

故障，变得无法再继续工作，说明其质量不好。高质量的网络往往价格也较高。

3. 标准化

网络的硬件和软件的设计既可以按照通用的国际标准，也可以遵循特定的专用网络标准。最好采用国际标准的设计，这样可以得到更好的互操作性，更易于升级换代和维修，也更容易得到技术上的支持。

4. 可靠性

可靠性与网络的质量和性能都有密切关系。高速网络的可靠性不一定很差。但高速网络要可靠地运行，则往往更加困难，同时所需的费用也会较高。

5. 可扩展性和可升级性

在构造网络时就应当考虑到今后可能会需要扩展（即规模扩大）和升级（即性能和版本的提高）。网络的性能越好，其扩展费用往往也越高，难度也会相应增加。

6. 易于管理和维护

网络如果没有良好的管理和维护，就很难达到和保持所设计的性能。

1.7 计算机网络体系结构

在计算机网络的基本概念中，分层次的体系结构（或架构）是最基本的。计算机网络体系结构的抽象概念较多，在学习时要多思考。这些概念对后面的学习很有帮助。

1.7.1 计算机网络体系结构的形成

计算机网络是个非常复杂的系统。为了说明这一点，可以设想一种最简单的情况：连接在网络上的两台计算机要互相传送文件。

显然，在这两台计算机之间必须有一条传送数据的通路。但这还远远不够。至少还有以下几项工作需要去完成：

- (1) 发起通信的计算机必须将数据通信的通路激活(activate)。所谓“激活”就是要发出一些信令，保证要传送的计算机数据能在这条通路上正确发送和接收。
- (2) 告诉网络如何识别接收数据的计算机。
- (3) 发起通信的计算机必须查明对方计算机是否已开机，并且与网络连接正常。
- (4) 发起通信的计算机中的应用程序必须弄清楚，在对方计算机中的文件管理程序是否已做好接收文件和存储文件的准备工作。
- (5) 若计算机的文件格式不兼容，则至少其中一台计算机应完成格式转换功能。
- (6) 对出现的各种差错和意外事故，如数据传送错误、重复或丢失，网络中某个节点交换机出现故障等，应当有可靠的措施保证对方计算机最终能够收到正确的文件。

还可以列举一些要做的其他工作。由此可见，相互通信的两个计算机系统必须高度协调工作才行，而这种“协调”是相当复杂的。为了设计这样复杂的计算机网络，早在最初的ARPANET设计时即提出了分层的方法。“分层”可将庞大而复杂的问题，转化为若干较小

的局部问题，而这些较小的局部问题就比较易于研究和处理。

1974 年，美国的 IBM 公司宣布了系统网络体系结构 SNA (System Network Architecture)。这个著名的网络标准就是按照分层的方法制定的。现在用 IBM 大型机构建的专用网络仍在使用 SNA。不久后，其他一些公司也相继推出自己公司的具有不同名称的体系结构。

不同的网络体系结构出现后，使用同一个公司生产的各种设备都能够很容易地互连成网。这种情况显然有利于一个公司垄断市场。但由于网络体系结构的不同，不同公司的设备很难互相连通。

然而，全球经济的发展使得不同网络体系结构的用户迫切要求能够互相交换信息。为了使不同体系结构的计算机网络都能互连，国际标准化组织 ISO 于 1977 年成立了专门机构研究该问题。他们提出了一个试图使各种计算机在世界范围内互连成网的标准框架，即著名的开放系统互连基本参考模型 OSI/RM (Open Systems Interconnection Reference Model)，简称为 OSI。“开放”是指非独家垄断的。因此只要遵循 OSI 标准，一个系统就可以和位于世界上任何地方的、也遵循这同一标准的其他任何系统进行通信。这一点很像世界范围的有线电话和邮政系统，这两个系统都是开放系统。“系统”是指在现实的系统中与互连有关的各部分（我们知道，并不是一个系统中的所有部分都与互连有关。OSI/RM 把与互连无关的部分除外，而仅仅考虑与互连有关的那些部分）。所以 OSI/RM 是个抽象的概念。在 1983 年形成了开放系统互连基本参考模型的正式文件，即著名的 ISO 7498 国际标准，也就是所谓的七层协议的体系结构。

