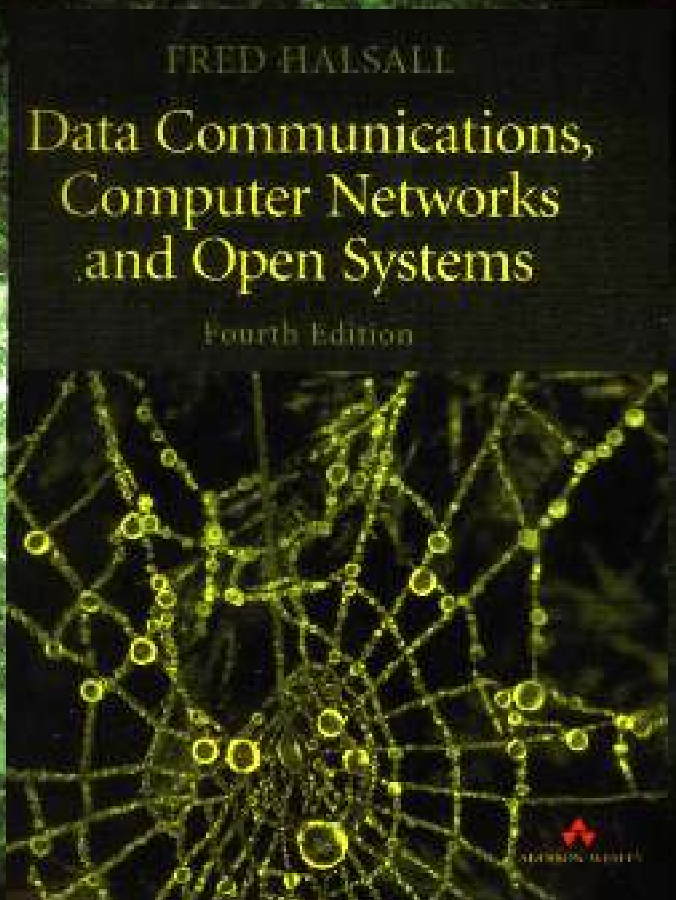
计算机科学丛书

原书第4版

数据通信、计算机网络与开放系统

（英）Fred Halsall 著　吴时霖　吴永辉　魏霖　等译



Data Communications, Computer Networks and Open Systems Fourth Edition

机械工业出版社
China Machine Press

在本书中，作者根据其多年的研究和教学经验，对复杂的数据通信和计算机网络领域技术进行了清楚而透彻的讲解。本书的第4版在原有基础上，增加了对数据通信、计算机网络和开放系统领域新技术的介绍，以反映该领域的最新进展，学生或者专业技术人员都能从中获益匪浅。

本书中包含的重要主题如下：

- 包含数字传输的基本理论
- 数字租用电路，包括PDH、SONET和SDH
- 协议基础，包括协议的规范说明和实现方法
- 遗留LAN和无线局域网
- 高速局域网，包括100 Base T和100 VG AnyLAN
- 透明源路由选择网桥
- 包交换和帧中继网络，及其使用的协议
- 多业务宽带网络，包括ATM LAN和MAN
- 网际互连结构、协议和路由选择算法
- TCP/IP和OSI应用协议，包括X.400和X.500
- 数据加密和网络安全算法
- 网络管理体系结构，包括SNMP和CMIP

作者简介

Fred Halsall 是英国威尔士大学的通信工程教授。他在通信工程领域有20多年的研究经验，发表过大量的著作。

ISBN 7-111-12212-7

9 787111 122128

封面设计 刘永青

HZ Books

华章图书

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037

读者服务热线：(010)68995259，68995264

读者服务信箱：hzedu@hzbook.com

http://www.hzbook.com

ISBN 7-111-12212-7/TP · 2693

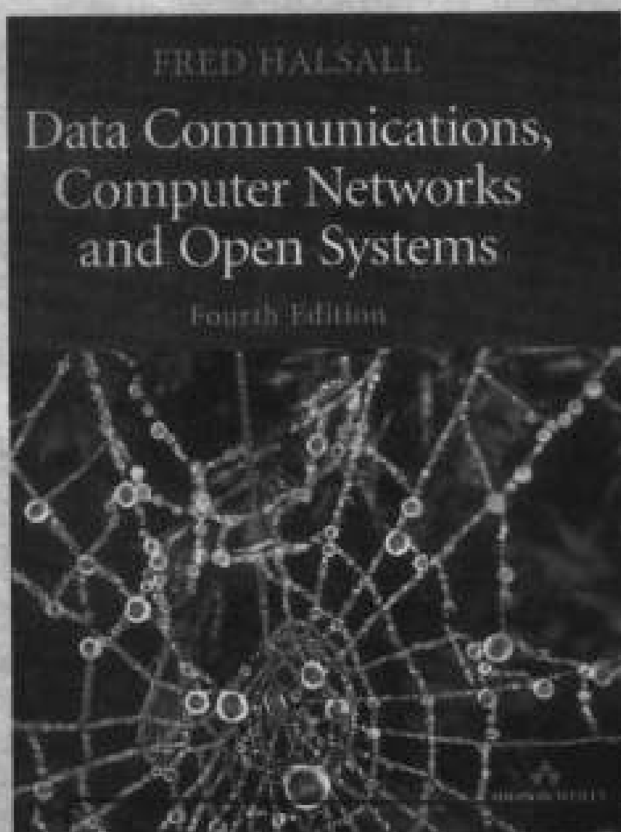
定价：69.00 元

计 算 机 科 学 丛 书

原书第4版

数据通信、计算机网络与开放系统

(英) Fred Halsall 著 吴时霖 吴永辉 魏霖 等译



Data Communications, Computer Networks and Open Systems

Fourth Edition

机械工业出版社
China Machine Press

本书是一本颇有影响的计算机通信与网络教材，它全面地介绍了近些年来数据通信领域，特别是计算机网络领域的一些重大进展。全书由三个相对独立的部分组成：数据通信、计算机网络与开放系统。第一部分主要讨论在串行数据链路中数据转换的基本问题；第二部分讨论计算机网络并描述不同类型的计算机网络的操作；第三部分介绍了附加协议的操作。

本书适合作为计算机、通信、电子工程等专业本科生、研究生的教材，也适合相关专业的技术人员参考。

本书版权登记号：图字：01-2002-1888

图书在版编目（CIP）数据

数据通信、计算机网络与开放系统（原书第4版）/（英）哈尔索尔（Halsall, F.）著；吴时霖等译. -北京：机械工业出版社，2004.1

（计算机科学丛书）

书名原文：Data Communications, Computer Networks and Open Systems, Fourth Edition

ISBN 7-111-12212-7

Ⅰ. 数… Ⅱ. ① 哈…② 吴… Ⅲ. ① 数据通信－教材 ② 计算机网络－教材 Ⅳ. ① TN919 ② TP393

中国版本图书馆CIP数据核字（2003）第039009号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李 炎

北京瑞德印刷有限公司印刷 · 新华书店北京发行所发行

2004年1月第1版第1次印刷

787mm × 1092mm1/16 · 45.75印张

印数：0 001-5 000册

定价：69.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线电话：（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，不仅擘划了研究的范畴，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray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、研究及庋藏。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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。华章公司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：

电子邮件：hzedu@hzbook.com
联系电话：(010) 68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

译 者 序

本书是一本颇有影响的计算机通信与网络的教材，自1985年第1版问世后，经历1988年第2版，1992年第3版，1995年第4版不断的修订再版。本书是按第4版翻译，该书尽可能全面地介绍了这些年来数据通信领域，特别是计算机网络领域一些重大的进展。全书以三个相对独立部分组成：数据通信、计算机网络与开放系统，面向电子工程、计算机系统和计算机科学专业的本科生和研究生，也适用于开发数据和计算通信系统的工程技术人员。

自第1版、第2版以来，本书在国内也有较大的影响，一些高等院校与研究部门按照该书的内容与组织方式编写了通信、计算机等专业本科生和研究生的教材，以及各种研讨班、专业讲座的讲授内容，深受欢迎。本书有三个主要特点：第一，在协议描述方面使人们了解协议如何以程序代码实现；第二，不同协议之间如何相互作用和通信来完成特定分布式信息处理；第三，采用大量的图表配合文字说明，便于更直观地理解细节。

本书前言、第1～5章由复旦大学计算机科学与工程系吴永辉、吴时霖翻译，第6～10章由上海阿尔卡特网络资源系统有限公司魏霖与复旦大学计算机科学与工程系周正康翻译，第11～14章、附录、索引和缩略语由复旦大学计算机科学与工程系吴时霖、徐公权、胡绍鑫翻译。全书最后由吴时霖进行审校，本书涉及学科广，限于水平，翻译不妥或错误在所难免，敬请广大读者批评指正。

吴时霖

2002年12月于复旦大学

前　言

目标

现在计算机可以说无处不在：在家庭里、在办公室里、在银行里、在中小学和大专院校里，等等。虽然在有些情况下，计算机能以单机的方式来完成任务，但在许多情况下，计算机之间必须互相交换信息。这意味着，目前所安装的大多数形式的计算设备，设计时要考虑数据通信设施的类型，使得该设备能与其他计算机进行通信。在许多情况下，我们不仅要了解可以使用哪些类型的数据传输线路，而且要了解不同类型的计算机通信网络的接口需求。在目前有关计算机系统设计的课程中，数据通信和计算机网络已经成为重要的主题。

在计算机网络的大量应用中，提供在两个系统之间交换信息的方法只能解决部分问题。例如，异构的（即不同的）计算机组成的分布式系统要在网络上交换信息文件，就必须解决在系统间以不受约束的（开放的）方式进行通信的情况下，使用不同的操作系统（进而是文件系统）和不同的字符集合和字长的问题。在计算机网络应用中了解各种面向应用的通信协议也是必要的，这些通信协议用于建立通信环境，在这样的通信环境中，来自不同制造商的计算机可以以开放的方式交换信息。本书的三个部分——数据通信、计算机网络和开放系统——涵盖了所有这些问题。

预期的读者

本书主要作为学生学习数据通信、计算机网络和计算机通信协议课程的教科书，供电子工程、计算机工程、计算机系统和计算机科学专业的学生使用。此外，也适合于希望获得相关内容的操作知识的工程师和计算机专业人员。

在大多数大专院校中，书中涉及的主题分设在教学过程中的不同课程，有些内容设在本科生课程中，而有些内容设在研究生课程中。因此，除了在基础课程中，课程安排不必过多地考虑背景知识。本书的先导课程是基本逻辑电路和计算机体系结构，并且有使用高级结构化程序设计语言进行实际项目开发的工作经验。

本书没有过多涉及数字通信理论，因为这部分内容主要属于电子工程领域。本书在开始部分简要介绍了用于传输数据的不同类型传输介质和确定最大数据传输速率的关键理论。然后描述了接口部件具有的功能，这些部件用在每个计算机上，通过基本数据传输设备在两个计算机间实现可靠的数据传送。这包括用于检测接收数据中出现差错的不同方法以及要求发送计算机再发送数据拷贝的规程。这些规程构成了数据链路协议，并且在许多不同的协议中使用。

本书首先阐述了通过物理数据链路连接在一起的计算机如何实现可靠数据传送。要实现多台计算机之间的数据传送，就要通过计算机网络。所以在后续的章节里，首先描述了不同类型的计算机网络，并讨论了与每种网络类型相关联的接口电路的操作及其相关通信协议；然后描述了使得两台或多台计算机连接到计算机网络中以完成特定的分布式应用功能的附加协议。

实际上，在数据传输层之上的通信协议主要是以软件来实现的。所以，在讨论各种通信协议的应用时，除了量化描述其操作外，还要描述协议是如何实现的。而且，对于读者，了解这些协议之间如何协作及互相通信，以完成整个通信功能也是同样重要的。本书将详细讨论这些问题。

本书结构

本书严格参照开放系统互连的ISO参考模型。第一部分“数据通信”主要讨论在串行数据链路中实现可靠数据传送的基本问题。

第二部分“计算机网络”讨论计算机网络并描述不同类型计算机网络的工作原理，这是分布式环境下计算机之间进行通信的基础。

第三部分“开放系统”介绍并描述了网络协议，这是计算机上运行的分布式应用进程能以开放的方式交换数据的基础。这使得在各种计算机提供的字符集、字的大小或者服务方式之间存在差异的情况下，能实现大量的分布式处理功能。

本版中的新内容

从本书的上一版问世以来，在数据通信领域，特别是计算机网络领域，取得了许多重大的进展。所以，本版的主要目的就是将这些进展结合进来，为读者带来最新的内容。并且，作者还给出了针对大学生和研究生的有关本书三个主题的讲座课程和许多用于实际工程师和计算机学者的自学课程。这些内容也在新版中。

在数据通信领域，许多公用通信公司开始提供新一代的数字化租用电路，这些电路构成部分新的同步光纤传送网（SONET）——也称为同步数字系列（SDH）。有关这些内容的阐述在数据传输的章节中。此外，数字传输的基本理论也在这一章中阐述，使读者更加充分地了解数据通过不同的介质类型时的速率限制因素。

在计算机网络领域，将无线电和光作为传输介质的局域网（LAN）已经在使用，也称为无线LAN。本书完整地阐述了无线LAN及其操作特性。对传输速率的更高需求导致高速LAN的产生。除了FDDI外，还有两种CSMA/CD（以太）LAN——100 Base 4T和100 Base X——和新的LAN类型（100 Base VG-AnyLAN）。

无论无线LAN还是各种高速LAN，现有的LAN提供的服务是相似的。然而，新一代的计算机网络不仅提供了数据服务，而且提供了对不同介质间数据传送的支持，例如音频和视频。这些网络称为宽带多业务网，本书专门有一章介绍这类网。这些网络使用不同于现有LAN的操作模式，称为异步传输模式（ATM）。这些网络包括ATM LAN和ATM城域网（MAN）。MAN现在已通过公用通信公司开发出来，用于城镇之间LAN的互连。

使用方法

对于教师

本书涉及三门相对独立的课程：电子工程、计算机系统和计算机科学专业的本科生和研究生课程，也适合于开发数据和计算机通信系统的工程技术人员。本书仅仅要求读者具备基本的知识：基本的逻辑电路和计算机体系结构，并且有用高级程序设计语言进行结构化设计的实践经验。

使用本书以前版本的读者将本书与其他同类书籍进行了比较，总结出三个主要优点。第

一，在协议的描述上，使学生了解协议如何以程序代码来实现；第二，本书解释了构成协议族的不同协议之间如何交互和通信以完成特定的分布式信息处理任务；第三，采用大量的图形解释细节，并减少备课时间。在新版中我们继续保持了这三个优点。

本书内容丰富，分别涉及数据通信、计算机网络和开放系统。数据通信的课程主要在第一部分，另外附录A介绍了正向差错控制的内容。如果学生具有必要的背景知识，也可以学习附录B中有关传输控制电路的内容。

计算机网络的完整课程在第二部分。如果单独讲授计算机网络课程，可以加上第11章中有关传输协议的内容。第三部分阐述开放系统的面向应用协议的主要内容，这些内容包括TCP/IP和OSI协议族。如果单独讲授开放系统，可加上第9章中有关网际互连的内容。第一部分和第二部分的内容加起来可以作为数据通信与计算机网络的课程；第二部分和第三部分的内容加起来可以作为计算机通信与开放系统的课程。

对于学生

本书可用于自学，许多章节包含实例，并用大量插图进行解释。在每一章的结尾有大量的练习，用于测试学生的理解程度。

本书在每章的结尾以图形的方式总结每章知识点及其顺序，这些图形不仅直观地展示了该章知识点之间的相互关系，也提供了该章内容与其他章节内容的关系。

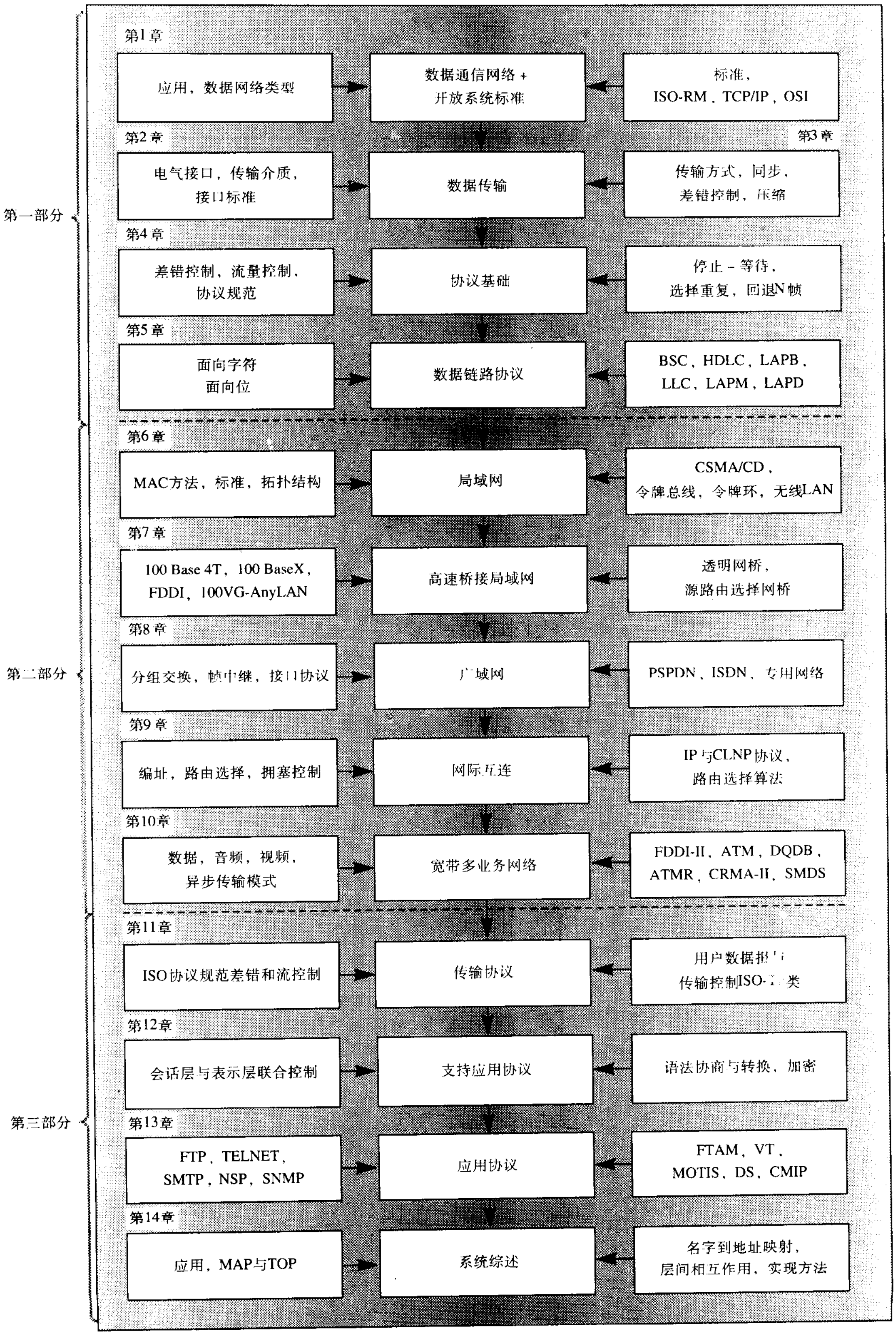
感谢

借此机会，我感谢对本书以前的版本提出意见和建议的同仁，他们使我在新版中对结构和内容进行改进：A. Houghtòn, Sheffield大学, 英国；G. Tagg, Oxford Brookes大学, 英国；A. Koelmans, Newcastle大学, 英国；L. MacKenzie, Glasgow大学, 英国；S. Benson, Staffordshire大学, 英国；R. Newman-Wolfe, Flordia大学, 美国；B. Veenendaal, Curtin大学，美国；A. Ruighauer, Melbourne大学，美国；J. Silvester, Southern California大学, 美国；A. Shaout, Michigan大学, 美国；D. Jacobson, Iowa州立大学, 美国；J. Jormakka, Helsinki技术大学，芬兰；T. Karvi, Helsinki大学，芬兰；T. Bellika, Finnmark学院，芬兰；S. Knapskog, Trondheim大学，挪威；P. Vestøl, Agder工程学院，挪威；L. Christoff和T. Walasek, Uppsala大学，瑞典。特别感谢Queen Mary and Westfield学院的S. Wilbur博士和她的学生（F. Ojuri, A. Killick, R. Payne, M. McDonald, T. Blomfield, A. Plewes, B. Robson）。我也感谢如下对新版书提出建设性意见的同仁：Allan Fisher, Carnegie Mellon大学, 美国；Gong Su, Columbia大学，美国；Andrew Scott, Lancaster大学，英国；Jon Crowcroft, College London大学，英国；Ian Whitworth, Cranfield大学，英国。

最后，我要借此机会表达我对Irene Dendle的诚挚的感谢，她帮助我作了手稿的整理和大量的校对工作；我要感谢我的研究助手，他们帮助整理了大量与新内容有关的论文和文献；感谢我的妻子Rhiannon在我写书的时候给予我的支持与理解，谨以此书献给她。

Fred Halsall

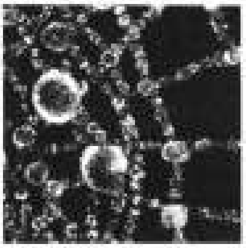
1995年9月


第一部分
第1章
应用，数据网络类型
数据通信网络＋
开放系统标准
标准，
ISO-RM，TCP/IP，OSI
第2章
电气接口，传输介质，
接口标准
数据传输
第3章
传输方式，同步，
差错控制，压缩
第4章
差错控制，流量控制，
协议规范
协议基础
停止－等待，
选择重复，回退N帧
第5章
面向字符
面向位
数据链路协议
BSC，HDLC，LAPB，
LLC，LAPM，LAPD
第二部分
第6章
MAC方法，标准，拓扑结构
局域网
CSMA/CD，
令牌总线，令牌环，无线LAN
第7章
100 Base 4T，100 BaseX，
FDDI，100VG-AnyLAN
高速桥接局域网
透明网桥，
源路由选择网桥
第8章
分组交换，帧中继，接口协议
广域网
PSPDN，ISDN，专用网络
第9章
编址，路由选择，拥塞控制
网际互连
IP与CLNP协议，
路由选择算法
第10章
数据，音频，视频，
异步传输模式
宽带多业务网络
FDDI-II，ATM，DQDB，
ATMR，CRMA-II，SMDS
第三部分
第11章
ISO协议规范差错和流控制
传输协议
用户数据报
传输控制ISO-
第12章
会话层与表示层联合控制
支持应用协议
语法协商与转换，加密
第13章
FTP，TELNET，
SMTP，NSP，SNMP
应用协议
FTAM，VT，
MOTIS，DS，CMIP
第14章
应用，MAP与TOP
系统综述
名字到地址映射，
层间相互作用，实现方法

目 录

第二部分 计算机网络

第三部分 开放系统



第一部分 数据通信

数据通信和早期的计算技术几乎是一起发展起来的。然而，虽然我们对与计算机相关的基本术语和设备，诸如二进制位和字节，门和干线，BASIC和Pascal等等已经有所了解；但是，对于与数据通信相关的基本技术和术语可能并不了解。因此在描述不同类型的计算机网络之前，本书的第一部分概述了作为计算机网络所有形式基础的数据通信的基本概念和术语，讨论了数字传输的基本理论和在两台计算机之间实现可靠数据传输所采取的技术。物理上分离的两台计算机之间的距离可以从几十米，如一个办公室或实验室中的两台计算机，到上百公里，如通过电话网传输通路连接的两台计算机。

第1章阐述了由技术的发展导致的分布式计算系统的演化历史；并能够识别分布式计算系统中应用的数据通信网络，然后阐述了每类网络的应用领域和已经定义并在网络中应用的标准。该章奠定了全书的基础。

第2章论述了各类物理传输介质的电气特性以及确定其用途的规则和理论，并且，该章描述了为数据编码和设备与不同类型的介质进行交互而定义的各种国际标准。

众所周知，在两个设备之间的数据传输是按块进行的按位串行传输，这些块包含不同数目的二进制位。第3章首先讨论接收方如何确定传输过来的块从何处开始，又在何处结束；以及在传输过程中发生差错（位受损）时的各种处理技术。该章也讨论数据压缩的问题和不同类型的多路转换设备。

第4章给出已经采用的各种技术。首先，总结了为解决传输差错的影响而做的工作；其次， 1[⊖]
讨论如何控制通过数据链路的数据流速率。这两种功能是数据链路层相关联的协议的组成部分。这一章也阐述了协议是如何说明的，以及用程序代码实现协议的方法。

第5章是在第4章介绍的一般原理的基础上，阐述标准的数据链路层协议，这些数据链路协议已经广泛地使用，对基于分布式计算机系统的部门之间的数据交换进行控制。 2

⊖ 本书边栏号码为该书原书页码，与书末索引中的页码相呼应。

第1章　数据通信网络和开放系统标准

本章目的

读完本章，应该能够：

- 识别计算机通信网络的不同应用；
- 评价在上述应用中使用的数据通信网络的各种类型；
- 理解分层的概念和开放系统互连的ISO参考模型的结构；
- 描述ISO参考模型中每一层的功能；
- 了解与TCP/IP协议集相关的协议以及这些协议如何与ISO参考模型相关联；
- 了解基于ISO协议的某些标准协议集。

背景

现在计算机已经应用于生活中的各个方面。在家中，用计算机玩游戏和进行文字处理；在办公室，用计算机进行文字处理，管理电子表格和数据库；在银行和其他金融机构，用计算机管理客户账目；在旅行社，用计算机处理订票和其他预订业务；在中小学校和专科学校，用计算机进行辅助教学；在大学和其他研究机构，将计算机用于科研和实验数据的分析；在加工业，将计算机用于化工厂及其他工厂的控制；在制造业，可以用计算机控制机床和机器人；在百货商店，用计算机对售货点的账目进行处理；等等。

虽然在许多情况下，计算机是以单机的方式工作，但在其他情况下，需要对计算机进行互
3 连，以与其他计算机交换数据。例如，在普通的家庭里，将数据文件从一台个人计算机传送到另一台个人计算机，或者通过电话交换网访问公共数据库上的信息；在办公室内，要在内部或部门之间交换电子邮件；在金融系统内将资金从一个机构通过计算机和网络转到另一个机构；旅游公司要访问各个航空公司的订票系统；在中小学和专科学校里，多台计算机通过网络共享如激光打印机等贵重的设备；在大学和其他研究机构里，访问远端的超级计算机以获取计算的结果；在加工业，用于协调工厂内的仪器设备的控制；在制造业，用于控制把零件及相关数据从一个自动化装置传送到另一个；在百货商店对库存进行控制和进行自动订货。

本章主要论述在上述应用中两台计算机间进行数据通信时必须考虑的问题。本书不仅使读者了解目前应用的各种数据网络和计算机网络，而且了解计算机与网络接口的软硬件的工作细节。此外，本书还描述了在不同类型的计算机上以不同的字长和字符集运行的应用程序如何协调，以实现特定的分布式应用功能。图1-1给出了要阐述的三种基本的通信功能。

对于涉及两台以上计算机的应用，提供合适的数据通信设备是最基本的需求。然而，在实际中可以应用的通信设备很广泛，每类通信设备针对特定的应用领域。例如，如果是很简单的要求，只是将一台计算机中的数据文件传送到同一房间或同一办公室的另一台相同类型
4 的计算机中，那么通信设备就比在不同地点不同类型的计算机间传送数据要简单。

无论采用哪种数据通信设备，在大多数应用中，计算机间的数据传送主要以**位串行方式**进行。由于在一台计算机内的子系统之间的数据传送是以**字并行方式**进行的，因此在数据输出之前，在计算机与网络的接口上，必须执行**并串行转换**操作，而在输入之前执行相反的**串**

并行转换操作。此外，传输方式的类型和所需的电路是变化的，并且依赖于计算机的物理间隔和数据传输速率。

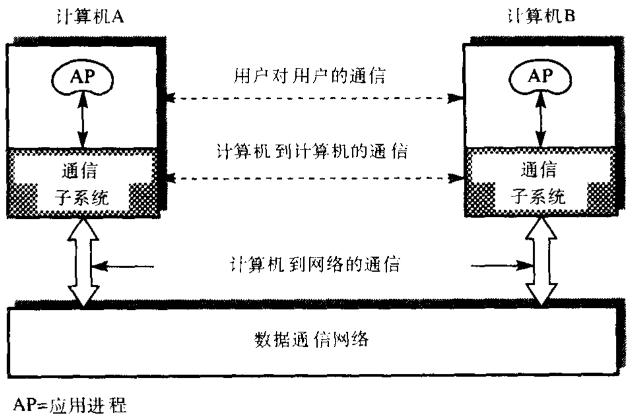


图1-1 计算机通信原理

当数据在计算机外部传送时，发生错码（误码）的概率就会增加。所以，在许多应用中，不仅要有检测（传输中的）差错何时发生的方法，而且要有获取受损数据的正确拷贝的方法。这称为**差错控制**，在两台计算机之间传递数据时要考虑到这种情况。本章还讨论其他问题，包括调节数据传送速率的**流量控制**问题，以及在有中间数据网络的情况下，建立网络通信路径的问题，等等。

在有些情况下，应用软件能直接使用这种基本的计算机到计算机的通信设施，而在其他情况下，要对应用软件增加附加的功能。例如，在有些应用中，进行通信的计算机可能是不同类型的，这意味着其内部的字符和数值的表示方式也可能是不同的，因此，必须要融入一些工具，保证每一台计算机以同样的方式对传送的数据进行解释。方法要与应用软件相结合；而且，计算机可能使用不同的操作系统，例如，一台计算机可能使用小型的单用户操作系统，而另一台计算机可能使用大型的多用户操作系统。这就说明，用户（应用）程序之间的接口也可能是不同的——和基本的计算机到计算机的通信服务也是不同的——通常被称为**应用进程**（AP）。在计算机间进行通信时，必须考虑这些方面。

1.1 数据通信网络

如前所述，采用的数据通信设备的类型是由应用的性质，所涉及的计算机数目和物理上距离确定的。典型的数据通信设备如图1-2到图1-6所示。

如果两台计算机在同一房间或办公室里，则传输设备就只是简单的点对点有线连接，如图1-2(a)所示。然而，如果两台计算机分散在一个城市或者一个国家中的不同地区，就要使用**公用载波**设备。通常包括要用调制解调器传输数据的**公用交换电话网**（PSTN）。总体框架如图1-2(b)所示。 5

在应用中如果有两台以上的计算机，交换通信设施（网络）使得计算机在任何时刻可以互相通信。如果所有的计算机分散在一个办公室里或一幢建筑物里，那么可以安装专门的网络，这样的网络称为**局域**（**数据**）**网**（LAN）。可以使用大量的这种LAN和连接设备。图1-3给出了两个基于LAN的系统。

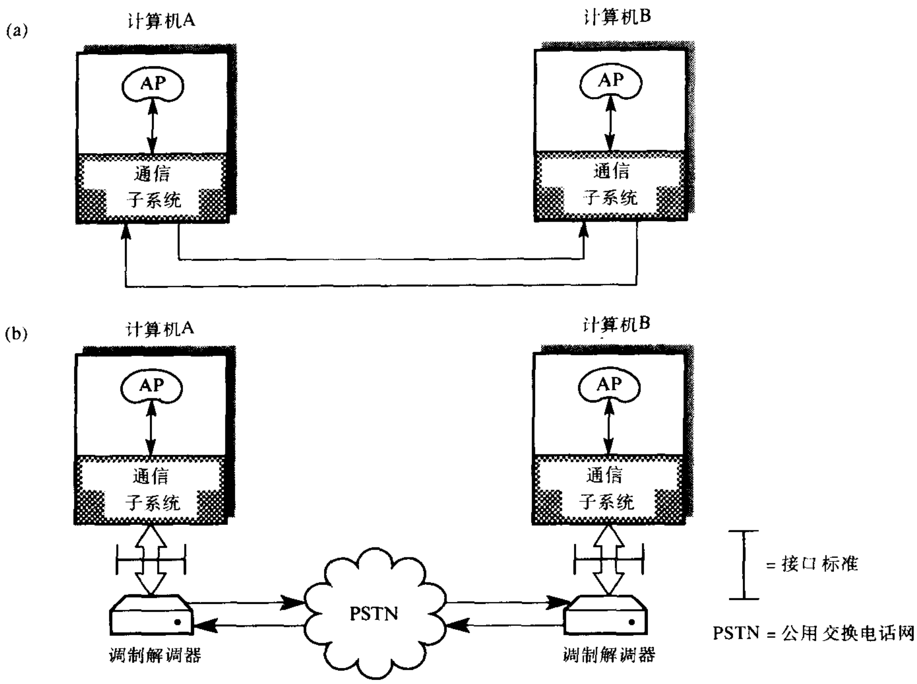


图1-2　简单的计算机到计算机连接的两种选择
(a) 点对点有线连接　(b) PSTN+调制解调器连接

当计算机分散在不同的机构（站点）时，就要用到公用载波设备。这样的网络称为**广域网**（WAN）。用何种类型的WAN取决于实际的应用性质。例如，如果所有的计算机属于同一个企业，并且需要在站点之间传递大量数据，一个解决方案是建立**企业专用网**：从公用载波上租用传输线路（电路），并在每个站点上安装一个专用的交换系统。许多大企业采用这一方案，这样的网络通常结合了语音通信与数据通信。一般的模式如图1-4所示。

6 这样的解决方案仅仅对大型企业有效，因为有大量的站间数据要传输，而租用线路以及安装、运行专用网络的花费对于大企业是可以承受的。在其他情况下，就要采用公用载波网络。除了提供公用交换电话服务，许多的公用载波现在还提供公用交换数据服务。这类网络，如PSTN，在全球范围内互连在一起，专门用于传输数据，而不是用来传输语音的。因此通常用**公共交换数据网**（PSDN）实现分散在一个国家中或全球范围内的计算机的分布式应用。许多公用载波现在转换为PSTN，这样数据传输不需要调制解调器；这类全部以数字方式操作的网络称为**综合业务数字网**（ISDN），在需要大范围的应用时，也可以考虑用ISDN来实现。一般的模式如图1-5所示。

在所有的这些应用中我们假定所有的计算机与相同的LAN或WAN相连。然而，在有些应用中，数据通信设备包含了多种网络，如LAN-WAN-LAN。例如，一个机构内与LAN相连接的一台工作站（计算机）可能需要同另一个机构内与LAN相连接的另一台计算机进行通信，这两个LAN通过PSDN互连。这类通信设施被称为**互联网**或**互连网络**，需要解决额外的与网络
7~8 本身相关的问题，以及计算机何时与这类网络连接的问题。这样的网络实例如图1-6所示。

迄今为止，我们讨论的网络还主要是实现只支持数据服务的工作站之间的数据传输。最近，工作站支持的服务已经进一步发展了，不仅包括数据传输，而且还包括其他类型的信息服务。这些工作站支持桌面视频电话、视频会议和一般的多媒体服务。为了支持更加丰富的

服务。现在已经开发了被称为**宽带多业务网**的新一代网络，因为其具有高传输速率，所以使用术语“宽带”。

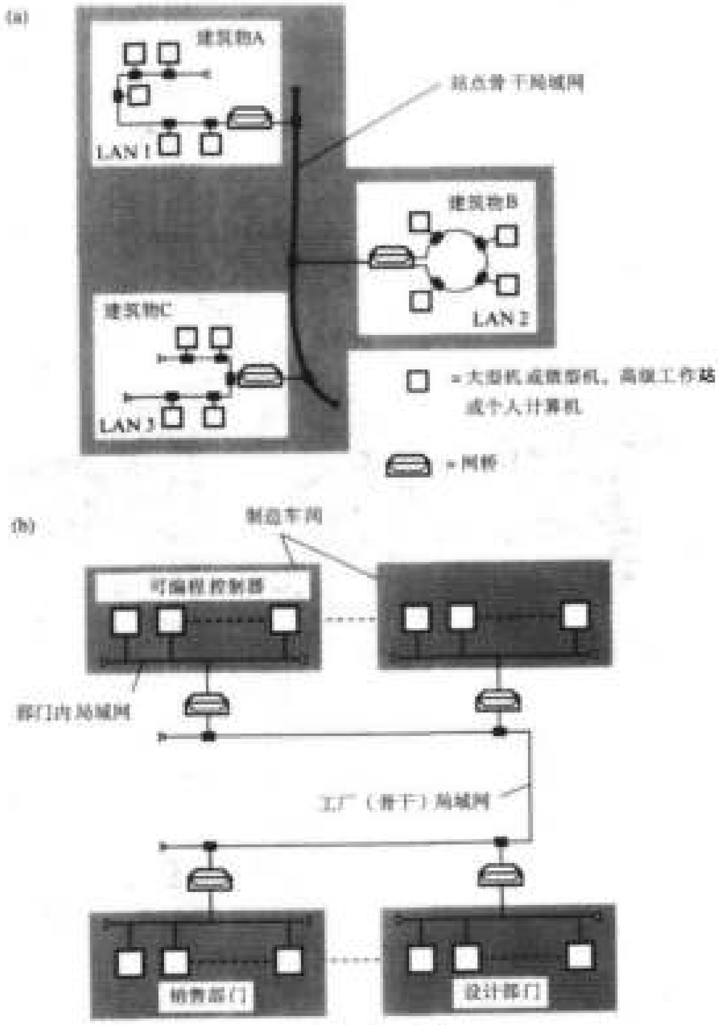


图1-3　基于LAN的分布式系统

(a) 科技和办公自动化　(b) 工业生产自动化

由于数据通信和音频、视频通信不同，所以采用新方法进行网络传输和网络交换，这一新方法被称为**异步传输模式**（ATM）。基于此操作模式的ATM LAN现在已经投入应用。已开发的新一代的ATM WAN与ATM LAN互连。此外，被称为**城域网**（MAN）的新的网络类型已经应用在ATM LAN的互连中，其中数据工作站分散在城市或城镇中的不同地区，多业务网络的模式图如图1-7所示。 9 10

1.2　标准

一直到最近，计算机产业界应用的、由各种国际组织给出的标准主要涉及计算机内部的操作或计算机与本地的外围设备的连接。这一情况造成不同制造商提供的早期通信子系统的

硬件和软件只能与他们自己的计算机以及所谓的插接兼容系统交换信息。这样的系统被称为封闭系统，因为不同制造商提供的计算机如果不符合某个特定的制造商的（专有）标准，它们就不可能交换信息。

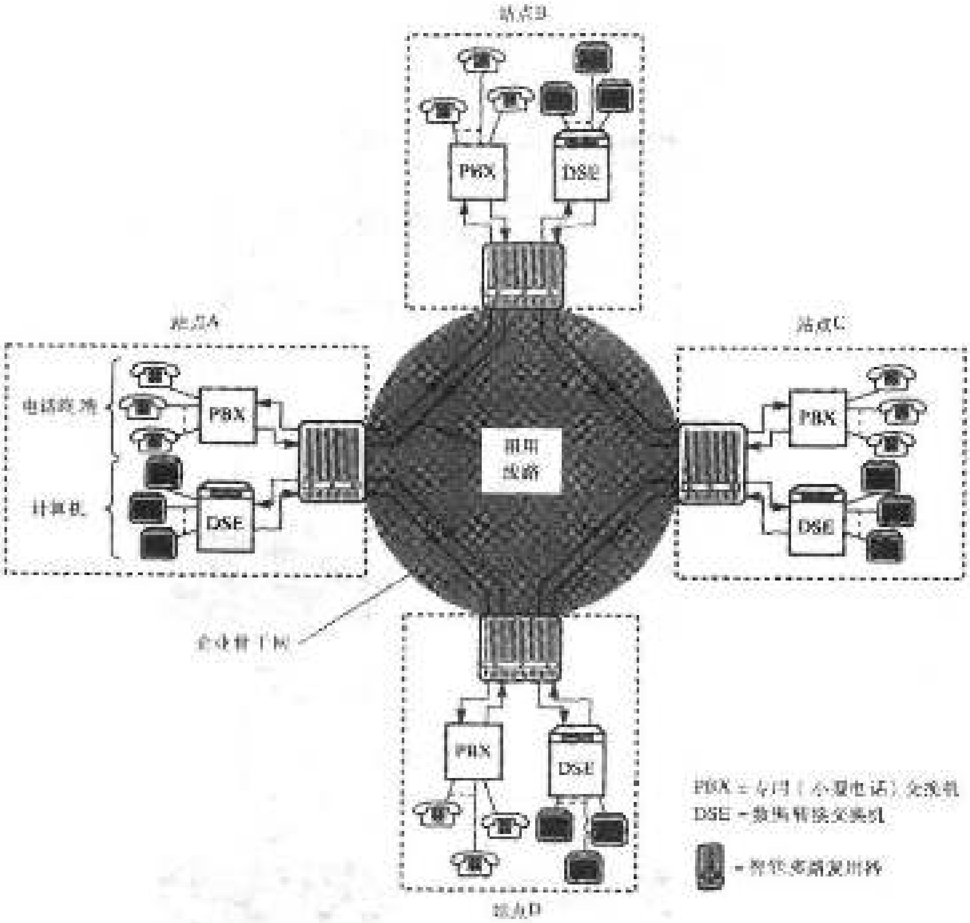


图1-4 典型的企业专用网

与之相反，许多年来，关心公用数据网络的各种国际组织对于网络上的设备连接提出了国际认可的标准。例如V系列建议是关于设备的连接标准，通常将这类设备称为数据终端设备（DTE），例如连接到PSTN的调制解调器；X系列建议是关于DTE连接到公共数据网的标准；I系列建议是关于将DTE与正在形成中的ISDN连接的标准。这些标准使得来自不同厂商的设备之间能够兼容，也使得购买者可以从许多制造商中选择适合的设备。

起初，大多数公用载波提供的服务主要涉及数据传输，所以相关联的标准仅仅与设备和网络的连接接口方法。近来，公用载波开始提供更广泛的分布式信息服务，例如电子消息交换（Teletex）和访问公共数据库（Videotex）。为了实现这样的服务，电信产业的标准化组织不仅提出与网络接口的标准，而且制订系统间进行信息（数据）交换的格式（语法）和控制的高级标准。按照这些标准制造的设备可以与按照同样标准制造的其他制造商的设备互换使用。这样的系统称为开放系统，完整的称呼是开放系统互连环境（OSIE）。标准演化的汇总和主要的标准化组织在图1-8中给出。

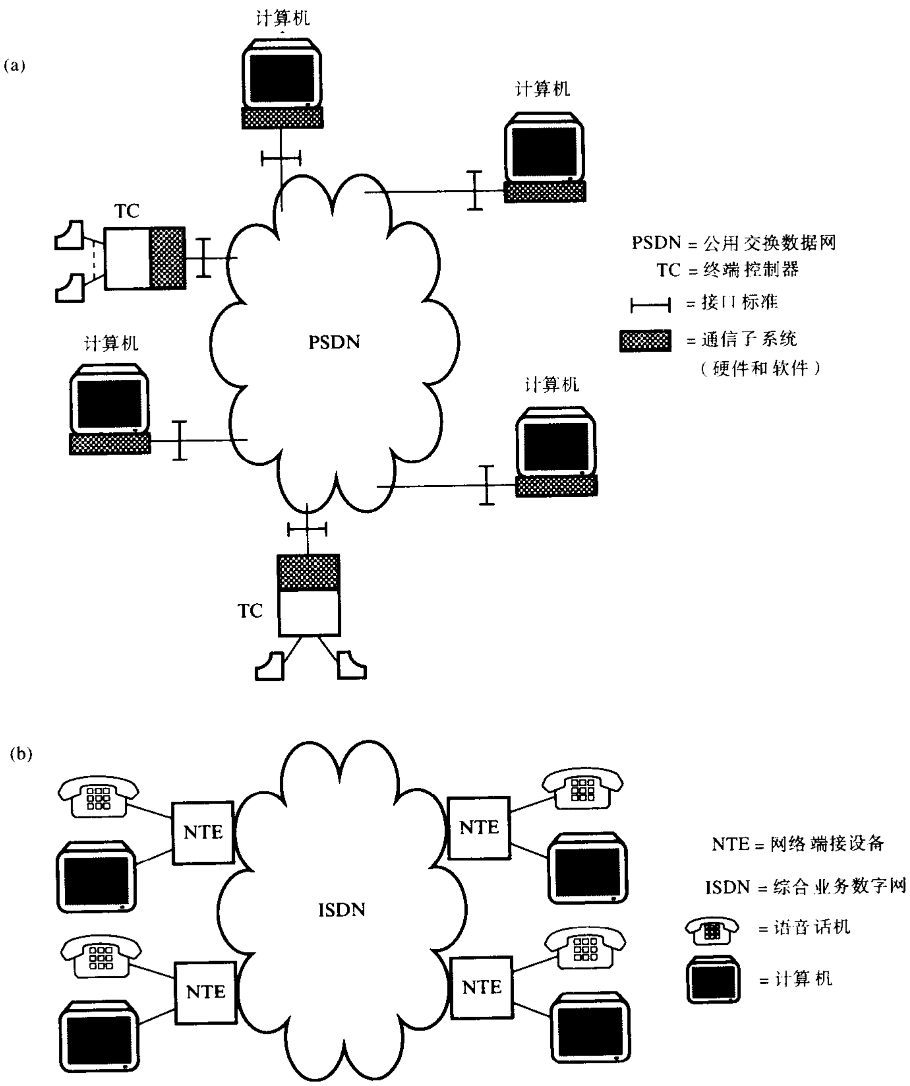


图1-5 公用载波数据网

(a) PSDN (b) ISDN

在20世纪70年代的中期，由于基于公用和专用数据网的各种分布式系统开始激增，开放系统潜在的优点也开始被计算机产业界认可。因此，开始引入了一系列的标准。首先给出的标准是关于计算机内的完整的通信子系统的总体结构。这一标准由**国际标准化组织**（ISO）制定，称为**开放系统互连**（OSI）的**ISO参考模型**。

ISO参考模型的目的是提供一个用于协调标准开发的框架，并把现存的和将要发展的标准 11
化活动置于该框架之内。这一目的允许在计算机中支持特定标准集的应用进程能顺利地与其他计算机中支持相同标准的应用进程进行通信，而不管它们的制造商是谁。

以开放的方式进行通信的应用进程实例如下：

- 在一台计算机上执行的进程（程序）要访问远程文件系统；
- 一个进程对一组分布式（客户）进程提供中央文件服务（作服务器）；

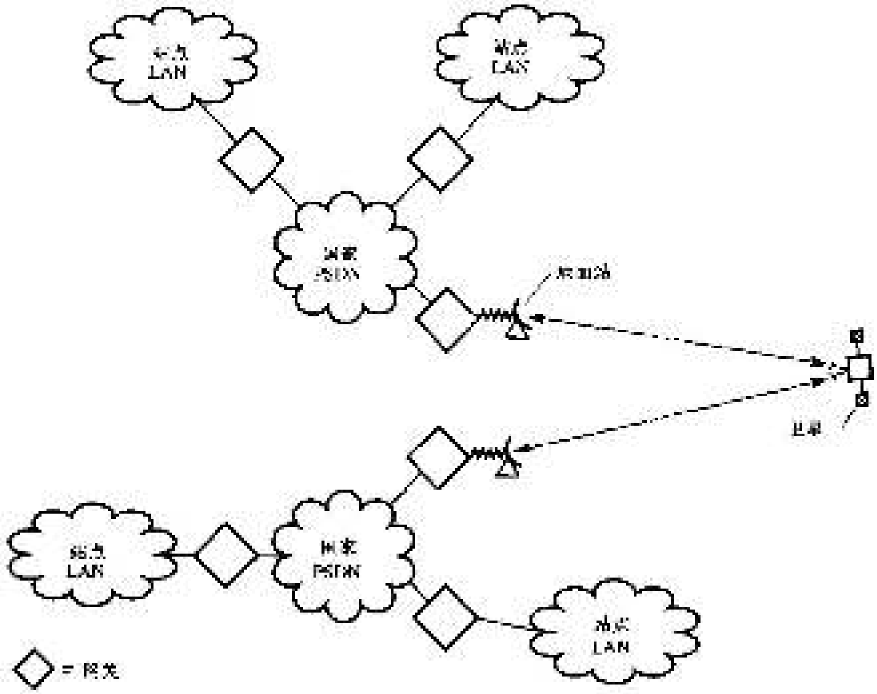


图1-6 全球互联网

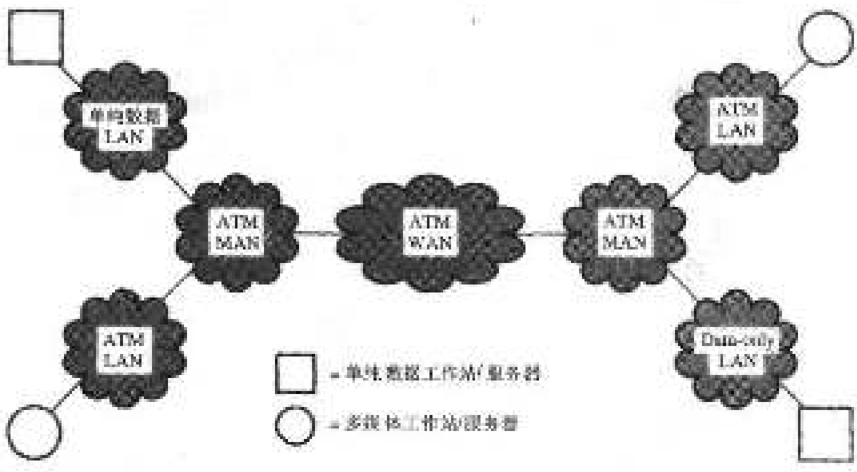


图1-7 宽带多业务网

- 在一台办公室工作站（计算机）上的进程要访问电子邮件（e-mail）服务器；
- 一个进程充当一组分布式（客户）进程的电子邮件服务器；
- 在加工工厂或自动化制造工厂中，一个监控计算机中的进程，控制一组分布式的基于计算机的仪器或机器人控制器；
- 在仪器或机器人控制器中的进程从监控系统中接收指令并返回操作结果；
- 在银行计算机中的进程启动远程系统中的借贷操作。

OSI涉及进程间的信息交换。OSI的目的是使得应用进程在执行一个特定的（分布式的）

信息处理任务中进行协作，而不管它们是在什么计算机上运行的。 12

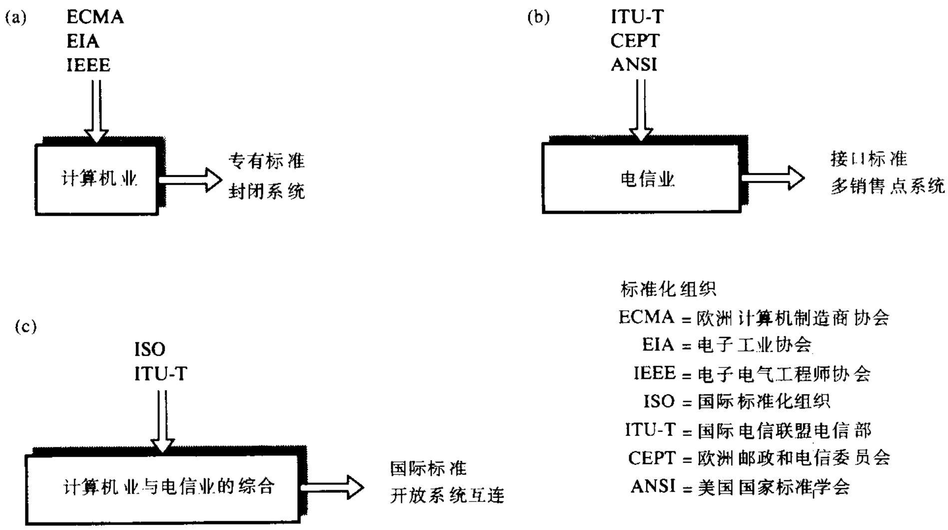


图1-8 标准的演化和主要的标准化组织

1.3 ISO参考模型

通信子系统是一个软件和硬件构成的复杂结构。在早期的实现中，这些子系统的软件通常基于有许多交互组件的功能单一的、复杂的、非结构化的程序（通常用汇编语言实现）。这样实现的软件测试困难，并且修改也非常困难。

为解决这一问题，ISO对参考模型采用了分层的方法，将完整的通信子系统分成许多层，每一层完成特定的功能。从概念上讲，这些层次执行两类功能之一：基于网络的功能和面向应用的功能。这就产生了三种不同的操作环境：

1）**网络环境**，包括与不同类型的底层数据通信网络相关的协议和标准。

2）**OSI环境**，包括网络环境，并增加了允许终端系统（计算机）以开放的方式与其他计算机进行通信的额外的面向应用的协议和标准。

3）**实际系统环境**，建立在OSI环境的基础上，包括制造商专有的软件和服务，这些软件和服务执行某个特定的分布式信息处理任务。

这些环境如图1-9所示。 13

在OSI模型中，无论依赖网络的组件，还是面向应用的（不依赖于网络的）组件，都是以多层的方式实现的。每一层之间的界限及其各层完成的功能是在早期的标准化活动所获取的经验基础上确定的。

在整个通信子系统中，每一层执行确定（已经定义好）的功能。每一层根据定义的**协议**——规则集合——进行操作，同远程系统相应的**对等层**（相似层）交换信息（包括用户数据和附加的控制信息）。每层与其相邻上下层有严格定义的接口。每个协议层的实现独立于所有其他的层。

ISO参考模型的逻辑结构由七个协议层组成，如图1-10所示。最下面的3层（第1层～第3层）基于网络，是关于数据通信网络的协议，数据通信网络用于连接两个要求通信的计算机。

最上面的3层（第5层～第7层）面向应用，是关于允许两个终端用户应用进程彼此交互的协议，通常通过本地操作系统提供的服务来实现。中间的传输层（第4层）使得在它上面的面向应用的层能将具体操作交给在它下面的依赖网络的层来实现。从本质上说，传输层构建在后者提供的服务上，提供面向应用层独立于网络的信息交换服务。

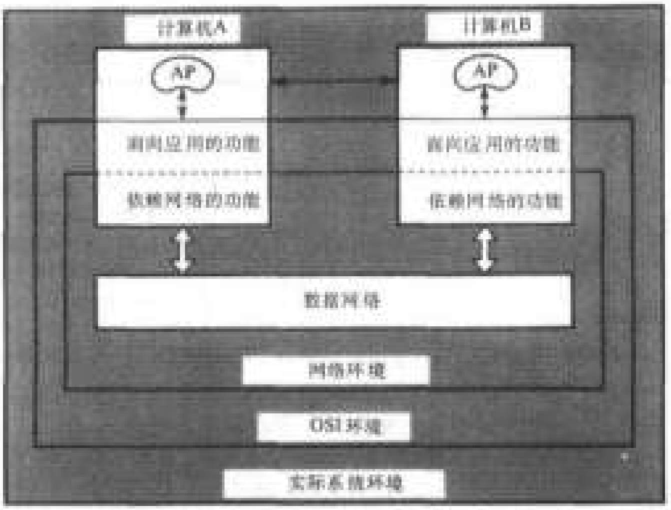


图1-9　操作环境

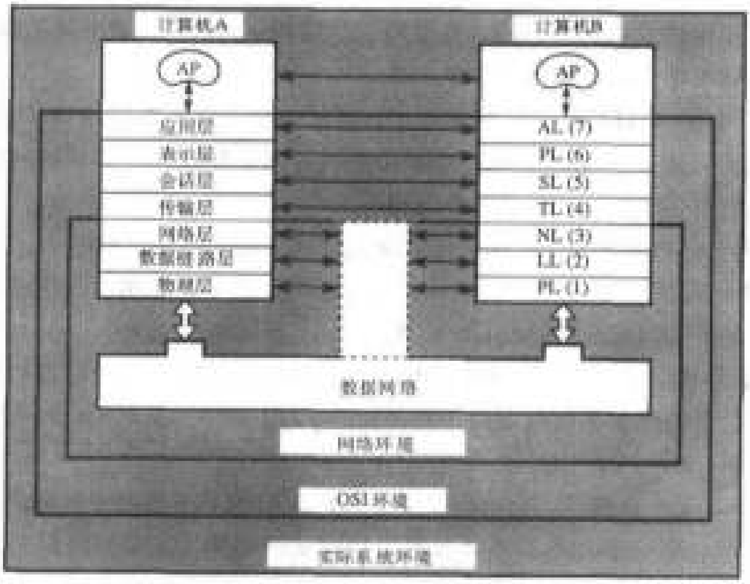


图1-10　ISO参考模型的总体结构

14　每一层的功能被形式化地用一个协议来说明，该协议定义了该层与另一远程系统中相应对等层进行通信时采用的规则和约定的集合。每层向它上面的一层提供给定的服务集合，也获得它下面一层提供的服务，将与协议相关的消息单元传送到远程的对等层中。例如，传输层给它上面一层——会话层提供独立于网络的消息传输服务，并通过它下面一层——网络层提供的服务，将与传输协议相关联的消息单元的集合传递到另一个系统的对等传输层。按照

给定的协议，每个层与远程系统中的对等层进行通信。然而，实际上，每一层的协议消息单元利用下一层提供的服务传递。每层的基本功能如图1-11所示。 15

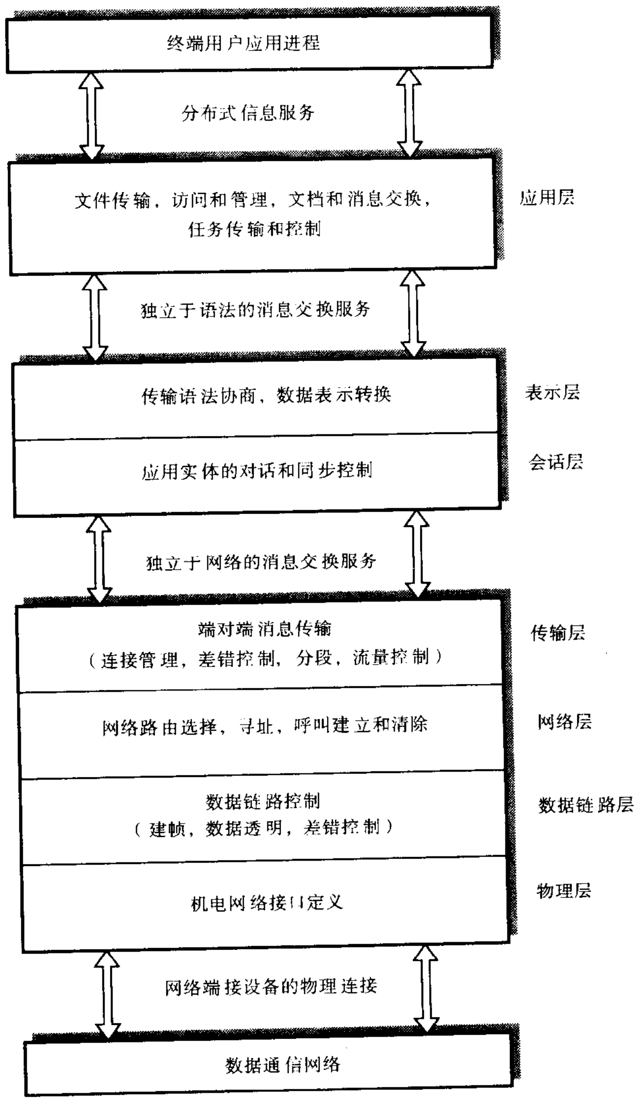


图1-11　协议层一览

1.3.1　面向应用的层

1. 应用层

应用层对各种各样网络的分布式信息服务提供用户接口——这类接口通常是一个应用程序/进程。应用层包括了文件传输访问和管理，以及一般的文档和消息交换服务，如电子邮件。许多标准的协议对于这些服务和其他的服务类型是适用的。

访问应用服务通常通过一组定义的原语来实现，每个原语有相关的参数，原语由本地操作系统支持。访问原语与其他操作系统调用是一样的（比如访问一个本地的文件系统），并由此激活一个操作系统过程（进程）。这些操作系统过程就像本地的设备（比如类似于磁盘控制器）

使用通信子系统（软件和硬件）。通信子系统的具体操作和实现对于（用户）应用进程是透明的。当应用进程使得调用重新调度（执行）时，要返回一个或多个状态参数以指示已执行的网络事务处理是否成功。

除了信息传输外，应用层提供如下服务：

- 通过名称或地址来标识要连接的通信对象。
- 确定一个要连接的通信对象当前是否可用。
- 通信权限的建立。
- 协定保密（加密）机制。
- 一个要连接的通信对象的认证。
- 对话方式的选择，包括启动和释放过程。
- 协定错误恢复责任。
- 标识数据语法（字符集、数据结构等）的约束。

2. 表示层

表示层是关于在两个通信应用进程之间传输期间数据的表示方式（语法）。为了实现开放系统的成功互连，许多普通的**抽象数据语法**形式被定义为应用进程的使用以及相关的**传输（或**
16 **具体）语法**。表示层协商和选择合适的在事务中使用的传输语法，使得在两个应用实体之间按规定的语法（结构）交换消息。如果表示的形式不同于内在的抽象形式，表示协议要做必要的转换。

现在通过一个说法语的人和一个说西班牙语的人通电话时进行的电话转换的实例，来说明表示层提供的服务。假设每人有一个翻译，并且两个翻译仅懂英语。每个翻译要将他们的母语翻译为英语，将英语翻译为他们的母语。这两个通信者同应用进程是相似的，两个翻译代表表示层的实体。法语和西班牙语是本地的语法，而英语是传输或具体的语法。可以注意到，一定有种被大家理解的语言定义为用于协商的传输语言（语法），也可以注意到，作为翻译，不必理解转换的含义（语义）。

表示层的另一项功能是关于数据的安全性。在有些应用中，由应用发送的数据首先使用**密钥**加密，要求仅被预期的接收方的表示层理解；而后者通过使用先前传递过来的相应的密钥解密收到的数据。尽管这不是标准的一部分，有关加密的主题将在讨论表示层内容的12.4节进行阐述。

3. 会话层

会话层允许两个应用层协议实体组织和同步它们的对话，并管理它们的数据交换。在完整的网络事务处理过程中，在两个通信应用层协议实体（实际的表示层协议实体）之间，会话层用于建立（清除）通信（对话）信道。会话层提供了许多可选的服务，包括如下服务：

- 交互管理　与对话相关的数据交换可以是双工的——双向同时——或半双工的——双向可选。在后一情况下会话层协议提供了以同步方式控制数据（对话单元）交换的能力。
- 同步　对于较长的网络事务，用户（通过会话层提供的服务）可以选择周期性地建立与传输相关联的同步点。如果在事务期间对话可以在一个约定的（早期的）同步点重

启动。

• 异常报告　在一个事务无法恢复的异常情况下，由会话层向应用层发信号。 17

4. 传输层

传输层作为面向应用的高层和依赖网络协议的低层之间的接口，它向会话层提供了独立于底层网络类型的消息传递功能。通过提供给会话层一组定义的消息传输功能，传输层隐藏来自会话层底层网络的操作细节。

传输层提供了许多的**服务类**，以弥补由与不同类型网络相关的网络层提供的变化的**服务质量**（QOS）。有5类服务，从第0类到第4类；第0类仅仅给出了建立连接和数据传输所需的基本功能；第4类提供了完整的差错控制和流量控制过程。

作为一个实例，第0类可以被选作与PSDN一起使用，而第4类可以与PSTN一起使用。我们将在11.6节讨论传输层细节时深入讨论这一部分。

1.3.2 依赖网络的层

由于ISO参考模型的最下面的三层是依赖网络的，所以它们的详细操作随网络类型而变化。然而，一般**网络层**负责建立和清除在两个传输层协议实体之间的网络连接。它的功能包括网络路由选择（寻址）和在某些情况下的计算机到网络接口的流量控制。在网络互连的情况下，它在互连的网络之间提供了不同的协调功能。

链路层建立由特定网络提供的物理连接，以提供给网络层可靠的信息传输能力，还负责在传输错误的情况下进行错误检测，并重新传递消息。通常情况下，链路层提供两类服务：

1）**无连接**，将每个信息帧作为一个自包含的实体，采用最佳试验法进行传输，即是说，如果帧中的错误被检测出，则帧被简单地丢弃。

2）**面向连接**，努力提供无错误的信息传输功能。

最后，**物理层**是关于用户设备和网络终端设备间的物理和电气接口，它提供链路层在两个设备之间传递一连串比特流的能力。 18

1.4 开放系统标准

ISO参考模型已经被作为一个正式的通信子系统结构的模板提出，与每一层相关联的标准活动基于通信子系统。每一层并不只有一个标准协议，而是存在一个标准协议的集合，每个协议实现了不同级别的功能。对于一个实际的OSI环境，例如在一个完全自动化的制造工厂中，许多基于计算机的系统要连接，就要定义一个精选的标准集合，在这一环境中所有系统都采用这一标准集合。

三个主要的国际组织制定计算机通信的标准，它们是ISO、美国的国际电子电气工程师协会（IEEE）和国际电信联盟电信部（ITU-T），后者即是以前的国际电报电话咨询委员会（CCITT）。ISO和IEEE制定计算机制造商使用的标准，而ITU-T定义不同类型的国家和国际公共网络的连接设备标准。随着计算机和电信产业的重叠部分的增加，这些标准组织产生的标准之间的合作和协同也增加了。

此外，在ISO开展标准化活动之前，美国国防部通过**美国国防部高级研究计划局**（DARPA）开展计算机通信和网络方面的基础研究。作为这一研究的组成部分，与许多大学和研究机构相

关的计算机网络与DARPA相连接，产生的互连网络称为ARPANET；现在ARPANET已经扩展，与其他政府部门开发的互联网合并。这样结合的互联网现在被简称为**因特网**。

用于因特网的协议集称为**传输控制协议/网际互连协议**（TCP/IP），包括了面向网络的协议和支持应用的协议。因为TCP/IP目前在互联网上广泛地使用，所以TCP/IP协议的许多部分已经作为ISO标准的基础。而且，因为与TCP/IP协议相关联的所有协议规范是属于公用域的——所以不必支付使用许可费——因此它们在商业和官方领域被广泛地使用，以创建开放系统网络环境。所以，实际上有两个主要的开放系统（独立于厂商的）标准：TCP/IP协议族和以ISO标准为基础演化的协议。

图1-12给出与TCP/IP协议族相关联的标准。正如我们所看到的，因为TCP/IP是与ISO同时发展起来的，所以它不包含与所有ISO层相关联的协议；而且，TCP/IP协议族使用的说明方法
19 不同于ISO标准使用的说明方法。但是与ISO层相关的大部分功能可以嵌入到TCP/IP协议族中。

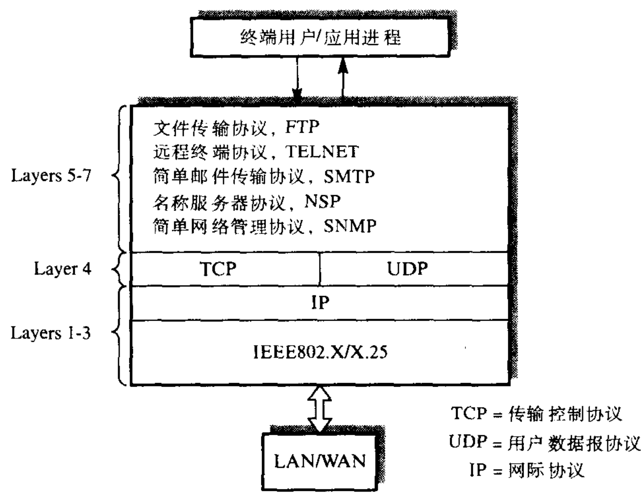


图1-12　TCP/IP协议族

如图1-13所示，ISO/ITU-T标准的范围涉及每一层。它们集中在一起，允许建立开放系统环境的管理机构从中为应用选择最适合的标准集合。产生的协议集称为**开放系统互连概要**。许
20 多这样的概要现在已经给出，包括以下内容：TOP，在技术和办公环境应用的协议集合；MAP，在制造自动化方面应用的协议集合；US和UK GOSIP分别在美国和英联邦国家的政府项目中使用；相似地，在欧洲使用的协议集称为**欧洲标准化委员会**（CEN）**功能标准**。最后一个由**标准倡议与应用组织**（SPAG）（定义由12家欧洲公司组成）。

如图1-13所示，最低的三层随网络类型的不同而变化。ITU-T对于公共载体网络的使用已经定义了V系列、X系列和I系列标准。V系列由现存的PSTN使用，X系列由现存的PSDN使用，I系列由刚出现的ISDN使用。X系列和I系列将在第8章中讨论。ISO/IEEE产生的用于LAN的标准在第6、7和8章中讨论。

虽然ISO和ITU-T被不同系统采用，传输、会话和表示层的功能和说明几乎是相同的。在应用层标准的范围内，对于私有网络由ISO定义，对于公用载波服务由ITU-T定义。面向应用

的（独立于网络的）协议层的功能和操作在第11、12、13和14章中阐述。 21

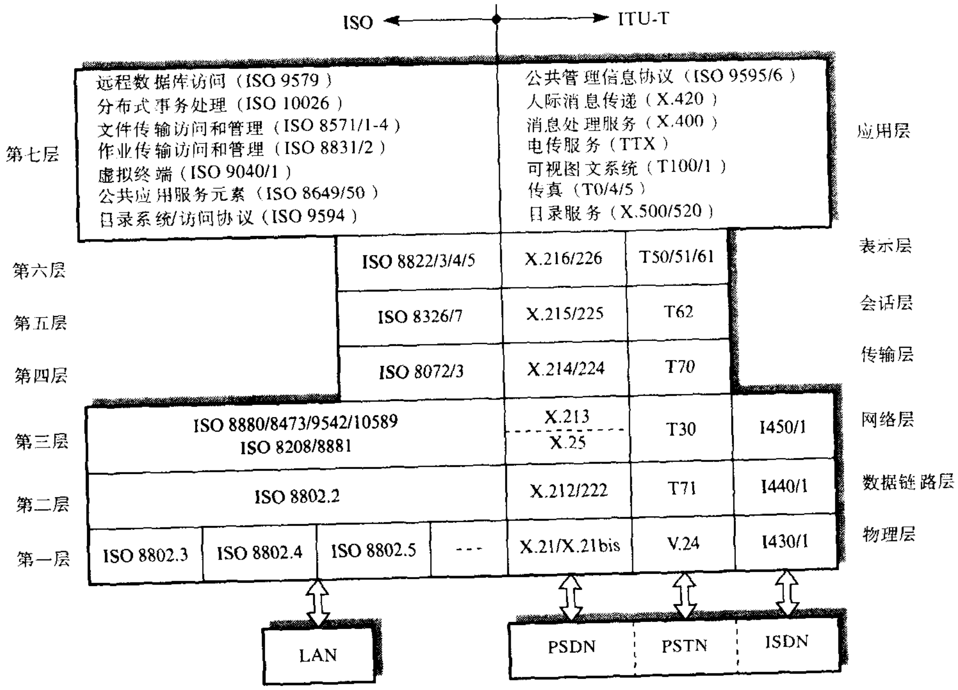
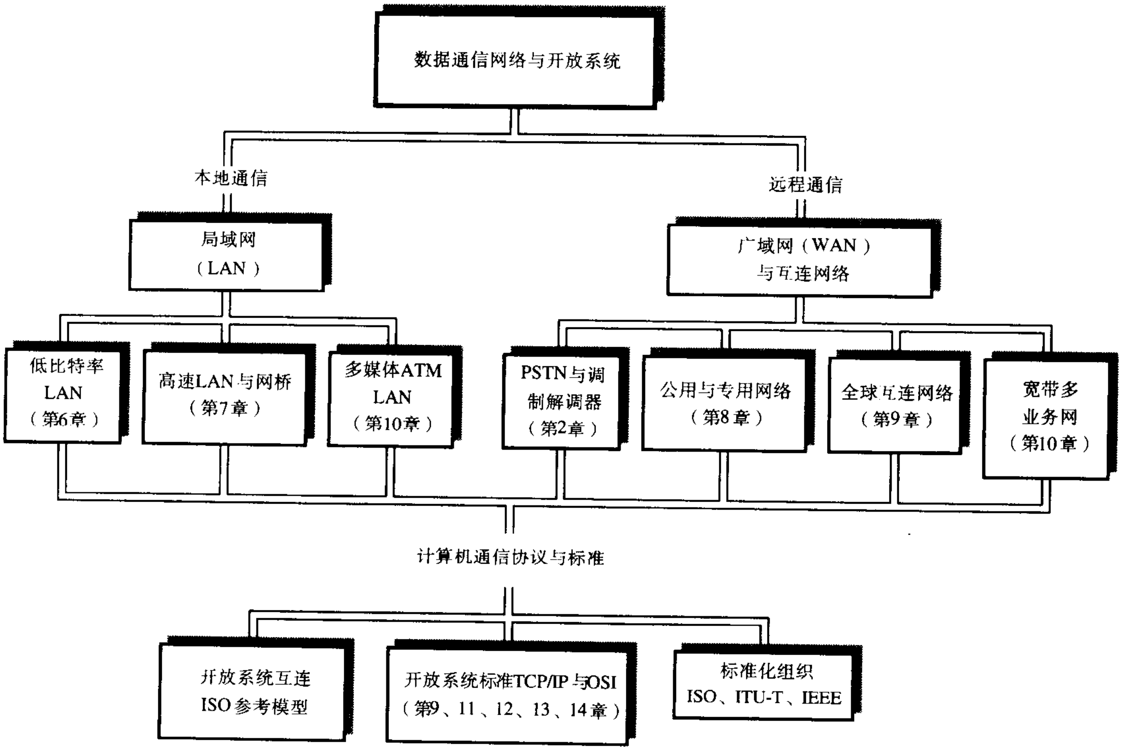


图1-13　标准一览

本章概要



数据通信网络与开放系统
本地通信
远程通信
局域网
（LAN）
广域网（WAN）
与互连网络
低比特率
LAN
（第6章）
高速LAN与网桥
（第7章）
多媒体ATM
LAN
（第10章）
PSTN与调
制解调器
（第2章）
公用与专用网络
（第8章）
全球互连网络
（第9章）
宽带多
业务网
（第10章）
计算机通信协议与标准
开放系统互连
ISO参考模型
开放系统标准TCP/IP与OSI
（第9、11、12、13、14章）
标准化组织
ISO、ITU-T、IEEE

第2章 电 气 接 口

本章目的

读完本章，应该能够：

- 描述用于传输数据的不同类型的物理传输介质；
- 理解各种介质的特性与适用范围以及基本理论与基本定律；
- 说明传输介质所用电信号的不同形式；
- 理解在公用电话交换网（PSTN）上传输数据时要用的调制解调器；
- 描述调制解调器中调制器部分与解调器部分的设计方案；
- 理解数字租用线路的由来与结构；
- 识别计算机与数据电路终端设备连接的标准，包括传输介质的种类，所用电信号的形式以及用于调节通过接口的数据流的附加控制线的作用；
- 理解公用标准规定的若干附加控制线的功能。

引言

在传输线上传送二进制数据，每个被传送的由二进制数字组成的元素必须转换成电信号。例如，二进制的1可用$+V$伏的电压（或电平）信号发送到传输线的终端，而二进制的0可用$-V$伏的电压发送。在接收时，接收设备须将$+V$伏的信号转换成二进制1，$-V$伏的信号转换成0。 23
实际上，由于传输介质不可能是理想的，传输的电信号存在**衰减**（减弱）与**失真**（变形）。在极端情况下，接收方不能正确区分二进制0信号与1信号，如图2-1所示。信号衰减与失真的程度受下列因素影响：

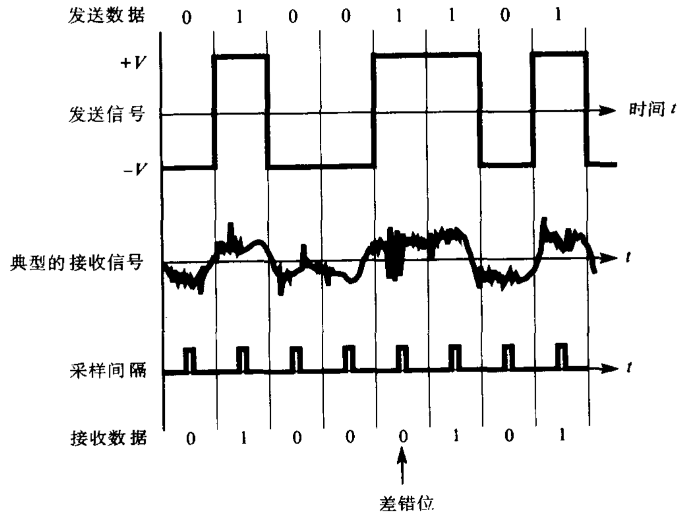


图2-1 传输介质干扰的影响

- 传输介质类型；

• 数据传输的比特率；
• 两个通信设备之间的距离。

在2.2节，将讨论传输质量降低的根源。

对于各种类型的传输介质与物理间隔，衰减与失真可定量地描述，国际标准定义了两类数据通信设备之间的电气接口。这些标准不仅规定使用的电信号电平，而且还规定附加控制信号的用法与意义，以及在物理接口中使用的约定。制定互连数据通信设备标准的两个组织是**欧洲国际电信联盟电信部**（ITU－T），其前身是**国际电报电话咨询委员会**（CCITT），以及**美国电气工业协会**（EIA）。虽然这两个组织制定的标准在术语上略有不同，但基本信号及其含义是相同的。

本章分六节，前两节描述最常用传输介质，接下来两节说明电信号的各种形式，2.5节叙述公用载波电路的特征，2.6节叙述常用物理层接口标准的附加内容。在大多数实例中，虽然考虑的是一台计算机接口到不同数据通信接口的连接，但还是用通用术语数据终端设备
24 （DTE）而不用计算机，因为它隐含着任何类型设备。

2.1　传输介质

电信号传输需要传输介质，一般的传输介质都采用**传输线**。通常，由一对导线或电线组成，但也可以采用由玻璃纤维引导的光束或自由空间传播。因为传输介质类型决定每秒可传输的最大**比特数**（二进制数）或**bps**来表示，所以传输介质类型很重要。在下面各小节介绍一些常用的传输介质类型。

2.1.1　双线开放线

双线开放线是最简单的传输介质。两根线彼此绝缘，并且对空间开放。这一类线适合于连接以中等比特率传送信息（小于19.2 kbps）、相距不超过50米的设备。采用典型的电压或电流信号，一条线加信号电平，另一条线为地线。

虽然双线开放线可直接连接两台计算机（DTE），但主要用于连接DTE到本地数据电路端接设备（DCE），例如调制解调器。并且通常采用多线连接。通常每一种信号有一根单独的绝缘线，而地线共用。然后，把这组线封装在一根受保护的**多芯电缆**中或者塑制成**扁平电缆**，如图2-2(a)所示。

使用这类线必须注意避免同一根电缆中相邻线之间的电信号的交叉耦合。这种耦合由两线间**电容耦合**引起，称为**串扰**。此外，开放的结构易受其他电信号源**电磁辐射**的**噪声信号**的影响。这种信号的一个主要问题是有可能仅在一根线上——例如信号线——拾取信号导致在两根线间产生了额外的不同信号。由于接收方通常在两根线间使用不同的信号操作，导致对合并的（信号加噪声）接收信号的错误解释。这些因素限制这类线在一定的长度和传输速率
25 之内才能可靠使用。

2.1.2　双绞线

通过使用**双绞线**（两根交叉绞合在一起的导线）可以达到较强的抗噪声性能。信号线和地线的接近意味着通过两线拾取的干扰信号减弱了对差异信号的影响。同时，多个双绞线封闭于同一电缆中，电缆中每对线的绞合减少了串扰。双绞线如图2-2(b)所示。

采用线路驱动电路与接收电路并利用双绞线结构上的优点，在短距离内（小于100米）传输，比特率可达1 Mbps量级，甚至可在较长距离以较低比特率传输。此时，双绞线是适合的。

采用更复杂的驱动电路与接收电路，在更长距离上甚至可达更高的比特率。这类线称为**非屏蔽双绞线**（UTP）。它用于电话网以及许多数据通信中（带有专用集成电路）。在双绞线电缆中，采用屏蔽防护层来进一步减少干扰信号的影响可使用**屏蔽双绞线**（STP）（参见图2-2(c)）。

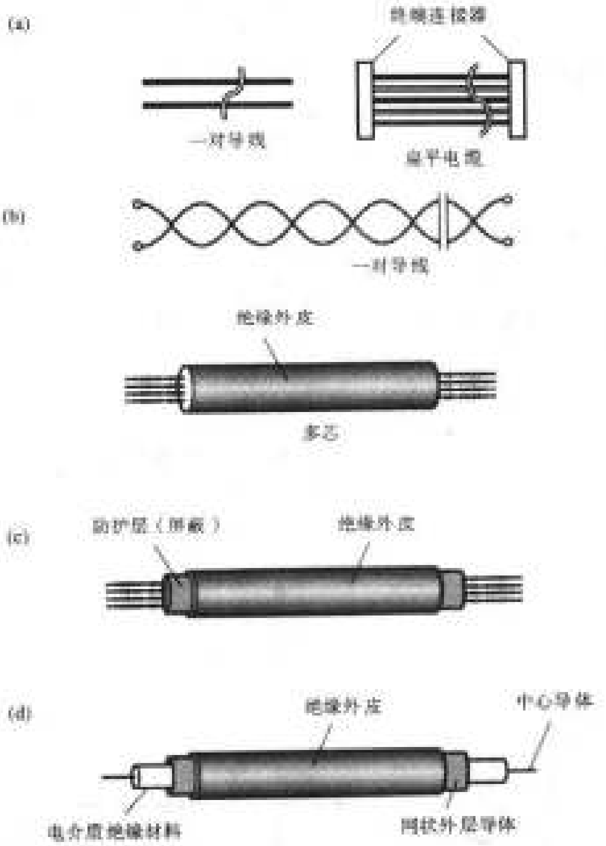


图2-2 铜线传输介质

(a)双线开放线 (b)非屏蔽双绞线 (c)屏蔽双绞线 (d)同轴电缆

2.1.3 同轴电缆

双绞线的主要限制因素是它的电容与称为**集肤效应**的现象。由于传输信号的比特率（频率）增加，传输线中的电流集中于导线外层表面，因此可利用的有效截面减少。导线对高频信号的电阻增大，引起较强的衰减。另外，在高频时，由于电磁辐射效应有较多信号能量损失，所以在比特率大于1 Mbps的应用时，必须用复杂的驱动电路与接收电路或另一类型的传输介质。 26

同轴电缆使这种效应的影响达到最小。图2-2(d)展示信号线与地线，由中心的一根固态导体，和外层的固体（网状）环形导体聚合（同轴）而成。理想的情况，两个导体的间隙应该填空气，但实际上，通常是用固体的或蜂窝状结构的绝缘材料填充。

中心导体通过外层导体有效地屏蔽外部干扰信号，由于外层导体存在电磁辐射效应与集肤效应会使损失达到最小。同轴电缆适用于各种不同类型的信号，尤其适用于在几百米距离内，速率为10 Mbps的信号传输（带调制的更高）。在2.3.4节中还将看到同轴电缆适用于点对点与多点拓扑结构。

2.1.4 光纤

尽管同轴电缆有效地减弱了许多不利影响，但它使用金属导线（一般是铜）传输信息的速率，最大信号频率虽然高，也是有限的。双绞线的情况也相同。光缆则完全不同于这两种传输介质，它依靠光束在玻璃纤维中反射传播来传送信息，而不是在导线上传送电信号。光波的带宽比电波大得多，因而光缆可以用于传输每秒几百兆位的信号。由于使用光束，避免外界电磁干扰和串扰的影响，因此光缆尤其适用于速率虽低但有电气干扰的环境，例如在高压交流设备集中的钢厂就常用光缆。由于使用光缆难于窃听，故在高度机密的环境中也越来越多地采用光缆。

光缆由玻璃纤维组成（每个传送信号在一根玻璃纤维中传输），包含在屏蔽外界光源干扰的防护层内。光缆的结构如图2-3(a)所示，光信号由一个光发送器发出，它将DTE中使用的电信号转换成光信号。同样在接收端，由一个光接收器执行相反的功能。通常，光发送器使用发光二极管（LED）或激光二极管（LD）完成转换的操作，而接收器使用光敏感的光电二极管或
27 光晶体管。

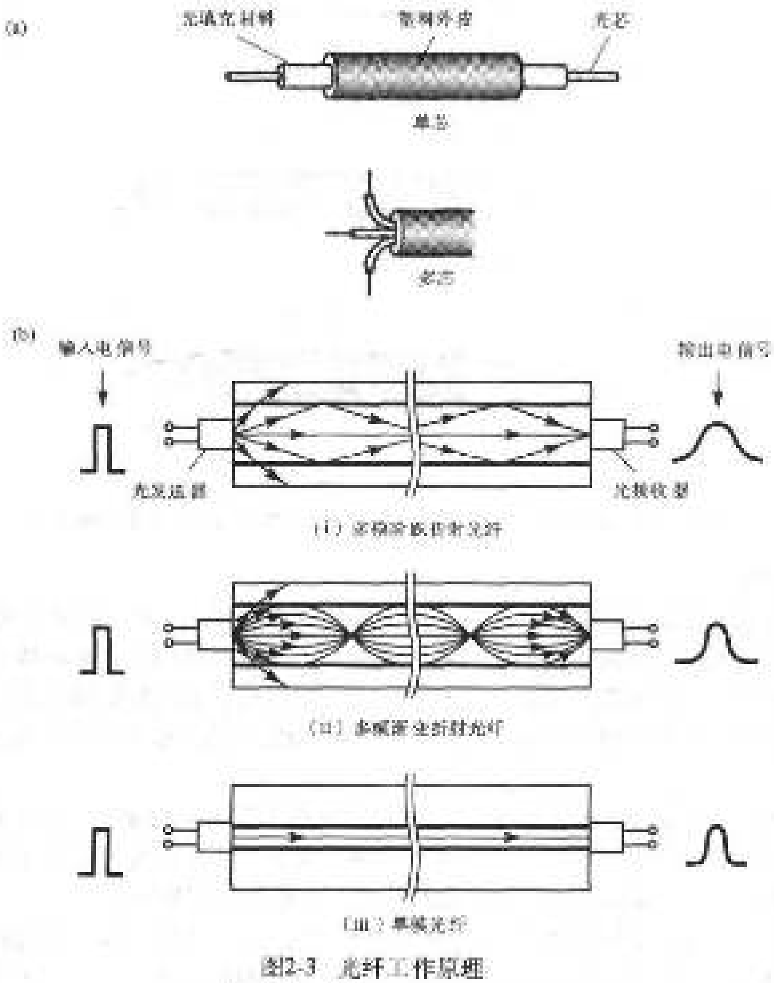


图2-3　光纤工作原理

(a) 光缆的结构　(b) 传输模式

光纤本身由两部分组成：玻璃纤芯与具有低折射率的玻璃填充材料。光沿着玻璃纤芯，依

赖于填充材料的类型与芯的宽度，用三种传播方法中的一种传播。传输模式如图2-3(b)所示。

在**多模阶跃折射光纤**中，填充材料与芯材是不同的，但它们的折射率是一致的。由二极管发射出的入射角小于临界角的所有光线在填充材料的界面被反射，并沿着芯材利用多次（内部）反射传播。由于二极管发射的光线依赖于入射角不同沿着光缆传播到达终点的时间不同，所以，接收信号比输入信号脉冲宽度更宽，因而相应地减少最大允许比特率。这种类型光缆采用价格相对便宜于激光二极管的LED，主要适用一般比率传输。

具有变化（非常数）折射率的芯材可降低散射。如图2-3(b)所示，在**多模渐变折射光纤**中， 28
当光远离开芯材中心向外移动时，被逐步折射。这样，接收信号脉冲宽度与阶跃光纤相比有变窄的影响，允许相应增加最大比特率。

进一步的改进可通过减少芯材的直径为单个波长（3～10μm）获得，使得所有发射光沿着单一（无散射）路径传播。这种**单模光纤**通常采用LD（速率每秒几百兆比特）。

2.1.5 卫星

到目前为止，讨论过的所有传输介质都是携带传输信息的物理线路，然而，数据也可以通过电磁（无线电）波在自由空间传播，如**卫星系统**。在卫星通信系统中，经数据调制的**平行微波束**从地面发送到卫星，使用称为**发射机应答器**的运载电路将收到的光束转发（延时）到预定目的地。一个卫星具有多个发射机应答器，每一个覆盖一个特殊频带范围，一个典型卫星信道具有极高带宽（500 MHz），并使用**多路复用**技术提供几百条高比特率的数据链路。这将在2.5.2节讨论。信道总的可用容量被划分成若干子信道，每一子信道可支持一条高比特率的数据链路。

以通信应用为目的的卫星，一般是对地**静止的**，就是卫星每24小时绕地球一周，与地球旋转同步。因此从地面上来看，卫星是固定的。卫星的轨道是这样挑选的，能够提供到达发送站与接收站的视线通信路径。卫星转发的并行微波束在广域范围拾取粗略的信号，仅在限定范围拾取精细的信号。在第二种情况，使用**天线或抛物面天线**（称为**超小型口径天线终端VSAT**）等较小直径的接收器信号性能较高。通信卫星作为数据传输的介质正被广泛使用，从提供国际计算机通信网互连，到以高速率提供国内不同地区通信网互连通道。

卫星系统如图2-4(a)所示。图中仅表示单向传输通路，而大多数实际应用（每个地面站以不同频率操作上行与下行信道）使用一个双工通路。另外共用的配置包含中央集中地面站，它与若干个分布在各处的VSAT地面站通信。一般，计算机连到每个VSAT，它与连接到集线器的中央计算机通信。如图2-4(b)所示。一般，中央站用单一频率向所有VSAT广播，而每一 29
个VSAT以不同频率反方向发送。

为了与某个特定VSAT通信，中央站广播报文并在报文头中带有指定VSAT的标识。对于VSAT到VSAT的通信，所有报文都首先发送到中央站——经过卫星，然后，广播报文到指定接收者。下一代高质量卫星，可能不通过中央站在运载卫星上实施路由，直接从VSAT到VSAT通信。

2.1.6 地面微波

在架设物理传送介质非常昂贵或地理条件不允许的情况下，广泛使用**地面微波链路**提供通信链路。例如，横穿河流、沼泽或沙漠。由于并行微波束通过地球表面的大气层传播，会受到一些因素（如建筑结构与不利天气条件）的干扰。而通过卫星链路，其波束多在自由空间传播，受建筑物和气候的影响较小。然而，通过地球大气层的视线微波的可靠使用极限距离是50 km。

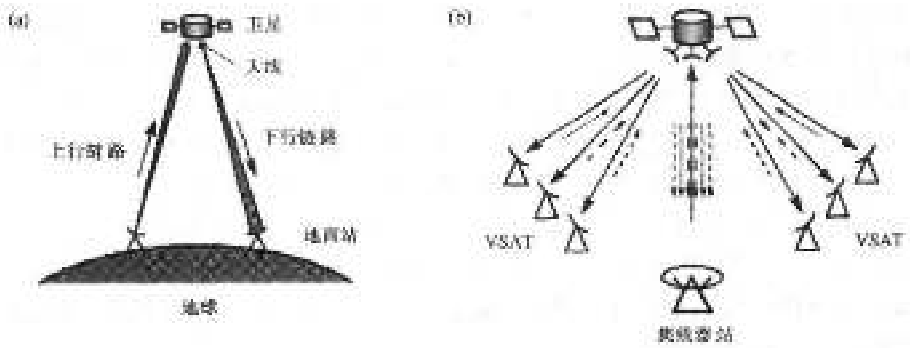


图2-4 卫星传输

(a) 点到点 (b) 多点

2.1.7 无线电波

在多数合适距离范围内，采用地面发送器与接收器设备用低频无线电波取代固定线路的连接。例如，为了连接分散在郊区的一些采集大量数据的计算机到远程数据记录/监控计算机，或为了将一个乡镇或城市中的计算机（或计算机终端）连接到本地或远程计算机。

对这样的应用，安装固定电缆将是昂贵的。在一个固定终端设备与一些分布计算机之间，

30 无线电波提供无绳（无线）链路。无线电波发送器（称为基站）置于固定终端设备点，如图2-5(a)所示，无线电波提供每一台计算机与中心站点之间的无线连接。

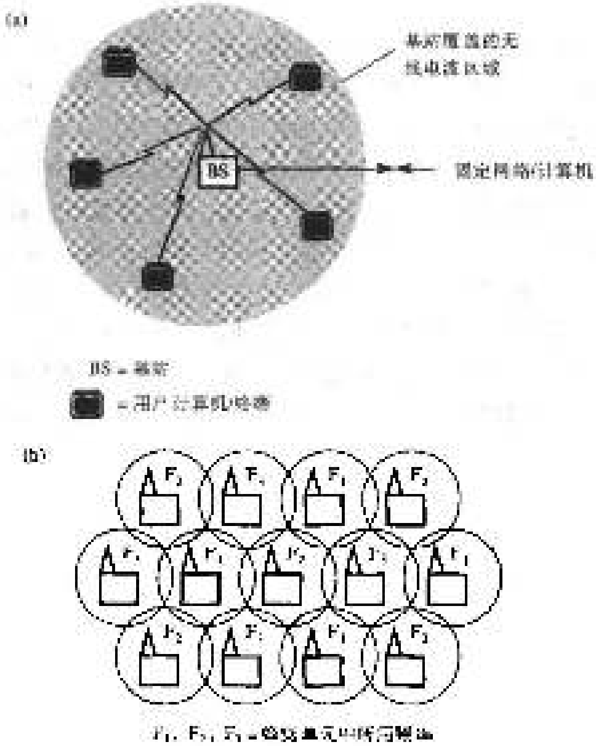


图2-5 地面无线电传输

(a) 单个蜂窝单元 (b) 多个蜂窝单元

对于需要更大覆盖区域或更高用户密度的应用，必须采用多个基站。每一个基站覆盖的区域是有限的（由于其输出功率的限制），所以，要提供足够信道以支持那个区域的全部负载。用蜂窝结构排列多个基站可以获得较广覆盖，如图2-5(b)所示。实际上，每个蜂窝单元的大小是可变的，由终端密度、地理范围等因素决定。

每个基站与它邻近的基站使用不同频带，然而，因为每一个基站覆盖的区域是有限的，网络的其他部分重用它的频带是可能的。基站与固定网络连接同前。一般，在每个蜂窝单元中每一台计算机可用的有效数据速率是每秒几万比特。

在一幢大楼中，每个办公室为了提供到计算机设备的无线连接可以使用相同排列。在这样的情况下，一个或多个基站固定在大楼的每一层，并连接到固定网络。为了覆盖区域中的所有计算机，每个基站提供到固定网络的无线连接。每当安装或移去一台新的计算机时，可 31
以避免重新接线，但代价是需要提供从无线信号转换成数据以及从数据转换成无线信号的无线单元。可用的数据速率通常比固定连接数据速率低。

2.2 衰减与失真源

不同的衰减与失真对传输信号的影响如图2-6所示。信号通过传输介质会受到衰减、带宽限制、时延失真和线路噪声等影响。虽然这些影响都会出现并产生综合的影响，但还将分别考虑每一个减损。 32

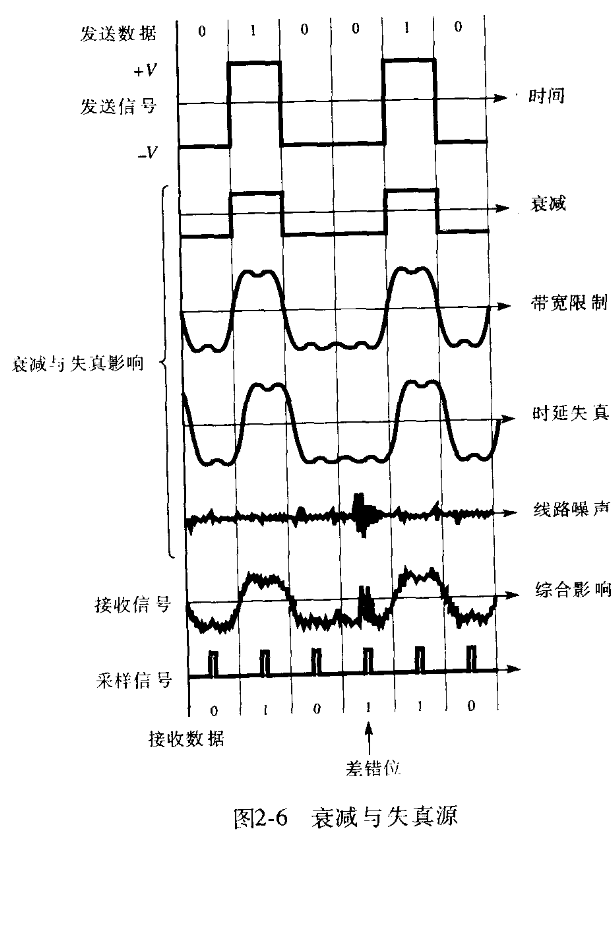


图2-6 衰减与失真源

2.2.1 衰减

信号沿着传输介质（导线）传播，它的幅度会减弱，这称为**信号衰减**。通常，若允许衰减，需对电缆的总长加以限制以保证衰减后的信号能被接收电路有效地分辨与处理。如果电缆过长，则应沿着电缆间隔地插入一个或多个**放大器**（也称为**中继器**）用来将接收到的信号恢复为其原来的电平。

信号衰减的增大可看作频率的函数。因此，由于信号由频段组成，信号也会失真。为了克服这个问题，对不同频率的信号采用不同值进行放大设计出了放大器。换句话说，采用称为**均衡器**的设备去补偿确定频带内的衰减。

我们用**分贝(dB)**测量衰减与放大（也称为**增益**）。如果 P_1 是发送信号的功率电平，P_2 是接收信号的功率电平，则

$$\text{衰减} = 10\log_{10}\frac{P_1}{P_2}\text{dB}$$

和

$$\text{放大} = 10\log_{10}\frac{P_2}{P_1}\text{dB}$$

因为 P_1 与 P_2 两者都有相同单位瓦特，所以分贝是无量纲的，它是两个功率电平的相对大小的度量。采用对数方法意味着多段传输信道的衰减/放大可简单地看作每一段衰减/放大的和。

实例2-1

两台通信设备DTE之间的传输信道由三段组成。第一段产生16 dB的衰减，第二段有20 dB的放大，而第三段产生10 dB的衰减。假定平均发送信号功率电平为400 mW，求信道的平均输出功率。

解1：

对于第一段，$16 = 10\log_{10}\frac{400}{P_2}$，因此 $P_2 = 10.0475\text{mW}$

对于第二段，$20 = 10\log_{10}\frac{P_2}{10.0475}$，因此 $P_2 = 1004.75\text{mW}$

对于每三段，$10 = 10\log_{10}\frac{1004.75}{P_2}$，因此 $P_2 = 100.475\text{mW}$

33 所以，平均输出功率电平=100.475mW

解2：

信道全部衰减 = (16−20)+10 = 6 dB

因此 $6 = 10\log_{10}\frac{400}{P_2}$，所以 $P_2 = 100.475\text{mW}$

更一般情况，在2.5.1节展开讨论。当在有限带宽信道（如PSTN）上发送二进制信号时，通常要用多个信号电平。这意味着每个信号单元可用多个二进制位表示。一般，如果信号电平的个数为M，则表示信号单元的二进制位的个数 m 由下式给出：

$$m = \log_2 M$$

例如，如果发送二进制的数据流采用四个信号电平，则每个信号单元用两个二进制数字表示。

信号变化的速率称为**信号速率**（R_s），用**波特**量度，即每秒发送信号单元的个数。它与数据比特率 R 的关系由下式表示：

$$R = R_s \log_2 M$$

2.2.2 带宽限制

任何通信信道/传输介质（双绞线、同轴电缆、无线电波等）都有一个相关的固定带宽，它指定发送信号在信道中不衰减有效成分的正弦波频率的范围。因此，当在信道上传送数据时，我们需要定量地考察信道带宽对发送数据信号产生的影响。

我们用称为**傅立叶分析**的数学工具，将任何周期信号（即信号在称为周期的相同时间间隔内重复）表示成一个正弦波频率成分的无穷级数。信号的周期确定**基频**成分：以秒为单位的周期的倒数，它以**每秒循环次数**（Hz）为单位。其他频率分量都是基频的倍数，这些称为基频的**谐波分量**。我们可用数学形式将任何周期的波形表示如下：

$$v(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega_0 t + \sum_{n=1}^{\infty} b_n \sin n\omega_0 t$$

其中

$v(t)$是信号电压，表示为时间的函数，

ω_0是基频分量，以每秒的弧度为单位，34

$T=2\pi/\omega_0$是波形的周期，以秒为单位，

a_0, a_n 与 b_n 称为**傅立叶系数**，对于一个确定的波形，由下列积分可得：

$$a_0 = \frac{1}{T}\int_0^T v(t)\mathrm{d}t$$

$$a_n = \frac{2}{T}\int v(t)\cos(n\omega_0 t)\,\mathrm{d}t$$

$$b_n = \frac{2}{T}\int v(t)\sin(n\omega_0 t)\,\mathrm{d}t$$

第一个积分求得的a_0是周期T上信号的平均值，称为**直流成分(DC)**。

从数据传输观点来看，值得注意的信号是二进制序列。虽然实际中传输的二进制信息是随机的可变序列，但为了分析方便起见，我们将考察选定周期序列，如101010…，110110…，11101110…等。第一例是有限序列10的重复，它的周期是两个比特间隔；第二例是有限序列110的重复，它的周期是三个比特间隔，等等。由此，我们能推断序列101010具有最短周期，它产生最高的基频。这意味着其他序列产生的频率都小于它。为了分析方便起见，具有最短周期的序列称为**最坏情况序列**。

有两种基本二进制信号类型用于传输目的：**单极性**与**双极性**（参见2.3节）。每一种的实例如图2-7(a)所示。关于单极性信号，信号的幅度在正电压+V伏与0伏之间变化。这样的信号称为**归零信号**（RZ）。关于双极性信号，信号的幅度在正电压与负电压（+V与–V）之间变化。这样的信号称为**非归零**（NRZ）信号。一个单极性信号的平均信号电平为$V/2$，而一个双极性信号的平均信号电平为0。单极性信号幅度变化是V，而双极性信号幅度变化是$2V$。这些差别产生稍微不同的傅立叶级数，这两个信号类型的傅立叶级数如下：

单极性

$$v(t)=\frac{V}{2}+\frac{2V}{\pi}\{\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t-\cdots\}$$

双极性

$$v(t)=\frac{4V}{\pi}\{\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t-\cdots\}$$

其中

$v(t)$是信号电压，表示为时间的函数

ω_0是基频分量，以每秒的弧度为单位

$f_0=\omega_0/2\pi$是基频，单位为Hz

$T=1/f_0$是基频的周期，单位为秒

35 从这些表达式可推出任何周期的二进制序列由下列正弦信号的无穷级数组成：基频成分f_0，第3个谐波分量$3f_0$，第5个谐波分量$5f_0$ ……注意，对于二进制序列，只有奇数谐波分量出现，并且幅度随着频率增加而减小。为了用实例说明这个结果，双极性信号的基频、第3个谐波及第5个谐波如图2-7(b)所示。

因为通信信道具有有限带宽，从前面的讨论可推出，当二进制数据信号在信道上发送时，只有处于信道带宽范围内的那些频率分量能被接收。关于二进制信号101010……的实例来说，结果如图2-7(c)所示。由此我们能看到，信道带宽越宽，可接收的较高频率分量越多。因此，接收到信号越接近原始（发送）信号。

信道的带宽以赫兹（Hz）度量，带宽作为频率的函数，如图2-7(d)所示。在该图中，有三个可选的带宽：第一个带宽允许频率不超过f_0的正弦波没有衰减地通过；第二个带宽允许频率不超过$3f_0$的正弦波没有衰减地通过；每三个带宽允许频率不超过$5f_0$的正弦波没有衰减地通过。然而，当发送一个仅有两个电平（二进制）的信号时，接收器只在每个比特单元间隔的中点采样信号，这意味着接收器仅需要在采样的瞬时识别二进制1与0的电平，而其他时间实际信号形状并不重要。由于前面的讨论，序列101010……产生最高频率的分量，而全部是1或者全部是0的序列等价于某一幅度其频率为0，所以，带宽从0 Hz到比特率的一半（以Hz表示）的信道（即仅通过从频率为0到最大频率分量为二进制序列101010的基频的信道）给出的良好性能。

实例2-2

通信信道以速率500 bps发送二进制信号，假定在最差接收序列情况下(a) 仅接收基频信号；(b) 接收基频及第3个谐波；(c) 接收基频、第3个谐波及第5个谐波。求要求的最小带宽。

解：速率为500 bps的最差序列101010……，基频分量是250 Hz。因此，第3个谐波的频率是750 Hz，第5个谐波的频率是1250 Hz，所以每种情况要求的带宽如下：

(a) 0 ~ 250 Hz (b) 0 ~ 750 Hz (c) 0 ~ 1250 Hz

在2.5节，我们将会看到，被发送信号的每次幅度改变也可以多于1比特，因此，增加了数据比特率，但是信道的带宽通常限制了可获得的最大数据速率。**奈奎斯特**给出无噪声信道最

36 高信息发送速率C的公式：

$$C=2W\log_2 M$$

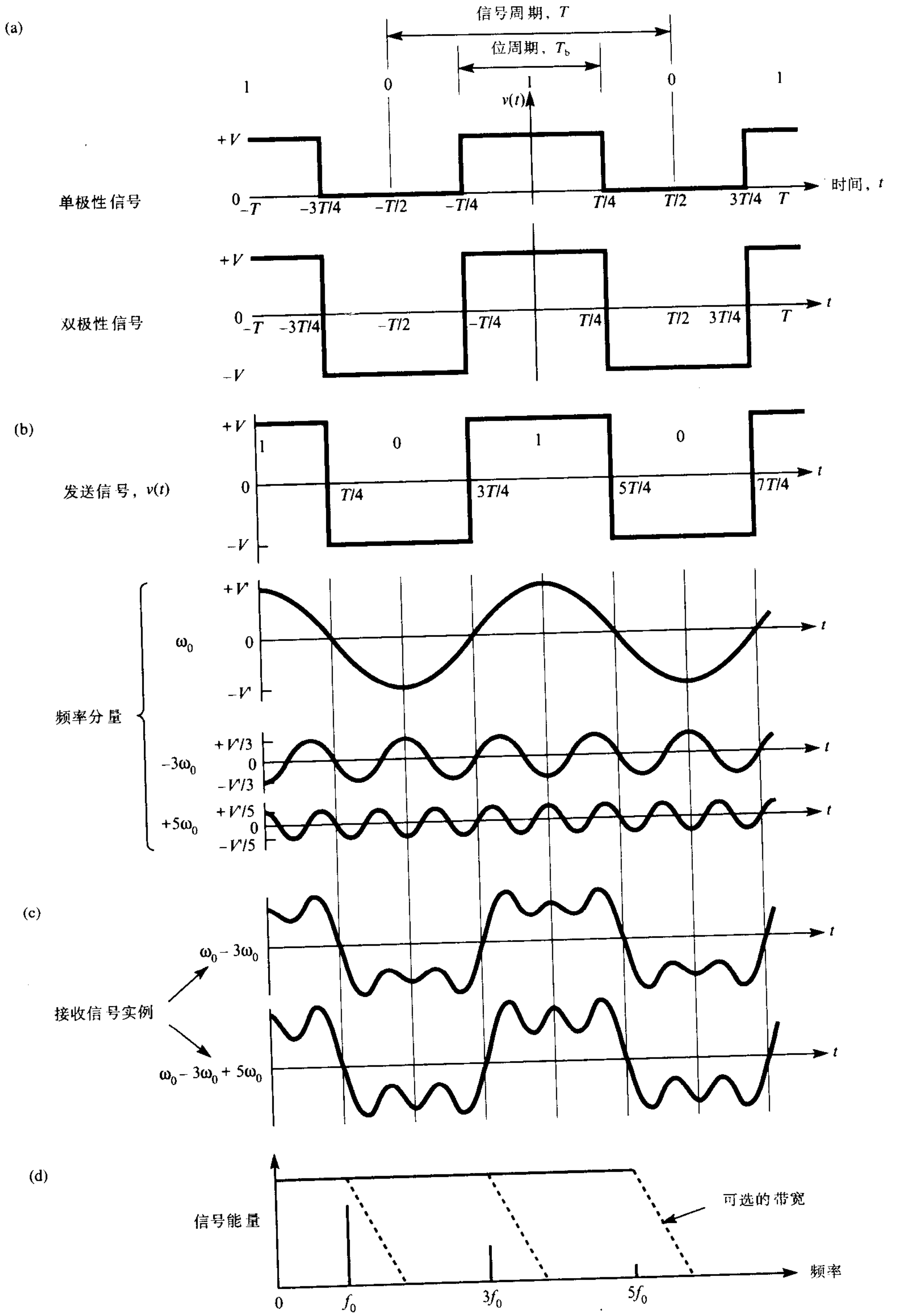


图2-7 有限带宽影响

(a) 可选的二进制信号 (b) 周期二进制序列的频率分量 (c) 接收信号实例 (d) 带宽表示

其中W是信道的带宽，单位为Hz，M是信号单元的电平数。实际上，我们将在3.2.3节看到，为了传输控制的目的，二进制数据会增加额外的位，所用的数据速率时常比实际的比特率低。当我们在信道上发送信息时，涉及三种速率：**信号单元**速率，比特率以及数据速率——它们可以相同，也可以不同。

每位的持续时间（用秒表示）T_b是比特率R（用每秒位数表示）的倒数。因此，R与信号单元时间周期T_s的关系由下式表示：

$$R=\frac{\log_2 M}{T_s}=\frac{m}{T_s}\ \text{bps}$$

由于T_b是R的倒数，每位的有效持续时间T_b与T_s的关系由下式表示：

$$T_b=\frac{1}{R}=\frac{T_s}{m}$$

结合这两个表达式求得众所周知的传输信道**带宽效率**B，它定义如下：

$$B=\frac{R}{W}=\frac{m}{WT_s}=\frac{1}{WT_b}\ \text{bps Hz}^{-1}$$

从这个表达式，我们可推出比特率与可用带宽之比越高，带宽效率也越高。B的典型值在0.25到3.0 bps Hz^{-1}之间，第一个数值对应于可用带宽低比特率，第二个数值对应于要求相对高信号速率的高比特率。所以，带宽效率越高，有关设备的设计参数越精确，因此成本也越高。

实例2-3

在PSTN上传输数据，传输方案采用每个信号单元8个电平。如果PSTN的带宽是3000 Hz，求奈奎斯特最高数据发送速率C与调制效率B

解：

$$\begin{aligned}C&=2W\log_2 M\\&=2\times3000\times\log_2 8\\&=2\times3000\times3\\&=18\,000\text{bps}\end{aligned}\qquad\qquad\begin{aligned}B&=\frac{1}{WT_b}\\&=\frac{18\,000}{3000}=6\ \text{bps Hz}^{-1}\end{aligned}$$

37～38

实际上，由于噪声等影响，这两个值将小于上面的计算值。

2.2.3　时延失真

正弦信号沿着传输线传播的速率随信号频率而变化。因此，当传输由各种频率分量组成的数字信号时，在接收端，到达的各种频率分量信号有不同的时延，其结果引起接收信号的**时延失真**。当传输数据比特率增加时，失真的数大小也增加。其理由如下：因为比特率增加，所以与每个比特转换相关的若干频率分量被延迟，开始干扰与之后的比特相关的频率分量。时延失真也称为**码间干扰**。它的影响是需要修改接收信号比特转换的时刻。通常因为接收信号在每个比特单元的中点采样，由于比特率增加，这可能导致对接收信号错误的解释。

最好借助于**眼图**（eye diagram）观察传输信道上的码间干扰，实例如图2-8所示。这个图是通过在示波器上显示接收到的信号得到的，而示波器被信号比特转换触发。所以，假定接收信号包括随机二进制信号1与信号0的转换，则示波器显示所有可能信号相互重叠。如图所示的两个实例，没有码间干扰信号如A所示，有码间干扰信号如B所示。我们可推出，干扰强

度越高，中央部分（称为眼）变得越小。

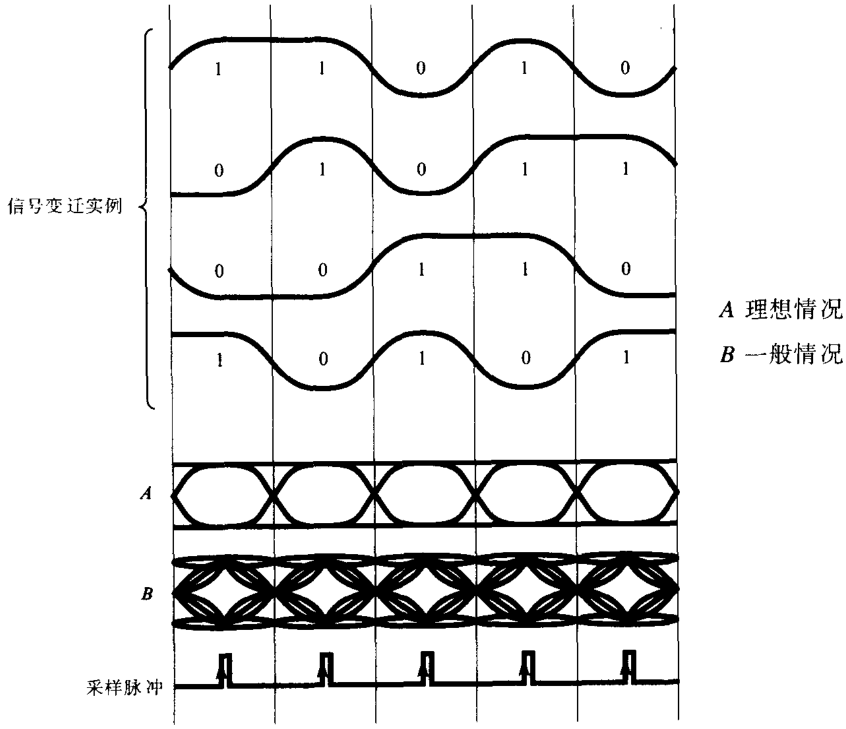


图2-8 码间干扰产生的（二进制）眼图实例

2.2.4 噪声

在没有信号传输时，理想的传输线路或传输信道应具有零电平的信号。但实际上，即使没有传送信号，线路上也存在随机微扰。这称为**线路噪声电平**。在极端情况下，由于发送信号衰减，它的幅度降至与线路（背景）噪声同一级别。为表明噪声大小，有一个与传输介质相关的重要参数是接收信号的平均功率S与噪声功率电平N的比。比值S/N称为**信噪比**（SNR），通常以分贝为单位表示：

$$\mathrm{SNR}=10\log_{10}\left(\frac{S}{N}\right)\mathrm{dB}$$

显然，SNR值高意味着高功率的信号压倒噪声信号，是高质量信号。相反，SNR值低意味着是低质量的信号。传输信道理论上的最大信息（数据）速率与SNR有关，可用香农－哈利公式确定，这称为**香农－哈利定律**，表示如下： 39

$$C=W\log_2\left(1+\frac{S}{N}\right)\mathrm{bps}$$

其中C是信息（数据）速率，单位为bps；W是线路/信道带宽，单位为Hz；S是平均信号功率，单位为瓦；N是随机噪声功率，单位为瓦。

实例2-4

假设PSTN具有带宽3000 Hz和典型的信噪功率比20 dB，求理论上最高的信息（数据）速率。

40 解：$\mathrm{SNR}=10\log_{10}\left(\frac{S}{N}\right)$ 所以 $20=10\log_{10}\left(\frac{S}{N}\right)$

因此，$\frac{S}{N}=100$ 现在 $C=W\log_2\left(1+\frac{S}{N}\right)$ 所以 $C=3000\times\log_2(1+1000)$
$=19\,963\text{bps}$

在2.1节中，当我们讨论双线开放线与双绞线传输线时，已发现串扰是噪声的一种来源。串扰由相邻导线间非预期的电容耦合引起。耦合引起一条导线上正在传输的信号被相邻导线拾取为一个小的但是有限的（噪声）信号。信号串扰的实例是打电话时听到其他电话的声音；尽管没有讲话，但在线路上依然有信号出现。

有几种串扰类型，但在大多情形下，对减损限制最大的是**近端串扰**或**NEXT**。这就是所谓**自串扰**。因为它是由发送电路与本地接收电路输入端许多微弱信号耦合（因而干扰）产生的强信号输出导致的。如2.2.1节指出，接收信号通常有较大的衰减与失真。因此来自发送部分的耦合信号的幅度与接收信号相差不大。

众所周知，目前使用专用集成电路**自适应NEXT消除器**克服这种类型的减损。典型的结构如图2-9所示。消除器电路自适应地形成一个同串扰信号一模一样的衰减信号，由本地发送器耦合到接收线上，从接收信号中减去这个信号。目前，这样的电路有许多应用，包括在UTP电缆中以高比特率传送数据。

另一种噪声形式是**脉冲噪声**。此名称意味着这是由与外界活动相关的电能脉冲引起。例
41 如闪电或电话网中老式电话交换系统交换机电路上的电脉冲。一般后例会在线路上听到卡嗒声。它对通话影响不大，但严重影响数据传输。例如，一个延续半秒的脉冲噪声（卡嗒），可以影响以2400 bps数据传输速率传输的数据的1200位。幸好，这种噪声源相对较稀少。

串扰与脉冲噪声都是由外部活动在传输线上的电活动引起的。相反，还有第三种类型的噪声，称为**热噪声**，出现在与任何外界影响无关的电子设备与传输介质上，它由电子设备或传输线路材料中与每个原子相关联的电子的热扰动引起。温度在绝对零度以上，所有传输介质都会经受热噪声。它由振幅连续变化的随机频率分量（完全频谱）组成，所以也称为**白噪声**。

应该强调是香农－哈利定律给出理论上最大信息速率。实际上，当考虑噪声影响时，计算所用的最小信号电平与噪声电平之比是重要的，以求得最小**位误差率**。即在规定时期内，接收器可接受低概率的单比特出错率。例如：10^{-4} 的位误差率意味着平均每收到10^4位就有1位错误。

信号的每位的能量E_b用焦耳（瓦×秒）表示，它由

$$E_b = ST_b\ \text{瓦–秒}$$

给出，其中S是以瓦为单位的信号功率，T_b是以秒表示1位的时间周期。现在数据传输速率R等于$1/T_b$，因此

$$E_b = \frac{S}{R}$$

任意传输线上带宽1Hz的热噪声的电平由公式

$$N_0 = kT\ \text{瓦}\ \text{Hz}^{-1}$$

给出，其中N_0是以瓦Hz^{-1}为单位的噪声功率密度。k是波尔兹曼常数（1.3803×10^{-23}焦耳K^{-1}），T是以开尔文（K）为单位的温度。

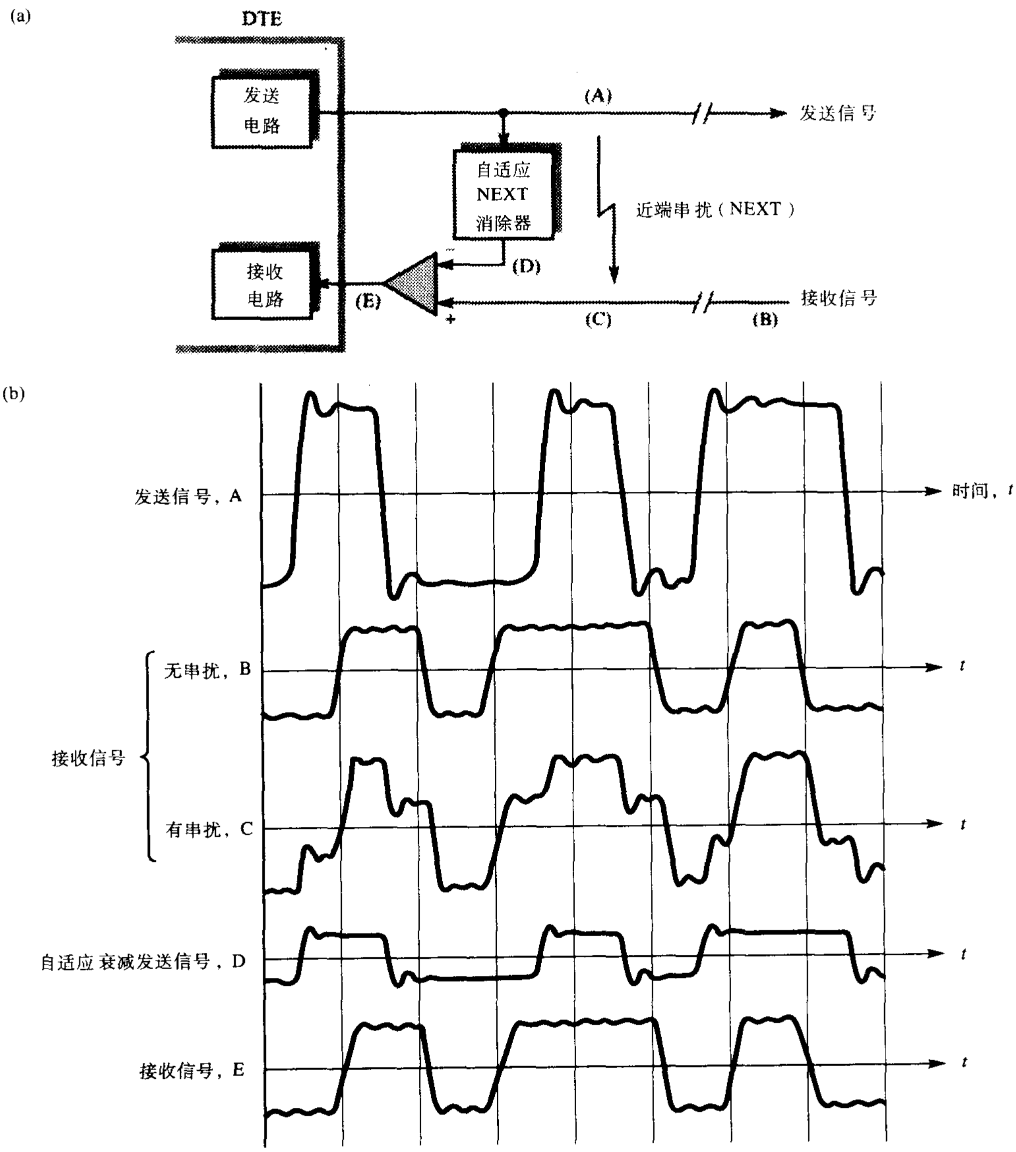


图2-9　自适应NEXT消除器

(a) 电路结构　(b) 波形实例

为了定量说明噪声的影响，每位的能量E_b可用每Hz的噪声能量比N_0来表示：

$$\frac{E_b}{N_0}=\frac{S/R}{N_0}=\frac{S/R}{kT}$$

或者，以分贝表示：

$$\frac{E_b}{N_0}(\mathrm{dB})=10\log_{10}\left(\frac{S}{R}\right)-10\log_{10}(kT)$$

我们可立即推出所要求信号功率电平S，以求得可接受的E_b/N_0——最小位误差率。它随着温度T与比特率R增加。 42~43

表达式E_b/N_0也可用信道带宽 W 表示。因为N_0是用瓦Hz^{-1}为单位的噪声功率密度，对于信道带宽W Hz，接收信号的噪声功率N由下式

$$N = WN_0$$

给出。因此

$$\frac{E_b}{N_0} = \frac{S}{N}\frac{W}{R}$$

或者用分贝表示：

$$\frac{E_b}{N_0}(\mathrm{dB}) = 10\log_{10}\left(\frac{S}{N}\right) + 10\log_{10} W - 10\log_{10} R$$

在2.5节中将看到，在PSTN上发送数据，必须首先将数字数据用称为**调制**的方法转换成模拟形式。在目的地，同样地，用称为**解调**的相反过程恢复数字数据。实际上，有各种各样的调制/解调方案，对每个不同方案，要达到一个指定位误差率，要求一个可变的E_b/N_0的值。例如，为了达到10^{-6}的位误差率，采用幅移键控或频移键控（参见2.5.1节），要求E_b/N_0是13 dB，而相移键控，E_b/N_0要求10 dB。所以，已知信道带宽，对于不同调制方案，为了得到相同的数据速率，最小可接受信噪功率比是变化的。

实例2-5

在带宽3000 Hz的PSTN上发送数据。如果接收器平均信噪功率比是12 dB，假定(a)$E_b/N_0 =$ 13 dB，(b)$E_b/N_0 =$ 10 dB，确定能获得的最大数据速率，确定每种情况的带宽效率。

解： $$10\log_{10} R = \frac{S}{N}(\mathrm{dB}) + 10\log_{10} W - \frac{E_b}{N_0}(\mathrm{dB})$$

带宽效率，$B = R / W$

(a) $10\log_{10} R = 12 + 10\log_{10} 3000 - 13 = 33.77$ 因此 $R = 2382.32$ bps与 $B = 0.79$

44 (b) $10\log_{10} R = 12 + 34.77 - 10 = 36.77$ 因此 $R = 4753.35$ bps与 $B = 1.58$

2.3 信号类型

当两个设备（DTE）距离较近，并且传输速率不高时，可通过双线开放线和简单的接口电路直接传送数据，接口电路应将通信设备内部的信号电平转换成适用于互连电缆的信号电平。但当DTE之间距离增大并且比特率增高时，则必须采用更复杂的接口电路与技术。进一步，如果DTE是在国家（或世界）的不同地区并且没有合适公用数据通信设施可用时，则最经济的方法是采用负责提供电话与其他电信服务的邮电部门提供的公用线路，后者称为**PTT**。当我们采用这种通信介质时，一般必须将源DTE输出的电信号转换成用于传递话音消息的模拟信号。同样地，在接收方，我们必须将这些信号转换目的DTE所适用的信号。完成这些功能转换的设备称为**调制解调器**。以后几节，将讨论调制解调器中使用的不同信号类型以及其他传输线路形式。

2.3.1 V.28

为了不同厂商生产的设备可使用电话交换网中的传输设备，可使用各种调制解调器。它们以不同的速率操作，并用不同类型的调制方法。这将在2.5.1节讨论。有许多调制解调器制造商，为了保证不同厂商生产的两台调制解调器能互相通信，每种调制解调器类型的精密操作已经标准化。所有类型的调制解调器都使用规定的标准接口。它指定电信号的个数与功能、

插头与插座的物理尺寸以及针脚布局。这称为**EIA-232D接口**（由EIA定义）或**V.24接口**（由ITU-T定义）。EIA-232D接口是早期RS-232A、B与C标准的最新版本，最早起源于20世纪50年代晚期。

通常，DTE与调制解调器的距离相对较近，因为模拟电话线路的带宽相对较低（常规为3000 Hz），因此最大比特率也很低。因为引入调制解调器，采用同一接口作为连接任何面向字符的外围设备到计算机的标准。例如，终端或打印机与同一计算机相连。因此，允许不同厂商生产的外围设备与同一计算机相连。

由于在DTE中，相邻的集成电路之间距离很近（小于几个厘米），因此用来表示二进制数据的信号电平通常很低。例如，数字设备中常用的逻辑系列是**晶体管－晶体管逻辑**（TTL）。 45
采用2.0V～5.0V的电压表示二进制的“1”，而用0.2V～0.8V的电压表示二进制的“0”。在最坏情况下，如果电压靠近某一极限，两个信号电平之间的电压会产生一个中间状态，即使普通信号衰减或电干扰也会引导出一个错误的解释。因此，连接两个设备时所用的电压电平通常大于设备内部集成电路所用的电平。

EIA-232D/V.24接口的信号电平是V.28推荐标准规定的。两个标准所用的信号与相应的接口电路如图2-10所示。线路上所用信号电平，对于地信号来说是对称分布的，至少是3V：即+3V表示二进制“0”，而-3V表示二进制“1”。实际上，使用的电压电平由接口电路上电源电压决定，也有±12V，甚至不常用的±15V。在设备中，发送电路将低电平信号转换成传输线路的高电平信号。同样，接收电路执行相反转换。称为**线路驱动器**与**线路接收器**的接口电路执行所需电压转换功能。

接口采用相对高的电压电平意味着信号衰减与噪声比TTL电平得到改善。EIA-232D/V.24接口通常使用扁平电缆或具有单地线的多芯电缆来连接各设备。这样势必受到单线噪声的影响。为了降低串扰，常在发送器电路输出端并联一个**电容器**，并使传输信号变缓从而降低信号中的高频噪声。当线路加长或信号比特率提高时，线路衰减使接收信号电平降低，这样，即使外部噪声信号不大，也将产生操作错误。EIA-232D与V.24标准一般适用的最大物理距离小于15 m，比特率低于20 kbps。但将外围设备连接到计算机时，也可能超出上述极限。 46

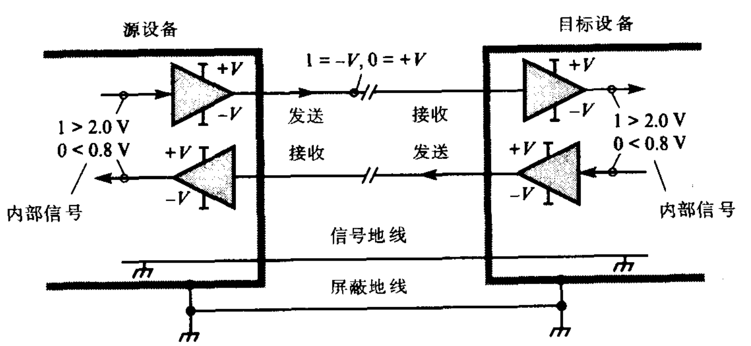


图2-10 V.28信号电平－单端/非平衡

2.3.2 20mA电流环路

可代替EIA-232D/V.28接口标准的是20 mA电流环路。如名字所说，它采用电流信号而不是电压信号。虽然它没有提高可用比特率，但实质上增加两台通信设备间可能的物理距离，基本原理如图2-11所示。

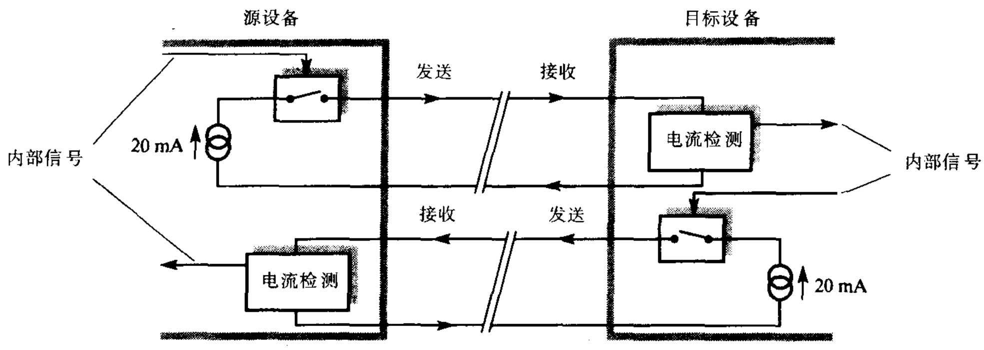


图2-11　20mA电流环路

基本上，用发送的比特流控制开关的状态（如继电器或其他类似设备），开关闭合表示"1"，通过20 mA的电流（脉冲），开关断开表示"0"，禁止电流通过。在接收端，采用电流－敏感电路和复制的发送二进制信号检测电流。

电流环路接口噪声抗扰性比基本电压驱动接口好得多，从图2-11可看到，每个信号用了一对线。这意味着任何外部噪声信号一般影响一对线，通常称为**共模噪声**或**拾取噪声**，在基本的电流－灵敏接收器电路上影响最小。因此，20 mA电流环路接口特别适合于长线（直到1km）驱动，但因传输速率受开关转换速度与电流灵敏电路限制，比特率适中。由于这个原因，一些厂商通常对一个设备提供两个独立接口，一个产生输出信号电压，而另一个是20 mA电流信号。然后用户可按照设备之间的物理距离决定使用何种接口。

2.3.3　RS-422A/V.11

如果物理距离与比特率都增大，则应采用另一个RS-422A/V.11标准。它是基于双绞线电缆与一对**差分**（也称为**平衡的**或**双端的**）**发送器**与**接收器电路**。典型电路结构如图2-12(a)所示。

47 差分发送器分别产生一对大小相等极性相反的信号，用于发送二进制"1"与"0"信号。差分接收器仅在它的两个输入端对两个信号的差异是敏感的。两根线上拾取到的任何噪声不会影响接收器的操作。所以差分接收器称为具有优良的**共模抑制**性质。RS-422A的一种变型RS-422A/V.10可用于接受由EIA-232D接口的差分接收器输出的单端（非平衡）电平。RS-422A适于距离在10米之内的，速率10 Mbps，以及距离在1000米之内，速率100 kbps，并采用双绞线电缆。

传输线路中的一个重要参数是**特征阻抗**Z_0，只有当线路终端连接一个等于Z_0的电阻时，接收端才能接收全部传输信号，否则将发生**信号反射**，使接收信号大大地失真。所以正常的线路终端连接一个电阻Z_0，阻值位于50到200Ω之间。两种信号类型的电缆长度与传输速率的汇总在图2-12(b)中给出。

2.3.4　同轴电缆信号

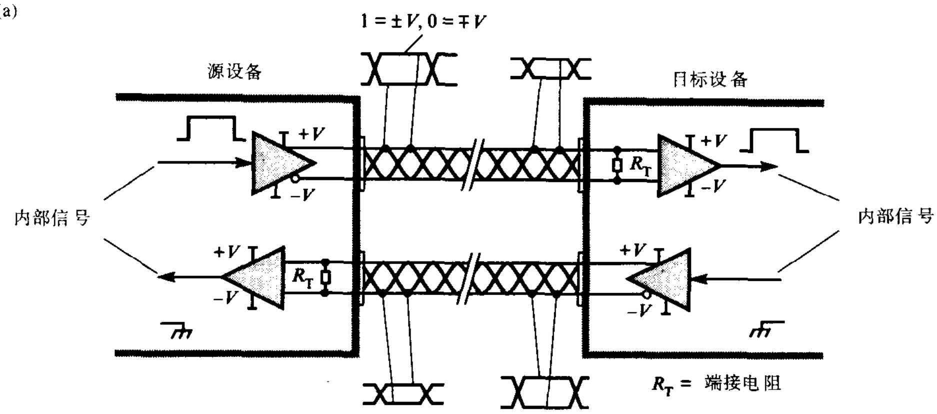
与模拟电话交换网的连接可用的低带宽相反，同轴电缆具有350 MHz（或更高）的带宽。

48 高带宽的使用有两种方法：

1）**基带方式**　所有可利用的带宽都被用作传输高比特率（10 Mbps或更高）信号的传输通路（信道）。

2）**宽带方式**　电缆中可利用的带宽被分为一系列更小带宽的子信道（因此有若干个传输通路）。

(a)



(b)

信号类型	长度（m）	最大传输速率（bps）
RS-423A/V.10	10	100 000
	100	10 000
	1000	1 000
RS-422A/V.11（非终结）	10	1 000 000
	100	100 000
	1000	10 000
RS-422A/V.11（终结）	10	10 000 000
	100	1 000 000
	1000	100 000

图2-12 差分信号

(a) EIA-422A/V.11信号电平 (b) 最大传输速率/距离

1. 基带方式

在基带方式中，通常电缆被单端电压源驱动。由于同轴电缆的形态，外部干扰的影响极小。许多匹配的发送与接收接口电路可使用同轴电缆。一种典型连接如图2-13(a)所示，它也说明线路以正确端接阻抗 Z_0 终结的作用。这种方式适合于在几百米的范围内以10 Mbps传输数据。

在某些应用中，电缆专门用于两个系统之间的数据传输，即**点对点配置**。在另一些应用中，通常由若干个系统分时共享这条高比特率传输信道，称为**多站**或**多点配置**。两种结构如图2-13(b)所示。 49

时分复用（TDM）用于共享基带传输信道可用容量，有两种TDM类型：

1）同步（或按固定周期） 每个用户按一定的时间间隔（同步）接入信道。

2）异步（或按请求） 每个用户随机地接入信道，但当一个用户接入后，在传输期间该信道为其独用。这两种TDM方式如图2-13(c)所示。

在第3章中介绍，两个系统（DTE）之间传输的数据，采用同步方式以固定长度帧（字符块或字节块）的形式进行。为了保证所有与（共享）电缆连接的系统都能在规定的时间内发送数据。在每一帧开始时，都有一个特殊位模式，称为**同步模式**（简称**sync模式**）。因此，系统可以在整个帧周期中识别出每个帧的起始位置和帧的位置（帧数）。采取异步TDM方式时，因为每个系统都是随机接入的，要能检测出每个新帧的开始（同步模式），就需要采用一种机制使得每个系统都以公平的机会接入信道。这将在第6章讨论。异步TDM用于特定类型的**局域数据网**上。

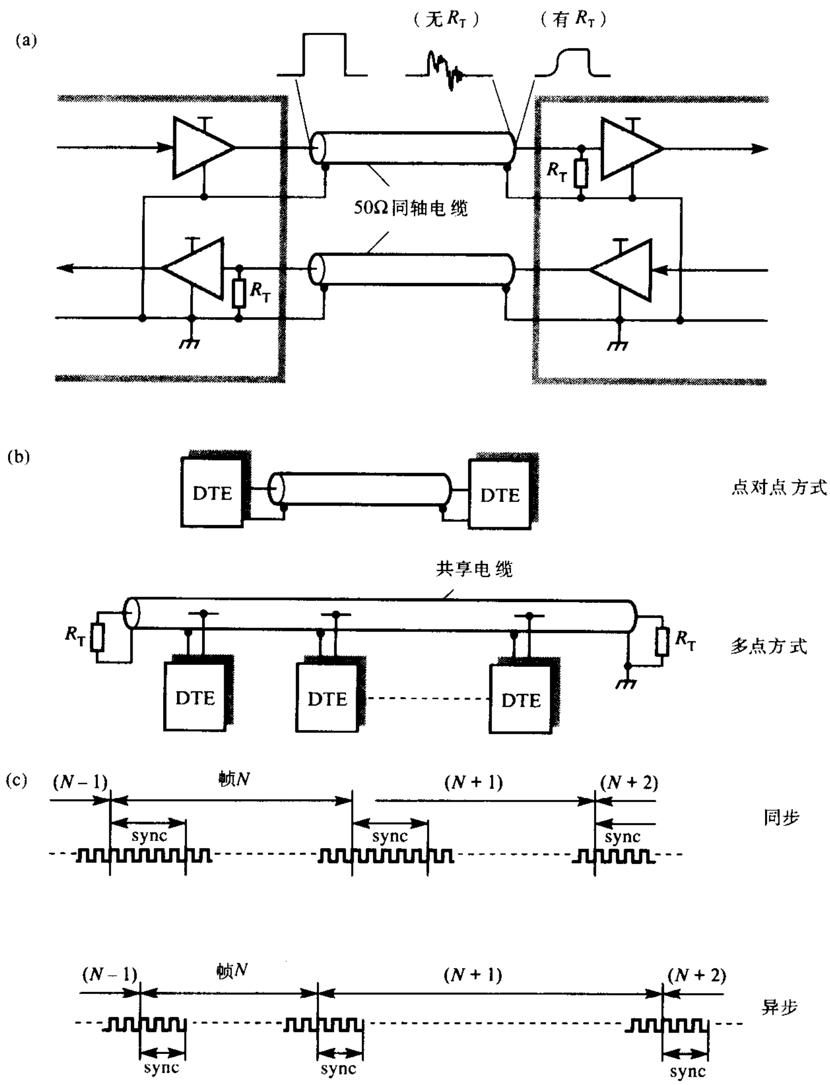


图2-13 基带原理

(a) 同轴电缆信号 (b) 连接方法 (c) TDM方法

2. 宽带方式

使用宽带方式时，采用称为**频分复用**（FDM）的技术，每条分布（同轴）电缆可提供多个（独立的与并发的）传输信道。FDM要求在每个连接设备与电缆之间有一个称为**射频**（RF）**调制解调器**的设备——原理与PSTN上所用（音频）调制解调器相同。我们用术语“射频”是
50 因为每个信道所用的频率是在无线电频谱中。在传输方向（正向）选择的（载波）频率用发送的数据进行调制，在接收方向（反向）解调选择的频率，获得接收的数据。

每个信道要求带宽由期望数据（比特）率与调制方法决定。典型的RF调制解调器带宽效率是每Hz在0.25与1.0比特之间。因此，一个9600 bps信道要求带宽约20 KHz，而10 Mbps信道要求带宽大约18 MHz。

宽带传输与RF调制解调器工作原理在图2-14中表示。在调制解调器内调制通常分两个阶段进行。首先采用相移键控或频移键控方式，用被传输的数据对所选频率进行调制。然后将

调制信号与第二个频率混频（复合），使变换后的频率落在指定频带内。通过如图2-14所示的滤波器，仅允许发送（在输出上）或处理（在输入上）指定频带的信号。 51

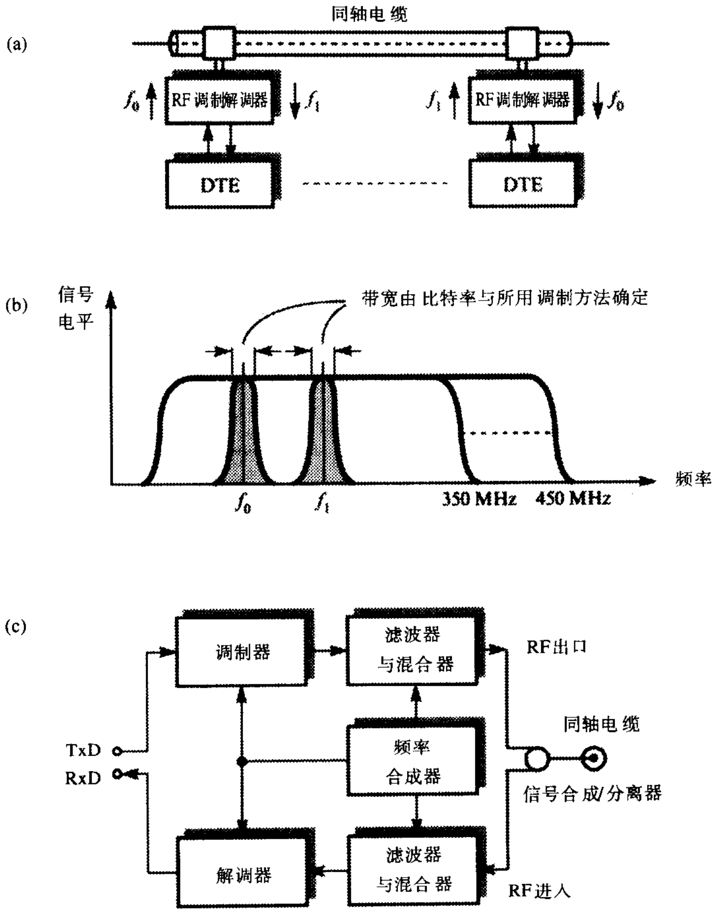


图2-14 宽带原理

(a) 电缆结构 (b) 带宽 (c) RF调制解调器结构

2.3.5 光纤信号

光纤信号编码有各种方案，双极性编码方案如图2-15所示。这种编码产生三级光强度输出。适用于处理从直流信号（对应于连续的二进制全“0”或全“1”串的零频率）直到50 Mbps的电信号。三级光强度输出分别是0、最大功率的一半与最大功率。发送模块采用特殊的连接器与高速LED完成内部二进制电压信号到三级光强度信号的转换。光纤信号的其他编码方案在3.3.1节中给出。

在接收端，光纤通过特别的连接器，接入带有高速光电二极管的接收模块。采用必要的电子控制，使发光二极管输出与光强度级别相应的电信号，转换成与二进制“1”与“0”对应的内部电平。

目前光纤主要采用点到点模式，如同基带同轴电缆，将有效传输容量用于单独一条高比特率信道，或者用于（通常同步）TDM技术，在单一链路上分出多条低比特率信道。

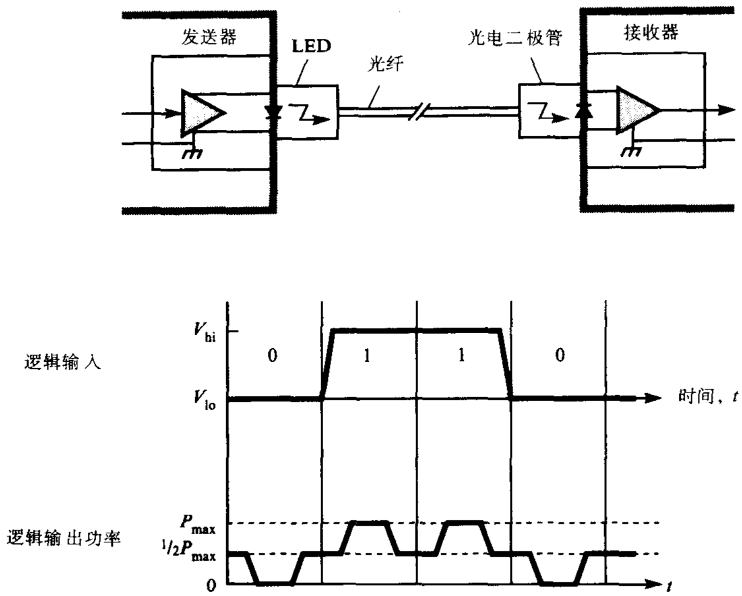


图2-15 光纤信号

2.3.6 卫星与无线电

52 在2.1节，我们简要说明由频分多路复用得到的卫星与其他无线电波系统的传输信道。此外，每个信道有效（基带）容量采用同步时分多路复用技术进一步被划分。

为了控制对有效容量的访问使用不同**访问控制方法**：

随机访问 所有站点以随机（非控制）方式竞争传输信道。

固定分配 每个地面站或无线站都预先规定信道频率与信道时隙。

按需分配 当一个站点需要发送数据时，它首先向中心站请求信道容量，中心站向请求站分配请求容量（时隙）。

随机访问是最早的访问控制方法，用于控制访问单一（共享的）卫星信道。它仅在下列情况使用：第一，总提供负载只是有效信道容量极小的一部分；第二，所有传输都是随机分布。该技术称为Aloha，它首先由夏威夷大学使用，将分布在一些岛上的计算机连接至位于
53 Oahu岛上的中央计算机。方案的两种版本如图2-16(a)所示。

关于纯Aloha，当某个站点有报文（数据）要发送时，它就简单地发送（广播）。当第二个站点开始发送，而第一个站点还在发送它的报文时，两者发送受损，一个**冲突**发生。因此，仅当两个发送重叠概率很小时，该方案可放心地工作。假定随机生成报文，按这个方案获得的吞吐量平均值小于有效容量的20%。该方案可通过建立一个同步时隙结构改进，在时隙中对所有报文发送做出限制。这个方案称为**时隙Aloha**。一个传送可能破坏另一个传送，仅发生在相同时隙内，此时信道利用率超进30%是可能的。

关于固定分配。提前对每个站点都预先分配信道频率与信道时隙内。一般预先分配频率信道比分配时隙容易。例如，在基于中央集线器的卫星应用中，对每个VSAT通常预分配一个固定的频率信道，然后中央站点在另一预分配频率信道上广播。一般，因为仅有单个集线器到VSAT的信道，这个频带（因此，比特率）比VSAT到集线器传送所用频带宽。对每个VSAT到集线器信道，典型的比特率是64 kbps，而集线器到VSAT广播信道的比特率可达到2 Mbps。这种访问控制方法，称为**预分配频分多路访问**或**预分配FDMA**。如图2-16(b)所示。

我们用按需分配访问控制方法可得到更好信道利用。这个方案提供若干个时隙，称为**申请时隙**。一个VAST或无线站向集线器或基站提出申请使用一个或多个时隙。如果有空闲的容量可用，则中央站分配指定时隙用作传输，同时用一个**确认时隙**通知申请站。图2-16(c)说明这个方案，称为**按需分配时分多路访问**（TDMA）。

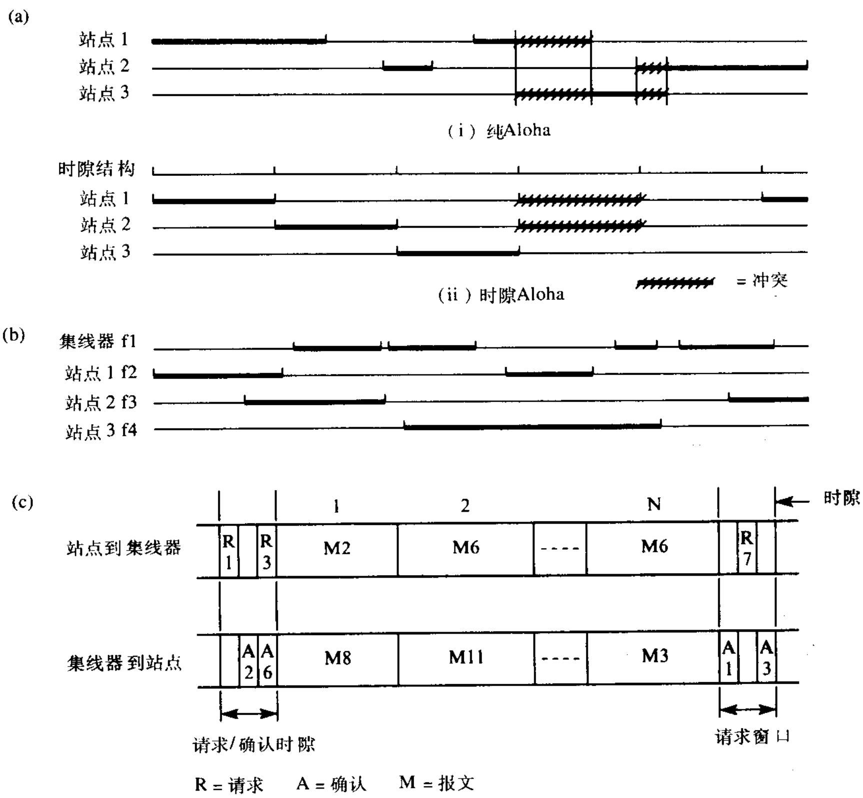


图2-16 卫星与无线电访问控制方法

(a) Aloha (b) 预分配FDMA (c) 按需分配TDMA

从图2-16(c)可看到，一个申请时隙比一般报文时隙短。申请报文包含申请站点的标识，假定每个报文多个时隙及需要的报文时隙的个数。然后对应要求确认报文说明应该用哪个报文时隙。所有传输由中央站直接做出，在报文头有要接收站的标识。为了降低申请时隙中冲突的概率，站点随机选择一个使用的时隙。如果确定一个冲突发生（由无响应表明），然后在下一个**申请窗口**再次进行。

最后，离开本小节话题前，应该强调，一段无线链路（信道）的BER通常比固定链路的高。因此在第3章中，将看到一般采用的小的报文分组和复杂的差错检测与纠错方法。 54

2.4 信号传播延迟

信号（电、光或无线电波）从介质一端传播到另一端常有一个短暂且有限的时间延迟。这称为介质的**传输传播延迟** T_p。最快的信号以光的速度（$3\times10^8\text{ms}^{-1}$）在自由空间传播（辐射）。双绞线或同轴电缆信号传播速度小于这个数，常用的近似值为$2\times10^8\text{ms}^{-1}$，就是说信号通过介质1m需要$0.5\times10^{-8}$秒。虽然这个数字看起来很小，但在某些情况，造成的延迟结果还是重要的。

在第3章中，我们一般将数据以块（也称为帧）的形式发送。对于块的接收，收到正确（或不正确）的确认要返回给发送方。所以，一个重要的数据链路的参数是链路上的**往返延迟**。即发送方发送块的第1位与接收方确认最后1位之间的延迟时间。显然，这不仅与以链路比特率发送帧所需时间有关，称为传输延迟T_x，而且与链路传播延时T_p有关。对于各种类型的数据链路，这两个时间所占的比例是可变的。因此，通常这两个时间用比率a表达如下：

$$a=\frac{T_p}{T_x}$$

其中

$$T_p=\frac{\text{以米表示的物理距离}S}{\text{以米每秒表示的传播速度}V}$$

而

$$T_x=\frac{\text{被发送的位的个数}N}{\text{以位每秒表示的链路比特率}R}$$

实例2-6

一个数据块长度为1000位，在两个DTE之间传送。对下列类型的数据链路，确定传播延迟与传输延迟的比率a

(a) 100 m双绞线传输，传输速率为10 kbps。

(b) 10 km同轴电缆传输，传输速率为1 Mbps。

(c) 50 000 km自由空间传输（卫星链路），传输速率为10 Mbps。

假定每种类型的电缆内电信号传播速度是每秒$2\times10^8 \text{ms}^{-1}$，而在真空传播速度是每秒$3\times10^8 \text{ms}^{-1}$。

55　解：

(a)
$$T_p=\frac{S}{V}=\frac{100}{2\times10^8}=5\times10^{-7}\text{s}$$
$$T_x=\frac{N}{R}=\frac{1000}{10\times10^3}=0.1\text{s}$$
$$a=\frac{T_p}{T_x}=\frac{5\times10^{-7}}{0.1}=5\times10^{-6}$$

(b)
$$T_p=\frac{S}{V}=\frac{10\times10^3}{2\times10^8}=5\times10^{-5}\text{s}$$
$$T_x=\frac{N}{R}=\frac{1000}{1\times10^6}=1\times10^{-3}\text{s}$$
$$a=\frac{T_p}{T_x}=\frac{5\times10^{-5}}{1\times10^{-3}}=5\times10^{-2}$$

(c)
$$T_p=\frac{S}{V}=\frac{5\times10^7}{3\times10^8}=1.67\times10^{-1}\text{s}$$
$$T_x=\frac{N}{R}=\frac{1000}{10\times10^6}=1\times10^{-4}\text{s}$$
$$a=\frac{T_p}{T_x}=\frac{1.67\times10^{-1}}{1\times10^{-4}}=1.67\times10^3$$

由实例2.6可得出结论：

若a小于1，则往返延迟主要由传输延迟决定。

若a等于1，则两种延迟的影响相同。

若a大于1，则传播延时将起主要作用。

进一步，在情况(c)中，注意当块与块之间不间断发送时，则任一时刻，两台DTE之间总共有$10 \times 10^6 \times 1.67 \times 10^{-1} = 1.67 \times 10^6$位数据在传送，即第1位到达接收DTE之前，发送DTE已发送1.67×10^6位。将在第3章中讨论这些的含义。

2.5 公用载波电路

当我们想要在同一座大楼或部门的两台DTE之间传输数据时，可以（相对）简单地装置电缆。常用的电缆是非屏蔽或屏蔽双绞线，同轴电缆或光缆。在某些场合下，也可以用无线电波。当我们想要在不同部门的两台DTE之间传输数据时，我们仅能用微波或卫星链路，或通信公司的公用载波线路实现。后一解决方案用途广泛，交换电路与租用（专用）电路都有可能。 56

我们按照可用性使用模拟PSTN或ISDN建立交换电路。虽然模拟PSTN是专门为语音通信设计的，但也可以利用调制解调器传输数据。在ISDN情形下，我们可直接建立呼叫与发送数据。高比特率也是可能的。

关于租用电路，在某些情况下，我们还必须租用PSTN线路（因此也需要调制解调器），但将在2.5.2节看到，目前，绝大多数租用线路是全数字化的。因为公用载波电路广泛地用于数据通信，所以本节以它的用法说明这种电路的特性。

2.5.1 模拟PSTN电路

当我们想用已有的模拟PSTN传输线发送数据时，则必须将源DTE输出电信号转换成PSTN能接受的形式。后者是为语音通信设计的，假定声音频率是在400～3400 Hz范围内的合成，如图2-17所示。

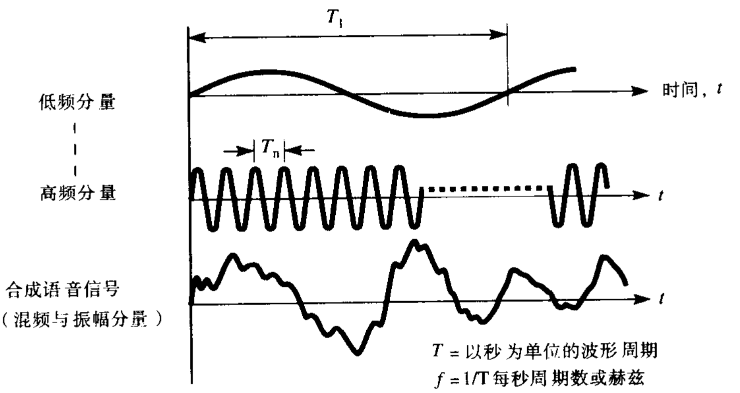


图2-17 语音波形频率分量

PSTN通过的信号频率范围（即它的带宽）是400～3400 Hz，或简单说带宽为3000 Hz。这意味着电话线不通过低频信号，例如，二进制数据流是由连续的“1”或“0”组成的串。由于以上理由，在电话线上不可以简单地使用两个电压电平。如果二进制数据流是全“1”或全“0”，则两个电平将得到零输出。换言之，我们必须在线路的发送端将二进制数据转换成

57 与语音信号兼容的形式。而在接收端，重新将信号转换成二进制形式。执行前一个操作的转换电路称为**调制器**，执行相反转换的电路称为**解调器**。由于数据链路每端通常既发送也接收，因此将两种电路合并在一起，称为**调制解调器**。

利用调制解调器通过PSTN发送数据，或者按照常规电话呼叫拨号，通过电话交换网，建立一条交换线路或者从PTT租用一条**专用线**（或**租用线**）发送数据。因为租用线能越过网络中常规的交换设备，建立一条固定线路或长期线路，仅对高利用率来说是经济上合算的。租用线的另一个优点，它的操作特性比短期交换电路质量要高，使用高信号速率（比特率）也更加灵活。现在考虑一些调制方法。

调制

将二进制信号转换成适用在PSTN上传输的形式，我们采用三种基本调制类型：调幅、调频和调相。因为发送二进制数据，只有两个信号电平，所以信号在两个电平之间进行转换（移位）。如同二进制数据信号在二进制1与0之间交替（键）。三种基本调制类型分别称为**幅移键控**（ASK），**频移键控**（FSK）与**相移键控**（PSK）。下面将分别讨论每种方案的一般原理。

幅移键控　ASK操作原理如图2-18(a)所示，波形组如图2-18(b)所示。单频信号**音调**的振幅是在两个电平之间转换，按发送的二进制数据信号的比特率确定转换的速率。单频信号音调称为**载波频率**，经过信道传输期间，载波信号有效地携带二进制数据信号。在PSTN可用频率范围内选择载波频率。发送二进制数据信号要求的带宽值由信号的比特率决定：比特率越高，要求带宽越大。实际上，不同调制方法，对发送二进制信号要求不同的带宽值，因此，需要定量地表示每种方法要求的带宽范围。

调制操作（ASK、FSK或PSK）数学上等价于用二进制数据信号乘载波信号。因为载波是

58 单频信号，假定振幅为1，则可用下面表达式表示：

$$v_c(t) = \cos\omega_c t$$

其中ω_c是以每秒弧度表示的载波频率，在2.2节讨论，我们可用振幅为1，基频为ω_0的傅立叶级数表示单极性的周期数据信号$v_d(t)$：

$$v_d(t) = \frac{1}{2} + \frac{2}{\pi}\{\cos\omega_0 t - \frac{1}{3}\cos 3\omega_0 t + \frac{1}{5}\cos 5\omega_0 t - \cdots\}$$

因此ASK信号可用下面的数学表达式表示：

$$v_{\mathrm{ASK}}(t) = v_c(t)\cdot v_d(t)$$

即

59
$$v_{\mathrm{ASK}}(t) = \frac{1}{2}\cos\omega_c t + \frac{2}{\pi}\{\cos\omega_c t\cdot\cos\omega_0 t - \frac{1}{3}\cos\omega_c t\cdot\cos 3\omega_0 t + \cdots\}$$

由于

$$2\cos A\cos B = \cos(A-B) + \cos(A+B)$$

因此

$$v_{\mathrm{ASK}}(t) = \frac{1}{2}\cos\omega_c t + \frac{1}{\pi}\{\cos(\omega_c-\omega_0)t + \cos(\omega_c+\omega_0)t - \frac{1}{3}\cos(\omega_c-3\omega_0)t - \frac{1}{3}\cos(\omega_c+3\omega_0)t + \cdots\}$$

所以，我们可得到ASK信号等价于用载波信号的频率ω_c移动的原始数据信号，但对于数据信号的基频ω_0，ASK信号有两个频率分量$(\omega_c-\omega_0)$与$(\omega_c+\omega_0)$。而对于数据信号的谐波，ASK信号有两个谐波频率分量$(\omega_c-3\omega_0)$与$(\omega_c+3\omega_0)$。ASK信号的所有频率分量都是等间距地分布在载波频率的每一边，称为**边带**。ASK信号的带宽如图2-18(c)所示。

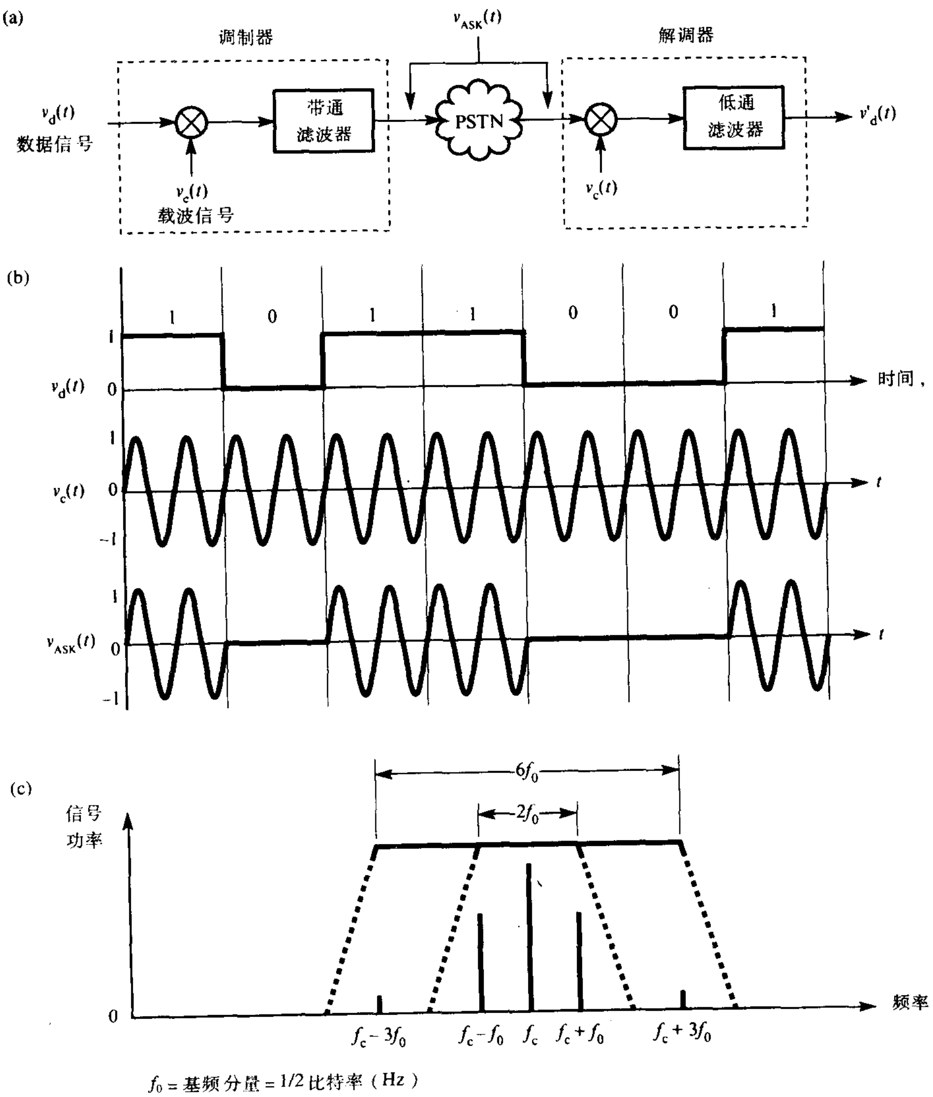


图2-18 幅移键控

(a) 电路结构 (b) 波形组 (c) 带宽选择

由前面带宽的讨论，知道信道的带宽越大，接收信号越接近发送信号。然而，通常若信道带宽足够使二进制序列101010…的数字信号的基频分量通过，则该信道即可满意的工作，其原因是所有其他二进制数字信号对信号带宽的要求都比它小。这个二进制数字信号的基频分量f_0（以Hz为单位）等于它的比特率的一半（以bps为单位）。所以，ASK信号要求的信道最小带宽等于它的比特率，其值为$2f_0$，为了能接收第三个谐波分量，要求带宽为三倍以赫兹为单位的比特率$6f_0$。

由图2-18(c)可得到关于ASK信号，即使没有信息信号，也就是没有f_0, $3f_0$等等，载波信号频率仍出现在接收信号中。由2.2节奈奎斯特公式，知道二进制信号在信道上可达到的最大数据速率是带宽的两倍。反过来，设想一个信道带宽为$2f_0$，而奈奎斯特速率刚好是$4f_0$。这样考虑的理由是两个主要边带用于确定最小的要求带宽，如图2-18(c)看到，每一个边带都包含有要求信号f_0。为了更有效地利用带宽，使用电路结构图中的**带通滤波器**，让频带中除分量$f_c+(f_c+f_0)$通过外，其他都限制通过，因此下边带(f_c-f_0)也移去，这就把要求带宽降低到f_0，因此产生奈奎斯特速率。但是，从图2-18(c)可见，主边带信号的功率是载波信号功率的一半。功率减弱反过来也降低信噪比，因此引起位差错率增加。

为了从接收信号恢复发送数据信号，我们通过**解调器电路**来完成。此时，接收信号再乘同一载波信号。这产生接收信号的两种形式：一种以频率$2f_c$ (f_c+f_c)为中心，另一种以0频率(f_c-f_c)
60 为中心。这两种形式包含边带内要求的信息，但我们选择后一种，让接收信号通过**低通滤波器**。这个滤波器仅允许0 ~f_0的频率通过，如果接受第三个谐波，则仅对从0 ~ $3f_0$的频率通过。所以，低通滤波器的输出是发送限制数据信号的带宽版本。

虽然ASK的实现相对简单，在早期低比特率的调制解调器中不采用。那时在PSTN中，所有长距离发送与交换系统都采用模拟信号，即所有原始语音与调制数据信号的发送与交换全部以原始模拟形式进行。例如，通过网络的不同路由的变化的传播环境，极容易引起不同程度的信号衰减，因此不能采用调幅方案。但是最近在网络上所有传送与交换已实施数字化，这就是说原始信号以模拟形式发送仅在客户前置设备到局部网接入端，例如局部交换机。在该端点信号转换成数字形式，从此它保持信号原始的特征。这导致通过PSTN建立电路在电子特征方面的重大改进。受此影响，现在ASK结合相移键控在更高比特率的调制解调器设计中采用。

实例2-7

假定使用ASK调制，并设(a)接收到序列101010…的基频分量；(b)除基频分量外，还加上第三个谐波频率分量。计算以下列比特率：300 bps、1200 bps与4800 bps发送时，要求的信道带宽。在公用电话交换网（PSTN）上说明结果。

解：

比特率	300 bps	1200 bps	4800 bps
基频分量	150 Hz	600 Hz	2400 Hz
第三个谐波分量	450 Hz	1800 Hz	7200 Hz
仅有基频分量的带宽	300 Hz	1200 Hz	4800 Hz
有基频与第三个谐波分量的带宽	900 Hz	3600 Hz	14400 Hz

PSTN可用带宽是3000 Hz，因此，只有300 bps速率能接收第三个谐波。1200 bps速率仅能接收基频分量，而4800 bps速率不能用ASK发送数据。

频移键控　FSK调制方法早期多用于低速调制解调器中，它的操作原理如图2-19(a)所示。
61 在FSK中，为了不采用振幅变化，而使用两个固定振幅载波信号，一个用来表示二进制“0”，另一个用来表示二进制“1”，两个载波频率之差称为**频移**。如图2-19(b)所示，FSK调制操作等价于两个独立的ASK调制器输出的和：一个用原始数据信号调制载波，另一个用原始数据的反码调制另一个载波信号。我们用如下数学表达式推出FSK带宽要求：

$$V_{\mathrm{FSK}}(t)=\cos\omega_1 t\cdot v_d(t)+\cos\omega_2 t\cdot v_d^{'}(t)$$

其中ω_1与ω_2是两个载频信号的频率，而$v_d'(t)$是原始数据信号$v_d(t)$的反码，数学上，$v_d'(t)=1-v_d(t)$。
62 如果假定周期数据信号的基频为ω_0，则

$$v_{FSK}(t)=\cos\omega_1 t\{\frac{1}{2}+\frac{2}{\pi}(\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\cdots)\}$$
$$+\cos\omega_2 t\{\frac{1}{2}-\frac{2}{\pi}(\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\cdots)\}$$

即

$$v_{FSK}(t)=\frac{1}{2}\cos\omega_1 t+\frac{1}{\pi}\{\cos(\omega_1-\omega_0)t+\cos(\omega_1+\omega_0)t$$
$$-\frac{1}{3}\cos(\omega_1-3\omega_0)t-\frac{1}{3}\cos(\omega_1+3\omega_0)t+\cdots\}$$
$$+\frac{1}{2}\cos\omega_2 t+\frac{1}{\pi}\{\cos(\omega_2-\omega_0)t+\cos(\omega_2+\omega_0)t$$
$$-\frac{1}{3}\cos(\omega_2-3\omega_0)t-\frac{1}{3}\cos(\omega_2+3\omega_0)t+\cdots\}$$

我们得出，FSK要求的带宽简单地是两个频率分别为ω_1和ω_2的独立的ASK已调载波之和。所以，一个FSK信号的带宽要求如图2-19(b)所示。

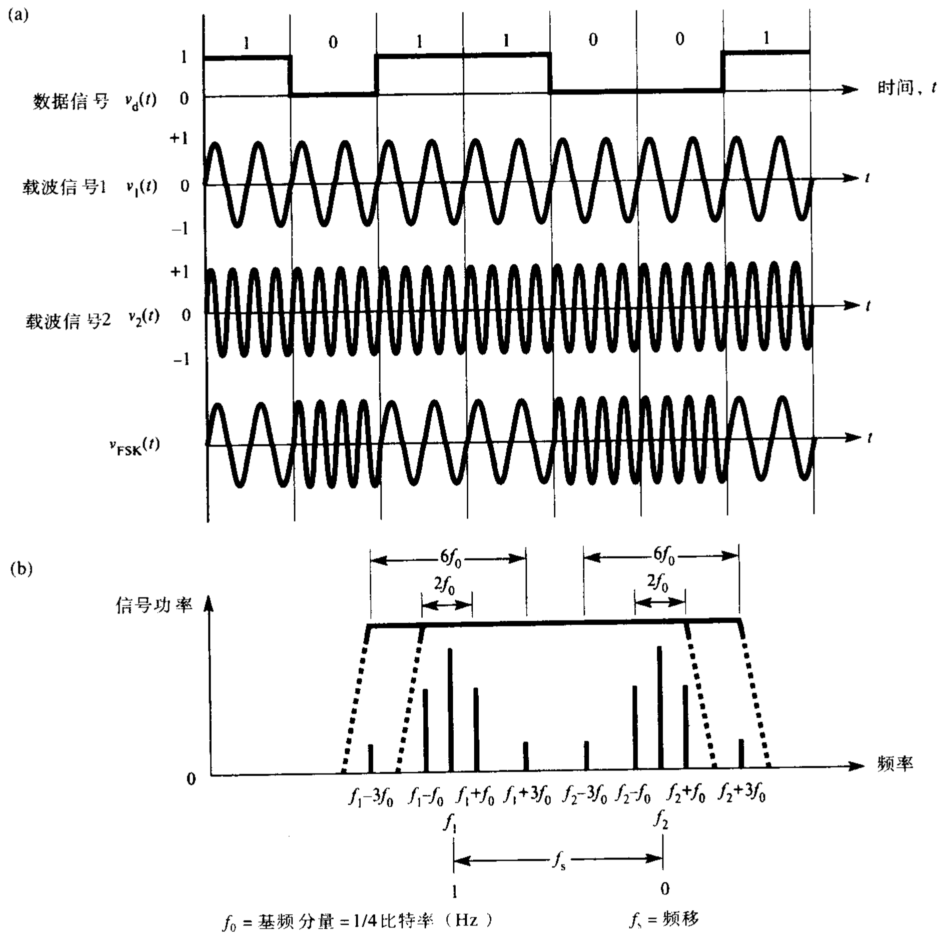


图2-19 频移键控

(a) 操作原理 (b) 带宽选择

按2.2.2节说明，二进制序列101010...信号产生频率最高的信号。因为FSK把每一个二进制“1”信号与“0”信号调制成一个载波，每个载波要求最小带宽是比特率的一半，即每个载波的最高基频分量f_0是ASK要求的最高比特率的一半。因此，如我们假设（最高）基频分量刚好被接收到，FSK要求的总带宽是$4f_0$加频移f_s。由于f_0是ASK比特率的一半，所以总的要求带宽是ASK要求带宽加频移。同样的，如果一对第三个谐波被接收到，要求带宽是$6f_0$加频移。

例如，如果最大比特率是600 bps，则每个载波的最大比特率是300 bps，并具有最大基频分量150 Hz。所以频谱包含两个载波每侧的150 Hz的主边带。因此，假如选择两个载波之间频移为400 Hz，则两个载波的主边带提供100 Hz的间隙。这就是说，总的带宽要求是900 Hz左右。显然，通过模拟PSTN连接的信道带宽是3000 Hz，则从一个PSTN连接，可获得两个这样的信道（每个用于一个方向传输）。

图2-20作为实例说明两种FSK调制解调器的频率分配，调制解调器在两个DTE间提供全双工（同时，两个方向）300 bps链路。一个用于EIA规定的Bell 103调制解调器，另一个是用于ITU-T规定的**V.21调制解调器**。注意：这样的调制解调器中，与每个载波相关的基频分量是
63 75 Hz，因此频移为200 Hz允许两个主边带之间有50 Hz间隙。

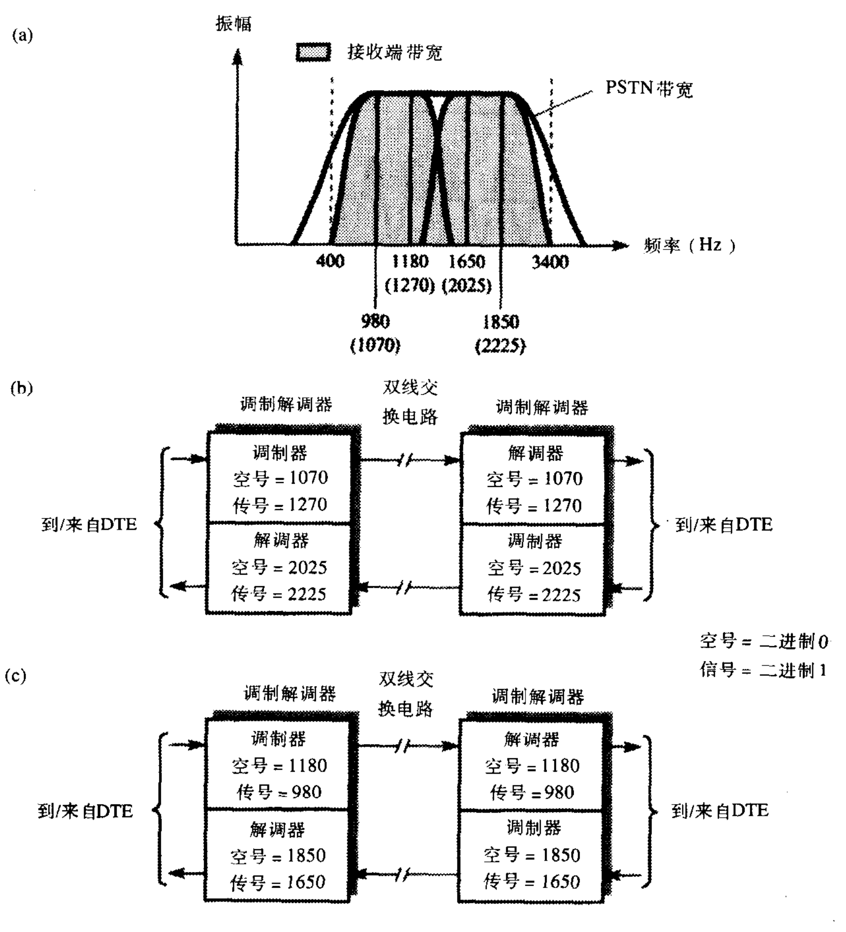


图2-20　全双工300 bps调制解调器

(a) 频谱　(b) EIA频率分配　(c) ITU-T频率分配(V.21)

相移键控 在PSK中，载波信号的频率与振幅保持不变，而按照发送数据流中的每一位移动载波的相位。操作方案的原理如图2-21(a)所示，可看到有两种PSK类型。第一种使用两个固定的载波信号，用180°的相位差表示0与1，由于一个信号是另一个信号的反转，所以称为**相位相干PSK**。这个方案的缺点是要求在接收方有一个基准载波信号对接收到信号相位进行比较。实际上，它比第二种**差分PSK**要求更加复杂的调制电路。在差分相移键控方案中，不管是否有二进制1或0信号在发送，相移在每位转换时发生。当前信号相移90°说明下一位是0，而相移270°说明下一位是1。因此，解调器电路仅需确定每次相移的大小而不是绝对值。 64

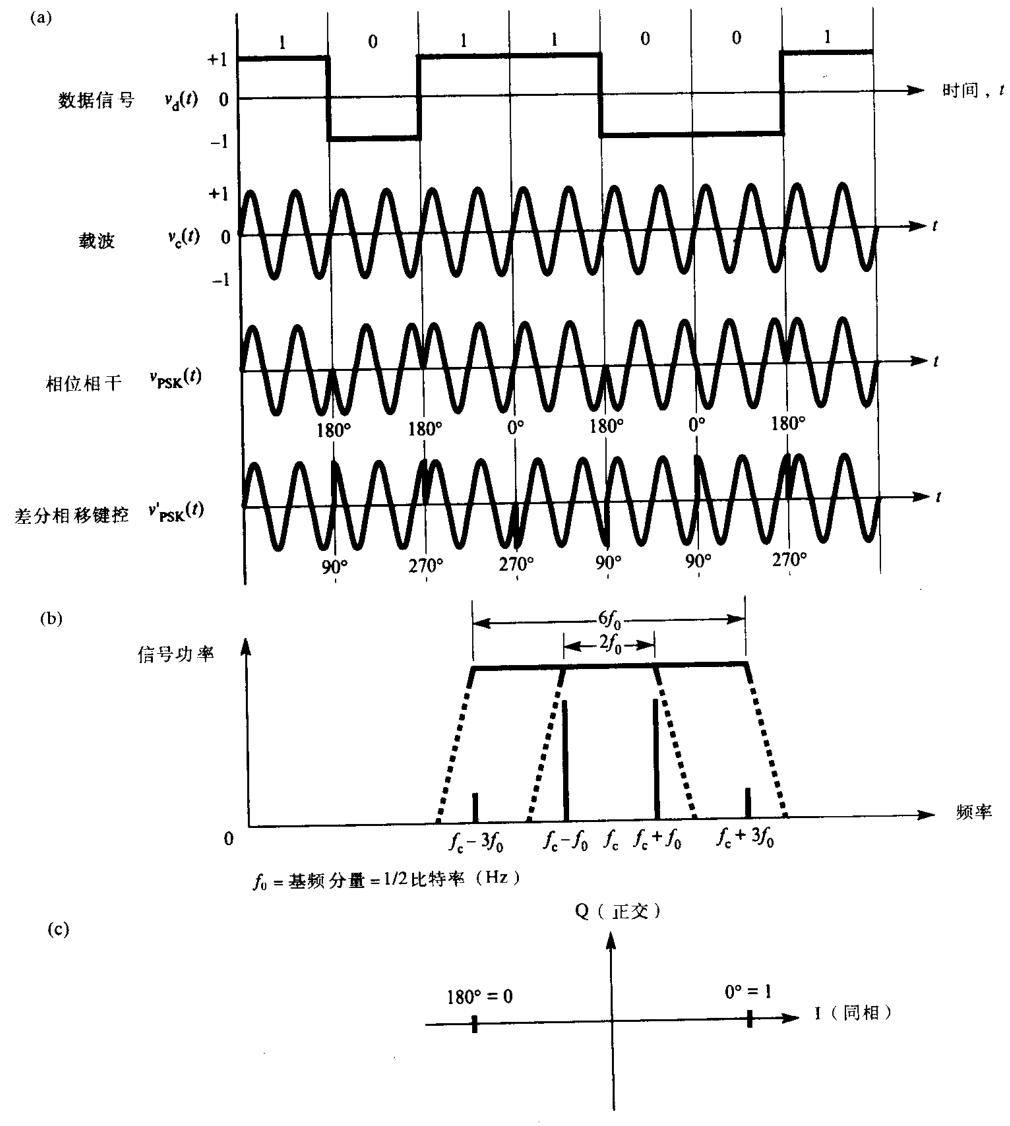


图2-21 相移键控

(a) 操作原理 (b) 带宽选择 (c) 相位图

我们可用二进制数据信号的双极性形式的数学表达式推出PSK调制要求的带宽，由于负信号电平导致载波信号相移180°。正如2.2节指出，振幅为1的双极性周期数字信号，基频为

ω_0，可用傅立叶级数表示：

65

$$v_{\mathrm{d}}(t)=\frac{4}{\pi}\left\{\cos\omega_0 t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t-\cdots\right\}$$

因此

$$\begin{aligned}v_{\mathrm{PSK}}&=\frac{4}{\pi}\left\{\cos\omega_{\mathrm{c}}t\cdot\cos\omega_0 t-\frac{1}{3}\cos\omega_{\mathrm{c}}t\cdot\cos 3\omega_0 t+\cdots\right\}\\&=\frac{2}{\pi}\left\{\cos(\omega_{\mathrm{c}}-\omega_0)t+\cos(\omega_{\mathrm{c}}+\omega_0)t\right.\\&\quad\left.-\frac{1}{3}\cos(\omega_{\mathrm{c}}-3\omega_0)t-\frac{1}{3}\cos(\omega_{\mathrm{c}}+3\omega_0)t+\cdots\right\}\end{aligned}$$

这说明，PSK的频谱是除去载波分量的ASK频谱。PSK信号的带宽如图2-21(b)所示。因此，如果假定只接收序列101010...的基频分量，则最小带宽要求是$2f_0$，等于比特率（以Hz为单位）。设有载波分量说明边带中（包括数据）存在较多能量，使PSK比ASK或FSK对噪声更适应。也限制发送信号从 f_C 到 f_C+f_0，即带宽为 f_0，则可得到奈奎斯特速率。再者，这种情况除了没有载波信号外，完全与ASK相同，因此所有接收的功率是在信息携带信号f_C+f_0中。

因为PSK采用单频载波，用不同相移表示每个二进制位，所以我们通常用**相位图**形式表示PSK，如图2-21(c)所示。相位图把正弦载波表示为一根线，称为**向量**，它的长度等于信号的振幅。并以恒定的频率（等于角频）ω逆时针绕轴旋转。二进制“1”表示与载波同相的向量，而二进制“0”表示与载波反相的向量。两个轴称为I（同相）轴与Q（正交）轴。

多级调制方法　如同2.5节开始时提到，目前所有数字传输与交换已渗透到PSTN中。从而，为了超过基本调制方案可得到的比特率需采用更复杂的调制方法，如信号单元多级电平，或基本调制方法中振幅与相位调制相结合。

到目前为止，在我们所讨论过的实例中，比特率都与信号单元速率相同，即等于每秒发送的信号单元的个数。但是，每个信号单元可有多于2个的值，通常有4值或8值，这就是说信号单元可表示为两位（4值）或三位（8值）的二进制信息。比特率是信号单元（波特）速率的两倍或三倍。例如，在PSK调制解调器中，采用四个不同的相位（0°、90°、180°、270°）代替刚才的两个不同相位。这能使每个相位的变化传递两位信息，有两种表示形式，如图2-22所示。由于使用四个相位，称为**正交相移键控**（QPSK）或**4-PSK**。

66 利用8相甚至16相变化可获取更高比特率，虽然减少相位差使得方案易于改进传输系统中引入的噪声与相位减损，但是，实际中可用多少相位是有限制的。因此，为了进一步提高比特率，通常如同相位变化一样，引入振幅变化。这种类型调制称为**正交幅度调制**（QAM）。实例如图2-22(c)中相位图。每个信号单元有16个电平，因此有4位，称为16-QAM，相位图也称为16点**星座图**。

由于刚才的说明，所有调制方案的健壮性由星座图中相邻相位的接近度决定。因此，关

67 于这个方案，由图2-22可见，使用8个相位而相邻相位的幅度是不同的。总而言之，这样使得在接收方易于区分，降低差错，但增加冗余的成本。因为对于相位来说，不是所有四个幅度都用到。所有这些调制方案，在调制之前，首先让比特流通过**扰频器**电路，转换成一个伪随机序列。其结果是减小序列中连续位处于相邻比特位置的概率。在接收方，解调过程之后，比特流通过一个相应**解扰器**电路，还原成原始顺序的比特流。**V.29调制解调器**采用这种调制类型，主要用于传真发送，比特率可达9600 bps。

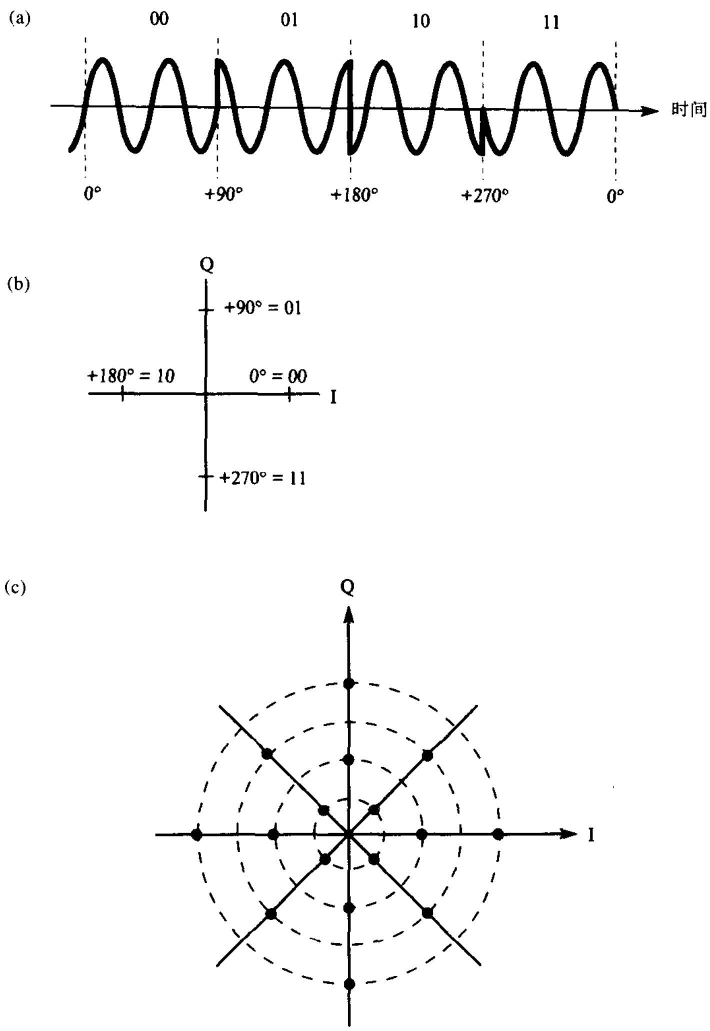


图2-22 另一种调制方法

(a) 4-PSK相位－时间波形 (b) 4-PSK相位图 (c) 16-QAM相位图

另一种冗余技术是使用所有32个振幅－相位选择方法，5位组的**记号**只有4位数据，而用**卷积编码器**生成第5位用于纠错目的。其操作原理在附录A中描述。然而，源数据流中的每个4位组用一种方法转换成5位，在接收方，甚至接收序列中存在位错误的情况下，也可以确定最合理4位数据。因为卷积编码器利用网格图（参看附录A）标识每个可能的5位组记号，所以这种方案称为**网格编码调制**（TCM）。这个方法用于V.32调制解调器中，比特率可达14 400 bps。**V.34/V-高速调制解调器**类型的调制解调器，现在有速率19 200，24 000及28 800 bps可用。在本章最后，给出各种调制解调器类型的概要。

2.5.2 数字租用线路

数字租用线路不仅在两个DTE之间提供直接的连接，而且还作为大多数专用数据（与语音）网的基础。我们将在后面的第8章中讨论各种专用网。租用线路适用于内部通信量极大的机构或企业。

目前大多数公用载波网的呼叫（语音与数据）信息在网络内部交换机之间以数字形式传

输。再者，数字方式操作正在稳步地延伸到各个用户领域。由于用户不用调制解调器就可以迅速地发送数据（关于语音的），所形成的网络称为**综合业务数字网**（ISDN）。我们将在第8章中讨论用ISDN传送数据。数字传输的使用使得现在可向许多公用载波运营商租用全数字线路，其速率从每秒数万比特到数百兆比特。

这样的线路是从常规的交换通信演变而来，因此必定与之共存，所以当使用这样的线路
68 时，一定要知道如何按照线路有效容量组织数据。

1. 语音数字化

由于我们需要以数字形式传送语音，产生了数字线路，正如2.5节表示，语音传输最大带宽被限制小于4KHz。把语音信号转换成数字形式，**奈奎斯特采样定理**表明，振幅采样的最小速率必须大于信号最高频率分量的两倍。因此，4 KHz语音信号转换成数字形式，采样速率为每秒8000次。一般的结构如图2-23(a)所示。图2-23(b)是更详细的说明。

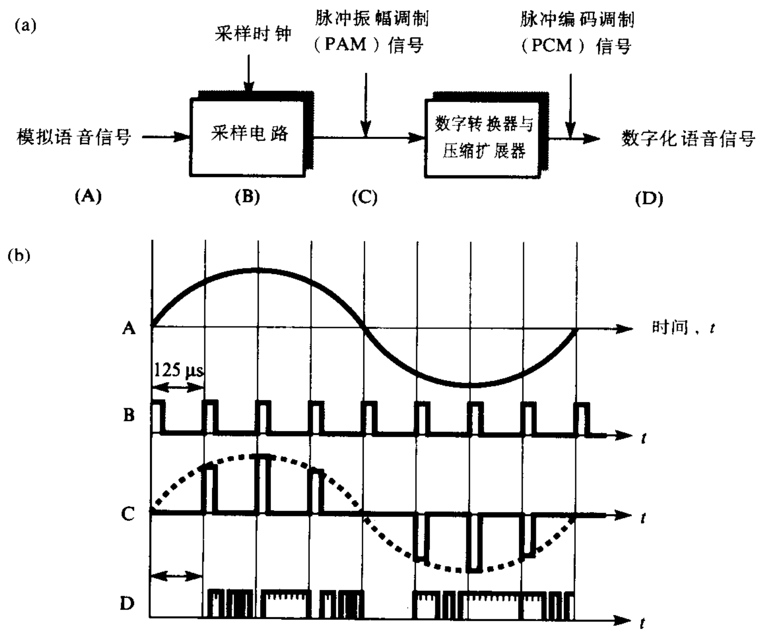


图2-23 数字化原理

(a) 编码器方案 (b) 编码器信号

虽然典型语音信号是由混合频率组成，但图2-23(b)表示单一（模拟）频率信号，正如我们所见，首先采样信号转变成采样脉冲流，采样脉冲的振幅等于原始模拟信号在采样时刻的振幅。因此，形成的信号称为**脉冲振幅调制**（PAM）信号。

由于PAM信号的振幅仍然在模拟信号振幅上变化，所以它还是模拟信号。然后通过把每个脉冲**量化**成为等价的二进制数据形式，转换成全数字形式。每个PAM信号量化为8位二进制数，其中有一位说明信号极性是正或负。这意味着从8个全0到8个全1有256个量化级。这样形成的数字信号称为**脉冲编码调制**（PCM）**信号**，而且具有比特率64kbps，即每秒采样8位组
69 8000次。该比特率是数字租用线路有效的传输容量最小单位。

正确表示PAM模拟采样信号的振幅需要无限个可能的二进制数字。这里只使用8位说明每个采样仅能用相应的有限个离散值的级表示，相邻级的差值称为**量化间隔**q，它决定量化过程的精确性。考察如图2-24(a)所示的量化实例。

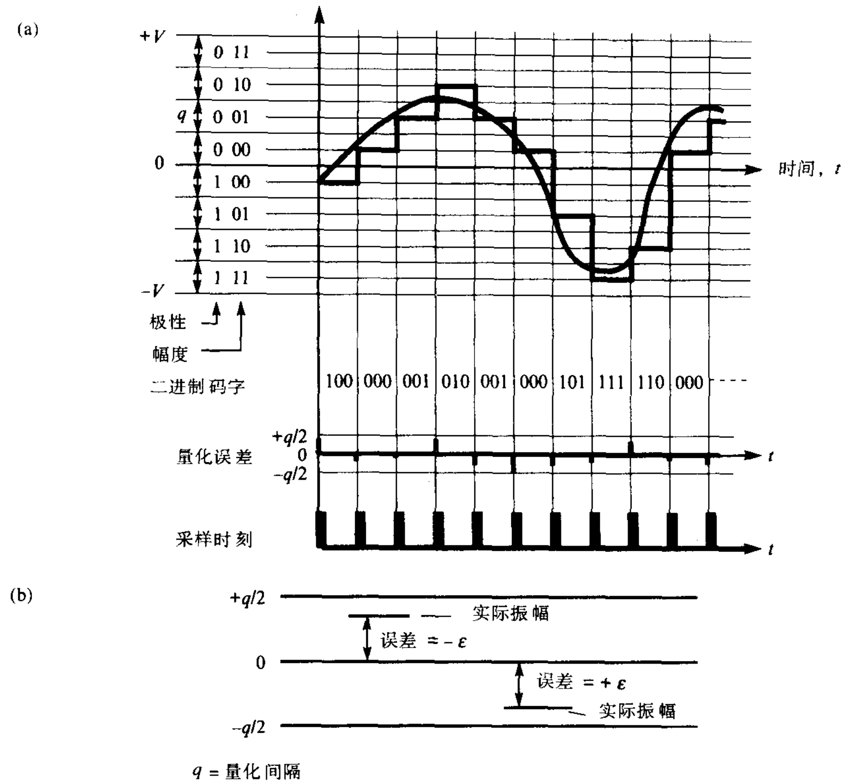


图2-24 量化过程

(a) 误差源 (b) 量化噪声范围

在这个实例中，我们用三位二进制数表示采样，包括极性（符号）位，它产生8个量化间隔。一般符号位为0，表示正极性，符号位为1，表示负极性。振幅位出模拟输入信号在每个采样时刻所在的特定量化间隔决定。

图2-24(a)表示信号在一个量化间隔内都用同一个二进制码字表示，每个码字对应量化间隔中心点的额定输入电压值。在量化间隔中它与实际输入电压值差可达正或负$q/2$。实际信号振幅与对应的采样振幅的差值，称为**量化误差**。它的标记值如图2-24(b)所示。语音信号量化值的误差随机地在各采样之间会不相同，因此也称**量化噪声**。 70

正如刚才表明，用线性（相等）量化间隔，小幅度信号比大幅度信号遭受的量化噪声大。然而关于噪声，人耳对安静环境里的噪声——语音信号（低幅度）比大声语音信号更敏感。为了减轻这样的影响，在实际的PCM系统中，量化间隔采用非线性（不相等）方法，通过改变与每个量化间隔相关的输入信号幅度范围，即由于输入信号的幅度增加，相应的码字代表大的信号范围，方案的原理如图2-25(a)所示。

如图2-25(a)所示，在输入信号采样与转换成数字形式之前，先通过**压缩器**。相似的，在目的地，通过**扩展器**电路对**数模转换器**（DAC）的输出执行相反的操作。因此组合操作称为**压扩**。两个电路输入、输出关系如图2-25(b)和(c)所示。图中(b)部分称为**压缩特性**，而(c)部分称为**扩展特性**。

用实例说明原理。仅取5位码字，由一个极性位，一个2位段码及一个2位量化码组成。在输入信号值域上形成段码与量化码。由图2-25可见，数字化操作由两个阶段完成。首先，压

缩阶段，输入信号用段码压缩，段码是由输入信号采样落在哪一段决定。然后形成的压缩
71 （模拟）信号传递到**模数转换器**（ADC），依次对压缩信号执行线性量化操作。

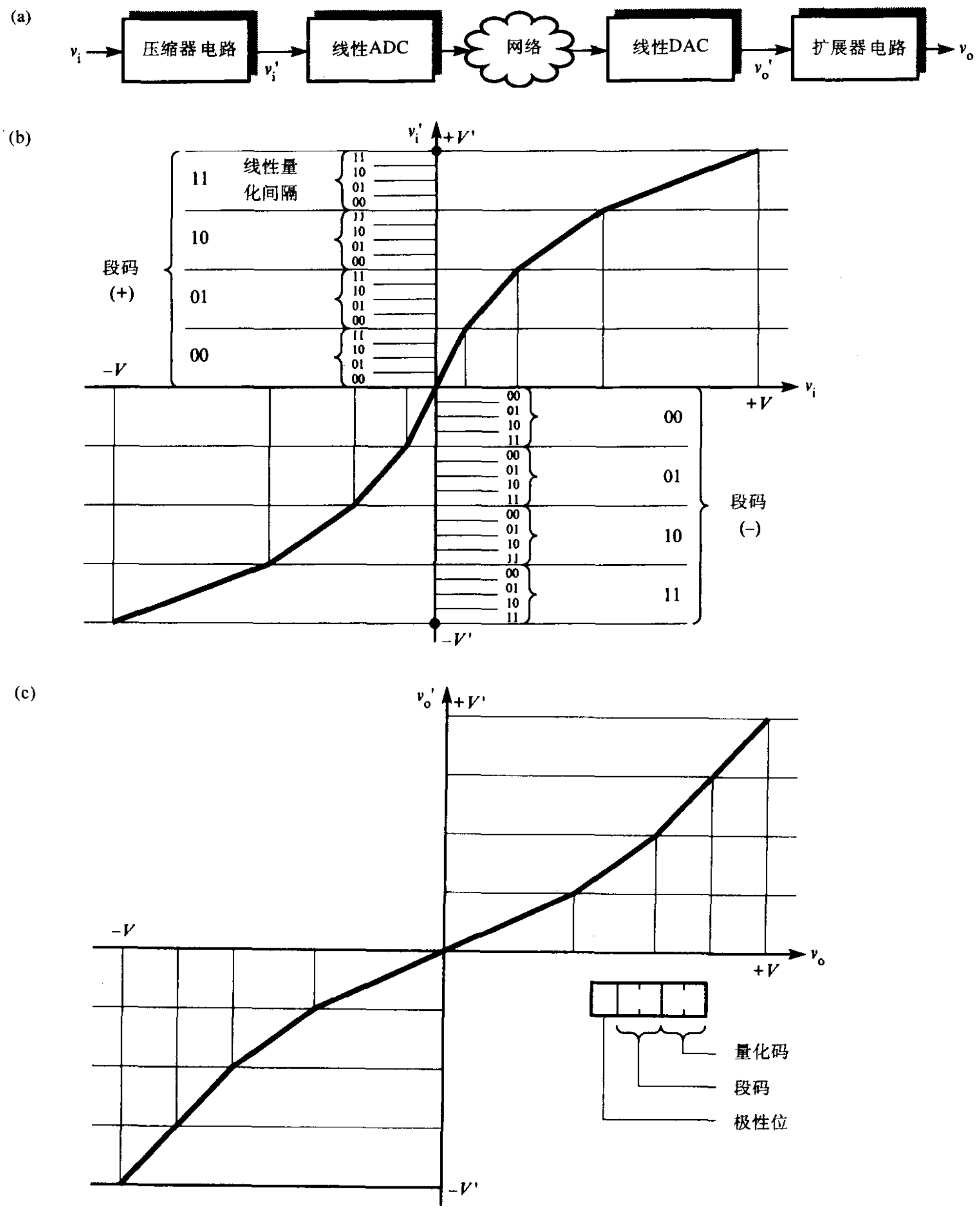


图2-25 压扩原理

(a) 线路结构 (b) 压缩方案 (c) 扩展特性

相似的，在目的地，每个接收到的码字，用两个阶段过程还原成原始模拟信号。首先把码字送到线性DAC，然后把（模拟）输出传递给扩展器电路，执行压缩器电路的逆操作。

实际上，虽然早期PCM编解码器（编码器/解码器电路）以这种方式操作，但现在，大多数编解码器数字化地执行压缩与扩展操作。采用的压缩特性有两种：μ定律特性（北美和日本

采用）与ITU-T推荐的A定律特性。当跨大陆使用不同标准的租用与交换线路时，需要特性的转换，这仅对语音通信是必须的，而数据信号不必。

2. 多路复用（Multiplexing）

用时分多路复用（TDM）技术，线路可以数字形式同时传送多个呼叫。在TDM中，我们已知道将许多个不同信息源的数字信号，每一个都赋予一个特定的时间间隙复接成一条高比特率的线路。由于每个模拟信号每秒采样8000次，每隔125μs产生一次8位采样。所以复接线路比特率是它传送的语音信道个数的函数。在北美和日本，24个语音信道复接成基群，而遵守ITU-T推荐标准的国家用30个信道。这就产生基群比特率分别为1.544 Mbps和2.048 Mbps。一般方案如图2-26(a)所示。 72

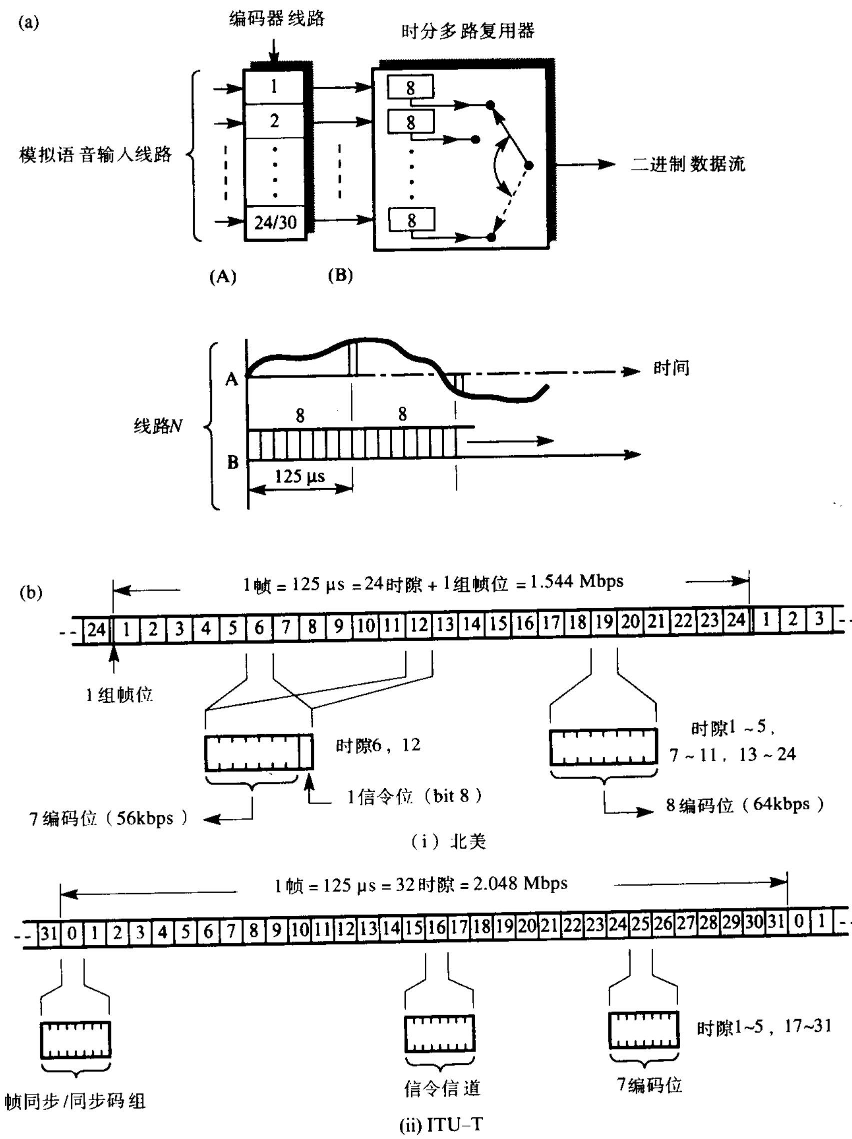


图2-26 多路复用结构

(a) TDM方案 (b) 帧结构

为了其他目的，我们还要包括一些附加位（或信道），如图2-26(b)所示，如指示每个帧的起始位——帧同步——和呼叫建立位（信令）。北美系统的帧同步使用单独一位作为帧起始，对于连续帧，每个帧在起始位0与1之间转换（交替）。信令信息位于时隙6与12的第一位中，在这些时隙中，其余7位为用户位。因此，基群的比特率是（$24\times 8+1$）位/125μs=1.544 Mbps，
73 这种线路称为**DS1**或**T1线路**。

在ITU-T推荐系统中，时隙0用作帧同步，也称为**帧同步**。因为它允许接收方中断每个帧的时隙以对齐边界。信令信息置于时隙16中，产生基群比特率32×8位/125μs=2.048 Mbps。这种线路称为**E1线路**。在T1与E1两个系统中，低比特率的线路称为**分路T1/E1**。

若干个基群复接可获得更高比特率合并线路。对这两个系统，表2-1给出线路名称及其比特率。更高比特率线路要求增加建帧与控制位，例如，$4\times 2.048=8.192$，因此0.256 Mbps是用于控制的功能。各种各样比特率分别缩写为1.5，3，6，44，274与2，8，34，140，565。

表2-1　北美与ITU-T数字传输系统的多路复用结构

	线路	比特率（Mbps）	语音/数据信道
北美	DS1	1.544	24
	DS1C	3.152	48
	DS2	6.312	96
	DS3	44.736	672
	DS4E	139.264	1920
	DS4	274.176	4032
ITU-T	E1	2.048	30
	E2	8.448	120
	E3	34.368	480
	E4	139.264	1920
	E5	565.148	7680

虽然，可直接产生高阶多路复接线路，但当为用户提供高阶多路复接线路时，会产生一些复杂的情况。最早，数字传输系统以递增方法引入，如同早期模拟系统升级一样。这项工作是用独立的定时源来产生TDM数据流。因此，当组合两个或多个低阶复接数据流时，由于每个数据流时钟信号的微小差异引起每个数据流定时略有不同，我们必须要有校正的办法。为此，我们用输出（复接）比特率略大于组合输入比特率之和，在不丢失有效信息的情况下，在数据流中填充称为**调节位**的空闲位改变每个数据流的速率，使之为复接器输出的规定的速率保持一致，以便在合路上传输。这样的高阶复接称为**准同步**（含义接近同步）**复用**，有时也称**异步多路复用**。有效高阶多路复用速率称为**准同步数字系列**（PDH）。

74 虽然，系列中每一级调节位本身不存在问题，但它的出现说明，我们不能正确识别高阶数据流中低阶复接的起始位。调节位的作用，最好通过一个典型操作说明。假定在不同的城镇有三个交换中心/交换机，用140 Mbps(PDH)中继线路互连，如图2-27(a)所示。客户与它们中某一方以2Mbps请求建立一个租用网，方案如图2-27(b)所示。由于在高阶比特流中不能识别低速信道。操作员在分配给客户之前必须将140 Mbps流信号完全分离为2 Mbps，然后再重新复合成140 Mbps向前传输。这种分接/复合的操作用**分接器与插入器**或**添加/丢弃多路复用器**（ADM）来完成。从图2-27(c)所见，符合这个相当简单请求的设备是很复杂的。

尽管图中没有表明，每个交换局/交换机必须相同地分配2 Mbps的租用线路，不必通过交换可以直接在客户间建立链路。所以，当用这种方法向客户提供租用线路时，对每一个客户，
75 必须有他们使用的线路和设备的详细记录。以便当报告有故障时，做出相应校正动作。实际

上，仅提供在PDH帧格式中的基本监控性能，这意味着客户通常必须向供应方提出发生故障的报警。

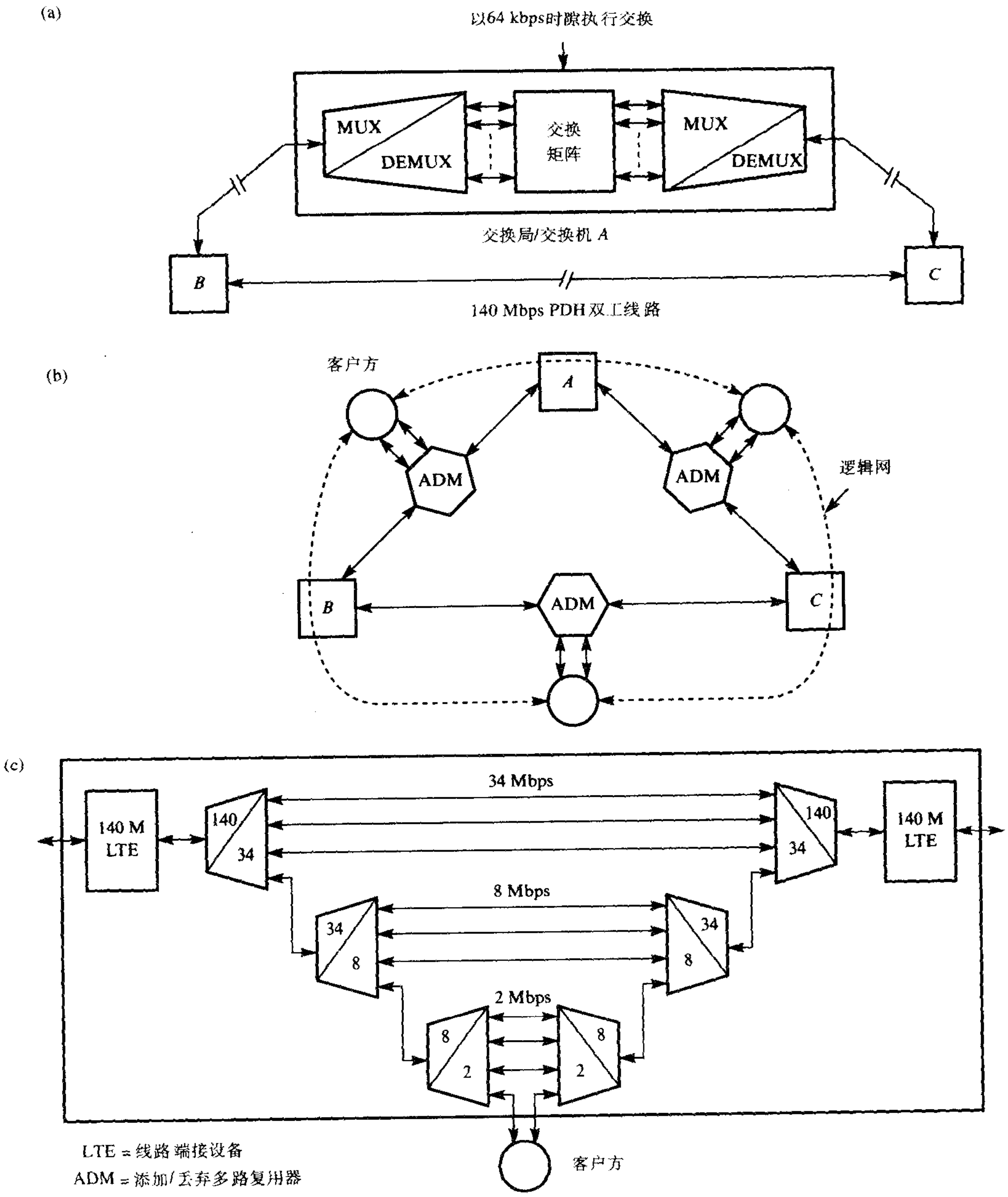


图2-27 应用PDH的专用网

(a) 原有网 (b) 重建网 (c) ADM原理

我们将在第10章看到，由于引入高速的新型网，高比特率线路租用需求迅速增长。运营商必须提供适应性更强的传输网，以满足不同部门要求的新业务。为克服PDH的局限性，引入一个完全新型传输系统，该系统称为**同步数字系列**（SDH）。因为要求高阶复接速率的大多

数租用线路可从SDH线路得到，现简要介绍SDH。

3. 同步数字系统

SDH最初是由美国贝尔中心（Bellcore）在**同步光纤网络**（SONET）项目下开发成的，所有设备与一个主时钟同步，基本传输速率为155.52 Mbps，简记为155 Mbps，称为**同步传输模块等级1**信号或简称为**STM-1**，还定义了更高速率的，如STM-4为622 Mbps，STM-16为2.4 GMbps。在SONET系列中，**同步传输信号**（STS）或**光纤信号**（OC）用于定义STM信号的等价物。SONET系列中，最低速率是51.84 Mbps，构成第一级信号STS-1/OC-1。一个STM-1信号通过多路复用三个这样的信号获得，因此等价于STS-3/OC-3信号。

如同PDH，STM-1信号由帧的重复组构成，以125 μs为周期重复，每一帧的信息内容可用
76 于携带多个1.5/2/6/34/45或140 Mbps PDH数据流。

每一个数据流用不同容器携带，它也包含用于允许实际速率变化的额加填充位。外加某些控制信息，称为**通路开销**，如网络管理在端到端的基础上监测相关容器的BER，容器外加通路开销构成**虚容器**（VC），而STM-1帧可包含多个类型相同或不同的VC。可选复用实例如图2-28所示。注意：最低级容器的第一位数字（同时VC）是1，第2位数字是1代表1.5 Mbps PDH信号，是2代表2 Mbps。

SONET	SDH	比特率（Mbps）
STS-1/OC-1		51.84
STS-3/OC-3	STM-1	155.52
STS-9/OC-9		466.56
STS-12/OC-12	STM-4	622.08
STS-18/OC-18		933.12
STS-24/OC-24		1244.16
STS-36/OC-36		1866.24
STS-48/OC-48	STM-16	2488.32

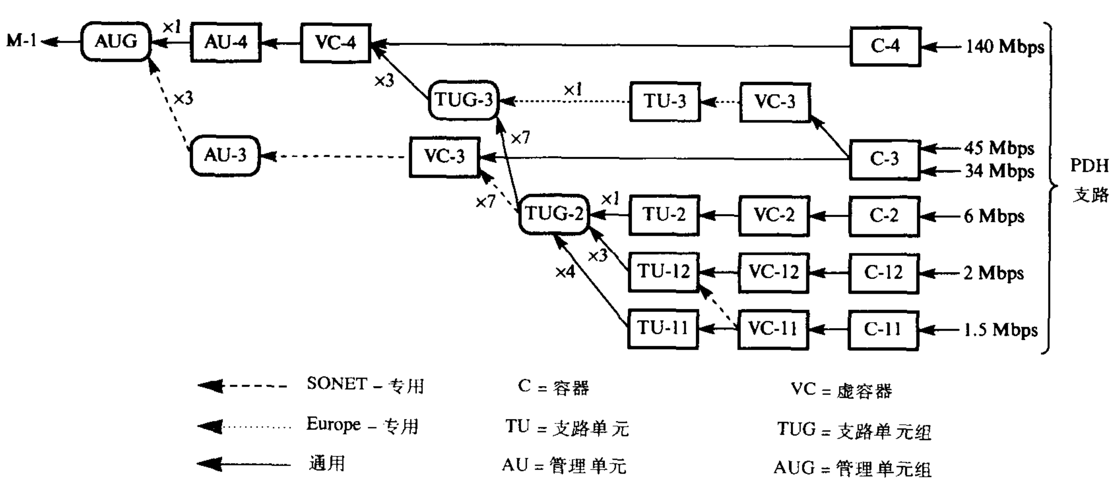


图2-28 SDH/SONET 多路复用层次结构和术语

高阶传输速率是多路复用多个STM-1(STS-3/OC-3)信号，例如，STM-16(STS-48/OC-48)信号是多路复用16个STM-1(STS-3/OC-3)信号或4个STM-4(STS-12/OC-12)信号产生的。为了对每个高阶信号提供必要的灵活性，除了在低阶STM帧的头部外加开销，一个指针用来说明

在高阶帧内低阶STM帧的位置。

因为每个帧包含每一个构成部分的管理信息，我们最好描述完全同步传输系统的SDH/SONET的结构。这些管理信息部分是**段**、**线路**和**通路**，它们的关系如图2-29(a)所示。

段是单根传输电缆，它的两端以**段端接设备**（STE）终结。STE的实例是中继器。在这个电缆段上，它重新生成传输的光/电信号。线路经过多个电缆段的伸延，以**线路端接设备**（LTE）终结。LTE的实例是多路复接器与交换结点。通路是经过整个传输系统端到端的传输路径。通路的每端以**通路端接设备**（PTE）终结。 77

STM-1(STS-3/OC-3)帧结构如图2-29(b)所示。每个帧由2430个字节/8位组组成，每125 μs重复1帧，因此，比特率为155.52 Mbps。一个帧有9个段，每段有270个字节，头部有9个字节，其余261个字节为**净荷**，头部字节称为**开销**。由于每个头部指定字节彼此相关，因此段以一个叠另一个的方式组成帧。 78

段开销字节用于指定段的管理。正如图2-29所示，某些字节是为了防止错误复制的。每个字节用法如下：

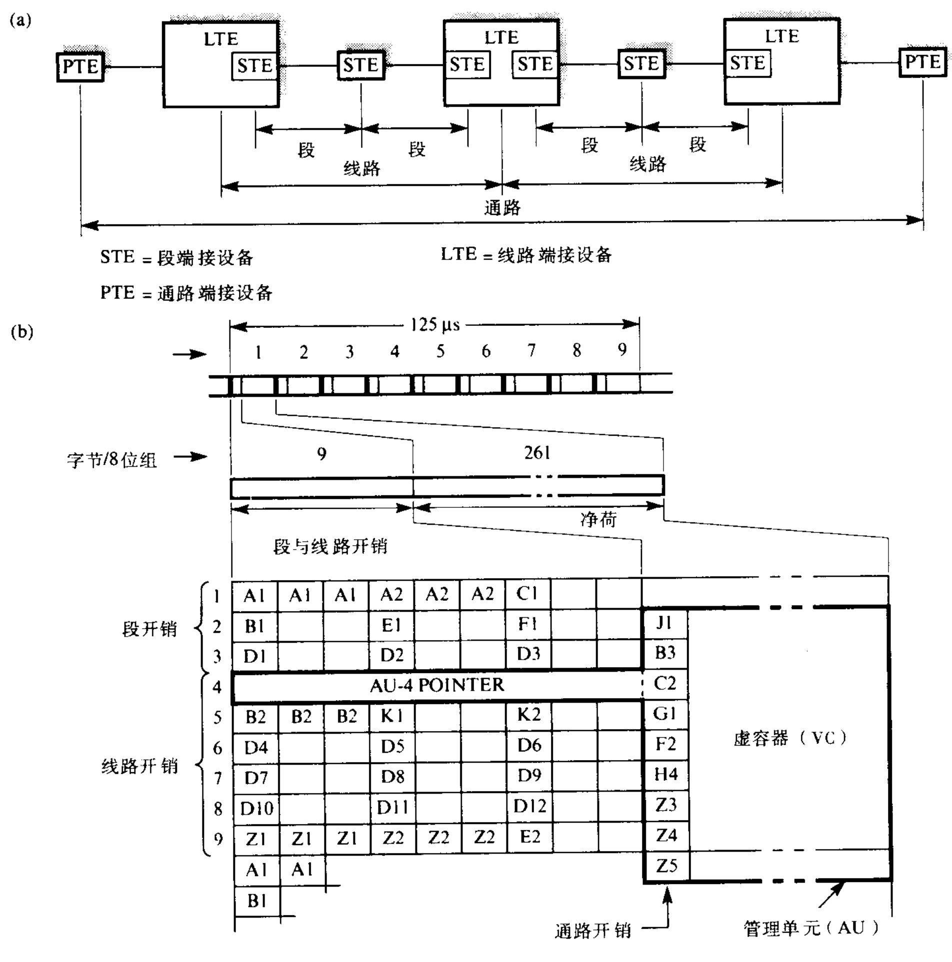


图2-29 SDH详述

(a) 管理实体 (b) 帧格式 (c) VC映射实例

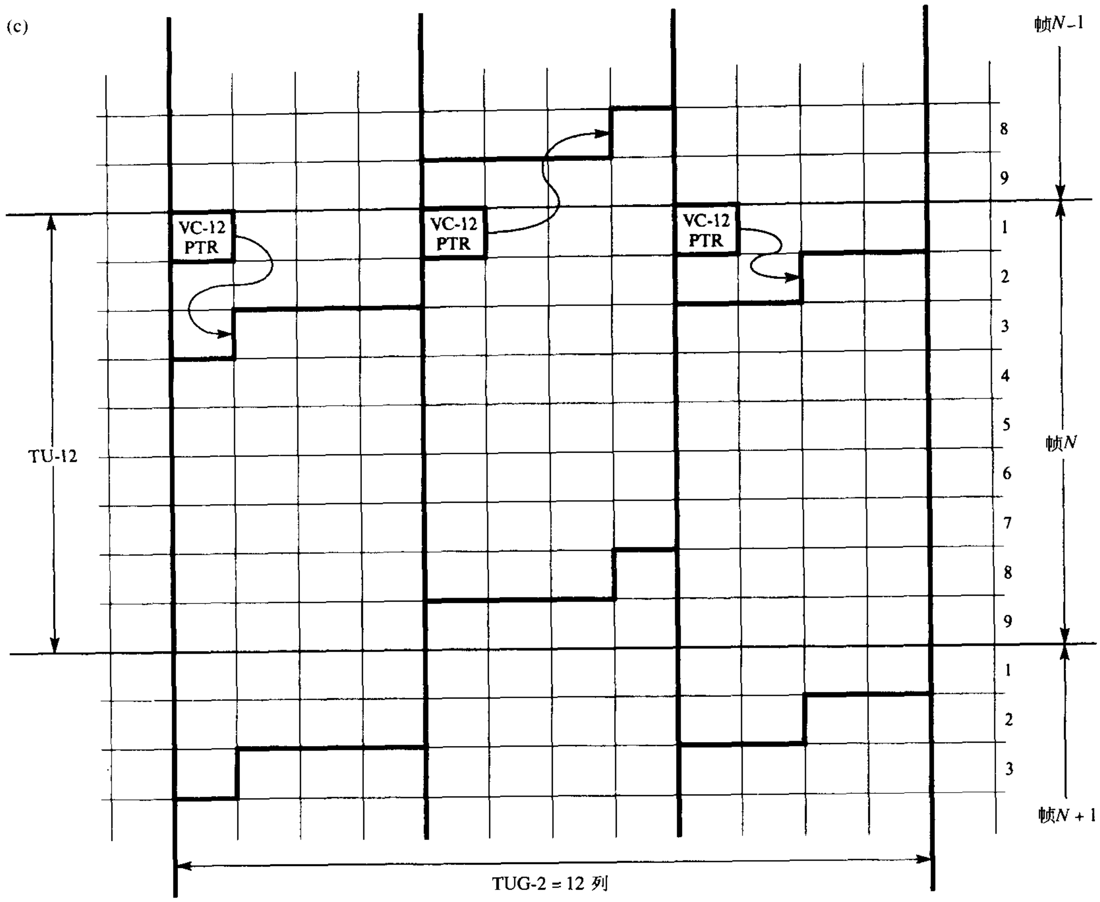


图2-29 （续）

A1-A2: 通常首先发送的字节，用于组帧（确定帧的起始位置）；

B1: 8位奇偶较验，用于监测段的误码；

C1: 在高阶（STM-n）帧中，标识一个指定STM-1帧；

D1-D3: 形成一个数据通信信道；用于与段有关的网络管理消息；

E1: 用于公务联络（orderwire channel），该信道是维护人员使用的语音信道；

79 F1: 用户信道，可用于客户场地的设备管理。

线路开销字节用于整个线路管理，每个字节作用如下：

B2: 8位奇偶校验，用于监测线路的误码；

D4-D12: 形成一个数据通信信道，用于与线路有关的网络管理消息；

E2: 与整个线路有关的公务联络；

K1-K2: 形成一个信令信道，用于整个线路的自动保护切换；

Z1-Z2: 保留为国内使用。

净荷字段的列可以各种方式指定传送低比特率信号。传送低阶PDH数据流称为**支路**，在每个容器中净荷分配在全部列中。每个容器有一个通路开销字节列，容器外加通路开销构成虚容器（VC）。

在图2-29中，给出每个字节的标识，它们的用法如下：

J1: 这个字节校验VC通路连接；

B3: 8位奇偶校验，用于监测通路误码率；

C2: 指明VC净荷组成；

G1: 用于接收方向发送方返回的接收信号状态；

F2: 向用户提供数据通信信道；

H4: 指明净荷是否是多帧的一部分；

Z3-Z5: 保留为国内使用。

在每个VC的头部有指针用于标明VC相对于每帧开始的位置。注意：如果容器包含PDH支路，则指针值可能由于可能的定时差在帧之间发生改变。不同VC的组合可用于填满较大VC中容纳的较小VC帧的净荷区。如果VC容纳一个低阶支路，则VC与它的指针称为**支路单元**（TU）；如果VC容纳若干个低阶支路，则称为**支路单元组**（TUG）。在STM-1帧中，最大VC称为**管理单元**（AU），如图2-29(b)所示，按照管理单元的开始，指针写入线路开销的第一个8位组的位置。

图2-29(c)的实例给出在一个TUG-2帧中如何容纳三个VC-12。每个VC-12由4个STM-1的4列净荷组成，因此TUG-2由12列组成。VC-12与它的指针组成一个TU-12。指针通常占第一个字节位置，但如果VC-12的时钟满足有关STM-1帧变化，则VC-12的位置允许滑动以适应指针变化值，使得它经常指向VC中第一个字节，VC-12容纳4 ×9=36字节。因此，由于一个VC-12由33个字节组成，32个字节用于E1帧加上1个指针字节，剩余字节用称作**固定填充**的字节去填足。 80

为了承载其他没有定义容器接受的信号，采用**链接**技术组合两个或多个TU。例如，5个TU-2链接承载32 Mbps信号。然后，4个这样的信号承载在一个VC-4中，代替三个已用的标准C-3容器。这个技术也适用ATM通信传送，后者称为信元，我们将在第10章中说明它的应用。

所有SDH设备都有称为**网络管理**（NM）**代理**的软件与开销字节通信信道，用于向中央网络管理站报告段、线路或通路的任何故障。它们也用来对后者下载命令改变每个STM-1帧的净荷字段的分配。例如，SDH ADM可以远程配置（重配置）以提供任何没有分路分解需求的混合要求带宽。一般原理如图2-30所示。冗余（备份）链路是用于每对SDH复接器之间并从远程网络管理站接收的命令实施服务。有关网络管理的协议与操作的详细说明在第13章给出。

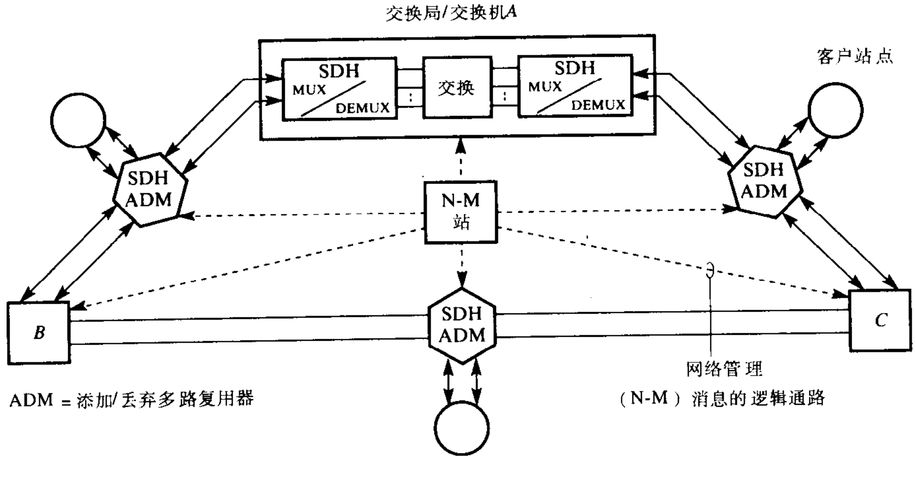


图2-30 用网络管理SDH提供服务

2.6 物理层接口标准

前面几节，我们已讨论了几种可选用的传输介质以及在两个DTE之间发送二进制数据流时，使用的电信号。然而，重要的是，各种引入的标准除限定所用的电信号形式外，还定义
81 了一系列附加信号，用于控制数据流通过相应接口的顺序和时间。综合起来，就说这些信号构成**物理层接口标准**。当然，本书不可能给出各种标准文件所定义的全部信号，我们讨论以前介绍的各种接口标准所用的几种附加控制信号。

2.6.1 EIA-232D

正如在2.3.1节看到，EIA-232D/V.24标准是最早定义的连接DTE与PTT提供（或许可）的调制解调器的接口标准。更一般，调制解调器称为**数据电路端接设备**（DCE）。说明两个通信DTE接口标准位置的框图如图2-31(a)所示。DTE与调制解调器之间所用的连接器如图2-31(b)所示，它是25针的连接器，如图2-31(b)所示。由ISO 2100标准定义，称为**DB25连接器**。同时表明，接口的全部信号集与其名称及针脚配置。

数据发送（TxD）与数据接收（RxD）线是分别用于DTE发送与接收数据。其他线分别具有控制和定时功能，以通过PSTN完成交换连接的建立和拆除，并执行选择的测试操作。第二（辅助）组线允许在一个接口上同时发生两个传送数据，所有线使用2.3.1节描述的V.28电信号。

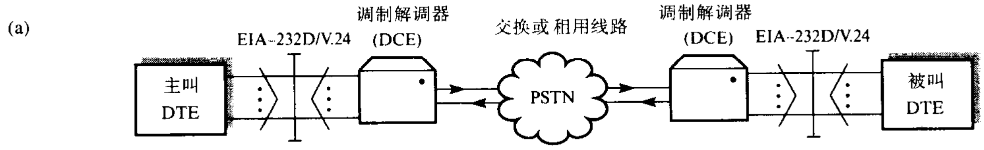
定时控制信号用于对应数据线上的数据发送（TxClk）与接收（RxClk）。在第3章中将看到，数据的发送用异步传送方式或者同步方式进行。在异步方式中，发送与接收时钟是由独立的时钟源产生的，直接送给相应的DTE针脚。在这种方式中，与调制解调器连接的只有发送与接收的数据线。但在同步方式中，用相应的时钟信号以同步方式发送数据与接收数据，时钟信号通常是由调制解调器产生的。后者称为**同步调制解调器**。当信号速率（波特率）小于数据比特率时，即有多个电平可用，则由调制解调器产生的发送与接收时钟按线路信号速率适当比例操作。

我们最好通过考察呼叫的建立或清除来了解各种控制线的功能与顺序。图2-32表示了如何首先建立一次连接（呼叫），在两台DTE之间实现一次半双工（双向交替）数据交换，然后清除呼叫。假定呼叫DTE是一个终端/个人计算机的用户，而它的调制解调器带有自动呼叫设备，被叫DTE也是一台计算机，而它的调制解调器带有自动应答设备。这些设备由V.25推荐标准规定。当DTE已就绪或接收到数据传送请求时，则它置数据终端就绪（DTR)线为1，本
82 地调制解调器通过设置DCE的就绪（DSR）线为1作为响应。

由呼叫DTE发送一个被叫DTE的调制解调器的电话号码建立连接，接收方从本地交换局/电话交换机收到铃声，被叫调制解调器置振铃指示（RI）线为1，同时被叫DTE置发送请求(RTS)为1作为响应。在响应过程中，被叫调制解调器向呼叫调制解调器发送一个载波信号（对应二进制1的数据音频），表示呼叫已被被叫DTE接受。经过短暂时延后，允许呼叫调制解
83~84 调器准备接收数据，以及被叫调制解调器置清除发送（CTS）线为1，通知被叫DTE可以开始发送数据。呼叫调制解调器检测到载波信号，置载波检测（CD）线为1。至此，建立连接并且开始数据传送阶段。

通常作为响应，被叫DTE（计算机）在建立连接的链路上发送一个简短的邀请发送消息。当消息发送完，它通过设置RTS线为0准备接收呼叫DTE的响应。这时被叫调制解调器停止发送载波信号，并置CTS线为0。在呼叫方，呼叫调制解调器检测到无载波信号，便将载频检测

（CD）线置0作为响应。为了发送它的响应信息，呼叫DTE（PC）置RTS线为1，接收来自调制解调器的CTS信号后，用户就可发送消息响应过程重复。作为两个DTE之间交换信息。最后，当整个传送完成之后，呼叫清除。这是通过依次设置两台DTE的RTS为0引起两个调制解调器关闭载波信号完成的。两个调制解调器检测到，并置它们的CD线为0。然后两个DTE置它们的DTR线为0，它们的调制解调器置DSR线为0作为响应，因此呼叫结束。然后，一般被叫DTE（计算机）在短暂时延后设置它的DTR线为1，准备接受新的呼叫。



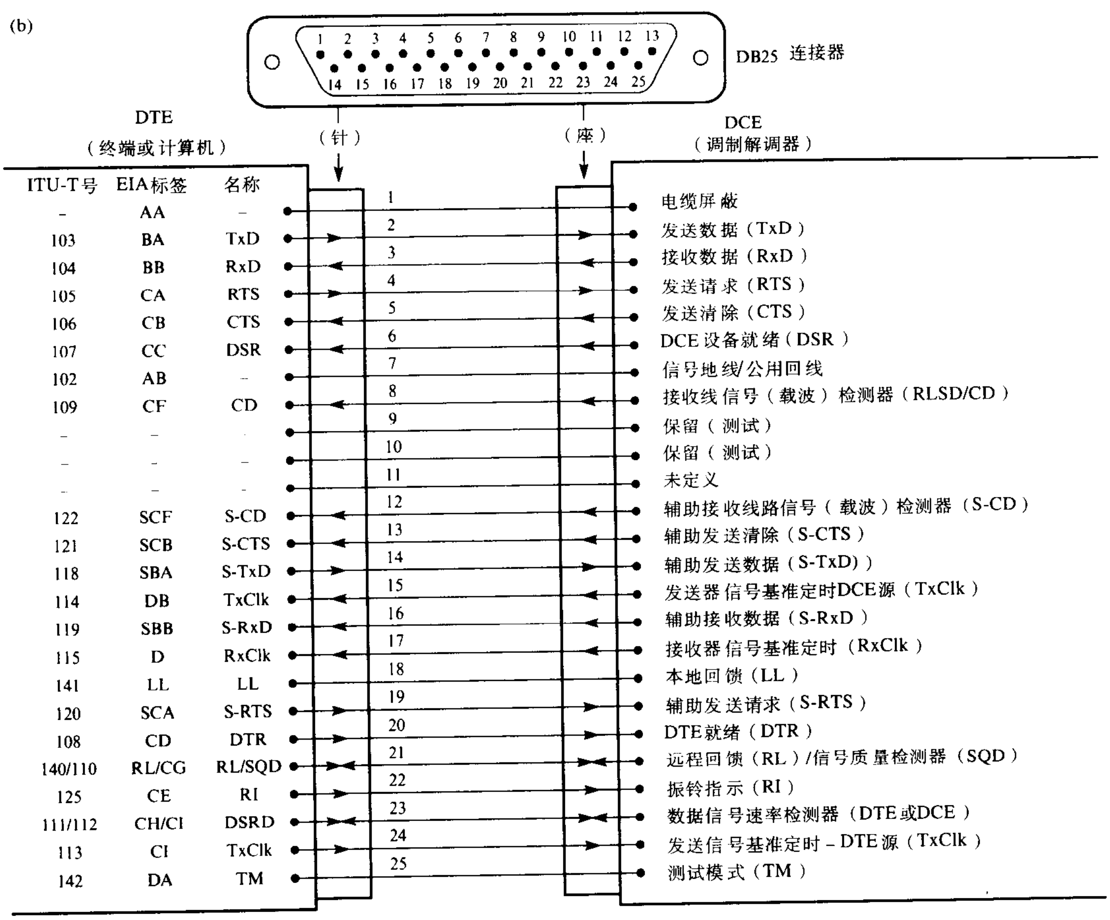


图2-31　EIA-232D/V.24标准接口

(a) 接口功能　(b) 插座、针脚与信号定义

我们已描述了半双工交换连接的用法，阐述了标准中控制线的含义和用法。然而，实际

上，在半双工模式中，从接收方式到发送方式的转换需要一定的时间，称为**转向时间**，它是重要的。甚至当要求半双工工作时，只要可能，最好用**双工方式**操作。在双工方式中，两个RTS线永远为1，而两个调制解调器CTS线置1，并保持向远程调制解调器发送载波信号。

当两个DTE还在通信时发生故障，判断故障发生的原因是来自本地调制解调器、远程调制解调器、通信线路还是远端DTE，时常是困难的。为了识别故障的原因，接口有三个控制线：本地与远程回馈（LL与RL）及测试模式（TM）。它们的功能如图2-33所示。(a)是用本地回馈测试，而(b)是用远程回馈测试。

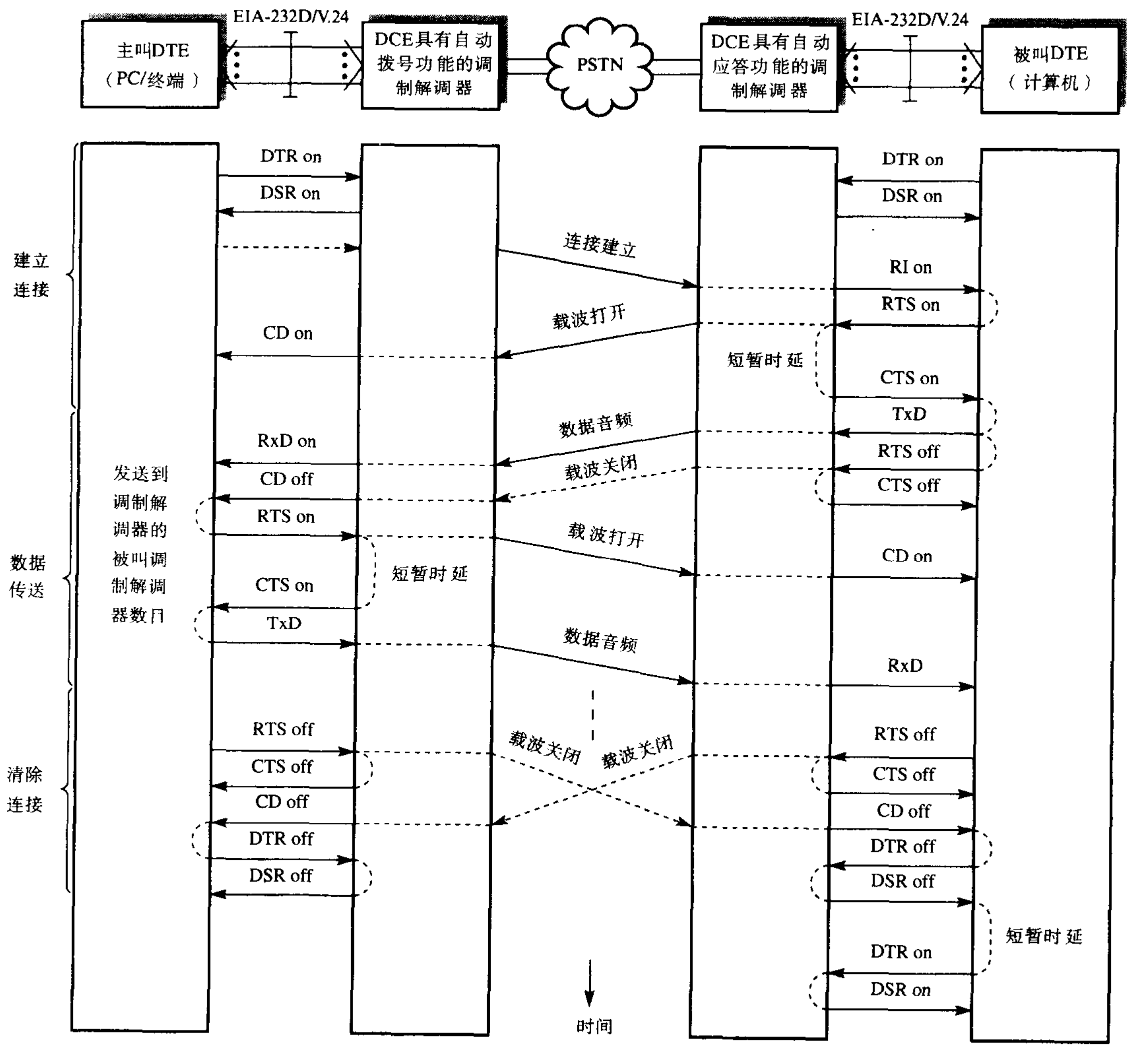


图2-32　EIA-232D/V.24连接建立、双向交替数据传送与清除连接的时序

DCE（调制解调器）常在准备发送或接收数据时设置它的SDR线为1。为了在本地调制解调器上执行测试，DTE置LL线为1作为响应，调制解调器将调制电路输出内部回送到解调电路输入，然后置TM线为1，它在TxD线发送一个已知测试（数据）模式，进行DTE检测，同时从它的RxD端读数据。如果接收数据与测试数据相同，则DTE认为本地调制解调器正常工作。如果不一样，或者一点信号也没有，则认为本地调制解调器有故障。

如果认为本地调制解调器工作正常，则此时DTE设置RL控制线为1继续测试远程调制解调器。检测出RL控制线为1，本地调制解调器向远程调制解调器发送一个预先规定命令，依次，如所示执行远程回馈。然后远程调制解调器置它的TM线为1，通知远程DTE,它有一个测试——因此不发数据——返回一个确认命令到调制解调器启动测试。本地调制解调器接收到这个信息，置它的TM线为1，DTE开始发送测试数据模式进行检测。如果正确地接收到这个数据，则认为两个调制解调器工作正常，故障是在远程DTE。另一种情况，如果接收数据是受损的，则认为远程调制解调器有故障。或者，根本没有接收信号，则认为PSTN线路有故障。 85

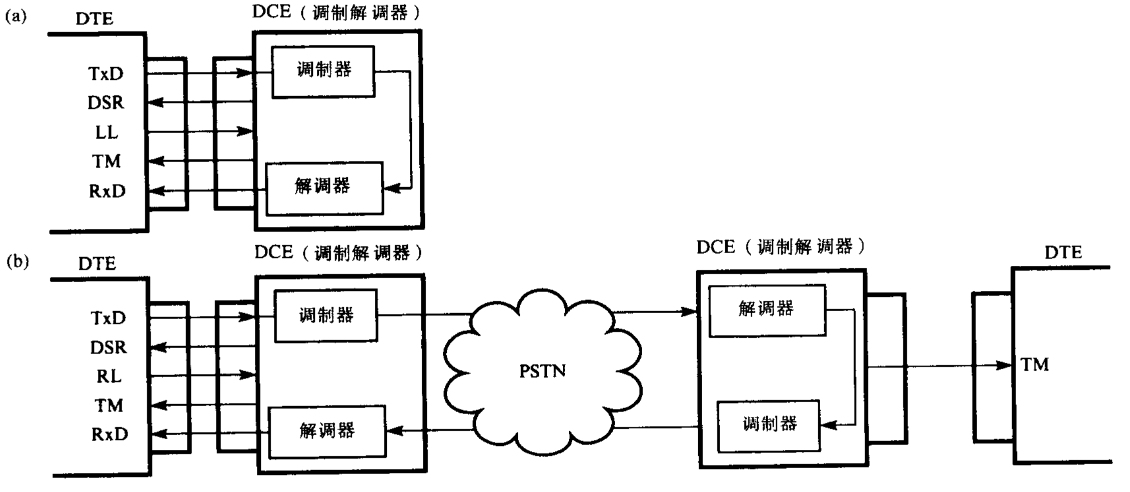


图2-33 回馈测试

(a) 本地 (b) 远程

空调制解调器

从图2-34所示信号分配可以看出，调制解调器对终端与计算机提供相同的功能，终端和计算机在同样的线路上发送与接收数据。然而，原先规定的EIA-232D/V.24标准也被用作面向字符的外围设备（终端、打印机等等）与计算机连接的标准接口。因此，在使用时，必须确定哪一种设备（外围设备或计算机）仿效调制解调器，因为这两种设备不能在同一条线路上发送与接收数据，当连接两台计算机的串口时，用法相同。

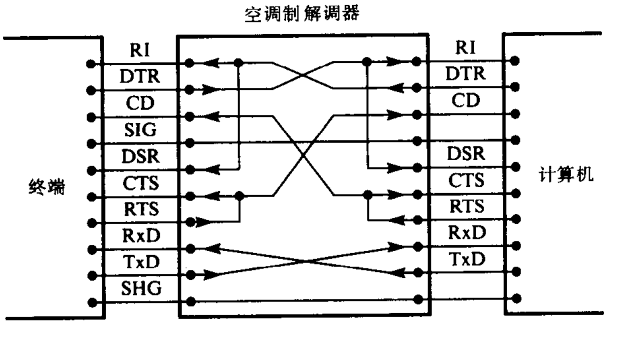


图2-34 空调制解调器连接

有三种可能的选择：

1）终端仿效调制解调器，并相应地采用合适线路定义；

2）计算机仿效调制解调器；

86 3）终端与计算机两者保持不变而变更互连线。

前两种选择的缺点是终端或计算机不能直接与调制解调器一起使用，通常采用的方法是将计算机EIA-232D端口连接成仿效调制解调器，这样不需要改变终端就可以与计算机直接相连。第三种选择被广泛采用，但要使用一个**空调制解调器**（或者**转换盒**），插在终端与计算机之间，对互连线进行必要的更改，如图2-34所示。

如图所示，除掉发送数据线与接收数据线相反连接外，一些控制线也要相反连接。例如，因为计算机与终端通常是在全双工模式下操作，所以两端都把RTS与CTS线连在一起，然后将此信号线与另一端设备的CD输入线相连。同样，每一端的DSR与RI线相连，然后将信号交叉连到DTR输入端上。信号线与屏蔽地线直接相连。

当两个设备通过同步数据链路通信时，每个设备的发送时钟通常交叉连接，用作另一设备的接收时钟，在有些情况下，两个设备都没有时钟信号源。这时两个设备的时钟由空调制解调器产生，这种空调制解调器称为**调制解调消除器**。

2.6.2 EIA-530

EIA-530接口的信号线与EIA-232D接口的信号组相同。不同之处是EIA-530接口采用RS-422A/V.11差分（平衡）电信号实现长电缆长度与高比特率。这意味着，如果辅助信号线也要用，37针的连接器要求附加一个9针的连接器。

2.6.3 V.35

V.35接口原为DTE到宽带（模拟）同步调制解调器的接口定义，以速率48～168 kbps操作。
87 接口所用的信号线除掉不支持辅助信道线或测试线外，与EIA-232D相同。电信号是非平衡（V.28）与平衡（RS-422A/V.11）的混合。非平衡线用于控制信号，而平衡线用于数据与相关的定时/时钟线。信号类型混合器指最大电缆长度与EIA-232D/V.24相同。然而，在有些应用中，只用发送数据线与接收数据线（时钟）。因为所有这些线采用差分/平衡信号，适用于长电缆。完整V.35接口用34针连接器，但在那些只用数据与相关时钟信号的应用中，通常采用较少针的连接器。

2.6.4 X.21

X.21接口是规定DTE到公用数据网的DCE的接口。公用数据网的实例在第8章中描述，它包含X.25分组交换网与电路交换数据网。X.21接口也适用于$n\times64$ kbps的数字租用线路的终端接口。连接器与相关信号线如图2-35所示。

所有信号线采用平衡（RS-442A/V.11）驱动器与接收器。它是一个同步接口，另外发送（T）与接收（R）数据线/数据线对具有一个信号单元定时（时钟）线S，与字节定时线B。在第8章将看到，控制线（C）和指示线（I）连同发送与接收线一起通过全数字电路交换数据网建立连接。

X.21的一种变形称为X.21bis，有时也用于与带有（模拟）同步调制解调器一起使用的DTE到公用数据网的接口，使用V.24/V.35接口的常规控制线。

2.6.5 ISDN接口

ISDN是PSTN（模拟）接口的全数字化替代。如2.5.2节说明，一个全数字化语音电路以

64 kbps（双工）速率工作，将在第8章看到，ISDN中一个基本速率终端提供两个64 kbps的电路及一个16 kbps的附加电路（双工）用于呼叫建立与清除。虽有三个独立电路，但它们与最 88 近电话局/交换中心复接在一个独立线对上用于传输，**网络终端**（NT）设备在两个独立线对上将向前通路与返回通路分隔开——每个通路包含三个标识电路。如果需要，NT也向用户终端设备提供电源。向前通路与返回通路合称为***S*接口**，ISO 8877标准规定称为**RJ45**的8针连接器，用于连接终端设备到NT，这个接口如图2-36所示。

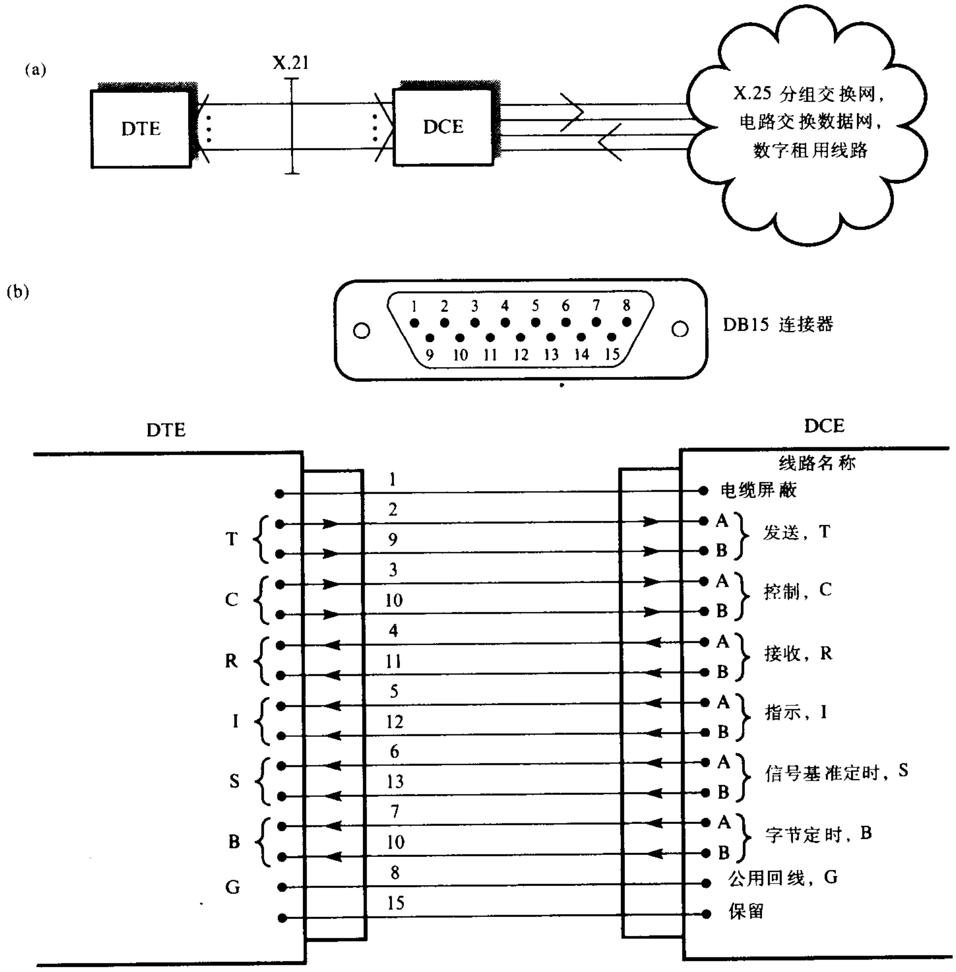


图2-35　X.21 标准接口

(a) 接口功能　(b) 插口、针脚与信号定义

从NT到终端设备的主电源，供给发送与接收线对，也可通过针脚7、8选用第二电源。为了连接低速率的设备（为模拟PSTN设计使用）到高速率的S接口，采用称为**终端适配器**（TA）的设备。我们将在第8章看到TA执行**速率适配功能**的内容。这个功能在V系列标准V.110与V.120中定义。

2.6.6　标准综述

此处所讨论标准仅是ITU-T提出的用于公用电话网（PSTN）的系列标准的一部分。这些标准称为V系列，其概况如图2-37(a)所示。 89

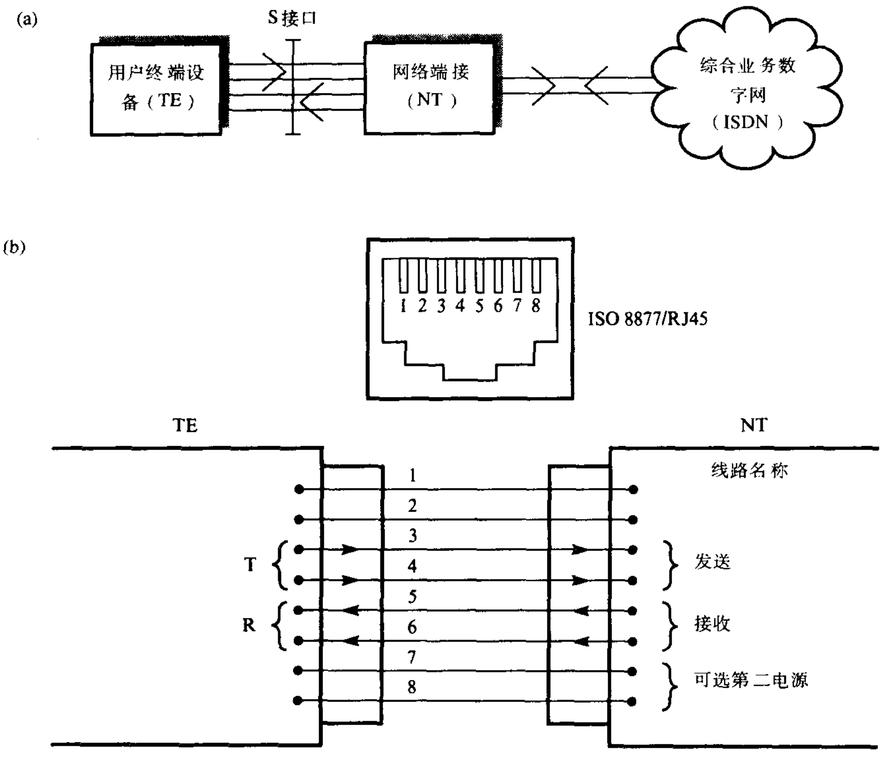


图2-36 ISDN S接口

(a) 接口功能 (b) 插座、针脚与信号定义

如图2-37(b)所示，两个物理接口标准是V.24(EIA-232D)与V.35(EIA-430)。前者适用于普通（亦即低比特率）电话线路，而后者适用于从PTT机构租用的**宽带**线路。由于租用线路是专线，不经过交换设备，所以能在两地之间直接提供点对点线路（链路），一般可操作数据率在48 kbps ~ 168 kbps。在3.3节将看到，由于数据速率相对高，这些线路需采用同步方式转输。

图2-37(a)所示的各种标准经严格规定，包括调制方案类型和附加接口控制线的数量和用法。这样，用户购买遵循V.21标准的调制解调器，就可以与不同厂商生产的V.21兼容调制解调器互换使用。

有些调制解调器设计采用双线交换连接，而另一些采用双线交换电路（连接）与四线租用电路。两种不同电路配置如图2-37(b)所示。

关于四线租用（固定）电路，一对线路通过PSTN建立，并租给用户。一对线用于发送，
90 而另一对用于接收。因此，提供双工的能力。这种线路适用于在两个DTE之间提供固定连接。由于运行成本高，适合高密度用户通信量的应用。

请求一个交换连接，或者还没有达到足够通信量要求，一个固定电路的应用需采用一般交换连接。这样的连接仅用单独一对线（双线）、双工（两个方向）操作，通过称为四线到双线**混合转换器**的设备来完成。这个设备有两个绕组，用于端接和互连，使得从调制解调器调制部分输出的调制（模拟）信号仅耦合到双线电路。同样的，从双线电路来的接收信号仅耦
91 合到解调器输入端。

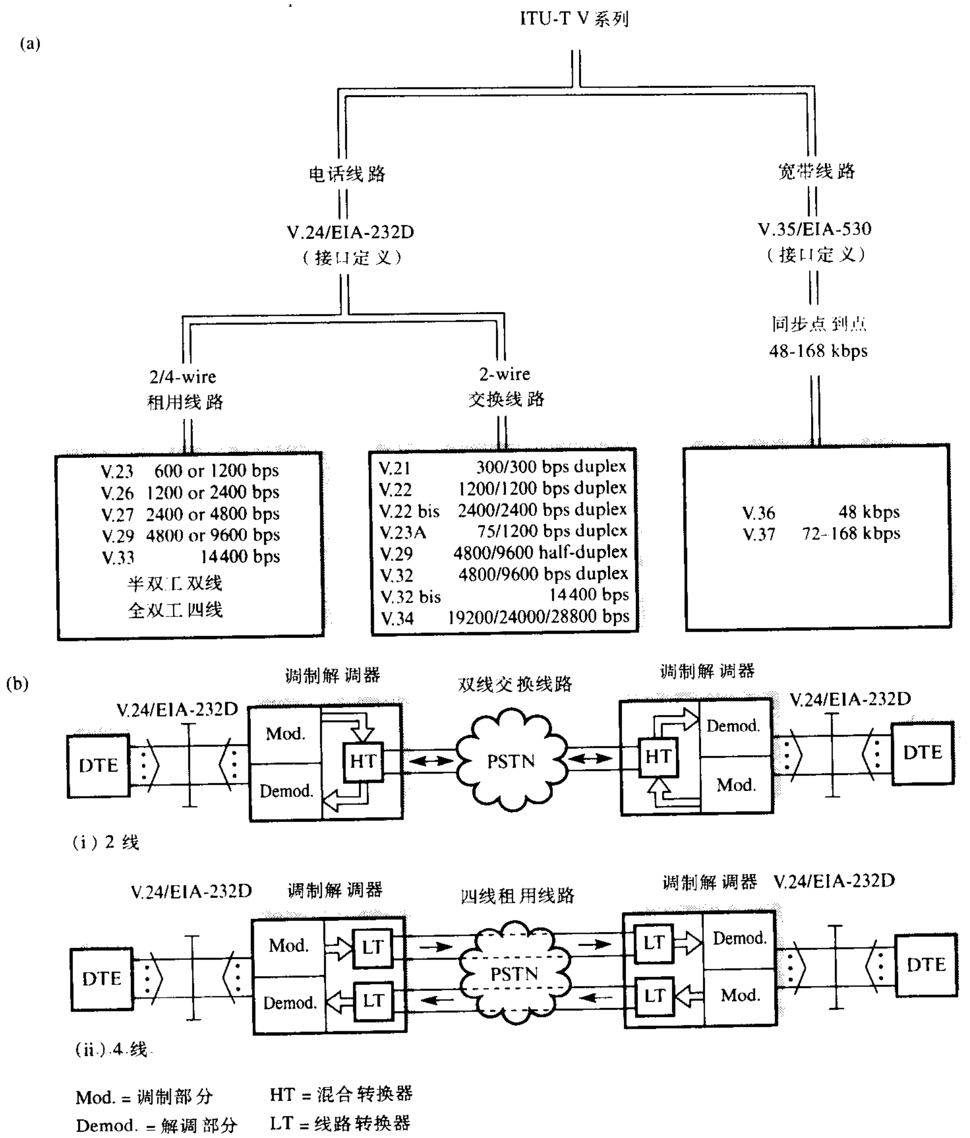


图2-37 调制解调器概要

(a) ITU-T V系列标准 (b) 调制解调器可选性

由于混合转换器不完善，从调制器输出一部分信号，称为**回波信号**，被本地解调部分的输入端接收。所以，一般只有半双工（两个方向交替）被这种电路采用。采用较复杂的调制技术与附加线路（称为**回波消除器**）使双工（同时两个方向）操作在双线交换电路上以高比特率（9600 bps）实现成为可能。这种方法用在V.32/V.33调制解调器上，也包括安装在调制解调器中的纠错电路。这个特性将在5.3.4节进一步讨论。

习题

2.1 简要描述下列传输介质的应用和局限性：

(a) 双线开放线

(b) 双绞线

(c) 同轴电缆

(d) 光纤

(e) 微波

2.2 概要说明下列光纤使用传输模式之间差异：

(a) 多模阶跃折射

(b) 多模渐变折射

(c) 单模

2.3 两个地面微波反射器之间的最大传输距离 d 为：

$$d = 7.14\sqrt{Kh}$$

其中h是反射器离地面的高度，K是考虑到地球弯曲的修正系数。假设K=4/3，用选择的h值确定d。

2.4 参照图2-5(b)，假定7个单元重复模式，确定蜂窝系统的频率分配。解释如图所示的3个单元重复模式的优点。

2.5 概要说明下列因素对传输二进制信号的影响：

(a) 衰减

(b) 有限带宽

(c) 延迟失真

(d) 线路与系统噪声

2.6 利用2.2节积分公式求周期二进制信号的傅立叶级数，假定：

(a) 单极性编码

(b) 双极性编码

2.7 解释当推导信道最小带宽要求时，为什么用二进制序列101010…表示最坏情况序列。

2.8 按下列比特率发送数据，求信道要求最小带宽，假定最坏情况的信号，(i)仅接收到基频分量 (ii)接收到基频分量及第3谐波频率分量：

(a) 500 bps

(b) 2000 bps

(c) 1 Mbps

2.9 在PSTN中采用的调制解调器用QPSK调制方式，每个信号单元有4个值（相）。假定一个无噪声信道带宽为3000 Hz，求

(a) 以bps表示的奈奎斯特最大信息传输率

(b) 调制方案带宽效率

2.10 从习题2.6的单极性二进制信号的傅立叶级数出发，求下列调制器的输出表达式。

(a) ASK

(b) FSK

对每个方案用图形方式表示，假设信道的要求带宽仅接收最坏情况周期信号的基频分量

92 与第三谐波分量。

2.11 假定采用FSK调制，估计以下列假定速率传送数据，求信道发送数据的最小要求带宽

(i) 最坏情况下基频被接收；(ii) 基频分量与第三谐波分量被接收。

(a) 300 bps

(b) 1200 bps

(c) 4800 bps

比较该结果与书中ASK调制所得结果。

2.12 从2.5节双极性信号傅立叶级数出发，求出一个二进制（两个电平）PSK调制器输出的表达式。

用该表达式如何求出通信信道的最小带宽要求。

2.13 解释术语"NEXT消除器"以及该电路如何改善数据传输速率。

2.14 说明下列卫星/无线电波访问控制方法的操作原理。

(a) Aloha

(b) 预分配FDMA

(c) 按需分配TDMA

2.15 解释术语"信号传播延迟"与"传输延迟"。假定电信号传播速度等于光速，对下列数据链路类型与1000位的数据，确定信号传播延迟与传输延迟的比a：

(a) 100m的UTP线以及1 Mbps传输速率

(b) 2.5 km的同轴电缆以及10 Mbps传输速率

(c) 卫星链路以及512 kbps传输速率

2.16 画图表示下列的信号类型和传输介质，在两个DTE之间传输二进制数据时使用的接口电路和相应的信号电平：

(a) V.28与开放线

(b) 20 mA电流环与开放线

(c) RS-422与双绞线

(d) 同轴电缆

(e) 光纤

并指出各种信号类型的性质

2.17 (a)为什么经过PSTN传输二进制数据必须使用调制解调器？利用图阐明下列调制方式：

(i) 幅移键控（ASK）；

(ii) 频移键控（FSK）；

(iii) 相位相干PSK；

(iv) 差分PSK。

(b) 讨论在选用调制解调器解调部分的载波频率和带宽时应考虑的因素。

2.18 推导下列传输信道的最大理论信息速率：

(a) 具有带宽为500 Hz和信噪比为5 dB的用户电报网（国际报文交换）

(b) 具有带宽为3100 Hz和信噪比为20 dB的电话交换网

2.19 用9600 bps的调制解调器发送数据，确定下列调制方法的最小系统带宽：

(a) FSK

(b) 16-QAM

2.20 在PSTN上用PSK调制解调器以4800 bps发送数据，最小位差错率为10^{-6}。如果假定温度为20℃时得到E_b/N_0是10 dB，求最小接收信号电平。

2.21 列出EIA-232D/V.24标准接口的主要信号并说明其功能，用一个时序图指明每条线的作用。可采用下例：一个终端用户采用半双工方式经过PSTN与远程计算终端连接，进行

93 包括数据交换在内的传输。

2.22 (a) 什么是空调制解调器的功能？表示空调制解调器中内部的连接，并指出每个连接的意义。

(b) 列出EIA530/V.35标准接口的主要信号，并简述其功能。

2.23 概要说明有关模拟语音信号的数字化的下列术语：

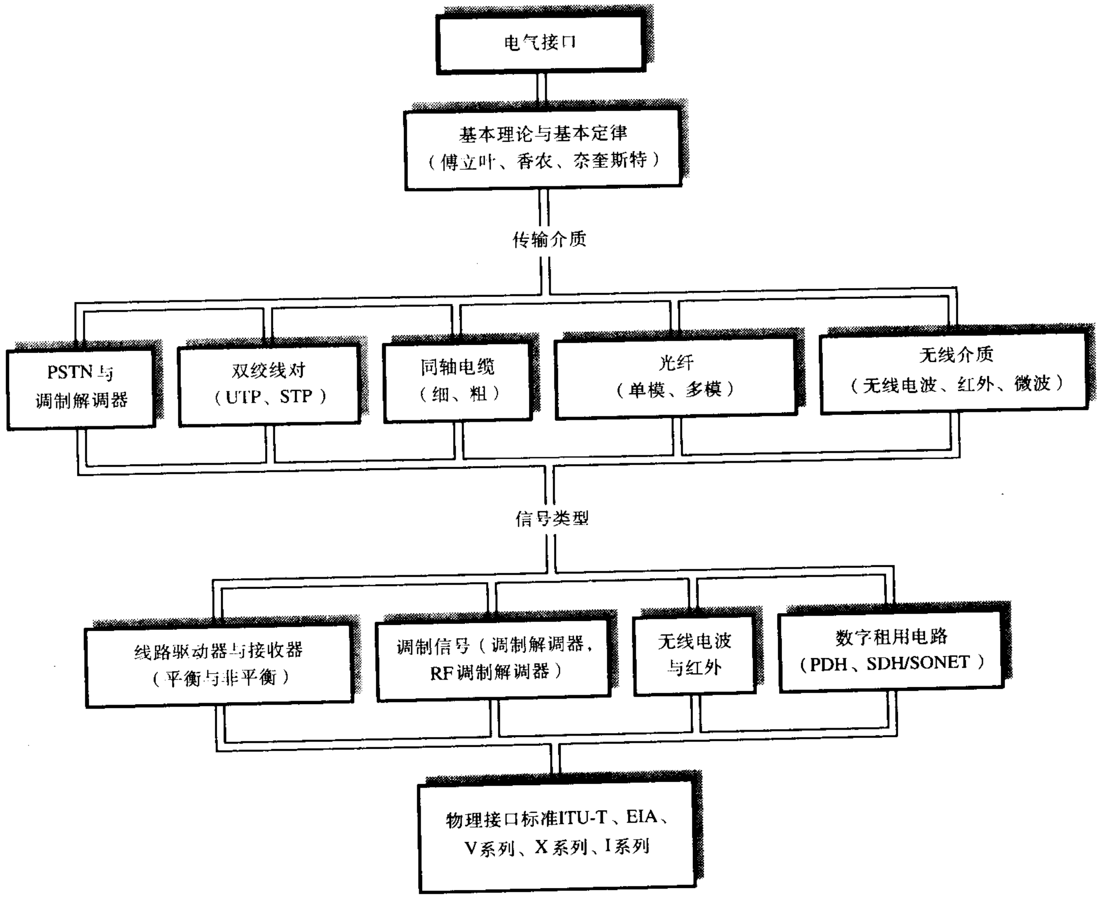
(a) 采样与PAM信号

(b) 量化

(c) 压扩

94 2.24 说明北美与ITU-T推荐标准关于租用数字线路的数字化结构。在这些方案中每一个帧结构如何表示成基本复接群，求每个线路可用数据速率。

本章概要


电气接口
基本理论与基本定律
（傅立叶、香农、奈奎斯特）
传输介质
PSTN与调制解调器
双绞线对（UTP、STP）
同轴电缆（细、粗）
光纤（单模、多模）
无线介质（无线电波、红外、微波）
信号类型
线路驱动器与接收器（平衡与非平衡）
调制信号（调制解调器，RF调制解调器）
无线电波与红外
数字租用电路（PDH、SDH/SONET）
物理接口标准ITU-T、EIA、V系列、X系列、I系列

95

第3章 数据传输

本章目的

读完本章，应该能够：

- 了解两种最常用信息交换码的结构；
- 理解异步传输控制方案与同步传输控制方案的差别；
- 说明异步传输的位（时钟）同步方法与字符（字节）同步方法以及所用不同编码方法；
- 描述异步传输与面向字符的同步传输如何实现帧同步；
- 理解术语数据透明性、字符填充与位填充；
- 说明用同步传输达到位（时钟）同步采用的几种可选技术；
- 说明检测传输（位）差错采用的几种可选方法的操作与应用领域；
- 描述数据压缩的几种最常用方法；
- 说明不同类型多路复用器操作原理。

引言

数据通信是两台DTE之间交换数字编码信息。两台设备的物理距离可能从几十米（例如
96 两台个人计算机）到数百公里（例如通过公用载波网连接的两台设备）。

在数据通信领域中，常用术语“数据”表示两台设备间交换的由一个或多个用数字编码的字母和数字字符组成的集合或块码，这些数据典型地表示一串数字或计算机所存文件的内容。当用数据通信设备传送这类数据时，除了数据报文外，为了克服通信中传输差错的影响，需要通信双方（DTE）交换控制报文。为了识别这两类报文，我们采用更一般的术语**信息**描述通过数据通信设备交换的实际用户数据。

在数字系统中，信息只要有一位（二进制位）丢失或损坏，影响都是严重的。当我们设计数据通信设备时，在需要的情况下必须采取适当措施检测及纠错传输过程中任何可能的信息丢失或损坏。因此，数据通信不仅涉及在物理介质上传输数据，还涉及检测技术以及纠正传输差错，也关系到控制数据传输速率，规定数据传输格式以及相关问题。

本章与下一章讨论数据通信的基本概念，特别是，信息经过**位串行传输介质**可靠（无错误与无丢失或无重复）传送的技术。正如在第2章所见，传输介质可以是物理线路（双绞线、同轴电缆或光纤），也可以是无线信道。所以，在许多情况下，我们采用更一般的术语**数据链路**描述连接两台DTE的链路。本章将讨论有关两台DTE之间数据传输的基本技术与电路，而第4章讨论通信双方之间控制数据传送的基本技术。但必须强调一点，不论采取哪种检错（纠错）方案，100%检测出所有可能的传输错误组合是不可能的。所以，实际上，各种检错与纠错技术的目的是给出可接受概率，即在接收到报文中出现未被检测出错误的可接受低概率。

3.1 数据传输基础

当我们通过键盘向计算机输入数据时，每一个字符键（例如字母字符或数字字符）通过键盘内的电子线路**编码**成一个二进制编码模式，该模式使用信息交换国际标准码方案的一种。
97 为了用惟一的模式表示键盘上的所有字符，我们使用7位或8位，使用7位意味着可表示128个

不同字符，而使用8位可表示256个字符。接下来的计算机输出是相似过程，只不过打印机**解码**接收到的二进制编码模式，并打印相应的字符。我们把每个字符编码位模式称为**码字**

两种最广泛用于此功能的代码是**扩充二—十进制交换码**（EBCDIC）与**美国国家信息交换标准码**（ASCII）。EBCDIC是8位码，IBM制造的大多数设备都使用此码。它如同**专用码**，由于IBM设备在计算机工业中广泛的使用，所以这种码也时常用到。EBCDIC码字的定义如图3-1(a)所示。

ASCII码等同于ITU-T规定的**国际字母表5**（IA5），也被国际标准化组织使用，称为ISO 645。每个码字是7位，定义如图3-1(b)所示。

(a)

位的位置			4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
			3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
			1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
8	7	6	5																
0	0	0	0	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI
0	0	0	1	DLE	DC1	DC2	DC3	RES	NL	BS	IL	CAN	EM	CC		IFS	IGS	IRS	IUS
0	0	1	0	DS	SOS	FS		BYP	LF	EOB	PRE			SM			ENQ	ACK	BEL
0	0	1	1			SYN		PN	RS	UC	EOT					DC_4	NAK		SUB
0	1	0	0	SP										¢	.	<	(	+	|
0	1	0	1	&										!	$	*	)	;	¬
0	1	1	0	-	/										'	%	_	>	?
0	1	1	1											:	#	@	,	=	"
1	0	0	0		a	b	c	d	e	f	g	h	i						
1	0	0	1		j	k	l	m	n	o	p	q	r						
1	0	1	0			s	t	u	v	w	x	y	z						
1	0	1	1																
1	1	0	0		A	B	C	D	E	F	G	H	I						
1	1	0	1		J	K	L	M	N	O	P	Q	R						
1	1	1	0			S	T	U	V	W	X	Y	Z						
1	1	1	1	0	1	2	3	4	5	6	7	8	9						□

(b)

位的位置			7	0	0	0	0	1	1	1	1
			6	0	0	1	1	0	0	1	1
			5	0	1	0	1	0	1	0	1
4	3	2	1								
0	0	0	0	NUL	DLE	SP	0	@	P	\	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	_	o	DEL

图3-1 标准交换码

(a) EBCDIC (b) ASCII / IA5

这两种编码方案都提供所有常用的字母、数字和标点字符，统称为**可打印字符**；再加上一组附加的**控制字符**，也称为**不可打印字符**。控制字符包含有：

- 格式控制符——BS（退格），LF（换行），CR（回车），SP（空格），DEL（删除），
98 ESC（转义）与FF（换页）。
- 信息分隔符——FS（文件分隔符）与RS（记录分隔符）。
- 传输控制符——SOH（标题开始），STX（正文开始），ETX（正文结束），ACK（确认），NAK（否定确认）与SYN（同步空闲信号）。

其中，某些控制字符在3.2.3节中说明。

尽管这些码用于输入与输出，但一旦作为数字数据输入计算机，通常转换成等价的固定长度的二进制形式，可能是8位、16位或32位，并被存储。我们称8位二进制组为一个**字节**，更长的称为**字**。由于每个字用一组二进制位表示，所以通常两台DTE间通信以8位固定长或它的倍数为单元。因此，在某些情况下，通过数据链路发送8位组可表示一个二进制编码的可打印字符（7位加一个检错位），而在另一些情况下，它可表示某数值的一个8位部分。后一种情况，称其为字节单元，或用作通信时称为**8位组**。

3.1.1 位串行传输

在一台设备内部，各子单元间距离近，因此连接的线也短，通常在子单元间传送数据时
99 使用分离的连线携带数据的每一位。这意味着采用多根导线连接每个子单元，并称为以**并行传输方式**交换数据，目的是使每个字传送有最小的时延。

当在两台物理上分离的设备之间传送信息时，特别如果距离大于几米，由于线路成本贵，通常采用一对线路以**位串行传输**，即组成数据的8组位的每一位用一个固定时隙传送，每次发送一位。

并行和串行传输方式如图3-2所示。我们通常把数字电子设备中的一根导线作为一位，表示成相对于参考电平的一个特定电压电平。在图3-2中，高电平信号表示二进制的“1”，而等于参考电平的低电平信号表示“0”。相反，在第2章中，串行传输描述高电平信号与低电平信
100 号是相对参考电平的正电平与负电平。

3.1.2 通信方式

某人作报告或演讲，信息基本上以一个方向传播。然而两人交谈，消息（信息）通常是在两个方向交换。一般，信息交换是交替进行的，但也有可能同时交换。同样地，在两台设备间传送数据时，有三种类似的操作方式：

1) 单工　仅用于一个方向的数据传输。例如，在数据录入系统中，监控器以一定的时间间隔返回一个读数到数据收集设备。
2) 半双工　用于两个互连设备交替地交换信息（数据）。例如，某设备仅对另一设备的请求发回数据作为响应。显然，每次传输后，两个设备必须在发送方式与接收方式间切换。
3) 双工　也称作全双工，用于两互连设备彼此同时双向交换数据。例如，根据吞吐量要求，数据可在各个方向上独立流动。

选择通信方式是重要的，因为在许多分布式系统中，通信线路一般是向PTT机构租用。租用一条线路比租用两条线路费用低，如果仅要求单工，则一条线路就能满足要求。

3.1.3 传输方式

正如3.1节开始时提到，数据在两台DTE之间传输以多个固定长单元的形式进行，典型的

单元长是8位。在有些情况下，如果计算机传送一个数据文件，如源程序，数据是由8位二进制编码的字符块组成。在另一些情况下，如果数据文件是目标（编译）程序，数据是由8位字节块组成。

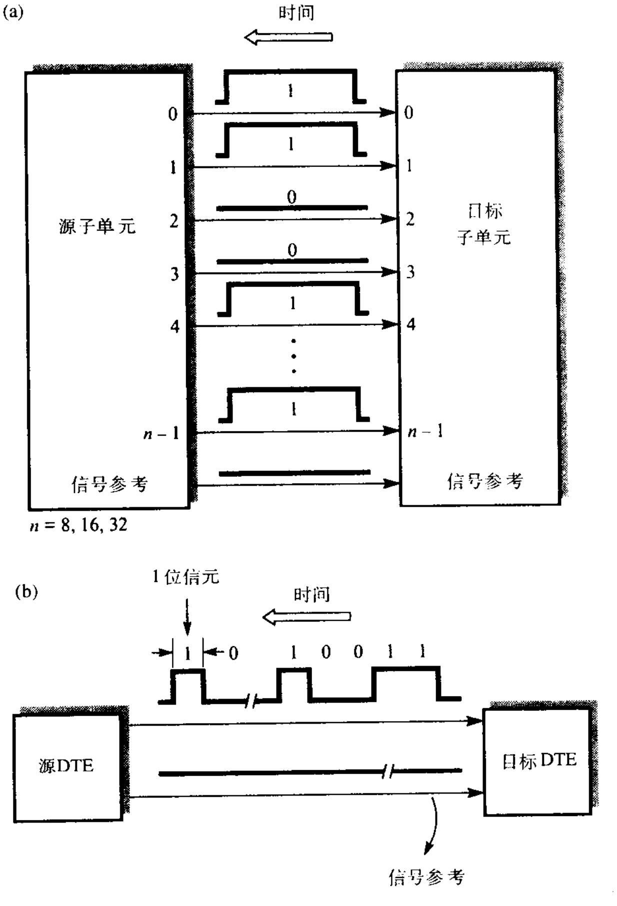


图3-2 传输方式

(a) 并行 (b) 串行

由于每个字符或字节是以位串行方式传输，接收DTE收到随位模式（字符串）变化的信号电平组成的消息。为了接收设备能正确译码和恢复位模式，必须确定下列几点：

1）每个位信元周期的开始（为了在进来信号的位信元中心采样）。

2）每个单元（字符或字节）的开始位与结束位。

3）每个完整信息块（也称为帧）的开始与结束。 101

以上三点分别称为**位或时钟同步**、**字符或字节同步**与**块或帧同步**。

通常有两种实现同步的传输方式：**同步传输**与**异步传输**。它们的区别取决于发送器与接收器的时钟是独立的（异步的），还是同步的。**异步传输发送器**按时钟（位）同步与字符（字节）同步独立地处理每个字符（字节），接收器在收到每一个新字符的开始重新同步。**同步传输发送器**发送的数据是由连续位串组成的完整字符帧（块），接收器尽力在整个帧（块）发送期间与进入的位流保持同步。

1. 异步传输

如前所述，这种传输方式主要用于发送数据是随机产生的情况。例如，用户用键盘与计算机通信。显然，用户在终端上以随机的时间间隔、不确定的速率键入每个字符。这意味着，传输线长时间处于空闲（称为**传号**）状态。接收器必须能够在每个新字符到达时重新开始同步。为此，对于异步传输的每个字符或字节必须封装于附加**起始位**与一个或多个**停止位**之间，如图3-3所示。

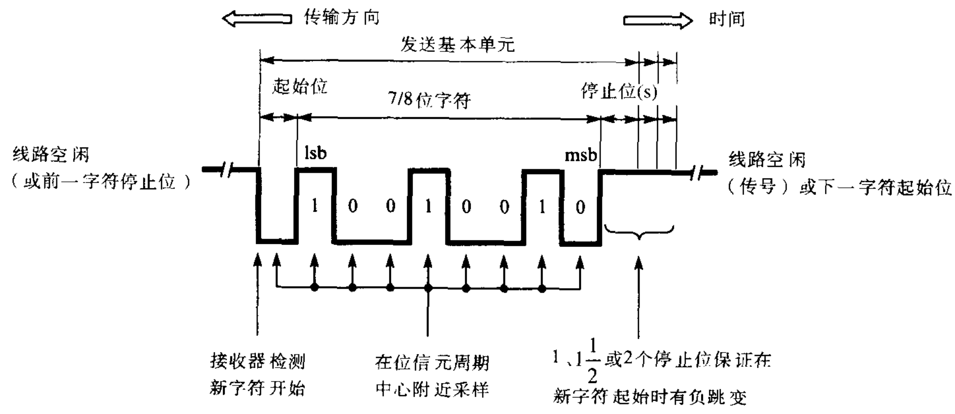


图3-3　异步传输基础

虽然，异步传输主要用于键盘（或更一般终端）与计算机之间的字符传输，但它也可以用于两台计算机之间的字符（或字节）块传输。此时，块中字符之间的发送是无时延地一个
102 接着一个，后继字符的起始位紧跟前一个字符的停止位之后。

如图3-3所示，起始位与停止位极性相反，这就保证每个连续字符之间跳变最小（1→0→1），它与所传输字符位序无关。空闲周期后第一个跳变（1→0）是接收设备用于确定新字符起始位。并且利用接收器的时钟频率是发送数据比特率的N倍（通常$N=16$），接收设备对所收到信号在每个位信元周期中心采样，从而可靠地确定（好的近似）发送字符每个位的状态。如图3-3所示，将在3.2节中进一步讨论。

由此可推出，发送每一项用户数据须用10或11位（1个起始位与1个或2个停止位）。如果假定每8位有1个起始位和2个停止位，数据传输速率为1200 bps，则数据速率为1200/11，或每秒约110字节。实际上，由于某种原因，可用数据速率要低于此值，理由将在3.2.3节描述。

当定义线路的传输速率时，通信工程师常用术语**波特**。然而，这个术语的正确含义是每秒线路信号转换的次数。如果每个发送信号是两种状态之一，则术语波特与每秒位数（bps）是等价的。然而，在第2章描述，有些情况，信号可取两个或多个状态，因此，每个发送单元可用于传送多于一位的信息。为了免于混淆，我们用**信号速率**规定每秒线路信号转换数（波特），而数据或信息传送速率表示每秒传送数据位数（bps）。例如，每个信号单元有4位，信号速率为300波特的数据速率是1200 bps。在异步线路上通常的信号速率是110、300、1200、4800、9600与19 200 bps几种，其中高速率用于短距离。

最后，当发送字符（或字节）块时，每块封装在一对特定（传输控制）字符之间以达到块（帧）同步。它保证接收器在空闲周期后，接收到起始字符（字节）能确定一个新帧开始发送，同样地，接收到结束字符（字节）表示帧结束。

2. 同步传输

正如前述，我们通常在字符产生速率不确定，或者字符块的传输速率相对较低时使用异步传输。在异步传输中，对于字符同步，使用每个字符附加位，以及使用较多位的同步方法，都是可接受的。然而，以高比特率传输大量数据块，可选用同步传输方法。 103

同步传输中，整个数据块或帧在每8位元素之间作为连续二进制数据流无时延地传输。为了使接收设备得到各级同步必须做到下列几点：

1）发送的二进制数据流经适当的编码，以使接收器保持位同步。

2）每个帧的前面设置一个或多个保留字节或字符，以保证接收器能在正确的字符或字节边界可靠地解释收到的比特流（字符或字节同步）。

3）每一帧的内容封装在一对保留字符或字节之间作为帧同步。

在同步传输情况下，连续传输帧之间可以持续发送同步空闲字符（字节）以使接收器保持位同步与字节同步，或者每个帧先发送一个或多个特定同步字节或字符以便接收器能恢复同步，如图3-4所示。

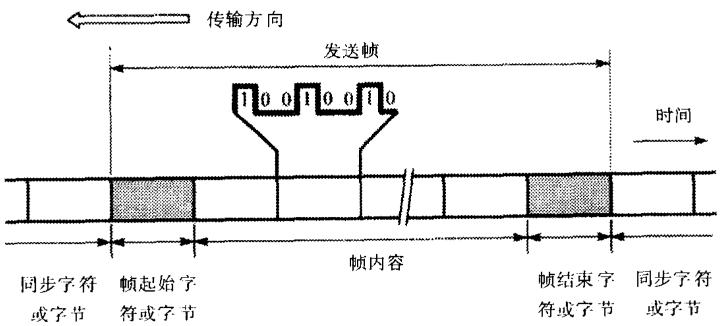


图3-4　同步传输基础

我们将在3.3.1节介绍几种实现位同步的位编码方案。如同异步传输一样，我们必须保证帧的起始与结束字符（字节）是惟一的，即它们不出现在传输的帧内容中。显然，如果帧的内容包含二进制码文件，这就不能保证同步传输。对于这种可能性，必须采取额外措施。我们将在3.2.3节、3.3.2节与3.3.3节详细讨论。

实例3-1

由100个8位字符组成的报文在数据链路上发送，用下列两种传输控制方案求所需的附加位的个数：

（a）每个字符有1个起始位与2个停止位，而每个报文有1个帧起始字符与1个帧结束字符； 104

（b）每个报文包括2个同步字符和1个帧起始字符及1个帧终止字符的同步传输。

解：

（a）每个字符附加位的个数=1+2=3，因此需要附加位个数是（3×100）+$2\times$（3+8）=322。

（b）在同步传输中，附加位的个数简单地为2个同步字符与1个帧开始字符与1个帧结束字符，即$4\times8=32$。

3.1.4　差错控制

在两台DTE之间发送串行二进制数据流期间，一个很常见的问题是发送信息被损坏，特别当两台DTE之间距离很长时，如通过PSTN进行。也就是说，对应于二进制0的信号电平在

极端情况被改变使接收器解释为二进制1的信号电平，或者反过来。为此，需要寻找检测与纠正此传输差错的方法。

对应不同的传输方式，可选用不同差错检测与纠正方案。当使用异步传输时，由于每个字符作为独立单元处理，我们通常在每个被发送字符中插入一个附加二进制码（位）。这个附加的数码称为**奇偶校验位**，我们将在3.4.1节讨论它的功能。

对于同步传输，帧是传输的基本单位。因而通常是根据整个帧来检测可能的传输差错。并且由于帧的内容可能很大，因而差错多于一位的概率增加，故必须采用更为复杂的差错检测技术。其中之一是采用差错检测序列方法，通常是由传输设备根据被传帧的内容计算出差错检测数序列，并把它附加到帧的末尾，即插到字符之后或帧终止字符之前。

在帧传输期间，接收器根据收到的内容及帧终止字符或字节，重新计算一个新的差错检测数，并将它同发送的检测序列数比较，如果不等，则说明传输有差错。

上述两种方案只能使接收器检测出传输差错。因此，我们需要一种方案能使接收器在检测错误的同时纠正差错，从而得到正确的发送信息。同样，纠错也有多种方案。例如，考虑终端与计算机之间用异步传输发送数据的情形。当用户键入字符，按已列出字符的编码向计
105 算机发送，然后对应于接收到位流的字符被计算机“回波”返回并在用户终端屏幕显示。如果显示字符与键入字符不同，说明传输出现差错，这时，用户发送一个特殊（删除）字符通知计算机取消上次收到的（错误）字符，这称为**差错控制**。当发送字符块时，采用执行相同功能的方法。我们在第4章讨论更为一般的差错控制方法。

3.1.5　流量控制

两台设备之间发送的数据量不多，发送设备立即就能发完所有数据，因为接收设备有足够资源（存储空间）来接受这些数据。然而，许多情况不是这样，时常需要控制数据传输的流量，以确保接收设备不致因为没有足够的存储空间而丢失数据。特别是两台DTE经过中间数据通信网进行通信，时常由于网络只能缓存有限的数据。如果两台设备运行在不同的数据速率下，我们就需要控制快的那台设备的平均输出速率以防止通信网络拥塞。控制两台DTE之间的信息流量称为**流量控制**。在第4章将介绍一些方法。

3.1.6　数据链路协议

差错控制与流量控制是数据链路控制协议中最基本的两个组成部分。协议原则上是通信双方必须遵守的一组约定或规则，保证经过串行数据链路的交换信息能正确地接收与解释。除差错控制与流量控制外，数据链路协议还定义下列内容：

- 交换数据的格式，即每个数据单元的位数以及采用的编码方案。
- 交换的消息类型和顺序，为了通信双方达到可靠的（无差错和无重复）的信息传输。

例如，一台DTE向另一台DTE传送数据之前，通常在它们之间建立连接以保证接收DTE是自由的并准备好接收数据。这个时候通常由发送设备发送一个特殊控制消息（例如呼叫请求或连接请求），接收设备返回一个确定的响应消息（例如呼叫连接或拒绝）。我们将在第5章
106 中讨论几种不同数据链路控制协议。

3.2　异步传输

正如3.1节讨论，两台DTE之间的数据通常是用异步传输或同步传输方式并按多个8位单元（字符或字节）以位串行方式传输。但是，在DTE内部，每个8位单元用并行形式存储、操作与传送。因此，在DTE设备与串行数据链路之间有接口，形成该接口的传输控制电路必须具有下列功能：

- 为准备在数据链路上传输，对每个字符或字节进行并到串转换。
- 为接收设备存储与处理做准备，对每个收到字符或字节进行串到并转换。
- 为接收器提供位同步、字符同步与帧同步的一种方法。
- 产生用于差错检测的差错校验位数字，倘若发生差错，接收端能检测出差错。

并行到串行转换由**并行输入串行输出（PISO）移位寄存器**完成，这个移位寄存器如同它的名字所指，允许一个完整字符或字节以并行形式输入并以串行形式移位输出。同样地，串行到并行转换由**串行输入并行输出（SIPO）移位寄位器**完成。

为了获得位同步与字符同步，我们必须设置接收传输控制电路（通常是可编程的电路）操作特性与发送器每个字符的位数以及采用的比特率相同。

3.2.1 位同步

在异步传输中，接收器时钟（用于移位送入SIPO移位寄存器的进入信号）对进来的信号进行异步操作。为了接收过程可靠地工作，我们必须设计一个方案，借助本地（异步）接收器时钟采样（移入SIPO移位寄存器），使进来的信号尽可能靠近位信元的中心。

为了达到这个要求，本地接收器时钟频率采用N倍（通常N=16）的发送比特率，而每个新位在N个时钟周期之后移入SIPO移位寄存器。每个字符的起始位有关的第一个1→0跳变用于计数过程的开始。每一位（包括起始位）在每个位信元中心（附近）采样。第一个跳变检测出，信号（起始位）在$N/2$时钟周期后采样，然后对接下来的每一位在N时钟周期后采样。一般方案如图3-5(a)所示，定时原理如图3-5(b)所示，而三个不同时钟速率系数如图3-5(c)所示。 107

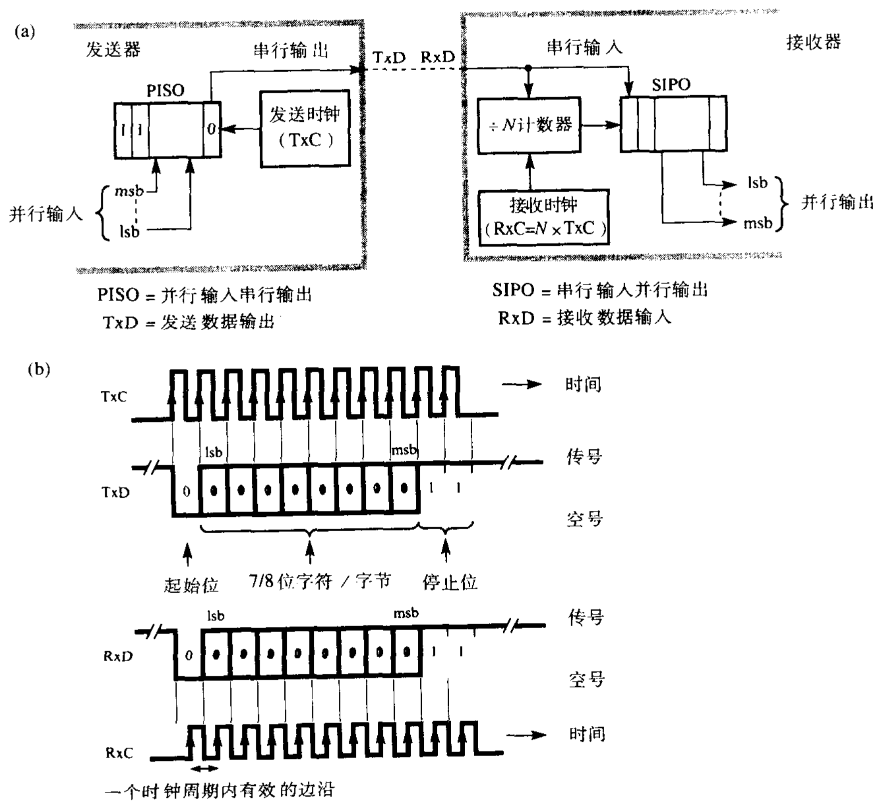


图3-5 异步传输

(a) 操作原理 (b) 定时原理 (c)不同时钟速率实例

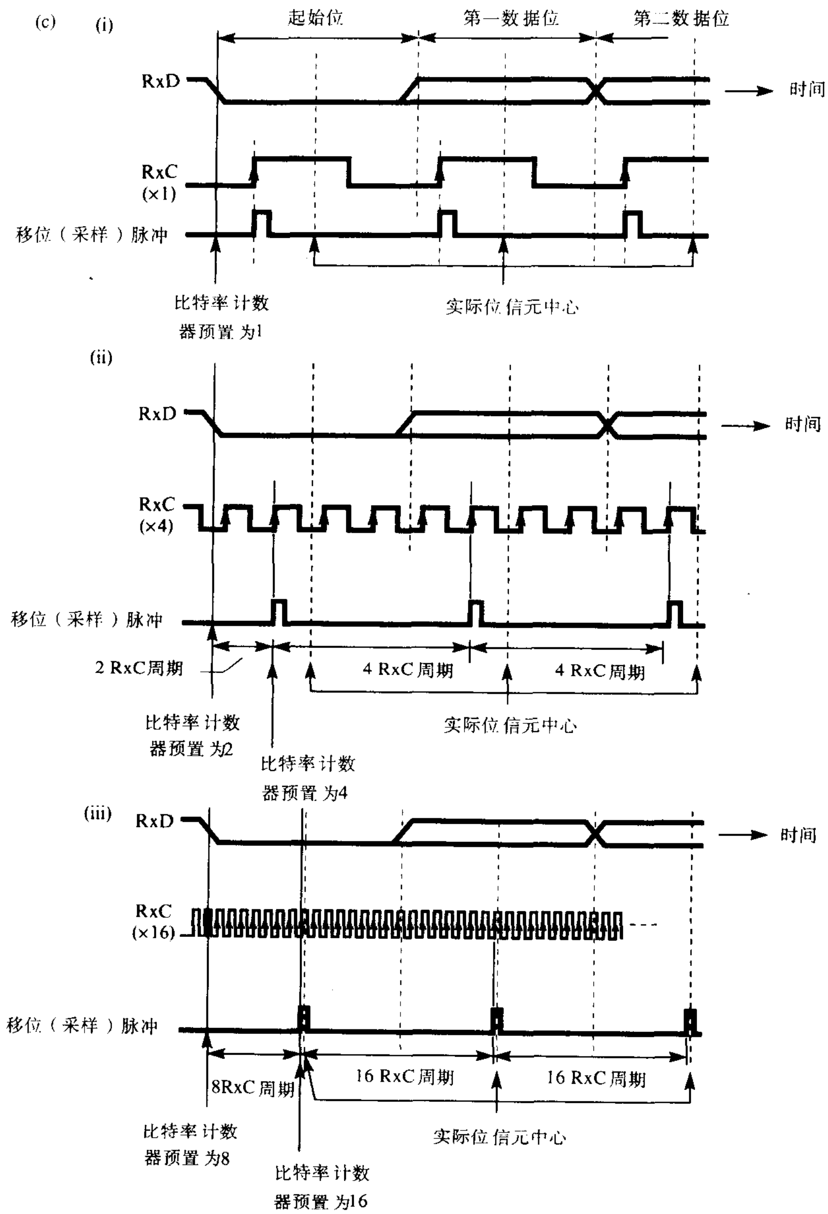


图3-5　（续）

记住接收器时钟信号（RxC）对于进来的信号（RxD）是异步操作，这两个信号的相对位置可以在接收器时钟一个周期的任何地方。正如图3-5(c)所示的那些任意位置。但是，由于异步传输方式的最大比特率是19.2 kbps，故从这些实例我们可推出，时钟速率系数越高，采样瞬间越靠近标准位信元中心。

实例3-2

108　经串行数据链路发送一个数据块，如果接收器采用19.2 KHz的时钟，对于数据传输速率

(a) 1200 bps

(b) 2400 bps

(c) 9600 bps

推出适当的时钟速率系数，并求离标准位信元中心最大的偏差，以位周期百分比表示。

解：这很容易从图3-5(c)推出，与标准位信元中心最大偏差接近一个接收时钟周期。因此： 109

(a) 对于1200 bps，最大RxC系数为 ×16，即19.2/1.2=16，最大偏差为1/16=6.25%。

(b) 对于2400 bps，最大RxC系数为 ×8，即19.2/2.4=8，最大偏差为1/8=12.5%。

(c) 对于9600 bps，最大RxC系数为 ×2，即19.2/9.6=2，最大偏差为1/2=50%。

显然，最后一种情况是不能接受的。对于低质量（特别是时延失真较大）的线路，第二种情况也不是很可靠。因此，应尽可能采用×16的时钟速率系数。

3.2.2 字符同步

正如3.1.3节指出，接收传输控制电路如同发送器一样，规定操作每个字符同样的位数与同样的停止位数。接收并检测出起始位，接收器简单地按规定位数计数得到字符同步。然后，传送接收到的字符（字节）进入本**地缓冲寄存器**，并向控制设备（例如微处理器）发信号表示已接收到一个新字符（字节），再等待下次线路信号跳变指出接收一个新（字符）的起始位。

3.2.3 帧同步

除了位同步与字符同步外，当发送的报文由字符块（通常称为**信息帧**）组成时，接收器必须要确定每个帧的开始与结束，这就称为帧同步。

发送可打印字符块的最简单方法是把完整的块封装在两个特定的（不可打印的）传输控制字符之间：STX（正文开始）表示在一个空闲周期后一个新帧开始，而ETX（正文结束）表示帧结束。由于帧内容仅由打印字符组成，接收器解释接收到的STX字符，作为新帧开始的信号，而一个ETX字符作为帧结束的信号。如图3-6(a)所示。

虽然，所示的方案对可打印字符块的传输是满意的，但当发送纯二进制数据组成的块（如包含编译后程序的文件）时，使用一个ETX字符表示帧结束是不够的。因为此时二进制数 110
据中可能有字节与ETX字符相同，这将会使接收器异常地终止接收过程。

