Delphi 网络高级编程

鲍敏 吴昊 等 编著

人 民 邮 电 出 版 社

图书在版编目（CIP）数据

Delphi 网络高级编程/鲍敏、吴昊编著.—北京：人民邮电出版社，2001.8

ISBN 7-115-09525-6

I.D… II.①鲍…②吴…III. DELPHI 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字（2001）第 048457 号

内 容 提 要

本书介绍了使用 Delphi 实现 Windows 的网络编程的方法。书中的实例充分利用 Delphi VCL 与 Winsock API 的特点，方便、快捷地实现各种网络功能。全书由 4 部分组成：第一部分介绍 TCP/IP 协议、Winsock 知识和 Delphi 的基本网络组件；第二部分讲解如何利用 Delphi 和 Winsock 的优点实现主要协议以及局域网内部使用的应用程序；第三部分是网络上常用的编码算法与加密的实现等内容；第四部分附录提供了网络编程时的常用参考资料。

本书提供了内容丰富的实例，并附有全部实现代码。本书适合于利用 Delphi 进行网络程序开发的程序员阅读。

Delphi 网络高级编程

◆ 编　　著　　鲍敏　吴昊等

责任编辑　张立科

执行编辑　孙玉华

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@pptph.com.cn

网址　http://www.pptph.com.cn

北京汉魂图文设计有限公司制作

北京　　印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张　24

字数：582　千字　　2001 年 7 月第 1 版

印数：1—0 000 册　　2001 年 7 月北京第 1 次印刷

ISBN 7-115-09525-6/TP.2378

定价：54.00　元　（附光盘）

关 于 本 书

“真正的程序员用 C，聪明的程序员用 Delphi。”这句话体现了 C 与 Delphi 的最大区别。Delphi 的特色就在于它的 VCL 完美地封装了 Windows API，使用 VCL 可以非常方便地实现许多复杂的功能。选择 Delphi，就意味着可以获得相当数量的第三方组件的支持，可以使用大量的源代码，从而节省程序员的时间与精力。可是选择 C 或者 C++就不一样了，在编程的时候需要程序员深入了解并掌握 Windows 的复杂体系。

Delphi 程序总是运行在 Windows 平台上的。Delphi 的 VCL 虽然提供了对 Windows API 的完美封装，为一个程序员在初级阶段的快速成长提供了极好的学习环境，但这种成长是有着很强的依赖性的，一旦脱离了 VCL 的支持进入陌生领域，Delphi 程序员就会不知所措。只有了解了 Windows API，才能突破 VCL 组件的限制，深入 Windows 编程。特别是在进行网络编程的时候，需要调用大量的 Winsock API 协同工作，如果仅仅是使用 Delphi 提供的有限的 VCL 组件，最多也就是进行一些“大众化”的程序设计，更多的特殊网络功能就无法实现。

本书探讨了 Windows 的网络编程的问题，Delphi 只是其中使用的主要实现工具。实现过程充分利用 Delphi VCL 与 Winsock API 的特点，通过结合两者的长处，从方便、快捷的角度实现大多数的网络功能。

本书由 4 个部分组成。第一部分由第 1、2、3 章组成，主要介绍 TCP/IP 协议、Winsock 知识和 Delphi 的基本网络组件，讲解使用 Delphi 进行网络编程的基础知识。第二部分（第 4 章到第 12 章）是主要协议的实现部分。这部分的内容充分结合 Delphi 和 Winsock 的优点，根据 TCP、UDP、ICMP 等协议编制 CGI、ISAPI 以及局域网内部使用的应用程序。第三部分（第 13 章）是网络上常用的编码算法与加密的实现等一些在编制网络程序时与安全性等相关的内容。第四部分附录提供了一些网络编程的时候经常使用的参考资料。

本书适用于不满足仅仅利用 Delphi 的 VCL 网络相关组件进行网络程序开发的程序员。

编　者

2001 年 7 月

目　　录

第 1 章　TCP/IP 协议

本章主要介绍 Windows 网络编程中经常涉及的网络协议和功能，同时简单地介绍网络编程需要具备的基本概念。

1.1　TCP/IP 协议族

在今天，网络已经深入到世界各个角落了，它使用户脱离地域的分隔与局限，在网络所能达到的范围内实现资源的共享利用。在各种型号、各种系统的计算机之间，有多种通信协议负责它们之间的沟通。在这众多的协议中，TCP/IP 协议的应用是最为广泛的，已经成为了事实上的工业标准，主要用于广域网（WAN）和局域网（LAN）。至于其他的诸如 SLIP（Serial Line IP，即串行线路接口协议）和 PPP（Point-to-Point Protocol，即点到点协议）等都提供有效串行线的点到点连接。当然，SLIP 和 PPP 都被设计成支持 IP，这样任何基于 IP 的服务都可以通过这两种连接来运转。这里主要讨论有关 TCP/IP 协议族的内容。

