

Visual C++.NET
编程解疑系列丛书

Visual C++.NET

网络编程

万跃华 主编

万跃华

胥芳 编著

张宪

计时鸣 主审



科学出版社

Visual C++.NET 编程解疑系列丛书

Visual C++.NET 网络编程

万跃华 主编

万跃华 胥芳 张宪 编著

计时鸣 主审

科学出版社

北 京

内 容 简 介

全书共 10 章, 涉及 240 多个问题, 覆盖了 Visual C++.NET 网络编程的大部分内容。本书利用大量生动有趣的编程案例、编程技巧从解决问题和答疑解惑入手, 以因特网上最新资料为蓝本, 以简洁明快的语言、清晰直观的条理, 比较全面地对 Visual C++ 网络编程过程中常见问题及故障给予了具体解决办法和答案。深入浅出地说明了 Visual C++ 中最典型的和用途最广的程序设计方法, 并且每个专题都有较全面的论述, 许多内容是一般介绍 Visual C++ 网络编程的书籍没有涉及的。

全书结构清晰、合理, 范例实用、丰富, 具有实用性和指导性。本书对从事 Visual C++.NET 开发与应用的广大科研人员、高校相关专业师生及编程爱好者亦不失为一本重要的自学、教学参考书。

图书在版编目(CIP)数据

Visual C++.NET 网络编程 / 万跃华主编. —北京: 科学出版社, 2002
(Visual C++. NET 编程解疑系列丛书)

ISBN 7-03-010770-5

I. V… II. 万… III. C 语言—程序设计 IV. TP 312

中国版本图书馆 CIP 数据核字 (2002) 第 065196 号

策划编辑: 陈晓萍 / 责任编辑: 韩洁

责任印制: 吕春珉 / 封面设计: 王浩

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

新 蕾 印 刷 厂 印 刷

科学出版社总发行 各地新华书店经销

*

2002 年 9 月第 一 版 开本: 787×1092 1/16

2002 年 9 月第一次印刷 印张: 26

印数: 1—5 000 字数: 580 000

定价: 38.00 元

(如有印装质量问题, 我社负责调换〈路通〉)

丛 书 编 委 会

主 编 张立彬

副主编 万跃华

委 员 张立彬 万跃华 计时鸣

赵章风 陈胜勇

主 审 计时鸣

前 言

Visual C++.NET 是 Microsoft 公司 2001 年中期推出的最新一代可视化开发工具，是 Visual Studio.NET 中的组成部分，因版本延续及称呼方便之故，Visual C++.NET 又被称为 Visual C++ 7.0。利用 Visual C++.NET，用户可以开发出企业规模的 Web 应用程序，以及高性能的桌面应用程序。

Visual C++.NET 具有开发过程简明、开发方式灵活、扩展性强等特点，是一种被广泛使用的前端开发平台。Visual C++.NET 提供了多种多样的网络编程技术，例如 Winsock、WinInet、ActiveX、DCOM、ATL 等，为用户开发网络应用程序提供全方位的支持。

能够得心应手地运用 Visual C++ 进行网络开发是广大程序员的愿望，本书的目的是希望读者能够在较短的时间内较好地掌握并运用 Visual C++ 进行网络应用开发，我们通过自己在开发软件和教学过程中所得到的经验和教训，总结出了通过编程案例、编程技巧进行学习的一种高效的方法。通过编程案例、编程技巧，可以使我们更加清楚地感受到一种语言或一类开发者们最初的意图，用户只有真正了解一种语言的意图所在，使用它们时才不至于感到迷茫，才能真正达到随心所欲使用的地步。

