

- ▶ 学习Windows环境下开发网络应用程序
- ▶ 提供大量的实例源代码及详细解析



网络程序设计 应用教程

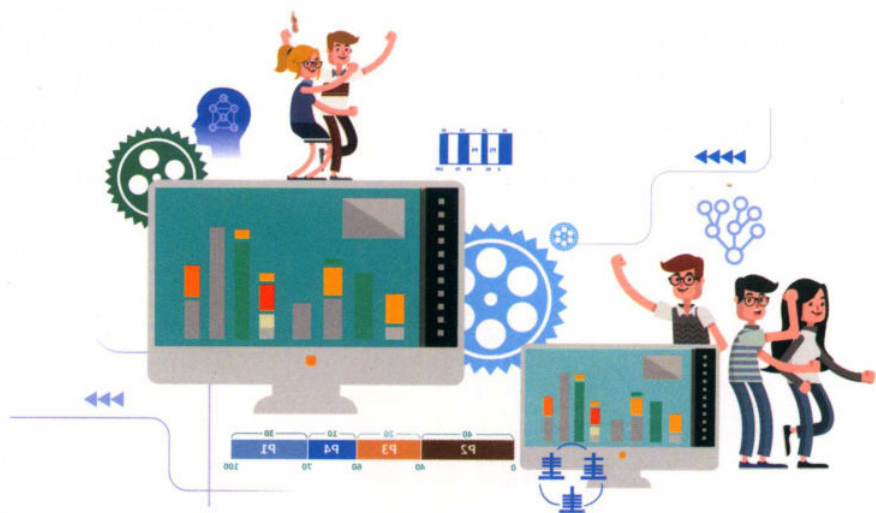
Application Course of Network Programming

曾凡锋 肖珂 著



清华大学出版社





本书遵循理论与实践相结合的教学目的，以实例讲解在Windows环境下如何开发网络应用程序的方法。

全书内容包括：

- 网络编程的基本概念；
- 套接字的基本概念以及Windows Socket的内容；
- 流式套接字、数据报套接字和原始套接字的编程原理，并用实例详细介绍如何开发相应的网络应用程序；
- 常见的应用层协议（如FTP、SMTP、HTTP、MQTT）和WinInet的内容，并用实例描述如何编写基于应用层协议的网络应用程序；
- 使用广泛的WinPcap的概念，以及如何用WinPcap开发相关网络应用程序。

适合计算机、信息安全、通信工程等相关专业的师生和网络工程师阅读参考。

清华社官方微信号



扫 我 有 惊 喜

ISBN 978-7-302-53990-2



9 787302 539902 >

定价：49.00元

内容简介

本书全面系统地介绍了网络编程的理论和实践。全书共分10章。第1章介绍网络编程的基本概念；第2章介绍网络编程的基本原理；第3章介绍网络编程的基本协议；第4章介绍网络编程的基本应用；第5章介绍网络编程的基本技术；第6章介绍网络编程的基本开发；第7章介绍网络编程的基本测试；第8章介绍网络编程的基本维护；第9章介绍网络编程的基本安全；第10章介绍网络编程的基本优化。本书可作为高等院校计算机专业及相关专业的教材，也可供从事网络编程工作的工程技术人员参考。

网络程序设计 应用教程

曾凡锋 肖珂 著



清华大学出版社
北京

此为试读, 需要完整PDF请访问: www.ertongbook.com

内 容 简 介

本书遵循理论与实践相结合的教学目的,以实例讲解在 Windows 环境下如何开发网络应用程序的方法。全书共分 7 章:第 1 章介绍网络编程的一些基本概念;第 2 章介绍套接字的基本概念以及 Windows Socket 的内容;第 3~5 章分别介绍流式套接字、数据报套接字和原始套接字的编程原理,并用实例详细介绍如何开发相应的网络应用程序;第 6 章介绍常见的应用层协议(如 FTP、SMTP、HTTP、MQTT)和 WinInet 的内容,并用实例描述如何编写基于应用层协议的网络应用程序;第 7 章介绍使用广泛的 WinPcap 的概念,以及如何用 WinPcap 开发相关网络应用程序。本书的特点是有大量的实例源代码及详细解析。

