第1章 概 述

1.1 引言

很多不同的厂家生产各种型号的计算机，它们运行完全不同的操作系统，但 TCP/IP协议族允许它们互相进行通信。这一点很让人感到吃惊，因为它的作用已远远超出了起初的设想。TCP/IP起源于60年代末美国政府资助的一个分组交换网络研究项目，到 90年代已发展成为计算机之间最常应用的组网形式。它是一个真正的开放系统，因为协议族的定义及其多种实现可以不用花钱或花很少的钱就可以公开地得到。它成为被称作“ 全球互联网 ”或“ 因特网(Internet) ”的基础，该广域网（WAN）已包含超过100万台遍布世界各地的计算机。

本章主要对TCP/IP协议族进行概述，其目的是为本书其余章节提供充分的背景知识。如果读者要从历史的角度了解有关 TCP/IP的早期发展情况，请参考文献 [Lynch 1993]。

1.2 分层

网络协议通常分不同层次进行开发，每一层分别负责不同的通信功能。一个协议族，比如 TCP/IP，是一组不同层次上的多个协议的组合。 TCP/IP通常被认为是一个四层协议系统，如图 1-1所示。

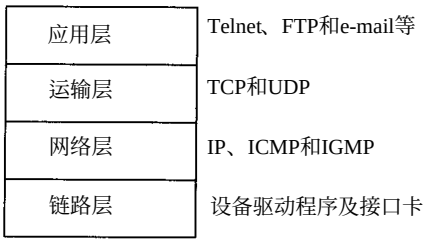

图1-1 TCP/IP协议族的四个层次

每一层负责不同的功能：

1) 链路层，有时也称作数据链路层或网络接口层，通常包括操作系统中的设备驱动程序和计算机中对应的网络接口卡。它们一起处理与电缆（或其他任何传输媒介）的物理接口细节。
2) 网络层，有时也称作互联网层，处理分组在网络中的活动，例如分组的选路。在TCP/IP协议族中，网络层协议包括 IP协议（网际协议），ICMP协议（Internet互联网控制报文协议），以及IGMP协议（Internet组管理协议）。
3) 运输层主要为两台主机上的应用程序提供端到端的通信。在 TCP/IP协议族中，有两个互不相同的传输协议：TCP（传输控制协议）和UDP（用户数据报协议）。
   TCP为两台主机提供高可靠性的数据通信。它所做的工作包括把应用程序交给它的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置发送最后确认分组的超时时钟等。由于运输层提供了高可靠性的端到端的通信，因此应用层可以忽略所有这些细节。
   而另一方面， UDP则为应用层提供一种非常简单的服务。它只是把称作数据报的分组从一台主机发送到另一台主机，但并不保证该数据报能到达另一端。任何必需的可靠性必须由应用层来提供。
   这两种运输层协议分别在不同的应用程序中有不同的用途，这一点将在后面看到。
4) 应用层负责处理特定的应用程序细节。几乎各种不同的 TCP/IP实现都会提供下面这些通用的应用程序：

- Telnet 远程登录。
- FTP 文件传输协议。
- SMTP 简单邮件传送协议。
- SNMP 简单网络管理协议。

另外还有许多其他应用，在后面章节中将介绍其中的一部分。

假设在一个局域网（LAN）如以太网中有两台主机，二者都运行 FTP协议，图1-2列出了该过程所涉及到的所有协议。

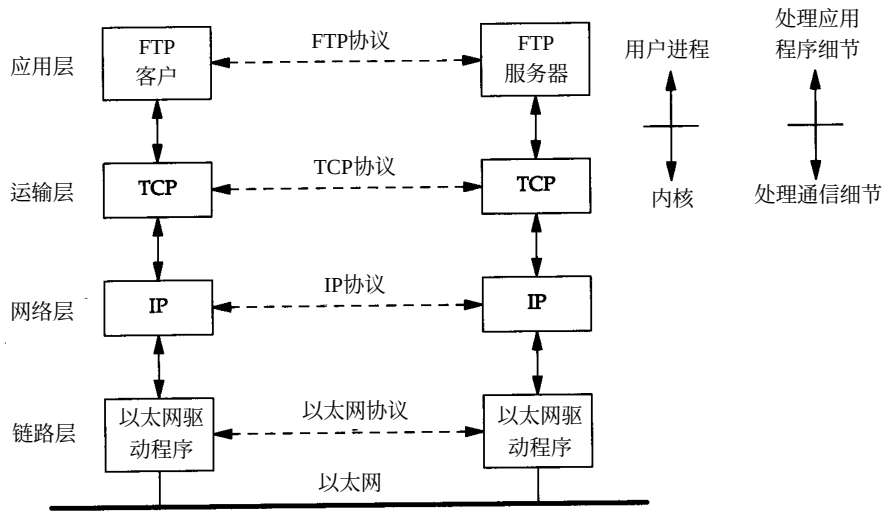

图1-2 局域网上运行FTP的两台主机

这里，我们列举了一个FTP客户程序和另一个FTP服务器程序。大多数的网络应用程序都被设计成客户—服务器模式。服务器为客户提供某种服务，在本例中就是访问服务器所在主机上的文件。在远程登录应用程序Telnet中，为客户提供的服务是登录到服务器主机上。

在同一层上，双方都有对应的一个或多个协议进行通信。例如，某个协议允许 TCP层进行通信，而另一个协议则允许两个IP层进行通信。

在图1-2的右边，我们注意到应用程序通常是一个用户进程，而下三层则一般在（操作系统）内核中执行。尽管这不是必需的，但通常都是这样处理的，例如 UNIX操作系统。

在图1-2中，顶层与下三层之间还有另一个关键的不同之处。应用层关心的是应用程序的细节，而不是数据在网络中的传输活动。下三层对应用程序一无所知，但它们要处理所有的通信细节。

在图1-2中列举了四种不同层次上的协议。 FTP是一种应用层协议， TCP是一种运输层协议， IP是一种网络层协议，而以太网协议则应用于链路层上。 TCP/IP协议族是一组不同的协议组合在一起构成的协议族。尽管通常称该协议族为 TCP/IP，但TCP和IP只是其中的两种协议而已（该协议族的另一个名字是Internet协议族(Internet Protocol Suite)）。

网络接口层和应用层的目的是很显然的 —— 前者处理有关通信媒介的细节（以太网、令牌环网等），而后者处理某个特定的用户应用程序（ FTP、Telnet等）。但是，从表面上看，网络层和运输层之间的区别不那么明显。为什么要把它们划分成两个不同的层次呢？为了理解这一点，我们必须把视野从单个网络扩展到一组网络。

在80年代，网络不断增长的原因之一是大家都意识到只有一台孤立的计算机构成的“孤岛”没有太大意义，于是就把这些孤立的系统组在一起形成网络。随着这样的发展，到了90年代，我们又逐渐认识到这种由单个网络构成的新的更大的“岛屿”同样没有太大的意义。于是，人们又把多个网络连在一起形成一个网络的网络，或称作互连网(internet)。一个互连网就是一组通过相同协议族互连在一起的网络。

构造互连网最简单的方法是把两个或多个网络通过路由器进行连接。它是一种特殊的用于网络互连的硬件盒。路由器的好处是为不同类型的物理网络提供连接：以太网、令牌环网、点对点的链接和FDDI（光纤分布式数据接口）等等。

这些盒子也称作IP路由器（IP Router），但我们这里使用路由器(Router)这个术语。从历史上说，这些盒子称作网关（gateway），在很多TCP/IP文献中都使用这个术语。现在网关这个术语只用来表示应用层网关：一个连接两种不同协议族的进程（例如，TCP/IP和IBM的SNA），它为某个特定的应用程序服务（常常是电子邮件或文件传输）。

图1-3是一个包含两个网络的互连网：一个以太网和一个令牌环网，通过一个路由器互相连接。尽管这里是两台主机通过路由器进行通信，实际上以太网中的任何主机都可以与令牌环网中的任何主机进行通信。

在图1-3中，我们可以划分出端系统（End system）（两边的两台主机）和中间系统（Intermediate system）（中间的路由器）。应用层和运输层使用端到端（End-to-end）协议。在图中，只有端系统需要这两层协议。但是，网络层提供的却是逐跳（Hop-by-hop）协议，两个端系统和每个中间系统都要使用它。

