

热点  
编程

系列丛书

内附范例光盘

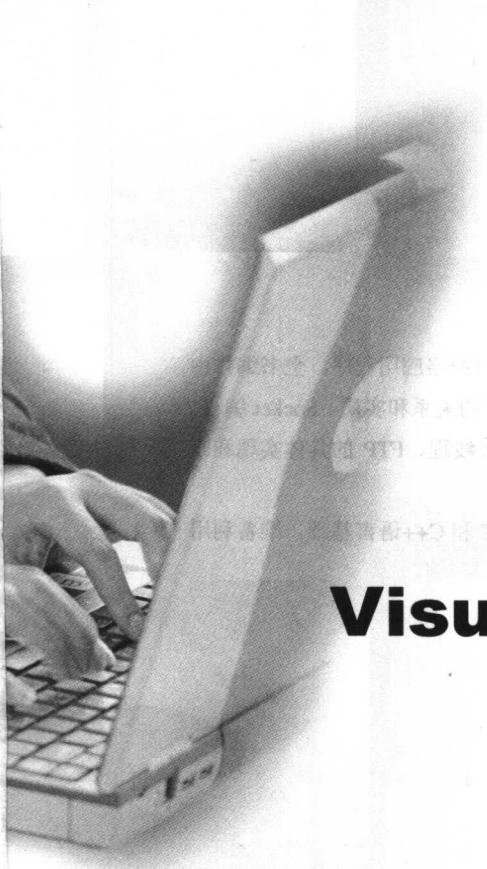
易君 编著

# Visual C++ .NET

## 网络编程

```
SOCKADDR_IPX ipxAddr;  
ipxLen = sizeof(ipxAddr);  
  
s = socket(AF_IPX, SOCK_DGRAM, IPPROTO_IP);  
if (s == INVALID_SOCKET)  
{  
    ShowErrorMessage(...);  
    return;  
}  
  
memset(&ipxAddr, 0, sizeof(SOCKADDR_IPX));  
ipxAddr.ss_family = AF_IPX;  
ipx.ss_socket = htons(6800);  
if (!bind(s, (SOCKADDR*)&ipxAddr, sizeof(SOCKADDR_IPX))){  
    ShowErrorMessage(...);  
    return;  
}  
  
getsockname(s, (SOCKADDR*)&ipxAddr, len);
```

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE



# Visual C++.NET 网络编程

易君 编著

中国铁道出版社

2003 • 北京

(京)新登字 063 号

## 内 容 简 介

本书详细地讲述了如何在 Visual Studio.NET 集成开发环境下编写网络应用程序。全书编排由浅入深,内容丰富。主要包括因特网的历史、IP 地址、各种网络协议的关系和实质、Socket 编程、客户/服务器模型、WinSock 高级编程、远程控制、WinInet 编程与多线程、FTP 的具体实现和断点续传等内容。

本书适合有一定 Visual C++ 应用程序的开发经验,具备一定的 C 和 C++ 语言基础,准备利用 Visual Studio.NET 开发网络应用程序的用户使用。

## 图书在版编目 (CIP) 数据

Visual C++.NET 网络编程/易君编著. —北京:中国铁道出版社, 2003. 4

(热点编程技术系列)

ISBN 7-113-05242-8

I. V… II. 易… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 031623 号

书 名: Visual C++.NET 网络编程

作 者: 易 君

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 魏 春

责任编辑: 苏 茜 袁秀珍

封面设计: 孙天昭

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张: 18.75 字数: 446 千

版 本: 2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN7-113-05242-8/TP·932

定 价: 33.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

随着科学技术的进步, 计算机网络技术的发展日新月异, 网络在无声无息间悄悄地走进了每一个人的生活中。无论你承不承认, 在将来每个人的生活都将或多或少地依赖于网络, 网络已经并将继续成为我们获取信息、学习及娱乐的主要工具之一。对于每一个在未来的职业战场上想取得优势的人, 特别是从事信息业、计算机业和准备从事该行业的人来说, 一定的网络的理论知识和实践技能将是不可或缺的。本书作为一本介绍实际网络编程知识并与一定理论相结合的计算机书籍, 将为你揭示网络编程奥秘, 并作为打开网络知识的敲门砖, 将你带入网络知识的殿堂。

本书选用微软公司的最新开发工具 VC.net 作为编程载体, 是因为 Visual C++ 自从其出现以来, 因为其开发过程的简明、开发方式灵活、可扩展性强等优点赢得了广大开发人员的青睐。而使用 C++ 语言可以使得编程者对面向对象的思想有更深入的了解。VC.net 捆绑的 MFC 提供了大量的封装类, 为我们网络程序的开发带来很大的方便, 同时如果想编写较为底层一些的网络应用程序, VC.net 更是首选的编程开发工具。因此, 无论在灵活性还是程序的效率方面 VC.net 都较 Delphi、VB 等开发工具更胜一筹。

## 本书的读者对象

本书的读者最好有一定的 Visual C++ 应用程序的开发经验, 具备一定的 C 和 C++ 语言基础。理解面向对象的一些概念和基本特征。只要有了这些最基本的知识, 就可以读懂本书了。如果你还没有学过 C++, 没有写过 Visual C++ 应用程序, 建议你先购买一本最基本的 C++ 书籍和 Visual C++ 的入门书籍。

