标识符（通常是一个SIP URI）。TRIP协议提供IP电话管理域之间的路由能力。SIP位置服务器可以利用TRIP来形成网络出口点的相关信息通知给其他位置服务器。

进一步的阅读资料

RFC3216定义了SIP，RFC2916涉及E.164号码和DNS，RFC3219定义了TRIP。RTP及其相关的控制协议RTCP则归入到RFC1889文档中。有关的概念和协议也可以在RFC2915、2871、3015、3435和3475中找到。

练习题

29.1　试定义多媒体数据。用于克服抖动的两种技术是什么？

29.2　试解释即便因特网引入了抖动，抖动缓冲区是如何允许音频流回放的。

29.3　如果一个RTP在穿越因特网时被截获，或获者能否对时间戳域的内容进行解释？如果能，如何解释？如果不能，那又是为什么？

29.4　由于RTP报文封装在UDP中传递，因此可能会出现重复。接收方是否需要保留所有先前收到的报文的副本才能确定到达的报文是重复的呢？为什么？

29.5　RTP包含一个辅助协议称为实时控制协议（Real-Time Control Protocol，RTCP），允许接收方向发送方报告接收到的报文质量。自适应视频编码怎样才能利用接收报文的状态信息呢？

29.6　如果采用PCM将话音转换成数字形式，请问在1/2s内将产生多少数据位？

29.7　扩展上一道题。将1/4s的PCM音频数据先打包成RTP分组，再封装成UDP报文，最后装配成IP数据报。请估计这个IP数据报的长度（字节数）。提示：RFC1889定义了RTP头部的长度。

29.8　H.323处理IP电话的哪些方面？

29.9　H.323用于发送数据和音频（或视频）时，会使用哪个传输协议？

29.10　SIP使用的6个基本方法是什么？

29.11　阅读关于SIP的RFC文档，然后修改图29-9，以表示出发生呼叫转移时的报文交换过程。提示：查看SIP重定向报文。

29.12　ENUM协议和TRIP协议的主要用途是什么？

29.13　考虑一下IP电话和模拟电话的工作情况。请问：在战争时期，哪种电话更好？为什么？

29.14　查阅有关e164.arpa域的资料，请问是哪个机构负责管理这个域？

第30章 网络安全

30.1 引言

前面几章阐述了组成因特网的硬件和软件系统，并解释了客户和服务器应用程序如何使用底层设施进行通信。本章将介绍一个重要的方面——网络安全问题，描述因特网犯罪的类型，讨论有关安全方面的关键话题，以及介绍用于增强网络安全的技术。

30.2 网络犯罪与攻击

每当有新技术出现时，罪犯们总会考虑如何利用新技术去进行犯罪。因特网也不例外——像大多数用户所知，利用因特网进行的犯罪活动常见诸报端。虽然网络犯罪（如诡计和身份偷窃）会影响个人，但最大的伤害是对商务活动造成威胁。除了直接的货物和服务盗窃外，特别是会威胁到公司的长期生存力。声誉的损害、客户信心的丢失、知识产权的失窃，以及用户访问的妨碍，所有这些对商业经营都是至关重要的。

涉及安全方面的几个问题：

- 主要的因特网安全问题和威胁是什么？
- 协议的哪些技术方面会被罪犯所利用？
- 什么是安全的关键方面？
- 什么技术可用于增强安全性？

图30-1归纳了当前因特网存在的主要安全问题。

问　题	描　述
网络钓鱼	伪装为著名网站（如银行）以获取用户的个人信息（通常是账号和密码）
假冒	制造虚假或夸大的货物或服务要求，或者递交假冒伪劣产品
欺诈	意在欺骗动机用户去投资或协助犯罪的各种形式的圈套
拒绝服务	有意阻塞特定网站，以阻止或利用商务活动和贸易
失控	入侵者获取计算机控制权，并用此计算机进行犯罪
数据丢失	丢失知识产权或其他独立拥有的有价值的商业信息

图30-1 因特网上存在的主要安全问题

我们在考虑安全问题时，有一点很重要，就是必须区分开使用因特网意外犯错的传统犯罪与专门利用因特网犯罪的情况。例如，罪犯使用VoIP与同伙进行通信联络，罪犯进行国际欺骗和诈取受骗者钱财的圈套，或是罪犯使用因特网定制犯罪工具的犯罪事件等。虽然执法部门必须处理这些犯罪，但这些情况与网络技术本身几乎没有什么关系——人们很容易找到其他的通信机制来替代因特网的作用。在因特网上进行的最广泛传播的两种犯罪是意外地使用因特网的一般犯罪，即以拍卖方式提供假冒货物是虚假广告的一种形式，通过拍卖所购的货物不能递交是类似于常规的邮购欺骗。

我们的讨论将专注于罪犯所利用的技术手段，以及使网络犯罪更困难和更高成本而创建

的相关技术上。图30-2列举了攻击者所用的特殊技术：

攻击技术	描　述
窃听	复制网络传输中的数据分组，以获取信息
重放	不新发送捕捉到的先前会话数据分组（如先前登录的口令分组）
缓冲区溢出	发送超过接收缓冲区容量的数据以便在缓冲区边界外存储值
地址欺骗	伪造IP源地址从而欺骗接收者去处理虚假IP包
域名欺骗	利用形似的拼写，冒充知名域名或改改域名服务器的地址绑定
拒绝服务和分布式拒绝服务	以泛洪方式攻击网络服务器以阻止其提供正常的服务
SYN洪泛攻击	发送一连串建机TCP SYN片段，从而耗尽服务器TCP连接
密钥破解	自动猜解密钥或口令，以获得非授权的数据访问
端口扫描	查询连接目标主机的各协议端口，以寻找攻击漏洞
数据包拦截	从网络网上攻击数据包，以便允许暗失攻击和中间人攻击

图30-2　攻击者常用的攻击技术

窃听（wiretapping）和重放（replay）是众所周知的技术，就不必解释了。缓冲区溢出（buffer overflow）是常出现的计算机系统缺陷，它是粗糙软件开发所造成的一种症状：当执行数据输入操作时，程序员没有认真检查缓冲区大小。典型的缓冲区溢出攻击行为，就是发送一个庞大的数据包（超出缓冲区设计大小），或是发送一连串密集分组从而导致输入缓冲区溢出。

欺骗（spoofing）攻击被用于冒充一个受信任的主机。最简单的地址欺骗形式就是利用ARP——攻击者广播一个ARP响应包，将一个任意的IP地址A绑定到攻击者的MAC地址上，这样任何主机向A地址发送的IP包都将送到攻击者那里。其他形式的欺骗包括：使用路由协议发送不正确的路由信息，发送导致DNS服务器中产生错误域名和IP地址绑定的DNS报文，以及使用一个与著名网站域名非常相似的域名以误导用户，以为他是在访问一个被信任的网站。

拒绝服务（DoS）攻击是将大量数据包以泛洪方式冲击主机（通常是Web服务器），即使该服务器能够继续运行，但攻击将使其消耗大量资源，意味着其他用户的正常服务时延增大或请求被拒绝。由于服务器管理员可以侦查来自单个源的攻击包并使之无效，于是又出现了分布式拒绝服务（DDoS）攻击，其做法是由攻击者操控大量遍布因特网的主机发送IP包，如图30-3所示。通常，攻击者首先控制因特网上的一批主机并加载运行相关软件，然后使用这些受控主机去攻击服务器。因此，DDoS的攻击包都不是直接来自攻击者主机的。

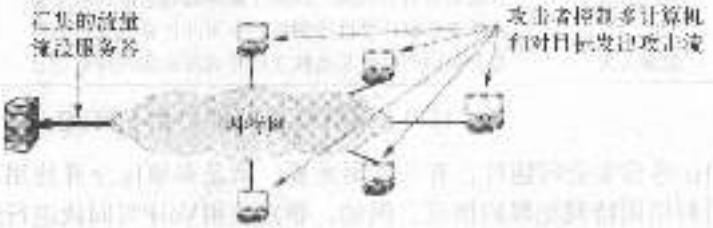


图30-3　分布式拒绝服务攻击示意图

SYN泛滥（SYN flooding）是针对TCP连接拒绝服务的特殊技术——它使每个接收进来的分组都包含一个TCP SYN报文，请求建立新的TCP连接。接收方对每个连接都要分配一个TCP控制块，发送SYN+ACK，然后等待连接请求方的响应。最终，导致接收方所有的连接控制块耗尽，再也无法打开TCP连接。

包截获（packet interception）可能导致“中间人”攻击，就是在源点向目的点传送的过

程中数据包被“中间人”截取并修改。虽然“中间人”攻击是在工程上最难实现的一种攻击，但它潜在的危害性却是很大的。图30-4是这种攻击的示意图。



图30-4 “中间人”原理及其发起的攻击

30.3 安全策略

什么是安全的网络？虽然安全网络的概念一直都在吸引着大多数的用户，但我们不能简单地分类为安全的或不安全的网络，因为“安全”这个含义不是绝对的——每个单位都可以定义允许或拒绝的访问等级。例如，公司为了保护有价值的商业信息，拒绝外部人员访问本公司的计算机；一个拥有可用信息网站的单位，可以将安全网络定义为可任意访问数据但禁止外部人员修改数据；也有一些单位注重通信机密性，他们将安全网络定义为除了发送者或真正的接收者外都不能获取和阅读报文。最后，对于一个大型组织而言，对安全的定义可能比较复杂，既允许访问有选择的数据和服务，又禁止访问和更改敏感或机密的数据与服务。

由于“安全网络”没有绝对的定义，因此为了实现安全系统，一个单位首先必须采取的步骤是制定安全策略。所谓的策略并不是规定如何实现保护，而是要清晰地、无二义性地阐明将要保护的各个条目。

安全策略由于涉及到人的行为、计算机以及网络设施而变得非常复杂（例如，把闪存ROM带出单位的来访人员，无线网络信号在单位建筑物以外被接收，或是在家办公的员工）。评估策略实现所需的投入和收益，同样也会增加策略本身的复杂性。特别需要指出的是，如果单位不能很好地理解自身信息的价值，则安全策略是不可能定义出来的。在很多情况下，信息的价值也是很难评估的。例如，一个包含员工档案、工作时间和薪水等级的简单工资数据库系统，假如员工可以访问工资数据库，一些员工就可能产生不满，要求更高的薪水或者威胁离职等；假如公司的竞争对手得到这些数据，他们可以依此引诱公司职员跳槽，更可怕的是竞争者利用这些信息来做预想不到的事情（例如，评估公司花费在某个项目上的投入等）。

概括：

> 制定网络安全策略可能是一项复杂的工作，因为合理策略要求一个单位把网络和计算机安全与使用者的行为、信息价值的评估联系起来。

由于每个单位必须考虑保护的侧重点，所以定义安全策略也是复杂的事情，而且要在安全性和易用性之间作出权衡。例如，一个单位需要考虑以下几个方面：

- 数据完整性（data integrity）。完整性是指要防止数据被改变，即到达接收方的数据是否与发送出来的数据完全相同？
- 数据可用性（data availability）。可用性是指要防止服务受到破坏，即对于合法的使用是否能保持数据的可访问性？
- 数据机密性（data confidentiality）。机密性是指要防止未授权的数据访问（例如，通过偷窥或窃取），即数据是否能防止非授权的访问？

- 私密性（privacy）。私密性是指发送者保持其匿名身份的能力，即发送者的身份是否会被泄露出去？

30.4 安全责任与控制

除了上述几项外，一个单位还必须正确地规定对信息的安全责任如何分派或控制。对信息的责任问题，包含以下两个方面：

- 会计责任（accountability）。会计责任是指如何保留审计踪迹，即哪个小组对哪项数据负有责任？如何保留各个小组对数据进行访问和修改的记录？
- 授权（authorization）。授权是指对每个信息项的责任，以及如何把这样的责任委派给他人，即由谁来负责决定将信息存储在那里，以及负责人如何审批访问和修改权限？

会计责任和授权的关键点是控制（control）问题——一个单位必须控制对信息的访问，这与一个单位对使用有形资产（如办公楼、设备和供给等）要加以控制相类似。控制的关键又是认证（authentication），即怎样确定身份。例如，假定一个单位详细制定了一套给予单位员工比普通来访者更高权限的授权策略，那么除非该单位具有一套区分本单位员工和普通来访者的认证机制，否则其授权策略是毫无意义的。认证对象除了人以外，还可扩展到计算机、设备和应用软件。

要点　如果没有能够明确地检验请求者身份的认证机制，那么授权策略就是毫无意义的。

30.5 安全技术

目前，已经有许多安全产品为单台和一组计算机提供各种各样的安全功能。图30.5归纳了这些产品所用的技术，我们在后面将逐一对这些技术进行介绍。

技　术	目　的
散列法（检查）	数据完整性
加密	私密性
数字签名	对消息发送
数字证书	发送者认证
防火墙	网站完整性
入侵检测系统	网站完整性
深度包检查与内容扫描	网站完整性
虚拟专网（VPN）	数据私密性

图30.5　用于执行安全策略的主要技术

30.6 散列法：完整性与鉴别机制

前面的章节已经讨论了用于保护数据不受偶然性破坏的技术，如奇偶位、校验和以及循环冗余校验（CRC）等。这些技术并不能保证数据的完整性，其中有两个原因：第一，如果硬件故障同时改变了检验值以及数据的值，这就有可能使改变后的检验值与改变后的数据值正好使校验有效；第二，如果数据的改变是有意攻击所造成的，那么攻击者就能够使被改变的数据仍能校验有效。于是，人们就创造了其他机制以确保消息的完整性不受有意攻击而改变。

一种方法是使用攻击者不能破解或伪造的消息鉴别码（Message Authentication Code，

MAC)，其典型编码方案是采用密码散列（cryptographic hashing）机制。而散列方案则是依赖于只有发送方和接收方才知道的密钥（secrete key）。发送方将消息作为散列函数的输入，利用密钥计算密码散列函数H。发送方将密码散列函数H随同消息一起发送。H一般是一个较短的字符串，且其长度与消息长度无关。接收方使用同样的密钥来计算接收消息的散列值，然后与H进行比较。如果两者一致，则表示接收的消息是未被篡改的。对于没有密钥的攻击者，则不能更改消息，也不能引入差错。因此，H提供了一种消息鉴别机制，因为接收者知道，只有伴随一个有效散列值而到达的消息才是可信的。

30.7 访问控制与口令

访问控制（access control）机制是要控制哪些用户或计算机程序可以访问数据。例如，有些操作系统对每个对象实行一个访问控制表（Access Control List, ACL），以确定谁被允许访问该对象。在另一些系统中，为保护每个资源，对每个用户分配一个口令（password），当用户需要访问一个受保护的资源时，要求用户输入口令。

如果将这种访问控制表和口令机制推广到网络上应用时，就必须采取必要的步骤以防止无意的泄露。例如，如果在某个地点的一个用户要通过网络发送未加密的口令字到另一个地点的计算机上，窃听网络的任何人就能够获取该口令的副本。当通过无线局域网传输数据时，窃听就更容易，因为它不需要物理连接即可实现——在传输信号辐射范围内的任何人都能得到每个分组的副本。除此以外，还必须采取必要的步骤确保口令是不容易猜测的，这为网络允许攻击者可自动尝试破解口令。因此，管理员要强化选择口令的规则，例如，制定最小口令长度和禁止使用常用的词（如可在字典中找到的词）。

30.8 加密：基本的安全技术

因为密码技术可以保证数据的机密性（有时也称为私密性）、消息认证、数据完整性和阻止重放攻击等，所以加密是安全领域里一种最基本的工具。实质上，发送者是应用加密方法将消息位元扰乱。这样只有期望的接收者才可以对加密消息进行解扰，即使加密消息的副本被截取，也无法从中提取信息。而且，一个加密的消息还可以包含诸如消息长度等的信息，这样攻击者也就不可能剪裁消息而不被发觉。

