面向软件工程的 Visual C++ 网络程序开发

孙晓刚　韩　东
贾宗宣　黄新才　等 编著
博嘉科技　审

清华大学出版社

面向软件工程的
Visual C++网络程序开发

孙晓刚　韩　东
贾宗宣　黄新才
等 编著

博嘉科技 审

清 华 大 学 出 版 社
北 京

内容简介

近几年来，Internet 在我国日益普及，用户对网络应用的需求也不断增长。Visual C ++ 以其强大、高效的特点，成为开发 Windows 网络应用程序的首选开发工具。本书除了向读者介绍开发 Windows 网络程序的基本技术——Winsock、WinInet、COM 等外，还突出了从实际出发的特点，从软件工程的角度展示了如何分析、设计、开发一个功能完善、细节众多的网络应用系统，本书的最后还介绍了关于组织测试和制作安装程序的内容。整本书形成一个从技术基础到设计、开发，进而到测试、发布的体系。另外，书中有实例的章节都附有源代码，以方便读者学习和研究之用。

本书语言通俗易懂，内容丰富翔实，突出了实例，处处体现了软件工程的思想，适合有一定 Windows 平台下的 C 和C ++ 语言编程经验的高等院校计算机及其相关专业学生、网络应用程序开发人员及网络编程爱好者学习使用，同时也可作为从事 Windows 网络应用程序开发的软件工程师参考用书。

图书在版编目(CIP)数据

面向软件工程的 Visual C ++ 网络程序开发/孙晓刚等编著. —北京：清华大学出版社，2004.11
ISBN 7-302-09692-9

I. 面…　II. 孙…　III. C 语言—程序设计　IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 106566 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社总机：010-62770175　　**客户服务**：010-62776969
责任编辑：冯志强
印 刷 者：世界知识印刷厂
装 订 者：三河市金元装订厂
发 行 者：新华书店总店北京发行所
开　　本：185 × 260　**印张**：14.75　**字数**：367 千字
版　　次：2004 年 11 月第 1 版　2004 年 11 月第 1 次印刷
书　　号：ISBN 7-302-09692-9/TP · 6711
印　　数：1~3000
定　　价：24.00 元(附光盘)

前　言

计算机网络是计算机技术和通信技术密切结合的产物，当今世界正处于信息时代，计算机和通信网络是这一时代的“信息基础设施”。随着计算机网络技术的发展，基于 TCP/IP 协议的 Internet 正以它强大的异种机互联功能吸引着越来越多的国家、组织加入 Internet 互联的行列。在我国，Internet 用户数量呈几何级数的增长势头，CNNIC 公布的中国互联网络调查报告数字显示，截至 2003 年年底，中国互联网用户数量已经达到 7800 余万人。互联网行业正以它强大的生命力和巨大的用户市场吸引着众多的软件开发企业加入到网络应用研发的队伍中来。

Windows 系统是现在互联网用户主要使用的操作系统，它倡导的对用户友好的图形界面已成为用户普遍认可的人机交互界面，在众多的 Windows 应用程序开发工具中，Visual Studio 家族的 Visual C++以其高效、强大的优点，成为开发 Windows 网络应用程序的利器。在程序界流行这样一句话：“真正的程序员用 VC，聪明的程序员用 Delphi。”能使用 VC 开发网络应用程序需要一个长期的学习积累的过程，需要在掌握了 C++语言和 Windows SDK 及 MFC 的基础上，再进一步学习网络应用程序开发所关注的系统调用接口和通信协议。这个过程需要逐步积累的耐心，不可以急于求成。

现在市面上介绍 Visual C++网络应用程序开发的书籍，通常都是以介绍网络程序开发所需的技术基础为主，而缺乏对网络应用程序开发的总体把握。读者很可能有这样的体会：遇到一个有网络通信需求的系统开发任务时，往往不知如何将仅能完成数据收发的 Windows 网络编程接口和系统具体的通信需求结合起来。正因为如此，本书除了向读者介绍 Windows 网络编程最常用的方法之外，还着重介绍了两个具有相当规模的网络应用程序的例子，依照软件工程的思想指导对实际问题的分析，并利用了 UML(统一建模语言)这一面向对象软件工程的标准语言来表达系统的分析和设计。希望读者通过本书的阅读，不但能够掌握 VC 网络程序开发的基本方法，还能对分析、设计一个功能完善的网络应用系统有一定的认识和体会。希望通过本书的阅读，能对读者实际的工作有所帮助。

