第一章　概念介绍

本书向熟悉计算机系统的系统工程师、系统分析员、系统程序员提供有关数据通信及网络的知识，从而对计算机系统中增加的数据通信及网络的理解提供了一个很好的起点。

1.1　数据通信和网络

在数据通信和网络之间很难定义一个确切的界限。数据通信倾向于描述低级数据传送，通常是通过点对点的线路从一台计算机将数据传送到另一台计算机上。相反，计算机网络描述有关交换系统和使系统能互相协作的高级事宜。

也许可以认为数据通信就是从一个计算机环境将数据位或数据字节传输到另一个计算机环境，与使用软盘传输数据相比，性能大大提高。但是，计算机网络提供了互连及互相协调工作的机制，以实现网络上各种资源的共享。这些资源包括计算机、外围设备以及数据库或文件形式的数据，其最终目标是提供企业范围内的互相协调性及访问所需要的资源的能力。

在网络上以各种各样的方式提供了有效的资源。以电子邮件或广告板形式出现的电子信息，提供了交流思想的方便性，远程事务提供了数据库的及时修改，并保证分布数据的一致性。数据采集系统能实时地整理数据，以说明当前公司或企业的状况。远程文件访问则允许用户存储或使用数据信息，而不必顾及其在网络上的位置。

许多网络是包括网络上的进间程通信能力(IPC)，这些 IPC 方便地被用在完成文件传输、远程注册及电子邮件传送。另外，远程通信还可以进一步地实现对等的远程过程调用，这样的远程调用除网络传输的延时外，其形式及函数与本地的过程调用完全类似。除此之外，在一些网络上，还可以提供资源的透明访问，例如，不管是访问本地数据文件还是访问远程数据文件完全一样(透明的概念是指用户不需要知道网络的存在性，以及明确地处理网络的事务)。

数据通信和网络应用的基础是一组硬件及软件机制，一般分为两部分：

1. 硬件/软件设备，它们是网络上的组成成份。
2. 运行于 MAINFRAME 主机上或其它终端系统上的软件，这些终端系统通常被称为主机。

网络上的硬件/软件设备包括终端、多路复用器、终端访问结点、主机、前端处理器，路由器、桥以及其它的各种部件。我们先来描述终端设备。

终端可以是字符模式或页模式的。字符模式的终端一次发送一个字符，接收方的计算机为了便于处理，将这些字符组织成行的形式，而对于其它系统则是一个字符一个字符地处理。另一种模式的终端是页模式终端，在这种情况下，终端上所做的编辑只有在用户按下“传输”键，说明要做发送的情况下，才将数据发送到计算机。

许多终端都将其输出口与多路复用器(multiplexer)相连,以共享昂贵的点对点通信信道。此时在计算机一方也要安装一个同样的设备,用以将收到的信息进行分类,并把不同终端的信息分别发送到计算机的输入端口。输出信息的处理方法与输入信息的处理方法相类似。

许多终端可以通过终端集中器(terminal concentrator)实现与点对点等价的网络共享。集中器有时也称为分组装配/拆卸(Packet Assembler/Disassembler),简称为 PAD,其原因是它把一组单一的字符组装成为一个称之为信息包(Packet)为单位的方式进行传输,在接收方又将信息包拆卸为一组字符。

由 PAD 形成的消息包首先要送往称之为服务器的计算机,我们一般称之为主机(host),主机可以是从 PC 机到大型机的任何计算机,其不同点在于它们对网络上的用户提供不同的服务。

一些主机,特别是大型机,是通过称之为前端处理器(front-end processor)与网络接口的。前端处理器可以从主机上下卸数据通信和网络处理工作。

还有许多不同的网络互连设备,包括路由器,实现任意网络的互连,桥实现类似的局域网络(LANs)的互连,中断器实现共享的 LAN 技术段间的互连。所有这些互连设备都要解决下面的两个问题:

1. 如何互连网络。
2. 如何设置一堵"屏蔽墙",将问题分离,从而阻止将问题扩散到互连系统的另一部分。

所有这些设备都将在后面的章节中加以详细讨论。

在考虑到典型网络系统不同的厂商的时候,我们必须考虑提供相互可操作标准的影响。标准是非常重要的,如果没有灯泡的标准,没有胶卷的标准等等,没有日常生活中所需要的标准,可以想象将是如何一团糟。在数据通信和网络中需要,并且正在开发一些标准,以解决相互操作的问题。但是这些标准与目前灯泡、胶卷等方面存在的标准相去甚远,好在不断地制定并完善这些方面的标准。一些重要的标准包括:美国信息交换标准编码(ASCII),高级数据链路控制协议(HDLC),X.25 网络服务及接口规范,计算机交换协议和局域网协议。这些协议都适合于主要计算机厂商为了解决相互可操作性而提出的开放系统互连(OSI)的概念,关于这些协议将在后面的章节中给出详细的讨论。

在实现网络数据通信的过程中,除了考虑相互可操作性外,还有其它需要考虑的东西。由于电话公司提供的通信线路还比较昂贵,所以许多决策在增加计算能力和更有效地使用通信线路的权衡中,以更有效地使用通信线路为最终结果。但是在局域网络的情况下是例外的,在局域网中,通信非常便宜,特别是在 PC 的情况下,经常实现简单的计算机资源的共享。通信与处理之间的权衡过程与过去程序员权衡计算机内存和处理的方式非常类似。本书在讲述数据通信及网络的过程中,以计算机和计算机程序设计作为知识基础,将会有许多类似的权衡过程。