当然如果具备一定的网络知识那就更好了, 不过如果你一点网络知识都不懂, 也没有关系, 本书会概要性地介绍一些基础的网络理论知识。

## 关于本书

笔者经过多年的学习和编程实践认为选书犹如选老师, 好的老师可以让你收获很大, 让你对某个领域有个全面、深刻的了解, 并能告诉你将来的方向。尤其是某一门课程的入门老师更是如此。一本好书尤其是一本带领你走入某个知识领域殿堂的领路书, 必须具备下列特点:

1. 它的语言必须清楚明确, 就像一位老师, 水平高深, 知识全面固然重要, 但要能使 学生最大受益, 必须能清楚明晰地表达自己的见解, 传授知识。
2. 书的内容应该符合其定位的读者的知识水平、口味。书中的每一个重要的知识点都应该讲明白, 让读者能看懂, 领会。如果一本书只是大量知识点的简单罗列, 那对于读者的帮助也是不大的。所以一本书不应该追求其讲述了几个知识点, 而要看讲清楚了几个知识点。
3. 语言简练, 繁简得当, 对读者无关的或者可有可无不重要的内容应一笔带过, 力求 每一个字每句话都有它的价值, 书不在厚而在于精。就像我们考试复习一样, 书本越看越薄、越看越精。同时从经济上考虑, 也力求使该书对于读者物有所值。
4. 要与实践结合, 尤其是计算机方面的知识, 很强调其应用性。同时中国有句古话“授之以鱼不如授之以渔”。一本书要力求对读者有所启发, 使读者能举一反三, 触类

旁通。不必将所有的实现均显现给读者，而应选择性地展示有代表的实例，留给读者一些思考模仿的空间，这样读者才是真正学到了方法而非死的代码。

在学习知识，获取能力的道路上，书籍是人类永远的老师，上面几条是笔者多年来实践的一些感想体会。本书也是按照上面的几条原则进行编写的。力求使读者物有所值、取得最大的收获。本书信息量大，但力求使语言简练，言之有物，实例丰富。

本书的内容组织如下所示：

第 1、2 章讲述必要的计算机网络理论知识，为以后的编程工作打下基础。如果读者已经对计算机网络理论知识有深刻的了解，这两章可以忽略不看。

第 3 章详细介绍套接字编程的知识和方法，并以实例介绍对套接字的封装和代码重用的方法。

第 4 章介绍经典的网络程序编程算法，理论与实际相结合，这一章可以使你深刻理解网络编程的实质。

第 5、6 章详细介绍如何使用 VC.net 编写网络程序，详细介绍 VC.net 在网络编程方面的特性，封装类和 VC.net 的实际应用

第 7 章详细介绍网络应用层各协议的定义及实质和它们的实现方法，并以用得最广的 FTP（文件传输协议）为例展示了协议的应用，并力求使读者举一反三，对应用层各协议有深刻的理解。

第 8 章介绍断点续传技术的理论知识、使用方法，使读者能在实际中更加完善自己的网络应用程序。

本书适合于编写 VC.net 网络程序的初学者，准备使用 VC.net 开发网络应用程序或者较为低层网络程序的用户。特别适合于具备一定的 Visual C++ 或者 VC.net 编程经验的中高级用户使用。

本书是多人劳动的共同成果，除了封面署名的作者以外，参与本书的编写、资料搜集、程序测试和修订的人员还有张京、齐占锋、易霞、李锐、陈海峰、周密、李吉、曹杰峰、王辉、吴振新、周宁、刘志成、于自跃、吕伟伟、刘振涛、李柯、吴波、李超、樊凤斌、朱毅、陈兰芳、崔仙翠、程瑞芬等同志参与了本书的编排工作，在此一并表示感谢。

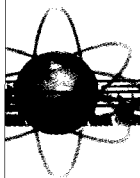
由于作者水平有限，书中不可避免存在疏漏之处，希望广大读者不吝赐教，我们也会在适当时间进行修订和补充，并发布在天勤网站：<http://www.tqbooks.net> “图书修订”栏目中。

作 者  
2003 年 4 月

# 目 录

|                             |    |
|-----------------------------|----|
| 第 1 章 因特网概述.....            | 1  |
| 1-1 因特网的发展历史.....           | 2  |
| 1-2 因特网的组织机构.....           | 2  |
| 1-3 因特网的分类.....             | 3  |
| 1-4 OSI 参考模型.....           | 5  |
| 1-5 TCP/IP 分层模型.....        | 6  |
| 1-6 TCP/IP 协议簇简介.....       | 7  |
| 本章总结.....                   | 9  |
| 第 2 章 IP,TCP/UDP 详述.....    | 11 |
| 2-1 IP.....                 | 12 |
| 2-1-1 IP 数据报格式.....         | 12 |
| 2-1-2 IP 数据报在网络中的传输.....    | 13 |
| 2-1-3 MTU.....              | 15 |
| 2-2 UDP.....                | 16 |
| 2-2-1 UDP 数据报格式.....        | 16 |
| 2-2-2 UDP 数据报的传输.....       | 16 |
| 2-3 TCP.....                | 17 |
| 2-3-1 TCP 的传输和为应用提供的服务..... | 17 |
| 2-3-2 TCP 数据报格式.....        | 18 |
| 2-3-3 TCP 连接的建立与终止.....     | 18 |
| 2-3-4 包的丢失与重传.....          | 20 |
| 2-4 IP 地址.....              | 20 |
| 2-4-1 IP 地址基本格式.....        | 21 |
| 2-4-2 IP 地址划分.....          | 21 |
| 2-4-3 特殊 IP 地址.....         | 22 |
| 本章总结.....                   | 23 |
| 第 3 章 Windows 套接字.....      | 25 |
| 3-1 什么是套接字.....             | 26 |
| 3-2 客户机/服务器模型.....          | 26 |
| 3-3 Socket 编程基础.....        | 28 |
| 3-4 WinSock 函数概述.....       | 30 |
| 3-4-1 socket 函数.....        | 30 |





