# 第3章 网络体系结构与协议

**本章学习目标**

- 掌握计算机网络的体系结构。
- 掌握网络测试命令的使用。
- 熟练掌握IP地址的配置。
- 熟练掌握子网的划分方法。
- 掌握捕获信息工具Sniffer的使用。
- 掌握抓包工具Wireshark的使用。

计算机网络体系结构是计算机网络和它的部件所执行功能的精确定义，并用协议、实体逻辑环境等加以描述。网络各层次及其协议的组合，称为网络体系结构。网络体系结构应当具有足够的信息，以允许软件设计人员给每层编写实现该层协议的有关程序，这些程序应正确地遵循其对应的协议。网络体系结构和每层协议的确定是网络设计的基本课题。

## 3.1 协议分层

**1. 网络协议**

“协议”是计算机的网络语言。如果一台计算机不能使用某个协议，它就不能与使用该协议的其他计算机通信。

协议是一种通信规则。计算机网络中不同系统的两实体间只有在通信的基础上才能相互交换信息，共享网络资源。一般来说，实体是能发送和接收信息的任何对象，可以指用户应用程序、文件传送包、数据库管理系统、电子邮件设备和终端等。系统可包含一个或多个实体(如主机和终端等)。两个实体间要想进行正常的通信，就必须能够相互理解，共同遵守有关实体的某种互相都能接受的规则，我们把这些为进行网络中的数据交换而建立的规则的集合称为协议。

网络协议的作用是约束计算机之间的通信过程，使之按照事先约定好的步骤进行。一个网络协议主要由语法、语义和定时三个要素组成。

(1) 语法：规定通信双方信息与控制信息的结构或格式，包括数据格式、编码和信号等级等。

(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，包括数据的内容和含义等。

(3) 定时：规定事件顺序，确定通信过程中状态的变化，包括速率匹配和排序等。

**2. 网络体系结构**

计算机网络体系结构是网络层次结构模型与各层次协议的集合。根据协议之间的相互协作关系，把它们按层次结构进行组织，每个层次可以包含若干个协议，层与层之间定义了信息交互接口，使每个层次具有相对的独立性。位于某个层次中的协议既可以为上层协议提供服务，也可以使用下层协议提供的服务。具体应该划分多少层次，每个层次应该包含哪些协议，是研制网络技术时确定的，对于具体的网络技术这些问题都已确定。

采用分层结构组织网络协议的必要性如下所述。

(1) 每一层可以实现一种相对独立的功能，且不必知道相邻层是如何实现的，只要明确下层通过层间接口提供的服务是什么及本层向上层提供什么样的服务，就能独立地进行本层的设计。由于每一层只实现一种相对独立的功能，因而可将一个复杂问题分解为若干个比较容易处理的小问题。

(2) 系统的灵活性好。当某个层次的协议需要改动或替代时，只要保持它和上下层的接口不变，则其他层次都不受其影响。

(3) 每个层都可以采用最合适的技术来实现。

(4) 有利于标准化工作。每层的功能及其所提供的服务都已有了精确的说明，就像一个被标准化了的部件，只要符合要求就可以拿来使用。

**3. 协议和服务的关系**

为了进一步理解网络体系结构的概念，有必要明确协议和服务之间的关系。为方便理解，先介绍几个相关的名词：实体、服务访问点、服务原语。

(1) 实体：在研究计算机网络时，可以用实体抽象地表示任何可发送或接收信息的硬件或软件。实体究竟是一个进程还是一个计算机，对研究问题没有实质上的影响。但在多数情况下，实体通常是一个特定的软件模块。

(2) 服务访问点：在同一系统中相邻两层的实体进行交互(即交换信息)的地方，通常称为服务访问点(Service Access Point，SAP)。

(3) 服务原语：上层使用下层所提供的服务必须通过与下层交换一些命令来实现，这些命令称为服务原语(Service Primitive)。服务原语可被划分为四类，分别是请求(Request)、指示(Indication)、响应(Response)和确认(Confirm)。

需要注意的是由于实体含义的多样性，我们可以将体系结构中的一个层次看成是一个实体，对等实体则可以理解为两台计算机中相同的体系结构层次。

在协议的控制下，两个对等实体间的通信使得本层能够向上一层提供服务。要实现本层协议，还需要使用下面一层所提供的服务，体系结构中层与层之间的关系如图 3-1 所示。在对等层实体间传送的数据的单位都称为该层的协议数据单元(Protocol Data Unit，PDU)。

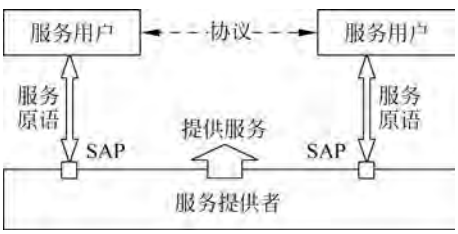


图 3-1　体系结构中层与层之间的关系

协议和服务是两个不同的概念。首先，协议的实现保证了本层能够向上一层提供服务。本层的服务用户只能看见下层提供的服务而无法看见下层的具体协议，即下层的协议对上层的服务用户是透明的。其次，协议是“水平的”，即协

议是控制对等实体之间通信的规则。而服务是“垂直的”，是由下层向上层通过层间接口提供的。另外，并非在一个层次内完成的全部功能都称为服务，只有那些能够被上层看得见的功能才能称为“服务”。层与层之间交换的数据的单位称为服务数据单元 SDU(Service Data Unit)。

## 3.2 体系结构与 OSI 模型

为了实现不同分层的网络体系结构之间的互联。为此，国际标准化组织(ISO)在 1984 年正式颁布了“开放系统互联大基本参考模型(Open System Interconnection，OSI)”国际标准，使计算机网络体系结构实现了标准化。

所谓开放系统互联，是指按照这个标准设计和建成的计算机网络系统可以互相连接，这个 OSI 参考模型对应于由两台主机组成的计算机网络。通信领域通常采用 OSI 的标准术语来描述系统的通信功能。

OSI 参考模型规定了一个网络协议的框架结构，如图 3-2 所示，自下而上分别是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。其中 1～3 层通常称为通信子网，负责提供网络服务，不负责解释信息的具体语义。5～7 层称为资源子网，它负责进行信息的处理，信息语义的解释等。第 4 层是传输层，它是上 3 层与下 7 层之间的隔离层，负责解决高层应用需求与下层通信子网提供的服务之间的匹配问题。

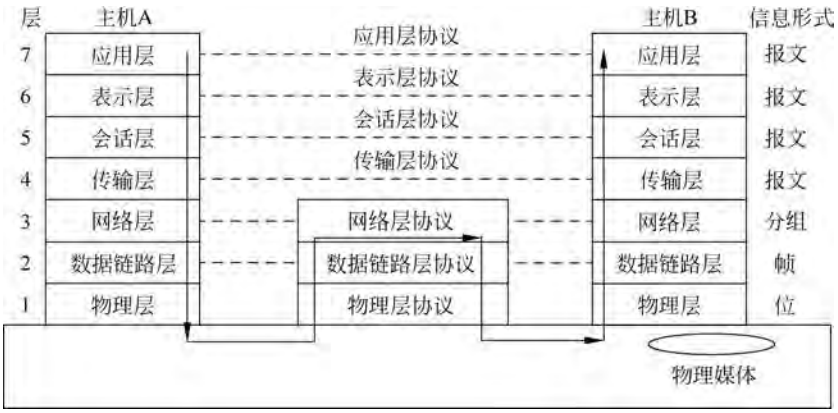



图 3-2 OSI 体系结构

### 3.2.1 物理层

物理层主要处理与物理传输介质有关的机械、电气、功能特性和接口问题。物理层与传输媒体直接相连，因此也称为物理层接口，是计算机与网络连接的物理通道。其功能是控制计算机与传输媒体的连接，即可以建立、保持和断开这种连接，以保证比特流的透明传输。物理层传送的数据基本单位是比特(b)，又称位。

传送数据所使用的物理媒体，如双绞线、同轴电缆、光缆等，不属于物理层。

### 3.2.2 数据链路层

数据链路层位于OSI参考模型中的第2层，它以物理层为基础，向网络层提供可靠的服务，因此要求数据链路层能够建立和维持一条或多条没有数据发送错误的数据链路，并在数据传输完毕后能够释放数据链路。

实际上，“链路”与“数据链路”并不是一回事。“链路”就是一条无源的点到点的物理线路段，中间没有任何其他的交换节点。一条链路只是一条通路的一个组成部分，而“数据链路”则是指当需要在一条线路上传送数据时，除了必须有一条物理线路外，还必须有一些必要的规程来控制这些数据的传输。因此，数据链路就是在链路上加上了实现这些规程的硬件和软件后构成的。当采用复用技术时，一条实际的物理链路上可以有多条数据链路。我们有时又把链路称为物理链路，把数据链路称为逻辑链路。

数据链路层最重要的作用就是通过一些数据链路层协议（即链路控制规程），在不太可靠的物理链路上实现可靠的数据传输。为此，数据链路层完成的主要功能有以下7个方面。

（1）链路管理。链路管理就是进行数据链路的建立、维持和拆除。当网络中的两个节点要进行通信时，数据的发送方必须确切知道接收方是否已经处于准备接收的状态。通信双方必须要交换一些必要的信息，也就是必须先建立一条数据链路。同时，在传输数据时要维持数据链路，而在通信完毕时要拆除数据链路。

（2）帧同步。在数据链路层，数据的传送单位是帧。数据一帧一帧地传送，就可以在出现差错时，将有差错的帧再重传一次，从而避免了将全部数据都进行重传。因为物理层上交给数据链路层的是一串比特流，所以帧同步是指接收方应当能从收到的比特流中准确地区分出一帧的开始和结束的地方。

（3）流量控制。为防止双方速度不匹配或接收方没有足够的接收缓存而导致数据拥塞或溢出，数据链路层要能够调节数据的传输速率，即发送方发送数据的速率必须使接收方来得及接收。当接收方来不及接收时，就必须及时控制发送方发送数据的速率。

（4）差错控制。数据链路层必须要配备一套检错和纠错的规程，以防止数据帧的错误、丢失与重复。在计算机通信中，广泛采用两类编码技术用于纠错。一类是前向纠错，即接收方收到有差错的数据帧时，能够自动将差错改正过来。这种方法的开销较大，不适合于计算机通信。另一类是检错重传，即接收方可以检测出收到的帧中有差错，但并不知道是哪几个比特错了，于是让发送方重复发送这一帧，直到收方正确收到这一帧为止。这种方法在计算机通信中是最常用的。

（5）区分数据和控制信息。在多数情况下，数据和控制信息处于同一帧中，因此一定要有相应的措施使接收方能够将它们区分开来。

（6）透明传输。所谓透明传输就是不管所传数据是什么样的比特组合，都应该能够在链路上传送。当所传数据中的比特组合恰巧出现了与某一个控制信息完全一样的情况时，必须采取适当的措施，使接收方不会将这样的数据误认为是某种控制信息。这样才能保证数据链路层的传输是透明的。

（7）寻址。必须保证每一个帧都能发送到正确的目的站。接收方也应知道发送方是哪个站。

其中，差错控制和流量控制是数据链路层的两个重要的功能，最简单也是最基本的就是

采用停止等待协议。

### 3.2.3 网络层



网络层是OSI参考模型中的第3层，介于传输层和数据链路层之间，它在数据链路层提供的两个相邻节点之间数据帧的传送功能上，解决整个网络的数据通信，将数据设法从源端点经过若干中间节点传送到目的端，从而向传输层提供最基本的端到端数据传送服务。

网络层的数据传送单位称为分组或包，通信子网及广域网的最高层就是网络层，因此网络层的主要作用是控制通信子网正常运行，以及解决通信子网中分组转发和路由选择等问题。网络层的主要功能如下。

(1) 分组转发和路由更新。网络层要逐个节点地把分组从源站点转发到目的站，而转发的路由不是一成不变的，网络层执行某种路由算法，根据当前网络流量及拓扑结构的变化动态地更新路由表，进行路由选择。

(2) 网络连接的建立和管理。提供虚电路和数据报两种分组传输服务，这两种服务分别是面向连接服务和无连接服务方式。

(3) 防止通信子网信息流量过大造成拥塞。若注入网络的分组太多，在某段时间，如果对计算机网络中的链路容量、交换节点的缓存及处理机等某一资源的需求超过了该资源所能提供的能力，网络的性能就要变坏，甚至大幅度下降，这种情况就叫作拥塞。网络层可以采用预先分配缓存资源、允许节点在必要时丢弃分组、限制进入通信子网的分组数等方法进行拥塞控制。

### 3.2.4 传输层

有了前面的3个层次，网络的功能已经比较完善，但对不懂网络的人使用起来却很不方便，因此需要一个层次为用户提供简洁的网络服务界面。有了传输层，高层用户就可利用传输层的服务直接进行主机到主机的数据传输，从而不必关心通信子网的更替和技术的变化，复杂的通信细节被传输层所屏蔽。传输层通常为高层用户提供两种服务，即可靠的数据交付和尽最大努力的数据交付。此外传输层还具有复用功能，可以同时为一台计算机中的多个程序提供通信服务。传输层数据传送的基本单位是报文段。

传输层是计算机网络中的资源子网和通信子网的接口和桥梁，完成资源子网中两节点间的直接逻辑通信。

传输层下面的3层属于通信子网，完成有关的通信处理，向传输层提供网络服务；传输层上面的3层完成面向数据处理的功能，为用户提供与网络之间的接口。由此可见，传输层在OSI/RM中起到承上启下的作用，是整个网络体系结构的关键。

传输层完成的主要功能有以下4点。

(1) 分割和重组报文。

(2) 实现通信子网端到端的可靠传输(保证通信的质量)。

(3) 传输层的流量控制。

(4) 提供面向连接的和无连接数据的传输服务。

### 3.2.5 会话层

会话层允许不同机器上的用户建立会话(Session)关系。会话层的任务就是提供一种有效的方法,以组织并协商两个表示层进程之间的会话,并管理它们之间的数据交换。具体地说,就是发信权的控制与同步的确定方法等。

会话层的主要功能如下。

(1) 提供一个面向用户的连接服务,并为会话活动提供有效的组织和同步所必需的手段,为数据传送提供控制和管理。

(2) 把会话地址变换成它的传送地址,建立会话连接。

### 3.2.6 表示层

表示层主要完成某些特定的功能。同时表示层为应用层(或用户进程)提供服务,该服务可以解释所交换数据的意义,进行正文压缩及各种变换,以便用户使用,如对数据格式和编码的转换,以及数据结构的转换等。如把 ASCII 变换为 EBCDIC 码等。当网络中使用不同的计算机代码和文件格式,以及各种不兼容终端(用不同字符、不同显示长度、屏幕显示行数、行结束符号和光标寻址方法)等时,均可在这一层进行转换。此外,利用密码对正文进行加密、保密也是该层的任务。表示层的功能通常由用户调用库子程序来实现。

表示层主要完成的功能有以下 5 点。

(1) 对数据编码格式进行转换。

(2) 数据压缩与恢复。

(3) 数据的安全与加密等工作。

(4) 建立数据交换格式。

(5) 其他特殊服务。

### 3.2.7 应用层

应用层为应用进程提供访问 OSI 环境的方法。例如,世界上有成百不兼容的终端类型,如果希望一个全屏幕编辑程序能工作在网络中许多不同的终端类型上几乎是不可能的。解决这一问题的方法之一是,定义一个抽象的网络虚拟终端,编辑程序和其他所有程序都面向该虚拟终端。而对每一种终端类型,都写一段软件将面向虚拟终端的编辑程序变换为面向实际终端的软件,控制实际终端的运行。

这一层的例子有虚拟终端协议、虚拟文件协议、文件传送协议、公共管理信息协议。虚拟终端协议是用来提供给终端使它能访问远程系统中的用户进程。虚拟文件协议提供对文件的远程访问、管理和传送。文件传送协议是在两个终端之间提供文件传送服务。公共管理信息协议通过提供的 7 项基本服务支持对网络的性能管理、故障管理和配置管理服务。

应用层是 OSI 参考模型的最高层,是利用网络资源向应用程序直接提供服务的层次。与传输层不同,应用层提供的是特殊的网络应用服务,如邮件服务、文件传输服务等。用户可直接使用,也可以在此基础上开发更高级的网络应用。

应用层完成的主要功能有。

(1) 作为用户应用程序与网络间的接口。

(2) 使用户的应用程序能够与网络进行交互式联系。

在 OSI 7 层模型中,每一层都提供了一些明确的网络功能。一般数据通信子网中的交换节点包含 OSI 模型的下 3 层,表示节点的这 3 个层次被称为中继开放系统。

若从功能角度看,下面 4 层主要提供通信传输功能,以节点到节点之间的通信为主;高层协议(会话层、表示层和应用层)则以提供用户与应用程序之间的处理功能为主。或者说,低 4 层协议完成通信功能,高 3 层完成处理功能。

若从产品看,低 3 层协议一般由硬件完成,高层由软件完成。如网卡、集线器实现物理层功能;网桥(交换机)实现链路层功能;路由器实现网络层功能;而电子邮件软件则完成应用层的功能。

## 3.3 TCP/IP 体系结构

TCP/IP(Transmission Control Protocol/Internet Protocol)传输控制协议/网际协议参考模型是当今计算机网络领域所使用的专用模型(或称为体系结构),其目的是将各种异构计算机网络或主机通过 TCP/IP 实现互联互通,TCP/IP 提供了一个开放的环境,能够把各种计算机平台,包括大型机、小型机、工作站和微型计算机很好地连接到一起,从而达到不同网络系统互联的目的。

**1. TCP/IP 协议簇**

TCP/IP 模型与 OSI 参考模型并不完全一致,但两个标准之间存在一定的兼容。OSI 参考模型由 ISO 和 ITU 统一发布标准的文件来规定这一层次数量及每一层次的名称及其功能,而 TCP/IP 模型没有官方标准的文件加以统一规定。一般来说 TCP/IP 可以分为 4 层,即应用层、运输层、网际层、网络接口层,TCP/IP 和 OSI 参考模型的对应关系,如图 3-3 所示。

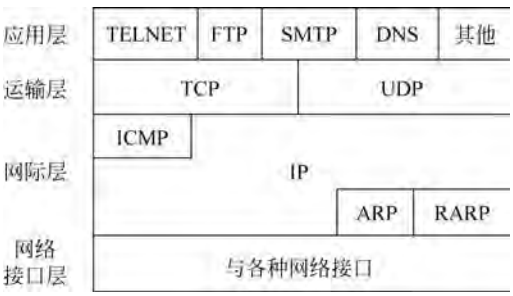



图 3-3 TCP/IP 协议簇

(1) 网络接口层。

网络接口层提供了 TCP/IP 与各种物理网络的接口,为数据报的传送和校验提供了可能。这些物理网络包括各种局域网和广域网。也为在其之上的网际层提供服务。

(2) 网际层。

网际层也叫网络互联层,它是整个体系结构的关键部分。这一层以 IP 为标志,提供基于 IP 地址的、不可靠的、尽最大能力的、面向无连接的数据传送服务。主要有 4 个协议:

IP、ICMP、ARP 和 RARP。其中，IP 是最主要的网际层协议，用于网络互联；Internet 控制报文协议(ICMP)主要用于报告差错，向主机和路由器发送差错报文；地址解析协议(ARP)用来将 IP 地址解析成物理地址；逆地址解析协议(RARP)将物理地址解析到 IP 地址。

(3) 运输层。

运输层的作用与 OSI 参考模型中传输层的作用是一样的，即在不可靠的互联网络上，实现可靠的端对端的数据传送，允许具有相同 IP 地址的不同机器独立地接收和发送数据。所以传输层弥补了网络层得到的服务和用户对服务质量的要求之间的差距。

IP 提供无连接的不可靠服务，需要具有良好差错控制功能的传输控制协议来保证端对端的数据传输质量。在 TCP/IP 中，传输层定义了 3 个端到端的协议，即传输控制协议(TCP)、用户数据报协议(UDP)和差错与控制报文协议(ICMP)。

