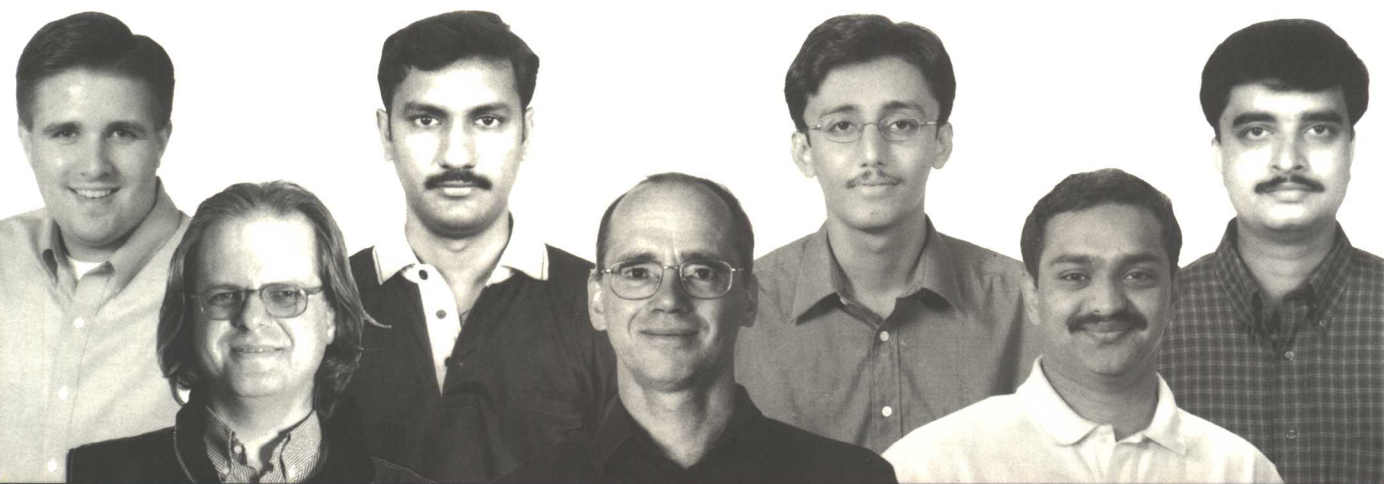


PROGRAMMER TO PROGRAMMER™



Professional .NET Network Programming

# .NET

# 网络高级编程

Andrew Krowczyk  
Vinod Kumar  
吴旭超

等著  
译



清华大学出版社

# .NET 网络高级编程

Andrew Krowczyk      等著  
Vinod Kumar  
吴旭超                      译

清华大学出版社

北 京

# 北京市版权局著作权合同登记号：01-2002-3199

## 内 容 简 介

网络编程是.NET 企业级应用程序开发的重要内容，也是开发人员当前所面临的难点之一。本书共分为 11 章，全面深入地介绍了.NET 网络编程技术，内容分别涉及到物理网络的体系结构、网络协议和 OSI 模型、.NET 中的套接字编程、TCP 和 UDP、HTTP 和多播套接字、Internet 编程和用于发送电子邮件的.NET 类、.NET 中的密码术以及身份验证协议等。

本书适合那些熟悉 C# 基础编程知识、并希望切实掌握.NET 网络编程技术的各类开发人员；本书并不要求您先前了解网络编程，但具有网络编程的相关知识有利于学习本书。

EISBN: 1-86100-735-3

Professional .NET Network Programming

Andrew Krowczyk, Vinod Kumar et al

Copyright©2002 by Wrox Press Ltd.

Original English language Edition Published by Wrox Press Ltd.