1.2 数据通信和网络开发的基础

读者可能在计算机系统开发中具有实时性要求的，很多输入/输出并且强调嵌入系统的经验，如果是这样，在已有知识的基础上，你将会发现数据通信和网络与它们有很多相似之处。我们将建立在这些现有知识基础上，因为这种类同为我们学习数据通信和网络提供了相当有用的基础。本书中用到的基础知识如下：

中断句柄(interrupt handler)对于数据通信程序是非常重要的，数据包的到来通常是由中断来处理的，中断使得中断服务程序建立新的数据缓存区，并且将新到来的数据存入数据缓存队列中，以便由可调度的任务或进程进行处理。如果传输速度高的话，在中断服务程序中是没有时间建立缓存区的，所以硬件设备必须有第二个或者更多的缓存区，用于处理一个接一个到来的数据包。

缓存区(buffering)在数据通信中同样是一个非常重要的概念，当新数据到来时，一般将其放入缓存区。缓存可以以环形的方式组织，也可以组织成一组固定尺寸的块。如果组织成一组固定尺寸的块，那么块的尺寸一般就为数据包中的最大尺寸，这样可以避免在处理一个数据包的过程中做缓存区的切换。

终端句柄(terminal handler)同样包括缓存区，但是其处理速度较慢。另外的终端处理事务包括是否做字符的回应，即回送到终端上去。此时有两种模式：

1. 局部回应或半双工(local echo 或 half-duplex)。此时当输入字符时，它们被打印在屏幕上。
2. 远程回应(remote echo)。此时，只有输入的字符先送回到计算机，然后由计算机返回到终端时，字符才被回应到屏幕上。远程回应的目的是让终端上的用户可以确信输入的字符在传输过程中没有出错。

在命令和编辑模式中，经常使用回应，但是在口令的输入过程中，要禁止回应，所以使得口令不能在屏幕上显示出来。

终端处理的另外一个主要任务是处理控制码。由于终端处理器有两个任务：处理用户的数据和处理用户的命令，所以要有一套方法来区别数据和命令。一般来说，先定义换码字符(escape character)，在换码字符之后的字符将作为控制字符。通常换码字符是一个用户在其数据中不会使用的字符，如"^]"。

除终端输入的控制外，还有计算机产生的控制，这种控制一般为控制块的形式，即包含控制信息而不是用户数据的缓存区。

进程间通信(interprocess communication)。英文缩写为 IPC，进程间的通信是计算机操作系统的扩充。网络中的 IPC 一般包括连接(长项合作)或远程过程调用(短时的合作)，无论哪一种情况，进程间的通信都是通过传输信息块，而不是内存共享的方式完成的。

计时(timer)是数据通信和网络的重要的概念，在网络中，有许多计时情况(如重传包)和许多不同计时的实现方法过程。例如，可以在每个信息包后设定一个重复传输的时间片，也可以使同一连接中的所有的包共享一个时间片。

对于数据通信方面的硬件支持，包括板子级和芯片级的产品。板子级的产品通常执行一个或多个协议，进而向系统传输封装好的数据包，当然，它们也发送数据包，而且在协议包含

的情况下，还要实现重复传输。这类硬件的软件接口通常是以I/O驱动程序的方式出现的。

芯片级的产品向用户提供数据通信开发的接口，这些芯片从支持最基本的数据通信到支持完整的协议，它们通常作为微处理器的协处理器操作，以完成数据通信的功能。

上述的描述是希望在读者已有知识的基础上，做一些数据通信和网络方面的扩充。下面我们将以传统的引入术语和概念的方式，介绍数据通信和网络的基础。

第二章　数据通信和网络的基本概念

在这一章中，我们介绍一些数据通信及网络方面的重要概念，并用例子和图示说明它们与计算机系统和系统程序设计之间的关系，这种说明对于在现有知识上的提高将成为基础，数据通信和网络处理与我们过去所做的工作相比，没有什么不同。然而，许多系统的出现将带来新的概念，这些概念在说明中将被指出来。

2.1　通信信道

数据通信和网络的重要概念之一是通信信道。毫无疑问，我们已经熟悉计算机的I/O通道，在计算机通道和数据通信信道之间有许多共同点，它们都被共享(很复杂)，都包括控制信息、数据传输以及建立在专门硬件上的I/O处理软件。

然而，在计算机通道和数据通信信道之间有一些重要不同点。最重要的不同点之一是在抽象概念的不同层次上信道概念的出现方式。从最低层看，信道可能是物理上的一对导线或是一段具有数据载波能力的同轴电缆。从抽象概念的较高层次看，信道可能是在两台直接连接的计算机上的软件。这个软件实现的信道可以有某些物理(硬件)信道不具有的特点，如非常低的出错率。

通道的初始化通常包括一个通道两端点之间的"握手"过程，在这个"握手"过程中，通过交换信息，使通道两边端点达成一个有关通道特性的协议，如通道上输出的最大包的尺寸，同时建立很多配置参数。这些内容将在后面的章节中讨论。

