



附光盘

Visual C++ 网络 高级编程

◆ 陈坚 陈伟 等 编著

本书面向具有一定 Visual C++ 基础并希望深入学习网络编程技术的读者

深入剖析 Visual C++ 的各种编程技术，系统介绍 TCP/IP 协议，

重点讲解网络编程中的高级应用、使用技巧和难点

对于每个主题，都给出其开发要领及应用的实例源代码，

读者可通过分析实例源代码获得提高



人民邮电出版社
www.pptph.com.cn

Visual C++ 网络 高级编程

◆ 陈坚 陈伟 等 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Visual C++ 网络高级编程/陈坚, 陈伟等编著.—北京: 人民邮电出版社, 2001.8

ISBN 7-115-09563-9

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

中国版本图书馆 CIP 数据核字 (2001) 第 050989 号

内容提要

本书在对 Visual C++ 各种编程技术和 TCP/IP 进行系统介绍的基础上, 重点讲解网络编程的高级应用、使用技巧和难点。包括基本网络编程技术, Telnet 协议的实现, HTTP 协议的实现, FTP 协议的实现, 文件下载, UDP 协议的实现, ICMP 协议的实现, PPP 协议的实现, 代理服务器的实现, ATL、DCOM、ActiveX 技术, 网络安全, 多媒体网络编程等。

对于每个主题, 书中都给出其开发要领及应用的实例和技巧, 本书主要面向具有一定 Visual C++ 网络编程基础并希望深入研究网络编程技术的读者。

Visual C++ 网络高级编程

◆ 编 著 陈 坚 陈 伟 等

责任编辑 张立科

执行编辑 孙玉华

◆ 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@pptph.com.cn

网址 <http://www.pptph.com.cn>

读者热线: 010-67129212 010-67129211(传真)

北京汉魂图文设计有限公司制作

北京顺义向阳胶印厂印刷

新华书店总店北京发行所经销

◆ 开本: 787 × 1092 1/16

印张: 28.5

字数: 699 千字 2001 年 8 月第 1 版

印数: 1 - 6 000 册 2001 年 8 月北京第 1 次印刷

ISBN 7-115-09563-9/TP·2411

定价: 56.00 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

Visual C++具有开发过程简明、开发方式灵活、扩展性强等特点，是一种被广泛使用的前端开发平台。Visual C++提供了多种多样的编程技术，例如 Winsock、WinInet、ActiveX、DCOM、ATL 等，为用户开发网络应用程序提供全方位的支持。

当前市场上关于 Visual C++网络编程的书籍对各种编程技术的介绍大多浅尝辄止，很少涉及到具体的网络协议，更缺乏对一些高级应用的深刻剖析，因此很难对读者独立开发专业网络应用程序有实质性的帮助，本书的编写将很好地填补这一空缺。

本书主要面向具有一定 Visual C++网络编程基础并希望学习较高层次编程技术的读者，在对 Visual C++各种编程技术和 TCP/IP 进行系统介绍的基础上，重点讲解网络编程中的高级应用、使用技巧和难点，如常用协议的实现、DCOM 和多媒体网络编程等。对于每个主题，都给出其开发要领及应用的实例和技巧，使读者通过学习实例获得提高。

本书涉及的高级内容及其实现有以下几个方面。

- **基本网络编程技术**：详细介绍了获取本地主机的网络配置信息、系统所支持的协议及其特性的编程方法，其中涉及到 Winsock 编程接口、NetBIOS 编程接口及注册表等不同的编程技术，并给出了两个综合实例。
- **Telnet 协议的实现**：在对 Telnet 协议进行简要介绍的基础上，着重对 Telnet 客户的实现方法进行介绍，并给出一个远程登录的实例。
- **HTTP 协议的实现**：在对 HTTP 协议进行简要介绍的基础上，利用 Winsock 分别实现了一个支持 CGI 的多线程 HTTP 服务器和 HTTP 客户端程序。
- **FTP 协议的实现**：对 FTP 协议进行分析并分别用 Winsock 和 WinInet 两个不同的编程接口实现了两个完整的 FTP 客户端程序。
- **文件下载**：通过一个类似于网络蚂蚁的支持断点、多线程文件下载的实例对文件下载的编程技巧进行分析。同时对另一种文件下载方法——异步 Moniker 文件进行简要的介绍。
- **UDP 协议的实现**：分别对网络广播、多播及语音通信的实现机制进行介绍，并给出两个非常综合的实例：多播讨论组及网络语音聊天。
- **ICMP 协议的实现**：在对 ICMP 协议进行简要介绍的基础上，对 ICMP 协议的两个重要应用 Ping 和 Traceroute 程序进行分析并给出它们的实现。
- **PPP 协议的实现**：主要讨论远程登录的编程方法，并给出一个拨号网络的实例。
- **代理服务器的实现**：在对代理服务器的实现机制进行简明扼要的分析的基础上，给出了一个支持二级代理的 HTTP 代理服务器的实例，并封装了一个支持 SOCKS 5 代理的类。
- **ATL、DCOM、ActiveX 技术**：分别对 DCOM、ActiveX 控制、ActiveX 文档服务器的编程方法进行介绍，并分别利用 ATL 和 MFC 实现了分布式应用程序及 ActiveX 文档服务器。

■ 网络安全：对 TCP/IP 协议簇的三个层上的安全实现进行介绍。

本书所附的光盘包括了涉及实例的全部源代码和可执行文件，并且所有实例均在 Visual C++ 6.0 环境下调试实现，有关各实例的具体用法请参考各实例的 Readme 文件。

本书由陈坚、陈伟合作编写而成，由于作者水平有限，书中疏漏之处在所难免，恳请广大读者批评指正。

编者

2001 年 8 月

目 录

第一章 TCP/IP 协议	1
1.1 TCP/IP 协议簇简介	1
1.1.1 TCP/IP 结构	1
1.1.2 常用协议	3
1.1.3 RFC 和标准简单服务	3
1.2 TCP/IP 基本概念	4
1.2.1 IP 地址	4
1.2.2 地址解析	5
1.2.3 域名系统	7
1.2.4 数据包的封装和分用	7
1.2.5 IP 数据报	8
1.2.6 UDP 数据报	11
1.2.7 TCP 数据报	12
1.2.8 端口号	17
第二章 Winsock 网络编程接口	19
2.1 Winsock 概述	19
2.1.1 套接字 (Sockets)	19
2.1.2 基本概念	20
2.1.3 Winsock 编程原理	21
2.2 一个简单的例子	28
2.2.1 服务器程序的实现	29
2.2.2 客户程序的实现	31
2.3 Winsock I/O 模型	33
2.3.1 select 模型	34
2.3.2 WSAAsyncSelect 模型	35
2.3.3 WSAEventSelect 模型	36
2.4 Winsock 2 的扩展特性	38
2.4.1 原始套接字	38
2.4.2 重叠 I/O 模型	39
2.4.3 服务的质量 (QOS)	40
2.5 套接字选项和 I/O 控制命令	41
2.5.1 套接字选项	41
2.5.2 I/O 控制命令	43

第三章 Visual C++ 与网络编程	45
3.1 WinInet 编程	45
3.1.1 MFC WinInet 类	45
3.1.2 HTTP 客户实例	57
3.1.3 利用 WinInet API 进行编程	61
3.2 MFC Windows Sockets 类	69
3.2.1 创建聊天客户	70
3.2.2 创建聊天客户	74
3.3 一个很酷的例子——在线聊天室	78
3.3.1 创建聊天客户	78
3.3.2 创建聊天服务器	93
3.4 多媒体与网络编程	100
3.4.1 窄带流传输的现状	100
3.4.2 宽带多媒体流的发展趋势	102
3.4.3 音频流的获取与处理	105
3.4.4 音频压缩	109
3.4.5 视频流的获取与压缩	109
3.5 多线程编程	111
3.5.1 Win32 基础上的多线程编程	112
3.5.2 基于 MFC 的多线程编程	115
第四章 基本网络编程技术	117
4.1 获取网络配置信息	117
4.1.1 获得主机名和 IP 地址	117
4.1.2 获取域名、子网掩码、网卡类型等信息	119
4.1.3 获取网卡的 MAC 地址	125
4.1.4 一个综合的例子	128
4.2 获取系统支持的网络协议信息	132
4.2.1 Win32 支持的协议	132
4.2.2 Winsock 2 的 WSAEnumProtocols 函数	133
4.2.3 获取系统支持的网络协议信息实例	134
第五章 Telnet 协议的实现	145
5.1 Telnet 协议	146
5.1.1 Telnet 的运行方式	146
5.1.2 网络虚拟终端 NVT	146
5.1.3 Telnet 命令	147
5.1.4 选项协商机制	148
5.1.5 子选项协商	149
5.1.6 同步信号	149

5.1.7	客户的转义符	150
5.2	Telnet 客户实例	150
第六章	HTTP 协议的实现	165
6.1	HTTP 协议	165
6.1.1	HTTP 协议的特点	165
6.1.2	HTTP 协议的运作方式	165
6.1.3	HTTP 通信实例	168
6.2	多线程 HTTP 服务器实例	169
6.2.1	封装 CRequest 类	170
6.2.2	CListenSocket 和 CRequestSocket 类	171
6.2.3	支持 CGI	185
6.2.4	程序的组织	192
6.3	HTTP 客户程序的实现	198
第七章	FTP 协议的实现	207
7.1	FTP 协议	207
7.1.1	数据表示与保存	208
7.1.2	数据连接的建立	210
7.1.3	FTP 命令	211
7.1.4	FTP 应答	214
7.2	利用 Winsock 类实现 FTP 客户	214
7.2.1	封装 FtpClient 类	215
7.2.2	程序的组织 and 界面	225
7.3	利用 WinInet 实现完整的 FTP 客户实例	229
7.3.1	界面的编写	230
7.3.2	进行实际操作的类 —— CMyAppThread	245
7.3.3	程序的组织 and 界面	253
第八章	文件下载	254
8.1	一个类似网络蚂蚁的实例	254
8.1.1	实现拖放界面	254
8.1.2	程序的组织	258
8.1.3	封装 CHTTPDownload 类	263
8.1.4	断点续传的支持	272
8.2	异步 Moniker 文件	272
8.2.1	Moniker	272
8.2.2	MFC CAsyncMoniker 类	273
8.2.3	使用 CAsyncMoniker 类	273
第九章	UDP 协议的实现	274

9.1	网络广播	274
9.1.1	局域网广播 (Broadcast)	274
9.1.2	多播通信 (MultiCast)	276
9.1.3	IP 多播的 Socket 实现	281
9.2	利用 Winsock 2 实现 IP 多播的实例	283
9.2.1	封装多播类 CMultiCast	284
9.2.2	程序的组织	288
9.3	利用 UDP 实现语音全双工通信	289
9.3.1	通信过程	290
9.3.2	音频压缩	294
9.3.3	音频的采集和回放	305
第十章	ICMP 协议的实现	313
10.1	ICMP 协议	313
10.1.1	ICMP 报文格式	313
10.1.2	回显请求和回显应答报文	315
10.1.3	ICMP 地址掩码请求与应答	315
10.1.4	ICMP 时间戳请求与应答	316
10.2	Ping 程序的实现	316
10.2.1	Ping 程序的编程原理及步骤	316
10.2.2	ICMP 报文的定义和初始化	317
10.2.3	封装 CPing 类	318
10.2.4	组织程序	325
10.3	TraceRoute 程序	326
10.3.1	Traceroute 编程的原理及实现方法	326
10.3.2	TraceRoute 实例	327
第十一章	PPP 协议的实现	332
11.1	PPP 协议	332
11.1.1	基本概念	332
11.1.2	PPP 数据帧格式	333
11.1.3	PPP 链路操作过程	334
11.2	RAS 编程	335
11.2.1	RAS 通用对话框	335
11.2.2	建立和断开 RAS 连接	338
11.2.3	管理连接	343
11.2.4	RAS 拨号实例	345
11.3	TAPI 编程接口	349
11.3.1	CTI 应用程序接口	349
11.3.2	TAPI 3.0 特性	350

11.3.3	TAPI 3.0 结构	351
11.3.4	TAPI 3.0 接口	352
第十二章	代理服务器的实现	353
12.1	Socket 5 协议	353
12.2	HTTP 代理服务器实例	356
12.2.1	封装 CProxyServer 类	356
12.2.2	使用 CProxyServer 类	366
12.3	一个支持 Socket 5 代理的类	367
12.3.1	封装 CProxySocket 类	367
12.3.2	使用 CProxySocket 类	373
12.3.3	支持 UDP 的 CAsyncProxySocket 类	374
第十三章	ATL、DCOM 和 ActiveX 技术	377
13.1	ATL 技术	377
13.1.1	ATL 与 C++ 模板类	379
13.1.2	多重继承	379
13.1.3	自动化和双重接口	380
13.1.4	ATL 的体系结构	380
13.1.5	利用 ATL 开发进程内 COM 服务器	382
13.2	DCOM	386
13.2.1	DCOM 的结构	387
13.2.2	DCOM 特性	387
13.2.3	DCOM 安全性	390
13.3	DCOM 实例——白板程序	391
13.3.1	DCOM 服务器的实现	391
13.3.2	DCOM 客户的实现	393
13.3.3	程序的运行	400
13.4	ActiveX 控制	401
13.4.1	ActiveX 控制基本结构	401
13.4.2	利用 ATL 创建 ActiveX 控件	403
13.5	ActiveX 文档服务器	408
13.5.1	ActiveX 文档特性	409
13.5.2	ActiveX 文档结构模型	410
13.5.3	ActiveX 文档服务程序的实现	410
13.5.4	ActiveX 文档服务程序实例	411
第十四章	网络安全	413
14.1	数据加密和数字签名	413
14.1.1	数据加密技术	414
14.1.2	数字签名技术	416

14.1.3 数字时间戳技术	418
14.2 网络层的安全性	418
14.2.1 IP 安全协议 (IPSP)	419
14.2.2 密钥管理协议 (IKMP)	419
14.3 传输层的安全	420
14.4 应用层的安全	421
14.4.1 修改应用层协议	421
14.4.2 SSH 认证和密钥分配系统	422
附录 1 RFC 文档分类	423
附录 2 Winsock 2 扩展函数参考	430
附录 3 Windows Sockets 错误码	443

第一章 TCP/IP 协议

毫无疑问，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之一。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，它允许分布在各地的装着完全不同操作系统的计算机互相通信。随着 PC 的普及，TCP/IP 以其开放性的特点，成为了 Internet 的基础，通过 Internet 把全世界数以千万的计算机连接在了一起。

千里之行，始于足下，一个程序员要想自如地进行网络编程，对 TCP/IP 底层协议的结构、基本概念有个充分的了解是必不可少的，但这也往往是大多数程序员的“软肋”。本章力求通过简明的论述，使读者对 TCP/IP 有个整体的把握，为后续的章节提供充分的背景知识。

1.1 TCP/IP 协议簇简介

学过计算机网络的读者一定对 OSI（Open System Interconnection）的七层模型非常熟悉，七层模型结构严密、功能强大而复杂，但是由于支持该协议的软件很少问世，TCP/IP 仍是事实上的工业标准。

1.1.1 TCP/IP 结构

TCP/IP 协议是一个四层协议，它的结构如图 1-1 所示。

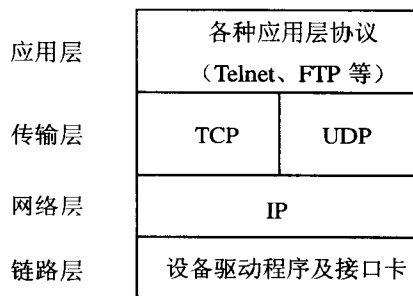


图 1-1 TCP/IP 协议簇的体系结构

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：有时也被称为互联网层，负责分组在网络中的活动，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: 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 协议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 协议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。这两种协议各有各的用途, 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 的四层结构, 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 1-2 所示。

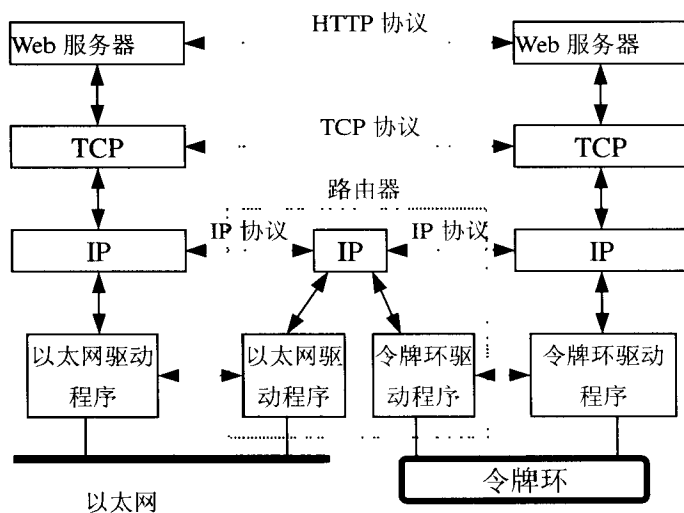


图 1-2 通过 TCP/IP 和路由器连接的两个网络

在图 1-2 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。

协议的各个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簇的进程 (例如 TCP/IP 和 IBM 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。

1.1.2 常用协议

下面简要介绍一些常用的协议。本书的后面章节将详细介绍常用的 TCP、UDP、IP 层的协议，并给出它们的实现方法。

1. TCP 和 UDP 层协议

这两个协议是最为著名两个的传输层协议，二者都使用 IP 作为网络层协议。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。本书第五、六、七、八章将详细讨论基于 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 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地。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）。本书第九章将讨论 UDP 的应用，鉴于 UDP 的服务大多比较复杂，只介绍两个比较简单却有趣的例子：多播讨论组和全双工语音通信。

2. IP 层协议

该协议同时被 TCP 和 UDP 使用。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。本书第十章将对 ICMP 协议及其两个非常重要的应用 Ping 和 Traceroute 程序进行详细介绍。

IP 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。

3. 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。

1.1.3 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。另外，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
Subject: getting rfcs
      help: ways_to_get_rfcs
```

最新的 RFC 索引总是搜索信息的起点。这个索引列出了 RFC 被替换或局部更新的时间。下面是一些重要的 RFC 文档：

- 赋值 RFC（Assigned Numbers RFC）列出了所有 Internet 协议中使用的数字和常数。所有著名的 Internet 端口号都列在这里；
- Internet 正式协议标准，这个 RFC 描述了各种 Internet 协议的标准化现状；
- 主机需求 RFC，该 RFC 包括大量的协议资料；

■ 路由器需求RFC，它与主机需求RFC类似，但是只单独描述了路由器的需求。本书附录1分类列出了一些常用RFC文档的索引。

表1-1列出一些常用的标准简单服务，它们的服务类型一般是Telnet。

表 1-1 比较普遍的标准简单服务

名字	TCP 端口号	UDP 端口号	描述
Echo	7	7	服务器返回客户发送的所有内容
Discard	9	9	服务器丢弃客户发送的所有内容
Daytime	13	13	服务器以可读形式返回时间和日期
Chargen	19	19	当客户发送一个数据报时，TCP 服务器发送一串连续的字符流，直到客户中断连接。UDP 服务器发送一个随机长度的数据报
Time	37	37	服务器返回一个二进制形式的 32 bit 数，表示从 UTC 时间 1900 年 1 月 1 日午夜至今的秒数

1.2 TCP/IP 基本概念

通过上节的学习，想必读者对 TCP/IP 协议簇已经有了一个整体的了解，但这只是认识 TCP/IP 的第一步。作为一个整体的结构体系，TCP/IP 必然要涉及到一系列基本但非常重要的概念，它们同样也是读者学习的起点。本节主要对 IP 地址、地址解析、IP 数据报、UDP 数据报、TCP 数据报及端口号进行简明扼要的介绍。

1.2.1 IP 地址

网络互联的目的是提供一个无缝的通信系统，为此，互联网协议必须屏蔽物理网络的具体细节，并提供一个虚拟网络的功能，使设计者可以在不考虑物理硬件细节的情况下自由地选择地址。在 TCP/IP 协议栈中，编址由 IP 协议规定，IP 标准分配给每台主机一个 32 位的二进制数作为该主机的 IP 地址。在最新出台的 IPv6 中 IP 地址升至 128 位，这样 IP 资源就变得更加丰富。目前支持 IPv6 协议的软件已经有很多，但是离实用化还有一段距离，有兴趣的读者可以参考其他资料。

每个 32 位 IP 地址被分割成两部分：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。

IP 地址共分 5 类：A 类、B 类、C 类、D 类和 E 类。其中 A 类、B 类和 C 类为基本类，D 类用于多播传送，E 类属于保留类，现在不用。它们的格式如下（其中，*代表网络号位数）：

A 类：0***** xxxxxxxx xxxxxxxx xxxxxxxx

B 类：10***** *****x xxxxxxxx xxxxxxxx

C 类：110***** ***** ***** xxxxxxxx

D 类：1110xxxx xxxxxxxx xxxxxxxx xxxxxxxx

E 类: 1111xxxx xxxxxxxx xxxxxxxx xxxxxxxx

IP 地址一般采用点分十进制的表示方法, 例如:

10000001 00110100 00000110 00000000 → 129.52.6.0

采用点分十进制表示方法后的地址分类如表 1-2 所示:

表 1-2 各类 IP 地址的范围

类型	范围
A 类	0.0.0.0 - 127.255.255.255
B 类	128.0.0.0 - 191.255.255.255
C 类	192.0.0.0 - 223.255.255.255
D 类	224.0.0.0 - 239.255.255.255
E 类	240.0.0.0 - 247.255.255.255

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号 (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号

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比特用于主机号。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。

1.2.2 地址解析

地址解析 (Address Resolution) 就是将计算机中的协议地址翻译成物理地址 (或称 MAC 地址, 即媒体映射地址), 地址解析只能在本地网内进行。

地址解析技术可分为如下 3 种：

■ 表查询 (Table-lookup)

该方法适用于 WAN (广域网)，通过建立数组 (协议地址，物理地址) 的方法解决。用户可通过查询找到物理地址。

■ 相近形式计算 (Closeform-Computation)

该方法适用于可以自行配置的网络，IP 地址和物理地址相互对应，例如：

220.123.5.1 → XXX1

220.123.5.2 → XXX2

可通过算法得到物理地址：物理地址 = 协议地址 & 0XFF

■ 信息交换 (Message-Exchange)

该方式适用于 LAN，是基于分布式的处理方式，即主机发送一个解析请求，以广播的形式发出，并等待网络内各个主机的响应。

ARP 中规定了两种信息的基本类型：请求 (Request) 和应答 (Response)，ARP 的格式如图 1-4 所示。

硬件地址类型		协议地址类型
硬件地址长度	协议地址长度	操作类型
发送方硬件地址 (前 32 位)		
发送方硬件地址 (后 16 位)		发送方协议地址 (前 16 位)
发送方协议地址 (后 32 位)		接收方硬件地址 (前 16 位)
接收方硬件地址 (后 32 位)		
接收方协议地址		

图 1-4 ARP 的格式

ARP 信息的发送过程 (信息包含在数据帧中) 如下：

- (1) 主机向网络发出广播；
- (2) 如信息涉及本机则响应、否则丢弃；
- (3) 将响应信息直接发给请求方。

ARP 信息处理的过程如下：

- (1) 发送：将信息封装成数据帧，然后向网络发送广播；
- (2) 确认：将帧头的某标记位标记为 ARP 信息，而不是一般信息。对以太网为 0X806；
- (3) ARP 响应缓冲：在内存中建立 ARP 缓冲区，存储 IP 地址和硬件地址的绑定；
- (4) 处理发送来的 ARP 信息：
 - ① 更新缓冲区的信息；
 - ② 判断信息类型，是请求还是响应；
 - ③ 如果是请求，则判断 ARP 信息中的接收方协议地址是否与本机协议地址一致，如果一致，则响应该请求，否则丢弃该信息包；
 - ④ 如果是响应，则开始通信。

此外，用户可以用 ARP 命令及参数 -a 来显示 ARP 高速缓存中的所有内容。超级用户可以用选项 -d 来删除 ARP 高速缓存中的某一项内容，还可以使用 -s 参数来添加一条新的高速缓存。