All Rights Reserved.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

**版权所有，翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。**

### 图书在版编目(CIP)数据

.NET 网络高级编程/(美)克罗克齐科等著；吴旭超译.—北京：清华大学出版社，2003

书名原文：Professional .NET Network Programming

ISBN 7-302-06323-0

I. N... II. ①克...②吴... III. 计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字(2003)第 009011 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)

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

责任编辑：李阳

印 刷 者：北京通州区大中印刷厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 印张：27 字数：691 千字

版 次：2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号：ISBN 7-302-06323-0/TP·4771

印 数：0001~4000

定 价：56.00 元

## 出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 [www.wrox.com](http://www.wrox.com) 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 [p2p.wrox.com](http://p2p.wrox.com) 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

# 前 言

网络编程是企业级开发中的一项核心任务——不同的计算机(不管是在同一栋建筑中,还是遍布在世界各地)之间能够高效并且安全地进行通信这一需求,仍然是许多系统成功的基础。在.NET Framework 中随带了一组新的类,这些类用于处理联网(networking)任务。

在阅读本书后,您将成为一个自信的.NET 网络编程人员,并对底层的协议有一定的了解。.NET 类所支持的当前协议集仅限于传输层协议 TCP 和 UDP,以及应用层协议 HTTP 和 SMTP。在本书中,我们不仅详尽地介绍了这些类,而且还提供了在.NET 中实现应用层协议的示例——因此,无论是对于那些需要使用.NET 当前不支持的协议的读者,还是任何想要认真处理预定义协议的读者,本书都将是一本必备的读物。

## 本书主要内容

在第 1 章中,首先对一些联网的基本概念和协议进行了介绍。无论对网络编程有什么样的要求——假设您计划开发作为 Windows 服务运行的服务器应用程序,利用一个自定义协议为客户提供数据,假设您想要编写从 Web 服务器请求数据的客户应用程序,或者想要创建多播应用程序,或者是创建使用邮件功能的应用程序,你都需要首先阅读本章。本章首先介绍了物理网络以及在局域网中所使用的硬件,接着,进一步介绍了 OSI 7 层模型,以及 TCP/IP 协议套件是如何与 OSI 各层相适应的。此后,讲解了有关各种网络、Internet 以及电子邮件协议的内容。

第 2 章为我们提供了使用流的背景——流是某些串行设备的一个抽象表示,这些设备每次只存储和检索一个字节的数据——底层设备可以是一个文件、一台打印机或者一个网络套接字等。通过该抽象化,您可以利用同一进程来访问不同的设备,并且可以使用某些代码从一个网络输入流读取数据,同时使用类似的代码从一个文件输入流读取数据。这样就可以使编程人员不用考虑设备的实际物理机制。在本章中,我们介绍了.NET 中的流——Stream 类,并且使用了具体的 FileStream 类。还包括了读取和写入二进制文件和文本文件,以及将对象串行化为 XML 和二进制格式的内容。

在第 3 章中,我们开始利用 System.Net 命名空间中的类来处理.NET 中的网络编程。首先对这些类本身进行了论述——它们在本书剩余的各章中都扮演了重要的角色。特别是,我们介绍了 URI、IP 地址以及 DNS 查找等内容,以及如何通过 WebRequest 和 WebResponse 类来处理请求和响应,并且开始考虑与网络编程相关的身份验证、授权和许可的问题。

第 4 章讲述了有关套接字编程的内容,并且包括了执行网络相关任务的底层编程。套接字是在网络上运行的两个程序之间的双向通信链路的其中一端。我们介绍了.NET 中的套接字支持——System.Net.Sockets.Socket 类,并创建了同步和异步客户/服务器应用程序。

在第 5 章中,论述了.NET Framework 中较高级别的网络类,首先介绍的是那些用于处理



Transmission Control Protocol(传输控制协议, 简写 TCP)的类。开始我们对 TCP 和它的体系结构以及数据结构进行了概要的介绍,接着深入探讨了用于处理 TCP 的 TcpClient 和 TcpListener 类。我们用 TcpClient 和 TcpListener 类构建了客户/服务器应用程序,也构建了一个功能完备的电子邮件客户程序以验证 TcpClient 的能力,还利用 .NET 多线程类的支持创建了一个多线程响应服务器。在本章结尾处,我们快速浏览了 .NET Remoting Framework,特别是随同 .NET Framework 一起提供的 TcpChannel 传输信道。

第 6 章介绍了有关 UdpClient 类的内容,我们通过该类来实现 User Datagram Protocol(用户数据报协议, 简写 UDP)。首先介绍的是 UDP 协议的一些基本知识,接着介绍了 UdpClient 类的用法。虽然 TCP 是一种比 UDP 更为可靠的协议,但它也增加了许多系统开销。因此,相比较而言,UDP 的速度更快,并且很适合如视频流之类的多媒体传输,这时候数据包到达的准确顺序可能并不是很重要。在这一章中,还简单介绍了较高级的基于 UDP 的协议。