|        |                                                 |    |
|--------|-------------------------------------------------|----|
| 3-4-2  | bind 函数 .....                                   | 31 |
| 3-4-3  | listen 函数 .....                                 | 32 |
| 3-4-4  | accept 函数 .....                                 | 32 |
| 3-4-5  | connect 函数 .....                                | 33 |
| 3-4-6  | select 函数 .....                                 | 34 |
| 3-4-7  | send, sendto 函数 .....                           | 35 |
| 3-4-8  | recv, recvfrom 函数 .....                         | 37 |
| 3-4-9  | shutdown 函数 .....                               | 38 |
| 3-4-10 | closesocket 函数 .....                            | 39 |
| 3-4-11 | gethostbyaddr, gethostbyname 函数 .....           | 39 |
| 3-4-12 | gethostname, getpeername 和 getsockname 函数 ..... | 40 |
| 3-4-13 | getprotobyname, getprotobyname 函数 .....         | 41 |
| 3-4-14 | getservbyname, getservbyport 函数 .....           | 42 |
| 3-4-15 | getsockopt, setsockopt 函数 .....                 | 43 |
| 3-4-16 | htonl, htons, ntohl, ntohs 函数 .....             | 44 |
| 3-4-17 | inet_addr, inet_ntoa 函数 .....                   | 45 |
| 3-4-18 | ioctlsocket 函数 .....                            | 46 |
| 3-4-19 | winsock2 扩展函数 .....                             | 46 |
| 3-5    | 使用套接字编程 .....                                   | 47 |
| 3-5-1  | 公共函数库的编写 .....                                  | 47 |
|        | 本章总结 .....                                      | 67 |

## 第 4 章 客户端/服务器编程 .....

69

|       |                             |    |
|-------|-----------------------------|----|
| 4-1   | 服务器的并发 .....                | 70 |
| 4-2   | 客户端程序设计模型 .....             | 71 |
| 4-2-1 | TCP 客户机程序模型 .....           | 71 |
| 4-2-2 | UDP 客户机编程模型 .....           | 71 |
| 4-3   | 服务器端程序设计模型 .....            | 71 |
| 4-3-1 | 面向连接(TCP)的循环处理服务器模型设计 ..... | 72 |
| 4-3-2 | 面向连接(TCP)的并发服务器模型设计 .....   | 77 |
| 4-3-3 | 无连接的循环服务器模型设计 .....         | 77 |
| 4-3-4 | 无连接并发服务器模型设计 .....          | 78 |
| 4-3-5 | 各种服务器设计模式的比较 .....          | 78 |
|       | 本章总结 .....                  | 79 |

## 第 5 章 MFC 套接字编程 .....

81

|       |                      |    |
|-------|----------------------|----|
| 5-1   | CAsyncSocket 类 ..... | 82 |
| 5-1-1 | m_hSocket .....      | 82 |
| 5-1-2 | CAsyncSocket .....   | 82 |

|        |                           |     |
|--------|---------------------------|-----|
| 5-1-3  | Create .....              | 82  |
| 5-1-4  | Accept .....              | 83  |
| 5-1-5  | AsyncSelect .....         | 83  |
| 5-1-6  | Bind .....                | 84  |
| 5-1-7  | Connect .....             | 84  |
| 5-1-8  | GetLastError .....        | 85  |
| 5-1-9  | Listen .....              | 85  |
| 5-1-10 | Send .....                | 85  |
| 5-1-11 | SendTo .....              | 85  |
| 5-1-12 | Receive .....             | 86  |
| 5-1-13 | ReceiveFrom .....         | 86  |
| 5-1-14 | ShutDown .....            | 87  |
| 5-1-15 | Close .....               | 87  |
| 5-1-16 | 一些应该重载的函数 .....           | 88  |
| 5-2    | 使用 CAsyncSocket 类编程 ..... | 88  |
| 5-3    | CSocket 类 .....           | 117 |
| 5-3-1  | CSocket .....             | 117 |
| 5-3-2  | Create .....              | 117 |
| 5-3-3  | Attach .....              | 117 |
| 5-3-4  | CancelBlockingCall .....  | 118 |
| 5-3-5  | FromHandle .....          | 118 |
| 5-3-6  | IsBlocking .....          | 118 |
| 5-3-7  | OnMessagePending .....    | 118 |
| 5-4    | 使用 CSocket 进行网络编程 .....   | 118 |
| 5-5    | 主机地址与 MAC 地址的获取 .....     | 160 |
| 5-5-1  | 主机名与地址 .....              | 160 |
| 5-5-2  | MAC 地址 .....              | 161 |
| 5-6    | 远程控制软件简介 .....            | 170 |
| 5-6-1  | 远程控制软件的原理 .....           | 170 |
| 5-6-2  | 远程控制软件的基本功能 .....         | 171 |
| 5-6-3  | 与远程控制软件相关的一些知识 .....      | 171 |
| 5-7    | 多播与广播的编程方法 .....          | 174 |
| 5-7-1  | 广播 .....                  | 174 |
| 5-7-2  | 多播 .....                  | 176 |
|        | 本章总结 .....                | 177 |