TCP 提供面向连接的端到端的可靠数据传送，根据协议内容，把用户数据分成 TCP 数据段进行发送，在接收端按顺序号进行重组，恢复原来的用户数据信息，TCP 的主要功能是差错校验、出错重发和顺序控制等，以保证数据的可靠传送，减少端到端数据传输误码率。

UDP 提供了基本的错误检查特性，UDP 是面向无连接的协议。UDP 服务的优点是避免在面向连接的通信中所必需的连接建立和连接释放的过程，避免额外开销的增加，有助于提高速度。

在传输的分组发生错误或出现丢失时，利用 ICMP 发送出错信息给发送端，其次在分组流量过大时，通过 ICMP 还可以实现流量控制。

(4) 应用层。

应用层是为用户提供各种应用服务，它包含所有的高层协议。应用层服务是由应用层软件来提供的，应用层软件是由各种应用软件模块组成的，TCP/IP 集中提供的应用服务主要有以下 5 种。

① Telnet(远程注册协议)，它是指允许一台计算机登录到远程的计算机上并且进行工作。

② FTP(文件传送协议)，在服务器和客户机之间用两台计算机之间传送文件。

③ SMTP(电子邮件协议)，在两个用户之间传送电子邮件。

④ HTTP(超文本传输协议)，发布和访问具有超文本格式的信息。

⑤ SNMP(简单网络管理协议)，对 TCP/IP 网络进行管理。

**2. TCP/IP 的特点**

与 OSI 相比，TCP/IP 具有很多不同之处。

TCP/IP 一开始就考虑到各种异构网络的互联问题，将网际协议 IP 作为 TCP/IP 的重要组成部分。但 OSI 制定时最初只考虑到全世界都使用一种统一的标准将各种不同的系统互联起来。

TCP/IP 一开始就采取面向连接服务和无连接服务并重的原则，并在网际层使用无连接服务，但 OSI 在开始时各个层都采用面向连接服务，降低了效率。

TCP/IP 在较早时就有了较好的网络管理功能，但 OSI 到后来才开始考虑。

TCP/IP 的不足主要在于 TCP/IP 模型对“服务”“协议”和“接口”等概念没有很清楚地区分开，TCP/IP 模型的通用性较差。此外，TCP/IP 的网络接口层严格来说并不是一个层次，而仅仅是一个接口。

# 3.4 IP

Internet(因特网)是一个全球范围的、由众多网络连接而成的互联网,所使用的协议是TCP/IP。其中网际协议(IPv4)是用于互联许多计算机网络进行通信的协议,因此这一层也常常称为网际层,或IP层。

## 3.4.1 IP地址及子网掩码

网络互联的目标是提供一个无缝的单一的通信系统。为达到这个目标,互联网协议必须屏蔽物理网络的具体细节并提供一个大虚拟网的功能。虚拟互联网操作上要像任何网络一样,允许计算机发送和接收信息。互联网和物理网的主要区别是互联网仅仅是设计者想象出来的抽象物,完全由软件产生。设计者可在不考虑物理硬件细节的情况下选择地址格式、分组格式和发送技术。因此编址是使互联网成为单一网络的一个关键组成部分。

**1. IP地址**

为了以一个单一的统一系统出现,所有主机必须使用统一编址方案,但物理网络地址却不能满足这个要求。因为一个互联网可包括多种物理网络技术,每一种技术有自己的地址格式。这样,两种技术采用的地址因为长度不同或格式不同而不兼容。因而网际层的IP定义了一个与底层物理地址无关的编址方案,这就是IP层的地址,称为IP地址。这样任何在互联网上要进行通信的计算机都要使用IP地址,发送方把目的地IP地址放在分组中,将分组传送给网络,由协议软件来发送。统一编址有助于产生一个虚拟的、大的、无缝的网络,因为它屏蔽了下层物理网络地址的细节。两个应用程序的通信不需要知道对方的硬件地址。

那么,什么是IP地址呢?IP地址就是给每一个连接在Internet上的主机分配一个在全世界范围是唯一的32位二进制数。IP地址在进行编址时是采用两级结构的编址方法,因为两级层次结构设计使得在Internet上能够很方便地进行寻址。每个32位IP地址被分割成前后两部分。IP地址的前半部分确定了计算机从属的物理网络,后半部分确定了网络上一台单独的计算机。互联网中的每一个物理网络都被分配了唯一的值作为网络号,网络号在从属于该网络的每台计算机IP地址中作为前缀出现,而同一物理网络上给每台计算机分配一个唯一的主机编号作为IP地址的后缀。可以用下列公式表示IP地址。

IP地址=网络号+主机号

IP地址的层次结构使以下两点得到了保证。

(1) 每台计算机都被分配一个唯一的地址(即一个地址从不分配给多台计算机)。

(2) 虽然网络号分配必须全球一致,但主机号可本地分配,不需要全球一致。

第二点说明了在保证没有两个物理网络被分配为同一网络号的情况下,同一个主机号是可以在多个网络上使用的,但同一网络中不能有两台计算机具有相同主机号。例如,一个互联网包含三个网络,它们被分配的网络号为1、2、3,挂接在网络1的三台计算机可分配主机号为1、2、3。同时挂接在网络2的三台计算机也可被分配的主机号为1、2和3。

**2. IP地址分类**

当决定了IP地址的长度并把地址分为两部分后,就必须决定每部分应该包含多少位。

前缀部分需要足够的位数以允许分配唯一的网络号给互联网上的每一个物理网络。后缀部分也需要足够位数以允许从属于一个网络的每一台计算机都分配一个唯一的主机号。这不是简单地进行选择就可以，因为一部分增加一位就意味着另一部分减少一位。选择大的前缀适合大量网络，但限制了每个网的大小；选择大的后缀意味着每个物理网络能包含大量计算机，但限制了网络的总数。

因为互联网可以包括任意的网络技术，所以一个互联网可由一些大的物理网络构成，同时也可能由一些小的网络构成，而且单个互联网还可能是一个包含大网络和小网络的混合网。因此 IP 编址选择了将地址分类的方案，使之能满足大网和小网的组合。IP 地址分成了五类，即 A 类到 E 类。A、B、C 三类称为基本类，每类有不同长度的网络号和主机号，它们用于主机地址。地址的前几位决定了所属类别，并确定地址的剩余部分如何进行划分网络号和主机号。如图 3-4 所示为 IP 地址的五种类型。


图 3-4　IP 地址的类型

网络号字段。A 类、B 类和 C 类地址的网络号字段分别为 1 字节、2 字节和 3 字节长，由网络号字段最前面的 1～3 位区别类别，其数值分别规定为 0、10 和 110。

主机号字段。A 类、B 类和 C 类地址的主机号字段分别为 3 字节，2 字节和 1 字节长。

D 类地址是组播地址，主要留给 Internet 体系结构委员会 IAB 使用。E 类地址保留在今后使用。目前大量使用的 IP 地址仅 A 类、B 类、C 类 3 种。

32 位二进制的 IP 地址对于输入或读取是非常不方便的，因此常常用一种更适合人们习惯的表示方法，称为点分十进制表示法。其做法是将 32 位数中的每 8 位分为一组，每组用一个等价十进制数表示，在各个数字之间加一个点。如图 3-5 表示了一些 32 位地址与等价点分十进制数表示的例子。

| 32位二进制IP地址 | | | | 等价的点分十进制数 |
|---|---|---|---|---|
| 10000001 | 00110100 | 00000110 | 00000000 | 129.52.6.1 |
| 11000000 | 00000101 | 00110000 | 00000011 | 192.5.48.3 |
| 00001010 | 00000010 | 00000000 | 00100101 | 10.2.0.37 |
| 10000000 | 00001010 | 00000010 | 00000011 | 128.10.2.3 |
| 10000000 | 10000000 | 11111111 | 00000000 | 128.128.255.0 |

图 3-5　IP 地址的 32 位表示与等价点分十进制表示

点分十进制数表示法把每一组作为无符号整数处理，最小可能值为0（当8位都为0时），最大可能值为255（当8位都为1时）。所以点分十进制数的地址范围为0.0.0.0～255.255.255.255。

表3-1给出了A、B、C三类IP地址的使用范围。

**表3-1　IP地址的使用范围**

| 网络类别 | 第一字节数值范围 | 第一个可用的网络号 | 最后一个可用的网络号 | 最大网络数 | 每个网络中最大主机数 |
|---|---|---|---|---|---|
| A类 | 1～126 | 1 | 126 | 126 | 16 777 214 |
| B类 | 128～191 | 128.1 | 191.254 | 16 382 | 65 534 |
| C类 | 192～223 | 192.0.1 | 223 255.254 | 2 097 150 | 254 |

**3. 特殊IP地址**

观察表3-1，可以发现有些地址没有包含在内。那就是一些特殊地址，称为保留地址，而且特殊地址从不分配给主机。表3-2列出了特殊IP地址。

**表3-2　特殊IP地址**

| 网络号字段 | 主机号字段 | 地址类型 | 用　途 |
|---|---|---|---|
| 全0 | 全0 | 本机 | 启动时作为本主机地址 |
| 网络号 | 全0 | 网络地址 | 标识一个网络 |
| 网络号 | 全1 | 直接广播 | 在特定网络上广播 |
| 全1 | 全1 | 有限广播 | 在本地网络上广播 |
| 127 | 任意 | 回送 | 本地软件回送测试 |

（1）IP地址保留主机地址为全0表示一个网络。因此，地址168.121.0.0表示一个B类网络。网络地址指网络本身而不是指连到那个网络的主机，所以网络地址不能作为目的地址出现在分组中。

（2）在网络号后面跟一个所有位全1的主机号，便形成了网络的直接广播地址。当这样的地址作为一个分组的目的地址时，该分组就会被发送到该网络，并被发给网络中的所有主机。为了确保每个网络具有直接广播，IP保留包含全1的主机号。管理员不能分配全0或全1主机号给一个特定计算机，否则会导致错误发生。

（3）有限广播是指在一个本地物理网的一次广播。这种地址的所有位都是1。

（4）本机地址是用于当计算机刚启动还不知道自身IP地址的情况下，把它作为自身的源地址，去网络获得它的IP地址。

（5）当网络号为127，主机号为任意时，它是一个用于测试网络应用程序的回送地址。经常使用的形式是127.0.0.1。

**4. IP地址特点**

IP地址具有以下一些特点。

（1）IP地址不能反映任何有关主机位置的地理信息。

（2）当一个主机同时连接到两个网络上时，该主机必须同时具有两个IP地址，其网络号不同。这种主机称为多接口主机（如路由器）。

（3）由于IP地址中有网络号，因此IP地址不仅仅是标识一个主机（或路由器），而是指

明了一个主机(或路由器)和一个网络的连接。

(4) 按 Internet 的观点,用中继器或网桥连接的若干个局域网仍为一个网络,因此这些局域网都具有同样的网络号。

图 3-6 画出了两个路由器将 3 个局域网互联起来,其中局域网 LAN2 由两个局域网通过网桥 B 互联。每台主机以及路由器都需分配 IP 地址。

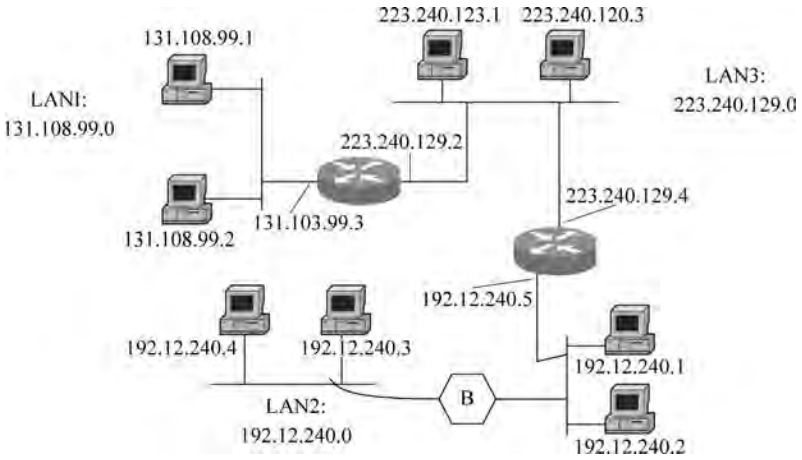



图 3-6 IP 地址的分配例子

通过图 3-6 所示的例子,我们应当注意以下 3 点。

(1) 与某个局域网相连接的计算机或路由器的 IP 地址,它们的网络号都必须一样。

(2) 用网桥互联的局域网仍是一个局域网,由网桥互联起来的主机都有同一个网络号。

(3) 路由器总是具有两个或两个以上的 IP 地址。

**5. 子网掩码**

为了更好地利用 IP 地址的资源,目前 Internet 都采用子网划分的方式。子网划分是把单个网络细化为多个规模更小的网络的过程,使得多个小规模物理网络可以使用路由器互联起来并且共有同一个网络号。

一个网络中的所有主机必须有相同的网络号,一个单位分配到的 IP 地址是 IP 地址的网络号,而后面的主机号则由本单位进行分配。本单位所有的主机都使用同一个网络号。但如果一个单位的主机很多,且分布在较大的地理范围,则往往需要在几个地区构建物理网络。如果考虑到需要使用同一个网络号,就需要用网桥将这些主机互联起来。但网桥的缺点很多,而且很容易引起广播风暴,并且当网络出现故障时不太容易隔离和管理,因此需要用路由器将几个物理网络互联起来。而路由器连接的两个网络必须具有不同的网络号,因而也需要采用子网划分的办法。要注意的是,划分子网只是单位内部的事,而在外部看来仍像任何一个单独网络一样只有一个网络号。

划分子网时,是用 IP 地址中的主机号字段中的前若干位作为“子网号字段”,后面剩下的仍为主机号字段,如图 3-7 所示,这样 IP 地址就进一步被划分为三部分,以支持子网的划分,分别为网络号字段、子网号字段和主机号字段。在原来的 IP 地址模式中,网

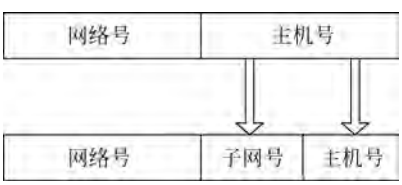



图 3-7 IP 地址进一步划分出子网号

络号部分就标识一个独立的物理网络，引入子网模式后，网络号部分加上子网号才能全局唯一地标识一个物理网络。也就是说，子网的概念延伸了 IP 地址的网络部分。子网编址使得 IP 地址具有一定的内部层次结构，便于分配和管理。

这样通过把 IP 地址的主机号字段划分成两个字段，就能够建立子网地址。这时必须用子网掩码来确定如何进行这样的划分，即要确切指出子网号字段需要多少位进行编码。子网中的每台主机都要指定子网掩码，而且子网中的所有主机必须配置相同的子网掩码。

如何用子网掩码来确定子网的划分呢？TCP/IP 体系规定由一个 32 位二进制的子网掩码来表示子网号字段的长度。子网掩码由一连串的 1 和一连串的 0 组成，对应于网络号和子网号字段的所有位都为 1，1 必须是连续的，或者说连续的 1 之间不允许有 0 出现，对应于主机号字段的所有位都为 0。那么使用一个什么子网掩码，如何划分子网号字段的位数，就取决于具体的需要。例如，一个 B 类的网络地址 172.16.0.0，如果要划分出 62 个子网，则子网号字段的位数要有 6 位，这样 $2^6-2$ 就可以划分出 62 个子网（去掉全 0 和全 1 的子网号）。因此要使用子网掩码 11111111 11111111 11111100 00000000 进行划分子网。这时第一个子网可使用的 IP 地址为 172.16.4.1～172.16.7.254，第二个子网可使用的 IP 地址为 172.16.8.1～172.16.11.254，依次类推。子网掩码一般也采用点分十进制数表示，如该例中的子网掩码可表示成 255.255.252.0。

如果网络没有被划分为子网，那么就要用默认子网掩码。要注意的是，即使网络没有被划分为子网，所有主机也必须要有一个子网掩码。默认的子网掩码中 1 的长度就是网络号的长度。因此，对于 A、B、C 三类 IP 地址，其对应的子网掩码默认值如表 3-3 所示。

**表 3-3　A、B、C 三类 IP 地址的默认子网掩码**

| 地 址 类 型 | 点分十进制数表示 | 二进制数表示 |
|---|---|---|
| A | 255.0.0.0 | 11111111 00000000 00000000 00000000 |
| B | 255.255.0.0 | 11111111 11111111 00000000 00000000 |
| C | 255.255.255.0 | 11111111 11111111 11111111 00000000 |

若子网掩码用点分十进制数表示，如 255.255.240.0，为了知道对应的子网号字段的长度，往往需要将它转化为二进制数。在子网掩码中使用到的数字实际上只有几个，所以只要记住表 3-4 中列出的几个数值转换关系，就可以很自如地使用子网掩码了。

划分子网后，根据子网掩码可以得出能够划分出多少个子网，以及每个子网中最多可以分配的主机号。例如，一个 B 类网络，使用的子网掩码是 255.255.248.0，则首先将子网掩码转换成二进制数，即 11111111 11111111 11111000 00000000，这样就得出子网号字段是 5 位，主机号字段为 11 位，因此最多可有 $2^5-2=30$ 个子网，这里减 2 是去掉全 0 和全 1 的两个地址，每个子网中最多可有 $2^{11}-2=2\,046$ 个供分配的主机号。

**表 3-4　十进制与二进制的转换**

| 十　进　制 | 二　进　制 | 十　进　制 | 二　进　制 |
|---|---|---|---|
| 128 | 1 0000000 | 248 | 11111000 |
| 192 | 11000000 | 252 | 11111100 |
| 224 | 11100000 | 254 | 11111110 |
| 240 | 11110000 | 255 | 11111111 |

**注意**：有些网络可以将全 0 用作子网地址，但应该避免这样做，除非确信所有的路由器都支持这个特性。

前文提过，划分子网后网络地址的概念就延伸了，那么已知一个 IP 地址和子网掩码，如何得到地址呢？答案就是 IP 地址和子网掩码进行“AND(与)”运算，所得的结果就是网络号。例如，IP 地址为 156.36.20.68，子网掩码为 255.255.255.224，要进行 AND 运算，首先把这两个数换成二进制数表示形式，然后对每一位进行 AND 操作，即可得到网络地址，如图 3-8 所示。

| | 网络号 | | | 主机号 |
|---|---|---|---|---|
| 156.36.30.68 | 10011100 | 00100100 | 00010100 | 010 00100 |
| AND<br>255.255.255.224 | 11111111 | 11111111 | 11111111 | 111 00000 |
| | 10011100 | 00100100 | 00010100 | 010 00000 |
| 网络地址 | 156 | 36 | 30 | 64 |

图 3-8　网络号的计算

**6. 专用互联网可用的 IP 地址**

如果一台计算机必须要访问 Internet 的资源，才需要从 Internet 地址管理机构申请注册 IP 地址。但对于没有与 Internet 连接的专用网络来说，就没有必要注册网络地址，管理员可以使用他们想用的任何 IP 地址，只要在同一个网络中没有重复的地址即可。但是如果该网络上的任何一台计算机采用某种方式与 Internet 连接起来，那么就可能发生该网络中的一个内部地址与 Internet 上的注册地址之间的冲突。

为了防止出现这种冲突，Internet 规定了 3 个地址范围，用于专用网络。这些地址不分配给任何 Internet 上的注册网络，因此可以供任何单位内部专用网络使用。这 3 个地址范围表示如下。

10.0.0.0～10.255.255.255　　1 个 A 类地址

172.16.0.0～172.31.255.255　　16 个连续的 B 类地址

192.168.0.0～192.168.255.255　　255 个连续的 C 类地址

**7. 超网**

B 类的网络地址资源有限，因此 Internet 又提供了一种可以将多个 C 类网络合并为一个逻辑网络的方法，这就是无类域间路由 CIDR 技术。超网技术是将多个 C 类地址合并，要求合并的 C 类地址必须具有相同的高位，也就是说要合并的必须是一些连续的地址，此时子网掩码被缩短，以便将 C 类地址网络字段中的部分位变成主机字段，并使得这些被合并的 C 类地址都在一个子网中。