用于加密的4个术语是：

- 明文——加密之前的原始消息。
- 密文——加密之后的消息。
- 加密密钥——用于加密消息的一个短字符串。
- 解密密钥——用于解密消息的一个短字符串。

我们将看到，在一些加密技术中，加密密钥和解密密钥可以是相同的，也可以是不同的。

在数学上，可以把加密看作一个函数encrypt，它有两个参数：密钥K和将被加密的明文消息M。函数的输出就是该消息的加密形式，即密文C：

$$C = \text{encrypt}(K, M)$$

解密函数decrypt是产生原始消息的逆映射：

$$M = \text{decrypt}(K, C)$$

在数学上，解密其实就是加密的逆过程：

$$M = \text{decrypt}(K, \text{encrypt}(K, M))$$

30.9 私有密钥加密

虽然已有许多加密技术，但从使用密钥的方法上可以分为两大类：

- 私有密钥。
- 公开密钥。

在私有密钥（private key）系统中，每一对通信的实体共享一个密钥，即这个密钥既是加密密钥又是解密密钥。这个密钥必须是保密的——如果有第三方获取了该密钥的副本，它就可以解密通信双方传递的消息。私有密钥系统是一种对称的系统，其中通信的每一边都能收发消息。为了发送消息，其密钥可用于产生密文，并通过网络传输；而密文到达接收端时，该密钥又将用于解密密文，并提取原始消息（明文）。因此，私有密钥系统中收发双方使用相同的密钥K，其函数关系表达为：

$$M = \text{decrypt}(K, \text{encrypt}(K, M))$$

30.10 公开密钥加密

另一类主要的加密体系称为公开密钥加密（public key encryption）。公开密钥加密系统给每个实体分配一对密钥。为了便于讨论，我们假设每个实体是单个用户，用户持有一个密钥称为私有密钥（private key），它是保密的；还有另一个密钥称为公开密钥（public key），它随同用户名一起都是公开的，所以大家都知道这个密钥的值。该技术的加密函数具有如下数学特性：用公开密钥加密的消息除非使用相应的私有密钥外不能被解密；同样，用私有密钥加密的消息除非使用相应的公开密钥外也不能被解密。

这种用两个密钥的加密与解密之间的关系可以表示成数学形式，我们用M表示明文，public_u1表示用户1的公开密钥，private_u1表示用户1的私有密钥，那么公开密钥加密函数可表示为：

$$M = \text{decrypt}(\text{public\_u1}, \text{encrypt}(\text{private\_u1}, M))$$

和

$$M = \text{decrypt}(\text{private\_u1}, \text{encrypt}(\text{public\_u1}, M))$$

通过展现对每个方向发送消息加密的密钥，图30-6说明了为什么公开密钥体系会被归类为非对称加密系统。



图30-6　公开密钥加密系统的非对称性示意图

即使公开密钥被泄露了也是安全的，因为加密和解密的函数具有单向特性。也就是说，仅知道了公开密钥并不能伪造出相应私有密钥加密过的消息。公开密钥加密可用于保证机密性。希望通信中保持消息机密性的发送方要使用接收方的公共密钥去加密待发消息；而通过网络传输过程中获得的密文副本，则无人能读懂该报文的内容，因为解密必须要有接收方的私有密钥。由于只有接收方可以解密消息，因此公开密钥方案确保了传输数据的机密性。

30.11 用数字签名的鉴别

加密机制还可以用于鉴别消息的发送方，这种技术称作数字签名（digital signature）。若要在一个报文上签名，发送方只要使用只有自己知道的密钥对报文加密即可。[9]接收方使用逆函数对报文进行解密，接收方知道是谁发送的报文，因为只有发送方持有执行加密的密钥。为了保证加密的报文不被复制和稍后重传，原始报文中可包含创建该报文的日期和时间。

公开密钥系统是如何用于数字签名的呢？若要给报文签名，发送者可用他（她）的私有密钥对报文加密；如要验证签名，接收方就要查找该用户的公开密钥并用它对报文解密。这是因为只有发送者知道该私有密钥，所以也只有发送者才能产生能够被公开密钥解密的报文。

有趣的是，对一个报文加密两次就可以保证报文的可鉴别性和机密性。首先，使用发送者的私有密钥对报文签名做一次加密。然后，再用接收者的公开密钥对已加密的报文进行再次加密。在数学上，这两次加密步骤可表示如下：

$$X = \text{encrypt}(\text{public\_u2}, \text{encrypt}(\text{private\_u1}, M))$$

其中，M表示原始报文，X表示经两次加密后得到的密文，private_u1表示发送方的私有密钥，public_u2表示接收方的公开密钥。

在接收端，解密过程是加密的逆过程。首先，接收方用他（她）的私有密钥对报文进行解密，这样就去除了一个加密层次，但还遗留了报文被数字签名的加密层次。然后，接收方再利用发送方的公开密钥对报文进行解密。这个过程可以表示为：

$$M=\text{decrypt}(\text{public\_u1}, \text{decrypt}(\text{private\_u2}, X))$$

其中，X表示经过网络传输的已加密位串，M表示原始报文，private_u2表示接收方的私有密钥，public_u1表示发送方的公开密钥。

如果一个有意义的报文是由以上两个步骤产生的，那这个报文就一定是可信的和可靠的。这个报文必能到达期望的接收方，因为只有他（她）才拥有正确的私有密钥以便解开外层加密。同时，该报文也必定是经过鉴别的，这是因为只有该报文的发送方才拥有私有密钥对它进行了加密，因而用发送方的公开密钥才能正确地对它解密。

30.12 密钥分发和数字证书

围绕公开密钥技术讨论的基本问题之一，就是关于获取公开密钥的方法问题。虽然也可以采用传统发行的方法（类似电话簿的形式），但这样做需要手工将密钥输入计算机，所以比较麻烦而且可能出错。于是问题又产生了：能否设计出一种自动系统来分发公开密钥呢？当然，这个分发系统必须是安全的——如果分发给用户的公开密钥是错误的，那么安全性就没有保障，进一步的加密也将是不可信的。这就是所谓的密钥分发问题（key distribution problem），如何形成一个切实可行的密钥分发系统，已经成为扩展公开密钥系统适用性的主要障碍。

目前，已经提出了几种密钥分发机制，其中包括一种使用类似域名系统的方法。每种解决方法都是基于一个简单的原理：只要用户知道一个密钥——密钥权威机构的公开密钥——从密钥权威机构那里就可以安全方式获得任何其他人的公开密钥。因此，系统管理员只需要配置一个公开密钥，如图30-7所示为一个用户决定要与一个新的网站W交互时的报文交换过程。

---

[9] 如不要求机密性，报文就不需要加密，也可采用更有效的数字签名来加密报文的摘要（散列）值。

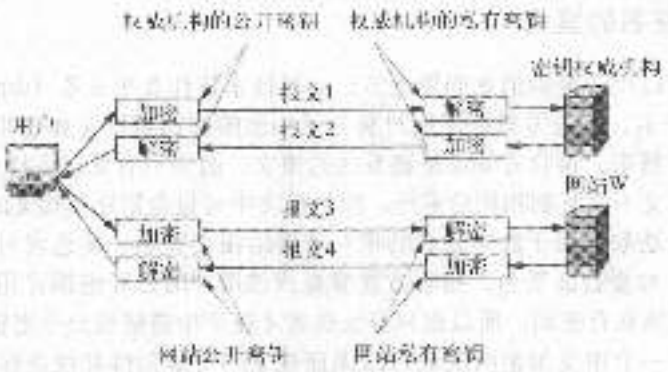


图30-7　使用密钥权威机构来获得公开密钥的示意图

在图30-7中，用户欲与网站W进行一次安全的事务处理，4个被传递的报文都是机密的。报文1使用了密钥权威机构的公开密钥[⊖]进行加密，只有密钥权威机构可以阅读；报文2必然是由密钥权威机构产生的，因为只有密钥权威机构才拥有与该公开密钥相匹配的私有密钥；一旦用户获得网站W的公开密钥后，用户即可向网站W发送保密的请求，并获得只有网站W才能产生的响应（因为只有网站W才有该私有密钥）。

显然可能有许多的变化形式，但其重要原理都是：

只要构建一种只要求人工配置一个公开密钥的安全密钥分发系统，就可以了。

30.13　防火墙

虽然加密技术有助于解决许多安全问题，但还是需要第二种技术。一种称为因特网防火墙（Internet firewall）的技术能帮助防止那些不想要的因特网业务进入单位的计算机和网络。与传统的防火墙相似，因特网防火墙设计成能让因特网上的问题不致于扩散到本地内部的计算机上。

防火墙放置在本单位网络与外部的因特网之间，所有进入或离开本地网络的分组都要通过防火墙。其系统结构如图30-8所示。

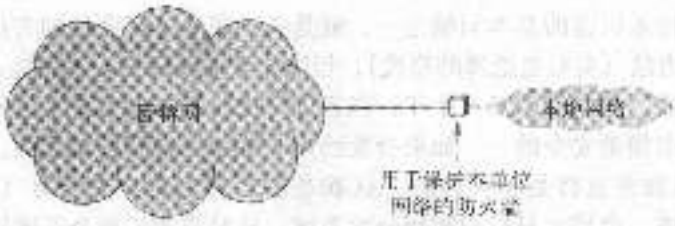


图30-8　部署在因特网与本地网络之间选择点的防火墙示意图

如果本单位有多个因特网连接，那么在每个连接上都要设置一个防火墙，而且这些防火墙都必须按照本单位制定的安全策略进行配置。另外，防火墙本身也必须是安全的，以免造成“回火”（tempering）现象。概括：

- 进入单位内部的所有数据流都必须通过防火墙。
- 离开单位出去的所有数据流都必须通过防火墙。

⊖　即W知道其公开密钥。——译者注

- 防火墙要实现安全策略，并丢弃那些不遵循策略的任何数据分组。
- 防火墙本身应具有免受安全攻击的能力。

防火墙是用来对付互不信任单位之间网络连接的最重要的安全工具。通过在每个外部网络连接上设置防火墙，本单位可以定义一个安全边界（secure perimeter），防止外界对单位内部计算机的干扰。特别是，防火墙能防止外界用户发现单位内部的计算机，防止那些不需要的业务流涌入到本单位网络中来，或者防止通过发送一系列IP数据报来攻击计算机而引起计算机失常（如崩溃）。而且，防火墙还能防止不良数据的输出（例如，内部用户在发送磁盘文件副本给外部用户时，无意中输入一个病毒）。

相对其他的安全方案，防火墙有一个关键的优点：它实施集中控制，因而极大地改善了安全性。为了安全而又不设置防火墙，单位就必须使所有的计算机获得安全保障，而且每台计算机必须执行相同的安全策略。这样，雇用人员来管理很多计算机的费用也就提高了，单位也不可能依靠个人用户正确地配置他们的计算机。设置防火墙后，管理员可以限制所有的因特网业务到达其一小范围内的计算机上，通过配置和监测来允许使用特定的业务。在极端的情况下，可以让所有的外部访问只集中在一台计算机上。因此，防火墙能使一个单位节省开支并取得良好的安全性。

30.14 包过滤防火墙的实现

虽然可以是一个独立的设备，但实际上大多数防火墙都是作为一个模块嵌入到交换机或是路由器中。无论防火墙是哪种形式，用于构建防火墙的基础技术就是所谓的包过滤（packet filtering）。过滤器由一套可配置的机制组成，它检查数据包的头部参数域来决定是否让它通过或将它丢弃。管理员配置包过滤器时，要规定好在各个方向上那些包可以通过。（与其规定哪些包不能通过，倒不如规定哪些包可以通过，这样才具有更好的安全性。）

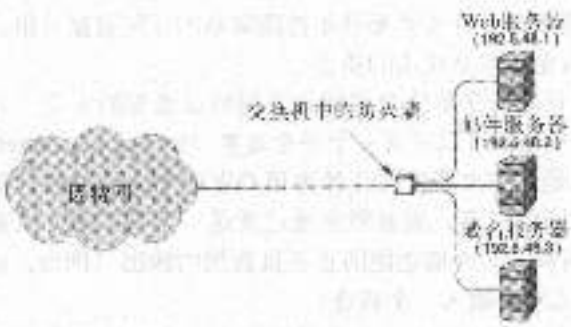
对于TCP/IP，包过滤器通常规定要检查帧类型0800（对于IP），IP源地址或目的地址（或两者），数据报类型以及协议端口号等。例如，如果允许外部用户访问本单位的Web服务器，那么包过滤器就必须允许含有任意源地址、目的地址为内网Web服务器的IP包、任意源端口、目的端口为80的TCP数据包通过。

由于包过滤器允许管理员指定数据包的目的地址、源地址和服务端口的组合，所以防火墙中的包过滤器允许管理员控制对特定计算机上某项服务的访问。例如，管理员可能选择允许外部业务去访问一台计算机上的Web服务器，另一台计算机上的邮件服务器和第三台计算机上的DNS服务器。当然，管理员还必须安装有关的防火墙规则，规定那些响应数据包被允许流出站点。在图30-9中表示了这样一个网站的防火墙配置。

这种有选择地允许特定服务的数据报通过的能力，意味着管理员可以小心地控制那些外部可见的服务。这样，即使用户无意地（或有意地）在自己计算机上启用了一个邮件服务器，但外部计算机也是不可能连接该服务器的。

概括如下：

> 防火墙利用包过滤技术来防止不希望有的通信交互。每种过滤器规则都要给出数据包头部参数的组合，包括IP源地址、目的地址、协议端口号和传输协议类型号。



Dir	Frame Type	IP Src	IP Dest	IP Type	Src Port	Dst Port
in	0800	*	192.5.48.1	TCP	*	80
in	0800	*	192.5.48.2	TCP	*	25
in	0800	*	192.5.48.3	TCP	*	53
in	0800	*	192.5.48.3	UDP	*	53
out	0800	192.5.48.1	*	TCP	80	*
out	0800	192.5.48.2	*	TCP	25	*
out	0800	192.5.48.3	*	TCP	53	*
out	0800	192.5.48.3	*	UDP	53	*

图30.9　有3个服务器的网站防火墙配置示例。表中的星号表示任意值的通配符

30.15　入侵检测系统

入侵检测系统（Intrusion Detection System，IDS）对每个进入网站的数据包进行检查，如果发现危害安全则向系统管理员报警。IDS提供一种安全告警的附加层——即使防火墙对攻击进行了阻止，IDS还会对事件的发生通知网站管理员。

大多数入侵检测系统都能被配置成检测具体不同类型的攻击。例如，IDS可以用来检测端口扫描攻击。在攻击过程中，攻击者对逐个UDP协议端口发送UDP数据报，或者对逐个TCP协议端口不断尝试开启TCP连接。类似地，IDS也可以配置成通过观察来自同一源地址的重复TCP SYN请求，来检测可能的SYN泛洪攻击。在一些情况下，入侵检测系统还可以与防火墙互连以提供自动过滤——对出现的问题不只是通知网站系统管理员，而是由IDS产生防火墙规则，自动地阻止造成攻击的数据包。例如，如果入侵检测系统检测到来自某源地址的SYN泛滥攻击，其IDS即刻安装防火墙规则以阻止来自该源地址数据包引起的问题。采用这种自动联动方法的理由是为了提高速度——从通知出现攻击到作出响应，一般需要花许多秒的时间，而在千兆网络上每秒钟有超过50 000个IP包到达。因此我们需要更快速的响应来阻止攻击，以免演变成灾害。

入侵检测系统与防火墙的主要区别在于，IDS包含状态信息（state information），它不是像防火墙那样一次只对一个数据包应用规则，IDS可以保留数据包历史记录，因此虽然防火墙可以决定是否过滤固定源的SYN攻击包，但是入侵检测系统可以观察分析来自某个源地址的许多SYN包。当然，入侵检测系统相对于防火墙而言，需要更多的存储单元和计算量，因此IDS每秒钟不能处理很多的数据包。

30.16 内容扫描和深度包检查

防火墙虽然可以处理许多安全问题，但它也存在严格的限制：只能对数据包头部进行检测，也就是说，防火墙不能检测数据包的载荷。为了理解为什么数据包内容是重要的，不妨考虑一下计算机病毒。病毒受入本地网络和主机的最常见方法之一，就是通过邮件附件——攻击者将计算机程序作为附件和电子邮件报文一起发送。如果无警觉的用户打开了这个附件，就会自动地执行程序并在计算机上安装有害软件，例如恶意软件（malware）。

