

网络编程系列丛书

梅晓冬 颜焱青 编著

Visual C#

网络编程 技术与实践



书附光盘中给出案例源代码、案例开发和运行过程的全程多媒体讲解视频

本书主要内容

- ◆ Visual C# 网络编程技术概述
- ◆ 开发基于TCP/UDP协议的应用程序
- ◆ 开发基于SNMP协议的应用程序
- ◆ 使用.NET发送邮件
- ◆ FTP下载与文件传输
- ◆ 基于C#.NET的网络管理模块
- ◆ 编写基于.NET的Web Service
- ◆ 基于.NET的远程技术
- ◆ 流媒体在线播放
- ◆ 在线五子棋系统模块
- ◆ 基于HTTP协议的下载工具

清华大学出版社



TP312/2939D

2008

网络编程系列丛书

Visual C#网络编程 技术与实践

梅晓冬 颜烨青 编著

清华大学出版社

北 京

内 容 简 介

本书详细介绍了利用 Visual C# 2005 进行网络编程的方法和技巧。全书共分 13 章, 主要内容包括网络编程原理、Visual C# 2005 的套接字以及多线程的开发、基于各种不同协议的网络编程应用模块, 并通过几个典型的实例介绍了 Visual C# 2005 网络编程的实际应用。

本书注重代码的通用性和工程实践性, 书中提供的通用模块和典型实例稍加修改就可以为读者所用。

本书不仅适用于使用 Visual C# 2005 进行软件开发的广大软件开发人员, 也适合高等院校师生学习和参考使用, 特别对高校计算机专业的学生进行毕业设计具有非常好的指导价值, 也可以作为广大计算机编程爱好者的自学参考书。

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

Visual C#网络编程技术与实践/梅晓冬, 颜烨青 编著. —北京: 清华大学出版社, 2008. 6

(网络编程系列丛书)

ISBN 978-7-302-17520-9

I. V… II. ①梅…②颜… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 063058 号

责任编辑: 王 定

封面设计: 久久度文化

版式设计: 康 博

责任校对: 胡雁翎

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 31.25 字 数: 722 千字

附光盘 1 张

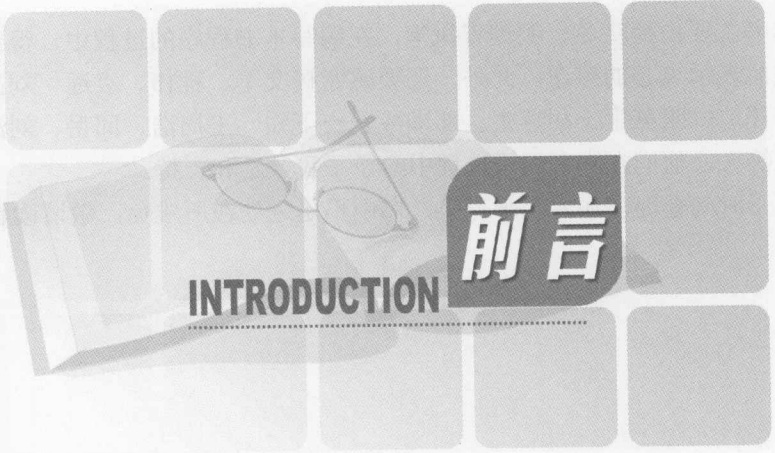
版 次: 2008 年 6 月第 1 版

印 次: 2008 年 6 月第 1 次印刷

印 数: 1~5000

定 价: 56.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 023329-01



INTRODUCTION

前言

Microsoft Visual C# 2005 是一种新的编程环境，它是为生成在 .NET Framework 上运行的多种应用程序而设计的。C# 简单，功能强大，类型安全，而且是面向对象的。C# 凭借它的许多创新实现了对应用程序的快速开发。

Visual Studio 支持 Visual C#，这是通过功能齐全的代码编辑器、项目模板、设计器、代码向导、功能强大且易于使用的调试器以及其他工具实现的。通过 .NET Framework 类库，可以访问多种操作系统服务和其他有用的精心设计的类，这些类可显著加快开发周期。

本书大量地使用了 .NET Framework 提供的类库中的函数来协同开发，通过使用这些已经被封装的类来协助开发确实提高了不少效率，为开发工作带来了诸多的便利。

本书共分 13 章，第 1 章介绍了利用 Visual C# 2005 提供的网络命名空间下的诸多 API 进行套接字的开发，并对网络编程的原理进行了详细的阐述，为后续的章节奠定了基础。