本书主要供普通高校计算机、信息安全、通信工程等专业的师生使用,也可供对网络编程感兴趣的各类人员自学使用。

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

网络程序设计应用教程 / 曾凡锋, 肖珂著. —北京: 清华大学出版社, 2019.11

ISBN 978-7-302-53990-2

I. ①网… II. ①曾… ②肖… III. ①JAVA 语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 230710 号

责任编辑:夏毓彦

封面设计:王翔

责任校对:闫秀华

责任印制:李红英

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

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:190mm×260mm

印 张:15

字 数:384千字

版 次:2019年12月第1版

印 次:2019年12月第1次印刷

定 价:49.00元

产品编号:084730-01

前 言

互联网的普及和发展给人们日常生活带来了极大的便利,它呈现了一个高度便捷的信息网络环境,能联网的设备也不局限于计算机,而是扩展到了手机、家用电器、可穿戴设备等多种产品,这是一个一切皆 TCP/IP 的网络时代!这就对网络编程开发人员提出了新的要求,通过本书内容的学习,读者可以在了解 TCP/IP 的相关协议基础上掌握 Socket 编程。

本书主要介绍在 Windows 环境下如何开发网络应用程序,并用实例来介绍开发方法。全书共分 7 章:第 1 章介绍网络编程的一些基本概念,并用实例介绍如何获取计算机网络的基本信息。第 2 章阐述套接字的基本概念,重点介绍 Windows 环境下的套接字实现 Windows Sockets。第 3 章介绍了流式套接字的编程原理,用实例介绍如何实现阻塞流式套接字和非阻塞流式套接字的网络应用程序,并分析两者之间的差别;还介绍了 Windows 下多线程编程的方法。第 4 章介绍了数据报套接字的编程原理,用实例介绍如何实现连接模式和非连接模式的数据报套接字网络应用程序。第 5 章介绍了原始套接字的基本概念和编程原理,分析了原始套接字和标准套接字(流式套接字、数据报套接字)的区别,用实例介绍如何用原始套接字开发有些特殊功能的网络应用程序。第 6 章介绍常见的应用层协议(如 FTP、SMTP、HTTP、MQTT)和 WinInet 的内容,并用实例描述如何编写基于应用层协议的网络应用程序。第 7 章介绍了使用广泛的 WinPcap 的概念,以及如何用 WinPcap 开发相关网络应用程序。

本书提供了 PPT 课件、使用 Visual Studio 开发的 Visual C++ 应用程序源代码等辅助教学资源。本书的特点是有大量实例的源代码以及详细解析。配套示例源码下载请扫描右边二维码获得,如果下载有问题,请联系 booksaga@126.com, 邮件主题为“网络程序设计应用教程”。



本书是编者根据多年网络应用程序开发和相关课程教学的经验逐步形成的,但由于网络技术的快速发展,加之作者水平有限,疏漏和错误之处在所难免,恳请读者和有关专家不吝赐教。

编 者

2019 年 7 月

目 录

第 1 章 网络编程基础	1
1.1 计算机网络基础	1
1.1.1 计算机协议及协议分层	1
1.1.2 网络寻址方式	4
1.2 网络应用程序特点及类型	6
1.3 网络通信模型	7
1.4 客户端/服务器模型	8
1.4.1 客户端和服务器的关系	9
1.4.2 服务器的关键点	10
1.5 其他常见通信模型	11
1.6 网络数据	13
1.6.1 字节顺序	14
1.6.2 结构的对齐	14
1.7 NetBIOS	16
1.8 编程实现计算机网络基本信息查询	17
1.8.1 问题	17
1.8.2 实现方法	17
1.8.3 具体实现	17
1.9 习题	22
第 2 章 套接字基础	23
2.1 TCP/IP 协议软件接口	23
2.2 套接字	24
2.2.1 套接字编程接口的起源与发展	24
2.2.2 套接字的抽象概念	24
2.2.3 套接字接口的特点和内容	25
2.3 Windows 套接字	26