本书的主要特色是采用了编程案例、编程技巧的常见问题解答的形式来编写，着重讨论了在 Visual C++ 网络编程中可能遇到的典型问题、疑难问题和编程技巧。利用大量生动有趣的编程案例、编程技巧从解决问题和答疑解惑入手，以因特网上最新资料为蓝本，以简洁明快的语言、清晰直观的条理，比较全面地对 Visual C++ 网络编程过程中常见问题及故障给予了具体解决办法和答案。在解答常用问题的同时向读者介绍可视化编程的技术和软件开发的思维方式，以便读者能够从中领悟到一些深层次的、无法用文字表述的编程技巧；并使读者对其有一个完整的感性认识和美好印象，从而通过本书中的编程范例、编程技巧轻松地领会和了解 Visual C++.NET。大部分编程案例、编程技巧是一般介绍 Visual C++ 网络编程的书籍没有涉及的。每个问题的范例程序都涉及到 Visual C++.NET 网络编程的各个方面的内容并经过调试。为了帮助读者更好地理解，我们将自动生成的代码和手工增加的代码用不同字体加以区别(程序中必须使用手工添加的代码以斜体表示)。这些范例大多是作者的实践结果，有些是我们科研课题的实例，是针对程序员在开发过程中最迫切需要、使用频率最高的内容特意制定的，读者可以轻而易举地重新实现，读者还可以对这些范例进行研究、修改和扩展，使其符合自己的要求。

本书的另一个特点是不同于一般的 Visual C++ 网络编程参考书，在解答常见问题的同时，向读者介绍网络编程的基本概念、编程技巧和开发方法，系统剖析了 Visual C++ 的各类知识，且不需要花费大量的时间系统学习，可以跳跃阅读，各种类型的读者均可通过阅读，从中发现对自己有用的内容。

本书由万跃华主编，浙江工业大学万跃华、胥芳、张宪、赵章凤编著，吴翌、郑科、方喆也参加了部分编写及程序调试工作。全书由万跃华策划，万跃华、胥芳负责修改、

整理和定稿。本书由浙江工业大学计时鸣博士主审。

在本书编写的过程中,我们得到了单位领导和许多同仁以及网友的大力支持和帮助,并承蒙他们提出了很多宝贵意见和建议,有些内容是受到他们的启发才编入此书的,在此深表谢意。感谢香港城市大学陈胜勇博士,浙江工业大学的王忠飞、姚春燕等老师,王杭亮等同学提供了许多资料并对本书提出了许多中肯的修改意见。没有他们的支持和通力合作,本书就不可能与读者见面。

在本书编写过程中,我们曾参阅了国内外多种同类教材和教学参考书,以及国内外 Visual C++ 网站、VC 讨论组、VC 新闻组、VC 邮件列表、VC 专业杂志、VC 电子期刊等,所涉及的内容非常丰富、广泛。鉴于作者水平有限,书内谬误和疏漏之处在所难免,敬请读者批评指正。

万跃华

2002 年 5 月于杭州

目 录

第一章 网络编程基础	1
1.1 网络拓扑结构	1
1.2 以太网技术	3
1.3 其他高速网络技术	4
1.4 光以太网的现状及展望	6
1.5 Internet 整体结构	10
1.6 网络分层模型	11
1.7 TCP/IP 协议	14
1.8 网络进程通信原理	21
第二章 Windows Sockets (WinSocket)	23
2.1 Socket 的由来	23
2.2 Windows Sockets 规范	23
2.3 Windows Sockets 中一些重要概念	24
2.4 基于 Windows Sockets 网络编程	27
2.5 Windows Socket API 函数	27
2.6 Windows Sockets 与 UNIX Sockets 连接编程	30
2.7 利用 MFC 的 Windows Sockets 类进行网络编程	33
2.8 异步 Windows Sockets 编程	33
2.9 使用 CAsyncSocket 类编程	34
2.10 使用 CSocket 类编程	37
2.11 用 CSocket 实现无连接通信	44
2.12 CAsyncSocket 和 CSocket 编程的比较	47
2.13 用自己派生的 Socket 类进行网络通信	47
2.14 在服务器端控制 Socket 通信的客户端数目	52
2.15 一个 WinSock 程序中同时可以打开多少个 Socket 连接	52
2.16 为服务器设置一个合理的端口号	53
2.17 在 Windows Socket 程序中得到本机的 IP 地址	53
2.18 原始套接字	56
2.19 Ping 命令简介	56
2.20 Internet 控制信息协议 (ICMP)	57
2.21 用 Windows Sockets 实现 Ping 命令	58
2.22 用 Windows Sockets 实现 HTTP	63
2.23 用 Windows Sockets 实现 FTP	65