主要内容

本书共分为三部分。

第一部分是基础篇，它包括第 1~7 章。该部分主要介绍网络编程的基础知识(如 TCP/IP 原理、客户机/服务器模式)和开发 Windows 网络程序的常用方法，这些方法主要包括 Winsock API、WinInet 和 COM/DCOM 编程，而其中最常用的是 Winsock API 编程，这也是本部分的介绍重点。

第二部分是应用篇，它包括第 8~10 章，该部分是本书的特色之所在，内容涉及两个完整的网络应用系统的例子——远程控制系统和进销存管理系统的分析和设计的内容及系统关键技术的编码实现。读者阅读这部分时，应将重点放在系统的分析、协议的设计和关键技术的实现上面。

第三部分是软件的测试与发布，它包括第 11~12 章。该部分是本书软件工程思想的补充，介绍的内容包括软件测试的作业流程、程序的打包发布及软件说明书的制作。这部分的内容对开发一个网络应用系统而言是必不可少的。

本书特色

全书从介绍 TCP/IP 协议簇和客户机/服务器模式等基本概念入手，向读者全面而深入地介绍了利用 Winsock API、MFC Socket、WinInet 及 COM/DCOM 开发 Windows 网络应用程序的基本技术和方法。本书的特色在于密切结合软件工程思想介绍了两个实际的应用系统的设计开发过程，从需求分析、框架设计、协议设计到后续的详细设计和编码都有详细的论述，并配以 UML 图来表达分析和设计思想。两个系统的开发流程清晰，系统规模适中，不论从技术还是项目的难度上都具有一定层次，是对网络程序开发感兴趣的读者的难得的学习项目。

本书及配套光盘使用方法

本书中所有 VC 均是 Visual C++的简称。同时，书中列出的实例程序源代码只是该程序的部分核心 C++代码，读者可以从本书的附赠光盘中参考程序的完整源代码。

本书附赠一张配套光盘，光盘上的内容包括第 4~7 章和第 10 章的程序的源代码，它们都按照章节目录和工程名称组织，详情请参阅光盘上的 readme.txt 文件。读者用 Visual Studio 6.0 或 Visual Studio.Net 都可以打开，所有附带的代码都在 Visual C++ 6.0 上调试通过。

适应对象

本书语言通俗易懂，内容丰富翔实，突出了实例，处处体现了软件工程的思想，适合有一定 Windows 平台下的 C 和 C++语言编程经验的高等院校计算机及其相关专业学生、网络应用程序开发人员及网络编程爱好者学习使用，同时也可作为从事 Windows 网络应用程序开发的软件工程师的参考用书。

编写分工

本书由博嘉科技资讯有限公司组织编写，由孙晓刚，韩东，贾宗宣，黄新财担任主要的编写工作。参与本书编写的人员还有王安贵、陈郭宜、程小英、谭小丽、卢丽娟、刘育志、吴淬砺、赵明星、贺洪俊、李小平、史利、张燕秋、周林英、黄茂英、李立、李小琼、李修华、田茂敏、苏萍、巫文斌、邹勤、粟德容、童芳、李中全、蒋敏、刘华菊、袁媛、李建康等，在此一并感谢。

技术支持

由于本书篇幅较大，涉及技术内容广泛，加之时间仓促，书中难免存在错误或疏漏之处，希望广大读者给予批评指正。

如果读者愿意参加“面向软件工程的 Visual C++网络程序开发”的学习与培训，或是在学习过程中发现问题，或有更好的建议，欢迎与我们联系。我们非常愿意随时同面向软件工程的 Visual C++网络程序开发的读者朋友保持经常的联系。我们的电子邮件地址是 bojia@bojia.net，网址是 http://www.bojia.net。我们将认真、负责地对待每一位读者的来函。