OSI 试图达到一种理想境界，即全球计算机网络都遵循这个统一标准，因而全球的计算机将能够很方便地进行互连和交换数据。在 20 世纪 80 年代，许多大公司甚至一些国家的政府机构纷纷表示支持 OSI。当时看来似乎在不久的将来全世界一定会按照 OSI 制定的标准来构造自己的计算机网络。然而到了 20 世纪 90 年代初期，虽然整套的 OSI 国际标准都已经制定出来了，但由于基于 TCP/IP 的互联网已抢先在全球相当大的范围成功地运行了，而与此同时却几乎找不到有什么厂家生产出符合 OSI 标准的商用产品。因此人们得出这样的结论：OSI 只获得了一些理论研究的成果，但在市场化方面则事与愿违地失败了。现今规模最大的、覆盖全球的、基于 TCP/IP 的互联网并未使用 OSI 标准。OSI 失败的原因可归纳为：

- (1) OSI 的专家们缺乏实际经验，他们在完成 OSI 标准时缺乏商业驱动力；
- (2) OSI 的协议实现起来过分复杂，而且运行效率很低；
- (3) OSI 标准的制定周期太长，因而使得按 OSI 标准生产的设备无法及时进入市场；
- (4) OSI 的层次划分不太合理，有些功能在多个层次中重复出现。

按照一般的概念，网络技术和设备只有符合有关的国际标准才能大范围地获得工程上的应用。但现在情况却反过来了。得到最广泛运用的不是法律上的国际标准 OSI，而是非国际标准 TCP/IP。这样，TCP/IP 就常被称为是事实上的国际标准。从这种意义上说，能够占领市场的就是标准。在过去制定标准的组织中往往以专家、学者为主。但现在许多公司都纷纷加入各种标准化组织，使得技术标准具有浓厚的商业气息。一个新标准的出现，有时不一定反映其技术水平是最先进的，而是往往有着一定的市场背景。

顺便说一下，虽然 OSI 标准在一开始由 ISO 来制定，但后来的许多标准都是 ISO 与原

来的国际电报电话咨询委员会 CCITT^①联合制定的。从历史上来看，CCITT 原来是从通信的角度考虑一些标准的制定的，而 ISO 则关心信息的处理。但随着科学技术的发展，通信与信息处理的界限变得比较模糊了。于是，通信与信息处理就都成为 CCITT 与 ISO 所共同关心的领域。CCITT 的建议书 X.200 就是关于开放系统互连基本参考模型的，它和上面提到的 ISO 7498 基本上是相同的。

1.7.2 协议与划分层次

在计算机网络中要做到有条不紊地交换数据，就必须遵守一些事先约定好的规则。这些规则明确规定了所交换的数据的格式以及有关的同步问题。这里所说的同步不是狭义的（即同频或同频同相）而是广义的，即在一定的条件下应当发生什么事件（例如，应当发送一个应答信息），因而同步含有时序的意思。这些为进行网络中的数据交换而建立的规则、标准或约定称为网络协议(network protocol)。网络协议也可简称为协议。更进一步讲，网络协议主要由以下三个要素组成：

- (1) 语法，即数据与控制信息的结构或格式；
- (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响应；
- (3) 同步，即事件实现顺序的详细说明。

由此可见，网络协议是计算机网络不可缺少的组成部分。实际上，只要我们想让连接在网络上的另一台计算机做点什么事情（例如，从网络上的某台主机下载文件），都需要有协议。但是当我们经常在自己的个人电脑上进行文件存盘操作时，就不需要任何网络协议，除非这个用来存储文件的磁盘是网络上的某个文件服务器的磁盘。

协议通常有两种不同的形式。一种是使用便于人来阅读和理解的文字描述。另一种是使用让计算机能够理解的程序代码。这两种不同形式的协议都必须能够对网络上的信息交换过程做出精确的解释。