2.24	Windows Sockets 1.1 与 Windows Sockets 2 的区别	70
2.25	截获 Windows Sockets	73
2.26	将 Sockets 程序从 UNIX 移植到 Windows	74
2.27	建立客户 CSocket	76
2.28	利用“监听-转发”程序破译网管协议	76
2.29	穿透代理服务器编程	82
第三章	Win32 Internet (WinInet)	88
3.1	什么是 WinInet	88
3.2	WinInet 如何使创建 Internet 客户端应用程序更容易	88
3.3	MFC 如何使创建 Internet 客户端应用程序更容易	89
3.4	用 MFC WinInet 类编写 Internet 客户端应用程序	90
3.5	用 MFC 建立通用的 Internet 客户端连接	91
3.6	用 MFC 编写 HTTP 客户端应用	94
3.7	用 MFC 实现 FTP 客户端应用	96
3.8	用 MFC 编写 gopher 客户端应用	98
3.9	下载 Web 页	101
3.10	FTP 文件	102
3.11	检索 Gopher 目录	102
3.12	使用 OnStatusCallback	103
3.13	传输文件时显示进度信息	103
3.14	利用 Win32 网络函数创建一个网络浏览器	104
3.15	用 Win32 网络函数编写 HTTP 网络浏览器	107
3.16	用 WinInet 实现 HTTP 的 POST 方法	112
3.17	HTTP 查询应用实例	113
3.18	FTP 查询应用实例	117
3.19	gopher 查询应用实例	119
3.20	Finger 查询应用实例	121
3.21	Whois 查询应用实例	123
3.22	使用 WinInet 访问加密协议	126
3.23	在使用 Internet 后挂断线路	126
第四章	Internet Server API (ISAPI)	128
4.1	什么是 ISAPI	128
4.2	什么是 Internet 服务器扩展	128
4.3	什么是 ISAPI 过滤器	133
4.4	ISAPI 与 CGI 的比较	138
4.5	ISAPI 的工作原理	139
4.6	需要什么样的 HTTP 服务器来运行 ISAPI	140
4.7	ISAPI 如何使创建 Internet 服务器应用程序变得更容易	140
4.8	使用 MFC 创建 Internet 服务器应用程序的优点	141

4.9 用 ISAPI 可以执行哪些操作	141
4.10 MFC 支持 ISAPI 的类和函数	142
4.11 MFC 的支持	144
4.12 创建典型的 ISAPI 扩展	146
4.13 创建典型的 ISAPI 过滤器	146
4.14 ISAPI 编程提示	147
4.15 服务器扩展 (ISA) 内存管理	147
4.16 过滤器内存管理	148
4.17 调试 ISA	148
4.18 生成基于 MFC 的 ISAPI DLL	149
4.19 应在何时将 ISAPI 用于数据库	149
4.20 安装和升级 ISA	150
4.21 安装 ISAPI 过滤器	150
4.22 使用 ISAPI 过滤器实现虚拟主机 WWW 站点	150
4.23 使用 ISAPI 过滤器实现访问内容进行统计分析	152
4.24 实现 CGI 到 ISAPI 的转换	154
4.25 使用 MFC 开发 ISA 扩展应用程序	155
4.26 用 MFC 实现基于 Web 的计算器	158
4.27 用 ISAPI 实现对 Web 数据库的访问	160
4.28 用 ISAPI 过滤器限定访问的 IP	167
第五章 Telephony API (TAPI)	174
5.1 什么是 Windows 开放服务结构 (WOSA)	174
5.2 什么是 TAPI	174
5.3 什么是 TSP	175
5.4 Unimodem 和 Unimodem/V	175
5.5 电话设备和 TAPI	175
5.6 什么是服务器 TAPI 模型	176
5.7 TAPI 的标准有哪些	176
5.8 TAPI 的服务	176
5.9 TAPI 通信过程	177
5.10 异步 TAPI 函数的使用	185
5.11 增补电话 TAPI	187
5.12 扩展电话 TAPI	190
5.13 TAPI 3.0 概念	190
5.14 TAPI 3.0 的四个组件	192
5.15 TAPI 3.0 的呼叫控制模型	193
5.16 得到 LINEERR_OPERATIONUNAVAIL	193
5.17 往外打出时为什么在电话接通前得到一个 LINECALLSTATE_CONNECTED 消息	194