第 2 章介绍了具有多线程能力的网络应用程序，这使得用户开发的程序能够适应更加复杂的情况。

第 3 章~第 8 章是本书的重要部分，主要介绍基于各种不同协议的网络编程应用模块。第 3 章介绍了基于 TCP 协议的聊天程序；第 4 章介绍了基于 UDP 的通信程序；第 5 章介绍了使用 SNMP 协议的网络管理程序；第 6 章介绍了使用 SMTP 以及 POP3 协议的 E-mail 系统；第 7 章介绍了基于 FTP 的文件传输模块；第 8 章介绍了使用 ICMP 协议的网络探测程序。

第 9 章介绍了基于 Web Service 的电子公告板系统，并结合采用了 SQL Server 2005 数据库；第 10 章介绍了使用远程控制技术的应用程序；第 11 章则介绍了流媒体在线播放系统。

第 12 章介绍了基于 TCP 通信协议的在线五子棋系统；第 13 章主要介绍了基于 HTTP 协议的下载工具，该工具具有多线程下载、断点续传的功能。

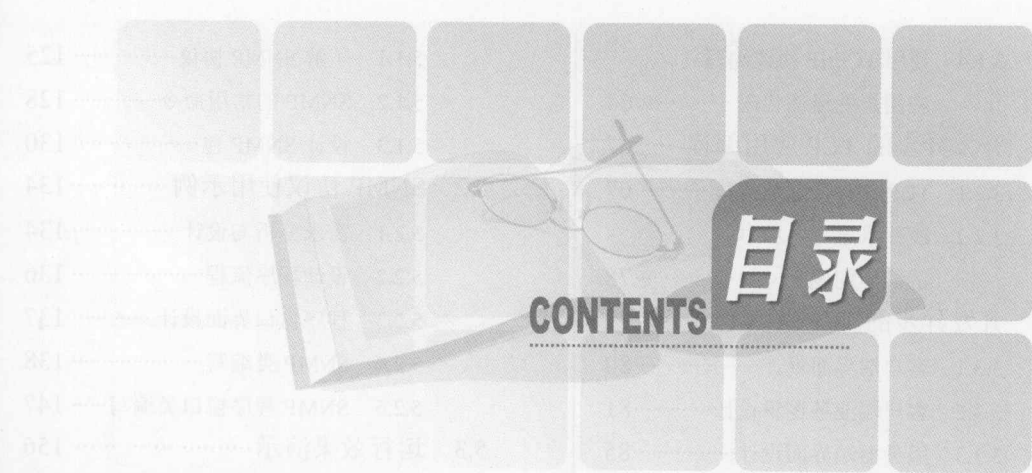
本书注重理论指导性和工程实践性，书中提供的各个网络应用程序只要稍加修改就可以为读者所用。本书不仅适用于使用 Visual C# 2005 进行软件开发的广大软件开发人员，也适合高等院校师生学习和参考使用，特别对高校计算机专业的学生进行毕业设计具有非常好的指导价值，也可以作为广大计算机编程爱好者的自学、参考用书。

本书主要由梅晓冬、颜焯青执笔，在编写本书程序的过程中，得到了陈璧元的大力支持，在此表示衷心的感谢。此外，还要感谢杨文军、程伟、袁远、刘武、彭澜、李通、李杰、卢茂琼、张燕生、胡燕生、邓湘成、卢下知、王周浩、邱岳、刘流、代本、刘明星、孙靖华等人，他们在本书编写过程中给予了我鼓励和支持。

由于时间仓促，加之水平有限，书中不足之处在所难免，敬请读者批评指正。

编者

2008年1月



目录

CONTENTS

第 1 章 C#.NET 网络编程概述	1
1.1 网络通信概述	1
1.1.1 网络通信模型概述	1
1.1.2 网络通信协议、接口和 服务概述	2
1.1.3 TCP/IP 网络架构概述	4
1.1.4 IP 地址与端口	6
1.2 C#.NET 网络编程相关类	9
1.2.1 IPAddress 类	9
1.2.2 DNS 类	13
1.2.3 IPHostEntry 类	16
1.2.4 IPEndPoint 类	19
1.2.5 Socket 类	21
1.3 套接字概述	21
1.3.1 套接字的类型和常用属性	22
1.3.2 建立面向连接的套接字	24
1.3.3 建立面向无连接的套接字	25
1.4 使用套接字的简单示例	26
1.4.1 C/S 与 B/S 架构 通信模式概述	27
1.4.2 编写客户端代码	28
1.4.3 编写服务器端代码	30
1.4.4 无阻塞套接字	32
1.5 本章小结	33

