

21世纪高等学校规划教材 | 计算机科学与技术



C#网络程序开发

(第二版)

何波 傅由甲 主编



清华大学出版社

非外借

C#网络程序开发

(第二版)

何波 傅由甲 主编

清华大学出版社
北京



内 容 简 介

本书是 C# 网络程序开发的经典教材,涵盖 C# 网络程序开发的理论、实验和课程设计。全书共包含五大部分:第一部分为 C# 网络程序开发基础,包括网络程序开发基础知识和 C# 网络程序开发基础;第二部分为 C# 网络传输程序开发,包括 TCP、UDP 和 P2P 网络程序开发技术;第三部分为 Internet 应用程序开发,包括 FTP、电子邮件、HTTP 和 Web Service 网络程序开发技术;第四部分为 C# 网络程序开发实践,包括 TCP、UDP、P2P、FTP、电子邮件、HTTP 和 Web Service 网络程序开发实践;第五部分为 C# 网络程序开发课程设计,包括课程设计目的、题目及要求、考核方式。

本书遵循由浅入深、逐步深入的原则,实验以理论课课程为基础进行扩展和提高,有利于激发学习兴趣,增强学习的自信心和成就感,进而牢固地掌握网络程序开发技术。本书提供了配套的课件和源程序。

本书可作为高等院校计算机及相关专业的教材,也可作为信息技术领域教师、学生和工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C# 网络程序开发/何波等主编. —2版. —北京:清华大学出版社,2019
(21 世纪高等学校规划教材·计算机科学与技术)
ISBN 978-7-302-50676-8

I. ①C… II. ①何… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 161082 号

责任编辑:闫红梅 王冰飞

封面设计:傅瑞学

责任校对:梁 毅

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:13.25

版 次:2014 年 5 月第 1 版 2019 年 1 月第 2 版

印 数:1~1500

定 价:39.00 元



字 数: 印 次:2019 年 1 月第 1 次印刷

产品编号:079863-01

清华大学出版社
TSINGHUA UNIVERSITY PRESS

前言

随着计算机网络的日益普及,各种计算机网络程序已深入到大众生活的方方面面,使得网络程序开发成为程序开发的一个非常重要的方向,近年来很多高校也都开设了网络程序开发的相关课程。Visual C#是微软公司在吸取Java和C++优点的基础上研发的面向对象的程序设计语言,非常适合各种网络应用程序的开发。由于PC用户90%以上仍使用Windows操作系统,因此Visual C#成为PC上网络软件开发的首选。