ARPANET 的研制经验表明，对于非常复杂的计算机网络协议，其结构应该是层次式的。我们可以举一个简单的例子来说明划分层次的概念。

现在假定我们在主机 1 和主机 2 之间通过一个通信网络传送文件。这是一项比较复杂的工作，因为需要做不少的工作。

我们可以将要做的工作划分为三类。第一类工作与传送文件直接有关。例如，发送端的文件传送应用程序应当确信接收端的文件管理程序已做好接收和存储文件的准备。若两台主机所用的文件格式不一样，则至少其中的一台主机应完成文件格式的转换。这两项工作可用一个文件传送模块来完成。这样，两台主机可将文件传送模块作为最高的一层（如图 1-15 所示）。在这两个模块之间的虚线表示两台主机系统交换文件和一些有关文件交换的命令。

扫一扫



视频讲解

^① 注：鉴于“有电线”和“无线电”的关系日益密切，国际电信联盟 ITU (International Telecommunication Union)已将国际电报电话咨询委员会 CCITT 和国际无线电咨询委员会 CCIR 合并为电信标准化部门 TSS (Telecommunication Standardization Sector)。从 1993 年 3 月 1 日起，CCITT 和 CCIR 不复存在，有关电信的标准就由国际电信联盟 ITU 的电信标准化部门颁布，并在每个建议书的前面加上 ITU-T 这几个字，例如，原来的 CCITT X.25 现在就称为 ITU-T X.25。为了节约经费，以后不再是每隔四年就出版全套的建议书，而是只出版新通过的建议书或旧建议书中有变化的部分。CCITT 虽然不存在了，但过去 CCITT 所制定的标准并未作废，凡未过时的标准我们在需要时都可继续引用。

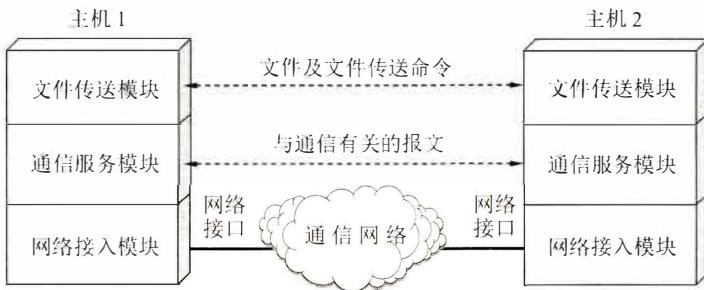


图 1-15 划分层次举例

但是，我们并不想让文件传送模块完成全部工作的细节，这样会使文件传送模块过于复杂。可以再设立一个通信服务模块，用来保证文件和文件传送命令可靠地在两个系统之间交换。这就是我们要做的第二类工作。也就是说，让位于上面的文件传送模块利用下面的通信服务模块所提供的服务。我们还可以看出，如果将位于上面的文件传送模块换成电子邮件模块，那么电子邮件模块同样可以利用在它下面的通信服务模块所提供的可靠通信的服务。

我们要做的第三类工作可以是再构造一个网络接入模块，让这个模块负责做与网络接口细节有关的工作，并向上层提供服务，使上面的通信服务模块能够完成可靠通信的任务。

从上述的简单例子可以更好地理解分层能带来很多好处，如：

(1) 各层之间是独立的。某一层并不需要知道它的下一层是如何实现的，而仅仅需要知道该层通过层间的接口（即界面）所提供的服务。由于每一层只实现一种相对独立的功能，因而可将一个难以处理的复杂问题分解为若干个较容易处理的更小一些的问题。这样，整个问题的复杂程度就下降了。

(2) 灵活性好。当任何一层发生变化时（例如由于技术的变化），只要层间接口关系保持不变，那么在这层以上或以下各层均不受影响。此外，对某一层提供的服务还可进行修改。当不再需要某层提供的服务时，甚至可以将这层取消。

(3) 结构上可分割开。各层都可以采用最合适的技术来实现。

(4) 易于实现和维护。这种结构使得实现和调试一个庞大而又复杂的系统变得更加容易，因为整个系统已被分解为若干个相对独立的子系统。

(5) 能促进标准化工作。因为每一层的功能及其所提供的服务都已有了精确的说明。

分层时应注意使每一层的功能非常明确。若层数太少，就会使每一层的协议太复杂；但层数太多又会在描述和综合各层功能的系统工程任务时遇到较多的困难。通常各层所要完成的功能主要有以下一些（可以只包括一种，也可以包括多种）：