为了克服这个问题，当发送二进制数据时，在两个传输控制字符STX与ETX之前冠以第三个传输控制字符，称为**数据链路转义字符**（DLE）。修改后的帧格式如图3-6(b)所示。

记住：发送器知道发送的每一帧的字节个数。发送帧开始序列（DLE-STX）后，发送器在发送前检查帧的内容，如果发现数据链路转义字符DLE，不管下一字节是什么，都在下一字节前插入第二个DLE字符发送。重复这个过程直到帧中所有字节都被发送完为止。然后发送器发送惟一的DLE-ETX序列表示帧结束。

这个过程称为**字符填充**或**字节填充**。在DLE-STX帧开始序列之后，对每个收到字节，接收器确定它是否是DLE字符（字节）。如果是，接收器处理下一个字节，确定其为另一个DLE还是ETX。如果是DLE，接收器废弃它并等待下一个字节。如果是ETX，这时可以可靠地作为帧的结束。 111

3.3 同步传输

每个字符或字节使用一个附加起始位与一个或多个停止位意味着异步传输的传输容量利用率是相对低的，特别当发送信息由大量字符块组成时。异步传输所用的位（时钟）同步方法，由于比特率提高也降低了可靠性。这是由于，首先来自检测到的第一个起始位跳变仅是近似的事实，其次虽然接收器时钟是标准发送时钟速率的N倍，但两者的微小差异也会引起收到一个字符或字节期间，采样时刻的随机移动。通常我们用同步传输克服这些问题。然而，关于异步传输我们必须采用一个合适方法使得接收器获得位（时钟）、字符（字节）与帧（块）

同步。实际上，有两种同步传输控制方案：面向字符的与面向位的。我们将分别讨论每一种，由于两者都采用同样的位同步方法，因此先讨论位同步方法。

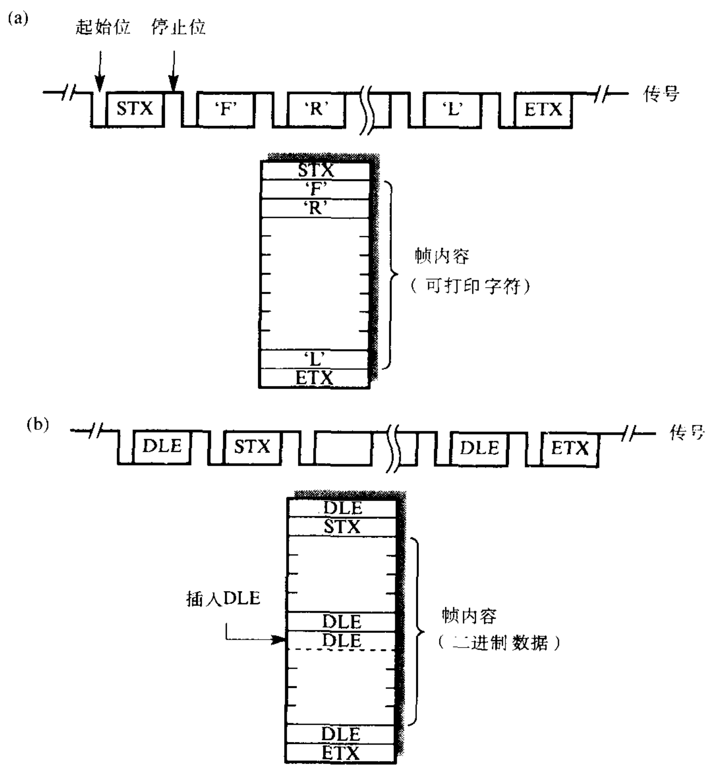


图3-6　帧同步
(a) 可打印字符　(b) 二进制数据

3.3.1　位同步

虽然我们时常用每个字符有没有起始位与停止位区分异步传输与同步传输，但两者的基本区别在于，异步传输时，接收器时钟与进来的（接收）信号异步地（非同步地）运行，而同步传输时，接收器时钟与接收信号同步地运行。

正如我们刚才指出，同步传输不采用起始位和停止位，而将每个帧作为一个连续二进制数据流来传输。接收器用下面两种方法之一获得（保持）位同步。一种将时钟（定时）信息嵌入发送信号，然后由接收器提取；另一种接收器有本地时钟（如同异步传输），但它与接收信号保持同步，通过**数字锁相环**（DPLL）设备。我们将看到，DPLL利用接收信号中的0→1或1→0位跳变以使接收信号在一个很长的可接受时期内保持位（时钟）同步。也利用这两种方法的混合方案。这些方法的操作原理如图3-7所示。

1. 时钟编码与提取

将定时（时钟）信息嵌入发送二进制数据流的三种方法如图3-8所示。在图3-8(a)中，发
112 送的二进制数据流经过编码，用正脉冲表示二进制1，负脉冲表示二进制0。这种编码称为**双极性编码**。双极性编码信号中每个位信元都包含时钟信息，一个简单的电路使得可从接收到的双极性信号中提取时钟信号。由于编码信号在每个编码位（正或负）后都回复到零电平，因此称这种类型的信号为**归零**（RZ）**信号**。

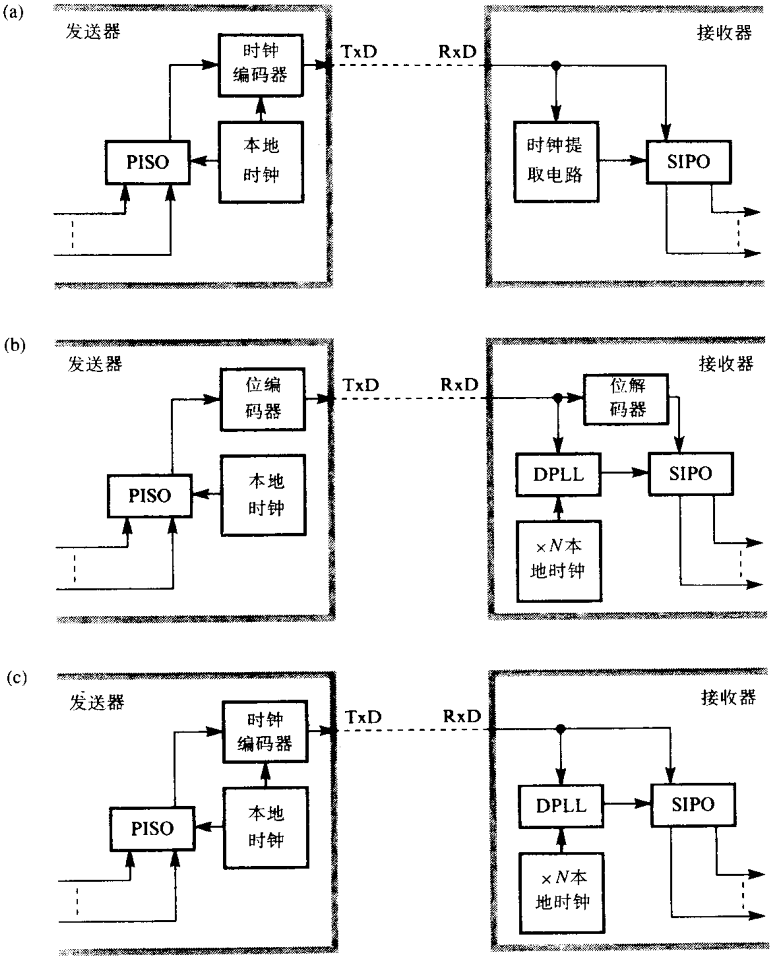


图3-7 同步传输时钟同步选择

(a) 时钟编码 (b) DPLL (c) 混合方案

双极性编码利用三种不同幅度电平（+、0、-）。可是，图3-8(b)所示方案只需要两个电平。编码形成结果信号称为**非归零**（NRZ）**信号**，而编码方案称为**相位或曼彻斯特编码**。在这个方案中，二进制1编码为从低到高的信号，而二进制0编码为从高到低的信号。然而，在每个位信元中心，总存在着跳变（1→0或0→1），时钟提取电路利用这个跳变在位信元后半部分中心产生一个时钟脉冲。在这个位置上接收（编码）信号或者是高电平（二进制1）或者是低电平（二进制0），所以正确信号移位进入SIPO移位寄存器。 113

图3-8(c)所示的方案称为**差分曼彻斯特编码**。这个编码与曼彻斯特编码是不同的。虽然，它依然在每个位信元的中心存在着跳变，但仅当下一位编码是0时，在该位信元开始处有一个跳变。这样的编码效果是输出信号可取两种形式之一，由假设起始电平（高或低）决定。但我们可以看到，其中一种形式是另一种形式的取反。这是有用的特性，如在点到点双绞线链路；如果我们采用差分驱动器与接收器，则接收器连接的两根终端线转到相反方向是无关紧要的。非差分编码方案，输出反向会带来不正确运行。时钟在每个位信元结束时产生，而在位信元中的跳变确定接收位是0还是1。

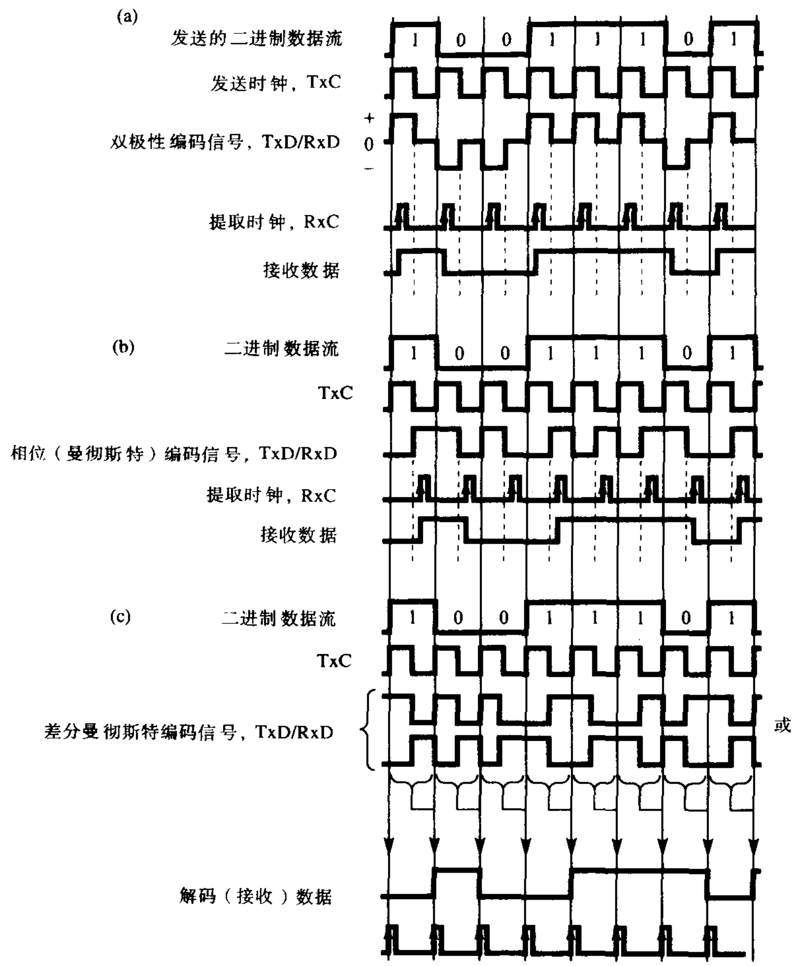


图3-8 时钟编码方法

(a) 双极性 (b) 曼彻斯特 (c) 差分曼彻斯特

114 这两种曼彻斯特编码方案是**平衡码方案**，意味着它们不存在平均（DC）值。这是因为由二进制1（或0）组成的串总是有跳变而不是一个常数（DC）电平。这也是一个重要特征，因为它意味着接受信号可**AC耦合**到接收器电子线路，例如**转换器**。因为发送器的电源单独使用，接收器电子线路可用自己的电源运行。

2. 数字锁相环

另一个发送二进制数据流的时钟编码方法是在接收器利用一个稳定时钟源，它与进来的二进制数据流保持时间同步。但是，由于同步传输方案没有起始位与停止位，我们必须采用在发送波形中有足够的位跳变（1→0或0→1）（它能使接收器时钟在频繁的间隔中重新同步）的办法去编码信息。一种方法是发送数据到**扰频器**，去掉由多个1或多个0连成一片的那些串，形成不规则的发送位流。换句话说，数据可用这样一种方法编码，经常出现合适的跳变。

首先，图3-9(a)表示发送的二进制数据流的不同编码。我们将形成的编码信号称为**非归零反相**（NRZI）波形。关于NRZI 编码，发送一个二进制1信号电平不改变，而发送一个二进制0则信号电平改变。这意味着，NRZI波形进来的信号，不是由1组成的连续的二进制数据流，

总存在位跳变。表面上这似乎与通常NRZ波形没有不同，但在3.3.3节我们将描述如果在面向位的方案中采用0插入，线路上发送的二进制数据流至少每五个位信元带有一个二进制0。因此由于0的长串使每个位信元有一个跳变，编码形成的波形保证跳变的个数，这就允许接收器调整它的时钟，使得它与进来的二进制数据流信号保持同步。

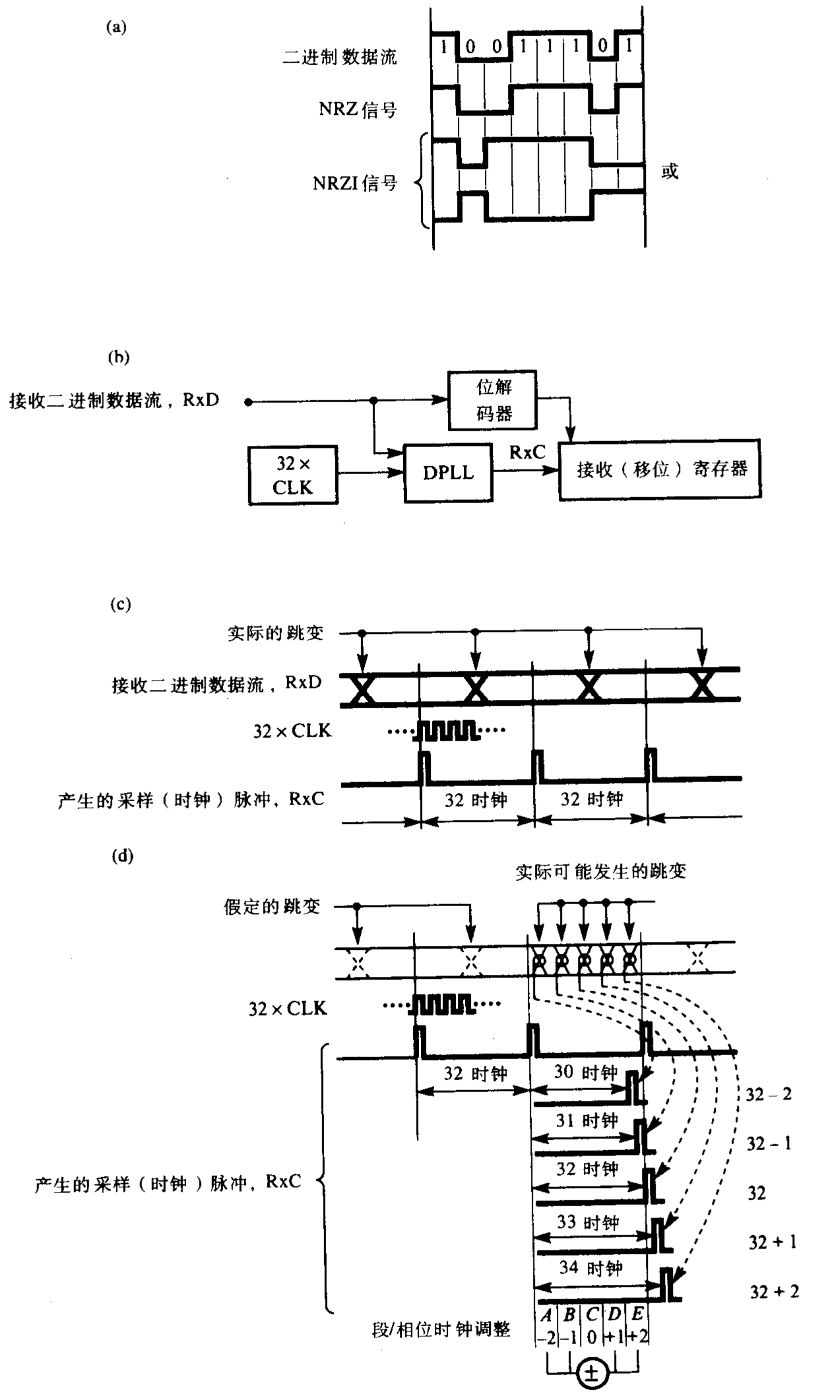


图3-9　DPLL操作

(a) 位编码　(b) 电路框图　(c) 同相　(d) 时钟调整规则

维持位同步的电路称为数字锁相环（DPLL）。一个晶体控制振荡器（时钟源）可以保证频率足够稳定，只是需要在不规则区间对频率作微小的调整。将它与DPLL连接。典型情况下，时钟速率是数据链路比特率的32倍，并且用DPLL驱动接收二进制数据流连续采样间的定时间隔。

假定进来的二进制数据流与本地时钟同步，线路上进来的信号状态（1或0）在每个位信元中心被采样（被时钟送入SIPO移位寄存器）。采样时间间隔严格为时钟周期的32倍，如图3-9(c)所示。

现在假设进来的二进制数据流与本地时钟失去同步，由于本地时钟有微小的变化，采样
115 时刻如图3-9(d)所示进行离散的增量调整。如线路上没有跳变，DPLL在前一次采样后每32个时钟周期产生一个采样脉冲。然而，每当检测到一个跳变（1→0或0→1），按照DPLL认为应出现跳变的位置确定前一个采样脉冲与下一个采样脉冲的时间间隔。为此，每个位周期划分成5段，如图3-9(d)所示的A、B、C、D与E。例如，一个跳变发生在段A期间表明上一次采样脉冲太靠近下一个跳变，因此上一次采样脉冲有点迟，所以，下一个采样脉冲周期缩短为30个时钟周期。同样，一个跳变发生在段E期间表明上一次采样脉冲相对这个跳变太早，所以下一个采样脉冲周期应延长为34个时钟周期。跳变在段B与段D中发生明显较接近假定的跳变，有关的调整较小（分别为–1和1），最后，在段C发生的跳变认为足够接近假定的跳变，无需调整就是正确的。

用这种逐步调整的方法，使产生的采样脉冲保持在每个位信元中心位置附近。实际上，每个段的长度（用时钟周期表示）不相等。靠外面段（A与E）比三个靠里面段离标准中心较远。对于所示的电路，一个典型划分是A=E=10，B=D=4和C=4。我们可容易地推出，在最坏情况下，DPLL需要10个位跳变就可以收敛到波形标准位中心：粗略调整为5个位周期（±2），精细调整为5个位周期（±1）。因此，当使用DPLL时，通常在线路上发送第一帧之前，或在两个帧之间的一段空闲之后，发送若干字符或字节，以提供最少10个位跳变。例如，发送两个由二进制0组成的字符/字节，NRZI编码提供16个跳变，这就保证DPLL在收到帧起始字符或字节后能够在每个位信元的标准中心产生采样脉冲。我们必须强调，一经同步（锁定），则在帧接受期间通常只要微小的调整。

从图3-9可推出关于NRZI编码，编码信号改变极性的最大速率是双极性编码与曼彻斯特编码的一半，如果位周期是T，NRZI编码最大速率是$1/T$，而双极性或曼彻斯特编码是$2/T$，最大速率称为调制速率。如2.2节中描述，每个方案的最大基频分量分别是$1/T$与$2/T$，这意味着，对于相同数据速率，双极性与曼彻斯特编码要求两倍的NRZI编码信号传输带宽，即调制速率越高，要求带宽越宽。

这个结果使曼彻斯特与差分曼彻斯特编码在LAN中被广泛地应用，而NRZI方案最早应用
116～117 于WAN。LAN在一个办公室或一幢大楼里运行，使用电缆线相对短，这意味着，即使以高比特率（例如10 Mbps或更高）运行，传输介质的衰减与带宽通常没有问题。

相反，在WAN双绞线电缆上通常采用高比特率，距离超过数公里，因此，编码方案（如NRZI）通常使用每位占有完整宽度脉冲。在WAN中所用编码方案的实例如图3-10所示。图3-10(a)中的实例都是差分编码信号，每个信元周期采用多级电平。使用差分编码意味着如果选取起始点极性不同，所示信号可能完全取反，使用多级电平意味着，引起可能的跳变扰动的任何差错是可识别的。

图3-10(a)中三种编码采用三电平码（$+V$，0，$-V$）表示二进制数据流。**传号交替反转**（AMI），由输入二进制数据流的二进制1（传号）触发信号跳变。这个方案的缺点是一长串0没有信号跳变。因此，每当出现一串0时，DPLL有关的电流可能丢失位同步。

为了克服这个局限性，由这个方案派生出若干个方案。例如，一个改进形式是信号跳变用0触发而不用1触发。这将在3.3.3节中讨论，对于面向位方案，每当有5个1组成的串发送时，自动地将0插入发送数据流，这就是说至少每5个位信元出现一次信号跳变，比特率可达每秒几百kb，信号跳变足够使DPLL保持时钟同步。与欧洲综合业务数据网（ISDN）有关的用户网络采用这种方法。 118

AMI编码方案的第二个派生称为**双极性8连0替换**（B8ZS），也常被采用。B8ZS除检测出8个零的串，在发送前将这个串编码为000VB0VB之外，其余都与AMI相同，其中B表示一个标准跳变（极性相反），V表示一个扰动跳变（极性相同）。0位的最大个数是7。这个方案在北美数字传输网中广泛采用。

第三个方案称为**3阶高密度双极性**（HDB3），用于北美以外数字传输网。采用任何4个零的串用3个零后跟一个扰动代替，而该扰动与前面跳变有相同的极性。因此，在这种情况下，首先的4个0组成的串用000V代替。基于这个基本规则，一个0的长串以每组4个0按同方式编码，导致信号引入DC平均电平，为避免这种长0串，每连续4个0的编码改为B00V，信号产生交替的极性。可推出HDB3无跳变最大0的个数是3。我们称HDB3与前面两个方案为**调制格式**。

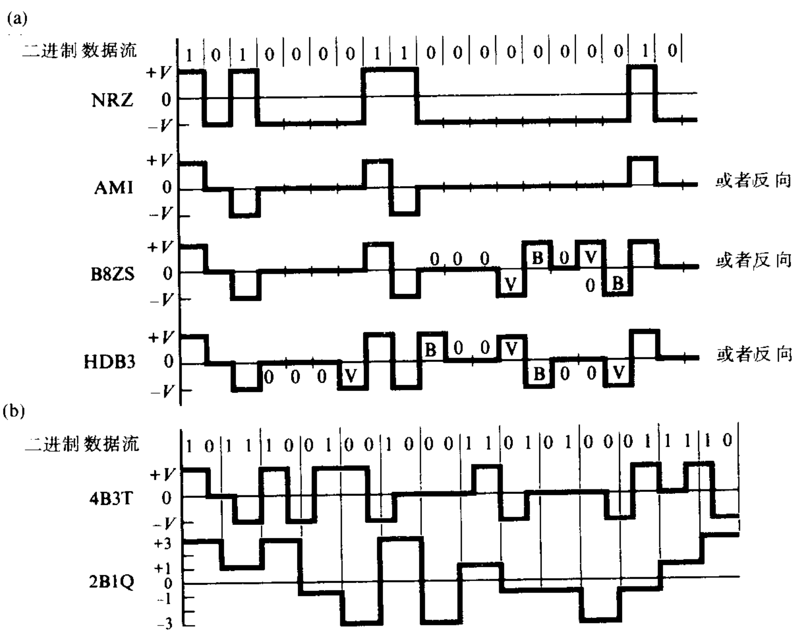


图3-10 WAN 采用的某些可选编码方案

(a) 二进制码 (b) 多级电平码

图3-10(b)所示的两个编码方案用于ISDN的接入电路。在第8章中，将看到以速率160 kbps采用双绞线可在数公里距离上工作。这两种编码都是**波特率压缩编码**，就是说单脉冲（时间单元）表示多于1位。主要优点是相邻脉冲间较小信号幅度变化降低了串扰。

这两种码归类为**mBnL码**，就是说m个输入位的序列用n个脉冲表示，每个有L级，其中$n<m$和$L>2$。因此，4B3T码也称为**改进监控状态43或MMS 43**，其中T指出为三级（三元），用符号+、–、0表示。因此，四个输入位用三个脉冲表示，而每个脉冲有三级。因此波特率是3/4，有一个1/4的**波特率压缩**。

对每4位输入序列发送3个符号编码从表3-1中4列选取1列表示。一般，在广域网上为了在线路上将发送与接收彼此分开，使用转换器。这就是说不存在直流（DC）通路。因此，必须保证发送信号平均（DC）电平为零。否则，接收器的零信号将变化。这个现象称为**DC漂移**，它的结果是接收器错误解释接收信号。

检查不同列的码，我们可看到每个码字的组合权重（平均电压）是可变的。在第1列中，例如，码字0–+的权重是0，而码字++–的权重是+1。显然，直接发送所有权重为+1的码字串，接收器平均信号电平将远离0。为了克服这个影响，每个二进制序列所用编码按平均信号电平
119 趋于零的方式从一列到另一列变动。

表3-1　4B3T编码模式

	1		2		3		4	
二进制序列	码	下一列	码	下一列	码	下一列	码	下一列
0001	0–+	1	0–+	2	0–+	3	0–+	4
0111	–0+	1	–0+	2	–0+	3	–0+	4
0100	–+0	1	–+0	2	–+0	3	–+0	4
0010	+–0	1	+–0	2	+–0	3	+–0	4
1011	+0–	1	+0–	2	+0–	3	+0–	4
1110	0+–	1	0+–	2	0+–	3	0+–	4
1001	+–+	2	+–+	3	+–+	4	–––	1
0011	00+	2	00+	3	00+	4	––0	2
1101	0+0	2	0+0	3	0+0	4	–0–	2
1000	+00	2	+00	3	+00	4	0––	2
0110	–++	2	–++	3	––+	2	––+	3
1010	++–	2	++–	3	+––	2	+––	3
1111	++0	3	00–	1	00–	2	00–	3
0000	+0+	3	0–0	1	0–0	2	0–0	3
0101	0++	3	–00	1	–00	2	–00	3
1100	+++	4	–+–	1	–+–	2	–+–	3

（注意：符号000解码为二进制序列0000。）

在一列中与每个码相关的是数字（1～4），它指出接着的码应该从下一列选取。在图3-10(b)中，最前面4位序列1011从列1中选取码（+0–），而下一列是1。因此下一序列1001从列1中选取（+ – +），而下列是2，等等。从表推出有27种不同码字。由于仅有16种可能输入序列（4位），码包含用于差错控制的冗余。表的内容是这样选择的，随机输入序列要求的平均带宽低于不用编码时的带宽。

图3-10(b)的第二个码称为**2B1Q**，Q指出四级（四元）脉冲，称为**四元**，每2位输入序列用一个4级脉冲发送。从图3-10(b)可看到，4级可用符号+3、+1、–1、–3来表示，它们关于0对称，相邻状态之间等距离。每对二进制数字中第1位表示符号（1=+，0=–），而第二位表示

幅度（1＝1，0＝3）。这种编码无冗余，但波特（信号）率与4B3T的3/4相比是1/2。

3. 混合方案

因为比特率增加，所以达到并保持时钟（位）同步增加了难度。虽然曼彻斯特与DPLL两套方案被广泛应用，但还有其他混合方案。一个典型方案如图3-11(a)所示。它采用曼彻斯特编码与DPLL相结合的方法。 120

DPLL保持本地时钟与来到的接收信号同步，然而曼彻斯特编码说明在每个位信元至少有一个信号跳变而不是如NRZI信号每5位有一个跳变。因此，本地时钟更可靠地保持同步。由图3-11(b)所示的波形组可推出本地（×2）时钟与进来的信号同步提供接收到（曼彻斯特）信号解码的可靠方法。曼彻斯特编码与NRZI比较付出的代价是增加的带宽要求。

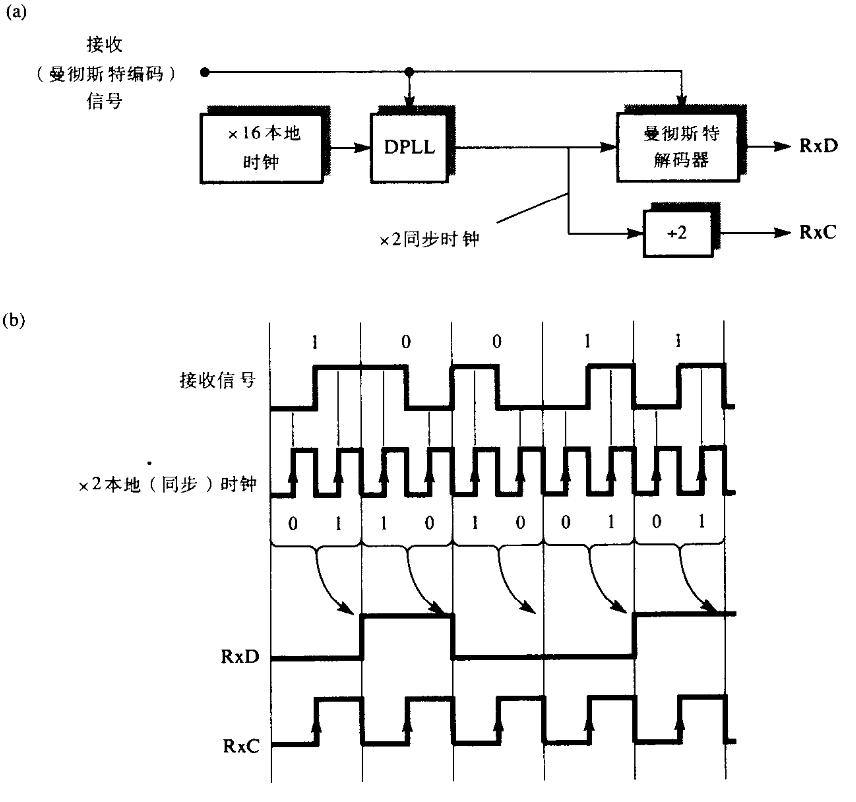


图3-11 用曼彻斯特编码与DPLL进行位同步
(a) 线路原理图 (b) 波形组

3.3.2 面向字符的同步传输

我们在3.3节开始时指出，同步传输控制方案有两种类型：面向字符的与面向位的。两者使用相同的位同步方法。两种方案的主要差别是实现字符同步与帧同步所用的方法。 121

面向字符传输最初用于字符块的传输，如ASCII字符文件。由于同步传输没有起始位与停止位，必须要有另外方法获得字符同步，为此发送器在发送字符块前需在每个字符块前增加两个或多个传输控制字符，称为**同步空闲**字符或**SYN**字符。这些控制字符有两个功能：首先，它们使接收器获得（或维持）位同步；其次，一旦做到这一点，它们会使接收器以正确的字符边界开始解释接收到二进制数据流——**字符同步**。一般结构如图3-12所示。

图3-12(a)表示达到帧同步（用面向字符同步传输），所用方法与在一对传输控制字符STX-ETX间封装字符块（帧内容）的异步传输完全相同。SYN控制字符用于使接收器在STX帧开始字符之前得到字符同步。一旦接收器得到位同步，它进入所谓的**搜索方式**。如图3-12(b)所示。

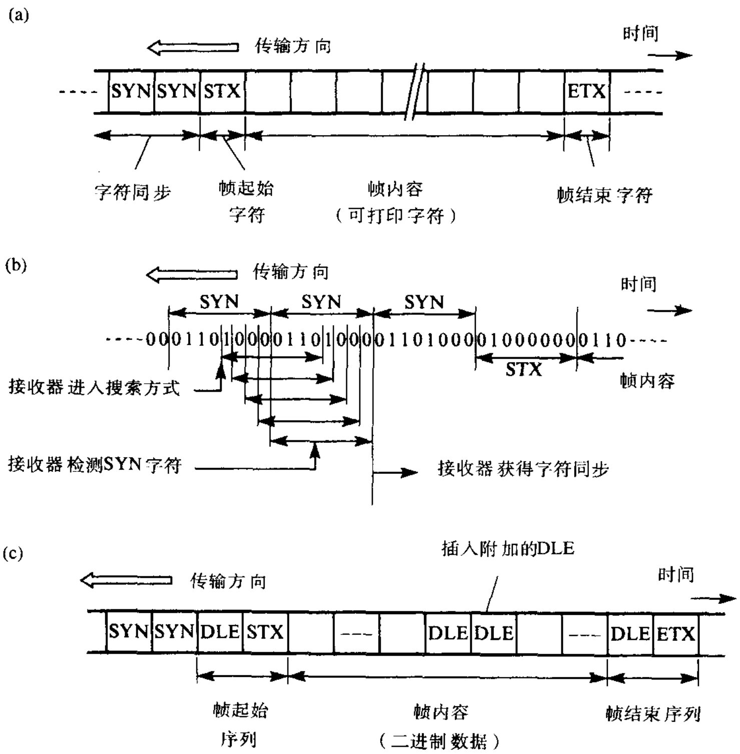


图3-12 面向字符传输

(a) 帧格式 (b) 字符同步 (c) 数据透明性（字符填充）

当接收器进入搜索方式，它开始随着接收到的新位以8位的窗口解释接收到的二进制数据
122 流。在这个方法中，随着接收每一位，检测最近的8位是否等于已知的SYN字符，如果不等于，接收下一位并重复检测；如果等于，则说明已找到正确字符边界，然后按照已接收的每个连续8位读出下面字符。

一旦字符同步（从而在正确位边界上读取每个字符），接收器开始处理每个后来接收的字符，搜索说明帧开始的STX字符。接收到STX字符，接收器继续接收帧的内容，当检测到ETX字符，终止这个过程。在点到点链路上，发送器通常恢复原状发送SYN字符允许接收器维持同步。另外，每次发送新帧，重复上述过程。

最后，如在图3-12(c)中看到，当发送二进制数据时，用前面描述的同样方法得到数据的透明性，在STX和ETX字符前外加一个DLE（数据链路转义）字符，无论何时检测到帧内容中的DLE，并插入（填充）一个额外的DLE字符。所以，在这种情况下，SYN字符位于第一个DLE字符前。

3.3.3 面向位的同步传输

为了帧同步需要一对字符用于帧的开始与结束，与附加DLE字符结合得到数据透明性，

就是说面向字符传输控制方案对于二进制数据是相对低效率的。再者，各种字符组所用的传输控制字符格式是不同的，尽管帧内容可以是纯二进制数据，可是该方案仅能用于一种类型的字符组。为了克服这些问题，更一般的称为**面向位传输**的方案作为由打印字符或二进制数据组成的帧传输的首选方案。三种面向位的传输控制方案如图3-13所示。它们的主要不同之处是方法中帧开始与结束的标志。

图3-13(a)中的方案最初用于点到点链路，帧起始与结束使用相同的8位模式01111110标志，称为**标志字节**或**标志模式**。我们用“面向位”的术语是因为接收器对收到的二进制数据按位逐个搜索帧起始标志，在帧内容接收期间搜索帧结束标志。因此，帧的内容原则上不需要由多个8位组成。

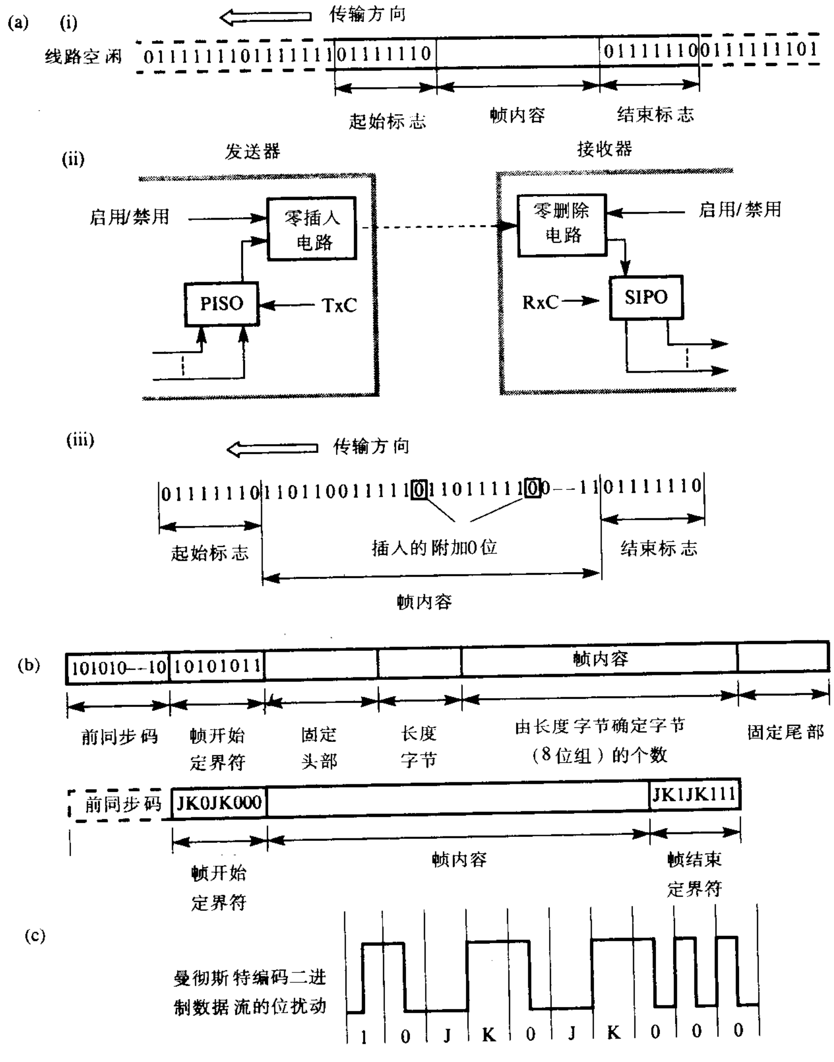


图3-13　面向位传输帧同步方法

(a) 标志　(b) 帧开始定界符与长度说明　(c) 位扰动编码

为了使接收器获得并维持位同步，发送器在帧起始标志前发送一串**空闲字节**(01111111)。

回忆有关NRZI编码，在空闲字节中的0使DPLL在接收器获得并保持时钟同步。接收到帧起始
123 标志后，读出接收到的帧内容并以8位（字节）为界解释，直到检测到帧结束标志为止，然后接收过程终止。

为了这个方案达到数据透明性，我们必须保证标志模式不在帧内容中出现。我们用**零位插入**或**位填充**技术实现这一点。在PISO寄存器的输出端安置执行这个功能的电路，如图3-13(a)所示，电路仅当发送帧内容期间由发送器启用，电路在启用时，检测发送内容，每当发现有5个连续的1时，则自动插入一个附加的二进制数0。这样，在起始标志与结束标志之间永远不
124 会出现01111110标志序列。

在接收器中，在SIPO移位接收器的输入端安置相似电路完成相反的功能。每当检测到5个连续1后面出现一个0时，电路自动从帧内容中移去（删除）它。通常，帧也包含帧结束标志前的附加差错检测数字。如同帧内容一样，它也服从位填充操作。填充位操作的实例如图3-13(a)(iii)所示。

图3-13(b)所示的方案用于某些LAN中，这些网络将在第6章中说明，传输介质是所有接入DTE（站）共享的广播介质。当一个站想发送一个帧，在帧的头部有一个接收站的地址（标识）的帧简单地发送到传输介质上。

为了使所有其他站都能获得位（时钟）同步，发送站在帧内容的前面放置一个称为**前同步码**的位模式。它由一串10组成，一旦位同步，接收器逐位地寻找接收到二进制数据流直到检测出已知的帧起始字节（10101011），即**帧开始定界符**。然后，跟一个固定头部，它包括想要接收的站地址。跟在固定头部后的两个**长度字节**指明帧内容中字节个数，因此，这个方案中，接收器简单计数字节（8位组）个数决定帧的结束。

图3-13(c)所示的方案也用于LAN，帧的起始与帧的结束两者都用于非标准编码信号，称为**位编码扰动**。例如，如果使用曼彻斯特编码代替在位信元中心发生的信号跳变，信号电平在完整位周期保留前一位相同电平（J）或相反电平（K）。

再者，为了检测到每个帧起始与结束，接收器逐位寻找接收到的二进制数据流，首先对帧的起始定界符（JK0JK000），然后对帧的结束定界符（JK1JK111）。因为J与K符号都是非标准位编码，帧内容不包括这样的码，并获得数据透明性。

在第5章与第6章中，当介绍不同网络类型与协议时，我们将详细讨论每个方案。

3.4　差错检测方法

在第2章指出，当数据在两台DTE之间传输时，传输线路（如PSTN）通常存在电噪声。电信号表示的二进制数据流受到线路邻近电子设备的电磁干扰引起改变。这就是说，表示1的
125 信号被接收器理解为0信号，或相反。为了保证目标DTE接收到信息与发送DTE所发出信息高概率地相同，接收器必须用某种方法以高概率推断接收信息包含差错的概率。再者，如果检测出差错，需要有一种机制获得（希望得到的）正确信息的拷贝。

为此，存在两种方法。

1）正向差错控制每个发送字符或帧包含附加（冗余）信息使接收器不仅能检测何时出现错误，还能确定错误出现在接收到二进制数据流的具体位置。通过改变所在位置的数据得到正确数据。

2）反馈（反向）差错控制每个发送字符或帧包含必要附加信息能使接收器检测何时发生

错误，但不能确定所在位置。采用重新传输控制方案，向另一方请求，发送希望得到的正确信息的拷贝。

实际上，为了得到可靠的正向差错控制，需要的附加位数随发送信息位数的增加而迅速增加。因此，反馈差错控制是本书中数据通信与连网系统所用的主要方法，正向差错控制的简要介绍在附录A中给出。

反馈差错控制分为两个部分：

1）实现可靠差错检测的技术。

2）重新传输控制方案的有效控制算法。

本节，我们讨论目前最常用的差错检测技术。可选的重新传输控制算法在第4章中讨论。

确定差错检测方案类型的两个因素是线路或电路的**误码率**（BER）与差错类型，即差错发生是作为单个位随机差错还是作为连续的位串差错，后者称**突发差错**。BER是一个位在规定时间区间受损的概率P。因此，BER为10^{-3}就意味着在规定时间周期内，平均10^3位中有1个位受损。

如果我们用异步传输方式发送一个字符（每个字符8位，加1个起始位和1个停止位），则一个字符受损的概率为$1-(1-P)^{10}$，约等于10^{-2}。换句话说，如果我们用异步传输方式发送字符块（比如说，每块125个字符，每个字符8位），则一个块（帧）包含错误的概率约为1。这就是说平均每块都有错误，必须重新传输。显然，对于这类线路来说这样的帧太长，为了得到可接受量必须减少帧长度。

差错的类型是重要的，我们将看到不同差错类型可以用不同的差错检测方案去检测。某些方案中所用位的个数决定被检测突发差错长度。三种最常用的方案是奇偶校验、块和校验以及循环冗余校验。我们将分别考察。 126

3.4.1 奇偶校验

异步传输与面向字符同步传输的最常用检测位差错方法是**奇偶校验位方法**。在每个字符传输之前，发送器先增加一个附加位（奇偶校验位）。奇偶校验位是组成发送字符的位组的函数。接收到每位字符后，接收器对接收到的字符执行相同的函数运算，并将结果与接收到奇偶校验位比较。如果它们相同，那么就认为没有出现差错，但如果它们是不同的，那么就认为传输出现差错。

为了计算一个字符的奇偶校验位，对字符中1的位数相加（模2），然后选择奇偶校验位的值，如果1位的总数（包括奇偶校验位本身）是偶数，称为**偶校验**，如果是奇数，称为**奇校验**。方案的原理如图3-14所示。

图3-14(d)中两个实例表明奇偶校验位方法仅能检测出单个（或奇数个）位差错，而不能检测出两个（或偶数个）位差错。

计算每个字符的奇偶校验位电路由一组连接的**异或**（XOR）门组成，如图3-14(c)所示。XOR门也称为**模2加法器**，因为两个二进制数字异或操作的输出等于不带进位的两个数字的和。图3-14(b)表明它的**真值表**。最低两位首先异或，然后这个门的输出与下一位异或，等等。最后门的输出是所要求的奇偶校验位，它在字符传输前输入到发送PISO寄存器。同样地，在接收方，重新计算奇偶校验位的值并与接收到的奇偶校验位比较。如果不同，说明已检测出传输错误。

用编码理论的术语**码字**描述有用数据位与附加差错校验位组成的消息单元。任意两个有效码字间最小的不相同位的数目，称为这种码的**编码汉明距离**。作为实例，考虑每个码字是7位

数据与1位奇偶校验方案。如果我们假定采用偶校验，在这个方案中连续的码字是：

00000000

00000011

00000101

00000110

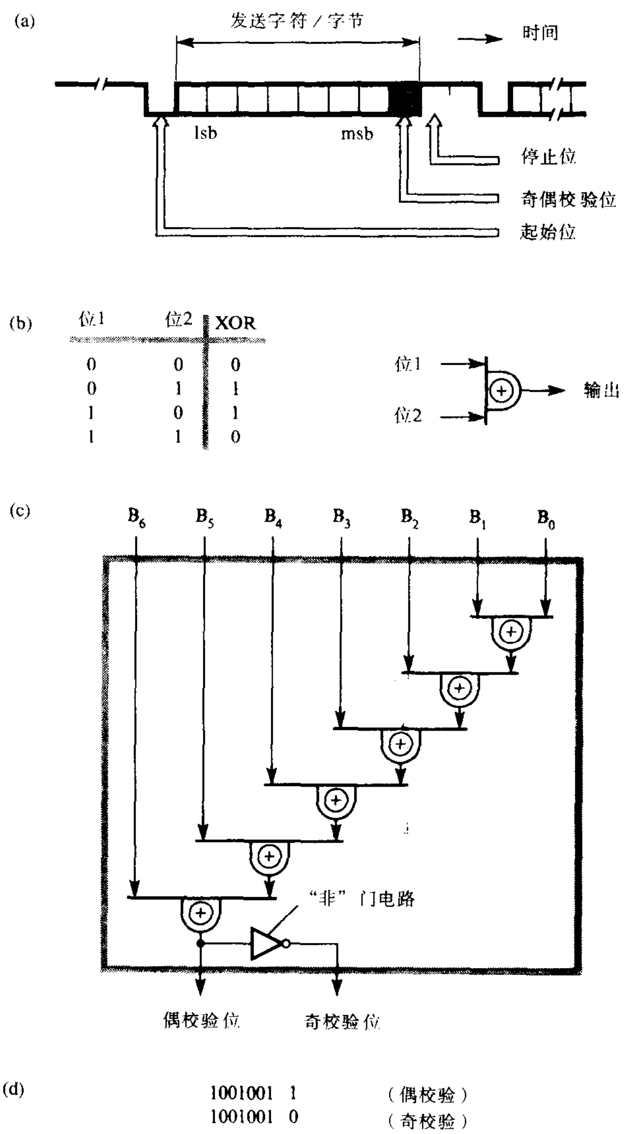


图3-14　奇偶校验位方法

(a) 字符中位置　(b) XOR门真值表与符号　(c) 奇偶校验位生成电路　(d) 两个实例

从上面列出码字可推出，由于每个有效码字至少有两位不同，汉明距离为2。这就是说该
127 方案不能检测两位的差错，这是由于形成的（损坏的）位模式不同，但是有效码字。它的确能检测一位差错，因为码字中如果有一位损坏，将产生一个无效码字。

3.4.2 块和校验

发送字符块时，一个字符由于一位错引起块差错的概率增加。一个块具有差错的概率称为**数据块差错率**。当发送字符块（帧）时，我们可扩充从每个字符（字节）的单奇偶校验位 128
获得的差错检测的能力，使用从帧中整个字符（字节）块计算得到的附加奇偶校验位组。这种方法，对帧中每个字符（字节）如前面那样指定一个奇偶校验位（**横向校验或行校验**）。另外，在整个帧中，一个额外位用于计算每个位的位置（**纵向校验或列校验**）。对于每列形成的奇偶校验位的集合称为**块（和）校验字符**，该字符的每一位是对应列所有位的模2和。图3-15中的实例用行奇偶校验位组的奇校验和列奇偶校验位组的偶校验，假定帧由可打印字符组成。

(a)

P_R	B_6	B_5	B_4	B_3	B_2	B_1	B_0	
0	0	0	0	0	0	1	0	= STX
1	0	1	0	1	0	0	0	帧内容
0	1	$\boxed{0}$	0	0	$\boxed{1}$	1	0	
0	0	1	0	0	0	0	0	
1	0	1	0	1	1	0	1	
0	1	0	0	0	0	0	0	
1	1	$\boxed{1}$	0	0	$\boxed{0}$	1	1	
1	0	0	0	0	0	1	1	= ETX
1	1	0	0	0	0	0	1	= BCC

横向（行）奇偶校验位（奇校验）

纵向（列）奇偶校验位（偶校验）

传输方向 ⇓

□ = 不能检测差错组合的实例

P_R = 行奇偶校验位

(b)

```
        发送方                                   接收方

    0 0 0 0 0 1 0 ┐                            0 0 0 0 0 1 0
    1 0 1 1 0 1 1 │ 实例内容                   1 0 1 1 0 1 1
    1 1 0 1 1 0 0 │                            1 1 0 1 1 0 0
    0 0 0 0 0 1 1 ┘                            0 0 0 0 0 1 1
[1] 1 0 0 1 1 0 0                              0 1 1 0 0 1 0
 └──────────────► 1                        [1] 1 1 1 1 1 1 0
    1 0 0 1 1 0 1  = 反码求和                   └──────────────► 1
          ⇓ 取反                                1 1 1 1 1 1 1  = 反码的零
    0 1 1 0 0 1 0  = BCC
```

图3-15　块和校验实例

(a) 行与列的奇偶校验位　(b) 反码求和

我们从这个实例能推出，虽然一个字符中的两个差错不能从行奇偶校验检测出，但能从对应列奇偶校验检测出。当然，这仅是两个位差错不会同时发生在同一列是对的。显然，这 129
种发生的概率比单个字符发生两位错的概率小。块检验和的使用大大改善了方案的差错检测性能。

方案使用反码求和作为块检验和的基础，代替原来的模2求和，反码求和的原理如图3-15(b)所示。

在这个方案中，被传递块中的字符（字节）是作为无符号二进制数据来处理。首先使用反码算术相加。然后，对所得和的所有位求反，这个数作为块校验字符（BCC）。接收器计算块中所有字符反码的和（包括块校验字符），如果没有差错，其结果应该是零。记住，反码算术，使用循环进位，即将最高位的进位输出加到现存二进制和中。反码中的零有两种形式，或者是全0或者是全1。

我们从图3-15(b)推出这个方案的差错检测性能比模2求和方法要好。我们将在第9章和第10章看到。由于反码求和容易计算，作为差错检测方法在仅要求用软件执行的差错检测操作中有一些应用。

3.4.3 循环冗余校验

前面两个方案最适合用于随机的单个位差错的检测，然而当出现突发性差错时，我们必须使用更加严密的方法。一个突发差错是以一个错误位开始和结束的，然而中间位可以受损也可以不受损。因此，一个突发差错要规定两个接连错误位之间位的个数包括两个错误位。进一步，当确定一个突发差错长度，突发差错的最后一个错误位与下一个突发差错的第一个错误位之间必须有大于或等于B个正确位的间隔，其中B是突发差错的长度。两个不同长度突发差错如图3-16所示。注意：第1个错误位与第3个错误位不能用来定义单个11位突发差错，这是因为差错在下一个11个位之内发生。

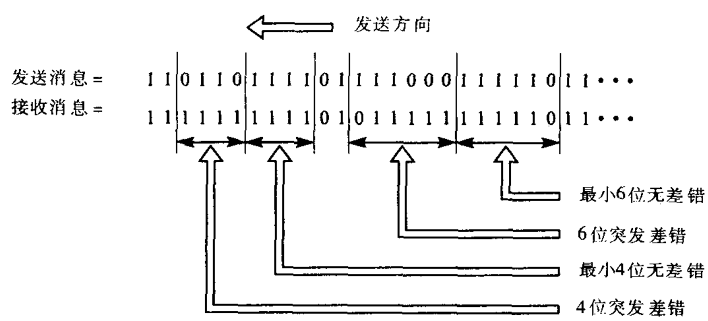


图3-16 突发差错实例

校验位或它派生出的块校验和不能对突发差错提供一个可靠检测方案。在这样的情形下，最常用选择是基于**多项式编码**使用。多项式编码用于帧（或块）传输方案，基于帧的内容，为每个发送帧计算一组校验数字，发送器将它们附在帧的结尾处。然后，接收器对整个帧与
130 校验数字执行相同计算。如果没有差错，应该得到已知的结果；如果答案不同，说明有差错。

选择每个帧校验位的个数要适合检测传输差错的类型，最常用的是16位与32位。计算校验数字称为**帧校验序列**（FCS）数字或**循环冗余校验**（CRC）数字。

多项式编码的基础数学理论超出本书的讨论范围，如采用模2运算，方法实质上是利用下列二进制数的性质。设：

$M(x)$是k位二进制数（被发送的消息）

$G(x)$是$(n+1)$位二进制数（除数或生成因子）

$R(x)$是n位二进制数，其中$k > n$（余数）

那么如果：

$$\frac{M(x)\times 2^n}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$，其中$Q(x)$是商

$\frac{M(x)\times 2^n+R(x)}{G(x)}=Q(x)$，假定模2运算

将 $\frac{M(x)\times 2^n}{G(x)}$ 的表达式代入第2个等式得到：

$$\frac{M(x)\times 2^n+R(x)}{G(x)}=Q(x)+\frac{R(x)}{G(x)}+\frac{R(x)}{G(x)}$$

因为任何数加上它本身模2的结果是零，也就是说余数为零，所以上式等于$Q(x)$。

利用这个结果，整个帧内容$M(x)$附加一组0，0的个数等于生成FCS数字位的个数（等价于消息用2^n乘，其中n是FCS数字位的个数），用第2个二进制数，即**生成多项式**$G(x)$做模2除，其中$G(x)$是n+1位。除法运算等价于对帧中每位依次并行执行异或操作。然后，余数$R(x)$是FCS，它附在信息数字的尾部被发送。同样地，接收方接收到的包括FCS数字的二进制数据流再用同一 131
生成器多项式 $\frac{M(x)\times 2^n+R(x)}{G(x)}$ 去除。如果没有差错，则余数为0；如果有差错，则余数不为0。

实例3-3

一组8位消息块（帧）利用CRC差错检测通过数据链路发送。使生成器多项式为11001。用实例说明：

(a) FCS生成过程

(b) FCS校验过程

解：

消息11100110的FCS生成如图3-17所示。首先由于FCS是4位，用4个0附加于消息的尾部， 132
等价于用2^4乘消息，然后用生成器多项式（二进制数）去除（模2）。模2的除法等价于对被除数每位依次并行执行异或操作。有关模2运算，可以对每个部分余数做除法，只要两个数的长度相等，即两者的最高有效位都为1，而不考虑数的相对大小。结果的4位余数（0110）是FCS，然后当发送时将它附于原始消息的尾部。商是不用的。

接收器接收到全部二进制数据流，用发送器使用的同一个生成器多项式去除。两个例子如图3-17(b)所示。第一个例子，假定没有差错出现，所以余数为0，商不用。第二个例子，假定发送的二进制序列的尾部有一个4位突发差错，因此，结果的余数不为0，说明发生传输差错。

生成器多项式的选择是重要的，因为它确定检测差错的类型。假定发送帧$T(x)$是

110101100110

差错模式$E(x)$是

000000001001

即1所在位表示差错。因此，进行模2运算：

接收到的帧 = $T(x)+E(x)$

现在 $\frac{T(x)+E(x)}{G(x)}=\frac{T(x)}{G(x)}+\frac{E(x)}{G(x)}$

由于$T(x)/G(x)$没有余数，因此仅当$E(x)/G(x)$有余数时，检测到差错。

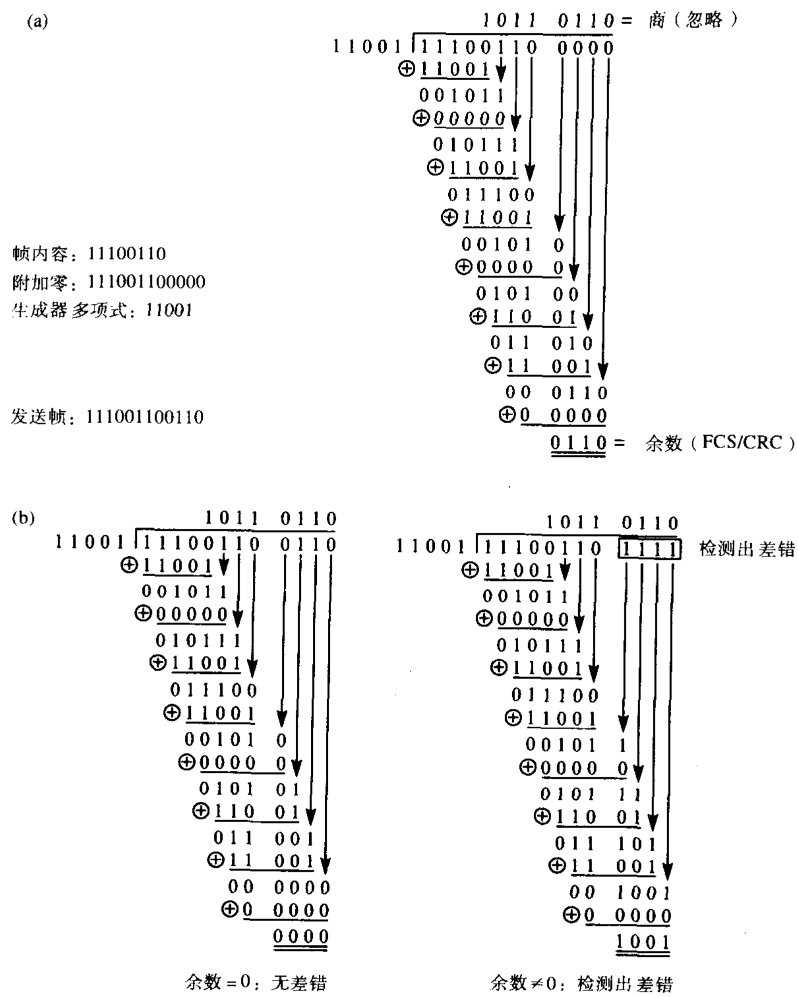


图3-17 CRC实例

例如，所有$G(x)$至少具有3项（1位），而$E(x)/G(x)$对所有单个位差错与所有双位差错执行模2运算都产生余数，因此能检测出单个位差错与双位差错。相反，同$G(x)$一样长的突发差错可能是$G(x)$的倍数，因此余数为零，不能检测出差错。

总之，R位的生成器多项式$G(x)$可检测：

- 所有的单个位差错
133 • 所有的双位差错
- 所有的奇数位差错
- 所有突发差错长度$<R$
- 大多数突发差错长度$\geq R$

表示生成器多项式的标准方法是用X的幂次系数为1的位置表示。例如实际使用的CRC实例是：

$$\begin{aligned}
\text{CRC}-16 &= X^{16}+X^{15}+X^{2}+1\\
\text{CRC}-\text{CCITT} &= X^{16}+X^{12}+X^{5}+1\\
\text{CRC}-32 &= X^{32}+X^{26}+X^{23}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}\\
&\quad +X^{5}+X^{4}+X^{22}+X+1
\end{aligned}$$

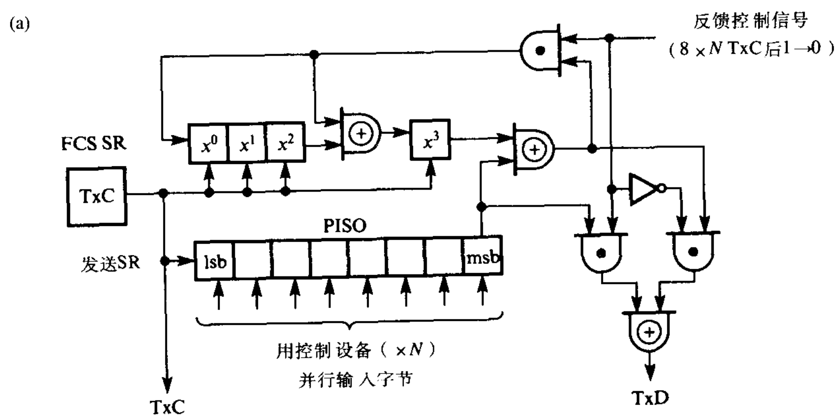
因此，CRC-16等价的二进制数形式：

1 1000 0000 0000 0101

关于这个生成器多项式，在生成FCS前，用16个0附在帧内容的后面。然后，后者应该是16位余数。CRC-16将检测出所有小于16位的突发差错与大多数大于或等于16位的突发差错，CRC-16与CRC-CCITT都广泛用于WAN，而CRC-32用于多数的LAN。

虽然，要求做多次除法（模2）可能较复杂，实际上用硬件或软件很容易做到。为了说明这一点，图3-17方案的硬件实现在图3-18(a)给出。

在这个实例中，因为我们打算生成四位FCS数字，所以仅需要4位移位寄存器表示生成器多项式的位x^3、x^2、x^1和x^0。通常称这些位为生成器的**活动位**。对于这个生成器多项式，x^3与x^0的系数是1，而x^2与x^1的系数是0。移位寄存器单元x^1与x^2的新状态简单采用单元x^0与x^1的状态；移位寄存器单元x^0与x^3的新状态由前面数字反馈通过异或电路的状态决定。



TxC 脉冲	发送SR								FCS SR			
	lsb							msb	x^0	x^1	x^2	x^3
0	0	1	1	0	0	1	1	1	0	0	0	0
1	0	0	1	1	0	0	1	1	1	0	0	1
2	0	0	0	1	1	0	0	1	0	1	0	0
3	0	0	0	0	1	1	0	0	1	0	1	1
4	0	0	0	0	0	1	1	0	1	1	0	0
5		0	0	0	0	0	1	1	0	1	1	0
6			0	0	0	0	0	1	1	0	1	0
7				0	0	0	0	0	1	1	0	0
8					0	0	0	0	0	1	1	0
9						0	0	0		0	1	1
10							0	0			0	1
11								0				0

时间

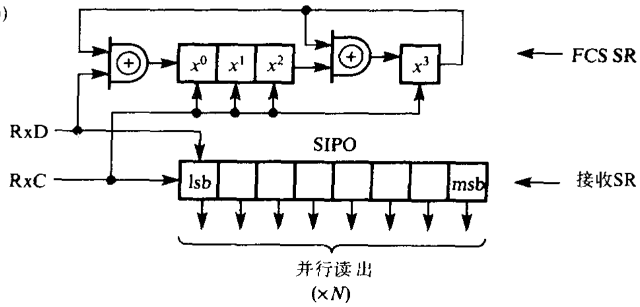
SR = 移位寄存器

TxD

图3-18 CRC硬件方案

(a) 生成结构 (b) 校验检测

(b)



RxC	RxD	接收SR								FCS SR			
		lsb							msb	x^0	x^1	x^2	x^3
0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1								1	0	0	0
2	1	1	1							1	1	0	0
3	0	1	1	1						1	1	1	0
4	0	0	1	1	1					0	1	1	1
5	1	0	0	1	1	1				1	0	1	0
6	1	1	0	0	1	1	1			1	1	0	1
7	0	1	1	0	0	1	1	1		0	1	1	1
8	0	0	1	1	0	0	1	1	1	1	0	1	0
9	1	从控制设备读出字节								0	1	0	1
10	1									0	0	1	1
11	0									0	0	0	0
12										0	0	0	0

时间

余数＝0

图3-18 （续）

电路操作如下：FCS移位寄存器清除，帧中的第一个8位字节并行输入PISO发送移位寄存器。然后按发送器时钟TxC所决定速率移位输出到传输线上，首先是最高有效位。按时间同步，同样的二进制数据流与FCS移位寄存器单元x^3异或通过反馈回路到FCS移位寄存器的输入端。随着后来的每个8位输入到发送移位寄存器，并按位串行方式发送到线路上，此过程重复。最后，帧中最末字节完成输出后发送移位寄存器输入零，反馈控制信号由1变0，所以FCS移
134 位寄存器现在的值是计算出的余数，接着帧内容发送到传输线上。

在图3-18(a)中，发送的内容与FCS移位寄存器的内容假定刚好是单字节帧（N=1），因此它对应于前面图3-17的实例。在图3-18中，每次移位（发送时钟）脉冲后，发送的内容与FCS移位寄存器的内容都表示出。发送的二进制数据流如虚线框中所示。

对应的接收器硬件与发送器所用的相似，如图3-18(b)所示。接收数据（RxD）在位信元中心（或曼彻斯特编码的最新位）采样（移入）送入SIPO接收移位寄存器。正如以前所说，
135 二进制数据流与x^3的异或以及反馈输入到FCS移位寄存器，这些也需要时间同步。每8位字节被接收到，由控制设备读出。帧的内容只包含单字节数据。

通常图3-18中的硬件是将输入控制电路与面向位传输结合在一起。然而，在某些情况下，CRC优于面向字符传输的块和校验。在有些情况下，CRC通常由控制设备的软件实现而不用硬件。我们可以从图3-19中直接看到软件伪代码。

代码假定8位生成器多项式（除数）以及特定帧（STE、ETX）存储在一个数组中。同样代码可用于CRC生成与检验，生成数组尾部包含全0组成的字符/字节。 136

假设要发送的特定帧（包括尾部的0）或接收帧存储在字节数组缓冲[1..count]中。9位除数中的8个活动位存储在16位整数CRCDIV的最高有效8位中。以下函数将计算并返回8位CRC。

```
function CRC : byte;
var      i, j : integer;
         data : integer

begin    data := buff[1] shl 8;
         for j := 2 to count do
             begin
                 data := data + buff [j];
                 for i := 1 to 8 do
                     if ((data and $8000) = $8000) then
                     begin data := data shl1;
                           data := data xor CRCDIV; end
                     else  data := data shl 1;
             end;

         CRC := data shr 8;
end;
```

图3-19 8位CRC伪代码计算/检验

3.5 数据压缩

直到目前为止，我们假设发送帧（块）的内容都是由固定长度的字符或字节串的原始数据组成。虽然，许多数据通信应用是这种情况，但也有另外一些情况，在传输之前原始数据经过压缩。在应用中，大部分线路包括公用传输设备，如PSTN，收费是基于时间（持续时间）与距离。因此，对于一个特殊呼叫，如果发送每个数据块的时间能减少，则它自动降低呼叫费用。

例如，假定在PSTN上以4800 bps发送数据。与这个呼叫相关的数据发送所需时间是20分钟。显然，使用数据压缩，我们能减半发送数据总量，可获得50%传输费用的节约。这就等于说，用压缩4800 bps调制解调器与无压缩9600 bps调制解调器可以获得相同的性能。

实际上，我们可采用一系列的压缩算法，每一种适用于某一特殊数据类型。有些调制解调器，通常称为**智能调制解调器**，现在提供一种**自适应压缩**特性，它能选择一种压缩算法适应正在被发送的数据。在本节，我们将描述某些较常用数据压缩算法类型。

3.5.1 压缩十进制数

当发送仅由数字字符组成的帧时，我们可将每个字符的位数从7位减少到4位得到较大的节约，即简单使用二进制编码的十进制码（BCD）取代ASCII。我们可从图3-1(b)的ASCII代码表看到10个数字（0～9），它们三个的最高位都是011。通常，需要用这三位（011）在代码组中区别数字与字母（和其他）字符。如果数据仅是数字，这三位显然是冗余的，不必发送。如图3-20(a)所示。 137

正如我们现在所见，如果采用面向字符传输，不可能在STX与ETX控制字符之间简单插入压缩二进制编码十进制数字，因为数字对0，3将被理解为ETX。因此，在数据开始处用一个（已知的）控制字符码说明后跟的是压缩十进制数。然后接下来的第二个字符（字节）说明后跟数字的个数，在同一列的两个字符如“：”与“；”通常用作小数点和空格。

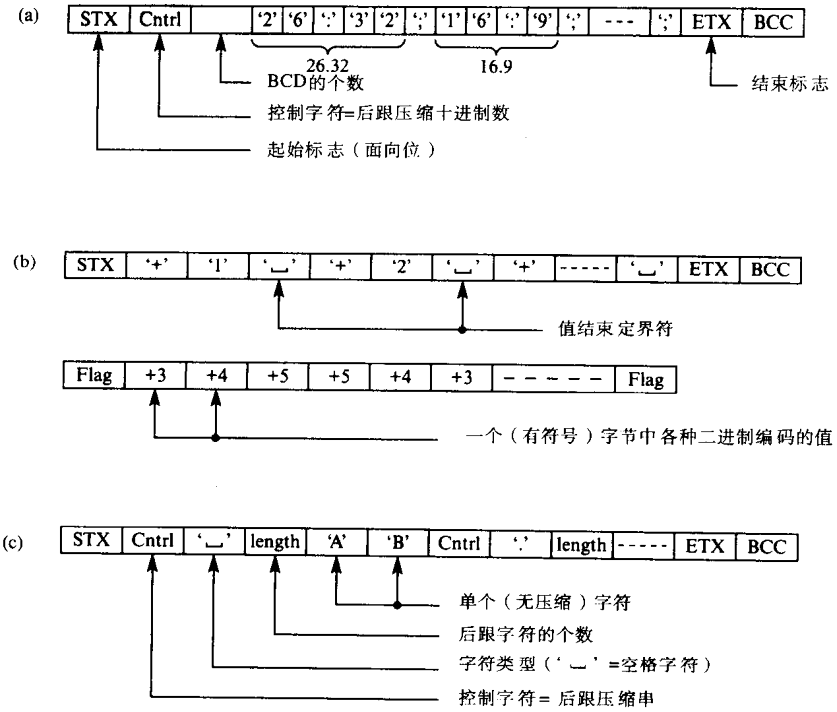


图3-20　压缩算法实例

(a) 压缩十进制数　(b) 有关的编码　(c) 字符压缩

3.5.2　相对编码

当发送的数值数据在连续值之间只有很小差别时，另一个压缩十进制数仅发送参考值与每个值的差值大小。这称为**相对编码**，它在数据录入应用中特别有效。

例如，如果我们正在遥控河流水位，一般在规定时间间隔记录一次水位，并在传送前存
138 储（录入）这些数据。当我们已得到一组预置数据时，例如通过PTSN与自动**拨号调制解调器**发送这组数据到中央监控站点。为了达到发送数据所需时间最少，代替发送水位的绝对值而发送差值。这种方案帧的内容如图3-20(b)所示。可看出得到的节约是表示不同差值格式的函数。一般，用面向位协议的二进制值可达到最大节约。

3.5.3　字符压缩

相对编码方案的一种变型，我们可用更一般方法压缩其他字符如图3-20(c)所示。一般，当发送的帧由可打印字符组成时，在帧序列中时常存在着同一字符重复，例如空格字符。发送器的控制设备在发送前扫描帧的内容，如果有三个或者多个字符连成字符串，则用三字符序列代替这些串，如图所示。

序列由（已知的）控制字符组成，指明后跟压缩串、字符类型和串中字符个数的计数。一般，计数是二进制形式，但由于前面是一个已知控制字符，接收器能区别计数值3与控制字符ETX。在这个方案中，任何字符能被压缩。接收器检测到压缩控制字符，容易读出下面的字符类型与计数值，并在这一点插入这些字符（适当个数）到接收帧。这个方案是称为运行长度编码的更一般编码技术的一个实例。

3.5.4 霍夫曼编码

霍夫曼编码利用帧中所有代码出现次数不同的性质，例如在一个由字符组成的帧中，某些字符出现次数比其他字符要多。取代每个字符用固定位长编码方案，我们用一个不同编码方案，最常用字符编码比不常用字符位数少，因此，它是一种**统计编码**。由于每个字符的位数是变化的，我们必须使用面向位的传输。

首先，分析发送的字符串，确定字符类型与它们的相对频率。然后，编码操作包含建立一棵具有短分支（实际中是码字）的非平衡树，非平衡树的程度是字符出现次数的函数。分散越广，树越不平衡。形成的结果树称为**霍夫曼编码树**。

霍夫曼（编码）树是**二叉树**，分支赋予值0或值1。实际上，树根的几何顶部一般称为**根结点**。而分散出分支的点称为**分支结点**，分支的终止点称为**叶结点**。在叶结点分配被编码的符号。霍夫曼树的实例如图3-21(a)所示。该例是对应字符串AAAABBCD的编码。 139

每一个分支结点分出赋予二进制值0或1的新分支：在左边分支为0而在右边分支为1。每个字符的码字是从根结点出发到每个叶结点的路径并形成经过每个分支轨迹的一串二进制值所确定。我们从编码推出，发送整个串AAAABBCD共花费

$$4\times1+2\times2+1\times3+1\times3=14\text{位}$$

我们用每个码字位的个数乘它出现的概率的总和计算每个码字位的个数平均值即

$$1\times0.5+2\times0.25+3\times0.125+3\times0.125=1.75\text{（位/每个码定）}$$

每个传送信息串的码字理论上最小平均位数称为信息的**熵*H***。利用香农定律，计算出

$$H=-\sum_{i=1}^{n}P_i\log_2 P_i\text{（位/每个码字）}$$

其中 n 是信息中字符个数，P_i 是信息中第 i 个字符出现的概率。因此，对于信息AAAABBCD，每个码字最小平均位数由下式给出：

$$\begin{aligned}H&=-(0.5\log_2 0.5+0.25\log_2 0.25+0.125\log_2 0.125+0.125\log_2 0.125)\\&=1.75\text{（位/每个码字）}\end{aligned}$$

在这种情况下，结果与使用霍夫曼编码相同。

用7位ASCII码字，发送整个信息串要求8×7＝56位。它比霍夫曼编码要求的14位多得太多。

在实际中，关于霍夫曼编码这个比较是不完善的。接收器必须知道发送的缩减的字符组与相应的码字，即它们出现的次数或霍夫曼树。一个较好的比较是考察普通二进制编码需要位的个数：4个字符（A–D）要求每个字符编码是2位，所以发送8个字符共需要16位。显然，霍夫曼编码的节约是令人满意的。一般当发送字符的频率分布很广并具有长的字符串时霍夫曼编码是高效率的。相反地，它对传送二进制编码的数据是不适用的，因为8位字节通常出现的频率相同。

图3-21(a)说明如何确定霍夫曼树，我们必须附加每个字符出现次数的信息，如图3-12(b)所示。字符在一列中按递减（权）次序排列。我们如下确定树。 140

首先在表的底部的两个叶结点（C1和D1）分别赋予一个分支结点的（1）与（0）分支，然后两个叶结点用一个分支结点代替，分支结点的权是两个叶结点权的和，为2。形成一个包含新结点、以及第一列剩余结点组成的新列。再重新排列它们正确权的顺序。这个过程重复直到仅有两个剩余结点为止。

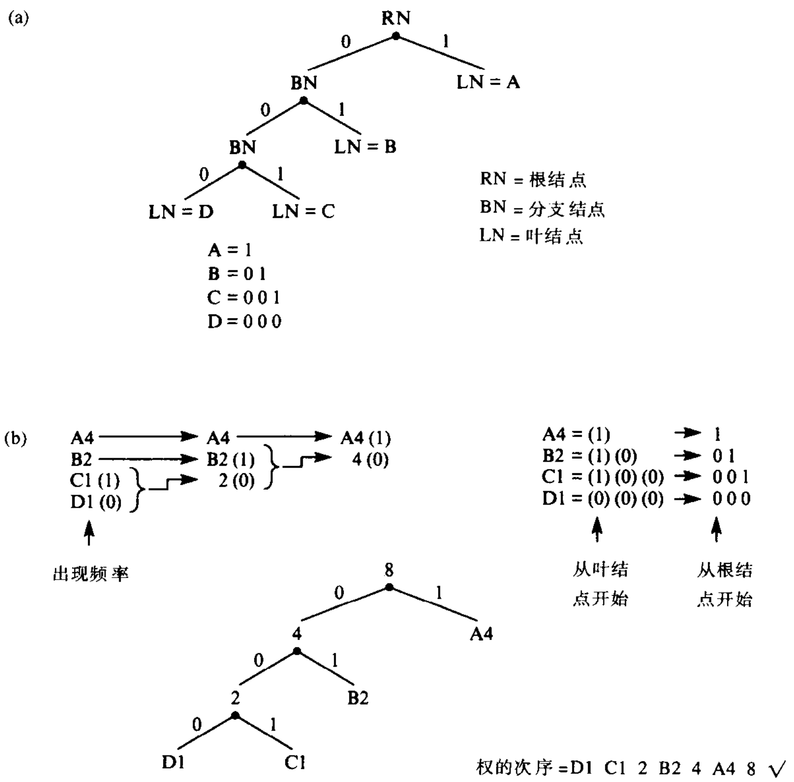


图3-21 霍夫曼编码树结构

(a) 具有编码的最后树 (b) 树的形成

为了求出每个字符的结果码字，我们从第一列字符开始，然后列出分支编号0或1，如同上面所做那样。因此，字符 A 在最后一列第一个（仅有一个）分支编号为1，而字符C在第一列是（1），然后在分支结点2是（0），以及最后在分支结点4是（0）。但是，码字从根结点开始而不是叶结点，因此实际码字是这些数的反。然后从码字集合很容易构造出霍夫曼树。

我们用树开始处是最小权并从左到右、从底到顶列出所有叶结点和分支结点结果权的方法，

141 检验这是优化树（因此也是优化码字集）。如果结果列表按权次序递增，则码字是优化的。

实例3-4

两台计算机通过PSTN发送一系列信息，信息刚好由从A到H的字符组成，统计结果表明每个字符出现的概率如下：

A与B的概率 = 0.25，C与D的概率 = 0.14，E、F、G与H的概率 = 0.055

(a) 用香农公式求每个字符的最小平均位数。

(b) 用霍夫曼编码推导码字组，构造对应的霍夫曼代码树并证明这是最小集。

(c) 求码字组中每个字符平均位数，并与下列比较：

(i) 香农值

(ii) 定长二进制码字

(iii) 7位ASCII码字

解：(a) 由香农公式，

$$\text{熵 } H = -\sum_{i=1}^{8} P_i \log_2 P_i \quad （位/每个码字）$$

得

$$H = -\left(2(0.25\log_2 0.25) + 2(0.14\log_2 0.14) + 4(0.055\log_2 0.055)\right)$$
$$= 1 + 0.794 + 0.921 = 2.715\ (\text{位/每个码字})$$

(b) 用霍夫曼编码推导码字组如图3-22(a)所示，字符首先按权的次序排列，并在列的底部两个字符赋予（1）与（0）分支。注意，在这样情况下，当两个结点合并时，结果分支结点的权（0.11）大于两个字符E与F的权（0.055）。此时分支结点插入第二列，高于字符E与字符F的位置。然后重复同样过程直到列中仅有两项剩余为止。

对应于码字组的霍夫曼编码树如图3-22(b)所示。我们可看出，因为所有叶结点与分支结点按数字次序递增，所以这是优化树。

(c) 使用霍夫曼编码每个码字平均位数是

$$2(2\times 0.25) + 2(3\times 0.14) + 4(4\times 0.055) = 2.72\ (\text{位/每个码字})$$

即是香农值的99.8%。

采用定长二进制码字：A到H共有8个字符，因此每个码字3位，即是霍夫曼值的90.7%。 142

采用7位ASCII码字：每个码字7位，即是霍夫曼值的38.86%。

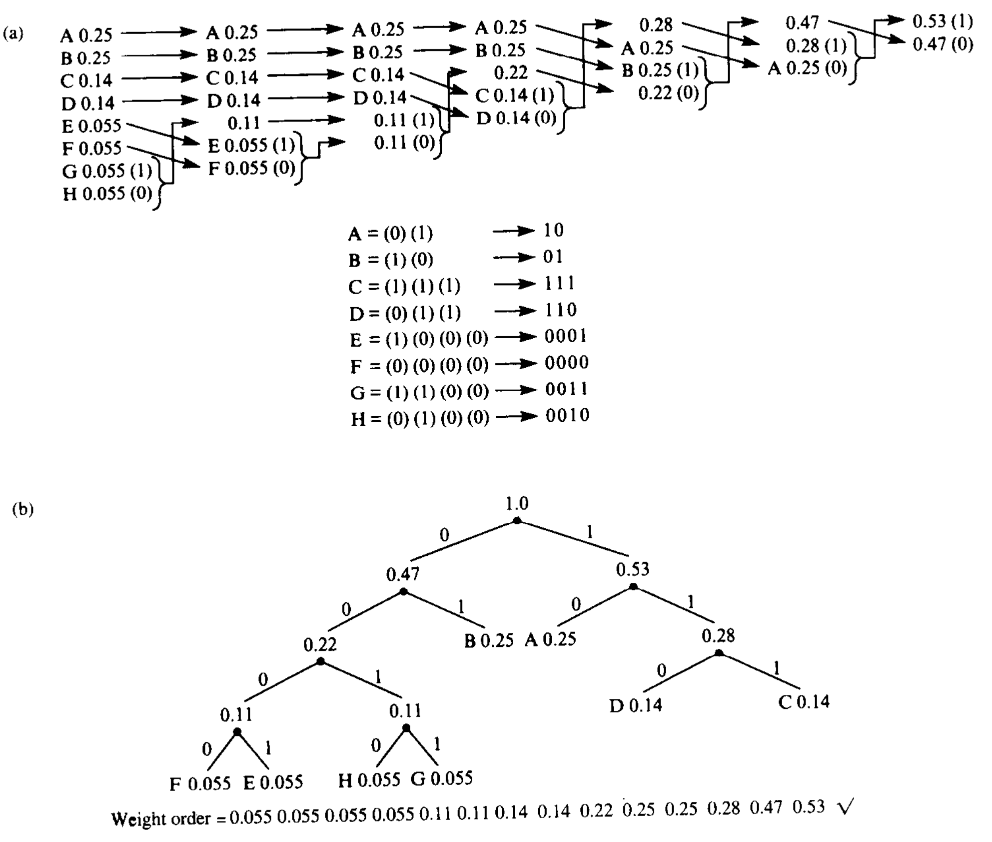


图3-22 霍夫曼编码实例

(a) 码字生成 (b) 霍夫曼编码数

因为每个字符的编码具有可变的位数，接收二进制数据流必须用面向位方式解释（解码），而不能用固定8位界限。由于编码过程中，赋予位的次序，因此霍夫曼码字具有独特

性质，一个短码字决不是另一长码字的开始部分。如果说011是有效码字，则011不可能是任何长码字的开始部分的模式。在前面图3-21与图3-22的实例中，考察码的推导就能确信这个性质。

这个性质称为**前缀性质**，就是说接收到二进制数据流可简单用逐位递归搜索直到找到有
143 效码字为止的办法来译码。解码算法的流程图在图3-23(a)给出。算法假定在接收器有一个可用的码字表并有相应的ASCII码字。接收到二进制数据流存入变量BITSTREAM，CODEWORD是用来存放每个码字位的变量，该变量的内容逐步构造。从流程图可看出，一旦识别一个码字，对应的ASCII码字写入RECEIVE_BUFFER。过程重复直到接收串中所有都被处理完。对应于图3-21中求得的码字组，一个解码串的实例在图3-23(b)中给出。

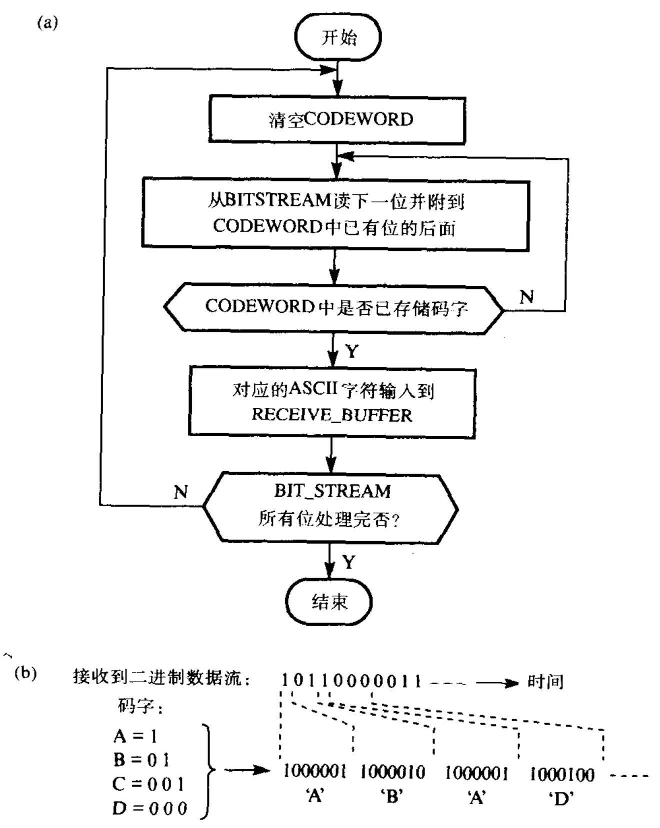


图3-23 假定码字按图3-21求出，对接收到二进制数据解码

(a) 解码算法 (b) 实例

由于霍夫曼编码树（码字）对于不同发送字符集合是可变的，接收器执行解码操作，它必须知道与发送数据相关的码字。这可用两种方法做到，一种在发送数据之前，发送与下次数据集有关的码字，另一种方法下，接收器提前知道使用什么样的码字。

第一种方法，由于能改变码字以适应发送数据类型，产生一种适应的压缩形式，它的缺
144 点是每当发送一种新的数据类型时，发送新的码字（和对应字符）组要有开销。另一种方法，接收器要有一个或多个不同码字组而发送器向接收器指明（通过同意消息）下次发送数据集时采用什么样的码字组。

例如，因为有发送由字处理程序生成的文本文件（通常包含文本信息）的共同要求，所以，在通常书写文本中，对英文字母字符的出现频率已作过详细的统计分析。这种信息已用于建立字母表的霍夫曼编码树。如果正在发送这样的数据，则发送器与接收器自动地使用这个码字组。其他常用数据组已用同样方法分析，进一步的实例查阅本书最后的参考文献。

3.5.5 动态霍夫曼编码

基本霍夫曼编码方法要求发送器与接收器两者都要知道正在发送数据的码字表。另一种方法允许发送器（编码器）和接收器（解码器）动态地构造霍夫曼树（因此，也构造码字表），作为发送的/接收的字符，这就是**动态霍夫曼编码**。

对于这种方法，如果被发送的字符出现在树中，则它的码字被确定，按常规方法发送。如果被发送的字符不在树中，即第一次出现，则以未压缩形式发送字符。编码器修改霍夫曼树，或者增加发送字符的出现次数，或者添加新字符到树中。

这样的方法中，每次编码发送码字，接收器除具有确定接收到字符能力，也能对自己的霍夫曼树作同样的修改，使得它能够按照新修改树结构解释下一个收到的码字。

为了详细描述这种方法，假定被发送的数据开始的字符串为：

This is simple...

发送器每步引起的变化如图3-24 (a ~ g)所示。

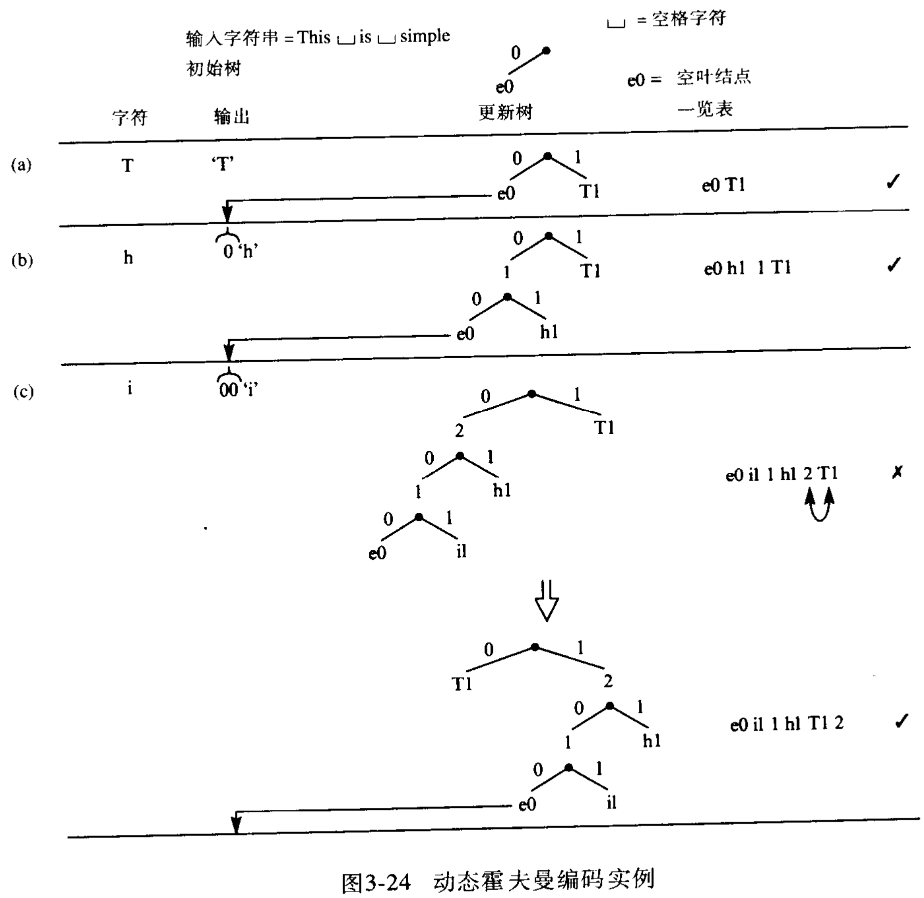


图3-24 动态霍夫曼编码实例

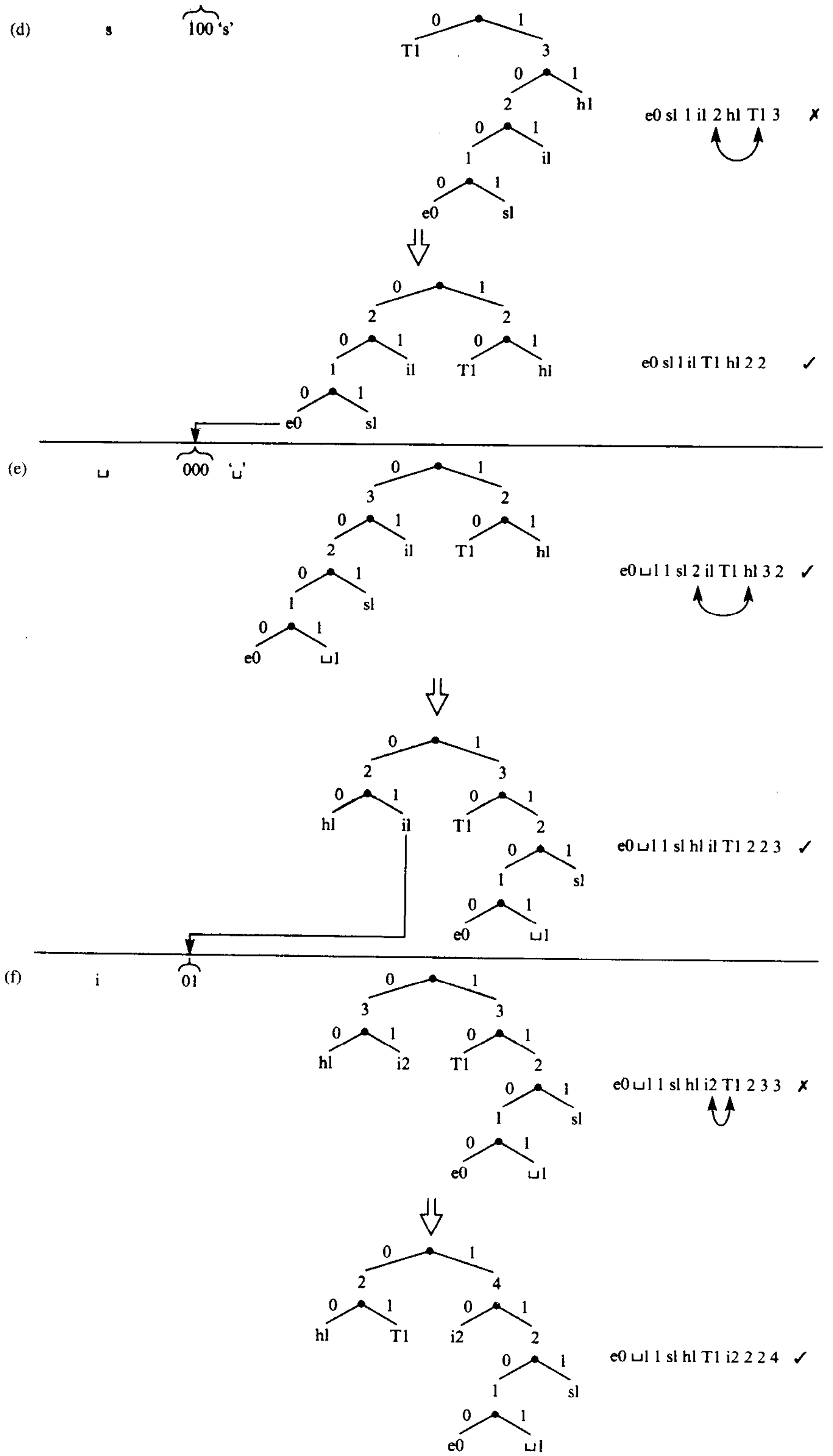

(d)
s
100 's'
e0 sl 1 il 2 hl T1 3 ✗
e0 sl 1 il T1 hl 2 2 ✓
(e)
␣
000 '␣'
e0 ␣1 1 sl 2 il T1 hl 3 2 ✓
e0 ␣1 1 sl hl il T1 2 2 3 ✓
(f)
i
01
e0 ␣1 1 sl hl i2 T1 2 3 3 ✗
e0 ␣1 1 sl hl T1 i2 2 2 4 ✓

图3-24 （续）

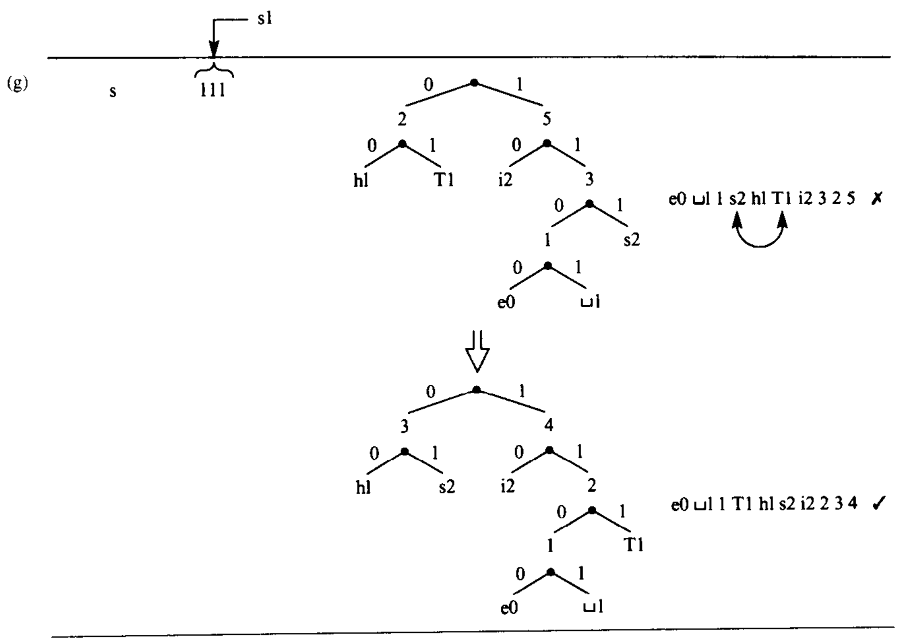


图3-24 （续）

开始时，发送器与接收器两者的树，只有根结点和单个**空叶结点**（即出现0次的叶结点，把它的分支赋值为0）。在树中正好有一个这样的结点，它的位置（码字）随着正在构造的树变化。开始的状态在图3-24中以e0表示。

然后，编码器从读取第一个字符T开始，并把T指定给根的分支1。因为这个字符第1次出现，它用T1表示，按未压缩形式发送（比如ASCII形式）。因为解码器的树是空的，它作为未压缩的字符解释接收到二进制串，并按同样的方法继续把字符指定到树中（见图3-24(a)）。 145

对每个后续字符，编码器首先检查字符是否已在树中出现过，如果出现过，则编码器按常规方法发送字符的码字，该码字由字符在树中的位置确定。如果没有出现过，则编码器发送空叶结点现在的码字（由空叶结点在树中位置决定）再加上未压缩字符码字。因为解码器与编码器有相同的树，它会容易地从接收到位串推断出是一个（压缩）字符的码字还是空叶结点再加上字符未压缩形式的码字。

编码器与解码器基于最近已发送/收到的字符继续修改它们的树。如果是新字符，则树中现有空叶结点被一个新分支结点代替，空叶结点被指定为0分支，而字符结点被指定为1（见图3-24(b)）。

如果字符已在树中出现过，则叶结点出现次数增加1。这样做后，叶结点现在的位置可能不是树中最佳的位置。因此，每次修改树，增加一个新的字符或者已有字符出现次数增加1，编码器与解码器需要检验。如需要，则修改树中所有字符现有的位置。 146～148

为了保证编码器与解码器两者一致地这样做，首先从修改树中的空叶结点出发，从底到顶，从左到右，列出叶结点与分支结点的权。如果所有结点的权都是顺序的，则树不修改，如果存在一个结点的权失序，则树结构按照权重递增的顺序交换该结点与其他结点（包括它的分支结点与叶结点）的位置来修改。第一次出现是在图3-24(c)中，而其他的例子在图3-24(d)到图3-24(g)中。

当发送的字符在先前已发送过，接下来的步骤如图3-24(f)所示。此时，发送字符是“i”，当编码器搜索树时，确定“i”已出现，发送它现有码字01。然后编码器将字符的权（出现次数）增1成为2，如前述方法修改结点位置。另一个实例是发送字符“s”，如图3-24(g)所示。

从这个例子可推出仅当字符开始重复时节约传输宽带。实际上，文本文件节约是很大的，目前动态霍夫曼编码应用于许多数据通信中，如V.32调制解调器中采用的数据压缩算法。

3.5.6 传真压缩

虽然使用霍夫曼编码，对于字符文件我们可获得达到2:1的压缩比，但**传真机**用扫描器产生数字化图像传输实现最显著节约。如图3-25所示，以每毫米3.85线或7.7线的垂直分辨率，近似于每英寸100线或200线。即每条扫描线以每毫米8.05**图像元素**或**像素**的速率进行水平方
149 向的数字化，白色为0与黑色为1。因此典型扫描一页产生大约二百万个二进制数。不进行数据压缩，如果以4800 bps传输速率传输，一页需要超过6分钟的时间。

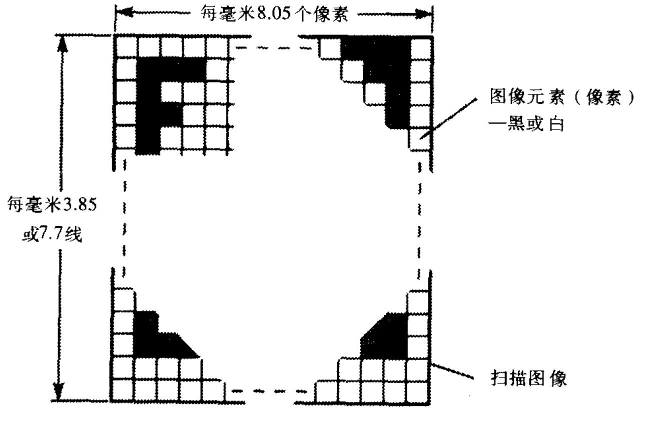


图3-25　传真扫描图像结构

实际上，在大多数文件中，许多扫描线仅由一长串的白像素组成，而另一些由长串白像素与长串黑像素混合组成。因为传真机通常在公用载波网上使用，所以ITU-T制定有关的标准，分为四类：T2（一类机）、T3（二类机）、T4（三类机）与T6（四类机）。前两类是早期的标准，现很少使用，后两类操作是数字化，三类机采用调制方法，用于模拟PSTN上，四类机全数字化，用于数字网络上（如ISDN）。两者都采用数据压缩，对大多数文档页面，压缩比例通常超过10:1。三类机传输一页的时间降低到一分钟内，由于高速传输速率（64 kbps），四类机传输一页少于几秒。

作为标准化过程部分，需作出典型扫描文档页面的广泛分析。根据扫描线中黑像素和白像素持续长度出现频率进行统计后，制定码字表。形成固定的码字表并划分为两个表：**结尾码表**和**组合基干码表**。每张表的码字如图3-26所示。

持续长度为0～63的黑像素或白像素只用结尾码表中码字，持续长度为64倍数的黑像素或白像素用组合基干码表表示。使用**过扫描**技术，就是说所有扫描线均以白像素最小长度码字开始。这个方法，接收器知道第一码字常涉及白像素，然后黑像素与白像素交替。因为方案使用两组码字（结尾码和基干码），它们称为**改进霍夫曼编码**。例如，一个持续长度12的白像素直接编码为001000，同样，一个持续长度12的黑像素直接编码为0000111。然而，一个持续长度140的黑像素编码为000011001000+0000111，即128+12像素。持续长度超过2560的像素

编码，使用多个基干码加一个结尾码。

白像素持续长度	码字	黑像素持续长度	码字
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000
21	0010111	21	00001101100
22	0000011	22	00000110111
23	0000100	23	00000101000
24	0101000	24	00000010111
25	0101011	25	00000011000
26	0010011	26	000011001010
27	0100100	27	000011001011
28	0011000	28	000011001100
29	00000010	29	000011001101
30	00000011	30	000001101000
31	00011010	31	000001101001
32	00011011	32	000001101010
33	0010010	33	000001101011
34	00010011	34	000011010010
35	00010100	35	000011010011
36	00010101	36	000011010100
37	00010110	37	000011010101
38	00010111	38	000011010110
39	00101000	39	000011010111
40	00101001	40	000001101100
41	00101011	41	000001101101
42	00101011	42	000011011010
43	00101100	43	000011011011
44	00101101	44	000001010100
45	00000100	45	000001010101
46	00000101	46	000001010110
47	00001010	47	000001010111
48	00001011	48	000001100100
49	01010010	49	000001100101
50	01010011	50	000001010010
51	01010100	51	000001010011
52	01010101	52	000000100100
53	00100100	53	000000110111
54	00100101	54	000000111000
55	01011000	55	000000100111

(a)

白像素持续长度	码字	黑像素持续长度	码字
56	01011001	56	000000101000
57	01011010	57	000001011000
58	01011011	58	000001011001
59	01001010	59	000000101011
60	01001011	60	000000101100
61	00110010	61	000001011010
62	00110011	62	000001100110
63	00110100	63	000001100111

(a) 续

白像素持续长度	码字	黑像素持续长度	码字
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
256	0110111	256	000001011011
320	00110110	320	000000110011
384	00110111	384	000000110100
448	01100100	448	000000110101
512	01100101	512	0000001101100
576	01101000	576	0000001101101
640	01100111	640	0000001001010
704	011001100	704	0000001001011
768	011001101	768	0000001001100
832	011010010	832	0000001001101
896	011010011	896	0000001110010
960	011010100	960	0000001110011
1024	011010101	1024	0000001110100
1088	011010110	1088	0000001110101
1152	011010111	1152	0000001110110
1216	011011000	1216	0000001110111
1280	011011001	1280	0000001010010
1344	011011010	1344	0000001010011
1408	011011011	1408	0000001010100
1472	010011000	1472	0000001010101
1536	010011001	1536	0000001011010
1600	010011010	1600	0000001011011
1664	011000	1664	0000001100100
1728	010011011	1728	0000001100101
1792	00000001000	1792	00000001000
1856	00000001100	1856	00000001100
1920	00000001101	1920	00000001101
1984	000000010010	1984	000000010010
2048	000000010011	2048	000000010011
2112	000000010100	2112	000000010100
2176	000000010101	2176	000000010101
2240	000000010110	2240	000000010110
2304	000000010111	2304	000000010111
2368	000000011100	2368	000000011100
2432	000000011101	2432	000000011101
2496	000000011110	2496	000000011110
2560	000000011111	2560	000000011111
EOL	000000000001	EOL	000000000001

(b)

图3-26 ITU-T 三类与四类传真机转换码

(a) 结尾码 (b) 组合基干码

三类机没有纠错协议，从码字表我们能推出，如果由于传输差错一位或多位损坏，接收器以出错位为界开始解释后续码字。接收器不再同步并且不能解码收到的位串。为了使接收器恢复同步，每个扫描线用一个已知**线结束**（EOL）码终止。以这种方式，如果接收器在最

大位的码字已扫描（分析）后，发现是一个无效码字，它开始搜索EOL格式。如果在预置线的个数后，也失败，则废弃接收过程并通知发送机器。在每个扫描页码字前设置一个EOL，而每页结束用6个连续的EOL表示。

因为每个扫描线独立编码，T4编码方案称为**一维编码**方案。我们可推知扫描图像提供黑
150 像素或白像素的有意义范围，这种编码令人满意。例如，由字母和表格组成的文件。但当文件包含照片图像时不是很满意。因为用黑像素与白像素的可变密度表示不同黑与白阴影。用T4编码方案会产生大量很短的持续长度黑像素与白像素，会引出负压缩比，即压缩形式的文件比未压缩文件需要更多的位。

由于这个原因，已规定另一种T6编码方案，它是三类传真机的可选特性，但四类机强制使用。当支持三类机时，每个扫描线结束的EOL码有一个附加的标志位。如果它是1，则下一条线采用T4编码方案；如果它是0，采用T6编码方案。后一个方案称为**改进—改进READ（MMR）编码**，也称为**二维编码**或**2D编码**，这是因为比较相邻两条扫描线识别黑像素持续长度与白像素持续长度来编码。READ代表**相对元素地址指定**，"改进"是指对于早期编码方案的改进版本的改进。

MMR编码利用多数扫描线与前一扫描线仅有少量像素不同这一事实。例如，如果一条扫描线出现黑像素持续长度，则下一个扫描线通常也出现同样的黑像素持续长度（为原长度加3或减3）。关于MMR编码，它是用一条扫描线中像素持续长度相对比较前一条扫描线中像素持续长度进行二维编码。正在编码的扫描线称为**编码线**（CL），前一条参考编码线称为**参考线**（RL）。我们常设定第一条参考线为一条（假想）全白像素的扫描线，而真正的第一条扫描线参考这条线进行编码。接着已编码过的线变成下一条编码线的参考线，等等。为了保证完整扫描一页，扫描头常从页的左边开始，每条线常以假想白像素开始。

我们识别正在编码扫描线的持续长度相对于参考扫描线持续长度的三种可能或模式。三种模式如图3-27所示。三种模式通过参考扫描线的下一个持续长度（$b_1\,b_2$）相对于正在编码的扫描中下一对持续长度的开始位置和结束位置（$a_0\,a_1$与$a_1\,a_2$）来确定。注意这些持续长度可以是黑像素或白像素。三种可能性是：

1) 通过模式　这种情况，参考扫描线上持续长度（$b_1\,b_2$）位于正在扫描线下一个持续长度（$a_1\,a_2$）的左边，即b_2位于a_1的左边。一个实例在图3-27(a)中给出。此时，持续长度$b_1\,b_2$的编码由图3-26的码字给出。注意，如果正在编码扫描线的下一个像素a_1正好位于b_2之下，则这种状态不是通过模式。

2) 垂直模式　这种情况，参考扫描线上持续长度（b_1b_2）重叠于正在扫描线下一个持续长
151~152 度（$a_1\,a_2$），a_1与b_1的距离小于或等于3个像素。两个实例在图3-27(b)中给出。此时，根据不同持续长度$a_1\,b_1$（a_1与b_1的不同位置）进行编码。大多数码字是这一类。

3) 水平模式　这种情况，参考扫描线上持续长度（$b_1\,b_2$）与持续长度（$a_1\,a_2$）的重叠部分大于±3个像素，两个实例在图3-27(c)中给出。此时，两个持续长度$a_0\,a_1$与$a_1\,a_2$的编码由图3-26的码字给出。

编码过程流程如图3-28所示。注意：每条扫描线的第一个像素之前，设想有一个白像素a_0
153 作为起始像素，因此每条扫描线中第一个持续长度$a_0\,a_1$用$a_0\,a_1-1$来代替，如果在扫描线的整个编码期间，没有检测出a_1、a_2、b_1或b_2，则立即认为它们就在参考扫描线上最末一个像素之后的假想像素位置上。

一旦a_0的第一/下一个位置确定，则下一个码字的a_1、a_2、b_1和b_2位置也可找到。计算b_2相

对a_1的位置，确定模式。如果b_2在a_1左边，则认为是通过模式；如果不是，则由距离$a_1 b_1$的绝对值决定是垂直模式还是水平模式。然后对确定的模式进行编码，并将a_0修改到适当的位置作为下一个码字的起始位置。这个过程重复直到扫描线终点。这就是进行到扫描线最后一个像素之后的假想像素并假定最后像素有不同颜色。然后，当前编码的扫描线变成新参考扫描线，下一条扫描线成为新的编码扫描线。 154

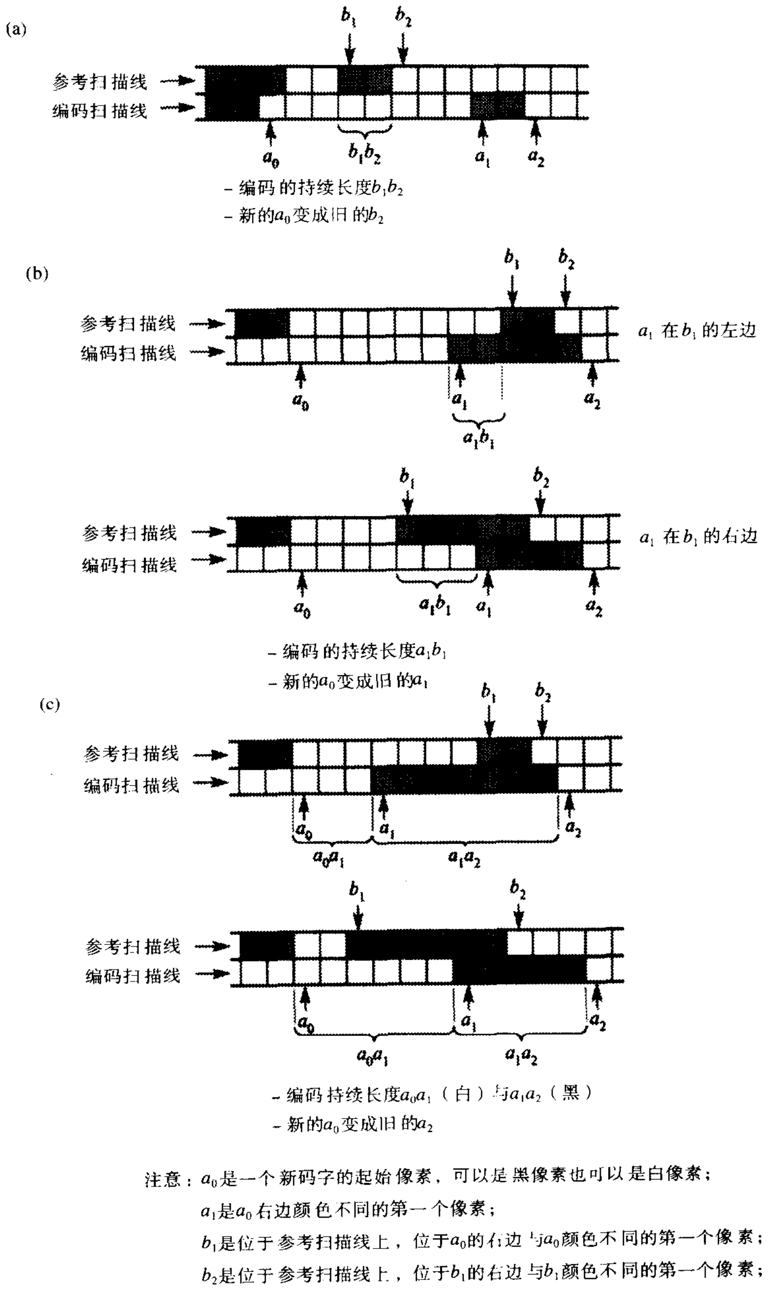


图3-27 持续长度编码模式实例

(a) 通过模式 (b) 垂直模式 (c) 水平模式

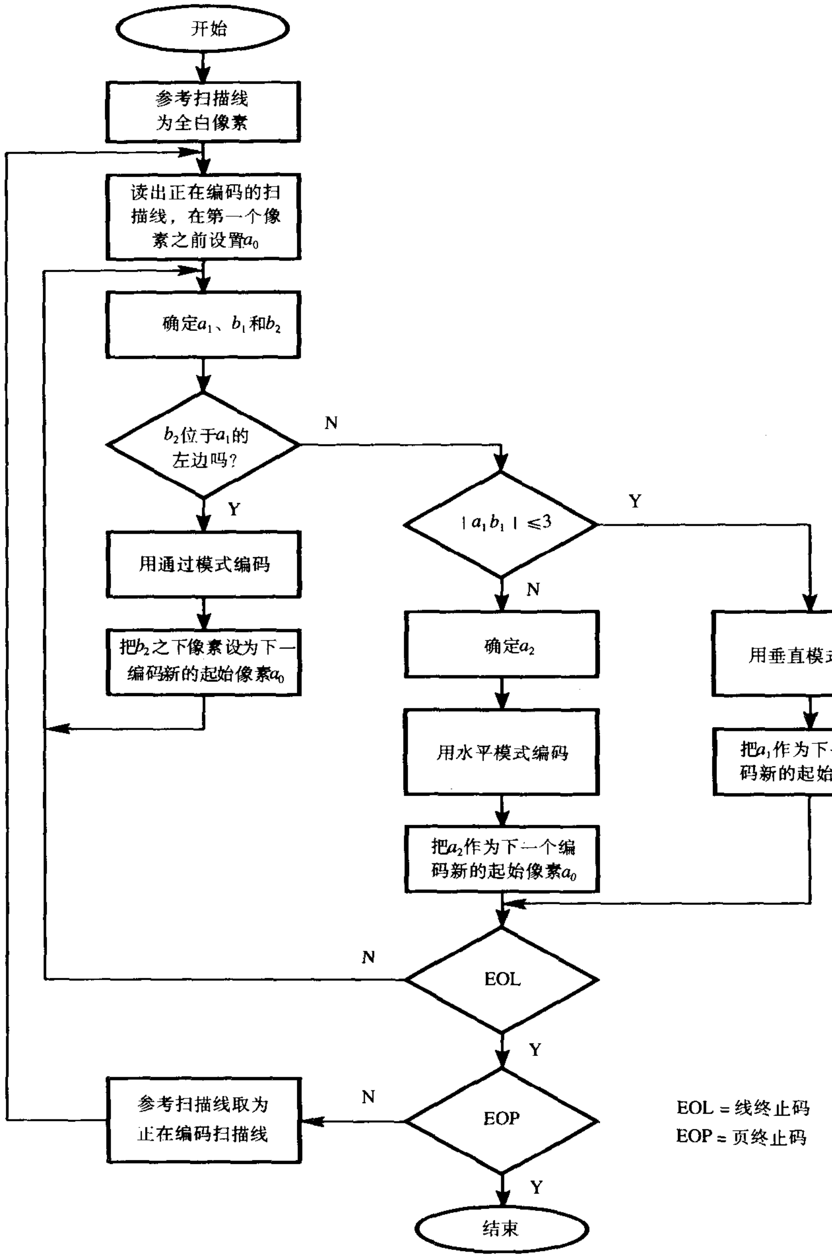


图3-28 改进-改进READ编码方案

因为编码持续长度关系到三种模式之一，采用附加码字指明接下来的码字是通过模式还是水平模式，或者对垂直模式直接给出码字长度。附加码字在第三张表给出，称为**二维码表**，如表3-2所示。表中最后一项称为**扩充方式**。它只有一个码字，该码字用于在一页结束之前提前终止编码操作。这提供了允许一页的不同部分的发送用非压缩形式或者用不同编码方案。

表3-2 二维码表

模 式	编码持续长度	简 写	码 字
通过	b_1b_2	P	$0001 + b_1b_2$
水平	a_0a_1, a_1a_2	H	$001 + a_0a_1 + a_1a_2$
垂直	$a_1b_1 = 0$	V(0)	1
	$a_1b_1 = -1$	$V_R(1)$	011
	$a_1b_1 = -2$	$V_R(2)$	000011
	$a_1b_1 = -3$	$V_R(3)$	0000011
	$a_1b_1 = +1$	$V_L(1)$	010
	$a_1b_1 = +2$	$V_L(2)$	000010
	$a_1b_1 = +3$	$V_L(3)$	0000010
扩充			0000001000

3.6 传输控制电路

大多数半导体制造商提供一系列集成电路（IC），可以执行本章讨论的所有功能。在异步传输时，有执行时钟（位）与字符同步功能的集成电路和生成与校验每个字符奇偶校验位的集成电路。同样，在同步传输时，有与面向字符传输相关的字符同步及奇偶校验位生成与检测集成电路，其他还有面向位传输的零插入与删除集成电路和CRC生成与检测集成电路。通常，特别是高比特率的时钟编码与解码由各自独立的集成电路实现，有许多可用于传输控制电路的集成电路。

大多数传输控制电路是**可编程的**，就是说，我们可通过规定二进制模式写入选定内部寄 155
存器定义设备的具体操作方式（异步的/同步的，面向字符的/面向位的，奇偶校验位/CRC等等）。我们称这些电路为**通用通信接口电路**。一般，一个单片电路提供一个、两个甚至四个独立（全双工）传输线接口电路。

控制电路操作的设备（例如微处理机），首先把定义好的字节（位模式）写入**模式寄存器**编制想要的操作模式。然后，设备写入第二个字节到**命令寄存器**为发送/接收字符或字节做准备。发送与接收通道常常是**双缓冲**的，就是说控制设备有一个完整字符（字节）时间用于在传输前或接受后处理单个字符，而不是单个位时间。

绝大多数通用设备名称与功能如下：

- 通用异步收发器（UART）
  - — 起始与停止位插入与删除；
  - — 位（时钟）同步；
  - — 字符同步；
  - — 每个字符的奇偶校验位产生与检测（由控制设备计算BCC）。
- 通用同步收发器（USRT）
  - — 低比特率DPLL时钟同步；
  - — 字符同步；
  - — 同步空闲字符生成；
  - — 每个字符的奇偶校验位生成与检测（由控制设备计算BCC）。
- 通用同步/异步收发器（USART）

— 可编程作为UART或USRT操作；
— 具有UART与USRT所有可编程的功能。
* 面向位协议电路（BOP）
  — 起始与结束标志的插入与删除；
  — 零位插入与删除；
  — CRC生成与检测；
  — 空闲模式生成。
* 通用通信控制电路
  — 可编程操作，作为UART、USRT或BOP；
  — 具有每个电路的所有可编程功能。

3.7　通信控制设备

在许多数据通信应用中，一个共同的要求是要有分布式终端（如个人计算机），所有终端
156 请求接入一台中央计算设备。这台中央计算设备可以为企业运行集中电子邮件服务或驻留分布式终端请求访问的中央数据库。

如果所有终端坐落在不同地方，惟一的解决方案是每个终端单独提供传输线，如图3-29所示。图中(a)部分假定终端分布在机构周围，而(b)部分假定每个终端在不同的机构中。在后一种情况下，依据发送数据量与呼叫次数很可能调制解调器需要在交换或租用线路上操作。交换连接情况下，终端一般在通信接口有自动拨号装置。

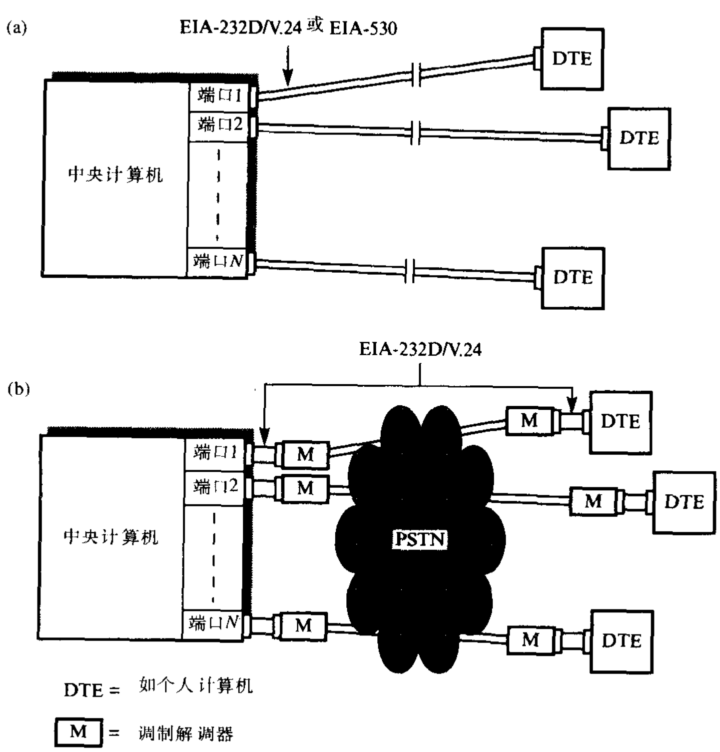


图3-29　简单终端网络
(a) 本地分布　(b) 远程分布

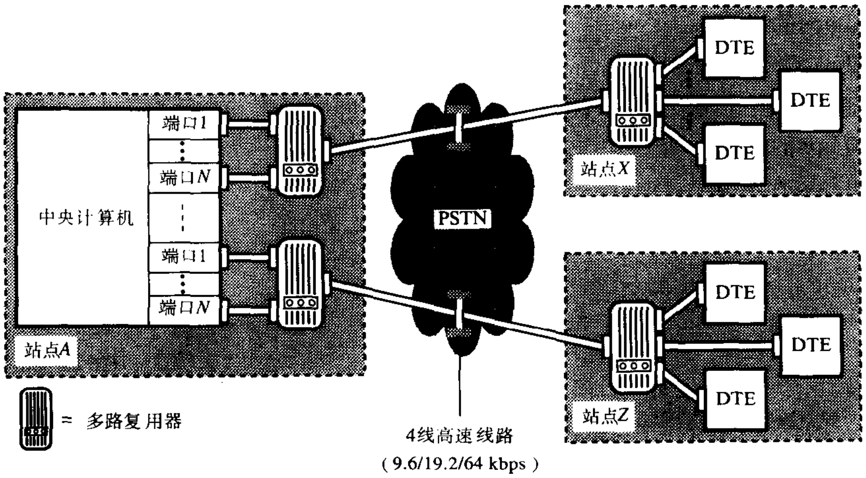


图3-30 基于多路复用器的网络结构

对于许多终端集中在一起的应用，我们采用称为**多路复用器**的设备以最小化需要的传输线数目。这样的使用一条传输线路的设备以比单个用户终端速率高的比特率运作。如图3-30所示，通常在链路每端使用相同的多路复用器。采用多路复用器的方法对终端与中央计算机是透明的。

多路复用器有两种类型：**时分多路复用器**与**统计多路复用器**。时分多路复用器分配每个终端共享线路上传输容量的一个规定部分，统计多路复用器在按需或统计基础上分配传输容量。 157

3.7.1 时分多路复用器

一个典型的时分多路复用器的应用如图3-31 (a)所示。每个机构中的终端有关内部的所有请求接入中央计算机，假定每个站点都有大量的终端，它们产生的通信量足以认为租用高比特率线路连接不同站点到中央计算机是合算的，典型租用线路速率是64 kbps，或者更高，依赖于终端的数目。

图3-31 (b)表示每个MUX的内部结构。每个终端连接一个UART，并以异步传输方式操作。MUX内部的控制微处理器控制UART之间字符的传送与高速链路接口电路。由于接口电路一般按面向字符同步传输方式操作，所以，它包含一个USRT。

为了保证MUX对终端/计算机透明，高比特率线路的传输容量按终端UART以及按规定速率运作的计算机端口划分，这种技术称为**速率适配**。它包含把有效链路容量分成若干个帧，如图3-31 (c)所示。

每个帧包括N个字节，与帧中单个字节位置有关联的比特率形成一个适当基本多路复用速率，然后用每个帧的多个字节得出每个终端的比特率。但是每个字节中不是所有位都用作用户数据。每个字节的第一位用于组帧，一个帧中所有字节的这个位位置作为重复的位模式发送，使得接收器可以确定每帧的开始和结束。对于每个UART，V.24/EIA-232D接口控制线的第8位用于发送握手控制位（DSR/DTR和RTS/CTS）的状态。 158

作为一个实例，假设高速线路的比特率是64 kbps，并采用20字节的帧长度。每个字节6个数据位，用户数据速率是2400 bps。因此这条高速线路可用于支持下列的任一种： 159

20个2400 bps的终端

10个4800 bps的终端

5个9600 bps的终端

1个48 kbps的终端

另一方面，该条高速线路可支持不同速率的终端混用，例如，

8个2400 bps的终端

4个4800 bps的终端

1个9600 bps的终端

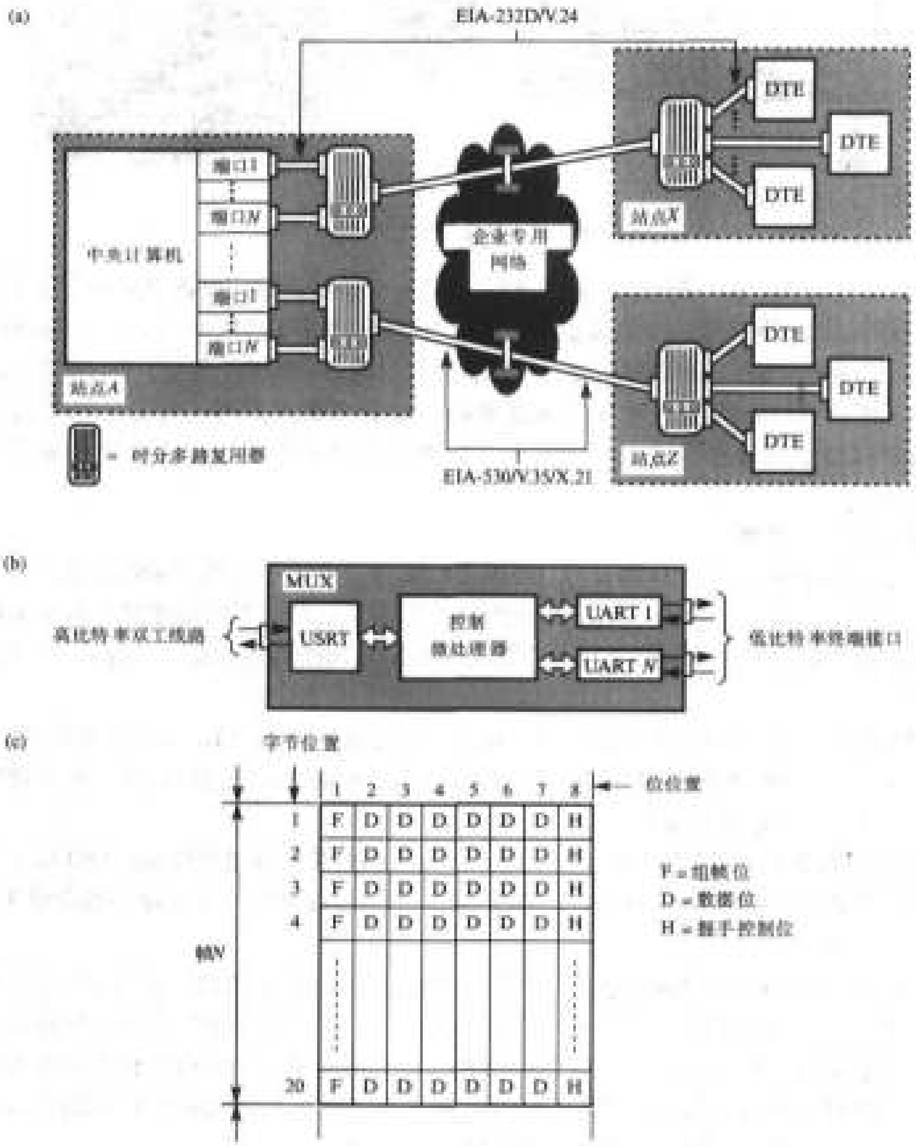


图3-31 时分多路复用器原理

(a) 应用 (b) MUX结构 (c) 速率适配

为了执行多路复用操作，对每个UART，微处理器使用两个2字节缓冲器，一个用于发送，另一个用于接收。对于发送，从UART接收到每个字节（例如，一个7位字符加1个奇偶校验位）逐个字符地送入2字节（循环）缓冲器。同时，微处理器与高速链路比特率同步，以每6

位为一组读出现在缓冲器的内容。相反过程用另外2字节缓冲器从高速链路执行接收。然后以一致的方法设置握手控制位以反映有关接口相应线路状态。

3.7.2 统计多路复用器

时分多路复用器对每个终端在每一帧中分配一个固定字符时隙。当控制处理器轮询到相关UART时，如果相应的终端或计算机没有字符准备发送，微处理器必须在这个时隙插入一个NUL字符。因此，导致有效带宽使用率降低。如果数据链路是专用线路，这可能还不是重要的，但如果使用的是PTT线路，代价是高的。一个高效的方法是采用统计多路复用器（统计MUX）。

统计多路复用器操作原理如下：在终端键盘输入字符平均速率通常大大低于线路有效传输容量——当然指人类用户情形。如果使用平均用户数据速率，而不是传输线路速率，则公用数据链路的比特率可能很低，传输线路的真正使用代价降低。例如，假定有8个终端需要通过PTT线路连接到一台远程中央处理机，而线路最大比特率是4800 bps。采用一个标准MUX和一条线路，每个终端标准操作速率少于600 bps，比如300 bps。该限制的影响是计算机对终端键入每个字符的响应时间相对是慢的，或者计算机向终端发送一个字符块的延迟是显著的。另一方面，如果终端的平均数据速率是300 bps，那么采用统计多路复用器，一个终端可用最大有效比特率4800 bps发送数据，因此，对每个键入字符的平均响应时间有很大的改进。 160

为了实现这个方案，统计MUX中的控制微处理器不仅要对每个终端的UART轮询，还要提供并管理一个有限缓存器，使得当若干个终端同时激活时，允许公用数据链路接受暂时的过载。由于在公用数据链路上按统计模式发送数据而不是预先分配方式，因此每个字符或字符块必须附带标识信息。

统计MUX中微处理器的另一个功能是差错控制。对于通常MUX，简单的回波校验是完全可接受的，这是因为每个终端和计算机端口在双工链路每个帧中分配一个固定字符时隙。然而，因为统计MUX按统计模式工作，我们在共享数据链路上更通常对字符块执行差错控制。一个典型处理如图3-32(a)所示。

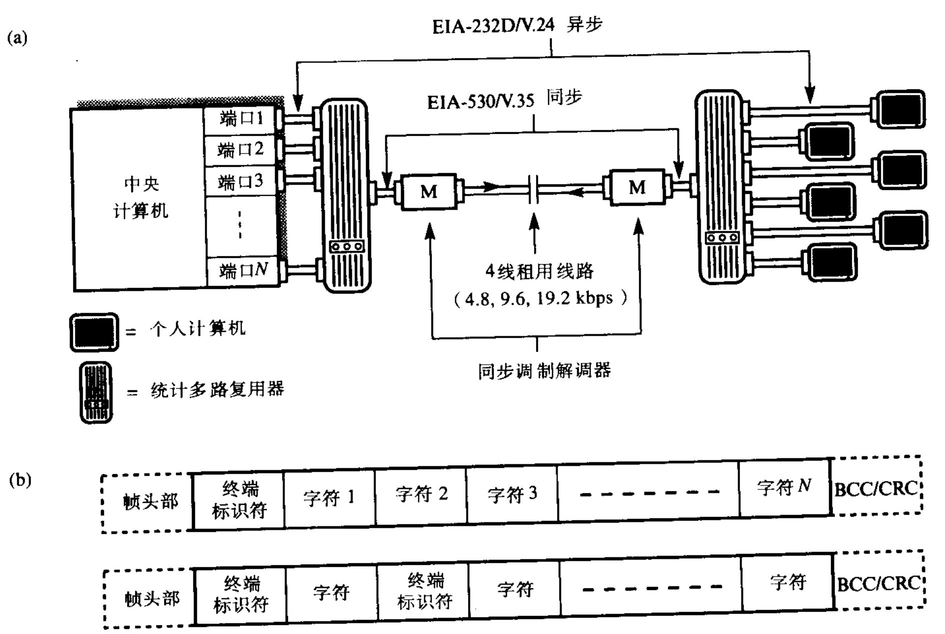


图3-32 统计多路复用器原理

(a) 网络结构 (b) 组帧选择

为了降低每个发送字符的开销，我们通常在共享数据链路上将若干个发送字符组合为一帧。有几种方法能做到这一点。图3-32(b)表示其中两个实例。其一，控制微处理机等待直到有许多来自同一终端的字符，即组成单个线路的字符串，然后在它们的头部附加终端（通道）标识符作为完整的块发送。另一种，每个块包含来自目前所有激活终端的字符串，每个终端都有一个独立的终端标识符。正如我们所见，封装字符在共享数据链路上发送要有选择地占用每帧或每块的信息字段（I字段）。链路使用通信协议一般是面向字符或面向位的同步协议，我们将在第4章进行讨论。

3.7.3 块方式设备

高级终端、银行数据登录终端设备和百货公司销售点终端，通常以块方式操作，而不是字符方式。在块方式中，当键入字符后，它由终端本地处理器直接回显到屏幕上，然后数据传送到中央计算机处理，并组成一个完整数据块（一个信息）。这样的终端支持更复杂的面向
161 块的通信协议以控制信息的传送。由于每个发送消息的可接受响应时间比交互性的字符方式网络期待响应时间慢得多，块方式终端网络通常用在以响应时间为代价降低通信线路成本的设备上。

1. 多站线路

块方式终端网络降低传输成本的常用方法是采用**多站线路**（也称为**多点线路**），使用多站线路的网络结构如图3-33(a)所示。代替用单独线路将终端直接连到中央计算机，一些终端共享线路。这种方法，大大减少线路的数目（因此减少依赖网络地理分布的调制解调器与线路驱动器的数目）。然而，每个终端共同体仅有一条线，在同一时间只有一个信息块可以被终端或中央计算机发送。所以，每条线的所有传输由中央计算机使用**轮询—选择**仲裁方法控制。

2. 轮询—选择

为保证每条共享通信线路任何时刻仅有一个信息发送，中央处理机或它的代理，按规定顺序对连接在这条线上的每一个终端**轮询**或**选择**。由于对共享线上的每个终端指定惟一标识符，中央计算机与某个终端通信采用发送头部具有终端标识的信息来实现。信息可分为两种类型：控制和数据。

中央计算机依次地周期发送轮询控制消息，询问终端是否有信息等待发送。如果有，它用一个数据消息响应；否则，它用一个无发送消的控制消息响应。同样地，每当中央计算机要发送一个消息给某个终端，它发送一个选择控制消息给特定终端。假设被选中的终端能接
162 收信息，它返回一个接收就绪控制消息响应，然后中央计算机发送数据信息。最后，终端确认数据信息正确接收，中央处理机继续轮询或选择其他终端。这种轮询类型称为**呼叫轮询**，因为网络中每个终端在发送或接收信息前，必须被轮询或选择，对于一个大的网络会造成相当长的响应时间。中央计算机承担的通信开销很高。

为了克服这些问题，多站网络的一个常用类型使用**群集控制器**减少网络的响应时间和使用**前端处理机**（FEP）降低中央计算机通信开销。这样的网络实例如图3-33(b)所示。每个群
163 集控制器相当于一个中央计算机的代理，对连接终端进行轮询和选择，因此管理所有进来与出去的终端信息传送。实际上，中央计算机或FEP仅需询问或选择每个控制器。

一台FEP是典型的小型计算机，它与中央计算机紧密地耦合。FEP可编程处理所有轮询和选择，使中央计算机集中时间去处理主要任务。FEP的主要优点是仅当数据已被接收或者被发送时才需要用中央计算机。进一步，由于信息传送所有通信开销都由FEP完成，所以中央计算机仅需在信息传送到FEP和来自FEP时工作。

如果终端群集广泛地分散，有时称为**集线器轮询**的另一个机制会是合理方案。集线器轮询如图3-33(c)所示。这种配置中，每个群集控制器与最接近的相连接，而不是都连接到中央计算机。如前所说，中央计算机管理所有来自群集控制器或进入群集控制器的传送，中央计算机任意时间都在线路顶端选择并发送数据信息到任意控制器，如图3-32(c)所示。

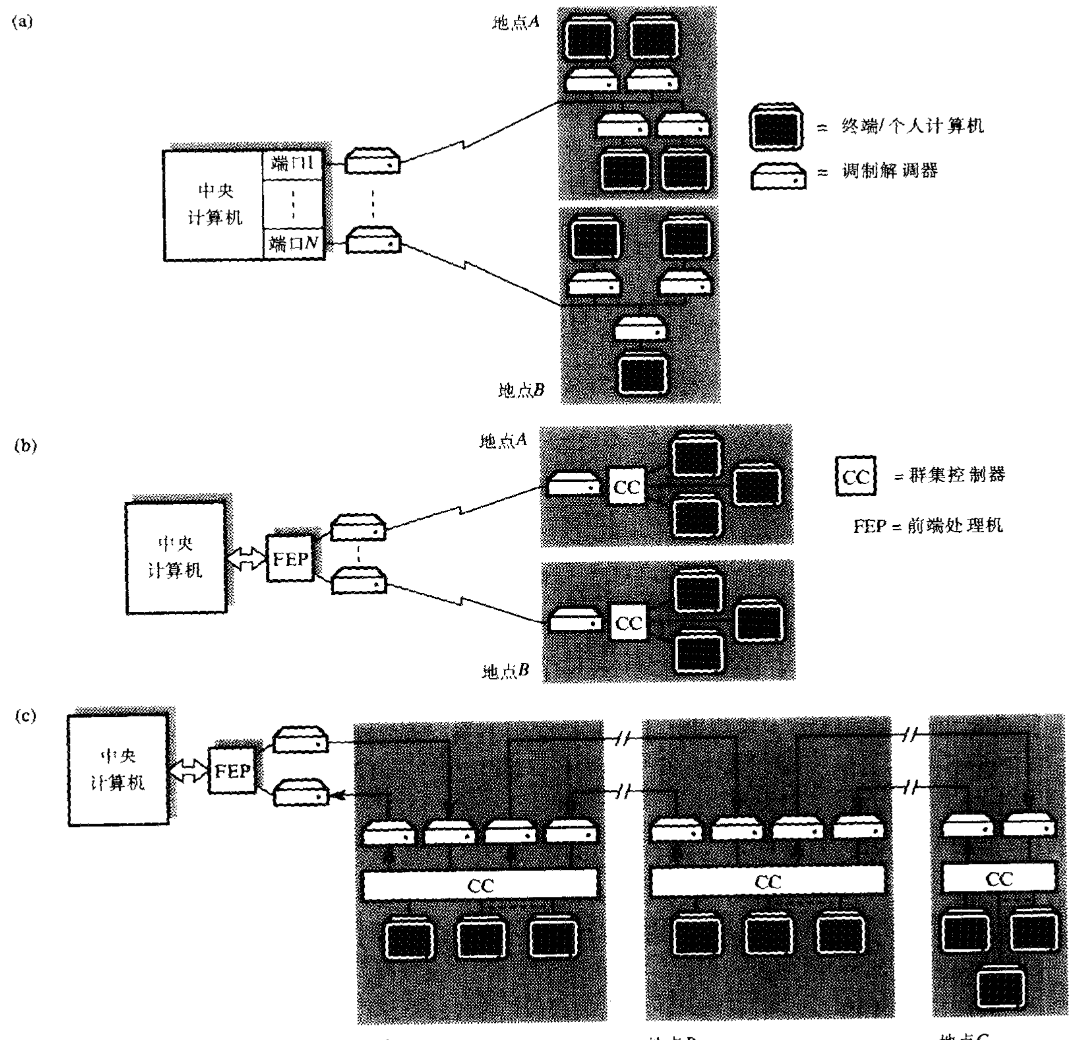


图3-33 网络轮询选择

(a) 多站点 (b) 群集控制器 (c) 集线器轮询

为了接收来自控制器的信息，中央计算机向最远控制器发送一个轮询控制消息，最远控制器在底线（控制线）向最邻近控制器发送一个数据消息或者没有信息要发送的控制消息响应。收到这个响应消息，下一个控制器解释为轮询消息，如果有等待的信息，在该信息附加到来自它的上游邻近控制器的接收信息尾部作为响应，然后，组成的信息向它的返回路线下游邻近控制器发送，该过程持续直到线路链终止，每个控制器添加自己的响应信息继续向中央计算机传递。最后，接收到合成响应消息，FEP拆卸消息，把它包含的有效信息传给中央计

算机进一步处理。

习题

3.1 (a) 解释异步传输与同步传输的差别；

(b) 假定采用异步传输，1个起始位，2个停止位和1个奇偶校验位，每个信令单元是2位，针对下列信号（波特）速率求信号传输速率（用bps表示）：

(i) 300

(ii) 600

(iii) 1200

(iv) 4800

3.2 接收器时钟与发送器时钟速率系数为 ×1 和 ×4。借助图表，说明在异步传输控制方式中，时钟（位）和字符同步的方法。

3.3 借助图表说明如何得到时钟同步，使用：

(a) 双极性编码

(b) 相位(曼彻斯特)编码

164 (c)差分曼彻斯特编码

3.4 用波形实例配合说明下列编码方案的主要特点：

(a) AMI

(b) B8ZS

(c) HDB3

(d) 2B1Q

评述每个方案的优点。

3.5 假设比特率为64 kbps，求习题3.4中每个编码方案的信号速率（用波特表示）。

3.6 (a) 解释在什么情况下数据编码与DPLL电路可用来实现时钟同步，也借助于图表说明DPLL电路的工作原理。

(b) 假设接收器起初不同步，给出DPLL电路调整到传输波形标准位中心所需的最少跳变次数，且说明实际上是如何实现的？

3.7 假设采用同步传输控制方案，说明在下列情况下，字符同步和帧同步是如何实现的：

(a) 面向字符的传输

(b) 面向位的传输

3.8 解释何为数据透明性，并说明如何用下述方法实现：

(a) 字符填充

(b) 零位插入

3.9 说明差错检测的奇偶校验位方法，以及如何将具扩充到字符块。针对一个字符描绘计算奇偶校验位的电路框图，并解释奇校验与偶校验之间的不同。

3.10 借助实例，定义下列术语：

(a) 单个位差错

(b) 双位差错

(c) 突发差错

举一个帧的内容为例，说明利用块和校验不能检测出的传输差错类型。

3.11 (a) 解释CRC差错检测方法的工作原理，采用生成多项式x^4+x^3+1，用实例方法 如何表示：

(i) 生成差错检测位

(ii) 接收帧检测传输差错

(b) 举例说明(a)中给出的生成器多项式相同的差错类型为何不能被检测出。列出该生成多项式不能检测出的其他差错类型。

3.12 一个工程管理系统，在数据链路上发送从0000到1111共16个二进制信息。每个信息采用3位CRC保护，而生成多项式是x^3+x^2+1。

(a) 对下列三个信息的每一个求3个校验位：

0000 0001 0010

(b) 说明术语“汉明距离”的意义，求0000，0001，0010的“汉明距离”。

(c) 假定发送信息1111，表示如何检测到传输码字中的单个位差错与双位差错。

(d) 给出一个不能检测出的无效接收码字。

3.13 (a) 作出电路原理框图，生成与检测被传输帧的CRC，假设生成器多项式为x^4+x^3+1。当在线路上发送和接收数据时，给出发送数据的发送内容与FCS移位寄位器内容。

(b) 用软件伪码如何实现生成与校验功能的算法。

3.14 用实例说明下列数据压缩算法的操作：

(a) 压缩十进制数

(b) 相对编码

(c) 字符压缩法 165

3.15 采用实例针对一个发送字符组说明构造霍夫曼编码树所用的规则。

3.16 由7个从A到G不同字符组成的信息在数据链路上传输，分析显示，每个字符的出现概率为：

A0.10 B0.25 C0.05 D0.32 E0.01 F0.07 G0.2

(a) 求信息的熵；

(b) 用静态霍夫曼编码求相应的码字组；

(c) 求发送信息对应码字组中每个码字的平均位数并与定长二进制码字及ASCII码字比较。

3.17 (a) 表述霍夫曼编码的前缀性质，说明例题3.16求出码字组能满足这个性质；

(b) 推导对接收到的用码字组编码的位串进行解码的算法流程；

(c) 假定接收到位串由7个字母混合组成，给出解码操作实例。

3.18 用霍夫曼编码发送下面的字符串：

A B A C A D A B A C A D A B A C A B A B

(a) 求霍夫曼编码树；

(b) 针对常规ASCII编码与二进制编码确定节约的传输宽带。

3.19 参照图3-24的动态霍夫曼编码实例：

(a) 假定采用ASCII编码，写出发送字符串“This is”的实际二进制数据流的形式；

(b) 如果下一个发送字是“the”，求出已有霍夫曼树的扩展。

3.20 已知三类传真机在二进制数组中的一条扫描线，依照图3-26的霍夫曼表推导一个算法：

(a) 确定发送码字

(b) 解码接收到的码字串

3.21 借助像素模式实例，解释四类传真机用的下列术语的含义：

(a) 通过模式

(b) 垂直模式

(c) 水平模式

用实例推导执行编码操作的算法。

3.22 解释时分多路复用器与统计多路复用器之间的不同。

画出表示时分多路复用器的内部结构的框图并解释它的操作。描述共享数据链路的组织方式，控制设备如何确定每个接收到字符目的地。

3.23 画出采用统计多路复用器的终端网络的框图。描述使用这种设备的共享数据链路的组织方式，控制设备如何确定每个字符的目的地。

3.24 画出采用多站线路与轮询—选择控制协议的典型块方式终端网络的框图，解释网络操作，并描述计算机如何向每个终端发送信息并从每个终端接收信息。

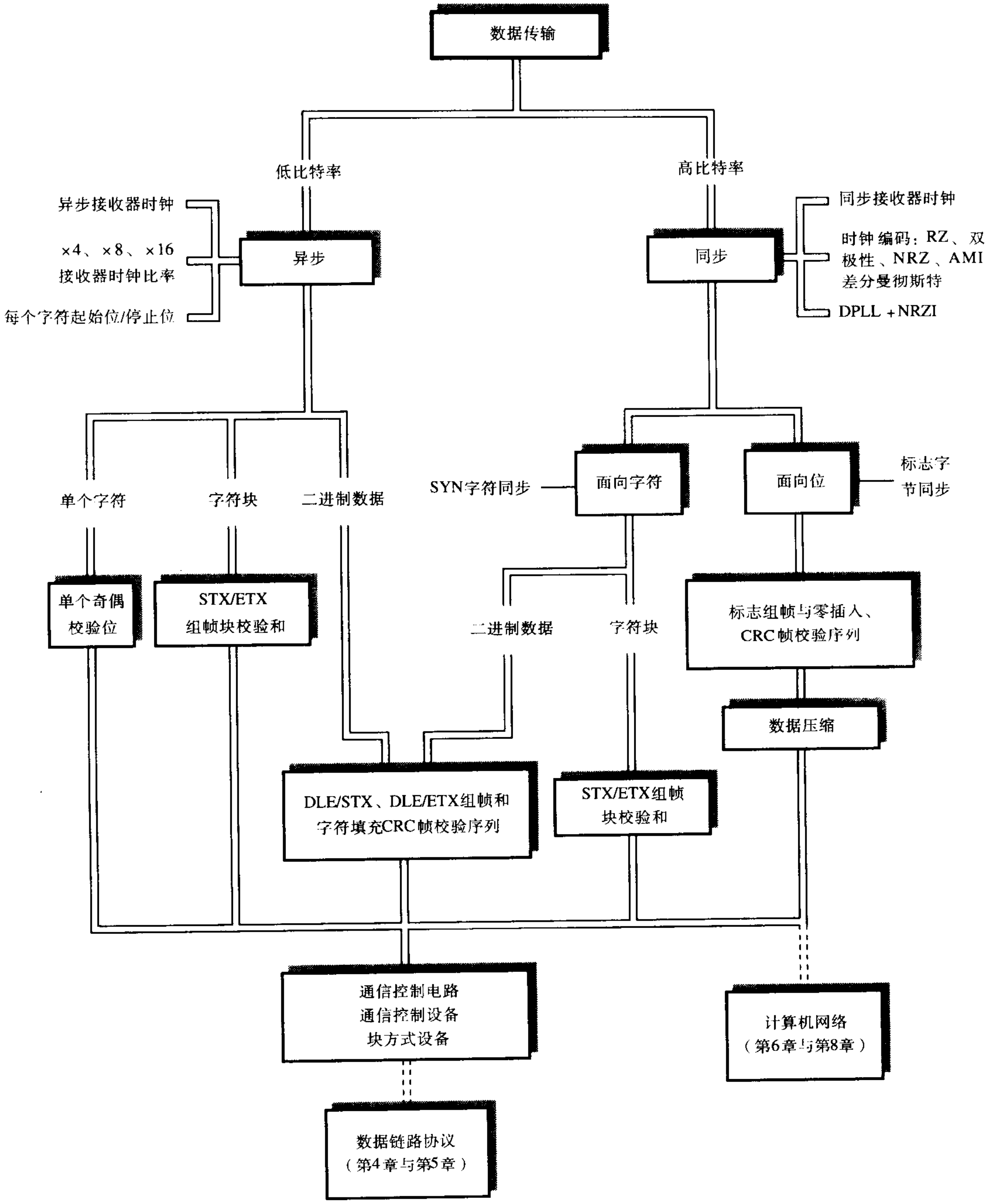
3.25 (a) 说明下列设备的功能：

(i) 群集控制器

(ii) 前端处理机

166 (b) 区别呼叫轮询与传递轮询。对每种轮询方式画出它的框图并解释其操作。

本章概要


数据传输
低比特率
高比特率
异步接收器时钟
×4、×8、×16
接收器时钟比率
每个字符起始位/停止位
异步
同步
同步接收器时钟
时钟编码：RZ、双极性、NRZ、AMI
差分曼彻斯特
DPLL +NRZI
单个字符
字符块
二进制数据
SYN字符同步
面向字符
面向位
标志字节同步
单个奇偶校验位
STX/ETX
组帧块校验和
二进制数据
字符块
标志组帧与零插入、
CRC帧校验序列
数据压缩
DLE/STX、DLE/ETX组帧和
字符填充CRC帧校验序列
STX/ETX组帧
块校验和
通信控制电路
通信控制设备
块方式设备
计算机网络
（第6章与第8章）
数据链路协议
（第4章与第5章）

第4章 协议基础

本章目的

读完本章，应该能够

- 识别数据链路协议是由差错控制，流量控制和连接管理等功能部件组成；
- 说明空闲RQ和连续RQ差错控制方案的操作以及表示它们操作的帧时序图；
- 理解分层结构的原理；
- 了解规定协议操作所使用的不同方法；
- 用状态转换图、扩充事件—状态表和结构化程序代码等形式规定每种差错控制方法的操作；
- 用空闲RQ、选择重发和回退N帧等差错控制方案确定链路容量的利用；
- 了解选择重发和回退N帧控制方案的差异；
- 说明用于差错控制的滑动窗口流量控制方案的操作及其序列号的作用；
- 理解并说明协议连接管理的任务。

引言

第3章描述两台DTE通过点到点数据链路发送信息帧所采用的线路与技术。还描述各种差错检测方案，用已知的概率，接收DTE确定发送二进制数据出现差错的情况。换句话说，如
168 果采用正向差错纠正，接收器可以用接收到二进制数据流，再加已知概率甚至差错出现时推导正确发送的信息。

但是，一般描述的技术仅提供发送信息和接收器检测传输差错的基本机制。当检测出一个传输差错时，甚至仅有一个（未知）位错，则整个数据块必须丢弃，因此这种类型的方案称为**最佳尝试传输**。由于某些理由，我们将在4.4节讨论**无连接式传输**。

除这种操作方式外，另一种称**可靠传输**的操作或者**面向连接传输**也常被应用。这种方式除检测外，当出现差错时，通信双方必须遵守一组规则或控制方法，保证可靠的（即，高概率）无差错与无重复以及以正确时序传送信息。为了达到这个要求，目的地控制设备要通知源设备已发送数据存在差错，必须重发受影响帧的另一个备份。结合差错检测/纠正操作过程称为**差错控制**。另外，还有其他一些控制机制，必须观察通信双方。这些构成**数据链路（控制）协议**，我们将在本章考虑数据链路协议的基本内容。在第5章中将给出引入的基本协议的实际实现的描述。

4.1 差错控制

当我们通过键盘输入数据到计算机时，由于每个键的结果码字利用UART与异步传输以位串行方式发送到计算机，然后计算机输入过程的控制程序读出并存储接收到字符，开始送到显示屏幕输出。如果显示的字符与想要的字符或输入的字符不同，那么可简单输入一个合适的控制字符，比如说一个删除字符或退格字符。当控制器收到时，控制程序就废弃以前输入的字符并从屏幕删去该字符。这种方法中，我们执行的是**人工差错控制**。

当一个终端通过模拟PSTN和调制解调器连接远程计算机时，使用的方法相似。代替每个输入字符直接显示在终端屏幕，它首先发送到远程计算机，后者读出并存储字符并重新发送返回到终端显示。如果显示的字符与想要的字符或输入字符不同，那么可再开始一个合适删除字符传送。这种差错控制方式称**回送检测**。

可是，当一台计算机经过串行数据链路向另一台计算机传送字符块（帧），接收计算机的接收过程控制程序必须自动地执行差错控制方法，无需任何用户介入。典型地，接收计算机检测接收到帧的可能传输差错，然后返回一个短控制信息（帧），确认它正确收到或者请求发送帧的另一个备份。这种差错控制类型称为**自动重发请求（ARQ）**。 169

ARQ有两种基本类型：**空闲RQ**，它用于面向字符（或面向字节）数据传输方案；**连续RQ**既可采用**选择重发策略**，又可采用**回退N帧重传策略**。连续RQ原先用于面向位传输方案，虽然空闲RQ在许多应用中被效率高的连续RQ代替，但仍然有许多数据链路协议采用空闲RQ。然而，更重要的是，由于它是差错控制方案最简单类型，它是说明数据链路（控制）协议许多更一般问题的理想模式，我们将讨论空闲RQ和连续RQ方案。

4.2 空闲RQ协议

空闲RQ差错控制方案定义由打印字符与格式控制字符组成的块（帧）能够经过串行数据链路在源DTE与目标DTE之间做到可靠传送，即提交一个高概率、无差错或无重复的相同时序的传送。为了区别发送器（源）和接收器（目标）的数据帧（更一般地称为**信息帧或I帧**），通常分别使用术语**主站（P）**与**从属站（S）**。因此，空闲RQ差错控制方案牵涉到主站与从属站间经过串行数据链路可靠地传送I帧。

空闲RQ协议以半双工方式操作，因为主站发送一个I帧后必须等待直到收到来自从属站指示该帧是否正确收到的一个说明。然后，如果前面帧正确收到，主站发送下一帧，如果没有正确收到，主站重发前一帧的备份。

有两种方法实现这个方案。一种为**隐式重传**，从属站S仅确认正确接收到的帧，而主站P解释没有确认指明前一帧损坏；另一种为**显式请求**，当从属站S检测到损坏帧，它返回一个**否认确认**请求发送帧的另一个备份。

隐式重传控制方案的帧序列实例如图4-1(a)所示。当解释帧序列时，应该注意下列几点：

- 主站P每次只能有一个待确认I帧（等待**确认帧或ACK帧**）；
- 从属站S接收到一个无差错I帧，即向主站P返回一个ACK帧； 170
- 主站P接收到一个无差错ACK帧，主站P可发送另一个I帧（见图4-1(i)部分）；
- 当主站P开始发送I帧，启动定时器；
- 当从属站S检测出收到的是含有差错的I帧或主站P收到含有差错的ACK帧时，则丢弃该帧；
- 如果主站P在预定时间间隔（**超时间隔**）未收到ACK帧，则主站P重发待确认的I帧（见图4-1(ii)部分）；
- 如果ACK帧损坏，则从属站S收到帧的另一备份，应丢弃这个重复（见图4-1(iii)部分）。 171

正如我们在（i）部分所见，主站P开始发送一帧后，在发送下一帧之前必须等待一个最小时间，这个等待时间等于从属站S接收与处理I帧时间加上ACK帧发送与处理时间。在最坏的情况下，主站P等待时间为超时间隔，超时间隔必须等于最小时间加上一个适当余量，以避免前一帧的另一备份已被重传后收到ACK帧。

构成等待最小时间各部分的相对时间随数据链路类型不同而各不相同。如随两个通信系

172 统（主站P与从属站S间）的物理距离与链路的数据传输速率而异。然而，一般利用有效链路容量，每当从属站S收到一个损坏I帧，从属站S立即返回一个**否认确认帧或NAK帧**通知主站P，获得很大的改进。这种方案的帧序列实例如图4-1(b)所示。

当解释帧序列时，应注意下列几点：

- 正如隐式确认方案的情况一样，接收到一个无差错I帧，从属站S向主站P返回一个ACK帧；
- 接收到一个无差错ACK帧，主站P停止定时器，然后开始另一个I帧发送（见图中（i）部分）；
- 如果从属站S接收到一个有差错I帧，则丢弃该帧并返回一个NAK帧（见图中（ii）部分）；
- 如果主站P在超时间隔未收到ACK帧（或NAK帧），则主站P重发待确认的I帧（见图中（iii）部分）；

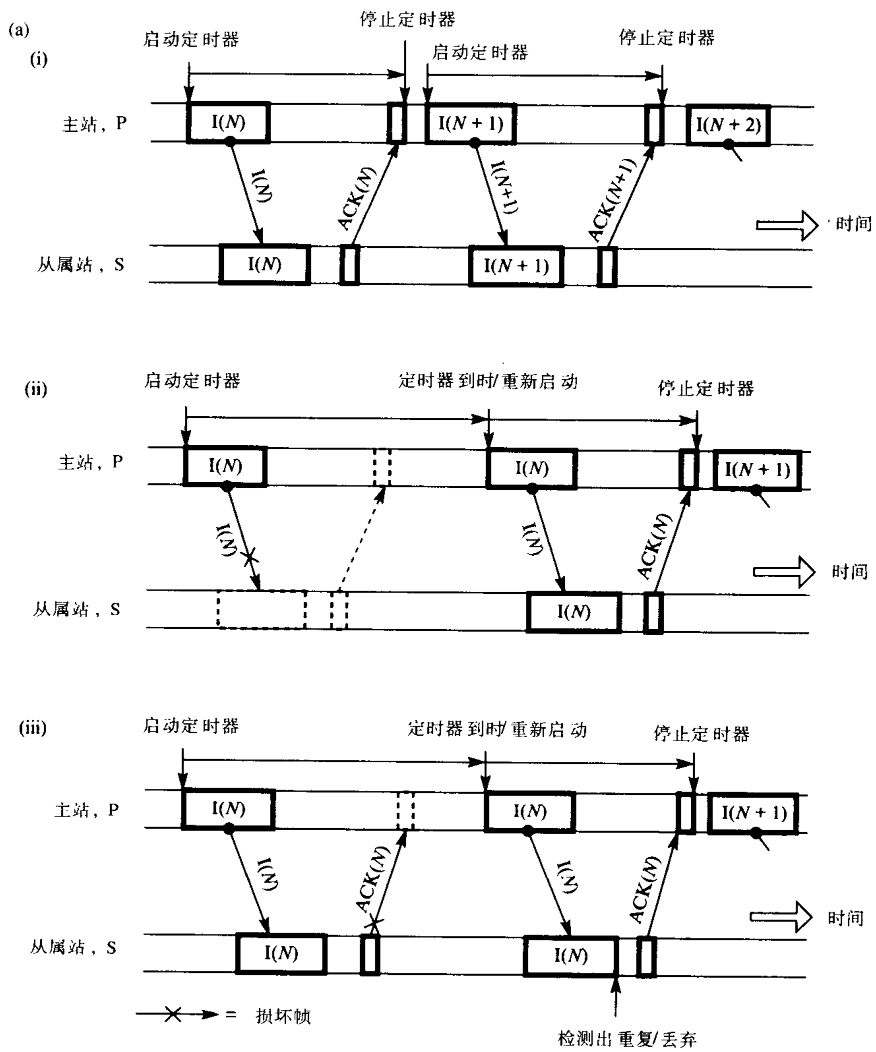


图4-1 空闲RQ操作

(a) 隐式重传 (b) 显式请求

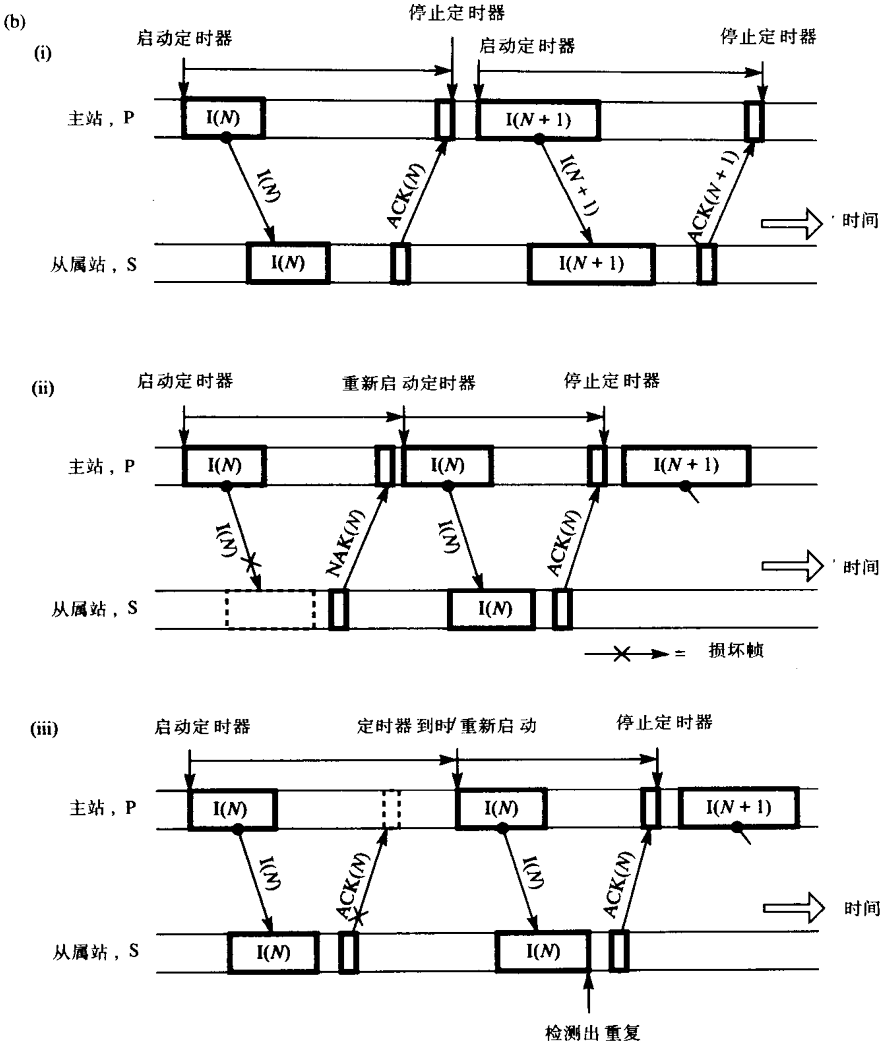


图4-1 （续）

因为空闲RQ方案中，主站发送一个帧后必须等待一个确认帧，所以这种方案也称为**发送等待**或**停止等待**。正如从图4-1所见，该方案保证从属站S至少可接收到主站P发送各帧的一份备份。但是，两种方案，对于从属站S可能接收到某个特定I帧的两份（或多份）备份，这称之为**重复**。为了从属站S能区分下一个有效I帧（期待的帧）与一个重复帧，每个发送帧包含惟一的称为**序列号**的标识符（如N、N+1等），如图4-1所示，这样，从属站S在正确接收的最后一个I帧中保存序列号的记录，如果从属站S接收到这个帧的另一个备份，则丢弃这份备份。为了使主站重新同步，从属站S对每一个正确接收的帧均返回具有相关I帧标识符的ACK帧。

我们通过考察每个方案的（ii）部分的帧序列，采用显式请求方案在链路利用上得到改善。隐式重传，下一个I帧发送的时间是超时间隔，而用NAK帧这个时间更短。链路利用的相对改进归根结底与误码率（BER）有关，因而也与需要重传的损坏帧个数有关。然而，大多数数据通信应用中采用带有NAK帧显式请求方案的空闲RQ协议。

每个I帧中携带的序列号称为**发送序列号**或**N(S)**，每个ACK帧或NAK帧中携带的序列号称为**接收序列号**或**N(R)**。如在第3章描述过ASCII（与EBCDIC）字符集包含一些控制字符（例

如STE、ETX），其中某些用作传输控制字符。实现基本空闲RQ差错控制方案，需要三个传输
173 控制字符：SOH（标题起始符），NAK与ACK，它们的用法如图4-2所示。

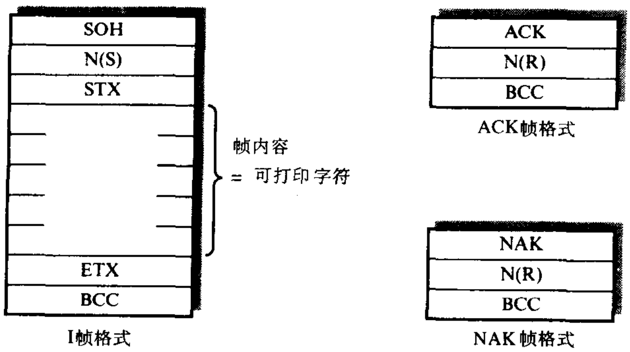


图4-2 空闲RQ帧（PDU）格式

每个I帧在帧头部包含一个序列号，如我们所见，它在STX字符前面。在完整块的头部添加SOH字符，使得惟一的字符告知一个新帧开始的信号。

用作确认目的的ACK与NAK控制字符后跟接收序列号，为了提高差错检测概率，完整NAK帧或ACK帧包含一个块校验和字符。三个帧：I帧、ACK帧和NAK帧，合称为空闲RQ协议的**协议数据单元**（PDU），而主站P与从属站S 称为**协议实体**。

4.2.1 层次结构

图4-1所示的帧序列图说明空闲RQ协议操作的基本特性，我们继续进行有关协议的更详细讨论前，引入**分层**的概念。这牵涉到去除应用与通信任条的耦合，建立两个意义明确的子任务或**层**，在这两个层之间有一个正式接口。

为了说明这个，考虑用空闲RQ差错控制协议在一台计算机运行的**应用进程**（AP）经过串行数据链路发送一个数据文件到第二台计算机运行的相似应用进程。如同已描述过的，空闲RQ协议将尽力用一个可靠方法经过串行数据链路发送一系列信息块（可打印字符或字节）。可靠的方法依赖于链路的BER，规定的最大块长度保证高概率，即无差错发送I帧的好的百分比。

因此，源计算机空闲**RQ协议层**向其上用户AP层提供一个规定**服务**。它传送一系列信息块，每块规定最大长度，并送到目标计算机的相应（对等）AP。两个对等的空闲RQ协议实体牵涉到与先前讨论的差错有关的各种问题：帧的生成，确认帧的返回，超时与信息块以提交的相
174 同顺序传送等。

可是，两个对等AP仅牵涉使用通信（空闲RQ）层提供的服务传送文件，如文件名、文件长度，在向通信层提供文件内容前分段为较小的块与接收时重新把块组装成一个完整文件。因此，对每个应用，期待的信息块的序列必须定义它们的语法与结构。这暗含着AP到AP协议具有自己的一组（AP到AP）PDU，然后两个AP协议实体用下层通信服务传送它们自己的PDU。但是，对于通信层，所有这些信息块都简单地以同一方法传送。

一般，通信层提供的服务以带有传送数据（通常称为**用户数据**）作为**参数**的**服务原语**表示。因为服务关系到链路层（L）与数据（信息块）传送，所以在发送接口的用户服务原语表示作L_DATA.request，接收接口的用户服务原语表示作L_DATA.indication 如图4-3(a)所示。

在许多情形下，因为一个层的用户只考虑提供服务而不管服务如何实现。当我们定义有关（协议）层的服务时，我们通常用图4-3(b)所示的形式表示它们。这称为（服务）**时序图**。

用一个清楚的方法分开两层，我们在它们之间引入一个**队列**。如图4-3(c)所示。这是一个实现简单**先进先出**（FIFO）队列规则的数据结构，元素加入队列的**尾部**并从**头部**移出。

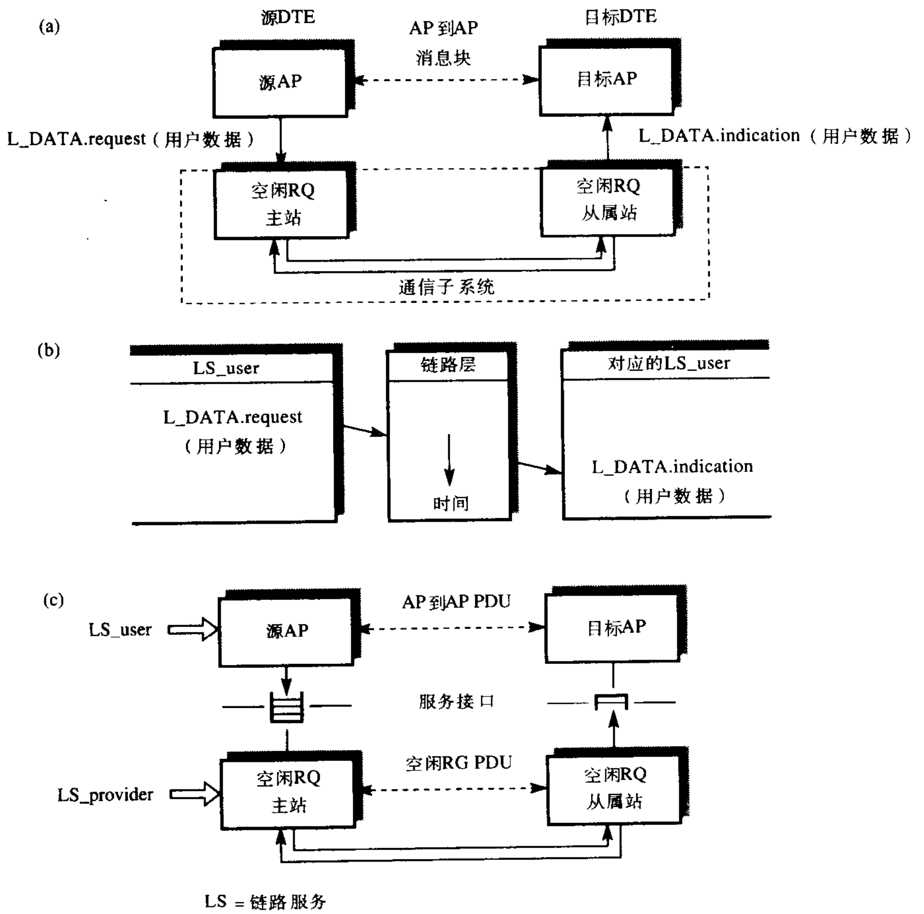


图4-3 层次结构

(a) 服务原语 (b) 时序图 (c) 服务接口

通常，某层的用户服务原语用一个称为**事件控制块**（ECB）的数据结构在两层之间传递。一般，它是一个记录或结构，其中第一个字段是原语类型，而第二个字段是字符或包含用户数据的字节数组。每当源AP（或高协议层）想要发送一个信息块，通过空闲RQ的主站，首先得到一个空闲的ECB，用字符或字节数组写入用户数据段，设置L_DATA.request的原语类型字段，并将ECB插入链路层（LS_用户）输入队列的尾部准备读出。

当空闲RQ协议实体软件下一次运行时，它检测链路层（LS_用户）输入队列的实体（ECB）出现与否，从队列头部读出该项，用信息块（字符/字节数组的内容）作为帧内容并加上适当的帧头与帧尾建立一个I帧。然后开始向从属站协议实体发送帧。假定接收帧没有差错，从属站实体除掉头部与尾部的字符，把帧的内容（信息块）向上移动到目标AP的ECB中，通过把链路层（LS_provider）输出队列原语类型设置为L_DATA.indication，然后，建立与返回一个ACK帧到主站P。

当目标AP下一次运行时，它从LS_provider队列检测并读出ECB，按照规定AP到AP协议

175 继续处理信息块的内容。典型的，如果这是第一个信息块，比如说，它包含的是文件名，则用这个名创建一个文件，并打开它为后续写/附加文件操作做准备。

在发送方，假定接收到没有差错的ACK帧，主站P释放内存缓冲器中已确认过的I帧，对另外的等待ECB检测LS_user输入队列。如果有一个，则过程重复直到所有文件段已被发送。一般源AP发送一个传送结束信息块通知目标AP，完整文件内容已被传送。

我们可得出结论，采用分层结构就是说，每一层执行整个通信任务范围内它自己的意义明确的功能（例如，文件传送）。低层（数据链路层）向上一层提供规定服务，并按照规定协议操作。我们将在本书的第三部分看到，分层结构就是说更复杂通信子系统可通过增加应用

176 层与数据链路层之间的层次来实现。每个新层关系到补充功能。

4.2.2 协议规范说明

虽然，结合图4-1中各部分展示的三帧序列图的描述文本足以说明空闲RQ协议的操作，但是对较复杂协议，只是用这个方法实际上不可能完整描述协议的操作。我们在本章与下一章将会确切地看到这一点，定义协议所有可能发生事件与一些出现的差错情况的操作是很复杂的。所以，一般我们采用更严密的形式与方法详细描述协议。如图4-1所示的帧序列图只是简单说明协议某个方面，而不能当作描述协议的工具。

描述通信协议最常用三种方法是**状态迁移图**、**扩充事件—状态表**和高级**结构化程序**。在多数情况下，将这三种方法的组合并与相应的帧时序图结合起来，对协议的用户服务原语进行说明。

不管采用什么方法，总可以把协议模型化为一个**有限状态机**或**自动机**。这就是说协议（或者更正确地说协议实体）在任何时刻所处的状态都是所定义若干**状态**之一，例如：等待发送信息或等待接收确认。状态之间迁移是**入事件**发生的结果，例如信息准备好发送或接收到ACK帧。由于入事件引起结果产生一个相应的**出事件**，例如，接收信息，在链路上发送I帧或接收到NAK帧，重发等待确认的I帧。

显然，有些入事件可能产生若干个出事件，由一个或多个**谓词**（布尔变量）计算状态决定选中哪一个特殊出事件。例如如果接收到ACK帧的N(R)等于等待确认的I帧的N(S)，谓词P1可能为真。因而，P1为真，释放内存缓冲器中保存的I帧。P1为假，开始重发该帧。

最后，一个入事件，除了产生出事件（可改变状态），也可能伴有一个或多个相关的**本地动作**或**规范动作**，例如，启动定时器与发送序列号变量增1两个特定动作。

现在我们将通过考察空闲RQ协议的差错控制过程的详细说明阐述协议规范说明的各个方面。

4.2.3 空闲RQ协议规范

有限状态机（因此协议实体）以**原子**方法操作。就是说一旦入事件开始处理，所有与此

177 事件相关的处理功能，包括出事件的发生、本地（特定）动作与状态的可能改变，在另一入事件被接受前这一切都作为单一整体执行（即以不可分方式）。

为了保证做到这一点，利用图4-4所示队列各种入事件（与出事件）接口与协议实体本身解耦合。协议实体与发送—接收过程之间有一对附加队列，发送与接收过程控制特定的传输控制线路。同样的，协议实体与定时器过程之间也有一对队列。一般，后者以**中断**的方式在标准（滴答）时间间隔内运行，如果定时器正在运行，它的现行值按照滴答值递减。如果定

时器值为零，则**定时器到时**消息通过适当的队列返回到协议实体。

发送—接收过程是发送一个预先已格式化帧或从链路私队列接收一个帧，并按协议实体处理帧。也可中断引起运行，但此时来自传输控制电路。最后，虽然，原则上主站与从属站到它们各自的AP接口仅需要一个输入与输出队列，但实际上为了处理原语双向流，在每个接口需要一对队列。

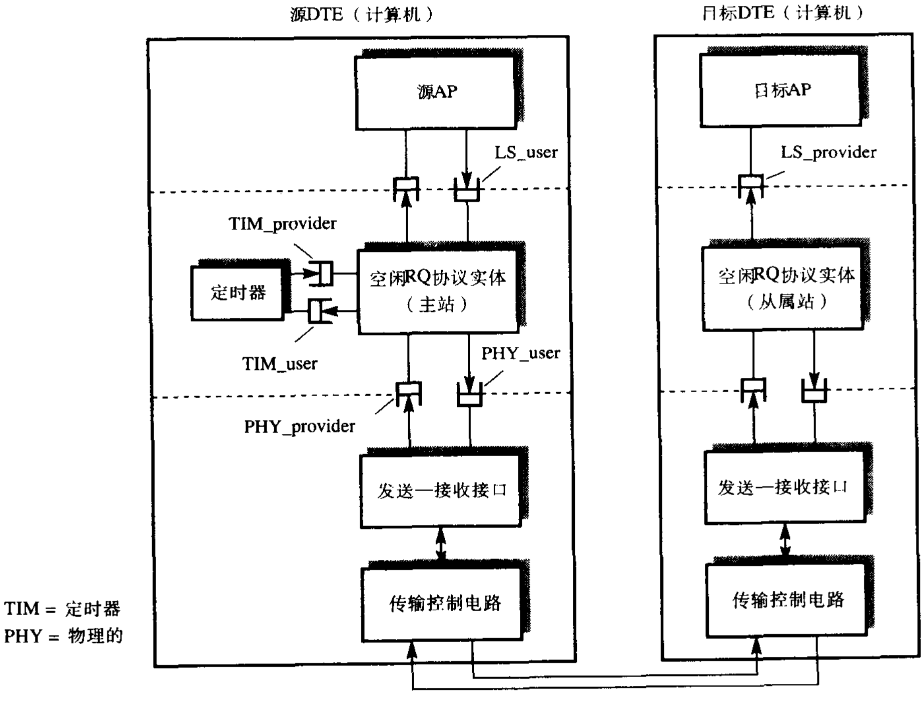


图4-4 通信子系统结构与协议实体接口

为了简化规范说明，我们给出与每个协议实体相关的所有入事件、出事件、谓词、特定动作与状态的缩写名。在说明协议之前，一般列出各种缩写名，以后均参照这些名称。对空闲RQ协议的差错控制组件，主站与从属站的缩写名表分别如图4-5(a)与图4-5(b)所示。 178

因为每个协议实体本质上是一个时序系统，我们必须保留随接收到不同入事件变化的信息。这个信息保留在若干个**状态变量**中。对于主站，实例是发送序列变量V(S)（规范说明中为Vs），它保留分配的序列号给下一个发送的I帧，PresentState变量是协议实体的现在状态，RetxCount变量是用于限制重传帧的数目，ErrorCount变量是接收到有差错帧的计数。典型地，如果RetxCount或ErrorCount达到它的最大极限值，则帧丢弃，并向上面AP层发送一个差错消息并重新初始化协议（实体）。

从属站只需要两个状态变量：接收序列变量V(R)（在规范说明中为Vr），它保留最后正确接收到I帧的序列号，而ErrorCount保持接收到差错帧的个数记录。如果ErrorCount达到它规定的极限，向上面AP层发送一个差错消息。 179

主站和从属站的形式化规范说明以状态迁移图与扩展事件状态表的形式表示，分别如图4-6与图4-7所示。

(a)

入事件

缩写名	接口类别	含　义
LDATA req	LS_user	接收到服务原语L_DATA.request
ACKRCVD	PHY_provider	接收到来自从属站的ACK帧
TEXP	TIM_provider	Wait_ACK定时器到时
NAKRCVD	PHY_provider	接收到来自从属站的NAK帧

状态

缩写名	含　义
IDLE	空闲，没有消息在传递
WTACK	等待确认

出事件

缩写名	接口类别	含　义
TxFrame	PHY_user	格式化与发送I帧
RetxFrame	PHY_user	重发等待确认I帧
LERRORind	LS_provider	差错消息：为特定理由而丢弃帧

谓词

缩写名	含　义
P0	等待I帧的N(S) = ACK帧的N(R)
P1	ACK/NAK帧的块和校验（BSC）正确

特定动作

[1]=用TIM_user队列启动定时器
[2]=Vs增1
[3]=用TIM_user队列停止定时器
[4]=RetxCount增1
[5]=ErrorCount增1
[6]=RetxCount复位为0

状态变量

Vs=发送序列变量
PresentState = 协议实体现在状态
ErrorCount = 接收差错帧的个数
RetxCount = 帧重发次数

(b)

入事件

缩写名	接口类别	含　义
IRCVD	PHY_provider	接收到来自主站的I帧

状态

缩写名	含　义
WTIFM	等待来自主站的一个新I帧

出事件

缩写名	接口类别	含　义
LDATAind	LS_provider	用L_DATA.indication原语传递接收到I帧到用户AP
TxACK（X）	PHY_user	用N(R) = X格式化并发送一个ACK帧
TxNAK（X）	PHY_user	用N(R) = X格式化并发送一个NAK帧
LERRORind	LS_provider	为特定理由发送差错消息

谓词

缩写名	含　义
P0	I帧的N(S) = Vr
P1	I帧的块和校验（BSC）正确
P2	I帧的N(S) = Vr − 1

特定动作

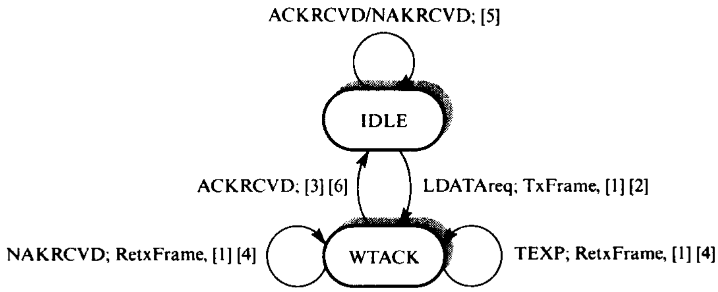
[1]=Vr增1
[2]=ErrorCount增1

状态变量

Vr=接收序列变量
ErrorCount=接收差错帧的个数

图4-5　空闲RQ规范说明所用缩写名
(a) 主站　(b) 从属站

(a)



(b)

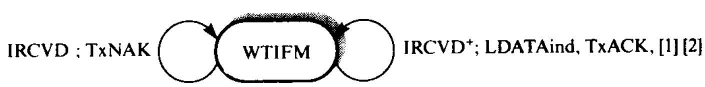
Present state \ Incoming event	LDATAreq	ACKRCVD	TEXP	NAKRCVD
IDLE	1	0	0	0
WTACK	0	2	3	3

0 = [5], IDLE（差错条件）
1 = TxFrame, [1] [2], WTACK
2 = P0 and P1: [3] [6], IDLE
    P0 and NOT P1: RetxFrame, [1] [4], WTACK
    NOT P0 and NOT P1: [5], IDLE
3 = RetxFrame, [1] [4], WTACK

图4-6 空闲RQ协议规范说明——主站

(a) 状态迁移图 (b) 扩充事件—状态表

(a)



(b)

Present state \ Incoming event	IRCVD
WTIFM	1

1 = NOT P1: TxNAK, [2]
    P1 and P2: TxACK
    P0 and P1: LDATAind, TxACK, [1]

图4-7 空闲RQ协议规范说明——从属站

(a) 状态迁移图 (b) 扩充事件—状态表

用状态迁移图方法，协议实体的可能状态用其中带有特定状态的椭圆形表示。**有向箭头**（也称**弧**）指明状态间的可能迁移，由入事件引起迁移，结果的出事件与特定动作在入事件旁边标明。例如，如果接收到来自LS_user接口的一个L_DATA.request（LDATAreq），则格式化帧并输出该帧到PHY_user接口（TxFrame），定时器启动[1]与发送序列号变量增1[2]并进入WTACK状态。同样的，如果接收到ACK帧，它的N(R)等于等待帧的N(S)，同时块校验和正确，则定时器停止[3]与重发计数器复位为0[6]，进入IDLE状态，其他过程按相同方法解释。

虽然状态迁移图对表示协议的正确操作是有用的，但由于空间限制表示所有可能入事件包括的差错条件不切实际。因此，大多数状态迁移图是不完全的规范说明。此外，除最简单 180
协议外，对所有的协议，即使协议的操作正确，我们仍需要许多图去定义。由于这些理由，我们使用事件—状态表与结构化程序代码方法。

利用扩展事件—状态表方法，如图4-6与图4-7所示，我们可用表的形式表示所有可能入事件与协议（现在）状态。每个状态对于所有可能入事件，表项定义出事件，特定动作与新状态。如果有谓词，也包含一些可选的动作，显然，由于它允许所有可能入事件与现在状态的
181 各种组合，扩充事件—状态表是一个精确得多的方法。基本事件—状态表对每个入事件只有一个可能动作和下一状态。谓词的出现（因此有可选的动作/下一状态）产生术语“扩展”事件—状态表这一用法。

当我们解释动作后如果跟有谓词，我们必须注意它们的顺序。如主站处于WTACK状态，并接收到ACK帧（ACKRCVD），接下来的动作首先要确定P0∧P1是否为真，如果是真，则产生特定动作[3]与[6]并进入IDLE状态，否则确定{ P0∧NOTP1}是否为真等等。如果条件也不是真，那么怀疑有错，如所示动作。

扩展事件—状态表的特点使得它比状态迁移图更容易用程序代码实现，我们考察一个实例，看看空闲RQ主站与从属站的实现。它们的概要结构分别如图4-8(a)与图4-8(b)所示。为了易读性它们用高级伪码编写，每个站用一个独立程序表示。实际上，它们是过程，但这不影响基本操作。

当每个程序运行时，首先初始化过程，执行的功能如初始化状态变量为初始值与设置EventStateTable数组为扩展事件—状态表的内容。然后程序进入无穷循环等待输入队列的某一入事件到达。

```
(a)  program   IdleRQ_Primary;
     const     MaxErrCount;
               MaxRetxCount;
     type      Events = (LDATAreq, ACKRCVD, TEXP, NAKRCVD);
               States  = (IDLE, WTACK);
     var       EventStateTable = array [Events, States] of 0..3;
               PresentState : States;
               Vs, ErrorCount, RetxCount : integer;
               EventType : Events;
     procedure Initialize;     } Initializes state variables and contents of EventStateTable
     procedure TxFrame;        }
     procedure RetxFrame;      } Outgoing event procedures
     procedure LERRORind;      }
     procedure Start_timer;    }
     procedure Stop_timer;     } Specific action procedures
     function  P0 : boolean;   }
     function  P1 : boolean;   } Predicate functions

     begin     Initialize;
               repeat Wait receipt of an incoming event
                      EventType := type of event
                      case EventStateTable [EventType, PresentState] of
                         0 : begin ErrorCount := ErrorCount + 1; PresentState = IDLE;
                                   if(ErrorCount = MaxErrCount) then ERRORind end;
                         1 : begin TxFrame; Start_timer; Vs := Vs + 1; PresentState := WTACK end;
                         2 : begin if(P0 and P1) then begin Stop_timer; RetxCount := 0; PresentState := IDLE end;
                                   else if (P0 and NOTP1) then begin RetxFrame; Start_timer;
                                                                      RetxCount := RetxCount + 1;
                                                                      PresentState := WTACK end;
                                   else if (NOTP0 and NOTP1) then begin PresentState := IDLE; ErrorCount := ErrorCount + 1 end;
                                           if (ErrorCount = MaxErrorCount) then begin LERRORind; Initialize; end;
                             end;
                         3 : begin RetxFrame; Start_timer; RetxCount := RetxCount + 1; PresentState := WTACK;
                                   if (RetxCount = MaxRetxCount) then begin LERRORind; Initialize; end;
                             end;
               until Forever;
     end.
```

图4-8 空闲RQ规范说明

(a) 主站伪代码 (b) 从属站伪代码 (c) Estelle主站概要结构

(b)
```
program   IdleRQ_Secondary;
const.    MaxErrorCount;
type      Events = IRCVD;
          States = WTIFM;
var'      EventStateTable = array [Events, States] of 1;
          EventType : Events;
          PresentState : States;
          Vr, X, ErrorCount : integer;
procedure Initialize;        } Initializes state variables and contents of EventStateTable
procedure LDATAind(X); )
procedure TxACK(X);    |
procedure TxNAK(X);    } Outgoing event procedures
procedure LERRORind;   )
function  P0 : boolean; )
function  P1 : boolean; } Predicate functions
function  P2 : boolean; )

begin     Initialize;
          repeat Wait receipt of incoming event; EventType := type of event;
                 case EventStateTable[EventType, PresentState] of
                    1 : X := N(S) from I-frame;
                      if (NOTP1) then TxNAK(X);
                      else if(P1 and P2) then TxACK(X);
                      else if(P0 and P1) then begin LDATAind; TxACK(X); Vr := Vr + 1; end;
                      else begin ErrorCount := ErrorCount + 1; if (ErrorCount = MaxErrorCount) then
                               begin LERRORind; Initialize; end;
                           end;
          until Forever;
end.
```

(c)
```
  – Type and constant definitions
  – Interface definitions {interface queue names and ECB record structure definitions}
  – Formal finite state machine definition.

        module
           – states
           – incoming events
           – outgoing event procedures
           – specific action procedures
           – predicate functions
        trans (* begin state transition definitions *)
           from IDLE to IDLE
           when PHY_provider.ACKRCVD
                begin
                end;
           from IDLE to IDLE
           when TIM_provider.TEXP
                begin
                end;
                  .
                  .
                  .
           from WTACK to IDLE
           provided P0 and P1
           when PHY_provider.ACKRCVD
                begin
                end;
                  .
                  .
                  .
        end.
```

图4-8 （续）

引起程序运行的入事件，首先指定EventType。然后PresentState与EventType的现行内容用作索引EventStateTable数组确定整数值（0、1、2或3）以定义那个事件相关的处理动作。例如，如果使用的整数是2，则其结果为调用谓词函数P0 $\wedge$P1，依赖计算结果（真或假），调用相应出事件过程并与规范说明定义的特定动作过程结合。例如，启动或复位定时器与更新PresentState。

我们简化了伪码以突出每个程序的结构和实现方法。没有为各种出事件过程及谓词函数

显示代码。实际上，这些必须使用规范说明中列出的必要步骤以明显的方式实现。

虽然许多协议是以扩充事件—状态表形式说明，但有一些协议是用结构化程序代码说明。规范说明类型与图4-8所示的伪码实现相似。但是，因为所有要求的变量与各种出事件与特定动作过程与功能都有详细的表述，所以结构代码更加形式化。因此，协议规范说明用结构代码实现更易懂，并易于减少差错。

为了比较目的，用结构化代码方法表示的空闲RQ主站的差错控制过程的概要结构如图4-8(c)
182 所示，这种方法称为**扩展的状态迁移模型**（Estelle）。我们可看到，它除掉实际的协议实体以**模块**形式定义和所有可能移迁以更正规方式各自定义外，相似于图中（a）与（b）部分规范说明的结构。实际中，仅有少数规范说明采用这种形式，大部分采用扩充事件—状态表的形式。应用概要实现方法，可生成相似结构化程序代码。

4.2.4 链路利用

在考察连续RQ协议两种类型的差错控制方法之前，我们首先对空闲RQ协议有效链路容量利用率进行量化的讨论，利用率U是两个时间之比，每个都从发送方开始发送这一点度量。由下式决定：

$$U = \frac{T_{\text{ix}}}{T_{\text{t}}}$$

其中T_{ix}是发送帧的时间，T_{t}是T_{ix}加上发送器等待确认所花费的时间。

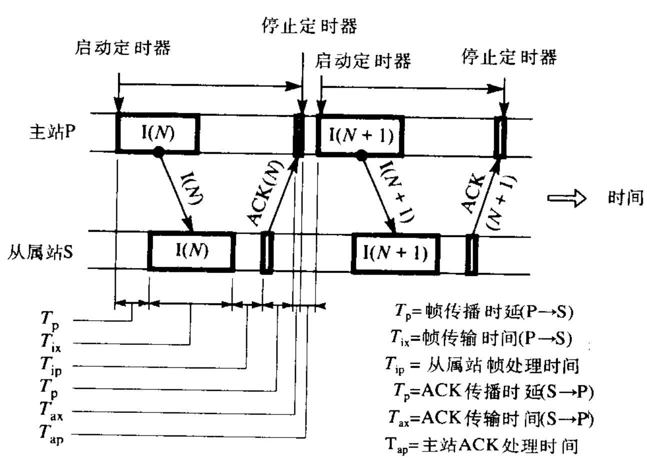


图4-9　空闲RQ链路利用结构

为了确定空闲RQ的链路利用率的值，在图4-9中给出用不同时间部分标识的帧序列图。实际上，空闲RQ协议的大多数情况，I帧的处理时间T_{ip}和ACK帧处理时间T_{ap}分别小于它们的
183～184 传输时间T_{ix}与T_{ax}，并且由于ACK帧远短于I帧，则相对于T_{ix}，T_{ax}是可以忽略的，因而到下一帧发送之前总的最小时间约为$T_{\text{ix}} + 2T_{\text{p}}$。因此$U$的近似公式为：

$$U = \frac{T_{\text{ix}}}{T_{\text{ix}} + 2T_{\text{p}}}$$

或

$$U=\frac{1}{1+2T_{\mathrm{p}}/T_{\mathrm{ix}}}$$

正如2.4节所描述，比率 T_p/T_{ix} 经常用一个符号a表示，因此

$$U=\frac{1}{1+2a}$$

在实例2-6中，我们看到，a可以从中等距离低比特率链路的一个很小数到长距离链路高比特率的一个大的整数之间变化。对于两个极端，U在接近1（100%）与一个很小数之间变化。

实例4-1

使用空闲RQ协议，发送1000位的帧。设数据传输速率为（i）1 kbps（ii）1 Mbps。链路传播速度为每秒2×10^8m，忽略数据误码率。求下列各数据链路的利用率。

（a）1 Km双绞线电缆；

（b）200 Km租用线路；

（c）50 000 Km卫星链路。 185

解：发送一个帧所需的时间 T_{ix} 为

$$T_{\mathrm{ix}}=\frac{\text{每帧位的个数}N}{\text{比特率}R}$$

对于1 kbps

$$T_{\mathrm{ix}}=\frac{1000}{10^3}=1\,\mathrm{s}$$

对于1Mbps

$$T_{\mathrm{ix}}=\frac{1000}{10^6}=10^{-3}\mathrm{s}$$

$$T_{\mathrm{p}}=\frac{S}{V}\qquad \text{和}\qquad U=\frac{1}{1+\dfrac{2T_{\mathrm{p}}}{T_{\mathrm{ix}}}}=\frac{1}{1+2a}$$

所以

(a) $T_{\mathrm{p}}=\dfrac{10^3}{2\times10^8}=5\times10^{-6}\mathrm{s}$

(i) $a=\dfrac{5\times10^{-6}}{1}=5\times10^{-6}$ 因此 $(1+2a)\simeq1,\ U=1$

(ii) $a=\dfrac{5\times10^{-6}}{10^{-3}}=5\times10^{-3}$ 因此 $(1+2a)\simeq1,\ U=1$

(b) $T_{\mathrm{p}}=\dfrac{200\times10^3}{2\times10^8}=1\times10^{-3}\mathrm{s}$

(i) $a=\dfrac{1\times10^{-3}}{1}=1\times10^{-3}$ 因此 $(1+2a)\simeq1,\ U=1$

(ii) $a=\dfrac{1\times10^{-3}}{10^{-3}}=1$ 因此 $(1+2a)>1,\ U=\dfrac{1}{1+2}=0.33$

$$(c)\ T_p = \frac{50\times10^6}{2\times10^8} = 0.25\text{s}$$

$$(i)\ a = \frac{0.25}{1} = 0.25 \qquad \text{因此}\ (1+2a)>1,\ U = \frac{1}{1+0.5} = 0.67$$

186 $$(ii)\ a = \frac{0.25}{10^{-3}} = 250 \qquad \text{因此}\ (1+2a)>1,\ U = \frac{1}{1+500} = 0.002$$

计算结果如图4-10所示。从中可以得出有趣结论：首先，对于较短的链路，且a小于1时，链路利用率近似为100%，并与所用的数据速率无关。这就是说，空闲RQ协议最适用于短链路和适中的数据速率。基于调制解调器与模拟公用交换电话网（PSTN）的网络是实例。其次，对于较长地面链路，数据速率较低时（因此a值低），链路利用率很高，但当数据速率（因此a也增加）增高时，利用率显著下降。第三，对于卫星链路，即使是在较低数据速率的情况下，链路利用率也很低。由此可以得到结论：空闲RQ协议不适用于卫星链路与高速率的地面数据链路，如局域网（参见第6章）与大多数公用载波WAN（参见第8章）。

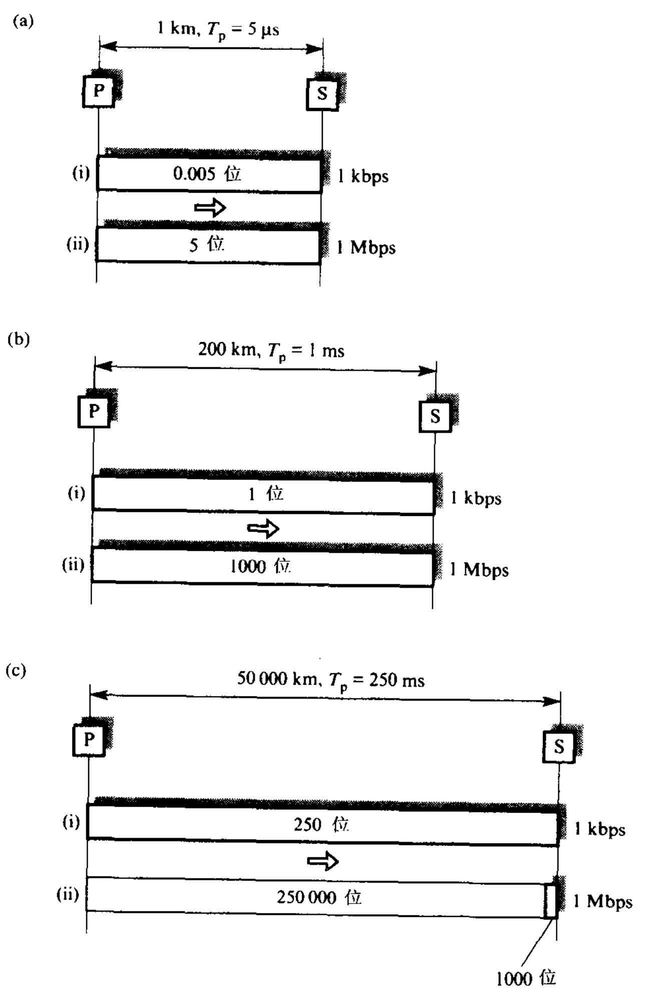


图4-10 数据传输率对传播延迟的影响；对应于实例4-1的部分

实例4-1是假设在无传输差错情况下计算链路利用率。实际上，链路传输总是会出现差错（非零BER），因而，为成功地发送一个帧，平均每帧需要进行N_r次重发，这时，链路利用率表达式可修改为： 187

$$U=\frac{T_{ix}}{N_r T_{ix}+2N_r T_p}=\frac{1}{N_r\left(1+\frac{2T_p}{T_{ix}}\right)}$$

我们可以由链路的位差错率（BER）P求出N_r的值。如果P是一位可能发生差错的概率，则假定差错是随机的，无差错接收到长度为N_i的一帧的概率为

$$P_f = 1-(1-P)^{N_i}$$

因此，一帧可能发生差错的概率为：

$$\begin{aligned}P_f &= 1-(1-P)^{N_i}\\ &\simeq N_i P \text{ if } N_i P \ll 1\end{aligned}$$

例如，P_f=10^{-4}就是说平均每传输10^4位将有1位发生差错。因而，对于长度为1000位的帧，帧的位差错率P_f=1-（1-10^{-4}）^{1000}=0.095或P_f=$10^3\times10^{-4}$=0.1，即平均传输10帧将有1帧发生差错。

现在，如果P_f是帧的差错率，则1-P_f是帧无差错率。因此

$$N_r=\frac{1}{1-P_f}$$

例如，如果P_f = 0.5，则N_r = 2，即如果有50%的帧出现差错，则平均每帧将被发送2次。当然，这里假设ACK帧不发生差错。由于ACK帧远短于I帧，出错率要低得多，因而这个假设是合理的。所以实际上，所有链路效率值都要除以N_r，即

$$U=\frac{1-P_f}{1+2a}$$

空闲RQ方案的主要优点是它需要最少的缓冲存储空间，这是因为主站P与从属站S都只须存储一帧的缓冲容量。并且为了能保证检测出重复，从属站S仅需保留上次正确接收帧的标识符记录。各种重发方案为提高传输效率都需要缓冲存储空间。然而，空闲RQ方案的缓冲存储空间是最小的，因此在链路端使用简单终端设备（例如，终端或个人计算机）的环境中被广泛使用。 188

4.3 连续RQ协议

连续RQ差错控制方案以增加缓冲存储空间为代价，改善链路利用率。我们将看到要求实现全双工的链路，说明传送I帧序列与它们的返回ACK帧的传输实例如图4-11所示。

当解释方案操作时，注意下列几点：

- 主站P连续发送I帧不等待ACK帧返回；
- 由于有多个I帧在等待确认，主站P在**重传列表**中保存所发送的每个I帧的备份，重传列表按先进先出（FIFO）队列规则操作；
- 从属站S对每个正确收到的I帧返回一个ACK帧；
- 每一个I帧有一个惟一的标识符，该标识符在相应的ACK帧中返回；
- 接收到一个ACK帧，主站P从重传列表中删除相应的I帧；
- 正确收到I帧保存在**链路接收列表**中等待处理； 189

- 接收到下一个期待顺序的I帧，从属站S立刻将已处理的帧传递给上层（LS_user）；
- 接收到失序的一些帧（参见4.3.1节），从属站S将这些帧保存在链路接收列表中，直到接收到下一次序的帧。

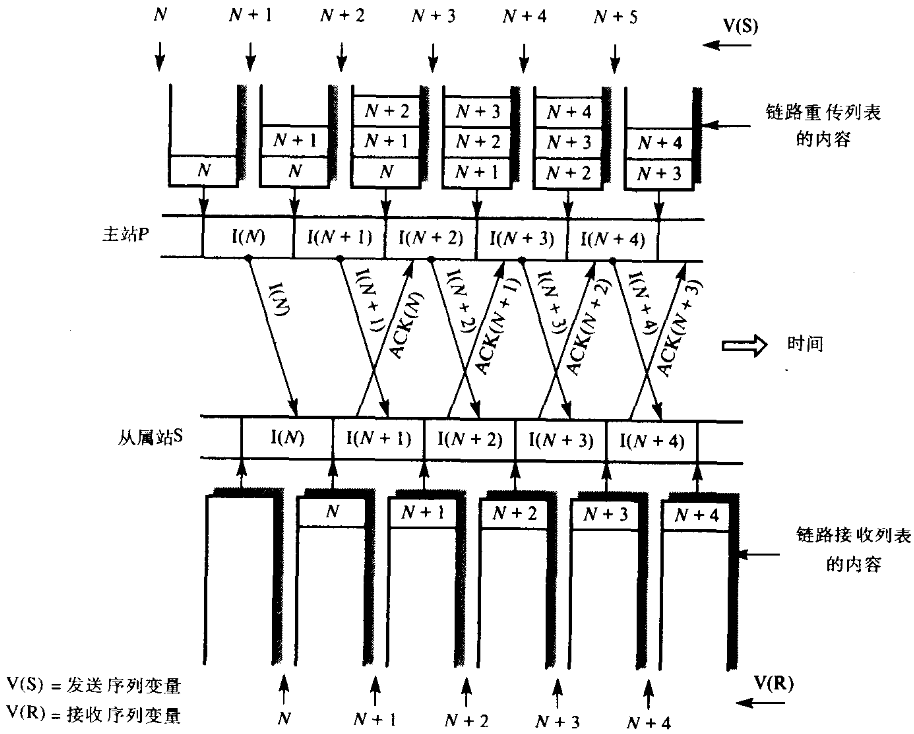


图4-11 连续RQ帧序列

我们在4.2.1节描述过，高层软件与通信协议软件之间的接口，一般采用两个FIFO队列。我们必须强调，这些与链路重传列表（在主站P）之间无任何联系，同样地，这些与链路接收列表（在从属站Q）之间也无任何联系。这两个列表由通信层内部使用，保证信息块在两个高层实体之间可靠传送。

为了实现这个方案，主站P必须保留发送序列变量V(S)，它指明下一个发送的I帧的发送序列号N(S)。同样的，从属站S必须保留一个接收序列V(R)，它指明正在等待下一个按序传送I帧。

由图4-11我们可以得出结论，在不考虑传输差错时，假定主站对I帧的发送不受任何限制，则连续RQ方案的链路利用率接近100%。我们将在4.3.3节看到这一假设通常并不需要，正常情况下在相应的ACK帧接收到之前，主站只能发送有限个数的I帧。对于连续RQ方案的链路利用率将在4.3.6节进一步讨论。

图4-11假定没有传输差错发生。当差错确实发生时，可用下列两种重传策略：

- 从属站S检测并请求只重发序列中那些被损坏的帧（选择重发）；
- 从属站S检测接收到的失序帧，请求主站重发上次正确接收以后所有未确认的I帧。因此，后退N帧确认I帧。

注意：对于所有连续RQ方案，损坏的帧都是丢弃的，而重发请求仅在下一个无差错帧接收到以后开始。

4.3.1 选择重发协议

如同空闲RQ差错控制方案一样，实现选择重发有两种方法：其一，从属站S确认正确接收到的帧与主站P从接收到的ACK帧的序列确定丢失的帧（隐式重发）；或者，从属站S对序列中丢失的帧返回一个否定确认（显式请求）。在这两个方案中，接收到失序帧的事件，从属站S在链路接收列表保留那些帧直到接收到下一次顺序到达的帧。 190

说明第一个方案的情况的两个帧序列图如图4-12所示。部分(a)假定正确地接收到所有ACK帧，而部分(b)说明一个损坏的ACK帧的影响。对于下面图4-12(a)序列，注意下列几点：

- 假定I帧*N*+1被损坏；
- 如同前面一样，从属站S对每个正确接收到I帧返回一个ACK帧；
- 从属站对I帧*N*，*N*+2，*N*+3，……返回一个ACK帧；
- 当接收到I帧（*N*+2）的ACK帧时，主站P检测出I帧*N*+1没有被确认；
- 允许多个I帧损坏的可能性，主站P检测到一个未确认帧时，主站P进入**重发状态**；
- 当处于重发状态，新帧的发送被禁止直到所有未确认帧都已被重发；
- 主站P从重传列表中将I帧（*N*+2）清除，并在发送I帧（*N*+5）之前重传I帧（*N*+1）；
- 接收到I帧（*N*+1），从属站S把链路接收列表中排队帧的内容按正确的顺序传递给LS_user。 191

图4-12(b)所示的帧序列涉及到所有I帧正确地接收但ACK帧（*N*）损坏。注意：

- 接收到ACK帧（*N*+1），主站P检测出在重传列表中有待确认的I帧（*N*），因此重发I帧（*N*）；
- 接收到重发的I帧（*N*），从属站S从接收序列变量确定它已被正确接收过，所以认定I帧（*N*）是一个重复；
- 从属站S丢弃该I帧但返回一个ACK帧给主站P，以保证I帧（*N*）从重传列表中清除。

上面方案的操作取决于ACK帧的接收，接下来的帧启动最早损坏帧的重发。另一种方法是用一个显式否定确认帧去请求一个特殊帧重发。否定确认帧称为**选择拒绝**。图4-13是表示这个方案操作的两个帧序列图。部分(a)所示的序列假设确认没有损坏，而部分(b)所示序列表示丢失确认的影响。注意：

- 一个ACK帧确认重发列表所有I帧，序号直到ACK帧的序号； 192
- 假定I帧（*N*+1）损坏；
- 从属站对I帧（*N*）返回一个ACK帧；
- 当从属站S接收到I帧（*N*+2），并检测出I帧（*N*+1）丢失，因而返回一个包含丢失的I帧（*N*+1）的标识符的NAK帧（*N*+1）；
- 当接收到NAK（*N*+1），主站P解释为从属站S还在等待I帧（*N*+1），因而重发I帧（*N*+1）；
- 当从属站S返回一个NAK帧后进入重发状态；
- 在重发状态时，禁止ACK帧返回；
- 接收到I帧（*N*+1），从属站S离开重发状态，假设返回ACK帧；
- ACK（*N*+4）确认直到包括帧（*N*+4）的所有帧；
- 每个NAK帧采用一个定时器，保证如果损坏（因此，帧（*N*+1）不能收到），它可以重发。

为了理解为什么同时仅有一个NAK帧不能得到解决，考察图4-13(b)所示的帧序列图。注意：

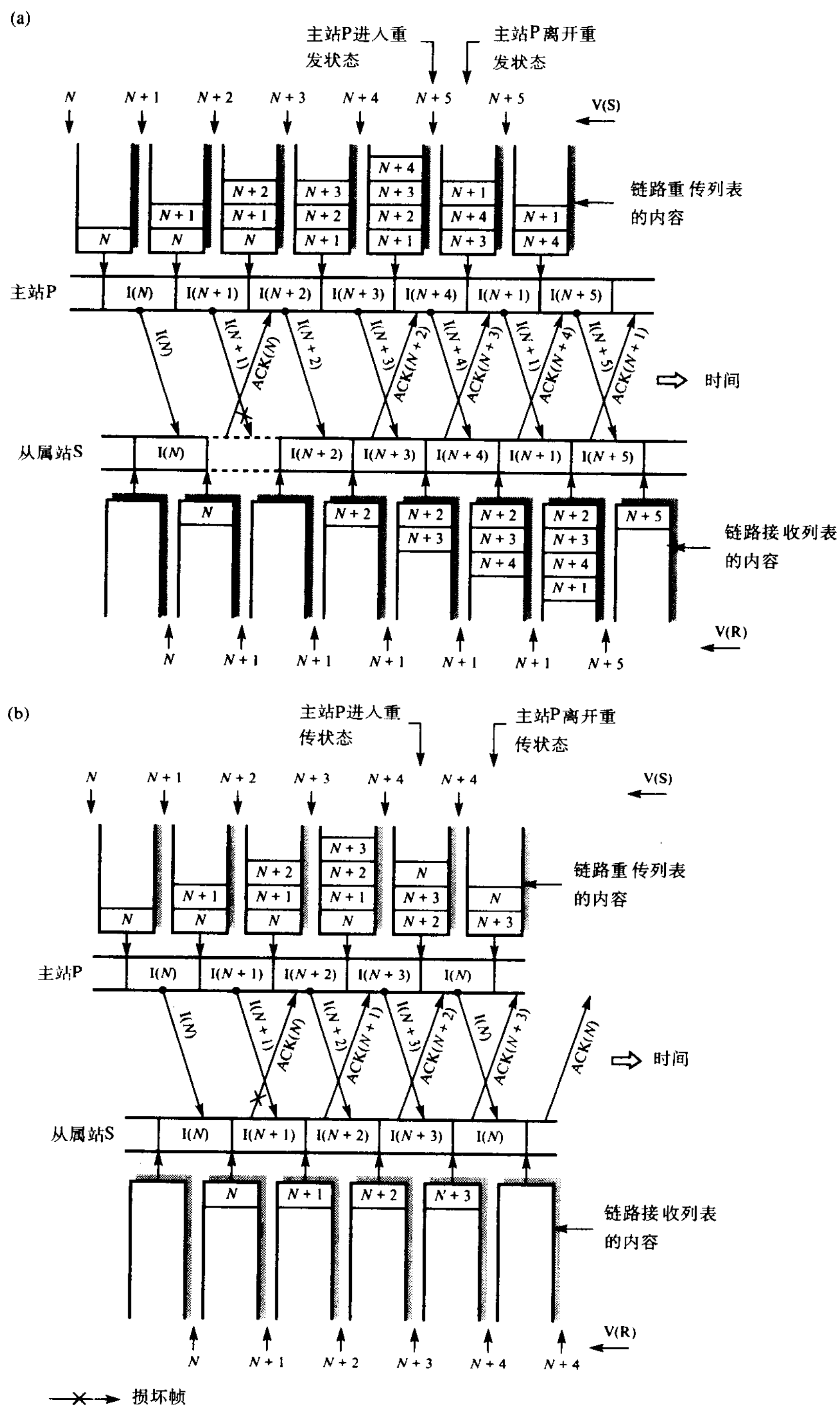


图4-12　选择重发——隐式重发
(a) I帧损坏　(b) ACK帧损坏

- 假定I帧（N+1）再次损坏；
- 如前面一样，从属站S返回NAK（N+1），但这次它也被损坏；

• 接收到ACK帧（$N+3$），这样确认所有帧，包括I帧（$N+1$），因此这个帧将不被重发；
• I帧（$N+1$）丢失。

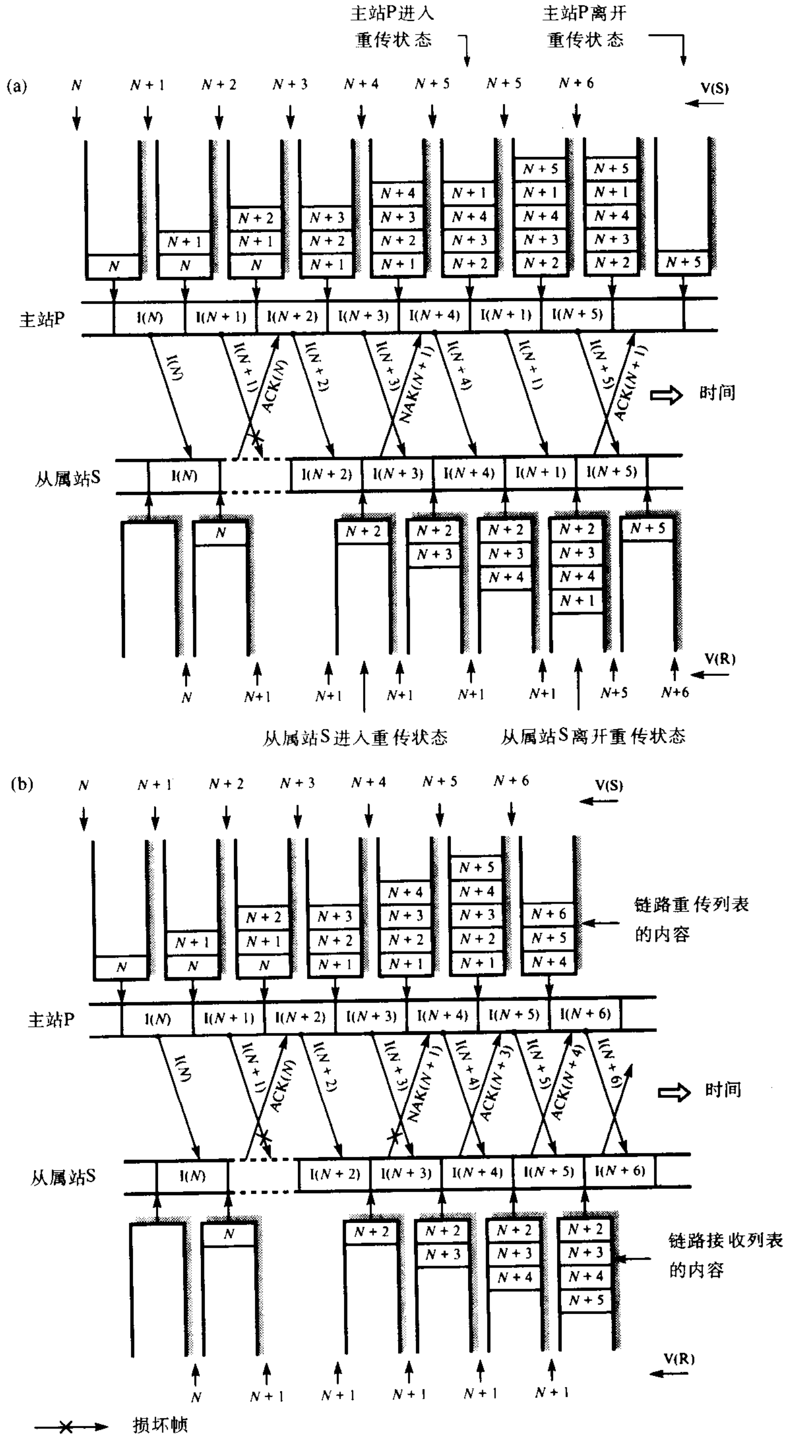


图4-13 选择重发——显式请求

(a) 正确操作 (b) 无重传状态的结果

我们能从所示的帧序列得出结论，虽然从属站S接收主站P发送的每一个正确帧，但不能保持接收的顺序。例如，从属站S在帧（N+1）之前接收帧（N+2）、（N+3）与（N+4）。当发送帧的内容是独立的，与其他帧内容无关（即，接收顺序并不重要），或同一信息的所有帧（或更大的帧）与信息（帧）将被从属站S重新装配时，采用选择重发方法。后一种实例情况是在一个高BER链路上（例如无线链路），适应链路的帧最大长度是短的，对与每个较大帧相关的小帧段采用选择重发的方法，在目的地传递帧之前，须对较小帧段重新按正确的次序装配，然后用NAK帧请求重发丢失的帧段。

然而，在许多通信中，发送帧的顺序必须与提交时相同。因此，接收到失序帧必须由从属站S相应的缓冲空间存储，直至收到失序帧。由于必须同时处理许多帧，存储量可能变得很
193~194 大，通信子系统需要的缓冲器存储量高得不能接受。为此，多数通信类型，如地面网络采用回退N帧重传控制方案。

4.3.2 回退N帧协议

回退N帧，正如它名字暗含，当从属站检测出一个失序帧，它通知主站从指定帧序列号开始重发帧。从属站通过返回一个称为**拒绝**的特殊否定确认帧来执行。说明回退N帧操作的帧序列的两个实例，如图4-14所示，解释图4-14的序列时，注意下列几点：

- 假定I帧（N+1）损坏；
- 从属站S接收到失序的I帧（N+2）；
- 当从属站S收到I帧（N+2），立即返回NAK（N+1）通知主站P返回，并从I帧（N+1）开始重发；
- 接收NAK（N+1）后，主站P进入重发状态；
- 处在重发状态，主站禁止发送新帧并开始重发在重发列表中等待确认的帧；
- 从属站S丢弃接收到I帧（N+1）前的所有帧；
- 接收I帧（N+1）后，从属站S恢复接受新帧并返回确认；
- 从属站S发送NAK帧时启动定时器，如果超时间隔到还没有接收到正确序号的I帧，则返回第二个NAK帧。

在图4-14(a)中，我们假设一个I帧损坏，并且确认帧正确收到。在帧传送序列中损坏确认帧的影响，如图4-14(b)所示，注意下面几点：

- 从属站S正确接收每个发送I帧；
- 假定两个ACK帧（N）与（N+1）都损坏；
- 接收到ACK帧（N+2），主站P检测出在重传列表中有两个未解决的I帧（N与N+1）；
- 由于这是一个ACK帧而不是NAK帧，所以主站P得出I帧（N）和I帧（N+1）的确认帧损坏，因而主站P接受ACK帧（N+2），作为两个未解决的I帧的确认。

图4-14(b)表示回退N帧的策略，能保持正确帧顺序，因而所需的缓冲器存储容量最小。但由于一些已正确接收到帧必须重发，因而有效传输容量的利用比选择重发方案效率低。因而在缓冲存储要求与链路利用之间存在折衷选择。

最后我们设定在图4-11到图4-13中涉及到连续RQ协议。I帧的丢失检测仅在下一个I帧正
195 确收到后，这要求I帧的连续流已准备发送，否则从属站S等待下一个发送I帧经历长时间的时延是不可接受的。

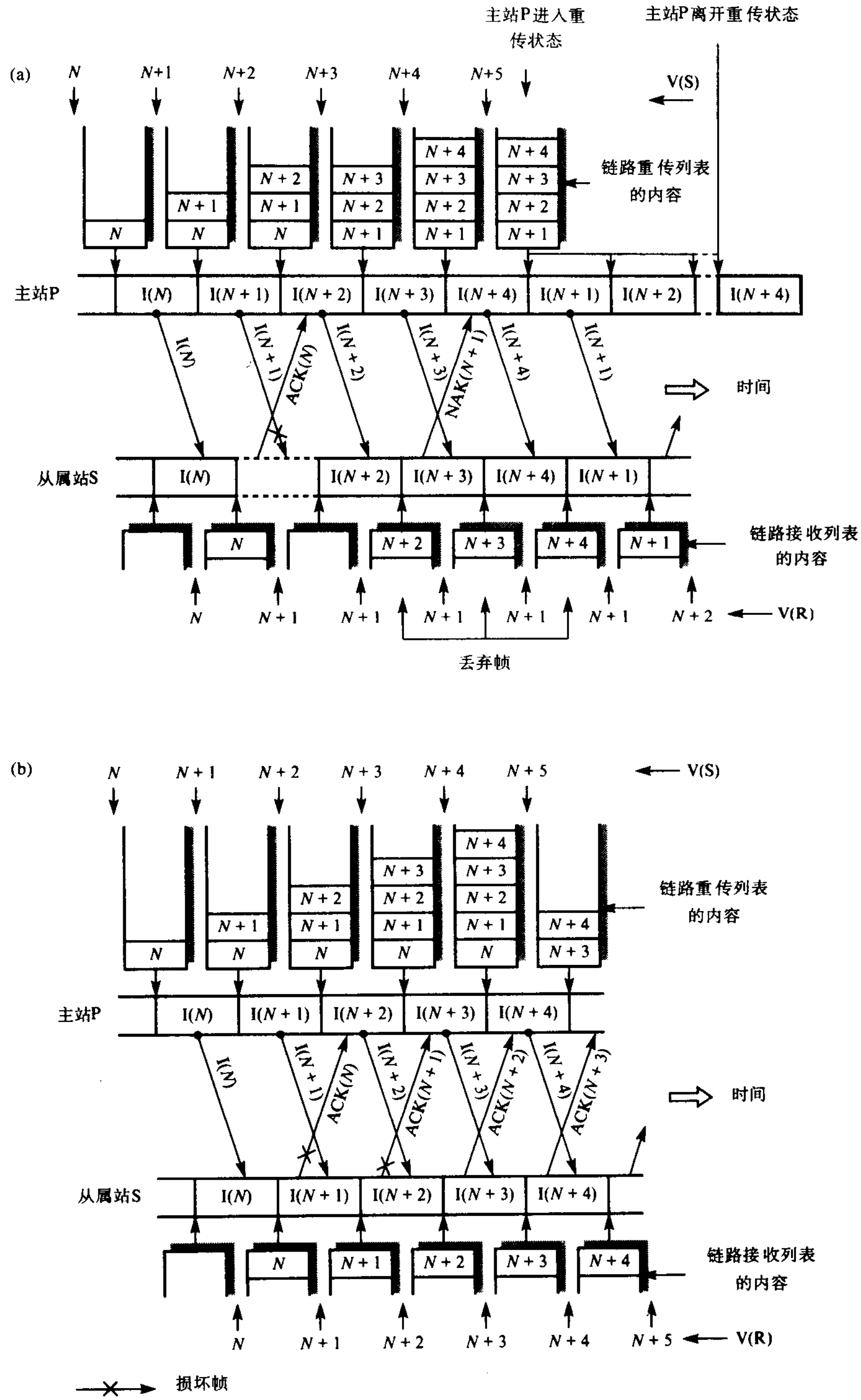


图4-14 回退N帧重传策略

(a) 损坏I帧 (b) 损坏ACK帧

考虑这种情况，类似空闲RQ控制方案的一个要点是主站P采用附加超时机制。有若干个

可用的方法，此处选用早期协议中的一个办法，每次主站P发送I帧，立即启动一个特定的定时器，当接收到一个说明正确收到的确认时，定时器复位。如果直到超时间隔到达，还没有接收到确认，则该帧重发，如图4-15所示。

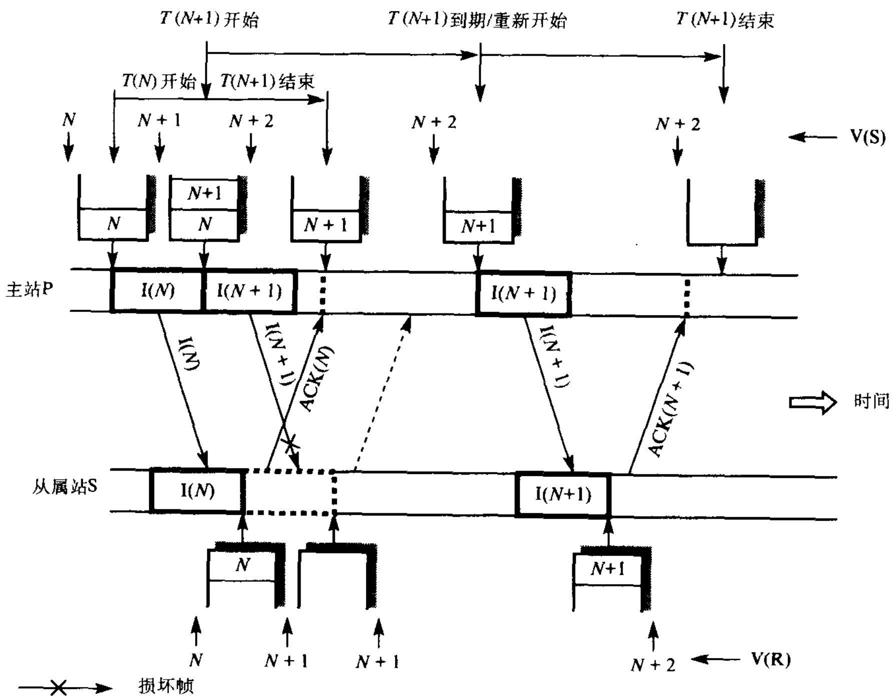


图4-15　超进机制

超时间隔的选择必须大于发送一个帧以及接收到确认之间最差情况下的传播延迟。关于超时机制，由于确认帧返回主站P过程中可能会损坏，从属站S有可能会收到正确帧的两份重复备份。这是由于确认帧不出现，主站P认为原来的帧受损，错误地重传另一备份。

回退N帧的控制方案，不会发生这个问题。这是因为在重发帧中N(S)不等于从属站S中的
196~197 现行V(R)值，从属站将会自动地丢弃。但在选择重发方案中，从属站S按序保存的最近正确接收到的N个I帧（接收列表）中，可能有一个或多个重复。回退N帧的方案，从属站S可以检测出接收到的帧是已正确收到帧的重复，还是新帧。我们将在4.3.4节中看到，由流量控制算法影响保存的帧的个数。

为了清楚起见，我们假定信息帧向一个方向发送，而确认简单地返回，但通常大多数通信链路使用连续RQ，是全双工的，即信息帧是双向发送的。为了适应这个情形，链路的每端都存在主站与从属站：前者控制发送I帧序列，后者控制接收I帧序列。因此，链路的每端有主站控制的V(S)与从属站控制的V(R)。虽然，可以独立地利用ACK帧与NAK帧，但由于当另一个方向的ACK帧或NAK帧返回时可能有I帧等待发送，所以有些协议利用返回方向的I帧携带向前方向I帧传送的确认，作为改善链路利用的方式。每个发送的I帧有N(S)，指明发送的序列号，还有N(R)，指明反方向的确认信息的序列号。这种方案称为**捎带确认**。高级数据链路控制协议（HDLC）使用这种技术（参见第5章）。

最后，尽管我们讨论过的所有ACK帧和NAK帧都携带N(R)，等同于它们带有N(S)。在实

际中，在从属站中V(R)接收到I帧立即加1，是在确认生成之前，这就是说大多数实际协议实现ACK N确认帧（N–1），HDLC也使用这种技术。

4.3.3 流量控制

差错控制仅是数据链路控制的一部分，另一个重要部分是**流量控制**。正如名字暗示，它涉及对链路上字符或帧传输速率的控制，以使接收器在处理信息前有足够的存储空间来接受每一个字符或帧。如对于面向字符的终端到计算机的链路，如果远程计算机为许多终端服务，则该计算机有可能出现暂时过载，因而不能处理按现用传输速率发送给它的全部字符。同样的，对于面向帧的选择重发方案，如果试图缓存不确定数量的帧时，接收器有可能超出缓冲存储容量。我们将讨论两种最常用的流量控制方案。

1. X-ON/X-OFF

由4.1节我们得知，回显检测除进行（手工）差错控制，也隐含着执行了流量控制。因为远程计算机用完了缓冲存储空间，它将停止向终端回显字符，因而用户自动停止键入。然而， 198
当计算机暂时过载时，一般不会产生回显字符，此时，如果用户继续键入新的字符，则计算机虽然只是简单读每个字符，然后将其清除，但还是承担不必要的处理开销。

因此，常使用自动流量控制装置，以保证计算机过载时，终端不再发送任何字符，一直到过载情况消失。这通过计算机完成。当计算机过载时，它返回一个特定控制字符X-OFF控制终端设备，令其停止发送。收到X-OFF字符，终端就不再理会键入的任何字符，或者将它们存入一个本地缓冲器中，直到计算机过载结束。此时，计算机可以不必承担任何不必要的处理开销。当过载条件消失，计算机可接受字符时，它返回一个控制字符X-ON通知终端的控制设备可以重新发送字符。这种机制也可用于计算机向打印机或其他终端（不能承受计算机输出速率）发送字符。在这种情况下，打印机（或终端）中的控制设备用以控制字符流量。

在2.6.1节所讨论的RTS/CTS握手控制规程EIA-232D/V.24接口执行相似的功能。为了区别两种方法，后者称为**带外信号传输流量控制**，而X-ON/X-OFF是**频带信号传输流量控制**的实例。

2. *滑动窗口*

为了控制经过数据链路的帧流量可用另一个称为**滑动窗口**的机制。回忆空闲RQ差错控制方案，虽然传输带宽的利用效率低，但要求最小缓冲存储容量。主站P发送帧必须等待到从属站S返回确认后，才能发送下一帧，所以通过链路的I帧流量是自动地严格控制。

然而，对于连续RQ差错控制方案，主站P在接收确认以前可以连续发送I帧，这种方案，如果接收器不能以接收帧的速率处理帧，则很可能用完所有可用缓冲容量，而暂时过载。因而，在方案中引入类似于空闲RQ控制方案的调整措施，其本质是在收到一个确认帧之前，对主站P可发送帧的数目加以限制。这由主站P监视保留在重传列表中待确认I帧的个数来实现。如果接收方来不及对收到帧进行处理，则从属站S停发返回确认帧，这时主站P构造重传列表，依次解释为P停止发送新帧的信号，直到再次收到确认帧。 199

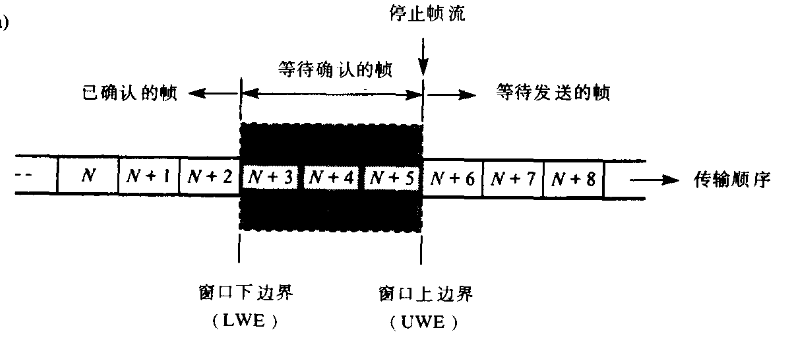
为了实现这个方案，重传列表中应设置等待确认I帧个数的最大限度。这个限度称为链路**发送窗口*K***。如果窗口为1，则传输控制方案就回到空闲RQ方案，此时传输效率下降。故该限度应选为使接收方能处理或接受所有收到的帧。当然选择发送窗口时必须考虑诸如帧的最大长度、可使用的缓冲存储容量、链路的传播时延与传输速率等因素。

这种方案的操作，如图4-16所示。每当发送一个I帧，**窗口上边界**（UWE）就增加1。同样，每当确认一个I帧，**窗口下边界**（LWE）减1。如果UWE 与LWE的差等于发送窗口K，停止接收任何新帧（信息块）。因而也停止I帧流的发送。假定传输无差错，K是滑动经过发送帧

全体的固定窗口，这种技术称为“滑动窗口”。

从属站S需要帧缓存的最大个数称为**接收窗口**。由前面空闲RQ方案与回退N帧方案的帧序列图，我们可得出接收窗口为1。然而，选择重发方案需要K个帧以保证按序传递给LS_user。

(a)



发送窗口，$K=3$

(b)

协议	发送窗口	接收窗口
空闲RQ	1	1
选择重发	K	K
回退N帧	K	1

图4-16　流量控制原理

(a) 滑动窗口实例　(b) 发送窗口与接收窗口限度

4.3.4 序列号

200 直到目前为止，我们假定由主站P插入到每个帧的序列号仅是前一个I帧的序列号加1，这样所用的序列号范围是无限的。定义通过链路可能传送I帧序列号的最大范围不仅限制重传列表与接收列表的大小，并且可能限制用于惟一识别每个传送帧所需的序列号的范围。标识符数量是重发控制方案和所用发送与接收窗口大小的函数。

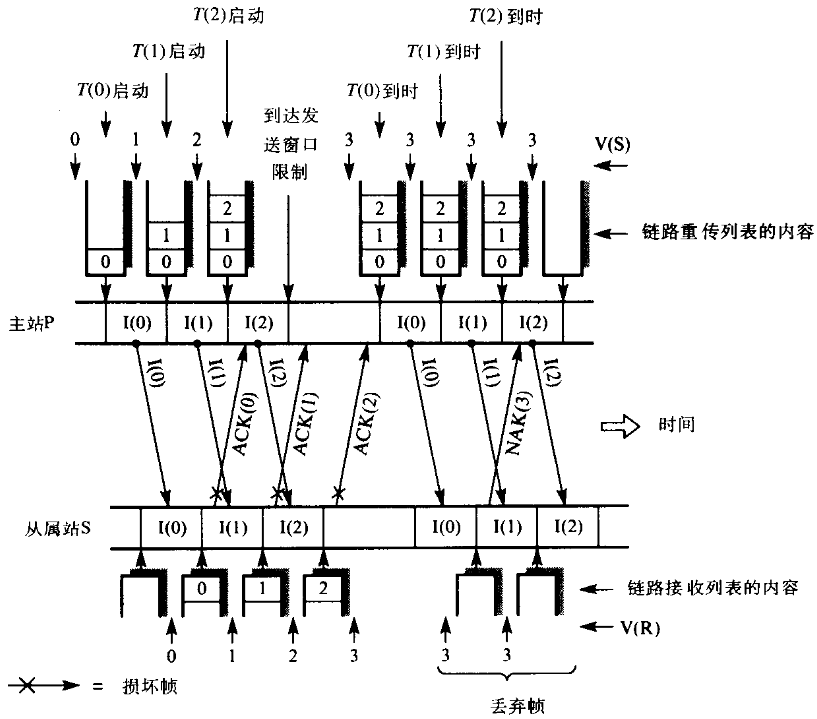
例如，对于空闲RQ控制方案，发送与接收窗口都是1，因而为使从属站S能确定收到的I帧是新帧还是一个重复，仅需要两个标识符，即二进制数0与1，而发送序列变量则应由主站P按模2增加。

采用回退N帧控制方案，发送窗口为K，则标识符的数目至少是$K+1$。从图4-17(a)所示可清楚地看出这一点，在说明该例时，应注意下列几点：

- 实例中发送窗口$K=3$。
- 主站P发送I帧数等于3（满窗口）。
- 从属站S正确收到3个I帧。
- 从属站S返回的3个ACK帧都损坏。
- 主站P对每个超时I帧都重发。
- 从属站S丢弃每一个重复的I帧，并返回一个NAK帧(3)，其中I(3)是期待的下一帧。
- 由于NAK帧中的序列号等于V(S)，主站P将其作为3个等待帧的确认，因而重新打开窗口。

如果只用3个标识符，即与发送窗口相同，则从属站S将不能确认I帧(0)是一个新帧，还是一个重复，因为0也可能是下一个按序标识符。故采用4个标识符（发送窗口加1），从属站S就能知道下一个按序I帧的标识符为3，而重发（复）帧的标识符为0。所以，将标识符序列号为0的帧丢弃。

(a)



(b)

协议	帧标识符的最大个数
空闲RQ	2
选择重发	2K
回退N帧	$K+1$

(c)

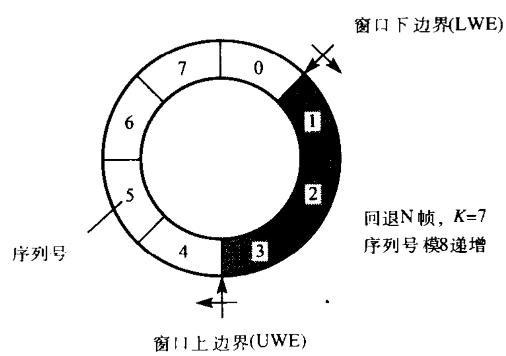


图4-17　序列号

(a) 最大范围的实例　(b) 每个协议的最大序列号　(c) 假定8个序列号的实例

对于选择重发方案，设其发送窗口与接收窗口都是K，则标识符的个数不能小于$2K$。同样可以设想主站P发送的I帧等于窗口K，并且所有相应的确认都损坏。从属站必须确认后面的K个帧是新帧还是重复。惟一能保证这一点的方法是发送I帧的下一个窗口必须分配K个新标识符，因而至少必须有$2K$个标识符。每种方案帧标识的限制如图4-17(b)所示。

实际上，由于一个帧的标识符是以二进制形式给出，所用的二进制数字必须预先给定。
201 例如对于回退N帧控制方案，发送窗口为7，则需要3位二进制数才能产生0～7的8个发送与接收序列号，发送与接收序列变量应由主站P与从属站S分别按模8增加。如图4-17(c)所示。

4.3.5　协议规范说明

图4-11到图4-18的帧序列图与有关的文字提供连续RQ协议的差错控制与流量控制定量的描述。现在我们基于4.2.2节的方法给出更形式化规范说明。为了简化描述，我们仅考察I帧单方向流动，即从源DTE到目标DTE。然而大多数应用，每一端都有主站与从属站，需要双向I帧的流动。

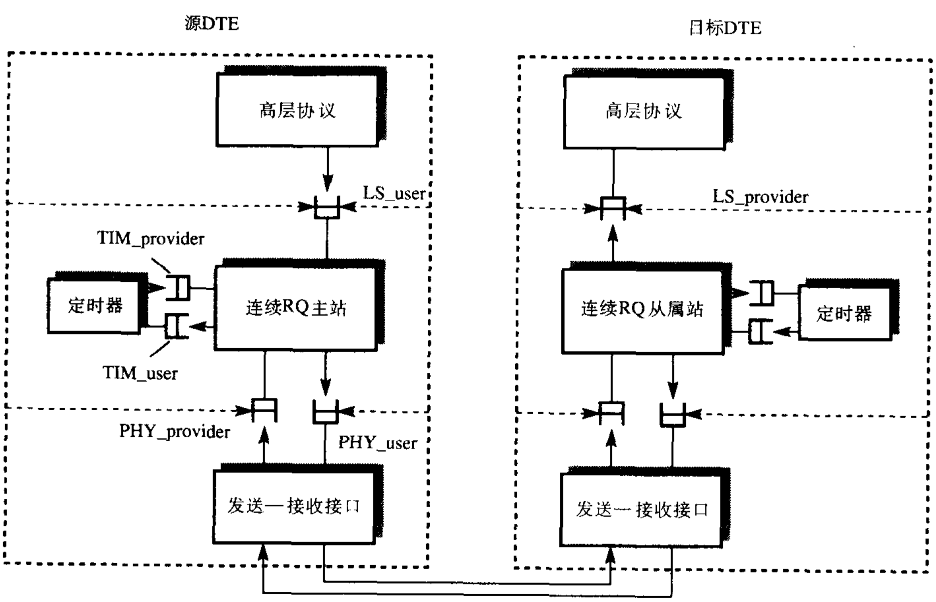


图4-18　连续RQ协议层接口

首先，我们在图4-18中标识每个协议实体的各种接口，回退N帧差错控制策略与窗口流量控制机制作为图4-19与图4-20中两个规范说明的基础。其中图4-19描述主站，图4-20描述从属站。

在两个规范说明中，所用的简化名称在图(a)部分定义，每个协议的规范说明（主站与
202～203 从属站）所用的三种方法在图(b)、(c)与(d)中表示。正如我们从每个规范说明所见，尽管状态迁移图给出每个协议基本操作的概述，操作的所有可能方面却不能用这个方法详细说明。

扩充事件—状态表由于引入谓词规范允许加入更多细节。用结构化代码（伪码）给出的每个协议规范说明可以系统方式执行。在图4-19与图4-20的部分(d)中，给出伪码规范说明。在以后几章，对各种协议求它们的扩充事件—状态表或基本特性，可用同一方法求相应的结
204 构化代码。

(a)

入事件

缩写名	接 口	含 义
LDATAreq	LS_user	接收L_DATA .request服务原语
ACKRCVD	PHY_provider	接收来自从属站S的ACK帧
TEXP	TIM_provider	等待ACK帧到时的定时器
NAKRCVD	PHY_provider	接收来自从属站S的NAK帧

状态

缩写名	含 义
DATA	数据传送

出事件

缩写名	接 口	含 义
TxFrame(X)	PHY_user	用N(S)=X格式化一个帧，并传送到PHY_user队列尾部
RetxFrame(X)	PHY_user	传送帧（X）到PHY_user队列尾部

谓词

缩写名	含 义
P0	打开发送窗口

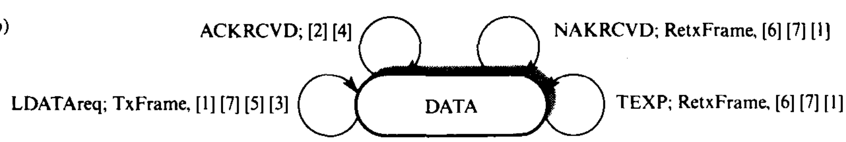
特定动作

[1]= 用TIM_user队列启动定时器（X）
[2]= 用TIM_user队列停止定时器（X）
[3]= Va增1
[4]= Va减1
[5]= Vs增1
[6]= 由重传列表头部获得帧
[7]= 置帧于重传列表的尾部

状态变量

Vs =发送序列变量
Present State= 协议实体现在状态
Va = 重传列表中未确认帧的个数

(b)



(c)

当前状态 \ 入事件	LDATAreq	ACKRCVD	TEXP	NAKRCVD
DATA	1	2	3	4

1 = P0: TxFrame, [1] [7] [5] [3]
2 = [2] [4]
3 = RetxFrame, [6] [7] [1]
4 = RetxFrame, [6] [7] [1]

(d)

```
program  Continuous RQ_Primary;
const    K; { Send Window Limit }
type     Events = (LDATAreq, ACKRCVD, TEXP, NAKRCVD);
         States = DATA;
var      EventStateTable = array[Events, States] of 1..4;
         EventType : Events;
         PresentState : States;
         Vs = 0..K;
         Va = 0..K - 1;
         RetxList; { FIFO queue holding I-frame buffer pointers awaiting
                acknowledgment }
```

图4-19 连续RQ（回退N帧）协议的规范说明

(a) 缩写名 (b) 状态迁移图 (c) 扩充事件—状态表 (d) 结构化伪码

```
procedure Initialize; { Initializes contents of EventStateTable and state variables }
procedure TxFrame (X : integer);     } Outgoing event procedures
procedure RetxFrame (X : integer);   }
procedure Start_timer (X : integer); }
procedure Stop_timer (X : integer);  } Specific action procedures
procedure Get_frame;                 }
procedure Put_frame;                 }
function  P0 : boolean;              } Predicate function

begin   Initialize;
        repeat Wait for an incoming event;
               EventType := type of incoming event;
               case EventStateTable[EventType, PresentState] of
                 1 : begin if P0 then begin TxFrame(Vs); Start_timer(Vs); Put_frame;
                                            Vs := Vs + 1; Va := Va + 1;
                              if Va = K then P0 := false end;
                     end;
                 2 : repeat; X := N(R) from frame; Stop_timer(X);
                                  Va := Va - 1; P0 := true;
                     until  X = N(R) in ACK-frame;
                 3 : begin  Get_frame; X := N(S) from frame; RetxFrame(X);
                                  Start_timer(X); Put_frame; end;
                 4 : repeat Get_frame; X := N(S) from frame; RetxFrame(X);
                                  Start_timer(X); Put_frame;
                     until  X = N(R) in NAK-frame;
        until Forever;
end.
```

图4-19　（续）

(a)

入事件		
缩写名	接口	含义
IRCVD	PHY_provider	接收来自主站P的I帧

状态	
缩写名	含义
DATA	等待N(S)=Vr的下一个I帧
NAK_SENT	等待N(S)=Vr的丢失的I帧

出事件		
缩写名	接口	含义
LDATAind(X)	LS_provider	传送N(S)=X的I帧到上一层
TxACK(X)	PHY_user	格式化并发送N(R)=X的ACK帧
TxNAK(X)	PHY_user	格式化并发送N(R)=X的NAK帧
LERRORind	LS_provider	给出一个差错消息

谓词	
缩写名	含义
P0	接收到I帧中N(S)=Vr

特定动作

[1]= 启动定时器
[2]= 停止定时器
[3]= Vr增1
[4]= RetxCount增1
[5]= ErrorCount增1
[6]= 丢弃帧

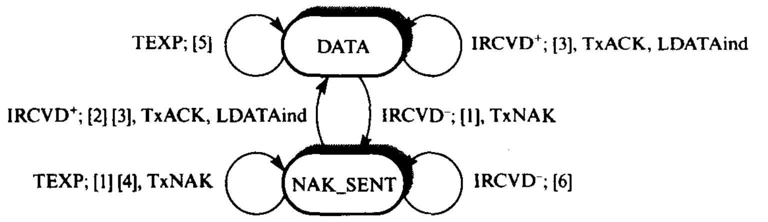
状态变量

Vr = 接收序列变量
RetxCount = 重传NAK帧的次数
ErrorCount = 差错的次数
PresentState=协议实体当前状态

图4-20　连续RQ（回退N帧）规范说明——从属站

(a) 缩写名　(b) 状态迁移图　(c) 扩充事件—状态表　(d) 结构化伪码

(b)



(c)

入事件 / 当前状态	IRCVD	TEXP
DATA	1	0
NAK_SENT	2	3

0 = [5]
1 = P0: TxACK, LDATAind, [3]
  NOT P0: TxNAK, [1], NAK_SENT
2 = P0: TxACK, LDATAind, [2] [3], DATA
  NOT P0: [6]
3 = TxNAK, [1] [4]

(d)

```
program    Continuous RQ_Secondary;
const      MaxRetxCount;
           MaxErrorCount;
type       Events = (IRCVD, TEXP);
           States = (DATA, NAK_SENT);
var        EventStateTable = array [Events, States] of 0..3;
           EventType : Events;
           PresentState : States;
           Vr, RetxCount, ErrorCount : integer;
procedure  Initialize; { Initializes contents of EventStateTable and state variables }
procedure  LDATAind(X);  }
procedure  TxACK(X);     } Outgoing event procedures
procedure  TxNAK(X);     }
procedure  LERRORind;    }
procedure  Start_timer;  }
procedure  Stop_timer;   } Specific action procedures
procedure  DiscardFrame; }
function   P0 : boolean;   } Boolean function

begin      Initialize;
           repeat Wait for an incoming event; EventType = type of event;
                  case EventStateTable[EventType, PresentState] of
                     0 : begin ErrorCount := ErrorCount – 1; if (ErrorCount = MaxErrorCount) then
                                  begin LERRORind; Initialize; end;
                         end;
                     1 : X := N(S) from I-frame received;
                         begin if P0 then begin TxACK(X); LDATAind(X); Vr := Vr + 1; end;
                                  else begin TxNAK(X); Start_timer; PresentState := NAK_SENT; end;
                         end;
                     2 : X := N(S) from I-frame received;
                         begin if P0 then begin TxACK(X); LDATAind(X); Vr := Vr + 1; Stop_timer; end;
                                  else DiscardFrame;
                         end;
                     3 : begin TxNAK(Vr); Start_timer; RetxCount := RetxCount + 1;
                               if (RetxCount = MaxRetxCount) then begin LERRORind; Initialize; end;
                         end;
           until Forever;
end.
```

图4-20 （续）

4.3.6 链路利用

在4.2.4节讨论空闲RQ协议时，我们给出链路利用率U的近似结果。U是I帧传输时间T_{ix}与链路传播时延T_p的函数。然而，对于T_p大于T_{ix}的链路，发送窗口K也将影响链路利用率。

在图4-10所示的实例中，典型的卫星链路T_p远大于T_{ix}。对于1 Mbps信道与长度为1000位的帧，T_{ix}是1 ms，而T_p为250 ms。而且在理论上，主站与从属站之间任何时刻可能有250个这样的帧被传输。如果要实现100%的链路利用率，则发送窗口需要超过500，由于序列中第一个帧的确认帧将在500 ms（$2T_p$）后才收到。

205~207 通常U（链路利用率）与K（发送窗口）的关系由下式给出，当K大于等于$1+2a$时，U=1；当K小于$1+2a$时，

$$U=\frac{KT_{\text{ix}}}{T_{\text{ix}}+2T_{\text{p}}}=\frac{K}{1+\dfrac{2T_{\text{p}}}{T_{\text{ix}}}}=\frac{K}{1+2a}$$

设$T_p = T_{ix}$，即a=1，很容易说明上式。此时，由主站P发送帧的最后一位经$2T_p$（即$2T_{ix}$）后才被从属站S收到，相应的ACK帧再经过T_p（即T_{ix}）才被主站P接收。如果K=1（空闲RQ），则U =1/3，为使其达到100%（K大于$1+2a$），K必须增加到3，即当ACK帧被接收前，主站P必须发送3个或更多的帧。

实例4-2

采用连续RQ协议发送1000位的帧。假设传播速率为2×10^8 ms^{-1}，忽视链路位误码率的影响，求下列几种数据链路的利用率：

（a）1 Mbps的链路长为1 Km及与发送窗口K=2；

（b）200 Mbps的链路长为10 Km及与发送窗口K=7；

（c）2 Mbps的卫星链路长为50 000 Km及与发送窗口K=127。

解：因为 $T_p=\dfrac{S}{V}$，$T_{ix}=\dfrac{N_i}{R}$，$a=\dfrac{T_p}{T_{ix}}$，N_i = 信息位的个数，则

（a）
$$T_p=\frac{10^3}{2\times10^8}=5\times10^{-6}s$$
$$T_{ix}=\frac{1000}{1\times10^6}=10^{-3}s$$

因此

$$a=\frac{5\times10^{-6}}{1\times10^{-3}}=5\times10^{-3}$$

由于K=2 大于$1+2a$，所以U=1

（b）
$$T_p=\frac{10\times10^3}{2\times10^8}=5\times10^{-5}s$$
$$T_{ix}=\frac{1000}{200\times10^6}=5\times10^{-6}s$$

因此

208
$$a=\frac{5\times10^{-5}}{5\times10^{-6}}=10$$

由于K=7 小于1+2a，所以$U=\dfrac{K}{1+2a}=\dfrac{7}{1+20}=0.33$

（c）$T_p=\dfrac{50\times10^6}{2\times10^8}=0.25\ s$

$T_{ix}=\dfrac{1000}{2\times10^6}=5\times10^{-4}\ s$

因此 $a=\dfrac{0.25}{5\times10^{-4}}=500$

由于K=127 小于1+2a，所以 $U=\dfrac{K}{1+2a}=\dfrac{127}{1+1000}=0.127$

从这些结果，我们可推出K的选取对某些情形的链路利用有很大的影响。正如我们看到，即使K为127，卫星链路的利用率还是非常低。因此，需采用一个较大的K值（因此序列号范围也较大）。这一点也适用于高速率的地面链路，我们将在第5章讨论。

此例是在假设没有传输差错的情况下得到的计算结果。如果将差错考虑在内，则有些帧必须重发，这样势必降低链路利用率。但其影响对两种不同的重发方案是不一样的。例如，选择重发方案，U的减少仅取决于发送每一帧时重传尝试的次数N_r，因为只有损坏的帧才重传。假定差错是随机的，P_f是链路的帧差错率，则

$$N_{\mathrm{r}}=\frac{1}{1-P_{\mathrm{f}}}$$

当K小于1+2a时，U值修正为：

$$U=\frac{K}{N_{\mathrm{r}}(1+2a)}=\frac{K(1-P_{\mathrm{f}})}{1+2a}$$

当K大于或等于1+2a时，我们发现在表示式中U简单用K=1+2a代替，因为1+2a是在时间$T_{\mathrm{ix}}+2T_{\mathrm{p}}$中，接收到确认帧前帧重发的最大次数。

因此

$$U=\frac{(1+2a)(1-P_{\mathrm{f}})}{1+2a}=1-P_{\mathrm{f}}$$ 209

回退N帧方案，链路利用率则进一步降低，因为如果一个帧损坏，则必须重传多个帧，且重传的帧数取决于K与1+2a的关系。

当K小于1+2a时，（K–1）个帧必须重传的次数等于P_f（K–1），而每重传一次，将有1+2a的时延，因而U表达式修正为：

$$U=\frac{K(1-P_{\mathrm{f}})}{(1+2a)+(1+2a)P_{\mathrm{f}}(K-1)}=\frac{K(1-P_{\mathrm{f}})}{(1+2a)(1+P_{\mathrm{f}}(K-1))}$$

当K大于或等于1+2a时，

$$U=\frac{(1+2a)(1-P_{\mathrm{f}})}{(1+2a)(1+P_{\mathrm{f}}(K-1))}=\frac{1-P_{\mathrm{f}}}{1+P_{\mathrm{f}}(K-1)}$$

应该注意：这些公式只是近似值，而且没有考虑重传帧发生差错所带来的影响。但是给出每种方法的有关性能与期待性能的指导说明。

实例4-3

在100 Km的数据链路上以20 Mbps的速率串行发送1000位的帧。如果链路的传播速率为2×10^8 ms^{-1}，差错率为4×10^{-5}，使用下列链路协议确定链路利用率：

(a) 空闲RQ

(b) 选择重发与发送窗口为10

(c) 回退N帧与发送窗口为10

解：

因为

$$T_p = \frac{S}{V} = \frac{100\times10^3}{2\times10^8} = 5\times10^{-4}s$$

$$T_{ix} = \frac{N_i}{R} = \frac{1000}{20\times10^6} = 5\times10^{-5}s$$

$$a = \frac{T_p}{T_{ix}} = \frac{5\times10^{-4}}{5\times10^{-5}} = 10$$

所以　$1+2a=21$

210 由于　$P_f \simeq N_iP = 1000 \times 4 \times 10^{-5} = 4 \times 10^{-2}$

因此　$1-P_f = 96 \times 10^{-2}$

(a)

$$U = \frac{(1-P_f)}{1+2a} = \frac{96\times10^{-2}}{21} = 0.046$$

(b) 由于K小于$1+2a$

$$U = \frac{K(1-P_f)}{1+2a} = \frac{10\times96\times10^{-2}}{21} = 0.46$$

(c) 由于K小于$1+2a$

$$U = \frac{K(1-P_f)}{(1+2a)(1+P_f(K-1))} = 0.336$$

4.4　链路管理

差错控制与流量控制两者关系到经过一个不完善通信线路帧的正确传送（按序与无差错或无重复）。对于各种方案，我们已正确地概述功能，假定通信双方已初始化，交换信息已准备就绪。例如，链路两端在信息帧发送前，必须有相同的发送与接收序列变量。通常这称为初始化或**链路建立**阶段。类似地，经过链路交换数据完成后，有一个**链路拆接阶段**。因为链路建立与拆接阶段不涉及到真实的用户数据的传送，它们称为**链路管理**。

一个终端与一台计算机之间的线路距离靠近（如最多20 m），管理功能可在物理接口上附加握手（控制）线通过交换信号完成，这称为**握手过程**。用户首先接通终端，开始与计算机对话，这时激活控制线之一，向计算机指示终端准备发送字符（数据终端准备就绪）。然后，终端等待直到相应的响应控制线指示计算机准备接收字符，最后交换字符可以开始。

当双方通信设备都是计算机，它们经过数据链路传送，则链路的建立是在每台计算机中

的链路层协议交换一组规定的**控制**或**监控帧**实现。在我们终端到计算机的实例中，当用户接通终端时，链路就建立。但在计算机到计算机链路的情况下，链路建立通常是由一台计算机的高层软件（例如，应用软件）发信号通知通信软件发起的，它要与远程计算机对话。典型地，这可以是一个发送执行或者应用程序中的请求语句（原语），反过来应用程序引起通信软件执行。实际上，我们将在本书的第三部分看到，通信软件通常由一些独立的协议层组成，每一层负责全部通信任务的一个特殊功能。在发送数据前，每一层首先初始化。作为一个实例，数据链路层的初始化原语，如图4-21(a)所示。 211

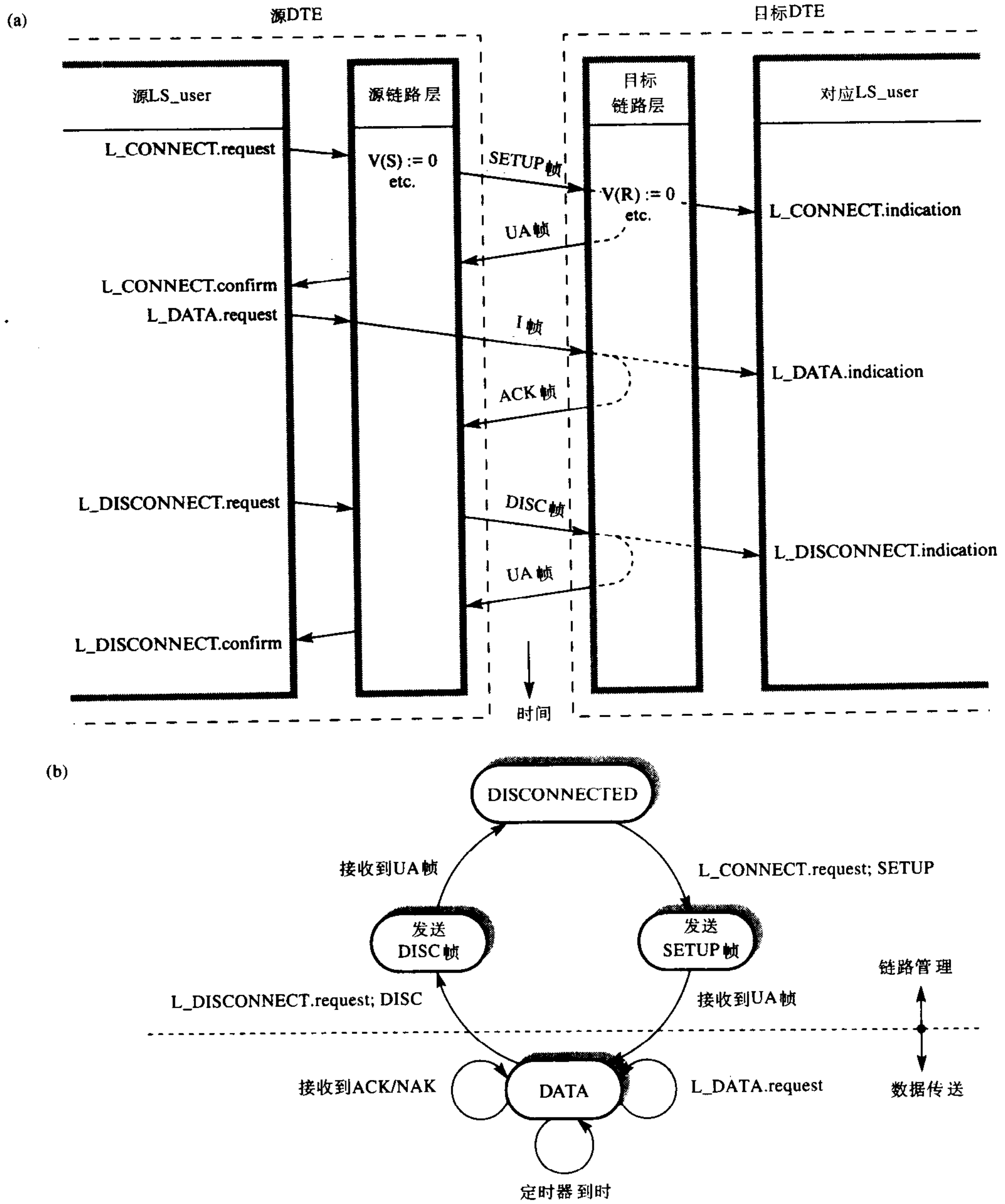


图4-21 链路管理

(a) 时序图 (b) 状态迁移图

正如我们所见，在发送数据前（用L_DATA.request服务），最初LS_user（层）发送L_CONNECT.request（连接请求）服务原语到链路层。和L_DATA服务不同，这称为**证实服务**。当源链路协议实体与目标（链路）协议实体建立一条链路（逻辑连接）时，它向源
212 LS_user返回L_CONNECT.confirm原语。然而，注意这证实仅与目标链路协议实体建立一条逻辑链路，而不是与目标LS_user建立。我们将在本书的第三部分看到，后者通常是针对所有更高协议层的情况。

一接收到事件控制块（ECB）形式的L_CONNECT.request原语（参见4.2.1节），源链路实体初始化所有状态变量，然后产生一个**链路SETUP帧**（PDU）。以选择重发方式发送到对应（对等）的目标DTE的链路协议实体。一接收到SETUP帧，目标DTE初始化它自己的状态变量，继续向对应的LS_user发送一个L_CONNECT.indication原语并向源LS_user返回一个确认帧。

由于这个确认帧与I帧无关，它不含有序列号，因而称为**未编号确认帧**或**UA帧**。接收到这个UA帧，源协议实体向LS_user发出一个L_CONNECT.confirm原语及目前链路为用L_DATA服务传送数据已准备就绪。最后，当所有数据传送完后，用L_DISCONNECT服务释放建立的链路，这个服务也是一个认证服务。对应的帧称为**拆接帧**或**DISC帧**，也用UA帧确认。这种操作模式也称为面向连接的模式。

显然，增加链路管理功能涉及到前面讨论的空闲RQ与连续RQ协议规范说明。为了清楚说明这一点，对主站的状态迁移图作修改，如图4-21(b)所示。

正如我们所见，需要三个新状态，入事件使这些新状态之间发生迁移，数据传送状态如图所示。对于从属站状态迁移图同样容易扩充。再者，扩充事件—状态表与对应伪码两者的
213 结构是把增加的部分简单插入，不作任何主要结构的改变。

习题

4.1 假设一个终端与计算机相连，解释用于实现差错控制和流量控制的两种技术，概括每种方案对终端用户的影响。

4.2 解释下列与数据链路控制协议相关的术语的意义：
(a) 无连接
(b) 面向连接

4.3 假定空闲RQ差错控制方法，借助帧时序图，描述隐式重传与显式重传控制方法之间的区别。

4.4 假定采用显式重传的空闲RQ差错控制方法，借助帧时序图，描述下列各项：
(a) 影响两个连续信息帧之间最短传输时延的因素
(b) 如何解决损坏信息帧的丢失问题
(c) 如何解决损坏确认帧的丢失问题

4.5 说明下列与协议相关术语的含义
(a) 协议层
(b) 用户服务
(c) 时序图
(d) 层间的队列

4.6 解释下列与协议规范说明相关的术语的含义

(a) 有限状态机或自动机

(b) 协议状态

(c) 入事件

(d) 出事件

(e) 谓词

(f) 本地或特定动作

4.7 使用习题4.4中得到的时序图定义在采用空闲RQ差错控制方案时，链路两端主站和从属站的操作：

(a) 状态迁移图

(b) 扩充事件—状态表

(c) 以伪码高级语言编写的结构化程序代码

4.8 使用空闲RQ协议经过下列数据链路发送一串平均长度为100位的信息帧。如果链路的传播速度为$2\times10^8\ ms^{-1}$，对每种类型的数据链路，确定链路的效率(利用率)：

(a) 10 Km链路,BER为10^{-4}，数据传输速率为9600 bps

(b) 500 Km链路,BER为10^{-6}，数据传输速率为10 Mbps

4.9 借助帧时序图,描述空闲RQ与连续RQ差错控制方法之间的不同。为简单起见，假定传输过程中无损坏帧。

4.10 假定采用选择重发差错控制方案，借助于帧时序图，使用隐式与显式重发,描述如何解决下列问题：

(a) 损坏信息帧

(b) 损坏ACK/NAK帧

4.11 用习题4.10得到的时序图定义在采用连续RQ与选择重发差错控制方案时，链路两端主站和从属站的操作：

(a) 状态迁移图

(b) 事件—状态表

(c) 以伪高级代码编写结构化程序代码

推出关于选择重发策略影响链路接收列表的最大长度的因素。

4.12 假定采用回退N帧差错控制方案，借助于帧序列图，描述如何解决下列问题：

(a) 损坏I帧

(b) 损坏ACK帧

(c) 损坏NAK帧 214

在这些图中包括链路重传列表与链路接收列表的内容，还包括每个发送帧与接收帧的发送与接收序列变量的状态。

4.13 利用连续RQ差错控制方案作为实例，描述链路一端的主站与从属站的操作如何以有限状态机形式定义：

(a) 状态迁移图

(b) 状态迁移表

(c) 以伪码编写高级语言程序段

4.14 超时机制的功能是什么？利用帧时序图说明超时间隔如何用于克服损坏信息帧的影响。

假定：

(a) 空闲RQ

(b) 选择重发

(c) 回退N帧

推出每种方案使用的确定超时间隔的因素以及如何检测重复。

4.15 区分链路的发送窗口与接收窗口，并说明它们与下述方案的联系：

(a) 选择重传方案

(b) 回退N帧方案

4.16 借助于帧时序图，说明当发送窗口达到流量控制极限时的影响。假定发送窗口为2及使用回退N帧差错控制方法。

4.17 假定发送窗口为K，推出下列差错控制方案要求最小序号(帧标识符)数目:

(a) 空闲RQ

(b) 选择重发

(c) 回退N帧

明确采用最大标识符数目的条件。

4.18 经过数据链路4000 Km，数据速率2 Mbps，发送一串平均长度1000位的信息帧，如果链路传播速度为2×10^8 ms^{-1}，BER为10^{-4}，确定下列协议的链路利用率:

(a) 空闲RQ

(b) 选择重发与发送窗口为7

(c) 回退N帧与发送窗口为127

指明对所得到每个结果误码率的影响。

4.19 说明术语“链路管理”的含义是什么？利用一组用户原语的实例与时序图表示逻辑通信通路在两个系统之间如何建立以及今后如何拆接的。

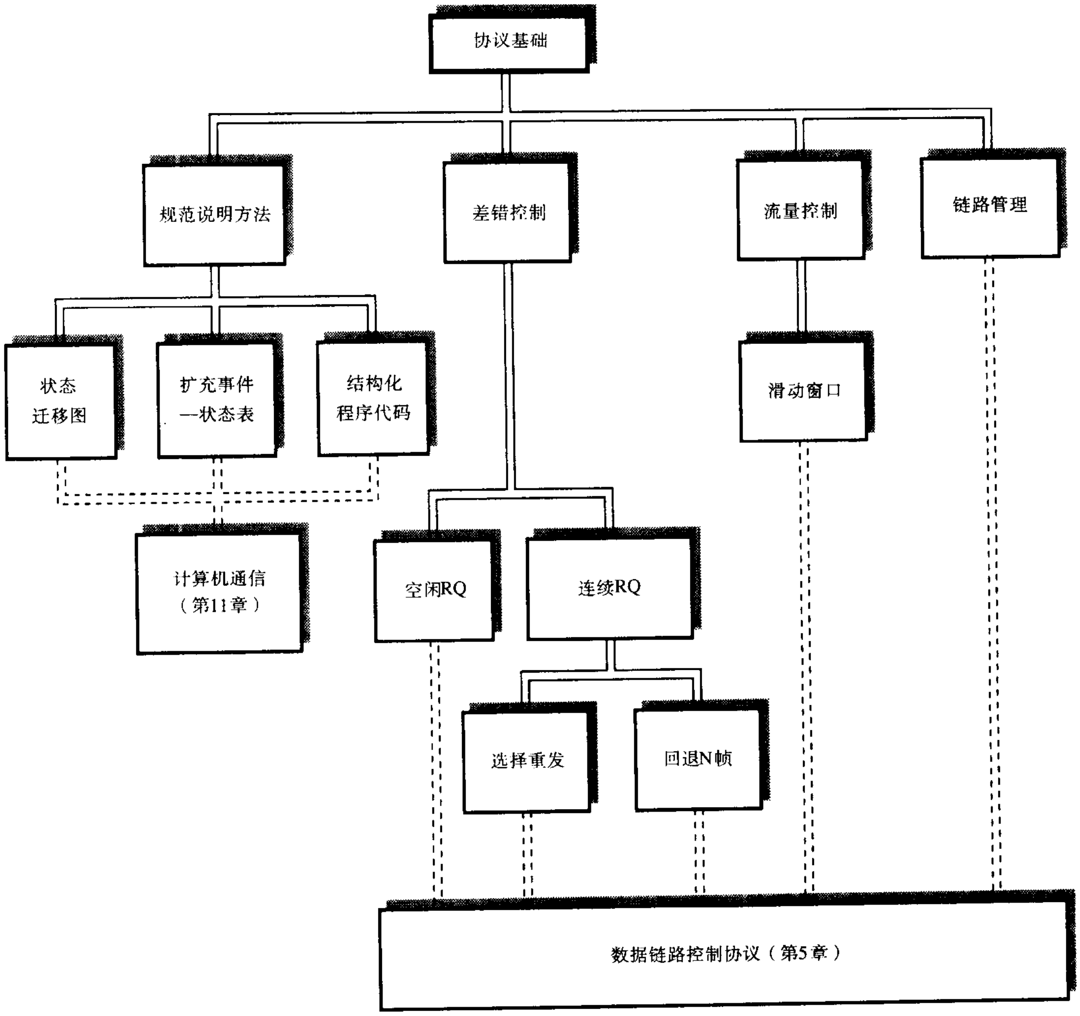
4.20 假设使用习题4.19所用的用户原语与空闲RQ差错和流量控制方案，推出:

(a) 状态迁移图

(a) 扩充事件—状态表

215 对于该协议，概述协议如何以伪码高级代码形式定义。

本章概要


协议基础
规范说明方法
差错控制
流量控制
链路管理
状态
迁移图
扩充事件
—状态表
结构化
程序代码
滑动窗口
计算机通信
（第11章）
空闲RQ
连续RQ
选择重发
回退N帧
数据链路控制协议（第5章）

216

第5章 数据链路控制协议

读完本章，应该能够

- 了解面向字符与面向位数据链路控制协议的不同类型及每个类型应用的范围。
- 理解单工、面向字符Kermit协议的应用与操作。
- 描述面向字符二进制同步控制协议用于轮询—选择多点网络中。
- 说明全双工面向字符协议的操作。
- 理解高级数据链路控制（HDLC）协议的帧格式与帧类型并说明它的操作选择方式。
- 说明HDLC 各种派生版本包括LAPB、LAPM、LAPD与LLC的应用范围与操作。
- 了解单链路与多链路控制过程的不同。

引言

数据链路控制层（有时简称为数据链路层）是牵涉到在串行数据链路上传送数据。链路可以是点对点的物理线路（双绞线、同轴电缆或者光纤），或者基于无线信道（如卫星线路），或者通过交换网络的物理或逻辑链路。传输方式可以是异步的或者同步的，以及面向字符或
217 面向位的传输控制协议。因此，数据链路层是所有数据通信应用操作的基础。

在最简单点对点的应用中，数据链路层通常直接为应用层服务，在复杂的应用中，如利用交换网，它为一组（高层）协议层提供良好的服务。依赖于应用，数据链路层提供的用户服务可以是简单**最佳尝试**（无连接）**服务**或者**可靠的**（面向连接）**服务**。这两种服务类型的时序图如图5-1所示。

无连接（最佳尝试）服务就是说差错校验位用于检测差错，发现包含传输差错的帧被链路层协议实体简单地丢弃，这也称为**未确认服务**，并由上一协议层的功能实现重传。例如，基于交换网（其中传输线路的BER非常低）的应用就是这样做的。因此，重传概率相对小，
218 LAN与ISDN就是如此。

面向连接方式的用户服务原语与第4章讨论的原语相同，回忆这类服务原语，数据链路协议采用差错控制与流量控制方法提供可靠的服务。协议高概率提交无差错、无重复的数据以及与提交同样次序传递的信息（数据块）。为了达到这一目的，在发送数据（信息帧）前，在两个数据链路协议实体之间用L_CONNECT服务建立逻辑连接，用适当的重传与流量控制协议传送所有数据。所有数据（信息）交换完毕，用L_DISCONNECT服务清除逻辑连接。

考虑数据链路层应用的范围，我们首先观察某些不同的应用环境，在5.1节与5.2节中将考察不同协议的详细操作。

5.1 应用环境

图5-2表示了某些应用环境，正如所见，在有些情形下，数据链路协议置于两个通信DTE中（例如，计算机），协议被认为**基于端对端操作**。在另一些情形下，它在本地连接链路上操
219 作，例如DTE对网络，协议被认为仅在**本地有效**。

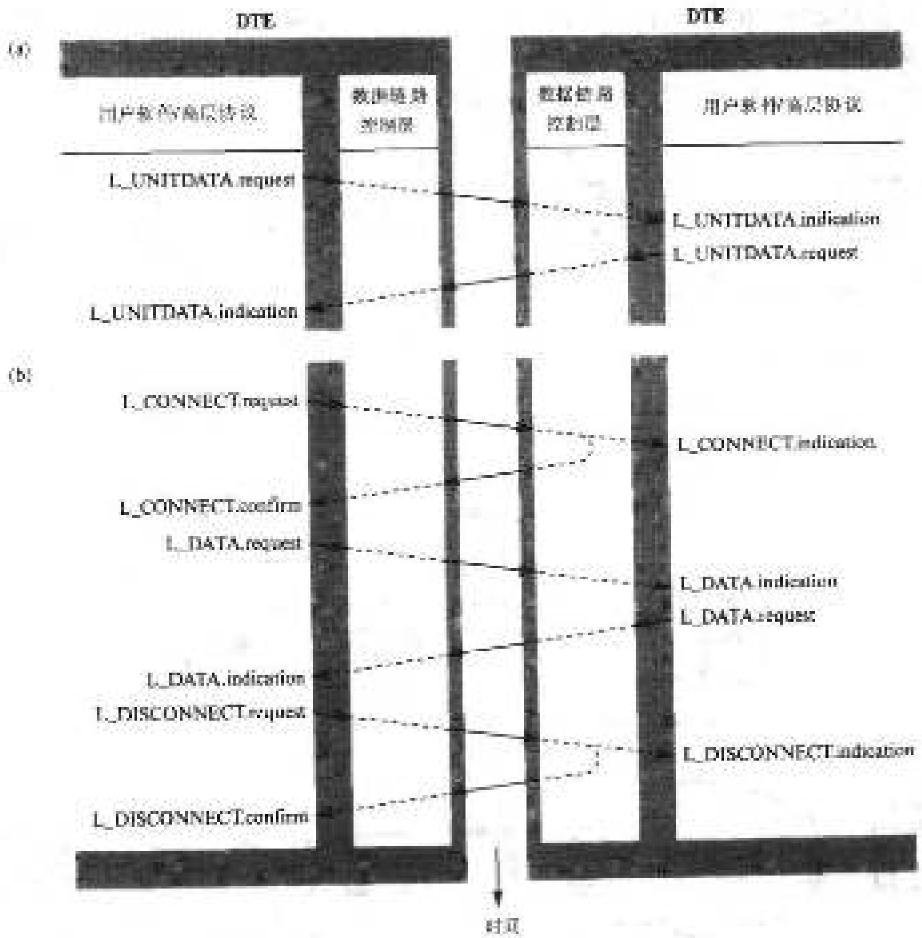


图5-1 数据链路控制层——用户服务原语

(a) 无连接（最佳尝试）服务 (b)面向连接（可靠的）服务

在图5-2 (a)中，数据链路是点对点的线路。它可以是直接物理连接（双绞线、同轴电缆或光纤），也可以用调制解调器经过模拟交换电话网建立线路，通过专用多路复用网络的线路，或基于无线的链路，如卫星或地面微波链路。数据链路基于端对端操作，在许多应用中，直接服务于应用层，所以通常用于可靠的面向连接服务。

所用数据链路协议的类型与两个通信DTE的物理距离及链路比特率有关。对于低比特率链路，如使用调制解调器，常采用面向字符的空闲RQ（停止等待）协议。这种协议的例子是Kermit协议与X-modem协议。这两者都是字节文件传输协议，它们广泛用于PC对PC的通信。它们与第4章中描述的基本空闲RQ协议十分类似。

对于高比特率链路，特别是那些长的物理距离，如基于无线的卫星链路或通过专用多路复用网络的线路，采用另一种（更有效）称为高级数据链路控制（HDLC）的连续RQ协议。 220
这个面向位的协议适用于许多不同的操作方式。

图5-2 (b)所示的应用结构称为多点或多站拓扑。正如所见，称为总线或数据主通道的单根传输线连接所有的计算机，所以我们必须以某种控制方式实施传输，保证不同时发生两个传输。这样的结构通常用于单独一台主（管理）计算机与一组分布式从属计算机通信。实例

是一台**后台存储计算机**控制百货公司的一组分布的销售点终端（计算机）或者制炼厂的一台管理计算机控制一组分布的智能（基于计算机）设备。所有传输是在主计算机与某台选中从属计算机之间进行，因此，主计算机控制所有传输的次序。

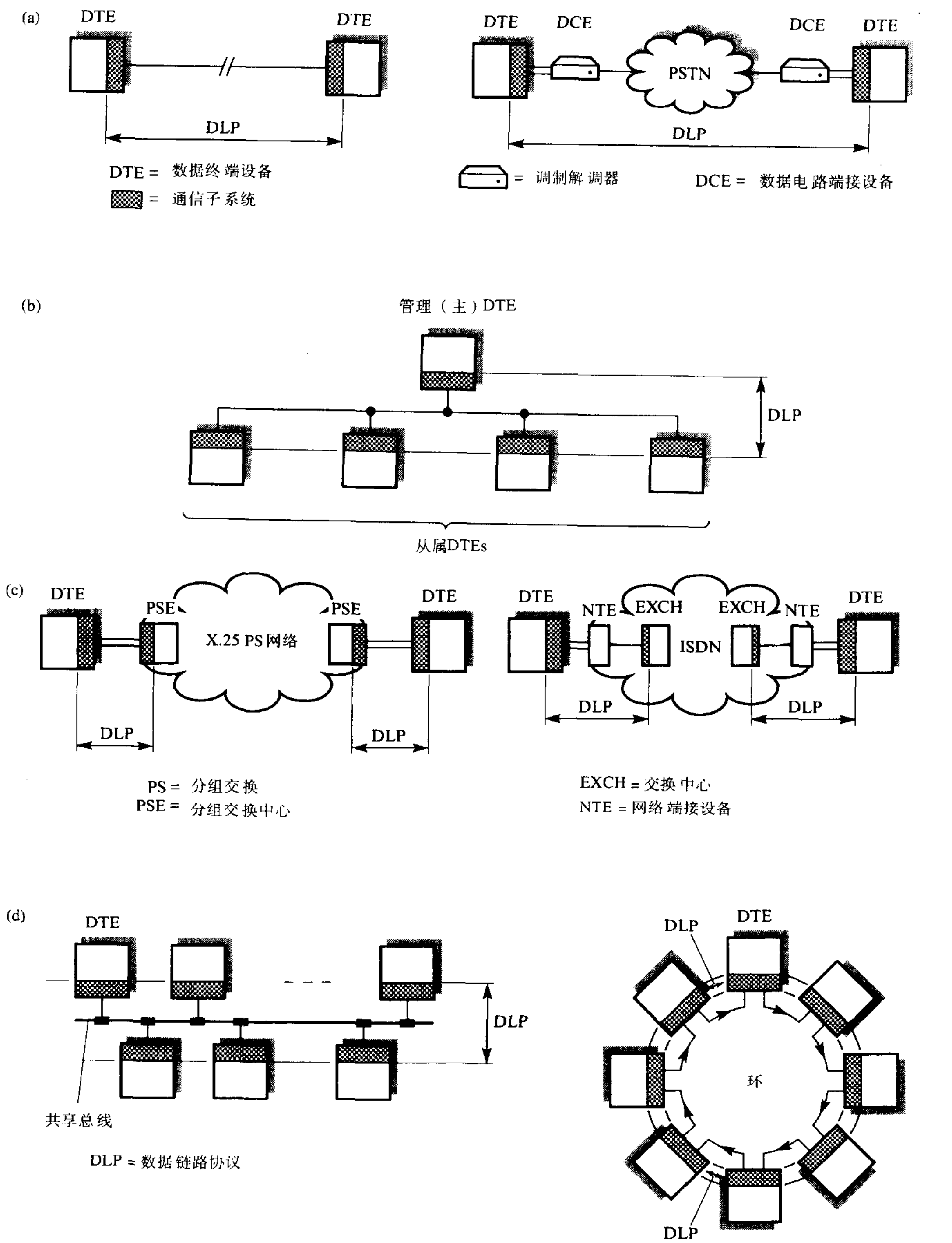


图5-2 数据链路协议应用环境
(a) 点对点 (b) 多点（多站） (c) WAN (d) LAN

合理地控制对共享传输介质的访问，采用面向连接的数据链路协议。关于这种结构，早期所用的协议是基于面向字符空闲RQ协议的发展，称为**二进制同步协议**（BSC）或**bisync**。最近的实现基于面向位HDLC协议的另一种操作方式，称为**正常响应方式**（NRM）。bisync与NRM两者都以**轮询—选择方式**操作，当主计算机想要由从某台从属计算机接收数据，发送**轮询消息**；如果主计算机要发送数据到从属计算机，它发送**选择消息**。

图5-2(c)所示的两种结构都包含与交换WAN相关的应用。在第一个实例中，由于是在X.25分组交换网中，链路协议仅是本地有效，在DTE与本地的数据电路端接设备（DCE）之间操作。这种网络所用的X.25协议组仅应用于DTE与DCE之间的本地链路。X.25的数据链路协议也基于HDLC，称为**平衡式链路访问规程**（LAPB）。

第二个实例使用交换电路数据网，如ISDN。一旦通过网络建立一条线路（连接），它提供等价的点到点链路，称为虚电路，用于数据传输阶段。协议可以面向连接（可靠的），或者无连接（最佳尝试），分别称为**帧交换**与**帧中继**。另外，ISDN呼叫建立规程，使用单独一条链路，称为**信令**或**D信道**实施。这里使用的链路协议是HDLC的一种变种，称为**D信道链路访问规程**（LAPD）。

最后，图5-2(d)所示的两种结构包含LAN的应用，这种网络的特点是链路距离相对短，链路数据误码率低，并以高比特率（~10 Mbps）操作。所有结果差错不常见，而端对端帧传送时间很快。这种方式，重传与流量控制功能由两个端系统（DTE）的高层协议完成。LAN所用的链路层协议是HDLC的一个子类，称为**逻辑链路控制**（LLC）。

总之，存在各种数据链路协议，每一个供特殊应用环境使用。 221

5.2 面向字符协议

面向字符协议用于点对点与多点应用。它们的特征是使用选择传输控制字符实现链路管理的各种传输控制功能。诸如帧的开始与帧的结束定界，差错控制与数据透明等。

在第4章我们讨论面向字符协议，考虑点对点的数据链路与单工（单个方向）信息帧流，介绍链路协议各个方面。但是，在大多数实际应用中，我们必须扩充概念引入双向传送数据（信息）。如果在多点配置中，还包括多个通信方，需要有一个控制共享传输介质访问的方法。我们将把这些问题作为各种协议讨论。

5.2.1 单工通信协议

由于仅允许单工（仅是一个方向），这类协议是最简单的，数据传送是在点对点的数据链路上从一台计算机（DTE）到另一台计算机（DTE）。拓扑结构如图5-2(a)所示。一个典型的应用是从一台计算机到另一台计算机传送数据文件，这种功能最广泛使用的协议之一是Kermit协议。

Kermit协议广泛地用于在点对点的数据链路上从一台计算机（例如，个人计算机）到另一台计算机传送指定文件内容或文件组。链路可以是使用调制解调器经过交换电话网建立的电路或用适当线路驱动器与接收器的一对双绞线电路。通常使用同步传输。Kermit协议是第4章讨论的空闲RQ（停止等待）协议的一个实例。

Kermit协议的许多版本允许在两台个人计算机之间传送文件或者在一台个人计算机与文件服务器或主机之间传送文件。每种版本的基本文件传送机制是相同的。主要的差别是源计算机用户如何开始获得（通过Kermit程序）访问目标计算机的Kermit程序。我们将考察在两个单用户计算机之间的文件传送版本。

Kermit程序已在两个系统中运行后，两个用户都可使用一组简单命令。命令的时序图如图5-3所示。

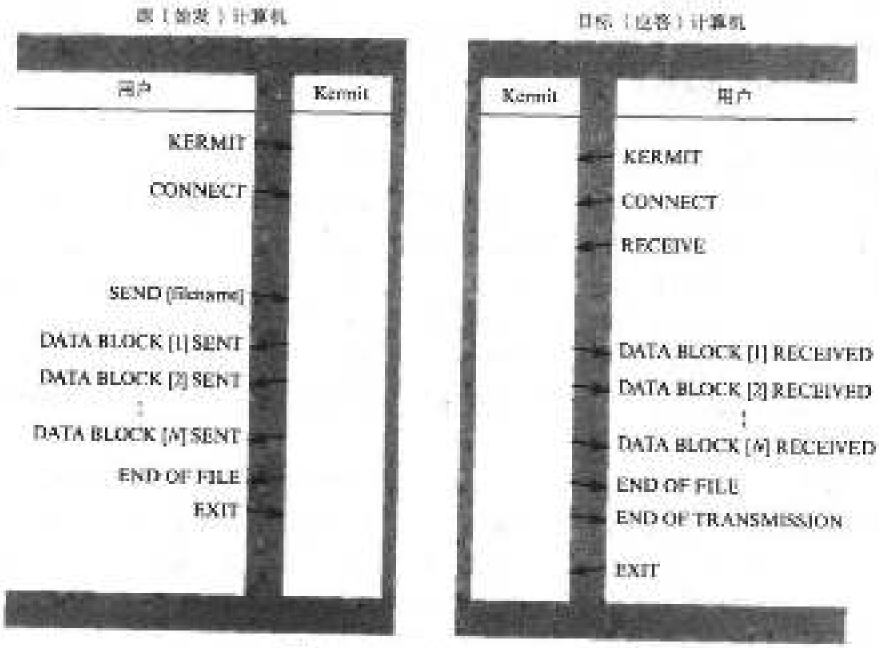


图5-3 Kermit协议的用户命令

如果使用调制解调器，那么一个调制解调器必须设置为始发方式，而另一个调制解调器设置为应答方式。当然，两个调制解调器必须设置以相同的波特率操作。每个用户运行Kermit程序，然后输入CONNECT命令，其结果在两个系统间建立一条物理链路。系统中用户
222 想要接收一个文件（或文件组），输入RECEIVE命令，发送系统中用户输入SEND命令后跟文件名。发送系统中的Kermit程序作为一个整体传送文件（或文件组），当传送每个文件段时，在用户双方屏幕上有一个消息输出。所有文件段传送完毕，双方用户退出Kermit程序，用EXIT命令返回到本地操作系统。若以相反方向传送文件，命令的次序相反执行。

我们可见Kermit协议不是简单的数据链路协议，因为它执行若干附加功能，诸如文件读/写、文件分段与文件重新组装。每一个功能有相关的帧类型，正如图5-4(a)所示的标准的帧格式。

Kermit协议帧格式与第4章所讨论的帧格式有两个主要不同。首先，长度字符取代传输控制字符ETX指定每个帧的长度；其次，信息（数据）帧、ACK帧与NAK帧都是相同基本格式。每个帧的结束用一个附加（冗余）的控制字符：回车（CR）。利用长度字符的好处是帧（文件）内容可以是文本字符，也可以是二进制字节，由于接收方简单接收与附加适当长度的字符或字节（由帧的头部指定）可重新组装文件。一般接收计算机的用户或者知道文件类型或者能从它的名推出文件类型。

223 发送的文本文件内容作为字符块序列发送，每块80个字符，每块用回车字符与换行字符结束。但是，发送二进制文件，只简单作为8位字节串发送。文件内容（文本或二进制）中的任何格式控制字符在发送前必须编码以保证传送过程中通信设备状态不受影响，这是某些调制解调器流量控制操作的一个特性。每当检测出一个控制字符转换成双字符可打印字符序列号，它由一个控制前缀字符ASCII#后跟ASCII表中位于同一行的可打印字符组成，其中列0或

列1的控制字符分别对应列4或列5的可打印字符。例如，Ctrl-A变成#A，CR变成#M，FS变成#\，以及#字符前再附加一个#变成##。

为了传送文件，Kermit协议实体交换帧序列如图5-4(b)所示。在启动文件传送之前发送的第1个信息帧是一个发送邀请（S）帧，它包括协议的参数表，如最大帧长度与重传超时间隔，接收方返回一个确认（Y）帧，同意传输控制参数。

然后，发送方着手开始传送文件内容。首先发送包括文件名的一个文件头（F）帧，接着发送包含帧内容的数据（D）帧序列，文件最后的数据帧发送后，发送一个文件结束（Z）帧通知接收方。然后，用同样的方法发送其他文件。最后，当所有文件发送完毕，则源计算机发送一个事务结束（B）帧。

Kermit协议是一个空闲RQ协议的实例。因此，发送每个信息（I）帧后，源计算机等待直到它接收到肯定的确认（Y）帧（块校验和正确），或者否定确定（N）帧（BCC不正确）。也允许这两个帧中任一个损坏的可能性，每次发送新帧启动定时器。每个I帧的发送序列号模64增1，每个ACK(Y)与NAK(N)帧中的接收序列号携带相同的序列号用作肯定确认或否定确认。 224~225

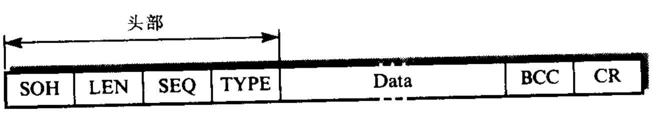
我们已描述Kermit的最基本的特征，查阅本章参考资料，包含更详细叙述。

5.2.2 半双工通信协议

大多数面向字符的协议以半双工、停止—等待方式操作。一些大的计算机生产厂家常常有各自的版本协议，彼此略有不同。最著名的是IBM开发的，称为二进制同步控制协议，通常简写为bisync或BSC。由于它是ISO面向字符协议基础，称为**基本型**协议，我们用BSC作为实例。

正如名字暗示，BSC通常采用同步传输控制方案。它是面向连接的协议，最初用在多点（多站）应用中，有一个单独的主站（计算机）控制到或来自一组从属站的所有信息传输。如果所有站点安放在不同的部门，并使用调制解调器，则从属站与主站连接采用**多点网络**；如果所有站点都在同一个地方，并使用线性驱动器/接收器，则从属站与主站连接用**多站总线网络**，这两种配置如图5-5所示。

(a)



SOH = 标记一个帧的开始

LEN = 这个字符后面直到包含BCC在内帧中字符/字节的个数。数字以余32表示法编码，采用从ASCII#（十进制35）到ASCII~（十进制126）的单个字符，其中#表示长度为3（没有数据），~表示最大长度为91。

SEQ = 帧的发送序列号，序列号模64增递增1，也编码为单个字符，范围从ASCII SP（十进制32）作为0到ASCII _（十进制95）作为63。

TYPE = 用单个字符编码的帧类型
- S = 发送邀请（参数）
- F = 文件名
- D = 文件数据
- Z = 文件结束
- B = 事务结束
- Y = 确认（ACK）
- N = 否定确认（NAK）
- E = 致命差错

Data = 帧内容

BCC = 块校验字符

CR = 块结束标记（ASCII回车符）

图5-4 Kermit操作

(a) 帧格式与类型 (b) 帧序列

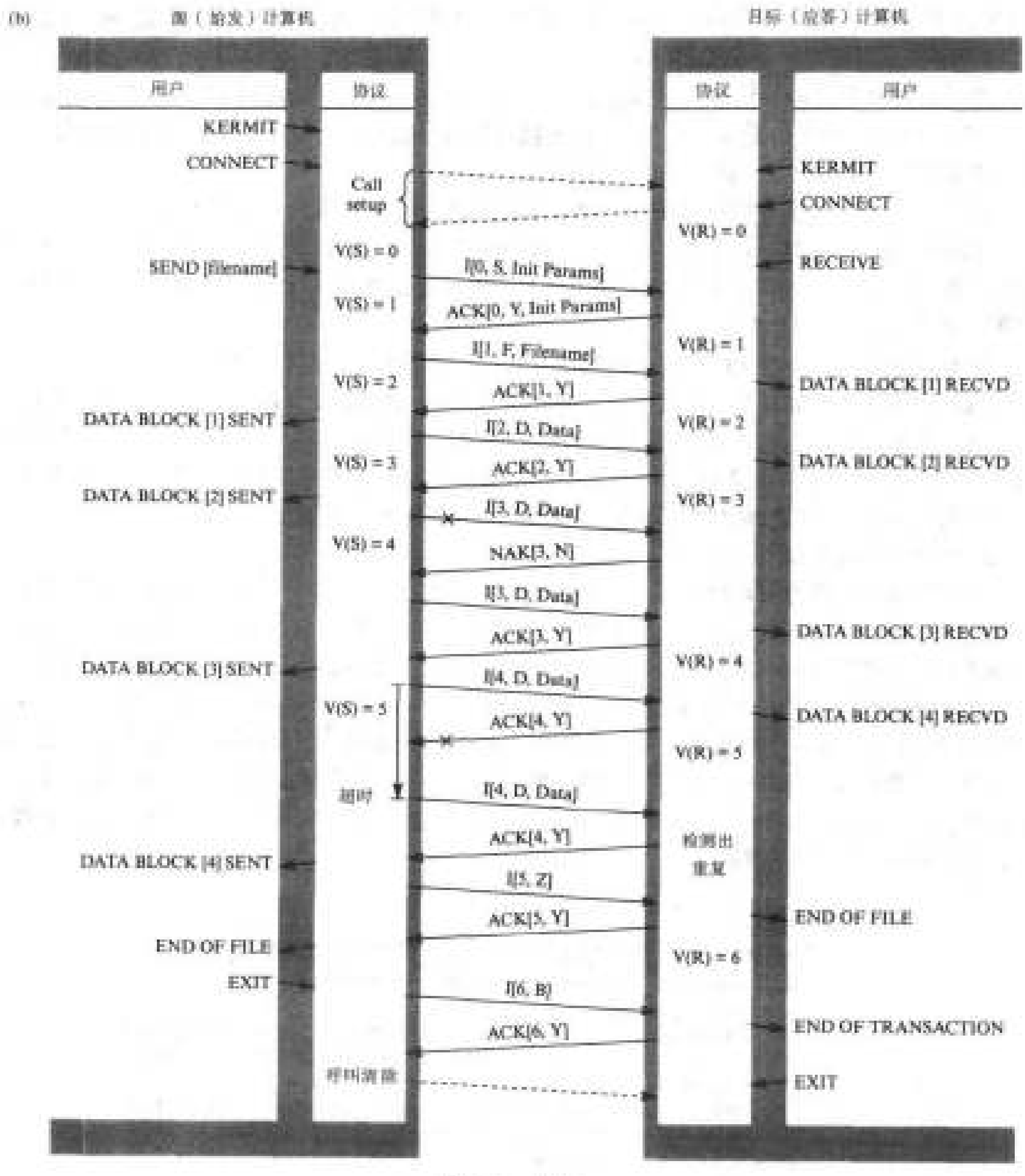


图5-4　（续）

1. 帧格式

226　回忆链路管理执行的各种功能，除通常的信息（携带数据）帧（块）外，还需要控制帧，面向字符同步传输的接收器也必须能实现字符（字节）同步与帧同步两者。

在基本BSC中，使用由EBCDIC（或者基本方式ASCII/IA5）定义的传输控制字符，以实现相应功能。我们在第4章中描述过这些控制字符的作用。它们的功能在表5-1中更完整地列出。

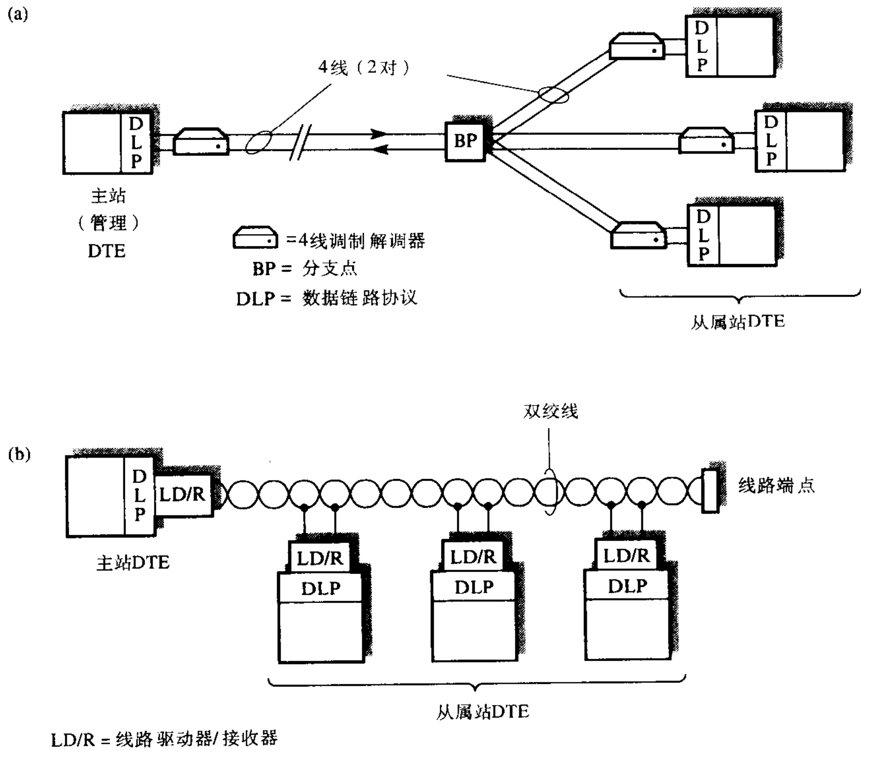


图5-5 面向字符的总线网络

(a) 多点 (b)多站

表5-1 BSC所用的传输控制字符

字 符	功 能
SOH(TC1)	头部开始：表示一个信息帧（块）的头部的开始（如果它出现）
STX(TC2)	文本开始：表示头部的终止（如果它出现）与文本串的开始
ETX(TC3)	文本结束：表示文本串的结束
EOT(TC4)	传输结束：表示一个或多个文本（信息）块传输的结束并终止（清除）连接
ENQ(TC5)	请求响应：用作请求远程站给出响应，响应可能包括站的身份和/或状态
ACK(TC6)	确认：由接收方发送的肯定确认，作为正确接收来自发送方信息的响应
DLE(TC7)	数据链路转义：用来改变选择的传输控制字符的意义
NAK(TC8)	否定确认：由接收方发送的否定确认，作为未正确接收来自发送方信息的响应
SYN(TC9)	同步空闲：在同步传输控制方案中，用于接收器实现或保持（空闲情况）数据终端设备之间的字符同步
ETB(TC10)	传输块结束：表示当信息分成许多数据块（帧）时，一个数据块结束

