

Visual C++ .NET网络与通信

高级编程范例

马 展 李守勇 等编著

- ▲ 网络与通信编程基础
- ▲ 获取网络配置信息
- ▲ Ping和TraceRoute程序
- ▲ Telnet服务器/客户机
- ▲ 基于C/S模型的聊天室
- ▲ FTP客户机
- ▲ 多线程下载程序
- ▲ 串口通信
- ▲ RAS客户机与拨号连接
- ▲ 语音电话
- ▲ Web浏览器
- ▲ Web服务器
- ▲ E-mail客户程序



清华大学出版社

Visual C++ .NET

网络与通信高级编程范例

马 展 李守勇 等编著

清华大学出版社

北 京

内 容 简 介

本书详细介绍了利用 Visual C++ .NET 集成开发环境开发网络通信应用程序的方法,同时对当今流行的 FTP、E-mail、Telnet、HTTP、ICMP、串口通信等通信协议程序设计和应用开发进行了详细的介绍和讲解。

本书在进行理论介绍的同时,配合以大量的实例,让读者能够深入地了解各种网络应用程序的开发技巧。另外,结合具体的设计实例,本书还讲解了网络通信协议的工作原理等内容。

本书面向具有 Visual C++ .NET 网络通信编程基础知识的读者,此外也适合于中、高级程序员在网络和通信应用开发时阅读和参考。

版权所有,翻印必究。举报电话:010-62782989 13801310933

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。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual C++ .NET 网络与通信高级编程范例/马展,李守勇等编著. —北京:清华大学出版社,2005.1

ISBN 7-302-10260-0

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

中国版本图书馆 CIP 数据核字(2004)第 141290 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:孟毅新

文稿编辑:鲍芳

封面设计:久久度文化

版式设计:康博

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市兴旺装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印 张:28.25 字 数:652 千字

版 次:2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

书 号:ISBN 7-302-10260-0/TP·6995

印 数:1~4000

定 价:49.80 元(含光盘)

本书如存在文字不清、漏印以及缺页、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)62770175-3103 或(010)62795704

前 言

展望未来的 IT 产业,网络将成为世界的中心。面对这个已经存在或者即将来到的世界,IT 行业也在进行飞速的变革: .NET 是微软为 Web Service——以统一的个性化的方式联系信息世界、设备和人的下一代互联网软件所提供的平台。

微软公司发布的 Visual Studio.NET,进一步改善了原来 Visual Studio 6.0 集成开发环境,特别是在 Visual C++.NET 中,增加了 C++的托管功能,并且沿用了 ISO/ANSI 标准。托管的 C++不仅改善了内存的管理,而且能够访问 .NET 类库的所有功能,还允许与已有的 C++代码、COM 对象以及 Win32 API 进行交互操作。

网络的飞速发展使得网络应用程序的开发地位越来越重要,而且网络应用程序的开发和传统应用程序的开发在思想上和实现上有很大的区别,很多软件公司都开发出了它们各自的网络应用程序开发环境,如 Delphi, C++Builder 等。相比之下,微软公司的 Visual Studio.NET 集成开发环境在进行网络应用程序开发方面有如下特点:

- 开发出的应用程序占用资源少。
- 使用扩展了的 MFC 类库,可以很大程度的简化 Web 服务器开发过程。
- 减少了源代码的开发量。
- 开发出的应用程序有很强的扩展性。

基于网络通信编程日益盛行的发展趋势,以及 Visual C++.NET 对网络通信编程所提供的方便快捷的接口和全面的技术支持等方面考虑,我们编写了本书。和其他的 Visual C++.NET 网络通信编程不同的是,本书注重理论和实践细节的有机结合,这样既可提供给读者丰富的网络编程细节和编程思想,又可提高读者的实际开发能力。在本书配套光盘中,附有书中全部实例的源代码和可执行程序,以方便读者学习和提高。

在结构安排上,本书第 1 章介绍了基础知识,包括 .NET 编程开发环境和网络协议;从第 2 章开始,每章都以一个大实例的方式,从理论到实践循序渐进指导读者使用 Visual C++.NET 来进行网络编程。读者可以根据书中的实例直接上手进行编程,亲自调试程序代码,这样在学习本书的过程中会受益更大。