第 7 章向您介绍了多播技术。利用这种技术,我们能够通过 Internet 来实况传输一场 1994 年的 Rolling Stones 音乐会,并能够观看位于太空中的宇航员,或者与其他人在 Internet 上举行会议。利用多播技术,服务器只需发送消息一次,这些消息就能够被发布到整个客户组。在本章的开头,我们对单播、广播和多播进行了比较,并介绍了多播的体系结构,以及在 .NET 中实现多播套接字的方法。我们使用多播功能创建了两个 Windows 应用程序——第一个应用程序使得能和多个系统聊天,其中的每个系统都既是发送方也是接收方。第二个应用程序以画展的形式演示了如何将大数据包发送到多个客户,而无须使用较高的网络带宽。

第 8 章包含了 HTTP 协议,以及 .NET 为它提供的健壮的实现工具。作为一个应用程序协议,HTTP 协议具有非常显著的重要性,因为当今很大一部分的 Web 通信量都使用该协议。在本章中,首先对 HTTP 协议进行了概述——HTTP 报头以及 HTTP 请求和响应的格式。我们还介绍了 .NET 中用于处理 HTTP 的类,并讲述了读取和编写 cookie 的方法。接着,创建了一个带有 ASP.NET 支持的 HTTP 服务器,然后继续介绍 .NET Remoting 以及 HTTP 传输信道。

第 9 章开始介绍电子邮件。在这一章中,首先对各种电子邮件协议和它们在 .NET 环境中的访问和使用的方法进行了高级概述。并介绍了 SMTP、POP3、IMAP 和 NNTP 协议的基本内容,以及这些协议如何协同工作,以通过 Internet 发送和接收电子邮件消息。另外还讲述了利用 .NET Framework 的类(这些类用于通过 SMTP 发送电子邮件)发送电子邮件,还有为 POP3 和 SMTP 开发某些底层协议实现类的内容。

第 10 章中介绍有关保护网络通信的内容。.NET Framework 中的 System.Security.Cryptography 命名空间提供了对各种加密服务的程序式访问,我们可以将其合并到应用程序中,以加密和解密数据、确保数据完整性以及处理数字签名和证书。在本章中,深入探讨了该命名空间,但同时也介绍了密码术和它的所有关键/密钥概念(希望您能原谅这一双关语)。我们还介绍了如何保障前面在第 6 章中所创建的聊天应用程序的安全。

## 本书读者对象

本书并不假定您先前就具备有关网络编程方面的知识,因此,其中适当地包含了基本和较高级的联网概念。当然,如果您早已通过其他环境熟悉了网络编程技术,则将大有裨益,因为

这样可以提高您掌握本书内容的速度和深度。

#### 说明:

本书中所有的代码示例都是用 C# 编写的, 因此, 您需要拥有 C# 语言的基础编程知识。

## 使用本书的条件

运行本书中的示例的先决条件是您需要有一台装有 .NET Framework 的计算机。这意味着您需要运行以下两者之一:

- Windows 2000 Professional(或更高版本)
- Windows XP

推荐您在阅读本书的同时使用某一版本的 Visual Studio .NET。

## 用户支持

我们一贯重视您的意见, 并想知道您对本书的看法, 包括您喜欢哪些内容和不喜欢哪些内容, 以及您希望我们下一次需要完善的地方。您可以通过发送电子邮件(地址为 [feedback@wrox.com](mailto:feedback@wrox.com))来向我们反馈意见。请确保在反馈信息中提到本书的书名。

### 如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时, 通过 Search 工具或书名列表, 可以方便地定位到所需要的书目。然后, 单击 Code 栏中的 Download 超链接, 或者单击本书的详细信息页面中的 Download Code 超链接, 就可以下载相应的示例代码。

从我们的站点上下载的文件都是已使用 WinZip 压缩过的文档。保存文件到本地磁盘上的文件夹中后, 需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时, 通常将代码解压缩到每一章所在的文件夹中。在解压缩过程中, 应确保解压缩程序(WinZip、PKUnzip 等)被设置为使用原有的文件夹名。

