

乔林 杨志刚 黄建东 魏志强 编著

Internet 应用实务篇

程序设计

Delphi 5.0



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Delphi 5.0 程序设计

——Internet 应用实务篇

乔林 杨志刚 黄建东 魏志强 编著



中国铁道出版社

2000·北京

(京)新登字 063 号

内 容 简 介

本书着重讨论如何使用 Delphi5.0 进行 Internet 编程。我们将注意力集中在如何使用 Delphi5.0 的部件进行客户端编程上,这部分内容是我们 Internet 上冲浪时会频繁遇到的。

作为本书的结尾,我们专辟一章详细讨论如何使用 Delphi5.0 开发 Web 服务器应用程序。

本书面向 Delphi5.0 的中、高级读者。

图书在版编目(CIP)数据

Delphi5.0 程序设计. Internet 应用实务篇/乔林等编著. —北京:中国铁道出版社, 2000.1

ISBN 7-113-03589-2

I. D… II. 乔… III. ①Delphi 语言-程序设计 ②因特网-程序设计 IV TP312

中国版本图书馆 CIP 数据核字(1999)第 55414 号

书 名: Delphi5.0 程序设计——Internet 应用实务篇

作 者: 乔林 杨志刚 黄建东 魏志强

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓丹

责任编辑: 苏 茜

特邀编辑: 曹宝惠

封面设计: 新创工作室 冯龙彬

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张: 26 字数: 624 千

版 本: 2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN7-113-03589-2/TP·416

定 价: 41.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社发行部调换。

前 言

Delphi 5.0 一个功能强大的可视化编程环境，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的可重用的面向对象 Pascal 程序语言。对于广大的程序开发人员来讲，使用 Delphi 5.0 开发应用软件和数据库应用程序，无疑会大大地提高编程效率，而且随着应用的深入，我们将会发现编程不再是枯燥无味的工作。

Delphi 5.0 的程序首先是一个应用程序框架，而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西，也仍然可以严格地按照设计运行。程序员的工作只是在“骨架”中加入自己的程序。缺省的应用程序是一个空白的窗体 (Form)，我们可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以放大缩小、移动、最大化和最小化等，但我们却没有编写一行程序。因此，可以说应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。Delphi 5.0 已经为读者做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，只是不处理任何事情。我们所需要做的，只是在程序中加入完成所需功能的代码而已。Delphi 5.0 把 Windows 编程的回调、句柄处理等繁复过程都放在一个不可见的覆盖物下面，这样读者可以不为它们所困扰，轻松从容地对可视化部件进行编程。

本书共分四个专题，涵盖了 Delphi 5.0 编程的方方面面：

《Delphi5.0 程序设计——基础教学篇》：该书面向 Delphi5.0 的初、中级读者。全书分 3 个部分介绍了 Delphi5.0 的基础知识和基本的编程方法。前 3 章介绍了 Delphi5.0 的新特性、集成开发环境和各种辅助工具的用法以及 Delphi 程序可视化开发方法；第 4、5、6 章介绍了对象 Pascal 语言的基础知识和各种语法实体的用法；第 7、8、9、10 章为编程实践部分，介绍了菜单的设计方法、文本编辑器、文件管理和图形图像应用程序的设计方法，这 4 章中结合实例程序的设计，介绍了 VCL 中菜单、文件、对话框、工具条、图形图像等众多组件的用法。全书由浅入深，循序渐进。在各部分内容的介绍中均配备了实例程序，力求通过这些实例加深理解，掌握编程方法和编程技巧。

《Delphi5.0 程序设计——数据库应用实务篇》：该书面向 Delphi 数据库程序设计编程人员。书中详细介绍了数据库访问部件、数据库控制部件、Delphi 环境中的 SQL 编程、客户机/服务器环境的构造和编程方法、数据仓库技术、InterBase 数据存取部件、综合数据报表制作、Delphi5.0 与 ADO 编程、DCOM、MTS 和 CORBA 等分布式数据库应用程序开发、多层数据库应用程序开发、Delphi 与 Word97、Excel97 等 Office 组件的集成等技术。同时，本书提供了通用数据查询、图形化数据报表分析等实用化范例。本书最后，详细地分析了一个集成化财务管理系统并提供了核心模块的全部源代码，该财务系统已经开始应用于商务环境中。