本书由马展和李守勇共同执笔编写。此外,喻波、马天一、魏勇、郝荣福、孙明、李大字、武思宇、牟博超、李彬、付鹏程、高翔、张巧玲、李辉、李欣、柏宇、郭强、金春范、程梅、黄霆、钟华、高海峰、王建胜、张浩和邵蕴秋等同志在整理材料方面给予作者很大的帮助,在此,作者对他们表示衷心的感谢。

由于作者水平有限,书中难免有疏漏和不足之处,恳请广大读者批评指正。

作 者

读者意见反馈卡

亲爱的读者：

感谢您购买了本书，希望它能为您工作和学习带来帮助。为了今后能为您提供更优秀的图书，请您抽出宝贵的时间填写这份调查表，然后剪下寄到：北京清华大学出版社第五事业部(邮编 100084)；您也可以把意见反馈到 cwkbook@tup.tsinghua.edu.cn。邮购咨询电话：010-62770175/77 转 3505。我们将充分考虑您的意见和建议，并尽可能地给您满意的答复。谢谢！

本书名：_____

个人资料：_____

姓名：_____ 性别：男 女 出生年月(或年龄)：_____

文化程度：_____ 职业：_____ 通讯地址：_____

电话(或手机)：_____ 传真：_____ 电子信箱(E-mail)：_____

您是如何得知本书的：_____

- 别人推荐 出版社图书目录 网上信息 书店
杂志、报纸等的介绍(请指明)_____ 其他(请指明)_____

您从何处购得本书：书店 电脑商店 软件销售处 邮购 商场 其他

影响您购买本书的因素(可复选)：

- 封面封底 装帧设计 价格 内容提要、前言或目录 书评广告
出版社名声 作者名声 责任编辑
其他：_____

您对本书封面设计的满意度：很满意 比较满意 一般 较不满意 不满意 改进建议_____

您对本书印刷质量的满意度：很满意 比较满意 一般 较不满意 不满意 改进建议_____

您对本书的总体满意度：

- 从文字角度：很满意 比较满意 一般 较不满意 不满意
从技术角度：很满意 比较满意 一般 较不满意 不满意

本书最令您满意的是：

- 讲解浅显易懂 内容充实详尽 示例丰富到位 指导明确合理 其他：_____

您希望本书在哪些方面进行改进？_____

您希望增加什么系列或软件的图书：_____

您最希望学习的其他软件：1. _____ 2. _____ 3. _____ 4. _____

您对使用中文版软件或外文版软件介意吗？更喜欢使用哪一种版本？

- 介意 无所谓 中文版 外文版

您对图书所用软件版本是否很介意？是否要求用最新版本？

- 是，要求是最新版本 无所谓 不，因为硬件或软件跟不上要求

您是如何学习最新软件的？

- 看计算机书 看多媒体教学光盘 自己摸索或查看软件的帮助信息 参加培训班 向其他人请教
其他：_____

您的其他要求：_____

目 录

第 1 章 Visual C++ .NET 与网络通信编程概述	1
1.1 计算机网络的发展历程	1
1.1.1 计算机网络的形成	2
1.1.2 电路交换、报文交换和分组交换的比较	2
1.2 网络体系结构	4
1.2.1 OSI 分层协议	4
1.2.2 TCP/IP 模型	7
1.3 IP 层和传输层	10
1.3.1 IP 层数据报	10
1.3.2 TCP 数据报	12
1.3.3 UDP 数据报	14
1.4 TCP/IP 端口	15
1.5 Visual C++ .NET 编程应用概述	16
1.5.1 Visual C++ .NET 优点概述	16
1.5.2 Visual C++ .NET 对开发网络应用程序的支持	18
1.6 本章小结	24
第 2 章 获取网络配置信息	25
2.1 提出问题	25
2.1.1 需求描述	25
2.1.2 UML 建模分析	26
2.1.3 实例功能介绍和运行时的界面	27
2.1.4 NetBIOS 功能	30
2.2 设计方案	35
2.3 解决方案	36
2.3.1 框架资源的设计	36
2.3.2 实现代码的设计	42
2.4 本章小结	59
第 3 章 ICMP 高级编程	60
3.1 提出问题	60
3.1.1 需求描述	60