网站怎样防止类似病毒传播的安全问题呢？那就是进行内容分析（content analysis）。目前内容分析有两类方法：

- 文件扫描。
- 深度包检查（DPI）。

文件扫描（file scanning）——分析文件内容最简单的方法就是在整个文件上进行操作。文件扫描也是一种大家熟知的技术，安装在普通PC机中的安全杀毒软件就采用这种技术。从本质上来说，文件扫描器把文件作为输入，对它查寻包含安全问题的字节特征。例如，许多病毒扫描器查寻称为指纹（fingerprint）的字节串。也就是说，销售杀毒软件（病毒扫描器）的公司收集各种病毒样本，把每个样本放在一个文件中，然后发现和通常不一样的特征字节串，并建立特征库。当一个用户运行病毒扫描器软件时，该软件搜索用户磁盘上的文件，然后用特征库中的每个条目与文件字节匹配，看看是否一致。文件扫描器可以有效地捕捉常见的安全问题。当然，如果普通文件中巧合的具有特征库的特征串，文件扫描器也可能发生错报（假阳性，false positive）；如果病毒是新的，特征库还没有该病毒的任何特征串，就会发生漏报（假阴性，false negative）。

深度包检查（Deep Packet Inspection，DPI）——内容分析的第二种方法就是对数据包而不是对文件进行处理。也就是说，DPI机制不是只对进入网站的数据包头部进行检查，而且还要对数据包载荷中的数据进行检查。需要注意的是：DPI同样也要检查头部的内容——在很多情况下，如果不对数据包头部的域进行检查，往往就不能对载荷的内容作出解释。

作为DPI的一个例子，不妨考虑攻击者采用相似拼写的虚假域名来欺骗用户以为在访问可信网站的情况。某个单位如果想要防止这种攻击的发生，可以将该一系列被认为有危险的URL列入黑名单（black-list），还有一种使用“代理”的方法，它要求网站内的所有用户要将浏览器配置成为使用Web代理（Web proxy）（即在获得被请求网页之前要检查URL的一种中间Web系统）。另有一种可选的方法是，DPI过滤器可以设置为检查每个向外流的数据包，将HTTP请求的URL与黑名单站点逐一核对。

深度包检测的主要不足就是需要较大的计算量。因为一个以太网帧中的数据包载荷是一个IP包头部的20倍以上，DPI可能需要比一般包头检查多花20等的处理时间。而且，IP包载荷并不是固定长的域，这就意味着DPI机制必须在检查的过程中不断地分析载荷的内容。得出结论：

> 由于DPI需要检查比包头部大得多的数据包载荷，且这种载荷域不是固定长度的域，所以深度包检测机制只能限制应用在低速网络中。

30.17 虚拟专网

有一种重要且广泛应用的安全技术，即使用加密来提供从远地站点通过因特网对单位内联网（Intranet）的访问，这就是所谓的虚拟专网（Virtual Private Network，VPN）。该技术起初设计作为一个公司分布在不同地理位置上的站点间提供低成本的互连。为了理解这样的动

机。我们先考虑以下两种可选的互连方案：

- 专用网络连接。公司租用数据线路连接它的各个站点，每个站点都通过路由器连接租用线路到另一个站点的路由器上，数据直接在站点路由器之间传输。
- 公共因特网连接。每个站点与当地的ISP签订因特网服务合同，数据通过全球因特网在公司站点之间传输。

如图30-10所示为一个具有3个站点的公司其可能的互连方案。

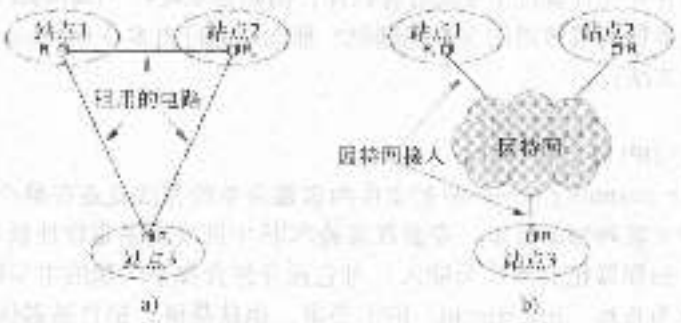


图30-10　a)使用租用线路连接站点；b)通过因特网连接站点

使用租用线路来互连站点的主要优点在于：所形成的网络是完全专用的（private）（即具有机密性）。因为没有其他单位接入这种租用线路，所以其他单位也就不能读取站点间传输的数据。采用因特网连接的主要优点是成本低——它不需要为每一条连接线路付费用，只需要为每个站点付因特网服务费。可惜的是，因特网不能保证机密性。数据报从源点传输到目的点的过程中，它所经过的中间网络可能是共享的，因而外界用户就有可能获取数据报副本并检查其内容。

虚拟专网VPN正好组合了以上两种方式的优点，通过利用因特网在站点间提供数据传输，并采用其他方式确保传输的数据不被外界访问。也就是说，VPN不采用昂贵的租线连接，而是采用加密技术——将本单位站点之间待传递的所有数据包，在发送之前进行加密。

为了使对虚拟专网的攻击不受影响，单位甚至可以选用专用路由器来实现VPN功能，并使用防火墙来禁止VPN路由器接收任何非授权的数据包。例如，假设图30-10b中的每个路由器都是专门提供VPN功能的（即假设站点另有其他的路由器来处理通常的进出因特网业务）。保护站点1的VPN路由器的防火墙可以限制所有进入的数据包必须具有站点2和站点3的VPN路由器的IP源地址。类似地，其他站点的防火墙也将进入数据包限制其具有特定站点的IP源地址。这样，就使虚拟专网有效地防止地址欺诈和DoS攻击。

30.18　VPN技术应用于远程办公

VPN的初衷是为提供站点之间的互连而设计，但它在远程办公（telecommute）的雇员之间使用已被为流行。VPN有两种实现形式：

- 独立VPN设备。
- VPN软件。

独立VPN设备（Stand-along Device）。这是单位分派给雇员的一个物理联网设备，有时也称为VPN路由器。该设备与因特网连接，自动建立与本单位站点的VPN服务器之间的安全通信，并且提供LAN端口来连接用户的计算机和IP电话。在逻辑上，VPN设备将一个公司的网络扩展到用户居住点，使连接在VPN设备上的计算机在操作上与连接在公司网络上的计算机

一样。因此，在用户计算机启动引导获取IP地址时，其地址将由单位的DHCP服务器分配。相类似，用户计算机中的转发表的建立也与在单位本地的计算机一样——在VPN连接的计算机发送IP包的任何时候，VPN设备对IP包进行加密，并以加密的形式通过互特网传递到单位；在收到来自单位的IP包的任何时候，VPN设备将该IP包解密，并把解密后的IP包传送到用户计算机。

VPN软件（VPN Software）。对于在家里或在远地办公室工作的雇员来说，独立的VPN设备工作无可挑剔，但是这种设备对于出差流动人员来说，却感到很麻烦。为了解决这种情况，公司可选用运行在用户计算机上的VPN软件。用户先接入互特网，然后启动VPN软件程序。一旦启动运行后，VPN软件就将自己插入到因特网的特定连接中。亦即，VPN软件将截获所有进入和送出的数据包；对每个送出的IP包加密并将加密后的数据包传送到公司的VPN服务器；对每个进入的IP包进行解密。

30.19 数据包加密与隧道技术

以上VPN的讨论引出了一个感兴趣的问题：如何对通过因特网传输的数据进行加密？有3种可选择采用的技术：

- 载荷数据加密。
- IP-in-IP隧道技术。
- IP-in-TCP隧道技术。

载荷数据加密。为了保持数据报内容的机密性，载荷加密方法就是对数据报的载荷域数据进行加密，但不要触及头部的内容。由于数据报头部是未加密的，所以外界用户能够知道正在使用的源地址、目的地址以及协议端口号。例如，假设财务总监（CFO）处在一个地点，公司经理处在另一个地点，进一步假设每当有利好财务消息时财务总监就要发送一个简短电子邮件给经理，而每当有差的财务消息时就会发送一个详细的解释。外界用户就可能会观察到这两个特殊计算机之间有数据报流动后不久，公司的股票价格就上涨了。

IP-in-IP隧道技术。当数据报通过因特网从一个站点传输到另一站点时，有些VPN是采用IP-in-IP隧道方法将头部信息隐藏起来。也就是说，当向外发送数据报时，发送方VPN软件对整个数据报进行加密（包括头部），并将结果装入另一个数据报里面进行传输。例如，如图30-10所示的连接情况，假设站点1的计算机X产生一个数据报要发送给站点2的计算机Y。该数据报通过站点1转发到路由器R1（即连接站点1到因特网的那个路由器），R1上的VPN软件功能对原始数据报加密处理，封装成新的数据报传输到站点2的路由器R2。当封装的数据报到达R2后，R2的VPN软件解密其载荷，提取出原始数据报，然后就转发给计算机Y。如图30-11所示为封装的过程。

在图30-11a表示原始数据报，图30-11b表示数据报被加密后的密文，图30-11c表示从R1送往R2的输出数据报。需要注意的是：因为站点1和站点2之间通过因特网传递的所有数据报都是使用R1和R2的源地址和目的地址，所以站点1、2的内部IP地址就被隐藏起来。

概括：

当VPN采用IP-in-IP封装时，原始数据报的所有域（包括头部）都被加密了。

IP-in-TCP隧道技术。第三种用于保护数据机密性的可能的方法是采用TCP隧道技术，也就是，双方先建立TCP连接，然后使用该连接发送加密的数据报。当要发送数据报时，整个数据报被加密，再加上一个小头部用于标记数据报之间的分界点，然后通过TCP连接发送出

去。通常，附加的头部由两字节整数组成，它指定数据报的长度。在TCP连接的另一端，接收VPN软件先读出头部，然后读出附加字节的特殊数值，从而获取加密的数据报，一旦接收方得到完整的数据报密文后，就将其解密，即可获得并处理原始数据报。



图30-11　用于VPN的IP-in-IP封装示意图

与IP-in-IP隧道相比，IP-in-TCP隧道技术的主要优点是可靠递交，即TCP确保两个站点之间发送的所有数据报都能可靠地到达和按序接收。采用IP-in-TCP隧道技术的主要问题是存在线头阻塞（head-of-line blocking）——由于数据报必须按序递交，假如有一个TCP段丢失或延迟，TCP就不能从后继段继续递交数据，即使收到的数据都是正确的。如果我们把VPN认为是在传送一个分组队列，那么在第一个数据报成功递交前，整个分组队列都会处在阻塞状态。

关于VPN隧道的最后一个问题：性能。它包括以下3个方面：

- 迟延。
- 吞吐量。
- 开销和分片。

迟延（latency）。为了理解迟延问题，让我们考虑美国西海岸的一个公司，并假设公司的一个雇员出差到东海岸，相距大约3000mile（1mile=1609.344m）。记住，VPN软件只是将数据报传回母公司——一旦数据报到达公司，必须将它转发到目的主机。例如，该雇员要浏览自己公司的网页，每个请求必须从该雇员当前地点发向公司VPN服务器，然后再从那里转发到Web服务器，相应的应答也必须先传回公司的VPN服务器，最后再到达远地点的雇员处。当雇员要访问就近的资源时，造成的迟延特别大，因为数据报必须从雇员所在地通过VPN到西海岸的公司，再返回东海岸的资源所在地，结果该数据报的往返行程需要来回穿越大陆4次。

吞吐量（throughtput）。传统VPN的另一个问题是通过隧道的有效吞吐量。当要使用针对高速LAN而设计的应用软件时，吞吐量的问题显得非常重要。例如，在有些单位，职员所用的有关公司内部业务的网页中含有大量的图片。站点内的LAN可提供足够的吞吐量来快速下载这些网页。但对网络带宽的需求对于通过虚拟专网连接的远程用户来说，小吞吐量的VPN就会严重阻碍这种网页的下载。

开销和分片（overhead and fragmentation）。性能的第三个方面是隧道对数据报增加了额外开销。为了更好地理解这个问题，假设一站点使用以太网，以及应用软件产生的数据报长度为1500字节（亦即，数据报长度正好和网络的MTU一样）。当VPN路由器封装加密的数据报成为另一个IP数据报时，其实际输出的数据报至少将额外增加20字节，这样就会导致数据报超过网络的MTU，在传输前只能被分片。由于必须在两个分片都到达后才能进行该数据报的处理，所以，数据报产生迟延和丢失的概率也就更高了。

30.20 安全技术

目前，已经开发出各种各样的安全技术应用于因特网，大致包括：

- 电子邮件加密标准（Pretty Good Privacy，PGP）或称极佳隐密性，这是一种对传输前的数据进行加密的密码系统，由美国麻省理工学院开发，在计算机科学家中应用特别广泛。
- 安全外壳（Secure Shell，SSH），一种通过加密因特网传输前的数据来确保机密性的远程登录应用层协议。
- 安全套接层（Secure Socket Layer，SSL），一种最早由网景通信公司设计，利用加密来提供认证和机密性的技术。SSL软件适合配置在应用程序与套接口API之间，在数据进入因特网传输之前对它进行加密。SSL用于Web连接情况下实现用户与金融机构的安全交互（例如，发送信用卡号到Web服务器）。
- 传输层安全（Transport Layer Security，TLS），20世纪90年代后期由IETF设计的，作为SSL的换代技术。TLS建立在SSL第3版基础上，两者都可用于HTTPS。
- HTTP安全（HTTP Security，HTTPS），HTTPS不是真正的独立技术，它是HTTP协议与SSL或TLS和证书机制组合，为用户提供通过Web的认证、保密通信。HTTPS使用TCP端口号为443，而不是80。
- IP安全协议（IP Security，IPsec），IPsec用于IP数据报的安全标准，它采用密码技术，允许发送者选择认证（即验证数据报的发送方和接收方）或保密（即对数据报内容加密）。
- 远程认证拨入用户服务（Remote Authentication Dial-In User Service，RADIUS），这是一种用于提供集中认证、授权和计费的协议，普遍用于拥有拨号用户的ISP和用于提供远程用户接入的虚拟专网。
- 有线等效隐私性（Wired Equivalent Privacy，WEP）。WEP原是Wi-Fi无线局域网标准[⊖]的一部分，用于保证传输的机密性。后来伯克利加州大学的研究者们发现WEP存在几个弱点，于是就开发了名为Wi-Fi受保护访问（Wi-Fi Protected Access，WPA）作为替代技术。

30.21 本章小结

计算机网络和因特网可被用于犯罪活动，其主要的威胁包括：网络钓鱼、假冒、欺诈、拒绝服务、失控（僵尸机）和数据丢失等。所采用的攻击技术包括：链路窃听、重放、缓冲区溢出、IP地址和域名欺骗、利用数据包和SYN泛滥的拒绝服务、口令破解、端口扫描和数据包嗅探等。

每个单位都需要定义自己的安全策略，对多个方面作出规定，包括：数据完整性（保护数据不被修改）、数据可用性（保护数据服务不致发生混乱）、数据机密性或私密性（保护数据不被窃取和探察）。此外，单位还必须考虑可审计性（即如何保存好审计踪迹）和授权（即如何在不同人之间传送对信息所负的责任）。

人们已经开发出一系列有关技术以提供对各个方面的安全性，这些技术包括：加密、哈希散列、数字签名和证书、防火墙、入侵检测系统、深度包检测、内容扫描和虚拟专网等。其中，加密是最基本的，许多安全机制都要依赖于这个基础技术。

⊖ WEP应用于多种IEEE802.11协议。

私有密钥加密是利用单个密钥对数据进行加密和解密，加密者或解密者都必须保证该密钥的安全。公开密钥加密系统使用一对密钥，一个密钥是要保密的，另一个密钥（公开密钥）则可以公布。数字签名利用加密技术实现消息鉴别（或认证）。密钥权威机构可以通过分发证书来使公开密钥有效。

防火墙通过限制输入和输出的数据包的措施来保护网站免受攻击；为了配置防火墙，网管软件要设计一系列对数据包头部域给出特定值的规则。保留状态信息的入侵检测系统可以识别诸如重复SYN之类的攻击。