现用一个例子来说明上面所述。例如，现在有如下 8 个连续的 C 类网络地址。

192.168.168.0

192.168.169.0

192.168.170.0

192.168.171.0

192.168.172.0

192.168.173.0

192.168.174.0

192.168.175.0

使用子网掩码 255.255.248.0 将这些地址合并，按照子网掩码划分 IP 地址的方式，此时每一个 IP 地址都属于相同的子网，如表 3-5 所示。

**表 3-5　使用子网掩码合并的 C 类网络地址**

| 地　　址 | 使 用 的 位 |
|---|---|
| 掩码 255.255.248.0 | 11111111 11111111 11111000 00000000 |
| 192.168.168.0 | 11000000 10101000 10101000 00000000 |
| 192.168.169.0 | 11000000 10101000 10101001 00000000 |
| 192.168.170.0 | 11000000 10101000 10101010 00000000 |
| 192.168.171.0 | 11000000 10101000 10101011 00000000 |
| 192 168.172.0 | 11000000 10101000 10101100 00000000 |
| 192.168.173.0 | 11000000 10101000 10101101 00000000 |
| 192.168.174.0 | 11000000 10101000 10101110 00000000 |
| 192 168.175.0 | 11000000 10101000 10101111 00000000 |

观察表 3-5，我们发现这些地址的高 21 位是一样的(11000000 10101000 10101)，从而有效地创建了一个 21 位的网络号，因此如 IP 地址为 192.168.168.12 的主机和 IP 地址为 192.168.174.3 的主机就在相同的子网中。

## 3.4.2　地址转换协议

当主机中有数据需要在网络上进行传送时，IP 会将数据放进分组中，在分组中包含了源 IP 地址和目的 IP 地址。每个主机或路由器通过分组中的目的 IP 地址来选择传送此数据的下一站。一旦下一站确定了，协议就通过网络将数据传送给选定的主机或路由器。为了提供一个独立大网络的假象，协议利用 IP 地址转发分组时，下一站地址和分组的目的地址都是 IP 地址。但是在通过物理网络硬件传送帧时，却不能使用 IP 地址，因为硬件并不懂 IP 地址，因而通过物理网络传送帧时必须使用硬件的帧格式，即帧中的硬件地址。因此，在传送帧之前，必须将下一站的 IP 地址翻译成等价的硬件地址。

将一台计算机的 IP 地址翻译成等价的硬件地址的过程就叫地址转换，这个转换是由地址解析协议(ARP)来完成的。

由于 IP 地址有 32 位，而局域网的硬件地址是 48 位，因此它们之间不是一个简单的转换关系。而且在一个网络上可能经常会有新的计算机加入进来，也会撤走一些计算机。另外，更换计算机的网卡也会使其硬件地址发生改变。可见在计算机中应存放一个从 IP 地址到硬件地址的转换表，并且能够经常动态更新。ARP 很好地解决了这些问题。

每一个主机都应有一个 ARP 高速缓存(ARPcache)，里面有 IP 地址到硬件地址的映射表，这些都是该主机目前知道的一些地址。当主机 A 要向本局域网上的主机 B 发送一个 IP 数据报时，就先在其 ARP 高速缓存中查看有无主机 B 的 IP 地址。若有，就可以查出对应的 B 的硬件地址。然后将此硬件地址写入 MAC 帧，再通过局域网发到该硬件地址。

也有可能查不到主机 B 的 IP 地址的项目。这可能是主机 B 才入网，也可能是主机 A 刚刚加入，其高速缓存还是空的。在这种情况下，主机 A 就自动运行 ARP，去找出主机 B

的硬件地址。其做法是，主机 A 在本局域网上发送一个 ARP 请求，上面有主机 B 的 IP 地址，该 ARP 请求被广播给本局域网上的所有计算机，所有主机上收到此请求后都会检测其中的 IP 地址，与 IP 地址匹配的主机 B 就向主机 A 发送一个 ARP 响应，上面写入自己的硬件地址，A 收到 B 的 ARP 响应后，就在 ARP 高速缓存中写入主机 B 的 IP 地址到硬件地址的映射。

当主机 A 向主机 B 发送数据报时，很可能以后不久主机 B 还要向主机 A 发送数据报，因而主机 B 也可能要向主机 A 发送 ARP 请求。因此主机 A 在发送 ARP 请求时，就将自己的 IP 地址到硬件地址的映射写入 ARP 请求。当主机 B 收到主机 A 的 ARP 请求时，主机 B 就将主机 A 的这一地址映射写入主机 B 自己的 ARP 高速缓存中。这样主机 B 以后向主机 A 发送数据时就方便了。这样做可以减少网络上的通信量。

还有一种地址转换会经常用到，那就是逆地址解析协议(RARP)。RARP 使只知道自己硬件地址的主机能够知道自己的 IP 地址。这种主机往往是无盘工作站。这种无盘工作站一般只要运行其 ROM 中的文件传送代码，就可用下载方法从局域网上其他主机得到所需的操作系统和 TCP/IP 通信软件，但这些软件中并没有 IP 地址。无盘工作站要运行 ROM 中的 RARP 来获得其 IP 地址。RARP 的工作过程大致如下。

为了使 RARP 能够工作，在局域网上至少有一个主机要充当 RARP 服务器，无盘工作站先向局域网发出 RARP 请求，并在此请求中给出自己的硬件地址。RARP 服务器有一个事先做好的从无盘工作站的硬件地址到 IP 地址的映射表，当收到 RARP 请求分组后，RARP 服务器就从这个映射表查出该无盘工作站的 IP 地址，然后写入 RARP 响应分组，发回给无盘工作站。无盘工作站用此方法获得自己的 IP 地址。

### 3.4.3 IP 数据报

TCP/IP 在网际层采用的是无连接服务，并使用 IP 数据报作为数据传送单位。IP 数据报与帧格式类似，也是以首部开始，后跟数据区。其格式如图 3-9 所示，首部的前一部分长度是固定的 20 字节，后一部分长度是可变的。首部各字段的含义如下。


图 3-9 IP 数据报的格式

(1) 版本：占 4 位，指 IP 的版本。目前使用的 IP 版本是 4。通信双方使用的 IP 的版本必须一致。

(2) 首部长度：4 位，可表示的最大数值是 15 个单位，每个单位为 4 字节，因此 IP 的首部长度的最大值是 60 字节。当 IP 数据报的首部长度不是 4 字节的整数倍时，必须利用最后一个填充字段加以填充，因此数据部分总是在 4 字节整数倍处开始。

(3) 服务类型：8 位，用来获得更好的服务，表明发送方需要有什么优先级，是否要求有低延时，是否要求有更高吞吐量以及更高的可靠性。

(4) 总长度：指首部和数据之和的长度，单位为字节。总长度字段为 16 位，因此数据报的最大长度为 65 535 字节。但实际使用的数据报长度很少有超过 1500 字节的。

(5) 标识：该字段是为了使分片后的各数据报片最后能准确地重装成为原来的数据报。

(6) 标志：占 3 位。目前只有前两位有意义。表示后面是否还有分片，以及是否允许分片。

(7) 片偏移：片偏移指出较长的分组在分片后，某片在原分组中的相对位置。

(8) 寿命：该字段记为 TTL，即数据报在网络中的生存时间，其单位为秒(s)。

(9) 协议：占 8 位，协议字段指出此数据报携带的上层数据使用什么协议，以便目的主机的 IP 层知道应将此数据报上交给哪个进程。

(10) 首部校验和：此字段只检验数据报的首部，不包括数据部分。

(11) 地址：这是首部中最重要的字段。源 IP 地址和目的 IP 地址字段都各占 4 字节。

(12) 可变部分字段：IP 首部的可变部分，此字段长度可变，从 1 字节到 40 字节不等。

### 3.4.4 Internet 控制报文协议

因为 IP 定义的是一种尽最大努力的通信服务，其中数据报可能丢失、重复或延迟。因此在 IP 层制定了一种机制，用于减少 IP 数据报的丢失，尽量避免差错并在发生差错时报告信息。这就是 Internet 控制报文协议(Internet Control Message Protocol，ICMP)，一个专门用于发送差错报文的协议。IP 在需要发送一个差错报文时要使用 ICMP，而 ICMP 利用 IP 数据报来传递报文。

ICMP 定义了 5 种差错报文和 4 种信息报文。5 种差错报文分别如下。

(1) 源抑制。当路由器收到太多的数据报以至于缓冲区容量不够时，就发送一个源抑制报文。当一个路由器缓冲区不够用时，就不得不丢弃后来的数据报。因此在丢弃一个数据报时，路由器就会向创建该数据报的主机发送一个源抑制报文，使源主机暂停发送数据报，过一段时间再逐渐恢复正常。

(2) 超时。当路由器将一个数据报的生存时间减到零时，路由器会丢弃这一数据报，并发送一个超时报文。另外，在一个数据报的所有分片到达之前，重组计时器到点了，则主机也会发送一个超时报文。

(3) 目的不可达。当路由器检测到数据报无法传送到它的最终目的地时，就向源主机发送一个目的不可达报文。这种报文告知是特定的目的主机不可达还是目的主机所连的网络不可达。也就是说，这种差错报文能使人区分是某个网络暂时不在互联网上，还是某一特定主机临时断线了。

(4) 改变路由。当一台主机将一个数据报发往远程网络时，主机先将这一数据报发给一个路由器，由路由器将数据报转发到它的目的地。如果路由器发现主机错误地将应该发

给另一个路由器的数据报发给了自己，就使用一个改变路由报文通知主机应改变它的路由。

(5) 要求分片。一个 IP 数据报可以在头部中设置某一位，规定这一数据报不允许被分片。如果路由器发现这个数据报的长度比它要去的网络所规定的最大传输单元(MTU)大时，路由器向发送方发送一个要求分片报文，随后丢弃这一数据报。

另外，ICMP 还定义了以下 4 种信息报文。

(1) 回送请求/应答。回送请求报文是由主机或路由器向一台特定主机发出的询问，收到此请求报文的主机要发一个回送应答报文。这种报文用来测试目的站是否能达。如应用层有一个 PING 服务，用来测试两个主机之间的连通性。PFNG 使用了 ICMP 的回送请求和回送应答报文。

(2) 地址掩码请求/应答。当一台主机启动时，会广播一个地址掩码请求报文，路由器收到这一请求就会发送一个地址掩码应答报文，其中包含了本网络使用的 32 位子网掩码。

可以使用 ICMP 差错报文来跟踪路由。前面我们介绍过，IP 数据报首部有一个寿命字段，该字段是为了避免一个数据报沿着一个环形路径永远不停地走，每一个路由器都要将数据报的寿命计时器减 1。如果计时器到零了，路由器会丢弃这一数据报，并向源主机发回一个 ICMP 超时错误。

另一个用得比较多的 ICMP 差错报文是改变路由。以图 3-10 为例，主机 A 向主机 B 发送 IP 数据报应经过路由器 R2。现在假定主机 A 启动后其路由表中只有一个默认路由器 R1，当主机 A 向主机 B 发送数据报时，数据报就被送往路由器 R1。路由器 R1 从它的路由表查出，发往主机 B 的数据报应经过路由器 R2。于是数据报从路由器 R1 再转到路由器 R2，最后传到主机 B。显然，这个路由是不好的，需要改变。于是，路由器 R1 向主机 A 发送 ICMP 改变路由报文，指出此数据报应经过的下一个路由器 R2 的 IP 地址。主机 A 根据收到的信息更新自己的路由表。以后主机 A 再向主机 B 发送数据报时，根据路由表就知道应将数据报送到路由器 R2，而不再送到默认路由器 R1 了。

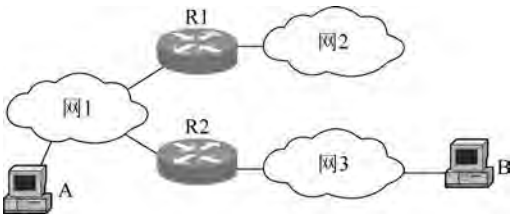



图 3-10　ICMP 改变路由

## 3.5　运输层协议

TCP/IP 的运输层与 7 层模型的第 4 层对应，TCP/IP 在运输层有两个不同的协议：传输控制协议(TCP)和用户数据报协议(UDP)。TCP 的数据传输单位是 TCP 报文段，UDP 的数据传输单位是 UDP 报文或用户数据报，如图 3-11 所示。

TCP 和 UDP 是性质完全不同的传输层协议，被设计用来向高层用户提供不同的服务。两者都使用 IP 作为其网络层的传输协议。TCP 和 UDP 的主要区别在于服务的可靠性。TCP 是高度可靠的，而 UDP 则是一个相当简单的、尽力而为的数据报传输协议，不能确保

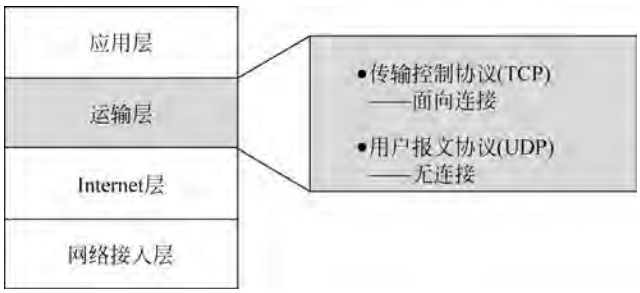



图 3-11　TCP 和 UDP

数据报的可靠传输。两者的这种本质区别也决定了 TCP 的高度复杂性，因此需要大量的功能开销，而 UDP 却由于它的简单性获得了较高的传输效率。

### 3.5.1　传输控制协议

传输控制协议（Transmission Control Protocol，TCP）是一个面向连接的传输层协议，是传输层中使用最广泛的一个协议。TCP 是提供可靠通信的传输层协议，一旦数据报被破坏或丢失，将其重新传输的工作则由 TCP 而非高层应用程序来完成。TCP 也会检测传输错误并予以修正。TCP 提供可靠的全双工数据传输服务。

TCP 是面向连接的。从概念上讲，TCP 传输数据报就像打电话一样，在传输数据开始前，发送端和接收端先建立一个连接，当连接建立好后，开始传输数据，当所有的数据传输完毕后，断开连接，释放资源。TCP 能为应用程序提供可靠的通信连接，使一台计算机发出的字节流无差错地发往网络上的其他计算机，对可靠性要求高的数据通信系统往往使用 TCP 传输数据。

TCP 是面向连接的端到端的可靠协议。它支持多种网络应用程序。TCP 对下层服务没有多少要求，它假定下层只能提供不可靠的数据报服务。TCP 的下层是 IP（网际协议），TCP 可以根据 IP 提供的服务传送大小不定的数据，IP 负责对数据进行分段、重组，在多种网络中传送。

**1. 端口**

在 Internet 中，IP 使用 IP 地址来标识一台主机，但是对于应用来说，仅仅知道哪台主机是不够的，即使两台主机之间也可能有多个应用同时运行，如一个应用进程执行邮件传输，而另一个进程执行文件传输。为了区分不同的应用，TCP 与 UDP 中引入了端口（port）的概念，用于标识通信的进程。

端口实际上是一个抽象的软件结构，它是操作系统可分配的一种资源。运输层通过端口为应用提供服务，TCP 和 UDP 都用端口把信息传递给上层，如图 3-12 所示。端口号用来跟踪同一时间内通过网络的不同会话，也就是说一个应用进程是与某个端口连接在一起的。但是，仅仅依靠端口号不能完全确定一个应用的位置，因为有可能存在不同的主机，它们分别运行了具有相同端口号的应用，而一旦两者与第三个主机的应用通信时，那个第三者就无法知道究竟是哪个主机上的进程与自己通信。

按照 TCP/IP 运输层协议的定义，完全确定一对应用之间的关系必须使用 4 个参数：源 IP 地址、目的 IP 地址、源端口号和目的端口号。这也称为连接，连接还可以从插口

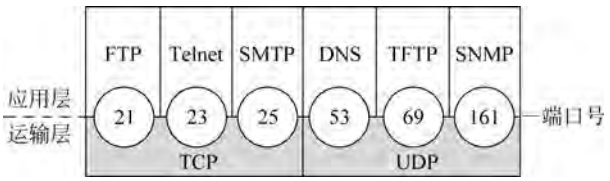



图 3-12　端口号描述了正在使用运输层的上层协议

(Socket)概念的角度来定义,即一个连接由两个插口构成,一个插口由两部分信息标识,即IP 地址与端口号。端口号是一个 16 位二进制数,约定 256 以下的端口号被标准服务保留,取值大于 256 的为自由端口。自由端口是在端主机的进程间建立传输连接时由本地用户进程动态分配得到。由于 TCP 和 UDP 就完全独立的两个软件模块,所以各自的端口号是相互独立的。

**2. TCP 报文**

TCP 报文段的格式如图 3-13 所示。TCP 头部的固定部分为 20 字节,以下分别介绍每个字段的含义。


图 3-13　TCP 报文段的格式

(1) 源端口号:16 位的源端口号字段包含源端应用连接的端口值。源端口号与源 IP 地址一起用于标识报文段的返回地址。

(2) 目的端口号:16 位的目的端口号字段定义了传输目的的应用连接的端口,这实际上是报文接收端主机上的应用程序的地址。

(3) 发送序号:32 位的发送序号指出段中首字节数据的序号,该序号值由接收端主机使用,用于重组报文段中的数据,获取一个完整的应用层报文。

(4) 确认号:32 位的确认号仅当 ACK 标志设置时有意义。其中的值表示期望接收的下一个报文段数据部分第一字节的序号。该字段用于对已收到的报文段进行确认。

(5) 报头长度:4 位的报头长度字段指出段中数据部分相对于首部起始处的偏移位置,以 32 位为单位。这实际上是指 TCP 首部的大小。

(6) 预留:6 位保留字段应该设置为 0,这是为将来增加新功能而保留的。

(7) 编码位:6 位编码位字段用于指定报文段的性质。6 个编码标志分别为:紧急标志(URG)、确认(ACK)标志、推(PSH)标志、连接复位(RST)标志、同步(SYN)标志和终止(FIN)标志。这 6 个编码标志并非完全独立,在很多情况下可以同时设置其中的多个标志。编码标志的意义参见表 3-6。

表 3-6　TCP 头部编码标志位的含义

| 标　志　位 | 含　　义 |
| --- | --- |
| URG | 紧急指针(Urgent Pointer)有效 |
| ACK | 确认号有效 |
| PSH | 要求接收端尽快将这个报文段交给应用层 |
| RST | 复位一个 TCP 连接 |
| SYN | 同步序号用于建立一个连接 |
| FIN | 发送端口已经完成发送任务并要求终止传输 |

(8) 窗口：接收端主机使用 16 位的窗口字段通告本地可用缓冲区的大小，这也是对方能够发送的未被确认的最大数据量，单位为字节。

(9) 校验和：TCP 头部包括 16 位的校验和字段。采用 IP 校验和的计算算法，即求 TCP 段中所有内容(包括首部与数据部分)的 16 位二进制字的反码和的反码。此外也要求在计算中包含一个相关的 TCP 伪头部。伪头部共有 12 字节，它仅仅用于校验和的计算，如图 3-14 所示。

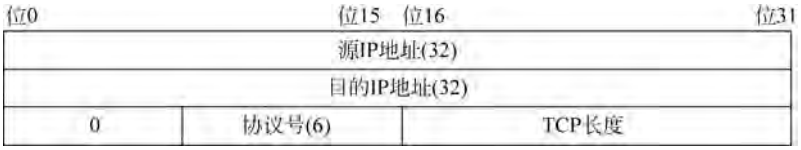



图 3-14　TCP 的伪头部结构

(10) 紧急指针：该字段的长度为 16 位，指向数据段所包含的紧急数据的最后一字节的位置，该字段只有在 URG 标志为 1 时才有意义。紧急数据的发送与窗口大小无关。

