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

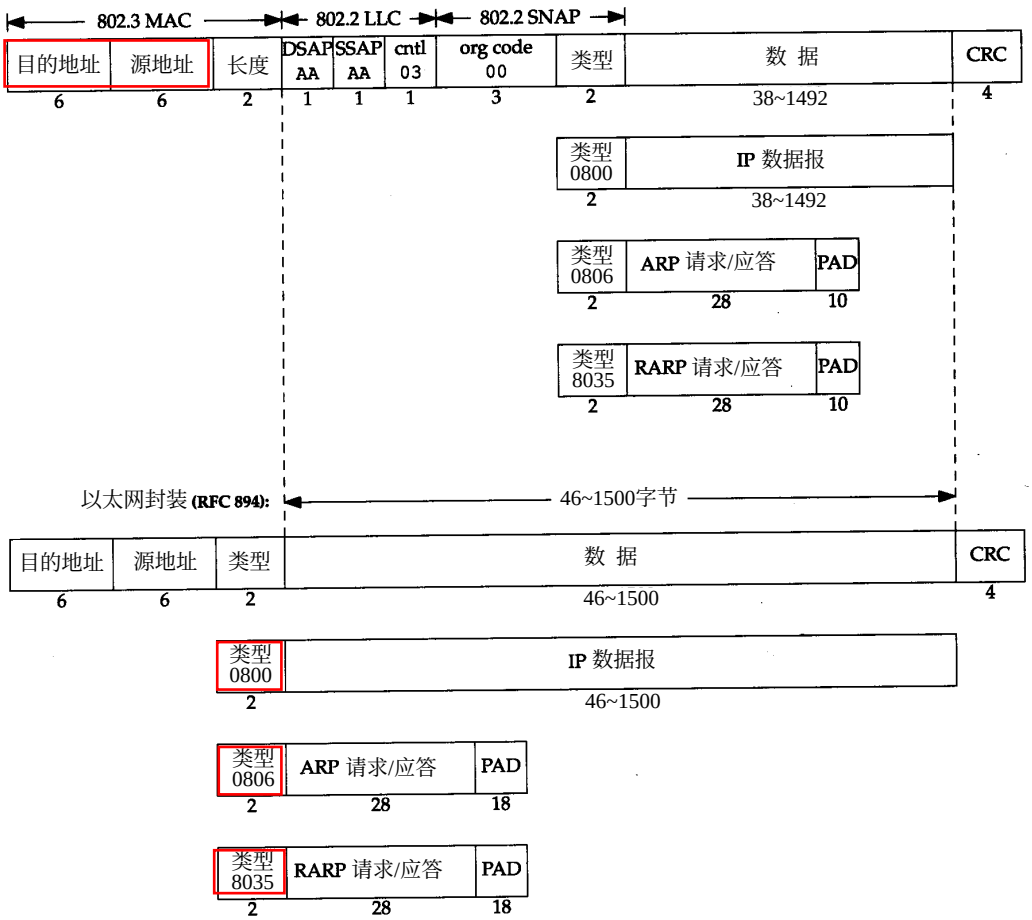

图2-1 IEEE 802.2/802.3（RFC 1042）和以太网的封装格式（RFC 894）

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。

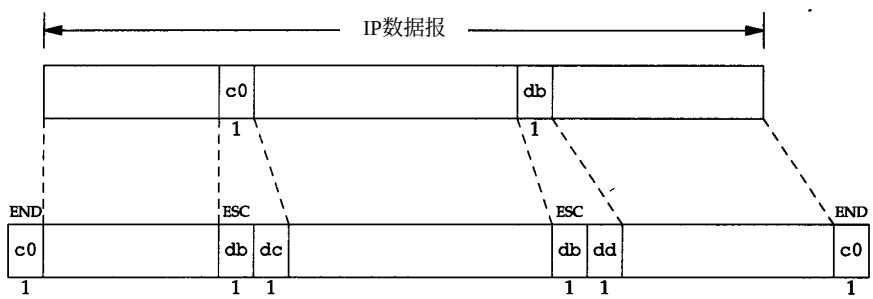

图2-2 SLIP报文的封装

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。

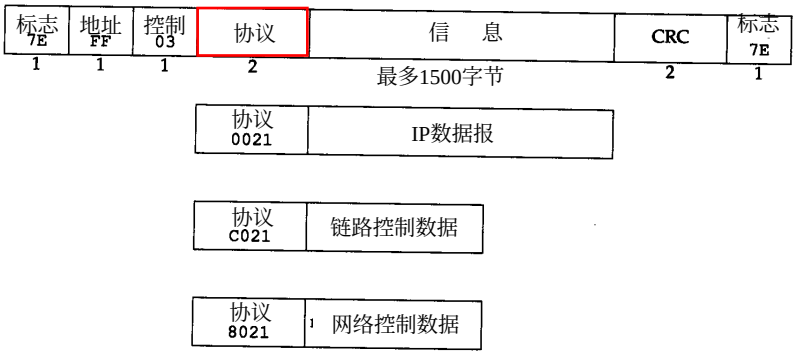

图2-3 PPP数据帧的格式

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。

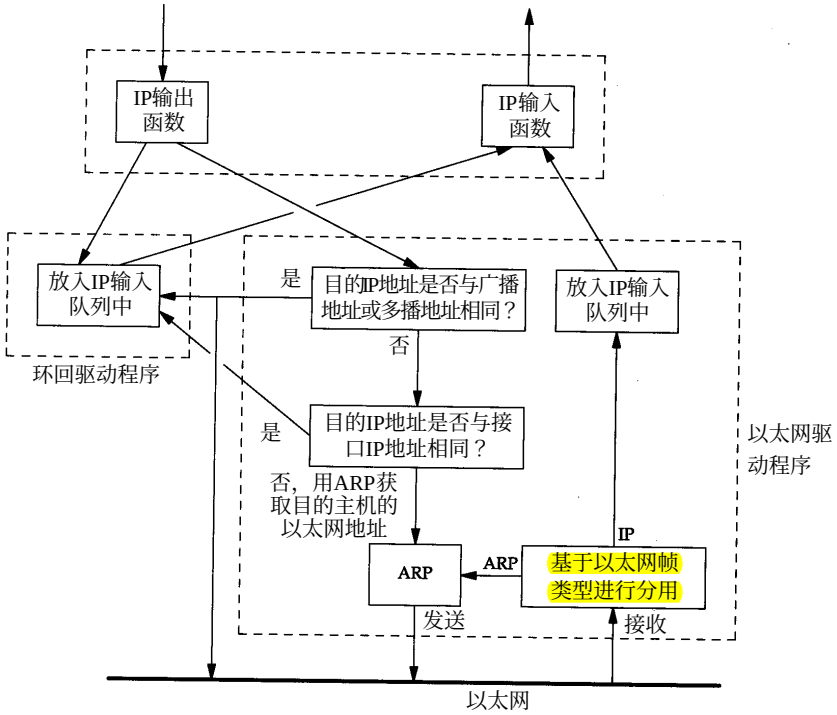

图2-4 环回接口处理IP数据报的过程

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。