3.1.2	UML 建模分析	61
3.1.3	实例功能介绍和运行时的界面	65
3.1.4	ICMP 协议介绍	68
3.2	设计方案	71
3.3	解决方案	72
3.3.1	框架资源的设计	72
3.3.2	实现代码的设计	80
3.4	本章小结	96
第 4 章	Telnet 服务器/客户机	97
4.1	提出问题	97
4.1.1	需求描述	97
4.1.2	UML 建模分析	98
4.1.3	实例功能介绍和运行时的界面	99
4.1.4	Telnet 协议	102
4.2	设计方案	103
4.3	解决方案	105
4.3.1	资源框架的设计	105
4.3.2	实现代码的设计	115
4.4	本章小结	131
第 5 章	基于 C/S 模型的聊天室	132
5.1	提出问题	132
5.1.1	需求描述	132
5.1.2	UML 建模分析	133
5.1.3	实例功能介绍和运行时的界面	135
5.1.4	基于 Windows Sockets 应用开发介绍	138
5.2	设计方案	142
5.3	解决方案	143
5.3.1	资源框架的设计	144
5.3.2	实现代码的设计	151
5.4	本章小结	176
第 6 章	FTP 客户机	177
6.1	提出问题	177
6.1.1	需求描述	177
6.1.2	UML 建模分析	178
6.1.3	实例功能介绍和运行时的界面	179

6.1.4	FTP 命令控制字	182
6.2	设计方案	193
6.3	解决方案	194
6.3.1	框架资源的设计	194
6.3.2	实现代码的设计	199
6.4	本章小结	220
第 7 章	多线程下载	221
7.1	提出问题	221
7.1.1	需求描述	221
7.1.2	UML 建模分析	222
7.1.3	实例功能介绍和运行时的界面	223
7.1.4	WinInet 类的功能	225
7.1.5	创建和终止线程	228
7.2	设计方案	228
7.3	解决方案	229
7.3.1	框架资源的设计	230
7.3.2	实现代码的设计	232
7.4	本章小结	243
第 8 章	串口通信高级编程	244
8.1	提出问题	244
8.1.1	需求描述	244
8.1.2	UML 建模分析	245
8.1.3	实例功能介绍和运行时的界面	247
8.1.4	串口通信原理与多线程技术	247
8.2	设计方案	262
8.3	解决方案	263
8.3.1	框架资源的设计	263
8.3.2	实现代码的设计	265
8.4	本章小结	277
第 9 章	RAS 高级编程	278
9.1	提出问题	278
9.1.1	需求描述	278
9.1.2	UML 模型分析	279
9.1.3	实例功能介绍和运行时的界面	281
9.1.4	RAS 客户机和拨号连接建立介绍	283

9.1.5	RAS 函数说明	285
9.2	设计方案	288
9.3	解决方案	290
9.3.1	框架资源的设计	290
9.3.2	实现代码的设计	290
9.4	本章小结	314
第 10 章	语音电话高级编程	315
10.1	提出问题	315
10.1.1	需求描述	316
10.1.2	UML 建模分析	317
10.1.3	实例功能介绍和运行时的界面	317
10.1.4	Modem 原理	318
10.1.5	AT 命令	320
10.2	设计方案	325
10.3	解决方案	325
10.3.1	资源框架的设计	326
10.3.2	实现代码的设计	328
10.4	本章小结	338
第 11 章	Web 浏览器	339
11.1	提出问题	339
11.1.1	需求描述	339
11.1.2	UML 建模分析	340
11.1.3	实例功能介绍和运行时的界面	341
11.1.4	关键 CHtmlView 类介绍	343
11.2	设计方案	350
11.3	解决方案	351
11.3.1	框架资源的设计	351
11.3.2	实现代码的设计	355
11.4	本章小结	363
第 12 章	Web 服务器	364
12.1	提出问题	364
12.1.1	需求描述	364
12.1.2	UML 建模分析	365
12.1.3	实例功能介绍和运行时的界面	366
12.1.4	HTTP 协议	369