(11) 选项：TCP 定义了几种选项，其中最有用的是 MSS(最大段长)。这个选项在两个主机建立连接时交换，分别告诉对方自己可以接收的最大报文段的长度。为了提高网络的利用效率，在允许情况下应该选择一个尽可能大的 MSS 值。

(12) 填充：如果使用选项，为了确保 TCP 头部以 32 位为基本单位，可以使用填充字段来满足这一要求。

(13) 数据：上层协议数据(不固定)。

**3. TCP 的编号与确认**

TCP 的协议数据单元称为报文段(Segment)，但是 TCP 的数据并非以一个段作为基本单位，而是采用字节流形式。TCP 不是按传送的报文段来编号，而是将所要的整个报文(可能包括许多报文段)看成是由字节组成的数据流，然后对每一字节编一个序号。在连接建立时双方要商定初始序号，TCP 就将每一次所传送的第一个数据字节的序号放在 TCP 首部的序号字段中。TCP 提供了报文段排序功能，源工作站在传输之前对每个报文段都进行编号，在接收端，TCP 再将这些段重组成一条完整的消息。

TCP 被设计用于全双工通信，一个送往对方的报文段中除了送往对方的数据外，往往还有对所收到的报文段的确认，这种确认方式称为“捎带”。利用这种全双工特点，TCP 会话中的报文段几乎大多数都携带了确认标记，如图 3-15 所示。

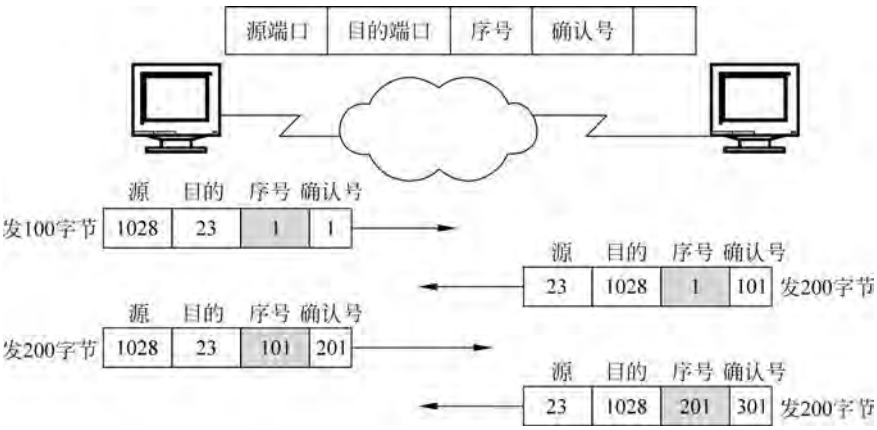



图 3-15　TCP 序号和确认号

**4. TCP 的传输连接管理**

TCP 是一个面向连接的协议，所以需要在传输数据之前建立连接。面向连接的服务包括三个阶段：连接建立阶段、数据传输阶段和连接终止阶段。在连接建立阶段，源和目标之间建立连接或会话。资源通常在这一阶段进行保留，以确保服务等级的一致性。在数据传输阶段，数据通过建立的路径按顺序传输，并按发送的顺序到达目标。连接终止阶段是在源和目标之间不再需要连接时，将连接终止的阶段。

为确保连接的建立和终止都是可靠的，TCP 使用了三次握手方式来交换信息。就是说，它在送出真正的数据之前会先利用控制信息和对方建立连接。首先连接发起端 TCP 先送出同步信息，另一端收到后回答同步、确认信息，接着发起端再回答确认信息完成连接建立，然后它们才开始传出第一组真正的数据。

为建立或初始化连接，两台主机的初始序号必须同步。由 TCP 主机随机选取的序号用于跟踪报文段的顺序，以确保在传输过程中没有丢失。初始序号是建立 TCP 连接时使用的起始号。在连接阶段交换初始序号可确保重新获得丢失的数据。

同步通过交换建立连接报文段来完成，这些报文段承载了称为 SYN(同步)的控制位和初始序号。承载 SYN 位的分段也叫 SYN。同步要求每方都发送各自的初始序号并接收对方对初始序号的确认(ACK)。下面是三次握手的步骤，如图 3-16 所示。

(1) 主机 A 到主机 B 的 SYN：在 SYN 分段中，主机 A 告诉主机 B，序号(SEQ)为 X。

(2) 主机 B 到主机 A 的 SYN：主机 B 接收 SYN，记录序号 X，并用 ACK＝X＋1 和自己的序号(SEO＝Y)确认 SYN，以回答主机 A。ACK＝X＋1 意味着主机(这里为主机 B)收到了 X，并希望下一个 SYN 是 X＋1。这一技术称为转发确认。

(3) 主机 A 到主机 B 的 ACK：主机 A 随后对主机 B 发送的数据进行确认，指出主机 A 希望收到的下一个报文段的序号是 Y＋l(ACK＝Y＋1)。

在完成了上述 3 步后，连接就建立好了。此时，双方就可以开始数据传输了。

三次握手是必需的，由于 IP 是不可靠的，先前发出的 SYN 报文段可能迟到，为了避免建立无用连接，需要源端对目的端发来的 SYN 报文段进行确认。

连接释放过程和建立时的三次握手在本质上是一致的。终止一条，TCP 连接实际上也是三次握手。只是在实际中，连接释放分为两个方向上的独立过程，因此其中可能会涉及 4

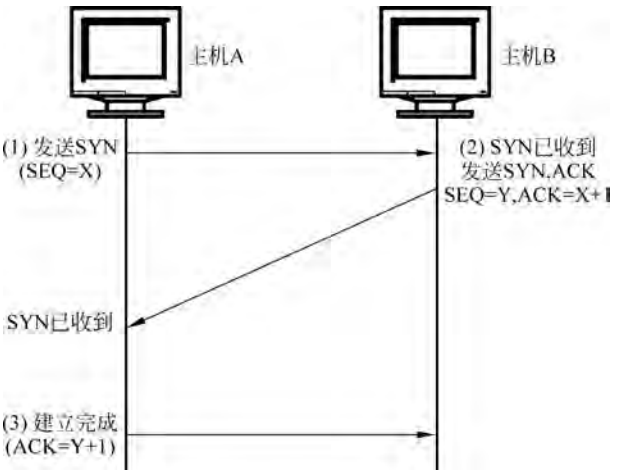



图 3-16　三次握手

个报文段的交换，但是可以将其中的第 2 个与第 3 个报文段理解为一个，而且事实上当目的端也无任何数据传递时这两个段是合二为一的。由于 TCP 连接是全双工的，因此每个方向必须单独进行关闭。当一方完成它的数据发送任务后就能发送一个完成信息来终止这个方向连接。当一端收到一个完成信息后，它必须通知应用层另一端已经终止了那个方向的数据传送，发送完成信息通常是应用层进行关闭的结果。收到一个完成信息只意味在这一方向上没有数据流动。正常关闭的过程首先是发起关闭的一方执行主动关闭，而另一方执行被动关闭。但是实际操作中也可能双方同时执行主动关闭而切断连接。按不同方向分别终止连接的主要目的是为了避免连接终止时可能产生的数据丢失。

**5. TCP 流量控制与拥塞控制**

为管理设备之间的数据流，TCP 使用了流量控制机制。当创建一条连接时，连接的每一端都会分配一个缓冲区用于保存输入的数据，并将缓冲区尺寸放在交换的报文段中传输到对方。当数据到达时，接收端发送确认信息，其中包含了目前本地剩余的缓冲区的大小。这个缓冲区空间的可用大小值被称为通知窗口，该窗口指定了接收方 TCP 此时准备接收的 8 位字节数(从确认号开始)。在相应的报文段中给出窗口的过程称为窗口通告。接收端在发送的每一个确认中都含有一个窗口通告。如果接收端能够及时处理到达的数据，则总是会在每个接收确认中发送一个正的(大于 0 的)窗口通告。如果发送端传输的速度高于接收端，那么接收端的数据缓冲区最终将被接收到的报文段填满，因此导致接收端给出一个零窗口值。当发送端收到一个零窗口通告时，必须停止发送，直到接收端重新给出一个正的窗口值为止。窗口尺寸决定了收到目标的确认之前能传输的数据量。在连接期间，TCP 窗口的尺寸是可变的。每个确认中都包含窗口通告，指出了接收方能接受的字节数。窗口尺寸(字节数)越大，主机能传输的数据量越多。传输了窗口尺寸指定的字节数后，主机必须在接收到对这些报文段的确认后才能继续发送数据。

TCP 还维护了一个拥塞控制窗口，其尺寸通常与接收方窗口相同，但是在段丢失时(如出现拥塞时)，窗口的尺寸将减半。这种方法使得窗口尺寸将随管理缓冲区空间和处理的需要扩大或缩小。

TCP 的流量控制常常用于拥塞控制。如果网络系统产生了拥塞，则由此造成的分组丢

失将会十分严重，而且由于 TCP 的重传机制的作用，可能会加重拥塞现象。这将是一连锁反应，最终导致整个系统进入死锁状态，使网络彻底停止运行。为了避免出现这一问题，TCP 总是用网络中的分组丢失来估计拥塞。如果发生了分组丢失，则 TCP 将降低重传数据的速率，并开始实施拥塞控制。TCP 开始时不会发送大量的数据，而是只发送一个报文段。如果确认到来，则 TCP 就将发送的数据量（作为拥塞窗口）加倍，即发送两个报文段。如果对应的所有确认都到来了，则 TCP 就再发 4 个报文段，依次类推。这种递增过程是按指数增长的，一直持续到 TCP 发送的数据量到达内部维护的一个拥塞控制阈限值为止。这时，TCP 将降低增长率，使之以缓慢的速率增加，这就是所谓的拥塞避免。而一旦产生重传，TCP 又将重复前述的过程。

上述过程中，TCP 使用了几个技术："慢启动"、拥塞控制和加速递减的算法。为了实施这一过程，除了原先的通知窗口之外，在发送端的 TCP 实体内部增加了一个拥塞窗口。当与一台主机建立 TCP 连接时，拥塞窗口被初始化为一个报文段大小，随着每次都接收到对方返回的确认，拥塞窗口值将翻倍。但是无论发送端如何加速，总的发送窗口大小有一个上限，该值为发送端维护的拥塞窗口与通知窗口中的较小者。其中拥塞窗口是发送方使用的流量控制，而通知窗口则是接收方使用的流量控制。如果在某个时刻发生了超时，则 TCP 将把当前拥塞窗口值的一半作为新的阈限窗口值，而同时将拥塞窗口再次变为 1。

**6. TCP 的重传机制**

TCP 实现可靠传输的技术是重传机制。重传机制可确保从一台设备发送的数据流传送到另一台设备时不会出现重复或数据丢失。当 TCP 发送报文段时，启动一个时钟，跟踪对该报文段的确认；当 TCP 正确收到一个报文段时，应该送回一个确认信息。若收到有差错的报文段，则丢弃此报文段，而不发送否认信息。若收到重复的报文段，也要将其丢弃，但要发回（或捎带发回）确认信息。如果发送端在时钟超时之前没有收到确认信息，将重传该报文段，这一过程称为超时重传。

TCP 的重传采用自适应的算法。重发机制是 TCP 最重要、最复杂的问题之一。TCP 之所以值得信赖，其关键是数据传输服务是建立在一个称为正向认可与重传的机制上。采用这种传输机制的系统会每隔一定时间送出一个相同的报文段，直到收到对方的确认信息之后，再送出下个报文段。TCP 每发送一个报文段，就设置一次定时器，只要定时器设置的重发时间已到而还没有收到确认，就要重发这一报文段。

TCP 也采用校验和（Checksum）计算数据的正确性，该校验和位于每个报文段的首部。当 TCP 收到一个报文段时，会首先将它所计算的检查值与包中的校验和进行比较，若相同，则送出确认信息，否则丢弃该报文段，等待对方重传的报文段。

## 3.5.2 用户数据报协议

用户数据报协议（User Datagram Protocol，UDP）也是传输层的一个重要协议，用来支持那些需要在计算机之间传输数据的网络应用。包括网络视频会议系统在内的众多客户/服务器模式的网络应用都需要使用 UDP。UDP 从问世至今已经被使用了很多年，虽然其最初的光彩已经被一些类似协议所掩盖，但是即使在今天，UDP 仍然不失为一项非常实用和可行的网络传输层协议。

与我们所熟知的 TCP 一样，UDP 直接位于 IP 的顶层。根据 OSI 参考模型，UDP 和

TCP 都属于传输层协议。

与 TCP 不同的是，UDP 是一个无连接的协议，无连接的通信不提供可靠性，即不通知发送端口是否正确接收了报文。无连接协议也不提供错误恢复能力。UDP 比 TCP 要简单得多，它可以简单地与 IP 或其他协议连接，只充当数据报的发送者或接收者。如图 3-17 所示为 UDP 分段报文的格式，UDP 没有顺序字段或确认字段。UDP 报文中的字段为终端工作站之间提供了通信能力。

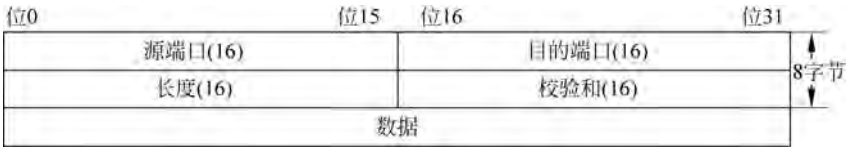



图 3-17　UDP 分段报文的格式

UDP 报头的长度通常为 64 位。UDP 段中的字段定义包括如下。

(1) 源端口。呼叫端口号(16 位)，源端口用于标识发送端口的应用程序，当无须返回数据时置为 0。

(2) 目的端口。被叫端口号(16 位)，目的端口用于标识目的端口的应用程序。

(3) 长度。UDP 报头和 UDP 数据的长度(16 位)，长度以字节为单位。

(4) 校验和。通过计算得到的报头和数据字段的校验和(16 位)，校验和是一个可选段，置 0 时表示未选，全 1 表示校验和为 0。

(5) 数据。上层协议数据(不固定)。

UDP 使用端口号为不同的应用保留其各自的数据传输通道。UDP 和 TCP 正是采用这一机制实现对同一时刻内多项应用同时发送和接收数据的支持。数据发送一方(可以是客户端或服务器端)将 UDP 数据报通过源端口发送出去，而数据接收一方则通过目标端口接收数据。有的网络应用只能使用预先为其预留的静态端口；而另外一些网络应用则可以使用未被指派的动态端口。因为 UDP 报头使用两字节存放端口号，所以端口号的有效范围是 0～65 535。

数据报的长度是指包括报头和数据部分在内的总的字节数。因为报头的长度是固定的，所以该域主要被用来计算可变长度的数据部分(又称为数据负载)。数据报的最大长度根据操作环境的不同而异。从理论上说，包含报头在内的数据报的最大长度为 65 535 字节。不过，一些实际应用往往会限制数据报的大小，有时会降低到 8192 字节。

UDP 使用报头中的校验值来保证数据的安全。校验值首先在数据发送方通过特殊的算法计算得出，在传递到接收方之后，还需要重新计算。如果某个数据报在传输过程中被第三方篡改或者由于线路噪声等原因受到损坏，发送和接收方的校验计算值将不会相符，由此 UDP 可以检测是否出错。其实在 UDP 中校验功能是可选的，这与 TCP 不同，后者要求必须具有校验值。

UDP 和 TCP 的主要区别是两者在如何实现信息的可靠传递方面不同。TCP 中包含了专门的传输保证机制，当数据接收方收到发送方传来的信息时，会自动向发送方发出确认消息；发送方只有在接收到该确认消息或超时后才继续传送其他信息。

与 TCP 不同，UDP 并不提供数据传送的保证机制。如果在从发送方到接收方的传递

过程中出现数据包的丢失，协议本身并不能做出任何检测或提示。因此，通常人们把 UDP 称为不可靠的传输协议。

相对于 TCP，UDP 的不同之处在于如何接收突发性的多个数据包，而且 UDP 并不能确保数据的发送和接收顺序。

“无连接”就是在正式通信前不必与对方先建立连接，不管对方状态如何就直接发送。这与现在流行的手机短信非常相似：在发短信时，只需要输入对方手机号就可以了。UDP 适用于一次只传送少量数据、对可靠性要求不高的应用环境，如 SNMP（简单网络管理协议）。

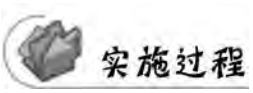
TCP 和 UDP 互不相同，适用于不同要求的通信环境。TCP 和 UDP 之间的差别如表 3-7 所示。

**表 3-7　TCP 和 UDP 的差别**

| 对　比　项 | TCP | UDP |
|---|---|---|
| 是否连接 | 面向连接 | 无连接 |
| 传输可靠性 | 可靠的 | 不可靠的 |
| 应用场合 | 传输大量的数据 | 少量数据 |
| 速度 | 慢 | 快 |

或许人们要问，既然 UDP 是一种不可靠的网络协议，那何必还要保留使用呢？其实不然，在有些情况下 UDP 可能会变得非常有用，因为 UDP 具有 TCP 所望尘莫及的速度优势。虽然 TCP 中植入了各种安全保障功能，但是在实际执行的过程中会占用大量的系统开销，无疑使速度受到严重的影响。而 UDP 由于排除了信息可靠传递机制，将安全和排序等功能移交给上层应用来完成，从而极大地降低了执行时间，使速度得到了保证。

使用 UDP 的协议包括简单文件传输协议（TFTP）、简单网络管理协议（SNMP）、网络文件系统（NFS）和域名系统（DNS）等。




## 任务 1　常用网络命令的使用

计算机网络在使用过程中，各种网络设备出现故障的情况在所难免，网络管理人员除了使用各种硬件检测设备和测试工具之外，还可以利用操作系统本身内置的一些网络命令，对所在的网络进行故障检测和维护。

**1. ping 命令的使用**

ping 命令是一个使用频率极高的 ICMP 的程序，可用来测试目标机或路由器的可达性。它是一个连通性测试程序，如果能 ping 通目标，就可以排除网络访问层、网卡、Modem 的输入输出线路、电缆和路由器等存在的故障；如果 ping 目标 A 通，而 ping 目标 B 不通，则网络发生在 A 与 B 之间的链路上或 B 上，从而缩小故障范围。

通过执行 ping 命令可获得如下信息。

（1）检测网络的连通性，检验与远程计算机或本地计算机的连接。

（2）确定是否有数据报被丢失、复制或重传。ping 命令在所发送的数据报中设置唯一

的序列号(Sequence Number),以此检查其接收到应答报文的序列号。

(3) ping 命令在其所发送的数据报中设置时间戳(Time Stamp),根据返回的时间戳信息可以计算数据包交换的时间,即 RTT(Round Trip Time)。

(4) ping 命令校验每一个收到的数据报,据此可以确定数据报是否损坏。

ping 命令的格式如下。

ping 目的 IP 地址

```
[-t][-a][-n count][-l size][-f][-i TTL][-v TOS][-r count][-s count]
[[-j host-list]|[-k host-list]][-w timeout]
```

ping 命令各选项的含义如表 3-8 所示。

**表 3-8　ping 命令各选项的含义**

| 选　　项 | 含　　义 |
|---|---|
| －t | 连续 ping 目标机,直到手动停止(按 Ctrl＋C 组合键) |
| －a | 将 IP 地址解析为主机名 |
| －n count | 发送回送请求 ICMP 报文的次数(默认值为 4) |
| －l size | 定义 echo 数据报的大小(默认值为 32 B) |
| －f | 不允许分片(默认为允许分片) |
| －i TTL | 指定生存周期 |
| －v TOS | 指定要求的服务类型 |
| －r count | 记录路由 |
| －s count | 使用时间戳选项 |
| －j host-list | 使用松散源路由选项 |
| －k host-list | 使用严格源路由选项 |
| －w timeout | 指定等待每个回送应答的超时时间(以 ms 为单位,默认值为 1000,即 1s) |

