



# Visual Basic

## 网络通信协议分析与 应用实现

汪晓平 钟军 等 编著

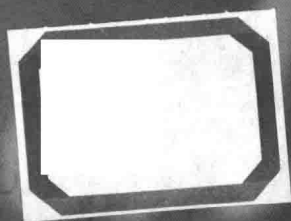


4



附光盘  
CD-ROM

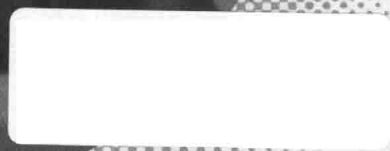
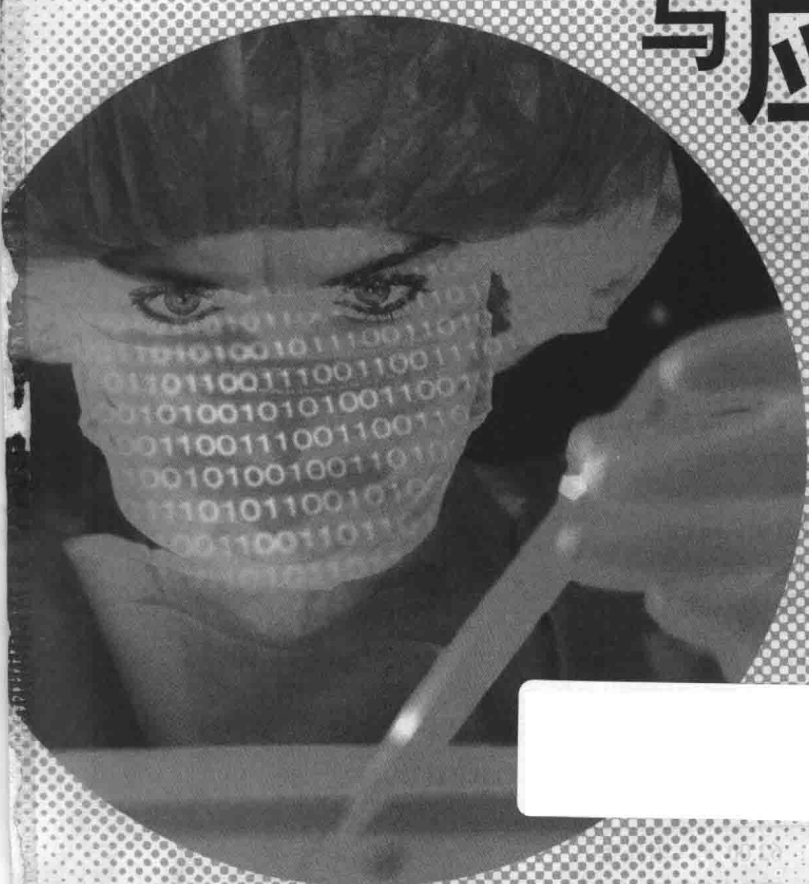
人民邮电出版社  
POSTS & TELECOMMUNICATIONS PRESS



*Visual Basic*

# 网络通信协议分析 与应用实现

■ 汪晓平 钟军 等 编著



人民邮电出版社

## 图书在版编目 ( C I P ) 数据

Visual Basic 网络通信协议分析与应用实现/汪晓平, 钟军等编著. —北京: 人民邮电出版社, 2003.1

ISBN 7-115-11004-2

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

中国版本图书馆 CIP 数据核字 (2002) 第 109071 号

## 内容提要

本书主要针对目前流行的 FTP、HTTP、E-mail、Telnet、ICMP、Modem 串口通信编程、拨号网络编程等内容进行详细的讲解, 并结合大量的实例使读者能够深入地了解各种网络应用程序的开发技巧。除了深入剖析各种网络协议之外, 本书还介绍了 Visual Basic 6.0 中各种开发网络通信的方法和技巧, 以及各种网络通信的基础应用。

本书适合中高级 Visual Basic 程序员阅读、参考。

### Visual Basic 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: 42.5  
字数: 1 045 千字 2003 年 1 月第 1 版  
印数: 1-5 000 册 2003 年 1 月北京第 1 次印刷

---

ISBN7-115-11004-2/TP · 3304

---

定价: 62.00 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

Internet 有强大的通信功能,如文件传送、远程登录、E-mail、Internet Phone 和 Internet Fax 等,它使传统的电信产业发生了巨大变化。同时,它是一个大型信息资源库,所含信息不仅包罗万象,而且日新月异。尤其是 WWW (World Wide Web) 的出现使全球信息联成一体,并使千家万户可以随时共享这一人类伟大的资源。对于任何一位软件开发来说,学习开发 Internet 应用程序已经是一件刻不容缓的事情了。串口通信虽然历史悠久,但是它在目前的通信领域,仍然占据着重要的地位,在许多的工业领域广为使用。