TCP/IP 起源于 20 世纪 60 年代末美国政府资助的一个分组交换网络研究项目，到 20 世纪 90 年代已发展成为计算机之间最常应用的组网协议。它是一个真正的开放系统，该协议族的定义及其多种实现可以在公开的场合免费得到。Internet 就是采用 TCP/IP 协议作为共同的通信协议，将世界范围内许许多多计算机网络连接在一起，成为当今最大的和最流行的国际性网络。

1.1.1　OSI 模型

要了解 TCP/IP 协议族，先要了解国际标准化组织 ISO（International Standards Organization）为计算机网络通信制定的一个七层框架。这个七层协议的框架，称为“ OSI/RM ”（Open System Interconnection / Reference Model，开放系统互联参考模型），如图 1-1 所示。

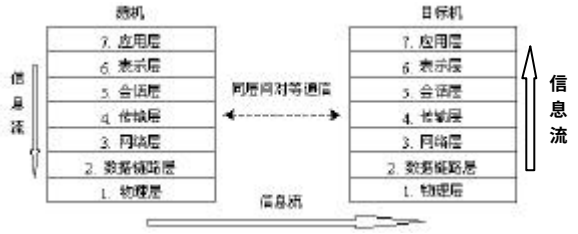

图 1-1　OSI 模型与通信流程

OSI 模型把计算机网络通信的组织与实现按功能划分为七个层次，即从一个计算机系统发出通信请求开始，到信息经过实际物理线路传送到另一个目标计算机系统为止，把通信功能从高到低划分为应用层、表示层、会话层、传输层、网络层、数据链路层和物理层，各层

的具体功能如表 1-1 所示。网络通信协议按层次组织，也是为了减少协议的复杂性。每一层协议建立在它的下层协议的基础上，每一层又为其上层协议提供服务，完成上层协议提交的任务。至于在同一层内进行服务的细节，对上层则是隐蔽的。一台计算机的某指定层同另一台计算机的相应层对话，对话的全部规则和约定就构成了该层的协议。当然，信息（数据和控制信息）并不是从某一计算机系统的第 *N* 层直接传到另一计算机系统的第 *N* 层，而是从这台计算机的某一层逐层向下传送至底层，最后经过物理介质到达另一台计算机，然后再由底层逐层向上传送，如图 1-1 中的数据流程所示。

当然，OSI 模型只是一个框架，它的每一个层并不执行某种功能，功能的具体实现还需通信协议，主要是软件来进行。当数据在层间向下传播时（源机部分），每一个层都会为传输中的数据增加一个包头（header），用于标识包的来源与目的。到了目标机位置时，每一层都从数据中读取并去除相应包头，执行请求的任务，并负责向上传输数据包。每一种具体的协议一般都定义了 OSI 模型中的各个层次具体实现的技术要求。

表 1-1　　OSI 模型中各个层的功能

名称	层次	功能
物理层	1	实现计算机系统与网络间的物理连接
数据链路层	2	进行数据打包与解包，形成信息帧
网络层	3	提供数据通过的路由
传输层	4	提供传输顺序信息与响应
会话层	5	建立和中止连接
表示层	6	数据转换，确认数据格式
应用层	7	提供用户程序接口

1.1.2　DoD 模型

OSI 模型为各种协议的实现规划了一个非常优秀的框架。但是在 TCP/IP 协议产生的时候，OSI 模型还没有产生，当时使用的是 DoD 模型。之所以使用这个名字，是因为该模型起源于美国国防部（Department of Defense，DoD）资助的一个项目。当然，也可以使用 OSI 模型去理解 TCP/IP 协议。

DoD 模型使用的是四层结构，分别是：进程/应用层、主机－主机层、Internet 层和网络访问层。每一层指定了一个通信协议集合，这整个协议集合称为 TCP/IP 协议组。虽然 DoD 模型定义的是四层结构而 OSI 模型定义的是七层结构，两者之间还是具有某种可比性的，如图 1-2 所示。