- ① 差错控制 使相应层次对等方的通信更加可靠。
- ② 流量控制 发送端的发送速率必须使接收端来得及接收，不要太快。
- ③ 分段和重装 发送端将要发送的数据块划分为更小的单位，在接收端将其还原。
- ④ 复用和分用 发送端几个高层会话复用一条低层的连接，在接收端再进行分用。
- ⑤ 连接建立和释放 交换数据前先建立一条逻辑连接，数据传送结束后释放连接。

分层当然也有一些缺点，例如，有些功能会在不同的层次中重复出现，因而产生额外开销。

计算机网络的各层及其协议的集合就是网络的体系结构(architecture)。换种说法，计算机网络的体系结构就是这个计算机网络及其构件所应完成的功能的精确定义[GREE82]。需

要强调的是：这些功能究竟是用何种硬件或软件完成的，则是一个遵循这种体系结构的实现(implementation)的问题。体系结构的英文名词 architecture 的原意是建筑学或建筑的设计和风格。它和一个具体的建筑物的概念很不相同。例如，我们可以走进一个明代的建筑物中，但却不能走进一个明代的建筑风格之中。同理，我们也不能把一个具体的计算机网络说成是一个抽象的网络体系结构。总之，体系结构是抽象的，而实现则是具体的，是真正在运行的计算机硬件和软件。

扫一扫



视频讲解

1.7.3 具有五层协议的体系结构

OSI 的七层协议体系结构（如图 1-16(a)所示）的概念清楚，理论也较完整，但它既复杂又不实用。TCP/IP 体系结构则不同，它现在得到了非常广泛的应用。TCP/IP 是一个四层的体系结构（如图 1-16(b)所示），它包含应用层、运输层、网际层和链路层（网络接口层）。用网际层这个名字是强调本层解决不同网络的互连问题。在互联网的标准文档[RFC 1122, STD3]中，体系结构中的底层叫作链路层，但接着又说明了链路层就是媒体接入层。但也有把链路层称为网络接口层的[COME06]或子网层的[PETE12]。从实质上讲，TCP/IP 只有最上面的三层，因为最下面的链路层并没有属于 TCP/IP 体系的具体协议。链路层所使用的各种局域网标准，并非由 IETF 而是由 IEEE 的 802 委员会下属的各工作组负责制定的。在讲授计算机网络原理时往往采取另外的办法，即综合 OSI 和 TCP/IP 的优点，采用如图 1-16(c)所示的五层协议的体系结构，这对阐述计算机网络的原理是十分方便的。

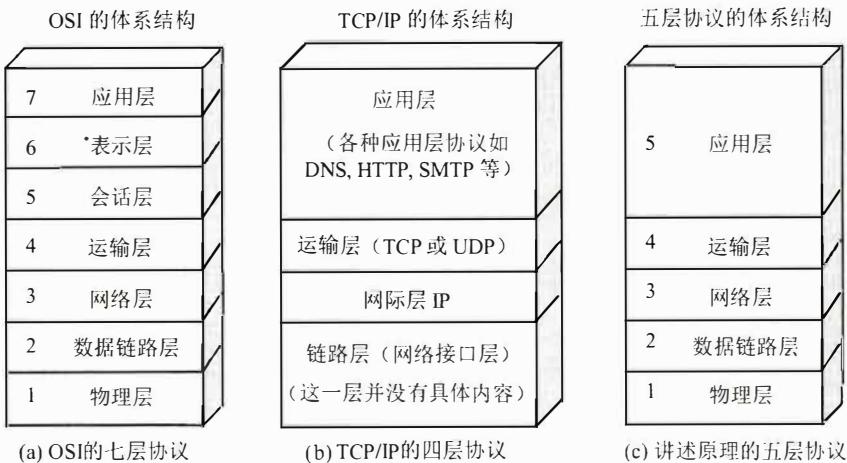


图 1-16 计算机网络体系结构

现在结合互联网的情况，自上而下地、非常简要地介绍一下各层的主要功能。实际上，只有认真学习完本书各章的协议后才能真正弄清各层的作用。

(1) 应用层(application layer)