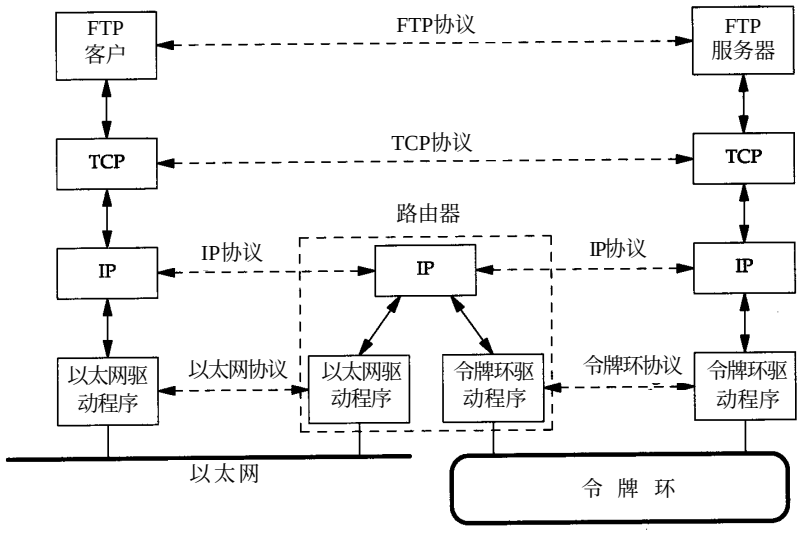

图1-3 通过路由器连接的两个网络

在TCP/IP协议族中，网络层IP提供的是一种不可靠的服务。也就是说，它只是尽可能快地把分组从源结点送到目的结点，但是并不提供任何可靠性保证。而另一方面，TCP在不可靠的IP层上提供了一个可靠的运输层。为了提供这种可靠的服务，TCP采用了超时重传、发送和接收端到端的确认分组等机制。由此可见，运输层和网络层分别负责不同的功能。

从定义上看，一个路由器具有两个或多个网络接口层（因为它连接了两个或多个网络）。

任何具有多个接口的系统，英文都称作是多接口的(multihomed)。一个主机也可以有多个接口，但一般不称作路由器，除非它的功能只是单纯地把分组从一个接口传送到另一个接口。同样，路由器并不一定指那种在互联网中用来转发分组的特殊硬件盒。大多数的 TCP/IP实现也允许一个多接口主机来担当路由器的功能，但是主机为此必须进行特殊的配置。在这种情况下，我们既可以称该系统为主机（当它运行某一应用程序时，如 FTP或Telnet），也可以称之为路由器（当它把分组从一个网络转发到另一个网络时）。在不同的场合下使用不同的术语。

互联网的目的之一是在应用程序中隐藏所有的物理细节。虽然这一点在图 1-3由两个网络组成的互联网中并不很明显，但是应用层不能关心（也不关心）一台主机是在以太网上，而另一台主机是在令牌环网上，它们通过路由器进行互连。随着增加不同类型的物理网络，可能会有20个路由器，但应用层仍然是一样的。物理细节的隐藏使得互联网功能非常强大，也非常有用。

连接网络的另一个途径是使用网桥。网桥是在链路层上对网络进行互连，而路由器则是在网络层上对网络进行互连。网桥使得多个局域网（ LAN）组合在一起，这样对上层来说就好像是一个局域网。

TCP /IP倾向于使用路由器而不是网桥来连接网络，因此我们将着重介绍路由器。文献[Perlman 1992]的第12章对路由器和网桥进行了比较。

1.3 TCP/IP的分层

在TCP/IP协议族中，有很多种协议。图1-4给出了本书将要讨论的其他协议。

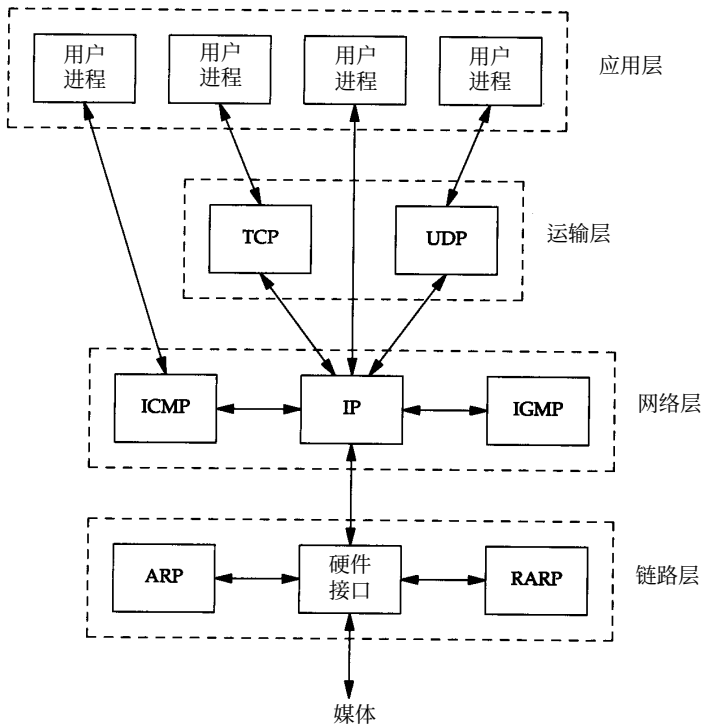

图1-4 TCP/IP协议族中不同层次的协议

TCP和UDP是两种最为著名的运输层协议，二者都使用IP作为网络层协议。

虽然TCP使用不可靠的IP服务，但它却提供一种可靠的运输层服务。本书第17～22章将详细讨论TCP的内部操作细节。然后，我们将介绍一些TCP的应用，如第26章中的Telnet和Rlogin、第27章中的FTP以及第28章中的SMTP等。这些应用通常都是用户进程。

UDP为应用程序发送和接收数据报。一个数据报是指从发送方传输到接收方的一个信息单元（例如，发送方指定的一定字节数的信息）。但是与TCP不同的是，UDP是不可靠的，它不能保证数据报能安全无误地到达最终目的。本书第11章将讨论UDP，然后在第14章（DNS: 域名系统），第15章（TFTP：简单文件传送协议），以及第16章（BOOTP：引导程序协议）介绍使用UDP的应用程序。SNMP也使用了UDP协议，但是由于它还要处理许多其他的协议，因此本书把它留到第25章再进行讨论。

IP是网络层上的主要协议，同时被TCP和UDP使用。TCP和UDP的每组数据都通过端系统和每个中间路由器中的IP层在互联网中进行传输。在图1-4中，我们给出了一个直接访问IP的应用程序。这是很少见的，但也是可能的（一些较老的选路协议就是以这种方式来实现的。当然新的运输层协议也有可能使用这种方式）。第3章主要讨论IP协议，但是为了使内容更加有针对性，一些细节将留在后面的章节中进行讨论。第9章和第10章讨论IP如何进行选路。

ICMP是IP协议的附属协议。IP层用它来与其他主机或路由器交换错误报文和其他重要信息。第6章对ICMP的有关细节进行讨论。尽管ICMP主要被IP使用，但应用程序也有可能访问它。我们将分析两个流行的诊断工具，Ping和Traceroute（第7章和第8章），它们都使用了ICMP。

IGMP是Internet组管理协议。它用来把一个UDP数据报多播到多个主机。我们在第12章中描述广播（把一个UDP数据报发送到某个指定网络上的所有主机）和多播的一般特性，然后在第13章中对IGMP协议本身进行描述。

ARP（地址解析协议）和RARP（逆地址解析协议）是某些网络接口（如以太网和令牌环网）使用的特殊协议，用来转换IP层和网络接口层使用的地址。我们分别在第4章和第5章对这两种协议进行分析和介绍。

1.4 互联网的地址

互联网上的每个接口必须有一个唯一的Internet地址（也称作IP地址）。IP地址长32 bit。Internet地址并不采用平面形式的地址空间，如1、2、3等。IP地址具有一定的结构，五类不同的互联网地址格式如图1-5所示。

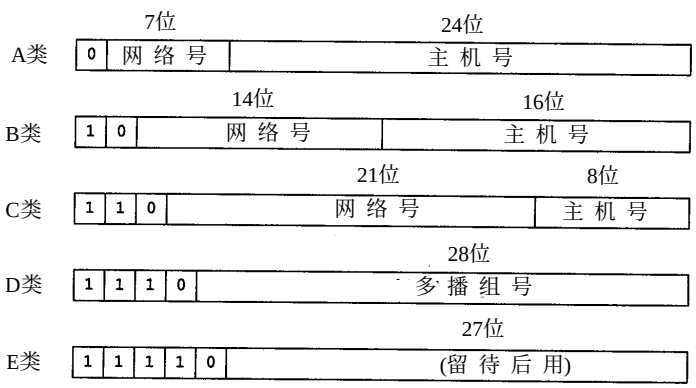

图1-5 五类互联网地址

这些32位的地址通常写成四个十进制的数，其中每个整数对应一个字节。这种表示方法称作“点分十进制表示法（Dotted decimal notation）”。例如，作者的系统就是一个B类地址，它表示为：140.252.13.33。

区分各类地址的最简单方法是看它的第一个十进制整数。图1-6列出了各类地址的起止范围，其中第一个十进制整数用加黑字体表示。

类型	范 围
A	**0**.0.0.0 到 **127**.255.255.255
B	**128**.0.0.0 到 **191**.255.255.255
C	**192**.0.0.0 到 **223**.255.255.255
D	**224**.0.0.0 到 **239**.255.255.255
E	**240**.0.0.0 到 **247**.255.255.255

图1-6 各类IP地址的范围

需要再次指出的是，多接口主机具有多个IP地址，其中每个接口都对应一个IP地址。

由于互联网上的每个接口必须有一个唯一的IP地址，因此必须要有一个管理机构为接入互联网的网络分配IP地址。这个管理机构就是互联网络信息中心（Internet Network Information Centre），称作InterNIC。InterNIC只分配网络号。主机号的分配由系统管理员来负责。