## 第 6 章 MFC WinInet 编程..... 179

|       |                          |     |
|-------|--------------------------|-----|
| 6-1   | WinInet 封装的类 .....       | 180 |
| 6-1-1 | CInternetSession 类 ..... | 181 |





|       |                             |     |
|-------|-----------------------------|-----|
| 6-1-2 | CInternetConnection 类 ..... | 185 |
| 6-1-3 | CFtpConnection 类 .....      | 186 |
| 6-1-4 | CHttpConnection 类 .....     | 190 |
| 6-1-5 | CGopherConnection 类 .....   | 191 |
| 6-1-6 | 其他类及其成员函数 .....             | 192 |
| 6-2   | 使用 WinInet 编程的方法简介 .....    | 194 |
| 6-2-1 | 如何访问 WWW 服务器 .....          | 195 |
| 6-2-2 | 如何访问 FTP 站点 .....           | 195 |
| 6-2-3 | 如何访问 Gopher 站点 .....        | 196 |
|       | 本章总结 .....                  | 197 |

## 第 7 章 文件传输协议 (FTP) 编程 .....

199

|       |                              |     |
|-------|------------------------------|-----|
| 7-1   | FTP 协议简介 .....               | 200 |
| 7-1-1 | 数据表示与存储 .....                | 200 |
| 7-1-2 | 文件传输模式 .....                 | 202 |
| 7-1-3 | FTP 命令 .....                 | 203 |
| 7-1-4 | FTP 应答 .....                 | 206 |
| 7-2   | 使用 WinSock 编写 FTP 客户端 .....  | 208 |
| 7-3   | 使用 WinInet 来编写 FTP 客户端 ..... | 253 |
|       | 本章总结 .....                   | 287 |

## 第 8 章 断点续传 .....

289

|     |                    |     |
|-----|--------------------|-----|
| 8-1 | 断点续传的用途 .....      | 290 |
| 8-2 | FTP 协议的断点续传 .....  | 290 |
| 8-3 | HTTP 协议的断点续传 ..... | 291 |
|     | 本章总结 .....         | 291 |

# 1

## Chapter

# 因特网概述

### 本章重点:

在我们学习网络编程以前,我们必须学习一些基本的网络知识,实践必须以理论作为指导,所以我们有必要对基本的网络知识有一个全面的了解。

在这一章里,我们主要介绍网络的发展历史,网络的分类与拓扑结构,网络标准是由谁制订,如何制订的以及一些著名的参考模型。通过本章的学习,读者将对计算机网络有一个整体上的概念,并为以后的深入学习打下一个良好的基础。

### 本章要点:

- 因特网的发展历史
- 因特网的分类
- 因特网的组织机构
- OSI 参考模型
- TCP/IP 参考模型



## 1-1 因特网的发展历史

人类步入 21 世纪, Internet 通讯已经成为人类生活中不可缺少的部分。电子邮件, 网络聊天使人们能够方便快捷地传递消息, 现在, 计算机通信已成为我们社会结构的一个基本组成部分。网络被用于工商业的各个方面, 包括广告宣传、生产、发运、计划、报价和会计等。绝大多数公司拥有了多个网络。从小学到大学教育的各级学校都使用计算机网络为教师和学生提供联网图书信息的即时检索。计算机网络已遍布各个领域。

任何事物都有它的发展历程, 因特网这个家喻户晓的技术也有其发展的历史, 现在就让我们来看看因特网的发展历史: 首先让我们先用形式化的方法定义一下什么叫计算机网络, 大量分散但又互联的计算机共同为某一机构中所有计算机需求提供服务, 即服务是由互联的计算机共同完成的, 这种系统就被称为计算机网络。随着越来越多的机构都有了自己的计算机网络系统, 而大多数网络技术是面向具体的任务而设计的, 每个企业会选择具体通讯需要和预算的硬件技术, 而且没有哪种网络能满足所有的用途, 所以在单一的网络技术上不可能建立一种通用网络。在过去 20 年里, 一项被称为网际互连的技术取得了进展, 它把不同的物理网络互联起来, 使它们成为一个协调的整体。