5.18	为什么不能检测对方挂机	194
5.19	怎样用 TAPI 收发传真	195
5.20	用 TAPI 实现电话拨号程序	206
5.21	如何实现拨号上网	215
第六章	Messaging API (MAPI)	225
6.1	什么是 MAPI	225
6.2	MAPI 的体系结构	225
6.3	安装 MAPI	227
6.4	什么是简化 MAPI	228
6.5	什么是扩展 MAPI	230
6.6	MFC 中的 MAPI 支持	231
6.7	用 MFC 实现“发送邮件”命令	231
6.8	在 API 层次上实现 MAPI 编程	233
6.9	初始化 MAPI 并登录到电子邮件对象	233
6.10	阅读电子邮件	236
6.11	发送电子邮件	239
6.12	什么是通用消息调用 (CMC)	242
6.13	如何开始 CMC 对话	243
6.14	用 CMC 发送消息	245
6.15	用 CMC 分辨消息的地址	248
6.16	用 CMC 接收消息	249
6.17	使用 CMC 扩展	252
6.18	用 API 编写的一个完整的邮件发送程序	253
第七章	CryptoAPI (CAPI)	262
7.1	加密 101	262
7.2	密码认证的控制措施	264
7.3	电子商务中的安全交易手段	265
7.4	CryptoAPI 的体系结构	267
7.5	CryptoAPI 的配置	267
7.6	与服务提供程序连接	268
7.7	生成密钥	269
7.8	输入/输出密钥	273
7.9	简单数据加密	276
7.10	多块数据 (文件) 加密	278
7.11	加密时进行散列编码	278
7.12	数据解密	279
7.13	数字签名	280
7.14	加密/解密文件	287

第八章 ActiveX 及 ActiveX 模板库 (ATL)	301
8.1 什么是 ActiveX	301
8.2 ActiveX 控件	301
8.3 ActiveX 是否只是 OLE 的一个新名字	304
8.4 ActiveX 是否与 Java 竞争	304
8.5 ActiveX 控件与 Netscape Plug-Ins 的区别	304
8.6 ActiveX 控件与 Java Applet 的优劣	305
8.7 哪种平台可运行 ActiveX	306
8.8 ActiveX 与 Internet	306
8.9 ActiveX 如何解决 Internet 安全性问题	308
8.10 ActiveX 的安全性讨论	308
8.11 设置 IE 中的安全级别来启动与禁用 ActiveX 控件	309
8.12 ActiveX 控件下载后, 放在什么位置	309
8.13 压缩和打包 ActiveX 控件	309
8.14 在 Web 页中嵌入 ActiveX 控件	310
8.15 开发 ActiveX 控件的几点要求	313
8.16 MFC ActiveX 控件向导中的控件设置	313
8.17 测试 ActiveX 控件	314
8.18 调试 ActiveX 控件	315
8.19 在应用程序中加入 ActiveX 控件	316
8.20 什么是 ATL	319
8.21 ATL 的基本技术	319
8.22 为什么要用 ATL	320
8.23 模板库与标准 C++库有什么不同	322
8.24 哪些 ATL 类便于 ActiveX 控件包容	322
8.25 什么是 ATL 控件宿主 API	322
8.26 向 MFC 项目添加 ATL 支持	323
8.27 调试 ATL 对象	323
8.28 什么是 AtlAxWin7	324
8.29 在运行时加载指定的控件	324
8.30 加载在对话框资源上指定的控件	325
8.31 加载授权的控件	325
8.32 处理授权的控件	325
8.33 何时需要调用 AtlAxWinInit()函数	326
8.34 什么是宿主对象	326
8.35 可以在单个窗口中承载多个控件吗	326
8.36 可以重用宿主窗口吗	326
8.37 调整控件大小	326
8.38 获取指向控件的接口指针	326