2.2　虚拟电路和数据包服务

我们通常使用虚拟这个词，虚拟电路具有物理电路上不存在的特性，这一点与虚拟内存的概念类似。在虚拟内存中，不管硬件实际内存有多少，程序员能开发使用很大地址空间的程序。虚拟电路的典型的特点是传输信息的可靠性和连续性(如：很少的错误位，不丢失信息，以正确的顺序传输信息)。虚拟电路通常具有其它的特点，如对发送方实际传输数据的速度，接受方有能力给于控制。虚拟电路中具有三个基本机制：

1. 错误控制。
2. 顺序控制。
3. 流控制。

每一个机制将在具有虚拟电路的各个不同的抽象层次上作详细介绍。

虚拟电路要求对通信双方建立连接。与每个连接相关的是一组可选择的在建立连接过程中达成的协定，这些协定一般包括非缺省参数的设置，如被交换的最大包的尺寸。

虚拟电路，即通常的连接，包括“状态信息”，如下一个包的顺序号码，这种状态信息是面向连接的通信信道区别于无连接信道的基本特征。

通过无连接通道发送数据的单位通常被称之为数据报(datagrams)。数据报提供一个“最有效的传送”服务。数据报服务称作“无连接”是因为它没有包含连接(或状态信息)服务，而且不保证数据的可靠性和连续性。相反，无连接的数据报被传输到设备力所能及的范围内，这个范围由网络来决定。那么什么时候选择无连接网络呢?

1. 传输少量的数据，不需要调整在其上建立的连接。
2. 当网络包含在通信提供的不同层次的服务中时。
3. 通过少量丢弃数据包来处理拥塞等网络问题时。

2.3 位串行传送和段内控制

典型的计算机 I/O 通道和数据通信信道之间的不同之一就是数据传送的性质的不同。计算机的 I/O 通道经常以并行的方式传送一个或多个字节，并具有单独的控制信息线。由于控制线和数据线是独立的，所以被称为超带控制(out of band control)。在另外一些计算机的 I/O 通道中，控制和数据可以共享相同的线路，而接收设备根据所接收的控制/数据信息传输序列中的相对时间来区别数据信息或控制信息，这就是带内控制(in-band control)。

电话公司提供了一个很好的带内控制的例子，而且已投入使用很多年了。当我们拨一个电话号码时，在老式的脉冲式电话系统中，总是听到卡嗒声，而在当前的电话系统中，听到音频声，这些都是带内控制，因为我们在话音信道中能听到它们。但是，带内控制不仅限于电话号码，也可以包括其它信息，如免费的“800”号码。带内控制在过去的年代里，给电话公司带来了不少的麻烦，因为窃用电话者现在已经能在适当的时间产生出这种信号，因此可以与世界上任何一个电话通话，而不需要付费。我们将会看到，带内的数据通信和网络将忍受这种潜在的严重的影响。

计算机 I/O 通道和数据通信之间的另一个不同点是，计算机的 I/O 通道是并行的数据传送，而数据通信信道是以位串行的方式进行，控制和数据共享同一个通信介质，这样的控制是带内的，所有的位看上去都是一样的，问题是怎样区别控制位和数据位，把它们区别出来的关键是要有一个方法去识别位序列的开始，然后对位组成的模式要有一个共识的解释协议，如图 2－1 中的例子，说明一个简单的在字符异步传输中一个字符的传输过程。

从稳定状态(标志级 mark level)到空间级(space level)的一个位的状态转换表明一个字符的到来，此位称为起始位(Start bit)。对于少于一个位时间的状态转换将视为噪音脉冲而丢弃。起始位之后是由七个数据位组成的字符位序列(Character bits)，然后是一个控制位，在当前的情况下，该控制位是一个错误控制位，称之为奇偶校验位(Parity bit)，该校验位后是最少一个位的标志级，它被称之为终止位(Stop bit)。终止位也可以认为是控制信息，但它的主要目的是把线路上的信号恢复到标志级状态，从而可以识别下一个起始位(Next Start)的状态转换。被传输的两个字符之间的时间是不定的，下一个字符可以在终止位之后马上到来，也可以在一个规定的时间内到来，这便是熟知的异步传输，但这仅仅是对字符而言。例

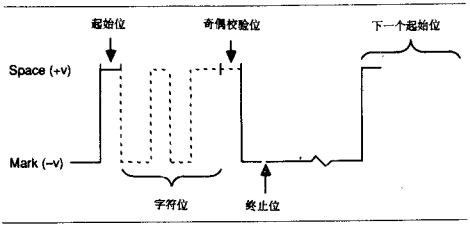

图 2—1　字符异步通信是自同步

如:字符中的位之间的时间是已知的,因为是一个位后面紧接着后面的另一位。

2.4　异步传输和同步传输

"异步"在数据通信中的应用与实时系统和其它计算机系统一样,异步事件发生是不定时的。在我们当前讨论的数据通信中,"事件"是一个字符的到来,但在另外一些情况下也指一组字符的到来,字符组可能是一个独立的信息包,也可能是一个电子邮件信息的到达或阅读一个电子邮件。

与异步通信相对应的是同步通信,"同步通信"的典型应用是在字符组成的包中,一个字符接另一个字符地传输,因此,两个字符之间的传输时间是已知的。由于字符是连续地到达,所以接收方知道何时接收字符,这一点与前面的示例中一个字符内的位的情形是一样的。和前面讲述的一样同步应用于字符,但是在这种情况下,同步也应用于位。我们把通信的这种形式也归结为"字符同步"或"位同步"。