12.2	设计方案	373
12.3	解决方案	374
12.3.1	框架资源的设计	375
12.3.2	实现代码的设计	379
12.4	本章小结	394
第 13 章	Email 协议及高级编程	395
13.1	提出问题	395
13.1.1	需求描述	395
13.1.2	UML 模型分析	397
13.1.3	实例功能介绍和运行时的界面	399
13.1.4	电子邮件格式	401
13.1.5	SMTP 与 POP3 原理介绍	402
13.1.6	MD5 算法	410
13.2	设计方案	411
13.3	解决方案	414
13.3.1	框架资源的设计	414
13.3.2	实现代码的设计	414
13.4	本章小结	439

第1章 Visual C++ .NET与网络通信编程概述

从网络出现至今，互联网已经成为现代社会生活不可缺少的一部分，它深刻影响着这个社会的方方面面，改变了我们的生活方式。不会使用网络的人将没有办法和外界进行更加频繁地交流，也将无法得到最新的信息。很显然，“网络就是计算机”这个先前就提出来的概念将会更加深入人心。

网络上运行着各种千差万别的计算机，从安装 UNIX 高端服务器(企业 Sun 工作站)到安装 Microsoft Windows 系列的个人 PC，此外还有 Mac 系列，虽然它们运行的操作系统不一样，应用程序的设计上也存在着很大差别，但是由于网络协议的一致，保证在这样的千差万别下面有着同样的交互信息接口，这样也就保证了连接在网络上的每一台计算机都能够和网络中的其他计算机进行交互和资源共享。不难理解，只有在统一的标准下，所有的应用，包括程序和数据才会有实现共享的可能。

统一的标准下，实现的过程将是一件相对简单的事情，微软新近推出的针对未来网络通信的 Visual C++ .NET 环境提供了更加简便的接口和菜单操作，程序员将轻松地这样的环境中“创造天地”。

本章知识要点：

- 计算机网络的发展历程
- 网络体系结构
- IP 层和传输层
- TCP/IP 模型
- Visual C++ .NET 网络通信编程支持
- Sockets 编程

1.1 计算机网络的发展历程

计算机的发展经过了一个漫长的过程，从开始的终端主机方式，到后来 ARPANet 的形成，这时期经历大概 20 年的时间。ARPANet 的出现为网络的高速发展奠定了基础。近 20 多年来，网络接入用户数以指数的形式在增长，这些为现代网络的发展带来了很多的机遇，也带来了很大的挑战。

在网络发展的过程中，交换技术是网络发展的关键技术，没有好的交换技术，网络将

不能满足很多方面的工作。现代高要求下的网络和通信，交换技术也在进一步的高速发展和改进以适应服务的需要。下面首先向读者介绍交换网络发展的历程，从这个过程中，读者可以了解到网络的形成，以及网络交换等方面的基本知识，建立网络通信的基本概念。

1.1.1 计算机网络的形成

20 世纪 50 年代，简单的“终端-通信线路-计算机”系统，构成了最初计算机网络的雏形，被称为第一代计算机网络。信息交换的方式以电路交换为主。即计算机通过多重线路控制器与远程终端相连，这种连接方式如图 1-1 所示。