第 2 章 C#.NET 高级网络 编程技术概述	35
2.1 线程与网络通信	35
2.1.1 基于线程的网络通信概述	36
2.1.2 在网络编程中使用多线程	36
2.1.3 线程基础	36
2.1.4 多线程在网络编程 中的应用	40
2.2 网络通信的常见问题	47
2.2.1 让网络通信代码更强壮	47
2.2.2 数据缓冲区处理方法	49
2.3 TCP 无保护消息边界	51
2.3.1 发送固定长度的消息	51
2.3.2 采用变长的消息	53
2.3.3 使用特殊标记处理消息	55
2.4 本章小结	55
第 3 章 开发基于 TCP 协议 的应用程序	57
3.1 套接字与 TCP 协议	57
3.1.1 使用套接字传输数据	57
3.1.2 NetworkStream 对象 同数据发送与接收	59
3.1.3 TcpClient 与 TcpListener 类	62

3.1.4	使用 TCP/IP 协议编写 应用层的通信代码	64
3.2	开发异步的 TCP 应用编程	67
3.2.1	TCP 的异步通信流程	67
3.2.2	线程阻塞与异步中 的同步问题	78
3.3	开发异步的 TCP 聊天程序	79
3.3.1	客户端界面设计	80
3.3.2	客户端业务逻辑设计	81
3.3.3	服务器端界面设计	85
3.3.4	服务器端业务逻辑设计	86
3.3.5	运行界面	93
3.4	本章小结	96
第 4 章	开发基于 UDP 的应用程序	97
4.1	UDP 通信协议概述	97
4.1.1	UDP 协议与 TCP 协议的差别	97
4.1.2	UDP 协议的使用场合	99
4.2	在 C# 下 UDP 协议的相关类	100
4.2.1	IPAddress 类	100
4.2.2	UdpClient 类的构造函数	101
4.2.3	UdpClient 类的常用 方法与实例	102
4.3	UDP 协议使用示例	105
4.3.1	UDP 模块功能概述	106
4.3.2	设计通信流程	106
4.3.3	开发服务器端程序	107
4.3.4	开发客户端程序	110
4.3.5	使用多线程开发 UDP 协议	113
4.3.6	使用校验保证信息完整性	118
4.3.7	效果演示	121
4.4	本章小结	123
第 5 章	开发基于 SNMP 协议 的应用程序	125
5.1	SNMP 协议概述	125

5.1.1	了解 SNMP 协议	125
5.1.2	SNMP 的常用命令	128
5.1.3	设计 SNMP 包	130
5.2	SNMP 协议使用示例	134
5.2.1	需求分析与设计	134
5.2.2	设计程序流程	136
5.2.3	程序窗口界面设计	137
5.2.4	SNMP 类编写	138
5.2.5	SNMP 程序窗口类编写	147
5.3	运行效果演示	156
5.4	使用供货商提供的 MIB 库	158
5.5	本章小结	159
第 6 章	使用 .NET 发送邮件	161
6.1	邮件发送与接收协议概述	161
6.1.1	SMTP 协议与邮件发送	162
6.1.2	POP3 协议与邮件接收	163
6.1.3	.NET 下支持 SMTP 和 POP3 的类	166
6.2	邮件发送与接收模块	169
6.2.1	需求分析与设计	169
6.2.2	设计邮件发送和接收 的流程	171
6.2.3	界面设计	172
6.2.4	编写主窗口的业务逻辑	176
6.2.5	编写发送邮件的业务逻辑	179
6.2.6	编写接收邮件的业务逻辑	186
6.2.7	使用多线程发送 与接收邮件	199
6.3	运行效果演示	200
6.4	本章小结	203
第 7 章	FTP 下载与文件传输	205
7.1	FTP 协议概述	205
7.1.1	使用 FTP 协议下载 文件的流程	206
7.1.2	相关类库说明	207

7.2	FTP 客户端设计	210
7.2.1	需求分析	211
7.2.2	界面设计	211
7.2.3	业务逻辑设计	213
7.3	FTP 服务器端设计	224
7.3.1	需求分析	224
7.3.2	FTP 响应码	225
7.3.3	业务逻辑设计	233
7.4	运行界面	268
7.5	本章小结	269

第 8 章 基于 C#.NET 的

网络管理模块

8.1	ICMP 协议概述	271
8.1.1	ping 命令	271
8.1.2	tracert 命令	273
8.2	ICMP 包	275
8.3	编写网络管理模块	278
8.3.1	需求分析	278
8.3.2	界面设计	279
8.3.3	编写核心 icmp 类的 业务逻辑	280
8.3.4	编写具有 ping 功能的 业务逻辑	286
8.3.5	编写具有 tracert 功能的 业务逻辑	293
8.3.6	编写 findmask 功能的 业务逻辑	297
8.3.7	编写时间戳功能的 业务逻辑	300
8.4	运行界面	303
8.5	本章小结	305