8.39	为控件设置环境属性	327
8.40	销毁控件	327
8.41	何时需要调用 AtlAxWinTerm()函数	327
8.42	用 ATL 开发 ActiveX 控件	327
第九章	Microsoft.NET 和 Visual Studio.NET 下的网络编程	341
9.1	Microsoft .NET 的定义	341
9.2	.NET 的核心架构	341
9.3	.NET 是运行库还是开发平台	341
9.4	为什么要使用 .NET	342
9.5	.NET 类的设计目标	344
9.6	使用 .NET 类	344
9.7	什么是 XML	349
9.8	什么是 SOAP	350
9.9	ATL 和 COM 在 .NET 中会扮演怎样的角色	350
9.10	可以在 .NET 框架程序中使用 COM 对象吗	350
9.11	可以在 COM 程序中使用 .NET 框架组件吗	350
9.12	Microsoft.NET 与 Java	351
9.13	什么是 C#语言	351
9.14	C#能开发什么	352
9.15	C#与 C++相比有什么新特性	352
9.16	为什么要用 C#进行开发	353
9.17	在 C#程序中调用 Win32 API	354
9.18	用 C#开发简单的 Windows 应用程序	355
9.19	用 C#得到本机的 IP 地址	356
9.20	用 C#发送邮件	357
9.21	在 C#程序中下载网页	359
第十章	网络编程实例：收发电子邮件	360
10.1	基于 Windows Sockets 实现 SMTP	360
10.2	基于 Windows Sockets 实现 POP3	385

第一章 网络编程基础

1.1 网络拓扑结构

在学习 Internet 网络编程前，我们应该对其结构有一个较全面的认识，这对于深入学习 Internet 网络编程是非常有益的。Internet 是由众多的局域网和局域网连接而成的。其中局域网技术发展最为快速，是网络发展的一个热点。因此了解局域网的一些基础知识，是一个网络程序员所必需的。

网络的拓扑结构是抛开网络物理连接来讨论网络系统的连接形式，网络中各站点相互连接的方法和形式称为网络拓扑。拓扑图给出网络服务器、工作站的网络配置和相互间的连接，它的结构主要有星型结构、总线结构、树型结构、网状结构、蜂窝状结构、分布式结构等。

1. 星型结构

星型结构是指各工作站以星型方式连接成网。网络有中央节点，其他节点（各节点可以是计算机也可以是一个由连接器连接起来的局域网段，通过连接器可以将网络扩展。）都与中央节点直接相连，这种结构以中央节点为中心，因此又称为集中式网络。它具有如下特点：结构简单，便于管理；控制简单，便于建网；网络延迟时间较小，传输误差较低。但缺点也是明显的：成本高、可靠性较低、资源共享能力也较差。

正是因为星型结构每一个连接都对应相应的设备，所以当网络中连接点出了故障时，只会影响到响应的连接点而不是整个网络，同时在出故障时也易于检测和修理。在星型结构中，连接方式比较简单，从而使用的访问协议也比较简单。

由于星型网的结构特性，使得中央节点比较复杂，负担较重。在扩展时由于每增加一个节点都要在中央节点上增加相应的连接，这就在扩展时增加了困难。另一方面，网络结构对中央节点的依赖性过大，如果中央节点出了故障将导致整个网络的瘫痪。

该结构现在主要用于一些有大量数据处理和多媒体通信的信息网络中，还有就是有一些智能集中于中央节点的网络。

2. 环型结构

环型结构由网络中若干节点通过点到点的链路首尾相连形成一个闭合的环，这种结构使公共传输电缆组成环型连接，数据在环路中沿着一个方向在各个节点间传输，信息从一个节点传到另一个节点。

环型结构具有如下特点：信息流在网中是沿着固定方向流动的，两个节点仅有一条道路，故简化了路径选择的控制；环路上各节点都是自举控制，故控制软件简单；由于信息源在环路中是串行地穿过各个节点，当环中节点过多时，势必影响信息传输速率，

使网络的响应时间延长；环路是封闭的，不便于扩充；可靠性低，一个节点故障，将会造成全网瘫痪；维护难，对分支节点故障定位较难。

环形网络是由一些中继器（环接口）和连接设备的通信通道组成的一个环形网络。中继器可以接收到一条通信通道上的信息，同时也可以相同的速度把信息用串行通信的形式发送到另一条通信通道上去。信息在中继器中不缓存，传输时为单向。