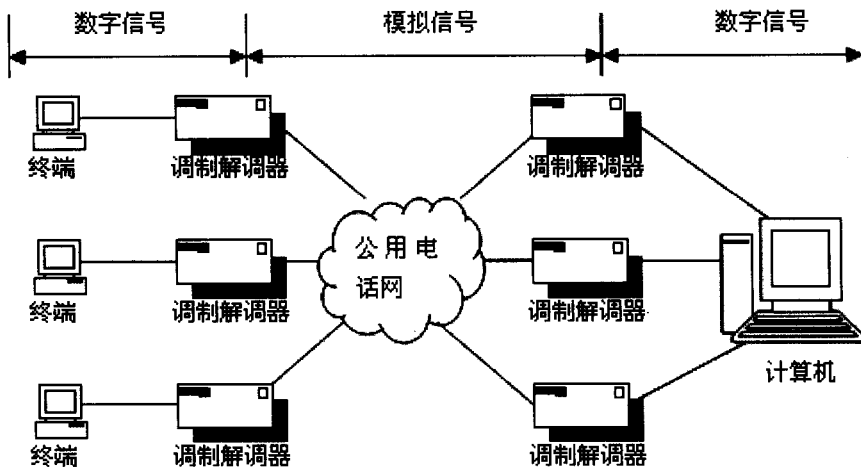


图 1-1 多重线路控制器与远程终端连接图

20 世纪 60 年代，出现了计算机与计算机之间的直接通信。此时，传统的电路交换方式(Circuit Switch)对于不同类型、不同规格、不同速率的计算机终端很难实现互相通信，而且线路使用率低，开销很大。

在此背景下出现了分组交换(Packet Switch)技术。1969 年 12 月美国的分组交换网 ARPANet 投入运行。现在大家公认 ARPANet 是分组交换网之父，并将分组交换网的出现作为现代电信时代的开始。它标志着第二代计算机网络的产生。此后，ARPANet 获得了迅猛的发展，并成为日后国际互联网 Internet 的基础。

1.1.2 电路交换、报文交换和分组交换的比较

在计算机和通信相互结合的过程中，交换方式也随着实际的需求不断地修改和完善。每一个时期的交换方式都有着自己的特点和关键技术，下面就电路交换、报文交换、分组交换进行比较，从中体会每一种交换技术的优缺点。

(1) 电路交换(Circuit Switch): 在通话之前，通过用户的呼叫，由网络预先给用户分配

传输带宽。用户若呼叫成功，则从主叫端到被叫端就建立了一条物理通路。此后双方才能互相通话。通话完毕挂机后即自动释放这条物理链路。

电路交换的关键点就是：在通话的全部时间内用户始终占用端到端的固定传输带宽。电路交换的示意图如图 1-2 所示。

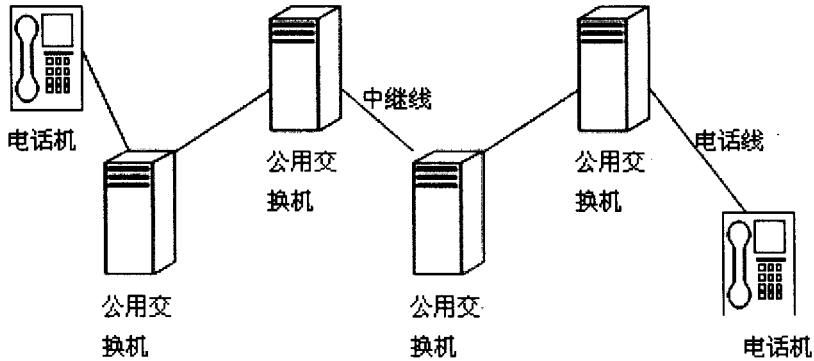


图 1-2 电路交换示意图

(2) 报文交换(Speech Switch)：它基于存储转发的原理。在报文交换中心，一份份电报被接收下来，并穿成纸带。操作员以每份报文为单位，撕下纸带，根据报文的目的地地址，拿到相应的发报机转发出去。这种报文交换的时延较长，从几分钟到几小时不等。

(3) 分组交换(Packet Switch)：通常我们将欲发送的整块数据称为一个报文。在发送报文之前，先将较长的报文划分为一个个更小的等长数据段，例如，每个数据段为 1024 位。在每一个数据段前面，加上报头，就构成了一个分组。分组又称为“包”，而分组的首部也可称为“包头”。分组中的首部是非常重要的，因为正是分组的首部才包含了诸如目的地址和源地址等重要控制信息，而分组交换网只有从分组的首部才能获知应将此分组发往何处。

图 1-3 示意了电路交换、报文交换和分组交换的主要区别。图中的 A 是源结点、D 是目的结点(宿结点)，而 B 和 C 是在 A 和 D 之间的中间结点。图 1-3(c)中 P1、P2、P3 分别代表报文的分组。