第 9 章 编写基于 .NET 的

Web Service

9.1	Web Service 概述	307
9.1.1	Web 服务基本概念	307

9.1.2	Web 服务的优势	308
9.1.3	Web 服务的架构	309
9.2	需求分析与设计	310
9.2.1	需求分析	310
9.2.2	文件功能设计	311
9.2.3	数据库设计	311
9.3	编写 Web Service 服务系统	314
9.3.1	构建 Web Service	314
9.3.2	编写 Web 服务代码	316
9.3.3	主页面与登录相关的 Web Service 数据访问模块	316
9.3.4	发表主题相关的 Web Service 访问模块	321
9.3.5	投票相关的 Web Service 访问模块	328
9.3.6	管理相关的 Web Service 访问模块	332
9.4	主页面与登录模块	337
9.4.1	界面设计	337
9.4.2	编写业务逻辑	340
9.5	发表主题模块	346
9.5.1	界面设计	346
9.5.2	编写业务逻辑	351
9.6	投票模块	356
9.6.1	界面设计	356
9.6.2	编写业务逻辑	358
9.7	管理模块	360
9.7.1	界面设计	360
9.7.2	编写业务逻辑	364
9.8	本章小结	366

第 10 章 基于 .NET 的远程技术

10.1	远程技术概述	369
10.1.1	远程技术开发结构	369
10.1.2	远程技术使用	370
10.2	远程控制开发客户端 的设计	372

10.2.1	界面设计与控件使用	372	12.1.2	模块设计	436
10.2.2	客户端的代码设计	378	12.2	五子棋规则类设计	436
10.3	远程控制开发服务器端的设计	389	12.2.1	胜负判定类	437
10.3.1	界面设计与控件使用	390	12.2.2	行棋规则类	440
10.3.2	服务器端的代码设计	390	12.3	基于 TCP 协议的通信类	443
10.4	运行效果演示	397	12.4	网络配置模块	447
10.5	本章小结	402	12.4.1	网络配置模块的 界面设计	447
第 11 章	流媒体在线播放	403	12.4.2	网络配置模块的 代码设计	447
11.1	需求分析与设计	403	12.5	扫描主机模块	448
11.1.1	需求分析	403	12.5.1	扫描主机模块的 界面设计	449
11.1.2	模块设计	404	12.5.2	扫描主机模块的 代码设计	449
11.1.3	数据库设计	405	12.6	主界面模块	454
11.2	DirectShow 接口	406	12.6.1	主界面模块的界面设计	454
11.2.1	滤波图模型	406	12.6.2	主界面模块的代码设计	454
11.2.2	如何使用 DirectShow 接口	406	12.7	运行界面	462
11.2.3	流媒体在线播放 相关接口	407	12.8	本章小结	464
11.3	流媒体协议	408	第 13 章	基于 HTTP 协议的 下载工具	465
11.3.1	实时传输协议(RTP)	409	13.1	HTTP 协议	465
11.3.2	实时传输控制 协议(RTCP)	409	13.1.1	HTTP 下载理论	466
11.3.3	实时流传输协议(RTSP)	409	13.1.2	.NET 的几个支持 HTTP 协议的类	467
11.3.4	流媒体服务过程	410	13.2	异步机制	471
11.4	建立.NET 的窗体项目	411	13.2.1	异步机制的原理	471
11.4.1	主窗口界面设计	411	13.2.2	异步类	471
11.4.2	收藏夹数据库相关 业务逻辑设计	414	13.3	下载工具的设计	472
11.4.3	流媒体在线播放器 业务逻辑设计	419	13.3.1	需求设计	472
11.5	运行效果演示	430	13.3.2	模块设计	473
11.6	本章小结	433	13.3.3	界面设计	474
第 12 章	在线五子棋系统模块	435	13.3.4	业务逻辑设计	475
12.1	需求分析与设计	435	13.4	运行效果演示	489
12.1.1	需求分析	435	13.5	本章小结	491

C#.NET 网络编程概述

本书着重深入讲解 C#.NET 语言的各种网络编程开发技术。C#是基于 .NET 开发平台的语言。因为它是从 C 语言衍生来的，所以从类库的角度来分析，它支持的功能非常强大，此外，这种语言又像 VB 一样简单。

不仅如此，对于网络编程开发来说它很像 Java，它还有很多 Delphi 的特点。微软也声称，C#是编写 .NET Framework 应用程序最合适的语言。

同时，C#语言的网路开发功能比较强大，可以用它来构建一些复杂的基于网路的应用程序。

1.1 网络通信概述

为了网络通信的需要，人们往往要对通信模型做一定的分析。在此基础上，提出了各种参考模型的概念，一般的参考模型有 OSI 模型和 TCP/IP 模型。

其中 OSI 模型是国际标准化组织提出的网络通信模型，而事实上，目前采用的是基于 TCP/IP 模型的网络通信方式。

1.1.1 网络通信模型概述

网络通信模型设计的总体目标是向“简单明了”和“实用化”发展。为了降低网络设计的复杂性，绝大多数的网络都形成了一堆相互层叠的通信层，每一层都建立在其下面一层的基础上。