(1) 测试本机是否正确安装了 TCP/IP。

执行“ping 127.0.0.1”命令,如果能 ping 成功,说明 TCP/IP 已正确安装。“127.0.0.1”是回送地址,它永远回送到本机。

(2) 测试本机 IP 地址是否正确配置或者网卡是否正常工作。

执行“ping 本机 IP 地址”命令,如果能 ping 成功,说明本机 IP 地址配置正确,并且网卡工作正常。

(3) 测试与网关之间的连通性。

执行“ping 网关 IP 地址”命令,如果能 ping 成功,说明本机到网关之间的物理线路是连通的。

(4) 测试能否访问 Internet。

执行“ping 60.215.128.237”命令,如果能 ping 成功,说明本机能访问 Internet。其中,“60.215.128.237”是 Internet 上新浪服务器的 IP 地址。

(5) 测试 DNS 服务器是否正常工作。

执行“ping www.sina.com.cn”命令,如果能 ping 成功,如图 3-18 所示,说明 DNS 服务器工作正常,能把网址(www.sina.com.cn)正确解析为 IP 地址(60.215.128.237)。否则,说明主机的 DNS 未设置或设置有误等。


图 3-18　使用 ping 测试 DNS 服务器是否正常

如果计算机打不开任何网页,可通过上述的 5 个步骤来诊断故障的位置,并采取相应的解决措施。

(6) 连续发送 ping 探测报文。

```
ping -t 60.215.128.237
```

(7) 使用自选数据长度的 ping 探测报文,如图 3-19 所示。


图 3-19　使用自选数据长度的 ping 探测报文

(8) 修改 ping 命令的请求超时时间,如图 3-20 所示。


图 3-20　修改 ping 命令的请求超时时间

(9) 不允许路由器对 ping 探测报文分片。如果指定的探测报文的长度太长,同时又不允许分片,探测数据报就不可能到达目的地并返回应答,如图 3-21 所示。

```
C:\Users\Administrator>ping  -l 64 60.215.128.237

正在 Ping 60.215.128.237 具有 64 字节的数据:
来自 60.215.128.237 的回复: 字节=64 时间=132ms TTL=51
来自 60.215.128.237 的回复: 字节=64 时间=88ms TTL=51
来自 60.215.128.237 的回复: 字节=64 时间=84ms TTL=51
来自 60.215.128.237 的回复: 字节=64 时间=82ms TTL=51

60.215.128.237 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 82ms, 最长 = 132ms, 平均 = 96ms
```

图 3-21　不允许路由器对 ping 探测报文分片

**2. ipconfig 命令的使用**

ipconfig 命令可以查看主机当前的 TCP/IP 配置信息,如 IP 地址、网关、子网掩码等刷新动态主机配置协议(DHCP)和域名系统(DNS)设置。

ipconfig 命令的语法格式如下。

```
ipconfig[/all][/renew[Adapter]][/release[Adapter]][/flushdns][/displaydns][/registerdns]
[/showclassid Adapter] [/setclassid Adapter [ClassID]]
```

ipconfig 命令各选项的含义如表 3-9 所示。

**表 3-9　ipconfig 命令各选项的含义**

| 选　　项 | 含　　义 |
|---|---|
| /all | 显示所有适配器的完整 TCP/IP 配置信息 |
| /renew[Adapter] | 更新所有适配器或特定适配器的 DHCP 配置 |
| /release[Adapter] | 发送 DHCP RELEASE 消息到 DHCP 服务器,以释放所有适配器或特定适配器的当前 DHCP 配置并丢弃 IP 地址配置 |
| /flushdns | 刷新并重设 DNS 客户解析缓存的内容 |
| /displaydns | 显示 DNS 客户解析缓存的内容,包括从 Local Hosts 文件预装载的记录以及最近获得的针对由计算机解析的名称查询的资源记录 |
| /registerdns | 初始化计算机上配置的 DNS 名称和 IP 地址的手工动态注册 |
| /showclassid Adapter | 显示指定适配器的 DHCP 类别 ID |
| /setclassid Adapter[ClassID] | 配置特定适配器的 DHCP 类别 ID |
| /? | 在命令提示符下显示帮助信息 |

(1) 要显示基本 TCP/IP 配置信息,可执行 ipconfig 命令。

使用不带参数的 ipconfig 命令可以显示所有适配器的 IP 地址、子网掩码和默认网关。

(2) 要显示完整的 TCP/IP 配置信息(主机名、网卡的 MAC 地址、主机的 IP 地址、子网掩码、默认网关地址、DNS 服务器等),可执行“ipconfig/all”命令。

(3) 仅更新“本地连接”适配器中由 DHCP 分配的 IP 地址配置,可执行“ipconfig/renew”命令。

(4) 要在排除 DNS 的名称解析故障期间刷新 DNS 解析器缓存,可执行“ipconfig/flushdns”命令。

**3. arp 命令的使用**

地址解析协议 ARP 是一个重要的 TCP/IP,并且用于确定对应 IP 地址的网卡物理地址。arp 命令用于查看、添加和删除缓存中的 ARP 表项。

ARP 表可以包含动态(Dynamic)和静态(Static)表项,用于存储 IP 地址与 MAC 地址的映射关系。

动态表项随时间推移自动添加和删除。而静态表项则一直保留在高速缓存中,直到人为删除或重新启动计算机为止。

每个动态表项的潜在生命周期是 10 分钟,新表项加入时定时器开始计时,如果某个表项添加后 2 分钟内没有被再次使用,则此表项过期并从 ARP 表中删除。如果某个表项始终在使用,则它的最长生命周期为 10 分钟。

(1) 显示高速缓存中的 ARP 表,命令如图 3-22 所示。

```
C:\Users\Administrator>arp -a

接口: 169.254.63.207 --- 0xb
  Internet 地址         物理地址              类型
  169.254.255.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态

接口: 192.168.56.1 --- 0xc
  Internet 地址         物理地址              类型
  192.168.56.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态
```

图 3-22　显示高速缓存中的 ARP 表

(2) 添加 ARP 静态表项,命令如图 3-23 所示。

```
C:\WINDOWS\system32>arp -s 192.168.0.101 1c-bd-b7-cd-e3-db

C:\WINDOWS\system32>arp -a

接口: 169.254.63.207 --- 0xb
  Internet 地址         物理地址              类型
  224.0.0.22            01-00-5e-00-00-16     静态

接口: 192.168.56.1 --- 0xc
  Internet 地址         物理地址              类型
  192.168.56.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态

接口: 169.254.27.151 --- 0x10
  Internet 地址         物理地址              类型
  192.168.0.101         1c-bd-b7-cd-e3-db     静态
  224.0.0.22            01-00-5e-00-00-16     静态

接口: 192.168.18.104 --- 0x12
  Internet 地址         物理地址              类型
  192.168.18.1          00-60-7e-59-7a-c6     动态
  224.0.0.22            01-00-5e-00-00-16     静态
```

图 3-23　添加 ARP 静态表项

(3) 删除 ARP 表项,命令如图 3-24 所示。

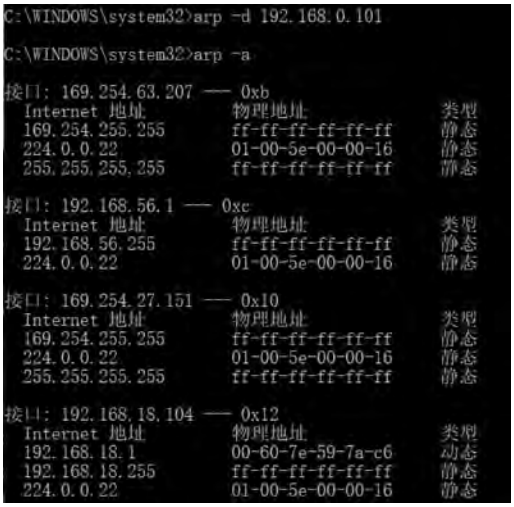



图 3-24　删除 ARP 静态表项

**4. tracert 命令的使用**

tracert(跟踪路由)是路由跟踪的一个实用程序,用于获得 IP 数据报访问目标时从本地计算机到目标机的路径信息。

tracert 命令的语法格式如下。

```
tracert [ - d] [ - h MaximumHops] [ - i HostList][ - w Timeout][ - R][ - s SrcAddr][ - 4][ - 6]
TargetName
```

tracert 命令各选项的含义如表 3-10 所示。

**表 3-10　tracert 命令各选项的含义**

| 选　　项 | 含　　义 |
|---|---|
| -d | 防止 tracert 试图将中间路由器的 IP 地址解析为它们的名称 |
| -h MaximumHops | 指定搜索目标(目的)的路径中“跳数”的最大值。默认“跳数”值为 30 |
| -i HostList | 指定“回显请求”消息将 IP 报头中的松散源路由选项与 HostList 中指定的中间目标集一起使用 |
| -w Timeout | 指定等待“ICMP 已超时”或“回显答复”消息(对应于要接收的给定“回显请求”消息)的时间(ms) |
| -R | 指定 IPv6 路由扩展报头应用来将“回显请求”消息发送到本地主机,使用指定目标作为中间目标并测试反向路由 |
| -s SrcAddr | 指定在“回显请求”消息中使用的源地址。仅当跟踪 IPv6 地址时才使用该参数 |
| -4 | 指定 tracert 只能将 IPv4 用于本跟踪 |
| -6 | 指定 tracert 只能将 IPv6 用于本跟踪 |
| TargetName | 指定目标,可以是 IP 地址或主机名 |
| / | 在命令提示符下显示帮助 |

(1) 跟踪名为“www.163.com”的主机的路径,可执行“tracert www.163.com”命令,结果如图 3-25 所示。


图 3-25　使用 tracert 命令跟踪主机的路径(1)

(2) 跟踪名为“www.163.com”的主机的路径,并防止将每个 IP 地址解析为它的名称,可执行“tracert-d www.163.com”命令,结果如图 3-26 所示。


图 3-26　使用 tracert 命令跟踪主机的路径(2)

**5. netstat 命令的使用**

netstat(网络状态)命令可以显示当前活动的 TCP 连接、计算机侦听的端口、以太网统计信息、IP 路由表、IPv4 或 IPv6 的统计信息等,用于检测本机各端口的网络连接情况。

netstat 命令的语法格式如下。

```
netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]
```

netstat 命令各选项的含义如表 3-11 所示。

**表 3-11　netstat 命令各选项的含义**

| 选　项 | 含　义 |
|---|---|
| -a | 显示所有活动的 TCP 连接及计算机侦听的 TCP 和 UDP 端口 |
| -e | 显示以太网统计信息,如发送和接收的字节数、数据包数等 |
| -n | 显示活动的 TCP 连接,不过只以数字形式表示地址和端口号 |
| -o | 显示活动的 TCP 连接并包括每个连接的进程 ID(PID)。该选项可以与-a、-n 和-p 选项结合使用 |
| -p Protocol | 显示 Protocol 所指定的协议的连接 |

续表

| 选　　项 | 含　　义 |
| --- | --- |
| -r | 显示 IP 路由表的内容。该选项与“route print”命令等价 |
| -s | 按协议显示统计信息 |
| Interval | 每隔 Interval 秒重新显示一次选定的消息。按 Ctrl＋C 组合键停止重新显示统计信息。如果省略该选项，netstat 将只显示一次选定的信息 |
| /? | 在命令提示符下显示帮助 |

（1）显示所有活动的 TCP 连接及计算机侦听的 TCP 和 UDP 端口，可执行 netstat 命令，结果如图 3-27 所示。


图 3-27　显示所有活动的 TCP 连接

（2）显示以太网统计信息，如发送和接收的字节数、数据包数等，可执行“netstat-s”命令，结果如图 3-28 所示。

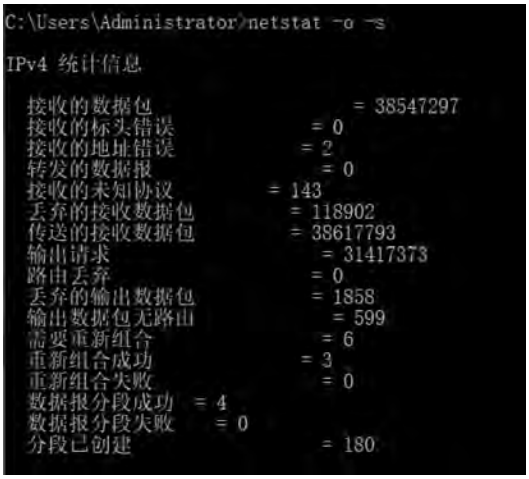



图 3-28　显示以太网统计信息

## 任务 2　配置与测试 IP 地址

### 1. IP 地址修改设置方法

步骤 1：进入“网络连接”窗口。

方法 1：在桌面右击“网上邻居”，在弹出的菜单中选择“属性”，如图 3-29 所示。

方法 2：在桌面单击“开始”→“设置”→“控制面板”，如图 3-30 所示。在弹出的“控制面板”中双击“网络连接”，如图 3-31 中黑框标出图标所示。


图 3-29 “网上邻居”下拉菜单

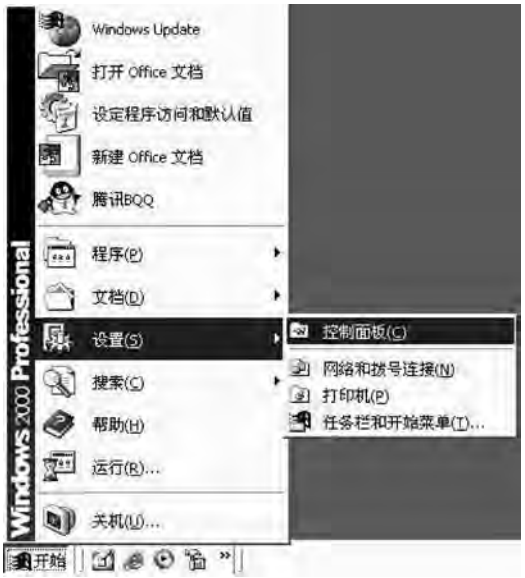



图 3-30 选择“设置”的子菜单“控制面板”


图 3-31 “网络连接”图标

步骤 2：进入“本地连接属性”窗口。

在弹出的“网络连接”窗口中，右击“本地连接”（或者是“本地连接 1”“本地连接 2”“本地连接 3”等），选择“属性”，如图 3-32 所示。

步骤 3：进入“Internet 协议（TCP/IP）属性”窗口。

在弹出的“本地连接属性”窗口中，选择“Internet 协议（TCP/IP）”，再选择“属性”，如图 3-33 所示。


图 3-32　由“本地连接”选择“属性”


图 3-33　“本地连接属性”窗口

步骤 4：在“Internet 协议（TCP/IP）属性”窗口中修改 IP 地址。

在弹出窗口中，方框标出位置以外空白处均按图 3-34 所示填写。

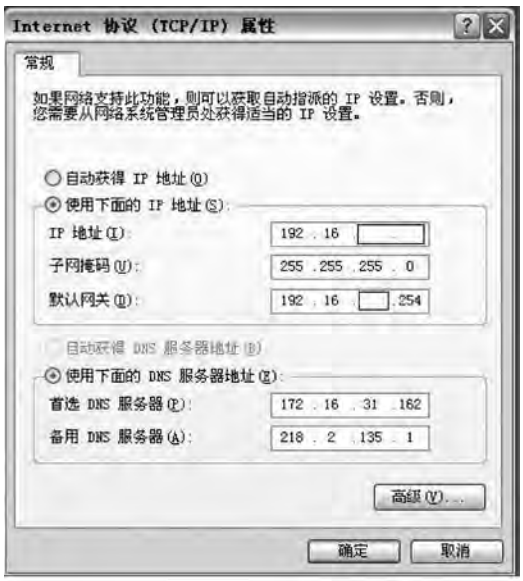



图 3-34　修改 IP 地址

方框标出的位置，IP 地址方框处对照机器桌面号与 IP 地址对应关系填写(11 为机器所在桌面号)，默认网关方框处(第三组数字)与 IP 地址第三组数字相同。

以 192.16.1.11 为例，IP 地址：192.16.1.11。默认网关：192.16.1.254。

**2. 计算机名修改设置方法**

(1) 计算机起名规则。

用户可按如下规则修改计算机名：如果是个人使用计算机，将计算机名改为“单位名称＋用户名(用户名可用中文或拼音字母，推荐使用中文)”，如教育局李四；如果是单位公用计算机，将计算机名改为“单位名称＋房间号＋数字(两位)”，如教育局 101601。

(2) 具体修改步骤。

修改方法：右击桌面上的“我的电脑”，选择“属性”选项，在弹出的“系统属性”对话框中选择“计算机名”标签，单击“更改”按钮，在弹出的“计算机名称更改”对话框中改好计算机名后按“确定”按钮，重启计算机以使它生效。如图 3-35～图 3-37 所示。


图 3-35 “我的电脑”子菜单


图 3-36 “计算机名”选项卡

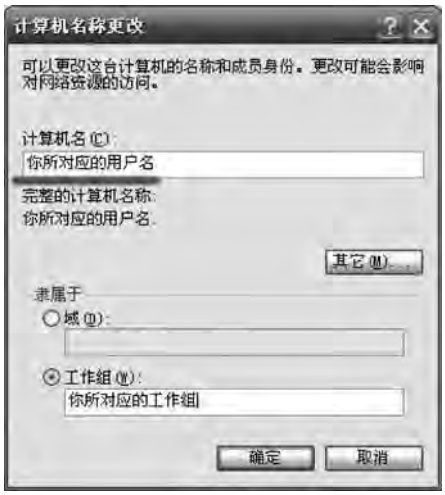



图 3-37 “计算机名称更改”对话框

**3. 使用自动专用 IP 寻址 APIPA(Automatic Private IP Addressing)**

Windows 2000 的 TCP/IP 工具支持一种新方式，为基于 LAN 的简单网络配置自动分配 IP 地址的方式。这种寻址方式是 LAN 适配器的动态 IP 地址分配的一处扩展，可以不使用静态 IP 地址分配或安装 DHCP 就可以配置 IP 地址。这种方法的前提是在 TCP/IP 属性对话框中单击“自动获取 IP 地址”。下列步骤描述了 APIPA 如何分配 IP 地址。

(1) Windows 2000 TCP/IP 试着在连接的网络上查找一个 DHCP 服务器,以得到一个动态分配的 IP 地址。

(2) 在启动过程中,如果没有 DHCP 服务器(如服务器停机或维修),客户机则无法得到一个 IP 地址。

(3) APIPA 生成一个 169.254.X.Y(这里 X.Y 是客户机的唯一标识符)形式的 IP 地址和 255.255.0.0 的子网掩码。如果此地址已被使用,在需要的情况下,APIPA 会选择另一个 IP 地址,重新选择地址的次数最多为 10 次。