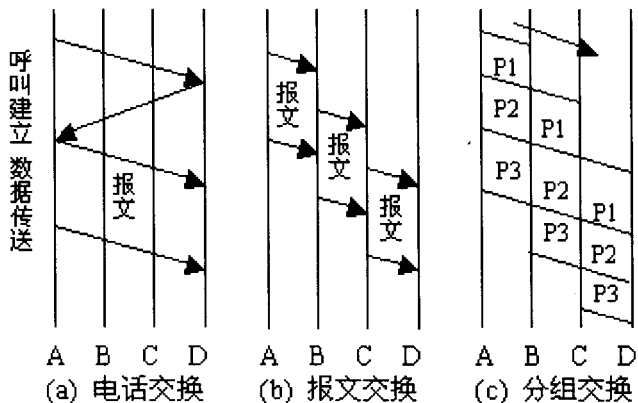


图 1-3 三种交换技术的比较

1.2 网络体系结构

网络要具有很高的可靠性，分层工作是基本要求。网络是一个复杂的“联合运作体”，只有分配好各个部分的工作，各个部分协调工作，及时处理，才能建立一个稳健的网络。而这里所涉及的网络的各个部分，在狭义上来说，即为网络的“层”(layer)。由于网络本身结构的复杂，加上不同的网络分布区域受到的外界因素影响形形色色，所以将网络分成各个层次，分而治之是最好的选择。这样的情况下，既能降低网络处理问题的复杂程度，而且能够更加有效。能够在第一层网络完成的操作，就不再需要传到第二层，提高了整个网络的效率。

基于这些简单的概念和介绍，下面将介绍网络的分层结构。

1.2.1 OSI 分层协议

一个计算机网络有许多互相连接的结点，在这些结点之间要不断地进行数据的交换。要做到有条不紊地交换数据，每个结点就必须遵守一些事先约定好的规则。这些为进行网络中的数据交换而建立的规则、标准或约定称为网络协议。主要由以下 3 个要素组成：

- (1) 语法，即数据与控制信息的结构或格式。
- (2) 语义，即需要发出何种控制信息，完成何种动作以及做出何种应答。
- (3) 同步，即事件实现顺序的详细说明。

ARPANet 的研制经验表明，对于非常复杂的计算机网络协议，最好采用层次式结构。其优点有如下 6 个方面。

(1) 层与层之间是独立的。某一层并不需要知道它的下一层如何实现的，而仅仅需要该层的通信协议即可。

(2) 对应层通过该层间的接口所提供的服务。由于每一层只实现一种相对独立的功能，因而可将一个难以处理的复杂问题分解为若干个较容易处理的更小一些的问题。这样，整个问题的复杂程度就下降了。

(3) 当任何一层发生变化时(例如由于技术的变化)，只要层间接口关系保持不变，则在这层以上或以下各层均不受影响。此外，对某一层提供的服务还可进行修改。当某层提供的服务不再需要时，甚至可以将这层取消。

(4) 各层都可以采用最合适的技术来实现，那么结构上各层是可分割开的。

(5) 分层结构使得实现和调试一个庞大而又复杂的系统变得易于处理，因为整个的系统已被分解为若干个相对独立的子系统。

(6) 每一层的功能及其提供的服务都已有了精确的说明，能促进网络标准化工作。

分层协议存在上述优点，那么只要将所有的网络按照一定的结构分层就可以解决所有的问题了。这时读者可能会有所误解，因为对于网络任务来说，分层越多，那么分解的子

任务就越多, 解决起来也就越简单。实际情况并不是这样的, 分层太多必定造成网络建设、维护费用等额外的开销。分层在存在优点的同时也存在着不可避免的缺点: 分层时应使每一层的功能非常明确。若层次太少, 就会使每一层的协议太复杂。但层数太多又会在描述和综合各层功能的系统项目任务时遇到较多的困难。

我们将计算机网络的各层及其协议的集合, 称为网络的体系结构。国际标准化组织(ISO)为了实现不同计算机网络的互相通信, 制定了一整套计算机网络的标准, 即“开放系统互连”(简称 OSI)。所谓“开放”, 就是指只要遵循 OSI 标准, 一个系统就可以和位于世界上任何地方的, 也遵循这一标准的系统进行通信。因此, 开放系统互连可使世界范围内的应用进程能开放式(而不是封闭式)地进行信息交换。1983 年, ISO 公布了开放系统互连基本参考模型的正式文件。通常人们也将它称为 OSI 参考模型, 或简称 OSI/RM, 如图 1-4 所示。