虚拟专网VPN可提供保密通信和低成本。该技术使远程办公成为了可能。为了保证信息的机密性，发送方可加密数据报载荷数据并采用IP-in-IP隧道或IP-in-TCP隧道技术。隧道技术的优点是对数据报头部和载荷部进行加密。但由于VPN造成的迟延较长、吞吐量较低和开销较大，所以有些应用并不适合通过VPN来运作。

还有很多其他的安全技术，例如，PGP，SSH，SSL，TLS，HTTPS，IPsec，RADIUS和WEP等。

练习题

30.1　列出因特网上存在的主要安全问题，并分别给出简要说明。

30.2　说出用于网络攻击的主要技术名称。

30.3　假设攻击者找到一种方法可以存储你本地DNS服务器中的“域名-IP地址绑定表”，那么攻击者如何利用这样的脆弱性来获取你的银行账户信息呢？

30.4　DoS攻击常常发送TCP SYN段，那么，攻击者是否也可以通过发送TCP数据段来实现DoS攻击？解释之。

30.5　如果一个用户口令是由8个字符组成的，其中含有大写字母、小写字母和数字，那么攻击者要破解口令可能需要尝试多少次？

30.6　为什么说定义一个安全策略不是一件容易的事？

30.7　假设一个公司要实现一种只有人力资源部门的人员才可以查看员工的薪资资料，那么实现这种策略需要什么机制？

30.8　列出和描述8种基本的安全技术。

30.9　什么是访问控制表（ACL）？如何使用ACL？

30.10　密码学（cryptography）是一门什么科学？

30.11　阅读数据加密标准（DES），对你其重要的数据加密的密钥应该采用多大长度？

30.12　假设你朋友有一个公开密钥和用于公开密钥加密的私有密钥，你朋友可以给你发送保密的消息（即消息只有你能阅读）吗？为什么？

30.13　假设你和朋友各有一对公开密钥加密系统的公开密钥和私有密钥，你和朋友如何才能保障每天进行通信而不会发生重放攻击的欺骗？

30.14　两方如何利用公开密钥加密来签署一个合同然后发送给第三方？

30.15　什么是数字证书？

30.16　什么是防火墙？防火墙一般安装在什么位置？

30.17　很多商业防火墙产品都允许管理员规定将拒绝哪些数据包和接受哪些数据包。允许“拒绝”的这种配置有什么缺点？

30.18　修改图30-9中防火墙的配置，让3个服务器都能被外界ping得通。

30.19　修改图30-9中防火墙的配置，把E-mail服务器移到运行Web服务器的计算机上。

30.20 阅读一个商业入侵检测系统，然后列出其可以检测到的攻击。

30.21 考虑一个在数据包中搜索K字节串的DPI系统。如果数据包含有1486B载荷，采用直接匹配算法，那么最坏情况下的比较次数是多少？

30.22 深度包检测方案为什么不适用于高速网络？

30.23 虚拟专网的两个目的是什么？

30.24 虚拟专网采取哪3种方式通过因特网传递数据？

30.25 当虚拟专网采用IP in IP隧道技术时，为什么可以防止攻击者查看原始的头部信息？

30.26 在有些VPN系统中，发送方要在数据报被加密之前往里面填入随机数量的“0”字节，而接收方收到并经解密之后删去这些额外的字节。所以，这种随机填充“0”字节的唯一作用就是使加密后的数据报长度与未加密数据报长度无关。请问：为什么数据报长度这么重要？

30.27 列出8个用在因特网中的安全技术，并说明各自的用途。

30.28 阅读有关WEP协议存在缺陷的资料。请问：WPA协议是如何避免产生那些问题的？

第31章 网络管理

31.1 引言

前面介绍了使用因特网的各种传统应用。本章我们介绍网络管理，以扩大我们对网络应用的研究。本章介绍在工业界使用的一种网络管理概念模型，并利用该模型来解释网络管理活动的范围；在解释了为什么网络管理是既重要又困难之后，描述各种网络管理技术。本章还讲述各种可用的网络管理工具，包括网络管理员测量、控制那些构成网络的设备（如路由器）所用的应用软件。本章还解释用于网络管理系统的一般模式，并描述网络管理系统所提供的功能。最后，以一个网络管理协议为例，解释该协议软件的工作过程。

31.2 管理内部网

网络管理员（network manager），有时也称网络管理者（network administrator），负责规划、安装、操作、监测及控制那些构成计算机网络或内部网的硬件和软件系统。管理员对网络进行规划，以满足性能要求，监控操作，检测和纠正导致通信效率低下或无法通信的问题，改善有关条件以避免类似错误重复出现。由于软件、硬件都可能导致问题，所以网络管理员必须对两者都要实行监视。

有3个原因使网络管理变得困难。第一，在大多数组织中，内部网是异构的——内部网中包含的硬件和软件组件是由多个公司制造的。第二，技术不断变化，这意味着新设备和新服务不断出现。第三，大部分内部网规模比较大，这意味着内部网的某些部分与其他部分相隔距离较远，而且要检测出远程设备中通信问题的原因可能会特别困难。

由于网络被设计成能自动地克服出现的一些问题，这些也使网络管理变得困难。例如，路由协议寻路故障以及断续性丢失分组都是可以被忽视的瞬间故障，就很难觉察出来，因为TCP会对传输中被破坏的数据进行自动重传。可惜的是，自动差错恢复却造成了不良后果，因为分组重传要占用网络带宽，而这些带宽本可用于传输更多的数据。类似地，未被检测到的硬件故障，在备份路径也失效的情况下，可能变得很危险。

概括：

> 虽然网络硬件或协议软件包含的机制能够自动地绕过故障，进行路由或者重传丢失的分组，但网络管理员仍然需要检测并纠正潜在的问题。

31.3 FCAPS：行业标准模型

网络行业使用FCAPS模型来表征网络管理的范围，这个缩写字是来自国际电信联盟（International Telecommunications Union，ITU）发布的M.3400建议。[⊖]FCAPS扩展为网络管理的5个方面，图31-1归纳了该模型每个字母所代表的含义。

⊖ M.3400是规定电信管理网（Telecommunication Management Network，TMN）如何配置与操作元素列标准中的一部分。

故障检测与纠正（fault detection and correction）。故障检测占据网络管理运作方面的主要部分。管理员监视网络设备以便检测出现的问题，并采取适当的步骤纠正问题。可能的故障包括软件故障（例如，服务器上的操作系统崩溃了）、链路故障（例如，有人意外地切断了光纤）以及设备故障（例如，路由器的电源出现故障）等。

缩写字母	含　义
F	故障检测与纠正
C	配置与操作
A	计费与结算
P	性能评估与优化
S	安全保证与保护

表31-1 网络管理的FCAPS模型

通常，用户是通过引用一个高层次的症状来报告发生故障的，例如，"我刚才不能访问一个共享磁盘"。管理员必须调查以确定问题是否出现在软件、安全性（例如，系统设置了新的密码）、服务器或者链路上。我们称管理员执行根本原因分析（root-cause analysis）。通常，网络管理员可以通过把多份报告相互联系进行分析来确定原因。例如，如果一个站点的很多用户突然开始抱怨很多服务不能使用，管理员可能就会怀疑问题出现在所有服务都要使用的共享连接上。

配置与操作（configuration and operation）。配置似乎是网络管理被不足道的一个方面，因为配置只需要执行一次——在配置被创建之后，可以保存在磁盘中，因此设备重启时可以自动地安装该配置。事实上，配置是复杂的，原因有3个：第一，网络包含很多设备和服务，而所有设备的配置必须是一致的；第二，新增设备和服务或者改变策略时，网络管理员必须考虑所有的配置，保证整个网络能够正确地实现变化；第三，目前的工具允许管理员配置单个设备和单一协议，却没有简便的方法来配置一组异构的设备。

计费与结算（accounting and billing）。在很多企业内部网中，计费和结算是不重要的。公司收取运行网络的费用并记在中央账户上，这很像电力或电话服务的费用。然而，在ISP网络中，计费和结算比管理的其他方面可能消耗管理员更多的时间，例如，如果ISP提供的是按允许的通信量限度来划分的层级式服务，那么系统必须对每个用户的通信量分开计费。通常，服务协议规定客户支付的费用取决于一项量度标准，如客户每天发送的总字节数。因此，测量所有客户的流量并保存详细记录以便用于生成一张账单是非常重要的。

性能评估与优化（performance assessment and optimization）。管理员执行两种类型的性能评估：诊断评估和趋势评估。诊断评估用于检测问题和低效率的情况，而趋势评估允许管理员预期需要增加的容量需求。诊断评估寻找使现有网络利用率最大化的方法。例如，如果管理员找到一条低利用率的路径，他就可能会想办法把通信量移到该路径上。趋势评估寻找提高网络性能以满足未来需要的方法。例如，大部分管理员监视他们组织与因特网之间链路的利用率，并在平均利用率超过50%时，采取行动以增加链路的容量。

安全保证与保护（security assurance and protection）。由于安全问题要跨越协议栈的各个层次和多个设备，所以安全方面的管理就成为网络管理中最困难的一个方面。特别是，安全往往遵循"最薄弱环节类推"原则——如果一个设备上的配置是不正确的，整个站点的安全性就可能受到威胁。此外，由于攻击者不断地发明新的方法来破坏安全，所以在某一特定时间内安全的网络，却不能保证在后面的时间内不会遭到损害，除非管理员及时作出改变。

31.4 典型的网络元素

网络管理系统使用通用术语网络元素（network element），一般简称网元，是指任何可以被管理的网络设备、系统或机制。虽然很多网元是由物理设备组成，但该定义还涵盖了它所提供的服务（如DNS）。图31-2列出了典型的网元。

业界使用术语元素管理（element management）来指单个网络元素的配置与操作。遗憾的是，大部分可使用的工具仅提供元素管理。因此，要创建一个端到端的服务，管理员必须配置沿着路径的每个网络元素。例如，要创建一个跨过多个路由器的MPLS隧道，管理员必须独立地配置每个路由器。类似地，为了要在整个网络中实施一种策略，管理员必须配置每个元素。

可管理网元	
第二层交换机	IP路由器
VLAN交换机	防火墙
无线接入点	数字线路
头结DSL调制解调器	DSLAM
DHCP服务器	DNS服务器
Web服务器	负载均衡器

图31-2　必须被管理的网络元素例子

当然，配置很多设备时，很容易犯错，这将导致元素管理容易受到错误配置的影响。更重要的是，为了诊断错误，管理员必须逐个检查系统。

要点　由于每次只允许管理员配置、监视或控制一个网络元素，所以对系统的元素管理是繁重的劳动且容易出错。

31.5　网络管理工具

根据其一般用途，网络管理工具可以分为12类：

- 物理层测试。
- 可达性与连通性。
- 分组分析。
- 网络发现。
- 设备问询。
- 事件监控。
- 性能监控。
- 流分析。
- 路由与流量工程。
- 配置。
- 安全实施。
- 网络规划。

物理层测试包括线波传感器测试，这可以在很多LAN接口卡和用来测量RF信号强度的无线强度计量器中看到。ping提供了可达性工具的最好例子，并得到网络管理员的广泛使用。分组分析器（packet anlayzer），也称为协议分析器（protocol analyzer），捕捉并显示分组或关于分组的统计信息。在网上可以下载Ethereal分析器。

网络发现工具通过探测设备来生成一幅网络地图。通常，管理员使用这样的地图寻找网络中的元素，然后使用设备问询工具访问每个元素。事件监控工具产生告警——通常情况下，管理员配置设备在超过某些阈值（例如，链路的利用率达到80%）时发送警报，而监控工具在管理员工作站中显示警报。性能监控工具可以绘制出随时间变化的性能曲线，帮助管理员洞察趋势。

流分析工具（如NetFlow分析器）也能帮助管理员洞察趋势，还能帮助管理员发现特定类型的流量（如VoIP流量的上升）改变情况，而不仅仅是报告整体业务流量。

路由、流量工程以及配置工具是相互关联的，它们各自从不同角度帮助管理员控制相关的网络元素。路由工具控制配置和路由更新协议的监控，以及由路由改变而产生的转发表，

流量工程工具关注于与QoS参数相关的配置及MPLS隧道的监控。通用目的的配置工具允许管理员安装或改变网络元素中的配置。特别地，一些配置工具可以在一组（通常是类似的）网元中自动重复同样的配置改动。例如，如果一个防火墙的规则改变了，而一个站点有多个防火墙，那么自动配置工具（通常是Perl脚本）可以在每个防火墙中安装同样变化的规则。

存在很多安全工具控制各种安全元素。有些安全工具允许管理员确定一种策略，或者试图配置设备以实施该策略，或者试图测量设备以确保该策略是有效的。管理员可以使用其他安全工具来测试安全性——用该工具去攻击设备或者服务，并报告管理员该攻击是否成功。

网络规划是复杂的，且规划工具是其中最先进的。例如，有一些运行线性规划算法的工具，有助于管理员优化网络体系结构或者规划流量管理。也有一些工具可帮助管理员评估网络弱点（例如，识别网络中存在两个或多个硬件故障而将导致用户连接不上因特网的地方）。

概括：

有各种各样的工具可以帮助管理员配置、测量、诊断及分析网络。

31.6 网络管理应用

上述的大部分工具都是离网运行的。也就是说，管理员在一个单一地点使用网络技术与特定网元通信。令人奇怪的是，网络管理并没有被定义为较低层协议的一个必需部分，而是将用于监控网络设备的协议放在应用层上运行。当管理员需要与特定硬件设备互操作时，他就运行一个应用程序来充当一个客户，而网络设备上的应用程序则充当一个服务器。客户和服务器使用传统的传输协议（如TCP或UDP）进行交互通信，由客户发送请求，服务器作出响应。而且，大多数管理员都是在实际运作的网络上发送管理业务，而不是另外构造一个单独的网络。

为了在普通用户可调用的应用程序和专为网络管理员保留的应用程序之间避免混淆，网络管理系统中避免使用术语客户（client）和服务器（server），而把在管理员的计算机上运行的客户程序称为管理员（manager），把网络设备上的服务器应用程序称为代理[⊖]（agent）。

利用传统的传输协议来承载网络管理业务可能看起来有点令人费解，因为无论是协议还是在底层硬件上发生的故障都会阻碍分组的传输，使得在故障发生期间不可能对设备进行控制。有些管理员安装单独的硬件去处理更关键的设备管理（如在路由器上直接连接一个拨号调制解调器）。但在实际中，却很少需要这种系统。实际上，使用应用层协议进行网络管理效果很好，其原因有二。第一，在由于硬件设备故障而无法通信的情况下，管理员可以与仍然维持工作的设备进行通信，根据通信是成功还是失败的情况来帮助定位故障源。第二，使用传统的传输协议意味着来自管理员的分组与正常通信业务是在同样的条件下传输。这样，如果网络延迟太大，管理员就可以立即发现问题。

31.7 简单网络管理协议

用于网络管理的标准协议称为简单网络管理协议（Simple Network Management Protocol，SNMP）。目前的标准是版本3，写为SNMPv3。SNMP协议精确定义了管理员如何与代理之间进行通信。比如，SNMP定义了管理员传输给代理的请求格式以及从代理返回的响应格式。另外，SNMP还定义了每种可能的请求和响应的确切含义。SNMP还特别规定SNMP报文要采用

⊖ 虽然我们会遵循传统使用术语管理员（manager）和代理（agent），但读者应该记住它们是按照客户与服务器模式运作的。

标准的抽象语法表示版本1（Abstract Syntax Notation.1，ASN.1）进行编码。

虽然ASN.1编码的全部细节超出了本书的讨论范围，但这里仍要举一个简单例子来帮助理解其编码过程：假如要在管理员和代理之间传输一个整数，为了能容纳大的整数值又不浪费空间，ASN.1使用长度和值的组合作为要传输的对象。例如，0－255之间的整数用一个字节传输即可，而从256－65535之间的整数则需要两个字节，更大的整数则需要3个或更多的字节。为了对整数进行编码，ASN.1要发送一对值：先是该整数的长度L，接着是L个字节的整数值。为了能够对任意大小的整数进行编码，ASN.1又允许整数长度可以占用多于一个字节，但通常SNMP并不需要用这种扩展的长度来发送整数。如图31-3所示为ASN.1编码的例子。