60 年代末, 因特网作为一个试验, 开始于美国国防部远景规划局 (DARPA), 在那时, DARPA 主要致力于分组交换网络的研究, 它的主要成果是 ARPANET, ARPANET 使用传统的点到点租用线路互联, 同时 DARPA 也资助研究在无线网络和卫星通信信道上进行分组交换。网络硬件技术的多元化也促使 ARPA 开始网络互联的研究, 并推动了网际互联的发展。1969 年 12 月, 这个实验性网络通过 56K/s 的电路连接了四个接点的网络, 由于技术的可靠性, 被引入到两个类似的军事网络的建设中: 美国的 MILNET 和欧洲的 MINET, 数以千计的主机和用户把他们专用的网络接到 ARPANET, 这就形成了最初的“ARPA 因特网”。美国国家科学基金 (NSF) 在因特网的发展中起了重要作用, 20 世纪 70 年代末, 基金会筹建了 CSNET (Computer Science NETwork) 计划, 目的是为了将更多的计算机科学家联系在一起, NSF 规划围绕它的六个超级计算机中心, 建立接入网络。1986 年, 为了扩展网络, NSF 资助了一个新的广域主干网 (NSFNET), 它连接了所有的超级计算中心并实现了和 ARPANET 互联。由于 ARPANET 的可使用政策, 它禁止把因特网用于商业目的, 所以 ARPANET 于 1989 年停用。

1991 年, 由于业务量的增加, NSFNET 已经不能满足当前的需要, 于是, 1991 年 10 月, IBM、MERIT 和 MCI 组成了一个非盈利的公司, 叫做高级网络和服务 ANS (Advanced Network and Services)。1993 年 ANS 安装了一个新的网络取代 ARPANET, 叫 ANSNET。

随后的几年, 因特网逐年增长, 横跨了几百个位于美国和欧洲的网络。1994 年, 已经覆盖了 61 个国家和地区的近 300 万台计算机。到 2000 年, 因特网已经覆盖了 209 个国家和地区的 5000 多万台计算机。再随后出现了许多因特网服务提供商 (ISP), 因特网结构从核心网络 (NSFNET) 转移到由商业运营的结构体系上。

## 1-2 因特网的组织机构

从上节我们知道, 因特网起源于 ARPANET, 它是由 DARPA 资助的, 在 80 年代, 因特

网的技术开发改为由因特网活动委员会 (IAB) 来负责。

IAB 是 DARPA 于 1983 年重新组建 Internet 控制和配置委员会时形成的。IAB 的每个成员都是一个 Internet 任务组的主持者。分管研究某个或某个系列的重要课题。IAB 大约由 10 个任务组组成。这些组被命名为“因特网工程特别工作组”(Internet Engineering Task Force: IETF), 由对特定工程感兴趣的研究人员组成。随后, IAB 又组建了因特网工程指导小组 (Internet Engineering Steering Group: IESG)。

随着 TCP/IP 和因特网的发展, IAB 于 1989 年夏进行了重新组织。IAB 组织了一个新的委员会, 在委员会管辖下, IAB 组织包括两个主要的工作组: 因特网研究特别工作组 (Internet Research Task Force: IRTF) 和因特网工程特别工作组 (Internet Engineering Task Force: IETF), 如图 1-1 所示:

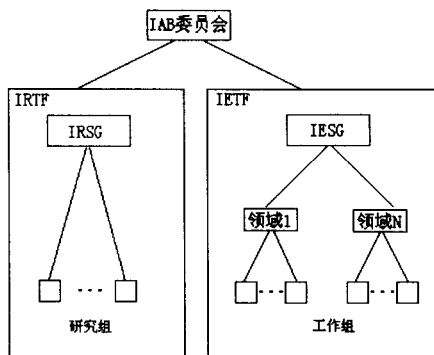


图 1-1 1989 年后重组的 IAB 结构

如图 1-1 所示, IAB 的两个主要组成部分是 IRTF 和 IETF, IETF 致力于中短期的工程问题, IESG 是 IETF 的指挥部, 管理协调下属的工作组; IRTF 即研究部, 协调有关 TCP/IP 协议或一般互联网结构的研究活动。与 IETF 一样, 也有一个指挥部 IRSG, 管理协调下属的研究组。IETF 是一个非常活跃的组织, 规模庞大, 而 IRTF 相对来说就是规模比较小的工作组, 不太活跃, 它的大部分研究工作已经由 IETF 代为完成了。

### 1-3 因特网的分类

目前还不存在统一的所有计算机网络都适用的分类方法, 一般主要以两个标准进行分类: 传输技术和规模。

按传输技术进行分类, 则可以把因特网分为两类:

1. 广播式网络。
2. 点对点网络。

顾名思义, 广播式网络就是像广播一样, 一台机器发出的消息可以被网络中任何一台机器接收到, 网络上所有的机器再检查该消息是不是发给自己的而采取不同的响应。就像用广播播送寻人启事一样, 路上的每个行人都听到了这条消息, 而只有被寻找的对象才根据广播去联系发寻人启事的人。而点对点网络由一对对机器之间多条连接构成, 点对点的网络就类

似于我们实际生活中的电话，两方的数据都有确定目标的发给对方。

按照网络的规模，计算机网络可以分为局域网、城域网、广域网和互联网等。

局域网就是我们经常称的 LAN，这种网络在我们的公司、部门、实验室很常见，一个小范围的区域中的机器互联在一起，可以共享文件，互相访问等。局域网的覆盖范围较小，网络传输延迟低，出错率也较低。广域网的拓扑结构有很多种，最常用的有两种：总线型和环形，如图 1-2 和 1-3 所示：