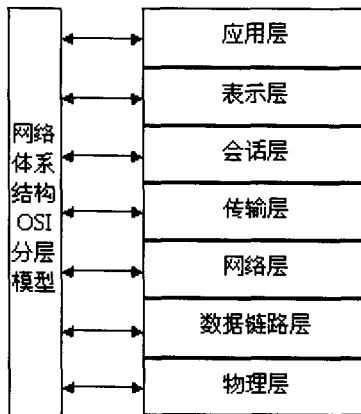


图 1-4 OSI 参考模型

OSI 参考模型具有 7 层的体系结构, 各层以及各层的功能描述分别如下。

(1) 物理层(Physical Layer)

物理层的任务是为它的上一层(数据链路层)提供一个物理连接, 以便透明地传送比特流。“透明地传送比特流”表示经实际电路传送后的比特流没有发生变化, 这种传输与数据通信的物理或电气特性有关。物理层以比特流的方式传送来自数据链路层的数据, 而不去理会数据的含义或者格式。同样, 它接收数据后, 不加分析, 直接传给数据链路层。

(2) 数据链路层(Data Link Layer)

数据链路层负责监督相邻网络节点的信息流动。它使用差错或纠错技术来确保正确的传输。当数据链路检测到错误时, 它请求重发, 或是根据情况纠正。数据链路层还要解决流量控制的问题: 流量太大, 网络会出现阻塞; 太小了, 又会使发送方和接收方等待时间过长。

另外, 数据链路层还管理数据格式。数据通常被组合成帧加以传输。帧是按某种特定格式组织起来的字节集合。数据链路层用惟一的比特组合对将要发送的每一帧的开始和结束进行表示, 对接收进来的每一帧进行判断, 然后把无错的帧送往上一层, 即网络层。

(3) 网络层(Network Layer)

网络层传送的数据是分组或包。它的任务就是要选择合适的路由和交换节点，使发送站的传输层所传下来的分组能够正确无误地按照地址找到目的站，并交付给目的站的传输层。这就是网络层寻址功能。它的另一个功能是防止网络拥塞。同时，网络层控制着通信子网(Communication Subnet)。所谓通信子网就是实现路由和数据传输所必需的传输媒体和交换组件的集合。网络层是针对子网的最高层次。这一层也可能包含计费软件。其费用取决于传输数据的数量，也可能和使用网络的时段有关。网络层负责管理这些信息，并负责计费。

(4) 传输层(Transport Layer)

传输层是处理端对端通信的最低层(更低层处理网络本身)。传输层负责选择通信使用的网络。一台计算机可能连接着多个网络，其速度、费用和通信类型各不相同。到底选择哪一个取决于很多因素。比如，传输的信息是很长的连续数据流，还是分为多次间歇传送。电话网络适用于前者。一旦建立连接，其线路将一直保持到传输完毕。

传输层另外一种传输方法是把信息划分成多个小的分组(数据子集)。当报文较长时，先分成几个分组，然后再交给下一层(网络层)进行传输，而两个子网之间不需要稳定的连接。每一分组通过网络被独立地传输，因而它可根据通信子网的特征最佳地利用网络资源，并以可靠和经济的方式，在两个端系统(源站和目的站)的会话层之间，建立一条运输连接，以透明地传送报文。或者说，传输层为上一层(会话层)提供一个可靠的端到端的服务。

(5) 会话层(Session Layer)

会话层允许不同主机上的应用程序进行会话，或建立虚连接。但是它不参与具体的数据传输，只处理一些差错恢复和进行管理，在两个互相通信的应用进程之间建立、组织和协调其交互。在会话层及以上更高层次中，数据传送的单位一般都称为报文。

(6) 表示层(Presentation Layer)

表示层主要解决用户信息的语法表示问题，即以用户可理解的格式为上层用户提供必要的信息。表示层将欲交换的数据从适合某一用户的抽象语法变换为适合于 ISO 系统内部使用的传送语法。有了这样的表示层，用户就可以把精力集中在其所要交流的问题本身，而不必更多地考虑对方的数据格式。此外，对传送的信息加密/解密也是表示层的任务之一。在把数据交给其下层进行传送前，它可以先对数据进行加密。另一端的表示层负责在接收到数据后解密。对用户来说，根本不知道数据曾经改变过。对于广域网来说，这一技术特别重要。

(7) 应用层(Application Layer)

应用层是 OSI 参考模型中的最高层，直接与用户和应用程序打交道，用来确定进程之间通信的性质，以满足用户的需要。其次就是负责用户信息的语义表示，并在两个通信者之间进行语义匹配。这就是说，应用层不仅要提供应用进程所需要的信息交换和远地操作，而且还要作为互相作用的应用进程的用户代理，来完成一些信息交换所必须的功能。另外，应用层也定义了一些协议集，以支持通过全屏幕文字编辑器方式模拟各种不同类型的终端。

上面介绍了 OSI 网络模型，在网络的发展过程中，还有一个模型起到了很大的作用，

就是下面我们要介绍的 TCP/IP 模型。

1.2.2 TCP/IP 模型

TCP/IP 协议实际上是一个协议族，TCP 和 IP 是最著名的两个协议，其他协议包括用户数据报协议(UDP)、互联网控制报文协议(ICMP)以及地址解析协议(ARP)等，整个协议族称为 TCP/IP。

1. TCP/IP 历史渊源

TCP/IP 的起源可以追溯到 1969 年，当时它是作为一个关于网络互联的分组交换研究项目而开始的，该网络被称为 ARPANet，是由美国国防部的国防高级研究项目署创建的。ARPANet 在位于华盛顿特区的国家科学基金会的帮助下成长为 Internet。

ARPANet 开始使用的是一种称为网络控制协议(NCP)的协议。随着 Internet 的发展，需要更复杂的协议。1973 年，引进了传输控制协议(TCP)，接着，在 1981 年引进了网际协议(IP)。1982 年，TCP 和 IP 被标准化成为 TCP/IP 协议簇，并在 1983 年，取代了 ARPANET 上的 NCP。

1983 年，自由的电子通信和信息共享与其他一些内容被加入了广为接受的 TCP/IP，使其成为大学和政府部门的标准。TCP/IP 作为一个标准组件被包含到柏克利标准发行中心 UNIX 的实现中，从那时起，TCP/IP 就与 UNIX 操作系统关系密切了，最近几年，用户促使供应商也把 TCP/IP 加入其他操作系统中，现在，所有的每个计算机平台都支持 TCP/IP 协议。这也就使得不同的操作系统可以在网络上进行交互。

2. TCP/IP 模型

由于 TCP/IP 协议模型把该协议分成 4 个概念层，所以它与 OSI 模型略有不同，如图 1-5 所示。

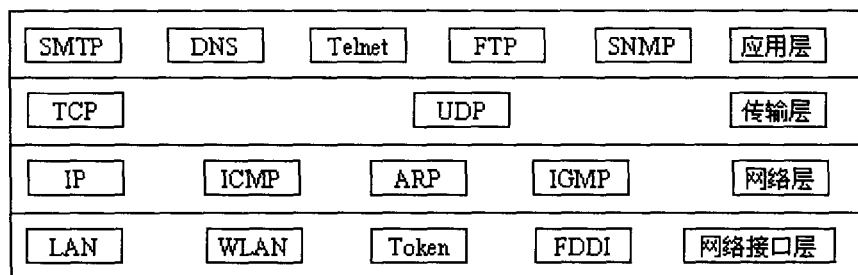


图 1-5 TCP/IP 协议 4 层通信体系结构

TCP/IP 模型比起 OSI 模型，简化了一些层次，使用网络接口层来屏蔽 OSI 模型中的下层操作。下面介绍各个层的功能和作用。

(1) 网络接口层

网络接口层是 TCP/IP 协议模型的最低层，它包括那些能使 TCP/IP 与物理网络进行通