十进制整数	等效的十六进制数	长度（字节）	整数值（十六进制）
27	1B	01	1B
792	318	02	03 18
24 567	5FF7	02	5F F7
190 345	2E789	03	02 E7 89

图31-3　ASN.1对整数编码的例子

31.8　SNMP的取/存操作模式

SNMP协议没有定义一个很大的命令集，而是采用取/存操作模式（fetch-store paradigm）。该模式有两种基本操作：获取操作（fetch）用于从某个设备获得一个值，存储操作（store）用于给某个设备设置一个值。每个可以执行取/存操作的对象都被赋予一个唯一的名字；每个指定获取或存储操作的命令必须指明操作对象的名字。

如何利用获取操作来监视设备运行或获得其状态信息，应该好理解。首先必须定义出一组状态对象并给它们命名。为了获得状态信息，管理员只需获取与给定对象相关联的值即可。例如，定义一个计数器对象来对某设备因校验和出错而丢弃的帧进行计数。设备必须设计成每当检测到一个校验和出错时能对该计数器的值加1，管理员就可以使用SNMP获取与该计数器相关联的值，从而确定是否出现校验和出错。

采用取/存操作模式来控制设备就不是那么直观。控制操作被定义为存储到对象里面的伴随操作（side-effect）。例如，SNMP并不包括单独对校验和出错计数器执行复位（reset）或者重启（reboot）设备的命令。在校验和出错计数器这种情况下，往对象中存入0值最直接，因为将计数器复位即是置为0。但是对于重启操作，SNMP代理必须被编程成能够解释管理员的存储请求，并执行正确的操作顺序才能达到期望的结果。因此，SNMP可以定义一个重启对象，且规定对该对象赋给0值后（具伴随操作）就能使系统重新启动，当然，由为底层设备没有直接实现它们，所以从这个意义上来说，SNMP对象是虚拟的，而代理只是接收请求并执行对应每个获取或存储操作的动作。

概括：

> 管理员和代理之间采用取/存操作模式的交互操作。管理员从对象那里获取一些值来确定设备的状态；至于对设备的控制操作，则被定义为往对象中储存信息的伴随作用。

31.9　管理信息库和对象名

SNMP要访问的每个对象都必须定义并给定一个唯一的名字。而且，管理员和代理程序双

方必须对这些名字及取/存操作的含义取得一致。SNMP可以访问的所有对象的集合称为管理信息库（Management Information Base，MIB）。

事实上，SNMP并没有定义MIB，其标准只是规定了报文格式以及描述如何对报文进行编码；至于MIB变量以及相关的取/存操作的含义，则由独立的标准作出规范。其实，有专门的标准文档为每种类型的设备规定了MIB变量。

MIB中的对象采用ASN.1命名方案，每个对象分配一个值长的前缀以保证其名字的唯一性。例如，一个负责对某设备所接收到的IP数据报数量进行计数的整数对象，可以命名为：

iso.org.dod.internet.mgmt.mib.ip.ipInReceives

而且，当SNMP报文中要表示对象名字的时候，名字的某个部分都被赋予一个整数。这样，在一个SNMP报文中，ipInReceives的名字就是：

1.3.6.1.2.1.4.3

31.10 MIB变量的种类

因为SNMP没有指定MIB变量集，所以MIB的设计比较灵活，而且可以根据需要定义新的MIB变量并使之标准化，无需改变基本协议。更重要的是，通信协议从对象定义中分离出来，这样可允许各部分的工作组按各自的需要去定义MIB变量。例如，当要设计新的协议软件时，负责开发协议的工作组就可以定义出用于监控协议软件的MIB变量；类似地，当要开发新的硬件设备时，负责这方面工作的工作组可以去定义用于监控设备的MIB变量。

正如原来设计者所料，目前已经定义了很多种MIB变量集。例如，有对应于协议（如UDP、TCP、IP、ARP）的MIB变量集，也有对应于网络硬件（如以太网）的MIB变量集。此外，工作组还为一些硬件设备（如路由器、交换机及打印机等）也定义了相应的MIB变量。[○]

31.11 对应于数组的MIB变量

除了像计数器整数那样简单的变量外，MIB还包含对应于表或数组的变量。定义这样的变量很有用，因为它们对应计算机系统中信息的实现。例如，考虑一个IP路由表，在大多数实现中，都可以把路由表看作是记录项的数组，其中每项都包含目的地址以及到达该地址的下一站点地址。

与传统编程语言不同的是，ASN.1不包括索引操作，索引引用是隐式的——发送方必须要知道被引用的对象是一个表，而且必须把索引信息添加在对象名上。例如MIB变量：

standard MIB prefix.ip.ipRoutingTable

对应于IP路由表[○]，它的每个记录项包括了几个域，在概念上，路由表总是被目的IP地址索引的。为获取一个记录项中特定域的值，管理员可以规定出如下形式的名字：

standard MIB prefix.ip.ipRoutingTable.ipRouteEntry.field.IPdestaddr

其中，field对应该记录项中的一个合法域，IPdestaddr是用于索引的4字节长的目的IP地址。例如，域ipRouteNextHop对应于项中下一站地址。当转换成整数表示时，要请求的下一站地址就变成：

○ 除了可用于任意设备的通用MIB变量，许多提供者为他们的硬件和软件定义了特定的MIB变量。

◎ 回顾一下，转发表原先称为路由表，术语的改变大概是在2000年。

1.3.6.1.2.1.4.21.1.7.destination

其中，1.3.6.1.2.1为标准MIB的前缀，4是ip的代码，21是ipRoutingTable的代码，1是ipRouteEntry的代码，7是ipRouteNextHop的代码，destination是目的IP地址。

提示：

> 虽然ASN.1不提供索引机制，但MIB变量支持数组或表结构。为了仿真带有ASN.1变量的表或数组，可以对记录项的索引标识进行编码，然后将它添加在变量名上。当代理软件遇到对应于一个表的名字时，就把索引标识提取出来，并利用这个索引信息去选择正确的表项。

31.12　本章小结

网络管理员负责监视和控制组成内部网的硬件和软件系统。FCAPS模型定义了网络管理的5个基本方面：即故障检测、配置管理、计费管理、性能分析和安全管理。已有各种工具提供给管理员帮助执行管理职能。由于大部分工具仅提供元素管理，所以管理员必须手工处理跨设备的任务。

由于网络管理软件采用客户/服务器模式，所以该软件要求由两个部分组成：运行在管理员计算机上的部分作为客户，称为管理员（manager），运行在网络设备上的部分作为服务器，称为代理（agent）。

简单网络管理协议（Simple Network Management Protocol，SNMP）是因特网上使用的标准网络管理协议，它定义了管理员与代理之间交换的报文格式和含义。SNMP没有定义很多的操作命令，而是采用取/存操作模式，即由管理员发出"取"操作请求从变量中取得它的值，或者发出"存"操作请求将值存储到变量中。SNMP中对网络设备的控制操作都被定义为"存"操作的伴随操作。

SNMP没有定义可用的MIB变量集，所有MIB变量及其含义都由另外的标准来定义。这样就可能由各个工作组去针对每种硬件设备或协议定义相应的MIB变量集。MIB变量的命名采用ASN.1标准，所有MIB变量都有一个很长的、有层次的ASN.1名字，并可转换为紧凑的数字表示形式以便于传输。ASN.1没有定义集合数据类型（如表或数组），也没有下标操作符。为了要实现这类操作，采取让MIB变量仿真成表或数组的做法，ASN.1通过在后面添加索引信息来扩展变量名。

练习题

31.1　举出一个隐藏错误协议机制的例子。

31.2　如果用户抱怨说他们不能访问某一特定的服务，该投诉可能涉及FCAPS的哪方面？

31.3　如果一个防火墙出现故障，这种情况属于FCAPS的哪一方面？为什么？

31.4　找出两个可管理的元素的例子，且该元素没有在图31.2中列出。

31.5　什么是协议分析器？

31.6　流分析工具能够帮助管理员理解什么东西？

31.7　网络管理软件使用什么术语来代替"客户端"和"服务器"？

31.8　ASN.1定义了一个整数的精确格式，为什么ASN.1标准没有规定每个整数是一个32位的值？

31.9 一直有争议的是，我们不该使用同一网络来调试网络中的问题。为什么SNMP基于同一网络来进行管理操作和调试排错操作？

31.10 编写一个计算机程序读取一个任意大的十进制整数，按表31-3那样对其进行编码，并打印结果。

31.11 SNMP使用哪两种基本操作？

31.12 下载免费的SNMP管理软件，并用它试图联系一个设备（如打印机）。

31.13 SNMP为每个可能的MIB变量定义一个名称吗？请解释。

31.14 采用在名字后面添加索引信息的数组表示方法，而不是使用整数下标的传统数组表示方法，请问：这种做法的主要优点是什么？

31.15 阅读有关ASN.1对名字及其值编码的资料，编写一个计算机程序，对ASN.1名字（例如ipInReceives）进行编码和译码。

第32章 网络技术及应用发展趋势

32.1 引言

因特网最具魅力的一个方面，就在于它不断引入新的网络应用和网络技术。在过去十年中所发明的网络应用已经占据了绝大部分因特网流量，而新的网络应用却由于要依赖于新的底层技术和基础结构，所以最发明之初的因特网来实现这些新的东西是不可行的。

本章将概括一些网络技术、网络应用及网络服务的发展趋势，并关注网络目前发展以及长期的研究课题。

32.2 可扩展网络服务的需求

从狭义上讲，通信的客户—服务器模式意味着一个应用程序（服务器）先启动，然后等待另一个应用程序（客户）的连接请求。从广义上讲，网络也是使用客户/服务器这个术语来表征一种体系结构，在此结构中有很多潜在客户要连接到单个中央式服务器。例如，运行Web服务器的公司希望来自任意多的用户能立连接。中央式服务器的缺点在于它所导致的性能问题，即随着客户数量的增加，服务器（或通向服务器的接入网络）迅速成为瓶颈。尤其是如果每个客户下载大量字节的内容，问题会变得更为严重。

服务器瓶颈问题被认为是因特网服务的最重要的限制之一。于是，不仅是网络研究机构，乃至网络企业都在寻找新的途径，以提供允许扩展网络服务和适合网络发展新趋势的体系结构和技术，而且许多方法正在被使用。下面章节将介绍其中的几种。

概括：

为了能使因特网服务进行扩展和升级，已经规划设计了各种各样的技术；虽然方法各异，但在不同场合它们都是各有所用的。

32.3 内容缓存加速

首先采用的扩展技术之一是对Web网页内容的高速缓存。例如，ISP经常使用缓存区对每个静态网页（即内容不经常变化的网页）的副本进行保存。假如ISP的N个用户要获取同一个网页，则只要求发送一个请求给原服务器（origin server），其余N−1个请求都可以从缓存中得到满足。

像Akamai这些公司，通过提供分布式的缓存服务扩展了缓存的理念。Akamai将一系列服务器遍布整个因特网部署，任一单位可以与Akamai签订合同，在特定Akamai缓存上预装内容；为了确保缓存内容的即时性，单位用户需要定期更新其Akamai缓存。这样，对该单位Web网站的访问者，可以就近从Akamai的缓存上（而不是从该单位的中央服务器上）获取大部分的内容。结果，中央服务器的负载量可大大减少。

32.4 Web负载均衡器

由于利用率很高，而且许多零售商务都要依赖于对客户直销的Web网站，所以对Web服务器进行优化的问题，引起人们极大的注意。有一种用来构建大型网站的有趣机制，叫做负载均衡器（load balancer）。负载均衡器允许一个网站拥有多台计算机，而每台计算机都运行同一个Web服务器，并在多个物理服务器之间分配访问请求。如图32-1所示为这种体系结构的示意图。

负载均衡器检查每个输入请求，并将请求送到其中的一个服务器上。负载均衡器要记住当前的请求，并把来自某特定访问源的所有请求都引导到同一个物理服务器。为了保证所有的服务器对一个请求返回相同的应答，服务器要使用共同的共享数据库系统。因此，假如一个顾客发出一个订单，Web服务器的所有副本都将能够处理这个订单。

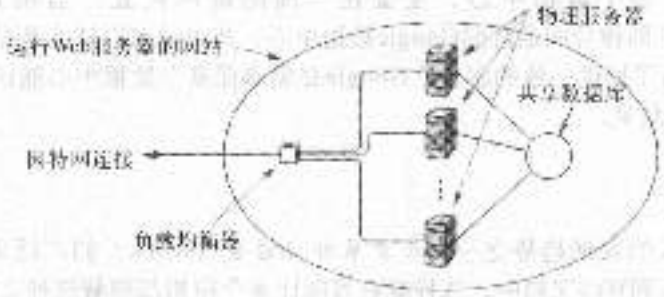


图32-1 用于大规模Web网站的负载均衡器示意图

32.5 服务器虚拟化

可扩展性的另一个转折起因于服务器虚拟化（server virtualization），其动机始于对下面情况的观察：许多网站都运行着多个服务器（例如，email服务器，Web服务器和数据库服务器等）。在传统的架构中，每个服务器必须放置在一台物理计算机上。那么，由于运行在计算机A上的服务器软件可能是最忙的，而运行在计算机B上的服务器软件却可能是闲的，这样就会产生性能上的问题。

服务器虚拟化解决了上述问题，它的做法是：允许管理员随时将服务器软件从一台计算机迁移到另一台计算机。当然，其中有很多技术细节需要处理，包括推进过程的各种变化，但其思路是简单易懂的——在支持进程迁移的虚拟机（Virtual Machine，VM）系统下运行服务器软件。如果某台物理计算机变得不堪重负，那么管理员可以将一个或多个进程迁移到另一台计算机上。

32.6 P2P通信

20世纪90年代，有几个研究小组采用一般技术做试验，目的是提高下载速度。其做法不是客户从中央服务器获取完整的文件，而是只获取文件的一个个片段，而且这些文件片段被放置在遍布因特网的许多服务器中。当客户需要一个文件片段的任何时候，就选择从就近的服务器中获取该片段副本。为了增加可寻找的文件片段存放位置的数量，每个获取了一个文件片段的客户都自己起到服务器的作用，以允许其他客户来获取该文件片段。这种方法就被称为P2P体系结构（peer-to-peer architecture）。

已经开发出一些广为人知的P2P系统，可以用于音乐文件的共享。例如Napster和Kazaa都采用P2P方法，而且都在年轻人中广为流行。当然，普通的用户并不在意底层技术——只在乎系统是否允许他们能获得音乐文件的副本。许多用户并不知道他们是在使用P2P系统，其实他们的计算机也在自愿地为其他用户传播文件。

32.7 分布式数据中心

虽然网页缓存、负荷均衡、服务器虚拟化和P2P结构都能提高服务器的可扩展性，但有一些网站的业务量非常大，还需要有另外的解决方案，即对整个网站的复制。我们用术语分布式数据中心（distributed data center）来表征这种方法。

下面以Google搜索引擎作为例子。它每天要接到数以百万计的连接请求，为了处理这些请求，Google建立了多个数据中心，配置在不同的地理位置。当用户输入域名www.google.com时，用户即被导向最近的Google数据中心，其方法可以认为是站点间负载均衡的一种形式。当然，为了提供一致的服务，Google必须确保某个数据中心能正确地返回与其他数据中心相同的搜索结果。

32.8 通用表示

网络业界中最吸引人的发展趋势之一是可扩展标记语言（XML）的广泛采用。起初，XML被设计成将结构引入到Web文档中，这样就有可能让多个应用都理解这种文档。XML不使用固定的标签，而是允许编程员选择任意标签，这样就有可能给每个域以一个直观的名字。例如，人们可以假设一个含有标签<name>、<street>、<city>、<country>和<country>的文档，它包括一个个人地址的记录。XML所隐含的关键思想之一，就是它的对自描述文档进行编码的能力，亦即，XML文档包含一个可指定合法文档结构的样式表（style sheet）。