应用层是体系结构中的最高层。应用层的任务是通过应用进程间的交互来完成特定网络应用。应用层协议定义的是应用进程间通信和交互的规则。这里的进程就是指主机中正在运行的程序。对于不同的网络应用需要有不同的应用层协议。互联网中的应用层协议很多，如域名系统 DNS、支持万维网应用的 HTTP 协议、支持电子邮件的 SMTP 协议，等等。我们把应用层交互的数据单元称为报文(message)。

(2) 运输层(transport layer)

运输层的任务就是负责向两台主机中进程之间的通信提供通用的数据传输服务。应用进程利用该服务传送应用层报文。所谓“通用的”，是指并不针对某个特定网络应用，而是多种应用可以使用同一个运输层服务。由于一台主机可同时运行多个进程，因此运输层有复用和分用的功能。复用就是多个应用层进程可同时使用下面运输层的服务，分用和复用相反，是运输层把收到的信息分别交付上面应用层中的相应进程。

运输层主要使用以下两种协议：

- 传输控制协议 TCP (Transmission Control Protocol)——提供面向连接的、可靠的数据传输服务，其数据传输的单位是报文段(segment)。
- 用户数据报协议 UDP (User Datagram Protocol)——提供无连接的尽最大努力 (best-effort)的数据传输服务（不保证数据传输的可靠性），其数据传输的单位是用户数据报。

顺便指出，有人愿意把运输层称为传输层，理由是这一层使用的 TCP 协议就叫作传输控制协议。从意思上看，传输和运输差别也不大。但 OSI 定义的第 4 层使用的是 Transport，而不是 Transmission。这两个词的含义还是有些差别的。因此，使用运输层这个译名较为准确。

(3) 网络层(network layer)

网络层负责为分组交换网上的不同主机提供通信服务。在发送数据时，网络层把运输层产生的报文段或用户数据报封装成分组或包进行传送。在 TCP/IP 体系中，由于网络层使用 IP 协议，因此分组也叫作 IP 数据报，或简称为数据报。本书把“分组”和“数据报”作为同义词使用。

请注意：不要将运输层的“用户数据报协议 UDP”和网络层的“IP 数据报”弄混。此外，无论在哪一层传送的数据单元，都可笼统地用“分组”来表示。

网络层的具体任务有两个。第一个任务是通过一定的算法，在互联网中的每一个路由器上生成一个用来转发分组的转发表。第二个任务较为简单，就是每一个路由器在接收到一个分组时，依据转发表中指明的路径把分组转发到下一个路由器。这样就可以使源主机运输层所传下来的分组，能够通过合适的路由最终到达目的主机。

这里要强调指出，网络层中的“网络”二字，已不是我们通常谈到的具体网络，而是在计算机网络体系结构模型中的第 3 层的名称。

互联网是由大量的异构(heterogeneous)网络通过路由器(router)相互连接起来的。互联网使用的网络层协议是无连接的网际协议 IP (Internet Protocol)和许多种路由选择协议，因此互联网的网络层也叫作网际层或 IP 层。在本书中，网络层、网际层和 IP 层都是同义语。

(4) 数据链路层(data link layer)

数据链路层常简称为链路层。我们知道，两台主机之间的数据传输，总是在一段一段的链路上传送的，这就需要使用专门的链路层的协议。在两个相邻节点之间传送数据时，数据链路层将网络层交下来的 IP 数据报组装成帧 (framing)，在两个相邻节点间的链路上传送帧 (frame)。每一帧包括数据和必要的控制信息（如同步信息、地址信息、差错控制等）。

在接收数据时，控制信息使接收端能够知道一个帧从哪个比特开始和到哪个比特结束。这样，数据链路层在收到一个帧后，就可从中提取出数据部分，上交给网络层。

控制信息还使接收端能够检测到所收到的帧中有无差错。如发现有差错，数据链路层就简单地丢弃这个出了差错的帧，以免继续在网络中传送下去白白浪费网络资源。如果需要

改正数据在数据链路层传输时出现的差错（这就是说，数据链路层不仅要检错，而且要纠错），那么就要采用可靠传输协议来纠正出现的差错。这种方法会使数据链路层的协议复杂些。

(5) 物理层(physical layer)