Visual Basic 简单易懂,只要写少量的代码,就可以实现特定功能。不仅是对于专业程序员而且对于那些特别希望能够尽早看到自己的编程成果的业余程序员来说,这无疑是个好消息。同时可以通过 Visual Basic 进行程序的快速开发,可以迅速增加对新知识和内容的体验,而不是把大部分的精力放在怎么优化程序代码,以及解决编程过程中出现问题等细节上。

对于网络和通信编程而言,使用 Visual Basic 无疑是一个能够快速开发网络应用程序的选择。本书主要内容针对 Visual Basic 进行网络高级开发,本书的特点体现在以下几个方面:

- 突出网络高级编程,重点讲解网络编程中的使用技巧及难点;
- 针对比较流行的协议如 Telnet、FTP、HTTP、E-mail 等进行详细的理论讲解;
- 每一章都有丰富的实例讲解,使读者通过实例深入了解各个协议的内容。

本书共包含 10 章,各章内容的介绍如下:

## 第 1 章:网络编程基本知识

本章主要介绍的是有关 TCP/IP 协议、Winsock 编程的基础知识,安排本章的内容是为了让读者能对后面的程序更容易理解。

## 第 2 章:网络与通信控件

本章介绍了 Visual Basic 中进行网络通信开发的各种控件的属性、方法和事件,为后面进行各种网络通信应用的开发提供坚实的基础。

## 第 3 章:实现网络基本应用

本章介绍几种读者比较感兴趣的网络基础应用,这些都是进行网络编程不可缺少的,如端口扫描程序、Ping 程序的实现、根据域名获得计算机的 IP 地址、获取网卡地址、超级链接和发送 E-mail 等。

## 第 4 章:TCP/UDP 编程实例

本章是对第 2 章的各种网络技术的综合,通过各种聊天程序的开发,让读者对各种网络技术的开发有一个升华,弄清楚了这些编程的原理,也是进行后面各种网络协议开发的基础。

## 第 5 章:E-mail 协议及高级编程

本章首先通过介绍 SMTP 或 POP3 了解收发 E-mail 的工作原理,结合程序实现收发 E-mail,接着详细分析 E-mail 的内容结构及格式,在这一章中还详细讲解了发送附件的 E-mail 格式以及各种编码解码的算法和相应的源程序,并且对 E-mail 的乱码进行了分析。最后讲解

设计 E-mail 程序的另一种方法——MAPI。对于每个理论（SMTP 发送 E-mail，POP3 接收 E-mail，E-mail 的编码解码，发送 E-mail 附件，MAPI）都有相应的实例源程序介绍。

#### 第 6 章：Telnet 协议及高级编程

本章不但详细讲解了 Telnet 的基本理论，而且给出了 BBS 客户端的源程序。

#### 第 7 章：HTTP 协议及高级编程

本章提供了类似网络蚂蚁功能的源程序、WWW 服务器源程序、下载整个站点的应用程序的源代码等。同时向读者揭示了断点续传、代理服务器的工作原理。

#### 第 8 章：FTP 高级编程

本章介绍了从使用 Internet Transfer 控件，到 FTP API 函数，甚至是使用 Winsock 进行较底层的通信应用开发。

#### 第 9 章：RAS 高级编程

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 等内容。

#### 第 10 章：串口通信高级编程

本章首先介绍了如何利用 MSComm 控件进行数据通信、文件传输，从基本理论分析入手，最后提供功能丰富的串口通信程序，能进行文件传输、自动监控和应答等等。接着介绍了如何利用 Windows API 进行串口高级编程，介绍了各种 API 的含义、以及开发步骤，并在最后提供了一个串口通信程序。相信通过最后两章的学习，读者能够开发出各种功能的串口通信程序。

本书主要由钟军、汪晓平编写而成，此外参与资料收集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、闫海荣、刘理、谭春华、张益贞、刘韬、刘鹏、黄良大、周金萍、黄超、王非、冉光志、金海、赵振、孙越、纪慧波、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，书中错误、浅陋之处在所难免，恳请广大读者批评指正。本书责任编辑的 E-mail: zhanglike@ptpress.com.cn, 作者的 E-mail: busywxp@163.com, 欢迎联系讨论。