2.5　数据流的方向

在某一时刻,通过计算机 I/O 通道的信息流通常是同一方向的,向磁盘控制器发送完控制(命令)之后,接着就返回与数据方向相反的信息流,这相当于在数据通信中的"半双工"传送,与此相对应的是在线路上同时存在有可能是相互独立的两个方向上的信息流,我们称它为"全双工"传送,或有时简称为"双工"。当然,也有可能是只有一路通道,我们称之为"单工通道",事实上,我们可以将双工通道想象为两个相反方向的单工通道。

除了半工或双工之外,通道的另一个重要特性是它的数据率。在数据通信中,数据率是以位/秒或 bps 为单位(有时,单位可能是字节/秒或 Bps),通道的数据率是由物理电路的带宽限制的,通道的带宽是电流能通过的频率范围。例如:在图 2--2 中是一个典型的电话信道

的带宽说明。

这个电话信道通过的频率(Frequency)是在 300Hz～3300Hz(Hz (hertz)赫兹,或周期/秒),那么带宽(Bandwidth)为它们的差,或 3000Hz。对带宽的每一赫兹,一个比较粗糙而有用的度量方法是我们可以在一秒内大约获得一位数据。因此,我们能在电话信道上传送 3000bps 左右的数据。当然,也有最小和最大数据率的例子,如:普通的 1200bps 传送速率或 9600bps 的传送速率,不同点一方面是使用的技术问题;另一方面是调整数据传送(Transimission),使其适合于电话带宽的设备,这个设备我们称之为调制解调器(modem)(调制器/解调器)。调制解调器和它的可编程接口将在第四章中讨论。

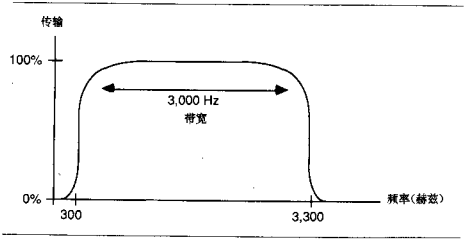

图 2—2　提供 3000Hz 带宽的电话信道

2.6　数据率、带宽和错误率

从上述电话信道上的有效数据率的例子能看到,在数据率和带宽之间存在一个关系,由任意给定的带宽能获得数据率的宽度范围。在给定带宽的信道上,最大数据率理论上可通过 Shannon 定律获得:

$$\text{最大数据率(bps)} = BW\log[1+S/N]$$

这里的 log 是以 2 为底的对数,S/N 是信号与噪音的比率。例如:如果 S/N 的值是 15,说明信号量在噪音量的 15 倍的级别上;对于这个 S/N 的值,上述方程表明最大理论数据率是带宽的 4 倍。较高的信号和噪音的比率将产生较高的数据率与带宽比。

噪音的存在有时引起以坏位形式出现的传输错误,所谓坏位即指从二进制的 1 变成 0 或相反的情形。数据通信信道的出错特性是由位错误率(BER)来度量,即出错位的比率,例如,在一个电话信道中,平均的位错误率是每传输 100,000 位,出现一个位的错误,那么 BER 为 10^{-5}。其它的介质,如 LAN 中使用的介质,通常有非常低的 BER,如同轴电缆是 10^{-9},光

纤电缆是 10^{-10}。

2.7 多路复用

到现在为止，已经讨论了信道的特性，如：数据率和错误率。另外所关心的就是怎样有效地利用信道，例如，一个信道可以用来专门连接一对通信双方，但更多的是共享一个信道。在计算机 I/O 系统中，通常有称为多路传送（或多路复用器）的信道，多路复用是共享一个通信信道的机制，它可以是几种连接形式中的任何一种形式，但最终目的是相同的，即共享一个共同的通信资源。有一些情况下，通过共享来减少通信费用，如：访问一个远程计算机。另一方面，如在 LANS 中，通过共享一个公共介质可以实现两个计算机间的通信。不管是出于何种原因，共享一个公共的通信信道被称为多路复用。

两种非常普通的多路复用的形式是频分多路复用（Frequency Division Multiplexing — FDM）和时分多路复用（Time Division Multiplexing—TDM）。它们在分隔信道资源的方法上是不同的。FDM 把频率分成独立的子信道，把每个子信道分配给每一个用户（per user），即一对通信实体。TDM 与 FDM 相反，它将时间做分隔，在给定的时间内将整个信道分配给一个用户。两者的形式如图 2—3 所示。

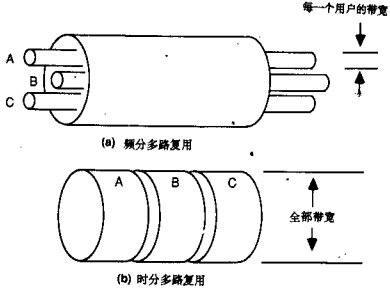

图 2—3　频分多路复用和时分多路复用以两种不同的方法共享物理信道

频分多路复用（Frequency Division Multiplexing）在计算机系统中并不多见，但在我们生活中是非常普通的，如图 2—3 所示的 FDM 子信道可以是分配给广播或有线电视的频段。在有线电视中，300～400MHz（兆赫兹）的带宽将分隔成大量的 6—MHz 的子信道，而分配给电视信道 2，3，4 等。