信息帧的不同类型（在BSC中称为数据块）如图5-6(a)所示。BSC采用面向字符同步传输方式，因此，所有发送的数据（控制）块前面至少有两个SYN字符以使接收方能实现字符同步。短的用户信息（小于定义的最大长度）可作为单数据块发送，但长的信息用多块发送。当有头部字段出现时，一般它用作定义如何解释数据字段。另外，基本模式是数据块结束定界符（ETX或ETB）之后有一个块校验和，这是一个纵向（列）奇偶校验（参见3.4.2节），它

以STX字符表示校验开始，并以特殊的块结束定界符（ETX或ETB）结束。由于奇偶校验仅有有限的差错校验能力，BSC常采用CRC-16代替单个BCC计算双字符（字节）CRC。这两种
227 方案，对每个发送数据块的字符的个数都有限制，这个限制由链路的BER决定，选定最大块长度保证绝大多数块无差错的接收。对较长的信息分成多个具有固定长度的短的数据块发送，每一个数据块以控制字符ETB结束，最后一个数据块以控制字符ETX结束。

BSC协议的各种控制帧如图5-6(b)所示。ACK与NAK控制字符有两个功能：

• 首先是确认：返回的ACK或NAK作为先前所发送数据块的响应，因而包含标识符（序列号）。
• 其次是作为对选择的控制信息的响应：ACK表示所选择站能接收数据块，而NAK表示
228 不能接收数据块。

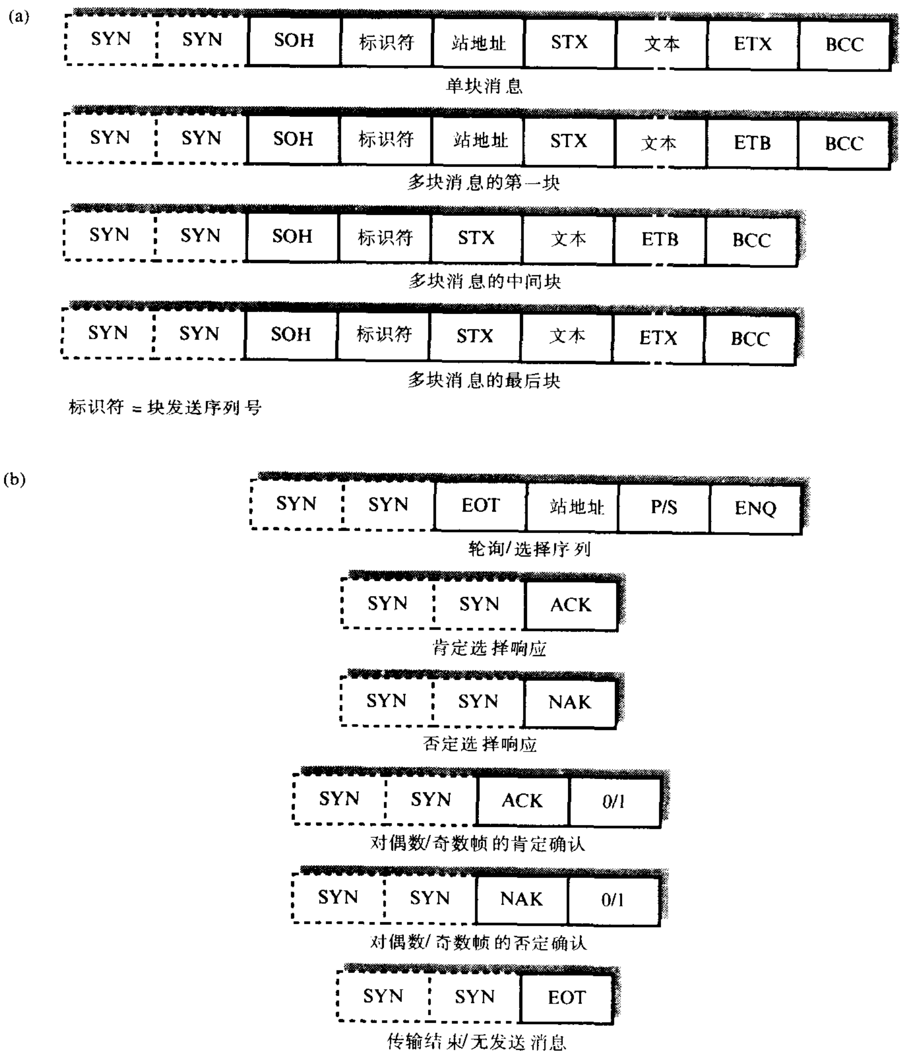


图5-6 BSC块/帧格式

(a) 数据块/帧 (b)监管帧

ENQ控制字符用作轮询与选择控制帧，轮询或选择从属站的地址后跟P（轮询用）或者S（选择用）字符，依次跟着ENQ字符。

最后，EOT控制字符有两个功能：

• 首先是标志整个信息交换序列的结束，并清除通信双方之间的逻辑链路。
• 其次是将链路重置于空闲状态。

2. 数据透明性

在3.2.3节中，当发送纯二进制数据而不是发送字符串时，使用DLE转义字符实现数据的透明传输，即将图5-6中各种帧字符序列定义为DLE/STX，DLE/ETX等等。在发送时，每当文本中出现与DLE字符对应的二进制码型时，即增加（插入）一个额外的DLE。接收方则进行相同的检测，当发现两个连续DLE时，则在数据进一步处理前将插入的DLE删去。当用透明方式操作时，差错控制方法有进一步的差别，不能用每块8位纵向奇偶校验，而是更完善的多项式代码，在每块的结束处不是8位块校验字符BCC，而是16位的帧循环冗余校验CRC。

3. 协议操作

我们先前描述，主计算机（站）负责共享数据链路上所有传输的调度。它利用轮询控制消息请求特定从属计算机发送等待的数据信息，并利用选择控制消息询问所选的从属计算机是否准备好接收数据信息。

图5-7(a)表明典型轮询和选择序列。在多站线路上交换帧的典型序列如图5-7(b)与图5-7(c)所示。(b)部分表示与选择操作有关的成功的与未成功的两个实例，而(c)部分表示有关轮询操作的两个序列。

为了选择特定从属站，主站发送ENQ选择控制消息，ENQ字符紧跟在所选从属站地址后面。假设所选从属站准备接收信息就绪，则返回一个ACK控制消息作为响应。然后主站发送信息，或者发送一个数据块（如图5-7所示），或者发送一系列数据块，最后一个数据块以一个ETX字符结束。从属站收到数据块并存入缓冲器，同时对数据块重新计算奇偶校验序列。假定没有传输差错，则对每个块用ACK控制消息响应。最后，当全部信息发送完毕，主站发送一个EOT控制消息，以结束信息传输并清除逻辑连接。 229

在某些情况，选择一个从属站时，在数据发送前不需要等待ENQ控制消息的确认。例如，如果一个从属站事前已被选择，并且逻辑连接未关闭。此时，主站在发送选择控制消息之后，不必等待ACK（或NAK）响应，可立即发送信息。这称为**快速选择序列**。

在轮询过程中，主站首先发送ENQ轮询控制消息，在ENQ字符前带有被轮询从属站的地址。假定被轮询从属站有信息等待发送，它将发送这些信息作为响应。主站一接收到数据块，就重新计算奇偶校验序列，假定传输无差错，则确认其正确收到。最后，全部信息已发送并被确认后，主站发送EOT控制消息并清除逻辑连接。 230~231

图5-7说明BSC是一个空闲RQ协议。由于发送每个数据块之后，在发送下一个数据块前，发送方等待ACK或NAK控制消息。在收到后者情形下，重发有错误的数据块。利用附加的NAK控制消息保证一收到NAK消息即重发有错误的数据块，而不是等到超时再重发。正如4.2节讨论，如果传输块全部损坏，仍需要采用超时机制以保证重发受影响的帧。标识符（发送序列号）用于使接收方检测出重复块。

注意，在BSC中，发送序列号简单地增1（按一个约定数进行取模运算），而ACK帧与NAK帧的接收序列号简单按模2增1（0或1）。因此，接收序列号为0指的是偶数帧，而接收序列号为1指的是奇数帧。因为可能仅有一个帧重复，所以这足够使接收方能检测出帧的重复。

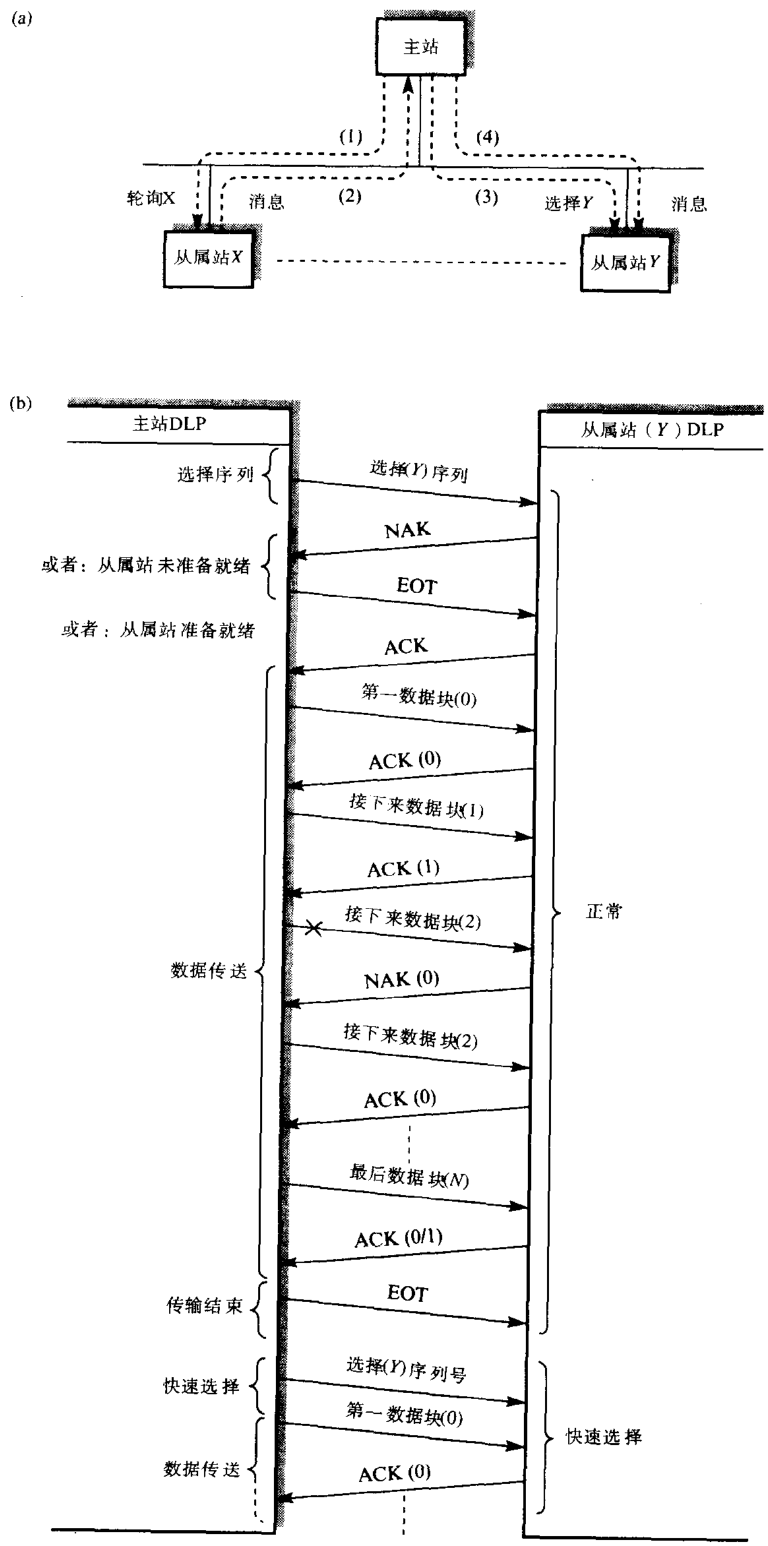


图5-7　BSC帧序列

(a) 轮询—选择结构　(b) 选择　(c) 轮询

(c)

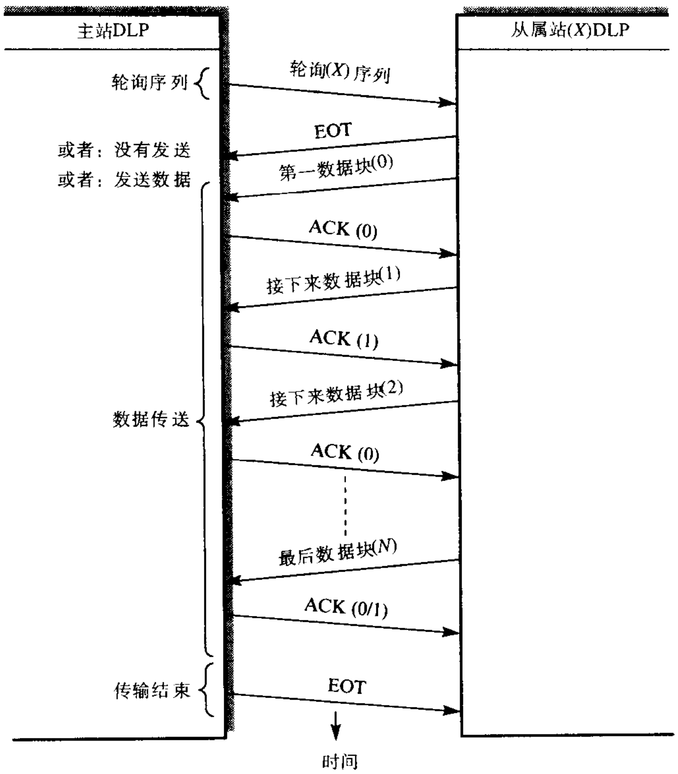


图5-7 （续）

4. *用户接口*

区别链路层提供服务与链路层协议实体的操作是重要的。为了说明这个，链路层提供的用户服务与BSC协议的各种信息块（控制与数据）之间的关系如图5-8所示。正如所见，最初的选择控制消息被确认，将它作为远程站已准备就绪接受信息的证实。但是，对于轮询序列，最初的轮询消息并不返回ACK作为响应，所以证实原语必须由本地的协议实体在发出轮询消息后产生。后面的链路断开过程也遵循相似过程。

与Kermit协议相同，BSC执行分段与重新装配功能。因此，接收到L_DATA.request原语（参数为信息），发送协议实体把信息分段为传送数据块的序列。同样地，在用L_DATA.indication原语传递给用户前，接收的实体重新装配这些块为一个完整的信息。

由于BSC事实上是一个半双工协议，即使物理链路支持全双工传输，BSC也不能加以利用。但是，由于BSC协议需要的缓冲存储容量最少，因此在各种类型网络中仍然广泛应用。然而，最近几年，已有变化，趋向于更灵活有潜在效率的面向位协议。当然，计算机网络要求透明性工作。

5. *协议性能*

在4.2.4节讨论过空闲RQ协议的基本链路效率（利用率）。然而，BSC的主要用法是存在一个主站，它要对多个从属站发送与接收信息。这种配置有一个重要的性能参数是链路对所有从属站轮询与选择花费的平均时间。

实际上，由于空闲RQ相对于连续RQ链路利用率低。多站链路基本上使用空闲RQ协议操作，其数据速率可达64 Kbps。对这样的链路，在轮询或选择序列时发送消息所花费的时间是 232

主要的。例如，如果平均信息是1000位而数据速率是10 Kbps，发送一个信息要求的时间是0.1 s。轮询（或选择）序列相关的控制是短的（比如说30位）。因而发送这些消息的时间也是短的（在10 Kbps下是0.003 s），甚至允许增加少量附加的时间，比如说0.001 s去处理这些消息，所以每个轮询（或选择）序列的全部时间（0.004 s）仍然比发送信息时间少。

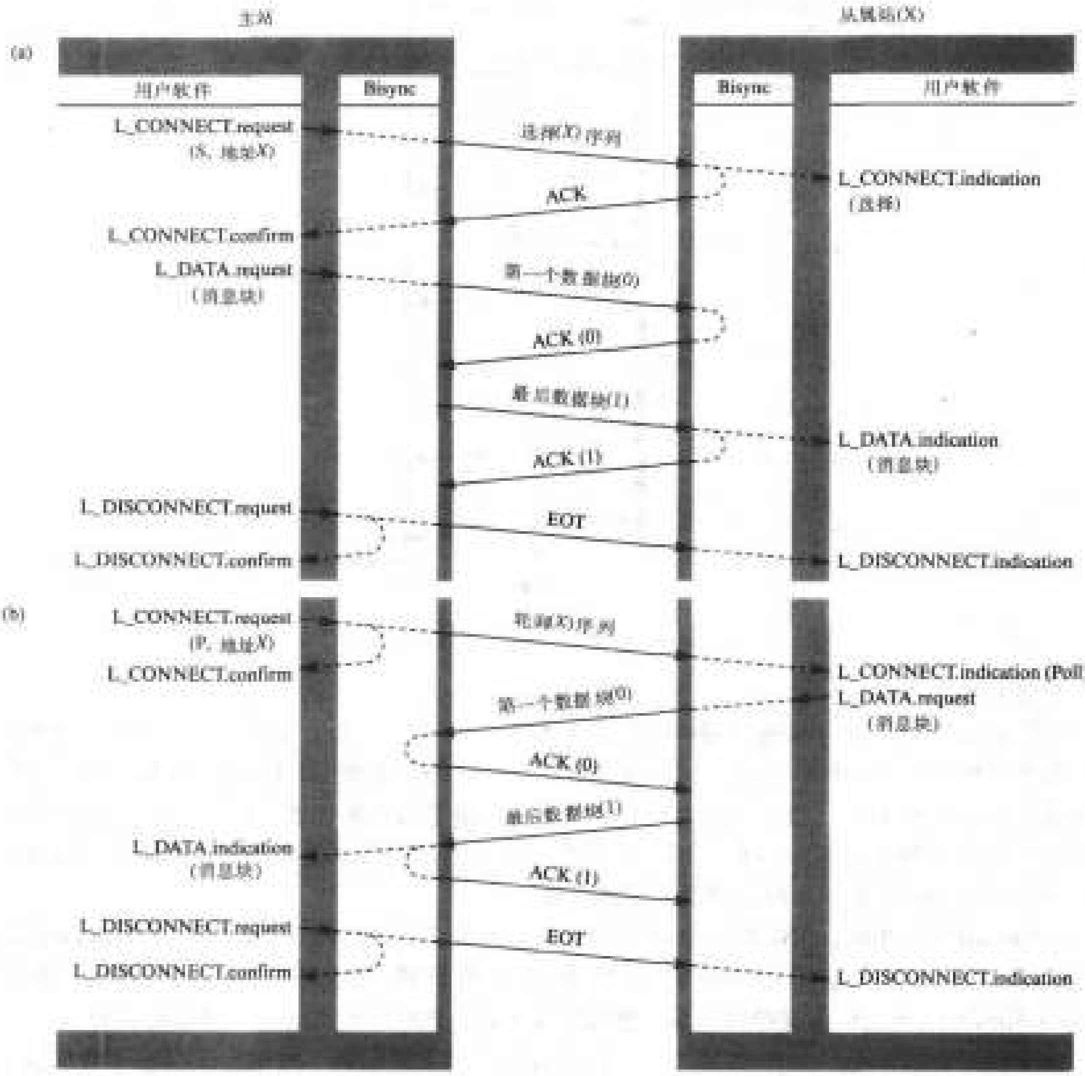


图5-8　用户/链路层相互作用

(a) 选择　(b) 轮询

在无任何信息发送时，主站轮询所有从属站要求最小时间是轮询单个从属站时间的N倍，
233 其中N是链路上从属站的个数。当信息发送准备就绪，轮询所有从属站平均时间增加，它与信息生成的平均速率有关。当信息生成平均速率接近或超过链路比特率时，链路过载，延迟进一步增加，则最大轮询时间发生。

一般，轮询每个从属站的平均时间可表示作

$$T_{avr}=\frac{T_{min}}{1-M_r T_{ix}}$$

其中，T_{min}是轮询所有从属站的最小时间，M_r是生成信息的平均速率，T_{ix}是发送一个平均长度信息的时间。如果同T_{ix}比较起来M_r低，则T_{avr}近似等于T_{min}。然而，当M_r增加时，T_{avr}也增加。

实例5-1

在一个多点的数据链路上，1台计算机（主站）与10个块方式的终端（从属站）之间采用BSC协议控制信息流。链路数据速率R是10 Kbps，信息平均长度N_i是1000位。如果轮询信息与相关的ACK是30位以及处理这些信息的全部时间是1ms，如果信息生成的平均速率如下：

(a) 每分钟1个信息

(b) 每秒6个信息

确定轮询每个终端的平均时间。

解：假定数据误码率与信号传播延迟的时间可忽略。

发送一个平均信息的时间T_{ix}是：

$$\frac{N_i}{R}=\frac{1000}{10^4}=100\text{ms}$$

发送一次轮询与它的ACK的时间是

$$\frac{30}{10^4}=3\text{ms}$$

因此轮询一个从属站的时间是

$$3+1=4\text{ms}$$

轮询所有从属站的最小时间

$$T_{min}=10\times 4=40\text{ms}$$

现在

$$T_{avr}=\frac{T_{min}}{1-M_rT_{ix}}$$ 234

(a) $M_r=$ 每分钟1个信息$=\frac{10^{-3}}{60}$个信息ms^{-1} 因此

$$T_{avr}=\frac{40}{1-\frac{10^{-3}}{60}\times 100}=40\text{ms}$$

(b) $M_r=$每秒6个信息 $=6\times 10^{-3}$个信息ms^{-1} 因此

$$T_{avr}=\frac{40}{1-6\times 10^{-3}\times 100}=\frac{40}{0.4}=100\text{ms}$$

5.2.3 全双工通信协议

少量面向字符协议以全双工方式操作。作为实例，我们将考察早期ARPANET网络所用的数据链路协议，它经过链路连接称为**接口信息处理机**（IMP）的内部网络交换结点控制信息帧的流量。协议连接两个交换结点，在点到点全双工方式链路上操作。

协议同时支持两个方向的信息帧传输（全双工），每个方向利用连续RQ传输控制方案。

协议用一个有效发送窗口操作，对于地面链路有效发送窗口值为8，而对于卫星链路，值为16。为了保证连续的帧流量，任何时刻会有8（或16）条独立停止—等待信息流在工作。

为了达到这一目的，一条物理链路作为8条（或16条）独立的**逻辑链路**操作，对每条逻辑链路上帧流量的控制是由逻辑链路本身的停止—等待协议管理。在物理链路上，发送的每个帧头部中的发送序列号是两个字段的连结：1位序列号字段与**逻辑信道号**（LCN）字段，其中序列号（0或1）是空闲RQ（发送与等待）协议中的发送序列号，逻辑信道号是发送该帧的逻辑信道。

面向字符与面向位的大多数双工方案有个共同点，流向某个方向的信息帧流的确认信息是由流向反方向的信息帧头部捎带。因此，使用单一的信息帧类型，它的头部的各个字段涉及某个特殊的功能。数据链路协议的帧格式与头部各个字段以及协议的操作如图5-9所示。

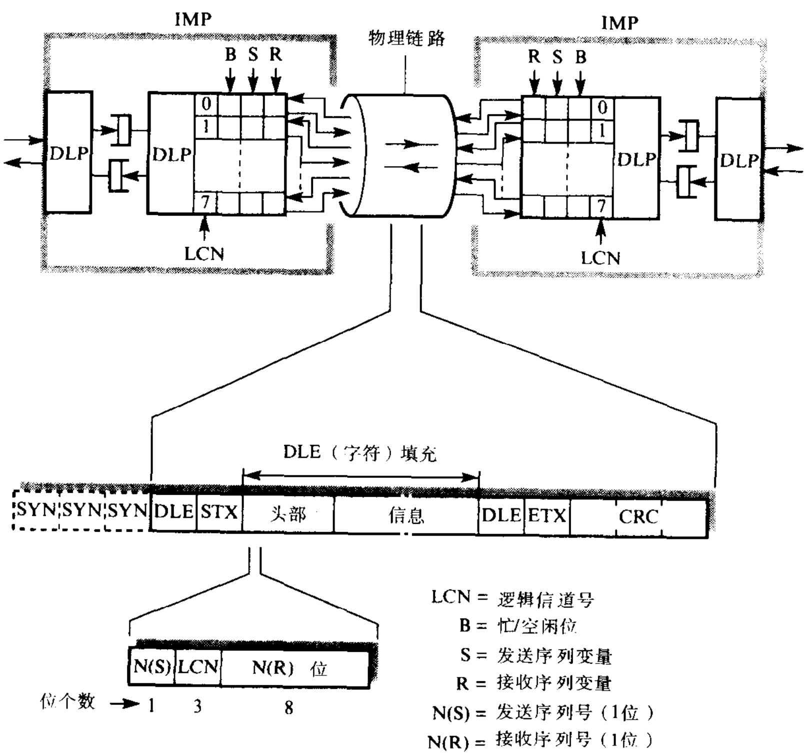


图5-9 ARPANET IMP到IMP数据链路协议说明

为了支持双工的帧流，正向与反向的物理链路各有8（或16）条逻辑链路，因此，每条逻辑信道，数据链路协议在链路每一端有独立发送与接收序列变量。发送序列变量（取值0或1）
235 是正向信道上发送下一个新帧指定的发送序列号，而接收序列变量是反向信道上期待接收下一个信息帧的接收序列号。另外，为了保证每个正向信道以停止—等待方式操作，信道的每一端也有一个忙/空闲位指明信道是否忙，即这个信道还未解决确认。

在ARPANET网络中，每个帧作为一个单独的实体处理，即同一（用户）信息的帧都独立地处理。收到IMP传送的帧，发送数据链路协议扫描每个逻辑信道忙/空闲位确定某个信道是否空闲，如果空闲，在帧的头部插入相应的发送序列号（0或1）与逻辑信道号；如果不空闲，帧留在输入队列中等待空闲信道。

协议利用24位（3个字节）CRC用于纠错。纠错利用帧头部的确认字节，确认字节的8位是

反向传输的8个逻辑信道的有关帧流的8个接收序列号的连结。接收方收到一个帧，数据链路协议读取帧头部的确认字节，对所有逻辑信道，按4.2节描述的空闲RQ协议解释对应的位。在这种方法下，每次接收一个新帧，确认信息是对于所有逻辑信道而言的。这就是说，与链路利用 236
8位（或16位）发送窗口有相同效果。也就是说使用了一种隐式的确认方案（仅用ACK）。

最后，虽然许多面向字符的协议依然广泛使用，但廉价集成电路支持更高效率的面向位的协议，就是说所有新的（目前许多）协议都是面向位类型。

5.3 面向位通信协议

所有新的数据链路协议都是面向位协议，回忆这种协议是用位模式定义，而不是用传输控制字符标志一个帧的开始与结束。接收器逐位搜索收到二进制数据流，确定位模式帧的开始与结束。标志一个帧的开始与结束（称为**帧定界**）的三种方法如图3-13所示。它们是：

- 惟一的帧开始与帧结束位模式（01111110），称为标志，并具有0位插入。
- 惟一的帧开始位模式（10101011），称为开始定界符并在帧开始的头部有一个长度（字节）计数。
- 惟一的帧开始与帧结束的定界符，这两个定界符包含位**扰动编码**。

一般，第一种方法用于高级数据链路控制（HDLC）协议，而其他两种方法用于逻辑链路控制（LLC）协议。实际上，所有面向位的协议都是HDLC协议派生出来的，所以我们首先描述高级数据链路控制。

5.3.1 高级数据链路控制

HDLC协议是ISO为点对点与多点（多站）数据链路制定的国际标准。它支持全双工透明方式操作，目前广泛用于多点网络与计算机网络。尽管缩写HDLC已被广泛接受，但一些大的制造商与其他标准化组织仍然使用它们自己的缩写词，包括IBM公司的SDLC（同步数据链路控制），它是HDLC的先驱；美国国家标准化协会（ANSI）的ADCCP（高级数据通信控制规程）。

由于HDLC是通用的数据链路控制协议，我们在如图5-10所示的不同网络配置中使用它。在HDLC中，由主站发送到从属站的帧称为**命令**，而由从属站发送到主站的帧称为**响应**。（a） 237
与（b）部分所示的两种配置只有一个主站，称为**不平衡配置**，而（c）部分有两个主站，称为**平衡配置**。平衡配置中，由于每个站都包含有主站与从属站，所以也称为**复合站**。

HDLC有三种操作方式：

1）**正常响应方式**（NRM） 用于不平衡配置。在这种方式中，从属站仅当主站有特殊指令时，才能发送。链路可以是点对点或多点链路。后者只允许有一个主站。

2）**异步响应方式**（ARM） 也用于不平衡配置，允许从属站在没有得到主站许可的情况下，开始发送。通常用于点对点配置和双工链路，允许从属站相对于主站异步发送帧。

3）**异步平衡方式**（ABM） 主要用于计算机通信的双工点对点链路，或用于计算机到PSDN之间的连接。这种操作方式，每个站是平等的，均有主站与从属站的功能。它使用X.25协议，将在8.2节讨论。

1. 帧格式

与BSC不同，在HDLC中数据信息与控制信息均以标准帧格式传送，图5-11表示帧格式以 238
及帧头部的控制字段定义的不同帧类型。HDLC有三种不同类型帧：

（1）**无编号帧** 用于链路的建立与断开等控制功能，因为不包含任何确认信息。而确认信息包含序列号，所以这些帧称为无编号帧；

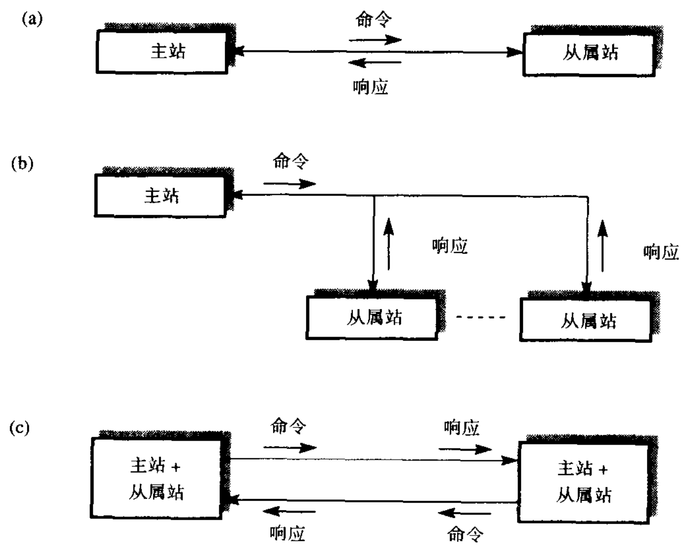


图5-10　另一种HDLC网络配置

(a) 具有一个主站与一个从属站的点对点配置

(b) 具有一个主站与多个从属站的多点配置　(c) 具有两个主站与两个从属站的点对点配置

（2）**信息帧**　用于传送有效信息或数据，通常简称为I帧。当链路以ABM或ARM方式通信时，I帧也可用于反向I帧流的确认信息捎带；

（3）**监管帧**　用于差错控制与流量控制，因此包括发送序列号与接收序列号。

标志字段作为帧起始与帧结束的定界符，为了得到数据透明性，采用“0”位插入与“0”位删除，在第3章中已讨论过，此处不重复。

帧校验序列（FCS）使用16位循环冗余校验（CRC），是对两个标志定界符之间整个帧内容的校验。HDLC使用的生成多项式通常是CRC-CCITT：

$$X^{16}+X^{12}+X^{5}+1$$

利用3.4.3节描述的过程生成CRC，通过使检测更为鲁棒的过程得到增强。一个FCS生成检测是在除之前，被除数的尾部添加16个“1”（代替16个“0”），并取余数的反码使检测更为可靠。如果传输正确，接收方计算的余数不是全0，而是一个特定的位模式0001110100001111。

地址字段的内容取决于所用操作方式。在正常响应方式（NRM）中，例如，多站点线路，每一个从属站被分配一个惟一地址。每当主站与特定从属站通信时，地址字段包含从属站的地址。并且，某一地址可能分配给不止一个从属站，这称为**组地址**。发送带有组地址的所有帧都能被这个组组的所有站接收。**广播地址**用来向链路上所有从属站发送帧。

当从属站向主站返回响应消息（帧），地址字段时常包含惟一的从属站地址。大型网络可能含有许多从属站。这时，地址字段就多于8位。每个8位字段的最后1位（lsb）用于指明接下来是否有另一个8位组（lsb=0）或者是否是最后一个8位组或惟一8位组（lsb=1）。应该指出，
239～240 在异步平衡方式（ABM）中，因为仅是点对点的直接链路，所以地址字段并不这样使用。通常用它表示命令方向与相关的响应。

各种控制字段位的定义，如图5-11(b)所示。监管帧中S字段与无编号帧中M字段用于定义特殊帧类型。发送序列号与接收序列号（N(S)与R(S)）连同差错与流量控制方法一起使用。

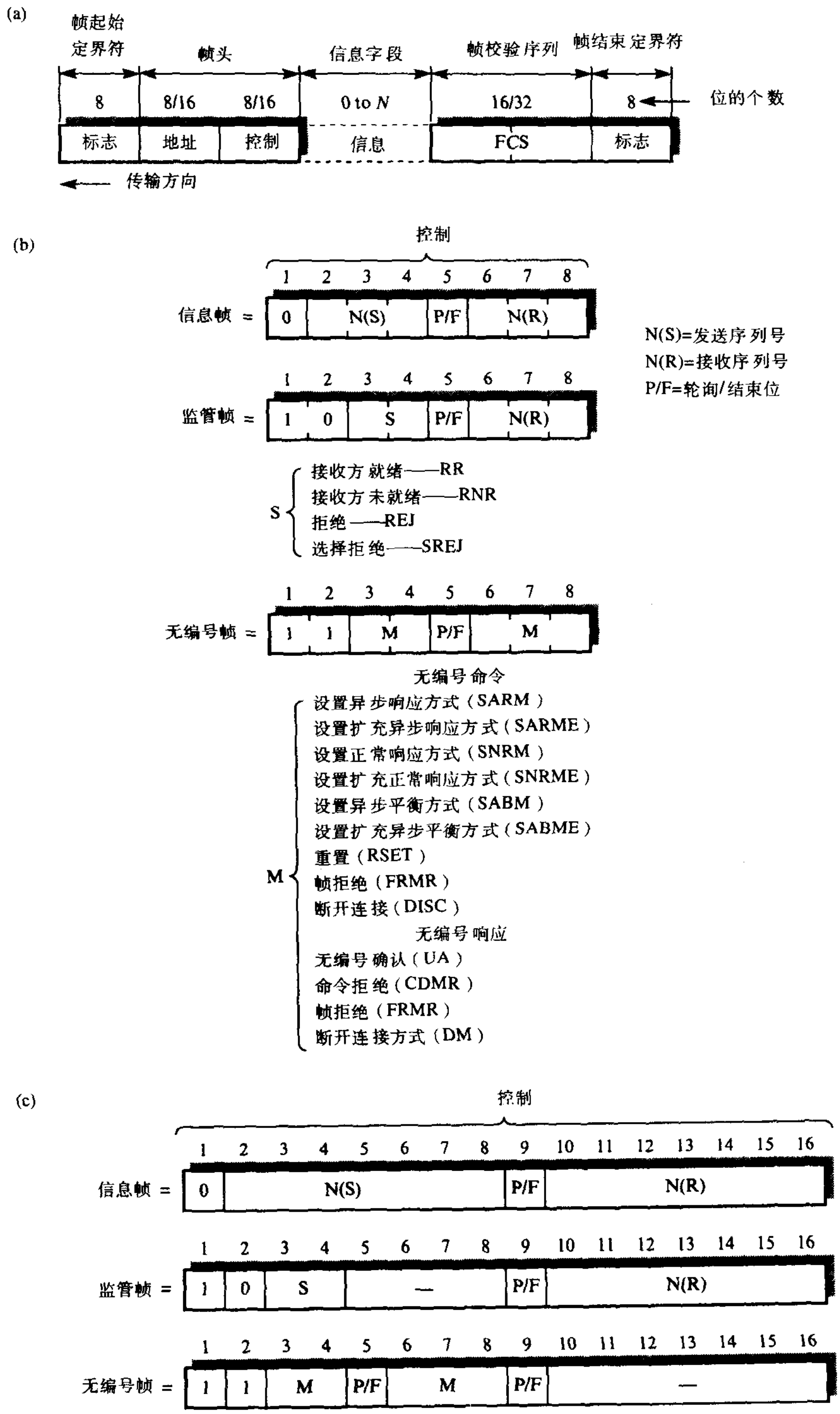


图5-11 HDLC帧格式与类型

(a) 标准/扩充帧格式 (b) 标准控制字段位定义 (c) 扩充控制字段位定义

P/F位称为**轮询/结束位**。主站发送任何类型的帧，称为**命令帧**；从属站发送的帧，称为

响应帧。当轮询/结束（P/F）位用于命令帧时，称为轮询位，该位置位表示接收方必须确认这个帧，接收方返回P/F位置位的一个适当响应帧确认这个命令帧，那时P/F位称为结束位。

每个N(S)与N(R)使用3位就是说序列号的范围是0～7，也就是说最大发送窗口可选作7。对于许多应用，这个数字已足够大。如长距离链路（例如卫星链路）或高比特率，如果想得到链路高效率的利用，则要求发送窗口更大。**扩充格式**使用7位（0～127），因此增加的最大发送窗口为127。

地址字段标识发送帧的从属站地址，点对点链路不需要地址字段。然而，多点链路地址字段或者是8位（正常方式），或者是8位的倍数（扩充方式）。在后一种情形下，8位组地址的最低有效位置0而最后的8位组地址的最高位置1。在这两种地址模式中，全1的地址用作对所有站点广播地址。

2. 帧类型

前面我们描述过HDLC协议的操作，它或许有助于列出某些帧类型以及概述它们的功能。使用三种不同类的帧，每一类中又有一些不同类型的帧，在图5-11(b)列出。

无编号帧用于链路管理。例如，SNRM帧与SABM帧两者用于主站与从属站之间建立逻辑链路并通知使用该操作方式的从属站。根据主站发送DISC帧清除逻辑链路。UA帧用作这一类帧中其他帧的确认。

虽然监管帧有4种不同格式，其中只有RR帧和RNR帧可用于NRM和ABM。这些帧用来表示从属站已准备好或未准备好接收来自主站的I帧响应。拒绝REJ帧与选择拒绝SREJ帧仅用于ABM，这种方式允许通过点对点链路同时进行双向通信。这两个帧用于向其他站指出发生了序列差错，即已收到的某个I帧包含失序的N(S)。SREJ帧用于选择重发策略，而REJ帧用于回
241 退N帧策略。

3. 协议操作

本节主要内容是描述HDLC协议的一些重要特征，而不是给出协议操作的全貌。下面描述链路管理与数据传送（包括差错与流量控制）两个基本功能。

（1）链路管理

在多站点链路上主站与从属站之间或点对点链路上两个站点之间，在发送任何信息（数据）前，必须在通信双方之间建立逻辑连接。这是通过交换两个**无编号帧来**实现的，如图5-12所示。

在多点链路中（见图5-12(a)），首先由主站发送SNRM帧，其中轮询位P置1并在地址字段置相应从属站地址。从属站则返回一个UA帧作为响应，其中置结束位F为1，并在地址段置本
242 站地址。由此可见建立规程使每站的序列变量初始化，这些变量用于差错与流量控制规程。最后，当所有数据传送完毕。主站发送DISC帧，从属站返回一个UA帧响应，置F位为1，清除链路。

建立点对点链路的方法与建立多点链路的方法相同，如图5-12(b)所示。如果选择ABM方式工作，则首先发送SABM帧。用这种方式，链路两端独立地发起I帧传送，由于每个站必须有主站与从属站的动作，所以它时常称为复合站点。在这种方式中，任何一个站既可发起建立链路，又可以清除链路。在图5-12(b)中，站*A*发起链路建立，而站*B*发起清除（逻辑）连接。在两个方向上帧的一次交换建立起链路。正如所见，地址字段用于表示命令帧（SABM/DISC）的方向及其相关响应。

如果接收方想要拒绝以任何一种方式建立命令，则可返回一个断开连接方式（DM）帧，作为初始方式设置帧（SNRM或SABM）的响应。DM帧表示响应站被逻辑断开连接。

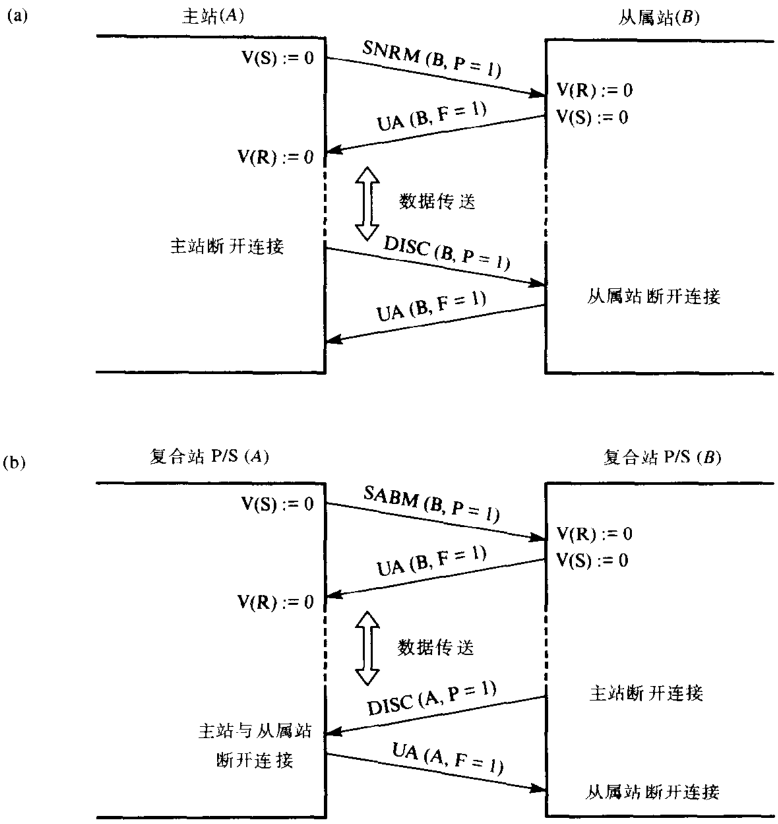


图5-12 链路管理规程

(a) 正常响应方式——多站链路 (b) 异步平衡方式——点对点链路

（2）数据传送

在NRM中，所有数据I帧都是在主站控制下传送。主站轮询一个从属站，通常将无编号轮询（UP）帧的P位置1。如果从属站没有数据发送，则返回一个F位为1的RR帧。如果有数据等待发送，则典型地以I帧序列形式发送数据，序列的最后一个I帧的F位置为1。

数据传送阶段的两个重要内容是差错控制与流量控制。差错控制采用连续RQ方案，或者选择重发策略或者回退N帧重传策略。而流量控制基于窗口机制。在第4章中，我们讨论这两种方案的基本操作，所以这里仅给出典型帧序列以说明不同帧类型的用法。

图5-13(a)说明基本确认与重传方法，这个实例仅使用RR帧并假设使用回退N帧策略。图中仅表示了单方向的I帧流，因而所有确认信息均采用确认监管帧返回。可以看出，链路的两端都保存着发送序列变量V(S)与接收序列变量V(R)。V(S)表示该站待发的下一个I帧序列号
N(S)，而V(R)则为该站希望接收的下一个I帧的序列号。 243

每一个RR（肯定确认）监管帧包含一个接收序列号N(R)，它确认至N(S)=[N(R)−1]为止发送的所有I帧已正确接收。同样的，图5-13(b)表示每一个REJ（否定确认）监管帧包含一个N(R)，表示N(R)接收到一个失序的I帧，发送方必须重发从N(S)=N(R)起始的I帧。

图5-13(a)表示接收到任何失序的I 帧都简单地丢弃。因此，接收到I帧(2, 0/P=1)（由于前面帧没有确认，P位为1）是失序的，该帧丢弃，接收方没有动作发生。在没有确认时，I(1)帧与I(2)帧相关的定时器到时，这两个帧重发。实例假设正确收到每一个帧并使用RR帧确认。

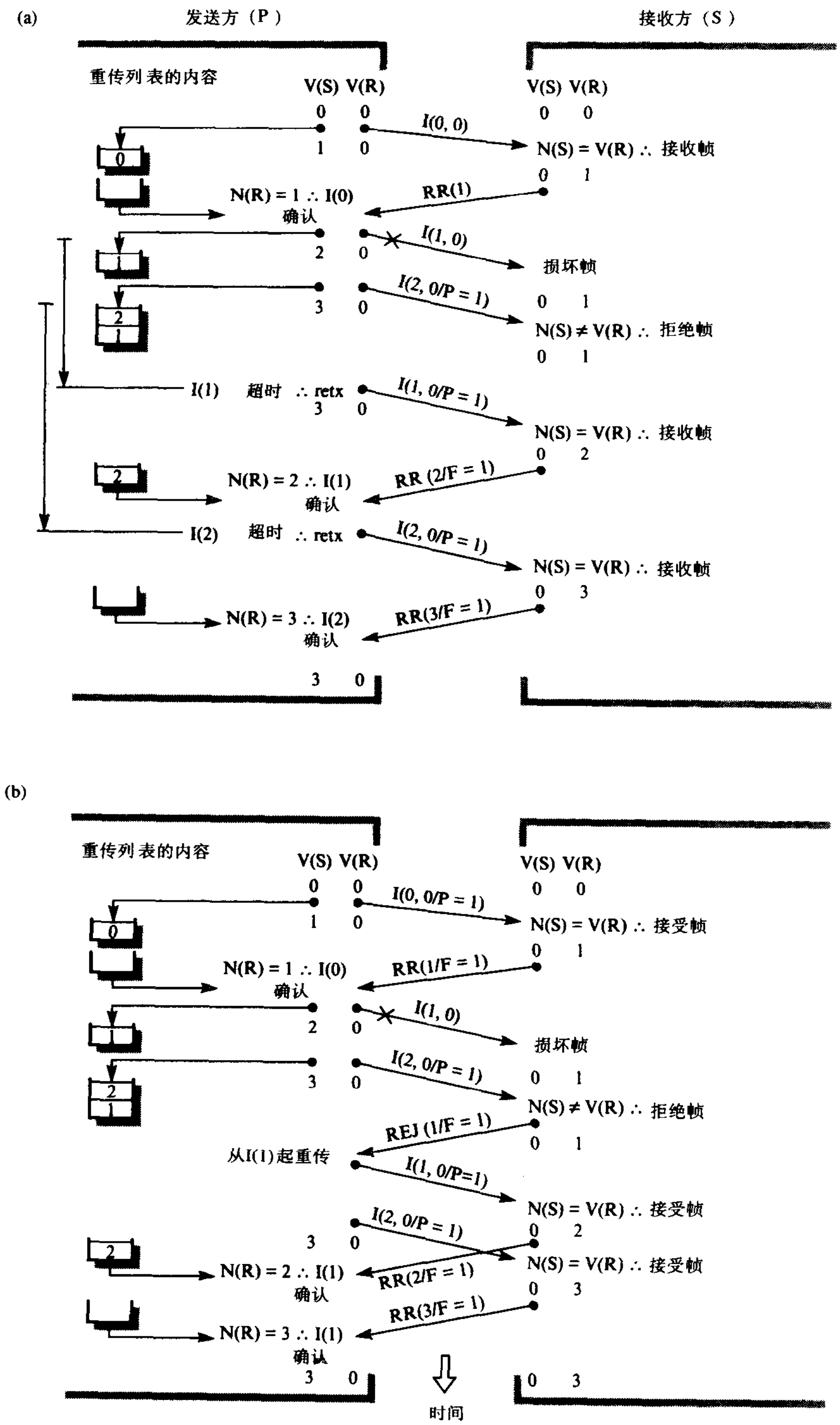


图5-13 确认帧的用法

(a) 仅有肯定确认（RR） (b) 否定确认（REJ）

在图5-13(b)中使用否定确认（REJ）帧，当接收方检测出I帧(2, 0/P=1)（即P=1的序列中

最后的帧）失序时，则返回F位为1的REJ帧。然后发送方重传I帧(1, 0)与I帧(2, 0)，在I帧(2, 0)中P位仍置1。接收方在最后一个RR帧中，F位置为1确认正确接收到每一个I帧。如果采用选择重发，则将接受I帧(2, 0/P=1)，并返回一个SREJ帧请求重发I帧(1, 0)。

图5-13中的帧序列是通过多站链路以NRM操作的典型信息传送过程。然而，对于ABM点对点链路，很可能出现双向I帧流。因而NRM中，与一个方向I帧流有关的确认信息可由反方向的I帧流捎带，见图5-14给出的实例。为了清晰起见，这里假设没有传输差错。

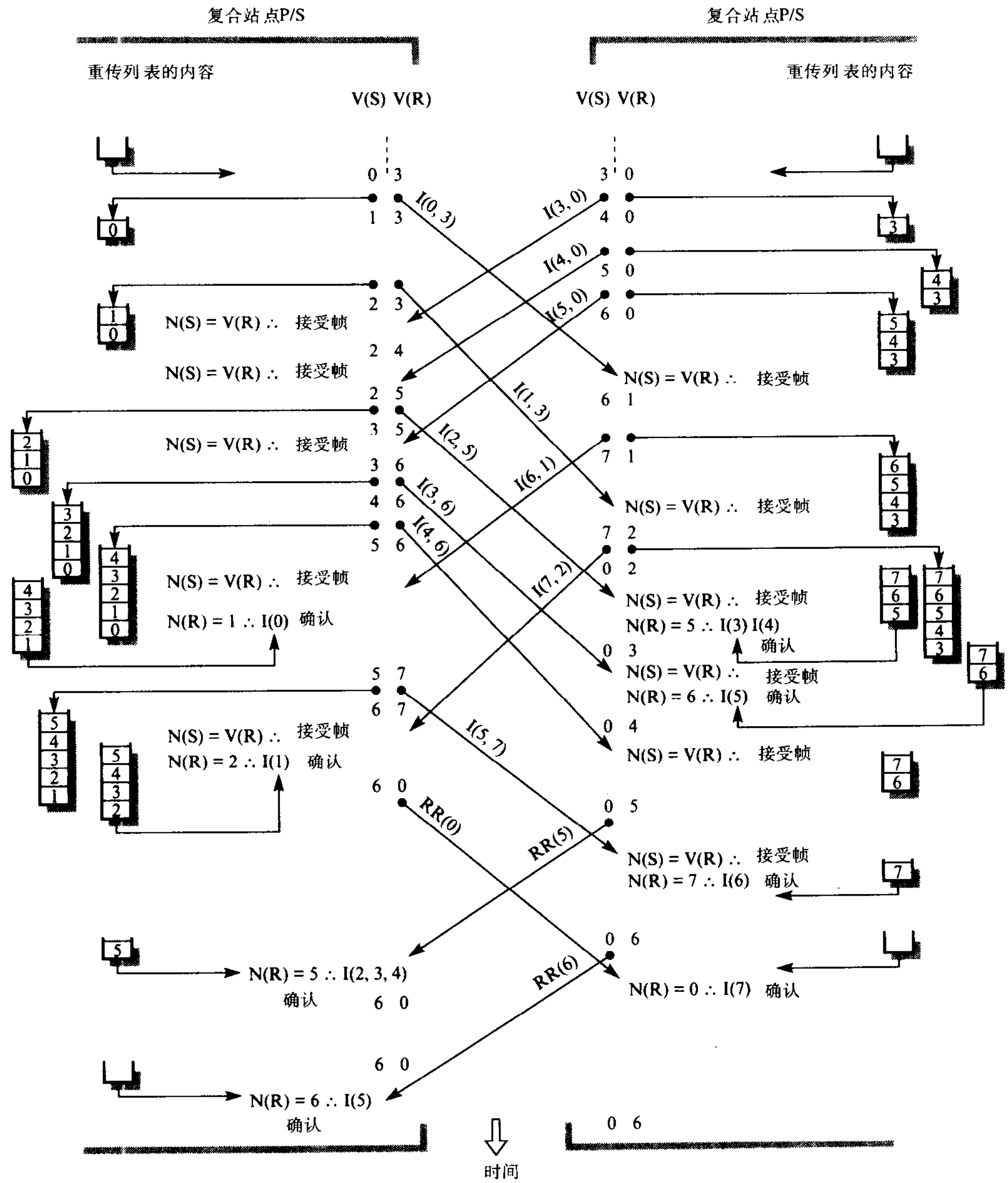


图5-14 捎带确认过程

每当接收到一个I帧，就阅读I帧的N(S)与N(R)。首先，将N(S)与接收方V(R)比较，如果

它们相等，则该帧按正确序列接收；如果不相等，则该帧丢弃，并返回REJ帧或SREJ帧。然后检查N(R)，用以确定重传列表中是否还有未被确认的帧。最后，当没有等待传输的I帧时，则用RR帧确认每个重传列表中待确认的帧。

当采用双向同时工作方式，链路工作在异步平衡方式（ABM）时，流量控制特别重要。对NRM，当主站处于暂时过载情况时，它可以简单地暂停轮询操作，借此消除过载。而当链路双方独立地操作时，我们必须采用另一种方案。HDLC中采用的流量控制方案与4.3.3节讨论的滑动窗口方案相同。

图5-13与图5-14所示的实例，我们看出发送序列号与接收序列号是模8增1，所以最大发送窗口K能够使用7。因此，任何时刻在重传列表等待确认的最大帧数为7。链路两端都保留有独立变量，这称为重传计数（RetxCount）。当逻辑链路刚建立时，初值为0，每次发送一个I帧，即放在重传列表中，重传计数增1。每当接收到一个肯定确认，并从重传列表移去一帧时，重
244~245 传计数减1。当重传计数到达K时，主站停止发送I帧，直到收到一个肯定确认才重新开始发送，这种确认既可以是独立RR监管帧，也可以由反向I帧流捎带确认。我们可以得出，当V(S)=最后接收到N(R)+K时，I帧停止发送。

246 注意，窗口机制仅在一个方向上控制I帧流，该机制不影响监管帧与无编号帧发送。因此，当操作窗口时，这些帧仍然发送。实例如图5-15所示。为了清晰起见，这里仅考虑一个方向的I帧流。

窗口机制应用就是说，所有进来的帧的序列号必须限定在某个范围内。每收到一个帧，
247 从属站必须检测它是否满足这一条件。如果不满足，则采取相应的动作。所以，每个收到的N(S)与N(R)必须满足下列条件：

1）V(R)小于或等于N(S)，N(S)小于V(R)+K；

2）V(S)大于N(R)，N(R)大于或等于V(S)–RetxCount（重发计数）。

如果N(S)等于N(R)，则一切正常，接收帧；如果N(S)不等于N(R)，但它仍在范围内，则帧已损坏，故返回REJ帧（回退N帧）或SREJ（选择重发），通知主站产生了失序差错，并从该帧开始重发，如图5-13所示。

如果N(S)或N(R)超出范围，则链路两端的序列号已变成不同步，因此链路必须重新初始化（重新建立）。这由从属站来实现，当检测到一个超出范围的序列号，则丢弃接收到的帧，并返回一个**帧拒绝**（FRMR）帧（ABM）或一个**命令拒绝**（CMDR）帧（NRM）给主站。主站丢弃重传列表中所有等待确认的帧并发送一个SABM/SNRM等待相应UA响应，以期重新建立链路。主站一收到这个响应，链路双方重置它们的序列与窗口变量，因而可以重新开始I帧流。事实上，这仅是链路重置的一种原因，还有其他原因，例如数据传送阶段收到未编号帧，诸如UA帧，表示主站和从属站已不同步，也需要重置。

刚刚列出的流量控制过程是由链路的主站方控制的，链路根据发送窗口控制I帧流量。此外，由于链路从属站方发生某些事件，需要从属站停止I帧流量。例如，回退N帧重发策略，接收窗口为1，接收方保证有足够的存储缓冲区。如果使用选择重发，则从属站可能用完所有缓冲区以存储新帧。因而，当缓冲区即将占满时，从属站给主站返回一个RNR监管帧，通知主站停止发送I帧。当然，这时不影响确认帧的发送。当满缓冲区数量低于某一预置限制时，从属站向主站返回一个带有N(R)的RR帧，表示从该帧开始又可重新传输。

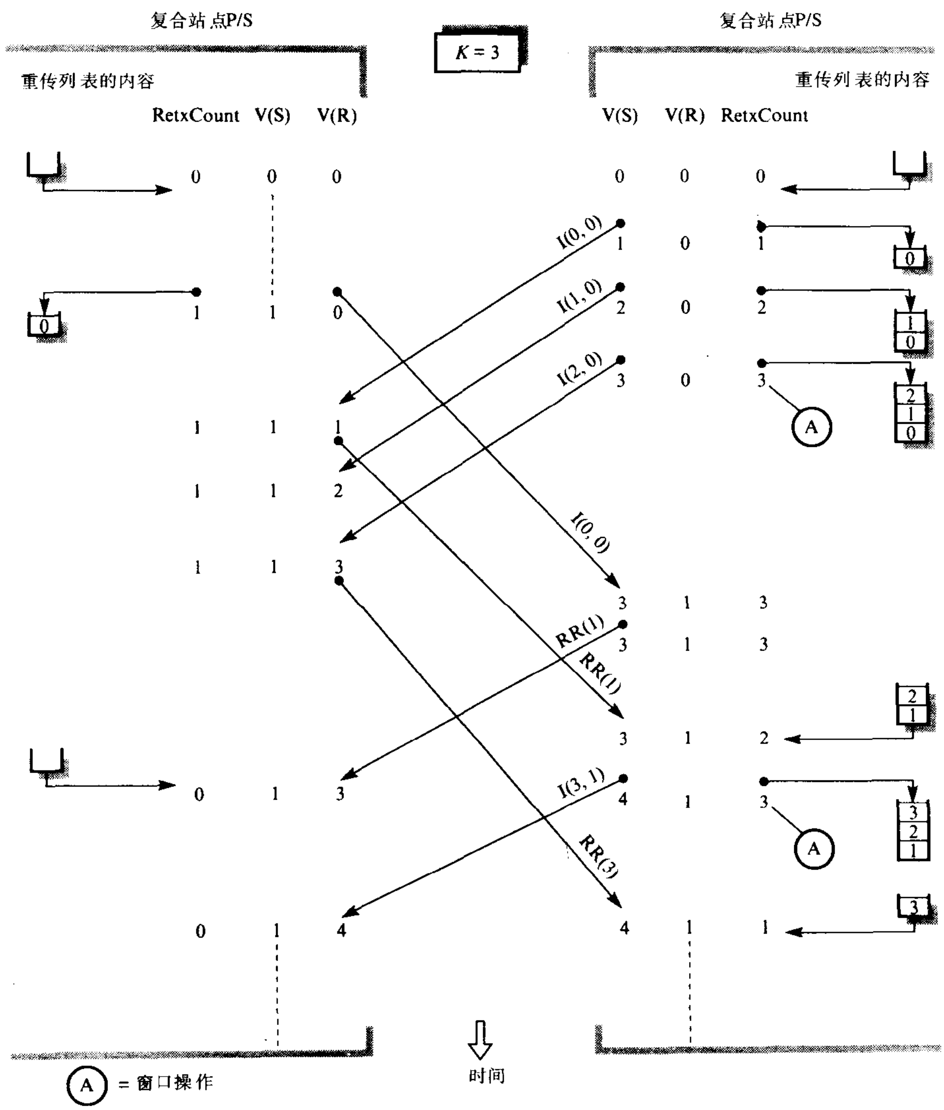


图5-15　窗口流量控制过程

4. 用户接口

现在我们建立前面图5-1所示的链路层用户服务与HDLC各协议信息单元之间关系，如图5-16所示。

一接收到来自服务用户的最初L_CONNECT.request（连接请求）原语，主叫站的链路层协议实体首先向被叫站的链路层协议实体发送SNRM/SABM监管帧。后者一收到该帧，便产生并传递一个L_CONNECT.indication（连接指示）原语给被叫服务用户。此外，它产生一个UA帧并将其返回给主叫站。当主叫站接收到该UA帧，便产生一个L_CONNECT.confirm（连接证实）原语给主叫服务用户。这时，就可以利用L_DATA服务原语实现用户数据传送。最后，当所有数据（信息）交换完毕后，用DISC与UA监管帧断开链路。 248

各种服务原语与HDLC相关的帧类型（协议数据单元）汇总于图5-17(a)中。实际上，除图中表示外，还有许多与HDLC有关的无编号帧，但正如以前提到的，这里并不给出HDLC的完

整描述，而是选择一些重点操作进行介绍。为了加强理解，给出HDLC的（简明）状态迁移图，如图5-17(b)所示。每个弧线上的第一项是引起迁移的入事件，第二项是引起的动作。注意，状态迁移图仅表示协议实体的正确操作，通常伴有事件—状态表与/或伪码的形式，以给出更
249 完整的定义。

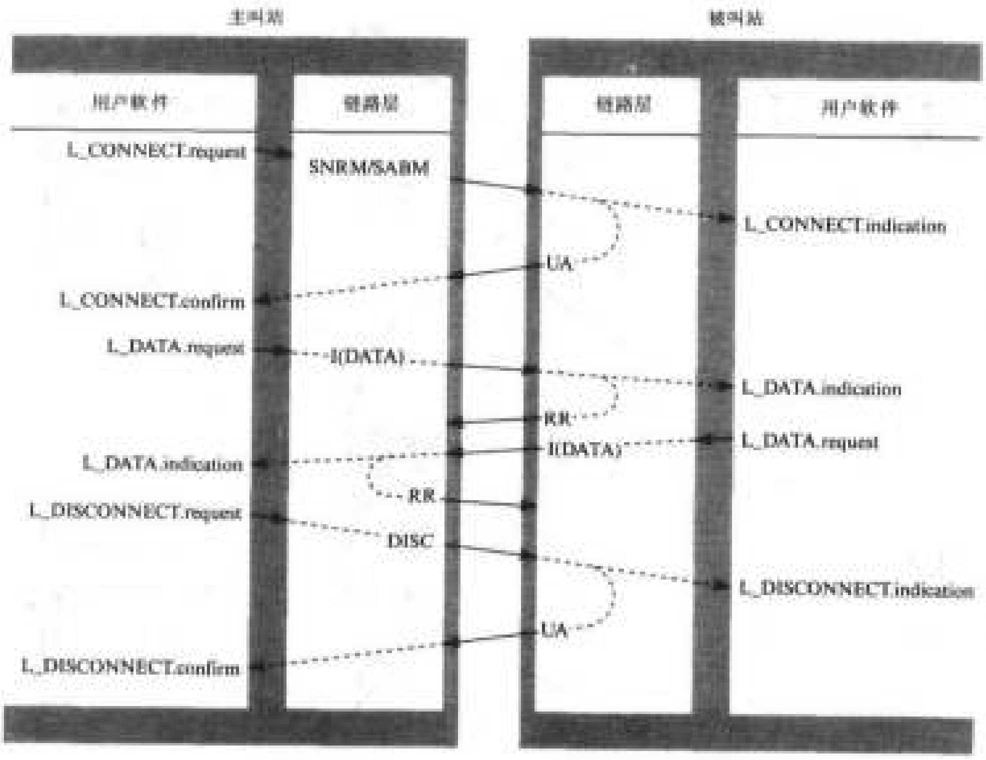


图5-16　用户/链路层相互作用

5.3.2　链路访问规程版本B

链路访问规程版本B（LAPB）是HDLC的一个子集，用于连接计算机到公用（或专用）分组交换网的一个点对点双工数据链路上I帧的传送控制。这样的网络一般称为X.25网络，将在8.2节更详细讨论。事实上，LAPB是称为**链路访问规程版本A**（LAPA）的早期子集的扩充
250 版本。

表　5-2

类型	LAP 命令	LAP 响应	LAPB 命令	LAPB 响应
监管	RR	RR	RR	RR
		RNR	RNR	RNR
		REJ	REJ	REJ
无编号	SARM	UA	SABM	UA
	DISC	LMDR	DISC	DM
信息	I		I	FRMR

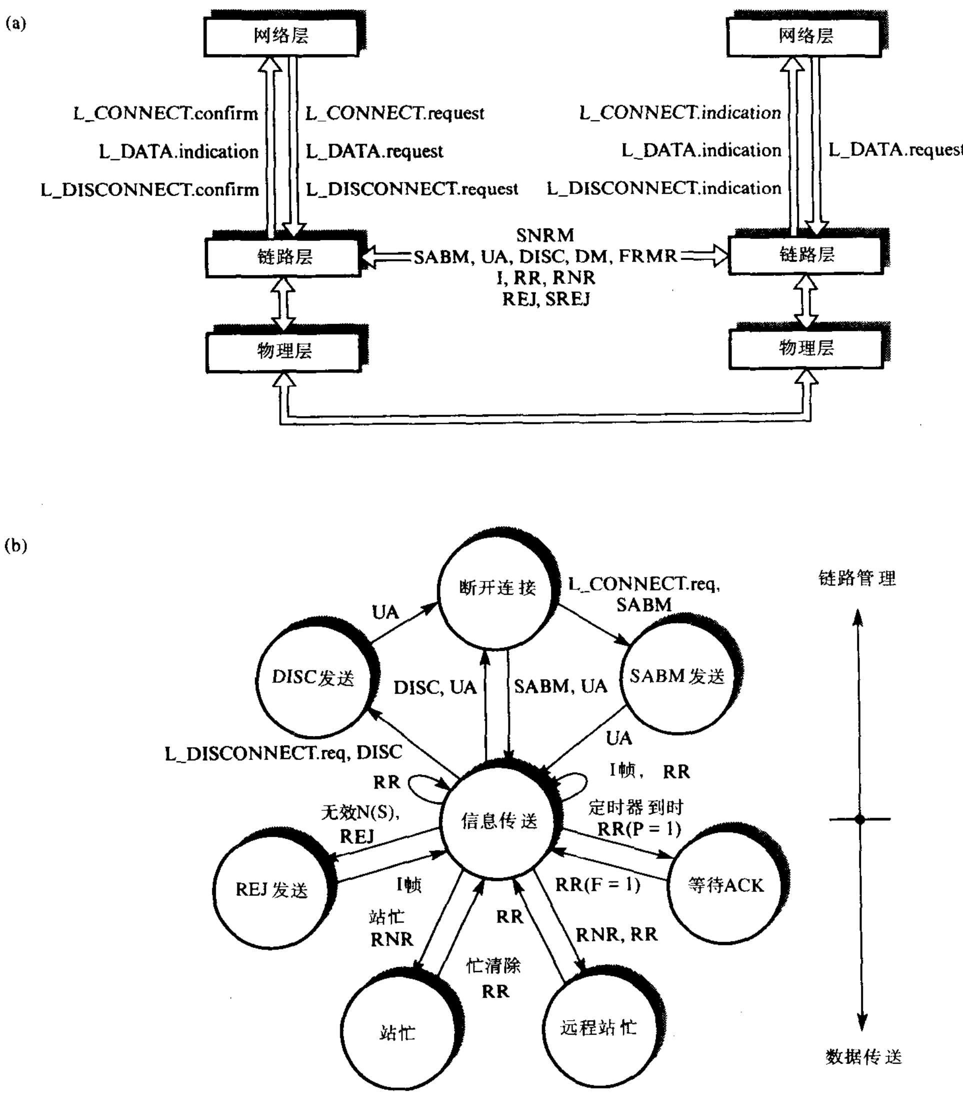


图5-17 LAPB汇总

(a) 服务原语 (b) 状态迁移图（ABM）

LAPB的适用环境如图5-2(c)所示，其中计算机是DTE，分组交换中心是数据电路端接设备（DCE）。LAPB用于通过本地DTE-DCE接口控制信息帧的传送。LAPB被认为本地有效。

LAPB采用异步平衡方式（ABM）操作，DTE与DCE作为复合站点，所有I帧都作为命令帧处理。早期LAP协议采用异步响应方式，REJ帧和RNR帧不作为命令帧使用。LAP与LAPB所用帧汇总于表5-2中。RR帧和REJ帧用于差错控制，而RNR帧用于流量控制。这些帧不支持选择重发（SREJ）。前面有关HDLC操作的图中所示的帧序列实例可直接用于LAPB。正如我们已指出，发送P位为1的信息（命令）帧引起接收站点返回一个F位为1的监管帧响应。任何一个站都可以建立链路。为了区别两个站，DTE与DCE的地址使用方法如表5-3所示。如果不是逻辑操作，一个DTE接收到一个建立请求帧（SABM/SABME），它必须用一个DM帧回复。

LAPB的P/F位的用法概要在表5-4中给出。

表 5-3

方 向	地 址	
	命 令	响 应
DTE → DCE	01Hex(B)	03Hex(A)
DCE → DTE	03Hex(A)	01Hex(B)

回忆在通常SABM方式中，控制字段使用单个8位组，发送与接收序列号是3位（8个序列号），允许的最大发送窗口为7。如果选用扩充方式（SABME），控制字段是两个8位组，因而发送与接
251 收序列号扩充到7位（128个序列号），同样也允许较大的窗口。这用于长距离高比特率的链路。

表 5-4

P=1发送命令帧	F=1返回响应帧
SABM/SABME	UA/DM
I帧	RR, REJ, RNR
	FRMR
RR, REJ, RNR	RR, REJ, RNR, FRMR
DISC	UA/DM

现在集成电路可将LAPB实现程序固化在存储器中。虽然它们仅实现LAPB协议，而不是全部X.25协议组，但这些时常称为X.25电路。这些电路的存在大大增进了LAPB用于计算机到计算机的通信应用。

5.3.3 多链路规程

我们已描述将HDLC用于经过单个双工链路控制信息帧的传送。因此，HDLC称为**单链路规程**。但在有些情况下，单条链路的有效吞吐量（或可靠性）不够满足应用要求，我们必须使用多条（物理）链路。考虑到这一点，扩充LAPB定义为**多链路规程**（MLP）。

图5-18(a)表示经过每条物理链路的帧传送是由刚才描述的方法用独立单链路规程控制。单个MLP操作上述链路规程集合，并简单地处理它们作为有效传送用户信息的缓冲池。这就是说用户软件不知道正在使用多条物理链路，如以前以单条（逻辑）链路接口出现。

作为简述，MLP简单地看作在单条链路规程组上传送用户帧的链路缓冲池。它用自己的一组序列号操作，差错控制与流量控制方法被每个SLP彼此独立地操作。因此，如果一个SLP成为不可操作的，则MLP以正常方法发起帧重传，但使用有效链路（SLP）的减少组。

为了实现这个方案，MLP在每个帧的头部增加一个附加控制字段，它在传递帧到SLP前用作传送接收，称为**多路控制**（MLC）字段，它实际上对于SLP是透明的。SLP处理复合的MLC
252 与作为信息字段的帧内容以及着手增加它自己的地址（A）与控制（C）字段，如图5-18(b)所示。MLP差错控制与流量控制机制本质上与LAPB所用的相同。

MLC字段由两个8位组构成，它有12位序列号，提供4096个序列号（0～4095）因此最大窗口为4095，允许所用的有效链路个数可以在更高数据速率操作。例如，两个X.25分组交换网连在一起是个实例，我们将在第8章阐述。

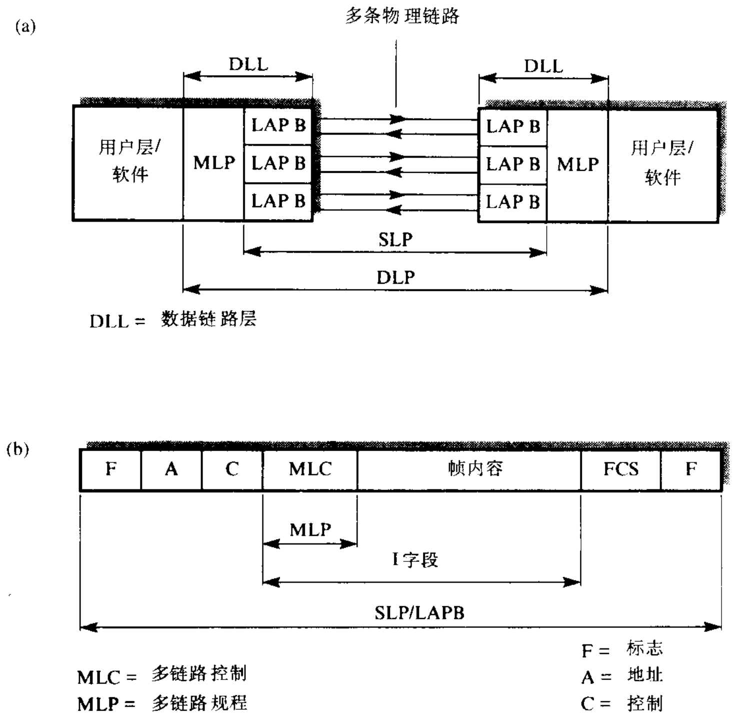


图5-18　多链路规程

(a) 关于数据链路层的位置　(b) 帧格式

5.3.4 调制解调链路访问规程

调制解调链路访问规程（LAPM）是纠错调制解调器所用的协议，如V.32调制解调器，这些调制解调器接受来自DTE的异步数据（开始—停止），用面向位同步传输方式以帧形式发送数据，还有基于HDLC的纠错协议，LAPM的适用范围如图5-19(a)所示。

每个调制解调器有两个功能单元：用户（DTE）**接口部分**（UIP）与**纠错部分**（ELP）。LAPM协议与ECP有关，而UIP牵涉到经过本地V.24接口传送单个字符/字节，以及通过这个接口解释流量控制信号。 253

用一组规定的服务原语，UIP与ECP通信，时间序列图在图5-19(b)给出。也给出了LAP从协议机使用的不同HDLC帧类型实现的不同服务。

在建立（逻辑）链路前，发起与响应ECP必须一致同意协议所用的操作参数。这些参数包括I帧中最大的8位组个数，确认定时器的设置，最大重传的次数以及窗口的大小。每个都有相关的默认值，但如果不使用它们，发起的UIP必须发出一个带有要求操作参数值的L_SETPARM.request原语。两个ECP交换两个特殊的无编号帧（称为**交换识别**（XID）），一个作为命令，另一个作为响应协商该值。 254

一旦操作参数已接受，UIP发出L_ESTABLISH.request原语，那么链路可以建立。这是依次由ECP发送一个SABM（通常的）或SABME（扩充的）监管帧的结果。接收ECP然后发出L_ESTABLISH.indication原语到本地UIP，接收到响应原语，接收ECP返回一个UA帧。接收到这个帧，发起的ECP发出一个证实原语，与此同时链路建立。然后用L_DATA服务开始数据传送。

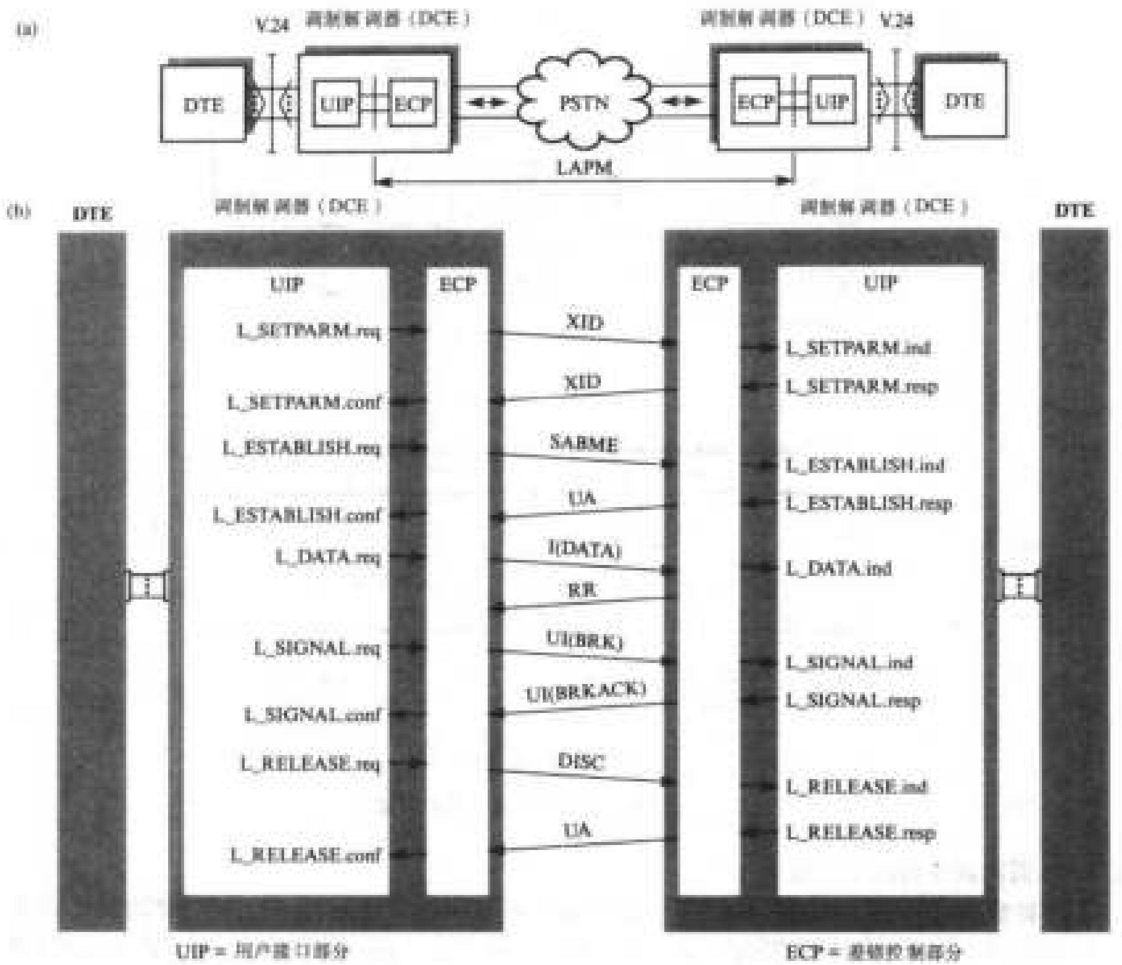


图5-19 LAPM

(a) 操作范围 (b) 用户服务原语与对应帧类型

典型地，UIP首先将V.24接口上收到的字符或字节装配成数据块，然后用L_DATA.request原语传递完整数据块到ECP。ECP把数据封装在I帧的信息字段中作为8位组串，用通常HDLC协议的纠错方法传送I帧。然后接收到的ECP传递数据块（可能纠正差错）到本地UIP，通过本地V.24接口每次一个字符（字节）地传送它。

如果在数据传送阶段，检测到流量控制（破坏）条件（例如，接收到一个X-OFF字符或者DTR线路失效），那么UIP停止输出数据到本地DTE，并立即发出一个L_SIGNAL.request原语到本地ECP。然后，本地ECP发送一个BRK（失效）消息——称为无编号信息（UI）帧的特殊帧通知远方的ECP（暂时）停止发送任何数据。按名称所示，这个帧不含序列号，由于它越过任何差错/流量控制机制。然后接收ECP发出一个L_SIGNAL.indication原语到本地UIP，再返回另一个UI帧中的BRKACK消息确认失效信息收到。然后UIP通过自己的V.24接口发起同样流量控制信号。

最后，所有数据传送后，当发起UIP发送L_RELEASE.request原语时，链路清除。这时证实服务与相关的LAPM帧是DISC与UA。

5.3.5 D信道链路访问规程

D信道链路访问规程（LAPD）是综合业务数字网（ISDN）采用的HDLC子集。它定义控制与信令（呼叫建立）信道相关的I帧流，信令信道称为**D信道**。LAPD也用于控制带有称为**帧中继**服务的用户信道上I帧流的一种略微扩充形式。有关ISDN留到第8章讨论广域网（WAN）时详细论述。这一节，我们将简单讨论LAPD的基本操作以及它与HDLC的关系。

LAPD使用两种服务类型，两组服务原语的时序图在图5-20表示。从中可以看出，LAPD提供无确认（最佳尝试）与确认（面向连接）两种信息传送方式。代替模拟PSTN的ISDN基 255
本上是一个电路交换网，就是说在传送任何用户数据前，必须建立电路。建立电路由独立的信令信道（D信道）完成，它的协议组是LAPD的一部分。

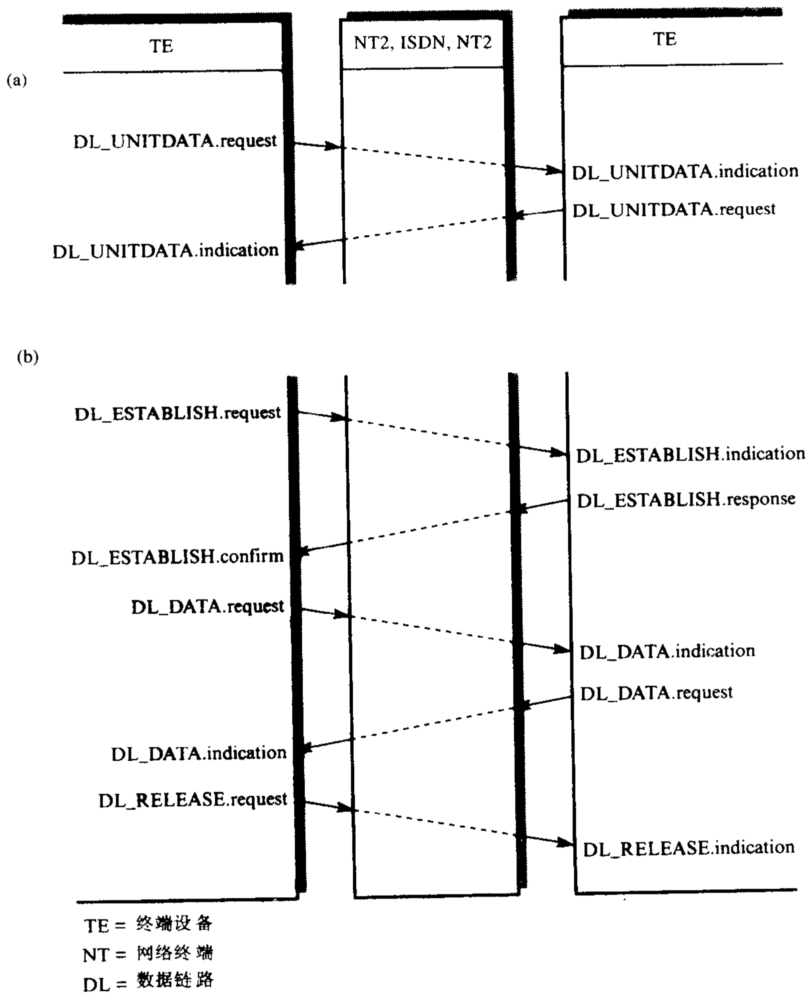


图5-20 LAPD用户服务原语

(a) 无连接 (b) 面向连接

面向连接服务用于用户设备（电话或DTE）与本地交换机间传送呼叫建立消息，相应协议包含差错控制。无连接服务用于传送管理消息，而相应协议采用最佳尝试与无确认方法。

我们将在第8章中看到，在客户机构与本地ISDN交换中心之间可有8个终端（电话、DTE

或它们两者复合）共享一条基本访问电路（D信道）。然而，向一个指定终端设备发送所有
256 （高层）呼叫建立消息使用LAPD的地址段。除掉LAPD没有主站以及连接终端设备的物理总线结构允许每个终端直接访问总线外，它与NRM方式所用的寻址机制原理相似，LAPD的帧的一般结构如图5-21所示。

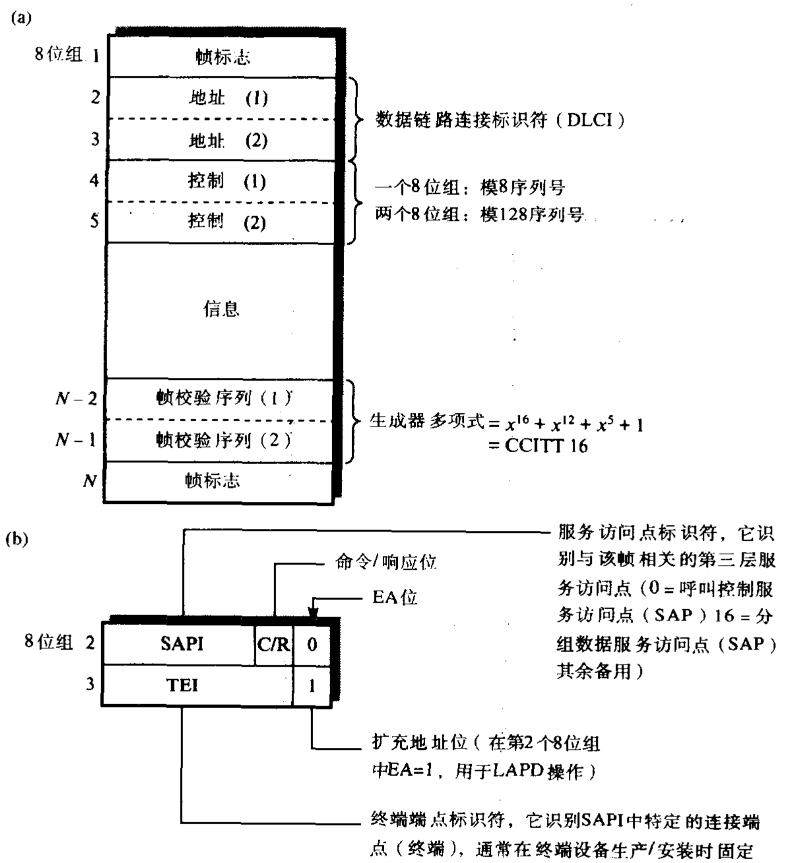


图5-21 LAPD

(a) 帧格式 (b) 地址字段用法

两个8位组用作地址字段后，它包含两个子地址：**服务访问点标识符**（SAPI）与**终端设备端点标识符**（TEI）。SAPI识别与该终端相关的服务类（语音、数据与语音及数据），然后TEI惟一识别类中不同终端。全部为1作为广播地址，允许向类中所有终端发送消息。例如，这可用于允许所有电话接收一个进入呼叫建立请求信息。

LAPD的各种控制字段格式（8位组4与5）在图5-22中概括，它也表示哪些帧作为命令帧发送与哪些帧作为响应帧发送。

257 LAPD如同LAPM，使用称为**无编号信息**（UI）的附加无编号帧。它用于LAPD无连接服务。由于这种服务（最佳尝试）没有差错控制，发送既没有N(S)也没有N(R)只有单个控制字段的所有信息。而这种帧有FCS字段，如果失败，该帧丢弃。一般，使用这种服务，高（用户）层必须检测出丢弃帧（例如，由于缺少适当响应（在UI帧中）），另外寻求解决办法。在5.3.6节中，将看到它用于局域网。

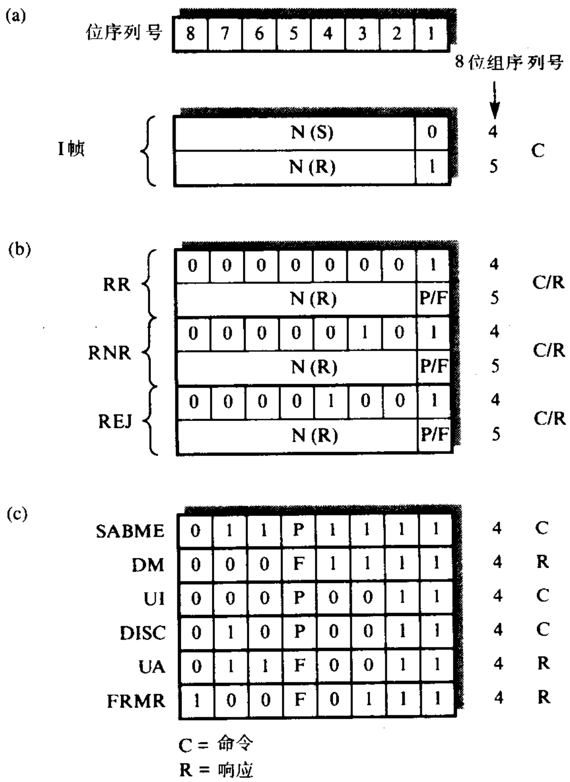


图5-22 LAPD控制字段位定义：

(a) 信息帧 (b) 监管帧 (c) 无编号帧

LAPD服务的定义与协议规范说明分别由ITU-T建议I.440与I.441规定，它们与建议Q.920与Q.921相同。

5.3.6 逻辑链路控制

逻辑链路控制（LLC）是HDLC用于局域网的派生。在第6章中，详述两种基本类型局域网的拓扑结构（总线与环型），以及如图5-2所示的数据链路协议DLP的领域（LLC）。

两种拓扑都利用共享传输介质（总线或环型）实施所有帧传输。作为多点网络，我们需 258
要控制发送帧的次序。不像多点网络，不存在主机，分布式算法通过所有连接的DTE（工作站、服务器等）保证使用传输介质。对于局域网，数据链路层由两个子层组成：**介质访问控制**（MAC）**子层**与LLC子层，前者执行分布的访问控制算法。不同MAC子层的操作在第6章讲述局域网时描述，这一节考虑LLC子层的操作。注意，局域网由于网络自身没有交换中心，则LLC（DLP）层对等操作，即在两个通信DTE中的LLC子层之间操作。

1. *用户服务*

LLC层提供两类用户服务：**无确认的无连接服务**与面向连接服务。前者以最小的协议开销向用户提供服务数据单元的传送。典型地，在高层协议层已提供差错恢复和排序功能的场合，使用这类服务更加适合，不需要在LLC层重复。面向连接服务允许用户建立链路层逻辑连接，再启动服务数据单元传送，如果需要，对通过已建立连接的数据流执行差错恢复和排序。

在某些实时LAN系统中，如化工厂用于连接分布式的计算机仪表设备的生产过程控制应用，建立逻辑连接所需的时延不可接受，也不能接受无确认的数据接收。尽管，经常需要对

传送的数据项的正确接收进行确认，所以基本的无确认无连接服务是不可接受的。为此提供一种称为**确认的无连接服务**。类似地，无连接服务类中还提供一种服务。这种服务无需先建立连接就允许向远程用户请求数据项，这称为**获得应答服务**。

这两类服务的各种原语及时序图，如图5-23所示。每个原语都带有源（本地）地址、目标（远程）地址等相应参数的说明。所有原语中出现的目标址与源地址规定最小网络物理地址。通常这两个地址包括物理地址与本地服务访问点标识符（LLC-SAP）。在第6章中，进一步考察LLC-SAP。

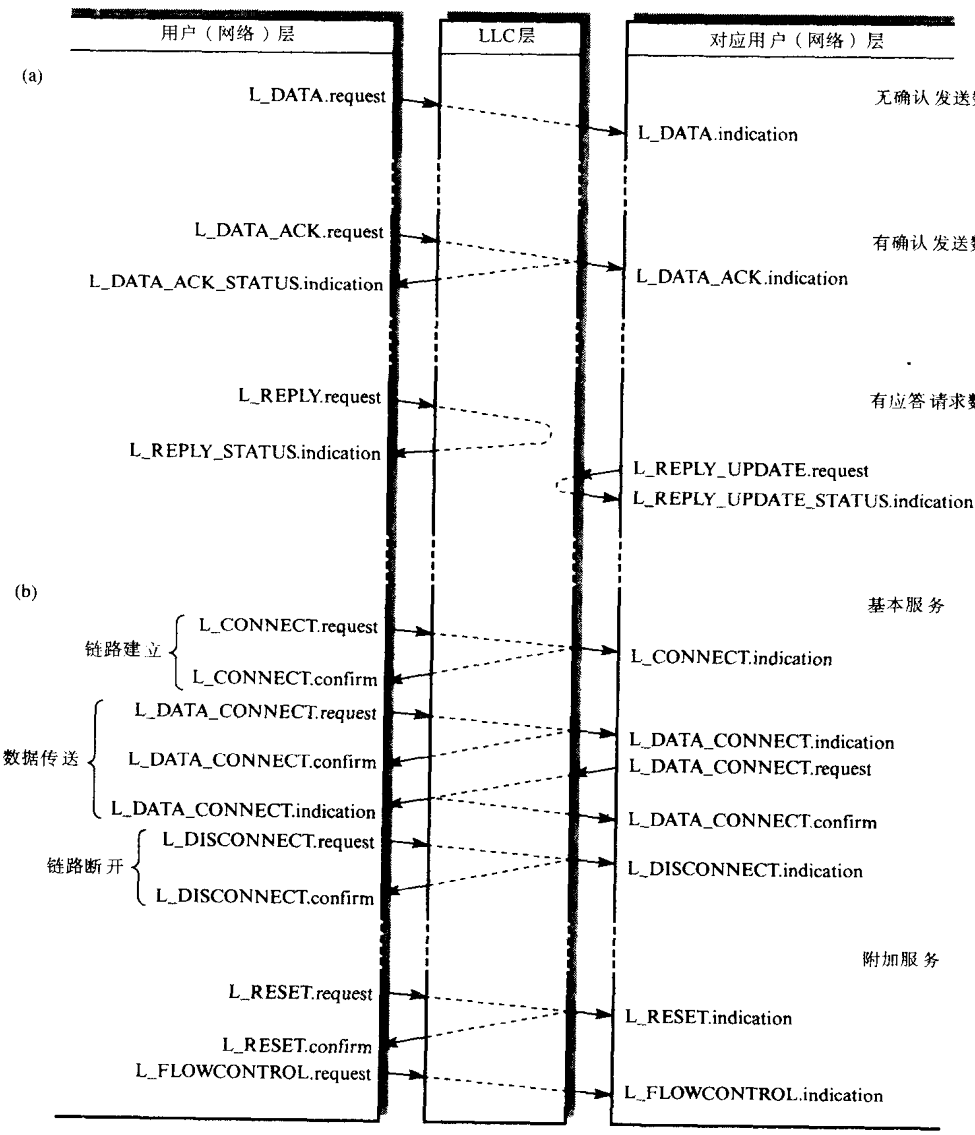


图5-23 LLC用户服务原语

(a) 无连接 (b) 面向连接

对于无确认的无连接服务，LLC协议实体接收到数据传送请求原语（L_DATA.request）
259 后，利用MAC子层发送数据，但无法证实传送是否成功。然而，对于确认的无连接服务，通

过L_DATA_ACKNOWLEDGE.STATUS.indication原语通知用户：给远程用户传递的L_DATA_ACKNOWLEDGE.indication原语是成功还是失败。

获得应答服务的不同原语能使用户：

- 采用L_REPLY.request/indication原语向远程LLC实体请求信息缓冲器内容。 260
- 采用L_REPLY_UPDATE.request和L_REPLY_UPDATE_STATUS. indication原语，修改本地LLC实体的信息缓冲器的内容。

对于面向连接服务，在传送数据之前，使用L_CONNECT原语建立逻辑连接。通过这个连接完成数据传送之后，必须使用L_DISCONNECT原语断开连接。在数据传送阶段，由远程LLC实体确认正确接收到每个数据单元，并由本地实体变换为L_DATA_CONNECT.confirm原语，传递给用户。

RESET与FLOWCONTROL服务原语能使用户通过已建立的连接控制服务数据单元流。RESET服务原语有一个无效的行动，因为它导致删除了未确认的数据，所以用于网络层协议实体弄乱了传送的数据单元顺序的情况。

两个流量控制原语仅在本地有意义：L_FLOWCONTROL.request原语规定了用户准备从本地LLC协议实体接受的数据量，而L_FLOWCONTROL.indication原语规定了LLC协议实体准备从用户接受的数据量两者都与指定的连接相关。如果规定量为0，则数据流停止；如果数据量是无穷，连接上没有应用流量控制。按每次请求动态地更新数据流量。

2. 协议操作

LLC的帧格式如图5-24 (a)所示。源地址与目标地址字段只指向LLC服务访问点，格式不包含网络地址，也没有FCS字段。将完整的LLC帧（包括帧和网络物理地址参数），以原语形式传递到MAC子层，由MAC子层处理网络寻址并执行差错检测功能。所以ISO参考模型的内容，可认为链路层等价于LLC层与部分MAC子层的组合。

每个帧的控制字段是一个8位组。它规定帧的类型以及差错控制与次序控制的发送序列号和接收序列号。其中字段中每位的用途如图5-24 (b)所示。

LLC协议实体支持两类操作：类型1支持无确认的无连接服务，类型2支持面向连接服务。除了组帧与差错检测功能由MAC子层提供，类型2实际上类似于HDLC协议。

类型2的数据链路控制功能，如图5-24 (c)所示。LLC协议和HDLC之间的最大差别是存在 261
无确认的无连接服务（类型1）。类型1支持表5-5中所列的一组命令与响应。

表5-5

命　令	响　应
UI（无序号信息）	–
XID（交换标识符）	XID
TEST（测试）	TEST

UI命令帧用于发送数据（信息）块给一个或多个LLC。由于类型1操作没有确认和序列控制，因此UI帧不包括N(S)和N(R)字段，也没有对UI帧的响应。

XID（交换标识符）与TEST（测试）命令帧是可选的。如果发送它们，则要求接收的LLC给以响应，这两个命令如下：

- 具有组地址的XID 命令用于确定组的现有成员。组的每个成员（返回一个专门对应于原始LLC实体的XID响应帧）作为对命令的响应。

262 • 某个LLC实体可使用具有广播（全局）目标地址的XID 命令，向网络宣布它的存在。

• TEST命令为每个LLC到LLC传输路径提供回路测试设施。

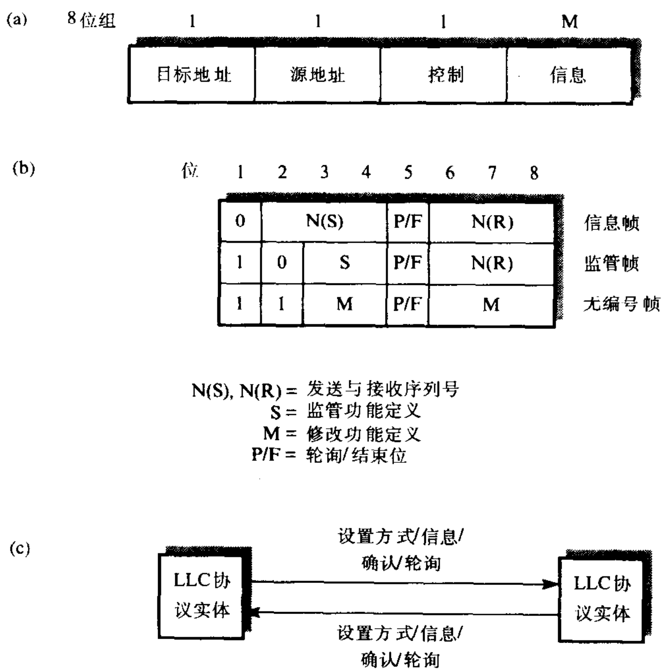


图5-24　LLC协议概要

(a) 帧格式　(b) 控制字段位定义　(c) 数据链路控制功能（类型2）

3. MAC服务

MAC子层为LLC层规定一组标准的用户服务，这组服务与MAC层的操作方式无关，LLC层使用这些服务发送LLC帧到相应的LLC层。支持的用户服务原语如下：

• MA_UNITDATA. request

• MA_UNITDATA. indication

• MA_UNITDATA. confirmation

时序图说明这些原语的用法，如图5-25所示。对于某些局域网，证实原语表示数据请求是否已成功地发送（(a)部分），而对于另一些局域网，它表示数据请求是否已成功递交（(b) 部分）。

每个服务原语都有相应参数，MA_UNITDATA.request原语包括请求目标地址（可以是单个地址、组地址或广播地址），服务数据单元（包含LLC帧）以及与帧相关的服务类。当采用优先级别介质接入控制协议时，某类局域网是最低级。

MA_UNITDATA.confirm原语包含一个指示MA_UNITDATA.request原语发送是否成功的参数。正如图5-25所见，证实原语是由本地MAC实体产生的，不是远程的LLC层。它包含一个指示MA_UNITDATA.request原语是否成功的参数。如果是成功的，说明MAC协议实体成功将服务数据单元传送到网络上；如果不成功，参数说明为什么传送失败。我们在第6章讨论

263 这个问题以及其他与MAC层有关的问题。

图5-26汇总了LLC层与MAC层的各种服务，列出两个LLC协议实体之间交换的各种LLC帧类型。正如我们所见，除了UI、XID与TEST外，其他帧类型与HDLC相同。

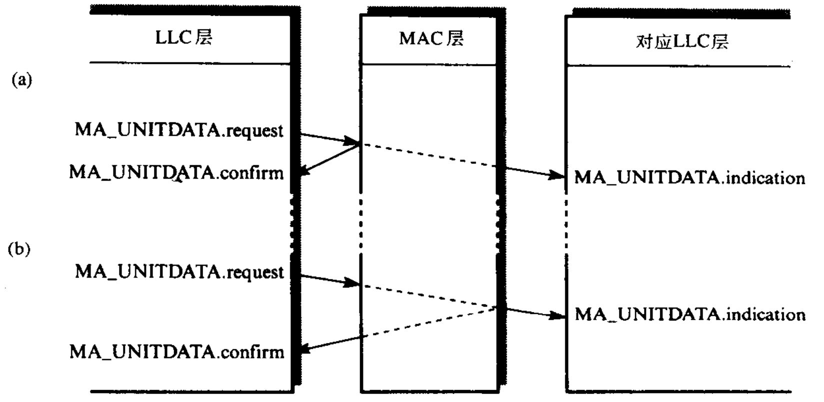


图5-25 MAC子层用户服务原语
(a) 本地证实 (b) 远程证实

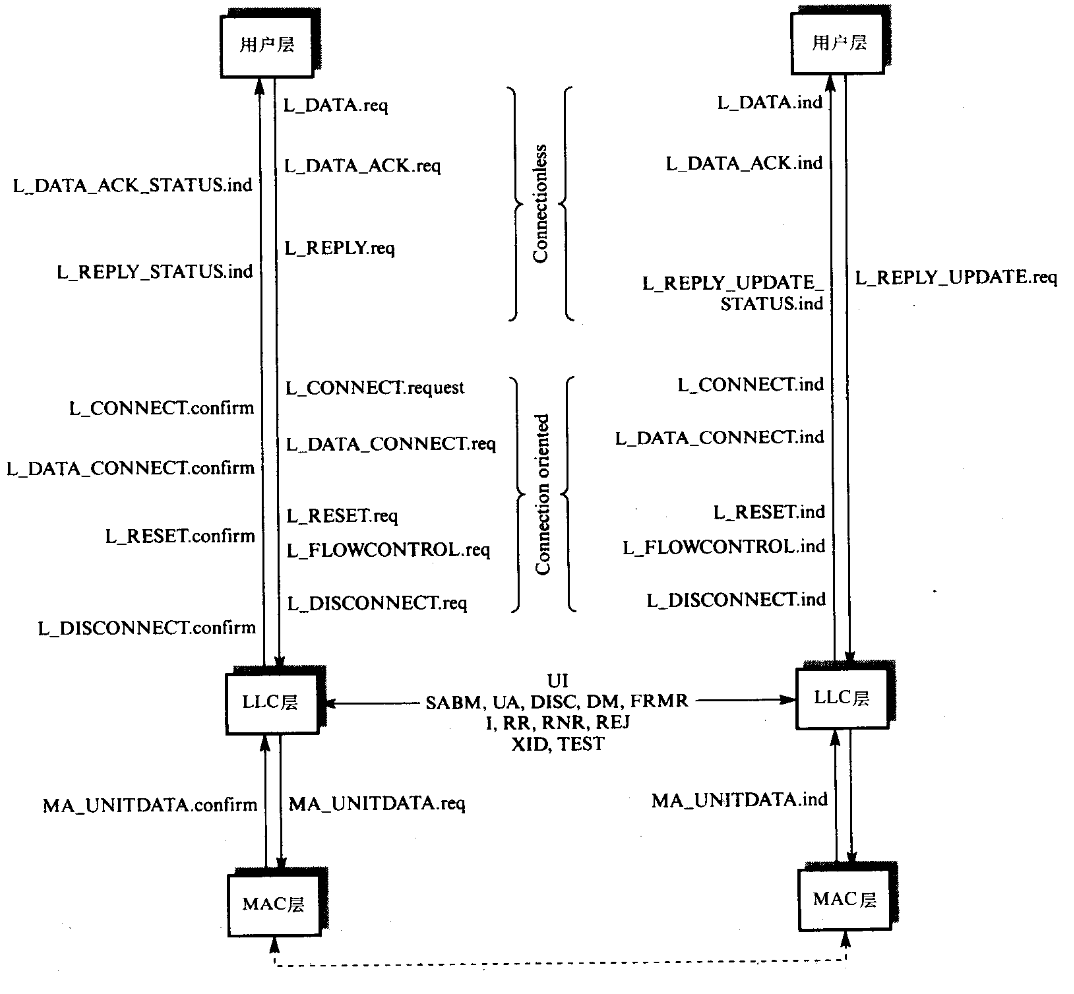


图5-26 LLC子层汇总

习题

5.1 解释与数据链路协议相关的下列术语的含义：
(a) 面向字符
(b) 面向位
(c) 组帧与数据透明性
(d) 轮询—选择
(e) 主站与从属站

5.2 画出时序图描述数据链路控制层的相关用户服务原语，假定以下列操作方式：
(a) 无连接（最佳尝试）方式
264 (b) 面向连接（可靠）方式

5.3 借助草图，在下列应用环境下，概述数据链路协议的操作范围：
(a) 点对点
(b) 多点（多站）
(c) 广域网
(d) 局域网

5.4 解释并描述用Kermit协议从一台计算机到另一台计算机传送文件的操作，包括：
(a) 用户命令
(b) 帧格式与帧类型；
(c) 包括重发帧序列实例

5.5 列出ASCII字符集中10个传输控制字符，并解释在面向字符协议中它们的功能。

5.6 列出二进制同步控制（BSC）协议的数据帧（块）与监管帧（块），并表明各传输控制字符的位置。

5.7 区分BSC协议中的"轮询"、"选择"与"快速选择"几个术语。在BSC协议中说明这三种操作方式的典型数据帧（块）与监管帧（块）序列。

5.8 在ARPANET数据链路协议中，列出用于解释保持连续双工I帧流的格式。识别实现组合差错控制方案的数据链路每端要求的变量状态。

5.9 借用图形，说明下列网络配置中高级数据控制（HDLC）协议命令与响应的方向：
(a) 具有一个主站与从属站的点对点
(b) 具有复合站的点对点
(c) 具有一个主站的多点

5.10 列出HDLC协议的基本帧格式与扩充帧格式。给出每种情形的控制字段的结构，解释每个字段的意义与用途。

5.11 假设HDLC协议，区别NRM与ABM的工作方式。针对每一种方式，表示首先建立链路然后断开链路的典型帧序列。表示所用的各种帧类型与每个帧的地址与轮询/结束位的用法。

5.12 定义HDLC协议中用作确认的监管帧。假定I帧流是单方向，用时序图说明HDLC协议的确认过程的一个典型帧序列。图中包括链路重传列表与发送序列变量与接收序列变量的内容。也给出每个帧中包含的发送序列号以及轮询/结束位状态。

5.13 解释"捎带确认"术语的含义。采用一个典型的帧序列说明HDLC协议为何使用捎带确

认，表明每个发送帧包含的发送序列号与接收序列号，重发列表的内容与链路每端的发送变量与接收变量。

5.14 概述窗口流控制的操作。假定发送窗口K与发送窗口变量V(W)，推导出每个接收到帧中包含的发送序列号与接收序列号的范围。

假定发送窗口为3，单向I帧流，采用HDLC协议，利用帧序列说明窗口流控制机制的操作。 265

5.15 解释数据链路协议“多链路过程”术语的含义。利用图说明单链路过程，并表示如何扩充帧格式以包括附加序列号。

5.16 列出使用LAPD协议传送下列信息所用的用户服务原语的时序图：

(a) 无确认（最佳尝试）

(b) 确认

概述LAPD的帧格式并解释8位组地址字段的结构与用法。

5.17 借助时序图，列出LLC协议下列操作的用户服务原语：

(a) 无连接

(b) 面向连接

针对以上情况，区别术语：

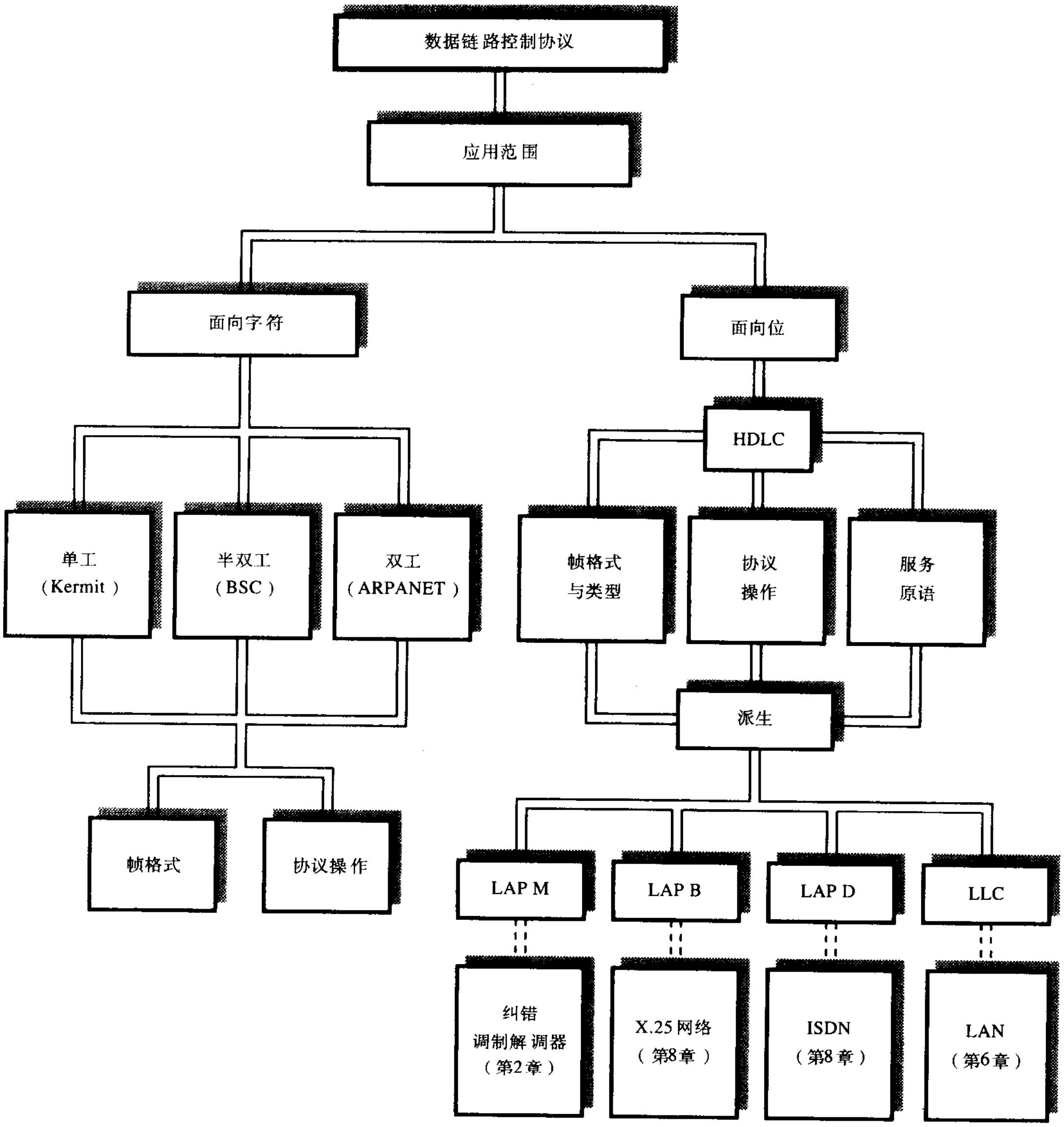
(a) 无确认发送数据

(b) 带确认发送数据

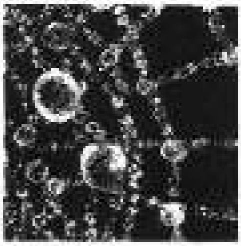
(c) 应答请求数据

5.18 利用时序图表示LAN的数据链路层使用的介质访问控制（MAC）子层相关的用户服务原语。利用各种MAC用户原语表示上面的LLC子层用户服务传送的监管帧与数据帧。 266

本章概要


数据链路控制协议
应用范围
面向字符
面向位
HDLC
单工
（Kermit）
半双工
（BSC）
双工
（ARPANET）
帧格式
与类型
协议
操作
服务
原语
派生
帧格式
协议操作
LAP M
LAP B
LAP D
LLC
纠错
调制解调器
（第2章）
X.25网络
（第8章）
ISDN
（第8章）
LAN
（第6章）

267



第二部分　计算机网络

在第二部分中，我们关注的是用来互连分布式计算机群的不同类型计算机网络的运行模式，以及它们的各种接口标准和协议。当网络中的计算机分布在某一局部区域（例如在一幢建筑或者一所校园中）时，我们称这种网络为局域网（LAN）。当网络中的计算机分布在一个更大的地理区域（例如国家）时，我们称这种网络为广域网（WAN）。当整个网络由多个局域网和广域网互连组成集合时，我们称之为互联网。

第6章描述了与目前成为国际标准的不同类型有线局域网相关的操作和接口协议，以及使用无线电或光（红外线）作为传输介质的无线局域网的操作特点。

在第7章中，我们描述了高速局域网的操作，还讨论了关于网桥的工作机制和协议。其中的两种网桥被用来扩展局域网的容量和覆盖域。

第8章描述了关于两种广域网的操作和接口协议：公共数据网和专用网络。前者是由公共载体安装和管理，而后者由大型国内公司或者跨国公司安装、管理和运行。

第9章讨论了当网络由诸如局域网和广域网的混合网络的互连集合组成（就是互联网）时，必须考虑的问题以及所需的额外协议。问题包括寻址和路由选择。

第10章是关于宽带多业务网络。这些网络不仅提供数字业务，而且支持诸如声音和视频等其他媒体类型的传输业务。

第6章 局 域 网

本章目的

读完本章，应该能够：

- 描述通常应用于有线局域网的各种拓扑结构和传输介质。
- 了解基带和宽带工作方式之间的差异。
- 描述应用于局域网的可选介质访问控制方式。
- 描述带冲突检测的载波侦听多路访问（CSMA/CD）总线网络的主要组成部分和工作模式。
- 描述令牌环网络的主要组成部分和工作模式。
- 描述令牌总线网络工作模式的一些选定方面。
- 描述跟无线局域网相关的技术问题。
- 理解无线局域网的操作特点。
- 识别应用于局域网的各种网络依赖协议的功能，以及能够描述逻辑链路控制层和网络协议层的服务和工作机制。

引言

局部区域数据网络，我们通常简单地称之为**局域网**或**LAN**，通常用来互连位于一幢大楼内或者局部区域建筑群内的基于计算机的DTE分布式群体。例如，我们既可以使用LAN来互连分布在一幢大楼内或者诸如大学校园的建筑群内办公室的工作站，也可以用它来互连分布在一个工厂或医院混合建筑群内的基于计算机的设备。既然所有的设备都位于单一设施内，
271 局域网通常由组织机构安装和维护。因此我们也把局域网称为**专用数据网**。

使用LAN建立的通信通路和通过公共数据网连接建立的通信通路之间的主要差异是，LAN由于相对较短的物理距离通常能提供更高的数据传输速率。在国际标准化组织（ISO）为开放式系统互连（OSI）所制定的参考模型中，这个差异只表现在较低的网络依赖层。在很多情况下，对于参考模型中的较高协议层来说，两种网络是没有差异的。本章将主要描述不同类型的LAN以及相关网络依赖协议层的功能和工作机制。

有两种类型完全不同的局域网：**有线局域网**和**无线局域网**。正如名称所体现的，有线局域网是利用（固定）线路（比如双绞线或同轴电缆）作为传输介质，而无线局域网是利用无线电或者光波。我们会看到，与两种类型相关的选择方案和技术问题是截然不同的，因此我们会分别介绍每一种类型。

6.1 有线局域网

在开始描述不同类型有线局域网的结构和操作之前，让我们先来确定一些我们必须考虑的问题。这些问题的概要在图6-1中给出。注意这只是一个概要，可能在图中枝节末梢间还有许多连接。我们将较详细地介绍每一个被确定的问题。

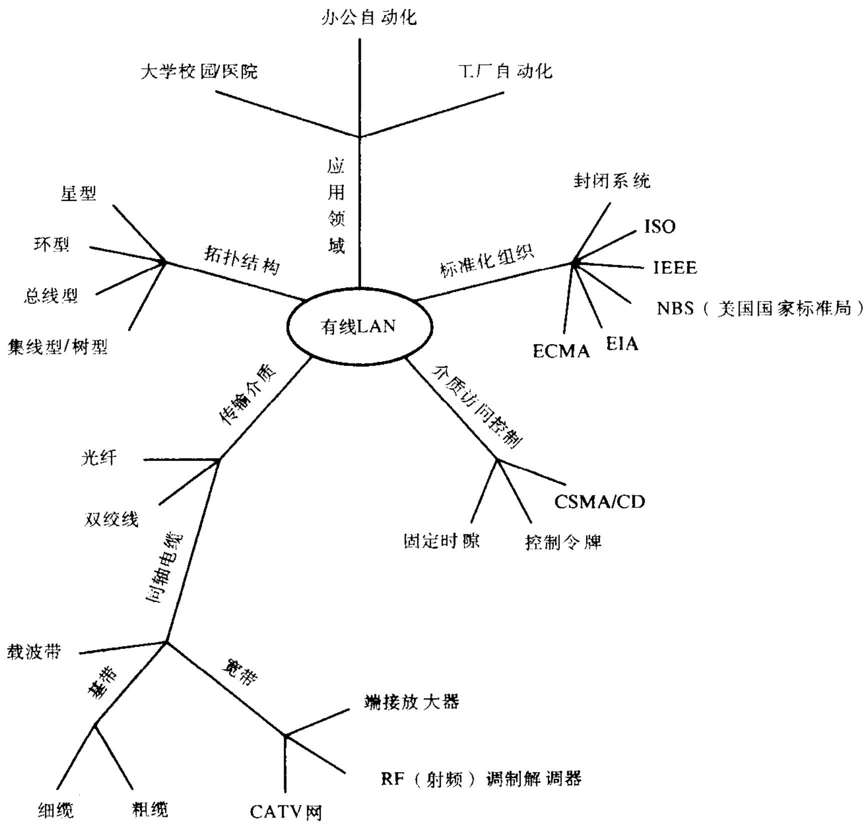


图6-1 LAN 问题选择

6.1.1 拓扑

大多数广域网，例如公共交换电话网（PSTN），使用**网格拓扑**结构（有时称为**网络拓扑**）。但是，对于局域网来说，用户DTE之间有限的物理距离允许我们使用更简单的拓扑。四种通常使用的拓扑结构是星型拓扑、总线型拓扑、环型拓扑和集线型拓扑。如图6-2所示。

星型拓扑LAN的最好例子是数字**专用自动小交换机**（PABX）。通过传统模拟PABX建立的连接在许多方式上和通过模拟PSTN建立的连接相似，因为网络中的所有通路被指定传送有限带宽的模拟语音。所以，用它们传送数据，我们需要调制解调器，这已在第2章中讨论过。但是大多数现代的PABX在交换机中使用了数字交换技术，因此我们也称之为**专用数字交换机**（PDX）。而且，已经实现的能执行必要的模拟—数字、数字—模拟转换的集成电路并不昂贵，这使得它正迅速成为扩展到用户出口的数字工作模式的通用惯例。这意味着可交换的64kbps（通常用于数字语音的数字化速率）通路可以在每一个用户出口得到，因此使得它既可以传送语音也可以传送数据。 272

然而，PDX的主要应用是为集成语音数据终端（工作站）的局域群体间提供除正常语音通信之外，用于交换电子邮件、电子文档等的可交换通信通路。而且，PDX内部的数字技术使得它能提供诸如**语音存储转发**（就是说一个用户保留（存储）另一个用户的语音报文用于稍后再恢复（转发））和电话会议（一个呼叫中有多个用户参与通信）业务。

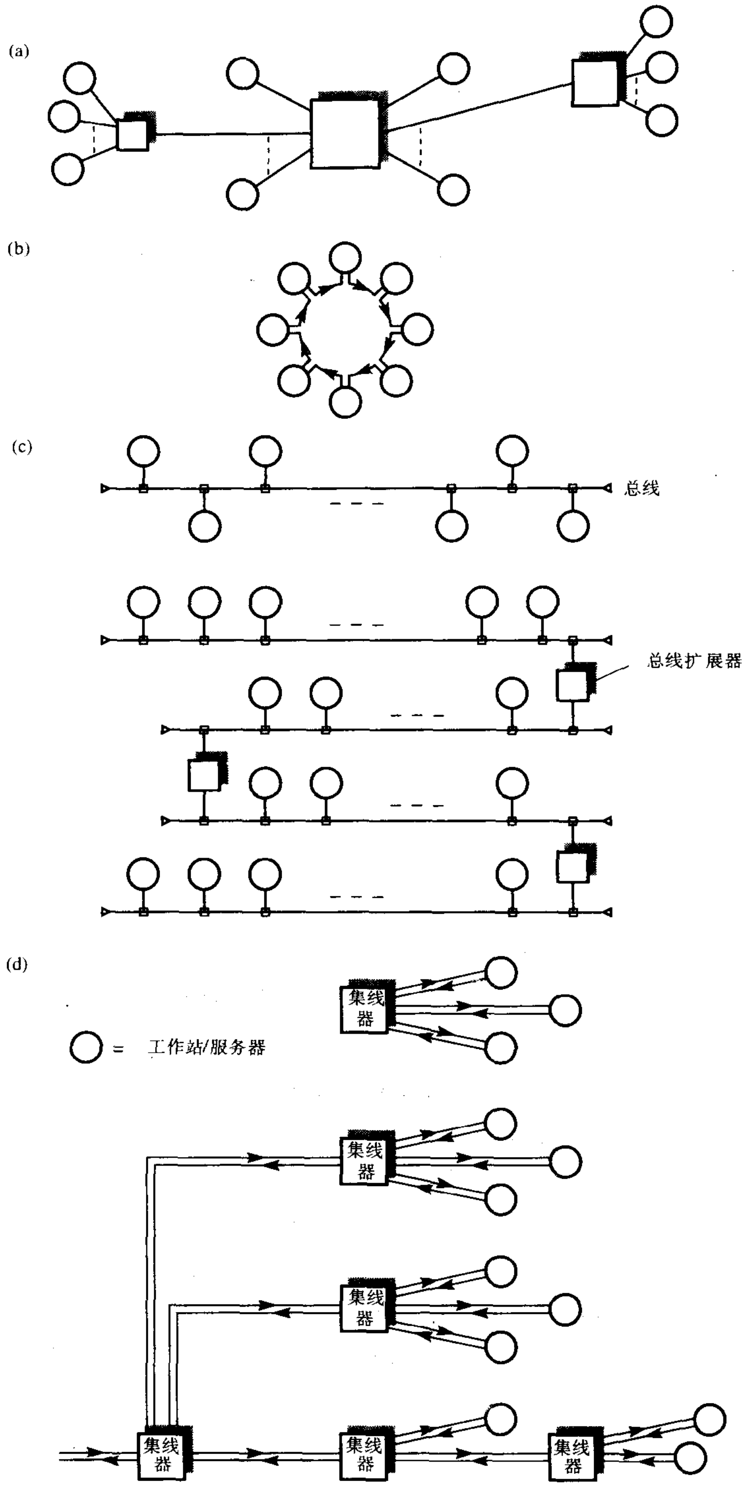


图6-2　LAN拓扑

(a) 星型　(b) 环型　(c) 总线型　(d) 集线型/树型

设计互连基于计算机的局域设备的数据通信子网LAN，总线型（线性）和环型是首选的拓扑结构。实际上，总线型网络通常扩展成一组总线互连，从而构成一棵**连根拔起树**。通常，**总线拓扑**由单一干线电缆连接各DTE所在位置（比如办公室）组成网络，电缆上的物理连接允许用户DTE访问网络提供的业务。然后使用适当的介质访问控制（MAC）电路和算法，使得所属的各DTE可以共享传输带宽。 273

使用**环型拓扑**，网络电缆从一个DTE连到另一个DTE直到所有的DTE以一个环的形式互连起来。环型拓扑的特点是每个单向运行的相邻DTE间是直接的点对点连接。适当的MAC算法确保各用户能共同使用环。

环型和总线型拓扑所使用的数据传输速率（通常1～10Mbps）意味着它们最适合于诸如办公室中的工作站或者加工车间中智能控制器等基于计算机设备的局域群体的互连。

图6-2(d)显示的是一种环型和总线型拓扑的变体，我们称之为**集线型拓扑**。虽然这种网络有着类似于星型拓扑的形状，但是集线型只是总线型或者环型布线简单地塌陷成一个中心单元。用来连接每一个到总线的DTE或者环的线路现在从集线器中扩展出来。因此，不像PDX，集线器并不执行任何交换功能，而只是简单地由一组中继器组成，以和总线型或环型网络一样的方式把从一些DTE接收到的所有信号重发到另一些DTE去。正像我们所看到的，集线器可以用分层的方式连接形成**树型拓扑结构**。这种混合拓扑工作起来就像简单的环型或者总线型网络或者这两类网络的互连集合。我们会在6.2节中讨论实际的例子。

6.1.2 传输介质

双绞线、同轴电缆和光纤是用于LAN的三种主要传输介质。

双绞线（非屏蔽和屏蔽）主要用在星型和集线型网络中。因为比起同轴电缆或者光纤，双绞线比较软，更加容易安装。同时，大多数办公室为电话铺设的布线槽适合双绞线，所以比起同轴电缆或光纤，为数据目的而额外安装的双绞线成本比较低，因为安装同轴电缆或光纤需要铺设新的缆槽。图6-3(a)说明了主要方案。

正如第2章所讲的，双绞线长度的最大限制依赖使用的传输比特率。一般1Mbps可以传输100米，如果借助额外的线路除去串扰的话，可以达到10 Mbps传输100米。一种典型的做法是用双绞线连接DTE和同一层中最近的线路盒，而用同轴电缆把每层的线路盒连接到建筑中的主集线器。对于涉及多幢建筑的网络应用来说，通常用光纤连接每幢建筑的主集线器到中央集线器。后者一般以较高的传输比特率工作，逻辑上配置成环型网络。这种做法经常被称做**结构化布线**。

同轴电缆同样被广泛地应用在以基带或宽带传输方式工作的局域网中，主要是总线型网络。我们已经在第2章中讨论了它们的基本工作机制。两种类型的电缆用于基带：**细缆**和**粗缆**。这些名称是根据电缆的直径得出的：细缆的直径是0.25英寸，而粗缆的直径是0.5英寸。通常，两种电缆可以运行相同的传输比特率10Mbps，但是细缆会导致更大的信号衰减，连接中继器的细缆最大长度是200米而粗缆可以达到500米。回忆一下中继器是用来把接收到的衰减信号重新生成、放大还原成初始信号的设备。细缆、粗缆这两种工作模式我们又分别称为10 Base 2（10Mbps、基带、200米最大传输距离）和10 Base 5（10Mbps、基带、500米最大传输距离）。 274～275

细缆通常用来互连同一办公室或者实验室中的工作站。同轴电缆的物理连接器直接连在工作站的接口卡上。电缆总线采用从一个DTE连到另一个DTE的菊花链形状。

相比之下，粗缆因为它更为坚硬的结构通常远离工作站安装，比如沿着走廊。在称为**连接单元接口**（AUI）的主要同轴电缆分接（连接）点和每个工作站附接点间必须使用称为**分**

276 **支电缆**（drop cable）的额外配线和称作**收发器**的发送接收电子设备。这种做法比较昂贵，因此主要用于工作站位于不同的办公室时或者用于互连细缆段时。图6-3(b)分别说明了这两种类型。

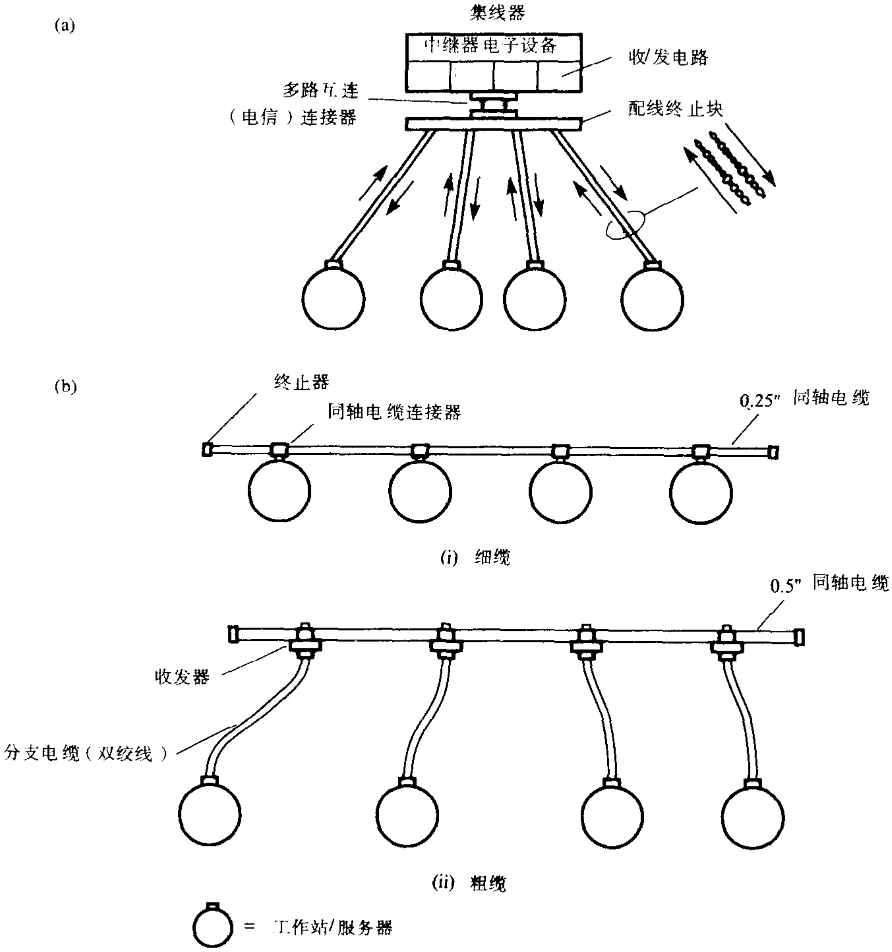


图6-3　传输介质

(a) 双绞线　(b) 基带同轴电缆

正像我们在第2章中所描述的，宽带传输中电缆的总可用带宽（频率范围）被分成许多更小的子频带或子信道，而不像基带中对应的传输二进制数据流，以两种电平的方式在电缆中传送信息。每个子频带在一对特殊调制解调器的帮助下提供一个单独的数据通信信道。这种工作方式被称为频分多路复用，因为使用的频率属于无线电频带，调制解调器是射频（RF）调制解调器。这个方式被称为宽带工作方式，广泛地应用在**有线电视**（CATV）行业中，负责把多路电视频道复用到一条同轴电缆中。

图6-4(a)显示了一个典型的有线电视系统。每个电视频道分配一个特定的频带，一般为6MHz带宽。每个接收到的视频信号（从多个天线）被调制成选定频带中的**载波频率**。经过调制的载波信号通过有线电视网传送到每一个用户终端出口。用户通过调节到适当的频带来选择特定的电视频道。

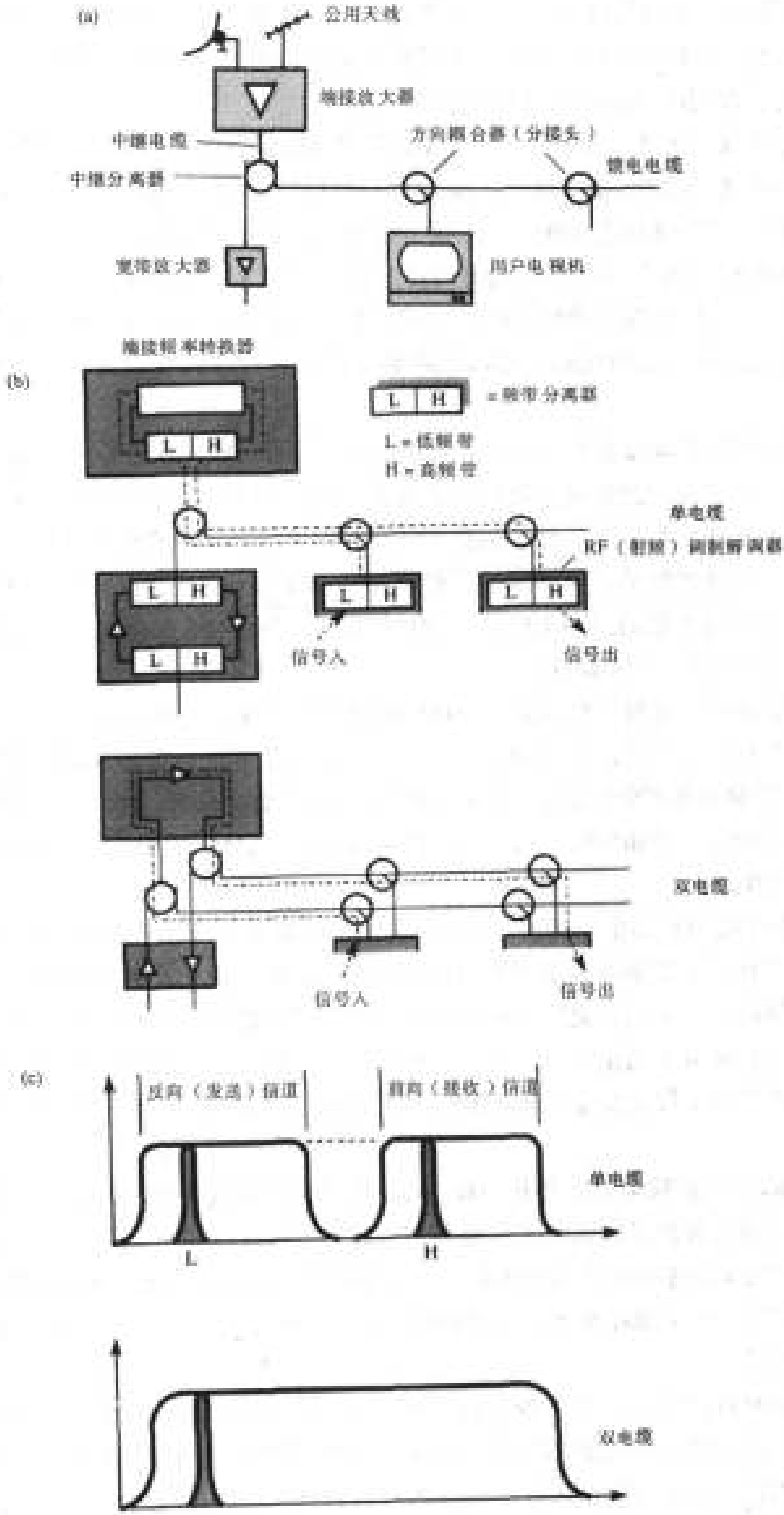


图6-4 宽带同轴电缆系统

(a) 基本CATV系统组成部分 (b) 数据网络方案 (c) 使用频率

相似的情况下，我们可以通过分配每个信道总带宽的一部分而从一条电缆中得到一个数据传输信道范围，每个信道的带宽取决于所需数据传输速率。但是，数据通信通常需要双通道（22z）能力。我们可以通过下面两种方式得到：

1）**单电缆系统** 在同一电缆中分别给发送和接收通路分配两个不同的频带。

277 2）**双电缆系统** 使用两条单独的电缆，一条作为发送通路，另一条作为接收通路。

图6-4(b)显示了两种电缆系统的示意图。它们的主要差异是双电缆系统需要安装两倍数量的电缆和分接头，而它在每个方向可以得到全部带宽（一般为5～450MHz）。而且，**端**

278 **接**（headend（HE））设备只是简单的放大器，但是在单电缆系统中，需要用称为**频率转换器**的设备来把流入的与各种接收通路有关的频率信号转化成相应的用于发送通路的流出频率。

首先用射频调制解调器使反方向（朝头端）选定频带中的正弦信号被要发送的数据调制。这个信号通过一种特殊的**方向耦合器**或者**分接头**（用来使大多数要传输的信号以反方向流入电缆头端）流入电缆。然后频率转换器负责把收到的不同接收频带的信号转换成相应的一组发送频带信号。这样接收到的调制信号经过端接器进行频率转换，与接收DTE对应的射频调制解调器就调频接收匹配的经频率转换过的信号频带。最后接收调制解调器从收到的信号中解调出发送数据再送给连接的DTE。

我们可以推断出一对频率只能为两个DTE间提供一个单工数据通路。因此，两对独立的频率能支持双工通信。不过，一个9.6kbps的单工数据信道总共只需要20KHz的带宽，所以一对6MHz的子频带能提供300个这样的单工信道或者150个双工信道。较高的数据传输速率信道需要更多的可用带宽，例如5Mbps的全双工信道需要两个6MHz带宽，而10Mbps的全双工信道则需要三个6MHz带宽。

我们从一条电缆得到多条不同数据通路所付出的代价是相对较昂贵的射频调制解调器对。但宽带同轴电缆相对于基带电缆来说可以传输更远的距离。因此，宽带同轴电缆的主要用途是作为一种灵活的传输介质，应用于制造工业或者由多幢建筑组成的设施，尤其当这些建筑相当分散（最远相隔数十公里）时。当以这种方式应用时，其他诸如闭路电视和语音等服务能较容易地整合到用于传输数据的电缆中。因此宽带是基带的一种可行性替代品，为网络提供一系列业务。

光纤（见第2章）由玻璃或者塑料制成，能以超越双绞线或者同轴电缆的一切可能的数据传输速率工作。因为数据是通过光束传输的，因此信号不受电磁干扰影响。由此光纤最适合满足有较高数据传输速率要求或者对电磁有高抗干扰性（诸如拥有大型电子设备的工业工厂）要求的应用。同时，由于光纤不会有电磁辐射（通常会被窃听者截取），因此它适合于有较高安全性要求的项目。

因为光纤中的数据是通过光束传输的，所以需要特殊的电光和光电传送接收电子设备。同时，用于光纤的物理连接器要比用于双绞线或者同轴电缆的连接器昂贵，而且在光纤上安

279 装分接头更加困难。基于这些原因，我们在集线型网络、高速环型网络或者需要使用点对点传输通路的网络中使用光纤。后者的两个例子是光纤分布式数据接口（FDDI）网络和分布式队列双总线（DQDB）网络，我们将在第7章中讲述。

6.1.3 介质访问控制方式

当两个DTE通过星型网络建立通路时，中央控制模块（比如PDX）会确保该传输通路在整个呼叫期间被保留。但是环型和总线型拓扑只有一条连接所有DTE的逻辑传输通路。因此，必须实施一种规定，确保所有连在网络上的DTE能公平地访问传输介质。在各种标准文件中被采用的两种技术是用于总线型网络拓扑的**带冲突检测的载波侦听多路访问**（CSMA/CD）和用于总线型或者环型网络的**控制令牌**。基于**分槽环**（slotted ring）的访问方式也广泛地应用在环型网络中。

1. CSMA/CD

CSMA/CD方式只应用在总线型网络。在这种网络拓扑中，所有的DTE直接连接到同一根电缆，它负责传输任何一对DTE间的所有数据。电缆以**多路访问**（MA）模式工作。为了传输数据，发送方的DTE先把数据和所需的目标地址封装在一个帧中，地址在帧的头部。然后该帧在电缆上传送（或者称之为**广播**）。任何时候有一个帧传送，所有连在电缆上的DTE都会检测。当目标DTE发现当前传送的帧头部有自己的地址时，它继续读取包含在帧中的数据并根据定义的链路协议响应。源DTE地址被包括在帧的头部以使接收方DTE可以发送响应给源DTE。

在这种工作方式下，两个DTE可能同时尝试向电缆发送一个帧，从而引起两个源DTE发送的数据被破坏。为了减少这种概率，源DTE在发送一个帧之前先对电缆进行电子侦听以判断当前电缆上是否有帧在传输。当一个**载波信号被侦听到**（CS），该DTE就延迟它的数据发送，直到正在传送的帧完成传送才尝试发送。甚至这种情况下，两个想发送帧的DTE可能会同时断定在总线上没有活动（传送），然后同时开始发送它们的帧。这时冲突就发生了，因为两个帧会发生冲突然后被破坏。图6-5概要说明了这种现象。

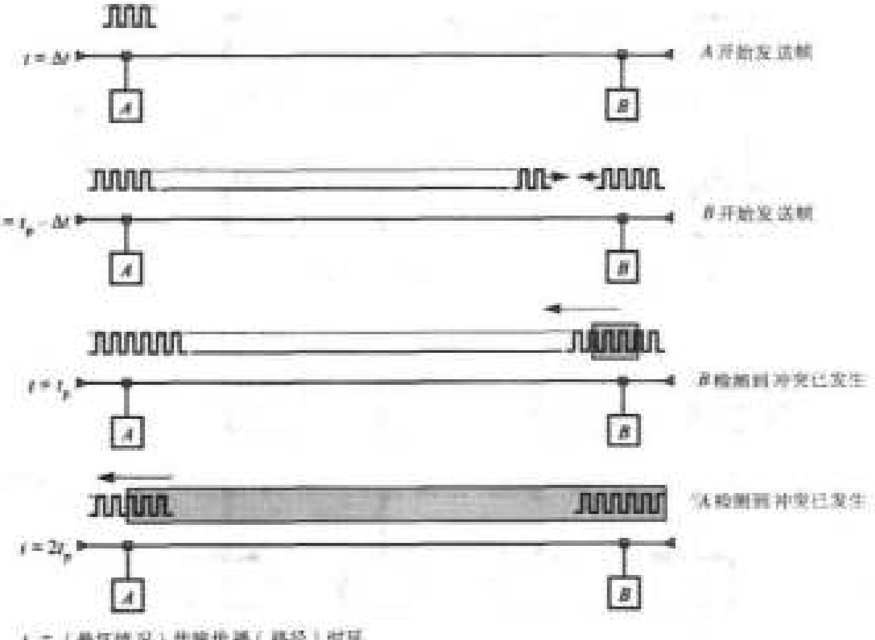


图6-5 CSMA/CD冲突示意图

280 一个DTE在电缆上传送帧时会同时监视电缆上的数据信号。如果发送的信号和检测到的信号不同时，冲突发生了——冲突检测（CD）。为了让其他陷入冲突的DTE知道冲突已经发生了，第一个发现冲突的DTE会在一个短周期内继续发送一个随机的位模式以加强冲突，这称之为**阻塞序列**（jam sequence）。两个（或多个）DTE在尝试重发受影响的帧前会额外等待一个较短的随机时间间隔。我们可以得出结论：对CSMA/CD总线的访问是随机的并取决于网络（电缆）负载。注意既然线路上的传输比特率相当高（可以达到10Mbps），因此网络负载会显得很低。同时，既然只有当线路上无活动时帧传送才开始，实际上发生冲突的概率同样很低。

2. 控制令牌

另一种对共享传输介质的控制方法是通过控制（许可）令牌实现的。该令牌根据一套定义好的能够被连在传输介质上的所有DTE理解和遵守的控制规则从一个DTE传递到另一个DTE。一个DTE只有在获得令牌的情况下才有权发送帧，当帧传送完毕后，它会把令牌传递给另一个DTE从而允许该设备访问传输介质。工作顺序如下：

281 1) 首先建立一个逻辑环，它把所有DTE连接到物理传输介质上，然后建立一个控制令牌。

2) 令牌沿着逻辑环从一个DTE传递到另一个DTE，直到等待发送帧的DTE接收到它。

3) 然后得到令牌的DTE使用物理传输介质发送帧，完成后它把控制令牌传递给逻辑环中的下一个DTE。

连到物理传输介质上的活动DTE中存在的监控功能为逻辑环连接及令牌消失后的初始化和恢复提供了基础。虽然监控功能通常在传输介质上所有DTE中都有，但是在某一个时刻只有一个DTE负有恢复和重新初始化的责任。

物理传输介质不必是环型拓扑，令牌同样可以应用到对总线网络的访问控制中。图6-6说明了两种类型网络上逻辑环的建立。

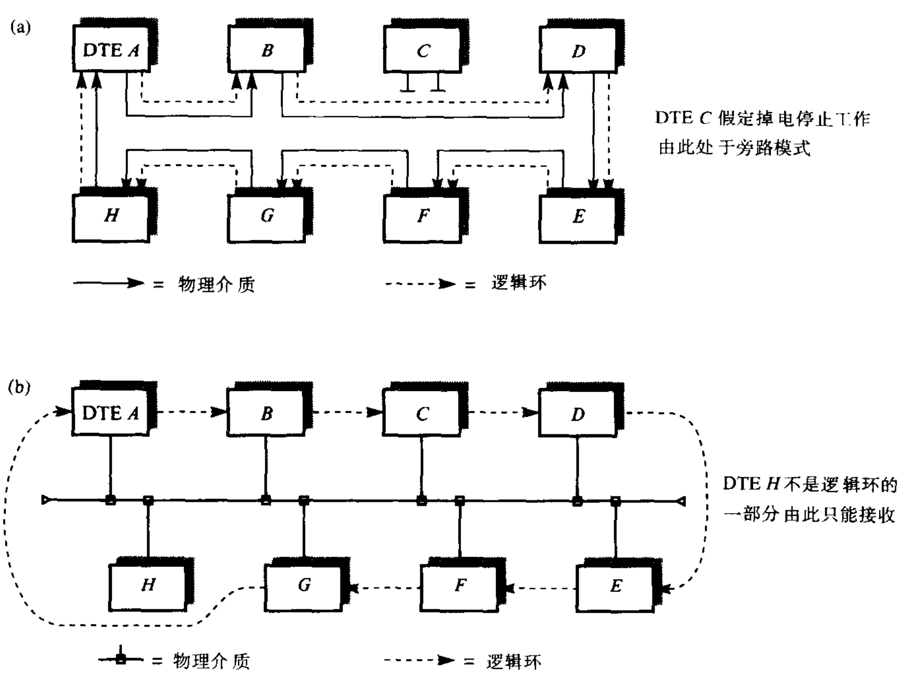


图6-6 控制令牌MAC

(a) 令牌环 (b) 令牌总线

在一个物理环结构中（见图6-6(a)），令牌传递环的逻辑结构与物理环结构相似，令牌沿着连在物理环上的DTE顺序传递。但是，在总线型网络中，逻辑环的顺序不必同连在线路上的DTE物理顺序一致。而且，在一个总线型网络中应用令牌访问控制方式，所有的DTE不必都连到逻辑环上。例如，图6-6(b)中DTE *H*不属于逻辑环。这意味着它只能以接收模式工作，因为它不可能得到控制令牌。令牌访问控制方式的另一个特点是优先级与令牌相关，因此先发送拥有较高优先级的帧。我们会在6.2.2节和6.2.3节中更深入地介绍令牌访问控制方式的其他方面。 282

3. 分槽环

分槽环主要应用在环型网络的访问控制。首先由环上一个名为**监控站**的特殊结点初始化环，在环中产生固定数目的位。该二进制数据流沿着环从一个DTE到另一个DTE循环传递。每一个DTE接收到一个位，DTE接口检查（读取）这些位然后传递（转发）给下一个DTE，依次进行。监控站不管环中有多少DTE，始终确保恒定个数的位在环中循环。整个环配置若干个**时隙**（slot），每一个时隙由一组位组成，并能携带一个固定长度的信息帧。图6-7(a)显示了一个帧时隙的格式。

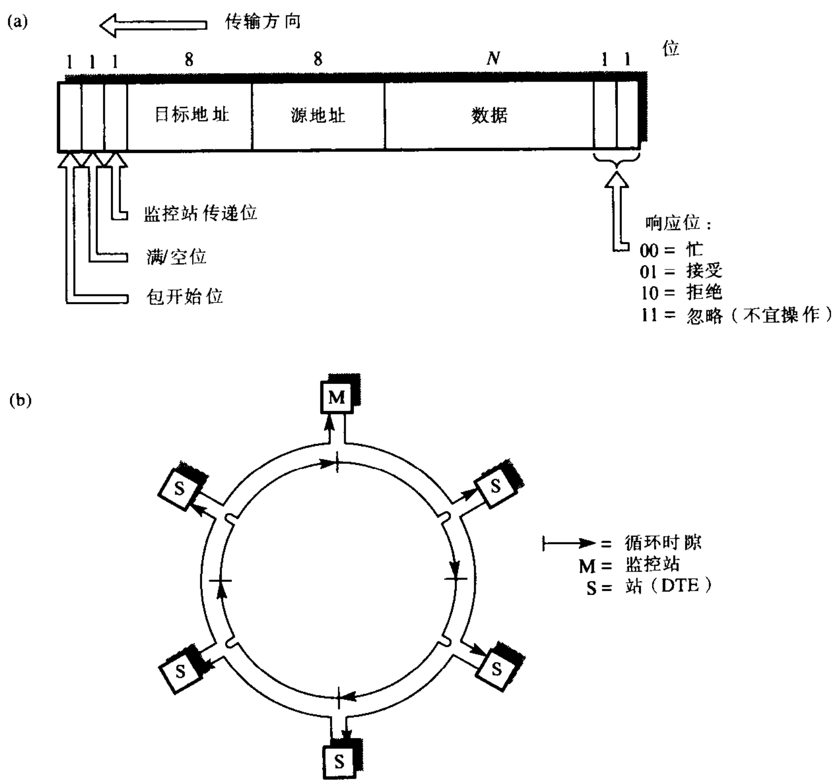


图6-7 分槽环原理

(a) 每个时隙的位定义 (b) 拓扑结构

初始化时，监控站会把每个时隙头部的满/空位设成空从而使所有的时隙都标记为空。当某个DTE想发送帧时，它等待，直到检测到一个空时隙。然后，DTE标记这个时隙为满，接着把帧内容插入时隙中，连同所需的目标DTE地址和源DTE地址放入帧头部，帧尾部的两个 283

响应位都设成1。含有这个帧的时隙会沿着物理环从一个DTE向另一个DTE依次传递。环中的每个DTE检查标记为满的时隙头部的目标地址，如果检测到是自己的地址，则认为该时隙的帧就是要接受的帧，它就会从时隙中读取帧的内容，同时不变地把帧内容沿环转发。读取完帧内容以后，目标DTE会修改时隙尾部的响应位对，表示已经读过该帧。如果目标DTE正忙或不宜操作，则响应位对作相应的标记或者保留不变（表示不宜操作）。

源DTE在启动帧的发送后，一直等待，直到该帧绕环一周。由于每个站均知道环上时隙总数（一个固定值），由环接口计算时隙转发数即可知道所发帧的到来。接到发送帧所用时隙的第一位时，再次把它标记为空，然后等待读取时隙尾部的响应位，来决定下一步要采取的动作。

监控站传递位用于监控站检测DTE在发送完帧以后，是否释放时隙失败。这个位在源DTE向环发送帧时被置“0”，随后当该位在环接口处被转发时由监控站置“1”。如果监控站在转发满时隙时，发现该位被置“1”，说明源DTE标记时隙为空有故障，因此把时隙头部的满/空状态位置“0”。

注意使用分槽环介质访问控制方式，每个DTE在某一时刻只能在环上传送一个帧。同时在传送另一帧之前它必须释放用于传送前一帧的时隙。这样的情况下，对环的访问是公平的，并被所有互连的DTE设备共享。分槽环的主要缺点如下：

1）需要特殊的（脆弱的）监控站结点来保持环状基本结构。

2）每个完整链路层帧的传送通常需要多个时隙。

当然，对于令牌环，一旦某个DTE得到控制令牌，它能传送一个含有多字节信息的完整帧，将它作为一个单元。

6.1.4 标准

当局域网在20世纪70年代末80年代初出现的时候，许多不同的网络类型被提出来并得到实现。但是，由于它们之间存在小的差异，这些网络只能用来互连由局域网供应商提供的计
284 算机或工作站。这些网络被称为**封闭系统**。

为了改变这种状况，许多致力于给局域网制订一套公认标准的国家标准机构开展了主要的工作。这方面做出主要贡献的是IEEE，它制订了IEEE 802系列标准，国际标准化组织现已采纳它作为国际标准。正如所见，有线局域网不止单一的类型，它有不同的类型，每一种还有特有的拓扑结构、介质访问控制方式和应用范围。我们会看到标准文档中一些不同类型的有线局域网，在6.5节我们还会看到相关的协议。

6.2 有线局域网类型

用来互连基于计算机的本地设备集合的有线局域网的两种最主要的类型是总线型和环型。当前，这两个类型有各种变型，虽然许多不遵循LAN的国际标准。在标准文档中有三个类型，是CSMA/CD总线型、令牌环型和令牌总线型。下面的描述只限于这三个类型。

6.2.1 CSMA/CD总线型

CSMA/CD总线型网络被广泛地应用到科技和办公环境。基于历史的原因，它又称为**以太网**。虽然在标准文档中也支持其他电缆介质，但是它一般以10Mbps基带同轴电缆网络或者10Mbps双绞线网络实现。它们包括：

10 Base 2　细缆（直径为0.25英寸）同轴电缆，最大段长为200米

10 Base 5　粗缆（直径为0.5英寸）同轴电缆，最大段长为500米

10 Base T　双绞线分支电缆集线型（星型）拓扑

10 Base F　光缆分支电缆集线型（星型）拓扑

虽然使用了不同的传输介质，但它们都以相同的MAC方式工作。使用同轴粗缆和细缆，主要的差异在收发电子设备的位置。如果是粗缆，它位于电缆分接头，被称为**集成分接头与收发器**单元。如果是细缆，就直接接在DTE的接口卡上，因此收发器位于后端。因为细缆网络安装起来比粗缆网络便宜，所以又称为**廉价网**。

图6-8显示了跟粗缆配置相关的各种部件。**分接头**用来产生电缆上的不插入物理连接，就是说不必割断电缆。它由一个螺旋装置组成，能穿透电缆的外层保护屏蔽然后与中心的导体直接接触。螺旋的外部与电缆的外层表面接触进而完成电子连接。 285

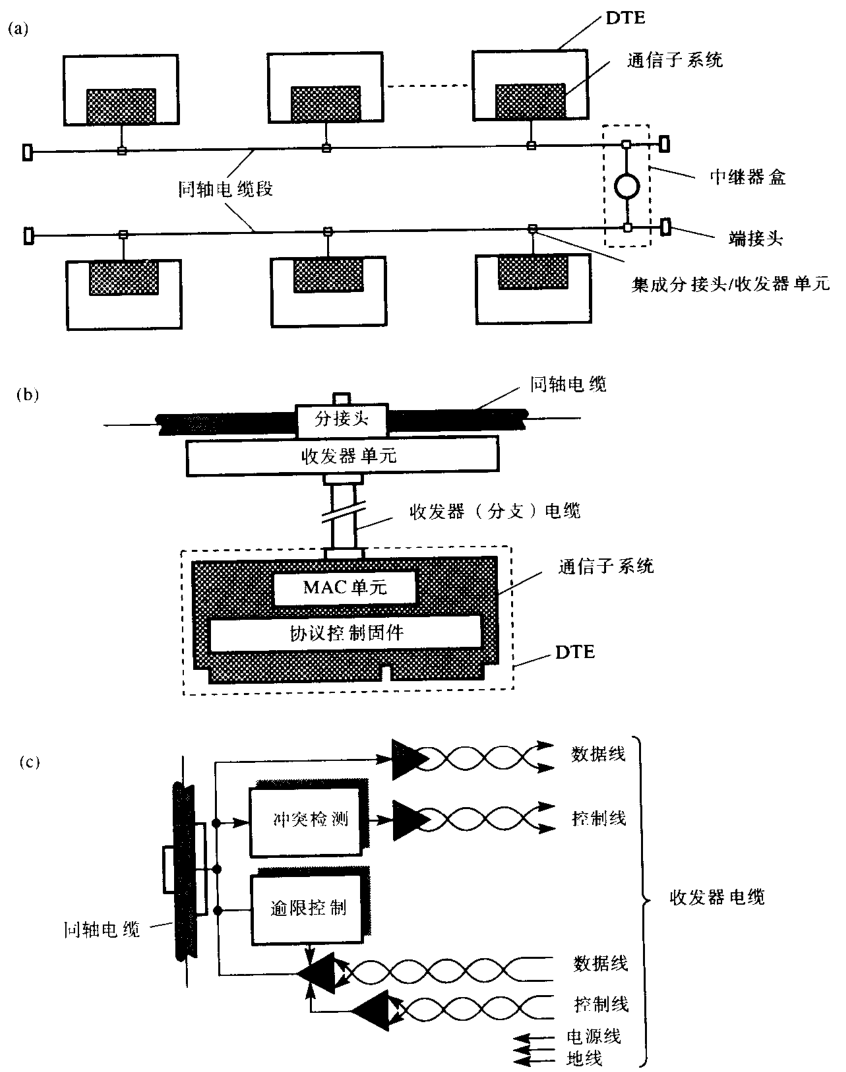


图6-8　粗缆CSMA/CD总线网络组成部分

(a) 电缆布局　(b) DTE接口　(c) 收发器示意图

收发器包含具有如下功能的必要电子设备。

- 向电缆发送数据和从电缆接收数据。
- 检测电缆介质上的冲突。
- 在同轴电缆和电缆接口电子设备间提供电子屏蔽。

286 • 保护电缆，防止来自收发器或者连接DTE的故障。

最后一个功能通常称为**逾限控制**。因为如果没有适当的保护电子装置，则当故障发生时出错的收发器（或DTE）会持续地向总线电缆介质发送随机数据（逾限），以致限制或破坏其他传输。逾限控制就是当违反某个规定的时间界限时，把发送数据通路同总线电缆隔离开来。例如，所有在总线电缆上发送的帧都有一个规定的最大长度。如果超过这个长度，逾限控制就会禁止数据进一步输出，防止其到达电缆。

收发器通过含有五对双绞线的屏蔽电缆连到DTE主机上：一对从DTE为收发器供电；两对用于传送数据（一对发一对收）；另外两对用于控制目的（一对允许收发器发信号给DTE告知冲突，另一对用于DTE把数据发送通路与电缆隔离开来）。四对信号线分别驱动，这意味着主机DTE可以远离收发器达50米，即远离分接点。

如同轴细缆一样，使用双绞线和集线器配置的话，冲突检测功能位于DTE的接口电路板中。集线器的功能只是纯粹地接收和可靠地重发（转发）电子信号。图6-9给出了集线器配置和中继电子装置功能。

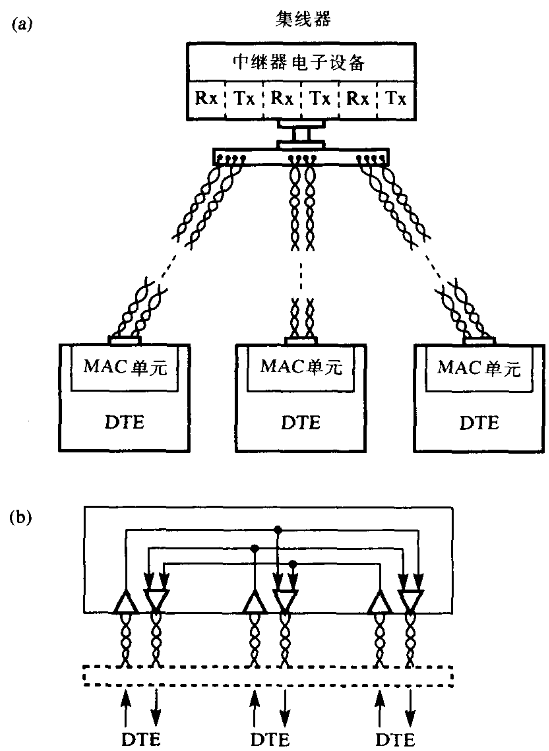


图6-9 集线器配置原理

(a) 拓扑结构 (b) 中继器示意图

我们看到每个DTE有两对双绞线（或者光纤）连到集线器——一对发送而另一对接收。为了使DTE中的冲突检测装置起作用，集线器中的中继器装置会把从某个输入对收到的信号

重发给其他所有输出对。中继器装置的主要作用是确保在输出对输出的（较强）重发信号不会干扰在输入对接收到的因为衰减而变弱的信号。这个效应称为近端串扰。需要称为**自适应串扰抵消电路**的特殊集成电路来确保100米长线路上能以10Mbps速率稳定工作。

无论使用何种传输介质，在每个DTE中的通信控制卡由以下部分组成：

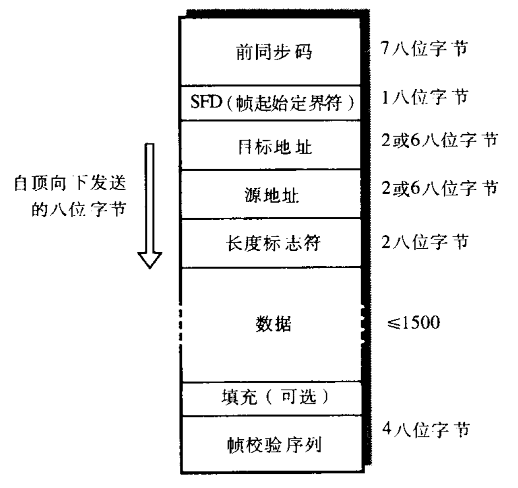
- 介质访问控制（MAC）单元，它负责封装与解封电缆上发送与接收的帧、差错检测和MAC算法实现等功能。
- **双端口随机访问存储器**（RAM），它允许MAC单元以较高的链路比特率接收和发送帧，并允许主机读写帧中的信息内容。

完整的通信子系统通常位于插在主机系统总线槽中的单块印刷电路卡上，相关的低级例程为主机软件提供一套定义好的帧发送和接收服务。大多数商业卡提供多种连接器以支持不 287
同类型的传输介质。

1. 帧格式和操作参数

图6-10显示了一个典型CSMA/CD总线型网络的帧格式和操作参数。当讲到MAC单元的操作时，我们会描述这些参数的含义和用法。

(a)



(b)

比特率	10 Mbps（曼彻斯特编码）
时隙间隔	512 位周期
帧间间隔	9.6μs
尝试极限	16
退避极限	10
拥塞长度	32 位
最大帧长度	1518 字节
最小帧长度	512 位

图6-10 CSMA/CD总线网络特征

(a) 帧格式 (b) 操作参数

在电缆上传送的每个帧含有8个字段，除了数据和相关的填充字段外，其他字段都是定长的。

前同步码字段位于帧的头部。它的功能是在实际的帧内容被接收前，允许每个介质访问控制单元中的接收装置可靠地得到位同步。前同步码的模式是7个8位组串，每个8位组是二进

制10101010。电缆上传送的所有帧使用曼彻斯特编码。正像我们在第3章中所看到的，前同步码使得每个DTE中的接收器收到一个周期性的波形。帧起始定界符（SFD）是一个8位组
288 10101011，它紧跟在前同步码后面告知接收器有效帧的开始。

目标网络地址和**源网络地址**分别说明了目标DTE和源DTE的身份。每个地址字段可以是16位或者48位，但是对于特定局域网，所有的DTE的地址应该是一样长的。目标地址字段的第一位指明了它是**单地址**还是**组地址**。如果是单地址，说明该帧是要传送给单个目标DTE。如果是组地址，说明该帧是要传送给一组逻辑上相关的DTE（组地址）或者其他所有连在网络上的DTE（**广播地址**或者**全局地址**）。后一种情况，地址字段被设成全1。

长度标志是2字节字段，它说明数据字段的字节个数。如果该值小于一个有效帧需要的最小字节数（最小帧大小），则可附加一串字节，称为**填充**。最后，帧校验序列字段含有4八位字节（32位）CRC值，用于差错检测。

2. 帧传输

289 当一个帧被传输时，首先MAC单元把帧内容封装在如图6-10(a)所示的格式中。为了避免介质上的其他传输帧争用，MAC单元中的MAC部分，先监听载波侦听信号，如果侦听到线路上存在正在传输的帧，就经历短暂的额外时延（称为**帧间间隔**），允许该帧由目标DTE接收和处理后，随后启动自身帧的传输。

当二进制数据流传输时，收发器同时会监听接收到的信号来检测是否有冲突发生。假设冲突没有被检测到，则发送一个完整帧，当帧校验序列字段被发送后，MAC单元等待来自电缆或者控制微处理器的新帧的到达。如果检测到冲突，收发器立即发出冲突检测信号，此信号被MAC单元检测到，它通过发送阻塞讯号以确保其他DTE能检测到该冲突。在阻塞讯号被发送后，MAC单元终止帧的传送，并安排好短暂的随机时间间隔后，进行帧重发尝试。

在冲突事件中，帧重发会以**尝试极限**的方式进行规定好的最大次数的尝试。因为重复出现冲突表明传输介质忙，MAC单元通过逐步增加两次帧重发尝试间隔时间来调整传输介质负载。帧重发的方案由一个称为**截断的二进制指数退避**进程控制。它是这样工作的：当阻塞讯号发送完后，假设还没到达尝试极限，MAC单元会延迟（退避）随机整数个时隙间隔，然后才尝试重发受影响的帧。如图6-5所示，一个给出的DTE在传输初始阶段经历冲突，**冲突窗口**是前同步码的第一位向电缆传输介质（网络）所有部分传播所需时间间隔的两倍。因此时隙间隔是一个DTE在确知发生冲突前必须等待的最坏情况的时间延迟。时隙间隔定义如下：

$$时隙间隔 = 2 \times（传输路径时延）+ 安全边际$$

传输路径时延是在电缆网络中从任意发送方到任意接收方的最坏情况的信号传播时延。这包括任何中继器经历的时延。时隙间隔是这个时延的两倍（允许差错的信号返回给发送DTE）加上安全边际。时隙间隔等同于以所用传输比特率传输字节个数所用时间来表示。例如，对于任意发送方和接收方间的最大距离为2.5公里而且传输速率为10Mbps的基带同轴电缆网络来说，时隙间隔等于512位周期或者64个八位字节。那么在第N次帧重发尝试前的时隙间隔数量选择一个分配的随机整数R，R的范围是[0，2^K]，这里K=min（N，退避极限）。图6-11(a)给出
290 了概要描述帧传输序列的流程表。

3. 帧接收

图6-11(b)概要描述了帧接收的过程。在每一个连接在电缆上的活动DTE中，MAC单元首
291 先检测到来自收发器的流入信号，然后开启载波侦听信号以禁止这个DTE的任何新发送。流

入的前同步码用于获得位同步，然后经过曼彻斯特编码的数据流会转换回通常的二进制形式，接着这些二进制数据流才被处理。

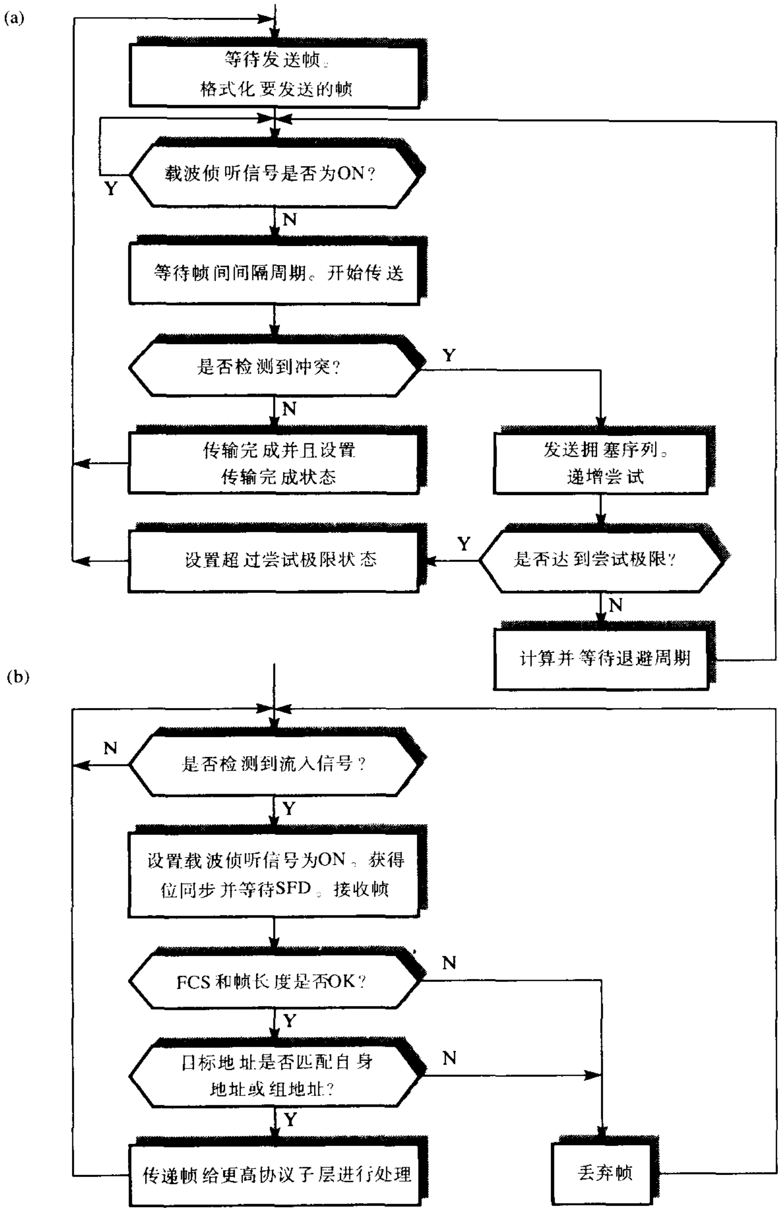


图6-11 CSMA/CD MAC子层操作
(a) 发送 (b) 接收

首先，检测到帧起始定界符时，剩余的前同步码位和帧起始定界符就一同被丢弃。接着处理目标地址字段来决定该DTE是否接收这个帧。如果接收，由目标地址、源地址和数据字段组成的帧内容被载入帧缓冲器中等待进一步处理。接收到的帧校验序列字段然后同在接收帧时由MAC单元计算的值比较。如果相等，含有接收帧的缓冲器开始地址以服务原语的形式被传送给较高协议层来进一步处理接收帧。在处理前还要对帧进行其他有效性校验，包括确

保帧含有整数个字节以及不会太短或太长。如果任何一种校验失败，这个帧就会被丢弃并给高层子层发送差错状态。我们会在6.5节讨论关于高层子层更多的细节。

最初，因冲突导致的传输二进制数据流会被每个活动DTE当作有效帧一样接收。在DTE检测到冲突并发送阻塞讯号后，它们停止传送。以这种方式接收到的帧碎片破坏了最小帧大小限制，由此会被接收DTE丢弃。另外，**最大帧长度**的采用意味着用于传送和接收的帧缓冲长度可以确定。帧校验序列字段是使用CRC-32（第3章讲到它有32次多项式的产生器）产生的32位序列。

6.2.2 令牌环

令牌环网络主要应用在科技和办公环境。图6-12解释了它的操作原理。无论什么时候一个DTE（站）想发送帧，它先等待令牌。接到令牌后，它开始帧的传输，帧头部包含接收者的地址。帧被环中所有的DTE转发（每个位被接收然后再被重新发送）直到它绕环一周返回到始发DTE，在那里帧从环中被去掉。除了帧的转发之外，帧的接收方拥有帧的一个副本并通过设置帧尾部的**响应位**来说明已经接收到帧了。

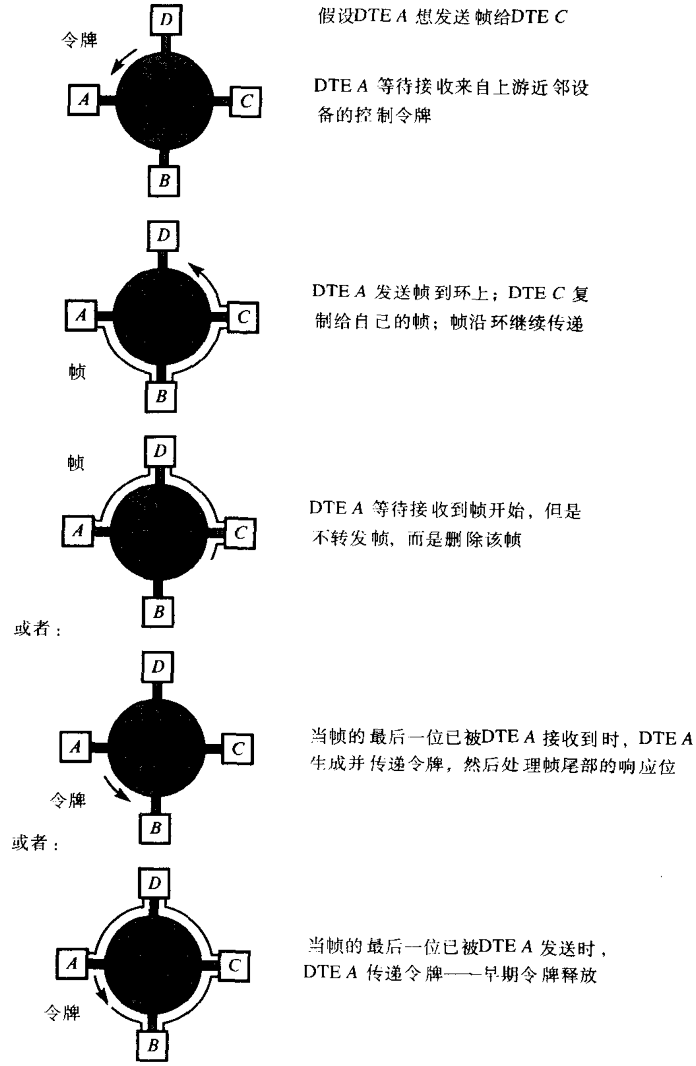


图6-12 令牌环网络、工作原理

DTE释放令牌有两种方式，由环的传输比特率（速率）来决定采用哪种方式。在较慢传输比特率的环（4Mbps）中，令牌在接收到响应位后释放。在较高传输比特率的环（16Mbps）中，令牌传送完帧的最后一位后就被释放。我们称之为**早期**（令牌）**释放**。

图6-13(a)给出了一个典型的令牌环网络，在(b)和(c)部分分别给出了连接DTE到电缆介质的各种必需组件。因为环中的每段都形成了点对点的连接，主干电缆通常是以4Mbps～16Mbps的传输比特率驱动传输的屏蔽双绞线。 292

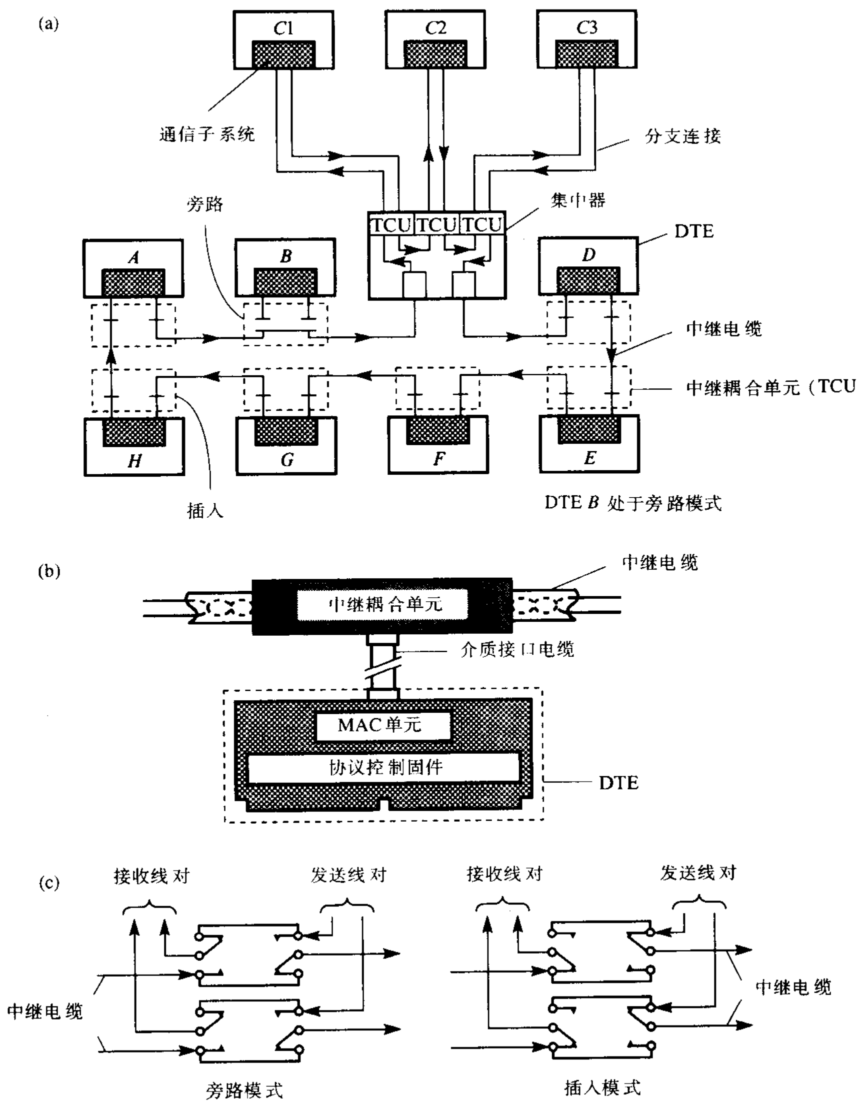


图6-13 令牌环网络组成部分

(a) 环结构 (b) DTE接口 (c) TCU示意图

在图6-13(b)中，DTE可以直接连到环上或者通过一个**集中器**（见图6-13(a)）。这个设备直接连在主干电缆上并提供给许多DTE直接分支连接器。集中器一般用来简化建筑内的布线。通常，它位于主干电缆进入（离开）办公室处。直接分支连接器用来把办公室中的每个DTE连接到集中器。它又被称为**线路集中器**，一个典型的安装会用到很多这种设备。 293

1. 环接口

中继耦合单元（TCU）构成了电缆传输介质的物理接口。它包含一组中继装置以及用于向电缆发信号或从电缆收信号的额外电子装置。中继装置是这样工作的，当DTE关闭时，TCU就处于**旁路状态**并且通过TCU继续维持传输路径。DTE加入到环中，由设备中通信控制
294 卡上的MAC单元控制。MAC单元通过激活TCU中的两对中继装置把DTE设备加入到环中。我们在图6-13(c)中看到，在加入以后，这种方式使得所有接收到的信号通过MAC单元发送。如果该DTE不是帧的首发者，在MAC单元中的收/发装置只是简单地读取和中继（转发）接收到的信号到发送方，如果该DTE发起了这次传输，就从环中去掉收到的信号。

以这种方式连接的两对中继装置的使用意味着MAC单元能从发送或接收信号线对检测到开路和短路故障。还有，在旁路状态下，MAC单元能进行自检，因为发送信号线对输出的任何数据都回送到接收信号线对上。DTE与TCU通过屏蔽电缆连接，这种电缆有两对双绞线：一对用于发送，一对用于接收。

MAC单元负责的功能包括：帧封装和帧拆卸，帧校验序列的生成和差错检测以及MAC算法的实施。当DTE作为活动的环监控站时（见“环管理”一节），它还提供用于数据编码和解码的环主时钟。每个传输二进制数据流都被活动环监控站进行（差分）曼彻斯特编码，然后环上的其他DTE使用DPLL电路对该二进制数据流进行锁频或锁相。另外，当DTE是活动的环监控站时，它确保环有**最小等待时间**。这个时间是这样计算的：一个信号以环的数据传输率沿着环传播一圈所需的位时间。包括在环传输介质上的传播时延加上经过每个MAC单元的传播时延之和。当没有DTE需要使用环时（就是说所有的DTE都是简单的转发模式），控制令牌沿着环持续传递，环必须有令牌序列中最少位数的最小等待时间来确保令牌没有被破坏。

所以当DTE是活动的令牌监控站（它的MAC单元提供固定的24位缓冲）时，令牌为24位长，它有效地成为确保各种情况下环正常工作的环部分。虽然环中的平均数据信号速率由活动监控站中的惟一主时钟控制，每个MAC单元中使用独立DPLL电路意味着实际的绕环信号速率会有微小的变化。最坏情况的变化是当最大数目的DTE（250）都是活动时，等效于加上或减去三个位。因此，除非环的等待时间保持不变，否则当等待时间减少时有些位会被破坏掉，而当等待时间增多时会有额外的位被加上去。为了保持恒定的环等待时间，6位长的额外**弹性（变量）缓冲**会被加到固定的24位缓冲上去。由此得到的30位缓冲初始化时只有27位。当主MAC单元收到的信号要快于主振荡器时，缓冲就会扩充一位，而当收到的信号较慢时，缓冲会相应地减少一位。这种情况下，环都有足够的位允许令牌沿着静态（空闲）状态下的
295 环持续传递。

2. 帧格式

令牌环中使用两种基本格式：一种为控制令牌，另一种为正常帧。控制令牌沿环循环，将传输权利（与正常的转发过程相反）从一个DTE传递到另一个DTE。而正常帧是DTE用来在环中发送数据或者MAC信息的。图6-14给出了两种帧的格式以及相应的每个字段的位序列。

起始定界符字段（SD）和结束定界符字段（ED）是用来实现数据透明传输的特殊位序列。它们利用在电缆传输介质上使用的符号编码方式：除了起始定界符字段和结束定界符字段中的J、K位外，其他在介质上传输的信息位都采用曼彻斯特编码。相比之下，符号J和K不同于通常的编码规则，而是用作代表整个位单元周期的保留固定电平。符号J与前一位有相同的极性而符号K与前一位极性相反。这种方式下，接收方能可靠地判断每个传输令牌或帧的开始和结束，而不管它们的内容或长度。应该注意，在ED字段中仅有6位（JK1JK1，见图6-14(c)），

用来说明帧的有效结束。其他的两位I和E有其他的作用： 296

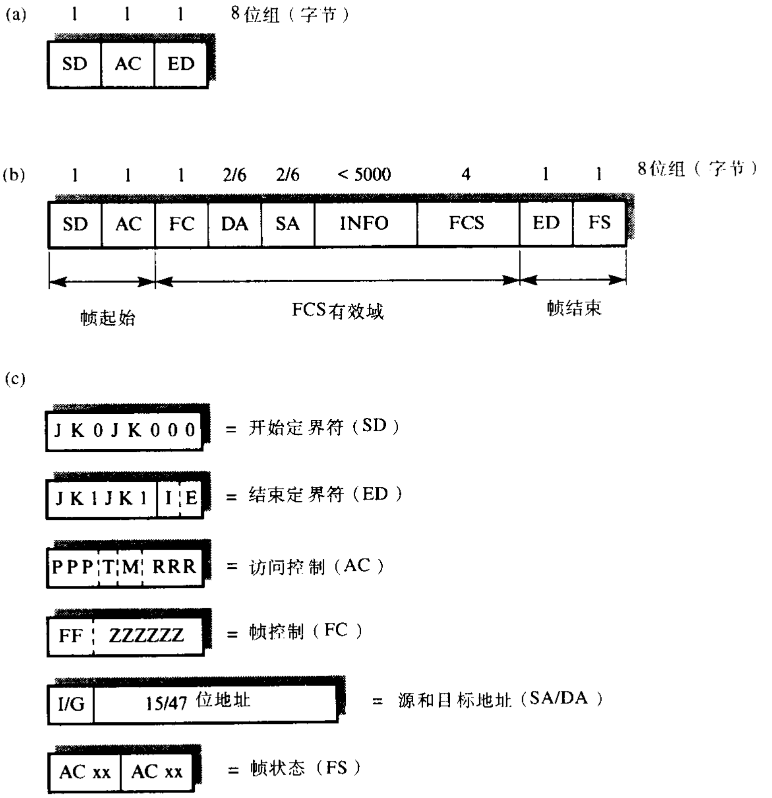


图6-14 令牌环网络帧格式和字段说明

(a) 令牌格式 (b) 帧格式 (c) 字段描述

- 在令牌中，I和E位都是0。
- 在一个正常帧中，I位用来说明：如果该帧是帧序列中的第一个帧（或是中间帧）则I=1；如果该帧是帧序列中的最后一个帧（或惟一帧）则I=0。
- E位用于差错检测。源DTE将其置为0，但是如果任何DTE在接收或者转发该帧时检测到差错（如帧校验序列差错），则把E位置为1，告知源DTE已经检测到差错。

访问控制（AC）字段由优先级位、令牌位和监控位，以及保留位组成。从名称上就能看出，访问控制字段是用来控制对环的访问的。当它是令牌的部分时，优先级位（P）就说明令牌的优先级，由此设备接到该令牌时知道应该发送哪些帧。令牌位（T）用来区别令牌和正常帧（0表示令牌，1表示帧）。监控位（M）用于活动监控站以防止帧绕环连续循环。最后，保留位（R）允许DTE维持高优先级帧而向转发的帧或令牌请求下一个发送的令牌时，具有必需的优先级。

帧控制（FC）字段定义了帧的类型（MAC帧或信息帧）和一些特定控制功能。如果帧类型位（F）指明是MAC帧，则环上所有的DTE都会译出控制位，并在必要时按控制位（Z）操作。如果是信息帧，则由目标地址字段定义的DTE译出控制位。

源设备地址（SA）和目标设备地址（DA）字段可以是16位长也可以是48位长，但是对于任何特定的局域网来说所有DTE都是一样长的。目标设备地址字段指明了要接收帧的DTE。

字段的第一位为0，指明是单地址；为1，指明是组地址。单地址定义了环上的某个DTE，而组地址用来发送帧给多个目标DTE。源设备地址永远是单地址，它定义了发送帧的DTE。另外，全1的目标设备地址是广播地址，它指明帧要发送给环上的所有DTE。

信息（INFO）字段用于携带用户数据，或者包含在MAC帧中传送的控制信息。虽然没有给信息字段设置长度限制，但是实际上它由DTE掌握控制令牌后允许发送帧的最大时间所限制。一个典型的最大长度是5000个字节。

帧校验序列（FCS）字段是32位的CRC。最后，帧状态（FS）字段由两个字段组成：地址识别位（A）和帧复制位（C）。它们由产生帧的DTE置为0。如果该帧被环上的一个或多个DTE识别，该DTE会置地址识别位（A）为1。同样，如果复制该帧，DTE会把帧复制位（C）置为1。这样，发送帧的初始DTE会知道目标设备不存在或关掉了，DTE存在但没有复制帧，
297 或者DTE存在并复制帧。

3. 帧传输

收到发送数据信息（包括有关数据优先级的参数）的服务请求，MAC单元先把数据封装成如图6-14所示的标准格式。MAC单元等待接收优先级小于或者等于该帧优先级的令牌。显然，如果一个系统应用多级优先级，它必须遵守一个处理规程来确保所有DTE以正确的优先级次序发送帧。该过程如下。

按格式要求封装完帧之后，在接收相应令牌（令牌环的优先级小于或等于等待发送帧的优先级）之前，每次在环接口上转发拥有更高优先级的帧或者令牌，MAC单元会读取其访问控制字段（AC）中保留位的值。如果它们等于或大于等待帧的优先级，保留位会不改变地被简单转发。如果小于，MAC单元就会用等待帧的优先级替换保留位的当前值。然后，假设环中没有拥有更高优先级的帧等待发送，则令牌当前的持有者（使用者）发送完帧后就会以这个优先级把令牌传下去。等待的MAC单元接到令牌后，会检测令牌的优先级是否等于等待发送帧的优先级。如果是的话，MAC单元就通过在转发访问控制字段中的令牌位之前把该值变为1接收该令牌，这样就有效地把令牌转为正常帧的开始帧序列。然后MAC单元就停止转发接下来收到的信号，而是把预先格式化的数据加在转换后的帧开始序列后面。当帧内容被发送的时候，FCS会计算，并且随后在发送帧结束序列之前加到帧内容后面。

一旦开始发送等待帧，MAC单元会停止转发，而是把绕环传输一周的发送帧去掉。另外，MAC单元会注意帧尾部帧状态字段的A位和C位来确定帧是否被复制或者忽略。然后它会产生一个新的令牌发送到环上，允许另一个等待DTE得到对环的访问权。首先，如果其他等待帧的优先级大于或等于令牌的优先级，其次，发送其他帧的时间总和在称为令牌持有时间（默认设置是10ms）的规定时间范围内，那么可能多于一个帧被发送。图6-15显示了帧传输和帧接收的流程图。

4. 帧接收

除了转发收到的信号（二进制数据）流，位于环中每个活动DTE的MAC单元还通过识
298 别特殊的帧开始位序列来检测每个帧的开始。然后它决定是否应该简单地转发或者复制该帧。如果F位表示它是一个MAC信息帧（见“环管理”一节），就会复制该帧，解释如果必要C位会作用于该帧。但是如果只是一个携带数据的正常帧，目标地址与DTE的单地址或相关群地址匹配，帧内容会被复制到帧缓冲中，继续下一步处理。任何一种情况下，帧尾部的帧状态字段中的A位和C位都会在转发前根据情况相应地被设置。图6-15(b)给出了接收机制
299 的流程表。

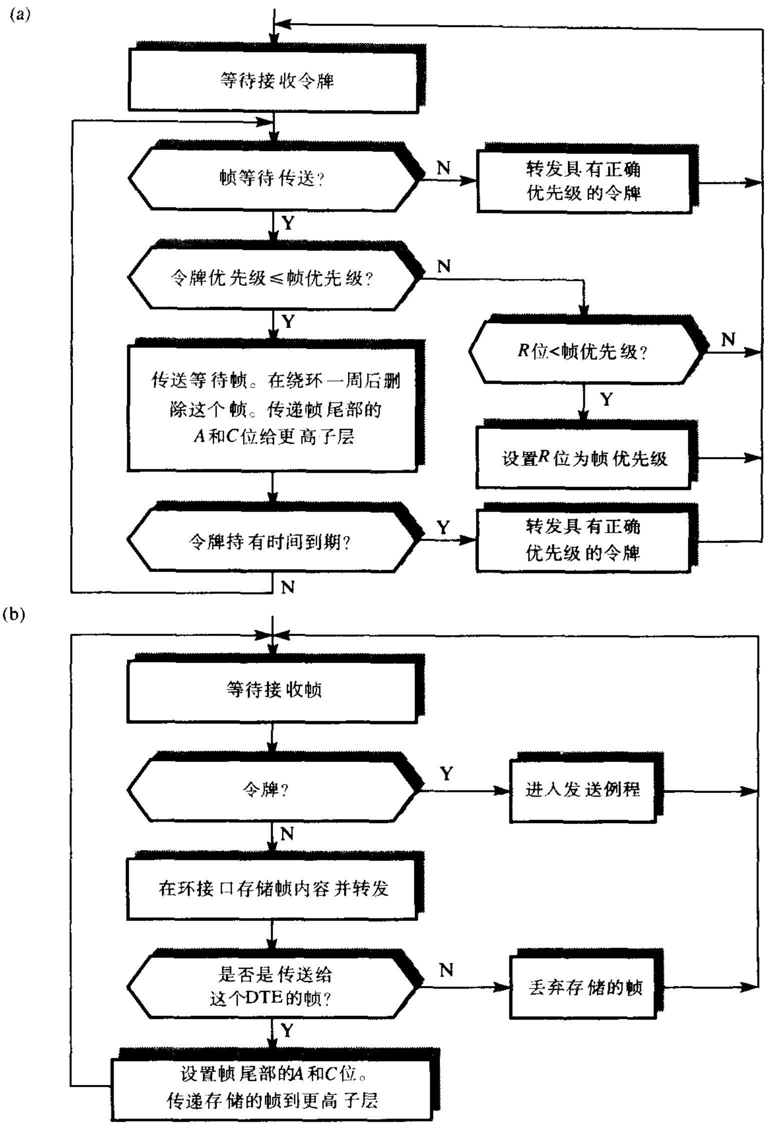


图6-15 令牌环MAC子层工作机制

(a) 传送 (b) 接收

5. 优先级操作

MAC单元在发送完所有等待帧后，分配给令牌的优先级由确保以下两点的机制决定：

1）比当前环服务优先级更高的帧总是先在环上传送。

2）所有拥有相同优先级帧的DTE对环拥有相等的访问权限。

这是通过每个帧的AC字段中的P位和R位连同一个机制共同实现的。该机制确保提高环服务优先级的DTE在更高优先级帧被发送完以后，绕环一周返回原来的等级。

为了完成这个方案，每个MAC单元使用两组值。第一组由Pm、Pr和Rr三个变量组成。Pm指明了当前在DTE中等待传送帧的最高优先级。Pr和Rr称为**优先级寄存器**，分别含有最近被转发的令牌或帧的AC位字段中存在的优先级值和保留值。第二组由称为Sr和Sx的两个堆栈

组成。用途如下：

接到可用的令牌后，DTE发送的所有帧都被指派为：AC字段中的优先级值等于当前的环服务优先级Pr，而保留值为0。在所有等于或大于当前环优先级的等待帧传输完后，或者在令牌持有时间到期前还没有传输完帧的情况下，MAC单元就会产生一个新令牌：

（1）P=Pr，R=Max（Rr，Pm）

如果DTE没有优先级（在寄存器Pm）等于或大于当前环服务优先级（在寄存器Pr）的等待帧，或者没有大于当前优先级的保留请求（在寄存器Rr）

（2）P=Max（Rr，Pm），R=0

如果DTE有大于当前优先级Pr的等待帧（在寄存器Pm），或者当前Rr的内容大于当前优先级

后一种情况中，DTE有效地提高了环的服务优先级等级，从而成为**堆栈站**（DTE），以前的环服务优先级（Pr）值存在堆栈Sr中，而新的环服务优先级（P）存在堆栈Sx中。这些值被保存下来，因为在环中任何点上都没有等于或大于位于堆栈Sx上的优先级值P的帧准备发送，DTE有责任作为堆栈站用这些值来降低环服务优先级等级。同样，使用一个堆栈而不是单个
300 寄存器，是因为堆栈站在服务优先级降到更低的级别前需要不止一次提高环服务优先级的等级。图6-16(a)显示了令牌的P位和R位被分配不同的值以及在两个堆栈执行的动作。

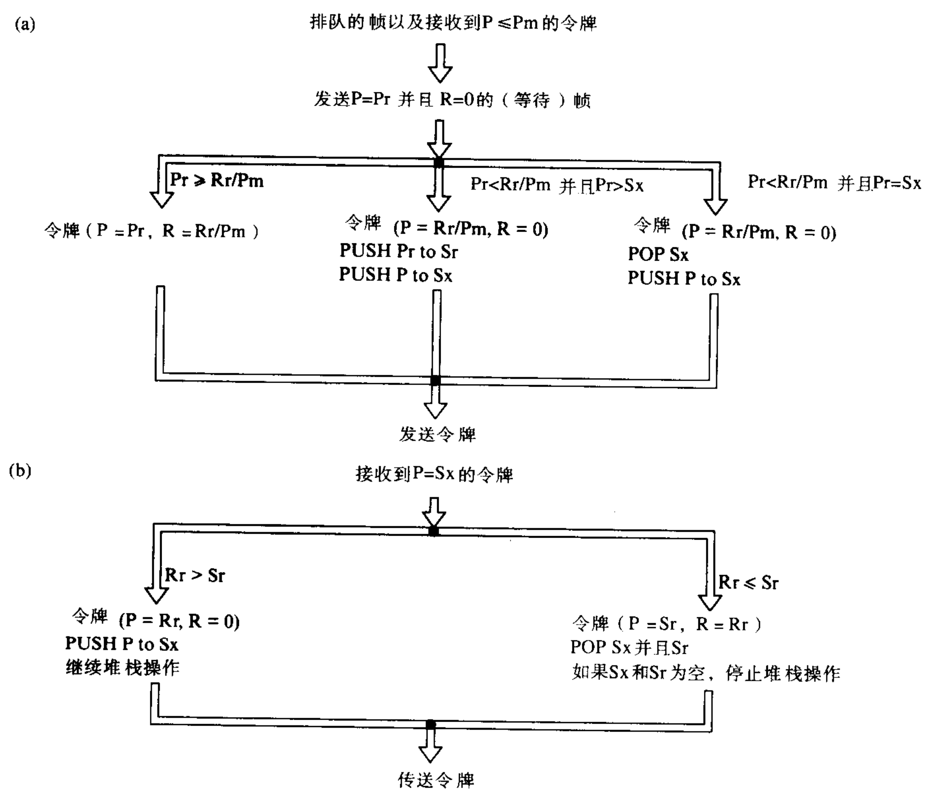


图6-16　令牌生成及栈更改

(a) 令牌生成（注意：如果栈为空，那么Sx=0）　(b) 栈更改

在成为堆栈站之后，MAC单元声明它收到的优先级等于堆栈Sx上优先级的每个令牌。检查AC字段的R位的值来决定环的服务优先级是提高等级，保持等级，还是降低等级。然后新

的令牌以下列方式发送：

（1）P=Rr，R=0

如果R位的值（寄存器Rr的当前值）大于Sr。新的环服务优先级（P）被压入堆栈Sx，DTE继续充当堆栈站。

（2）P=Sr，R=Rr（不变）

如果R位的值（寄存器Rr的当前值）小于或等于Sr。当前在堆栈Sx和Sr栈顶的值都被弹出，如果两个堆栈都为空，DTE就不再起堆栈站的作用。这两种操作如图6-16（b）所示。 301

实例6-1

一个令牌环网络配置以四个优先级等级工作：0、2、4、8，8是最高优先级。在令牌绕环一圈而没有帧传送的状态后，四个站产生了要发送的帧，如下：

站1　优先级为2的1个帧

站7　优先级为2的1个帧

站15　优先级为4的1个帧

站17　优先级为4的1个帧

假设在环中站以数字的顺序排列，以及站1接收到优先级字段和保留字段都为0的令牌，以表的形式给出了接下来令牌绕环8周过程中每个站所进行的传输情况。表6-1中还包括了每次生成的新令牌和每次绕环传输的帧的优先级字段和保留字段。同样还包括堆栈站所作的动作。

解：

表6-1给出了令牌在接下来绕环8周中每个站所进行的传输情况。

表6-1　令牌环优先级机制的例子

循环	站1 *Pm = 2* F/T (P,R)	站7 *Pm = 2* F/T (P,R)	站15 *Pm = 4* F/T (P,R)	站17 *Pm = 4* F/T (P,R)
0	T(0,0)	T(0,0)	T(0,0)	T(0,0)
1	T(0,0) Pr = 0, Rr = 0, Pm = 2 Pm > Pr F(0,0) P = Pr = 0; Rr = 0	F(0,0) Pr = 0, Rr = 0, Pm = 2 Pm > Rr F(0,2) P = Pr = 0; R = Pm = 2	F(0,2) Pr = 0, Rr = 2, Pm = 4 Pm > Pr F(0,4) P = Pr = 0; R = Pm = 4	F(0,4) Pr = 0, Rr = 4, Pm = 4 Pm = Rr F(0,4) P = Pr = 0; R = Rr = 4
2	F(0,4) Pr = 0, Rr = 4 Rr > Pr T(4,0) P = Rr = 4; R = 0 堆栈操作 Sr = 0, Sx = 4	T(4,0) Pr = 4, Rr = 0, Pm = 2 Pr > Pm > Rr T(4,2) P = Pr; R = Pm = 2	T(4,2) Pr = 4, Rr = 2, Pm = 4 Pm = Pr F(4,0) P = Pr = 4; R = 0	F(4,0) Pr = 4, Rr = 0, Pm = 4 Pm > Rr F(4,4) P = Pr; R = Pm = 4
3	F(4,4) 无帧发送 F(4,4) Sr = 0, Sx = 4	F(4,4) Pr = 4, Rr = 4, Pm = 2 Pm < Rr F(4,4) P = Pr; R = Rr	F(4,4) Pr = 4, Rr = 4 Rr = Pr T(4,4) P = Pr = 4; R = Rr = 4	T(4,4) Pr = 4, Rr = 4, Pm = 4 Pm = Pr F(4,0) P = Pr; R = 0
4	F(4,0) 无帧发送 F(4,0) Sr = 0, Sx = 4	F(4,0) Pr = 4, Rr = 0, Pm = 2 Pm > Rr F(4,2) P = Pr; R = Pm	F(4,2) 无帧发送 F(4,2)	F(4,2) Pr = 4, Rr = 2 Rr < Pr T(4,2) P = Pr; R = Rr

（续）

循环	站1 *Pm* = 2 F/T (P,R)	站7 *Pm* = 2 F/T (P,R)	站15 *Pm* = 4 F/T (P,R)	站17 *Pm* = 4 F/T (P,R)
5	T(4,2) Pr = 4, Rr = 2, Sr = 0, Sx = 4 Pr = Sx, Rr > Sr T(2,0) P = Pr = 2; R = 0 - - - - - - - - - - - - - - - - - - - Sr = 0, Sx = 2	T(2,0) Pr = 2, Rr = 0, Pm = 2 Pm = Pr F(2,0) P = Pr; R = Rr	F(2,0) 无帧发送 F(2,0)	F(2,0) 无帧发送 F(2,0)
6	F(2,0) 无帧发送 F(2,0) - - - - - - - - - - - - - - - - - - - Sr = 0, Sx = 2	F(2,0) Pr = 2, Rr = 0 Rr < Pr T(2,0) P = Pr; R = Rr	T(2,0) 无帧发送 T(2,0)	T(2,0) 无帧发送 T(2,0)
7	T(2,0) Pr = 2, Rr = 0, Sr = 0, Sx = 2 Pr = Sx, Rr ≤ Sr T(0,0) P = Sr; R = Rr - - - - - - - - - - - - - - - - - - - 停止堆栈操作	T(0,0) 无帧发送 T(0,0)	T(0,0) 无帧发送 T(0,0)	T(0,0) 无帧发送 T(0,0)
8	T(0,0)	T(0,0)	T(0,0)	T(0,0)

在令牌绕环的第一周中，站1得到令牌并开始传输等待帧。同样在这一周中，帧中的保留字段被站7提高到2，接着被站15提高到4。

在第二周中，站1读取帧的保留字段并决定以优先级4释放令牌。既然它提高了环的优先级，它就成为堆栈站。然后令牌继续绕着环传递最后被站15得到。同样在这一周中站17把保留字段从0提高到4。

在第三周中，站15以优先级4和保留值4释放令牌。站17得到令牌开始等待帧的传输。

在第四周中，站7把保留字段从0更新成2，这使得站17以同样的优先级但保留值为2释放令牌。

在第五周中，作为堆栈站的站1检测到Rr大于Sr，就把环/令牌的优先级从4降低到2，并把较低的优先级压入堆栈中。接着站7就能传输等待帧了。

在第六周中，既然保留值为0，站7就以相同的优先级释放令牌。

302～303 在第七周中，站1检测到令牌中的保留字段小于优先级字段，就把优先级降为0并停止充当堆栈站。这样令牌就恢复到了初始状态，继续绕环传递直到环中又有新的帧产生并要求发送。

6. 环管理

前面我们主要讨论了在环正常工作状态下的帧和令牌的传输。然而，在环正常工作之前必须先建立环。如果某个DTE想加入已经工作的环，它必须先经过一个初始化过程以确保不会干扰已建立环的正常运作。另外，在正常工作状态，环中每个活动DTE必须不断监控它的正确工作，如果有故障发生能采取正确的措施尝试重新建立一个正确工作的环。总之，这些功能都称为**环管理**。图6-17列出了跟这些功能有关的各种MAC帧类型。

初始化　当某个DTE在开启或者重置后想成为环的一部分时，它进入初始化规程以确保环中没有其他DTE拥有相同的地址并通知下游近邻DTE它已经加入（重新加入）了环。

初始化规程开始于DTE发送帧状态字段A位置为0的重复地址测试（DAT）MAC帧。环中

每个活动DTE收到这个帧后会检查DA字段，如果它确定DA字段同自己的地址一致就置A位为1。因此如果当DAT帧回到发送它的DTE时A位是1，就通知网络管理子层，并回到旁路状态。网络管理子层（见14.4.3节）接着决定是否重试加入环。如果当DAT帧回到发送它的DTE时A位是0，DTE就发送当前备用监控（SMP）MAC帧继续初始化规程。

帧 类 型	功 能
重复地址测试（DAT）	用于初始化规程，以确定环上不存在使用同样地址的其他站
当前备用监控（SMP）	用于初始化规程，以确定环中上游近邻（后继）地址
当前活动监控（AMP）	由当前活动的监控站以固定间隔发送这类帧，每个站监听这类帧的传递
申请令牌（CT）	如果当前活动监控站出故障，使用该帧确定一个新的活动监控站
清除（PRG）	由新的活动监控站使用，初始化所有站进入空闲状态
告警（BCN）	在告警规程中使用该帧

图6-17 用于令牌环管理的MAC帧类型

接到收到A位和C位都为0的SMP帧的DTE会认为该帧是上游近邻DTE产生的，接着把SA更新成上游近邻DTE地址（UNA）。UNA用于故障检测和监控功能。初始化阶段进行到这里就完成了。 304

备用监控 在完成初始化规程后，DTE开始发送和接收正常帧和令牌。另外，DTE进入备用监控状态，不断监控环的正常运行。它通过监控令牌的字段以及当前活动监控站定期发送当前活动监控（AMP）MAC帧，在环接口重复操作。如果令牌或AMP帧没有被定期地检测到，备用监控站超时（对这个功能它保存了两个计时器）并进入申请令牌状态。

在申请令牌状态，DTE连续发送申请令牌（CT）MAC帧并检查收到的任何CT帧中的SA字段。每个CT帧除了含有始发DTE源地址（SA）外还含有它所存的UNA。如果收到的CT帧的SA字段和自己的地址一致，并且UNA与它的UNA也一致，说明CT帧成功地绕环传递一周。接下来该DTE成为新的活动环监控站。相反，如果收到的CT帧的源设备地址（SA）大于本设备地址，说明另一个DTE较早地努力成为新监控站。在这种情况下，DTE有效地放弃了它的努力，并回到备用监控状态。

活动监控 如果DTE通过努力成功地成为新的活动监控站，它先向环插入一个等待缓冲，并激活自己的时钟（注意任一时刻环中只有一个活动监控站）。然后它开始发送清除（PRG）MAC帧以确保在传送一个新令牌以前环中没有其他的帧或令牌。当DTE接到PRG帧时发现所含的SA同自身地址一致，这说明环已经成功地清除完毕。该DTE通过广播AMP MAC帧来启动近邻设备通知规程。经过短暂的时延后就开始新控制令牌的传递。

活动监控站的下游近邻DTE检测到AMP帧中的A位都是0，因此从帧中读取UNA并用它更新现有的UNA变量。它把A位和C位都设成1然后转发该帧。环中的后续DTE检测到A位不是0，只复位AMP计时器，记录AMP帧的传递情况。

另外，活动监控站的下游近邻DTE在转发完AMP帧后继续通过广播一个类似SMP帧来进行邻近设备通知规程。依次地，下一个下游DTE检测该帧中的A位是0就更新UNA变量并把A位和C位都设成1然后转发该帧。它通过继续广播A位置为0的新SMP帧进行邻近设备通知规程。环中的每一个DTE都执行这个规程，随后活动监控站会隔固定时间间隔发送新AMP帧来重新开始。这样，环中每个活动DTE能检测到诸如DTE逾限（比如持续发送令牌）的故障：环中AMP帧流的消失意味着其他所有DTE中的AMP定时器超时，这样就开始CT帧的传输，如果故

305 障仍然存在，所有DTE进入称为告警的故障诊断规程。

告警　如果环中出现电缆断裂的严重故障，称为告警的规程通知环中的每一个DTE暂停令牌传送（直到受影响的故障范围被确定并修复）。故障范围如下：

- 报告故障的DTE，我们称之为**告警站**
- 告警站的上游DTE
- 它们之间的环传输介质

作为例子，图6-18(a)显示了假定DTE *F*和DTE *G*之间的电缆出现断路故障。在这个例子中，*G*是告警站而*F*就是上游近邻设备。通常，如果与AMP或者令牌传递规程相关的定时器到时就进入告警状态。当处于这个状态时，就连续地发送告警（BCN）监督帧，直到告警帧被接收或者定时器到时。如果后者发生，就通知网络管理子层，停止发送。另一种情况，如果告警帧被某一个DTE接收，并且SA同该DTE的地址一致，故障就假定被清除，该DTE进入申请令牌状态。或者如果告警帧被一个地址与SA不一致的DTE接收，该DTE就进入备用监控状态。

如果网络只是由一个环组成，在发生故障时，只有在故障段修复后网络才能恢复传输。
306 令牌环的一个可选特征是使用第二个冗余环，该冗余环传输方向和第一个环相反，图6-18(b)显示了网络的结构情况。

在这样的网络中，TCU不仅支持前面所述的功能而且能越过故障段或DTE。作为例子，图6-18(c)说明了图6-18(a)所介绍的故障段（故障范围）如何被越过。一般来讲，一旦故障范围被确定并报告，F和G的TCU中的中继装置会被激活，（有希望）重新建立一个连续环。如果隔离了可能的故障段但没有消除故障，下一步就开始完全隔离DTE G，正像图6-18(d)所示。注意在这些图中，冗余环并没有直接的通道连到MAC单元，而只是简单地提供了越过环中某个部分的一种方法。重新建立的环中的DTE顺序同原先的环一致。

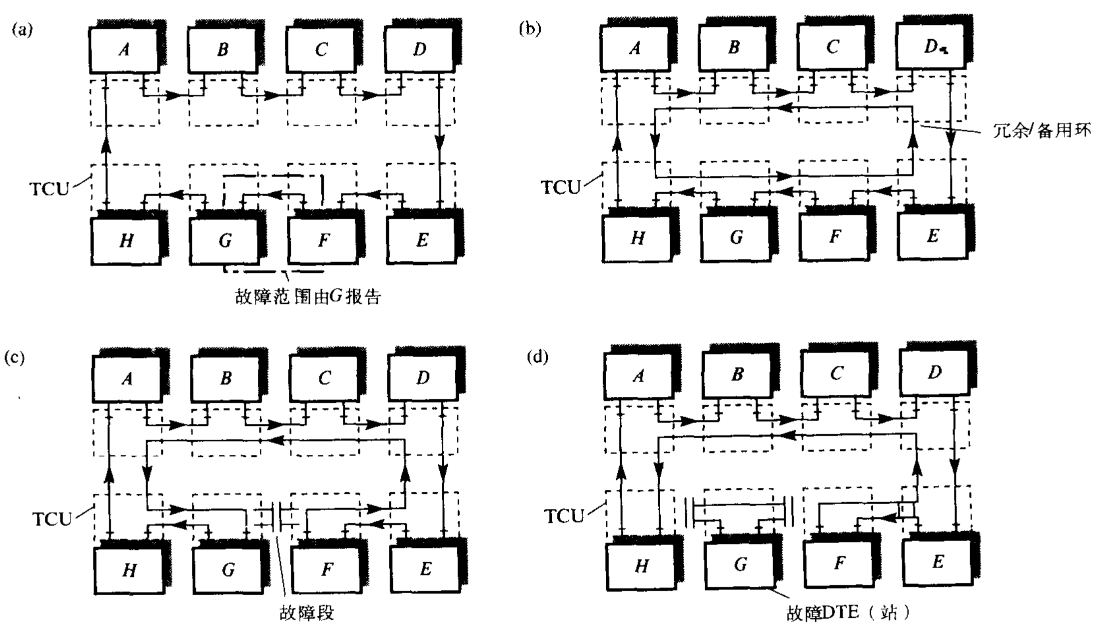


图6-18　环故障检测和隔离

(a) 故障检测　(b) 冗余环结构　(c) 段隔离　(d) DTE（站）隔离

我们可以看到，应用在令牌环网络中的MAC规程比CSMA/CD总线网络复杂。但是，记住大多数规程用在位于MAC单元的控制器集成电路。因此它们的操作对于用户是透明的。而且，许多这样的环管理规程只在故障发生时调用，因此它们的开销大体上相当。

6.2.3 令牌总线

标准文档支持的第三种局域网是**令牌总线**网络。因为令牌MAC方式的确定性特点以及对传输帧分优先级的能力，令牌总线网络被应用在制造行业（如工厂自动化）以及其他相关领域，诸如进程控制行业。在正常无差错情况下，这种网络的工作机制类似于令牌环网络，但是因为两种介质的访问方法不同（对总线是广播，对环是顺序），所以处理逻辑环管理的诸如初始化和令牌丢失的规程不同。为了避免重复，我们主要讨论与令牌总线网络相关的管理规程。

图6-19说明了跟令牌总线网络相关的操作及其组成部分。令牌总线网络通常采用同轴电缆作为传输介质，并以宽带模式或改进的基带模式（称为**载波带**）进行操作。图6-19(a)所示的调制和接口控制电路执行如下功能：

- 发送编码的数据（调制）
- 接收解码的数据（解调）
- 时钟生成

在物理接口模块（PIM）和附接的DTE之间有一个标准接口。在一些情况下，PIM被集成在DTE的通信板上。 307

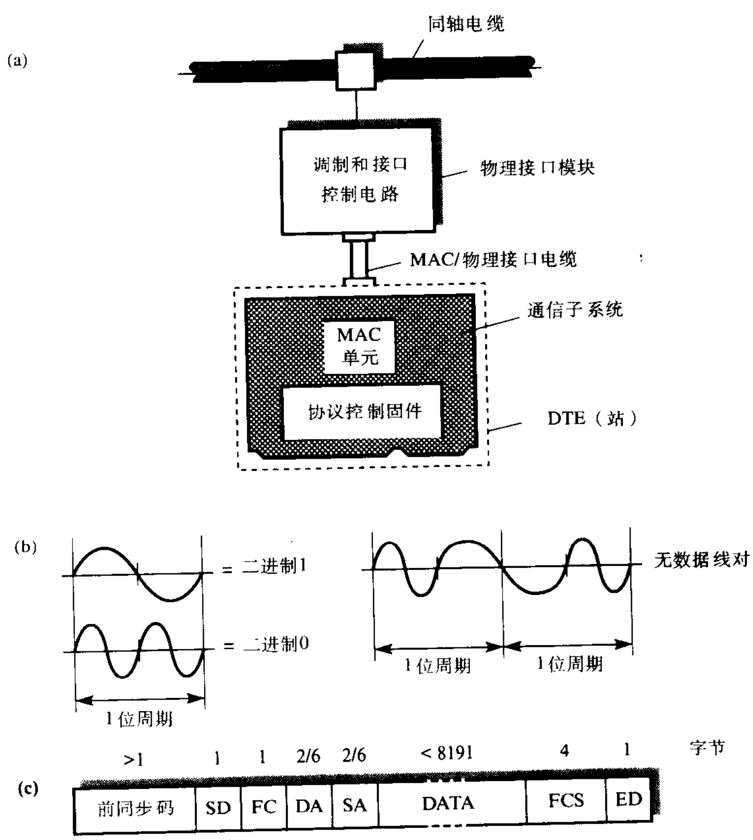


图6-19 令牌总线网络原理

(a) DTE接口示意图 (b) 载波带编码 (c) 帧格式

图6-19(b)说明了载波带的操作原理。虽然载波带模式同基带模式一样，每个传输占用了整个电缆带宽，但是在载波带模式中，所有数据在传输前先采用相位相干FSK方式调制。正如我们所看到的，一个二进制1以等比特率（通常是1Mbps ~ 5Mbps）的频率的正弦信号的一个周期传输，而二进制0以两倍比特率的正弦信号的两个周期传输。同样注意，在位单元边界没有相位变化，因此我们称之为相位相干。

回忆一下第2章中讲到，电缆中拾取的任何外来噪声信号是由无限个频率分量组成。一个基带信号（波形）同样由可能的无限个频率分量组成。相比之下，载波带波形只有两个频率分量。因此，可能在接收方（它只通过这两个频率）使用过滤装置有效地阻止大多数噪声信号，并大大提高系统的抗噪声能力。在基带模式中不能做到，因为过滤装置同样会影响数据
308 信号。

令牌总线网络的帧格式如图6-19(c)所示。它基本上和令牌环网络的帧格式相同，只是J和K非数据位（它们用在令牌环的SD和ED字段中来获得数据的透明性），在载波带模式中被特殊非数据符号对所代替。

1. 基本操作

图6-20说明了令牌总线网络的基本操作。只有单个控制令牌，并且只有令牌的拥有者能发送帧，所有能发起帧传输的DTE被连成**逻辑环**，令牌沿着逻辑环的物理总线传递。这样从环中的前驱站（上游近邻设备）处接收令牌，DTE可按规定最大值发送任何等待帧，然后它把令牌传递给环中的后继站（下游近邻设备）。

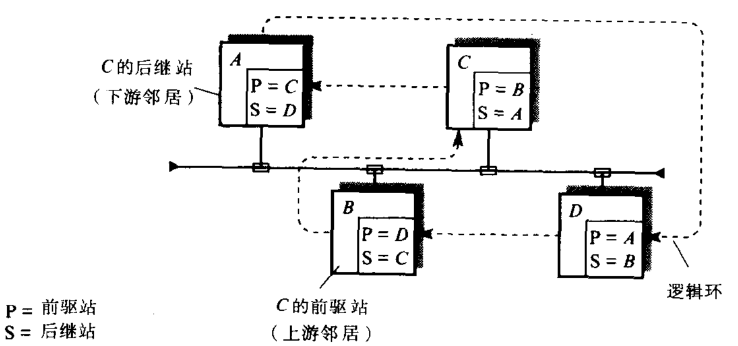


图6-20　令牌总线网络工作原理

在我们描述各种环管理规程之前，让我们重新讲一下总线网络的两种基本性质。首先，在总线网络中所有的DTE直接连到传输介质上，因此，当一个DTE在介质上传输（广播）一个帧时，它会被网络上的所有活动DTE接收（侦听）到。其次，DTE在假定传输帧被破坏或指定目标DTE不宜操作前，有一个最大时间用于等待响应。这个时间称为时隙间隔（它不同于CSMA/CD总线网络中的时隙间隔）并定义如下：

$$时隙间隔 = 2 \times 传输路径时延 + 处理时延$$

这里传输路径时延指网络中从任何发送方到任何接收方的最坏情况的传输时延，而处理时延指DTE的MAC单元处理接收到的帧并产生一个适当响应的最大时间，然后加上安全边际，这个时隙间隔以位周期表示，以字节为单位。

309 在正常操作下，令牌使用短令牌帧从逻辑环中一个DTE传递到另一个DTE。因此每个DTE只需要知道逻辑环中下一个DTE（下游近邻设备）的地址。如果某个DTE接收令牌失败，

发送DTE使用一系列恢复规程来找到新的后继者，如果该DTE获得近邻设备的响应，这些规程就会逐渐加强。其他规程包括环的初始化以及当DTE加入和离开环时维持环的正确操作。虽然可能像令牌环一样给令牌分优先级，但是我们开始只考虑单一优先级。图6-21说明了用于各种环管理规程的MAC帧格式以及它们应用的简要介绍。我们会在讨论各种规程时给出更详细的说明。

帧 类 型	功 能
申请令牌	在初始化（逻辑）环次序时使用
征求后继者	当一个站离开环时和允许另一个站（重新）进入环时使用
谁后继	一个站确定它的后继地址时使用
解决争用	允许新站加入环时使用
设置后继	允许加入环的新站通知连在环上的新前驱站时使用
令牌	控制令牌帧

图6-21 用于令牌总线环管理的MAC帧类型

2. 令牌传递

接收到有效的令牌帧后，DTE就可以发送任何等待的帧。然后它把令牌传递给它的后继者。在发送令牌时，该DTE会侦听总线上任何活动，以确保后继站是否活动并且接收到令牌。如果侦听到有效帧传输，它就假定一切正常，后继者正确地接收到令牌。经过时隙间隔后，如果没有侦听到有效帧传输，它必须采取校正措施。

在发送令牌后，如果该DTE听到噪声脉冲或具有不正确FCS的帧，它就继续侦听四个时隙间隔。如果没有侦听到任何活动，该DTE就假定令牌在传输中产生差错，并重发令牌。如果在四个时隙间隔中侦听到有效帧，该DTE就假定后继站得到了令牌。如果在这个时间内听到第二个噪声脉冲，该DTE认为这是后继站发送的有效帧，并假定令牌已经被传递。

在重复实施令牌传递和监听规程后，如果后继者没有响应第二个令牌帧，该DTE就假定后继站失效，因此着手建立一个新的后继站。发送方先广播一个数据字段含有当前后继地址的**谁后继帧**。接收到这种帧，每个DTE把帧中数据字段的地址与自己的前驱地址（通常发送令牌给它的DTE地址）比较，如果一致则发送一个含有自己地址的**设置后继站帧**作为响应。持有令牌的DTE就建立了一个新的后继者并以此隔离失效的DTE（原先的后继者）。 310

如果发送DTE没有收到“谁后继帧”的响应，就第二次重发这个帧。如果仍然没有响应，它就采取更进一步措施，发送在DA字段中含有自身地址的**征求后继站帧**。这要求网络中的任何DTE都响应它。如果任何正常工作的DTE听到这个帧，它们响应并使用称为**响应窗口**的规程来重新建立逻辑环。如果没有收到响应，该DTE假设出现严重故障，比如其他所有DTE都不工作，传输介质断路或者该DTE的接收装置有了故障（因此无法听到来自其他DTE对于发出请求的响应）。在这种情况下，该DTE静待但继续监听另一个DTE的发送。

3. 响应窗口

这个规程在随机时间间隔后发生，允许新DTE加入一个正常运行的逻辑环。响应窗口是一个时间间隔，在这个时间间隔中DTE需要在发送一个帧后等待响应，所需时间长度与网络时隙间隔相同。每个DTE发送的“征求后继站帧”规定了SA和DA。地址处在这个范围内，并希望进入逻辑环的DTE可响应该帧。当每个DTE是令牌持有者时会在随机时间间隔发送“征求后继站帧”。

当DTE发送了“征求后继站帧”，它就打开了响应窗口，因为在发送这种帧后，发送DTE就会在响应窗口时期内等待响应。如果一个拥有“征求后继站帧”所指定的地址范围内地址的DTE想加入环，它通过向帧发送者发送请求响应来成为逻辑环中新的后继者。如果发送方听到响应（称为“设置后继站帧”），它通过使新DTE成为它的新后继者加入环，然后把令牌传递给它。显然，规定的地址范围可能含有多个DTE，它们都等待加入环，这种情况下，每个DTE返回的响应帧会相互影响。如果发生这种情况，征求DTE必须通过进入一个仲裁规程来确定单一响应者，它如下工作。

确定在指定地址范围内有多于一个DTE等待加入环，征求DTE会开始通过发送**“解决争用帧”**给它们排序。这个规程继续直到该DTE收到一个肯定的应答。任何一个响应较早征求
311 后继站帧，但随后没有收到令牌的DTE，就会选一个0～3的值，然后在这个时隙间隔内侦听总线上任何进一步的活动。如果DTE在所选的时间周期内听到传送活动，它就延迟请求并等待下一次机会（当下一个响应窗口被开启）成为环的一部分。如果它在所选时间周期内没有听到传送活动，它就继续等待解决争用帧的接收。这样，最坏情况下征求DTE需要花费解决争用的时延被限定了。

4. *初始化*

初始化规程建立在响应窗口规程基础上。网络上每个DTE监听总线上的所有传输，无论何时它听到一个传输就重置**无效计时器**并预置一个值。如果DTE在正常工作下丢失令牌，无效计时器就会到时并且该DTE进入初始化阶段，在这个阶段中它发送**“申请令牌帧”**。如前所述，可能许多DTE同时发送“申请令牌帧”，因此下面的规程确保只产生一个令牌。

每个可能的初始化者发送“申请令牌帧”，其信息字段长度为时隙间隔的整数倍。该整数可以是0、2、4或6，选哪个整数取决于该DTE地址的前两位。在发送了“申请令牌帧”后，等待一个时隙间隔，DTE才开始侦听传输介质。如果它听到传输，它就知道另一个DTE已经发送了一个较长的申请令牌帧，因此它简单地放弃成为第一个令牌持有者的尝试。如果没有听到传输，它会使用地址字段中下两位重复上述过程直到所有的地址位被用光。如果传输介质仍然安静，则该DTE成功地成为令牌的第一个拥有者。这个令牌的惟一拥有者通过使用响应窗口规程允许其他等待DTE加入逻辑环来继续初始化过程。

虽然DTE可以简单地通过当令牌传递给它时不响应，来把自己从逻辑环中删除。有一个清除方法，DTE可以接收到令牌然后发送给信息字段中含有自身后继地址的设置后继站帧。接着该DTE照常把令牌发送给自己的后继站，我们知道它不再是（逻辑环）的一部分了。

5. *优先级操作*

如令牌环网络，优先级机制也可以应用到令牌总线网络。但是令牌总线的访问方式只有四个优先级级别，称为**访问类别**，分别是0、2、4和6，其中6是最高优先级。正像我们前面所讲的，令牌总线网络主要用在诸如制造自动化和进程控制等应用领域。如下是四种访问类别
312 的典型应用：

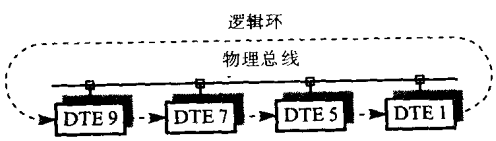
- 类别6：诸如关于严重告警情况和相关控制功能的紧急报文
- 类别4：关于一般控制措施和环管理功能的报文
- 类别2：关于用于数据日志的日常数据采集的报文
- 类别0：关于程序下载和主要文件传输的报文，或者说低优先级的长报文

每个DTE有两个控制帧传输的定时器：**令牌持有计时器**（THT）和**高优先级令牌持有计时器**（HP-THT）。后者控制高优先级帧的传输以确保所有DTE共享环容量（带宽）。当一个

DTE收到令牌，它先发送已等待了最长可达HP-THT确定的最大时间周期的任何高优先级帧。假定该DTE使用优先级机制并且THT没有超时，它开始使用下面的控制算法发送任何等待的低优先级帧。

逻辑环中的每一个DTE有一个定时器用来指示自从它上次接收令牌后开始计时的时间。这个时间是称为**令牌循环计时**（TRT）的变量。当该DTE再次接到令牌，它先把TRT中的当前值送入THT并把TRT重置为0。然后它传送任何等待的高优先级帧，同时增加TRT的值，并计算称为**目标令牌循环计时**（TTRT）的固定时间和当前THT的差。如果差是正的，到TTRT时该DTE还能发送任何等待的低优先级帧；如果差等于0或为负，该DTE就不能在这次令牌持有时间内再发送任何低优先级帧。每个使用优先级机制的DTE能够从较高访问类别向较低访问类别传送任何等待帧直到TTRT超时。

为了说明这个机制的工作，请考虑图6-22所示的实例。为了讲清楚，实例只假定两个访问类别。同样假定所有被发送的帧都是定长的，从而各种时间周期直接同帧的个数成正比。并假定DTE 9和DTE 1每次获得令牌只发送高优先级帧，而DTE 7和DTE 5只要有可能就发送低优先级帧。注意逻辑环按物理DTE的地址以递减数字顺序建立。且用于低优先级帧的TTRT固定为等价于8个帧的值。在每个DTE的左列标有TRT的值是该DTE为前一令牌循环所测量的令牌循环时间。右列标有XMIT的值是该DTE每次获得令牌所发送的帧数。每一行表示令牌的一次循环。



令牌循环	TRT	XMIT	TRT	XMIT	TRT	XMIT	TRT	XMIT
0	0	0	0	0	0	0	0	0
1	0	3	3	5	8	0	8	3
2	11	3	11	0	6	2	8	3
3	8	3	8	0	8	0	6	3
4	6	3	6	2	8	0	8	3
5	8	3	8	0	6	2	8	3
6	8	3	8	0	8	0	6	3
7	6	3	6	2	8	0	8	3
8	8	3	8	0	6	2	8	0
9	5	3	5	3	8	0	6	0
10	6	1	4	4	5	3	1	1
11	9	0	8	0	4	4	5	2
12	6	3	9	0	9	0	5	1

图6-22 优先级环实例

313 假定所有的传输在一段无效期后开始，在这段时间内令牌以尽可能快的速度循环。因此DTE 9中的TRT显示以0开始。假定令牌与正常帧相比，其传递和传播的时延可以忽略不计。同样也假设高优先级令牌持有时间为DTE获得令牌能发送3个高优先级帧所用的时间周期。

在令牌的第一次循环中，DTE 9获得令牌，在传递令牌前最多发送3个高优先级帧。当DTE 7从DTE 9获得令牌，它的TRT已经增为3，因为从上次获得令牌已经发送3个帧。这意味着DTE 7在传递令牌前可发送5（TTRT–TRT）个低优先级帧。DTE 5在获得令牌时，它的TRT值是8，因为自从上次获得令牌总共已经发送8个帧。因此在传递该令牌前它就不能发送任何低优先级帧。然后DTE 1不受计算的TRT的限制发送了3个高优先级帧。

在令牌的第二次循环中，DTE 9和DTE 1不受计算的TRT的影响发送了3个高优先级帧。
314 但是这次DTE 7不能发送任何低优先级帧（因为获得令牌时，TRT值超过了8），而DTE 5能发送2个低优先级帧（TTRT–TRT=2）。

在令牌的第三次循环中，DTE 9和DTE 1各自又发送了3个高优先级帧，DTE 7和DTE 5都不能发送低优先级帧，因为计算的TRT已经达到了TTRT的限制值8。

在令牌的第四次循环中，出现了同第二次循环相似的情况，但是注意这次计算的TRT使得DTE 7而不是DTE 5有机会发送2个低优先级帧，DTE 5这次不能发送任何帧。类似的，在令牌的第五次循环中，DTE 5能发送2个低优先级帧而DTE 7被禁止发送任何低优先级帧。然后重复这个周期。我们容易推断出：在任三次连续循环过程中，DTE 9和DTE 1使用了82%（18/22）的可用容量传送高优先级帧，而DTE 7和DTE 5共享剩余的容量（4/22）传送低优先级帧。

在令牌的第八次循环中，假定DTE 1暂时停止发送高优先级帧，因此DTE 7和DTE 5能多发几个等待低优先级帧。类似的，在第十次循环中，DTE 9也停止发送高优先级帧，依次类推。

虽然这是个简单的例子，但是它说明了优先级机制允许相对不受限制地发送高优先级帧，而在空闲容量可用的情况下以公平的方式发送低优先级帧。

6.3 性能

为了比较讨论过的三种介质访问方式的相对性能，作者实际了一组模拟，图6-23显示了它的结果。

在这个模拟中，所有的局域网段一样长，都为2.5公里并以相同的10Mbps的比特率传输。每种情况都有100个DTE/站。

这个曲线图给出了一个帧在局域网中传输所花的平均时间作为一种提供负载功能。负载表示为可用比特率的一部分，称为**标准化吞吐量**。在图6-23(a)中，所有被传输的帧都是512位长，而(b)是12 000位长。在更小帧长度的情况下，获得令牌时只有单一帧被传输。实际上，许多这种帧被传输，但是因为这是帧优先级的功能，所以只考虑单一帧。

另外，当然有长短帧混合传输的情况，因此它的平均传送时间位于两组曲线图所示的中间。

315 在每个站随机地生成帧，**传送时间**定义为从帧的生成（即它到达MAC子层输入队列时）时刻到帧被目标设备成功接收的时刻之间的时间间隔。这样它包括帧在MAC子层输入队列等待的时间，特殊MAC方式相关的时延和传输帧的时间。

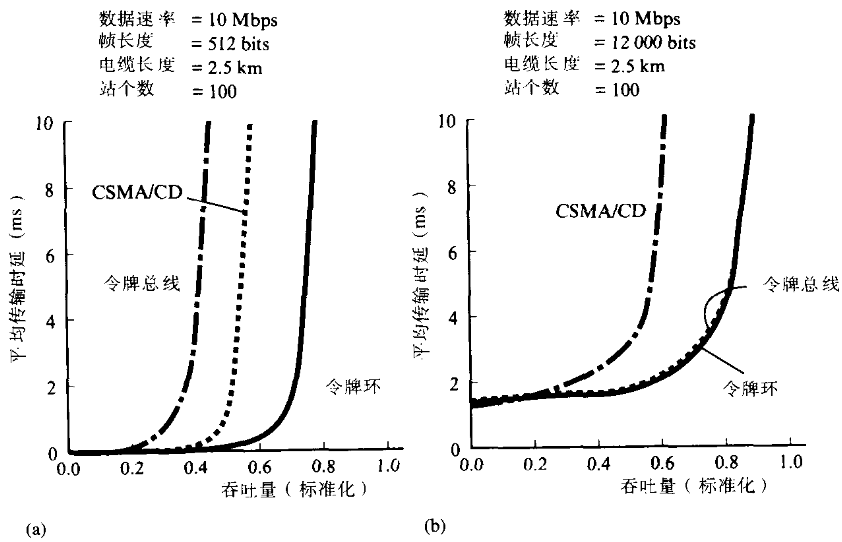


图6-23 LAN性能比较

(a) 512位帧 (b) 12 000位帧

正如在图中看到的，每种局域网类型中较大帧长度的平均吞吐量较高。这是因为与每个帧相关的开销相对于帧内容来说较大帧长度更小。但是同样要注意，令牌环局域网比起令牌总线局域网，吞吐量对较小帧长度更不敏感。这是因为在令牌环中令牌的大小只有24位而令牌总线中令牌大小为152位。同时，关于令牌总线的处理开销要高于令牌环。

注意每组图中的吞吐量被标准化了，因此为了得到特定的吞吐量，以较小的帧长度生成大量帧。由此在CSMA/CD访问方式中，在特定吞吐量级别，较小的帧长度冲突的概率会更高。另外，在较小帧长度情况下，按照为解决冲突损失的时间和相关的恢复时间，与介质访问方式相关的开销显著地提高。

总之，吞吐量的差异仅仅对于提供过度负载是显著的，比方说超过总吞吐容量一半。对于提供的小于它的负载，三种局域网的平均传输时间是差不多的。实际上，在第7章将要讲述的更先进的应用类型中，大多数局域网相对于它们最大的容量，仅提供适度的负载。但是在负载很重要的情况下，令牌访问方式更好。 316

6.4 无线局域网

到目前为止，本章中讨论的局域网类型都是使用双绞线或者同轴电缆作为物理传输介质的。这种局域网的主要花费是安装物理有线电缆。而且，如果互连计算机的布局改变，那么相应的布线也要改变，会导致再次安装的花费。这就是无线局域网出现的一个原因，无线局域网不使用物理电线作为传输介质。

第二个原因是手持终端和便携式计算机的出现。技术的不断进步意味着这些设备同许多静态计算机在能力上可以比较。虽然使用这些设备的主要原因是其便携性，但是它们一般需要与其他计算机通信。这些计算机可能是其他便携式计算机，或者是连在有线局域网上的计算机（服务器）。比如在零售商店中的手持终端与库房的计算机通信以更新库存记录，或者医院里护士的便携式计算机访问大型机上数据库中的病人记录。

图6-24(a)给出了无线局域网的两种应用的示意图。正如所见，在第一种应用中，为了访

问有线局域网中的服务器，使用了一种称为**便携式访问单元**（PAU）的中间设备。一般来讲，
317 PAU的覆盖范围是50米～100米，因此在大型应用中有许多这样的单元分布在站点周围。总之，这些提供了对站点局域网的访问，即通过手持终端、便携式计算机或静态计算机（它们可以位于站点周围的任何地方）访问服务器。这种应用称为**基础设施无线LAN**（infrastructure wireless LAN）。

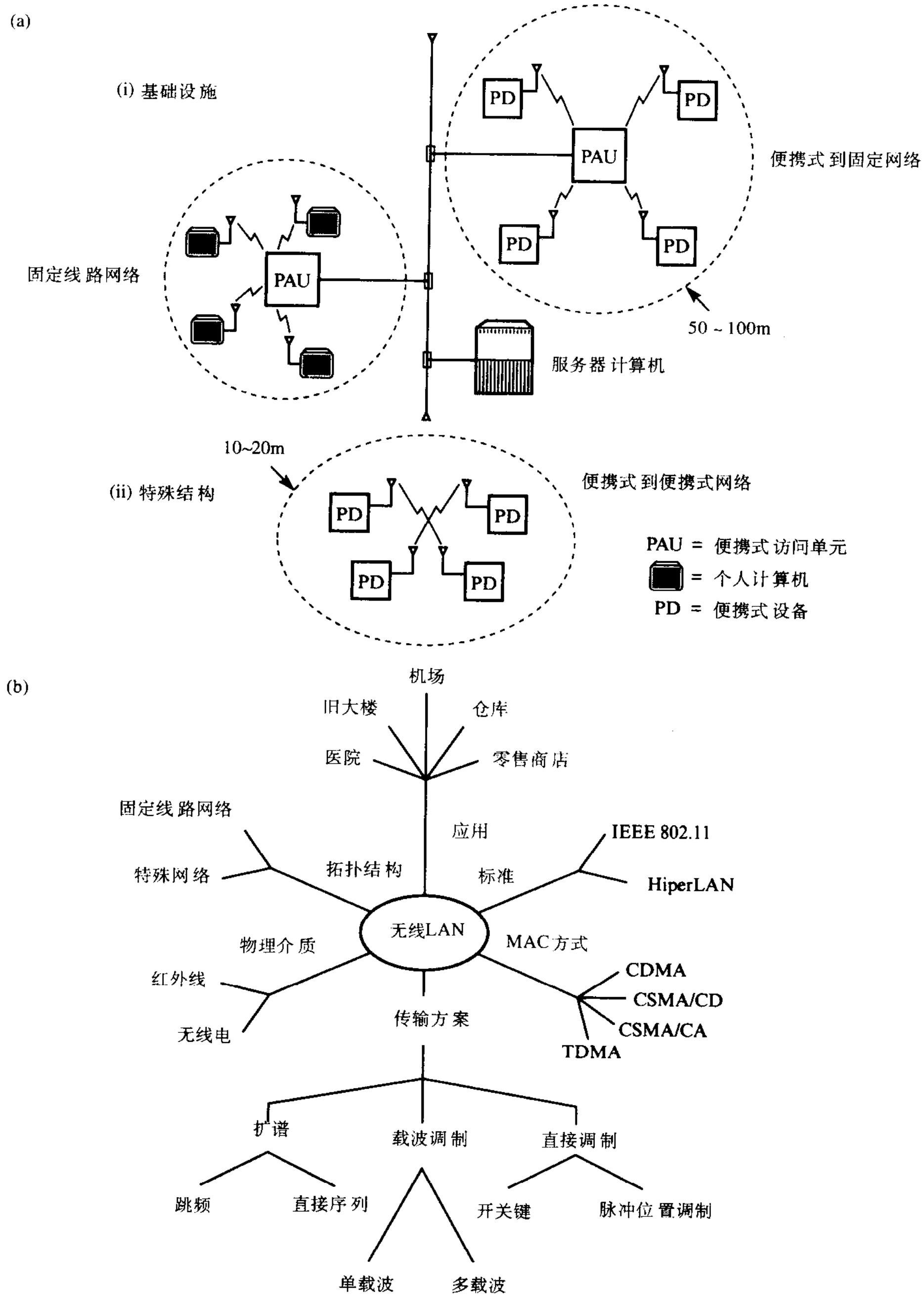


图6-24 无线LAN
(a) 应用拓扑结构 (b) 技术问题

在第二种应用中，一组便携式计算机想彼此通信来形成独立的局域网。例如，这可能是在开会的会议室中或者机场中。由于需要建立这种网络，称它们为**特殊结构无线LAN**。如同有线局域网，描述无线局域网的操作时要考虑许多要点，图6-24(b)概要地给出了这些要点。实际上，许多问题是互相交叉的，但为了描述的目的对每个都分别介绍。

6.4.1 无线传输介质

用于无线局域网的两种传输介质是无线电波和红外线光信号。虽然我们分别讨论每种的特征，但是仍可以看到两者使用了类似的技术。

1. 无线电

无线电波广泛地用到许多应用中。这包括无线电、电视广播和蜂窝电话网络。因为无线 318
电波能轻易地穿过如墙和门等物体传播，因此对无线电频谱的使用有严格的限制。应用的大范围意味着无线电带宽不足。对于一个特定的应用，必须由官方分配一个特殊频段。历史上这已在国家基础上完成，但是越来越多国际协议被签署，用来为那些有国际联系的应用留出可选的频段。

无线电发射被限制在特定频段内以及相关接收者只能在此频段内选择信号意味着基于无线电系统的相关电路通常会比应用在红外线光系统中的相关电路更加复杂。但是，无线电的广泛应用——尤其在高音量消费者产品中——意味着复杂的无线电系统设计能以合理的成本实现。

（1）路径损失

所有的无线电接收器工作在指定的信噪比（SNR）下，更确切地说，接收到的信号的能量同接收器噪声信号的能量的比率必须不低于指定值。通常，当SNR降低时，接收器的复杂性（由此需要的成本）会增加。但是，便携式计算机成本下降意味着无线电网络接口单元的可接受成本必须同便携式计算机的成本相当。因此，这意味着无线电接收器的SNR必须设置成尽可能高的值。

实际上，SNR由许多相互关联的因素决定，并且它们每个都和无线电接收器的设计有关。正如我们在2.2节中讲到的，接收器噪声指数是周围温度（它引起了热噪音）和接收信号的带宽两者的函数。带宽越大或温度越高，引起噪声指数会越大。因此对于特定的应用，接收器噪声指数基本上是固定的。

接收器端的信号能量不仅受传送信号的能量影响，而且还受发送器和接收器之间的距离影响。在真空，无线电信号的能量随着与信号源间距离的平方衰减。另外，在室内衰减会加剧，首先因为诸如家具和人等物体的存在，其次因为从这些物体反射的信号引起的传输信号破坏性干扰。这些共同产生了所谓的无线电信道**路径损失**。

因此为了使无线电接收器能以可接受的SNR工作，无线电必须以尽可能高的发送能量和（或者）有限的覆盖范围操作。实际上，对于便携式计算机，发送信号的能量受到无线电网络接口单元能量开销（它导致了计算机电池组的负载增长）的限制。由于这些原因特殊结构无线LAN的覆盖范围要比基本结构无线LAN小。

（2）邻道干扰

因为无线电通过多数物体传播只有适度的衰减，它又可能被位于同一幢大楼的邻近房间 319
或其他大楼内以同频带工作的发送器所干扰。因此，对于特殊结构无线LAN来说，多个这样的局域网建立在邻近房间/区域内，必须采用相应技术来允许同一频带的许多用户能共存。

在基本结构无线LAN中，因为拓扑结构是已知的并且无线网络的覆盖总范围更大（对于

现存的有线局域网来说），因此可用的带宽可以分成许多子带使得邻近子带的覆盖区域能使用不同的频率。图6-25(a)说明了这种方案，称这种方案为**三信元重复模式**，虽然更大的模式也可以。每个信元中可用带宽选择值是为该区域预测用户提供可接受的服务水平。这导致了可用带宽的更优使用并且通过保证邻近单元都使用不同的频率，**邻道干扰**被大大减少。

（3）多重路径（Multipath）

无线电信号像光信号一样受**多重路径**影响，就是说在任何时刻接收器会收到来自同一发送器的多个信号，每一个沿着从发送器到接收器之间不同的路径传输。我们称为**多路径色散**（multipath dispersion）或**时延扩展**（delay spread），它使得与前面位/符号相关的信号干扰与后面位/符号相关的信号。这被称为**码间干扰**（ISI），如图6-25(b)所示。显然，比特率越高，由此得到的每个位单元周期越短，码间干扰的程度越大。

另外，称为**频率选择衰减**（frequency-selective fading）的损失由不同接收信号的路径长度变化引起。它引起它们之间（无线电频率）的相对相移，这使得各种反射信号严重地减弱了直线路径的信号，甚至在某种程度两者互相抵消。这被称为**雷利衰减**（Rayleigh fading），如图6-25(c)所示。实际上，反射波形的振幅是直接波形振幅的几分之一，衰减的程度由反射材料决定。一种解决方法是利用无线电频率信号的波长很短（几分之一米），由此对天线位置的微小变化较敏感的事实。为了克服这种影响，通常使用相距四分之一波长放置的两根天线，从两根天线接收到的信号合起来形成复合的接收信号。这种技术称为**空间合成**（space diversity）。

另一种解决方法是利用称为**均衡**的技术，即直接信号的时延与衰减图像——相对于多重路径反射信号——从实际接收到的信号中去掉。因为反射信号会随着发送器和接收器的位置变化而变化，这个过程需要适应性。因此，使用的电路称为**自适应均衡器**。显然，这种电路的使用增加了接收器实施的成本。

320

2. 红外线

红外线发射器和检测器已经在许多应用中使用很多年了。这包括光纤传输系统以及诸如用于电视机、CD播放器和VCR的各种远端控制应用。红外线以比无线电频波高得多（大于10^{14}Hz）的频率发射，通常设备发射和探测以红外线信号的波长分类而不是频率。波长以纳米（10^{-9}米）为单位，定义为一个信号周期内光传播的距离。公式是：

$$波长\ \lambda = c/f$$

其中c是光速（3×10^{8}米/秒），而f是信号频率（单位Hz）。

应用广泛的两种红外线设备分别是800纳米波长和1300纳米波长。

红外线相对于无线电的一个优点是不需要任何调节。同样，红外线有与可见光类似的波长：如在光滑的表面有反射，会穿过玻璃但穿不过墙壁或其他不透明物体。因此红外线的发射在无线LAN应用中被限制在一个房间内，所以减轻了邻道干扰的程度。

当使用红外线作传输介质时，另一点必须注意的是背景（周围环境）光。太阳光以及电灯丝和荧光灯产生的光都含有大量显著的红外线。这些光和发射器发射的红外线一起被探测器接收到。这意味着噪声能量会很大，需要信号能量很大以得到可接受的信噪比。实际上，红外线的路径损失会很大。同样，红外线发射器的电光能量转换效率相对较低。总之，这些会导致对于电源的能量要求很高。为了减轻噪声电平，通常传送复合的接收信号通过**光带滤波器**（optical bandpass filter），它衰减发射信号频带以外的那些红外线信号。

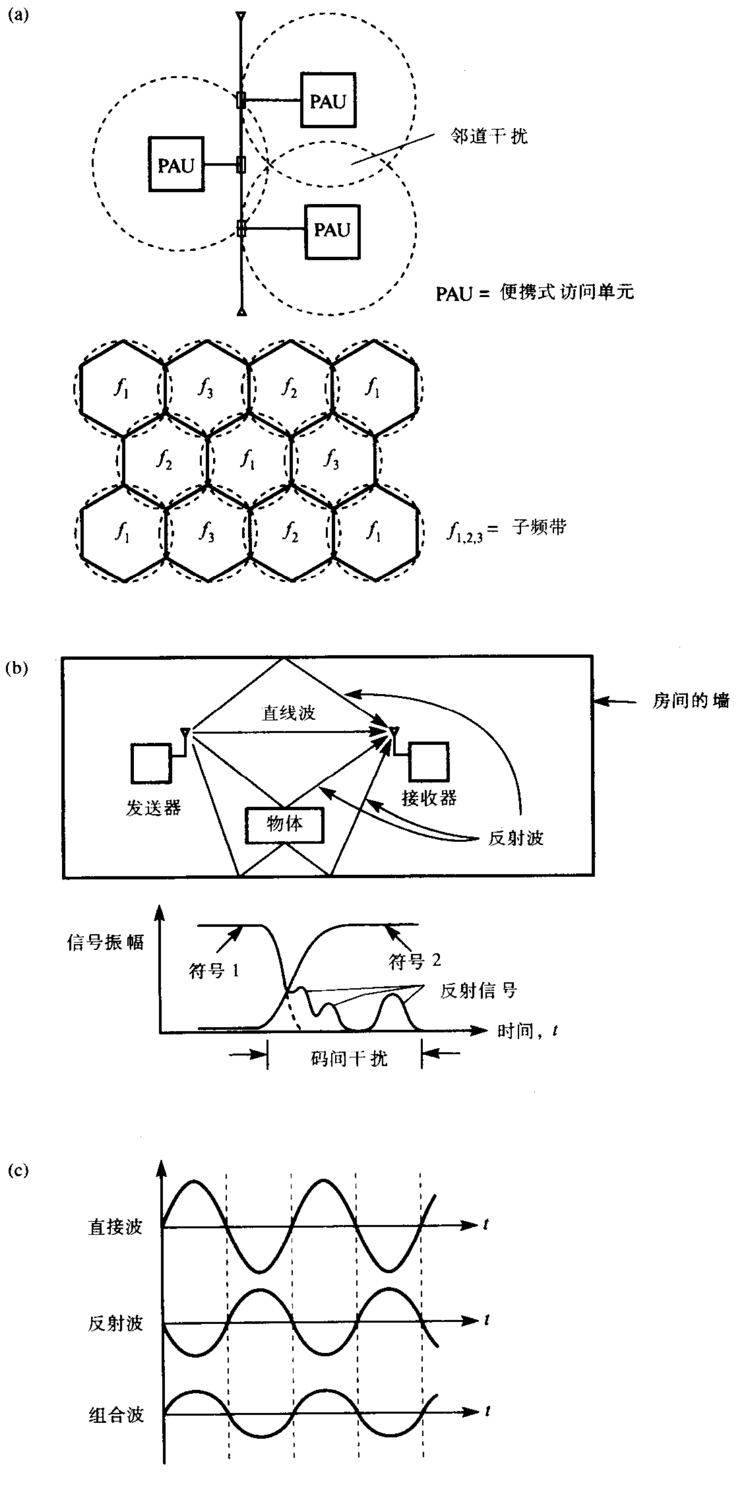


图6-25 无线电损耗

(a) 邻信道干扰以及频率分配策略实例 (b) 码间干扰 (c) 雷利衰减

（1）设备

对于无线LAN应用，工作模式是使用调制信号调制发射器红外线输出的强度。红外线信号的强度变化被探测器接收到，并直接转换成等价电信号。这种模式称为**直接检测强度调制**（intensity modulation with direct detection（IMDD））在6.4.2节中还将看到包括基带调制的各种调制方式。

321~322 有两种红外线发射器：**激光二极管**和**发光二极管**。激光二极管广泛地使用在光纤传输系统。它们产生连续光源，就是说很窄的频段（一般1～5nm）。当光被限制在一个小范围时就获得了高能量密度。在无线LAN应用中，因为光在光纤范围中传播没有受到抑制，激光源必须扩散，否则会导致严重的眼睛伤害。相比之下，发光二极管（LED）产生由一个频带（一般为25nm～100nm）组成的光源，并且使用了低能量输出，十分安全。LED可用的调制带宽限制在20MHz，使得使用的最大比特率限制在10Mbps以下。由于它们的低成本，我们通常使用LED将位速率升至这个级别。

对于比特率超过10Mbps的，必须使用激光二极管。激光二极管可用的调制带宽是几百兆赫兹。频率的宽带——谱宽——与LED相关意味着在接收端必须使用宽带通的光过滤器检测所有发送信号。但是，这样就增加了接收器噪声，使接收器的设计在高比特率下更困难。

（2）拓扑结构

红外线连接使用两种方式：点对点和扩散。在点对点模式中，发射器直接对准检测器（实际上是光电二极管），因此使用低能量的发射器和低灵敏度的检测器。这种方式适合提供两台装置之间的无线连接，比如使一台便携式计算机能从另一台计算机上下载文件。

对于无线LAN应用，需要点对多（广播）方式。因此，红外线源输出是扩散的，使得光在广角范围内传播。这就是**扩散方式**：如图6-26所示，有三种工作模式。在基本模式(a)部分，有一个与计算机相关的广角光发射器和检测器。任何发射器输出的红外线信号会在房间内经过多重反射后被检测器接收。这种工作模式的结果是同一个源信号的多个拷贝在由每个信号经过的物理路径所决定的不同时间间隔到达每个检测器。正如前所述，这是多重路径散射，结果又称时延扩展，因为在传输二进制数据流中表示每一位的脉冲被扩展或加宽了。像无线电波，各种反射信号的振幅与最直接信号相比随着传播路径引起的衰减而变化。在一般房间/办公室中，有意义的信号以高达100ns的时延扩展被接收。这种模式只在1Mbps的比特率下才令人满意，因为在更高的比特率下码间干扰会显著增加。

对于红外线（和无线电）来说，除了均衡之外，我们能通过使用多方向发射器和检测器（在无线电中用**定向天线**）来减轻时延扩展所产生的结果，如图6-26(b)所示。在这种方式中，
323 所有的发射器和检测器对准称为卫星的固定在天花板上的反射圆顶的一个点。为了使接收到的信号能量最大化和反射最小化，源信号集中形成相对较窄的光束。选择卫星反射圆顶的形状以确保所有的传输信号能被所有的检测器接收到。为了减轻多路径的影响，检测器的孔做得很小，这样它们只能接收来自卫星的直线信号。

刚刚描述的方案中卫星只是一个光反射装置。因此在探测器端为了得到可接受的信号能量，发射的信号能量相对要比较高。便携式设备从电池中获得能量，这是一个不利因素，因此基本方案的进一步改善是使用图6-26(c)所示的**有源卫星**（active satellite）。在这个方案中，一组检测器（光电二极管）和一组红外线发射器分布在圆顶周围。所有被一组或更多组检测器接收到的信号然后被发射器转发。这意味着每个便携式设备发射的信号能量可以很低，因
324 为它只需要形成到卫星直线路径的足够能量就行了。

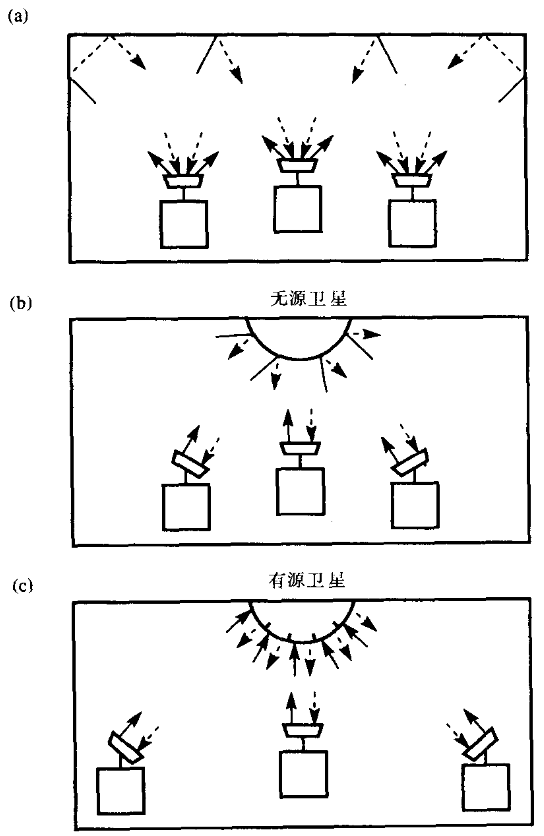


图6-26 红外线工作模式

(a) 点对点 (b) 无源卫星 (c) 有源卫星

6.4.2 传输方案

红外线和无线电的不同传播特性产生了不同的传输方案。我们会分别讨论用于每一种传输介质的方案。

1. 无线电

有四种传输方案用于无线电无线LAN：**直接序列扩频**（direct sequence spread spectrum），**跳频扩频**（frequency-hopping spread spectrum），**单载波调制**（single-carrier modulation）和**多子载波调制**（multi-subcarrier modulation）。

（1）直接序列扩频

同大多数其他无线电频谱应用相比，无线LAN相对较新。虽然存在可用的无线电频谱，但相对频率较高，一般是数千兆赫兹。对这样的频率要求有新的组件，成本相对也较高。这是无线LAN应用明显的缺点，固定有线LAN类型的网络接口卡的成本现在很低。而且，随着便携式计算机成本下降，无线LAN接口的可接受成本也已下降。基于这些原因，第一代基于无线电的无线LAN标准使用现存的频带，适用于这些频带的组件可以轻易地得到。有一个称为**ISM频带**的频带留给一般工业、科学和医学（ISM）应用。这个频带现存的应用包括高能量无线电频率加热设备和微波炉。业余无线电使用者也允许使用这个频带，通常在较高的传输能量级别。为了同这些应用并存，选择的传输方案具有高水平的联合频道干扰拒绝是基本的。对于无线LAN应用，这通过使用称为**扩频**的技术获得。有两种形式的扩频：直接序列和跳频。

我们会在本小节后面讨论前者，在下一小节讨论后者。

图6-27显示了直接序列扩频工作原理的示意图。将传输的源数据首先与**伪随机的二进制序列**异或（就是说组成序列的位是随机的，但相同序列速率比源数据速率更大）。然后再调制异或信号并传送，它占据的带宽（就是说扩展成）比起原始源数据带宽成比例增加，这使得信号对同频带的其他使用者成了（伪）噪声。

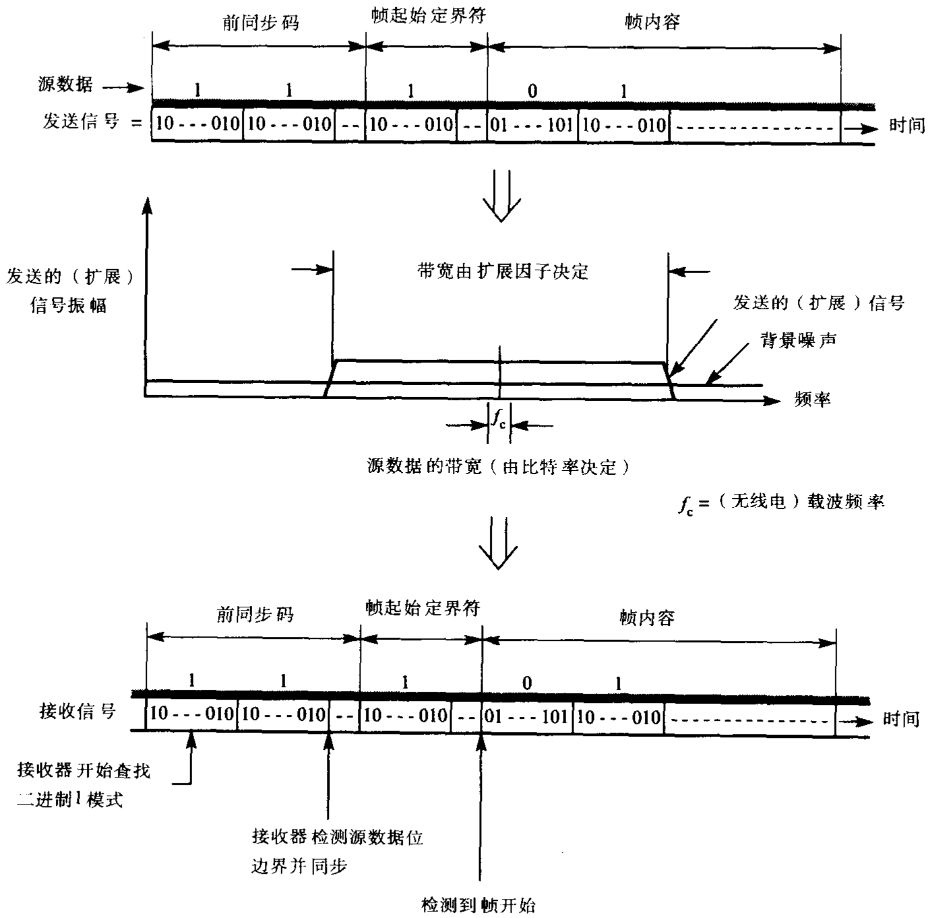


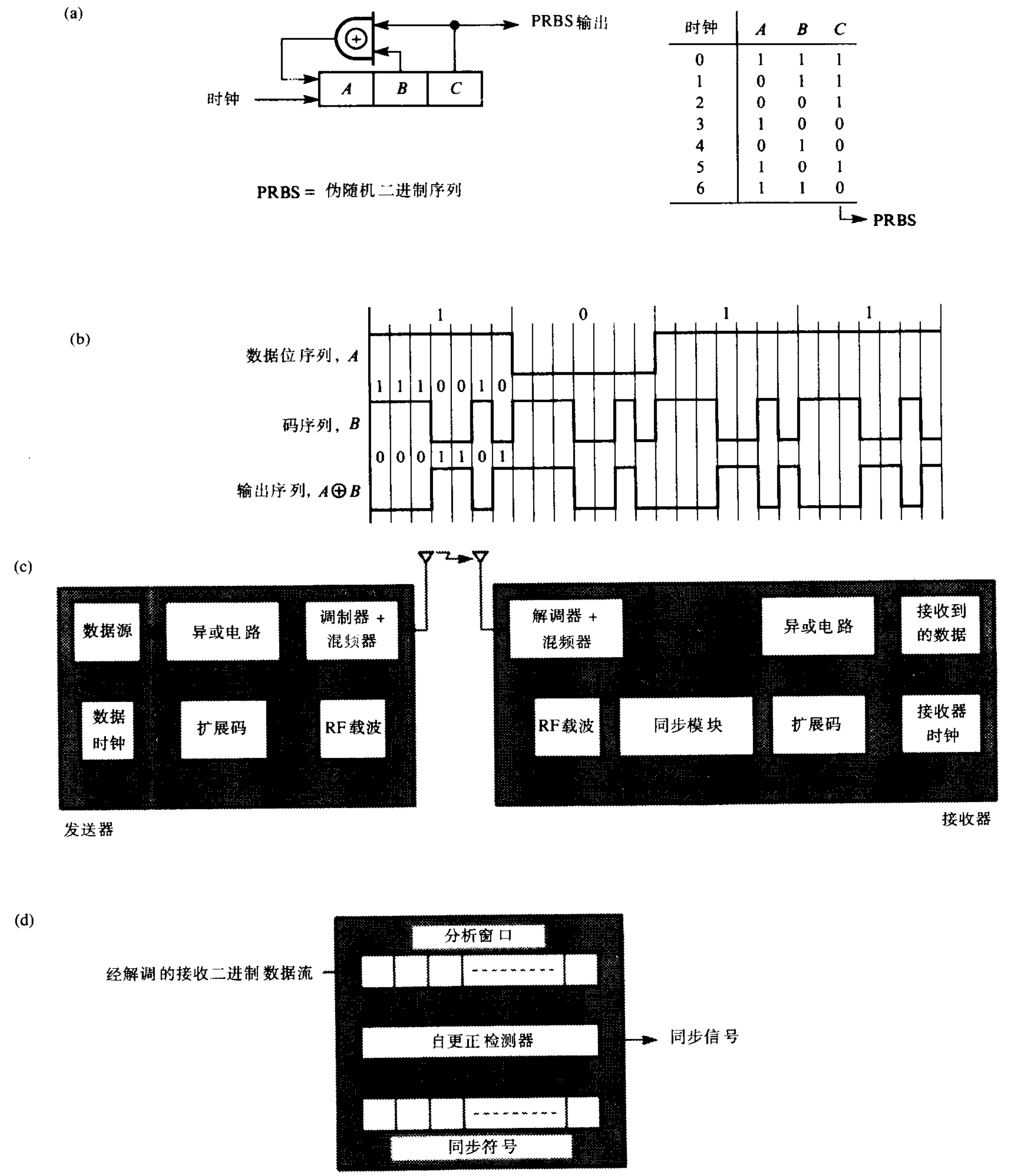
图6-27 直接序列扩频工作原理

同一个无线LAN的所有成员都知道使用的伪随机二进制序列。所有传输的数据帧跟在前同步码序列之后，前同步码序列后面是帧起始定界符。因此在解调传输信号后，所有的接收
325 方先查找已知的前同步码序列——通常是全1字符串，一旦找到，接收者就开始解释收到的在正确源数据位范围内的二进制数据流。然后它们等待接收帧起始定界符，接着就继续接收帧内容。正常情况下接收方由帧头部的目标地址决定。

显然，因为属于同一个无线LAN的所有站占用同一个分配频带，使用同一个伪随机二进制序列，它们的传输会互相干扰。因此必须使用正确的MAC方式来确保某个时刻只有一个传输发生。

实际上，伪随机二进制序列的产生相对简单，因为它只需使用连接在反馈环的一些移位寄存器和一些异或门就能产生。原理如图6-28(a)所示。在这个例子中，使用了一个3位移位寄

存器和一个异或门，转发前产生7个伪随机3位码（也称为（移位寄存器）状态）。注意000状态不存在，因为那样的话移位寄存器内容会在每个后继时钟脉冲后保持不变。一般，*n*位移位 326
寄存器最多有2^n-1个状态，如果反馈组合产生所有2^n-1个状态，我们称之为**最大长度移位寄存器**。从移位寄存器最重要元件得到的输出用作伪随机二进制序列，在这个例子中它是7位二进制格式1110010。



时钟	A	B	C
0	1	1	1
1	0	1	1
2	0	0	1
3	1	0	0
4	0	1	0
5	1	0	1
6	1	1	0

图6-28 直接序列频谱原理

(a) 伪随机序列生成器 (b) 扩展序列生成器 (c) 发送器和接收器示意图 (d) 同步模块示意图

伪随机序列是用来执行序列和每个传输数据位之间的异或操作。在例子中，如果我们假定图6-28(a)中得到的7位伪随机格式，然后(b)部分说明了对应一组4数据位的传输位格式。注意每一个数据位有7位被传输，同样注意，对于任何数据位来说，二进制0的传输位序列只是二进制1简单取反。伪随机二进制序列又称为**扩展序列**，序列中的每个位都称为**片**，因而产生的传输比特率称为**片率**，序列中的位数称为**扩展因子**。

扩展因子决定了扩频系统的性能。通常，以分贝（dB）表示，我们称之为**处理增益**，就是说处理增益是扩展因子的对数。比如一个扩展因子为10：1的扩频系统的处理增益是10dB，100：1是20dB，依次类推。按照信噪比（SNR，同样表示成分贝），处理增益有效地从中减去。因此一个需要10dB SNR的非扩频系统，信号能量必须是噪声能量的10倍才能令人满意地工作，而在一个扩频系统中，处理增益是10 dB，即使信号能量等于噪声能量也能令人满意地工作。

图6-28(c)显示了一个简单的直接序列无线电发送器和接收器的示意图。在每个数据位被伪随机序列异或后，由此产生的高比特率二进制信号被载波信号调制发送。使用混频器电路后结果信号的频率增加了，这使得传送信号在规定的频段范围内。通常的调制方案使用二进制相移键控（BPSK）和正交相移键控（QPSK），它们的原理在2.5.1节描述调制解调器的时候已经讲过了。

因此我们可以推断接收器必须对接收到的信号进行同步处理使得执行异或操作的结果能在正确数据位（符号）范围内被解释。为实现此目的，已知的二进制模式被放在帧开始部分（前同步码）传送，接收器用前同步码来获得时钟（位）和符号同步。同步模块的示意图如图6-28(d)所示。

使用3.1.1节描述的一个标准方法可以达到时钟同步（片率）。为了达到符号同步（数据速
327~328 率），每个传输帧跟在由二进制1符号（数据位）串组成的前同步码后面。当这个扩展前同步码被接收时，它通过*n*位移位寄存器（这里的*n*是扩展序列的位数）并跟已知的对应一个1数据位的序列按片比较。如果在特定片位置的两位相同，就出现一个一致（A），如果不同就出现一个不一致（D）。测量两个符号之间差异的方法是把计算的D个数从A个数中减去，这被称为**自相关函数**。显然，当已知符号被定位后，自相关函数会是一个最大的等于扩展序列中片数的正值。然后接收器就达到符号同步了。同步处理的可靠性由被选的扩展序列和它的移位转换之间的自相关所决定。当片错误在接收到（解调过的）扩展序列中存在时这变得相当重要，比如过度的噪声。这由实例6-2中有很好的例示。

实例6-2

用在直接序列扩展频谱系统中的一个典型的扩展序列是11位二进制序列10110111000。它是**巴克序列**的一个例子。确定并划分这个序列两边正负10位（片）的自相关。

解：

329 图6-29概要说明了这个解决方法。在(a)部分接收到的位（片）序列显示在分析窗口中，并与扩展序列（同步码）一一对应，两边是额外的序列。显然，在这个位置，所有位位置都一致因此自相关等于+11。另两个例子对应于扩展序列加减1位的情况，我们看到，得到的自相关都是-1。实际上，这个伪随机序列两边所有的位位置都得到-1。自相关划分如图6-29(b)所示，我们看到，通过同步模块的信号输出只有当同步码收到的时候才为正。

（2）跳频扩频

跳频扩频的工作原理如图6-30(a)所示。分配的频带被划分成许多较低频率的子频带，称为**信道**。每个信道的带宽相同并由数据比特率和所用的调制方式决定。接着发送器在移到

（跳跃到）不同的信道前在一个短时间周期内使用每个信道。当一个信道被使用时，信道中间的载波频率被那个时刻发送的位（流）调制。信道的使用形式是伪随机，被称为**跳跃序列**，用在每个信道的时间称为**片周期**，跳跃率称为**片率**。

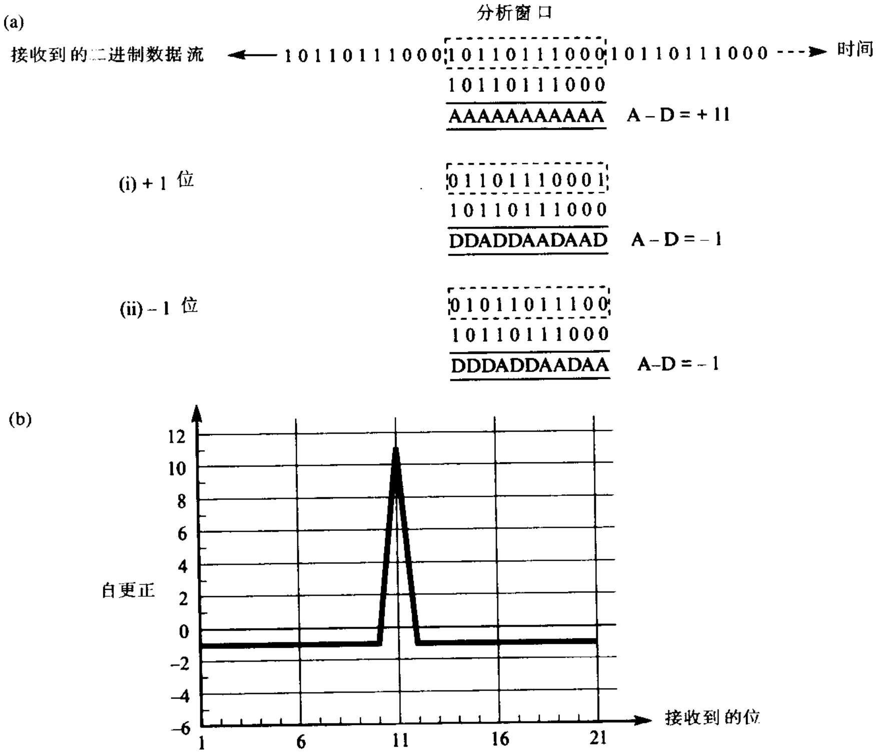


图6-29 同步例子

(a) 自更正例子 (b) 自更正图表

跳频的工作模式有两种，由片率与源数据率的比率决定。如图6-30(b)和(c)所示。当片率大于源数据率时，工作模式称为**快速跳频**，而当片率小于源数据率时称为**慢速跳频**。两种情况下，载波频率都用在每个信道中央。

跳频优于直接序列是因为避免使用在全部分配的频率中选定（窄带）的频道。这在ISM频带中尤其有用，因为有可能在局域网覆盖域出现一个或多个高能量窄带干扰源。如前所述，虽然直接序列干扰信号会散布到分配频带中，使用高能量源它仍然会导致严重的干扰，在极限情况下提供的特定频带不可用。但是，使用跳频，如果知道工作在特定频率的干扰源存在，那么可以从跳跃序列中停止对那个频率的使用。

这个技术在慢速跳频中尤其有用，因为使用快速跳频的话，每个数据位有多个跳频，由此只有单一片会受到影响。那么使用多数判定确定最可能传输的数据位（0或1）。但是，快速跳频系统比慢速跳频系统成本更高。还有，因为发送器和接收器必须同步（就是说一起跳跃），所以慢速跳频系统更容易达到同步。因此慢速跳频系统为无线LAN提供更低成本的可替换方案。 330

（3）单载波调制

使用这种方法，单频载波信号（位于分配频带中央）使用恰当的调制电路与要传送的数据调制。原则上，它是2.5.1节描述的用于在模拟交换电话网络上传输数据的调制方案的简单

扩展，不同的是，无线LAN所需的比特率（由此带宽）会更高。

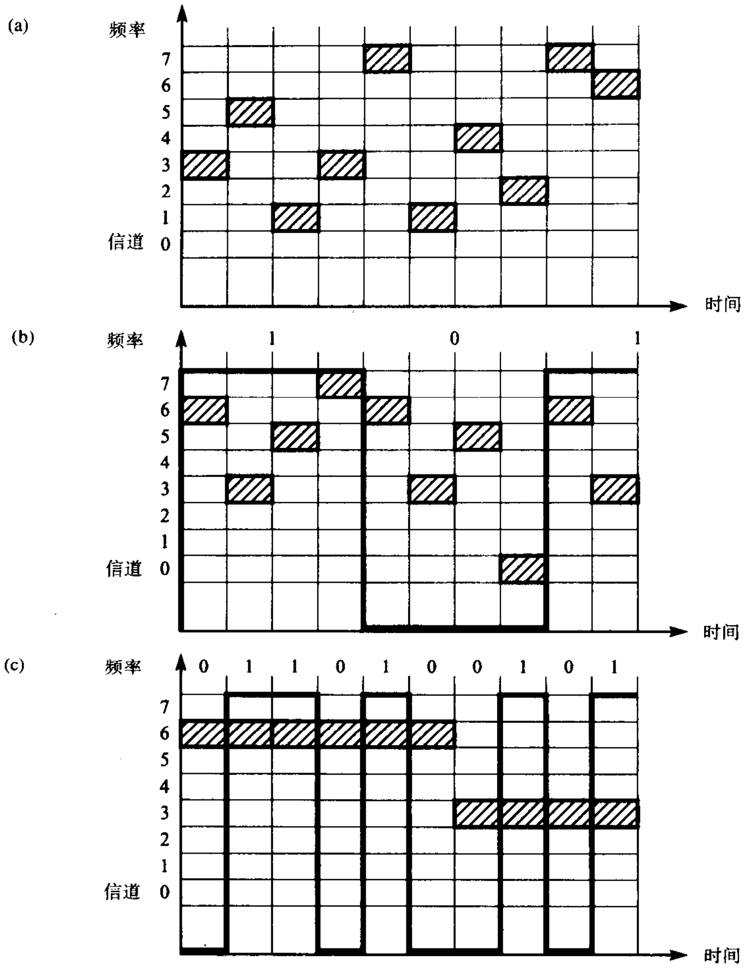


图6-30 跳频扩展频谱

(a) 工作原理 (b) 快速跳频 (c) 慢速跳频

回忆一下有许多种不同的调制方案，它们涉及振幅、频率和相位或者它们的组合。但是，无线LAN所需的高带宽减弱了涉及振幅变化调制方案的可行性，因为跟带宽呈线性关系的能
331 量放大器成本既高又消耗大量能量。通常使用基于单一恒定振幅载波相位变化的调制方案，如正交相移键控或者它的变型方案。此外，正如我们前面提到的，对于超过1～2Mbps的比特率来说，多路径的扩散引起了高电平的ISI，因此还必须使用更复杂的均衡电路。

（4）多子载波调制

这个方法的工作原理是首先把要传送的高比特率二进制信号分成许多较低比特率流。然后每个较低比特率流像单载波方案一样，用来调制单独子载波（来自分配频带）。但是在这种情况下，因为每个载波相对较低的比特率，ISI的电平大大降低，它消除了对均衡器的需要。虽然频率选择衰减仍然存在，但是可能只有一个（或少量）子载波会受到影响。转发差错更正技术（如附录A中描述的卷积码）用来提高信道的剩余BER。实际上，使用的子载波是第一

个子载波的整数倍（$f_1, 2f_1, 3f_1$等），因此这个方案还称为**正交频分多路复用**（OFDM）。

在传输前，使用**快速傅立叶转换**（FFT）数学技术，将调制的单独子载波组合成一个复合信号。产生一个定义在时间域上的输出信号，它有类似于单载波方案所需的带宽。但是在这种情况下，在接收器方信号使用逆FFT操作转换回它的多子载波形式。然后解调的低比特率流被重新组合成高比特率二进制输出流。

两种调制方案的选择取决于执行均衡操作所需处理能量的成本（功率需求）和执行FFT操作所需处理成本哪个更高。

2. 红外线

使用红外线信号传送数据有许多种方法，包括直接调制和载波调制。

（1）直接调制

不像无线电必须工作在指定频带，红外线本身受限于单个房间，所以有可能直接调制红外线源信号，二进制1开启发射器而二进制0关掉发射器。这种调制称为**开关键控**（OOK），广泛地应用在光纤传输系统。它是最简单的调制类型，它的实现需要相对简单的电子设备。这个方案的示意图如图6-31(a)所示。

像在固定线路链路上的基带传输一样，因为接收器要获得时钟/位同步，所以源二进制数据流必须在调制前在发送器方使用3.3.1节描述的一种标准时钟编码方式进行编码。一般，使用曼彻斯特编码或者有0位插入的NRZI和DPLL。 332

另外，一种称为**脉冲位置调制**（PPM）的技术用在光系统中，来减少LED红外线源的光能量需求。PPM的工作原理如图6-31(b)所示。使用这种方法，发送的二进制数据流先分成n位符号组。对于每个符号，在2^n个时隙位置之一发送一个单脉冲。例如，$n=2$，因此在4个可能时隙之一发送一个单脉冲。每个符号有4位（由此有16个脉冲位置）用于工作在1～2Mbps的高比特率系统。它是当前设备可用的最大符号长度。此外，在较高比特率下通常需要均衡电路来降低多路径扩散的影响。

正如6.4.1节所示，光过滤器用来减少由太阳光和人工光引起的干扰。任何剩余干扰的影响会提高光电二极管的关信号电平，并在极限情况下引起检测器电路错误理解接收到的信号。它是另一个限制比特率（使用OOK能获得大约2Mbps的比特率）的因素。 333

（2）载波调制（Carrier modulation）

为了获得更高的比特率，我们需要使用类似于无线电系统中使用的载波调制技术。结构如图6-31(c)所示。实际调制方法包括FSK和PSK，它们的原理在2.5.1节已描述过。因为使用这种方案发送经调制（以频率或相位）的二进制数据——载波信号，所以在接收器方可能在解调前让来自红外线检测器的信号输出通过附加的电子过滤器。正像2.5.1节中指出的，这些滤波器只通过载波周围的有限频带（含有源数据）。它的作用是进一步过滤任何剩余的干扰信号，因此给出比直接调制系统更高的性能。能轻易地获得2～4Mbps的比特率。

在较高比特率下，由多路径引起的ISI变成限制因素，因此必须采用克服它的技术。一种方法是使用多子载波调制。使用这种方案时可用带宽被分成许多子频带，每个子频带用来传送二进制数据流的一部分。例如，如果使用两个子频带，那么每个子频带能传输二进制数据流中的交替位。它意味着每个子频带只需要以一半比特率传输，由此每个位信元周期增加到原来的两倍。它使得信号较少受ISI影响并能获得10Mbps的比特率。显然信号周期的增加使得发送器和接收器电子设备的复杂程度也相应增加。

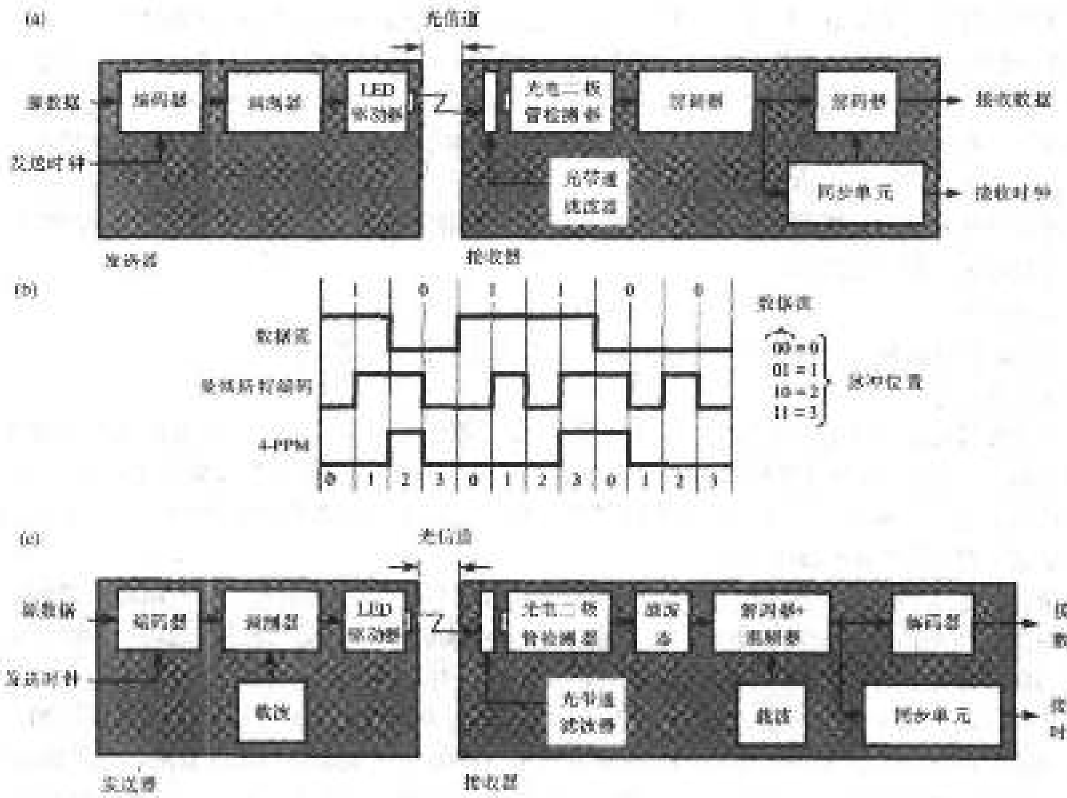


图6.31　红外线编码和调制方案

(a) 直接调制示意图　(b) 脉冲位置调制　(c) 载波调制示意图

6.4.3　介质访问控制方式

无线电和红外线是广播介质，就是说它们的所有传输都能被发送器覆盖范围内的所有接收器接收到。因此，就像我们需要使用MAC方式控制共享介质有线LAN（CSMA/CD，控制令牌等）确保只有一个发送器正在使用介质一样，在无线LAN中也需要MAC方式。使用的主要方案是CDMA、CSMA/CD、CSMA/CA、TDMA和FDMA。

1. CDMA

码分多路访问（CDMA）专门用于扩频无线电系统，如6.4.2节所述，直接序列和跳频都使用惟一的伪随机序列作为工作的基础。所以，在这种系统中给每个结点分配不同的伪随机序列，并且所有结点都知道序列完整集。为了同另一个结点通信，发送器简单地选择并使用将要接收方的伪随机序列。以这种方式，不同结点对之间的多个通信能并发进行。

实际上，如图6-32所示，只可能使用跳频系统，因为使用直接序列会产生称为近—远效
334 应的现象。当另一个发送器工作时就会出现这种现象。如图中结点X与将要接收结点A物理距离比另一个通信结点B更近。虽然来自结点X的发送会被结点A中的反扩散处理抑制，由于它更靠近些，(扩散)干扰信号比来自结点B的所需信号具有更大的能量，因此也使得结点A中的接收器错过传输。它也称为隐蔽终端效应。

相比之下，使用跳频，因为两个发送器不断改变信道频率，所以同一时刻两个发送器工作在相同信道的概率很小。它可以通过仔细规划跳频序列进一步减小。但是两个方案的缺点

是，所有结点都需要知道其他结点的伪随机序列，这在无线LAN很难管理。

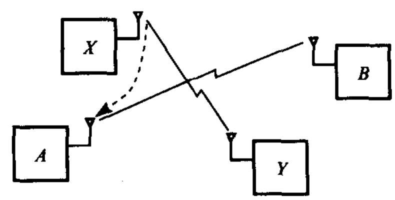


图6-32 无线MAC：CDMA近—远效应

2. CSMA/CD

6.1.3节描述了CSMA/CD（带冲突检测的载波侦听多路访问）以MAC方式广泛地应用在有线LAN中。在无线LAN中，CSMA还允许等待结点延迟另一个已经使用广播介质（无线电或红外线）的结点的发送。使用无线电和红外线，不可能同时发送和接收，因此基本形式的冲突检测不能使用。已经建议一种称为**冲突检测**（梳）的冲突检测功能的变型方案用在无线LAN中。

使用这个方案，当某个结点有帧要发送时，它先产生一个称为梳的短伪随机二进制序列并把它附加到帧前同步码的前面。然后结点以正常方式执行载波侦听操作。假定介质上无传输活动，它就开始传输梳序列。对于序列中的二进制1，结点在短时间间隔内发送信号；而对于序列中的二进制0，结点转换到接收模式。如果结点在接收模式期间检测到信号传输，那么它就放弃对信道的争用并延迟直到其他结点传送完帧。这个方案的工作原理如图6-33所示。

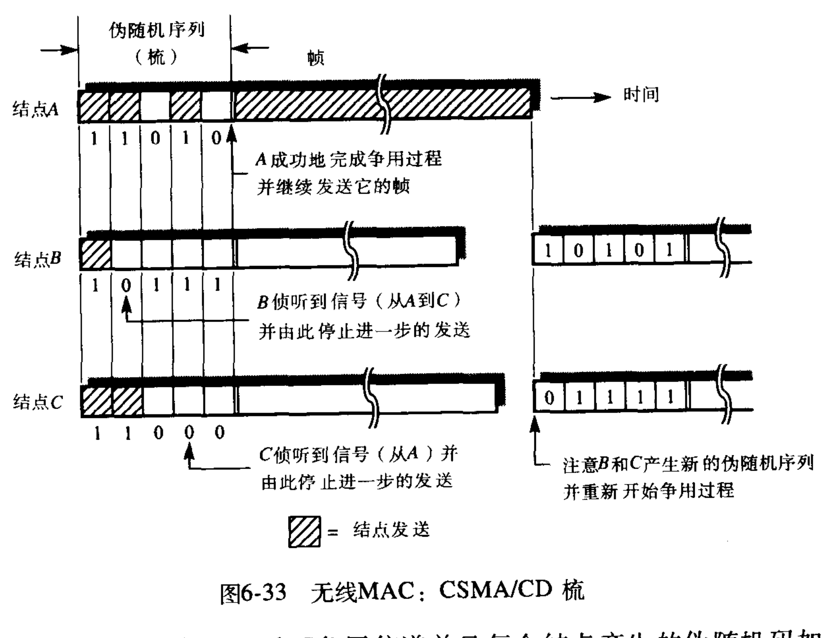


图6-33 无线MAC：CSMA/CD 梳

在这个实例中，三个结点A、B和C争用信道并且每个结点产生的伪随机码如图所示。因为所有结点的序列中的第一位是二进制1，所以没有结点侦听，并且无法检测到传输。在第二个梳时间间隔，结点A和C仍然发送而结点B处于接收模式，由此它能检测到信号，并在这个时间点上退出对信道的争用。在第三个时间间隔，因为结点B现在无活动，而结点A和C都处于接收模式，所以A和C都检测不到信号。在第四个时间间隔，结点A正在传送而结点C处于接 335

收模式，由此结点C侦听到信号并退出对通道的争用。然后在成功地完成剩余的争用过程后结点A留下来继续传送它的等待帧。

这个方案的效率取决于伪随机序列（梳）中的位数，因为如果两个结点产生相同的序列那么就会发生冲突。实际上，在任何时刻争用的结点个数可能很少，由此梳长度可以相对短。还有，因为对速率（无线电或红外线收发器以这个速率在发送和接收模式间转换，一般为1ms）有最大限制，所以较短的梳长度会减小争用解决（冲突检测）周期的时间。

3. CSMA/CA

还使用另一种称为**回避冲突的CSMA**——CSMA/CD的修改方案。它的工作原理如图6-34所示。

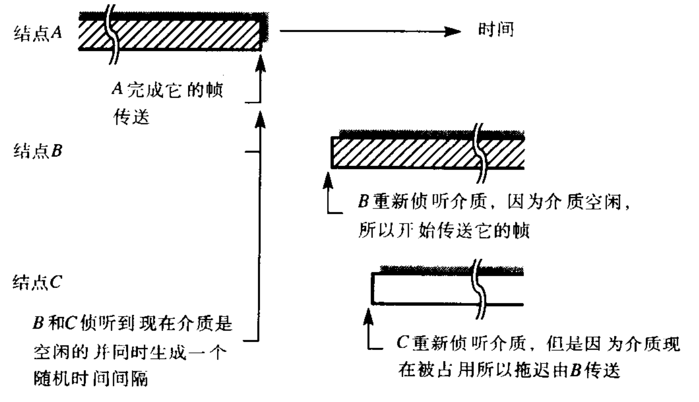


图6-34　无线MAC：CSMA/CA 协议

正如所见，与介质无活动时立即发起帧传输不同，首先结点进一步等待一个较短的随机时间间隔，并且只有在这个时间间隔后介质仍然无活动才开始发送。以这种方式，如果其他
336 结点也在等待，那么计算最短时间的结点先获得对介质的访问，而其余结点会延迟发送。还有，这个方案的效率是在最大冲突避免时间周期内的时间增量（由此伪随机序列中的位）的函数。

由于不能保证要通信的结点与源结点取得联系，当使用无线电（和红外线）时会产生另一个必须解决的问题。虽然CSMA/CA（或CSMA/CD）算法确保结点获得对介质的访问，但是由于帧的接收者超出无线电联系范围而永远不能收到它。所以除基本MAC方式之外，一个额外的握手规程加入到MAC协议中。因为它旨在用于不同类型的MAC方式，所以它称为**分布式基础无线MAC**（DFW MAC）协议。采用的四路握手规程如图6-35所示。它旨在用于基本结构和特殊结构应用中。

每当便携单元需要发送帧，它先使用刚才描述的一种MAC方式（CSMA/CD或CSMA/CA）发送一个短**请求发送**（RTS）控制报文/帧给它的PAU或者另一个便携单元。RTS控制报文含有源和目标单元的MAC地址，假定所需目标接收到请求并准备接收帧，它就广播一个**清除发送**（CTS）应答报文/帧（有地址对，但顺序与刚才相反）。另一种情况，如果目标不准备接收帧，它就返回一个**接收器忙**（RxBUSY）应答。如果应答是肯定的，那么请求单元就发送等待帧（DATA），如果它被正确接收，目标返回一个肯定的**确认**（ACK）报文。但是如果帧被损坏，那么会返回一个**否定的确认**（NAK）报文并且源单元尝试重新发送它。这个规程会重复直到规定的重试次数。记住识别的所有控制报文使用特定MAC方式发送。

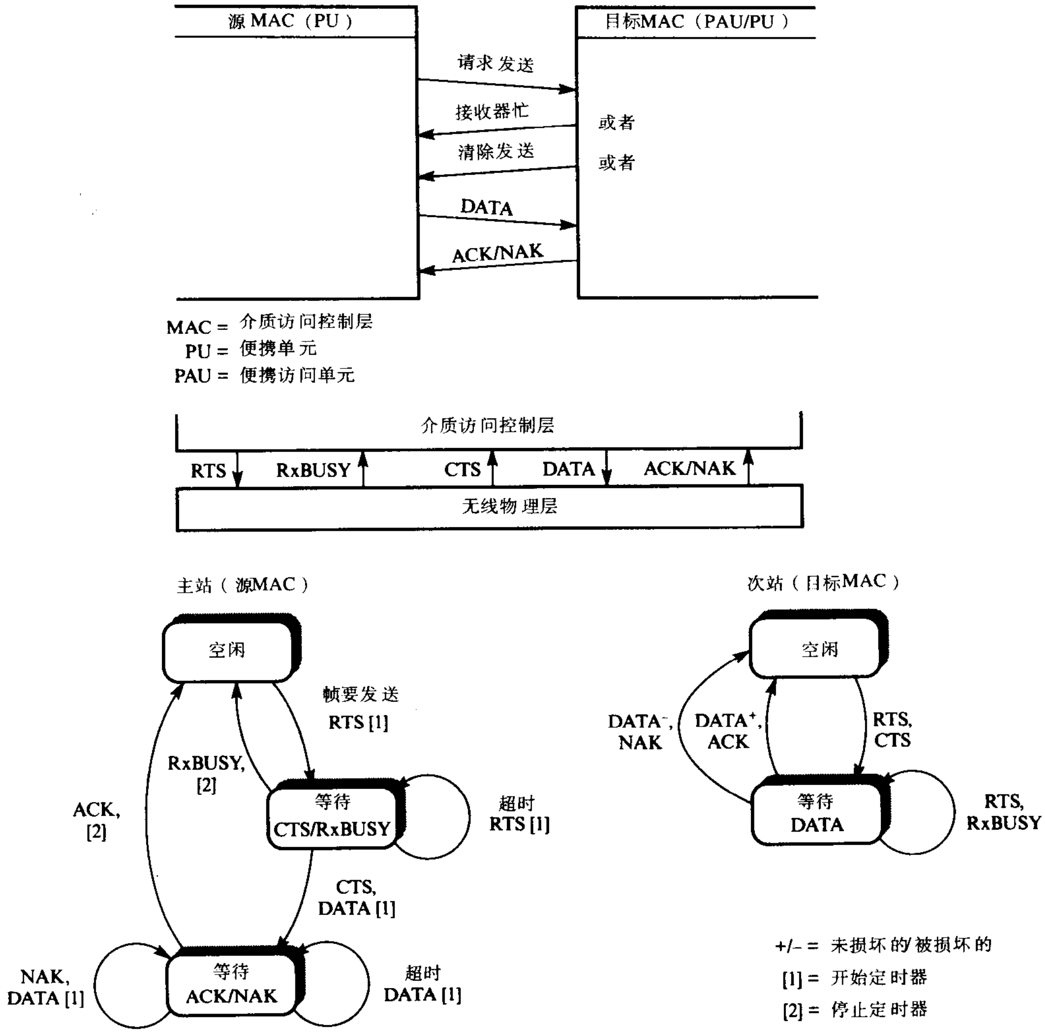


图6-35 无线MAC：DFW MAC协议中的四方握手规程

关于协议的状态迁移图也显示在图6-35中。回忆4.2.2节除了定义无差错状况下的协议工作机制外，还必须定义当损坏帧/控制报文出现时的操作。图中说明当后者发生时，计时器用 337
来发起损坏帧/报文的重新传输。通常，在每种情况使用规定好的重试次数。

4. TDMA

与无线LAN有关的时分多路访问（TDMA）的工作原理如图6-36所示。使用这种方法，每个发送器（结点）有个指定的时间间隔/时隙，并且一旦发送器的时隙到达，它在整个时隙（固定）持续时间内在全部带宽上发送。通常，每个时隙的持续时间较短，其内部出现传输差 338
错的概率就较低。帧/周期时段取决于每个时隙的持续时间以及支持的传输/时隙个数。

通常，当只有一个（基）站（所有传输通过它发生）时使用TDMA。例如，在固定线路取代应用PDU（见图6-24）充当基站，它负责建立时隙/计时结构。基站覆盖域内的每个便携计算机/终端分配一个指定时隙，或者更通常地，提供单独（信令）时隙，使得当每个便携设备有帧要发送时能向基站发出请求要求（空闲）时隙。从基站到便携设备的传输以使用指定

时隙的广播模式（每个传输帧的头部有所需接收者的地址）或者以使用信令信道建立的指定时隙形式发生。这种工作模式还称为**按需分配的时隙Aloha系统**。另一种情况，每个时隙的使用可以由单独信令子时隙控制。

如图6-36所示，在每个时隙的开始处有**防护频带**和**同步序列**。防护频带允许便携设备分布集和基站间的不同传播时延，而同步时间间隔允许接收器（便携设备或基站）能在接收到时隙内容前与发送器同步。

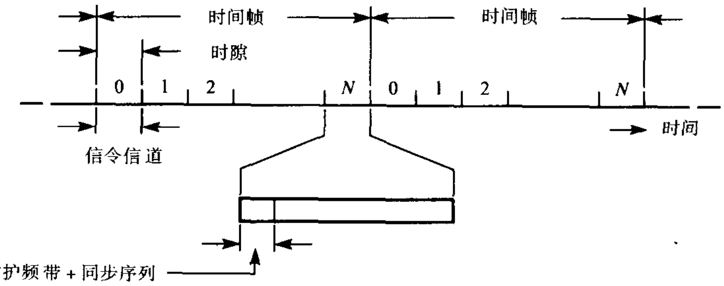


图6-36　无线MAC：TDMA

5. FDMA

339 频分多路访问（FDMA）的工作原理如图6-37所示。FDMA主要用于无线电系统，同TDMA一样，它需要基站来控制工作。使用FDMA，分配的总频率带宽被分成许多子带宽或信道，原理上类似于跳频扩频。但是在FDMA中，特定频率信道一旦被指定好，就在整个过程中用于帧传输。通常，频率信道使用单独的信令信道按需指定。

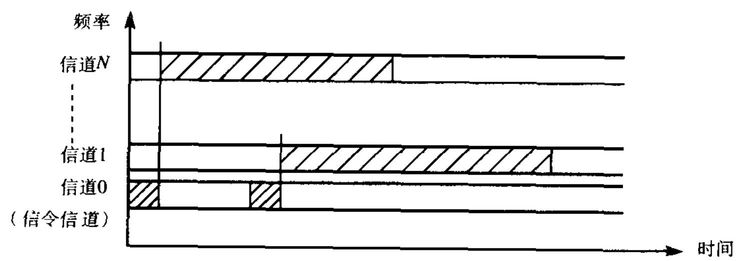


图6-37　无线MAC：FDMA

总体上，FDMA系统中的基站比TDMA系统中的基站更复杂，因此后者使用更广泛。也使用混合方案（如FDMA方案），产生多个频率信道，而每个频率信道使用TDMA。

6. *附加功能*

正如我们在6.5节中描述局域网协议时看到的，虽然有用于有线LAN和无线LAN的不同MAC协议，但是MAC层为其上面的层提供了一个服务原语的标准集合。在无线LAN情况下，意味着MAC层除了执行MAC功能之外还执行一些进一步的功能。包括分段、流控制和多速率处理。

因为无线电和红外线中具有明显较高的BER，所以分段是必需的。在不同的有线LAN类型中可能使用较长的帧长度，因为使用的传输介质（比如同轴电缆）的BER较低（一般为10^{-9}～10^{-11}）。相比之下，无线电和红外线的多路径干扰和其他干扰影响会显著地增加BER值（一般范围在10^{-3}～10^{-5}），这说明通过这些介质类型发送时必须使用较短的帧长度。如果MAC层旨

在提供类似于有线LAN提供的服务，MAC层必须把每个提交的帧分段成多个更短的子帧，使得能在无线介质上传输。类似地，接收到每个段，必须在递交前把它们重新组装成原始帧。

可能会把差错控制方案加入到DFW MAC协议中。但是，如果MAC层不执行差错控制，那么任何有差错的重新组装帧会被简单地丢弃。而且，有必要让MAC层执行流控制功能。当每个段传递给物理层时，MAC层必须在提交下一个段前等待直到这个段被传送。通常，在两层间的帧流，由前面图6-35所示的控制线路控制。

因为物理层经常以许多交替速率工作，因此多速率处理功能是必需的。例如，直接序列扩频使用1Mbps和2Mbps的速率，而红外线可能是1Mbps、2Mbps、4Mbps或者10Mbps。通常，工作速率由MAC层在与物理层服务原语相关的参数中指定。这个速率取决于物理层提供的服务质量，例如，如果较高比率的帧被损坏，那么就会选择一个较低的比特率，而如果没有帧被损坏那么可提高速率。当然接收物理层必须以与传送物理层相同的速率（和调制方法）工作，因此如果它们发生变化，通信双方必须达成一致。一般，它通过交换请求发送和清除发送帧中的附加控制参数完成。这个交换以较低速率进行并且只有接收者返回一个肯定的响应，发送者才向上运行到较高的速率。通常不同目标的当前工作速率由一张查询表保存以避免每次传输前速率的重新协商。 340