《Delphi5.0 程序设计——高级编程篇》：该书面向 Delphi5.0 的高级读者。Delphi 这种可视化的 RAD 工具，其核心思想可以认为是基于组件的。步入 Delphi 的世界并不难，但是要在里面建设更为高级的应用却要仔细理会 Delphi 的真谛。本书中详细介绍了组件的高级

用法、自定义组件、DLL 和 ActiveX 控件，以及 COM 应用；本书还对顺应 Windows 98、Windows NT 的多线程技术、在 Delphi 中的汇编嵌入问题进行了介绍。在本书的最后还系统地讲解了 Delphi 中应用程序帮助系统和安装问题。

《Delphi5.0 程序设计——Internet 应用实务篇》：本书讨论如何使用 Delphi 5.0 进行 Internet 编程。我们将注意力主要集中在如何使用 Delphi 5.0 的部件进行客户端编程上，这部分内容是在 Internet 上冲浪时会频繁遇到的。作为本书的结尾，我们专辟一章详细讨论如何使用 Delphi 5.0 开发 Web 服务器应用程序。该书面向 Delphi5.0 的中、高级读者。

本套丛书由杨志刚、魏志强、王忠华、乔林组织策划。《基础教学篇》由杨志刚、何志成、魏志强、王忠华主编，乔林、何敏、谢军、段军、刘文杰、张维平等参编；《数据库应用实务篇》由魏志强、王忠华、何敏、乔林主编，杨志刚、房桂英、魏青、张积中、黄建东、陈爱华等参编；《高级编程篇》由王忠华、魏志强、杨志刚、乔林主编，高群、黄建东、姜涛、林杜、费广正等参编；《Internet 应用实务篇》由乔林、杨志刚、黄建东、魏志强主编，王忠华、何隽、林杜、费广正、金传恩、王诚铭等参编。杨志刚、乔林审阅了全套丛书的内容。限于时间仓促，作者水平有限，书中如有错误，恳请读者谅解。

“边学边做”是学习一种编程工具的最有效方法，我们希望读者在阅读本丛书的过程中能够上机实践。每学完一个例子，尝试着改变一点点，或者添加一点东西，并改变一些代码，这将会使读者体验到进步和成功的乐趣。

编者

1999 年 10 月

目 录

第 1 章 计算机网络体系结构	1
1.1 计算机网络与通信的基本概念	1
1.1.1 计算机通信	1
1.1.2 网络的拓扑结构	2
1.1.3 数据冲突的解决	2
1.1.4 网络连接设备	3
1.1.5 网络结构	4
1.2 网络协议	4
1.2.1 ISO/OSI 参考模型	4
1.2.2 网络协议	4
1.3 网络通信服务	5
1.3.1 网络协议与服务模式	5
1.3.2 差错控制	6
1.4 网络分层	7
1.4.1 七个网络层次	7
1.4.2 物理层	8
1.4.3 数据链路层	9
1.4.4 网络层	9
1.4.5 传输层	10
1.4.6 会话层	11
1.4.7 表示层	11
1.4.8 应用层	11
1.5 客户/服务器模型	12
1.5.1 客户/服务器模型	12
1.5.2 虚电路与套接字	12
小结	13
第 2 章 TCP/IP 协议组	14
2.1 TCP/IP 协议	14
2.1.1 TCP/IP 协议栈	14
2.1.2 TCP/IP 数据流	14
2.1.3 TCP/IP 网络模型	15
2.2 TCP/IP 网络分层	16
2.2.1 物理层	16

