

Visual C++

网络通信协议分析 与应用实现

汪晓平 钟军 等 编著



附光盘
CD-ROM

人民邮电出版社

POSTS & TELECOMMUNICATIONS PRESS

TN915.04
28D

Visual C++

网络通信协议分析 与应用实现

■ 汪晓平 钟军 等 编著

北方工业大学图书馆



00525764

人民邮电出版社

图书在版编目(CIP)数据

Visual C++网络通信协议分析与应用实现/汪晓平, 钟军编著.—北京: 人民邮电出版社, 2003.2
ISBN 7-115-11005-0

I. V... II. ①汪... ②钟... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 000610 号

内容简介

本书介绍了如何利用 Visual C++ 6.0 开发网络通信应用程序的方法, 同时主要对目前流行的 FTP、HTTP、E-mail、Telnet、ICMP、网络监控、Modem 串口通信编程、拨号网络编程、语音电话等 Internet 上使用的协议与通信协议高级编程开发进行了详细的讲解, 并结合大量的实例使读者能够深入的了解各种网络应用程序的开发技巧。另外还介绍了在 VC 中进行网络通信开发的基本方法和技术以及各种网络的基础应用。

本书主要面向具有一定 VC 网络通信编程基础并希望学习较高层次编程技术的读者, 主要涉及到网络开发与通信两方面的内容, 如常用的 Internet 协议开发理论与实例, Modem 串口通信理论与实例, 流量监控等。本书适合中、高级 Visual C++程序员进行网络与通信开发时阅读和参考。

Visual C++网络通信协议分析与应用实现

- ◆ 编 著 汪晓平 钟 军 等
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 38.5
字数: 944 千字 2003 年 2 月第 1 版
印数: 1-5 000 册 2003 年 2 月北京第 1 次印刷

ISBN7-115-11005-0/TP·3305

定价: 60.00 元(附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前言

目前市场上关于 VC 网络通信编程的书虽然特别的多,但笔者收到许多读者朋友的咨询,并反映市场上虽然书多,但是真正对网络编程进行深入解析的书很少,通过对市场上大多数的 VC 网络编程的书进行了调查,发现存在以下几方面的不足。

- 基础东西太多,关于网络实际开发内容的内容少。
- 实例多,但是解析少,往往只是针对例题进行说明,读者无法进行扩展。
- 实例功能不够全面,往往只是一个简单的应用而已,读者要想深入了解太难。
- 虽然是网络通信的书,但是关于网络协议的介绍很少,而协议是网络之根本。

针对目前市场上此类图书存在的不足,我们组织策划编写了本书。首先,本书吸取了目前市场上其他许多网络通信编程书籍的精华,结合笔者实际编程经验并加以总结。其次,本书采用“协议分析和应用实例”的结构安排各章内容,书中加强了关于网络协议的阐述,使得读者能够了解每一种网络通信协议的来龙去脉,这样,读者才能够看懂书中介绍的每一个例程,而且更为重要的就是能够根据自己的需求以及书中介绍的协议规则对程序进行扩充。这些都是以往图书所做不到的。

本书涉及的理论内容及其实例有:

■ 第 1 章: TCP/IP

主要介绍 TCP/IP 的一些基础知识,因为目前大部分的网络通信协议都是建立在 TCP/IP 基础上的,因此这一部分内容,读者应该有一定的了解。

■ 第 2 章: Windows 网络编程基础

主要介绍 Windows 环境中各种网络编程的方法,如直接采用 Winsock API 进行网络开发,或者利用 MFC 的 WinInet 类、CAsyncSocket 类、CSocket 类来进行开发,最后还介绍了多线程的开发技术。

■ 第 3 章: TCP、UDP 编程

本章主要是对第 2 章网络编程技术的总结,通过聊天程序的开发,让读者熟悉网络编程的开发步骤、开发技巧、多线程技术等,而这些都是后面进行网络协议开发的基础。

■ 第 4 章: 基础的网络应用

本章主要是通过介绍一些基本的网络应用,让读者了解这些基本网络知识,并试图能够在自己的网络程序中加入这些功能。本章着重介绍了以下几种基础应用:获取计算机 IP 地址和计算机名、获取(域名/子网掩码/网卡类型)、获取网卡的 MAC 地址、获取系统支持的网络协议信息、端口扫描高级编程、超级链接、TCP/IP 超级终端等。