时分多路复用(Time Division Multiplexing)在计算机系统中是常见的，如I/O通道多路复用。首先将通道上的全部的数据负载交给一个用户，然后交给第二个，第三个等等用户，依次类推。

FDM和TDM各有其优点，但在数据通信中，TDM的例子比FDM多，其基本原因是我们使用的是数据信道，TDM更适合于数字信道，而FDM是一个更适合于模拟信道。然而，正如我们在后面将会看到的，有许多同时使用FDM和TDM的例子。例如，使用FDM把一个同轴电缆网络的整个带宽(Full Bandwidth)分隔成子信道，然后在其中的一个或多个子信道上使用TDM。

在数据通信中，多路复用的使用在终端到计算机的字符传输中相当普遍。如图2—4所示，终端(Terminal)与计算机(Computer)的连接可以是通过单独的电话线直接连接，也可以是通过共享通信链接来实现。在通过共享通信链接实现连接的情况下，为了共享公共的线路，需要使用一个多路复用器(Multiplexer)。

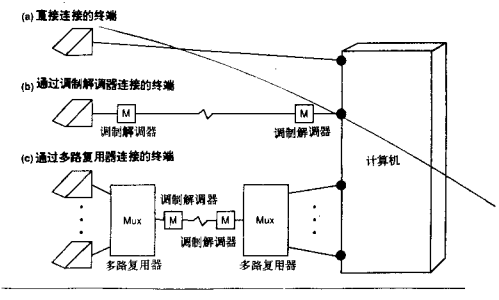

图2—4　终端与计算机可以用一些不同的方法连接作用是相同

如果使用了TDM多路复用器，那么便会将时间分割成为帧(Frame)的形式，每个终端在每一帧上将发送一个字符。如果终端上没有信息要发送，那么，一个空字符将通过分配给该用户的TDM通道分送出去。这种帧结构的效率比较低，因为有许多的时间间隔对于某些终端来说是不能接收或发送数据的，这个过程见图2—5的说明，图中还附带了一个统计时分多路复用器(Statistical TDM)的说明。这样多路复用设备通常称为统计时分多路复用器(StatMux)，在终端与主机的连接中，它是首选的多路复用设备。

图2—6为一个典型StatMux的配置，所连接设备的被组合的数据率大约是信道负载能力的4倍，当然，它是一个相当典型的例子。问题的关键不是用户在终端上的击键，而是来源于计算机的输出，它能导致通信的阻塞。如果计算机产生的通信超过了信道的容量将会发生

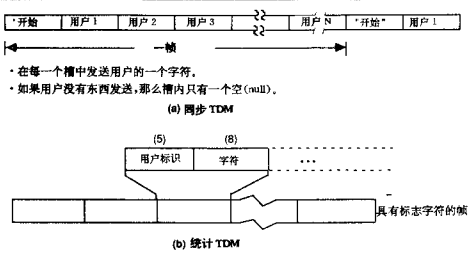

图 2—5　时分多路复用(TDM)

什么呢？通信中首先采用的机制是 StatMux 的缓冲，为了传输后面的信息，首先要将信息存贮起来，但是 StatMux 的缓冲区是有限的，所以为了防止缓冲区溢出，必须采用其它的机制。如在缓冲区占用一半时，就告诉计算机暂时停止发送，停止负载重的子信道上的信息流，但还要允许其在负载低的信道上继续流动。对于子信道上的信息流动的控制，我们采用 XOFF/XON 机制，即向主要的信息源上发送 XOFF ASCII 字符而阻止其信息的发送，当然不影响其它的信息流动，如果在其它的信息源上出现了许多的信息，那么它也会收到一个 XOFF，最后 StatMux 将会挂起，那么缓冲区的内容逐渐减少，直至到可接收数据的尺寸，然后，我们向已经发送 XOFF 而终止的子信道上发送 XON ASCII 字符而起动其数据传输。

StatMux 的工作是以通信是突发为假设前提的，即数据流是在短时间内发生，而中间的间隔时间较长，那么其它的用户便可以利用这些资源。使用统计多路复用器的代价是在每一个字符的发送中要附带一个标志。在图 2—6 的示例中，每一个 8 位的字符要附带 5 位的标志位，这样可以最多允许 32 个终端和计算机的端口进行唯一的连接。在使用帧和静态分配信道中，终端和端口在帧内字符上的位置是隐含的。

无论在 StaMux 中，还是在预分配的信道中，多路复用器的存在对主机来说是透明的。使用多路复用器后，主机的硬件和软件需要做如何的变动，这一点与每一个终端有一条单独的线路这一情况是不一样的。在数据通信和网络中，透明的概念是非常重要的，其准确的定义将依赖于所使用的上下文，但其基本的含义是不需要用户去处理的机制。在多路复用的示例中，当主机和终端之间使用多路复用器后，主机和终端都不会感觉什么不同。