本书涵盖C#网络程序开发的理论、实验和课程设计。

全书共包含五大部分,内容如下。

第一部分 C#网络程序开发基础(包括网络程序开发基础知识和C#网络程序开发基础)。

第二部分 C#网络传输程序开发(包括TCP、UDP和P2P网络程序开发技术)。

第三部分 C#Internet应用程序开发(包括FTP、电子邮件、HTTP和Web Service网络程序开发技术)。

第四部分 C#网络程序开发实践(包括TCP、UDP、P2P、FTP、电子邮件、HTTP和Web Service网络程序开发实践)。

第五部分 C#网络程序开发课程设计(包括课程设计目的、题目及要求、考核方式)。

本书由重庆理工大学何波、傅由甲主编。

本书可作为高等院校计算机及相关专业的教材,也可作为信息技术领域教师、学生和工程技术人员参考书。

在本书的编写过程中参考了国内外相关教材和著作,在此向文献作者表示真诚的感谢。由于编者水平有限,书中难免存在错误或不妥之处,恳请读者批评指正。

编者

2018年9月

目 录

第一部分 C#网络程序开发基础

第 1 章 C#网络程序开发基础知识	3
1.1 网络通信模型及分类	3
1.1.1 分散式、集中式与分布式系统	3
1.1.2 C/S、B/S 与 P2P 网络通信架构	4
1.2 TCP/IP 网络模型及协议	5
1.2.1 TCP/IP 网络架构	5
1.2.2 TCP 协议和 UDP 协议	7
1.2.3 IP 协议	7
1.3 网络程序通信机制	8
1.3.1 端口与套接字	8
1.3.2 基于套接字的网络进程通信机制	8
第 2 章 C#网络程序开发基础	10
2.1 Visual Studio .NET 集成开发环境概述	10
2.2 开发环境的安装与常见 C#网络编程简介	11
2.2.1 开发环境的安装	11
2.2.2 C#网络编程简介	14
2.3 C#.NET 网络程序开发基本类	14
2.3.1 IPAddress 类	15
2.3.2 IPEndPoint 类	15
2.3.3 IPHostEntry 类	15
2.3.4 Dns 类	16
2.3.5 Ping 及相关类	17
2.4 C#套接字与网络流	18
2.4.1 Socket 类	18
2.4.2 套接字的类型与使用方法	19
2.4.3 网络流	24
2.4.4 网络数据编码与解码	26
2.5 多线程技术	31
2.5.1 多线程概述	31
2.5.2 多线程的创建与使用	31
2.5.3 多线程的同步	39

2.5.4 线程池的概念与使用方法	40
-------------------------	----

第二部分 C#网络传输程序开发

第3章 TCP网络程序开发	45
---------------------	----

3.1 TCP程序开发主要技术	45
3.1.1 使用套接字进行TCP传输	45
3.1.2 使用TCP类进行网络传输	47
3.1.3 同步与异步	52
3.2 基于同步TCP的网络聊天程序开发	70
3.2.1 功能介绍及界面设计	70
3.2.2 服务器程序编写	72
3.2.3 客户端程序编写	78
3.3 基于异步TCP的网络聊天程序开发	82
3.3.1 异步程序编程方法	83
3.3.2 界面设计	89
3.3.3 服务器端程序编写	89
3.3.4 客户端程序编写	90

第4章 UDP网络程序开发	91
---------------------	----

4.1 UDP程序开发的主要技术	91
4.1.1 UDP与TCP的区别与优势	91
4.1.2 使用UDP类进行网络传输	92
4.1.3 UDP下的同步与异步通信	93
4.2 UDP的广播与组播程序开发	95
4.2.1 广播与组播的基本概念	96
4.2.2 组播组的加入与退出	96
4.3 基于广播和组播的网络会议程序开发	97
4.3.1 功能介绍及页面设计	97
4.3.2 程序实现代码	98

第5章 P2P网络程序开发	103
---------------------	-----

5.1 P2P基础知识	103
5.1.1 P2P的发展历程	103
5.1.2 P2P的架构	103
5.1.3 P2P通信步骤	105
5.2 .NET下的P2P程序开发	105
5.2.1 对等名称解析协议	105
5.2.2 PeerToPeer命名空间	107
5.3 P2P资源注册与发现程序开发	110

5.3.1	P2P 资源发现过程	110
5.3.2	P2P 资源注册程序开发	111
5.3.3	P2P 资源发现程序开发	115

第三部分 Internet 应用程序开发

第 6 章	FTP 网络程序开发	121
6.1	FTP 原理及规范	121
6.1.1	FTP 概述	121
6.1.2	FTP 工作原理和数据传输	121
6.1.3	FTP 规范	123
6.2	FTP 程序开发相关类	125
6.2.1	FtpWebRequest 类	125
6.2.2	FtpWebResponse 类	127
6.2.3	NetworkCredential 类	127
6.3	编写 FTP 的文件上传下载器	128
6.3.1	FTP 服务器的配置	128
6.3.2	功能介绍及界面设计	130
6.3.3	上传文件程序开发实现	132
6.3.4	下载文件程序开发实现	133
第 7 章	SMTP 与 POP3 网络程序开发技术	135
7.1	邮件发送与接收协议	135
7.1.1	邮件发送与 SMTP 协议	135
7.1.2	邮件接收与 POP3 协议	136
7.1.3	.NET 下的邮件收发相关类	137
7.2	邮件客户端程序开发实现	140
7.2.1	功能介绍及页面设计	140
7.2.2	邮件发送模块程序开发实现	141
7.2.3	邮件接收模块程序开发实现	143
第 8 章	基于 HTTP 的 Web 程序开发技术	150
8.1	HTTP 简介	150
8.1.1	HTTP 工作原理及特点	150
8.1.2	HTTP 协议	151
8.2	.NET 下的 HTTP 程序开发技术	155
8.2.1	HTTP 程序开发相关类	155
8.2.2	Web 中的数据提交	156
8.2.3	Web 数据交换举例	157
8.3	编写 HTTP 下的多线程文件下载器	161