■ 第 5 章: FTP 高级编程

本章详细介绍了 FTP 的基本原理、工作模型、各种 FTP 命令大全,并给出了一个综合的 FTP 程序,程序中提供了各种接口,使读者能够对本程序进行扩充。

■ 第 6 章: HTTP 协议高级编程

HTTP 是最常用的协议之一。本章首先介绍了 HTTP 的基础知识，再介绍了几个实例，如使用 HTTP 进行网站下载、实现 HTTP Web 服务器、HTTP 代理服务器等。同时介绍了网络浏览器的开发。

■ 第 7 章: Telnet 协议高级编程

本章介绍 Telnet 协议的基本理论，以及流行 BBS 系统的客户机/服务器应答原理，最后提供了一个简单的 BBS 程序，能够实现基本的 BBS 浏览。

■ 第 8 章: E-mail 协议及高级编程

本章从 E-mail 通信的基本原理开始，介绍了用 SMTP 发送 E-mail、POP3 接收 E-mail 的原理与实现。针对发送附件与解析附件所用的编码解码算法，以及乱码的出现进行了详细的介绍，同时还能够实现程序进行 SMTP 的验证以及通信信息的加密传输。

■ 第 9 章: ICMP 协议高级开发

本章首先介绍 ICMP 协议的原理，然后介绍 ICMP 的两个基本的应用，一个是 PING 程序的实现，然后就是 Tracert 路由功能实现。

■ 第 10 章: 远程访问 (RAS) 高级编程

RAS 拨号是 Windows 提供的一个网络接入服务。本章介绍了 RAS 的编程方法，包括对拨号网络的电话簿、拨号连接、挂断以及获取拨号网络的动态 IP 等。最后提供了一个能够进行各种拨号、设置的实用程序，该程序还可以监控各种网络流量，如拨号流量、网卡流量等。

■ 第 11 章: 语音电话高级编程

本章介绍了 Modem 的基本通信原理以及各种 AT 指令。除介绍常用的命令外，还介绍了关于利用 Modem 进行语音传输的 AT 命令，最后给出能实现语音电话功能的实例程序，通过该程序，读者能够实现自动应答、定时呼叫客户等高级功能。

■ 第 12 章: 串口通信高级编程

本章介绍了在 Windows 环境中进行串口开发的方法、开发步骤和错误处理等内容，另外本章还给出了一个简单的串口通信的实例，读者可以根据书中介绍知识来开发符合自己要求的程序。

本书所附的光盘包括书中全部实例的源代码和可执行文件，并且所有实例均在 VC 环境下调试实现。

本书由汪晓平、钟军合作编写而成，此外参与编写工作的还有张宏林、肖洪伟、李廷文、张增强、王洪涛、吴继刚、周学明、李闽溟、黄沙、宣小平、但正刚、张文毅、张小磊、胡昱、范国平、陈晓鹏、王凯封、潘邦传、王锐、闫卫东、赵明华、许福、施新刚、郑刚、李现勇、谭思亮、邹超群、郭瑞军、杨枝灵、彭珂珂、赵苏琦、徐建军、胡伟、刘江、王茹、阎海荣、刘理、谭春华、张益贞、刘韬、杨茂林、董晓宇、王三暖、刘星等，在此一并表示感谢。

限于笔者的能力，错误、浅陋之处在所难免，恳请广大读者批评指正。

编者

2003 年 1 月

目 录

第 1 章 TCP/IP.....	1
1.1 TCP/IP 族简介	1
1.1.1 OSI 模型	1
1.1.2 TCP/IP 结构	2
1.1.3 常用协议	3
1.1.4 进程/应用层协议	5
1.1.5 RFC 和标准简单服务	6
1.2 TCP/IP 基本概念	7
1.2.1 IP 地址与子网掩码	7
1.2.2 地址解析	9
1.2.3 域名系统	10
1.2.4 数据包的封装和分用	10
1.2.5 IP 数据报	11
1.2.6 UDP 数据报	14
1.2.7 TCP 数据报	15
1.2.8 端口号	21
第 2 章 Windows 网络编程基础	22
2.1 Winsock 基本概念	22
2.1.1 套接字 (Sockets)	22
2.1.2 基本概念	23
2.1.3 字节顺序	25
2.2 Winsock 编程原理	25
2.2.1 Winsock 的启动和终止	25
2.2.2 错误检查和控制	26
2.2.3 Winsock 编程模型	26
2.3 Winsock I/O 模型	31
2.3.1 Select 模型	32
2.3.2 WSAAsyncSelect 模型	33
2.3.3 WSAEventSelect 模型	34
2.4 Winsock 2 的扩展特性	36
2.4.1 原始套接字	37
2.4.2 重叠 I/O 模型	37
2.4.3 服务的质量 (QOS)	38