不同的网络系统设计中，其层的数目、各层的功能、含义也不尽相同。每一层的目的

就是向上一个层提供一定的需求服务，而上一层不需要了解下一层如何实现这些服务，即每一层的操作实现步骤对上一层是屏蔽的。

可以把每一层想象成一台提供服务的功能机，它能向上一层提供特定的服务。这个概念的基本思想形同计算机领域的一些术语，如信息隐藏、抽象数据类型、数据封装等。

1.1.2 网络通信协议、接口和服务概述

在网络领域，所谓协议，就是指一套大家都公认的通信规则。不同的协议是工作在不同的层上的，而层与层之间则通常用接口来描述。

服务是指具体完成的工作，这里是抽象的概括。

1. 协议

网络设计体系中的个体(每一台 host)，形象地理解为都包含一个“一堆相互叠加的层”。

当两台在同一个网络设计体系中的主机要进行相互通信时，即一台机器上的第 n 层要与另一台机器上的第 n 层交流，在计算机对话中用到的规则和约定合起来称为第 n 层的协议(protocol)。

就像人与人之间相互交流是需要遵循一定的规矩一样，计算机之间的相互通信需要共同遵守一定的规律原则，这些规则就称为网络协议。常见的协议有 TCP/IP(传输控制协议/网际协议(Transmission Control Protocol/Internet Protocol)、IPX/SPX、NetBEUI 等。在局域网中用得比较多的是 IPX/SPX。用户如果访问 Internet，则必须在网络协议中添加 TCP/IP。

本节将重点讲述 TCP/IP。

2. 接口

两台主机间通信的传输数据不是从一台计算机上直接传递到另一台计算机上的同一层中，虽然在我们平时的印象里，计算机在执行网络通信时确实是采用“直接”的方式，但事实上，这只是给我们的“虚拟”印象。

事实上，每一个网络层都将数据的控制信息传递到它的下一层，这样一直传递到最底下的一层。最底下的一层的传输数据的介质，即物理介质，如光缆，通过它我们才可以进行实际的通信。

网络系统的每一个实体中，联系相邻两层的通路就是接口(interface)。在接口的规范里，定义了下层向上层该提供哪些网络服务和操作。作为一个网络设计者，最重要的一点是考虑好如何定义层与层的接口，即设计好不同层之间该如何通信。

结合协议和接口的关系，给出如图 1-1 所示的 5 层网络的概念图。

在这张概念图中，可以很明显地得出如下结论：在对等实体之间通过协议来进行横向

通信，在单个实体内每一层通过服务接口来传递信息。



图 1-1 5 层网络概念图

3. 服务

从实现的细节来看，下层向上层所提供的服务分为两种形式：面向连接的服务和无连接的服务。

面向连接的服务是一种基于电话模型的服务。在连接过程中，必须要经过三个步骤来完成数据通信、建立连接、使用连接、释放连接的工作。如图 1-2 演示了面向连接服务的通信模型。

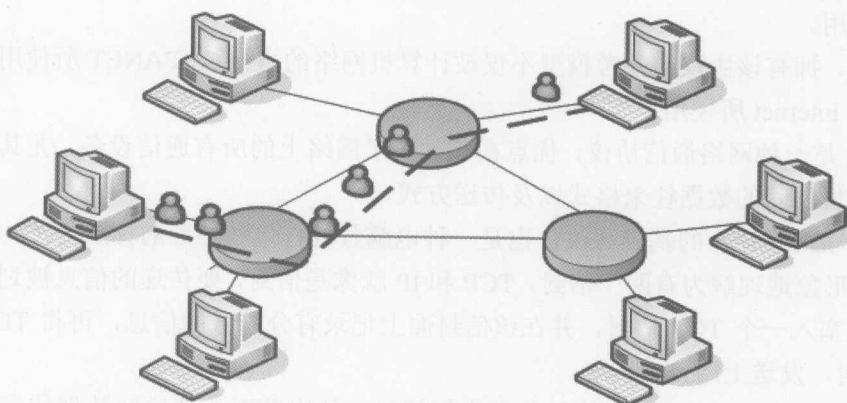


图 1-2 面向连接服务的通信模型

无连接的服务是一种基于邮政系统的模型。用户所发送的数据被分成相应的份数，每一条报文都携有完整的目标地址。其特点是每一条报文都相对独立，系统独立地进行路由。各个报文到达目的地的顺序可能是乱序。