2.3.1	Windows Sockets 规范	26
2.3.2	WinSock 头文件和库文件	27
2.3.3	WinSock API	27
2.3.4	WinSock 初始化	29
2.3.5	错误检查和处理	30
2.3.6	WinSock 的地址描述	30
2.3.7	创建套接字	33
2.4	套接字选项和 I/O 控制命令	33
2.4.1	套接字选项	33
2.4.2	I/O 控制命令	38
2.5	套接字的 I/O 模式	39
2.5.1	网络通信中的 I/O 等待	40
2.5.2	套接字的 I/O 模式	40
2.6	习题	43
第 3 章	流式套接字编程	44
3.1	TCP 传输协议	44
3.1.1	TCP 报文格式	44
3.1.2	TCP 协议的传输特点	45
3.1.3	TCP 连接的建立与关闭	46
3.2	流式套接字的编程模型	48
3.2.1	流式套接字的通信流程	48
3.2.2	流式套接字的编程模式	49
3.3	相关套接字函数	52
3.3.1	创建和关闭套接字	52
3.3.2	绑定地址	54
3.3.3	监听套接字	55
3.3.4	连接套接字	56
3.3.5	数据通信	58
3.4	阻塞流式套接字编程实例	59
3.4.1	问题	59
3.4.2	实现方法	60
3.4.3	基于阻塞流式套接字的服务器实现	60
3.4.4	基于阻塞流式套接字的客户端实现	62
3.5	非阻塞流式套接字编程实例	64
3.5.1	问题	64

3.5.2	实现方法	65
3.5.3	基于非阻塞流式套接字的服务器实现	65
3.5.4	基于非阻塞流式套接字的客户端实现	68
3.6	Windows 下的多线程编程	71
3.6.1	线程的创建和终止	71
3.6.2	线程的同步	72
3.6.3	同步对象	73
3.7	基于流式套接字支持多客户端的编程实例	76
3.7.1	问题	76
3.7.2	实现方法	76
3.7.3	基于阻塞流式套接字的多线程服务器的实现	76
3.7.4	基于阻塞流式套接字的多线程客户端的实现	80
3.8	习题	83
第 4 章	数据报套接字编程	84
4.1	UDP 用户数据报协议	84
4.1.1	UDP 报文格式	84
4.1.2	UDP 协议的传输特点	85
4.2	数据报套接字的编程模型	85
4.2.1	数据报套接字的通信流程	86
4.2.2	数据报套接字的编程模式	87
4.3	相关套接字函数	89
4.3.1	创建和关闭套接字	89
4.3.2	绑定地址	90
4.3.3	数据通信	90
4.4	数据报套接字编程实例	92
4.4.1	问题	92
4.4.2	实现方法	92
4.4.3	数据报套接字的服务器应用程序的实现	92
4.4.4	数据报套接字的客户端应用程序的实现	94
4.4.5	连接模式下数据报套接字的客户端应用程序的实现	97
4.5	数据报套接字广播通信编程实例	99
4.5.1	问题	99
4.5.2	实现方法	99
4.5.3	数据报套接字广播通信的服务器应用程序的实现	99
4.5.4	数据报套接字广播通信的客户端应用程序的实现	101