2.5 套接字选项和 I/O 控制命令	40
2.5.1 套接字选项	40
2.5.2 I/O 控制命令	41
2.6 WinInet 网络编程基础	42
2.6.1 MFC WinInet 类	43
2.6.2 利用 WinInet API 进行编程	53
2.7 MFC Windows Sockets 网络编程基础	61
2.7.1 CAsyncSocket 类	62
2.7.2 CSocket 类	66
2.8 多线程编程	70
2.8.1 Win32 基础上的多线程编程	71
2.8.2 基于 MFC 的多线程编程	73
2.9 Winsock API 详解	75
第 3 章 TCP、UDP 高级编程	76
3.1 TCP 实现客户机/服务器聊天	76
3.1.1 服务器端程序开发	76
3.1.2 客户端程序开发	82
3.1.3 程序运行演示	85
3.2 UDP 实现点对点聊天程序开发	86
3.2.1 实现原理	87
3.2.2 代码分析	87
3.2.3 程序演示	93
3.3 TCP/UDP 聊天程序高级开发	95
3.3.1 实现原理	95
3.3.2 代码分析	95
3.3.3 程序运行演示	124
3.4 小结	124
第 4 章 基本网络应用	126
4.1 获取计算机 IP 地址和计算机名	126
4.1.1 实现原理	126
4.1.2 程序实现	127
4.2 获取域名、子网掩码、网卡类型	130
4.2.1 实现原理	130
4.2.2 程序实现	133
4.3 获取网卡的 MAC 地址	136
4.3.1 实现原理	136
4.3.2 程序实现	139
4.4 获取系统支持的网络协议信息	140

4.4.1 实现原理	140
4.4.2 程序实现	143
4.5 端口扫描高级编程	152
4.5.1 实现原理	153
4.5.2 程序实现	153
4.6 超级链接高级编程	158
4.6.1 实现原理	159
4.6.2 程序实现	159
4.7 TCP/IP 超级终端	164
4.7.1 实现原理	165
4.7.2 程序实现	165
第 5 章 FTP 高级编程	170
5.1 FTP 简介	170
5.2 安装设置 FTP 服务器	171
5.3 使用 Windows 内置 FTP 客户端程序	175
5.4 深入 FTP	178
5.4.1 FTP 工作原理	178
5.4.2 FTP 命令	184
5.5 FTP 服务器高级编程	203
5.5.1 建立工程项目	203
5.5.2 实例分析	204
5.5.3 实例测试	216
5.6 FTP 客户端高级编程	217
5.6.1 建立工程项目	218
5.6.2 实例分析	219
5.7 小结	245
第 6 章 HTTP 高级编程	246
6.1 HTTP 介绍	246
6.1.1 HTTP 背景	246
6.1.2 HTTP 的内容	249
6.1.3 消息 (Message)	250
6.1.4 请求 (Request)	251
6.1.5 响应 (Response)	255
6.1.6 访问认证	258
6.1.7 URL 编码	261
6.1.8 HTTP 的应用	262
6.2 获取远程服务器文件信息	262
6.3 网站下载高级开发	266