环接口就是中继器。环形结构最广泛的应用就是令牌环网（Token Ring）。所谓令牌环就是一种特殊的控制帧，也就是一种控制信息。一个环只有一个令牌。令牌不断在网中循环，某个环接口一旦抓住令牌，便可进入传输方式，发送数据。这时，其他的站点只能进入接收方式。

3. 总线型结构

总线结构是指各工作站和服务器均挂在一条总线上，各工作站地位平等，无中心节点控制，公用总线上的信息多以基带形式串行传递，其传递方向总是从发送信息的节点开始向两端扩散，如同广播电台发射的信息一样，因此又称广播式计算机网络。各节点在接受信息时都进行地址检查，看是否与自己的工作站地址相符，相符则接收网上的信息。

总线型结构的网络特点如下：结构简单，可扩充性好。当需要增加节点时，只需要在总线上增加一个分支接口便可与分支节点相连，当总线负载不允许时还可以扩充总线；使用的电缆少，且安装容易；使用的设备相对简单，可靠性高；维护难，分支节点故障查找难。

在总线结构中，一个设备发出的信号能被所有设备接收到。连接器软件可以通过设备的标识号（即地址）判断是否接受信号。因为所有的设备公用一个传输通道，所以一次只能有一个设备发出传输信息。通常要通过分布控制的方式来决定哪个设备可发送信息。

在发送信息时，信息源节点先将信息报文分组，然后按顺序发送这些被分成组的报文，有时还会与其他节点发出的报文在介质中交替传输。但信息经过各站点时，各站点都将检查报文中的地址，来确认是否该接收，只有目的站点接受下报文。

由于这种结构是建立在一个公用的信息通道上，所以在安装和维护上较容易，也便于扩充新的站点。虽然它简单可靠，但故障检测却不是很容易，因为这种结构的网不是集中控制的，检测时需要在各个站点上进行，并且在对某个站点进行隔离时也要把整个网络停止下来。

4. 分布式结构

分布式结构的网络是将分布在不同地点的计算机通过线路互连起来的一种网络形式，分布式结构的网络具有如下特点：由于采用分散控制，即使整个网络中的某个局部出现故障，也不会影响全网的操作，因而具有很高的可靠性；网中的路径选择最短路径算法，故网上延迟时间少，传输速率高，但控制复杂；各个节点间均可以直接建立数据链路，信息流程最短；便于全网范围内的资源共享。缺点为连接线路用电缆长，造价高；网络管理软件复杂；报文分组交换、路径选择、流向控制复杂；在一般局域网中不采用

这种结构。

5. 树型结构

树型结构是分级的集中控制式网络，与星型相比，它的通信线路总长度短，成本较低，节点易于扩充，寻找路径比较方便，但除了叶节点及其相连的线路外，任一节点或其相连的线路故障都会使系统受到影响。

6. 网状拓扑结构

在网状拓扑结构中，网络的每台设备之间均有点到点的链路连接，这种连接不经济，只有每个站点都要频繁发送信息时才使用这种方法。它的安装也复杂，但系统可靠性高，容错能力强。有时也称为分布式结构。

7. 蜂窝拓扑结构

蜂窝拓扑结构是无线局域网中常用的结构。它以无线传输介质（微波、卫星、红外等）点到点和多点传输为特征，是一种无线网，适用于城市网、校园网、企业网。

在计算机网络中还有其他类型的拓扑结构，如总线型与星型混合、总线型与环型混合连接的网络。在局域网中，使用最多的是总线型和星型结构。

1.2 以太网技术

以太网（Ethernet）是局域网中技术最为成熟的、应用最广泛的一种网络。国际电子电气工程师协会（IEEE）早在 20 世纪 70 年代就制定了三个局域网标准：IEEE 802.3（CSMA/CD）、IEEE 802.4（令牌总线）和 IEEE 802.5（令牌环）。以太网就是 IEEE 802.3 的一个典型产品。