8.3.1	网络资源有效性检测	161
8.3.2	使用多线程下载文件	162
第9章 Web Service 程序开发技术		171
9.1	Web Service 技术概述	171
9.1.1	Web Service 基本概念	171
9.1.2	Web Service 的优势与短处	171
9.1.3	Web Service 的架构	172
9.2	创建和使用 Web 服务	173
9.2.1	创建 Web 服务	173
9.2.2	调用 Web 服务	175
9.3	Web Service 实用程序开发举例	177
9.3.1	使用 Web Service 编写天气预报程序	177
9.3.2	使用 Web Service 查询股票行情	183

第四部分 C#网络程序开发实践

实验一	C#网络程序开发基础——使用多线程扫描主机及端口	187
实验二	TCP 程序开发实践——C/S 模式的局域网聊天程序开发	188
实验三	UDP 程序开发实践——局域网视频聊天程序开发	189
实验四	P2P 程序开发实践——双人对战五子棋	190
实验五	FTP 程序开发实践——编写自己的 FTP 服务器	191
实验六	电子邮件程序开发实践——电子邮件客户端	192
实验七	HTTP 程序开发实践——编写自己的简单 Web 浏览器	193
实验八	Web Service 程序开发实践——学生网络选课管理程序	194

第五部分 C#网络程序开发课程设计

参考文献	204
------	-----

第一部分

C#网络程序开发基础

第1章 C#网络程序开发基础知识

第2章 C#网络程序开发基础

C#网络程序开发基础知识

1.1 网络通信模型及分类

1.1.1 分散式、集中式与分布式系统

为了网络通信的需要,人们经常需要对网络通信模型做一定的分析,为此,提出了各种参考模型。随着科技的进步,网络技术高速发展,网络的通信模型也在不断变化,但总体目标向着“简单明了”和“实用化”的方向发展。根据数据的通信方式,可以将通信模型分为分散式(Decentralized)系统、集中式(Centralized)系统和分布式(Distributed)系统三大类。

1. 分散式系统

在分散式系统中,用户只需负责管理自己的计算机系统,各自独立的系统之间没有资源或信息的交换或共享,由此引起大量共享数据的重复存储,造成数据冗余,容易导致共享的不同用户之间数据的不一致性,同时造成硬件的运营维护等成本大量增加。

2. 集中式系统

在集中式系统中,通过一台主计算机保存共享的全部数据,用户通过终端连接到这台主计算机进行数据访问。终端包含键盘和显示器,使用通信链路接收和发送数据。

集中式系统的优点是资源集中,硬件成本低,数据共享访问方便,减少或消除了数据的冗余与不一致。但它的主要缺点是可靠性不如分散式系统,一旦主机出现故障,整个系统都会瘫痪;另外,由于系统为所有用户共享,无法满足特殊用户的计算需要,系统响应较慢。

3. 分布式系统

分布式系统是集中式系统和分散式系统的混合,由多个连接起来的独立计算机组成。与计算机网络相比,分布式系统的资源以透明的形式供给用户使用,用户在使用资源时无须知道该资源是本地的还是远程的,对于远程资源也可以像本地资源一样任意调用,而计算机网络则需要先知道资源的位置,与资源所在的主机建立连接后才能使用;此外,分布式系统还具有高度的内聚性,每个数据库分布节点高度自治,有本地的数据库管理系统。分布式系

统的著名例子是万维网(World Wide Web),在万维网中,所有的 Web 看起来就好像是放在一个主机上一样。

当然,分布式系统和计算机网络还是有相通的地方,多数分布式系统的建立是以计算机网络为基础的,所以分布式系统与计算机网络在物理结构上基本相同,它们的区别主要在软件层面。

1.1.2 C/S、B/S 与 P2P 网络通信架构

1. C/S 模型

C/S(Client/Server)模型也叫作 C/S 结构,即客户机/服务器结构,它是在分散式系统、集中式系统和分布式系统的基础之上发展出来的,当前的大多数通信网络都是这种模型。

C/S 模型将一个网络事务处理分为两部分,一部分是客户端(Client),主要负责界面和处理业务逻辑,并为用户提供网络请求服务的接口,如数据查询请求;一部分是服务器端(Server),一般以数据处理能力较强的数据库管理系统作为后台,负责接收和处理用户对服务的请求,并将这些服务透明地提供给用户。C/S 结构一般采用两层结构,如图 1-1 所示。