目　录

第1章 Socket编程基础

本章概述

TCP/IP协议是Internet的基础。协议就好比语言，它定义了不同主机之间通信的语义、语法和文法等一系列规则。Socket一般译为“套接字”，它是通用的网络编程接口，是开发通信程序的利器。本章概要性地介绍了TCP/IP协议和Socket的一些知识，它们是开发通信程序的基础。

1.1 TCP/IP协议

1.1.1 引言

TCP/IP协议是当今异种机互联的工业标准，它支持不同厂家、不同操作系统的计算机之间的通信。TCP/IP的起源可追述到20世纪60年代美国政府资助的一个分组交换计算机网研究项目——ARPNET，其研究目的之一是异种机之间的连接。

TCP/IP协议只是众多较为完善的网络协议的一种。其他成熟的网络协议，包括Xerox公司的XNS，DEC公司的DECNET和IBM的SNA等，这些协议虽不乏用户，但是它们的开发目的都是为了给同一厂家生产的计算机提供通信能力。例如，DEC的计算机只能和DECNET通信，IBM计算机只能和SNA通信。一般地，称它们为专有协议，它们在异种机互联方面能力较弱。TCP/IP协议弥补了此方面的不足，经过20多年的发展，已经成为用户和厂商共同认可的异种机互联的工业标准。

TCP/IP协议的第一个实现方式出现在1983年，由加利福尼亚大学计算机系统研究小组完成，它被包含在UNIX操作系统的一个版本——4.2 BSD中。UNIX BSD提供了一种应用程序访问通信协议的系统调用接口——Socket，后来称为Berkeley Socket，以表明它和Berkeley版UNIX的渊源。需要说明的是，TCP/IP协议不是单个协议，而是一组协议，它们用分层的方式开发，以协作方式完成在具体物理介质上的通信功能。在这众多协议中，以TCP(Transmission Control Protocol，传输控制协议)和IP(Internet Protocol，网际协议)最为著名。关于TCP/IP协议体系结构和分层的知识，在后续内容中介绍。

1.1.2 TCP/IP的体系结构

TCP/IP协议族的分层结构如图1-1所示。

可以看到，整个TCP/IP协议族分为4层：

- 链路层：包括系统中的设备驱动程序和网络接口卡，它们和具体的物理媒介(如电

缆)一起，向高层提供物理链路。

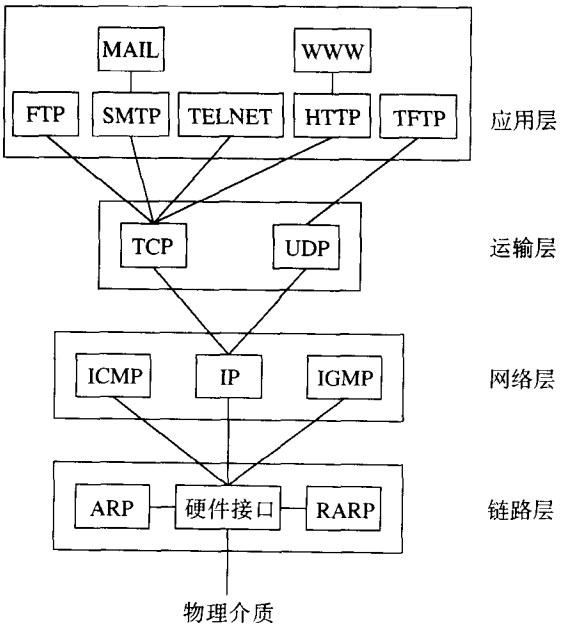

图 1-1 TCP/IP 的分层和协议体系结构图

- 网络层：网络层的主要任务是做出数据分组的路由选择。它由 IP 协议和 ICMP、IGMP 协议组成。
- 运输层：运输层为相互通信的主机提供端到端的通信能力。其中，TCP 协议向应用层提供可靠的数据连接，它保证进程间数据传输的正确、有序和不重复。UDP 协议仅仅为应用层提供数据报的分组发送服务，数据传输的可靠性只能通过应用层来保证。
- 应用层：负责处理具体的 TCP/IP 协议应用的细节。常见的应用包括 Telnet(远程登录)、FTP(文件传输协议)和 SMTP(简单邮件传输协议)。