XML已经成为一个事实上的表示标准，并正被应用在各种新的场合，这是XML设计之初所没有预料到的。比如，XML被用在Web服务器和数据库的接口上，而且也已开发出一些能解析XML的负载均衡器。此外，XML还用来控制移动设备的下载，以及被网管系统用来表示它的操作规范。

32.9 社区网络

21世纪初，因特网由顾客使用模式转向同类人群之间的社区交互模式。最初，因特网上的大多数信息是由生产商、机构（如传媒公司）提供，个人用户只是使用和消费这些信息，但不产生信息。2000年以后，出现一批像facebook，myspace和youtube那样的网站，它们允许任何用户创建内容，这就意味着普通的用户也可以上传更多的数据。

这种交互模式的转移，在年轻用户中最为引人注目。许多十几岁的年轻人都在上面提到的网站上注册，或者开设自己的博客（blog）。在美国，有相当比例的新近结婚夫妇都是通过在线服务而结识的。此外，在线聊天的使用以及其他形式的人与人通信，也在不断增长。

32.10 移动性及无线联网

移动通信也是一个非常引人注目的发展趋势，而且人们希望能够不间断地访问因特网。目前，大多数酒店都提供了因特网接入服务，航班也提供飞行中的因特网服务。笔者在近期的一次乘船巡游中，欣喜地发现船上所提供的因特网连接非常流畅，而且可以进行VoIP

通话。

移动通信的需求刺激了无线技术的发展，其中制定了许多无线标准，802.11n技术与它的前身802.11b相比，可以提供更高的吞吐率。然而，最显著而重要的变化却发生在移动蜂窝电话业。IP协议将很快取代所有现有的蜂窝电话协议，特别是，一旦移动蜂窝运营商使用WiMAX技术，整个移动电话系统将是基于IP的，也就意味着蜂窝电话服务与因特网融合了。

具有讽刺意味的是，当移动电话业把IP技术作为其长期战略时，被称为移动IP（mobile IP）的技术却没有被网络业界所采纳。作为替代，大多数移动用户都依靠Wi-Fi作为本地接入技术，并使用VPN软件连接自身的商务系统。

32.11 数字视频

有线电视提供商正在用数字技术来替代模拟传输设备，而且很快将在分组网络上以数字形式来传递内容。事实上，目前许多有线电视提供商使用IP作为分组协议，术语IPTV就是使用IP技术的电视。

IP用于视频产生了一些有趣的机会。首先，电视与因特网的融合，易于使用PC来观看电视节目，或使用数字电视来充当PC显示器。其次，使用IP易于实现视频点播（on-demand video），即用户可以按需要来访问节目内容，使用暂停、倒片功能来控制播放，还可以录存实况节目以备稍后观看。

32.12 多播传递

虽然因特网大范围内的多播并没取得很大成功，但是转向IPTV却刺激了多播方面的兴趣，其动机出自于优化视频传送的需要。尽管电视提供商可以提供数以百计的节目频道，但某一个签约用户在一个时间段内通常仅接收少数几个电视演播内容，而且往往只有少数节目频道能吸引大多数观众。

IP多播通过加入节目多播组，以允许用户注册自己感兴趣的节目，相邻的用户连接成一个逻辑局域网段。一旦用户加入多播组后，有线电视提供商在该网段上以多播形式发送一个节目的副本，只要该网段内还有一个用户在观看节目，多播就一直继续发送。

要点 采用IP多播时，只需在一个逻辑网段上发送一份电视节目的副本；只要网段内不再有用户观看该节目，就立刻停止多播。

32.13 高速接入与交换

在因特网边缘，接入技术（如DSL和电缆调制解调器）可以提供每秒几兆比特的数据速率，其吞吐量比传统电话拨号电话线连接要高出两个数量级。在美国的部分地区，服务提供商还提供光纤到户（FTTH），其潜在数据速率可提高到每秒千兆比特，超出DSL和电缆调制解调器的速率3个数量级。

用在企业数据中心的以太网交换机可提供1Gbit/s速率到桌面，更高容量的链接可工作在10Gbit/s，并有望提高到40Gbit/s，这样数据速率足以支持高清视频传输了。

32.14　光交换

在因特网的核心又，存在一个大问题：如何解决光与电的结合？光设备可提供10Gbit/s端到端的光通路（通常称为λ网）。如果要建立一条光通路，采用大多数目前流行的技术需要较长的时间（数秒），但是已出现更新的光技术，允许减少这种建立时间，最终有可能将这种时间缩短到1ms以下。

如果光通路可以快速建立，那么应该如何去利用它呢？ISP应该使用光通路来连接路由器，再采用分组技术来接入吗？ISP应该是在用户每次形成TCP连接时才建立光通路吗？这些都是该重要研究领域的基本问题，多数大型ISP都坚信：光交换势必引起高度的重视。

32.15　网络的商务应用

大多数公司都依赖计算机网络来处理商务活动的所有事务。然而，网络也在3个方面改变着商务过程。第一，RFID技术的应用改变了生产、航运和仓储；第二，第二层高速交换机和分组技术在音频和视频方面的应用，正在使以高品质视频会议系统替代人工出差成为可能；第三，许多商务活动正在从严格的命令—控制模式转向更加协同的管理风格，团队以这种风格一起工作并作出决定。支持群组交互（如wikis，即用户可以提交内容和查看内容的互动网页或网站）的工具和网络基础结构的应用，正在使通过网络进行协同工作成为可能。

32.16　传感器普遍应用

低成本的有线和无线网络以及低功率传感器件，已经使构建大型传感器网络并将这种网络连接到因特网成为可能。传感器正在被用来监测环境（如监测空气、水质和搜集气象信息等），跟踪野生动物，帮助农民监测庄稼，监视写字楼的人员，以及估测高速公路的交通流量。

一种特别有趣的传感器应用是在居住建筑物中的使用。现在人们可以通过安装传感器来查看居住房屋的温度、湿度，监测可能的家庭危险因素（例如，烟雾和一氧化碳含量）。家庭传感器网络可以被连接到因特网，这样您上即使在旅行途中也可查看房屋的状况。[3]很快就有可能买到便宜的各种传感器，如在每个灯泡或者在每个器具上安装的传感器。

32.17　Ad Hoc网络

自分组网络出现的早期以来，美国军方就已经资助研究自组织的Ad Hoc网络。这种网络是指：一组无线台站自动发现临近站、选择邻接、建立任一无线台站都彼此互相可达的路由。美军研究Ad Hoc网络的动机是来自一种假想的军事行动：士兵每个人都携带有无线网络站，这些无线站可以自动地形成一个通信系统。

这种Ad Hoc自组织网络在民用领域也正日益显得重要，特别是很适合一些发展中国家的不发达地区。在美国，一些农场主们正在利用Ad Hoc技术将乡下的农场连接到因特网。每个农场主建立一个无线站（通常在高的建筑物上，如筒仓），且每个站允许按需要转发分组。在发展中国家，Ad Hoc网络被作为为整个村庄提供接入因特网的廉价方法。

32.18　多核CPU和网络处理器

高数据率对网络设备制造商引发一个问题：如何构建出能够快速处理分组的系统。高端

[3] 您可已经在自己家中建立了这样的监测网络。

路由器必须具有处理10Gbit/s到达接口的分组能力。当然可以使用专用芯片（ASIC），但是其价格昂贵，设计和调试需耗时数月。传统处理器可以满足低端网络设备的（如家庭用无线路由器）需要，但对更高数据率缺乏足够的计算能力。

对此，芯片供应商提供了两种解决方案：一是由芯片供应商提供多核CPU，也就是一个CPU中含有多个处理器，该方法可在N核间分配处理输入分组，即一个核只处理1/N量的输入分组；二是由芯片供应商提供网络处理器（network processors）。我们可将网络处理器看作是一个含有多核的快速CPU，再加上它含有高速执行通用分组处理任务的特殊指令系统。值得注意的是，设备供应商在他们的设备中正在使用更多的可编程处理器。

32.19 IPv6

谈到网络发展趋势不能不提及IPv6。IPv6的工作起步于1993年，其设计完善又持续了数年。最初，倡导者声称：由于IPv4不能够有效处理音频和视频，不够安全，地址即将耗尽，因此需要IPv6来替代现有的IPv4。在IPv6提出后，每年都有来自学术界和企业界的各种团体对IPv4的延续和IPv6的崛起作出了预测分析。在此期间，IPv4也不断进行适应调整，运行了多媒体应用，在安全性上已达到了IPv6的程度，NAT和CIDR编址技术也扩展了IPv4的地址容量。IPv4依然是因特网的基本协议。一些移动电话运营商，特别在亚洲地区，一直认为IPv6是让手机拥有IP地址的一种途径，但是运营商同样也可以选择设计第二层编址方案。

在选择IPv4还是IPv6这个问题上，技术上并没有采用IPv6的理由，事实上，IPv6的分组处理需要更大的开销，迁移到IPv6会限制IP分组发送的速度。因此，采用IPv6的动机就变成了一个经济上的权衡：从互特网上去除NAT转而采用完全的端到端的寻址，这当然可以做到，但这样做就意味着要更换所有的网络设备和软件。所以，很难告诉消费者何时能决定这种高花费的改变被证明是合理的。

32.20 本章小结

因特网在继续发展和进步，新的应用和技术层出不穷。当前的趋势包括：更高速度的技术，提高可移动性和可扩展性。在网路应用方面，其趋势已经转向于社区网络。此外，新技术已经使普通用户可以生产信息内容，商业上也正在使用能支持协同环境的各种工具，且正在使用高端的远程会议系统来替代差旅活动。

练习题

32.1 请解释内容级存是如何实现因特网扩展的。

32.2 负荷均衡器用在什么地方？

32.3 因为共享资源会造成热预效应，所以具有N台物理服务器的一个网站就有可能处理不了N倍的单台每秒请求数。请举出两种被共享的资源。

32.4 服务器虚拟化除了能实现扩展外，还可以允许网站在低负荷时（如周末）节省能源。请解释该怎样做。

32.5 P2P计算通常要和什么一般应用联系在一起？

32.6 分布数据中心方法对于每个Web请求都要访问中心数据库的商务系统有意义吗？为什么？

32.7 举出3个社区网络应用的例子。

32.8 如何让因特网与移动通信网融合?

32.9 数字视频能提供用户什么?

32.10 当使用光纤传送数据到生产或商户时，其速率比DSL和cable modem快多少?

32.11 举出几个在商业应用方面网络发展新趋势的例子。

32.12 传感器网络用在哪些场合?

32.13 什么技术正在被用于提供对乡村的远程接入?

32.14 说出两种用于提高路由器和交换机速度的技术。

32.15 为什么移动电话运营商对IPv6特别感兴趣?

附录　一种简化的应用编程接口

引言

第3章叙述了提供给程序员用于构建客户-服务器的套接字API，本附录介绍另外一种API，它不要求程序员掌握套接字接口的细节就可以构建网络应用程序的简化API。附录内容是独立的，不要求读者理解因特网和TCP/IP，因此读者可以在学习本书其他章节内容之前，就可以阅读和理解附录的内容。

附录中所介绍的例子体现出一个重要思想：

> 程序员可以无须理解底层网络技术和通信协议的情况下，就能开发网络应用程序。

为了达到这个目的，我们先介绍一个处理底层通信的库函数小集，并将展示如何利用这些库函数来编写网络应用程序。这里提供的实例代码可以在网站上得到，并且鼓励读者修改这些例子或编写另外的应用程序。

网络通信模型

所有因特网的传输都是由应用程序来完成的，而且应用程序在使用因特网的时候是成对工作的。例如，当用户浏览网页时，运行在用户计算机上的浏览器程序要连接到运行在远端计算机的Web服务器，先由浏览器发出请求，然后Web服务器为此做出回答。只有两端的应用程序才彼此理解其报文的格式和含义。

客户—服务器模式

为了通过因特网进行交互通信，一对应用程序采用一种简单直接的机制：一方应用程序先启动并等待另一方应用程序的连接请求，后者必须知道正在等待的前者的位置，这种组织形式就是大家熟知的客户—服务器交互模式。其中，等待连接请求的程序叫服务器，而发起连接的程序叫客户。为了发起连接，客户必须知道如何连接服务器，在因特网中，服务器的位置由一对标识符给出：

（计算机，应用程序）

这里“计算机”标识正在运行服务器的计算机，“应用程序”则识别在该计算机上的某个特定应用程序。虽然应用软件将两者表示为二进制数值，但人们并不需要直接涉及这些二进制表示。作为替换的方法，也会对这两个二进制值用人类熟悉的字母名字来表示，并由软件将每个名字自动地转换为对应的二进制数值。

通信模式

大多数因特网应用程序在通信时都遵循相同的基本模式，即两个应用程序建立通信关系，来回交换报文，然后终止通信。操作步骤如下：

- 服务器程序首先启动，并等待来自客户的连接请求。
- 客户指明服务器位置，并请求建立连接。
- 一旦连接完成，客户和服务器使用该连接交换报文。
- 客户和服务器完成发送数据后，各自发送文件结束标志，终止连接。

应用编程接口举例

迄今，我们已经在概念层面上讨论了两个应用程序之间的交互操作，现在我们来考虑详细的实现过程。计算机科学家所定义的应用程序接口API，是提供给应用程序员使用的一套操作，它规定了一组函数、每个函数的参数说明以及函数调用的语义。

为了演示网络编程的方法，我们已经设计出一种简单明了的网络通信API，在描述完API后，我们将给出使用这种API的应用程序。图A-1列出了应用程序中会用到的7个函数。

操　作	含　义
await_contact	由服务器用于等待来自客户的连接请求
make_contact	由客户用于请求连接服务器
appname_to_appnum	用于将程序名转换为等效的内部二进制值
cname_to_comp	用于将计算机名转换为等效的内部二进制值
send	由客户或服务器用于发送数据
recv	由客户或服务器用来接收数据
send_eof	客户和服务器完成发送数据后，用来发送文件结束标志

图A-1　由7个函数构成的API例子，它足以为大多数网络应用[⊖]所用

说明：我们的实例代码还将使用第8个函数recvln，但在表中没有单独将这个函数列出来，是因为它只是调用recv函数的一个循环而已，这个循环直至遇到一个行结束符为止。

直观了解API

服务器通过调用await_contact开始等待来自客户的连接请求，客户则通过调用make_contact来建立连接。一旦客户连接上服务器后，双方都以调用send和recv进行数据交换。双方应用程序都必须知道要么发送要么接收——如果两方都在接收而没有发送，将出现永久阻塞。

一方数据发送完成后，调用send_eof发送文件结束符（end-of-file），另一方，recv函数返回一个零值表示已收到文件结束符。例如，如果客户调用send_eof，则服务器将从其recv调用中发现一个零的返回值。一旦双方都完成send_eof调用，则通信终止。

一个简单例子将帮助理解这个API。考虑这样一个应用情况：客户请求连接服务器，发出一个请求，并接收一个应答。图A-2表示出客户和服务器完成以上交互的API调用顺序。



图A-2　客户发出一个请求和接收一个来自服务器应答的API调用示例

API的定义

除了标准C数据类型，我们还定义了用于实例编码的3

⊖ 函数send和recv直接由操作系统提供，该API的其他函数由我们已编写的实例组成。

个数据类型，使用这些类型可以使API独立于任何具体的操作系统和网络软件。图A-3是这3种数据类型的名称和它们的含义。

类 型 名	含 义
appnum	用于标识一个应用的二进制值
computer	用于标识一台计算机的二进制值
connection	用于标识客户与服务器间某连接的值

图A-3 用在范例API中的3个类型名

利用图A-3的3个类型名，我们可以精确定义范例API，对每个函数，下面的类C说明列出了每个参数的类型以及函数返回的具体类型。

await_contact函数

服务器程序调用await_contact函数等待来自客户的连接请求。

```
connection await_contact(appnum a)
```

这个调用使用了appnum类型的一个参数，并返回一个connection类型的值。这个参数指定了一个标识服务器应用的值；客户端请求连接该服务器时，也必须指定相同的值。服务器则利用返回值（connection类型）来传送数据。

make_contact 函数

客户调用make_contact函数来建立与服务器的连接。

```
connection make_contact(computer c, appnum a)
```