6.4	网络浏览器高级开发	287
6.4.1	实现原理	287
6.4.2	实例实现	288
6.5	Web Server 高级开发.....	296
6.5.1	Web Server 相关理论.....	296
6.5.2	实例实现	297
6.6	小结.....	317
第7章	FTP 高级编程.....	318
7.1	Telnet 简介.....	318
7.2	使用 Windows 的 Telnet 程序登录远程服务器.....	319
7.3	深入 Telnet 协议.....	320
7.3.1	Telnet 概述	320
7.3.2	Telnet 命令	325
7.3.3	NVT ASCII 字符集.....	326
7.3.4	协商选项	326
7.3.5	子协商选项	327
7.3.6	Telnet 操作方式	328
7.4	Telnet 客户端——BBS 高级程序开发.....	329
7.4.1	实例实现	329
7.4.2	实例分析	330
7.5	小结.....	346
第8章	E-mail 协议及高级编程	347
8.1	SMTP、POP3 与 E-mail.....	347
8.2	信件结构详述	347
8.2.1	RFC822 信件的格式和内容.....	347
8.2.2	构造符合 RFC822 的信件.....	355
8.2.3	RFC822 信件的语法分析.....	356
8.3	MIME 编码解码与发送附件	357
8.3.1	RFC822 的局限.....	357
8.3.2	Uuencode 编码与解码	357
8.3.3	MIME 及其编码	360
8.3.4	构造 MIME 信件	377
8.3.5	MIME 信件的语法分析	379
8.4	SMTP 及发送电子邮件.....	380
8.4.1	SMTP 的模型描述.....	380
8.4.2	SMTP 的会话过程.....	380
8.5	发送无附件 E-mail 程序	388
8.5.1	实例实现	388

8.5.2 代码分析	389
8.6 发送有附件的邮件	391
8.6.1 实例实现	391
8.6.2 代码分析	391
8.7 POP3 与接收电子邮件	421
8.7.1 POP3 的模型描述	421
8.7.2 POP3 的会话过程	421
8.8 接收 E-mail 的程序	429
8.8.1 实例实现	429
8.8.2 代码分析	430
8.9 E-mail 乱码	442
8.9.1 乱码的常见形式及形成原因	442
8.9.2 避免乱码的方法	443
8.10 MAPI 概述	444
8.11 小结	444
第 9 章 ICMP 高级开发	445
9.1 ICMP	445
9.1.1 ICMP 报文格式	445
9.1.2 ICMP 时间戳请求与应答	447
9.1.3 回显请求和回显应答报文	447
9.1.4 ICMP 地址掩码请求与应答	448
9.1.5 ICMP 端口不可达差错	448
9.2 Windows 自带程序应用	448
9.2.1 ping 命令测试	449
9.2.2 Traceroute 命令测试	450
9.3 ICMP 高级实例开发	451
9.3.1 实现原理	451
9.3.2 代码分析	452
9.3.3 实例演示	462
9.4 小结	463
第 10 章 远程访问 (RAS) 高级编程	464
10.1 RAS 客户机	464
10.2 建立拨号连接	465
10.3 RAS 重要函数说明	469
10.3.1 连接函数	470
10.3.2 连接管理函数	474
10.3.3 电话簿和用户凭证管理	478
10.3.4 拨号方式	479

10.4	RAS 高级程序开发实例.....	481
10.4.1	实例实现	481
10.4.2	代码分析	483
10.5	RAS 应用实例——远程文件共享	505
10.6	小结.....	509
第 11 章	语音电话通信协议及高级编程	510
11.1	Modem 通信协议	510
11.1.1	Modem 状态	510
11.1.2	AT 命令	511
11.1.3	S 寄存器	526
11.1.4	Modem 返回信息码	529
11.2	语音电话高级编程.....	529
11.3	小结.....	537
第 12 章	串口通信高级编程	538
12.1	串口通信硬件理论	538
12.2	Windows 串口通信编程	542
12.2.1	打开和关闭串口	543
12.2.2	串口配置和串口属性	545
12.2.3	读写串口	556
12.2.4	通信事件	558
12.2.5	设备控制命令	560
12.3	串口数据传输高级编程	561
12.3.1	代码分析	562
12.3.2	CSerialCommHelper 类	562
12.4	小结.....	570
附录一	Winsock 1.1 函数参考	571
附录二	Winsock 2 函数参考	590
附录三	Windows Sockets 错误码.....	600

第 1 章 TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol, 传输控制协议/网际协议) 是发展至今最成功的通信协议。它起源于 20 世纪 60 年代末美国政府资助的一个分组交换网络研究项目, 到 20 世纪 90 年代已发展成为计算机之间最常应用的组网协议。它允许分布在各地的装着完全不同系统的计算机互相通信。它是一个真正的开放系统, 该协议族的定义及其多种实现可以通过多重途径无偿得到。随着 PC 的普及, TCP/IP 以其开放性的特点, 成为了 Internet 的基础, 该广域网 (WAN) 把全世界数以百万的电脑连接在了一起。TCP/IP 族的如此重要的作用是它的初创者完全没有想到的。