如图 1-3 所示为无连接服务的通信模型。

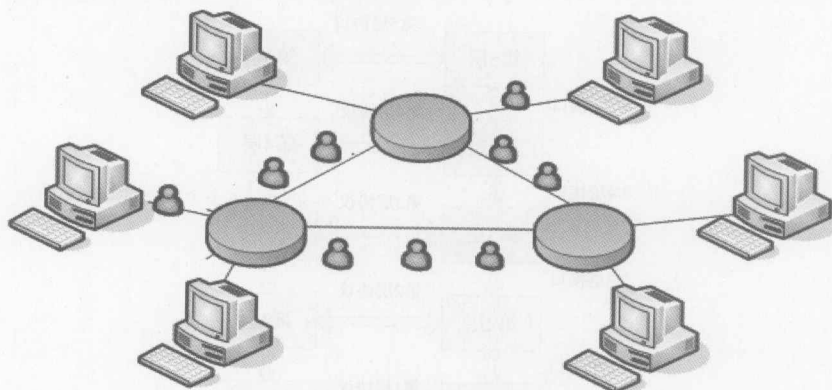


图 1-3 无连接服务的通信模型

在对比这两种服务的方式时，不要被它们的名字所迷惑，并不是“无连接”的服务，其通信质量就一定比“有连接”方式的差。

事实上，“有连接”的服务方式需要一定的代价来维护这条“通信信道”，所以，它比“无连接”的服务方式要耗费更多的网络资源，虽然它能提高通信数据传输的可靠性，但如果在“需重点考虑通信代价”的在线游戏数据传输的应用里，“无连接”服务方式的可行度就要比“有连接”的好。

1.1.3 TCP/IP 网络架构概述

从构造的角度来看，TCP/IP 网络架构的模型本身不是非常复杂，但这套协议却被广泛地流传和使用。

事实上，拥有该协议的参考模型不仅被计算机网络的祖先 ARPANET 所使用，也被之后的继承者 Internet 所使用。

TCP/IP 是一种网络通信协议，优点在于规范了网络上的所有通信设备，尤其一个主机与另一个主机之间的数据往来格式以及传送方式。

TCP/IP 是 Internet 的基础协议，也是一种电脑数据打包和寻址的标准方法。在数据传送中，可以形象地理解为有两个信封，TCP 和 IP 就像是信封，要传递的信息被划分成若干段，每一段塞入一个 TCP 信封，并在该信封面上记录有分段号的信息，再将 TCP 信封塞入 IP 大信封，发送上网。

而在接收端，一个 TCP 软件包负责收集信封，抽出数据，把这些数据按发送前的顺序还原，并加以校验，若发现差错，根据协议，接收端会发出“重发该数据”的请求。因此，TCP/IP 在 Internet 中几乎可以无差错地传送数据。如图 1-4 所示为 TCP/IP 参考模型。

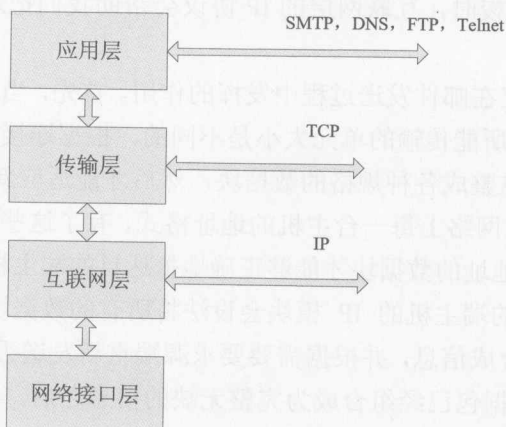


图 1-4 TCP/IP 参考模型

从图 1-4 来看，网络接口层是 TCP/IP 模型的底层。它的功能是控制网络硬件，并发送 IP 数据报。

实质上，网络接口层可以用来管理网络介质里的诸多协议，定义如何使用实际网络来传送数据。

互联网层是将整个网络体系结构贯穿在一起的关键层。它负责提供基本的封包传送功能，将信息分组传送到目标端，各个分组相互独立。互联网层定义了正式的分组格式和协议，就是我们所说的 IP。

传输层是提供对等实体间的数据传送服务，确定数据已被送达并接受。主要的协议有传输控制协议(TCP)和用户数据报协议(UDP)。

应用层是应用程序间沟通的协议，包含所有的高层协议。

从使用的角度来看，高层协议包含虚拟终端协议(Telnet)、文件传输协议(FTP)、电子邮件协议(SMTP)、现在的 DNS 域名系统、HTTP 超文本传输协议等。这些协议和实际应用密切相关。

上文已经讲述了 TCP/IP 各重要的网络通信层，下面将通过一个实例来了解一下基于 TCP/IP 网络架构的工作方式。

每当我们寄一份电子邮件的时候，首先需要打开收发电子邮件的程序，如 Outlook，并且规定邮件的收件人和寄件人姓名及地址，还有邮件的内容。上述的所有邮件资料都涵盖在电子邮件协议中。当填完邮件内容，并单击“发送”按钮后，电子邮件协议(SMTP)将根据传输层中的 TCP 模块，将整份邮件信息由本地发送到收件人的信箱。