> Internet注册服务(IP地址和DNS域名)过去由NIC来负责，其网络地址是nic.ddn.mil。1993年4月1日，InterNIC成立。现在，NIC只负责处理国防数据网的注册请求，所有其他的Internet用户注册请求均由InterNIC负责处理，其网址是：rs.internic.net。
>
> 事实上InterNIC由三部分组成：注册服务（rs.internic.net），目录和数据库服务（ds.internic.net），以及信息服务（is.internic.net）。有关InterNIC的其他信息参见习题1.8。

有三类IP地址：单播地址（目的为单个主机）、广播地址（目的端为给定网络上的所有主机）以及多播地址（目的端为同一组内的所有主机）。第12章和第13章将分别讨论广播和多播的更多细节。

在3.4节中，我们在介绍IP选路以后将进一步介绍子网的概念。图3-9给出了几个特殊的IP地址：主机号和网络号为全0或全1。

1.5 域名系统

尽管通过IP地址可以识别主机上的网络接口，进而访问主机，但是人们最喜欢使用的还是主机名。在TCP/IP领域中，域名系统（DNS）是一个分布的数据库，由它来提供IP地址和主机名之间的映射信息。我们在第14章将详细讨论DNS。

现在，我们必须理解，任何应用程序都可以调用一个标准的库函数来查看给定名字的主机的IP地址。类似地，系统还提供一个逆函数——给定主机的IP地址，查看它所对应的主机名。

大多数使用主机名作为参数的应用程序也可以把IP地址作为参数。例如，在第4章中当我们用Telnet进行远程登录时，既可以指定一个主机名，也可以指定一个IP地址。

1.6 封装

当应用程序用TCP传送数据时，数据被送入协议栈中，然后逐个通过每一层直到被当作一串比特流送入网络。其中每一层对收到的数据都要增加一些首部信息（有时还要增加尾部信息），该过程如图1-7所示。TCP传给IP的数据单元称作TCP报文段或简称为TCP段（TCP segment）。IP传给网络接口层的数据单元称作IP数据报(IP datagram)。通过以太网传输的比特流称作帧(Frame)。

图1-7中帧头和帧尾下面所标注的数字是典型以太网帧首部的字节长度。在后面的章节中我们将详细讨论这些帧头的具体含义。

以太网数据帧的物理特性是其长度必须在 46～1500字节之间。我们将在4.5节遇到最小长度的数据帧，在2.8节中遇到最大长度的数据帧。

所有的Internet标准和大多数有关TCP/IP的书都使用octet这个术语来表示字节。使用这个过分雕琢的术语是有历史原因的，因为TCP/IP的很多工作都是在DEC-10系统上进行的，但是它并不使用8 bit的字节。由于现在几乎所有的计算机系统都采用8 bit的字节，因此我们在本书中使用字节（byte）这个术语。

更准确地说，图1-7中IP和网络接口层之间传送的数据单元应该是分组（packet）。分组既可以是一个IP数据报，也可以是IP数据报的一个片（fragment）。我们将在11.5节讨论IP数据报分片的详细情况。

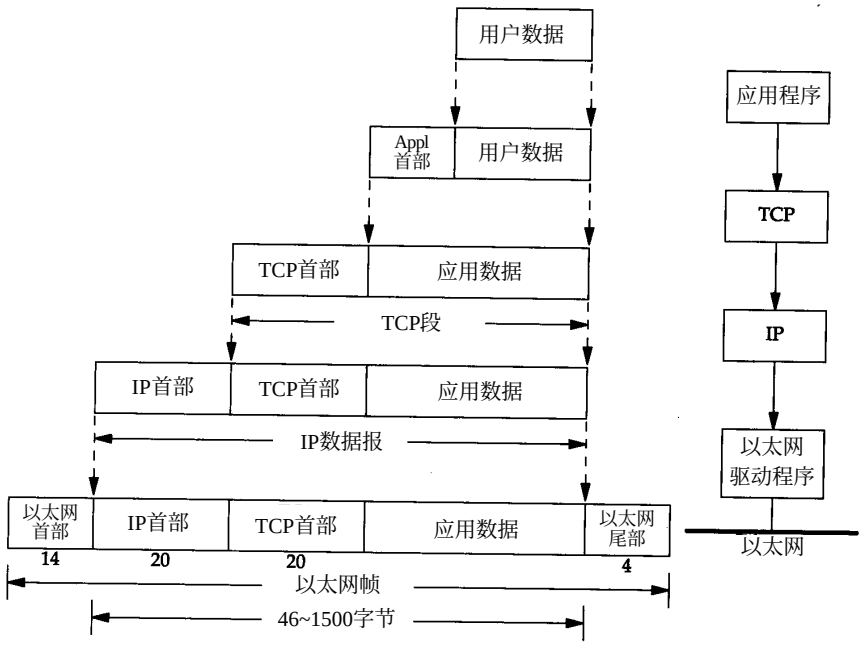

图1-7 数据进入协议栈时的封装过程

UDP数据与TCP数据基本一致。唯一的不同是 UDP传给IP的信息单元称作 UDP数据报（UDP datagram），而且UDP的首部长为8字节。

回想1.3节中的图1-4，由于TCP、UDP、ICMP和IGMP都要向IP传送数据，因此IP必须在生成的IP首部中加入某种标识，以表明数据属于哪一层。为此， IP在首部中存入一个长度为8bit的数值，称作协议域。 1表示为ICMP协议， 2表示为IGMP协议， 6表示为TCP协议， 17表示为UDP协议。

类似地，许多应用程序都可以使用 TCP或UDP来传送数据。运输层协议在生成报文首部时要存入一个应用程序的标识符。 TCP和UDP都用一个16bit的端口号来表示不同的应用程序。TCP和UDP把源端口号和目的端口号分别存入报文首部中。

网络接口分别要发送和接收IP、ARP和RARP数据，因此也必须在以太网的帧首部中加入

某种形式的标识，以指明生成数据的网络层协议。为此，以太网的帧首部也有一个 16 bit的帧类型域。

1.7 分用

当目的主机收到一个以太网数据帧时，数据就开始从协议栈中由底向上升，同时去掉各层协议加上的报文首部。每层协议盒都要去检查报文首部中的协议标识，以确定接收数据的上层协议。这个过程称作分用（Demultiplexing），图1-8显示了该过程是如何发生的。

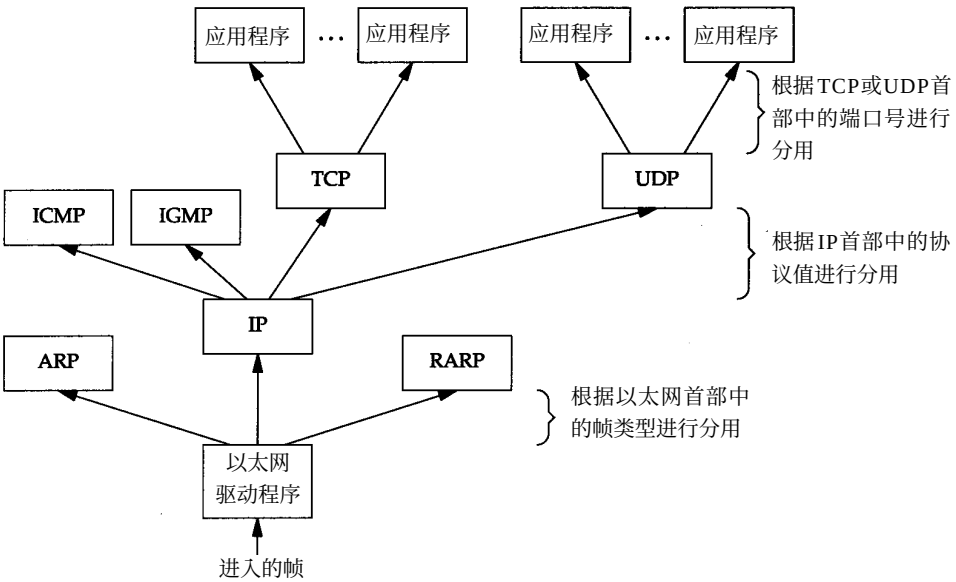

图1-8 以太网数据帧的分用过程

为协议ICMP和IGMP定位一直是一件很棘手的事情。在图1-4中，把它们与IP放在同一层上，那是因为事实上它们是IP的附属协议。但是在这里，我们又把它们放在IP层的上面，这是因为ICMP和IGMP报文都被封装在IP数据报中。

对于ARP和RARP，我们也遇到类似的难题。在这里把它们放在以太网设备驱动程序的上方，这是因为它们和IP数据报一样，都有各自的以太网数据帧类型。但在图2-4中，我们又把ARP作为以太网设备驱动程序的一部分，放在IP层的下面，其原因在逻辑上是合理的。