- DoD 模型的进程/应用层对应于 OSI 模型的上三层，主要的功能是在两个计算机系统间提供具体的应用接口。
- DoD 模型中的主机－主机层对应于 OSI 模型中的传输层，主要负责建立与维护连接。
- DoD 模型中的 Internet 层对应于 OSI 模型中的网络层，主要职责是在不同的网络系统之间寻找路由。
- DoD 模型中的网络访问层对应于 OSI 模型中的最底两层，体现网络间物理连接的物理特性。

图 1-2　DoD 模型与 OSI 模型层次对比

1.1.3　TCP/IP 主要协议

TCP/IP 协议族作为主要的网络协议，内部包含了超过 100 个的具体协议。在每个具体协议的制定和发展过程中，需要经过广泛地协商与讨论。在成熟或者基本成熟之后，就会推出相应地一个或多个 RFC（Request for Comments）文档，在文档中详细地描述了协议的使用场合、方法以及实现过程。这些 RFC 文档都是公开发表的，用户可以在网络上免费地下载。获得 RFC 文档的方式很多，包括使用 WWW、FTP 以及 E-mail 等。目前提供 RFC 下载的 FTP 站点主要有：

ftp.rfc-editor.org
nis.nsf.net
sunsite.org.uk（该站点另一个域名是 src.doc.ic.ac.uk）
wuarchive.wustl.edu
ftp.ncren.net
ftp.ietf.rnp.br
ftp.gigabell.net
ftp.fccn.pt
ftp.oasisstudios.com

- 该站点列表是随时更新的，可以通过电子邮件获得（必须按照指定的格式）：
  收信人：rfc-info@ISI.EDU
  主题：getting rfcs
  内容：help: ways_to_get_rfcs
- RFC 文档的编写是一个非常庞大的系统工程，目前该系列文档的 ZIP 压缩包超过 26MB，所以在下载 rfc.zip 的时候千万别忘记把 Rfc-index.txt 文件一起下载。Rfc-index.txt 文件包含了 RFC 文档的目录索引。表 1-2 罗列了一些常见协议的 RFC 编号，具体内容参阅本书附录的“RFC”部分。

表 1-2　　常用协议与 RFC 文档号

协议	相关 RFC 文档号
TELNET	RFC854，RFC855
HTTP	RFC2068，RFC2069，RFC2109，RFC2145，RFC2169 等
FTP	RFC959
POP3	RFC1734，RFC1957
SMTP	RFC821，RFC1441
DNS	RFC2181，RFC2182，RFC2219，RFC2230 等

目前 Windows 系统中间使用较多的协议包括 IP、ICMP、ARP、TCP、UDP 以及一些应用层的 TELNET、FTP、HTTP、SMTP、POP3 等。这些协议所从属的 DoD 层不同，在程序中的实现也有很大的区别，如图 1-3 所示。

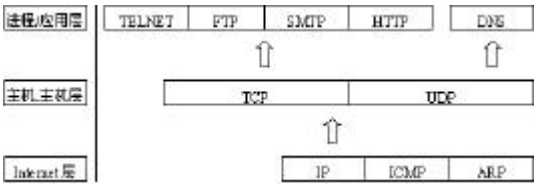

图 1-3　网络中常用协议以及层次关系

TELNET、FTP、SMTP、HTTP 等协议都是建立在 TCP 协议基础之上的，属于应用层的范畴。TCP 与 UDP 则是建立在 IP 基础上的。而 IP、ICMP 以及 ARP 等协议的层次就很低了，一般编程的时候极少涉及到这个层次，一般需要进行特殊处理或者是系统内部允许的时候才会对其进行操作，一般在 Windows 中进行的网络程序开发就到这个层次为止。若需要对更低的层次如数据链路层进行编程控制，就需要通过编制设备驱动程序（VXD）或者是利用 NDIS 接口才可以实现。如果达到这个层面，就可以检测流过网络适配器的所有数据包，通过编程分析就可以提取有效的信息，从而达到网络监听（Sniffer）、检测等目的。关于这方面的详细资料可以在网络上搜索到。