在 TCP 协议中，规定了如何将信息正确无误地送抵目的端，TCP 模块先将信息切割打包成一块块数据包，根据记录及追踪送出的数据包，可以得知哪些数据包已经到达目的端主机。而对于那些没有到达目的端主机的资料包，根据反馈信息就必须再送一次，直到对方确定收到为止。

这些数据包将会在网络中路由前进，经过各种不同类型的网络及主机才能到达目的

端。当数据包在网络中穿梭时，互联网层的 IP 协议会帮助我们把这些包发送到它们的目的地。

下面来看一下 IP 协议在邮件发送过程中发挥的作用。首先，当数据包经过不同类型的网络时，由于每一种网络所能传输的单元大小是不同的，根据连接网络内的不同主机内的 IP 模块，需要把资料包重整成各种规格的数据块，然后才能适应新的网络环境进行传输。

此外，IP 也定义了在网上每一台主机的地址格式，有了这些可唯一确认每一台主机的地址，每一块包含 IP 地址的数据块才能够正确地抵达目的端主机。

到达目的端之后，目的端主机的 IP 模块会设法将所有数据块组合起来成为数据包，TCP 模块再将数据包组合成信息，并根据需要要求源端重新发送丢失的数据包。目的端主机上的 TCP 模块确定数据包已经组合成为完整无缺的信息之后，再通知源端主机的 TCP 模块，任务完成。最后，目的端主机上的邮件传输协议(SMTP)负责将信息转成收件人看得懂的邮件，与寄件人所发的原件一样。

从这一个过程中看出，IP 协议是 TCP/IP 的心脏，IP 协议能提供基本的封包传送服务，而 TCP 协议则规范了在网络里不同主机之间的建立连接的方式，这两个协议的相互协作，是 TCP/IP 网络赖以建立的基础。

1.1.4 IP 地址与端口

IP 地址是网络上标识主机的编号。而所谓的端口号则标识的是哪一个具体的应用程序。

打个比方来讲，IP 地址是构建在网上不同服务机构所在大楼的“门牌号码”，而端口号，则是具体服务机构里提供服务部门的“房间号”。

1. IP 地址

因特网是全世界范围内的计算机联为一体而构成的通信网络的总称，有成千上万台主机在网上漫游。为了区分这些主机，我们给每一台主机分配一个专门的地址，称为 IP 地址。IP 地址由国际组织 NIC(Network Information Center)负责统一分配。IP 地址包括网络号和主机号，网络号和主机号是 IP 地址唯一的组合。IP 地址由 4 部分数字组成，每部分数字对应于 8 位二进制数字，各部分之间用小数点分开。如某一台主机的 IP 地址为 192.168.65.112。原则上，Internet 上的任意两台机器不会有相同的 IP 地址。

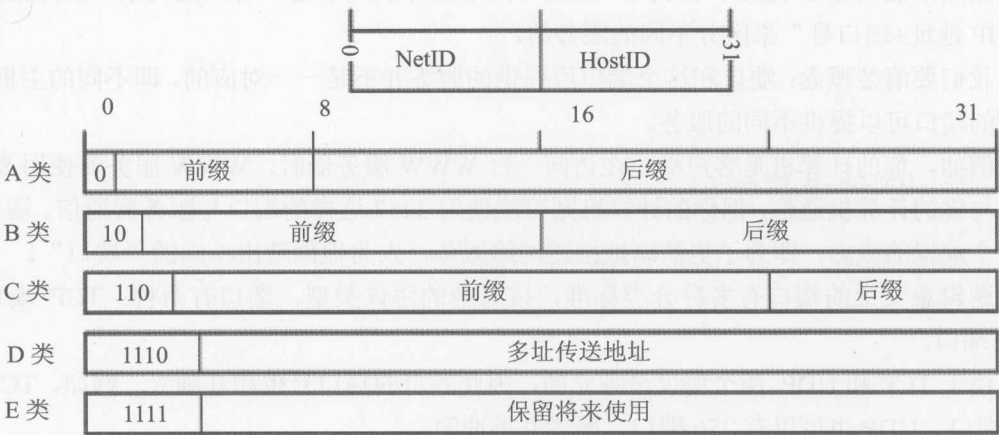
许多人会认为 IP 地址其实代表着一台主机，是一台主机在网络上的唯一标识号。其实不然，从功能角度来看，IP 地址真正引用的是一个网络接口，所以，我们可以假设如果一台主机位于两个网络上，那么这台主机可以并且必须拥有两个 IP 地址。然而，在实际的应用中，大多数的网络主机都在一个网络上运行，所以一般对于一台主机而言，只有一个 IP 地址。