这些分层协议盒并不都是完美的。

当进一步描述TCP的细节时，我们将看到协议确实是通过目的端口号、源 IP地址和源端口号进行解包的。

1.8 客户-服务器模型

大部分网络应用程序在编写时都假设一端是客户，另一端是服务器，其目的是为了让服务器为客户提供一些特定的服务。

可以将这种服务分为两种类型：重复型或并发型。重复型服务器通过以下步骤进行交互：

I1. 等待一个客户请求的到来。

I2. 处理客户请求。

I3. 发送响应给发送请求的客户。

I4. 返回I1步。

重复型服务器主要的问题发生在I2状态。在这个时候，它不能为其他客户机提供服务。

相应地，并发型服务器采用以下步骤：

C1. 等待一个客户请求的到来。

C2. 启动一个新的服务器来处理这个客户的请求。在这期间可能生成一个新的进程、任务或线程，并依赖底层操作系统的支持。这个步骤如何进行取决于操作系统。生成的新服务器对客户的全部请求进行处理。处理结束后，终止这个新服务器。

C3. 返回C1步。

并发服务器的优点在于它是利用生成其他服务器的方法来处理客户的请求。也就是说，每个客户都有它自己对应的服务器。如果操作系统允许多任务，那么就可以同时为多个客户服务。

对服务器，而不是对客户进行分类的原因是因为对于一个客户来说，它通常并不能够辨别自己是与一个重复型服务器或并发型服务器进行对话。

一般来说，TCP服务器是并发的，而UDP服务器是重复的，但也存在一些例外。我们将在11.12节对UDP对其服务器产生的影响进行详细讨论，并在18.11节对TCP对其服务器的影响进行讨论。

1.9 端口号

前面已经指出过，TCP和UDP采用16 bit的端口号来识别应用程序。那么这些端口号是如何选择的呢？

服务器一般都是通过知名端口号来识别的。例如，对于每个TCP/IP实现来说，FTP服务器的TCP端口号都是21，每个Telnet服务器的TCP端口号都是23，每个TFTP(简单文件传送协议)服务器的UDP端口号都是69。任何TCP/IP实现所提供的服务都用知名的1～1023之间的端口号。这些知名端口号由Internet号分配机构（Internet Assigned Numbers Authority, IANA）来管理。

> 到1992年为止，知名端口号介于1～255之间。256～1023之间的端口号通常都是由Unix系统占用，以提供一些特定的Unix服务 —— 也就是说，提供一些只有Unix系统才有的、而其他操作系统可能不提供的服务。现在IANA管理1～1023之间所有的端口号。
>
> Internet扩展服务与Unix特定服务之间的一个差别就是Telnet和Rlogin。它们二者都允许通过计算机网络登录到其他主机上。Telnet是采用端口号为23的TCP/IP标准且几乎可以在所有操作系统上进行实现。相反，Rlogin最开始时只是为Unix系统设计的（尽管许多非Unix系统现在也提供该服务），因此在80年代初，它的有名端口号为513。

客户端通常对它所使用的端口号并不关心，只需保证该端口号在本机上是唯一的就可以了。客户端口号又称作临时端口号（即存在时间很短暂）。这是因为它通常只是在用户运行该客户程序时才存在，而服务器则只要主机开着的，其服务就运行。

大多数TCP/IP实现给临时端口分配1024～5000之间的端口号。大于5000的端口号是为其

他服务器预留的（Internet上并不常用的服务）。我们可以在后面看见许多这样的给临时端口分配端口号的例子。

Solaris 2.2是一个很有名的例外。通常TCP和UDP的缺省临时端口号从32768开始。在E.4节中，我们将详细描述系统管理员如何对配置选项进行修改以改变这些缺省项。

大多数Unix系统的文件/etc/services都包含了人们熟知的端口号。为了找到 Telnet服务器和域名系统的端口号，可以运行以下语句：

```
sun % grep telnet /etc/services
telnet     23/tcp     称它使用TCP端口号23
sun % grep domain /etc/services
domain     53/udp     称它使用UDP端口号53和TCP端口号53
domain     53/tcp
```

保留端口号

Unix系统有保留端口号的概念。只有具有超级用户特权的进程才允许给它自己分配一个保留端口号。

这些端口号介于1～1023之间，一些应用程序（如有名的 Rlogin，26.2节）将它作为客户与服务器之间身份认证的一部分。

1.10 标准化过程

究竟是谁控制着 TCP/IP协议族，又是谁在定义新的标准以及其他类似的事情？事实上，有四个小组在负责Internet技术。

1) Internet协会（ISOC，Internet Society）是一个推动、支持和促进Internet不断增长和发展的专业组织，它把Internet作为全球研究通信的基础设施。

2) Internet体系结构委员会（IAB，Internet Architecture Board）是一个技术监督和协调的机构。它由国际上来自不同专业的 15个志愿者组成，其职能是负责 Internet标准的最后编辑和技术审核。IAB隶属于ISOC。

3) Internet工程专门小组（IETF，Internet Engineering Task Force）是一个面向近期标准的组织，它分为9个领域（应用、寻径和寻址、安全等等）。IETF开发成为Internet标准的规范。为帮助IETF主席，又成立了Internet工程指导小组（IESG，Internet Engineering Steering Group）。

4) Internet研究专门小组（IRIF，Internet Research Task Force）主要对长远的项目进行研究。

IRTF和IETF都隶属于IAB。文献[Crocker 1993]提供了关于Internet内部标准化进程更为详细的信息，同时还介绍了它的早期历史。

1.11 RFC

所有关于Internet的正式标准都以RFC（Request for Comment）文档出版。另外，大量的RFC并不是正式的标准，出版的目的只是为了提供信息。 RFC的篇幅从1页到200页不等。每一项都用一个数字来标识，如 RFC 1122，数字越大说明RFC的内容越新。

所有的RFC都可以通过电子邮件或用 FTP从Internet上免费获取。如果发送下面这份电子邮件，就会收到一份获取RFC的方法清单：

```
To: rfc-info@ISI.EDU
Subject: getting rfcs
help: ways_to_get_rfcs
```

最新的RFC索引总是搜索信息的起点。这个索引列出了 RFC被替换或局部更新的时间。下面是一些重要的RFC文档：

1) 赋值RFC（Assigned Numbers RFC）列出了所有Internet协议中使用的数字和常数。至本书出版时为止，最新 RFC的编号是1340 [Reynolds和Postel 1992]。所有著名的Internet端口号都列在这里。

   当这个RFC被更新时(通常每年至少更新一次)，索引清单会列出RFC 1340被替换的时间。

2) Internet正式协议标准，目前是RFC 1600[Postel 1994]。这个RFC描述了各种Internet协议的标准化现状。每种协议都处于下面几种标准化状态之一：标准、草案标准、提议标准、实验标准、信息标准和历史标准。另外，对每种协议都有一个要求的层次、必需的、建议的、可选择的、限制使用的或者不推荐的。

   与赋值RFC一样，这个RFC也定期更新。请随时查看最新版本。

3) 主机需求RFC，1122和1123[Braden 1989a, 1989b]。RFC 1122针对链路层、网络层和运输层；RFC 1123针对应用层。这两个RFC对早期重要的RFC文档作了大量的纠正和解释。如果要查看有关协议更详细的细节内容，它们通常是一个入口点。它们列出了协议中关于“必须”、“应该”、“可以”、“不应该”或者“不能”等特性及其实现细节。

   文献[Borman 1993b]提供了有关这两个RFC的实用内容。RFC 1127[Braden 1989c]对工作小组开发主机需求RFC过程中的讨论内容和结论进行了非正式的总结。

4) 路由器需求RFC，目前正式版是RFC 1009[Braden and Postel 1987]，但一个新版已接近完成[Almquist 1993]。它与主机需求RFC类似，但是只单独描述了路由器的需求。

1.12 标准的简单服务

有一些标准的简单服务几乎每种实现都要提供。在本书中我们将使用其中的一些服务程序，而客户程序通常选择 Telnet。图1-9描述了这些服务。从该图可以看出，当使用 TCP和UDP提供相同的服务时，一般选择相同的端口号。

名 字	TCP端口号	UDP端口号	RFC	描 述
echo	7	7	862	服务器返回客户发送的所有内容
discard	9	9	863	服务器丢弃客户发送的所有内容
daytime	13	13	867	服务器以可读形式返回时间和日期
chargen	19	19	864	当客户发送一个数据报时，TCP服务器发送一串连续的字符流，直到客户中断连接。UDP服务器发送一个随机长度的数据报
time	37	37	868	服务器返回一个二进制形式的32 bit数，表示从UTC时间1900年1月1日午夜至今的秒数

图1-9 大多数实现都提供的标准的简单服务

如果仔细检查这些标准的简单服务以及其他标准的TCP/IP服务（如Telnet、FTP、SMTP等）的端口号时，我们发现它们都是奇数。这是有历史原因的，因为这些端口号都是从NCP端口号派生出来的（NCP，即网络控制协议，是ARPANET的运输层协议，是TCP的前身）。NCP是单工的，不是全双工的，因此每个应用程序需要两个连接，需预留一对奇数和偶数端口号。当TCP和UDP成为标准的运输层协议时，每个应用程序只需要一个端口号，因此就使用了NCP中的奇数。

1.13 互联网

在图1-3中，我们列举了一个由两个网络组成的互联网 —— 一个以太网和一个令牌环网。在1.4节和1.9节中，我们讨论了世界范围内的互联网 —— Internet，以及集中分配IP地址的需要（InterNIC），还讨论了知名端口号（IANA）。internet这个词第一个字母是否大写决定了它具有不同的含义。

internet意思是用一个共同的协议族把多个网络连接在一起。而 Internet指的是世界范围内通过TCP/IP互相通信的所有主机集合（超过100万台）。Internet是一个internet，但internet不等于Internet。