**4. 配置并验证 TCP/IP**

步骤 1:配置静态 IP 地址。

(1) 右击桌面上的"网上邻居"图标,选择"属性"后会出现"网络和拨号连接"窗口。再右击想要配置的连接,默认情况下为"本地连接",选择"属性"。

(2) 在打开窗口中,选择"Internet 协议(TCP/IP)属性"。按图 3-38 所示配置 IP 地址。

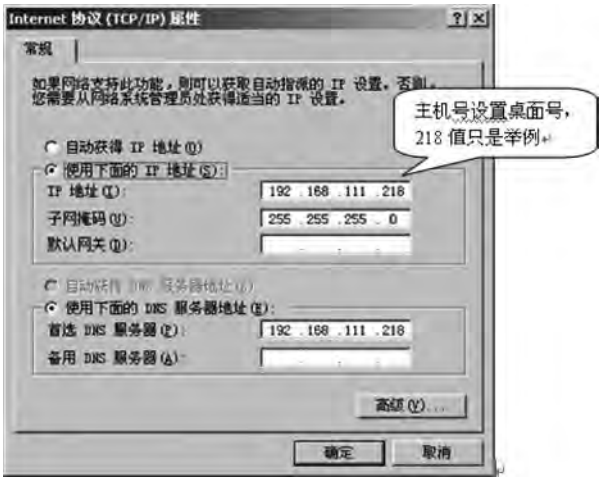



图 3-38　配置 IP 地址

步骤 2:验证计算机的 TCP/IP 配置。

在此步骤中,要使用两种 TCP/IP 实用程序 ipconfig 和 ping 来验证计算机的静态配置。

(1) 用 administrator 身份登录到服务器。

(2) 打开命令提示,即在"开始"菜单的"运行"对话框中输入 cmd 即可。

(3) 在命令提示中,输入 ipconfig/all,按 Enter 键,如图 3-39 所示。

(4) 要验证 IP 地址是否在正常工作,输入 ping 127.0.0.1。

(5) 要验证计算机是否和网络上其他计算机拥有相同的 IP 地址,则对本地计算机的 IP 地址使用 ping 命令。

步骤 3:配置 TCP/IP 自动获得一个 IP 地址。

在这一步中,要配置 TCP/IP,使其自动获得一个 IP 地址。然后验证配置,确认 APIPA 的确提供了适当的 IP 寻址信息。

(1) 在 TCP/IP 属性对话框中单击"自动获取 IP 地址"。

(2) 在命令提示中,输入 ipconfig/all|more,按回车键。


图 3-39　静态配置的验证

步骤 4：用 ping 命令验证本机与前后左右 4 台机器的 IP 地址连通情况。

## 任务 3　使用 Sniffer 抓取 ftp 口令并用安全的方式保护 FTP

### 1. 使用 Sniffer 抓取 ftp 口令

Sniffer 软件是 NAI 公司的协议分析软件，可运行在各种 Windows 平台上。

(1) 将 Sniffer 安装在本机上。

在进行流量捕获之前应选择网络适配器，确定从计算机的哪个网络适配器上接收数据。

(2) 打开并运行 Sniffer，如图 3-40 所示。


图 3-40　Sniffer 运行后显示的界面

(3) 选择“捕获”→“定义过滤器”→“地址类型”(由默认的 Hardware 改为 IP),并将位置设为 166.111.5.36—166.111.5.37,如图 3-41 所示。


图 3-41 设置地址位置

(4) 打开捕获按钮(或按 F10 键)开始捕获,如图 3-42 所示。

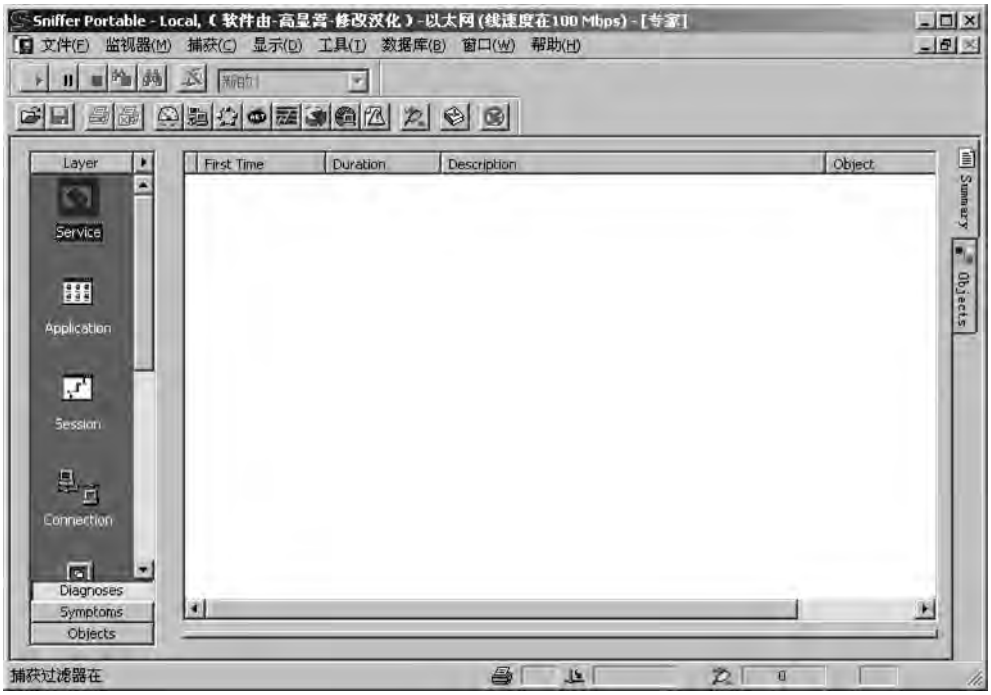



图 3-42 开始捕获

(5) 通过运行 FTP 访问 166.111.5.37,如图 3-43 所示。

(6) 停止捕获,并选择解码。结果显示访问的几次握手,如图 3-44 所示。

(7) 从捕获的信息中可以找到使用的 FTP 用户名(USER lisa)和密码(PASS 123456),如图 3-45 所示。

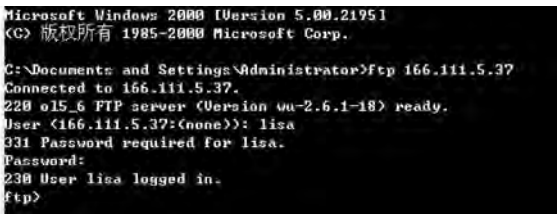



图 3-43　运行 FTP 访问 166.111.5.37

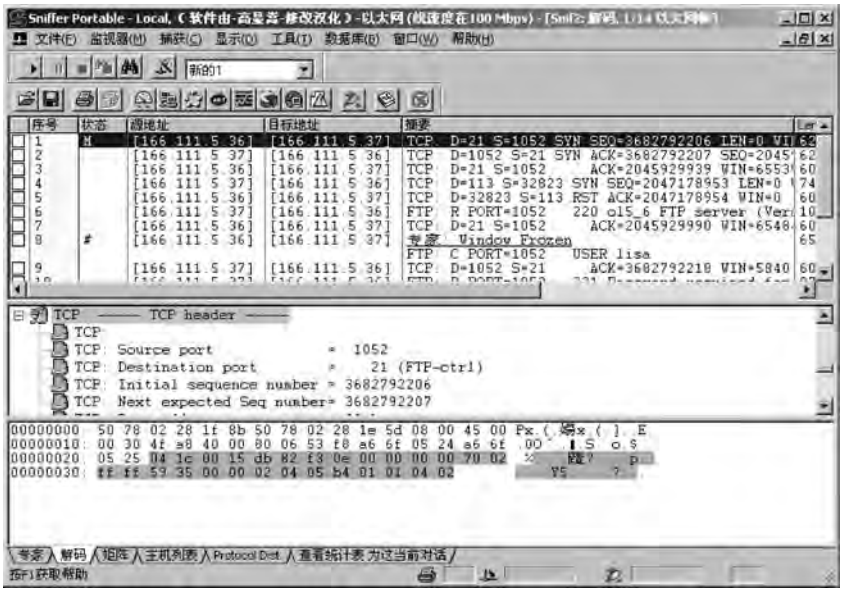



图 3-44　显示访问结果

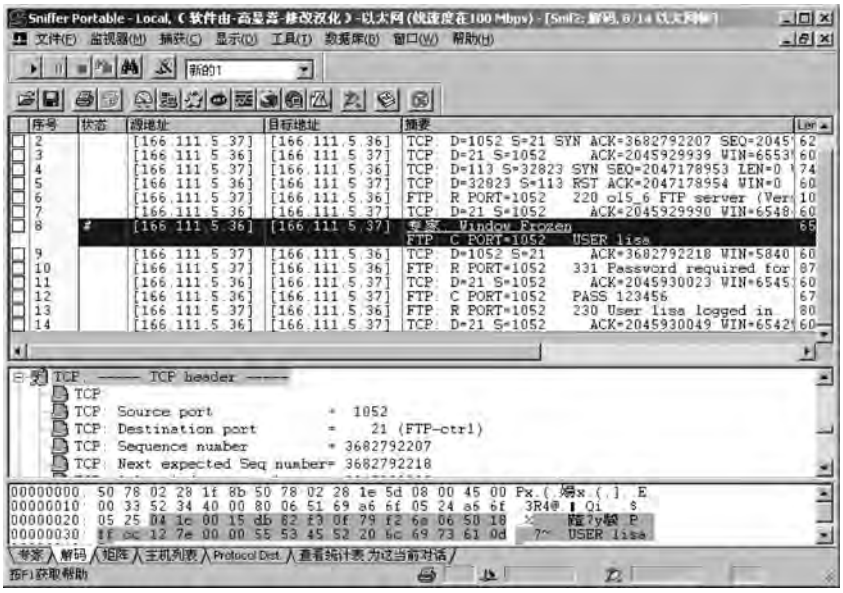



图 3-45　找到使用的 FTP 用户名和密码

至此，成功地捕获了信息。

**2. 使用 Sniffer 工具进行 TCP/IP、ICMP 数据包分析**

(1) 启动 Sniffer 软件，打开网络监视界面，如图 3-46 所示。


图 3-46　Sniffer 软件的网络监视界面

(2) 定义过滤器来捕捉 192.168.0.40 上的 IP 数据包，如图 3-47 和图 3-48 所示。


图 3-47　定义过滤器

(3) 从 Sniffer 软件的 Monitor 菜单中单击 Matrix 命令，显示 192.168.0.40 的通信情况，并通过右击该地址，在快捷菜单中单击 Capture 命令开始捕捉，如图 3-49 所示。


图 3-48　捕捉 IP 数据包

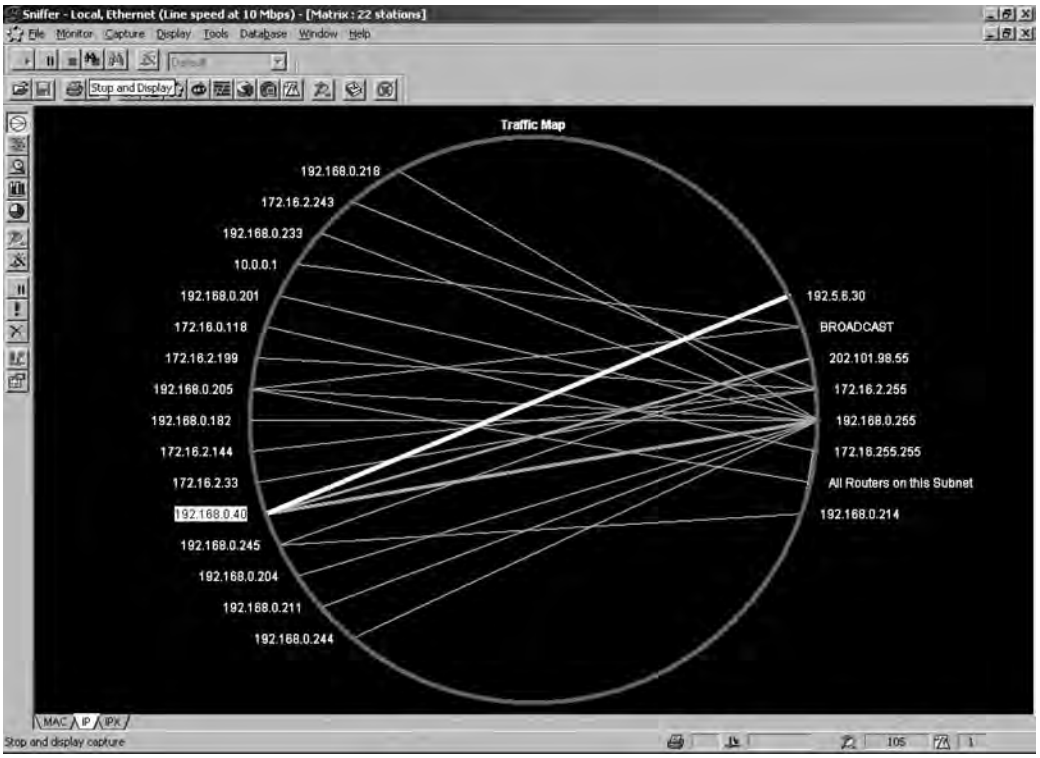



图 3-49　开始捕捉

(4) 一会儿停止捕捉后，选择 Decode 选项，查看捕捉到的 IP 包，如图 3-50 所示。

(5) 从图 3-51 中可以看出有 3 个显示框，最上面的显示框是捕捉的数据，中间的显示框是数据分析，最下面的显示框是原始数据包，用十六进制表示，例如，TCP：Source port＝1282 对应下面的 0502。