4.6	数据报套接字通信可靠性的提高	103
4.6.1	数据报套接字的不可靠问题	103
4.6.2	判断服务器是否启动	104
4.6.3	处理传输中数据报丢失的问题	105
4.6.4	处理接收到的干扰数据	106
4.6.5	流量溢出问题	106
4.7	习题	106
第5章	原始套接字编程	107
5.1	原始套接字简介	107
5.1.1	原始套接字和标准套接字的差别	107
5.1.2	原始套接字的功能	108
5.1.3	原始套接字的应用场合	109
5.2	原始套接字的编程模型	109
5.2.1	原始套接字的通信流程	109
5.2.2	原始套接字的编程模式	110
5.3	相关套接字函数	112
5.3.1	创建和关闭套接字	112
5.3.2	绑定地址	113
5.3.3	数据通信	114
5.4	通过原始套接字实现网络数据包捕获实例	115
5.4.1	问题	115
5.4.2	实现方法	115
5.4.3	IP 协议介绍	115
5.4.4	实现网络数据包捕获应用程序	117
5.5	通过原始套接字实现主机连通性检测实例	121
5.5.1	问题	121
5.5.2	实现方法	121
5.5.3	ICMP 协议介绍	122
5.5.4	实现主机连通性检测应用程序	123
5.6	习题	130
第6章	应用层网络编程	131
6.1	应用层协议	131
6.1.1	FTP 协议	132
6.1.2	SMTP 协议	135

005	6.1.3	POP3 协议	138
005	6.1.4	HTTP 协议	140
505	6.1.5	MQTT 协议	144
505	6.2	WinInet	146
505	6.2.1	HINTERNET 句柄	147
505	6.2.2	WinInet 函数	148
405	6.3	Internet 应用相关类	156
405	6.3.1	CInternetSession 类	156
505	6.3.2	CFtpConnection 类	159
505	6.3.3	CHttpConnection 类	162
505	6.3.4	CFtpFileFind 类	163
805	6.4	下载 HTTP 资源编程实例	164
805	6.4.1	问题	164
905	6.4.2	实现方法	164
515	6.4.3	下载 HTTP 资源应用程序的实现	164
515	6.5	发送电子邮件编程实例	168
515	6.5.1	问题	168
515	6.5.2	实现方法	168
515	6.5.3	发送电子邮件应用程序的实现	168
515	6.6	FTP 应用程序编程实例	179
515	6.6.1	问题	179
555	6.6.2	实现方法	180
555	6.6.3	FTP 应用程序的实现	180
555	6.7	主机屏幕监控应用程序编程实例	185
555	6.7.1	问题	185
455	6.7.2	位图	185
455	6.7.3	实现方法	187
455	6.7.4	主机屏幕监控应用程序服务器的实现	187
455	6.7.5	主机屏幕监控应用程序客户端的实现	193
555	6.8	MQTT 通信编程实例	197
555	6.8.1	问题	197
555	6.8.2	MQTT 服务器搭建	197
555	6.8.3	编程实现 MQTT 客户端	198
555	6.9	习题	199

第 7 章	网络底层编程	200
7.1	WinPcap 简介	200
7.2	WinPcap 组成	202
7.2.1	WinPcap 体系架构	202
7.2.2	网络接口驱动	203
7.2.3	NPF 设备驱动	203
7.2.4	Packet.dll	204
7.2.5	Wpcap.dll	204
7.3	WinPcap 安装及编程环境配置	205
7.3.1	下载安装 WinPcap	205
7.3.2	在 Visual Studio 下配置 WinPcap 编程环境	207
7.4	Wpcap.dll 中定义的主要数据结构和函数	208
7.4.1	主要数据结构	208
7.4.2	主要函数	209
7.5	Packet.dll 的主要数据结构和函数	212
7.5.1	主要数据结构	212
7.5.2	主要函数	214
7.6	使用 WinPcap 实现网络数据包捕获实例	217
7.6.1	问题	217
7.6.2	实现方法	217
7.6.3	实现网络数据包捕获应用程序	217
7.7	使用 WinPcap 发送 ARP 伪装数据包实例	222
7.7.1	问题	222
7.7.2	实现方法	223
7.7.3	ARP 协议介绍	223
7.7.4	实现网络发送 ARP 伪装数据包捕获应用程序	224
7.8	习题	229

第1章

网络编程基础

网络编程的基础是计算机网络。本章主要介绍网络程序设计中所涉及的计算机网络方面的基础知识，包括一些网络术语、网络协议、NetBIOS 等。

1.1 计算机网络基础

1.1.1 计算机协议及协议分层

计算机网络是指将位于不同地理位置且具有独立功能的多台计算机，通过通信线路以及网络设备连接起来，在网络操作系统、网络管理软件及网络通信协议的管理和协调下，实现资源共享和信息传递的计算机系统。总的来说，计算机网络的组成包括计算机、网络操作系统、传输媒体、网络设备以及网络应用软件几部分。