1.14 实现

既成事实标准的TCP/IP软件实现来自于位于伯克利的加利福尼亚大学的计算机系统研究小组。从历史上看，软件是随同4.x BSD系统（Berkeley Software Distribution）的网络版一起发布的。它的源代码是许多其他实现的基础。

图1-10列举了各种BSD版本发布的时间，并标注了重要的TCP/IP特性。列在左边的BSD网络版，其所有的网络源代码可以公开得到：包括协议本身以及许多应用程序和工具（如Telnet和FTP）。

在本书中，我们将使用“伯克利派生系统”来指SunOS 4.x、SVR4以及AIX 3.2等那些基于伯克利源代码开发的系统。这些系统有很多共同之处，经常包含相同的错误。

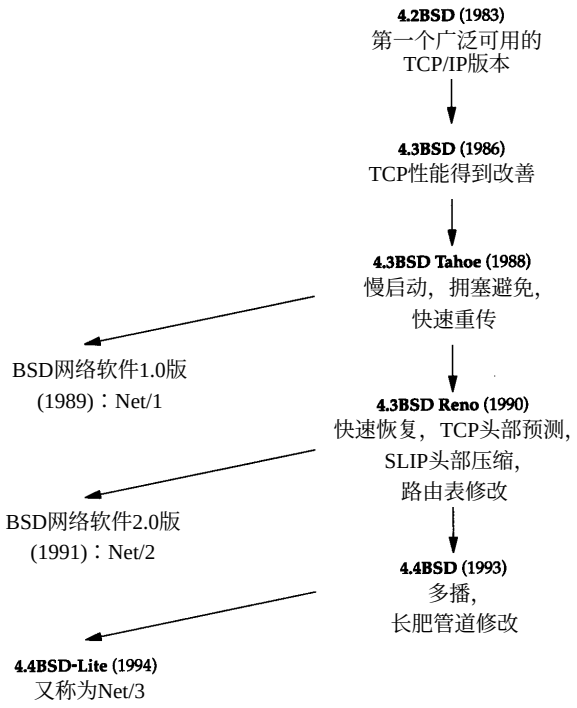

图1-10 不同的BSD版及其重要的TCP/IP特性

起初关于Internet的很多研究现在仍然在伯克利系统中应用 —— 新的拥塞控制算法（21.7节）、多播（12.4节）、“长肥管道”修改（24.3节）以及其他类似的研究。

1.15 应用编程接口

使用TCP/IP协议的应用程序通常采用两种应用编程接口（API）：socket和TLI（运输层接

口：Transport Layer Interface）。前者有时称作“ Berkeley socket ”，表明它是从伯克利版发展而来的。后者起初是由AT&T开发的，有时称作XTI（X/Open运输层接口），以承认X/Open这个自己定义标准的国际计算机生产商所做的工作。 XTI实际上是TLI的一个超集。

本书不是一本编程方面的书，但是偶尔会引用一些内容来说明 TCP/IP的特性，不管大多数的API（socket）是否提供它们。所有关于 socket和TLI的编程细节请参阅文献 [Stevens 1990]。

1.16 测试网络

图1-11是本书中所有的例子运行的测试网络。为阅读时参考方便，该图还复制在本书扉页前的插页中。

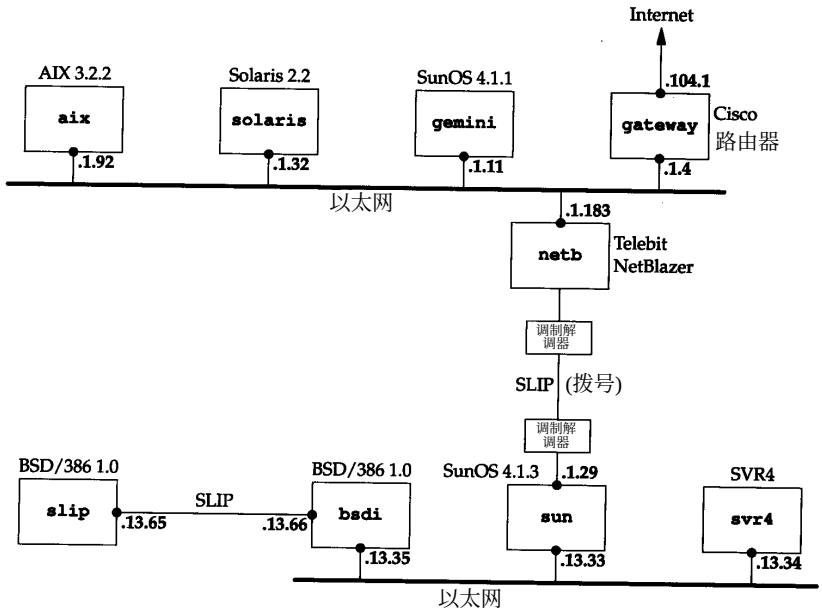

图1-11 本书中所有例子运行的测试网络，所有的IP地址均从140.252开始编址

在这个图中（作者的子网），大多数的例子都运行在下面四个系统中。图中所有的 IP地址属于B类地址，网络号为140.252。所有的主机名属于 `.tuc.noao.edu`这个域（`noao`代表National Optical Astronomy Observatories，`tuc`代表Tucson）。例如，右下方的系统有一个完整的名字：`svr4.tuc.noao.edu`，其IP地址是：140.252.13.34。每个方框上方的名称是该主机运行的操作系统。这一组系统和网络上的主机及路由器运行于不同的 TCP/IP实现。

需要指出的是，`noao.edu`这个域中的网络和主机要比图1-11中的多得多。这里列出来的只是本书中将要用到的系统。

在3.4节中，我们将描述这个网络所用到的子网形式。在 4.6节中将介绍`sun`与`netb`之间的拨号SLIP的有关细节。2.4节将详细讨论SLIP。

1.17 小结

本章快速地浏览了TCP/IP协议族，介绍了在后面的章节中将要详细讨论的许多术语和协议。

TCP/IP协议族分为四层：链路层、网络层、运输层和应用层，每一层各有不同的责任。在TCP/IP中，网络层和运输层之间的区别是最为关键的：网络层（ IP）提供点到点的服务，而运输层（TCP和UDP）提供端到端的服务。

一个互联网是网络的网络。构造互联网的共同基石是路由器，它们在 IP层把网络连在一起。第一个字母大写的Internet是指分布在世界各地的大型互联网，其中包括 1万多个网络和超过100万台主机。

在一个互联网上，每个接口都用 IP地址来标识，尽管用户习惯使用主机名而不是 IP地址。域名系统为主机名和 IP地址之间提供动态的映射。端口号用来标识互相通信的应用程序。服务器使用知名端口号，而客户使用临时设定的端口号。

习题

1.1 请计算最多有多少个A类、B类和C类网络号。

1.2 用匿名FTP（见27.3节）从主机`nic.merit.edu`上获取文件`nsfnet/statistics/history.netcount`。该文件包含在NSFNET网络上登记的国内和国外的网络数。画一坐标系，横坐标代表年，纵坐标代表网络总数的对数值。纵坐标的最大值是习题 1.1的结果。如果数据显示一个明显的趋势，请估计按照当前的编址体制推算，何时会用完所有的网络地址（3.10节讨论解决该难题的建议）。

1.3 获取一份主机需求RFC拷贝[Braden 1989a]，阅读有关应用于TCP/IP协议族每一层的稳健性原则。这个原则的参考对象是什么？

1.4 获取一份最新的赋值RFC拷贝。“ quote of the day ”协议的有名端口号是什么？哪个 RFC对该协议进行了定义？

1.5 如果你有一个接入TCP/IP互联网的主机帐号，它的主 IP地址是多少？这台主机是否接入了Internet？它是多接口主机吗？

1.6 获取一份RFC 1000的拷贝，了解RFC这个术语从何而来。

1.7 与Internet协会联系，`isoc@isoc.org`或者+1 703 648 9888，了解有关加入的情况。

1.8 用匿名FTP从主机`is.internic.net`处获取文件`about-internic/information-about-the-internic`。

第2章 链 路 层

2.1 引言

从图1-4中可以看出，在TCP/IP协议族中，链路层主要有三个目的：（1）为IP模块发送和接收IP数据报；（2）为ARP模块发送ARP请求和接收ARP应答；（3）为RARP发送RARP请求和接收RARP应答。TCP/IP支持多种不同的链路层协议，这取决于网络所使用的硬件，如以太网、令牌环网、FDDI（光纤分布式数据接口）及RS-232串行线路等。

在本章中，我们将详细讨论以太网链路层协议，两个串行接口链路层协议（SLIP和PPP），以及大多数实现都包含的环回（loopback）驱动程序。以太网和SLIP是本书中大多数例子使用的链路层。对MTU（最大传输单元）进行了介绍，这个概念在本书的后面章节中将多次遇到。我们还讨论了如何为串行线路选择MTU。