编者

# 目 录

第 1 章 网络编程基础知识 .....	1
1.1 网络的基本应用 .....	1
1.2 TCP/IP 模型 .....	1
1.3 UDP 数据包 .....	5
1.4 TCP 数据包 .....	5
1.5 小结 .....	11
第 2 章 网络与通信控件 .....	12
2.1 Winsock 规范 .....	12
2.2 Winsock 控件 .....	14
2.2.1 TCP 协议基础知识 .....	14
2.2.2 UDP 基础知识 .....	14
2.2.3 Winsock 控件属性 .....	15
2.2.4 Winsock 控件方法 .....	18
2.2.5 Winsock 控件事件 .....	21
2.2.6 Winsock 控件的使用 .....	24
2.3 Internet Transfer 控件 .....	30
2.3.1 Internet Transfer 控件属性 .....	30
2.3.2 Internet Transfer 控件方法 .....	35
2.3.3 Internet Transfer 控件事件 .....	39
2.3.4 Internet Transfer 控件的使用 .....	40
2.4 MSComm 控件 .....	45
2.4.1 MSComm 控件的属性 .....	45
2.4.2 MSComm 控件的事件 .....	49
2.4.3 利用 MSComm 控件通信步骤 .....	49
2.5 Winsock API .....	49
2.5.1 Winsock API 的函数声明 .....	49
2.5.2 WinsockAP 的函数使用 .....	69
2.6 串口通信 API .....	73
2.6.1 打开和关闭串口 .....	73
2.6.2 串口配置和串口属性 .....	76
2.6.3 读写串口 .....	86
2.6.4 通信事件 .....	94
2.6.5 设备控制命令 .....	96
2.7 小结 .....	97

第 3 章 实现网络基本应用 .....	98
3.1 端口扫描程序 .....	98
3.2 Ping 程序的实现 .....	102
3.3 根据域名或者计算机名获取 IP 地址 .....	112
3.3.1 获取本机 IP 地址 .....	113
3.3.2 根据域名或者计算机名获取 IP 地址 .....	114
3.4 获取网卡地址 .....	118
3.5 增加超级链接和发送 E-mail .....	122
3.6 小结 .....	124
第 4 章 TCP/UDP 应用开发 .....	125
4.1 Winsock API 实现 TCP 聊天 .....	125
4.1.1 建立工程项目 .....	125
4.1.2 代码分析 .....	127
4.2 Winsock API 实现 UDP 聊天 .....	134
4.2.1 建立工程项目 .....	134
4.2.2 代码分析 .....	136
4.3 Winsock 控件实现 TCP 聊天 .....	141
4.3.1 建立工程项目 .....	141
4.3.2 代码分析 .....	142
4.4 Winsock 控件实现 UDP 聊天 .....	148
4.4.1 建立工程项目 .....	149
4.4.2 代码分析 .....	150
4.5 小结 .....	152
第 5 章 E-mail 协议及高级编程 .....	153
5.1 Foxmail 发送接收 E-mail .....	153
5.2 SMTP、POP3 与 E-mail .....	157
5.3 SMTP 及发送电子邮件 .....	157
5.3.1 SMTP 的模型描述 .....	157
5.3.2 SMTP 的会话过程 .....	157
5.4 发送无附件 E-mail 程序 .....	166
5.4.1 建立工程项目 .....	166
5.4.2 代码分析 .....	167
5.5 POP3 与接收电子邮件 .....	171
5.5.1 POP3 的模型描述 .....	171
5.5.2 POP3 的会话过程 .....	171
5.6 接收 E-mail 的程序 .....	179
5.6.1 建立工程项目 .....	179
5.6.2 代码分析 .....	180

5.7	信件结构详述 .....	192
5.7.1	RFC822 信件的格式和内容 .....	192
5.7.2	构造符合 RFC822 的信件 .....	200
5.7.3	RFC822 信件的语法分析 .....	202
5.8	MIME 编码解码与发送附件 .....	204
5.8.1	RFC822 的局限性 .....	204
5.8.2	Uuencode 编码与解码 .....	204
5.8.3	MIME 及其编码 .....	209
5.8.4	构造 MIME 信件 .....	232
5.8.5	MIME 信件的语法分析 .....	234
5.9	E-mail 客户端高级编程 .....	235
5.9.1	建立工程项目 .....	235
5.10	E-mail 乱码 .....	239
5.10.1	乱码的常见形式及形成原因 .....	239
5.10.2	避免乱码的方法 .....	240
5.11	MAPI 概述 .....	241
5.11.1	Windows 的 MAPI 介绍 .....	241
5.11.2	在 VB 中使用 MAPI .....	241
5.12	MAPI 高级编程 .....	247
5.12.1	建立工程项目 .....	247
5.12.2	代码分析 .....	248
<b>第 6 章</b>	<b>Telnet 协议及高级编程 .....</b>	<b>262</b>
6.1	Telnet 简介 .....	262
6.2	使用 Windows 的 Telnet 程序登录远程服务器 .....	263
6.3	深入 Telnet 协议 .....	264
6.3.1	NVT ASCII 字符集 .....	264
6.3.2	Telnet 命令 .....	264
6.3.3	协商选项 .....	266
6.3.4	子协商选项 .....	267
6.3.5	Telnet 操作方式 .....	267
6.4	BBS 客户端高级开发 .....	268
6.4.1	建立工程项目 .....	269
6.4.2	关键代码分析 .....	269
<b>第 7 章</b>	<b>HTTP 协议及高级编程 .....</b>	<b>296</b>
7.1	HTTP 协议介绍 .....	296
7.1.1	HTTP 背景 .....	296
7.1.2	HTTP 的内容 .....	299
7.1.3	消息 (Message) .....	300