从构建的形式上来看，因特网是网中套网，网中嵌网，即每个网络中的计算机通过其自身的 IP 地址而被唯一标识。

据此，也可以设想，在 Internet 这个庞大的网间网中，每个网络也有自己的标识符。例如，日常生活中的电话号码 021-64646464，号码中的 021 表示该电话是属于哪个地区，后面的数字表示该地区的某个电话号码。与上例类似，计算机的 IP 地址也分成两部分，分别为网络号和主机号。

同一个物理网络上的所有主机都用同一个网络标识，网络上的一个主机(包括网络上工作站、服务器和路由器等)都有一个主机标识与其对应的 IP 地址。IP 地址的 4 个字节划分为 2 个部分，一部分用以标明具体的网络段，即网络号；另一部分用以标明具体的节点，即主机号。

根据每个网络中所包含的主机的多少，有的网络可能含有较多的计算机，也有的网络包含较少的计算机。按照网络规模的大小，为了便于寻址和层次化的构造网络，IP 地址被分为 A、B、C、D、E 五类。商业应用中只用到 A、B、C 三类。如图 1-5 所示为 IP 地址分类图。



如表 1-1 所示为特殊的 IP 地址。

表 1-1 特殊 IP 地址

NetID	HostID	地 址 类 型	用 途
全 0	全 0	本机	启动时使用
网络 ID	全 0	网络	标识一个网络
网络 ID	全 1	直接广播	在指定网上广播
全 1	全 1	有线广播	在本地网上广播
127	任意	回送	测试 127.0.0.1

表 1-1 中的特殊地址，虽然不会像常规 IP 地址那样，有较高的使用频率，但像一些“广播地址”，却是在特定的应用场合(如群发)有不可替代的作用。这些在后面的章节中会讲到。

2. 端口

在网络技术中, 端口(Port)有许多方面的理解。hub、switch、router 的端口指的是连接其他网络设备的接口, 如 RJ-45 端口、Serial 端口等。这里所指的端口不是指物理意义上的端口, 而是特指 TCP/IP 中的端口, 理解为逻辑意义上的端口。

把一个居民小区比喻成一个网络, 每户人家是这个小区的成员, 就像网络中的主机一样, 那用不同的 IP 地址来标识每户人家, 而端口就像每户小区居民的房屋大门。通过这个比喻就可以理解什么是 IP 地址的端口。真正的房子只有几个门, 但是一个 IP 地址的端口可以多达 65536 个。

端口是通过端口号来唯一标记的, 端口号只有整数, 它的范围是 0~65535。

一台拥有 IP 地址的主机可以提供许多服务, 如 Web 服务、FTP 服务、SMTP 服务等, 这些服务完全可以通过一个 IP 地址来实现。那么, 这么多网络服务主机怎样才能区别开呢? 显然不能只靠 IP 地址, 因为 IP 地址与网络服务的关系是一对多的关系。实际上是通过“IP 地址+端口号”来区分不同的服务的。

我们要清楚概念: 端口和这个端口所提供的服务并不是一一对应的, 即不同的主机上, 相同的端口可以提供不同的服务。

例如, 你的计算机是客户端, 在访问一台 WWW 服务器时, WWW 服务器使用 80 号端口与你的计算机通信, 但你的计算机则可能使用 3457 这样的端口与服务器通信。端口只是一个虚拟的概念, 即为了更清晰地描述网络服务, 人为地创造出不同的“端口”。

逻辑意义上的端口有多种分类标准, 按对应的协议类型, 端口有两种: TCP 端口和 UDP 端口。

由于 TCP 和 UDP 两个协议是独立的, 因此各自的端口号也相互独立。例如, TCP 有 256 端口, UDP 也可以有 256 端口, 两者并不冲突。

(1) TCP 端口

TCP 端口, 即传输控制协议端口, 需要在客户端和服务端之间建立连接, 这样可以提供可靠的数据传输。常见的包括 FTP 服务的 21 端口, Telnet 服务的 23 端口, SMTP 服务的 25 端口, 以及 HTTP 服务的 80 端口等。

(2) UDP 端口

UDP 端口, 即用户数据包协议端口, 无需在客户端和服务端之间建立连接, 安全性得不到保障。常见的有 DNS 服务的 53 端口, SNMP(简单网络管理协议)服务的 161 端口, QQ 使用的 8000 和 4000 端口等。

除了按照对应协议的分类外, 还有按照端口号分布划分的知名端口和动态端口。

(1) 知名端口

知名端口即家喻户晓的端口号, 范围为 0~1023, 这些端口号一般都是固定分配给一些常用的服务。例如, 21 端口分配给 FTP 服务, 25 端口分配给 SMTP(简单邮件传输协议)服务, 80 端口分配给 HTTP 服务, 135 端口分配给 RPC(远程过程调用)服务等。