2.2 以太网和IEEE 802封装

以太网这个术语一般是指数字设备公司（Digital Equipment Corp.）、英特尔公司（Intel Corp.）和Xerox公司在1982年联合公布的一个标准。它是当今TCP/IP采用的主要的局域网技术。它采用一种称作CSMA/CD的媒体接入方法，其意思是带冲突检测的载波侦听多路接入（Carrier Sense, Multiple Access with Collision Detection）。它的速率为10 Mb/s，地址为48 bit。

几年后，IEEE（电子电气工程师协会）802委员会公布了一个稍有不同的标准集，其中802.3针对整个CSMA/CD网络，802.4针对令牌总线网络，802.5针对令牌环网络。这三者的共同特性由802.2标准来定义，那就是802网络共有的逻辑链路控制（LLC）。不幸的是，802.2和802.3定义了一个与以太网不同的帧格式。文献[Stallings 1987]对所有的IEEE 802标准进行了详细的介绍。

在TCP/IP世界中，以太网IP数据报的封装是在RFC 894[Hornig 1984]中定义的，IEEE 802网络的IP数据报封装是在RFC 1042[Postel and Reynolds 1988]中定义的。主机需求RFC要求每台Internet主机都与一个10 Mb/s的以太网电缆相连接：

1) 必须能发送和接收采用RFC 894（以太网）封装格式的分组。

2) 应该能接收与RFC 894混合的RFC 1042（IEEE 802）封装格式的分组。

3) 也许能够发送采用RFC 1042格式封装的分组。如果主机能同时发送两种类型的分组数据，那么发送的分组必须是可以设置的，而且默认条件下必须是RFC 894分组。

最常使用的封装格式是RFC 894定义的格式。图2-1显示了两种不同形式的封装格式。图中每个方框下面的数字是它们的字节长度。

两种帧格式都采用48 bit（6字节）的目的地址和源地址（802.3允许使用16 bit的地址，但一般是48 bit地址）。这就是我们在本书中所称的硬件地址。ARP和RARP协议（第4章和第5章）对32 bit的IP地址和48 bit的硬件地址进行映射。

接下来的2个字节在两种帧格式中互不相同。在802标准定义的帧格式中，长度字段是指

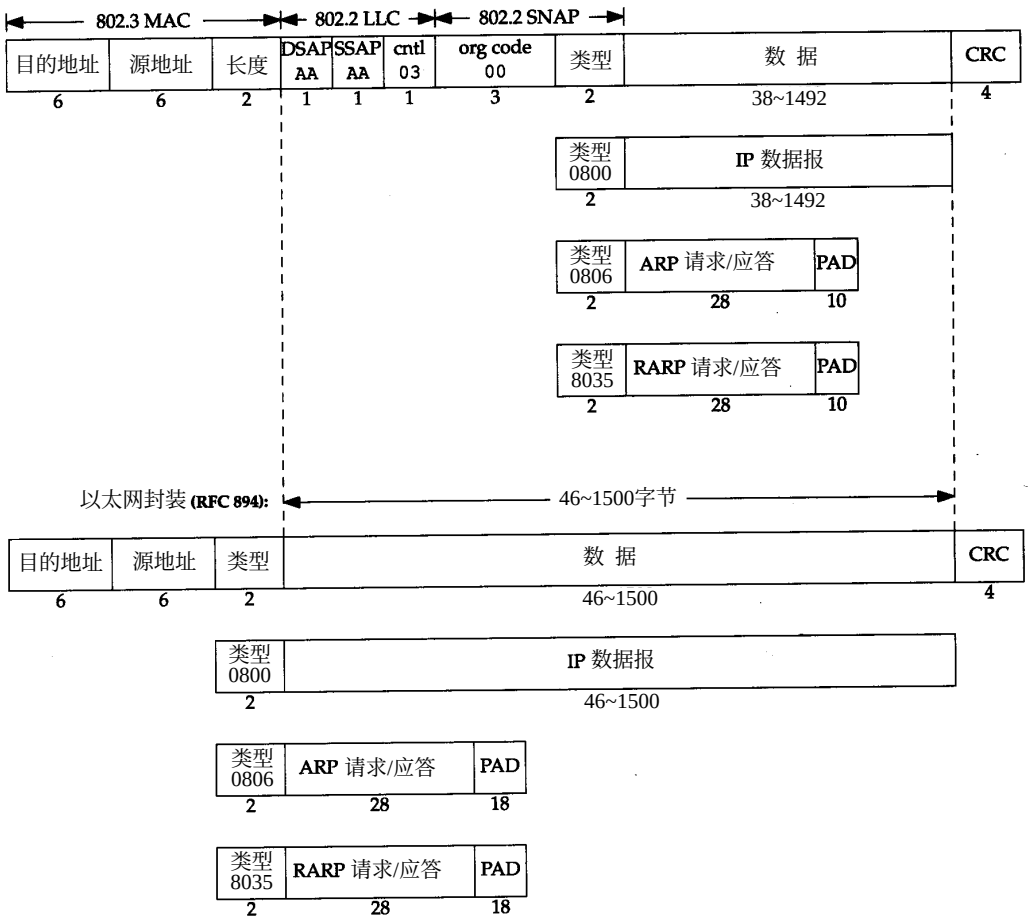

图2-1 IEEE 802.2/802.3（RFC 1042）和以太网的封装格式（RFC 894）

它后续数据的字节长度，但不包括 CRC检验码。以太网的类型字段定义了后续数据的类型。在802标准定义的帧格式中，类型字段则由后续的子网接入协议（ Sub-network Access Protocol，SNAP）的首部给出。幸运的是， 802定义的有效长度值与以太网的有效类型值无一相同，这样，就可以对两种帧格式进行区分。

在以太网帧格式中，类型字段之后就是数据；而在 802帧格式中，跟随在后面的是 3字节的802.2 LLC和5字节的802.2 SNAP。目的服务访问点（ Destination Service Access Point, DSAP）和源服务访问点（Source Service Access Point, SSAP）的值都设为0xaa。Ctrl字段的值设为3。随后的3个字节org code都置为0。再接下来的2个字节类型字段和以太网帧格式一样（其他类型字段值可以参见RFC 1340 [Reynolds and Postel 1992]）。

CRC字段用于帧内后续字节差错的循环冗余码检验（检验和）（它也被称为FCS或帧检验序列）。

802.3标准定义的帧和以太网的帧都有最小长度要求。 802.3规定数据部分必须至少为38字节，而对于以太网，则要求最少要有 46字节。为了保证这一点，必须在不足的空间插入填充（pad）字节。在开始观察线路上的分组时将遇到这种最小长度的情况。

在本书中，我们在需要的时候将给出以太网的封装格式，因为这是最为常见的封装格式。

2.3 尾部封装

RFC 893[Leffler and Karels 1984]描述了另一种用于以太网的封装格式，称作尾部封装（trailer encapsulation）。这是一个早期BSD系统在DEC VAX机上运行时的试验格式，它通过调整IP数据报中字段的次序来提高性能。在以太网数据帧中，开始的那部分是变长的字段（IP首部和TCP首部）。把它们移到尾部（在CRC之前），这样当把数据复制到内核时，就可以把数据帧中的数据部分映射到一个硬件页面，节省内存到内存的复制过程。 TCP数据报的长度是512字节的整数倍，正好可以用内核中的页表来处理。两台主机通过协商使用 ARP扩展协议对数据帧进行尾部封装。这些数据帧需定义不同的以太网帧类型值。

现在，尾部封装已遭到反对，因此我们不对它举任何例子。有兴趣的读者请参阅 RFC 893以及文献[Leffler et al. 1989]的11.8节。

2.4 SLIP：串行线路IP

SLIP的全称是Serial Line IP。它是一种在串行线路上对IP数据报进行封装的简单形式，在RFC 1055[Romkey 1988]中有详细描述。SLIP适用于家庭中每台计算机几乎都有的 RS-232串行端口和高速调制解调器接入Internet。

下面的规则描述了SLIP协议定义的帧格式：

1) IP数据报以一个称作END（0xc0）的特殊字符结束。同时，为了防止数据报到来之前的线路噪声被当成数据报内容，大多数实现在数据报的开始处也传一个 END字符（如果有线路噪声，那么END字符将结束这份错误的报文。这样当前的报文得以正确地传输，而前一个错误报文交给上层后，会发现其内容毫无意义而被丢弃）。

2) 如果IP报文中某个字符为 END，那么就要连续传输两个字节 0xdb和0xdc来取代它。0xdb这个特殊字符被称作SLIP的ESC字符，但是它的值与ASCII码的ESC字符（0x1b）不同。