6.4.4 标准

当前有许多可用的无线LAN产品，但是它们都是由某一公司单独开发，所以互相之间有很大的差异。现在对于国际标准的需求得到了承认，并且当前有两个用于无线LAN产品的标准正在制定中。在美国研发的标准术语IEEE系列，称为**IEEE 802.11**。在欧洲，由欧洲电信标准机构（ETSI）研发的标准，称为**HiperLAN**。两个标准都使用了前面描述的许多特性。

像有线LAN一样，它们不仅仅是一个单一标准。例如，IEEE 802.11考虑了许多基于两种介质类型的不同物理层标准。它们包括如下：

1Mbps和2Mbps，使用跳频扩频无线电

1Mbps和2Mbps，使用直接序列扩频无线电

1Mbps和2Mbps，使用直接调制红外线

4Mbps，使用载波调制红外线

10Mbps，使用多子载波调制红外线

HiperLAN标准旨在用于基本结构和特殊结构两种应用中。一些操作参数仍在最后确定中，但是当前的规格说明如下：

用户比特率10～20Mbps

工作范围50m

无线电介质

使用改进后的称为**偏移QPSK**的正交相移键控和均衡器的单载波调制

CSMA/CD或CSMA/CA MAC方式

为了满足不同调制方式和介质类型的需要，物理层由两个子层组成：物理层会聚子层（PLC）和**物理介质依赖**（PMD）。PMD子层因调制方式和介质类型不同而异，并且它提供的服务也由这些决定。PLC子层执行会聚功能，它把物理层接口提供的标准服务映射成使用的特定PMD子层提供的标准服务。 341

6.5 协议

IEEE 802定义的用于LAN的各种协议标准，只涉及ISO参考模型中的物理层和链路层。这些标准定义了一组协议，每个协议涉及一种特定类型的MAC方式。各种IEEE标准以及它们与ISO参考模型的关系，如图6-38所示。

三个MAC标准以及它们相关的物理介质规格说明包含在如下的IEEE标准文档中：

- IEEE 802.3：CSMA/CD总线
- IEEE 802.4：令牌总线
- IEEE 802.5：令牌环
- IEEE 802.11：无线

ISO标准采用该组标准，只是在名称前附加一个8：如8802.3等。

到目前为止给出的描述已涉及到这四个标准的MAC层和物理层。虽然每个标准的内部机制不同，但是它们都为逻辑链路控制（LLC）层给出了标准的服务集，它们旨在与任何底层MAC标准关联。一般而言，像6.2节中提到的，各种MAC层和物理层通常在特定用途集成电路的固件中实现。所以，在本部分我们只集中讨论LLC和网络层并简单定义LLC层和MAC层之间的接口。注意在LAN中，网络层、LLC层和MAC层都是对等（端对端）协议，因为在网络本身内部没有中间交换结点（类似于公共数据网络中的分组交换机（见8.1节））。

342 我们在图6-38中看到，在ISO参考模型中MAC层和LLC层共同执行ISO数据链路（控制）层的功能。在这个环境中，MAC层和LLC层称为子层而不是层。回忆第5章，数据链路层的功能是建帧（发信号告知每个帧的开始和结束）和差错检测。还有，用于可靠（面向连接的）服务、差错控制、流控制和链路管理。这样MAC子层（和MAC操作一起）执行建帧和差错检测部分，而LLC子层执行剩余部分。

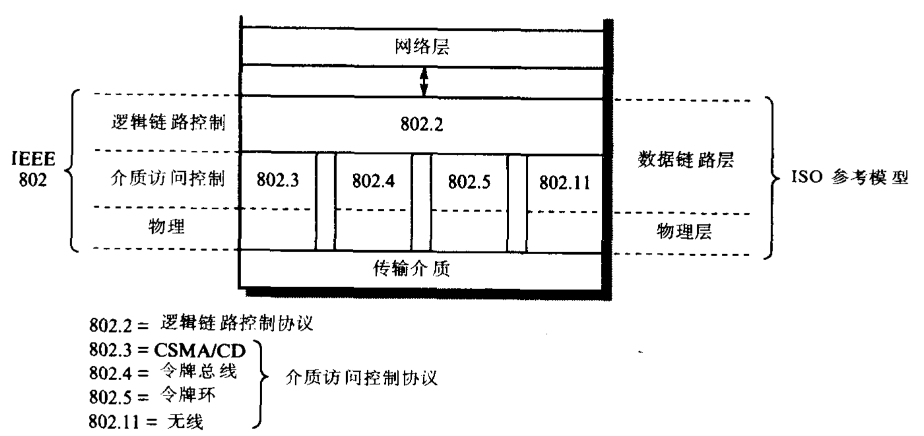


图6-38 IEEE 802协议集

6.5.1 MAC子层服务

不管底层MAC子层的工作方式——CSMA/CD、令牌环、令牌总线、无线——定义了一组用户服务标准集供使用，通过LLC子层向相应层传送LLC PDU。该组的用户服务原语有：

- MA_UNITDATA.request（请求）
- MA_UNITDATA.indication（指示）
- MA_UNITDATA.confirm（证实）

说明它们使用的时序图如图6-39所示。对于CSMA/CD LAN，证实（confirm）原语指示请求已成功（或不成功）发送，而对于令牌LAN，它指示请求已成功（或不成功）递交。

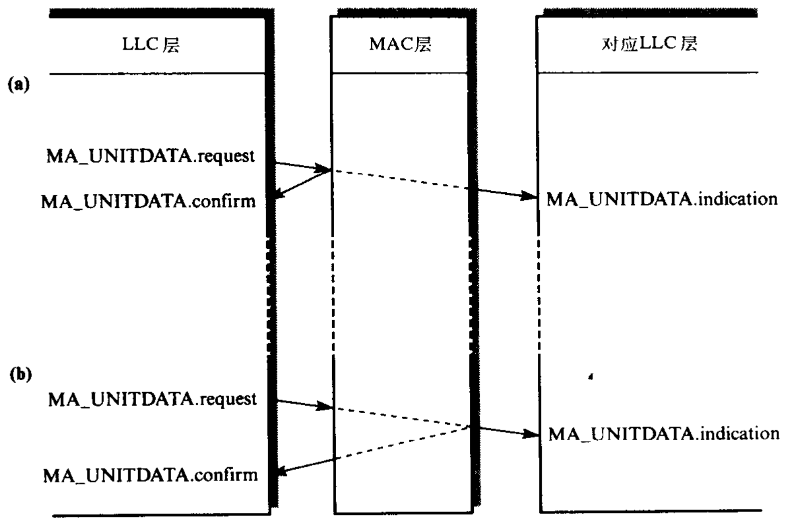


图6-39 MAC用户服务原语
(a) CSMA/CD (b) 令牌环/总线

每个服务原语均有相关的参数。MA_UNITDATA.request原语中的参数是请求的目标地址（它可以是单地址、组地址或广播地址）、服务数据单元（包含要发送的数据，即LLC PDU）和与PDU相关的所需服务类。最后一个参数用于令牌环和令牌总线网络，例如，当使用带优先级的MAC协议时。 343

MA_UNITDATA.confirm原语包括一个说明相关的MA_UNITDATA.request原语成功或失败的参数。但是，图6-39显示证实（confirm）原语不是远端LLC子层而是本地MAC实体作为响应的结果产生的。如果这个参数指示成功，说明MAC协议实体（层）已成功地传送服务数据单元到网络介质上。如果不成功，参数指明传输尝试为什么失败。例如，如果网络是CSMA/CD 总线型的，“过多冲突”可能是个典型的故障参数。

6.5.2 LLC子层

LLC子层的用户服务和工作机制在第5章描述数据链路控制协议时已经讨论过了。回忆一下LLC协议基于高级数据链路控制（HDLC）协议并支持两种类型的用户服务和有关协议：无连接和面向连接。但是，在几乎所有LAN应用中，尤其在科技和办公环境中，只使用**无确认发送数据**（SDN）无连接协议。由此仅有的用户服务原语是L_DATA.request，并且因为它是最佳尝试协议，所以所有数据使用无编号信息（UI）帧传输。LLC子层和MAC子层间的交互如图6-40所示。

L_DATA.request原语含有相关的参数。它们是源（本地）地址和目标（远端）地址的说明以及用户数据（服务数据单元）。后者是**网络层协议数据单元**（NPDU）。源地址和目标地址都是DTE的MAC子层地址和附加**服务访问点**（SAP）层间地址（LLC SAP）的连结，参见8.2.3节。 344

LLC子层和MAC子层间交互的更详细说明如图6-41所示。LLC子层从与L_DATA.request服务原语相关的两个地址参数中读取目标和源LLC服务访问点地址（DSAP和SSAP），并把它

345 们插入到LLC PDU的头部。然后把网络层协议数据单元（NPDU）加到这个PDU中，并把产生的LLC PDU作为MA_UNITDATA.request MAC原语的用户数据参数传递给MAC子层。与这个原语相关的其他参数包括MAC子层目标地址和源地址（DA和SA）、所需服务类以及用户数据字段中的字节数（长度指示）。一般而言，如果使用令牌网络，MAC子层协议实体使用服务类来确定与帧相关的优先级。

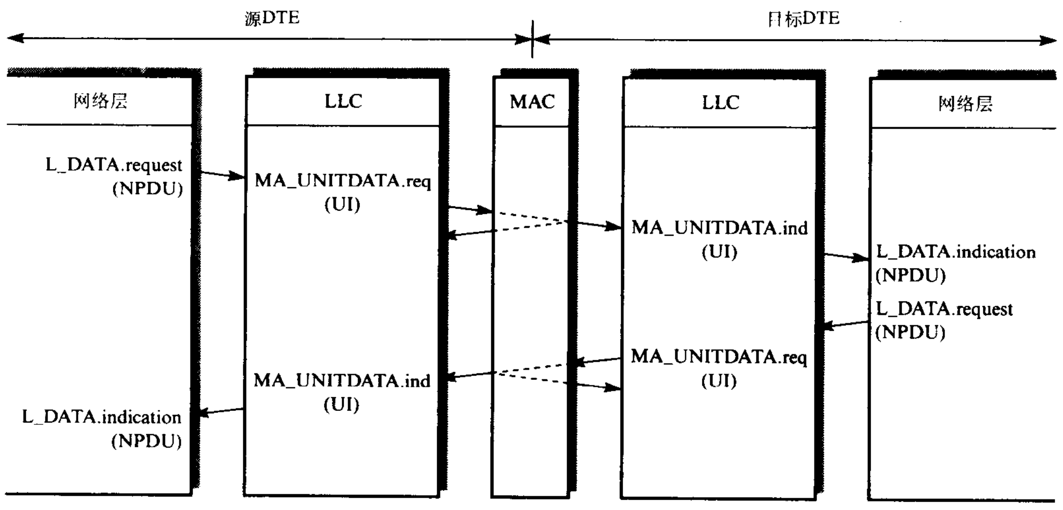


图6-40　LLC/MAC子层交互作用

MAC协议实体收到这个请求，产生一个准备在链路上传输的帧。在CSMA/CD总线网络情况下，它产生含有前同步码和SFD字段、DA和SA字段、I字段以及恰当FCS字段的帧。这个完整帧使用合适的MAC方法以位串行方式传送到电缆介质上去。

在目标DTE也遵循类似的规程，除了每个PDU中的相应字段由每层读取并解释。然后每个PDU中的用户数据字段与恰当的地址参数一起向上传递给下一层。我们会在讨论面向应用的协议后在第14章进一步讨论层间交互。

6.5.3　网络层

网络层的主要作用是在连接DTE分布群的网络中传送与其上更高协议层（ISO参考模型关系中）相关的报文。像数据链路层一样，网络层既能以无连接模式又能以面向连接模式工作。在LAN情况下，报文（帧）使用它们的**连接点**（MAC子层）**地址**在连到同一个LAN的DTE间编址路由。而且，因为LAN使用有较低BER的高比特率传输介质，与每个报文传输相关的DTE到DTE**转接时延**（transit delay）和报文损坏率可能较低。由此，当所有DTE都连到一个LAN上时使用无连接网络层服务和相关协议。然后把需要的差错控制和流控制留给它上面的传输层处理。

由于在LAN中它的功能缺乏，网络层通常被称为**非活动层**或**空层**。与网络层相关的用户服务原语及其参数如图6-42所示。

基本报文传输服务是N_UNITDATA（请求（request）和指示（indication）)，它是最佳尝试服务。与它相关的DA和SA参数是DTE（源或目标）的MAC子层连接点地址和LLC SAP层间地址扩展的连结。还使用一种称为**网络服务访问点**（NSAP）的层间地址扩展。它的作用与LLC SAP类似，允许报文路由通过各种协议层到同一个DTE内的不同AP（程序）。一个例子
346 是支持诸如电子邮件和文件传输等多种应用的网络服务器DTE。

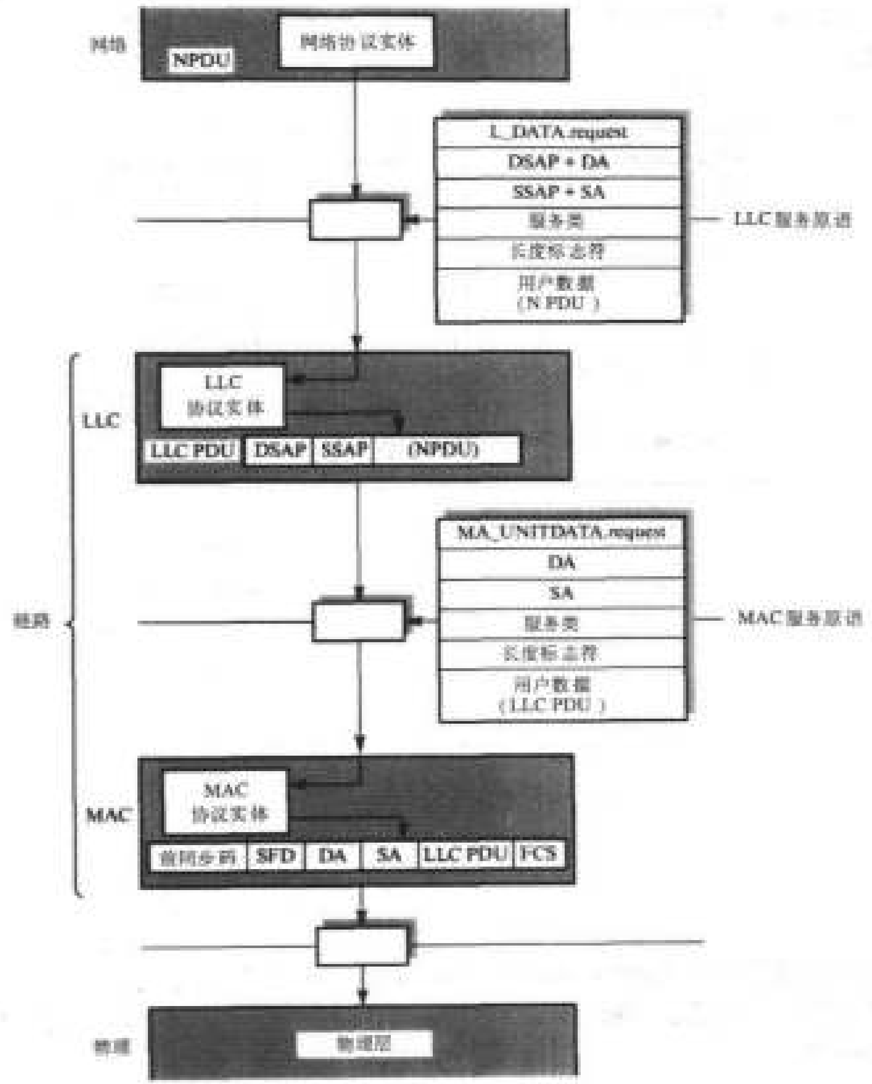


图6-41 层间原语及参数

一般而言，**服务质量**（QOS）参数包括允许指定转接时延、报文优先级和其他网络参数的字段。在单一LAN情况下，只有优先级字段有意义。最后，用户数据参数指向要传送的报文数据。

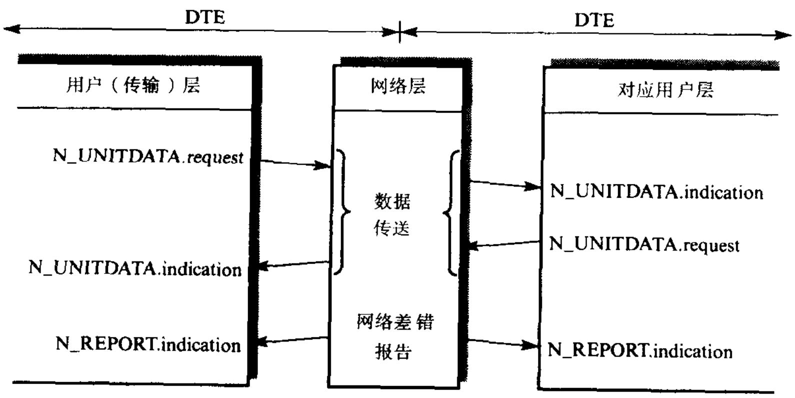
N_REPORT.indication原语由网络提供者（LLC子层和MAC子层）用来报告可能发生的关于传输请求的任何差错状况。在LAN情况下，一个例子是如果使用CSMA/CD LAN冲突过量。

所以我们可以总结出，与网络层相关的协议很少。它涉及根据与流入N_UNITDATA.request原语相关的参数产生NPDU，并把它放在L_DATA.request用户数据参数中传递给LLC子层。类似地，从LLC子层接收到NPDU（在与L_DATA.indication相关的用户数据参数中），协议从NPDU中取出源和目标网络地址，然后和剩余用户数据一起使用N_UNITDATA.indicaiton原语传递给用户（传输）层。

总之，注意如果网络由许多互连网络而不是单一网络组成，那么网络层协议会更加复杂。

整个网络称为网际互连或因特网，独立网络称为子网。我们会在第9章讨论网际互联时进一步讨论网络层。

(a)



(b)

服务原语	参　数
N_UNITDATA.request/indication	目标地址 源地址 服务质量 用户数据
N_REPORT.indication	原因代码

347 图6-42　网络层服务

(a) 时序图　(b) 服务参数

习题

6.1 列出当前普遍用于LAN的四种主要网络拓扑类型，并画图解释它们的操作。

6.2 借助图和相关文字描述一组DTE如何使用下列传输介质互连形成CSMA/CD LAN：

（a）双绞线

（b）细同轴电缆

（c）粗同轴电缆和分支电缆

6.3 解释同轴电缆LAN中的术语“宽带工作方式”的含义。画出典型宽带LAN，说明所需的主要连网组件并解释它们的功能。描述这种网络的完整操作，以及如何从一条电缆得到多个数据传输业务。

6.4 描述下列用在LAN中的MAC方式的操作原理：

（a）CSMA/CD

（b）控制令牌

（c）分槽环

6.5 解释下列与CSMA/CD相关的术语的含义：

（a）时隙间隔

（b）阻塞序列

（c）截断二进制指数退避

6.6 描述控制令牌MAC方式的操作原理，并用图解释它如何用于总线和环网络拓扑。清楚说明与每种拓扑相关的各个控制字段。

6.7 说明用在分槽环中的典型帧的结构和内容。描述帧中每个字段的含义以及相关的环协议的操作。清楚地解释对环的访问是如何由组成环的站（DTE）共享的。

6.8 画图说明连接DTE（站）到粗缆CSMA/CD总线网络所需的组件。给出这种网络整个操作关系中每个组件的功能描述。说明每个传输帧的结构和内容并解释每个字段的含义。

6.9 借助图解释使用CSMA/CD MAC方式时冲突是如何发生的。

解释术语“帧间空隙”和“阻塞序列”的含义，并用流程图的方式描述MAC子层发送和接收部分的操作原理。

6.10 解释有关集线器布线配置的CSMA/CD MAC方式的操作。

6.11 画出示意图说明连接DTE（站）到令牌环网络所需的组件，并给出每个组件的功能描述。详细说明一旦连接到网络，DTE如何以插入或旁路方式工作。还要说明线路集中器的位置和功能。

6.12 用一系列图解释有关令牌环网络的令牌MAC方式的操作原理。

清楚地描述两种可选令牌释放方式。

6.13 描述用于令牌环LAN的令牌帧和信息帧的结构和内容。解释两种帧类型内每个字段的含义，并借助流程图描述MAC子层的发送和接收部分的操作原理。

6.14 解释用于令牌环LAN的故障检测方法。还要解释如何通过引入冗余环提高LAN的可靠性。 348

6.15 解释用在令牌环网络中的下列术语的理解：

（a）最小延迟时间

（b）令牌持有时间

（c）差分曼彻斯特编码

6.16 描述令牌环网络用来控制不同优先级帧传送到环上的顺序的优先级控制方案的操作。在描述中包括下列功能：

（a）每个帧含有的优先级位和保留位

（b）每个站保留的优先级寄存器和堆栈

（c）堆栈站

6.17 说明下列用于令牌环网络的环管理规程的目的，并解释它们的操作：

（a）初始化

（b）备用监控站

（c）活动监控站

（d）告警

6.18 假定四个站（A、B、C和D）互连到一个令牌环网络上。在一段静默（无活动）后，站A和B都有优先级为2的帧要发送而站C和D有优先级为4（最高）的帧要发送。

假定站A恰好获得令牌并开始发送它的等待帧，在下六个帧/令牌沿环循环周期内跟踪优先级字段和保留字段的状态。还要注明堆栈站采取的动作。

6.19 讨论下列因素对用于无线LAN的无线电接收器设计的影响：信噪比、热噪声、信号带宽、覆盖范围、发送器能量级别。

6.20 讨论涉及无线电环境和无线LAN的下列各项：

邻近信道干扰以及如何减小这个干扰、多路径及其影响、均衡、多向天线。

6.21 得出波长为（a）800nm和（b）1300nm的红外线发射的频率。

6.22 讨论涉及红外线系统的下列各项：

（a）为何邻近信道干扰低于无线电

（b）光过滤器的需要

（c）红外线源在能获得的最大比特率下的调制带宽的作用

6.23 借助图解释红外线LAN的下列工作模式的操作原理以及如何减少多路径扩散的影响：

（a）点对点

（b）无源卫星

（c）有源卫星

6.24 解释直接序列扩频无线LAN的操作原理，以及为什么这种应用能与相同频带的其他用户共存。还要解释在无线LAN中为什么所有结点都需要MAC协议。

6.25 一个伪随机二进制序列产生器由一个4位移位寄存器、*ABCD*和一个异或门组成。如果*A*输入是*C*和*D*输出的异或并且初始内容是1111，得出移位寄存器的状态并由此得到输出序列。它是最大的长度吗？

6.26 说明下列有关直接序列扩频系统的术语：扩展序列、时间片、片率、扩展因素、处理增益。

6.27 用图解释如何在直接序列扩频系统中达到符号同步。比如使用序列0101001100000。求出这个序列的自动相关。

6.28 用图解释快速和慢速跳频扩展频谱系统的操作原理。清楚解释每个方案中在每个载波

349 （当它活动时）上传输的信息以及接收器如何确定传输位序列。

6.29 讨论在工作频带内存在强窄带干扰信号时跳频相对于直接序列的优点。

6.30 讨论用于高比特率无线LAN的单载波调制方案与多子载波方案的优缺点。

6.31 区别有关红外线传输系统的直接调制和载波调制。

6.32 用波形图说明如何使用16-PPM方案调制位序列。

6.33 借助图解释用于无线LAN的CSMA/CD（梳）MAC方式的操作原理。说明决定这个方案效率的因素。

6.34 借助图解释用于无线LAN的CSMA/CA MAC方式的操作原理。说明决定这个方案效率的因素。

6.35 说明为什么在无线LAN中需要附加规程（除了基本MAC方式外）确保无线介质上的成功传输。由此解释用在分布基础MAC协议中的四路握手规程的操作原理。

6.36 定义令牌总线网络术语“时隙间隔”的含义，并解释在正常操作和异常操作期间用于这种网络的令牌传递规程。在描述中包括下列条件：

（a）桥接绕过故障DTE

（b）允许新DTE进入工作逻辑环

（c）在环刚建立时产生新令牌

6.37 解释每个DTE持有的下列变量（它们用来控制不同优先级帧在令牌总线网络上的传输顺序）的功能：

（a）高优先级令牌持有时间

（b）令牌循环定时器

（c）目标令牌循环时间

画图并附带实例进行描述，说明当传输帧以及令牌沿环循环时令牌循环定时器如何变化。假定只有两个优先级级别，从实例中推断每个级别可用传输容量的比例。

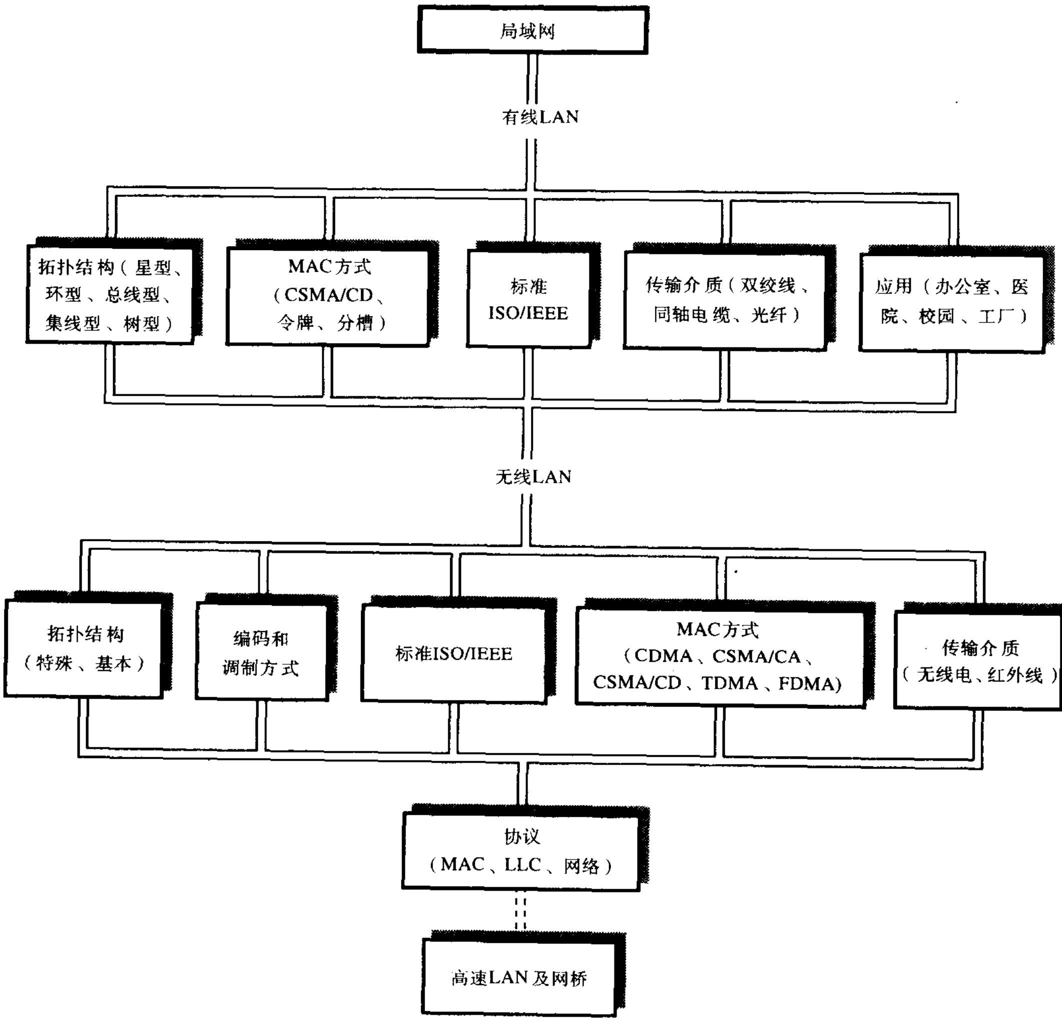
6.38 (a) 概要说明在IEEE 802标准文档中定义的LLC和MAC协议层的功能，并说明它们与ISO参考模型关系中较低协议层的关系。定义用于LLC和MAC层的用户服务原语典型集，并画出时序图说明每个LLC原语如何使用定义好的MAC服务实现。

(b) 解释当LLC层接收到L_DATA.request服务原语时发生的层间交互。清楚说明与每个原语相关的参数。

6.39 解释无连接网络层的功能。使用时序图说明这个层提供的服务原语以及与每个原语相关的参数。

6.40 假设使用图6-23(a)和(b)中的图，解释CSMA/CD、令牌总线和令牌环LAN的相对性能。 350

本章概要


局域网
有线LAN
拓扑结构（星型、环型、总线型、集线型、树型）
MAC方式（CSMA/CD、令牌、分槽）
标准 ISO/IEEE
传输介质（双绞线、同轴电缆、光纤）
应用（办公室、医院、校园、工厂）
无线LAN
拓扑结构（特殊、基本）
编码和调制方式
标准ISO/IEEE
MAC方式（CDMA、CSMA/CA、CSMA/CD、TDMA、FDMA)
传输介质（无线电、红外线）
协议（MAC、LLC、网络）
高速LAN及网桥

351

第7章 高速桥接局域网

本章目的

读完本章，应该能够：

- 描述以太交换LAN及两种派生高速以太网的操作；
- 描述新IEEE 802.12标准及FDDI高速LAN的操作；
- 理解网桥的功能和基本结构；
- 了解网桥相对于中继器的优势；
- 描述透明桥接LAN的操作及其生成树算法；
- 描述源路由选择桥接LAN的操作及其路由查找规程；
- 描述生成树桥接LAN和源路由选择桥接LAN的差异和优缺点。

引言

局域网（LAN）相关标准的快速建立，连同计算机接口廉价芯片集的主要半导体生产商的大力推动，意味着LAN构成了所有商用、研究和大学数据通信网络的基础。

随着LAN应用的发展，对其数据吞吐和可靠性的需求也随之提高。例如，一个早期的LAN是使得个人工作站（如个人计算机）的分布式群体能访问一台电子邮件服务器或者一台激光打印机。这些应用包含较少的事务处理，因此LAN传输带宽需求也很低。但是，应用对 352
高带宽的需求最近显著增长。例如，无盘工作站本地群体共享公用（网络）文件系统，意味着每个工作站访问文件都要通过网络进行，由此显著地增加对网络带宽的需求。同时，高分辨率图像文档传输的复杂应用变得越来越普遍，这大大增加对LAN容量的需求。

为满足这些需求，已经开发出多种高速LAN。它们包括基本CSMA/CD（以太网）LAN的变型版本。这是到目前为止应用最广泛的LAN类型。厂商的目的是在现有软件和电缆安装的最小改变下，得到更高的性能。基本LAN的变型版本，一个称为**交换以太网**，另一个称为**快速以太网**。第三个是IEEE 802.12标准。像快速以太网一样，它也是在现有电缆安装基础上开发的版本。但是，IEEE 802.12使用不同的MAC协议。

多数早期LAN应用只由单一LAN网段组成，工作站的分布式群体和相关的外设服务器都附接在这个LAN网段上。但是如第6章指出，能够附接到单一LAN网段的站（DTE）个数和物理长度都是有限制的。因此，随着LAN的接受和应用越来越普遍，出现了由多个网段连接组成的LAN应用。这种趋势持续发展，目前多数大型LAN就是这种类型。小型、典型的跨地区LAN如图7-1所示，它给出了一些不同的互连设备和拓扑。

正如在第6章中所见，互连LAN网段的基本方法是使用物理层中继器。然而使用中继器使得每个站的帧传输会在整个网络中传播（由此加载到整个网络上），即使在这些帧中，许多打算发往的目标站与始发站处于同一网段。这意味着这种先进的应用会给整个网络带宽带来巨大负载，尽管每个网段中只有一小部分通信量发送给其他网段的系统。为了解决这个问题，引入称为**网桥**的设备，它作为互连LAN网段的可选方案。基本网桥只能互连两个网段，但更

复杂的网桥——称为**多端口网桥**——用来互连更多数量的网段。通常网段间只有相对很短的物理距离，因此多端口网桥主要应用于单一办公室综合体。

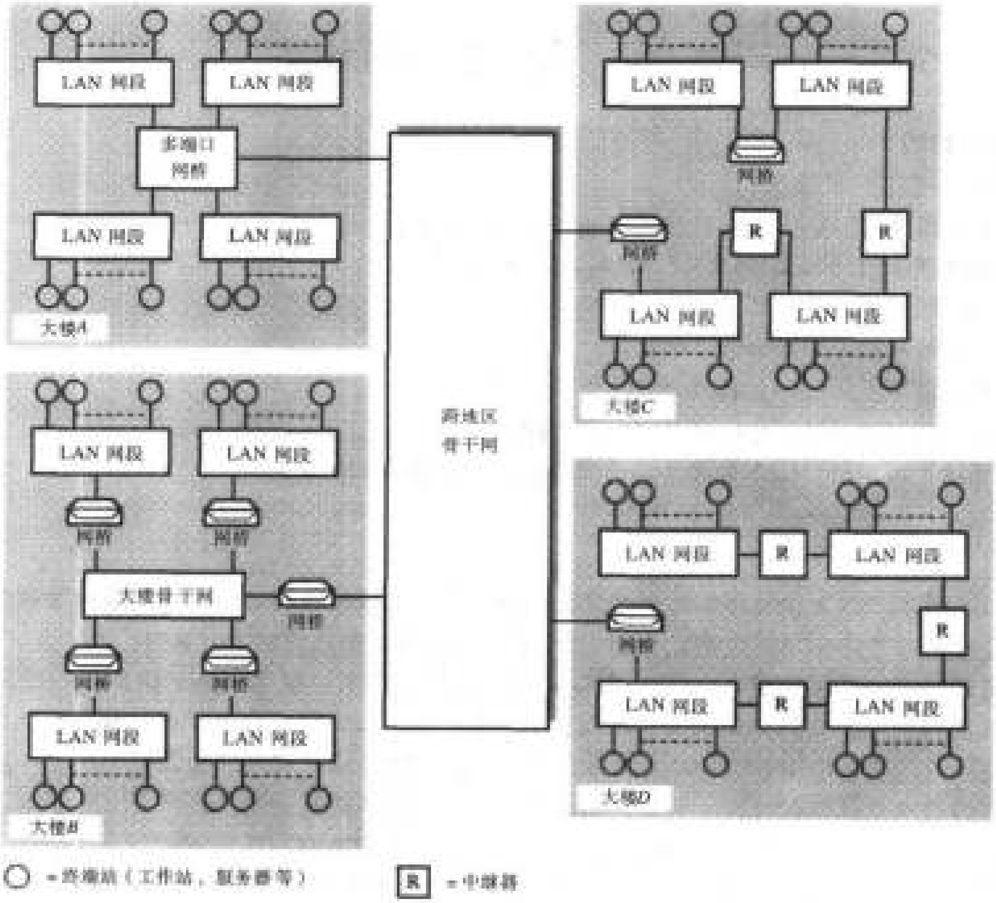


图7-1　典型跨设施LAN

使用网桥的一种可选方式是建立骨干子网。通常没有终端系统（工作站、服务器等）连
353 到骨干网上，它们只是用于网段间通信。例如只互连单一大楼中的少量网段，就使用像互连网段（CSMA/CD、令牌环或令牌总线）的骨干网，我们称之为**大楼骨干网**。

随着互连网段数量的增加，骨干网用于满足网段间通信量所需传输带宽开始超出基本LAN类型可用带宽。为了解决这个问题，使用基于更新高速LAN类型的骨干网。一个例子是**光纤分布式数据接口**（FDDI）局域网。它支持基于光纤的传输比特率为100Mbps的环型网络。它能用来互连比单一大楼覆盖地理区域（如大学校园或制造工厂）更广的网段。建立的网络
354 称为**地区骨干网**。本章会描述这类高速LAN的每一种以及现在成为国际标准的两种网桥。

7.1　交换以太网

应用最广泛的LAN类型是基于CSMA/CD访问协议的LAN。它定义在IEEE 802.3/ISO 8802.3中，鉴于历史原因它称为以太网。早期应用使用粗同轴电缆，或者细同轴电缆（在实验室或其他类似场所）。最近的应用使用双绞线和集线器。如图6-9所示，它基于星型拓扑，

比如集线器位于布线盒中。在其覆盖域内的DTE/站通过语音级双绞线连到其上。

可以看到，有单独的发送和接收线对，集线器中的中继器装置把从任何一对输入线上收到的信号转发（重新发送）到所有输出线对上。它仿效了使用同轴电缆的广播传输模式，并允许以正常方式通过每个附接DTE检测冲突。显然，在任何时刻只有一个传输活动在进行。

增加中继器装置的复杂程度，集线器就能够以非广播模式工作。因为集线器转发每个帧，所以通过读取转发的每个帧头部的源地址，可以知道附接在它每个端口上的DTE的MAC地址。这样集线器能建立一张（路由选择）表，它含有附接在它每个端口上的DTE的MAC地址。一旦这个规程完成（在每个附接DTE的第一次传输活动后），中继器装置在接到每个接收帧头部的目标MAC地址时就能把该帧只转发给地址所在的那个端口。如果该帧不是发给任何附接在本集线器上的DTE，就把它转发到下一个集线器链路上。这就是交换以太网的原理。

这种方法的潜在优势在于，假设传输发生在不同DTE间时，可以通过集线器同时传输多个帧。但是这需要集线器底板能同时转发多个帧。实际上，这有很多种方法，但最简单的方法是为每个DTE提供一条单独的（串行的）底板总线线路。收到每个帧，帧通过端口自己的底板总线线路中继到所需输出端口。图7-2(a)给出这种方法的示意图。

每个端口输入线路末端是先进先出（FIFO）缓冲器，所有流入帧的内容通过它转发。在侦听模式下，在FIFO缓冲器接收到帧头部的源地址后，控制处理器读取该地址，并在端口地址的路由选择表中增加一条端口号及相应DTE地址的记录。然后控制处理器通过FIFO缓冲器在底板总线线路中发起完整帧的转发。

一旦控制处理器知道每个端口的MAC地址，接到所有后继帧时，它简单地从每个帧头部
读取目标MAC地址，在路由选择表中查询后决定相应目标端口号，并使用相应的底板总线线 355
路向这个端口发送帧。对于组地址或广播地址，转发这个帧的副本到多条线路上。类似地，在多集线器网络中，如果目标MAC地址不属于本集线器的端口/DTE，那么该帧就通过连到下一个集线器的端口转发。

最后考虑冲突检测。发生冲突的惟一可能是，当接收到的帧需要转发到目标端口时，发
现所需目标端口正从另一个端口接收帧。考虑到这种情况，允许使用一条额外线路（线对） 356
发信号给发送DTE，告知冲突已发生。实际上可以轻易地得到这条额外线路，因为对于每个附接DTE，非屏蔽语音级双绞线通常有四对线。

因此可以推断，尽管多个传输活动能并行进行，但是每个传输只能以10Mbps的速率进行。还有，在许多工作组情况下，单一服务器DTE被多个（客户端）DTE共享。所以大多数传输涉及服务器。显然，因为在某一时刻只能有一个传输涉及服务器，因此获得的性能是有限的。

为了克服这种限制，已经开发了一种基本交换集线器的派生产品，它提供一个以高于其他端口传输速率工作的端口。通常，它用来连接两个集线器或者把服务器连到集线器上，如图7-2(b)所示。在这种方案中，通过提供更多的内存以及以不同速率操作FIFO缓冲器的输入输出端口来执行变速操作。从一个客户端DTE输入一个帧，该帧在以较高速率输出到服务器端口前全部存储起来。类似的，在反方向上，服务器FIFO能缓冲许多帧，然后每个帧以较低速率输出。这意味着服务器能支持多个并发事务处理，其中每一个事务工作在10Mbps速率下。相似情况下，当高速端口用来连接两个集线器时，多个帧的传输能同时进行。

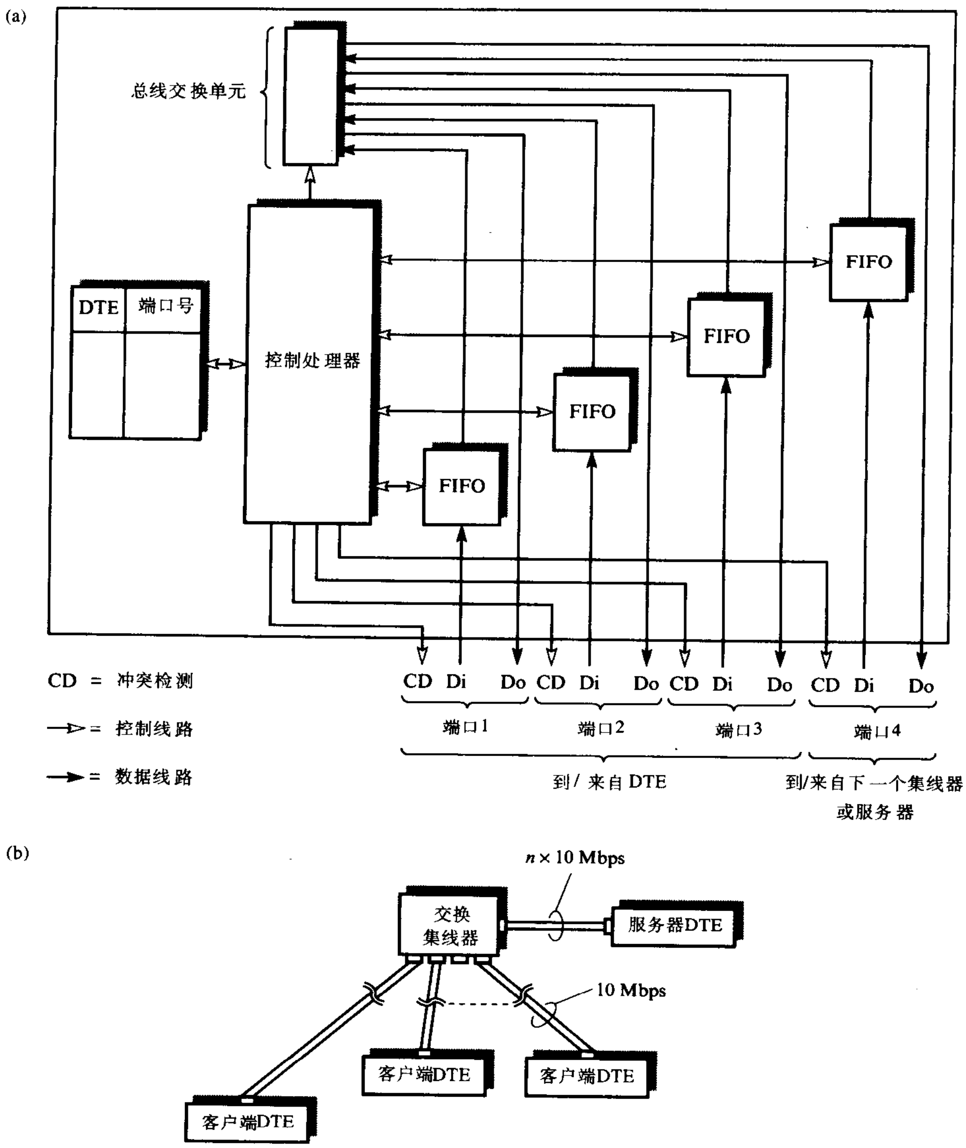


图7-2 交换以太网

(a) 交换集线器示意图 (b) 派生的交换集线器

7.2 快速以太网

快速以太网的目的是使10 Base T以太网（IEEE 802.3）的速度能获得一个数量级增长，而同时保留同样的布线系统、MAC方法和帧格式。

IEEE 802.3规范考虑了2.5公里的总电缆长度限制（用于中继器）。最坏情况的信号传播时延是信号传播两倍该长度所需的时间。标准允许50μs的最坏情况信号传播时延（包括转发时延），它等同于以10Mbps速率传播500位的时间。加入安全边际就给出512位的最小帧长度。显然，最大长度减少了，CSMA/CD访问方式就能以更高传输比特率工作。这就是快速以太网标准的基础。

实际上，绝大多数10 Base T应用使用短于100米的电缆来连接每个DTE到集线器。这意味着任何两个DTE的最长距离是200米，因此用于冲突检测目的的最坏情况路径长度是400米。显然，在仍保留同样CSMA/CD MAC方法和512位的最小帧长度情况下可使用更高的传输比特率。在标准中，数据传输比特率设置为100 Mbps，由此该标准又称为100 Base T。

快速以太网的主要问题是如何在100米非屏蔽双绞线（UTP）上获得100Mbps数据传输率。实际上，有两个标准，一个旨在用于语音级3类电缆而另一个可用于高质量5类电缆、屏蔽双绞线（STP）或者光纤。第一个标准称为100 Base 4T而第二个标准称为100 Base X。图7-3(a)给出了两个标准协议结构的示意图。 357

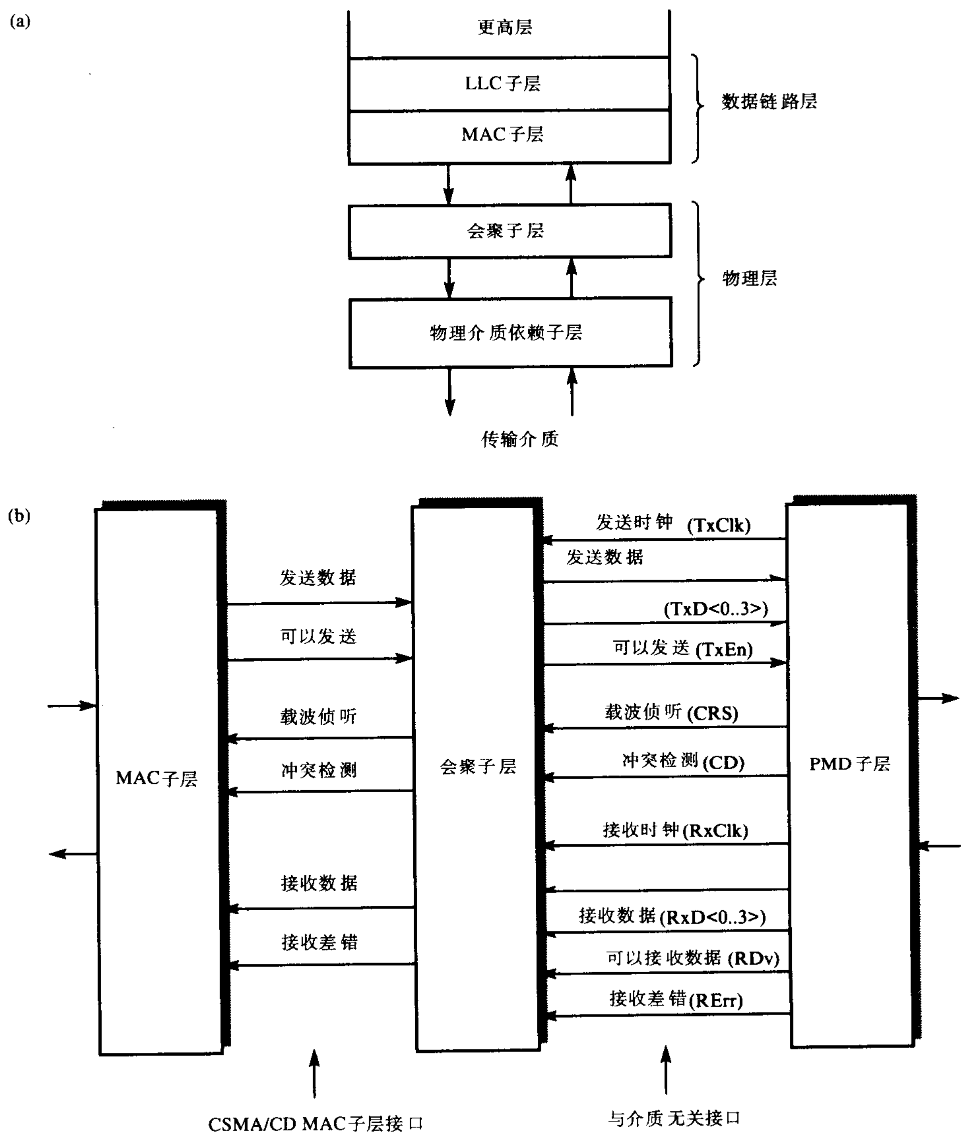


图7-3 100 Base T 协议结构

(a) 协议子层 (b) 接口信号

可以看到会聚子层（CS）为CSMA/CD MAC子层与底层物理介质依赖（PMD）子层之间
358 提供接口。CS的作用是利用更高比特率和对MAC子层透明的各种不同介质。为了使用不同的介质类型，已经定义了**介质无关接口**（MII），用在会聚子层和PMD子层之间。图7-3(b)给出了与该接口相关的信号集。

在100Mbps的数据传输速率下，使用时钟编码（例如曼彻斯特编码）是不可行的，因为由此产生的高时钟频率不适用于UTP电缆的有限集。所以采用了位编码方案。该方案确保每个经编码的符号有足够的跳变来让接收方保持时钟同步。

两个标准中的编码方案使用一组或多组4位数据产生每个编码符号。因此，所有经过MII传输的数据都是4位半字节。其他控制线路关注这些半字节在接口的可靠传输。所以，CS的主要功能是在MAC子层接口的发送和接收串行数据流和经过MII传送的4位半字节间进行转换，并把PMD子层产生的载波侦听和冲突检测信号转发到MAC子层。两个标准使用不同的PMD子层获得100Mbps的数据传输速率。我们会更详细地讨论两个标准。

7.2.1　100 Base 4T

3类UTP电缆包含四对单独双绞线。为了减少每对线上使用的比特率，在100 Base 4T中四对线共同用来获得每个方向上所需的100Mbps数据传输率。这就是“4T”的由来。

使用CSMA/CD访问控制方法，当无介质争用时，所有传输都是半双工的，就是说或者从DTE到集线器或者从集线器到DTE。在一个10 Base T应用中，四对线中只有两对线用于数据传输，每个方向一对线。当发送DTE（或集线器）在发送线对上发送数据时，检测到接收线对上有信号就检测到冲突。因为冲突检测功能必须在100 Base T中执行，相同的两对线用于这个功能。剩余的两对线以图7-4（a）所示的双向模式工作。

图中显示在每个方向上使用三对线进行数据传输，1、3和4线对用于DTE与集线器间的传输，而2、3和4用于集线器与DTE间的传输。像10 Base T一样，1、2线对上的传输用于进行冲突检测和载波侦听。这意味着每对线上的比特率只需要33.33Mbps。

1. 线路码

如果使用曼彻斯特编码，33.33Mbps的比特率需要33.33MHz的时钟频率，它超过用于这类电缆的30MHz的限制集。为了减少这个时钟频率，使用3电平（**3元）码**而不是标准（2电平）二进
359 制编码。使用的这种代码称为**8B6T**，它意味着在传输前每组8个二进制位先转换成6个三元（3电平）符号。从图7-4（b）所示的例子中能推断出它获得的符号信号速率为：$\frac{100\times6/8}{3}=25\text{MHz}$，它刚好在限制集内。

使用的三个信号电平是$+V$、0、$-V$，它们简单地以+、0、−代表。码字是经过选择的，这样线路才能是电平衡的，就是说平均线路信号为0。它使得接收方对三个信号电平的区别能力最大化，因为它们总是相对于恒定的0（DC）电平而言的。为了达到这个目的，可以利用6个三元符号使用中存在的固有冗余。6个三元符号意味着有729（3^6）种可能码字。因为代表8位字节组合的完整集只需要256个码字，使用的码字要经过选择，首先是获得电平衡，其次确保所有码字中至少有两个信号跳变。这样做是为了使接收方能保持时钟同步。

360 为了满足第一个条件，我们只选择组合电平权重为0或+1的码字，267个码字符合这个条件。为了满足第二个条件，我们除去那些少于两个信号跳变的码字（5个码字）以及以四个连续0开始或结束的码字（6个码字）。这就剩下了需要的256个码字了，它们列在表7-1中。

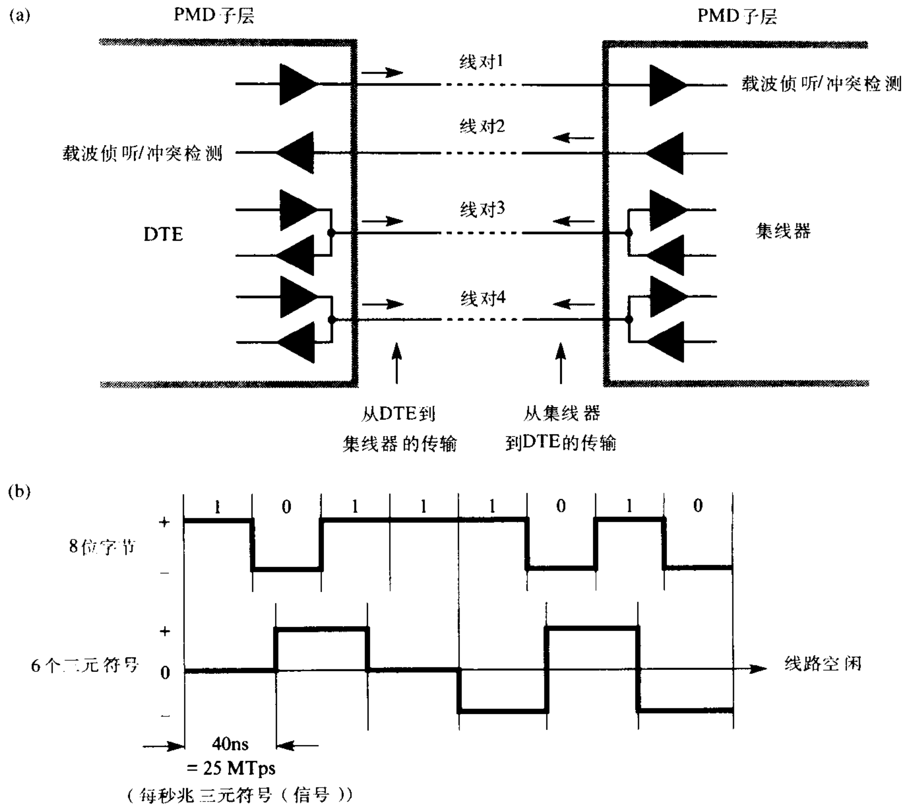


图7-4 100 Base T

(a) 线对的用法 (b) 8B6T编码

表7-1 8B6T码字集

数据字节	码字	数据字节	码字	数据字节	码字	数据字节	码字
00	− + 0 0 − +	20	− + + − 0 0	40	− 0 0 + 0 +	60	0 + + 0 − 0
01	0 − + − + 0	21	+ 0 0 + − −	41	0 − 0 0 + +	61	+ 0 + − 0 0
02	0 − + 0 − +	22	− + 0 − + +	42	0 − 0 + 0 +	62	+ 0 + 0 − 0
03	0 − + + 0 −	23	+ − 0 − + +	43	0 − 0 + + 0	63	+ 0 + 0 0 −
04	− + 0 + 0 −	24	+ − 0 + 0 0	44	− 0 0 + + 0	64	0 + + 0 0 −
05	+ 0 − − + 0	25	− + 0 + 0 0	45	0 0 − 0 + +	65	+ + 0 − 0 0
06	+ 0 − 0 − +	26	+ 0 0 − 0 0	46	0 0 − + 0 +	66	+ + 0 0 − 0
07	+ 0 − + 0 −	27	− + + + − −	47	0 0 − + + 0	67	+ + 0 0 0 −
08	− + 0 0 + −	28	0 + + − 0 −	48	0 0 + 0 0 0	68	0 + + − + −
09	0 − + + − 0	29	+ 0 + 0 − −	49	+ + − 0 0 0	69	+ 0 + + − −
0A	0 − + 0 + −	2A	+ 0 + − 0 −	4A	+ − + 0 0 0	6A	+ 0 + − + −
0B	0 − + − 0 +	2B	+ 0 + − − 0	4B	− + + 0 0 0	6B	+ 0 + − − +
0C	− + 0 − 0 +	2C	0 + + − − 0	4C	0 + − 0 0 0	6C	0 + + − − +
0D	+ 0 − + − 0	2D	+ + 0 0 − −	4D	+ 0 − 0 0 0	6D	+ + 0 + − −
0E	+ 0 − 0 + −	2E	+ + 0 − 0 −	4E	0 − + 0 0 0	6E	+ + 0 − + −
0F	+ 0 − − 0 +	2F	+ + 0 − − 0	4F	− 0 + 0 0 0	6F	+ + 0 − − +
10	0 − − + 0 +	30	+ − 0 0 − +	50	+ − − + 0 +	70	0 0 0 + + −
11	− 0 − 0 + +	31	0 + − − + 0	51	− + − 0 + +	71	0 0 0 + − +
12	− 0 − + 0 +	32	0 + − 0 − +	52	− + − + 0 +	72	0 0 0 − + +

（续）

数据字节	码　字	数据字节	码　字	数据字节	码　字	数据字节	码　字
13	−0−++0	33	0+−+0−	53	−+−++0	73	000+00
14	0−−++0	34	+−0+0−	54	+−−++0	74	000+0−
15	−−00++	35	−0+−+0	55	−−+0++	75	000+−0
16	−−0+0+	36	−0+0−+	56	−−++0+	76	000−0+
17	−−0++0	37	−0++0−	57	−−+++0	77	000−+0
18	−+0−+0	38	+−00+−	58	−−0+++	78	+++−−0
19	+−0−+0	39	0+−+−0	59	−0−+++	79	+++−0−
1A	−++−+0	3A	0+−0+−	5A	0−−+++	7A	+++0−−
1B	+00−+0	3B	0+−−0+	5B	0−−0++	7B	0++0−−
1C	+00+−0	3C	+−0−0+	5C	+−−0++	7C	−00−++
1D	−+++−0	3D	−0++−0	5D	−000++	7D	−00+00
1E	+−0+−0	3E	−0+0+−	5E	0+++−−	7E	+−−−++
1F	−+0+−0	3F	−0+−0+	5F	0++−00	7F	+−−+00
80	−00+−+	A0	−++0−0	C0	−+0+−+	E0	−++0−+
81	0−0−++	A1	+−+−00	C1	0−+−++	E1	+−+−+0
82	0−0+−+	A2	+−+0−0	C2	0−++−+	E2	+−+0−+
83	0−0++−	A3	+−+00−	C3	0−+++−	E3	+−++0−
84	−00++−	A4	−++00−	C4	−+0++−	E4	−+++0−
85	00−−++	A5	++−−00	C5	+0−−++	E5	++−−+0
86	00−+−+	A6	++−0−0	C6	+0−+−+	E6	++−0−+
87	00−++−	A7	++−00−	C7	+0−++−	E7	++−+0−
88	−000+0	A8	−++−+−	C8	−+00+0	E8	−++0+−
89	0−0+00	A9	+−++−−	C9	0−++00	E9	+−++−0
8A	0−00+0	AA	+−+−+−	CA	0−+0+0	EA	+−+0+−
8B	0−000+	AB	+−+−−+	CB	0−+00+	EB	+−+−0+
8C	−0000+	AC	−++−−+	CC	−+000+	EC	−++−0+
8D	00−+00	AD	++−+−−	CD	+0−+00	ED	++−+−0
8E	00−0+0	AE	++−−+−	CE	+0−0+0	EE	++−0+−
8F	00−00+	AF	++−−−+	CF	+0−00+	EF	++−−0+
90	+−−+−+	B0	+000−0	D0	+−0+−+	F0	+000−+
91	−+−−++	B1	0+0−00	D1	0+−−++	F1	0+0−+0
92	−+−+−+	B2	0+00−0	D2	0+−+−+	F2	0+00−+
93	−+−++−	B3	0+000−	D3	0+−++−	F3	0+0+0−
94	+−−++−	B4	+0000−	D4	+−0++−	F4	+00+0−
95	−−+−++	B5	00+−00	D5	−0+−++	F5	00+−+0
96	−−++−+	B6	00+0−0	D6	−0++−+	F6	00+0−+
97	−−+++−	B7	00+00−	D7	−0+++−	F7	00++0−
98	+−−0+0	B8	+00−+−	D8	+−00+0	F8	+000+−
99	−+−+00	B9	0+0+−−	D9	0+−+00	F9	0+0+−0
9A	−+−0+0	BA	0+0−+−	DA	0+−0+0	FA	0+00+−
9B	−+−00+	BB	0+0−−+	DB	0+−00+	FB	0+0−0+
9C	+−−00+	BC	+00−−+	DC	+−000+	FC	+00−0+
9D	−−++00	BD	00++−−	DD	−0++00	FD	00++−0
9E	−−+0+0	BE	00+−+−	DE	−0+0+0	FE	00+0+−
9F	−−+00+	BF	00+−−+	DF	−0+00+	FF	00+−0+

2. DC平衡

如刚才指出的，所有被选码字的组合电平权重为0或+1。例如码字+—+00的组合电平权重为
361 0而码字0+++—的组合电平权重为+1。显然，如果传输每个码字电平权重为+1的一串码字，那么

在接收方的平均信号电平会迅速偏离0电平，使得信号被错误解释。这称为**DC偏移**（DC wander），由线路每端的转换器引起。转换器的存在意味着没有用于直流电（DC）的线路。

为了克服这个问题，无论何时发送一串电平权重为+1的码字时，在传输前交替码字中的符号取反向电平。例如，如果由码字0+++−−组成的串要发送，实际发送的码字会是0+++−−，0−−−++，0+++−−，0−−−++等，由此获得电平权重为0的平均信号。在接收方，会应用相同的规则，交替码字在解码前会被再次取反回到原始形式。图7-5(b)中的状态迁移图说明了用于传输的规程。 362

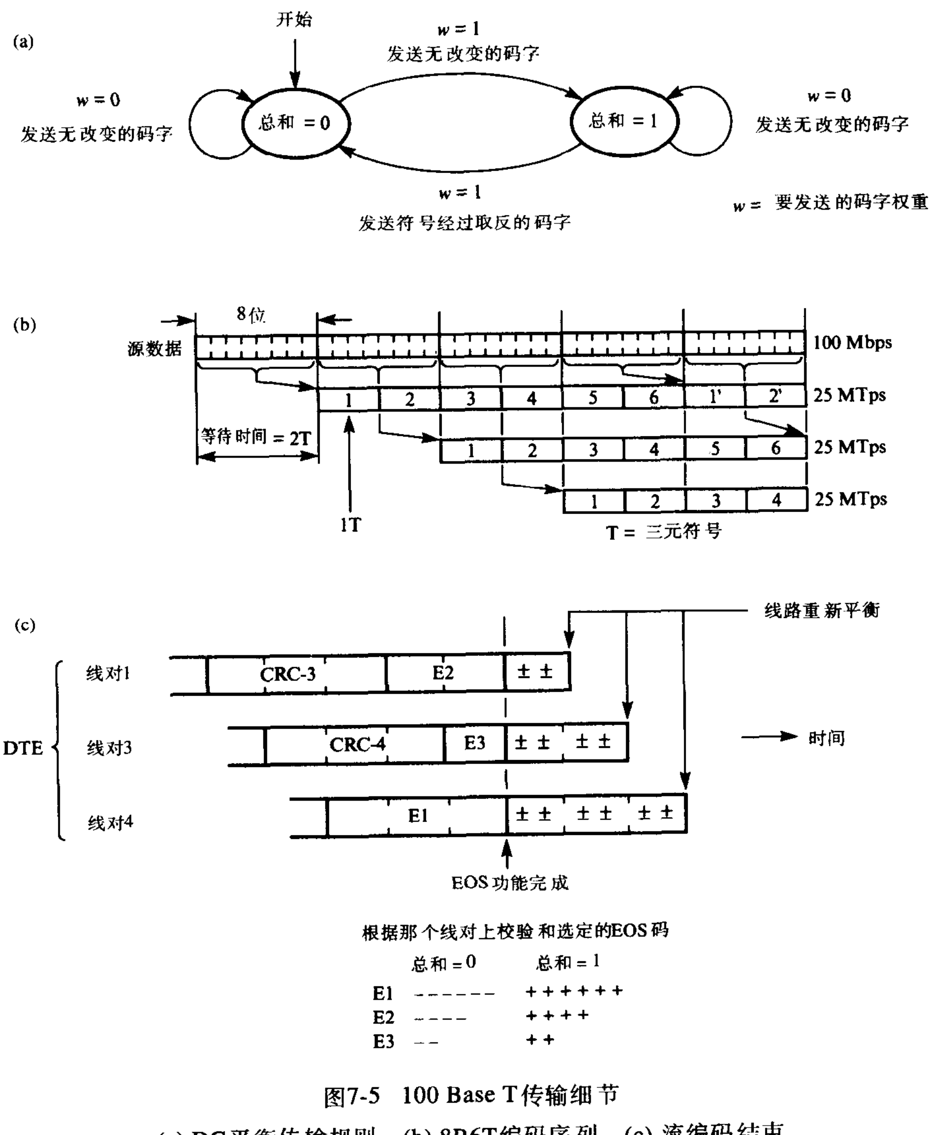


图7-5 100 Base T传输细节

(a) DC平衡传输规则 (b) 8B6T编码序列 (c) 流编码结束

在解码过程中为了减少等待时间，与每个解码字节对应的6个三元符号会以如图7-5（b）所示的序列在相应的三对线上传输。这意味着每对线上收到的符号序列会单独解码。还有，在收到最后一个符号后会立即处理这个帧。

3. 帧结束序列

363 传输规程采用基本CRC外加进一步的差错校验。从图7-5(a)的状态迁移图可以推出，电平权重的总和是0或+1。在每个帧发送结束时（就是说4个CRC字节被发送后），在三对线的每对线上发送两个不同的**流结束**（EOS）码中的其中一个。所选的码有效地形成了那对线的校验和。图7-5(c)说明了这个方案的原理。

在这个图中，假定4个CRC字节的最后一个（CRC-4）在线对3上。线对4下一个要发送的码字取决于那对线上的电平权重总和（称为校验和）是0或+1。在这个码字结束时就完成了EOS功能，另两个EOS码长度减少了两倍或一倍等待时间，就是4T或2T。这意味着接收方能可靠地检测到帧结束，因为所有的信号应该在另一个较短时间内停止。它考虑到了每对线上的传播时延的微小变化。

4. 冲突检测

图7-6(a)显示了无争用DTE集线器传输实例。回忆一下DTE发送时，在线对2上检测冲突
364 信号。类似地，集线器发送时，在线对1上检测冲突信号。然而，如图7-6(a)所示，DTE方发送的在1、3和4线对上传输的强（无衰减）信号会在冲突检测（2）线对上引入信号。这称为**近端串扰**（NEXT）。在限度内，会被DTE认为是从集线器接收到的（冲突）信号。同样的情况也出现在从集线器到DTE的反向传输中。

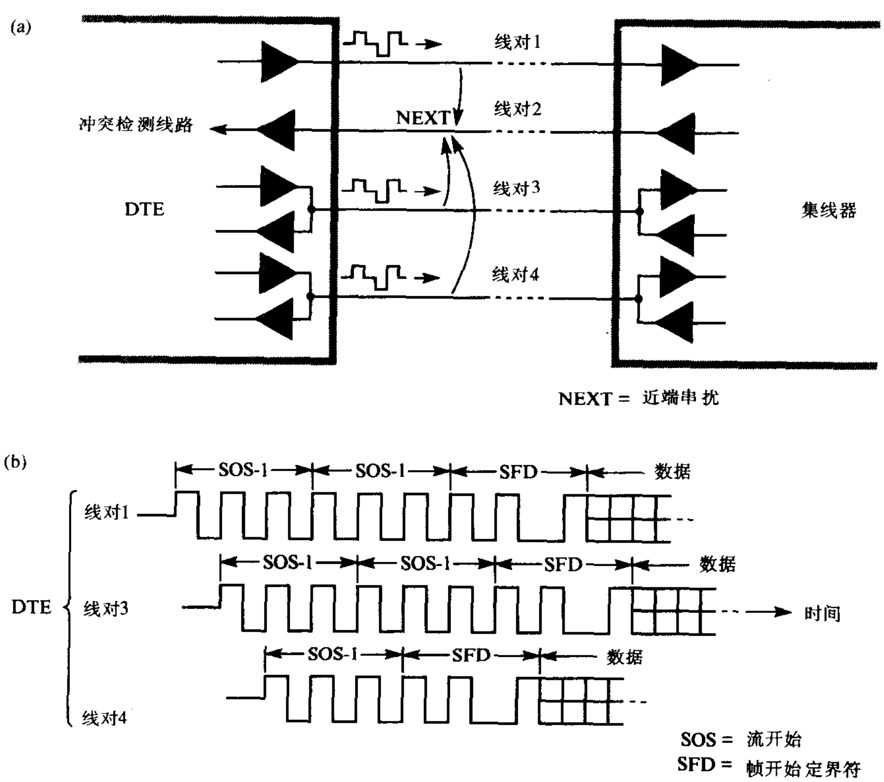


图7-6　帧开始细节

(a) NEXT的效应　(b) 前同步码序列

为了使NEXT最小化，每个帧开始的前同步码被编码成2电平（相对于3电平）符号串，就

是每个编码符号只有正负信号电平。这增加信号电平的振幅变化，反过来帮助DTE/集线器区分引入的NEXT信号和冲突帧的前同步码。

每对线上的前同步码称为**流开始**（SOS），由两个2电平码字组成（SOS-1和SFD）。三个线对传输的完整模式如图7-6(b)所示。可以看到，每线对上的SFD码字是交错出现的，在线对4上只发送单个SOS-1，这说明帧的第一个字节在线对4上传输，第二个字节在线对1上传输，第三个字节在线对3上传输，依此类推。一个帧开始能被接受需要检测到所有3个SFD码字，它们的交错形式意味着至少4个符号差错才会引起检测不到帧开始的差错。

在检测到冲突时，DTE发送阻塞序列然后停止发送。在这个时间点上，为了开始重新发送尝试，该DTE必须能够确定何时另一个涉及冲突的DTE停止发送。实际上，这相对较容易，因为在无传送（空闲）状态下使用8B6T编码，在三对数据线上存在一个0信号电平。这说明在冲突检测线对上没有产生的NEXT信号，反过来，能轻易确定集线器方是否发送阻塞序列。还有，为了提高电缆利用率，帧间间隔时间从9.6μs减少到960ns。

7.2.2　100 Base X

在本节开始时提到，100 Base X是为目前多数新应用中使用的高质量5类电缆设计的。另外，它还旨在用于屏蔽双绞线和光纤。可以使用不同类型传输介质是名称中“X”的由来。

每种传输介质需要不同的PMD子层。首先开发的是应用于FDDI网络的多模式光纤。正如在本章开始时提到的，FDDI局域网主要作为骨干子网，因为不像100 Base T，它能横跨达100公里的距离。FDDI网上的传输使用一种称为**4B5B**（有时写成4B/5B）的位编码方案。这种方案已被100 Base X采用。

使用4B5B，每组4个数据位被编码成5位（二进制）符号。符号是经过选择的，使得至少每2位就保证有一个信号跳变（它用来保持时钟同步）。此外，5位的使用（代表16组4位数据），意味着还有16个未使用的5位组合。它们用于各种链路控制功能。在100 Base X情况下，这些 365 符号中的两个（J和K）指示（MAC）帧的开始，其他两个（T和R）指示帧的结束。帧内容（包括前同步码和帧校验序列）使用5位符号进行编码（对应于组成帧的每个4位半字节）。这样J—K和T—R符号对是惟一的，并且对帧内容是透明的。

电缆由两根光纤组成，一根用于从DTE到集线器的传输，另一根用于从集线器到DTE的传输。像100 Base T一样，当DTE发送时，接收光纤上有（冲突）信号存在就检测到冲突。关于4B5B编码方案的更详细内容会在7.4节和5位符号集合一起讲。

7.3　IEEE 802.12

像100 Base T（快速以太网）标准一样，制定了IEEE 802.12作为IEEE 802.3 10 Base T标准的高速衍生标准。在7.2节中看到，设计100 Base T标准的主要目标是保留CSMA/CD协议。相比之下，IEEE 802.12标准只提供相同的MAC服务接口而使用完全不同的MAC协议。这样做的目的是：首先，使得可以不用网桥建立大型（分级）网络；其次，使得提交的（MAC）服务数据单元长度可变以便与所有现有LAN类型交互；再次，支持实时需求的附加无数据通信。

已经看到，绝大多数现有10 Base T布线使用4对语音级（VG）3类、非屏蔽双绞线的星型拓扑结构。其中设计系统的目的之一是使用安装好的已有基础而无需重新布线。IEEE 802.12标准又称为100（Base）VG-AnyLAN，这里“VG”表示语音级电缆，“AnyLAN”表示能配置与任何现有LAN类型交互。但是，除了3类电缆外，还可以使用4类或5类非屏蔽双绞线以及2对屏蔽双绞线或2对光纤。

7.3.1 拓扑结构

图7-7显示MAC协议设计成能在不同网络拓扑中工作。最简单的拓扑称为**一级网络**，如图7-7(a)所示。

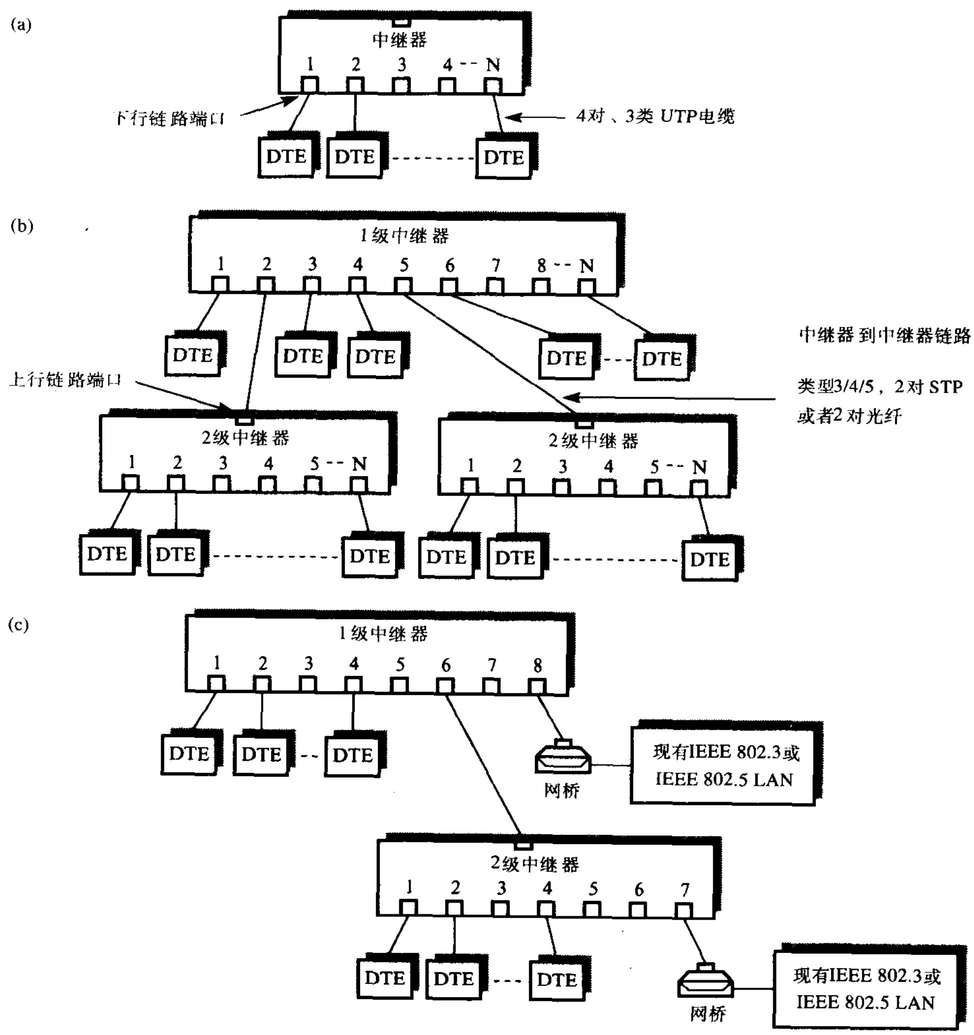


图7-7　IEEE 802.12网络拓扑

(a) 一级网络　(b) 多级网络　(c) 混合网络

可以看到，这是一个使用**中继器**（与IEEE 802.3中继器有不同的功能）的简单星型拓扑。在集线器上有许多终端站/DTE连到它的**下行链路端口集**。DTE可以是个人计算机、工作站、
366 服务器等。使用已有应用布线，在中继器与任何DTE间使用3类电缆的最大长度是100米，其他电缆类型会成比例增长。

为了建立有更广覆盖/更多端口个数的网络，可以把许多这类配置以多级结构形式进行分级，如图7-7(b)所示。由此产生的拓扑称为多级网络，其中最高级中继器有全部控制权。当每个较低级中继器是较大网络的一部分时，它有**上行链路端口**。反过来，在多级结构中，较低中继器连到上一级中继器的其中一个下行链路端口。以这种方式把中继器分级，使得在不使用网桥的情况下网络能覆盖2.5公里的距离。在7.5节中将会看到，网桥有效地用它的每一个端

口端接LAN。 367

还可能如图7-7(c)所示，与其他现有LAN类型交互工作。但是在这种情况下，必须使用网桥，而且它们必须在连接到IEEE 802.12网络的端口上支持新的MAC协议。为了促进这种方式的交互工作，IEEE 802.12帧格式可以是IEEE 802.3帧格式也可以是IEEE 802.5帧格式。但是对于任何特定网络只能使用一种帧格式。

7.3.2 MAC协议

MAC协议基于轮询原理。中继器除了中继附接DTE/中继器间的所有帧传输外，还控制所有传输序列。这用一组控制报文（信号）来完成，这些报文在中继器和每个附接DTE/中继器间交换。DTE在发送帧之前必须等待，直到它附接的中继器给它授权。当DTE有帧要发送，它首先在中继器链路上设置一个请求控制信号。这个信号还指明请求是正常优先级还是高优先级。通常正常优先级用于数据帧，而高优先级用于时延敏感信息。

在某一时刻，整个网络上只能有一个帧传输在进行，在每次传输完成后，中继器会轮询/检查所有端口来确定是否有未解决请求。传输顺序由**循环调度算法**控制，就是说所有传输以严格的数字（端口）次序进行。因为有两个不同优先级级别，中继器必须保留每个优先级级别允许下一个发送的端口（DTE）的记录。两个端口号分别称为**正常下一端口指针**（NNPP）和**高优先级下一端口指针**（HPNPP）。所有高优先级请求先被处理，然后才轮到正常优先级请求，两者都以相应优先级级别当前保存的下一端口指针中的端口号开始。这个调度算法称为**按需优先级调度**。

另外，为了确保等待正常优先级请求/帧在许多高优先级请求产生期间不被过度延时，为每个未解决正常优先级请求分配一个计时器。如果请求在超时时间间隔（200～300ms）内没有被处理，那么该请求就作为高优先级请求被处理。

图7-8显示了在一级网络中中继器中继帧的规程。在第一次轮询中，中继器决定有待处理请求的DTE能发送下一个请求，由此返回一个"清除"控制信号给这个等待DTE（这可以看作一种确认机制）。同时，通过向连到它端口的所有其他DTE发送"流入"控制信号通知它们准备接收可能到来的帧。作为响应，每个DTE返回一个"清除"控制信号，这反过来告知中继器已经准备好接收帧了。 368

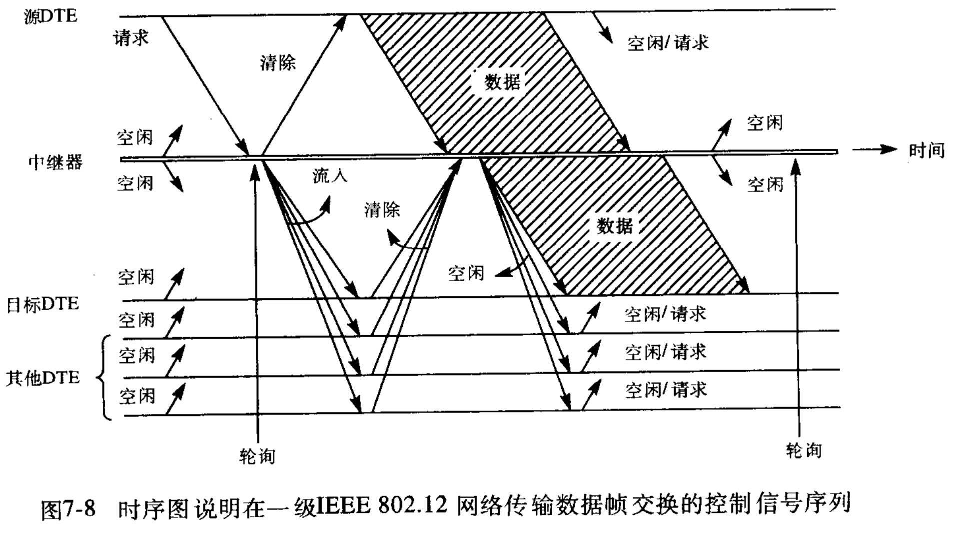


图7-8 时序图说明在一级IEEE 802.12 网络传输数据帧交换的控制信号序列

接到“清除”信号，选定的DTE开始发送它的等待帧。中继器开始接收该帧并把它临时存储在FIFO缓冲器中（又称为**弹性缓冲存储器**）直到它接收到位于帧头部的目标地址字段。然后它使用这个地址确定所需输出端口（从它维护的存储查询表中），当接收该帧的其余部分时，它通过FIFO缓冲器中继完整帧内容到所需的目标端口/DTE，并发送“空闲”信号给其他所有DTE。当检测到帧结束时，中继器执行另一个轮询序列。可以推断出，不像CSMA/CD，只有标明地址的DTE才会接收到帧的副本，因此提高了系统的固有安全性。为了使中继器知道连到每个端口的DTE的MAC地址，当DTE第一次通电工作时，需要发送一个含有其MAC地址的短报文。

实例7-1

为了说明按需优先级控制算法的操作，图7-9显示了一个请求序列和相关传输的例子。这个序列假定网络初始状态为空闲状态，两个下一端口指示器（NPP）都设为1。以所示的时间顺序，每个DTE发送请求，正常请求表示为N，而高优先级请求为H。下面解释传输顺序是如何产生的。

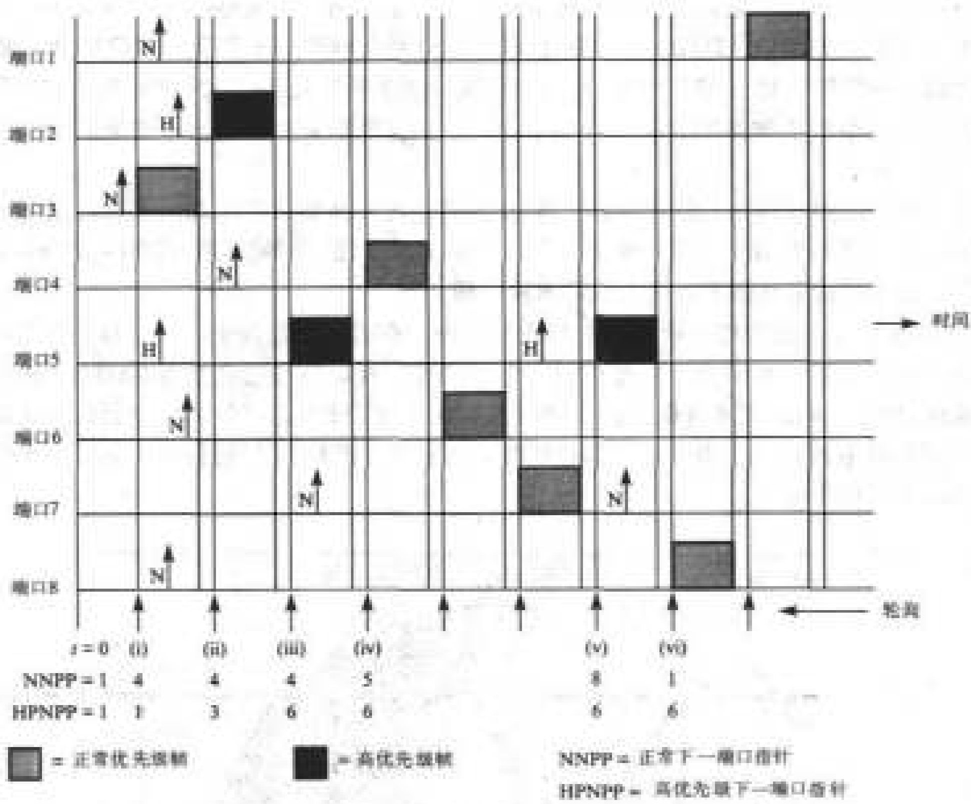


图7-9　一级网络的帧传输序列实例

解：

（i）在第一个轮询周期，检测到端口3上有正常请求，因此这个正常帧首先被传输，并且NNPP的值提高到4。

（ii）在传输这个正常帧期间，设置了许多正常和高优先级请求，它们在第一个传输结束后的下一个轮询周期被中继器检测到。如图7-9所示，虽然端口5的高优先级请求比端口2上的请

求先设置，但是中继器先选择端口2，因为它更接近当前的HPNPP值（1）。然后后者增加到3。 369

（iii）在端口2的帧传输结束后，端口5的高优先级帧被发送，HPNPP增加到6。

（iv）在下一个轮询周期，中继器在1、4、6、7和8端口检测到未解决的正常请求。因为当前NNPP值是4，因此中继器选择端口4为下一次传输的端口，接着端口6和7为下两次传输的端口。NNPP值现在设成8。

（v）在端口7传输帧期间，端口5产生了一个新的高优先级请求。因为在完成端口7帧转发后，中继器从它的轮询中检测到这个请求，并在端口5启动这个高优先级帧的传输，HPNPP值仍然是6。

（vi）在这个传输过程中，端口7、4和3收到新的正常优先级请求。因此下一个轮询周期在端口1、3、4、7和8检测到未解决的正常优先级请求。因为NPP现在是8，端口8的帧先被传输，下一次是端口1，依此类推。

（vii）注意为了清楚起见，每个请求只显示一次，但实际上在每个帧传输过程中请求会反复出现直到该请求被处理掉。 370

在分级网络中，最高级中继器（也称为**根中继器**）控制整个网络中的所有帧传输。为了解释轮询规程，考虑7-10(a)所示的二级网络的情况。

每个较低级中继器以正常方式轮询它的端口，如果存在请求，它就在上行链路端口设置一个正常请求或高优先级请求（这由当前请求决定）。每个较低级中继器都执行一遍这个规程，由此在分级结构中较低级中继器端口的请求会在对应的根中继器端口上产生一个或多个请求。

每个较高级中继器存储了连在它下行链路端口上的设备类型（DTE或中继器）的记录。根中继器开始轮询序列，如果要处理的下一端口请求来自DTE，就以一级网络中类似的方法处理。首先，附接在根中继器上的所有中继器（以及连在其上的DTE）和其他DTE会被告知帧有可能到来。然后根中继器收到帧时，获得帧头部的目标地址，如果这个帧要发送给连在根中继器上的本地DTE，就直接中继给这个特定DTE。另外，根中继器会中继这个帧的副本给所有连有中继器的端口。多级结构中的所有其他中继器都执行这个规程，使它们能检测到每个帧传输的结束。在这个时刻，它们都开始一个新的轮询序列。

另一种情况，如果根中继器要处理的下一端口请求来自连有中继器的端口，就以正常方式把请求确认传给这个中继器，然后较低级中继器在其端口整个循环周期持有选择权。所以逻辑上网络工作好像是一级网络一样，图7-10的实例网络中可见端口号排序。类似地，任何发生在较低级中继器上的传输被中继到所有中继器端口，因此也就通过整个网络传播。 371

为了确保网络中任何端口（DTE）的高优先级请求比正常优先级请求先处理，需要额外的机制。如果根中继器在较低级中继器拥有控制权并传输正常优先级帧期间，检测到它的端口有高优先级请求；一旦检测到帧传输结束，根中继器就通过向其所有连有中继器的端口发送“预空”控制信号挂起较低级中继器在它当前循环周期的进一步传输。这使得当前拥有控制权的较低级中继器挂起对进一步请求的处理，并把控制权返还给根中继器。然后根中继器在把控制权返还给较低级中继器使得它完成被挂起的循环周期前，以正常方式传输高优先级帧。

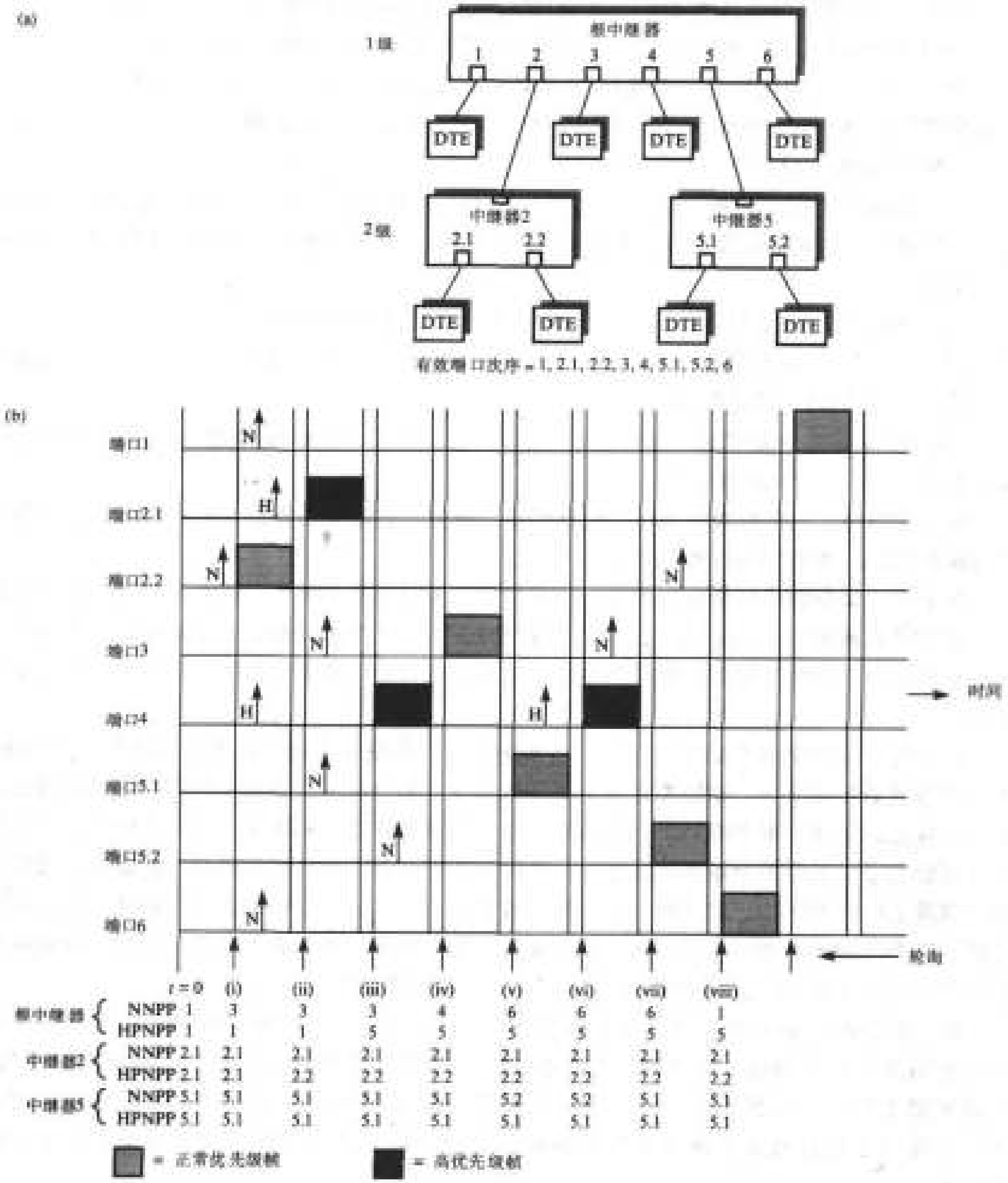


图7-10　二级IEEE 802.12网络实例

(a) 有效端口次序　(b) 帧传输序列实例

实例7-2

为了说明多级网络中按需优先级算法的操作，图7-10(b)给出了图7-10(a)所示二级网络的

372 请求序列及相关传输的实例。该序列假定网络初始为空闲状态，所有的NPP为1。下面解释传输序列是如何产生的。

解：

（i）端口2.2上的正常优先级请求使得中继器2在连到根中继器端口2的上行链路产生一个正常优先级请求。在下一次轮询中，根中继器检测到该请求并通过端口2返回给中继器2一个确认并把NNPP提高到3。中继器2接到确认会在它的端口发起一次轮询，然后检测到未处理正常优先级请求在端口2.2。接着连到这个端口的DTE就开始传输等待帧。中继器2把这个帧中继到根中继器端口2的上行链路并从帧头部读取目标地址以确定这个帧是否发送给连在它的其他下行端口的DTE。根中继器接到这个帧就把它中继到中继器5，这样两者都能检测到帧传输结束。

（ii）中继器2检测到帧传输结束，继续两个级别的循环轮询，检测到现在在端口2.1有高优先级请求。所以它发起下个传输并更新HPNPP。在它的端口没有其他等待请求时，中继器2把控制权返还给根中继器。

（iii）在这段时间，其他中继器端口已产生了许多其他请求，这使得在根中继器端口1、3、5和6出现正常优先级请求，而在端口4出现高优先级请求。后者的传输先发起并且HPNPP值增加到5。

（iv）在这个传输中，又产生一个正常优先级请求，但是因为根中继器端口5已经有未处理正常优先级请求，所以这个请求没有任何结果。根中继器NNPP现在是3，因此先发起连到端口3的DTE的传输而且NNPP增加到4。

（v）在下一个轮询中，根中继器检测到端口5上的请求，因此发确认给连到该端口的中继器并把NNPP增加到6。中继器5接到确认发起对它的端口的循环轮询序列。结果发起端口5.1上的帧传输，并把它的NNPP增加到5.2。

（vi）在这段时间，在端口4检测到一个高优先级请求，由此根中继器打断中继器5并在把控制权返还给中继器5前发起这个传输过程。

（vii）接到确认，中继器5继续它的循环并在把控制权返还给根中继器前发起端口5.2上的帧传输。

（viii）下一个轮询中根中继器检测到端口1、2、3和6上有等待正常优先级请求。因为它的NNPP现在是6，因此先发起这个端口的帧传输并把NNPP更新成1。继续下面的规程。 373

7.3.3 物理层

如7.3节提到的，MAC协议不使用冲突检测，因此所有四对线都可以用来传输MAC数据帧，每对线以25Mbps工作。实际上，通过特定的编码方案可以把传输速率提高到30Mbps，但仍然在定义的限度内。图7-11显示了组成物理层的主要单元的示意图。

使用的编码方案称为**5B6B**，因为源数据中的每个5位组（称为**五位字节**）被转换成一个6位符号（称为**六位字节**），每个符号有相同数目的二进制1和0。它确保能保持时钟/位同步。另外，为了有助平衡传输的二进制数据流中的1和0的数量（除去DC偏移量），在执行编码操作前，4个源数据流都分别进行扰乱，就是说以伪随机方式随机将数据扰乱（扰乱与密码混合），这样接收方才能根据规则还原为有效信息（称为反扰乱）。

因为流入数据流在四对线上并行传输，每个五位字节的传输顺序需要选择，这样就能轻易地被接收方重新组合成源数据流。图7-11(b)给出了使用的五位字节传输顺序（扰乱处理除外）。MAC帧的内容先被分段成五位字节。在编码后，它们以循环的方式在四对线上传输。可以看到，在对这些五位字节进行编码（实际上它们会被扰乱）后，由此产生的每个六位字节有相同数量的1和0，因此除去了DC偏移量。当通过转换器传输时，为了防止DC偏移这是必需的。

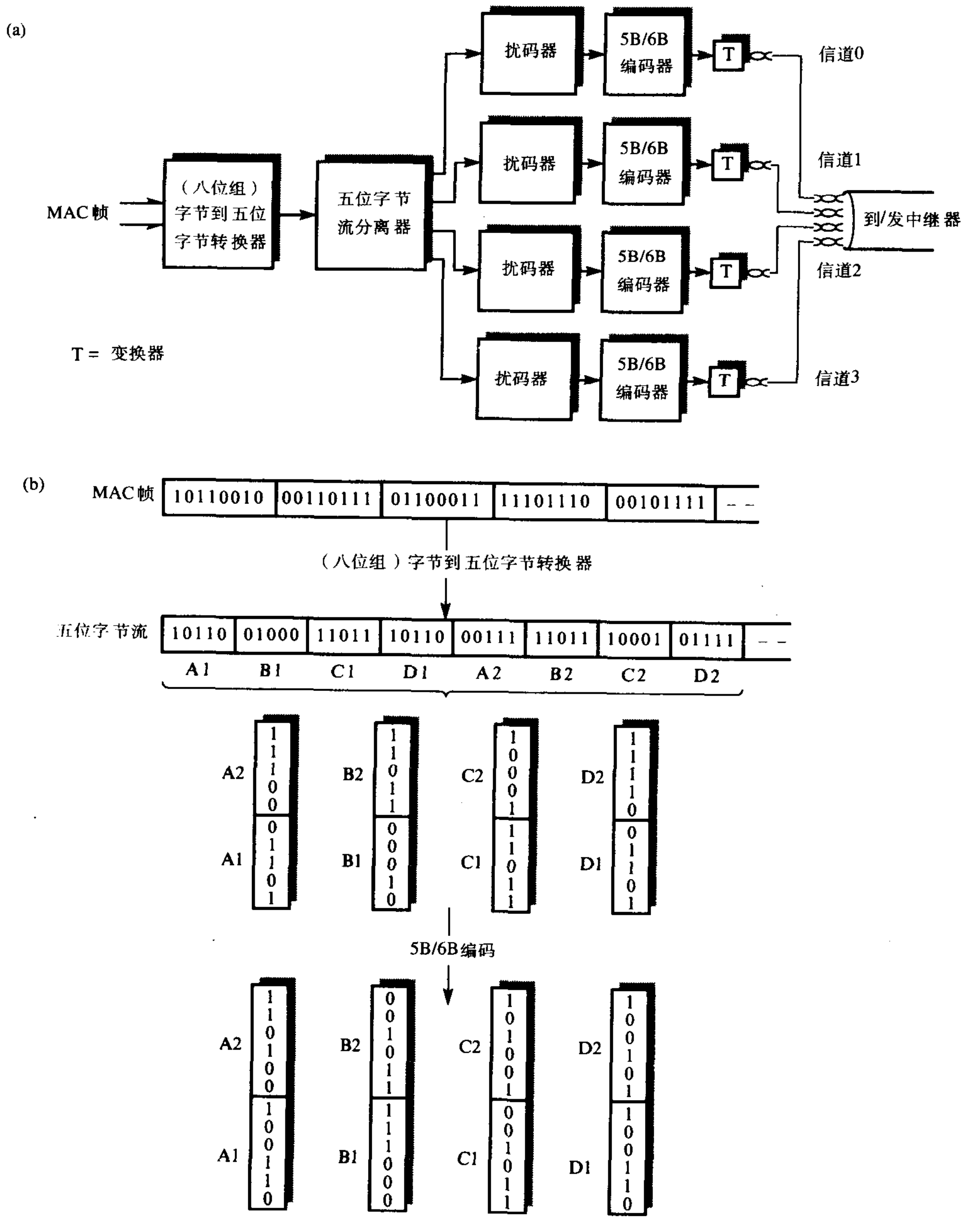


图7-11　IEEE 802.12物理层原理

(a) 主要单元　(b) 编码例子

与MAC协议相关的各种控制信号形成于两个固定频率音调的组合（一个低而另一个高）。因为它们在任何数据传输前交换，所以它们像数据一样在相同的四对线上传输。但是，对于控制信号，两对线用于一个方向（比如从DTE到中继器），另两对线用于反方向。每对音调线的含义取决于这对线是传输还是接收，其中一种选择如图7-8所示。

7.3.4　性能

如果假定一级网络只有正常优先级请求，那么DTE在中继器端口产生发送帧的请求和这个帧被目标DTE接收之间的最大时延，等于从一个DTE传输一个最大长度帧到另一个DTE的

时间乘上中继器的现用端口数量。使用IEEE 802.3（以太）帧和一个32端口中继器，在标准中这个时延是3.84ms的数量级。这当然是最坏情况的时延。如果在一个负载很小的网络中使用不同长度的帧，实际的时延明显比它小。同样，所有DTE确保平等地共享可用带宽。 374

虽然这个时延满足了大多数实时要求，但是对附接有许多DTE的大型网络来说产生的时延会变长，而对于其中一些应用来说，时延会变得令人无法接受。在这些应用中，必须采用双优先级协议。使用这个协议，高优先级帧的最坏情况时延可以用单优先级网络中正常优先级帧一样的方式量化。另外，因为低优先级帧的存在，必须引入一种机制，它负责分配可用高优先级带宽的使用。 375

为了设置网络中任何DTE经历的最长时延的边界值，定义了**目标传输时间**（TTT）。首先确定在这个时间内，特定的网络配置所能获得的吞吐量，然后在那些需要传输时间关键通信的DTE中划分这个吞吐量。接着使用高优先级请求传输这个通信，并且协议确保这个吞吐量被所有传输这些通信的DTE均分。当然在限度内，对于整个网络来说这些通信的比特率总和不能超过100Mbps。对于有许多工作站（传输时间关键通信）的大型网络来说，这是个限制因素。第10章会描述能满足这种需求的另一种网络体系结构。

7.4 FDDI

FDDI LAN标准是由美国国家标准协会（ANSI）制定的。现在它成为了国际标准，定义在ISO 9314中。这个标准基于环型拓扑并以100Mbps的数据传输率工作。如令牌环，它使用双计数器轮转环来提高可靠性。多模式光纤把每个站连在一起，整个环最长能达100公里。最多能有500个站（DTE）连到环中，由此形成了一个理想的骨干网。MAC方式基于控制令牌，除了正常数据通信外，环还能可选地支持需保证最小访问时延的时延敏感通信的传输，例如数字化语音。现在来更详细地描述这个标准。

7.4.1 网络配置

FDDI使用双计数器轮转环来提高可靠性：**主环和次环**。可以从图6-18中看到，次环可用作附加传输路径，也可在主环发生断裂时单纯地充当备用环。一个典型网络配置如图7-12所示。

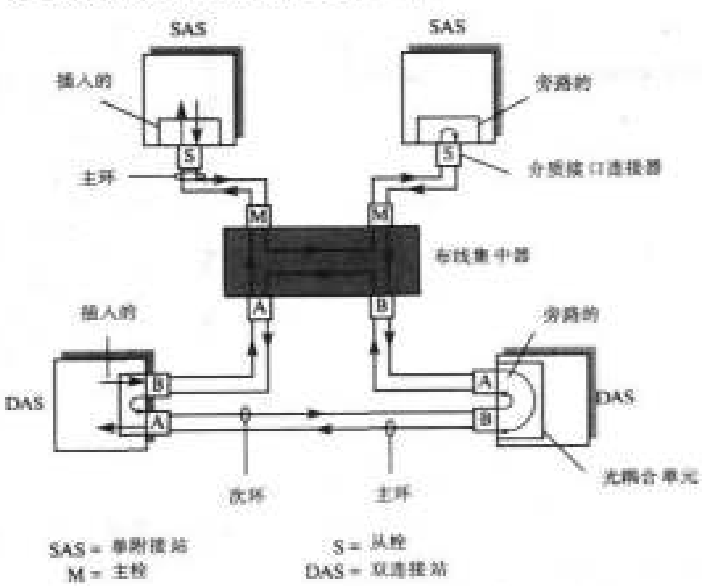
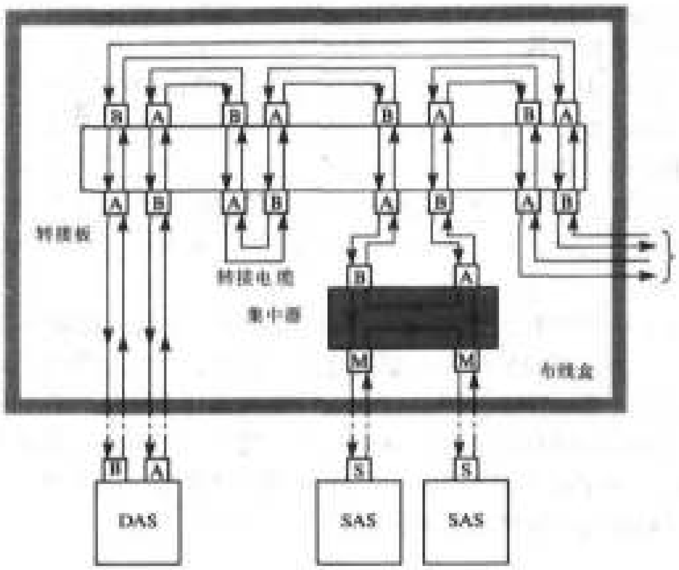


图7-12　FDDI连网组件

(a)



(b)

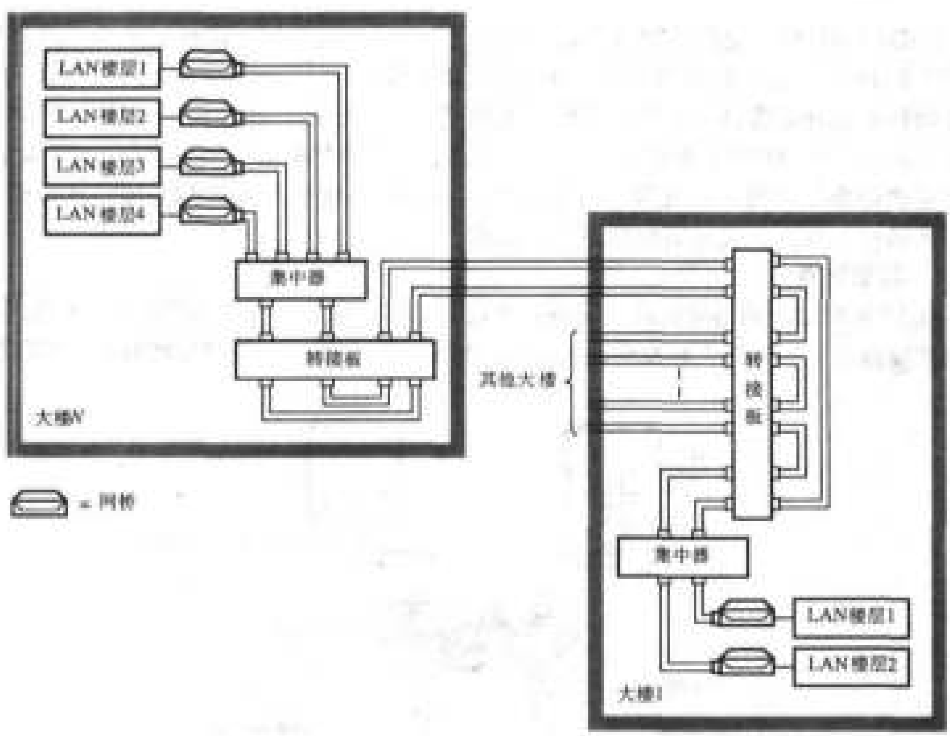


图7-13 FDDI布线示意图

(a) 大楼 (b) 地区

可以看到有两种站（DTE）：**双连接站**（DAS）（它连接在两个环上）以及**单连接站**（SAS）（它只连接在主环上）。实际上，大多数用户站通过**布线集中器**附接到环上，因为只需要一对光纤，这样连接成本会较低。

如果LAN用作骨干网，那么大多数附接站是网桥。当发生环故障时，用来**重新配置**LAN

成为一个单环的协议已在6.2.2节中描述过。 376

基本的光纤电缆是两端有**极性双工**（双位置）连接器的**双芯**电缆。这意味着电缆每端有不同的物理栓，这样它只能连到匹配的插座上。避免了因发送光纤和接收光纤无意中被互换而引起的整个网络瘫痪。进一步的防范措施是使用不同的连接器连接两种站（SAS和DAS）。和基本令牌环相同，当某个站掉电停止工作时，使用特殊耦合单元隔离（旁路）这个站。在FDDI中有有源或无源光纤设备。

虽然拓扑结构逻辑上是个环，但是物理上它通常使用集线器/树型结构的形式实现。如图7-13(a)所示。为了确保线路变化在可控制范围内，使用了**转接板**和布线集中器。它们一般位于与楼层（如果使用的是本地FDDI环）或大楼（如果使用的是骨干环）相关的布线盒（布线室）中。在后者情况下，转接板如图7-13(b)所示互连建立树型结构，根位于计算机中心。使用分支电缆连接每个站（工作站或网桥）到环上。

每个转接板有若干可能的连接点。当某个特定点没有连接时，环使用**转接电缆**维持，每条转接电缆有相同类型的连接器。增加一个新的站或集中器只是简单地除去一条转接电缆并把它替换成相应的分支电缆。这种方式称为结构化布线（见6.1.2节）。 377

7.4.2 物理接口

光纤的物理接口如图7-14所示。在基本令牌环网络中，任意时刻只有一个活动的环监控站，它为环提供主时钟。这个活动的环监控站使用差分曼彻斯特编码对每个循环二进制数据流进行编码。然后环中所有其他DTE（站）的频率和相位和二进制数据流中提取的时钟锁定。但是这种方法不适合FDDI环的数据传输率，因为这需要200兆波特的速率。相反，每个环接口都有自己的本地时钟。流出数据使用这个时钟发送，而流入数据使用流入二进制数据流的频率和相位锁定跳变的时钟接收。可以看到所有数据在传输前被编码，这样在二进制数据流中至少每两个位信元周期就保证有一次跳变，它确保采样的每个接收位十分接近位信元中心。

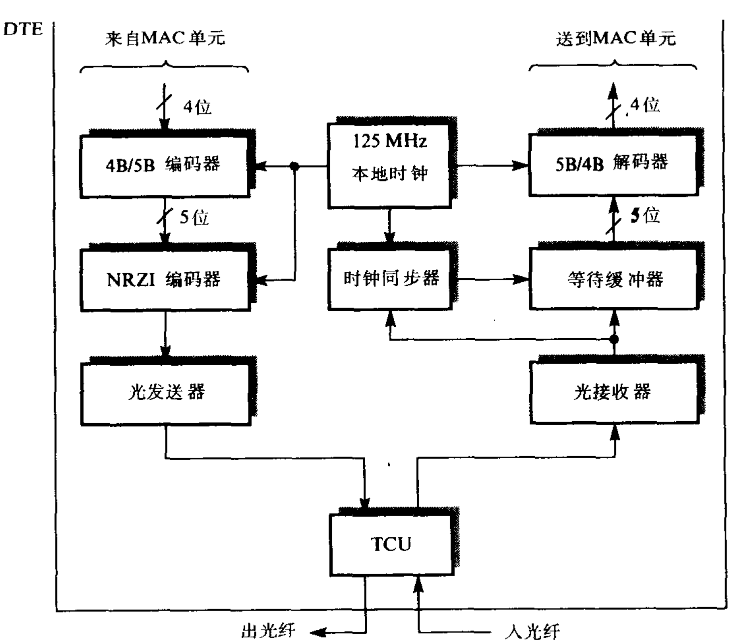


图7-14 FDDI物理接口示意图

所有要发送的数据在传输前先使用**4/5组码**（4 of 5 group code）进行编码。对于每个要发

送的4位数据，**4B5B编码器**会产生一个相应的5位码字或5位符号。16种可能的4位数据组各对应一个5位符号，如图7-15(a)所示。可以看到，在每个符号中连续0的最大个数是2。然后，5
378~379 位符号还要通过NRZI编码器移出，当发送1时它会产生一个信号跳变，而当发送0时它不产生信号跳变。这样至少每两位保证有一个信号跳变。

(a)

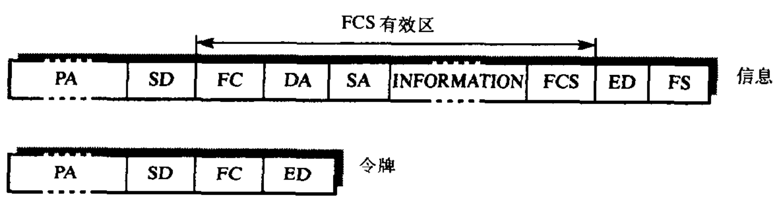
数据符号

4位数据组	5位符号
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

控制符号

IDLE	11111
J	11000
K	10001
T	01101
R	00111
S	11001
QUIET	00000
HALT	00100

(b)



PA = 前同步码（16个或更多个符号）
SD = 起始定界符（2个符号）
FC = 帧控制（2个符号）
DA = 目标地址（4个或12个符号）
SA = 源地址（4个或12个符号）
FCS = 帧校验序列（8个符号）
ED = 结束定界符（1个或2个符号）
FS = 帧状态（3个符号）

图7-15　FDDI线编码及建帧细节

(a) 4B5B 码　(b) 帧格式

用5位来代表16个4位数据组，意味着还有16个未使用的5位组合。在这些组合（符号）中，有一些用作其他（链路）控制功能，例如指示每个传输帧或令牌的开始和结束。图7-15(a)列出了这些链路控制符号，而图7-15(b)给出了帧和令牌的格式。总体上，每个字段的用途和含义跟基本令牌环中的一样，但是由于使用了符号而不是位，因此每个字段结构有些不同。

前同步码（PA）字段由16个或更多IDLE符号组成，因为IDLE符号由5个二进制1组成，因此使得线路信号以最高频率变化。线路信号跳变用来建立（和保持）接收方的时钟同步。起始定界符（SD）字段由两个控制符号（J和K）组成，它使得接收方能在正确的符号边界解释其后的帧内容。FC、DA和SA字段与以前的含义相同，但是在FDDI中数据帧中的（经解码后的）信息字段可以达到4500个字节。结束定界符（ED）字段含有一个或两个控制符号（T）。最后，帧状态（FS）字段虽然与基本环中的FS字段有相似的功能，但它由三个符号组成，这
380 些符号是两个控制符号R和S的组合。

物理接口使用的本地时钟为125MHz，因为4B5B编码法可获得100Mbps的数据传输速率。因为所有传输都编码成5位符号，因此每个5位符号在接收方解码前首先需要缓冲。但是，使用J和K两个符号让SD字段建立正确符号边界，意味着接收方需要使用10位缓冲器。因为它给

环引入了10位的时延（等待时间），所以它称为**等待缓冲器**（或弹性缓冲器）。在125Mbps情况下，它等同于0.08μs的时延，但考虑到额外的门和寄存器传输时延，通常在每个环接口它接近1μs。

实例7-3

假定光纤中的信号传播时延是每1km 为5μs，如果在100Mbps传输速率下，推出在下列FDDI环配置下的等待时间和等待位。

（a）2km 环，有20个站

（b）20km 环，有200个站

（c）100km 环，有500个站

解：

环等待时间T_1 = 信号传播时延$T_p + N \times$ 站等待时间T_s，其中N是站数量。

（a）$T_1 = 2 \times 5 + 20 \times 1 = 30$μs或者3000位

（b）$T_1 = 20 \times 5 + 200 \times 1 = 300$μs或者30 000位

（c）$T_1 = 100 \times 5 + 500 \times 1 = 1000$μs或者100 000位

注意得到上述值的前提是只使用主环。如果发生故障，三个信号传播时延值都会加倍。同样对于双连接站，站等待时间也会加倍。

7.4.3　帧传输和帧接收

基本令牌环的较短等待时间（以及由此循环的位数）就是指某个站（DTE）在发起一个信息帧的传输后，就等待，在环使用中无任何显著损失情况下，直到收到这个帧尾部的FS字段（发送一个新令牌前）之间的时间间隔。但是从实例7-3可推断出，在FDDI环中如果采用这种操作模式，环使用中的损失会变得很显著。因此在FDDI环中，采用早期令牌释放方式，就 381
是说，在该站发送帧尾部FS符号后立即发送一个新令牌。然后该站把IDLE符号跟在令牌后，直到接收到指示新帧开始或新令牌开始的SD符号。基本的原理如图7-16所示。

可以看到，如在基本令牌环中一样，源站在帧绕环一周后把它从环中除去。但是因为FDDI环有较长的等待时间，可能在某时刻会有多于一个帧在绕环传递。虽然图7-16没有显示，但环接口接到任何帧，在确定SA字段中是否是自身地址前必须转发SD、FC和DA字段（符号）。这会导致一个或多个由SD、FC和DA字段组成的帧段绕环传递。这就是说一个站接到令牌后，开始发送等待帧，同时接收并丢弃绕环传递的任何帧段。

7.4.4　计时令牌循环协议

不像基本令牌环使用的传输控制方式，基于对令牌帧和信息帧中访问控制字段的优先级位和保留位的使用，FDDI环使用与令牌总线网相同的传输控制方式原理。回忆一下它是基于称为目标令牌循环时间（TTRT）的预设参数进行控制的计时令牌循环协议。

令牌的每次循环，每个站会计算自上次收到令牌后经过的时间。这称为令牌循环时间（TRT），包括在令牌的这次循环中该站发送等待帧的时间加上环中其他所有站发送等待帧的时间。显然，如果环负载较低，那么TRT较短，但是当环负载增加时，每个站计算的TRT也随之增长。这样TRT就是整个环负载的一种测量方式。

计时令牌循环协议通过允许一个站只在它计算的TRT小于环中预设的TTRT时才发送等待帧，确保所有站公平地共享对环的访问。因此，当一个站有帧等待发送，接到令牌时它先计算TTRT与当前TRT的差值。这称为令牌持有计时器（THT），因为它决定在释放令牌前该站有多长时间能够发送等待帧。如果THT为正，那么该站能在这个时间间隔内发送帧。如果它

为负，那么该站必须在这次令牌循环中放弃发送等待帧。正THT称为**早期令牌**（early token）而负THT称为**晚期令牌**（late token）。6.2.3节给出的例子说明了TTRT怎样控制混合优先级通信对令牌总线网络的访问。但是在FDDI网络中，TTRT用来控制拥有相同优先级的数据帧对
382 环的访问。

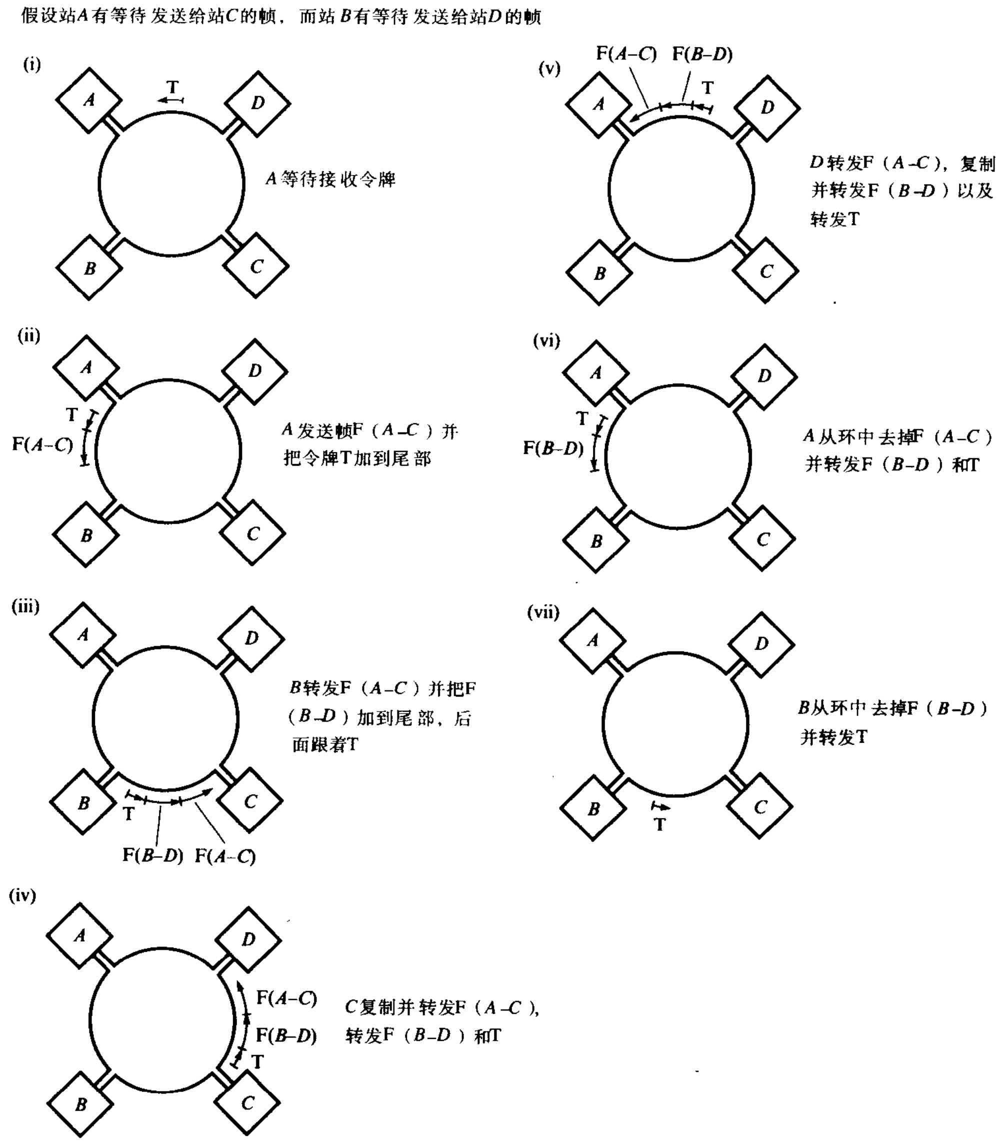


图7-16　FDDI传输实例

实例7-4

计时令牌循环协议用来控制对有4个站的FDDI环网络的访问。所有传输的帧一样长，并且TTRT等于4个帧传输时间加上环等待时间。在没有帧准备发送的空闲时段之后，所有4个站
383 都有帧要发送。假定令牌在空闲状态绕环一周的时间等于传输令牌的时间T_t加上环等待时间T_1，表中显示了每个站在下4个令牌循环中能传送的帧数量。

解：

图7-17中的表说明了在最初的4个令牌循环中每个站能发送的帧个数。它们如下产生。

令牌循环	站1		站2		站3		站4	
	TRT	XMIT	TRT	XMIT	TRT	XMIT	TRT	XMIT
0	T_t+T_l	0	T_t+T_l	0	T_t+T_l	0	T_t+T_l	0
1	T_t+T_l	4	$4+T_t+T_l$	0	$4+T_t+T_l$	0	$4+T_t+T_l$	0
2	$4+T_t+T_l$	0	T_t+T_l	4	$4+T_t+T_l$	0	$4+T_t+T_l$	0
3	$4+T_t+T_l$	0	$4+T_t+T_l$	0	T_t+T_l	4	$4+T_t+T_l$	0
4	$4+T_t+T_l$	0	$4+T_t+T_l$	0	$4+T_t+T_l$	0	T_t+T_l	4

TRT = 令牌循环时间

XMIT = 在令牌的这次循环中发送的帧个数

TTRT = 目标令牌循环时间

T_t = 发送令牌的时间

T_l = 环等待时间

TTRT = $4+T_t+T_l$

图7-17　FDDI计时令牌循环协议实例

在空闲时段后，因为没有帧在此之前被传送，所以所有站的TRT都是T_t+T_1。一旦有帧准备发送，站1接到令牌计算THT为4，因此在传递令牌前传送（XMIT）4个等待帧。但是，因为站1已经发送了4个帧，其他站就不能在这个令牌循环中发送帧了。这是因为每一个TRT都大于4，因此计算出的相应THT为负。

在令牌的第二个循环中，站1的TRT已经增加到4加上环等待时间，因此它不能发送任何等待帧。这意味着站2的TRT小于TTRT，因此它能发送4个等待帧。这又阻止了站3和站4在这个令牌循环中发送等待帧。

在令牌的第三个循环中，站1和站2的THT都是4加上环等待时间，因此它们都不能发送帧
了。但是站3的TRT这次小于TTRT，因此它能发送4个等待帧。站4再次被阻止发送帧。 384

最后在令牌的第四个循环中，站1、2和3都被阻止发送帧而站4能发送4个帧。这个简单例子说明所有4个站平等地共享可用传输容量。

7.4.5　性能

关于共享介质网络有两个重要性能指标：**最大可获得吞吐量**和**最大访问时延**。两者受到使连接站共享可用传输容量的MAC算法的较大影响。刚才看到，在FDDI环情况下，它基于控制令牌和计时令牌循环协议的使用。关于这个协议的重要参数是TTRT，因为它必须在所有站中预设成相同值，量化它对可获得吞吐量和访问时延两者的影响显得很重要。

虽然FDDI环的名义吞吐量为100Mbps，但因为环等待时间和访问控制机制，最大可获得吞吐量小于这个值。最大可获得吞吐量（和访问时延）意味着在每个环接口上收到令牌时总是有帧等待发送。在这种情况下，可以通过考虑实例7-4的操作流程来计算最大可获得吞吐量。

在这个实例中，接收到令牌，第一个站发送一组帧，直到传递令牌前TTRT过期为止。在这次令牌循环中，环中其他所有站都被阻止发送帧，直到下一次令牌循环时令牌传递给环中的下一个站才发生第二次帧传输。

因此令牌的连续两次循环中的传输时间损失由两部分组成：第一个站绕环传输一组帧的时间（环等待时间）以及令牌传到环中下一个站的时间。可以表示名义环容量的最大利用率$U_{\max}$如下：

$$U_{\max}=\frac{\mathrm{TTRT}-T_1}{(\mathrm{TTRT}-T_1)+T_1+T_t+(T_1/n)}=\frac{n(\mathrm{TTRT}-T_1)}{n(\mathrm{TTRT}-T_1)+(n+1)T_1+T_t}$$

这里TTRT指目标令牌循环时间，T_1指环等待时间，T_t指传输令牌时间，而n指环中站数量。实际上TTRT比T_t大得多，因此最大利用率近似为：

385

$$U_{\max}=\frac{n(\mathrm{TTRT}-T_1)}{n\mathrm{TTRT}+T_1}$$

可以推断，为了获得高级别的环利用，必须选择一个远大于环总等待时间的TTRT。另外，所选的TTRT必须允许环中至少存在一个最大长度帧。最大长度为4500字节的帧需要以100Mbps的速率传输0.36ms。如果允许使用次环并且所有500个站都是双连接站的话，由实例7-3得出的最大环等待时间大约为2ms。因此允许的最小TTRT是2.36ms，考虑安全边际，它在标准中设为4ms。

访问时延定义成从帧到达源站环接口到被目标站环接口交付之间的时间延迟。这样的访问时延包括在源站环接口队列中等待直到可用令牌（早期令牌）到达的时间。它只有在提供的负载小于环最大可获得吞吐量时才有意义，否则接口队列会持续地建立，并且访问时延会变得越来越大。

假定提供的负载小于最大可获得吞吐量，由实例7-4可推断出最大（最坏情况）访问时延。假定所有站在它们的环接口队列同时接收到一组帧，这样它们在每次令牌循环中能使用全部TTRT定量。假定站1是第一个能发送帧的站，那么环中最后一个站（就是站4）会经历最大访问时延，因为它只有在第四个令牌循环开始时才能获得可用令牌。同样，如果环接口队列中的4个帧都要发送给站3，那么当它们都沿环发送给该站时会经历一个等于环等待时间的额外时延。用来计算FDDI环最大访问时延的通用表达式是：

$$\begin{aligned}A_{\max}&=(n-1)(\mathrm{TTRT}-T_1)+nT_1+T_1\\&=(n-1)\mathrm{TTRT}+2T_1\end{aligned}$$

这里$A_{\max}$指接收可用令牌的最坏情况时间，其他含义都与以前一样。

显然，因为对于特定环配置来说环等待时间是固定的，所以环TTRT越大，最大访问时延也就越大。如实例7-5所示，对几乎最大的网络来说，FDDI环的最大可获得吞吐量可以用等于最小值4ms的 TTRT得出。因此虽然在标准中最大TTRT可以达到165ms，但是一般环中工作的TTRT明显小于该值。

实例7-5

得出下列三种环配置下的最大可获得吞吐量和最大访问时延。假定每种配置已选择4ms

386 的TTRT。

（a）2km环，有20个站

（b）20km环，有200个站

（c）100km环，有500个站

解：

这三个环的配置与例7-3中使用的环配置相同，因此用相同的计算得到环等待时间。

最大可获得吞吐量：$U_{\max}=\dfrac{n(\mathrm{TTRT}-T_1)}{n(\mathrm{TTRT})+T_1}$

最大访问时延：$A_{\max}=(n-1)\mathrm{TTRT}+2T_1$

（a）从例7-3得到：$T_1=0.03\mathrm{ms}$。所以：

$$U_{\max}=\frac{20(4-0.03)}{20\times4+0.03}=0.99$$

并且

$$A_{\max}=19\times4+0.06=76.06\mathrm{ms}$$

（b）从例7-3得到：$T_1=0.3\mathrm{ms}$。所以：

$$U_{\max}=\frac{200(4-0.3)}{200\times4+0.3}=0.92$$

并且

$$A_{\max}=199\times4+0.6=796.06\mathrm{ms}$$

（c）从例7-3得到：$T_1=1\mathrm{ms}$。所以：

$$U_{\max}=\frac{500(4-1)}{500\times4+1}=0.75$$

并且

$$A_{\max}=499\times4+2=1.998\mathrm{s}$$

7.4.6　同步数据

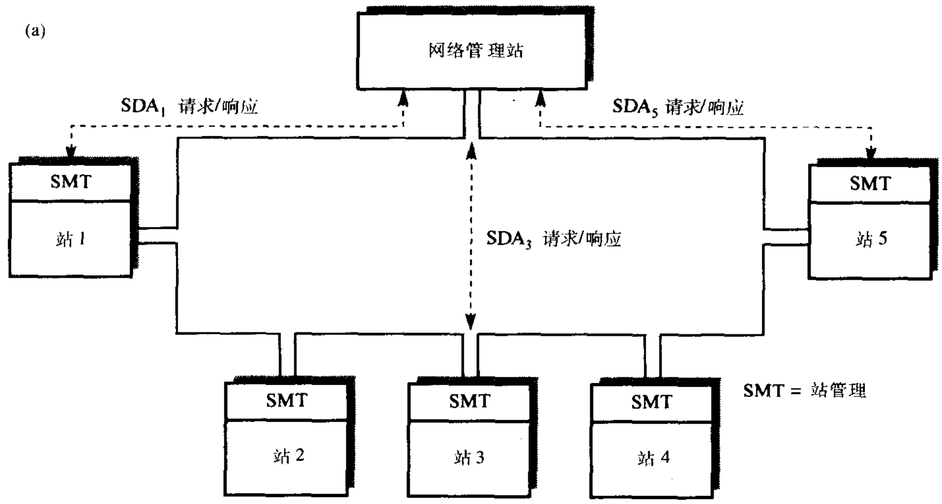
计时令牌循环协议确保所有站能公平地访问共享环传输介质来传输同优先级的数据帧。在这里，相同优先级说明数据帧由纯文本工作站产生，一般涉及文件传输、电子邮件和其他相似事务。因为这些帧以随机时间间隔产生，称它们为**异步数据**帧。

除了异步数据帧外，标准还包括针对**同步数据**（数据是时延敏感的，由此必须在一个保证的最大时间间隔内传输）传输的可选服务。例如一组帧，每个包含连续语音采样块。第2章中已经提到，数字化语音采样以恒定时间间隔（125μs）产生，因此这类帧基本上在保证最大时间间隔内被传输，以避免到达接收方的连续语音采样的时间发生变化。 387

为了满足这些同步数据的需求，分配给那些支持这类通信的站一个固定部分的环带宽，每次收到令牌时使用它。它称为**同步分配时间**（SAT），并定义了站每次收到令牌能发送同步数据的最大时间间隔。同步数据不是由计时令牌循环协议控制的，相反，给单个站分配的用于同步数据传输的环带宽由一个独立的环管理站控制。一般方案如图7-18(a)所示。

所有同步带宽请求表示成环TTRT的比例。这些请求发送给网络管理站，假定有空闲可用的同步带宽，分配给每个站它们所请求的带宽量。不是所有站都需要提供同步数据服务，并且不是所有同步带宽分摊都一样。为了实现这个方案，每个站必须能与网络管理站通信。每个站含有自己的环管理（称为**站管理**（SMT））代理，它们使用定义好的协议以正常方式通过环与环管理站通信。一般方案如图7-18(b)所示。

(a)



站1、3、5支持异步和同步数据

站2和4只支持异步数据

SDA_n = 站n的同步数据分配时间

同样 $\sum_{i=1}^{n} SDA_i$ = 环目标令牌循环时间，TTRT

(b)

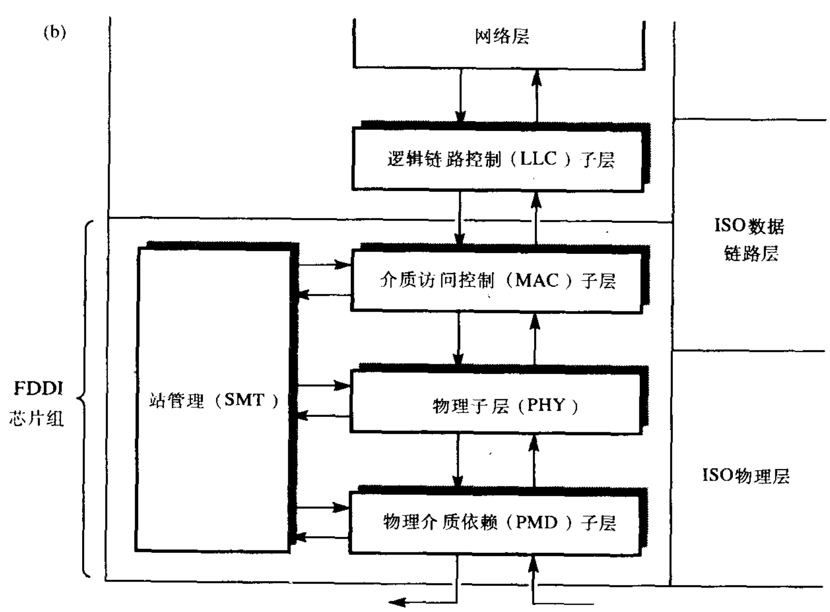


图7-18 FDDI同步数据协调

(a) 示意图 (b) 站协议体系结构

除了用于协调同步数据分配，相同的SMT配置还用来协调环TTRT。回忆一下环TTRT有

最小值4ms和最大值165ms。实际值由环管理站决定。如果某个站需要高于4ms（默认值）的值，那么它能向环管理站申请将环TTRT设成更高的值。如果同意，那么更高的值就下载到每个站（使用同样的环管理协议）。会在第13章中进一步讨论SMT协议。

为了确保令牌在保证的最大时间间隔内被提供同步数据传输服务的每个站收到，网络管理站拥有一份授予每个站同步分配时间的记录。它们的总和不允许超过各站用来控制异步数据帧传输的协定TTRT。回忆7.4.5节，TTRT定义成当传输异步数据时令牌沿环循环的最大时间间隔。由此，因为分配给同步数据的总带宽等于TTRT，所以在一个站收到令牌前期望的最大时间间隔会是两倍TTRT。

当一个站同时支持同步和异步数据传输时，这个站一获得令牌就先在同步数据分配时限内发送等待的同步数据。在这段时间内，它的TRT被挂起。然后，假定该站有异步数据等待发送，它以正常的方式从TTRT中减去当前TRT。得到的结果（正或负）决定在令牌的这个循环中是否传送等待的异步数据。 388~389

注意前述规程定义了令牌在被某个站接收前可能经历的最大时间间隔。如果环上的异步数据通信量比较少，那么令牌大约会在一半的这个时间间隔被接收。而且，如果同步数据在令牌循环期间实际传输量小于分配的最大时间间隔，那么最大时间间隔仍然会根据这个传输量进一步减少。这说明，连续令牌循环的实际时间间隔会在可能较小的值和规定的最大值之间变动。对于时间敏感（同步）数据，这不存在问题，因为对其的主要要求是最坏情况下的最大时间间隔。对于以固定时间间隔产生（以及由此必须被传递和输出）的数据，就会出现问题。这种数据称为**同步数据**，包括数字化的语音和视频。这种数据的一个特点是不仅对时延而且对时延偏差或抖动敏感。刚才描述的固有时延偏差限制了FDDI用于这种通信。所以开发了一种称为FDDI-II的FDDI的派生版本。将在第10章中讨论混合媒体（也称为多媒体）网络时讨论它。将会看到，FDDI-II同时支持异步和同步数据。

还开发了FDDI的第二个派生版本，称为**铜带分布式数据接口**（CDDI）。它以类似于FDDI的方式操作，惟一不同的是（正如名称中提到的）它使用铜线作为传输介质。互连各个站的最大物理间距大大缩短。它主要用于跨越单一办公室或大楼的较短距离的局域网。

7.5 网桥

在描述网桥的操作前，先复习一下中继器的基本操作。中继器用来确保每个DTE（站）内的驱动电子设备发送的电子信号能在整个网络中传输。在ISO参考模型环境中，它完全工作在物理层，如图7-19(a)所示。

对于任何类型的LAN网段来说，网段的物理长度和可以连接在它上面的（终端）站个数都有一个规定好的最大限制。当互连网段时，中继器限制了与网段物理接口相关的输出电路的电子驱动要求。这样，传输路径上多个网段的存在（由此有多个中继器）对于源站是透明的。中继器简单地重新生成一个网段收到的所有信号然后把它们发送（转发）到下一网段上。

可以推断出，中继器没有智能装置（比如微处理器），因此如果仅仅用中继器互连网段，那么来自连到某网段的任何站的帧传输会在整个网络中传播。这意味着，在带宽方面，网络工作起来如同单个网段。随着每个LAN网段的需求（传输带宽方面）增加，整个LAN负载也随之增加，并伴随着整个网络响应时间的相应恶化。 390

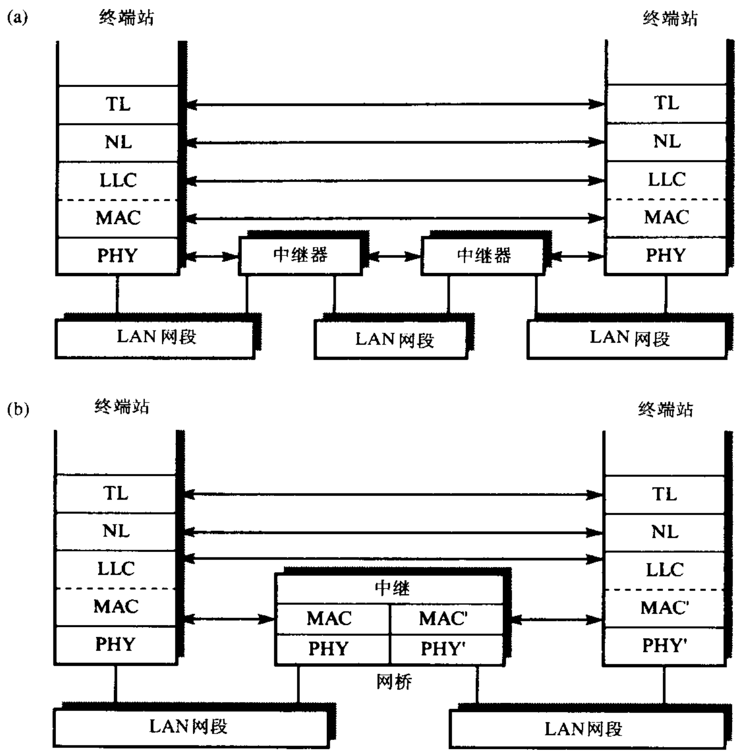


图7-19 LAN互连示意图
(a) 中继器 (b) 网桥

网桥功能

由于网桥也用来互连LAN网段，因此功能类似于中继器。但是当使用网桥时，收到的来自某网段的所有帧会被缓冲（存储）并在转发前进行差错校验。而且只有当这个帧无差错并且指向与发送该帧的源站所处网段不同的站时，才会被转发。因此，连到同一LAN网段的各站间的所有传输不被转发，由此不会负载网络其余部分。网桥就这样工作在ISO参考模型环境中的MAC子层。如图7-19(b)所示，由此产生的LAN称为**桥接LAN**。

缓存帧相对于中继器既有优点也有缺点。总的来讲，优点多于缺点，但都列出来可能会
391 有所帮助。

网桥的优点如下：

- 没有了任何有关互连功能的物理限制，意味着组成LAN的网段数和连接站总数可以轻易地增加。这对于建立分布在较广地理区域内的大型LAN，尤其重要。
- 在把从一网段接收到的帧转发到另一网段前进行缓存，意味着两个互连网段可以以不同的MAC协议工作。这样就更容易建立由不同类型的基本LAN混合组成的LAN。
- 网桥仅执行基于帧中MAC子地址的中继功能，它的效果是网桥对于协议栈中更高层使用的协议是透明的。这意味着网桥可用于支持不同协议栈的LAN。
- 网桥允许通过LAN自身轻易并有效地管理大型网络。例如，通过把与管理相关的软件加到网桥设计中，与LAN网段相关的性能数据能轻易地放入日志并供以后查阅。同样，还可以加入访问控制机制来提高网络安全，以及通过动态地控制单个网桥端口的状态来改变LAN的工作配置。

- 把LAN分成更小的网段提高了网络的整体可靠性、实用性和适用性。

网桥的缺点如下：

- 因为网桥在执行中继（转发）功能前，完整地接收和缓存所有帧，这就引入了相对于中继器额外的存储转发时延。
- 在MAC子层没有针对流量控制的防备措施，由此网桥可能在高通信量期间过载，就是说网桥需要存储比空闲缓冲区容量更多的帧（在把它们转发到每条链路之前）。
- 用网桥连接的网段以不同的MAC协议工作，意味着由于帧格式不同帧内容在转发前必须修改。每个网桥必须产生新的帧校验序列，影响是当帧通过网桥中继时引入的任何差错不会被检测到。

已经指出，由于网桥的优点远大于缺点，现在得到了广泛的应用。应用最广泛的两类网桥是**透明**（也称为生成树）**网桥**和**源路由选择网桥**，它们之间的主要差别是路由选择算法。

如果使用透明网桥，网桥自身作出所有路由选择决定；而使用源路由选择网桥，则由终端站执行主要的路由查找功能。现在有了透明网桥的国际标准IEEE 802.1（D），而源路由选择网桥也形成了一部分用于互连令牌环网段的IEEE 802.5（令牌环）标准。 392

将分别讨论这两种网桥。但是注意这些讨论只打算作为技术概述，而且标准仍在发展，因此如果需要更多的细节，可以直接参考相关标准。

7.6 透明网桥

使用透明网桥，像中继器一样，在两个正在通信的站之间的线路上存在的一个（或多个）网桥对于这两个站来说是透明的。所有路由选择决定由网桥作出。而且透明网桥在开始服务后，会以动态方式自动初始化和配置自身（路由选择信息方面）。网桥示意图如图7-20(a)所示，而一个简单桥接LAN如图7-20(b)所示。

LAN网段通过**网桥端口**连接到网桥上。基本网桥只有两个端口，而**多端口网桥**有若干个端口（由此可连接多个网段）。实际上，每个网桥端口由与特定LAN类型网段（CSMA/CD、令牌环和令牌总线）相关的MAC集成电路芯片集以及一些相关端口管理软件组成。这些软件负责在启动时初始化芯片集（芯片集全都是可编程的）以及缓冲区管理。通常，可用存储空间逻辑上被分成若干个称为缓冲区的固定单元。缓冲区管理涉及把一个空闲缓冲区（指针）传给准备接收帧的芯片集以及把帧缓冲区（指针）传给向前发送（转发）的芯片集。

每个网桥以**混杂方式**工作，就是说网桥接收和缓冲在每个端口接收到的所有帧。当帧在某个端口被接收时，就被MAC芯片集放入指定的缓冲区，端口管理软件准备一个用于新帧的芯片集，然后把含有接收帧的存储缓冲区的指针传给**网桥协议实体**进行处理。因为两个（或多个）帧可能同时到达端口，并且两个或多个帧需要从同一输出端口转发，端口管理软件和网桥协议实体软件间的存储空间指针的传递通过一组队列来进行。

我们会在7.6.1节看到，每个端口可以有许多可能的状态，接收帧的处理根据规定的协议进行。网桥协议实体软件的功能是用来实现使用的特定网桥协议。

1. 帧转发（过滤）

网桥含有一个**转发数据库**（也称为**路由选择目录**）。对于每个端口，它指定转发该端口接收到的帧到流出端口（如果有）。如果在某端口接收到的帧指向接收该帧的网段（端口）上的某个站，该帧就被丢弃。否则它通过转发数据库指定的端口转发。正常路由选择决定涉及一个简单的查询操作：先读取每个接收帧中的目标地址，然后根据这个地址从转发数据库中读 393

取相应的端口号。如果与接收帧端口一致，该帧就被丢弃。否则它在（从数据库）访问到的端口排队，等待转发给相应网段。这个处理过程也称为**帧过滤**。

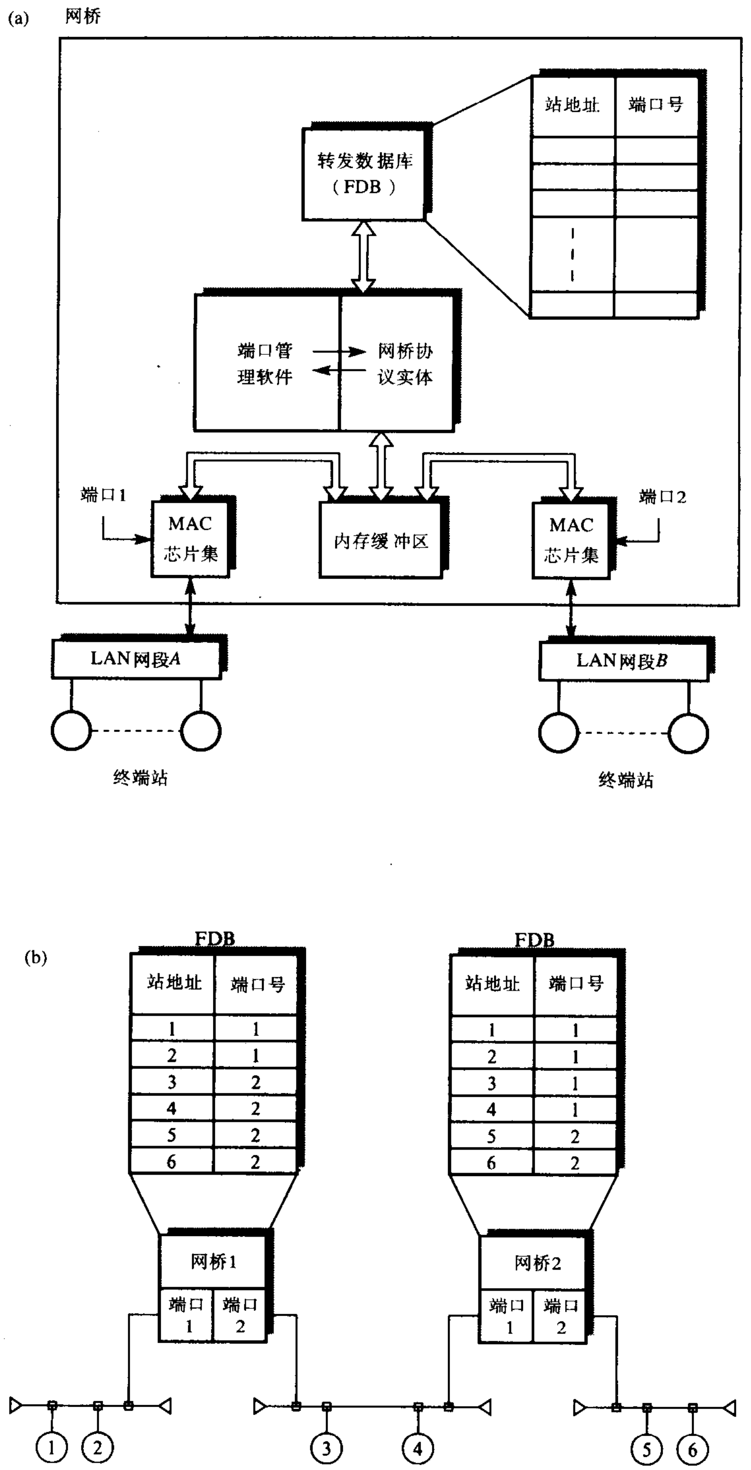


图7-20　网桥示意图

(a) 体系结构　(b) 应用实例

2. 网桥认知

透明网桥的主要问题是转发数据库的生成。一种方法是预先生成转发数据库中的内容并放在固定的存储空间，例如可编程只读内存（PROM）。它的缺点是，当网络拓扑改变时（比如新增一个网段或者用户改变站的连接点（网段）)，所有网桥中的转发数据库内容都要相应地改变。为了解决这个问题，大多数桥接LAN中的转发数据库内容不是静态建立的，而是在网桥工作期间动态生成和维护的。它通过一个认知过程以及与其他网桥对话确认整个LAN的拓扑来共同完成。

当网桥开始服务时，它的转发数据库初始为空。无论何时接收到帧，读取帧内的源地址以及接收该帧的流入端口号并把它们存入转发数据库。另外，因为此时转发端口不可知，该帧的副本被转发到网桥的其他所有输出端口。当这些帧通过网络传播时，每个网桥重复这个规程。首先，流入端口号被输入转发数据库（针对源（站）地址），帧的副本被转发到网桥上所有其他输出端口。这个动作一般称为**扩散**（flooding），因为它确保每个传输帧的副本被整个LAN的所有网段接收到。在认知阶段，对网桥接收到的每个帧重复这个规程。这样，LAN中的所有网桥迅速建立它们的转发数据库内容。

只要各站不被允许在网络中移动（改变它们的连接点）并且整个LAN拓扑是简单的树型结构（就是说在任何两个网段间没有重复路径），那么这个规程会令人满意地工作。这种树型结构称为**生成树**。然而，因为在许多网络，尤其是大型网络中，这些可能性都存在，所以基本认知操作进一步改进如下。

与站相关的MAC地址在它制造时就固定下来了。如果用户改变工作站网络的连接点，每个网桥的转发数据库内容必须定期更新以反映这些变化。为了实现它，有一个**无活动定时器**（inactivity timer）与数据库中每条记录相关。当从某个站接收到帧时，该记录相应的定时器就被重置。如果在预定义的时间间隔内没有接收到来自该站的帧，定时器会超时并且该记录会被删除。当接收到来自已删除记录的站的帧时，又执行认知规程，用端口号（可能是新的）更新每个网桥中的记录。这样网桥中的转发数据库就能持续更新，来反映当前LAN拓扑和当前连接在互连网段上的站地址。无活动定时器还限制了数据库的大小，因为它只含有当前活动的那些站。因为数据库大小影响了转发操作的速度，所以这显得相当重要。 394~395

只有整个桥接LAN是一个简单（生成）树拓扑，认知处理才能起作用。这意味着网络中任何两个网段间只有单一路径。但是，因为可能使用额外网桥来连接两个网段，例如为了提高可靠性或当LAN更新时发生误操作，所以不可能总是满足这个条件。

使用已概述的基本认知算法，两个网段间不可能存在多条路径，因为认知阶段中的扩散操作会使得网桥持续重写转发数据库记录。可以通过考虑图7-21所示的简单LAN拓扑来领会这一点。显然，如果在认知阶段网段1上的站10发送帧，那么网桥B1和B2都会在转发数据库产生一条记录并把该帧的副本转发到网段2。每个副本反过来会被另一个网桥接收到，（在端口2）产生一条记录并在端口1输出这个帧的副本。反过来，每个副本被另一个网桥接收到， 396
由此端口1的相应记录被更新。因此这个帧会持续地循环，每个端口的记录被持续地更新。

因此，对于提供站间多条路径的拓扑，需要额外算法为任何两个网段间转发帧选择单一网桥。由此产生的**逻辑拓扑**或**活动拓扑**（active topology）像单一生成树一样工作，这个算法称为**生成树算法**。注意虽然该算法只选择单一网桥来连接两个网段（使得任何引入的用来提高（例如）可靠性的可选网桥变得多余），但是它以规定时间间隔运行并且会动态地从当前工作网桥中选出一组网桥。

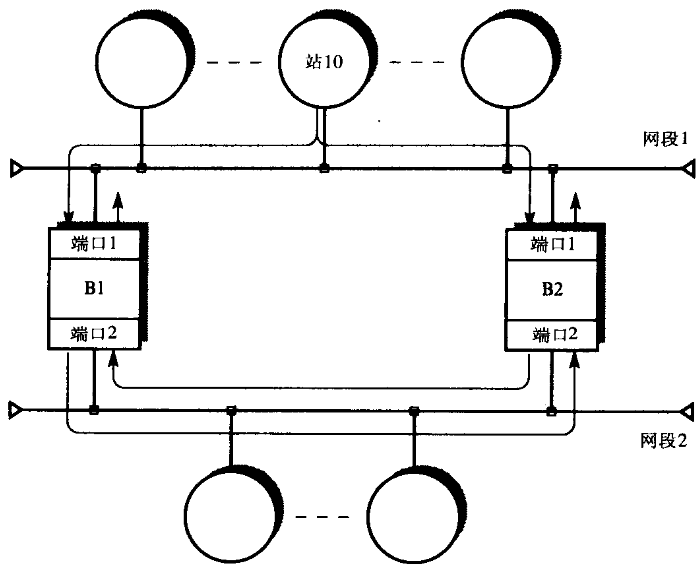


图7-21 双路径对于认知算法的影响

7.6.1 生成树算法

使用生成树算法，所有网桥定期交换称为**网桥协议数据单元**（BPDU）的特殊帧（报文）。每个网桥有一个优先级值和惟一的标识符。对于整个桥接LAN来说，通过生成树算法动态选择一个网桥成为**根网桥**，它有最高的优先级和最小的标识符。它在规定的时间间隔内被确定/证实。

在根网桥确定后，从根网桥到所有其他网桥端口的路径成本也被确定。称为**根路径成本**，是帧从根网桥发送到考虑中端口所经过的最小成本路径。然后每个网桥确定它的哪一个端口有到根网桥的最小路径成本，它称为**根端口**，因为随后它会在这个端口接收所有来自根网桥的BPDU。

与端口相关的路径成本通过它连接网段的传输比特率（称为**指派成本**）来确定。传输比特率越高，标志成本越低。如果从根网桥有两条可选路径，一条由2个10Mbps CSMA/CD网段组成，另一条由2个2Mbps网段组成，那么两个更高比特率网段的路径会有更低的路径成本。当同一网桥的两个端口的路径成本相同时，端口标识符作为判断标准。

一旦确定了每个网桥的根端口，选择某个网桥（端口）来转发帧到每个网段。它称为**指派网桥**。它的选择基于从考虑中的网段到根网桥的最小路径成本。如果连到某网段的网桥端口有相同的路径成本，就选择更小标识符的网桥。连接该网段到它的指派网桥的端口称为**指派端口**。对于根网桥来说，它总是所有连接到它的网段的指派网桥。

当建立用于某一网段的指派网桥端口时，注意一旦某网桥端口被选为根端口，它不会参
397 加仲裁规程成为指派端口。这样指派端口在连接考虑中网段的非根端口中选择。在涉及的两个（或多个）网桥间交换配置BPDU，允许这两个（或多个）网桥在选择指派端口时能共同作出决定。

在建立根网桥以及所有其他网桥的根端口和指派端口后，网桥端口状态被设成**转发**或**阻塞**。初始时，因为根网桥的所有端口是指派端口，它们被设成转发状态。对于所有其他网桥，只有根端口和指派端口设成转发状态而其他设成阻塞状态。这就建立了等同于生成树的活动拓扑。

1. 拓扑初始化

LAN中的所有网桥都有惟一的MAC群地址，用来发送网桥间的所有BPDU。网桥收到的BPDU不是直接被转发，网桥协议实体使用它们含有的信息生成BPDU然后转发到其他端口。

当网桥刚开始服务时，它假定自己是根网桥。相信它是根网桥的网桥（最初）隔固定时间间隔发起到所有端口（以及连到它的网段）的称为**呼叫时间**的**配置BPDU**的传输。

每个配置BPDU含有若干字段，包括：

- 发送BPDU的网桥认为是根网桥的那个网桥的标识符（最初是根网桥自己）；
- 从接收BPDU的网桥端口到根网桥的路径成本（最初是0）；
- 发送BPDU的网桥的标识符；
- 接收传送过来BPDU的网桥端口标识符。

收到配置BPDU，连接发送它的网段的每个网桥通过比较BPDU含有的根网桥标识符与自身标识符，来确定是否该网桥有更高的优先级，或者如果优先级相同，是否自身标识符小于接收帧中的标识符。如果小于，它会假定自己就是根网桥并简单地丢弃接收帧。

另一种情况，如果接收到的BPDU中的根网桥标识符指出它不是根网桥，那么该网桥把与接收BPDU的端口相关的路径成本加到帧中已有的路径成本上。通过早先发给它的网络管理信息，网桥知道连到它端口上的网段的指派成本。它产生含有这个信息以及自身标识符（网桥和端口）的新配置BPDU，并转发它的副本给其他所有端口。LAN中所有网桥重复这个规程。这样配置BPDU会从根网桥向网络末端扩散。

当配置BPDU从根网桥扩散时，计算其他所有网桥的每个端口的相关路径成本。这样，除 398
了建立的单一根网桥，其他网桥都会确定自己每个端口的相关路径成本。这样它们就可以选择自己的根端口，因为在分级结构的任何点上，两个（或多个）连到同一网段的网桥会交换配置BPDU，所以它们能从收到的BPDU中的合计根路径成本确定那个网段的指派网桥。当出现路径成本相同时，网桥标识符再次作为判断标准。然后选定指派端口。

可以得到下列基本观察结果：

- 网桥在它的根端口接收BPDU并把它们发送到它的指派端口。
- 所有根端口和指派端口都处于转发状态。
- 根端口连到某网段的网桥不能作为那个网段的指派网桥。
- 每个网段只有一个指派端口。

2. 拓扑改变

在前面提到过，通常引入冗余的网桥来提高LAN的整体可靠性。由此，因为刚才描述的规程会把与这些网桥相关的一些（或所有）端口设成阻塞状态，所以必须在算法中加入规程，允许在网桥或端口发生故障时网桥以及相关端口的状态可以动态地改变。这称为**拓扑改变规程**。

一旦根网桥和相关现用拓扑建立，只有根网桥传送配置BPDU。每次呼叫定时器超时，它们会隔规定时间间隔发送到根网桥的每个端口。这些BPDU通过网络传播。所以，当每个网桥更新BPDU中所含的信息时，每个网桥和相关端口的状态会在规定时间间隔内得到证实。

为了使其他网桥能检测到何时发生故障，每个网桥为其所有端口保持一个**报文定时器**。在无差错状况下，每次收到配置BPDU它被重置。如果指派网桥或现用网桥的端口发生故障，配置BPDU停止从这个网桥或端口转发。其影响是故障网桥或端口下游网桥中的报文定时器会超时。

与端口相关的报文定时器的超时使得网桥协议实体调用**变成指派端口规程**。它由所有受影响的网桥调用。在执行这个规程之后，建立一个或多个新指派网桥和/或端口。如果当前根网桥发生故障，那么会选定新的根网桥。

另外，无论何时端口状态从阻塞变成转发，会从这个升级后的端口发送一个**拓扑改变通知BPDU**到根网桥方向上。在它和根网桥之间的所有指派网桥会注意到这个变化并通过
399 它们的根端口把它中继到根网桥。以这种方式，所有受故障影响的网桥都会知道拓扑变化。为了确保这些BPDU能可靠地到达根网桥，确认BPDU以及定时器与用来中继它们的发送规程相关。

在拓扑改变后，可以通过与每个网桥转发数据库中当前端口不同的端口到达连到每个LAN网段的终端站。因为网桥用来使其数据库中记录超时（就是说，自从定时器开始计时到超时为止，与终端站相关的记录没有传输任何帧）的定时器的超时时间设定相对较长，所以根网桥接到拓扑改变通知BPDU后发送的下一组配置BPDU有一个字段（位）通知网桥缩短这个时间。这样每个数据库中的现有记录超时，并且当每个站下次发送帧时建立新的记录。

3. 端口状态

为了确保在现用拓扑建立期间没有循环，网桥端口不允许直接从阻塞状态变为转发状态。相反，定义了两个中间状态：称为**监听状态**和**认知状态**。还定义了第五个状态（**失效状态**）允许网络管理站能通过网络发送特殊管理BPDU来永久阻塞特定网桥端口。

在各种状态，正常BPDU和信息帧可以或不可以转发。每个状态能转发的帧如下：

- 在失效状态，只能接收和处理管理BPDU。
- 在阻塞状态，只接收和处理配置和管理BPDU。
- 在监听状态，能接收和处理所有BPDU。
- 在认知状态，能接收和处理所有BPDU；信息帧提交给认知进程而不转发。
- 在转发状态，能接收和处理所有BPDU；能接收、处理和转发信息帧。

4. 状态变迁

端口状态间的变迁受到网桥协议实体的影响，并且在收到与端口相关的BPDU或者与网桥协议相关的定时器超时的情况下发生状态变迁。可能的变迁（（1）~（5））显示在图7-22的状态变迁图中。

通常，当网桥初次通电工作时，所有网桥端口进入失效状态。当通过网络接收到来自网络管理站的特定网络管理BPDU时，发生到阻塞状态的变迁（1）。类似地，网络管理站能够通
400 过网络在任何时刻发送管理BPDU，通过使特定网桥端口失效（2）。

一旦网桥从网络管理站接收到初始化命令，它把所有端口设成阻塞状态并开始发送配置BPDU。然后网桥参与前面提到的拓扑初始化规程。在这个规程中，网桥开始建立它的端口为根端口或指派端口。能够断定在这个规程中端口状态可以改变。例如，作为本地BPDU交换的结果，树结构中较低的网桥（就是指远离根网桥的网桥）会开始假定它们的一些端口为指派端口或根端口。而且，当BPDU从真正的根网桥向下过滤时，它们的状态也会改变。与根端口和指派端口直接从阻塞状态变迁为转发状态不同，它们经过中间的监听状态和认知状态，每个状态需要一个固定的时期。

当网桥确定它的一个端口是根端口或指派端口时，它从阻塞状态转变为监听状态（3）并

且**转发定时器**开始计时。如果定时器超时时端口仍然是根端口或指派端口，它会变迁到认知状态（5）。然后转发定时器重新开始计时，当定时器超时时重复同样的规程。但是这次如果端口仍然是根端口或指派端口，它就设成转发状态（5）。如果在此期间端口不再是根端口或指派端口，它会直接回到阻塞状态（4）。

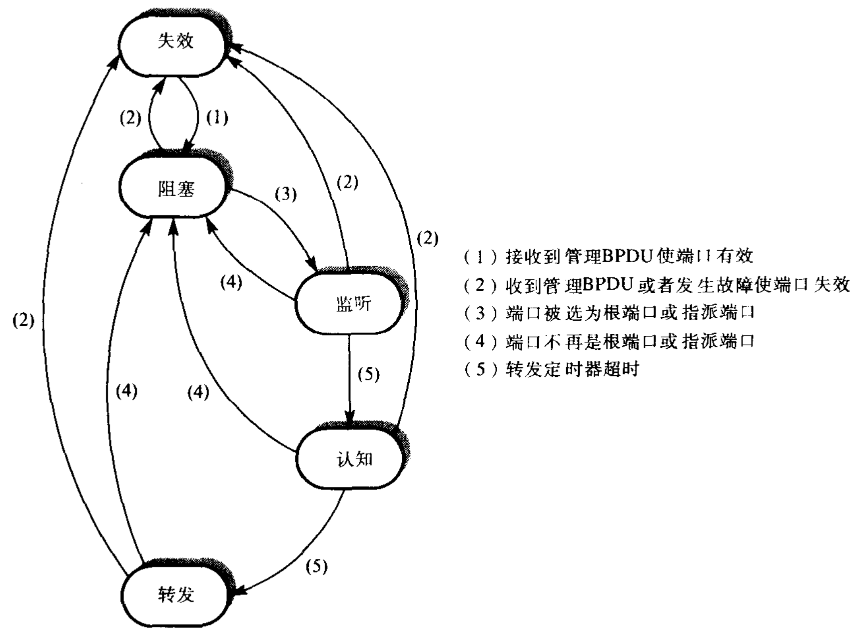


图7-22 端口状态以及状态变迁的各种可能 401

实例7-6

为了说明生成树算法的不同单元如何工作，考虑如图7-23(a)所示的桥接LAN。每个网桥的惟一标识符显示在代表该网桥的框内，而端口号显示在连接网桥到每个网段的内框中。通常，每个网段上加入额外网桥来提高网桥发生故障时的可靠性。同样，假定这个LAN刚开始服务。 402

根据下列情况确定现用（生成树）拓扑：

（a）所有网桥有相同的优先级并且所有网段具有与自身相关的相同指派成本（比特率）；

（b）除了网桥B1发生故障，其余情况同(a)；

（c）所有网桥处于服务中，但是网段S1、S2、S4和S5有三倍于网段S3和S6的指派成本，就是说网段S3和S6有更高的比特率；

（d）（c）中各网段有相同的指派成本，但是网桥B5的优先级（由网络管理）设置成比其他网桥更高的级别。

解：

（a）

（i）首先通过配置BPDU的交换建立网桥B1为根网桥，因为它有最小的标识符。

（ii）在配置BPDU交换后，每个端口的根路径成本（RPC）会计算出来。它们如图7-23(b)所示。

（iii）然后选择每个网桥的根端口（RP），因为根端口有最小的RPC。例如网桥B3，端口1的RPC是1而端口2的RPC是2，所以选择端口1。网桥B2的两个端口有相同RPC，选择端口1是因为它有更小的标识符。选定的RP也显示在图中。

（iv）B1是根网桥，所以它所有端口的指派端口成本（DPC）都为0。因此它是网段S1、S2和S3的指派端口。

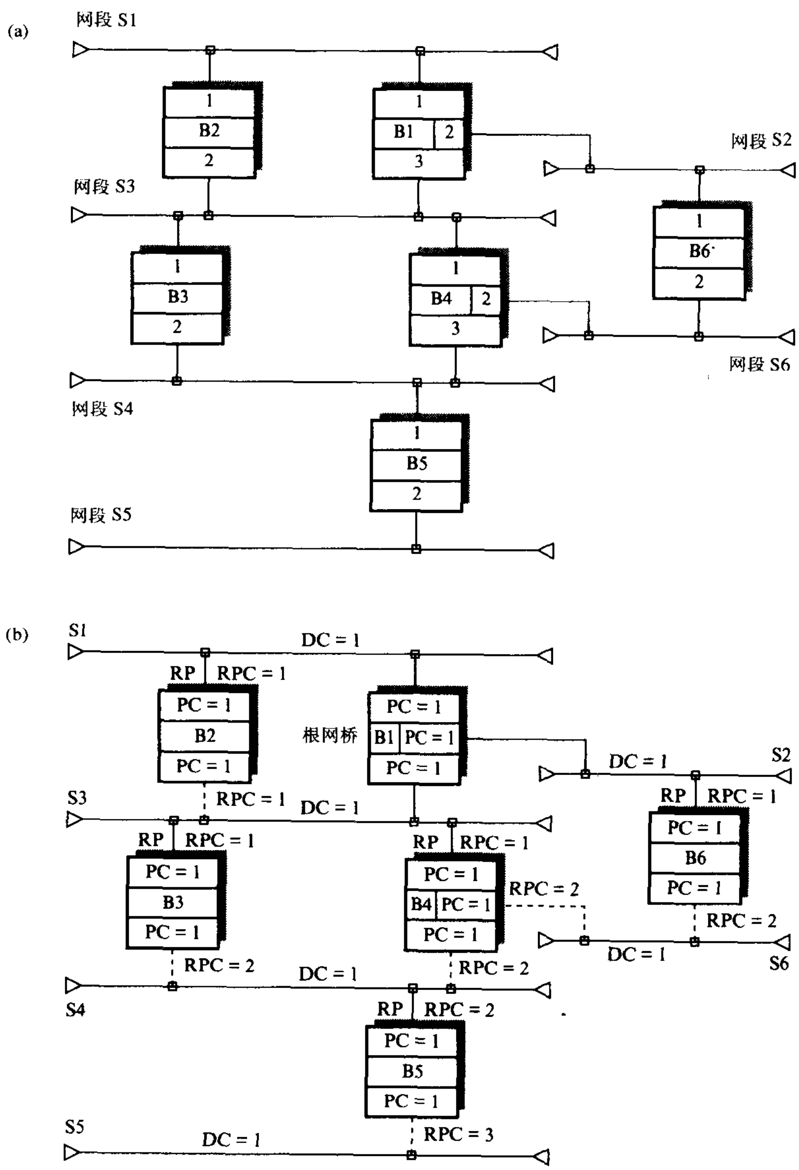


图7-23　现用拓扑生成实例

(a) LAN拓扑　(b) 根端口选择

(c) 指派端口选择　(d)～(g) 现用拓扑实例

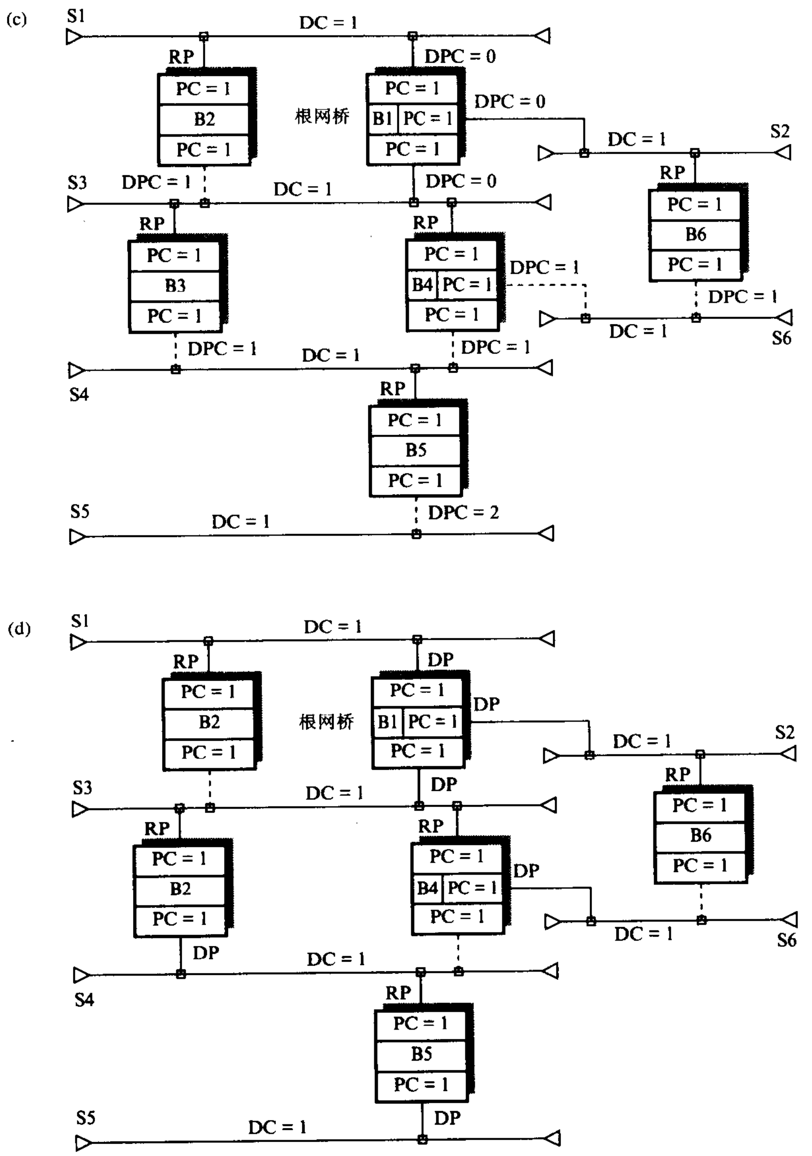

(c)
S1
DC = 1
RP
DPC = 0
PC = 1
B2
PC = 1
根网桥
PC = 1
B1
PC = 1
PC = 1
DPC = 0
DC = 1
S2
S3
DPC = 1
DC = 1
DPC = 0
RP
PC = 1
B6
PC = 1
RP
PC = 1
B3
PC = 1
RP
PC = 1
B4
PC = 1
PC = 1
DPC = 1
DPC = 1
DC = 1
S6
DPC = 1
DC = 1
DPC = 1
S4
RP
PC = 1
B5
PC = 1
S5
DC = 1
DPC = 2
(d)
S1
DC = 1
RP
DP
PC = 1
B2
PC = 1
根网桥
PC = 1
B1
PC = 1
PC = 1
DP
DC = 1
S2
S3
DC = 1
DP
RP
PC = 1
B6
PC = 1
RP
PC = 1
B2
PC = 1
RP
PC = 1
B4
PC = 1
PC = 1
DP
DC = 1
S6
DP
DC = 1
S4
RP
PC = 1
B5
PC = 1
S5
DC = 1
DP

图7-23　（续）

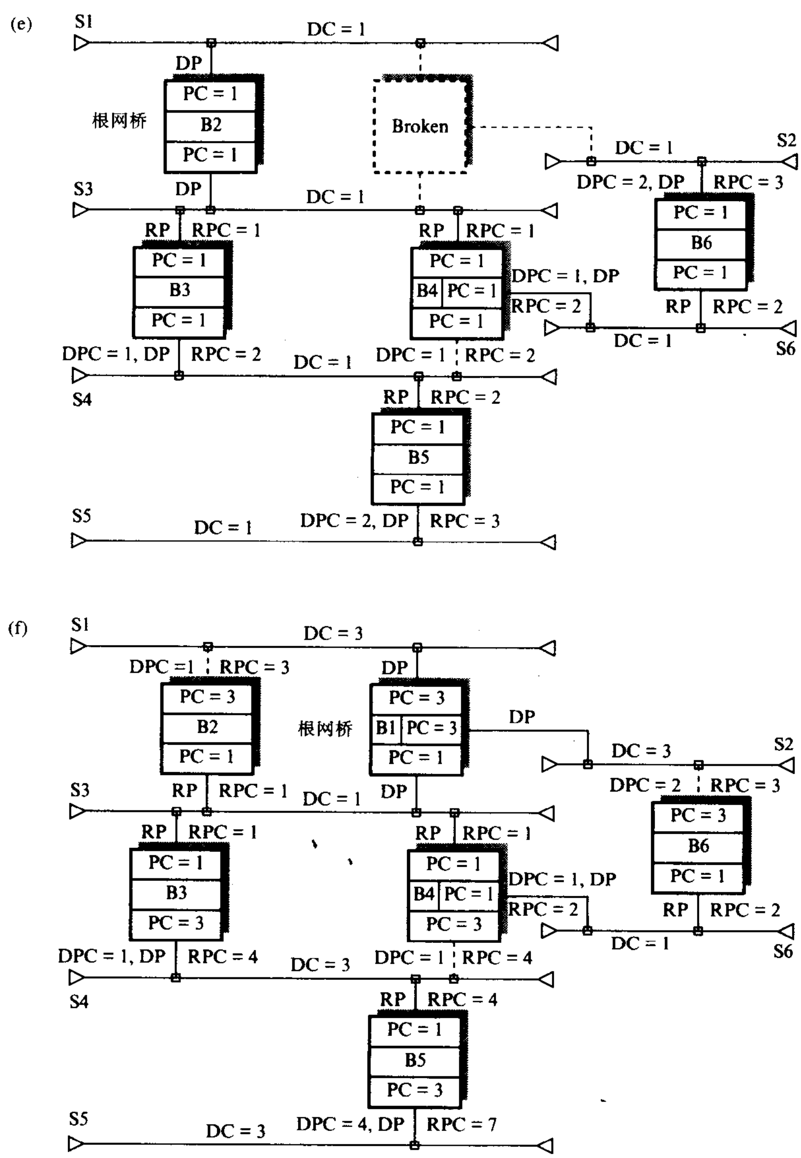

(e)
S1
DC = 1
DP
PC = 1
B2
PC = 1
根网桥
Broken
DC = 1
S2
DPC = 2, DP
RPC = 3
S3
DP
DC = 1
RP
RPC = 1
RP
RPC = 1
PC = 1
B6
PC = 1
PC = 1
B3
PC = 1
PC = 1
B4
PC = 1
PC = 1
DPC = 1, DP
RPC = 2
RP
RPC = 2
DC = 1
S6
DPC = 1, DP
RPC = 2
DC = 1
DPC = 1
RPC = 2
S4
RP
RPC = 2
PC = 1
B5
PC = 1
S5
DC = 1
DPC = 2, DP
RPC = 3
(f)
S1
DC = 3
DPC =1
RPC = 3
DP
PC = 3
B2
PC = 1
根网桥
PC = 3
B1
PC = 3
PC = 1
DP
DC = 3
S2
DPC = 2
RPC = 3
S3
RP
RPC = 1
DC = 1
DP
PC = 3
B6
PC = 1
RP
RPC = 1
RP
RPC = 1
PC = 1
B3
PC = 3
PC = 1
B4
PC = 1
PC = 3
DPC = 1, DP
RPC = 2
RP
RPC = 2
DC = 1
S6
DPC = 1, DP
RPC = 4
DC = 3
DPC = 1
RPC = 4
S4
RP
RPC = 4
PC = 1
B5
PC = 3
S5
DC = 3
DPC = 4, DP
RPC = 7

图7-23 （续）

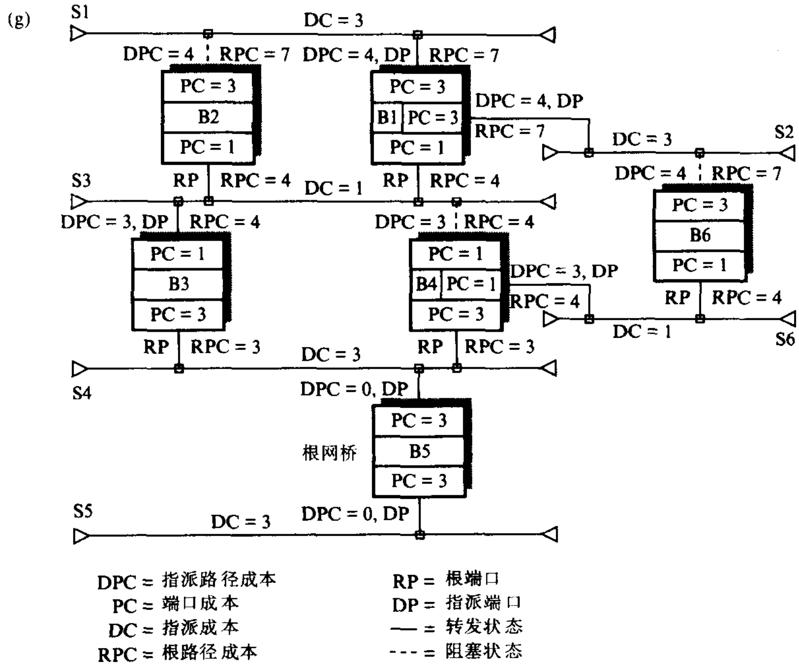


图7-23 （续）

（v）对于网段S4，网桥B5的端口1是RP，因此它不参与指派端口选择。其他两个连到S4的端口的DPC都是1。因为B3的端口2有较小的标识符，所以它被选为指派端口。

（vi）对于网段S5，与它相连的惟一端口是B5的端口2，所以它被选为指派端口。

（vii）最后，对于网段S6，两个端口的DPC都是1，因此B4的端口2被选为指派端口而不是B6的端口2。

DPC如图7-23(c)所示，由此产生的现用拓扑如图7-23(d)所示。

（b）

（i）首先作为拓扑改变规程的部分，交换BPDU建立B2为新的根网桥，因为现在它有最小的标识符。

（ii）然后计算每个端口新的RPC，并且建立每个网桥的RP。

（iii）因为B2现在是根网桥，它的端口成为S1和S3的指派端口。

（iv）对于S2，与它相连的惟一端口是B6的端口1，因此它选为S2的DP。 403

（v）对于S4，两个争用端口又有相等DPC，因此选B3的端口2为它的DP。

（vi）对于S5，选B5的端口2为DP。

（vii）最后对于S6，与它相连的两个端口有相等DPC，因此选择B4的端口2为DP。

修改后的现用拓扑如图7-23(e)所示。

（c）

（i）根网桥又是B1，因为它有最小的标识符。 404

（ii）RPC和RP如图所示，注意B6的端口2现在是RP，因为它现在的RPC比端口1低。

（iii）B1是根网桥，由此它的端口是S1、S2和S3的指派端口。

（iv）对于S4，B3的端口2又选为它的DP。

（v）对于S5，B5的端口2又选为它的DP。

（vi）对于S6，B4的端口2又选为它的DP。

405 新的现用拓扑如图7-23(f)所示。

（d）

（i）现在网桥B5有最高优先级，因此被选为新的根网桥。

（ii）因为是根网桥，所以它的端口成为S4和S5的指派端口。

（iii）每个端口的新RPC以及由此得到的RP如图所示。

（iv）对于S1，选择B1的端口1为它的DP。

（v）对于S2，选择B1的端口2为它的DP。

（vi）对于S3，选择B3的端口1为它的DP。

（vii）对于S6，选择B1的端口2为它的DP。

现用拓扑如图7-23(g)所示。

7.6.2 拓扑调整

可以从实例7-6推断出，如果LAN中所有网桥有相同的优先级，生成树算法不可能得到关于带宽使用方面的最优现用拓扑。拓扑调整可能成为大型网络中的一个重要因素，因为它对于任何可用高比特率网段的使用最大化显得相当重要。

406 每个网桥标识符中包括优先级字段，用以帮助达到这个目标。虽然一个网桥的惟一标识符字段在网桥制造时就固定了，但是优先级字段可以通过网络由网络管理站动态设置。响应网络管理命令的网桥称为**被管理网桥**（managed bridge）。通过选择性地设置网桥优先级，网络管理站能最优化或调整整个网络的性能。

通常，通过建立物理拓扑的计算机模型来观察不同现用拓扑和通信量分布情况下整个LAN的性能，由此来确定潜在瓶颈。对选定网桥优先级的仔细选择可以使现用拓扑性能最优化。

7.6.3 远端网桥

许多大型公司分布在一个国家或多个国家的若干地区（以及LAN）。除了能在单一地区内连到同一LAN的各个站之间交换信息外，许多大型公司要求在连到不同地区LAN的各个站之间交换信息。显然，需要一种设备来互连这些LAN。

可能有许多可选方法。一种解决方案是将公共（或专用）分组交换网络用于LAN间通信。另一种方案是使用有限数量的永久虚拟电路或动态建立交换连接。但是，两种方案都需要完整的网络层地址用于路由选择，这使得有必要使用**路由器**连接每个LAN到分组交换网络。我们会在第9章讨论网际互连时充分讨论这些解决方案。

一种更简单的可选解决方案是用租用（专用）线路互连各LAN。虽然这种方法失去了一些使用路由器能获得的优点，但通常能提供更快的中继服务。

正像第2章中提到的，从公共或私有电话公司租用线路在LAN分布组（在这个例子中）间建立直接的（点对点）链路网络。通常，由此产生的WAN中不涉及路由选择。每个帧头部使用48位MAC地址。LAN中使用的MAC地址通常在整个网络（就是说所有LAN）中是惟一的。由此，MAC地址和网桥能用来提供路由选择功能。这些网桥通常是一个端口直接连到跨地区骨干子网而另一个端口连到租用线路上。在租用线路的另一端使用类似的网桥。为了区分这些网桥和用于互连LAN网段的网桥，称它们为**远端网桥**。基于远端网桥的一个网络实例如图7-24所示。

许多大型公司以这种方式使用租用线路来互连每个地区内的（专用）电话交换机，由此

407 建立跨企业专用电话网络。用于数据通信的租用线路通常与用于电话的线路整合。租用线路

的比特率范围从56kbps（欧洲是64kbps）的倍数到1.54Mbps（欧洲是2.048Mbps）的倍数。这些线路可能跨越长距离意味着，当确定线路的传输时延时必须考虑它们的传播时延。

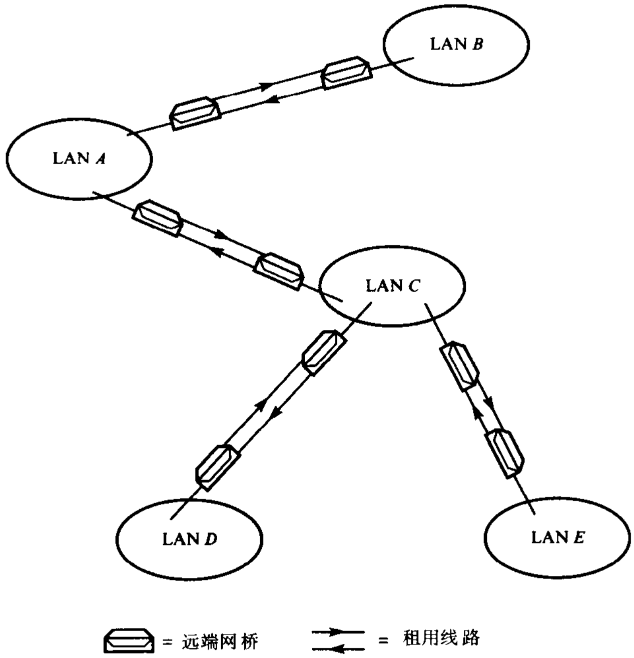


图7-24 LAN通过远端网桥互连

总体上，租用线路的可靠性明显比LAN网段低，因此通常有备用路径（线路）以防故障。虽然原则上生成树算法能应用在远端网桥（这样把生成树算法的应用范围扩展到整个网络），但实际上并不总是可行的。

在实例7-6中看到，使用生成树算法，与选定网桥（比如那些引入的用来增强可靠性的网桥）相关的一些端口被设成阻塞状态以确保现用生成树拓扑。使用租用线路，意味着可用的线路可能未使用，因为与它连接的网桥端口处于阻塞状态。不像用于LAN的传输介质，租用线路很贵，因此这对于最大化它们的利用率很重要。

在许多情况下，租用线路是更大的跨企业语音数据网的一部分。通常，这种网络有一个完备的网络管理设备，其主要任务之一是在租用线路发生故障时重新给每个语音和数据业务分配带宽。 408

一种通用解决方案是网络管理站（通过网络）在线路发生故障时动态指定可选线路（信道）。远端网桥不涉及生成树算法，但简单地执行基本认知和转发（过滤）操作。虽然这会引起重新配置期间性能的微小下降，但通常使得可用传输容量能被更有效地使用。

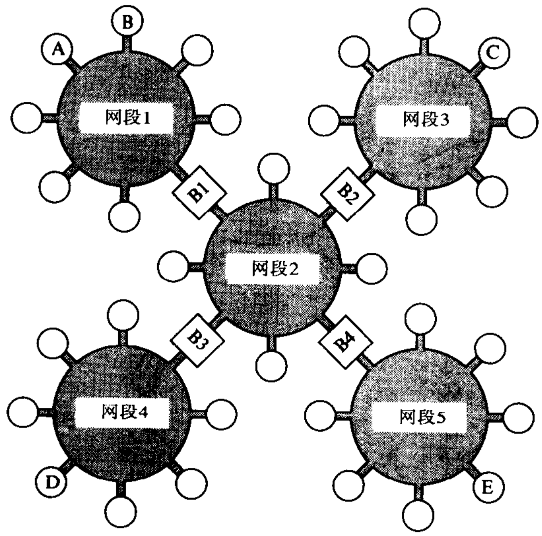
7.7 源路由选择网桥

虽然可以在任何类型LAN网段使用**源路由选择网桥**，但主要把它们用于互连令牌环LAN网段。基于源路由选择网桥的一种典型网络如图7-25(a)所示。

基于源路由选择网桥LAN和基于生成树网桥LAN之间的主要差别在于后者的网桥以相对于终端站透明的方式共同执行路由选择操作。相反，使用源路由选择，终端站执行路由选择 409

功能。使用源路由选择，在传输帧前由站来确定帧到达每个目标要经过的路由。这个信息插入帧头部并且每个网桥用它来确定是否需要把接收到的帧转发到另一个网段。路由选择信息由网段—网桥、网桥—网段标识符的序列组成。实例网络中选定站的路由选择表如图7-25（b）所示。

(a)



(b)

A的路由选择表：目标B = 网段1（同—网段）
C = 网段1，B1，网段2，B2，网段3
|
|
E = 网段1，B1，网段2，B4，网段5
B的路由选择表：目标A = 网段1（同—网段）
|
|
D = 网段1，B1，网段2，B3，网段4
E的路由选择表：目标A = 网段5，B4，网段2，B1，网段1
|
|
C = 网段5，B4，网段2，B2，网段3

图7-25 源路由选择桥接LAN实例
(a) 拓扑 (b) 路由选择表记录

接到每个帧，网桥只需要在帧头部的路由选择字段查询它的标识符。只有在其标识符存在并且后面跟着连到它的某个输出端口的网段标识符时才转发这个帧到指定的LAN网段。否则它就不转发。在任意情况下，这个帧在环接口被网桥复制，如果转发，帧尾部的帧状态（FS）字段中的地址识别（A）位和帧复制（C）位被设置，用来向源站（网桥）说明它已经被目标站（网桥）接收（转发）。

7.7.1 路由选择算法

每个帧含有的路由选择信息字段紧跟普通（IEEE 802.5）信息帧头部的源地址字段。这样修改后的格式如图7-26(a)所示。

因为不总是需要路由选择信息字段（例如，源站和目标站在同一网段上），源地址的第一位（单独/群（I/G）地址位）用来说明这个帧中是否存在路由选择信息（1表示存在而0表示不存在）。这可以做到，因为帧中的源地址必须总是单独地址，因此不需要I/G位用于此目的。

如果路由选择信息存在，其格式如图7-26(b)所示。路由选择信息字段由一个**路由选择控制字段**和一个或多个路由指示字段组成。路由选择控制字段本身又由三个子字段组成：帧类型、最大帧长度和路由选择字段长度。除了普通信息帧之外，还有其他两个帧类型与路由选择算法有关。**帧类型**说明了帧的类型。

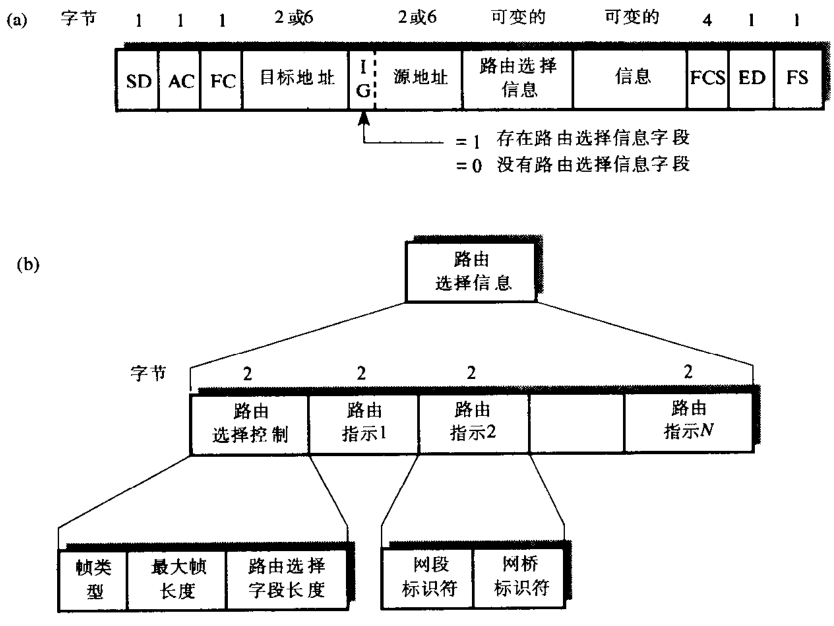


图7-26 令牌环帧格式

(a) 路由选择信息字段的位置 (b) 路由选择信息字段的结构

源路由选择网桥可以用来互连除了令牌环的不同类型的LAN网段。因为每种网段类型有不同的最大帧长度，所以**最大帧长度**字段在连到LAN的两个站间传输帧时确定能使用的最大帧长度。

在发送路由查找帧前，站设置最大帧长度字段为整个LAN中（已知）最大帧长度。在网桥转发该帧到网段前，先用新网段的（已知）最大帧长度检查这个字段。如果后者较小，网桥就把帧长度字段减小成较低的值。以这种方式，源站接到相应的路由应答帧，就能在准备传输帧到目标时使用该信息了。 410

因为帧从源到目标所经过的网段（和网桥）的数量可能会变化，**路由选择字段长度**说明路由选择信息字段的其余部分存在的**路由指示**的数量。每个路由指示由一对网段网桥标识符组成。

与路由查找算法相关的两个额外帧类型是**单路由广播帧**和**全路由广播帧**。为了找到路由，站先产生并发送一个单路由广播帧，它的路由选择字段长度为0而最大帧长度设成整个LAN已知的最大值。像生成树网桥一样，源路由选择网桥以混杂方式操作，并由此在每个端口接收和缓冲所有帧。网桥接收到单路由广播帧就简单地在连到它其余端口的每个网段广播这个帧的副本。因为这个规程由LAN中的每个网桥重复执行，所以帧的副本会在LAN中传播并且由此被期待的目标站接收到，不论它连接在哪个网段上。

如图7-21中指出的，如果在LAN拓扑中有冗余网桥（由此存在回路），那么会有多个帧副本在LAN中传播。为了防止这种情况出现，在发送任何路由查找帧前，网桥端口被配置以给

出一个生成树现用拓扑。从表面看，它与用在透明网桥的规程相同。但是，使用源路由选择
411 网桥，由此产生的生成树现用拓扑只用于路由选择最初的单路由广播帧。它确保只有帧的单一副本在网络中传播。生成树现用拓扑不用于路由选择普通信息帧或者全路由广播帧。

接到单路由广播帧，目标站就返回一个全路由广播帧给始收站。不像单路由广播帧，不强迫这个帧在每个中间网桥沿生成树现用拓扑传播。相反，接到这种帧，网桥简单地增加新的路由指示字段（由接收这个帧的网段标识符及其网桥标识符组成），路由选择字段长度加1，然后在其余每个端口网段广播这个帧的副本。

以这种方式，这个帧的一个或多个副本会经过两个站间的所有可能路由，最后被发送单路由广播帧的始发站接收。通过检查路由选择控制字段中的路由指示，源站能选择传输帧到目标站的最佳路由。然后这个路由加入到路由选择表中，随后传输帧到那个站时使用这条路由。

因为全路由广播帧不受沿着生成树现用拓扑传播的限制，所以每个网桥接到这种帧时必须采用额外的步骤来确保没有帧简单地在回路中循环。在发送全路由广播帧的副本到输出网段前，每个网桥先检查帧中的现有路由选择信息，来确定与入端口、出端口相关的网段标识符及其网桥标识符已经存在了。如果存在的话，这个帧的副本已经在路由中，所以这个帧的副本不发送到那个网段。

注意不必为每个传输帧执行路由查找操作。一旦到所需目标的路由被确定并加入（缓存）到站的路由选择表中，它会用于随后到这个目标的所有帧的传输。而且，因为多数站发送它的多数帧到有限数量的目标，所以与中等规模LAN的普通信息帧相比路由查找帧的数量相对较少。

实例7-7

假定如图7-27(a)所示的桥接LAN要使用源路由选择操作。还假定所有网桥有相同的优先级并且所有环有相同的指派成本（比特率）。当站*A*想发送帧到站*B*时推出下列结果：

（a）LAN的现用生成树

（b）单路由广播帧经过的所有路径

（c）全路由广播帧经过的所有路径

412 （d）*A*选定的路由（路径）

解：

（a）

（i）网桥B1有最小的标识符，被选为根网桥。

（ii）然后如图所示得出每个网桥的根端口。

（iii）现在能得到每个网段的指派端口，它们如图所示。

（iv）现用拓扑如图7-27(b)所示。

（b）单路由广播帧的路径为：

R1→B1→R2→B2→R3
　　　　　　└→B3→R5→B6→R6
└→B4→R4→B5

（c）全路由广播帧的路径为：

R6→B6→R5→B3→R2→B2→R3
　　　　　　　　　└→B1→R1
　　　　　　　　　　　└→B4→R4→B5→R5→B3

413

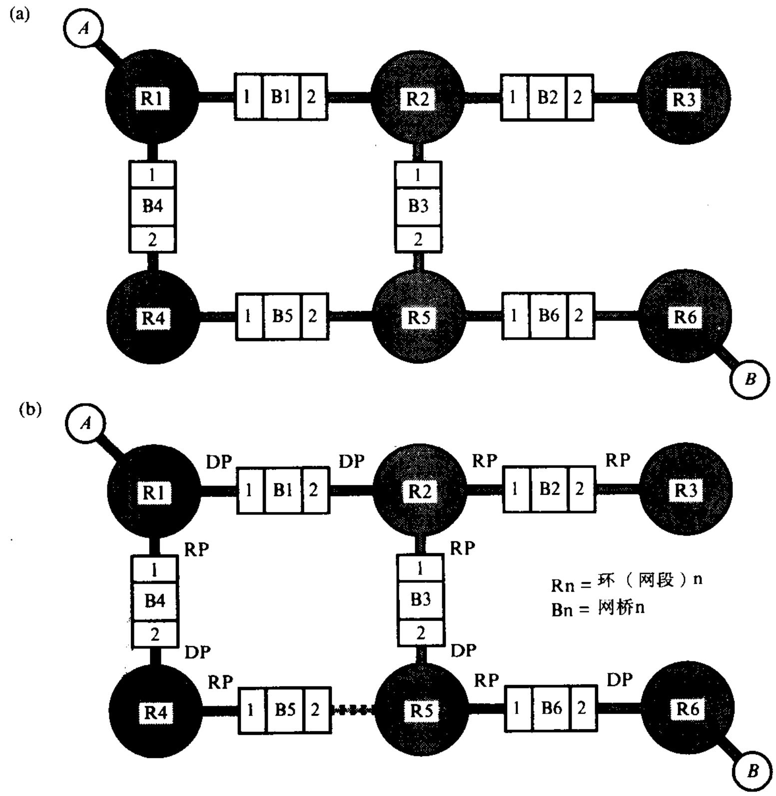


图7-27　源路由选择实例

(a) 拓扑　(b) 生成树

（d）因为每个环有相同的比特率，所以选定的路由（路径）为：

$$R1 \longrightarrow B1 \longrightarrow R2 \longrightarrow B3 \longrightarrow R5 \longrightarrow B6 \longrightarrow R6$$

或者：

$$R1 \longrightarrow B4 \longrightarrow R4 \longrightarrow B5 \longrightarrow R5 \longrightarrow B6 \longrightarrow R6$$

7.7.2　与透明网桥的比较

两种桥接方案（源路由选择和透明）都各有优缺点。下面基于不同标准比较两种方案。

1. *路由选择原理*

在基于透明网桥的LAN中，帧的路由选择对于终端站是透明的。后者简单地增加目标站的MAC地址到每个帧的头部，然后网桥共同经过网络路由帧到达它们所需的目标站。由此是网桥合作执行路由查找操作的。

在基于源路由选择网桥的LAN中，帧经过的路由在源站传输前插入到帧头部。由此帧经过的路由由终端站而不是网桥确定。源路由选择网桥根据帧头部的路由选择信息简单地从一个网段转发帧到另一个网段。

2. *路由质量*

在透明网桥的情况下，生成树算法确保在现用拓扑中没有回路，并且由此产生的树用作路由选择所有帧通过网络的基础。虽然分配网桥优先级和网段的指派成本会有帮助，但是由

此产生的生成树拓扑不可能在所有站间获得最佳路由。生成树算法不可避免地阻塞一些网段，而这些网段在两个站间路由选择帧时可作为理想网段使用。

使用源路由选择网桥，全路由广播帧确定源站和目标站间的所有可能路由。由此源站能选择到每个目标站经过的最佳路由。但是，注意这只在假定路由选择不仅基于路由中网段（由此网桥）的数量（**跳数**）而且基于每个网段的指派成本（比特率）时才成立。而且，通信
414 分配不均，仍然不能给出最佳路由。另一种可选的方法是简单地选择接收到的第一个全路由广播帧，因为它可能经历最小的时延。

3. 可用带宽的使用

为了确保生成树现用拓扑，透明网桥阻塞选定的端口。因此相应的连接网段不用于转发来自该网桥的帧，并且LAN中所有网段提供的全部可用带宽不是总被使用。

使用源路由选择网桥，在路由查找过程中所有可用网段被使用，所以理论上全部可用带宽被使用。但是，再一次说明，它只在源站选定的路由使用确保网络中所有网段负载平均的策略情况下才成立。实际上，不可能总是这种情况。

4. 路由转发开销

使用透明网桥，每个接收帧的处理过程会比使用源路由选择网桥花费时间更长。这是因为透明网桥必须在转发数据库中，为网络中每个活动的并且使用经过该网桥的路由的站维护一条记录。由此在大型LAN中，转发数据库可能会有许多记录，因此处理每个帧所花的时间（就是说为了确定与接收到帧所需目标站相关的网桥端口）可能会比较显著。

为了最小化数据库中的记录数量，每条记录使用一个**无活动定时器**。当定时器运转时，如果没有接收到任何来自该站的帧，那么这条记录就**变老**并被删除。显然，如果某条记录被删除，那么当下一次源站要发送新帧到这条变老的记录，接收到的帧就必须在处于转发状态的每个输出端口广播。为了最小化这种广播的数量，通常要选择无活动时间周期，这样的选择结果使得转发数据库在任何时刻仍然有相当数量的记录。

相比之下，使用源路由选择网桥，路由选择功能处理每个帧所需的开销较小，因为只需在帧头部的路由选择字段查询它的标识符。

在低比特率（比方小于16Mbps）LAN网段中的处理开销通常不显著。但是，在像FDDI等较高比特率LAN类型（传输速率超过100Mbps）中，如果可用传输带宽的使用最大化，那么最小化每个帧的处理开销会变得很重要。

5. 路由查找效率

使用透明网桥，一旦建立现用生成树拓扑，当站开始传输帧时网桥迅速地在转发数据库
415 建立记录。数据库中的记录必须变老，以确保它只为定期发送帧经过该网桥到相应目标站的那些站保存记录。结果是到达的来自数据库中再无有效记录的站的任何新帧的副本必须由网桥在其他每个端口上广播。由于现用生成树拓扑，产生的帧数量受到限制，从该网桥结点，每个树分支一个帧。

相比之下，使用源路由选择网桥，虽然初始单路由广播帧受到限制，必须沿着生成树传播，但是随后的来自目标的全路由广播帧不受这个限制。它意味着可能会有与这些帧相关的可观开销（关于传输带宽的使用和网桥处理），尤其在由许多网段和重复网桥组成的大型LAN情况下。

6. 可靠性

使用透明桥接LAN，网桥定期检查网桥或链路故障。它由根结点（隔规定时间间隔（由

呼叫定时器设置确定）发送配置BPDU）触发。如果发生故障，生成树算法的拓扑改变部分会建立一个新的现用拓扑用于剩余的网桥。

使用源路由选择网桥，源站必须检测何时发生故障，因为它们拥有路由选择信息。当网络故障发生时，每个站持有的路由选择信息可能会不正确，受其影响错误指向的帧（无可用路由的帧）会被发送。而且没有收到响应，帧一般会触发定时器（通常在协议栈的传输层），它会导致源站发送帧的另一个副本。假定故障仍然存在，这会不必要地增加网络负载。由此源站在故障发生后会尽可能地采取正确的措施（更新路由选择表来反映某个站或者某条新的路由不可用）。

一种方法是每个站有一个与它的路由选择表中每条记录相关的定时器（类似于网桥使用的无活动定时器）。如果定时器超时（就是说在相关定时器运转期间没有帧发送到目标站），那么在发送新帧到该站前会建立一条新的路由。显然，这个时间越短，越不可能产生含有不正确路由选择信息的帧。由此定时器设置与网桥使用的无活动定时器设置等同。因为每个定时器的处理必须由每个站而不是网桥执行，对网络负载的影响会相当大。

一种可选的方法是利用与生成树算法相关的拓扑改变规程。既然需要生成树算法用于路由选择单路由广播帧，因此也可以利用相关的拓扑改变规程。回忆与这个规程相关的内容，无论何时链路或网桥故障触发拓扑改变，根结点就发送一个拓扑改变通知BPDU。每个网桥使用该BPDU对转发数据库中的每条记录进行超时设定，这样这些记录会被更新以反映新的拓扑。原则上，接到这种BPDU，网桥会发送一个相应帧给连到该网段中的每个站，通知它们拓扑改变已经发生。这就不需要各个站来维护路由选择表中记录的定时器了，由此显著地减少了对于网络负载的影响。 416

7.7.3 与不同LAN类型网际互连

如7.3.1节指出的，由于网桥的存储转发操作模式，它们能用来互连以不同MAC方式操作的LAN网段。实际上，由于某些原因这并不简单。

1. 帧格式

由于三种基本LAN类型使用不同的传输模式（802.3和802.4的广播模式以及802.5的点对点模式）以及不同的MAC方式（802.3的CSMA/CD、802.4和802.5的令牌），因此有不同的帧格式，如图7-28所示。例如，802.3和802.4使用的广播模式意味着它们在每个帧头部使用前同步码来允许接收站在接收帧开始内容前获得时钟（位）同步。在令牌环中并不需要，因为所有站的本地时钟由持续循环的二进制数据流保持同步。

类似地，令牌MAC的使用意味着，802.4和802.5在地址字段前面有个帧控制字段，而在FCS后面有个结束定界符。802.3 LAN不需要这些字段，虽然它也使用长度字节以及一些用于短帧的额外填充字节。令牌环还在帧开始处使用额外的访问控制字段来管理优先级和保留特性，这些进一步混淆了情况。 417

总之，这些特性意味着当帧从一种类型LAN网段传输到另一种类型LAN网段时，它在转发到新的LAN类型前必须重新格式化。通常，这不是真正的问题，因为在实际帧内容发送（和存储）前大多数能识别的字段由LAN接口的不同MAC集成电路芯片集自动处理（添加和删除）。但是，这对于用于802.3 LAN的长度和填充字段行不通，因为在帧转发前整个帧内容必须由网桥用软件重新格式化。

帧重新格式化增加了网桥中的处理开销（和时延）。此外，更重要地，当帧转发时必须使用新的FCS字段。这也能轻易地做到，因为它会由MAC芯片集计算和加上。桥接LAN的一个

潜在的差错源是从每个网桥的存储空间存储和访问帧时引入的额外位差错。显然这种差错不会被新的FCS字段检测到。

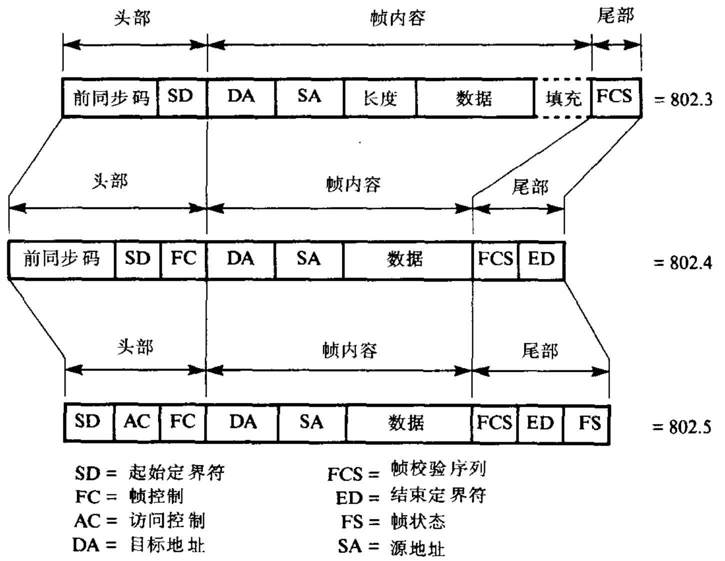


图7-28　LAN帧格式比较

当所有LAN网段是同一类型时，这个问题的通用解决方法是从源到目标使用相同的FCS字段。如果帧由网桥重新格式化（这种情况下任何引入的位差错（存储和处理期间）都不会被检测到)，这就不可能了。然后这些差错像剩余差错一样被携带转发。为了最小化这种概率，一般使用差错更正存储空间。

2. 比特率

一个传输比特率范围用于某种LAN。它们包括如下：

802.3　　1、2、10Mbps
802.4　　1、5、10Mbps
802.5　　1、4、16Mbps

如果低比特率网段接收到帧并转发到高比特率网段，不会有问题。如果是相反的情况，尤其LAN负载严重，那么帧必须在与较慢LAN相关的输出端口排队等待，问题就出现了。甚至两个LAN网段是同一类型时，这种问题也会出现。

例如，两个令牌总线LAN网段用网桥连接（一个工作在10Mbps而另一个工作在1Mbps），在大量通信期间快速建立帧。因为可用的存储空间量是有限的，网桥由于不能获得足够的存储空间就开始丢弃帧。虽然实际上受影响的源站中的传输协议实体会发起这些帧的另一个副本的重新传输，但是与这个操作相关的较长超时设定意味着帧的传输时延会显著增加。而且，
418 不能保证新的副本是否会经历相似的遭遇。

3. 帧长度

三种LAN类型使用不同的最大帧长度：802.3使用1518个字节，802.4使用8191个字节，而802.5中使用的最大帧长度由环的规模决定。如果帧先发送到802.4（令牌总线）网段，随后转发到802.3网段，问题就出现了。假定使用了最大帧长度，惟一可以克服的方法是在发送前，802.3网段接口的网桥会把较大帧分成较小的子信元，每个具有相同的目标和源地址。

虽然这可以做到，但是所谓的**分段**不是802.1（d）标准的一部分，因此网桥通常不提供这个功能。而且，它还增加了网桥中的开销。所以，如果标准网桥用于执行互连功能，惟一的解决方法是每个源站知道整个桥接LAN中的最大帧长度范围然后选择最小的。显然，这个解决方法破坏了802.1桥接LAN的透明特性，因此一般使用有额外分段功能的网桥。

一种可选的解决方法是使用称为**网桥—路由器**的设备而不是传统的（透明或源路由选择）网桥来执行不同类型网段间的互连功能。将在第9章看到，路由器在网络层而不是在MAC子层执行中继功能。而且，从一开始就指定它们执行从一种网络类型到另一种网络类型的协调功能（包括分段）。这样网桥—路由器能作为传统网桥或者路由器执行中继功能，用于互连不同网段类型。

习题

7.1 借助图描述下列集线器的操作原理：
　（a）无源中继集线器
　（b）有源交换集线器

7.2 画出交换集线器的示意图。解释它的操作，包括FIFO缓冲器和路由选择表的作用。为什么每个DTE需要单独的冲突检测线路。

7.3 关于100 Base 4T LAN，借助图解释如何使用4对UTP电缆获得半双工通信。在你的描述中要包括：
　（a）载波侦听和冲突检测功能是如何执行的
　（b）8B6T符号编码方案的操作原理以及这些符号如何通过电缆传输
　（c）NEXT对冲突检测的影响如何最小化

7.4 画出一级100VG AnyLAN（IEEE 802.12）网络的草图。一个源DTE在这个网络中产生一个要发送帧的请求。作为请求的结果，借助图说明集线中继器（和所有连接在中继器上的其他DTE）产生的信号。还要说明在帧传输期间和结束后产生的信号。 419

7.5 借助图7-9给出的关于一级8端口IEEE 802.12命令优先级传输周期的实例，给出缺少的第五个和第六个轮循周期的NNPP和HPNPP值。

7.6 对于图7-10（a）中给出的IEEE 802.12二级网络，得出与下列请求对应的帧传输序列。假定所有请求发生在空闲时段后的相同时刻，此时NPP为1：

$$\text{port } 1 = \text{H}, \text{port } 2.1/2.2 = \text{N},$$
$$\text{port } 3/4 = \text{H}, \text{port } 5.1/5.2 = \text{N},$$
$$\text{port } 6 = \text{H}$$

7.7 借助图描述数据是如何通过100VG AnyLAN网络电缆传输的。在描述中包括使用的编码方案的操作原理。

7.8 借助示意图，描述关于FDDI网络的下列组成部分的含义：
　（a）主环和次环
　（b）单连接站和双连接站
　（c）光耦合单元
　（d）布线集中器
　（e）极性双工连接器

7.9　画出基于跨地点FDDI骨干网络地区的布线计划图。假定有三幢大楼并且每幢大楼的转接板有四个端口。假定每层有一个布线集中器并且每个集中器有两个单连接站，画出布线计划图。

7.10　说明为什么在FDDI网络中使用4B/5B编码而不是差分曼彻斯特编码。

画出单一光纤的物理接口图并解释下列术语的含义：等待缓冲区，4B/5B编码/解码器和时钟同步。

7.11　100Mbps FDDI环网络长度能达100km并且最多可连接500个站。作出合理的假设，并得出以位表示的环等待明间。

7.12　使用环等待时间T_1和目标令牌循环计时器设置TTRT，得出FDDI环网络的利用率U和最坏情况访问时延A_{max}的表达式。由此得出下列环配置的U和A_{max}。假定TTRT为4ms，而信号传播速率为$2\times10^8 ms^{-1}$：

（a）2km环，有20个站

（b）100km环，有500个站

7.13　解释下列桥接LAN的相关术语含义：

（a）网桥

（b）多端口网桥

（c）大楼骨干网

（d）地区骨干网

7.14　列出并讨论网桥相对于中继器的优点和缺点。

7.15　画出典型网桥的体系结构图，用实例描述它的操作原理。在描述中包括下列术语：混杂模式、转发数据库、网桥认知和生成树。

7.16　关于生成树算法，解释下列术语的含义：

（a）根网桥

（b）根端口

（c）指派端口

（d）配置BPDU

（e）拓扑改变BPDU

7.17　画出说明网桥的端口状态和变迁可能性的状态变迁示意图，并解释涉及每个变迁的输出事件和动作。

7.18　对于如图7-23(a)所示的桥接LAN实例，得出下列情况下的现用（生成树）拓扑：

（a）所有网段有相同的指派成本但网桥B6有比其他网桥更高的优先级

（b）除了网桥B4发生故障，其余同（a）

420 （c）所有网桥处于服务中，但网段S3和S4只有其他网段路径成本的一半

7.19　借助图解释下列术语的含义：

（a）拓扑调整

（b）远端网桥

7.20　画出基于源路由选择的桥接LAN实例图，借助一些路由选择表记录解释帧如何从一个站到另一个站路由通过LAN。在描述中包括每个帧携带的路由选择控制信息的结构。

7.21　关于源路由选择桥接LAN的路由选择算法，解释下列术语的含义：

（a）单路由广播帧

(b) 全路由广播帧

7.22 假定如图7-27(a)所示的桥接LAN，但是引入额外的网桥用来连接环网段R3和R6，当站*A*要发送帧到站*B*时得出下列结果：

(a) 这个LAN的现用生成树

(b) 单路由广播帧经过的所有路径

(c) 全路由广播帧经过的所有路径

(d) *A*选定的路由（路径）

7.23 重复习题7.22，假定环R5以两倍于其他环的比特率工作，就是说它的路径成本是其他环的一半。

7.24 从下列方面比较透明桥接LAN和源路由选择LAN的操作：

(a) 路由选择原理

(b) 路由质量

(c) 带宽的使用

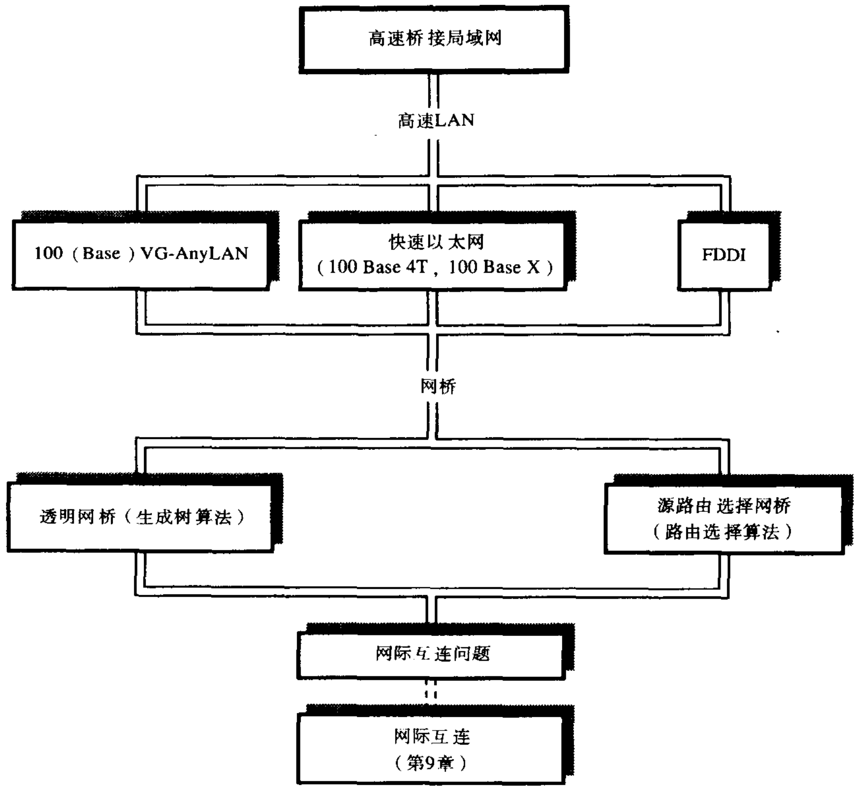
(d) 路由选择开销

(e) 路由查找效率

(f) 可靠性

7.25 借助图说明不同的基本LAN类型的格式，讨论由不同LAN网段类型组成的桥接LAN产生的问题。 421

本章概要


高速桥接局域网
高速LAN
100（Base）VG-AnyLAN
快速以太网
（100 Base 4T，100 Base X）
FDDI
网桥
透明网桥（生成树算法）
源路由选择网桥
（路由选择算法）
网际互连问题
网际互连
（第9章）

422

第8章 广 域 网

本章目的

读完本章，应该能够：

- 描述不同类型的公共数据网；
- 理解电路交换网和分组交换网之间的差异以及它们各自的优缺点；
- 描述分组交换网使用的X.25协议的结构以及ISO参考模型环境中分组（网络）层的操作；
- 描述打包和解包装置的功能以及各种协议的使用和操作；
- 解释如何使用X.21协议和电路交换数据网建立和清除呼叫；
- 理解ISDN的目标以及它的各种用户接口和协议；
- 描述帧中继网络和反向多路（复用）器的工作机制；
- 理解专用综合语音数据网的主要特点。

引言

在第2章和第5章中，讨论了利用PSTN传输数据。实际上，在公共数据网出现之前，它是位于不同地区的用户设备间进行数据传输的惟一可用方式。我们看到，通过PSTN建立的交换连接支持的用户数据传输率太低，一般小于9600bps。而且，电话呼叫根据时间和距离收费，因此通常情况下较长的距离和时间也会使得一般事务非常贵，尤其涉及到个人用户。

基于这些原因，许多大型组织建立自己的国内和国际专用数据网。一般，这些网从电话机构租用专用线路来互连专用交换结点或多路复用器（将在本章最后讨论）。虽然这种网络提供给用户安全、灵活性和最终控制，但是购买或租用设备需要高投资。所以，这些网络通常由大型企业（诸如主要结算银行）拥有和管理，它们既能支付初始投资费用又能产生足够的业务量来充分利用这种级别的投资。这些网称为专用**跨企业网**。

当许多专用数据网出现时，PTT机构会租用线路给某个企业，使得它能建立自己的专用数据网，但不会提供线路允许这些网互连起来。对公共数据网的需求以及随后公共数据网的建立，阻止了用户不断增长的希望建立专用网络把彼此设施互连起来的需求。

连接不同地域的DTE（或者实际上是跨地域的LAN）的网络的通用名称是**广域网**（WAN）。WAN包括公共数据网和跨企业专用数据网。本章讨论有关这类网络的操作和协议。

8.1 公共数据网的特征

多数WAN标准已经制定出来，并用于PTT和公共载波数据网。**公共数据网**（PDN）是由国家网络管理机构专门为传输数据建立和运行的网络。对PDN的基本需求是使不同厂商的通信设备能交互工作，反过来需要达成一致的标准以访问和使用这些网络。经过国家水平和国际水平的大量实验与讨论后，一组达成国际一致的标准被ITU-T接受用于PDN范围。这些标准称为X系列和I系列建议，它们包括用于这些网络的用户数据信令速率标准和用户接口标准。

PDN有两种主要形式——**分组交换**（PSPDN）和**电路交换**（CSPDN）——每种都有自己的标准。因为PSTN仍然广泛地用于数据传输，同样还建立了与这类网络的接口标准。通常，

每种这类网络的标准指的是ISO参考模型的最低三层，每一层的功能如图8-1所示。记住ISO参
424 考模型中的网络依赖层的特点是通过传输层到达高层协议层是透明的，即为高层提供网络无关的报文传输服务。我们会讨论PDN的不同类型以及已定义的用于每种类型的各种接口协议。

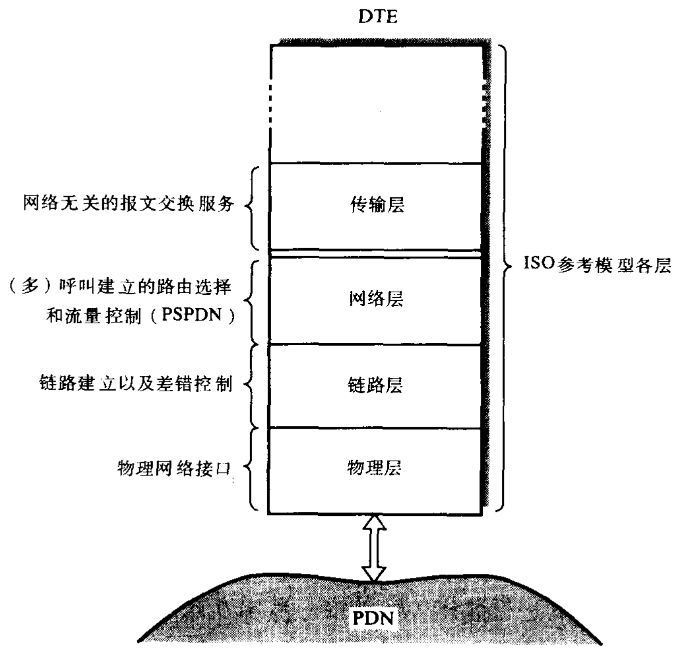


图8-1　PDN中的网络依赖协议层

8.1.1　电路交换和分组交换

在讨论有关PDN的各种接口标准前，简单介绍一下这些网络使用的两种不同类型交换方式的差异。通过电路交换网络的每条连接，在网络中建立主叫用户设备到被叫用户设备之间的一条物理通信通道。这个连接在呼叫期间由两个用户设备专用。电路交换网络的一个例子就是PSTN，实际上，所有通过PSTN建立的连接都是电路交换类型的。

在数据传输中，电路交换连接的一个特点是它有效地提供了一个固定数据传输速率的信道，两个用户设备都必须以这个速率工作。同样，在数据传输之前，网络必须先建立连接。当前，由于各交换点受所用设备类型的限制，PSTN建立呼叫所需要的时间相对较长（几十秒）。所以，通常传输数据时，建立连接并在事务期间保持连接。但是，新型程控交换机的广泛引
425 入以及整个网络数字传输方式的采用使得通过PSTN建立连接的时间迅速地缩短（几十毫秒）了。而且，用户设备数字传输能力的提高意味着可以为用户提供高比特率（一般64kbps或更高）的交换传输路径。这个路径无需调制解调器就能用于数据传输。产生的数字PSTN也认为是公共CSDN或者是综合业务数字网络（ISDN），因为这种网络同时支持数字化语音和数据。我们将在8.4节更详细地介绍ISDN。

虽然全数字电路交换网络的连接建立时间相对较快，但是这样的连接仍然具有固定数据传输速率，双方用户都必须使用这个传输率来发送和接收。相比之下，使用分组交换网络，两个通信用户（DTE）能以不同的数据传输率工作，因为数据经过网络接口的速率由每个用户设备单独调整。还有，使用分组交换网络无需建立全网的物理连接。相反，*所有要传输的*数据先被源DTE分成一个或多个报文单元，称为**分组**（或包）。这些包含有源DTE和目标DTE

的网络地址。由源DTE将包按位串的形式传递给本地分组交换交换机（PSE）。接到每个包，交换机先存储这个包，然后检查它所含的目标地址。每个PSE含有指明每个网络地址的出链路（传输路径）的路由选择目录。接着PSE在正确的链路上以最大的可用比特率转发这个包。这种工作模式常常称为分组存储转发。

类似地，沿路出方向的每个中转PSE接收（和存储）每个包，并将它和其他正被转发的包一起转发至相应的链路上。最后，根据目标地址将包送到目标PSE，再由目标PSE传递给目标DTE，如图8-2所示。

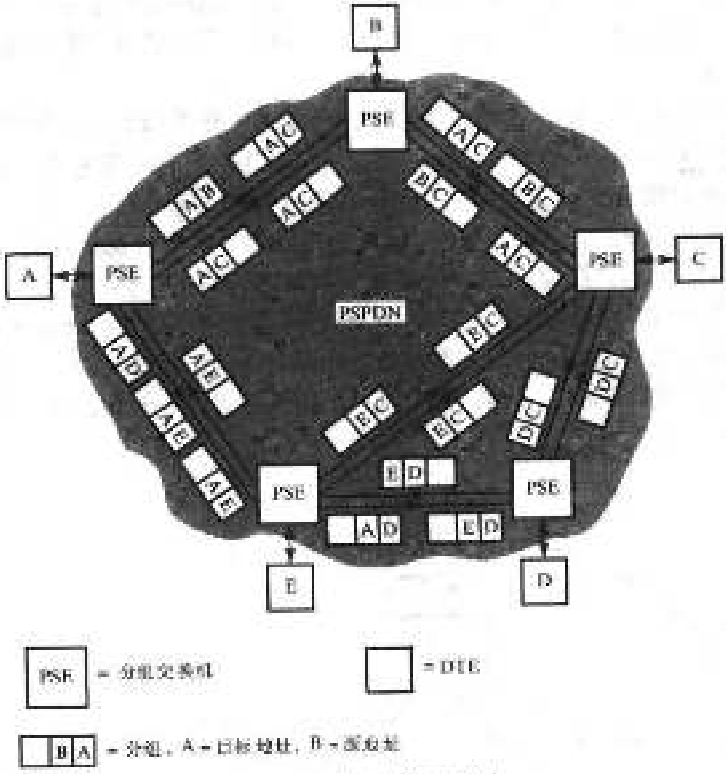


图8-2 分组交换示意图

我们看到，每个完整的事务过程只占用每个链路上可用带宽的（随机）一部分，这是因为来自不同源的包夹杂着来自不同网络链路上其他源的包。在极限情况下，可用带宽会从用户不发送任何数据时的零到用户持续发送包时的整个带宽。

在一个PSE上，可能有若干包从不同入链路上同时到达，并且每一个需要转发到相同的出链路上。如果在相同链路上，若干个特别长的包等待发送，则其他包会经历不可预知长时间的延迟。为了避免这种情况并确保网络有可靠的较快发送时间，每个包规定一个允许最大长度。因此，当使用分组交换网络时，在DTE中，递交到传输层的报文在传输前由源传输协议实体分成若干个较小的分组单元。在目标DTE中它们由对应传输协议实体重新装配成一个报 426
文，这对提供网络无关报文传输的传输层用户是透明的。

CSPDN和PSPDN的另一个不同是，CSPDN网络对传输数据不提供任何差错和流量控制，因此这必须由用户来完成。而对于PSPDN，在每条链路上由网络PSE提供复杂的差错和流量控制。因此，由PSPDN提供的服务级别通常比CSPDN高。

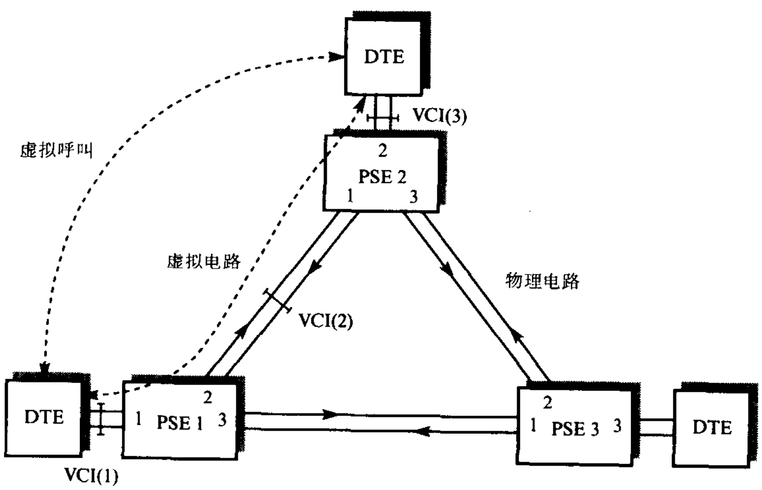
电路交换和分组交换提供给用户两种不同类型的服务。即使出现了全数字网络，这两种类型的服务仍然被支持，并提升至允许用户选择特定的服务。

8.1.2　数据报和虚拟电路

PSPDN通常支持的两种业务类型是：**数据报**和**虚拟呼叫**（电路）。我们通过用信件和电话呼叫方式来进行报文交换的类比以解释两种业务类型的差异。在第一种情形中，含有报文的
427 信件被邮政机构按独立（自包含）实体进行处理并且它的传送与其他信件是无关的。在电话呼叫的情形中，首先在网络中建立一条通信路径，然后开始报文交换。

数据报业务类似于通过信件方式发送报文，因为进入网络的每个包被当作与其他包无关的独立实体。包以独立的方式简单地接收和转发。因此，我们通常将数据报业务用于传输较短的单个包的报文。

如果一个报文含有多个包，我们选择虚拟呼叫业务。这类似于通过电话呼叫的方式发送报文，当我们使用这个业务时，在有关呼叫的任何信息（数据包）发送前，源DTE向本地PSE发送一个特殊的**呼叫请求包**，这个包除了含有必须的目标DTE地址外，还含有称为**虚拟电路标识符**（VCI）的参考号。它由PSE记录，这个包按前述方式通过网络转发。在目标PSE，在转发到所需目标DTE的出链路前，第二个VCI指定给呼叫请求包。然后，假定呼叫被接受，则相应的响应包返回到主叫DTE。此时，我们称在两个DTE间存在一条**虚拟电路**（VC）。然后进入信息传输阶段，所有与这个呼叫相关的数据包在每个网络接口链路上指定相同的参考号。用这种方法，源和目标DTE能轻易地区别来自同一链路但属不同呼叫的包。虚拟呼叫和VC间的关系如图8-3所示。



Virtual circuit = VCI(1) + VCI(2) + VCI(3)

VCI = 虚拟电路标识符
PSE = 分组交换机

	In		Out
PSE 1 路由选择表	VCI(1)/Link (1)	→	VCI(2)/Link (2)
	VCI(2)/Link (2)	→	VCI(1)/Link (1)
PSE 2 路由选择表	VCI(2)/Link (1)	→	VCI(3)/Link (2)
	VCI(3)/Link (2)	→	VCI(2)/Link (1)

428 图8-3　虚拟呼叫/虚拟电路关系和包路由选择原理

注意虽然对于用户来说VC看起来类似于电路交换网络建立的连接，但是它完全是一条纯

逻辑连接。而且，因为PSPDN除了提供用在链路级的规程外，还提供用在分组级的差错控制和流量控制规程，所以VC支持的业务级别很高。这意味着与一个特定呼叫相关的所有包在传送时无差错、按顺序又无重复的概率是非常高的。

通常，与一个呼叫相关的所有数据交换完成后，虚拟电路被清除并且相应的VCI被释放。但是虚拟电路可能被永久保留，因此经常需要与另一个用户通信的用户不必为每次呼叫建立新的VC。这称为**永久虚拟电路**（PVC）。虽然该用户必须为此付出代价，但每个呼叫的成本仅取决于传输数据的数量。前面我们提到，在电路交换网络中计费通常取决于距离和呼叫时间。

8.2 分组交换数据网络

定义连接DTE与PSPDN接口的国际上一致的网络访问协议是X.25。实际上，X.25是一组协议。各协议层如图8-4所示。从图8-4可以看到，组成X.25的三个协议层与端对端操作的传输层 429
不同，仅具有**本地意义**。

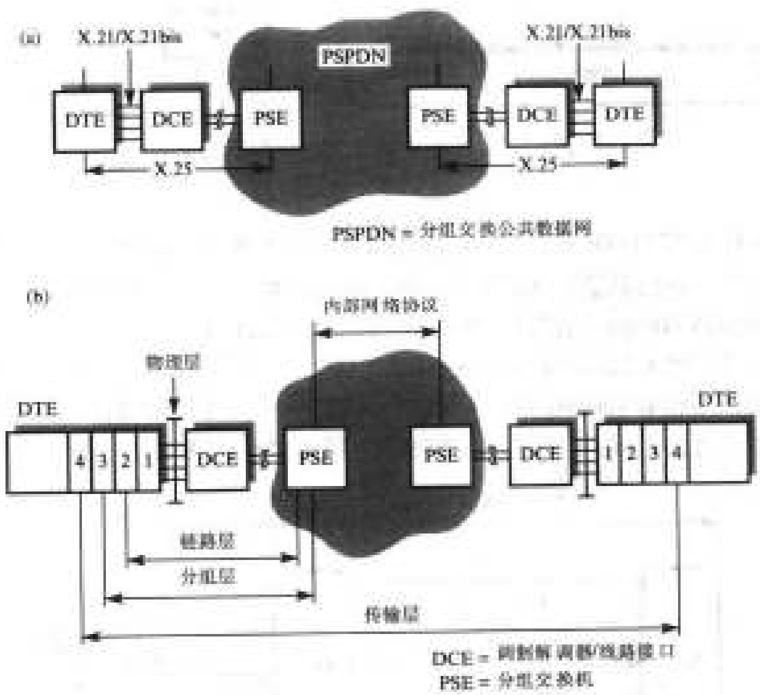


图8-4 X.25网络访问协议

(a) 适用范围 (b) 协议组成部分

在最低层，X.21接口标准定义DTE和提供PTT的本地DCE间的物理接口。X.25的链路层协议是LAPB，它是HDLC协议的一个版本，其功能是通过DTE和它的本地PSE间的物理链路向分组层提供无差错包传输业务。最后，数据分组层涉及称为**传输协议数据单元**（TPDU）的传输层报文的可靠传输，以及链路层控制的单一物理链路上的一个或多个虚拟呼叫（网络服务访问点（NSAP））的多路复用。报文单元和各层间的交互如图8-5所示。下面将分别考虑每一层。

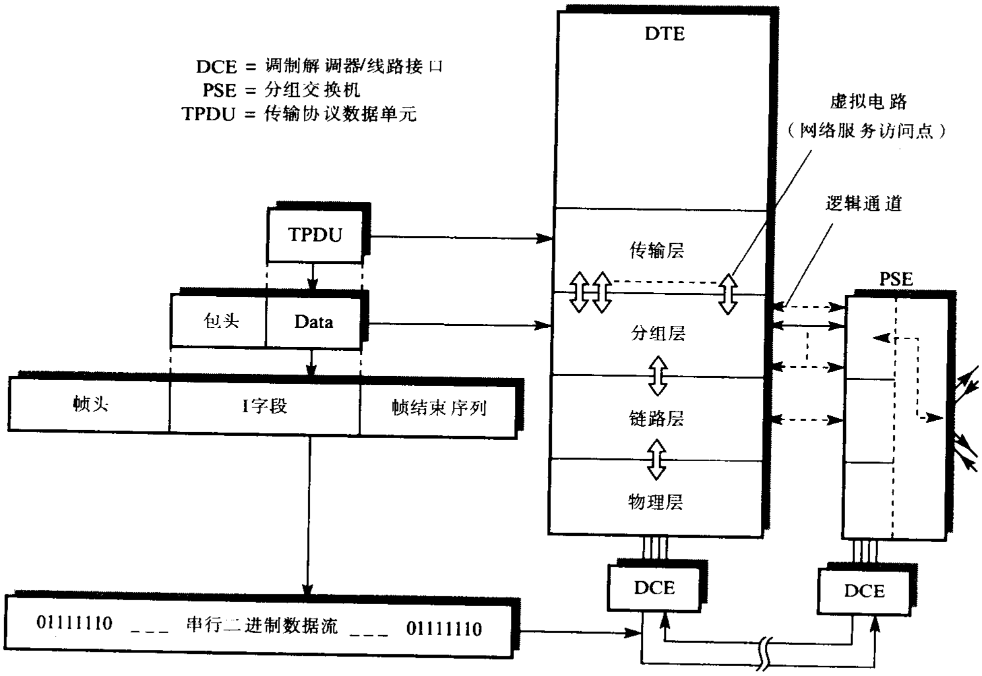


图8-5　X.25报文单元和层之间的交互

8.2.1　物理层

DTE和本地提供PTT的DCE之间的物理接口按ITU-T建议X.21规定。DCE发挥与同步调制解调器相似的作用，它的功能是在DTE和本地PSE间提供全双工、位串行同步传输路径。数据 430 传输速率从600bps到64kbps。在8.3节将看到，实际上X.21与全数字电路交换网络的接口是一样的。注意第二个标准X.21bis用于现有的模拟网络。X.21bis是EIA-232D/V.24的子集，因而现有的用户设备可以很容易地用这个标准接入网络。关于X.21和X.21bis的各种相互交换电路按照X.24建议来定义，如图8-6所示。每条线路的用途将在8.3节详细讨论。

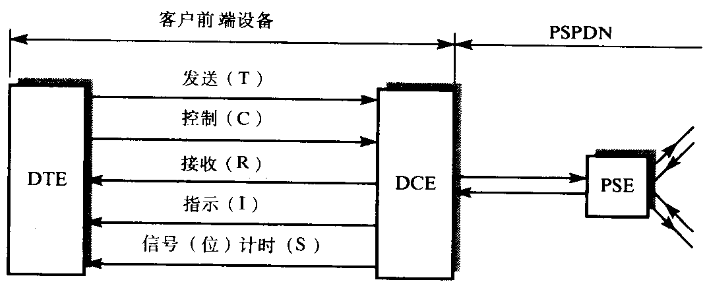


图8-6　X.21物理层接口电路

8.2.2　链路层

链路层（因为它在ISO参考模型中所处的位置，一般称之为第二层）的目的是向分组层提供在DTE和本地PSE间物理链路上的可靠（无差错和无重复）传输。链路层并不能识别包所在的逻辑信道，这只由分组层识别。因此链路层使用的差错控制和流量控制规程适用于所有包，而与包所属的虚拟电路无关。

链路层使用的帧结构以及差错控制和流量控制规程基于HDLC协议。在第5章中已介绍了

HDLC的基本操作。HDLC使用ABM操作，也称为ITU-T X.25标准文档中的LAPB。它因为取代较早的链路访问规程版本A，称为“链路访问规程”平衡式版本B。

在ISO参考模型环境中，链路层向上面的网络（分组）层提供的服务以及与链路层协议实体操作相关的PDU，概括在图8-7(a)中。服务原语按序表示，发起服务原语及其响应原语在同一行表示。记住每个分组层都能发起所示的三个服务请求。

采用ABM，DTE和PSE都以异步操作，因而在任何时刻，两者都可发起命令和响应的传输。且因为协议只控制点对点链路，就是说DTE和本地PSE间的链路的I帧流，所以每个帧的地址字段不用来传输网间地址信息。网间地址信息由I字段携带，因为由分组层处理网络寻址。
链路层的地址字段含有DTE或DCE（PSE）的地址：如果是命令帧，地址指明接收方的地址； 431
如果是响应帧，地址指明发送方的地址。如图8-7(b)所示。

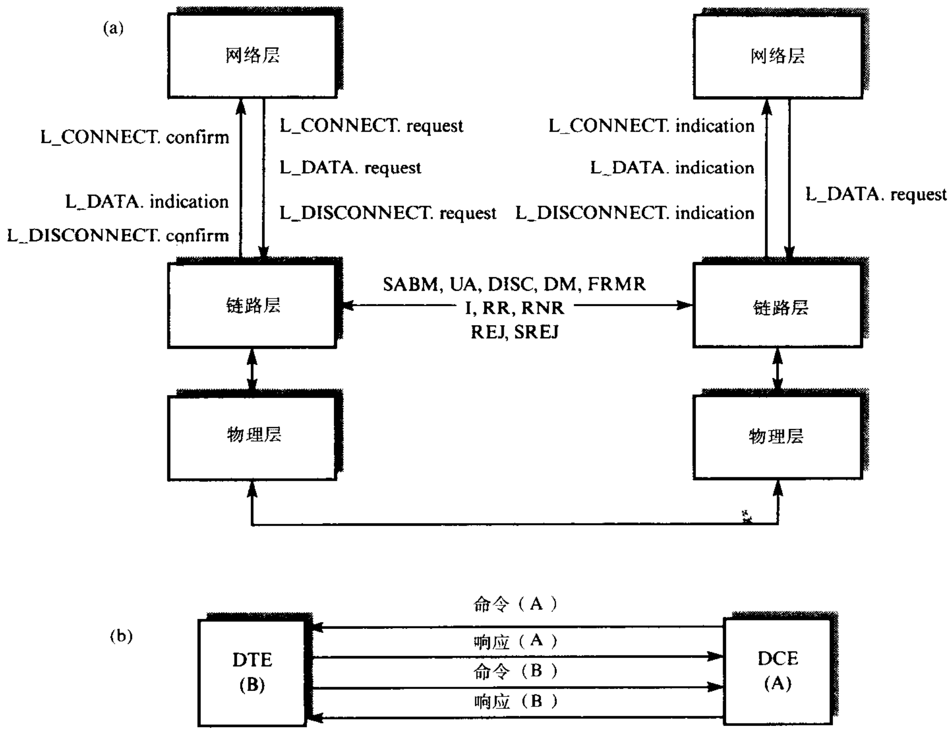


图8-7 链路层

(a) 概括 (b) 地址用法

8.2.3 分组（网络）层

在ISO参考模型环境中，分组层与网络层一致。同样，因为它在参考模型中所处的位置，简单地称分组层为第三层。传输层使用分组层提供的服务和一个或多个远端传输层交换TPDU。

1. 用户服务

网络（分组）层提供的用户服务列在图8-8(a)中，它们的使用顺序如图8-8(b)的时序图所示。可以看出，除了正常数据传输服务原语（N_DATA）外，还提供了进一步的服务原语（N_EXPEDITED_DATA）。这是个可选服务，它允许用户通过连接（逻辑信道）发送单数据包，
即使正常数据包流受流量控制的限制停止。另一个可选服务（N_DATA_ACKNOWLEDGE）特 432

别允许用户确认先前使用N_DATA服务发送的用户数据包的接收。最后，如果逻辑信道上的包流量不同步，N_RESET服务允许两个用户重新同步；而DISCONNECT服务用来清除虚拟电路。

(a)

原语	参数
N_CONNECT.request	被叫NSAP，主叫NSAP，QOS，收到证实选择，加速数据选择，NS_user 数据
.indication	被叫NSAP，主叫NSAP，QOS，收到证实选择，加速数据选择，NS_user 数据
.response	响应（被叫）NSAP，QOS，收到证实选择，加速数据选择，NS_user 数据
.confirm	响应（被叫）NSAP，QOS，收到证实选择，加速数据选择，NS_user 数据
N_DATA.request	NS_user 数据
.indication	NS_user 数据
N_DATA_ACKNOWLEDGE.request	-
.indication	-
N_EXPEDITED_DATA.request	NS_user 数据
.indication	NS_user 数据
N_RESET.request	发起者，原因
.indication	发起者，原因
.response	-
.confirm	-
N_DISCONNECT.request	发起者，原因，NS_user 数据，响应地址
.indication	发起者，原因，NS_user 数据，响应地址

NSAP = 网络服务访问点（地址）；QOS = 服务质量

(b)

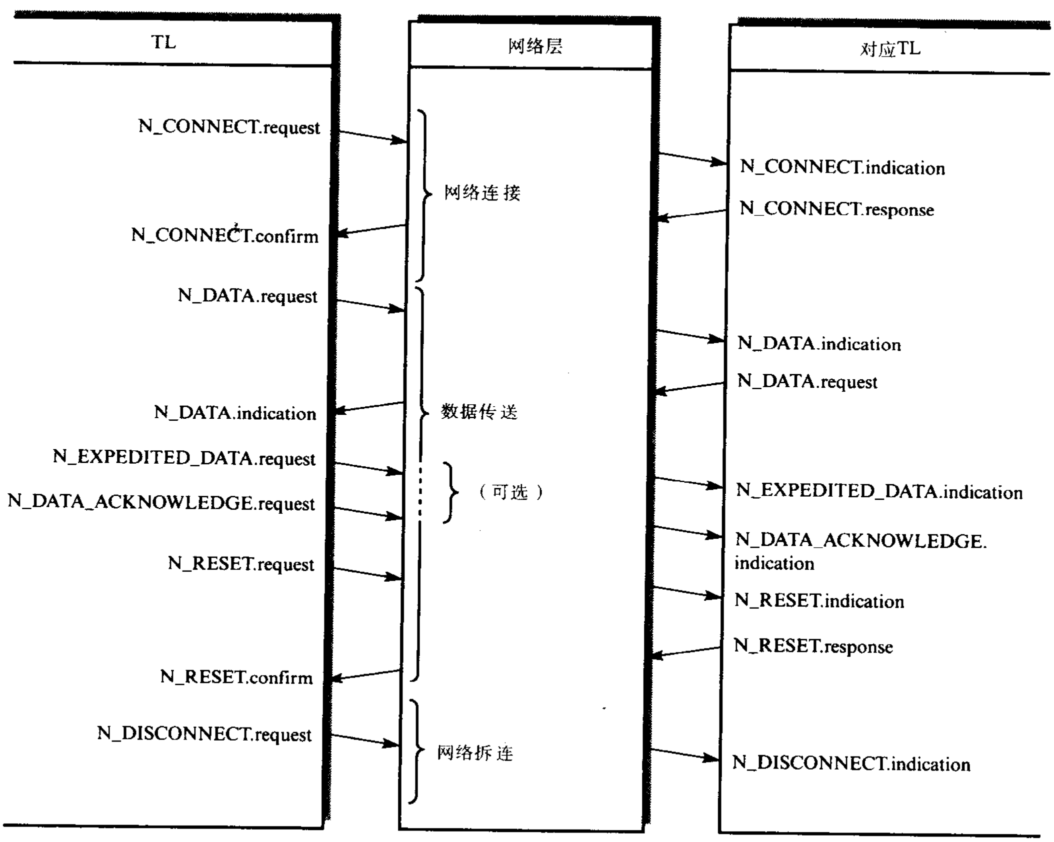


图8-8 网络层服务

(a) 原语和它们的参数 (b) 原语使用的顺序

所有原语都有相关的参数，列在图8-8(a)中。例如，与N_CONNECT原语相关的参数包括**被叫**（目标）NSAP和**主叫**（源）NSAP。**网络服务访问点**（NSAP）是物理连接点地址和称为**地址扩展**的内部（层间）逻辑信道（虚拟电路）标识符的连接。

服务质量（QOS）参数由两列参数组成：正在建立的连接从网络中预期的期望参数和最小可接受参数。这些包括（包）传输时延、残留差错概率、优先级（如果可用）、成本（呼叫费用）以及指定（而不是仲裁）路由。最后，两种选择参数允许两个对应的网络服务用户（传输协议实体）来协商是否在随后的数据传输阶段中包含可能使用的两个可选服务。

正如在图8-5中所见，传输层可能在某一时刻有若干个已建立的网络连接（呼叫），每一个呼叫与单一NSAP相关。这样分组层执行多路复用功能，所有的连接——VC和PVC——被多路复用到由链路层控制的单一数据链路上。然后每条虚拟电路上的包流量分别由分组层协议单独控制。 433

2. NSAP地址结构

多数国家有一个或多个公共载波PSDN，它们全球互连形成了国际PSPDN。由此与每个DTE相关的NSAP地址必须是全球惟一的网络地址。通常，因为整个网络由许多跨国家PSPDN组成，因此在整个网络环境中每个国家网络称为**子网**（subnetwork或subnet）。国际PSPDN中使用的NSAP地址结构由ITU-T X.121建议规定。实际上，所有与国际网络相关的地址都由ISO管理，ITU-T提出的标准格式如图8-9所示。

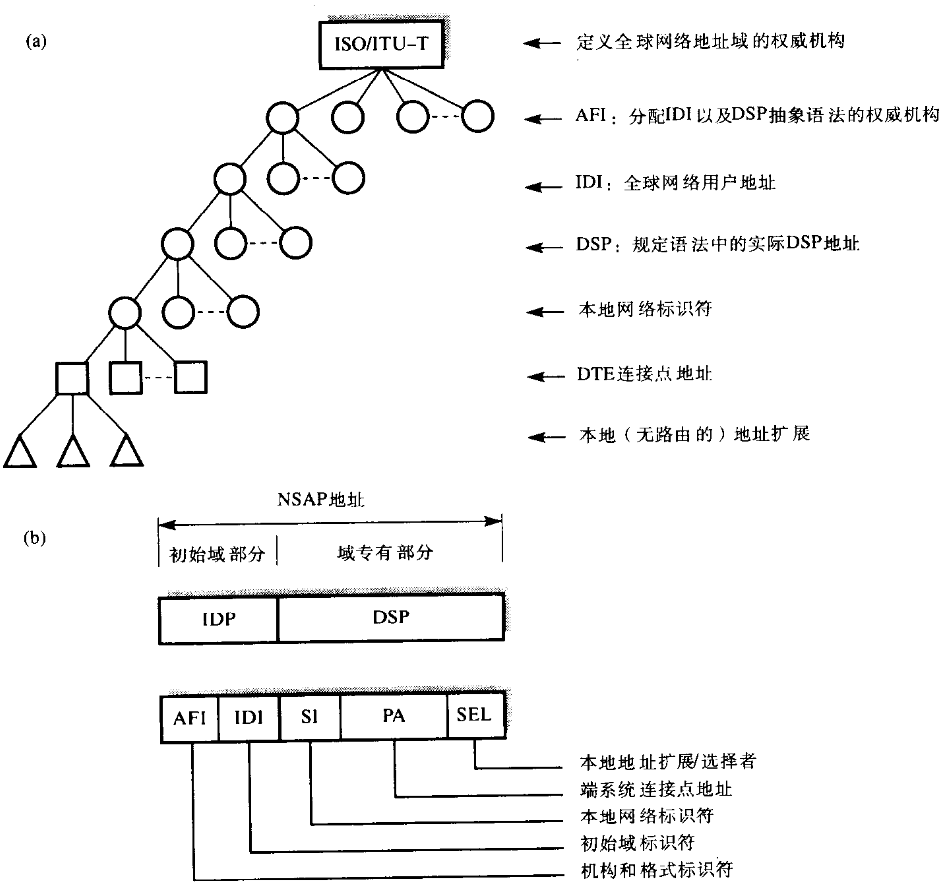


图8-9 NSAP地址

(a) 分级结构 (b) 地址内容 (c) 实例

(c)

	AFI值		IDI长度	DSP长度
	十进制	二进制	（十进制数）	（十进制数）
国际PSPDN-X.121	36	37	14	24
国际Telex-F.69	40	41	8	30
国际PSTN-E.163	42	43	12	26
国际ISDN-E.164	44	45	15	23
ISO指定国家代码-ISO DCC	38	39	3	35
本地（专用）	48	49	Null	38

图8-9 （续）

地址是分级的，并由两部分组成：**初始域部分**（IDP）和**域专有部分**（DSP）。两个部分定义成压缩BCD数字或纯二进制形式，NSAP总长度能达到40个十进制数字或20个字节。通常
434 使用的实际地址长度位于地址字段的前面，这样接收DTE能在正确的字节边界解释各个字段。

IDP由两个子字段组成：**机构和格式标识符**（AFI）和**初始域标识符**（IDI）。AFI指明了负责分配IDI的机构、IDI的格式和DSP的抽象语法。IDI指明了客户DTE的全球网络地址。如果有DSP，它允许定义更多的地址组用在客户地点。在国际PSPDN中，IDI是14个十进制数字
435 位的分级地址，由3个数字位的国家代码和1个数字位的网络代码，以及区域代码和本地号组成。通常，DSP也是分级的，由**子网标识符**（SI）和本地连接点字段组成。最后，**选择者**（SEL）子字段只有本地意义，不用来在网络中路由帧/包。例如，在X.25网络中它含有虚拟电路标识符，而在ISDN中它标识了8个可能终端中的某一个。

3. 包类型

与分组层协议（PLP）相关的包类型称为**分组协议数据单元**（PPDU）。PPDU的各种类型和使用如图8-10（a）所示，而结构如图8-10（b）所示。通过两个接口（DTE—DCE和DCE—DTE）的PPDU对有不同的名称，但它们在两个接口的语法是相同的。呼叫请求的语法（结构）和入呼叫相同，以此类推。

(a)

包（PPDU）类型		
DTE→DCE	DCE→DTE	协议用法
呼叫请求 呼叫接受	入呼叫 呼叫证实	呼叫建立
清除请求 DTE清除证实	清除指示 DCE清除证实	呼叫清除
DTE数据 中断请求	DCE数据 中断证实	数据传送
DTE接收方准备好 DTE接收方未准备好 DTE拒绝 复位请求 DTE复位证实	DCE接收方准备好 DCE接收方未准备好 复位指示 DCE复位证实	流量控制
重新启动请求 DTE重新启动证实	重新启动指示 DCE重新启动证实	重新同步
诊断	诊断	网络差错报告

图8-10 PPDU类型

(a) 用法 (b) 格式

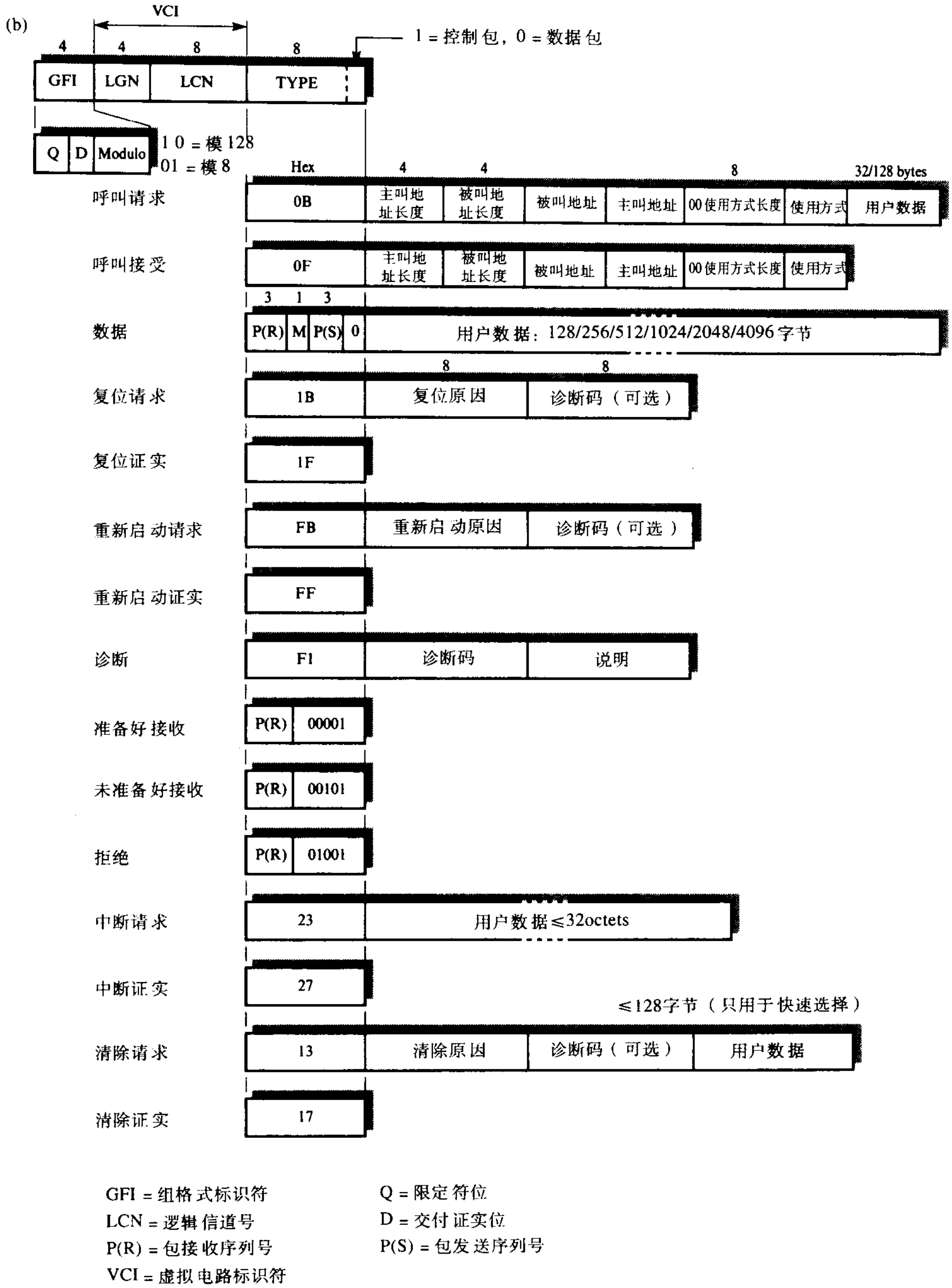


图8-10 （续）

所有的PPDU（包）都有一个固定的包头，它由**群格式标识符**（GFI）、**逻辑群号**（LGN）和**逻辑信道号**（LCN）组成。GFI是4位字段，由数据限定符（Q位）、交付证实（D位）和两个附加**模位**组成。在本节稍后会讨论Q位和D位的用法。模位指明用于流量控制目的的（包）序列号的模（值）——8或128。LGN和LCN共同组成了12位VCI。

下一个字节是**包类型**字段。控制PPDU的最低位是1而数据PPDU是0。最低位称为**控制位**。

数据PPDU和三种流量控制PPDU——准备接收（RR），未准备接收（RNR）和拒绝（REJ）——
436~437 每种PPDU的类型字段中都有一个接收序列号。数据PPDU还有一个发送序列号和一个**更多数据**（M）位。在图8-12中，考察三种流量控制PPDU与M位的用法。数据PPDU中的最大用户数据量是128个字节，虽然一些公共载波使用达到4096个字节的更长长度。中断请求PPDU在正常流量控制机制外还允许网络服务用户（NS_user）发送长达32个字节的控制数据，这样PPDU用中断证实来确认。

当呼叫正在建立时，呼叫请求和呼叫接受PPDU中的机制字段允许协商选定的工作参数。这些参数包括快速选择的使用、扩展序列号、可替换窗口、包长度、反向计费以及其他。最后，**诊断**PPDU由网络用来通知用户DTE可能检测到的差错情况，这包括无效发送和接收序号、接收到的无效包类型、呼叫建立问题以及其他。

（1）虚拟呼叫建立和清除

说明虚拟呼叫各个阶段的时序图如图8-11所示。当用户在用户服务访问点（SAP）发出一个N_CONNECT.request原语时，一条VC被建立。与这个原语相关的参数包括被叫DTE的NSAP地址和有限的用户数据。可以看到，建立网络连接有两种可选规程。

第一种（**正常**）方式中，接到N_CONNECT.request原语就建立一条X.25 VC，然后在这条电路上使用一个X.25数据包传递网络连接请求（N_CR）。

采用这种方式开销很大，因此，引入了另一种方式（**快速选择**）。在这种方式中，网络连接请求直接映射到X.25呼叫请求包上，大大减少了呼叫建立开销。重置和断开连接服务以类似的方式映射，如图8-11(a)（i）所示。快速选择的第二个应用是提供有限的**数据报服务**，如图8-11(a)（ii）所示。

虽然面向连接的服务适合传输涉及大量数据的应用，但是多数应用只涉及单个请求/响应报文的交换。例如信用卡授权交易，它只简单地涉及信用卡号的传输以及随后的短响应报文。没有必要为这种传输建立VC，在支持快速选择的网络中，能使用单一报文交换执行这种类型的传送，如图所示。

使用快速选择，呼叫请求和呼叫接受包以及清除请求和清除接受包都包括一个128字节的用户数据字段。接到从流入N_CONNECT.indication中得到的用户数据，目标只是简单地以N_DISCONNECT.request响应，响应报文放在NS_user数据参数中。然后NS_data放在清除请求/证实包中返回发送方，同时清除VC。

说明VCI（逻辑信道）使用的时序图如图8-11(b)所示，假定是快速选择方式。接到
438 N_CONNECT.request原语，源协议实体先选择下一个空闲VCI，并产生一个含有主叫和被叫DTE地址以及选定VCI的呼叫请求包（PPDU）。然后把这个包传递给链路层，再转发给它的本地PSE。

接到这个包，本地PSE注明选定的VCI，并根据网络内部协议转发这个包到相应的目标PSE。PSE选择到被叫DTE链路上的下一个空闲VCI，把它写入包中并改变流入呼叫包的包类
439 型。然后它被转发给被叫DTE，相应的网络协议实体使用入呼叫包的内容产生一个N_CONNECT.indication原语，该原语被传递给相应的用户。

假定对应用户准备接受呼叫，则返回一个N_CONNECT.response原语响应，反过来网络协议实体产生一个呼叫接受包。指定该包与相应的入呼叫包使用的VCI相同。然后这个呼叫接受包转发给被叫DTE的本地PSE，并且这条链路上保留的逻辑信道进入数据传输阶段。类似地，源PSE接到呼叫接受包，把先前保留的用在这部分电路上的VCI插入包中，并且把这条逻辑信

道设成数据传输状态。它把这个包转换为呼叫连接包，并且转发给主叫DTE。最后，主叫包协议实体接到这个包，发起一个N_CONNECT.confirmation原语给用户，并进入数据传输状态。

如果相应的用户不想或者不能接受入呼叫，它以N_DISCONNECT.request原语作为对N_CONNECT.indication原语的响应。这样被叫网络协议实体返回一个清除请求包给它的本地 440
PSE。本地PSE先释放先前保留的VCI并返回一个清除证实包给被叫DTE。然后本地PSE发送一个清除请求包给源PSE，源PSE反过来把该包以清除指示包的形式传递给主叫DTE中的网络协议实体。DTE先释放保留的VCI，然后传递一个N_DISCONNECT.indication原语给用户。然后该DTE返回一个清除证实包给它的本地PSE来完成VC的清除。类似地，在任何时候，用户双方都可以通过在相应的用户接口发出一个N_DISCONNECT.request原语发起呼叫的清除。

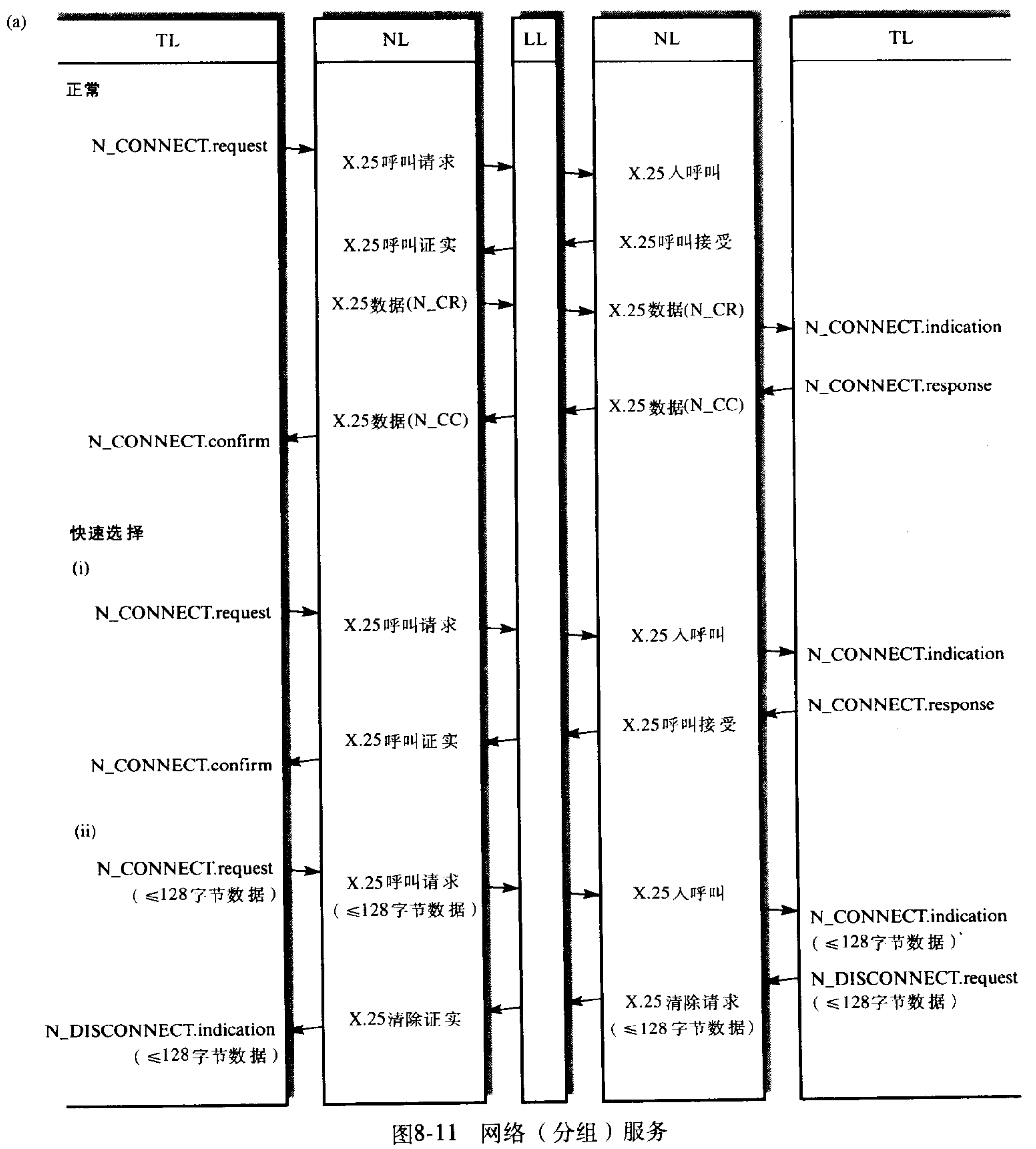


图8-11 网络（分组）服务

(a) 可选映射方式 (b) VCI的使用实例

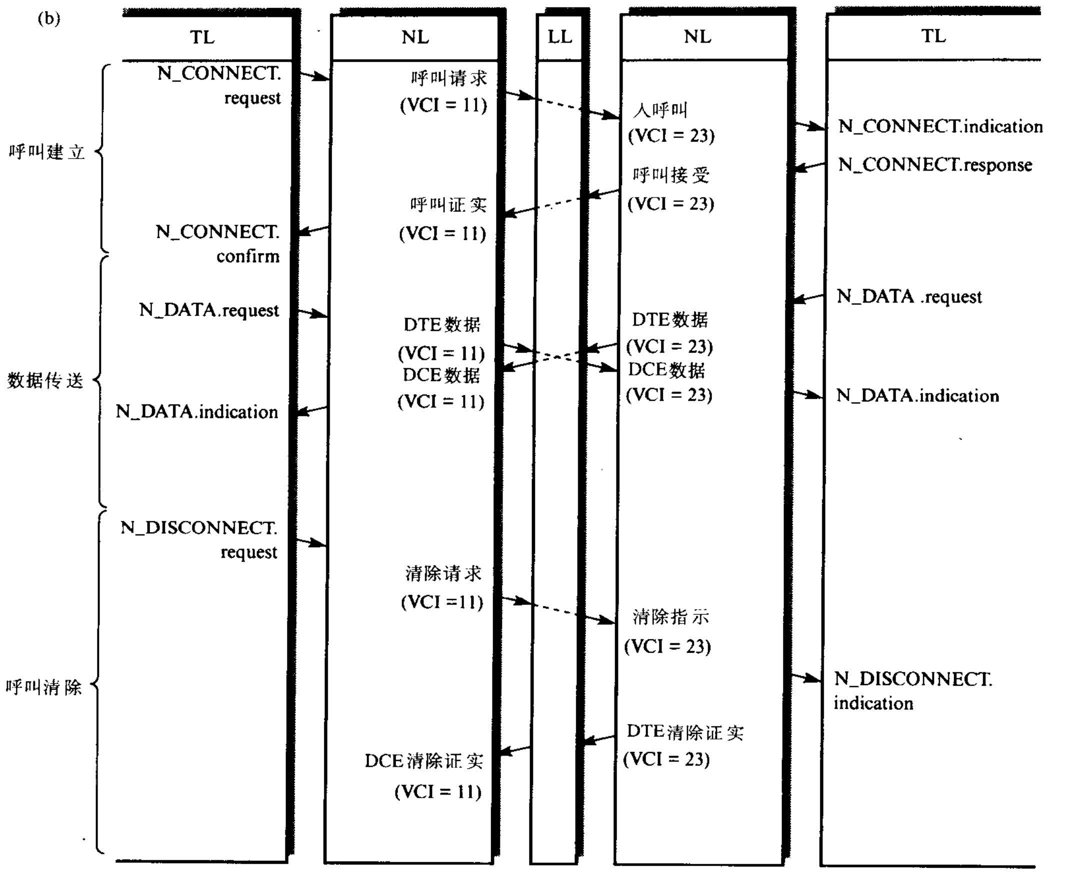


图8-11 （续）

（2）数据传输

当一个虚拟呼叫（网络逻辑连接）被建立后，用户双方均独立发起数据传输。用户可在网络接口处发出N_DATA.request原语，并采用要传输的数据作为参数以实现这一过程。已经提到过，在公共载波分组交换网络中的每个数据包长度都是有限制的，一般为128个字节，以确保可靠的快速响应时间。如果用户想传输含有多于这个字节数的报文，该报文先要被分成适当个数的数据包并且每个包分别发送。每个通过网络发送的数据包都在头部含有一个称为**更多数据位**或**M位**的位，它在需要更多的数据包来完成用户级（就是说，传输层）报文传输时被设置，这样接收方用户就能知道何时完整报文接收完毕。

虽然传输层通常在传输自己的协议控制报文（TPDU）给一个或多个对等传输层时，采用N_DATA.request原语并将TPDU作为数据参数，但是X.25分组层还允许用户指定相关参数是否含有用户级控制或数据信息。信息类型由分组层嵌入到生成的数据包。它设置包头部一个称为**限定符位**或**Q位**的特殊位。目标分组层收到每个数据包，把这个相关数据信息传递给相应用户。

虽然与X.25协议相关的三个协议层通常只有本地意义，但是通过在每个包头部使用称为**交付证实位**或**D位**的特殊位，分组级的确认信息给出了端对端的意义。如果源DTE需要远端对等分组层正确接收的端对端确认信息，数据包头部的D位设为1。将在8.3节看到，这个信息放在反方向流动的包头部。

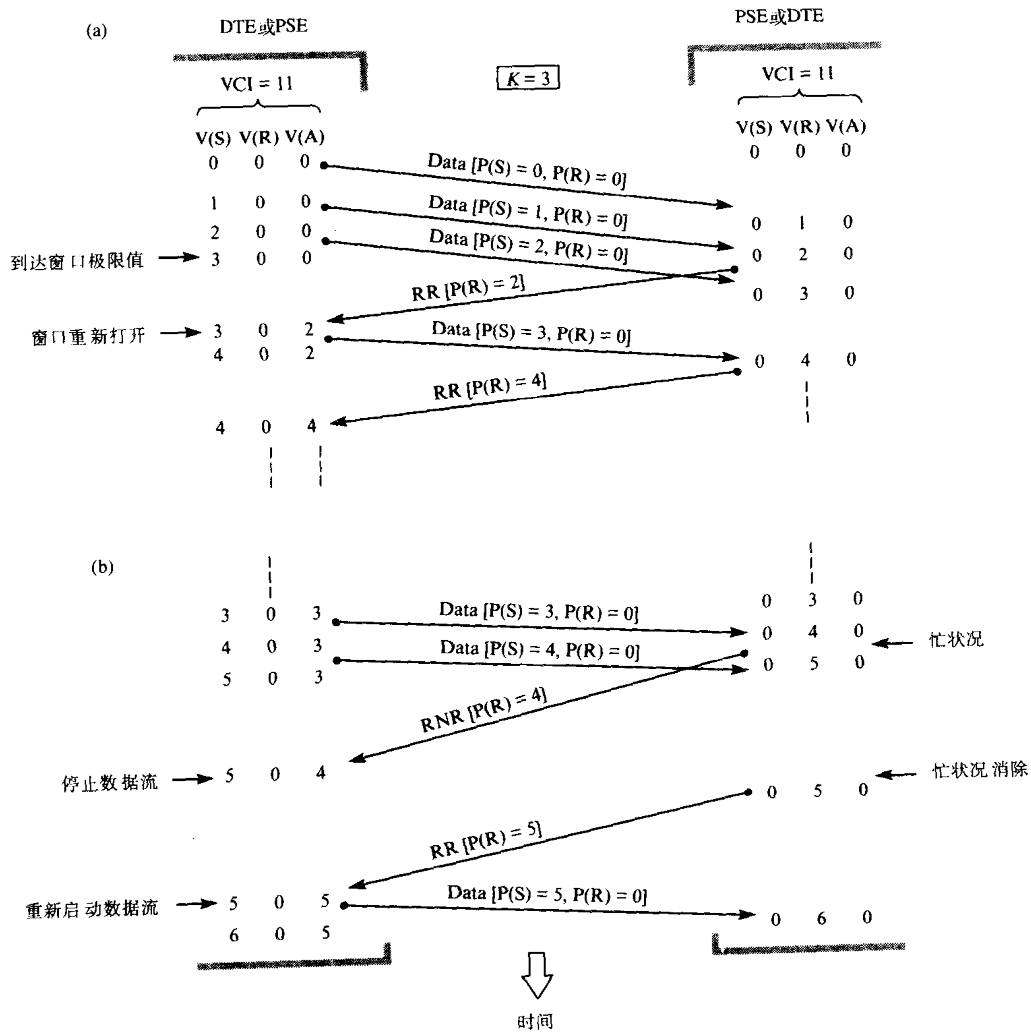


图8-12 流量控制实例

(a) 窗口操作 (b) RNR操作

（3）流量控制

所有分组层的包使用链路层提供的服务，从DTE传输到它的本地PSE。链路层HDLC协议的使用意味着基本包传输方式相对可靠。这样，分组层的重点在流量控制而不是差错控制。流量控制算法基于类似于第4章介绍的滑动窗口机制。分别控制一个呼叫的每个逻辑信道和每个方向上的包流，就是说关于每个呼叫的从DTE到PSE的数据包流和从PSE到DTE的数据包流被分别控制。 441

为了实现窗口机制，所有数据包含有**发送序号P（S）**和**接收序号P（R）**。每个数据包中的P（R）涉及反方向上的数据包流。另一种情况，如果在反方向上没有数据包等待传输，P（R）可能由接收方放在特殊**接收方准备**（RR）监管包中发送。

一个逻辑信道的每个方向上（DTE到PSE和PSE到DTE）的第一个数据包给出为0的P（S），同一方向上的后继包含有的P（S）依次加1。在收到响应前，每个方向上发送的与同一呼叫相关的包个数受到信道窗口大小K限制，由于第4章所描述的原因，如果使用8个惟一的序号K就有最大值7。这样，一旦发送方发起达到窗口大小的数据包的传输，它必须停止发送包直到它

接收到包含P（R）的数据包或者含有指明接收方愿意在这个信道上接收更多包的RR监管包。

为了实现这个方案，每个DTE和PSE为每个活动逻辑信道（VC）保留三个变量：

- V（S）　**发送序列变量**用来指明在这个逻辑信道上发送的下一个数据包的P（S）。
- V（R）　**接收序列变量**用来指明这个逻辑信道上期望接收的序列中下一个数据包的P（S）。
- V（A）　**确认变量**用来决定何时应该停止数据包流。

所有三个变量在VC刚建立时设为0，或者随后被复位（见后面的“差错恢复”）。当每个数据包准备发送时，就给它指定一个发送序号P（S），其值等于当前V（S），然后以GFI字段定义的模8或128递增。类似地，接到来自目标分组层的每个数据包或RR流量控制包，接收序号P（R）用来更新V（A）。发送方继续发送数据包直到达到窗口大小（就是说递增V（A）达到*K*），或者收到含有预设当前V（A）的P（R）的RR包。然后更多的数据可以发送直到又达到窗口限制。图8-12(a)给出的一个典型包序列说明了窗口大小为3的这个规程。为了简便起见，假设只有单一逻辑通道，而且仅有单方向数据流。

442 控制数据包流的窗口机制的使用意味着处理每个呼叫所需的包缓冲区最大个数可以轻易得到。实际上，提供满足当前可能有效的所有呼叫的缓冲区总数量小于所需的最大数量。因此，在协议中提供一种方式，使DTE（或PSE）能暂时挂起与特定呼叫（虚拟电路）相关的数据包流，这由接收方返回一个**接收方未准备好**（RNR）包，而不是RR包来实现。每个RNR包含有为信道定义新V（A）的P（R）。但是，发送方接到RNR必须停止更多包的发送直到接收方准备在这个信道上继续接收数据包。接收方通过返回RR包来指示这个信息。图8-12(b)的典型包
443 序列说明了RNR包的使用情况。我们可以从图中推断出RNR包不能立即停止包流，因为一些包可能正在链路上传输。但是，因为缺少任何有关分组层的差错控制，以这种方式收到的任何包必须被接受。

虽然刚才描述的两种机制提供了对每个逻辑信道上的数据包流量的控制，但还在协议中作了有关规定，允许DTE独立于正常流量控制规程发送单个高优先级数据包（中断包）给相应DTE。因为中断包不受正常流量控制机制的影响，它可以在该电路上乱序接收其他数据包。接收DTE（网络层）接到中断包必须返回中断证实包，因为在任何时刻每条VC上只能有一个待处理未确认中断包。这允许在需要情况下再发送一个该包，如图8-13(a)所示。

（4）差错恢复

分组层的主要差错恢复机制是复位和重新启动规程。复位规程只用在数据传输阶段，并且只
444 影响一个虚拟呼叫（一条虚拟电路)。但是，重新启动规程影响当前正在进行的所有虚拟呼叫。

如果某DTE收到超出当前窗口限制的数据包就发送一个复位请求包。它表示两个DTE不同步了，因此数据包流必须重新启动。关于复位规程的典型包序列如图8-13(b)所示。任何有关受影响VC的数据包被分组层丢弃，并且通知用户网络连接已被清除。清除的理由作为参数被传递，然后由用户（实际上是传输层）对任何可能的数据丢失进行恢复。

重新启动规程用来同时清除一个DTE上当前所有的VC。当DTE和PSE不同步以致影响当前所有活动呼叫时使用该规程。例如如果收到的来自PSE的呼叫使用了当前被使用的LCN。重新启动规程以及会对若干活动VC产生影响的典型包序列如图8-13(c)所示。该图不仅说明了在单一的相应分组层上的可能影响，而且指明许多其他DTE也会受到类似的影响。

4. 分组层概要

图8-14总结了分组层的整个操作。我们已用了相同的形式来说明链路层。它清楚地规定分组层操作的三个方面：

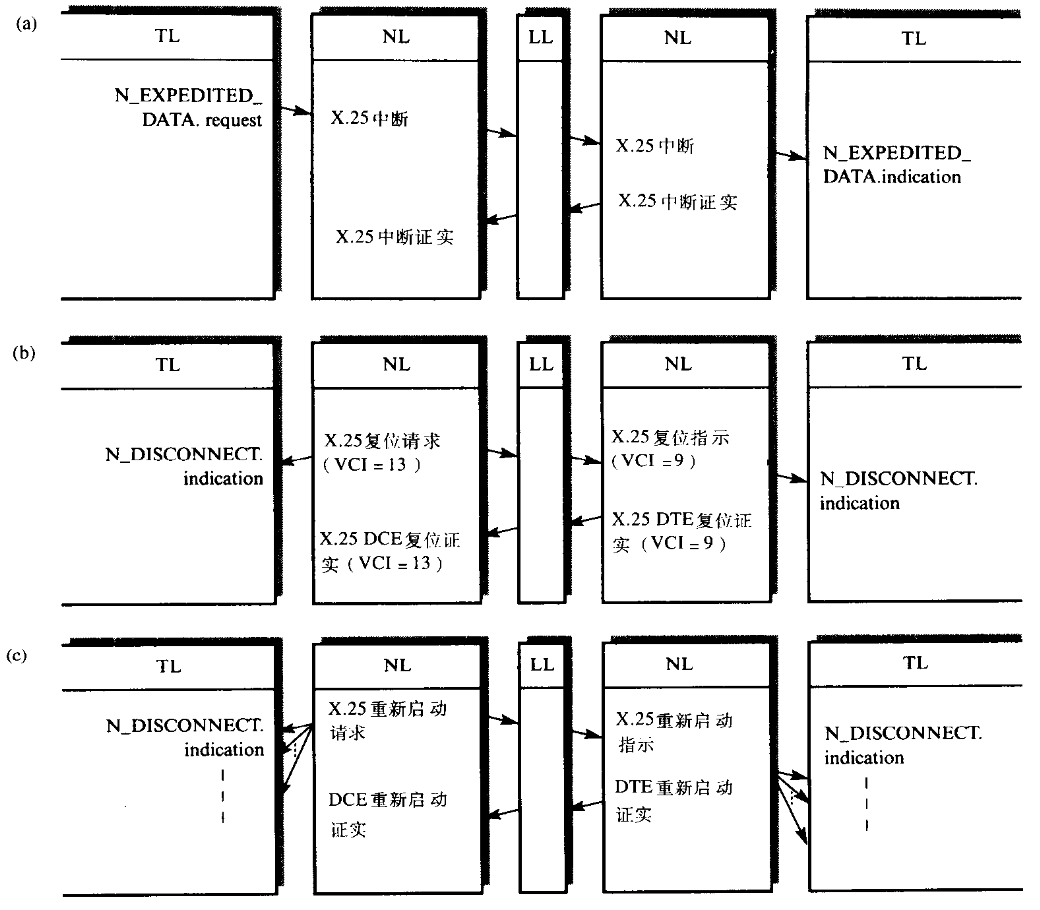


图8-13 附加服务

(a) 加速数据 (b) 复位 (c) 重新启动

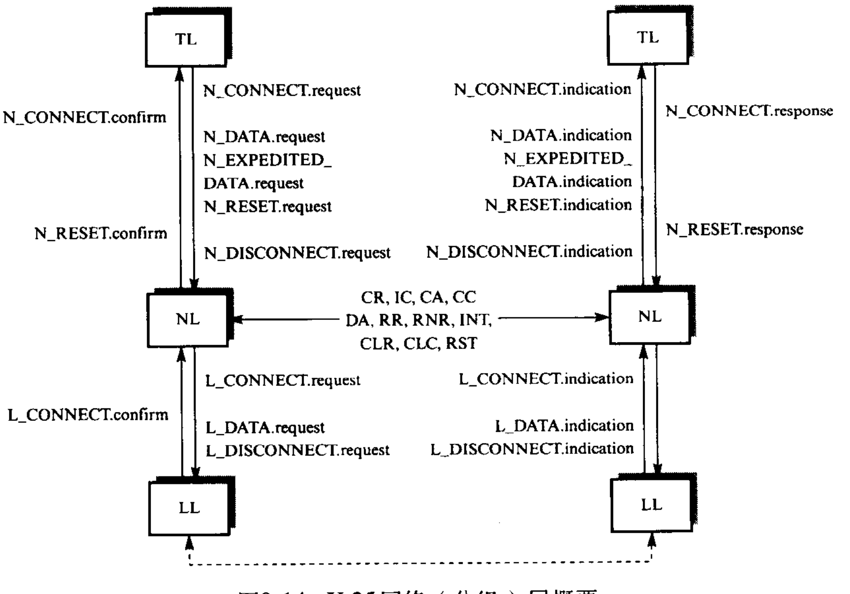


图8-14 X.25网络（分组）层概要

445 • 它给上层传输层提供服务；
• 两个对应网络（分组）层协议实体间交换的PDU；
• 用链路层服务传输这些PDU。

5. 协议规范

现在通过考虑分组层协议的呼叫建立阶段，来了解分组层协议。这个阶段的时序图如图8-11所示。为了清楚起见，我们假定使用快速选择方式并且数据链路已经建立。

为了保持第4章中使用的方式，图8-15列出了所有与协议的呼叫建立阶段相关的入事件、状态、出事件、谓词和特殊动作。为了帮助理解，呼叫建立阶段的状态迁移图如图8-16(a)所示。注意入事件和相关出事件表示在每个变迁弧线旁。图8-16(b)以事件—状态表的形式给出了更正式的定义。

(a)

名　称	接　口	含　义
NCONreq	NS_user	接收到N_CONNECT.request
NCONresp	NS_user	接收到N_CONNECT.response
CALLconn	Link layer	接收呼叫连接包
INCcall	Link layer	接收入呼叫包
TCALLconn	计时器	呼叫连接计时器到时
⋮		

(b)

名　称	含　义
IDLE	没有建立连接
WFCC	等待呼叫连接包
	等待来自NS_user的N_CONNECT.response
WFNCR	等待清除证实包
WFCLCF	连接建立并准备传输数据
DATA	
⋮	

(c)

名　称	接　口	含　义
NCONind	NS_user	发送N_CONNECT.indication
NCONconf	NS_user	发送N_CONNECT.confirm
NDISind	NS_user	发送N_DISCONNECT.indication
CALLreq	链路层	发送呼叫请求包
CALLacc	链路层	发送呼叫接受包
CLRreq	链路层	发送清除请求包
⋮		

(d)

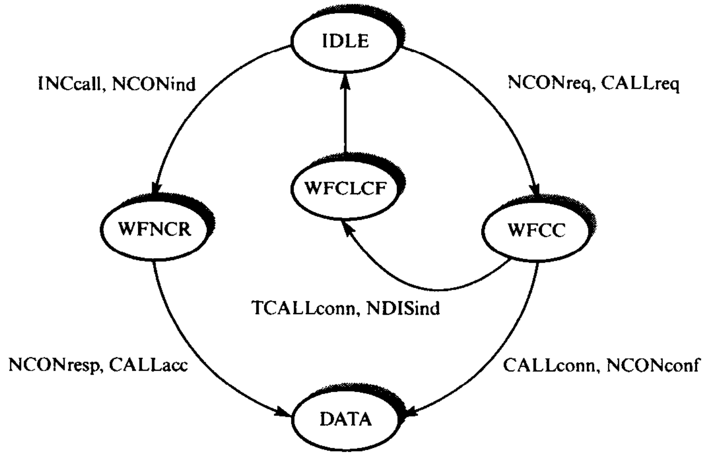
名　称	含　义
P0	无法接受来自NS_user的N_CONNECT.request
P1	无法接受来自NS_user的N_CONNECT.response
⋮	

(e)

名　称	含　义
[1]	启动TCALLconn计时器
[2]	停止TCALLconn计时器
⋮	

图8-15　网络（分组）层呼叫建立阶段的缩写名称
(a) 入事件　(b) 状态　(c) 出事件　(d) 谓词　(e) 特定动作

(a)



(b)

事件＼状态	IDLE	WFCC	WFNCR	WFCLCF	DATA	---
NCONreq	1	0	0	0	0	
NCONresp	0	0	3	0	0	
CALLconn	0	2	0	0	0	
INCcall	4	0	0	0	0	
TCALLconn	0	5	0	0	0	
⋮						

0 = NDISind, CLRreq, WFCLCF （差错条件）

1 = P0: NDISind, IDLE;
NOT P0: CALLreq, [1], WFCC

2 = NCONconf, [2], DATA

3 = NOT P1: CALLacc, DATA

4 = NCONind, WFNCR

5 = NDISind, CLRreq, WFCLCF

图8-16 用于呼叫建立的协议规范

(a) 状态变迁图 (b) 事件—状态表

注意没打算全部给出这些定义，而只是当作理解X.25分组层协议正式规范的介绍。 446～447

8.2.4 终端访问

在前面有关X.25网络访问协议的小节，假定连到网络上的DTE有足够的智能（或者处理能力）来实现刚才描述的各种协议层。通常，如果DTE是计算机，这个假设当然是成立的。但是，在一些情况中，DTE既不以分组方式工作也没有足够的处理能力实现像X.25的协议。为了把这种DTE接入网络我们必须提供额外的装置来代替DTE实现各种协议层以及提供给DTE更简单的用户级接口。这种DTE的例子是像个人计算机或VDU的简单异步字符方式终端。它一般只带有简单EIA-232D/V.24物理接口的有限智能水平。

为了满足这种要求，用户可以选择必要的附加装置，把终端字符串组装成网络包，并把网络包拆解成终端字符串。另一方面，因为这是一个平常的要求，各个PSPDN机构提供给用户另一种网络访问协议，称为X.28，用于异步字符方式终端。需提供这种接口的附加装置称为**包装卸设备**（PAD）。因为PAD由PSPDN提供，它通常位于本地PSE中。单个PAD可以用来支持若干字符方式DTE。PAD的功能和位置以及为使用它而定义的附加协议如图8-17所示。

可以看到，协议X.3定义了PAD提供的操作和使用方式，而X.29定义了PAD和远端分组方式DTE间的接口。现在考虑这种工作方式的某些方面。

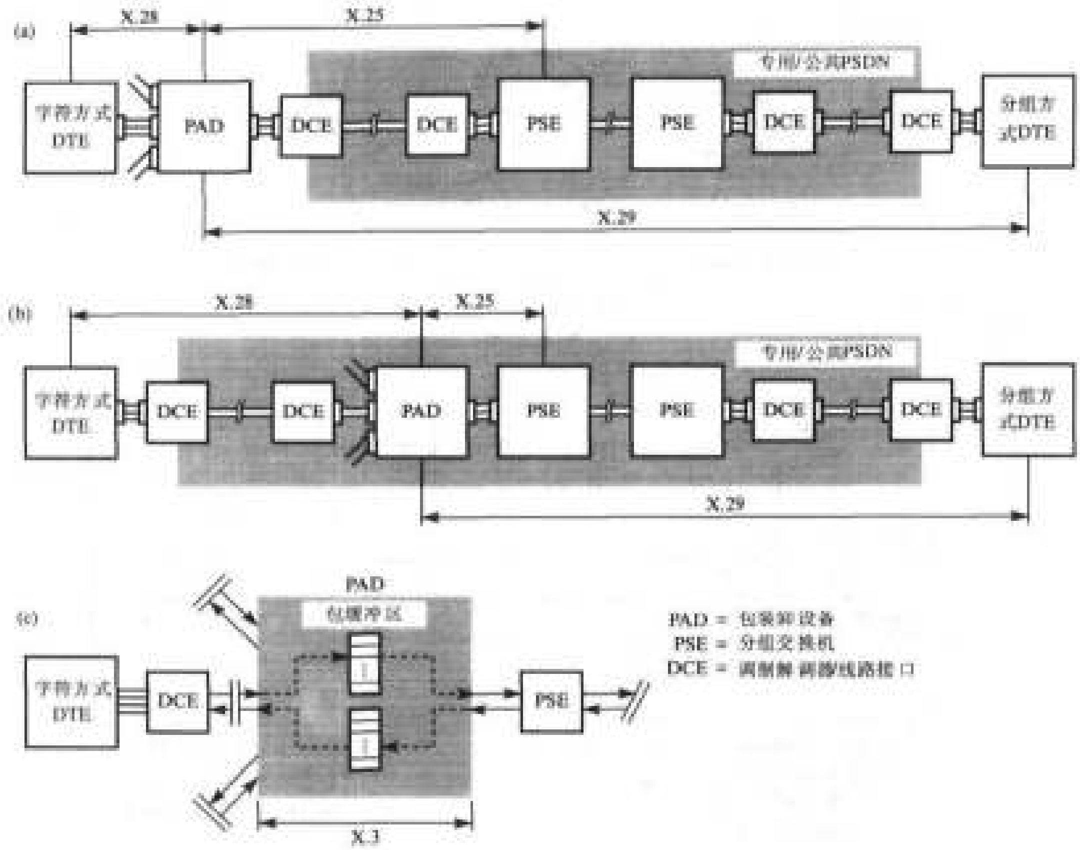


图8-17　PAD位置及其协议

(a) 用于字符方式DTE　(b) 用于PSE　(c) 内部示意图

1. PAD和X.3

基本上，PAD的功能是把字符方式异步终端的用户输入的单个字符组装成适合在X.25 PSPDN上传输的有意义的分组。类似地，PAD收到这种分组，把它拆卸并在某个时间逐字符地传递给字符方式终端。这样PAD必须代表终端执行所有X.25协议功能（比如呼叫建立、流量控制），并且通常使网络的分组工作方式对用户透明。

X.3建议定义了PAD的功能和使用方式。除了上述的基本功能之外，每个连到PAD的终端有若干相关参数，因为字符方式终端在操作和特性上变化很大。这些参数通常由终端或者接
448 人的远端分组方式DTE输入命令设置，并涉及如下的特性：

- 是否需要本地回显校验；
- 选择分组端接（数据转发）字符，它允许终端用户发信号通知PAD应该发起（部分完整）包的传输；
- 其他可能的控制字符的说明，如换行与回车等功能。

为了方便PAD的使用，所有有关终端的参数都有**默认值**。只有与默认值不同的参数才需要改变。初始参数环境由为设备选择的**标准说明文件**确定。已经为较流行终端定义了许多可选的标准说明文件。当终端和PAD之间的通信链路刚建立时，通常选择并输入标准说明文件及其变化。在X.28建议中定义了该规程。 449

2. X.28建议

该建议详细说明了用在异步字符方式终端和PAD间的协议。它包括如下的规程：

- 访问PAD
- 把终端参数设成所需值
- 建立到目标分组方式DTE的虚拟呼叫
- 控制终端和PAD间用户数据交换
- 清除建立的呼叫

访问PAD可以有几种形式。可以使用由PSTN建立的交换连接，或者租用线路。如果PSTN使用模拟传输，在链路两端必须使用调制解调器。如果网络提供数字数据服务，可以建立或租用直接的数字通路，并且可以使用传统的EIA-232D/V.24接口。

一旦终端得到对PAD的访问权，就发送服务请求字符序列。这使得PAD能确定终端所用的数据传输率，并允许终端选择初始标准说明文件。标准定义的规程允许终端用户读取有关说明文件的参数，如果需要可以把它们改成其他值。然后PAD准备通过PSPDN建立和远端分组方式DTE的虚拟呼叫。

为了建立虚拟呼叫，用户先向PAD指明所需分组方式终端的地址。PAD遵循前述的虚拟呼叫建立规程。一旦呼叫被建立，PAD进入数据传输阶段。

在数据传输阶段，PAD执行必要的包装配和拆卸功能。在装配过程中，当用户输入一致的包端接控制字符或者超时后，PAD启动包的传输。最后，在所有信息交换完后，用户可以请求PAD启动呼叫的清除。

3. X.29建议

该建议详细说明了PAD和远端分组方式DTE间的交互。X.29涉及的呼叫建立和数据传输规程基本上同X.25一样。但是，在该建议中定义的附加规程反映了终端和远端分组方式DTE间PAD的存在。例如，在呼叫建立阶段，PAD使用呼叫请求包中的可选用户数据字段的前4个字节作为所谓的**协议标识符**字段。不同类型的主叫用户（终端）可以（使用协议）被识别，这样被叫分组方式DTE（如果需要）能使用可替换的协议。 450

类似地，在反方向的数据传输阶段，分组方式DTE能直接使用每个数据包头部的Q位与PAD通信。当Q位是“1”时，则包中保留供PAD使用的控制信息，不应该被拆卸传给用户终端。例如，这个规程允许远端分组方式DTE读取以及（如果需要）设置主叫终端参数的当前值。

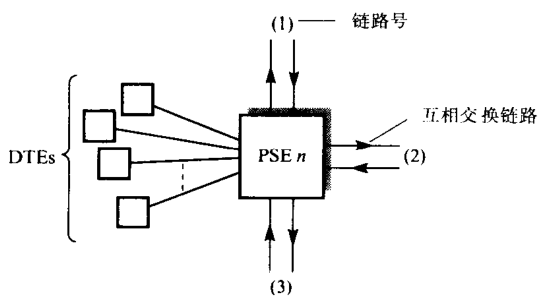
8.2.5 X.25网络的互连

已经描述过X.25是用来说明DTE到与PSDN相关DCE的接口的协议集。它是面向连接的协议，这意味着在传输数据前两个通信DTE间要建立VC。与VC相关的是两个逻辑信道标识符：一个用于连接主叫DTE到它的本地DCE/PSE的链路，而另一个用于被叫DCE—DTE的链路。这些标识符用来把与呼叫相关的后继数据包联系到那条VC上去。

虽然它不是X.25标准的一部分，但是当建立VC时，它也必须在网络中建立，使得与每个

呼叫相关的数据包能通过网络内的分组交换机被路由。将讨论一些用于第9章WAN的路由选择算法。但是，与计算路由的路由选择算法无关，每个交换机都有若干张用于路由包的路由选择表：一张**网络路由选择表**，指明到网络中每个目标DTE所需的交换机的出链路，以及一组附加的**链路路由选择表**，每条链路有一张表。

收到呼叫请求包，这条路线的交换机先使用包中的目标DTE地址从网络路由选择表中确定使用哪条出链路来转发这个包。然后它从空闲VCI列表中获得该链路下一个空闲VCI，并且在两个链路路由选择表中加入记录。实例如图8-18所示。



PSE *n* 网络路由选择表

Dest.DTE	链路出口
⋮	⋮
25	2
⋮	⋮

注意：连到该PSE的DTE有0条记录。

1号链路路由选择表

VCI入口	链路出口	VCI出口
	⋮	⋮
10	2	15
	⋮	⋮
15	3	8

2号链路路由选择表

VCI入口	链路出口	VCI出口
	⋮	⋮
15	1	10
	⋮	⋮

3号链路路由选择表

VCI入口	链路出口	VCI出口
8	1	15

图8-18　网内路由选择实例

我们假定VCI是10的呼叫请求包到达1号链路，并且目标DTE地址是25。交换机先从网络路由选择表确定所需出链路是2号链路。假定2号链路上的下一个空闲VCI是15，交换机先在1号链路路由选择表位置（VCI）10加入相应出链路号（链路出口）和VCI（VCI出口）的记录，然后在2号链路路由选择表位置15加入1号链路和VCI 10的记录。然后它在2号链路启动头部含VCI为15的呼叫请求包的转发。

这个规程被每个交换机重复，直到呼叫请求包到达目标DTE连接的交换机。随后的呼叫接受包和相关数据包沿着这条建立的路由（VC）转发，因为每个交换机简单地从入链路的路由选择表中读取出链路号和VCI，把它们写入包头部并在出链路上启动包的转发。
451

可以推断出，如果整个网络由多个互连网络组成，并且两个DTE连在不同的网络上，那么一条单独的VC必须越过每个中间子网建立。而且，为了把包与指定呼叫联系，VC还必须通过互连这些网路的链路建立。这样在由多个子网组成的大型网络中，两个DTE间建立的VC是由若干单独VC组成（逻辑信道标识符），每条VC用于特定链路。如图8-19所示。为了简单

起见，每个子网只表示了一条VC。

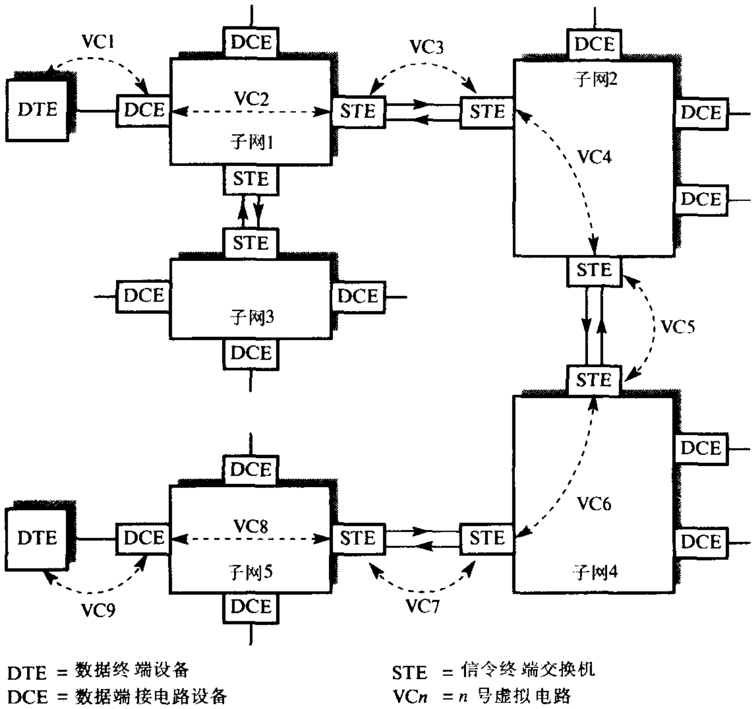


图8-19 通过由多个子网组成的网络的VC结构

图中表示的单个网络是公共载波数据网，这样成千上万的DTE可能连在每个网上。网络间的互连链路通常是多条64kbps或56kbps线路或信道。用于互连每个网络的设备是称为**信令终端交换机**（STE）的特殊DCE。X.75协议用来在这些链路上建立和释放VC。这个协议也经常用来在单个网络上建立和释放VC，虽然这没有被ITU-T规定。图8-19表示了两个DTE间与单个呼叫相关的VC。

一个呼叫存在多条VC对于两个通信DTE是透明的，每个DTE只知道到它的DCE/PSE的本地链路的VC标识。这些链路上所有数据包的流量控制和差错控制以正常方式执行。与其他每条VC（DCE—STE、STE—STE等）相关的控制设备以同样的方式操作，并在这些电路上实施自己的流量控制和差错控制规程。因为每对网络间的STE由两个等分部分组成，每部分与它自己的网络相关，因此STE也称为**半网关**。 452

X.75建议

X.75的适用性表示在图8-20中。如同X.25，X.75标准有三个相关协议：数据分组层协议（PLP）、数据链路层协议（DLP）和物理层协议（PHY）。通常，为了提高可靠性和吞吐量，我们使用多链路来互连两个STE，这样**多链路规程**（MLP）用于DLP。回忆一下第5章，使用MLP，当在链路联合集（或者涉及更高比特率电路的子信道）上传输帧时，这些链路被当作单个实体。当要传送帧（包）时，选择任何可用的链路，而不管其内部的VC（逻辑信道标识符）。每个帧内的多链路控制（MLC）字段，由接收方MLP用来在帧传送到PLP前重新排列它们。 453

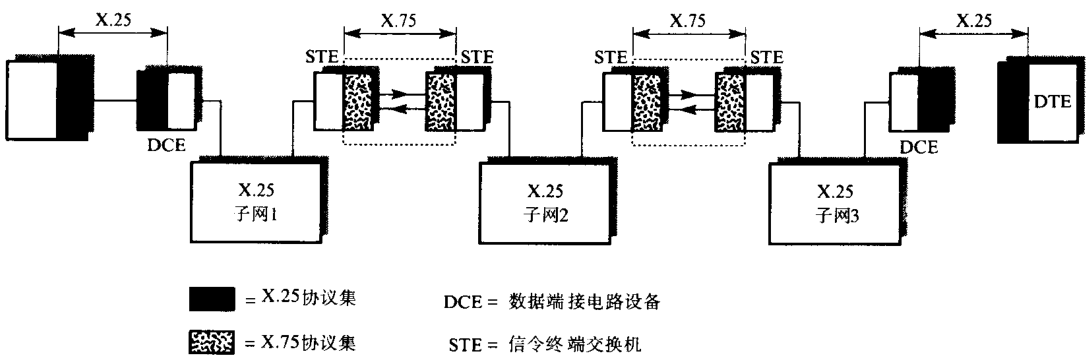


图8-20　X.75协议的适用性

X.75 PLP比X.25 PLP简单，实现它需要更少的包类型。这是因为在X.75中通信是STE之间的两个直接连接，而在X.25中两个DTE间的通信路径涉及两个中间DCE。这减少了包类型集，它们的一般格式分别如图8-21(a)和图8-21(b)所示。

各个字段及其用途实际上同X.25中的一样。主要的附加字段是每个呼叫请求包中的**网络杂项字段**（network utilities field）。回忆X.25中同样的包有个使用方式字段，它用来请求连接的某些操作的使用方式，诸如传输策略的选择。类似地，网络杂项字段允许一个STE向另一个STE指明它连接网络相关的参数。包括窗口和包长度指示，估算传输时延和快速选择支持。这个字段使得STE能确定连接所需的指定最小化使用方式是否满足。

454 PLP的操作如图8-22概述。(a）表示各种协议而（b）表示建立VC、然后传输一个数据包、最后清除电路的端对端包序列。为了方便，我们假定X.75协议用于通过每个子网和STE—STE链路传输。每个STE内的中继功能用来执行任何包格式转变。

(a)

包（PPDU）类型	协议用法
呼叫请求 呼叫连接	呼叫建立
清除请求 清除证实	呼叫清除
数据 中断 中断证实	数据传送
接收方准备好 接收方未准备好 复位证实	流量控制
重新启动 重新启动证实	重新启动

(b)

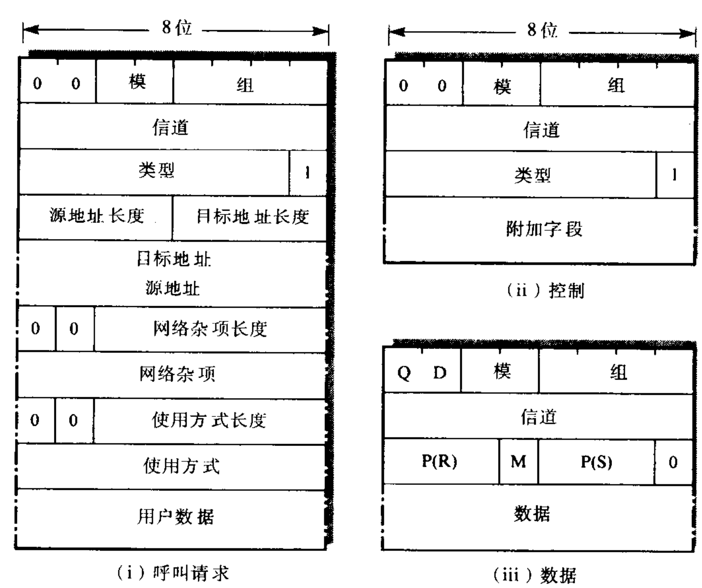


图8-21　X.75包

(a) 包类型及其用法　(b) 包格式

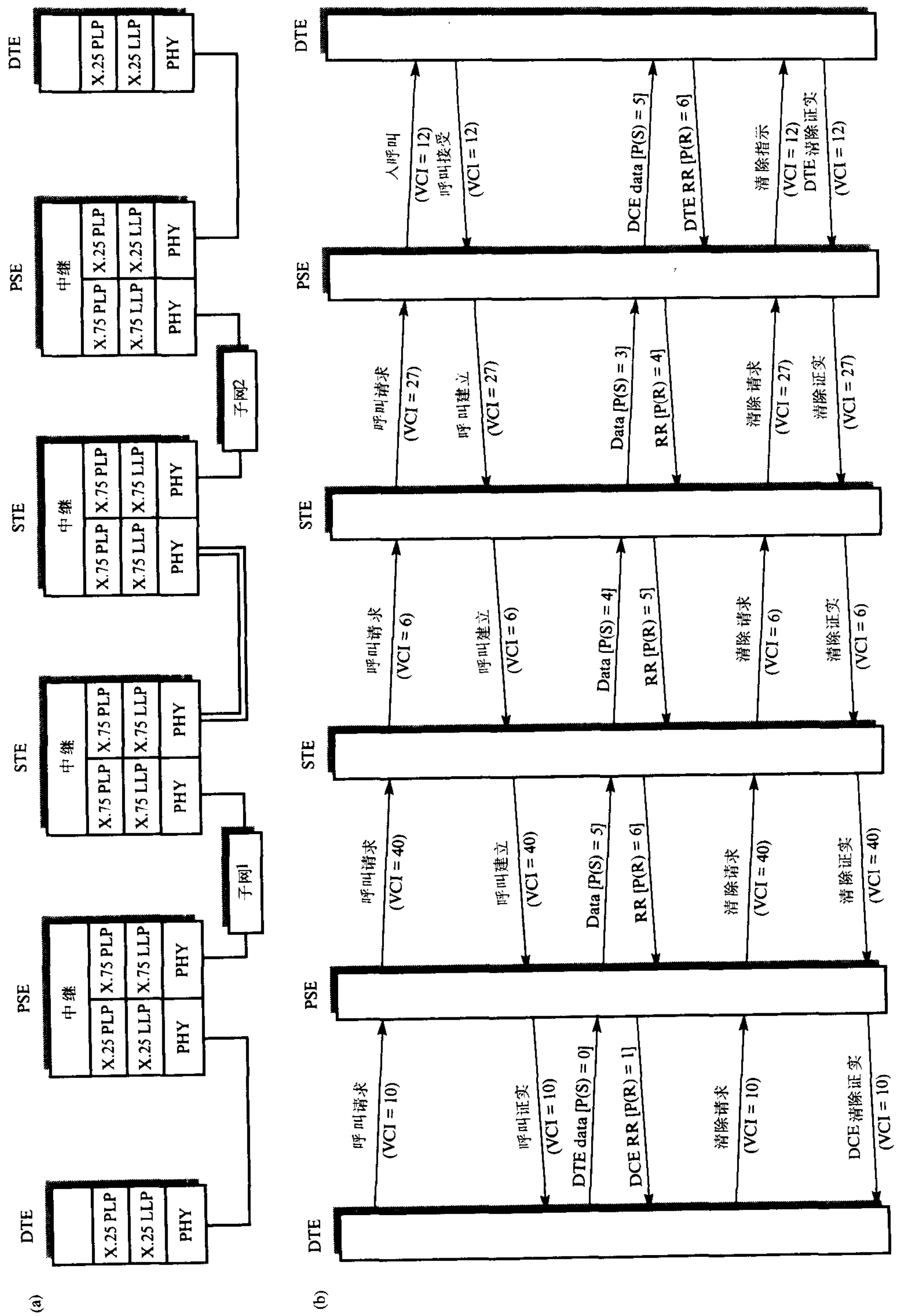


图8-22 端对端包流

(a) 协议栈 (b) 包序列图

每个STE有特定的X.121地址。因为该地址是分级的，所以接收STE（以及网络内的每个中间分组交换机）接收到包时，能使用目标地址的高位部分来确定包到下一个STE的路由或者到目标DCE/DTE的路由。

接到呼叫请求包，源PSE从包内的（DTE）目标地址判断出该地址在不同的网络。PSE使用它的路由选择表确定使用哪条出链路来转发这个包到相应的STE。STE内的中继层接到这个包，先建立该呼叫的相关VCI记录，然后用新的VCI和相应网络杂项参数重新格式化这个包。然后使用相应的协议栈发起通过STE—STE链路的包传输。

接收STE内的中继层建立包内的网络杂项及VCI的记录，输入新的VCI，再使用包内含有的DTE目标地址，来转发包通过第二个子网到目标PSE。目标PSE内的中继层重新格式化包成为入呼叫包，改变VCI，并使用X.25 PLP转发这个包给目标DTE。

类似的规程用于呼叫接受/呼叫连接包，这种情况中除了第二个网络相关的杂项都让其他STE都知道。最后，接到呼叫连接包，数据包由源DTE通过建立的VC传输。

如图8-12和图8-13所示，正常X.25流量控制和差错控制规程依次应用到每条VC。这样如果流量在任何网络点上被停止，它依次会把反应返回给邻接网VC，最后到源DTE。这种类型的流量控制称为**回压**。

图8-22（b）表示如何使用每条VC的单独证实来传输单个数据包。另一种情况，如果在数据包中设置了D位，那么它在整个网络传输过程中保留该设置。由此产生的确认以相同方式处理，并由此具有了端对端意义。最后，呼叫清除规程同X.25中的类似，除了它能扩展到全部网络。

8.2.6 LAN上的X.25 PLP

虽然连到许多LAN的DTE（站）以最佳尝试无连接网络层协议工作，但它们还以X.25
455～456 PLP操作。例如当连到LAN的DTE（站）需要通过基于X.25的WAN和远端系统通信。那么所有网络都是相同的类型，互连问题大大简化。这种类型的一个简单互连网络如图8-23所示。

在这个实例中，我们假定整个网络由连到基于X.25的WAN的每个分布式LAN群体组成。它的目标是连到每个LAN的所有站能与同一个LAN上的另一个站直接通信，或者访问连到X.25 WAN上的大型机。

为了满足这种要求，每个LAN有一个特殊结点充当X.25 WAN的接口点。由于它扮演的角色，我们称之为（ITU-T术语）**交互单元**（IWU）。它有已知的X.25WAN地址。如果站中的X.25 PLP确定所需的目标网络地址不同于它自己的地址，那么它使用IWU（已知的）MAC地址发送呼叫请求包给IWU。然后IWU的中继功能在X.25 WAN中以正常方式发送一个单独呼叫请求包。

这种交互类型的两个问题是VC（逻辑信道）标识符的选择和帧（包）长度的不一致。显然，与LAN相关的IWU基本上有若干连到它上面的独立DTE，每一个DTE能发起呼叫请求包
457 的发送。为了避免VCI选择冲突的（高）概率，一个（或多个）标识符预先指定给每个DTE，DTE能识别它们。在不同包长度的情况下，正常方法是为所有参与WAN的通信采用最大X.25 WAN包长度。

LLC子层可能是LLC1或者LLC2，两者都在5.3.6节中描述过。因为经常使用这种网络，在LAN上的X.25 PLP的使用现在已在ISO 8881定义为国际标准。

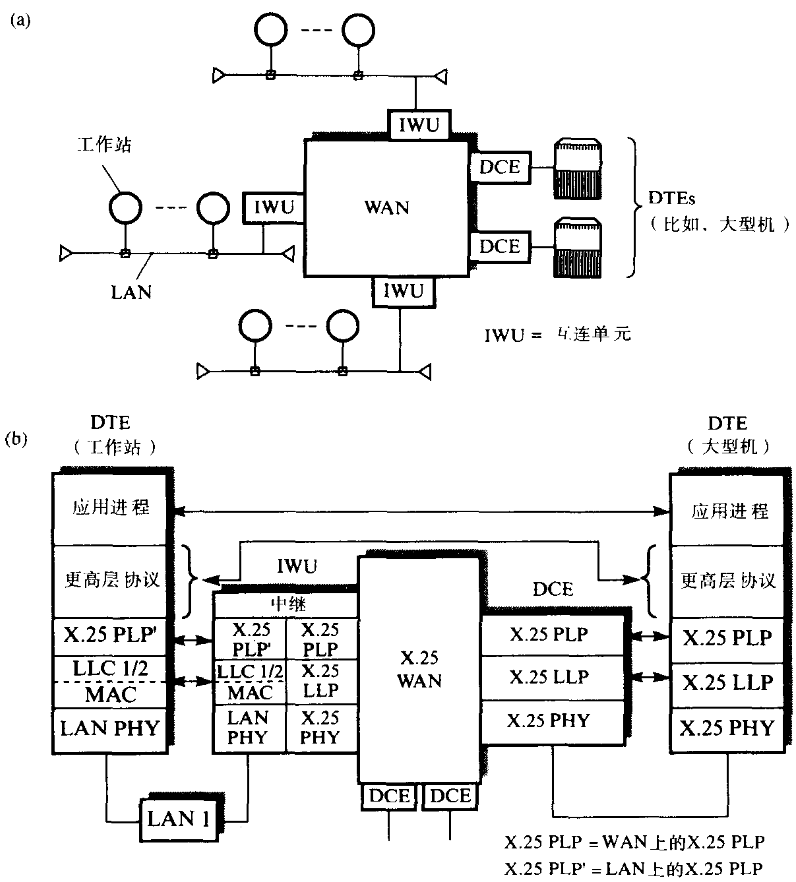


图8-23 LAN上的X.25 PLP

(a) 应用实例 (b) 可能的协议分级结构

8.3 电路交换数据网

用于CSPDN的ISO参考模型中与最低三个网络依赖层有关的各种协议如图8-24（a）所示。CSPDN物理接口的操作特点定义在X.21建议中。它的目标是为用户提供在整个呼叫期间可用的全双工同步数据传输路径。规定X.21的各种交换电路如图8-6所示，所以在本节将集中讨论X.21接口协议的操作。

8.3.1 X.21 接口协议

在CSPDN中，一旦呼叫建立，物理通信路径就存在于主叫和被叫DTE间。所以，X.21接口协议只关心与每一次呼叫相关的建立和清除操作。确保数据传输的控制是链路层的责任，因为CSPDN在端对端基础上操作。如图8-24(b)所示。

建立一个呼叫的典型交换序列、交换数据，以及使用各种有关X.21的交换电路清除呼叫 458 如图8-25所示。图8-25(a)表示通过主叫DTE—DCE接口的交换序列而图8-25(b)表示通过被叫DTE—DCE接口的交换序列。初始时，主叫和被叫DTE的发送（T）电路都设为逻辑1，表明它们都准备启动一次呼叫或接收一次呼叫。类似地，双方DCE的接收（R）电路也处在逻辑1，表明它们可用。

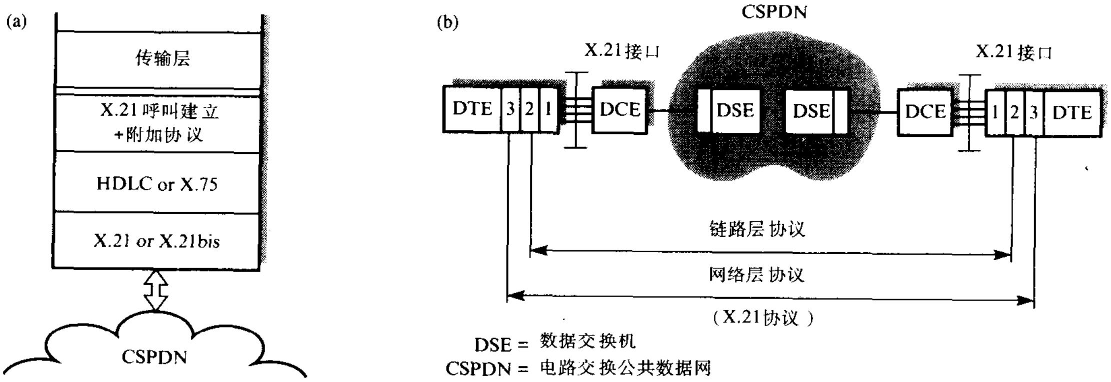


图8-24 多个CSPDN

(a) 网络依赖层协议 (b) 适用范围

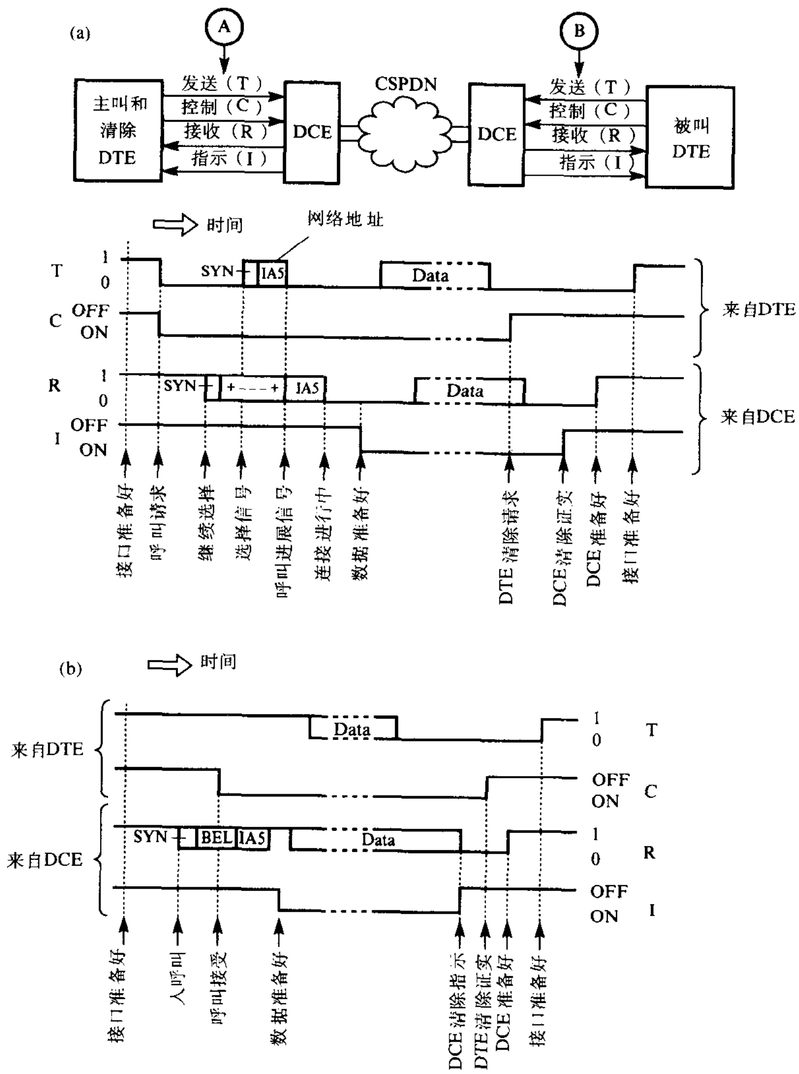


图8-25 成功的呼叫和清除交换序列

(a) 主叫DTE—DCE接口(A) (b) 被叫DTE—DCE接口(B)

459

主叫DTE先把它的控制（C）电路设成开启状态，同时把它的发送电路设成逻辑0，表示它想进行呼叫（见图8-25a）。当DCE准备好接受这个呼叫时，它在接收电路上发送两个（或多个）SYN字符并紧跟一串“+”IA5（ASCII）字符响应。接到“+”字符串，主叫DTE发送两个（或多个）SYN字符以及紧跟的所需目标DTE的网络地址（同样以IA5字符形式，每个字符包含一个奇偶校验位）。地址以一个“+”字符终止。然后DTE进入等待状态，如果DCE想要建立呼叫，则发送空闲（呼叫进行）字符来响应。

当呼叫请求到达所需目标DCE时，后者先发送两个SYN字符和紧跟的一串BEL字符来通知被叫DTE（图8-25(b)）。SYN和BEL是ASCII字符集中的两个控制字符。被叫DTE把它的控制电路设成开启状态来接受这个呼叫，并且DCE依次在接收电路上以IA5字符串的形式传递其他呼叫建立信息。这些信息包括反向计费和类似信息。最后，当主叫和被叫DCE设置它们的指示（I）控制电路来表明电路已经建立并且网络准备传输数据时，呼叫建立阶段完成。

在连接建立以后，有一条数据透明、全双工通信路径在主叫DTE和被叫DTE间用于传输链路层数据（一般，这涉及根据HDLC协议交换帧）。每个DTE在它的发送电路上发起数据帧的传输。它们通过网络传送并沿着来自本地DCE的流入接收电路传递给接收DTE。最后，在一个DTE完成其所有数据的传输后，它把控制电路转变为关闭状态发起呼叫（电路）清除（DTE清除请求）（见图8-25(a)）。但是可以看到，因为电路是全双工的，请求清除的DTE必须准备在流入接收电路上接受更多的数据。

清除请求信号通过网络传递给远端DCE，远端DCE通过把指示控制电路置成关闭状态，来通知它的本地DTE（DCE清除指示）（见图8-25(b)）。本地DTE通过把它的控制电路置成关闭状态来响应（DTE清除证实）。信号通过网络传递给请求清除的DCE，它通过把指示电路置成关闭状态来通知DTE（DCE 清除证实）（见图8-25(a)）。最后连接的两端都回到接口准备好状态。

8.3.2 X.21bis

X.21接口协议主要用于全数字CSPDN。然而，在这种网络广泛应用前，为了使现有的基于EIA-232D/V.24的设备能容易地向更新的X.21设备转换，已经定义了一种称为X.21bis的接口协议，“bis”表示它是供选择的协议。在第2章讨论过的用于同步调制解调器的接口就是这种接口（就是说，它提供了位计时时钟信号），因为同步调制解调器向网络发送数据时必须执行数字到模拟的转换，而从网络接收数据时必须执行模拟到数字的转换。 460

8.3.3 链路层和网络层

如图8-24所示，在CSPDN中链路层和网络层都是端对端协议。但是，因为建立了全双工电路，在全数字网络中链路层协议可以与X.25中的一样（就是说，称为LAPB的一种HDLC）。在以前的模拟接入电路和网络中，只建立了双线路、半双工电路，因此必须使用X.75建议。X.75的链路建立规程是称为LAPX的LAPB的一个派生版本。它主要用于半双工物理电路上逻辑数据链路的建立。

如果不支持流量控制功能，CSPDN网络层（第3层）可以相对简单，因为在连接建立后，每个由传输层发出的网络数据传输服务原语能向中间链路层直接映射成一个类似请求。然后传输层执行自己的流量控制功能。另一方面，为了使各种网络容易交互，可以有类似于X.25中的网络层。如果是这种情况，讨论过的各种VCI会有端对端意义而不是本地意义，如X.25 PSPDN。同样，在CSPDN中每次呼叫当然会是一条单独的电路。

8.4　综合业务数字网

多数国家的PTT机构迅速地把它们现有的PSTN升级成全数字操作。当完成这个转变时，在每个用户出口处，全数字接口可以得到高比特率。同样，因为新网络使用全数字传输和交换，可以为本地、国内、国际呼叫提供很短的连接（呼叫）建立时间。新的用户接口不仅有足够的容量处理语音通信，而且有足够的容量直接处理数据通信。如果需要，这两种业务可同时工作。这种新网络称为**综合业务数字网**（ISDN）。

因为这种网络的深远影响，ITU-U已经为接口设备定义了一组标准，这些标准称为I系列
461 建议。在本章的余下部分会给出ISDN的各种用户接口以及一些I系列建议的概要。

8.4.1　用户接口

ISDN提供的标准多目的用户接口的有限集总结在图8-26中。正如所见，提供的基本业务是语音通信，如已有的电话网络的情况。但是注意，因为用户出口是数字的，语音通信传输前在用户（电话）听筒被数字化而在接收时又转换回模拟形式。因为传输数字化语音需要的比特率为64kbps，用户接口提供若干倍这个基本速率的速率。

在图8-26（a）的第2个例子中，同样的出口（但是不同的用户终端）为数据通信提供电路交换连接。基本数据传输率是64kbps，并且因为使用数字传输和交换可以获得很短的呼叫建立时间。同样，在第3个例子中，用户可以使用出口作为综合语音和数据的设施，两者同时工作。

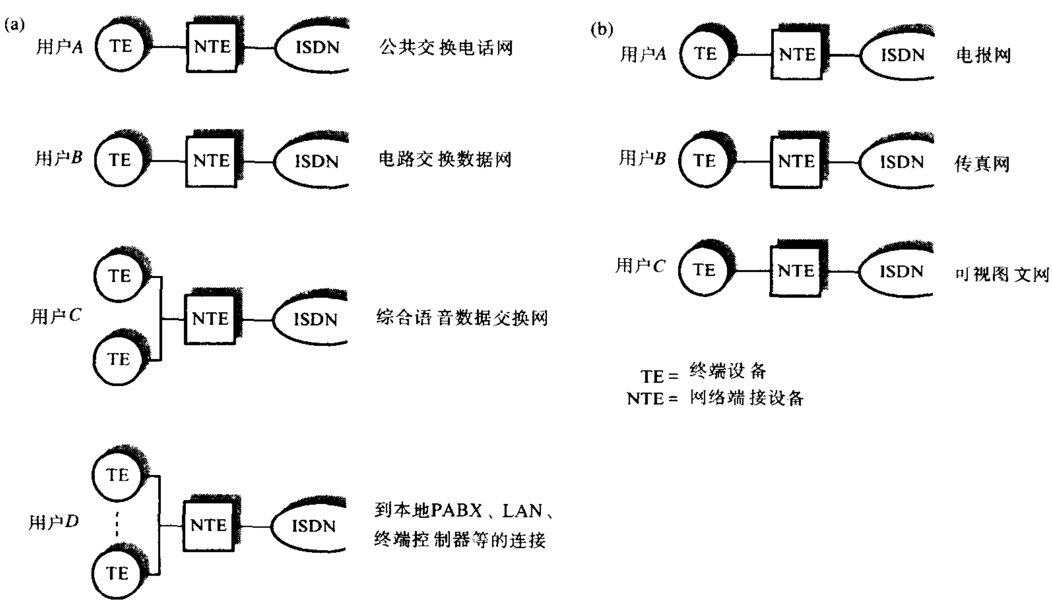


图8-26　ISDN用户业务

(a) 提供基本传输功能的承载业务　(b) 提供终端功能的终端业务

第4个例子说明，使用恰当的终端，基本出口可以提供对分组交换业务的访问。可以看到，它能以64kbps操作或者可选择以更低的16kbps操作。

这四种用户业务简单地使用ISDN提供交换传输路径，称为提供基本传输功能的**承载业务**。但是，PTT机构提供额外的更加复杂的设备来允许ISDN被用作快速**电报**、**传真**或**可视图文网**。

我们称这些为**用户终端业务**，如图8-26(b)所示。用户智能电报网为相似终端间交换报文（由字符和数字以及图形字符组成）提供了通用目标设施。传真网为相似终端间传输电子文档的扫描图片提供了通用目标设施。而可视图文网为获得对存储各种诸如股票和分配价格信息的远端数据库的访问提供通用目标设施。 462

8.4.2 网络访问点

考虑到使用的广度，PTT机构提供的**网络终端设备**（NTE）有许多可能的访问点。它们显示在图8-27中。

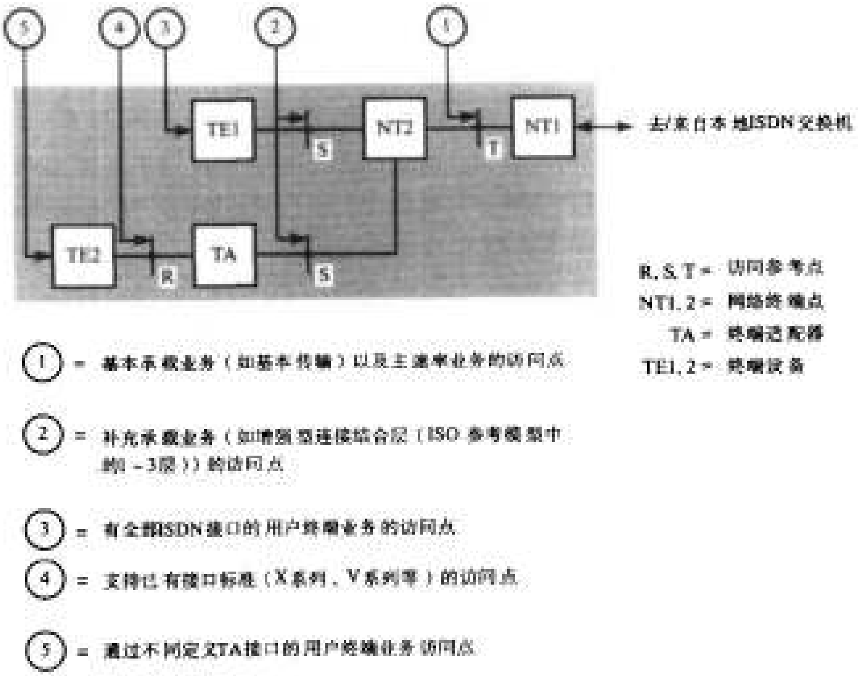


图8-27 ISDN用户访问点

到ISDN网络投入运行为止，许多用户已经在遵守现有标准的设备上作了很大的投资，诸如V系列和X系列。为了适应这种情况，ISDN的NTE提供一类**终端适配卡**，执行必要的映射功能，不仅支持更新一代的设备也支持现有设备和接口。

图8-27中的可能访问点（R、S和T）清楚地说明了NTE中的不同智能水平。与ISO参考模型相关，这些访问点需要不同数量的协议层。图8-28总结了需要支持每个业务的协议层。正如所见，它的范围从支持基本传输业务的1层，到支持诸如交换语音或数据终端业务的1－3层， 463
到支持各种用户终端业务的1－7层。

图8-28 用户访问概述

8.4.3 信道类型

ISDN的**基本传输率接口**（BRI）提供两个称为**B信道**的64kbps信道和一个附加的称为**D信道**的16kbps信道。D信道的基本用途是发信令；NTE用它来通知本地ISDN交换机目标NTE的地址。开辟单独信道用于发信令大大加快了呼叫建立时间。另外，因为建立新的呼叫相对很少发生，当不发信令时D信道也对用户可用，写进了I系列标准中。例如建议D信道应该用于减少包长度的分组交换。这样用户在BRI中总的可获比特率是144kbps（2B+D）。

在一些国家根照请求可以获得更高比特率的业务。这些都可以通过**主传输率接口**（PRI）获得，称为**H信道**。当前，有关这些信道的比特率包括如下：

- H0　　384kbps
- H11　　1536kbps
- H12　　1920kbps

这些信道的用途包括可视电话、可视会议，高速LAN互连，以及用于高速用户终端业务的交换信道。它们在参考点1的T接口被访问。包括信令信道位置的PRI结构如图2-25所示。各种接口和可能比特率的建议范围总结在图8-29中。

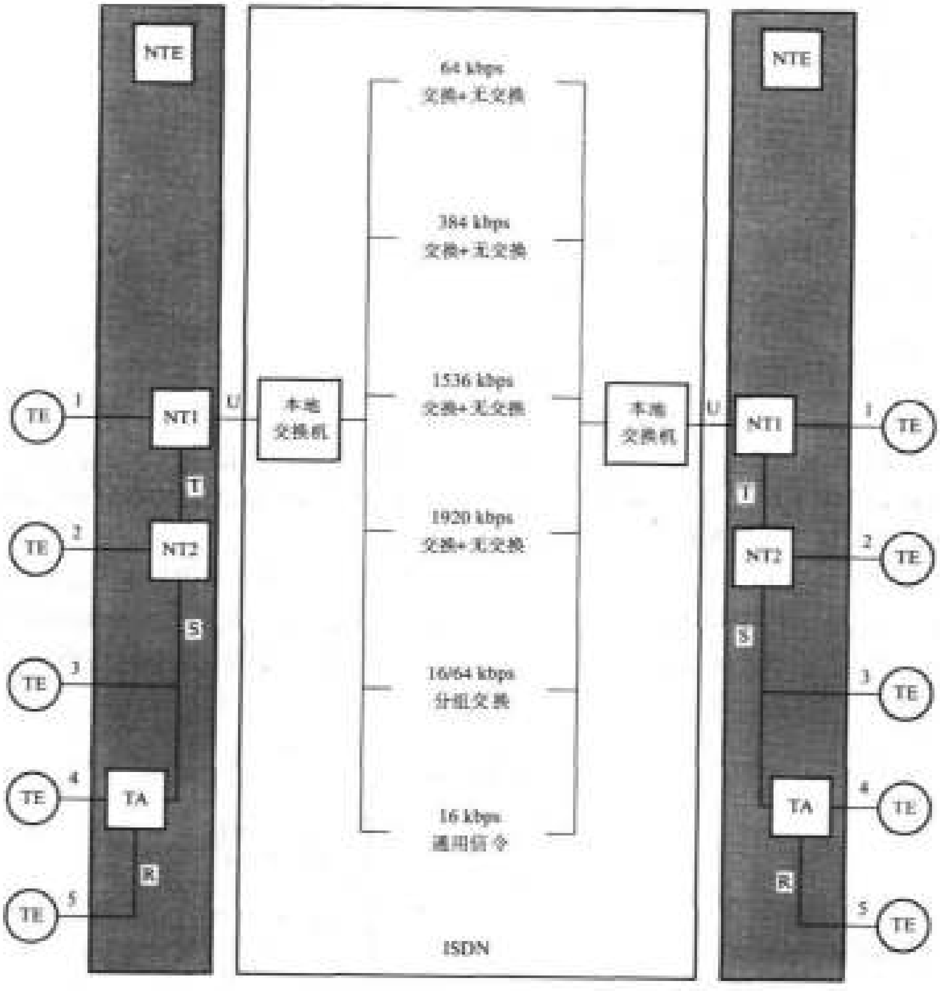


图8-29 用户-网络接口概述

8.4.4 用户–网络接口

用户–网络物理（第1层）接口的示意图如图8-30所示。本地交换机和客户前端间的电路是一根双绞线。为了基本传输率访问业务，它必须支持2B+D信道的双工（2路同时）传输。 464
如第2章描述，用在这条电路的传输线编码是2B1Q；混合转变器用作双工传输。原理上，转换器只允许接收到的信号传递给接收方部分，但由于非理想情况下，（更强）发送信号的一部分也反馈给接收方部分。为了克服这个问题，一种**适配混合器**或**回波消除器**用来从接收到的合成信号中消除（已知的）发送信号。为了使回波消除器正常工作，发送信号和接收信号间必须没有关联。为了避免这种关联，使用了两个**扰频电路**来以伪随机（可重复的）方式随机化位序列，其中一个用于发送而一个用于接收。

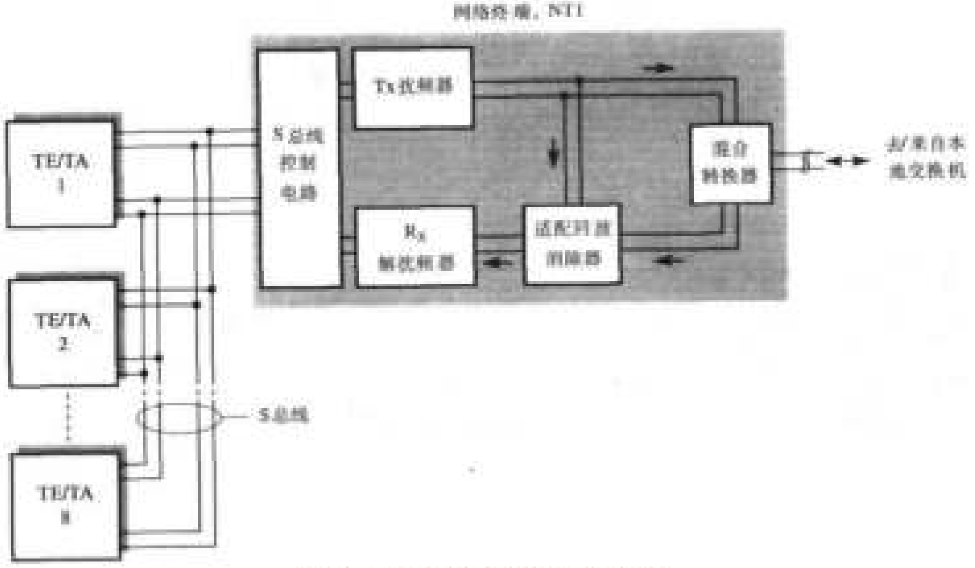


图8-30 用户–网络物理接口示意图

在一些国家执行网络端接功能（NT1）的设备在技术上属于客户端。但是，在另一些国家它们属于网络提供者，这样用户接口启动NT设备的客户端。它是4线（2对）接口，一对用 465
于传送而另一对用于接收。每对线使用允许与B信道和D信道相关的位周期被确定的标准帧结构，承载2B+D二进制数据流。

因为NT的客户端是S接口，4线电路称为S总线。它允许最多连接8个独立的终端设备或终端适配器。合成了争用控制方案，使得各种设备项能以公平的方式分时共享两个B信道和一个D信道的使用。因此，NT设备由于功能丰富所以是个相对复杂的盒子。

8.4.5 用户接口协议

ISDN的关键特性是信令信道跟正常语音信道和/或数据信道的逻辑分离。信令信道因为用于呼叫建立，据说是控制平面或C平面的一部分，而用户信道属于用户平面或U平面。已经指出，ISDN支持电路交换和分组交换两种业务。它还支持两种新的帧级别业务，一种称为**帧中继**而另一种称为**帧交换**。各种业务类型如图8-31所示。

不论是哪种业务类型，在传输用户数据前必须先在网络中建立电路/虚拟路径。虽然也提供**半永久连接**（类似于租用电路），但在交换连接中它在呼叫时完成。为了建立电路或虚拟路径，使用3层协议栈在TE/TA及其本地交换机间的D信道上交换信令报文。在网络本身，包括 466
全7层协议栈的单独**信令网络**用来在网络中建立必要的电路或虚拟路径。这称为**公共信道7号**

信令系统（CCSS-7）。单独信令网络允许网络运营商能更轻易地提供大范围的高级服务，因为可以根据要求的必需服务建立路径。

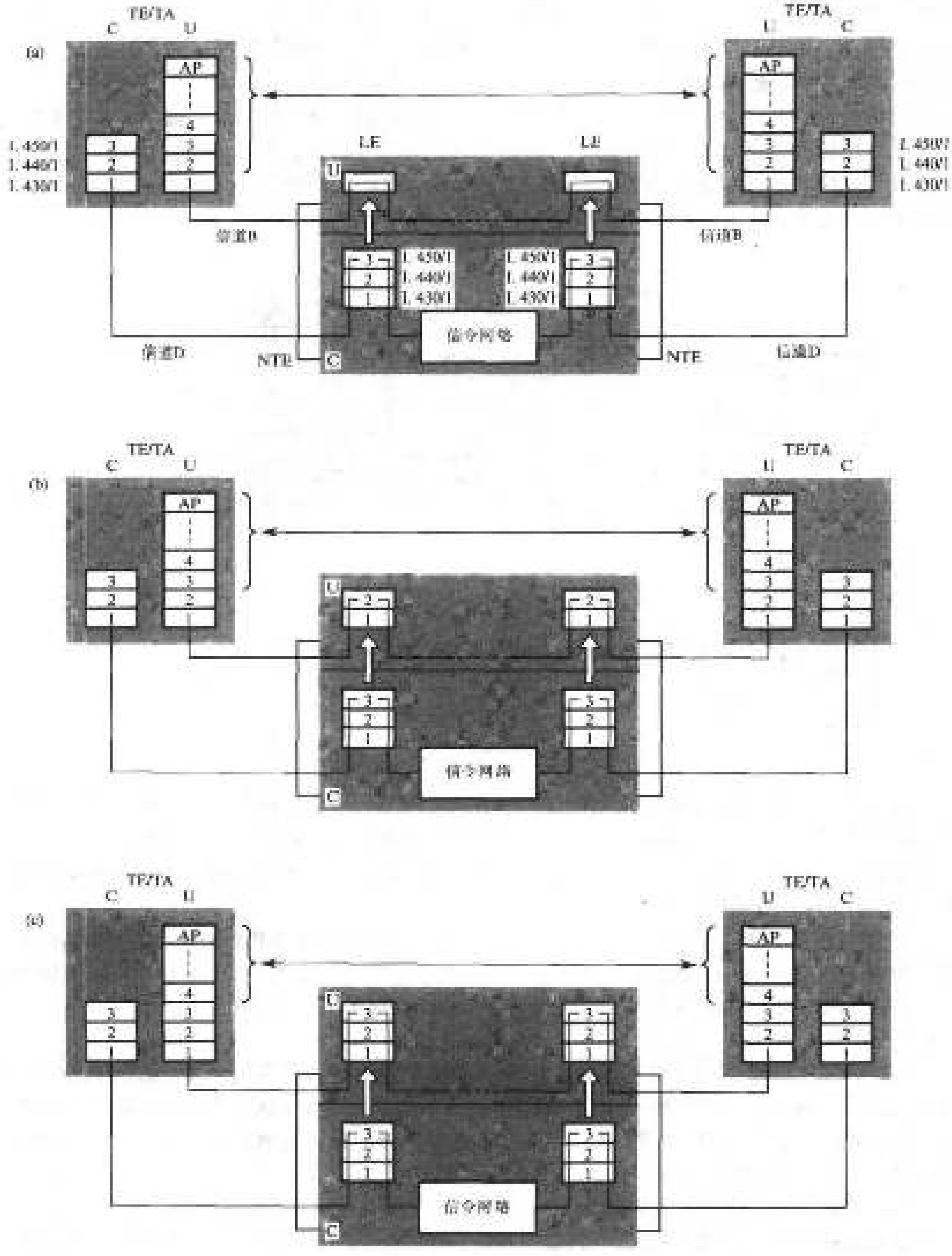


图8-31　用户接口协议

(a) 电路交换　(b) 帧中继/帧交换　(c) 分组交换

对于电路交换业务，以PSTN中同样的方式建立电路，如图8-31(a)所示。建立的电路提供透明的64kbps通信路径。

对于两种基于帧的服务，与PSPDN中的VC类似的方式建立称为**虚拟路径**的VC。额外的路由选择信息被保留在每个中间交换机，使得用户数据的随后帧通过建立的虚拟路径被路由（中继），如图8-31(b)所示。在帧中继中，支持简单的最佳尝试服务。另一方面，在帧交换中每个帧都执行差错控制和流量控制。与分组交换比较，帧的路由选择更简单并且能以更高的传输比特率进行。一个用户同时可以建立多条到不同目标的虚拟路径，并且网络会使用包含在每个帧内的寻址信息路由帧到它们所要的目标设备。还可以请求半永久虚拟路径。

分组交换服务使用类似于X.25 PSPDN中的全3层用户栈，如图8-31(c)所示。实际上，因为现在PSPDN应用广泛，它的可选方案也被建议到I系列标准中，用来提供ISDN与PSPDN的交互工作。它定义在ITU-T X.31建议和I.462建议中。在这个方案中，在ISDN中建立一条到PSPDN网关的基本电路交换连接，如图8-31(a)所示。网关以类似于STE的方式操作，ISDN为分组方式终端设备和PSPDN网关间简单地提供一条透明64kbps交换或半永久电路。用户协议栈使用第3层的X.25 PLP，链路层协议定义在I.420。

I系列标准建议以类似的方式使用D信道。但是在这种情况下，X.25包会使用信令网络通过ISDN路由到网关，而不是使用B信道电路。虽然这可以避免建立电路，但D信道的使用把包长度限制到260个字节，相比之下其他服务的包长度可以到1024个字节。D信道的16kbps传输率也会更低。

8.4.6 信令协议

三个有关信令（D）信道的协议表示在图8-31(a)中。两个第1层协议I.430/1合起来定义了从用户TE到本地交换机的物理接口。回忆一下这些协议除了包括建议的接口方案的规定外，还包括用于位、字节和帧同步的机制以及电源流入，它使得在发生本地电源故障时仍能进行（电话）呼叫。这样4线到2线的转换、不同的传输格式以及总线上存在其他TE对第2层（和第3层）是透明的，它只看到一条2B+D信道的双工传输路径。 467~468

第2层协议定义在I.440/1中，同Q.920/1一样。称为LAPD的规程用来传输级别为3的呼叫建立报文。第5章已描述了LAPD的基本操作，因此在这里我们只考虑它与信令规程的联系。

回忆一下早先的关于物理接口的讨论，因为最多能有8个不同的设备共享用户信道和信令信道，我们必须加入一种识别指定设备项的方法。这可以通过使用每个第2层（LAPD）帧头部的地址字段获得。该地址字段含有两个子地址（标识符）：**服务访问点标识符**（SAPI）和**终端末端点标识符**（TEI）。SAPI允许不同终端属于一个不同的**业务类**（比如语音或数据），而TEI确定了关于业务类的特定终端。**广播地址**还允许交换机发送帧到多个终端，使得一个流入语音（电话）呼叫能被所有的语音TE接收到。

但是，注意这个地址在用户访问线上只有本地意义，并且在网络信令系统内的信令报文路由选择中不发挥任何作用。如图8-9所示，地址的最后部分只有本地意义并且会被网络信令系统透明地传递。因此，在ISDN地址中，一种典型的方案是使用该地址的一部分来确定所需的特定TE/TA。如果用户数据也在D信道（分组交换）上传送，那么剩余部分用作层间地址选择器（SAP）。另一种情况，如果用户数据在B信道上传送，那么就不需要地址选择器了。后者的情况下，使用的SAP会成为与U平面相关应用协议的功能。

第3层协议定义在I.450/1中，同Q.931一样。这个协议涉及在D信道上建立呼叫要交换的报文（分组）序列。使用的简写报文类型列表如下：

- 呼叫建立　ALERTing
  CONNECT
  CONNect ACKnowledge
  SETUP
  其他
- 信息传输　USER INFOrmation
  其他
- 呼叫清除　DISConnect
  RELEASE
  RELease COMPlete

469 其他

这些报文中的一些有本地意义（TE/NT）而另一些有端对端意义（TE/TE）。但是，所有的报文在第1层I帧内被传输。说明这些报文的使用情况的一个例子如图8-32所示，它假定一条可以在B信道上用作语音和/或数据传输的电路交换呼叫。

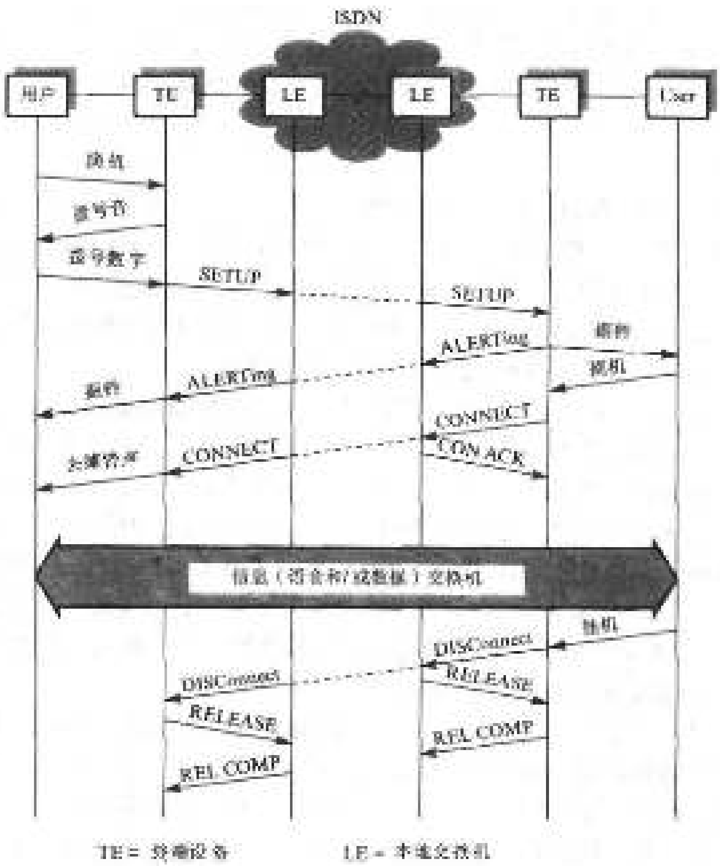


图8-32　电路交换呼叫的报文序列实例

8.4.7　帧中继服务

如8.4.5节所述，关于ISDN有两种新的帧方式服务：帧中继和帧交换。两种服务使用相同

的信令规程。主要的差别是网络只在帧交换方式中执行差错控制和流量控制。关于帧服务的规程定义在ITU-T I.122/Q.922建议中。实际上，到目前为止帧中继由于它最小限度的开销成为占主导地位的服务，所以将更详细地讨论它。 470

回忆在X.25中，多条VC（呼叫）的多路复用由分组层处理，而链路（帧）层只关心由此产生的帧在本地DTE—DCE链路上的差错控制。所以，组合的链路/PLP相当复杂，并且每个网络交换机的相应包吞吐量受到每个包的高处理开销的限制。

相比之下，在帧中继中，多路复用和路由选择在链路（帧）层执行。而且，帧的路由选择非常简单，所以组合链路传输比特率比分组交换更高。因此虽然它为ISDN的使用而定义，但帧中继在专用网络中还有广泛的用途。

帧中继允许到不同目标的多个呼叫可以同时进行。由此，当每个呼叫（虚拟电路）首先建立时（使用D信道作为对L_CONNECT.request服务原语的响应），分配给它一个称为**数据链路连接标识符**（DLCI）的惟一连接标识符。所有随后与这个呼叫相关的数据传输请求都把这个分配的DLCI作为参数。DLCI被嵌入由此产生的帧的头部，并用来路由（中继）帧到它们所要的目标。在半永久虚拟路径的情况下，DLCI在注册时被分配。

每个在用户B信道传输的帧的格式如图8-33(a)所示。因为缺乏差错控制，它由无控制字段的2字节（扩展）地址头部组成。除了DLCI，头部还含有**正向显式拥塞通知**（FECN）**位**，**反向显式拥塞通知**（BECN）**位**以及**丢弃合法**（DE）**位**。这些用来控制网络内的拥塞。

DLCI像分组交换网络中的VCI一样，在特定网络链路上只有本地意义，所以帧通过与虚拟路径相关链路时会变化。当虚拟路径被建立时，每个路径（路由）上的交换机在D信道接到呼叫请求包后，先确定帧到所需（ISDN）目标地址要走哪条出链路，接着获得该链路的空闲DLCI，然后在入链路/DLCI和相应出链路/DLCI的链路路由选择表中增加一条记录，如图8-33(b)所示。此外，对于半永久虚拟路径，在预定时增加记录。

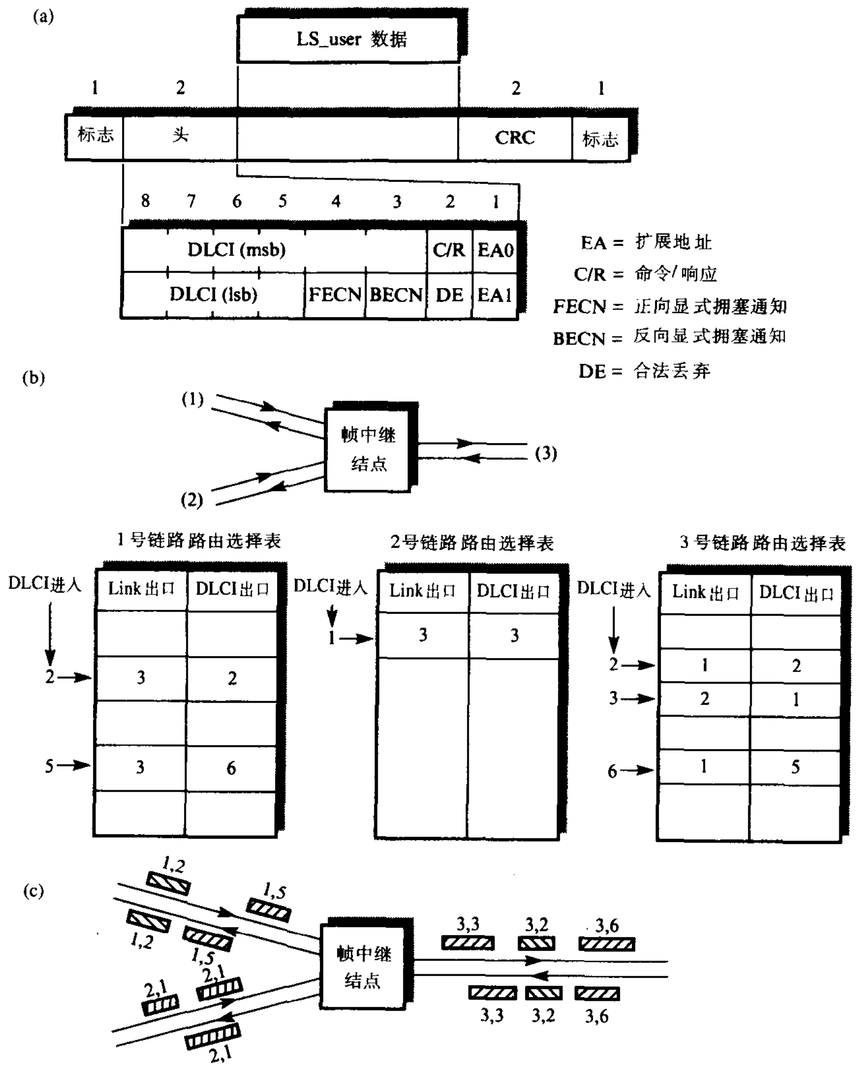
在随后的数据传输（帧中继）阶段，当收到帧时，每个交换机内的帧处理器简单地读取帧内的DLCI并把它与入链路号组合来确定相应的出链路和DLCI。新的DLCI写入帧头部然后该帧排队等待在适当链路上转发。这样中继帧的顺序被保存，并且它们的路由选择非常快。

因为多个呼叫能在网络中的每条链路上同时进行，并且与每次呼叫相关的帧以随机时间间隔生成，所以在大量通信期间出链路会临时过载，其结果是开始建立它自己的队列。这称 471
为**拥塞**：每个帧的附加拥塞控制位用来减轻这种状况。

无论何时帧处理器中继帧到链路输出队列，它都会检查队列的长度。如果超过了规定的限度，帧处理器会发信号把此状况通知给这个呼叫的两个终端用户。这在正向方向上通过设置帧头部的CF位完成。在反向方向上，它通过设置所有在该链路上接收到的帧头部的CB位来完成。如果状况继续，帧处理器返回一个称为**统一链路层管理**（CLLM）的特殊帧给受影响链路上涉及的路由（路径）的所有用户设备。这些帧被每个中间交换机以正常方式简单地中继。 472

当终端用户设备中的帧处理器接到网络拥塞的指示时，它暂时降低帧转发速率直到没有更进一步的拥塞指示。但是如果过载加剧，交换机必须开始丢弃帧。使用帧头部的DE位来获得公平性，因为无论何时一个用户超过了协商的吞吐速率，每个用户系统中的帧处理器就会设置该位。

为了使传送差错帧的概率降到最低，每个帧尾部的CRC用来检测帧头部（和信息）字段的位差错。然后，如果检测到差错，这个帧就被丢弃。在帧中继服务中，差错恢复留给终端用户设备中的更高协议层。



DLCI进入	Link出口	DLCI出口
2	3	2
5	3	6

DLCI进入	Link出口	DLCI出口
1	3	3

DLCI进入	Link出口	DLCI出口
2	1	2
3	2	1
6	1	5

图8-33　帧中继原理

(a) 帧格式　(b) 帧路由选择　(c) 帧中继示意图

虽然定义的帧中继服务与ISDN相关，但是许多公共运营商已建立了只提供帧中继服务的网络。一个典型公共帧中继网络的示意图如图8-34(a)所示。一般，大型合作客户使用帧中继服务互连不同地点的LAN形成一个跨企业数据网络。客户先通知服务提供方需要互连的地点。然后提供方通过在每个帧中继交换结点建立相应的路由选择表记录来产生互连所有地点的一组永久虚拟连接。接着提供方通知每个地点的网络管理者要放入到达其他每个地点的帧头部的DLCI值。

所有帧被多路复用在一起送到连接**用户接口设备**（CIE）到最近交换结点的链路上。逻辑
473 上，这看起来像一组它自己与其他所有地点间点对点连接的用户设备，每个连接由相应的DLCI确定。CIE可以是网桥或路由器。假定它是网桥，虽然它只有两个物理端口，但逻辑上它作为每个DLCI（等同于各个端口）的多端口网桥工作。指定到达每个其他地点的DLCI值由网络管理载入网桥。无论何时该网桥接收到需要转发到不同地点（端口）的帧，网桥把这个

帧封装成帧中继帧，相应的DLCI放在帧头部。帧中继网络中的交换结点通过相关永久虚拟连接路由这个帧到所要的目标。

在公共网络中提供给用户的无连接数据服务称为**交换多兆位数据业务**（SMDS），在欧洲它还称为**无连接宽带数据业务**（CBDS），并且CIE作为**SMDS边缘网关**。假定远端网桥用于网关，典型的互连协议体系结构如图8-34(b)所示。 474

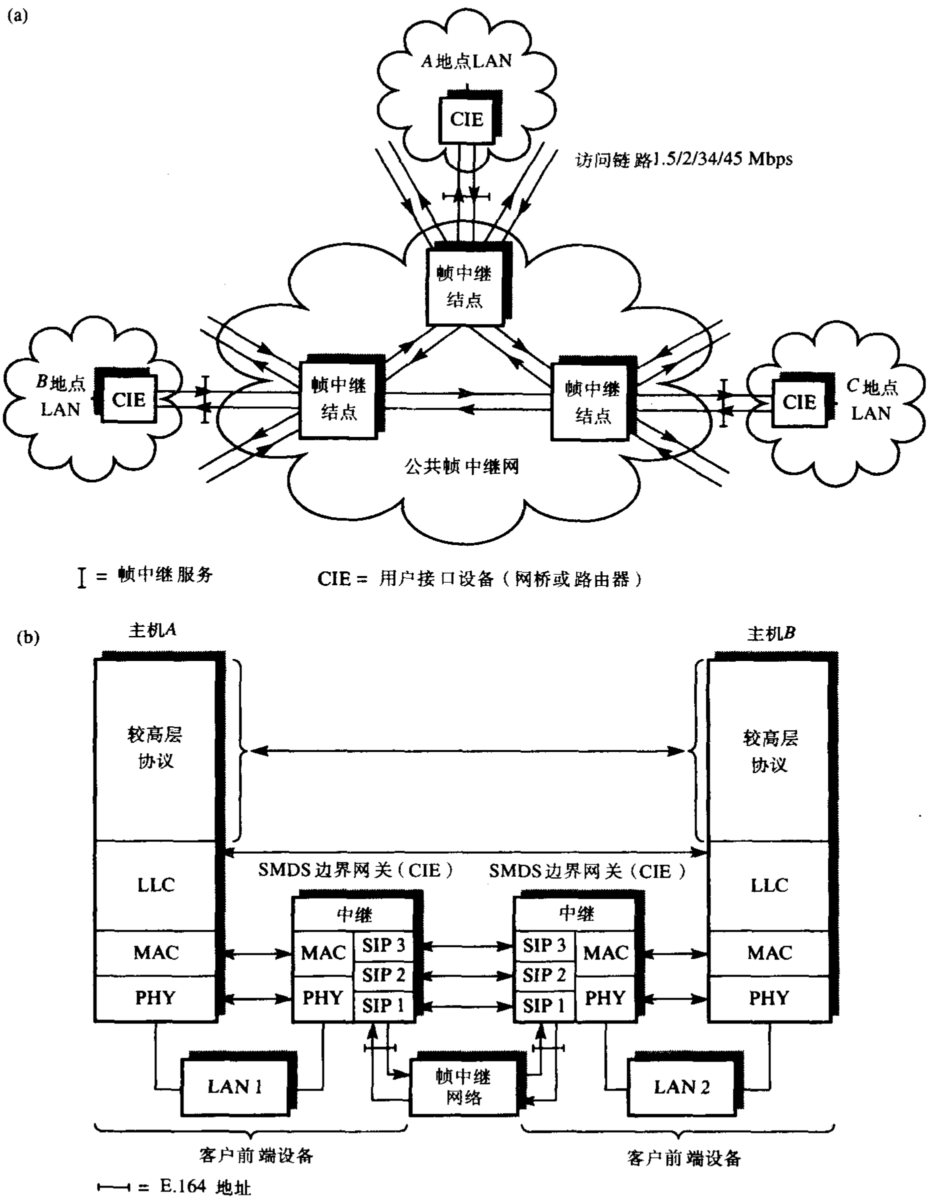
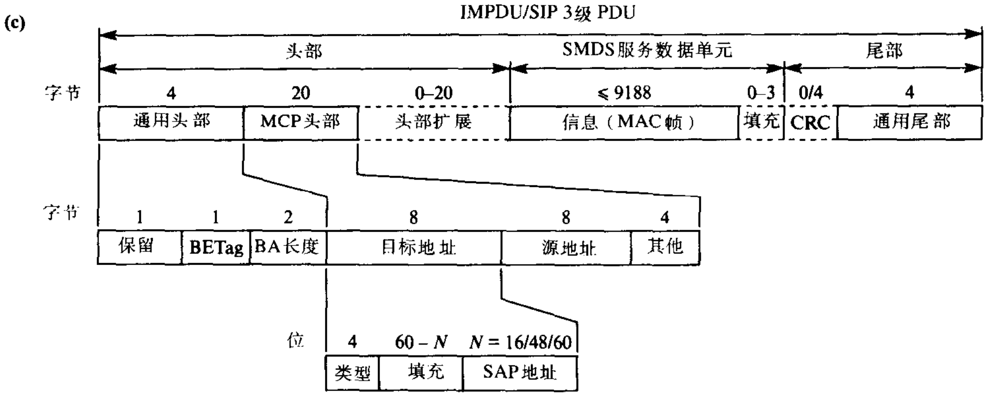


图8-34 公共帧中继网

(a) 网络示意图 (b) SMDS协议体系结构 (c) 初始MAC PDU格式



IMPDU = 初始MAC PDU　　MCP = MAC会聚协议
SIP = SMDS接口协议　　SAP = 服务访问点
BETag = 开始—结束标记　　BA = 缓冲区分配

图8-34 （续）

有关SMDS的协议层称为**SMDS接口协议**（SIP）1级、2级和3级。因为不同的LAN类型使用不同的头格式和最大帧长度，SMDS服务数据单元可以达到9188个字节以适应最大的LAN帧。同样，由于使用不同的地址格式，在传输MAC帧前，3级 SIP 先在标准头部和尾部间封装帧。如果需要，为了简化目标端的缓冲操作，它在提交帧的尾部加入额外的填充字节，这样它的长度达到若干倍4字节长。由此产生的报文单元（当作包）称为初始MAC协议数据单元（IMPDU），它的格式如图8-34(c)所示。

可以看到，头部由两个或（可能）三个字段组成。普通头部含有一个称为开始—结束标记的8位序号，用来使3级SIP 能检测丢失帧，以及需要存储完整IMPDU的缓冲内存量的说明。MAC会聚协议（MCP）头部含有许多关于该协议的子字段。它们包括源和目标网关的地址，在公共网中是定义用于ISDN的60位E.164地址。但是，为了满足其他地址类型，它们都是64位地址字段，其中四个最高位用来确定剩余60位中的地址类型，例如16/48位MAC地址。其他子字段包括存在的PAD字节数和是否存在CRC的指示。还包括了头部扩展允许将来加入额外子字段。如刚才所述，尾部可以包括一个可选32位CRC，如果存在针对完整IMPDU的差错检测。通常尾部含有与通常头部相同的信息。由此产生的IMPDU由SIP 2级（帧中继）协议层封装成帧中继帧。1级1SIP执行物理层会聚功能并负责由此产生帧在访问链路上的传输。

8.4.8 反多路复用

在8.4.3节确定的各种较高比特率业务（H0/H11/H12（384/1536/1920kbps））都是利用PRI电路提供。像BRI，PRI使用单独的（频带外的）信令信道来建立呼叫，并且作为呼叫建立阶段的一部分提供所需的带宽（384/1536/1920）。另外，一种称为**ISDN 多传输率**（ISDN multirate）的新业务可从一些载波中获得，它允许用户请求呼叫的带宽是64kbps的任意倍，因此该业务还称为**交换*N*倍64kbps**，根据载波*N*最多可以是24或30。

475 另外，用户能使用位于客户室内的称为**反多路复用器**的设备从PRI得到相似的服务集。多路复用作为在单独一条高比特率电路传输多个低比特率(64kbps）数字化语音信号的一种方式，首先在2.5.2节被引入。反之，反多路复用用来使用多条较低比特率电路得到一条高比特率信道。例如，一个反多路复用器能用来从6条独立64kbps电路得到一条384kbps信道。图8-35(a)给
476 出了反多路复用器的工作原理的示意图。

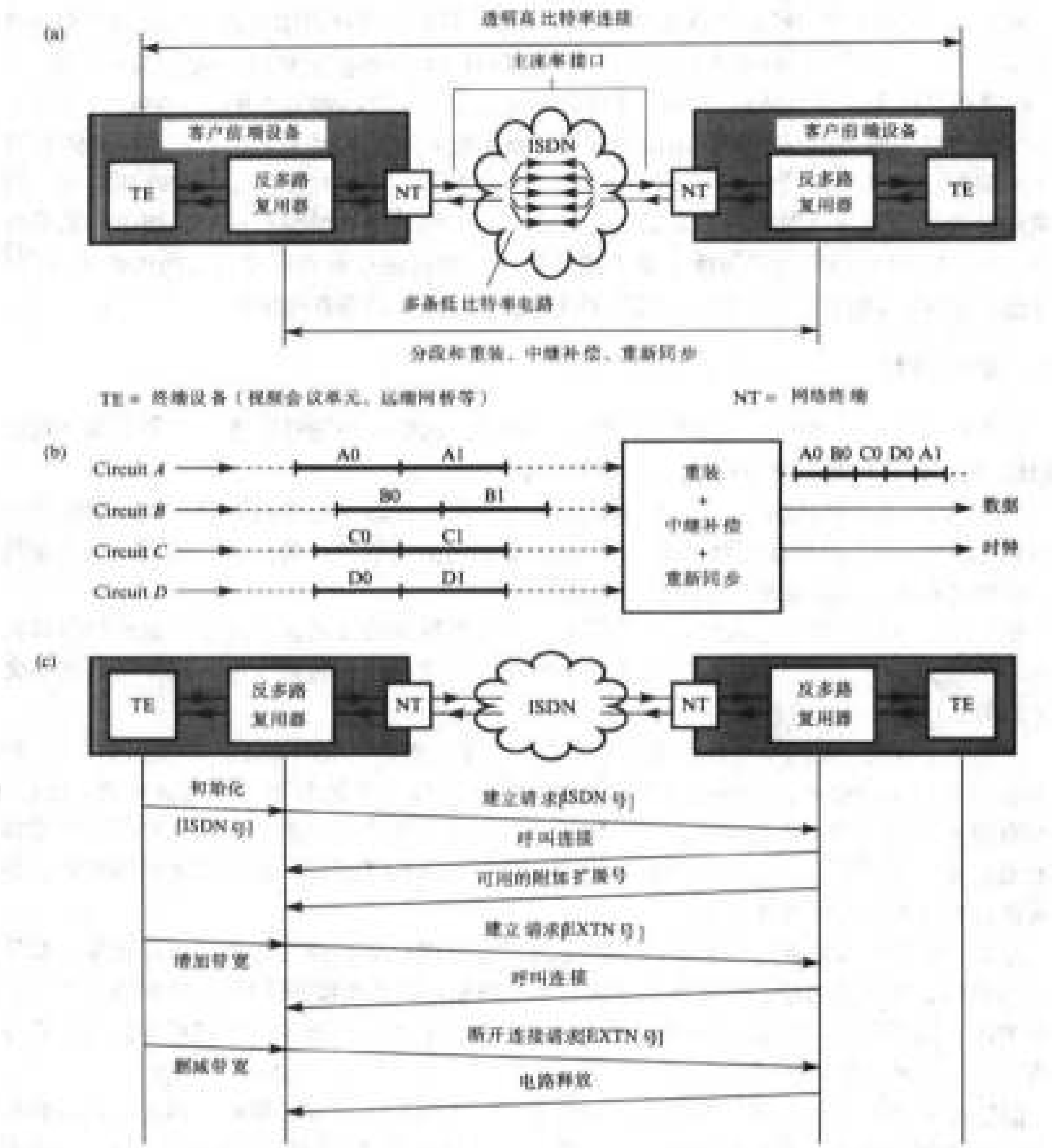


图8-35 反多路复用

(a) 工作原理 (b) 重装示意图 (c) 结合协议

在发送方，反多路复用器先建立到所需目标的适当数目的64kbps电路。然后继续把用户设备（比如可视会议单元）输出的高比特率数字流分段，准备在多条低比特率电路上传输。在接收方，一旦已建立多条电路，反多路复用器接受从这些电路上接收到的比特流并把它们重装回一条高比特率信道向前传输给接收终端设备。

所以，反多路复用器的作用除了建立呼叫，还执行对用户设备透明的分段和重装操作。实际上，因为每个信道是单独建立的，它们都经过不同的路由/路径横跨ISDN主干网。这导致从每条电路接收到的信号的时隙差异。为了补偿这个差异，接收方的反多路复用器必须执行重装二进制数据流的时延补偿和重新同步。概要如图8-35(b)所示。

为了获得交换N倍64kbps传输率提供的相类似的服务，用户终端设备可以使用反多路复用器来按需建立和清除64kbps电路。这种技术称为**结合**（bonding），概要如图8-35(c)所示。

响应用户请求，主叫方的反多路复用器请求建立一条到远端地点的单一64kbps电路。一旦顺利进行，被叫方的反多路复用器会发回一张本地可用（扩展）号的表。主叫多路复用器每个时刻建立一条所需的额外电路，然后两个用户设备就能交换数据了。在呼叫期间，用户终端设备通过动态地建立或清除电路，可以请求可用的总带宽增加或减少。例如，如果终端设备是用于LAN互连的远端网桥，那么能在任何时刻调整总带宽以适合交换的数据通信量。现在已经由ITU-T制定了一个国际标准**H.221建议**，涉及这种设备的操作。

8.5 专用网络

在本章引言中已提到，作为使用PTT或公共载波网的一种选择方案，许多大型公司建立（管理）它们自己的跨企业专用综合语音数据网。

通过PSTN或公共ISDN建立的电话（语音）呼叫，按时间和距离收费。如果网络用于数据传输（在PSTN中使用调制解调器），呼叫也按相同方法收费。类似地，使用公共数据网建
477 立的呼叫通常按相同的方法或者按传输数据的质量收费。

在大多数部门（公司、大学、医院等）中，大多数通信（语音和数据）是部门内部的而只有一小部分是对外的。为了降低公共网络费用，所有部门安装用于电话的专用自动小交换机（PABX）和用于数据通信的（专用）局域网（LAN）。

只在一个地点（部门）工作的企业，所有外部呼叫必须使用PTT或公共载波网建立。但是分布在多个地点的企业，一种可选方式是扩展每个地点有关的专用设备把所有地点连起来，因为有很大比例的呼叫是地点间呼叫。实际上，选择基于许多因素，一个主要的因素是彼此间的通信量。这是因为建立跨企业的专用网通常涉及从PTT或公共载波租用传输电路。租用按每日计费而不是按呼叫次数计费。

许多小型和中型企业，内部呼叫的数量还不足以用租借线路把各个地点连起来。相反外部呼叫使用公共载波网建立，当然指电话。对于数据，如果要使用（模拟）PSTN，那么一个额外的因素是交换呼叫的建立时间相对较长。还有，公共设施的安全级别经常作为选择专用网络还是公共网络的一个因素。

但是对于许多大型企业，部门间的通信量（语音和数据）是大量的。因此，都安装和管理自己的专用综合语音和数据网络。好处是可以容易地提供更加复杂的业务，因为网络是专用的，远离传输线路，当然更安全。这种网络简单地称为**专用网或跨企业网**。如果它们跨越多个国家，就称为**全球网**。

体系结构

图8-36(a)是典型专用网配置的示意图。通常专用网由租用线路互连的一组**智能多路复用器**（IMUX）（每个地点一个）组成，形成**跨企业骨干**传输网。一般，网络使用2.5.2节介绍的高速数字租用线路。它们的速率通常是若干倍64kbps；1.544Mbps（DS1/T1）和2.048Mbps（E1）。

每个IMUX有一类语音和数据接口来满足地点的需求。在语音的情况，它们涉及与电话听筒的直接链路或者更普通的与PABX的高传输比特率链路。我们在第3章提到，通常每个语音电路使用64kbps，但一些多路复用器采用复杂的压缩算法来提供使用32kbps、16kbps甚至8kbps的高质量语音通信。这意味着两个、四个或八个语音电路能多路复用到一个64kbps时隙，
478 站间所需的电路数得到可观的节省。这种技术称为**子传输率多路复用**。

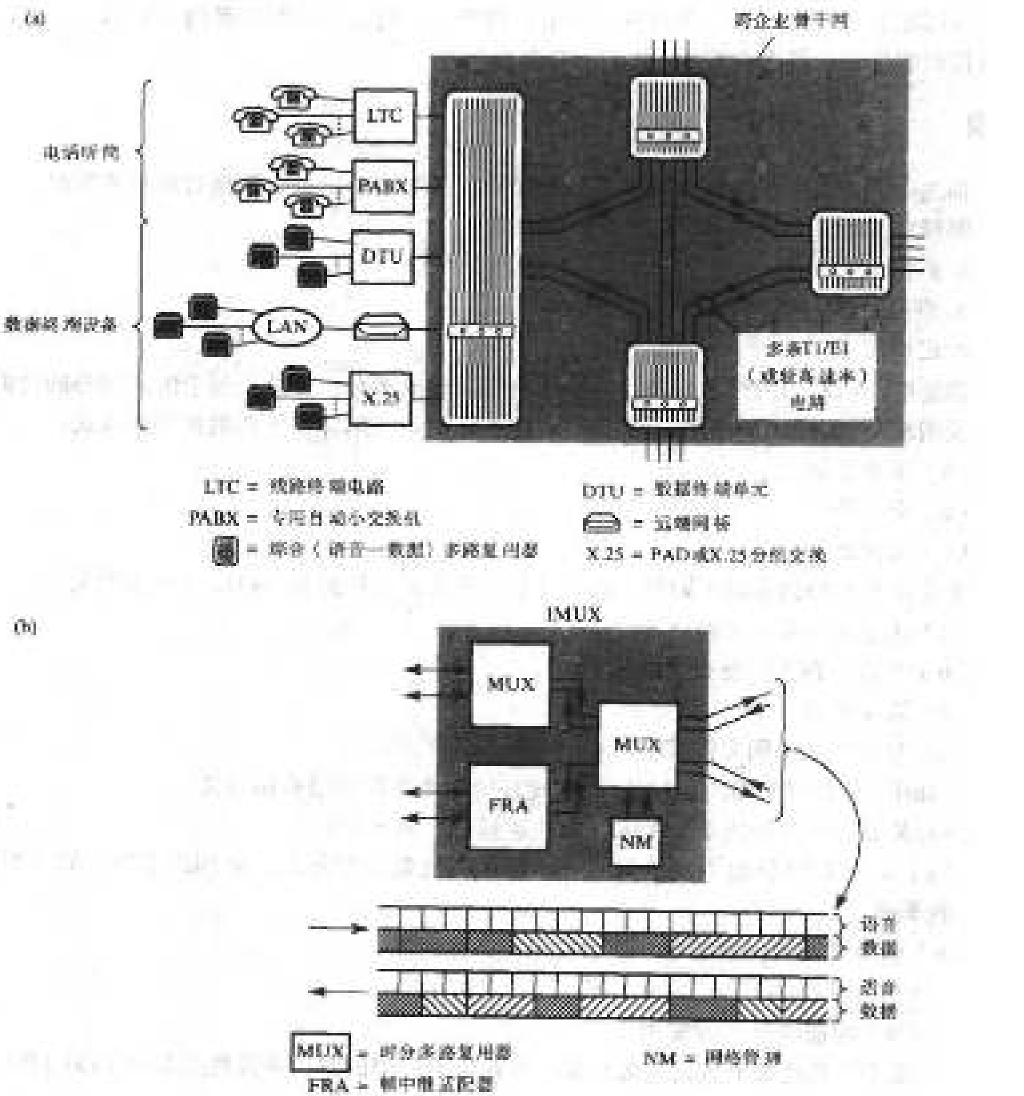


图8-36 专用WAN

(a) 网络示意图 (b) 站点示意图

如3.7.1节所述，一种类似的技术用于连接异步或同步终端，如PC。在这种情况下，一个64kbps信道能用来支持多个终端。例如20×2.4kbps和6×9.6kbps。这种技术称为传输率自适应。这些链路通常用于连接分布的数据终端群体到中央计算机，例如跨企业的电子邮件服务器或者数据库。

LAN在许多部门中的应用不断增长，意味着提供了一种连接LAN方法的通用惯例。如 479
第7章所述，通常使用远端网桥做到。有时也加入专用X.25分组交换机。另一种情况，如图8-36(c)所示，现在加入专用帧中继适配器（FRA）获得（统计）多路复用附加级别。因为在专用网中，电路通常在半永久基础上使用网络管理在网络中建立，每个FRA只需要包长度的数据。这只在要传输数据时完成，就消除了分配永久信道的需要。

虽然许多专用网络有它们所属的企业运营和管理，但是现在许多PTT和公共载波运营商合

作提供能在公共网络上建立等价专用网络的设施。它们称为**虚拟专用网**（VPN），提供与专用网络相似的服务；但由PTT或公共载波管理和运营。

习题

8.1　描述电路交换网络和分组交换网络的差异。清楚地给出这些网络对用户的影响。

8.2　解释对如下分组交换数据网络术语的理解：

a. 数据报

b. 虚拟呼叫（电路）

c. 逻辑信道

8.3　借助图说明X.25网络访问协议的适用范围及组成部分，并给出每个组成部分的功能描述。

8.4　说明X.25协议分组（网络）层的用户服务原语集。解释如下的附加使用方式：

（a）更多数据

（b）限定符

（c）交付证实

8.5　列出协议X.25的分组（网络）层执行如下操作所使用的主要PDU（包类型）：

（a）建立虚拟呼叫（电路）

（b）在该电路上交换报文单元

（c）清除呼叫

画出时序图来说明交换PDU实现这些操作的序列。

8.6　解释用于X.25 PSPDN 接口的X.121地址的各个字段的结构和含义。

8.7　解释X.25分组（网络）层的术语“快速选择”的含义。

8.8　（a）描述X.25分组（网络）层协议使用的流量控制方式，并列出实现它的分组层PDU（包类型）。

（b）画图说明如何控制单个逻辑信道的数据包流：

（i）窗口机制

（ii）附加监管包的使用

图中应包括每个数据包被传输时逻辑信道两侧的发送和接收变量（（V(S)）和（V(R)））

480 以及确认变量（V(A)）的状态。

8.9　区别用于X.25分组（网络）层的复位和重新启动差错恢复规程，并且解释它们的操作。

8.10　（a）描述基于X.25网络的PAD的功能，并且在图中说明它使用的各种协议。

（b）概述PAD使用的如下协议的基本特点：

（i）X.3

（ii）X.28

（iii）X.29

8.11　描述X.25网络中的分组交换机如何路由包。在描述中包括网络以及链路路由选择表的结构和用途。

8.12　解释互连X.25网络的信令端接交换机（半网关）的作用。

8.13　使用基于X.25的网际互连的草图，说明X.75协议的适用范围。解释X.75 PPDU中如下字段的用途：

（a）网络杂项

(b)D位

8.14 (a)概述用于电路交换数据网的三层最低的与网络有关协议层的功能。

(b)画图表示X.21的各种互换电路并概述它们的作用。

(c)借助时序图，描述X.21接口协议的操作。清楚说明在主叫DTE—DCE接口和被叫DTE—DCE接口的每个互换电路的转换。从图中识别出呼叫建立、数据传输和呼叫清除阶段。

8.15 (a)解释术语“ISDN”的含义。

(b)给出如下关于ISDN的用户业务的实例：

(i)提供基本传输功能的承载业务

(ii)用户终端业务

8.16 画图并给出ISDN NTE的相关描述，并在图中说明如下用户访问点：

(a)提供基本传输功能的基本承载业务；

(b)提供基本传输功能的补充承载业务；

(c)用户终端业务；

(d)现有的X系列服务。

8.17 画图说明ISDN用户接口的NTE的概要结构，并解释它的主要特点。

8.18 解释ISDN网络用户访问点的相关术语“控制(C)平面”和“用户(U)平面”的含义。用图说明用于如下信元类型的C平面和U平面的作用。

(a)电路交换

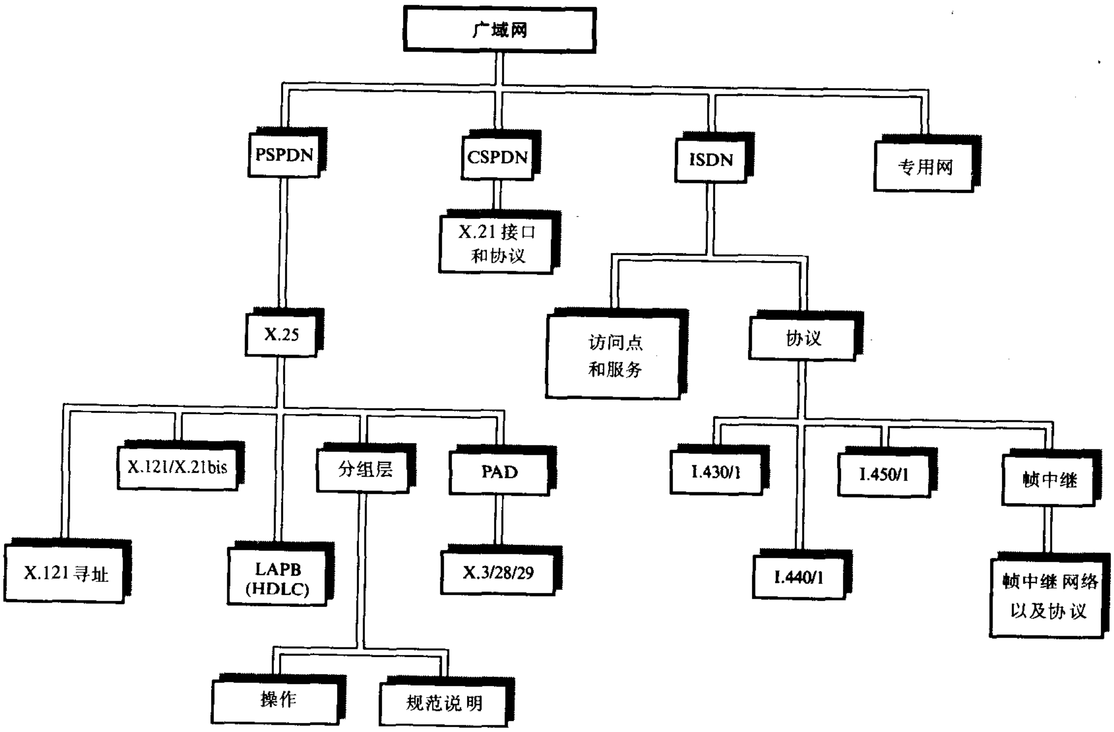
(b)帧中继

(c)分组交换

8.19 使用ISDN电路交换呼叫的建立过程，解释各种在D信道操作的信令协议的功能。

8.20 解释帧中继的操作原理以及它与X.25分组交换的差异。在描述中要包括帧是如何被路由的。 481

本章概要



广域网
PSPDN
CSPDN
ISDN
专用网
X.21 接口和协议
X.25
访问点和服务
协议
X.121/X.21bis
分组层
PAD
I.430/1
I.450/1
帧中继
X.121 寻址
LAPB (HDLC)
X.3/28/29
I.440/1
帧中继网络以及协议
操作
规范说明

482

第9章 网际互连

本章目的

读完本章，应该能够：

- 了解建立由多个互连网络组成的开放系统连网环境所用到的术语以及涉及到的问题；
- 理解建立开放连网环境所需的与网络层相关的三个子层协议的功能；
- 描述互连不同网络类型所需的不同协调功能；
- 解释组成TCP/IP集相关网际互连协议的一些协议的操作；
- 描述下一代IP的主要特点；
- 描述ISO网际互连协议的操作；
- 理解如何在大型开放连网环境进行路由选择，以及使用的两个主要路由选择协议的操作。

引言

在前面关于LAN和WAN的几章中，假定站/DTE（通常称为**端系统**（ES）或**主机**）都连接到一种网络类型上，就是说所有系统都连接到单个LAN（或桥接LAN）或单个WAN。作出这样的假设是为了推迟讨论当两个系统在由两个或多个不同网络类型组成的网络中通信时必须考虑的额外问题。

除了只由单一类型网络（LAN或WAN）组成的开放系统连网环境之外，还存在由网络互连集组成的连网环境。一个实例是LAN分布式群体，每个LAN位于不同的大学并通过跨国家WAN互连，这种已建立的WAN允许连接在不同LAN的ES交换电子邮件或计算机文件。另一个实例是WAN的互连集，使得银行计算机分布式群体能进行资金传输和其他事务。这样的应用还有很多。 483

当应用涉及两个或多个网络时，通常称系统间的工作方式为**网际互连**。用**网际**或**互联网**表示使用的复合网络（例如LAN/WAN/LAN）。互联网的每个组成网络（LAN或WAN）称为**子网**。

互连两个网络的设备在ISO术语中称为**中间系统**（IS）或**网际互连单元**（IWU）。另外，因为IS执行的一个主要功能是路由选择，有时称为**路由器**；或者因为它提供两个网络间的链路而称为**网关**。**协议转换器**是一种连接两个以完全不同协议栈操作的网络的IS，例如ISO栈和与特定制造商相关的专用栈。将会看到，路由器在网络层执行它的路由选择（和其他）功能。在开放系统环境中，更上层协议（从传输层到应用层）在所有ES中都是一样的。路由器和协议转换器的差异显示在图9-1中。

在图9-1(a)中，假定每个网络的类型是不同的，因此路由器有与每个网络端口相关的不同网络协议集。这样从一个网络收到的包（NPDU）先经过处理，然后通过与那个网络相关的协议集向上传递，在中继后，通过不同的与另一个网络相关的协议集向下传递。

相比之下，能在图9-1(b)中看到，协议转换器在应用层之上执行它的中继功能。这是因为两种网络类型除了网络协议的差异，更高的网络无关协议层也是不同的。而且，用于不同专用协议栈的网络协议也是不同的。这意味着还必须使用协议转换器来互连同一网络中的两个

专用系统。这个需求使得通过向开放系统方式转变而获得的优点显得突出：首先连到同一网络的系统能直接通信，其次，如果涉及到两个不同的网络，虽然也可以用路由器来执行中继功能，但是更高层的协议在所有系统中都是一样的。

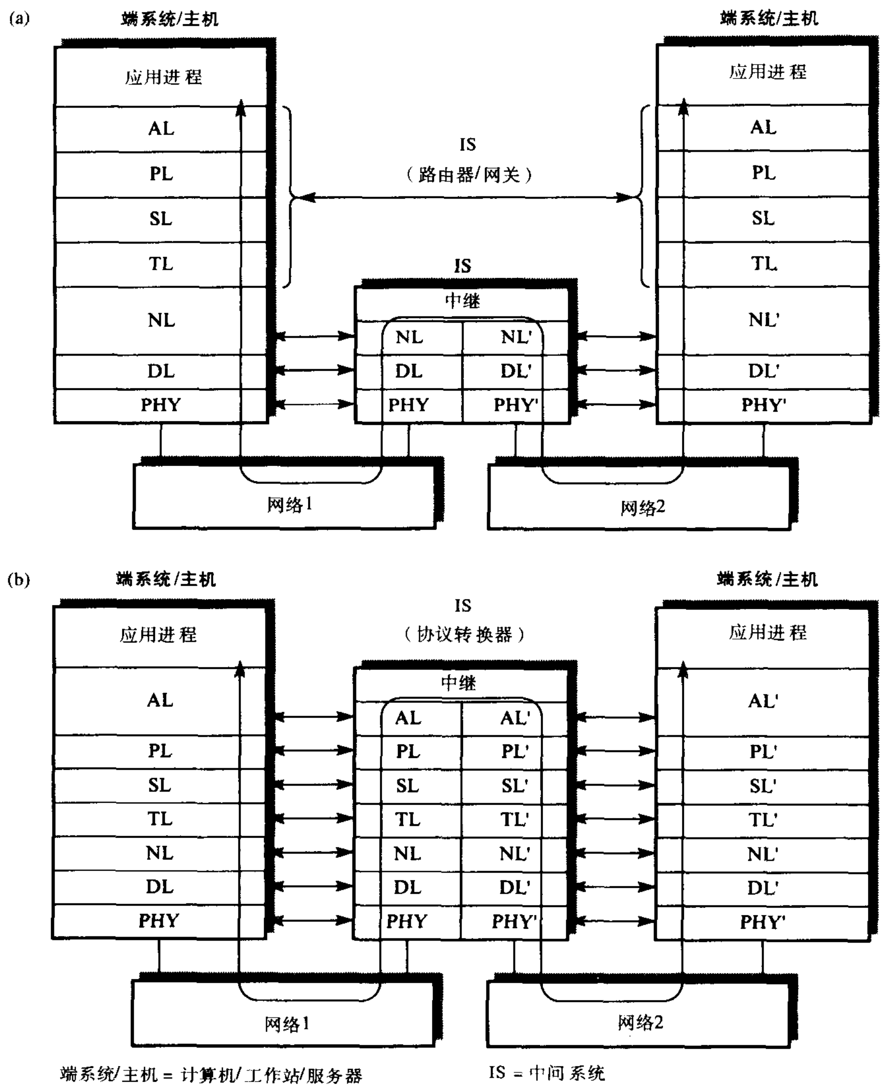


图9-1 IS示例

(a) 路由器/网关 (b) 协议转换器

9.1 网际互连体系结构

在开始讨论网际互连相关的问题前，先考虑一些典型的网际互连体系结构。一些体系结构的实例如图9-2所示。

484

图9-1(a)给出了两个单一网络类型的实例。第一个实例是跨地点LAN，它如第7章中讨论的，一般由一组LAN组成，每个办公室或每幢大楼一个LAN，通过骨干网互连。如果所有的LAN是同一类型，连接每个LAN到骨干网的设备可以是网桥；如果是不同类型，可以是路由器。

第二个实例是单一WAN，例如X.25网络。在这种情况，如第8章中所讨论的，每个分组交换机（DCE/PSE）直接或通过PAD，为它自己的DTE集提供服务，并且PSE通过网状拓扑交换网互连。 483

从网际互连的观点看，与网络的类型（LAN或WAN）无关，可以把每个网络当作有自己内部路由选择协议的单一网络来考虑。两个互联网如图9-2(b)所示，每个由这种网络的链接集组成。在本章中要讨论的是，建立所示类型并且尤其是那些由多种网络组成的连网环境时所产生的额外问题。还要讨论可选解决方案以及相关的网络协议。

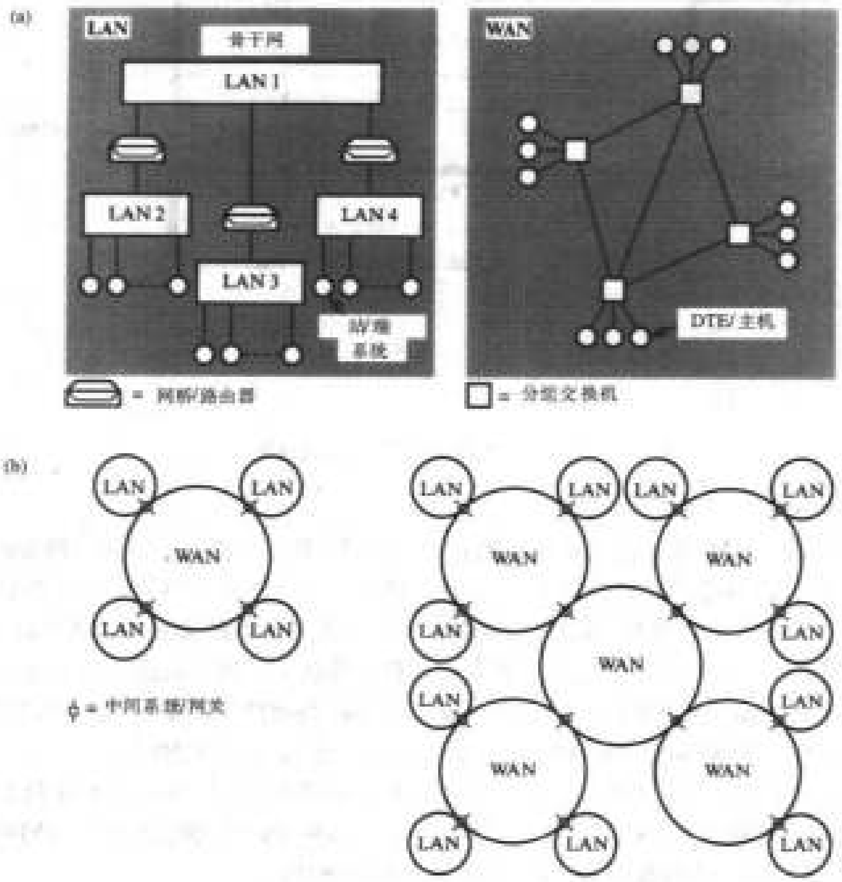


图9-2 互联网体系结构

(a) 单—LAN和单—WAN (b) 互连 LAN/WAN 的实例

9.2 网际互连问题

从互联网用户（实际上是传输协议实体）的观点看，互联网应该在用户网络服务访问点（NSAP）地址中提供规定的网络服务，这使得它能同远端系统中的相似用户通信。可能存在多个网络（可能是多种类型）对于用户是透明的，提供规定的网络服务使得用户简单地把互联网看作是单—LAN或单—WAN。如图9-3所示。 486

在讨论用来达到这个目的的解决方案前，先列出必须考虑的要点：

- 网络服务

- 寻址
- 路由选择
- 服务质量
- 最大包长度
- 流量控制和拥塞控制
- 差错报告

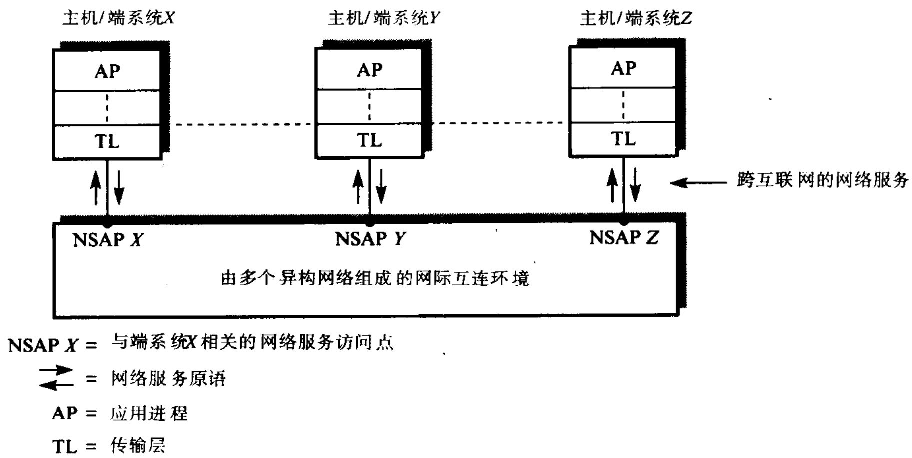


图9-3　互联网网络服务示意图

1. 网络服务

回忆在LAN中，MAC子层地址用来确定ES（站/DTE）以及在使用透明网桥的系统间路由帧。而且，由于LAN的较短传输时延和较低数据误码率，通常使用简单的（最佳尝试）无连接网络协议。这意味着许多基于LAN的网络有相关的**无连接网络服务**（CLNS）。

与LAN相比，大多数WAN的链路层地址只有本地意义；网络层地址用来确定ES并在网络中路由包。还有，由于WAN相对较长的传输时延和较高数据误码率，通常使用更复杂的面向连接协议。这意味着大多数WAN有相关的**面向连接网络服务**（CONS）。 487

既然NS_user可以连到互联网的不同网络，因此必须作的第一个决定是在每个ES的互联网接口使用的网络服务类型（CL或CO）。其次，在由多种子网类型组成的互联网中，必须考虑所选服务如何同组成互联网的不同网络相关的各种服务协调。

2. 寻址

回忆用来确定ES中的NS_user的NSAP地址是惟一的跨网络的地址，它允许用户在整个网络中被惟一确定。这样在单一LAN或WAN中，NSAP地址必须在有限的单一网络寻址域中是惟一的。既然在单一网络中ES的连接点（PA）地址在该网中是惟一的，NS_user的NSAP地址是由该系统的PA地址以及该系统内的LSAP（链路）和NSAP（网络）层间地址选择组合而成的。

对于由多个不同类型网络组成的互联网（例如LAN和X.25 WAN）来说，ES（由此IS）的PA地址的格式（结构）和语法会因网络不同而不同。因为已经存在许多这种网络并且已经给这些网络指定了地址，每个系统的**网络连接点**（NPA）地址不能被用作组合互联网中NS_userNSAP地址的基础。相反，当建立一个**开放系统网际互连环境**（OSIE）时，必须使用完全不同的NSAP地址集来惟一地确定每个NS_user。这些地址独立于NS_user（传输实体）所

在系统的NPA地址。NSAP地址与NPA地址间的联系如图9-4所示。

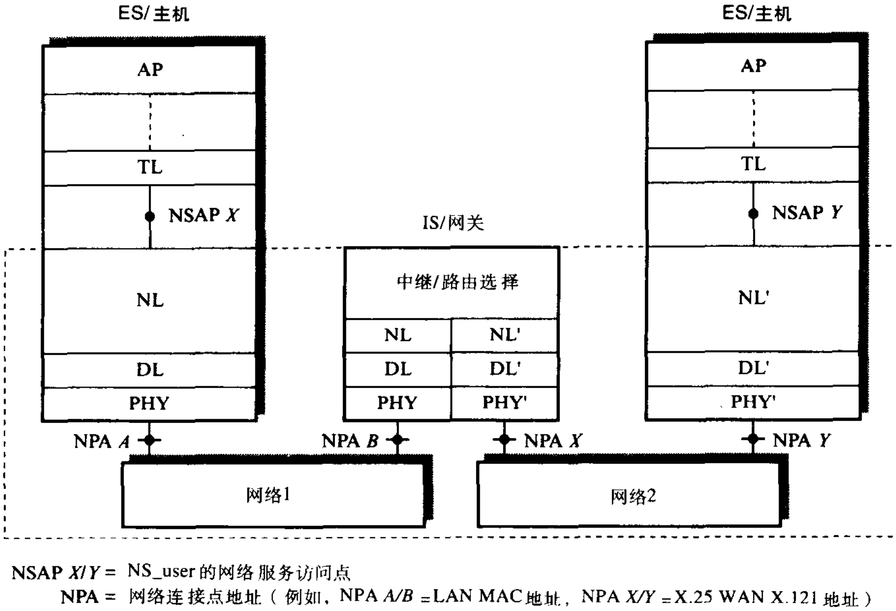


图9-4 NSAP地址和NPA地址之间的关系

能从图中推断出与每个连到互联网的ES相关的两个完全不同的地址：NPA和NSAP。NPA地址能使系统在本地网络中发送和接收NPDU，因此只在那个网络中有意义。但是，它的NSAP地址是跨互联网的标识符，它在整个OSIE中惟一地确定了NS_user。还有，因为一个IS连接到多个网络，它对于连接的每个网络都有相关的NPA地址。

3. 路由选择

ES的NSAP收到的服务请求原语只有所需目标NSAP的说明。对于单一网络，NSAP中的PA子地址足够用于路由由此产生的NPDU到达所需的目标。例如，如果网络是LAN，NPDU使用帧头部的目标MAC地址在LAN中被广播。另一种情况，如果网络是X.25分组交换网络，那么NPDU在它的本地DCE/PSE中被传输到X.25 PLP。从那里使用目标NSAP通过网络把它直 488
接路由到目标DCE（由此到达DTE）。

对于通过IS互连的多个网络组成的互联网，目标NSAP地址不必让连接的ES同始发ES都属于同一个网络。而且，它可以让连接的ES属于互联网上其他的任一网络。显然，NPDU的路由选择在互联网中更加困难。

为了确定路由选择需求，考虑一下图9-5中假设的互联网。首先记住系统的NPA地址与NS_user NSAP地址不一样，不能直接使用目标NSAP地址来路由NPDU到它的目标。还有，某个IS的每一个NPA地址与互连网络上每一个网络的任何其他ES有类似的格式。所以可以假定ES能直接发送NPDU给同一网络中的IS，如果它知道后者的NPA地址。另外，因为IS对于它所连接的每个网络有一个NPA地址，它可以发送NPDU给连接在同一网络中的另一个IS，只要它知道那个IS的NPA地址。

假设这些基本能力，考虑NPDU从网络1上的ES 1.1发送到网络4上的ES 4.1。可能有许多可选的路由，但恐怕最明显的（假定所有的网络有相同的操作参数）是：

ES1.1→IS 1→IS 2→ES 4.1 489

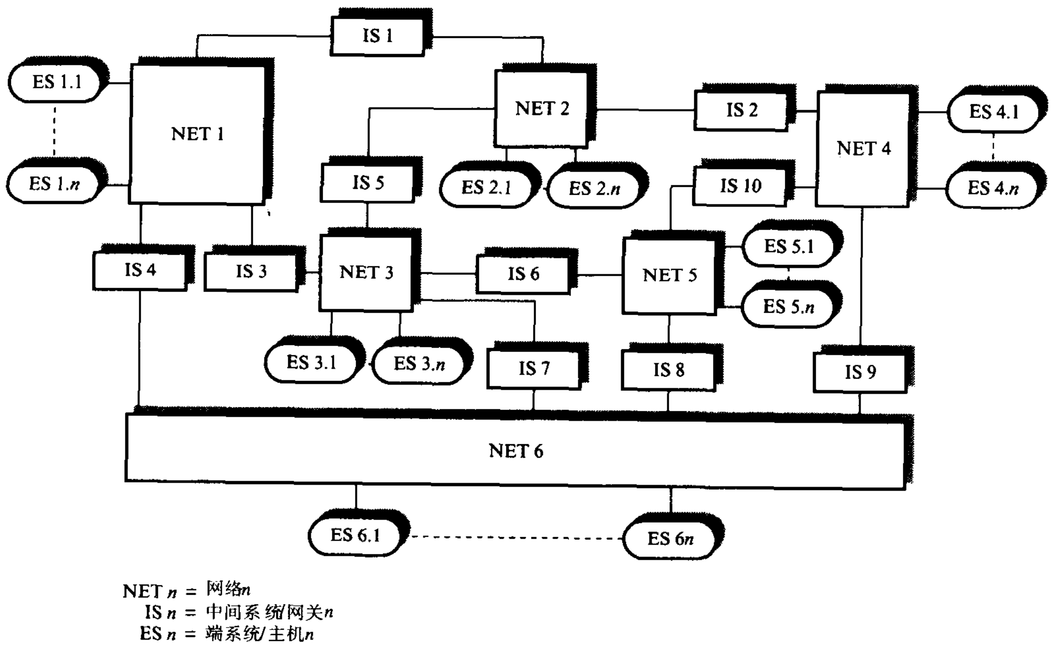


图9-5　假想互联网

其他包括：

ES 1.1 → IS 3 → IS 6 → IS 10 → ES 4.1

ES 1.1 → IS 4 → IS 9 → ES 4.1

ES 1.1 → IS 4 → IS 7 → IS 6 → IS 8 → IS 9 → ES 4.1

虽然只是通过简单地看互联网拓扑推断出这些，但是实际上必须回答许多问题。这些问题包括如下：

- ES如何判断连接在它的网络上IS的NPA地址？
- IS如何判断连接在它的网络上ES的NPA地址？
- 当发送NPDU时ES如何选择特定的IS？
- IS如何判断连接在同一网络上的其他IS的NPA地址？
- IS如何选择特定的IS来路由NPDU到给定的目标ES？

4. 服务质量

服务质量（QOS）参数与在NSAP中收到的每个服务请求原语有关。实际上它是一组参数，
490 共同来说明NS_user期望从与该请求相关的NS_provider得到的网络服务性能。另外，QOS还用来说明在该请求中使用的可选服务。

QOS参数包括如下：网络（NS_provider）发送NSDU到指定目标所期望的传输时延，未授权的监控或修改NSDU所需的保护级别，与该请求相关的成本限制，期望的剩余差错概率，与NSDU相关的相对优先级。

在CONS中，当建立呼叫时对等协商发生在两个NS_user间。始发NS_user指定期望的QOS参数，而响应者（如果需要）可以修改所选的参数。相比之下，在CLNS中，因为没有建立虚拟呼叫，发起请求的NS_user必须知道NS_provider所期望的QOS。

这样当网际互连不同网络类型时，因为QOS可能随着网络变化，所以当给任意指定目标

NSAP发送时，每个ES中的网络实体必须能知道期望的跨互联网QOS。

5. 最大包长度

不同网络中的最大包长度会不同，它由下列因素决定：

- 数据误码率 网络链路的数据误码率越高，最大包长度必须越小才能确保接收的大量包很少有差错。
- 传输时延 最大包长度越长，其他包转发前在每个链路上必须等待的时间越长，由此增加了包传输时延。
- 缓冲存储需求 最大包长度越小，存储所需的内存缓冲区大小越小。
- 处理开销 最大包长度越小，发送每个报文（NSDU）所需的包数量越大，随之每个报文的处理开销增加。

典型的最大包长度从128个字节（用于一些公共载波网）到8000个字节（或更多）（用于一些LAN）变化。

在单一网络中，最大包长度通常是已知的，因此传输层协议实体自己就能把较大的报文分成（**分段或分割**）较小的单元以便在网络中传输。但是在由不同最大包长度网络组成的互联网中，要么最小包长度必须是已知的和约定的，要么每个ES和IS中的网络层必须执行必要的分段（分割）和重装操作。第一种方案导致一些网络效率不高，而第二种方案需要网络层执行额外的功能。 491

6. 流量控制和拥塞控制

流量控制是为了克服发送包的源系统速率和接收包的目标系统速率之间的差异而对涉及单个呼叫的包流量进行的控制。如果目标系统接收包快于源系统发送包，显然没有问题。但是，如果是相反的情况就必须提供协调（流量控制）功能。

拥塞控制关心的是网络自身的类似功能。如果包进入网络的综合速率超过了包离开的速率，那么网络就变得拥塞。类似地，在更本地化的级别，如果包到达一个网络结点（比如某个IS）快于处理和转发，那么结点就变得拥塞，这样影响了涉及通过该结点所有呼叫的包流量。

在诸如X.25的面向连接的网络，流量控制在通过本地DTE－DCE和 DCE－DTE接口的VC基础上执行。定义了一个发送窗口，并且当这个数量（等于发送窗口）的包被发送（一般两个）后发送方必须等待，直到收到与任一个包相关的确认。因为在网络外设执行基于每个呼叫的流量控制，因此除了调整进入网络的包流量外，它还帮助控制了拥塞。但是它不能完全避免拥塞。

相比之下，在无连接网络中没有应用流量控制到与网络中呼叫相关的包上。相反，它把端对端基础上的流量控制留给每个ES中的传输协议实体来执行。如果在网络中发生拥塞，流量控制信息被延迟，源传输协议实体停止向网络发送新的数据。此外，虽然发送新数据像面向连接网络中一样帮助减轻了网络拥塞，但是它不能永远避免拥塞。所以，必须结合两种方案产生一种网络中的拥塞控制算法。而且，对于由多种网络类型组成的互联网，拥塞控制算法必须能协调不同的网络算法。

7. 差错报告

差错报告的方式随着网络类型的变化而变化。因此，必须建立一种适用网络的差错报告方法。

所有这些问题必须由网际互连方案解决。

9.3 网络层结构

每个ES的网络层的作用就是为本地NS_user提供端对端、跨互联网的网络服务。它可以是CONS或者CLNS。在两种情况中，NS_user应该不知道存在多个网络（可能网络类型不同）。
492 由此涉及NSDU中继的路由选择和其他所有功能必须由每个端系统和中间系统中的网络层实体以透明的方式执行。

为了达到这个目标，在ISO参考模型关系中每个ES和IS中的网络层不只由一个单一协议组成而是由三个（子层）协议组成，每个协议在提供网络层服务时扮演互补的角色。在ISO术语中，每个组成互联网的网络称为子网，由此三个协议称为如下：

- 子网无关会聚协议（SNICP）
- 子网有关会聚协议（SNDCP）
- 子网有关访问协议（SNDAP）

493 三个协议在ES中的相对位置如图9-6(a)所示，图9-6(b)给出IS的相关协议。

SNICP支持提供给互联网接口NS_user的网络服务。它的作用是执行各种协调（会聚）功能，这些功能可能需要经过互联网路由和中继用户数据（传输协议数据单元）。它的操作独立于互联网中特定子网（网络）的特点，并且假设这些子网（网络）提供的是标准网络服务。

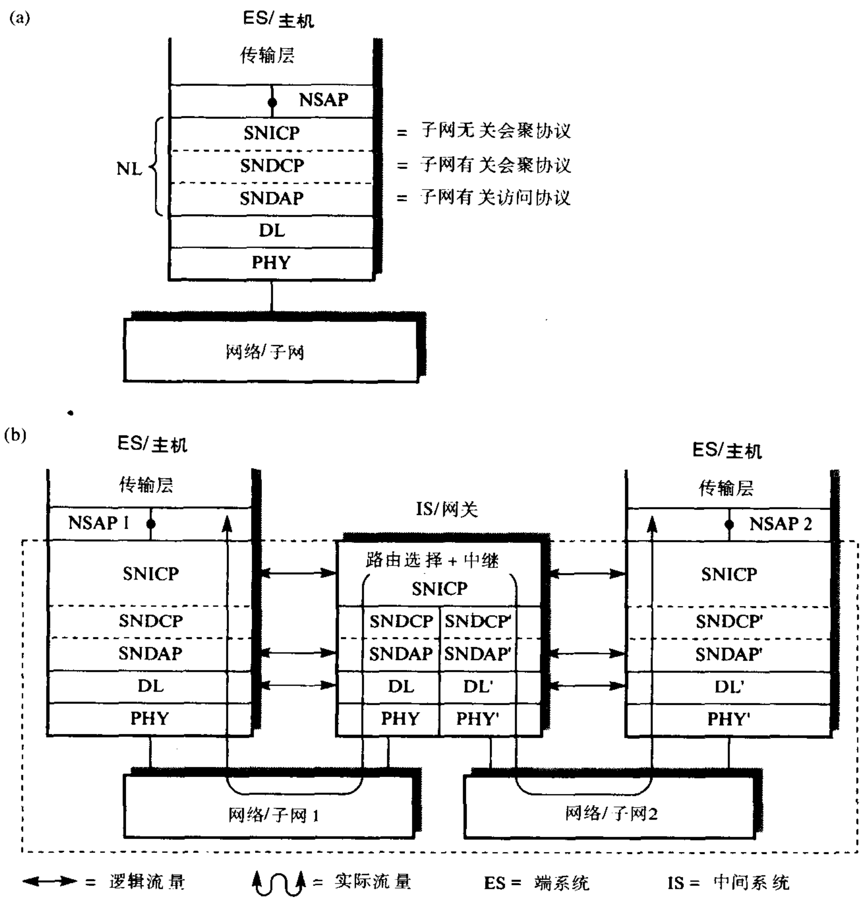


图9-6 网络层结构

(a) 子层协议 (b) IS结构

SNDAP是互联网中与特定子网（网络）相关的访问协议。例如用于X.25 网络的X.25数据包层协议和一般用于LAN的无连接网络协议。因为与SNDAP相关的服务和操作特点会随网络类型不同，因此必须在SNICP和SNDAP间提供中间子层。这就是SNDCP的角色。显然，它执行的具体映射操作会因子网/网络类型不同而不同。

9.4 互联网协议标准

在第8章中讨论过，多个X.25 WAN可以通过基于X.75的网关互连。对用于LAN的X.25数据包层协议操作的标准说明的介绍，意味着网际互连的一种方法是采用X.25作为跨互联网协议。通过使用快速选择，后者可以工作在面向连接模式或者无连接模式下。

这个解决方案使得各种网际互连功能大大减少。缺点是与X.25分组交换相关的开销较高并且因此这些网络的包吞吐量较低。这在快速选择情况下也是存在的，因为使用同样的VC/差错控制功能。而且，下一代WAN（比如ISDN）大大改进的数据误码率性能意味着帧中继和信元（快速包）交换会成为优于传统分组交换的操作模式。

ISO采用的解决方案基于无连接互联网服务和相关的无连接SNICP。在ISO 8475定义了SNICP。它基于**美国国防部高级研究规划局**（DARPA）资助的作为网际互连研究一部分开发的互联网协议。早期的DARPA互联网（ARPANET）用来把少量研究场所和大学的计算机网络与DARPA互连起来。当它在20世纪70年代早期出现时，ARPANET只含有少量的网络和相关主机计算机。从那时起，互联网稳定地发展。与每个地点只有少量大型机不同，现在有大量的工作站。而且，LAN的引入意味着现在有几千个网络/子网。ARPANET现在与其他互联网相连。组合后的网络，由许多机构共同资助，被简单地称为**因特网**。 494

互联网协议只是因特网的完整协议集（栈）的一个协议。完整的协议集称为TCP/IP，包括现在用作许多其他商业网络和研究网络基础的传输协议和应用协议。TCP/IP的所有规范都是公开的，因此因特网是到目前为止基于开放标准的最大的运行互联网。将在本章中讨论的两个协议，是与因特网相关的互联网协议和ISO 因特网协议（它计划用于OSI栈），分别称为**因特网IP**或简称**IP**以及**ISO-IP**或**ISO CLNP**。两个标准的主要方式如图9-7所示。

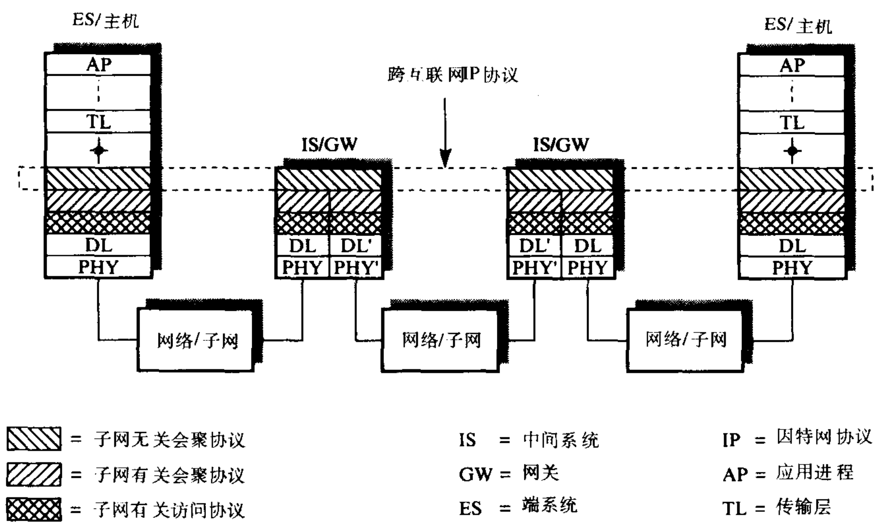


图9-7 跨互联网IP示意图

IP是跨互联网协议，它使得位于不同ES/主机的两个传输协议实体能以透明的方式交换报文单元（NSDU）。这意味着多个（可能类型不同）网络/子网和IS/网关的存在对于两个通信传输实体是完全透明的。因为IP是无连接协议，报文单元使用无确认最佳尝试方式传输。

虽然关于ISO CLNP的操作特点基于从IP演变和使用中获得的经验，但是在术语和操作细节间有差异。因此会单独讨论每个协议。

9.5 因特网IP

除了因特网使用TCP/IP外，现在还有许多商业互联网和研究互联网广泛使用TCP/IP。但是，几乎所有TCP/IP协议是作为因特网的一部分被研究和制定的。实际上，随着组合因特网
495 的相关研究继续深入，新的协议也会相对频繁地被引入。但是，有一个协议核心集，它形成了TCP/IP实现的完整部分。其他可选协议用于不同规模以及不同复杂程度的开放系统。我们只考虑核心协议。

9.5.1 地址结构

回忆一下，连接到互联网的主机/ES有两种网络地址。在ISO术语中，有网络服务访问点（NSAP）地址和子网连接点（SNPA）地址。在TCP/IP中，分别有IP地址和网络连接点（NPA）地址。NPA地址对于每种网络/子网类型是不同的，而IP地址是惟一跨网络标识符。IP地址的结构如图9-8所示。为了使建立互联网的权威机构在分配地址时有一些灵活性，采用了如图9-8(a)所示的地址结构。

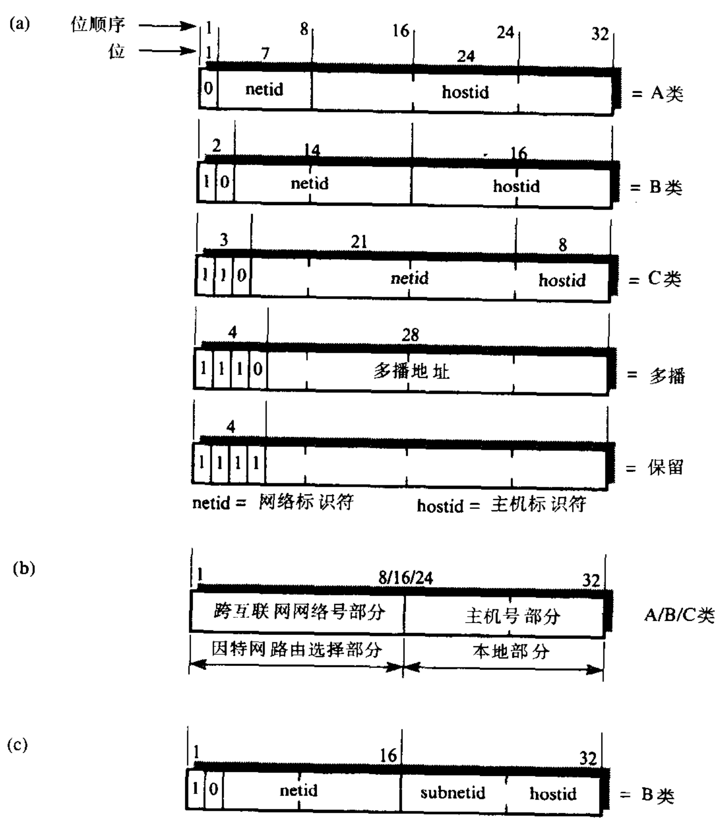


图9-8　IP地址格式

(a) 帧　(b) 子网寻址　(c) 修改的B类地址

为了确保所有主机有惟一标识符，每个IP地址使用32位整数。然后定义3类地址格式以满足主机连接的不同规模网络。每种格式称为一个**地址类**，单个互联网可以使用所有的类。三个主要类是A、B和C，每个地址类用于不同规模的网络。地址所属的类由前四位中第一个0位的位置决定。剩下的位指定两个子字段——网络号（netid）和主机号（hostid）。子字段的边界位于字节边界以便于解码。 496

A类地址有7位用于网络号，24位用于主机号。B类地址有14位用于网络号，16位用于主机号。而C类地址有21位用于网络号，8位用于主机号。A类地址用于连接有大量主机（最多可达2^{24}）的网络而C类地址用于大量只连接少数主机（最多达256个）的网络。A类网络的实例是ARPANET。C类网络的实例是单个跨地点LAN。

主机号是0的地址代表网络号字段的网络而不是主机。类似地，主机号全为1的地址代表连接在网络号字段网络的所有主机，如果网络号字段还是全为1，那么地址代表互联网中的所有主机。这些地址用于广播目的。

为了使IP地址更简洁以及更容易阅读，32位分成4个字节。它们转换成等价的中间由点隔开的十进制数形式。它称为**点分十进制**。实例地址如下：

00001010 00000000 00000000 00000000 = 10.0.0.0= A 类地址
= 网络号 10（ARPANET）
10000000 00000011 00000010 00000011 = 128.3.2.3=B 类地址
= 网络号128.3，主机号 2.3
11000000 00000000 00000001 11111111 = 192.0.1.255 =C 类地址
= 网络号192.0.1广播地址

D类地址被保留用于多播。在LAN中，帧可以发送给单一地址、广播地址和**组地址**。最后一个允许一组主机（例如工作站）以某种方式协作组织网络传输发给组内的所有成员。这一般称为**计算机支持的协同工作**（CSCW），D类地址允许以这种工作模式在互联网中传送。

虽然这个基本结构对于大多数寻址目的是足够了，但是每个地点上多个LAN的引入意味着路由选择相关的开销变得难以接受的高。正如第7章所描述的，MAC网桥通常用于互连同一类型的LAN。这种解决方案对于路由选择是有吸引力的，因为组合LAN能像单一网络一样工作。当互连不同LAN类型时，帧格式差异以及更重要的帧长度差异意味着通常需要使用路由器，因为包/帧的分段和重装是网络层的功能而不是MAC子层的功能。但是，路由器的使用意味着LAN必须有自己的网络号。在大型地点的情况下，可能有大量的这种LAN。 497

这意味着使用基本寻址方案每个地点的所有路由器需要参与到整个互联网的路由选择功能。任何路由选择方案的效率受到组成互联网的路由选择结点数量的强烈影响。已经引入子网的概念来把有关单一地点的路由器（由此路由选择）从整个互联网路由选择功能中解脱出来。基本上，与地点中每个LAN都有自己的网络号不同，地点只分配一个互联网网络号。然后每个LAN的标识就成为主机号字段的一部分。修改后的地址格式显示在图9-8(b)中。

虽然使用相同的地址类及其格式，但是现在网络号是关于一个完整地点而不是单个网络。由此，因为只有连接在本地网络的单一网关执行跨互联网路由选择，网络号被认为是**互联网部分**。对于一个有许多相关子网的单一网络号，主机号部分由两个子字段组成：**子网号部分**和**本地主机号部分**。因为它们只有本地意义，它们总称为**本地部分**。

由于可能存在大量子网与不同地点网络相关，没有为本地地址部分规定严格的子地址边界。相反，**地址掩码**用来为特定网络（由此网络号）规定子地址边界。地址掩码由那个地点

的互联网网关和路由器保存。那些包含网络地址（包括网络号和子网号）的位为1并且那些包含主机号的位为0形成了地址掩码。由此地址掩码

11111111 11111111 11111111 00000000

表示前三个字节含有网络/子网号而第四个字节含有主机号。

例如，如果地址是个B类地址（0位在第二个位的位置），它能轻易地解释为：前两个字节是跨互联网网络号，下一个字节是子网号而最后一个字节是该子网内的主机号。这样的一个地址如图9-8(c)所示。

点分十进制通常用来定义地址掩码，上面的掩码写为

255.255.255.0

通常选择字节边界是为了简化地址解码。由此在这个掩码中，假定网络号是128.10，那么所有连接在这个网络上的主机会有相同的互联网路由选择部分。这样，可能有大量子网和相关路由器对用于路由选择目的的互联网网关是透明的。

为了确保IP地址是惟一的，它们必须由建立开放系统环境的中央授权机构分配。对于小
498 互联网，这相对比较简单。但是，对于诸如因特网的大型互联网，这通常分两步完成。首先，建立中央授权机构分配网络号和多播地址。其次，每个网络的相关授权机构在那个网络上分配主机号。因特网的中央授权机构称为**网络信息中心**（NIC）。

9.5.2 数据报

在考虑各种IP的相关功能和协议前，描述一下IP数据单元的格式。它称为**数据报**。数据报的格式和内容如图9-9所示。

版本字段含有生成数据报的IP版本，确保在数据报经过互联网传输期间处理它的所有其他系统（网关和主机）能正确地解释各个字段。当前的版本号是4，称为**IP 版本 4** 或简称为**IPv4**。

头部的长度是可变的。报文**头部长度**以32位码字的倍数说明数据报的实际长度。最小长度（无选项）是5。如果数据报有可选项，它们必须是32位的倍数。任何未用字节必须以**填充**
499 字节填满。

服务类型与ISO网络中QOS参数的作用一样。它允许应用进程指定与路由相关的首选属性，由每个网关在路由选择期间使用。例如，如果一个可靠的传送服务更倾向于最佳尝试传输，那么假设可选择的话，网关应该选择面向连接的网络而不是无连接网络。**总长度**定义了包含报文头和用户数据部分的数据报的总长度。最大长度是65 536个字节。

将在9.5.4节中解释，用户报文可以在多个数据报中通过互联网传输，目标主机使用**标识**字段把同一用户报文的各个不同数据报关联起来。

下三位称为**标记位**，现在使用其中的两个。第一个称为**不分段位**或**D位**，由中间网关使用。D位设成1，说明应该选择能把数据报作为单一实体而不是多个称为**段**的较小数据报来处理的网络。由此如果目标主机连到该网络（或子网），它会收到单一数据报形式的用户数据或者什么也收不到。所以用户数据的传输时延可以被更准确地量化。

第二个标识位称为**更多段位**或**M位**，也用在涉及多个数据报与用户数据传输相关的重装规程。**段偏移**也用于相同的规程，用来指明数据报中（数据）内容相对初始用户数据报文的位置。将在9.5.4节中描述重装规程。

生存时间值定义了数据报能在互联网中传送的最长时间。该值以秒计，由源IP设置。然后由每个网关以规定量递减。如果该值变为0，数据报就被丢弃。这个规程允许目标IP在重装

规程中等待数据报段一个已知的最长时间。它还能丢弃循环的数据报。

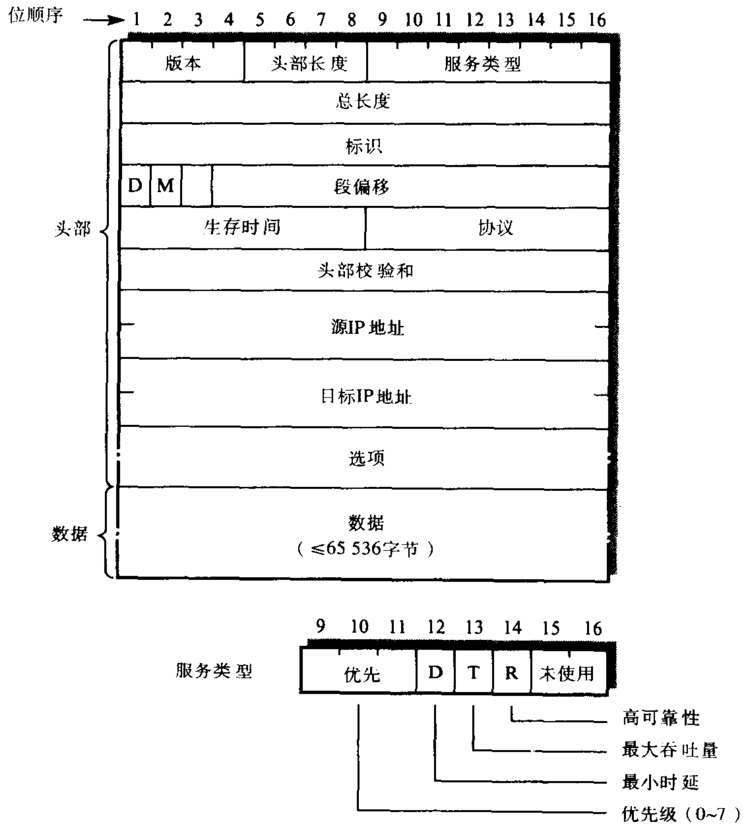


图9-9 因特网数据报格式及内容

TCP/IP协议族含有多个协议。因此**协议**字段用来使目标IP把数据报传递给所需的协议。

报文头校验和只应用在数据报的报文头部分，它是防止已被破坏的数据报路由到不正确目标的安全措施。通过把每16位字段看作一个整数，并使用反码运算求和（循环进位）算术来把它们加在一起。然后校验和是总和取反。

源地址和目标地址是源主机和目标主机的跨互联网IP（NSAP）地址。

最后，**可选**字段由数据报用来装载与下面有关的额外信息：

- 保密性　例如数据字段可以被加密或者只有特定用户组可接收。
- 源路由选择　如果源路由可知，在互联网中经过的实际路由可以在这个字段以网关地址表说明。 500
- 路由记录　该字段在互联网中数据报的传递期间由每个访问的网关用来记录该网关的地址。由此得到的地址表可以用在随后数据报的源路由选择字段。
- 流标识　它使得源主机能说明装在数据报中的数据类型，如果它不是计算机数据，比如语音采样。
- 时间戳　如果存在这个选项，数据报经过路径上的每个网关用它来记录处理数据报的时间。

9.5.3 协议功能

IP协议提供许多核心功能和相关规程来执行不同类型网络交互工作时的各种协调功能。

它们包括如下：

- 分段/重装 它关心用户数据报文在支持比用户数据更小包类型的网络中的传输。
- 路由选择 为了执行路由选择功能，每个源主机的IP必须知道互联网网关的位置或者连接在同一网络或子网的本地路由器的位置。还有，每个网关的IP必须知道到达其他网络或子网经过的路由。
- 差错报告 当在一个主机或网关中路由选择或重装数据报时，IP可能丢弃一些数据报。这个功能关心这类事件返回给源主机中的IP的报告以及许多其他报告功能。

下面将分别讨论这些功能。

9.5.4 分段/重装

与NS_user请求相关的用户数据（通常称为NSDU）的长度可达64K或65 536个字节。各种类型网络的最大包长度比它短得多，范围从用于X.25分组交换网的128个字节到用于LAN的超过8000个字节。关于IP的分段和重装功能把与NS_user请求相关的NSDU分成更小的段（ISO术语），这样它们可以以适当长度的数据报在特定网络中传输。接收到与每个IP数据报所含相同NSDU相关的数据段，IP在把它传递给目标NS_user前会重装NSDU。

501 因为最大包长度会随网络而变化，可以采用两种方法。在每个网络基础上（**网内分段**）和端对端（跨互联网）基础上（**互联网分段**）执行分段和重装功能。这两种方法分别如图9-10(a)和图9-10(b)所示。

一般来讲，主机内的IP只知道它的本地网络的最大包长度。类似地，每个网关中的IP只知道与它连接的两个网络的最大包长度。在网内分段中，源主机中的IP先把NS_user数据
502 （NSDU）分段成许多由该主机连接网络规定的单独标明地址的数据报。然后使用SNDCP把它们发送给目标主机或者路由上的第一个IS（网关），SNDCP包含主机或网关的NPA地址。

将在9.5.5节中讨论它通过哪种方式获得NPA地址。接到每个数据报，主机或网关中的IP重装NSDU。然后把重装的NSDU重新分段成由第二个网络规定的最大包长度（可能不同）指示的单独标明地址的数据报组。

每个网关重复这个规程直到数据报到达目标主机的IP，在那里NSDU被重装并交付给目标NS_user。

在互联网分段中，源主机中的IP执行同前面一样的分段规程并把由此得到的数据报发送给第一个网关中的IP。但是，这次IP不重装NSDU。相反，它修改相应字段并把收到的数据报直接发送给第二个网络（如果后者能支持这个数据报长度）或者把数据报重新分段成更小的段（数据报）。在图9-10中，假定第二个网络/子网的最大包长度比第一个网络使用的更小。因此，IP会把收到的每个数据报分段成许多更小的数据报，每个都有相同的源地址和目标地址。

下一个网关重复这个规程。但是，因为在图9-10中最后一个网络/子网支持比收到的数据报更大的长度，接收到的数据报直接发送，只在一些报文头字段做了适当的修改。同前面一样，目标主机中的IP把收到的每个数据报重装成用户数据并把由此得到的NSDU传递给目标NS_user。

可以推断出（尤其从图9-10中第三个网络的包流量）网内分段允许使用每个网络的最大包长度，因为单独段被路由中的每个网关重装。虽然在互联网分段中不必这样，但是它有优势，就是在每个网关不必进行重装处理。

实际上IP的确使用互联网分段。由于没有数据报丢失的问题所以使用它，起初可能会觉得意外。一些网络以最佳尝试无连接协议工作，可能在传输过程中一个或多个与某个NSDU相

关的数据报被破坏。已经看到，在网内分段中每个网关中的接收IP在把它中继到下一个网络前重装完整的NSDU。如果有段丢失（例如数据报由于已经被破坏而被丢弃），接收IP必须决定何时放弃重装功能。

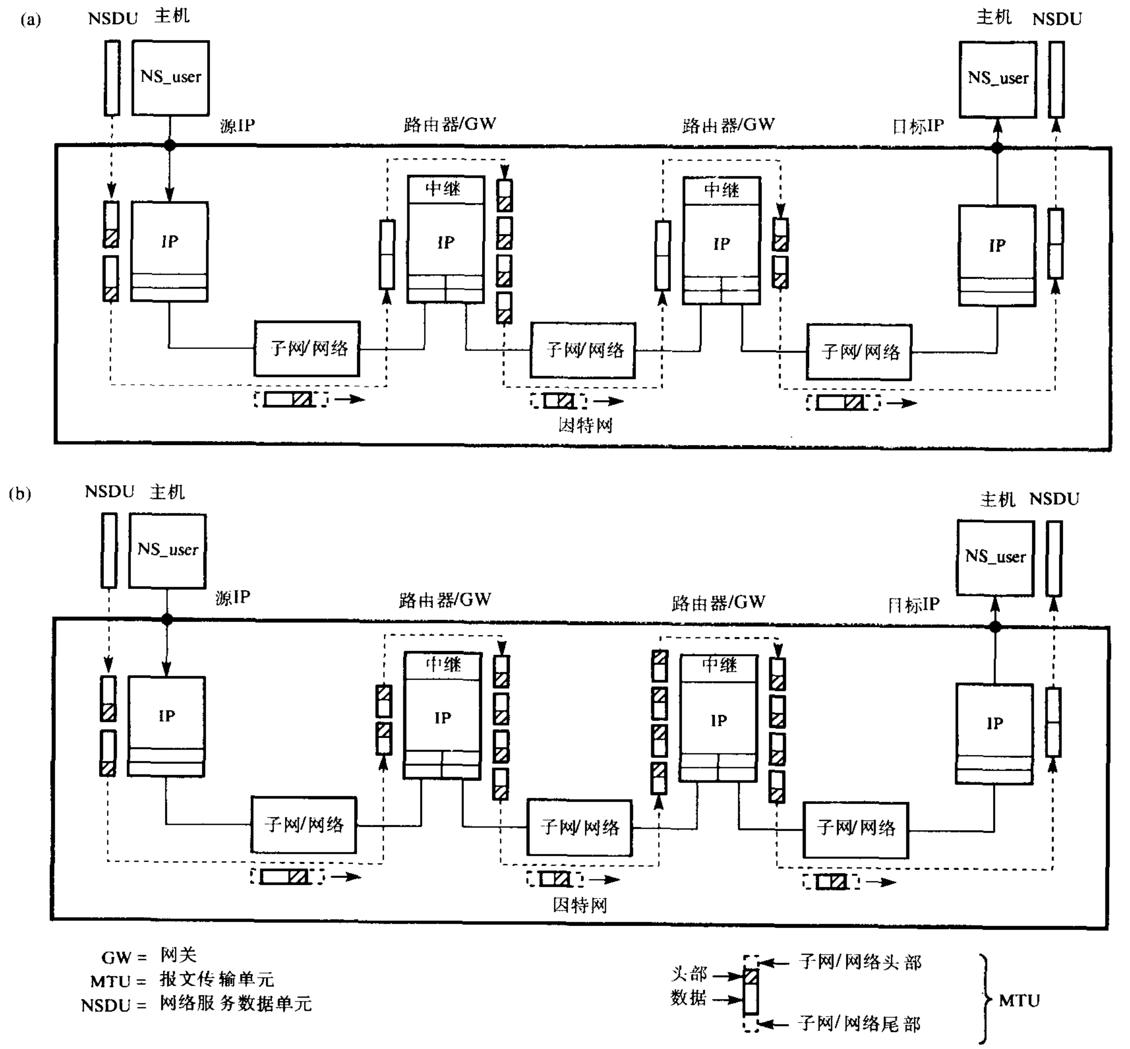


图9-10 分段的两种可选方案

(a) 网内分段 (b) 互联网分段

为了判断这一点，源主机中的IP规定了网关在重装过程中等待与NSDU相关数据报的最大时间限制。这个时限被称为生存时间，放在与该NSDU相关的所有数据报的报头。它由源主机的IP设置，然后由处理数据报的每个IP递减。如果一个数据报被分段，它的当前值被复制到新数据报的报头。如果它在网关（或主机）的重装处理期间任一时刻达到0，重装功能就被放弃，与那个NSDU相关的所有段被丢弃。 503

每个数据报的生存时间以1秒为单位，因此被每个IP递减的量依赖相关网络的（已知）平均传输时延。在互联网分段的情况下，每个网关中的IP仍然减少收到的每个数据报中的生存时间周期字段并且丢弃该值为0的数据报。目标主机中的IP以相同方式放弃重装功能。在两种

情况中，如果段丢失以及重装功能被放弃，就产生**超时**差错报文并且把它返回到源主机的IP。

实例9-1

一个1000字节长的NSDU要在支持最大NS_user数据长度为256个字节的网络中传输。假定每个IP数据报的报头需要20个字节，求所需的数据报（段）的数量以及每个数据报报头中随后字段的内容：

- 标识
- 总长度
- 段偏移
- 更多帧标记

解：

每个数据报最大可用数据长度 = 256 - 20 = 236个字节

假定可用数据，比如，29 × 8 = 232个字节

由此需要五个数据报，四个232字节的用户数据和一个72字节的用户数据。

字段如下：

标识	20（假定）	20	20	20	20
总长度	252	252	252	252	92
段偏移	0	29	58	87	116
更多段标记	1	1	1	1	0

9.5.5　路由选择

因为互联网中的每个网络（或子网）可以使用不同类型的连接点地址，因此连到某个网络中的系统（主机或网关）只能直接发送数据报给连到同一个网络中的其他系统。为了路由
504 数据报通过多个网络，每个网际网关中的IP必须知道目标主机的连接点地址（如果它与该网关同在一网络上）或者知道路由到所需目标网络的下一个网关（不在同一网络）的连接点地址。并且，下一个网关必须连到该网关连到的网络上。路由选择的主要问题是互联网中的主机和网关如何获得和维护它们的路由选择信息。

两种基本方法用于互联网的路由选择：集中式和分布式。在**集中式路由选择**方案中，每个网关的相关路由选择信息使用网络管理报文和特殊网络管理报文从中央地点下载。网络管理系统致力于及时维护它们的内容，当网络和主机加入和除去时，以及差错被诊断和修改时。一般来讲，除了最小的互联网，其他的互联网只有当每个单独网络自身具有结合复杂配置和差错管理规程的网络管理系统时，这才是可行的解决方案。

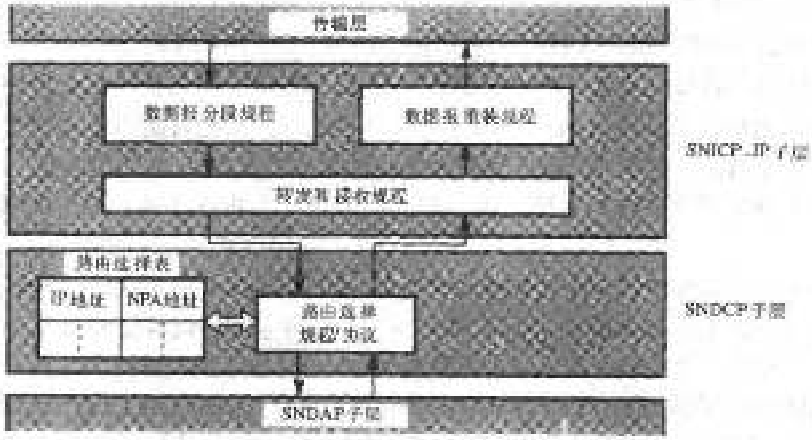
在**分布式路由选择**方案中，所有的主机和网关以分布的方式合作来确保每个系统（主机和网关）拥有的路由选择信息是最新的并且一致。路由选择信息以**路由选择表**的形式由每个系统保存，路由选择表含有用来转发每个数据报的NPA地址。互联网使用这种方案。

IP的路由选择规程先从数据报中读取目标IP（NSAP）地址，然后用它从路由选择表中找到相应的连接点地址（主机或网关）。另外，使用一组路由选择协议以分布的方式产生和维护
505 每张路由选择表的内容。在主机IP中使用的一般方案如图9-11所示。

1. 自治系统

在讨论与因特网相关的各种路由选择协议前，先看一下它的体系结构和相关术语。为了反映因特网是由许多单独管理和运行的互联网组成的事实，每个互联网看作是有自己内部路由选择算法和管理机构的**自治系统**。组合后的因特网被认为是连接了许多自治系统的**核心骨**

干网。一般体系结构以及一些（十分简化的）自治系统拓扑如图9-12所示。



SNICP = 子网无关会聚协议
SNDCP = 子网有关会聚协议
SNDAP = 子网有关访问协议
NPA = （子）网连接点（地址）

图9-11 主机内的一般路由选择方案

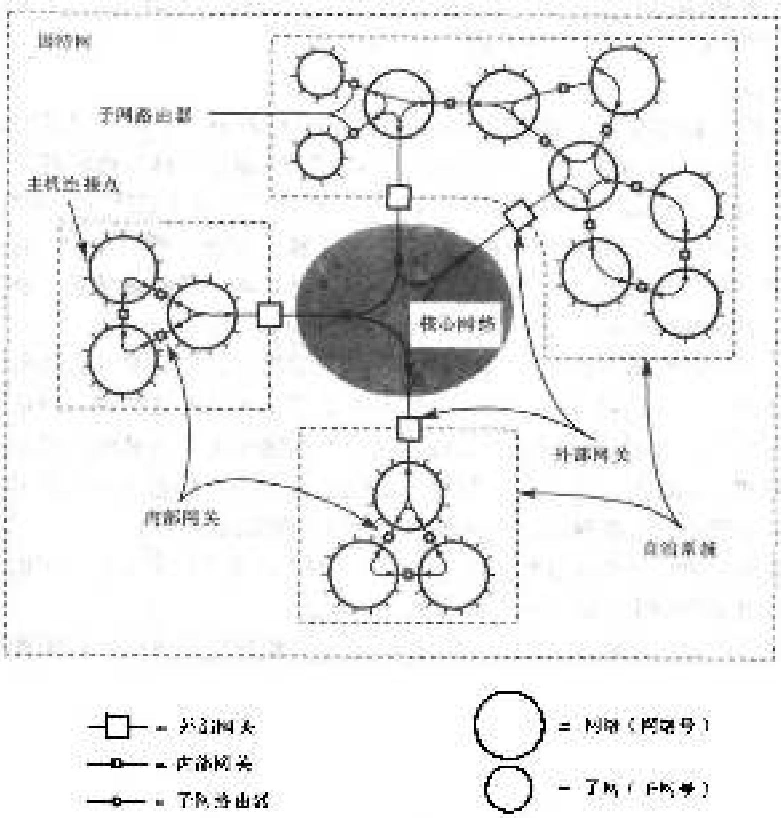


图9-12 一般互联网体系结构和术语

为了区别用在自治系统内的网关和那些用于连接自治系统到核心网络的网关，分别使用

术语**内部网关**和**外部网关**。相应的路由选择协议是**内部网关协议**（IGP）和**外部网关协议**

506 （EGP）。因为因特网由一组互连的互联网组成，每个互联网经过较长时间的发展，因此每个自治系统有自己的IGP。但是因特网EGP实际上必须是一个跨互联网标准。

虽然因特网内的每个自治系统由大量的网络/子网组成，但是在一般应用中自治系统可能只由一个网络组成，并且该网络由一家公司管理和运行。其他可能由使用单个外部网关连到一个跨地点骨干网的一组子网组成。例如一个地点使用路由器互连多个LAN。为了简化讨论，只考虑由多个网络组成的自治系统，因为子网的出现只是在内部网关和主机间增加了一层路由选择。

如果互联网中的每个网关和主机系统在它的路由选择表中为其他所有系统保存单独的记录，路由选择表的大小以及维护路由选择表所需的处理量和传输容量会过大并且因特网会变得无法管理。相反，总的路由选择信息被分级组织如下：

- 主机维护足够的路由选择信息用于转发数据报到连接在同一网络上的其他主机和内部网关。
- 内部网关维护足够的路由选择信息用于转发数据报到位于同一自治系统内的主机或其他内部网关。
- 外部网关维护足够的路由选择信息用于转发数据报到内部网关（如果数据报要发到同一自治系统）或者另一个外部网关（如果不是）。

已经开发了许多实现这个方案的路由选择协议。它们包括一个称为**地址解析协议**（ARP）的网内协议，许多内部网关协议（IGP）和一个外部网关协议（EGP）。每个协议的范围和相关路由选择表如图9-13所示。我们分别讨论每个协议。

2. 地址解析协议

为了使内部网关能把收到的数据报转发到连接在它的其中一个本地网络中的主机，它必须保存连接在这些网络中的所有主机的主机号和相应NPA地址（称为**地址对**）的记录。为了获得这个信息，每个主机通过发送它的IP/NPA地址对简单地通知它所在的本地网关。一般，它存储在主机的永久存储空间（比如硬盘），然后被广播。在非广播网络中，它的本地网关的地址对也被存储并被直接使用。结果，每个内部网关建立**本地路由选择表**，表中含有连接在内部网关上的所有网络所连接的所有主机的IP/NPA地址对。

当某个主机想发送数据报给同一网络上的另一个主机时，IP简单地把数据报发送给本地网关来转发。虽然对于发送给其他网络上主机的数据报来说必须这样做，但是对于连接在同

507 一网络上的主机它会引起过高的开销，尤其当网络中连接了大量主机的情况下。为了解决这个问题，每个主机上的IP努力得到同一网络上要与之通信的所有主机的主机号/NPA地址对。

508 这使得一个主机能直接发送数据报给同一网络中的主机而无需涉及网关。

执行这个功能的协议是地址解析协议（ARP）。ARP形成了每个主机中IP的主要部分，在每个内部网关中有对等ARP，如图9-13(a)所示。

无论何时IP分段规程要转发产生的数据报，它先把数据报存储的内存缓冲地址指针传递给ARP。ARP维护着一张本地路由选择表，它含有连到该网络的与主机通信的所有主机的主机号/NPA地址对。如果数据报中的目标IP地址在表中，那么ARP简单地把数据报地址指针以及相应NPA地址传递给SNPAD协议，IP地址的网络号字段设成0来标识本网络。然后SNDAP通过广播或直接启动数据报的发送。

如果NPA地址不存在，ARP通过产生和发送一个**ARP请求报文**然后等待应答来努力找到它。请求报文含有它自己的IP/NPA地址对和所需的（目标）IP地址。再者，它可以广播（这

种情况下它会被所有主机的ARP接收）或者使用网关（已知）NPA地址直接发送给该网关中的ARP。在第二种情况中，网关中的ARP使用它自己的本地路由选择表以及请求报文中所需的目标IP地址来简单地中继报文给所需的主机。

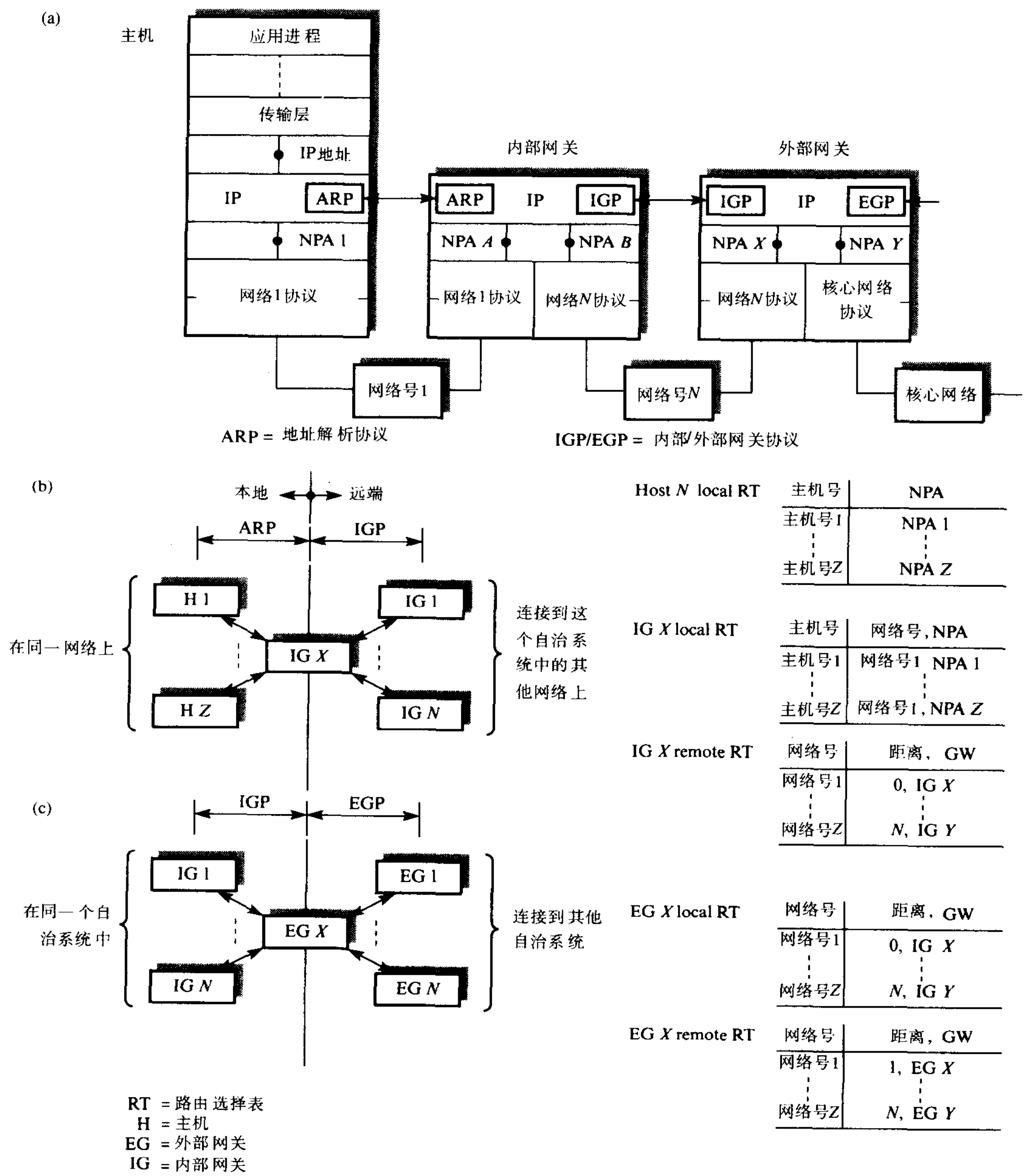


图9-13 路由选择协议

(a) 一般体系结构 (b) ARP/IGP范围及路由选择表 (c) IGP/EGP范围及路由选择协议

所需目标主机中的ARP在请求报文中发现了自己的IP地址就继续处理它。它先检查源主机号/NPA地址对是否在它的路由选择表中，如果没有，它就插入。然后它通过返回一个含有

自己NPA地址的**ARP应答报文**响应给请求主机的ARP。接到应答报文，源主机中的ARP先在它的路由选择表中插入请求的主机号/NPA地址对的记录，然后把等待的数据报地址指针以及相应NPA地址（该地址表明数据报的去处）传递给SNDAP协议。目标主机会记录主机号/NPA地址对，因为很有可能当稍后更高层协议响应数据报时目标主机会需要它。

在前面看到，一个主机的IP/NPA地址对通常保存在永久存储空间中并由计算机操作系统在启动时读取。在无盘主机中，这是不可能的，所以使用了称为**逆地址解析协议**（RARP）的相关协议。有关每个无盘主机组的服务器有一份它服务的所有主机的IP/NPA地址对的副本。当某无盘主机先进入服务状态时，它广播一个RARP请求报文（有该主机的物理硬件网络地址（就是它的NPA））给服务器。接到这类报文，服务器中的RARP会响应一个含有请求者IP地址以及服务器自己IP/NPA地址对的应答报文。实际上，关于ARP和RARP的请求和应答报文的格式是相同的，如图9-14所示。

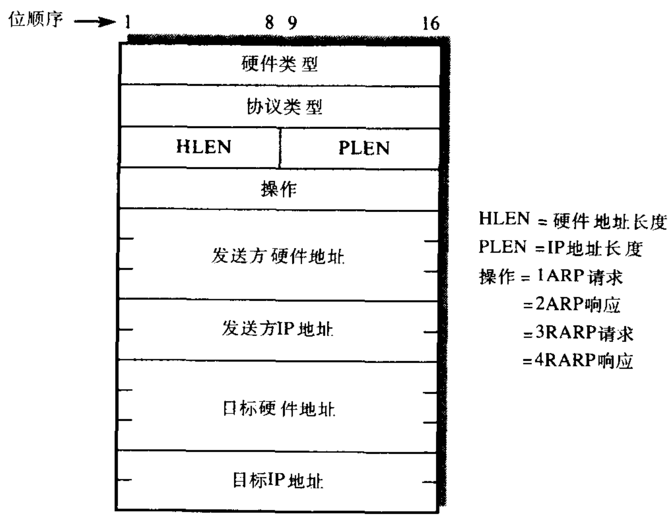


图9-14 ARP和RARP的报文格式

操作字段说明了特定报文的类型：ARP 请求/应答，RARP 请求/应答。当产生一个ARP请
509 求时，发送方把它自己的**硬件地址**（HA）和IP地址写到相应字段，并把目标IP地址写到**目标IP地址**字段。在RARP情况下，发送方简单地包含了自己的HA地址。为了确保HA地址被正确地解释，**硬件类型**字段说明了LAN的类型，例如CSMA/CD是1。**协议类型**字段说明了使用的协议类型：ARP、RARP以及随后几节规定的其他协议。

3. 内部网关协议

在前面指出，内部网关路由选择协议会随自治系统变化。使用最广泛的协议是**IP路由选择信息协议**（RIP）。它是**分布式的路由选择协议**，基于**距离向量算法**（DVA）的技术。引入的更新协议基于**链路状态**（LS）和**最短路径优先算法**（SPF）。**链路状态开放最短路径优先**（link state OSPF）协议已经采用，作为与ISO CLNP一起使用的国际标准。因为DVA专用于TCP/IP，所以在这里讨论它。在9.8.3节中讨论ISO CLNP的关系时，将讨论链路状态OSPF。

术语距离用作两个网关间的**路由选择度量**。例如，如果度量以跳数计算，那么它是两个网关间的中间网络个数。如果度量以时延计算，那么它是两个网关间的平均发送时延。依此类推。无论使用的是哪种度量，DVA使用分布式算法使得自治系统中的每个内部网关都能建

立一张表，该表含有它自身同那个系统中所有其他网络间距离。

最初，每个网关只知道它所连接的每个网络的网络号以及连接在这些网络的每个网关的IP/NPA地址对。一般这个信息由管理在网关初始化时载入，网络号放在**远端路由选择表**中，510 而IP/NPA地址对放在网关的**邻接表**中。内部网关的远端路由选择表的格式如图9-13(b)所示。如果度量以跳数计算，远端路由选择表只简单地含有它的每个本地网的网络号（距离是0）以及它自身的IP地址（作为计算距离的起点网关地址）。类似地，如果度量是以时延计算，它由网关发送报文（数据报）给连接在它自身网络的每个网关，并在接到响应前计算的时延决定。然后距离设成这些值的一半。

周期地，每个网关发送它的（远端）路由选择表的当前内容给它的每一个邻网关。基于收到的邻网关表的内容，它更新或者增加自身路由选择表的内容。接收到的邻网关表中所含距离加上接收方网关到它的紧邻网关的已知距离，接收方网关就简单地得到实际距离。因为重复这个规程，因此在每次重复后，当告知新的距离时，开始更新路由选择表。如果告知到某个网络的距离小于当前的记录就更新该记录。在若干次重复后，每个网关针对自治系统中每个网络都含有一条记录。获得这个结果所花的时间取决于系统的规模以及路由选择信息交换的频率。路由选择信息在系统中传播的时间称为**路由传播时延**。

例如考虑如图9-15(a)所示的简单网络并假定度量是跳数。建立路由选择表的方式如图9-15(b)所示。每个网关的初始内容只简单地含有它本地网络的网络号。对于这个网络，每个路由选择表的内容仅仅在两次路由选择表交换后就完成了。每个网关的最终路由选择表含有系统中每个网络的距离以及用来到达该网络的紧邻网关。这样从网关1，到网络号为6的网络的距离是2个跳数（就是说经过两个中间网络），通过网关2。在网关2，到网络号为6的网络是1个跳数的距离，通过网络号为6的网络所连接的网关3。

可以轻易地推出跳数度量能导致选择较差的路由。例如，如果每个网络的时延度量跟它的网络号一致，从网关4到网络号为6的网络会选择更快地以3个跳数经过网关1、2和3，而不是以2个跳数经过5和6。时延度量通常给出更佳的性能。使用时延的一个协议是**呼叫**（HELLO）。从它的名称可以看出，时延由周期地发送**呼叫报文**给它的每个邻网关并计算响应时间来决定。

为了确保表中的记录反映了差错发生时网络的当前拓扑结构，每条记录有个相关的计时器。如果记录在规定时间内未被证实，那么它就超时。这意味着每个网关以恒定的时间间隔发送它的完整路由选择表，一般30秒。对于一个小型网络，这不是问题。但是对于大型网络，距离向量算法的相关开销会非常高。还有，因为记录按接收顺序建立并且距离相等路由会被丢弃，所以网关到相同的目标可以经过不同的路由。结果，某些路由间的数据报可能会循环 511 而不是直接到达所要的网关。同样，只有一条路由保留在路由选择表中，因此不使用可选的路由。基于这些原因，对于大型互联网来说，OSPF渐渐成为首选的IGP。

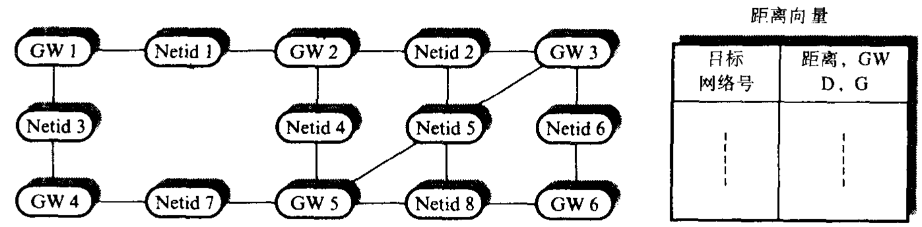
4. 外部网关协议

每个自治系统的相关管理机构指定一个或多个网关作为该系统的外部网关工作。在自治系统内，它们使用系统的IGP与其他内部网关通信。每个外部网关通过它的本地路由选择表知 512 道那个系统内的网络号和从该网关到它们的距离。路由选择表的内容以刚才描述的方式建立。

当每个外部网关刚开始启动时，给它一个它连接的自治系统的惟一标识。它还收到称为**可达表**的路由选择表内容，这使得它能通过核心网络同其他所有外部网关通信。然后EGP使每个外部网关与选定外部网关联系并与它们交换路由选择信息。这个路由选择信息由相应自

治系统的网络号列表以及从报告外部网关到它们的距离和路由组成。发送网关用这个信息为转发数据报到特定自治系统选择最佳外部网关。

(a)



(b)

初始路由选择表

GW 1		GW 2		GW 3		GW 4		GW 5		GW 6	
网络号	D,G	网络号	D,G	网络号	D,G	网络号	D,G	网络号	D,G	网络号	D,G
1	0,1	1	0,2	2	0,3	3	0,4	4	0,5	6	0,6
3	0,1	2	0,2	5	0,3	7	0,4	5	0,5	8	0,6
		4	0,2	6	0,3			7	0,5		
								8	0,5		

2 4		1 3 5		2 5 6		1 5		2 3 4 6		3 5	
1	0,1	1	0,2	1	1,2	1	1,1	1	1,2	2	1,3
2	1,2	2	0,2	2	0,3	3	0,4	2	1,2	4	1,5
3	0,1	3	1,1	4	1,2	4	1,5	3	1,4	5	1,3
4	1,2	4	0,2	5	0,3	5	1,5	4	0,5	6	0,6
7	1,4	5	1,3	6	0,3	7	0,4	5	0,5	7	1,5
		6	1,3	7	1,5	8	1,5	6	1,3	8	0,6
		7	1,5	8	1,5			7	0,5		
		8	1,5					8	0,5		

最终路由选择表

2 4				2 5 6		1 5				3 5	
1	0,1	1	0,2	1	1,2	1	1,1	1	1,2	1	2,3
2	1,2	2	0,2	2	0,3	2	2,1	2	1,2	2	1,3
3	0,1	3	1,1	3	2,2	3	0,4	3	1,4	3	2,5
4	1,2	4	0,2	4	1,2	4	1,5	4	0,5	4	1,5
5	2,2	5	1,3	5	0,3	5	1,5	5	0,5	5	1,3
6	2,2	6	1,3	6	0,3	6	2,5	6	1,3	6	0,6
7	1,4	7	1,5	7	1,5	7	0,4	7	0,5	7	1,5
8	2,2	8	1,3	8	1,5	8	1,5	8	0,5	8	0,6

(c)　GW1的邻网关表

网关	网络号，NPA
GW 2	网络号 1，NPA *X*
GW 4	网络号 3，NPA *Y*

NPA = 网络连接点地址

图9-15　距离向量算法实例

(a) 拓扑结构　(b) 假定使用跳数为度量，路由选择表的建立　(c) 网关的可达表

EGP的三个主要功能如下：

- 邻网关获取
- 邻网关可达
- 路由选择更新

每个功能通过交换请求—响应报文实现。每个功能的相关报文如表9-1所示。

表9-1 EGP报文类型及它们的含义

功 能	EGP报文	含 义
邻网关获取	获取请求	请求网关成为邻网关
	获取证实	网关同意成为邻网关
	获取拒绝	网关拒绝成为邻网关
	中止请求	请求终止同邻网关的联系
	中止证实	证实断开联系
邻网关可达	呼叫	请求邻网关证实先前建立的联系
	我听到你	证实联系
路由选择更新	轮询请求	请求网络可达更新
	路由选择更新	网络可达信息
差错响应	差错	响应任何不正确的请求报文

因为每个自治系统由不同机构管理和运行，在交换路由选择信息前连接到不同系统的两 513
个外部网关必须先同意交换这些信息。这是**邻网关获取和终止**规程的任务。当两个网关同意这类交换时，它们被认为已成为**邻网关**。当一个网关先想交换路由选择信息，它发送一个**获取请求**报文给相应的网关的EGP，然后返回一个**获取证实**报文；如果它不想接受这个请求就返回一个包含原因代码的**获取拒绝**报文。

一旦两个网关（由此自治系统）间建立邻接关系，会周期性地证实它们的联系。这通过交换特定报文（**呼叫**（hello）和**我听到你**（I-heard-you））或者通过嵌入证实信息到普通路由选择信息报文的头部来完成。

实际的路由选择信息交换由其中一个网关来执行，它发送一个**轮询请求**报文给其他网关，请求通过这个网关可达的网络（网络号）列表以及从它到它们的距离。响应是**路由选择更新**报文，它包含所请求的信息。最后，如果请求报文不正确，**差错报文**作为响应返回，并带有相应的差错原因代码。

像其他IP协议一样，所有EGP相关的报文（PDU）由IP数据报的用户数据字段携带。所有EGP报文有一样的固定报头，格式如图9-16所示。

版本字段定义了EGP的版本号。**类型**和**代码**字段共同规定了报文的类型，而**状态**字段含有报文有关的状态信息。**校验和**与IP中的一样，用作对差错报文处理的安全措施。**自治系统数**是发送网关所连接的自治系统的指定数，**序列号**用作响应与相应请求报文同步。

邻网关可达报文只含有报头，报头中有一个值为5的类型字段和一个代码字段（0表示呼叫而1表示我听到你）。

邻网关获取报文有一个值为3的类型字段，代码号定义了特定报文类型。**呼叫间隔**说明呼叫报文发送的频率，**轮询间隔**为轮询报文执行相同的功能。

轮询报文有值为2的类型字段。代码字段用来携带邻网关可达信息：代码为0表示呼叫而代码为1表示我听到你。在轮询报文和路由选择更新响应报文中的**源网络IP地址**指明了连接两个外部网关的网络。这允许核心网络由多个网络组成。

路由选择更新报文含有自治系统内每个网关可达的网络（网络号）列表，按照响应外部网关的距离顺序排列。正如指出的，它使得请求网关能在自治系统内选择最佳的外部网关用于转发数据报。注意为了保留空间，每个网络号地址以3个字节（24位）发送而最高8位主机号字段丢失。后者对于所有地址类类型都是多余的。 514

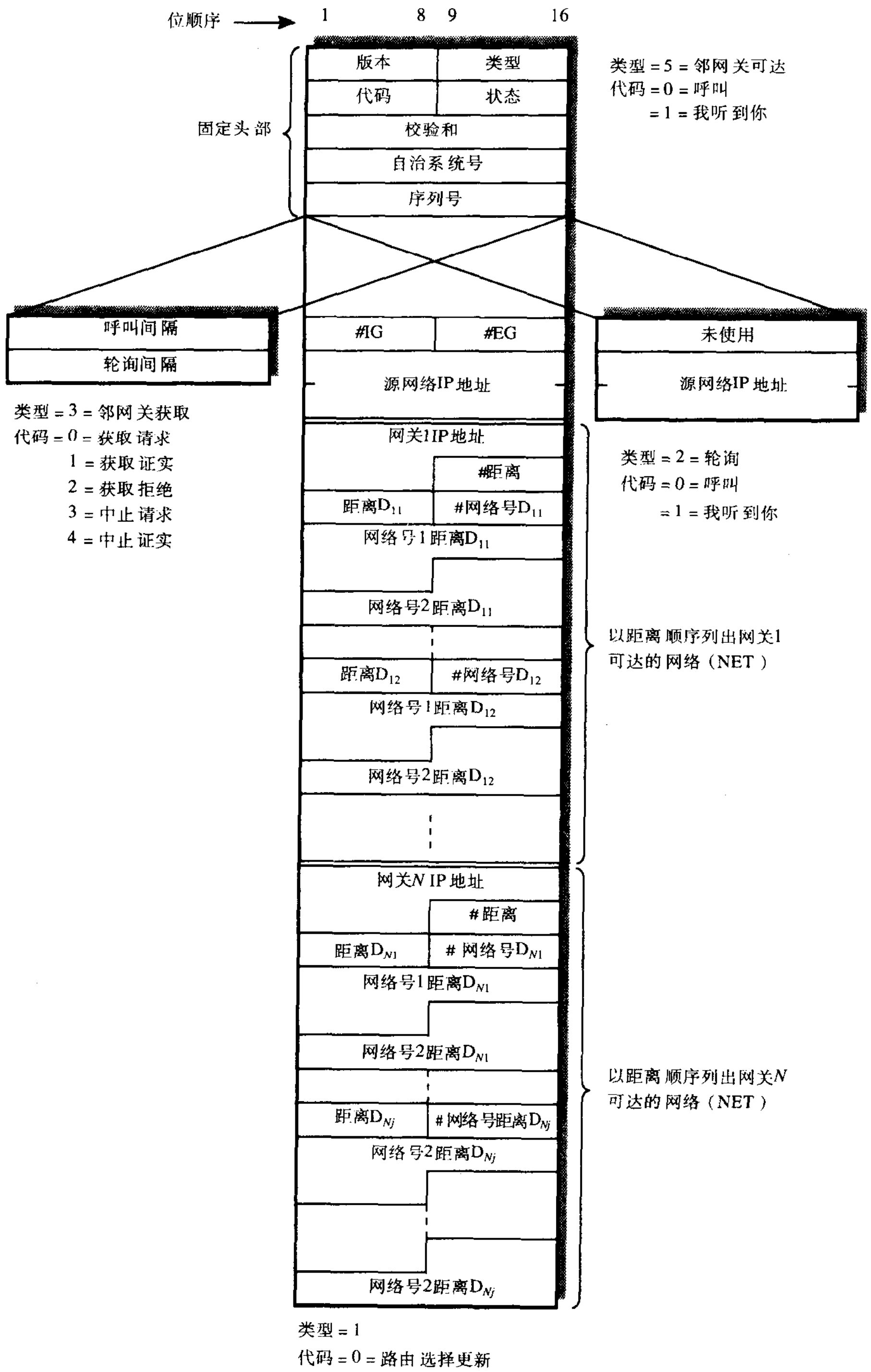


图9-16　EGP报文格式

9.5.6　因特网控制报文协议

因特网控制报文协议（ICMP）形成了所有IP实现的主要部分。它由主机和网关，尤其是网络管理用来实现许多功能。关于ICMP的主要功能如下：

- 差错报告
- 可达测试
- 拥塞控制 515
- 路由改变通知
- 性能测量
- 子网寻址

关于每个功能的报文类型如表9-2所示。每个报文以标准IP数据报传输。

表9-2 ICMP报文类型及其使用

功能	ICMP报文	使用
差错报告	目标不可达	数据报由于报文中指明的原因而被丢弃
	超时	数据报中的生存时间参数超时并由此被丢弃
	参数错	数据报头部的参数不可识别
可达测试	回显请求/应答	检查指定主机或网关是否可达
拥塞控制	源抑制	请求主机减小发送数据报的速率
路由交换	重定向	由网关用来通知连到它的其中一个网络的主机使用同一个网络上的可选网关转发数据报到指定目标
性能测量	时间戳请求/应答	确定两个主机间的传输时延
提交寻址	地址掩码请求/应答	由主机用来确定与子网相关的地址掩码

ICMP = 互联网控制报文协议

因为IP是一种最佳尝试（无确认）协议，当数据报在互联网中传输时会被丢弃。当然，传输差错是一个原因，但是数据报会由于多种原因被主机或网关丢弃。在缺少差错报告功能时，主机不知道发送数据报到指定目标的重复失败是因为传输线路不好（或网络中的其他故障）还是目标主机被关闭。各种有关**差错报告**功能的报文用作这个目的。

由于传输差错破坏数据报，它被简单地丢弃。如果数据报因为其他原因被丢弃，丢弃数据报的主机或网关中的ICMP会产生**目标不可达**差错报告报文（带有差错原因代码），并把它返回给源主机中的ICMP。差错原因如下：

- 目标网络不可达
- 目标主机不可达 516
- 目标主机上指定协议不能用
- 需要分段但数据报报头的不分段（DF）标记被设置
- 由于管理原因与目标网络通信被禁止
- 由于管理原因与目标主机通信被禁止

其他差错报告报文包括**超时**，它说明在被丢弃的数据报中的生存时间参数已经超时，以及**参数差错**，它说明被丢弃的数据报报头中的参数不可识别。

如果网络管理方收到来自用户的报告说指定目标没有响应，原因必须使用**可达测试**功能来判断。一般，网络管理方收到这类报告就发送**回显请求**报文给可能出错的主机来判断它是否在工作状态并响应请求。接到回显请求报文，目标的ICMP简单地把它变为**回显应答**报文返回给请求发送方。如果需要，会在选定网关上执行类似的测试。

如果数据报由于临时过载状况造成的无空闲存储缓冲区可用而被丢弃，返回**源抑制**报文给源主机中的ICMP。这类报文可以由主机或网关产生。它们可以请求源主机降低发送数据报

的速率。当主机接到这类报文，它以协定的量来降低发送速率。每次数据报被丢弃时会产生一个新的源抑制报文，这样源主机会逐渐降低发送速率。这些报文有助减轻互联网内的拥塞控制。拥塞会在9.7.3节中讲述ISO CLNP 时进一步讨论。

当网络连接有多个网关时，一个网关会收到来自某主机的数据报，尽管该网关从它的路由选择表判断，该主机通过连接在同一网络中的不同网关发送该数据报会更好。为了把这个情况通知源主机，该网关中的ICMP会返回一个**重定向**报文给源主机中的ICMP，该报文会指出源主机到指定目标的更优网关。然后源主机中的ICMP会在它的路由选择表中增加一条针对该目标的记录。

互联网的一个重要操作参数是数据报的平均传输时延。它是数据报在互联网中从指定源到指定目标传输所花的时间的衡量方式。为了确定这个时间，主机或网络管理方会发送**时间戳请求**报文给指定目标。每个报文包含如下三个时间相关参数（称为**时间戳**）：

- 源发送数据报的时间
- 目标接收数据报的时间
- 517 目标返回数据报的时间

接到时间戳请求报文，目标中的ICMP会简单地填入相应时间戳字段并返回该数据报给源。接到应答，源会确定到那个目标的当前往返时延并由此得到数据报传输时延。

最后，当使用子网寻址时，主机使用**地址掩码请求**和相应应答报文来确定本地子网相关的地址掩码。它需要由主机来判断，例如指定目标是否连接在同一网络上。地址掩码由子网相关的本地路由器保留。主机中的ICMP会通过发送请求报文并读取应答中的掩码来获取该地址掩码。

9.6 IPv6

组成因特网的互连网络数量的快速增长意味着如果当前的增长率继续的话目前使用的IPv4的32位地址在不久的将来会需要扩展。当我们期待的时候，**因特网工程任务组**（IETF）已开始着手IPv4协议后继版本的规范制定。它称为**下一代IP**（IPng）或者更准确的，**IP 版本6**（IPv6）。

发展新协议的第二个动机是随着网络数量的增长每个外部网关内所需的路由选择表的规模也在增长。正像在9.5节中解释的，每个外部网关必须知道用来到达因特网中任何网络的下一个网关地址。显然，由于网络数量的增长，所以每个外部网关的路由选择表的规模也在增长。打算制定的IPv6应该减小这些网关内所需的路由选择表的规模。其他动机包括对于多播的更好支持以及网络安全级别的提升。因为新协议当前正在制定中，这里的目的仅仅是给出它的主要特点的概述。

9.6.1 数据报结构

为了加速IP数据报头部的处理，IPv6的头部已经分成两部分：**基本头部**以及一个或多个可选**扩展头部**。基本头部的格式如图9-17所示，可以看到它比IPv4数据报头部包含更少的字段。这意味着许多数据报的处理会更快，因为只有那些包含扩展头部的数据报才有额外的字段。

版本号和以前的版本有一样的位置和功能。它用来指定版本号为6。这样，数据报的接收方能轻易地判断该数据报的相关版本号并且由此依据特定格式来处理其余位。这使得新协议
518 能在转变阶段与IPv4共存。

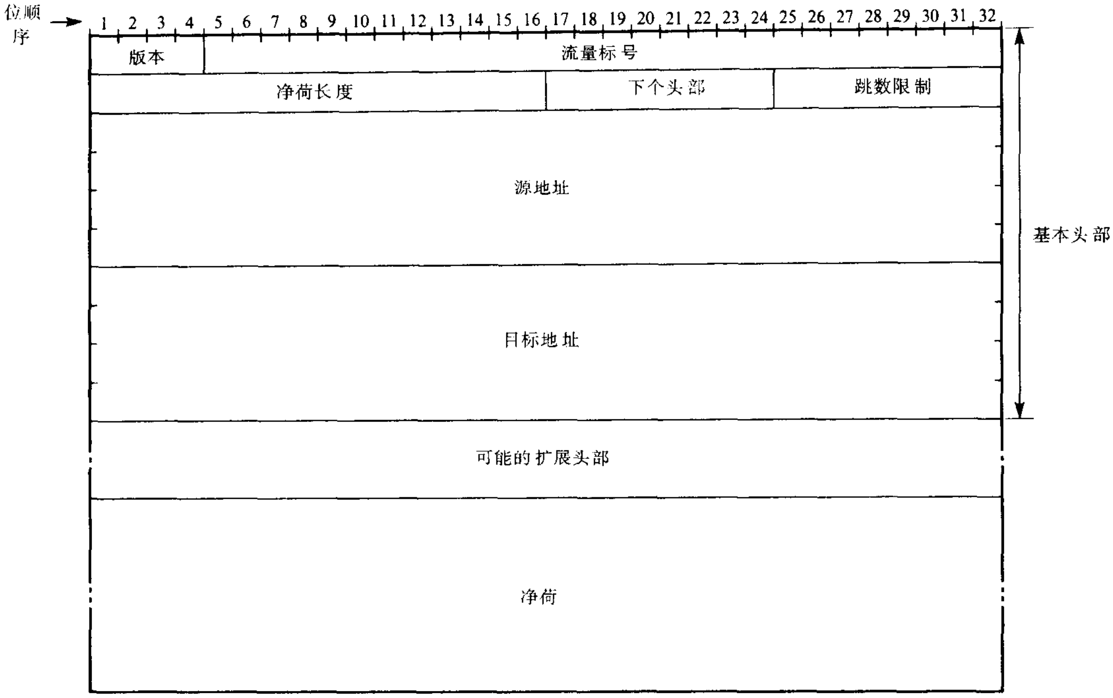


图9-17 IPv6数据报格式及头部字段

流量标号被引入，使得源能说明**净荷**的数据字段中的信息类型。例如它可以是语音采样或者视频帧，两者都应该在因特网数据报传输期间由中间网关给予比正常计算机数据更高的优先级。

净荷长度说明头部后面的净荷字段中的字节数。像以前一样默认最大长度是64K字节，但是通过把这个字段设为0并包括含有实际长度值的扩展字段可以使用更大的值。

额外（头部）信息携带在单独扩展头部中。它们紧跟目标地址字段并且当前已经规定了少量这类头部。每个通过**下个头部**字段中的不同值来确定。

跳数限制用来阻止数据报持续地陷入循环。该字段的值每访问一次网关就会减1，如果在到达所需目标前为0，那么这个数据报会被丢弃。

源地址和目标地址都是128位分级地址。它显著地增加了可用地址的数量。为了减小连到骨干核心网络的外部网关的路由选择表的规模，引入了新的高阶地址（就是说除了网络号和子网 519
号以外）。它称为**群集地址**（cluster address），用来确定网络（由此主机）所在的拓扑区域。

当前定义的扩展头部包括如下：

- 逐跳头部 用于携带必须由路由上所有网关检查的信息。
- 端对端头部 用于携带只有所需目标检查的信息。
- 路由选择头部 当使用源路由选择时会存在。它含有经过预期路由的网关地址列表。当数据报从一个网关路由到下一个网关时，基本头部中的目标地址会改变。
- 分段头部 只有源数据大于路由到目标所经过任何网络的最大报文传输单元，它才需要。这类情况中，源在发送前会把数据分段成多个数据报。这样中间网关不会涉及分段，并且分段头部中的字段只在目标重装时才使用。
- 认证头部 将在第12章中看到，认证用来确保源发送信息的完整性。由此认证头部中的

字段用来认证数据报的源。

- 加密头部 当数据在互联网中传输对安全有要求时才需要。数据先由源加密然后放在该头部的数据部分发送。

9.6.2 多播支持

除了定义新一代的IP外，一种更有效的支持多播的实验覆盖骨干网正被引入。该网络称为**多播骨干网**（m-bone）。回忆一下，多播涉及发送多播组中每个成员（主机）产生的所有数据报的副本给组中其他所有成员。使用当前的骨干网络，每个主机数据报的多个副本会导致网络中通信大量增加。将在第10章看到，当含有来自计算机数据的不同介质数据报要发送时，比如音频采样和视频帧，每个多播会话的数据报数量会显著增加。

为了减少这个负载，所有多播数据报的单个副本先被发送给称为**多播路由器**（m-router）的新型交换结点。在每个骨干网的接口都有一个这样的路由器，它们通过高带宽链路互连形
520 成多播骨干网。为了使每个多播会话使用的传输带宽最小化，当多播组建立时，在每个多播路由器和组中所有其他多播路由器间建立**路由选择树**。然后，当路由一个多播数据报时，只有当该多播路由器是特定树中的一个分支结点时才产生多个副本。

9.7 ISO 网际协议

ISO网际互连方式基于称为**ISO网际协议**（ISO-IP）或**ISO CLNP**的跨互联网、无连接的子网无关会聚协议。它定义在ISO 8473，并且有许多在9.5节中描述过的IP协议特点，它正是基于IP协议的。

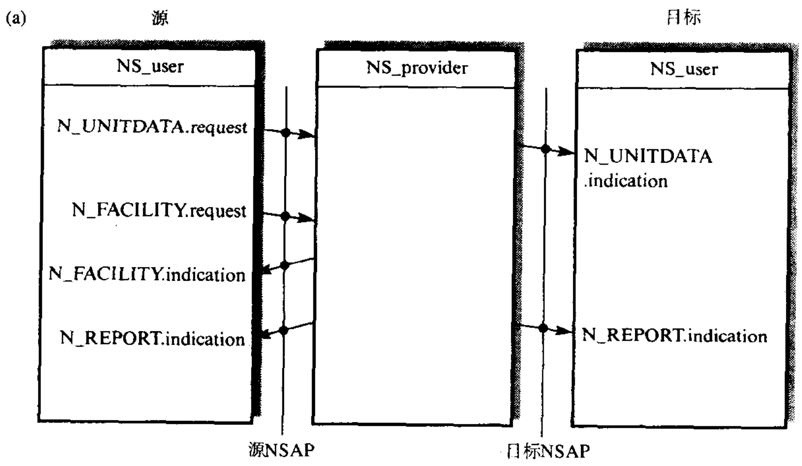
除了全网际互连协议之外，还有两个子集：**非活动网络层协议**和**非分段协议**。前者是经常用于LAN的无连接网络协议，旨在用于单一网络的相关应用。这样，源和目标ES（站）都连到同一个网络，所以不需要前面定义的协调功能。

非分段（非分段是IP术语）协议旨在用于由子网（IP术语中为网络）组成的互联网，这些子网最大包长度小于或等于单个互联网协议数据单元中传输的NS_user数据（NSDU）。该协议的相关分段功能在这种情况中不需要。

9.7.1 用户服务

这个协议的相关用户服务原语及其参数如图9-18所示。它们除了QOS和服务特征涉及整个互联网而不是单一网络外，和非活动网络层协议（见6.5.3节）一样。通常，N_FACILITY服务由NS_user用来判断当与指定目标NSAP地址通信时期望从互联网得到的QOS和服务特征。当说明每个N_UNITDATA.request原语相关QOS时，源NS_user使用这个信息。

源和目标地址是两个通信NS_user的跨互联网NSAP地址。它们的格式如图9-19所示，并且和第8章中由ISO和ITU-T规定用于公共载波WAN和其他网络的地址格式一样。回忆一下，NSAP地址是分级的，有20个字节长（40个BCD数字）。AFI指定负责分配IDI、IDI格式和
521 DSP抽象语法的机构。IDI说明了与DSP地址相关的特定网络寻址方案。DSP本身是分级的并由下面部分组成：可选地址域部分、区域部分、子网标识符部分和系统标识符（ID）部分。用于各个部分的长度（数字个数）可以随开放系统环境不同而不同。但是，一旦规定用于某个环境，所有系统（ES和IS）都有一样的NSAP格式。在每个NSAP结尾，最后的SEL字节用在NS_user接口允许标识单个ES内的多个传输实体（可达256），例如在支持多个应用层实体的ES中。

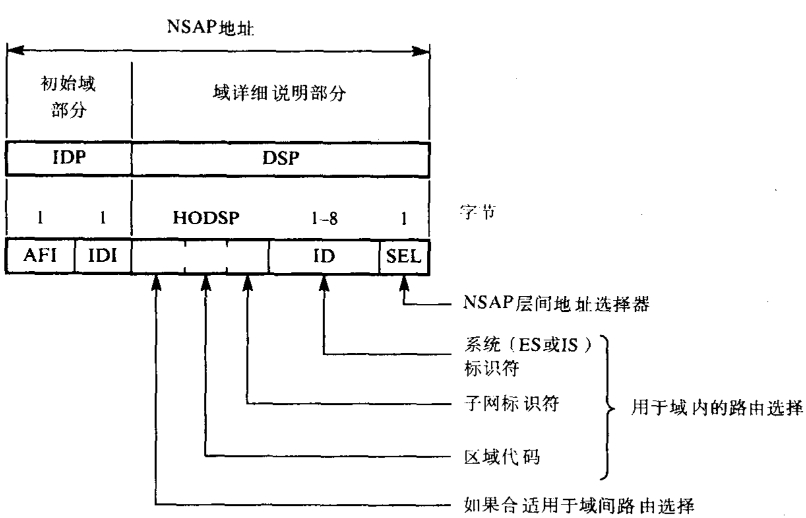


(b)

原语	参数
N_UNITDATA.request .indication	目标地址（NSAP） 源地址（NSAP） QOS/服务特征 用户数据（NSDU）
N_FACILITY.request .indication	目标地址（NSAP） QOS/服务特征
N_REPORT.indication	目标地址（NSAP） QOS/服务特征 原因代码

QOS = 服务质量

图9-18 因特网服务原语
(a) 序列 (b) 参数



HODSP = 高阶域详细说明部分

图9-19 ISO CLNP NSAP 地址结构

将在9.8.3节看到，在大型互联网中单个寻址域可以分成许多区域。一般，**高阶域详细说明部分**（HODSP）的两个低位字节用于确定域内的每个区域/子网。HODSP字段的剩余字节全设成0。另一种情况，如果有互连域的实际需要（类似于因特网内的自治系统），HODSP字段的剩余字节用于域间路由选择。在小型开放系统环境中区域可以与某个地点相关，在这种情况每
522 个区域地址会确定单个跨地点LAN。但是，在大型环境中，区域地址本身是分级的，允许一个区域含有多个地点。在两种情况中，ID字段惟一地确定了区域/子网分层结构内的NS_user。

用户数据参数是（NS_user）传输协议数据单元，它可以达到64 512字节长。因为协议只提供最佳尝试无连接服务，NS_user（传输协议实体）通常执行额外的端对端差错控制，因此由于与重新传输NSDU相关的传输时延可能较长，不太可能会使用这种较大的用户数据长度。传输时延涉及到互联网期望的QOS，可以在选择采用的用户数据字段长度时由传输实体使用。

实际上，QOS参数是共同表达与指定目标（NSAP）地址相关的互联网服务性能的一列参数。该列参数包括如下：

- 传输时延　成功地通过互联网从源NSAP传输NSDU到指定目标NSAP所花的平均时间。由本地源传输协议实体用来确定重新传输协议的超时时间间隔。
- 成本决定因素　该可选字段由一组允许NS_user影响IS为该NSDU选择路由的成本和选项组成。
- 剩余差错概率　它规定了丢失、重复和不正确传输NSDU的数量与传输NSDU的总数量
523 的百分比。这个信息影响源传输实体选择最大用户数据字段。
- 优先级　该可选参数允许NS_user指定一个NSDU相对于其他传输的NSDU的优先级。例如，它可以用来发送加速数据。
- 源路由选择　该可选参数由一组允许NS_user指定该NSDU经过互联网路由（实际上是IS的顺序列表）的子参数组成。通常，这在大型互联网中不可知，因此该参数不存在。

关于N_FACILITY服务的互联网特征列表包括如下：

- 拥塞控制　它说明了流量控制是否由互联网（NS_provider）在NS_user接口执行。使用无连接服务，本地ISO CLNP返回差错原因代码设成NS_provider拥塞的N_REPORT.indication原语作为对N_UNITDATA.request的响应。
- 序列保存概率　它是本地ISO CLNP作出的测量结果，说明序列保留传输与总传输的比率，由本地传输实体用作差错控制和流量控制。
- 最大NSDU生存时间　它指出了互联网在丢弃NSDU前传输它所花的最大时间。它允许传输协议实体量化在重新传输NSDU前必须等待直到接收到确认信息的最大时间。

9.7.2　使用的服务

正如在9.7.1节中指出的，互联网协议（SNICP）提供同样的用户服务原语集给所有NS_user，而不管ES连接的基础子网提供的服务。在一个由多种子网（网络）类型组成的互联网中，这些子网可以是面向连接的或无连接的。因为ISO CLNP是子网无关的，每个子网必须对提供的服务作出选择。在ISO CLNP中，假定所有子网提供无连接服务。因为这与它自己的NS_user接口提供的服务相同，所以使用前缀SN而不是前缀N来说明由每个组成子网提供的传输协议数据单元的服务。显然，在一些ES中无连接服务与基础子网提供的服务是相同的，而其他一些则不同。考虑到这种差别，使用SNDCP来执行必要的映射操作。图9-20给出了两个实例。

在图9-20(a)中，假定ES连接在一个X.25分组交换子网，因此实际的子网服务是面向连接的。所示的交互涉及发生在源DTE-DCE接口的交互。但是在图9-20(b)中假定ES连接在一个提供无连接服务的LAN。一般，所示的交互发生在ES的ISO CLNP与连接在同一个LAN的IS的
524 ISO CLNP之间。

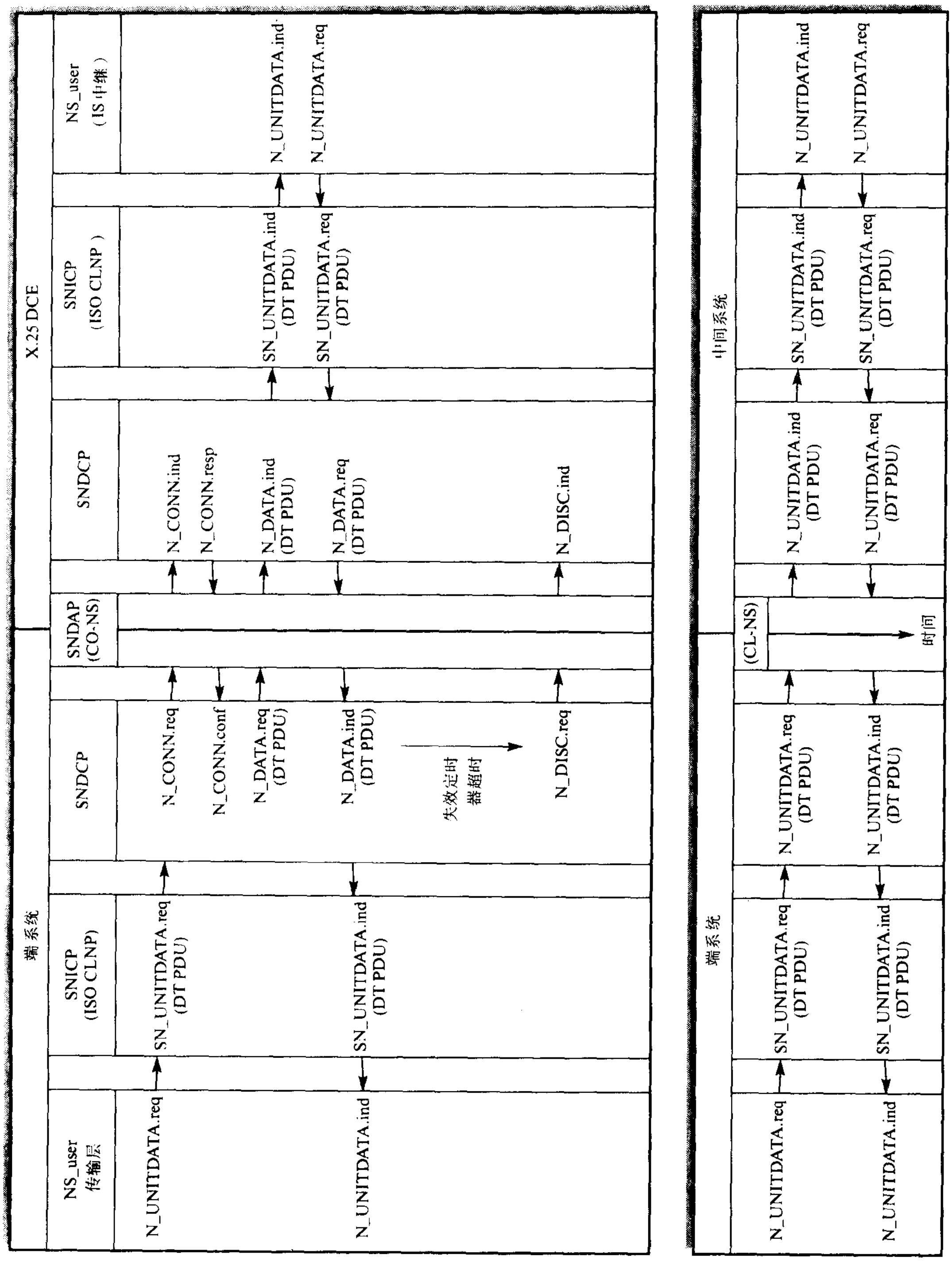


图9-20 SNDCP功能

接到N_UNITDATA.request原语，ISO CLNP使用该原语相关的参数产生**数据协议数据单元**（data PDU）。它使用把PDU作为用户数据参数的SN_UNITDATA.request原语通过SNDCP发送该PDU（到对等IP）。因为图9-20(a)中的子网是面向连接的，SNDCP必须在发送PDU前使用N_CONNECT服务建立虚拟呼叫（电路）。它等待N_CONNECT.confirm原语，然后使用把PDU作为用户数据参数的N_DATA.request原语发送PDU。随后能直接使用N_DATA服务发送PDU。

因为IP提供无连接用户服务，因此产生了用什么来触发该VC清除的问题。在没有来自IP（SNICP）的特定断开连接请求的情况下，SNDCP必须自己发起该虚拟呼叫的清除。为了做到这一点，SNDCP有个计时器（**失效计时器**），当它每次收到来自IP或者SNDAP（X.25 PLP）的虚拟呼叫的原语时就重新开始计时。如果定时器超时，SNDCP假定该呼叫的对话已结束并通过发出N_DISCONNECT.request原语进行呼叫清除。

在图9-20(b)中，因为子网是无连接的，所以SNDCP仅仅需要执行简单的一对一映射。这样它使用N_UNITDATA.request原语直接把PDU发送给把它作为用户数据参数的SNDAP。反方向上的传输以相同的方式进行。

9.7.3　协议功能

ISO CLNP包括许多独立功能，它们执行9.2节规定的各种协调操作。包括执行如下操作的功能：

- 分段和重装
- 路由选择
- 流量和拥塞控制
- 差错报告

每个数据PDU中的各个字段与上述功能相关。数据PDU的结构如图9-21所示。将在讨论各种功能时讨论各个字段的含义和用法。

1. 分段和重装

ISO CLNP的分段和重装功能基本上跟因特网IP相关的分段和重装功能相同。实际上ISO CLNP支持网内分段和互联网分段（封装），虽然后者是默认的。当一个PDU（或者一个NSDU）
526 被分段成许多更小的PDU（在IP术语中称数据报）时，更小的PDU称为**派生PDU**而前者称为**初始PDU**。还有，在ISO CLNP中生存时间字段称为**PDU生命时间**，以500毫秒为单位。

与分段和重装功能相关的数据PDU（数据报）中的字段如下：

- 段长度　PDU中（包括头部和数据）的字节数。
- 段偏移　段中数据开始处距离初始NSDU中第一个字节开始处的偏移。
- 更多段标记　如果它不是初始PDU的最后一个PDU（从该初始PDU派生而来），设成1。
- 总长度　说明初始NSDU的整个长度。

527 派生PDU通过源和目标NSAP地址以及创建时源ES中IP指派的额外的惟一**数据单元标识符**（DUI），与它们的初始NSDU相关联。它与IP数据报中的标识字段相同。**分段允许标记**用来向每个IS中的IP指明是否允许进一步的分段。它在PDU路由选择期间使用，如果分段不被允许（标记=0）必须选择能支持当前PDU长度的路由（子网）。当发生没有合适路由可用时，PDU会被丢弃并且产生一个差错报告（差错原因代码设成目标不可达）。实例9-2说明了每个字段的用途。

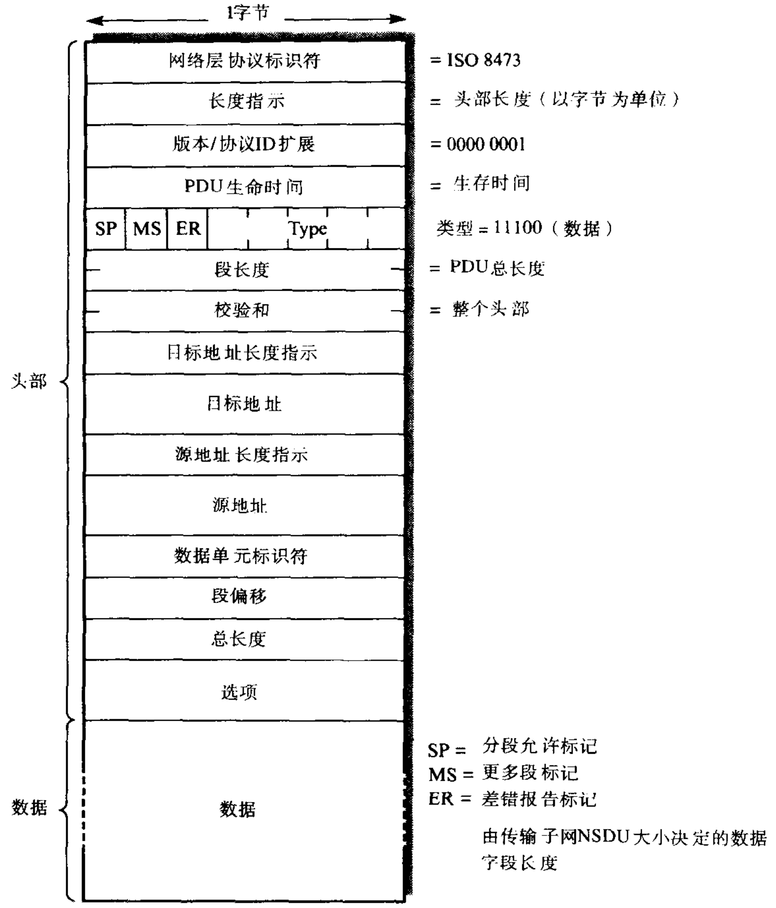


图9-21 ISO CLNP——数据PDU结构

实例9-2

假定两个NS_user通过如图9-10所示的互联网通信，而且NSDU以1024字节的长度传输。还有，三个子网的SNDAP都是无连接的，SN 1的最大用户数据长度为512字节，SN 2为128字节，SN 3为256字节。每个数据PDU的头部是固定的，等于24个字节长。

假定互联网分段，当它经过这三个子网传输时在每个PDU的头部得出如下字段的内容：

- 总长度
- 段长度
- 段偏移
- 更多段标记
- 数据单元标识符

解：

（a） SN 1

SN 1的用户数据字段 = 512个字节

头部 = 24个字节，由此实际数据 = 488个字节

所以，对于SN 1有三个初始PDU：前两个是488个数据字节而第三个是48个数据字节。

每个PDU的不同字段内容如下：

总长度	1024	1024	1024
段长度	512	512	72
段偏移	0	488	976
更多段标记	1	1	0
数据单元标识符，如，	20	20	20

（b）SN 2

SN 2的用户数据字段 = 128个字节

头部 = 24个字节，由此实际数据 = 104个字节

所以，SN 1的前两个初始PDU针对SN 2，每个需要5个派生PDU。第三个初始PDU不变地被传输。

528 每个PDU的不同字段内容如下：

总长度

1024 1024 1024 1024 1024 | 1024 1024 1024 1024 1024 | 1024

段长度

128 128 128 128 96 | 128 128 128 128 96 | 72

段偏移

0 104 208 312 416 | 488 592 696 800 904 | 976

数据单元标识符

20 20 20 20 20 | 20 20 20 20 20 | 20

（c）SN 3

SN 3的数据字段 = 256个字节，并且由此在互联网分段中，所有进来的PDU经过SN 3无变化地被传输。

在目标ES中的接收IP端：

（i）根据总长度，能重装1024字节的NSDU。

（ii）具有源和目标地址的DUI用来把所有派生PDU联系到初始NSDU。

（iii）相对于NSDU总长度的段偏移能用来以正确的序列重装所有段。

2. 路由选择

除了使用不同的术语，与ISO CLNP相关的基本路由选择操作与IP中使用的类似。为了使每个ES和IS能建立关于某子网（网络）的本地路由选择信息，使用了称为**端系统到中间系统**（ES到IS）的协议。它定义在ISO 9542中，执行与IP中使用的ARP类似的功能。IS（网关）间用于路由选择的协议称为**中间系统到中间系统**（IS到IS）协议。它定义在ISO 10589中，执行与IP中使用的IGP类似的功能。执行与IP中使用的EGP类似功能的ISO协议定义在ISO 10747中。

因为路由选择是SNICP的主要功能而且用于ISO CLNP的术语是不同的，将会详细地描述与ISO CLNP相关的两种路由协议。两种路由协议以及它们的相关路由选择信息数据库如图9-22(a)和图9-22(b)所示。

因为ES到IS和IS到IS使用的都是SNICP子层的一部分，所以使用ISO CLNP路由选择协议相同SNDCP/SNDAP子层在系统间交换它们的PDU。每个ES中的路由选择数据库（称为**ES路由选择信息库**（ES-RIB））由ES到IS协议单独维护，而每个IS中的路由选择数据库（称为**IS路由选择信息库**（IS-RIB））由ES到IS和IS到IS协议共同维护。

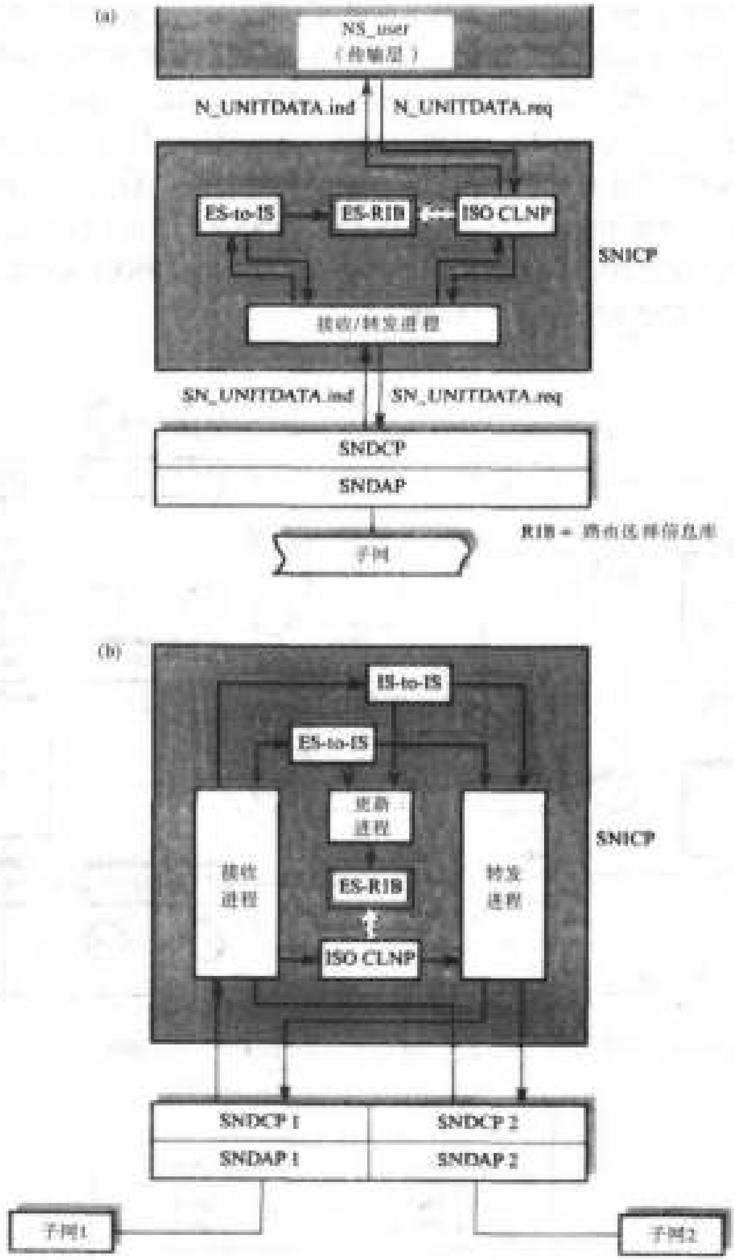


图9-22 SNICP概要结构

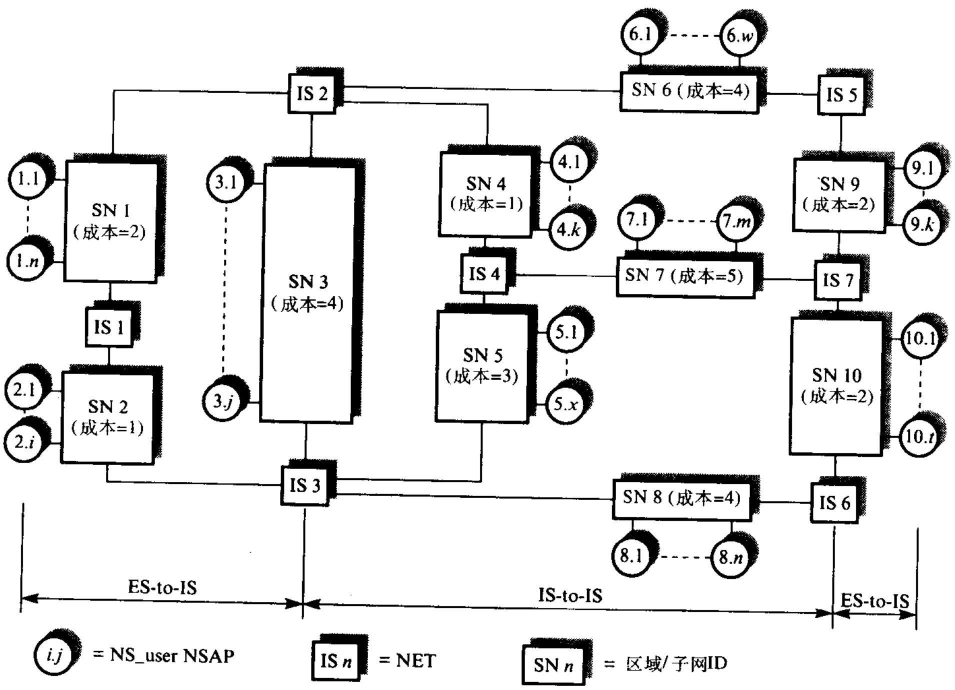
9.8节给出了两个协议的更详细描述。基本上，每个协议的相关PDU（含有路由选择信息）定期地产生和交换。由此每个路由选择数据库中的信息会不断地更新。在任何时刻，两个数 529 据库的当前内容用来路由数据PDU。

图9-23(a)给出了与实例互联网相关的每个协议的范围。针对这个互联网的两个RIB中的实例记录如图9-23(b)图9-23(c)所示。可以看到，每个RIB由许多表组成。为了标识方便，SNPA 530 地址只显示它们相关的特定子网的标识。实际上，它们可以是ISO 8802 MAC地址、ITU-T X.121地址甚至专用的指定地址。

记住，NSAP地址是分级的并由一个跨互联网的子网（加上区域）标识符和惟一标识整个互联网内NS_user/IS的系统标识符（ID）组成。还有，每个IS有多个SNPA地址，每个对应它连接的子网，但它是单一ID。后者在ISO术语中称为**网络实体标题**（NET）。

最初，连接在某个子网的ES不知道它们本地IS（就是说连接在同一子网中的IS）的SNPA
531 地址。类似地，IS不知道连接在它的子网中的ES的SNPA地址。这样，作为ES到IS协议的一部分，每个ES在它的本地子网中广播它的标识（NSAP/SNPA），每个IS也在它连接的每个子网中广播它的标识（NET/SNPA）。通常，ES只记录它的本地IS的NET/SNPA地址对而IS记录连接在它们子网的所有ES的NSAP/SNPA地址对。

(a)



(b)　ES-RIB

ES1.1

NET	SNPA
IS 1	SN 1.*i*
IS 2	SN 1.*j*

NSAP	SNPA
1.*n*	SN 1.*n*
6.*n*	SN 1.*j*

ES10.t

NET	SNPA
IS 6	SN 10.*m*
IS 7	SN 10.*n*

NSAP	SNPA
10.*s*	SN 10.*s*
8.*n*	SN 10.*n*

使用ES到IS 协议得到

图9-23　互联网及路由选择表实例

(a) ES到IS和IS到IS 的范围　(b) 选定ES的ES-RIB表实例　(c) IS 1的IS-RIB表实例

(c) IS 1的IS-RIB表

使用ES到IS 协议得到

NSAP	SNPA
1.1	SN 1.1
⋮	⋮
1.n	SN 1.n
2.1	SN 2.1
⋮	⋮
2.i	SN 2.i

由网络管理输入

线路数据库：

SN ID	Costs
SN 1	2, X, Y, Z
SN 2	1, X', Y', Z'

邻近数据库：

NET	SNPA
IS 2	SN 1.j
IS 3	SN 2.i

使用IS到IS 协议得到

链路状态数据库：

IS 1: SN 1, 2/SN 2, 1
IS 2: SN 1, 2/SN 3, 4/SN 4, 1/SN 6, 4
IS 3: SN 2, 1/SN 3, 4/SN 5, 3/SN 8, 4
IS 4: SN 4, 1/SN 5, 3/SN 7, 5
IS 5: SN 6, 4/SN 9, 2
IS 6: SN 8, 4/SN 10, 2
IS 7: SN 7, 5/SN 9, 2/SN 10, 2

转发信息数据库：

SN ID	Attached NETs
SN 1	IS 1, IS 2
SN 2	IS 1, IS 3
SN 3	IS 2, IS 3
SN 4	IS 2, IS 4
SN 5	IS 3, IS 4
SN 6	IS 2, IS 5
SN 7	IS 4, IS 7
SN 8	IS 3, IS 6
SN 9	IS 5, IS 7
SN 10	IS 6, IS 7

NET	Path	, Cost
IS 1	Local,	0
IS 2	IS 2,	2
IS 3	IS 3,	1
IS 4	IS 2,	3
IS 5	IS 2,	6
IS 6	IS 3,	5
IS 7	IS 3,	7

图9-23 （续）

由此每个ES的最少路由选择信息由一列NET/SNPA地址对组成，每对地址对应于每个连到同一子网（如ES）的IS。如图9-22(c)所示，并由ES用来转发PDU到连接在同一子网的相应IS，它会中继该PDU到所需目标。接到PDU，IS要么直接把它转发到目标ES（如果它连接到某个本地子网），要么把它转发给**邻IS**（如果PDU被路由到远端子网）。如果两个IS连接在同一子网上就称为邻IS。 532

连接到某个IS的子网列表以及它们的路径成本由网络管理来载入并保存在称为**线路数据库**的表中。另一个称为**邻近数据库**的表用来保存IS的邻IS的NET以及它们的SNPA地址。

作为ES到IS协议一部分的PDU被交换的同时，邻IS互相交换路由选择信息（作为IS到IS协议一部分）。包含在称为**链路状态PDU**的IS到IS PDU内。每个PDU含有发送该PDU的IS当前连接（由此可达的）子网（链路）的列表以及相应的路径成本值（针对每种度量就有一个值）。这样，接到链路状态PDU，IS到IS协议先在邻近数据库中开始发送IS的NET的记录增加/证实，并且在另一个称为**链路状态数据库**的表中使用**更新进程**记录PDU内所含的信息。然后转发PDU的副本给它的每个邻IS（除了发送PDU的IS）。这样互联网中的所有IS建立相同的连接矩阵

（图），其中IS作为结点，它们间存在的当前直接链路（子网）以及关于每条链路的成本。

一组新的链路状态PDU隔规定时间间隔发送。在接到每个新组后，每个IS在更新过的连接矩阵上执行称为**最短路径优先**（SPF）的算法。9.8.2节描述了SPF的实际操作。基本上，它计算指定源IS和互联网中其他所有IS（由此子网）间的最短路径。SPF的输出值被插入到合称为**转发信息库**（FIB）的两张表之一。另一张表直接从链路状态数据库得到，并且含有连接在互联网中每个子网的IS（NET）列表。

在两个规程执行完后，互联网准备在任何NS_user（ES）对间路由数据PDU。源ES中的ISO CLNP，使用从它的RIB获得的后者的SNPA地址，简单地发送每个数据PDU给某个本地IS。接收IS的ISO CLNP先从PDU中的目标NSAP地址确定子网标识符，然后与它的RIB比较来确定是否应该被转发。如果要发送给连接在它的某个子网上的ES，ISO CLNP直接把它发送给那个ES。但是如果它要发送给连接在远端子网的ES，那么IS把它发送给邻IS（邻近），该邻IS到所需目标的路径最短。例如，针对图9-23(a)中的互联网，如果目标ES连接在子网5，那么它通过IS 3或IS 4到达。从IS 1到达它们的路径成本分别是1个和3个单元，因此选择IS 3。

另外，在转发PDU之后，如果它的地址指向连接在与源ES同一子网的ES上，作为ES到IS
533 协议的一部分，IS发送另一个含有目标ES NSAP/SNPA对的PDU给源ES。它通知源ES目标ES的SNPA刚刚被请求。然后源ES能（可选地）在它的RIB中增加一条记录使得它能直接发送接下来的PDU给目标ES。

以同样的方式，如果在同一子网上存在更优的IS（就是说有更直接路径的IS），它应该用来路由PDU到前面PDU所含的目标NSAP，IS就会通知ES。这样图9-23(b)中ES-RIB所示的NSAP记录成为ES中NS_user所谓的仲裁ES。

可以从这段讨论中看到，对于由几百个子网组成并且每个子网连接了大量ES的大型互联网，每个IS中的FIB会非常大。基于这个原因大型寻址域被分成许多**区域**。还有，虽然图9-23中每个IS只显示了一个FIB，实际上可以有许多FIB，每个FIB对应一种路由选择度量。此外，将会在9.8节更详细地讨论它。

转发进程中的各子层间的交互如图9-24所示。接到NS_user请求，源ES中的IP先使用该请求的参数生成数据PDU（DT PDU）。然后查询它的RIB并使用子网请求服务发送该PDU，子网请求服务使用从RIB获得的IS的SNPA地址作为目标地址参数，而生成的PDU作为用户数据。在图9-24中，假定两个子网都是无连接的。由此，接到网络请求，每个SNDAP依次产生新的NPDU，DT PDU放在数据字段而来自RIB的IS SNPA放在目标地址字段。然后SNDAP以正常方式转发它。

IS执行类似的规程。PDU先作为用户数据通过子网的子层传递给IP。IP使用来自PDU的目标NSAP地址查询FIB来决定ES的SNPA地址（如果它连接在本地子网）或者连接在邻近子网的IS的SNPA地址（如果它要发给远端子网）。假定没有分段的必要，那么使用SNDCP服务（从FIB获得的下一个IS地址作为参数）不加改变地转发PDU到邻近子网。

已经假定，对于地址指向远端子网的PDU来说，每个ES中的RIB只含有路由上第一个（本地）IS的SNPA地址。但是，在一些情况中，RIB可以含有不止一条记录，而是可以含有路由上IS标识符（NET）的完整（或部分）列表。这个信息由源ES中的IP通过选择PDU选项字段中的**记录路由**参数获得。当该参数被设置时，每个处理该PDU的IS中的IP除了路由该PDU之外，还在选项字段中记录本IS的NET。这样当PDU经过互联网传输时就建立了它经过的路由。这个信息可以保存在目标ES的RIB中并且随后用于反方向转发PDU。反方向执行类似的规程。这样，在PDU的第一次交换后，两个ES间的路由就已知了。

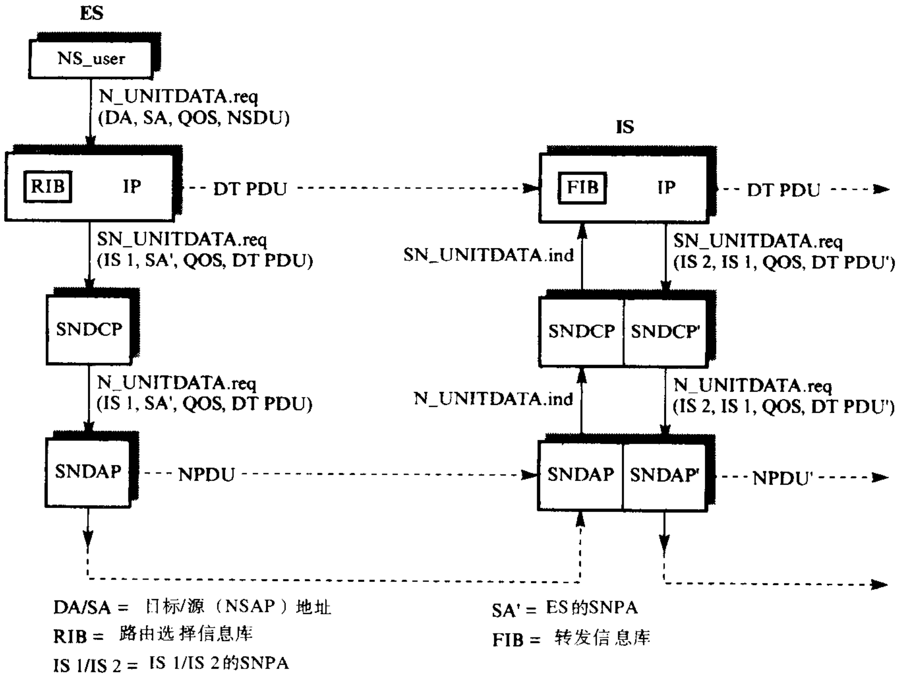


图9-24 子层交互

当使用这个特性时，ES中IP接收到的请求是要发送到路由已知的ES，它简单地把IS标识符的完整列表载入到选项字段并选择**源路由选择**参数代码。与该字段相关的是偏移量，当该PDU从一个IS转发到下一个IS时它会递增。接收到源路由选择参数被选中的PDU，IP简单地访问列表中下一个IS的标识，偏移量加1，然后转发该PDU到这个IS。 534

IS到IS协议，正如在9.8.3节中看到的，尽力在所有路由选择表中维护最新的记录使得它们能反映当前的通信分布和IS故障。结果，当IS故障发生时，所有涉及那条IS的记录路由只会部分有效。如果是这种情况，那么只有这部分列表会载入并且**部分源路由选择**参数被选中。在到达列表中最后一条记录后，该路由的剩余部分会从IS的FIB中的记录动态得到。

除了用于路由选择外，记录路由选项可以用作网络管理功能。通过分析PDU经过的路由，能获得对互联网行为的了解并且确定潜在的问题/瓶颈。实际上，它是记录路由选项作为正常动态路由选择方式的主要功能，通常获得每个PDU经过的最佳路由。

3. 流量和拥塞控制

因为ISO CLNP是无连接协议，流量控制不应用在互联网内每个呼叫基础上。它留给每个ES中的传输协议实体来控制涉及端对端基础上的每个呼叫的数据流量。将在第11章中看到，用在面向连接（可靠的）传输协议的流量控制算法基于改进滑动窗口机制，类似于第8章中描述的用在X.25分组交换协议的机制。当互联网内的拥塞开始加剧时，经过拥塞区域的呼叫相关的流量控制信息被延迟并且随后传输实体使新NSDU的流入慢下来。互联网引入了对受影响呼叫进行**隐式流量控制**的附加形式。虽然它有助减轻互联网中的拥塞，但是它不必要地停止了新呼叫的建立。在限度内，如果呼叫数量继续增大，那么所有呼叫相关的PDU流量会受到影响。 535

如9.2节中提到的，当进入互联网的NSDU（由此数据PDU）数量开始接近互联网内处理它们的可用总资源时，出现拥塞。或者，在更本地化的级别，如果进入某IS的PDU数量开始

接近它的总可用资源时，该IS就变得拥塞。因为所有子网以存储—转发方式工作，资源包括每个IS内用来处理收到PDU的处理能力，以及可获得的在特定输出链路上用于存储等待转发的PDU的存储缓冲区数量。

通常，指定IS以快于它们能达到的最大综合速率处理接收到的PDU。这样对于每个输入链路只需要单个输入缓冲区。但是，因为在不同输入链路上接收到的两个或多个PDU可能需要在处理后转发到相同的输出链路，对于每个输出链路需要独立的（输出）队列。当互联网的负载增加并且PDU的到达速率加快时，输出队列的长度会增加并且增加的时延会在每个IS中经历。它发信号告知拥塞的开始；在限度内，如果所有缓冲区被完全占用，那么就超过了可获得的资源。IS就称为过载，并且PDU经历的时延会迅速增加。拥塞对于互联网（或者IS）性能的影响如图9-25所示。

图9-25(a)说明了，虽然在理想情况中当过载发生时，保持了PDU的最大平均吞吐率，但是实际上没有拥塞控制算法它不是必然发生的。这是因为互联网内的分段和重装功能会引起称为**死锁**的现象。每个IS中的拥塞控制方案必须结合起来防止死锁。但是，因为这些方案不完美，实际的吞吐量少于理想情况的吞吐量。两者之间的差别是拥塞控制方案有效性的衡量标准。拥塞和过载对于PDU经历的平均传输时延的影响如图9-25(b)所示。它直接与平均吞吐量图表相关。

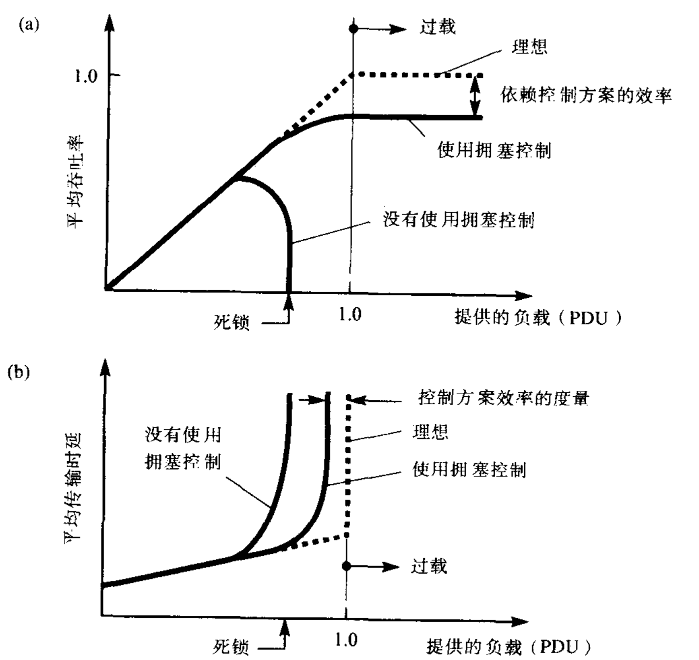


图9-25　拥塞对下列的影响

(a) 吞吐量　(b) 传输时延

加入到协议中帮助控制跨互联网基础上的拥塞的特征用于源ES中的IP监控互联网的性能，如果怀疑有拥塞就开始拒绝新的N_UNITDATA.request。这由源ES中的IP返回一个原因代码设为互联网拥塞的N_REPORT.indication给源NS_user来完成。ES中的IP通过维护互联网产生的**差错报告PDU**中的选中参数个数来监控互联网的性能。如果接收到的这些以互联网拥塞作为原因代码的PDU数量超过定义的阈值，IP开始拒绝新请求。这类似于X.25分组交换网络中

在DTE—DCE接口使用的流量控制。

协议中另一个帮助减轻（而不是控制）跨互联网基础上的拥塞的特征是在每个初始和派生PDU中包含PDU生命时间字段。正如在9.5.4节中提到的，它先由源ES中的IP设置，然后由处理该PDU的每个IP递减。如果它在互联网传输期间的任何时刻达到0，它就会被丢弃。因此，当一个或多个IS变得拥塞时，与受影响PDU相关的PDU生命时间会在处理时超时，这些PDU就会被丢弃。它意味着只要ES中传输实体使用的超时时间间隔超过两倍的PDU生命时间值（允许相关的确认信息返回），那么NSDU的副本不应被发送。这样包含PDU生命时间字段通过阻止ES发送PDU的多个副本进入互联网来提供一种**隐式拥塞控制**的形式。

虽然这些方案努力想控制跨互联网基础上的拥塞，但是不平均的通信分布意味着它们不需要避免单个IS变得过载。必须在每个IS的基本操作中加入额外的规程来使这种概率最小化。

在描述IS内可能发生的拥塞类型前，先考虑与SNICP子层相关的存储缓冲区的分布和用途。537 一个典型布局如图9-26(a)所示。在这个实例中假定可获得12个缓冲区并且使用网内分段。这样接收到的所有（派生的）PDU在处理前必须先被重装。

在任何时刻，12个缓冲区在输入缓冲区（每个子网一个）、重装缓冲区和输出队列间分配。实际上，一旦已经接收到一个PDU并存储在（存储器）缓冲区中，所有对该PDU的引用都相对于缓冲区的开始地址而言，它称为**缓冲指针**。当缓冲区（PDU）从一个地方传输到另一个地方，传输的是缓冲指针而不是缓冲区内容。

在一个输入缓冲区收到（派生）PDU（来自SNDCP），接收进程把它传输给相应的重装缓冲区，从空闲缓冲池获得一个新的缓冲区并把新缓冲区传输给输入缓冲准备接收下一个PDU。如果接收到的PDU重装成初始PDU，它由接收进程传输给ISO CLNP进行处理。

在图9-26中，假定与三个子网相关的包长度相同。这样ISO CLNP简单地改变每个派生PDU中的相应字段并把它们传递给转发进程进行转发。转发进程从每个PDU中读取目标NSAP地址，查询使用的输出链路（子网）对应的FIB，把缓冲区传输到相应输出队列的尾部。这个输出队列的SNDCP发送该PDU并由转发进程把空闲的缓冲区返回给空闲缓冲池。

通过考虑图9-26(b)所示的缓冲区分配可以推断出拥塞的最基本起因。假定三个子网输入以接近最大容量操作，并且从这些链路接收到的所有PDU在重装后需要转发到同一条输出链路。如果三条链路以相同的速率工作，链路（子网）1相关的输出队列开始增加，所以增加了使用该链路的PDU经历的传输时延。如果这种状况持续，在限度内所有缓冲区都会被占用并排队等待输出链路1。然后当没有空闲的缓冲区用来存储新的PDU时，它们被丢弃，尽管它们可能需要不同的输出链路。

注意增加缓冲区的数量不一定能解决这个问题，因为它可能简单地允许单个输出队列变得更长。所以，PDU在那些队列中等待转发经历的增加时延可能引起ES中传输协议实体使用的计时器超时。这些会开始重新发送受影响的NSDU（由此PDU），显然它反过来又增加了受影响链路上的负载。

可以通过考虑如图9-26(c)所示的缓冲区分配，观察到一种更微妙的拥塞类型。假定使用网内分段并且由此所有接收到的（派生）PDU必须在转发前被重装。如果假定可获得12个缓冲区并且每个初始PDU由3个派生PDU组成，那么6个正重装的初始PDU没有一个完成并能被转发。但是，没有任何更多的存储缓冲区，意味着没有新的PDU能被接收来完成它们。死锁 538 可以说已经发生，因为通过IS的所有PDU流量会停止。这种拥塞类型称为**重装死锁**或**重装锁住**。使用互联网分段的ES也会出现类似的状况。

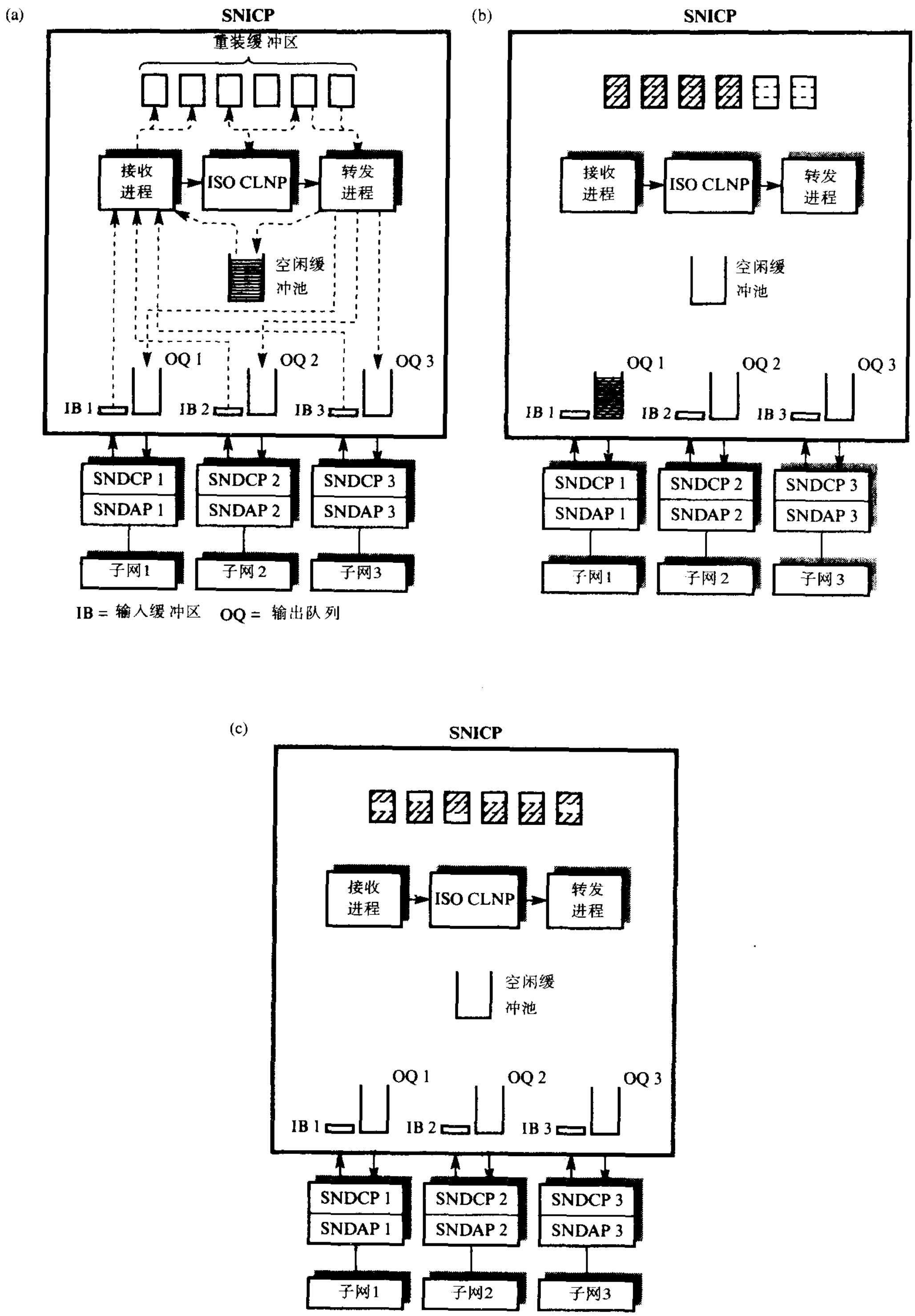


图9-26 拥塞概率

(a) 缓冲区使用 (b) 直接存储转发死锁 (c) 重装死锁

图9-26(b)所示的拥塞类型可以通过限制单条输出链路允许排队的PDU数量得到控制。允

许所有缓冲区在单条链路排队会使得使用那条链路的PDU经历无法接受的时延并且影响其他所有链路上的PDU流量，因为没有可用的空闲缓冲区来接收它们。为了防止这个情况发生，对每条输出链路上能排队的缓冲区数量设置了限制值。通常，设置了该限制值，除了其他每条输出链路上可获得的最小数量缓冲区外每条输入链路永远可获得空闲缓冲区。首先，它确保拥塞发生时所有链路上的PDU流量。其次，它对每个IS经历的时延设置了最大限制值。后者在确定每个PDU中携带的PDU生命时间值时尤其有帮助。这在由许多子网组成的大型互联网中很重要。

实际上，在单条输出链路上确定排队的缓冲区的最佳数量（作为可用缓冲区总数量的百分比）是个复杂的问题，并且取决于输出链路的数量和使用的路由选择方案。还有，最佳值不是静态的，因为它随着IS的负载而变化，这样引出了一些动态分配形式。

虽然已经研究出很多方案，但是所示的在大型互联网（ARPANET）上给出较好（不是最佳）的性能水平的方案基于一个经验公式，它定义了每条链路上的最大缓冲区数量，是链路数量和使用的空闲缓冲区数量的函数。**平方根限制值**是当前建议在ISO CLNP使用的方案。该公式如下：

$$U_{\mathrm{d}}=\frac{N_{\mathrm{b}}}{\sqrt{N_{\mathrm{c}}}}$$

这里U_{d}是一个链路输出队列的最大缓冲区限制值，N_{b}是可用的空闲缓冲区（不包括输入缓冲区）数量，而N_{c}是（现用）输出链路的数量。

另外，缓冲区最小数量专门用于使每条输出链路确保流量在一条（或多条）其他链路超载时不停止。一个实例如图9-27(a)所示。

假定使用互联网分段，这样就不需要PDU重装并且三个子网都使用相同的最大包长度。还假定有12个可用的空闲缓冲区加上用在输入链路的3个额外缓冲区。这样每个输出队列的最大排队缓冲区数是$6(12/\sqrt{3})$。因此可以获得足够的缓冲区来确保每条输入链路有1个缓冲区并且每条输出链路最少有2个缓冲区。这样如果在链路3接收到新的要转发PDU，它被丢弃并且空闲缓冲区返回给缓冲池。但是，如果在链路1或者链路2接收到新的要转发PDU，它能被 539
转发，因为它不受链路3上过载的影响。类似地，如果出现OQ2和OQ3分别含有4个和6个排队 ~540
缓冲区的情况，并且PDU可以在任何一条链路上被转发，那么它会被丢弃以确保OQ1上至少可获得2个缓冲区。

如果使用网内分段，必须控制用于重装目的的缓冲区分配以阻止重装死锁。一个方案是在任何时刻给重装处理的初始PDU数量设置限制值，并且当达到限制值时丢弃接收到的这个与新PDU相关的派生PDU。这样，因为连到IS的每个子网使用的最大PDU长度是已知的，该IS能确保它永远有足够的可用空闲缓冲区来完成重装处理。

一个实例如图9-27(b)所示。此外，假定网内分段并且每个初始PDU由三个派生PDU组成。通过在任何时刻设置重装4个初始PDU的限制值，在12个缓冲区加上用于输入缓冲区的3个额外缓冲区的情况下不会出现任何重装死锁。

注意这些拥塞控制方案不是协议的一部分，而是**缓冲区管理**的实例。这些基本方案有许多变型版本。但是目标是确保能保持PDU流量，甚至在临时过载状况发生时。

总之，丢弃PDU不是理想的解决方案，因为它不可避免地导致该互联网的传输时延的增加。但是，如果打算控制拥塞和死锁，它是必须的。大型互联网的拥塞控制主题以及路由选 541
择的类似主题仍然是研究的活跃领域。

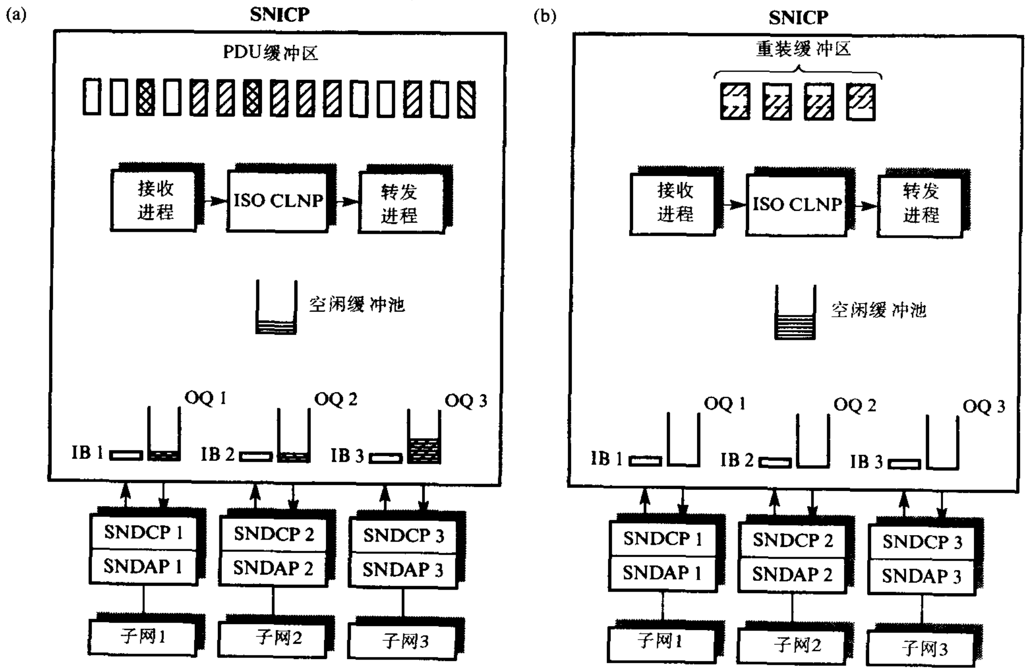


图9-27 拥塞控制

(a) 平方根限制（互联网分段） (b) 重装死锁控制（网内分段）

4. 差错报告

如图9-21所示，每个数据PDU在头部含有一个差错报告（ER）标记。该标记使得源ES中的IP能请求另一个IP（IS中或者目标ES中的）在该PDU经过互联网传输遭到丢弃时能通知它。那么，如果该标记被设置并且PDU被某个IS丢弃，该IS中的IP会产生一个携带有说明该PDU为何被丢弃的原因代码、该PDU的标识和互联网中检测到差错处的**差错报告PDU**。然后该IS中的IP把目标（NSAP）地址换成从数据PDU得到的源地址，并以正常方式返回该PDU给源IP。

每个ES中的IP能通过维护特定监控期间的各种原因代码个数来增进对互联网性能的了解。标记的使用意味着这种PDU只在ES想更新对互联网性能的了解时才产生。

PDU可能因为如下原因被丢弃：

- IS 拥塞
- PDU 生命时间超时
- 目标地址不可达

通过维护收到的原因代码为IS拥塞或PDU生命时间超时的差错报告PDU的个数，源ES中的IP能检测到拥塞的发生，并且（如果需要）开始拒绝新的NS_user服务请求。类似地，PDU生命时间超时和目标地址不可达的原因代码个数能帮助确定使用中的路由选择协议的有效性。

最后，每个数据（和差错报告）PDU头部包括校验和字段，使处理有出错字段PDU的概率最小化。校验和通常由软件产生并由两个字节组成。使用的算法与用于传输协议数据单元的相同。将会在第11章描述这些。每次处理PDU中字段和处理发生前验证校验和。如果PDU中的字段在处理中被改变，那么计算该PDU的新校验和。如果在设置了ER标记的PDU中检测到校验和差错，那么产生一个原因代码设为校验和差错的差错报告PDU。

9.8 ISO路由选择协议

SNICP子层的概要结构，说明了在每个ES和IS中的两个ISO路由选择协议以及它们与ISO CLNP之间的关系，如图9-22所示。每个ES中ES-RIB和每个IS中IS-RIB含有的路由选择信息 542 通过路由选择信息交换（作为ES到IS和IS到IS协议的一部分）生成。两个协议的PDU定期地生成和交换。因此，每个数据库中的信息持续地被更新，并且反映了当前互联网的现用拓扑。在任何情况下，两个数据库的当前内容用来路由数据PDU到目标NSAP地址所在路由上的下一个IS。正如在9.7.3节中讨论路由选择规程一样，这里只关注如何使用两个路由选择协议建立这个信息。

9.8.1 ES到IS协议

回忆一下，ISO CLNP相关的PDU不能通过检查它含有的目标NSAP地址直接被路由。通常，所有路由选择信息存储在IS中。为了发送PDU，源ES必须先把它发送给连接在同一子网上的IS。为了做到这一点，每个子网上的所有ES必须知道它们的本地IS的SNPA地址。还有，对于能路由从其他子网接收到的PDU的IS，它必须知道连接在它的本地子网的所有ES的NSAP/SNPA地址对。这是ES到IS协议的基本功能。

每个ES和IS中的ES到IS协议在它的本地子网上广播它的标识。它主要用于诸如LAN的广播子网，因为这种情况下系统向其他所有系统广播的开销很低。为了在通用（网状）拓扑中执行相同的功能，子网涉及高（被禁止的）开销，尤其是在大量ES的情况下。在这种子网中，等价的路由选择信息必须通过其他方式（诸如网络管理）载入每个系统（ES和IS）中。

有三种关于ES到IS协议的PDU形式，它们是：

- 端系统呼叫（ESH）　它由ES用来通知ES中的NS_user NSAP所属子网上所有IS。
- 中间系统呼叫（ISH）　它由IS用来通知该IS的NET所属子网上所有ES。
- 重定向（RD）　它由IS用来通知（在它自身子网上的SNPA地址的）ES，子网属于刚请求的目标NSAP或者路由PDU到那个NSAP的另一个（更好的）IS的NET。

每个PDU由通用头部部分、地址部分和选项部分组成。每个PDU类型的一般结构如图9-28所示。头部部分中的所有字段与那些用在ISO CLNP数据（和差错）PDU中的字段有相同的含义，除了**持有时间**字段替代了段长度。因为这些PDU没有数据，所以不需要段长度。持有时间说明了接收协议实体应该保留PDU内含有地址（路由选择）信息的最大时间。为了确保每个ES 543 和IS相关的地址信息保持最新，新的信息由每个系统隔规定时间间隔交换。这样持有时间基本上会使当前地址信息超时，如果某个ES停止工作，那么它的路由选择记录会自动地从每个IS中的RIB除去，直到它又开始工作并发出新的网络请求。

每个PDU的地址部分含有NSAP/NET和/或SNPA地址，取决于特定的PDU类型。最后，选项部分的内容与用在ISO CLNP中的类似。

两个广播（组）地址与用于广播PDU的协议相关：All ES和All IS。ES为了通知本地IS它（和它的SNPA地址）存在于子网上，它产生一个含有当前该ES中存在的NSAP地址的ESH PDU，并发送一个SN_UNITDATA.request原语（PDU作为用户数据而All IS作为目标地址）给它的SNDCP。

用在特定子网到达所有IS的广播地址会随不同的子网变化，由SNDCP执行解释。当本地SNDAP广播PDU，它在特定子网NPDU内包含ES的SNPA作为它的源地址。接到NPDU，IS的

544 对等SNDAP向上传递作为参数（和ESH PDU）的SNPA给ES到IS协议。后者在它保留的RIB中增加一条记录（NSAP/SNPA）直到规定的持有时间超时。

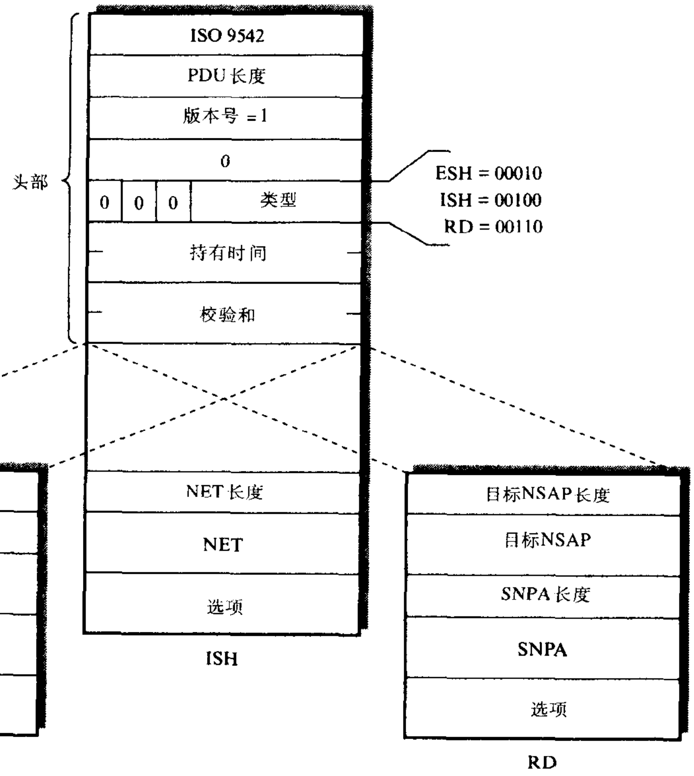


图9-28 ES到IS PDU结构

类似地，为了连接在子网上的ES知道它们本地IS的SNPA地址，每个IS隔规定时间间隔广播ISH PDU（地址字段是它的NET）。此外相应的SNPA地址作为随PDU的参数被所有现用ES接收。然后后者的ES到IS协议在它们的RIB中增加一条记录（NET/SNPA）。

最后，IS使用RD PDU通知连接在它其中一个本地子网上的ES，子网属于另一个本地NS_user NSAP的SNPA地址，或者是到达早先数据PDU中指定目标NSAP的更好IS的SNPA地址。回忆一下这个信息（NSAP/SNPA）可以（可选地）保留在它的RIB中以加速路由选择处理。

9.8.2 路由选择算法

在讨论IS到IS协议的操作以前，先描述一下IS用来通过互联网路由PDU的称为**最短通路优先算法**（SPF）的路由选择算法。有许多这类算法，但ISO使用Dijkstra **算法**。它已经应用在许多大型网络中，包括ARPANET。

因为所有的路由选择由IS执行，所以为了路由选择目的互联网可以简单地被看作通过（逻辑）链路互连的IS的分布式群体。实际上，链路可以是通过LAN或WAN的通路或者直接的点到点链路。每条链路有许多相关的路由选择度量，每个度量有个与之相关的成本值。它们是：

• 容量 线路吞吐量是以每秒位计的一种衡量标准，较高的值表示较低的容量。它是实际

使用的默认度量。

- 时延 它涉及每条链路（子网）的平均传输时延并且包括网桥和交换机中的队列时延。仍然，较高的值表示较长的传输时延。
- 花费 使用链路的货币成本的一种衡量标准，较高的值表示较大的货币成本。显然，如果能使用专用子网，那么可以说使用通过交换公共载波子网的链路更合适。
- 差错 关于线路的平均剩余差错概率的一种衡量标准，较高的值表示未检测到差错的较大概率。

回忆一下上述所有度量都与QOS参数相联系，所以当选择路由选择度量时使用QOS参数来衡量。

术语“通路成本”用来表示任何一对ES间通过互联网的特定通路（路由）使用链路的合计总成本。与路由相关的通路成本可以随度量的变化而变化。所以术语“最短通路成本”指的是使用单一路由选择度量的通路（路由）。 545

可以通过考虑特定互联网（例如如图9-23所示的用来描述ISO CLNP路由选择规程的互联网）来最好地描述最短通路优先算法。简化的表示如图9-29(a)所示。

每个IS间的链路表示为点对点线路，每一条有个相关的成本值，它可能随不同的度量而变化。对于每种度量，它的任务就是找到一条通路（路由），该通路通过互联网从信号源IS到互联网中其他每个IS有最小合计成本。算法中的各个规程如图9-29中(b)到(h)所示。假定IS 1是源IS。

与其他每个IS相关的是从该IS通过指定IS返回到源IS的合计距离的度量（显示在圆括号中）。这样记录（2，IS 4）表示通过IS 4返回到IS 1的通路成本是2个单元。最初，不是直接连到源IS的所有IS的通路成本是不可知的，因此以无限通路成本符号（就是说最大可能成本）标记。还有，直到已知一个成本值是最小成本，它是**暂定的**并且IS框没有黑色阴影。当某个IS到源IS的最短通路成本已知时，它是**永久的**并且IS框有黑色阴影。

最初，因为IS 1是源IS所以它以黑色阴影显示。两个直接相连的（邻）IS（IS 2，IS 3）的通路成本如图所示等于它们各自的链路通路成本。这样IS 2有个记录（2，IS 1），表示直接返回到IS 1的成本是2个单元，而IS 3有个记录（1，IS 1）。其他所有成本仍然是最大值。从那些仍然是暂定的IS中选出有最小成本值的IS。显然，最小成本值是IS 3为1的成本值。现在它被标记为永久IS，并重新计算一组新的通过IS 3的合计通路成本值。它们如图9-29(c)所示。

例如，IS 4的记录（4，IS 3）表示通过IS 3返回到源IS 的合计通路成本是4个单元（IS 4回到IS 3的3个单元加上IS 3回到IS 1的1个单元）。IS 2的情况下，因为通过IS 3的通路成本会是5个单元（4加1），现有的更小的暂定值保持不变。又从剩余的所有暂定IS中选出有最小成本值的IS，它是成本值为2的IS 2。它被标记成永久IS并且又重新计算通过该IS的新通路成本，如图9-29(d)所示。

可以看到，IS 4通过IS 2的新通路成本只有3个单元，这样现有的记录替换为（3，IS 2）。一旦其他所有距离都计算好了，再次选择有最小通路成本的IS——IS4。然后重复这个规程。这样剩余的步骤如图9-29(e)到(h)所示。

最后从IS 1到所有其他IS的最小路径成本被确定，就是说现在它们都是永久的，从IS 1到其他每个IS的最短（最小）通路成本路由能被确定。如图9-29(i)所示。比如从IS1到IS 5的路由是IS 1 →IS 2 →IS 5。

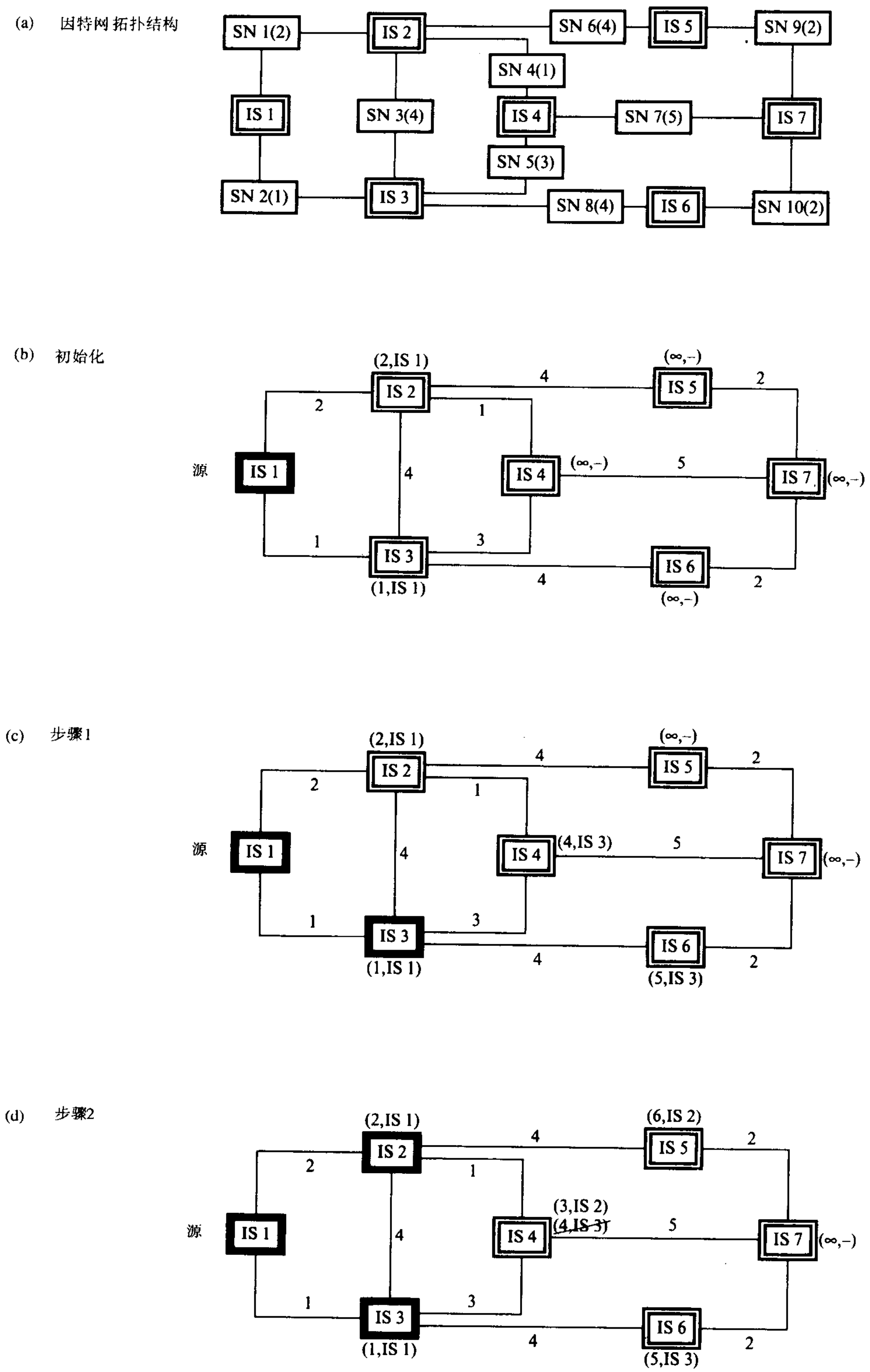
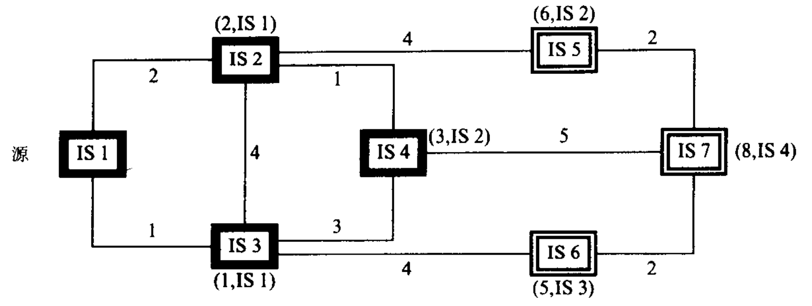
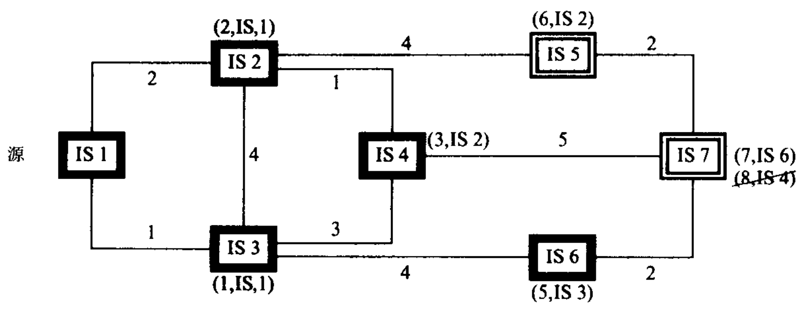


图9-29 最短通路成本路由计算

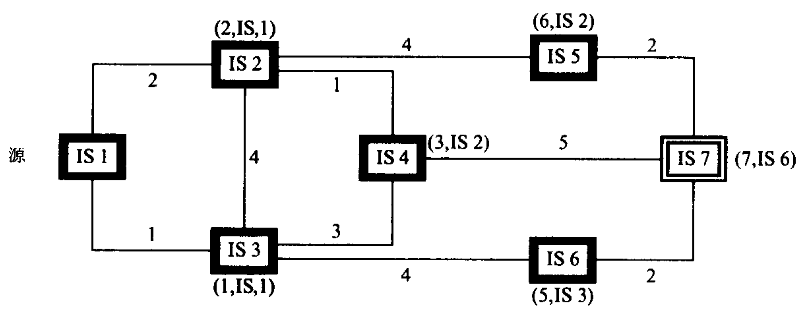
(e) 步骤3



(f) 步骤4



(g) 步骤5



(h) 距离IS 1的最终通路成本

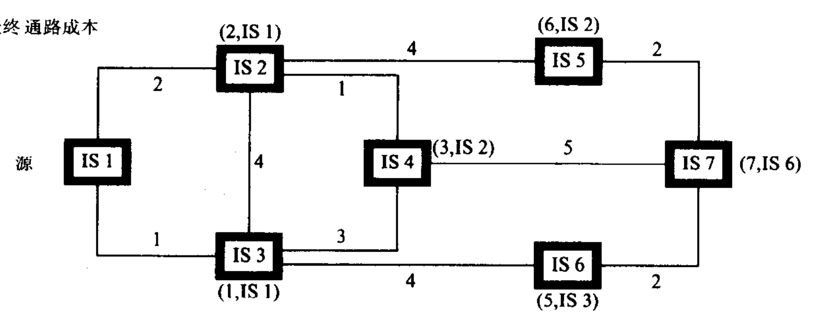
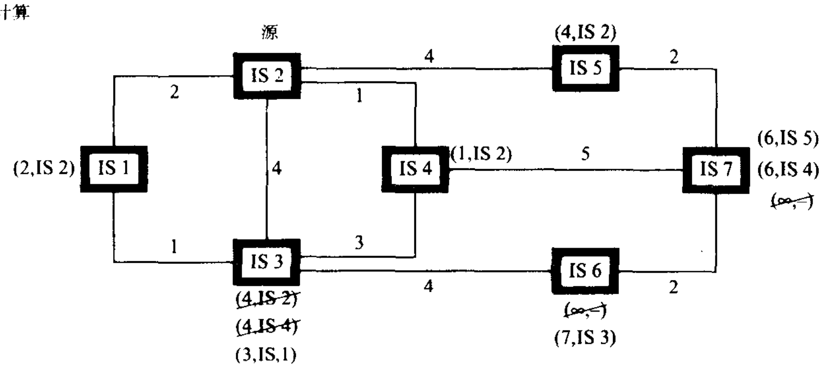


图9-29 （续）

(i)　距离IS 1的最短通路成本路由

IS 1 → 源（成本＝0）
IS 1 → IS 2（成本＝2）
IS 1 → IS 3（成本＝1）
IS 1 → IS 2 → IS 4（成本＝3）
IS 1 → IS 2 → IS 5（成本＝6）
IS 1 → IS 3 → IS 6（成本＝5）
IS 1 → IS 3 → IS 6 → IS 7（成本＝7）

(j)　距离IS 2 的通路成本计算



(k)　距离IS 2的最短通路成本路由

IS 2 → IS 1（成本＝2）
IS 2 → 源（成本＝0）
IS 2 → IS 1 → IS 3（成本＝3）
IS 2 → IS 4（成本＝1）
IS 2 → IS 5（成本＝4）
IS 2 → IS 1 → IS 3 → IS 6（成本＝7）
IS 2 → IS 4 → IS 7（成本＝6）
　　 → IS 5 → IS 7（成本＝6）

(l)　距离IS 3的通路成本计算

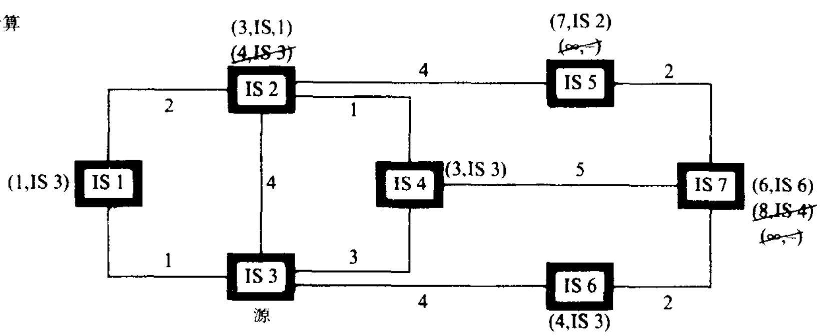


图9-29　（续）

(m) 距离IS 3的最短通路成本路由

IS 3 → IS 1（成本 =1）
IS 3 → IS 1 → IS 2（成本 =3）
IS 3 → 源（成本 =0）
IS 3 → IS 4（成本 =3）
IS 3 → IS 1 → IS 2 → IS 5（成本 =7）
IS 3 → IS 6（成本 =4）
IS 3 → IS 6 → IS 7（成本 =6）

(n) 路由选择表实例

IS 1:

目标	通路	成本
IS 1	–	–
IS 2	IS 2	2
IS 3	IS 3	1
IS 4	IS 2	3
IS 5	IS 2	6
IS 6	IS 3	5
IS 7	IS 3	7

IS 2:

目标	通路	成本
IS 1	IS 1	2
IS 2	–	–
IS 3	IS 1	3
IS 4	IS 4	1
IS 5	IS 5	4
IS 6	IS 1	7
IS 7	IS 4/IS 5	6

IS 3:

目标	通路	成本
IS 1	IS 1	1
IS 2	IS 1	3
IS 3	–	–
IS 4	IS 4	3
IS 5	IS 1	7
IS 6	IS 6	4
IS 7	IS 6	6

图9-29 （续）

现在考虑一下相同的规程，这次先把IS 2作为源IS然后是IS 3。关于每个IS的最终通路成本值以及它们相应的最短通路成本路由如图9-29(j)到(m)所示。可以得出一些关于算法的观察结果： 546~549

- 如果两个IS有相同的计算通路成本，那么必须作出关于确定固定IS的仲裁选择。
- 如果通过不同路由（IS）到暂定IS的新的计算通路成本与已存在的相同，两个都被保留，因为共享载入变得可能。
- 如果某个IS在从IS 1到另一个目标IS的最短通路成本路由上，那么它也是该IS到相同目标IS的最短通路。比如，从IS 1到IS 7的最短路由是IS 1 → IS 3 → IS 6 → IS 7，那么从IS 3到IS 7的路由是IS 3 → IS 6 → IS 7。
- 计算的最短通路成本路由是可逆的，比如IS 2 → IS 1 → IS 3和IS 3 → IS 1 → IS 2。

它的综合效果是如果每个IS计算从它到其他所有IS的自身最短通路集，那么每个IS计算的通路会一致。它意味着IS只保留允许它路由PDU到路由上第一个（邻）IS的路由选择信息，就是说，路由选择能在**逐跳**基础上进行。这样IS 1、IS 2和IS 3的路由选择表如图9-29(n)所示。

9.8.3 IS到IS协议

IS到IS协议的目的是确定每个IS中RIB的内容，就是说，当路由ISO CLNP数据PDU时使用邻IS（邻近）。但是，在路由寻找算法开始前，每个IS必须先建立它的每个邻IS的NET/SNPA对。 550

在广播子网中，每个IS使用All IS目标地址在它的每个子网上广播IS-to-IS（Ⅱ）PDU。该PDU中含有源IS的NET，并且每个接收SNDAP使用中间SNDCP向上传递给IS到IS协议一个作为参数的相应SNPA地址（在这个子网上）。在无广播子网中，IS到IS PDU又必须由网络管理来载入。在任何一种情况中，这个信息被保存在称为**邻近数据库**的独立数据库中。一旦它被执行，IS到IS协议能参与到最短通路成本生成规程中。

回忆一下，连接在IS上的子网标识以及它们的成本度量由网路管理插入一张表（称为线路数据库）。所以，隔规定时间间隔每个IS（实际上是IS到IS协议）会给它的每个邻IS发送一

个含有该IS的NET、该IS连接的子网标识符列表以及关于每个子网的成本值（每种度量一个值）的**链路状态**（LS）PDU。这样，IS从它的每个邻IS接收到LS PDU，该PDU通知它连接在其上的子网（链路）以及它们的成本值。这个信息通过SNICP中的更新进程被保存在**链路状态数据库**中。

当每个IS中的更新进程已完成这步时，它发送LS PDU 的副本给它的每个邻IS（除了发送该PDUj49的IS）。结果，每个IS接收到更多的LS PDU集，它们来自邻IS的邻IS。继续这个规程。所以，经过一段时间每个IS收到完整的LS PDU集，含有连接在互联网所有其他IS上的子网标识以及它们的通路成本值（每种度量一个）。这种路由选择规程称为**扩散**。

无论何时一个新的LS PDU集输入链路状态数据库，运行**决策进程**。它的作用是首先在链路状态数据库执行SPF算法，其次从各个数据库的所有记录中确定应该使用哪个邻IS到达其他每个IS（基于它们相应的通路成本值）。每种路由选择度量依次完成。针对每种度量的独立转发信息库产生。

为了说明这个规程，考虑如图9-29所示的子网实例的应用。在图9-30(a)重复它，为了说明清楚，再次使用单一路由选择度量。初始近邻和线路数据库如图9-30(b)所示，并且每个IS接收到的第一个LS PDU集显示在图9-30(c)中。例如，IS 1收到2个LS PDU，一个来自IS 2而另一个来自IS 3。类似地，IS 2收到4个LS PDU，分别来自IS 1、IS 3、IS 4和IS 5。

正如刚才描述的，接到每一个这类PDU，每个IS产生另一组LS PDU并把它们传递给每个邻IS。然后重复这个规程。IS 1收到的第一组、第二组和第三组LS PDU如图9-30(d)所示。

最后，图9-30(e)说明了IS 1的决策进程输出。它的RIB通过对链路状态数据库的内容执行
551 SPF算法确定到每个目标IS的最短（最小）通路成本来计算。可以看到，RIB的内容与图9-29(n)所示的相同。也可以轻易地推断IS如何执行重定向功能。例如，如果连接在SN 2上的ES发送地址指向SN 5上的ES的PDU给IS 1，那么IS 1能轻易地推断最短的路由通过IS 3，实际上它是SN 2上的其中一个邻IS并且由此用于自身的首选。

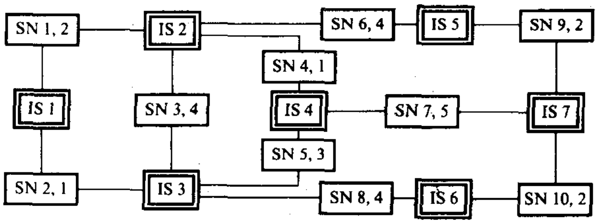
在讨论中，已经假定LS PDU的传输是可靠的并且发生传输差错的结果是没有LS PDU丢失。为了考虑到发生差错的概率，每个LS PDU含有一个序列号。当它被放在链路状态和转发数据库时随PDU保留。隔规定时间间隔，IS会交换称为**序列号PDU**的其他PDU，它含有发送IS的RIB的LS PDU的序列号。使用这个信息来确保所有IS中的RIB都是一致的。

在非常大型的网络中，每个数据库中的记录数会变得令人无法接受得多。为了克服这个问题，协议允许**分级路由选择**。使用这个技术，总路由选择（寻址）域被分成许多区域，每个含有许多子网和第一级（1级）IS。如果所要的目标ES连接在这个区域的子网上（由DSP地址的高位部分中的区域地址决定），它使用已描述过的规程被直接路由。

另一种情况，如果所要的目标在不同的区域，PDU先被发往第二级（2级）IS。该IS使用其他2级IS路由该PDU，直到它到达连接在所要目标区域的（2级）IS。从那儿使用1级IS在区域内路由。在这种情况中，每个数据库会更小并且由此路由选择开销会显著降低。图9-31说明了这种方法的概要。

最后，注意IS到IS协议是复杂的，尤其在包括分级路由选择时。本章只给出该协议的概述。如果需要更详细的描述，请直接参考ISO 10589文档。

(a)



(b)

IS 1

从 IS 2(1) – IS 2 : SN 1, 2/SN 3, 4/SN 4, 1/SN 6, 4
IS 3(1) – IS 3 : SN 2, 1/SN 3, 4/SN 5, 3/SN 8, 4

IS 2

从 IS 1(1) – IS 1 : SN 1, 2/SN 2, 1
IS 3(1) – IS 3 : SN 2, 1/SN 3, 4/SN 5, 3/SN 8, 4
IS 4(1) – IS 4 : SN 4, 1/SN 5, 3/SN 7, 5
IS 5(1) – IS 5 : SN 6, 4/SN 9, 2

⋮

IS 7

从 IS 4(1) – IS 4 : SN 4, 1/SN 5, 3/SN 7, 5
IS 5(1) – IS 5 : SN 6, 4/SN 9, 2
IS 6(1) – IS 6 : SN 8, 4/SN 10, 2

(c)

IS 1 的邻数据库:

NET	SNPA
IS 2	SN 1.*j*
IS 3	SN 2.*i*

IS 1的线路数据库:

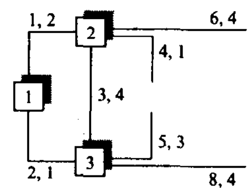
SN ID	成本
SN 1	2, X, Y, Z
SN 2	1, X', Y', Z'

(d)

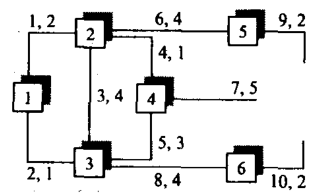
线路数据库: IS 1 : SN 1, 2/SN 2, 1

IS 1

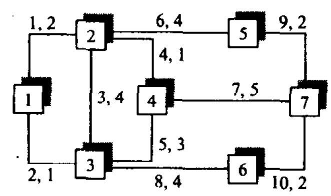
Set (1) { IS 2(1) – IS 2 : SN 1, 2/SN 3, 4/SN 4, 1/SN 6, 4
IS 3(1) – IS 3 : SN 2, 1/SN 3, 4/SN 5, 3/SN 8, 4 } ⇒



Set (2) { IS 2(2) – IS 3 : SN 2, 1/SN 3, 4/SN 5, 3/SN 8, 4
IS 4 : SN 4, 1/SN 5, 3/SN 7, 5
IS 5 : SN 6, 4/SN 9, 2
IS 3(2) – IS 2 : SN 1, 2/SN 3, 4/SN 4, 1/SN 6, 4
IS 4 : SN 4, 1/SN 5, 3/SN 7, 5
IS 6 : SN 8, 4/SN 10, 2 } ⇒



Set (3) { IS 2(3) – IS 3 : SN 2, 1/SN 3, 4/SN 5, 3/SN 8, 4
IS 4 : SN 4, 1/SN 5, 3/SN 7, 5
IS 5 : SN 6, 4/SN 9, 2
IS 6 : SN 8, 4/SN 10, 2
IS 7 : SN 7, 5/SN 9, 2/SN 10, 2
IS 3(3) – IS 2 : SN 1, 2/SN 3, 4/SN 4, 1/SN 6, 4
IS 4 : SN 4, 1/SN 5, 3/SN 7, 5
IS 5 : SN 6, 4/SN 9, 2
IS 6 : SN 8, 4/SN 10, 2
IS 7 : SN 7, 5/SN 9, 2/SN 10, 2 } ⇒



(e)

SN ID	连接的IS
SN 1	IS 1/IS 2
SN 2	IS 1/IS 3
SN 3	IS 2/IS 3
SN 4	IS 2/IS 4
SN 5	IS 3/IS 4
SN 6	IS 2/IS 5
SN 7	IS 4/IS 7
SN 8	IS 3/IS 6
SN 9	IS 5/IS 7
SN 10	IS 6/IS 7

目标IS（NET）	通路	成本
IS 1	Local	0
IS 2	IS 2	2
IS 3	IS 3	1
IS 4	IS 2	3
IS 5	IS 2	6
IS 6	IS 3	5
IS 7	IS 3	7

图9-30 路由选择信息库生成实例

(a) 互联网拓扑结构 (b) 初始化邻IS数据库和线路数据库

(c) 在第一个LS PDU集后的链路状态数据库实例 (d) IS 1的链路状态数据库发展 (e) IS 1 的最终RIB

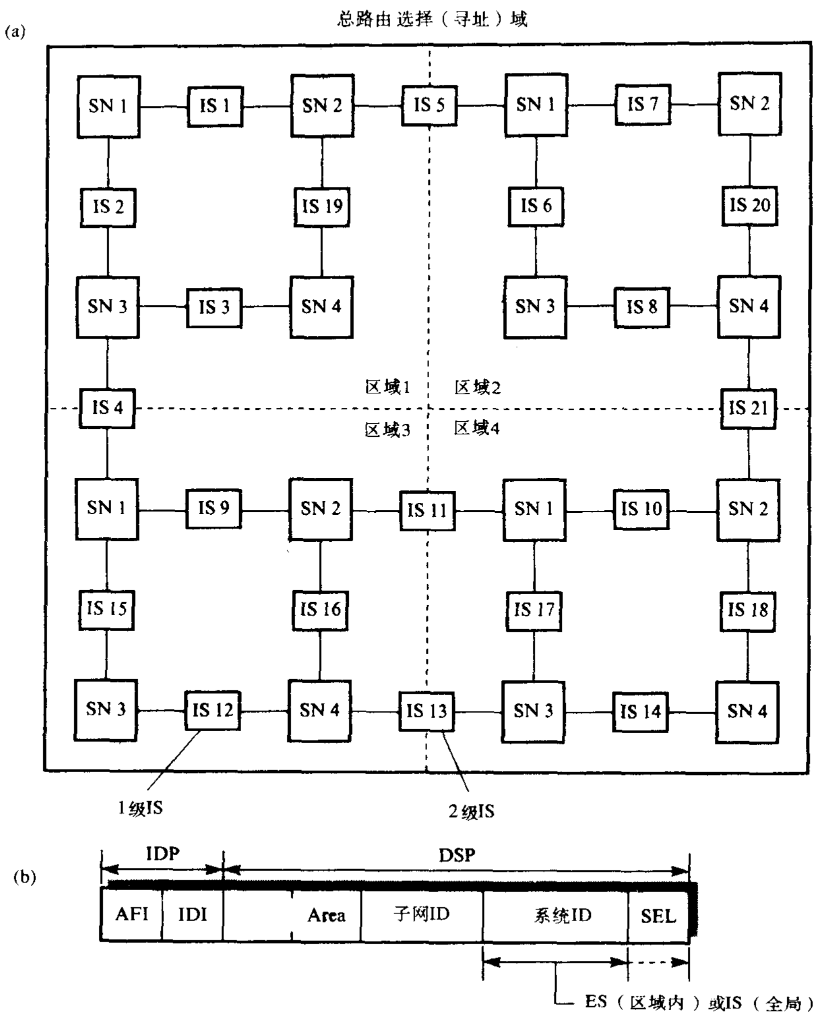


图9-31　分级结构路由选择

(a) 拓扑结构实例　(b) 地址结构

习题

9.1　解释网际互连的下列术语：

（a）因特网

（b）子网

（c）中间系统/网关/路由器

（d）协议转换器

9.2　借助图，描述如下的结构：

（a）IP 地址

（b）IP数据报

清楚地说明每个字段并解释它在IP协议关系中的功能。

9.3　一个有2000个字节的NSDU使用IP协议在网络中传输，它支持的最大NS_user数据长度为

512个字节。假定最小IP头部长度，求每个数据报头部中的如下字段：

（a）标识

（b）总长度

（c）段偏移

（d）更多段标记

9.4 （a）画一张互联网的草图说明子网路由器、内部网关和外部网关的作用。（b）根据（a）中画出的互联网，说明下列路由选择协议的范围并写出每个协议路由选择表中的记录实例：

（i）ARP

（ii）IGP

（iii）EGP

9.5 借助互联网实例，解释距离向量算法的操作。并用互联网实例说明每个网关中的路由选择表是如何得出的，假定路由选择度量是跳数。

9.6 列出互联网控制报文协议（ICMP）的相关报文类型，并由此解释该协议的各种功能。

9.7 有关用户请求的数据字段长度为1200个字节。假定该数据要发送给连接在同一网络上的主机并且该网络的NS_user数据长度为512个字节，如果每个数据报的头部为20个字节，求发送该数据所需的IP数据报数量。 552～554

并求出每个数据报头部下列字段的内容：

（a）标识

（b）总长度

（c）段偏移

（d）更多段标记

9.8 列出IPv6协议规范并讨论其制定的原因。并指出新协议改进IPv4中可识别限制的特性。

9.9 解释ISO CLNP下列术语的含义：

（a）NS_user

（b）NS_provider

（c）NSAP

（d）CONS和CLNS

9.10 借助图，描述ISO互联网NSAP地址的结构以及NSAP地址和SNPA地址间的关系。

9.11 列出组成网络层的三个子层的名称，并描述它们在ES和IS中的功能。

9.12 假定一个基于ISO CLNP的互联网，分别给出下列情况中在两个NS_user间双向传输NSDU，三个网络子层间交换的服务原语的时序图：

（a）面向连接的子网

（b）无连接子网

9.13 画出ISO CLNP数据PDU的结构图并解释每个字段的含义。

9.14 区分ISO CLNP的网内分段和互联网分段。

假定如图9-10所示的互联网，画图说明每个ES和IS中的分段和重装操作，如果NSDU = 512个字节，三个子网都是无连接的并且分别以最大用户数据长度为128个字节、256个字节和512个字节操作，使用网内分段。

9.15 考虑习题9.14，如果现在使用互联网分段，求每个PDU（当它经过三个子网传输时）头部中的下列字段内容：

(a) 总长度

(b) 段长度

(c) 段偏移

(d) 更多段标记

9.16 假定ISO网际互连，画图说明SNICP子层在ES和IS中的概要结构。在图中包括ISO CLNP和两个路由选择协议ES到IS和IS到IS。

写出每个系统操作以及协议间交互的概要描述。

9.17 假定互联网结构和表内容如图9-23所示，描述来自NSAP 1.1到下列目标的ISO CLNP数据PDU的路由选择：

(a) NSAP 1.10

(b) NSAP 3.5

(c) NSAP 10.3

9.18 借助图，解释互联网中IS的术语“过载”、“拥塞控制”和“拥塞控制效率”的含义。

9.19 描述在ISO CLNP中提供的避免互联网拥塞的两个功能。

假定如图9-26(a)所示的存储缓冲区结构，在平方根限制、15个可用空闲缓冲区以及互联网分段条件下，求最坏情况缓冲区分配实例。

9.20 借助图9-26(c)，假设使用网内分段，解释术语“重装死锁”的含义。

9.21 假定基于不同的路由选择度量，图9-29所示的10个子网的通路成本值分别是1、2、3、2、

555 2、3、4、1、2和2个单元。使用Dijkstra算法求从IS 1到其他6个IS的最短通路成本路由。

9.22 描述ISO CLNP数据PDU的路由选择，使用图9-30(e)所示的转发信息库表，在NSAP 1.1以及如下NSAP之间：

(a) 1.5

(b) 3.4

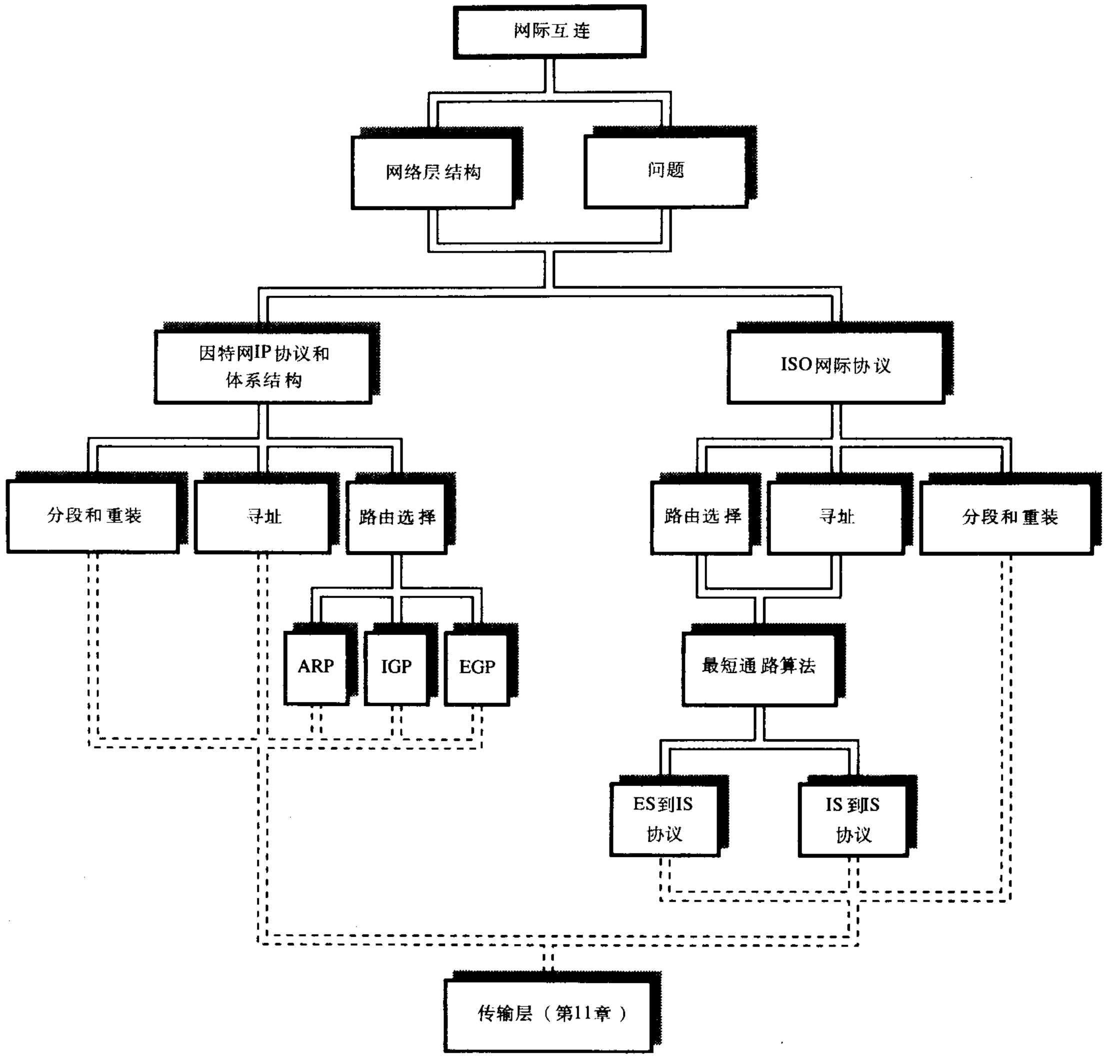
(c) 9.3

9.23 假定连接在SN 1上的ES发送地址指向SN 2上NSAP的PDU给IS 2，描述IS 2路由PDU的步骤以及随后的重定向操作。清楚地说明IS 2如何确定可能的重定向。

9.24 假定如图9-30(a)所示的互联网以及习题9.21中的通路成本值，求每个IS接收到的第一组

556 链路状态PDU。

本章概要


网际互连
网络层结构
问题
因特网IP协议和体系结构
ISO网际协议
分段和重装
寻址
路由选择
路由选择
寻址
分段和重装
ARP
IGP
EGP
最短通路算法
ES到IS协议
IS到IS协议
传输层（第11章）

第10章　宽带多业务网络

本章目的

读完本章，应该能够：

- 解释宽带多业务网络的连网需求；
- 描述FDDI-II网络的操作；
- 理解基于信元的异步传输模式（ATM）网络的操作原理；
- 描述ATM LAN的体系结构和网络组件；
- 理解一些ATM交换机体系结构的操作原理；
- 解释ATM网络的接口协议的功能；
- 理解城域网（MAN）的作用；
- 描述三种MAN类型DQDB、ATMR和CRMA-II的操作以及相关的协议体系结构。

引言

前几章描述的所有网络主要用于支持纯数据业务。一般，这些业务涉及由特定字符集（ASCII，EBCDIC等）的字符串或者二进制字节串（比如编译程序输出）组成的数据帧的交换。最近，工作站及用来互连它们的网络已经不仅能支持涉及数据的业务，还能支持涉及其他信息类型的业务。例如，工作站结合视频摄像头、麦克风和扬声器使通信双方能保持电话
558 通话（**电话**）或者在它们的显示器上打开窗口显示另一方的交谈图像（**可视电话**）。此外，通信双方能讨论报告或文档的内容，比如显示在另一个窗口。

另外，这些工作站一般能支持文档的显示，它们通常由多种媒体类型（就是说**多媒体文档**）组成。一般，它们由多个窗口组成，这些窗口除了含有文本信息之外还含有高分辨率图像或者可能带声音的移动图像（视频）。现在有许多教学和其他信息包，含有这类能被用户交互访问的文档组。这些包能保存在本地的工作站内（比如CD-ROM）或者连到网络的服务器上。应用实例包括**计算机辅助学习**（CAL）和各种**计算机支持协同工作**（CSCW）**活动**。总之，由于这些工作站提供的业务范围，它们称其为**多业务工作站**，而且把互连它们的网络称为**宽带多业务网络**，术语“宽带”说明它们使用非常高比特率的传输电路。图10-1(a)给出了
559 一种这类网络应用的示意图，并且可视电话事务实例涉及图10-1(b)所示的两个多业务工作站。

10.1　网络需求

多业务网络必须比前面几章描述的数据网络要灵活得多。在数据网络中，典型的事务可能涉及10 KB的数据传输。在10Mbps传输速率下它需要8毫秒而在100Mbps下为0.8毫秒。显然，若干个这种事务能轻易地分时共享相同的传输介质并且仍然能获得满意的业务。

相比之下，涉及声频和视频信息传输的事务可能需要整个传输期间**恒定比特率**的传输容量（由此保留的带宽）。例如，如果多媒体文档包括一个5秒带声音视频的窗口，那么它在整个5秒时间间隔持续需要传统LAN的全部10Mbps的绝大部分。显然，对于诸如涉及视频摄像

头的可视电话和**视频会议**应用来说，对网络的需求更加高，因为在整个会话/呼叫期间需要类似的带宽。

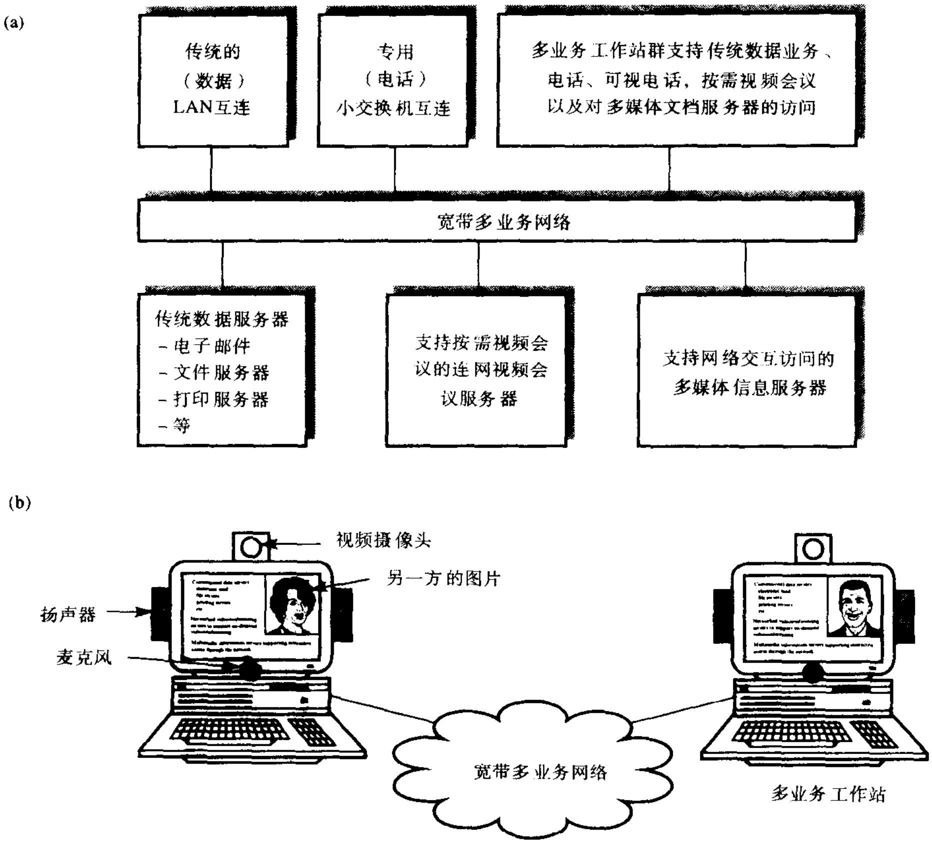


图10-1 宽带多业务网络

(a) 选定的应用实例 (b) 涉及两个多业务工作站的可视电话事务实例

为了减少支持这些额外媒介类型所需的网络带宽，通常在源信息传输前对它进行**压缩**。在接收到的信息输出前对其应用相关的解压缩算法。正如在3.5节中看到的，用于数据（文本）压缩的算法必须是**无损的**，就是说在压缩和随后的解压缩过程中没有信息丢失。但是对于语音、图像和视频的相关压缩算法，主要目的是在只保留源信息**完整性**的同时使需要传输的信息量最小化。通常，用于这些媒介类型的压缩算法在解压缩后不是获得源信息的副本而是一个可接受质量的版本。所以称为**有损压缩算法**。经过最近几年，压缩领域已取得重大进步，实际上该领域的研究仍在继续。为了帮助量化这类网络的通信需求，表10-1给出了各种媒介类型相关的无压缩和典型压缩带宽。显然，涉及多种媒介类型的事务成比例地需要更多的带宽。

在2.5.2节得出有关传统（窄带）电话的无压缩带宽。回忆一下，数字化过程使用每秒8000个样本的采样率并且每个样本量化为8位值来获得无压缩的64kbps比特率。一般来讲，诸如涉及多个用户的视频会议的应用需要比传统电话更宽的带宽（由于更高的采样率）来允许会议中的个体能更轻易地被确定。

表10-1 不同媒体类型所需带宽

媒体	事务类型	格式	采样尺度（像素，线，帧/秒）	未压缩比特率	经过压缩的最大比特率
语音和音乐	电话		8 ksps × 8 位/样本	64 kbps	8-32 kbps
	电话会议		16 ksps × 8位/样本	128 kbps	48-64 kbps
	CD—音频		44.1 ksps × 16位/样本	705.6 kbps	128 kbps
图像	一般分辨率图像	SVGA	640像素 × 480线 × 8位/像素	2.458 Mbits	24k-245kbits
		JPEG	720像素 × 576线 × 16位/像素	6.636 Mbits	104 k-830 kbits
	很高分辨率图像		1280像素 × 1024线 × 24位/像素	31.46 Mbits	300 k-3 Mbits
商务视频	视频电话	QCIF (H.261)	176像素 × 144线 × 12位 × 30帧/s*	9.115 Mbps	p × 64 kbps (p = 1，2)
		MPEG-4 (H.320)	176像素 × 144线 × 12位 × 10帧/s	3.04 Mbps	64 kbps
	视频会议	CIF (H.261)	352像素 × 288线 × 12位 × 30帧/s*	36.45 Mbps	m × 384 kbps (m = 1，2，…，5)
		MPEG-1 (PAL)	352像素 × 288线 × 12位 × 25帧/s	30.4 Mbps	1.15 M-3 Mbps
		MPEG-1 (NTSC)	352像素 × 240线 × 12位 × 30帧/s	30.4 Mbps	1.15 M-3 Mbps
娱乐视频	VCR	CIF (MPEG-2)	352像素 × 240线 × 12位 × 30帧/s	30.4 Mbps	4 Mbps
	广播电视	MPEG-2 (PAL)	720像素 × 576线 × 12位 × 25帧/s	124.4 Mbps	15 Mbps
		MPEG-2 (NTSC)	720像素 × 480线 × 12位 × 30帧/s	124.3 Mbps	15 Mbps
	高质量电视	HDTV	1920像素 × 1080线 × 16位 × 30帧/s	994.3 Mbps	135 Mbps
		MPEG-3	1920像素 × 1080线 × 12位 × 30帧/s	745.8 Mbps	20 M-40 Mbps

帧速率可以是30、15、10、7.5帧/秒。
CIF = 通用中间格式
MPEG = 运动图像专家组
QCIF = 1/4CIF格式
JPEG = 联合图像专家组

560~561 有关高分辨率图像无压缩形式的信息量由每个**像素**的图像空间分辨率以及色彩范围（位数）确定。表中两个条目，第一个涉及的空间分辨率为480线，每线640个像素，每个像素8位用于表示颜色，而第二个为1024线，每线1280像素，每个像素24位。可以看到，尽管使用压缩，带宽还是很高，并且如果这类图像在可接受的时间间隔内传送，需要较高带宽的链路。

视频应用的需求不仅取决于每帧（图像）的空间分辨率还取决于帧刷新率。为了防止抖动，宽带视频中至少使用每秒25帧的刷新率。对于可视电话来说，通常从一帧到下一帧有非常小的移动，移动只局限于整个图形的一小部分。所以使用较低的刷新率和较高压缩级别。

可视电话和视频会议的无压缩和压缩带宽远远小于广播电视所需的带宽。

从表10-1可以推出，虽然压缩减少了所需的带宽量，但它仍然是很大的，尤其对于视频来说。而且，由于许多这种信息都需要恒定比特率，前面几章描述的网络类型不足以支持涉及多个媒体类型的应用。所以，开发了一类新的网络，它除了支持传统数据通信外还支持许多媒介类型的通信。本章将讨论这些网络的操作。

10.2 FDDI-II

在第7章中讨论FDDI的操作时看到，虽然它有个可选特性能支持时延敏感（同步）数据的传输，但是使用令牌访问控制方式获得的时延变化使得这种业务不适合恒定比特率数据的传输。回忆一下，后者称为同步数据，因为它在规定时间间隔生成并且必须以相同的恒定速率传输。

为了支持同步数据，已开发出一种称为FDDI-II的FDDI的变种。在它的**基本模式**中，FDDI-II以和FDDI相同的方式运行，就是说，所有传输由控制令牌控制，然后可用环带宽（容量）使用计时令牌循环协议（见第7章）在连接站间分时共享。但是，在FDDI-II，假定环中所有站已经升级到对此支持的话（就是说，它们有可选的芯片集），那么它们可以从基本模式转变为**混合模式**。在这个模式中，带宽使用**时分多路复用**分成许多更小的单元，称为**信道**，而不是把全部100Mbps环带宽整个用于传统数据（通常称为异步数据）的传输。由此产生的每个信道可以用于异步数据或者同步数据。 562

由称为**周期主站**的选定站执行多个信道的建立，它还控制信道的利用。使用站管理子系统响应从各站接收到的同步服务请求报文。周期主站通知所有站每个新的同步信道的分配，并且只把剩余带宽用于异步数据传输。接下来详细地讨论这一点。

10.2.1 周期结构

周期主站每隔125μs（回忆一下2.5.2节用作数字化模拟语音信号的采样时间间隔）产生一个重复周期。在100Mbps下，周期长度为12 500位。为了环中所有站与每个周期的开始同步，每个周期的头部有20位的时钟同步模式，后跟8位惟一的周期开始帧模式。连续的周期串以正常方式从站到站沿着环循环。因为环等待时间永远大于125μs，所以在任何情况下环会含有若干个周期，实际环等待时间由周期主站控制，它是周期时间的整数倍。一般方案以图方式显示在图10-2(a)中，图10-2(b)说明了每个周期如何再被分成许多信道。

正如指出的，在每个周期的开始处是20位的时钟同步模式，它称为**前同步码**，也称为**帧间隙**。使用与FDDI中相同的线路码符号，并且前同步码由5个IDLE符（它们由于传输目的被4B/5B编码器编码成5个五位符号）组成。每个周期中剩余的12 480位被解释成一个1560字节串。前同步码之后是12个字节的**周期头部**字段，它的第一个字节用作**周期起始定界符**。头部中剩余的字节定义了信道如何组成当前使用的周期体（净荷）。周期体中的字节形成一个称为**专用包组**的12字节的单字段，和16个称为**宽带信道（WBC）**的96字节长字段。

如图10-2所示，周期体中的1548个字节由12个129字节组组成。每个组中的第一个字节形成专用包组（DPG0-11）的一部分。然后剩余的128个字节再被进一步分成16个8字节子组，每个宽带信道一个。将在10.2.3节中看到，它在使用每个信道时提供了更大的灵活性。

组成周期头部的12个字节被解释成24个4位符号，如前面指出的它在传输前由4B/5B编码器编码成五位符号。周期头部的字段如下（关于它们使用的进一步详细信息在后面的部分给出）： 563

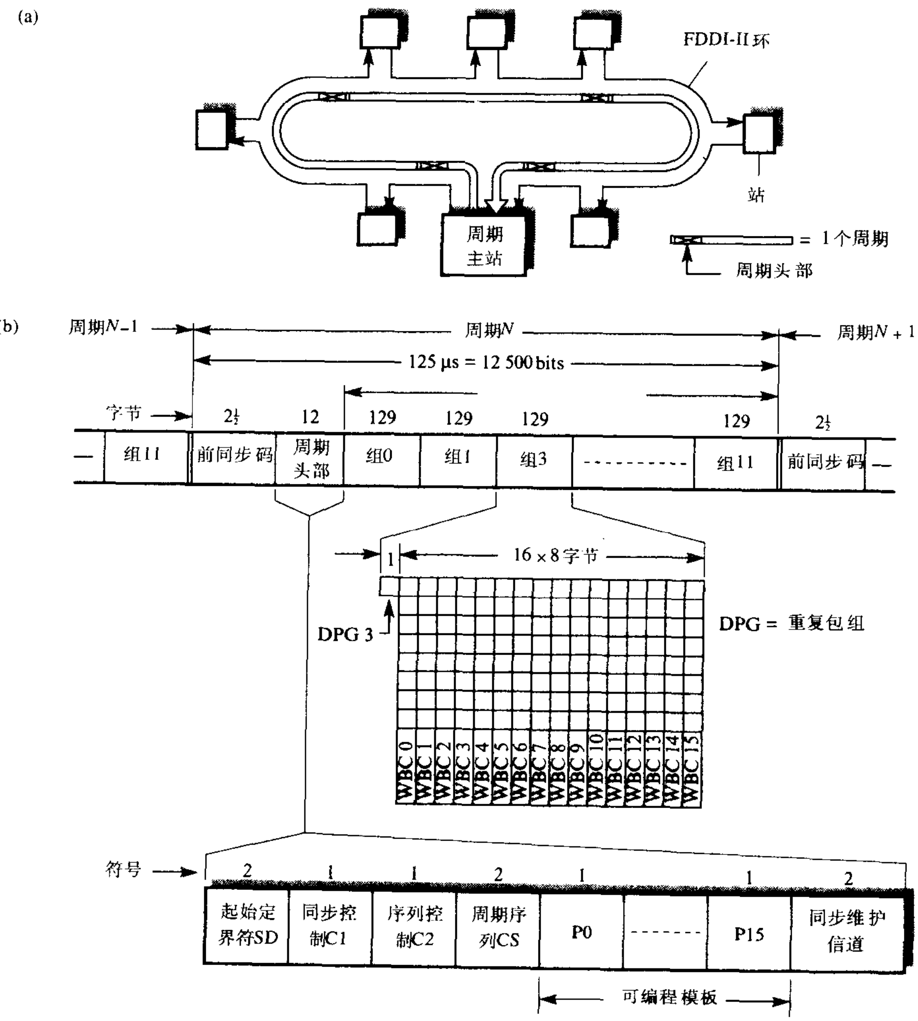


图10-2 FDDI-II

(a) 周期示意图 (b) 周期帧格式

- **起始定界符** 紧跟前同步码的2符号字段，用来说明新周期的开始。它由J-K符号对组成。
- **同步控制**（C1） 单个符号，说明环中所有站是否获得同步。它由未达到同步的站设成符号R，并由周期主站检测到所有站达到同步时设成符号S。
- **序列控制**（C2） 单个符号，它可以是S符号或者R符号。环设成混合模式的初始化过程
564 期间，站竞争成为周期主站。在这个过程中，每个站把符号R写入序列控制字段。一旦竞争过程结束并且周期主站被指定，周期头部中的周期序列（号）每收到一个后继周期就应该加1。当周期主站开始每个新周期时它把符号S写入序列控制字段，但是如果某个站检测到无效序列号，那么该字段重置成符号R。
- **周期序列**（CS） 2符号字段，解释成一个8位二进制数。它的范围为0～255。在初始化过程期间，它含有正在竞争成为周期主站的站的预指定号（称为**监控值**）。它的范围为0～63，并且具有最高值的站能从竞争中获胜。在正常操作期间，该值指明周期序号。后继周期增1，它们的范围从64～255。