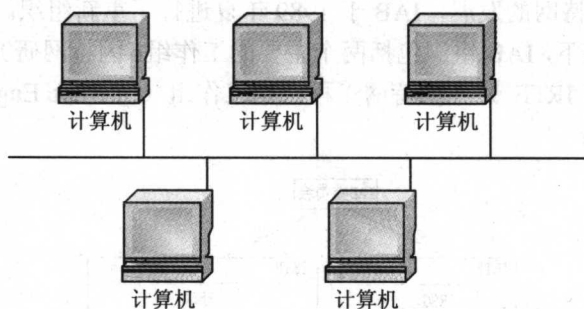


图 1-2 总线型拓扑结构

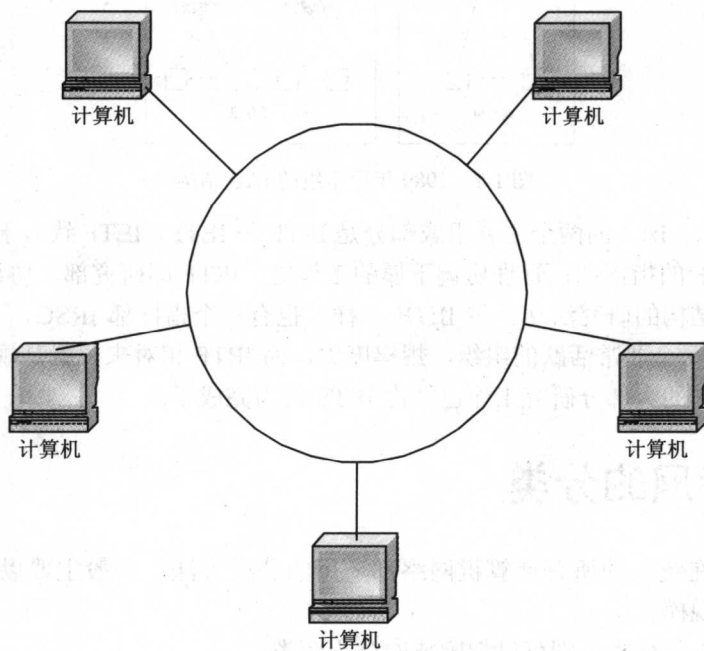


图 1-3 环形拓扑结构

在实际的组建局域网中，通常使用的是总线型拓扑结构，使用集线器（HUB）和网线将各台机器连接起来。

城域网又被称为 MAN，它使用与局域网相似的技术，不过覆盖的范围要比局域网大。通常可以覆盖一个公司或者一栋大楼或者一个城市。

广域网又被称为 WAN，它是更大规模的网络，通常可以跨域一个国家。它通过路由器或者交换机等网络设备将多个 LAN，MAN 连接起来。

互联网就是我们经常接触到的 Internet，它跨越几个大洲，跨越全世界的各个国家，将全世界连接起来，使世界变小了，不同国家和地区的人可以方便的交换信息或者聊天互致问候。

## 1-4 OSI 参考模型

OSI 参考模型的全称是 ISO OSI 开放互联参考模型 (Open System Interconnection Reference Model)，它是国际标准化组织 (ISO) 和国际电报电话咨询委员会 (CCITT) 共同提出的。它把计算机的网络过程应用到的网络介质共分为七层，所以又称网络七层参考模型，图 1-4 是 OSI 参考模型的示意图：