3) 如果IP报文中某个字符为SLIP的ESC字符，那么就要连续传输两个字节 0xdb和0xdd来取代它。

图2-2中的例子就是含有一个END字符和一个ESC字符的IP报文。在这个例子中，在串行线路上传输的总字节数是原IP报文长度再加4个字节。

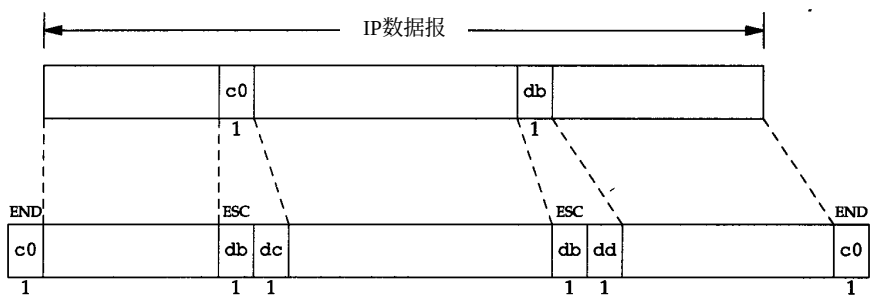

图2-2 SLIP报文的封装

SLIP是一种简单的帧封装方法，还有一些值得一提的缺陷：

1) 每一端必须知道对方的IP地址。没有办法把本端的IP地址通知给另一端。

2) 数据帧中没有类型字段（类似于以太网中的类型字段）。如果一条串行线路用于SLIP，那么它不能同时使用其他协议。

3) SLIP没有在数据帧中加上检验和（类似于以太网中的 CRC字段）。如果SLIP传输的报文被线路噪声影响而发生错误，只能通过上层协议来发现（另一种方法是，新型的调制解调器可以检测并纠正错误报文）。这样，上层协议提供某种形式的 CRC就显得很重要。在第 3章和第17章中，我们将看到IP首部和TCP首部及其数据始终都有检验和。在第 11章中，将看到UDP首部及其数据的检验和却是可选的。

尽管存在这些缺点， SLIP仍然是一种广泛使用的协议。

SLIP的历史要追溯到1984年，Rick Adams第一次在4.2BSD系统中实现。尽管它本身的描述是一种非标准的协议，但是随着调制解调器的速率和可靠性的提高， SLIP越来越流行。现在，它的许多产品可以公开获得，而且很多厂家都支持这种协议。

2.5 压缩的SLIP

由于串行线路的速率通常较低（ 19200 b/s或更低），而且通信经常是交互式的（如 Telnet和Rlogin，二者都使用TCP），因此在SLIP线路上有许多小的TCP分组进行交换。为了传送1个字节的数据需要20个字节的IP首部和20个字节的TCP首部，总数超过40个字节（19.2节描述了Rlogin会话过程中，当敲入一个简单命令时这些小报文传输的详细情况）。

既然承认这些性能上的缺陷，于是人们提出一个被称作 CSLIP（即压缩SLIP）的新协议，它在RFC 1144[Jacobson 1990a]中被详细描述。CSLIP一般能把上面的40个字节压缩到3或5个字节。它能在CSLIP的每一端维持多达16个TCP连接，并且知道其中每个连接的首部中的某些字段一般不会发生变化。对于那些发生变化的字段，大多数只是一些小的数字和的改变。这些被压缩的首部大大地缩短了交互响应时间。

现在大多数的SLIP产品都支持CSLIP。作者所在的子网（参见封面内页）中有两条SLIP链路，它们均是CSLIP链路。

2.6 PPP：点对点协议

PPP，点对点协议修改了SLIP协议中的所有缺陷。 PPP包括以下三个部分：

1) 在串行链路上封装 IP数据报的方法。 PPP既支持数据为 8位和无奇偶检验的异步模式（如大多数计算机上都普遍存在的串行接口），还支持面向比特的同步链接。

2) 建立、配置及测试数据链路的链路控制协议（ LCP：Link Control Protocol）。它允许通信双方进行协商，以确定不同的选项。

3) 针对不同网络层协议的网络控制协议（ NCP：Network Control Protocol）体系。当前RFC定义的网络层有IP、OSI网络层、DECnet以及AppleTalk。例如， IP NCP允许双方商定是否对报文首部进行压缩，类似于 CSLIP（缩写词NCP也可用在TCP的前面）。

RFC 1548[Simpson 1993]描述了报文封装的方法和链路控制协议。 RFC 1332[McGregor 1992]描述了针对IP的网络控制协议。

PPP数据帧的格式看上去很像 ISO的HDLC（高层数据链路控制）标准。图 2-3是PPP数据帧的格式。

每一帧都以标志字符0x7e开始和结束。紧接着是一个地址字节，值始终是 0xff，然后是一个值为0x03的控制字节。

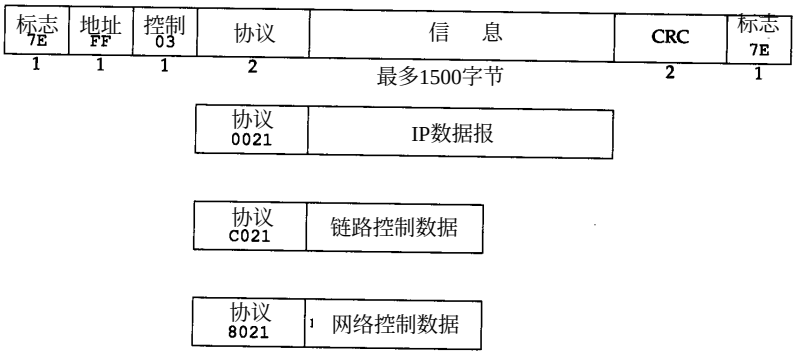

图2-3 PPP数据帧的格式

接下来是协议字段，类似于以太网中类型字段的功能。当它的值为 0x0021时，表示信息字段是一个IP数据报；值为0xc021时，表示信息字段是链路控制数据；值为 0x8021时，表示信息字段是网络控制数据。

CRC字段（或FCS，帧检验序列）是一个循环冗余检验码，以检测数据帧中的错误。

由于标志字符的值是 0x7e，因此当该字符出现在信息字段中时， PPP需要对它进行转义。在同步链路中，该过程是通过一种称作比特填充 (bit stuffing)的硬件技术来完成的[Tanenbaum 1989]。在异步链路中，特殊字符 0x7d用作转义字符。当它出现在 PPP数据帧中时，那么紧接着的字符的第6个比特要取其补码，具体实现过程如下：

1) 当遇到字符0x7e时，需连续传送两个字符：0x7d和0x5e，以实现标志字符的转义。

2) 当遇到转义字符0x7d时，需连续传送两个字符：0x7d和0x5d，以实现转义字符的转义。

3) 默认情况下，如果字符的值小于 0x20（比如，一个ASCII控制字符），一般都要进行转义。例如，遇到字符0x01时需连续传送0x7d和0x21两个字符（这时，第6个比特取补码后变为1，而前面两种情况均把它变为0）。

这样做的原因是防止它们出现在双方主机的串行接口驱动程序或调制解调器中，因为有时它们会把这些控制字符解释成特殊的含义。另一种可能是用链路控制协议来指定是否需要对这32个字符中的某一些值进行转义。默认情况下是对所有的 32个字符都进行转义。

与SLIP类似，由于PPP经常用于低速的串行链路，因此减少每一帧的字节数可以降低应用程序的交互时延。利用链路控制协议，大多数的产品通过协商可以省略标志符和地址字段，并且把协议字段由 2个字节减少到 1个字节。如果我们把 PPP的帧格式与前面的 SLIP的帧格式（图2-2）进行比较会发现， PPP只增加了 3个额外的字节： 1个字节留给协议字段，另 2个给CRC字段使用。另外，使用 IP网络控制协议，大多数的产品可以通过协商采用 Van Jacobson报文首部压缩方法（对应于CSLIP压缩），减小IP和TCP首部长度。

总的来说， PPP比SLIP具有下面这些优点：(1) PPP支持在单根串行线路上运行多种协议，不只是IP协议；(2) 每一帧都有循环冗余检验；(3) 通信双方可以进行 IP地址的动态协商(使用IP网络控制协议)；(4) 与CSLIP类似，对TCP和IP报文首部进行压缩；(5) 链路控制协议可以对多个数据链路选项进行设置。为这些优点付出的代价是在每一帧的首部增加 3个字节，当建立链路时要发送几帧协商数据，以及更为复杂的实现。

尽管PPP比SLIP有更多的优点，但是现在的SLIP用户仍然比PPP用户多。随着产品越来越多，产家也开始逐渐支持PPP，因此最终PPP应该取代SLIP。

2.7 环回接口

大多数的产品都支持环回接口（Loopback Interface），以允许运行在同一台主机上的客户程序和服务器程序通过TCP/IP进行通信。A类网络号127就是为环回接口预留的。根据惯例，大多数系统把IP地址127.0.0.1分配给这个接口，并命名为localhost。一个传给环回接口的IP数据报不能在任何网络上出现。

我们想象，一旦传输层检测到目的端地址是环回地址时，应该可以省略部分传输层和所有网络层的逻辑操作。但是大多数的产品还是照样完成传输层和网络层的所有过程，只是当IP数据报离开网络层时把它返回给自己。