- **工作方式** 它由16个符号组成，每个宽带信道一个，每个符号可以是符号S或符号R。符号R表示相应的信道用于异步数据，而符号S表示相应的信道用于同步数据。完整的工作方式由周期主站以响应来自各站同步带宽的请求指定。
- **同步维护信道** 2符号字段，已作为维护功能保留，其用法已经定义。

10.2.2 初始化过程

在初始化期间，每个站竞争成为周期主站。环中所有站先初始化成基本模式，然后每个站发出一个新周期（周期头部的C1和C2字段均为符号R，CS字段为预指定监控值）开始它的竞争。

如果某站收到监控值大于自己监控值的周期，那么它停止产生新周期，并开始转发收到的周期。最后，有最大监控值的站仍旧产生新周期而其他站都转发这些周期。当发送周期站开始接收有它自己监控值的周期时检测到这些周期，然后确定自己是周期主站。在这个时刻它开始发送C1和C2字段都是S符号的周期并增加CS字段中范围为64～255的序列号。当发送每个新周期时CS字段的值加1，并且当它达到255时重新从64开始。

10.2.3 带宽分配

16个宽带信道每个提供6.144Mbps（由于前面描述的多路复用方式能以许多方式应用）的传输带宽。实际的用途由组成每个128字节组的字节如何多路复用在一起决定。还能在位上通过多路复用提供较低比特率服务。说明如何获得站协议体系结构的示意图在图10-3（a）中给出，图10-3(b)给出了一些如何使用单一宽带信道的实例。任何不用于同步数据的宽带信道加到分组数据组上形成用于传输异步数据的单一较高比特率信道。正如指出的，基本分组数据由来自每个周期中的12个129字节组的单个字节组成。等同于768kbps的比特率，并能以6.144Mbps递增。 565 566

实例10-1

一个FDDI-II网络用作连接城市中5个接近的金融机构的骨干通信网络。在每个机构地点是专用（电话）交换机（PBX）、视频会议组和局域网（LAN）。在高峰（繁忙）期间相应位置最多可以同时有来自每个地点的46个语音呼叫。还有，必须作出规定，提供所有地点间永久视频会议链路。假定每个语音呼叫需要一条64kbps双工链路，并且对于每一组23个语音呼叫，每个PBX由于信令（呼叫建立）目的还需要一条64kbps双工链路。此外，每个地点间的视频会议链路需要384kbps。求支持语音和视频会议通信所需的带宽以及用于LAN互连的剩余带宽。

解：

语音 必须在每个地点的PBX和其他地点的PBX间建立双工信道。还有，必须假定在最坏情况下所有源自某个地点的呼叫可以到达另一个地点。因此每个信道必须能支持2×24（23个语音＋1个信令）个子信道，每个64kbps。

所有FDDI-II同步信道都是单一方向的（单工）并且因此需要两条信道建立一条双工信道。在所有5个PBX间建立双工信道所需的单工信道数由5×（5－1）给出：

1→2，3，4，5

2→1，3，4，5

3→1，2，4，5

4→1，2，3，5

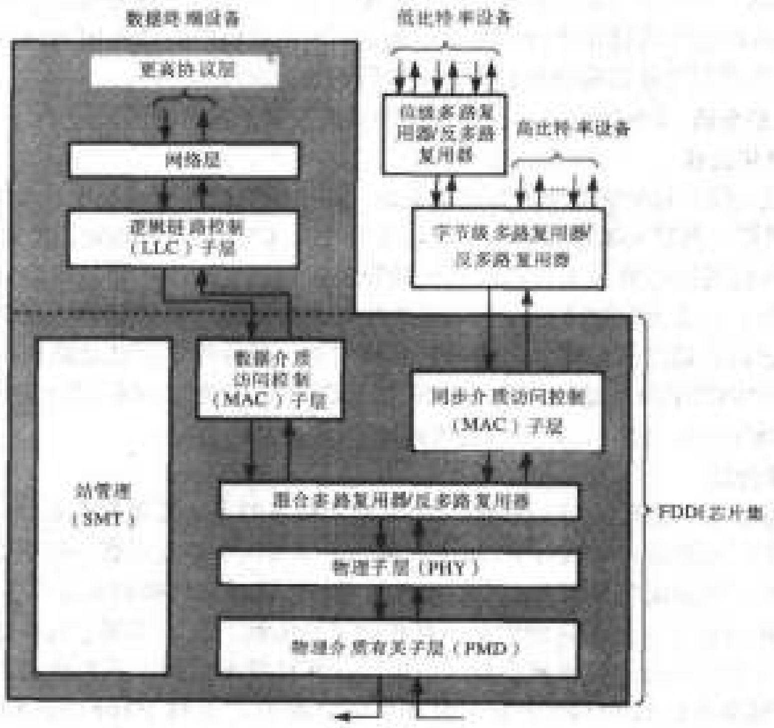
5→1，2，3，4

就是说，所需的单工通道数是20。因为每个信道必须支持2×24个子信道（每个64kpbs），语音通信所需的总带宽由

$$20 \times 2 \times 24 \times 64 = 61\,440\text{kbps}$$

得出，就是说10个宽带通道。

(a)



(b)

位/周期	复合的子信道速率（kbps）	使用实例
1/2/4	8/16/32	压缩语音或数据
8	64	未压缩语音信道
16	128	可视电话
48	384	视频会议
192	1536	电话（PBX）互连
256	2048	
768	6144	广播视频

图10-3　FDDI-II

(a) 协议体系结构　(b) WBC使用实例

电视会议　电视会议链路需要来自每个地点的视频源发送到其他地点。如同语音，它需要20个单工通道。因为每个通道是384kbps，视频会议通信所需的总带宽由

$$20 \times 384 = 7680\text{kbps}$$

得出，就是说2个宽带通道。

567 **局域网互连**　在每个地点会有连到区内骨干的单一连接点。例如网桥（如果所有地点是相同类型的LAN）。由此对于数据，每个地点间只需要单一方向（单工）链路以形成骨干环。

因为语音和电视会议通信需要12个宽带通道，剩余的4个宽带通道可以用于异步数据（LAN互连）。数据（分组数据组）的基本规定是768kbps。因此总可用带宽由

$$0.768 + 4 \times 6.144 = 25.344\text{Mbps}$$

得出。

10.3 信元网络

在10.2节中描述过，提供一种支持多媒体业务传输和交换的方法，先把不同的媒体源分离开来，然后为每种媒体提供支持。一种可选的方法是采用独立于源媒体但必须能灵活地支持任何一种媒体的传输和交换系统。它是信元传输和交换网络的基础。

历史上，使用预分配时隙传输和交换诸如数字化语音的恒定比特率通信。相比之下，数据通常以可变长度帧的形式（它在统计基础上被多路复用）传输和交换。在必须支持两种媒体类型的网络中已采用了混合方案。所有源媒体被分成称为**信元**的定长单元流。涉及不同媒体类型的信元流在统计基础上多路复用在一起用于传输，而采用适合所有媒体的标准长度信元意味着信元流的交换会以非常高的速率执行。由此产生的网络称为**信元网络**，或者**异步传输方式**（**ATM**）**网络**（因为涉及不同媒体的信元被统计地多路复用在一起用于传输，因此它们之间有随机时间间隔）。

下一个要做的决定是信元的长度。短信元有恒定比特率通信的优势，因为同一信元的后继字节在网络的外部设备被组装成信元和从信元拆分时经历较短的时延。反之，因为每个信元必须含有用于路由选择的额外信息，短信元有每个信元的相关开销高得不成比例的劣势。各个国际标准机构达成一致，选用了53个字节（8位组）的信元长度。它由48个字节信息（净荷）字段以及含有路由选择和其他字段的5个字节的头部组成。采用5个字节头部，首先因为信元不执行差错控制并且不需要用于重传目的的序列号，其次因为每个信元中不携带跨网络地址。注意在网络中采用信元传输和交换对于工作站和其他连到其上的设备的有关应用是透明的。信元简单地用来提供针对多媒体通信的更统一的传输和交换系统，并且所有源媒体和信元的转换在网络接口执行。已经开发了多种基于信元的网络类型，稍后部分讨论其中的一些。将会看到，所有网络类型都使用异步传输方式来传输。 568

10.4 ATM LAN

有关下一代多媒体网络应用的示意图如图10-1所示。这种网络必须提供下一代多业务工作站分布群体间的交换通信路径，以及到一类除数据服务外还支持许多更新服务的连网服务器的工作站访问。所以在一些情况下涉及数据、图像、语音和视频的网络通信需求有可能被综合在一起。显然，因为许多网络事务（呼叫）涉及语音和视频信息，它的时间敏感特性意味着必须为每个呼叫在网络中提供一条路径，并具有与之相关的保证传输延迟时间。

因此，多业务工作站的带宽需求显然比数据惟一工作站要高。由此，在一个互连多业务工作站的网络中，用于数据的介质共享拓扑类型是不适合的，因为在每个工作站提供非常高的性能接口的成本太高了。为了满足这种需求类型，ATM LAN使用若干个类似于现有电话网中的交换机互连组成的网状拓扑。一种典型跨设施ATM LAN的示意图如图10-4所示。

将在10.4.2节中看到，**ATM 交换机**有规定的端口数量并且其功能是在端口间提供较高传输率的交换通信路径。交换机的成本与它支持的端口数量有关。如果所有工作站直接连到交换机，那么当多业务工作站的部署增加时需要有大量端口数的交换机。但是，所有工作站并不是同时需要网络服务。为了使端口数最小化，工作站组（比如在一幢大楼中）通过**远程集中器**（RCU）连接到交换机。在RCU中没有交换功能，它的作用只是把来自网络事务工作站的信元流多路复用到连接集中器到交换机端口的链路上，以及把来自连接集中器到交换机端口链路上的信元流反多路复用到网络事务工作站。假定互连链路有足够的容量来支持预期的并发事务数量，所需的端口数量会大大减少。 569

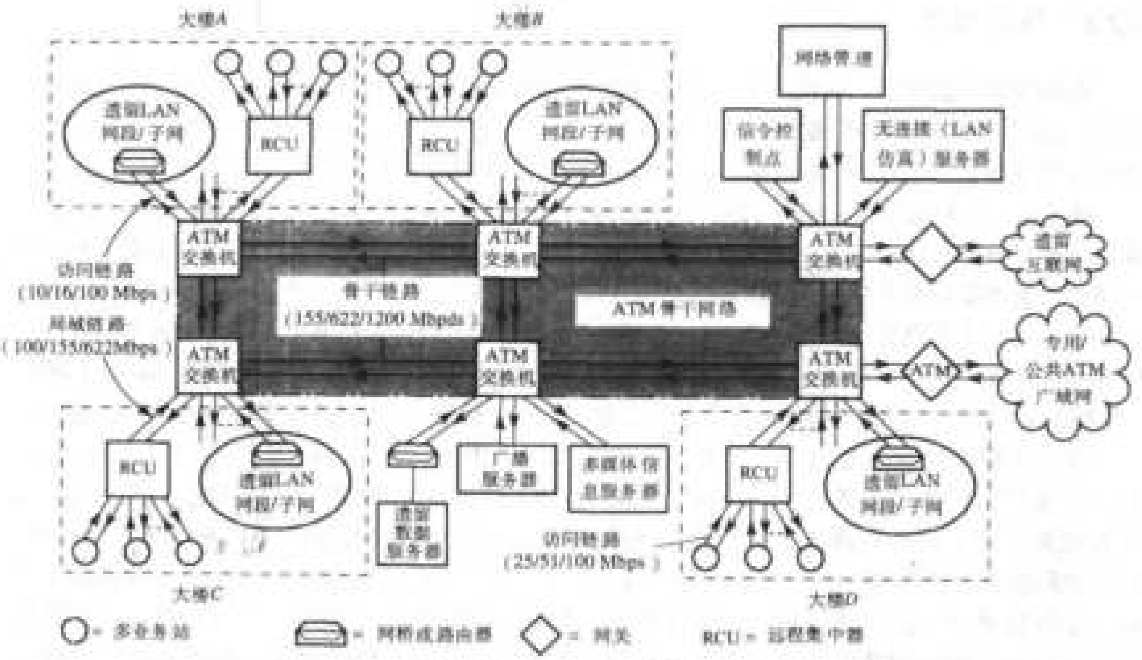


图10-4　ATM LAN示意图

在发送与某个呼叫相关的信息前，首先应在网络中建立通信路径。所有与该呼叫相关的信元被限制沿着这条路径并且以它们提交的顺序传输。回忆一下，所有信息转换成信元流有个优点，就是不同媒体类型的信元可以以统一方式交换并且独立于与它们相关的媒体类型。另外，信元的使用在传输带宽的利用方面有优势。

如10.1节中讨论的，为了减少图像和视频所需的带宽量，源信息在传输前先被压缩。它对于视频尤其有利，因为以未压缩形式每个视频帧含有定量信息，该定量信息以帧刷新率（例如25或30帧每秒）决定的恒定速率生成。但是，使用压缩的话只有后继帧间发生的变化才被传输，其影响是每帧传输的信息量会显著地从少量信元（如果只有很少变化）到大量信元（发生完整的帧变化）变化。当为呼叫保留传输容量时利用信元传输率的变化，它以平均传输
570 率保留而不是以最大传输率保留。为了确保每个呼叫可获得足够的容量，不同呼叫的信元流会在统计的基础上多路复用在一起。网络中的连接/路径称为**虚拟连接**（VC），术语“虚拟”说明连接是逻辑意义上的而不是物理连接。

多业务工作站支持的连网服务包括电话、可视电话、传统数据连网以及对相关服务器的访问。除了那些在现有网络中存在的（电子邮件、打印服务等）服务器外，还包括广播服务器、支持多媒体信息包的数据库服务器等等。例如，**广播服务器**使得多业务工作站的用户通过实时地直接发送来自所有参与工作站的视频输出到服务器，在三个和更多工作站间按需建立电视会议会话。当会议进行时服务器中继相应的视频图像（带声音）到其他工作站。类似地，持有多媒体信息包的服务器能得工作站的用户能访问特定包，然后通过它交互地工作。

当然，多业务工作站是被逐渐引入的，并且当前绝大多数LAN应用仍然是纯数据类型。在ATM LAN环境中，它们一般称为**遗留LAN**。这类网络的主要瓶颈是对服务器的访问，因为需要很大带宽以支持多个并发事务。由此骨干交换机除了提供对较新多媒体服务器的直接访问外，还支持对每幢大楼中现有纯数据LAN（通过网桥或路由器）以及与它们相关服务器的

连接。通常，它们通过一组点对点VC互连。在现存LAN关系中，骨干交换机的互连集被看作能提供与高速骨干子网相同的功能。

ATM网络的所有通信在先前建立的VC上执行。它们由用户按需建立或者由网络管理半永久的建立。使用**按需连接**，用户设备在发送任何信息信元前要发送建立**交换虚连接**（SVC）（在它与所需目标间）的请求给称为**信令控制点**（SCP）的中央控制单元。它负责传输带宽和网络中交换连接建立和清除的整个管理。接到称为**信令信息**的请求，SCP先判断所需的目标以及适合传递该呼叫的网络的传输带宽是否可用。所有的工作站通过独立VC连到SCP（通常是个功能强大的工作站），假定所需的目标和网络资源可用，SCP在连接参与呼叫的两个用户设备（VC）的交换网络中建立路由选择信息。然后通知请求发送者可以开始发送信息的信元。有关建立和清除呼叫/连接的所有信令信息通过为这个功能永久建立的独立VC集，以信元方式 571
经过网络传输给SCP或从SCP传输出来。后者称为**信令虚通道连接**（SVCC）。

将在第三部分看到，为了访问诸如电子邮件和多媒体信息服务器等连网服务器，使用诸如TCP/IP的协议栈。回忆第9章，IP提供无连接最佳尝试服务。由于ATM 网络是面向连接的，在可以传输任何数据报前VC必须在每个工作站和一组服务器间存在。实际上，服务器的数量可以很大，并且能通过在所有工作站、所有服务器和一台称为**无连接服务器**（CLS）（因为它提供类似于遗留（广播）LAN提供的服务又称为**局域网仿真服务器**（LES））的中央数据转发点之间建立**永久VC**（PVC）满足需求。将在10.4.6节更详细地讨论它和SCP的操作。

有关各种服务的永久虚拟连接都是在**网络管理站**的统一控制下建立。保留VC在网络管理站和每个骨干ATM交换机以及RCU中的控制处理器之间永久地存在。此外还存在于所有多业务工作站、服务器和它们的网络连接点之间。网络管理方使用它们来建立有关各种服务的VC。网络管理者通过这些连接下载路由选择信息到这些设备持有的路由选择表中。将会在10.4.1节中研究这些连接的建立过程。

在现有纯数据网络中，多业务工作站的用户除了跟同一地点的其他用户通信外，还与从不同地点连到ATM LAN上的用户通信。因此，ATM LAN用**网关**连到现有（遗留）互联网以及较新的专用/公共广域ATM网络。为了满足这些需要，引入了基于ATM的新一代专用网络。此外，公共载波引入了基于相同技术的新一代公共网络。它由ATM MAN组成，在将来会连到ATM WAN。由此产生的网络称为**宽带综合业务数据网**（BISDN）。

10.4.1 信元格式和交换原理

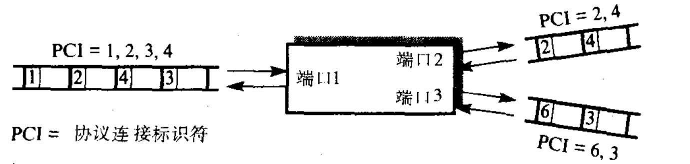
涉及某个呼叫的所有信元沿着网络中先前建立的相同VC传输。在建立规程中，给网络中每条链路上的连接分配一个**协议连接标识符**（PCI）。它用来确定和路由网络中每条链路上的相同呼叫/连接的信元，并确定网络外部设备的每个呼叫的不同信元流。但是，分配的标识符对于链路只有本地意义，当某个呼叫连接的相关信元通过每个交换机从一条链路转到下一条链路时会改变。这意味着每个信元头部携带的路由选择信息相对较小。路由选择方案的原理如图10-5(a)所示。 572

每条输入链路/端口有一张路由选择表，它为每个输入PCI对应一个输出链路/端口和要用的新PCI信息。这样两个方向上的信元路由选择都会非常快，因为只涉及简单的查询操作。因此，来自每条链路的信元可以以非常高的速率独立交换。它允许使用并行交换体系结构和千兆比特范围内的高速传输链路（每条链路以其最大速率工作）。

实际上，PCI由两个子字段组成：**虚通路标识符**（VPI）和**虚通道标识符**（VCI）。能使用其中一个字段和两个字段的组合来执行路由选择。图10-5给出了两个实例。在图10-5(b)中，

在虚通路上执行交换并且每条虚通路中的VCI不变。在图10-5(c)中，在每条虚通路内的虚通道上独立地执行交换，并且虚通路简单地终止于每个交换端口。

(a)



链路/端口1 RT

输入		输出	
端口	PCI	端口	PCI
1	1	2	2
1	2	2	4
1	3	3	3
1	4	3	6
•	•	•	•

链路/端口2 RT

输入		输出	
端口	PCI	端口	PCI
•	•	•	•
2	2	1	1
•	•	•	•
2	4	1	2
•	•	•	•

链路/端口3 RT

输入		输出	
端口	PCI	端口	PCI
•	•	•	•
3	3	1	3
•	•	•	•
3	6	1	4
•	•	•	•

RT = 路由选择表

(b)　(c)

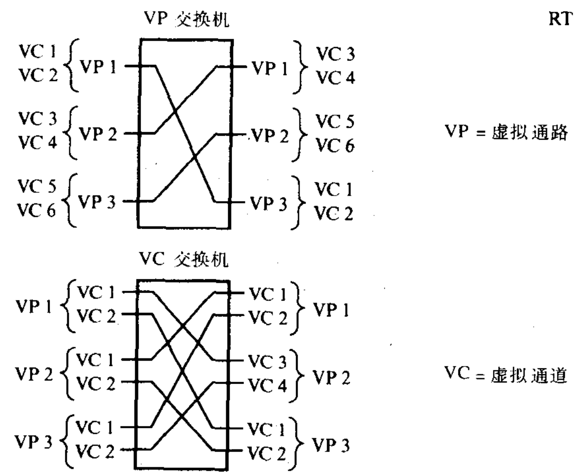


图10-5　信元交换原理

(a) 路由选择示意图　(b) VP路由选择　(c) VC路由选择

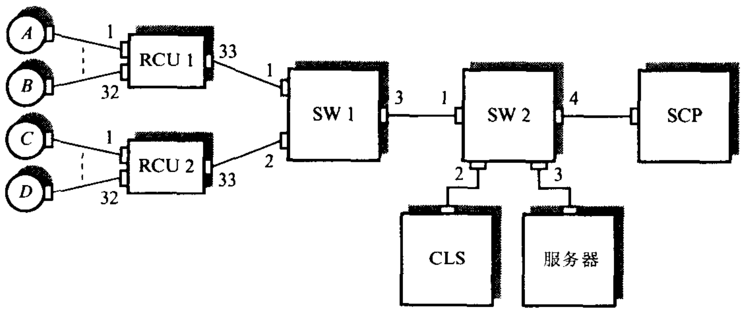
使用虚通路交换的一个实例是来自相同网络入口点的多个呼叫都要指向相同的目标出口
573 点。为每个呼叫分配一个单独的VCI，并且源接口上的呼叫被多路复用到单一虚通路上。呼叫的多路复用集只使用VPI字段交换并且都沿着网络中的相同通路进行。这样，VCI集保持不变并在目标方用来确定（反多路复用）独立的呼叫。

使用虚通道交换的一个实例是网络入口点的每个呼叫指向不同的目标。这种情况下，分配给每个呼叫不同的VCI，但是仍然使用VCI字段执行交换。VPI字段对每条链路只有本地意义，用来允许呼叫多路复用在一起传输。

实例10-2

图10-6(a)给出了ATM网络的一段。每个RCU/交换机旁边的数字是端口标识符。假定在站*A*、*B*、*C*和*D*间网络管理首先要为按需呼叫（用于信令和信息传输）建立到SCP的半永久VC，其次为无连接呼叫相关的信元流建立到CLS/LES的半永久VC。此外，假定需要单独的VC连接服务器到CLS/LES。在网络交换机中使用VP惟一交换，并且在RCU中使用VPI/VCI来确定特殊呼叫/站，得出RCU1、RCU2、SW1和SW2的典型路由选择表记录。

(a)



(b)

RCU 1:

	输入			输出			输入			输出		
	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI
SC	1	0	1	33	1	1	33	1	1	1	0	1
CC	1	0	2	33	2	1	33	2	1	1	0	2
CLS	1	0	3	33	3	1	33	3	1	1	0	3
SC	32	0	1	33	1	32	33	1	32	32	0	1
CC	32	0	2	33	2	32	33	2	32	32	0	2
CLS	32	0	3	33	3	32	33	3	32	32	0	3

RCU 2:

	输入			输出			输入			输出		
	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI
SC	1	0	1	33	1	1	33	1	1	1	0	1
CC	1	0	2	33	2	1	33	2	1	1	0	2
CLS	1	0	3	33	3	1	33	3	1	1	0	3
SC	32	0	1	33	1	32	33	1	32	32	0	1
CC	32	0	2	33	2	32	33	2	32	32	0	2
CLS	32	0	3	33	3	32	33	3	32	32	0	3

SW 1:

	输入			输出			输入			输出		
	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI
SC	1	1	X	3	1	X	3	1	X	1	1	X
CC	1	2	X	3	2	X	3	2	X	1	2	X
CLS	1	3	X	3	3	X	3	3	X	1	3	X
SC	2	1	X	3	4	X	3	4	X	2	1	X
CC	2	2	X	3	5	X	3	5	X	2	2	X
CLS	2	3	X	3	6	X	3	6	X	2	3	X

X = 1–32

SW 2:

	输入			输出			输入			输出		
	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI	端口	VPI	VCI
SC	1	1	X	4	1	X	4	1	X	1	1	X
CC	1	2	X	1	Y	X	1	Y	X	1	2	X
CLS	1	3	X	2	3	X	2	3	X	1	3	X
SC	1	4	X	4	4	X	4	4	X	1	4	X
CC	1	5	X	1	Y	X	1	Y	X	1	5	X
CLS	1	6	X	2	6	X	2	6	X	1	6	X
CLS'	2	10	Z	3	10	Z	3	10	Z	2	10	Z

X = 1–32
Y = 2/5
Z = 1–64

图10-6　路由选择实例

(a) 网段　(b) 路由选择记录实例

CLS:

输入		输出	
VPI	VCI	VPI	VCI
3 6	X X	10 10	Y Y

输入		输出	
VPI	VCI	VPI	VCI
10 10	Y Y	3 6	X X

X = 1-32
Y = 2/5

SCP：呼叫进行中

信令通道		呼叫通道						呼叫类型细节
		输入			输出			
VPI	VCI	端口	VPI	VCI	端口	VPI	VCI	
1 4	X X	1 1	2 5	X X	1 1	Y Y	X X	– –

SC = 信令通道　　CC = 呼叫通道
CLS = 工作站/CLS通道　　CLS' = CLS/服务器通道

图10-6 （续）

路由选择表记录的适用集在图10-6(b)中给出。解释这些记录时注意以下要点：

- SCP、CLS和服务器都通过单独传输链路连接到它们的交换机。
- 对于按需呼叫，所示的两条单独VC：一条在每个工作站和SCP间用于与该呼叫相关的信令信息（信令通道（SC）），另一条用于与该呼叫相关的信元流（呼叫通道（CC））。因为后者是半永久的，它们如图所示在每个工作站和SW2间建立。另一种情况是，它们可以由SCP按需在涉及呼叫的每对工作站间建立。 574~575
- 对于无连接通信，在每个工作站和CLS间以及在CLS和服务器间需要单独VC。
- SCP、CLS和服务器都在信元头部使用组合VPI/VCI来确定涉及特定呼叫/服务器事务的信元。
- 在每个RCU的工作站方，端口号定义了每个工作站，VCI定义了特定VC。
- 在每个RCU的网络方，VCI字段在每条虚通路内定义了端口号（工作站）。此外，在这个例子中，每个RCU需要三条虚通路而不是每个工作站。它允许升级到大型应用。
- 在网络内，所有交换只使用VPI执行。
- 为了建立按需呼叫，SCP在SW2的路由选择表中建立主叫方端口/VPI/VCI到被叫方端口/VPI/VCI的连接的记录。
- 对于无连接通信，把每个工作站收到的信元流中继到服务器时，CLS分配新的VPI/VCI。此外为了在反方向中继信元流，它维持一张表把来自工作站的输入VPI/VCI定向到与服务器通信的VPI/VCI。
- 当响应请求时，服务器为信元使用与请求中相同的VPI/VCI值组成响应。

每个信元的格式如图10-7所示，头部由六个字段组成。它们的功能如下：

- **一般流量控制**（GFC） 只在用户—网络接口（UNI）传输的信元内存在，并包括用来使本地交换机/RCU能调整（流量控制）成通过用户进入网络的信元入口。在网络内，通过交换链路（称为网络—网络接口（NNI））传输的信元中，该字段不存在，这4位是
576 VPI字段的一部分。

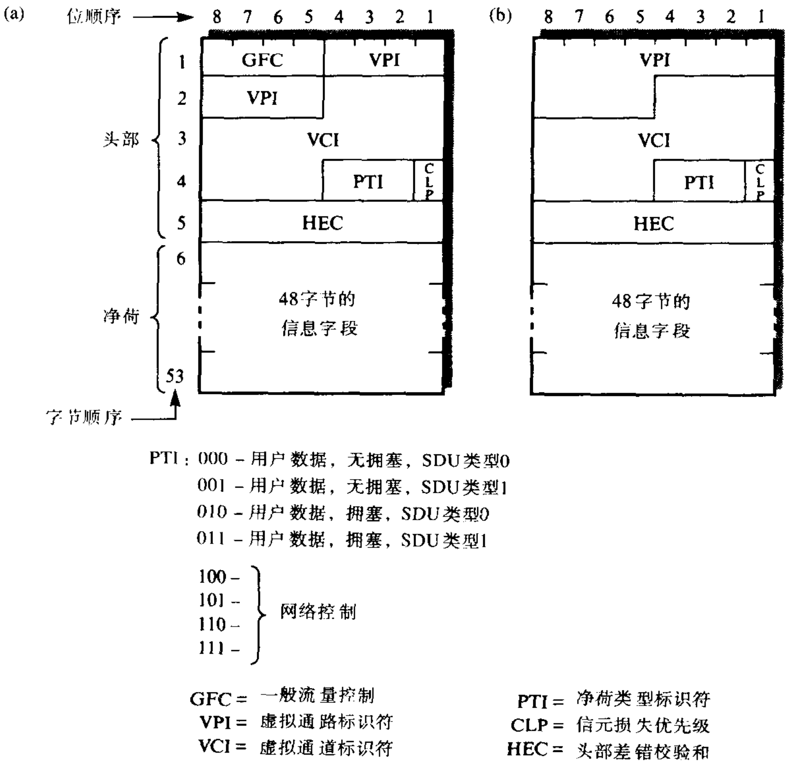


图10-7 ATM信元格式

(a) 用户—网络接口 (b) 网络—网络接口

- **虚通路标识**（VPI） 它在UNI中是8位，正如刚指出的在NNI中是12位。如先前描述的，它用于网络内的标识/路由选择。
- **虚通道标识**（VCI） 16位字段，用于网络内的标识/路由选择。
- **净荷类型指示**（PTI） 说明信元内携带的信息类型。各种类型如图10-7所示。所有含有用户数据的信元最高位为0。下一位说明信元是否遭受过度时延/拥塞，第3位指示服务数据单元（SDU）类型——0或者1。将在10.4.4节讲解有关AAL 5的服务时讨论它的使用。4个其余信元类型用于网络控制。
- **信元丢失优先级**（CLP） 在网络内，每条链路上对信元统计基础上的多路复用可能偶尔会导致在严重负载情况下必须丢弃信元。包括该字段使得用户能确定哪些信元更应该先被丢弃。CLP=0为高优先级，CLP=1为低优先级并由此先被丢弃。
- **头部差错校验和**（HEC） 由物理层产生并且是在头部前4个字节的8位CRC。 577

10.4.2 交换机体系结构

ATM 交换机的主要结构如图10-8(a)所示。每条输入链路由**输入控制器**（IC）终止，它执行每条链路（端口）到所需输出链路的信元路由选择。根据流入信元头部的VPI/VCI进行相应输出VPI/VCI的简单查询和映射操作。通常，从路由选择表获得的输出端口号用来确定通过交换结构到所需输出控制器要经过的通路。

因为在输出链路不保留时隙，信元可以在需要相同输出端口/链路的两条（或更多）输入链路同时到达。它以两种方式处理：输入控制器拥有保留额外信元的一组信元缓冲区，或者

在输出控制器提供缓冲。在两种情况中，缓冲区以FIFO队列的方式组织以确保来自每个输入控制器的信元按它们到达的顺序输出。输出控制器简单地以适当的链路比特（信元）率转发收到的信元。

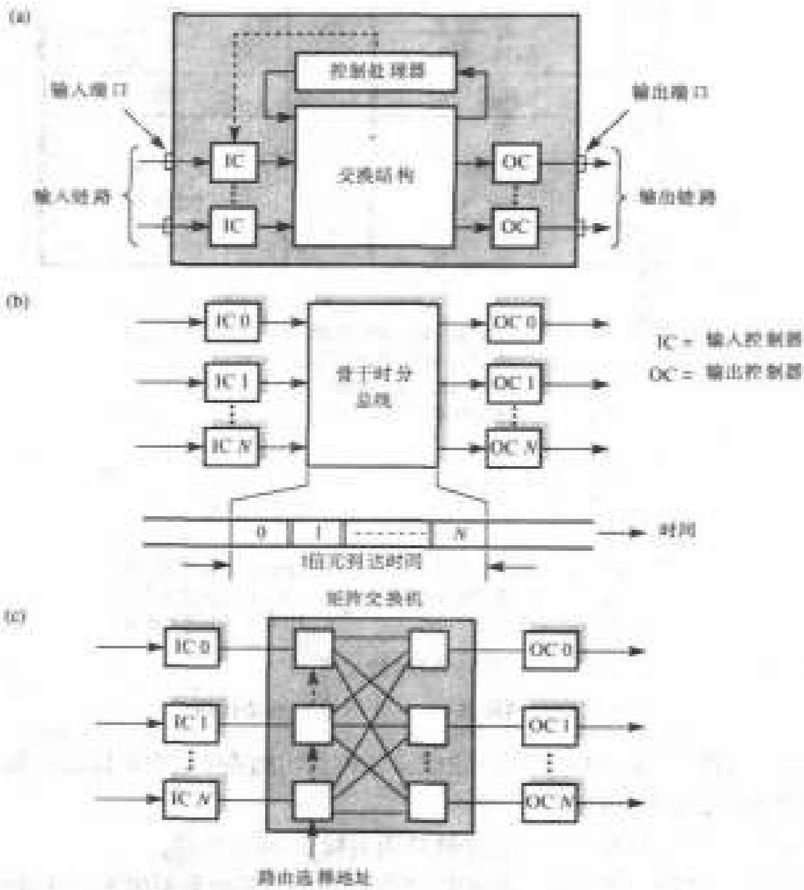


图10-8　ATM交换机体系结构

578 (a) 一般结构　(b) 时分总线示意图　(c) 全连接矩阵交换机

控制处理器的主要功能是下载路由选择信息到每个输入控制器中的路由选择表中。通常，通过网络从网络管理或者从信令控制点处理器接收路由选择信息。在两种情况中，使用半永久VC并且相关信元到达交换机时直接通过交换结构从接收信元的输入控制器路由到控制处理器。此外控制处理器本身可以产生网络管理信息（故障报告、性能统计等），它们通过交换结构被路由到所需输出控制器并进一步传输到网络管理处理器。

许多交换结构应用在ATM交换机中。它们分成时分和空分两种。通常，所有的输入控制器同步，这样来自所有控制器的流入信元能同步地提交给交换结构。交换结构也同步地工作，意味着来自每个输入控制器的信元在单信元周期内传输到它们所需的输出控制器。

时分交换的示意图如图10-8(b)所示。在这种交换机中，使用时分底板总线，它能在单一信元到达时间内传输N个信元（这里N指输入端口的数量）。分配给每个输入控制器自己的信元（时隙）周期来通过底板总线传输信元。输入控制器附加所需的输出端口号到信元头部，由输出控制器组来确定哪个输出控制器应该读取和缓冲该信元。如果在一个单信元到达时间

内输出控制器收到多个信元，那么它们排在控制器的队列中。此外，对于一对多通信（多播），可以指定多个输出控制器来接收信元。通常，这种交换结构用在交换机设计中，它有相对较少的端口数，数量由底板总线和输出控制器的工作速度限制，例如2.5Gbps总线能支持16 × 155Mbps或者4 × 622Mbps双工链路/端口。

在空分交换中，交换结构由互连的**交换单元**矩阵组成，交换单元共同提供若干条通过交换机的可选通路。图10-8(c)给出了一个实例。称为**全连接交换矩阵**，因为它提供了从所有输入端口/控制器到所有输出端口/控制器的通路。在所示的实例中，每个输入交换单元能传递每个收到信元的副本给任何输出交换单元。然后后者接收提供的信元并把它们传递给输出控制器来传输。虽然排队在输入和输出控制器中是必需的，但其目标通常是为了防止交换结构内额外的排队。为了防止这种交换机的内部排队，信元传输操作必须以N倍信元到达速率执行，这里N指输入端口数。首先来自第一个输入交换单元的信元被传输，然后是来自第二个输入交换单元的信元，依此类推。假定这能做到，那么在交换矩阵中不需要额外的缓冲，交换机可以说是内部**无阻塞的**。 579

从图10-8可以推出，在全连接交换机中需要通过交换机的互连通路（以及与每个交换单元相关的输出/输入电路）的数量以N^2增长，并且输出交换单元的工作速度以N增长。实际上，它限制了这种交换机的最大规模，因此大多数交换矩阵设计使用多级交换，每级由许多在规则矩阵内互连的较小交换单元组成。它还大大简化了以集成电路形式实现交换结构。

由多级交换组成的交换结构是**delta交换矩阵**，它的实例如图10-9所示。可以看到，这些交换机由相同的交换单元互连组组成。这个例子使用的是2 × 2交换单元，虽然可以使用更大的规模。通常，每级的交换单元数量X由公式$X = M/N$决定，M是输入线路的总数量，而N是每个交换单元的输入数量。此外，所需的级数Y由公式$N^Y = M$决定。在这个例子中，$N = 2$而$M = 8$，因此需要三级，每级由4个交换单元组成。

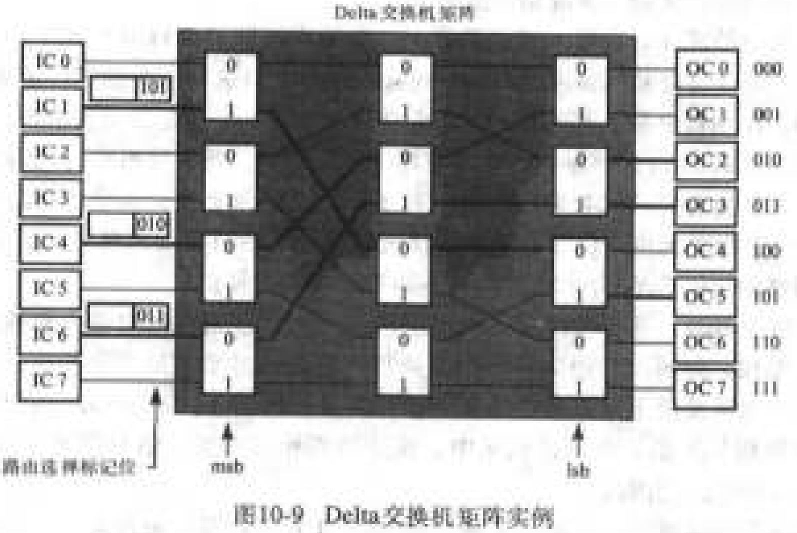


图10-9 Delta交换机矩阵实例

交换单元的内部互连是这样的，通过交换机从任何输入到任何输出都有通路。与每个交换单元相关的是称为**路由选择标记**的路由选择控制。如果标记是0，那么到达任何输入端口的信元路由到上面的输出端口，如果标记是1，那么路由到下面的输出端口。可以从图10-9推出，3个路由选择标记位的相同组会通过矩阵路由信元从任何输入端口到相同的输出端口。这种网

络可以说是**自主路由选择**。

使用这种交换机，为了通过交换机矩阵路由信元，每个输入控制器简单地从它的路由选择表中读取所需的输出端口号/地址并把它附加到信元头部。然后沿着通路经过矩阵的每级交
580 换单元从路由选择标记中使用各自的位（最高位在先）来执行它的路由选择操作。这样，路由选择非常快并且每个信元都能到达它指向的交换机目标端口。

这种交换结构的缺点是**阻塞**。通过交换机的3个通路实例如图10-9中粗线所示。可以看到，虽然地址指向端口5（101）的信元无阻碍地经过矩阵，但是那些地址指向端口2（010）和端口3（011）的信元同时到达第二个交换单元。它们都需要相同的输出线路，称为发生阻塞。实际上，这种矩阵的性能会由于大量端口数而迅速降低。

有许多方法用来克服阻塞。一种方法是交换单元丢弃两个信元中的一个。为了支持这种方法，注意在每个信元到达时不是所有端口都把信元输入矩阵，因为只有在最严重负载情况下信元才会在所有输入端口同时到达。这意味着在正常负载下，多个端口会接收空闲/空信元，它不需要路由并且实际上阻塞的概率很小。但是，以这种方式丢弃信元通常会导致不可接受的高信元丢失率。

第二种方法是以多倍信元到达速率执行交换操作，例如在前面的信元离开第一级交换单元后允许每个信元进入矩阵。对于交换单元和它们的互连链路的操作速度有个限制，因此对于较大型交换结构它变得不可行。第三种方法是给每个交换单元引入缓冲，但是它有个缺点，就是会给交换操作引入额外的时延。实际上，在实际交换机设计中使用这三种方法的组合。

防止内部阻塞的交换结构的实例如图10-10所示。它称为**Batcher-Banyan 交换机**。使用delta交换机，当不同输入都需要经过交换机到达相同输出通路时，或者不同输入和输出端口间通过交换机的通路涉及来自交换单元的共同输出线路时发生阻塞。在Batcher-Banyan交换机中，首先确保没有两个进入交换机矩阵的信元需要相同的输出端口，其次在交换矩阵内确保在交换机的通路内没有共同互连链路来避免阻塞。

为了满足第一种情况，在输入控制器而不是在输出控制器执行信元缓冲。如果两个（或更多）信元同时到达需要相同输出端口的不同输入端口，只选择一个信元传输通过交换结构，而另一个在输入控制器中排队直到下一个信元传输周期。为了满足第二种情况，如图10-10所示，在交换矩阵（Banyan）前加一个**排序矩阵**（Batcher），两者使用称为**洗牌互换交换机**互
581 连。组合的结果是所有到达交换矩阵的信元被排序，这样每个信元都沿着交换矩阵的惟一通路，并且在每个交换单元内提供独立路由选择通路，使来自所有输入端口的信元能同时交换。Batcher排序网络的规模以N（$\log N^2$）增长并可用于大型交换机。

为了降低阻塞概率，用于实际交换机设计的另一种方法是在每对输入和输出端口间有多条通路的交换结构。例如，使用简单delta交换机，它能通过重复适当次数总交换机矩阵来获得。每个输入控制器使用不同矩阵传输每个信元。另一种选择，基本Banyan交换矩阵可以通过增加额外交换级别扩展。在两种情况中，能选择额外的级数，这样阻塞（以及信元丢失）概率能在一个可接受的范围内。

另一个影响交换机设计的问题是多播。回忆9.5.1节，它需要来自某个（多播）组内每个工作站的所有信元发送给组内其他所有工作站。实际上，做到这一点的最有效方法是使用交换机路由信元副本给多个目标。虽然可以使用时分和全连接交换机设计轻易地完成，但是，使用矩阵交换机，由于它们的自主路由选择属性，如果支持多播需要额外的交换级别。关于这两个问题的进一步细节可以在参考文献中找到。

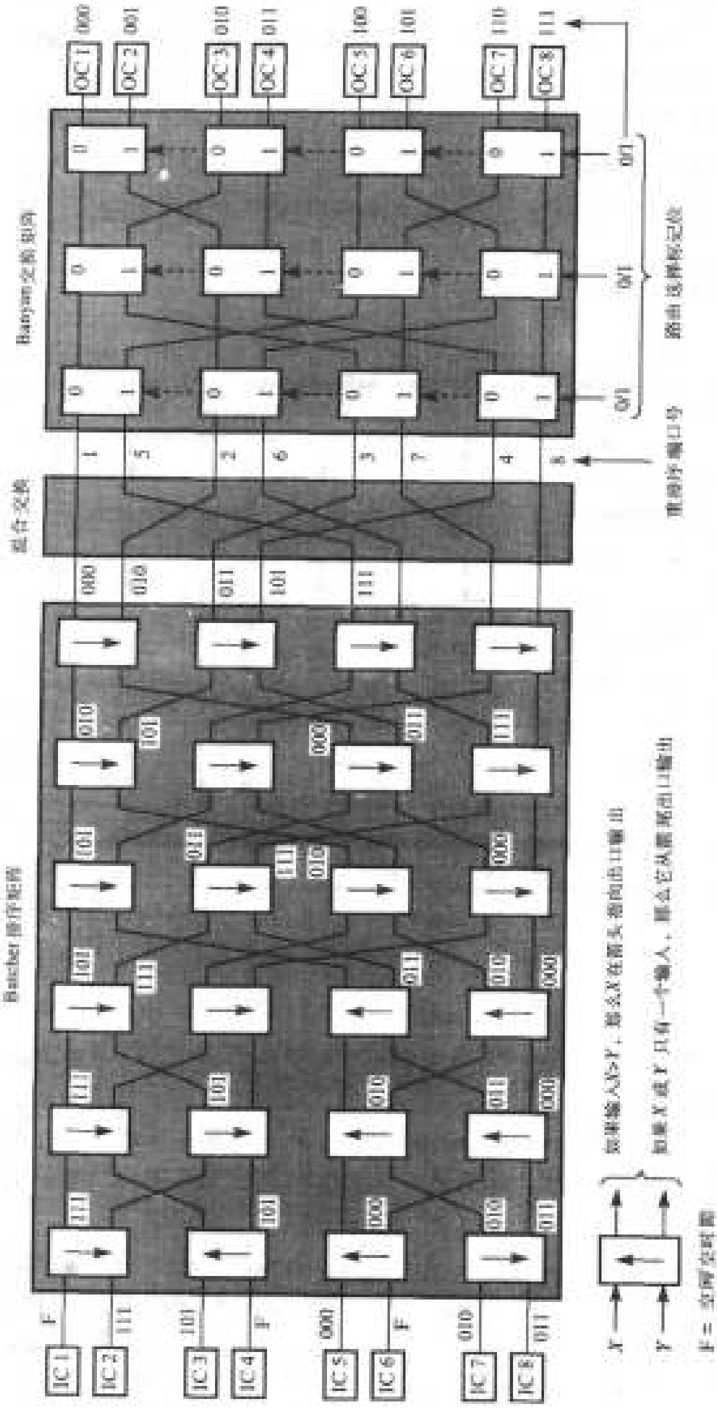


图10-10　Batcher-Banyan 交换机矩阵

10.4.3　协议体系结构

如图10-11所示，ATM 协议体系结构支持三个单独应用功能（平面）。它们是控制（C）平面、用户（U）平面和管理（M）平面。有关**C平面**的协议关注信令方面，就是按需VC的建立和清除。通常，使用站内的信令协议组与设在网络信令控制点的类似协议组通信来建立和清除它们。**U平面**内依赖应用的协议通常在用户对用户（对等）基础上与目标站内的类似协议通信。**M平面**内的协议关注站的管理，包括报告任何在对网络管理站的正常操作期间可能产生的差错状况以及接收到已分配给PVC的虚通路/通道标识符的确认。这三个应用功能使用由三个较低ATM协议层提供的服务来经过基于信元的ATM交换网络传输相关信息。

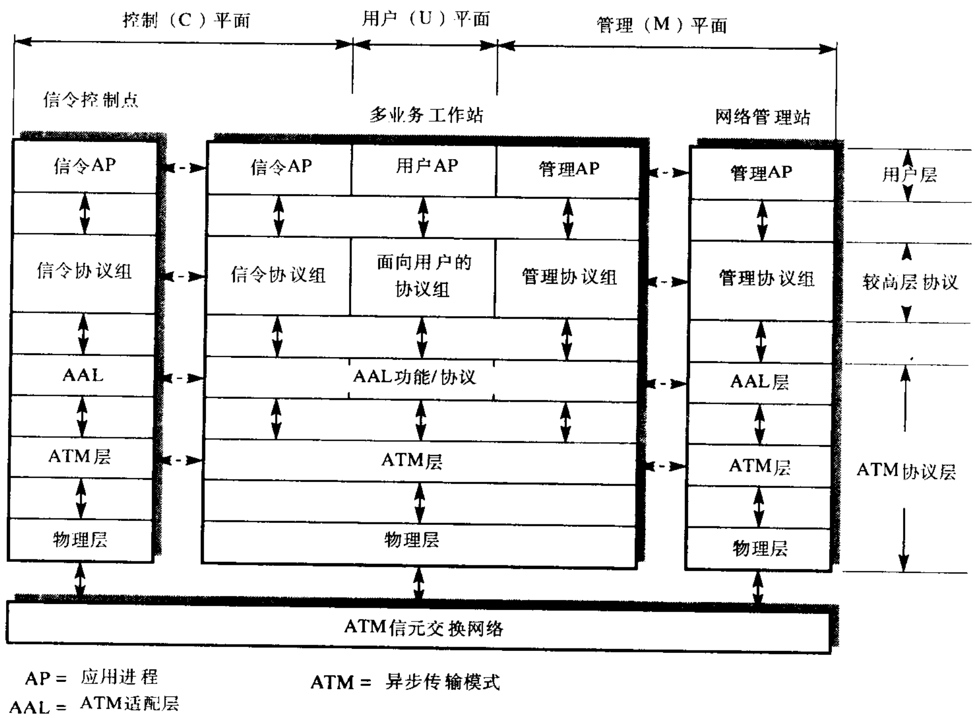


图10-11　ATM协议体系结构

ATM网络支持一类不同的服务。网络内信元交换和传输的使用对于更高层应用协议是透明的，它把ATM网络简单地看作用户传输任何媒体类型相关信息的灵活设施。为了达到透明性，三个ATM协议的最高层称为**ATM适配层**（AAL）。正如名称所指的，它在提供给用户层的服务类（例如在两个遗留LAN间传输数据帧）和由基础ATM层提供的基于信元服务间执行适配（会聚）功能。

582~583

为了支持各种信息源，AAL提供称为**服务类**的一类可选服务类型。每个服务类是把源信息转换成48字节段流的不同适配功能/协议。它把这些传递给**ATM层**用于通过网络传输。ATM层负责插入正确的信元头部信息到每个段，并把不同连接的信元多路复用成在网络上传输的单一信元流。它还负责反多路复用接收的信元流并中继它们的内容到目标的适当AAL协议。

物理层具有很多形式并依赖于使用的传输电路类型。物理层的较高部分称为**传输会聚子层**，它负责在信元头部生成头部校验序列以及描述信元边界等功能。较低部分具有不同的形

式，称为**介质依赖子层**，它负责线路编码和位/时钟同步等功能。我们将会详细地讨论AAL和ATM层的操作。 584

10.4.4 ATM适配层

AAL提供一类可选的服务类型/类，用于传输与U平面、C平面和M平面相关的各种较高协议层产生的字节流/信息单元。它把提交的信息转换成48字节段流并把它们放在多个ATM信元的净荷字段传输。接到相同呼叫的信元流，它把每个信元中含有的48字节信息字段转换成所需形式并递交给特定高协议层。

服务类型根据三个标准分类：源和目标用户间时间关系（例如语音）、传输相关的比特率（恒定的或可变的）以及连接方式（面向连接的或无连接）。当前，已定义了五种服务类型。基于这些标准，它们称为AAL 1～5，它们的相互关系显示在图10-12(a)中。

AAL 1（类A）和AAL 2（类B）都是面向连接的并且在源和目标用户间有定时关系。两者的不同之处在于AAL 1提供**恒定比特率**（CBR）**服务**，而AAL 2提供**可变比特率**（VBR）**服务**。使用AAL 1的例子是用于传输语音通话的恒定比特率字节流，例如每125μs秒1个字节。AAL 1也称为**电路（交换）仿真**。使用AAL 2的例子是传输与压缩视频相关的可变比特率流。 585
虽然视频以恒定速率产生帧，但是视频编码器会产生含有可变量压缩数据的帧。

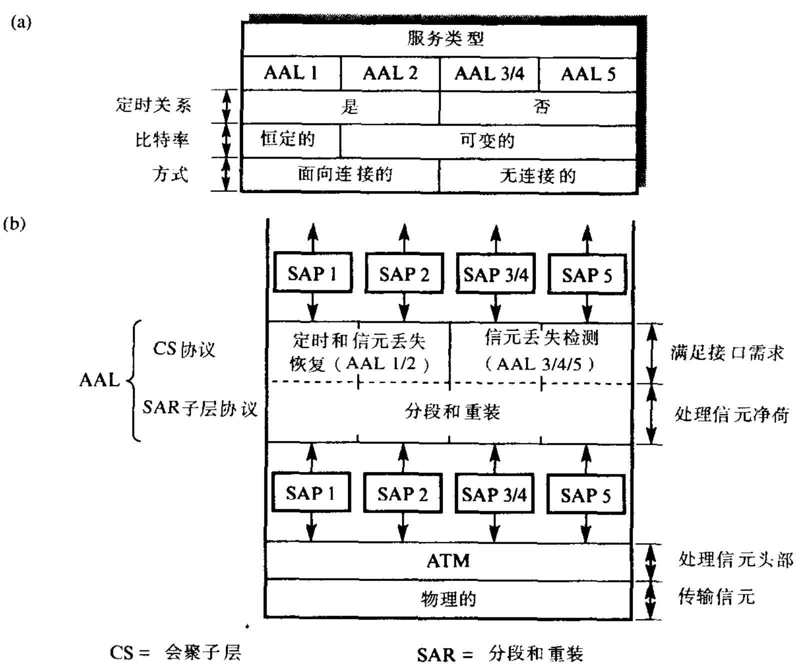


图10-12 ATM适配层

(a) 服务类关系 (b) 子层协议及其功能

AAL 3/4（类C/D）和AAL 5在源和目标间没有定时关系。最初，定义AAL 3提供面向连接的VBR数据服务。后来不使用这种服务类型，现在把它跟AAL 4融合在一起。AAL 3/4和AAL 5都提供无连接VBR服务。使用的例子是传输两个遗留LAN间的数据帧（通过远端网桥或路由器）或者在多业务工作站和服务器间传输含有多媒体信息的帧。它还用于传输与这些工作站内的信令和管理协议集相关的信息单元。所有这些情况将在10.4.6节看到，虽然提供的

服务是无连接的，但是由此得到的由AAL产生的信元流通过先前建立的PVC传输到信令控制点或网络管理站。

为了实现这类服务，AAL由如图10-12(b)所示的两个子层组成。**会聚子层**（CS）在层接口提供的服务与基础ATM层提供的服务间执行会聚功能。**分段和重装**（SAR）**子层**对准备放在信元48字节净荷字段传输的源信息分段以及在目标端递交源信息前执行相应的重装功能。

提交的信息因每种服务类型而不同，所以每种服务类型有不同的会聚功能（以及不同的CS协议）。与每个协议相关的是**服务访问点**（SAP），用来把所有要传输的提交信息（服务数据单元（SDU））映射到适当的CS协议。类似地，有四种不同的SAR协议，每一种有自己的PDU结构。并以不同的方式使用每个信元内的48字节信息字段。接下来开始讨论每种协议的操作。

1. AAL 1

对于这种服务，CS协议努力在源和目标SAP间维持恒定的比特率流。比特率的范围从每秒几千位（比如压缩语音通话）到每秒几十兆位（比如未压缩视频）。但是，必须维持恒定的速率，甚至当偶然的信元丢失或者信元传输周期发生变化时。信元丢失能以协商的方式克服，例如通过插入虚构位/字节到传输流。信元传输时延变化通过在目标缓冲分段来补偿：涉及某个呼叫的位/字节输出只在接收到预定义段数后才开始，这个数量由用户比特率决定。一般在
586 千位速率为2段而在兆位速率为100段。在目标端使用缓冲还提供了克服源接口输出速率与目标端输出速率间微小变化（例如每个速率基于单独的时钟）的一种自然方式。对于输入和输出时钟的更好解决方案由网络提供。

与每个SAR协议相关的PDU格式如图10-13(a)所示。为了检测到段丢失，第一个字节含有4位序号并且相关4位保护字段用来保护序号不出现单比特差错。序号本身由3位序号计数字段（用于检测信元丢失）和一个会聚子层指示位组成。后者用于定时和/或其他关于净荷字段信息的传输。序号保护字段由一个3位CRC（由多项式x^3+x+1生成）和一个偶校验位组成。后者用于检测CRC中的差错。

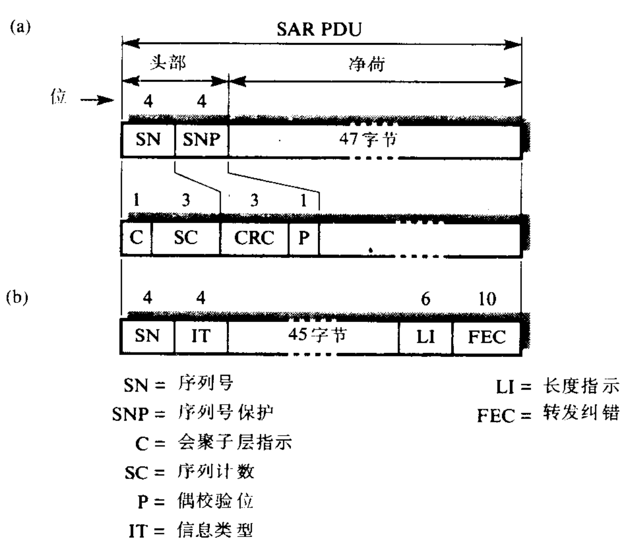


图10-13　SAR协议数据单元类型

2. AAL 2

这种服务类型虽然在源和目标SAP间有定时关系（由帧速率决定，例如对于压缩视频），但是每个压缩帧的信息量会随帧变化。源SAP的CS协议以帧速率接收到突发信息，每次突发含有可变的信息量。因此在目标SAP中对等的CS协议必须以相同的方式努力输出接收到的信息，甚至当信元偶然丢失或者信元传输周期发生变化时。AAL 2使用那些描述用于AAL 1的类似技术能克服时间变化。

与SAR协议相关的PDU格式如图10-13(b)所示。和AAL 1一样，存在序号用于检测丢失信元（以及从丢失信元恢复）和携带定时信息。信息类型字段指出段相对于提交信息单元（例如压缩帧）的位置或者段是否含有定时信息或其他信息。关于位置信息的三个段类型是报文开始（BOM）、报文继续（COM）和报文结束（EOM）。此外，由于提交信息单元的长度可变，最后（EOM）一个段不必是满的并且在尾部的长度指示（LI）中指出段内有用的字节数。最后，FEC字段使得能检测到位差错并纠正选择。 587

3. AAL 3/4

AAL 3初始时被定义成能提供面向连接的数据传输服务。后来，这种服务不再使用并和AAL 4组合。AAL 3/4提供用于传输最长65 535字节的可变长度帧的无连接数据传输服务。每个帧传输前把差错检测和其他字段加到帧上，由此结果帧被填充，这样它是32位的整数倍。

最好考虑用CS和SAR协议两者都相关的PDU格式描述AAL 3/4的操作。由CS协议加到每个提交用户SDU的额外字段（在相应的SAP）的格式如图10-14(a)所示。该图还给出了由此产生的CS PDU如何由SAR协议分段成多个48字节SAR PDU。

在源SAP由CS协议加到提交SDU头部和尾部的字段由目标SAP的对等CS协议用来检测任何丢失或变形SDU。PDU类型字段是早先AAL 3的遗留，它需要多种PDU。使用AAL 3/4它设成0。开始—结束（BE）标记是个模256的序号并在尾部为附加弹性重复。它使得SDU能在用户接口以提交的相同序列传输，再次说明，这种设施通常用作无连接服务。缓冲区分配（BA）字段由源插入到头部来帮助目标端的CS协议分配适当的（缓冲区）存储空间量给完整SDU。在尾部，填充字段用来使完整CS PDU中的字节总数成为4字节的整数倍。类似的，对齐（AL）字段是个单（虚构）字节以产生尾部4个字节。它们共同使得在目标端的存储空间管理更容易。长度字段说明完整PDU的总长度，它由接收协议用来检测任何变形的SDU。

接到每个CS PDU，SAR协议把它分段成多个如图10-14所示的48字节段（SAR PDU）。在头部，段类型（ST）说明段是第一个段、中间段、最后一个段或者CS PDU分段产生的惟一段。序号（SN）用来检测丢失段。实际上，像网络服务器的设备可以有许多从多个源同时接收到的帧（以及许多CS PDU）。因此为了使接收SAR协议把每个流入段关联到正确的PDU，源SAR协议增加相同的多路复用标识符（MID）到与相同CS PDU有关的每个段的头部。在尾部，长度指示（LI）字段说明段内有用字段的长度，因为CS PDU不必是44字节段的整数倍。显然，该字段在涉及多段CS PDU的最后一个段或者惟一段时有意义。CRC用来检测段传输期间可能引入的传输差错。 588

4. AAL 5

由于AAL 3/4的起源，CS PDU在头部含有许多字段，主要用来支持面向连接的服务。每个SAR PDU头部的MID字段（它使得目标能把帧关联到特定帧）也执行与每个ATM信元内的协议连接标识符（VPI、VCI）类似的功能。因此，定义了多种AAL 5服务类。它提供了与ALL 3/4类似的服务但是在CS和SAR PDU中控制字段数减少了。它称为**简单有效适配层**（SEAL）。和在AAL 3/4中一样，最好考虑用与CS和SAR协议相关的PDU格式描述它的操作。 589

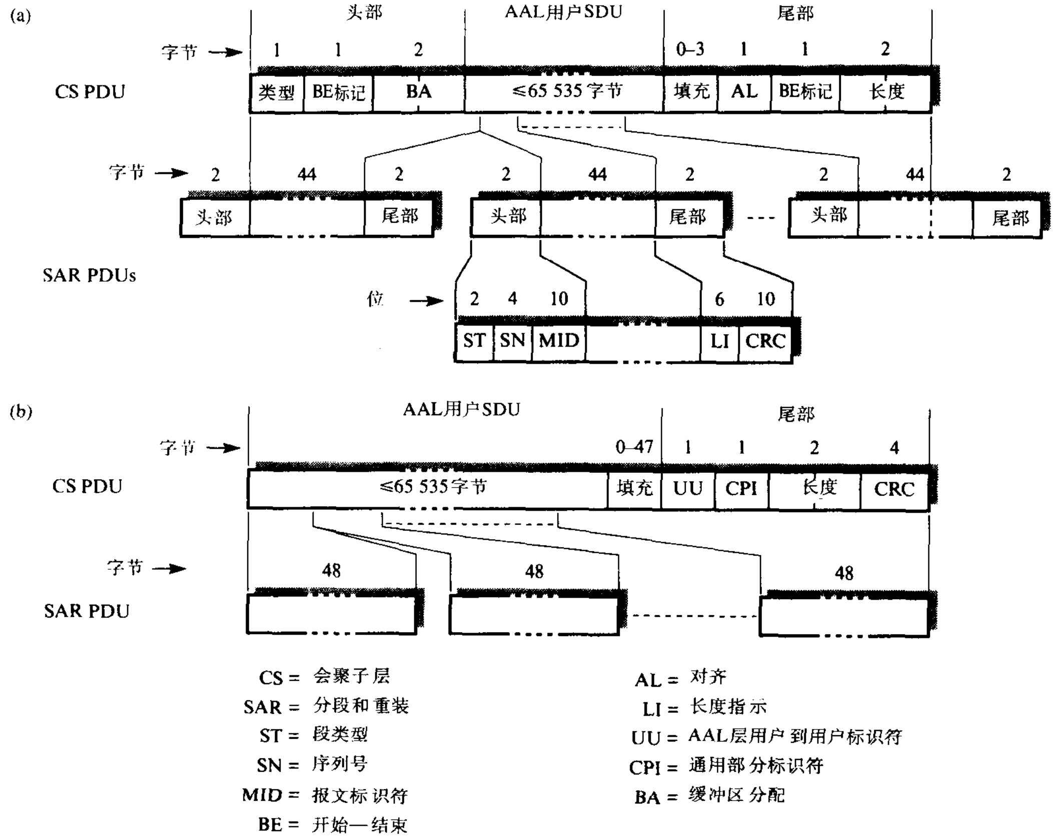


图10-14　CS和SAR PDU格式

可以看到，AAL 5没有构造CS PDU相关的头部。此外，尾部的填充字段更长（0～47个字节），这样每个CS PDU 的长度可以是一个完整的48字节段的整数。它有个优点，CS PDU尾部的字段永远是最后段的最后8个字节，这样导致在目标方更快的处理速度。AAL 用户对用户标识符（UU）使得两个对应用户层能把AAL SDU关联到特定SAP。还定义了CPI字段的使用，当前它使得尾部是字节的偶数并支持将来扩展的功能。长度字段说明了用户数据字段中的字节数，它是范围0～65535的整数值。CRC字段检测在重装CS PDU中任何传输差错的存在。如果检测到差错，那么当包含在PDU内的SDU传输时通知上面的用户层。这样，留给用户层来决定采取什么措施：一些情况下会丢弃SDU（如果它含有正常数据）而在另一些情况下会接受它并采取适当的恢复步骤（如果SDU内容涉及多媒体文档的视频或语音部分）。

正如看到的，SAR PDU没有相关的头部和尾部。每个由48字节组成并且SAR协议可以为空。在每个段头部缺少MID字段意味着可以使用传输段的ATM信元头部的字段来确定涉及相同CS PDU的段。使用AAL 5，用户数据信元内的SDU类型位（还称为ATM用户对用户信元）用来说明信元内容是否形成CS PDU的开始或继续（二进制0）或者结束（二进制1）。虽然这意味着AAL的操作现在要与ATM层联系起来，但是提高了分段处理的效率。AAL 5还用作C平面内的AAL，用于有关信令协议的信息的分段和重装。它又称为**信令AAL**（SAAL）。

10.4.5　ATM 层

ATM 层执行有关VC（它可以是半永久或者按需建立的）上信元路由选择和多路复用的

所有功能。它的主要功能是分配头部给由与特定呼叫相关的AAL产生的段流。类似地，接到信元流，它的功能是从每个信元中除去头部并把信元内容（段）传递给相应的AAL协议。

为了执行这些功能，ATM层保持一张含有VCI列表的表。通常，在ATM LAN中只使用VPI执行所有交换，信元头部中的VCI字段用来多路复用/反多路复用在相同通路上传输的特定 590
呼叫/事务的信元。在半永久VC的情况下，VPI由网络管理通过网络管理协议栈下载，而在按需VC情况下使用适当的信令协议集包含VPI。在两种情况下，有关管理/信令协议信息的信元流在永久分配VC上传输。

10.4.6　呼叫处理

从图10-4以及相关文本看到，有两种关于ATM LAN的通信类型：在多业务工作站分布式群体与它们相关服务器间交换的信息流的通信以及在各种遗留LAN段/子网的网桥/路由器与它们相关服务器间交换的信息流的通信。此外，在多业务工作站情况下，有两种呼叫：一种面向连接而另一种无连接。第一种涉及电话和可视电话等网络服务，而第二种涉及与遗留LAN中类似的更传统的数据服务。下面分别讨论这两种类型。

1. 面向连接呼叫

诸如电话和可视电话业务，需要在呼叫期间在任何一对工作站间建立单独VC。这种业务类型的标准基于ISDN用来建立呼叫的标准，因为原则上建立和清除呼叫的操作与ISDN中执行的功能相同。所有建立和清除呼叫的（信令）信息在单独通道（信令虚拟通道连接（SVCC），它独立于那些用来传输与呼叫相关的报文的通道）上传输。当第一次配置网络或者增加新出口（outlet）时，由网络管理在每个网络出口和中央信令控制点间建立一条半永久VC。类似于在每个电话出口与本地（电话）PBX间建立物理线路连接。

当SVCC由网络管理建立时，也在SCP的路由选择表中建立一条记录。它由工作站的ATM网络地址以及分配的路由选择地址（就是用在由SCP从/到整个工作站接收/输出的信元头部内的VPI/VCI地址）组成。每个工作站的网络地址类似于电话出口的地址，并且是惟一定义整个网络关系中工作站的分级地址。如8.2.3节所示，它们有标准格式，在ATM LAN中是20字节的NSAP地址。

为了启动呼叫的建立，工作站的用户进行对话，为呼叫目标接收者指定地址以及呼叫类型（电话、可视电话等）。这使得在工作站内使用信令协议组和在SCP计算机中的类似协议组 591
交换信令报文。支持它的协议体系结构如图10-11所示，并且信令协议定义在Q.2931建议中。它基于早先在8.4.6节描述的ISDN用户—网络信令协议。接到呼叫请求，SCP使用信令协议组与被叫工作站用户通信来确定用户是否准备接收呼叫。如果响应是肯定的，它返回给SCP，SCP在路由选择表中增加一条主叫工作站的端口/VPI/VCI到被叫工作站的端口/VPI/VCI的记录来建立网络的（双工）VC链路。然后SCP通知主叫工作站VC已经建立并开始呼叫。以类似的方式，每个用户可以在任何时刻在单独信令通道发送断开连接报文以发起连接清除。

为了建立有多个工作站参与的呼叫（例如电话会议或视频会议会话），所有工作站必须是完全互连的。对于少量工作站它可以由SCP直接使用类似用于两方呼叫的规程来完成。因为在某些情况下参与的工作站数量可能很大，一个可选的方法是通过称为广播服务器的中央路由选择点来路由所有涉及这种呼叫的信息流。

使用这种方法，为了建立会议会话，会话发起者像前面一样使用信令通道与SCP通信，但是指明会议呼叫（和它的类型）以及参与的工作站地址。SCP又使用它们的信令通道和相关协议集与每个工作站的用户通信，来确定它们的可用性以及是否愿意参与会议会话。接到它

们的响应，SCP开始在每个返回肯定响应的工作站和广播服务器间建立交换VC。然后首先通知广播服务器呼叫类型以及参与的交换VC组，其次通知工作站用户可以开始会话。广播服务器中继接收自每个工作站的信息流到所有其他工作站来继续会话。在工作站用户控制下它们安排接收到的信息，比如每个视频会议会话成员使用一个单独窗口。

2. 无连接呼叫

正如在10.4节开始时看到的，现有纯数据工作站由提供更大服务范围的多业务工作站逐渐替代。考虑无连接工作，主要问题是不仅要在ATM（基于信元）工作站间交互，还要在工作站与现有纯数据工作站间交互。

592 正如在第7章看到的，多数大型遗留LAN由若干网桥互连的相同类型LAN网段组成。另一种情况，如果网段类型不同（正如在第9章看到的），网段由路由器互连。回忆一下，当使用网桥时，所有路由选择在MAC子层执行，而当使用路由器时，路由选择在IP层（OSI栈中的CLNP层）执行。MAC子层和IP层都为预配置帧或数据报的传输提供最佳尝试无连接服务。为了支持ATM工作站间和连到遗留LAN的工作站间的交互，必须支持两种方案：一种用于桥接LAN，另一种用于基于路由器的LAN。第一种，ATM LAN与遗留LAN间的接口是网桥，而第二种，接口是路由器。使用两种不同的协议体系结构，每种都提供在相应环境下的无缝交互。下面分别讨论它们。

（1）局域网仿真

网桥作为遗留LAN的接口，已被生产LAN连网设备的一组公司开发。这个组称为**ATM论坛**。图10-15(a)给出这种方式的各个联网组成部分的示意图。

各个联网组成部分的目标是在面向连接的ATM LAN上模拟遗留LAN的广播工作模式。这种方式是**LAN仿真**（LE）。它的三个组成部分是**LE 配置服务器**（LECS）、**LE 服务器**（LES）和**广播/未知地址服务器**（BUS）。虽然每个组成部分在图10-15中都表示成单独实体，但是它们可以在一台计算机中实现。在这种情况下，每个组成部分的报文/帧由它们到达的虚通道连接（VCC）的类型确定。

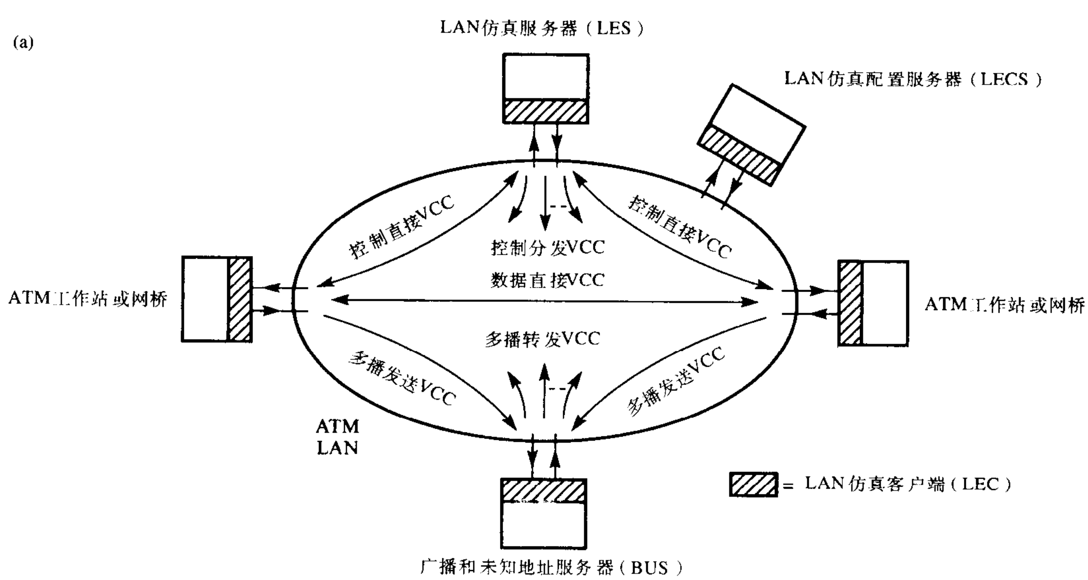
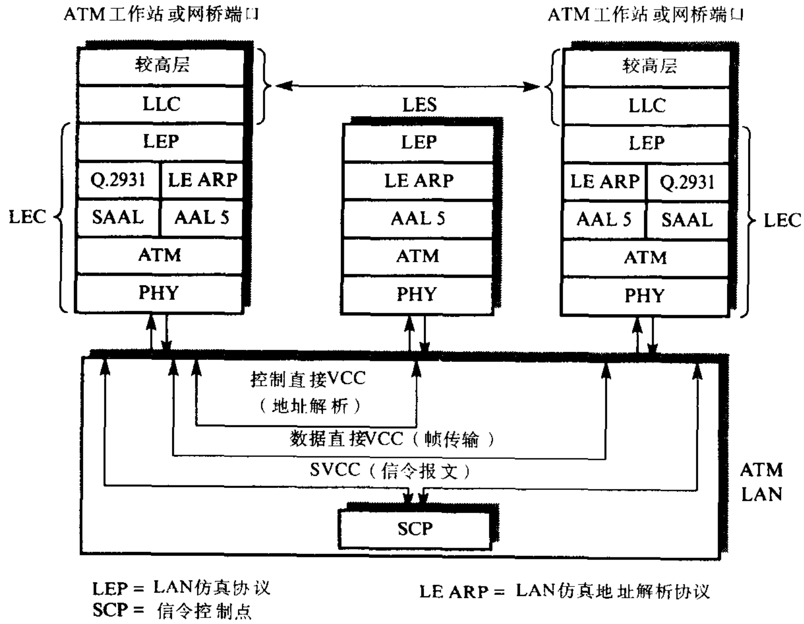


图10-15 LAN仿真

593 (a) 术语和连网组成部分 (b) 单播协议体系结构 (c) 多播协议体系结构

(b)



(c)

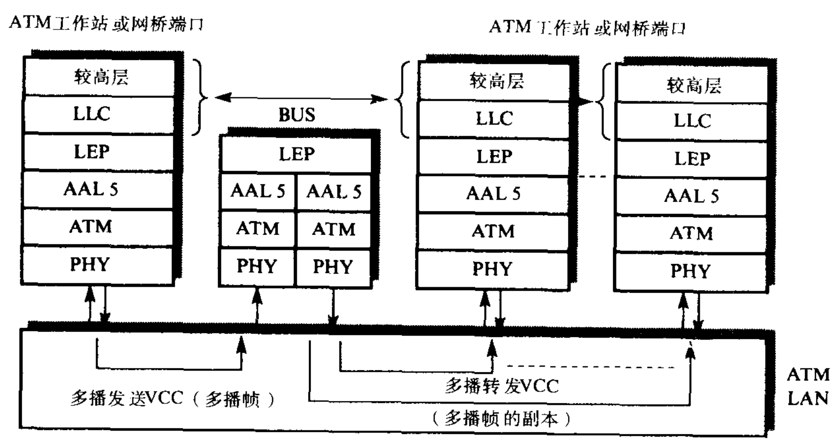


图10-15 （续）

在大型设施中可以有多个仿真LAN，每个有自己的LES和BUS。连到整个ATM LAN的所有站用LECS来确定LES和BUS的ATM地址。为此，用保留VCC来连接所有站到LECS。LES提供地址解析服务来把传统的48位MAC地址转换成20字节ATM地址。BUS首先负责支持多播/广播，其次负责中继帧到LES未知MAC地址的站。 594

所有连到ATM LAN的ATM工作站和网桥有一个**LE客户机**（LEC）子系统。它由连到遗留LAN的工作站内的MAC芯片集以及相关软件提供的相似服务的硬件和软件组成。当某个站通电时，它在执行初始化规程后初始化某些工作参数和自己的地址集（如下描述，一个站有多于一个地址）。在图10-15(a)中看到，独立的VCC对用来连接每个LEC到LECS、LES和BUS。一旦初始化完毕，LEC从LECS获得LES和BUS的VCC对，然后继续在LES和BUS上注册自己。当整个初始化阶段结束后，LES和BUS已经有了所有连接到ATM LAN上的活动站的地址集。

在网桥情况下，LEC可以注册连到遗留LAN端口的所有站的MAC地址，或者只注册连到ATM LAN时经过的端口的地址集。在第一种情况网桥工作在**非代理模式**而在第二种情况网桥工作在**代理模式**。

从图10-15(b)看到，每个LEC内的协议栈包含**LAN仿真协议**（LEP）层（紧挨LLC层的下面一层），它与LES中的类似层通信。为了使下面的网络对于LLC子层透明，LEP提供的服务类似于MAC子层提供的无连接服务。两个用户服务原语是LE_UNITDATA.request和LE_UNITDATA.indication。

为了仿真广播LAN，两个原语都把源和目标MAC地址作为参数。这意味着每个连到ATM LAN的工作站和网桥除了一个20字节的ATM 地址外，还有一个与它相关的48位MAC地址。还有2字节的**LEC标识符**（LECID），它用来在那些当前连到LES的工作站和网桥中惟一地确定ATM工作站或网桥端口。会在稍后讨论LECID的功能。

（逻辑上）连接LEC到LES的VCC称为**控制直接VCC**。接到LE服务请求原语，源LEC中的LEP读取源和目标MAC地址，并把它们传递给**LE地址解析协议**（LE ARP）。LE地址解析协议生成地址解析请求报文（含有两个MAC地址和LEC的ATM地址），并在控制直接VCC上把它发送给LES中的LE ARP。假定LES中的LE ARP知道目标LEC的ATM地址，它把地址放在应答报文中返回给请求LE ARP，然后后者开始在自己和目标LEC的LE ARP之间建立**数据直接VCC**。直接由参与的两个LE ARP使用SCP和前面描述的面向连接呼叫的信令协议集，或者一些ATM交换体系结构来完成这一步。在规定的超时时间间隔内同一个目标LEC没有接收到接下来的帧时，连接由LE ARP清除。

595 如果LES中的LE ARP没有目标LEC的ATM地址（例如，目标MAC地址涉及连到网桥的遗留LAN端口的站），LES使用**控制分布VCC**集发送LE ARP请求报文副本给所有注册的LEC。然后知道目标MAC地址的LE ARP使用控制直接VCC以相应的ATM地址或者自己的ATM地址（如果它是网桥的话）回复。接着LES中的LE ARP再次使用相应的控制直接VCC中继该回复给源LE ARP。一旦源LE ARP获得目标的ATM地址，它建立数据直接VCC。

然后开始传输数据帧（它们的头部都有源和目标MAC地址）流。因为涉及到网桥，它们可以是802.3帧或802.5帧（帧类型定义在帧头部）。不需要FCS字段，因为在ATM LAN中它由AAL 5子层执行。由于中间地址解析阶段的使用，LE无连接服务可以说是非直接提供的。

前面的情况涉及单播，就是说对于每个提交的MAC帧只有一个目标站。为了实现多播，必须转发生成的MAC帧的副本给属于同一个多播组的所有站。显然，使用刚才描述的方法，在每个组成语和其他所有成员间需要建立多条交换VCC。为了避免这种要求，使用了BUS，相关的协议体系结构如图10-15(c)所示。

在每个站的LEC和BUS中的LEC间使用一对额外的VCC。在站到BUS方向，VCC称为**多播发送VCC**而在反方向称为**多播转发VCC**。接到有多播目标地址的服务原语，LEP产生一个同前面一样的帧，但是在头部有它自己的2字节LECID。它直接通过多播发送VCC发送该帧到BUS中的LEP。接到该帧，LEP使用多播转发VCC集广播该帧的副本给所有站。每个站的LEP先从帧头部的LECID确定该帧是否由它发出。如果是，那么LEP简单地丢弃该帧。这称为**回显抑制**。如果LECID不与站的LECID匹配，接收LEP从帧头部的多播地址中确定本站是否是多播的成员。如果是，该帧被向上传递给LLC层，如果不是，该帧被丢弃。

与BUS相关的未知地址服务使得LEP能在数据直接VCC建立期间发送有限数量帧到它们所需的目标。遵循的规程跟多播帧的规程一样，并且因为每个帧的多个副本（每个经过单独的转发多播VCC）被发送，对LEP能发送的这类帧的数量要加以限制。然后任何接收到的后续帧必须被保留（存储）直到数据直接VCC就位。

（2）在ATM上的传统IP

将路由器作为遗留LAN的接口，这种方法已经由**因特网工程任务行动组**（IETF）开发。基本方法称为**在ATM上的传统IP**。 596

回忆第9章，在工作站能与另一个工作站交换数据报前，它必须先使用IP（或CLNP）知道本地路由器（中间系统）的IP/NPA（CLNP/SNPA）地址对。此外，本地路由器必须知道连接到它连接的LAN网段的所有工作站的地址对。使用ARP（IP）或者ES到ES（CLNP）协议以及利用遗留LAN的广播特性来获得这些。一旦获得这些信息，所有数据报通过路由器/IS在两个站间交换并且使用对应数据报头部中IP目标地址的NAP地址（例如MAC地址）中继这些数据报给它们所需目标。

为了在非广播ATM LAN中仿真相同的操作，在所有站间和连到ATM LAN的路由器端口和称为**无连接服务器**（CLS）的中央结点间建立永久VCC。CLS提供地址解析和数据报中继服务功能。因为使用这种方法，所有数据报直接由CLS中继，所以可以说无连接服务直接提供。使用CLS的协议体系结构的示意图在图10-16中给出。

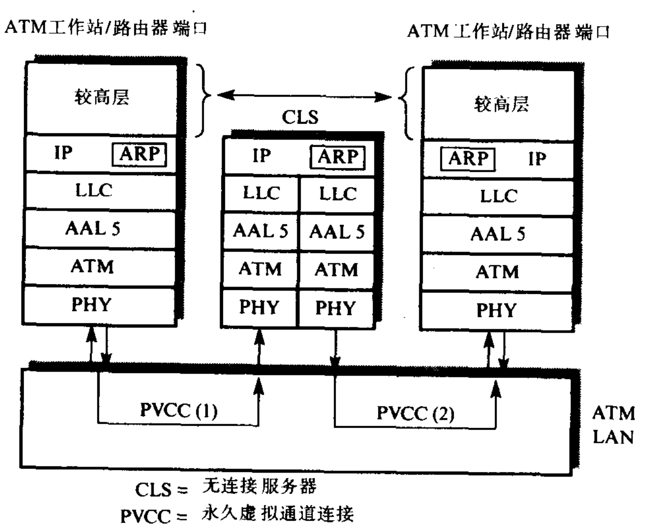


图10-16 在ATM LAN上支持传统IP的协议体系结构

每个站/路由器端口中的ARP先使用相应的永久VCC在CLS中的ARP注册自己的地址对（IP和ATM地址）。以类似的方式，CLS中的ARP通过VCC能获得连到其上的站或路由器端口的地址对。这样，CLS中的ARP建立路由选择表，它含有连到CLS上的所有站和路由器端口的地址对。无论何时站IP有数据报要发送，它简单地在连接该站到CLS的VCC上发送数据报给CLS中的IP层。接到该数据报，CLS中的IP先从数据报头部读取目标IP地址并用它从路由选择表中获得目标的对应ATM地址。然后IP在相应的VCC上转发该数据报给目标。 597

使用CLS的多播以类似于前面描述的使用LES的多播方式进行，除了在使用CLS的多播中

不需要单独的广播服务器。相反，收到含有多播地址的数据报，CLS中的IP层使用所有其他站的相应VCC发送数据报的副本给它们。然后每个接收IP从数据报头部的多播地址决定它是否是多播组的成员。如果是，数据报被传递给更上面的协议层。如果不是，该数据报被丢弃。

当使用CLS时，有两种中继模式。第一种，完整的数据报在由IP层处理前由AAL重装。显然，它引入了时延并需要可观的用于实现它的缓冲区存储空间。为了减少这些开销，可以使用称为**流水线**的第二种工作方式。实际上，用于路由选择的目标地址由第一个ATM段/信元携带，因此在这种方式中，第一个（报文开始）段一但被AAL协议接收到就被直接传递给IP层处理。然后确定新的VC并且它和组成数据报的剩余段被无重装地直接中继。使用这种方式的AAL是AAL 3/4，因为在重装处理期间，需要每个信元头部的MID字段用于标识。

3. *广域连网*

在前一节讨论的两种无连接体系结构用于单ATM LAN中。然而对于经过中间WAN的这类LAN间的通信，已定义了不同的协议体系结构。因为它涉及（公共）WAN，协议体系结构由ITU-T定义。

将在10.5节看到，第一代ATM WAN由基于ATM的**城域网**（MAN）的互连集组成。10-17(a)给出了通过这种网络互连两个ATM LAN的典型安排以及相关的协议体系结构。在这个例子中，假定无连接服务在IP层提供，虽然它同样可以由MAC层提供。

可以看到，ATM网关位于每个客户站点，它的一个端口连到站点ATM LAN而另一个端口连到ATM MAN。通常，在特定区域有许多连到MAN的网关（地点），它提供本地采集和分配功能。**MAN交换系统**（MSS）也连到每个MAN并且MSS集互连形成跨国家交换业务。在网关内提供给用户的服务是交换多兆位数据业务（SMDS）（类似于8.4.7节中描述的用于帧中继网络的服务），网关称为**SMDS边界网关**。

将在10.5.7节讨论与SMDS（ITU-T称其为无连接宽带数据服务（CBDS））相关的各种协
598 议的作用。基本上，收到带有指明不同站点网络地址（网络号）的目标地址的数据报，ATM LAN中的CLS会中继该数据报给SMDS边界网关的ATM LAN端口。数据报或者MAC帧（如果使用LAN仿真）先由SIP 3级协议封装成标准化帧格式，然后通过SMDS网络中继到适当的目标网关。从那里它先被中继到站点CLS，继而到目标站。

在更长时间帧，ATM WAN中的MSS由ATM交换网络互连。整个网络称为宽带ISDN（BISDN）。这种结构的示意图以及与它相关的协议体系结构显示在图10-17(b)中。可以看到，它类似于图10-17(a)，除了在这种情况下存在中间ATM交换网络。**无连接网络接口协议**（CLNIP）提供类似于SIP 3级协议的服务，每个CLNIP PDU格式类似于稍后图10-27中给出的格式。主要差异是CLNIP PDU没有头部和尾部，因为它由ATM网络中的AAL提供。

10.5 DQDB

分布式队列双总线（DQDB）是基于信元的广播网络，它主要开发作为高速LAN互连设施。此外，它能支持有限同步（恒定比特率）通信服务。LAN可以物理地分布在单一地点，更普遍的情况下是分布在多个地点。这种网络称为城域网（MAN）。分布在多个地点网络的连接点是远端网桥或路由器。因为工作站不是直接连到网络，通常MAN使用34/45/140/155Mbps公共载波线路的高比特率共享传输。现在DQDB是国际标准并在IEEE 802.6（ISO 8802.6）中定义。

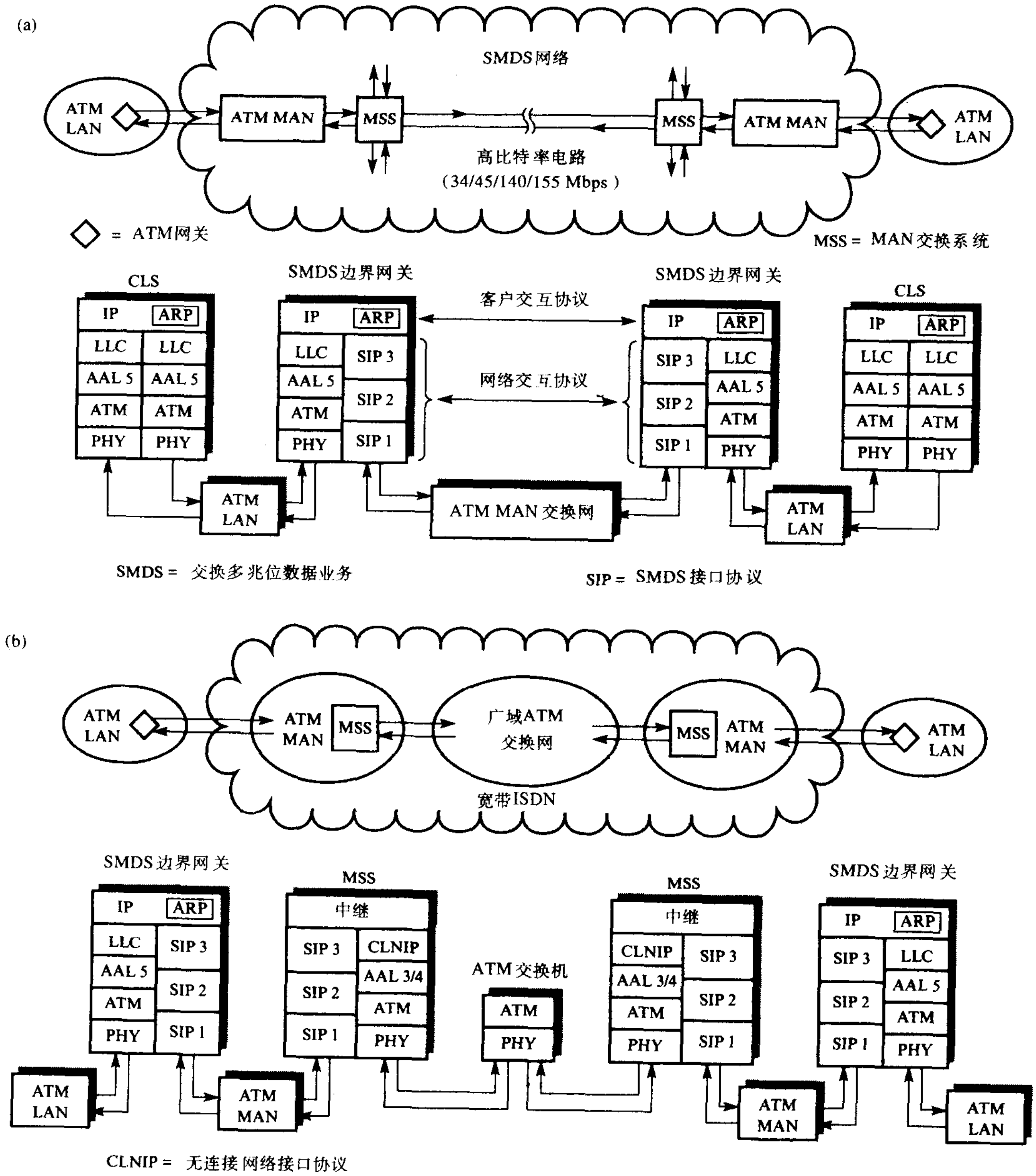


图10-17 在广域ATM网络上的无连接工作

(a) ATM MAN 交换网 (b) 宽带ISDN

在较大网络中DQDB标准涉及单一子网。通常，较大型网络由许多互连的DQDB子网组成。每个子网由**双反向总线**（就是两条工作在相反方向的单向总线）组成，**访问结点**（还称为**用户网络接口单元**）的分布群体连接在上面。总线可以是**开放式总线拓扑**或**环形总线拓扑**的形式，在这种情况下两个端点是独立的，但物理上是联合定位的。稍后会看到，它可以用来提供更好的容错。图10-18给出了三个应用实例：(a) 显示了一个单子网ATM；(b) 显示典型地点分离专用网络；(c) 显示了由多个互连MAN组成的大型广域公共网。

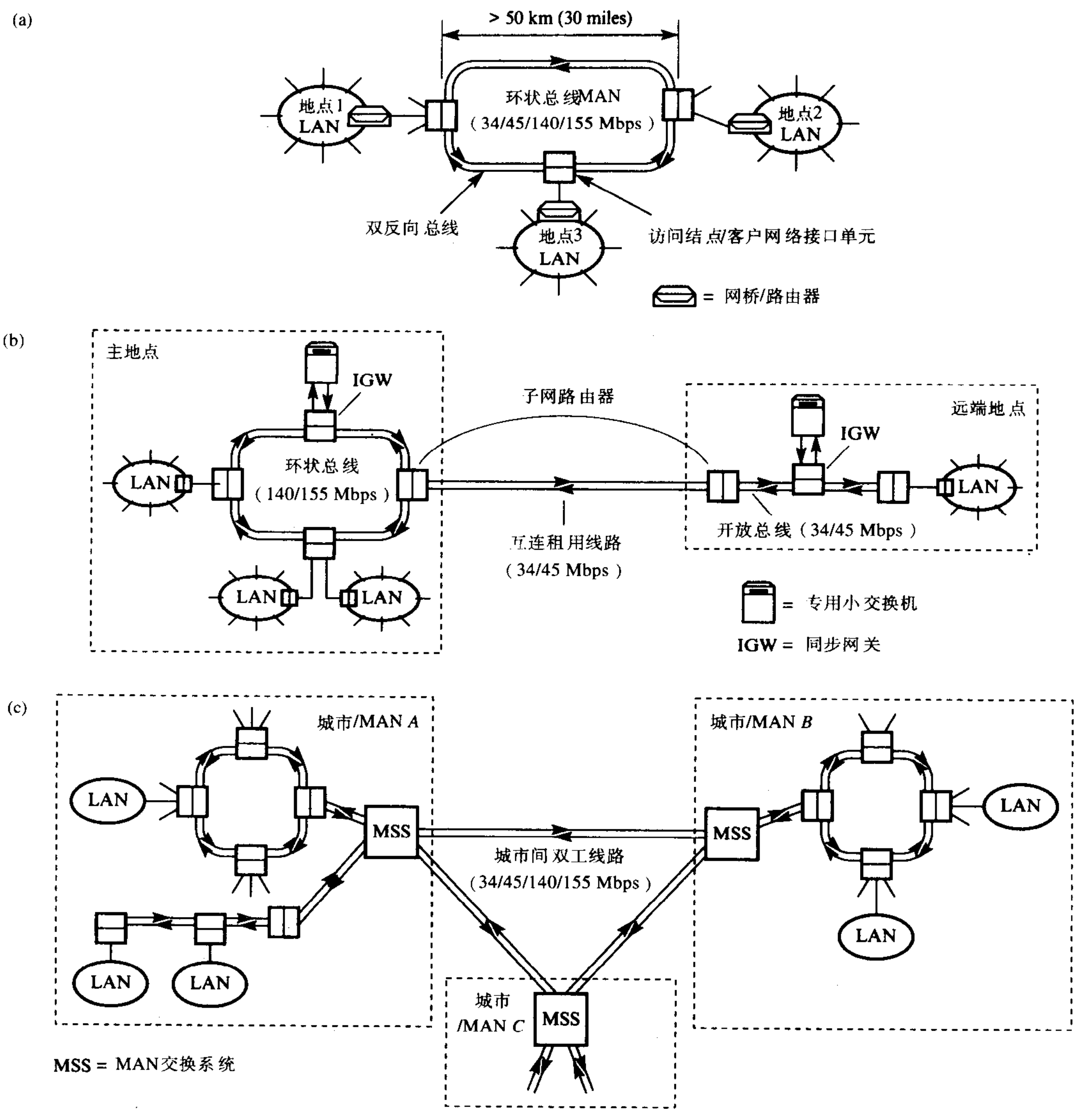


图10-18　DQDB/MAN网络体系结构

(a) 单MAN示意图　(b) 双地点专用网络　(c) 广域多MAN网络

599~600 如图10-18(a)所示，单DQDB MAN子网能覆盖超过50公里（30英里）的区域。在这个例子中，访问结点位于该区域的电话交换中心。34/45/140/155Mbps电路的使用为分布在城市的许多LAN提供**无缝互连**，就是在不同地点的与另一个通信的两个用户不知道中间网络参与的事实。

图10-18(b) 说明了在专用网络应用中的两个DQDB子网的使用。在这个实例中，所有位于两个地点的连网设备是私自拥有和运行，并使用从公共载波租借的高比特率电路连接两个地点。这个实例还说明了DQDB网络如何用于互连需要恒定比特率通道的两个专用电话交换

601 机（PBX）。

图10-18(c) 说明多子网MAN如何互连形成跨多个城市的更大的网络。可以看到，在这种情况下每个城市中的DQDB子网连到MAN交换系统（MSS）。MSS收到每个LAN帧或者把它

直接中继到所需的子网（如果它连到该MSS上），或者通过适当的城市间双工电路把它转发到所需的MSS。这种网络提供交换多兆位数据服务（SMDS），类似于帧中继网络提供的服务。

通常，LAN通过远端网桥或者路由器连到最近的访问结点。所有帧以称为段的定长单元形式通过每个DQDB子网传输。在到每个子网的接口处，帧先被分成段然后在目标端再重装回它们的原始形式。由于LAN帧的格式随LAN类型不同而不同，在传输帧前，要把标准头部和尾部加到帧上面。使用公共网络，头部含有两个新的源和目标地址，它们是对应的用户访问/接口单元的跨网络地址。将在10.5.7节看到，有定义了MSS、子网和用户访问单元的分级地址。所有段使用两条总线（它由于可在反方向传输数据，确保了传输的每个段的副本能被子网上的所有结点接收到）通过每个子网传输。在子网间，对重装后的帧使用每个帧头部的跨网络目标地址执行路由选择。

10.5.1　子网体系结构

开放双总线子网的示意图如图10-19(a)所示。每条总线的前端是**时隙发生器**，它产生连续时隙流，每个时隙能传输一个标准53字节信元。访问结点连接在读和写方向的两条总线上，读操作先于写操作执行。每个访问结点中的物理层读取每个时隙中的内容而不修改它，只有访问结点要发送新数据时才重写现有的内容。所以可以推断，访问结点只从总线复制（读取）数据而不移去它。因此，假定故障不会引起结点的持续写操作，结点内的访问单元故障不会影响到两条总线上传输的时隙内容。

环形总线体系结构如图10-19(b)所示，在这种配置中两个时隙器在一起。同一个时隙器用于两条总线。在这两种情况下，两条总线仍然是独立的，像环状网络中一样不连在一起。还有，除了产生时隙，**总线前端**还负责传递管理信息给访问结点。通常，管理信息由一个单独网络管理站产生并涉及同步带宽的以及子网的操作完整性。 602

使用环形总线能在链路和结点故障发生时，通过复制总线前端（这样多个结点能承担总线前端作用）支持重构。重构环的两个实例如图10-19(c)所示。第一个说明了链路故障发生后的环重第二个说明了访问结点。重操作在远端网络管理站控制下执行。 603

10.5.2　协议体系结构

定义在IEEE 802.6中的协议体系结构的组成部分的示意图如图10-16（a）所示。同其他的IEEE 802标准一样，它定义了MAC子层（在标准中称为DQDB层）以及物理层的操作。

除了提供的用于互连基本LAN类型的正常无连接（最佳尝试）数据业务外，DQDB MAC子层还提供两个额外业务：面向连接的（可靠的）数据业务和同步业务。通常，后者用于互连两个专用（电话）交换机。对于这类应用，在连接交换机到它们的访问结点的双工链路内的多路复用语音采样必须以访问链路相同的速率通过总线传输。

1. MAC子层

如图10-20所示，MAC子层由四个主要功能组成：会聚、总线仲裁、公共和层管理。正如前面指出，所有信息以定长段的形式通过双总线传输。需要会聚功能在所有流入源信息传输前转换成段，并且在它递交前转换回原始形式。例如MAC会聚功能把由网桥或路由器提交的数据帧分成多个段准备传输，并且在接收方，在递交前把它们重装成帧。

要获得两条总线上的时隙访问，有两种可选的控制方式：**队列仲裁**（QA）和**预仲裁**（PA）。当支持同步业务时，为了提供恒定的比特率服务，所需时隙数量由总线前端结点预分配和标记。每个结点中提供该业务的预仲裁功能，负责确定它的保留时隙（来自每个时隙头部的标

识符)，然后在这些时隙内发起同步数据的传输或者接收这些时隙中含有的数据。所有剩余的总线时隙用于传输LAN数据帧。对于这些时隙的访问由队列仲裁功能块中的分布式排队算法控制。

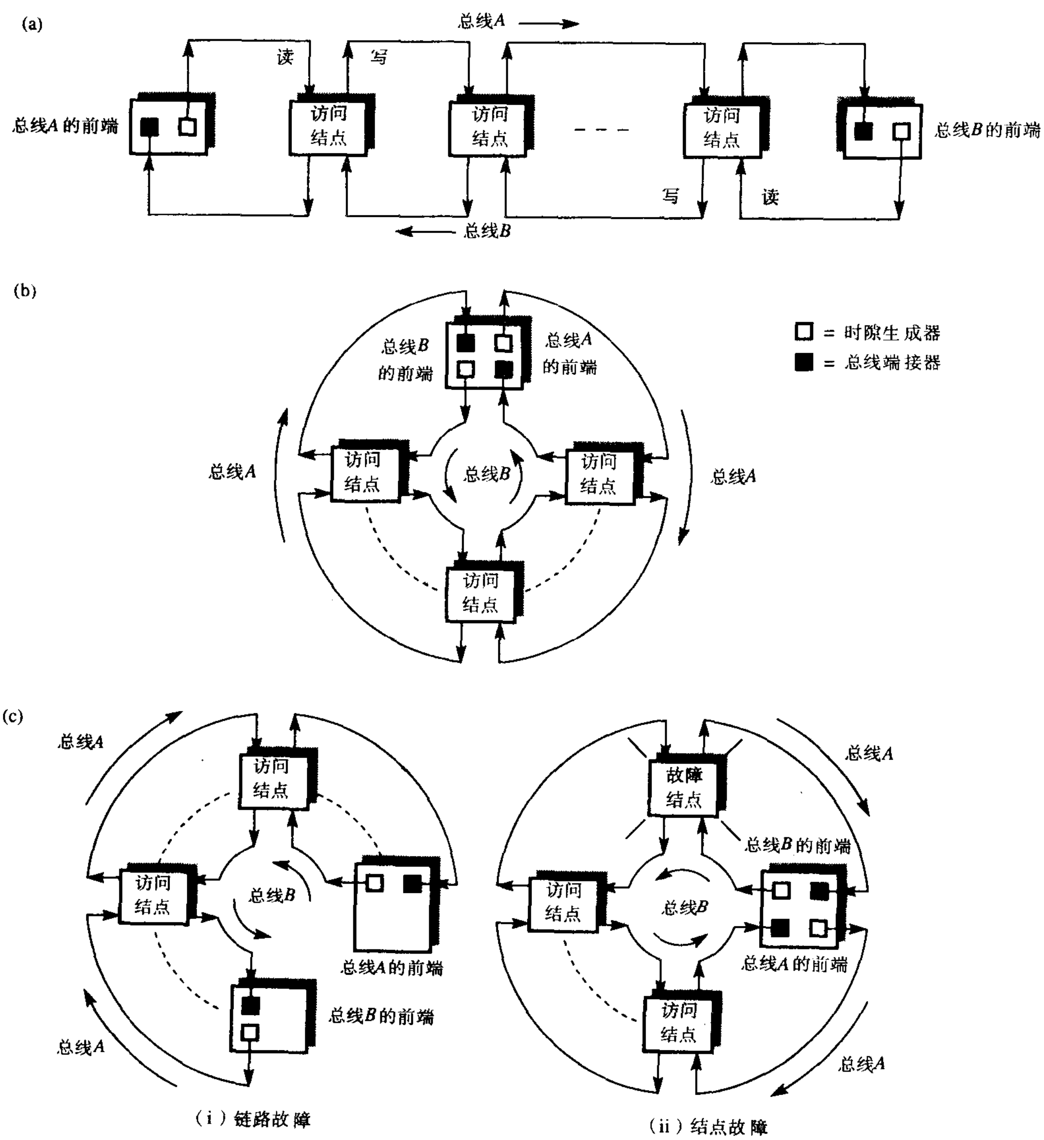


图10-19 DQDB体系结构

(a) 开放总线 (b) 环状总线 (c) 环状总线网络重构的实例

从图10-20看到，公共功能块在物理层和两个总线仲裁功能块间形成接口。物理层接口采取单字节的形式传输(读和写)到/从两条总线。公共功能块的主要作用是在物理层与适当的总线仲裁功能间中继字节。它包括检测每个时隙的开始和结束，以及根据定义的格式检查时隙/信元头部的选定字段。在10.5.6节将详细地讨论。此外，因为多个结点能充当总线前端
604 (为了增强可靠性)，当检测到故障时能执行该功能的结点参与总线/环的重构操作。还有，当

被选为总线前端时，结点产生连续时隙流。两种操作构成了公共功能的部分。

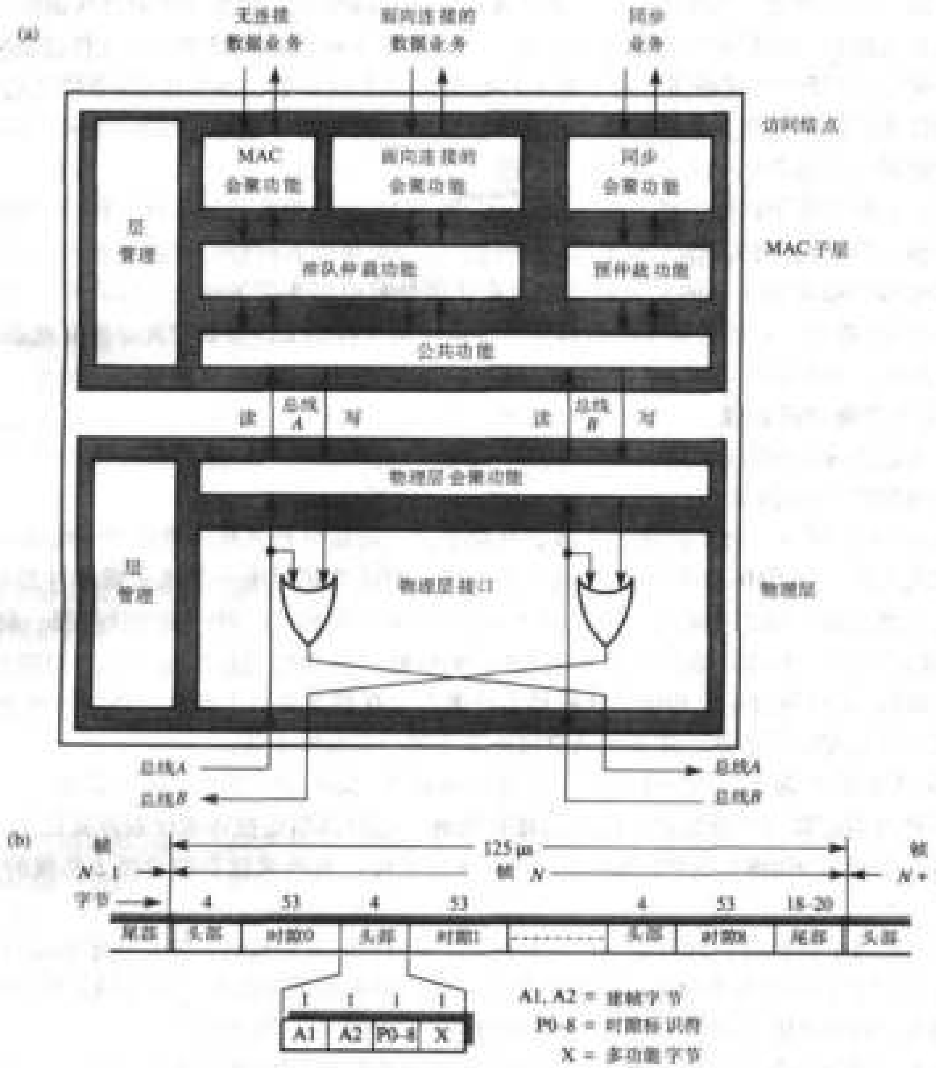


图10-20 DQDB协议体系结构

(a) 层功能 (b) 物理层会聚功能实例

2. 物理层

物理层提供到MAC子层的标准接口。如图10-18所示，物理层可以使用一类不同的传输介质实现。例如，在公共网中它可以是34Mbps、45Mbps、140Mbps或155Mbps。正如在第2章中看到的，这些线路使用不同的帧格式，因此必须提供在选定物理传输介质上建立连续时隙 605
序列的**物理层转换功能**。图10-20(b)显示了如何实现这个功能。

实例涉及使用34.368Mbps E3数字电路。从编号可以看出，它位于（准同步）ITU-T多路复用分级结构中的第3级，并用于传输四个8.448Mbps流。许多位/字节用于帧格式化和其他目的，并且由此可获得用于传输DQDB时隙的较低比特率。当每条总线使用这种线路，总线前端使用有效传输带宽建立如图10-20所示的帧结构。周期为125μs，确保帧能支持各种恒定比特率语音服务和数据传输。可以看到，在每53字节时隙前的是4字节头部。头部的前两个字节

使得每个结点的物理层会聚功能同步每个新帧的开始。第三个字节确定每个帧内的时隙。第四个字节用于许多功能，包括从总线前端到连接结点内的两个层管理实体的管理信息的传输。物理层会聚功能读取并解释头部字段的字节，但只把每个时隙中的53字节信元传递给MAC子层。由于固定的125μs时间间隔，每个帧不严格含有多个57字节。由于在第2章描述的这种线路的准同步工作方式在该字段出现未使用字节。使用较高比特率电路，125μs时间间隔被保持并且每个帧成比例地含有更多（DQDB）时隙。

最后，正如所有协议层一样，提供与物理层和MAC子层相关的层管理实体用于远端管理实体控制和监控这两层的工作。对于MAC子层，它参与接收和操作与时隙相关的信息，例如携带在每个预仲裁时隙头部的标识符表以及在队列仲裁时隙头部的新报文标识符，它们的使用将在10.5.6节描述。它还包括定时器和激活设备控制规程的工作参数设置。所有这些操作信息在每个时隙之前的头部携带。

10.5.3 队列仲裁访问协议

同步数据传输的时隙访问基于分布式排队算法。它称为**队列分组分布交换**（QPSX），该方法的原理如图10-21所示。

如图10-21(a)所示，每个时隙的头部含有两个位，用在分布式排队算法中：**忙位**或**B位**以及**请求位**或**R位**。如图10-21(b)所示，对每条总线上时隙的访问由一个独立**请求计数器**控制。

606 请求某条总线上的时隙需使用另一条总线上时隙内的R位来做出。对于每个计数器，每当R位为1的时隙经过时，对应计数器的内容就加1。类似地，每当反向总线接口的空时隙经过时，计数器就减1。在任何时刻，相应总线的请求计数器含有那条总线上来自该访问结点下游的访问结点的等待时隙的请求数。注意每条总线独立于另一条总线工作。

图10-21(c)说明为了发送一个信元（含有数据段），发送访问结点传输所需总线的请求计

607 数器的当前内容给第二个计数器，称为**递减计数器**。然后该结点把请求计数器复位为0，并对在反向总线上接收到的R位为0的第一个时隙的R位置位。其效果就是把段插入总线的分布式队列中。

当段在总线中排队时，同前面一样，任何R位置位的时隙经过时，使相应的请求计数器加1。但是，B位复位的时隙在反向总线接口处经过时，只使得那条总线上的递减计数器减1。当相应的递减计数器为0，并接收到一个空时隙时相应的段就可送到总线上。

实际上，因为两条总线独立地工作，可能收到的所有时隙串的R位都是置位的。它意味着在反向总线上，直到多个段已被排队并且被传输，才可能在一个时隙内置位R（响应新请求）。考虑到这一点，第三个计数器，称为（本地）**请求队列计数器**，用来保持待处理请求的记录。每当有新请求时，计数器就加1，并且计数器内容一直都大于0，每当接收到R位复位的时隙时，把R位置位并且计数器减1。

实例10-3

画出流程图，说明队列仲裁功能采取的步骤，这些步骤实现由MAC会聚功能在双总线DQDB网络的单总线上产生的一组队列段的传输。

解：

流程图表示控制图10-22给出的总线A上段传输的步骤。解释流程图时注意下列要点：

- 某总线接到满时隙，队列仲裁功能简单地把时隙净荷的内容直接传递给MAC会聚功能。该功能确定该段是否发送给这个结点。
- 每次只能有一个段由队列仲裁功能放入队列传输。由此只有在该功能传输完一个段后它

才返回给MAC会聚功能的输出队列以确定是否另一个段等待发送。

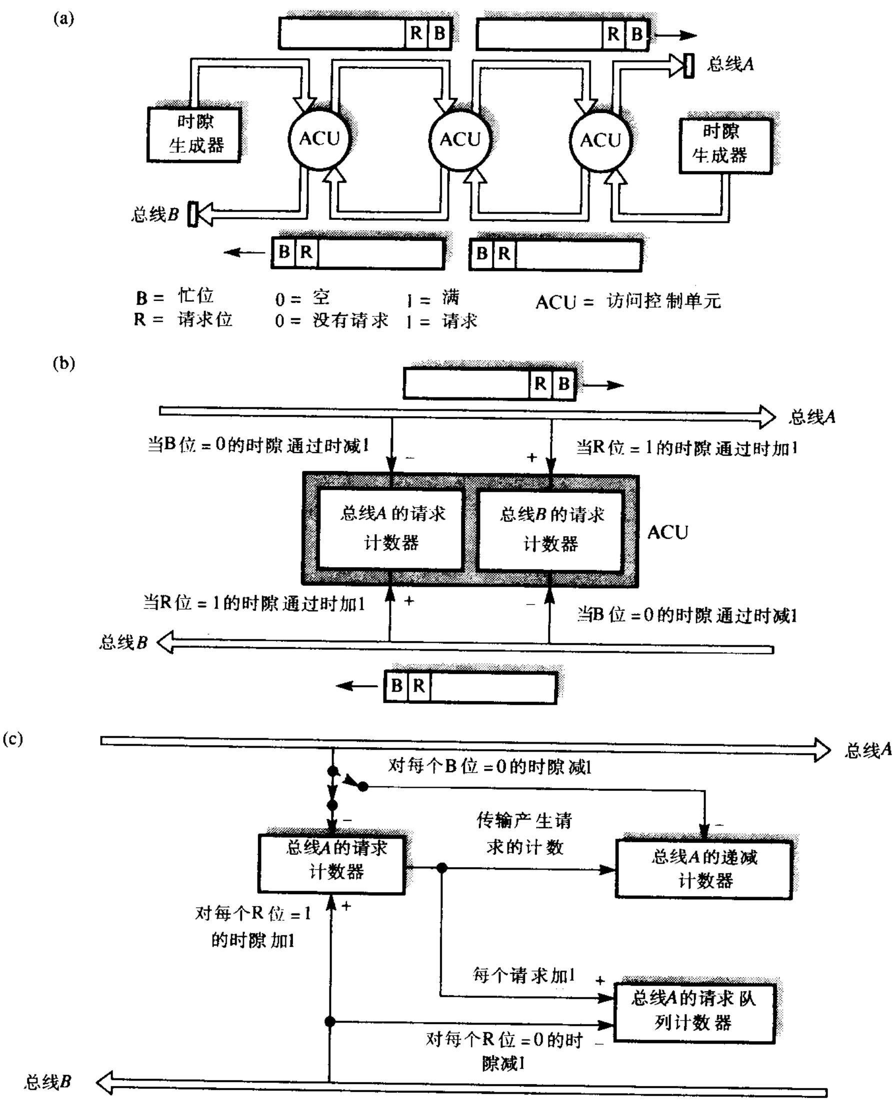


图10-21 DQDB访问控制原理

(a) 请求/忙位 (b) 请求计数器 (c) 排队机制

10.5.4 带宽平衡

随着DQDB草案标准的引入，队列仲裁协议的具体性能分析显示，在严重负载状况下，离每条总线前端最近的结点比离总线中央较近的结点更先获得对两条总线的访问。通过记住每条总线前端的结点就能最好地理解不公平的原因，第一个呼叫使用经过其中一条总线的时隙内的请求位。还有，虽然同一个结点是另一条总线上最后一个发出请求的，但是相关的空时隙第一个经过该结点。在严重负载状况下当时隙的需求开始超过供应时，它会显示图10-23(a)中的图形效果。访问时延变化涉及严重负载的子网并且两条总线是相同的。可以看到，当网络规模增大和/或比特率增加时不公平加剧。 608

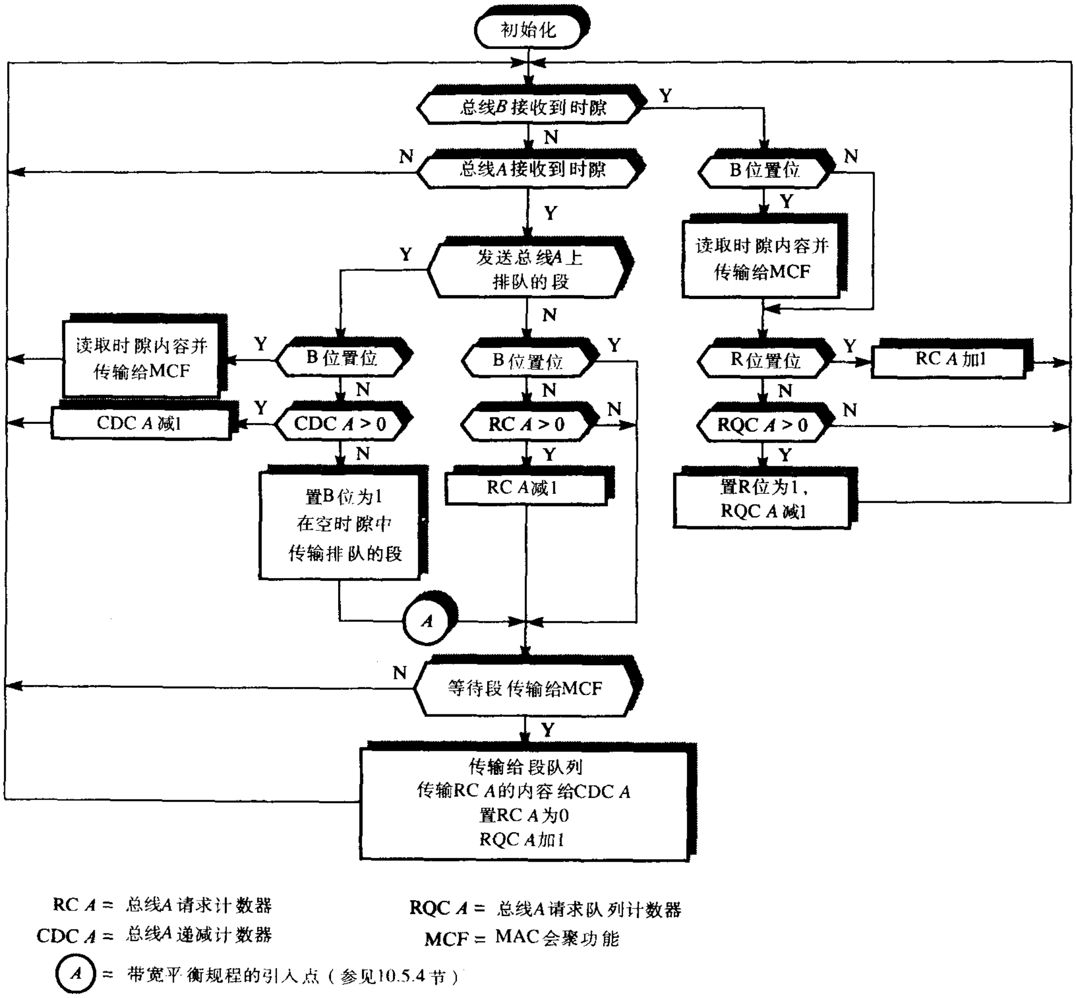


图10-22 在双总线DQDB子网总线A上控制段传输的算法流程图

为了消除这个影响，引入了**带宽平衡机制**的算法，它对基本访问控制算法进行了修改。为了实现这个方案，每条总线引入了称为**带宽平衡计数器**（BWB）的第四个计数器。每当段在总线上传输，那条总线的BWB计数器就加1。然后，每当计数器到达预设限制值，结点通
609 过增加相应的请求计数器允许额外的空时隙在该总线上通过。然后BWB计数器被重置成0。重复该处理流程。预设限制值称为**带宽平衡模数**。这个操作意味着每个结点在传输完等于BWB模数的段块后必须允许一个额外时隙经过相关总线。它能由第一个结点减少有要发送队列段以及递减计数器为0的总线使用。所需的额外处理步骤如图10-23(b)所示，它们在图10-22的流程图中的A点引入。

带宽平衡机制通过减少每条总线的使用来获得需要的效果；BWB模数越小，带宽损失越大。显示在图10-23(c)中的图表说明了减少单一网络类型模数（规模和比特率）的效果。可以看到，当模数减少时不公平也减少，但代价是平均访问时延的增加。实际上，一种折衷方法
610 通常是使用值8。在最坏情况下，它导致1/（8+1）或者11.1%的利用损失。

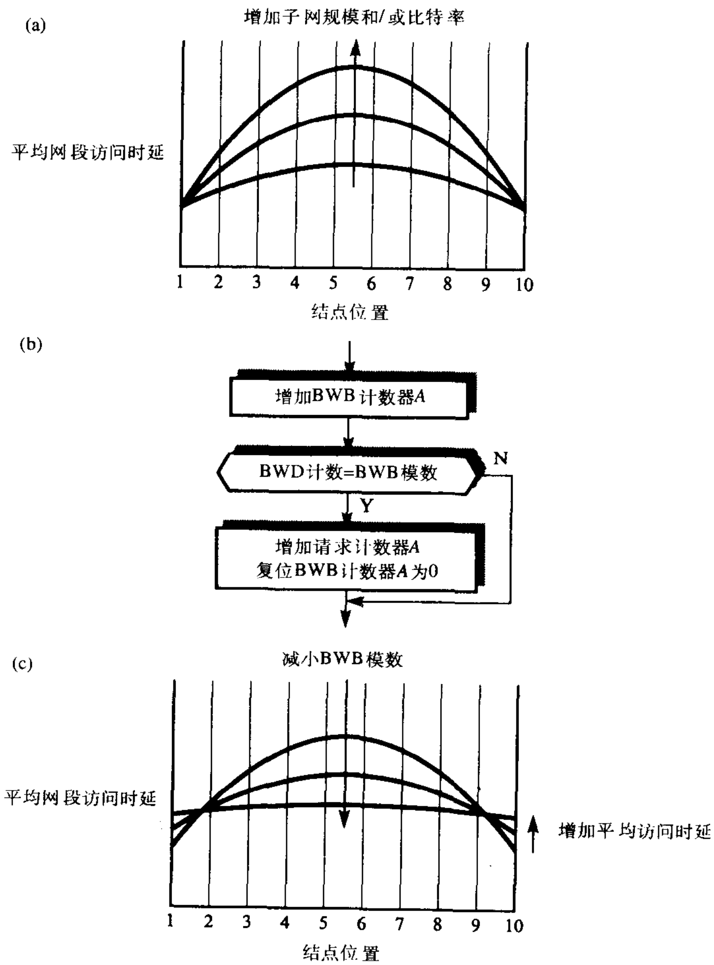


图10-23 带宽平衡

(a) 不公平的影响 (b) 补救措施 (c) 对于平均访问时延的影响

10.5.5 优先级分布队列

虽然不是经常应用，但是刚描述的基本队列仲裁访问控制方案能扩展成支持优先级信元/段的传输。每条总线有三个优先级并且每个优先级有单独的计数器组：请求计数器、递减计数器和请求队列计数器。三个优先级分别是0、1和2，其中2为最高优先级。总是分配给只涉及数据LAN的信元/段的优先级为0。优先级1用于对包含时延或时延变化敏感信息的信元的传输。注意带宽平衡不用于这种模式。

使用优先级级别1的一个应用实例是传输压缩视频信息。虽然以恒定的速率（取决于视频帧刷新速率）产生信息，但是与每个压缩帧相关的信息量是变化的并取决相对于先前帧发生的移动水平。可以推出，如果使用同步服务，那么（预分配）带宽需要是以刷新速率传输一个完整新帧所需的带宽。但是，通过使用队列仲裁访问方式，传输的信息量会随帧变化。通过给这些段分配更高的优先级，优先级控制方法试图确保在含有纯数据LAN通信的时隙前传输它们。

控制对总线A访问的通用方案如图10-24所示。为了清楚起见，只显示请求计数器和递减

计数器。类似的方案用于对总线B的访问。可以看到，在每个时隙/信元头部每个优先级有单独的R位。它们为R0、R1和R2，这里R2是最高优先级。假设图10-24(a)中初始时在总线A上没有来自这个结点的等待传输段。工作过程如下：

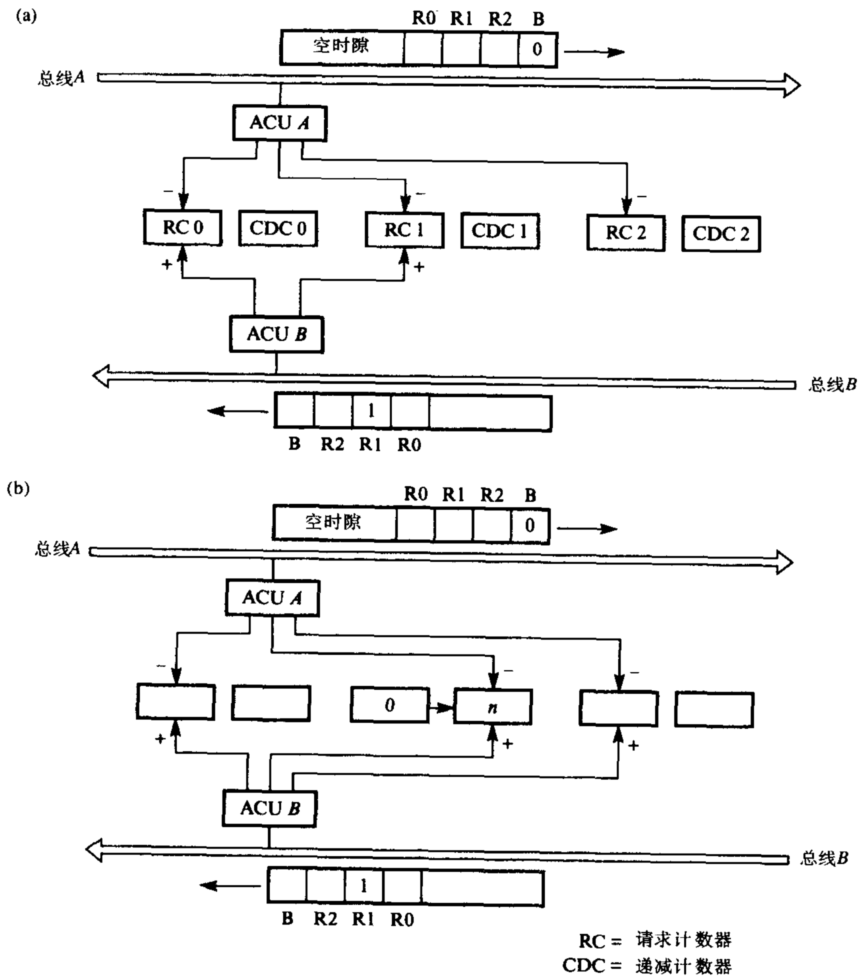


图10-24　优先级访问控制示意图

(a) 没有段等待　(b) 优先级为1的段排队

- 当一个空时隙（B=0）经过总线A的接口处时，该总线的访问控制单元使所有三个请求计数器都减1。
- 当时隙经过总线B接口处时（比如优先级是1），总线B的访问控制单元只递增请求计数器0和1（RC 0和RC 1），而更高优先级计数器（R2）内容不变。这意味着较低优先级请求不会延迟较高优先级段的传输。

现在假设优先级为1的段准备在总线A上传输（见图10-24(a)）。步骤如下：

- 当相应请求位复位成0的时隙经过总线B接口处时，RC 1的当前内容传输给CDC 1（并且RC 1重置为0）。
611
- 当空时隙经过总线A接口处时，请求计数器RC 0和RC 2以及递减计数器CDC 1都减1；
- 如果R位为优先级2的时隙经过，那么请求计数器RC 0和RC 2以及递减计数器CDC 1都加1；

• 当CDC 1变为0并且收到一个空时隙时发送段。

所以，当接到对更高优先级时隙的请求时较低优先级递减计数器加1，有效地延迟了较低优先级段的传输。这意味着有较高优先级的段总是先于较低优先级段发送。

10.5.6 时隙和段格式

正如10.5.2节指出的，每条总线上的53字节时隙/信元由5字节头部和48字节净荷（内容）字段组成。头部的结构如图10-25(a)所示，类似ATM网络的头部。 612

访问控制字段除了忙位和三个请求位以外还含有**时隙类型**位，它说明时隙是否用于队列仲裁或预仲裁（同步）数据。对于含有面向连接或同步数据的时隙来说，20位**虚通道标识符**（VCI）确定了与信元内容有关的逻辑连接。对于LAN（也称为无连接）数据，VCI全部设为1。**净荷类型**指明了携带的数据类型。00用于所有用户数据（队列仲裁和预仲裁）而其他位组合被保留以便将来用于管理信息。**优先级**有默认值00，暂时没有定义其他值。最后，**头部校验序列**是用于差错检测的8位CRC。

为了在子网中传输无连接数据（比如在两个远端网桥间传输MAC帧），提交的帧先由源访问控制单元中的MAC会聚协议分（分段）成许多段。在接收方，目标端的相同协议把接收到的段重装成原始帧。对于面向连接数据和同步数据，信元头部的VCI由目标用来确定哪些信元是要发送给它的。但是对于无连接数据，要在48字节净荷字段开始处使用额外的2字节头部。另外，如图10-25(b)所示，在净荷字段结束处增加一个2字节尾部。

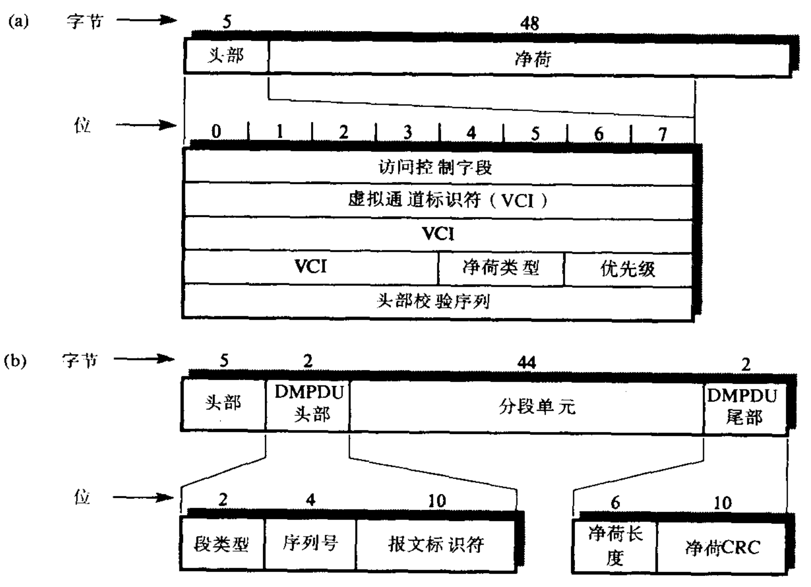


图10-25 时隙和段格式

(a) 时隙头部 (b) 无连接数据段格式

因为含有无连接数据的段是更大（MAC）数据帧的部分，所有它们称为**派生MAC协议数据单元**（DMPDU）。有关帧的段（在标准中称作报文）以下面四种段类型传输： 613

• **单段报文**（SSM） 如果帧/报文能在单段中携带。

• **报文开始**（BOM） 指明它是多个段帧/报文的第一个段。

• **报文继续**（COM） 指明内容在多个段帧/报文的开始报文和结束报文间。

• **报文结束**（EOM） 指明是最后一个段。

序号和**报文标识符**共同用来使得目标能把多段报文重装回它的原始形式。

序号用来检测任何丢失段。在第一个段（BOM）它设成0，每个后继的报文继续段（COM）以及最后一个段（EOM）的序号递增。如果检测到丢失段，帧/报文的剩余段被丢弃。

同一帧有关的所有段由源访问单元分配相同的报文标识符（MID）。使得总线上的剩余访问单元能识别同一帧的有关段。为了确保它们是惟一的，当初始化时每个访问单元分配一个单独标识符块。显然，在单段报文中不需要报文标识符，因此它设成0。

尾部由两个字段组成：**净荷长度**和**净荷CRC**。将在10.5.7节看到，所有提交帧都被填充成4字节的倍数。这意味着一个段可以用多个4字节含有4～44个字节。显然，不是所有提交帧都由多个44个字节组成，由此净荷长度在单段报文和报文结束段中指明实际的字节数。净荷CRC是10位CRC，它用来检测在整个48字节段中的传输差错。

10.5.7 SMDS

在公共网中由MAC会聚功能提供的无连接数据服务称为可交换多兆位数据服务（SMDS）。在这种网中，如图10-20所示的各种子层功能/协议称为SMDS接口协议（SIP）：MAC会聚协议称为SIP 3级协议；队列仲裁协议称为SIP 2级协议；物理会聚协议称为SIP 1级协议。一种典型互连示意图和相关互连协议体系结构如图10-26所示。图10-26(a)是两个LAN通过MAC网桥互连，而图10-26(b)是使用IP路由器互连。

回忆前面几章，不同类型LAN使用不同的头部格式、地址类型以及最大帧长度。为了适应所有类型MAC帧，**SMDS服务数据单元**的长度最大可达9188个字节。还有，由于寻址格式
614 不同，在对提交帧（SDU）分段前，SIP 3级协议先在标准头部和尾部间封装该帧。此外，为了简化目标端的缓冲操作，如果需要会在提交帧尾部增加额外的填充字节，这样它的长度会是4字节的倍数。由此产生的报文单元（称为分组）称为初始MAC PDU（IMPDU）或者SIP 3级PDU，它的格式如图10-27(a)所示。

因此SMDS网络为用户互连方法提供透明的无连接服务。为了达到这个目的，收到帧，访问网关（称为SMDS边界网关）简单地使用队列仲裁访问控制协议在本地DQDB网络中广播该
615 帧。用这种方式，所有提交帧的副本由连接在同一子网的所有其他网关（访问结点）接收到，并且通过它们被所有其他网桥/路由器接收到。后者必须作出决定是否在它的LAN上转发该帧或简单丢弃它。

正如所见，头部由两个字段或者三个字段组成。普通头部含有一个8位序号（称为**开始—结束标记**，它使得SIP 3级协议能检测到丢失帧）以及存储完整IMPDU所需缓冲存储容量的说明。MAC会聚协议（MCP）头部含有许多有关协议的子字段。包括源网关和目标网关的地址，在公共网中它是定义用于ISDN的60位E.164地址。但是，为了适应其他地址类型，两个64位地址字段中最高四位用来确定剩余60位的地址类型（比如16/48位MAC地址）。其他子字段包括存在的填充字节数以及CRC是否存在的指示。允许包括头部扩展部分以考虑未来要增加的子字段。

尾部包括可选的32位CRC，它用于完整IMPDU的差错检测。公共尾部含有与公共头部相同的信息。由此如果存在最大头部扩展和CRC字段，并且信息字段是最大的9188字节，在分

段后会是210段的整数倍。

如图10-18(c)所示，较大SMDS公共网由多个通过一组中间MAN交换系统（MSS）互连的DQDB子网组成。为了MSS执行路由选择/交换功能，每个IMPDU头部的E.164地址是跨网络地址，它确定了MSS以及每个访问网关连接的子网。另外，支持组地址寻址，并且当访问结点接收到在目标地址字段带有预分配组地址的IMPDU时，该IMPDU的副本会发送给组中的所有成员。

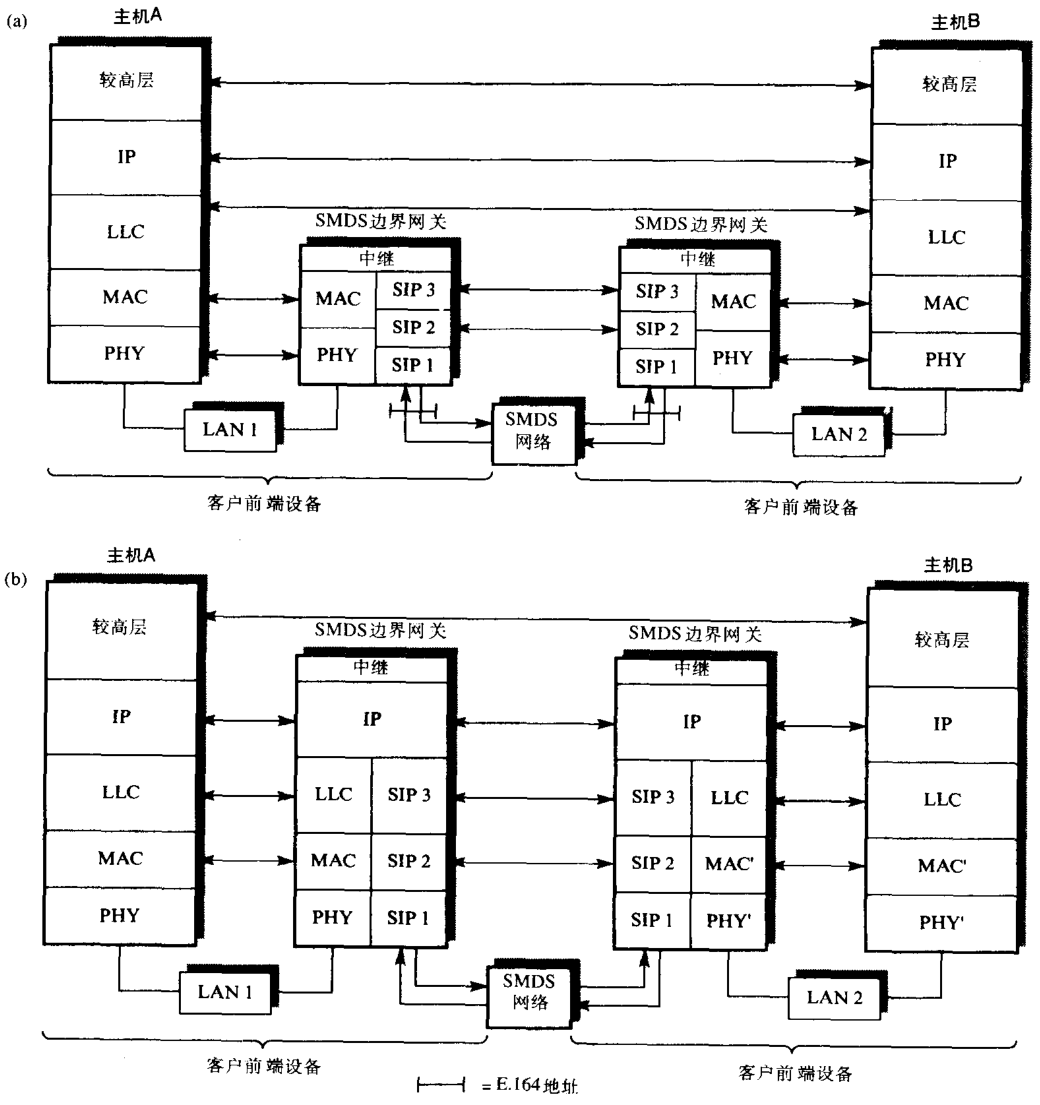


图10-26 SMDS互连协议体系结构

(a) 网桥 (b) 路由器

通过SMDS网络传输提交的MAC帧所采取的步骤以及与每个子层功能相关的开销总结在图10-27(b)中。提交的SMDS服务数据单元先由MAC会聚协议封装成IMPDU。然后把该IMPDU分段成许多DMPDU，每个有相应的头部和尾部。由此产生的48字节段传递给队列仲

裁功能，它增加适当的5个字节头部。最后，传递给公用功能开始通过物理层会聚子层发起传输。实例10-4定量地表示了与每个功能相关的开销。

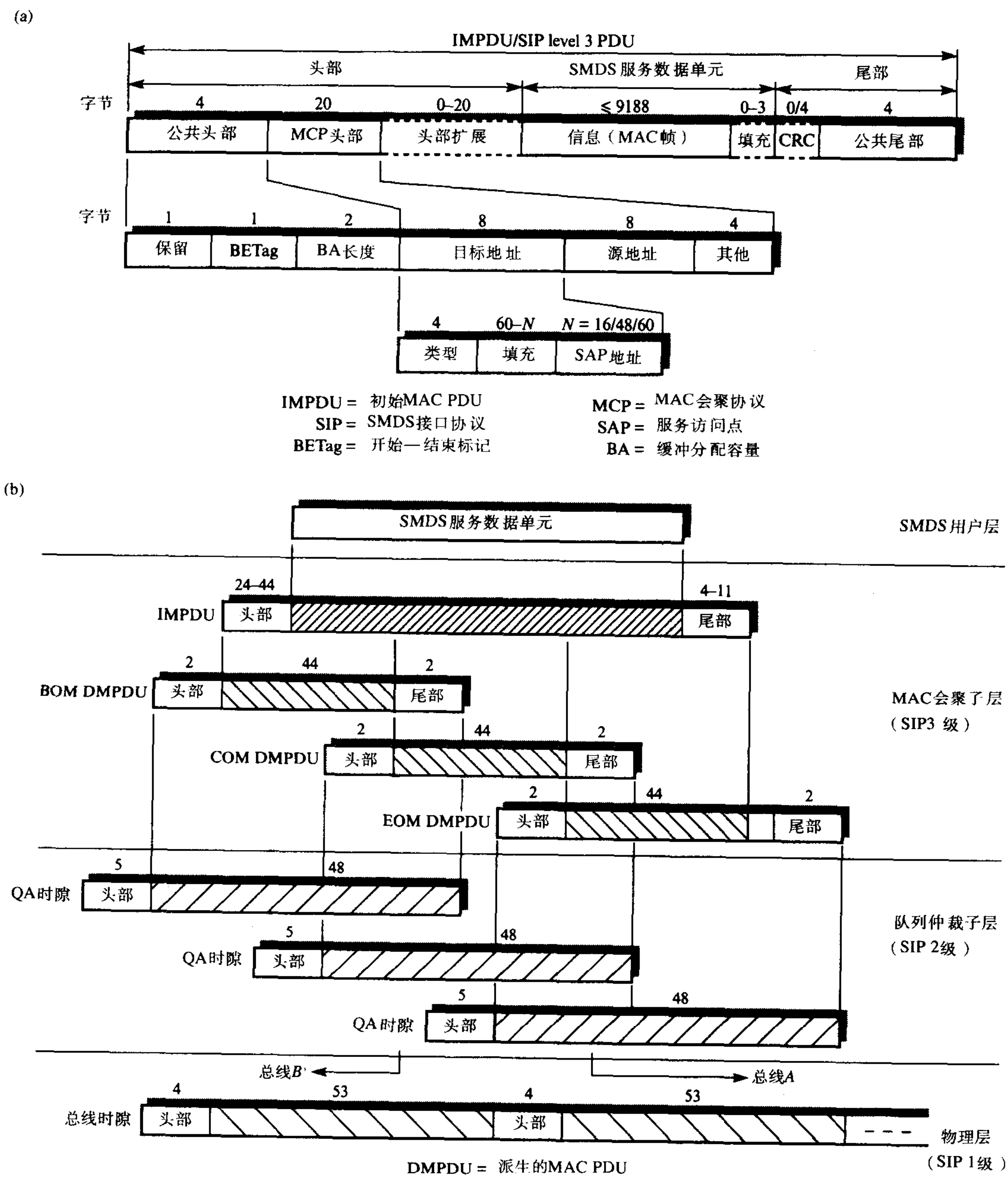


图10-27 帧传输开销

(a) 初始MACPDU格式 (b) 帧分段示意图

例10-4

一个510字节MAC帧要通过DQDB子网传输。清楚地给出假设，求执行传输所需的队列仲裁时隙数以及涉及的开销字节的总数。 616~617

解：

参照图10-27(b)：

- MAC会聚协议
  - 增加2个字节使该帧变成512个字节，它是4字节组的整数倍。
  - 假定不使用头部扩展和CRC，增加24字节头部和4字节尾部产生一个（512+24+4）540字节IMPDU。
  - 总开销 =2+24+4=30个字节
  - 在分段后，IMPDU需要13个DMPDU：12个包含完整的44字节而1个包含12字节。
  - 总开销是52个字节（13个DMPDU，每个的开销是4字节）加上32个字节（用于部分一全部EOM DMPDU）。
- 队列仲裁（QA）子层
  - 再给每个48字节DMPDU增加5个字节以产生13QA时隙。
  - 总开销=5 × 13=65个字节。
- 物理层
  - 再给每个QA时隙增加4个字节。
  - 总开销=4 × 13=52个字节。
- 总计
  - 所需QA时隙=13。
  - 总开销=30+52+32+65+52=231个字节。

10.6 ATMR

异步传输方式环（ATMR）是一种新ISO标准，用在类似于DQDB的应用领域。它基于高速共享介质，但是与双总线相反，它使用**双反向时隙环**。通常，物理介质由高比特率（155/622Mbps）公共载波电路组成。使用53个字节的标准信元长度，信元格式与图10-25(a)所示的相同。ATMR和DQDB的主要差别是获得共享传输介质访问的MAC方式不同。在随后的描述中，只集中介绍该标准的这个方面。

一般方案如图10-28所示。可以看到，两个环是反方向的，并且像FDDI环一样，一个环活动而另一个备用。环互连若干个**访问结点**，提供用于LAN互连的LAN接口以及直接的基于信元服务。信元头部的访问控制字段（ACF）含有三个子字段：监控位、重置位和忙地址。当 618
环第一次初始化时，类似于令牌环的竞争过程选择一个主结点。一旦选中，它首先通知所有其他结点两个环是活动的，其次给环提供时钟（位）定时并且建立时隙结构，再次从环中删除那些接收者未删除的信元。最后一个功能由每个信元头部的**监控位**实现。主结点对每个经过其环接口的满时隙中的监控位置位，并且如果接收到的满时隙该位已设置，那么访问单元置信元内容为0。

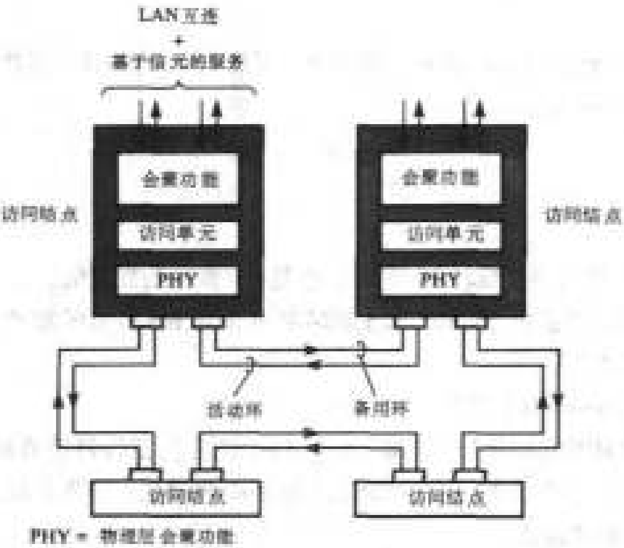


图10-28 ATMR配置示意图

10.6.1 访问控制协议

每个访问结点内的访问单元执行物理层会聚功能和访问控制功能。对活动环的访问由窗口机制和全局重置规程共同控制。窗口机制限制了结点在某个时刻能传输的信元数，而全局重置规程重新初始化所有访问单元中的窗口预定义值。

当访问单元有排队信元等待传输时，在环接口接收到空时隙时发送它们，其目标地址放在信元头部的VCI字段内。在ATMR网中，VCI字段称为**环虚通道标识符**（**RVCI**）。在每个结点环接口接收到的信元都要检验它的头部的RVCI字段，如果该信元地址指向本结点，那么信
619 元内容被复制并传递给适当的会聚子层。然后RVCI字段设为0（指明是个空信元）并将信元中继到环中的下一个结点。如果发现不匹配，那么该信元无改变地中继。这种称为**目标释放**的机制在绕环一周期间，允许有多个时隙。对于在访问结点和/或高比特率传输介质间有较大物理距离的环来说，这尤其重要，因为这种情况下环等待时间较长，就是说该环在任何时刻的循环中含有较大数量的时隙。

环中的传输周期进行，并且在每个周期中分配每个访问单元一个固定**窗口大小**，它指明访问单元在这个周期在环接口发出或接收**重置信元**前能传输的信元个数。**窗口计数器**由每个访问单元维护。它在每次环重置时初始化成窗口大小，然后每次访问单元用空时隙发送它的传输队列的信元时减1。当窗口计数变成0或者在传输队列中没有信元等待发送时信元传输停止。环中所有访问单元都遵循这种机制，因此最终所有单元变成不活动，绕环的信元流停止。

为了重新开始信元传输，终归要有一个访问单元是活动的，就是说窗口计数大于0并且在传输队列中有一个或更多信元等待传输，它把自己的地址重写在环接口经过的所有信元头部的**忙地址**字段。以这种方式，如果某活动访问单元接收到忙地址字段为自身地址的信元时，它判断出其他所有访问结点现在都不活动。在完成剩余队列信元发送后（假定窗口计数仍然大于0），它对环接口经过的下一个信元头部的重置位进行置位（产生重置信元），并重新初始化等待计数器为窗口大小值。重置信元沿着环以正常方式传递并使得其他所有访问单元重新初始化它们的窗口计数器。在该信元沿着环循环一周后由设置重置位的同一访问单元清除该

位。一旦重新初始化完成，任何有排队等待传输的信元的访问单元变成活动状态并重新开始发送信元。

实例10-5

ATMR环由4个结点组成并以等于4的窗口大小工作。假定访问单元1在它的传输队列中有6个信元要发送给结点3，而访问单元2有6个信元要发送给结点4。该环有8个信元的等待时间。假定访问单元2在访问单元1开始发送后2个信元时间开始传输信元，画出说明经过环传输信元组的序列图。

图10-29以及它的解释性注释说明了传输过程。注意访问单元2在成为最后一个活动结点时如何发送它的第二个两信元组。 620

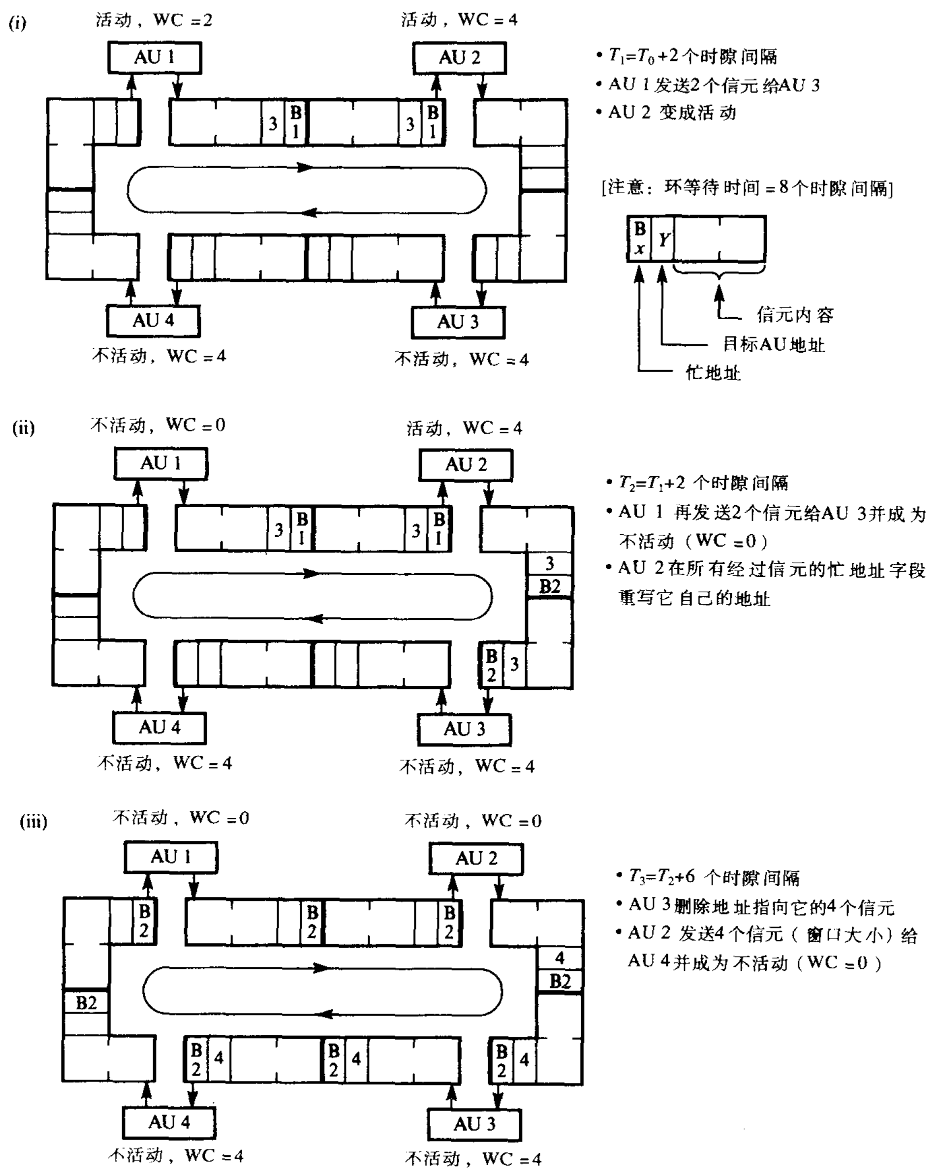
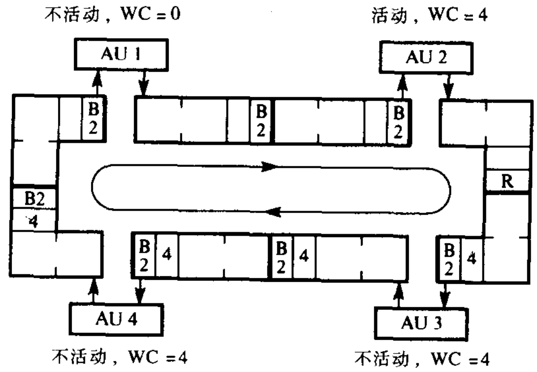


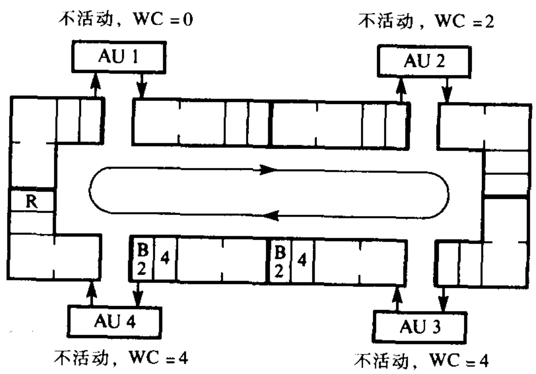
图10-29 ATMR访问控制实例 621

(iv)



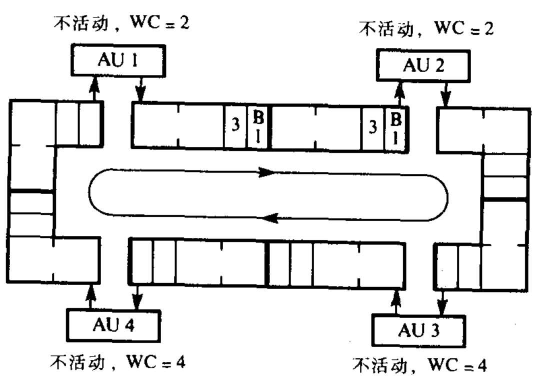
• $T_4=T_3+1$ 个时隙间隔
• AU 2 接到第一个有它的忙地址的返回信元：
  – 发送重置信元
  – 重置WC＝4
  – 变回活动状态

(v)



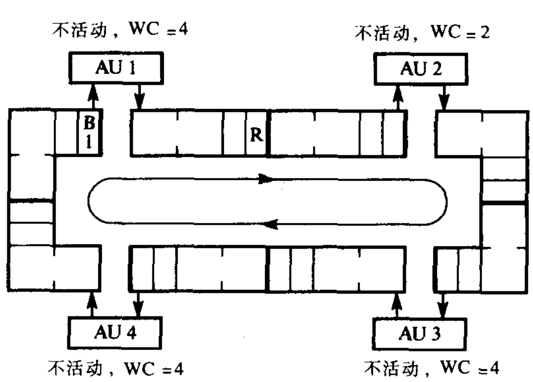
• $T_5=T_4+4$ 个时隙间隔
• AU 2 继续发送它的最后2个排队信元然后成为不活动

(vi)



• $T_6=T_5+3$ 个时隙间隔
• 接到重置信元，AU 1变成活动，发送它的最后2个信元然后再次成为不活动
• AU 2 从环中删除重置信元

(vii)



• $T_7=T_6+8$ 个时隙间隔
• AU 3 删除2个地址指向它的信元
• AU 1接到第一个有它的忙地址的返回信元，发送重置信元

图10-29　（续）

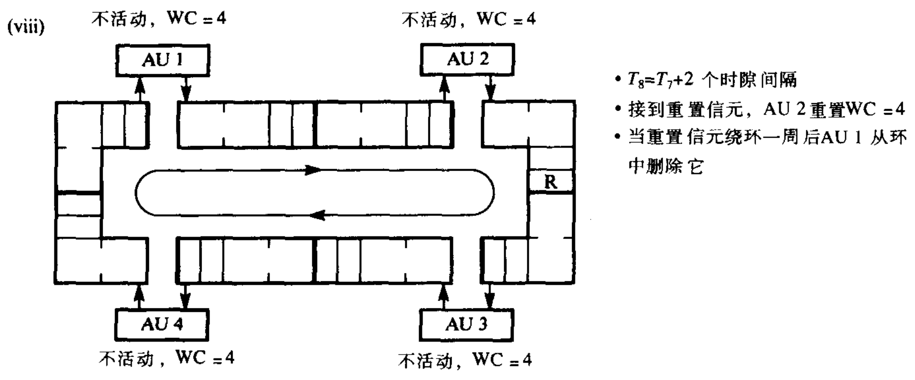


图10-29 （续）

10.6.2 多优先级协议

从10.6.1节可以推出，每个结点采用定义好的窗口大小意味着具有规定个数结点，访问单元在能够发送等待信元前最坏情况时延是受限制的。**重置周期**是使环能用来传输时间敏感信息的特性。最大重置周期取决于分配给所有结点的窗口大小总数以及执行环重置操作的开销。重置时间主要由环等待时间决定。虽然最坏情况时延是受限制的，但是实际时间会因环的当前负载而变化（直到最大值）。为了传输对时延变化敏感的同步通信，刚描述的基本窗口和重置机制可推广为两个优先级级别。在这种操作方式下，访问单元以称为**传输级**的两个状态工作。两个状态间可能的变迁以及引起每次变迁的事件如图10-30所示。 622~623

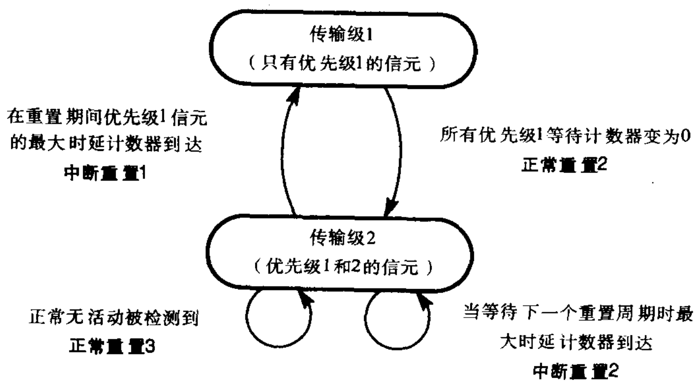


图10-30 多优先级状态变迁图

可以看到，当发送四种不同类型重置信元的一种时，就发生两个状态间的变迁。两个优先级分别称为1级和2级，其中1级是最高的优先级。每个优先级别的信元由每个访问单元保留在各自的队列中，并且每个级别使用各自的窗口大小和窗口计数器。正常（基本）状态是传输级2，在这个状态下所有访问单元传送属于高优先级和低优先级的全部信元。当所有访问单元都不活动时，访问单元以正常方式检测到它，并发送一个称为**正规重置3**的重置信元。

在基本状态下，1级优先级信元可能由于下列两个原因经历很长的时延：

1) 在当前重置周期中访问单元不能发送它的所有排队1级优先级信元，例如由于其他访问单元正发送2级优先级信元。

2) 特定重置周期中访问单元在它的1级优先级窗口计数器到达后还有排队的1级优先级信元。

为了克服这些可能性，有两种额外的重置——称为中断重置1和中断重置2。两种重置都由运行在每个访问单元的定时器触发，在与排队的级别1信元相关的最大信元延时到达前，访问单元设为到期。对于情况（1），如果在重置周期内它的定时器到达，那么像在图10-30中看到的，它发送**中断重置1**，使得所有访问单元转变为传输级1。在这个状态下，只有1级优先级信元才能被发送，由此只有窗口计数器1递减。以正常方式检测到访问单元不活动，传输它的排队1级优先级信元，最后发送一个称为**正规重置2**的信元。它使得所有访问单元转变为传输级2并初始化两个优先级的等待计数器。

对于情况（2），如果某个访问单元仍然有排队等待下一个重置周期的1级优先级信元并且定时器到达，那么访问单元就发送一个**中断重置2**。它使得所有访问单元重新初始化它们的1级优先级窗口计数器，但这次它们仍然保持相同的状态。它使得发送重置的访问单元能够发送它的延迟信元，并且那些已经在当前周期发送了2级优先级信元的访问单元在下一个周期中
624 不能发送任何信元，由此缩短了1级优先级信元的访问时延。

10.7 CRMA-II

循环预留多访问（CRMA）协议是开发用于与DQDB和ATMR类似的应用环境的MAC协议。它旨在支持基于帧的通信和基于信元的通信。已经定义访问协议用在单一环拓扑中，但也可用于双环和双总线拓扑中。该协议提供了对多个并发用户的公平访问并且环工作在千兆比特传输率下仍可维持公平。CRMA-II是该协议早期版本的发展。一般系统配置如图10-31(a)所示。

如在DQDB和ATMR中一样，对环的访问通过访问结点分布集实现。访问协议基于循环预留控制方案，它确保对所有结点公平以及最坏情况访问时延。方案由**调度结点**管理。实际上，所有结点都存在调度功能，但是在任何时刻只有一个结点是有效的。使用分布式竞争算法选择活动调度结点，并且在双环系统中每个环的活动调度结点可以是不同的结点。

在DQDB和ATMR中，某个结点排队等待传输的信元被独立地发送，就是说每个信元独立地竞争对总线或环上时隙的访问。但是，在CRMA中排队信元块（比如涉及某个LAN帧）一旦得到空时隙就能在连续时隙内发送。为了达到这个目标，在结点内的内部缓冲用来延迟相应的流入时隙块而发送那些排队的信元。这种技术称为**缓冲区插入**（因为时隙缓冲区集被有效地插入到环中），并且用来保留延迟时隙的FIFO队列称为插入缓冲区。

每个时隙的格式如图10-31(b)所示。所有时隙以一个称为**起始定界符**的规定的同步模式开始，后跟2字节**时隙控制**字段，它由许多子字段组成，随后描述每个子字段的功能。与源和目标结点地址相关的是时隙控制字段内的标记位，它指明时隙内容是由目标结点检测或是在时隙绕环一周后由源结点检测。回忆10.6节，对于大型环来说，时隙的目标释放能提高超越环传输率的潜在网络吞吐量。

净荷字段能达到60字节长并且终止于**结束定界符**。跟在它后面的是时隙控制字段的副本，它用于差错检测。但是，当在连续时隙内传输数据帧时，每个时隙含有一个起始定界符但只有最后一个时隙含有结束定界符。还有，只有第一个时隙含有目标和源结点地址的说明。时隙控制字段中有一个子字段说明该时隙是第一个时隙、中间的时隙、最后一个时隙或者涉及多时隙帧的惟一时隙。所以，这些时隙内的数据量会从第一个时隙56字节变化到后继时隙的
625 60字节。对于基于信元的服务来说，只使用53字节的净荷字段。

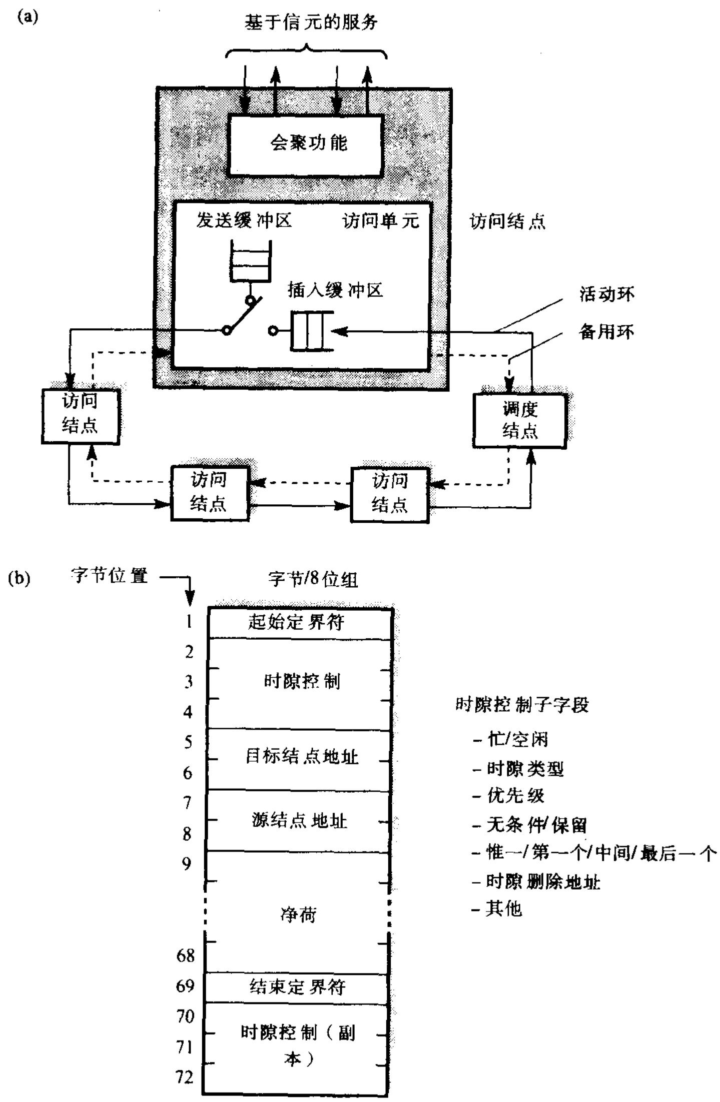


图10-31 CRMA-II

(a) 系统配置 (b) 时隙格式

10.7.1 帧传输

时隙控制字段含有一个**忙/空闲**位，它说明时隙是否忙（满）或空闲（空）。结点只能在空时隙内传输排队段/信元。此外，第二位无条件/保留位指明该时隙是否能由任一结点（无条件）使用或者只能由已接到对早先向调度结点发出的保留请求的肯定证实的结点使用。接到保留请求，调度结点在它的环接口转发时标记相应个数空时隙为保留状态。一旦被使用以及随后变成空闲，相同时隙会回到无条件状态。有关时隙保留规程的命令是独立实体，命令类型嵌入在起始定界符与结束定界符间。 626

无论何时某个结点在它的发送缓冲区中有排队的帧段组，它会在环接口等待空闲时隙的到来。然后在这个时隙中发送第一个段，并且立即在连续时隙中发送剩余段。当发送这些段时，该结点会把从它的前结点接收到的忙时隙放入插入缓冲区的队列中。类似地，当在环接口接收到一个忙时隙，该结点先检查这个时隙头部的目标地址，如果匹配，时隙内容会被复

制。然后这个时隙会被设成空闲或者如果时隙排队等待发送，接收到的时隙的内容会被插入缓冲区或者发送缓冲区前端的时隙的内容取代。

10.7.2　访问控制机制

有两种访问时隙的机制：使用无条件时隙的立即访问机制以及使用预先保留时隙的预留访问机制。**立即访问机制**在轻度通信状况下使用并由时隙控制字段中的忙/空闲位单独控制。预留访问机制在繁忙通信状况下使用。一个简单的实例如图10-32所示。

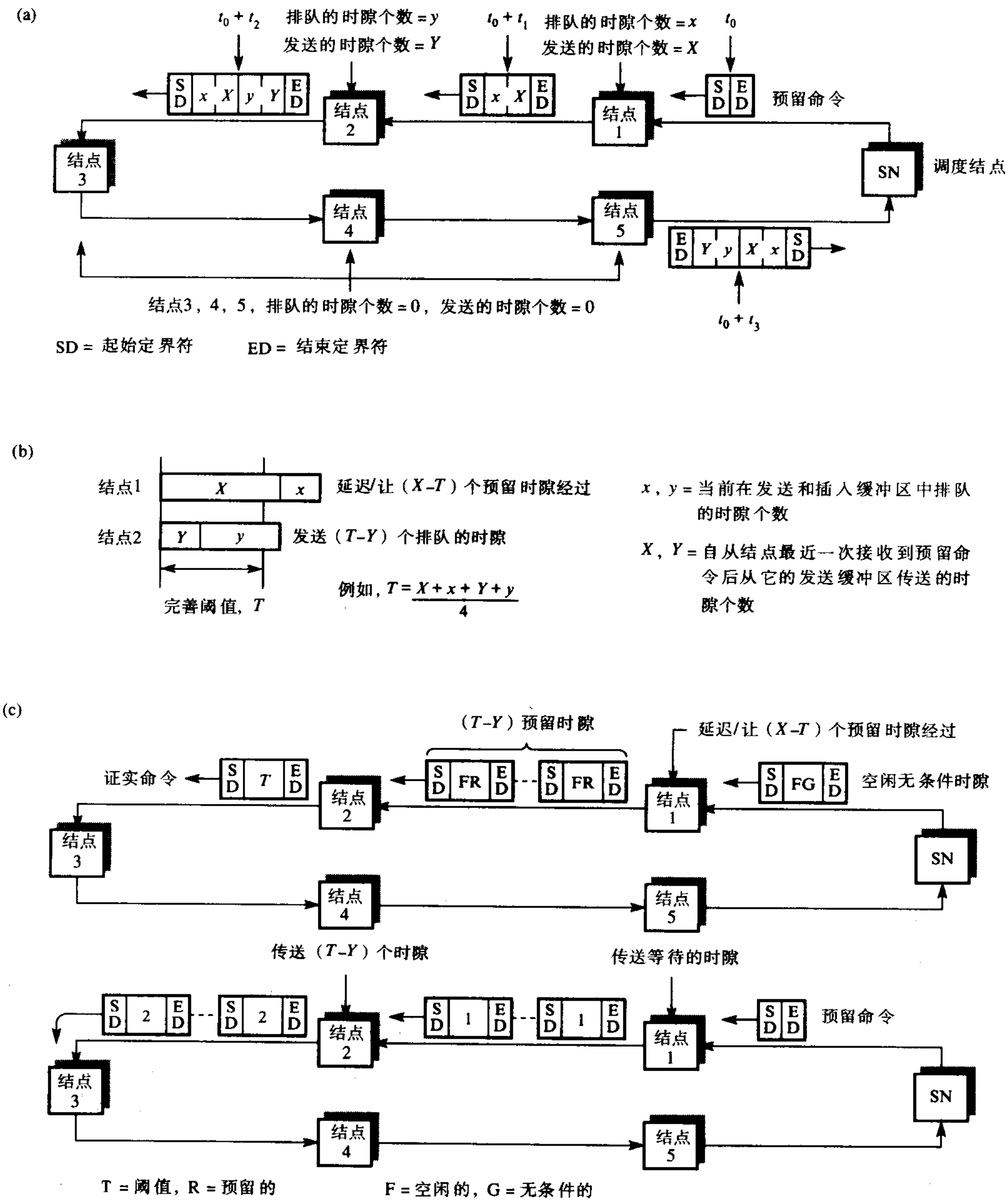


图10-32　CRMA-II预留机制

(a) 预留周期　(b) 完善阈值和结点传输计算　(c) 证实周期

预留机制在当前活动调度结点的控制下周期工作。通过发送**预留命令**（它只由开始和结束定界符组成）开始每个周期。接到这个命令，每个结点把它无改变地中继或者在中继操作中在起始和结束定界符对中插入两个参数。每个结点重复这个规程，预留命令沿着环循环一周返回调度结点。两个参数是结点的传输和插入缓冲区中排队的时隙总数（结点1为x，结点2为y），以及自从它最近一次接收到预留命令以来从它的发送缓冲区发送的时隙数（结点1为X，结点2为Y）。如果结点在任一缓冲区没有排队时隙并且在最近周期没有发送时隙，那么转发预留命令时不插入参数。

当预留命令回到调度结点时，调度结点使用命令中的参数对来确定要产生的（空闲）预留时隙的数量，通过累加来自每个结点的两个参数计算出称为**完善阈值**的数值T。算法试图确保所有结点对环的访问是公平的，而同时使结点必须延迟使用空闲时隙的机会最小化。最简单的算法是把所有参数加在一起（$X+Y+x+y$），然后得出平均值。该平均值就是阈值。调度结点通知其他所有结点这个值，通过在沿环循环的**证实命令**中插入该值。发送完证实命令，调度结点计算分配给所有结点的证实总数，并标记这个数量的无条件时隙为预留状态。 627～628

接到证实命令，每个结点通过计算阈值和最近一个周期中它传输的时隙数的差值来确定它能使用的预留时隙数，就是说结点1为（$T-X$）而结点2为（$T-Y$）。如果结果为正，那么结点就能在这个周期中传输该数量（例如$T-Y$）的预留时隙。如果为负，那么该结点已经超过了对环带宽的公平享用，由此在试图发送任何等待时隙前必须延迟该数量（例如$X-T$）的无条件时隙的使用。每个周期重复同样的规程。

当调度结点执行每个预留周期时，该结点继续使用任何经过它的空闲无条件时隙传输时隙。预留规程确保那些能在本周期内传送更多等待帧/信元的结点只允许在下一个周期中访问更少的时隙。还有，为了使预留访问时延最小化，预留请求命令**旁路**任何在它们经过每个结点时可以在插入缓冲区中排队的时隙。实例10-6说明了该访问控制方式的选定方面。

实例10-6

使用高比特率CRMA-II环在一组访问结点间传输LAN帧。假定所有帧都是由四个段组成的，在示意图中显示在下列环状况和访问控制方式下插入缓冲区的使用以及来自环中某个结点的段的传输顺序。

（a）轻度通信状况和使用空闲无条件时隙的立即访问方式。

（b）繁忙通信状况和使用空闲预留时隙的预留访问方式。

（c）繁忙通信状况和使用空闲无条件时隙的延迟访问方式。

（a）使用立即访问方式时的时隙传输序列如图10-33(a)所示。当解释该图时应注意下列几点：

- 在要传输的帧段到达前，插入缓冲区是空的并且在环接口帧转发空闲无条件时隙。
- 该结点已发送2个时隙后，在一段无活动时期后接收到有关不同帧的时隙。
- 一旦该结点已开始发送它的排队段，无论接收到什么都会全部发送。
- 只有当接收到空闲无条件时隙时插入缓冲区才为空。 629

（b）使用预留访问方式时的时隙传输序列如图10-33(b)所示。当解释该图时应注意下列几点：

- 假定插入缓冲区含有来自较早传输序列的两个排队时隙。
- 发送缓冲区中的四个段被经过的时隙/帧（在最先两个帧间插入了一个预留命令）串阻止。

• 预留命令旁路插入缓冲区中排队的2个时隙。
• 来自该结点的预留命令内容指明在本周期中6个时隙排队以及0个时隙被传输。
• 该结点在能传输任何段前必须等待直到它接收到证实命令。
• 返回的阈值是3，它使得该结点能清空插入缓冲区并且发送等待帧。

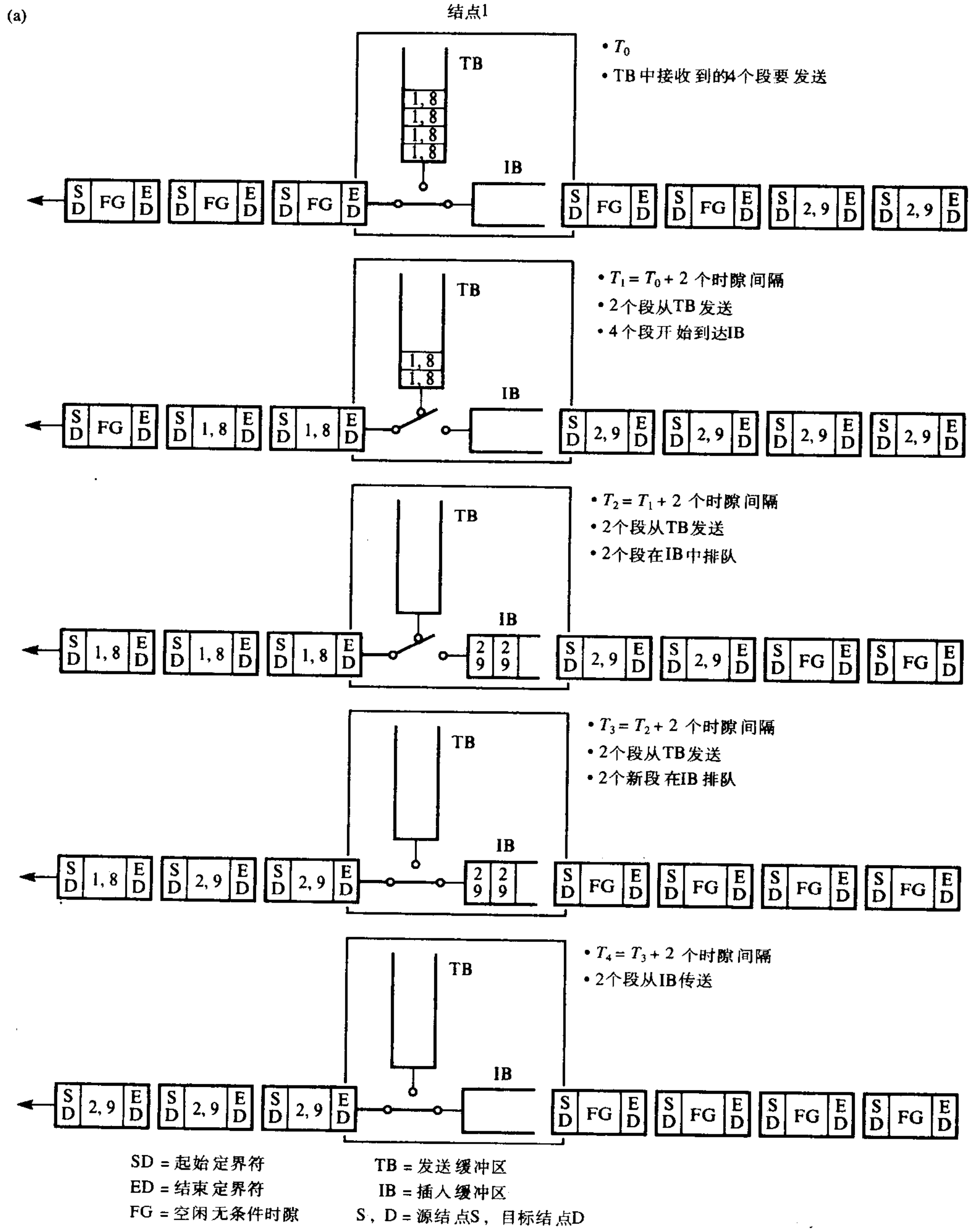
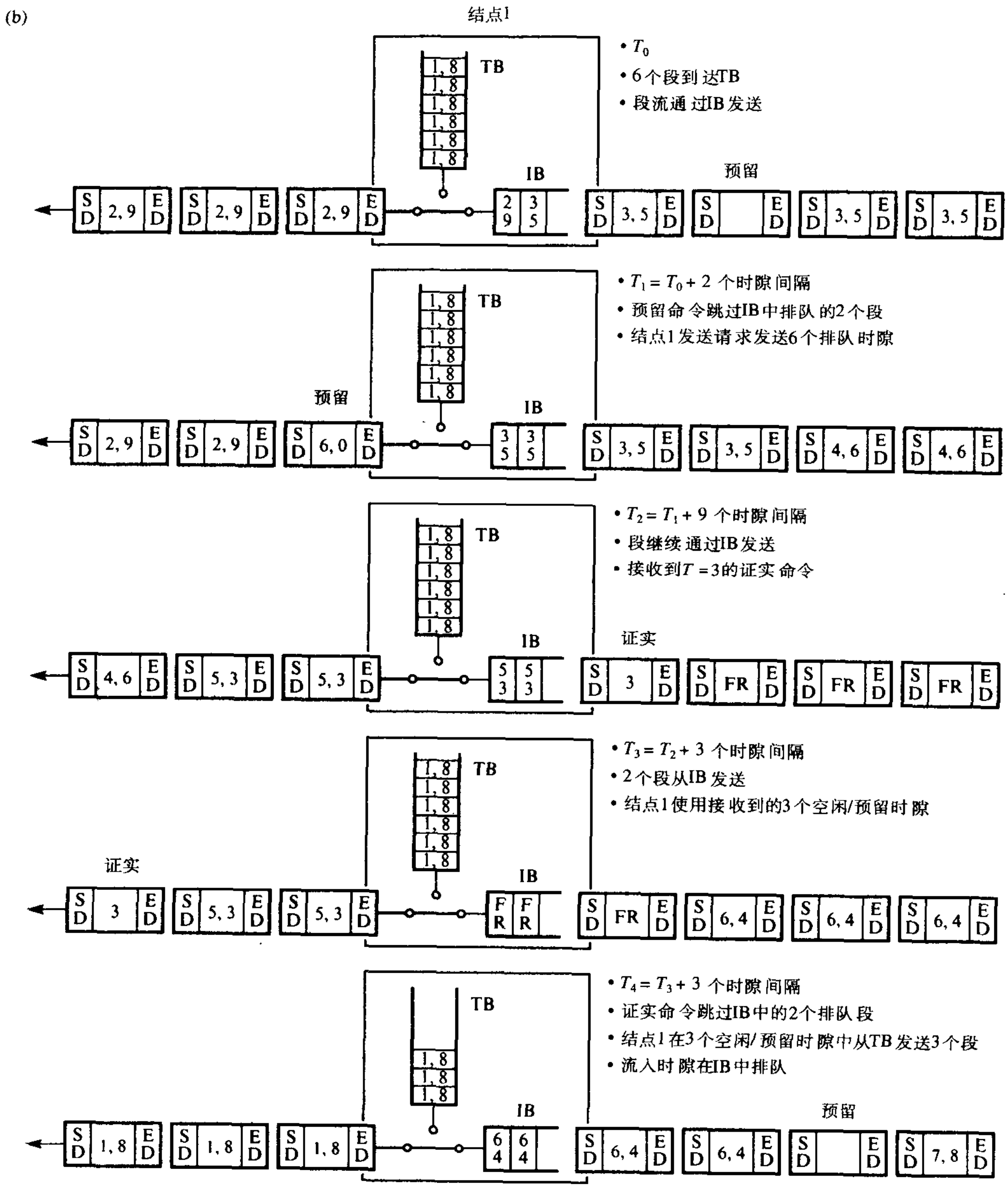


图10-33

(a) 轻度通信状况和使用空闲无条件时隙的立即访问方式下的帧传输实例

(b) 繁忙通信状况和使用空闲预留时隙的预留访问方式下的帧传输实例

(c) 繁忙通信状况和使用空闲无条件时隙的延迟访问方式下的帧传输实例



FR = 空闲预留时隙

图10-33 （续）

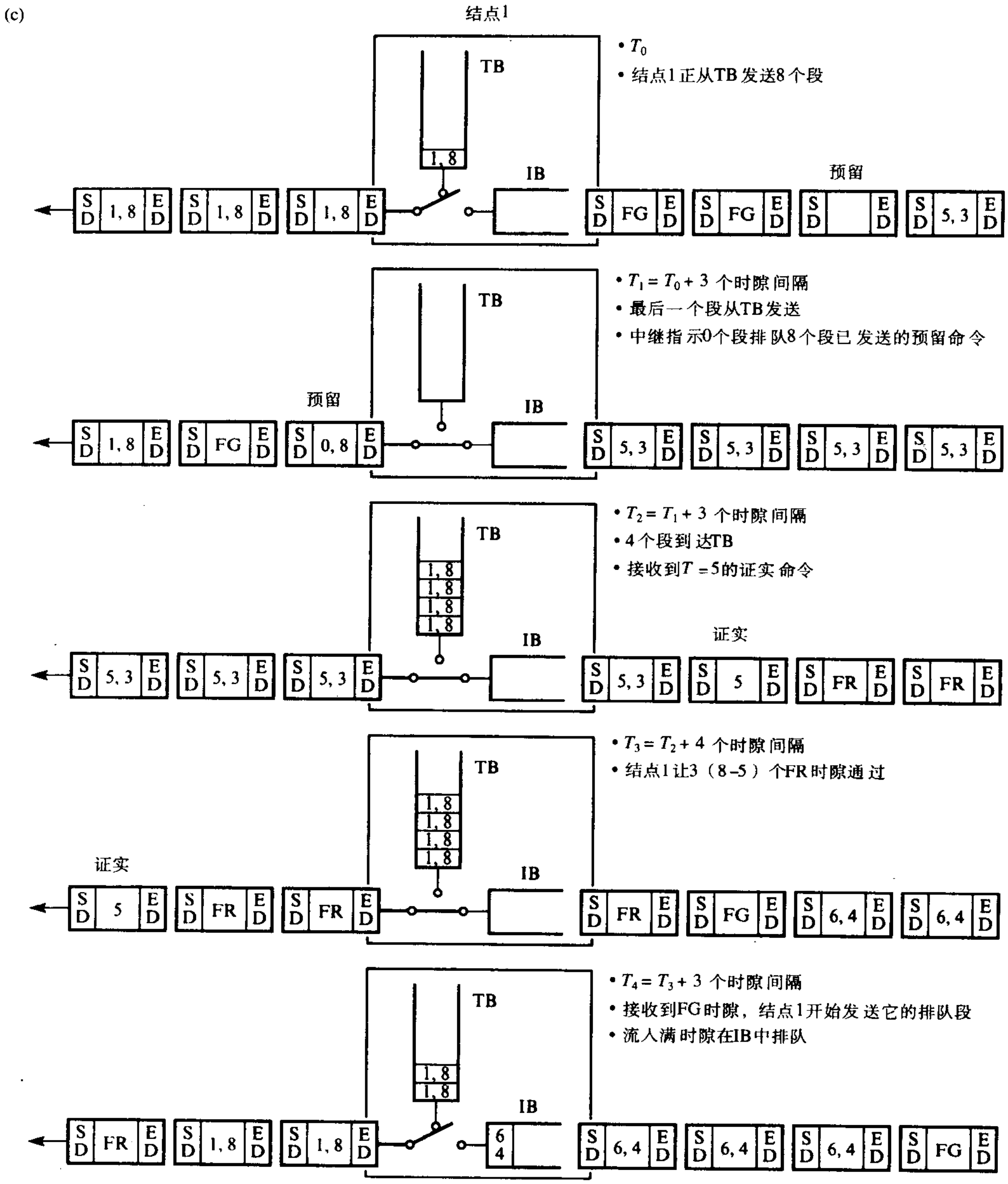


图10-33　（续）

（c）使用延迟访问方式时的时隙传输序列如图10-33(c)所示。当解释该图时应注意下列几点：

• 该结点在接收到预留命令前刚完成两个排队4段帧的发送。
• 该结点在预留命令中指明它在本周期中有0个时隙排队并且已发送了8个时隙。
• 接收到要传送的第二个4段帧但这次环忙。
• 证实命令含有一个只有5的阈值，因此该结点计算出在能开始传送它的排队段前必须延迟（让过）3个空闲无条件时隙。

习题

10.1 对表10-1中的各种媒体类型使用压缩带宽值，估计图10-1中每种应用所需的传输带宽。

10.2 借助图解释：

(a) FDDI-II环的操作原理。

(b) 如何划分链路传输带宽以提供多种业务。

(c) 用于连接站的协议体系结构。

(d) 利用单宽带通道满足各种语音和视频业务的实例。

10.3 解释信元网络的操作原理以及为什么选择53字节作为信元长度。

10.4 推出下列呼叫类型的信元流的概要结构（利用信元数量和它们之间的时间间隔）：

(a) 电话

(b) 可视电话

(c) 电子邮件（数据）

10.5 画出满足下列应用需求的基于DQDB网络的体系结构： 630~633

(a) 单子网MAN

(b) 双地点专用网络

(c) 广域多MAN网络

根据这些，解释用户接口单元、同步网关和子网路由器的作用。

10.6 借助图，区分开放式总线和环形总线DQDB拓扑结构。解释每种类型的操作原理。

10.7 借助图，描述DQDB环形总线子网如何能在（a）通信链路故障和（b）结点故障存在下继续工作。

10.8 在DQDB网络环境中：

(a) 定义术语“时隙”、“信元”和“段”的含义。

(b) 解释时隙、信元和段头部的功能。

10.9 借助图，介绍组成DQDB协议体系结构的MAC和物理层的主要功能块，并解释它们在无连接、面向连接和同步业务条件下的作用。假定使用的物理链路是E3数字电路。

10.10 借助图，解释用在DQDB子网的队列仲裁访问协议的操作原理。包括说明：

(a) 每个时隙头部中忙位和请求位的作用。

(b) 请求、递减和请求队列计数器的功能。

(c) 只访问一条总线的算法流程图。

10.11 解释为什么带宽平衡要用在DQDB队列仲裁访问协议中。描述实现带宽平衡的算法，并根据习题10.10中得出的流程图确定在哪一步执行它。

10.12 借助图，解释DQDB子网的分布式优先级队列机制的操作原理。包括，当：

(a) 没有段排队等待传送。

(b) 一个段在指定优先级级别排队。

使用的计数器的功能以及对于请求位和忙位内容的影响。

10.13 借助图，说明组成DQDB时隙5字节头部的字段并解释它们的功能。

10.14 假定一个数据帧要经过DQDB子网发送并且该帧需要多个时隙，画图说明每个时隙的净荷字段格式并解释每个字段的功能。

描述当重装该帧时接收方要执行的步骤。

10.15　定义由SMDS网络提供的服务。
假定使用SMDS网络，画图说明当每个用户地点使用（a）网桥和（b）路由器时，每个SMDS边界网关的协议体系结构。

10.16　解释SMDS网络边界网关中使用的SIP 3级（MAC会聚）协议的功能。
定义用在IMPDU头部的源和目标地址的格式以及SMDS服务数据单元的长度为什么能达到9188个字节。

10.17　一个1000字节MAC帧要经过DQDB SMDS网络传送。使用定义在图10-27中的传输开销，推出执行帧传输所需的队列仲裁时隙数以及使用的开销字节总数。注明有关的假设。

10.18　概要地解释ATMR网络访问控制协议使用窗口限制和全局重置规程如何工作。

10.19　使用类似于图10-29的一组图，假定窗口计数为2并且只有AU 1是活动的。说明AU 1如何发送4个信元给AU 3。包括环如何重置。

634 10.20　定义ATMR网络术语“重置周期”的含义。

10.21　借助状态变迁图，解释用在ATMR网络中的多优先级协议的操作原理。在描述中包括引起状态改变的条件以及涉及的各个计数器的功能。

10.22　CRMA-II网络的时隙头部内的访问控制字段有忙/空闲位和无条件/保留位。解释它们如何一起用来注册在网络中发送帧的插入。

10.23　借助图描述在CRMA-II网络中站如何竞争成为调度结点。还要描述结点如何确定它们在特定调度周期中能使用多少保留时隙。

10.24　假定只使用空闲无条件时隙，用类似于图10-33的一组图说明结点如何中断发送由四个时隙组成的等待帧的流入时隙流。
使用相同一组图说明当网络严重负载时预留/命令操作如何工作。

10.25　借助图以及路由选择表记录实例，解释信元如何通过ATM交换机被路由。解释为什么每个信元头部可以相对短。

10.26　借助图解释VP路由选择和VC路由选择之间的差异。给出每种路由选择类型的实例。

10.27　基于图10-4所示的ATM LAN体系结构，画出支持所示的相同服务集但只由两个RCU和单个ATM交换机组成的小ATM LAN。

10.28　使用习题10.27得到的单交换机的ATM LAN拓扑，确定用于下列呼叫类型的适合路由选择表记录（两个RCU和一个交换机），假定两个通信多业务工作站连到不同RCU：
（a）可视电话呼叫；
（b）对基于信元连网服务器的访问。

10.29　借助图，解释ATM交换机的操作原理，包括输入和输出控制器、交换矩阵和交换控制处理器的作用。还要解释为什么在输入和输出控制器中需要信元缓冲区。

10.30　解释下列ATM交换矩阵类型的操作原理：
（a）时分总线
（b）全连接
假定它们以非阻塞方式工作，估计与每种矩阵类型相关的I/O端口的最大数量以及取决于什么。

10.31　解释为什么大型交换结构由多级交换组成，每一级又由若干在正规矩阵互连的较小交换单元组成。

10.32 在由许多交换单元组成的多级delta交换矩阵环境中，如果M指输入端口数而N指每个交换单元的输入端口数，得到每级交换单元数X以及级数Y的公式。

10.33 画出由2×2交换单元矩阵组成的8端口delta交换矩阵图。

解释信元如何通过矩阵路由以及阻塞如何发生。

10.34 避免阻塞的交换矩阵的实例是Batcher-Banyan矩阵。解释这种交换机的操作原理以及阻塞如何被避免。

10.35 使用图10-10所示的Batcher-Banyan交换矩阵，确定下列信元组经过矩阵的路径。假定它们同时到达8个输入端口并且在端口1开始给出的路由选择标记为：111、100、011、000、101、001、110、010。假定现在到达端口4的信元路由选择标记是111而不是000。确定它的影响以及实际中如何被克服。 635

10.36 借助图说明用于连到ATM LAN的多业务工作站的协议体系结构。在图中包括与U平面、C平面和M平面相关的协议层。

解释每个协议层的功能并给出与每个平面相关的协议集应用的实例。

10.37 以图的形式总结ATM适配层（AAL）的四个级别的功能。由此解释与每个服务级别相关的两个AAL子层协议的功能。

10.38 说明ATM信元中的48字节信息字段如何由分段和重装（SAR）协议子层用于AAL 1和AAL 2。解释两种情况下每个字段的功能。

10.39 解释AAL 3/4和AAL 5层之间的差异以及为什么要开发后者的原因。

使用图10-14中所示的会聚子层（CS）和分段重装（SAR）子层的PDU的格式，求使用（a）AAL 3/4和AAL 5发送1024字节AAL服务数据单元所需的协议开销（8位组）字节数。

10.40 解释与图10-4所示ATM LAN相关的广播服务器的作用。

概述连到RCU的多业务工作站用户建立通过信令控制点（SCP）的可视电话呼叫要执行的步骤。还要描述这个规程如何扩展以使用广播服务器建立桌面电视会议会话。

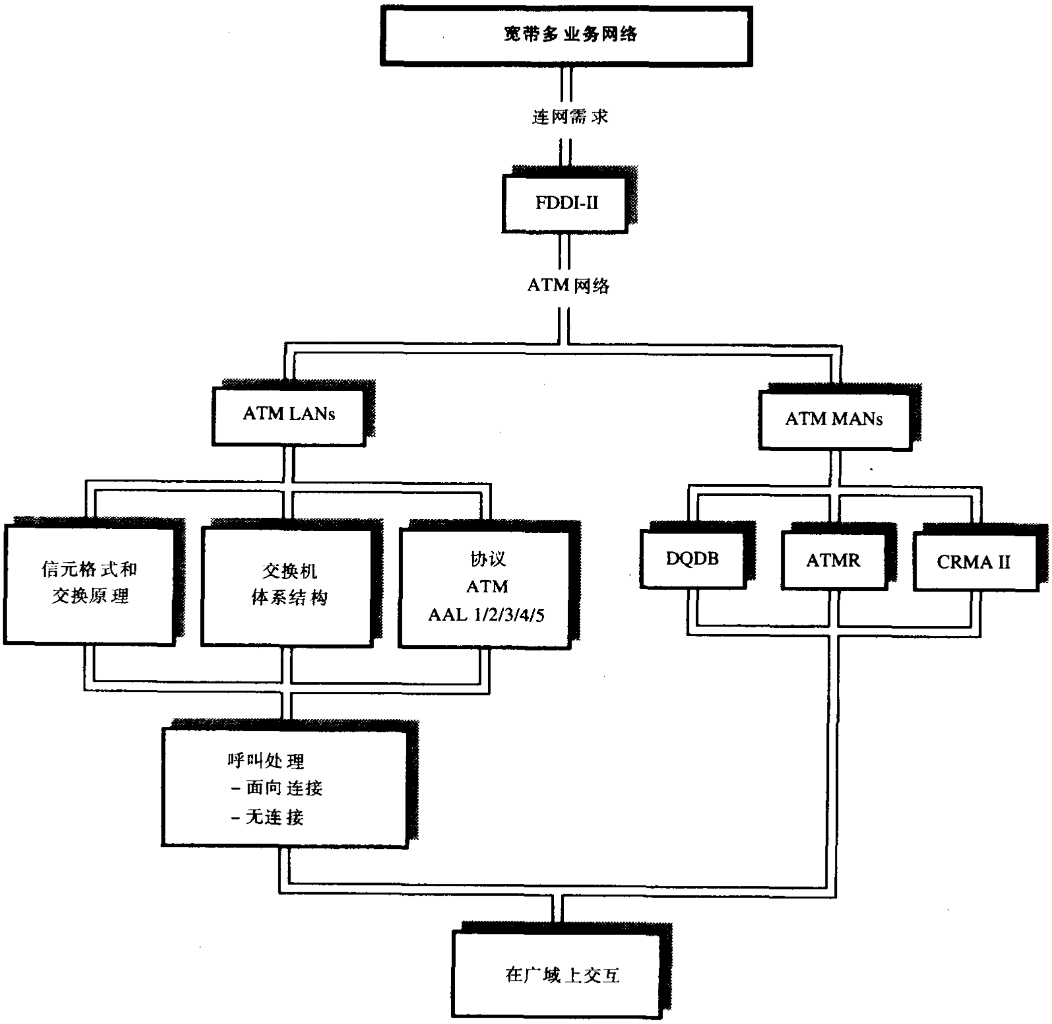
10.41 借助图10-15和图10-16所示的无连接协议体系结构，概述要执行的步骤使得：

（a）基于信元的客户端能访问基于信元的服务器。

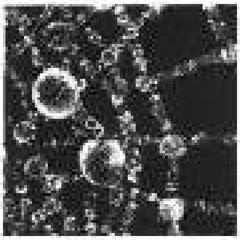
（b）连到遗留LAN的工作站能访问连到不同遗留LAN（相同类型）的服务器，两者都通过网桥连到ATM骨干网。

（c）连到遗留LAN的工作站访问直接连到ATM骨干网的服务器。假定遗留LAN通过路由器连到骨干网。 636

本章概要


宽带多业务网络
连网需求
FDDI-II
ATM 网络
ATM LANs
ATM MANs
信元格式和
交换原理
交换机
体系结构
协议
ATM
AAL 1/2/3/4/5
DQDB
ATMR
CRMA II
呼叫处理
－面向连接
－无连接
在广域上交互

637



第三部分 开放系统

本部分将描述启用一组应用程序的附加协议的功能和操作，这些应用程序运行在不同厂商生产的计算机中，并通过不同计算机网络互连，使得彼此之间能够通信，并执行特定的分布式应用功能。这种协议称为面向应用的协议，产生的通信环境称为开放系统互连环境。有两组面向应用的协议：TCP/IP协议族和OSI协议族，会分别进行讨论。

第11章介绍了TCP/IP协议族和OSI协议族中面向应用协议的功能概况，给出两种协议把不同网络类型提供的各种服务转换成独立于网络的信息传输服务的详细描述。

第12章介绍并描述了OSI协议族中提供常规服务和应用支持服务的协议，包括建立会话连接、提供语法转换（如果两个通信计算机上的数据表示不同）和其他支持服务所需要的协议。

第13章将讨论两个协议族中面向更特定应用协议的有关功能和操作，包括支持在不同邮件和文件系统之间传递电子邮件和文件的各种协议。

第14章将讨论为了实现开放系统互连而必须描述的额外系统功能，包括称为目录服务的名称到地址映射，以及在完整的协议族中，协议如何与相似协议族相互协作并交换信息。 639

第11章　传输层协议

本章目的

读完本章，应该能够：

- 了解在TCP/IP协议族和OSI协议族中的各种传输层协议；
- 了解TCP/IP协议族中使用的层间地址选择器的作用；
- 理解UDP的作用，并能描述UDP数据报头部中各个字段的格式和含义；
- 理解TCP的作用，以及在可靠的流服务中使用的服务原语；
- 解释与TCP相关的 PDU（段）的格式，以及在它的头部中各个字段的含义；
- 描述TCP连接建立、数据传递和连接终止等阶段；
- 理解ISO标准文档中用来描述协议层提供的服务、协议操作以及协议实体的规范的相关术语；
- 描述ISO无连接和面向连接传输层协议的服务和操作；
- 理解ISO传输层协议的形式化规范，以及使用结构化程序代码的实现方法。

引言

图11-1显示了传输层在OSI协议族以及TCP/IP协议族中的位置。

641 在采用TCP/IP协议族的网络上，无论基本数据通信网络是单个LAN或WAN或者互联网，网络层总是采用IP协议。从而，IP和传输层协议紧密地耦合在一起。并且，所有与传输层协议相关的协议数据单元（PDU）都通过底层网络以IP数据报进行传输。

相对而言，在OSI协议族中，面向网络的协议通常要反映出底层网络的类型。这就意味着，如果底层网络基于X.25，网络层接口是面向连接的；如果底层网络利用了ISO CLNP，则网络层接口是无连接的。此外，在TCP/IP协议族中，传输层直接向应用协议提供服务，而在OSI协议族中，它通过中间的会话层与表示层向应用协议提供服务。

642 尽管有这些差异，两者的传输层都实现了一组相似的功能。因而，两个协议族中既有最佳尝试（无连接）协议，又有一种可靠的（面向连接）协议。显然，不是所有的应用都需要可靠的服务。例如，对于包含数字化图像的数据文件的传输，保证传输速度比偶然出现传输错误更加重要。对于数字化语音信息的传输要求也与之类似。然而，要注意虽然在LAN中，误码率是比较小的，但是并不等于零。因此应该只对那些可以接受偶然错误的应用采用无连接的传输协议，而对于其他的应用，应该采用面向连接的传输层协议。

下面将分别描述两个协议族中的传输层协议，首先介绍TCP/IP协议中的UDP协议和TCP协议。

11.1　用户数据报协议

在TCP/IP协议族中传输层时常与IP交互操作。回忆一下，该协议对单独编址的称为数据报的信息提供最佳尝试（无连接）传输服务。

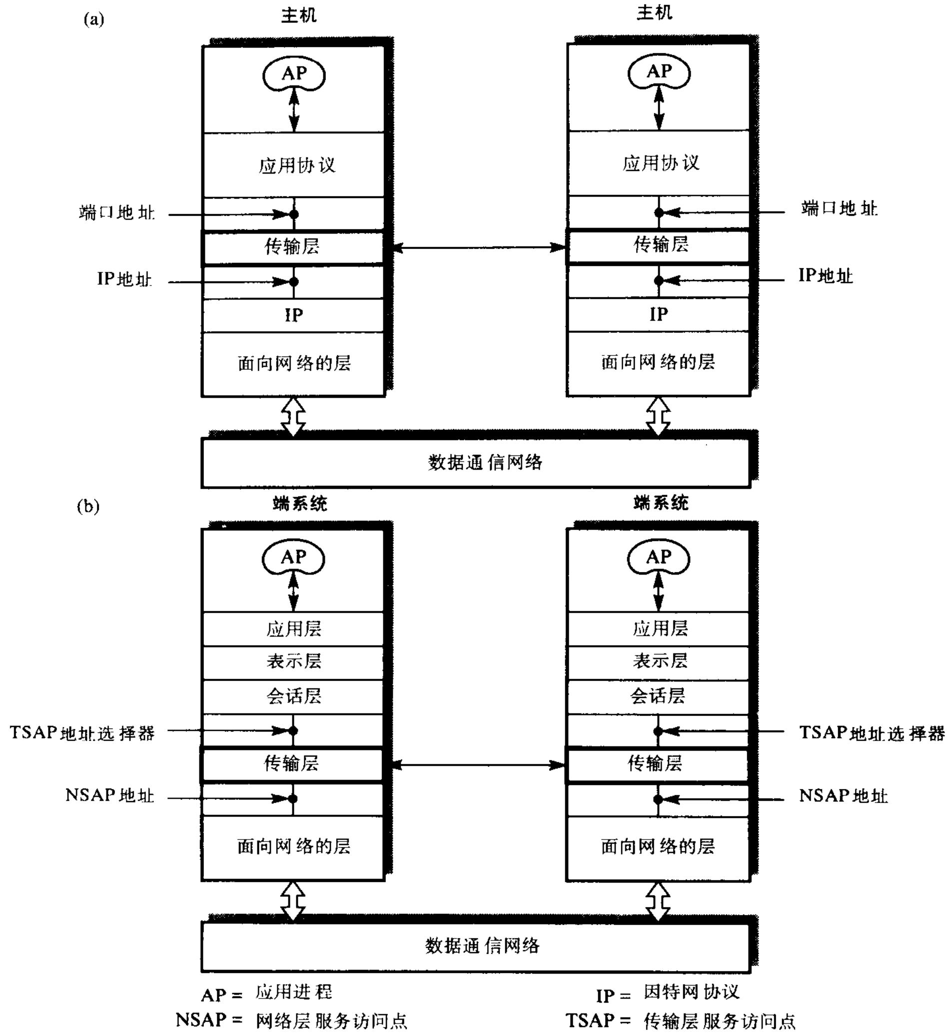


图11-1 传输层位置

(a) 在TCP/IP协议族中 (b) 在OSI协议族中

因为在发送报文（数据报）之前，不建立网络连接，所以这种方式传输报文的开销最小。IP层不执行任何差错控制。为了使应用协议能够使用这种特性，TCP/IP协议族提供了一种无连接的传输层协议，称为**用户数据报协议**（UDP）。

另外TCP/IP提供了一种带有可靠服务的用户应用进程，称为**传输控制协议**（TCP）。TCP协议和UDP协议与其他协议的位置关系，每个协议的PDU以及层间地址选择器如图11-2所示。本节描述UDP协议，在下一节描述TCP协议。

图11-2显示了主机传输/接收的信息帧的主要结构。在网络接口处，包含LAN/WAN帧头部，信息（用户数据）字段和一个相关的尾部（帧结束标记加CRC）。在9.5.1节说过IP地址由网络地址和主机地址组成，通过它可以把数据报发送到特定主机。数据报头部的协议字段指明了数据报中的用户数据要发送到的主机上的协议，它可能是与IP相关的某一个协议，例如ICMP，或者UDP与TCP协议中的某一个传输协议。

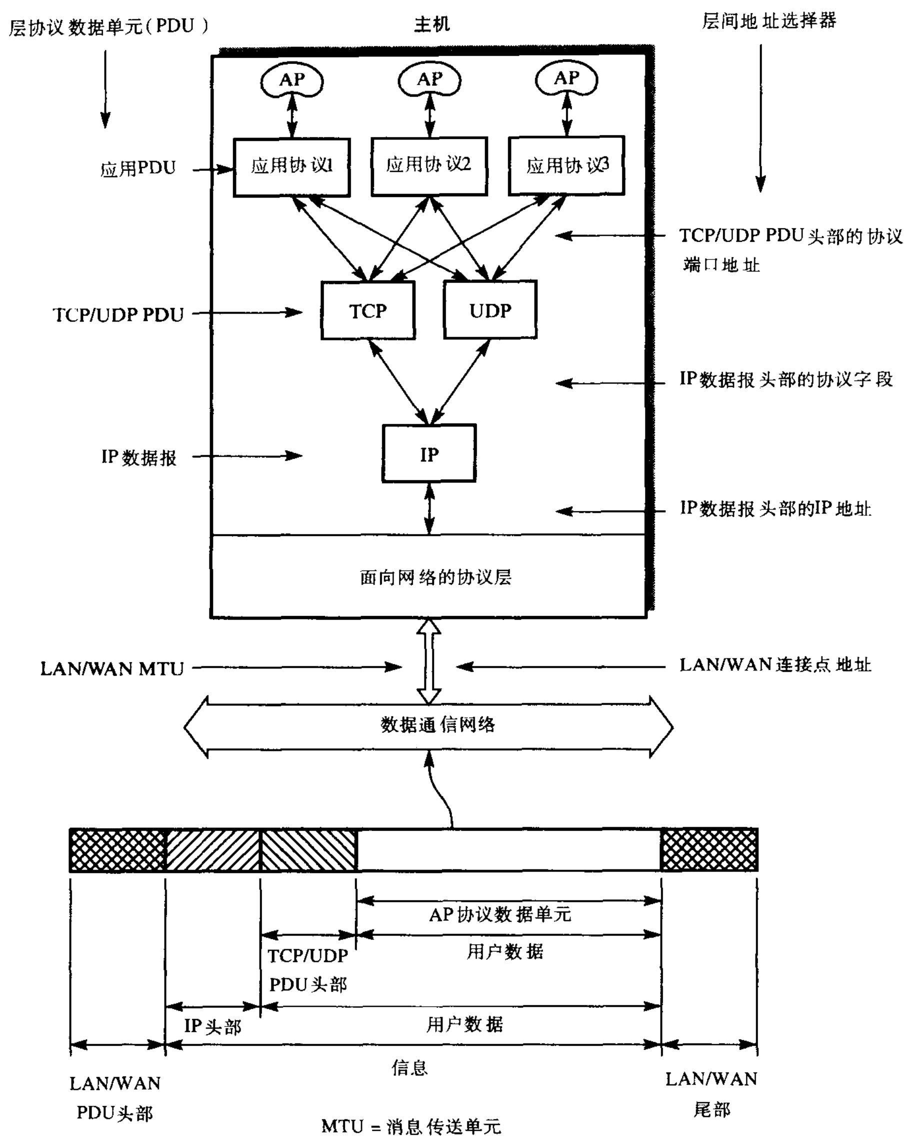


图11-2　TCP/IP和相关PDU以及层间地址选择器

从图11-2可以看到，UDP和TCP可以同时为多个应用提供服务。因此，对于UDP协议（或TCP协议），必须在每个PDU的头部再添加一个附加的地址，根据这个附加地址把各自PDU的用户数据部分传递给相关的应用协议。在OSI协议族中，这个功能由层间**传输服务访问点**
643 （TSAP）（或**地址选择器**）实现，而在TCP/IP中则使用（**协议**）**端口地址**来实现。

对于TCP/IP，一个应用协议的合成地址由主机的跨互联网IP地址加上附加的协议端口地址组成。在点分十进制表示法中，下面是应用协议地址的一个例子：

128.3.2.3, 53

第一部分是IP地址。其中网络地址是128.3主机地址是2.3；第二部分是端口地址（53）。在ISO术语中，这种复合地址被称为**完全限定地址**，它由跨互联网的NSAP地址加上位于传输层和其
644 他更高层协议之间的层间地址选择器构成。

如前所述，UDP是一个无连接的协议，在IP数据报中的用户数据字段中仅有一个PDU被发送，该PDU也称为数据报。为了把IP数据报和包含在UDP的用户数据字段中的数据报区分开来，我们称后者为**用户数据报**。图11-3显示了用户数据报（UDP PDU）的头部格式。

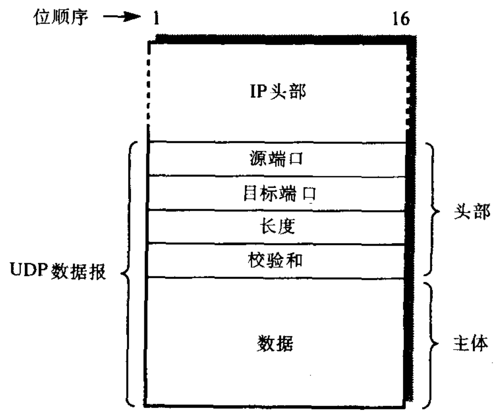


图11-3 UDP数据报头部格式

源端口指的是源（发送方）应用协议的端口地址，**目标端口**是接受方应用协议的端口地址。两者都是16位整数。在UDP数据报中，源端口是可选的，仅当需要答复时才被选用，否则源端口一般都设置为零。

长度字段是整个UDP数据报以8位组计算的总个数（包含UDP头部）。由于IP头部中的校验和字段只计算了IP头部中各字段的校验和，没有包含用户数据字段，因此，在UDP头部中的**校验和**字段包含了整个UDP数据报的校验和。在UDP的各种应用中校验和也是可选的，如果未使用，则它通常被设置为零。要注意的是采用反码表示，零可以是二进制的全0或全1，所以如果使用了校验和，而且校验和结果为零时，把校验和字段设置为二进制的全1。

因此，在一个应用协议发送用户数据报给远程主机上的应用协议之前，我们必须知道目标进程的IP地址和端口地址。这是应用程序而不是UDP或TCP的功能，将在11.4.2节讨论它。

11.2 传输控制协议

当数据的传输不需要纠错或者两个应用协议之间交换短请求/应答报文时，采用UDP协议。然而，在大多数的开放分布式应用中，需要可靠的（面向连接的）信息传输服务。例如，客户银行记录的文件内容的传输。显然在这样的应用中，即使一个简单的二进制位的错误都是严重的。 645

在TCP/IP协议族中，面向连接的传输层协议是**传输控制协议（TCP）**，通过应用层协议，它提供给用户的服务被称为**可靠流传输服务**。

11.2.1 可靠流传输服务

可靠流传输服务与OSI协议族中级别4的传输层协议用户服务是类似的（见11.6节）。提供服务原语，使得应用协议通过ISO协议族中的表示层和会话层可以同远端主机的（对等）应用协议建立逻辑连接，并在此连接上以双工（双向同时）方式交换信息，以及拆除连接。TCP协议目的就是以一种可靠的方式传送与这些交换相关的数据，这意味着没有重份和无差错的数据传输，并且保持数据流原顺序。

由于TCP处理所有与连接相关的用户数据，所以引入了术语“流”。例如，一系列的请求/响应消息可以被看作两个独立的数据流，每个方向上一个，每一个都包含了一串8位组/字节。为了实现服务的可靠性，TCP以**段**为单元发送数据。通常由TCP协议来决定何时传输一个新的段。目标方的TCP协议把接收到的数据按段存入与应用相关的缓存中，当缓存满了之后再把数据向上传递。因此当交换短的消息单元时，一个段可能会包含多个用户消息，或者发送一个大的数据文件时，报文段包含的是大报文的一部分。每个段的最大长度是TCP的函数，要尽量确保每个方向上发送的八位组流都能以可靠的方式传递到另一方。

另外，允许用户强制消息单元立即发送与传递，例如，一些数据的简短请求消息，这时用户可以通过设置数据发送请求原语中的参数来要求数据直接发送。同样，用户可以通过设置参数使数据以**紧急**的方式发送，这时数据会以不同于常规数据的流量控制机制被TCP发送。它提供与OSI协议族中加速数据服务相似的功能。

表11-1列出了TCP中的服务原语以及各原语的参数。大多数开放分布式应用是基于**客户—服务器模型**的。通过观察一个访问和使用远程文件系统的应用程序（进程）可以更好地理解这一点。文件系统是服务器，因为它只对文件服务的请求做出响应，而用户应用程序是客户，
646 因为它总是发起请求。注意单个的服务进程必须能同时支持分布式的多客户的访问请求。多数用户服务原语都反映了这种交互模式。有关用户/服务器的每个类型在表11-1中指明。

表11-1 TCP用户服务原语及其相关参数

原 语	类 型	客户端/服务器	参 数
UNSPECIFIED_PASSIVE_OPEN	请求	服务器	源端口、超时、超时动作、预处理、安全范围
FULL_PASSIVE_OPEN	请求	服务器	源端口、目标端口、目标地址、超时、超时动作、预处理、安全范围
ACTIVE_OPEN	请求	客户端	源端口、目标端口、目标地址、超时、超时动作、预处理、安全范围
ACTIVE_OPEN_WITH_DATA	请求	客户端	源端口、目标端口、目标地址、数据、数据长度、推入标记、紧急标记、超时
OPEN_ID	本地响应	客户端	本地连接名称、源端口、目标端口、目标地址
OPEN_SUCCESS	证实	客户端	本地连接名称
OPEN_FAILURE	证实	客户端	本地连接名称
SEND	请求	客户端/服务器	本地连接名称、数据、数据长度、推入标记、紧急标记、超时、超时动作
DELIVER	指示	客户端/服务器	本地连接名称、数据、数据长度、紧急标记
ALLOCATE	请求	客户端/服务器	本地连接名称、数据长度
CLOSE	请求	客户端/服务器	本地连接名称
CLOSING	指示	客户端/服务器	本地连接名称
TERMINATE	证实	客户端/服务器	本地连接名称、原因代码
ABORT	请求	客户端/服务器	本地连接名称
STATUS	请求	客户端/服务器	本地连接名称
STATUS_RESPONSE	本地响应	客户端/服务器	本地连接名称、源端口、源地址、目标端口、目标地址、连接状态、接收窗口、发送窗口、等待确认、等待接收、紧急、预处理、安全、超时
ERROR	指示	客户端/服务器	本地连接名称、原因代码

目标地址参数是目标主机的IP地址，**源端口**和**目标端口**分别是源协议和目标协议使用的层间端口地址。

超时参数允许用户应用协议指定一个最大的时间间隔，源TCP在该时间内等待一个已发送报文段的确认。记住因为IP提供的是最佳尝试服务，所以发送的报文段在传输期间可能会被丢弃。因此，超时值通常设置为大于每个IP数据报携带的生存周期值的两倍。**超时动作**参数指定了超时发生后要采取的动作，通常是关闭连接。

预处理级参数是一个参数集合，允许用户指定用来传输TCP报文段的IP数据报头部的服务类型字段的值。当讨论的是IP协议时，图9-9显示了这个字段的内容。它与相应的ISO原语中的QOS参数执行相同的功能。要注意这个参数是对IP协议使用的，而不是对TCP协议。**穿越**参数是一个实例，仅仅从一个协议层向下一个协议层传递，而不作修改。 647

安全范围参数允许服务器应用协议为潜在的用户指定安全级。**推入标记**和**紧急标记**使用户可以向TCP协议指示应该如何处理**数据**参数中的数据。推入是立即发送数据，急迫是在正常流外另外发送。最后，当一个连接建立后，由本地TCP实体分配一个**本地连接名称**。虽然，端口地址使TCP可以把收到的报文段与特定的应用协议关联起来，在服务器情况下，多事务服务（逻辑连接）可以并发的运行。对同一个连接，TCP通过**本地连接名称**使用相关服务原语。OSI协议族中，连接端点标识符参数执行同样的功能。STATUS原语中的大部分参数都是与TCP实体相关联的，所以将在11.2.2节讨论。

多数参数都是与传输协议TCP和IP中的低级别操作相关的。因此，在很多情况下，不同的原语中会包含相同的参数，其中斜体的参数都是默认值。如果一个参数的值没有被明确地提供，那么将假定默认值。图11-4显示了各个不同原语之间的相互关系的时序图。

服务器通常用UNSPECIFIED_PASSIVE_OPEN和FULL_PASSIVE_OPEN原语指示连接请求准备就绪。术语“被动”用来表示已经准备好接收一个连接请求，而不是（主动）发起连接建立。“未确定打开”指示服务器准备好接受来自指定安全级别（如果给出）的任一进程的请求。“完全打开”包含了一个目标端口和IP地址参数，用来指示服务器等待请求的特定的应用协议或进程。通常，从开始运行直到机器关闭一直保持服务器打开。

在客户端，一个用户通过应用协议可以使用两种方法发起一个连接的建立。第一种，客户协议可以发出ACTIVE_OPEN请求指定目标的端口和IP地址来建立连接。本地的TCP返回一个OPEN_ID响应，通知标识符的应用协议，标识符即赋予本次连接请求的连接名。然后发起一次连接建立。如果成功，双方就可以开始用SEND和DELIVER原语进行数据信息的交换了。

第二种，在客户端可以发送一个ACTIVE_OPEN_WITH_DATA请求原语建立连接。正如名字一样，它在参数中包含了用户数据信息。和第一种方法相同，本地TCP将返回一个连接名。然后，它发起连接建立的同时发送数据信息。由于有关呼叫开销降低，这种模式通常被用来进行简短的（可靠）请求/响应数据消息的交换。 648

如图11-4所示，如果需要，连接双方要使用带有推入和急迫标记的SEND/DELIVER原语进行数据传输。用户使用ALLOCATE原语在本地为一个连接增加消息缓存区的数量。另外，在数据传输阶段，用户可以通过STATUS原语请求连接的状态。本地TCP使用STATUS_RESPONSE原语响应。如果有错误发生，本地TCP使用ERROR原语提示用户发生错误，例如，当双方TCP协议实体不同步时。

最后，可以使用两种方法来释放连接。因为每一方的消息流都是独立控制的，所以连接要在每一方独立地释放。这被称为**适度拆连**，并且在向本地TCP发送完所有的数据后，需要双方用户独立地发出一个CLEAR请求原语。另外一种方式是用户发送一个ABORT原语，它使得双方丢弃所有的未完成数据并中止连接。 649

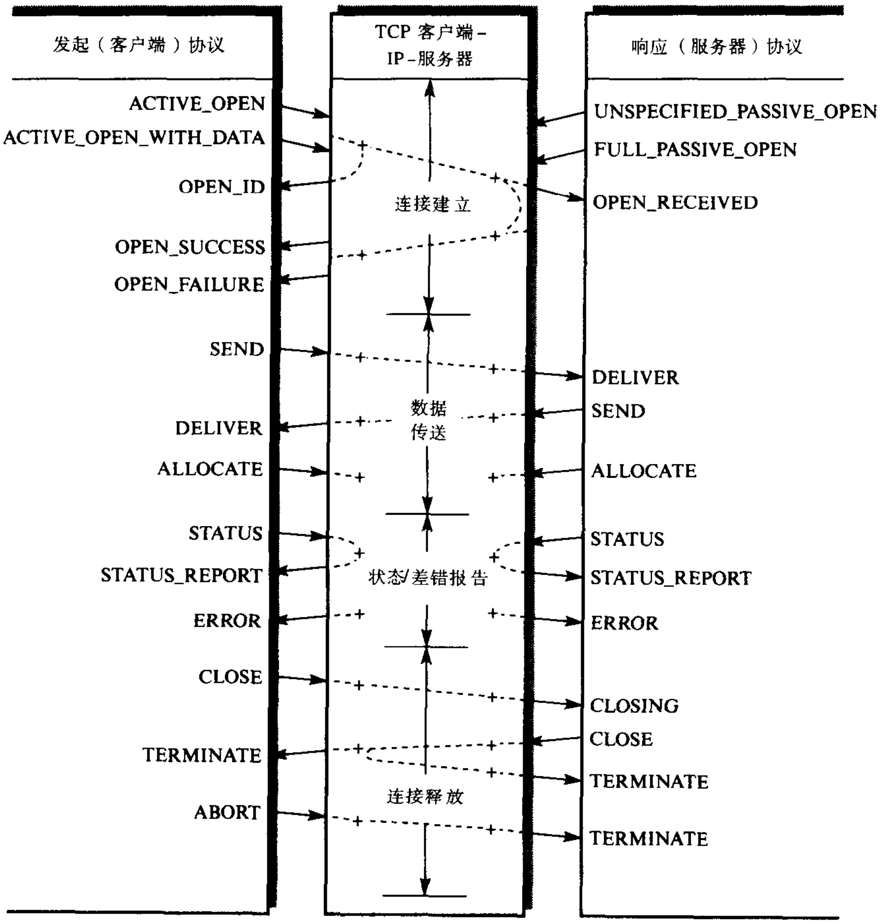


图11-4　TCP用户服务：时序图

11.2.2　协议操作

TCP协议包含了在第5章讨论的HDLC协议的许多特点。支持双向的信息传输并且采用一种滑动窗口流量控制机制实现回退N帧差错控制规程。

TCP用来建立连接、传送数据和释放连接的所有PDU都具有标准的格式，称为报文段。所有的报文段在IP数据报的用户数据字段中，在双方的TCP实体中进行传输。图11-5(a)显示了TCP报文段的格式。

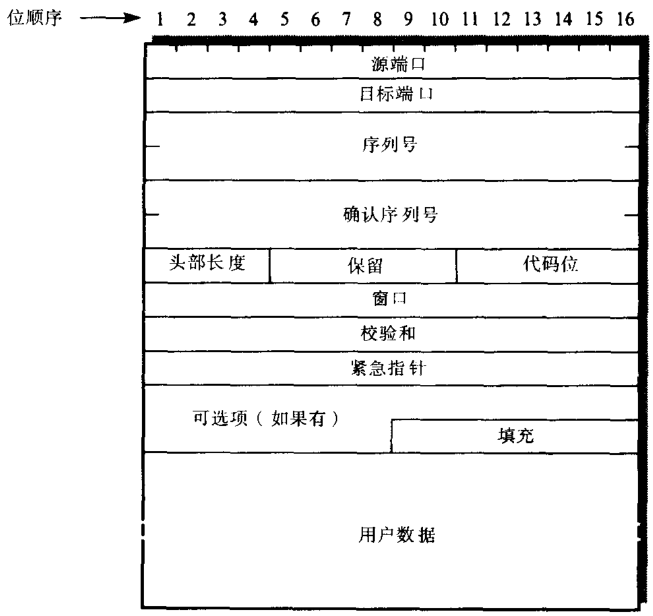
如同UDP数据报，报文头部的前两个字段是**源端口**地址和**目标端口**地址。它们和UDP数据报有着同样的含义，除了TCP两个地址指明了两个应用协议之间的逻辑连接的端点。

序列号和HDLC协议中的发送序列号有着相同的作用，**确认序列号**则和HDLC协议中的接收序列号有着相同的作用。前者与连接数据流中的发送TCP实体相关联，后者则与相反方向的实体相关联。然而，对于TCP，虽然数据是被分块提交并传递的，但是每个方向上的数据都是被视为八位一组的字节流进行差错和流量控制的。因此，序列号和确认序列号是关于整个信息流的八位组的位置，而不是一个报文块在整个信息流中的位置。序列号指的是本报文段的数据字段中的第一个八位组相对于整个信息的开始处的位置，确认序列号指的是相反方向的发送TCP期待接收的下一个序列号。

由于段头部可选字段的出现使得头部（理论上）具有变化的长度。所以**头部长度**表明了TCP头部共有多少个4字节的字。**保留**字段正如其名字所隐含的意义是作为保留使用的。

所有的报文段（因此PDU类型）都具有相同的头部格式，并且可以通过设置6个比特**代码**字段中的各位来指示段头部中选择字段的有效性。如果某位被设置为1，则相应的字段是有效的。要注意的是一个报文段中可以设置多个位有效。图11-5(b)显示了代码字段中各个位所代表的含义。

(a)



(b) 代码位：

位位置	名称—功能
11	URG—紧急指针字段有效
12	ACK—确认字段有效
13	PSH—接收到这个段就递交数据
14	RST—重置序列号/确认号
15	SEQ—序列号有效
16	FIN—发送方的字节流停止

图11-5 TCP

(a) 段（PDU）格式 (b) 代码位定义

窗口字段是用来实现滑动窗口流量控制的。它表示了在确认序列号字段的字节号之后源方还可以接受多少个数据字节。它是由源方为本连接分配的内存缓存数量的大小决定的。

如同UDP数据报，校验和是针对整个报文段加上头部的内容计算的。它使用反码形式计算报文段中所有16位字的校验和。 650

如果代码字段中的URG标志位被设置了，则**紧急指针**表示报文段中紧急（加速）数据的总数。通常由接受方TCP实体一接收到就立即传递。

数据字段包含的默认最大字节长度是536。那是因为考虑当前网络为WAN的情况时，这种选择会导致较小的误码率。当运行应用协议的两台主机位于一个低误码率的网络上时，例如LAN，可以选用较大长度的数据段。这时发送方TCP实体可以使用**选项**字段来提示接收方TCP实体，指示准备接收的报文段中用户数据字段的最大字节长度。

1. 建立连接

虽然在客户—服务器的交互过程中，通常是由客户端发起连接建立并初始化连接设置，但是很多应用并不是基于客户—服务器模式的，双方可能同时试图建立连接。考虑到这种情
651 况，TCP采用了一种三路消息（报文段）交换的方法来建立连接，被称为三次握手规程。同样，如11.2.2节所述的，连接中每个方向上的数据流都是被独立控制的。为了避免任何可能出现的不确定性，连接的两端都会设置初始的序列号，每个用户都通知对方它所希望的初始的序列号。这作为握手规程的一部分得到确认。图11-6提供了两个段交换的实例。

发起方发送带有SEQ标志并且要在序列号字段中设置期望的初始序列号（seq=X）的报文段来建立连接。响应方在接收时首先要为连接进来的序列号做一个标记，然后返回一个报文段：此报文段SEQ和ACK标志置位，序列号字段设置为自己的初始序列号（seq=Y），确认序列号字段置为X+1（Ack=X+1）表示已经注意到了对方的连接请求初始值。这时发起方在接收返回的报文时也要为连接进来的序列号Y做一个标记，然后返回一个ACK标志置位的报文段，其中的确认序列号设置为Y+1。考虑特殊情况，如果双方同时发送携带序列号的SEQ报文段时（见图11-6(b)），双方仅需要返回ACK报文段来确认收到的序列号。这样连接就从两个方向建立起来，每个方向都可以独立地发送数据了。

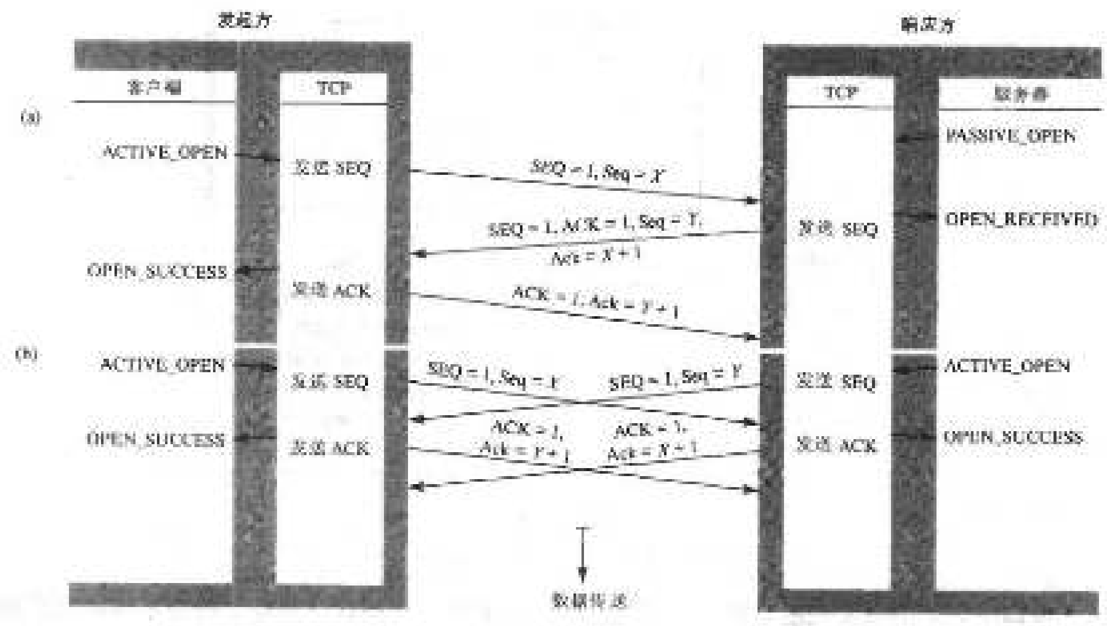


图11-6　TCP连接建立

(a) 客户端—服务器　(b) 冲突发生

2. 数据传送

652 正如前面提到的，数据传送阶段的差错和流量控制策略是以回退N帧差错控制策略和滑动窗口流量控制机制为基础的。因为在第4章和第5章讨论HDLC协议时，已经详细讨论过有关的细节，这里不再讨论它们。然而为了阐述推入标志的用法，考虑图11-7中用于客户—服务器中请求/响应报文段交换的实例。

我们假设客户端发送了一个包含N字节的简短的请求报文，为了紧急传递，把推入标志置

位。发送方TCP实体在报文段里对SEQ和PSH标志置位，并设置序列号为当前值（假设为X），然后立即发送报文段。当接收方接收此报文时发现了PSH标志被置位，然后立即传递，接着返回一个报文段，此报文段ACK标志置位，确认序列号设置为$X+N+1$，用来指示下一个期待接收的字节，窗口值设置为N，即窗口值的原始设定。

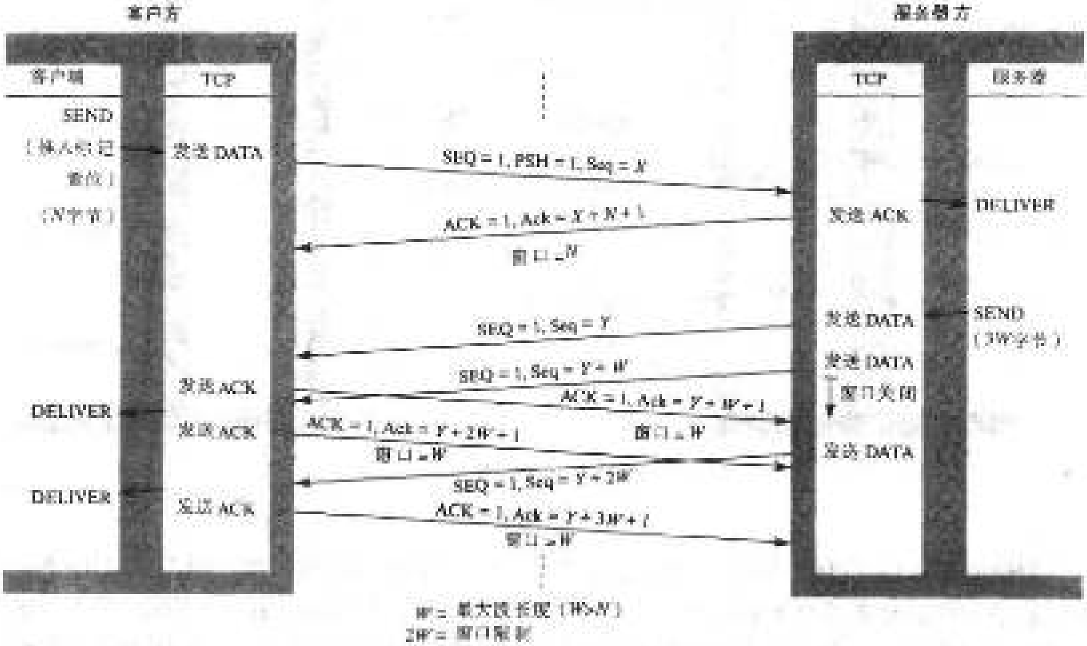


图11-7 TCP数据传送实例

服务器返回报文段响应，假设数据为最大报文段长度W的三倍，所以本地TCP要分三次发送报文段。为了说明窗口机制，假设发送窗口大小为$2W$，所以现在只允许发送两个最大长度的报文段。当发送完两个报文段之后，服务器必须等待，直到在其窗口字段中收到有进一步的信用分配的确认。

客户TCP在接收了每个报文段后要返回一个ACK标志置位的报文段。当客户端接收第二个报文段时，TCP把当前缓存中的内容（包含两个报文段）传递给客户应用进程。在服务器端，当TCP接收到第一个ACK报文段后，将对信用值进行增量，然后发送第三个报文段。图11-7假设接收TCP直接传递第三个报文段给用户应用进程。 659

虽然图中没有显示，但当含有差错次序或差错确认字段的报文段被接收时，接收方将返回一个报文段给发送方，该报文RST和SEQ/ACK标志置位，并在适当的字段中给出预定的值。

3. 释放连接

图11-8显示了对应两种不同的连接释放方法，每个TCP实体所采取的动作。第一个是对应正常的释放，第二个是对应异常的释放。

在图11-8(a)中，假设客户协议端已经发送完所有的数据，要简单地中止连接。因此当接收到CLOSE原语后，客户协议端发送一个带有FIN标志置位的报文段。接收到这个报文段后，服务器发送给服务器协议一个CLOSING原语，然后，返回客户一个带有ACK标志的报文段表示用户确认接收到FIN报文段。

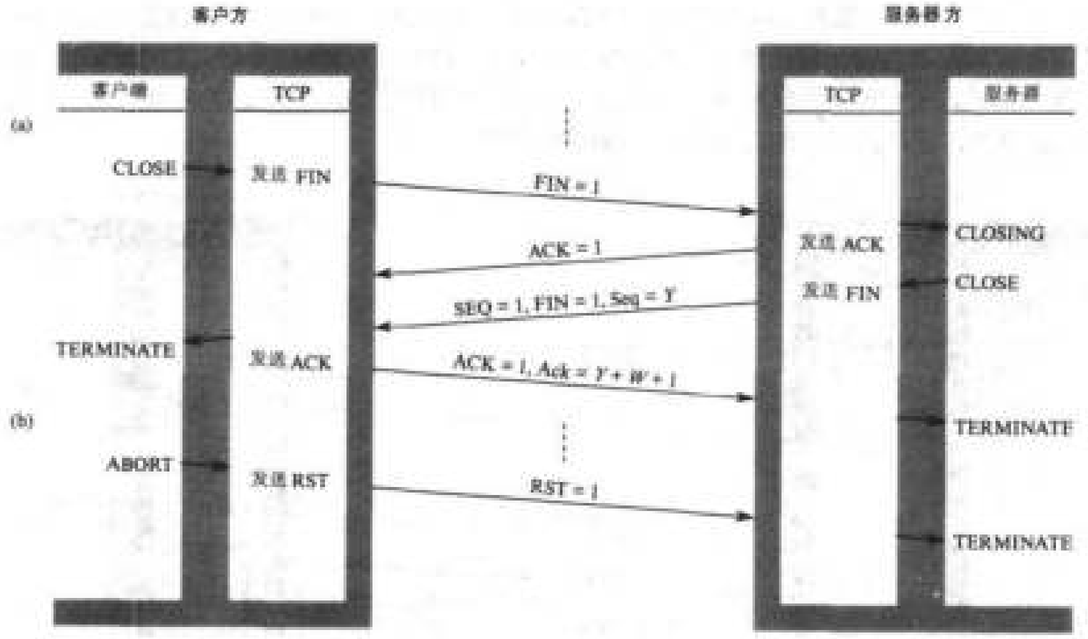


图11-8 TCP连接终止

(a) 正常 (b) 异常

假设服务器协议已经结束了数据传递，然后发出一个CLOSE原语响应CLOSING原语。然而，例子中假设服务器TCP仍然有剩余数据要发送，这些数据是包含在一个SEQ和FIN标志置位的报文段中发送。当收到了这样的报文段时，客户TCP发出一个TERMINATE原语并对刚收
654 到的数据返回ACK报文段。当接收到ACK时，服务器TCP发出TERMINATE原语给服务协议。如果服务器端在两个生存周期后还没有收到ACK报文段，则假设ACK报文段被破坏了，并发出TERMINATE原语给服务器。

当使用异常中止序列时，客户端立即中断两个方向的连接并发送一个RST标志置位的报文段。服务器端接收到这个报文后，立即中止两个方向的连接并发送一个带有连接异常中止的原因代码的TERMINATE原语。

655 为了加强对两个TCP实体之间建立连接和释放连接阶段的理解。图11-9表示了服务器方和客户方的状态变迁图。回忆第4章，每个自动机（有限状态机）的各种状态用圆圈表示，有向线段（称为弧）则表示状态之间的变迁。沿着弧是引起状态变迁的入事件（一个服务原语或一个TCP报文的到达），后跟引起的出事件。图11-6和图11-8显示了建立连接和释放连接阶段的各种报文交换过程，应该从中理解各方的变迁过程状态。

11.3 OSI 协议

当描述OSI协议族中任一个协议的操作时，从一开始就必须区别一个协议层提供的服务、层内部的操作（即协议）以及层使用的服务。这是非常重要的，因为只有这样才能在其他协议的环境中定义一个协议层的功能。每个协议层之间的独立性还使得只需要实现一个简单的协议层的程序员受益。程序员只需要知道本协议层提供什么服务给上一层，本层的内部协议以及下一层提供的用来传递适当信息项的服务，这些信息项是与远端系统中相应的协议层相

关联的。实现这个协议层的程序员除了这些以外不需要知道其他任何额外的知识。

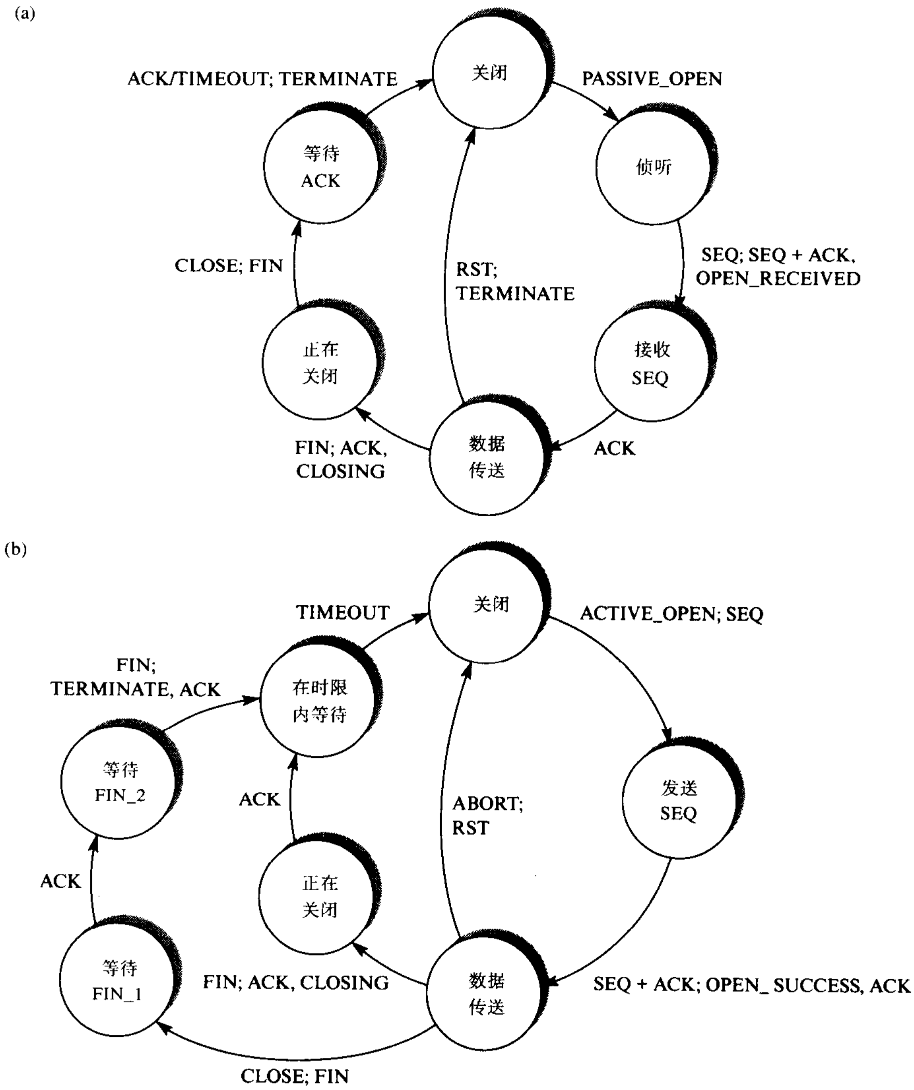


图11-9 TCP状态变迁图
(a) 服务器 (b) 客户端

例如，要描述传输层的功能，只需要考虑下面要点：

1）定义一组提供给会话层的服务（用来把会话层消息单元传输给远端系统中的对等会话层）。

2）传输层的内部操作（例如，同远端系统中对等的传输层建立和管理逻辑连接，通过已建立的连接进行传输层消息单元的差错和流量控制）。

3）由网络层提供服务来与对等传输层传递信息。

当描述一个协议层的功能是，要按以上这三个方面分别考虑。

每一个协议层的规格说明包含两部分的文档：**服务定义文档**和**协议规范说明文档**。服务定义文档包含了提供给上一层的服务的规范说明，称为**用户服务**。通常，它是以一组带有**服务参数**的**服务原语**的形式给出的。正如看到的，上一层的协议与远端系统中的**对等协议层**发

656 起传输信息时，要通过这些服务参数来进行。

协议规范说明文档包含以下几个部分：

1）要有一个精确的关于本层协议数据单元（PDU）的定义。层的协议实体与远端系统中的对等协议实体进行通信。

2）传输每一个PDU类型，本层协议使用服务的规范说明（由下一层实体提供服务）。

3）对于协议实体的操作，用一个形式化的描述方法进行精确定义。

图11-10概要地介绍了这些项。

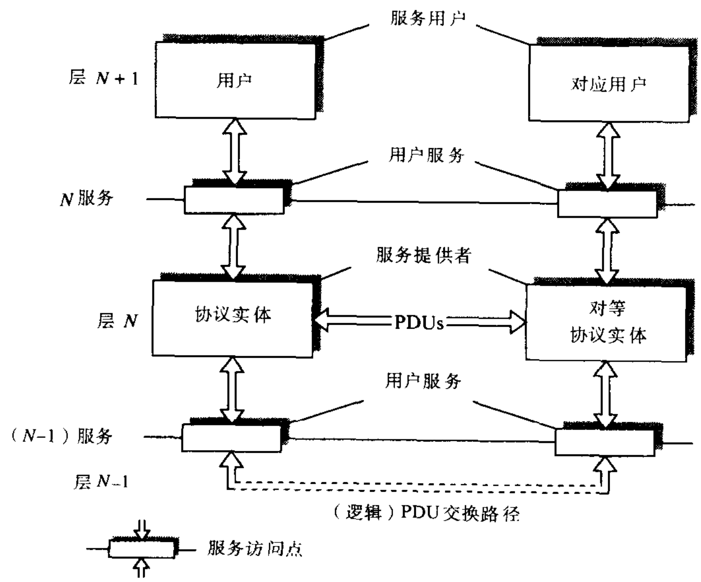


图11-10　协议层模型

11.4　服务定义

在任何网络系统中，都必须对用户应用进程标识与驻留此进程的网络地点进行区分。一个用户应用进程的标识通常以符号名称或称号出现，而网络中的地点通常是使用地址的形式。这和使用邮政系统发邮件时，使用名称和地址相类似：用名称来确定邮件的接收者，而地址则告诉收信人目前所在的地点。由于一台计算机上往往同时运行着许多应用进程，所以地址必须既包含计算机的物理地址又包含计算机内部的应用地址信息。这与公寓住宅中使用的楼

657 层及房间号是相类似的。

11.4.1　名称

在用户层，为了标识目的使用符号名称，对于每个用户应用进程（AP），它的符号名称在OSI环境中必须是惟一的。通常，一个应用进程并不知道网络中其他应用进程的实际的物理位置，所以一个AP与另一个AP用指定预期通信方名称或称号的方式进行通信。为了确保在特定的OSI环境中名称的惟一性，必须提供一些管理方法来管理应用进程（用户进程和服务提供进程）名称的分配。提供这一功能的系统称为**名字服务器**。它通常由管理环境的权威机构进行维护。

对于一个相对小的环境组成，例如一个由单个LAN连接起来的计算机系统群体，对于这

样的系统单一的名字服务器就足够了。然而，对于大的环境组成，例如，数千个系统通过若干个LAN和WAN互连起来，这时单一的名字服务器将变得不可管理。在这种情形下，对于每个子网络都需要一个独立的名字服务器。这时要在整个OSI环境中为每个子网络分配一个标识，同时为了确保每个名字在整个环境中是惟一的，要把子网络的标识加到用户名字的前面。

11.4.2 地址

虽然名字是在用户级使用的，但是OSI环境中使用地址是用来确定自身的。首先，是所要求应用进程所在的计算机当前的网络物理位置，其次，是应用进程所要连接的应用层协议实体的标识。而把一个用户应用进程指定的地点符号名称与特定的网络地址联系起来是OSI环境的职责。**系统目录**中包含了符号名称与地址之间的联系或映射的列表。理论上，一个实际的应用进程的物理位置可以通过改变它在系统地址目录中的入口来修改。

OSI环境中的地址包含了一系列的称为**服务访问点**（SAP）的子地址，从功能方面考虑，服务访问点也被称为**层间地址选择器**。应用进程所要连接的系统中的各个协议层之间的接口都要使用这样的地址。因此，一个应用进程的地址由如下形式组成：

$$\text{AP 地址} = \text{PSAP} + \text{SSAP} + \text{TSAP} + \text{NSAP}$$

其中PSAP是应用层协议实体（应用进程要连接的）与表示层之间的服务访问点子地址，SSAP是表示层协议实体与会话层之间的服务访问点子地址等等。NSAP包含了应用进程所在的系统的网络地址。系统使用P/SSAP和TSAP来确定用户应用进程要连接的是哪一个特定应用层协议实体。如图11-11所示。 658

一个应用进程地址或者称为**表示地址**，或者由于考虑到结构而称为**完全限定地址**。正如所见，PSAP和SSAP选择器可以有多种用途。在*A*中，PSAP和SSAP选择器并不用于多路复用/反多路复用，而是由TSAP地址选择器在系统中选择特定的应用实体/进程。这与TCP/IP协议族中的设计是相似的。然而，在*B*和*C*中通过PSAP和SSAP提供了更深层次的选择。

在*B*中，通过SSAP地址选择器提供了一种更深层次的选择，允许多个应用层和表示层实体共用同一个会话层连接。除此之外，在*C*中，PSAP允许多个应用层的实体和进程共用同一个表示层连接。将在第13章和第14章看到关于地址主题的更多细节。

如果一个应用实体包含有多个同时运行的事务的情况，例如，一个应用进程包含多个事务作为服务器（文件服务器），则它的实现依赖于系统内每个协议层要知道信息所涉及特定事务的标识。就是说，它不是SAP地址结构的一部分。为了允许这种方式，交换PDU都需要适当地与一个**连接（端点）标识符**或**实例号**联系起来。将在11.6节进一步讨论在传输层中的情况。 659

最后，图11-11中应用进程与应用实体是分开表示的，这只是为了标识OSI环境与实时系统环境之间的区分界线。实际上，将要在第13章看到，一个应用中包含的应用实体是与应用进程紧密相连的，由于这个原因，应用实体是**隶属**于应用进程的。

11.4.3 服务原语

一个层所能提供的服务是由一组服务原语确定的。一个层提供的服务可以分为两种类型：证实的和非证实的。图11-12阐述了两类之间的区别。

通常，用户通过层间接口发送一个**请求原语**来开始传输。这导致本地协议实体创建一个相关的PDU，然后使用下一层提供的服务把它传递给远端系统中的对应（对等）协议实体。远端系统中的对等协议实体接收到该PDU后，创建一个相关的**指示原语**并向上传递给对应用户。在非证实服务的情况下，传输就完成了。而当在证实服务的情况下，对应用户需要发送

一个**响应原语**。于是，接收方本地协议实体会创建一个相关的PDU，然后使用下一层提供的服务把它发送回开始的协议实体。开始的协议实体接收到这个PDU后，创建一个**证实原语**并向上传递给用户，这样传输就完成了。

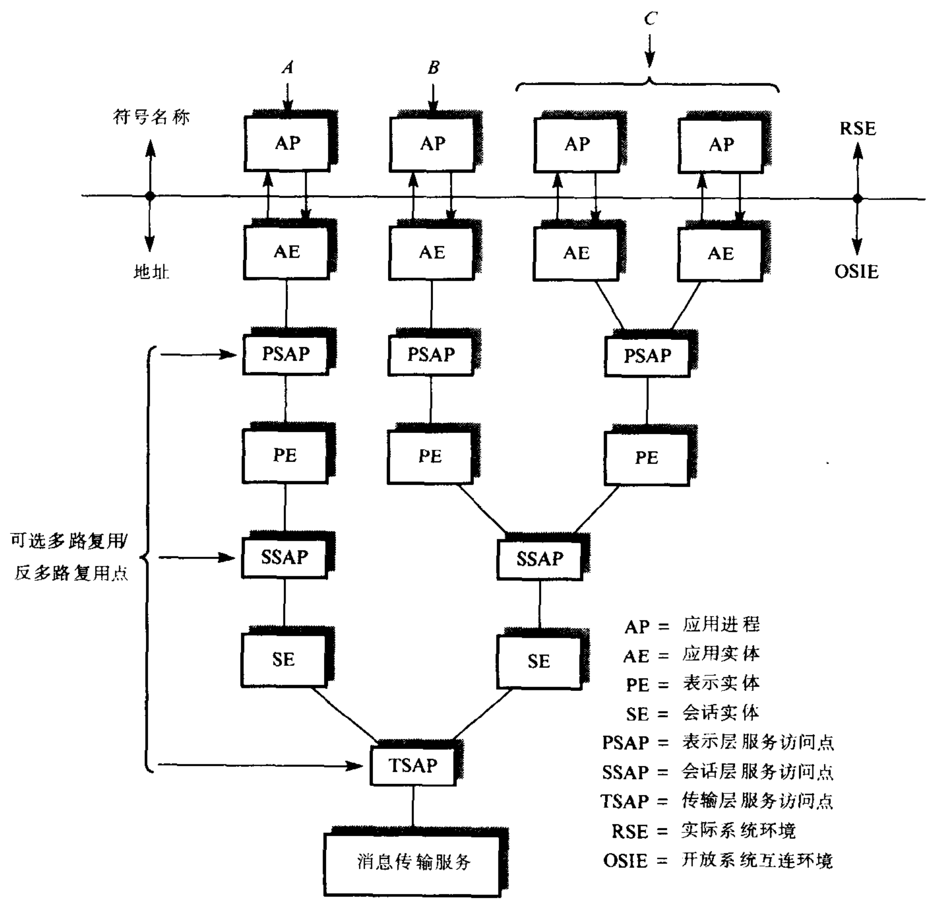


图11-11　PSAP/SSAP地址选择器的用法

不同的服务原语之间有一个逻辑关系，这些关系总是及时地联系起来。同一层内部的服务原语之间的内部关联通常用**时序图**的形式来表示。例如，图11-12(b)显示了刚刚讨论的四个原语之间的关系。显然，**时序图**只能以较抽象的方式表达一层内服务原语之间的关系，而不能显示出一个层是如何实现特定服务的。

通常，原语名称中包含了原语的类型和提供服务的层的标识。因而，

- T_CONNECT.request是一个请求原语，由传输服务用户（TS_user）发出。就是说，会话层使用它与一个远端TS_user（会话层）建立一个逻辑传输连接。
- S_DATA.indication是一个指示原语，由对等的会话层提供给上面的表示层。它关注从远端表示层接收到的数据的传送。

660

11.4.4　服务参数和层间交互

每个服务原语都有一组相关联的参数。通过这些参数在同一系统的邻近层之间可以传递信息，而且在不同系统的两个对等用户层之间也可以交换PDU。例如，前面的服务原语可能

包括如下参数：

T_CONNECT.request（被叫地址，主叫地址，……，用户数据）

S_DATA.indication（连接标识符，用户数据）

在第一个例子中，被叫地址和主叫地址参数指的是与已经建立的特定逻辑连接相关的相应SAP子地址的连接。通常，用户数据字段参数是一个指向内存缓冲存储器的地址指针，包含了以上的用户层协议实体所创建的PDU，这个PDU要被传递给远端系统中的用户层对应协议实体。然而，虽然使用了术语"用户数据"，但是并不一定要求用户应用进程创建数据。更确切地说，它指的传递数据是仅仅对上面的用户层有意义的数据。正如所见，它可能包含对应用户层双方交换的协议控制信息。 661

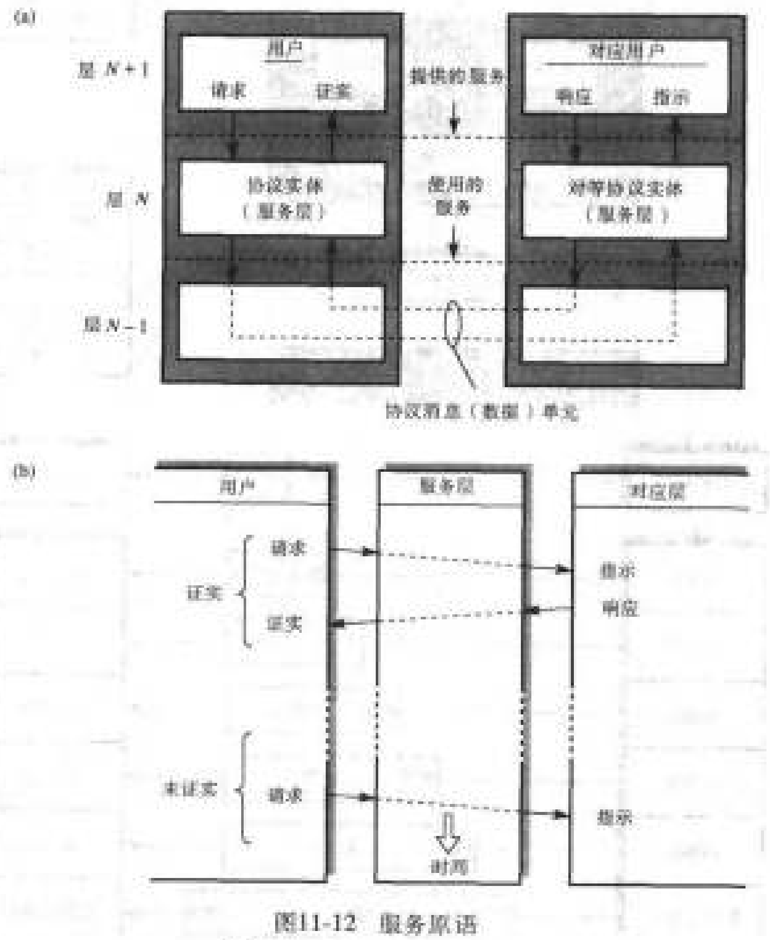


图11-12 服务原语

(a) 空间表示 (b) 时序表示

服务原语中的用户数据参数被接收层称为**服务数据单元**（SDU）。因为它包含了一个与上层有关的PDU，所以一个N+1层的PDU和一个N层的SDU是相同的。一般地说，当接收一个服务原语时，接收层的协议实体读取原语中的参数并把相关的参数组合成为一个附加的**协议控制信息**（PCI），从而形成本层的PDU。产生的PDU被封装在适当的原语中的用户数据字段并配以合适的附加参数，然后向下一层传递，如图11-13(a)所示。

从上面的讨论，可以得出如下结论：当用户数据向下穿越协议层时，每一层都会把自己层的

PCI加到用户数据上。这样，服务原语中的用户数据字段在穿过层间接口后，会有所增加。而且，当链路层协议实体添加完自身的PCI后，会对数据进行编码然后传递给远端系统。与之相反，当用户数据向上穿越远端系统中的协议层时，每一层的协议实体都要读取并解释相应的PCI，并把
662 它从用户数据中减掉，如图11-13(b)所示。讨论完每层的操作细节后，在第14章将再次讨论它。

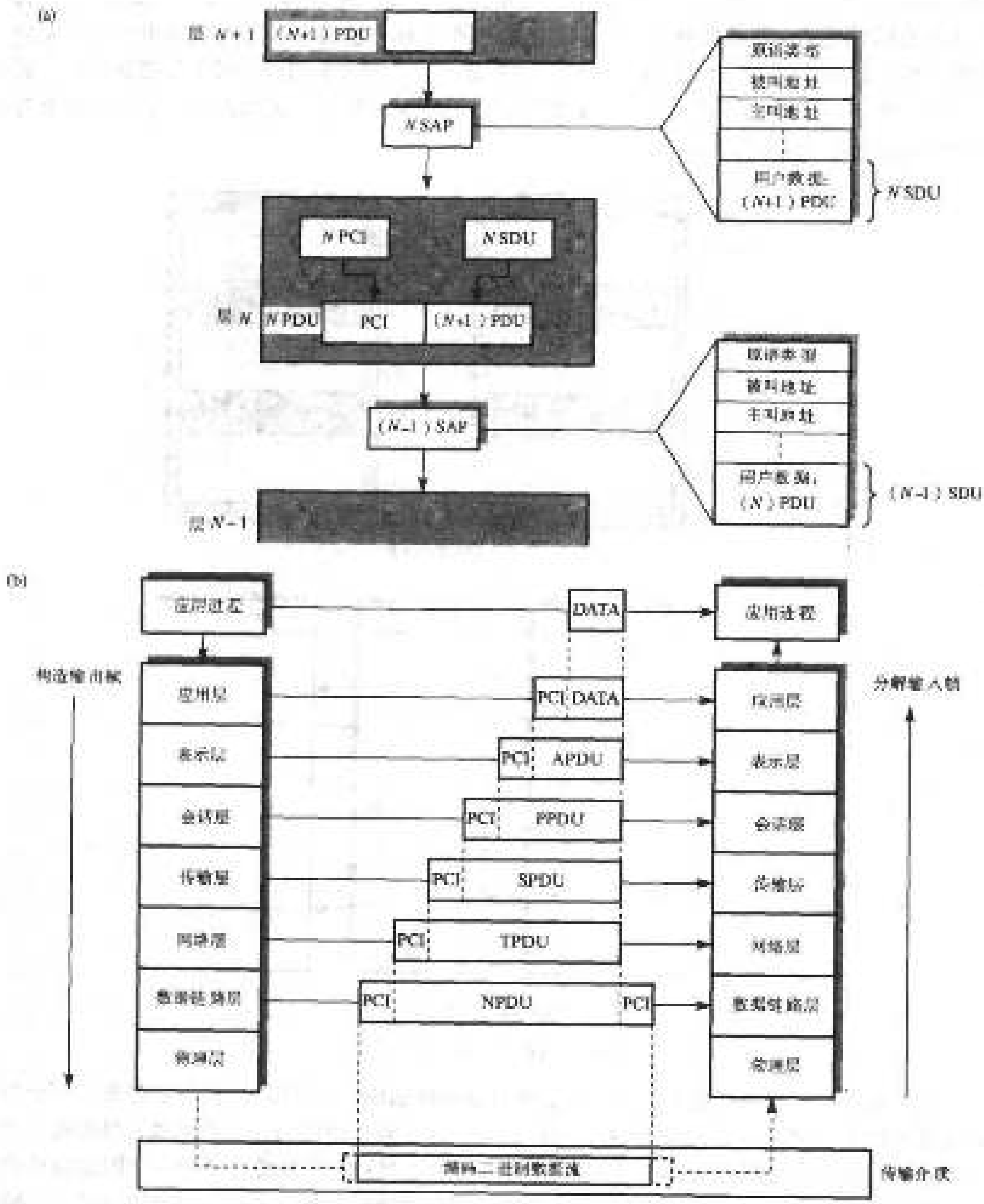


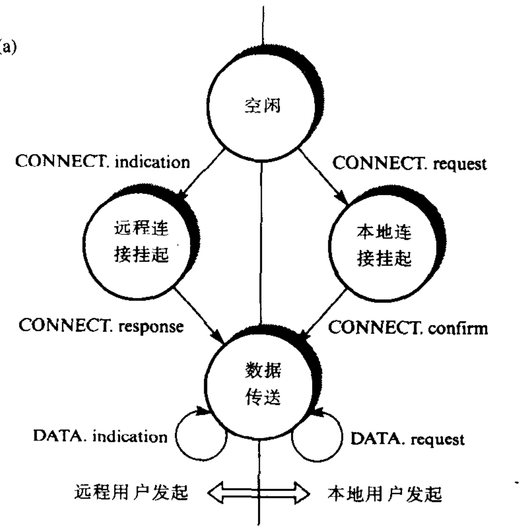
图11-13　层交互行为

(a) 单层　(b) 多层

11.4.5 原语顺序

每层都有一系列的原语，例如用来提供建立连接和传输数据的服务原语。当协议层的接口接收服务原语时，此层的协议实体必须进行原语顺序正确性的确认。例如，如果用户层在连接建立之前发送数据传输请求原语通常是不被允许的。因此，在标准文档里，一层中可接受的服务原语的接收顺序要以状态变迁图或**顺序表**的形式给出。图11-14显示了一个例子。

(a)



(b)

原语：→ / 可能随后的原语：↓	CONNECT.request	CONNECT.indication	CONNECT.response	CONNECT.confirm	DATA.request	DATA.indication	···
CONNECT.request							
CONNECT.indication							
CONNECT.response		+					
CONNECT.confirm	+						
DATA.request			+	+	+	+	
DATA.indication			+	+	+	+	
⋮							

+ = 允许　　空白 = 错误

图11-14 服务原语

(a) 状态变迁图 (b) 顺序表

在图中，支持用户服务首先允许用户与远端（对应）用户建立一个逻辑连接，然后通过这个连接传递数据，最后释放（清除）连接。为了更清楚地表述，图中只包含了连接原语和数据传递原语。通常，状态变迁图用来简单地表述用户接口所允许的正确的原语顺序。而顺序表则是个更精确的方法，它显示了所有可能的顺序，而不论此顺序是正确的还是错误的。因此，顺序表用来实现：确认接受到的原语是否遵循了正确的顺序。通常，接收到正确顺序 663
以外的原语意味着违反协议，这时相应的连接要被清除。

11.5 协议规范说明

一个层的协议规范说明文档包括如下：

- 关于协议实体中相关的PDU的类型及其目标的定性描述，每个PDU中包含的字段以及字段用途的描述。
- 关于协议实体在不同操作阶段所遵循的规程的描述，以及传输每个PDU类型要使用的服务的描述。
- 每个PDU类型的结构的形式化定义。
- 以形式化的说明方式给出关于协议实体操作的严格定义。

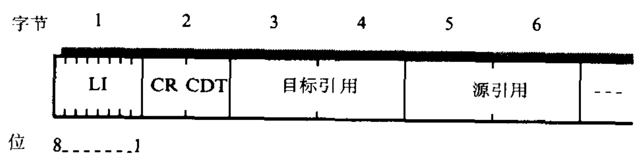
11.5.1 PDU定义

两个对等的协议实体通过彼此交换PDU来进行通信。通常PDU包含了用户数据和一个协

664 议实体自身创建的PCI。因为服务原语的参数只有本地意义，所以通常用抽象数据类型进行定义，例如，整数型，布尔型。相反，协议实体创建的PDU要在不同的系统之间传递，为了避免不确定性，必须使用一种精确的方式来定义PDU，这样双方系统才能有共同的理解。

为了达到这点，协议实体使用的PDU要在规范说明文档中给出精确的定义。有两种定义形式：一种是使用特定的位串的形式定义，另一种是使用抽象的数据类型（称为抽象语法表示法或ASN.1）加上一组相关的**编码准则**的形式定义，图11-15给出了两种方法的例子。图11-15(a)显示了传输层协议实体使用的PDU，这是一个连接请求TPDU（传输协议数据单元）。如
665 11.6.3节将介绍的，作为T_CONNECT.request服务原语作用的结果，传输层协议实体创建一个PDU用来与远端的对等传输协议实体建立一个逻辑连接。图11-15(b)显示了应用协议实体FTAM相关的PDU，将在第13章详细讨论。

(a)



LI = 长度说明 = PDU头部的字节个数（不包括LI）

CR = PDU类型；连接请求 = 1110（位8–5）

CDT = 信用分配；初始信用 = 0000（位4–1）

目标引用 = 本次连接目标使用的连接终点标识符；初始时设置为0

源引用 = 本次连接源使用的连接终点标识符

(b)

```
FINITIALIZErequest ::= SEQUENCE {
        protocolId [0] INTEGER { isoFTAM (0) },
        versionNumber [1] IMPLICIT SEQUENCE {
                major INTEGER, minor INTEGER },
                -- initially { major 0, minor 0 }
        serviceType [2] INTEGER {
                reliable (0), user correctable (1) },
        serviceClass [3] INTEGER { transfer (0),
                access (1), management (2) },
        functionalUnits [4] BITSTRING {
                read (0),
                write (1),
                fileAccess (2),
                limitedFileManagement (3),
                enhancedFileManagement (4),
                grouping (5),
                recovery (6),
                restartDataTransfer (7) }
        attributeGroups [5] BITSTRING {
                storage (0),
                security (1) }
        rollbackAvailability [6] BOOLEAN DEFAULT FALSE,
        presentationContextName, [7] IMPLICIT ISO646String {"ISO8822"},
        identityOfInitiator [8] ISO646String OPTIONAL,
        currentAccount [9] ISO646String OPTIONAL,
        filestorePassword [10] OCTETSTRING OPTIONAL,
        checkpointWindow [11] INTEGER OPTIONAL}
```

图11-15 PDU定义实例

(a) 位串形式 (b) ASN.1形式

正如在图11-15(a)中看到的，位串形式的PDU是由一串字节组成的，其中每个字节的用

途和格式都有详细的定义。虽然，所有低层网络相关层的规范说明中都使用这种形式的定义，但是大部分面向应用协议层的规范说明中都使用了ASN.1这种定义形式。本质上讲，ASN.1是以高级程序语言所使用的数据类型为基础建立的，如图11-15(b)所示。因而使用ASN.1定义的PDU是由一系列类型化的数据元素组成的。简单（原语）类型（INTEGER和BOOLEAN）和结构类型都可以被使用，结构类型（SET和SEQUENCE）与Pascal中的记录类型相似。

如名字的含义所指，ASN.1是一种抽象语法，这意味着虽然一个数据元素可以是一个规定类型（如INTEGER），但类型本身在位个数和位的使用顺序方面的语法都没有被指明。因此，我们必须使用一组编码规则来为一个用ASN.1定义的PDU创建本身的或具体的语法。于是，PDU就成为简单字符串，并且在每个系统上都用同样的（固定的）方式来解释。在第12章讨论应用支持协议时，将更全面地描述ASN.1。

11.5.2 协议操作概述

在第4章讨论链路级协议时，我们首次接触了协议实体的操作。把协议实体看作有限状态机或自动机模型。这意味着，在任何时刻协议实体只能处于众多的有限状态中的一个。自动机的当前操作状态和其他相关的协议状态信息是由自动机保留的一组状态变量来维持的。

当自动机的某个接口处产生了一个有效的入事件后，从一个状态到另一个状态的变迁就开始了。例如：

- 接收从上一层接口传来的服务原语。
- 接收从下一层接口传来的服务原语。
- 接收从**本地实体**的接口（例如**定时器**或**管理子层**）传来的服务原语。

如图11-16所示，通常出现一个有效的入事件导致协议实体创建一个PDU，然后在某个协议层接口处以出事件（动作）的形式向外发送出去。另外，自动机的状态改变可能发生，一个指定内部动作，例如定时器的启动，一个或多个自动机的状态变量也会相应改变。 666

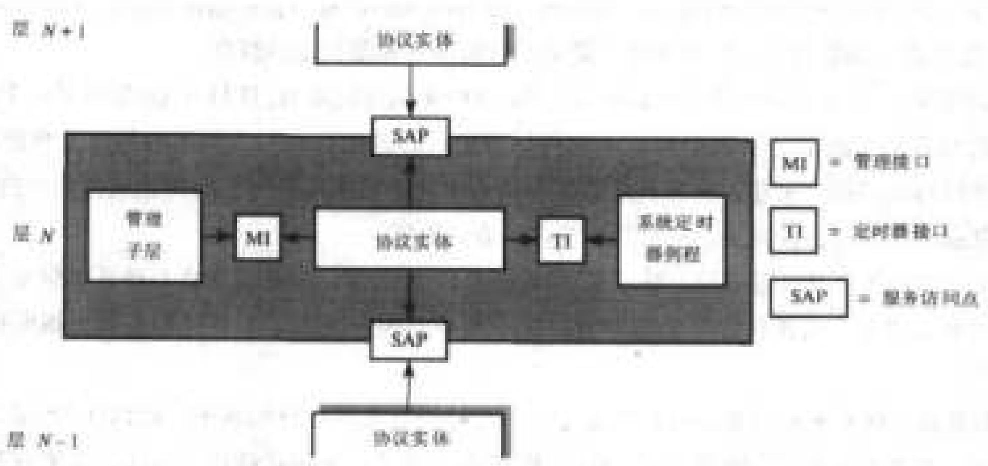


图11-16 入事件接口

所有的入事件都以原子方式操作，就是说一个协议实体（自动机）在完成与当前事件相关的所有操作（发送事件、特定动作、状态的改变）后，才能再处理另一个入事件。为了确保对于每个入事件的处理都是以原子的方式进行的，本地事件，例如超时的产生要跟其他入

事件以相同的方式传递给协议实体。通常，图11-16中显示的各个接口都要用一对队列（邮箱）形式来实现：一个用来向协议实体输入，另一个用来输出。然后，协议实体以指定的方式对队列服务。将在11.6.6节讨论更多的细节。

某些情况下，协议实体执行的操作以及它的新状态（如有）依赖于一个或多个**可能条件**或**谓词**的当前状态。谓词是一个布尔变量，它依赖于入事件相关的参数值与自动机状态变量的一个或多个当前值的组合。通常，谓词用符号P带一个数字来表示。例如：

P0：T_CONNECT.request是允许的

P2：重发计数＝最大值

关于它们使用的几个例子：

P5&（Not P1）：出事件*A*

新状态*X*

P0：出事件*B*

新状态*Y*

如果谓词条件不满足，同时没有定义另一个状态，就会产生一个协议错误条件，并且创建一个预先定义的出事件和一个新的自动状态。

667

11.5.3　协议规范说明方法

下面是ISO标准文档给出的协议实体的形式化规范说明：

- 每个接口所有可能发生的入事件的定义。
- 可能的自动机状态的定义。
- 协议实体创建的所有可能的出事件以及要执行的一系列特定动作表的定义。
- 自动机的操作的相关状态变量和谓词（可能条件）的定义。
- 扩充的事件—状态表的定义，包含所有可能的入事件和当前状态的组合，出事件（及特定动作）和新状态，以及依赖于谓词的任何事件和新状态的组合。

ISO使用的扩充事件—状态表的格式与第4章中使用过的类似。图11-17(a)显示了一个例子，表中的每项都指出了对于一个特定入事件与当前状态的组合会引发的出事件（及特定动作）和自动机的新状态。通常，只有有效的事件状态组合才被列入表中，而其他组合用空白填充。一个空白项意味着协议错误，它总是以规定的方式处理。

668

表中的每项都可以用两种方法定义，如图11-17(b)所示。项或者在自身单元中包含了实际出事件和新状态，或者包含一个参考号，这个数字指向一个关于出事件和新状态说明的列表的入口。

如果出事件和新的状态是由谓词确定的，这些会在表项中反映出来，图11-17(c)显示了一个例子。如果所有的谓词都不满足条件，则认为发生了一个协议错误。而且，如果在某项并没有相应的状态的变迁（就是说，自动机保持相同的状态），则只需指定一个出事件。类似的，如果某项含有相关的特定动作，给出动作表中有关动作的引用。图11-17(c)中的最后一个例子其特定动作[2]可以是终止相关定时器。

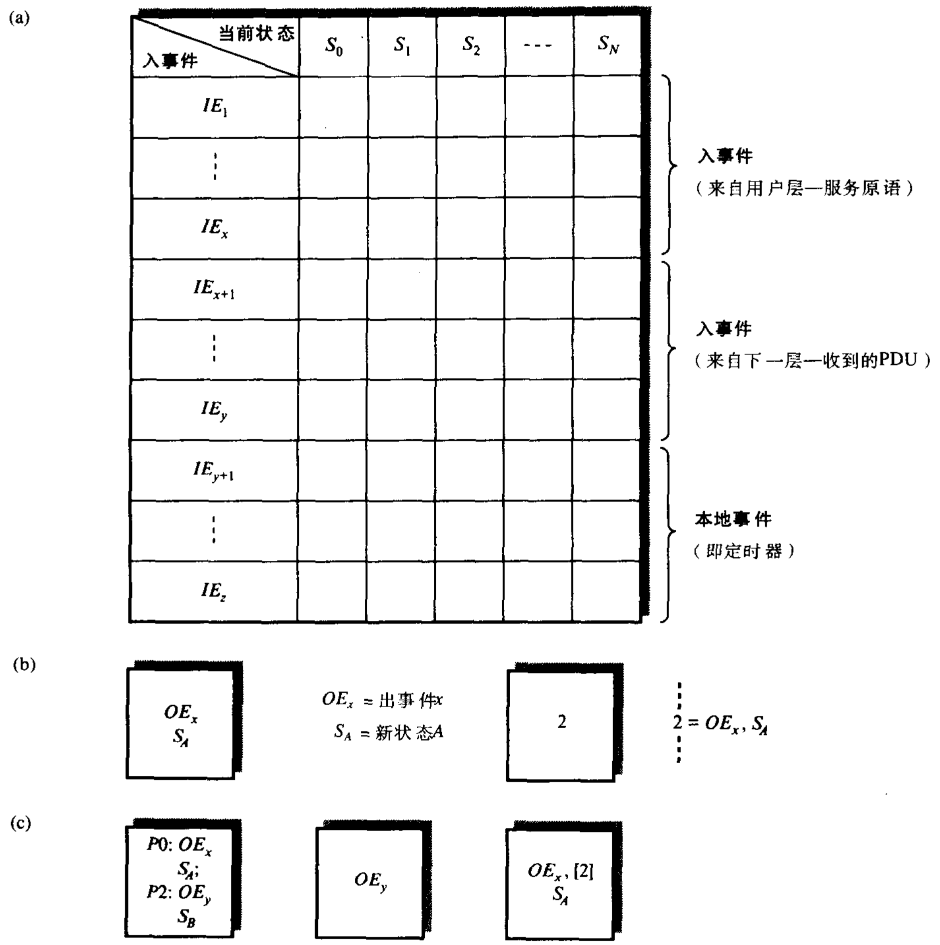


图11-17 扩充事件—状态表

(a) 表格形式 (b)～(c) 可选项格式

11.6 传输层

本节有两个目的：一个是给出关于前面几节讲过的规范方法的应用实例，另一个是介绍在ISO参考模型环境中传输层的操作和规范。

11.6.1 概述

图11-18显示了一个传输层模型，它与前面的方法是一致的。传输服务用户（TS_user）通过**传输服务访问点**（TSAP）使用一组定义好的用户服务原语与底层传输层实体（或服务提供者）进行通信。所使用的TSAP是与起始应用实体相关的。服务原语引起在**传输连接**（TC）上的两个对应（对等）传输实体之间交换**传输协议数据单元**（TPDU），或者，TPDU交换的结果产生服务原语。这些TPDU的交换是下面网络层通过**网络服务访问点**（NSAP）实现的。TSAP和NSAP地址共同惟一地确定连接中的应用实体（和连接AP）。 669

如前所述，传输层的功能是向会话层提供一个可靠的（无差错、顺序的、没有丢失和重复的）信息传递，并且本身不依赖下面网络层提供的QOS。为了适应不同种类的网络，向用户提供了五种级别的服务：

- 级别0：简单级（通常在提供高QOS的网络上使用，例如电报网络或PSDN网络）
- 级别1：基本差错恢复级
- 级别2：多路复用级
- 级别3：差错恢复和多路复用级
- 级别4：差错检测和恢复级。包含了最多的控制功能，例如差错检测、重传以及流量控制。它用在低传输质量的网络上，例如PSTN或WAN或使用无连接网络层的LAN。

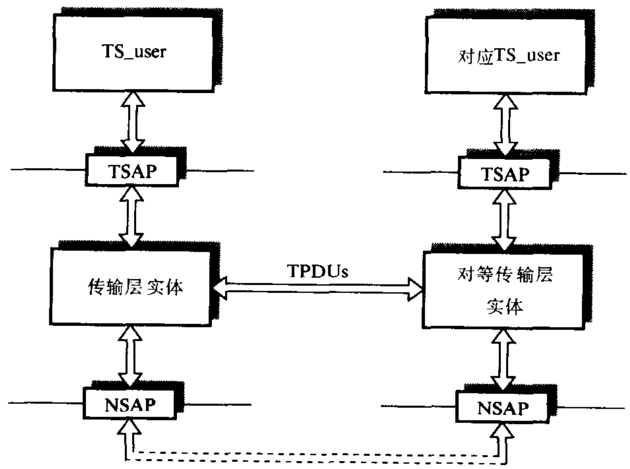


图11-18　传输层模型

以上所有级别的服务都假设操作是面向连接的模式，即两个通信传输实体在传输任何数据之前，必须在两者间建立一个逻辑TC。虽然这是大多数应用的推荐工作模式，但在TC连接的建立和释放阶段，通常不可避免地包含了一定层次的协议开销。对于某些应用环境，开销比较重要，我们期望使用更有效（但缺乏可靠性）的面向无连接的工作模式。在这种模式下，两个通信实体可以进行数据传输而不需要实际建立一个TC连接。

注意，传输层无需对所有级别都提供支持。通常，对于一个特定的OSIE，环境的控制机制指
670 定了要使用的服务的级别。所有的系统都期望所使用的级别能够最好地适合底层网络提供的QOS。

11.6.2　用户服务

可以把传输层提供的服务分为两类：面向连接和无连接。接下来，我们可以把面向连接的服务再分为两个子类：涉及连接管理的和涉及数据传输的服务。连接管理服务允许TS_user建立和维持一个与远端系统中TS_user的逻辑连接。数据传输服务允许用户在连接的两个对应用户之间交换数据。图11-19给出了传输层提供的一系列服务原语及其参数，而原语使用顺序的时序图如图11-20所示。

T_CONNECT服务中的主叫地址和被叫地址参数是TSAP和NSAP地址连结，用来指出使用本连接的主叫和被叫应用实体。QOS参数给出此TC所期望的某个特性，例如吞吐量和差错率。通常，这些是为特定的网络类型定义的。

在连接建立后附加的T_EXPEDITED_DATA服务只在数据传输阶段使用，通信双方TS_user首次建立TC时，会协商使用此服务的附带条件。此服务允许TS_user在级别4中的T_DATA服务使用的流量控制规程之外发送一个额外的数据项。在第12章，讨论应用支持层时会看到它的一个用法。

(a)

原　语	参　数
T_CONNECT.request .indication	主叫地址 被叫地址 加速数据选择 服务质量 TS_user数据
T_CONNECT.response .confirm	响应地址 服务质量 加速数据选择 TS_user数据
T_DATA.request .indication	TS_user数据
T_EXPEDITED_DATA.request .indication	TS_user数据
T_DISCONNECT.request	TS_user数据
T_DISCONNECT.indication	断开原因 TS_user数据

(b)

原　语	参　数
T_UNITDATA.request .indication	主叫地址 被叫地址 服务质量 TS_user数据

图11-19　用户服务原语和相关参数

(a) 面向连接　(b) 无连接

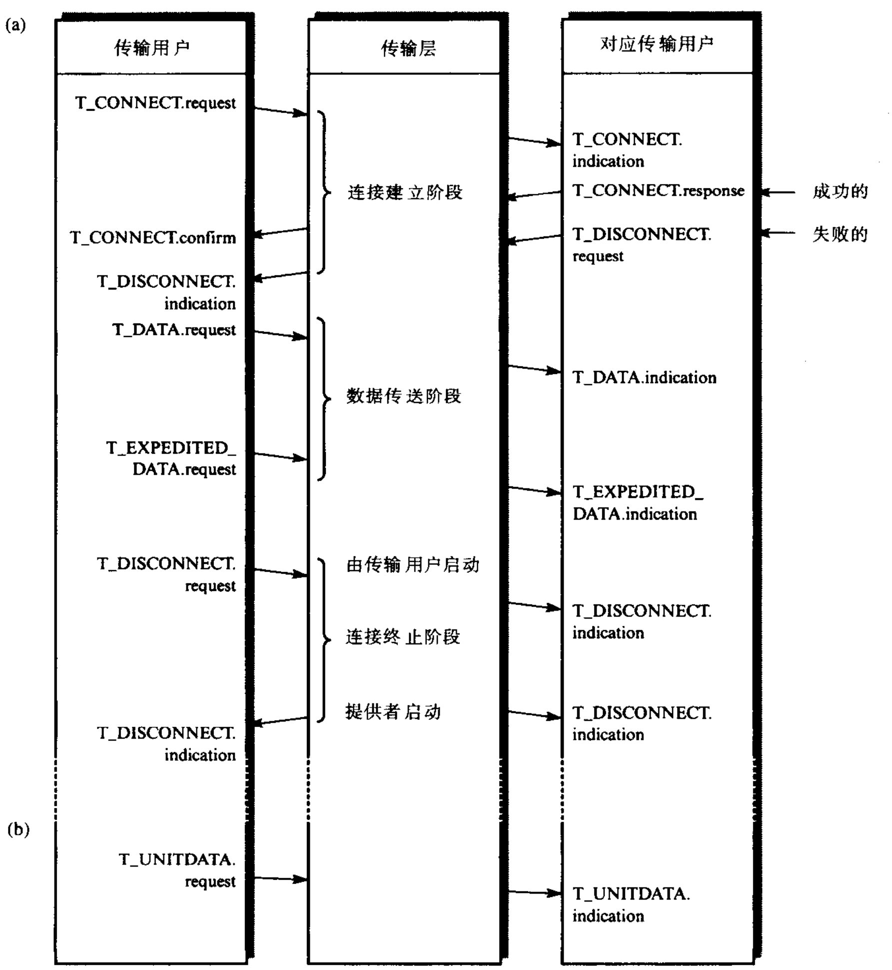


图11-20　用户服务的时序图

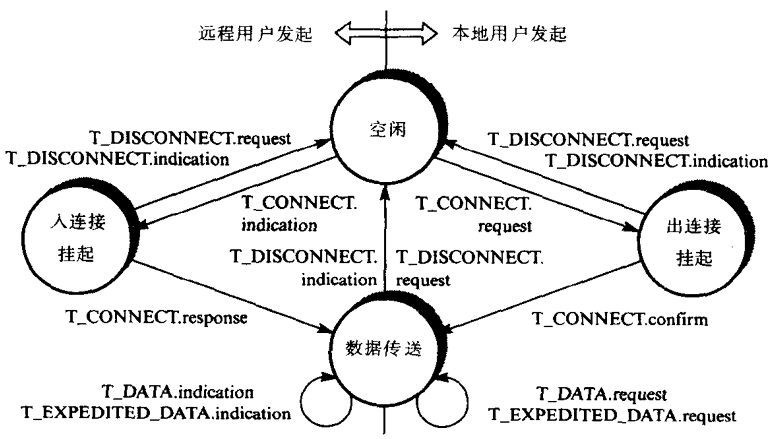
(a) 面向连接　(b) 无连接

671 图11-20中所示的原语顺序假设一个成功的连接建立过程。然而，如果对应的TS_user不接受连接请求，就要发送一条T_DISCONNECT.request原语而不是T_CONNECT.response原语，并把拒绝原因作为参数返回。一旦TC建立后，任何时刻两个用户中任一个都可以通过向用户接口发送一个带有原因参数的T_DISCONNECT.request原语来发起释放连接的请求。另外，如果底层网络连接变为不可连，传输层使用T_DISCONNECT.indication原语来发起释放TC。

无连接工作模式有两种原语，T_UNITDATA.request和T_UNITDATA.indication，使
672 TS_user无需先建立一个TC连接，就能发送用户数据。然而，此服务不保证传输成功，不测事件留给更高的面向应用层去恢复。不可避免的，面向连接服务的协议要比无连接服务的协议更复杂。因此在本章的其余部分，将讨论面向连接的操作模式。

673 图11-21显示了关于面向连接模式中的用户服务的状态变迁图和相关的顺序表。正如所见，顺序表对于每个用户可以接受的原语顺序给出了更完整的规范。

(a)



(b)

原语：→ / 可以随后的原语：↓	T_CONNECT.request	T_CONNECT.indication	T_CONNECT.response	T_CONNECT.confirm	T_DATA.request	T_DATA.indication	T_EXPEDITED_DATA.request	T_EXPEDITED_DATA.indication	T_DISCONNECT.request	T_DISCONNECT.indication
T_CONNECT.request										
T_CONNECT.indication										
T_CONNECT.response		+								
T_CONNECT.confirm	+									
T_DATA.request			+	+	+	+	+	+		
T_DATA.indication			+	+	+	+	+	+		
T_EXPEDITED_DATA.request			+	+	+	+	+	+		
T_EXPEDITED_DATA.indication			+	+	+	+	+	+		
T_DISCONNECT.request	+	+	+	+	+	+	+	+		
T_DISCONNECT.indication	+	+	+	+	+	+	+	+	+	

+ = 可以　　　空白 = 不允许

图11-21 用户服务

(a) 状态变迁图 (b) 顺序表

11.6.3 协议操作

当接收到一个有效的输入服务原语时（从TS_user或网络提供者），传输实体创建一个相关的TPDU。通常，假设输入原语来自TS_user，TPDU包含与原语有关的用户数据和传输层实体加上去的额外的协议控制信息。产生的TPDU使用下面网络层提供的服务传递给对应的传输实体。

与传输协议有关的TPDU包括：

- CR：连接请求
- CC：连接证实
- DR：请求断开
- DC：释放证实
- DT：数据
- AK：数据确认
- ED：加急数据
- EA：加急数据确认
- RJ：拒绝
- ER：差错

每个TPDU类型都有一些相关的字段，图11-22显示了每个字段的精确格式和含义。**LI（长度指示）**字段指示了头部的字节数，不包含LI字节。**级别**字段指定了连接使用的协议级别（0～4）。**选项**字段指明了是否使用常规（7位序列号和4位信用值）或扩展（31位序列号和16位信用值）序列号和信用（CDT）字段。

所示的字段组成每个PDU中被称为**固定头部**的部分。大多数的PDU都有一个**可变头部**部分和一个**用户数据**部分。可变部分也由一些字段组成，每个字段包含一个8位字段类型、一个8位字段长度和字段值。这些可变字段和使用的PDU如下所示：

- 主叫TSAP地址（CR和CC）
- 被叫TSAP地址（CR和CC）
- TPDU大小（CR和CC）
- 协议版本号（CR和CC）
- 用户安全参数（CR和CC）
- 校验和（级别4中所有的PDU，本节后面将讨论） 674
- 附加选项（CR和CC）
- 可接受的协议级别（CR和CC）
- 估计确认延迟时间（CR和CC）
- 吞吐量需求—八个值（CR和CC）
- 残留差错率（CR和CC）
- 连接优先级（CR和CC）
- 发送延迟请求（CR和CC）
- 网络重连请求（CR和CC，级别1和级别3） 675
- 用户定义的额外信息（DR）
- AK号（AK）

- 流量控制证实（AK）
- 无效TPDU（ER）

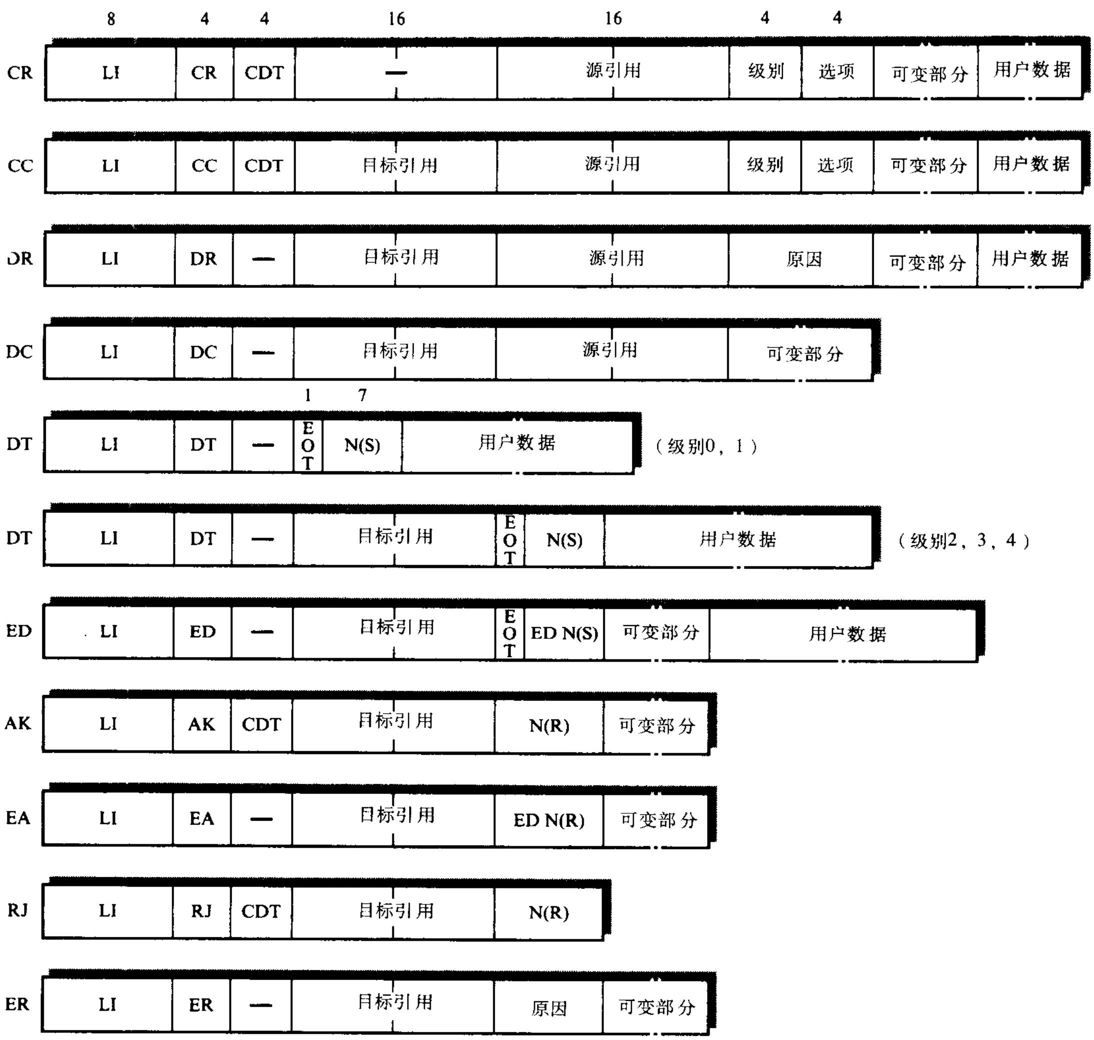


图11-22　TPDU类型和头部的字段

如上所示，大部分都使用CR和CC TPDU，用来协商传输连接的操作特性。大多数PDU中额外的用户数据部分允许一个用户通过这些PDU发送32字节的数据。

1. 建立连接

当TS_user发出一个T_CONNECT.request原语时，就开始建立一个TC。本地传输协议实体通过创建CR TPDU响应，然后把它传送给被叫系统的对等传输协议实体。当对等传输协议实体接收到后，将发送给本系统中的指定用户一个T_CONNECT.indication原语，通知有一个连接请求。假设对应用户准备好接受呼叫，会发送一个T_CONNECT.response原语作为应答。如果对应用户不打算接受连接，则发送一个T_DISCONNECT.request原语并附带拒绝原因作为参数。对等传输协议实体会以CC TPDU 或者DR TPDU产生一个适当的响应。最后，发起的传输协议实体转发给传输服务用户一个响应，利用T_CONNECT.confirm原语或者

T_DISCONNECT.indication原语，后者包含一个拒绝原因作为参数。要注意，使用的是双向交换，如果两个用户同时发起建立一个连接，将建立两个独立的连接。实际上，因为当前的ISO应用是建立在客户—服务器模式上的，这种情况不太可能发生。

CR和CC TPDU中包含的参数转发了所建立连接的相关信息，双方的传输协议实体通过这些参数和建立的连接来管理以后的数据传输。参数包含主叫（源）方与被叫（目标）方传输连接访问点（TSAP）标识符，要求的服务级别，后面的DT TPDU的最大长度等传输层实体进行操作所需的全部信息。通常，参数长度是由底层网络类型决定的，范围从128字节~8192字节（以2的幂数增加）。一旦传输连接建立，传输实体就可以在建立的逻辑连接上双向传输数据。

2. 数据传送

TS_user通过先前建立的连接使用T_DATA.request原语发起与对应用户的数据传送。本地的传输实体然后会将传输用户数据（TSDU）分成一个或几个DT TPDU进行传输，这依赖于TSDU中用户数据的数量和此连接允许的最大TPDU长度。每个DT TPDU包含一个标志（EOT），它指示此TPDU是否是TSDU中的最后一个。并且，每个DT TPDU包含一个发送序列号N(S)，用来提示此TPDU在序列中的次序，以及在AK TPDU中实现数据确认和流量控制。当目标传输实体接收并确认所有的DT TPDU后，会组成一个TSDU，然后发送一个T_DATA.indication原语，把重新装配后的数据块（TSDU）传送给对等传输服务用户。 676

不同的服务级别使用不同的确认和流量控制机制。通过使用底层数据网络的QOS传输TPDU，底层数据网络的QOS决定了使用什么样的服务级别。例如，对于X.25 PSPDN网络，传输TPDU的完整性和顺序是由网络层维护的。因此，传输协议只需要提供最小限度的确认和流量控制机制。然而，对于其他类型的网络就不再适用了，这时需要使用更加复杂的机制，这与第4章的讨论是类似的。

级别4中的确认规程是建立在回退N帧策略上的（见第4章），它的工作方式如下：当接收方接收到下一按序的DT TPDU时，如果TPDU完成了一个连续的TPDU序列，就会返回一个包含了接收序列号N(R)的AK TPDU，肯定地确认已经正确地接受到了发送序列号到N(R)–1为止的那些DT TPDU。当接收方接收到一个失序的DT TPDU（即收到的下一组程序DT TPDU中有一个N(S)大于所期望的值）时，接收方传输实体会返回一个RJ TPDU（否定确认），并在N(R)中提示它所期望接收的下一按序DT TPDU的N(S)。同时，双方都采用一个超时机制，来克服AK或RJ TPDU丢失的情况。

如果网络层不能确保DT TPDU总能够按序到达，接收方传输实体会使用包含在DT TPDU中的序列号按正确顺序进行数据重组。这种情况下，当只有一个DT TPDU是按序到达的或者一组连续的未解决TPDU序列完成后，接收方才会返回一个AK TPDU指示已经正确的接受。实现了用户数据的传递请求的TPDU序列如图11-23所示。