7.1.4	请求 (Request)	301
7.1.5	响应 (Response)	305
7.1.6	访问认证	308
7.1.7	URL 编码	310
7.1.8	HTTP 协议的应用	311
7.2	断点续传	312
7.2.1	建立工程项目	312
7.2.2	代码分析	313
7.3	网页服务器高级开发	342
7.3.1	Web Server 的一些理论	342
7.3.2	建立工程项目	343
7.3.3	代码分析	345
7.4	网站下载程序高级开发	372
7.4.1	实例介绍	372
7.4.2	WinInet HTTP API 实现文件下载的使用方法	373
7.4.3	代码分析	375
7.5	HTTP API 高级开发	399
7.5.1	实例介绍	400
7.5.2	WinInet HTTP API 实现断点续传	400
7.5.3	关键代码分析	404
7.6	HTTP 代理服务器高级开发	425
7.6.1	建立工程项目	426
7.6.2	代码分析	429
<b>第 8 章</b>	<b>FTP 协议及高级编程</b>	<b>433</b>
8.1	FTP 简介	433
8.2	安装设置 FTP 服务器	434
8.3	使用 Windows 内置 FTP 程序	439
8.4	深入 FTP 协议	441
8.4.1	FTP 命令大全	441
8.4.2	FTP 工作模式	460
8.5	Internet Transfer 控件实现 FTP 程序	461
8.5.1	建立工程项目	461
8.5.2	关键代码分析	462
8.6	Winsock 开发高级 FTP 客户端程序	475
8.6.1	建立工程项目	475
8.6.2	关键代码分析	477
8.7	API 开发高级 FTP 客户端程序	521
8.7.1	建立工程项目	521
8.7.2	关键代码分析	522

8.8	3种FTP客户端程序开发方法的比较 .....	542
<b>第9章</b>	<b>RAS高级编程 .....</b>	<b>543</b>
9.1	RAS客户机 .....	543
9.2	建立拨号连接 .....	544
9.3	RAS简单拨号程序 .....	548
9.4	RAS重要函数说明 .....	549
9.4.1	连接函数 .....	549
9.4.2	连接管理函数 .....	554
9.4.3	电话簿和用户凭证管理 .....	557
9.4.4	拨号方式 .....	558
9.5	RAS高级程序开发实例 .....	560
9.5.1	建立工程项目 .....	560
9.5.2	程序运行结果图 .....	561
9.5.3	关键代码分析 .....	565
9.5.4	RAS编程小结 .....	611
9.6	RAS应用实例——远程文件共享 .....	612
<b>第10章</b>	<b>串口通信高级编程 .....</b>	<b>616</b>
10.1	串口通信中字符传输 .....	616
10.1.1	ASCII控制字符 .....	616
10.1.2	通信中的字符和字节 .....	618
10.2	MSComm控件编程实例 .....	619
10.2.1	建立工程项目 .....	619
10.2.2	代码分析 .....	619
10.3	Windows API串口通信高级实例 .....	637
10.3.1	VB中调用Windows API .....	637
10.3.2	建立工程项目 .....	638
10.3.3	代码分析 .....	639

# 第 1 章 网络编程基础知识

## 1.1 网络的基本应用

目前网络的应用有许多方面，而对于广大的编程人员来说，在应用程序开发方面，应用最为广泛的就是以下几种类型。

- HTTP 协议的开发
- FTP 协议的开发
- Telnet 协议的开发
- E-mail 协议开发
- 新闻组协议的开发

当然还有其他的许多协议，以上列举的只是目前比较通用的一些协议。其实在实际的开发过程中，完全可以根据需要，自定义开发各种不同的协议以满足自己的需要。在本书中所涉及到的网络编程，基本上是网络的高级应用，而不涉及到网络的底层编程。网络程序的开发是包括许多方面的，在后面读者会知道，ISO 把网络结构划分为 7 层，每一层都可以进行程序开发。通常说的网络编程也就是最上层即应用层程序的开发。

上面列举出了 5 种开发协议，从这 5 种开发协议可以引申出许多的网络开发。其中较为复杂的就是 HTTP 协议的程序开发。目前，在网络上，最流行的也莫过于 WWW 应用，而它正是基于 HTTP 协议的。通过 HTTP 协议，可以进行许多方面的编程，比如 WWW 服务器的开发、网络浏览器的开发、代理服务器开发、CGI 的开发、ASP 开发（PHP、JSP 等）、网络搜索引擎开发等。