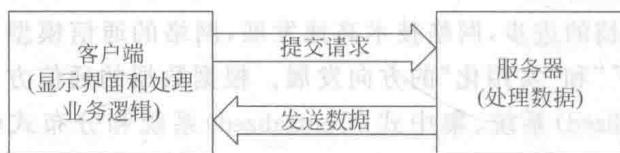


图 1-1 C/S 结构工作示意图

从程序实现角度来说,客户端和服务端间的通信先由服务器端启动 Server 进程,然后等待客户端的请求服务;客户端启动 Client 进程向服务器申请服务。服务器处理完一个客户端请求信息后再继续等待其他客户端的请求,周而复始地以这样一种方式进行。

在这种结构中,服务器硬件需要足够强的处理能力,才能满足客户的要求。

C/S 结构的技术较为成熟,其特点是交互性强,具有安全的存取模式,网络通信量低,响应速度快,利于处理大量的数据,可以充分利用两端硬件环境的优势,将任务合理分配到客户端和服务端来实现,既适用于实际应用程序,又适用于统一的计算和处理。但是它也有缺点,即该结构的程序为针对性开发,不能灵活变更,维护和管理的难度比较大,通常只局限于小型局域网,不利于扩展。

2. B/S 模型

B/S(Browse/Server)模型即浏览器/服务器模式,也叫 B/S 结构。它只安装维护一个服务器(Server),而客户端采用浏览器(Browse)运行软件。B/S 结构是随着 Internet 技术的兴起,对 C/S 结构的变化和改进。它和 C/S 并没有本质区别,是 C/S 模型的一种特例,特殊在于这种模型必须使用 HTTP(Hypertext Transfer Protocol,超文本传送协议)。

B/S 结构采用的是三层客户机/服务器结构,在数据管理层(Server)和用户界面层(Client)增加了一层结构,称为中间件(Middleware),使整个体系分为三层。三层结构是伴随着中间件技术的成熟而兴起的,核心概念是利用中间件将应用分别表示为界面层、业务逻

辑层和数据存储层 3 个不同的处理层,如图 1-2 所示。

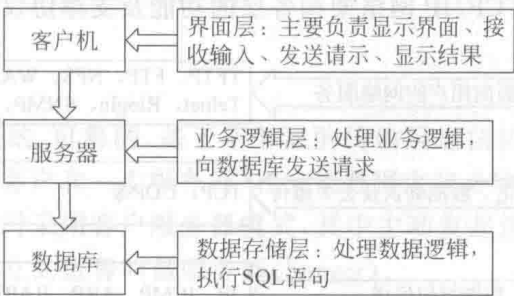


图 1-2 B/S 结构工作示意图

中间件作为构造三层结构的基础平台,具有如下主要功能:负责客户机与服务器、服务器与数据库之间的连接和通信;实现应用与数据库之间的高效连接。具有中间件的三层结构在层与层之间相互独立,任一层的改变都不会影响其他层的功能。

在 B/S 体系结构系统中,用户通过浏览器向分布在网络上的许多服务器发出请求,服务器对浏览器的请求进行处理,将用户所需信息返回到浏览器。而其余的工作,如数据请求、加工、结果返回以及动态网页生成、对数据库的访问和应用程序的执行等,全部由服务器完成。可以看出,B/S 结构相对于 C/S 结构是一个非常大的进步。

B/S 结构的主要特点是分布性强、维护方便、开发简单且共享性强,如一台计算机可以访问任意一个 Web 服务器,用户只需要知道服务器的网址即可访问,不需要针对不同服务器分别提供专门的客户端软件。但 B/S 结构的缺点在于数据存在安全性问题,对服务器要求过高,数据传输慢,软件个性化特点明显降低,而且实现复杂的应用构造有较大困难。

综上所述,两种模式各有利弊。C/S 结构适用于特定范围,如局域网;而 B/S 结构则可以弥补 C/S 结构在应用平台上的功能不足。从可扩展性和高灵活性显示,B/S 结构将是未来的发展方向。