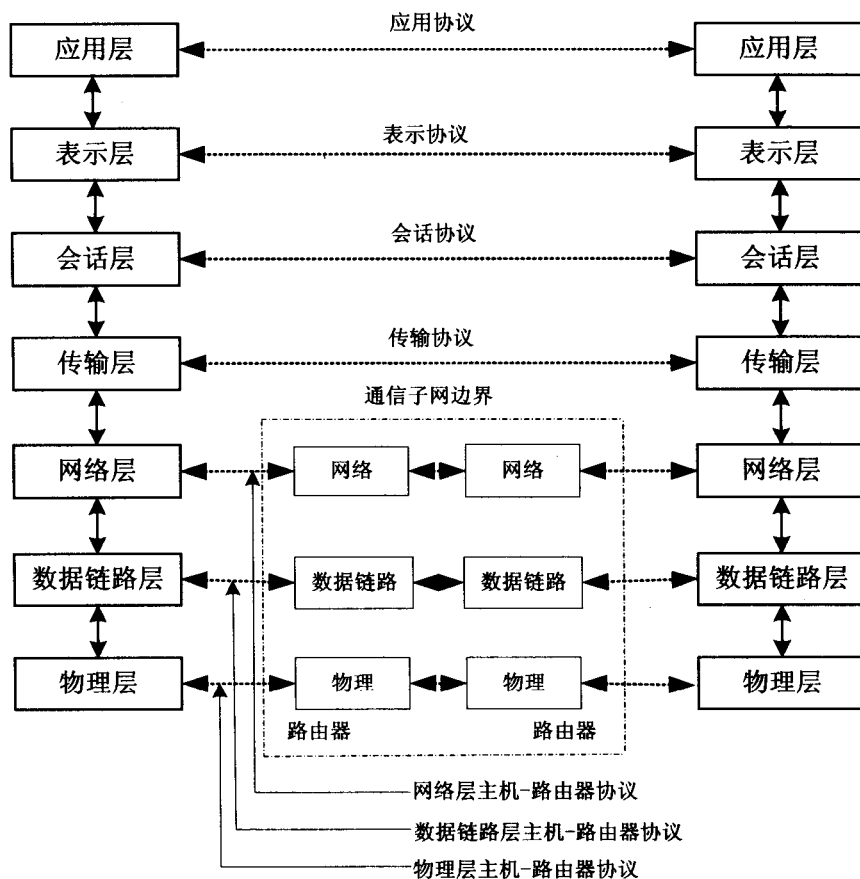


图 1-4 OSI 参考模型

OSI 参考模型从上到下依次是应用层、表示层、会话层、传输层、网络层、数据链路层和物理层。需要说明的是现在的网络体系结构不是基于 OSI 参考模型搭建的，但 OSI 模型的分层思想和方法还是具有重要的教学意义，因此几乎所有的网络书籍中都要涉及 OSI 参考模型。下面我们来大致看一下 OSI 这七层的的功能划分：



## 1. 物理层 (Physical Layer)

物理层涉及的是实际的网络物理介质, 涉及到在信道上传输的原始比特流。它来完成实际的数据流传输, 可定义电气信号、符号、线的状态和时钟要求, 数据编码和数据传输用的连接器。如常用的 RS-232 规范, 10BASE-T 的曼彻斯特编码等。

## 2. 数据链路层 (Data Link Layer)

数据链路层通过物理网络链路提供可靠的数据传输。其主要任务是加强物理层传输原始比特的功能, 使之对网络层提供更好的服务保障。数据链路层主要处理数据帧的可靠传输。该层和物理层都和硬件直接打交道, 对程序员是屏蔽的, 我们在实际的网络编程中是不会考虑这两层的情况的。

## 3. 网络层 (Network Layer)

网络层负责在源和终点之间建立连接。它包括网络寻径、流量控制和差错控制。它确定分组从源端到目的端路由的选择。网络层还必须处理分组在异质网络的传输问题。

## 4. 传输层 (Transport Layer)

传输层向高层提供可靠的端对端网络数据流服务。传输层的功能一般包括流控制、多路传输、虚电路管理以及差错校验和恢复。传输层要为会话层提供服务。会话层每请求建立一个传输连接, 传输层就为其创建一个独立的网络连接。如果传输连接需要较高的信息吞吐量, 传输层就为它创建多个网络连接, 以便让数据在这些网络连接上分流, 提高吞吐量。

## 5. 会话层 (Session Layer)

会话层允许不同机器上的用户建立会话关系。它建立、管理和终止表示层与实体之间的通信会话。会话允许进行类似传输层的普通数据的传输, 并提供了对某些应用有用的增强服务会话, 也可被用于远程登录到分时系统或在两台机器之间传递文件。

## 6. 表示层 (Presentation Layer)

表示层涉及数据的编码, 表示层以下的各层只关心可靠地传输比特流, 而表示层关心的是所传输信息的语法和语义。为了确保传输的信息能被对方应用层识别, 数据的编码要采用标准的格式。

## 7. 应用层 (Application Layer)

应用层是直接和用户打交道的最上面的一层, 也是程序员经常涉及到的, 应用层有很多的网络协议如文件传输, 网络管理等。应用层向应用软件提供访问网络资源的 API, 程序员通常使用这些 API 来访问、控制网络。

## 1-5 TCP/IP 分层模型

TCP/IP 分层模型又被称为因特网分层模型。与 OSI 参考模型不同, TCP/IP 模型更侧重于互联设备间的数据传送, 而不是严格的功能层次划分。它通过解释功能层次分布的重要性来做到这一点, 但它仍为设计者具体实现协议留下很大的余地。因此, OSI 参考模型在解释互联网络通信机制上比较适合, 但 TCP/IP 成为了互联网络协议的市场标准。TCP/IP 参考模



型比 OSI 模型更灵活。因此目前的网络产品大部分都是遵循 TCP/IP 分层模型的，图 1-5 就是 TCP/IP 分层模型示意图。该图同时给出了 TCP/IP 参考模型与 OSI 参考模型的对应关系。

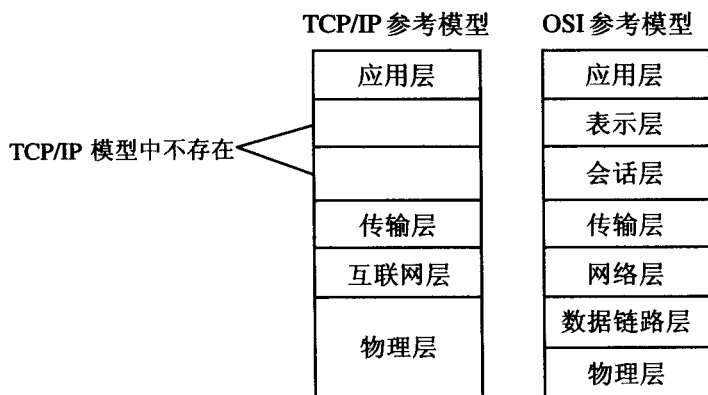


图 1-5 TCP/IP 分层模型示意图

TCP/IP 分层模型更强调功能分布而不是严格的功能层次的划分，因此它比 OSI 模型更灵活。下面简要介绍一下 TCP/IP 四个协议层的功能：

#### 1. 物理层（Physical Layer）

该层主要与网络的物理介质打交道，是 TCP/IP 中最底的一层，它要完成 IP 数据报的发送接收、寻址等功能。相当于 OSI 参考模型中的物理层和数据链路层，在该层定义了 ARP、RARP 等协议，提供 TCP/IP 协议的数据结构和实际物理硬件之间的接口。