计算机网络是一个极为复杂的系统，包括大量的应用程序和协议、各种类型的网络设备、多种网络操作系统，以及各种类型的链路级媒体。面对这种复杂的系统，为了简化管理、降低设计难度，计算机网络设计者以分层的方式组织协议以及实现这些协议的网络硬件和软件。协议分层具有概念化和结构化的优点，每一层都建立在下层之上，使用它的下层提供的服务，但不用关心下层服务细节的实现。

计算机协议是指计算机通信双方关于如何进行通信的一种约定。现有的计算机网络采用 OSI 参考模型，在此模型中每个协议都属于某个层次。

1. OSI 参考模型

在 OSI 参考模型出现之前，计算机网络中存在多种体系结构，其中以 IBM 公司的系统网络体系结构（System Network Architecture，简称 SNA）和 DEC 公司的数字网络体系结构（Digital Network

Architecture, 简称 DNA) 比较著名。为了解决不同体系结构的计算机网络互连问题, ISO (国际标准化组织) 在 1981 年制定了开放系统互连参考模型 (Open System Interconnection, 简称 OSI)。该体系结构标准定义了网络互连的七层框架, 这七层框架由低到高分别是物理层 (Physical Layer)、数据链路层 (Data Link Layer)、网络层 (Network Layer)、传输层 (Transport Layer)、会话层 (Session Layer)、表示层 (Presentation Layer) 和应用层 (Application Layer), 如图 1-1 (a) 所示。

第 7 层应用层: 为特定类型的网络应用提供访问 OSI 环境的手段, 为用户的应用程序和网络之间提供接口。应用层确定进程之间通信的性质来满足用户的需要。

第 6 层表示层: 主要用于处理两个通信系统中交换信息的表示方式, 协商数据交换格式。为上层用户解决用户信息的语法问题。

第 5 层会话层: 在两个节点之间建立端到端连接, 为端系统的应用程序之间提供对话控制机制。

第 4 层传输层: 面向连接或无连接的信息传输, 为会话层用户提供一个端到端的可靠、透明和优化的数据传输服务机制。

第 3 层网络层: 通过寻址来建立两个节点之间的连接, 为源端的传输层送来的数据包选择合适的路由和交换节点, 正确无误地按照地址传送给目的端的传输层。

第 2 层数据链路层: 将数据分帧, 为网络层提供一个数据链路的连接, 在一条有可能出差错的物理连接上进行几乎无差错的数据传输 (差错控制)。

第 1 层物理层: 主要功能是利用物理传输介质为数据链路层提供物理连接, 以便透明地传送比特流。常用设备有网卡、集线器、路由器、网线、双绞线、光纤光缆等。

第 5 层到第 7 层统称为上层协议, 负责端到端的数据通信。第 1 层到第 4 层统称为下层协议, 负责建立网络通信连接的链路。每层除了完成上述的功能, 也直接为其上层提供服务, 而网络通信则以自上而下 (在发送端) 或者自下而上 (在接收端) 双向进行。当然, 并不是每一通信都需要经过 OSI 的全部七层, 有的甚至只需要双方对应的某一层即可, 如中继器之间的连接就只需在物理层中进行, 而路由器之间的连接则只需经过网络层以下的三层。双方的通信是在对等层次上进行的, 不能在不同层次上进行通信。

2. TCP/IP 参考模型

ISO 制定的 OSI 参考模型过于庞大、复杂, 使用时有许多不方便。相比之下, 在此之前使用多年的 TCP/IP 协议簇获得了更为广泛的应用。TCP/IP 协议簇是美国国防部高级研究规划局在 20 世纪 60 年研究的计算机网络 (Advanced Research Projects Agency Network, 简称 ARPANET) 使用的参考模型, 后来的因特网 (Internet) 也采用了该参考模型。