必须考虑的另外一个因素是网络层提供了一个低质量的服务的情况。这时网络可能会丢失TPDU，但并不通知发送方或接收方。这时可能传递的TPDU包含传输错误。因此，在连接建立时用户可能会指定一种服务级别，它产生一个超时和重传规程，允许丢失TPDU的发生，以及校验和机制与差错检测机制，并保证每个传输TPDU的完整性。

协议中实现超时和重传方案的工作如下：当传输实体发送一个要求响应的TPDU时启动一个定时器。如果定时器已经到时间了，但是还没有收到适当的响应，就会重传此TPDU并复位定时器后再启动。这样的循环重复了多次后，如果还没有收到适当的响应，传输实体就假设 677

对等的通信丢失，然后给用户发一个T_DISCONNECT.indication原语并附加上作为参数的中断连接原因。超时的使用意味着会造成重复传递，例如，TPDU接收正确但确认丢失。如果发现DT TPDU与之前接收的TPDU重复了，这由它本身的序列号确定，则返回一个AK TPDU，并把重复丢掉。

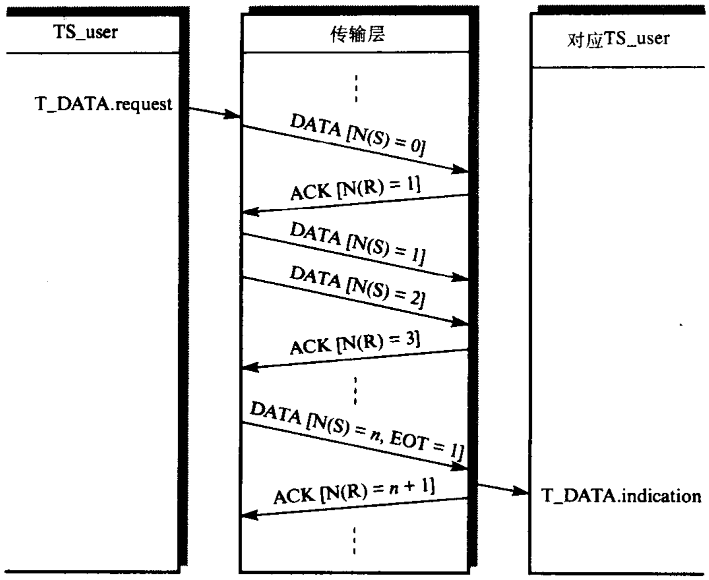


图11-23　数据传送实例

传输协议实体通过在每个TPDU生成一个16位校验和，并把它包含在头部中来实现数据的完整性。接收方使用相似的算法计算包括校验和参数的完整TPDU校验和，如果结果等于零，则此TPDU没有任何差错，否则就把TPDU丢掉。同时由于超时和重传机制，会确保有一个新的此TPDU副本被发送。

如果选择的是级别4，则CC和CR TPDU都使用校验和，其他TPDU也使用校验和。校验和是用来检查经过网络传输后，TPDU中存在的差错。标准文档中的算法是经过选择的，它对于每个TPDU使用最小的处理次数。算法计算两个校验和字节X和Y，如下：

678

$$X = -C1 + C0;\ Y = C1 - 2C0$$

其中，

$$C0 = \sum_{i=1}^{L} a_i \qquad （模 255）$$

$$C1 = \sum_{i=1}^{L} (L+1-i)a_i \qquad （模 255）$$

L 表示在整个TPDU中的字节数，a_i 表示TPDU中第 i 个字节。

图11-24显示了一个关于校验和的产生，并用它进行校验的过程的实例。TPDU中的内容被看作由一串无符号的8位整数和两个初始为零的校验和字节（X和Y）组成。两个校验和按以下方式计算：

1）初始化$C0$和$C1$为零。

2）从$i=1$到L，对每个字节按顺序处理。

3）在每一步中：

（a）令$C0$等于$C0$加上字节的值

（b）令$C1$等于$C1$加上$C0$

4）按下面方式计算X和Y：

$$X=-C1+C0\text{；}\ Y=C1-2C0$$

假设TPDU内容是：

$i=$	1	2	3	4	5
	5	9	6	X	Y

$L=5$

生成校验和：

$i=0$	$C0=C1=0$	$X=Y=0$
$i=1$	$C0:=0+5=5$	$C1:=0+5=5$
$i=2$	$C0:=5+9=14$	$C1:=5+14=19$
$i=3$	$C0:=14+6=20$	$C1:=19+20=39$
$i=4$	$C0:=20+0=20$	$C1:=39+20=59$
$i=5$	$C0:=20+0=20$	$C1:=59+20=79$

$X=-79+1\times 20=-59\ (196)$

$Y=+79-2\times 20=+39\ (39)$

检测校验和：

$i=0$	$C0=C1=0$	$X=-59\ (196)$ $Y=+39\ (39)$
$i=1$	$C0:=0+5=5$	$C1:=0+5=5$
$i=2$	$C0:=5+9=14$	$C1:=5+14=19$
$i=3$	$C0:=14+6=20$	$C1:=19+20=39$
$i=4$	$C0:=20-59=-39\ (216)$	$C1:=39-39=0\ (255)$
$i=5$	$C0:=-39+39=0\ (255)$	$C1:=0+0=0\ (255)$

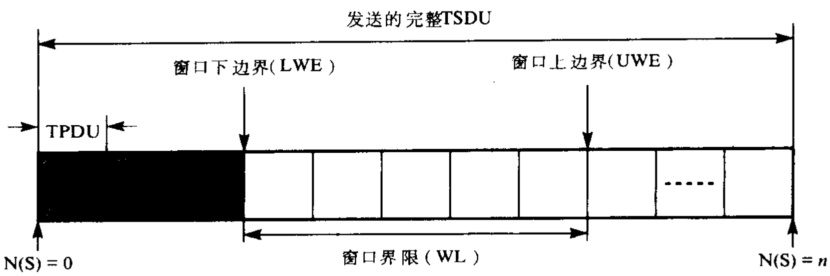
图11-24　传输协议校验和实例

注意，这种方法产生的$C1$与用（$L+1-i$）a_i求和方法产生的结果是一样的。一旦计算完，在传输之前两个校验和字节（X和Y）被插入TPDU。接收方在校验阶段使用相似顺序的步骤进行处理。这时，如果$C0$或$C1$中有一个为零，那么就认定没有错误发生。然而，如果$C0$和$C1$都不为零，就认定有错误发生，TPDU将被忽略。由于以上的原因，协议中要包括一个超时机制。 679

在计算$C0$和$C1$的过程中，采用了模255的运算，就是说，无符号运算（没有溢出并忽视进位），结果为0～255。为了计算两个校验和字节（X和Y），使用了反码运算，即使用了循环进位方法并且结果值255视为0。

流量控制机制的目的是限制发送方传输实体发送的数据（或DT TPDU）总量在接收方能适应的范围内。很明显，如果传输实体只为一个用户提供服务，适当数量的用来处理后续用户TSDU的缓冲存储器将被预先设计。从而，TC的建立可以无需流量控制机制的参与。然而，如果传输实体为多个用户提供服务，并且缓冲存储器是以统计基础设计的，这时协议必须支持流量控制机制。依然由传输实体提供的服务级别决定。

级别4中使用的流量控制机制是建立在**滑动窗口协议**的一个变型基础上的。在连接建立阶段，每个传输方向上交换的CC TPDU与CR TPDU中的CDT字段设置为一个初始的信用值（等于未确认的DT TPDU的个数）。每个方向上的传输的初始序列号在建立连接时设置为0，即**窗口下边界**（LWE）。发送方不断地计算**窗口上边界**（UWE），通过把连接的信用值模接收序列字段的大小加到LWE上。如果DT TPDU的N(S)等于UWE，则DT TPDU的流量被中断。当接收到未处理DT TPDU的AK TPDU时，LWE会持续递增，如图11-25所示。



注意：LWE初始设置为0，每当接收到AK TPDU时递增。当连接建立时，UWE被初始为协商的CDT值，随后通过每个接收到AK TPDU中的CDT值递增。如果N(S)达到UWE，则传输停止。

图11-25　流量控制机制

发送方可以发送的新DT TPDU的实际个数在连接期间是变化的，这是因为它完全由接收方控制。每个AK TPDU除接收序列号外都包含一个新的信用值，它指明了接收完当前的确认后，接收方准备接收的新的TPDU的个数。如果它的值为0，发送方必须停止此连接上的DT TPDU的发送。然而，通常当接收方为连接分配了一个固定数目的缓冲存储器时，使用信用值。当每个TPDU被接收后，接收方由于缓冲存储器被用光，准备接收的新的TPDU的个数（UWE）就会减少。

3. 连接终止

某方TS_user向本地传输实体发送一个以拆除连接原因作为参数的T_DISCONNECT.request
680 原语，这时一个连接终止（或释放）就开始了。对于级别0，TC的终止会导致相关的网络连接（NC）的终止，而对于其他级别，TC的终止是与NC终止无关的。当接收到T_DISCONNECT.request原语后，传输实体发送一个DR TPDU。并在收到DC TPDU前，丢弃所有收到的TPDU。对等的传输实体接收到DR TPDU，返回一个DC TPDU并发出一个T_DISCONNECT.indication原语给对应TS_user，这时，TC连接就关闭了。

11.6.4　网络服务

传输层，利用网络层提供的服务与远端系统中的对等传输层交换TPDU。网络层可以以面向连接或无连接方式操作。如在6.5.3节看到的，LAN通常用无连接网络层操作，而WAN通常用面向连接网络层操作。图11-26显示了每种方式提供的一组服务。如图11-26(b)所示，一个简单的（未证实的）服务原语（N_UNITDATA）提供用来以无连接方式传递所有的数据信息。

面向连接的服务需要额外的开销，图11-27显示了建立一个TC要使用的网络层原语：(a)面向连接的服务，(b)无连接的服务。如图11-27所示，在无连接方式中通过N_UNITDATA服务直接把CR TPDU传递给对方。相反，在面向连接方式中，必须首先建立一个网络层连接，然后，CR TPDU作为NS_user的数据进行传递。显然，无连接方式相关的QOS要比面向连接
681 方式相关的QOS低。

图11-28显示了传输层提供和使用的各种服务，并且可以在两个对应（对等）层之间传递的各种TPDU的类型列表。在讨论其他协议层时也可以使用相同的表达方式。

11.6.5　协议规范

如11.5.3节描述的，在ISO的标准文档中协议实体的形式化规范说明是用一个扩充的事
682～683 件—状态表形式来表述的。对所有可能的入事件和当前状态定义对应的出事件（包括特定动作）和新的状态。而且，如果包含谓词，对于所有可选的出事件和新状态都作了定义。

(a)

N_CONNECT.request
.indication
.response
.confirm

N_DATA.request
.indication

N_EXPEDITED_DATA.request
.indication

N_RESET.request
.indication
.response
.confirm

N_DISCONNECT.request
.indication

(b)

N_UNITDATA.request
.indication

N_FACILITY.request
.indication

N_REPORT.indication

图11-26 网络服务

(a) 面向连接 (b) 无连接

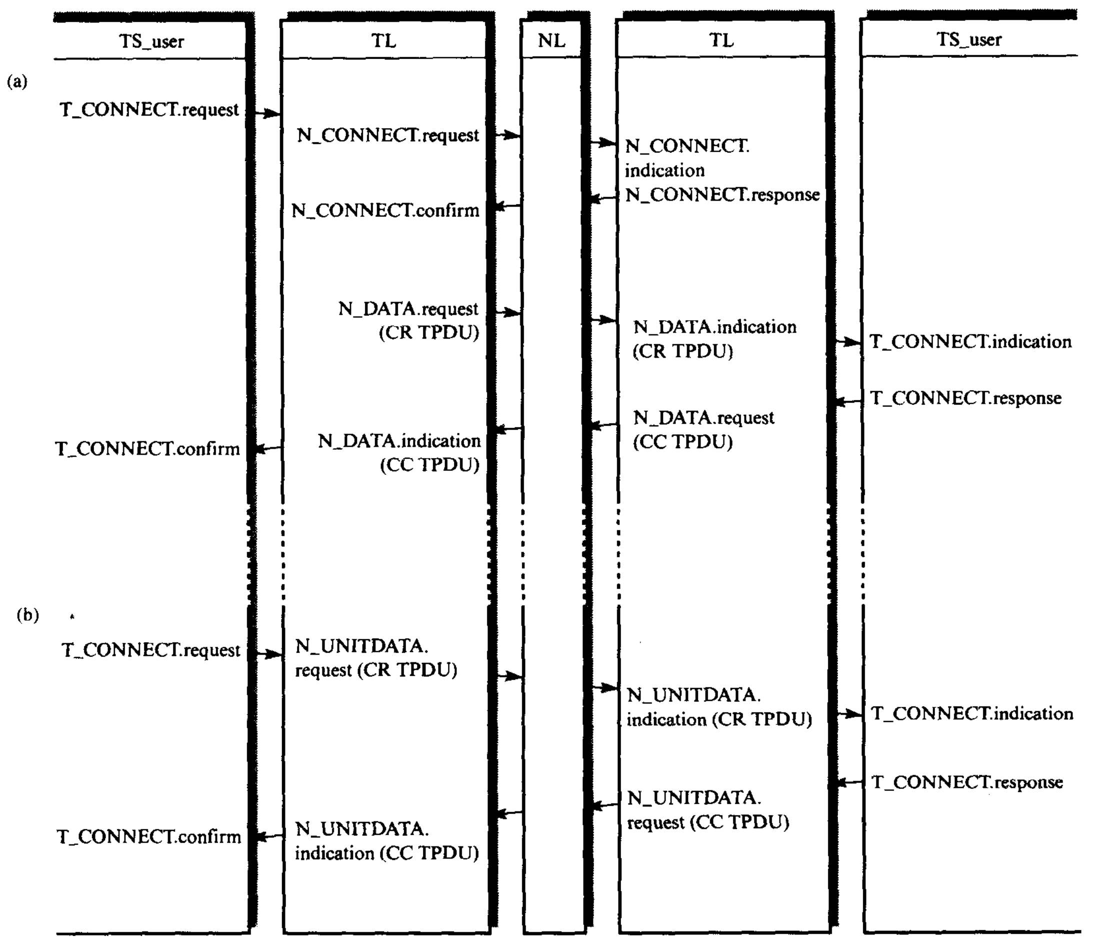


图11-27 连接建立阶段的网络服务

(a) 面向连接 (b) 无连接

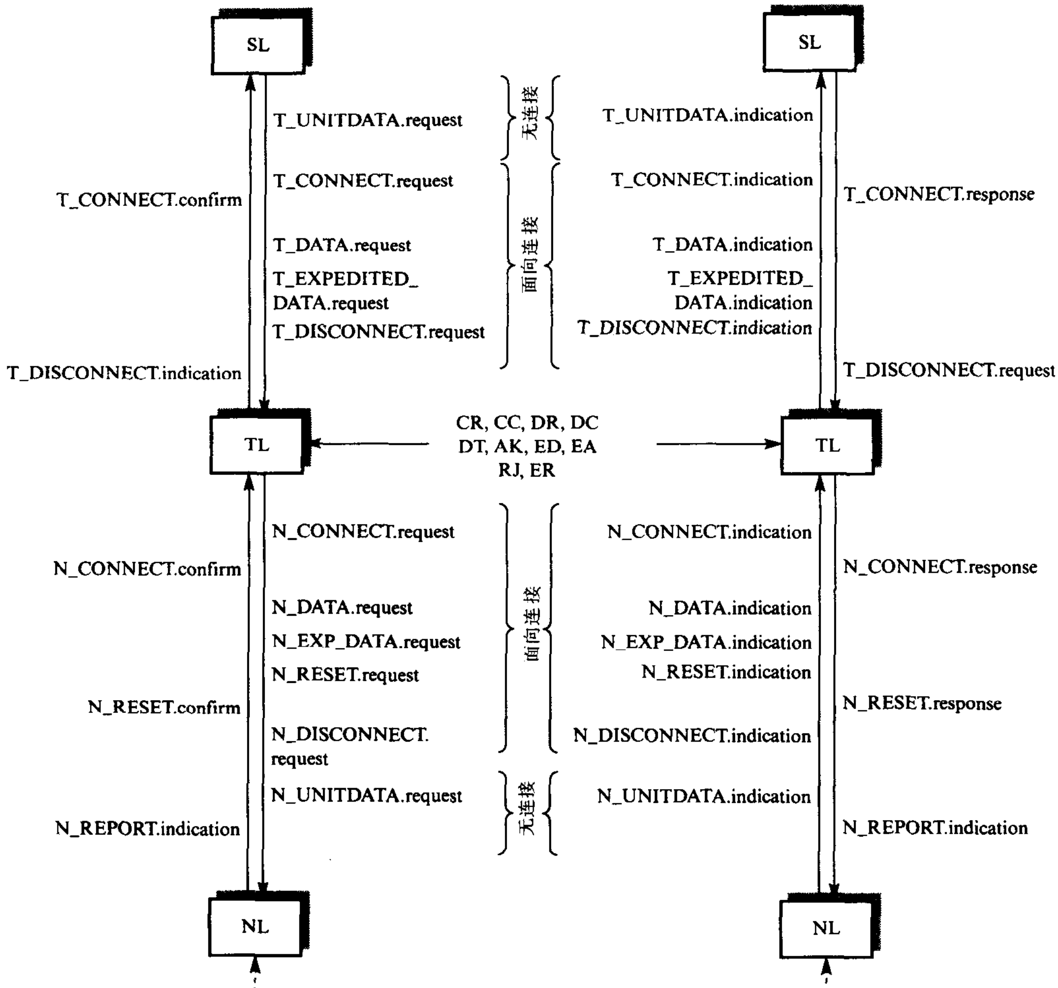


图11-28　传输层概要

通常，在协议的事件—状态表之前，要使用如下的名字：

- 入事件
- 自动机的状态
- 出事件
- 谓词
- 特定动作

为了阐述ISO使用的协议规范说明技术，图11-29给出了传输实体在连接建立阶段使用的名
684 称列表。对于不同的网络服务类型和不同的交换TPDU类型，传输不同TPDU类型所用网络服务原语也是有差异的。所以通常在不同的表中仅使用N_provider，代替特定的网络层服务原语。

图11-30显示了两种不同形式的关于连接建立阶段的事件—状态表。图11-30(a)表中每项都指定了实际的出事件和新的状态，而在图11-30(b)表中的项只是一个包含事件—状态组合表的位置量。（a）和（b）中的空白都意味着错误的条件组合。如果项中谓词没有满足条件，也
685 构成差错。所有的差错条件都使用预定义的方式处理。

(a)

名　称	接　口	含　义
TCONreq	TS_user	接收到T_CONNECT.request
TCONresp	TS_user	接收到T_CONNECT.response
NCONconf	N_provider	接收到T_CONNECT.confirm
CR	N_provider	接收到连接请求TPDU
CC	N_provider	接收到连接证实TPDU
⋮		

(b)

名　称	含　义
CLOSED	传输连接释放
WFNC	等待网络连接
WFCC	等待CC TPDU
OPEN	传输连接已经打开，并准备传送数据
WFTRESP	等待来自TS_user 的T_CONNECT.response
⋮	

(c)

名　称	接　口	含　义
TCONind	TS_user	发送T_CONNECT.indication
TCONconf	TS_user	发送T_CONNECT.confirm
TDISind	TS_user	发送T_DISCONNECT.indication
NCONreq	N_provider	发送N_CONNECT.request
CR	N_provider	发送连接请求TPDU
CC	N_provider	发送连接证实TPDU
DR	N_provider	发送断开请求TPDU
NDISreq	N_provider	发送N_DISCONNECT.request
⋮		

(d)

名　称	含　义
P0	来自TS_user的T_CONNECT.request不可接受
P1	收到的CR TPDU不能接受
P2	无可用的网络连接
P3	网络连接可用并已打开
P4	网络连接开放在进行中
P5	收到的CC TPDU 不可接受
⋮	

图11-29　连接建立阶段传输层实体的缩写名称

(a) 入事件　(b) 自动状态　(c) 出事件　(d) 谓词

(a)

事件 \ 状态	CLOSED	WFTRESP	WFNC	WFCC	OPEN	---
TCONreq	P0: TDISind CLOSED; P2: NCONreq WFNC; P3: CR WFCC; P4: WFNC					
TCONresp		CC OPEN				
NCONconf			CR WFCC			
CR	P1: DR CLOSED; NOT P1: TCONind WFTRESP					
CC ⋮	DR CLOSED			NOT P5: TCONconf OPEN; P5: TDISind NDISreq CLOSED		

(b)

事件 \ 状态	CLOSED	WFTRESP	WFNC	WFCC	OPEN---
TCONreq	1	0	0	0	
TCONresp	0	2	0	0	
NCONconf	0	0	3	0	
CR	4	0	0	0	
CC	5	0	0	6	
⋮					

0 = TDISind, NDISreq, CLOSED
（差错条件）
1 = P0: TDISind, CLOSED;
P2: NCONreq, WFNC;
P3: CR, WFCC;
P4: WFNC
2 = CC, OPEN
3 = CR, WFCC
4 = P1: DR, CLOSED;
NOT P1: TCONind, WFTRESP
5 = DR, CLOSED
6 = NOT P5: TCONconf, OPEN;
P5: TDISind, NDISreq, CLOSED

图11-30 连接建立阶段的事件—状态表格式

11.6.6 协议的实现

在第4章首次讨论链路层协议时，介绍了一个实现协议实体基本的方法。现在，介绍该方法如何具体应用于其他层，包括传输层。

如早先强调的，当描述一个根据ISO参考模型构造的通信子系统的各种操作时，我们必须把每个协议层看作一个独立的实体。这意味着它要向上层提供一组服务操作，并且通过使用下层提供的服务传递自身生成的PDU给远端系统的对等层。同样地，当用软件的方法实现各种协议层时，必须保持同样的方法，否则采用层次结构带来的好处将丢失。

通常，一个通信子系统由一组任务（进程）模块实现，每个协议层一组，并有附加任务执行本地管理和定时功能。任务之间的通信使用图11-31中采用的**FIFO 队列或邮箱**来完成。在第14章将看到，一个通信子系统通常使用一个独立的处理子系统来实现，这是因为一个完整的通信子系统会造成过高的处理开销。处理子系统与本地（实时）**内核**一同管理任务间的通信，例如任务调度和中断处理（例如，与每个协议实体相关的定时器）。

如先前描述的，相邻协议层的通信由服务原语实现。每个服务原语及其参数按定义的

（本地）格式首先产生于称为**事件控制块**（ECB）的**存储缓冲器**中。不像PDU严格的结构（语法）定义，服务原语的相关参数多采用抽象数据表。当使用高级语言来实现协议实体（和任务）时，在任务之间传送的参数所用数据结构由该语言确定。 686

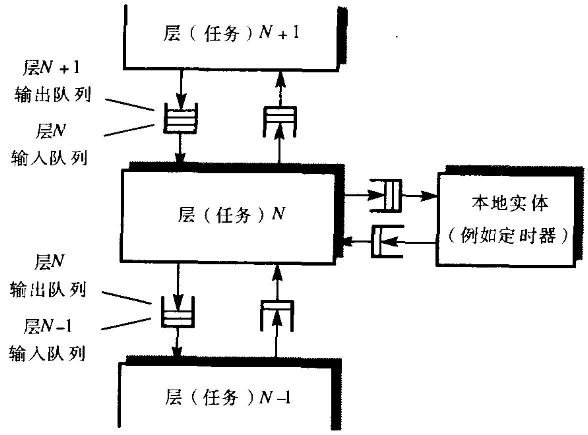


图11-31 任务间队列结构

通常，每个原语所拥有的参数的个数和类型都有所不同。为了避免对每个类型的原语都定义不同的数据结构，每个协议层通常只有单一类型的数据结构（ECB）。在ECB的头部中包含原语类型字段。后跟附加类型字段，用于本层中全体服务原语所有可能的参数。当在层间传递原语时，发送任务在ECB头部中提示发送的原语类型。接收任务由此决定使用特定的出事件程序。这个程序只需要读取与此相关的参数和赋予有效值。

图11-32(a)显示了一个ECB的结构。它是关于传输层的，用于会话层与传输层间所有的传输请求、指示、响应和证实服务原语。整个ECB是一个**记录**结构，服务原语类型为EventType。接下来将考虑UDBpointer和UDBlength的相关用法。主叫和被叫SSAP、TSAP和NSAP字段与T_CONNECT原语一同使用，而DestinationId和SourceId与随后的T_DATA原语一同使用，使该原语相关的用户数据和特定的传输连接（TC）联系起来。

服务原语在层间传递是由初始化任务（进程）实现的，该进程调用任务间通信原语，并将ECB地址指针作为其参数。随后内核会把这个指针加入到相应的层间队列尾部。当一个任务被调度运行，它检查每个人队列，包括上层或下层的队列以及定时和管理任务的队列，以此判断是否有ECB在等待处理。如果有，则传输层首先读取相应的队列头部指针，然后处理入事件。从而产生一个PDU和一个相应的格式化的信息（ECB），并把它传递给它的出队列。

上述的机制仅仅适合单应用层活动，即服务请求一次即被处理。然而，通常需要同时处理若干个不同活动（任务请求）。在这种情况下，层间信息或者通过单一队列组或者与每个主动SAP（信道）有关的独立队列组发送。对于前一种方法有如下问题，因为特定动作执行之前，与协议实体的状态相关的条件必须符合规范（例如，与流量控制机制有关的发送窗口关闭），某些时候必须暂停信息的处理直到中断条件消失。通常，后者仅仅影响单一的信道，因此其他的活动信道不受影响。 687

这样，机制的管理可能变得非常复杂（例如，当出现加急数据时）。为了避免这种情况，通常采取每个接入点都有一组相关的队列。那样，当一个信道暂时关闭，仅暂停受影响队列中项目的处理，直到中断条件清除。这样通过其他信道（队列）的报文不受影响。

(a)

```
const octet      = 0..255;
      maxSSAP = 2;
      maxTSAP = 2;
      maxNSAP = 11;

type SSAPaddrtype = array [1..maxSSAP] of octet;
     TSAPaddrtype = array [1..maxTSAP] of octet;
     NSAPaddrtype = array [1..maxNSAP] of octet;
     TransportECBtype =

     record EventType:      integer;
            UDBpointer:     ↑UDB;
            UDBlength:      integer;
            CallingSSAP:    SSAPaddrtype;
            CallingTSAP:    TSAPaddrtype;
            CallingNSAP:    NSAPaddrtype;
            CalledSSAP:     SSAPaddrtype;
            CalledTSAP:     TSAPaddrtype;
            CalledNSAP:     NSAPaddrtype;
            DestinationId:  integer;
            SourceId:       integer;
            QOS:            integer
     end;

var  TransportECB : TransportECBtype;
```

(b)

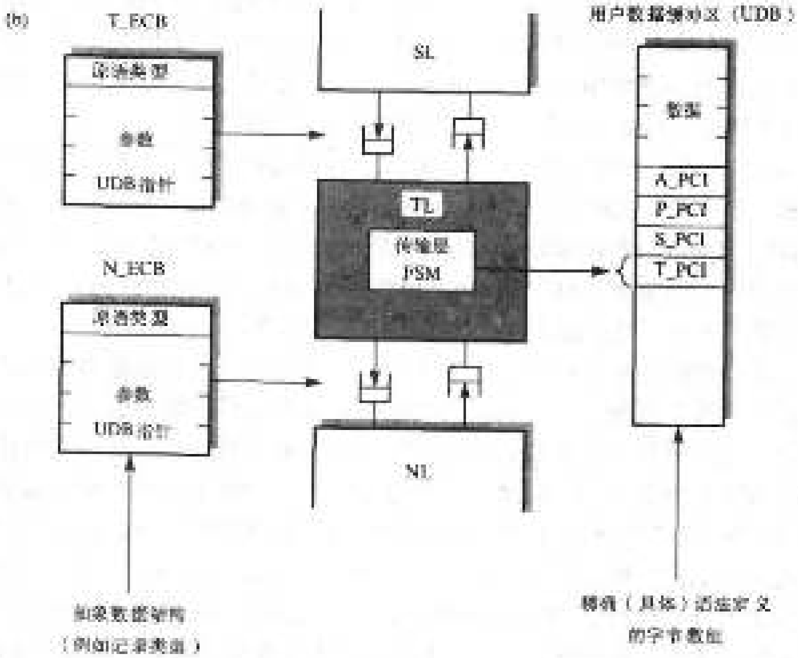


图11-32　层间通信示意图

(a) 传输ECB结构　(b) ECB与UDB之间的关系

如在11.4.4节描述，每个服务原语通常含有相关的用户数据，它与高层的PCI拼接在一起。接收到服务原语（ECB）后，协议实体使用原语的参数以及当前连接的协议状态信息，产生本层的PCI，并加到与输入服务原语相关的用户数据上，以形成本层的PDU，然后传递到与适当原语相关的用户数据字段的下一个低层。一般结构如图11-13(b)所示。

原语相关的用户数据存放在独立的**用户数据缓冲区**（UDB）中。可以推断出UDB中含有各个高层的累积（PDU）。因此，UDB的内容最终由物理层来传递。如11.5.1节提示的，每层的PCI必须以一种严格或具体的语法定义，以便不同的系统以相同的方式解释PDU。不管PDU的定义方式，它采用字节串的形式，因此每个UDB都简单声明为一个字节数组。在每个ECB的UDB中，指针字段指向与原语相关的UDB的地址。UDB长度表示缓存区中字节的个数。每经过一层就增加一层的PCI，直到物理层。然后按照UDB所存字节数全部发送。

整个通信软件包中，单个协议层的结构轮廓如图11-33所示。其中ECB和UDB是全局性的数据结构，以便各层访问。通常，在系统初始化时，建立ECB（对每一层）和UDB缓冲区，这些缓冲区的指针以自由表的形式连接起来。当需要新的缓冲区时，就从自由表中获得自由缓冲区指针。当一个缓冲区使用完毕后，就将指针放回自由表。

```
program Communications_Subsystem;
global  Intertask queues (mailboxes);
        Event Control Blocks;
        User Data Buffers;
            ⋮
  task Transport_Layer;
  local type Events = (TCONreq, TCONresp, NCONconf, CR,CC, ---);
             States = (CLOSED, WFNC, WFCC, OPEN, WFTRESP, ---);
            ⋮
     var EventStateTable = array [Events, States] of 0..N;
         PresentState: States;
         EventType: Events;
         ECB: ↑ ECB Buffer;
         UDB: ↑ UDB Buffer;
            ⋮
     procedure Initialize; {Initialize EventStateTable contents and state variables}
     procedure TCONind;  }
     procedure TCONconf; } List of outgoing event procedures
     procedure TDISind;  }
            ⋮
     function P0: boolean; } List of predicate functions
     function P1: boolean; }

     begin Initialize;
       repeat Wait for an ECB to arrive at an input interface queue;
              EventType := type of event in ECB;
              case EventStateTable [PresentState, EventType] of
                 0: begin TDISind, NDISreq, PresentState := CLOSED end;
                 1: begin if P0 then begin TDISind; PresentState := CLOSED end;
                    else if P2 then begin NCONreq; PresentState := WFNC end
                    else if P3 then begin CR; PresentState := WFCC end
                    else if P4 then PresentState := WFNC end
                    else begin TDISind; NDISreq; PresentState := CLOSED end
                    end;
                 2:
                 ⋮
       until Forever
     end.
            ⋮
```

图11-33 协议层的概要程序结构

尽管每个层（任务）都有一个EventStateTable数组，但如果该层可同时处理多个服务请求，则对应每个现用信道都有一组独立的状态变量。为了清晰起见，图中只显示了一组状态变量。图中的实例是传输层。因此，各种事件类型、自动机状态、出事件规程和谓词，如图11-30的事件—状态表所示。

688～689 通常任务调度是由本地实时内核完成的。如果一个任务处于空闲状态（等待入事件的发生），并且一个ECB指针传递给它的输入队列之一，内核自动调度该任务，使之运行。事件的类型（来自ECB）首先赋予EventType，并且与当前PresentState结合在一起，用以访问EventStateTable数组。表中各项定义了相应要调用的出事件规程和新的PresentState。如果包含了谓词，则定义了一个出事件和新状态组合表。通常，谓词涉及若干个不同条件，因而，它可以在每个事件处

690 理过程的合适点上；或者通过调用特定的计算谓词状态的写布尔函数设置位或复位。

习题

11.1 借助图说明TCP/IP协议族中的各协议，识别以下字段出现在哪一个协议层PDU中，并解释这些字段的功能。
- （a）IP 地址
- （b）协议端口地址

11.2 画出在LAN上传递的帧的草图，指明MAC、IP、TCP/UDP的相对位置和应用协议控制信息。

11.3 解释UDP的期望应用。借助图识别组成用户数据报头部的各个字段，并解释它们的作用。

11.4 解释术语“可靠的流服务”的含义，以及它与基本报文传输服务的不同。

11.5 借助时序图表示一般的用户服务原语交换，解释两个传输层用户是如何执行如下功能的：
- （a）客户端—服务器连接建立
- （b）常规的数据传输
- （c）加急数据传输
- （d）状态的协商
- （e）正常连接终止
- （f）异常终止

11.6 画出组成TCP段头部字段的草图，并解释各字段的功能。

11.7 使用描述性文本表示TCP之间一般的段交换过程，解释如下的TCP部分：
- （a）建立客户端—服务器连接
- （b）连接冲突

识别交换的段类型以及序列号值。

11.8 借助图显示TCP之间典型的段交换过程，解释TCP中的正常和加急数据传输方式。
假设最大段长度为W字节，窗口大小为$2W$，往返延迟为$3W$，作出一个传输$3W$字节的段交换情况。要包括序列号、确认序列号和每个段的窗口值。

11.9 借助图显示TCP之间一般的段交换过程，解释TCP的正常和异常连接终止。要包括对FIN、SEQ和ACK标记的使用描述。

11.10 利用图11-9中状态变迁图开发一个基于事件—状态表的TCP实体客户端和服务器端的规范说明。明确表示引发每次变迁和本地（特定）行为的人事件。

11.11 利用图解释ISO协议层中下列术语的含义：
- （a）服务用户

（b）用户服务
（c）服务提供商
（d）层协议实体
（e）使用的服务

11.12 开放系统互连ISO参考模型环境中，利用图简述：
（a）术语“网络环境”、“开放系统环境”和“实系统环境”的概念以及相互关系；
（b）用户名称与完全限定地址的不同；
（c）当用户报文传输前通过各个协议层向下传递时，它的开销如何增加，而在接收方向上传递时开销如何减少。

11.13 解释ISO参考模型的开放系统互连的目标，并概述各层的功能。 691

11.14 作出ISO参考模型的概要结构，并指出下列服务在何处提供：
（a）分布式信息服务
（b）语法无关的报文交换服务
（c）网络无关的报文交换服务

11.15 连同附加描述画出一个协议层的模型草图，并在适当地方解释这些术语的含义：
（a）层提供的服务
（b）层的服务访问点（SAP）
（c）对等协议实体间的PDU交换
（d）层使用的服务

11.16 给出ISO采用方法中使用的扩充事件—状态表的结构，该方法用来指明需特别强调每个行和列含义的协议实体。给出具有以下条件的表的例子：
（a）单一的出事件和新状态的组合；
（b）由谓词确定多个出事件和新状态的组合；
（c）出事件的特定动作。

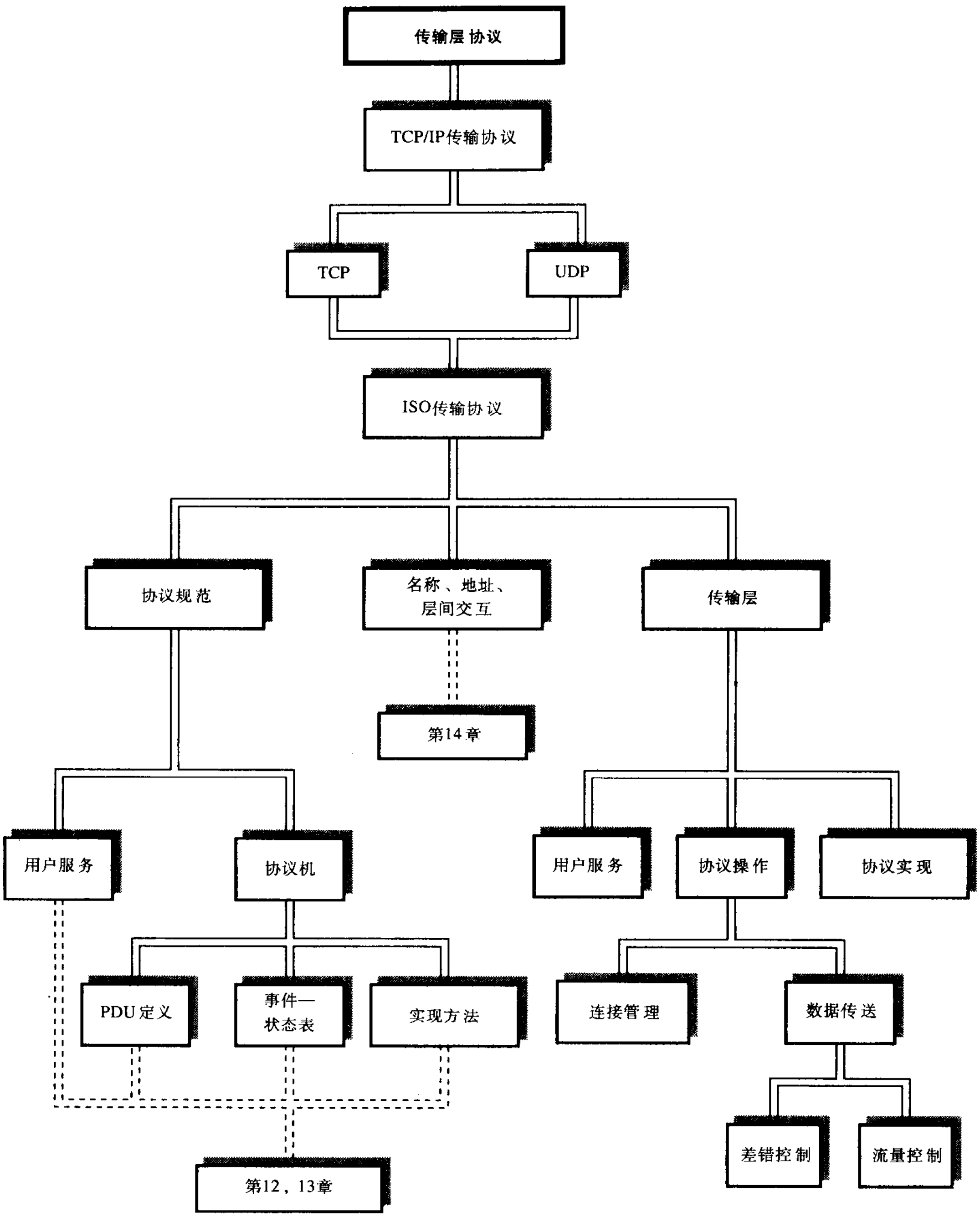
11.17 （a）用时序表说明传输层的一组用户服务，并描述它们的功能；
（b）定义一组TPDU实现上面的服务，并求出表示实现（a）中服务的TPDU交换顺的时序图；
（c）定义一组网络服务原语，并求出利用这些服务在（b）中定义的TPDU如何传递的时序图。

11.18 描述ISO传输层协议中数据传输阶段的如下过程：
（a）确认规程
（b）所用的差错检测方法，并举一个例子
（c）流量控制机制

11.19 ISO传输层协议中连接建立阶段的实现。
（a）生成入事件、自动机状态、出事件和与该阶段相关的谓词列表。
（b）为协议实体推导事件—状态表。对现在状态与每个入事件生成的可能的出事件-新状态。

11.20 概述实现一个ISO协议实体的方法。描述中要包括：
（a）使用的ECB的结构
（b）使用的UDB的结构 692

本章概要


传输层协议
TCP/IP传输协议
TCP
UDP
ISO传输协议
协议规范
名称、地址、层间交互
传输层
第14章
用户服务
协议机
用户服务
协议操作
协议实现
PDU定义
事件—状态表
实现方法
连接管理
数据传送
差错控制
流量控制
第12，13章

693

第12章 面向应用的协议

本章目的

读完本章，应该能够：

- 识别OSI协议栈中在应用层和传输层之间有若干中间面向应用协议；
- 了解这些协议的功能；
- 描述会话层的操作和服务，以及ISO标准中会话层协议实体是如何规定的；
- 了解表示层的各种功能；
- 了解抽象语法表示法1（ASN.1）的作用和操作；
- 了解所选择的数据加密标准的相关术语和准则；
- 描述表示层协议的服务和操作，以及在ISO标准中表示层协议实体是如何规定的；
- 理解联合控制服务要素（ACSE）和远程操作服务要素（ROSE）的作用；
- 在并发控制和多拷贝更新中会出现的问题，能够解释委托、并发和恢复（CCR）协议的各种服务和操作。

引言

回忆第11章，尽管TCP/IP协议族的应用协议（或应用进程）直接与传输层协议（UDP和TCP）交互操作，在OSI协议栈中它们通过与中间的会话层和表示层相关的协议实体交互。如图12-1所示，应用层由两组协议组成，每一个称为**应用服务要素**（ASE）。ASE是一个协议的 694
服务和协议规范的组合。其中一组执行特定的应用功能，而另一组执行通用的支持功能，称为**公用应用服务要素**（CASE）。然而在TCP/IP协议族中，CASE的职能以及表示层和会话层提供的职能被适当地嵌入到各个应用协议中。

OSI协议栈中包含会话层主要是为了使一个应用处理中差错造成的影响达到最小。在很多连网应用中，一个处理可能占用可观的时间并涉及大量数据的传输。一个例子是包含一组用户账号信息或雇员详细资料的数据库，数据库信息从服务器应用进程传送到客户进程。显然，如果在传递快结束时网络出错，会导致整个传递或者多个这样的传递不得不重新开始。会话层提供的服务就是为了降低这种出错的影响。 695

如果要求编写一个应用程序，它要处理与开放系统应用相关的一组文件中的记录，该应用涉及属于不同银行或金融机构上的多个计算机。这时一定想使用一个合适的高级编程语言来开发。每个记录都声明为记录结构，记录中包含各种字段并声明其类型。然而，虽然使用的数据类型同创建文件的程序员使用的类型相同，但是当程序通过编译后，在两台机器上，实际的机器上的每个字段的表示方式可能会有很大的区别。例如，在一台机器上整数类型可能表示为16位的值，而在另一台机器上可能表示为32位。即使两台机器都使用16位来表示整数类型，符号位的位置或者构成整数的两个字节的顺序都有可能不一样。同样，不同机器上使用的字符类型也不同。例如，一台机器使用EBCDIC，而另一台使用ASCII/IA5。因此，不同类型的表示要使用**抽象语法**形式。

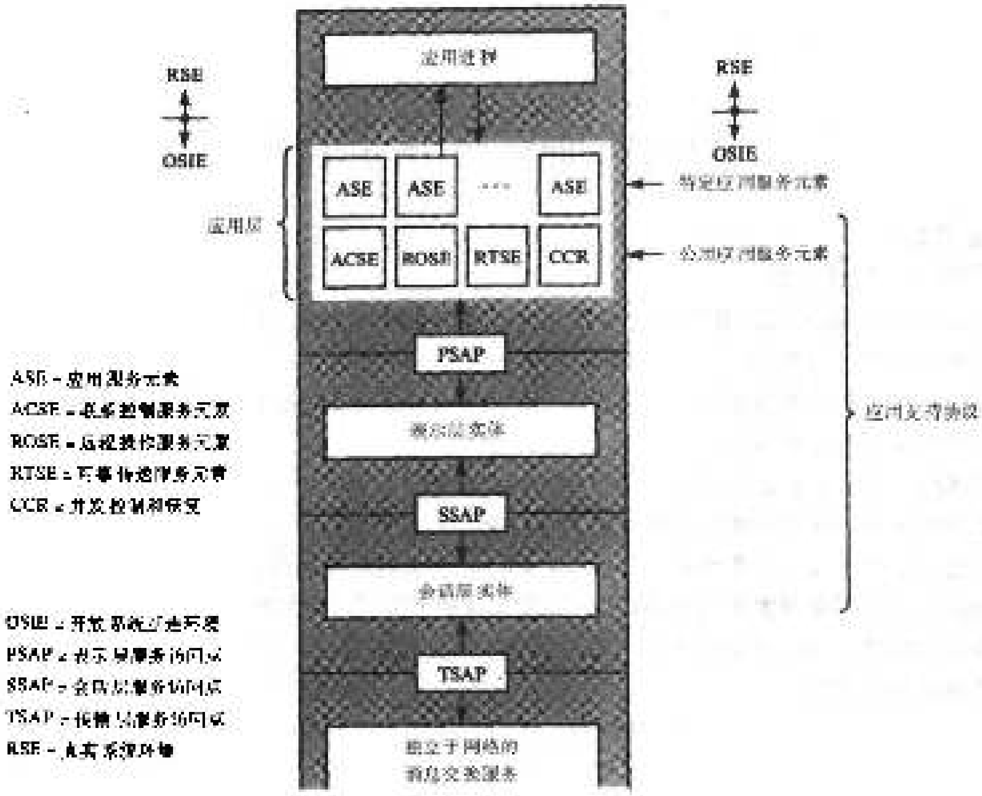


图12-1 ISO参考模型中的面向应用协议

当传输数据块时，例如从一台机器到另一台传输一组记录，不能仅仅传送数据块从一台机器的内存到另一台，因为接收方的计算机可能会以错误的字节边界解释数据。因此，当在两台机器之间传送数据时，必须确保接收方机器知道数据的语法。如果它本地语法不同，在处理前要把数据转换成用这种语法表示。

一种解决方法是为整个（分布式）应用定义一个数据字典，它包含了这个应用程序使用的所有数据类型的表示的应用程序范围的定义。如果这种表示与机器的本地表示方法不同，必须在处理之前，把所有接收数据转换成本地语法形式，如果数据发送给其他机器，再把数据转换成为标准形式。数据字典中使用的形式称为此应用的具体语法或传送语法。

在多数分布式应用中，这是共同要求，特别是在开放系统中不同厂家生产的计算机之间交互工作。ISO采用的方法是把这些功能整合到协议栈中，这是表示层的主要职能之一。表示层还包含数据压缩和数据加密功能，两者都关心传递数据的表示。

对于多数应用，许多其他功能是公用。ISO的方法是把它们从实际的专用应用功能独立出来，并把它们实现为一组支持协议。如果一个应用要求特殊的支持功能，那么用特定专用协议连接相应的支持协议的实例。如早先提到的，所有与应用层相关的协议都称为应用服务元素（ASE）。ASE执行通常的支持功能，包括联合控制服务要素（ACSE）、远程操作服务要素
696 （ROSE）、可靠传输服务要素（RTSE）和控制、并发和恢复（CCR）服务要素。

回忆第11章两个应用进程/协议可以用两种方法之一通信。在开始传递任何应用相关数据

之前，在两个进程之间建立一个逻辑连接（当它用于一个应用时称为**联合**）；或者一个应用进程直接发送信息，并等待响应。第一个方法适合于大数据量的传递，它可以确保在发送数据前接收方已经准备好。第二个方法更适合于简短的请求—响应交换。ACSE提供的服务适合第一个情况，而ROSE则适合第二个情况。另外，RTSE包括ACSE的服务加上选择的会话层服务。

很多应用需要CCR服务要素提供的功能。这些功能关注于一个共享资源的控制访问，例如，在一个航空预订系统中包含预定座位的文件，或银行系统中的客户账号文件。在这种应用中，文件必须允许从多个地方同时访问。因此，当这种文件的内容发生变化时必须采取适当的步骤以确保内容总是反映已经执行过的操作。与之相似，对于在不同地方有多个同一文件的有效拷贝的应用，要注意发生变化时确保多个拷贝的内容必须保持一致。CCR ASE就是实现这种功能的。

在本章将讨论所有提到的协议以及它们的功能和操作。

12.1 会话层

图12-2表示了在OSI协议族中会话层所处的位置。虽然会话层和表示层分别有特定的目的，但它们和各种应用层协议密切合作提供特定的应用支持功能。因此，即使PSAP和SSAP地址选择器都被用来提供传输层之上的附加的多路复用和反向多路复用功能，多数的应用层服务原语直接翻译成等价的会话层/表示层原语。实际上，很多实例中的表示层直接把选择的服务原语传递给会话层，而不加修改。另外，如果一个原语同时涉及表示层和会话层，则表示层实体只需把自己的协议控制信息加在它接收到的消息（PDU）的头部，并把它放在匹配的会话层服务原语的用户数据参数里传递给会话层。

回忆一下应用实体可以使用的面向连接和无连接通信方式。这两种方式可以由面向连接和无连接的表示层和会话层协议提供。然而实际上，到目前为止所有定义的OSI描述文件定义 697
都只应用了两种面向连接的协议。为了实现快速的响应，一些用户数据（512字节或更多）可以在初始连接PDU中发送。如果必要，也可以用一个返回的连接证实PDU，然后就释放连接。另外，一个表示层/会话层连接可以建立后保持连接状态，然后，使用这个连接根据需要执行数据的传递（双向）。将在12.1.2节看到，一个完整会话服务的子集称为**基本组合子集**（BCS），它通常作为最小化每个报文传递开销的方法。

下面将考虑两个面向连接协议的主要特性。由于两个无连接协议的功能最小化，使得协议比较简单。

会话层通过表示层提供服务，允许一个应用实体做如下事情：

- 同其他应用实体建立一个逻辑通信路径（会话连接），使用连接交换数据（对话单元），并按顺序释放会话连接。
- 在对话期间建立同步点，当出现差错时从商定的同步点上恢复对话。 698
- 中断（挂起）对话并从预定的同步点上恢复对话。
- 在会话期间，报告来自底层网络的异常情况。

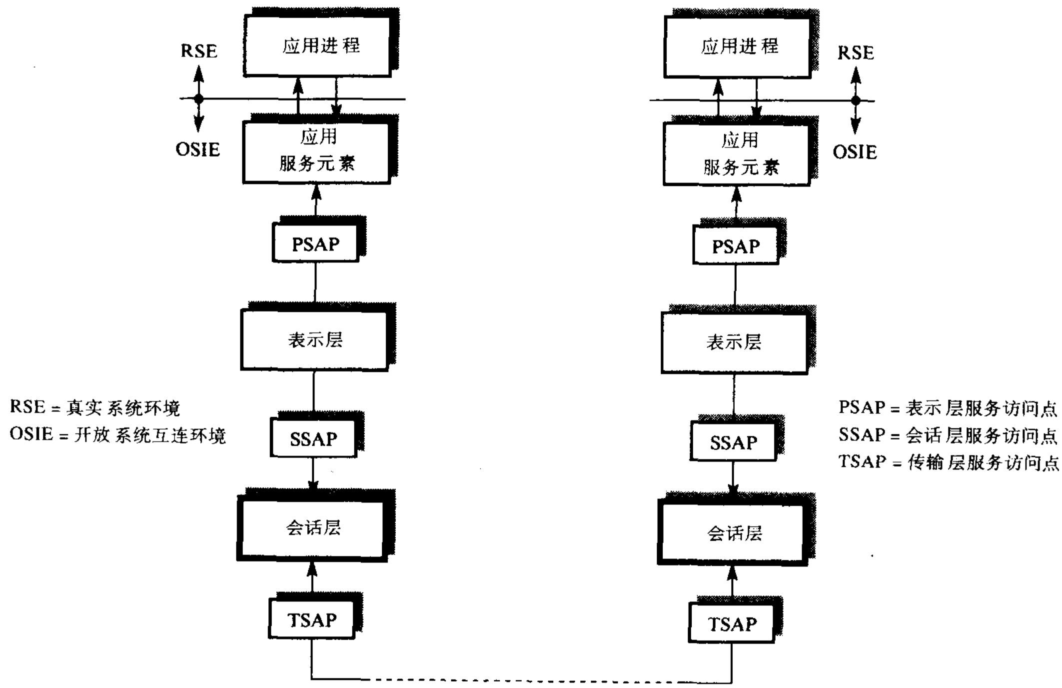


图12-2　会话层示意图

12.1.1　令牌概念

两个应用实体为了在已建立（会话）连接上管理对话，定义了如下一组令牌：

- 数据令牌
- 释放令牌
- 次同步令牌
- 主同步/活动令牌

每个令牌可在某一时刻动态地分配给一个会话服务（SS）用户（即通过相关表示层实体的应用实体），允许用户实施相关的服务。当前令牌的持有者允许用户独自使用相关的服务。例如，**数据令牌**用于使两个用户之间能进行半双工数据交换，**释放令牌**以控制的方式协商连接释放。

次同步和主同步/活动令牌在会话期间与同步进程联合使用。当两个会话服务用户交换大量数据时，把数据分成若干个可标识的单元。这样如果会话期间发生差错，只有局部数据受到影响。为了使用户实现这个功能，会话层用户在传递连续数据块前，插入一些**同步点**。每个同步点由会话协议实体保留的序号识别。提供两种类型的同步点：

- 主要同步点　通常用于两个用户之间交换的完整数据单元（对话单元）；
- 次要同步点　通常用于对话单元内部。

通过它们允许两个用户实现相应的同步过程。在整个会话连接期间同步点的建立如图12-3(a)所示。

活动的概念允许两个SS用户区分与会话相关的不同工作逻辑块。虽然一个完整的会话可能包含多个活动，但在一个时刻只可能有一个活动在处理。这样，一个活动可以中断，然后在同

一个会话连接中恢复，或者在不同连接上恢复。所以每个活动由许多**对话单元**组成。例如，一个活动可能涉及多个文件。一个对话单元涉及单个文件，而这个文件又由多个记录组成。 699

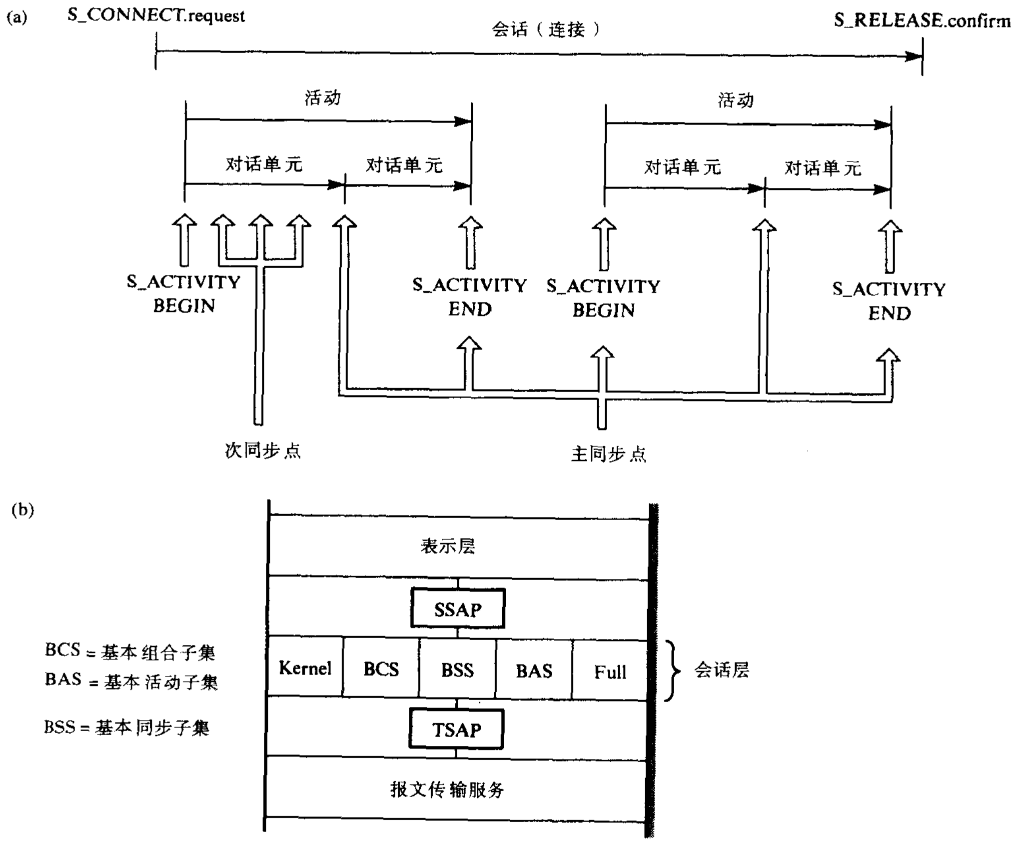


图12-3　会话层细节图

(a) 活动和对话单元同步概念　(b) 协议子集

12.1.2 用户服务

会话层可以向用户提供多种服务，为使两个SS用户在会话连接初次建立时，协商确切的服务，将服务分成若干个功能单元。下面列出了有效的功能单元：

- **核心功能单元**　提供基本（最小）的连接管理和全双工数据传送功能。
- **协商释放功能单元**　提供一个有序的释放服务。
- **半双工功能单元**　提供交替单向的数据传送。
- **同步功能单元**　提供会话连接期间的同步或重新同步功能。
- **活动管理功能单元**　提供识别、开始、结束、挂起和重新开始活动等管理功能。 700
- **异常报告功能单元**　提供会话连接期间的异常情况报告。

为了避免SS用户在会话连接初次建立时，必须指定每个需要的功能单元，定义了由不同单元组合而成的若干**子集**。这些在图12-3(b)中汇总并摘录如下：

- **基本组合子集**　包含核心和半双工单元。
- **基本同步子集**　包含同步单元。

• **基本活动子集** 包含活动管理和异常报告单元。

所有这些功能的执行都有相应的用户服务原语。例如，图12-4(a)的时序图显示了实现基本组合子集（BCS）的服务。

此外，每个服务都有相关的参数。例如，S_CONNECT原语的参数允许两个SS用户在会话连接期间，进行所用服务（即功能单元）的协商，包括初始的令牌拥有者和同步点序号的设定。此外，还有接下来的数据交换期间，使用的主叫和被叫地址以及连接标识符。两个TOKEN原语的参数包含令牌的类型，例如，数据，释放等等。S_TOKEN_PLEASE服务原语用于请求专用的令牌，而S_TOKEN_GIVE服务原语用于传递专用的令牌。

12.1.3 会话协议

前面提到的大多数服务原语导致会话层协议实体产生和发送一个相应的会话协议数据单元（SPDU）。例如，BSC服务原语的相关SPDU如下：

SPDU	响应的服务原语
CONNECT（CN）	S_CONNECT.request
ACCEPT（AC）	S_CONNECT.response（如果成功）
REFUSE（RF）	S_CONNECT.response
DATA（DT）	S_DATA.request
GIVE TOKEN（GT）	S_GIVE_TOKEN.request
GIVE TOKEN ACK（GTA）	接收的GT
PLEASE TOKEN（PT）	S_PLEASE_TOKEN.request
FINISH（FN）	S_RELEASE.request
DISCONNECT（DN）	S_RELEASE.response
ABORT（AB）	S_U_ABORT.request
701 ABORT ACCEPT（AA）	接收的AB

SPDU的一般格式如图12-4（b）所示。所有SPDU的前两个字段都是以字节为单位的**SPDU会话标识符**（SI）和**SPDU长度指示符**（LI）。不同的SPDU都有若干个与之相关的参数字段。每个参数都以标准的形式，由**参数标识符**（PI）、**参数长度指示符**（LI）和**参数值**（PV）组成。在有些情况下，参数组合到一起，这时参数字符串前面还包含一个**参数组标识符**（PGI）和一个**参数计数长度指示符**（LI）。

服务原语的用户数据划分成若干个SPDU后，再通过传输连接进行传送。然而，注意这是会话协议实体的功能，传输层不对此分段规程进行处理。

在发送SPDU之前，必须先建立一个传输连接。所有的SPDU，包括CONNECT SPDU，通过该连接使用T_DATA服务原语发送，如图12-4（c）所示。

为了阐述一些SPDU的用途，图12-5显示了建立连接，使用全双工和半双工传送数据，正常和异常释放连接的过程。

可以在一个CONNECT（CN）SPDU中包含最多512字节的发送数据。如果与S_CONNECT.request原语相关的用户数据超过512字节，源会话协议实体可以用第二个SPDU，称为CONNECT DATA OVERFLOW（CDO）发送额外的数据。同样的，如果与S_CONNECT.response原语相关的用户数据超过512字节，可以使用附加的ACCEPT DATA OVERFLOW（OA）SPDU立即发送。

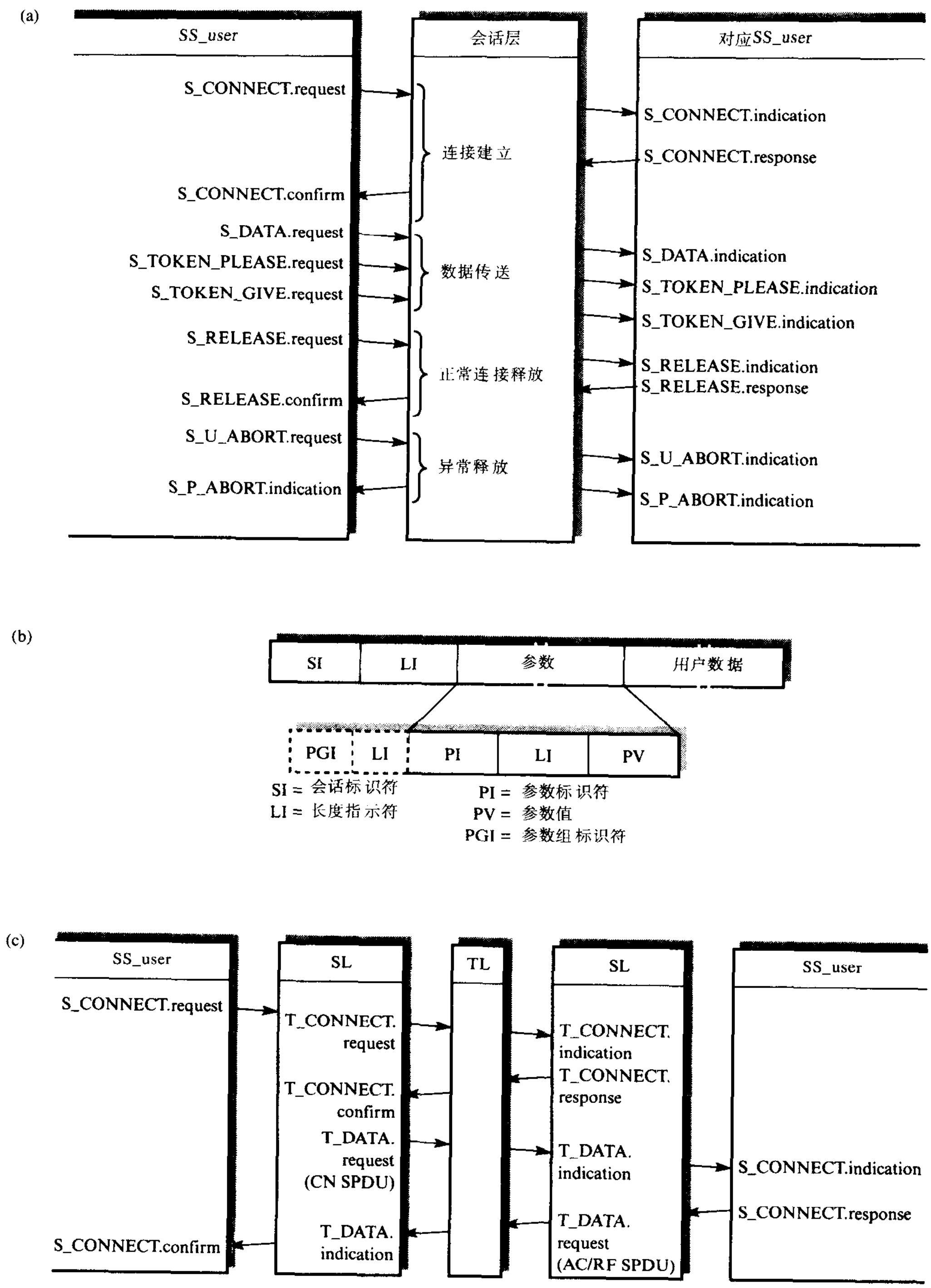


图12-4 会话服务

(a) 用户服务 (b) SPDU格式 (c) 使用的服务

使用双工的数据传送，此连接上的用户双方可以在任何时刻发送数据。但是对于半双工方式，只有当前的数据令牌拥有者可以传送数据。因此当响应方实体要返回数据时，它必须

在传送数据之前，向另一方请求令牌。在图12-5中，假设两个方向上都有两个DATA（DT）SPDU要发送。

最后，图12-5显示了两种释放方法：正常的和异常的。图12-6对以上提到的不同SS用户服务，连同交换的SPDU和BCS使用的服务作了一个总结。

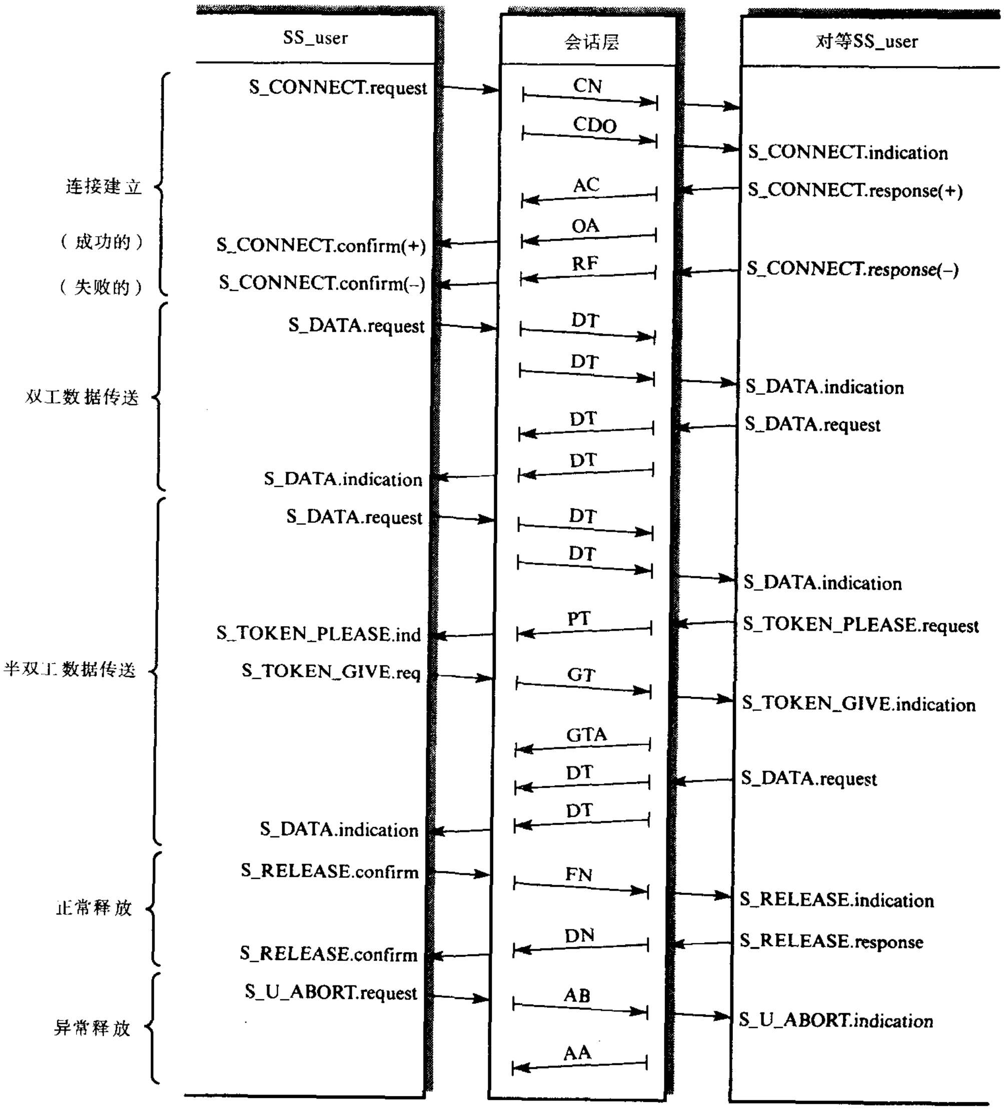


图12-5　会话协议数据单元（只对BCS）

12.1.4　协议规范

702~703 会话协议实体的形式化规范说明的标准化文档已在第11章描述过。此处的目的仅是以BSC的连接建立阶段为例进行一般介绍，参考相关的标准给出更完整的描述。

首先，图12-7给出各种入事件、自动机状态、出事件、谓词和特定动作的列表。给出了每个入事件和出事件所用的缩写名及其相关的层接口，在图12-8的事件—状态表中使用了同样的名字。

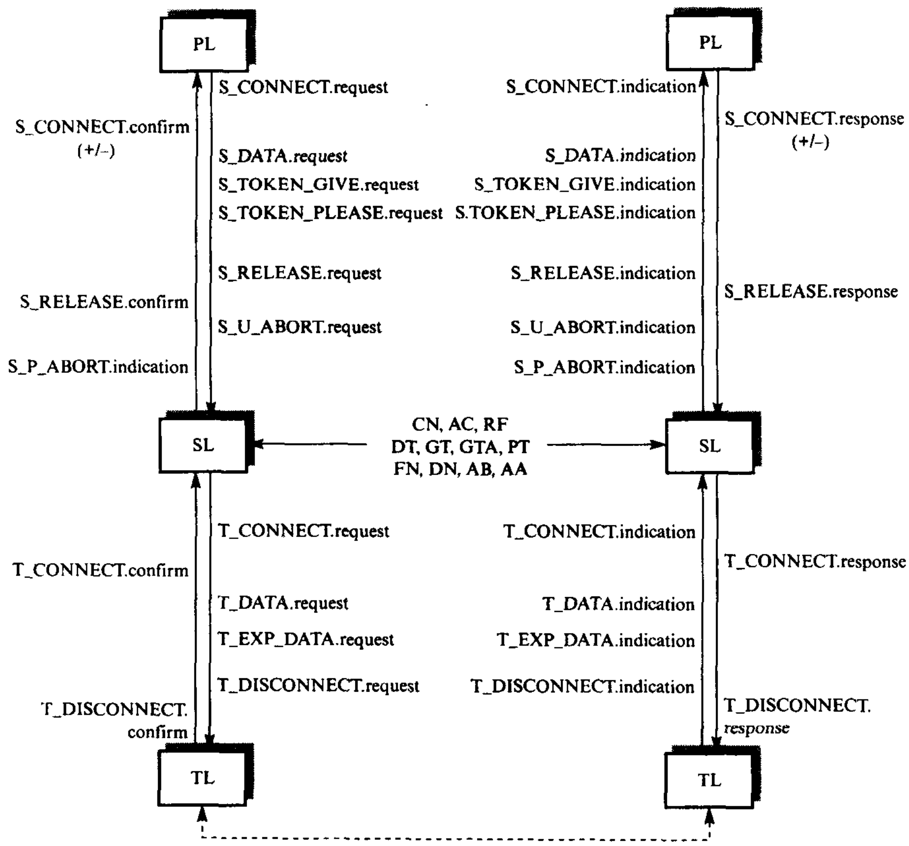


图12-6 会话层概括（只对BCS）

下面传输层连接可由每个会话实体发起。会话实体通常可看作**会话协议机**（SPM）。超时机制与RF（REFUSE）SPDU相关：如果在收到T_DISCONNECT.indication原语（表示传输连接已经断开）前，定时器（STIM）到时，则重发一个RF SPDU。实际上，如果使用不同的基本子集，SPM会有许多附加的状态变量，它们保留以下信息：与会话连接一起使用的令牌类型，令牌的当前分配，当前同步点序列号等等。不过，协议规范与第11章讨论的形式是一致的，所以可以使用相同的方法来实现。 704

12.2 表示层

表示层涉及两个应用进程间传送信息的数据表示方法（语法）。它的目的就是确保两个应用进程之间交换的信息对于两个进程具有一样的含义，这称为**共享语义**。表示层的示意图如图12-9所示。

对于分布式应用来说，所有应用进程都必须知道应用相关信息的语法。例如，如果应用包含客户的账户数据，所有系统中的应用进程必须使用相同的方式解释账户中的每个字段。然而，如12.1节所示，高级语言中的数据类型的表示在每台机器上都有可能不同。为了确保以相同的方式解释数据，在两个进程间传送任何数据之前，都要把它从本地的（抽象）语法转换成应用范围的传送或具体语法。相似的是，如果本地语法与传送语法不同，*在处理任何*接收到数据之前，也要把它转换成本地的语法形式。 705～707

(a)

缩写名	接　口	含　义
SCONreq	SS_user	接收到S_CONNECT.request
SCONresp(+)	SS_user	接收到S_CONNECT.response（接受）
SCONresp(-)	SS_user	接收到S_CONNECT.response（拒绝）
TCONind	TS_provider	接收到T_CONNECT.indication
TCONconf	TS_provider	接收到T_CONNECT.confirm
CN	TS_provider	接收到连接SPDU
AC	TS_provider	接收到接受SPDU
RF	TS_provider	接收到拒绝SPDU
|	|	
|	|	
|	|	
STIM	定时器	定时器STIM到时

(b)

缩写名	含　义
STA 01	空闲，没有传输连接
STA 01B	等待T_CONNECT.confirm
STA 01C	空闲，处在传输连接状态
STA 02	等待AC SPDU
STA 08	等待S_CONNECT.response
STA 16	等待T_DISCONNECT.indication
|	|
|	|
|	|
STA 713	数据传送

(c)

缩写名	接　口	含　义
SCONind	SS_user	发出S_CONNECT.indication
SCONconf(+)	SS_user	发出S_CONNECT.confirm（接受）
SCONconf(-)	SS_user	发出S_CONNECT.confirm（拒绝）
SPABTind	SS_user	发出S_P_ABORT.indication
TCONreq	TS_provider	发出T_CONNECT.request
TCONresp	TS_provider	发出T_CONNECT.response
TDISreq	TS_provider	发出T_DISCONNECT.request
CN	TS_provider	发送连接SPDU
AC	TS_provider	发送接受SPDU
RF	TS_provider	发送拒绝SPDU
AB	TS_provider	发送异常中止SPDU
|		
|		
|		

(d)

缩写名	含　义
P1	启动TC的SPM
|	|
|	|
|	|

(e)

缩写名	含　义
[1]	置P1为假
[2]	置P1为真
[3]	停止定时器STIM
[4]	启动定时器STIM
|	|
|	|
|	|

图12-7　会话协议规范中的缩写名

(a) 入事件　(b) 自动机状态　(c) 出事件　(d) 谓词　(e) 特定动作

这时就产生了两个问题：首先，应该使用什么抽象语法，其次，对于传送语法要采用什么样的表示。对于第一个问题可以用如下方法解决：假设所有的程序都使用同样的高级语言编写，并且使用这个语言声明与应用相关的所有数据类型。但是，不同的程序员会选择使用不同的语言。所以关于传送语法的问题还没有解决方案。

因为在分布式应用中数据的表示是一个共同的需求，ISO（与ITU-T协作）定义了一种通

用的抽象语法，它适合大多数分布式应用中数据类型的定义。称为**抽象语法表示法1**（ASN.1）。现在，多数的新应用都使用ASN.1。如名字所示，用ASN.1定义的数据类型是抽象数据类型。因此，除抽象的语法定义以外，还定义了相关的传送语法。

事件 \ 状态	STA 01	STA 01B	STA 01C	STA 02	STA 08	STA 16	---	STA 713
SCONreq	1	0	2	0	0	0		
SCONresp (+)	0	0	0	0	3	0		
SCONresp (–)	0	0	0	0	4	0		
TCONind	5	0	0	0	0	0		
TCONconf	0	6	0	0	0	0		
CN	0	0	8	0	0	7		
AC	0	0	7	9	0	12		
RF	0	0	7	10	0	12		
⋮								
STIM	0	0	0	0	0	11		

0 = SPABTind, AB, STA 01

1 = TCONreq, [2], STA 01B

2 = P1: CN, STA 02A

3 = AC, STA 713

4 = RF, [4], STA 16

5 = TCONresp, [1], STA 01C

6 = CN, STA 02A

7 = TDISreq, [3], STA 01

8 = P1: TDISreq, STA 01;
NOT P1: SCONind, STA 08

9 = SCONconf(+), STA 713

10 = SCONconf(–), TDISreq, STA 01

11 = RF, [4], STA 16

12 = STA 16

图12-8 SPM事件—状态表

借助使用ASN.1，现在很多公司销售一定编程语言范围的**ASN.1编译器**。有针对Pascal语言的编译器和针对C语言的编译器。图12-10显示了这些编译器的一般使用方法。

首先，使用ASN.1定义与应用相关的数据类型，例如，如果编写两个应用进程，一个使用Pascal另一个使用C，要使用各自的编译器对ASN.1类型的定义进行处理。编译器会输出与适当语言等价的数据类型定义，以及一组针对各个数据类型的**编码**和**解码过程/函数**。当系统中的表示层实体使用这些编码和解码过程/函数进行各种数据类型的编码和解码时，这些数据类型就和通信应用软件联系起来，并被这些软件使用。 708

将在12.3节看到与应用相关的每个数据类型都有一个**标记符**（tag），它用于标识。这个标记符与数据类型一同传送给表示层实体，表示层实体使用它们在传送信息之前调用编码程序。编码程序输出的是字节串，其中第一个字节表示数据的类型。接收方表示层实体使用这个字符串调用相应的解码程序。解码程序会以抽象语法的形式输出，然后把输出传送给应用程序进行处理。

虽然，表示层的主要功能是语法转换，但由于它在数据传送之前对数据进行处理并且对接收数据进行处理，所以它是执行**数据加密**和**数据压缩**（如需要）的最佳位置。当源表示实

体把信息中的数据从本地抽象语法转换成相应的传送语法后，首先使用预先协商并一致同意的加密算法和密钥对数据进行加密，然后对已经加密的数据使用适当的（协商同意的）压缩算法进行压缩。在解码之前，由接收协议实体实施逆功能把接收到的数据转换成为本地的抽象语法结构，并准备传递给接受方应用实体。在讨论表示层协议实体的具体操作之前，首先考虑一下ASN.1和数据加密，关于数据压缩已经在第3章讨论过。

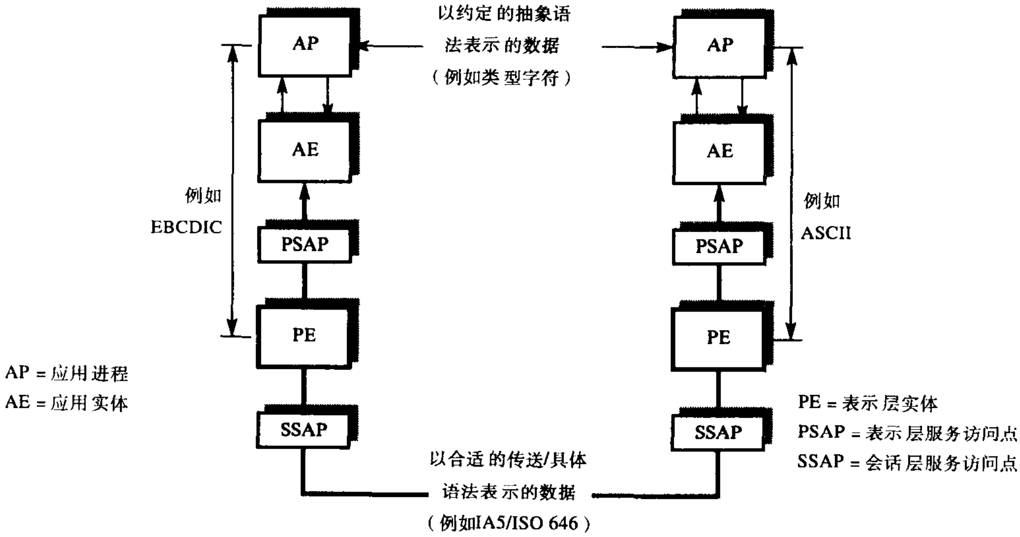


图12-9 表示层示意图

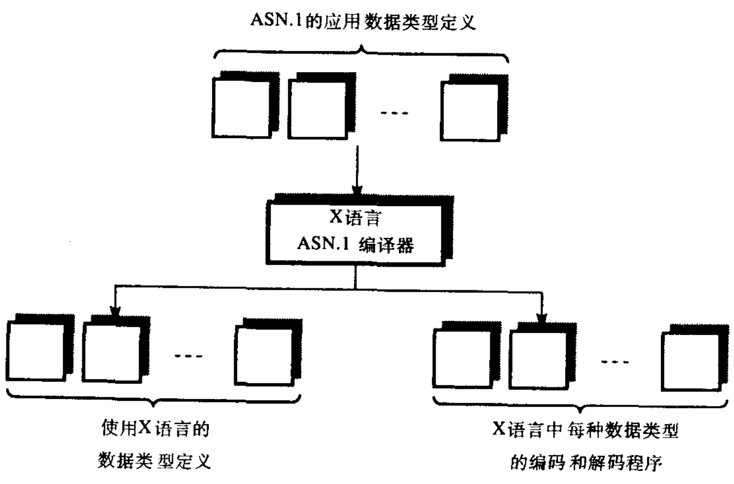


图12-10 共享的ASN.1语义

12.3 ASN.1

ASN.1有两种用途：定义应用语法（例如电子邮件的结构定义）；定义与协议实体相关的PDU结构的一种方法。虽然在本书中不会给出关于ASN.1的完整定义，但接下来的讨论足以使读者理解由ASN.1定义的特定协议实体PDU的含义。它使用的原理与高级语言中定义程序

中变量的数据类型（当声明变量时，都要定义数据类型）时使用的原理是相同的。当对变量赋值时，它的语法就是定义类型。

ASN.1支持很多类型标识符，各数据类型可分成四类：

UNIVERSAL（通用类）	例如整数类型等通用类型
CONTEXT-SPECIFIC（特定上下文类）	用于特定的上下文环境
APPLICATION（应用类）	普遍用于完整应用实体
PRIVATE（专用类）	由用户定义

709

UNIVERSAL类中包含的数据类型可能是基本（简单）类型也可能是构造（结构）类型。一个基本类型可以是一个不能分解的基本数据类型，例如BOOLEAN或INTEGER，或者在某些情况下是一个或多个有相同类型的基本数据元素串，例如一个或多个位串，字节串或IA5字符串/图形字符串。ASN.1关键字总是使用大写字母，下面列出了有效的基本类型：

UNIVERSAL（基本）: BOOLEAN
INTEGER
BITSTRING
OCTETSTRING
REAL
ENUMERATED
IA5String/GraphString
NULL
ANY

一些简单类型的例子如图12-11(a)所示。

作为一个程序列表，可以在列表每行的任何位置插入注释；注释以一对连字符开始，以另一对连字符或者该行的结束作为注释结束。赋值符号是::=，BITSTRING类型的单个位赋值在一对大括号中给出，并在一对圆括号中附加上位所在的位置。对于ENUMERATED类型的变量，可以使用相似的方法得到可能的值。INTEGER类型是有符号的理论上无限大的数字，而REAL类型用{m-B-e}的形式表示，其中m表示尾数，B表示基数，e表示指数，整个变量值是$m \times B^e$。 710

NULL型是一个简单的变量，通常被没有类型赋值的构造型的成分变量使用。ANY类型来声明要在其他地方确定变量的类型。

构造类型通过引用一个或多个其他类型定义，每个类型可以是简单类型也可以是构造类型。ASN.1使用的构造类型包括：

- UNIVERSAL SEQUENCE：一个定长（有界）和有序的类型表，其中有些类型声明为可选的，即实体在构造类型表时，可以忽略相关类型的值。
- SEQUENCEOF：一个定长的或无界的有序元素表，其中所有元素具有相同类型。
- SET：一个定长无序的类型表，有些可声明为可选的。
- SETOF：一个定长或无界的无序的元素表，其中所有的元素具有相同的类型。
- CHOICE：一个定长无序的类型表，每个类型从预先指定的类型集中选出。

图12-11(b)显示了构造类型的例子，还有用于比较的Pascal等价类型定义。

为了允许引用一个结构类型中的独立元素，ASN.1支持了**标记符化**的概念。它为每个元素

赋予一个**标记符**或**标识符**，这与大多数高级语言中数组类型所使用的下标是类似的。

标记符可以声明为以下类型：

CONTEXT-SPECIFIC：这种标记符只在当前的结构类型的范围中有效。

APPLICATION：这种标记符在整个应用的上下文环境（类型的集合）中有效。

PRIVATE：这种标记符只对用户有效。

针对图12-11(b)中使用的类型定义，图12-11(c)给出了一个标记符使用情况例子。在这个例子中假设empNumber会在其他类型定义中使用，所以给了它一个应用范围内惟一的标记符，而对于其他三个变量只在这个序列类型中引用。

ASN.1支持的另外一个特点是使用关键字IMPLICIT声明一个变量是**蕴涵类型**，关键字IMPLICIT位于变量名之后，如果有标记符时，写在类型标记符之前。

711 通常，变量的类型是显式定义的，但是如果一个变量用IMPLICIT类型声明，则这个变量的类型是蕴涵的，可以通过它与其他变量的次序来决定。它主要使用在标记符的类型上，因为这种变量类型可以通过标记符数字来确定。图12-11(d)显示了一个例子，在这个例子中最后三个变量的类型可以通过标记符数字确定，而不是显式地定义。接下来讨论ASN.1的编码解码规则时，会更清楚地看到它带来的益处。

```
(a)          married ::= BOOLEAN -- true or false
    yrsWithCompany ::= INTEGER
      accessRights ::= BITSTRING{read(0), write(1)}
       PDUContents ::= OCTETSTRING
              name ::= IA5String
                pi ::= REAL -- mantissa;base, exponent
           workDay ::= ENUMERATED{monday(0), tuesday(1) ... friday

(b)  personnelRecord ::= SEQUENCE{
                         empNumber INTEGER,
                         name IA5String,
                         yrsWithCompany INTEGER
                         married BOOLEAN}

   c.f. personnelRecord =  record
                         empNumber = integer;
                         name = array [1..20] of char;
                         yrsWithCompany = integer;
                         married = boolean
                       end;

(c)  personnelRecord ::= SEQUENCE{
                         empNumber [APPLICATION1] INTEGER,
                         name [1] IA5String,
                         yrsWithCompany [2] INTEGER,
                         married [3] BOOLEAN}

(d)  personnelRecord ::= SEQUENCE{
                         empNumber [APPLICATION1] INTEGER,
                         name [1] IMPLICIT IA5String,
                         yrsWithCompany [2] IMPLICIT INTEGER,
                         married [3] IMPLICIT BOOLEAN}
```

图12-11 ASN.1类型定义实例

(a) 基本类型 (b) 构造类型 (c) 标记 (d) 隐式类型定义

为了阐述ASN.1中的其他类型，讨论一下ASN.1中协议数据单元（PDU）的定义情况。至今为止讨论的PDU的类型定义都使用有序的位串形式定义，其中每个字段需要的位数目和位在串中的次序都被明确地定义了。这确保了每个PDU中的字段在所有的系统中都以相同的方式解释。

为了使PDU的长度最小化，很多字段都只包含相关的几个位。用这种定义方法，使用高级语言实现这些协议并不容易，这是因为在接收字节串中分离每个字段时（以及随后的编码中）包含了复杂的位处理。

为了克服这个问题，对于所有高层协议层（表示层和应用层）的PDU定义，使用ASN.1定义。把每个PDU定义传送给一个适当的ASN.1编译器，每个PDU中所有字段的类型定义会自动地转换成适当的高级语言兼容形式。可以使用这些类型定义以选择的语言编写协议。然而，因为这些字段都是抽象的语法形式，在两个对等的协议实体实际传送PDU时，必须使用ASN.1编译器产生的相应编码解码程序把每个字段在具体的语法之间进行转换。

ASE使得一个文件服务进程可以以一种开放的方式被访问和使用，这种方式称为FTAM（文件传送访问和管理）。将在第13章讨论它的操作。FTAM的PDU定义会在图12-12中给出。然而，作为ASN.1应用的一个例子，它与图11-15(b)中的例子的目的是相同的。

```
ISO8571-FTAM DEFINITIONS ::=

BEGIN

PDU ::= CHOICE {
                    InitializePDU,
                    FilePDU,
                    BulkdataPDU
                    }
InitializePDU ::= CHOICE {
                        [APPLICATION 1]   IMPLICIT FINITIALIZErequest,
                        [1]               IMPLICIT FINITIALIZEresponse,
                        [2]               IMPLICIT FTERMINATErequest,
                        [3]               IMPLICIT FTERMINATEresponse,
                        [4]               IMPLICIT FUABORTrequest,
                        [5]               IMPLICIT FPABORTresponse
                          }

FINITIALIZErequest      ::= SEQUENCE {
                        protocolId [0] INTEGER { isoFTAM (0) },
                        versionNumber [1] IMPLICIT
                                      SEQUENCE { major INTEGER,
                                                 minor INTEGER},
                                     -- initially { major 0, minor 0}
                        serviceType [2] INTEGER { reliable (0),
                                                  user correctable (1)}
                        serviceClass [3] INTEGER { transfer (0),
                                                   access (1),
                                                   management (2)}
                        functionalUnits [4] BITSTRING { read (0),
                                                        write (1),
                                                        fileAccess (2),
                                                        limitedFileManagement (3),
                                                        enhancedFileManagement (4),
                                                        grouping (5),
                                                        recovery (6),
                                                        restartDataTransfer (7) }
                        attributeGroups [5] BITSTRING { storage (0),
                                                        security (1) }
                        rollbackAvailability [6] BOOLEAN DEFAULT FALSE,
                        presentationContextName [7] IMPLICIT ISO646String {"ISO8822"},
                        identifyOfInitiator [8] ISO646String OPTIONAL,
                        currentAccount [9] ISO646String OPTIONAL,
                        filestorePassword [10] OCTETSTRING OPTIONAL,
                        checkpointWindow [11] INTEGER OPTIONAL }

FINITIALIZEresponse ::= SEQUENCE {
                              :
                              :
END
```

图12-12 ASN.1 PDU定义实例

与特定协议实体有关的完整PDU集合称为**模块**。模块的名字称为**模块定义**。FTAM的模块定义ISO8571-FTAM DEFINITIONS如图12-12所示。该模块从赋值符号（::=）后开始；模块体在关键字BEGIN和END之间定义。

在BEGIN后面的CHOICE类型指明FTAM中PDU属于三类PDU的一类：InitializePDU（启动FTAM），File PDU（FTAM文件）和BulkdataPDU（FTAM大块数据）。接下来，另一个CHOICE指明属于InitializePDU的有6个不同的PDU类型：FINITIALIZErequest，FINITIALIZEresponse，等等。注意，这些类型都注上标记，所以可以区分。此外，标记后面是IMPLICIT标志，就是说PDU类型蕴涵在标记字段中，无须定义PDU类型。由于FINITIALIZErequest PDU总是FTAM中第一个接收的PDU，所以它的特定应用标记是1。其余的PDU类型都有一个特定上下文标记，因为
712 它们在FTAM上下文中有意义，所以不需要前缀保留字CONTEXT。图12-12给出了每个PDU的定义和FINITIALIZErequest PDU的定义。

在这个定义中，使用SEQUENCE构造类型（和Pascal中的记录类型相似），SEQUENCE指明PDU由若干个数据元素组成，每个元素可以是基本类型，也可以是构造类型。虽然SEQUENCE类型中的变量类型表是有序集合，但通常各个元素都注有特定语意上下文标记。如将在12.3.1节中看到的，这样做使PDU的编码方案更有效。第一个元素是protocolId，属于
713 INTEGER并赋值为0，表明它是一个ISO FTAM（isoFTAM）。第二个元素是versionNumber，它定义为两个INTEGER类型SEQUENCE（主和次）。如前，关键字IMPLICIT意味着SEQUENCE类型蕴涵在先前的标记字段中，无需编码。注释字段指明两个变量的始值。接下来的两个元素都是INTEGER类型，大括号中是它的可取值。

接下来的元素是functionalUnits，属于BITSTRING类型，位串中8个位或者置1或者置0，这依赖于特定的单元是否被请求，请求为（1），未请求为（0）。最后，序列中接下来的一些元素声明为可选的，即它们可以出现在编码的PDU中，也可以不出现。因为PDU中的每个元素都注上标记，所以PDU的接收方可以确定各个元素是否存在。关键字DEFAULT具有相似的含义，除非PDU中没有出现这个元素，如果出现了，则该元素赋予默认值。

12.3.1 编码

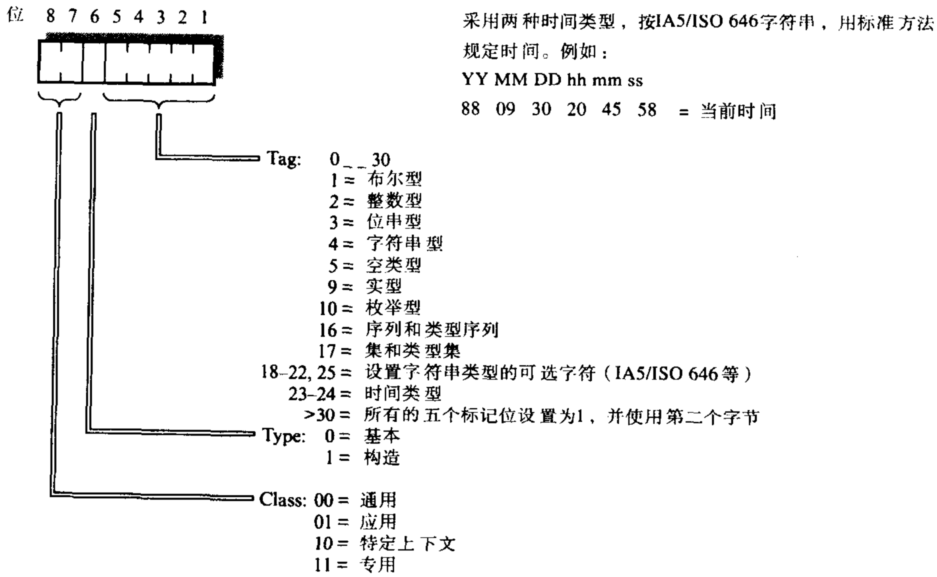
注意，ASN.1是抽象语法，意味着虽然一个数据元素被定义成特定类型，它却不必有一个
714 固定的语法。因此即使组成PDU的各个数据元素有相同的（抽象）类型，它们的结构（语法）可能是不同的。因此，ASN.1包含了相关编码规则，把用ASN.1形式定义的PDU中的每个字段转换成相应的具体语法形式。具体语法形式在两个应用实体之间传送，这时交换的PDU对两个实体具有相同的含义。

每种类型值的标准表现形式是包含如下三个字段的数据元素：

- 标识符字段，定义ASN.1类型。
- 长度字段，定义内容字段中的字节个数。
- 内容字段，定义内容（对于构造类型，定义的可以是其他数据元素）。

每个字段包含一个或多个字节。标识符字节的结构如图12-13所示。图12-14给出不同类型
715 值的编码实例。为了增加可读性，每个字节中的内容都用两个16进制的数字表示，并在每个例子的末尾给出这个例子的编码值。如果内容字段中的字节数目超过127，则第一个长度字节中的最高有效位置为1，用两个（或多于）字节定义长度。

在图12-14(a)中，标识符01（十六进制）表示它的类是UNIVERSAL（位8和位7是00），是一个基本类型（位6是0），并且标记符（位1~5）是1，所以它是Universal 1和BOOLEAN。长度字段是01（十六进制）就是说内容只有一个字节。TRUE被编码为FF（十六进制），而FALSE是00（十六进制）。



注意：采用空类型表示序列中没有元素

图12-13　ASN.1标识符位定义

整数值使用**补码形式**进行编码，其中最高有效位作为符号位来处理。一个字节可以用来表示－128～+127的值。对于更大的值可以使用多个字节来表示。只需要使用足够多的字节来表示实际的值，而不需要考虑原始形式下的位数目。就是说，如图12-14(a)中的值29，即使它在本地是用16位或32位表示的，当编码时只会用一个字节来表示它。对于一个BITSTRING类型的值，从最高有效位开始赋值，而对没有使用的位设置为0。

对于SEQUENCE（或SEQUENCEOF）类型的变量；标识符是30（=0011 0000 二进制）。
表示它的类是UNIVERSAL（位8和位7是00），是一个构造类型（位6是1），并且标记符（位5 716
为1，位4~1为0）是16。类似的，对于SET（或SETOF）类型标识符是31，表示它是一个UNIVERSAL，是标记符为17的构造类型。

注意UserName类型中包含两个字段，特定上下文标记[0]和[1]。这两个字段的标识符分别是80（=1000 0000二进制）和A1（=1010 0001二进制）。第一个标识符80表示类是特定上下文相关的（位8和位7为10），是一个简单类型（位6是1），并且标记符是0。而第二个则是特定上下文相关的，结构类型，并且标记符是1。这是因为第一个特定上下文标记符被声明为IMPLICIT，这种情况下类型字段可以蕴涵在标记符中。然而，对于第二个，必须指定类型字段，所以就需要两个附加字节。

```
(a)  BOOLEAN – UNIVERSAL 1

     e.g., Employed ::= BOOLEAN
     – – assume true

     Identifier = 01 (Hex)                     – – Universal 1
     Length     = 01
     Contents = FF

     i.e., 01 01 FF

     INTEGER – UNIVERSAL 2

     e.g., RetxCount ::= INTEGER
     – – assume = 29 (decimal)

     Identifier = 02                           – – Universal 2
     Length     = 01
     Contents = 1D                             – – 29 decimal

     i.e., 02 01 1D

     BITSTRING – UNIVERSAL 3

     e.g., FunctionalUnits ::= BITSTRING {read (0), write (1), fileAccess (2)}
     – – assume read only is required

     Identifier = 03
     Length     = 01
     Contents = 80                             – – read only = 1000 0000

     i.e., 03 01 80

     UTCTime – UNIVERSAL 23

     e.g., UCTTime ::= [UNIVERSAL 23] IMPLICIT ISO646String
     – – assume 2.58 p.m. on 5th November 1989 = 89 11 05 14 58

     Identifier = 17 (Hex)                     – – Universal 23
     Length     = 0A
     Contents = 38 39 31 31 30 35 31 34 35 38

     i.e., 17 0A 38 39 31 31 30 35 31 34 35 38

(b)  SEQUENCE/SEQUENCEOF – UNIVERSAL 16

     e.g., File ::= SEQUENCE {userName IA5String, contents OCTETSTRING}
     – – assume userName = "FRED" and contents = 0F 27 E4 Hex

     Identifier = 30 (Hex)                     – – Constructed, Universal 16
     Length     = 0B                           – – Decimal 11
     Contents = Identifier = 16                – – Universal 22
                Length     = 04
                Contents = 46 52 45 44
                Identifier = 04                – – Universal 4
                Length     = 03
                Contents = 0F 27 E4

     i.e., 30 0B 16 04 46 52 45 44 04 03 0F 27 E4

     Tagging/IMPLICIT

     e.g., UserName ::= SET {surname [0] IMPLICIT ISO646String, password [1] ISO646String }
     – – assume surname = "BULL" and password = "KING"

     Identifier = 31
     Length     = 0E                           – – Constructed, Universal 17
     Contents = Identifier = 80                – – Decimal 14
                Length     = 04                – – Context-specific 0 = surname
                Contents = 42 55 4C 4C
                Identifier = A1                – – Context-specific 1 = password
                Length     = 06
                Contents = Identifier = 16     – – Universal 22
                           Length     = 04
                           Contents = 4B 49 4E 47

     i.e., 31 0E 80 04 42 55 4C 4C A1 06 16 04 4B 49 4E 47
```

图12-14 ASN.1编码实例

(a) 基本类 (b) 构造型

```
(a)   FINITIALIZErequest = { protocolId = 0,
                             versionNumber {major = 0, minor = 0}
                             serviceType = 1,
                             serviceClass = 1,
                             functionalUnits {read = 0, write = 1, fileAccess = 2,
                                              limitedFileManagement = 3
                                              enhancedFileManagement = 4,
                                              grouping = 5, recovery = 6,
                                              restartDataTransfer = 7}
                             attributeGroups {storage = 0, security = 1}
                             rollbackAvailability = T,
                             PresentationContextName = "ISO8822"}

(b)   Identifier = 61                              -- Application-specific 1 = FINITIALIZErequest
      Length   = 31                                -- decimal 49
      Contents = Identifier = A0                   -- Context-specific 0 = protocolId
                 Length   = 03
                 Contents = Identifier = 02        -- Universal 2 - INTEGER
                            Length   = 01
                            Contents = 00          -- isoFTAM
                 Identifier = A1                   -- Context-specific 1 = versionNumber
                 Length   = 06
                 Contents = Identifier = 02        -- Universal 2
                            Length   = 01
                            Contents = 00          -- major
                            Identifier = 02        -- Universal 2
                            Length   = 01
                            Contents = 00          -- minor
                            Identifier = A2
                            Length   = 03
                            Contents = Identifier =02
                                       Length   = 01
                                       Contents = 01        -- serviceType = user correctable
                            Identifier = A3
                            Length   = 03
                            Contents = Identifier = 02
                                       Length   = 01
                                       Contents = 01        -- serviceClass = access
                            Identifier = A4                 -- Context-specific 4 = functionalUnits
                            Length   = 03
                            Contents = Identifier = 03      -- Universal 3 = BITSTRING
                                       Length   = 01
                                       Contents = E0        -- read, write, fileAccess = 1110 000
                            Identifier = A5                 -- Context-specific 5 = attributeGroups
                            Length   = 03
                            Contents = Identifier = 03
                                       Length   = 01
                                       Contents = 40        -- security 0100 000
                            Identifier = A6                 -- Context-specific 6 = rollbackAvailability
                            Length   = 03
                            Contents = Identifier = 01      -- Universal 1 = BOOLEAN
                                       Length   = 01
                                       Contents = FF        -- true
                            Identifier = A7                 -- Context-specific 7 = PresentationContextName
                            Length   = 07
                            Contents = 49 53 4F 38 38 32 32          -- "ISO8822"
```

上面的PDU的具体语法表示就是：

```
61 2F A0 03 02 01 00 A1 06 02 01 00 02 01 00 A2
03 02 01 01 A3 03 02 01 01 A4 03 03 01 E0 A5 03
03 01 40 A6 03 01 01 FF A7 07 49 53 32 38 38 32
32
```

图12-15 FTAM PDU编码实例

(a) PDU内容 (b) 编码格式

图12-15给出了一个FTAM PDU编码的实例。所选的是FINITIALIZErequest PDU，它在前面图12-12中用ASN.1的形式定义。PDU实际值在图12-15(a)中定义，图12-15(b)表示PDU是如何进行编码的。如将在第13章看到的，在PDU中的各个变量都是抽象的数据类型，它们在程序中都对应一个数据结构。然而，经过编码后的PDU是由一串精确定义的字节组成。而且为了增强可读性，它们都是用十六进制的形式表示的。然后整个字节串被传送给对应（对等）的FTAM协议实体，在此解码，转换成为本地的抽象形式。

12.3.2 解码

接收到编码字符串，对应表示实体要执行相应的解码（译码）操作。例如，字符串中的首字节，用来确定接收到的PDU的类型，特定应用1 = FINITIALIZErequest。显然，因为每个PDU都有惟一的结构，因此对每个PDU类型需要一个独立的解码规程。在确定了接收PDU的类型后，调用相应的解码规程。然后，组成PDU的各个字段就被转换成本地的（抽象）语法
717 形式，接下来就可以对PDU进行处理了。

12.4 数据加密

当前计算机网络和协议的知识广播的越来越广泛，在信息穿越网络时，以欺骗方式截取
718 和解码信息数据的情况也越来越多。例如，大多数应用的端系统（站）都连接在LAN上。这个应用可能包含单个LAN、一个互连环境、一个或多个中间WAN。然而，对于多数的LAN，在共享的传输介质上传输的信息可能会被任意一个系统截取，入侵者只需把适当的MAC芯片设置成为混淆模式并记录传输介质上所有的信息。然后，依靠所使用的LAN协议的相关知识，入侵者可以识别并分离出每个信息包中的协议控制信息，只留下内容。于是信息内容，包括密码和其他敏感信息，就可以被解释出来。

这被称为**监听**或**偷听**，它的效果十分明显。另外，更危险的是入侵者可以使用记录下来的信息序列创建一个新的序列，这被称为**伪装**，它的危害也是十分显著的。因此，对于要用网络传送的所有数据都要使用加密。在ISO参考模型环境中，最适合执行加密操作的协议层是表示层。在本节给出关于数据加密主题的介绍。

12.4.1 术语

数据加密涉及发送部分（例如表示层协议实体）在发送所有数据之前对其进行处理。这样如果在数据传递时它被意外地或故意地截取了，对于解密方式是不能理解的。当然，这些数据必须能被预期的接收方以可读的方式解释出来，称为**解密**或**译码**。因此，多数的加密方法包含了对**加密密钥**的使用，只有通信的双方知道密钥。密钥在加密和解密的过程中都起重要作用。加密之前的信息数据被称为**明文**，加密之后就生成了**密文**。图12-16显示了常规的加密解密过程。

当确定使用一个特定的加密算法时，必须假设所传送的信息会被截取和记录，并且入侵
719 者知道信息使用在何种环境中，即交换的信息类型。目的是选择的加密方法能使入侵者即使使用一个强大的计算机也不可能在一个可能的时间周期内把密文译解出来。当前有两种被广泛使用的加密算法，在讨论它们之前，要讲一些更加基础的理论技术，这两种算法就建立在这些基础之上。

12.4.2 基础技术

最简单的加密技术包括把明文中的字母表**替换**成为新的字母表（即**密文字母表**）。例如，可以通过简单地把明文字母表移动*n*位得到密文字母表，*n*是密钥。因此，如果密钥是3，所得

到的结果如下显示：

明文字母表：a b c d e f g...

密文字母表：d e f g h i j...

通过把明文信息中的每个字母替换成为等价的密文字母，就可以得到密文。

一个更有效的变化方法是把明文字母表进行随机的混合从而得到密文字母表。例如：

明文字母表：a b c d e f g...

密文字母表：n z q a i y m...

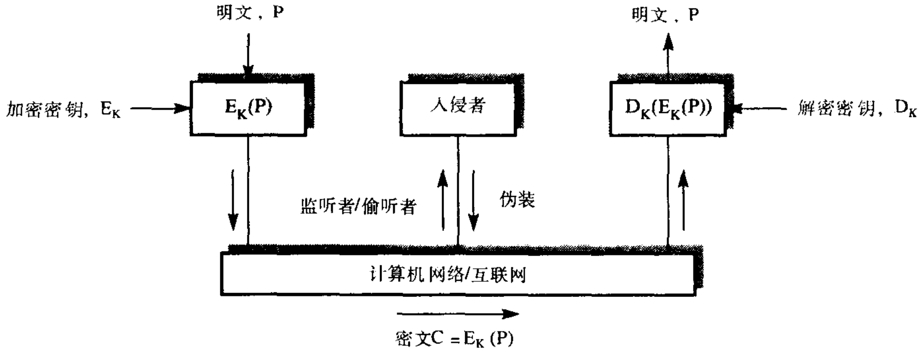


图12-16　数据加密术语

这时密钥由字母表中的字母数目决定，例如，如果传输的是小写字母符号，则是26，如果使用的是ASCII字母表，则是128。对于第一种情况有26！$=4\times10^{26}$种可能的密钥值，而多数情况下会是更大的字母表。通常对于比较大的密钥值，破解它所要花费的时间也更多。

虽然对字母表随机混合的方法是一种强大的技术，但是仍然有破解这种方法的捷径。入侵者很可能知道信息数据使用的环境以及使用的数据类型，例如，如果信息包含文本数据，入侵者可以使用文本数据的统计特性来进行破解：单个字母（e、t、o、a等等）的出现频率、两个字母组合（th、in、er等等）的出现频率和三个字母组合（the、ing、and等等）的出现频率都有相应的规律。通过对密文中的字母执行统计分析，入侵者可以快速对密文进行破解。

替换就是把每个字母用不同的字母代替，这样明文中的字母顺序也会在密文中反映出来。有一种方法是把明文中的字母进行重新排序（变换），例如，如果使用的密钥为4，整个信息首先被分割成4个字符组的集合。信息收送从每个组中的首字母开始，然后是第二个字母，等等。作为例子，假设一个明文信息是“this is a lovely day”，所得到的密文如下： 720

1 2 3 4 ←密钥

t h i s

- i s -

a - l o

v e l y

- d a y

密文 = t-av-hi-edisllas-oyy

显然，现在使用了更复杂的变换，但通常使用单独的变换密码算法会同替代密码算法一样面临同样的缺点。更多实际的加密算法中倾向于把两种技术结合起来应用，这被称为**乘积密码**。

乘积密码

使用变换和替代算法的组合算法。不同于变换/替代算法，因为信息中每个字符（码字）中单个位的顺序也改变了。图12-17(a)显示了三种变换（也称为置换）操作，每种对应一个P盒。

第一种变换操作把每个8位的输入通过交叉耦合转换成为一个8位的输出，它根据定义的密钥把每个输入线连接到一个不同的输出线，这称为直接置换。第二种对于输入位有更多的输出
721 位，它把输入位重新排序然后把选择的输入位传递给多个输出位。这称为扩展置换。第三种对于输入位有更少的输出位，它只对选择的输入位进行传递，这称为压缩置换或选择置换。

为了执行一个8位直接置换，需要定义一组新的256个8位字节。就是说对于一次替换的密钥是2048位。为了减少密钥的长度，采用在解密方和加密方之间封装一个P盒的替换形式，如图12-17(b)所示。现在的替换单元称为S盒。例子中使用P盒的密钥执行2位的替换操作，对于8位的替换需要4个S盒单元。

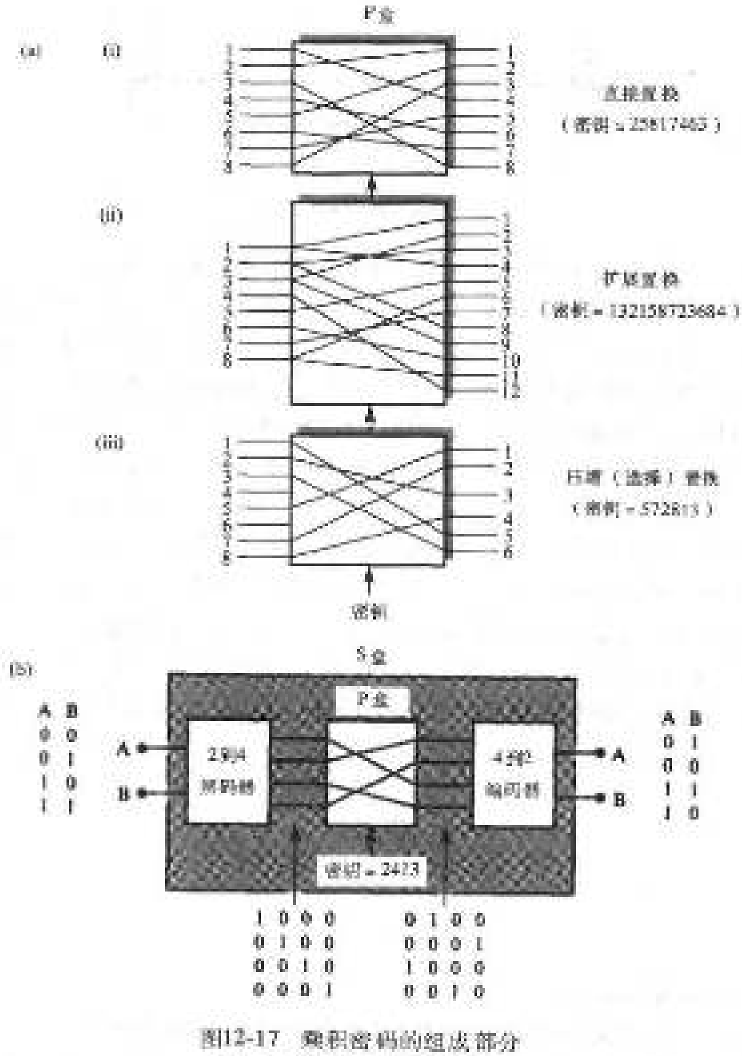


图12-17　乘积密码的组成部分

(a) P盒实例　(b) S盒实例

乘积密码是由这两种基本单元（P盒和S盒）的乘积组合形成的，如图12-18所示。每个P盒和S盒都与图12-17所示的相同。通常，级数越多密码就越强大。一个乘积密码的实例就是**数据加密标准**（DES），它由美国国家标准局定义。现在它在很多领域被广泛地使用。硬件中有各种集成电路执行加密操作，因此能实现更快的加密和解密操作。

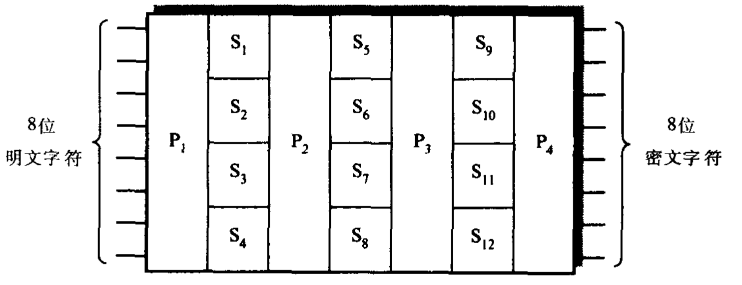


图12-18 乘积密钥实例

12.4.3 数据加密标准

DES算法是**分组密码**，它以固定长度的数据分组工作。首先一个完整的消息被分成若干个明文分组，每个分组包含64位。用惟一的56位密钥，对每一分组明文进行加密，使其成为 722
64位的密文分组，然后密文分组通过网络传输。接收方用相同的密钥，对每个接收到的64位密文数据分组执行反向操作（解密），并将解密后的数据分组重新装配，成为完整的消息。

密钥位数越多，加密就越可靠。同样，密钥越大，破解的难度就越大。在DES中使用56位的密钥意味着，要在 10^{17} 种可能的密钥中进行选择。因此，DES对于大多数的商业应用是足够安全的。

一个DES算法的示意图如图12-19所示。首先，通信双方选择一个56位的密钥，密钥产生16个的不同子密钥，每个子密钥48位，这些子密钥用于随后的替换操作。整个算法包含了19个不同的步骤。第一步用固定的变换规则对64位的明文分组进行简单的变换处理。然后将得

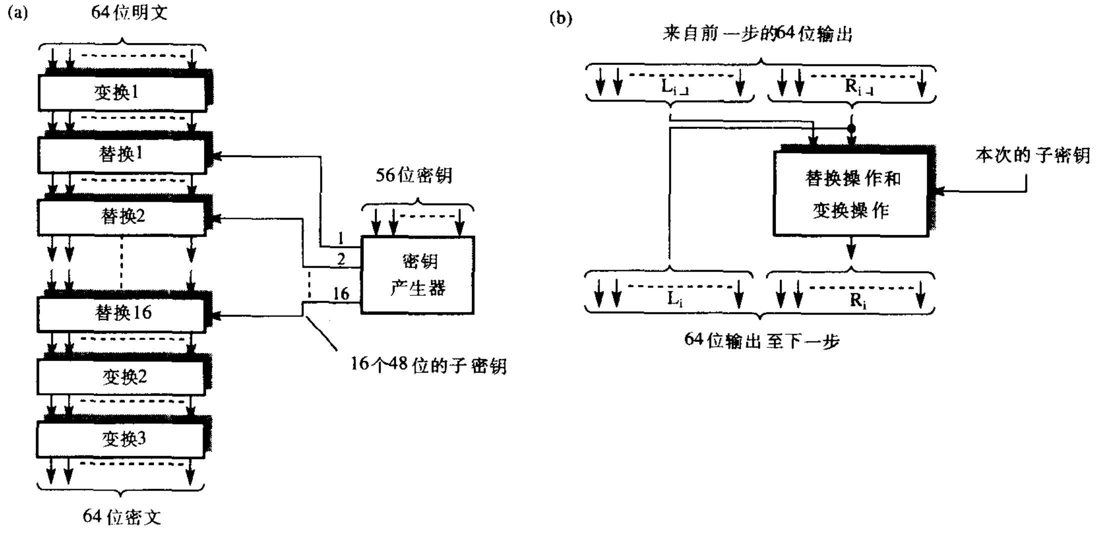


图12-19 DES算法

(a) 整个操作 (b) 替换示意图

到的64位文本通过16次相同的替换操作的迭代处理，所不同的是每一次迭代时，替换操作所用的子密钥不同。最后一次迭代输出的64位中的高32位要和低32位进行交换处理。最后执行与第一次相同的变换处理，实际上是第一次的逆变换，以产生最后的64位密文分组。DES算法设计为，接收方对接收到的分组，以相反的次序执行加密的相同步骤以解密得到明文。

每个替换步骤中使用的16个子密钥是按照下面的方法产生的。首先，对56位密钥执行固
723 定的换位。得到的结果密钥被分成两个独立的28位部分。这两个部分彼此独立地循环左移数位，再把两个部分合并成56位，然后再进行一次变换处理。每个子密钥包含48位，它是通过对变位后数据流的最后56位执行一系列选择操作得到的。其他的子密钥以相似的方法产生，只是循环左移的位数由当前子密钥的数目决定。

加密的16个中间步骤执行的处理是复杂的，这确保了DES算法的有效性。图12-19(b)中显示了大致过程。前面迭代（第i-1）次产生的64位输出首先分为两个32位的部分。然而，右边处于低位的32位直接构造输出块高位32位，即$R_{i-1}=L_i$，然而，输出块低位32位，即R_i，则由输入块的高位L_{i-1}执行一系列的变换和替换操作得到的，而执行这一精确的操作是这一次的子密钥的函数。

因为密文的每个分组都是独立于其他分组的，DES的这种工作方式称为**电子编码本**（ECB）。每个密文分组都有一个惟一的匹配的明文分组，它类似于编码本中一项记录。图12-20（a）显
724 示了ECB工作方式。

虽然DES操作使用ECB方式，很好地解决了单个加密分组的传输中发生的错误和变化，但它对于一连串的加密分组产生的错误没有解决。因为在ECB方式中每个分组都是分别对待的，如果把一个正确的密文分组插入到传输的连续加密分组中，接收方会对这个插入分组进行解密，并把它当作有效分组处理。因此，加密的分组流可能被知道密钥者截取并改动之后再插入，而接收方发现不了变化。另外，这种方式有一个弱点，就是重复的明文分组会产生一串相同的密文分组，而这一点可以为企图破解密钥的人提供很大的便利。另一种操作方式是引入**链式**概念。这种DES操作方式被称为**链式分组密码**（CBC）。

虽然链式加密使用与前面相同的分组加密方法，但每个64位明文分组要首先与前一个分组的密文输出进行异或运算，如图10-20(b)所示。第一个64位明文分组，它要与一个64位随机数字（称为**初始向量**）进行异或运算，接下来的分组就按照图中表示的链式顺序进行操作。
725 这样，一个输出分组就是分组内容和前一个分组的输出的函数，则传输序列中的任何变化都会被接收方检测出来。于是相同的明文分组会产生不同的密文分组。由于以上的优点，数据通信中常用CBC操作方式。

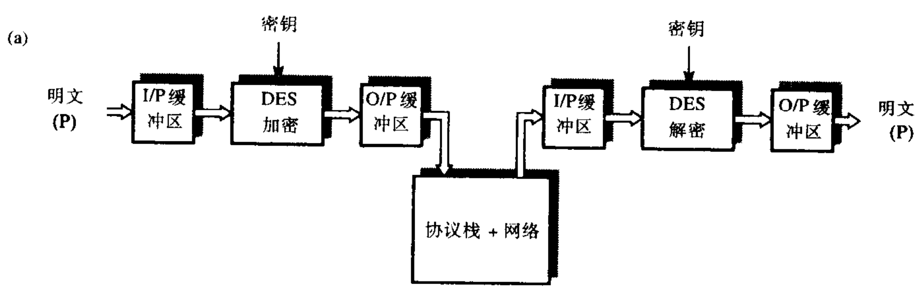


图12-20　DES操作方式

(a) ECB方法　(b) CBC方法　(c) CFM方法

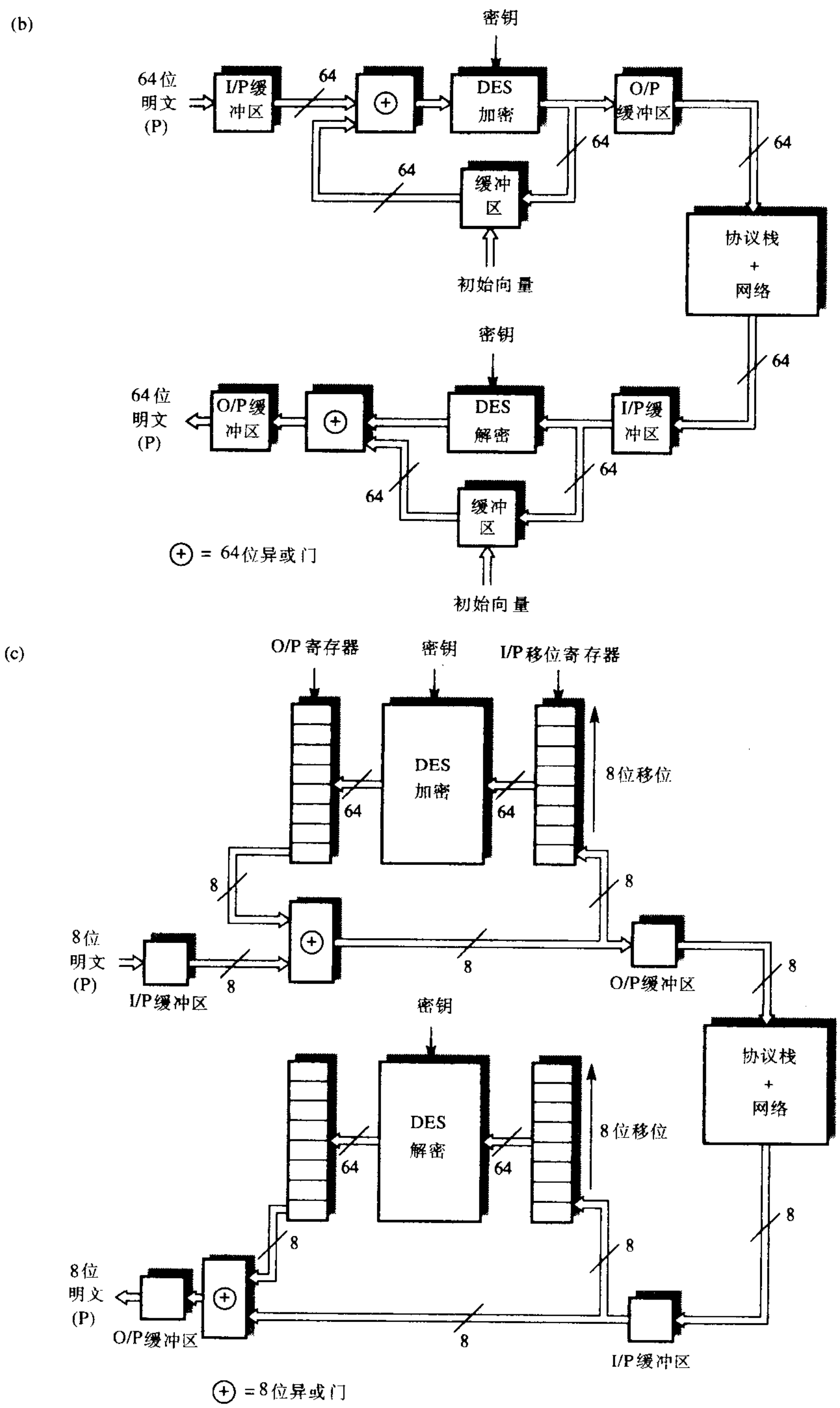


图12-20 （续）

因为基本CBC方式是以64位分组进行操作的，所以所有的信息必须是64位的倍数。否则必须加上补丁数据。然而，如前面描述的，所有消息的内容都由字节串组成，所以所有消息的基本单元是8位而不是64位。另一种DES方式就是**密文反馈方式**（CFM），它被定义为以8位操作。方案的示意如图12-20(c)所示。

这种方式，每当输入8位就执行一个DES加密操作，而不是CBC方式的64位。新的8位输出是DES输出的最低8位与输入的8位的异或结果。然后，当每8位输出装入输出缓存后，把64位的输入移动寄存器中的内容移动8位。8位最高位被丢弃，新的8位输入被装入输入移位寄存器低8位中。然后，在这个新的64位数据上执行DES操作，得到的64位输出装入输出寄存器。后者的最低8位与输入的8位进行异或，接下来重复过程。

当在串行传输线接口上执行加密操作时，CFM尤其有用。这种操作方式被用在DES集成电路上，每个新的8位输出被直接装入串行接口电路上。

这些DES方式都依赖于在加密和解密双方使用同一个密钥。它的一个明显的缺点就是：必须在加密的数据被传输之前发送关于密钥的某种形式的通知。如果密钥的改变不是很频繁，这是可以接受的。但是实际上密钥通常每天都要改变，或者更频繁。显然，通过网络发送新的密钥是不可靠的，所以必须使用另一种方法，例如信使。对于私有密钥加密系统来说，密钥的分发是一个很重要的问题。有时我们使用以公用而不是私有密钥为基础的另一种方法来克服这个困难。最广为人知的公用密钥算法是RSA算法，它是以三个发明者Rivest、Shamir和Adelman的名字命名的。

12.4.4 RSA算法

私有密钥系统与公开密钥系统的最基本的不同点就是后者在加密时使用的密钥和在解密时使用的密钥是不同的。公开密钥系统使用了一对密钥：一个是为发送方使用，另一个是为接收方使用。

虽然没有多少帮助，RSA算法的发明者基于数论开发了一种产生一对数字（即密钥）的
726 方法。使用第一个数字作为密钥进行加密的信息只能使用第二个数字进行解密。而且，这种方法保证不能从第二个数字得到第一个数字。这个特性意味着可以把这对数字中的第一个告诉任何希望发送加密消息给第二个数字持有者的人，只有这个持有者可以对密文进行解密。两个数字中的第一个称为公开密钥，第二个称为秘密密钥。方法的原理如图12-21所示。

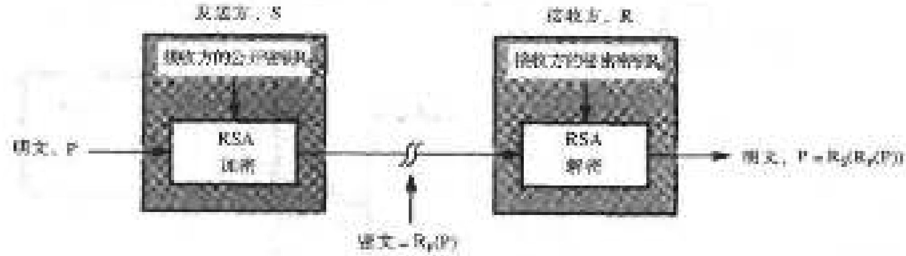


图12-21　RSA示意图

正如指出，这对密钥的推导是基于数论的，超出了本书的范围。然而，计算两个密钥的基本算法是比较简单的，在这里给出一个简单的实例：

生成公开密钥Kp：	例子：
• 选择两个正素数P和Q	P = 7，Q = 17
• 计算X =（P − 1）×（Q − 1）	X = 96
• 选取一个整数E，使E和X互素，即E不是X的因子或其倍数，并且满足下面计算Ks时的条件	E = 5
• 计算N = P × Q	N = 119

• Kp就是N和E的联合	$Kp = 119, 5$
生成秘密密钥Ks：	
• 计算D，使得MOD（D×E，X）=1	$D \times 5/96 = 1$，$D = 77$
• Ks就是N和D的联合	$Ks = 119, 77$
计算明文P的密文C：	$P = 19$
• 把P看成数字值	$C = \mathrm{MOD}(19^5, 119)$
• $C = \mathrm{MOD}(P^E, N)$	$C = 66$
计算密文C的明文P：	$P = \mathrm{MOD}(66^{77}, 119)$
• $P = \mathrm{MOD}(C^D, N)$	$P = 19$

这个实例中E和D的选择是通过考察96的因子来确定的。96的因子是1，2，3，4，6，8，12，16，24，32，48。与96互素的数字是5，7，9，10，11，等等。首先尝试这些数中的第一 727
个，E=5，可以得到D=77满足MOD（D×E，X）=1，所以E=5和D=77满足条件。

从这个实例得知算法中最关键的是两个素数P和Q，它们必须被严格保密。算法的目的就是得到一个足够大的数字N，使得它不能在可能的时间内被分解。关于分解时间的示例如下：

N = 100位≈1 周

N = 150位≈1 千年

N > 200位≈1百万 年

RSA算法需要相当多的计算时间来执行加密和解密操作中的求幂计算。然而，可以使用一个简单的方法来替代求幂计算，它由重复的乘法与除法运算组成：

```
C: = 1
begin for i = 1 to E do
        C: = MOD (C ×P, N)
end
```

解密使用同样的方法，只不过把前面的E换成D，P换成C；就可以得到明文P。例如，计算$C = \mathrm{MOD}(19^5, 119)$时：

步骤 1：$C = \mathrm{MOD}(1 \times 19, 119) = 19$

2：$C = \mathrm{MOD}(19 \times 19, 119) = 4$

3：$C = \mathrm{MOD}(4 \times 19, 119) = 76$

4：$C = \mathrm{MOD}(76 \times 19, 119) = 16$

5：$C = \mathrm{MOD}(16 \times 19, 119) = 66$

注意，N的值决定了可以加密的最大消息。实例中N是119，它等于ASCII中可以编码的字符个数。然而，由一串ASCII字符组成的消息在同一时刻只能对一个字符编码。

虽然公开密钥系统提供了替代私有密钥系统的方法来克服偷听欺骗，但如果公开密钥能够很容易被得到，伪装者会使用它传递伪造的消息。于是又有这样的问题出现了：一个正确的密文的接收者如何确保它收到的消息是由合法的使用者发送的，针对这个问题已经有了很多使用**消息认证**的解决方法。

12.4.5　消息认证

一个解决方法就是利用公开密钥系统的双重性质，也就是说不只一个接收者可以使用自

728 己的私有密钥对所有收到的消息（已经使用它自己的公开密钥加密过了）解密，其他的接收者也可以用发送方的公开密钥对使用发送方私有密钥加密的消息解密。

图12-22表示了如何使用这个性质进行消息认证。加密和解密操作都从两个层次执行。内部层次如前面描述的，而在外部层次，发送方使用它自己的私有密钥对原始数据进行加密。如果接收方能够使用发送方的公开密钥对消息解密，就证明发送方将确是实际上启动此消息的发送者。

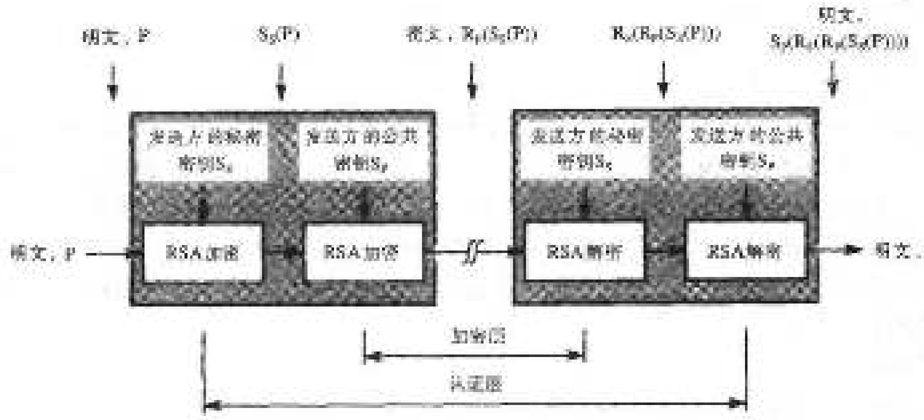


图12-22 使用RSA进行信息认证

虽然这是一个高明的解决方案，但它也有很多局限。首先，RSA算法的处理开销太高。如早先的例子中看到的，即使对于一个很小的消息，它所使用的资源也是很大的。而且，整个消息必须被分成很多的小单元，小单元的大小由所使用的计算机决定。因此，即使有集成电路的帮助，RSA的信息处理率仍然很低。第二，这种方法需要两层加密，即使不需对实际信息进行加密；就是说即使只需要进行消息认证，实际的消息内容也要被加密。

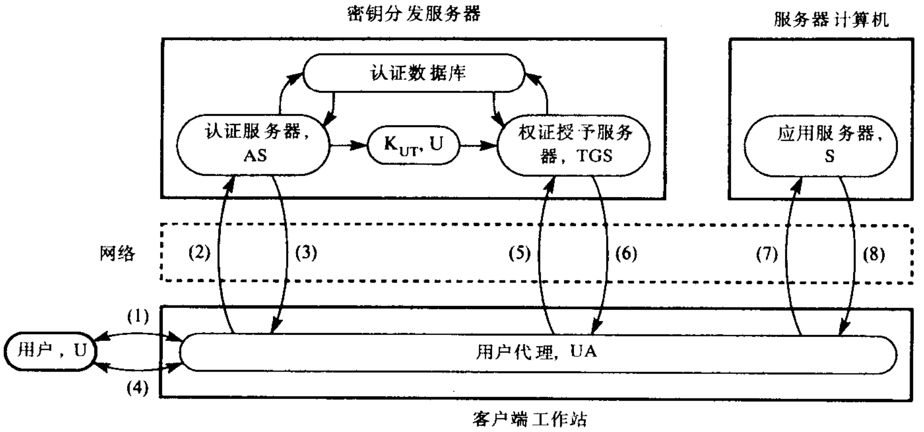
一个解决方法就是计算基于其内容的更短版本的消息。这个消息使用发送方的私有密钥加密后并附在待发送消息的尾部以明文发送。消息的这个加密过的尾部好比一封信结尾的签名，当使用发送方的公开密钥解密后，就可以验证是否真的是这个人发送了这个消息。这个尾部的数据被称为**数字签名**。一个例子就是消息CRC。显然，用于计算CRC的生成器多项式必须保密，只有发送方和接收方才能知道。

一种方法是以只使用私有密钥为基础建立的，它使用一个可靠的第三方机构作为**密钥分**
729 **配服务器**，例如**Kerberos安全系统**。Kerberos安全系统中使用的基本安全控制机制是一组加密标签（也被称为控制或许可令牌），这些标签用来控制系统中各个服务器的访问权限。这些服务器包含了一些应用服务器（文件服务器、电子邮件服务器，等等）和发布标签的系统服务器，也被称为**标签授予服务器**。在用户和标签授予服务器之间交换的所有消息与在用户和应用服务器之间交换的所有消息都使用作为对应标签一部分的私有密钥来加密。另外，每个消息/标签都有相关的现时记录，其中包含了两个日期和时间值，第一个指明了现时的创建时间。现时记录不只用来确定消息的来源，它也用来把标签的有效性限制在一定的生命期里。这是由现时记录中的第二个日期和时间值来实现的。这个特点使偷听者只能在有限的时间内对拦截的标签进行解密。

密钥分配服务器是一个网络系统，所有的用户和应用服务器必须在系统上注册。它包含

了两个服务器：**认证服务器**和**权证授予服务器**。首先，认证服务器提供了允许所有用户使用各自的口令进行注册的各种管理服务。它还提供了所有Kerberos服务器的名称和秘密密钥，这些服务器包括权证授予服务器和所有的应用服务器。这些信息都保留在一个**认证数据库**中。它提供了额外的运行时服务，使用户在被允许访问任何Kerberos服务器之前通过认证成为注册用户。用户和两种服务器之间的各种交互示意图如图12-23(a)所示。

(a)



(b) K_U = 用户的私有密钥——用户口令

K_T = TGS的私有密钥

K_S = 应用服务器的私有密钥

K_{UT} = 加密UA的会话密钥↔TGS对话单元

K_{US} = 加密UA的会话密钥→S对话单元

TGS权证，$T_{UT} = K_T(U, T, t_1, t_2, K_{UT})$

应用服务器权证，$T_{US} = K_S(U, S, t_1, t_2, K_{US})$

t_1，t_2 = 权证生存期的开始和结束

(c)

	方向	信息
(1)	U ↔ UA	用户名，U
(2)	UA → AS	(U, T, n_1)
(3)	AS → UA	$K_U(K_{UT}, n_1); T_{UT}$
(4)	U ↔ UA	用户口令，K_U
(5)	UA → TGS	$K_{UT}(U, t); T_{UT}, S, n_2$
(6)	TGS → UA	$K_{UT}(K_{US}, n_2); T_{US}$
(7)	UA → S	$K_{US}(U, t); T_{US}, n_3$
(8)	S → UA	$K_{US}(n_3)$

$K_{UT}/K_{US}(U, t)$ 都是认证机构，t是一个时间戳

图12-23 Kerberos认证系统

(a) 术语和信息交换顺序 (b) 密钥和权证定义 (c) 信息内容

每个用户工作站上运行的是一个称为**用户代理**的进程，通过用户代理用户与Kerberos服务器之间的所有交互动作发生。一个用户（代理）可以访问应用服务器前，必须首先从权证授予服务器获得一个**认证标签**和一个**会话密钥**；第一个用来核实这个用户已经通过认证成为一个注册用户，第二个是用来对本次对话中要在用户代理和应用服务器之间交换的所有后续对话单元加密。注意实际上，在一次对话中可能涉及了多个应用服务器。而且，这两个密钥都具有有限的生命期，这样就可以防止注册已经过期的用户再次使用标签。

在会话开始时，用户通过用户代理（UA）提示用户名字（1）。在UA使用标签授予服务器（TGS）进行通信之前，用户必须首先通过认证成为注册用户并得到一个访问TGS的标签。所有这些都是通过认证服务器（AS）实现的。当UA接收到用户名后，它会创建一个包含用户名（U）和TGS（T）以及现时记录（n_1）的消息。UA会保存所用现时的记录并把这个消息发送给AS（2）。

此次会话中所有后续的消息都使用不同的密钥进行加密。图12-23(b)中定义了这些密钥以及组成这两个标签的成分；其中第一个授予UA权限允许同TGS进行通信，第二个使UA可以
730 同应用服务器进行通信。一个成功的会话过程中交换的消息内容如图12-23(c)所示。

接收到初始的UA请求消息，AS首先验证用户已注册，如果成功，则继续创建一个响应消息。响应消息中包含两部分，第一个部分包含一个最新生成的会话密钥 K_{UT}（它被用来对接下来的 UA/TGS 对话单元加密）和现时记录 n_1（包含在初始 UA 请求消息中）。使用用户密码 K_U（从认证数据库中获得）作为密钥对存有 K_{UT} 和响应消息的记录进行加密。第二部分包含允许标签 UA 访问 TGS 的权限标签 T_{UT}，它使用 TGS 的私有密钥 K_T 进行加密。然后，这两个部分的消息返回给 UA（3）。

接着，UA 要求用户输入他的密码 K_U（4），从而首先得到现时记录 n_1，其次得到密钥
731 K_{UT}，其中 n_1 用来验证与前面请求相关的消息。显然，一个假扮注册用户的用户无法对这个消息解密，于是他在这个步骤就失败了。UA 继续使用检索密钥 K_{UT} 创建一个**认证符**，它作为验证用户已经通过认证的令牌，包含了用户名 U 和时间戳 t，两者都用 K_{UT} 加密。对请求应用服务器 S 和第二个现时记录 n_2 进行加密生成加密权限标签 T_{UT}。然后，完整的消息发送给 TGS（5）。

使用保留密钥 K_{UT} 由 TGS 对认证符解密，由于它被同一个用户 U 授予，TGS接受它作为这个用户已经被允许获得会话密钥并与应用服务器进行通信的证据。作为响应，TGS创建一个新的会话密钥K_{US}，这个密钥被UA用来对与已命名服务器S交换的对话单元进行加密。注意，如果在会话期间访问了多个服务器，在这个步骤就要创建多个密钥，每个密钥对应一个服务器。然后 TGS 会创建一个消息，其中第一部分包含 K_{US} 和现时记录 n_2（使用 K_{UT} 进行加密），第二部分包含一个允许 UA 访问 S 的加密权限标签 T_{US}，最后整个信息传递给 UA（6）。

接收到这个消息，UA 使用 K_{UT} 解密第一部分并获得 K_{US} 和现时记录 n_2。n_2 用来证实这条消息是与前面向 TGS 请求的消息相关的，而 K_{US} 用来创建一个认证符，它用来验证用户已经得到了访问已命名服务器 S 的许可。认证符由从 TGS 得到的权限标签 T_{US} 和第三个现时记录 n_3 组成。得到的消息被 UA 发送给服务器 S（7）。

如图12-23(b)所示，权限标签 T_{US} 使用 S 的私有密钥 K_S 加密。因此，接收到这个信息之后，S 使用自己的私有密钥对 T_{US} 解密，并得到用户名 U 和分配的会话密钥 K_{US}。后者用来对认证符进行解密，证实 U 已经通过认证成为注册用户并得到访问 S 的权限。服务器返回用 K_{US} 加密的现时记录 n_3 来响应（8）。现在认证过程就完成了，UA 和 S 之间可以进行对话单元交换了。如果需要，这些对话单元用 K_{US} 进行加密。

12.5 表示层协议

虽然数据加密的主题通常都是在表示层环境中讨论，但实际的表示层协议实体只关心通过网络传输的消息的语法。回忆前面，一个消息在网络传输过程中具有的语法称为具体（或传送）语法，应用进程使用这种语法对本地语法或抽象语法进行处理。

传送语法与抽象语法之间关系构成了**表示层上下文**。因此，表示层的功能之一就是商议一个合适的表示层上下文，用于会话/表示连接。而且，因为应用实体必须通过表示层提供的
服务来使用会话层提供的多个服务，表示层的另一个功能就是，把这种服务直接映射到由会 732
话层提供的对应服务。表示层执行的功能概括如下：

- 协商一个适当的传送语法，它适合于传递交换的 PS_user 数据消息（表示层）类型。
- 把 PS_user 数据从本地抽象语法形式转换成所选择的传送语法形式。
- 把接收到的数据从传送语法形式转换成本地 PS_user 所使用的抽象语法形式。
- 把应用层服务请求，例如对话控制（令牌管理）和同步控制，映射成为会话层提供的服务原语。

可以得到这样的结论：如果在一个应用会话中交换的 PS_user 数据由不同的抽象类型组成，可选择多个不同的表示层上下文。在连接建立阶段在会话协商中使用的表示层上下文，这些不同的表示层上下文汇集称为**表示层上下文集**。另外，两个 PS_user 可以使用已经存在的**默认上下文**来进行传送，例如IA5/ISO 646。

PS_user 数据和应用层用户数据，以**标记数据元素**的形式传送到表示层，其中的标记或元素名称指明表示层上下文。如果有必要，表示层实体应对转发给会话层之前的每个元素进行相应的转换。

12.5.1 表示层服务

图12-24(a)中的时序图表示了表示层所支持的基本（核心）服务。正如所见，每个原语都带有参数。例如，P_CONNECT参数包含主叫和被叫表示（会话）地址、会话连接标识符和其他会话层有关的参数，例如令牌请求。因为表示层和会话层形成应用层的综合功能，多个的表示层服务原语及它的参数生成一个对应的表示协议数据单元（PPDU）。

服务原语	创建的PPDU	
P_CONNECT.request	表示连接CP	
P_CONNECT.response（+）	表示连接接受CPA	
P_CONNECT.response（-）	表示连接拒绝CPA	
P_DATA.request	表示数据传送TD	
P_U_ABORT.request	异常释放（用户）ARU	
P_P_ABORT.indication	异常释放（提供者）ARP	733

以上这些都如图12-24(b)所示。

另外，有些表示层服务原语可不加改变直接映射成为对应SS原语，即不需创建一个PPDU。通常，与这些原语有关的参数在SS原语的用户数据字段中传送。这包括P_RELEASE服务原语
和提供如下（任选）功能的原语： 734

- 同步控制
- 令牌控制
- 异常报告
- 活动管理

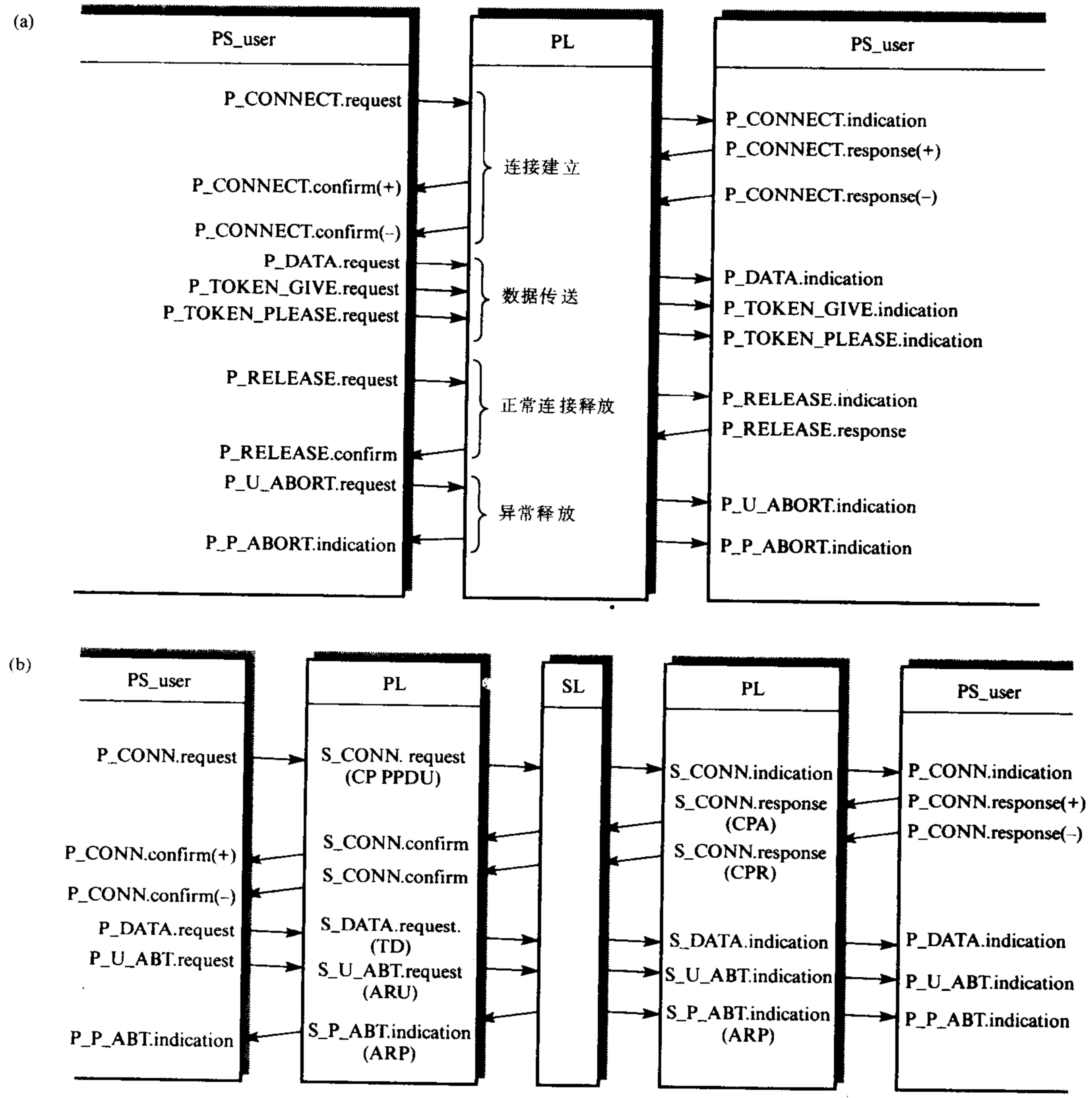


图12-24　表示服务

(a) 基本服务　(b) 有关会话服务

最后，虽然表示层为一个连接提供了一致的默认表示层上下文，但它也提供了服务原语允许两个 PS_user 对连接协商一个表示层上下文集。图12-25给出基本子集中的各种表示层服务，以及相应的PPDU和会话服务。

735

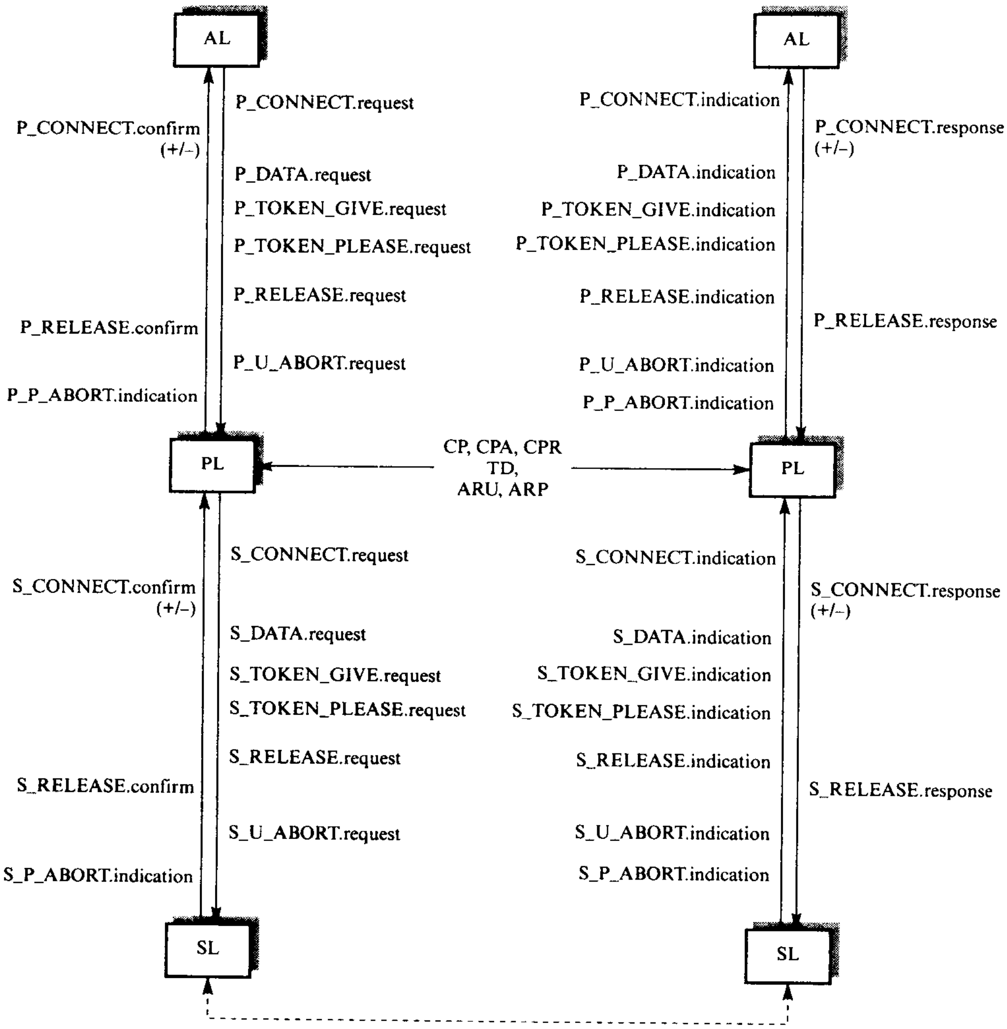


图12-25　表示层概要（只限基本服务）

12.5.2　协议规范

为了更好地理解表示层服务与相关的PPDU之间的相互关系和各种功能，图12-26给出了关于表示层协议中连接建立阶段的协议规范。如前面的例子，给出了在**表示协议机**（PPM）中的扩展事件—状态表中使用的缩写名。协议的实现遵循第11章中有关规程。 736

12.6　联系控制服务元素

回忆一下，应用层由多个协议实体组成，每个都称为一个应用服务元素（ASE）。因为有些功能是许多应用通用的，ISO 采用了这样的方法实现：把这些功能分离成独立的 ASE（协议），当请求支持服务时就把选择的特定应用 ASE 联系起来。组合后的实体被称为**应用实体**，并与用户应用进程联系起来。

为了区分执行通用应用支持服务的 ASE 和执行特定应用支持服务的 ASE，有时使用术语**通用应用服务**（CASE）表示前者，而用术语**特定应用服务元素**（SASE）表示后者。在本章接下来的部分中描述三种 CASE，而各种 SASE 将在第13章中描述。

(a)

名称	接口	含义
PCONreq	PS_user	接收到P_CONNECT.request
PCONresp(+)	PS_user	接收到P_CONNECT.response（接受）
PCONresp(-)	PS_user	接收到P_CONNECT.response（拒绝）
CP	SS_provider	接收到CONNECT PRESENTATION PPDU
CPA	SS_provider	接收到CONN.PRESENT.ACCEPT PPDU
CPR	SS_provider	接收到CONN.PRESENT.REJECT PPDU
⋮		

(b)

名称	含义
STA 10	空闲，无表示连接（PC）
STA 11	等待CPA PPDU
STA 12	等待P_CONNECT.response
⋮	⋮
STA CO	数据传送

(c)

名称	接口	含义
PCONind	PS_user	发出P_CONNECT.indication
PCONconf(+)	PS_user	发出P_CONNECT.confirm（接受）
PCONconf(-)	PS_user	发出P_CONNECT.confirm（拒绝）
PPABTind	PS_user	发出P_P_ABORT.indication
CP	SS_provider	发送CONNECT PRESENTATION PPDU
CPA	SS_provider	发送CONN.PRESENT.ACCEPT PPDU
CPR	SS_provider	发送CONN.PRESENT.REJECT PPDU
⋮		

(d)

名称	含义
P1	可接受的CN PPDU
P5	可接受的表示上下文
⋮	

(e)

名称	含义
[1]	为定义的和默认的上下文集记录抽象和传送语法
[2]	为每个上下文集选择传送语法
[3]	为每个上下文集申请传送语法
⋮	

(f)

事件 \ 状态	STA 10	STA 11	STA 12	----	STA CO
PCONrequest	1	0	0		
PCONresp(+)	0	0	2		
PCONresp(-)	0	0	3		
CP	4	0	0		
CPA	0	5	0		
CPR	0	6	0		
⋮					

0 = PPABTind, ARU, STA 10
1 = P5: CP, [3], STA 11
2 = CPA, [1], [2], STA CO
3 = CPR, STA 10
4 = P1: PCONDind, STA 12;
NOT P1: CPR, STA 10
5 = PCONconf(+), [1], STA CO
6 = PCONconf(-), STA 10

图12-26 表示协议规范

(a) 人事件 (b) 自动机状态 (c) 出事件 (d) 谓词 (e) 特定动作 (f) 事件—状态表

两个应用进程（应用实体）之间的通信，通过在数据交换前两个应用实体之间建立一条（逻辑）信道实施或者通过使用简单的请求/响应信息交换进行。两个应用实体之间的逻辑连接被称为一个联系。而在两个特定应用 ASE（SASE）之间发起建立和释放联系的ASE称为联系控制服务元素（ACSE）。

通常，联系是作为响应客户端用户应用进程访问特定网络服务（例如文件服务）的请求建立的。如将在第13章看到，为了以开放的方式执行这种服务，在每个连网（计算机）系统中，每个与服务相关的独立SASE都是在应用层中出现的。一方称为客户ASE（SASE），而另一方称为服务器ASE。

当SASE接收到来自客户端用户应用进程的初始服务请求后，它会首先创建一个自己的初始（连接接受）PDU，然后使用ACSE提供的服务同对应（被叫）SASE建立一个联系。初始 737
PDU和其他消息（例如主叫和被叫SASE地址），将作为ACSE联系请求服务原语的参数被传递。

一旦建立联系，在后续 SASE 对话期间，一直到SASE请求释放连接，ACSE不再起作用。ACSE相关的服务原语如下：

- A_ASSOCIATE.request/indication/response/confirm
- A_RELEASE.request/ indication/response/confirm
- A_ABORT.request/indication
- A_P_ABORT.indication

应用服务原语与表示服务原语之间的相互关系的时序图如图12-27所示。每个服务原语直接从一层映射到另一层（但是不显示出来），包括从表示层映射到会话层。对于所有面向应用的层都有一个连接标识符。 738

A_ASSOCIATE服务中的参数包括如下：

- 主叫和被叫应用实体标题：用全系统惟一的名称识别OSI环境中每个用户AP或AE。
- 主叫和被叫表示地址：与每一个AE相关的完全限定地址（P/SAP+TSAP+NSAP）对应。
- 应用层上下文名称：FTAM、JTM等等。
- 表示层上下文信息。
- 通信的服务质量（QOS）。
- 连接标识符。
- 会话请求：子集、初始令牌分配，等等。
- 用户数据：一般SASE启动PDU。

ACSE协议机（实体）从SASE中接收到各个服务原语时，创建一个相应的PDU。这个PDU加上服务原语的参数，作为一个相应的表示层服务原语的用户数据参数，传递到相应的ACSE实体。ACSE协议实体的PDU以及相应的表示层服务原语，如图12-27所示。

用户AP使用名称或标题进行通信。因此，在启动一次特定服务之前，用户AP必须获得通信双方AE的相应完全限定地址。这些是从本地目录服务代理（DSA）得到的，将在第13章讨论。接下来的对其他SASE的服务请求会把名称和两个AE的相应地址作为参数。

最后，为了增强对ACSE协议机的操作的理解，图12-28给出了ACSE协议机中联系建立阶段的形式化协议规范。如前面的例子，图中给出了在扩展事件—状态表中使用的缩写名列表 739
以及入事件、状态、出事件和谓词的定义。ACSE协议的实现遵循第11章中描述的规程。

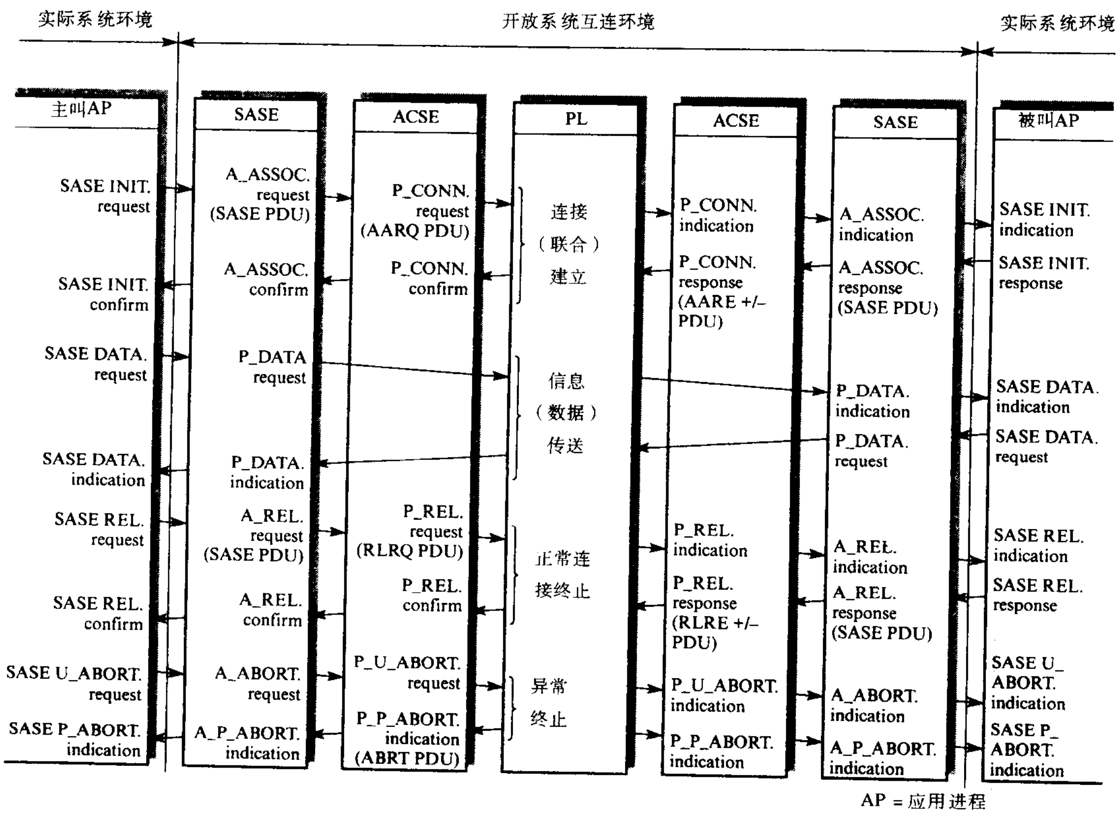


图12-27 服务原语间相互关系

12.7 远程操作服务元素

如将在第13章看到的，多数的特定应用ASE都以面向连接的方式操作：建立一个联系，然后在适当的事务已经被执行后，释放联系。对于每个联系，两个 ASE 保留的状态信息是很可观的。另外，少数 ASE 使用短小的请求/响应消息交换来执行操作，这种情况下的每个 ASE 保留的状态信息能达到最小。相关的支持ASE被称为**远程操作服务元素**（ROSE），它就是被定义用来支持这类应用的。

这种类型的通信从语义学的角度考虑显然与面向连接的通信方式是不同的。后者等价于
740 两个程序彼此之间进行通信，而ROSE用法等价于呼叫进程，调用一个（远程）规程执行特定操作或功能。这种通信方式称为**远程规程呼叫**或ISO术语中的**远程操作**。输入参数/变量是与每个调用请求相关的，可能返回结果参数。每个操作执行以后没有状态信息要保留，每个新的请求都作为一个独立实体处理。

图12-29表示了ROSE有关的用户服务原语以及在两个ROSE协议实体之间交换的PDU。

一个AP通过相关SASE使用RO_INVOKE服务原语调用对等AP上的远程操作，包含与这
741 个操作（规程）相关的变量作为参数。虽然只显示了单个INVOKE请求，但实际上在接收到响应之前可以发送多个请求。实际上，在某些情况下一个操作可能没有返回结果。

(a)

名 称	接 口	含 义
AASCreq	CS_user	接收到A_ASSOCIATE.request
AASCresp(+)	CS_user	接收到A_ASSOCIATE.response（接受）
AASCresp(−)	CS_user	接收到A_ASSOCIATE.response（拒绝）
AARQ	PS_provider	接收到AARQ PDU
AARE(+)	PS_provider	接收到AARE(+) PDU
AARE(−)	PS_provider	接收到AARE(−) PDU
PCONconf(−)	PS_provider	接收到P_CONNECT.confirm（拒绝）
¦		

(b)

名 称	含 义
STA 0	空闲（未联系）
STA 1	等待AARE PDU
STA 2	等待A.ASSOCIATE.response
¦	¦
STA 5	联系的

(c)

名 称	含 义
P1	CMP能支持连接
¦	

(d)

名 称	接 口	含 义
AASCind	CS_user	发出A_ASSOCIATE.indication
AASCconf(+)	CS_user	发出A_ASSOCIATE.confirm（接受）
AASCconf(-)	CS_user	发出A_ASSOCIATE.confirm（拒绝）
AARQ	PS_provider	发送AARQ PDU
AARE(+)	PS_provider	发送AARE(+) PDU
AARE)-)	CS_user	发送AARE(-) PDU
AABRind	CS_user	发出A_ABORT.indication
ABRT	PS_provider	发送ABRT PDU
¦		

(e)

事件 \ 状态	STA 0	STA 1	STA 2	---
AASCreq	1	0	0	
AASCresp(+)	0	0	2	
AASCresp(−)	0	0	3	
AARQ	4	0	0	
AARE(+)	0	5	0	
AARE(−)	0	6	0	
PCONconf(−)	0	6	0	
⋮				

0 = AABRind, ABRT, STA 0
1 = P1: AARQ, STA 1
2 = AARE(+), STA 5
3 = AARE(−), STA 0
4 = P1: AASCind, STA 2;
NOT P1: AARE(−), STA 0
5 = AASCconf(+), STA 5
6 = AASCconf(−), STA 0
⋮

图12-28 ACSE协议机的缩写名称

(a) 入事件 (b) 自动机状态 (c) 出事件 (d) 谓词 (e) 事件—状态表

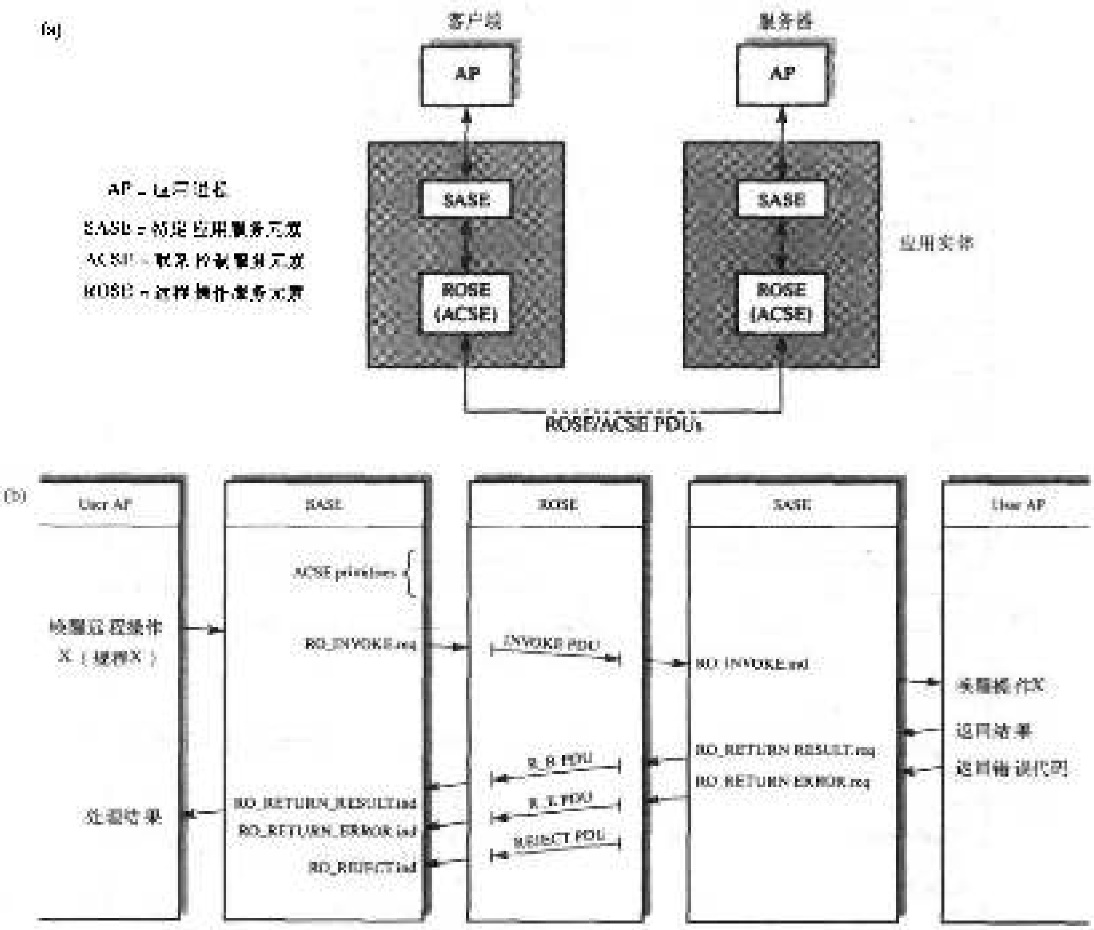


图12-29 ROSE示意图和相关的服务原语/PDU

RO_RETURN_RESULT报告了操作的成功完成，通常还伴随一个结果变量。然而不是所有操作都会返回一个结果。如果需要一个结果，但是操作却没有成功，就会返回一个RO_RETURN_ERROR原语（和相关的PDU）以及一个定义的错误代码号。如果ROSE协议实体接收到一个包含不可识别字段的PDU，它会返回一个REJECT。

因为ROSE是一个通常支持ASE，它会被多个ASE使用，所以对于执行的操作以及操作中的输入和输出参数的含义，在使用这些操作的AP中必须有定义，在ROSE外部。相类似，任何可能返回的错误消息必须只对这些AP有意义。四个原语携带的参数如下：

- RO_INVOKE.request/indication–InvokeID，OpCode，输入变量表；
- RO_RETURN_RESULT.request/indication–InvokeID，输出变量表；
- RO_RETURN_ERROR.request/indication–InvokeID，ErrorCode；
- RO_REJECT.indication–InvokeID，ProblemsCode。

可以执行的操作列表以及每个操作的输入变量列表都采用ASN.1定义。但是ROSE只关心它们的语法，而它们的语义只需要两个通信的AP知道。如果需要，结果（输出）变量列表也使用同样的方式定义。

为了确保通用性，每个操作都要赋予一个**操作代码号**（OpCode）。OpCode和输入变量表构成了INVOKE服务原语的参数。另外，因为在结果返回之前可能会发送多个调用，每个INVOKE原语使用了一个附加参数——**调用号**（InvokeID）。相似的是，为了使发送方AP可以把得到的结果与相应的调用请求联系起来，在RO_RETURN_RESULT原语中除了结果变量外，还有调用号参数。RO_RETURN_ERROR 和RO_REJECT原语中也包含了一个调用号参数，还有**错误代码号**和**问题代码号**。AP 会把这些代码号联系到特定的错误定义或问题定义。

作为一个例子，如果AP是（客户）目录用户代理AP，它的OpCode被设置为1，并有一个简单输入变量，可由接收的（服务器）目录服务代理AP解释，对输入参数中的名称执行地址解析操作。将在第13章看到，这个操作中涉及产生目录信息库，以获得对应给定名称的完全限定地址。每个调用请求都有一个不同的InvokeID，这个InvokeID加上结果或加上ErrorCode/ProblemCode号作为结果被返回。后者按定义方法由用户解释。 742

通常所有的ROSE PDU都使用表示层提供的服务来传递。为了反映远程操作的语义，将使用无连接的表示和会话协议。但是在本章开头，提到当前定义的OSI协议只包含面向连接的协议，所以在发送任何调用请求之前必须建立一个表示/会话连接，这是ACSE的职能。一般的应用实体由SASE、ACSE和ROSE组成。为了使开销最小，如果要使用一个无连接的网络服务，就要在会话开始时建立一个连接并一直保持开放状态。例如，当一个应用包含了多个/频繁的操作时，这就是合适的。不考虑这些，所有的ROSE PDU都使用表示层提供的P_DATA服务进行传递。

12.8 委托、并发和恢复

很多分布式应用包含若干个应用进程，而这些进程会请求访问同一个共享资源。一个例子就是银行应用系统中包含客户账号的文件系统。通常，多个客户系统会同时对不同账号执行贷方和借方操作。为了说明在这种应用中会出现的问题，考虑按图12-30中的顺序执行的两个客户系统的操作。 743

在例子中，假设客户*A*有10K（英镑、美元、任何货币）要从账号a传输给账号b，与此同时客户*B*有20K（英镑、美元、任何货币）要从账号b传输给账号c。每个客户仅从服务器读取特定的两个账号信息，执行对应的传输，然后向服务器返回更新账号。正如图中所见，客户*A*的操作覆盖客户*B*对账号b的操作。这种情况称为**丢失更新**。如果没有适当的（并发）控制机制，对单一共享资源并发访问的所有应用都会出现这种丢失更新。

相关的问题发生在一个文件的多个拷贝保存在不同地方的时候。如果所有备份不完全相同（一致性），类似的银行应用中就会发生一个客户已经在上一次的提款中提光了他的存款，但是现在这个客户在其他地方还可以继续提款。这种问题称为**多备份更新问题**。以上两个问题对于多数的分布式应用系统来说都是很常见的，通过定义**委托、并发和恢复**（CCR）ASE来帮助控制这些操作。

CCR是一个支持ASE，它提供的服务既可以直接使用AP，也可以通过相关SASE服务作用于AP。例如，称为分布事务处理的ASE用户服务，包含可直接映射成CCR同名原语的附加原语。

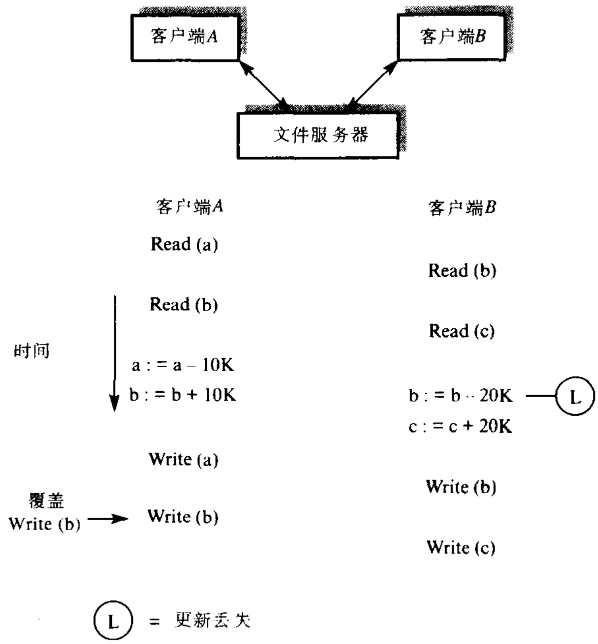


图12-30　丢失更新示意图

CCR是建立在**原子操作**的概念基础上的，包含**两阶段委托协议和回退差错恢复**。本质上，原子操作是一个操作序列，它由两个或多个AP协同操作：

- 执行操作序列不受非原子操作AP的干扰。
- 包含在原子操作中的每个AP执行的操作，或者全部成功，或者全部被终止，并将在操作过程中修改的数据全部恢复到原子操作开始之前的状态。

发起原子操作的AP直接或间接地控制与原子操作相关的全部活动，因而称为**主控AP**。同样，包含在原子操作中的所有其他AP受发起原子操作的主AP控制（采用CCR协议），因而称为**被控AP**。

受原子操作影响的所有数据都称为**约束数据**。在原子操作开始时，所有约束数据的值称为**初始状态**，当原子操作结束时，约束数据的值称为**终止状态**。当一个原子操作结束后，如果原子操作执行成功，则与原子操作相关的约束数据进入终止状态；如果有一个或多个操作失败，则仍回到初始状态。接下来的特定操作是使用**握手规程**来完成的：当原子操作中所有
744 的操作执行后，主控AP首先询问所有被控AP是否成功地完成它们的处理；于是它们都处于"准备递交"状态。如果所有被控AP都作出肯定的响应，则主控AP发出指令使所有约束数据进入终止状态。如果有一个或多个被控AP作出否定的响应，或响应错误，则主控AP发出指令，恢复所有约束数据到初始状态。各种术语如图12-31所示。

要注意，受控制的原子操作的特性是由AP和相关的SASE决定的，而SASE处理原子操作
745 中包含的约束数据，CCR协议仅对AP提供发起并控制原子操作的方法。就是说，对于原子操作中包含的AP来讲，设法保证原子操作所遵循的规则，纯粹是本地事务。因而，如果一个AP试图加入一个原子操作中，则对该AP而言，确保未包含在该原子操作中的AP不能访问或处理该原子操作的相关约束数据，也是它的本地事务。通常，这是采用**信号灯**的**锁定机制**实现的，

信号灯是令牌的一种形式，一旦信号灯被设置之后，所有对该信号灯控制的数据的其他访问都要被排斥在外，直到该信号灯复位。它与并发编程语言（例如Ada）中各种形式的互斥机制是相似的。

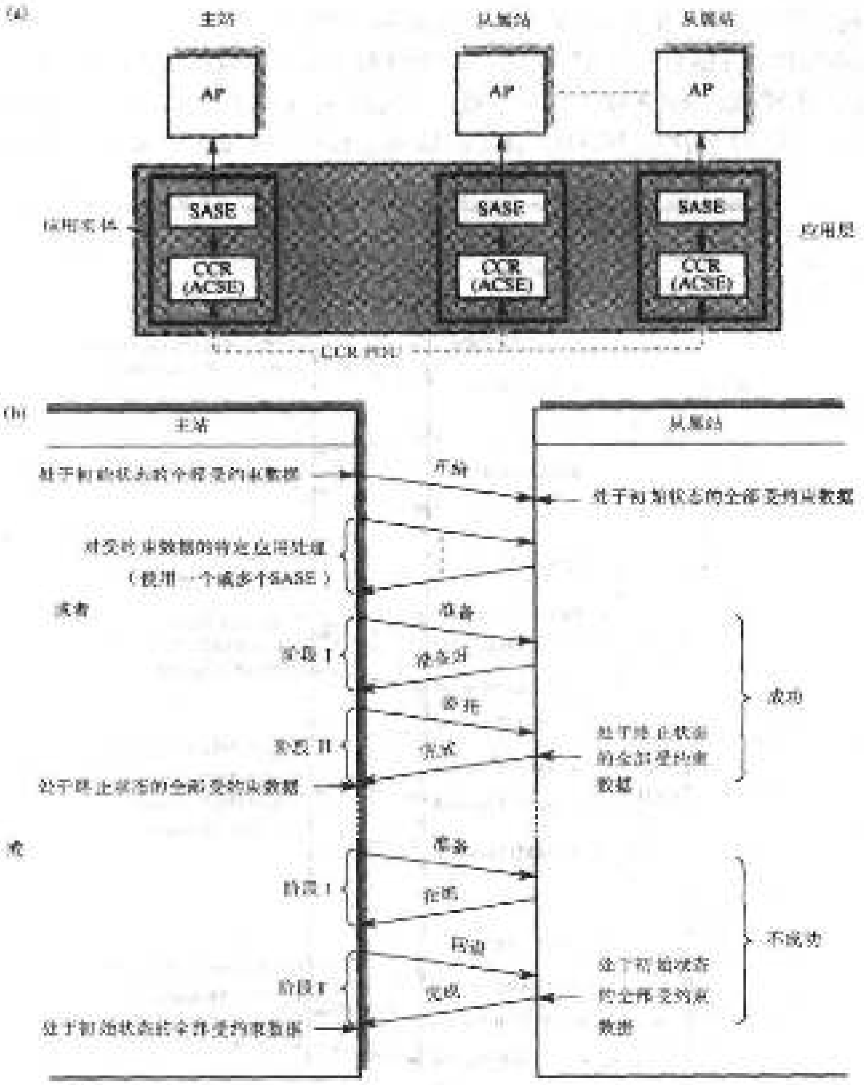


图12-31 原子操作原理

(a) CCR示意图 (b) 两阶段委托序列

用户服务

设计一组提供给CCR的服务原语。于是CCR控制的原子操作就受到下列条件限制：

- 当主控给出委托命令之后，主控和被控者有的约束数据处于终止（永久）状态，不支持允许任何数据返回到它的初始（或中间）状态的恢复能力。
- 发出委托命令之前，主控可在任何时候命令回退到初始状态。

- 主控不能命令被控委托，除非它收到了该被控愿意委托的表示。
- 一旦被控向主控发出愿意委托的表示，该被控就不能拒绝委托命令。
- 在返回委托表示之前，被控可在任何时刻中止原子操作，因此，所有约束数据返回到它们的初始状态。
- 如果被控拒绝发出委托表示，则主控可以命令其回退。

CCR服务原语时序图如图12-32所示。一个原子操作中可以包含多个被控。并且在启动原子操作之前，主控AP（和SASE）与每个被控AP必须存在一个联系。通常，这种联系使用ACSE建立。主控AP使用C_BEGIN服务通知每个被控，开始原子操作。当被控接收到

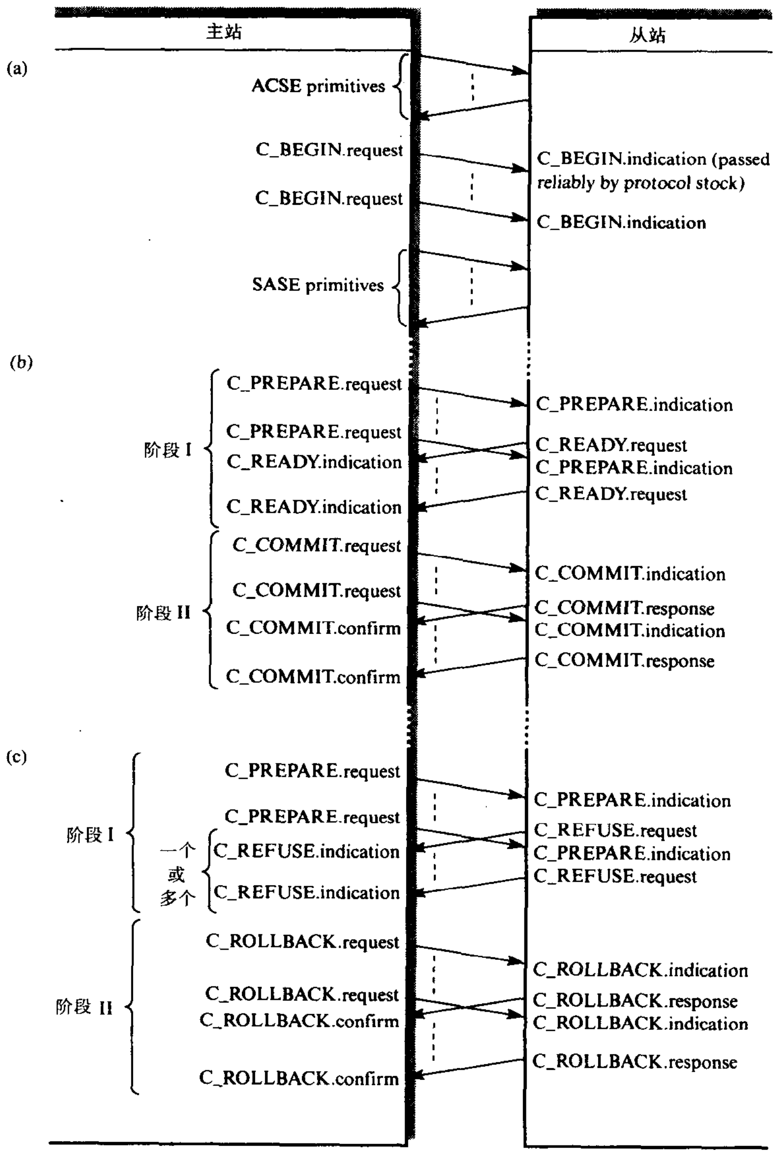


图12-32 CCR服务原语

(a) 建立 (b) 成功 (c) 不成功

C_BEGIN.indication原语后，必须创建一个包含所有约束数据值的新的实例（工作拷贝），例如从磁盘上读取一个文件的拷贝，并启动锁定机制。实际上，C_BEGIN原语通过这个联系影响到会话层主同步点的建立。

接下来，原子操作中使用SASE原语执行特定应用处理。图12-32(b)给出了两种可能的终止过程。第一个，所有的被控使用C_READY服务返回一个肯定的准备接受委托表示。第二个，746
一个或多个被控使用C_REFUSE服务返回一个否定的回答。实际上，C_PREPARE服务是可选的。如果不使用C_PREPARE服务，当相关的应用处理结束后，被控立即使用C_READY或C_REFUSE表示其是否准备接受委托。另外，还有一个证实服务，即C_RESTART服务，它既可用于主控，也可用于被控，用于被控时，它必须给出包含在原子操作中的所有约束数据的 747
早先已知值。通常，被控用它表示未准备好接受委托，而主控在被控响应超时时，多用它表示一个可能的应用故障或通信故障。

将更明确地阐述几种使用RESTART服务会发生的情况，考虑原子操作中几种可能发生的故障情况会引发的相应RESTART服务。在图12-33的时序图中，给出几个实例。虽然实例中只是COMMIT服务的故障，但是ROLLBACK服务的故障是与之等价的。然而，从实例中知道当故障发生之后，被控的准确状态是不可知的。例如，图12-33(a)表示的通信故障，是发生在接收到C_COMMIT请求之前还是之后。提供的RESTART服务允许主控把原子操作返回到早先已知状态，而不必考虑是因为何种原因。RESTART服务是一种证实服务，只要原子操作是完整无损的，就可以在任何时候使用。

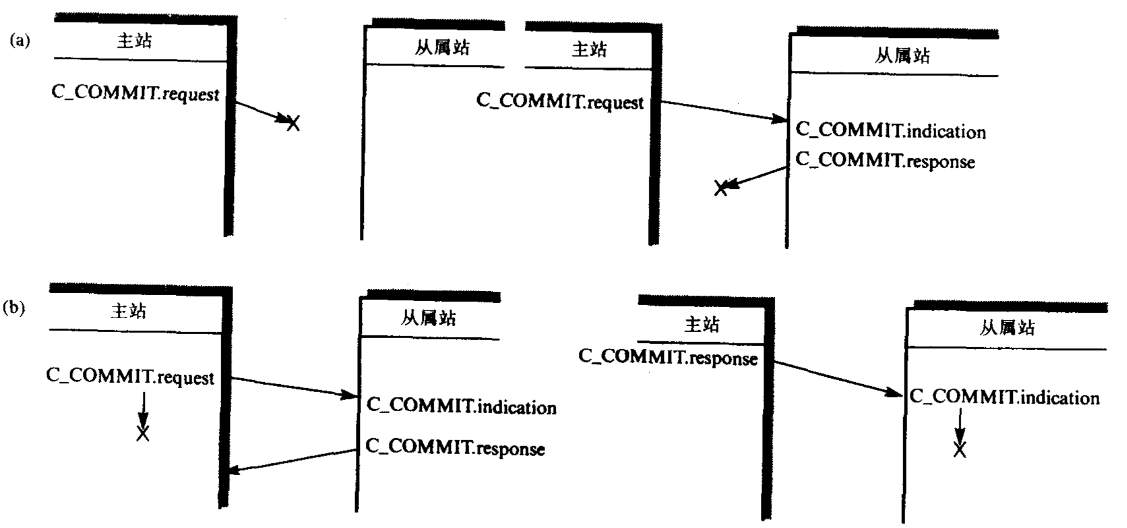


图12-33 某些故障实例

(a) 通信 (b) 应用

主控通过发送一个带有**恢复点**参数的C_RESTART.request原语来确定被控的当前状态。请求（和指示）原语中的恢复点包括如下几种：

- ACTION（操作）：用于请求被控从开始处重新启动原子操作。
- COMMIT（委托）：用于请求被控在上一个COMMIT点之后，重新启动原子操作。
- ROLLBACK（回退）：用于请求被控在上一个ROLLBACK点之后，重新启动原子操作。

同样，响应（和证实）原语中的恢复点包括如下几种：

748 • DONE（完成）：表示被控完成上一次委托或回退请求。

• REFUSE（拒绝）：表示被控没有执行上一次委托请求，因此主控要发送一个回退请求。

• ACTION（操作）：表示主控要从开始处重新启动原子操作。

• RETRY-LATER（稍后重试）：表示被控现在不能执行重新启动，主控应稍后再试。

当接收到一个委托命令（C_COMMIT.indication），被控必须把约束数据从初始状态改成终止状态，即改为当前工作拷贝的内容，并把它们写到磁盘。在写入过程，需要不断进行校验，保证最后写入成功，随后应被控返回一个肯定的C_COMMIT.response原语。然后，可称数据是安全的或稳定的。一种实现方法是写入两份数据稳定拷贝到磁盘，然后读出做比较，如果相同才认为数据是安全的。

各种CCR原语中都有相关的参数。例如，两个C_BEGIN原语中的参数包括如下：

• 原子操作标识符：包括主控AP的名称（标题）和一个后缀，允许多个原子操作并发进行。

• 原子操作定时器：表示主控在发出回退请求之前等待的时间间隔。

对于CCR协议，一个原子操作的被控可能是另一个原子操作的主控，并由此构成一个树结构。因此，使用另外一个参数分支标识符，用来在层次中识别特定分支，由主控（单个）赋予原子操作标识符的相关原子操作。分支标识符是树中特定（子）原子操作的主控名与分支后缀。通常CCR树如图12-34所示。

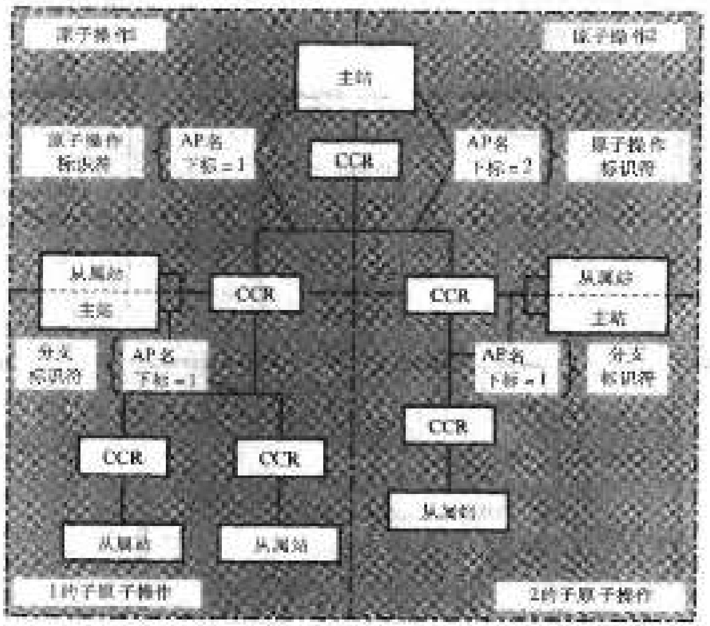


图12-34　CCR原子操作树示例

当创建所示类型的原子操作树时，必须注意避免可能的死锁发生。当原子操作中的某个AP向其他AP发送请求时，就可能会发生死锁：如果它在等待来自另一个AP的响应，而该AP正在等待已经锁定在第一个原子操作中的AP的响应，所以该AP不能返回响应命令，于是死锁发生了。下面会给出一个简单的说明。

在一个分布式信息处理服务中包含三个AP（A、B和C）。首先，假设A是原子操作的主控

AP，B和C是被控AP，并且A向B发出数据请求。在B向A返回响应之前，如果B首先必须与C形成一个原子操作，并向C请求数据。现在，系统死锁，A在等待B，B在等待C，而C的资源又被A锁定（约束）。该系统虽然不能克服这种死锁，但在这种情况下原子操作中的定时器参数是非常有用的，因为它可以确保一个AP不会无限期地等待另一个AP的响应器。 749

12.9 可靠的传输服务元素

将在第13章看到，当考察应用服务元素时，有些应用程序会在单一的应用实体中使用两个或多个已经描述的服务元素（ACSE、ROSE和CCR）。于是引入了第四个应用支持服务元素——**可靠传输服务元素（RTSE）**，它定义为两个独立的服务元素的有效组合。RTSE的目的就是向用户（一个特定应用服务元素）提供单个消息（APDU）或一串消息的可靠传输方法。

当全部表示层服务集不能提供足够的服务时，就需要RTSE。因此RTSE使用ACSE服务的组合，它是会话层服务的一个小的子集。后者可以作为表示层的穿越服务访问。使用RTSE的服务原语以及ACSE/表示层服务，每个映射如图12-35所示。

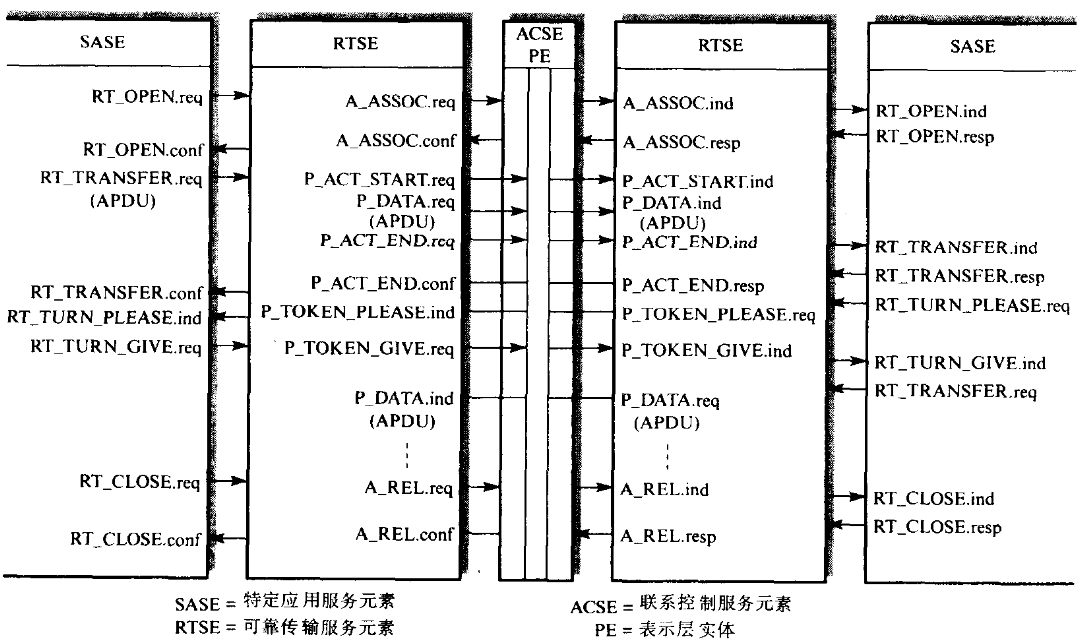


图12-35　RTSE服务原语和映射操作

为了可靠地发送每个消息，RTSE使用会话层提供的活动和数据令牌服务的组合通过发送（停止）—等待协议工作。一旦建立联系，用户通过使用RT_TRANSFER.request原语发送一个消息（APDU）。RTSE使用会话层提供的活动服务确保这是可靠的传递。因而，在APDU发送前，RTSE用信号通知开始一个活动，因为P_ACT_START是一种未证实服务，RTSE后面紧 750
跟一个带有APDU的P_DATA.request作为用户数据参数。然后，为了确保传递的可靠性，要紧接着设置一个同步点并结束这个活动。接收到证实原语后，就可以指明数据的确已经可靠地接收了，于是RTSE会向用户发送一个RT_TRANSFER.confirm原语。

如果用户打算发送APDU序列，那么RTSE不会每次都关闭活动，它会发送一个P_SYNC_MINOR到每个APDU检查点。只有当所有APDU都发送后，RTSE令发信号停止活动。

当需要响应时，RTSE会使用RT_TURN_PLEASE/GIVE原语向另一方发送一个数据令牌。这些原语会映射成为P_TOKEN_PLEASE/GIVE原语。然后，RTSE使用RT_CLOSE服务释放联合。

习题

12.1 画图表示本章中讨论的各种应用支持协议的相互关系，并解释各协议的功能。

12.2 描述会话层各相关术语的含义：

(a) 令牌

(b) 活动

(c) 对话单元

(d) 同步点

(e) 基本组合子集

12.3 借助于时序图说明在会话层的基本组合子集中的用户服务原语。在图中表示为了实现每
751 个用户服务，每个会话协议实体产生的SPDU。

12.4 使用第11章介绍的实现方法展示会话层协议机的连接组件如何用结构化程序代码形式表示。使用图12-8中给出的规范说明以及在程序中使用与图12-8相同的变量名。

12.5 描述如下有关表示层术语的含义：

(a) 抽象语法

(b) 具体/传送语法

(c) 表示上下文

每个术语给出一个例子。

12.6 对下列数据类型给出ASN.1抽象类型定义的例子：

(a) Boolean

(b) Integer

(c) Bitstring

(d) 字符串

12.7 解释ASN.1中的术语"隐式"、"显式"和"标记"的含义。利用在习题12.6中定义的类型给出一个序列类型定义的例子。修改定义，使之包含特定上下文标记和特定应用标记。

12.8 定义与ASN.1编码类型相关的组成标识符字节的类、类型与标记的含义。利用在习题12.6定义的数据类型赋值说明如何对每个类型编码。

12.9 利用从习题12.7中得到的两个序列类型定义表示序列类型的编码过程，以及使用标记而导致的附加开销。清楚识别你的编码例子中的特定上下文标记。

12.10 借助图解释数据加密中如下术语的含义：

(a) 明文

(b) 密文

(c) 监听

(d) 伪装

(e) 加密和解密密钥

12.11 简洁地描述替换密码与变换密码的不同。定义每种情况下使用的密钥。

12.12 设计一个由3到8解码器，直接8位置换（P盒）和8到3编码器组成的S盒。定义你的设计中的密钥。

12.13 设计一个基于乘积密码的6位值的编码单元。在设计中要使用3步，并包含习题12.12中设计的S盒。指定你的设计中的密钥。

12.14 借助于逻辑框图描述DES算法的如下操作模式：

(a) ECB

(b) CBC

(c) CFM

12.15 利用一个例子解释RSA算法的操作原理，以及如何求公共密钥和私有密钥。在你的例子中使用素数3和11作为原始数字，E设置为7。

12.16 解释术语"消息认证"和"数字签名"的含义。说明如何利用RSA算法获得消息认证。

12.17 区别术语"应用服务元素"和"应用实体"。利用时序图说明ACSE用户服务原语和表示层用户服务原语的相互关系。

12.18 解释术语"远程操作"的含义。列出远程操作服务元素（ROSE）中相关服务原语，并解释每个原语的参数的含义与用法。 752

12.19 利用一个例子，解释术语"丢失更新"和"多备份更新"的含义。解释具有差错恢复的两阶段委托协议的操作原理。描述中要包括如下术语的含义：

(a) 主控和被控

(b) 约束数据和安全数据

(c) 初始状态和终止状态

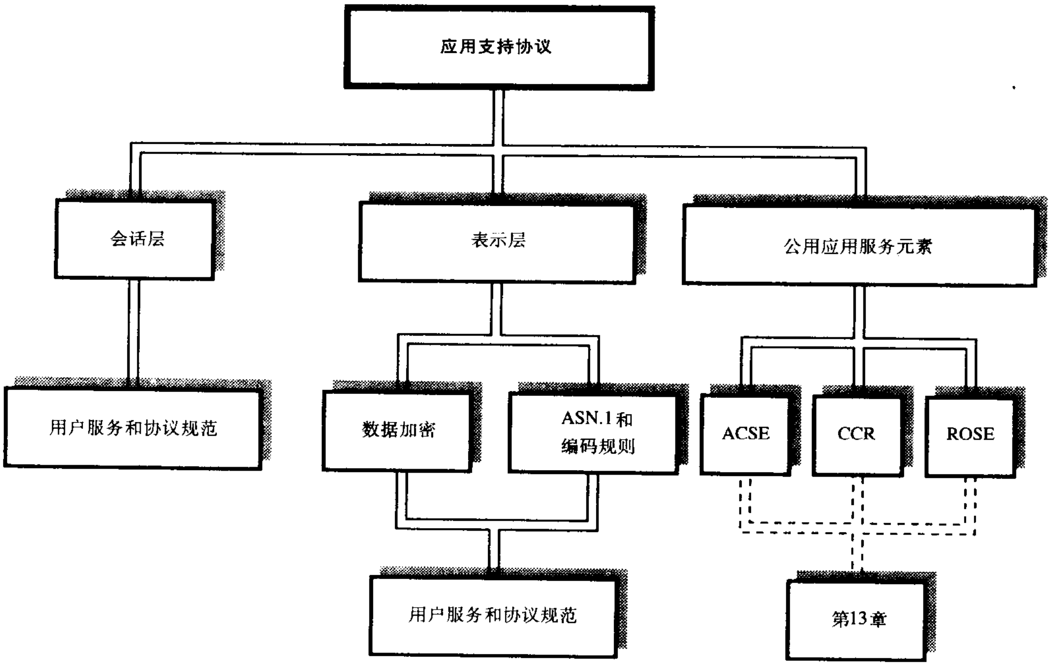
(d) 两阶段委托

(e) 回退恢复

12.20 假设一个试图对远程文件执行原子操作的应用进程，求出表示一个成功的和一个未成功的原子操作的CCR原语序列的时序图。 753

754

本章概要



应用支持协议
会话层
表示层
公用应用服务元素
用户服务和协议规范
数据加密
ASN.1和
编码规则
ACSE
CCR
ROSE
用户服务和协议规范
第13章

第13章　特定应用协议

本章目的

读完本章，应该能够：

- 了解TCP/IP和OSI协议族中已定义使用的主要特定应用协议的功能；
- 描述如下TCP/IP应用协议的服务和操作：

TELNET

FTP

SMTP

SNMP

- 描述等价的ISO应用协议的服务和操作：

VT

FTAM

MOTIS

CMISE

- 描述如下附加ISO应用协议的服务和操作：

MMS

JTM

DTP

引言

第12章描述了一些对执行特定应用功能的协议提供常规支持服务的ISO协议。执行特定功能的协议包括远程文件访问协议，电子邮件协议等等。正如所示，TCP/IP协议族并没有支持协议。而是如果需要特殊服务，TCP/IP协议族直接把特定应用协议整合起来。所以在TCP/IP 755
协议族中，执行特定功能的协议与传输协议直接通信。

在此，术语“应用进程”指的是执行一个（分布式）应用实际处理功能的程序/软件。例如，一个客户应用进程（AP）可能是一台机器上运行的程序，它需要访问运行在另一台机器上的文件服务器AP。为了使客户能够以开放的方式访问各种服务器AP，特定应用协议提供必要的支持，客户好像与服务器运行在同一台机器上。两个协议族的常规方案如图13-1所示。

在OSI协议族情况下，方法是对每个分布式应用服务定义一个**虚设备**以及一组用户服务原语。有虚文件存储器/服务器、虚邮件服务器等等。假定客户和服务器AP使用原语通信，这些原语同定义的虚设备原语相同，从而实现特定服务相关的应用协议。这种方法，在每个应用 756
协议接口以及开放系统互连环境的接口，所有请求和响应原语都具有标准格式。而且，每个应用协议以定义的方式操作，不会因为相关的实际AP操作模式的不同而必须改变。如果与AP相关的服务不同于虚设备中的服务，就要在每个AP和相关应用协议之间使用一个附加的依赖于设备的软件层，执行映射功能，把虚设备的服务原语转换成为实际设备的服务原语或反之。

于是，可以不加修改地使用已有的应用和服务软件。这种映射软件称为**用户元素**（UE）或**用户代理**（UA），一般方案如图13-1(a)所示。

实际上，UE是以一个独立的进程或链接到用户AP的一组库函数的形式实现的。对于多数服务，其中都有一个发起者AP（以及连到其上的应用实体）和一个响应者AP。前者作为虚设备请求发起者，而后者作为虚设备请求响应者。正如第12章描述，当两个用户AP之间交换的数据语法不同时，表示实体执行传送（具体）语法到本地语法的映射。当首次建立连接时，使用经协商的特定表示上下文。

可是，在TCP/IP协议族情况，除实现与定义的应用服务相关的协议外，每个应用协议还要执行其他支持功能，包括必要的映射操作。正如图13-1(b)所示，大多数协议提供了一个用户终端接口，以及用户AP接口。通常，TCP/IP应用协议与ISO应用协议相比相对复杂。本章先讨论部分TCP/IP应用协议，再讨论ISO应用协议。

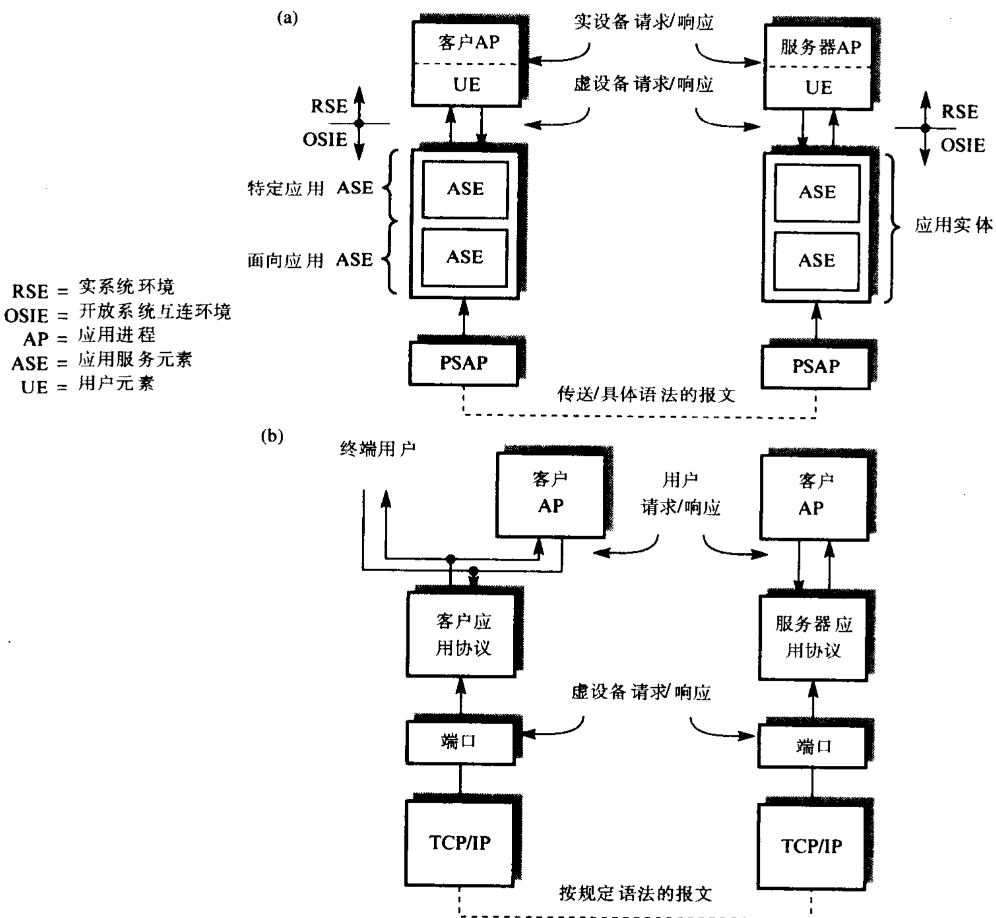


图13-1　应用层示意图
(a) OSI协议族　(b) TCP/IP协议族

13.1　TCP/IP应用协议

在TCP/IP协议族中选择的部分应用协议如图13-2所示。

- **TELNET** 使一台机器上的终端用户（或用户AP）可以与一个AP交互通信，如远程机器上运行的文本编辑器，好像用户终端是直接连接到远程机器上的；
- **FTP**（文件传输协议） 使终端用户（或用户AP）可以访问远程文件系统并同远程文件系统交互操作；
- **SMTP**（简单邮件传输协议） 在不同计算机的邮件系统之间提供网络范围的邮件传输服务； 757
- **SNMP**（简单网络管理协议） 使用户（例如网络管理者）可以收集性能数据或通过网络本身控制网络元素（如网桥或网关）的操作。

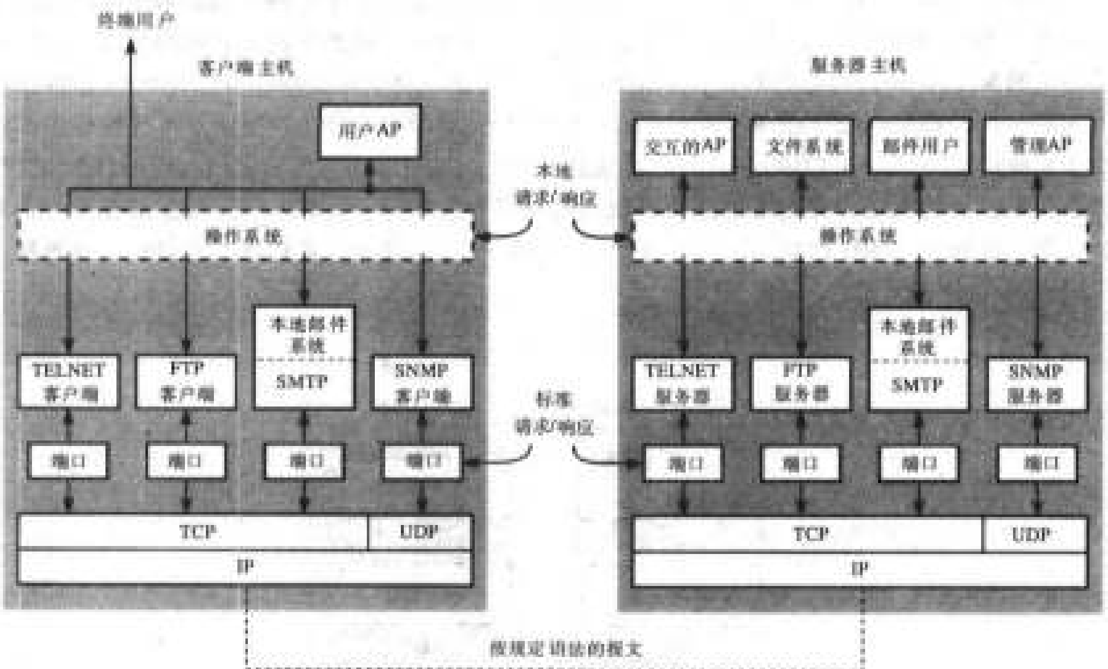


图13-2 TCP/IP应用协议概要

对所有的客户服务器交互的共同要求是在两个应用协议/进程之间建立一个通信路径，所以在讨论各种应用协议之前，要先描述如何建立一个通信路径。

13.1.1 建立一个传输连接

回忆第11章，所有的服务器AP都有一个相应的名称，并转换成一个相应的网络地址。转换规程是由**域名服务器**进程执行的（在第14章讨论该服务器的操作，以及其他系统功能）。服务器进程的网络地址包含两个部分：进程在其上运行的主机的网络IP地址和本地端口号。其中IP地址由每个互联网网关的IP使用，把数据报通过互联网传递给所要求的目标主机，端口号由主机内TCP或UDP用来确定收到的消息要传递给该主机上的哪个特定进程。 758

一个开放系统包含多个客户端和服务器，它们可以是不同类型的（FTP、SMTP等等），也可以是相同类型的（多文件服务器、邮件服务器等等）。然而，所有相同类型的服务器都赋予相同系统范围内的端口号，然后通过正运行的主机的IP地址确定特定的服务器。不同服务器类型的端口号称为**知名端口**，包含如下：

21 - FTP

22 - TELNET

25 - SMTP

因此，当一个客户端进程向一个对应服务器进程发起通信时，它要使用该服务器运行的主机IP地址和服务器的知名端口号作为目标地址，并且要把自身主机的IP地址和该主机上下一个空闲（未使用）端口号作为源地址。如果使用的是TCP，则本地TCP实体将要使用上述地址在客户和服务器之间建立一个传输连接，在这个连接上发生消息的交换。

13.1.2 TELNET

如图13-3所示，用户AP或更常见的终端用户通过本地操作系统访问TELNET客户端的协议/进程。TELNET客户端的协议/进程提供用户可以登录远程机器上操作系统的服务，执行该
759 机器上的某个程序/进程（例如，文本编辑器），并与该程序交互操作，好像用户终端/进程运行在同一台机器上。用户终端输入（或用户AP提交）的所有命令（控制字符）和数据由本地操作系统传递给TELNET客户端进程，再通过TCP提供的可靠流服务传递给对应的TELNET服务器。而后者代表用户通过本地操作系统向交互进程发送命令，TELNET服务器也称为伪终端。交互进程输出的任何数据信息都以同样的方式返回，或者在客户终端上显示，或者被用户AP解释。

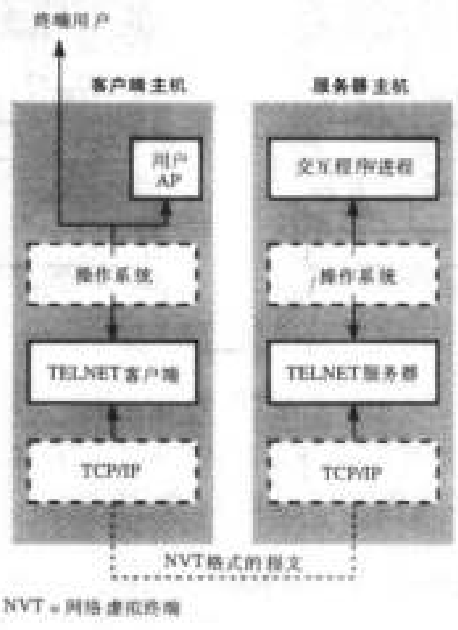


图13-3 TELNET客户端/服务器交互示意图

两个TELNET协议彼此之间使用命令进行通信，命令由一个或者一串字符组成，它使用一种称为**网络虚拟终端**（NVT）的标准格式编码，而命令使用的符号集是ASCII，所有与交互有关的输入和输出数据都要转换成ASCII字符串。如果本地使用的字符集不是ASCII，那对应TELNET要执行必要的映射功能。因此，两个TELNET协议实体还要执行OSI栈中表示层的某些职能。

回忆第3章，ASCII字符集使用7位二进制数表示。然而，NVT格式中的所有命令和数据都

使用8位字节进行编码。如果字节的最高有效位为0，那么所有的字符都是ASCII常规字符，包括ASCII控制字符。也可以选择一对字符，例如CR-LF表示一个新行。另外，如果把字节的最高有效位设置1，就定义了一组额外的命令字符集。因为这些命令字符的最高有效位是1，它们可以很容易地与ASCII标准字符集区分开来。如果需要，这些命令字符可以配置上按特定方向流动的数据流。表13-1给出已定义的命令字节的一部分，及相应的含义。

表13-1 NVT命令类型和含义

名　称	十进制代码	含　义
IAC	255	下一个字节作为命令字节
NOP	241	没有操作
EC	247	擦除字符
EL	248	擦除行
GA	249	继续
AYT	246	对方是否还在运行
IP	244	中断进程
AO	245	异常中止输出
BRK	243	中断（停止）输出
DMARK	242	标记输出
SB	250	选项协商字符串的开始
SE	240	选项协商字符串的终止
WILL	251	同意/请求选项
WON'T	252	拒绝选项请求
DO	253	接收请求选项
DON'T	254	拒绝接受选项请求

表中命令以全1（255）的IAC字节开始。如果一个命令由多个字节组成，该命令的字节串必须以SB命令开始，并以SE命令结束。正如所见，有些命令与多数的交互软件所用命令具有相同的含义。而之所以需要另外一些命令是因为交互软件（进程）运行在远程机器上。例如，多数的交互软件允许用户输入规定控制字符或控制字符对，把长字符串输出到终端屏幕上。BRK和DMARK命令字节就是用来实现这种功能的：BRK用来停止输出而DMARK用来恢复。AO命令用来对当前的输出进行异常中止。另外，如果当已经请求停止输出之后，输出进程仍然继续输出数据，例如，当命令字节被网络拥塞阻断时，这时用户通常会输入一个字符序列终止（远程）进程。如果出现这种情况，TELNET客户端使用TCP服务原语URGENT发送一个IP命令字节。回忆该原语所携带的数据是在当前连接的流量控制窗口之外发送的。因而，这个IP命令总会被服务器接收到。 760

正如指出的那样，所有的数据通信都是以7位的ASCII字符串的形式传递的。但是，有时需要传递8位的字符串，比如传输特定的显示字符，或者在AP客户端（不是终端用户）和服务器之间传递二进制数据块的时候。为了能够传递8位的字符串，一方需要输入称为**选项协商**的命令，其他的操作功能也使用同样的方式进行请求。表13-2列出了可以使用的选项代码。

该连接的任何一方可以使用WILL、WON'T、DO和DON'T命令字节后跟表中特殊代码号，发起选项请求命令。请求接收方接受8位二进制数的一般选项命令是：

IAC SB WILL 0 SE

如果接收方TELNET同意接收，则会返回：

761 IAC, SB, DO, ‘0’, SE

如果接收方TELNET不同意接收则会返回：

IAC, SB, DON’T, ‘0’, SE

另一方面，接收方也可以发起交换：

IAC, SB, DO, ‘0’, SE

如果发送方返回：

IAC, SB, WILL, ‘0’, SE

表示它正在进行交换，或者返回：

IAC, SB, WON’ T, ‘0’, SE

表示拒绝。在二进制模式时，发送两次连续的全1（255）IAC字节，表示接收方把字节作为数据解释而不是一个命令的开始。

表13-2 NVT选项代码和含义

名　称	代　码	含　义
二进制传输	0	请求/接受转换为8位二进制
回显	1	把接收到的字符回显到发送者
状态	5	请求/回答接收TELNET的状态
定时标记	6	在返回数据流中插入定时标记
终端类型	24	远程终端的请求/响应类型
行方式	34	发送完整的行，而不是单个字符（改变运行方式）

计时标志命令执行同OSI协议栈中的会话层的同步功能相似的功能。可以从其他命令的含义推出其用法。

13.1.3 FTP

访问远程文件服务器是多数分布式应用的基本要求。在某些例子中，单个文件服务器要支持多个客户端的访问，或者同一个文件的多个拷贝保存在多个服务器上。图13-4显示了一个应用的例子。

FTP客户端可以被终端上的用户或者用户AP访问。通常，单个FTP客户端可以同时支持多个用户。它对每个用户都提供了一组相似的服务并且对于多数的文件系统这些服务都是有效的。用户可以列出文件目录，创建新的文件，获得（读取）已有文件的内容，对文件执行更新操作，删除文件等等。同样，FTP服务器可以同时对多个客户端的请求作出响应。当接收到一个请求后，FTP服务器同本地文件系统交互完成该请求，就好像请求是来自本地的。

FTP客户端允许用户指定文件的结构，以及文件中的数据类型。这些服务对于多数的文件系统都是有效的。支持3种文件结构（无结构、有结构（记录结构）和随机访问（页面结构））和4种数据类型（8位二进制、文本（ASCII和EBCDIC）和可变长度二进制）。FTP服务器访问本地文件系统中的每一个文件，并根据该文件定义的结构以适当的方式把它传递给FTP客户端。

无结构文件可以包含任意类型的数据（二进制或文本）。这些数据作为透明的二进制数据流在两个FTP协议实体之间传递。用户根据数据的类型对二进制数据流进行解释。

结构文件由一组固定长度的记录组成。因此，这种文件的内容通常以固定长度的字符串

762 形式传递。另外，内容可以压缩形式传递，这适合文本文件，因为文本文件常常包含相同字符组成的长串，例如空格。如果每个FTP协议实体在传送之前使用一致的压缩算法对文件内容

字段压缩，这些冗余就可以被消除掉。在OSI协议栈中，这个功能由表示层执行。

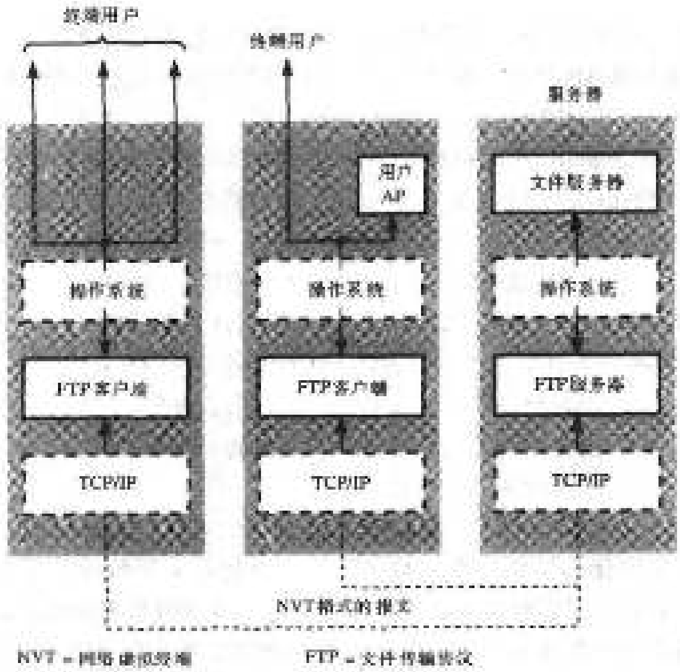


图13-4 FTP客户—服务器交互示意图

随机访问文件是由可变长度的记录组成的。通常，称这种记录为页，称这种文件为页式文件。每个记录或页都有一个头部，包含长度字段，类型字段和提示该页相对于整个文件内容的位置信息。每个页在两个协议实体之间都是以同样的形式传递。

在压缩情形下以块方式传输，两个FTP协议实体要执行检验点检测，从而使大的文件能以一种可控制的方式传递。这与OSI协议栈中会话层提供的同步服务是相似的。

为了同时处理多个请求，每个FTP服务器会对接收到的每个新请求都创建一个新的FTP协议/进程实例，如果FTP客户端支持多个请求的话，也要同样处理。通常，只有一个主进程，拥有知名端口号，发送该端口号给所有的新请求。主进程为这个连接建立新进程，称为控制进程，执行此次会话的各种控制功能，包括登录规程（带有口令）和与传输文件相关的结构以及数据类型定义。还定义是否使用压缩，以及压缩算法的类型。很多实现创建另一个进程 763
来处理的会话相关实际数据传输，在这种情况下，单个FTP会话包含两个传输连接，一个连接用来交换控制信息，而另一个连接用来传递文件内容。当然，这些对于用户和服务器软件都是透明的。

文件内容中的消息格式是由规定的文件结构确定的。在两个FTP控制进程之间传递的消息的格式（即FTP PDU），必须遵循一致的语法形式，以确保消息在两台机器上有相同的含义，并能以相同的方式解释。为了实现这个目标，使用13.1.2节中提到的NVT格式，除非FTP不需要选项协商。

TFTP

FTP是相当复杂的，这是因为它包含各种文件类型的特性，并且，如果需要，它还要包含压缩算法。虽然在涉及互联网的应用中，会需要这些功能，但是在多数的本地应用中，通常

不需要这种级别的功能，例如一个网络文件服务器和本地的无盘工作站群体之间的文件传送协议。

在这种情况下，因为每个客户端发送的文件传送请求都需要经过网络，而服务器和客户端工作站都连接到相同的LAN段上。这种传送不需要FTP中多种功能，因此在TCP/IP协议族中称为**单纯文件传输协议**（TFTP）——一个额外的文件协议，就是为这些文件传送而设计的。

如上所述，TFTP主要用于LAN应用上。回忆第6章中已提到LAN中的位误码率通常是非常小的，所以TFTP使用UDP而不是TCP来传递所有的消息。为了克服可能出现的消息（数据报）损坏，在协议中加入了一个简单的空闲RQ（停止—等待）差错控制规程。关于空闲RQ协议，只有当接收到报文的确认或者出现超时，才会发送一个新的块（报文）。当出现超时，等待确认的报文要重发。对于短传送延迟的LAN来说，这是合适的。

TFTP仅使用4种报文（PDU）类型，也用ASCII编码，它们是：

读请求报文	由客户端发送，发起一个文件的读操作。
写请求报文	由客户端发送，执行一个文件的写操作。
数据块报文	用来传递文件内容。
确认报文	由客户端（读）或服务器（写）发送，确认一个数据块的接收。

为了发起一个文件传递操作，客户端要发送一个包含文件名和文件类型的读或写报文。然后文件请求中的数据以固定长度的数据块传递，每个数据块最多包含512个字节。并且，在
764 每个数据块的头部中有一个序列号，这个序列号会在对应确认中返回。它由接收方（客户端）用于读，由服务器端用于写，以检测确认丢失时的重复出现。接收方收到小于512字节的数据块，就表示文件结束了。

与FTP相同，TFTP服务器可以同时支持多个客户端传送请求。因此，当接收到初始读或写请求报文后，服务器会利用客户端的IP地址和端口号，接着数据块和确认报文传送正确的文件。这些信息是由IP/UDP收到携带有数据块/确认报文的数据报时传递给TFTP的。

13.1.4 SMTP

电子邮件（通常称为e-mail）是一种最流行的计算机网络服务。本地邮件系统在多数大型的交互计算机系统上，已使用多年了。于是，当网络连接起来之后，对邮件服务实现网络扩展成为一个很自然的解决方案。

简单邮件传输协议（SMTP）管理邮件从一台主机的邮件系统到另一台主机的邮件系统的传送，而从本地用户处接收邮件，或者分发邮件到接收者，则不是它的职责，这些功能是由本地邮件系统处理的。SMTP与本地邮件系统的相互关系如图13-5所示。

因为SMTP与本地邮件系统交互，而不是同用户交互，所以当邮件传送限于本地计算机时，它就被屏蔽了。只有当一个邮件要被发送到另一台机器或者要从远程主机接收邮件时，SMTP才会运行。通常在本地邮件系统（也称为**本机邮件系统**）与SMTP客户和服务器之间有一个输入队列和一个输出队列。SMTP客户端负责把邮件发送到其他机器，而SMTP服务器负责接收邮件。虽然，图13-5表示了多用户的情况，但也可以是单个用户（例如个人电脑）。

本地邮件系统为每个用户保留了一个**邮箱**，用户使用它来存取邮件。每个邮箱都有一个惟一的名，它由两部分组成：**本地部分**和**域名部分**。第一部分是用户名，它在本地邮件系统是惟一的，第二部分是主机的标识，它必须在整个互联网是惟一的。通常，它由若干字段组成，它的具体格式随着所属机构类型的不同而变化，例如，教育、政府、军事等等。在第14

查讨论目录服务和名称到地址映射时，会看到一些例子。

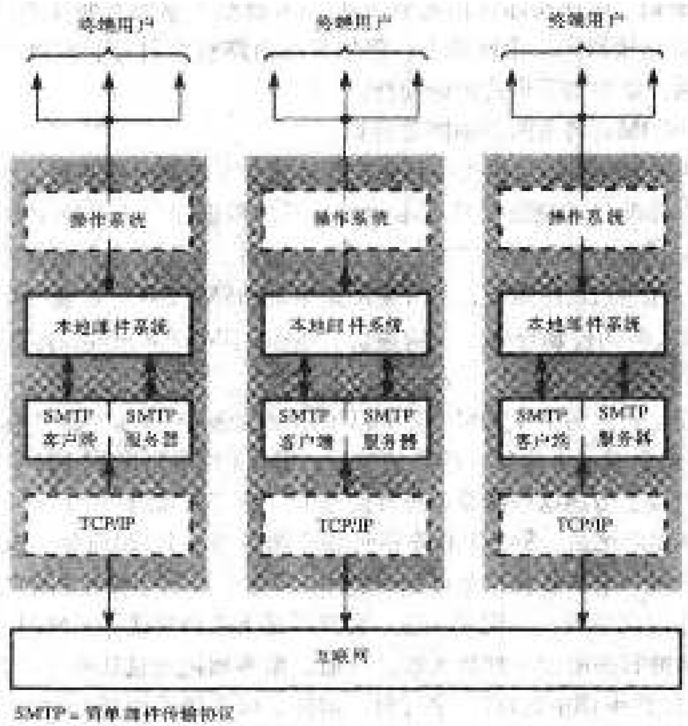


图13-5 SMTP—本地邮件系统示意图

在传送邮件时要考虑两点：一点是邮件的格式，它确保在每个系统上的邮件都以相同的方式进行解释；另一点是用于传送的SMTP，它把邮件从一台机器传送到另一台机器。邮件的格式由头部和体组成，它们由若干行的ASCII文本组成，并具有一个空白行用来分割头部和体。 765
整个邮件只接受ASCII编码。

邮件头部中的每行都由一个关键字和一个后续的字符串组成，两者之间用一个冒号分隔。其中有些关键字是必须有的，而另一些则是可选的。最小的可用邮件头部如下：

TO：收信人名

FROM：发信人名

其他还包括：

TO：收信人名

REPLY TO：回复人名

以及：

TO：收信人名

FROM：发信人名

CC：抄送人

SUBJECT：

DATE：

ENCRYPTED：加密指示

766 ENCRYPTED关键字指明邮件体部分（即邮件内容）使用密钥进行加密，接收方可以从加密指示中得到密钥。邮件头部包括的SUBJECT和收信人名都是明文形式。在有些情况下，虽然两个通信主机连接到同一个网络上，但如果路由路程中包含了互联网网关时，每个网关都会在邮件头部后面添加如下形式的附加行：

RECEIVED FROM：转发邮件的网关标识

它可由每个网关加在头部以提供一种确定邮件通过互联网时所采取的路径的方法。

当一个邮件以标准格式创建之后，本地邮件系统根据收信人名确定是把邮件存入本地邮箱，还是把它添加到输出队列进行转发。为了发送邮件，SMTP客户端首先从目录服务——即**域名系统**中得到目标主机的IP地址，然后使用该地址和SMTP的知名端口号25，发起与目标主机上SMTP服务器之间的传输连接。一旦建立了连接，SMTP客户端发起把等待的邮件传递给对方SMTP服务器的传输。

邮件传递过程包含称为**命令**的SMTP PDU的交换。所有的命令都使用ASCII字符串编码，其中包含3位数字命令或文本命令或两者的结合。SMTP使用TCP SEND/DELIVER用户原语，在已建立的传输连接上传递这些命令。邮件交互命令的序列如图13-6所示。

一旦TCP连接建立之后，SMTP服务器向客户端返回一个220命令，表示已经准备好接收邮件。客户端会返回一个具有客户主机标识的HELO命令。一旦服务器接收到该命令之后，服务器以它的标识作为邮件的一条记录响应。客户端接下来会发送一条MAIL命令，后跟邮件头部中的FROM：字符行开始发送邮件头部。一般，服务器返回确认命令250。客户端继续发送RCPT命令，后跟邮件头部中的TO：字符行。同样，服务器返回250命令；其他头部的行也以同样的方式发送。

客户端发送DATA命令后，就开始发送邮件主体内容。服务器用354命令响应，然后客户端把邮件内容看作字符行序列进行发送，以带有句号的行作为结束。服务器对带有句号的行返回250命令，表示对接收进行确认。客户端发送QUIT命令终止本次邮件传递，服务器端返回221命令，释放传输连接。

SMTP中有一些基本功能，但有些实现中也有一些附加的功能。例如如果收信人在服务器上没有邮箱，SMTP服务器可以返回一个转发地址。而且，当SMTP客户端发送完邮件之后，
767 如果服务器方有待发的邮件，它可以在释放传输连接之前以相反的方向发送邮件。

迄今为止，一直假设所有的收信人的主机都使用SMTP。实际上，在其他网络上使用多种不同的邮件协议。为了使邮件可以传递到其他类型的邮件系统上，必须使用**邮件网关**。一个例子就是TCP/IP-to-OSI网关。如名字所示，它在两个不相似的邮件系统/网络协议之间充当邮件中转站的角色。该网关在一个网络端口使用SMTP接收邮件，而在另一个网络端口使用MOTIS（ISO邮件协议）转发邮件。目前有许多邮件网关能执行这种服务。

13.1.5 SNMP

已经讨论过三个应用协议：TELNET、FTP和SMTP。它们都涉及提供网络范围的用户应用服务。相反地，**简单网络管理协议**（SNMP）不涉及用户服务，而涉及各个主机内的所有通信协议和提供这些服务的各种连网设备的管理，换言之，涉及整个连网环境的管理。

回忆本书的第二部分，一般开放系统连网环境包含各种不同的网络设备。包括用于相似的LAN网段互连的网桥，用于不相似的LAN网络互连的路由器，在分组交换网络中使用的分组交换器，用于网络互连的各种网关，以及连接这些设备的通信链路等等。

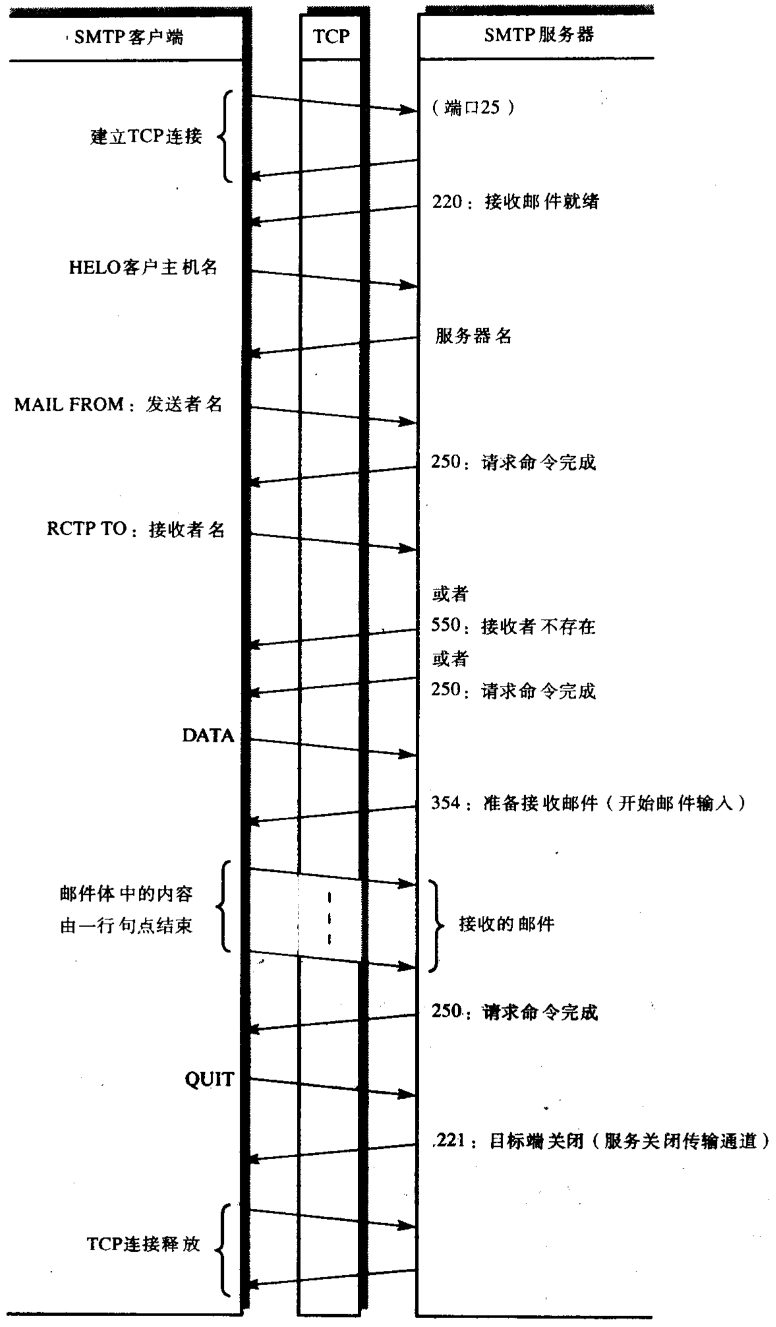


图13-6 SMTP命令交换序列

显然，在任何连网环境中，如果出现了故障，服务被中断，用户会期望在最短的时延内修复故障并恢复服务，这称为**故障管理**。相似的，如果因为某些网段的流量的增加，导致网络性能下降，例如网络响应时间或吞吐量下降，用户会期望这种情形能够反映出来，并且引入额外的设备或传输能力来避免这种问题。这只是**性能管理**的一个例子。另外，多数的TCP/IP协议都有相关的操作参数，例如IP协议中的生存时间参数，TCP协议中的重传定时器

等等。作为网络的扩展，这些参数可能会在网络操作时需要进行修改。这种类型的操作被称为**层管理**。其他的还有**名称管理**、**安全管理**和**计费管理**。SNMP就是用来帮助网络管理员执行故障和性能管理功能的协议。

网络管理的标准方法就是把所有被管理的网络元素（协议、网桥、网关等）看作**管理对象**。
768-769 对于每个管理对象都定义了一组相关管理信息，其中包括网络管理员通过网络可以读写的变量或属性，还包括当故障发生时管理对象要发送的一组**故障报告**。在IP情形下，一个读变量可表示因为生存时间参数到期而被丢弃的IP数据报的数目，一个写变量可表示实际的生存时间超时值。同样，在网关情形下，如果邻网关中止对呼叫的响应，除了对它的路由表进行修改以反映连接的丢失后，网关会创建并通过网络发送一个故障报告，通知管理系统发生了问题。如果管理系统从其他邻近设备收到若干这种报告，可推出是网关发生了故障，而不只是一个通信链路发生故障。

SNMP是一个应用协议，必须使用标准的通信平台，才能使有关的消息（PDU）以及用户服务中的消息能够同时传输。因此，SNMP同前面的三个协议一样也使用了TCP/IP。一般方案如图13-7所示。

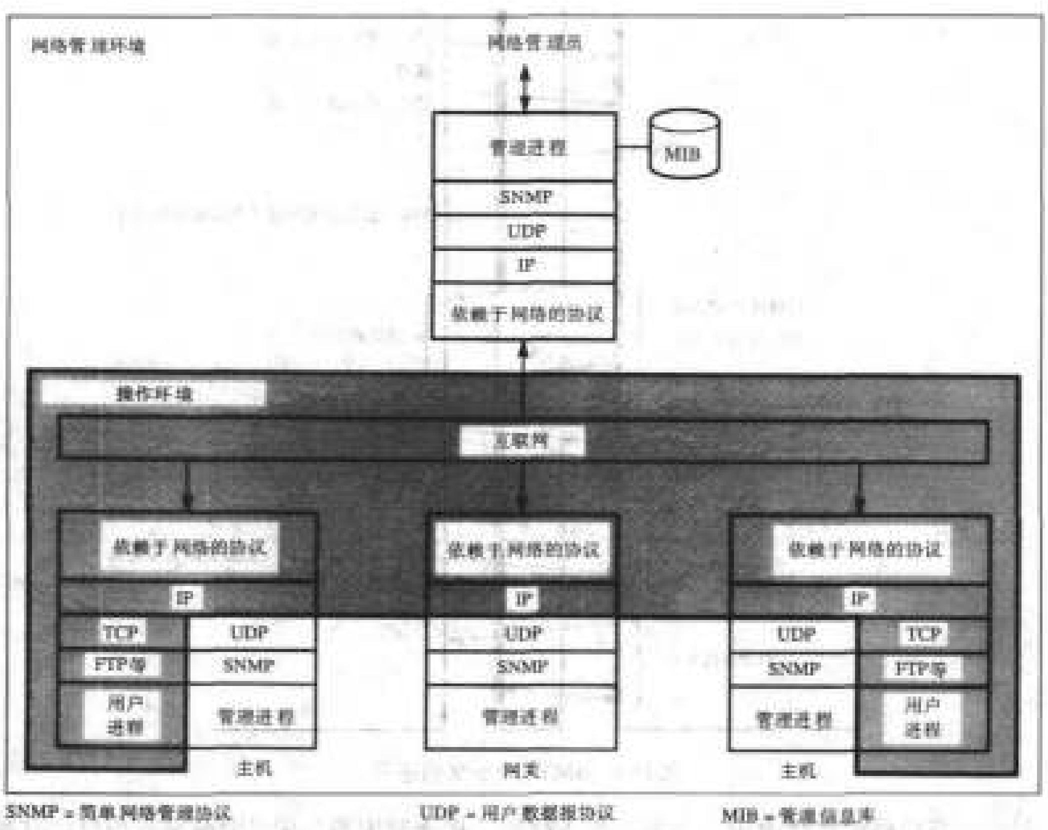


图13-7 SNMP网络管理软件

SNMP的职能就是允许在管理工作站上的**管理进程**可以同各种管理元素（主机、网关等）
770 上的管理进程交换管理信息。这些管理元素上的管理进程定义成用来执行与该元素有关的管理功能。例如对特定变量（计数）请求的响应，接收可操作变量的更新，产生和发送故障报告。

网络/互联网的管理信息保存在网络管理员工作站（主机）的**管理信息库（MIB）**中。网络管理员使用所提供的服务范围进行MIB中信息的查询，初始化条目，收集额外信息，以及修改网络配置。显然，管理员工作站并不是整个网络的中心，所以要采用严格的安全和认证机制。通常有很多层次的授权，这依赖于要执行的操作。在大范围的互联网络中，可以使用多个管理员工作站，每个工作站负责互联网的一个特定部分。

为了反映大范围的被管理对象，管理信息常常被保存在一个**相关数据库**中，这是因为一个被管理对象所拥有的信息常常是被该数据库的几个部分使用。一个简单的层次结构如图13-8所示。

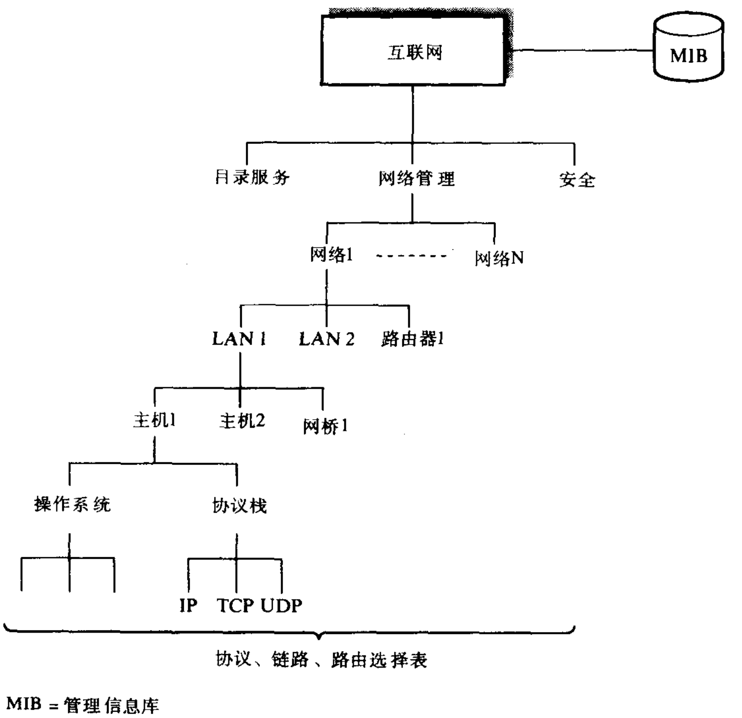


图13-8 管理对象的层次（信息树）

在层次结构的顶端是互联网，互联网是由若干主实体组成，包括目录服务、网络元素和安全服务。网络元素包括网络、内部和外部网关，如果包含子网，还有路由器和网桥。分支的叶是各种管理对象，每个对象都有一个惟一的名称。另外，每个对象都有一组已定义的变量和故障报告。因此，管理并运行互联网的管理机构的主要职责就是定义管理信息树的结构以及MIB的内容。所有设备供应商必须提供管理信息，即有关设备的要求以及如何集成。 771

与互联网相关的实际被管理对象和管理信息在不同开放系统中可能是不同的。因而，SNMP定义在各种连网环境中收集管理信息。但是，各种传送的管理信息的含义对于SNMP则是透明的，它只提供一组服务，每个服务都有相关的参数（按规定的语法定义）。各种管理进程按各自定义的方式解释传递过来的管理信息和命令。在ISO协议的操作模式下的解释方法比已经讨论的三种协议中的方法更多，这反映出SNMP是一个比较新的协议。回忆一下三种协议集成多种应用有关协议到它们的一般操作中。

SNMP用户服务和命令以及匹配的PDU都是远程规程/操作类型，因此比较简单。用来管理进程的三种服务原语是：

- get request（读取请求） 用来请求与被管理对象相关的某个变量（或变量列表）的当前值。
- get-next-request（读取下一个请求） 用来请求某个变量的下一个值，例如由一列元素组成的表。
- set-request（设置请求） 用于传送某一个值把它赋予被管理对象的某个特定变量，比如一个协议层的操作参数和适用于（代理） 管理进程的两个原语。
- get-response（读取响应） 用于返回早先的get request请求的值。
- trap（陷阱） 用来报告一个故障的发生，例如丢失同邻近网络的通信。

由SNMP产生的用来响应这些原语的结果消息（PDU）都使用UDP进行交换，并使用ASN.1定义。图13-9显示了一个简化的定义。

```
SNMP_ PDUs ::=
BEGIN
PDU ::= CHOICE {
                    GETrequest,
                    GET_NEXTrequest,
                    SETrequest,
                    GETresponse,
                    TRAP }
GETrequest ::= SEQUENCE {
                    requestID [0] INTEGER,
                    errorStatus [1] INTEGER,
                    errorIndex [2] INTEGER,
                    varBindList [3] SEQUENCE {
                                      objectID IA5String,
                                      value NULL }
                                   }
GET_NEXTrequest ::=
                    |
                    |
                    |
END
```

图13-9 SNMP PDU的简化ASN.1定义

两个SNMP协议实体并不保留状态信息，这意味着管理进程可能会有多个等待响应的未解决请求。因此，每个GETrequest PDU包含一个请求标识符requestID，它也在后续的响应中出现，网络管理进程利用它把相应的PDU与特定的请求联系起来。varBindList包含许多对象名/值对的序列。在GETrequest PDU中，每个对象名称的值都是NULL，但是在GETresponse PDU中会返回实际的值。因为对象名称是IA5String，所以它的含义对于SNMP来说是透明的。

772 最后，将在13.2节中看到，有一个称为**公共管理信息协议**（CMIP）的ISO管理协议。它相对于SNMP具有更复杂的特性，这使得它更适合大的网络或互联网。因为这一点，对于某些从不使用TCP/IP协议的网络而言，该协议是网络管理协议的首选，而不是等价的ISO协议。当使用在TCP/IP协议上时，CMIP也称为CMOT，即TCP/IP上的CMIP。

13.2 ISO应用协议

除了第12章描述的各种应用支持协议，还定义了一组完整的特定应用协议（应用服务元素，ASE）。其中大多数已经是完全的国际标准（IS），而其他一部分则仍处于草案形式（DIS）。某些协议提供了与13.1节描述的TCP/IP协议中的应用协议相似的服务。现有的ISO应用协议包括：

- 虚拟终端（VT） 这种IS提供了同TELNET协议相似的服务。
- 文件传输访问和管理（FTAM） 这种IS提供了同FTP协议相似的服务。
- 面向消息的文本交换标准（MOTIS） 这种IS提供了同SMTP协议相似的服务。
- 公共管理信息协议（CMIP） 这种IS提供了同SNMP协议相似的服务。
- 作业传输和操作（JTM） 这种IS为用户AP提供了向远程AP提交作业（工作规范）并由它处理的便利。 773
- 生产消息服务（MMS） 这种IS提供了一种标准方法，运行在自动加工车间的管理计算机上的AP能够向其他分布式的控制数字机工具（例如，可编程的控制器，机器人等等）的AP发送相关的加工报文。
- 远程数据库访问（RDA） 这种DIS为用户AP提供了访问远程数据库管理系统的方法。
- 分布式事务处理（DTP） 这种DIS提供了必要的支持服务，它使两个AP根据第12章为CCR协议定义的规则进行通信。

如同TCP/IP应用协议，各种ISO协议的目标就是使两个运行在不同计算机系统上的AP能够相互通信，执行特定的分布式应用功能。例如，FTAM的目标就是使运行在一台机器上的客户端进程可以同运行在远程计算机（可能是不同的）上的文件服务器进行交互，好像客户端进程与服务器运行在同一台计算机上。

为了实现这个目标，ISO为每种分布式应用功能定义一个虚设备模型——虚终端、虚文件库、虚制造设备等等。因此，实现两个通信的ASE，好像两个通信AP都是按这种模式操作的。利用这种方法，每个ASE的所有（用户）服务都具有标准的形式，与两个通信进程之间可能存在的差别无关，ASE协议能被独立地实现。任何存在的差别都被连接到该AP的用户元素在开放系统互连环境的外部解决。表示层提供的服务则确保在两个通信进程间交换的信息在两方都具有相同的含义。现在，看一下主要的ISO协议。

13.2.1 VT

虚终端（VT）ASE使一个终端上的用户可以同一个远程计算机上运行的AP进行交互，就好像终端是直接连接到那台计算机上的。有很多种不同的终端类型，每种都有不同的操作特征。例如，**滚动模式终端**对单字符操作，**屏幕**或**页模式终端**对全屏幕字符操作，**表格模式终端**通常被以固定模板（表格）输入字符或词的应用程序使用。每一个类别都有很多变型，而对于这些终端已经开发了大量的应用软件。VT ASE的目标就是使应用程序能够被各种类型的终端以开放的方式访问。 774

不同于TELNET协议，VT ASE（协议）客户端不提供直接的终端接口。而一个本地AP（用户元素）必须用于管理与终端的交互。用户元素根据已定义的VT特征同VT ASE客户端进行交互。如果有必要，它还要在两台终端之间执行特征映射操作。一般方案如图13-10所示。

由于大量不同种类的终端的存在，VT不只有一种。VT ASE允许两个用户协商应用所需要的特定VT的特性。在交互之前，双方用户必须一致同意**虚终端环境**（VTE），将为应用使用。在某些情况下，只需要协商少数的几个操作参数，而在另一些情况下，需要协商的参数则可能很多。

为了帮助两个用户协商操作参数（特性），已定义若干个**标准配置文件**。两个用户可以采用特定配置文件中的参数，或是协商所选择的参数的变化。

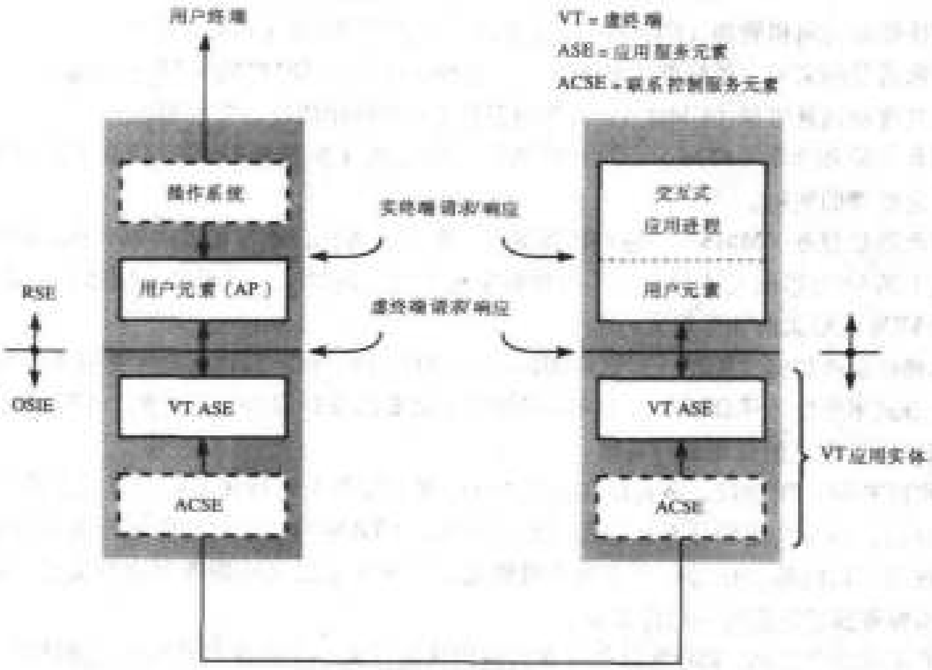


图13-10 VT交互示意图

两个VT用户使用称为**概念通信区域**（CCA）的共享数据结构进行通信。每个VT协议实体
775 都会保存一份独立的CCA拷贝。当本地用户元素/进程在接口发送一个已定义的VT请求服务原语时，每个实体所持有的CCA的内容就开始变化了。接下来的变化会在本地CCA上实施，并利用VT协议和相关的PDU转发到远程系统上的对等VT实体。然后，所传送的变化会执行并通过匹配的指示原语转发到对等的用户进程。一般方案如图13-11(a)所示。

每个CCA包含若干个数据结构，它们共同描述该VT的特性，如图13-11(b)所示。

对于每个终端输入和输出设备（键盘、鼠标、显示器、打印机等等）都有三个对象，每
776 个对象都由多个参数组成。三个对象如下：

- **显示对象** 它允许通过虚设备的对应事件表示实际设备上的相关事件。一个显示对象由元素数组组成，每个元素都包含属于该VT字符集的一个字符。
- **控制对象** 它用于不是由用户触发的模拟终端特性，例如，响铃。
- **设备对象** 模拟实设备的特性。每个设备对象中的一组布尔变量指明开关的状态。

VT支持同步和不同步的操作模式。在同步模式中，连接/联系每一方的输入和输出功能都被组合起来。使用了大量的令牌控制用户与远程进程之间的交互顺序。在非同步模式中，输入和输出功能是独立的，它使每一方可以同时启动一个事件。当使用同步模式时，**访问控制存储**（ACS）持有所使用的令牌的当前状态。实际上，令牌是通过会话层（通过表示层）提供的令牌控制服务传递的。**数据结构定义**（DSD）包含在该联系中使用的显示、控制和设备对象的相关参数的类型定义。

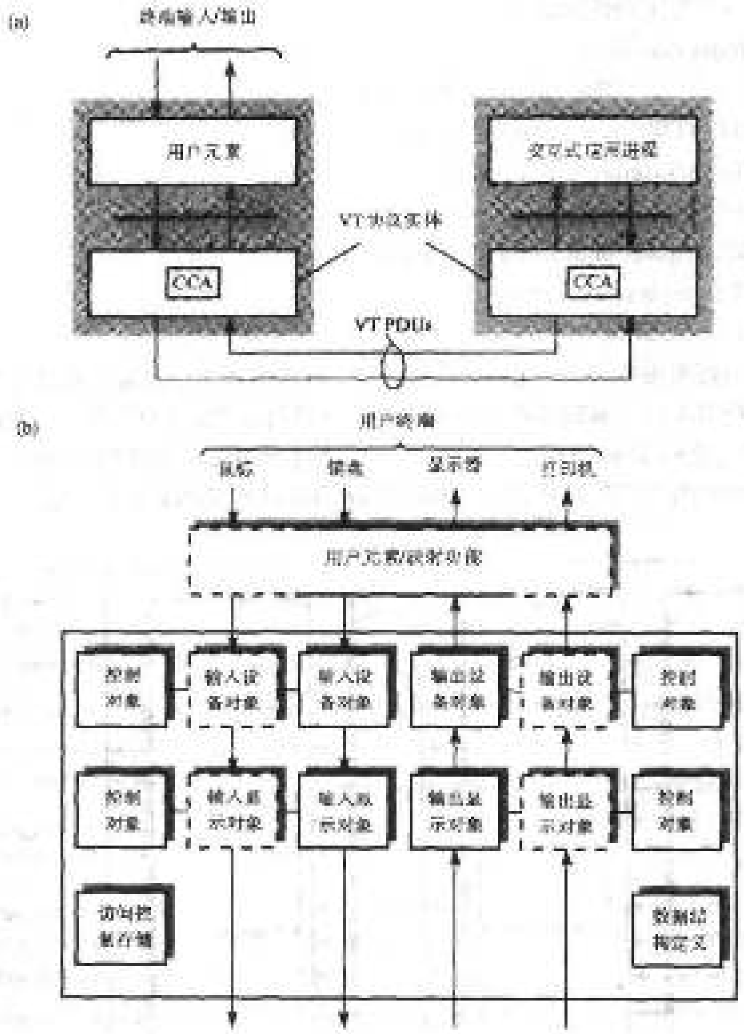


图13-11 VT协议数据结构

(a) 概念通信区域 (b) 对象定义

用户服务

VT ASE用于连接ACSE支持协议，因此，建立联系的原语包括：

VT_ASSOCIATE.request/indication/response/confirm

这个联系用于协商使用的终端特性的原语包括：

VT_START_NEG.request/indication/response/confirm

VT_NEG_INVITE.request/indication

VT_NEG_OFFER. request/indication

VT_NEG_ACCEPT. request/indication

VT_NEG_REJECT. request/indication

VT_END_NEG. request/indication/response/confirm

用于发起CCA的变化的传输的原语包括：

VT_DATA. request/indication

用于控制使用的令牌（同步模式）的原语包括：

VT_REQUEST_TOKEN. request/indication

777 VT_GIVE_TOKEN. request/indication

以及用于终止联系的原语包括：

VT_RELEASE. request/indication/response/confirm

VT_U_ABORT. request/indication

VT_P_ABORT.indication

VT协议实体对每个服务原语产生一个PDU响应。该PDU传递给对等的协议实体，如图13-12所示。VT_ASSOCIATE、RELEASE和ABORT原语所创建的PDU作为等价的ACSE中的A_ASSOCIATE原语的用户数据传递。除了两个令牌控制PDU，其余的PDU利用P_DATA传递，

778 而两个令牌控制PDU则利用P_TOKEN_PLEASE和P_TOKEN_GIVE服务传递。

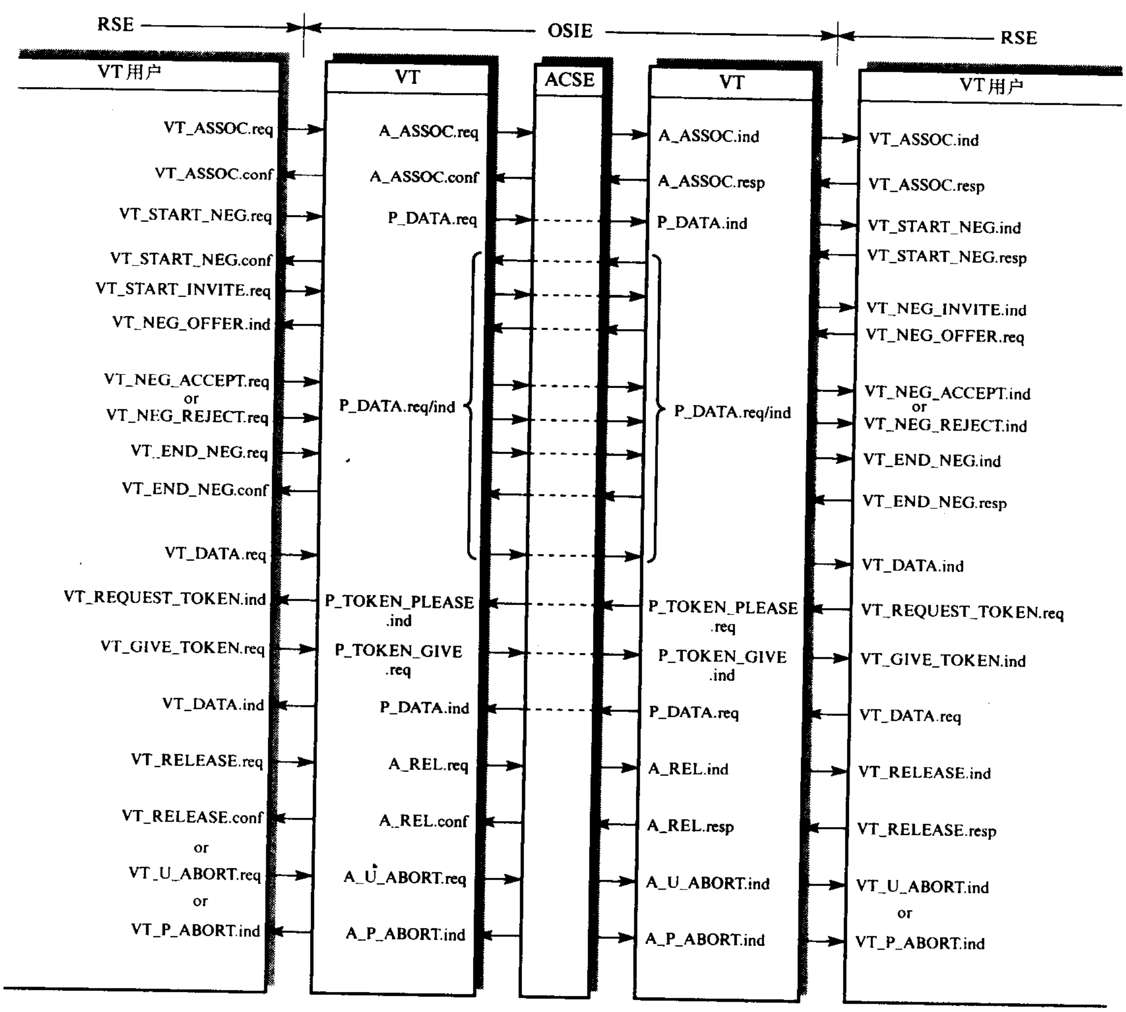


图13-12 VT用户服务和ACSE交互操作

13.2.2 FTAM

要注意的是，各种ASE协议元素并不涉及提供具体的应用服务，而是允许实系统环境（RSE）中的用户AP提供的服务以开放的方式访问和使用。例如，文件传输访问和管理（FTAM）ASE允许客户进程的分布式群体访问并管理一个远程文件服务器，它可能作为来自不同厂家或不同客户端系统的机器上用户AP实现。每个客户进程（及其相关的应用实体）称为**发起者AP**，而服务器进程则称为**响应者AP**，如图13-13所示。

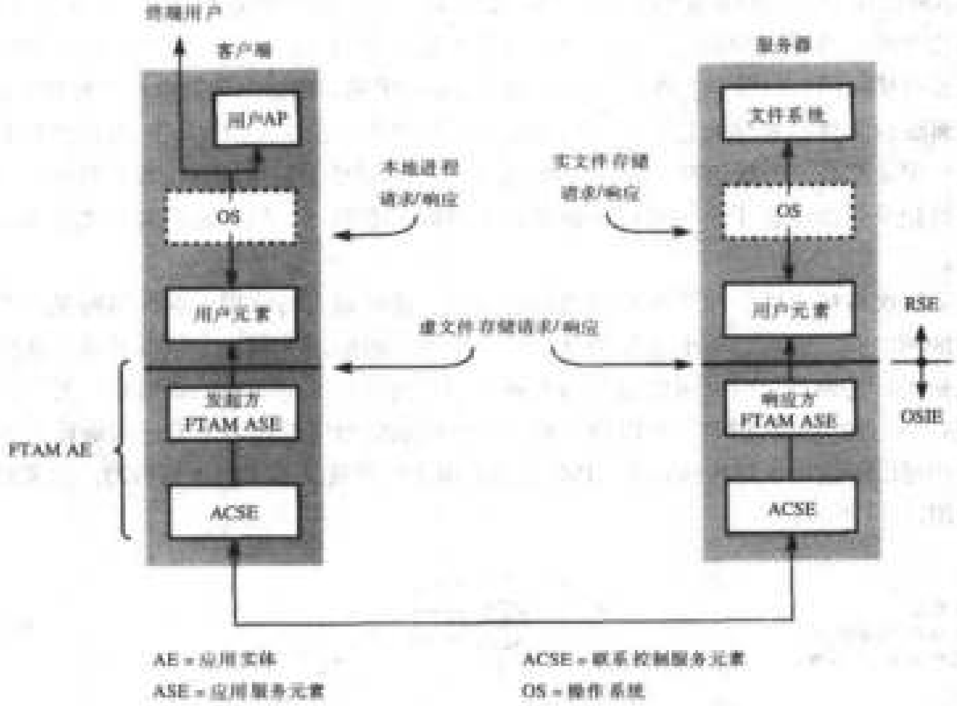


图13-13 FTAM虚设备接口

客户端AP访问和管理远程文件库所用的原语，以及访问和管理实文件库所用的原语，都是本地（就是说，只取决于机器）事件。以这种方式，就可使用现有文件系统和相关的访问软件。用户只需要提供（假设使用的是OSI软件）相关的用户元素，以实现必要的映射功能。

1. *虚文件库模型*

在描述FTAM的用户服务原语之前，必须考虑原语的（虚）文件库模型。显然，目标是采用一种足够灵活的模型，以最小的映射功能集访问和管理任何实文件系统。 779

虚文件库设计为可寻址的实体，通过它可以与远程用户（发起者）通信。在任何时刻，可以有任意数目的发起者同文件库（响应者）建立联系。文件库可以包含任意数目的文件，每个文件都有一些属性，这些属性包括：

- 文件名：能惟一地确定文件。
- 允许的操作：表示文件可执行的操作的范围（读取、插入、置换等等）。
- 访问控制：只读，读写等等。
- 文件规模。
- 文件内容的有关表示。

- 创建者标识。
- 文件建立的日期和时间。
- 最后的修改者/读者的标识。
- 最后一次访问的日期和时间。
- 内容类型。
- 密钥。

FTAM协议中有三种常规的文件结构：无结构，平面结构和层次结构。如它的名字所示，**无结构文件**内没有明显的数据结构。文本文件就是一个很好的例子。虽然，文本文件对于用户来说是有结构的，但对于文件系统而言该结构是透明的，因此只能对整个文件进行读或写。

平面结构文件由有序的记录（数据单元）序列组成，记录可以是不同的长度和类型。通常，每个记录都有一个标识符（标记），它使用户能够对单个记录进行读写。然而，要注意的是，虽然记录可以由若干个（用户可识别的）数据元素组成，但是它们对于文件系统是不可识别的。

对于**层次结构文件**，除了有可识别的记录外，还有相关的结构。这种结构是具有多个分支结点的树结构。通常，每个分支结点都有一个相关的标识符和一个数据记录。这些结点可以用已知的顺序确认，所以可以通过结点相对于其他结点的位置确认该结点。另外，如果某个结点有一个相关的标识符，可以使用相对于根结点的给定路径名对它进行确认。其他两种结构类型可以看成层次结构的特例。ISO采用的虚文件库就是基于层次结构的。有关的结构和术语如图13-14所示。

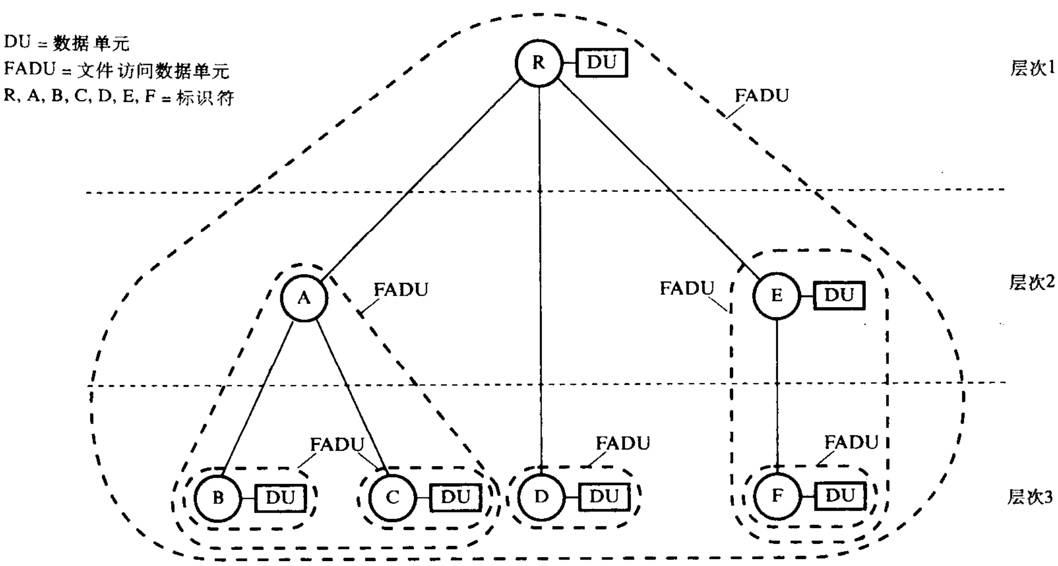


图13-14　虚文件库访问结构

树结构由**一个根结点**以及用**弧线**连接的**内部结点**和**叶子**组成。一个结点只属于一个惟一的层。每个结点对它的**子树**的访问，称为**文件访问数据单元**（FADU）。文件库中的文件内容可以由一个或多个**数据单元**（DU）组成。在大多数情况下，一个结点只能分配一个DU，意

780

味着可以通过识别FADU结点的标识访问DU。对FADU中每个结点的访问是按下列顺序进行的：R，A，B，C，D，E，F。

DU是一个典型的数据对象（标量、向量、集合），并包含不可再分的原子数据元素，称为**数据元素**，每种数据元素用抽象语法（字符、字节、整数、等等）表示；一个DU中的所有数据元素都相互有关。通常用树型结构表示元素间相互关系，即使是单数据单元或一个向量的相互关系也可用树表示。树是按刚才描述的顺序移动的，将访问的DU以这个顺序传递给表示实体。表示实体独立处理每个元素，并使用相应（协商）的传送语法按相应顺序传输数据元素，并维持元素的相关顺序。

用相应服务原语调用文件库的有关操作。文件库的操作包括文件建立、打开、关闭和删除，以及在文件中定位、读取、插入、置换、扩展和删除某个DU。

2. 服务原语

给出虚文件库模型的概括说明，现在可以定义FTAM中的相关用户服务原语。这些原语可以被组合进一个嵌套的**区间**集中，如图13-15(a)所示，每个嵌套区间有一系列允许执行的操作。

每个区间定义一组相应的服务原语，简化的状态变迁图如图13-15(b)所示。与每一区间对应的服务原语概括如下： 781

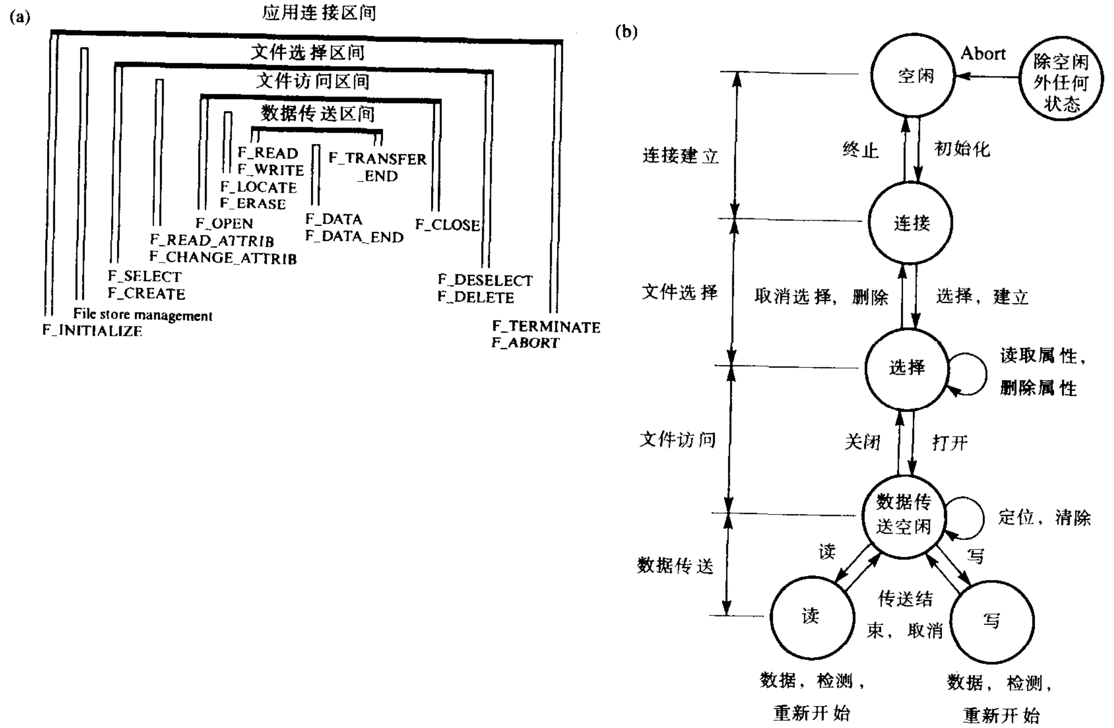


图13-15　FTAM服务原语

(a) 服务区间　(b) 简化状态变迁图

• **应用连接**（联系）　并不专用于FTAM，也涉及ACSE。对于FTAM，该原语建立授权和计费信息，这对确保文件库的操作是必要的。

- **文件选择** 识别（或建立）惟一文件名（FADU），它与接下来阶段的操作有关。根据文件名选择（识别）进程，在接下来的阶段对这个文件执行操作。
- **文件访问** 建立一个区间，在此区间发生文件数据传送。它包括传输请求能力和适当访问环境的建立。
- **数据传送** 可对访问（识别）的FADU有关DU执行命令操作（读，写等）。为了表示，可以识别更小的数据元素，但这些元素不能被独立地访问。

与其他协议层一样，FTAM的服务原语可分成若干个功能单元（FU）。在联系建立阶段，特定的（对话）会话期间请求的FU，可包括如下：

782 - **核心FU** 提供应用关联和终止，选择文件（取消选择）和打开文件（关闭）等功能。
- **读FU** 提供读取文件（大块数据）和单个DU的功能。
- **写FU** 提供写文件（大块数据）和单个DU的功能。
- **文件访问FU** 提供定位和删除FADU的功能。
- **有限文件管理FU** 提供建立文件、删除文件和读取文件属性的功能。

图13-16中的时序图表示了每个FU的相关原语。注意，为了使图更简洁，图13-16(c)只显
783 示了请求原语。实际上，还应包含相应的证实服务。

图13-16中的所有服务都是关于**常规传输模式**的，它们主要用于服务器为每个客户端保持一组独立的文件。另外还有一种模式称为**可靠的传输模式**，它用于单一文件可以同时被多个客户端访问和更新的情况，例如事务处理应用。文件内容的所有改变都以一种可以控制的方式执行。为此定义了CCR ASE；在可靠的模式中使用的附加FTAM原语用于直接映射等价的CCR原语。表13-3显示了这种映射关系。12.8节对此进行了详细的描述。

表13-3 关于可靠文件服务FTAM到CCR的映射

FTAM	CCR	用 法
F_BEGIN_GROUP	C_BEGIN	通知一个原子事务的开始
F_END_GROUP	C_COMMIT	通知一个原子事务的结束
F_RECOVER	C_ROLLBACK	返回到原子事务的开始
F_RESTART	C_RESTART	表明发送方的当前状态

为了实现文件服务的可靠性，FTAM应用实体包含了FTAM ASE和一个关于ACSE以及CCR应用支持ASE的实例。

3. 协议

如图13-16所示，FTAM协议接收到一个服务原语，根据该原语的类型与参数建立一个相应的PDU。然后，该PDU在相应的ACSE/表示原语的用户数据字段中进行传递。在所示的序列中，假设使用了数据令牌，因此，在数据传输阶段之前，要使用P_TOKEN_GIVE原语把令牌从发起FTAM实体传递给响应FTAM实体。注意，在数据阶段，没有对应于F_DATA服务的PDU。这是因为使用P_DATA服务把文件的数据当作标记的数据元素串（用它们抽象语法）直接传递给表示实体。

最后，为了说明如何确定ISO应用协议以及把图13-16所示序列形式化，在图13-17和图13-18中给出FTAM协议机的形式化规范部分说明。这一部分规范说明给出了联系建立和文件选择区间。图13-17给出了缩写名和入事件、状态、输出事件和谓词的含义，图13-18给出了有

关的事件—状态表。图13-18中的两个表是关于发起协议实体（连接到客户端进程）和响应协议实体（连接到服务器进程）事件—状态表。所有的ISO应用协议都使用这种规范说明模式。 784

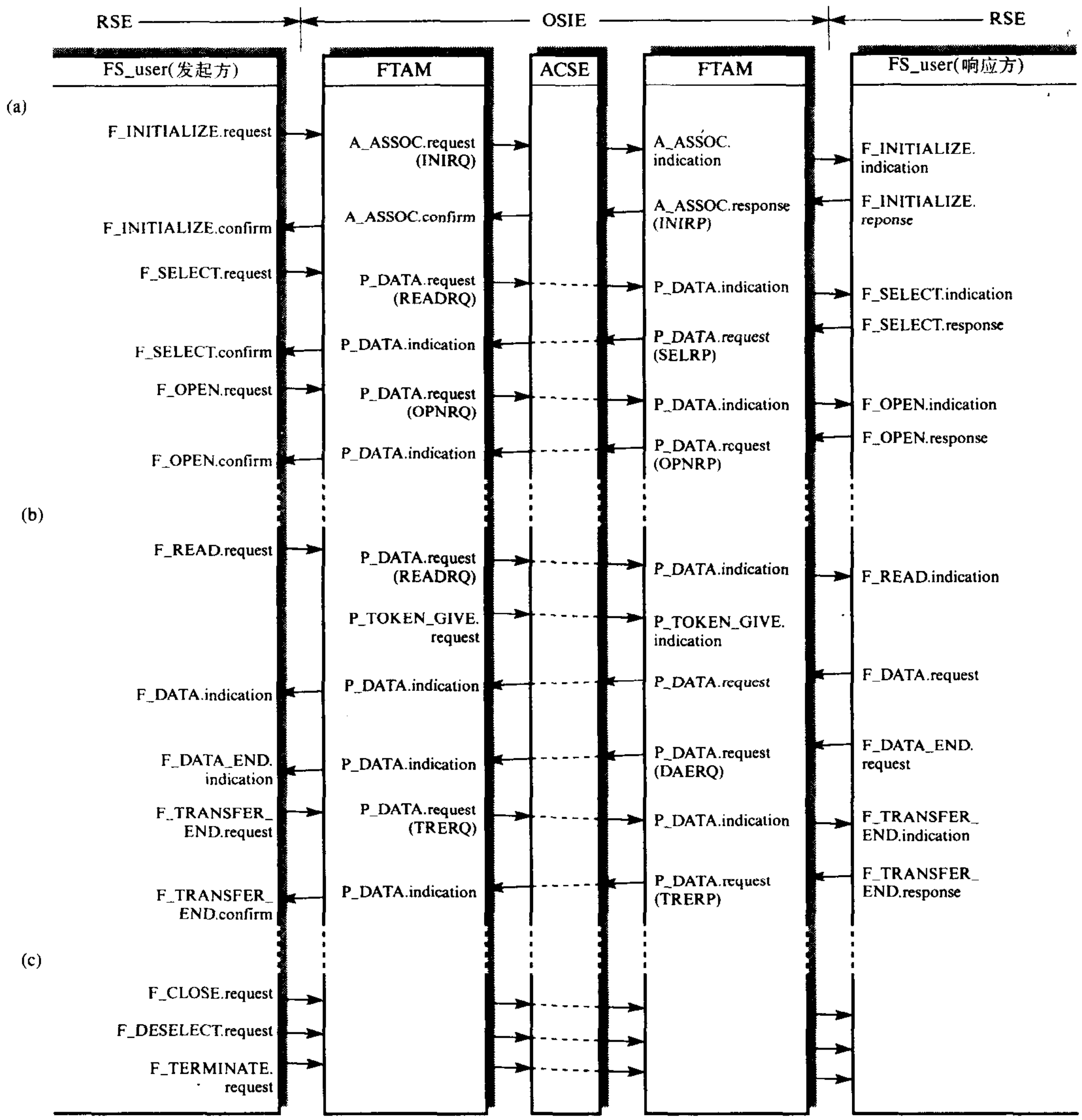


图13-16　FTAM服务原语

(a) (c) 核心FU　(b) 读FU

13.2.3 MOTIS

MOTIS是ISO的电子邮件系统，它与TCP/IP协议族中的SMTP协议发挥相同的作用。实际上，MOTIS不只是一个简单的协议，而是一个完整的电文（邮件）传输系统。它还称为ISO**电文处理系统**（MHS），基于由ITU-T定义的公共X.400电子消息处理服务。

ITU-T的建议X.400提供一种国际电子消息服务，它同当前的（手工传递）邮件系统是相似的。X.400包含一系列协议，每个协议执行整个MHS的一个特定功能。图13-19显示了组成X.400建议集的各种实体（和相关协议）。

(a)

缩写名	接 口	含 义
F_INIRQ	FS_user	接收到F_INITIALIZE.request
F_INIRP	FS_user	接收到F_INITIALIZE.response
F_SELRQ	FS_user	接收到F_SELECT.request
F_SELRP	FS_user	接收到F_SELECT.response
INIRQ	CS_provider	接收到初始化请求PDU
INIRP	CS_provider	接收到初始化响应PDU
SELRQ	PS_provider	接收到选择请求PDU
SELRP	PS_provider	接收到选择响应PDU

(b)

缩写名	含 义
STA 0	应用连接关闭
STA 1	联系挂起
STA 2	应用连接开放
STA 3	选择挂起
STA 4	选择的

(c)

缩写名	接 口	含 义
F_INIIN	FS_user	发出F_INITIALIZE.indication
F_INICF	FS_user	发出F_INITIALIZE.confirm
F_SELIN	FS_user	发出F_SELECT.indication
F_SELCF	FS_user	发出F_SELECT.confirm
INIRQ	CS_provider	发出初始化请求PDU
INIRP	CS_provider	接收到初始化响应PDU
SELRQ	PS_provider	发出选择请求PDU
SELRP	PS_provider	发出选择响应PDU
F_ABTIN	FS_suer	发出F_ABORT.indication
ABTRQ	CS_provider	发出异常中止请求PDU

(d)

缩写名	含 义
P1	可接受的F_INITIALIZE.request
P2	可接受的INIRP PDU
P3	可接受的INIRQ PDU
P4	可接受的SELRP PDU
P5	可接受的F_SELECT.response

(e)

缩写名	含 义
[1]	初始化状态变量
[2]	设置相应的拒绝参数

图13-17 FTAM协议机中的缩写名称

(a) 人事件 (b) 自动机状态 (c) 出事件 (d) 谓词 (e) 特定动作

(a)

状态 / 事件	STA 0	STA 1	STA 2	STA 3	···
F_INIRQ	1	0	0	0	
INIRP	0	2	0	0	
F_SELRQ	0	0	3	0	
SELRP	0	0	0	4	
⋮					

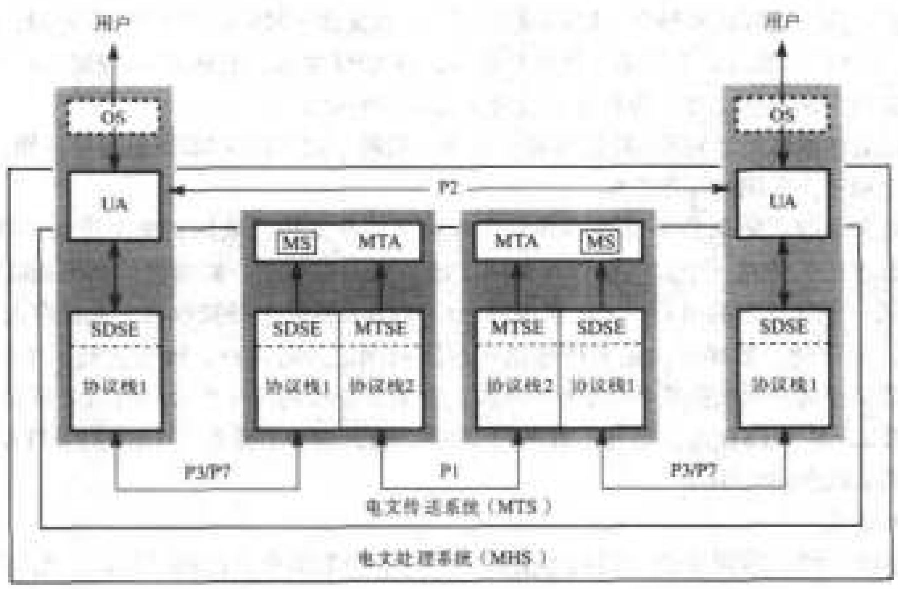
0 = F_ABTIN, ABTRQ, STA 0
1 = P1: INIRQ, [1], STA 1;
　NOT P1: F_INICF, [2], STA 0
2 = P2: F_INICF, STA 2;
　NOT P2: F_INICF, [2], STA 0
3 = SELRQ, STA 3
4 = P4: F_SELCF, STA 4;
　NOT P4: F_SELCF, [2], STA 2

(b)

状态 / 事件	STA 0	STA 1	STA 2	STA 3	···
INIRQ	1	0	0	0	
F_INIRP	0	2	0	0	
SELRQ	0	0	3	0	
F_SELRP	0	0	0	4	
⋮					

0 = F_ABTIN, ABTRQ, STA 0
1 = P3: F_INIIN, [1], STA 1;
　NOT P3: INIRP, [2], STA 0
2 = INIRP, STA 2;
3 = F_SELIN, STA 3
4 = P5: SELRP, STA 4;
　NOT P5: SELRP, [2], STA 2

图13-18　FTAM协议机的事件—状态表
(a) 发起实体　(b) 响应方实体



OS = 操作系统　　SDSE = 提交/投递服务元素
UA = 用户代理　　MTSE = 电文传送服务元素
MTA = 电文传送代理　　MS = 电文库

图13-19　X.400功能模型

假定用户到系统的接口是一个终端（例如，个人计算机），它有足够的内存和处理能力，使用户可以交互地产生一个电文（邮件）并对接收到的电文进行读和浏览。这是**用户代理**
785~786 （UA）的功能之一。并且，因为MHS使用各种类型的电文，UA必须能够同远程终端上的对等UA进行通信，因而所传递的电文在两个终端上有相同的含义。这是**P2协议**的职责。显然，电文的结构和含义对于每个应用都可能是不同的，所以要定义这样一组协议，每个协议用于一个特定的应用领域。例如，简单的个人到个人的电文标准称为**个人间邮件（IPM）协议**。

一旦准备好一个电文后，UA加入自己的协议信息，传递给**提交/传递服务元素**（SDSE）。SDSE的职责就是控制消息提交和接收到等价的本地邮政系统，本地邮政系统被称为**电文传送代理**（MTA）。在SDSE和本地MTA之间定义了控制电文传递的协议，称为**P3协议**，它包括了提交和传递规程以及相应的功能，例如收费功能。

UA使用全球惟一的名称彼此通信。然而，MTS使用一个完全限定的地址，即PSAP地址。当本地MTA接收到电文时，它首先必须执行一个名称到地址的翻译。名称和地址的结构在ITU-T建议X.500中定义。有一个与MTA相关的目录服务代理，它执行上面的翻译功能（参见第14章）。本质上，如果两个用户终端都是连接到国际X.25网络上，那么这个地址就是X.121的终端地址，具有第8章描述的结构。如果使用的是互联网，那么这个地址就是第9章中描述
787 的类型。

一旦MTA获得了地址，它会把发送方和接收方的地址以及其他指定给MTS的信息添加到从UA接收到的电文头部，从而产生一个电子信封。然后MTA会在第三个协议称为**P1协议**的控制下使用适当的协议栈发送该电文。

接收到发给某个UA（终端）的电文后，MTA会试图使用P3协议把该电文传递给终端上的SDSE。然而，因为用户终端通常是位于MTA之外的，所以这时MTA可能已经关闭或者并没有提供服务。为了适应这种情况，MTA提供了一个**电文库**（MS）。如果UA不能用，则该电文存入MS，等待以后发送。于是引入第四个协议，称为**P7协议**，它通过UA控制用户和本地MS之间的交互行为，这些交互行为是为了检索正在等待的电文。

虽然以上的描述是针对X.400公共电文服务，但基于此的ISO MHS也同样适用。图13-20显示了一个用于大范围的专用系统。

区域电文系统必须既能够支持本地电文传输又能够支持区域间的电文传输。在例子中，每个用户终端（工作站、个人电脑等）都同区域范围的电文服务器通信。如果接收到的电文是传递给同一个区域内的用户，那么服务器上的MTA直接转发给接收者，或者存入本地电文库中等待以后传递。如果电文是要传递给该区域范围之外的用户，因为这是一个专用系统，所以必须把消息提交给公共X.400系统。通常，电文服务器到同一个公共网络之间有一个直接连接，它被X.400 MTS使用。因此，在图13-20中，假设该服务器有一个相连的MTA。如果不是这样，那么就使用SDSE。

1. 电文格式

正如前面提到，有很多种不同的电文类型，每个类型都有自己的P2协议。然而不考虑电文类型，所有的电文都是由一个MTS头部（信封）和电文内容组成。MTS使用地址和头部中的其他字段把电文传递给目标。这是所有电文的标准形式。如同常规的手工传递邮件（一封信、发票等）的信封内容有接收者使用的附加头部一样，电文内容中也有一个UA使用的附加头部。这个头部随电文传输应用的不同而变化，例如个人（信件、备忘录等）和商业（发票、

订单等）等。IPM协议的电文的结构和内容，如图13-21所示。

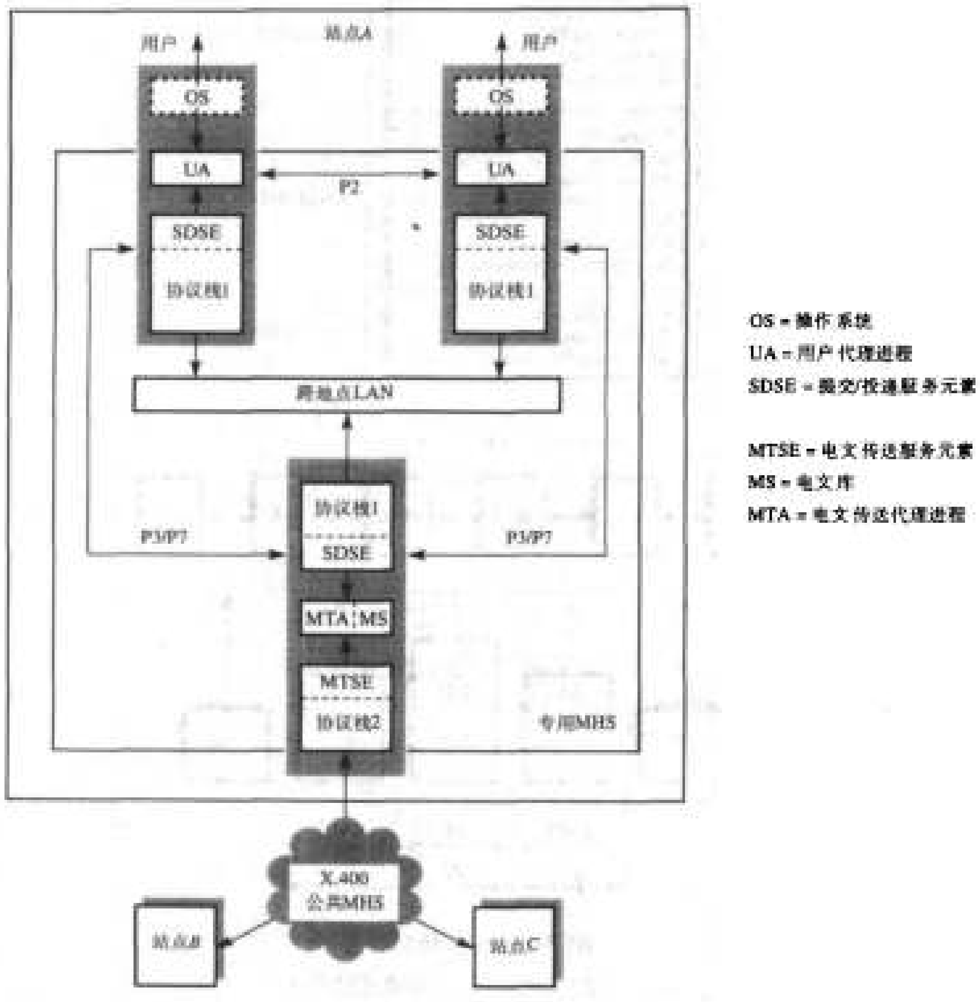


图13-20 MOTIS功能模型

实际上，MTS的头部和内容标题比起图中所示具有更多的字段。并且MHS允许各种类型的用户终端。因此，如果接收方终端不同于发送方终端时，MTA要尽量执行必要的转换。如果MTA不能执行这种转换，该电文就丢弃，并且向发送方MTA返回一个适当原因代码，再传递给UA。一个转换的例子就是X.400终端（和字符集）到图文终端或用户电报终端的转换。 788

当电文通过MTS传输时，MTS头部中的加密标识符字段同其他字段一样，它们共同用来使发送方MTA可以指定被发送的电文内容的安全级别。如果电文的内容被加密，加密标识符字段使目标MTA能够选取合适的密钥进行解密。其他字段还包括加密内容的校验和或发送方同接收方数字签名的校验和。12.4节对此进行了更详细的描述。另外要注意，不同于SMTP，完整的电文的内容（包含头部）作为一个实体进行加密。 789

2. MHS协议

如图13-19和图13-20所示，ASE和相关的协议栈与每个协议不同。实际上，SDSE是由三个独立的服务元素组成的，见图13-22。

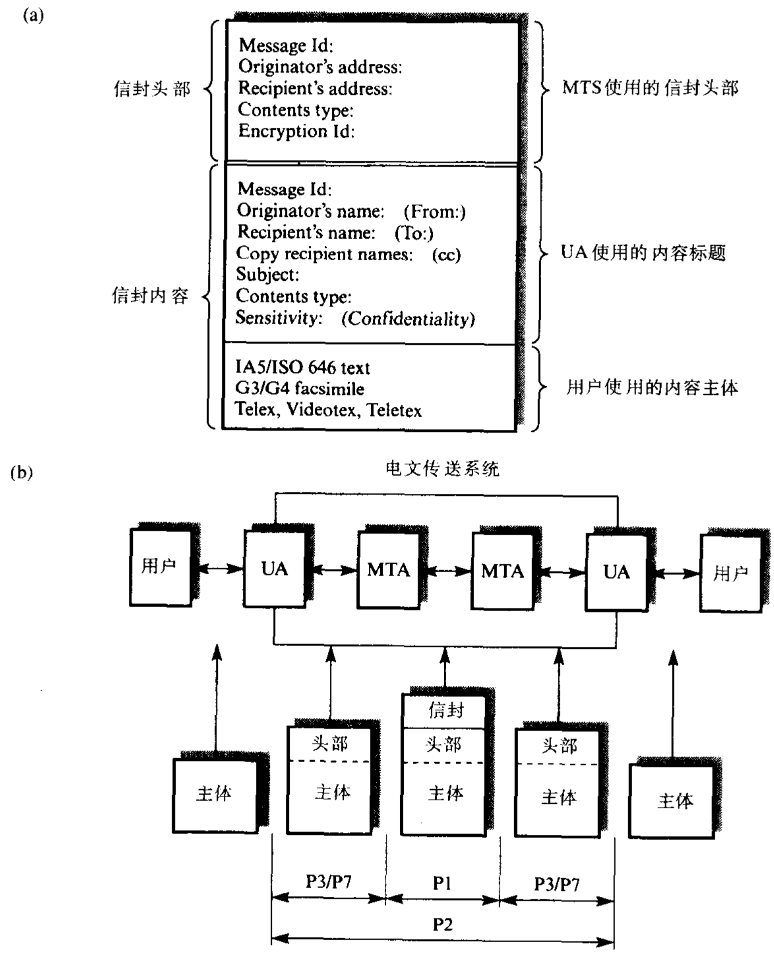


图13-21　电文格式

(a) 个人的电文格式　(b) 电文结构

与P3协议相关的两个ASE分别是**电文提交服务元素**（MSSE）和**电文传递服务元素**（MDSE），而同P7协议相关的ASE是**电文检索服务元素**（MRSE）。MSSE控制UA向它的本地
790
MTA提交电文。同样，MDSE控制接收到的电文从MTA到接收方UA的传递。MSRE被UA用来从MS中检索存储的电文。

这些协议使用ACSE和ROSE应用支持协议。回忆一下，ACSE允许在两个电文处理ASE之间建立联系，而ROSE处理常规的远程调用请求/回复消息传输服务。在应用中，消息在请求（MSSE）中或者回复（MDSE/MRSE）中携带。

与P1协议相关的**电文传输服务元素**（MTSE）与RTSE和ACSE服务元素一起使用。回忆一下，RTSE是一个使用ACSE和会话服务的早期协议。它用一个简单的停止—等待协议使电文（或一系列电文）可靠地在两个（MTSE）ASE之间传递。协议栈中其余用于每个应用实体的协议依赖于所使用的网络/链路类型。在P3和P7协议情形下，可能是一个LAN或拨号线路。同样，对于P1协议，可能会是一个基于X.25的PSPDN，一个帧中继ISDN，或一个专用互联网。

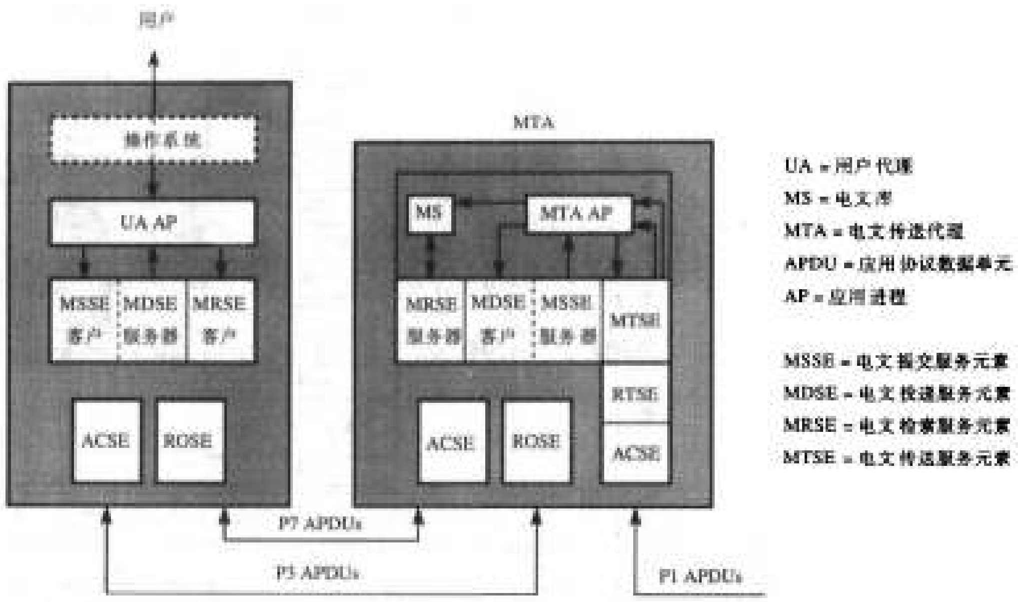


图13-22 MHS协议

13.2.4 SAME

ISO网络管理协议等价于TCP/IP协议族中的SNMP，称为**公共管理信息服务元素**（CMISE）。 791
它只是完整的OSI**系统管理应用实体**（SMAE）的一个成分。SMAE的一个功能就是允许网络管理员能够远程地管理开放系统互连环境相连的各种（被管理）对象，例如，协议、网桥、路由器和分组交换机等。它还提供了一种管理任何连网系统的工具，例如组成完整电信网的各个系统和子系统。而CMISE则只提供了一个用于发送和接收管理信息的基本服务。额外的功能是由另一个ASE提供的，即**系统管理应用服务元素**（SMASE）。完整的SAME的结构见图13-23。 792

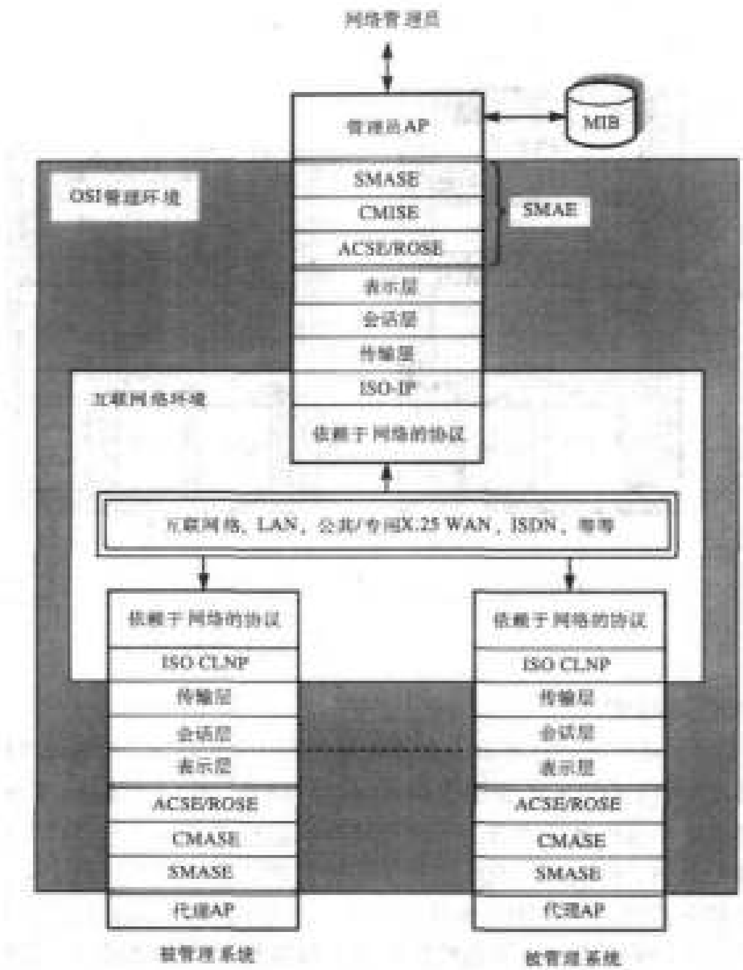
包含一个或多个被管理对象（即网络管理员管理的网络元素）的所有系统，必须包含完整的SAME。虽然图12-23中没有显示，某些情形下管理进程可能会使用FTAM/ACSE应用实体的文件传输服务，例如，对象被初始化后下载对象代码。管理员通过管理AP以及相关的MIB同特定系统的代理AP进行交互，该代理AP管理连接到该系统上的各种（被管理）对象。如同TCP/IP协议栈一样，管理工作站上的MIB包含有关整个网络的配置、性能、故障和其他信息。管理员AP中的各种软件组件为管理员管理整个网络提供了必要的工具。