2.2.2 链路层.....	16
2.2.3 网络层.....	17
2.3 Internet 地址.....	17
2.3.1 Internet 地址.....	17
2.3.2 Internet 地址类.....	17
2.3.3 Internet 地址协议.....	18
2.4 IP 数据报.....	19
2.4.1 IP 数据包.....	19
2.4.2 IP 报头.....	19
2.4.3 数据报的分割.....	21
2.4.4 数据报的重组.....	22
2.5 IP 路由选择.....	22
2.6 传输层.....	23
2.7 TCP 协议.....	23
2.7.1 TCP 的可靠性.....	23
2.7.2 滑动窗口技术.....	24
2.7.3 TCP 报文.....	24
2.7.4 建立 TCP 连接.....	26
2.7.5 关闭 TCP 连接.....	27
2.8 Delphi 5.0 的网络部件.....	27
小结.....	29
第3章 Windows Sockets 编程.....	30
3.1 Sockets 工作原理.....	30
3.2 Sockets 类与 Sockets 部件.....	30
3.2.1 TAbstractSocket 类.....	30
3.2.2 TCustomSocket 类.....	36
3.2.3 TClientSocket 部件.....	38
3.2.4 TCustomServerSocket 类.....	40
3.2.5 TServerSocket 部件.....	47
3.2.6 TCustomWinSocket 类.....	48
3.2.7 TClientWinSocket 类.....	56
3.2.8 TServerWinSocket 类.....	57
3.2.9 TServerClientWinSocket 类.....	58
3.3 Sockets 编程实例.....	59
3.3.1 窗体设计.....	59
3.3.2 源程序代码.....	63
3.3.3 源程序分析.....	68
小结.....	69

第4章 阅读HTML文档	70
4.1 HTML语言概要	70
4.1.1 SGML的组成	70
4.1.2 HTML元素	70
4.1.3 HTML注释	71
4.1.4 HTML文档的显示	71
4.1.5 HTML文档字符集	72
4.1.6 HTML字符实体	72
4.1.7 HTML数据类型	72
4.1.8 HTML文档结构	73
4.1.9 一个完整的HTML文档实例	74
4.2 THTML控件	77
4.3 HTML编程实例	82
4.3.1 窗体设计	83
4.3.2 源程序代码	88
4.3.3 源程序分析	92
小结	94
第5章 HTTP编程	95
5.1 HTTP协议概要	95
5.1.1 HTTP协议的基本特点	95
5.1.2 HTTP的工作原理	96
5.1.3 HTTP的请求与响应	97
5.1.4 HTTP头标	100
5.1.5 一般头标字段	100
5.1.6 请求头标字段	100
5.1.7 响应头标字段	101
5.1.8 实体头标字段	101
5.2 TNMHTTP部件	102
5.2.1 属性	102
5.2.2 方法	108
5.2.3 事件	112
5.2.4 任务	115
5.3 HTTP编程实例	115
5.3.1 窗体设计	115
5.3.2 源程序代码	129
5.3.3 源程序分析	137
小结	137

第 6 章 使用 ActiveX 控件.....	138
6.1 引入类型库与 ActiveX 控件.....	138
6.2 接口的基本概念.....	143
6.3 使用 TWebBrowser.....	149
6.3.1 窗体设计.....	149
6.3.2 源程序代码.....	171
6.3.3 源程序分析.....	186
小结.....	188
第 7 章 FTP 编程.....	189
7.1 FTP 协议的基本概念.....	189
7.1.1 FTP 会话.....	189
7.1.2 FTP 模型.....	190
7.1.3 FTP 文件类型.....	190
7.1.4 FTP 传输模式.....	191
7.1.5 FTP 应答.....	191
7.2 TNMFTP 部件.....	193
7.2.1 属 性.....	193
7.2.2 方 法.....	199
7.2.3 事 件.....	204
7.2.4 任 务.....	207
7.3 FTP 编程实例.....	208
7.3.1 窗体设计.....	208
7.3.2 源程序代码.....	222
7.3.3 源程序分析.....	234
小结.....	236
第 8 章 报文发送与接收.....	237
8.1 TNMMsg 部件.....	237
8.1.1 属 性.....	237
8.1.2 方 法.....	239
8.1.3 事 件.....	240
8.2 TNMMsgServ 部件.....	241
8.2.1 属 性.....	241
8.2.2 事 件.....	242
8.3 TNMMsg 与 TNMMsgServ 编程实例.....	243
8.3.1 窗体设计.....	243
8.3.2 源程序代码.....	247
8.3.3 对源程序的进一步改进.....	250