TCP/IP 参考模型分为四个层次: 应用层、传输层、网络互联层和网络接口层, 如图 1-1 (b) 所示。在 TCP/IP 参考模型中去掉了 OSI 参考模型中的会话层、表示层 (这两层的功能被合并到应用层) 和物理层 (这一层的功能被合并到网络接口层)。以下分别介绍各层的主要功能:

(1) 应用层。应用层面向不同的网络应用引入了不同的协议。TCP/IP 协议簇的应用层协议包括许多, 其中有基于 TCP 协议的, 如文件传输协议 (File Transfer Protocol, 简称 FTP)、超文本传输协议 (Hypertext Transfer Protocol, 简称 HTTP)、远程终端协议 (Telecommunication Network,

简称 Telnet)、简单邮件传输协议 (Simple Mail Transfer Protocol, 简称 SMTP)、邮局协议版本 3 (Post Office Protocol-Version 3, 简称 POP3) 等; 也有基于 UDP 协议的, 如网络时间协议 (Network Time Protocol, 简称 NTP)、简单文件传输协议 (Trivial File Transfer Protocol, 简称 TFTP)、简单网络管理协议 (Simple Network Management Protocol, 简称 SNMP) 等; 也有可基于 TCP 和 IP 两种协议的, 如域名解析服务 (Domain Name Service, 简称 DNS)。



图 1-1 OSI 参考模型与 TCP/IP 参考模型

(2) 传输层。在 TCP/IP 模型中, 传输层的功能是使得源端主机和目标端主机上的应用实体可以进行会话。在传输层定义了两种服务质量不同的协议, 即传输控制协议 (Transmission Control Protocol, 简称 TCP) 和用户数据协议 (User Datagram Protocol, 简称 UDP)。

TCP 协议是一个面向连接 (双向)、安全可靠的协议, 为应用程序提供面向连接的服务, 连接时通过三次握手协议实现。TCP 将源主机发送的数据有序无差错地发往目标主机。在发送端, 它负责把上层传送下来的消息流分成数据段并传递给下层; 在接收端, 它负责把收到的数据包进行重组后递交给上层。另外, TCP 协议还要处理网络拥塞控制, 在网络拥塞时帮助发送端抑制其传输速度; 提供端到端的流量控制, 避免缓慢接收的接收方没有足够的缓冲区接收发送方发送的大量数据。TCP 的协议数据传输单元为 TCP 数据段 (TCP Segment)。

UDP 协议是一个不可靠、无连接的协议, 为应用程序提供无连接的服务, 穿透性比 TCP 协议强。这种服务主要适用于广播数据发送和不需要对报文进行排序和流量控制的场合。UDP 的协议数据传输单元为 UDP 数据报 (UDP Datagram)。

(3) 网络互联层。网络互联层是整个 TCP/IP 协议簇的核心, 主要解决主机到主机的通信问题。网络互联层的功能是通过路径选择把数据包发往目标网络或主机, 进行网络拥塞控制以及差错控制。

网际协议 (Internet Protocol, 简称 IP) 是网络互联层的重要协议, 该协议定义了数据包中的各个字段以及端系统和路由器如何作用于这些字段。

网络互联层中的另一个协议 Internet 控制报文协议 (Internet Control Message Protocol, 简称 ICMP) 用于在 IP 主机、路由器之间传递控制消息。控制消息包括网络是否畅通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据, 但是对于用户数据的传递起着重要的作用。

另外，网络互联层也包括决定路由的选路协议（如 RIP、OSPF 等），数据包根据选定的路由从源传输到目的地。

网络互联层的协议数据传输单元为数据包（Packet），或称为分组。

（4）网络接口层。网络接口层负责监视数据在主机和网络层之间的通信，将网络层接收到的数据分割成特定的、可被主机传输的帧，并交付主机物理层进行实际的数据传送。

网络接口层提供的服务取决于应用于该层的协议，常用的协议包括以太网的 802.3 协议、Wi-Fi 的 802.11 协议和地址解析协议（Address Resolution Protocol，简称 ARP）等。