1. *术语*

在讨论两种管理服务元素的操作之前，先定义一些相关的术语。

术语**被管理对象**（MO）来源于面向对象设计方法，是整个结构的基础。被管理对象可以用来代表任何被管理的网络元素。另外，所有的被管理元素都要用MIB中的一个被管理对象来逻辑表示。因此，一个被管理对象是实际元素的逻辑表现形式，例如硬件、软件、数据结构等。它由下列几项定义：

- **属性** 对象的属性。
- **管理操作** 可应用于对象上的管理操作。
- **行为** 对象对各种操作的响应行为。
- **通知（事件）** 对象能创建的通知。



SMASE = 系统管理应用服务元素
CMISE = 公共管理信息服务元素
SMAE = 系统管理应用实体
MIB = 管理信息库

图13-23　OSI系统管理组件

虽然实际上正如网络管理系统关心的那样，一个被管理对象可以是一个复杂的设备元素的逻辑表示，例如网桥、路由器、计算机等。MIB中只有关于对象的属性、操作、行为和通知对于网络管理系统是可见的。这种表示风格是基于面向对象的设计特点封装的（或**信息隐藏**），这意味着所有的实现和其他细节对于网络管理系统都是不可见的。

一个属性通常指向SNMP中的一个变量，它是反映被管理对象当前状态的一个属性。一个属性可以有一个或多个相关的值，例如网桥端口、重传计数、定时器设置或路由表的内容。任何对某属性执行的操作都是直接指向拥有该属性的被管理对象，而不是直接指向该属性。
793 而被管理对象（规程/程序）如何执行指定的操作则是一个本地事件。

可以对一个被管理对象的属性执行以下操作：

- **获得属性值**　返回属性值。
- **设置属性值**　把特定属性设置为操作中给出的值。
- **恢复属性值**　把特定属性设置为它的默认值（或先前指定的值）。
- **添加属性值**　为由一组值组成的属性中添加额外值。
- **删除属性值**　从一组属性值中删除某个值。

另外，下面的三个操作是针对完整的被管理对象，而不是针对属性的：

- **创建** 创建一个新的被管理对象。
- **删除** 删除一个已有的被管理对象。
- **行为** 执行与该对象类型相关的行为。

当发生一个内部或外部事件时，被管理对象发送一个通知，例如，同邻近元素之间的通信丢失就是一个外部事件，而重传计数超过限制是一个内部通知。通知也称为**事件报告**。

被管理对象的行为与该对象的操作效果有关。它包括执行所选操作的规则，以及执行操作所得到结果。例子中包含有关创建和删除一个被管理对象以及执行这些功能得到的结果等规则。

所有的被管理对象都属于**被管理对象类**，它定义了被管理对象的类型。基于这个类，可以有很多被管理对象的实例，所有这些实例都具有相同的属性、操作、行为和通知。然而，每个实例被分配了一个惟一的名称，通过这个名称所有操作都与被管理对象类的指定实例联系起来。类的一个例子就是ISO-TP4传输协议，它可以有很多实例。

被管理对象类必须用标准的方法指定，因此它的属性值对于管理员AP和代理AP具有相同的含义。通常，管理员AP运行在高级工作站上，而代理AP运行在微处理机设备上，例如调制解调器或路由器。所有类都用ASN.1定义，并为整个系统维护了一个全部类库。一个被管理对象的所有属性值都被转换成具体语法形式，从而确保它们在所有的系统中都有相同的含义。

被管理对象类被组织成层次的形式，称为**继承树**。每个类是树中父类的子类。术语“继承”是面向对象设计的另外一个特性，它表明一个子类从它的父类中继承了所有的特性，即 794
属性等。当定义一个新类时，只需定义那些新的特性。例如，一个**被管理网桥**类可以定义成网桥类的子类。一个被管理网桥类的实例将有网桥类的实例的所有特性，并且还有网络管理软件。

被管理对象类和类的实例有着特定的**关系**。例如，如果两个对象实例是网桥1和LAN 2，则相互的关系可能是“网桥 1**连接到**LAN 2”。另一个关系是**包含**关系，它同术语“部分”的意思是相同的。一个例子就是“端口 2是网桥 3的一部分”。还有其他可能的关系，包括高层次的关系，例如“MOA被NM2管理”或“NM1被NM8管理”，其中MO指的是被管理对象，NM指的是网络管理员。图13-24显示了这些术语的使用情况和相互关系。

如同看到的，在管理员AP和代理AP之间交换的操作和通知是有关某个特定被管理对象的。中间的SMAE保证信息交换以及交互模式是以开放的方式执行的。而代理AP如何执行特定的操作或如何创建特定的通知则是本地的事情。

2. CMISE

公共管理信息服务元素（CMISE）通过SMASE为管理员（以及代理AP）提供了一组常规的服务，用于调用有关操作、通知和其他数据处理功能的远程管理规程。这些服务是远程操作（规程）类型；在某些情况中包含请求/回复（需要证实）电文传递，而在另一些情况中可能会包含一个单向的（无需证实）消息传递。CMISE使用ACSE和ROSE应用支持服务元素。 795
ACSE可以同对等的CMISE和ROSE建立联系，用来中继远程操作和通知调用请求（如果需要，还要传递响应）。表13-4给出了CMISE提供的服务的列表。

表13-4　CMIS服务原语

服　务	类型	支持ASE
M_INITIALIZE	C	ACSE
M_TERMINATE	C	ACSE
M_ABORT	U	ACSE
M_EVENT_REPORT	C/U	ROSE
M_GET	C	ROSE
M_SET	C/U	ROSE
M_ACTION	C/U	ROSE
M_CREATE	C	ROSE
M_DELETE	C	ROSE
M_CANCEL_GET	C	ROSE

C = 证实的　U = 未证实的

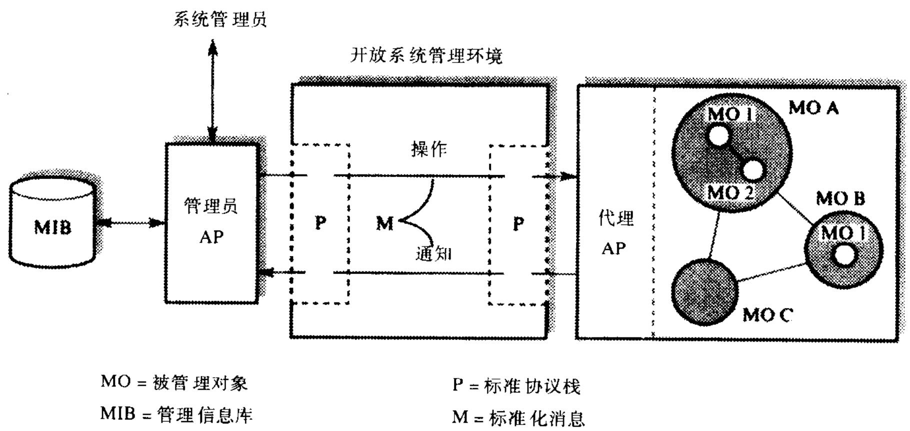


图13-24　管理员—代理交互示意图

CMISE提供的服务称为**公共管理信息服务**（CMIS），即通常的管理服务。它们的作用是使被管理对象执行解释每个服务信息的功能。M_INITIALIZE、TERMINATE和ABORT服务用来同一个特定的代理AP建立联系。所有的APDU直接映射成为等价的ACSE原语。同样，其他服务的APDU使用合适的ROSE原语进行传递。各种服务同前面列出的操作有直接的关系。

- M_GET是一个常规服务，它允许管理员AP从特定的代理AP请求检索管理信息。它是一个需要证实的服务，因为它有一个结果。一个例子就是检索一个被管理对象的当前状态。
- M_SET是一个常规服务，它使管理员AP能够请求修改被管理对象的某些管理信息。它可以是证实和无需证实的，这依赖于是否需要返回结果。一个例子是设置超时值，如果需要，返回确认。
- M_EVENT_REPORT使一个代理AP能够通知管理员AP有一个被管理对象事件发生，它也可以是证实的和无需证实的。
- M_CREATE、DELETE和ACTION都是关于一个完整的被管理对象，而不是关于对象属性的。一个例子就是使用CREATE创建一个协议实体的新实例，以及使用ACTION对

实例进行初始化。 796

M_CANCEL_GET用于取消早先的M_GET操作。它是一个比较有用的服务，例如，如果早先的M_GET操作返回一个非常长的结果串时，管理员可以终止该传输操作。

上面的这些服务都是常规服务。每个服务原语中都包含了**调用标识符**，接收方使用它执行前面定义的管理操作。接收方根据此对其他参数进行解释。例如，M_EVENT_REPORT原语包含的参数如下：

- 调用标识符——用于指定相关的通知，允许接收方把它同其他参数联系起来。
- 模式——证实的和无需证实的。
- 对象类——被管理对象的类。
- 对象实例——产生通知的特定被管理对象。
- 事件类型——事件的类型。
- 事件时间——事件发生的时间。
- 事件信息——关于事件的详细说明。

与CMISE相关的协议被称为**公共管理信息协议**（CMIP）。它为每个列出的服务原语产生一个等价的APDU。每个原语的参数在APDU中都有一个等价的字段。因此，当CMIP接收到一个服务原语后，它会创建一个等价的APDU，并使用ACSE或者ROSE提供的服务发送它。当接收到一个APDU时，它会执行相反的操作。

与ACSE相关的CMIS中的大多参数都被直接转换成对应的ACSE服务的等价参数。然而，每个原语中有一小部分参数形成CMIP的APDU，后者被放置在合适的ACSE原语的用户数据参数中传递。由于CMISE远程操作和通知与等价的ROSE服务之间的密切关系，它们通常被一起定义。为了阐述这一点，在图13-25中显示了包括ROSE操作的CMIP的定义。

回忆12.7节，ROSE没有实际的参数或变量。它们是被相连的特定应用ASE创建的。首先，定义了ROSE协议中的APDU类型，接着把ROSE APDU同CMIP APDU定义联系起来。因此CMIP APDU中的m-EventReport装入ROSE的ROIV PDU。而M_EVENT_REPORT原语中的**调用标识符**参数被分配给ROSE ROIVapdu的invokeID字段。CMIP中的每个APDU都有不同**操作值**；对于m-EventReport它是零。ROIVapdu中的变量字段是同CMIP相关的。因此在图13-25中它定义为与EventReportArgument相关。所有其他的CMIP APDU都是用同样的方式定义的。其他使用ROSE的ASE也都使用这种方法定义。 797

3. SMASE

现在，CMISE已经成为一种国际标准，很多电信和计算机连网设备的供应商都在各自的网络管理应用中采用。然而，它提供的管理服务只是常规服务。因为它把MIB的内容和结构定义以及对被管理对象的属性可以执行的操作和通知留给管理员和代理AP来实现。这对于中等的应用来说是完全足够了，例如单一网络的管理。但是，对于大的全球互连网络和电信系统的所有方面的管理，就意味着要在开放系统管理环境之外定义很多与管理有关的功能。附加的问题包括收费管理（管理访问费用或被管理对象的使用情况）和安全管理（用户试图获得某个被管理对象的访问权限，用户的访问控制和认证规程）。任何的系统管理工具都要从事这些主题。

回忆一下，组合的系统管理应用实体（SMAE）的目标不仅是提供管理各种组成互联网的网络元素的服务，例如通信线路、分组交换机、中间系统/网关、路由器、网桥与协议栈等，

它还要为整个**综合网络管理系统**（INMS）提供必要的服务。为了实现这个目标，如图13-23
798 所示，SAME在CMISE的上面包含了一个额外的服务元素，它称为系统管理应用服务元素（SMASE）。其功能就是在CMISE提供的服务的上面再提供一个更高层次的管理服务，它仍然属于开放系统管理环境的范围。

```
_ _ CMISE operations
ROSEapdus ::= CHOICE { roiv_apdu [1] IMPLICIT ROIVapdu,
                       rors_apdu [2] IMPLICIT RORSapdu,
                       rorer_apdu [3] IMPLICIT ROERapdu,
                       rorj_apdu [4] IMPLICIT RORJapdu }
ROIVapdu ::= SEQUENCE {  invokeID INTEGER,
                         linkedID OPTIONAL,
                         operation_value OPERATION,
                         argument ANY }
                |
                |
                |
ROIV_m_EventReport ::= ROIVapdu {
                         invokeID INTEGER,
                         linkedID ABSENT,
                         operation_value {m_EventReport (0) }
                         argument EventReportArgument }
EventReportArgument ::= IMPLICIT SEQUENCE {
                         objectClass [0] INTEGER,
                         objectInstance [1] OCTET STRING,
                         eventTimer [2] GENERALIZED TIME,
                         eventType [3] INTEGER,
                         eventInfo [4] ANY }
                |
                |
                |
END
```

图13-25　关于ROSE PDU的CMISE PDU定义示意图

SMASE由一组**系统管理功能**（SMF）组成。每个SMF都使用底层CMISE提供的一般服务执行它自身的系统管理功能。可以把整个SAME看作由一组**网络管理配置文件**组成的实体，每个配置文件在整个综合网络管理系统环境中执行特定的管理功能。SMF也被称为**管理应用服务**（MAS）。

每个SMF都提供了一组可以在被管理对象上执行的用于特定管理功能的服务。与每个SMF相关的是一组普通管理对象，这些对象关联到该功能并带有属性的定义和相关的操作和通知。关于每个被管理对象有一组ASN.1定义（模板）以及所有同对应CMISE原语有联系的参数。这样，把更高层次的抽象提供给管理应用，这是因为它只需指定被管理对象的名称以及要执行的操作和通知的参数列表。然后，SMASE使用相应的ASN.1模板创建同CMISE中的那些原语兼容的相应服务原语。

不同于CMISE，SMASE仍然处于开发中。计划过段时间会对各种应用中使用的OSI SMAE定义各种SMF/MAS。到目前为止已经定义了一些SMF。表13-5显示了一些SMF及其用法。

表13-5　系统管理功能的子集及其用途

系统管理功能	用　途
警告报告	定义由被管理对象创建的事件的语法和语义
事件管理报告	定义对事件报告的控制和过滤机制
日志控制	定义关于事件报告日志的建立和控制操作，以及内容选择等机制
安全警告报告	定义在违背安全的事件中创建的警告通知的语法和语义
统计记录	为了统计目的定义网络资源的使用记录和报告机制

（续）

系统管理功能	用　途
工作量检测	定义用于评价性能的关于网络资源使用的监控机制
统计摘要	定义获得网络资源操作特性的统计摘要的机制
对象管理	定义创建和删除对象以及管理对象属性的机制
状态管理	定义对象的状态以及检测和修改状态的机制
关系管理	定义在对象之间建立并维护联系的机制

799

4. MIB

OSI环境中的对象命名原则同13.1.5节中描述的TCP/IP协议族中的SNMP环境中的原则是完全兼容的。**管理信息树（MIT）**定义了在MIB中所有被管理对象的名称。树结构反映了对被管理对象的每个实例的包容规则。被管理对象的实例名称包含如下内容：

- 下一个层被管理对象的名称，它的上级。
- 一个额外的名称，在它的上级的范围内惟一标识它自身。

这样，被管理对象的名称由一列名称组成，这些名称开始于根结点的名称，终止于树中该对象的名称。一个对象的完整名称被称为**判别名**（DN），每个组成部分都是一个**相关的判别名**（RDN）。DN允许被管理对象在整个管理系统中被惟一地识别。然而，通常只需要在一个被管理对象的范围内定义一个名称，而不需要针对根进行定义，这种名称被称为**部分判别名**（PDU），由多个网络和引用单一网络范围内的对象名称的本地管理员组成的互联网就是一个比较好的例子。

除了MIT，MIB中还定义了两种其他类型的树结构。一个是**继承树**，另一个是**注册树**。继承树用来显示被管理对象类之间的**超类—子类关系**。在继承树中，特定的类只会显示一次，然而在MIT中，同一个类的多个实例可能会出现在树中的不同地方。注册树用来管理被管理对象的标识符的分配。记住，由于它们使用在CMIP APDU中，所以它们在整个管理系统环境中必须是惟一的。

13.2.5 MMS

生产消息服务（MMS）是为了满足完全自动化制造环境开发的ASE。在这种环境中，常常要求各种计算机设备之间以开放的方式相互交换报文。通常，每个制造部门都有一个相关的部门控制设备（控制计算机），该控制设备除了在全厂范围内与采用FTAM的其他系统通信外，还控制部门内各种计算机设备。通常，这种设备有机器人控制器（RC），数字化机械工具控制器（NC），自动导向运载装置控制器（AGV），可编程逻辑控制器（PLC）等等。MMS允许部门控制设备以开放的方式同各种自动化设备交换报文。

一个典型事件序列如下。首先，在工厂级使用FTAM，把有关部件或组件的信息传递给部门控制器。控制器使用MMS向该部门的各种自动设备发送适当的命令，这些命令会导致生产或组装（组件）。部门控制器（CC）会发送的命令如下： 800

- 在NC中选择并装入一组专门的工具。
- 请求RC选择一种原料，并把原料放入AGV。
- 命令AGV把原料传递给NC。

最后当任务完成，状态报文会返回CC。

服务原语

每个设备（CC、RC、NC等）使用的特定MMS服务原语是不同的，所以为每种设备定义一组服务子集。例如，同CC相关的服务有以下：

- 同一个特定的控制器建立联系（上下文管理）。
- 促使一个控制器从CC读取数据文件，包含工具信息或操作说明（获得文件）。
- 从CC中取出程序，装入控制器程序（装入程序）。
- 远程控制一个控制器的操作（作业控制）。
- 读取或改变（写）与控制器程序有关的所选变量（变量访问）。
- 请求控制器识别MMS支持的服务（识别）。

如同FTAM，服务原语分成一些功能单元（FU）；表13-6给出了与每个功能单元相关的原语。要注意，这些原语仅是一些例子，每个原语都有相关的参数。如表所示，MMS也为部门内本地使用提供有限的文件服务，例如传送包含与控制设备有关的部件列表。当然，它比FTAM要简单得多。

表13-6 MMS服务的子集

功能单元	服务原语	证实
上下文管理	Initiate	有
	Release（Conclude）	有
	Abort	无
获得文件	ObtainFile	有
文件传送	FileOpen	有
	FileClose	有
	（FileRead）	有
装入程序	LoadFromFile	无
	StoreToFile	无
作业控制	Start	无
	Stop	无
变量访问	Read	有
	Write	有
设备状态	Status	无
	UnsolicitedStatus	无
常规服务	Reject	无
	Cancel	有
	Identify	无

13.2.6 作业传送和处理

作业传送和处理（JTM）服务是分布在不同开放系统上的JTM SASE的集合。JTM元素（实体）构成了**JTM服务提供者**。FTAM并不真正实现一个文件服务（就是说，它只是提供了一种环境，供实文件系统以开放的方式访问和管理），同样JTM也只提供了一种环境，供与作业有关的文件（称为**作业规范**）在各个实（开放）系统之间传递与执行。实际上，传输的作业类型对于JTM服务提供者是透明的。

通过相关的UE提交某个OSI作业的规范的AP称为**发起代理**。作业规范完全地确定了所要执行的作业。例如，对于一个简单的JTM应用，作业规范可能是一个远程计算机系统上的AP运行的程序说明和程序数据，或者是一个远程打印机服务器AP要打印的文件。对于更复杂的应用，它可能是来自远程供应商计算机的关于某项设备的定单，或者是关于一个定单的发票或结算表。显然，与一个作业有关的处理类型不尽相同。在最简单的情况，可能只包含从作业规范中得到文件的处理，而在其他情况中，最初的作业规范可能会引出其他的相关作业规范（子作业）。 801

实际执行作业的AP称为**执行代理**，而从JTM服务提供者接收关于该作业的信息/数据的请求的AP被称为**源代理**，例如，本地文件库。由于在提交作业规范到作业被执行完毕之间的时间间隔可能会比较长，当发起代理提交作业规范时，它可指定一个AP遵循作业进度。这个AP称为**作业监视器**。在作业生存期内的任何时间如果发生了一个重要事件，JTM服务提供者就会创建一个**报告文件**并把它发送给相关的作业监视器。然后，发起代理可以作出关于当前状态作业的查询。最后，当执行代理完成所有与该作业规范有关的工作之后，JTM服务提供者会向一个推荐AP提交一个文件，该AP被称为**目的代理**。

可见，JTM主要涉及文件（作业）在AP之间的传送。因而，这些AP称为JTM服务提供者的代理。图13-26显示了与JTM有关的各种代理。虽然每个AP（代理）作为独立的实体表示，但一个或多个AP可以与同一个系统有关。例如，发起和目的代理可以在同一个系统中，或者目的和监控代理也可以在同一个系统中，等等。 802

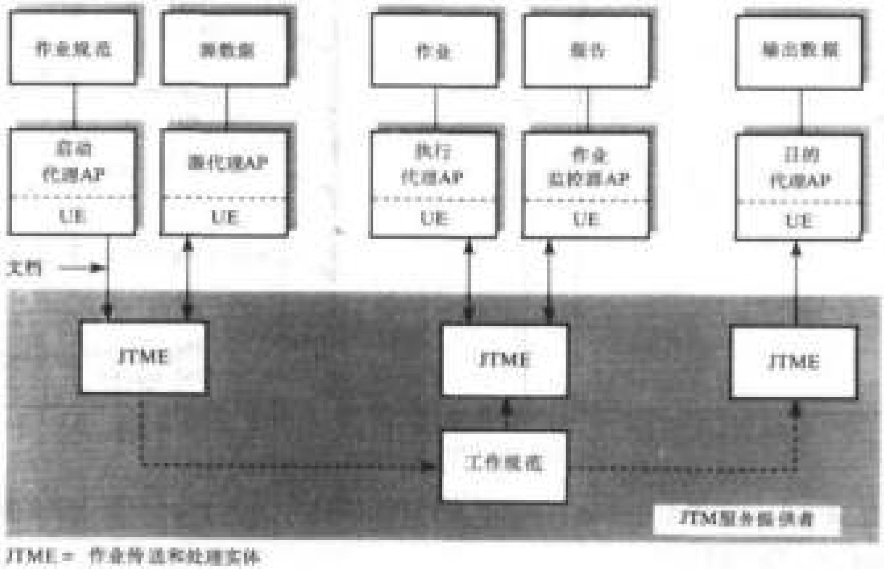


图13-26 JTM服务模型

JTM服务

完整描述与JTM有关的服务所需要的篇章太大了，此处只讨论**基本类**的用户服务。该类只支持一种严格的作业规范形式，即单个作业。单个作业不会导致任何子作业的发生，也不包含第二个监控代理，惟一的报告事件就是异常操作。

与基本类有关的用户服务原语如图13-27所示。发起代理（AP）称为**JTM服务请求者**，而从JTM服务提供者接收请求的各种代理称为**JTM服务响应者**。各种服务的用法如下：

- J_INITIATE_WORK 发起代理向它的本地JTM实体提交一个OSI作业规范文件。
- J_GIVE 允许JTM服务提供者的JTM实体请求来自源代理或执行代理的文件。
- J_DISPOSE JTM 实体把文件传递给执行代理或目的代理。
- 803 J_TASKEND 执行代理向本地JTM实体发送一个操作完成信号。
- J_STATUS JTM 实体获得作业操作进展信息。
- J_KILL JTM 实体突发地终止作业有关的所有操作。
- J_STOP JTM 实体临时性地暂停作业有关的所有操作。

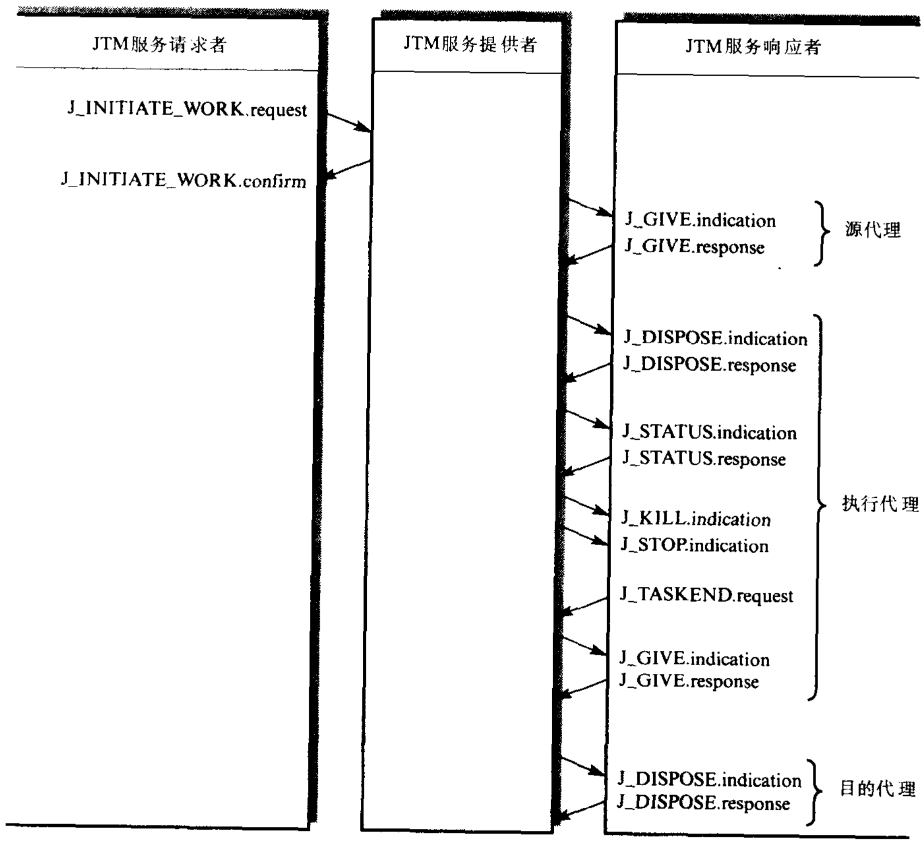


图13-27 JTM基本类服务原语

每个原语都有一些相关的参数。例如，J_INITIATE_WORK.request原语包括如下参数：

- 发起代理AP的名称。
- 本地标识符。
- OSI作业名称。
- 授权。
- 源代理AP的名称。
- 804 执行代理AP的名称。
- 目的代理AP的名称。

• 包含作业规范的文件的指针。

• JTM操作参数。

所有的原语中都包含JTM操作参数，它指明该原语的操作类型。这个操作既可以是文件传送，也可以是工作处理。例如，后者是J_DISPOSE原语中传递给执行代理的。通常，响应和证实原语只包含本地标识符，用于相应的提示和请求原语。

13.2.7 DTP

分布式事务处理（DTP）ASE用来控制包含多个开放系统的事务处理系统中的操作。回忆第12章，许多分布式应用都包含同时请求共享资源的多用户（客户）访问。

一个例子就是银行业应用中包含客户账号信息的数据库或文件系统，其中分布式客户系统群个体对该系统进行读或更新操作的访问。为了保证数据库/文件系统中的内容的一致性以及反映所有已经发生的操作，定义了CCR ASE，它帮助控制这类应用。在13.2.2节讨论FTAM ASE的可靠服务时，描述了涉及单一资源的CCR的应用。

除了这种应用，CCR还支持对包含多个操作系统的多事务的控制。这种请求通常是多个分布式事务处理的应用。为了与CCR嵌套服务一起应用已定义DTP ASE。

包含多个操作的事务的执行要遵循如下一些规则：

• 原子性　操作全部被执行或者全部没有被执行。
• 一致性　一个成功的事务操作把处于一种一致状态的数据转换成另一种一致状态。
• 独立性　操作的中间结果不能被外部访问，如果同时有几个事务在执行，要控制它们使它们看起来是顺序执行的。
• 耐久性　如果执行操作时发生了故障，可以被恢复成一致状态。

它们通常称为**酸性规则**。DTP ASE遵循这些规则为包含在分布式事务中的AP之间的信息交换提供控制服务。 805

通常DTP系统包含多个AP，它们在彼此之间执行相互关联的事务。一个例子就是访问多个航线预定系统的分布式客户端系统群体。显然，多个预定操作会导致几个系统之间的多个相关联事务发生。而我们的目的就是要把这些事务控制成一组等价的对系统的顺序访问。并且，如果在一系列的事务中间，客户系统发生故障，那么必须能够发现该故障，使每个服务器能够恢复到一致状态。

服务

一个完整DTP应用实体由DTP ASE和ACSE以及CCR ASE提供的支持服务组成。一个分布式事务处理应用如图13-28所示。

每个客户AP都可能会发起一个包含多个服务器的事务，因此也发生多个子事务的事务。DTP服务的目标就是为客户提供服务：它使客户能够同其他客户同时执行上面提到的这种事务。用户AP包含的每个（分布式）事务都被组织成树结构，其中发起原始事务的客户作为树的根结点。根AP称为支配者（上级），它会发起多个服务器的事务，每个都称为一个下级或被支配者。在更复杂的应用中，当执行一个事务中的操作时，下级可能会试图执行一个子事务。一个例子就是是否有服务器上的一个文件的多份拷贝。因此，在对一个提交请求作出响应之前，服务器可能会发起更新各拷贝的事务，这就是创建**事务树**。

事务处理系统有各种等级的复杂性。在某些情况，可能只需要控制对单一资源的访问；而在其他情况，可能需要控制多个同步发生的包含事务树的事务。因此，把DTP提供的服务 806

分成若干个功能单元（FU），每个单元提供了一个控制功能。各种FU包含的服务如下：

- **核心**　FU提供了用户同一个服务器建立联系，发起一个事务（开始一个对话），执行该事务的任一操作（数据传输），向发起方通知差错，关闭事务以及联系（终止对话）所需要的服务。
- **提交**　FU提供了用户需要的附加服务，它使客户根据酸性法则执行事务，包含了CCR提供的两阶段提交和回退差错恢复等服务。
- **极性控制**　FU提供了允许单一服务器在某一时刻只能被一个用户访问的服务。
- **解除事务**　FU允许上级阻塞一个事务子树的创建，并在之后的时间内允许它执行。
- **握手**　FU允许两个客户保持一个事务中的处理功能的同步。

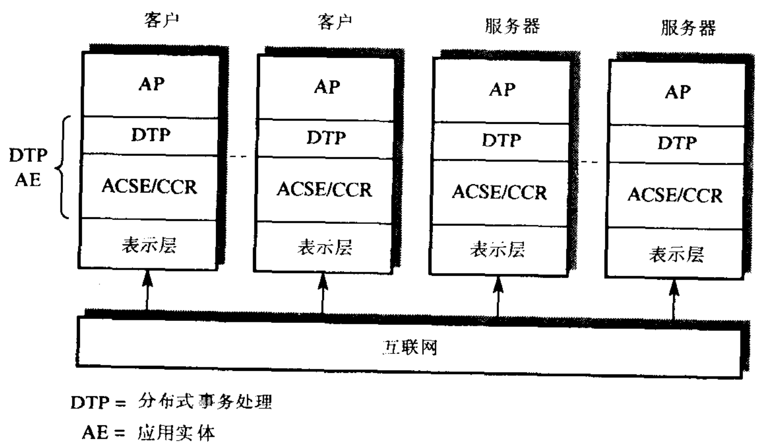


图13-28　DTP应用示意图

习题

13.1　借助图连同描述性的语言解释下面这些TCP/IP和ISO中的术语的含义：

（a）用户请求/响应

（b）虚设备请求/响应

（c）实设备请求/响应

指出这些术语的实现和结构在两者中的不同点。

13.2　解释TELNET协议中下列术语的含义：

（a）虚拟终端

（b）网络虚拟终端

（c）伪终端

使用表13-1中列出的命令，给出在TELNET客户端和服务器协议之间进行如下选项协商的命令序列：

（d）把7位ASCII转换成8位二进制数

（e）回显收到的每个字符

假设客户首先发送一个请求，然后是服务器。在8位二进制数模式中，如何发送FF（hex）字节？

13.3 解释文件库中下列术语的含义：
(a) 无结构文件
(b) 结构文件
(c) 随机访问文件
(d) 平面文件
(e) 层次结构文件
解释TCP/IP协议族中的FTP如何处理前三个结构。

13.4 列出TFTP中的四种报文类型并解释它们的含义。假设一个包含1K字节的无结构文件，使用时序图表示该文件是如何利用刚刚描述的报文类型在TFTP客户端和服务器之间传递的。

13.5 解释e-mail系统中下列术语的含义：
(a) 邮箱
(b) 本地邮件
(c) 邮箱名称
(d) 邮件头和关键字

13.6 借助时序图说明一个SMTP的命令电文交换序列，图中的命令把邮件从一个邮件系统发送给另一个邮件系统，并把邮件的副本发送给第三个系统。 807

13.7 借助图连同描述性语言说明术语“操作环境”和“网络管理环境”之间的相互关系。在图中要包含与主机和网关相关的协议族。

13.8 用示意图表示包含多个LAN并用路由器互连的校园网的管理信息树。列出五个SNMP的服务原语，并用相关MIT解释它们的功能。定义对应每个服务原语的PDU名并用ASN.1定义其中一个PDU实例。

13.9 在OSI协议栈的上下文中，解释如下术语的含义及其相互关系：
(a) 服务元素
(b) 特定应用服务元素
(c) 应用支持服务元素
(d) 应用实体
给出最后术语的一个实例，并指出与其他术语的关系。

13.10 借助示意图，解释ISO VT协议中如下术语的含义
(a) 虚终端
(b) 用户元素
(c) 概念通信区域
确定组成概念通信区域的数据结构，并解释它的功能。

13.11 利用示意图，描述ISO FTAM协议中使用的虚文件库模型中的文件元素如何确认以及它的结构。包括如下术语的含义：
(a) FADU
(b) DU
(c) 数据元素

13.12 借助状态变迁图，解释FTAM用户服务中的四种服务区间。描述从一个命名的文件读

取数据单元的一般顺序。明确解释F_READ/WRITE服务的操作。

13.13 区分FTAM中的“常规传输”和“可靠传输”模式。列出可靠传输模式的附加服务，并解释它们如何关联到CCR应用支持协议中的服务。

13.14 借助示意图，解释ISO邮件标准MOTIS中的术语：

(a) UA

(b) IPM协议

(c) SDSE

(d) MTA

(e) MS

识别MOTIS中的各种协议，以及在图中它们是如何关联的。给出每个协议功能的简单描述。

13.15 解释ISO MHS中的电文格式，以及题13.14中如何用标识的各个协议对各个部分进行解释。

13.16 确定同公共X.400标准兼容的校园e-mail系统中使用的各种要素，以及相关的协议，并解释邮件如何发送：

(a) 系统内部

(b) 系统外部

13.17 借助示意图说明在OSI管理系统中组成SAME的各种ASE，并解释每种服务元素的功能。描述中包括对术语“SMF”的解释。

13.18 解释ISO管理协议与TCP/IP中的SNMP的不同。

13.19 利用OSI管理框架，解释如下的术语：

(a) 被管理对象

(b) 属性

(c) 操作

(d) 行为

(e) 通知

808 给出一个对被管理对象的属性执行操作的例子，以及被管理对象如何产生通知。

13.20 解释用于管理对象的下列术语含义：

(a) 信息隐藏

(b) 被管理对象类

(c) 继承树

(d) 包容关系

(e) 被管理对象实例

(f) 管理信息树

(g) 判别名

(h) 相关和部分判别名

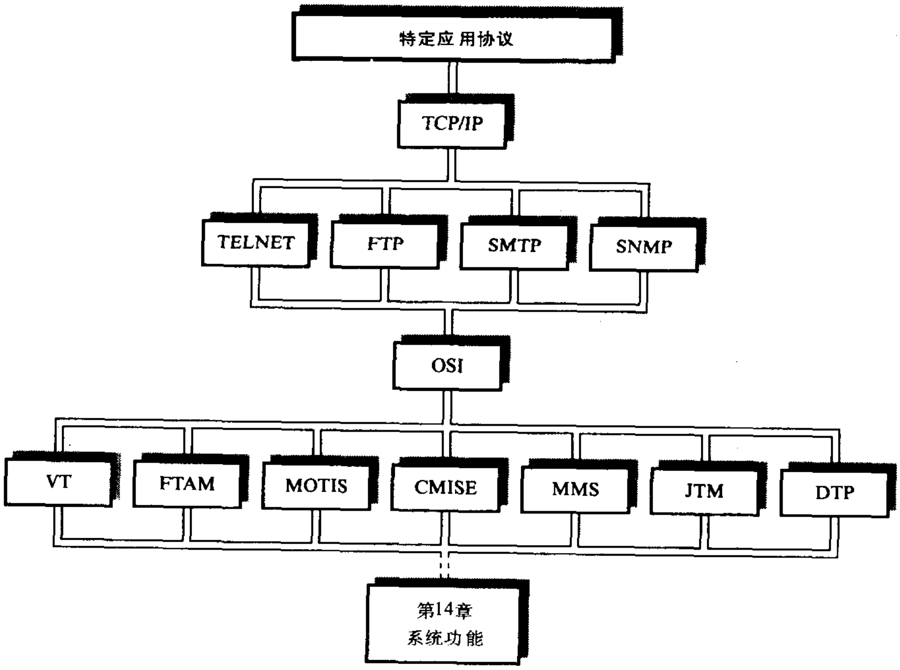
13.21 描述以下ISO ASE的功能和服务：

(a) MMS

(b) JTM

809 (c) DTP

本章概要


特定应用协议
TCP/IP
TELNET
FTP
SMTP
SNMP
OSI
VT
FTAM
MOTIS
CMISE
MMS
JTM
DTP
第14章
系统功能

810

第14章 系统综述

本章目的

读完本章，应该能够：

- 解释TCP/IP协议族使用的域名系统的结构和操作，以及与开放系统环境相关的名称如何管理；
- 解释X.500目录系统的结构和操作，以及其中的相关协议；
- 了解MAP和TOP协议族的体系结构，以及预期的应用；
- 理解构成TCP/IP协议族和OSI协议族的各种协议彼此之间如何交互，以实现特定的用户应用服务；
- 理解如何构造与本地操作系统有关的协议族；
- 解释与组成协议族的各个部分的实现相关的重要问题。

引言

第11章到第13章主要侧重于描述为了在一定范围的应用环境中建立开放系统所定义的各种协议实体的功能、操作和规范。本章将讨论基于这些开放标准的完整通信子系统的操作和实现的有关问题。

回忆一下，用户应用使用符号名称实现通信的目的而开放系统互连环境（OSIE）中使用

811 数字地址。因此，必须首先讨论一下开放应用中的协议族如何执行名称到地址的映射功能以及如何管理分配给用户的名称。这些功能都是目录服务的一部分。

本章还会讨论一些基于TCP/IP或ISO协议的开放系统环境的实例，并且详细地解释组成开放系统协议族中的协议如何协作，以及彼此之间如何交互执行所需的通信支持功能。然后，考虑两个完整的协议族的实现。最后，简短地讨论一下除了通信标准以外的标准。

14.1 目录服务

正如已经提到的，OSIE中使用地址来识别一个网络会话中涉及的源和目标应用进程（AP）。地址由两部分组成：一部分由网络/互联网用来把信息路由给要求的主机/端系统，另一部分由主机/端系统用来把接收到的信息路由给要求的AP。

其中有关网络的部分与电话号码很相似。如果所有的呼叫仅仅包含在一个区域（一个PABX或一个专用网络），那么这个号码可能会相对来说比较短小，这是因为只需要在有限的环境中确定电话出口。在计算机网络中，这与LAN网络中使用的连接点地址是相似的。然而，如果电话是连接到一个PTT或者公共载波网络，电话号码就必须加长，这样才能跨越国家识别和发送电话呼叫以及必要的国际通话。国际电话号码与X.25 PSPDN中使用的X.121地址是相似的，如果使用了多种网络类型，它与其中的IP或ISO CLNP地址也是相似的。在这两种情况下，地址确保了在整个网络/互联网中主机/端系统地址是惟一的。

地址中的第二部分用来识别主机中特定的应用进程，它在两组协议族中的形式有所不同。

在TCP/IP中，这是端口地址的职能；而在OSI中，因为在传输层和应用层协议之间具有多个中间层，这个职能是由层间地址选择器/服务访问点TSAP、SSAP和PSAP执行的。每个协议族使用的地址如图14-1所示。

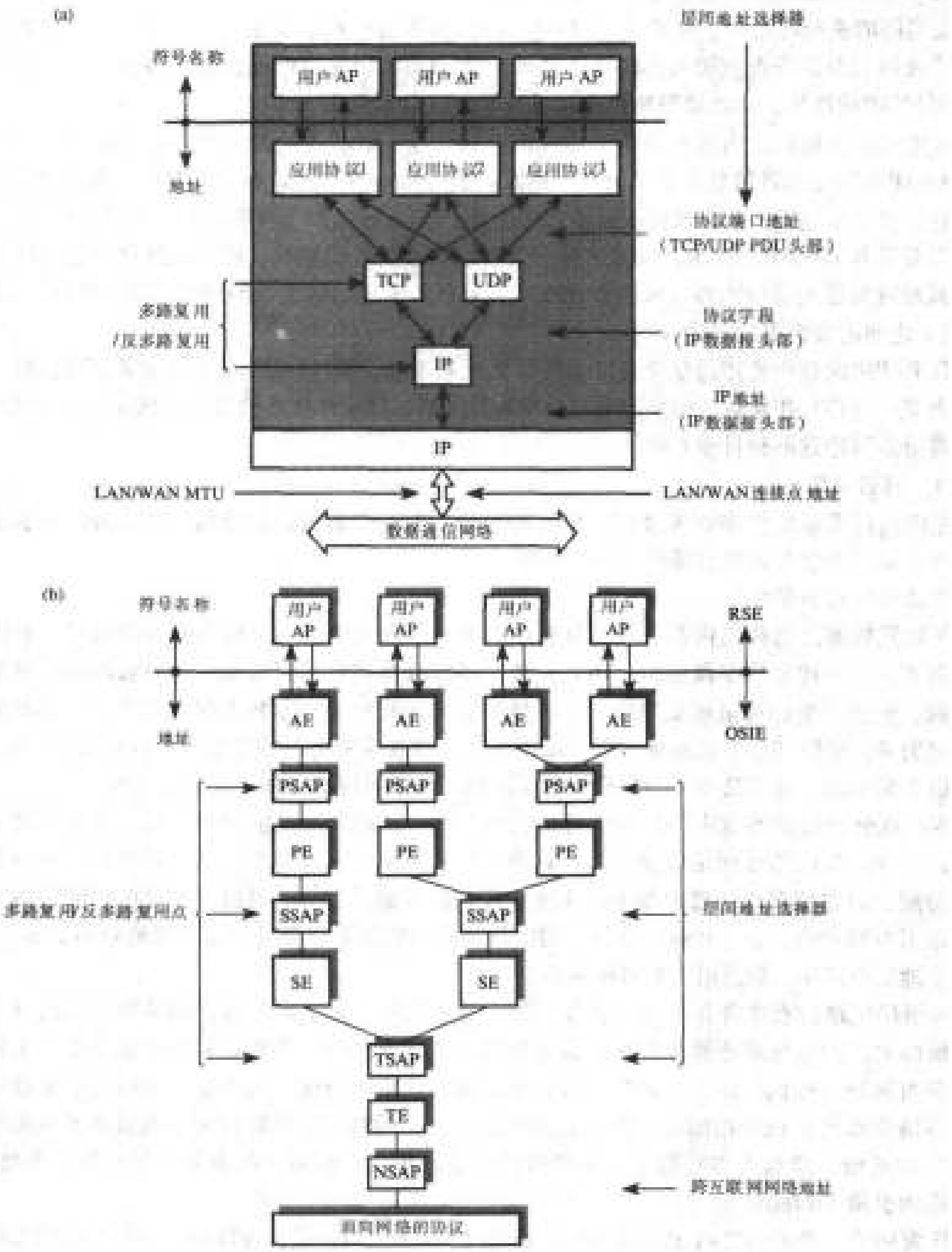


图14-1 地址组成成分

(a) TCP/IP (b) OSI

两组协议族的地址都是由一组有意义的数字组成。在TCP/IP中，IP地址是32位整数，用点分十进制法表示，最大可达12位十进制数。端口号是3位十进制数。在OSI中，有关网络地址可长达40位十进制数，包括附加的用于层间多路复用/反多路复用的位。因此，在OSIE之外即使是以十进制形式使用地址都是不方便的。

812~813 在实际的系统环境中，用户（人或AP）称为**符号名称**而不是地址。同样的，电话系统中的用户使用目录服务查找被叫方的电话号码，因此OSIE中的**目录服务**用来查找一个已命名的目标用户/AP的地址。而**地址解析服务**是目录服务中最常使用的。

虽然对于长地址使用名称是必要的，但还有其他一些必要的原因。在发生改变时，名称把用户/AP从网络配置信息中独立出来。例如，子网（LAN）需要增加或移除，或者用不同的公共数据服务互连子网。在某些应用中，一个用户AP能够从某个网络中的位置移动到另一个，而不需要通知其他AP。因此，目录服务不仅要提供地址解析服务，还要提供允许包含符号名称及其对应地址的目录内容以可控方式改变和更新的服务。这时，AP的符号名称与它的地址（位置）之间的**绑定**是动态的。

TCP/IP协议族中使用的整个目录系统称为**域名系统**，而在OSI中使用的是**X.500目录**。虽然后者是一个ITU-T建议，但它是与ISO共同开发的。ISO中使用的目录系统基于这个标准。下面将分别讨论这两种目录系统。

14.1.1　域名系统

在任何目录系统中都要考虑两个相关的主题，第一个是名称的结构以及已分配的名称如何管理；第二个就是各种目录服务如何实现。

1. 名称结构和管理

任何互联网的名称结构都是非常重要的，因为它极大地影响了目录服务的效率。有两种基本的方式。一种采用**平面结构**，而另一种采用**层次结构**。平面结构更加高效地使用整个名称空间，但是产生的目录要集中管理。于是在大的互联网中，通常要保持多个目录副本来加速目录服务，所以使用平面地址空间意味着当有目录改变发生时所有目录备份都要进行更新。由于以上的原因，除了最小的网络外，所有网络的目录系统都使用层次命名结构。

再一次分析电话系统中用户号码的结构与分配。最高层次是国家码，接下来是国家的区位码，等等。号码的分配是以分布的方式而不是集中的方式管理的。在国际级别管理国家代码的分配，而在国家级别管理每个国家区位代码的分配，等等，最后，本地区域的号码是在
814 地区范围内管理的。于是知道，在地址层次结构的对应级中只要每个较高级的号码是惟一的，在整个地址空间中，最后组合的号码就是惟一的。

采用层次结构意味着各种目录服务会更有效地执行，这是因为可以用某种方式把目录分区，使得更多的地址解析操作和其他服务都可以在本地执行。例如，如果名称是根据主机的地理分布进行分配的，那么目录就可以按类似的方法进行分区。因为多数的网络事务以及目录服务请求都是在位于相同地区的主机之间进行的，例如工作站群体和本地服务器或邮件系统之间的通信，所以大多数服务请求都可以在本地解决。相对少的事务必须引用其他地点，这被称为**引用**（referral）。

作为例子，考虑因特网中使用的层次地址结构。回忆第9章，因特网由一些互连的互联网组成。这些互联网是最近才连接起来的，在创建因特网之前，它们都有自己的命名层次。为了适应这种情况，在层次结构（树）的最高级有许多可能的分区，称为**组织**或**域**，域名在表14-1中列出。

表14-1 因特网使用的域名

域名	含义
COM	商业组织
EDU	教育机构
GOV	政府机构
MIL	军事组织
NET	（因特网）网络支持中心
ORG	其他组织
INT	国际组织
USA	
UK	按地理位置分配的国家代码
⋮	
⋮	
⋮	

因特网中连到网络或子网的所有主机必须用某个组织域名注册。因特网的整个目录是根据这些域名进行分割的。下面注册的域的选择要使引用的数量最小化。因此，如果一台主机是连接到教育机构的一个网络上，很可能这台主机的多数传输都会涉及本地网络或其他教育机构中的主机，它注册到EDU域名。相似的，如果一台主机连接到属于军事组织的某网络，那么它应该注册到MIL域名，等等。

每个域名都使用了一个相应的命名层次。在EDU域名中，下一层次是各个教育机构的名 815
称，而在COM域名中下一层次是各个注册的商业机构。一般方案与命名转换如图14-2所示。

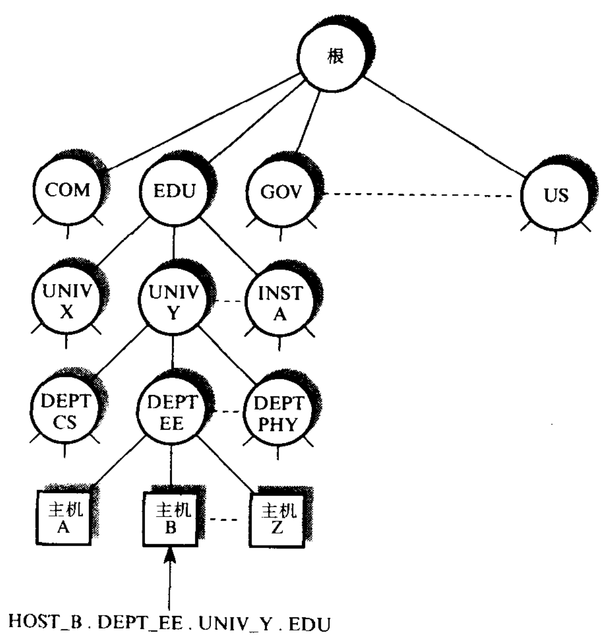


图14-2 域名层次

域名中的各个成分被称为**标号**。见图14-2，标号的书写是按本地标号在左，上层域名在右的形式，各个标号之间用逗点进行分隔。

每个标号名前缀都仅仅是为了阐述一个典型层次细分。实际上，每个标号都是层次体系中相应层次的注册域名。

如早先提到的，采用层次结构意味着名称可以在本地分配而不用集中分配。因此在教育机构中，一旦机构（名称）已经在EDU域名机构注册后，此机构的授权管理者可以为连接到本机构网络上的主机分配名称和IP地址。

2. 域名服务器

每个机构的网络中都有一个相应的主机，它运行称为域名服务器（DNS）的应用协议/进程。与此相关有一个称为目录信息库（DIB）的数据库，它包含了此机构中所有的目录相关信息。当一个新的主机注册时，管理者需要把分配给这台主机的名称、IP地址和其他信息交互地输入本地域名服务器的DIB中，用户就可以进行涉及因特网的处理了。
816
当发起特定应用协议的网络处理时，终端用户或AP将使用服务器所在的主机名与本地客户端协议进行通信。在协议建立与服务器的传输连接之前，它必须确定运行服务器的主机的IP地址。在TCP/IP协议中，所有的服务器协议/进程都被分配了一个固定的端口号（知名端口号），因此DIB中无须保存端口号的信息。

为了得到一个已命名的服务器的IP地址，每台主机中都装有一个称为名称解析器的客户协议/进程。它的位置以及为了执行名称到地址的映射客户协议遵循的事件顺序如图14-3所示。

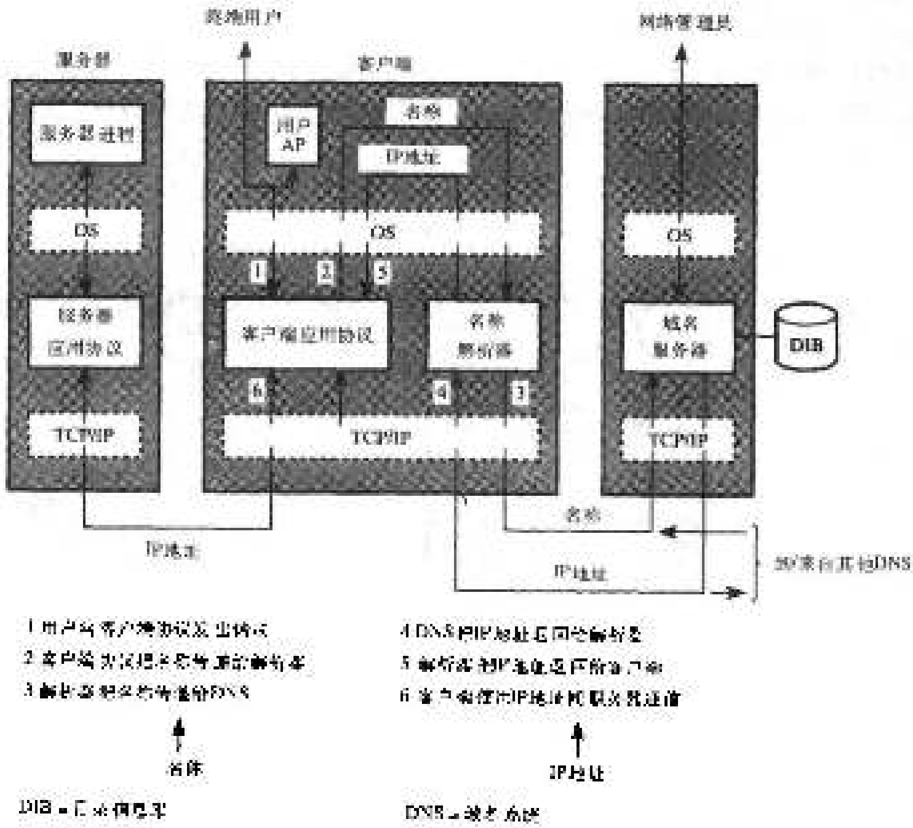
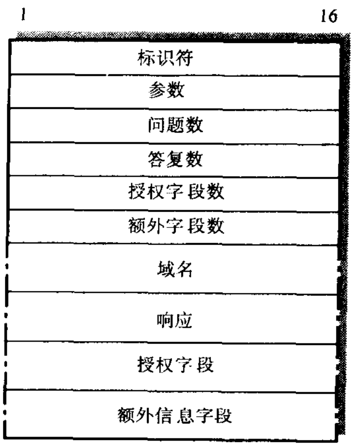


图14-3　名称到地址解析协议和顺序

接收到名称后，客户端应用协议使用本地操作系统提供的标准进程间通信原语把名称传

递给名称解析器。然后，解析器会按域名服务器协议中的标准消息格式生成一个解析请求消息。格式如14-4所示。

(a)



(b)

位顺序	含义
1	0=请求 1=响应
2–5	0000=标准 0001 = 反向
6	1=授权回答
7	1=可截断报文
8	1=期望递归（询问）
9	1=可用递归（响应）
13–16	0000=无差错（响应） 0001=格式错（响应） 0010=服务器失败（响应） 0011=名称不可知（响应）

图14-4　域名服务器协议
(a) 信息格式　(b) 参数位的定义

解析器在任何时刻可以有多个请求，因此**标识**字段用于把后续的响应消息与先前的请求消息关联起来。**参数**字段给出消息的类型（请求/响应）和与响应（格式错误、服务器失败等 817
等）有关的额外限定符，响应中还包含了服务器的信息（**授权**和**IP地址**）。另外，每个请求和响应消息中可以包含多个请求/响应。

名称解析器使用标准的TCP/IP和消息把请求消息传递给本地域名服务器。如果请求的服务器（主机）位于本地网络，本地域名服务器从它的DIB中得到相应IP地址，并在应答消息中返回。如果本地域名服务器没有相应的地址，它产生一个请求消息，向另一个服务器寻求帮助，这称为**引用**。目标是使需要的引用次数最小。为此，服务器按层次结构组织，如图14-5(a)所示。

可以看到，每个组织树的高层次的服务器，同在此域名层次中注册的组织/机构的名称服务器合作，构造了一个包含所有名称和IP地址的表。接收到一个它不能解析的请求后，本地域名服务器产生一个引用请求，并把它发送给更高层次的服务器。服务器首先会检测请求中的域名，并假设它位于本域名中的其他服务器上，然后从自身的名称表中使用这个名称访问IP地址。接下来在给本地服务器的响应消息中返回这个消息，它会向所要求的服务器发送一个直接请求，获得要求的IP地址。消息的交换顺序如图14-5(b)所示。

相似的是，解析与其他域名中的组织有关的请求时，更多的转移必须由更高层的服务器来进行。在所示方案中，根服务器有一个包含所有高层域名服务器名称和IP地址的表格，当接收到包含其他域名的请求后，它经过根服务器转发请求，所要求的域名服务器如前面一样会直接响应。

实际上，中间域名服务器中的信息的数量是比较小的，因为它的转移表中的条目数量是 818
由机构/组织的数量决定的而不是由每个位置上的主机数决定的。为了减少交换的消息数量，另一种结构如图14-5(c)所示。使用这种方法，根服务器包含所有注册机构中所有服务器的IP地址。每个转移请求最多经过4次消息交换可以得到回答。

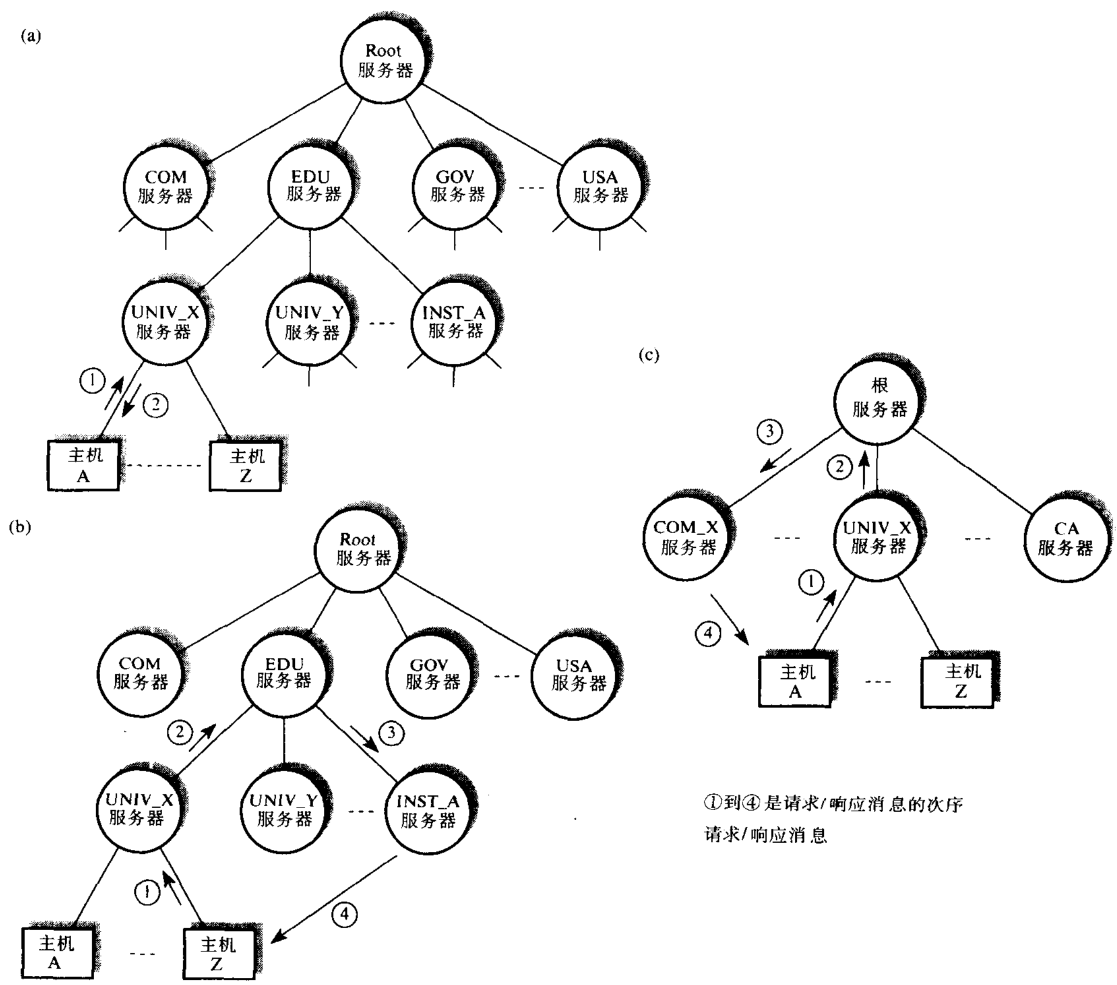


图14-5　域名服务器层次：

(a) 本地请求　(b) 高层请求　(c) 折叠层次

虽然这种方案降低了消息交换的数量，但是如图中所示，它导致了互联网中负载的增加，这是因为要解析每个请求，需要在网络上发送大量的消息。为了解决这种现象，每个本地服务
819 器都保持了一个关于最近引用名称（它们的IP地址）和提供这个地址的服务器名称和IP地址的记录。这些消息被保存在**名称缓存**中。接收到一个没有本地条目的请求后，本地服务器会搜索它的名称缓存，如果缓存中有，就把它作为响应消息。在响应消息中，把授权标记置为0，把提供回答的服务器名和IP地址放在授权字段。这样就允许消息可以是过时的，例如网络重新配置后。接收到答复后，客户端或者接受这个信息或者直接向名称服务器发送一个新的请求。

为了确保名称缓存表中的条目是在时效范围内的，对于每个条目都有一个超时时间。这个值是由提供信息的服务器指定的，因此每个条目的超时时间可能会有不同。一旦一个条目超时时间到达，这个条目就会删除，而关于这个条目的任何新的请求都会被转移。

14.1.2　X.500目录

OSI协议族中的目录服务使用的是X.500标准。这个标准由若干个部分组成，每个部分涉及整个目录功能的不同方面。整个系统被简单地称为**目录**。目录中的结构和服务都有相应的

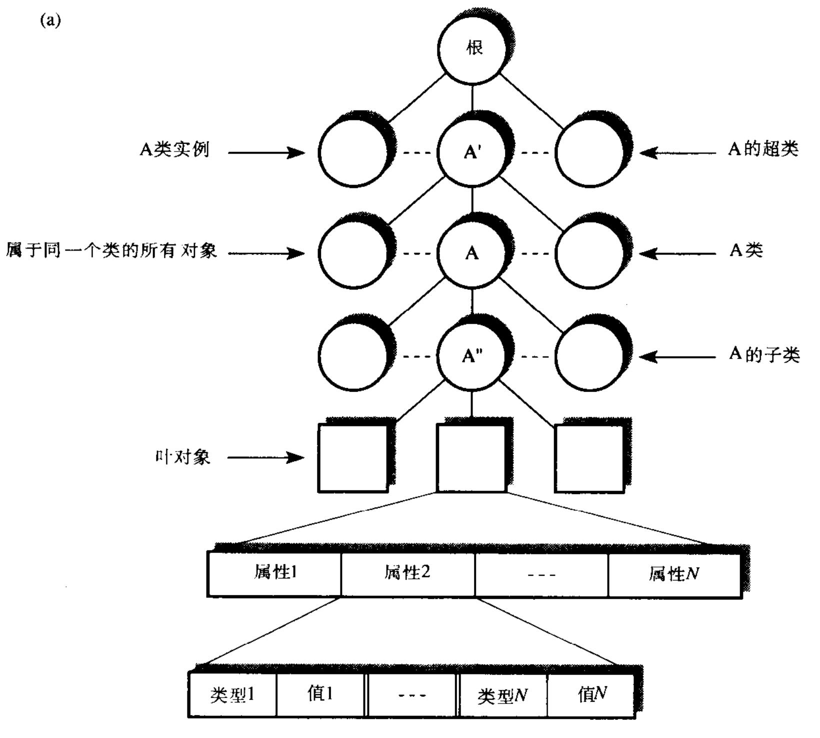
定义，所以它不仅可以对OSI协议族中的AP提供目录服务，还可以对其他应用提供服务，例如系统管理应用中的管理进程，或者X.400消息传递系统中的消息传递代理。

1. 信息和目录模型

目录的结构和命名惯例都基于面向对象的设计原则，这些已经在13.2.3节中描述过。它所使用的术语与X.400/MOTIS消息处理服务中使用的是相似的，见图14-6(a)。

DIB中的所有条目都被称为**对象**，它是特定的**对象类**的**实例**。对象被组织成树结构，这个树称为**目录信息树**（DIT）。在分支层次中特定级别的所有对象都属于同一对象类。对象类中邻近高层中的对象是它们的超类中的成员，而邻近低层次树中的对象是它们子类中的成员。

(a)



(b)

```
--属性数据类型--
    Attribute ::= SEQUENCE {
                        type AttributeType
                        values SET OF Attribute Value
                        -- at least one value is required -- }
    AttributeType ::= ObjectType
    AttributeValue ::= ANY
    AttributeValueAssertion ::= SEQUENCE {
                                AttributeType, AttributeValue }
--命名数据类型--
    Name ::= RDNSequence
    RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
    DistinguishedName ::= RDNSequence
    RelativeDistinguishedName ::= SET OF AttributeValueAssertion
```

图14-6 DIT结构

(a) 术语 (b) ASN.1的属性定义

每个对象类（对象）都有一组**属性**，每个属性都有一个**类型**和一个或多个**属性值**。属性类型识别对象的类，例如国家代码C。至少有一个属性值是对象的名称，例如UK。每个对象

的属性结构是与叶对象相关联的，虽然消息树中的所有对象都具有相同的结构。属性结构的
820 形式化定义以及对象名称的术语定义在图14-6(b)中给出，两者都是用ASN.1表达的。

命名惯例与域名系统中使用的原则是相似的。然而，对于X.500，每个标号都是**相对判别名**（RDN）（同X.400）和与消息树中条目（对象）相关的标号完整表，称为**判别名**（DN）。对于域名系统，所使用的RDN必须在分层中的任何级都是惟一的，所以，它们要由那一级的管理机构负责分配。图14-7(a)显示了一个DIT的分配。例子中所使用的名称涉及到一个X.400邮件系统。

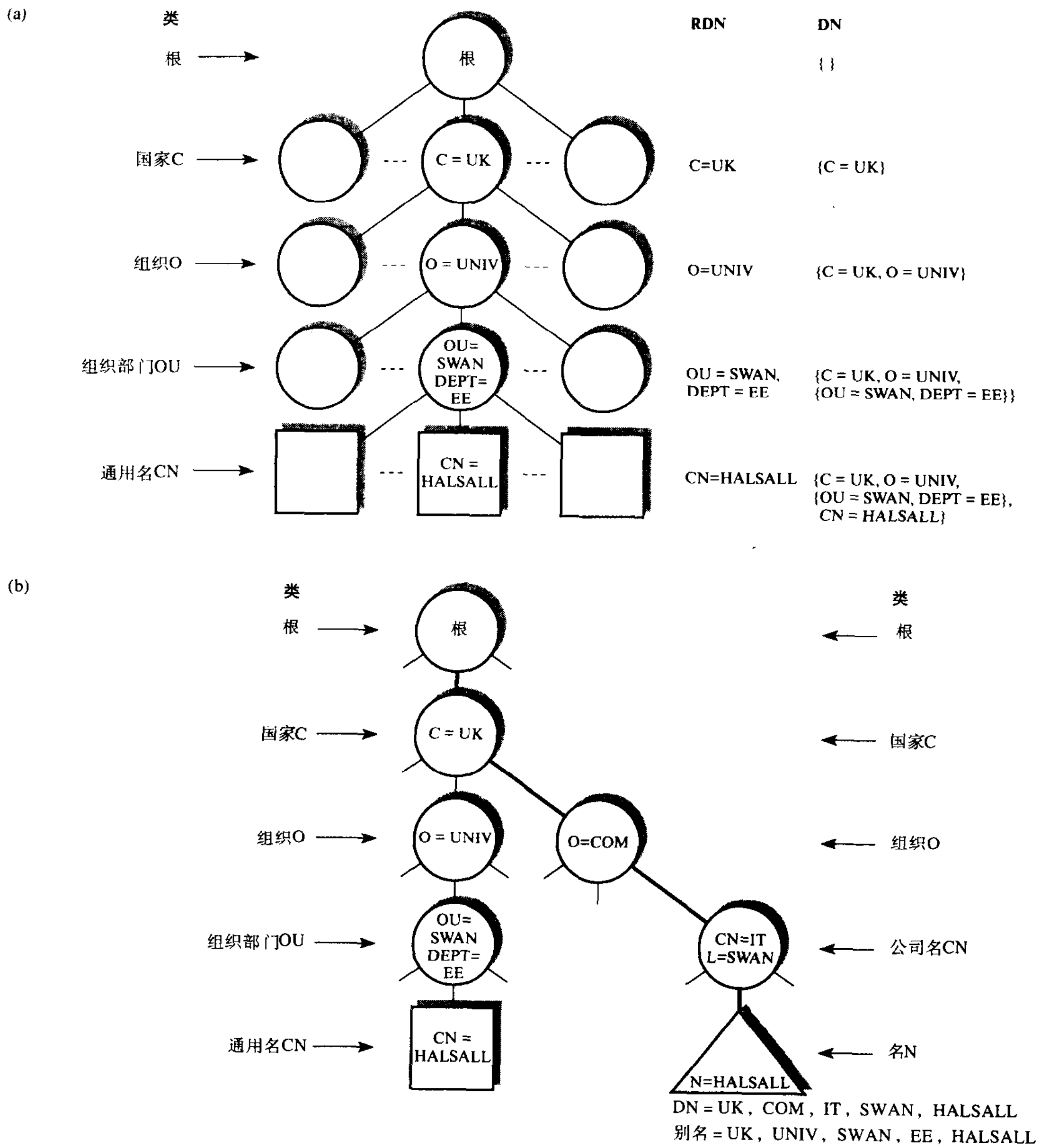


图14-7 DIT内容的实例

(a) 电子邮件名 (b) 别名

在实例中，所有的类（属性）类型包括国家（C）、组织（O）、组织部门（OU）和通用名（CN），而UK、UNIV、SWAN是相应的属性值。注意，对象类中的组织部门有一个是DEPT类型的附加属性，它的值是EE。在树中每个对象名的RDN和DN在树的旁边给出。 821-822

除了对象名条目外，带有别名的条目包含在DIT的叶中。除了有DN外，DIT中的别名（名称）条目常指向DIT中不同部分的对象。这个对象不需要是叶条目，因此别名允许一个对象属于多个DIT分支，对象可以有多个名称，实例如图14-7(b)所示。

实际的树是图14-7(a)中树的扩展。一个对象COM（公司）被定义成机构类（O=COM）。它有一个公司名称子类（CN=IT，L=SWAN），公司名称也有一个名称子类（N=HALSALL）。最后是一个叶结点，它是其他机构分支中HALSALL的一个别名。因此有一个附加别名属性，指明了树中其他条目的DN。

2. 目录服务

目录服务模型如图14-8所示。所有的用户使用一个称为**目录用户代理**（DUA）的AP访问目录。正如前面指出，目录要满足一定范围的不同应用。因此用户既可以是一个人也可以是一个AP。为了解决每个请求，DUA使用称为**目录服务代理**（DSA）的AP与目录交互。DUA同TCP/IP中的名称解析器以及域名服务器中的DSA起着相似的作用。 823

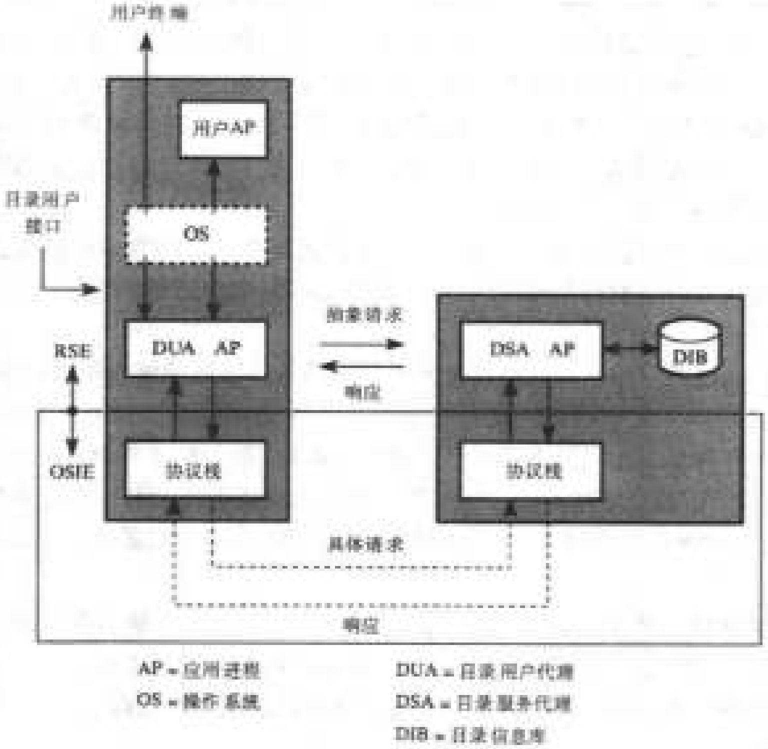


图14-8 目录服务模型

DUA提供给用户的服务包括**目录查询**和**目录修改**两类。它们的含义见表14-2。

多数的请求都有若干个限定符。例如，用户可以设置时间长度的限制、搜索的范围和请求的相关优先级（**控制限定符**）。另外，目录条目可能会有相关的安全机制，例如一个数字签名（**安全限定符**），它必须在发送请求之前给出。如果请求中包含若干个条目，这个请求就可

能包含**过滤限定符**。在某个条目成为响应的一部分之前，该限定符规定条目所必须满足的一个或多个条件。一个例子就是X.400消息系统中对于特定城镇中给定类型的所有公司名称的请求。相似地，一个用户可以使用LIST和SEARCH服务的组合浏览目录中的内容。

表14-2　目录服务：请求类型

目录查询请求	
READ	请求某个对象条目的一些或所有属性值；通过DUA，用户提供需要的属性类型
COMPARE	请求目录将给定值（例如口令）与给定条目中的给定属性进行比较
LIST	查询给定条目的直接下一级列表
SEARCH	在DIT的某个范围中请求满足给定过滤条件的所有属性值
ABANDON	通知目录，用户不再关注前一个请求的结果
目录修改请求	
ADD_ENTRY	把一个给定的对象名称或别名以及相应的属性添加到DIT中的特定位置，使之成为一个实际的条目
REMOVE_ENTRY	从DIT中删除一个给定的对象名称或别名的条目，该条目必须是叶子条目
MODIFY_ENTRY	对给定条目进行修改的顺序。或者所有的修改都成功或者所有的修改都不成功。其中包括：对一个给定属性或属性值进行添加、删除或替换
MODIFY_RDN	对一个对象或别名叶子条目的RDN进行修改

READ服务执行基本的地址解析（查找）服务。它使DUA提供对象的DN和所需结果的属性类型。本地DSA返回对应这个属性类型的所有值。在OSI因特网应用中，DN指的是一个服务和属性类型的PSAP地址。相似的，在OSI消息系统中，DN是发起者/接收者的名称以及在
824 整个消息系统中发起者/接收者地址的属性类型。另一方面，可以用别名替代DN。而且，在一个请求中可以请求多个属性类型的值。

目录总是报告每个请求的结果。显然，由于违背某个条目的安全机制或者所提供的参数出现问题（例如无效的名称或属性值），任何的请求都有可能失败。于是，返回一个差错消息并附有错误的类型。

在要求用户认证的应用中，访问前授予目录中的条目**认证属性**（例如口令），可包括在条目中。如果提供的口令是正确的，则目录返回一个肯定的响应，否则会返回一个错误消息。例如，对于网络管理员工作站是必要的。在被允许添加/删除/修改目录的条目之前，管理员必须通过适当的交互AP提供正确的口令。同样，对于连网资源（例如文件服务器，包含机密信息），那些被允许访问服务器的用户必须提供包括口令在内的一组属性。而其他用户只能得到一个错误消息。

有一些**简单认证**的例子。在很多请求式应用中，认证属性可能是使用用户公共密钥加密的用户口令。在这种情况，只有用户提供了使用他的私有密钥加密的口令，才能得到认证的口令。这是一种**强认证**的例子，它需要DSA进程执行附加的安全处理。

3. 目录结构

正如已经讨论的，除了最小的系统，目录必须以分布式的方式构造。物理分布通常会反映DIT的逻辑分布。整个DIB根据DIT的结构被分成若干个分区。每个分区有一个DSA提供访问，一般方案如图14-9(a)所示。

DSA从本地的DUA接收到一个请求，它首先为所请求的信息查找它的DIT分区。如果信息在当前DIT中，它直接返回给DUA。否则，它必须把请求转发给其他更高层次的DSA（即

引用），这个规程与域名系统中使用的方法是一样的。然而除引用之外，DSA提供了两个查找规程：链接和多播。

对于**链接**，如果DSA接收到一个请求，而对于这个请求，它只是请求信息的一部分，则产生一个请求消息，请求其他部分，并发送给它认为具有这些信息的DSA。这种请求类型称为**链式请求**，这是因为如果接收者DSA没有所有的请求信息，那么它也会产生一个新的请求等等。链式请求的响应必须总是沿着匹配请求所使用的路径返回。接收者DSA把接收到的信息与自己的信息组合到一起，并把它返回给链路中的下一个DSA，如果已经到了链路的结尾， 825
则把信息返回给DUA。方案如图14-9(b)所示。

多播与泛洪是相似的。如果DSA的DIB分区没有请求的必要信息，它会把这个请求的副本转发给它知道的所有其他DSA。如果其中一个拥有所请求的信息，它把这个信息返回给请求DSA，并由它再返回给用户。如图14-9(c)所示。

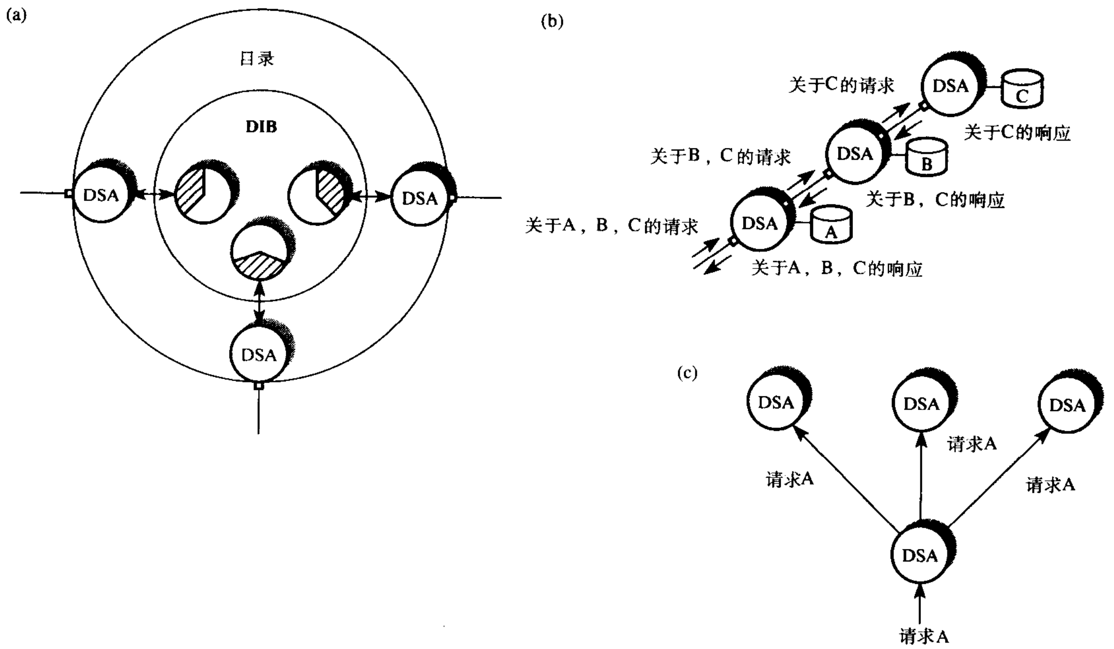


图14-9 目录操作

(a) 结构 (b) 链接 (c) 多播

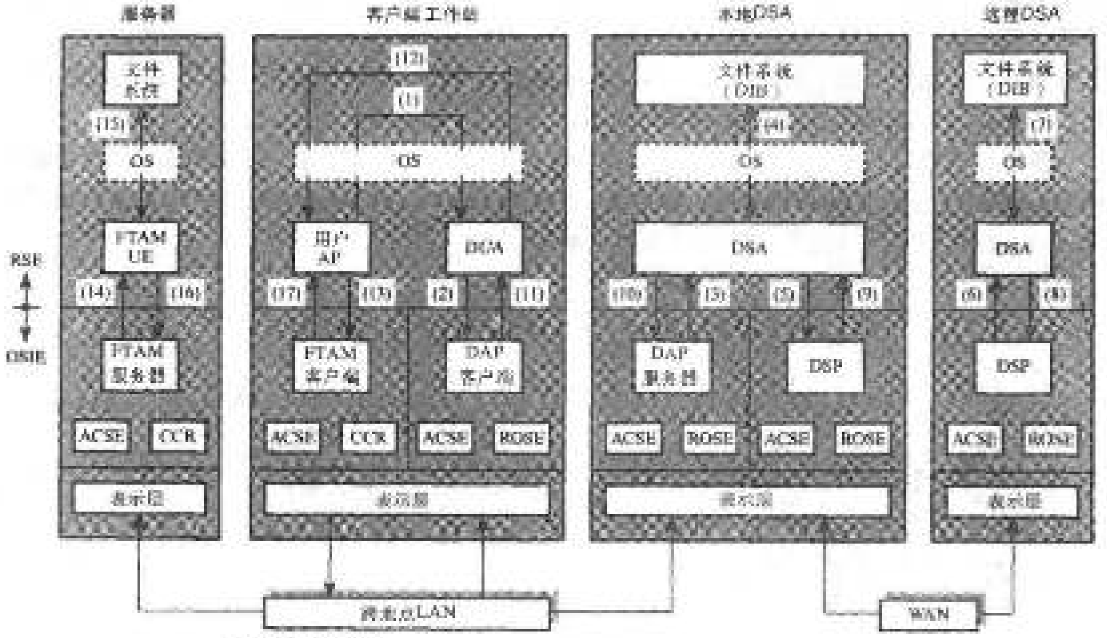
同域名系统一样，DSA维护系统中所有其他DSA的网络地址高速缓存，利用这个高速缓存使得引用的次数最小。因此，接收到一个会产生引用的请求，它通常直接把请求发送给相应的DSA，或者通知DUA用于回答这个请求的DSA的地址。然后DUA向这个DSA直接发送一个新的请求。

4. 目录协议

记住，DUA和DSA都是AP而不是协议。根据ISO参考模型，它们都是实系统环境，而不是OSIE。DUA和DSA之间的所有通信都使用标准的抽象语法定义的请求—响应（远程操作）交互消息。类似地，DSA之间的所有消息交换也都是抽象语法形式，同样是请求—响应类型。两种交换中的相关信息都是使用ASN.1定义的。除了这些消息的定义，支持协议必须确保这 826

些消息是以开放的方式交换的，就是说，所交换的消息在两个通信系统中具有相同的含义。

为了实现这个目的，定义了两种支持协议，每个都使用了ACSE和ROSE应用支持服务元素以及表示层提供的服务。有关DUA-to-DSA消息交换的协议称为目录访问协议（DAP），而有关DSA-to-DSA消息交换的协议称为目录系统协议（DSP）。实例如图14-10所示。



DAP = 目录访问协议　　DUA = 目录用户代理进程
DSP = 目录系统协议　　DSA = 目录服务代理进程
UE = 用户元素　　AP = 应用进程

（1）用户AP向DUA AP发送FTAM服务器名称
（2）DUA向DAP客户端发送请求消息
（3）DAP服务器向本地DSA传递请求
（4）本地DSA检查本地的DIB
（5）本地DSA向DSP发送引用请求
（6）远端DSP向它的DSA传递请求
（7）远程DSA从它的DIB中得到地址
（8）远程DSA向它的DSP返回响应
（9）DSP向本地DSA传递响应
（10）DSA转发响应给DAP
（11）DAP客户端向它的DUA传递响应
（12）响应送到用户AP
（13）用户AP发出FTAM请求
（14）FTAM服务器向UE发出请求
（15）执行本地文件访问
（16）数据返回给FTAM服务器
（17）数据传递给用户AP

图14-10　X.500目录协议以及消息/APDU交换的实例

实例中假设运行在客户端工作站的用户AP想要从连到同一个地区LAN上的文件服务器上
827 获得一个文件。在用户AP开始向本地FTAM应用实体（假设它是由FTAM客户端ASE和ACSE以及CCR服务元素组成）发送（远程）文件请求之前，必须首先获得服务器的完全限定PSAP地址，而服务器只能通过它的应用标题（名称）知道。图中的步骤1～12指出了获得地址的消息交换顺序。

如图所指，DAP和DSP使用了远程操作（远程规程调用）进行通信，因而这些是由DAP/ROSE提供的。当使用ACSE提供的服务在DUA和本地DSA之间建立联系之后，本地DSA

使用DAP/ROSE提供的服务调用相应远程操作/规程，实现DUA的所有请求。服务器名和PSAP地址属性类型在RO_INVOKE原语中作为参数传递，而结果作为RO_RESULT原语的参数返回。

在所示的序列中，假设站点本地DSA的DIB中没有所请求的地址，但实际上的可能性不大，它必须向一个远程DSA转发请求。步骤5~9指明为了从远程DSA获得地址所发生的消息交换。当本地DSA得到所请求的属性（PSAP地址）后，它把这个地址转发给请求DUA，并通过DUA传给用户AP（步骤10~12）。

然后，远程AP使用本地FTAM应用实体提供的服务执行对远程文件服务器/系统的文件访问请求。消息交换序列见步骤13~17。实际上，交换的消息比示例更多，因为联系必须首先使用ACSE服务在客户端和服务器AP之间建立，在传递任何文件数据之前，有关文件选择和打开规程的消息必须交换。图中的序列只用于X.500的操作，而不是FTAM。

14.2　OSI环境实例

第9章提到因特网是目前为止所存在的最大范围的OSIE。它不仅仅为成千上万用户提供了互联网范围应用的通信支持，还是很多关于网际互连以及相关网络协议问题的研究测试基地。

然而，现在只有少数几个网络是基于OSI协议族建立的。这主要是因为到目前为止还缺乏更高的面向应用层的稳定的标准。然而，现在已经有了一组可行的完整标准。因此很多新的开放系统环境都是以OSI协议族为基础定义的。

在公共部分，许多OSI环境是通过PTT和公共载波建立的，其中包括X.400公共消息网络，以及智能用户电报、可视图文和传真网络。接入这些网络的所有设备都使用完整的OSI协议族。828
通常，每个层中使用的协议都遵循ITU-T X系列标准，但是，等价的ISO标准具有更好的兼容性。如14.1.2节描述的，X.500目录也是一个完整的国际标准，它将会促进OSI协议的使用。这种网络所提供的服务被PTT称为**电信服务**。在第8章讨论WAN时，首次引入了它。

在专用部分，两个OSIE的例子就是MAP和TOP。在制造业，由美国通用汽车公司发起提出了一组协议，这些协议都是基于ISO标准的，用于实现自动制造业中的开放系统互连。这个协议组称为**制造业自动化协议**（MAP）。图14-11显示了一个MAP网络以及它所选择使用的协议。

正如所见，MAP协议是基于工厂范围的，沿着主干电缆分布于网络。使用的是同轴电缆并以10Mbps的速率操作。因为通信要求是应用在一个工厂的范围内，所以采用了宽带工作模式。由于ISO 8802.4令牌总线标准具有确定的访问时间，所以MAP协议在MAC子层采用了ISO 8802.4。网络层和LLC子层使用无连接模式进行操作，而传输层则采用称为TP 4的级别4面向连接协议。分布信息服务包括TFAM和MMS，它们都使用ACSE应用支持服务元素执行操作。

为了在一个制造部门内通信（就是为了在部门控制器与一组分布的机器人和其他计算机控制的机器之间通信），要采用一种尽可能简单以及成本较小的载波频带传输方式。因为部门内的大部分通信都是本地的，可以使用一组简化的协议。完整七层模型的时间开销是非常大的，在某些情况，对于部门控制器和自动设备之间的通信是无法接受的。简化后的这组协议包括直接连到LLC子层的MMS（包含基本的文件服务），而它是面向连接的。使用这种方法，部门内部通信的时间开销极大地减少了。现在这个系统称为**增强性能结构**（EPA）。通常一个部门控制器，支持完整的七层MAP结构，也支持EPA，因此，除执行本地的管理或控制功能外，部门控制器能够通过主干网络在工厂范围内通信。

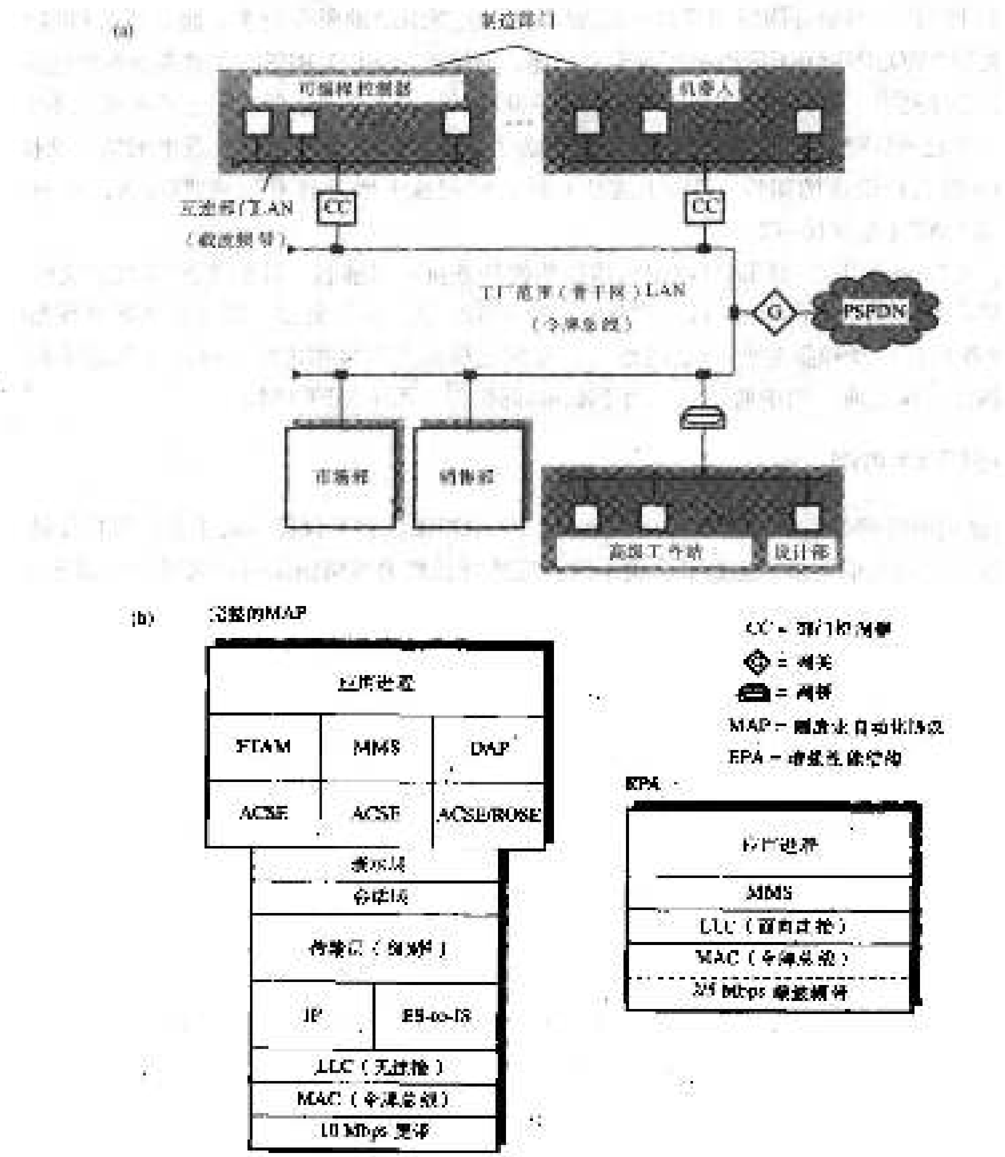


图14-11　MAP网络
(a) 示意图　(b) 协议

同样，由美国波音公司提出，并经ISO标准化，用于在一个技术和办公环境中实现开放系统互连的协议组称为技术办公协议（TOP）。图14-12表示了一个部门范围的TOP网络以及它选择使用的协议。

829/830 TOP网络使用的传输介质也是10Mbps同轴电缆。通常，这种环境下的通信需求仅限于声音（通常由现有的电话系统提供）和数据传输，后者主要关注高级工作站中的分布式群体之间的通信（例如，执行计算机辅助设计）。电缆以基带模式操作，并且MAC协议采用ISO 8802.3（CSMA/CD）。对于MAP，网络层和LLC子层采用无连接的模式，而传输层采用级别4的传输协议。所选择的分布信息服务包括FTAM、MHS、JTM和VT。

这仅是当前已建立起OSIE的实例。其他应用领域的管理权威机构也正在向开放系统方向发展，会有更多的OSIE出现。

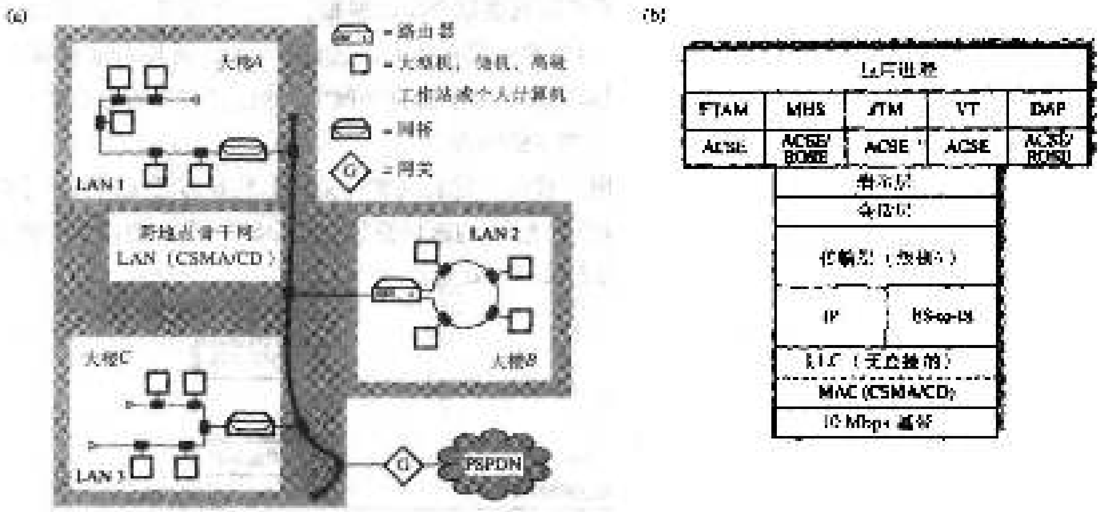


图14-12 TOP网络
(a) 示意图 (b) 协议

14.3 层间交互

为了进一步理解完整通信协议族中的操作，必须了解组成每个协议族的各个协议（协议实体）以及它们之间如何交互以实现特定的网络操作。将考虑由多个客户端工作站组成的应用领域，这些客户端要与网络文件服务器进行通信。假设网络是单一的LAN网络。首先考察TCP/IP协议族中的消息（即PDU）交换情况，接下来再考虑OSI 协议族中的情况。

14.3.1 TCP/IP

图14-13（a~f）显示了LAN总线上的一个通常的帧传递序列，传输是在一个（客户端） 831
用户AP和文件服务器之间进行的，图中还显示了每次传递发生的层间互操作。为了使说明更清楚，假设客户端FTP已经从本地名称解析器获得了服务器的IP地址，并且所有的交换消息都是正确形成、没有传输差错发生。接下来只涉及打开准备读或写的（远程）文件。

首先，服务器将要向本地TCP发送一个PASSIVE_OPEN，通知TCP它已经能起作用并准备接收新的文件传递请求。接下来，客户端用户AP（或者终端上的用户）向本地的客户端FTP发送一个f_open命令来启动一个文件操作，命令中要包含必要的附加信息，例如服务
器的名称和文件的名称。客户端FTP首先得到服务器的IP地址（没有表示出），然后使用服 832
务器的IP地址和FTP服务的知名端口号21，在FTP客户端和服务器之间发起传输连接（TC）建立。

接下来FTP客户端会发送一个ACTIVE_OPEN请求，层间互操作如图14-13(a)所示，图中圆括号里的数字表示交互操作发生的顺序。本地TCP首先会向FTP客户端返回一个OPEN_ID响应，使客户端可以把接下来的消息与这个端口号关联，因此连接建立（TC）。然后本地TCP

会产生和发送一个带有服务器知名端口号的SYN段（见第11章）。TCP把SYN段和服务器IP地
833 址传递给本地IP。本地IP产生一个IP数据报，它的头部包含服务器IP地址，而SYN段放在IP用户数据中。使用LLC和与LAN相连的MAC子层发送这个IP数据报。每个子层都会在IP数据报的头部加上自己的协议控制信息（PCI），最后在电缆介质上发送实际的帧，在图的底部显示。接收到这个帧，MAC、LLC和IP层会分别解释帧头部中各自的PCI，然后把剩余的数据部分传递给上一层。所以当消息到达TCP层后，只留下SYN段。

接收到SYN，TCP如图14-13(b)中响应。首先，它会通知本地FTP服务器已接收到一个新
834 的连接请求，然后通过返回自己的SYN段建立此TC的返回路径。这个SYN段按前面的相同方式传递。而客户方的TCP按图14-13(c)中的方法进行处理。

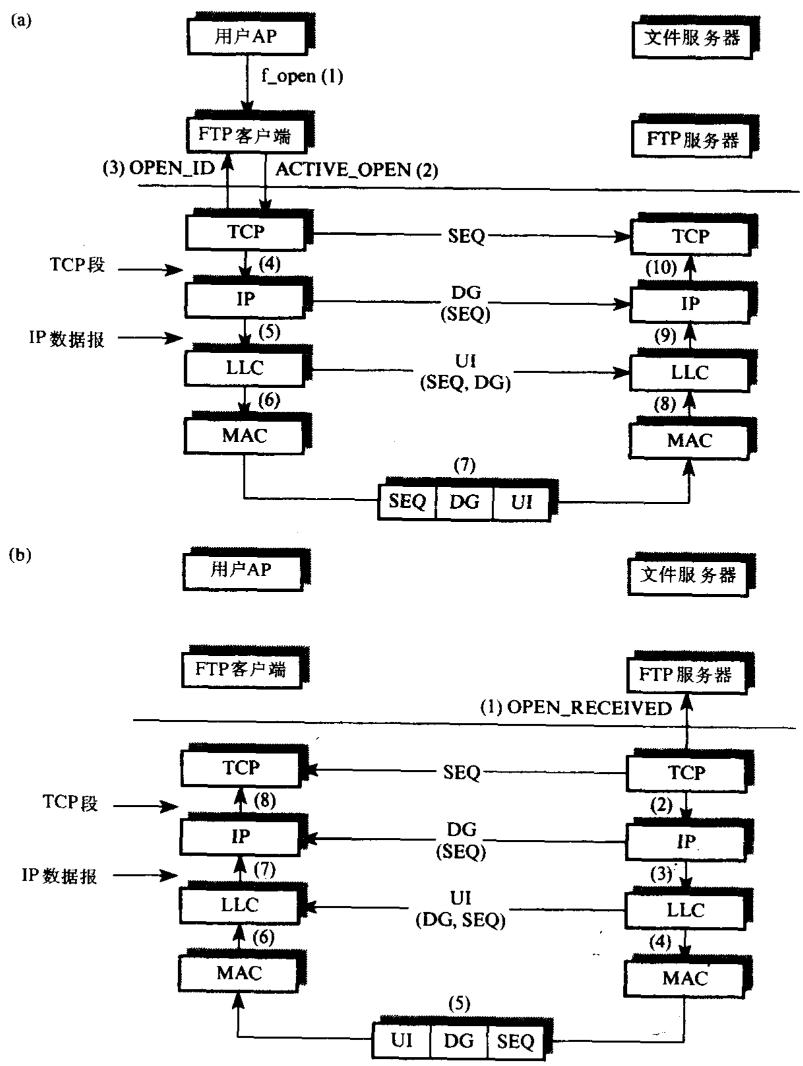


图14-13　实现远程文件打开操作的TCP/IP层间交互

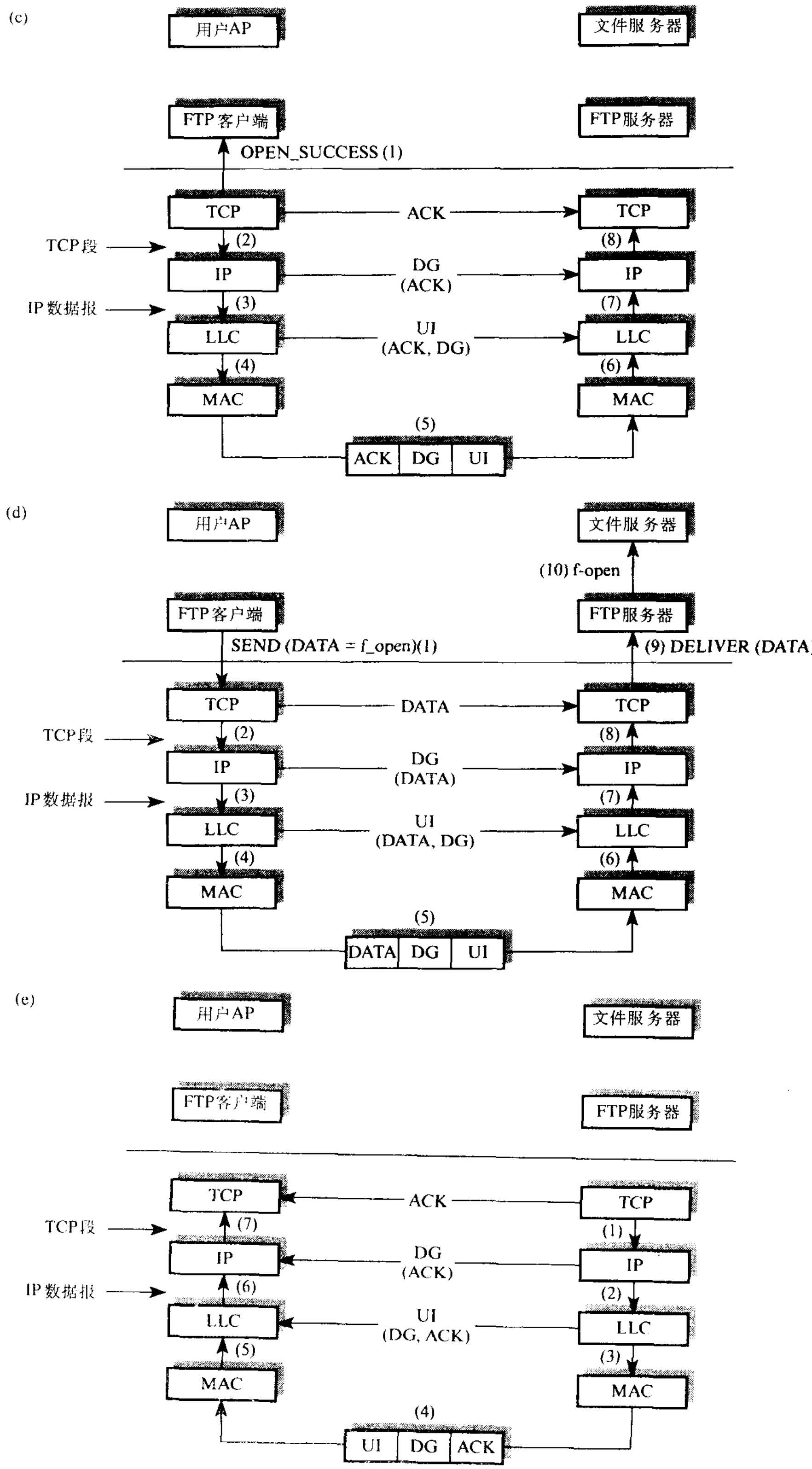

(c)
用户AP
文件服务器
FTP客户端
FTP服务器
OPEN_SUCCESS (1)
TCP
ACK
TCP
TCP段
(2)
(8)
IP
DG
(ACK)
IP
IP数据报
(3)
(7)
LLC
UI
(ACK, DG)
LLC
(4)
(6)
MAC
MAC
(5)
ACK
DG
UI
(d)
用户AP
文件服务器
(10) f-open
FTP客户端
FTP服务器
SEND (DATA = f_open)(1)
(9) DELIVER (DATA)
TCP
DATA
TCP
TCP段
(2)
(8)
IP
DG
(DATA)
IP
IP数据报
(3)
(7)
LLC
UI
(DATA, DG)
LLC
(4)
(6)
MAC
MAC
(5)
DATA
DG
UI
(e)
用户AP
文件服务器
FTP客户端
FTP服务器
TCP
ACK
TCP
TCP段
(7)
(1)
IP
DG
(ACK)
IP
IP数据报
(6)
(2)
LLC
UI
(DG, ACK)
LLC
(5)
(3)
MAC
MAC
(4)
UI
DG
ACK

图14-13 （续）

(f)

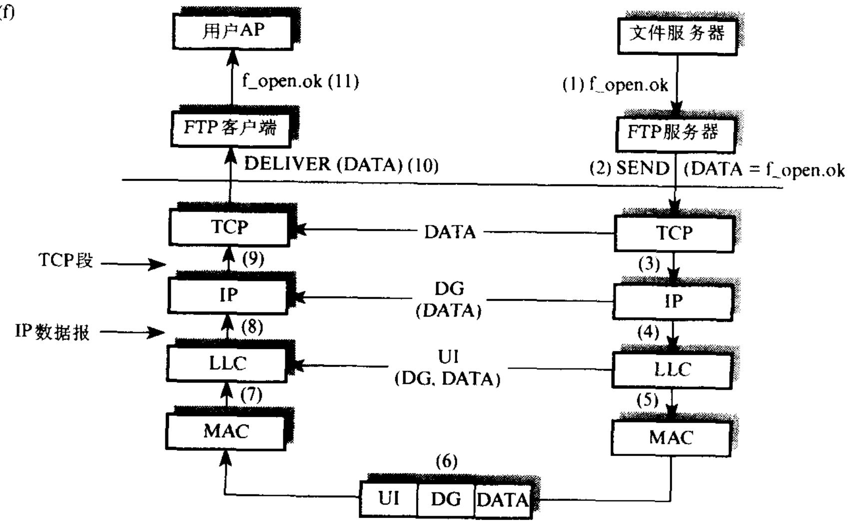


图14-13 （续）

TCP首先向客户端FTP发送一个OPEN_SUCCESS，表示已经建立了一个TC。向服务器TCP返回ACK段，完成连接建立规程。同时，FTP客户端如图14-13(d)所示作出响应。首先，在已建立的TC上，把一个f_open请求消息作为用户数据发送给FTP服务器。它是放在DATA段中传递给服务器方的。一旦接收到，TCP使用DELIVER原语把f_open请求传递给服务器。然后，服务器以正常方式向本地文件系统发送f_open请求，就好像它是一个本地请求。

因为f_open是在DATA段中传递的，服务器方的TCP返回一个ACK，顺序如图14-13(e)所示。图14-13(f)显示了文件系统对于本地FTP服务器打开文件请求肯定的响应；这个响应再次放在DATA段中返回。FTP客户端把打开确认转发给用户AP，用户AP在已经打开的文件上，处理后面的读或写操作。

为了描述目的，所示的序列都已简化。然而，它们同样阐述了在完整的协议族环境中，每个层是如何执行功能的。

14.3.2 OSI

在描述OSI协议族的等价序列之前，必须定义协议族使用的**应用环境**。假定如下：

- 以TOP为基础构成OSIE。
- 应用实体仅仅由FTAM和ACSE应用服务元素组成。
- 与网络有关的协议层（网络层、LLC和MAC）都采用无连接的模式。
- 传输层提供面向连接的级别4的服务。

在客户AP（发起方）与服务器AP（响应方）之间，通过LAN电缆介质，传输的帧序列，以及每次传输引发的层间交互操作，如图14-14所示。为了使说明更清楚，假设所有的服务原语和相关的PDU都是结构正确的，并且不会发生传输差错。

为了开始一个远程文件操作，客户端用户AP（通过与它相连的用户元素）首先发送一个F_INITIALIZE.request原语。随后的层间交互操作如图14-14(a)所示，图中圆括号里的数字表示操作发生的顺序。正如所见，接收到请求原语（1），发起方FTAM实体利用服务请求中的

参数产生一个INIRQ PDU，并将PDU的ASN.1编码版本写入用户数据缓存（UDB），而缓存只是一些字节数组，UDB中的内容最终会通过电缆介质传递。 835

UDB的地址指针放在A_ASSOCIATE.request原语的用户数据参数中同附加参数一同传递给ACSE（2）。接下来，ACSE实体产生自身的PCI，并把PCI添加在UDB中已有INIRQ PDU的尾部。由这两部分形成AARQ PDU。将UDB的地址指针放在P_CONNECT.request原语的用户数据参数中同其他附加参数一同传递给表示层（3）。

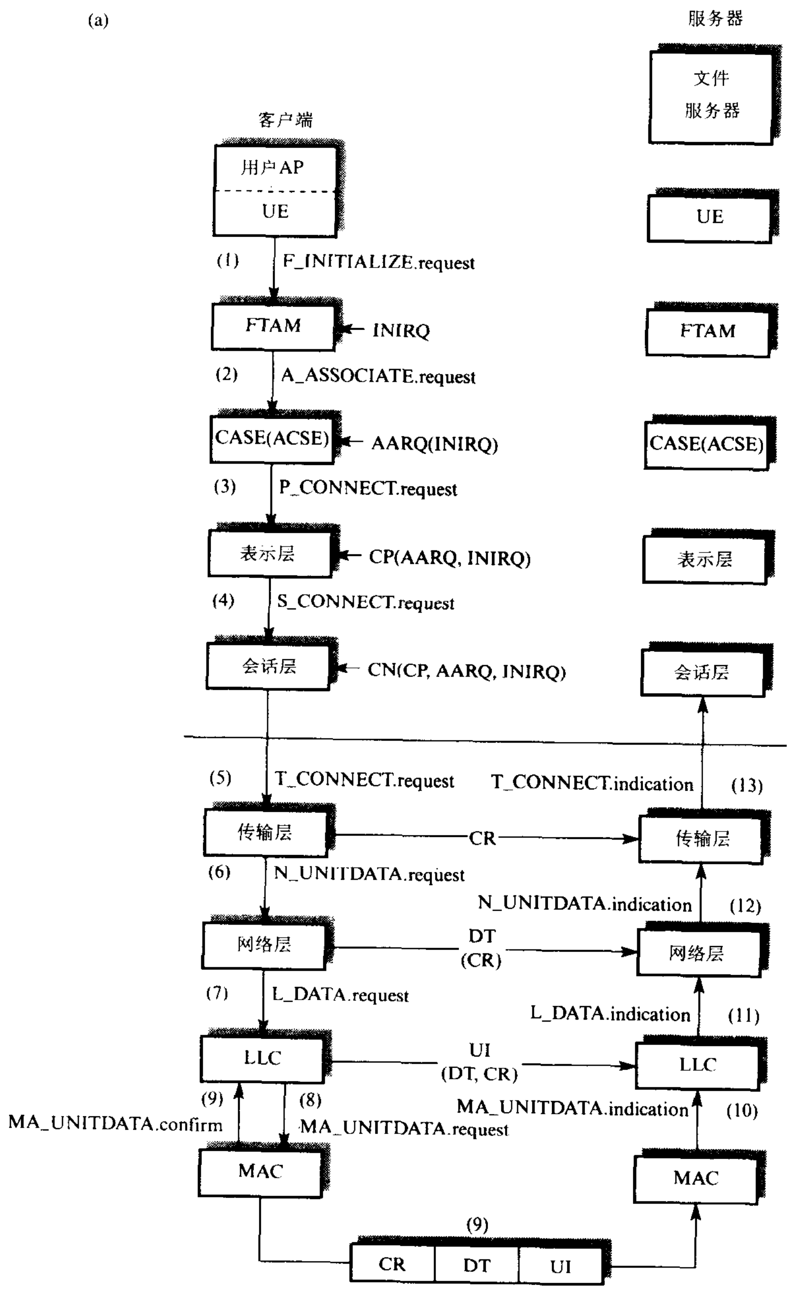


图14-14 OSI 层交互

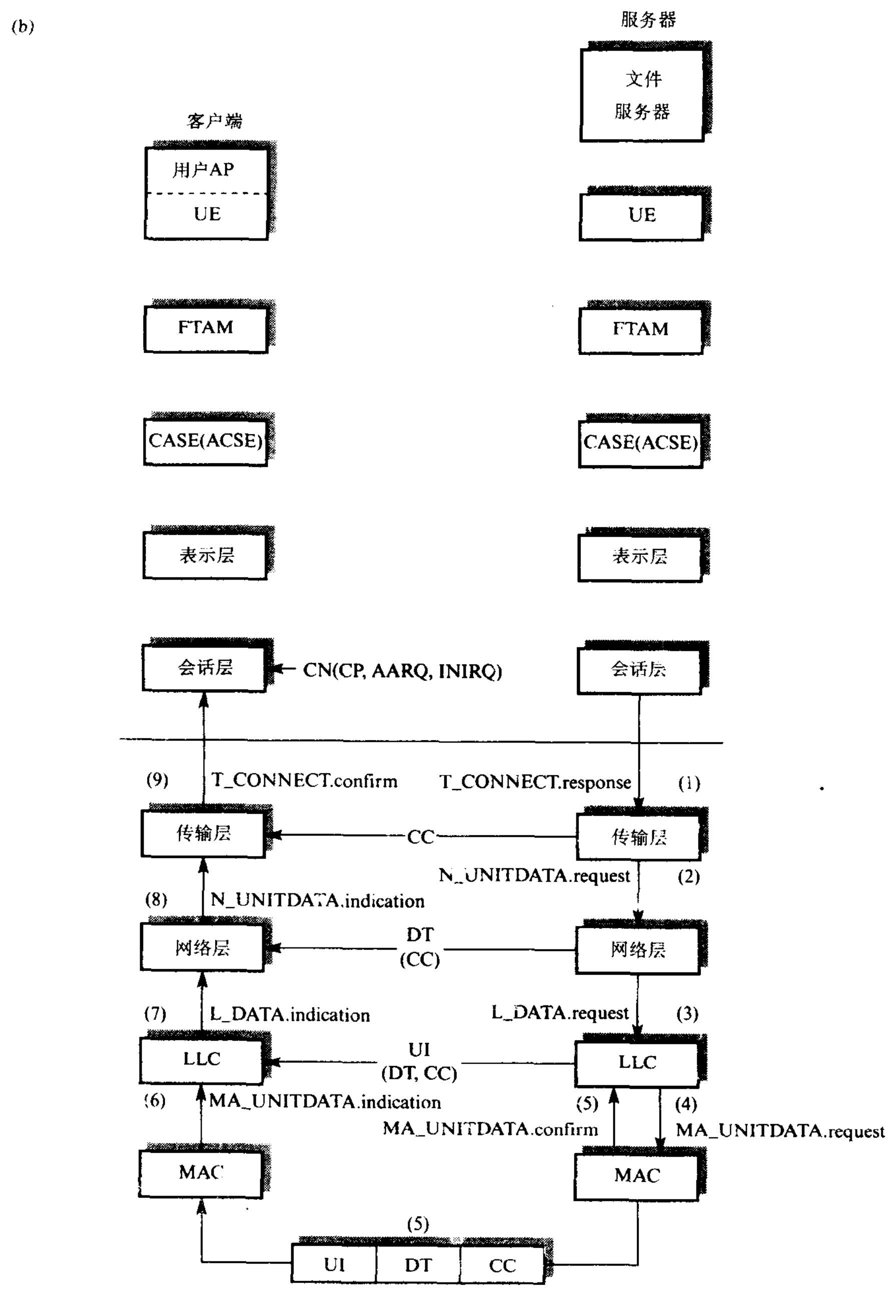

(b)
服务器
文件服务器
客户端
用户AP
UE
UE
FTAM
FTAM
CASE(ACSE)
CASE(ACSE)
表示层
表示层
会话层
CN(CP, AARQ, INIRQ)
会话层
(9)
T_CONNECT.confirm
T_CONNECT.response
(1)
传输层
CC
传输层
N_UNITDATA.request
(2)
(8)
N_UNITDATA.indication
网络层
DT
(CC)
网络层
(7)
L_DATA.indication
L_DATA.request
(3)
LLC
UI
(DT, CC)
LLC
(6)
MA_UNITDATA.indication
(5)
(4)
MA_UNITDATA.confirm
MA_UNITDATA.request
MAC
MAC
(5)
UI
DT
CC

图14-14　（续）

(c)

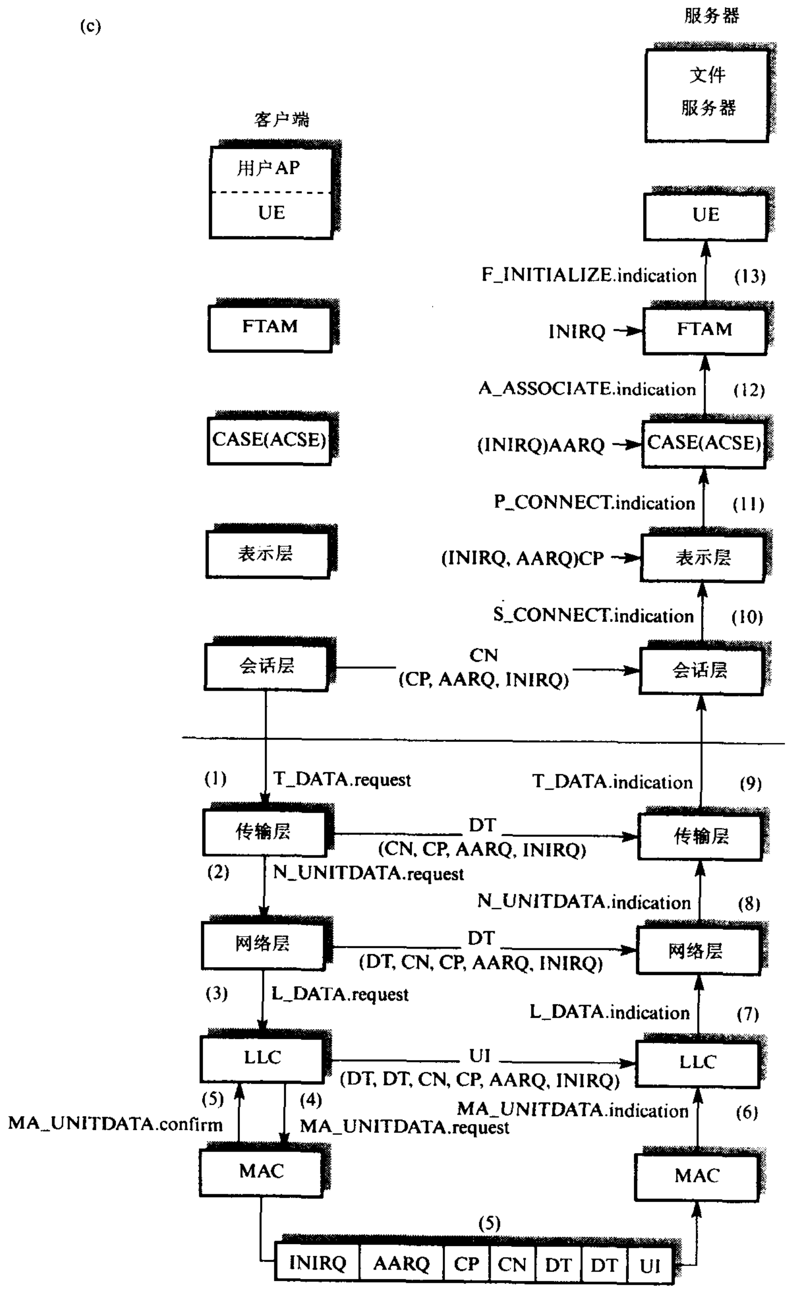

服务器
文件服务器
客户端
用户AP
UE
UE
F_INITIALIZE.indication
(13)
FTAM
INIRQ
FTAM
A_ASSOCIATE.indication
(12)
CASE(ACSE)
(INIRQ)AARQ
CASE(ACSE)
P_CONNECT.indication
(11)
表示层
(INIRQ, AARQ)CP
表示层
S_CONNECT.indication
(10)
会话层
CN
(CP, AARQ, INIRQ)
会话层
(1)
T_DATA.request
T_DATA.indication
(9)
传输层
DT
(CN, CP, AARQ, INIRQ)
传输层
(2)
N_UNITDATA.request
N_UNITDATA.indication
(8)
网络层
DT
(DT, CN, CP, AARQ, INIRQ)
网络层
(3)
L_DATA.request
L_DATA.indication
(7)
LLC
UI
(DT, DT, CN, CP, AARQ, INIRQ)
LLC
(5)
(4)
MA_UNITDATA.confirm
MA_UNITDATA.request
MA_UNITDATA.indication
(6)
MAC
MAC
(5)
INIRQ
AARQ
CP
CN
DT
DT
UI

图14-14 （续）

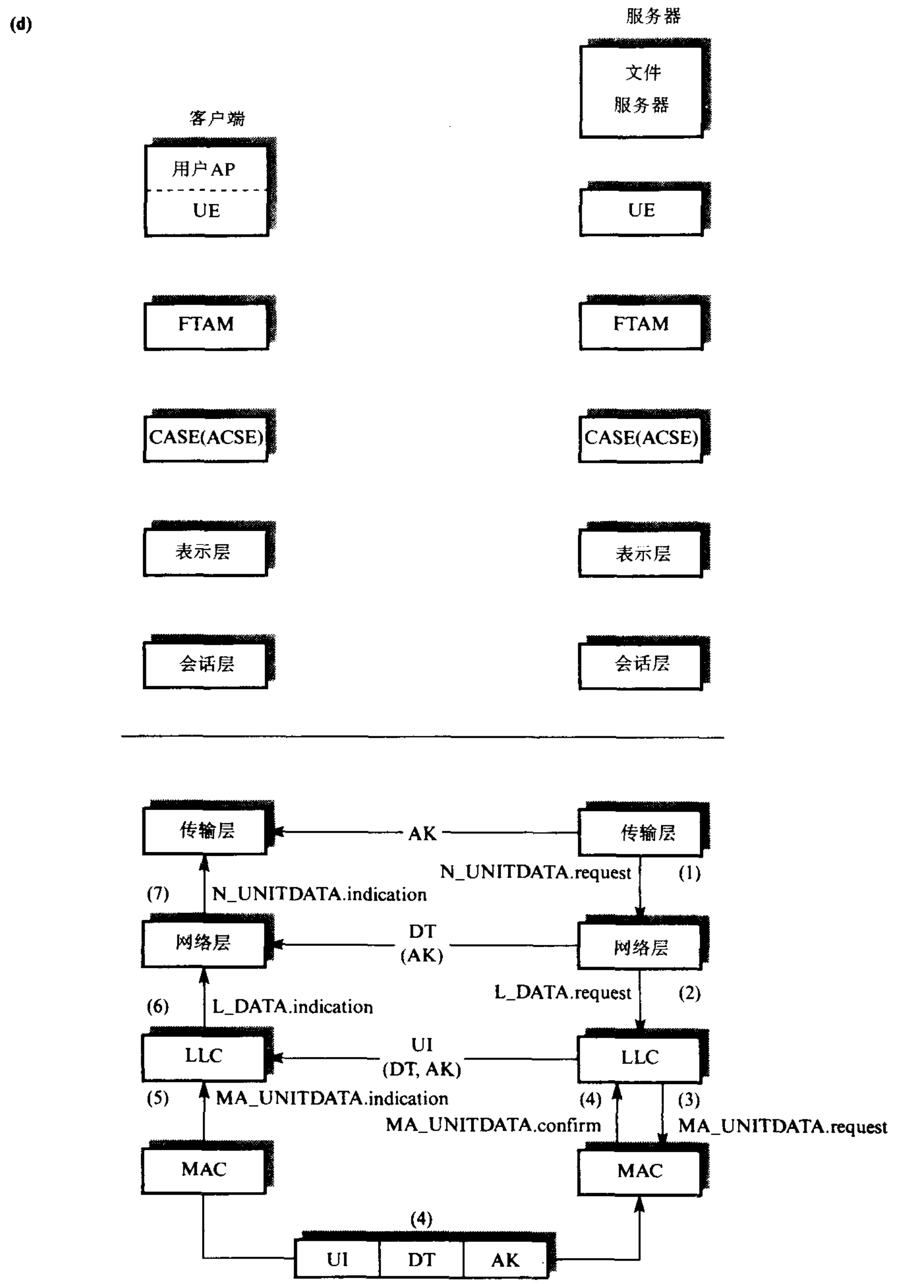

(d)
服务器
客户端
文件
服务器
用户AP
UE
UE
FTAM
FTAM
CASE(ACSE)
CASE(ACSE)
表示层
表示层
会话层
会话层
传输层
AK
传输层
N_UNITDATA.request
(1)
(7)
N_UNITDATA.indication
网络层
DT
(AK)
网络层
L_DATA.request
(2)
(6)
L_DATA.indication
LLC
UI
(DT, AK)
LLC
(5)
MA_UNITDATA.indication
(4)
(3)
MA_UNITDATA.confirm
MA_UNITDATA.request
MAC
MAC
(4)
UI
DT
AK

图14-14 （续）

(e)

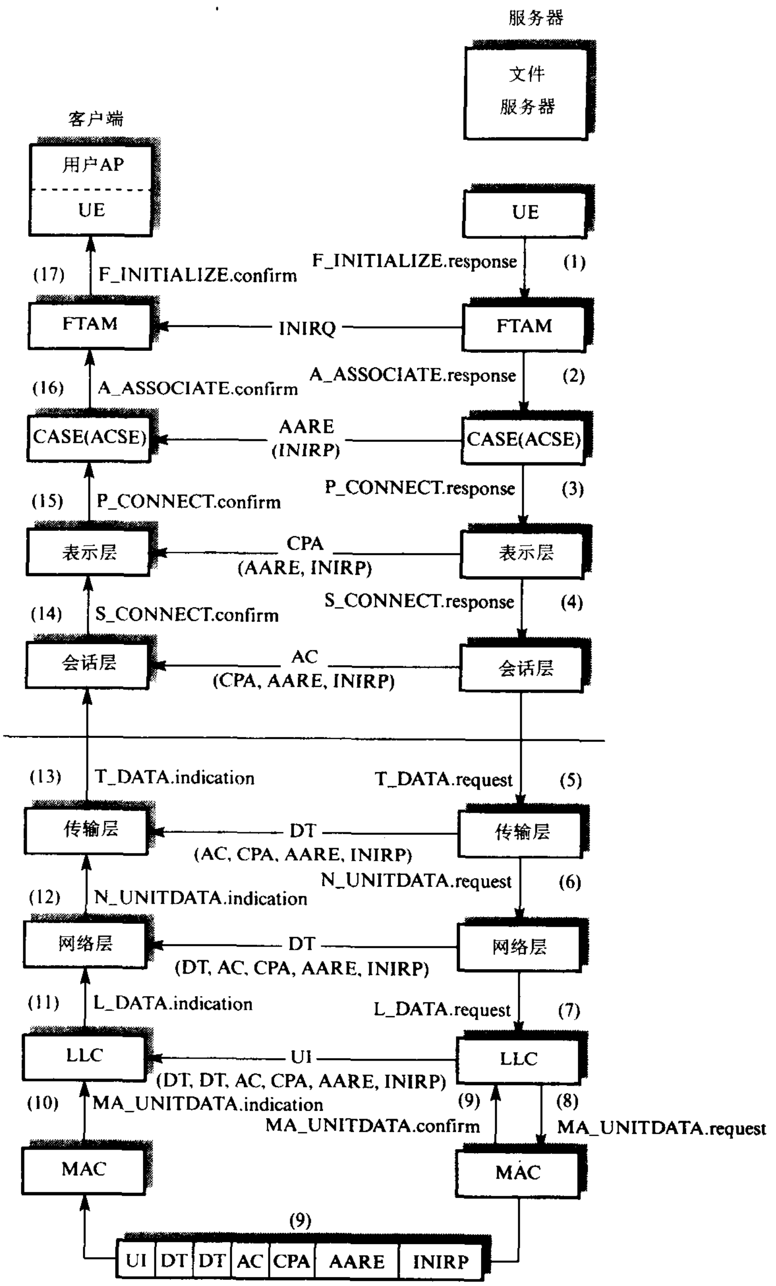

服务器
文件服务器
客户端
用户AP
UE
UE
F_INITIALIZE.response
(1)
(17)
F_INITIALIZE.confirm
FTAM
INIRQ
FTAM
A_ASSOCIATE.response
(2)
(16)
A_ASSOCIATE.confirm
CASE(ACSE)
AARE (INIRP)
CASE(ACSE)
P_CONNECT.response
(3)
(15)
P_CONNECT.confirm
表示层
CPA (AARE, INIRP)
表示层
S_CONNECT.response
(4)
(14)
S_CONNECT.confirm
会话层
AC (CPA, AARE, INIRP)
会话层
(13)
T_DATA.indication
T_DATA.request
(5)
传输层
DT (AC, CPA, AARE, INIRP)
传输层
N_UNITDATA.request
(6)
(12)
N_UNITDATA.indication
网络层
DT (DT, AC, CPA, AARE, INIRP)
网络层
(11)
L_DATA.indication
L_DATA.request
(7)
LLC
UI (DT, DT, AC, CPA, AARE, INIRP)
LLC
(10)
MA_UNITDATA.indication
(9)
MA_UNITDATA.confirm
(8)
MA_UNITDATA.request
MAC
MAC
(9)
UI DT DT AC CPA AARE INIRP

图14-14 （续）

(f)

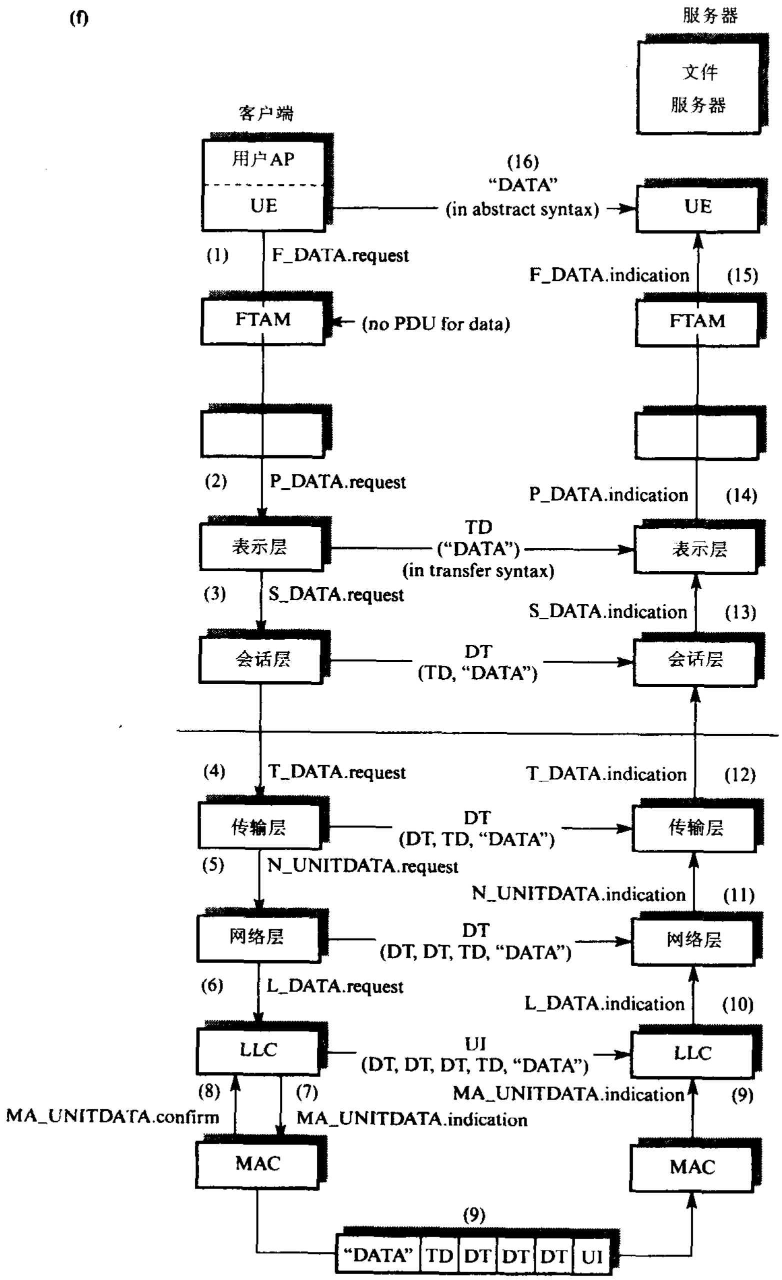


图14-14　（续）

表示实体遵循同样的处理规程：产生自身的PCI，把它加到UDB上（形成一个CP PDU），然后发送一个S_CONNECT.request原语（4）。同样，会话实体也产生自身的PCI，加到UDB上形成一个CN PDU。已经在第11章描述了与ACSE有关的各种表示层和会话层PDU。然而，假设此时TC与远程系统尚未建立。因此，在会话层实体向对等的会话层实体传递复合PDU之前，必须先建立一个TC连接。

会话层实体利用S_CONNECT.request原语参数中的地址信息（以及特定通信QOS），发送

一个T_CONNECT.request原语（5）。传输层实体接收到T_CONNECT原语后，利用与服务请求相关的参数产生一个CR PDU，并把它写入一个新的UDB。如已经提到的，假设网络层和链路层是无连接模式的。因此传输层实体发送一个N_UNITDATA.request原语，并将包含CR PDU的UDB地址指针作为参数（6）。

一收到该原语，网络实体利用其参数产生自身的PCI，并把它加到UDB中CR的尾部，从而形成DT PDU。然后将UDB的地址指针作为参数，发送L_DATA.request原语（7）。同样，LLC子层会把自身的PCI加入UDB，形成一个UI PDU，然后向MAC子层发送MA_UNITDATA.request原语（8）。最后，MAC子层把自身的PCI（包含目标和源LAN地址）加入UDB中。此时，UDB包含4个层的PCI，依次是传输层、网络层、LLC子层和MAC子层（在第6章已经描述过与网络层、LLC子层和MAC子层有关的PDU）。

现在，MAC子层将发起整个UDB内容的传输。它首先访问共享的电缆介质（根据CSMA/CD MAC方法），接下来产生FCS字段，并加在帧尾部，同已装配好的帧内容一起发送。然后，MAC子层向LLC子层发送一个MA_UNITDATA.confirm原语（9），通知它该帧已经成功发送。要注意，因为对于传递的每个帧，MAC PCI都是一样的（即相同的目标地址和源地址），为了清楚起见，图中没有给出。

接收到该帧后，服务器工作站上的MAC子层把它完整地存入UDB中，所以UDB的内容顺序与传送的内容顺序是相同的。虽然帧（即UDB内容）包含了若干个协议层的PCI，但因为每个层的PCI（以及PDU）都是精确定义的固定格式，所以每一层可以很容易地访问并译出自身PCI。因此，当MAC子层接收到完整的帧内容后，会首先译出MAC子层的PCI，使用MA_UNITDATA.indication原语（10）把UDB的地址指针，向上传递给LLC子层。 836~842

实际上，如后面将看到的，UDB中也包含了下一个要处理的PCI的开始地址的地址偏移。LLC子层接收到UDB指针后，它利用这个地址偏移确定自身PCI的开始地址。在确定接收到（UI）类型的PDU之后，LLC层要根据标准格式处理属于它的剩余PCI，并向网络层发送一个带有相应参数的L_DATA.indication原语（11）。接下来，网络层处理UDB中的自身PCI，并向传输层发送一个N_UNITDATA.indication原语（12）。最后，传输层实体处理自身PCI，并向会话层发送一个T_CONNECT.indication原语（13）。

假设我们准备建立一个TC，接收方会话层实体发送一个T_CONNECT.response原语作为响应（1）。产生的层间交互操作如图14-14(b)所示。传输层实体接收到T_CONNECT.response原语（1），首先产生一个CC PDU，把它写入一个新的UDB中并向网络层发送一个N_UNITDATA.request原语（2）。接下来遵循相同的规程，如图14-14(a)所示，最后传输层实体向发起的会话层发送一个T_CONNECT.confirm原语，证实TC已建立（9）。

现在，发起的会话层实体可以对正在等待的CN PDU（在它的用户数据字段中包括了CP、AARQ和INIRQ PCI）通过这个TC传输。它把包含正在等待的CN PDU的UDB的地址指针作为参数，向传输层发送一个T_DATA.request原语。接下来的层间交互操作如图14-14(c)所示。正如所见，除了与网络相关的各低层仅仅把自身的PCI加在UDB中CN PDU的后面，操作次序与前面的例子中是相同的。

服务器工作站接收到这个UDB后（5），每个协议层会处理自身的PCI，并把余下的内容传递给上一个协议层。最后，当FTAM实体（12）接收到这个UDB时，它会向响应方UE发送一个从INIRQ PDU构造的参数的F_INITIALIZE.indication原语（13）。另外，因为PCI与面向应

用的各层相关，是放在DT TPDU的用户数据中，由主叫传输层实体传递的，所以当被叫的传输层实体向会话层传递UDB（包含CN PDU）之后，它会为这个DT TPDU发送一个确认。操作序列如图14-14(d)中记录的结果交互。正如所见，此时除了TPDU是确认AK外，此过程的层间交互与图14-14(b)中是相同的。

服务器AP（通过它连接到UE）接收到F_INITIALIZE.indication原语后（1），它会向相连
843 的FTAM实体发送一个F_INITIALIZE.response原语作为响应。接下来的交互作用操作序列见图14-14(e)。正如所见，因为TC已建立，最终传递的UDB内容（9）包含所有协议层的相关PCI。因此，当接收到的UDB通过客户端系统各层向上传递时，每个层读取并解释与自身相关的PCI。最后，发起FTAM实体向客户UE发送一个F_INITIALIZE.confirm原语（17）。

正如第13章所示，FTAM的正常序列包括选择请求的文件，打开文件，随后才对该文件执行特定操作。上述每一步都将产生相似的后续操作序列。然而，图14-14(f)中表示的序列是客户AP向服务器发送数据时发生的层间交互。显然，这里有一个附加的F_WRITE.request原语。所要传递的数据可以是抽象的语法，但是在两个表示层实体之间传递的数据（TD PDU）必须采用协商的传送语法。而且，因为ACSE实体在数据传递阶段不起作用，所以ACSE实体用空格标出。

1. UDB 解码

为了增强对图14-14中所示的层间交互操作的理解，图14-15(a)显示了一个传输帧的内容。这个帧是根据图14-14(a)所示的操作定义的。帧的内容（即UDB的内容）是一个字节串，每个字节由两个十六进制数字表示。假设帧包含一个目标地址和一个源MAC地址。

为了阐述各个协议层是如何对接收到的帧进行解释的，图14-15(b)显示了帧的解码过程。假设使用的MAC地址是16位；而且MAC协议实体首先读取并解释UDB中的头四个字节（目标地址和源地址）。然后，UDB指针作为MA_UNITDATA.indication原语的用户数据参数传递给LLC子层，该原语所用的地址偏移为4。

收到UDB指针后，LLC协议实体从UDB中读取并解释自身的PCI。首先是SAP和SSAP字节，然后是控制字段字节。根据后者，可确定该帧是一个UI（无编号的信息）帧，说明该层PCI结束。然后，将地址偏移增加到7，并将UDB指针作为L_DATA.indication原语的用户数据参数传递给网络层。

网络层和传输层遵循相似的规程。首先，网络层协议实体根据网络层协议，从UDB中读取并解释相应数目的字节。然后，把经过适当地址偏移增量的UDB地址指针作为N_UNITDATA.indication原语的用户数据参数传递给传输层。同样，传输层协议实体读取并解释自身的PCI。现在，UDB已经读取完毕了，传输层协议实体向会话层发送一个T_CONNECT.indication原语，
844 并使用接收到的UDB中的某些字段构造必要的参数。

2. 地址参数

每个服务原语的相关参数都包含地址信息。例如，与F_INITIALIZE.request原语相关的主叫和被叫地址，是从本地目录服务中得到的完全限定地址。包括P/SSAP、TSAP和NSAP，而后者包括与层间接口相关的NSAP和LSAP扩展（或后缀），以及物理网络地址。由F_INITIALIZE.request原语引发的服务原语，从上向下穿越各个协议层，每个协议实体读取并把自己的SAP
845 嵌入到本层的PCI中。因此，在原语从上向下穿越各个协议层时，各层地址参数的大小是随之

减小的，到MAC子层，只剩下物理网络地址。同样，在服务器端，当服务原语从下向上穿越各个协议层时，每个协议层实体都将从PCI中读取SAP，并把它加到已有的地址上，从而重新构造地址参数，这一过程如图14-16(a)所示。

(a)

00	2E	36	39	FE	0E	03	81	21	01	09	1C	00	21	00	00	0B
49	00	02	02	00	00	00	00	2E	FE	00	0B	49	00	03	08	00
00	00	36	39	0E	00	08	E0	00	00	70	00	40	97	B7		

(b)

```
MAC PCI

00 2E = 目标地址
36 39 = 源地址

LLC PCI

FE = DSAP
0E = SSAP
03 = UI（无编号信息）

网络层PCI

81     = 网络协议标识符
21     = 头部长度
01     = 版本号
09     = PDU生命期
1C     = 0001 1100 = 不分段，DT PDU
00 21  = PDU段长
00 00  = 校验和（忽略）
0B     = 目标地址长度
49 00 02 02 00 00 00 00 2E FE 00  = 目标地址
                        49        = AFI (local)
                        00 02     = IDI
                        02 00     = SI
                        00 00 2E  = PA
                        FE        = LSAP }
                        00        = NSAP } SEL
0B     = 源地址长度
49 00 03 08 00 00 00 36 39 0E 00  = 源地址

传输层PCI

08     = 长度标识符
E0     = 1110 = CR TPDU, 0000 = CDT
00 00  = 目标引用
70 00  = 源引用
40     = 级别4服务
70 00  = 校验和
```

图14-15 UDB解码实例

(a) UDB（帧）内容 (b) 解码字段

为了建立应用联系和传输连接，指定了会话层和传输层连接标识符。后续的服务原语只包含作为参数的相关连接标识符。当实现各个协议实体时，除了保持与每个连接有关的协议状态信息之外，还要保存主叫和被叫地址的记录，如图14-16(b)所示。

一旦与某连接相关的每个TPDU传递到网络层时，网络层实体就访问相应的NSAP，它包含了物理网络地址及NSAP和LSAP扩展。利用这种方法，当完整PDU通过MAC层发送时全部地址信息便已插入。上例假设面向网络各层是以无连接模式工作的。然而，对于面向连接的模式，也须使用一种独立的网络连接标识符。 846

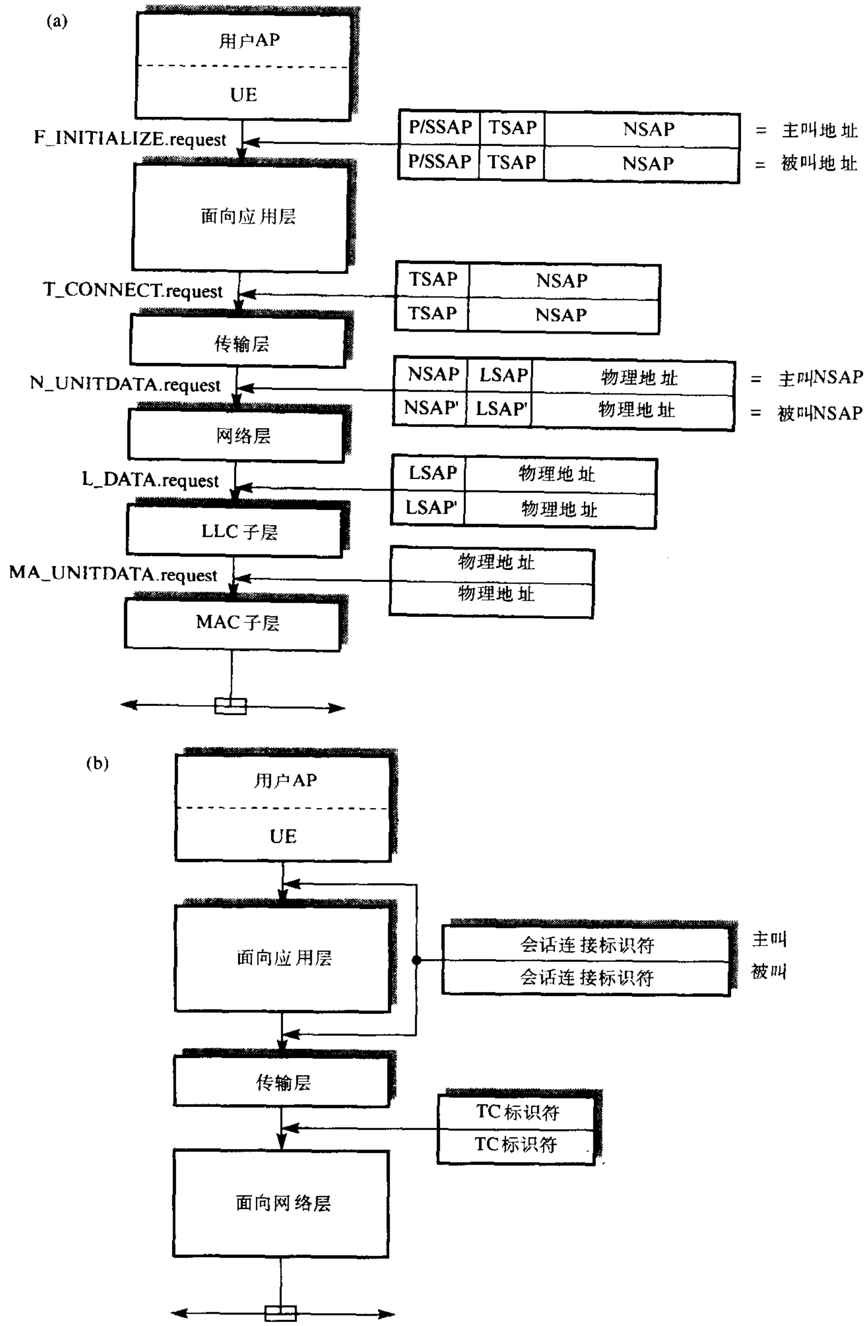


图14-16 地址参数
(a) 连接请求 (b) 数据传送

14.4 协议实现方法

任何计算机操作系统都有一组相关的原语（调用），每个原语也都定义了一组参数，通过这些参数允许一个用户程序/进程在高层次上以一种用户友好的方式访问计算机的各种外围设

备。这些原语包括与打印机交互的原语，与终端交互的原语等等。系统还提供了允许用户与本地文件系统交互的用于创建/删除/修改/读/写文件的原语。此处的重要问题就是应该提供给用户应用程序/进程何种原语才能和连网资源（如文件服务器）交互。

显然，用户必须从协议族的实现细节中解脱出来，并且对于相似的功能提供给用户的应该是一组相似的原语。为了实现这个目标，所有的通信软件都被分为两个部分：一个关注于协议族的实现，而另一个关注于用户接口的实现。

在TCP/IP协议族情况，客户方的用户接口通常是客户应用协议的一个整体部分，例如FTP客户端。它通常作为独立的AP实现，通过本地操作系统提供的原语与终端上的用户或用户AP进行通信。协议族中其他部分，例如TCP、IP等等，以一个独立的实体形式实现，可以被客户协议/进程按标准的方法通过操作系统提供的原语进行访问。

服务器方使用了相似的结构；服务器应用协议/进程使用操作系统提供的标准用户原语与本地文件系统通信。因此，不管服务请求来自本地AP还是来自远程进程，对于文件系统而言都是透明的。通常方案如图14-17(a)所示，图中的应用协议假定为FTP协议。 847

在OSI协议族情况下，应用协议（实体）和用户接口通常分成两个完全独立的实体。应用协议通常同协议栈的其余部分连接，通常方案如图14-17(b)所示。

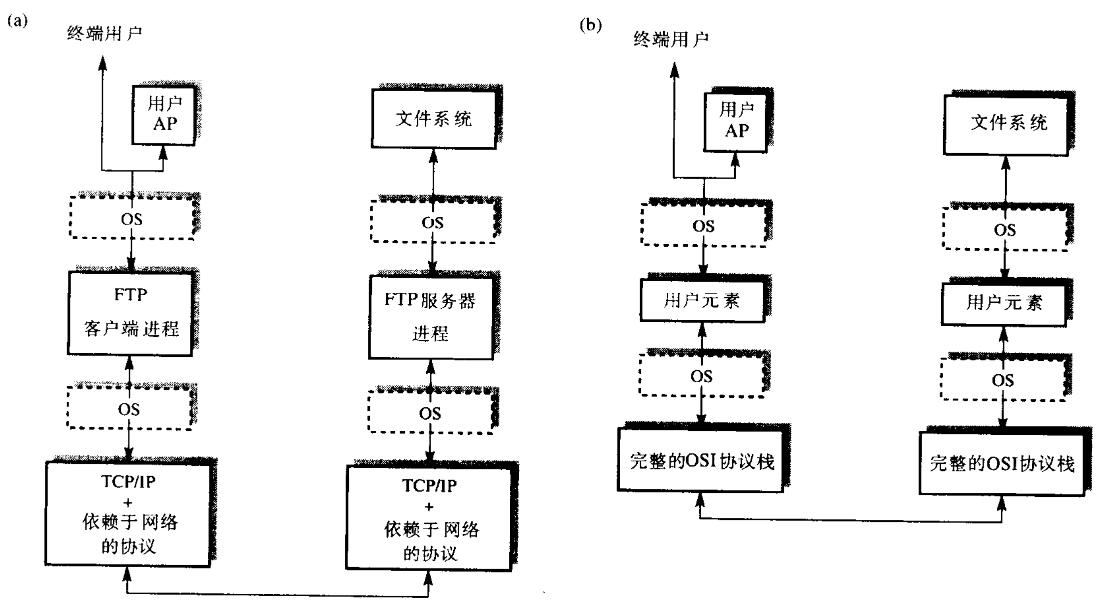


图14-17 协议结构

(a) TCP/IP协议族 (b) OSI协议族

通常，协议软件的接口是通过操作系统提供的一组高层次原语实现的。对于TCP/IP，它们与图11-4中所示的是类似的，而对于OSI，它们与所要访问的特定应用实体提供的原语是相同的。

协议软件可以通过两种方法实现。对于TCP/IP，通常它是作为本地操作系统的基本输入—输出系统（BIOS）实现的，因此它构成了操作系统的一个整体部分。而对于OSI协议族，从图14-4看到，对于整个协议栈的处理开销太高了。为了减小主机的负载，协议软件常常在一个独立的插入式的电路板上实现，电路板上包含了网络接口芯片，内存和本地处理器，这种方案如图14-18(a)所示。

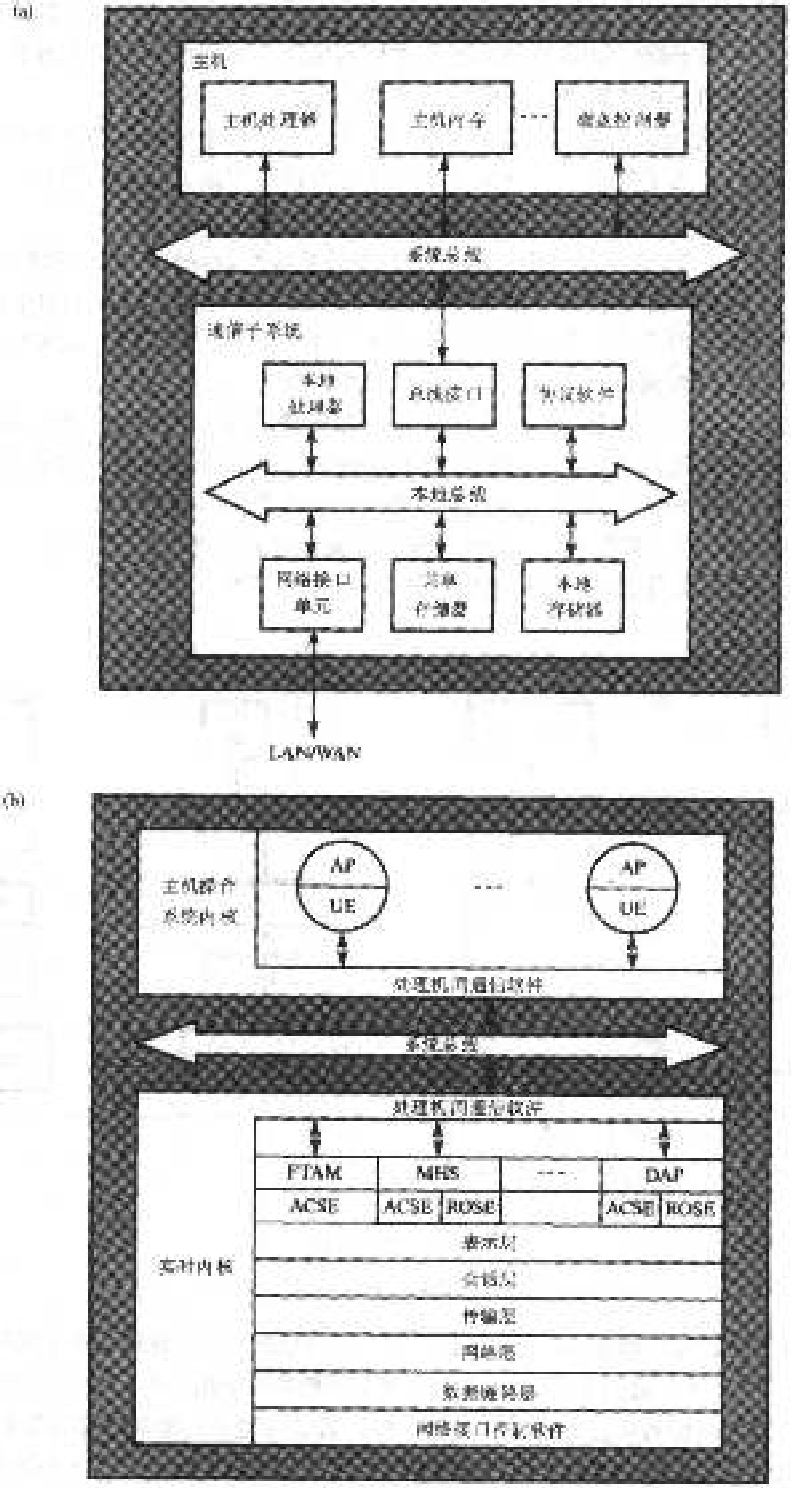


图14-18　总体结构
(a) 硬件　(b) 软件

通常，整个通信子系统由一块印刷电路板实现，插在主机系统的内部总线上，该总线用于实现互连机制。这块电路板除包含本地（通信）处理器外，还包括协议软件存储器，用来 848
存储每个协议层协议状态信息的附加存储器和两层之间传递消息的消息缓存（共享的），以及用于执行通信网络接口功能的电路。后者可以是公共WAN或LAN的物理接口，两者的不同仅在于所要求的接口电路类型不同，同第二部分描述的网络协议软件的差别不大。

通常，共享存储器是**多端口**存储器，它使网络接口单元、本地处理器和主处理器可以经系统总线直接访问层间消息和用户数据。当消息在缓存（和层）之间传递时，只需要使用缓存的地址指针，从而减少数据块在层间或设备间传送时额外的开销。

这种体系结构的相应软件结构如图14-18(b)所示。已经强调过，根据ISO参考模型建立的通信子系统的操作必须把每个协议层看作一个自治实体，它向上层提供了一组定义的用户服 849
务，并且使用下一层提供的服务，传递本层产生的PDU到远程系统上的对等层。这样，当软件实现各个协议层时，须遵循同样的方式，这样可以充分利用分层结构所带来的益处。

因此，通信子系统以一系列任务（进程）模块（每个协议实体一个模块）来实现，还有用于管理和定时功能的附加任务。任务通过一组FIFO队列或邮箱彼此进行通信，如图14-19所示。任务间的通信由本地实时内核管理，实时内核还担负任务调度和中断处理功能，例如，由定时器任务和网络接口单元产生的中断处理。子系统与主机之间的通信是通过处理器间通信软件进行的，在两个系统中都有该软件的拷贝。通常该软件是由中断驱动的，以确保两个系统之间消息的同步传输。因此无论何时，每当主机处理器向通信处理器传递信息时，首先把信息写入应用实体的一个空闲缓存中，然后产生一个中断给通信处理器。中断服务例程首先读取缓存的地址指针，确定所包含的SASE，然后把指针加入适当的输入队列的尾部，等待处理。在相反方向上的信息传递，遵循相似的规程。 850

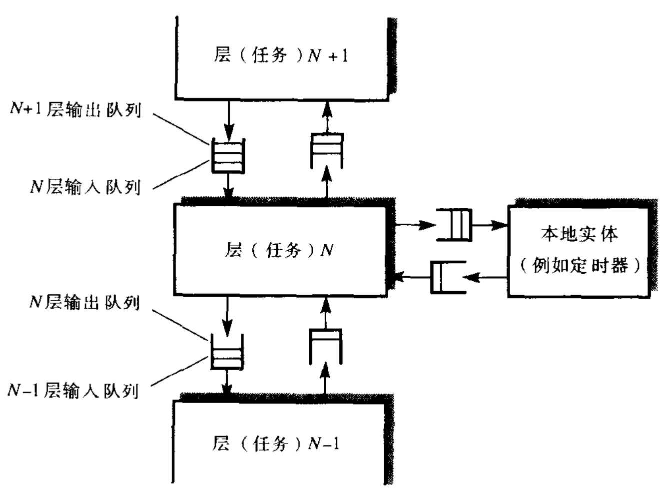


图14-19 层间通信示意图

14.4.1 层间通信

用户AP对于通信子系统支持的分布式信息服务的访问，是通过与其本地用户元素UE相关的一组原语实现的。UE或者是一个单独的进程或者是一系列库程序（函数），它首先与用户AP连接，然后运行。

如在11.6.6节描述的，层间原语和相关参数用称为**事件控制块**（ECB）的数据结构在层间传递。因为各层原语的参数类型和数量通常是不同的，所以对于每层都有一个独立的ECB数据结构，用它从高层到本层传递所有服务原语，包括已接收到的请求、指示、响应和证实原语，并通过本层发送这些原语到高层。关于传输层的ECB如图11-34所示。

为了与SASE进行通信，UE首先要获得一个空闲的ECB（实际上是指向共享内存中缓冲区的地址指针），把与原语相关的参数写入ECB中，并利用处理器间通信软件，把ECB的地址指针传递给SASE，将与原语有关的用户数据写入独立的用户数据缓冲区（UDB）。并将与原语相关的UDB的地址指针和UDB包含的数据量的指示一起写入ECB中。

SASE实体接收到服务原语（ECB）后，就利用原语的参数和与联系相关的协议状态信息产生该层的PCI。这个PCI添加到UDB中已有的用户数据之后，形成一个SASE APDU。此时，
851 SASE获得与表示层协议实体（任务）相关的空闲ECB，将相应的原语类型和参数信息写入（包含APDU的UDB地址指针和更新的内容长度等），并向本地实时内核发出一个发送消息请求。内核把ECB的地址指针传递给表示层任务的输入队列，如果输入队列是空闲的（等待入事件），则调度任务运行。

各个层的任务完成相似的功能，首先把自身的PCI加入UDB中，然后利用该层空闲的ECB把相应的原语和相关参数传递给下一层，如图14-20(a)所示。前面提到，每层的PCI都是一个按具体语法构成的字节串。因此，每个UDB只是一个字节数组。UDB的内容最终在网络接口硬件和软件的控制下传输出去。

显然，当UDB通过所有协议层传递之后，UDB中包含的用户数据量可能从几百个字节（如仅包含PCI）到几千个字节（如传递一个文件的内容）不等。为此，每个UDB都是固定长度的，并采用链接表策略处理超出这个长度的复合对话单元（PDU）。就是说，如果一个UDB已经满了，
852 要从空闲列表中获得一个新的UDB指针，并与前UDB链接，这种方案如图14-20(b)所示。

14.4.2 用户元素的实现

从14.4.1节中得知，UE接收的原语有两个来源：来自用户AP和经由处理器间通信软件来自SASE接口。显然，这些原语可能有不同的类型。例如，来自用户AP的原语可能是需证实服务（即需要返回证实原语的请求原语）或无需证实服务（即不需要返回证实原语的请求原语）。同样，来自SASE接口的原语类型也可能不同。例如，可能是对前一个请求原语返回的证实原语，也可能是不需要响应的指示原语，或需要响应的指示原语。

因此，为了UE能正确响应每一个收到的原语，有必要在UE中保持上次收到有关SASE会话的原语记录。实际上，利用类似于协议机中定义的事件—状态表是最好的实现方法。正如协议机的实现策略一样，UE为每个活动会话保留一个状态变量，它指明了当前接口状态，即前一个接收到的原语。当接收到新来的原语（事件），这个状态变量根据事件—状态表确定所要进行的处理。

举例说明，假设一个用户AP连接到了一个UE上，通过OSIE支持的FTAM访问一个远程文件服务器AP时，提供的一组用户原语如下：

- f_open （input_file，status）
- f_read （input_file，message_buffer，status）
- f_write （input_file，message_buffer，status）
- 853 f_close （input_file，status）

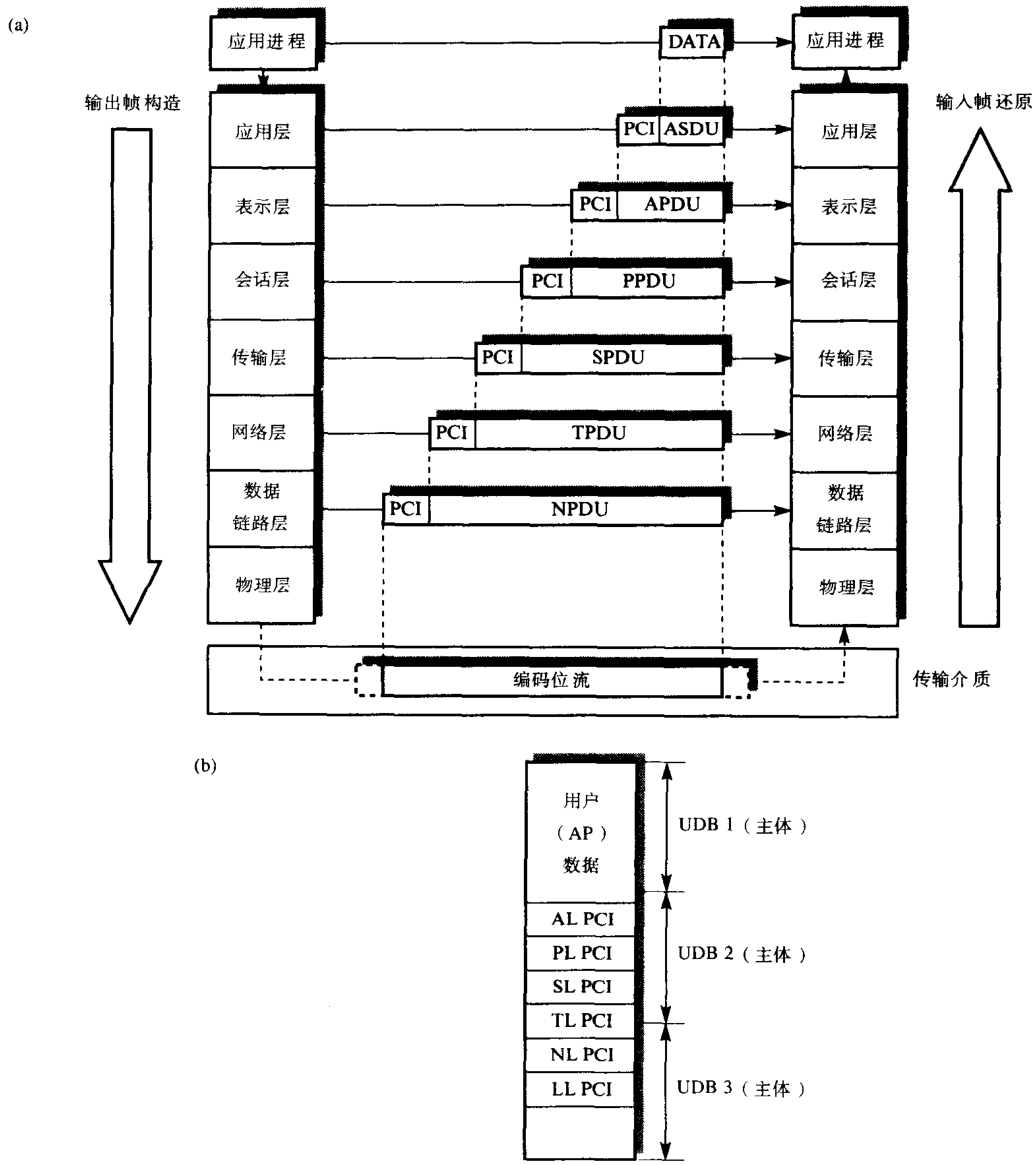


图14-20 层间交互

(a) 示意图 (b) UDB结构和内容

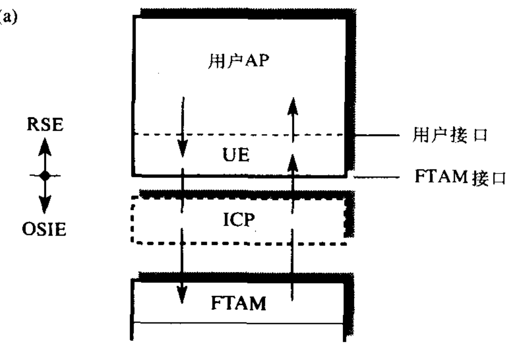
参数input_file是将要处理的文件的完整路径名（标题）；它包括服务器系统惟一的OSIE名称、系统中的FTAM实体名称和将要操作的文件名称。参数status是网络请求成功或失败的指示，以及失败的原因。参数message_buffer是一个指针，它指明传输请求（读或写）的数据的地址。

每个原语都对应一个库程序，所有库程序，即FTAM UE，连接到用户AP上。图14-21显示了一个UE的事件—状态表。

可以容易地从所提供用户原语的列表推断出：每个原语都能产生多个FTAM原语。例如，库程序f_open会产生一组与服务相关的FTAM原语：F_INITIALIZE原语、F_SELECT原语和

F_OPEN原语。类似地，f_read程序也会产生一系列原语：F_READ原语、F_DATA原语和F_TRANSFER_END原语。从而，程序f_open首先会从目录服务获得远程（被叫）FTAM实体的完全限定地址，然后产生一个F_INITIALIZE.request原语，把它和相关参数放入一个FTAM ECB中传递给本地FTAM实体。通常，这是使用相应的进程间通信原语通过进程间通信ICP软件实现的。与f_open库程序连接的用户AP等待F_INITIALIZE.confirm原语（ECB）时，被挂起。相应地，接口进入WTINICF状态。

(a)



(b)

原语 \ 状态	CLOSED	WTINICF	WTSELCF	WTOPNCF	OPEN	---	
f_open	1						来自用户接口
f_read					5		
f_write							
f_close							
F_INICF		2					来自FTAM接口
F_SELCF			3				
F_OPNCF				4			
⋮							

1: F_INIRQ, WTINICF

2: F_SELRQ, WTSELCF

3: F_OPNRQ, WTOPNCF

4: OPEN（返回）

F_INIRQ/CF = F_INITIALIZE.request/confirm
F_SELRQ/CF = F_SELECT.request/confirm
F_OPNRQ/CF = F_OPEN.request/confirm
WTINICF = Wait for F_INITIALIZE.confirm
WTSELCF = Wait for F_SELECT.confirm
WTOPNCF = Wait for F_OPEN.confirm

图14-21 FTAM UE 结构
(a) 接口 (b) 事件—状态表

F_INITIALIZE.confirm原语和相应参数，使用进程间通信原语和ICP软件，通过本地FTAM实体（在FTAM ECB中）传送。用户AP在f_open库程序挂起点上重新调度。从事件—状态表中可以推出，用户AP发送F_SELECT.request原语及其相应的由源呼叫中得到的参数，并进入WTSELCF状态。类似地，接收到F_SELECT.confirm原语，它会发送一个F_OPEN.request原语。最后，接收到F_OPEN.confirm原语，接口进入OPEN状态，立刻用

f_open库程序呼叫，并向发起用户AP返回，状态参数表明本次呼叫成功与否。

假设f_open库程序成功，通常紧接的是f_read或f_write呼叫，每个呼叫又产生一组FTAM原语。最后，当所有传输完成后，用户AP会使用f_close库程序启动联系中止，即对FTAM事务中止。

可以得出，如果采用上述用户原语的软件已存在，那使之适用于开放系统只需要改写FTAM接口所需的库程序即可。然而要记住，除了要生成相应的FTAM原语外，还必须把文件内容转换成为协商的传送语法，或进行反向转换。也可以在服务器端的UE采用特定服务器（文件系统），把输入FTAM原语转换成传送语法。 854

14.4.3 层管理

除了每层相关的任务（协议实体）之外，整个的通信子系统还包括两个任务：一个是定时器任务，它执行与各个协议实体（状态机）相关的超时功能；另一个是系统管理任务，如名称所示，它负责两层之间以及系统的管理功能。这些功能包括收集协议差错统计信息和每个协议层（实体）的操作参数的设置信息。一个完整的通信子系统如图14-22所示。 855

1. 定时器任务

如第11章提到的，为了确保每个协议机的入事件都是原子的，定时器任务和每个协议实体（任务）之间的接口，采用独立的任务间邮箱或队列实现。因为整个通信子系统只有一个定时器任务，所以通常只有一个相关的输入队列。并且，如同其他的任务间通信一样，所有与定时器任务之间的通信仅涉及某种类型的ECB。 856

与定时器任务有关的一组用户服务原语如下：

- TIMER.start（layer ID，timer ID，time）
- TIMER.cancel（layer ID，timer ID）
- TIMER.expired（timer ID）

因为定时器任务只有一个输入队列，为了识别发送原语的协议层，在两个输入原语start与cancel中各有一个层ID参数，ID参数标识出发送该原语的协议层。另外因为每层可能有若干个定时器在同时运行，所以使用定时器ID参数标识出该原语使用的是哪一个定时器。通常，为此目的还使用连接标识符。时间参数指明基于系统时钟滴答的时间间隔，定时器必须经过了规定时间参数后，定时器任务才通知相应的协议实体，指定的定时器已超时。

为了启动超时操作（例如，限制协议实体等待出事件适当响应的时间），协议实体首先要获得一个空闲（定时器）ECB，并把层和定时器标识符以及所需的时间间隔作为参数写入。然后通过向本地实时内核发送一个任务间通信原语（例如发送消息），启动传送ECB指针给定时器任务的输入队列（邮箱）。接收到这个请求后，定时器任务即在指明定时器标识以及相应定时长度的表中生成一项记录，定时长度实际上是一个计数变量。

通常，定时器任务是由系统时钟中断驱动的。这意味着信号（中断）是按预先规定的时间间隔产生的，等于最小的时钟滴答时间，如一秒钟，然后任务定时器被调度运行。首先，它要确定在输入队列中是否有等待处理的ECB，如果有就进行处理。与当前激活的定时器有关的时间间隔随着嘀答递减。如果定时器递减到0，表明定时器的定时间隔到期。在ECB中产生一个TIMER.expired原语，并使用任务间通信原语把ECB指针传递给相关层（定时器）的输入队列。当层任务下一次调度运行时，它不仅会检查有关上面层和下面层（协议实体）的输入队列，还需检查来自定时器任务的输入队列。每个队列中的每项记录按原子方式处理，如果该记录来自定时器任务的输入队列，则发生因超时而引发的特定操作。

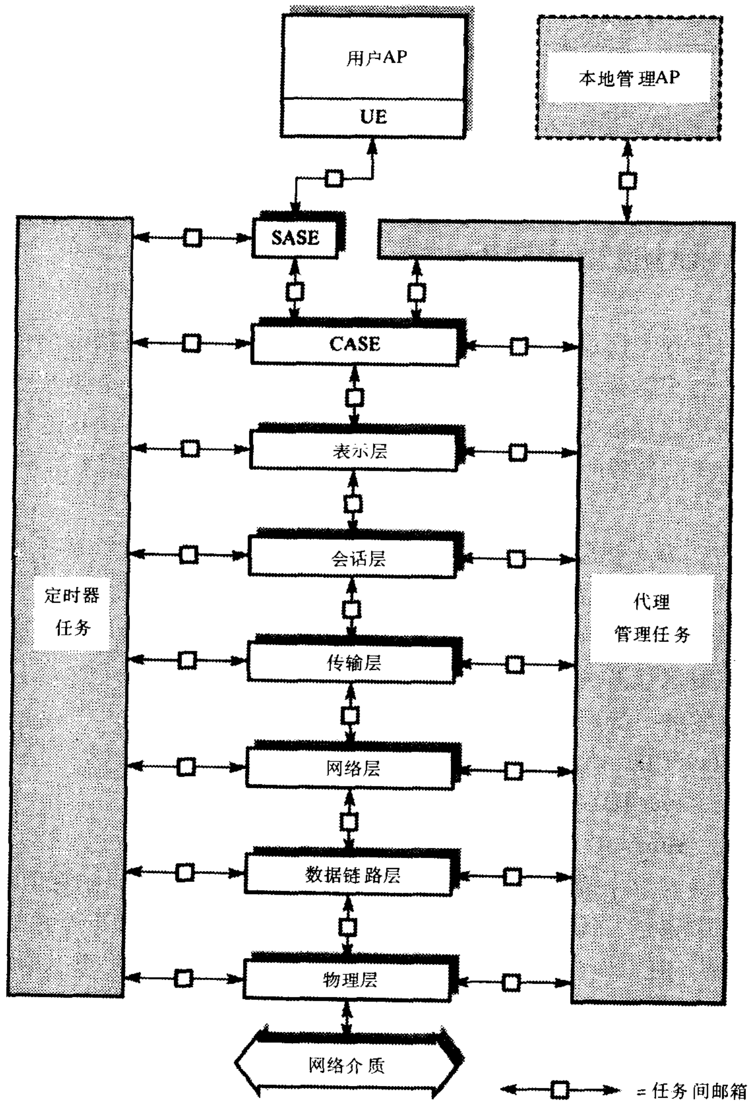


图14-22 完整的通信子系统示意图

如果对一个特定的出事件启动一个定时器后，协议实体接收到相应的响应（入事件），则它会使用实时内核支持的任务间原语，向定时器任务输入队列发送一个相应的TIMER.cancel
857 原语（在ECB中），来撤销指定定时器。当定时器任务下一次调度运行时，它从定时器表中删除相应的定时器。

2. 代理管理任务

每层除了要保持协议机当前操作状态的状态信息（变量）之外，还要保持一组变量，用作层管理的各种操作统计。每层任务收集的一些统计信息的例子如下：

- ACSE层：接收到的拒绝A_ASSOCIATE.response原语的个数；接收和发送的ABORT APDU的个数。
- 表示层：接收和发送的不成功的CPR（表示层连接响应）PPDU的个数及其原因代码；接收到的未能识别的PPDU的个数。

• 会话层：接收和发送的RF（拒绝）SPDU的个数；接收和发送的AB（异常中断）SPDU的个数。
• 传输层：已发生的协议差错的次数；传输TPDU时发生的定时器超时的个数；具有错误校验和的TPDU的个数。
• 网络层：接收和发送的NPDU的个数；由于未知NSAP而丢弃的NPDU的个数。
• LLC层：接收和发送的测试LPDU的个数；协议差错的个数。
• MAC层：冲突和重传尝试（CSMA/CD）的个数；令牌传递失败（令牌环总线）的次数。

通常，多数信息都是用简单计数器维持的，当响应网络管理AP请求时，**代理管理任务**（AMT）可以请求其中的内容。然而，如果某层发生了提示可能失败的特定事件，例如达到某个变量的预定义的边界限制时，该层（任务）可以直接通知AMT。

AMT必须有一个工具可以通知层所要使用的特定操作参数，例如：

• 窗口限制（传输层）
• 超时间隔（所有层）
• 路由表内容（网络层）
• 最大重传限制（CSMA/CD）
• 目标令牌循环时间（令牌环总线）

对于定时器任务，必须定义一组服务原语（以及相应ECB类型和任务间队列结构）使这些功能以原子方式实现。下列是一组用于AMT与各层交互的原语： 858

• GET_VALUE.request/confirm（参数值） AMT用来从一个特定层获得统计信息。
• SET_VALUE.request/confirm（参数ID，参数值） AMT用来对一个特定层设置操作参数。
• ACTION.request/confirm（动作ID，动作值） AMT用来向一个路由表添加一个或多个条目。
• EVENT.indication（层ID，事件标识符，事件值） 一个层实体用来通知AMT达到了一个阈值限制。

14.5 相关标准

开放系统协议族的目的就是使AP能够运行在不同厂家的计算机系统上，协同执行特定的分布式处理功能。协议族中的表示层服务确保交换消息的语法在所有系统上有相同的含义。然而，表示层服务并不关心所传递消息的结构与含义。它只是把数据看作一个适当定义的数据类型的流（例如字符串），然后使用适当的传送语法和语法转换（如果必要），确保对于每个协作的系统，这些语法都是兼容的。应用进程对所传递消息的结构和含义进行翻译。

显然，如果应用进程彼此协作完成特定的分布式处理任务，必须对应用中有关的消息的结构和含义以及消息的交换顺序作出协商。一种排序方法就是为应用定义虚拟设备，例如虚拟文件服务器。所有交换消息必须按虚拟设备定义的次序进行。另外，各方必须知道交换消息的类型和结构。

除了在消息处理系统中传递消息到目标所需的协议外，它们基于消息信封中的地址，还有位于通信栈之上的协议，它们是关于交换消息的类型和结构的，这是主体部分。第13章讨论了个人间的消息协议，但实际上有一部分开发的标准是关于其他类型信息的。例如，**电子数据交换**（EDI）标准用来定义商业文档的交换，如订货单、发票和发货记录等。类似的，制

859 造业中的产品定义了有关设计、材料和其他文档的标准，这些是有关办公自动化文档的结构和内容的标准。虽然所有这些标准都要关联到公有（和专用）X.400消息系统，但当文档使用文件传递进行交换时，这些标准同样适用。这些标准的完整的定义已经超出了本书的范围，但下面会给出有关的两个标准的概要。

14.5.1 EDI

EDI标准的目标就是使不兼容的计算机彼此之间能够以标准的消息格式交换贸易文档，例如订货单、购货文档和购货价格等。通常，这些标准是由行业中的领头企业推动的，它们需要各种不同的设备和产品的供应。只有当所有的相关文档都用标准格式实现之后，整个贸易流通才能自动化。这是一个以通信为目的的开放系统的最终目标。

EDI中的消息标准是一种具有定义结构的常规格式。构成一个文档的所有内容、数量和位置等要素都能使用标准进行定义。图14-23(a)给出的标准术语是在一个应用环境中使用的表格的一般结构。

文档中的页被称为**事务集**。它是由若干**数据段**组成的，每个数据段对应一行或表格中的一个框。一个数据段由一个或多个**数据元素**组成。整体的格式如图14-23(b)所示。每个组成部分的开始和结尾都用规定的字符串表示，单个的数据段/元素都使用分隔符分开。标准中的编码规则相对来说比较简单，所使用的字符集也很有限。

各个通信方对每种应用都定义并使用了一组特定的事务。现在已经定义了许多消息标准，它们分别应用于各种行业，例如汽车行业、铁路行业和食品行业等。

14.5.2 ODA

术语ODA是**开放文档结构**的缩写。办公文档包括备忘录、信件、表格和报告等，它们不仅包含文本信息还包含各种媒体，例如公司标识的图像或机构的纹章。

一个文档包含有关**内容**的信息，还包含有关**结构**的信息。文档内容是能够以两维形式表示的任何信息，例如纸张上的打印信息或屏幕上的显示信息。结构则用来进行如下处理：

- 对文档进行划分，例如一页中的图像区域或各种不同的内容元素的类型（文档的布局结构）。
- 860 对文档中具有逻辑含义的部分进行划分，例如章和段（文档的**逻辑结构**）。
- 对不同的内容类型使用不同的编码。
- 允许对文档进行加工。

表示结构化文档的规则作为一个整体被称为**文档结构**。

为了相互交换的目的，文档表示成组成部分的集合，各个组成部分都有一组属性。每个属性都有一个名称和一个值，表示结构元素的一个特性。各种组成部分的类型如下：

- 文档简介
- 逻辑结构
- 布局结构
- 内容描述
- 表示风格
- 861 布局风格

文档简介包含一组属性，它们完整指明了文档的规定特性。**内容**描述由一组内容元素的定义组成。**表示和布局**风格都是一组属性值，它们说明了文档内容在表示媒介上的格式和表

现。把风格从文档结构中分离出来，这就允许修改一个文档的布局和表示而不影响它的逻辑结构。例如，在出版行业这点是非常重要的。通常，作者关心的是内容，而编辑关心的是修改布局使之更好地与其他消息组合起来。图14-24显示了逻辑结构和布局结构之间的关系。

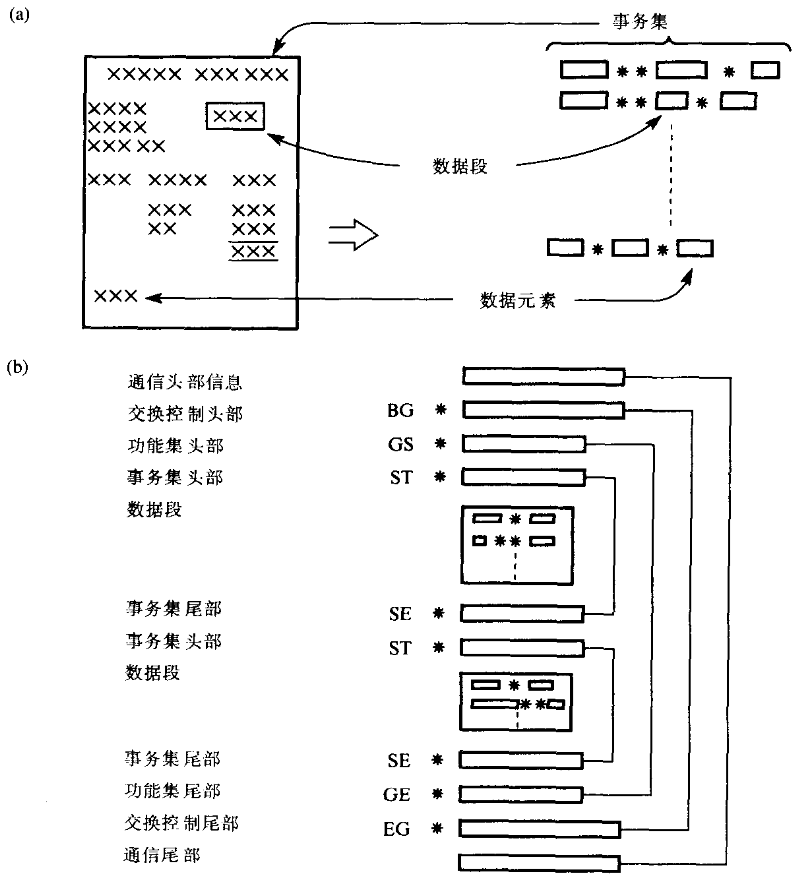


图14-23 EDI术语

(a) 表格的编码 (b) 整个文档结构

布局描述包括页、帧（一个矩形区域）、块、以及页中的帧和块的定位和度量数据。逻辑描述是独立于布局描述（例如页）的。例如，文本单元是最低级，它可作为段或符号定义； 862
它们可以组合成为一个逻辑元素（对象）。逻辑结构定义了这些元素的正确顺序，而布局结构定义它们是如何安排的，例如在一页或多页上。

不同于EDI，ODA使用ASN.1定义各种结构。目前，办公自动化软件和设备的主要厂商都对此提供了支持，最终，当前多数字处理软件中所必须的乏味的转换和格式化操作都将消失。

习题

14.1 给出一个可以引用网络服务器的符号名的例子。列出在如下环境中的服务器的等价地址的成分。

（a）TCP/IP环境

（b）OSI环境

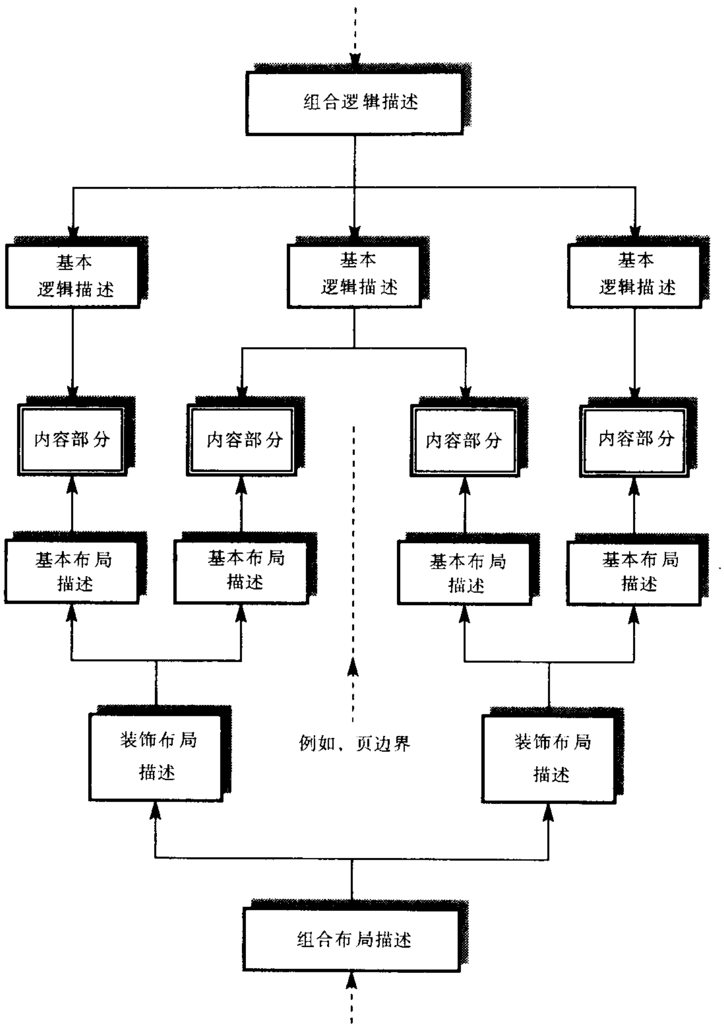


图14-24 逻辑描述与布局描述间相互关系的实例

14.2 给出下列结构的网络服务器的符号名称的例子：

(a) 平面结构

(b) 层次结构

用每个结构方法确定有关名称管理与目录划分的实现。

14.3 给出一个TCP/IP域名系统层次名的例子。在层次结构中确定选择域名（标记），并解释整个目录管理的名称分配原理。

14.4 确定域名系统的有关协议。画出有关客户端系统、服务器和域名服务器的协议的示意图。描述为了获得服务器的TCP/IP地址在这些协议之间交换的消息序列。

14.5 借助域名服务器层次图，解释当请求服务器是下列情形时，域名解析是如何进行的，并给出例子：

(a) 属于同一个名称服务器

(b) 位于层次体系的更高层

解释术语“名称缓存”的含义，以及如何使用该机制使引用次数最小。

14.6 借助例子解释X.500目录系统的下列术语的含义：

(a) 目录信息树

(b) 相对判别名

(c) 判别名

(d) 别名

14.7 解释X.500目录系统中的日录用户代理（DUA）和目录服务代理（DSA）的功能。解释DUA请求一个已命名服务器的PSAP地址时所要交换的消息，并清楚说明相关属性类型和属性值的用法。

14.8 借助于名称服务器层次图解释如下术语的含义：

(a) 引用

(b) 链接

(c) 多播

14.9 确定与DUA和DSA应用进程有关的通信协议。借助图，给出在DUA和DSA之间交换的消息序列，通过这些协议确定一个已命名对象的地址（例如服务器）。

14.10 解释缩写名MAP和TOP的含义，并指出每一个有关的应用域和协议。 863

14.11 使用组成TCP/IP协议族的各种协议，指出执行一个网络请求所要交换的消息序列。例如，使用网络文件服务器的f_open请求。

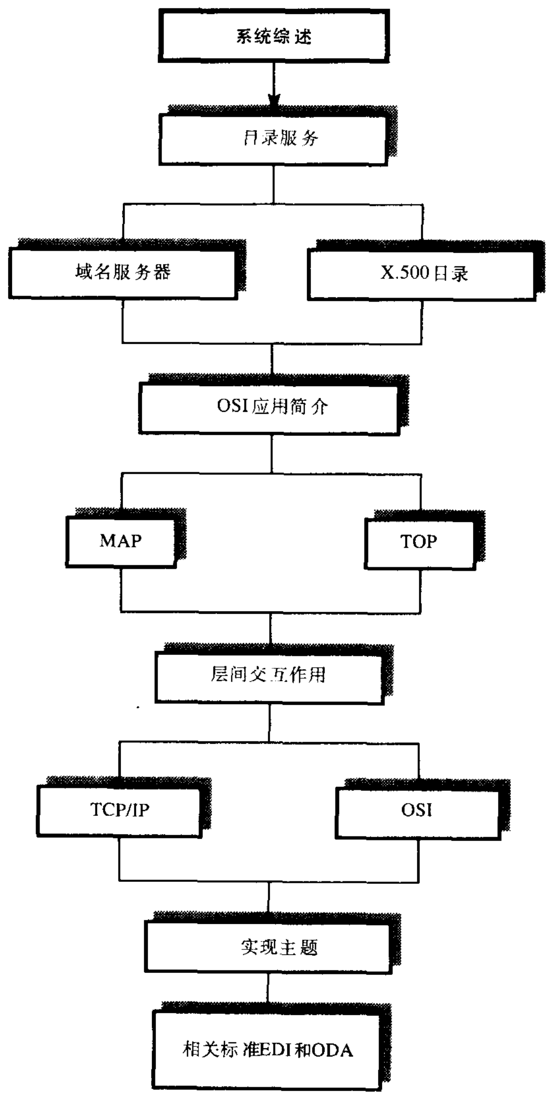
14.12 使用OSI协议族重做习题14.11。

14.13 区分OSI协议族中的术语“地址选择符”和“连接标识符”。

14.14 对于TCP/IP协议族和OSI协议族，用涉及的进程描述一个一般实现结构，以及这些进程如何与运行进程的机器上的本地操作系统交互。

14.15 解释为了通信目的使用OSI协议栈的应用进程的用户元素的作用。举一个例子描述它是如何协调应用进程使用的服务和协议栈接口所提供的服务的。 864

本章概要


系统综述
目录服务
域名服务器
X.500目录
OSI应用简介
MAP
TOP
层间交互作用
TCP/IP
OSI
实现主题
相关标准EDI和ODA

865

附录A 正向差错控制

介绍

利用自动重复请求（ARQ）差错控制机制，可以在每个传送消息的末尾添加附加的校验数字，利用这些数字，接收方可以发现消息中出现的错误，并确定错误的类型。如果接收方在接收到的消息中发现了错误，它会通过额外的控制规程向发送方请求一份新的消息。使用正向差错控制（FEC），我们可以在消息的末尾添加充分的校验数字，通过这些数字信息，我们不仅可以在接收到的消息中发现一个或多个错误，还能够对这些错误出现的位置进行定位。而且，因为消息是以二进制形式编码的，所以只需要对已经确认为错误的信息位进行转换就可以得到正确的信息。

实际上，用于纠错的附加校验数字的数目要远远大于仅仅发现错误所需要的数目。在大多数包含全球性连接的应用中，ARQ方法同第4章描述的一样，它比FEC方法更有效，因而经常被使用。这种方法依赖于用于确认目的的返回路径。然而，在某些应用中，可能无法使用返回路径，或者花在返回路径上的往返延迟比路径上数据传输的时间还长。例如，当从太空探测器返回信息时，通常使用非定向的连接。同样，对于很多卫星链路，传播延迟是如此之大，以至于在传递信息之前发送站要预先发送几百个信息，而且确认信息也要以相反的方向进行同样的发送。在这种应用中通常使用FEC方法，它同ARQ方法结合起来，能减少需要重传的信息的数量。附录A的目的就是简单介绍FEC方法中使用的技术。

A.1 单位汉明码

实际上，FEC方法在数据传输中的应用是有限的。我们将简要地介绍它的编码原理的相关主题和术语。显然，关于编码原理的详细描述已经超出了本书的范围，在这里我们只对它做一个简要的介绍。如果你对编码原理感兴趣，并且打算获得更多的有关知识，你可以查阅本书后面的参考文献中列出的书。 867

编码原理中用来描述组合信息单元的术语称为**码字**，它由有用的数据位和附加的校验位组成。两个有效码字之间最小的不同位的个数称为**汉明距离**。例如，考虑这样一种编码方法，每个码字由7位数据位和1位奇偶校验位组成。假定我们使用了偶校验位，用这种编码方法得到的连续码字如下：

0000000 0
0000001 1
0000010 1
0000011 0

从上面的列表中可以看出，这种方法的汉明距离是2，因为每两个有效码字之间至少有2位是不同的。这意味着，由于2位的错误会导致产生一个不同但是有效的码字，所以使用这种方法不能发现2位的错误。然而，如果码字中只有1位发生错误时，就会产生一个无效的码字，所以利用它可以发现所有的1位错误。

通常，一种编码方法的检错和纠错特性都同汉明距离有关。就是说，为了发现一个n位错误，必须使用具有$n+1$汉明距离的编码方法；而如果要能够纠正n位错误，必须使用具有$2n+1$汉明距离的编码方法。

最简单的纠错编码方法就是单位汉明码。使用这种代码，我们不仅可以发现接收到报文中的单位错误，还能发现错误所发生的位置。通过对错误位取反，就能得到正确的码字。因为原始报文在编码和后续的解码程序中作为单独的块（帧）进行处理，所以这种代码也称为**块代码**。通常，对于块代码来说，每个原始的k位信息被编码成一个n位的信息（n大于k）。编码器的作用就是产生一个（n，k）码字。比率 k/n 称为**码率**或**码效率**，而比率$1-k/n$称为**冗余比率**。

为了阐述这种原理，考虑一个对单位错误进行检测和纠正的汉明码，假定每个码字包含7位的数据字段，例如ASCII字符。因为每个校验位都位于2的n次位，所以这种汉明码需要4个校验位。这种代码被称为一个（11，7）块代码，它具有7/11的码比率和1－7/11的冗余比率。例如，1001101的（11，7）块代码的各位的位置如下：

$$
\begin{array}{ccccccccccc}
11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\
1 & 0 & 0 & x & 1 & 1 & 0 & x & 1 & x & x
\end{array}
$$

868 标记为x的4个位就是校验位，它们是按如下的方法计算的。把值为1的各个位的位置数，用4位的二进制形式表示，并把值相加，再把和模2，这样4位的校验位就是下面的4位和：

$$
\begin{array}{rcl}
11 & = & 1\ 0\ 1\ 1 \\
7 & = & 0\ 1\ 1\ 1 \\
6 & = & 0\ 1\ 1\ 0 \\
3 & = & \underline{0\ 0\ 1\ 1} \\
 & = & \underline{1\ 0\ 0\ 1}
\end{array}
$$

于是，得到要传输的码字为：

$$
\begin{array}{ccccccccccc}
11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\
1 & 0 & 0 & \boxed{1} & 1 & 1 & 0 & \boxed{0} & 1 & \boxed{0} & \boxed{1}
\end{array}
$$

同样，在接收方，把值为1的各个位的位置数（包括校验位），用4位的二进制形式表示的，并把值相加，再把和模2。如果没有错误发生，那么得到的结果为0：

$$
\begin{array}{rcl}
11 & = & 1\ 0\ 1\ 1 \\
8 & = & 1\ 0\ 0\ 0 \\
7 & = & 0\ 1\ 1\ 1 \\
6 & = & 0\ 1\ 1\ 0 \\
3 & = & 0\ 0\ 1\ 1 \\
1 & = & \underline{0\ 0\ 0\ 1} \\
 & & \underline{0\ 0\ 0\ 0}
\end{array}
$$

现在考虑发生了一个单位错误：假定11位的值从1变成0。于是，新的结果如下：

$$
\begin{array}{rcl}
8 & = & 1\ 0\ 0\ 0 \\
7 & = & 0\ 1\ 1\ 1 \\
6 & = & 0\ 1\ 1\ 0 \\
3 & = & 0\ 0\ 1\ 1 \\
1 & = & \underline{0\ 0\ 0\ 1} \\
 & & \underline{1\ 0\ 1\ 1}
\end{array}
$$

首先，结果不是0，这表示有错误发生，其次，结果为十进制的11，这表示错误所发生的位置是11位。把11位的值取反，就可以得到正确的码字，从而得到数据位信息。

另外，如果码字中有2位发生错误，最后得到的结果也不为0，此时我们不能从最后的结果中找出错误所发生的位置。单位汉明码能够纠正单位的错误，以及发现2位的错误，但是不能够发现其他的多位错误。

如同我们在第2章看到的，在很多的数据通信网络中发生的错误类型通常不是单位错误或2位错误，而是连续的位错误。虽然基本形式的汉明编码方法并不适合这些网络应用，但是我们通常使用一个简单的技术对汉明编码方法进行扩展。 869

例如，考虑在一个很可能发生连续错误的单工通道上，请求发送一个由8个ASCII字符组成的数据块。控制设备首先把每个ASCII字符转换成11位码字的形式，从而得到由8个11位码字组成的数据块。控制设备并不把每个码字独立传输，而是传输组成数据块的8个码字的相同位。就是说，首先传输8个码字中的最高位，接着传输8个码字中的次高位，最后是8个码字中的最低位。而接收方的控制设备则执行相反的操作，把内存中的数据块重新组合，然后执行错误检测，如果有必要，对每个码字还要执行纠错操作。

这种方法带来的效果就是，首先，它可以用标准USRT设备作为传输接口电路，其次，也是更重要的，如果发生了一个最多7位的连续错误，它只影响每个码字中的一位，而不是影响一个或两个码字中的若干位。这意味着，如果在88位传输中只发生一个连续的错误，接收方能够确定所传输的字符块的正确拷贝。

虽然，刚刚描述的方法为汉明编码方法提供了一种有用的扩展，但是汉明码仍主要用于具有独立的单位错误的应用中；半导体存储器系统中的纠错就是一个例子。我们推荐的在数据通信系统中实现FEC的方法基于**卷积码**，下面我们将简要地介绍这种类型的编码方法。

A.2 卷积码

块代码是一种无记忆的代码，每个输出码字仅取决于当前被编码的k位信息。与此相反，卷积码是对连续的原始位流进行操作，从而得到连续的输出位流。由于这种编码方法自身的特性，所以对输入的位序列进行卷积（指定二进制操作），产生输出位序列。输出序列中的每位不仅依赖于当前的被编码位，还依赖于先前的源位序列，因此隐含了某种形式的记忆功能。实际上，如我们将要看到的，它采用了定长的移位寄存器的形式，被称为**限制长度**，而卷积（二进制）操作则利用一个或多个模2加法器来执行（异或门）。

A.2.1 编码

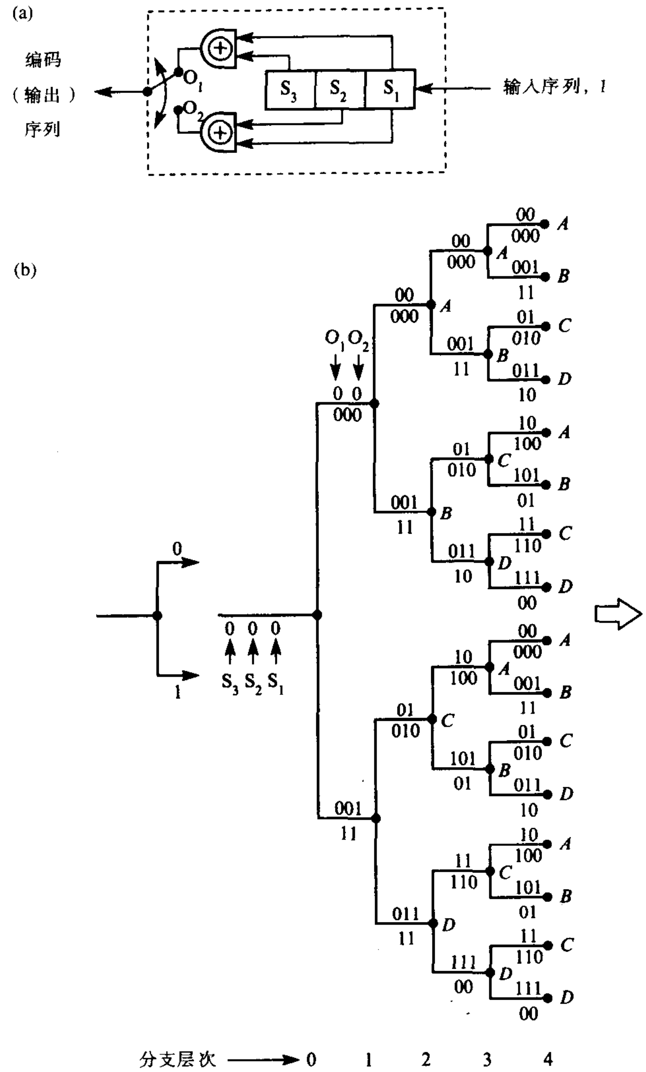
图A-1(a)显示了一个卷积编码器的例子。在这个编码器中，提供3位移位寄存器和两个模2加法器，执行卷积操作。对于输入序列中的每位，都有2位的输出，两个输出位都来自两个模2加法器。图中的编码器被称为1/2（k/n）比率的卷积编码器，它的限制长度为3。 870

由于卷积编码器的记忆功能，我们必须使用一种简便的方法，确定给定的输入序列产生的输出序列。有三种可以使用的技术，每种都对应一种图形表示方式：树形图，状态图和格状图。实际上，由于最后的格状图方法对于编码操作的示范更为有用，因此经常被使用。然而，在我们得出这个结论之前，必须了解利用树形图和状态图怎样为每个可能的输入序列确定输出。

871

图A-1(b)显示了一个对应图A-1(a)中编码器的**树形图**的例子。树中的分支点称为结点，在树中，对应每个结点都有两个可能的分支，上面的分支对应输入位为0的情况，下面的分支对

应输入位为1的情况。我们在结点的每个分支线的旁边显示了对应于可能的分支的输出位。



图A-1 卷积码

(a) 编码器电路实例 (b) 树形图表示

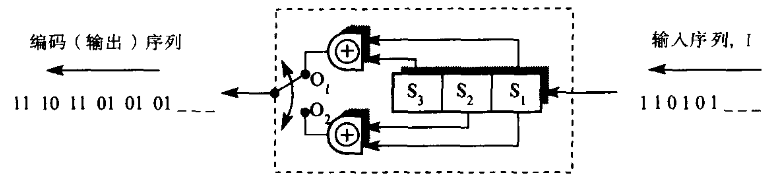
如同我们看到的，在树状图中，对于每个新的输入位，树中分支的数目都是输入数目的两倍。然而，因为第二层的分支之后只有四个分支结点，所以在第二层的分支之后的树，都是重复的。它被称为**状态**，图中显示了*A*、*B*、*C*和*D*四个状态。

如同我们在图中看到的，不考虑结点在树中的位置，对于树中的每个结点都有一对相同的输出位，并产生一个新的结点状态。例如，对于结点*C*，有一对可供选择的分支：对应输入0，产生10输出，以及新的状态*A*；对应输入1，产生01输出，以及新的状态*B*。

一旦使用树状图为编码器确定状态，就可以画出一个**格状图**。图A-2(b)中显示了对于同一个编码器的格状图的例子。我们将看到，在第二层分支之后，树状图中利用重复的特性，以更简化的形式描述所有可能的编码器输出。

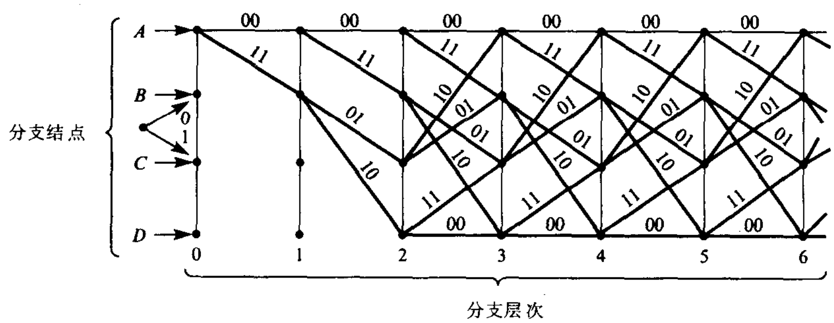
格状图显示了对于该编码器的所有可能的输入位序列得到的结果。对于特定的输入序列得到一个路径，即输出位序列。图A-2(c)显示了从格状图中得出的对应于输入序列110101...的路径，以及输出序列。

(a)

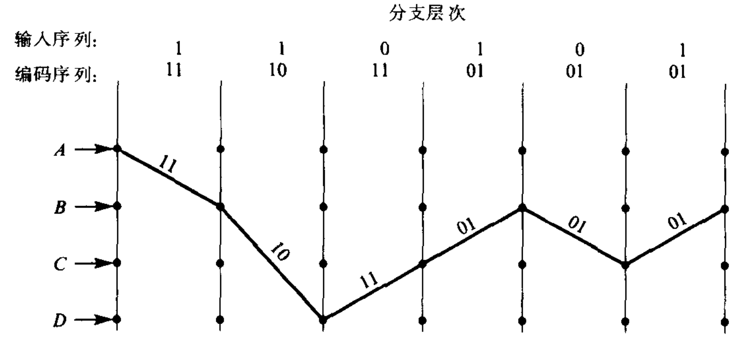


输入序列 I	移位寄存器内容 S_1	S_2	S_3	输出序列 O_1	O_2
1	1	0	0	1	1
1	1	1	0	1	0
0	0	1	1	1	1
1	1	0	1	0	1
0	0	1	0	0	1
1	1	0	1	0	1
⋮		⋮		⋮	

(b)



(c)



图A-2　卷积码

(a) 电路　(b) 格状图　(c) 输出实例

初始时假定移位寄存器被清零，即所有的位都被设置为0。当输入序列的第一位数据进入移位寄存器后，寄存器的内容变成001。两个模2加法器的输出分别是，0+1=1（加法器1）和0+1=1（加法器2）。于是，第一个二位输出为11，在下一个输入位进入移位寄存器前，结果被发送出去。因为输入位是1，所以选择格状图中的低分支路径，得到的输出为11。

当第二个输入位进入移位寄存器之后，寄存器的内容变成011。两个加法器的输出是，0+1=1（加法器1）和1+1=0（加法器）。因此，得到的输出位是10，同样在下一个输入位进入移位寄存器之前，结果被发送出去。因为输入位是1，所以选择格状图中的低分支路径，得到的输出为10。接下来，第三个输入位进入移位寄存器，寄存器的内容变成110，得到的输出位是11，1+0=1（加法器1）和1+0=1（加法器）。因为输入位是0，所以选择格状图中的高分支路径。接下来的输入序列的处理同上。

A.2.2 解码

解码器的目标就是，在给定接收到的位流（可能有错误）和源端编码器的情况下，确定
872 最为可能的输出序列。而解码器的任务就是把接收到的序列同对应编码器可能获得的所有序列进行比较，并从中选出与接收到序列最相似的序列。如同A.1节提到的，两个码字之间的汉明距离是它们之间不同位的位数。因此，当在所有可能的序列中选择同接收到序列最相似的序列时，我们要计算所有可能的序列与接收到序列之间的汉明距离，并选择具有最小距离的
873 序列。显然，在这种情况下，把接收到的完整序列同所有可能的序列进行比较就成为必要的了。但是，在大多数应用中，这是不可行的，因此我们必须采取一个折衷的方法。

本质上讲，可以使用计数器保存实际接收的序列同每个可能的序列之间的距离，但是，在格状结构中每个结点只保留了一条路径。每个结点总是有两条路径，所选择的路径具有最小的汉明距离，而另一条路径则被终止。所保留的那条路径称为**生存路径**，而最终选择的路径是一条穿越格状结构的连续路径，所有路径的汉明距离的合计具有最小值。这种方法称为**韦氏算法**。解码器的目的就是根据接收到的序列找出最可能的路径，这种解码器称为**最大相似度解码器**。例子A-1给出了韦氏算法的描述。

实例A-1

假定一个信息序列为1001110...，它使用图A-1(a)中的编码器进行发送。利用该编码器的格状图，可以推导出，编码器产生的传输（输出）序列为：

11 01 10 11 10 00 11...

现在，假设发生了一个突发性错误，使得在传输过程中，编码序列中产生了两位错误。接收到的序列如下显示：

11 01 00 11 11 00 11...
　　　↑　　　↑

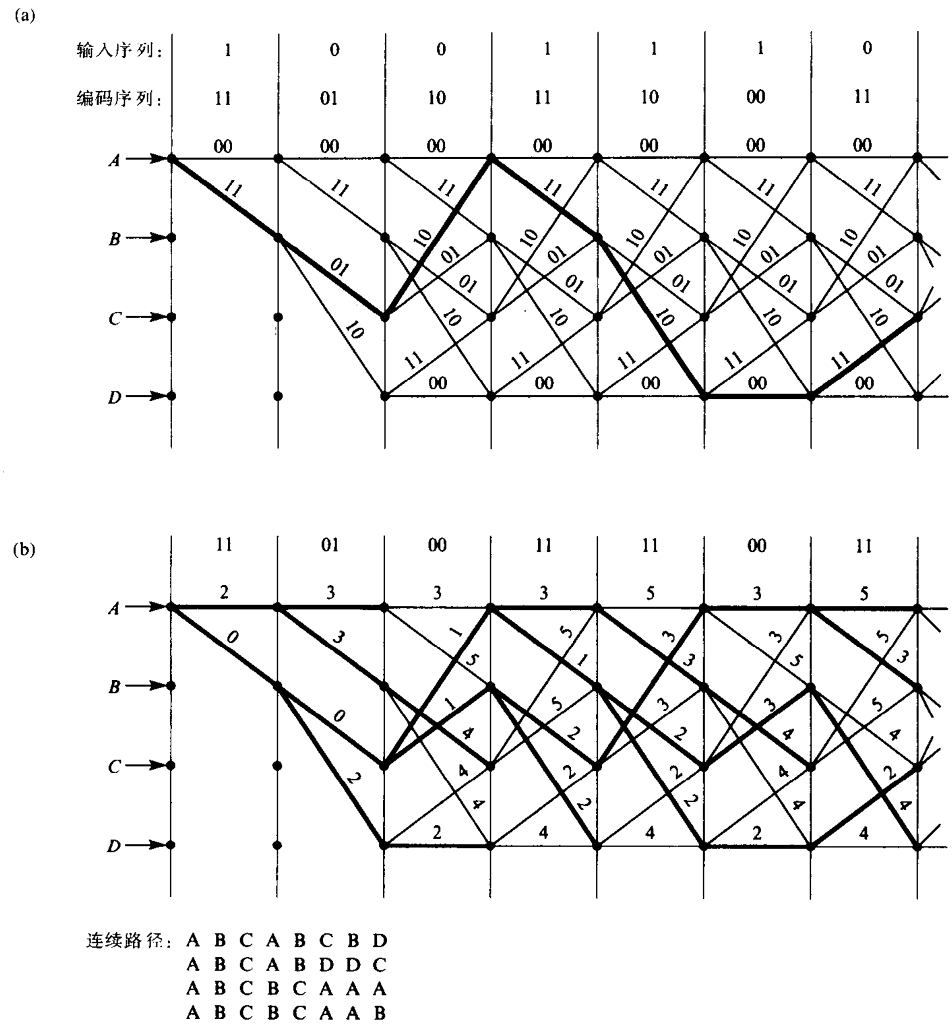
利用上面的序列，使用韦氏算法确定最可能的传输序列。

在编码和解码程序中采用的步骤见图A-3。图中的(a)部分显示了对应编码器原始输出的格状结构中的路径，(b)部分显示了所选择的生存路径。图A-3(b)中结点的每个路径旁的数字，表示该路径同实际的接收路径之间的累积汉明距离。

如果，所选的路径从根结点开始（0级分支），接收到的序列是11，对于路径00，两者之间的距离为2，对于路径11，两者之间的距离为0。这两个距离值被加到从该结点出发的路径上。因此，在1级分支上，所接收到的序列是01，从结点*A*出发的两条路径的汉明距离为1，对于路径00，距离为1，对于路径11，距离也为1。现在，每条路径的累积距离为2+1=3。同样，从结点*B*出发的两条路径，对于路径01，距离为0，对于路径10，距离为2，因此累积的距离分别为0+0=0和0+2=2。对于2级分支的处理程序同上面的程序相似。

874 然而，所选的程序从3级分支以及的分支开始。因此，从结点*A*出发的两条路径（3级分支）的累积距离分别为3和1，于是后者被选做该结点的生存路径，它在格状图中用黑体标出。结

点B、C、D使用了相似的选择方法。然而，我们能够看到，结点C的两条合并路径具有相同的累积距离4。在这种情况下，我们选择上面的那条路径。并且，当选择程序完成后，所有的后续距离都要累加到所选路径的累积距离上。



图A-3　卷积码

(a) 编码输出　(b) 生存路径

持续选择最有可能的路径和输出序列。虽然解码过程是连续的，但是通过检查格状图，我们知道：

- 格状图中只有4个有效的连续路径。
- 对应于路径$ABCABDDC$的距离是最小的。 875

因此，路径$ABCABDDC$就是所选的路径，对应的输出序列是11 01 10 11 10 00 11…，它对应于原始的编码（传输）序列。

最后要注意，FEC方法不可能发现所有的错误。通常，类似于卷积码的代码只是用来把一条链路上的错误发生率（误码率）降低到一个可以接受的水平。1/2比率的卷积码缩减率通常在10^2~10^3之间。因此，采用ARQ差错控制程序可以很好地提高整个链路的效率。 876

附录B　传输控制电路

如同3.6节所描述的，专门的集成电路可以执行不同类型传输控制机制中的大多数功能。我们将要描述有关面向字符数据传输中执行各种控制功能的电路。该电路被称为**通用同步/异步收发器**（USART）。

使用术语“通用”，是因为可以对设备进行编程，使之能够用两种面向字符传输模式进行操作：异步方式和同步方式。我们可以把预定义的位模式写入设备的一个内部控制寄存器中，从而选择特定的模式和操作特性。当设备以异步模式操作时，通常称为**通用异步收发器**（UART），当设备以同步模式操作时，通常称为**通用同步收发器**（USRT）。

一般来讲，UART以并行的方式从一个控制设备（如微处理器）接收字符，然后连同开始位和终止位以及奇偶校验位（如果有），一起以串行的方式发送出去。同样，在输入端，首先接收串行位流，然后从中分离出开始位和终止位，并检查奇偶校验位，最后由控制设备以并行方式把接收到的字符转换成为可读形式。它包含所有为了实现位同步的必要电路和附加的调制解调器控制电路。图B-1(a)中的示意图显示了USART中的主要寄存器；图B-1(b)显示了典型UART设备的接口安排情况。

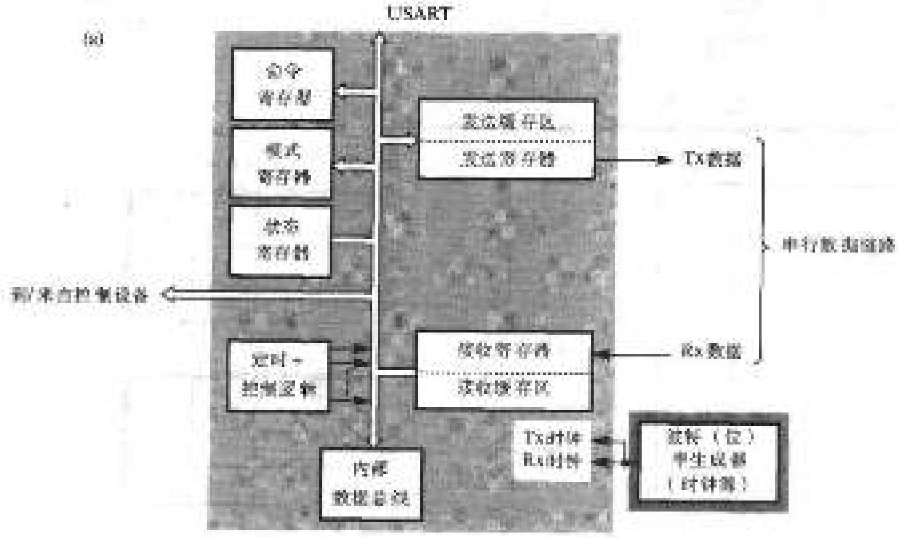
为了使用设备，首先要为**模式寄存器**装载所请求的用来定义请求的操作特性的位模式，这个操作被称为初始化。对于异步传输，用户可以选择每个字符有5、6、7或8位，奇校验位、偶校验位，或者没有奇偶位、一个或多个终止位和一个特定的时钟速率。最后一个也可以称为**波特率要素**。依赖于所请求的比特率，具有特定频率的时钟源被用于发送和接收时钟输入。异步传输中最为常用的比特率是110、300、1200、2400、4800、9600和19 200bps。因此，如果选
877 择的是16倍的时钟率，那么发送和接收的时钟输入分别是1760（110 × 16）和4800（300 × 16），等等。虽然有些设备的时钟电路是集成电路的一部分，但对于图中所显示的设备，我们假定时钟电路位于设备的外面。

图B-2(a)显示了Intel 8251A模式寄存器中各个位的含义。因此，假定模式字节01111010（16进制7A）被载入模式寄存器中进行初始化，该设备将以异步模式执行操作，每个字符有7位，1个偶校验位，1个终止位，和1个16倍时钟速率位。具有16倍速率的时钟源连接到发送和接收时钟输入上。

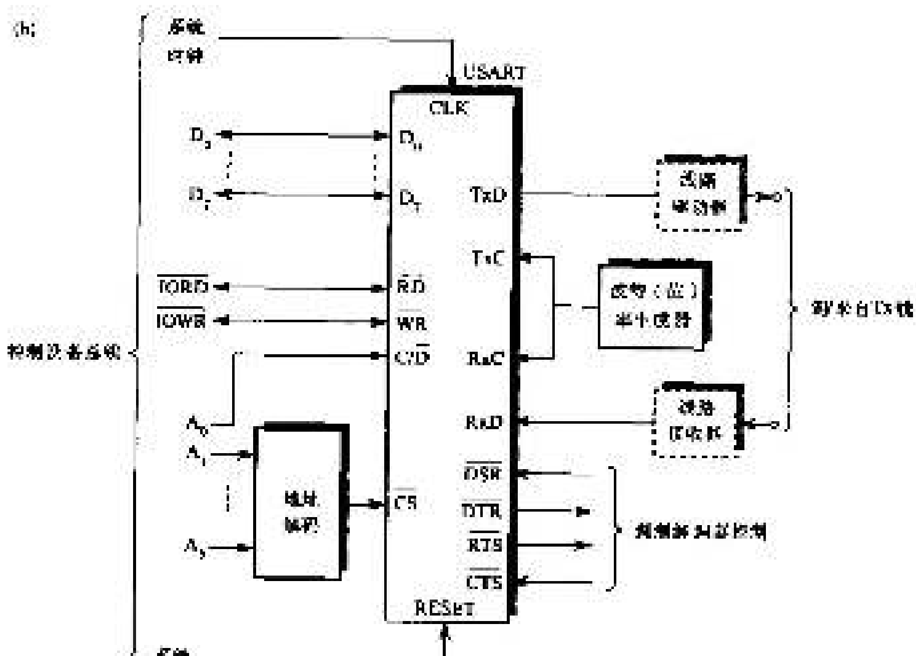
命令字节控制实际的设备操作。因此，当选择完请求的操作模式后（在发送数据之前），必须装载命令字节。图B-2(b)显示了Intel 8251A的命令字节中各个位的含义。正如我们看到的，这些位可以启用发送器和接收器部件以及控制调制解调器的接口。经常使用的三位是错误恢复（ER）、启用接收器（RXE）和启用发送器（TXEN）。因此，如果在发送数据之前，在模式字节之后装入00010011（16进制13）命令字节，任何的错误条件都会被恢复（在后面描述），
878 ~ 880 发送器和接收器都被启用。数据链路两端的UART必须设置成以同样的模式操作，并具有相同的操作特性。

通过读取第三个寄存器（**状态寄存器**）的内容并检测其指定位，控制设备可以确定UART的当前状态。这些是**状态位**或**标记位**；图B-2(c)显示了状态寄存器的组成。

(a)



(b)

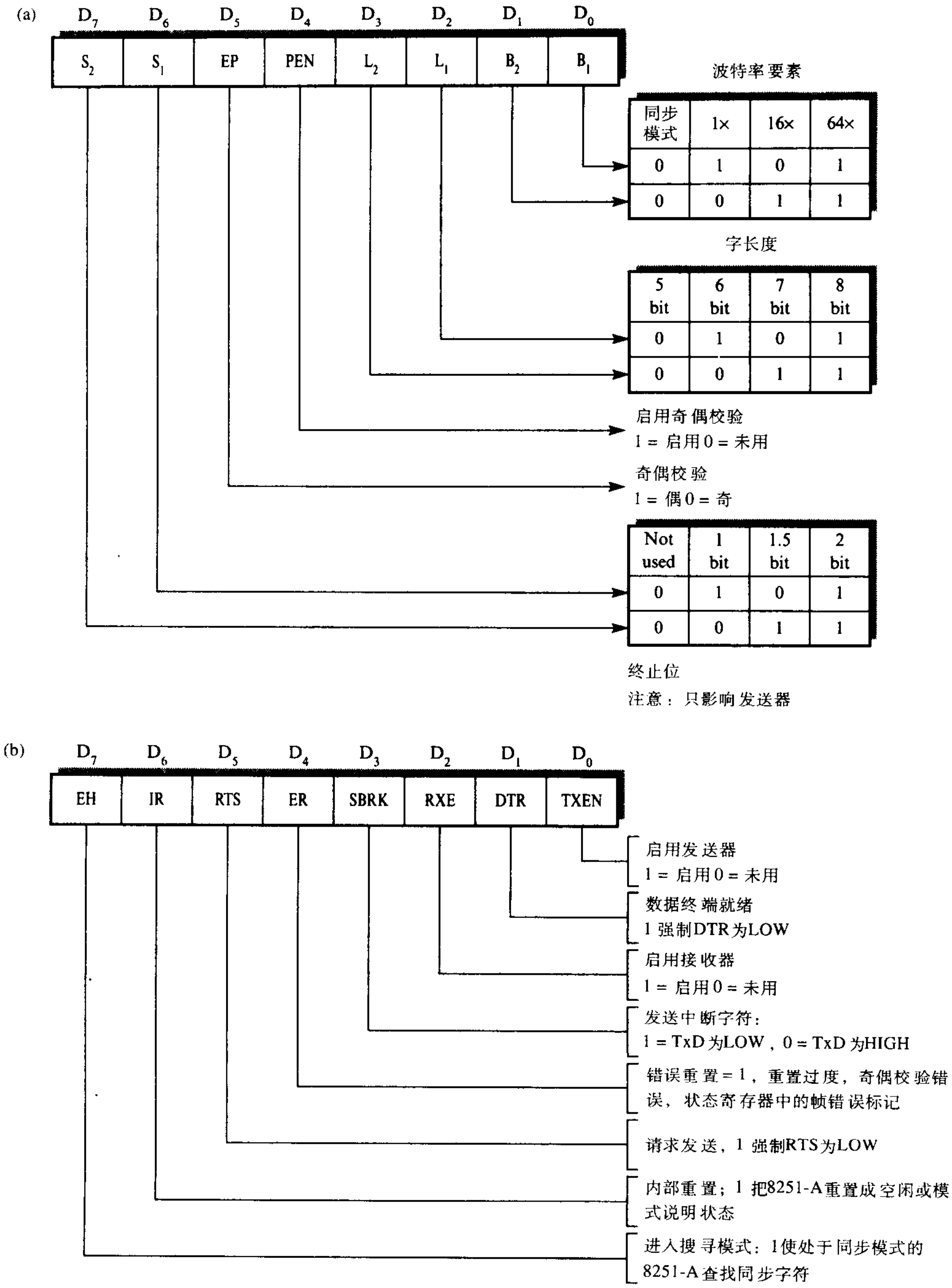


图B-1 USART示意图

(a) 主要的设备寄存器 (b) 设备接口

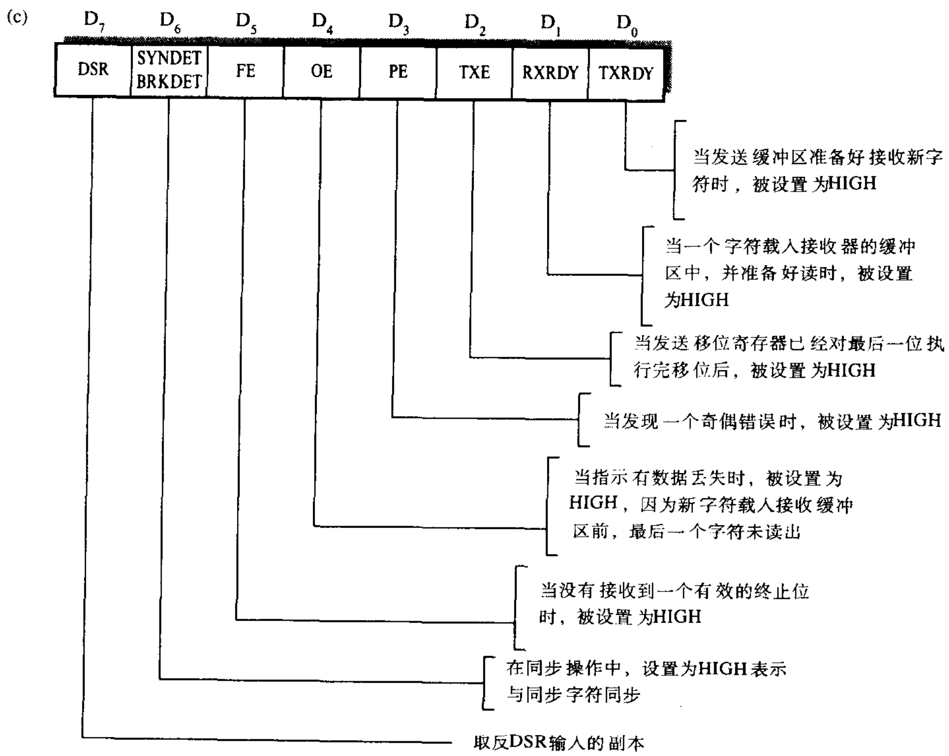
为了发送一个字符，控制设备首先要等待，直到TXRDY标记（传输就绪位）置位，它表示发送缓冲区为空。然后，字符被装入发送缓冲区，当前一字符的最后终止位被发送到总线上后，该字符被设备中的控制逻辑传递给PISO发送寄存器，这时TXEMPTY标记被置位，并在新字符装入发送寄存器时，TXEMPTY标记重置。当总线处于半双工模式时，TXEMPTY标

记非常有用，这是因为在这种模式下，控制设备必须知道信息中的最后一个字符（实际上是最后的终止位）已经发送给总线了。



图B-2　Intel 8251A 的模式/命令/状态寄存器中各个位的定义

(a) 模式　(b) 命令　(c) 状态



图B-2 （续）

当接收一个字符时，来自总线的每位被采样，并被串行地移位到SIPO接收寄存器中，当接收到字符的最后终止位，该字符被控制逻辑传递给接收缓冲区。于是，RXRDY（接收就绪）标记被置位，表示控制设备准备好读取一个字符。然后，当接收到的字符从接收缓冲区中读取出来后，UART对RXRDY标记重置。另外，在接收过程中，UART向控制设备提示如下三种错误条件的发生：

- 如果一个字符计算出的奇偶位同所选的值不同时，则PE（奇偶错）被设置。
- 如果接收到的字符在前一个字符读取之前就发送到接收缓冲区，导致前一个字符丢失时，则OE（**超时错**）被设置。
- 如果没有收到一个有效的终止位，通常表示时钟速率出现了错误，FE（帧错误）被设置。

为了使USART以同步模式工作（即可以作为一个USRT），模式寄存器中的两个最低有效位都设置为0。与异步传输方式一样，接下来四个位有相同的含义：即每个字符的位数和奇偶选择（奇校验，偶校验或无校验）。两个最高位有不同的含义，ESD（外部同步检测）位，即S1，用于指明字符同步过程要在外部执行还是在内部执行，通常设置成内部执行。而SCS（同步字符选择）位，即S2，用于指明当发送缓冲区为空时，是否发送一个或两个SYN字符，例如在发送一个帧之前。

与异步传输一样，必须在下载帧内容之前，装入命令寄存器。各个位的使用情况与异步传输中是一样的。同步传输中的重要附加位是EH位（进入搜索模式）。假如已经选择了内部同步检测，EH位用来启动设备的接收部件中的字符同步过程。 881

一旦定义了8251A的功能后（使用模式寄存器），就会装入两个或多个SYN字符（在接到帧内容之前，使接收器能够获得字符同步）。

通过读取状态寄存器的内容，并对该内容中的每一位进行解释，控制设备可以知道8251A的当前状态。这与异步传输中使用的方法是一样的。同步传输中使用的附加位是SYNDET（同步检测）位，它在搜索模式中指明：接收器部件已经实现了字符同步。通常，接下来控制设备要读取它所接收到的每个后续字符，然后等待标记新帧的帧开始字符。

对于8251A，利用芯片选择（CS），读（RD）和写（WR），以及控制/数据（C/D）总线的组合，控制设备访问各种控制寄存器和数据寄存器。各种总线的使用情况见表B-1。

表　B-1

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	操　作
0	1	1	0	写模式/命令寄存器
0	0	1	0	写发送缓冲区
0	1	0	1	读状态寄存器
0	0	0	1	读接收缓冲区
1	X	X	X	禁用设备

X = 0 或1

如同我们看到的，当控制设备同控制寄存器通信时，C/D总线被设置为1，而当控制设备同数据寄存器通信时，C/D总线被设置为0，实际使用哪个寄存器则由RD和WR总线决定。例如，如果要读状态寄存器，C/D总线要设置为1，当从接收缓冲区中读取字符时，C/D总线要设置为0。要注意，为了写入模式/命令寄存器，要执行两个连续的写操作：第一个是把字节装入模式寄存器，第二个是把它装入命令寄存器。

我们仅仅讨论了一种设备类型（USART）的操作，有其他传输控制设备都以同样的方式
882 进行操作。

附录C 标准化组织简介

美国国家标准学会（ANSI） 美国的标准化组织，成员包括美国国内的计算机制造商和用户。它还是ISO的美国成员组织，参与了ISO参考模型中各个层次的标准开发。

ATM论坛 一个世界范围的网络设备制造商组织。它为私有ATM网络制定相关标准。

英国标准学会（BSI） 英国的标准化组织，它关注各种形式的制造和消费产业的有关标准的制定。此外，它还是ISO的英国成员组织，充当所有文档的发起者。

电子工业协会（EIA） 美国国家标准化组织，其成员来自电子行业。它为数据和计算机通信环境中的外围设备同计算机之间的接口制定了一系列（物理）接口标准，最近还利用通用汽车的捐赠，积极参与了制造信息服务（MMS）七级标准的开发。此外，它还是ANSI和OSI的成员。

欧洲计算机制造商协会（ECMA） 一个由欧洲计算机制造商和一些美国公司的欧洲分公司组成的协会。它制定自己的标准，并积极推动了ITU-T和ISO的发展。它的成员都积极参与了ISO参考模型各级标准的制定。

欧洲电信标准化学会（ETSI） 一个由EU和EFTA成员国建立的欧洲标准化团体，它基于调整的目的制定相关标准。已制定了关于电信服务、公共数据网络、可视图文系统和ETS发布的数字移动服务的相应标准。

电子电气工程师协会（IEEE） 一个美国的职业团体，参与了计算机工业标准的开发。它负责计算机通信环境中关于LAN的标准的制定，以及物理层、MAC和LLC子层的相关标准的制定。

国际标准化组织（ISO） 一个由各参与国制定标准团体组成的国际标准组织。它关注很多领域的标准制定，每个标准都由一个特定的技术委员会进行控制。为计算机行业制定标准的技术委员会是TC97（即信息处理系统）。该委员会负责OSI基本ISO参考模型以及模型中各协议层标准的制定。

国际电报电话咨询委员会（CCITT） 一个由各成员国的邮政、电报、电话权威机构组成的国际标准组织。主要关注公共电信网络，包括模拟PSTN、ISDN和PPSDN的接口设备的标准的开发和制定。另外，该组织也为传真、电传，可视图文系统和其他的增值服务（电信业服务）制定相关标准。 883

美国国家标准局（NBS） 一个美国国家标准组织，主要关注ISO和ITU-T的标准制定成果。它为美国联邦政府购买的常规信息处理领域的设备发布相关标准。它被称为联邦信息处理标准（FIPS）。

OSI网络管理论坛（NMF） 一个全球范围的由电信和计算机公司、服务提供商和服务用户组成的组织。NMF为基于ISO/OSI管理协议的网络管理制定标准。 884

术 语 表

Abstract syntax notation one（ASN.1）（抽象语法表示法1） 一种用于定义特定协议实体的协议数据单元（PDU）结构的抽象语法。

Address resolution protocol（ARP）（地址解析协议） TCP/IP协议族中的协议，用于获得对应于主机跨互联网IP地址的网络连接点地址。

Advanced Research Projects Agency（ARPA）（高级研究规划局） 资助ARPANET和后来的因特网（现在的DARPA）创建的美国政府机构的名称。

American Standards Committee for Information Interchange（ASCII）（美国信息交换标准委员会） 它通常表示由该委员会定义的用于两个通信设备之间信息互换的字符代码。ASCII字符集广泛地应用于计算机和外围设备之间的信息传递中，例如一个显示器单元或打印机。

Amplitude modulation（AM）（振幅调制） 一种允许数据通过模拟网络进行传输的调制技术，例如电话交换网。单一载波频率的振幅被调制成两个电平：一个为二进制的0，另一个为二进制的1。

Application layer（应用层） 它对应于ISO参考模型中用于开放系统互连的第七层。由一组面向应用的协议组成，并且有支持各种分布式信息处理服务的用户接口。

ARPANET ARPA/DARPA资助成立的广域网，有很多美国或其他国家的大学、研究机构和国防机构连接到该网络。除了执行实际的传输，它还是研究网际互连的开发实验基地，现为因特网的一部分。

Association control service element（ACSE）（联系控制服务元素） 应用层中的一个协议实体。它提供在两个应用实体之间建立和释放一个逻辑连接（联系）的常规功能。

Asynchronous transfer mode（ATM）（异步传输模式） 宽带综合业务数字网络建议的操作模式。所有传递的信息（声音、数据、图像、视频），都首先被分割成固定长度的小帧，称为信元。然后这些信元利用包交换原则进行交换和路由选择，也称为信元或快速包交换。基于这种操作模式的第一个网络是ATM LAN。

Asynchronous transmission（异步传输） 严格地讲，它表示，当数据通过传输线路在两个互连的设备之间传递时，接收方时钟同发送方时钟是不同步的。一般，它表示传递的数据是独立的字符。每个字符由一个开始信号打头，并由一个或多个结束信号终止，这些信号被接收方用于同步的目的。

Automatic repeat request（ARQ）（自动重复请求） 一种在传输线路上进行差错控制的技术。如果接收方在传递消息中发现了错误，它会请求发送设备重新发送该消息以及其他可能被影响的消息。

Bandwidth（带宽） 它是在一个传输线路或网络上传递的最高和最低正弦信号频率之差，用赫兹（Hz）进行度量，并定义了线路或网络的最大信息负载容量。

Baseband（基带） 传输线路的一种特定操作模式：消息中的每个二进制数字都被转换成两个电平之一：一个对应二进制1，另一个对应二进制0。然后，该电压被直接应用于线路上。当数据传递时，线路信号在两个电压之间随数据的不同而改变。

Basic mode（基本模式） 一种ISO国际标准协议，用于控制主站和多个从站之间的数据交换，两个站点是通过多点链路连接的。

Baud（波特） 每秒钟线路信号的变化数。另外，当传输线路上一个信号电平表示一个比特时，它还表示数据在线路上的传输速率。此时比特率和线路信号率是相同的。

Binary synchronous control（BSC）（二进制同步控制） IBM使用的ISO基本模式协议的名称。

Bit stuffing（zero bit insertion）（位填充，零位插入） 在同步传输线路上传递二进制数据的一种技术。每个消息块（帧）被封装在两个标志中，它们是特定的位序列。如果消息数据包含相同的序列，那么发送方要在数据流中插入一个附加位（零），该位会被接收方移除。这种传输方式被称为数据透明的。

Block sum check（块校验和） 它用于传递的数据中的差错检测。它由一组二进制数字组成，是帧或消息中的字符/字节的模2和。

Bridge（网桥） 用于连接两个同构局域子网的设备，就是说两个子网使用同样的物理和介质访问控制方法。

Broadband（宽带） 同轴电缆的一种特定操作模式。通过分配每个数据流总带宽中的一部分带宽，一条同轴电缆可以同时传递若干个数据流。根据所选的频带，数据被调制成一个频率信号发送，并且通过对信号解调进行接收。

Broadcast（广播） 一种把消息传递给所有连接到网络上的设备的传输方法。通常，要为此保留一个广播地址，通过广播地址所有设备可以确定该消息是否是一个广播消息。

Bus（总线） 对分布在局部区域内的数字设备进行互连的一种拓扑结构，它被广泛使用。传输介质通常是一条同轴电缆，所有设备都连接到上面。每个传输都在整个介质上传播，并为每个连接到介质上的设备接收。

Checksum（校验和） 参见块校验和。

Circuit switching（电路交换） 电话网络和一些新的数字数据网络的操作模式。首先在源（主叫）和目标（被叫）终端之间建立一个通信路径，然后该路径只能用于该呼叫或事务的持续阶段中。两个终端必须使用相同的信息传输率进行操作。

Coaxial cable（同轴电缆） 一种由中心导体和同轴的外部导体组成的传输介质。它用于高速数据传输率的网络中（大于1Mbps）。

Commitment, concurrency, and recovery（CCR）（委托、并发和恢复） 应用层中的一个协议实体。它允许两个或多个应用进程在共享数据上执行互斥的操作。它还提供了用于确保操作被完全执行或完全不执行所需要的控制功能，使用了原子操作概念以及两阶段委托协议。

Community antenna television（CATV）（有线电视） 局域数据网中使用的一种设备，因为CATV网络中使用的原理和网络组件也被用于制造一个灵活的基本局域数据传输设备。CATV使用宽带工作模式进行操作。

Common application service element（CASE）（公共应用服务元素） 一组协议实体集，它是应用层的一部分，用于提供某些公共服务，例如在两个应用协议实体之间建立逻辑连接（联系）。

Common management information protocol（CMIP）（公共管理信息协议） ISO应用层协议，用于通过OSI网络发送和检索管理信息。

Continuous RQ（连续RQ协议） 用于差错控制的数据链路协议。如果在传递过程中帧（消息）被破坏了，它用于确保对该帧重发。为了提高数据链路的利用率，帧是被连续发送的，因此，在其他的信息帧被发送之后，才会接受对被破坏的帧的重发请求。

Crosstalk（串扰） 外部电子行为使导体拾取到非期望信号。

CSMA/CD 带冲突检测的载波侦听多路访问。一种控制对共享传输介质进行访问的方法，例如连接有多个站点的同轴电缆总线。发送消息的站点必须先侦听通道是否空闲。仅当通道空闲——无载波出现，才可以发送；否则等待。当消息发送时，站点监视传输介质上的实际信号。如果与发送信号不同，就发生并检测到一个冲突。站点立即中止当前发送，并稍后再试。

Cyclic redundancy check（CRC）（循环冗余校验） 一种用于传输数据的差错检测的方法。CRC是根据发送消息中的位计算出的数值。在消息传递之前，它被加到消息的尾部，接收方通过重新计算一个新的CRC，从而检测接收到的消息中可能出现的错误。

Data circuit terminating equipment（DCE）（数字电路端接设备） 一种网络机构（提供者）提供用来把用户设备连接到网络上的设备。对于不同的网络类型，它有不同的形式。

Data link layer（数据链路层） 它对应于开放系统互连的ISO模型中的第二层，用于保证通过数据链路传递的数据的可靠性。

Data terminal equipment（DTE）（数据终端设备） 连接到数据网络上的用户设备的通用名称，包括显示部件、计算机和办公室工作站等。

Datagram（数据报） 包交换数据网络（参见虚呼叫）提供的一种服务类型。数据报就是一种自含的信息包，它通过网络发送，并具有最小的协议开销。

Decibel（分贝） 一个信号相对于另一个信号的强度的一种测量方式。它的值相当于两个信号能量的比率的常用对数的10倍，或两个信号振幅（电压或电流）的比率的常用对数的20倍。

Defense Advanced Research Projects Agency（DARPA）（美国国防高等研究计划局） 参见ARPA。

Delay distortion（时延失真） 组成信号的频率分量通过传输介质时，由于各频率分量有不同的传播速率所造成的失真。

Directory service（DS）（目录服务） OSI协议族中应用层中的一个协议实体，主要把应用进程使用的符号名称或标题，翻译成为开放系统互连环境使用的完全限定网络地址。也被称为X.500。

Distributed queue，dual bus（DQDB）（分布式队列双总线） 一种基于光纤的网络，可以用于高速LAN或MAN，与宽带ISDN是兼容的。它以广播的模式进行操作，具有两个总线，每个总线以相反的方向传递固定大小的小帧（即信元）。每个总线都可以以每秒几百兆比特的速率进行数据传递。

Domain name system（DNS）（域名系统） TCP/IP协议族中使用的应用协议，它把人们使用的符号名称转换成等价的完全限定网络地址。

EIA-232D 美国EIA发布的标准，用于把数字设备连接到支持PTT的调制解调器，也作为连接外围设备的接口标准，例如显示器或打印机到计算机的连接。

Error rate（数字误码率） 数据链路或系统中出错位数与发送总位数之比。

Ethernet（以太网） 美国Xerox公司的Palo Alto研究中心开发的一种LAN名称。它使用

CSMA/CD介质访问控制方式进行操作。早期的规范由Xerox、DEC和Intel三家公司联合制定，现在由IEEE 802.3（ISO 8802.3）国际标准替代。

Extended binary coded decimal interchange code（EBCDIC）（扩充的二—十进制交换码）IBM计算机上使用的字符集。

Exterior gateway protocol（EGP）（外部网关协议） 由多个小互联网互连组成的大互联网中使用的一种协议。互连设备被称为外部网关，而EGP是外部网关用于通知各个小互联网中网络IP地址的协议。

Fast Ethernet（快速以太网） 用于描述高速100Mbps LAN的术语，它与10Mbps CSMA/CD以太网兼容。

Fast select（快速选择） X.25协议中的一个可选项，允许将用户数据包含在呼叫建立包或呼叫清除包中发送，从而减少了路径选择时间。

Fiber distributed data interface（FDDI）（光纤分布式数据接口） 一种用于高速LAN或MAN的光纤环网。它提供高达100Mbps的用户比特率，并使用控制令牌的介质访问方法。

Fiber optic（光纤） 参见optical fiber。

File transfer access and management（FTAM）（文件传输访问和管理） 应用层中的一个协议实体。它使用户应用进程可以管理和访问（分布式）文件系统。

File transfer protocol（FTP）（文件传输协议） TCP/IP协议族中的一个应用协议，提供对网络文件服务器的访问。

Flow control（流量控制） 一种控制两个通信实体之间帧或报文流的速率的技术。

Frame（帧） 在数据链路上传递的信息单元。通常有用于链路管理的控制帧和传递报文数据的信息帧。

Frame check sequence（FCS）（帧校验序列） 发送方添加到传输帧上的附加字段，它使接收方可以检测可能的传输错误。

Frequency-division multiplexing（FDM）（频分多路复用） 把单一传输介质分成若干个独立数据信道的技术，例如同轴电缆。每个数据信道分配有效带宽中的一部分。

Frequency-shift keying（FSK）（频移键控） 把二进制数据转换成由两个正弦频率组成的模拟信号的调制技术。它广泛使用在调制解调器中，允许所传递的数据通过（模拟）交换电话网络。

Full-duplex（全双工） 两个通信设备之间的信息交换策略，信息可以在两个方向上同时交换，也被称为双向并发。

Gateway（网关） 在两个网络之间路由数据报的设备。通常，两个网络使用不同的协议，所以网关要执行必要的协议转换功能。

Half-duplex（半双工） 两个通信设备之间的一种信息交换策略，信息（数据）可以在两个方向上交替传递，也被称为双向交替。

High-level data link control（HDLC）（高级数据链路控制） 一种国际标准协议，用于控制点到点数据链路或多点数据链路上的数据交换。

Host（主机）通常是一台属于用户的计算机，包含与数据通信网建立连接所必须的硬件和软件。

Idle RQ（空闲RQ协议） 具有差错控制的数据链路协议的一部分。如果一个帧（报文）在传递过程中破坏了，它确保对该报文重发。当发送方发送完一个帧之后，会等待直到收到正

确接受的指示，或者到达一定时间后发送另一帧，也被称为发送（或停止）并等待。

Integrated services digital network（ISDN）（综合业务数字网） 全球电信网络的新一代，利用数字技术进行传输和交换，它支持语音和数字通信。

Interior gateway protocol（IGP）（内部网关协议） TCP/IP互联网网关中使用路由协议，可以获得通过互联网的最短路径。

Intermediate system（IS）（中间系统） ISO用来描述实现两个网络互连的设备的术语，也被称为路由器或网关。

International alphabet number 5（IA5）（国际字符编号5） ITU-T定义的标准字符码，也是ISO的推荐标准。它同ASCII码是相同的。

Internet（因特网） 互联网络集合的缩写名称，也是基于TCP/IP协议族并由美国政府资助的互联网的名称。

Internet control message protocol（ICMP）（Internet控制报文协议） TCP/IP协议族中的一个互联网协议，负责处理由互联网主机和网关返回的差错和控制报文。

Internet protocol（IP）（网际互联协议） TCP/IP协议族中用来提供通过网关互连的多个包交换网络之间的无连接网络服务。

Job transfer and manipulation（JTM）（作业传送和处理） 应用层中的一个协议实体，它使用户应用进程能够对与作业（处理任务）相关的文档进行传输和处理。

Local area network（LAN）（局域网） 互连一个局部区域中分布的数字设备的数据通信网络，通常在10平方公里范围以内，可以包含工作站、小型机、微机或智能仪器设备等。

Logical link control（LLC）（逻辑链路控制） LAN数据链路层的一个协议，用于在两个通信系统之间的数据链路上实现可靠的数据传输。

Management information base（MIB）（管理信息库） 保存有关网络或网络互连的管理信息的数据库的名称。

Manchester encoding（曼彻斯特编码） 在传递数据之前，把时钟信息加入二进制数据流中的编码方法。得到的编码信号在每个位信元中心都有突变（正的或负的），所以时钟信息可以很容易地从接收信号中提取出来。

Manufacturing message service（MMS）（生产消息服务） 应用层中的一个协议实体，专门用于制造或处理控制工业中。它使监管计算机能够控制分布式基于计算机的设备的操作。

Medium access control（MAC）（介质访问控制） 很多LAN使用了单一公共传输介质，例如总线或环，而其他所有互连设备都是连接到该介质上的。每个设备都要遵循规程，来确保传输是以有序和公平的方式进行的，通常被称为介质访问控制规程，CSMA/CD和（控制）令牌就是两个例子。

Message handling service（MHS）（电文处理服务） 应用层中的一个协议实体，提供了常规的、在两个系统之间交换电子邮件的功能，也被称为X.400。

Metropolitan area network（MAN）（城域网） 连接分布在一个城镇范围中的LAN的网络。

Microwave（微波） 利用发送和接收天线（抛物面天线）基于电磁辐射的通信类型，可以用于地面链路或卫星链路。

Modem（调制解调器） 在通过模拟网络发送数据之前，把二进制（数字）数据流转换成模拟（连续可变）形式的设备（调制器），接收方再把接收到信号转换成二进制形式（解调器）。因为通常每个网络端口都使用全双工模式，所以该设备必须既执行调制功能又执行解调功

能，因此采用调制解调器的名称。例如，电话网络通常也需要用调制解调器传递数据。

Multidrop（多点） 在同一个传输介质上支持多于两个站点的网络配置。

Multiplexer（多路复用器） 使同一地点的若干个低比特率设备，共享同一条高比特率传输线路，而线路的数据携带能力必须超过各个低比特率设备的比特率和。

Multipoint（多点） 参见multidrop。

Network layer（网络层） 开放系统互连ISO参考模型中的第三层，用于网络上逻辑或者物理连接的建立和撤销。

Network management（网络管理） 网络管理中涉及的所有功能和实体的通用术语，包括配置管理、故障处理和收集网络使用情况的统计信息等。

Noise（噪声） 传输线路上产生或拾取的外部电信号。通常，它可能是由一个邻近的电子设备引起的。如果噪声信号超过了数据信号，那么就会使数据信号被破坏，引发传输错误。

NRZ/NRZI 两种相似的二进制数据流编码方法。第一个具有每当二进制数据流出现1时，信号发生跳变的属性；第二个具有每当二进制数据流出现0时，信号发生跳变的属性。后者用于特定时钟方法。

Open system（开放系统） 一组与厂商无关的互连计算机，并且都使用相同的标准通信协议栈，或者基于ISO/OSI或者基于TCP/IP。

Open systems interconnection（OSI）（开放系统互连） 为了创建一个开放的系统互连环境，基于国际标准化组织（ISO）协议的协议族。

Optical fiber（光纤） 一种传输介质类型，在其中数据是以光波或光脉冲的形式传播的，具有高带宽、数据携带能力强和对其他电子源干扰高的特性。

Packet assembler/disassembler（PAD）（包拆装器） X.25包交换网络中使用的一种设备，允许字符方式的终端同包方式的设备进行通信，例如计算机。

Packet switching（分组交换） 数据通信网络中的一种操作方式。把通过网络传输的消息，首先分成若干个较短的自包含消息单元，这种单元称为分组（包）。每个包都包含了寻址信息。当网络的中间结点接收到每个包，首先把包存储起来，然后依照包中携带的寻址信息，选择适当的路径把包转发给下一个结点，等等。属于同一个消息的包在目标设备上重新装配。这种操作方式确保长的消息不会降低网络的响应时间，并且源和目标设备可以以不同的数据速率操作。

Parity（奇偶校验） 当传输单字符时，用于检测传输差错的一种方法。称为奇偶位的二进制数，它的值由字符中二进制1的个数决定，奇偶位同该字符一同传递，接收方对奇偶位重新计算并进行比较，从而检测出字符中的单位差错。

Phase-shift keying（PSK）（相移键控） 把二进制数据转换成模拟形式的一种调制技术。模拟信号是由正弦频率信号组成的，正弦信号的相位由所传递的二进制数据的值确定。

Physical layer（物理层） 开放系统互连ISO参考模型中的第一层，关注于物理网络终端设备的电子和机械规范。

Piggyback（捎带） 在全双工数据链路上不使用专门确认帧返回确认信息的技术。关于某个方向上消息流的确认信息被携带在相反方向上的数据帧中返回。

Postal, Telegraph, and Telephone（PTT）（邮电管理机构） 管理一个国家中所有邮政和公共电信网络与服务的管理机构。

Presentation layer（表示层） 开放系统互连ISO参考模型中的第六层，关心应用会话中使用的

传送语法的协商，如果语法不同于本地语法，还要负责两种语法的转换。

Protocol（协议） 控制两个通信部分之间的数据交换的一组规则。

Protocol data unit（PDU）（协议数据单元） 两个协议实体之间交换的报文单元。

Protocol entity（协议实体） 控制协议层操作的代码。

Public switched data network（PSDN）（公共数据交换网） 由公共电信机构为了数据交换目的而建立的一种通信网络。

Public switched telephone network（PSTN）（公共电话交换网） （模拟）电话网络。

Remote operations service element（ROSE）（远程操作服务元素） 组成应用层的一个协议实体，提供了初始化和远程控制操作的功能。

Ring（环） 一种网络拓扑结构，广泛用于分布在局部区域中的数字设备的互连，如工厂或办公室大楼。每个设备都连接到最近的设备上，直到所有的设备都连接成一种闭环的形式。环上数据的传输只在一个方向上进行，当报文在环路上传递时，每个连接到环路上的设备都读取该报文。当在环路上循环之后，源设备把该报文从环路上移去。

Router（路由器） 用于把两个或多个LAN互连起来的设备，各个LAN可以使用不同的介质访问控制方法进行操作，也被称为网关或中间系统。

RS-422/RS-423 由美国EIA发布的标准，用于把数字设备连接到支持PTT的调制解调器上。

Send and wait（发送并等待） 参见空闲RQ协议。

Service access point（SAP）（服务访问点） 一个特定系统中两个协议层之间用来惟一识别特定链路的子地址。

Session layer（会话层） 开放系统互连ISO参考模型中的第五层，用于在两个应用实体之间建立逻辑连接，并对两者间的对话（交换报文）进行控制。

Shortest-path first（SPF）（最短路径优先） 一种在网关/路由器/中间系统中使用的算法，用来找出它与互联网中其他网关之间的最短路径。

Signal-to-noise ratio（信噪比） 线路或系统中的信号功率与噪声功率的比率，通常用分贝表示。

Simple mail transfer protocol（SMTP）（简单邮件传输协议） TCP/IP协议族中的一种应用协议，用来在一组互连的电子邮件系统之间传递电子邮件。

Simple network management protocol（SNMP）（简单网络管理协议） TCP/IP协议族中的一种应用协议，用来在TCP/IP网络上发送和检索管理信息。

Simplex（单工） 两个通信设备之间的一种信息交换策略，在其中信息（数据）只能在一个方向上传递。

Slotted ring（有槽环） 一种局域（数据）网络，所有设备都连接到环路上，并使用一个附加的监视设备确保环路上具有固定数量的报文槽，这些槽在一个方向上绕着环路循环。某个设备把报文放在空槽中进行发送，所有的设备都读取该报文，然后初始的设备把该报文移去。

Specific application service element（SASE）（特定应用服务元素） 应用层中的一组协议实体集，用于提供各种特定的应用服务，例如文件传输或作业传输。

Star（星） 一种网络拓扑结构，中心结点执行所有的报文交换（路由选择）功能。

Statistical multiplexer（stat mux）（统计多路复用器） 使同一地点内的若干个低比特率设备，共享同一条高比特率传输线路。该设备通常需要人工操作员，在共享线路上传递的数据是

基于统计信息的，而不是像常规多路复用器那样基于预先分配。每个设备要以比最大速率低的平均速率操作。

Subnet（子网） 在ISO文档中，它表示组成一个较大的互联网的独立网络。

Synchronous transmission（同步传输） 通过传输线路连接的两个设备之间传输数据的一种技术。数据通常是以块的形式传递的，每个块包含一串二进制数字。发送方和接收方时钟是同步的，有多种技术用于确保这一点。

TCP/IP 包括IP、TCP和相关应用协议的一组完整协议。

Teletex（智能用户电报） 国际电信服务，提供了准备发送和接收由文本和图形字符组成的报文时使用的方式。

TELNET（远程终端） 一种TCP/IP协议族中的应用协议，使终端上的用户能够与运行在远端计算机上的程序交互。

Time-division multiplexing（TDM）（时分复用） 共享传输设备上一种共享带宽（信道容量）的技术，它允许若干个通信同时进行，而同一时刻只执行一个通信。

Token bus（令牌总线） 一种局域（数据）网络，以总线的方式访问共享传输介质，所有的通信设备都连接到该总线上，利用单一的控制（权限）令牌进行控制。只有当前的令牌拥有者才能在介质上发送报文。所有期望发送报文的设备连接成一个逻辑环的形式。当一个设备接收到令牌之后，它会发送等待的报文，然后把令牌传递给环中下一个等待发送的设备。

Token ring（令牌环） 一种局域（数据）网络，所有的设备连接成一种（物理）环的形式，所传递的报文在环路上循环。只有当一个设备拥有控制（权限）令牌时，它才能传输报文。令牌沿着环路从一个设备传到另一个设备。

TP4 OSI协议族中的第四类传输协议，包括差错控制和流量控制功能，用于无连接网络/互联网。

Transmission control protocol（TCP）（传输控制协议） TCP/IP协议族中的协议，为应用协议提供了可靠的全双工报文传递服务。

Transmission medium（传输介质） 连接两个通信设备的通信路径，例如双绞线、同轴电缆、光纤和微波束。

Transport layer（传输层） 开放系统互连ISO参考模型中的第四层，主要为面向应用的各层提供网络无关的和可靠的消息交换服务。

Twisted pair（双绞线） 一种传输介质，它由两个独立的线绞合而成，提高了抵抗其他电信号干扰的能力，这些干扰可能会破坏所传递的数据。

User datagram protocol（UDP）（用户数据报协议） TCP/IP协议族中的无连接（最佳尝试）传输层协议。

Videotex（可视图文） 一种电信服务，允许用户对一个中心数据库设备上的信息进行存放或访问。访问是通过一个特殊的终端进行的，该终端包含一个带有特定解码器的TV。

Virtual call（circuit）（虚呼叫） 分组交换数据网络（参见数据报）上提供的一种服务类型。使用这种服务，有关特定呼叫（消息传输）的信息包发送之前，先在源和目标之间建立一个虚电路。该呼叫中所有携带信息的包，都通过同一路由进行传递，并且网络会确保这些包都是按它们输入的顺序进行传递的。

Virtual terminal（虚终端） 应用层中的一种协议实体。它允许应用进程同远程终端以标准的

方式进行对话，而无需考虑远程终端的组成。

V.24/V.35　ITU-T发布的标准，用于对数字设备同支持PTT的调制解调器进行连接。V.24还被用作连接外围设备的接口标准，例如显示器或打印机到计算机的接口。

Wide area network（WAN）（广域网）　覆盖了一个广阔的地理范围的任何形式的网络，可以是专用或公用。

Wireless LAN（无线LAN）　使用无线电波或红外线作为传输介质的LAN。它所使用的MAC方法与有线LAN中使用的MAC方法是不同的。

X.3/X.28/X.29　一组国际标准协议，它允许面向字符的设备，例如虚拟显示终端，连接到一个分组交换数据网络上。

X.25　一个国际标准协议，它用于数据终端设备（例如计算机）同分组交换数据网络的连接。

X.400　参见消息处理服务（MHS）。

X.500　参见目录服务（DS）。1

Zero bit insertion（零位插入）　参见位填充。

参考文献

下面两本书的内容同本书中所讲的内容在范围是近似的：

Stallings W.(1991).*Data and Computer Communications* 3rd edn. Macmillan

Tanenbaum A.S.(1988).*Computer Networks* 2nd edn. Prentice-Hall

第一本书偏向于电子工程方面，第二本书则偏向于计算机科学方面。另外，我们还建议读者有选择地阅读以下这些书：

第1章

Cargill C. (1989). *Information Technology Standardization: Theory, Process and Organizations.* Bedford MA: Digital Press

Cerf V. and Cain E. (1983). The DOD architecture model. *Computer Networks*, (October)

Clarke D. (1988). The design philosophy of the DARPA Internet protocols. In *Proc. SIGCOM 88 Symposium*

Day J.D. and Zimmermann H. (1983). The OSI reference model. In *Proc. IEEE*, **71**, 1334–40

Folts H. (1981). Coming of age: A long awaited standard for heterogenous networks. *Data Communications*

Green P. (1980). An introduction to network architectures and protocols. *IEEE Transactions on Communications*, (April)

ISO (1984). *Basic Reference Model for Open Systems Interconnection* (ISO 7498)

Vormax M. (1980). Controlling the mushrooming communications net. *Data Communications*, (June)

Walker S. (1982). Department of Defense Data Network. *Signal*, (October)

Wood D. (1985). Computer networks: a survey. In *Computer Communications* Vol. II, Englewood Cliffs NJ: Prentice-Hall

第2章

Ash J. and Richards D. (1994). Data over analog systems. In *Data Communications and Networks 3*, IEE London, pp. 6–24

Bachmann L. (1983). Statistical multiplexers gain sophistication and status. *Mini Micro Systems*, (March)

Bertine H.U. (1980). Physical level protocol. *IEEE Transactions on Communications*, **28**(4), 433–44

Bleazard G.B. (1982). *Handbook of Data Communications.* NCC Publications

Chou W. (1983). *Computer Communications* Vol. I. Englewood Cliffs NJ: Prentice-Hall

Cooper E. (1984). *Broadband Network Technology.* Mountain View CA: Sytek Press

Davies D.W. and Barber D.L.A. (1973). *Communication Networks for Computers.* Wiley

EIA (1987). *EIA-232D Standard Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*

Finnie G. (1989). VSATs: a technical update. *Telecommunications*, (February)

Freeman R. (1981). *Telecommunication Transmission Handbook.* Wiley

Jennings F. (1986). *Practical Data Communications.* Blackwell

McClelland F.M. (1983). Services and protocols of the physical layer. In *Proc. IEEE*, **71**, 1372–7

Mehravari N. (1984). TDMA in a random access environment: an overview. *IEEE Communications Magazine*, **22**, 54–9

Mok A.K. and Ward S.A. (1979). Distributed broadcast channel access. *Computer Networks*, **3**, 327–35

Murano K. *et al.* (1990). Echo cancellation and applications. *IEEE Communication Magazine*, (January)

Nielson D. (1985). Packet radio: an area coverage digital radio network. In *Computer Communications* Vol. II. Englewood Cliffs NJ: Prentice-Hall

Oetting J. (1979). A comparison of modulation techniques for digital radio. *IEEE Transactions on Communications*, (December)

Pearson J.E. (1992). *Basic Communication Theory.* Englewood Cliffs NJ: Prentice-Hall

Roberts L. (1973). Dynamic allocation of satellite capacity through packet reservation. In *Proceedings NCC*, pp. 711–16

Schwartz M. (1989). *Telecommunication Networks.* Reading MA: Addison-Wesley

Sklar B. (1988). *Digital Communications: Fundamentals and Applications.* Prentice-Hall

第3章

Bleazard G.B. (1982). *Handbook of Data Communications*. NCC Publications

Fletcher J. (1982). An arithmetic checksum for serial transmissions. *IEEE Transactions on Communications*, (January)

Jennings F. (1986). *Practical Data Communications*. Blackwell

McNamara J.E. (1982). *Technical Aspects of Data Communication*. Digital Press

Peterson W.W. (1981). *Error Correcting Codes*. MIT Press

Ramabadroan T. and Gaitonde S. (1988). A tutorial on CRC computations. *IEEE Micro*, (August)

Spragins J.D. *et al.* (1981). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

Storer J.A. (1988). *Data Compression: Methods and Theory*. Computer Science Press

Vitter J.S. (1987). Design and analysis of dynamic Huffman codes. *Journal of the ACM*, (October)

Welch T.A. (1984). A technique for high performance data compression. *IEEE Computer*, (June)

Witten I.H. *et al.* (1987). Arithmetic coding for data compression. *Comm. ACM*, **30**, (June) 520–40

第4章

Bleazard G.B. (1982). *Handbook of Data Communications*. NCC Publications

Budkowski S. and Dembinski P. (1988). An introduction to Estelle. *Computer Networks and ISDN Systems*, **14**, (January)

Choi T.Y. (1985). Formal techniques for the specification, verification and construction of communication protocols. *IEEE Communications Magazine*, **23**, (January) 46–52

Chou W. (1983). *Computer Communications, Vol. I: Principles*. Prentice-Hall

Conrad J. (1980). Character-oriented data link control protocols. *IEEE Transactions on Communications*, (April)

Conrad J. (1983). Services and protocols of the data link layer. *Proc. IEEE*

Danthine A.A.S. (1970). Protocol representation with finite-state models. *IEEE Transactions on Communications*. **COM28**, (April) 632–43

Davies D.W. *et al.* (1979). *Computer Networks and their Protocols*. Wiley

Pouzin L. and Zimmermann H. (1978). A tutorial on protocols. *Proc. IEEE*, (November)

Schwartz M. (1987). *Telecommunication Networks: Protocols, Modeling and Analysis*. Reading MA: Addison-Wesley

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

Vissers C.A. *et al.* (1983). Formal description techniques. *Proc. IEEE*, **71**, (December) 1356–64

第5章

Black U. (1982). Data link controls: the great variety calls for wise and careful choices. *Data Communications*, (June)

Black U. (1989). *Data Networks: Concepts, Theory and Practice*. Prentice-Hall

Bleazard G.B. (1982). *Handbook of Data Communications*. NCC Publications

Brodd W. (1983). HDLC, ADCCP and SDLC: what's the difference. *Data Communications*, (August)

Brodd W. and Boudrow P. (1983). Operational characteristics: BSC versus SDLC. *Data Communications*, (October)

Carlson D.E. (1980). Bit-oriented data link control procedures. *IEEE Transactions on Communications*, (April)

Field J. (1986). Logical link control. *IEEE Infocom 86*, (April)

Held G. (1983). Strategies and concepts for linking today's personal computers. *Data Communications*, (May)

IEEE (1985). *Logical Link Control* (ANSI/IEEE Std. 802.2). IEEE

Jennings F. (1986). *Practical Data Communications*. Blackwell

Schwartz M. (1987). *Telecommunication Networks: Protocols, Modeling and Analysis*. Reading MA: Addison-Wesley

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

第6章

Black U. (1987). *Computer Networks: Protocols, Standards and Interfaces*. Prentice-Hall

Bux W. *et al.* (1983). Architecture and design of a reliable token-ring network. *IEEE Journal on Selected Areas in Communications*, (November)

Chlamtac I. and Fanta W.R. (1980). Message-based priority access to local networks. *Computer Communications*, (April)

Chou W. (1983). *Computer Communications, Vol. I: Principles*. Prentice-Hall

Dixon R. *et al.* (1983). A token ring network for local data communications. *IBM Systems Journal*, (1 and 2)

Fine M. and Tobagi F. (1984). Demand assignment multiple-access schemes in broadcast bus local area networks. *IEEE Transactions on Computers*, (December)

Finley M. (1984). Optical fibres in local area networks. *IEEE Communications Magazine*, (August)

Golomb S.W. and Scholtz R.A. (1965). Generalized Barker sequences. *IEEE Transactions on Information Theory*, **11**(4), 533–7

Halls G.A. (1994). HiperLAN: the high performance radio local area network standard. *IEE Electronics and Communications Engineering Journal*, **6**(6), 289–96

Hammond J. (1986). *Performance Analysis of Local Computer Networks*. Reading MA: Addison-Wesley

Heyman D.P. (1982). An analysis of the carrier-sense multiple-access protocol. *Bell System Technical Journal*, (October)

Heywood P. (1981). The Cambridge ring is still making the rounds. *Data Communications*, (July)

Hopper A. *et al.* (1986). *Local Area Network Design*. Wokingham: Addison-Wesley

IEEE (1985). *802.3 CSMA/CD Access Method and Physical Layer Specifications*. IEEE

IEEE (1985). *802.4 Token-passing Bus Access Method*. IEEE

IEEE (1985). *802.5 Token Ring Access Method and Physical Layer Specifications*. IEEE

IEEE (1985). *802.2 Logical Link Control*. IEEE

Kahn J.M. *et al.* (1994). Non-directed infrared links for high capacity wireless LANs. *IEEE Personal Communications*, **1**(2), 12–25

Schwartz M. (1987). *Telecommunication Networks: Modeling and Analysis*. Reading MA: Addison-Wesley

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

Stallings W. (1987). *Local Networks – an Introduction*. 3rd edn. New York NY: Macmillan

Stallings W. (1990). *Handbook of Computer Communication Standards Vol. 2: Local Area Network Standards*. Sams

Stuck B.W. (1983). Calculating the maximum throughput rate in local area networks. *IEEE Computer*, **16**, (May) 72–6

Tsui T.S. and Clarkson T.G. (1994). Spread spectrum communication techniques. *IEE Journal Electronics and Communications Engineering*, **6**(1), 3–12

第7章

Backes F. (1988). Transparent bridges for interconnection of IEEE 802 LANs. *IEEE Network*, (January)

Bederman S. (1986). Source routing. *Data Communications*, (February)

Bux W. *et al.* (1987). Interconnection of local area networks. *IEEE Journal on Selected Areas in Communications*, special issue (December)

Caves K. (1994). Second generation LANs and MANs. In *Data Communications and Networks 3*, IEE London, pp. 149–83

Dixon R. and Pitt D. (1988). Addressing, bridging and source routing. *IEEE Network*, (January)

Hamner M. and Samsen G. (1988). Source routing bridge implementation. *IEE Network*, (January)

Hart J. (1988). Extending the IEEE 802.1 bridge standard to remote bridges. *IEEE Network*, (January)

Hewlett Packard (1994). *IEEE 802.12: Demand Priority Access Method and Physical Layer Specifications*. Draft standard

IEEE (1988). *802.1 D, MAC Bridges*. IEEE

IEEE (1988). *802.5 Appendix D, Multiring Networks (Source Routing)*. IEEE

Johnson M. (1987). Proof that timing requirements of the FDDI token ring protocol are satisfied. *IEEE Transactions on Communications*, (June)

Joshi S.P. (1986). High-performance networks – a focus on the FDDI standard. *IEEE Micro*, **6**, (June) 8–14

Karvelas D. and Papamichail M. (1992). DQDB: a fast converging bandwidth balancing mechanism that requires no bandwidth LAN. In *IEEE ICC 92 Conference Proceedings*, pp. 142–6

Kummerle K. (1987). *Advances in Local Area Networks*. New York: IEEE Press

Limb J.O. (1984). Performance of local area networks at high speed. *IEEE Communications*, **22**, (August) 41–5

Pitt D.A. (1988). Bridging – the double standard. *IEEE Network*, **2**, (January) 94–5

Ross F.E. (1986). FDDI – a tutorial. *IEEE Communications*, **24**, (May) 10–15

Ross F.E. (1989). An overview of FDDI – the fiber distributed data interface. *IEEE Journal on Selected Areas of Communications*, (September)

Seifert W.M. (1988). Bridges and routers. *IEEE Network Magazine*, **2**, (January) 57–64

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

St Clair J. (1994). *100 Base–T Specification*. Fast Ethernet Alliance

Stallings W. (1990). *Local Networks* 3rd edn. New York NY: Macmillan

Strole N. (1983). A local communication based on interconnected token access rings: a tutorial. *IBM Journal of Research and Development*, (September)

第8章

Black U. (1987). *Computer Networks: Protocols, Standards and Interfaces*. Prentice-Hall

Black U. (1989). *Data Networks: Concepts, Theory and Practice*. Prentice-Hall

Bleazard G.B. (1982). *Handbook of Data Communications*. NCC Publications

Burg F. (1983). Design considerations for using the X.25 packet layer on data terminal equipment. In *Proceedings IEEE Infocom 83*

Bush J. (1989). Frame-relay services promise WAN bandwidth on demand. *Data Communications*, (July)

Deasington R.J. (1988). *X.25 Explained: Protocols for Packet Switched Networks* 2nd edn. Ellis Horwood

Decina M. (1986). CCITT recommendations on the ISDN: a review. *IEEE Journal on Selected Areas of Communications*, **SAC.4**, (May) 320–5

Dhas C.R. and Konangu V.K. (1986). X.25: an interface to public packet networks. *IEEE Communications*, **24**, (September) 118–25

Duc N. and Chew E. (1985). ISDN protocol architecture. *IEEE Communications*, (March)

Gerla M. and Kleinrock L. (1980). Flow control: a comparative survey. *IEEE Transactions on Communications*, (April)

Irland M.I. (1978). Buffer management in a packet switch. *IEEE Transactions on Communications*, **COM.26**, (March) 328–37

Kostas D. (1984). Transition to ISDN – an overview. *IEEE Communications*, (January)

Lai W. (1989). Frame relaying service: an overview. *Proceedings IEEE Infocom 89*, (April)

Land J. (1987). *The Integrated Services Digital Network (ISDN)*. NCC Publications

Schwartz M. (1987). *Telecommunications Networks: Modeling and Analysis*. Reading MA: Addison-Wesley

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

Stallings W. (1989). *ISDN: An Introduction*. New York NY: Macmillan

第9章

Bell P. and Jabbour K. (1986). Review of point-to-point network routing algorithms. *IEEE Communications Magazine*, (January)

Boule R. and Moy J. (1989). Inside routers: a technology guide for network builders. *Data Communications*, (September)

Burg F. and Iorio N. (1989). Networking of networks: interworking according to OSI. *IEEE Journal on Selected Areas of Communications*, (September)

Comer D.E. (1991). *Internetworking with TCP/IP, Volume 1* 2nd edn. Prentice-Hall

DARPA (1981). *Internet Control Message Protocol* (RFC 792)

DARPA (1983). *Internet Protocol* (RFC 791)

Gopal I. (1985). Prevention of store-and-forward deadlock in computer networks. *IEEE Transactions on Communications*, (December)

ISO (1988). *Connectionless-mode Network Service (Internetwork Protocol)* (ISO 8473).

Markley R.W. (1990). *Data Communications and Interoperability*. Prentice-Hall

McConnell J. (1988). *Internetworking Computer Systems*. Prentice-Hall

McQuillan J. *et al.* (1980). The new routing algorithm for the ARPANET. *IEEE Transactions on Communications*, (May)

Moy J. (1989). *The OSPF Specification, RFC 1131 DDN Network Information Centre*. Menlo Park CA: SRI International

Moy J. and Chiappa N. (1989). OSPF: a new dynamic routing standard. *Network World*, (August)

Parulkar G. (1990). The next generation of internetworking. *Computer Communications Review*, (January)

Piscitello D. *et al.* (1986). Internetworking in an OSI environment. *Data Communications*, (May)

Sheltzer A. *et al.* (1982). Connecting different types of networks with gateways. *Data Communications*, (August)

Schoch J.F. (1978). Internetwork naming, addressing and routing. In *Proceedings COMPCON 78*

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

Weissberger A.J. and Israel J.E. (1987). What the new internetworking standards provide. *Data Communications*, (February)

第10章

ATM Forum (1994). *ATM User/Network Interface Specification*, (July)

Caves K. (1994). Second generation LANs and MANs. In *Data Communications and Networks 3*, IEE London, pp. 149–83

Caves K. (1994). Wide area networking. In *Data Communications and Networks 3*, IEE London, pp. 118–48

Cuthbert L.G. and Sapanel J.-C. (1993). *ATM – The Broadband Telecommunications Solution*. London: The Institution of Electrical Engineers

Gerla M. *et al.* (1993). Internetting LANs and MANs to B-ISDN for connectionless traffic support. *IEEE JSAC*, **11**(8), (October) 1145–59

Händel R., Huber M.N. and Schroder S. (1994). *ATM Networks: Concepts, Protocols, Applications* 2nd Edition. Wokingham: Addison-Wesley

Heinanen J. (1993). *Multiprotocol Encapsulation Over ATM Adaptation Layer 5*. RFC 1483

ITU-TS (1993). *B-ISDN ATM Adaptation Layer (AAL)*

Specificiation. Draft recommendation I.363

ITU-TS (1993) *Support of Broadband Connectionless Data Service on B-ISDN*. Draft recommendation I.364

Karak N. (1995). Data communication in ATM networks. *IEEE Network*, (May/June) 28–37

Lanbach M. (1994). *Classical IP and ARP over ATM*. RFC 1577

Le Boudec J.-L. (1992). The asynchronous transfer mode: a tutorial. In *Computer Networks and ISDN System 24*, North Holland, pp. 279–309

McDysan D.E. and Spohn D.L. (1995). *ATM Theory and Application*. McGraw-Hill

Newman P. (1994). ATM local area networks. *IEEE Communicatons Magazine*, (March) 86–98

Sutherland S.L. and Burgin J. (1993). B-ISDN interworking. *IEEE Communications Magazine*, (August) 60–3

Truong H.L. *et al.* (1995). LAN emulation on an ATM network. *IEEE Communicatons Magazine*, (May) 70–85

第11章

Black U. (1987). *Computer Networks: Protocols, Standards and Interfaces*. Prentice-Hall

Black U. (1989). *Data Networks: Concepts, Theory and Practice*. Prentice-Hall

Cockburn A. (1987). Efficient implementation of the OSI transport protocol checksum algorithm using 8/16-bit arithmetic. *Computer Communications Review*, (July)

Comer D.E. (1991). *Internetworking with TCP/IP, Volume 1* 2nd edn. Prentice-Hall

DARPA (1983). *Transmission Control Protocol* (RFC 793)

DARPA (1983). *User Datagram Protocol* (RFC 768)

Davidson J. (1988). *An Introduction to TCP/IP*. New York: Springer Verlag

Groenback I. (1986). Conversion between the TCP and ISO transport protocols as a method of achieving interoperability between data communication systems. *IEEE Journal on Selected Areas in Communications*, (March)

ISO (1985). *Connection-oriented Transport Service and Protocol* (ISO 8072/3)

Karn P. and Partridge C. (1987). Improving round-trip time estimates in reliable transport protocols. In *Proceedings ACM SIGCOMM 87*, pp. 2–7

Limmington P.F. (1983). Fundamentals of the layer service definitions and protocol specifications. *Proc. IEEE*, **71**, (December) 1341–5

Markley R.W. (1990). *Data Communications and Interoperability*. Prentice-Hall

McConnell J. (1988). *Internetworking Computer Systems*. Prentice-Hall

Neumann J. (1983). OSI transport and session layers: services and protocol. In *Proceedings INFOCOM 83*

Rose M.T. and Cass D.E. (1987). OSI transport services on top of the TCP. *Computer Networks and ISDN Systems*, **12**, 159–73

Spragins J.D. *et al.* (1991). *Telecommunications: Protocols and Design*. Reading MA: Addison-Wesley

Sunshine C.A. and Dalal Y.K. (1978). Connection management in transport protocols. *Computer Networks*, **2**, 454–73

第12章

Abbruscato C.R. (1984). Data encryption equipment. *IEEE Communications*, **22**, (September) 15–21

Caneschi F. (1986). Hints for the interpretation of the ISO session layer. *Computer Communication Review*, (July)

Diffie W. and Hellman M.E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, **IT-22**, (November) 644–54

Diffie W. and Hellman M.E. (1977). Exhaustive cryptoanalysis of the NBS data encryption standard. *IEEE Computer Magazine*, **10**, (June) 74–84

Emmons W.F. and Chandler A.S. (1983). OSI session layer: services and protocols. *Proceedings IEEE*, **71**, (December) 1397–1400

Hellman M.E. (1987). Commercial encryption. *IEEE Network Magazine*, **1**, (April) 6–10

Henshall J. and Shaw A. (1988). *OSI Explained – End-to-End Computer Communication Standards*. Ellis Horwood

ISO (1986). *Connection-oriented Session Service and Protocol Definitions* (IS 8326/7)

ISO (1987). *Connection-oriented Presentation Service and Protocol Definitions* (IS 8822/3)

ISO (1988). *Commitment Concurrency and Recovery* (IS 8649/50)

ISO (1988). *ASN.1 and its Encoding Rules* (IS 8824/5)

ISO (1988). *Association Control Service Element* (IS 8649/50)

Jueneman J.J. *et al.* (1985). Message authentication. *IEEE Communications*, **23**, 29–40

National Bureau of Standards (1977). *Data Encryption Standard*. Federal Information Processing Standard Publication

Needham R.M. and Schroeder M.D. (1978). Using encryption for authentication in large networks of computers. *Communications ACM*, **21**, (December) 993–9

Neumann J. (1983). OSI transport and session layers: services and protocol. In *Proceedings INFOCOM 83*

Rivest R.L., Shamir A. and Adleman L. (1978). On a method for obtaining digital signatures and public key cryptosystems. *Comm. ACM*, **21**, (February) 120–6

Tardo J.J. (1985). Standardizing cryptographic services at OSI higher layers. *IEEE Communications*, **23**, (July) 25–9

第13章

Black U. (1987). *Computer Networks: Protocols, Standards and Interfaces*. Prentice-Hall

Black U. (1989). *Data Networks: Concepts, Theory and Practice*. Prentice-Hall

Chilton P. (1990). *X.400: The Messaging and Interconnection Medium for the Future*. NCC Publications

Comer D.E. (1991) *Internetworking with TCP/IP, Volume 1* 2nd edn. Prentice-Hall

Davidson J. (1988). *An Introduction to TCP/IP*. New York: Springer Verlag

Gilmore B. (1987). A user view of virtual terminal standardization. *Computer Networks and ISDN Systems*, **13**, 229–33

Henshall J. and Shaw A. (1988). *OSI Explained: End-to-End Computer Communication Standards*. Ellis Horwood

ISO (1987). *File Transfer Access and Management* (IS 8571/4)

ISO (1987). *Job Transfer and Manipulation* (IS 8831/2)

ISO (1988). *Virtual Terminal* (IS 9040/1)

Klerer S.M. (1988). The OSI management architecture: an overview. *IEEE Network*, **2**, (March) 20–9

Lewan D. and Long H. (1983). The OSI file service. *Proc. IEEE*, (December)

Limmington P.F. (1984). The virtual filestore concept. *Computer Networks*, **8**, 13–16

McLeod-Reisig S.E. and Huber K. (1986). ISO virtual terminal protocol and its relationship to TELNET. In *Proceedings IEEE Computer Networking Symposium*, pp. 110–19

Svoboda L. (1984). File servers for network-based distributed systems. *Computing Surveys*, **16**, (December) 353–98

第14章

Baran P. (1964). On distributed communication networks. *IEEE Transactions on Communication Systems*, **CS12**, (March) 1–9

Black U. (1988). *Data Communications and Distributed Networks*. Prentice-Hall

Comer D.E. (1991). *Internetworking with TCP/IP: Volume 1* 2nd edn. Prentice-Hall

Dolan M. (1984). Minimal duplex connection capability in the top three layers of the OSI reference model. In *Proceedings SIGCOM 84*

Henshall J. and Shaw A. (1988). *OSI Explained: End-to-End Computer Communication Standards*. Ellis Horwood

Hutchison G. and Desmond C.L. (1987). Electronic data exchange. *IEEE Network Magazine*, **1**, (October) 16–20

Langsford A. (1984). The open system users programming interfaces. *Computer Networks*, **8**, 3–12

Partridge C. (1986). Mail routing using domain names: an informal tour. In *Proceedings USENIX Summer Conference*

Sloman M. and Kramer J. (1987). *Distributed Systems and Computer Networks*. Prentice-Hall

附录A

Hamming R.W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, **29**, (April) 147–60

Peebles P.Z. (1987). *Digital Communication Systems*. Prentice-Hall

Petersen W.W. (1961). *Error Correcting Codes*. MIT Press

Sklar B. (1988). *Digital Communications*. Prentice-Hall

Sweeney P. (1991). *Error Control Coding*. Prentice-Hall

Viterbi A.J. (1971). Convolutional codes and their performance in communication systems. *IEEE Transactions on Communication Systems*, **19**(5)

IEEE standards can be obtained from: IEEE Press, 345 East 47th Street, New York, NY10017, USA.

ISO and ITU-T standards can be obtained from: International Telecommunications Union, Place de Nations, 1211 Geneva, Switzerland.

缩 略 语

AAL ATM adaptation layer ATM 适配层

ABM Asynchronous balanced mode 异步平衡方式

AC Alternating current 交流

ACK Acknowledgment 确认

ACS Access control store 访问控制存储

ACSE Association control service element 关联控制服务部件

ADCCP Advanced data communications control procedure 先进数据通信控制规程

ADM Add drop multiplexer 添加丢弃多路复用器

AE Application entity 应用实体

AFI Authority and format identifier 授权与格式标识符

AMP Active monitor present 当前现用监控器

AMT Agent management task 代理管理任务

ANSI American National Standards Institute 美国国家标准学会

ARM Asynchronous response mode 异步应答方式

ARP Address resolution protocol 地址解析协议

ARPA Advanced Research Projects Agency 高级研究计划局

ARR Automatic repeat request 自动重复请求

ASCII American Standards Committee for Information Interchange 美国信息交换标准委员会

ASE Application service element 应用服务元素

ASK Amplitude-shift keying 幅移键控

ASN.1 Abstract syntax notation one 抽象语法表示法1

ATM Asynchronous transfer mode 异步传输模式

ATMR ATM ring ATM环

AU Administrative unit 管理单元

AUI Attachment unit interface 附件接口

AUU ATM layer user-to-user ATM层用户到用户

BCC Block check character 块校验字符

BCS Basic combined subset 基本组合子集

BER Bit error rate/ratio 位误码率

BISDN Broadband ISDN 宽带综合业务数字网络

BISYNC Binary synchronous control 二进制同步控制

BOM Beginning of message 报文开始

BPDU Bridge PDU 网桥协议数据单元

BPSK Binary PSK 二进制相移键控
BRI Basic rate interface 基本速率接口
BUS Broadcast and unknown address server 广播和未知服务器
BWB Bandwidth balancing 宽带平衡

CA Collision avoidance 冲突避免
CAL Computer aided learning 计算机辅助学习
CASE Common application service element 公共应用服务元素
CATV Community antenna television 公用天线电视
CBC Chain block cipher 链式字块加密
CBDS Connectionless broadband data service 无连接宽带数据服务
CBR Constant bit rate 固定比特率
CC Cell controller/Cluster controller 信元控制器/群集控制器
CCA Conceptual communication area 概念通信区域
CCITT International Telegraph and Telephone Consultative Committee(now ITU-T) 国际电话电报咨询委员会（现国际电信联盟标准化部）
CCR Commitment, concurrency, and recovery 托付、并发和恢复
CD Carrier detect/Collision detect 载波检测/冲突检测
CDC Countdown counter 递减计数器
CDMA Code division multiple access 码分多路访问
CFM Cipher feedback mode 加密反馈方式
CIE Customer interface equipment 用户接口设备
CLLM Consolidated link layer management 统一链路层管理
CLNAP Connectionless network access protocol 无连接网络访问协议
CLNIP Connectionless network interface protocol 无连接网络接口协议
CLNS Connectionless network service 无连接网络服务
CLP Cell loss priority 信元丢失优先权
CLS Connectionless server 无连接服务器
CMIP Common management information protocol 公共管理信息协议
CMISE Common management information service element 公共管理信息服务单元
COM Continuation message 连续报文
CONS Connection-oriented network service 面向连接网络服务
CRC Cyclic redundancy check 循环冗余校验
CRMA Cyclic reservation multiple access 循环预约多路访问
CS Carrier sense/Convergence sublayer 载波侦听/会聚子层
CSCW Computer-supported cooperative working 计算机支持的协同工作
CSMA Carrier sense multiple access 载波侦听多路访问
CSMA/CA CSMA with collision avoidance 冲突避免载波侦听多路访问
CSMA/CD CSMA with collision detection 带冲突检测的载波侦听多路访问
CSPDN Circuited switched PDN 电路交换PDN

CT Claim token 要求令牌
CTS Clear-to-send 允许清除发送

DA Destination address 目标地址
DAP Directory access protocol 目录访问协议
DAS Dual attach station 双连接站
DAT Duplicate address test 重复地址测试
DC Direct current 直流
DCE Data circuit terminating equipment 数据电路端接设备
DES Data encryption standard 数据加密标准
DFW Distributed foundation wireless 分布式无线基础
DIB Directory information base 目录信息库
DIT Directory information tree 目录信息树
DLC Data link control 数据链路控制
DLCT Data link connection identifier 数据链路连接标识符
DLE Data link escape（character） 数据链路转义（字符）
DMPDU Derived MAC PDU 派生MAC PDU
DN Distinguished name 判别名
DNA Distributed network architecture（DEC） 分布式网络体系结构（DEC）
DNS Domain name server 域名服务器
DPC Designated port cost 指派端口成本
DPLL Digital phase-locked loop 数字锁相环
DQDB Distributed queue dual bus 分布式队列双总线
DS Directory services 目录服务
DSA Directory service agent 目录服务代理
DSD Data structure definition 数据结构定义
DSP Directory system protocol/Domain specific part 目录系统协议/特定域部分
DTE Data terminal equipment 数据终端设备
DTP Distributed transaction processing 分布式事务处理
DU Data unit 数据单元
DUA Directory user agent 目录用户代理
DUI Data unit identifier 数据单元标识符
DVA Distance vector algorithm 距离向量算法

EBCDIC Extended binary coded decimal interchange code 扩充的二—十进制交换码
ECB Electronic code book/Event control block 电子代码书/事件控制块
ECMA European Computer Manufacturers Association 欧洲计算机制造商协会
ECP Error correcting part 纠错部分
ED End delimiter 结束定界符
EDI Electronic data interchange 电子数据交换

EGP Exterior gateway protocol 外部网关协议
EIA Electrical Industries Association 电子工业协会
EOM End of message 报文结束
EOS End of stream 流结束
EPA Enhanced performance architecture 增强性能体系结构
ES End system 端系统
ETSI European Telecommunications Standards Institute 欧洲电信标准学会

FADU File access data unit 文件访问数据单元
FC Frame control 帧控制
FCS Frame check sequence 帧校验序列
FDDI Fiber distributed data interface 光纤分布式数据接口
FDM Frequency-division multiplexing 频分多路复用
FDMA Frequency division multiple access 频分多路访问
FEP Front-end processor 前端处理机
FIB Forwarding information base 转发信息库
FIFO First-in, first-out 先进先出
FRA Frame relay adapter 帧中继适配器
FS Frame status 帧状态
FSK Frequency-shift keying 频移键控
FTAM File transfer access and management 文件传送、访问和管理
FTP File transfer protocol 文件传送协议
FU Functional unit 功能单元

GFC Generic flow control 生成流量控制
GFI Group format identifier 组格式标识符

HA Hardware address 硬件地址
HDLC High-level data link control 高级数据链路控制
HEC Header checksum 报头校验和
HODSP High-order domain specific port 高阶特定域端口
HPNPP High-priority next port pointer 高优先级下一端口指针
HPTHT High-priority token hold timer 高优先级令牌持有时间

IA5 International alphabet number five 国际字母表5
IC Input controller 输入控制器
ICMP Internet control message protocol 因特网控制消息协议
IDI Initial domain identifier 初始域标识符
IDP Initial domain part 初始域部件
IEEE Institute of Electrical and Electronics Engineers 电气与电子工程师协会

IETF Internet Engineering Task Force 因特网工程任务组
IGP Interior gateway protocol 内部网关协议
IMDD Intensity modulation with direct detection 直接检测强度调制
IMP Interface message processor 接口消息处理程序
IMPDU Initial MAC PDU 初始MAC PDU
INMS Integrated network management system 综合网络管理系统
IP Internet protocol 因特网协议
IPM Interpersonal messaging 个人间报文传递
IS Intermediate system 中间系统
ISDN Integrated services digital network 综合业务数字网
ISI Intersymbol interference 码间干扰
ISO International Standards Organization 国际标准化组织
ITU-T International Telecommunications Union-Telecommunications (Sector) 国际电信联盟—电信标准化部
IWU Interworking unit 网际互连单元

JTM Job transfer and manipulation 作业传送和操纵

LAN Local area network 局域网
LAPB Link access procedure, balanced 平衡式链路访问规程
LCN Logical channel number 逻辑信道号
LE LAN emulation 局域网仿真
LEC LE client 局域网仿真客户机
LECID LEC identifier 局域网仿真客户机标识符
LECS LE configuration server 局域网仿真配置服务器
LED Light-emitting diode 发光二级管
LEP LE protocol 局域网仿真协议
LES LE server 局域网仿真服务器
LGN Logical group number 逻辑组号
LLC Logical link control 逻辑链路控制
LS Link state 链路状态
LSAP Link service access point 链路服务访问点
LTE Line termination equipment 线路端接设备
LWE Lower window edge 窗口下边界

MA Multiple access 多路访问
MAC Medium access control 介质访问控制
MAN Metropolitan area network 城域网
MAP Manufacturing automation protocols 制造业自动化协议
MAS Management application service 管理应用服务

MCP MAC conversion protocol MAC会聚协议
MDSE Message delivery service element 报文投递服务单元
MHS Message handling service 消息处理服务
MIB Management information base 管理信息库
MID Message/Multiplexing identifier 报文标识符/多路复用标识符
MII Media-independent interface 介质无关接口
MIT Management information tree 管理信息树
MLP Multilink procedure 多链路规程
MMR Modified-modified read 改进—改进READ
MMS Manufacturing messaging service 制造业消息服务
MO Managed object 管理对象
MOTIS Message-oriented text interchange standard 面向报文的文本交换标准
MRSE Message retrieval service element 报文检索服务单元
MS Message store 报文存储
MSS MAN switching system 城域网交换系统
MSSE Message submission service element 报文提交服务单元
MTA Message transfer agent 消息传送代理
MTSE Message transfer service element 报文传送服务单元
MUX Multiplexer 多路复用器

NAK Negative acknowledgment 否定应答
NBS National Bureau of Standards 国家标准局
NC Network connection 网络连接
NEXT Near end crosstalk 近端串扰
NMS Network management system 网络管理系统
NNPP Normal next-port pointer 正常下一端口指针
NPA Network point of attachment 网络接入点
NPDU Network-layer PDU 网络层PDU
NRM (Unbalanced) normal response mode （非平衡）正常响应模式
NRZ Non-return to zero 不归零编码
NRZI Non-return to zero inverted 不归零反相编码
NS Network service 网络服务
NSAP Network service access point 网络服务访问点
NSDU Network service data unit 网络服务数据单元
NT Network termination 网络终端

OFDM Orthogonal frequency division multiplexing 正交频分多路复用
OOK On-off keying 开关键
OSI Open systems interconnection 开放系统互连
OSIE Open system internetworking environment 开放系统互连环境

OSPF Open shortest path first 开放式最短路径优先

PA Point of attachment/Pre arbitrated 接入点/预先仲裁
PABX Private automatic branch exchange 专用自动小交换机
PAD Packet assembler-disassembler 包装配器与拆卸器
PAM Pulse amplitude modulated 脉冲幅度调制
PAU Portable access unit 便携式访问单元
PBX Private branch exchange 专用小型交换机
PCI Protocol connection identifier 协议连接标识符
PDH Plesiochronous digital hierarchy 准同步数字系列
PDN Partial distinguished name/Public data network 局部判别名/公用数据网
PDU Protocol data unit 协议数据单元
PDX Private digital exchange 专用数字交换机
PE Presentation entity 表示层实体
PIM Physical interface module 物理接口模型
PISO Parallel in, serial out 并行输入串行输出
PIXEL Picture element 像素
PLC Physical layer convergence/Programmable logic controller 物理层会聚/可编程逻辑控制器
PLP Packet layer protocol 分组层协议
PMD Physical medium dependent 物理介质相关
PPDU Packet PDU 分组PDU
PPDU Presentation PDU 表示层PDU
PPM Presentation protocol machine/Pulse position modulation 表示层协议机/脉冲位置调制
PPSDN Public packet switched data network 公共分组交换数据网络
PRI Primary rate interface 基群速率接口
PSAP Presentation service access point 表示层服务访问点
PSDN Packet switched data network 分组交换数据网络
PSE Packet switching exchange 分组交换机
PSK Phase-shift keying 相移键控
PSPDN Packet switched PDN 分组交换PDN
PSTN Public switched telephone network 公用交换电话网
PTE Path termination equipment 通路终端设备
PTI Payload type identifier 净荷类型标识符
PTT Post, telephone, and telecommunications (authority) 邮电、电信管理（机构）
PVC Permanent virtual connection 永久虚连接

QA Queued arbitrated 队列仲裁
QOS Quality of service 服务质量

QPDS Queued packet distributed switch 队列包分布交换

RARP Reverse ARP 反向地址解析协议
RC Request counter/Robot controller 请求计数器/机器人控制器
RCU Remote concentrator unit 远端集中器
RDA Remote database access 远程数据库访问
RDN Relative distinguished name 相关判别名
RF Radio frequency 射频
RI Ring indication 振铃指示
RIP Routing information protocol 路由信息协议
ROM Read only memory 只读存储器
ROSE Remote operations service element 远程操作服务单元
RP Root port 根端口
RPC Root path cost 根路径费用
RSE Real system environment 实际系统环境
RTS Request to send 请求发送
RTSE Reliable transfer service element 可靠的传送服务单元
RVCI Ring virtual channel identifier 虚通道振铃标识符

SA Source address 源地址
SAAL Signaling AAL 信令AAL
SAP Service access point 服务访问点
SAPI Service access point identifier 服务访问点标识符
SAR Segmentation and reassembly 分段与重装
SAS Single attach station 单连接站
SASE Specific application service element 特定应用服务单元
SAT Synchronous allocation time 同步分配时间
SCP Signaling control point 信令控制点
SD Start delimiter 起始定界符
SDH Synchronous digital hierarchy 同步数字系列
SDLC Synchronous data link control 同步数据链路控制
SDSE Submission and delivery service element 提交发送服务元素
SDU Service data unit 服务数据单元
SE Session entity 会话层实体
SEAL Simple and efficient AAL 简单有效的AAL
SEL Selector 选择器
SFD Start-of-frame delimiter 帧起始定界符
SI Subnet identifier 子网标识符
SIP SMDS interface protocol SMDS接口协议
SIPO Serial in, parallel out 串行输入并行输出

SLP *Single link procedure* 单链路规程
SMAE System management application entity 系统管理应用实体
SMASE System management application service element 系统管理应用服务单元
SMDS Switched multimegabit data service 交换式多兆位数据服务
SMF System management functions 系统管理功能
SMP Standby monitor present 当前备用监控器
SMT Station management 站管理
SMTP Simple mail transfer protocol 简单邮件传送协议
SNA Systems network architecture (IBM) 系统网络体系结构
SNDAP Subnetwork dependent access protocol 子网相关访问协议
SNDCP Subnetwork dependent convergence protocol 子网相关会聚协议
SNICP Subnetwork independent convergence protocol 子网无关会聚协议
SNMP Simple network management protocol 简单网络管理协议
SNR Signal-to-noise ratio 信噪比
SONET Synchronous optical network 同步光纤网络
SOS Start of stream 流开始
SPDU Session PDU 会话层PDU
SPF Shortest-path first 最短路径优先
SPM Session protocol machine 会话协议机
SSAP Session service access point 会话服务访问点
SSM Single segment message 单段报文
STE *Section termination equipment* 段终端设备
STM Synchronous transport mode 同步传输模式
STP Shielded twisted pair 屏蔽双绞线
STS Synchronous transport signal 同步传输信号
SVC Switched virtual connection 交换虚连接
SVCC Signaling VCC 信令VCC

TA Terminal adapter 终端适配器
TC Transport connection 传输连接
TCP Transmission control protocol 传输控制协议
TCU Trunk coupling unit 中继耦合单元
TDM Time-division multiplexing 时分多路复用
TDMA Time division multiple access 时分多路访问
TE Transport entity 传输层实体
TEI Terminal endpoint identifier 终端界点标识符
TFTP Trivial file transfer protocol 简纯文件传输协议
THT Token hold time 令牌持有时间
TOP 技术和办公协议
TP4 （OSI）传送协议级别4

TRT 令牌循环定时器
TSAP Transport service access point 传输服务访问点
TSE Terminal switching exchange 终端交换机
TTL Transistor transistor logic 晶体管—晶体管逻辑
TTRT Target TRT 目标TRT
TTT Target transmission time 目标传输时间
TU Tributary unit 支路单元
TUG Tributary unit group 支路单元组

UA User agent 用户代理
UART Universal asynchronous receiver transmitter 通用异步收发器
UDB User data buffer 用户数据缓存
UDP User datagram protocol 用户数据报协议
UE User element 用户单元
UIP User interface part 用户接口部分
UNA Upstream neighbor' s address 上游邻居地址
UNI User-network interface 用户网络接口
UP Unnumbered poll 未编号轮询
USRT Universal synchronous receiver transmitter 通用同步收发器
UTP Unshielded twisted pair 非屏蔽双绞线
UWE Upper window edge 窗口上边界

VBR Virtual bit rate 虚比特率
VC Virtual connection 虚连接
VCC Virtual channel connection 虚通道连接
VCI Virtual channel identifier 虚通道标识符
VG Voice grade 音频段
VPI Virtual path identifier 虚路径标识
VPN Virtual private network 虚拟专用网
VT Virtual terminal 虚拟终端
VTE Virtual terminal environment 虚拟终端环境

WAN Wide area network 广域网

XID Exchange identification 交换识别

索　引

索引中的页码为英文原书页码，与书中边栏的页码一致。

B

E

J

K

L

M

P

S

T

Z