以太网是一种总线型局域网技术，它采用载波监听多路访问/冲突检测的传输方式（CSMA/CD）。局域网一般都是广播型网络，网上站点共享信道，某一站点发出数据，其他站点均能收到，就如同一个人在公共场所大声讲话，在场的所有人都能听见一样。而广播型网络需要解决的一个重要的技术问题就是信道竞争，也就是传输线路的竞争。网络中的每一个站点都可以使用信道，但信道某一时刻只能由一个站点使用，当很多站点同时申请使用信道时，怎么解决这个问题呢？以太网采用的方法就是 CSMA/CD。它的原理很简单：一个站点要传送数据时，它首先监听信道（载波监听），如果这时信道有数据传输，则等待信道空闲再传送数据。由于同时可能有多个站点监听到信道空闲发出数据（多路访问），就可能发生冲突。CSMA/CD 采用简单而有效的方法解决冲突，在发送数据的同时，进行冲突检测，一旦发生冲突，立刻停止数据传输，并等待冲突平息，再进行 CSMA/CD，直到数据传输完为止。

基于 IEEE 802.3 的总线局域网通常采用以下三种组网方式：

（1）10Base-T

将所有计算机通过双绞线连到一个中央集线器（Hub）上，其物理结构是星型的，但是逻辑结构是总线的。这种组网的特点是管理方便，容易增加节点，并且组网方便，

很多小型的局域网都采用这种方式。10Base-T 的含义是：10Mbps 的传输速率，采用基带传输方式，传输介质为双绞线。

(2) 10Base-2

又称为细缆以太网。采用直径 5mm 的同轴电缆作为传输介质，传输速率是 10Mbps。细缆以太网价格便宜，易于安装，但是当增加一个新的节点时，整个网络都要停止运行。网络的可靠行较差。

(3) 10Base-5

又称为粗缆以太网。采用直径 10mm 的同轴电缆作为传输介质，传输速率是 10Mbps。将计算机连接到粗缆上必须使用专门的收发器。粗缆以太网的传输距离可以达到 2.5km。

随着网络的发展，越来越多的 PC 连入局域网，加上 PC 上大量运行基于图形界面的应用程序以及大量网络服务的出现，使用网络流量激增，10Mbps 的传统以太网已经不能满足需要，为此迫切要求开发出快速网络结构以满足流量不断增长的需求。百兆以太网和千兆就是为满足这个要求开发研究的。现在百兆以太网技术已经很成熟，被用在高速的局域网和主干网络上，而千兆以太网正处于实验当中，目前，只有少许千兆位以太网产品推出并开始进入实际应用。

1.3 其他高速网络技术

1. 光纤分布数据接口 (FDDI)

FDDI 是由美国国家标准协会 (ANSI) 的 X3T9 委员会在 20 世纪 80 年代中期制定的标准，最初的设想是通过光纤传输 100Mbps 的数据。FDDI 采用光纤为主要的传输介质，与铜介质相比，光纤有很多优点，尤其在安全性、可靠性及整体性能方面。FDDI 通常被用作高速骨干网技术，因为它比铜轴电缆支持更高的带宽和更远的传输距离，应该注意到的是，最近出现了一种被称为 CDDI 的分布式铜轴电缆，它能够在铜轴双线上实现 100Mbps 的传输速度，是 FDDI 协议在铜轴双绞线上的实现。

FDDI 采用双环体系结构，两环上的信息反方向流动（称为反向循环）。双环中的一环称为主环，另一环则称为次环。在正常情况下，主环传输数据，次环处于空闲状态。双环设计的主要目的是提供高可靠性和稳定性。主干网上的各台计算机都已经安装了 FDDI 网络适配器，在网络环中传送分组数据。计算机从它相邻一边的计算机中收到一个分组，然后再把它传输到另一边相邻的计算机上。这种情况下，网络中的每台计算机都充当着中继器的角色，当它需要再次发送数据分组时，就重新产生分组。但是，大多数的 FDDI 适配器具有旁路模式，这种模式允许数据通过而不需要重新发送。在某台计算机暂停不用的时候，旁路模式用来保护 FDDI 环的完整。