在开发通信系统时，关心的主要是网络层的 IP 协议和运输层的 TCP 和 UDP 协议。在讨论协议细节之前，先介绍 IP 地址和端口的概念。

1.1.3 IP 地址和通信端口

1. IP 地址

IP 地址的全称是互联网地址，它是惟一标识的网络接口，即一台主机(但一台主机可能拥有多个 IP 地址，通常说这样的主机是多接口的)。IP 地址是一个 32 位二进制数，整个 IP 地址空间被分为 5 类地址，用 A、B、C、D、E 表示。通常关心的主要是 A、B、C 三类地址，如图 1-2 所示。这样的 IP 地址由网络号和主机号组成。对于一个大型的网络，有时还要根据需要建立若干子网，这样，原来 A、B、C 类 IP 地址中的主机地址实际上被分成了

“子网号+主机号”的形式。子网掩码可以用来将 32 位的 IP 地址分开成为主机地址和含子网的网络地址(网络地址+子网地址)。关于子网的概念，请参阅有关的文献。

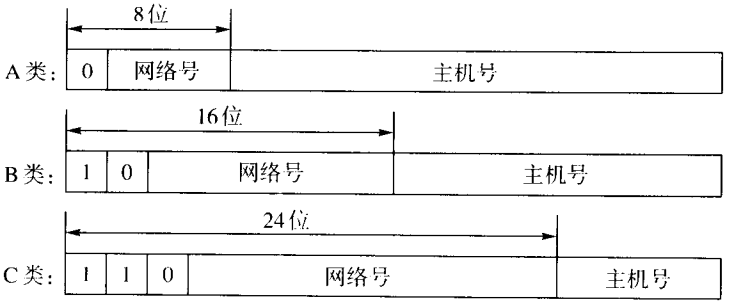

图 1-2　A、B、C 三类 IP 地址

IP 地址有三类：单播地址、组播地址和广播地址。广播地址是主机地址全为 0 或者 1 的 IP 地址，顾名思义，使用广播地址可以向网络内的所有主机发送数据。D 类地址是组播地址，它指定一个主机组，该主机组的成员可以遍布整个 Internet 网络，而不像广播地址仅仅局限于物理网络。

在实际应用中，习惯上把 32 位 IP 地址分成 4 个 8 位的十进制整数表示的形式。这种表示法被称为“点分十进制表示法(dotted decimal notation)”。在 Internet 上看到的 IP 地址都是这样表示的，例如，www.sohu.com 的 IP 地址是 61.135.150.71，它是一个 A 类地址。

Internet NIC(Internet Network Information Center)负责向接入互联网的网络分配网络地址，而具体主机的 IP 地址由网络管理员依照某种策略进行。

2. 端口号

主机之间的通信实际上是主机之间进程的通信。正如前面所述，IP 地址可以惟一地确定互联网上的主机，而在系统内部，为了区分不同的进程，TCP/IP 协议引入了协议端口(Protocol Port，简称端口号)的概念，用它来表示主机内的不同进程。

端口是一种抽象的软件结构(包括定义的一些数据结构和 I/O 缓冲区)，进程通过系统调用和端口绑定后，对通信端口的操作类似于对本地文件的 I/O 操作。可以说，端口是进程和运输层之间 I/O 操作的桥梁。TCP/IP 协议用 16 位二进制数(一个 WORD)表示端口号，因此，理论上系统可以分配利用的端口号共有 2^{16}=65536 个。端口号由 IANA(Internet Assigned Number Authority)控制和分配，65536 个端口划分成三类：

- 知名端口(well-known ports)：0~1023，它们由 IANA 分配，为固定服务保留。如大家众所周知的 FTP 协议的端口是 20，21，HTTP 协议的端口是 80，等等。详情可以参见 RFC1700。
- 注册端口(registered ports)：1024~49151，它们供系统内的普通应用程序和进程使用。
- 动态端口(dynamic ports)：49152~65535，又称私有端口(private ports)，为客户通信进程和程序动态、临时使用。