2.8　分组转接

统计多路复用在字符级上对每个字符增加了控制信息(标志)，这个概念可以进一步扩

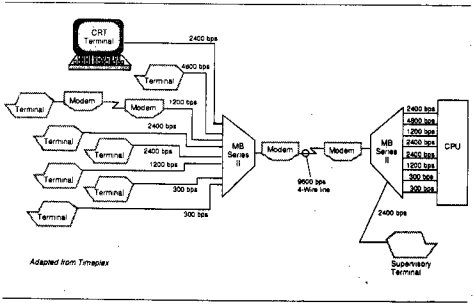

图 2－6　StatMux 示例

展，然后我们需要多于一块的控制信息，即这个特殊组中的一些字符。在更普遍的网络环境中，字符组是有顺序的，每一组字符连续顺序号。网络的效率就是把字符组变成了"数据包"，一个数据包包含几个控制字节和几个处理字节，这实质上是分组转换，而分组转换是统计多路复用对更普遍网络环境的简单扩展。

当一个 StatMux 于厂商(或所有人)协议配合操作时，在最低层到网络界面，分组转换已有了标准。然而，内部分组转换协议还是原有厂商和所有人的。这标准的网络界面是 X. 25，将在第五章中讨论。

网络协议所关心的是通过一个或多个网络数据包的信息传送，例如：在美国的一个网络用户可以访问加拿大的数据库信息，因此数据既可以通过美国的 Telenet，也可以通过加拿大的 Datapac，两者都处于网络上或网络间，但所关心的问题是一样的，该发送多大的数据包呢？如何通知发送方暂停发送呢？谁来处理错误？怎样保证数据包的正确顺序？这些问题及另外的问题将在第五章中详细讨论。

2.9　数据通信和网络性能

网络如何工作的机制将在它的协议中定义，其运行情况是由网络的协议以及协议的实现来决定的。假设某一协议有一些基本的功能，如在等待应答之前能够发送几个包，那么比较起来实现过程将更加重要，但是在给定的环境下，做适当的调整和成功地实现协议，两者都是必要的。实现不好的协议可能造成网络的低效率，并且降低网络的性能。最重要的网络性能指标是：

1. 延时
2. 容量
3. 有效性
4. 安全性

讨论如下：

延时是网络上传送一个单位信息(如：包)所需要的时间，为了度量的方便，通常采用传送一圈的时间作为度量，即一个包从发出到返回的时间。采用一圈作为度量单位而不采用单程作为度量时间，其理由是：这样可以避免发送主机和接收主机之间的时钟同步问题，因为在一圈中，发送和接收是同一个地点，回应主机可以记载发送时间和接收时间，因此能从整个延时得到主机的延时时间，这里只有时差，而没有主机的时钟同步问题。

延时是数据包尺寸、通过网络的路径、沿路径的链路数据率以及传送竞争(如队列延时)等几个方面的函数，任何网络延时的说明都将包括这些参数的值，延时说明也包括平均延时，还是95%情况下的延时，还是最坏情况下的延时。

容量是通过网络传送数据的一个不变的量。事实上，它是指传送方传送的数据位与时间的比，然而，这是一非常简单的说明量，存在一些对传送位的计数困难和超出网络控制的因素。

这些因素中，位的计数是协议的一个头痛的事，平均而言，协议头的位数同数据位的位数可能是一样的，那么容量的测量不应该包括这些协议头的位？X.25网络协议的厂商能很有说服力地说明只有X.25协议的头部位应该从容量计算中剔除，就X.25网络而言，更高层的协议位是数据位。关键是这些问题应该在开发的哪一阶段给予解决，而不是何时完成测试。

容量也可能受限于网络控制之外的因素，假设两个主机正在以"停止并等待"的协议工作，由于网络延时，发送方为了得到返回的应答(ACK)，不得不等待相对较长时间的情况下，所获得的容量是很小的，因为只有收到了返回的ACK，才能发送下一个包(这就是称为停止并等待的原因)。

有效性是指向申请者提供有效服务的时间，99%或0.99的有效性意味着在99%的时间内访问网络时，都可以提供服务，提高有效性的造价很高，因为只有很高的系统特性才能获得较高的有效性，提高有效性的方法有：很高的部件可靠性，快速的错误检测和恢复以及使用备用部件。

网络的有效性是一个复杂的事情，因为有时网络可能有一小段时间的故障，虽然一个故障还不能引起网络上所有用户服务的无效性，但是有效性计算过程中必须包括这些小的时间。

网络安全性的度量和定量都是很困难的。只有诸如偶发的包的误传可能以数据来刻画，许多安全性所关心的问题都与敌方的破坏有关，在这种情况下，如果出现对网络的成功攻击，那么几乎以100%的成功率使系统瘫痪，因此，网络安全性主要是一些所提供的机制方面的说明，并没有安全保护的数字值。

2.10 网络和数据通信中的安全性

网络和数据通信中的安全性是单计算机的安全性所关心的问题的延伸，除物理区域上的分布性外，所有的基本问题都存在，更不幸的是，这些部件通常在管理上也是分布的，这些比物理上的分布性存在更多的问题。

分布管理是主要问题的缘由可以由简单的“三腿凳”的安全模型来说明，在这里，有三个同等重要的东西形成三条“腿”，第一条腿是安全政策，在通过所有互连的部件过程中要有一条安全政策，如果是分布性的管理，这一点是很困难的；第二条腿是采用的安全机制，同样，除非在网络上有一个共同的安全措施，否则相互间的安全性操作是不一样的，分布管理便有不同的关心点，不同的政策和控制，无疑会有不同的安全机制；第三条腿是保障，如何确保各种机制被一致地执行，具有所需要的完整性，而且确实推行规定的政策？