FDDI 具有良好的性能，带宽分配机制的优势在多媒体和视频应用中真正得到了体现。例如，当运行视频应用程序时，像视频点播等服务，画面和声音都能既无时延又无抖动地显示在客户端。FDDI 具有相当强的容错和管理能力，一旦网络建立并开始运行，则需要很少的维护就可以可靠稳定地运行。并且 FDDI 自身也在不断发展中，FDDI-2 作为一种相关的标准已经被开发出来，它用来传输不能延时的数据，但是要求网络上所有

的节点都必须符合 FDDI-2 标准,如果有 FDDI-2 的设备存在,则 FDDI-2 的设备就像 FDDI 的设备一样运行。所以对于以下领域可以使用 FDDI,例如:多媒体和视频业务、服务器和主干网,而对于小型局域网不宜采用 FDDI,因为其实现价格过高,而选用 100Mbps 以太网则是一个很好的选择。

2. 铜缆分布数据接口 (CDDI)

铜缆分布数据接口 (CDDI) 是 FDDI 协议在铜轴双绞线上的实现,与 FDDI 一样, CDDI 的速率为 100Mbps,也采用双环体系结构提供冗余特性。CDDI 支持的工作站与集线器的最大距离为 100m。

CDDI 标准是由 ANSI 的 X3T9 标准委员会制定的其正式名称为双绞线物理介质相关 (TP-PMD) 标准,为了保持与 FDDI 术语的一致,它也被称为双绞线分布数据接口 (TP-DDI)。CDDI 与 ANSI 标准定义的物理层和介质访问控制层一致。

ANSI 只认可 CDDI 的两种电缆类型即屏蔽双绞线 (STP) 和非屏蔽双绞线 (UTP), STP 电缆阻抗为 150Ω ,符合 EIA/TIA568 (IBM Type1) 标准 UTP 是数据级的电缆 (5 类),由四对非屏蔽双绞线组成,每个双绞线紧绕在一起,并采用专门的绝缘酯塑料套,符合 EIA/TIA568B 标准。

3. 异步传输模式 ATM

简单地说,ATM 是一个基于信元、面向连接、全双工和点到点的通信协议,该协议对各个站点都具有专门宽带,它使用异步分时复用技术通过网络的信息流量。1991 年,由厂商、通信公司和用户组成了一个联合会,称为 ATM 论坛,目的是促进北美关于 ATM 接口的工业协议。ATM 论坛对建立广泛的工业化 ATM 标准是一种驱动力。尽管目前大部分开发工作集中在 155Mbps ATM 上,但是 ATM 可以运行在 25—622Mbps 的宽带范围上。ATM 具有以下优点:很好的扩展能力,按需分配宽带,处理各种网络业务,局域网 (LAN) 和广域网 (WAN) 的自适应能力。

ATM 是为替换现有的传输协议而开发的,现有的协议指的是在宽带和扩展能力上明显受限制的以太网和令牌环等协议。而且,ATM 设计称能够同时处理各种类型的数据,并且效率更高。因此,ATM 能够传输各种范围的比特速率,而且支持各种突发通信,如语音,数据和视频业务等。ATM 不同于其他局域网协议,它采用了信元结构,信元是一种固定长度的分组,不像其他协议的分组是可变长度的。各个 ATM 信元均由 48 字节的数据区和 5 字节的信头组成。采用固定长度的分组有以下优点:处理信元过程简单可靠,速度快,便于处理。

ATM 交换机是组成 ATM 网络的主要设备,ATM 的拓扑把交换机交织成一个网。这意味着网络中的任何一点可以经过在交换机间所包括的独立连接的多条路由到达网络中的任何其他一点。ATM 不需要任何特定的物理层协议,也没有任何距离限制,与那些受所用媒介衰减特性的影响而限制距离的网有所不同。这样就简化了缆线设备的建造,因为没有任何实际的规则来约束设计。155Mbps 的 ATM 将支持第三、四、五类屏蔽双绞线、类型 1 屏蔽电缆、光缆等局域网络媒介。

但是 ATM 毕竟是一项还在实验的技术,虽然很多厂家推出了他们的产品,可是某