普通应用程序应使用 1024~49151 的端口号，以避免和系统固定服务端口的冲突。关于

进程和端口绑定的系统调用，在 Socket 编程部分将详细介绍。

1.1.4 协议功能

从开发通信系统的角度出发，仅在此介绍 IP、TCP 和 UDP 协议的知识。事实上，通信协议是网络操作系统实现的一部分，通信的细节问题用户不必关心。用户只能感知到本地通信进程的对等端的存在，而对数据包的分组、路由、物理链路等通信细节毫不知晓，这很像平时通电话的情形。

1. IP 协议

简单地说，IP 协议的功能就是“尽力地”将数据从源端发送到目的端，不论具体的物理网络还是以太网、令牌环网或者是高速的 ATM 网络都是为了保证数据能从源端到目的端。IP 协议提供了下列服务：

- 寻址
- 路由选择
- 数据报的分段和重组

IP 协议以 IP 报文格式传输高层提交的数据，它为高层的数据传输提供无连接的服务。在发送数据之前，它不在通信端之间建立连接，因此 IP 协议不能保证传输的可靠性。IP 协议简单地对上层需要发送的数据进行分组(如果必要的话)、封装并发送，数据传输的正确性和可靠性需要协议族的高层协议提供的服务来保证(如 TCP 提供的差错控制服务)。数据到达目的端后，IP 目的端也没有相应的确认机制，仅仅是把数据包重组后交给高层协议处理。

2. TCP 协议

TCP 协议是运输层协议，它提供面向连接的、可靠的、端到端的字节流服务。TCP 协议主要提供下列功能：

- 建立、维持和终止两个进程间的连接
- 待确认应答机制的数据分段传输
- 保证传输的顺序性
- 差错控制
- 流量控制
- 全双工的数据传输

每次 TCP 数据通信(会话)都要经过建立连接、传输数据、解除连接的过程。连接一旦建立，TCP 分段不必携带目的地址发送。TCP 协议提供的是面向字节流的服务。所谓字节流，就是 8 bit 的信息流，协议本身不在应用提交的数据中插入分隔标识，数据的格式需要由应用自己区分。例如，发送的一方依次提交了三个 256 字节的数据，TCP 协议可能出于效率的考虑将应用提交的数据封装到一个数据分段中发送。由于没有区分数据边界的措施，在接收端看来，它收到的是一个 768 字节的数据包。

TCP 协议可以保证端到端数据传输的正确、有序和可靠。这是因为，TCP 报文头提供

了一系列的字段来控制流式的字节传输。虽然 TCP 依赖无连接的 IP 协议传输数据，每个 TCP 分段到达目的端的路由可能不同，但在目的端，TCP 协议要负责收到的数据分段的处理，包括分段排序，重复分段的丢弃等。

3. UDP 协议

UDP 协议提供的是无连接的、不可靠的用户数据报传输服务。UDP 协议的报文头很简单，如图 1-3 所示。

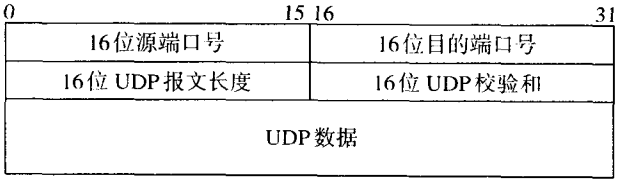

图 1-3　UDP 报文格式

从图中可见，UDP 报文没有任何机制保证所传输的数据能够到达目的地。UDP 报文可以保证用户数据的边界，因为 UDP 没有数据缓冲功能。实际应用中，数据传输的可靠性必须由应用程序实现。对于无连接的数据传输，通常采用超时重传机制来保证。

有的读者不禁要问，既然 IP 协议也提供了无连接的数据传输服务，那么，UDP 协议的作用是什么呢？由于 IP 协议只负责把数据从源主机传输到目的主机，而 UDP 协议提供了协议端口号，因此它的作用在于区分同一主机上的不同进程，提供复用的功能。