小结.....	254
第 9 章 收发电子邮件.....	255
9.1 MIME.....	255
9.1.1 MIME 的基本特征.....	255
9.1.2 MIME 类型.....	256
9.1.3 MIME 头标.....	257
9.1.4 MIME 的用户代理.....	257
9.2 TNMSMTP 部件.....	258
9.2.1 属 性.....	258
9.2.2 方 法.....	263
9.2.3 事 件.....	264
9.3 SMTP 编程实例.....	267
9.3.1 窗体设计.....	267
9.3.2 源程序代码.....	277
9.4 TNMPOP3 部件.....	283
9.4.1 属 性.....	284
9.4.2 方 法.....	286
9.4.3 事 件.....	286
9.5 TNMPOP3 编程实例.....	286
9.5.1 窗体设计.....	286
9.5.2 源程序代码.....	295
小结.....	302
第 10 章 其他网络技术.....	303
10.1 文件流.....	303
10.1.1 TNMStrm 部件和 TNMStrmServ 部件.....	304
10.1.2 窗体设计.....	304
10.1.3 源程序代码.....	308
10.1.4 源程序分析.....	312
10.2 报文编码与解码.....	314
10.2.1 TNMUUProcessor 部件.....	314
10.2.2 窗体设计.....	314
10.2.3 源程序代码.....	318
10.2.4 源程序分析.....	321
10.3 用户数据报协议.....	322
10.3.1 TNMUDP 部件.....	322
10.3.2 窗体设计.....	322
10.3.3 源程序代码.....	326

小结	329
----------	-----

第 11 章 Web 服务器应用程序编程	330
----------------------------	-----

11.1 创建 Web 服务器应用程序	330
11.1.1 术语与标准	330
11.1.2 URL 与 URI	330
11.1.3 HTTP 请求头标信息	331
11.1.4 HTTP 服务器活动	331
11.1.5 形成客户请求	331
11.1.6 服务客户请求	332
11.1.7 应答客户请求	332
11.2 Web 服务器应用程序向导	333
11.2.1 CGI 与 Win-CGI	333
11.2.2 ISAPI/NSAPI	335
11.3 TWebModule 部件	337
11.3.1 TWebModule 类	337
11.3.2 TWebRequest 类	339
11.3.3 TWebResponse 类	344
11.4 TWebApplication 类	349
11.5 TWebDispatcher 部件	351
11.5.1 TWebDispatcher 部件	351
11.5.2 TWebActionItems 类与 TWebActionItem 类	353
11.5.3 设置活动项的属性	355
11.5.4 使用活动项响应请求报文	356
11.5.5 调度请求报文	357
11.5.6 一个实际的例子	358
11.6 调试 Web 服务器应用程序	358
11.6.1 设置 ISAPI/NSAPI DLL 调试参数	359
11.6.2 设置 CGI/Win-CGI EXE 调试参数	361
11.6.3 调试 WebModule	361
11.7 生成 Web 页	363
11.7.1 访问客户请求信息	364
11.7.2 创建应答报文	365
11.7.3 HTML 模板	366
11.7.4 TPageProducer 部件	367
11.8 数据发布	370
11.8.1 TDataSetPageProducer 部件	370
11.8.2 数据发布	371
11.8.3 源程序代码	374

第 1 章 计算机网络体系结构



计算机网络是一台或多台计算机互相连接进行通信的基本形式。计算机网络的通信原理从本质上说与网络的大小无关。计算机网络通常分为两大范畴：局域网和广域网。局域网（LAN）将距离较近的计算机连在一起，而广域网（WAN）则将距离较远的计算机连在一起。因此，我们可以将广域网看作是远距离网络，因为网络交换的信息常常要传输很远的距离。虽然 Internet 包含了全世界范围的很多网络，但作为一名 Internet 程序员，我们可以将 Internet 看作单个网络。

1.1 计算机网络与通信的基本概念

如果将两台或者更多的计算机连接在一起，使它们能够互相通信，则我们就建立了一个计算机网络，而如果将两个或更多的网络连接在一起，则创建的计算机网络就是互连网（internet）。

Internet（以大写字母“I”开头）是全世界范围内现有的最大最通用的互连网，它连接了 100 多个国家的大量的计算机网络。虽然这些网络使用不同的软件，但当我们使用 Internet 网或者进行 Internet 编程时，几乎感觉不到这些差别。这是为什么呢？

1.1.1 计算机通信

我们知道，电信号有数字和模拟两种形式。因为计算机使用数字信号表示二进制的 0 和 1 很方便，因此常常使用数字信号在内部表示数据。然而当通过网络传输数据时，计算机却常常采用模拟信号。一般地，我们可以将模拟信号看作波。在固定时间间隔（如 1 秒）内波的重复数称为波的频率。我们可以使用频率调制（FM）的方法改变模拟波的频率。