1.1 TCP/IP 族简介

1.1.1 OSI 模型

OSI (Open System Interconnection) 将计算机网络通信制定的一个 7 层框架, 这个 7 层协议的框架, 称为“OSI/RM” (Open System Interconnection / Reference Model, 开放系统互联参考模型), 如图 1-1 所示。

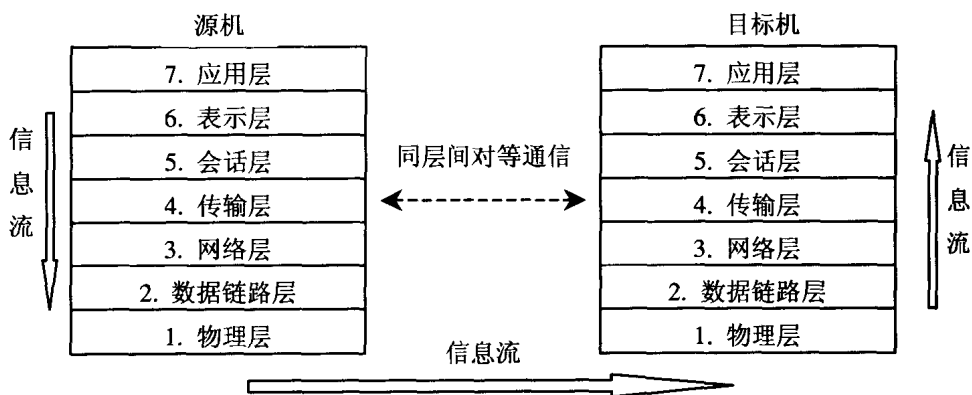


图 1-1 OSI 模型与通信流程

OSI 模型把计算机网络通信的组织与实现按功能划分为 7 个层次, 即从一个计算机系统发出通信请求起, 到信息经过实际物理线路传送到另一个目标计算机系统为止, 把通信功能从高到低划分为应用层、表示层、会话层、传输层、网络层、数据链路层和物理层, 各层的具体功能如表 1-1 所示。网络通信协议按层次组织, 也是为了减少协议的复杂性。每一层协议建立在它的下层协议的基础上, 每一层又为其上层提供服务, 完成上层提交的任务。至于在一层内如何进行服务的细节, 对上层则是隐蔽的。一台计算机的某指定层同另一台计算机的

相应层对话，对话的全部规则和约定就构成该层的协议。当然，信息（数据和控制信息）并不是从某一计算机系统的第 N 层直接传到另一计算机系统的第 N 层，而是从这台计算机的某一层直接传送至下层，最后经过物理介质到达另一台计算机，然后再由底层逐层向上传送，如图 1-1 中的数据流程所示。

当然，OSI 模型只是一个框架，它的每一层并不执行某种功能，功能的具体实现还需通信协议，主要是软件来进行。当数据在层间向下传播时（源机部分），每一个层都会为传输中的数据增加一个包头（header），用于标识包的来源与目的。到了目标机时，每一层都从数据中读取相应包头，执行请求的任务，并负责向上传输数据包。每一种具体的协议一般都定义了 OSI 模型中的各个层次具体实现的技术要求。

表 1-1 OSI 模型中各个层的功能

名称	层次	功能
物理层	1	实现计算机系统与网络间的物理连接
数据链路层	2	进行数据打包与解包，形成信息帧
网络层	3	提供数据通过的路由
传输层	4	提供传输顺序信息与相应
会话层	5	建立和中止连接
表示层	6	数据转换，确认数据格式
应用层	7	提供用户程序接口

1.1.2 TCP/IP 结构

TCP/IP 是一个四层协议，它的结构如图 1-2 所示。

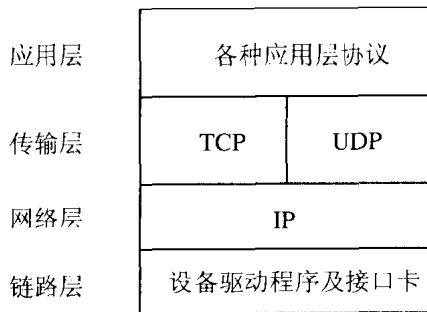


图 1-2 TCP/IP 族的体系结构

每一层负责的功能如下：

链路层：有时被称作数据链路层或网络接口层，通常包括操作系统中的设备驱动程序和计算机中对应的网络接口卡，它们一起处理与电缆（或其他任何传输媒介）的物理接口细节。该层包含的协议有：ARP（地址转换协议）和 RARP（反向地址转换协议）。