网络接口层的协议数据传输单元为帧（Frame）。

1.1.2 网络寻址方式

网络应用程序进行网络通信时，必须告诉网络某些信息以标识该程序。在 TCP/IP 中，网络应用程序使用两个信息来唯一标识特定应用程序：IP 地址和端口号。在具体的网络应用中，IP 地址和端口号的使用还会遇到一些更复杂的变化，如名称解析、网络地址转换等。

在 TCP/IP 协议簇中，寻址是在两个级别上发生的：网络接口层（用来处理物理地址，即 MAC 地址）和网络互联层（处理逻辑的分层寻址，即 IP 寻址）。

1. IP 地址

互联网上的每个主机都有 IP 地址，它将网络号和主机号编码在一起。这个组合在全网范围内是唯一的（原则上，互联网上不能有 IP 地址相同的主机）。

IP 地址是二进制数字，目前有 IPv4 和 IPv6 两种版本，分别对应于已经标准化的网际协议的两个版本。IPv4 地址的长度为 32 位，用于标识 40 亿个不同的地址。随着 Internet 的快速发展，最初 Internet 使用的 IPv4 地址范围已经不能满足使用的需要了。而 IPv6 地址的长度为 128 位，完全能够满足 Internet 发展的需要。现在大部分主机还是使用 IPv4 地址标识，因此本书重点说明 IPv4 寻址技术。

为了便于使用，标识 IP 地址时，按照惯例将 IPv4 地址写为一组 4 个用句点隔开的十进制数字（例如 10.0.0.3），称为“点分十进制”表示法。点分十进制字符串中的 4 个数字表示 IP 地址的 4 个字节的内容，每个部分都是 0~255 之间的数字。

对 IP 地址编址的方法是分类编址方法。最初设计互联网络时，为了便于寻址以及层次化构造网络，每个 IP 地址包括两个标识码（ID），即网络标识和主机标识。同一个物理网络上的所有主机都使用同一个网络标识，网络上的每一个主机（包括工作站、服务器和路由器等）有一个主机标识与其对应。IP 地址根据网络标识的不同分为 5 种类型：A 类地址、B 类地址、C 类地址、D 类地址和 E 类地址，如表 1-1 所示。

表 1-1 IP 地址分类编址方法

类别	网络标识	主机标识	地址范围	网络数	每个网络主机数
A 类	1 字节, 最高位为 0	3 字节	1.0.0.0~ 126.0.0.0	126	1 亿多
B 类	2 字节, 最高位为 10	2 字节	128.0.0.0~ 191.255.255.255	16382	6 万多
C 类	3 字节, 最高位为 110	1 字节	192.0.0.0~ 223.255.255.255	209 万多	254
D 类	最高位为 1110		224.0.0.0~ 239.255.255.255		用于多播, 一次寻址一组计算机
E 类	最高位为 11110		204.0.0.0~ 255.255.255.254		保留将来使用

另外还有一些特殊用法的地址段, 如:

(1) 10.0.0.0~10.255.255.255、172.16.0.0~172.31.255.255、192.168.0.0~192.168.255.255 对应的是 A 类、B 类、C 类的私有地址, 这些地址被大量用于内部的局域网中, 以免局域网接入公网时引起地址混乱。

(2) 127.0.0.0~127.255.255.255 是保留地址, 用于本地回环测试, 发送到这个地址的数据包不会被传输到网络线路上, 而是被当作到来的数据包直接在本地处理。这可用于本机两个应用程序的通信。

(3) 169.254.0.0~169.254.255.255 是保留地址。

(4) 0.0.0.0 表示所有不清楚的主机和目的网络。一般在网络配置中设置了默认网关, Windows 系统会自动产生一个目的地址为 0.0.0.0 的默认路由。