(6) 从显示框中可以看出：IP 数据包封装在 TCP 数据包的前面，如图 3-52 所示。


图 3-50　查看捕捉到的 IP 包

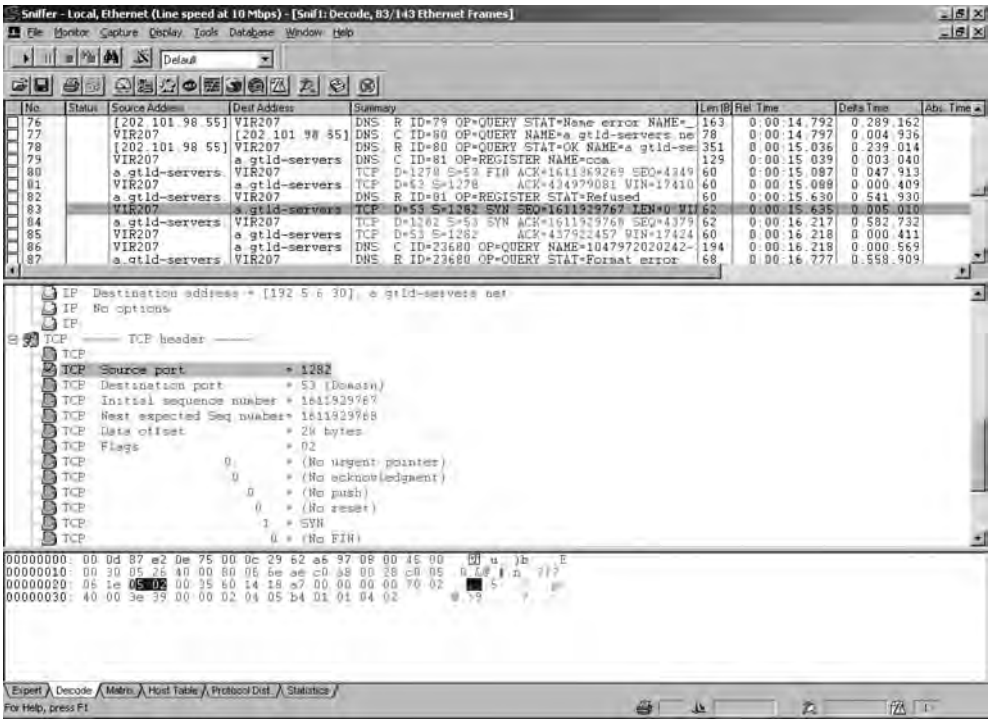



图 3-51　捕捉结果

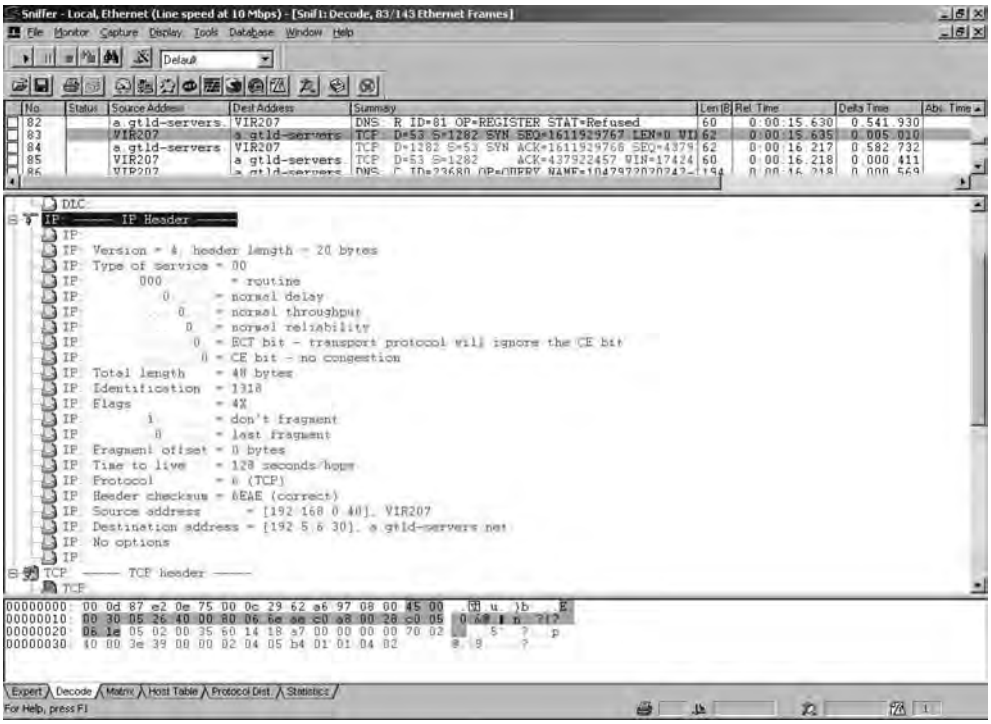



图 3-52　IP 数据包封装情况

（7）IP 数据包头的结构示意图，如图 3-53 所示。

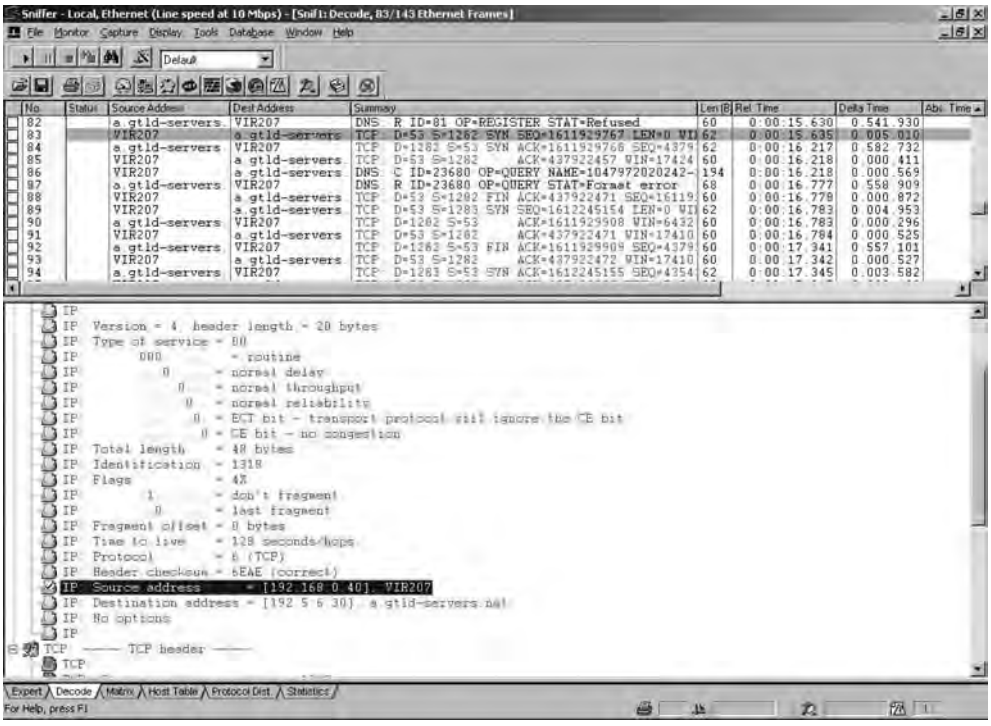



图 3-53　IP 数据包头的结构示意图

（8）TCP 的结构示意如图 3-54 所示。

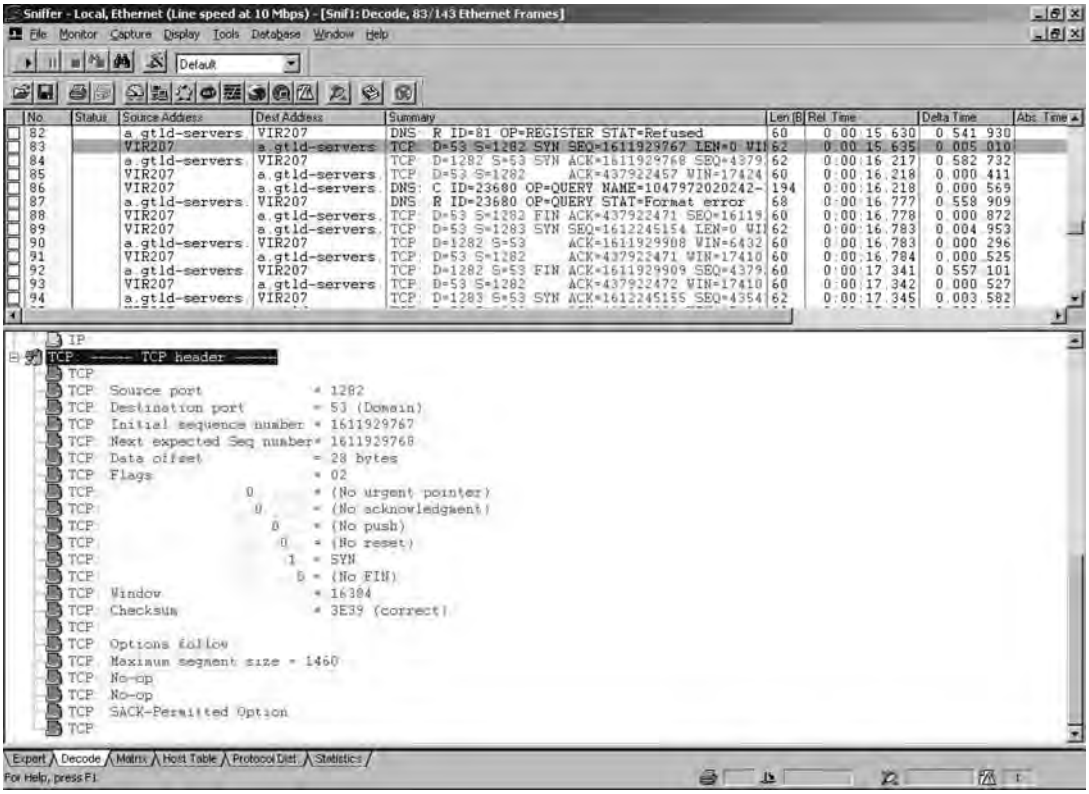



图 3-54　TCP 的结构示意图

（9）定义过滤器来捕捉 192.168.0.40 的 ICMP 数据包，如图 3-55 所示。


图 3-55　ICMP 数据包

(10) 从本机(192.168.0.245)PING 192.168.0.40,如图 3-56 和图 3-57 所示。

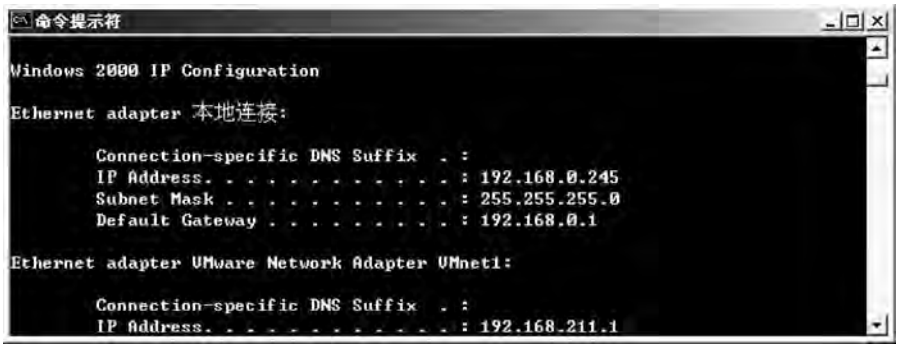



图 3-56　本机地址


图 3-57　源地址

(11) 停止捕捉后从 Decode 显示框中找出 Echo 及 Echo reply 数据包,如图 3-58 所示。

(12) 分析 ICMP 数据包头号信息,如图 3-59 所示。

显示结果如下。

ICMP 类型:8.
代码:0.
校验和:395C(正确).
确认号:1024.
序号:4096.
数据长度:32 字节.

### 3. 总结

(1) Sniffer 是一个强大的抓包工具,数据包分析功能强大,如果正确使用,对于分析、定位网络故障十分有用。

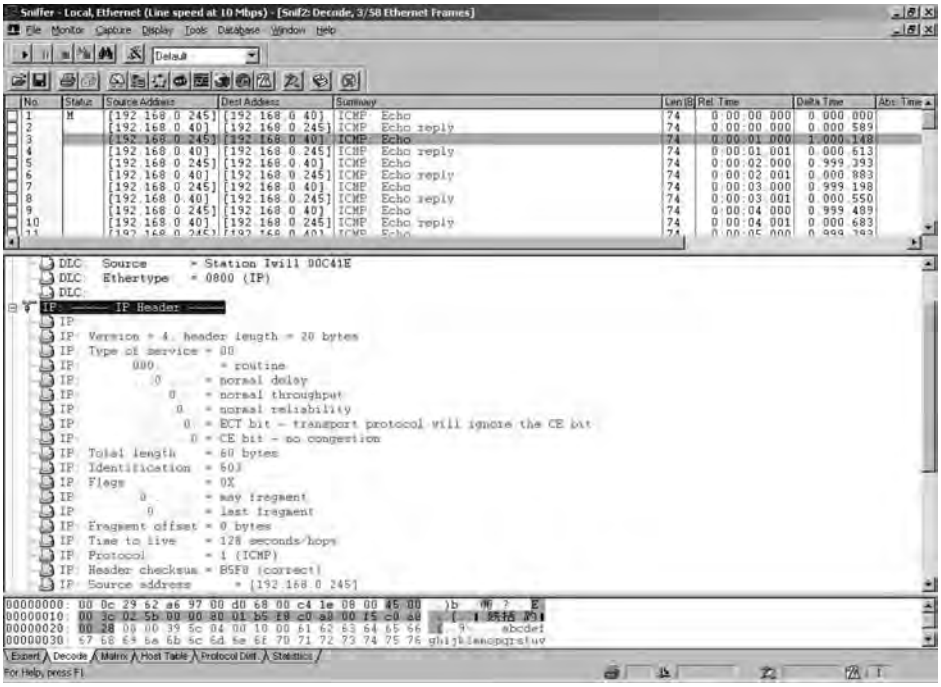



图 3-58　找出的 Echo 及 Echo reply 数据包

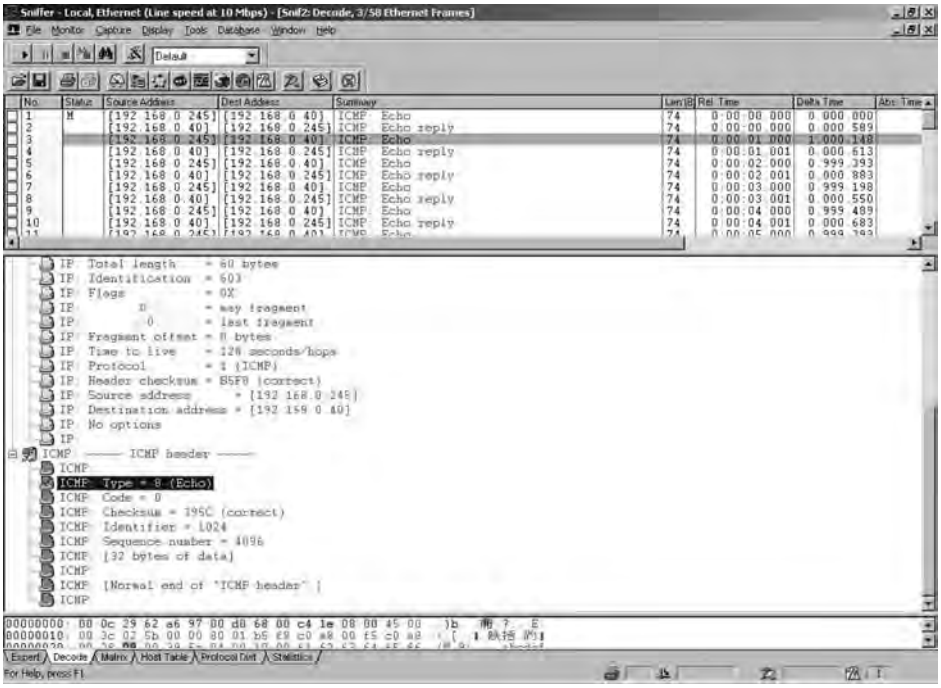



图 3-59　ICMP 数据包头号信息

（2）Sniffer 工具由于功能强大，甚至可以充当 Hacker 工具，因为很多协议是明文传输，如 FTP、Telnet 等，通过 Sniffer 工具可以查看用户名和密码。

(3) 从 OSI 结构上看,IP 包属于 3 层网络层,TCP 包是属于 4 层传输层,在数据包中 IP 头在 TCP 头的前面。

从实验中可以清晰地看出 TCP 的 3 次握手过程。

## 任务 4　网络协议分析工具 Wireshark(Ethereal)的使用

视频讲解

Wireshark 是非常流行的抓包软件,可截取各种网络封包,显示网络封包的详细信息。

### 1. 用 Wireshark 观察 ARP 以及 ping 命令的工作过程

(1) 用 ipconfig 命令获得本机的 MAC 地址和默认路由器的 IP 地址。

```
ipconfig/all 00-15-C5-7B-30-A6 192.168.1.1
```

(2) 用 arp 命令清空本机的缓存。

```
arp -d
```

(3) 运行 Wireshark,开始捕获所有属于 ARP 或 ICMP 的并且源或目的 MAC 地址是本机的包(提示:在设置过滤规则时需要使用(1)中获得的本机的 MAC 地址)。所得信息如图 3-60 所示。

| | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 1 | 0.000000 | Dell_7b:30:a6 | Tp-LinkT_23:e9:d4 | ARP | Who has 192.168.1.1? Tell 192.168.1.100 |
| 2 | 0.003925 | Tp-LinkT_23:e9:d4 | Dell_7b:30:a6 | ARP | 192.168.1.1 is at 00:21:27:23:e9:d4 |
| 3 | 13.049870 | 192.168.1.100 | 192.168.1.1 | ICMP | Echo (ping) request |
| 4 | 13.050437 | 192.168.1.1 | 192.168.1.100 | ICMP | Echo (ping) reply |
| 5 | 14.052357 | 192.168.1.100 | 192.168.1.1 | ICMP | Echo (ping) request |
| 6 | 14.052947 | 192.168.1.1 | 192.168.1.100 | ICMP | Echo (ping) reply |
| 7 | 15.052410 | 192.168.1.100 | 192.168.1.1 | ICMP | Echo (ping) request |
| 8 | 15.053008 | 192.168.1.1 | 192.168.1.100 | ICMP | Echo (ping) reply |
| 9 | 16.052171 | 192.168.1.100 | 192.168.1.1 | ICMP | Echo (ping) request |
| 10 | 16.052694 | 192.168.1.1 | 192.168.1.100 | ICMP | Echo (ping) reply |

图 3-60　属于 ARP 或 ICMP 并且源或目的 MAC 地址是本机的包

过滤规则:ether host 00:15:C5:7B:30:A6 and (arp or icmp)。

(4) 执行命令“ping 默认路由器的 IP 地址”。

```
ping 192.168.1.1
```

首先,通过 ARP 来找到所 ping 机器的 ip 地址,本机器发送一个广播包,在子网中查询 192.168.1.17 的 MAC 地址,然后一个节点发送了响应该查询的 ARP 分组,告知机器所查询的 MAC 地址。接下来,本机器发送了 3 个回显请求 ICMP 报文,目的地段回复了 3 个响应的回显应答 ICMP 报文,说明连通正常。

### 2. 用 Wireshark 观察 tracert 命令的工作过程

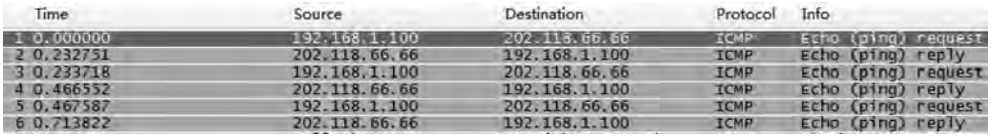
(1) 运行 Wireshark,开始捕获 tracert 命令中用到的消息。

(2) 执行 tracert-d www.dlut.edu.cn,如图 3-61 所示。

```
C:\Users\Administrator>tracert -d www.dlut.edu.cn

通过最多 30 个跃点跟踪
到 www.dlut.edu.cn [202.118.66.66] 的路由:

  1    232 ms    232 ms    246 ms  202.118.66.66

跟踪完成。
```

图 3-61　捕获结果

图 3-62 显示了 Wireshark 所观察到的现象。



| Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|
| 1 0.000000 | 192.168.1.100 | 202.118.66.66 | ICMP | Echo (ping) request |
| 2 0.232751 | 202.118.66.66 | 192.168.1.100 | ICMP | Echo (ping) reply |
| 3 0.233718 | 192.168.1.100 | 202.118.66.66 | ICMP | Echo (ping) request |
| 4 0.466552 | 202.118.66.66 | 192.168.1.100 | ICMP | Echo (ping) reply |
| 5 0.467587 | 192.168.1.100 | 202.118.66.66 | ICMP | Echo (ping) request |
| 6 0.713822 | 202.118.66.66 | 192.168.1.100 | ICMP | Echo (ping) reply |

图 3-62　Wireshark 所观察到的现象

由图 3-62 可见，本地主机发送了 3 次 ICMP 请求回显数据报，由于中途无别的路由器，直接到达目的端，目的主机随之发送相应的 ICMP 回显应答报文。

**3. 用 Wireshark 观察 TCP 连接的建立过程和终止过程**

(1) 启动 Wireshark，配置过滤规则为捕获所有源或目的是本机的 Telnet 协议的包(提示：Telnet 使用的传输层协议是 TCP，它使用 TCP 端口号 23)。

(2) 在 Windows 命令行窗口中执行命令 telnet bbs.dlut.edu.cn，登录后再退出。

过滤规则：host 192.168.1.100 and (tcp port 23)。

图 3-63 显示了 Wireshark 所观察到的现象，TCP 三次握手的连接建立过程如下。

| Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|
| 1 0.000000 | 192.168.1.100 | 202.118.66.5 | TCP | 56640 > telnet [SYN] Seq=0 Win=81 |
| 2 0.084982 | 202.118.66.5 | 192.168.1.100 | TCP | telnet > 56640 [SYN, ACK] Seq=0 Ac |
| 3 0.085206 | 192.168.1.100 | 202.118.66.5 | TCP | 56640 > telnet [ACK] Seq=1 Ack=1 W |
| 4 0.154696 | 202.118.66.5 | 192.168.1.100 | TELNET | Telnet Data ... |

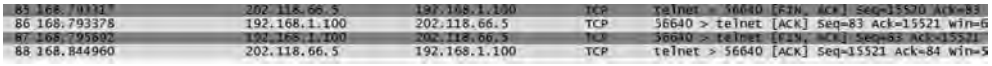
图 3-63　TCP 三次握手的过程

步骤 1，本地主机 192.168.1.100 向目的主机 202.118.66.5 发送一个 SYN＝1 的报文段。

步骤 2，目的主机收到该报文段后，回复一个允许连接的报文段，SYN 亦为 1。

步骤 3，收到上步报文段后，本地主机回复一个对其进行确认的报文段。

图 3-64 显示了 TCP 的连接终止过程时，Wireshark 所观察到的现象。



| | | | | |
|---|---|---|---|---|
| 85 168.78117 | 202.118.66.5 | 192.168.1.100 | TCP | Telnet > 56640 [FIN, ACK] Seq=15520 Ack=83 |
| 86 168.793378 | 192.168.1.100 | 202.118.66.5 | TCP | 56640 > telnet [ACK] Seq=83 Ack=15521 Win=6 |
| 87 168.795502 | 192.168.1.100 | 202.118.66.5 | TCP | 56640 > telnet [FIN, ACK] Seq=83 Ack=15521 |
| 88 168.844960 | 202.118.66.5 | 192.168.1.100 | TCP | telnet > 56640 [ACK] Seq=15521 Ack=84 Win=5 |

图 3-64　TCP 的连接终止过程

首先，目的主机 202.118.66.5 向本地主机 192.168.1.100 发送一个 FIN 比特置为 1 的 TCP 报文段。

接着本地主机收到上一步的报文段后回复一个确认报文段。

然后，目的主机发送其终止报文段，其 FIN 比特被置为 1。

最后，本地主机对该终止报文段进行确认，回复一个确认报文段。

由图 3-64 可知，目的主机首先发送了 FIN 比特为 1 的报文段，故是目的主机首先发起连接关闭。

**4. 用 Wireshark 观察使用 DNS 来进行域名解析的过程**

(1) 在 Windows 命令窗口中执行命令“nslookup”，按回车键，进入该命令的交互模式。

(2) 启动 Wireshark，配置过滤规则为捕获所有源或目的是本机的 DNS 协议中的包(提示：DNS 使用的传输层协议是 UDP，它使用 UDP 端口号 53)。

(3) 在提示符“>”下直接输入域名 www.dlut.edu.cn，解析它所对应的 IP 地址。

(4) 在提示符“>”下输入命令“set type＝mx”，设置查询类型为 MX 记录。

(5) 在提示符“>”下输入域名“tom.com”,解析它所对应的 MX 记录。

(6) 在提示符“>”下输入命令“set type=a”,恢复查询类型为 A 记录。

(7) 在提示符“>”下输入 MX 记录的查询结果,从而查出“tom.com”邮件服务器的 IP 地址。

(8) 在提示符“>”下输入“exit”,退出 nslookup 的交互模式。

Capture Filter 过滤规则为:host 192.168.1.100 and (udp port 53)。

根据图 3-65 所示的 Wireshark 所观察到的现象,解析域名“www.dlut.edu.cn”所对应 IP 地址的过程如下。

| Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|
| 1 0.000000 | 192.168.1.100 | 202.96.69.38 | DNS | Standard query A www.dlut.edu.cn.domain |
| 2 0.038859 | 202.96.69.38 | 192.168.1.100 | DNS | Standard query response A 60.19.29.24 |
| 3 0.040037 | 192.168.1.100 | 202.96.69.38 | DNS | Standard query AAAA www.dlut.edu.cn.domain |
| 4 0.057473 | 202.96.69.38 | 192.168.1.100 | DNS | Standard query response, No such name |

图 3-65　解析域名所对应 IP 地址的过程

首先,本地主机的 DNS 客户端向网络中发送一个查询“www.dlut.edu.cn”且资源记录类型为 A 的 DNS 查询报文。

接着,收到一个 DNS 回答报文,告知其 IP 地址为 60.19.29.24。

然后,本地 DNS 客户端又发送一个 RR 类型为 AAA 的查询报文。

最后,收到回答报文,告知无这样的域名。

根据图 3-66 所示的 Wireshark 所观察到的现象,解析域名“tom.com”所对应 MX 记录的过程如下。

| | | | | |
|---|---|---|---|---|
| 5 48.682971 | 192.168.1.100 | 202.96.69.38 | DNS | Standard query MX tom.com.domain |
| 6 48.699195 | 202.96.69.38 | 192.168.1.100 | DNS | Standard query response, No such name |
| 7 48.699824 | 192.168.1.100 | 202.96.69.38 | DNS | Standard query MX tom.com |
| 8 48.717208 | 202.96.69.38 | 192.168.1.100 | DNS | Standard query response MX 10 tommx.cdn.163.net |