与三腿凳一样，所有的腿是同等重要的，缺一不可，而且要足够结实以支撑人，另外，如果一条腿不强壮，其它的腿再强也不行。

安全政策必须说明网络系统的许多概念。保护什么资源？存在什么样的潜在危险？谁是有权限的用户？有权限的用户可以访问什么？在这种意义下，访问控制矩阵的目标和对象是什么？安全政策必须说明安全的定义，通常它们包含对某些数据的可信度，但也要考虑到一致性。谁能修改数据？安全性也许还会包含一些安全性的服务。

安全机制组成凳子的同样重要的第二条腿。这组安全机制形成一组保证安全政策的安全服务，这些服务通常认为 包括下列所列的各项，其中前三项主要地反映了安全政策。

- **可信性服务**：除非接收者被赋予权利去接收，否则将一直保持信息的秘密。
- **一致性服务**：能够检测到未授权者对信息的改变。
- **服务保障**：能够在拒绝服务时检测到访问，而且以降低级别的模式操作，最后恢复到完全服务。

下述的安全服务直接支持上述的政策。

- **证实服务**：证实一个人与他(她)所申明的身份是一致的，或者证实某个信息源是其说明的信息源。
- **访问控制服务**：确保只有当授权委托时，才可以访问资源，包括计算机资源的读、写和执行。
- **监听服务**：采集和分析用来指示谁访问了什么和什么时候访问的数据。
- **非否决服务**：证明一个实体实际被发送和实际被收到的能力，这些分别对应于公证和挂号邮件的功能。

安全服务和机制好象要去限定访问，这一点与开放系统互连(OSI)看起来是极端矛盾的，在 OSI 中我们希望提供完全的相互操作性和访问能力，二者所关心的问题并不矛盾，然而，他们简单地说明了限定访问边界应该被限制到为了安全的目的而故意设定的程度。如果没有这些安全边界，系统将是极端开放的，然而，如果两个厂商实现了相同的应该能够通信

的网络协议，但是每一个以不同的方式实现了安全性，或许在不同的层上，这时将会发生什么呢？安全协议的不同点呈现出一个边界，但不是所需要的边界，这个边界是完全的而不是可选择的。

安全服务和机制的标准仍处于发展之中，作为发展的一部分，关于在 OSI 哪一层上实现哪些安全服务的指南也正在确定。下面给出了大家正在逐渐接受的一些结果。

●**可信性服务**：在物理层（链路加密）和网络层或传输层之间的网络开关上提供，也可以在应用层上实现一些字段的加密。

●**一致性服务**：通过密码图的错误检测，在网络或传输层上提供。密码图有时称为密封，因为保护未授权的修改，一致性服务也可以在应用层上提供，在应用层上对相对重要信息有更多的理解。

●**服务保障**：关于应用保障层几乎没有指南，它的目的倾向于专门的应用。

●**证实服务**：这些服务可能在各种协议层上都需要，无论何时都需要准确地确认你在和谁通信。

●**访问控制服务**：访问控制服务向命令者（基于标签）和某些人（基于需要知道）提供访问目标的控制。

●**监听服务**：监听服务记载关于某些人企图访问目标的记录。这些记录包括授权的（允许的）和未授权的（不允许的）访问企图。

●**非否决服务**：应用层被认为是非否决服务普遍接受的位置。

有一些用于控制安全服务和机制的设计和实现的基本原理，在下列各项中描述它们：

●**参照监控器**：组织很好的，独立的，系统的一小部分将提供基本的访问控制，用来控制程序对目标的访问。参照监控器将需要开发三个特性，必须总是处于活动状态（对于每一次的访问）；必须是 tamperproof（以某种方式独立）；必须足够小，而且可以验证（根据情况所需要的程度）。

●**最低权限**：没有系统的用户、管理员、开发者以及维护员具有超出他（她）需要完成指定功能的权限。

●**两人控制**：在牵涉到人的对安全性特别敏感的情形，需要有两个独立的人进行操作。执行这些任务中不能相信单独的一个人。

●**多"火力墙"**：在授权的访问发生之前，要通过一个或多个独立的安全机制。这已经超出两人控制范围，必须包括多个不同的安全机制。

●**相互怀疑**：在得到肯定之前，系统的所有部分将把系统的另外部分认为是伪造的，这一点对于关键命令源更是如此，如网络控制中心和网络安全中心。但是，在某些系统中，这些应用于所有的系统部件。

在这一节中，我们试图指出一些开发安全网络的重点，这些方面涉及到安全政策的抽象内容和详细的设计、实现和安全系统的操作等各方面的概念。系统生命周期的所有阶段都必须以安全系统作为结束。

2.11 数据通信和网络的管理

任何一个提供资源的系统必须被管理，如数据通信和网络，当网络的范围扩展到包括共享服务器和访问外边的网络时，情况更是这样。正如下面的段落所讨论的那样，管理包括许多不同的方面。

网络管理涉及到下列问题的网络监控，如网络运行情况如何？瓶颈在哪里？是否要引起阻塞？哪些用户正在访问哪些资源？所有这些问题及其它成千上万的问题都通过网络监控来回答，该网络监控包括从每个网络部件上采集到的信息的接收、处理和显示，这些信息包括累计数，如发送的包的数量及重传数量，也可能包括运行统计，如所有发送和接收的包的字节的总数。