3. P2P 模型

P2P(Peer-to-Peer)称为对等互连模型。在此环境中,网络上的各台主机具有相同的功能,无主从之分,任一台计算机都是既可当服务器,设定共享资源供网络中其他计算机使用,又可作为工作站。从程序实现来说,一个应用程序同时起到客户端和服务器的作用。目前,它是小型局域网常用的组网方式,其优点是配置容易,通信便利,成本低;缺点是可靠性不如 C/S 模型,易遭黑客攻击。

1.2 TCP/IP 网络模型及协议

1.2.1 TCP/IP 网络架构

TCP/IP 网络架构也称为 TCP/IP(Transmission Control Protocol/Internet Protocol, 传输控制协议/网际协议)参考模型。它是目前全球互联网工作的基础,该架构将网络功能

从上至下划分为：应用层、传输层、网际层和网络接口层，每一层的功能由一系列网络协议进行体现，图 1-3 给出了 TCP/IP 网络架构各层的功能及支撑协议。

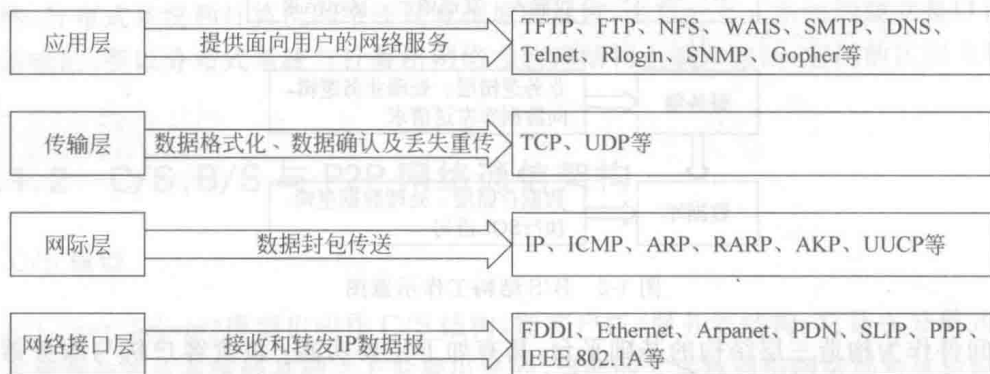


图 1-3 TCP/IP 网络架构各层的功能及支撑协议

TCP/IP 网络架构采用自顶而下的分层结构，每一层都需要下一层所提供的服务来满足自己的需求，本层协议生成的数据封装在下一层协议的数据中进行传输，因此各层间的协议有依赖关系。下面简单介绍一下 TCP/IP 模型各层的主要功能。

(1) 应用层：即最高层，提供面向用户的网络服务，负责应用程序之间的沟通，主要协议有简单邮件传输协议(SMTP)、文件传输协议(FTP)、超文本传输协议(HTTP)、域名系统(DNS)、网络远程访问协议(Telnet)等。

Socket 支持多个应用程序间基本的消息传递功能，通过遵循应用层上的某一种或几种协议的规范，使应用程序完成用户需要的相应功能，这是本书网络应用程序开发的目的。

(2) 传输层：位于第 3 层，完成多台主机间的通信，提供节点间的数据传送及应用程序间的通信服务，也称为“端到端”通信，通过在通信的实体间建立一条逻辑链路，屏蔽了 IP 层的路由选择和物理网络细节。传输层的功能主要是数据格式化、数据确认及丢失重传等。该层协议有传输控制协议(TCP)和用户数据报协议(UDP)，提供不同的通信质量和需求的服务。

(3) 网际层：位于第 2 层，也称为网络互联层或 Internet 层，由于该层最重要的协议是 IP 协议，所以也称为 IP 层。该层负责提供基本的数据封包传送功能，在它上面传输的数据单元叫 IP 数据报，或 IP 分组。网际层让每个 IP 数据报都能够到达目的主机，但是它不检查数据报是否被正确接收。

网络层的本质是使用 IP 将各种不同的物理网络互联，组成一个传输 IP 数据报的虚拟网络，实现不同网络的互联功能，该层协议除了 IP 协议外，还有 Internet 控制报文协议(ICMP)和 Internet 组管理协议(IGMP)。

(4) 网络接口层：该层位于协议架构的最底层，负责接收 IP 数据报并发送到其下的物理网络，或从网络上接收物理帧，抽取 IP 数据报转交给网际层。这里的物理网络指各种实际传输数据的局域网或广域网。