1.1.4　进程/应用层协议

进程/应用层的协议是平时使用最广泛的协议，这一层的每个协议都由客户程序和服务程序两部分组成。程序通过服务器与客户机的交互来工作。

1. TELNET 协议

TELNET 是网络虚拟终端协议的一个典型例子，该协议允许用户通过 Internet 登录远程计算机。客户程序需要自己实现 TELNET 协议，同时在某些键以及显示的特性上，不同终端类型的定义是不一样的，目前的大多数终端支持 DEC 的 VT100 终端类型。在实现 TELNET 协议时，工作量最大的还在于解决如何使自己的客户程序适应不同的终端类型这个问题上。

关于该协议的实现请参阅本书 “TCP 协议相关编程”中相关 TELNET 的部分。

2. FTP 协议

FTP 协议的全称是 File Transfer Protocol，即文件传输协议。在该协议中，要求使用者是经过授权的用户，从而使文件的访问具有一定的安全限制。由于 FTP 口令在网络上传输的信息是明文的，因此一旦被监听就会威胁到系统安全。许多站点也提供匿名 FTP 服务（Anonymous FTP），在这类站点上，通常的登录用户名为 anonymous，口令按照惯例是 guest，也有可能为用户的 E-mail 地址，当然提供一个假的地址也可以通过审查。

使用 FTP 协议可以在提供 FTP 功能的服务器上进行文件检索与传送等操作。经常使用的 FTP 客户端包括 CuteFtp、LeapFtp 等，还有众多的个人开发产品。

关于该协议的实现请参阅本书中“TCP 协议相关编程”的 FTP 的内容。

3. HTTP 协议

HTTP 协议的全称是 Hypertext Transfer Protocol，即超文本传输协议。它是 WWW 服务程序所用的协议，也是目前在 Internet 中使用最广泛的协议。著名的 HTTP 协议客户端浏览

软件包括 IE（Internet Explorer），以及 Netscape 公司的 Netscape Communicator 和 Netscape Navigator 等，还有众多如 Opera 这样的出色而小巧的 WWW 浏览器，这些小型的浏览器软件有很多是由个人开发的。

通过 WWW 方式，可以进行信息查询、文件下载、收发 E-mail、在线聊天等，功能非常强大。

关于该协议的实现请参阅本书中“TCP 协议相关编程”的有关 HTTP 的部分。

4. SMTP 与 POP3 协议

SMTP 是 Simple Mail Transfer Protocol 的缩写，即简单邮件传送协议。POP3 是 Post Office Protocol 3 的缩写，即邮局协议。通过这两个协议就可以实现收发 E-mail 的功能。

目前国内使用较广泛的电子邮件客户端软件 Foxmail 是由个人开发的，它所使用的开发工具就是 Delphi。

关于该协议的实现请参阅本书中“TCP 协议相关编程”的有关 SMTP 和 POP3 的内容。

5. DNS

DNS的全称是Domain Name Service，即域名服务。它实现了由主机名到IP地址的映射功能。

关于该协议的实现请参阅本书中“基本网络编程技术”的“获取 IP 地址”部分。

1.1.5 主机-主机层协议

主机-主机层协议的功能是建立并且维护连接，这些协议主要用于保证主机间数据传输的安全性。在这个层中定义了两个协议：TCP 和 UDP。

1. TCP 协议

TCP 的全称是 Transmission Control Protocol，即传输控制协议。在网络通信传输机制中，它属于“面向连接，可靠传输”的类型。这一点如果和 UDP 协议进行比较就会看得比较清楚。面向连接的传输意味着在进行通信以前，需要在两个系统之间建立逻辑连接，在每个数据传输的过程中都需要进行应答以保证数据包的完整。这种方法所需的网络开销较大，可是数据传输的可靠性可以保证。

在常见的上层协议中，TELNET、FTP、SMTP、HTTP 等都是使用 TCP 协议作为基础的，而一些简单的协议如 Echo（一种简单的回显服务，即客户端会从服务器端接受到自己的发送包的拷贝），则可以使用 TCP 也可以使用 UDP。

2. UDP 协议

UDP 的全称是 User Datagram Protocol，即用户数据报协议。它属于“面向无连接，不可靠传输”的类型。这一点必须注意。该协议只负责接收和传送由上层协议传递的消息，它本身不做任何的检测、修改与应答。

UDP 协议中，每个数据包成为“数据报”，它的包头只包括几个域，主要是地址信息、包的长度和校验信息。与此对应，TCP 包的头信息有十多个域，因此它的网络开销一般要小于 TCP 协议。