### 勘误表

我们已经尽最大努力确保本书中的文本和代码不出错, 但是错误仍然在所难免。如果您发现本书存在错误, 例如拼写错误或不正确的代码段, 请反馈信息给我们, 我们将不胜感激。勘误表的发送可以节省其他读者学习本书的时间, 而且能够帮助我们提供更高质量的信息。您的反馈信息将被检查, 如果正确, 将被粘贴到本书的勘误表页面上, 或者在本书的后续版本中使用。

要在我们的站点上找到勘误表, 请访问 <http://www.wrox.com/>, 并通过 Advanced Search 或者书名列表轻松定位到本书页面。然后, 单击 Book Errata 超链接即可, 该链接位于本书的详细信息页面中的封面图解下面。



## E-mail 支持

如果您希望直接向详细了解本书的专家咨询本书中问题，可以发送电子邮件到 [support@wrox.com](mailto:support@wrox.com)，要求在邮件的主题栏中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一封典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题所在的页码。
- 邮件正文中应包括读者的名字、联系信息和问题。

我们不会给您发送无用邮件，因为我们仅仅需要有用的详细资料，以便可节约您和我们的时间。当您发送一封电子邮件信息时，它将经过下面一系列支持：

- 用户支持：首先，您的信息将被递送到用户支持人员手中，并由他们阅读。对于一些被频繁提到的问题将被归档，并将立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持：接着，一些深层次的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的详细技术问题。
- 作者支持：最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持，对于超出常规书目支持的问题，您可以从 <http://p2p.wrox.com> 论坛中的公共列表中获得支持信息。

### p2p.wrox.com 站点

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们独特的系统将 programmer to programmer™(由程序员为程序员而著)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 [p2p.wrox.com](http://p2p.wrox.com) 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表：

- (1) 登录 <http://p2p.wrox.com> 站点。
- (2) 从左边的菜单栏选择一个适当的类别。
- (3) 单击希望加入的邮件列表。
- (4) 按照说明订阅并填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用预订管理程序加入更多的邮件列表并设置自己的邮件首选项。

### 本系统提供最好支持的原因

您可以加入整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请发送邮件到 [listsupport@p2p.wrox.com](mailto:listsupport@p2p.wrox.com)。



# 目 录

第 1 章 联网概念和协议 .....	1
1.1 物理网络 .....	1
1.1.1 WAN 线路 .....	2
1.1.2 Ethernet 协议 .....	2
1.1.3 物理组件 .....	4
1.2 分层 OSI 模型 .....	9
1.2.1 第一层: 物理层 .....	11
1.2.2 第二层: 数据链路层 .....	11
1.2.3 第三层: 网络层 .....	12
1.2.4 第四层: 传输层 .....	12
1.2.5 第五层: 会话层 .....	12
1.2.6 第六层: 表示层 .....	12
1.2.7 第七层: 应用层 .....	13
1.3 网络协议 .....	13
1.3.1 基本协议 .....	13
1.3.2 Internet 协议 .....	23
1.3.3 电子邮件协议 .....	27
1.3.4 其他应用程序协议 .....	28
1.4 套接字 .....	29
1.5 域名 .....	30
1.5.1 whois 服务 .....	31
1.5.2 域名服务器 .....	31
1.6 Internet .....	32
1.6.1 Intranet 和 Extranet .....	32
1.6.2 防火墙 .....	33
1.6.3 Web 代理 .....	33
1.6.4 XML Web 服务 .....	34
1.7 Messaging .....	36
1.8 访问网络对象的其他方法 .....	36
1.9 Internet 组织和标准 .....	37
1.10 小结 .....	37