图 3-66　解析域名所对应 MX 记录的过程

首先,本地主机的 DNS 客户端向网络中发送一个查询“tom.com.domain”且资源记录类型为 MX 的 DNS 查询报文。

接着,收到一个 DNS 回答报文,告知无这样的域名。

然后,本地主机的 DNS 客户端又发送一个查询“tom.com”且资源记录类型为 MX 的 DNS 查询报文。

最后,收到一个 DNS 回答报文,告知其规范主机名为“tommx.cdn.163.net”。

在“tom.com”域中,显示了邮件服务器和 IP 地址的信息,如图 3-67 所示,有 2 个邮件服务器,它们的 IP 地址分别是:202.108.252.141 和 202.108.252.210。

```
> set type=mx
> tom.com
服务器:  UnKnown
Address:  202.96.69.38

非权威应答:
tom.com MX preference = 10, mail exchanger = tommx.cdn.163.net

tom.com nameserver = ns2.tom.com
tom.com nameserver = ns1.tom.com
tom.com nameserver = ns3.tom.com
tom.com nameserver = ns4.tom.com
tom.com nameserver = ns5.tom.com
tommx.cdn.163.net       internet address = 202.108.252.141
tommx.cdn.163.net       internet address = 202.108.255.210
ns2.tom.com     internet address = 61.135.159.47
ns3.tom.com     internet address = 211.100.41.31
ns4.tom.com     internet address = 211.100.41.31
ns5.tom.com     internet address = 202.108.12.120
ns1.tom.com     internet address = 61.135.159.46
```

图 3-67　显示邮件服务器和 IP 地址的信息




## IPv6

**1. IPv6 简介**

现行 TCP/IP 体系结构中的网际协议是 IPv4(即网际协议版本 4),作为 Internet 的关键协议在全球获得了巨大的成功,但是随着 Internet 用户的快速增长和传输速率的不断发展,IPv4 面临着诸如地址不够用、难以很好地支持新业务等不容忽视的危机。为了解决 IPv4 所存在的不足,IETF 提出了一种新的网际协议,它就是 IPv6(Internet Protocol Version 6)。IPv6 又被称为下一代因特网协议。

IPv6 保留了 IPv4 的大多数选项(如分段和源端路由选择等),同时在许多方面提出了改进,例如路由方面、自动配置方面等,与 IPv4 相比,IPv6 的优势可表现在以下几个方面。

(1) 更大的地址空间。

IPv6 将当前的地址空间由 32 比特扩展到 128 比特,以支持大量的网络节点。根本上解决了地址壁垒问题,有利于向企业网和家庭网延伸。IPv6 的地址总数大约有 $3.4\times10^{38}$ 个。IPv6 采用了层次化的地址结构,把它的地址空间按照不同的地址前缀来划分,以利于骨干网路由器对数据包的快速转发。IPv6 中取消了广播地址而代之以任意点通信地址。而 IPv4 中用于指定一个网络接口的单点通信地址和用于指定由一个或多个主机侦听的多点通信地址基本不变。

(2) 更好的报头格式。

IPv6 的报头由一个基本报头和多个扩展报头构成,由于对数据报头做了简化,从而减少了处理器开销。IPv6 使用了固定格式的包头并减少了需要检查和处理的字段的数量,有助于加快路由速度,提高选路的效率。

通过定义多种扩展报头,使得 IPv6 变得极其灵活,提供对多种应用的支持,同时又为以后支持新的应用提供了可能。扩展报头必须按照它们在包中出现的次序依次处理。这些扩展报头有逐个路程段选项报头、分段报头、路由报头、身份认证报头、目的选项报头、有效载荷安全封装报头、最终目的报头等。

(3) 更高的安全特性。

随着 IPv6 的使用,大量设备接入到网络中对安全性提出了更高的要求。IPv6 的安全性就是主要通过 IPSec 架构来实现的。内置 IPSec 以及发送设备有永久地址后不仅可以实现端到端加密,而且解决了网络层溯源问题,给网络安全提供了根本解决措施。IPSec 的主要功能是在网络层对数据分组提供加密和鉴别等安全服务,它提供了两种安全机制:认证和加密。认证机制使 IP 通信的数据接收方能够确认数据发送方的真实身份以及数据在传输过程中是否遭到改动。加密机制通过对数据进行编码来保证数据的机密性,以防数据在传输过程中被他人截获而失密。

(4) 便捷的联网方式。

IPv6 具有自动将 IP 地址分配给用户的功能。只要设备一连接上网络便可自动设定地址。设备即插即用,能自动发现和自动配置,并且改善了路由性能和效率,实现了路由聚合、分层编址等。IPv6 具有两种自动设定功能:一种是和 IPv4 自动设定功能一样的名为“全状

态自动设定”；另一种是“无状态自动设定”。

(5) 改善了服务质量。

基于 IPv4 的 Internet 只有一种简单的服务质量，即“尽力而为”传输。IPv6 对 QoS 能力有所增强，能提供不同水平的服务质量。

IPv6 在报头中保留了类似 IPv4 的 TOS 域，称为传输级别域，以继续为 IP 提供有差别的 QoS 服务，同时 IPv6 报头中增加了一个 8 位的业务流类别和一个新的 20 位的流标签。其目的是允许发送业务流的源节点和转发业务流的路由器在数据包上加上标记，并进行除默认处理之外的特殊处理。流标签可以更好地支持综合 Qos 服务，可实现 PTP(在线游戏和聊天等)业务，“流”指的是从一个特定源发向一个特定(单播或是组播)目的地的包序列。

(6) 支持移动 IP。

移动通信与因特网的结合是网络发展的趋势之一。移动因特网将成为我们日常生活的一部分，在各个方面改变我们的生活。移动因特网不仅仅是移动接入因特网，它还提供一系列以移动性为核心的增值业务(如查询本地化设计信息、远程控制工具、无线互动游戏等)。

IPv4 没有足够的地址空间为移动 IP 中的每个设备提供一个全球唯一的 IP 地址。移动 IPv6 的设计汲取了移动 IPv4 的设计经验，并且利用了 IPv6 的许多新的特征，移动 IPv6 能够通过简单的扩展，满足大规模移动用户的需求，从而解决了在全球范围内网络和访问技术之间的移动性问题。移动终端在不改变自身 IP 地址的前提下实现不同接入媒质(如 3G 和 WLAN)间的自由移动。对比移动 IPv4，移动 IPv6 通过邻居发现自动配置等技术可以直接实现外地网络的发现以及转交地址的获得，而不必依赖于外地代理的存在；同时利用路由扩展头以及新增的本地地址等选项优化了报文路径，减少迂回路由，以及解决源地址过滤等问题，并使移动节点的应用层对转交地址进行透明处理，实现无缝的移动。

**2. IPv6 数据报格式**

IPv6 数据报格式由基本数据报头，扩展报头和高层数据 3 部分组成。

(1) IPv6 基本数据报头。

在 IPv6 中，报头以 64 比特为单位，基本报头的总长度固定为 40 字节。IPv6 基本报头的内容和格式如图 3-68 所示，下面对数据报头中定义的各字段做一简单介绍。


图 3-68　IPv6 数据报头格式

版本：长度为 4 位，指明协议版本，对于 IPv6，该字段必须为 6。

类别：长度为 8 位，定义该报提供何种“区分服务”。在最新的 IPv6 Internet 草案中称为“业务流类别”。该字段的定义独立于 IPv6，目前尚未在任何 RFC 中定义。该字段的默

认值是全 0。

流标识：长度为 20 位，用于标识属于同一业务流的包。一个节点可以同时作为多个业务流的发送源。流标识和源节点地址唯一地标识了一个业务流。也就是说，所有属于同一个流的分组具有相同的源地址、目的地址、选项和优先级。流标识可用来加速路由器对分组的处理，当路由器收到一个分组时，它不用进行路由选择，就可以很容易地在流标识表中找到下一跳的 IP 地址。流标识可用来支持(数字形式)实时音视频的传输。

有效载荷长度：长度为 16 位，其中包括包净荷的字节长度，即 IPv6 报头后的数据包中包含的字节数。在计算有效载荷长度时，包含 IPv6 扩展头的长度。

下一报头：长度为 8 位，若该数据有附加的扩展头，则这个字段用来指明 IPv6 头后所跟头字段中的协议类型。若无附加的扩展头，下一报头字段则可以用来指出高层是 TCP 还是 UDP。

跳数限制：即转发上限，长度为 8 位。该字段是为防止数据报传输过程中无休止地循环下去而设定的。该字段首先被初始化，每当一个节点对包进行一次转发之后，这个字段就会被减 1。如果该字段达到 0 仍未到达目的端，就丢弃该数据包。IPv4 中有一个具有类似功能的生存期字段，但在 IPv6 中，对过期包进行超时判断的功能可以由高层协议完成。

源地址：长度为 128 位，指出了 IPv6 包的发送方 IP 地址。

目的地址：长度为 128 位，指出了 IPv6 包的接收方 IP 地址。这个地址可以是一个单点通信、多点通信或任意点通信地址。如果使用了选路扩展头(其中定义了一个包必须经过的特殊路由)，其目的地址可以是其中某一个中间节点的地址而不必是最终地址。

(2) IPv6 扩展报头结构。

每一个扩展报头都由若干个域组成，它们的长度各不相同。但每个扩展报头的第一个域都应是长度为 8 位的“下一报头”域，当 IPv6 分组使用多个扩展报头时，它们应按照一定的顺序出现，即逐跳扩展头总是出现在最前面，而目的地扩展头总是出现在最后面。

**3. IPv6 的地址结构**

(1) 地址的类型。

IPv6 支持单点通信地址、多点通信地址和任意点通信地址 3 种不同类型的网络地址。所有类型的 IPv6 地址都是属于接口而不是节点。一个 IPv6 单点通信地址被赋给某一个接口，而一个接口又只能属于某一个特定的节点，因此一个节点的任意一个接口的单点通信地址都可以用来标识该节点。

在 IPv6 中有多种单点通信地址形式，包括基于全局提供者的单点通信地址、基于地理位置的单点通信地址、NSAP 地址、IPX 地址、节点本地地址、链路本地地址和兼容 IPv4 的主机地址等。IPv6 中的单点通信地址是连续的，适用于点到点通信时使用，一个标识符仅标识一个接口的情况。

IPv6 多点通信地址用于表示一组节点。多点通信地址是一个地址标识符对应多个接口的情况(通常属于不同节点)。该组节点可能会属于几个多点通信地址。这个功能被多媒体应用程序所广泛使用，它们需要一个节点到多个节点的传输。当分组的目的地址是多点地址时，网络尽力将分组发到该组的所有接口上。

任意点通信地址也属于一个标识符对应多个接口的情况。如果一个报文要求被传送到一个任意点通信地址，则它将被传送到由该地址标识的一组接口中的最近一个(根据路由选

择协议距离度量方式决定)。当用户发送一个数据包到这个任意点通信地址时,离用户最近的一个服务器将响应用户。这对于一个经常移动和变更的网络用户大有益处。

(2) 地址表示方法。

IPv6 的地址长度扩展到 128 位,为表示和理解方便,用冒号将其分割成 8 个字段,一个字段由 16 位二进制数组成。每个字段的最大值为 16 384,在书写时用 4 位的十六进制数字表示,字段与字段之间用":"隔开,而不是原来的"."。如:

1324:0000:0000:0000:0009:0900:100B:516A

字段中最高位为 0 的数值可以省略,如果整个字段为 0,那么也可以省略。于是上例可以缩写成:

1324:0:0:0:9:900:100B:516A

128 位地址空间包含的准确地址数是 340,282,366,920,938,463,463,374,607,431,768,211,456。

IPv6 地址前缀的表示方法类似于 CIDR 中 IPv4 的地址前缀表示法,可以表示为:

IPv6 地址=前缀+接口标识(类似 IPv4 网络号)+主机号

IPv6 的地址如图 3-69 所示。"FP"是地址前缀(也称为"格式前缀"),用于区别其他地址类型。随后是 13 位的顶级聚集体 ID 号(TLA ID)、8 位的 Res(保留位,以备将来 TLA 或 NLA 扩充之用)、24 位的次级聚集体 ID 号(NLA ID)、16 位的节点 ID 号(SLA ID)和 64 位的主机接口 ID 号(Interface ID)。TLA、NLA、SLA 三者构成了自顶向下排列的三个网络层次,并且依次向上一级申请 ID 号。分层结构的最底层是网络主机。

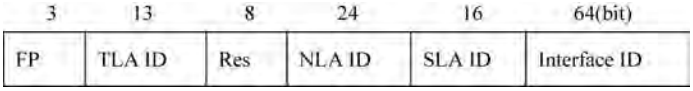



图 3-69 IPv6 的地址

IPv6 地址的基本表达方式是 x:x:x:x:x:x:x:x,其中 x 是一个 4 位十六进制整数(16 位)。每一个数字包含 4 位,每个整数包含 4 个数字,每个地址包括 8 个整数,共计 128 位(4×4×8=128)。例如:

1020.23AB.C2631.10DC.5680.6731.E129.ACB1

7952.9630.85CB.35C0.61BA.9103.EDA6.6565

这些整数都是十六进制数,它们都是合法的 IPv6 地址。地址中的每个整数都要表示出来,但起始的 0 可以不必表示。这是一种比较标准的 IPv6 地址表达方式,此外还有另外两种更加清楚和易于使用的方式。某些 IPv6 地址中可能包含一长串的 0,当出现这种情况时,标准中允许用"空隙"来表示这一长串的 0。如地址 3000:0:0:0:0:0:0:1 可以被表示为 3000::1,这两个冒号表示该地址可以扩展到一个完整的 128 位地址。在这种方法中,只有当 16 位组全部为 0 时才会被两个冒号取代,且两个冒号在地址中只能出现一次。

在 IPv4 和 IPv6 的混合环境中有第三种表示方法。IPv6 地址中的最低 32 位可以用于表示 IPv4 地址,该地址可以按照一种混合方式表达,即 x:x:x:x:x:x:d.d.d.d,其中 x 表示一个十六进制整数,而 d 表示一个十进制整数。

例如,地址 0:0:0:0:0:0:10.0.0.1 就是一个合法的 IPv6 地址。把两种可能的表达方式组合在一起,该地址也可以表示为::10.0.0.1。

(3) 地址的配置。

支持无状态和有状态两种地址自动配置方式是 IPv6 的一个基本特性。无状态地址自动配置方式是获得地址的关键。自动将 IP 地址分配给用户是 IPV6 标准功能,只要机器一连接上网络便可自动设定地址。IPv6 有两种自动设定功能,一种是和 IPv4 自动设定功能一样的名为“全状态自动设定”功能,另一种是“无状态自动设定”功能。

使用无状态自动配置,无须手动干预就能够改变网络中所有主机的 IP 地址。例如,当企业更换了连入 Internet 的 ISP 时,将从新 ISP 处得到一个新的可聚集全球地址前缀。ISP 把这个地址前缀从它的路由器上传送到企业路由器上。由于企业路由器将周期性地向本地链路中的所有主机多点传送路由器公告,因此企业网络中的所有主机都将通过路由器公告收到新的地址前缀。此后,它们就会自动产生新的 IP 地址并覆盖旧的 IP 地址。使用 DHCPv6 进行地址自动设定,连接于网络的机器需要查询自动设定用的 DHCP 服务器才能获得地址及其相关配置。但有两种情况应当以使用无状态自动设定方法为宜:一是家庭网络,因为家庭网络中通常没有 DHCP 服务器;二是移动环境,因为在移动环境中往往是临时建立的网络。

(4) 地址空间。

IPv6 中包含了地址分配的不同类型、前缀(地址分配中前面的位值)和作为整个地址空间一部分的地址分配的长度。地址划分采用多级体系,地址空间被划分成大小不同的地址块,以便于路由器更快地查找路由。

IPv6 又被称为下一代因特网协议,凭借其丰富的地址资源以及支持动态路由机制等优势,能够满足物联网对通信网络在地址、网络自组织以及扩展性等诸多方面的要求。然而,在物联网中 IPv6 并不能简单地“拿来就用”,而是需要进行一次适配。IPv6 在应用于传感器设备之前,需要对 IPv6 协议栈和路由机制进行相应的精简,以满足对网络低功耗、低存储容量和低传输速率的要求。IETF(互联网工程任务组)成立的 6LoWPAN 就是研究并推动对 IPv6 进行精简的标准化组织。另外,目前基于 IEEE 802.15.4 的网络射频芯片也有待进一步开发以支持 IPv6 协议栈进行精简。

## 小　　结

计算机网络中不同系统的两实体间只有在通信的基础上才能相互交换信息,共享网络资源。两个实体间要想进行正常的通信,就必须能够相互理解,共同遵守有关实体的某种互相都能接受的规则,我们把这些为进行网络中的数据交换而建立的规则的集合称为协议。

网络协议的作用是约束计算机之间的通信过程,使之按照事先约定好的步骤进行。

目前国际网络标准是 OSI 参考模型,OSI 将计算机网络分为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层 7 个层次,使简化网络和异构系统互联成为可能,但 OSI 模型设计时有一定的缺陷,实现起来较困难。另外,还有一个网络协议 TCP/IP,由于使用非常广泛,以至成为目前国际上事实的网络标准。

Internet 是一个全球范围的由众多网络连接而成的互联网,所使用的协议是 TCP/IP。其中网际协议(IP)是用于互联许多计算机网络进行通信的协议,因此这一层也常常称为网际层,或 IP 层。在这一层中有地址解析协议(ARP)、逆地址解析协议(RARP)、Internet 控

制报文协议(ICMP)三个协议与 IP 配套使用。

目前 Internet 都采用子网划分的方式。子网划分是把单个网络细化为多个规模更小的网络的过程,使得多个小规模物理网络可以使用路由器互联起来并且共有同一个网络号。

一个网络中的所有主机必须有相同的网络号,一个单位分配到的 IP 地址是 IP 地址的网络号,而后面的主机号则由本单位进行分配。本单位所有的主机都使用同一个网络号。

通过把 IP 地址的主机号字段划分成两个字段,就能够建立子网地址。这时必须用子网掩码来确定如何进行这样的划分,即要确切指出子网号字段需要多少位进行编码。子网中的每台主机都要指定子网掩码,而且子网中的所有主机必须配置相同的子网掩码。

将一台计算机的 IP 地址翻译成等价的硬件地址的过程就叫地址转换,这个转换是由地址解析协议(ARP)来完成的。

TCP/IP 在传输层有两个不同的协议:传输控制协议(TCP)和用户数据报协议(UDP)。TCP 的数据传输单位是 TCP 报文段,UDP 的数据传输单位是 UDP 报文或用户数据报。

TCP 即传输控制协议,它是一个面向连接的传输层协议,是传输层中使用最广泛的一个协议。TCP 是提供可靠通信的传输层协议,一旦数据报被破坏或丢失,将其重新传输的工作则由 TCP 而非高层应用程序来完成。TCP 也会检测传输错误并予以修正。TCP 提供可靠的全双工数据传输服务。

UDP 是一个无连接的协议,无连接的通信不提供可靠性,即不通知发送端口是否正确接收了报文。无连接协议也不提供错误恢复能力。UDP 比 TCP 要简单得多,它可以简单地与 IP 或其他协议连接,只充当数据报的发送者或接收者。

## 思考与练习

1. 试述网络的协议和体系结构的概念。
2. 计算机网络体系结构采用层次结构有什么特点?
3. OSI 参考模型有几层? 各层有什么功能?
4. TCP/IP 体系结构由哪几层构成? 各层有哪些主要协议?
5. IP 地址的结构是怎样的? IP 地址可以分为哪几种?
6. 试说明 IP 地址与硬件地址的区别。
7. 子网掩码的用途是什么? 对于 A、B、C 三类网络,其默认子网掩码是什么?
8. 对于子网掩码为 255.255.255.224 的 C 类网络,能创建多少个子网?
9. 对于子网掩码为 255.255.248.0 的 B 类网络,一个子网能连接多少台主机?