1.2.2 TCP 协议和 UDP 协议

1. TCP

TCP 是一种面向连接的、可靠的、基于字节流的传输层通信协议。面向连接意味着两个使用 TCP 的进程(一个客户和一个服务器)在交换数据之前必须先建立好连接,然后才能开始传输数据。建立连接时采用客户服务器模式,其中主动发起连接建立的进程叫做客户(Client),被动等待连接建立的进程叫做服务器(Server)。

TCP 提供全双工的数据传输服务,这意味着建立了 TCP 连接的主机双方可以同时发送和接收数据。这样,接收方收到发送方消息后的确认可以在反方向的数据流中进行捎带。“端到端”的 TCP 通信意味着 TCP 连接发生在两个进程之间,一个进程发送数据,只有一个接收方,因此 TCP 不支持广播和组播。

TCP 连接面向字节流,字节流意味着用户数据没有边界,例如,发送进程在 TCP 连接上发送了 2 个 512 字节的数据,接收方接收到的可能是 2 个 512 字节的数据,也可能是 1 个 1024 字节的数据。因此,接收方若要正确检测数据的边界,必须由发送方和接收方共同约定,并且在用户进程中按这些约定来实现。

TCP 接收到数据包后,将信息送到更高层的应用程序,如 FTP 的服务程序和客户程序。应用程序处理后,再轮流将信息送回传输层,传输层再将它们向下传送到网际层,最后到接收方。

2. UDP

UDP 与 TCP 位于同一层,但与 TCP 不同,UDP 协议提供的是一种无连接的、不可靠的传输层协议,只提供有限的差错检验功能。它在 IP 层上附加了简单的多路复用功能,提供端到端的数据传输服务。设计 UDP 的目的是为了以最小的开销在可靠的或者是对数据可靠性要求不高的环境中进行通信,由于无连接,UDP 支持广播和组播,这在多媒体应用中是非常有用的。

1.2.3 IP 协议

IP(网际)协议是 TCP/IP 模型的核心,也是网络层最重要的协议。

网际层接收来自网络接口层的数据包,并将数据包发送到传输层;相反,也将传输层的数据包传送到网络接口层。IP 协议主要包括无连接数据报传送,数据报路由器选择以及差错处理等功能。

由于网络拥挤、网络故障等问题可能导致数据报无法顺利通过传输层。IP 协议具有有限的报错功能,不能有效处理数据报延迟,不按顺序到达和数据报出错,所以 IP 协议需要与另外的协议配套使用,包括地址解析协议 ARP、逆地址解析协议 RARP、因特网控制报文协议 ICMP、因特网组管理协议 IGMP 等。IP 数据包中含有源地址(发送它的主机地址)和目的地址(接收它的主机地址)。

IP 协议对于网络通信而言有着重要的意义。由于网络中的所有计算机都安装了 IP 软件,使得许许多多的局域网构成了庞大而严密的通信系统,才形成了如今的 Internet。其

实,Internet 并非一个真实存在的网络,而是一个虚拟网络,只不过是利用 IP 协议把世界上所有愿意接入 Internet 的计算机局域网络连接起来,使之能够相互通信。

1.3 网络程序通信机制

1.3.1 端口与套接字

1. 端口

主机之间的通信,看起来只要知道了 IP 地址就可以实现。其实不然,真正完成通信功能的不是两台计算机,而是两台计算机上的进程。IP 地址只能标识到某台主机,而不能标识计算机上的进程。如果要标识进程,完成通信,需要引入新的地址空间,这就是端口(port)。

端口目前有两种意义:一是指物理端口,比如 ADSL Modem、集线器、交换机、路由器上连接其他设备的接口,如 RJ-45 端口、SC 端口等;二是逻辑端口,即进程标识,如 HTTP 的 80 端口,FTP 的 21 端口等。本书所指的端口都是指逻辑端口。定义端口是为了解决与多个应用进程同时进行通信的问题。端口地址由两字节的二进制数表示。端口号范围从 0 到 65535。由于 TCP/IP 传输层的两个协议 TCP 和 UDP 是独立的两个软件模块,因此各自的端口号也互相独立。端口号的分配规则如下:

(1) 端口 0: 不使用,或者作为特殊的使用。

(2) 端口 1~255: 保留给特定的服务。

(3) 端口 256~1023: 保留给其他服务。

(4) 端口 1024~49999: 可以用作任意客户的端口。

(5) 端口 5000~65535: 可以用作用户的服务器端口。

一个完整的网间通信需要两个进程组成,并且只能使用同一种高层协议,因此可以用一个 5 元组来标识: 协议、本地地址、本地端口号、远地地址、远地端口号。

2. 套接字

套接字是支持 TCP/IP 网络通信的基本操作单元,是不同主机间的进程进行双向通信的端点,使用套接字便于区分不同应用程序进程间的网络通信和连接。如图 1-4 所示,有三台建立了通信连接的主机。对通信的一对主机来说,套接字包括发送方 IP、发送方端口号、接收方 IP、接收方端口号、协议五部分。

1.3.2 基于套接字的网络进程通信机制

网络进程与单机进程之间的不同是前者可以在网络上和其他主机中的进程互通信息。在同一台计算机中,两个进程之间通信,只需要两者知道系统为他们分配的进程号(Process ID)就可以实现通信。但是网络情况下,进程通信变得复杂得多。首先,要解决如何识别网络中的不同主机;其次,不同的主机上的系统独立运行,进程号的分配策略也不同。套接字屏蔽了 TCP/IP 协议栈的复杂性,使得在网络编程者看来,两个网络进程间的通信实质上就

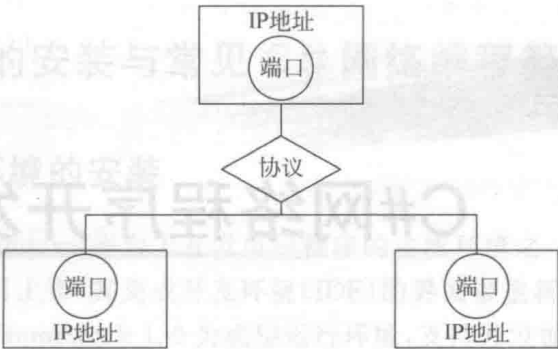


图 1-4 套接字概况图

是它们各自所绑定的套接字之间的通信。这时，通信的网络进程间至少需要一对套接字，分别运行于服务端和客户端，根据连接启动方式及本地套接字连接目标，套接字之间的连接可分为服务监听，客户端请求，连接确认 3 个步骤。图 1-5 给出了 TCP 协议下的网络进程通信的步骤。

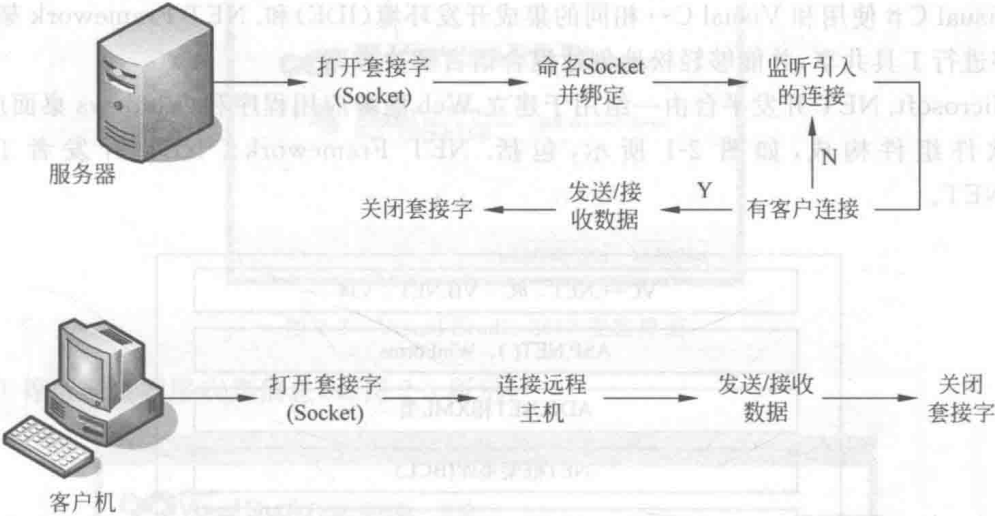


图 1-5 使用套接字传输数据