<b>第 2 章 .NET 中的流</b> .....	<b>39</b>
2.1 .NET 中的流.....	39
2.1.1 同步和异步 I/O.....	39
2.1.2 Stream 类.....	40
2.1.3 FileStream 类.....	43
2.1.4 利用 FileStream 进行读/写.....	45
2.1.5 BufferedStream 类.....	50
2.1.6 MemoryStream 类.....	50
2.1.7 NetworkStream 类.....	51
2.1.8 CryptoStream 类.....	54
2.2 流操纵.....	57
2.2.1 编码字符串数据.....	57
2.2.2 二进制文件.....	60
2.2.3 TextReader 类.....	63
2.2.4 TextWriter 类.....	65
2.3 串行化.....	67
2.3.1 串行化为 XML 格式.....	68
2.3.2 用格式化程序对象进行串行化.....	71
2.4 小结.....	75
<b>第 3 章 .NET 中的网络编程</b> .....	<b>76</b>
3.1 System.Net 类——概述.....	76
3.1.1 名称查找.....	76
3.1.2 IP 地址.....	76
3.1.3 身份验证和授权.....	77
3.1.4 请求和响应.....	77
3.1.5 连接管理.....	78
3.1.6 Cookies.....	78
3.1.7 代理服务器.....	79
3.1.8 套接字.....	79
3.2 使用 URIs.....	80
3.2.1 Uri 类.....	81
3.2.2 Uri 类的属性.....	82
3.2.3 用 UriBuilder 类修改 URI.....	84
3.2.4 绝对和相对 URI.....	84
3.3 IP 地址.....	85
3.3.1 预定义地址.....	86
3.3.2 本机或网络字节顺序.....	86

3.4	Dns 类	87
3.4.1	将名称解析为 IP 地址	87
3.4.2	解析 IP 地址的方式	88
3.4.3	异步解析 IP 地址	90
3.5	请求和响应	91
3.5.1	WebRequest 类和 WebResponse 类	92
3.5.2	可插接式协议	94
3.5.3	FileWebRequest 类和 FileWebResponse 类	95
3.5.4	连接池	97
3.6	使用 Web 代理	98
3.6.1	WebProxy 类	99
3.6.2	默认的 Web 代理	99
3.6.3	为特定的请求修改 WebProxy	100
3.7	身份验证	100
3.8	权限	101
3.8.1	使用权限属性	102
3.8.2	配置权限	104
3.9	小结	107
<b>第 4 章</b>	<b>使用套接字</b>	<b>108</b>
4.1	套接字	108
4.2	在 .NET 中使用套接字	111
4.2.1	System.Net.Sockets.Socket 类	111
4.2.2	创建一个 TCP 流式套接字应用程序	112
4.2.3	System.Net.Sockets 中的异常管理	121
4.2.4	套接字选项	124
4.3	异步编程	126
4.3.1	异步客户应用程序	127
4.3.2	异步服务器应用程序	132
4.4	套接字权限	137
4.5	小结	144
<b>第 5 章</b>	<b>TCP</b>	<b>145</b>
5.1	TCP 概述	145
5.1.1	封装	145
5.1.2	TCP 术语	146
5.1.3	TCP 报头	146
5.1.4	TCP 连接	146
5.1.5	TCP 操作	147



5.2	.NET 中的 TCP 概述	149
5.2.1	TcpClient 类	149
5.2.2	构建一个真正的套接字应用程序	155
5.2.3	实现 FtpWebRequest 类	160
5.2.4	TcpListener 类	178
5.3	.NET Remoting	186
5.3.1	应用程序域	186
5.3.2	Remoting 的工作方式	187
5.4	小结	195
<b>第 6 章</b>	<b>UDP</b>	<b>196</b>
6.1	UDP 协议概述	196
6.1.1	UDP 的术语	197
6.1.2	UDP 的工作方式	199
6.1.3	UDP 的缺点	200
6.1.4	UDP 的优点	201
6.1.5	使用 UDP 的时机	202
6.2	.NET 中的 UDP	202
6.2.1	UdpClient 类	203
6.2.2	创建一个使用 UDP 的聊天应用程序	214
6.3	文件传输应用程序	219
6.3.1	文件服务器	219
6.3.2	文件接收器	222
6.3.3	广播	225
6.4	基于 UDP 的高级协议	225
6.5	小结	227
<b>第 7 章</b>	<b>多播套接字</b>	<b>228</b>
7.1	单播、广播和多播	228
7.2	使用多播的应用程序模型	229
7.3	多播套接字的体系结构	231
7.3.1	IGMP 协议	231
7.3.2	多播地址	232
7.3.3	路由	234
7.3.4	可伸缩性	237
7.3.5	可靠性	237
7.3.6	安全性	238
7.4	利用 .NET 使用多播套接字	238
7.4.1	发送方	238