典型的过程是这样的，计算机使用调制解调器将二进制数转换为模拟信号，然后在电话线上传输。调制解调器使用频率调制表示二进制数据。调制解调器通过用 1 表示频率高的信号，用 0 表示频率低的信号来发送二进制数据。调制解调器对模拟信号的频率进行调制。当另一个调制解调器收到这个信号时，它对模拟信号进行解调，将模拟信号变为计算机理解 0 和 1 串。

网络通常使用三种不同的串行方式进行通信——单工、半双工或全双工通信。单工通信时，数据只能在一个方向上串行传输；半双工通信时，数据可以在两个方向上传输，但每次只能在一个方向上传输；全双工通信数据则可以同时在两个方向上传输。

为了完成数据传输，网络使用通信交换技术。通信交换技术允许硬件设备共享物理通信线路。许多计算机网络，包括 Internet 都使用分组交换。在分组交换网络中，程序将数据分割为称为包（PACKETS）的小块，在计算机之间进行传输。一个分组交换网络包含许多

计算机之间的多条通道。数据可以在两个方向传输。因此，每个包必须携带它的目的地址。

1.1.2 网络的拓扑结构

计算机网络的拓扑结构指网络的形状或者计算机的实际连接方法。网络拓扑结构实际上为我们提供了一种网络比较和分类的方法。

在众多的通用网络拓扑结构中，星形、环形和总线形是最常见的。星形拓扑结构中的所有计算机（节点）都连到中心计算机或集线器上。除了与集线器或与中心计算机相连外，任何计算机都不与其它计算机相连。星形网的所有数据包都必须先发送到中心集线器。中心集线器再将所有数据包传输到目的地。

对于星形网而言，一台计算机与集线器的通道发生故障不会影响网络中的其它计算机；然而，如果集线器损坏则整个网络将会崩溃。

环状拓扑结构中的所有计算机都连接在一起，形成一个环。从网络的任何一点，都可以沿同一方向传输数据，最后返回到起始点。由于是环形连接，故其中的数据在网中只能沿一个方向传输。

和星形结构不同，环形拓扑结构要求网络计算机之间的通道不能发生故障。环上任何一点发生故障都会使网络通信中止。此外，环形拓扑结构的另一个缺点是数据必须经过网络中的每一台计算机，因此，网络中的任何一台计算机都可以察看环中传输的数据。

总线拓扑结构使用一条总线作为传输媒体。网络中的所有计算机都直接连到总线上。同轴电缆常常在总线拓扑结构中当作传输媒体。总线拓扑结构中发送数据可以在任何一个方向传输，它要求总线的两端连到终结器上。与环形网一样，总线上任何一点发生故障都会使网络中的通信失败。由于数据经过总线上的每一台计算机，跟环形网一样，总线形网的安全性也不好。

1.1.3 数据冲突的解决

不管采用什么样的网络拓扑结构，当两台计算机在同一时间传输数据时，都可能发生数据冲突。计算机使用总线仲裁技术来确定由哪一台计算机使用总线以及什么时候使用总线。

冲突检测和令牌传输是总线仲裁最常见的两种方法。如果计算机使用冲突检测进行仲裁，则必须在传输前对总线进行侦听。如果某台计算机发现其他计算机正使用总线传输数据，则必须等到使用总线的计算机传输完毕后才能发送数据。

如果两台计算机同时传输数据，冲突检测系统要求计算机在发送数据时对总线进行侦听。如果某台计算机发现在总线上侦听的数据不属于自己，则停止发送数据，等待一个随机时间后重新开始。

如果计算机在传输数据前对网络活动进行侦听，则称该计算机使用了所谓的载波侦听冲突避免技术，此时计算机对总线进行侦听以查看其它计算机是否正在传输数据。如果计算机在传输数据时侦听其它活动，则称该计算机使用了所谓的冲突检测技术。

令牌传输与冲突检测不同，系统要求计算机在传输数据之前必须获得允许。为了获得

发送权,计算机要获取一个称为令牌的特殊数据包(我们可以将令牌看作一个允许发送标志,它通知计算机可以发送数据)。所有的令牌传输系统都必须能够检测和重新生成令牌。否则的话,如果令牌丢失(而这是有可能的),网络系统就会终止。如果计算机需要传输数据,则必须获取并保持令牌;在数据传输完成后,该计算机必须重新发送令牌。此时,令牌将继续在网络中传递。最终网络中的任何一台计算机都有机会获得令牌以进行数据传输。