## 1.2 TCP/IP 模型

TCP/IP 是协议的集合，其名称代表它的两个基本协议，TCP（Transmission Control Protocol，传输控制协议）和 IP（Internet Protocol，互联网协议）。虽然从名称上来说只有两个协议，其实它包含了 4 层协议。

### 1. TCP/IP 协议的分层

TCP/IP 通常被认为是一个由 4 层协议系统，由以下 4 层组成：

- 应用层
- 传输层
- 网络层
- 链路层

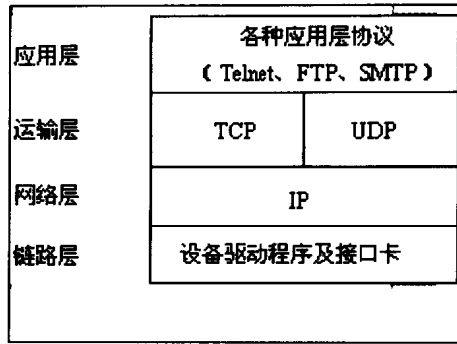


图 1-1 TCP/IP 协议族的体系结构

注意这里只有 4 层，它同 ISO 规定 7 层协议 OSI（开放系统互联）是有所区别的，稍后再介绍 OSI 系统。上面提到的每一层都有不同的功能。

■ 链路层

有时候也称为数据链路层或者网络接口层，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。它们一起处理电缆（或者其他任何传输媒介）的物理接口细节。

■ 网络层

有时候也被称为互联网层，处理分组在网络中的活动，例如分组的选录，在 TCP/IP 协议族中，网络层包括 IP 协议（网际协议）、ICMP 协议（Internet 互联网控制报文协议），以及 IGMP 协议（Internet 组管理协议）。

■ 传输层

传输层主要为两台主机上的应用程序提供端到端的通信。在 TCP/IP 协议族中，有两个互不相同的传输协议：TCP（传输控制协议）和 UDP（用户数据包协议）。TCP 为两台主机提供可靠的数据通信。它所作的工作包括把应用程序交给它的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置发送最后确定分组的超时时钟等。由于传输层提供了高可靠性的端到端的通信，因此应用层可以忽略所有的细节。而另一方面，UDP 则为应用层提供一种非常简单的服务。它只是把称作数据包的分组从一台主机发送到另外一台主机，但并不保证该数据包能到达另一端。任何必需的可靠性必须由应用层来提供。

■ 应用层

应用层负责处理特定的程序细节，在本书中的所有应用程序全部是在应用层进行的。

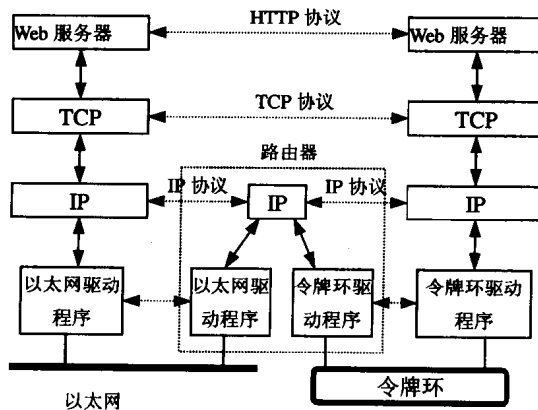


图 1-2 通过 TCP/IP 和路由器连接的两个网络

## 2. OSI 模型

OSI 网络模型主要是由 ISO（国际标准化组织）规定的一套协议。OSI 为 Open System Interconnection 的简写。实际上该模型是源自 TCP/IP 协议，但比 TCP/IP 多了 3 层，一共有 7 层协议。具体分层如下：

- 应用层（使用网络应用程序）
- 表示层（应用程序的标准化数据表示）
- 会话层（管理应用程序之间的会话）
- 传输层（提供端对端的错误检测和校正）
- 网络层（为上一层管理网络之间的连接）
- 数据链路层（提供跨越物理链路的可靠数据传输）
- 物理层（定义网络媒体的物理特性）

## 3. 网际地址

互联网上的每一个接口必须有唯一的 Internet 地址也就是 IP 地址。一个 IP 地址通常由 4 个用点隔开的 8 个二进制位的数字（从 0 到 255）组成的，地址总长为 32 比特。目前 IP 地址共分为 5 类即 A、B、C、D、E。其中具体的分布情况如下：

A 类：从 0.0.0.0 到 127.255.255.255

B 类：从 128.0.0.0 到 191.255.255.255

C 类：从 192.0.0.0 到 223.255.255.255

D 类：从 224.0.0.0 到 239.255.255.255

E 类：从 240.0.0.0 到 247.255.255.255

注意，多接口的主机具有多个 IP 地址，其中每个接口都对应一个 IP 地址。由于互联网上的每一个接口必须有一个唯一的 IP 地址，因此必须要由一个管理机构为介入互联网的网络分配 IP 地址。这个管理机构就是互联网络信息中心，称作 InterNIC。

每个 32 位 IP 地址被分割成两部份：前缀和后缀。前缀用于确定计算机从属的物理网络，后缀则用于确定网络上一台单独的计算机。互联网中每一个物理网络都有一个唯一的值作为网络号（Network Number）。IP 地址的层次性保证了以下两个重要性质：

- 每台计算机分配一个唯一的地址
- 虽然网络号分配必须全球一致，但后缀可本地分配，不须全球一致

此外，需要特别注意以下几个特殊的 IP 地址：

- 网络地址：IP 中主机地址为 0 的地址表示网络地址。如 128.211.0.0
- 广播地址：网络号后跟一个所有位全是 1 的后缀，就是直接广播地址
- 回送地址：127.0.0.1 用于测试

除了每个主机分配一个 IP 地址外，IP 协议规定也应给路由器分配一个 IP 地址。事实上，每个路由器被分配了两个或更多的 IP 地址。一个路由器连接到多个物理网络，每一个 IP 地址包含一个特定物理网络的网络号。一个 IP 地址并不标识一台特定的计算机，而是标识一台计算机和一个网络间的一个连接。

现在所有的主机都要求支持子网编址（RFC 950 [Mogul and Postel 1985]）。不是把 IP 地址看成由单纯的一个网络号和一个主机号组成，而是把主机号再分成一个子网号和一个主机号。

这样做的原因是因为 A 类和 B 类地址为主机号分配了太多的空间，可分别容纳的主机数为  $2^{24}-2$  和  $2^{16}-2$ ，而事实上，在一个网络中人们并不安排这么多的主机。在 InterNIC 获得某类 IP 网络号后，就由当地的系统管理员来进行分配，由他（或她）来决定是否建立子网，以及分配多少比特给予子网号和主机号。例如，这里有一个 B 类网络地址（140.252），在剩下的 16 位中，8 位用于子网号，8 位用于主机号，其格式如图 1-3 所示。这样就允许有 254 个子网，每个子网可以有 254 台主机。

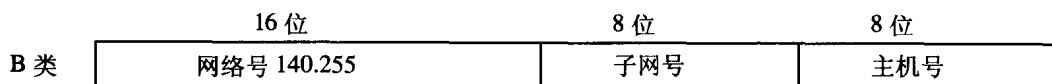


图 1-3 B 类地址的一种子网编址

除了地址以外，主机还需要知道有多少比特用于子网号及多少位用于主机号。这是在引导过程中通过子网掩码来确定的。这个掩码是一个 32 位的值，其中值为 1 的比特留给网络号和子网号，为 0 的比特留给主机号。在上面的例子中，子网掩码就是 255.255.255.0。

通常在记忆 IP 地址的时候比较困难，因此在实际的应用中，会经常使用到域名。在进行网络程序开发的时候，必须要给出远程主机的地址，一般可以使用 IP 地址或者使用域名。

这里要注意，因为域名通常要经过 DNS（域名服务器）的解析，所以比直接使用 IP 地址会慢一些。因为网络不理解域名，只能理解 IP 地址。将名称解析为地址的方法通常有两种。古老的方法是将所有的名称与地址存放在一个名为 HOSTS 的文本文件中。在 UNIX 中，这个表格可从一个共享目录中访问。在 Windows 95 中，这个文件位于网络上的每一台机器的“Windows”目录中。在 Windows NT 中，它驻留在“WINNT\System32\Drivers\etc”目录下。如果需要，可以由用户来编辑这个文件。

这个系统的缺点是非常明显的。因为如果每个人都使用一个 HOSTS 文件作为域名解析，就不得不把自己的 IP 地址和名称给别人，而其他人都必须更新，因此该系统只能在局域网内实现。目前实现名称解析的方法是使用 DNS（域名服务）系统。DNS 是一个分布式的数据库，它包含了所有注册过的 Internet 主机的 IP 地址，域名也是由 InterNIC 来管理的。

#### 4. 端口

在 TCP/IP 连接过程中，都是通过采用 16 位端口号来识别的。因为 IP 地址只是标志了一台机器在网络中的位置。而 IP 端口是对应了一个主机上运行的应用程序。服务器一般都是通过知名端口号来识别的。例如对于每个 TCP/IP 实现来说，FTP 服务器的 TCP/IP 端口号是 21、Telnet 的端口号是 23、TFTP（简单文件传送协议）服务器的 UDP 端口号是 69。任何 TCP/IP 实现所能提供的服务都用知名的 1~1023 之间的端口号。这些端口号由 Internet 号分配机构（IANA）来管理。

实际上，客户端对它所使用的端口号并不关心，只需要保证该端口号在本机上是唯一的就可以了。客户端口号又称作临时端口号（即存在时间很短）。这是因为它通常只是在用户运行该客户程序时才存在。而服务器只要没有停止服务，就一直运行并侦听连接。