由于 UDP 协议在传送数据过程中没有建立连接，也不进行检查，因此在优良的网络环境中，其工作的效率较 TCP 协议要高。目前常见的使用 UDP 协议工作的软件有 OICQ，它是一种在线聊天工具。由于自身的特点，它成为进行网络广播的首选协议。传统的 NFS（Net

File System，网络文件系统）只使用 UDP 协议，从 NFS v30 开始同时支持 TCP 与 UDP。

1.1.6 Internet 层协议

Internet 层协议的主要功能是负责数据的传输，在不同网络和系统间寻找路由，分段和重组数据报文，另外还有设备寻址。

这一层中的协议包括 IP 协议、ICMP 协议、ARP 协议和 RARP 协议。

1. IP 协议

IP 协议全称是 Internet Protocol，即 Internet 协议。它负责在 TCP/IP 主机之间提供数据报服务，进行数据封装并产生协议头。该协议是 TCP 与 UDP 协议的基础。

由于在以太网中最大的帧大小是有限制的，因此 IP 协议需要将较大的数据报文分割，并在目标主机处将这些打散的包重新组合。由于不同的包段可能是由不同的网络路径传送的，因此 IP 协议还需要负责将这些包按正确的顺序重新组合。注意一点，IP 协议也不负责包的校验，它也是一种无连接不可靠传输。“不可靠”（unreliable）的意思是它不能保证 IP 数据报能成功地到达目的地。如果发生某种错误，IP 协议有一个简单的错误处理方法，丢弃该数据报，然后发送 ICMP 消息报给信源端。数据包的检测校验是由上层协议如 TCP 等负责的。“无连接”（connectionless）的意思是 IP 并不维护任何关于后续数据报的状态，每一个数据包都是独立的。

IP 协议还负责寻找路由，因此它还需要配套一个确定的 IP 地址。在 IP 报文的包头中包含了源和目的的 IP 地址。

一般来说不会有应用程序直接访问 IP 协议。

2. ICMP 协议

ICMP 协议全称是 Internet Control Message Protocol，即 Internet 控制报文协议。ICMP 协议其实是 IP 协议的附属协议，IP 协议用它来与其他主机或路由器交换错误报文和其他的一些网络情况。在 ICMP 包中携带了控制信息和故障恢复信息，这些信息的功能如下：

- 源抑制：这是一个流控制信息，由接收方在接收主机缓冲区快满时，请求源主机停止发送数据。
- 路径重定向：由网关向请求其提供服务的主机发送，用于通知该主机在网络中还有其他的距离目的主机更近的网关。
- 通知主机不可到达：在网络状况不佳的网络中传送数据报时，发生故障（如链路失效、链路堵塞、主机失效等）的网关或者系统会发出此消息。在 ICMP 报文中通常包括失效的原因。
- 应答与回复请求：用于 ping 来检测目标主机是否可以到达。ping 指令调用 ICMP 消息的应答请求功能发送数据报，如果远程主机是可以到达的，则该系统会用应答回复功能来响应。

ICMP 协议主要用于路由器或者主机向其他的路由器或者主机发送出错报文和控制信息。

尽管 ICMP 协议主要被 IP 协议使用，但应用程序也有可能访问它，在 Windows 系统中有专门的 DLL 接口可以使用。除去前面所说的 ping，Traceroute 是另外一个流行的诊断工具，它们都使用了 ICMP 协议。

3. ARP 协议

ARP 协议的全称是 Address Resolution Protocol，即地址解析协议。每一块网卡（Network Interface Card ，NIC）都有一个唯一的硬件地址（由网卡的生产厂商设置的，需要使用特殊的方式才可以修改），这个硬件地址称为 MAC（Medium Access Layer）。一块网卡依据数据帧的包头信息中是否写有它的 MAC 地址来决定是否接收并上传该帧。

分配给主机使用的 IP 地址和它固有的 MAC 地址是互不相干的。IP 地址只对 TCP/IP 协议有效，MAC 地址只对网络访问层有意义。在物理网络上的数据帧交换依赖于 MAC 地址，ARP 协议实现了从 IP 地址到 MAC 地址的映射。

4. RARP 协议

RARP 协议的全称是 Reverse Address Resolution Protocol，即逆向地址解析协议。它负责根据 NIC 硬件地址去查询对应的 IP 地址。

1.2 TCP/IP 基本概念介绍

在对 TCP/IP 中的常用协议以及它们在整个协议族中所处的位置有了总体了解之后，接下来将介绍一些与网络编程相关的 TCP/IP 基本概念。

1.2.1 IP 报文数据封装

在发送报文的过程中，各层协议都要对数据进行处理，在数据报文加上各自的控制信息后交给下层协议处理。这些添加的控制信息称为协议头，用于控制用户信息的传送。这个添加协议头的过程称为数据封装。