7.4.2 接收方	238
7.5 创建聊天应用程序	240
7.5.1 用户界面	240
7.5.2 配置设置	242
7.5.3 加入多播组	243
7.5.4 接收多播消息	244
7.5.5 发送多播消息	245
7.5.6 撤销多播成员	246
7.5.7 启动聊天应用程序	247
7.6 画展应用程序	247
7.6.1 画展解决方案	248
7.6.2 创建一个图片协议	248
7.6.3 画展服务器	254
7.6.4 画展客户程序	264
7.7 小结	272
<b>第 8 章 HTTP</b>	<b>273</b>
8.1 HTTP 协议概述	273
8.1.1 HTTP 报头	275
8.1.2 HTTP 请求	276
8.1.3 HTTP 响应	278
8.2 .NET 中的 HTTP	279
8.2.1 HttpWebRequest 类和 HttpWebResponse 类	280
8.2.2 货币转换器应用程序	282
8.2.3 发送数据	284
8.2.4 HTTP 分块	286
8.2.5 HTTP 保持有效连接	288
8.2.6 HTTP 连接管理	288
8.2.7 WebClient 类	289
8.2.8 身份验证	296
8.2.9 代理支持	298
8.2.10 读/写 cookies	299
8.3 提供 ASP.NET 支持的 HTTP 服务器	303
8.3.1 服务器配置文件	304
8.3.2 为服务器编码	305
8.4 HTTP 和 .NET Remoting	316
8.5 小结	319



<b>第 9 章</b>	<b>电子邮件协议</b>	<b>320</b>
9.1	电子邮件简介	320
9.2	电子邮件的工作方式	320
9.3	电子邮件协议	321
9.3.1	SMTP	322
9.3.2	典型的电子邮件消息	326
9.3.3	在 Outlook 中查看报头	328
9.3.4	MIME	329
9.3.5	检索客户/服务器电子邮件	331
9.4	.NET 和电子邮件	335
9.4.1	SMTP	335
9.4.2	SMTP 邮件应用程序	339
9.4.3	POP3	340
9.4.4	NNTP	348
9.5	小结	357
<b>第 10 章</b>	<b>.NET 中的密码术</b>	<b>358</b>
10.1	密码术的历史	358
10.2	密码术的定义	358
10.3	使用密码术的原因	360
10.4	加密算法	361
10.4.1	对称算法	362
10.4.2	不对称算法	363
10.4.3	消息分类算法	364
10.4.4	数字签名	365
10.5	密码术的术语	366
10.5.1	块式密码和流式密码	366
10.5.2	填充	366
10.5.3	模式	367
10.6	System.Security.Cryptography 命名空间	368
10.7	用.NET 进行散列运算	369
10.8	用.NET 进行对称转换	376
10.8.1	SymmetricAlgorithm 类	376
10.8.2	使用其他的对称算法	380
10.9	用.NET 进行不对称转换	381
10.9.1	AsymmetricAlgorithm 类	381
10.9.2	使用 RSA 算法	382
10.9.3	加载公钥和私钥	385

---

10.10	读取 X509 证书 .....	388
10.11	密码术和网络编程 .....	390
10.12	小结 .....	396
<b>第 11 章</b>	<b>身份验证协议 .....</b>	<b>398</b>
11.1	身份验证协议 .....	398
11.1.1	NTLM .....	398
11.1.2	Kerberos .....	402
11.1.3	.NET 安全性和 Windows .....	406
11.1.4	System.Net.IAuthenticationModule 类 .....	410
11.2	小结 .....	412

# 第1章 联网概念和协议

在本章中，我们将介绍一些基本的联网概念和协议。它将作为网络的基础，使我们能够处理在本书剩余部分中的编程内容。无论是计划开发作为 Windows Services 运行的服务器应用程序，由它使用某个自定义的协议来为客户提供数据，或者是编写从 Web 服务器请求数据的客户应用程序，再或者是创建多播应用程序或者是使用邮件功能的应用程序，这都无关紧要，因为本章是实现所有这些目标的必读章节。如果不知道什么是路由器，什么是网络交换机，如果尚未确切地了解 OSI 协议中 7 个层的功能，或者只是想对不同的网络协议和它们的用法作概要的了解，或是要更新自己的知识结构，本章都是您最佳的起点。