在物理层上所传数据的单位是比特。发送方发送 1（或 0）时，接收方应当收到 1（或 0）而不是 0（或 1）。因此物理层要考虑用多大的电压代表“1”或“0”，以及接收方如何识别出发送方所发送的比特。物理层还要确定连接电缆的插头应当有多少根引脚以及各引脚应如何连接。当然，解释比特代表的意思，不是物理层的任务。请注意，传递信息所利用的一些物理传输媒体，如双绞线、同轴电缆、光缆、无线信道等，并不在物理层协议之内，而是在物理层协议的下面。因此也有人把物理层下面的物理传输媒体当作第 0 层。

在互联网所使用的各种协议中，最重要的和最著名的就是 TCP 和 IP 两个协议。现在人们经常提到的 TCP/IP 并不一定是单指 TCP 和 IP 这两个具体的协议，而往往是表示互联网所使用的整个 **TCP/IP 协议族(protocol suite)**^①。

图 1-17 说明的是应用进程的数据在各层之间的传递过程中所经历的变化。这里为简单起见，假定两台主机通过一台路由器连接起来。

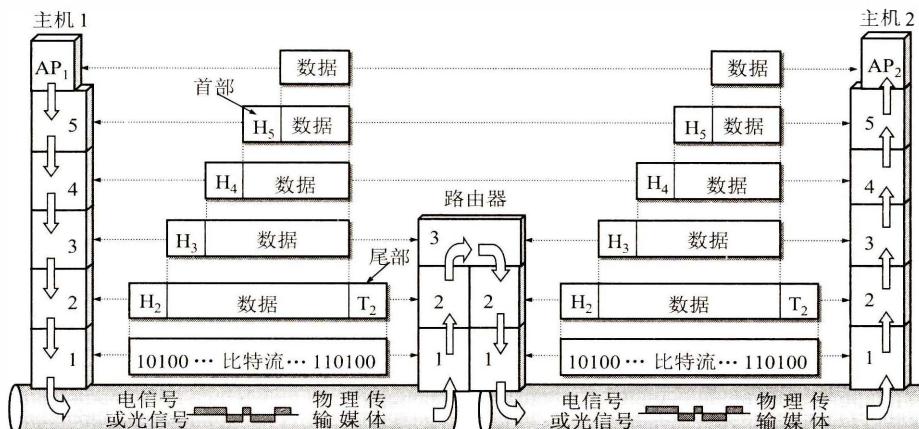


图 1-17 数据在各层之间的传递过程

假定主机 1 的应用进程 AP₁ 向主机 2 的应用进程 AP₂ 传送数据。AP₁ 先将其数据交给本主机的第 5 层（应用层）。第 5 层加上必要的控制信息 H₅ 就变成了下一层的数据单元。第 4 层（运输层）收到这个数据单元后，加上本层的控制信息 H₄，再交给第 3 层（网络层），成为第 3 层的数据单元。依此类推。不过到了第 2 层（数据链路层）后，控制信息被分成两部分，分别加到本层数据单元的首部（H₂）和尾部（T₂）；而第 1 层（物理层）由于是比特流的传送，所以不再加上控制信息。请注意，传送比特流时应从首部开始传送。

OSI 参考模型把对等层次之间传送的数据单位称为该层的协议数据单元 PDU (Protocol Data Unit)。这个名词现已被许多非 OSI 标准采用。

当这一串的比特流离开主机 1 经网络的物理传输媒体传送到路由器时，就从路由器的第 1 层依次上升到第 3 层。每一层都根据控制信息进行必要的操作，然后将控制信息剥去，

① 注：请注意 suite 这个词的特殊读音/swi:t/，不要读错。

将该层剩下的数据单元上交给更高的一层。当分组上升到了第 3 层网络层时，就根据首部中的目的地址查找路由器中的转发表，找出转发分组的接口，然后往下传送到第 2 层，加上新的首部和尾部后，再到最下面的第 1 层，然后在物理传输媒体上把每一个比特发送出去。

当这一串的比特流离开路由器到达目的站主机 2 时，就从主机 2 的第 1 层按照上面讲过的方式，依次上升到第 5 层。最后，把应用进程 AP₁ 发送的数据交给目的站的应用进程 AP₂。

可以用一个简单例子来比喻上述过程。有一封信从最高层向下传。每经过一层就包上一个新的信封，写上必要的地址信息。包有多个信封的信件传送到目的站后，从第 1 层起，每层拆开一个信封后就把信封中的信交给它的上一层。传到最高层后，取出发信人所发的信交给收信人。