1.1.4 网络连接设备

事实上,互连网并不采用某种特定形式的拓扑结构,它通常使用中继器、桥接器、路由器、网关这样一些特殊设备将不同形状不同技术的独立网络连接在一起。

网络传输线中的阻抗和噪声都会减弱模拟波信号,因此,网络使用中继器来解决信号的衰减问题。中继器对收到的信号进行复制,并在重新发送之前对该信号进行放大,即增加波的振幅(不改变频率)。增加波振幅减少了传到中继器波形信号的损失和衰减。

在网络中增加中继器可以延长相邻计算机间的距离。长距离网络包含许多中继器。以太网经常采用中继器增加局域网中总线电缆的长度。

桥接器主要用于将使用相同技术的两个网络(如以太网)连接在一起。桥接器比中继器更复杂。桥接器检查任何一个网络的数据总线上的数据包。如果桥接器发现一个网络的数据包要传送到另一个网络的某台计算机,则完成跨网络的数据包传输工作。如果包的目的地地址还在同一个网络上,则桥接器不会将该包传送到另一个网络。

除了将网络互连在一起,桥接器还可以增加效率、可靠性和安全性。我们知道,冲突检测系统要求计算机在数据冲突发生时推迟传输。随着网中计算机数目的增加,数据冲突的数目也会增加。冲突多发生在许多计算机希望在同一时间发送数据。随着网络中计算机数目的增加,数据传输量也将增加。在大多数情况下,每个网络都有一个可以使用的最大数据传输量。超过这个最大数据传输量,网络就会发生阻塞。我们可以使用桥接器解决数据传输量过重所引起的效率下降问题。实际上,我们可以使用桥接器将一个拥挤的网络分成两个或更多个网络,从而减轻每个网络的传输量。

网络管理员经常使用桥接器连接两个或更多的小网络。我们知道,在环形和总线拓扑结构中,数据总线的损坏将造成整个网络通信的终止。将单个的 LAN 分成许多个较小的 LAN,网络管理员就可以减轻数据总线损坏造成的影响。通过增加桥接器,将一个 LAN 分成两个网络,网络管理员就可以提高整个网络的可靠性,这是因为一个故障只会对网络的一半而不是整个网络产生影响。

路由器主要用在使用不同技术的两个网络间传输数据。由于 Internet 包含成千上万使用不同技术的网络,所以路由器是 Internet 的一个基本组成部分。

路由器在网络上有地址,而桥接器没有。这是桥接器和路由器的主要区别。网络经常将路由器当作中间站,即计算机将发送到另一个网络的数据包发送给路由器,路由器再把数据包发送给其它网络。而桥接器只对总线上的数据进行检查,决定哪一个数据包在网络之间传输(此时计算机不会直接将数据包发送给桥接器)。此外,除非数据包包含路由器地址,否则路由器决不会检查数据总线上的数据包。

网关是一个派生术语,它可以指三种类型的网络实体。你可以把路由器看作网关。网

关也可指应用网关，后者对特定网络程序使用的数据进行解释。

1.1.5 网络结构

从概念上讲，网络设计将网络分成两个基本组成部分：网络应用程序和网络通信子系统。网络应用程序使用通信子系统在网络上传输数据。在 Internet 中，我们常把网络通信子系统称为通信子网或传输系统。通信子系统的主要任务是在网络计算机和应用程序之间传输数据和信息。

Internet 通过通信线路和网络交换设备传送数据，采用的主要技术手段是分组交换网络，它包含两个基本的组成部分——交换设备和传输线路。通信子系统的交换设备由一台计算机或者桥接器、路由器之类的设备组成。我们可以将交换设备看作指导数据包寻址的设备。

Internet 文献通常使用接口信息处理机（IMP）、通信计算机、分组交换、节点或数据交换设备等术语表示网络交换设备。此外，传输线通常称为线路、通信信道或通路。如果专指网络中流动的数据，则称传输线为数据流。Internet 和网络术语将数据在 Internet 上的传输称为“把数据发送到信道”或“把数据发送到线路上”。