首先，介绍在局域网中所使用的一些硬件，如路由器、集线器和网桥。接着，介绍 OSI 模型的 7 个层和它们的功能，以及 TCP/IP 协议套件是如何与 OSI 各层相适应的。在此之后，我们将学习各种不同网络协议的功能。

**本章主要内容如下：**

- 物理网络
- OSI 7 层模型
- 基本网络协议
- 网际协议(IP)
- 电子邮件协议
- 套接字
- 名称查找
- Internet
- Remoting
- Messaging

## 1.1 物理网络

就本质上而言，网络就是通过通信链路连接到一起的一组计算机或设备。用网络术语来说，连接到网络的每台计算机或设备(打印机、路由器和交换机等)都称为节点。节点通过链路连接到一起，该链路可以是电缆或无线链路(如红外线或广播信号)，它们通过在网络上传输消息能够与其他任何节点进行交互。

我们可以按照网络的大小来对它们进行区分：

- LAN(Local Area Network, 局域网)在有限的区域内连接节点。该区域可以大到一个大公司的站点，或小到某人家中的一些相互连接的计算机。其中最为通用的 LAN 技术是





Ethernet 网络(参见下一节)。

- WAN 是 Wide Area Network(广域网)的缩写。WAN 将多个 LAN 站点连接到一起。您可能知道的 WAN 技术包括帧中继、T1 线路、ISDN(Integrated Services Digital Network, 综合业务数字网)、X.25 和 ATM(Asynchronous Transfer Mode, 异步传输模式)。在下一节中,我们将进一步论述连接到 WAN 的方法。
- MAN(Metropolitan Area Network, 城域网)与 WAN 很相似,因为它也连接多个 LAN。然而,MAN 将网络的区域限定在一个城市或者市郊之内。MAN 使用高速网络将学校、政府部门、公司和其他机构连接到一起,这时要利用到各个站点的快速连接,如光纤。

#### 注意:

在论述网络时经常会用到主干(backbone)这个词。主干是一种连接低速网络的高速网络。公司可以用一个主干来连接低速的 LAN 段。Internet 主干由那些承担 WAN 通信量的高速网络构成。Internet 供应商或者是直接连接到 Internet 主干,或者是连接到一个更大的供应商(后者直接连接到 Internet 主干)。

### 1.1.1 WAN 线路

要连接到一个 WAN,可以有多种选择:

- 如果某个特定的客户需要一个指定的网络容量,则可以使用专线(leased line)。通常情况下,无论所发送通信量的多少,这种线路都采用包价收费制。

专线的示例包括 Digital Data Service(数字数据服务,简写 DDS,运行速度为 2.4Kbps 和 56Kbps), T1(1.544Mbps)和 T3(与 28 条 T1 线路相当)。

- 常规的电话服务都使用交换线路(switched line)。在进行通话或者数据交换期间,传输器和接收方之间将建立一条线路。当不再需要该线路时,它将被释放以供网络供应商的其他客户使用。

交换线路的示例有 POTS(Plain Old Telephony Service, 简易老式电话业务——标准模拟线路,支持的最高速度为 56Kbps)、ISDN、DSL(Digital Subscriber Line, 数字用户线路)。

- 在数据包交换(packet-switching)网络中,服务提供商提供了到某种接口(具有主干网络)的交换技术。该解决方案提供了增强的性能并且可以在客户之间共享资源,从而使带宽能够即需即用。

用于交换网络的协议包括 X.25(最高可达 64Kbps), 帧中继(最高可达 44.736Mbps)和 ATM(最高可达 9.953Gbps)。

### 1.1.2 Ethernet 协议

为了对物理网络的工作方式有更好的了解,我们将介绍最为通用的 LAN 协议——Ethernet。90%连接到 LAN 的设备都使用 Ethernet 协议,该协议最初由 Xerox、Digital Equipment 和 Intel 在 1972 年开发成功。在 1980 年,IEEE 802.3 的 CSMA/CD 标准指定了一个 10Mbps 的 Ethernet。

现在, Ethernet 能够支持 100Mbps 和 1Gbps 的线路。对于 Ethernet,我们可以采用许多电缆技术。存在一种标准的命名,该命名指示了 Ethernet 网络的速度和所使用电缆技术的属性。