这个调用使用了两个参数，其中一个参数标识运行服务器程序的主机，另一个参数标识该计算机上运行的服务器程序；客户则利用返回值（connection类型）来传送数据。

appname_to_appnum函数

客户程序和服务器程序都使用appname_to_appnum函数将人可阅读的服务名转换为计算机使用的内部二进制值。该服务名在因特网中是标准化的（如，WWW表示World Wide Web）。

```
appnum appname_to_appnum(char *a)
```

这个调用使用了一个字符串（type string）类型的参数（C语言中是使用说明char *a来指明串类型），并返回用一个appnum类型的等价二进制值。

cname_to_comp函数

客户程序调用cname_to_comp函数将人可阅读的计算机名转换为计算机使用的内部二进制值。

```
computer cname_to_comp(char *c)
```

这个调用使用了一个字符串（char *）参数，返回一个computer类型的等价二进制值。

send函数

客户程序和服务器程序都利用send函数通过网络传送数据。

```
int send(connection con, char *buffer, int length, int flags)
```

这个调用使用了4个参数。第一个参数指定先前由await_contact或make_contact建立的连接；第二个参数指定待发送数据的缓冲区地址指针；第三个参数给出待发送数据的字节长度（8位

组）；第四个参数是零（对于正常传输情况）。Send成功时返回已传输的字节数，或传输发生错误时返回负数值。在所有数据发送结束后发送文件结束符（end-of-file），也可参看send_eof函数。

recv和recvln函数

客户程序和服务器程序都是调用recv函数来访问通过网络传送而到达的数据。

```
int  recv(connection con, char *buffer, int length, int flags)
```

这个调用使用了4个参数。第一个参数指定先前由await_contact或make_contact建立的连接，第二个参数是接收数据应该放置的缓冲区地址指针，第三个参数给出缓冲区的大小（以字节为单位）；在正常情况下，第四个参数是零。Recv返回已放置在缓存区的数据字节数，若返回0值则指示end-of-file的到达；若返回负数值则指示错误的发生。实例代码中还使用库函数recvln，它循环地调用recv直到所有文本行接收完毕。recvln被定义如下：

```
int  recvln(connection con ,char *buffer, int length)
```

send_eof函数

客户程序和服务器在发送完数据后，必须使用send-eof函数来通知对方知道后面不再有传输发生。在另一方，当它接收到end-of-file后，recv函数则返回0值。

```
int  send_eof(connection con)
```

这个调用只有一个参数，它指定先前由await_contact或make_contact建立的连接。该函数在发送出现错误时返回负值，否则返回非负值。

API类型归纳

图A-4归纳了范例API中用于各函数的各种参数以及每个参数的类型和函数返回值的类型。图A-4中的最后一列给出了在前两列以外的参数类型。虽然send和recv各自具有4个参数，但库函数recvln则仅有3个参数。

函 数 名	返回值的类型	参数1类型	参数2类型	参数3 & 4类型
await_contact	connection	appnum	—	—
make_contact	connection	computer	appnum	—
appname_to_appnum	appnum	char *	—	—
cname_to_comp	computer	char *		
send	int	connection	char *	int
recv	int	connection	char *	int
recvln	int	connection	char *	int
send_eof	int	connection		

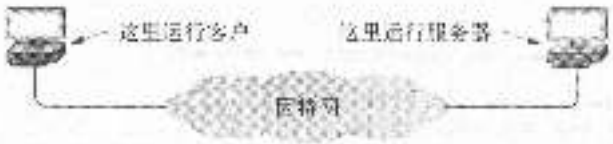
图A-4　范例API的参数及返回值类型的归纳

下面介绍几个应用程序的例子，从中可以了解客户和服务器软件是如何利用上述的API进行通信的。为了减少代码量和更易阅读，本章的程序都使用命令行参数，但没有检查这些参数的有效性。后面的练习题中有一题建议读者重写程序代码，对参数进行检查并将任何可能的错误报告给用户。

Echo应用程序

我们考虑的第一个例子是一个简单的小应用程序：客户发送数据，服务器只对收到的所有数据回送给客户。也就是说，客户应用程序反复地将用户输入的命令行发送给服务器，然后再将服务器回送的内容显示出来。虽然这个程序对一般的用户没有太多的实际用途，但这个Echo应用程序却常常被用来测试网络的连通性。

像附录中描述的所有程序一样，echo程序使用标准的因特网协议，即客户和服务器程序可以运行在连接到因特网的任何主机上，如图A-5所示。



图A-5 可运行在任意主机上的echo客户和服务器

为了调用服务器，用户必须选择一个在1～32767间未被其他应用程序使用的应用程序号，并作为命令行参数来指明此号码。例如，使用域名arthur.cs.purdue.edu主机的某用户选择20 000作为echo程序号，则可通过以下命令行调用服务器程序：

```
echoserver 20 000
```

如果恰好有其他程序已经在使用20 000，那么服务器就会发出适当的出错信息并退出程序，用户必须选择另一个程序号。

一旦服务器程序成功地被调用，就可以通过指定运行服务器的主机名和服务器程序所使用的程序号调用客户程序。例如，为了连接上述的服务器，用户可以在与因特网连接的任意主机上输入命令：

```
echoclient arthur.cs.purdue.edu 20000
```

echo服务器代码示例

echoserver.c是echo服务器的源代码文件。大家可以惊讶地发现该程序是如此简单，连同插入的注释和空行，整个源程序不超过一页。其实，经检查程序以确认可以正确运行后，该程序的主体部分只含有7行代码：

```
/* echoserver.c */

#include <stdlib.h>
#include <stdio.h>
#include <cmaispd.h>

#define BUFSIZE             256

/*------------------------------------------------------------------------
 *
 * Program: echoserver
 * Purpose: wait for a connection from an echoclient and echo data
 * Usage:   echoserver <appnum>
 *
 *------------------------------------------------------------------------
 */
int
main(int argc, char *argv[])
{
```

```
        connection      newsock;
        int             len;
        char            buff[BUFFSIZE];

        if (argc != 2) {
                (void) fprintf(stderr, "usage: %s <appname>\n", argv[0]);
                exit(1);
        }

        /* wait for a connection from an echo client */

        newsock = await_contact((appname) atoi(argv[1]));
        if (newsock < 0)
                exit(1);

        /* iterate, echoing all data received until end of file */

        while((len = recv(newsock, buff, BUFFSIZE, 0)) > 0)
                (void) send(newsock, buff, len, 0);
        send_eof(newsock);
        return 0;
}
```

从上面的程序可以看出，echo服务器采用单个命令行参数来指定应用程序号。在C中，命令行参数是作为一个字符串数组（argv连同一个参数的整数计数量argc）被传递给程序的。程序从argv[1]提取该命令行参数，并调用标准C函数atoi完成将ASCII字符串表示的值转换为二进制形式，然后再将其作为参数传递给await_contact。当调用await_contact返回后，程序循环调用recv从客户端接收数据，并调用send送回相同的数据给客户，当recv发现end-of-file后，停止循环收发并返回0值，此时服务器给客户也发送end-of-file，然后退出程序。

echo客户代码示例

echoclient.c是echo客户应用程序的源代码文件，虽然echo客户程序比echo服务器程序稍长，但也仅有十分有限的代码行。

```
/* echoclient.c */

#include <stdlib.h>
#include <stdio.h>
#include <xinuapi.h>

#define BUFFSIZE                256
#define INPUT_PROMPT            "input   > "
#define RECEIVED_PROMPT         "received> "

int readln(char *, int);

/*------------------------------------------------------------------------
 *
 * Program: echoclient
 * Purpose: contact echoserver, send user input and print server response
 * Usage:   echoclient <servername> [appnum]
 * Note:    Appnum is optional. If not specified the standard echo appnum
 *          (7) is used.
 *
 *------------------------------------------------------------------------
 */
int
main(int argc, char *argv[])
{
        servername      name;
        appnum          app;
        connection      conn;
```

```
    char            buff[BUFFSIZE];
    int             expect, received, len;

    if (argc < 2 || argc > 3) {
            (void) fprintf(stderr, "usage: %s <hostname> [approc]\n",
                           argv[0]);
            exit(1);
    }

    /* convert the arguments to binary format comp and appnum */

    comp = chame_to_comp(argv[1]);
    if (comp == -1)
            exit(1);

    if (argc == 3)
            app = (appnum) atoi(argv[2]);
    else
            if ((app = appname_to_appnum("echo")) == -1)
                    exit(1);

    /* form a connection with the echoserver */

    conn = make_contact(comp, app);
    if (conn < 0)
            exit(1);

    (void) printf(INPUT_PROMPT);
    (void) fflush(stdout);

    /* iterate: read input from the user, send to the server,      */
    /*          receive reply from the server, and display for user */

    while ((len = read(0,buff, BUFFSIZE)) > 0) {

            /* send the input to the echoserver */

            (void) send(conn, buff, len, 0);
            (void) printf(RECEIVED_PROMPT);
            (void) fflush(stdout);

            /* read and print same no. of bytes from echo server */

            expect = len;
                    for (received = 0; received < expect;) {
                      len = recv(conn, buff, (expect - received) < BUFFSIZE ?
                                 (expect - received) : BUFFSIZE, 0);
                            if (len < 0) {
                                    send_eof(conn);
                                    return 1;
                            }
                            (void) write(FILENO, buff, len);
                            received += len;
                    }
                    (void) printf("\n");
                    (void) printf(INPUT_PROMPT);
                    (void) fflush(stdout);
            }

            /* iteration ends when EOF found on stdin */

            (void) send_eof(conn);
            (void) printf("\n");
            return 0;
    }
```

客户程序使用一个或是两个参数，第一个参数指定运行服务器程序的主机名。假如出现第二个参数，就是指定服务器所用的应用程序号。如果第二个参数缺失，则客户程序就使用

参数echo来调用appname_to_appnum。

在将参数转化为二进制形式后，客户程序将它们传递给make_contact，向服务器发出连接请求。一旦连接建立后，客户程序向用户发出提示，然后进入循环：读用户输入行，将输入行发送给服务器，读来自服务器的应答，为用户显示应答信息，接着是新的提示。当客户程序执行到输入结尾的时候（即函数readln返回0值），客户程序就调用send_eof来告知服务器，并退出程序。

客户端代码较之服务器端程序更为复杂是因为以下几个细节问题。首先，客户程序要调用函数readln来读入一行输入；第二，客户程序需要检测每个所调用函数返回的值，当返回值显示发生错误时程序退出；第三，客户要调用函数flush以确保输出立即显示而不是堆积在缓冲区中；第四，客户程序不仅仅是在每次接收来自服务器的数据时调用recv，而是进入一种循环，不断调用recv直至接收到数据字节数和已发出的同样多为止。

对API来说，多次调用函数recv会出现一个关键点：

> 接收者不能假设将要接收的数据块大小会与已发出去的数据块一样大小；对函数recv一次调用返回的数据，可能会少于对函数send的一次调用所发送的数据。

文本解释了为什么调用recv会出现这种现象：因为数据是被分割成更小的分组来传输的。因此，一个应用程序可能一次从一个分组接收到数据，而令人惊讶的是相反也是对的，即使发送方重复地调用函数send，在应用程序调用函数recv之前，网络软件可能接收来自许多分组的数据，在这种情况下，recv将会一次性地返回所有的数据。

聊天应用程序

第二个例子是一个聊天程序的简化形式。互特网聊天程序允许一组用户通过输入文本消息并显示在各自屏幕上进行交互通信。我们提供的简化版聊天程序只支持一对用户——一方输入文本消息，并显示在另一方的显示屏上，反之亦然。因此，就像前述的echo应用程序一样，本例的聊天程序可以运行在连接互特网的任意计算机上。

聊天服务器代码示例

一个用户通过选择一个应用程序号并运行服务器开始，例如，假设用户在主机guenevere.cs.purdue.edu上运行服务器程序：

```
chatserver 25000
```

在另一台计算机上的用户则调用客户程序请求连接服务器：

```
chatclient guenevere.cs.purdue.edu 25000
```

为了让代码尽可能地简短，我们采用让交谈双方轮流输入的方案。当用户期待输入文本行时，客户或服务器端都将出现相应的提示符。首先提示客户端用户输入信息，当收到一行文本后，就将这行文本发送给服务器，依次交替循环，直到有一方发送end-of-file即终止聊天。

本例代码本身简单明了，服务器程序启动后等待客户端请求连接，然后开始进入"接收并显示来自客户的一文本行，提示本地用户输入，读取来自键盘的输入，向客户发送'输入的文本行'"循环。类比，直至收到end-of-file终止聊天。服务器在显示来自客户的输出和向客户发送键盘输入之间不断重复执行。

客户端程序从请求连接服务器开始。一旦通信关系已经建立，客户端程序也进入同样的

循环。其间，客户程序提示本地用户输入一文本行，读取键盘输入，向服务器发送该文本行，然后接收并显示来自服务器的文本行。因此，客户程序也是在向服务器发送用户输入文本行和显示来自服务器的文本行之间一直交替执行。

文件chatserver.c中包含了聊天服务器程序的源代码。

```
/* chatserver.c */

#include <stdlib.h>
#include <stdio.h>
#include <chatapi.h>

#define BUFFSIZE                256
#define INPUT_PROMPT            "Input  > "
#define RECEIVED_PROMPT         "Received> "

int recvln(connection, char *, int);
int readln(char *, int);

/*------------------------------------------------------------------------
 *
 * Program: chatserver
 * Purpose: wait for a connection from a chatclient & allow users to chat
 * Usage:   chatserver <appname>
 *
 *------------------------------------------------------------------------
 */
int
main(int argc, char *argv[])
{
        connection      conn;
        int             len;
        char            buff[BUFFSIZE];

        if (argc != 2) {
                (void) fprintf(stderr, "usage: %s <appname>\n", argv[0]);
                exit(1);
        }

        (void) printf("Chat Server Waiting For Connection.\n");
/* wait for a connection from a chatclient */

conn = await_contact((appname) atoi(argv[1]));
if (conn < 0)
        exit(1);

(void) printf("Chat Connection Established.\n");

/* Iterate, reading from the client and the local user */

while((len = recvln(conn, buff, BUFFSIZE)) > 0) {
        (void) printf(RECEIVED_PROMPT);
        (void) fflush(stdout);
        (void) write(STDOUT_FILENO, buff, len);

        /* send a line to the chatclient */

        (void) printf(INPUT_PROMPT);
        (void) fflush(stdout);
        if ((len = readln(buff, BUFFSIZE)) < 1)
                break;
        buff[len - 1] = '\n';
        (void) send(conn, buff, len, 0);
}

/* Iteration ends when EOF found on stdin or chat connection */
```

```
    (void) send_eof(conn);
    (void) printf("\nChat Connection Closed.\n");
    exit(0);
}
```

使用函数recvln和readln可以简化程序代码——函数recvln和readln分别由一个重复调用的循环构成，直至收到一个完整行或遇到end-of-file才结束循环。Recvln是调用recv来接收来自网络连接的数据，而readln则是调用read以读取来自键盘的字符。

聊天服务器程序的整体结构类似我们前面所述的echo服务器程序，像echo服务器程序那样，聊天服务器程序先期待输入作为应用程序号的一个命令行参数。一旦来自客户的连接请求到达，聊天服务器就向本地用户显示消息，进入重复循环。在每一次重复循环，服务器接收来自网络连接的一个文本行，在屏幕上显示该行消息，读取来自键盘的一文本行，并通过网络发送出去。只有当检测到end-of-file时，服务器才会发送end-of-file并退出程序。