图2-4是环回接口处理IP数据报的简单过程。

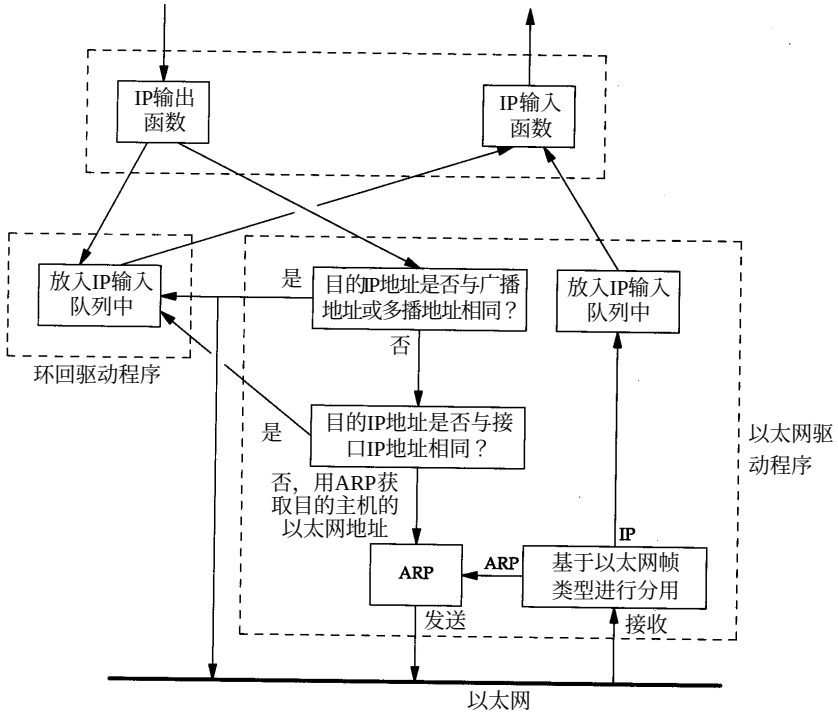

图2-4 环回接口处理IP数据报的过程

图中需要指出的关键点是：

1) 传给环回地址（一般是127.0.0.1）的任何数据均作为IP输入。

2) 传给广播地址或多播地址的数据报复制一份传给环回接口，然后送到以太网上。这是因为广播传送和多播传送的定义（第12章）包含主机本身。

3) 任何传给该主机IP地址的数据均送到环回接口。

看上去用传输层和IP层的方法来处理环回数据似乎效率不高，但它简化了设计，因为环回接口可以被看作是网络层下面的另一个链路层。网络层把一份数据报传送给环回接口，就像传给其他链路层一样，只不过环回接口把它返回到IP的输入队列中。

在图2-4中，另一个隐含的意思是送给主机本身IP地址的IP数据报一般不出现在相应的网络上。例如，在一个以太网上，分组一般不被传出去然后读回来。某些BSD以太网的设备驱动程序的注释说明，许多以太网接口卡不能读回它们自己发送出去的数据。由于一台主机必

须处理发送给自己的IP数据报，因此图2-4所示的过程是最为简单的处理办法。

4.4BSD系统定义了变量useloopback，并初始化为1。但是，如果这个变量置为0，以太网驱动程序就会把本地分组送到网络，而不是送到环回接口上。它也许不能工作，这取决于所使用的以太网接口卡和设备驱动程序。

2.8 最大传输单元MTU

正如在图2-1看到的那样，以太网和802.3对数据帧的长度都有一个限制，其最大值分别是1500和1492字节。链路层的这个特性称作MTU，最大传输单元。不同类型的网络大多数都有一个上限。

如果IP层有一个数据报要传，而且数据的长度比链路层的MTU还大，那么IP层就需要进行分片（fragmentation），把数据报分成若干片，这样每一片都小于MTU。我们将在11.5节讨论IP分片的过程。

网　　络	MTU字节
超通道	65535
16 Mb/s令牌环(IBM)	17914
4 Mb/s令牌环(IEEE 802.5)	4464
FDDI	4352
以太网	1500
IEEE 802.3/802.2	1492
X.25	576
点对点(低时延)	296

图2-5 几种常见的最大传输单元（MTU）

图2-5列出了一些典型的MTU值，它们摘自RFC 1191[Mogul and Deering 1990]。点到点的链路层（如SLIP和PPP）的MTU并非指的是网络媒体的物理特性。相反，它是一个逻辑限制，目的是为交互使用提供足够快的响应时间。在2.10节中，我们将看到这个限制值是如何计算出来的。

在3.9节中，我们将用netstat命令打印出网络接口的MTU。

2.9 路径MTU

当在同一个网络上的两台主机互相进行通信时，该网络的MTU是非常重要的。但是如果两台主机之间的通信要通过多个网络，那么每个网络的链路层就可能有不同的MTU。重要的不是两台主机所在网络的MTU的值，重要的是两台通信主机路径中的最小MTU。它被称作路径MTU。

两台主机之间的路径MTU不一定是个常数。它取决于当时所选择的路由。而选路不一定是对称的（从A到B的路由可能与从B到A的路由不同），因此路径MTU在两个方向上不一定是一致的。

RFC 1191[Mogul and Deering 1990]描述了路径MTU的发现机制，即在任何时候确定路径MTU的方法。我们在介绍了ICMP和IP分片方法以后再来看它是如何操作的。在11.6节中，我们将看到ICMP的不可到达错误就采用这种发现方法。在11.7节中，还会看到，traceroute程序也是用这个方法来确定到达目的节点的路径MTU。在11.8节和24.2节，将介绍当产品支持路径MTU的发现方法时，UDP和TCP是如何进行操作的。

2.10 串行线路吞吐量计算

如果线路速率是9600 b/s，而一个字节有8 bit，加上一个起始比特和一个停止比特，那么线路的速率就是960 B/s（字节/秒）。以这个速率传输一个1024字节的分组需要1066 ms。如果

用SLIP链接运行一个交互式应用程序，同时还运行另一个应用程序如 FTP发送或接收1024字节的数据，那么一般来说就必须等待一半的时间（ 533 ms）才能把交互式应用程序的分组数据发送出去。

假定交互分组数据可以在其他“ 大块 ”分组数据发送之前被发送出去。大多数的 SLIP实现确实提供这类服务排队方法，把交互数据放在大块的数据前面。交互通信一般有 Telnet、Rlogin以及FTP的控制部分（用户的命令，而不是数据）。

> 这种服务排队方法是不完善的。它不能影响已经进入下游（如串行驱动程序）队列的非交互数据。同时，新型的调制解调器具有很大的缓冲区，因此非交互数据可能已经进入该缓冲区了。

对于交互应用来说，等待 533 ms是不能接受的。关于人的有关研究表明，交互响应时间超过100～200 ms就被认为是不好的[Jacobson 1990a]。这是发送一份交互报文出去后，直到接收到响应信息（通常是出现一个回显字符）为止的往返时间。

把SLIP的MTU缩短到256就意味着链路传输一帧最长需要 266 ms，它的一半是133 ms（这是一般需要等待的时间）。这样情况会好一些，但仍然不完美。我们选择它的原因（与 64或128相比）是因为大块数据提供良好的线路利用率（如大文件传输）。假设CSLIP的报文首部是5个字节，数据帧总长为261个字节，256个字节的数据使线路的利用率为98.1%，帧头占了1.9%，这样的利用率是很不错的。如果把 MTU降到256以下，那么将降低传输大块数据的最大吞吐量。

在图2-5列出的MTU值中，点对点链路的MTU是296个字节。假设数据为256字节，TCP和IP首部占40个字节。由于MTU是IP向链路层查询的结果，因此该值必须包括通常的 TCP和IP首部。这样就会导致IP如何进行分片的决策。IP对于CSLIP的压缩情况一无所知。

我们对平均等待时间的计算（传输最大数据帧所需时间的一半）只适用于 SLIP链路（或PPP链路）在交互通信和大块数据传输这两种情况下。当只有交互通信时，如果线路速率是9600 b/s，那么任何方向上的1字节数据（假设有5个字节的压缩帧头）往返一次都大约需要12.5 ms。它比前面提到的100~200 ms要小得多。需要注意的是，由于帧头从40个字节压缩到5个字节，使得1字节数据往返时间从85 ms减到12.5 ms。

不幸的是，当使用新型的纠错和压缩调制解调器时，这样的计算就更难了。这些调制解调器所采用的压缩方法使得在线路上传输的字节数大大减少，但纠错机制又会增加传输的时间。不过，这些计算是我们进行合理决策的入口点。

在后面的章节中，我们将用这些串行线路吞吐量的计算来验证数据从串行线路上通过的时间。

2.11 小结

本章讨论了Internet协议族中的最底层协议，链路层协议。我们比较了以太网和 IEEE 802.2/802.3的封装格式，以及SLIP和PPP的封装格式。由于SLIP和PPP经常用于低速的链路，二者都提供了压缩不常变化的公共字段的方法。这使交互性能得到提高。

大多数的实现都提供环回接口。访问这个接口可以通过特殊的环回地址，一般为127.0.0.1。也可以通过发送IP数据报给主机所拥有的任一IP地址。当环回数据回到上层的协议栈中时，它已经过传输层和IP层完整的处理过程。

我们描述了很多链路都具有的一个重要特性， MTU，相关的一个概念是路径MTU。根据典型的串行线路MTU，对SLIP和CSLIP链路的传输时延进行了计算。

本章的内容只覆盖了当今TCP/IP所采用的部分数据链路公共技术。 TCP/IP成功的原因之一是它几乎能在任何数据链路技术上运行。

习题

2.1 如果你的系统支持`netstat`(1)命令（参见3.9节），那么请用它确定系统上的接口及其MTU。