- 网络层：有时也被称为互连网层，负责分组在网络中的活动，包括 IP（网际协议）、ICMP（Internet 互联网控制报文协议）以及 IGMP（Internet 组管理协议）。
- 传输层：该层主要为两台主机上的应用程序提供端到端的数据通信，它分为两个不同的协议：TCP（传输控制协议）和 UDP（用户数据报协议）。TCP 提供端到端的质量保证的数据传输，该层负责数据的分组、质量控制和超时重发等，对于应用层

来说，就可以忽略这些工作。UDP 则只提供简单的把数据报从一端发送到另一端，至于数据是否到达或按时到达、数据是否损坏都必须由应用层来做。这两种协议各有用途，前者可用于面向连接的应用，而后者则在及时性服务中有着重要的用途，如网络多媒体通信等。

- 应用层：该层负责处理实际的应用程序细节，包括大家都十分熟悉的 Telnet（电子公告板）、HTTP（World Wide Web 服务）、SMTP（简单邮件传输协议）、FTP（简单文件传输协议）和 SNMP（简单网络管理协议）等著名协议。

为了更好地理解 TCP/IP 的 4 层结构，下面通过一个具体的示例对其进行说明，示例的网络结构如图 1-3 所示。

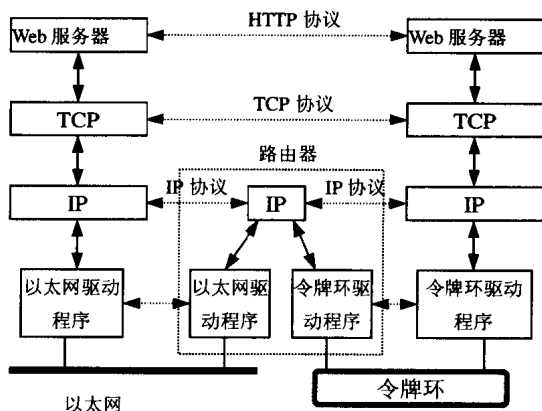


图 1-3 通过 TCP/IP 和路由器连接的两个网络

在图中所示的系统中，两个网络通过路由器互相连接。路由器可以用来把以太网、令牌环、点对点链接和 FDDI（光纤分布式数据接口）等不同的网络连接到一起。

协议中的各层对上一层是透明的，也就是说，应用层只负责应用程序之间的通信规则，完全不必理会数据是怎样在网络中传输，也不必理会它是怎样接入网络。而对于链路层来说，它完全不需要知道正在传送的是什么数据，只需要负责传输就行了。

这里顺便介绍一下网桥、路由器和网关的差别。网桥是一种在链路层将不同类型的局域网连接成一个更大的局域网的网络设备，路由器则是在网络层实现该功能，而网关是指连接不同协议族的进程（例如 TCP/IP 和 IBM 公司的 SNA），它为某个特定的应用程序（常常是电子邮件或文件传输）服务。

1.1.3 常用协议

下面对一些常用的协议进行简要的介绍。本书的后面章节将对其中 TCP、UDP、IP 层的协议进行详细的介绍。

1. TCP 和 UDP 层协议

这两个协议都使用 IP 作为网络层协议。TCP 的全称是 Transmission Control Protocol，即传输控制协议。在网络通信传输机制中，它属于“面向连接，可靠传输”的类型。这一点如果和 UDP 进行比较就会看得比较清楚。面向连接的传输意味着在进行通信以前，需要在两个

系统之间建立逻辑连接，在每个数据传输的过程中都需要进行应答以保证数据包的完整。这种方法需要的网络开销较大，可是数据传输的可靠性可以保证。虽然 TCP 使用不可靠的 IP 服务，但它却提供一种可靠的传输层服务。本书后面章节将详细讨论基于 TCP 的各种应用，如 Telnet、HTTP、FTP 和 SMTP 等。这些应用通常都是用户进程。

UDP (User Datagram Protocol)，即用户数据报协议。它属于“面向无连接，不可靠传输”的类型。这一点必须注意。该协议只负责接收和传送由上层协议传递的消息，它本身不做任何检测、修改与应答，上层协议需要自己处理这些事务。