UDP 协议的另外一个作用是发送广播数据和组播数据包。TCP 协议再次数据传输时需要建立和维持连接，这对于需要发送广播数据或是组播数据的场合，额外开销太大，因此广播和组播应用一般都建立在 UDP 协议之上。

1.2　Socket 简介

Socket 的英文原意是“孔”或“插座”，中文译为“套接字”，是一种进程的通信机制。Socket 是可以被命名和寻址的通信端点，它用一个 Socket 号标识一个与之相联系的进程。Socket 向应用程序提供了统一的网络通信编程接口，屏蔽了下层的通信协议和物理介质的细节，降低了通信系统开发难度。

Socket 存在于通信域中。所谓通信域，就是利用 Socket 通信的进程的相关特性、信息的集合。TCP/IP 协议使用的是网际通信域(AF_INET)。

1.2.1　Socket 历史

Socket 最早出现在 BSD UNIX 操作系统中，是 BSD 4.X 版 UNIX 内核的一部分，一般称之为 Bekeley Socket。事实上，TCP/IP 编程存在两套编程接口：Socket 和 TLI(Transport Layer Interface)。由于众多制造商，如 Sun Microsystem，Tektronix 等都支持 Socket，使得

Socket 得到了广泛的应用，至今已成为网络程序设计的标准。

Socket 最初并不支持 DOS 和 Microsoft Windows 平台。Socket 在 UNIX 平台下的成功使人们产生了将它移植到 DOS 和 Windows 环境的设想。1992 年，制定出了 Windows Socket 规范 1.0 版(WinSock 1.0)，随后又推出了 1.1 版(WinSock 1.1)。Windows Socket 不单实现了 Bekeley Socket 全部功能，而且在其基础上做了重要扩充。这些扩充主要是提供了一些异步的 Socket 函数，并增加了对网络事件的异步选择机制，使之更符合 Windows 消息驱动的特性。随着 32 位 Windows 的出现和其不断发展，使原来的 16 位 Windows Socket 规范不能满足需要。从 1994 年 5 月开始，Microsoft，Sun Microsystem 等几家公司组成的 Windows Socket 工作小组开始制定 Windows Socket 2 规范(简称 WinSock 2)，并于 1997 年发布了 WinSock 2.2.1 版。

WinSock 2 是 WinSock 1.1 的发展和扩充，它保持了和原先 Windows Socket 的兼容能力。WinSock 2 的主要特点在于其协议无关性，应用程序可以根据需要安装多种不同的通信协议；另外，WinSock 2 在数据传输能力方面也有重大提高，使之更适合网络多媒体和实时应用的场合。考虑到本书的定位，以论述现今广泛应用的 WinSock 1.1 为主，适当介绍 WinSock 2 的内容。

1.2.2 Socket 的类型

TCP/IP 协议使用的套接字分为三类，它们是：流式套接字(SOCK_STREAM)、数据报套接字(SOCK_DGREAM)和原始套接字(SOCK_RAW)。

1. 流式套接字(SOCK_STREAM)

流式套接字向应用提供面向连接的、可靠的通信流，数据发送无差错、无重复且保证接收到数据的顺序性。

流式套接字最典型的应用是 BBS 和 HTTP 服务。以 BBS 应用为例，用户通过 TELNET 协议和服务器建立连接，在浏览论坛帖子的交互过程中，数据总是顺序地在源端和目的端之间传输的。

流式套接字的通信能力是由 TCP 协议保证的。通过前面的介绍，可以知道 TCP 提供的是端到端的可靠的、面向连接的数据流服务，尽管它建立在无连接的 IP 协议(其可靠性和面向连接的协议相比较差)之上。

2. 数据报套接字(SOCK_DGRAM)

数据报套接字向应用提供的是无连接的服务。数据以报文的形式发送，且不能保证数据的无差错和顺序到达。使用数据报套接字的例子包括 DNS(Domain Name Service，域名服务)和 NFS(网络文件系统)等。

数据报套接字的一个显著优点在于它的开销较小，适合于网络广播和组播等需要一对多发送数据的场合和频繁交互的应用(如 DNS)。数据包套接字依赖 UDP 协议提供服务，这是它具有上述属性的原因。