网络管理也包括控制，当网络监控发现问题时，网络控制便开始诊断问题，并且将出错的部件做分离。测试方法包括在选择的点上发送一些数据，然后让其循环返回，直到找到问题为止。

有许多与网络管理相关事情的配置，设备必须被连接到网络上或从网络中去掉；必须建立和维护说明当前配置的表格；必须设定参数值，而且必须使它们的值对网络管理员是有效；这里的一个建议是将相关机制的配置过程自动化，除非是简单的小范围内网络的情况下，不要试着去采用人工的方法维护路由表、超时值等等，在各种网络管理的概念中，我们将会看到这些自动化的例子。

网络管理员将与本书中许多章节中的内容相联系起来；其中的例子包括前面讨论的网络安全性及下节将要讨论的网络测试。

2.12 数据通信和网络测试

数据通信和网络测试出于下面的四个原因面非常重要。第一，需要知道网络的运行情况。例如：网络是不是由于不恰当的时间设定而使得每一个包被发送两次，如果是这样，那么即便数据能成功地传送，但是这将消耗的网络资源比实际应该消耗的多得多。

测试的第二个主要原因是调试网络问题，在典型的多厂商网络中，网络管理有责任去检测问题，并且要将问题确定到可替换的单元或厂商提供的服务一级上，错误的分离通常依赖于循环测试的形式，以逐渐地从测试中去除网络的某一部分或包括网络的某一部分。

第三个原因是强度测试，即在极端的情况下(如高负载)，网络能正确完成功能的最大机会，这种形式的测试通常是鸟瞰。大多数的测试集中于说明能完成正常的需求(即网络是否完成文件说明的功能?)，这些测试是必要的，但是不要忽略强度测试，发现潜在的问题比起忽略是大快之事的假设要好的多。

最后，需要做一致性测试，以实现的协议与某一个认可的、标准的实现及测试实例相一致，这种一致性测试逐渐变得更普遍起来。

在下面的章节中，分布着各种数据通信和网络的测试，但是每一种情形都是基于一个或多个上述讨论。

第三章　系统设计和开放系统下的程序设计

本章引入开放系统概念，它是各厂商产品相互操作的关键点。现在，我们已经有了国际标准OSI(开放系统互连协议)，这一协议是所有的厂商期望遵循的协议。这些协议作为开发网络产品过程中共同协议的共识，以解除诸如兼容性的后顾之忧。协议来源于OSI参照模式(RM)，该参照模式为计算机网络的开发提供一个协议性标准的框架。许多的OSI网络概念仍然处于变革阶段，同时许多部分还有待于开发。

同时，我们可以看到一些其它开放产品的使用，如TCP/IP(传输控制协议/互连协议)协议，这个协议为网络提供了基本的需求。在数据通信和网络中使用官方的OSI协议的趋势还在增长，而且OSI将在OSI协议的开发过程中以及作为其它协议比较的标准中都将继续被使用。

3.1　背景、目标和开放系统简述

开放系统与厂商专有的网络相接近，它已经出现十多年了，如上所述，TCP/IP是一组用于通信的协议，但几乎所有厂商的硬件和软件均有一定变化。虽然目前真正的国际标准还未在世界范围得以使用，但是最终将会在世界范围内得以使用，即使今天没有得到普及应用，但是我们必须对最近十余年来这种发展给予足够的重视。那么今天我们使用什么呢，我们如何才能计划将来的转变过程呢？我们一般在讨论协议时，必须将这些重要问题记在心里。

开放系统是在60年代末期随诸如ARPAnet一类的网络的发展而提出的。那时的一个问题是要将许多不同种类的计算机系统通过计算机到计算机的通信而互连，而在此之前，大部分的网络是简单的终端到主机的通信，主机之间的直接互连要求定义新的主机到主机的通信协议，并且在各种不同的机种上实现。这样就导致了第一个传输协议，即网络控制协议(NCP)。它是一个开放系统的协议，在协议中包容了许多不同种类的计算机。

与此同时，IBM公司正在开发系统网络体系结构(SNA)，而DEC公司正在开发DECnet，它们分别是IBM公司和DEC公司控制下的专有协议。但是它们未曾将这些协议作为商业秘密，即其它的公司同样可以实现这些协议，因此，少部分协议(尤其是SNA)由其它的公司实现了。这是使这些协议开放吗？一个欲表明，协议能在具有产权的同时，又能开放吗。这正如一个厂商在考虑到自己的LAN服务系统后做出的论点。“专有”意味着在一方的控制之下，但“开放”意味着其他人可以使用。因此只有在控制网络协议的厂商做适当的改变协议才可以使得开放发展下去，如增加新的功能。而其他的实现网络协议的公司现处于追赶和不利的地位。

无论专有和开放的争论结果如何，我们将把“开放”一词限制在不是一个商业企业控制的协议之内，当然，这一般意味着协议将控制在某一个委员会，这也是追求开放所付出的代价。最早的NCP协议是由组成ARPAnet主机席位的代表组成的委员会设计的，之后的TCP是由Vint Cerf和Robert Kahn设计的，但很快就被管理委员会做了修订。OSI协议也遵循了