UDP 中，每个数据包成为“数据报”，它的包头只包括 4 个域，主要是地址信息与包的长度和校验信息。与此对应，TCP 包的头信息有十多个域。因此它的网络开销一般要小于 TCP。

由于 UDP 在传送数据过程中没有建立连接，亦不进行检查，因此在良好的网络环境中，其工作的效率较 TCP 要高。目前使用 UDP 工作的软件主要有 OICQ。由于 UDP 的特点，亦是进行网络广播的首选协议。传统的 NFS (Net File System，网络文件系统) 使用 UDP。

UDP 为应用程序发送和接收数据报。一个数据报是指从发送方传输到接收方的一个信息单元 (例如，发送方指定的一定字节数的信息)。但是与 TCP 不同的是，UDP 是不可靠的，它不能保证数据报能安全无误地到达最终目的。基于 UDP 的应用层协议有很多包括 DNS (域名系统)、TFTP (简单文件传送协议)、BOOTP (引导程序协议) 和 SNMP (简单网络管理协议)。

2. IP 层协议

IP 负责在 TCP/IP 主机之间提供数据报服务，进行数据封装、产生协议头。该协议是 TCP 与 UDP 的基础。TCP 和 UDP 的每组数据都通过端系统和每个中间路由器中的 IP 层在互联网中进行传输。ICMP 作为 IP 协议的附属协议，用来与其他主机或路由器交换错误报文和其他重要信息。

IP 层协议的另一个附属协议是 IGMP (Internet 组管理协议)，它用来把一个 UDP 数据报多播或组播到多个主机。

由于在以太网中帧的大小是有限制的，因此 IP 协议需要将较大的数据报文分割，在目的主机则负责将这些打散地包重新组合。由于不同的包段可能是由不同的网络路径传送的，因此 IP 协议还需要负责将这些包按正确的顺序重新组合。注意一点，IP 协议也是不负责包的校验的，它也是一种无连接不可靠传输。不可靠 (Unreliable) 的意思是它不能保证 IP 数据报能成功地到达目的地。如果发生某种错误，IP 协议有一个简单的错误处理方法，丢弃该数据报，然后发送 ICMP 消息报给信源端。数据包的检测校验是由上层协议如 TCP 等负责的。无连接 (connectionless) 这个术语的意思是 IP 并不维护任何关于后续数据报的状态。每一个数据包都是独立的。

IP 协议还需要负责寻找路由，因此它还需要配套一个确定的 IP 地址。在 IP 报文的包头中包含了源与目的的 IP 地址。一般来说不会有应用程序直接访问 IP 协议。

3. ICMP 协议

ICMP 协议全称为 Internet Control Message Protocol，即 Internet 控制报文协议。ICMP 协议其实是 IP 协议的附属协议，IP 协议用它来与其他主机或路由器交换错误报文和其他的一

些网络情况。在 ICMP 包中携带了控制信息和故障恢复信息，这些信息可以用于：

- 源抑制：这是一个流控制信息，通过由接收方向源主机发送来请求源主机停止发送数据。在接收主机在其缓冲区快满时发送。
- 路径重定向：由网关向请求其提供服务的主机发送，用于通知该主机在网络中还有其他的距离目的主机跟近的网关。
- 主机不可到达：在网络状况不佳的网络中传送数据报时，发生故障（如链路失效、链路堵塞、主机失效等）的网关或者系统会发出此消息。在 ICMP 报文中通常包括失效的原因。
- 应答请求与回复：用于 ping 来检测目标主机是否可以到达。ping 指令调用 ICMP 消息的应答请求功能发送数据报，如果远程主机是可以到达的，则该系统会用应答回复功能来相应。

ICMP 协议的主要指责就是用于路由器或者主机向其他的路由器或者主机发送出错报文和控制信息。

尽管 ICMP 协议主要被 IP 协议使用，但应用程序也有可能访问它，在 Windows 系统中有专门的 DLL 接口可以使用。除去前面所说的 ping，Traceroute 是另外一个流行的诊断工具，它们都使用了 ICMP 协议。

4. 数据链路层协议

ARP（地址解析协议）和 RARP（逆地址解析协议）是某些网络接口（如以太网和令牌环网）使用的特殊协议，用来转换 IP 层和网络接口层使用的地址。

ARP 的全称为 Address Resolution Protocol，即地址解析协议。每一块网卡（NIC, Network Interface Card）都有一个惟一的硬件地址（由网卡的生产厂商设置的，需要使用特殊的方式才可以修改）。这个硬件地址称为 MAC（Medium Access Layer）。一块网卡依据数据帧的包头信息中是否写有它的 MAC 地址来决定是否接受并上传该帧。