聊天客户代码示例

chatclient.c是chat客户端的源代码文件，与所想象的一样，聊天客户程序代码比服务器程序稍大。

```
/* chatclient.c */

#include <stdlib.h>
#include <stdio.h>
#include <mailcpi.h>

#define BUFFSIZE                256
#define INPUT_PROMPT            "input  > "
#define RECEIVED_PROMPT         "received> "

int recvln(connection, char *, int);
int readln(char *, int);

/*------------------------------------------------------------------------
 *
 * Program: chatclient
 * Purpose: connect a chatserver and allow users to chat
 * Usage:   chatclient <compname> <appnum>
 *
 *------------------------------------------------------------------------
 */
int
main(int argc, char *argv[])
{
        computer        comp;
        connection      conn;
        char            buff[BUFFSIZE];
        int             len;

        if (argc != 3) {
                (void) fprintf(stderr, "usage: %s <compname> <appnum>\n",
                                argv[0]);
                exit(1);
        }

        /* convert the compname to binary form comp */

        comp = cname_to_comp(argv[1]);
        if (comp == -1)
                exit(1);
        /* make a connection to the chatserver */

        conn = make_connection(comp, (appnum) atoi(argv[2]));
        if (conn < 0)
```

```
        exit(1);

    (void) printf("Chat Connection Established.\n");
    (void) printf("HOST_PROMPT");
    (void) fflush(stdout);

    /* Iterate, reading from local user and then from chatserver */

    while((len = read(0,buff, BUFFSIZE) > 0) {
        buff[len - 1] = '\n';
        (void) write(sock, buff, len, 0);

        /* receive and print a line from the chatserver */
        if ((len = recv(sock, buff, BUFFSIZE,0)) < 0)
            break;
        (void) printf("%.*s", len, buff);
        (void) fflush(stdout);
        (void) write(STDOUT_FILENO, buff, len);

        (void) printf("%s", HOST_PROMPT);
        (void) fflush(stdout);
    }

    /* Iteration ends when either of the connection indicates EOF */

    (void) printf("\nChat Connection Closed.\n");
    (void) send_eof(sock);
    exit(0);
}
```

客户端以请求连接服务器开始。一旦通信关系已经建立，客户程序进入如下循环：读取键盘输入，向服务器发送该数据，接收来自服务器的文本行并将其显示在用户屏幕上。这个循环一直重复到客户收到来自服务器或来自键盘的end-of-file条件（返回零值）为止，这时向服务器发出end-of-file，并退出程序执行。

Web应用程序

附录中讨论的最后一个例子是万维网（World Wide Web）的客户—服务器交互。为了运行服务器，用户需选择一个应用程序号，并调用服务器程序。例如，netbook.cs.purdue.edu计算机的用户选择应用程序号27000，则可用下面的命令来调用服务器：

webserver 27000

正如想象那样，客户程序的命令要指明：计算机、路径名（网页文件在服务器主机上）和应用程序号：

webclient netbook.cs.purdue.edu /index.html 27000

虽然Web服务器极其简单，但它也是遵循标准协议的，因此它可以支持传统的（例如，商业化应用的）Web浏览器来访问服务器。例如，可使用商业浏览器而不是我们示例中的Web客户程序，输入URL：

http://netbook.cs.purdue.edu:27000/index.html

为了保持程序代码尽可能的简短，我们做了一些简化假设。例如，服务器只提供3个Web网页，且网页仅含纯文本；而且，每个页面的内容都是硬性插入到程序代码中的。要改变其内容必须重新编译服务器程序才能生效（附录练习中建议扩展这个服务器程序代码，以便克服以上的限制）。

这里的Web应用程序中最重要的限制是在客户程序上。并不像传统的Web浏览器，我们这里提出的客户程序代码并不能解析网页的格式和显示网页，而只能显示网页的源代码，虽然有此限制，但我们的客户程序却可以和商业Web服务器互相操作，并可显示任何Web页面源代码。

Web客户代码示例

webclient.c是Web客户程序的源代码文件。

```
/* webclient.c */

#include <stdlib.h>
#include <stdio.h>
#include <sockpol.h>

#define BUFFSIZE        255

/*------------------------------------------------------------------------
 *
 * Program: webclient
 * Purpose: fetch page from webserver and dump to stdout with headers
 * Usage:   webclient <computer> <path> [appcon]
 * Note:    Appcon is optional. If not specified the standard www appcon
 *          (80) is used.
 *
 *------------------------------------------------------------------------
 */
int
main(int argc, char *argv[])
{

        computer        comp;
        appcon          app;
        connection      conn;
        char            buff[BUFFSIZE];
        int             len;

        if (argc < 3 || argc > 4) {
                (void) fprintf(stderr, "webclient", "usage: ", argv[0],
                                " <computer> <path> [appcon]\n");
                exit(1);
        }

        /* convert arguments to binary computer and appcon */

        comp = name_to_comp(argv[1]);
        if (comp == -1)
                exit(1);

        if (argc == 4)
                app = (appcon) atoi(argv[3]);
        else
                if ((app = appname_to_appcon("www")) == -1)
                        exit(1);

        /* connect the web server */

        conn = make_contact(comp, app);
        if (conn < 0)
                exit(1);

        /* send an HTTP/1.0 request to the webserver */

        len = sprintf(buff, "GET %s HTTP/1.0\r\n\r\n", argv[2]);
        (void) send(conn, buff, len, 0);

        /* dump all data received from the server to stdout */

        while((len = recv(conn, buff, BUFFSIZE, 0)) > 0)
                (void) write(STDOUT_FILENO, buff, len);

        return 0;
}
```

Web客户程序代码极其简单——在建立与Web服务器的通信关系后，客户向服务器发出请求。其形式必须是：

```
GET/path HTTP/1.0 CRLF CRLF
```

这里的*path*是指某个消息项（如index.html）所在主机上的路径名；而*CRLF*是指回车和换行符。在发出Web页面访问请求后，客户程序即可等待接收并显示来自服务器的输出。

Web服务器代码示例

webserver.c是微小型的Web服务器源代码文件。程序包含了3个Web页面源码，以及响应请求所需的代码。

```
/* webserver.c */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <sysinet.h>

#if defined(LINUX) || defined(SOLARIS)
#include <sys/time.h>
#endif

#define BUFFSIZE        256
#define SERVER_NAME     "CMAI Demo Web Server"

#define ERROR_400       "<html><head></head><body><h1>Error 400</h1><p>Th\
e server couldn't understand your request.</body></html>\n"

#define ERROR_404       "<html><head></head><body><h1>Error 404</h1><p>Do\
cument not found.</body></html>\n"

#define HOME_PAGE       "<html><head></head><body><h1>Welcome to the CMAI\
 Demo Server</h1><p>Why not visit: <ul><li><a href=\"http://netbook.cs.pu\
rdue.edu\">Netbook Home Page</a><li><a href=\"http://www.comerbooks.com\">
>Comer Books Home Page</a></ul></body></html>\n"

#define TIME_PAGE       "<html><head></head><body><h1>The current date is\
: %s</h1></body></html>\n"

int     recvln(connection, char *, int);
void    send_head(connection, int, int);

/*-----------------------------------------------------------------------
 *
 * Program: webserver
 * Purpose: serve hard coded webpages to web clients
 * Usage:   webserver <approx>
 *
 *-----------------------------------------------------------------------
 */
int
main(int argc, char *argv[])
{

        connection      conn;
        int             n;
        char            buff[BUFFSIZE], cmd[16], path[64], vers[16];
        char            *timestr;
#if defined(LINUX) || defined(SOLARIS)
        struct timeval  tv;
#elif defined(WIN32)
        time_t          tv;
#endif
```

```
        if (argc != 2) {
                (void) fprintf(stderr, "usage: %s <approach>\n", argv[0]);
                exit(1);
        }

        while(1) {
                /* wait for connect from a client on specified approach */

                conn = await_contact((approach) atoi(argv[1]));
                if (conn < 1)
                        exit(1);

                /* read and parse the request line */

                n = recvln(conn, buff, BUFFSIZE);
                sscanf(buff, "%s %s %s", cmd, path, vers);

                /* skip all headers - read until we get \r\n alone */

                while((n = recvln(conn, buff, BUFFSIZE)) > 0) {
                        if (n == 2 && buff[0] == '\r' && buff[1] == '\n')
                                break;
                }

                /* check for unexpected end of file */

                if (n < 1) {
                        (void) send_eof(conn);
                        continue;
                }

                /* check for a request that we cannot understand */

                if (strcmp(cmd, "GET") || (strcmp(vers, "HTTP/1.0") &&
                                           strcmp(vers, "HTTP/1.1"))) {
                        send_head(conn, 400, strlen(ERROR_400));
                        (void) send(conn, ERROR_400, strlen(ERROR_400),0);
                        (void) send_eof(conn);
                        continue;
                }

                /* send the requested web page or a "not found" error */

                if (strcmp(path, "/") == 0) {
                        send_head(conn, 200, strlen(HOME_PAGE));
                        (void) send(conn, HOME_PAGE, strlen(HOME_PAGE),0);
                } else if (strcmp(path, "/time") == 0) {
#if defined(LINUX) || defined(SOLARIS)
                        gettimeofday(&tv, NULL);
                        timeptr = ctime(&tv.tv_sec);
#elif defined(Win32)
                        time(&tv);
                        timeptr = ctime(&tv);
#endif
                        (void) sprintf(buff, TIME_PAGE, timeptr);
                        send_head(conn, 200, strlen(buff));
                        (void) send(conn, buff, strlen(buff), 0);
                } else { /* not found */
                        send_head(conn, 404, strlen(ERROR_404));
                        (void) send(conn, ERROR_404, strlen(ERROR_404),0);
                }
                (void) send_eof(conn);
        }
}

/*------------------------------------------------------------------------
 * send_head - send an HTTP 1.0 header with given status and content-len
```

```
 *------------------------------------------------------------------------
 */
void
send_head(connection conn, int stat, int len)
{
        char    *statstr, buff[MAXBUFSIZE];

        /* convert the status code to a string */

        switch(stat) {
        case 200:
                statstr = "OK";
                break;
        case 400:
                statstr = "Bad Request";
                break;
        case 404:
                statstr = "Not Found";
                break;
        default:
                statstr = "Unknown";
                break;
        }

        /*
         * send an HTTP/1.0 response with Server, Content-Length
         * and Content-Type headers.
         */

        (void) sprintf(buff, "HTTP/1.0 %d %s\r\n", stat, statstr);
        (void) send(conn, buff, strlen(buff), 0);

        (void) sprintf(buff, "Server: %s\r\n", SERVER_NAME);
        (void) send(conn, buff, strlen(buff), 0);

        (void) sprintf(buff, "Content-Length: %d\r\n", len);
        (void) send(conn, buff, strlen(buff), 0);

        (void) sprintf(buff, "Content-Type: text/html\r\n");
        (void) send(conn, buff, strlen(buff), 0);

        (void) sprintf(buff, "\r\n");
        (void) send(conn, buff, strlen(buff), 0);
}
```

虽然Web服务器程序看起来比前面几个例子要复杂些，但是大部分复杂性只是为了描述网页的细节而不是网络操作上的细节问题。除了读取和解析请求外，Web服务器必须在响应中发送“头部”和响应数据。“头部”由以回车和换行符终止的多个文本行组成，其形式如下：

```
HTTP/1.0 status_nnnn_string CRLF
Server: CNX Demo Server CRLF
Content-Length: datasize CRLF
Content-Type: text/html CRLF
CRLF
```

其中，datasize表示以字节度量的数据长度。

过程send_head负责处理生成“头部”的相关细节。当调用send_head时，参数stat包含整数状态码，参数len指定内容长度；switch语句则是利用相关代码来选择一个适当的由变量statstr赋值的文本消息。过程send_head使用C函数sprintf在缓冲区中生成一个完整的“头部”，然后再调用send将“头部”行通过连接发送给客户端。

以上代码由于出错处理而变得复杂——出错报文必须是以浏览器可以理解的形式发送出去。如果客户端的请求不符合规则，则服务器产生出错报文400；如果客户请求的数据不能找

到（如路径不正确），则服务器产生出错报文404。

这个Web服务器程序与前面例子相比，有一个重要的不同之处是：服务器程序在给一个客户提供应答后不必退出，相反，服务器继续运行等待接受其他的请求，也就是服务器程序进入一个调用awati_contact等待客户连接请求的死循环。当连接达到时，服务器调用recvln来接收请求，并调用send发送响应，然后服务器退回循环的顶部等待下一次连接请求。因此，就像一般的商用Web服务器一样，一旦启动就永久地运行下去。

用Select函数管理多连接

虽然我们的范例API支持客户-服务器间的"一对一"交互操作，但该API并不能支持"一对多"交互。为了理解其中的原因，我们要考虑多连接问题。为了生成这样的连接，应用程序必须多次调用make_contact，并将所有连接都指向一个computer和appnum。一旦已经建立了多个连接，可是应用程序却不知道其中的哪个连接首先接收数据。应用程序不能调用recv，因为在数据到达之前这个调用是被阻塞的。

很多操作系统都提供select函数用来解决多连接的管理问题。从概念上来讲，可以调用select来检查每个连接，在所有确定的连接中至少有一个已开始接收数据才能启动调用recv，然后Select函数返回一个值，告诉哪个连接已经接收到数据（也就是说，在该连接上可以开始调用recv函数了）。

作为一个例子，以下我们考虑一个在两个连接上分别接收请求和发送响应的应用实例，该应用程序代码具有如下的一般形式：

```
Call make_connect to form connection 1;
Call make_connect to form connection 2;
Repeat forever {
    Call select to determine which connection is ready
    If (connection 1 is ready) {
        Call recv to read request from connection 1;
        Compute response to request;
        Call send to send response over connection 1;
    } if (connection 2 is ready) {
        Call recv to read request from connection 2;
        Compute response to request;
        Call send to send response over connection 2;
    }
}
```

附录小结

在不理解网络底层技术如何在计算机间传递数据和网络操作细节的情况下，程序员编写网络应用程序是完全可能的，但必须给程序员提供形成应用编程接口（API）的一组高层函数。附录中所介绍的网络API只含有7个原语型函数，并综述了几个应用程序示例，表明这套API在开发与商业软件进行正确互操作的软件方面能满足基本的要求。

练习题

A.1 附录中的程序代码不能仔细地检查输入的命令行参数。请你修改代码以增加错误检查功能。

A.2 echo服务是用于调试的标准服务，它的程序号是7。请下载并编译echo客户软件，然后

用它来确定你单位的计算机是否运行一个echo服务器。

A.3 请修改echo服务器程序以使其在处理echo客户后不退出，而是等待另一个echo客户的请求。提示：参看Web服务器程序。

A.4 请下载、编译附录聊天软件示例，并运行在两台计算机上进行测试。

A.5 附录聊天软件需要用户轮流输入文本。请重新编写聊天程序使其允许两端用户可以在任意时刻输入任意行文本。提示：使用多线程。

A.6 请修改聊天客户程序使其伴随每个消息都发送用户姓名；修改聊天服务器程序使其在显示文本行时能识别发送的用户。

A.7 扩展上述练习，不是每个消息伴随发送用户姓名，而是聊天客户和服务器在第一次连接时就相互交换姓名，并记住双方姓名，然后在每行文本输出时显示用户姓名。

A.8 为什么附录中的应用示例代码混合使用write调用和各种形式的printf？提示：Windows也同样处理套接字、文件和管道吗？

A.9 设计一种多方聊天会话软件，它允许用户任意地加入和离开聊天会话。

A.10 使用远程登录telnet连接Web服务器，发送GET请求，并接收应答。

A.11 请尝试利用附录中的Web客户程序与因特网Web服务器联系。为此，你必须给出服务器域名、index.html或 index.htm的路径和应用程序端口号80。

A.12 对附录中的Web服务器程序增加请求另一个网页的功能。

A.13 请修改附录中Web服务器程序，改为从文件中提取每个网页内容，而不是从特定程序代码中提取网页的方法。

A.14 扩展上题练习，以便能辨认.gif结尾的文件，并使用带有image/gif值（而不是原来的text/html）的头部content-type，将其发送。

A.15 （高级题）编写一个完成文件传输服务的客户和服务器程序。

A.16 （高级题）实现通用信关接口CGI，其详细说明请查看：

http://hoohoo.ncsa.uiuc.edu/cgi/

A.17 （高级题）扩充Web服务器程序，以使其能并发处理多连接。提示：使用函数fork 或 pthread_create。

A.18 （高级题）编写一个能与SMTP电子邮件服务器连接并发送电子邮件报文的客户程序。