1.2 网络协议

为了创建模块化可重用的网络结构，网络设计人员使用分层设计方法，即网络的每一层负责完成不同的功能。例如，最底层完成最基本的通信功能，然后增加错误检测功能以形成第二层。通过各层完成不同功能的分工，网络设计人员建立了一个各层都容易理解的网络结构。

1.2.1 ISO/OSI 参考模型

ISO/OSI 参考模型将一个网络表示为按功能分层的结构。ISO/OSI 模型本身虽然不是某个具体网络的设计规范或蓝图，但我们可以使用它设计网络。在下一章里，我们将讨论 TCP/IP 协议组，ISO/OSI 网络模型的知识将有助于我们更好地理解 TCP/IP 协议组。

ISO/OSI 模型用层将一个网络分成定义好的功能化模块。网络设计人员按照此模型关于各层的描述建立实际的网络。然而我们应该了解，网络设计人员完全可以根据网络的作用改变层数、各层名称以及各层功能，这意味着一个按照 ISO/OSI 模型建立的网络可能和按照同样模型建立的其它网络有很大的不同。

在分层网络中，每层为其相邻层提供特殊的服务，实现专门的功能。另外，每层对其上面各层隐藏了低层的实现细节。也就是说，各层只关心与网络中下一层的接口。每个网络层为其上层提供通信服务。

1.2.2 网络协议

在网络中，协议表示一个同样类型的抽象，通过它可以设计网络通信。协议是一系列

通信规则与约定，即数据通信协议制定了一组规则，应用程序在发送和接收数据时必须遵守它们。

在分层网络中，每个网络层使用定义好的协议与它周围的各层通信。在 ISO/OSI 模型中，各层的名称表示各层的协议。例如，将传输层协议表示为传输协议。同样，网络层协议称为网络协议。你可将层与层之间的通信称为对话。协议定义了这些对话的规则。

从理论上说，当两台主机进行通信时，它们的相应层也进行对话，我们可以将不同网络主机内的各层称为对等实体。对等实体之间的通信叫做虚通信。实际上，网络计算机之间的通信只发生在网络的最低层——物理层，只有在那里才存在真正的物理连接。我们一般将发生在物理层之间的通信称为实通信。

当两台计算机通信时，每台计算机的不同层之间要进行对话。在网络模型中，主机发送的数据从最上层传递到最底层（也就是物理层）。在物理层中，数据通过实际的通信通道水平地传递到目的主机。在目的主机，数据从最底层向上传递到最上层。发送方的任意一层发送的数据最终要到达目的主机的相应层。网络不直接在对等实体之间传输数据。

网络设计人员通过使用虚拟通信，可以忽略通信实体下面的分层细节，我们一般将这种通信方式称为虚拟对话，它简化了网络编程。

1.3 网络通信服务

ISO/OSI 模型的每层为其上面各层提供专门的通信服务。除了最底层，上面各层都依赖下一层完成其特定的功能。网络设计人员用每层为其上层提供的服务来指代它的功能。网络的每层都遵循一定的规则（协议），这些规则定义了它所提供的服务。

1.3.1 网络协议与服务模式

网络服务定义了诸如网络的功能，而网络协议则定义了网络软件提供这些服务时使用的数据包格式和结构。网络各层请求下层提供服务时，各层的协议必须定义这些服务请求时使用的数据包格式和结构。这些服务请求最终传到物理层，在那里将变成数据包。

网络使用不同的方法（或称服务模式）在对等实体间提供同样的通信服务。简单地说，服务模式规定了本层如何实现功能的方法。网络设计人员一般需要将服务模式网络协议结合在一起，即当需要以特定的方式完成某种服务时，我们必须指定提供这种服务模式的协议。

一般而言，应用程序能够指定某种特定的服务模式类型。例如，如果应用程序需要进行差错控制，我们就应该按照具有差错控制功能的协议规范设计程序，如果现存的协议不能提供应用程序所要求的所有服务模式，则我们必须设计自定义协议以实现该模式。

我们可以请求某个通信服务，该服务必须由网络使用特定的协议来提供。此时如果使用该特定协议的服务不能实现，我们将会得到一个错误信息。如果没有指定使用的协议，网络将使用缺省协议。应用程序要求的服务模式通常决定你所选择的协议。

如前所述，电路交换是通信交换的一种方式。电路交换在需要通信的两台设备间建立一个称为电路（Circuit）的通道，习惯上将此类电路称为点到点连接。一般来说，网络连接