分配给主机使用的 IP 地址和它固有的 MAC 地址是互不相干的。IP 地址只对 TCP/IP 有效，MAC 地址只对网络访问层有意义。在物理网络上的数据帧交换依赖于 MAC 地址，ARP 协议实现了从 IP 地址到 MAC 地址的映射。

RARP 的全称为 Reverse Address Resolution Protocol，即逆向地址解析协议。它负责根据 NIC 硬件地址去查询对应的 IP 地址。

1.1.4 进程/应用层协议

进程/应用层的协议是平时使用最广泛的协议，这层的每个协议都由两部分组成：客户程序和服务程序。程序通过服务器与客户机的交互来工作。

1. Telnet 协议

Telnet 是网络虚拟终端协议的一个典型例子，该协议允许用户通过 Internet 登录到远程计算机中。客户程序需要自己实现 Telnet 协议，同时在某些键以及显示的特性上，不同的终端类型定义是不一样的，目前的大多数终端支持 DEC 的 VT100 终端类型。在进行 Telnet 协议的实现时，工作量最大的还在于使自己的客户程序适应不同的终端类型上。

关于该协议的实现请参阅本书的“TCP 相关编程”的 TELNET 相关部分。

2. FTP 协议

FTP (File Transfer Protocol), 即文件传输协议。在该协议中, 要求使用者是经过授权的用户, 从而使文件的访问具有一定的安全限制。由于 FTP 口令在网络上传输的是明文的, 因此一旦被监听泄密就会威胁到系统安全。许多站点也提供匿名 FTP 服务 (Anonymous FTP), 在这类站点上, 通常的登录用户名为 Anonymous, 口令按照惯例是 Guest, 也有可能为用户的 E-mail 地址, 当然提供一个假的地址也可以通过审查。

使用 FTP 协议可以在提供 FTP 功能的服务器上进行文件检索与传送等操作。经常使用的 FTP 客户端包括 CuteFtp, LeapFtp 等, 还有众多的个人开发产品。

关于该协议的实现请参阅本书的“TCP 相关编程”的 FTP 相关部分。

3. HTTP 协议

HTTP (Hypertext Transfer Protocol, 超文本传输协议) 是 WWW 服务程序所用的协议。也是目前在 Internet 中使用最广泛的协议。著名的 HTTP 协议客户端包括微软公司的 IE (Internet Explorer), 以及 Netscape 公司的 Netscape Communicator 和 Netscape Navigator 等, 还有众多的如 Opera 这类出色而小巧的 WWW 浏览器。这些小型的浏览器中也有很多是由个人进行开发的。

通过 WWW 方式, 可以进行信息查询、文件下载、访问 WWW 方式的 E-mail, 在线聊天等, 功能非常强大。

关于该协议的实现请参阅本书的“TCP 相关编程”的 HTTP 相关部分。

4. SMTP 与 POP3 协议

SMTP 是 Simple Mail Transfer Protocol 的缩写, 即简单邮件传送协议。POP3 是 Post Office Protocol 3 的缩写, 即邮局协议。通过这两个协议就可以实现 E-Mail 的收发功能。

目前国内使用最广泛的电子邮件客户端中, Foxmail 就是由个人开发的, 使用的开发工具是 Delphi。

5. DNS

DNS (Domain Name Service, 即域名服务) 实现了由主机名到 IP 地址的映射功能。关于该协议的实现请参阅本书的“基本网络编程技术”的“获取 IP 地址”部分。

1.1.5 RFC 和标准简单服务

所有关于 Internet 的正式标准都以 RFC (Request For Comment) 文档出版。不过, 大量的 RFC 并不是正式的标准, 出版的目的是为了提供信息。RFC 的篇幅从 1 页到 200 页不等。每一项都用一个数字来标识, 如 RFC112; 数字越大说明 RFC 的内容越新。

所有的 RFC 都可以通过电子邮件或用 FTP 从 Internet 上免费获取。如果发送下面这份电子邮件, 就会收到一份获取 RFC 的方法清单:

To: rfc-info@ISI.EDU