(5) 255.255.255.255 表示限制广播地址, 对本机来说, 它指本网段内 (同一广播域) 的所有主机。

2. 端口号

网络层 IP 地址用来寻址指定的计算机或者网络设备, 而一台计算机可以运行多个网络应用程序, 传输层的端口号就是用来区分这些应用程序的。端口号是 16 位的, 范围在 0~65535 之间。在设备上寻址端口号时经常使用的形式是“IP 地址:端口号”, 通信的两端都要使用端口号来唯一标识其主机内运行的特定应用程序。

许多公共服务都使用固定的端口号 (1~255 之间), 如 HTTP 服务器默认使用的端口号是 80, FTP 服务器使用的端口号是 21, SMTP 服务器使用的端口号是 25 等。端口号 256~1023 通常由 Unix 系统占用。自定义的服务一般使用高于 1024 的端口号, 通常分配 1024~5000, 而 5000 端口号大多为其他服务器预留 (Internet 上不常用的服务)。

3. 名称解析

在基于 TCP/IP 的网络中, IP 地址被用来唯一标识网络上的某台计算机。如果某台计算机想访

问网络中的其他计算机,首先要知道被访问的目标计算机的 IP 地址,然后通过该 IP 地址与其通信。

但在有些实际应用中,用户很少直接使用 IP 地址来访问网络中的资源,而是习惯使用便于记忆的计算机名或域名,比如 Web 应用。当用户在浏览器地址栏中输入“http://www.ncut.edu.cn”想要访问网络中的某台服务器时,用户的计算机必须通过一个地址转换过程,将该域名转换成该服务器的 IP 地址然后去访问该服务器,这个名称转换过程被称为名称解析服务。

名称解析服务的主要来源有两种,分别是域名系统(Domain Name System,简称 DNS)和本地配置数据库(一般是操作系统中用于本地名称与 IP 地址映射的特殊机制,如 Windows 系统中的 NetBIOS 解析)。

4. 网络地址转换

现有的 IPv4 的网络地址方案中 IP 地址是短缺的资源,不能满足整个 Internet 的需求,最终的解决方案是迁移到 IPv6。对于现有的 IPv4 网络,能够马上投入使用的解决方法是 NAT 技术。

网络地址转换(Network Address Translation,简称 NAT)是接入广域网(WAN)的一种技术,能够将保留私有地址(A类、B类、C类各有一段)转化为合法的公有 IP 地址,它被广泛应用于各种 Internet 接入方式和各种类型的网络中。NAT 解决了 IP 地址不足的问题,而且还能够有效地避免来自网络外部的攻击,隐藏并保护网络内部的计算机。NAT 的实现方式有三种:静态转换、动态转换和端口多路复用。

(1) 静态转换是指将内部网络的私有 IP 地址转换为公有 IP 地址时,静态转换中的私有和公有 IP 地址是一对一和一成不变的,某个私有 IP 地址只转换为某个公有 IP 地址。

(2) 动态转换是指将内部网络的私有 IP 地址转换为公用 IP 地址时,IP 地址是不确定和随机的,所有被授权访问 Internet 的私有 IP 地址都可随机转换为任何指定的合法 IP 地址。

(3) 端口多路复用是指改变外出数据包的源端口并进行端口地址转换(Port Address Translation,简称 PAT)。内部网络的所有主机均可共享一个合法外部 IP 地址实现对 Internet 的访问,从而可以最大限度地节约 IP 地址资源。目前网络中应用最多的就是端口多路复用方式。

NAT 有效解决了 IP 地址短缺的问题,但是它也带来了一些新的问题,使得开发点对点通信应用程序时需要有很多附加的考虑,主要体现在:

(1) 处于 NAT 后面的主机不能充当服务器直接接收外部主机的连接请求,必须对 NAT 设备进行相应的配置才能完成外部地址与内部服务器地址的映射。

(2) 处于不同 NAT 之后的两台主机无法建立直接的 UDP 或 TCP 连接,必须使用中介服务器来帮助它们完成初始化的工作。

1.2 网络应用程序特点及类型

随着计算机技术的发展和应用的深入,网络应用程序在构建企业级的应用系统时得到广泛使用。这类网络应用程序的主要特点是: