

高等学校计算机网络工程专业规划教材

Windows 网络程序设计

夏靖波 杜华桦 王晓东 段弢 编著

西安电子科技大学出版社
[http:// www.xduph. com](http://www.xduph.com)

高等学校计算机网络工程专业规划教材

Windows 网络程序设计

夏靖波 杜华桦
王晓东 段 弢 编著

西安电子科技大学出版社

2006

内 容 简 介

随着网络的日趋复杂和规模的日趋庞大,网络编程技术已成为计算机网络技术的重要分支之一。本书简洁而系统地介绍了网络编程的基本概念、基本知识和编程技术,主要内容包括TCP/IP 网络基础知识、基于 NetBIOS 的网络编程、WinSock 基础、基于 WinSock 的一些网络应用、直接网络编程和高级网络编程等。本书采用理论与实际相结合的方法进行编写,在介绍各种理论知识的基础上及时引入相对应的应用程序,以加深读者对理论知识的理解。各章均有适量习题,供读者练习和思考。另外,本书给出的附录包括 NetBIOS 命令和命令返回值、WinSock 1.1 和 WinSock 2 函数、WinSock 错误代码表和 Tracert 程序示例等,方便读者进行查询和进一步研究。

本书可作为高校计算机网络工程专业及相关专业研究生或本科生的教材,也可作为计算机网络和数据通信领域工程技术人员的参考书。

★本书配有电子教案,需要者可与出版社联系,免费提供。

图书在版编目(CIP)数据

Windows 网络程序设计 / 夏靖波等编著. —西安:西安电子科技大学出版社, 2006.2

高等学校计算机网络工程专业规划教材

ISBN 7-5606-1621-6

I. W… II. 夏… III. 窗口软件, Windows—程序设计—高等学校—教材 IV. TP316.7

中国版本图书馆 CIP 数据核字(2005)第 155186 号

策 划 云立实 臧延新

责任编辑 雷鸿俊 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http://www.xduph.com E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2006 年 2 月第 1 版 2006 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 23

字 数 541 千字

印 数 1~4000 册

定 价 26.00 元

ISBN 7-5606-1621-6/TP·0929

XDUP 1913001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

近年来，由于网络日趋复杂，规模日趋庞大，网络编程技术的需求日趋强烈，本书正是顺应这种需求而编写的。本书内容翔实、实例丰富，并遵循理论与实践相结合的原则，在系统介绍理论的前提下，结合实际工作经验，给出了适量的编程实例，以飨读者。

本书第1章是网络基础，主要包括TCP/IP的基础知识和网络编程应考虑的问题，介绍网络基础知识，指出网络编程与单机程序设计的不同之处，给读者建立起网络编程的宏观概念。第2章是基于NetBIOS的网络编程。网络基本输入/输出系统(NetBIOS)是一个应用程序接口(API)，用于数据源与目的地之间的数据交换，为程序提供请求低级网络服务的统一命令集，从而给局域网(LAN)提供网络以及其他特殊功能。在介绍NetBIOS的基本概念和编程基础上，设计了基于NetBIOS的数据报通信程序和会话通信程序。第3章是WinSock基础，主要介绍WinSock的基本概念、编程原理、I/O模型，WinSock 2的扩展特性，套接字选项和I/O控制命令。第4章是基本网络应用，主要介绍使用较广泛的各种网络应用，如获取主机的名字、域名和IP地址，获取子网掩码、网卡的MAC地址，获取网络协议信息，模拟FTP功能，TCP实现客户机/服务器聊天功能和UDP实现点到点会话功能等。第5章是直接网络编程，介绍几种不同的直接网络编程方法，以帮助读者了解网络低层协议的运行状况，并可以通过直接网络编程方法来构造自己的网络高级应用程序。第6章是高级网络编程，重点介绍基于微软基本类库(Microsoft Foundation Class Library, MFC)的WinSock网络编程和多线程技术，这些技术能够为程序员开发功能更为强大的网络程序提供有力支持。

本书给出的附录包括NetBIOS命令和命令返回值、WinSock 1.1和WinSock 2函数、WinSock错误代码表和Tracert程序示例等，以方便读者进行查询和进一步研究。

由于作者的水平有限，书中疏漏之处在所难免，欢迎读者批评指正，并提出宝贵意见和建议，以便不断改进。

编 者

2005年11月

目 录

第 1 章 网络基础	1
1.1 TCP/IP 简介	1
1.1.1 OSI 模型与 TCP/IP 结构.....	1
1.1.2 TCP/IP 基本概念	3
1.1.3 常用协议.....	7
1.1.4 进程/应用层协议	19
1.2 网络编程应考虑的问题.....	20
1.2.1 并发环境下的网络编程.....	20
1.2.2 异构环境下的网络编程.....	20
1.2.3 阻塞与非阻塞通信.....	21
1.2.4 服务类型的选择	22
1.2.5 差错处理.....	23
习题与思考题.....	24
第 2 章 基于 NetBIOS 的网络编程	25
2.1 NetBIOS 概述.....	25
2.2 NetBIOS 应用服务	27
2.2.1 NetBIOS 名字	27
2.2.2 NetBIOS 数据报	29
2.2.3 NetBIOS 会话.....	29
2.2.4 NetBIOS 一般命令.....	30
2.3 Ncb/Mcb.....	31
2.3.1 Ncb/Mcb 域.....	31
2.3.2 NetBIOS 命令调用.....	36
2.3.3 NetBIOS 命令的完成.....	37
2.4 NetBIOS 编程基础	38
2.5 数据报通信程序设计	45
2.5.1 数据报通信模型	45
2.5.2 广播式数据报程序.....	45
2.5.3 定向型数据报程序.....	51
2.6 会话通信程序设计	53
2.6.1 会话通信模型.....	53

2.6.2	服务器端程序.....	54
2.6.3	客户端程序.....	60
	习题与思考题.....	64
第 3 章	WinSock 基础	65
3.1	基本概念.....	65
3.1.1	套接字及类型.....	65
3.1.2	网间进程通信.....	66
3.1.3	服务方式.....	67
3.1.4	客户机/服务器模式.....	68
3.1.5	WinSock 对 Socket 的扩充.....	69
3.2	WinSock 编程原理.....	70
3.2.1	WinSock 的启动和终止.....	70
3.2.2	错误检查和控制.....	70
3.2.3	WinSock 编程模型.....	71
3.3	WinSock I/O 模型.....	73
3.3.1	Select 模型.....	73
3.3.2	WSAAsyncSelect 模型.....	74
3.3.3	WSAEventSelect 模型.....	75
3.4	WinSock 2 的扩展特性.....	77
3.4.1	原始套接字.....	77
3.4.2	重叠 I/O 模型.....	78
3.4.3	服务质量(QoS).....	78
3.5	套接字选项和 I/O 控制命令.....	80
3.5.1	套接字选项.....	80
3.5.2	I/O 控制命令.....	82
	习题与思考题.....	83
第 4 章	基本网络应用	84
4.1	获取计算机 IP 地址和主机名.....	84
4.1.1	实现原理.....	84
4.1.2	程序实现.....	85
4.2	获取网卡信息和子网掩码.....	87
4.2.1	实现原理.....	87
4.2.2	程序实现.....	89
4.3	获取计算机安装的协议.....	91
4.3.1	实现原理.....	91
4.3.2	程序实现.....	93
4.4	流套接字编程设计.....	95

4.4.1 实例一：模拟 FTP 功能	96
4.4.2 实例二：TCP 实现客户机/服务器聊天	103
4.5 数据报套接字编程	111
习题与思考题	116
第 5 章 直接网络编程	117
5.1 原始套接字编程	117
5.1.1 概念	117
5.1.2 ICMP 实现	118
5.1.3 Tracert	145
5.1.4 IP_HDRINCL 的使用	145
5.2 基于 Winpcap 的网络数据包捕获技术	146
5.2.1 Winpcap 简介	146
5.2.2 数据包捕获驱动器结构	148
5.2.3 数据包捕获驱动程序 API 的使用	150
5.2.4 数据包捕获函数库的使用	169
5.3 基于 Libnet 的网络数据包构造技术	201
5.3.1 Libnet 简介	202
5.3.2 Libnet 的使用方法	203
5.3.3 Libnet 函数	205
5.3.4 应用程序示例	217
习题与思考题	231
第 6 章 高级网络编程	232
6.1 MFC 概述	232
6.1.1 封装	232
6.1.2 继承	233
6.1.3 虚拟函数和动态约束	233
6.1.4 MFC 的宏观框架体系	234
6.2 基于 MFC Socket 类的网络编程	234
6.2.1 CAsyncSocket 类	234
6.2.2 CSocket 类	243
6.2.3 基于 MFC 的 Socket 类的网络编程方法	253
6.3 多线程 WinSock 网络编程	256
6.3.1 多线程概论	256
6.3.2 基本线程操作函数	258
6.3.3 线程同步	260
6.3.4 多线程网络程序设计	268
习题与思考题	280

附录 1 NetBIOS 命令	281
附录 2 NetBIOS 命令返回值	294
附录 3 WinSock1.1 函数	297
附录 4 WinSock 2 函数	312
附录 5 WinSock 错误代码表	323
附录 6 Tracert 程序示例	327
参考文献	355

第1章 网络基础

计算机网络程序设计就是利用网络应用编程接口编写网络应用程序，实现网络应用进程间的信息交互功能。本书主要讨论的是基于 TCP/IP 协议栈的网络编程，即网络应用程序访问 IPv4 协议提供的服务，以实现不同系统上进程间的通信。

网络通信基于网络协议，网络编程接口访问网络协议提供的服务。不同的网络协议提供不同的服务访问接口，同一网络应用编程接口可提供访问不同网络协议的接口。

随着计算机网络的飞速发展和日益普及，网络应用越来越多。业界对计算机网络程序设计的需求也相应增多。因此，计算机网络程序设计作为一种重要的知识技能开始受到人们的重视。要学好网络编程，对于操作系统、网络协议、网络编程模式和方法应有较好的理解。

1.1 TCP/IP 简介

TCP/IP(Transmission Control Protocol/Internet Protocol, 传输控制协议/网际协议)是一系列协议，或者说是一个协议族，它定义了数据传输如何通过因特网进行交换。TCP/IP 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，是一个真正的开放系统。TCP/IP 是根据两个最常用的协议命名的，已经实际应用了许多年，并已在世界范围内证明了它的有效性。随着 PC 的普及，TCP/IP 以其开放性的特点，成为了 Internet(因特网)的基础。

1.1.1 OSI 模型与 TCP/IP 结构

OSI/RM(Open System Interconnection/Reference Model, 开放系统互连参考模型)将计算机网络通信定义为一个七层框架模型，如图 1.1 所示。

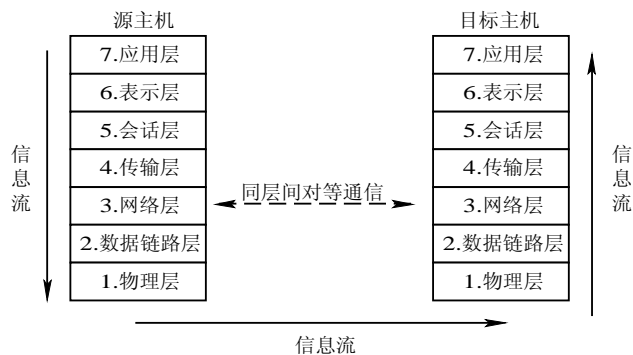


图 1.1 OSI 模型与通信流程

OSI 模型把计算机网络通信的组织与实现按功能划分为七个层次，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。每一层协议建立在它的下层协议的基础上，并为其上层提供服务。对于它的上层协议来说，该层是透明的。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 所示。

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	提供传输顺序与相应信息
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

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，主要通过软件来进行。当数据在层间向下传播时(源主机部分)，每一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。到了目标主机时，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。

TCP/IP 是一个四层的协议模型，它的结构及与 OSI 模型之间的关系如图 1.2 所示。

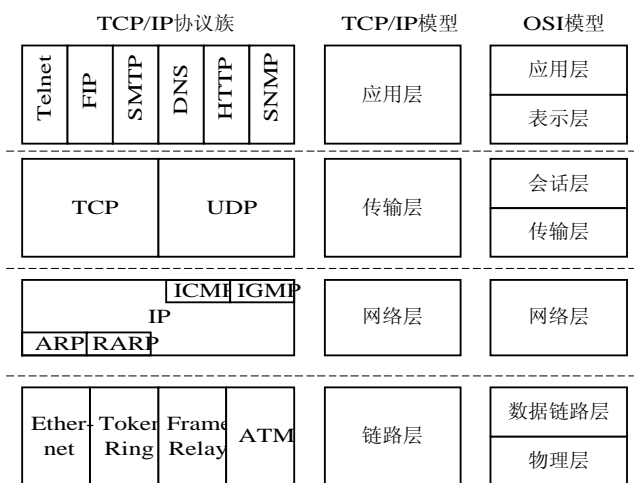


图 1.2 TCP/IP 族的体系结构

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。
- 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(网际协议)、ICMP(Internet 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 则只负责简单地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 Telnet、HTTP、SMTP、FTP、DNS 和 SNMP 等协议和应用。

图 1.3 是 TCP/IP 的四层结构的一个具体示例。

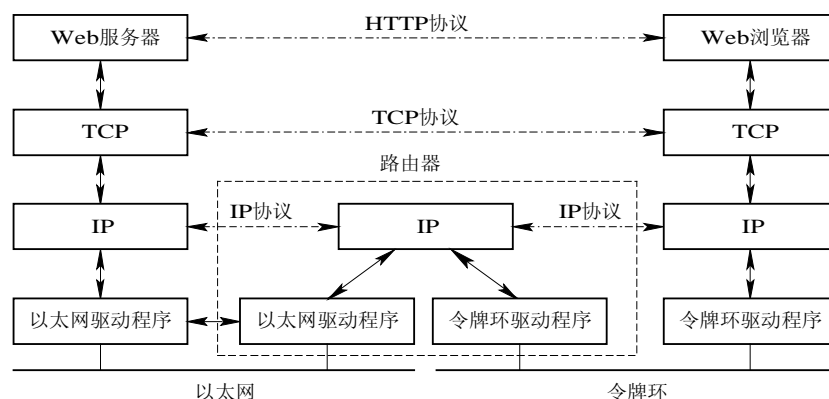


图 1.3 通过 TCP/IP 和路由器连接的两台主机

在图 1.3 所示的系统中，两台主机通过路由器互相连接。路由器可以把以太网、令牌点对点链接和 FDDI(光纤分布式数据接口)等不同的网络连接在一起。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，不必了解数据如何在网络中传输，也不必了解怎样接入网络。而链路层只需要负责传输，而不需要知道正在传送的是什么数据。

1.1.2 TCP/IP 基本概念

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，本节主要简要介绍 IP 地址、地址解析及端口号等基本概念。

1. IP 地址与子网掩码

网络互联的目的是提供一个无缝的通信系统。为此，必须用互联网协议屏蔽物理网络的具体细节，并提供一个虚拟网络的功能。在 TCP/IP 栈中，编址由 IP 协议规定，IP 标准

(140.252.0.0)，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 1.4 所示。这样就允许有 254 个子网，每个子网可以有 254 台主机。

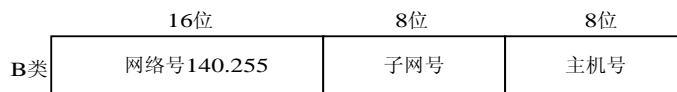


图 1.4 B 类地址的子网编址举例

除了地址以外，主机还需要知道有多少位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 的位留给网络号和子网号，为 0 的位留给主机号。在上面的例子中，子网掩码就是 255.255.255.0。

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三种：

(1) 表查询(Table-Lookup)。该方法适用于广域网(WAN)，通过建立映射数组(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。

(2) 相近形式计算(Close Form-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

可通过这种算法得到物理地址：物理地址 = 协议地址 & 0xFF。

(3) 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。

3. 域名系统

一个系统的全域名由主机名、域名和扩展名三部分组成，各部分间使用“.”分隔，例如 www.sina.com。在 TCP/IP 应用中，域名系统(DNS)是一个分布的数据库，由它来提供 IP 地址和主机名之间的映射信息，可以通过在程序中调用标准库函数来编程实现域名与 IP 地址之间的相互转换。通过从域名地址到 IP 地址的映射，使得在日常的网络应用中可以使用域名这种便于记忆的网络地址表示形式。

所有的网络应用程序理论上都应该具有内嵌的域名解析机制。

4. 数据包的封装和分用

当应用程序用 TCP 传送数据时，数据被送入协议栈中，然后逐个通过每一层，直到被当作一串比特流送入网络。其中每一层对收到的数据都要增加一些首部信息(有时还要增加尾部信息)，该过程如图 1.5 所示。TCP 传给 IP 的数据单元称作 TCP 报文段或简称 TCP 段。IP 传给网络接口层的数据单元称作 IP 数据报(IP Datagram)。通过以太网传输的比特流称作帧(Frame)。图 1.5 中帧头和帧尾下面所标注的数字是典型以太网帧首部的字节长度。以太网数据帧的长度必须在 46~1518 字节之间。

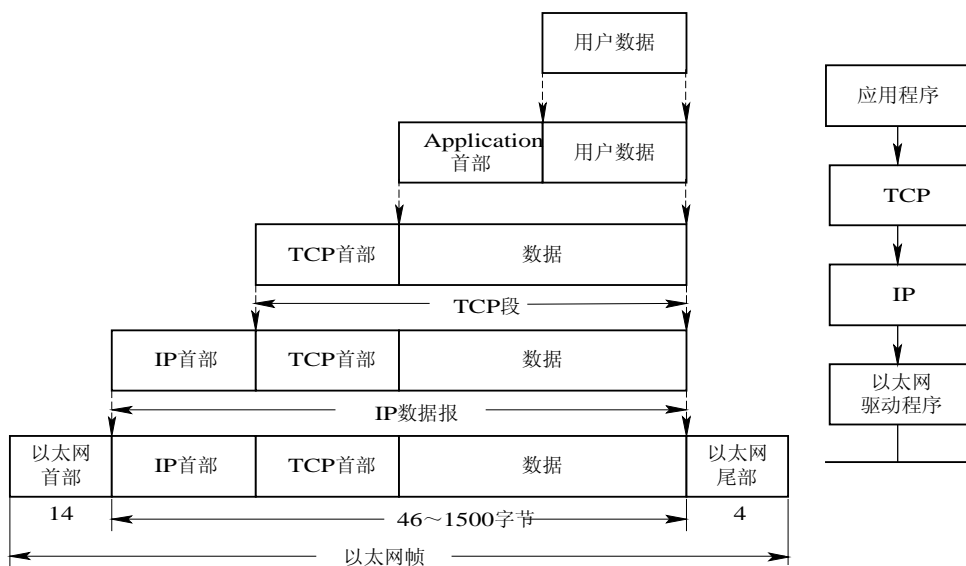


图 1.5 数据包的封装过程示意图

由于 IP 数据报通常是分组传输的，因此更准确地说，图 1.5 中 IP 和网络接口层之间传送的数据单元应该是分组(Packet)。分组既可以是一个完整的 IP 数据报，也可以是 IP 数据报的一个片(Fragment)。关于分组的细节算法不在本书的讨论范围之内。

UDP 数据与 TCP 数据基本一致。唯一不同的是，UDP 传给 IP 的信息单元称作 UDP 数据报(UDP Datagram)，而且 UDP 的首部长为 8 字节。由于 TCP、UDP、ICMP 和 IGMP 都要向 IP 传送数据，因此 IP 必须在生成的 IP 首部中加入某种标识，以表明数据属于哪一层。为此，IP 在首部中存入一个长度为 8 bit 的数据，称作协议域。1 表示 ICMP，2 表示 IGMP，6 表示 TCP，17 表示 UDP。

同样，许多应用程序都可以使用 TCP 或 UDP 来传送数据。传输层协议在生成报文首部时要存入一个应用程序的标识符。TCP 和 UDP 都用一个 16 位的端口号来表示不同的应用程序。TCP 和 UDP 把源端口号和目的端口号分别存入报文首部中。

网络接口分别要发送和接收 IP、ARP 和 RARP 数据，因此也必须在以太网的帧首部中加入某种形式的标识，以指明生成数据的网络层协议。为此，以太网的帧首部也有一个 16 bit 的帧类型域。

当目的主机收到一个以太网数据帧时，数据就开始从协议栈中由底向上升，同时去掉各层协议加上的报文首部。每层协议盒都要去检查报文首部中的协议标识，以确定接收数据的上层协议。这个过程称作分用(Demultiplexing)，图 1.6 显示了该过程。

图 1.6 中把 ARP 和 RARP 放在 IP 层是因为它们都有以太网头部和尾部，而在前面的论述中把它们作为链路层的协议，这并不矛盾，因为 TCP/IP 栈的层次划分本来就不是非常清晰。

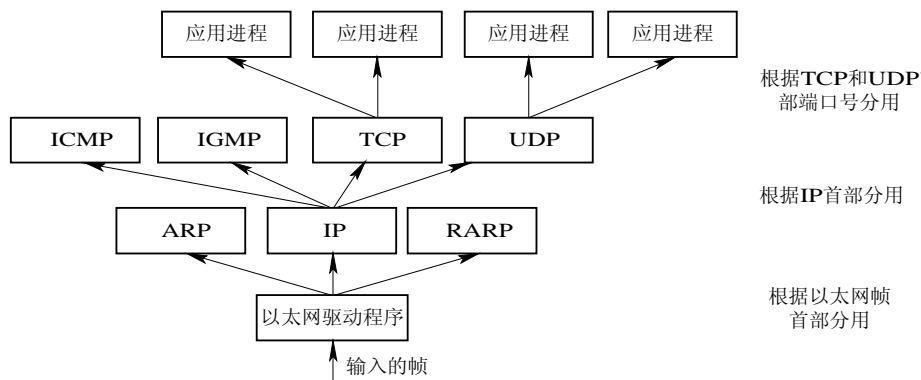


图 1.6 数据分用过程

5. 端口号

TCP 和 UDP 采用端口号来识别应用程序。例如，服务器提供的服务一般都是通过通用端口号来识别的，对于 TCP/IP 实现来说，FTP 服务器的 TCP 端口号都是 21，Telnet 服务器的 TCP 端口号都是 23，TFTP(简单文件传送协议)服务器的 UDP 端口号都是 69。任何 TCP/IP 实现所提供的服务都使用通用端口号 1~1023。这些通用端口号由 Internet 号分配机构(Internet Assigned Numbers Authority, IANA)来管理。

客户端通常对它所使用的端口号并不关心，只需保证该端口号在本机上是唯一的就可以了。客户端端口号又称作临时端口号(即存在时间很短暂)，这是因为它通常只是在用户运行该客户程序时才存在，而服务器则只要主机是开着的，其服务就运行。

大多数 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(Internet 上并不常用的服务)预留的。

1.1.3 常用协议

下面对一些常用协议进行简要的介绍。

1. 以太网数据链路层帧结构

IEEE 802.3 定义了一种具有七个字段的帧(MAC)：前导符、起始帧分界符、目标地址、源地址、PDU 的长度/类型、数据以及 CRC。以太网不提供任何对收到的帧进行确认的机制，确认在高层完成，这表明它是一种不可靠的介质。CSMA/CD 中 MAC 帧的格式如图 1.7 所示。

前导符	起始帧分界符	目标地址	源地址	长度/类型	数据	CRC
7字节	1字节	6字节	6字节	2字节	46~1500字节	4字节

图 1.7 IEEE 802.3 MAC 帧结构

- 前导符(Preamble)：该字段长 7 个字节(56 位)，其中 1 和 0 交替出现，警告接收系统即将有数据帧到来，同时同步系统时序。
- 起始帧分界符(SFD)：该字段长 1 字节，为 10101011，标志帧的开始。SFD 通知接

收方后面所有的内容都是数据。

- **目标地址(Destination Address):** 该字段长 6 个字节, 包含了数据帧的目的物理地址。一个系统的物理地址是一个在它的网络接口卡(NIC)上编码的比特模式。每一个 NIC 由一个独一无二的地址将它和其他所有的 NIC 区别开来。
- **源地址(Source Address):** 该字段同样长 6 个字节, 包含转发数据帧的最后一个设备的物理地址。该设备可以是发送站点, 也可以是接收和转发数据包的最近路由器。
- **PDU 的长度/类型(Length/Type):** 该字段的两个字节指出 PDU 的长度或封装的数据类型。当 PDU 的长度固定时, 这个字段可以用来表示数据类型, 如 IP(0x0800)、ARP(0x0806)、RARP(0x8035)等。在以太网中, 如果高层协议采用 TCP/IP 协议族, 则 MAC 帧的结构如图 1.8 所示(没有标出前导符和起始帧分界符)。



图 1.8 采用 TCP/IP 协议族的 MAC 帧结构示意图

- **数据:** 保存高层协议的数据(PDU)。
- **CRC:** IEEE 802.3MAC 帧的最后一个字段是检错信息, 通常为 CRC-32。

2. IP

IP 负责在 TCP/IP 主机之间提供数据报服务, 进行数据封装及产生协议头。由于在以太网中帧的大小要受到限制, 并且不同的帧可能由不同的网络路径传送, 因此 IP 协议需要将较大的数据报文分割开来, 并在目的主机处按正确顺序组合。另外, IP 协议不负责包的校验, 它是一种无连接、不可靠的传输。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。如果发生任何错误, IP 协议则丢弃该数据报, 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 等提供的。无连接(connectionless)的意思是 IP 并不维护任何关于后续数据报的状态, 每个数据报的处理是相互独立的, 即 IP 数据报可以不按发送顺序接收。

IP 协议还要负责寻找路由, 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 IP 地址。一般来说不会有应用程序直接访问 IP 协议。

IP 数据报是 Internet 上数据通信的基本单元, 这些数据报不超过 1000 字节长, 当人们打开 Web 页、下载文件或者发送 E-mail 时, 这些数据报就在世界各地来回传输。IP 数据报的报文格式如图 1.9 所示。

0	4	8	16	19	24	31	
版本		报头长度		服务类型		报文总长度	
标识符				标志	分段偏移量		
生命周期		协议		头校验和			
源IP地址							
目的IP地址							
选项					填充区		
数据							

图 1.9 IP 数据报格式

IP 数据报头的最小长度是 5 个字(word, 1 字=4 字节), 如果有其他选项, 报头可能会更长。IPv4 数据报中的数据(包括报头中的数据)以 32 位(4 字节)的方式来组织。IPv4 中包含至少 12 个不同字段, 且在没有任何选项时长度为 20 个字节, 但在包含选项时可达 60 个字节。

- 版本(VERS): 指定 IP 协议的版本号, 对于 IPv4 来说, 版本为 4。
- 报头长度(HLENS): 指定 IP 报头的长度, 以字为单位, 范围为 5~15 个字。
- 服务类型(ToS, Type of Service): 表示数据报的服务类型, 即处理的优先级, 包括延时、吞吐量、可靠性或代价, 它在 IPv4 中的应用并不广泛。
- 报文总长度(Total Length): 以字节为单位指定数据报的总长度, IP 数据报的长度最大为 65 535 字节, 网络主机可以使用数据报长度来确定一个数据报的结束和下一个数据报的开始; 当传送长度超过 65 535 字节的 IP 数据报时, 大多数的链路层都会分片。主机一般要求接收的数据报不超过 576 字节。由于 TCP 把用户数据分成若干片, 因此, 一般来说这个限制不会影响 TCP。
- 标识符(ID): 该 16 位标识符由产生它的主机唯一指定给数据报, 分段后的数据报共享同一个数据报 ID, 有助于接收主机对分段的数据报重装。
- 标志(FLG): 包括 3 个 1 位标志, 标识报文是否允许被分段和是否使用了这些域。第一位保留并设为 0; 第二位标识报文能否被分段, 其中 0 表示报文可以被分段, 1 表示报文不能被分段; 第三位只有在第二位为 0 时才有意义, 这一位标识此报文是否是这一系列分段的最后一个, 或者接收应用程序是否还希望有更多的段, 0 指示报文是最后一个。
- 分段偏移量(Fragment Offset): 指定分段在整个数据报中的位置。接收主机同时使用标志位和分段偏移, 以重组被分段的数据报。这个值以 64 位为单位递增。
- 生命周期(TTL, Time To Live): 代表数据报在被丢弃前能够穿越的最大主机跳数。TTL 的初始值由源主机设置, 其理论最大值为 255, 每经过一个处理节点减 1。当该字段的值为 0 时, 报文就被认为是不可转发的, 之后产生一个 ICMP 报文发回源主机, 并丢弃该不可转发的报文。
- 协议(Protocol): 指明数据报中携带的载荷类型, 主要标识所使用的协议, 一般是指 TCP 协议、UDP 协议、ICMP 报文和 IGMP 报文。
- 头校验和(Header Checksum): 目的是保证报头的正确性, 目的机、网络中的每个网