TCP/IP 协议使用协议头来实现与其他主机上的对等层进行通信。在目标主机上，每一层根据本层的协议头对数据包进行解释与处理，并去掉头信息提交上层处理。这与 OSI 模型中讲述的是一致的。

IP 协议是 TCP/IP 协议的基础，所有的 TCP/IP 协议都是通过收发 IP 数据报与网络中的对等协议实现通信的。理解 IP 包中的数据头的内容对于理解协议的实现非常有帮助。IP 数据报文格式如图 1-4 所示。

在 IP 数据报头中， Version 域指明了报文使用的 IP 版本号，目前使用的是 4，即 IP v4，IP v6 也即将推出，有关两者的讨论参见附录中 IP v6 的部分。IHL 域（网间网头长度），可选。TTL（Time To Live，生存时间）域由 IP 初始化，指明数据报最终到达目的地时经过的路由器的最大个数，ICMP 协议可以利用回应报文中的这个域作为有效的跳转计数来检测网络情况。TTL 的值每经过一个路由器就会减 1，一旦 TTL 变为 0，就会在下一个路由器被丢弃。TTL 域的这个特性可以避免数据报在网络中无限循环。Header Checksum 域用于保证信息头的完整性。IP 协议没有任何的错误检测机制与恢复机制，但是利用这个域它可以保证信息头中数据的完整性。若完整性检查失败，IP 就会将该数据报丢弃。源和目的的地址都是 32 位的，可以容纳 IP v4 的 4 节 8 位 IP 地址。

图 1-4　IP 报文格式

前面说过，所有的 TCP/IP 协议都使用 IP 数据报发送数据，为了使 IP 能够接收不同的数据报并交给相应的协议处理，在 IP 协议头中还包含了一个协议域（Protocol 域）。根据 TCP/IP 协议标准，给每一个协议都分配了一个协议号。TCP 协议的协议号是“6”，ICMP 的协议号是“1”，IP 协议的协议号是“0”（即该数据由 IP 自身来处理）。

1.2.2　IP 数据报的分段与重组

在图 1-4 中可以看到，IP 头的总长度域（Total Length 域）的长度是 16 个 bit，这意味着 IP 数据报的最大长度是 64KB，这个长度并不大。而且某些底层网络的访问技术还不允许在一个数据帧中容纳这么长的数据（如以太帧的长度规定最大不超过 1.514KB），在这种情况下，就需要对 IP 报文进行分段处理。

在数据段的传送过程中，很可能有一部分是通过不同的网络路径传送的，再考虑到数据段经过的网络上可能发生的拥堵现象和链路失效等情况导致的重发，很可能到达目标主机时已经完全打乱了发送的次序，需要重新进行排序。在 IP 协议头中包含的段偏移域（Fragment Offset 域）里就有段的顺序信息，IP 协议可以根据这个域中的值对网络中接收的数据段重新排序，并可以检测是否有报文丢失。直到所有的数据段都经过排序重组之后才会提交到相关的协议进行处理。

如果同时有两个甚至更多的数据段需要排序的话，就要使用标识域（Flags 域）的功能了，不同从属的数据段其标识域的值不同。

1.2.3　IP 地址与子网掩码

1.　IP 地址

IP 协议的功能是在网络间移动信息，这是通过检查网络层地址完成的。当以数据报形式在主机之间传递数据时，每个数据报的报头中有一个目的地址（Destination IP Address），这个地址可以准确地标识特定网络中的特定主机，而且决定了发送报文的路径。在 TCP/IP 中，该地址就称为 IP 地址。它是长度为 32 位的二进制数字，这 32 位的数字被组织成 4 个八位的二进制整数，中间使用“.”分隔。这种格式称为“dotted decimal notation”，直译为点分十进制标记法。IP 地址包括网络地址和主机地址。IP 地址的每一分隔部分定义了一个唯一的地址，其中一部分代表了网络 ID（netid，有时候可以用它选定一个子网），一部分代表了主机 ID（hostid）。

IP 地址分为 5 类：A 类、B 类、C 类、D 类和 E 类。

A 类地址：IP 地址第 1 位是 0，跟着的 7 位是网络地址，最后 24 位是主机地址。