大多数的 TCP/IP 实现给临时端口分配 1024~5000 之间的端口号。大于 5000 的端口号是为其他服务预留的。

## 1.3 UDP 数据包

### 1. UDP 数据包格式

UDP 数据包的格式如图 1-4 所示，伪首部是为了计算检验而设置的；伪首部包含 IP 首部一些字段，其目的是让 UDP 两次检查数据是否已经正确到达目的地（例如，IP 没有接收地址不是本主机的数据包，以及 IP 没有把应传给另一高层的数据包传给 UDP）。使用 UDP 协议时，“协议”字为 17。源端口是可选的，不用时时置 0。报文长度是包括头部和数据区的总长度，最小 8 个字节。校验和是以 16 位为单位，各位求补（首位为符号位）将和相加，之后再求补。

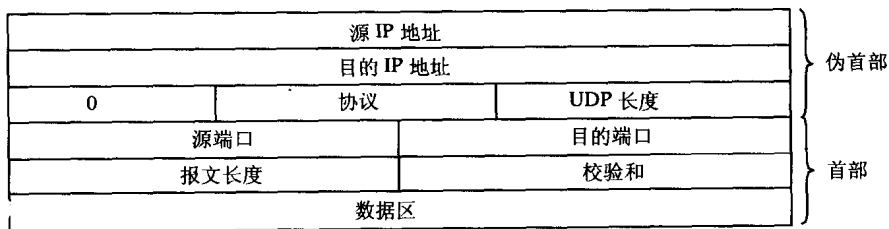


图 1-4 UDP 数据包格式

### 2. UDP 数据包的传输

理论上，IP 数据包的最大长度是 65 535 字节，这是由 IP 首部 16 比特总长度字段所限制的。去除 20 字节的 IP 首部和 8 个字节的 UDP 首部，UDP 数据包中用户数据的最长长度为 65 507 字节。但是，大多数实现所提供的长度比这个最大值小。

这样，可能会遇到两个限制因素。一个是因为应用程序可能会受到其程序接口的限制，Socket API 提供了一个可供应用程序调用的函数，以设置接收和发送缓存的长度；对于 UDP 套接字这个长度与应用程序可以读写的最大 UDP 数据包的长度直接相关，现在的大部分系统都默认提供了可读写大于 8192 字节的 UDP 数据包（使用这个默认值是因为 8192 是 NFS 读写用户数据数的默认值）。另一个限制来自于 TCP/IP 的内核实现，可能存在一些实现特性（或差错），使 IP 数据包长度小于 65 535 字节。

因为 UDP 协议是无差错控制的，所以发送过程与 IP 协议类似，就是先 IP 分组，然后再用 ARP 协议来解析物理地址，然后发送。

## 1.4 TCP 数据包

### 1. TCP 数据包格式

TCP 数据包格式如图 1-5 所示。其中，序列号字段是指该报头段在发送方字节流中的位置，确认序号是指接收方希望接收的下一个报文段序号，窗口字段是指接收缓冲区的大小，

码元比特字段是一个 6 位的标志字段，该字段每一比特（从左到右）置 1 时的意义如下：

- URG：紧急指针可用
- ACK：确认字段可用
- PSH：本报文段请求急迫 PUSH 操作
- RST：连接复位
- SYN：序号同步
- FIN：发送方字节流结束

紧急指针字段是用来处理带外数据（OOB）而设的；最大段长度用来选择适当的 MSS，目的是与 MTU 相符合，如果取值过小会导致带宽利用率太低，而如果取值过大又会造成 IP 数据包分段，若分段丢失导致重发整个报文，这样会浪费网络资源。最佳长度是使携带了长度 S 的 TCP 报文段的 IP 数据包在从源至目的地的路径上不被分段。

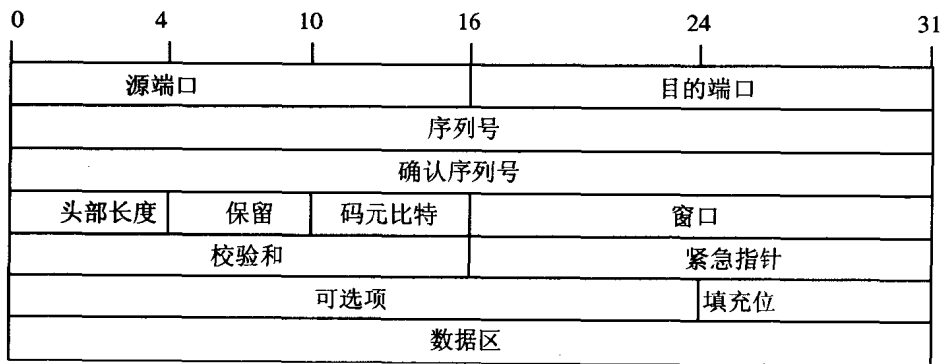


图 1-5 TCP 报头

## 2. 可靠的数据传输

可靠传输服务软件所应具有的特征如下：

- 面向数据流：数据流（stream）就是两个应用程序间传输的数据
- 虚电路连接：包括连接的建立，通信的开始，连接的结束要求所建立的连接是可靠的，连接的结束要完美（在连接终止前传送的所有数据为可靠的）
- 带缓冲的传送
- 无结构的数据流，即不考虑数据内容
- 全双工连接：包含两个独立且方向相反的连接，即 PIGGYBACKING（捎带应答）