#### 2. 互联网层（Network Layer）

对应于 OSI 参考模型的网络层，负责数据的包装、寻址、路由等功能。它相当于 OSI 参考模型中的网络层。本层包括 IP 协议、RIP 协议（路由信息协议），用于数据报的包装、路由传输。同时还包括 ICMP 协议（网间控制报文协议），用来提供网络诊断信息。

#### 3. 传输层（Transport Layer）

对应于 OSI 参考模型的传输层，提供端对端的通讯服务，提供两种数据发送方式：TCP 协议，提供可靠的面向连接的传输服务；UDP 协议，提供不可靠的面向无连接的传输服务。

#### 4. 应用层（Application Layer）

提供各种网络应用层协议：Whois、FTP、HTTP、Telnet 等。利用各种 API 向应用软件提供网络服务。

## 1-6 TCP/IP 协议簇简介

TCP/IP 协议是 TCP/IP 分层模型中的一系列协议的统称。TCP 协议和 IP 协议只是这个协议簇中的两个，TCP/IP 协议簇的示意图如图 1-6 所示：

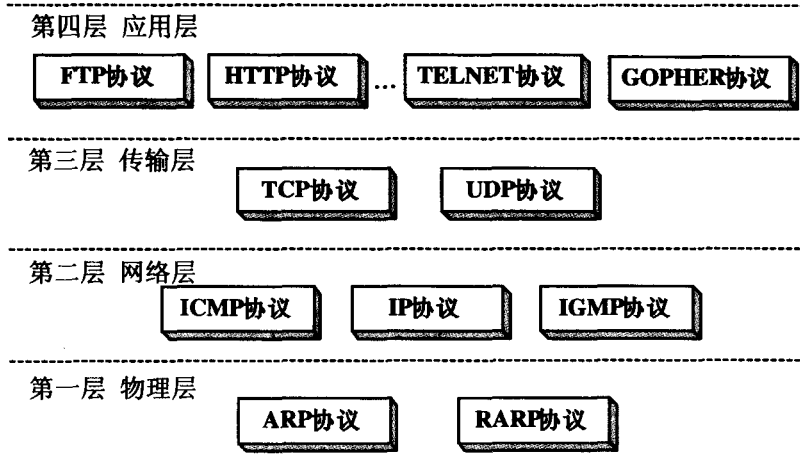
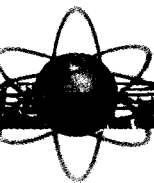


图 1-6 TCP/IP 协议簇示意图

下面我们大致介绍一下以上的各个协议：

**ARP** 即地址解析协议，ARP 的任务是把 IP 地址转化成物理地址，因为在局域网或广域网上把报文从一台机器发送到另一台机器，如果不知道目的机器的物理地址就会发生问题。需要一些方法能把 IP 地址(应用提供)解析为和网络相连的机器的硬件物理地址。ARP 就是把 IP 地址转换成相应物理地址的转换表。这个表称为 ARP 表。

**RARP** 即逆地址解析协议，RARP 以与 ARP 相反的方式工作。RARP 发出要反向解析的物理地址并希望返回其 IP 地址，应答包括由能够提供信息的 RARP 服务器发出的 IP 地址。虽然发送方发出的是广播信息，RARP 规定只有 RARP 服务器能产生应答。许多网络指定多个 RARP 服务器，这样做既是为了平衡负载也是为了作为出现问题时的备份。RARP 的用处在于：假如一个设备不知道它自己的 IP 地址，就没有办法产生 ARP 请求和 ARP 应答，这种情况下就必须使用 RARP 了。

**IP 协议**（网络协议）已经成为世界上最重要的网络协议，IP 协议定义了一个数据报的格式，所有在网络上要传输的数据最后都是以 IP 数据报的形式进行传输的。在网络上时时刻刻像流水一样流动的都是一个个的 IP 数据报。

**ICMP 协议**，又称 Internet 控制报文协议。它用于传递差错报文以及其他一些需要注意的信息。通常在发送报文失败时，错误发生处会向报文发送方发送一个 ICMP 报文，通过分析这个 ICMP 报文就可以知道发送失败的原因了。

**IGMP 协议**，即 Internet 组管理协议，它让物理网络上的所有系统知道主机当前所在的多播组。多播路由器需要这些信息以确定多播数据报应该向哪些端口转发。

**TCP 和 UDP**，即传输控制协议和用户数据报协议。TCP 是传输层协议( OSI 参考模型中第四层)，它使用 IP，提供可靠的应用数据传输。TCP 在两个或多个主机之间建立面向连接的通信。TCP 支持多数据流操作，提供流控制和错误控制，甚至完成对乱序到达报文的重新排序。用户数据报协议 UDP 是 IP 的另一个主机到主机层协议(对应于 OSI 参考模型的传输层)。UDP 提供了一种基本的、低延时的称为数据报的传输。UDP 被设计成一个有效的和最小的传输协议。这一点直接反映在其头结构中。它只包括用于转发数据报至合适应用(端口号)的