TCP 提供了一个完全可靠的、面向连接的、全双工的、流传输服务。允许两个应用程序建立一个连接，并在全体一个方向上发送数据，然后终止连接，每一个 TCP 连接可靠地建立完善的终止，在终止发生前，所有数据都会被可靠的传送。

## 3. 滑动窗口

提供可靠性的一般方法是采用带重发的肯定确认（PAR）机制，接收方在收到数据包后给源主机发回确认信息（ACK）发送方对送出的每一个数据包均存有一份记录，在发送下一个数据包时，等待确认同时启动计时器，若超时而无 ACK 返回，则重发同一数据包。为了解决同一帧数据发两次的都收到的情况，必须对 ACK 编号，一般的编号方法就是滑动窗口



方法。

实现数据确认的最简单的方法是使用停止等待协议，该方法编号只用一位，有两种状态，每发完一帧，都必须等到确认帧后，发下一帧，如果同一帧发了两次，因为第二次收到的帧编号不符合将被舍弃。这种方法的缺点是需要等待很长的时间，优点是实现简单。另一种可能是连续发送，但这样有两个缺点，一个是如果有帧丢失，将不得不重发所有的从丢失帧开始的所有帧，另一个是这样，编号位数必然增加，导致位浪费。所以通常会对未被确认的帧的个数加以限制。这就是本节要讨论的滑动窗口协议的基础。

设发送序号用3个比特来编码，同时设发送窗口为5，也就是说帧编号可以从有0~7共8个，而允许发送的未被确认的最大帧数目是5。如果把发送窗口也设成8的话，就只有等到8个确认帧全都收到后才能继续发送了，这样的话该方法就与只用一位的停止等待协议一样了，所以通常发送窗口要小于帧编号数。

发送的过程如图1-6所示，当发送端发完0~4号帧后，如果没有收到确认帧，就必须停止等待，如果收到一个确认帧，发送窗口就沿顺时针方向旋转一个号，使窗口后沿再次与一个为被确认的帧号相邻。这时5号帧落在了发送窗口之内，就可以发送了，以后又设有三个帧收到确认信息，发送窗口再次旋转3格，这时就可以依次发送6、7和0号帧了，如果超时没有收到任何确认帧，就把发送窗口里的帧全部重发。

下面来讨论为什么发送窗口不能为8，因为刚才的讨论都忽略了一个重要的东西：确认帧丢失的情形。一旦0、1号确认帧丢失，发送端以为接受端没有收到任何帧，因此重发所有的帧（0~7），而接收端不知道发送端没有收到确认帧，以为发送端发送的是3~10号，这样就导致数据协调错误。

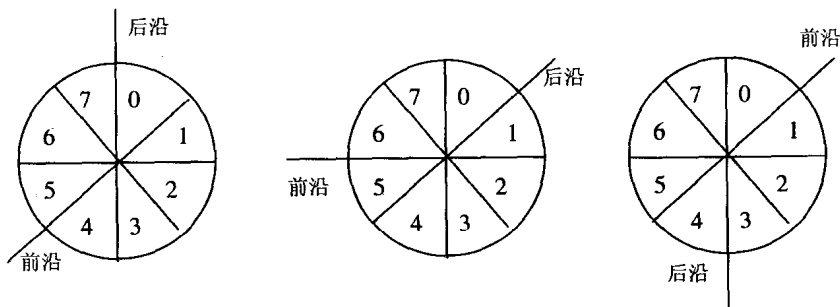


图1-6 滑动窗口示意图

在滑动窗口协议的基础上又产生了选择重传协议等一些改进版本，但是它们的使用都不如滑动窗口广泛，有兴趣的读者可以参考相关书籍。

#### 4. 超时与重发

TCP 针对互联网环境而设计的协议。由于互联网中具有多个网络和路由器，因此很难预测确认信息何时返回。TCP 采用自适应重发算法：

- 收集每一个报文段的往返时间，得来均值 RTT (ROUND TRIP TIME)。
- 将该 RTT 做加权平均值存储利用新的样本逐渐修改其值。
- 保留一个变化量利用该变化量与 RTT 的线性组合，计算出新的 RTT：

$$RTT = \alpha \times OLD\_RTT + (1 - \alpha) \times NEW\_RTT\_SAMPLE$$