虽然应用进程数据要经过如图 1-17 所示的复杂过程才能送到终点的应用进程，但这些复杂过程对用户屏蔽掉了，以致应用进程 AP₁ 觉得好像是直接把数据交给了应用进程 AP₂。同理，任何两个同样的层次（例如在两个系统的第 4 层）之间，也好像如同图 1-17 中的水平虚线所示的那样，把数据（即数据单元加上控制信息）通过水平虚线直接传递给对方。这就是所谓的“对等层”(peer layers)之间的通信。我们以前经常提到的各层协议，实际上就是在各个对等层之间传递数据时的各项规定。

在文献中也还可以见到术语“协议栈”(protocol stack)。这是因为几个层次画在一起很像一个栈(stack)的结构。

1.7.4 实体、协议、服务和服务访问点

当研究开放系统中的信息交换时，往往使用实体(entity)这一较为抽象的名词表示任何可发送或接收信息的硬件或软件进程。在许多情况下，实体就是一个特定的软件模块。

协议是控制两个对等实体（或多个实体）进行通信的规则的集合。协议的语法方面的规则定义了所交换的信息的格式，而协议的语义方面的规则就定义了发送者或接收者所要完成的操作，例如，在何种条件下，数据必须重传或丢弃。

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。

一定要弄清楚，协议和服务在概念上是很不一样的。

首先，协议的实现保证了能够向上一层提供服务。使用本层服务的实体只能看见服务而无法看见下面的协议。也就是说，下面的协议对上面的实体是透明的。

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。但服务是“垂直的”，即服务是由下层向上层通过层间接口提供的。另外，并非在一个层内完成的全部功能都称为服务。只有那些能够被高一层实体“看得见”的功能才能称之为“服务”。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，这些命令在 OSI 中称为服务原语。

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 SAP (Service Access Point)。服务访问点 SAP 是一个抽象的概念，它实际上就是一个逻辑接口，有点像邮政信箱（可以把邮件放入信箱和从信箱中取走邮件），但这种层间接口和两个设备之间的硬件接口（并行的或串行的）并不一样。OSI 把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)，它可以与 PDU 不一样。例如，可以是多个 SDU 合成为一个 PDU，也可以是一个 SDU 划分为几个 PDU。

这样，在任何相邻两层之间的关系均可概括为图 1-18 所示的那样。这里要注意的是，第 n 层的两个“实体(n)”之间通过“协议(n)”进行通信，而第 $n + 1$ 层的两个“实体($n + 1$)”之间则通过另外的“协议($n + 1$)”进行通信（每一层都使用不同的协议）。第 n 层向上面的第 $n + 1$ 层所提供的服务实际上已包括了在它以下各层所提供的服务。第 n 层的实体对第 $n + 1$ 层的实体就相当于一个服务提供者。在服务提供者的上一层的实体又称为“服务用户”，因为它使用下层服务提供者所提供的服务。

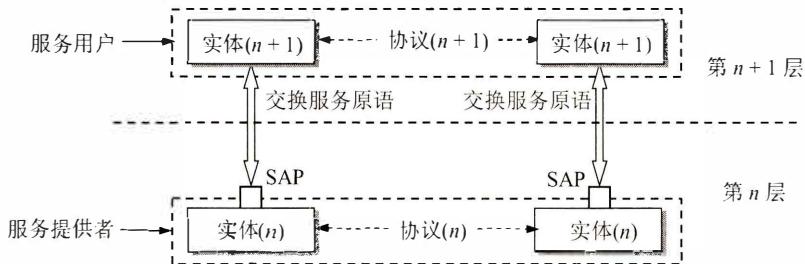


图 1-18 相邻两层之间的关系

计算机网络的协议还有一个很重要的特点，就是协议必须把所有不利的条件事先都估计到，而不能假定一切都是正常的和非常理想的。例如，两个朋友在电话中约好，下午 3 时在某公园门口碰头，并且约定“不见不散”。这就是一个很不科学的协议，因为任何一方临时有急事来不了而又无法通知对方时（如对方的电话或手机都无法接通），则另一方按照协议就必须永远等待下去。因此，看一个计算机网络协议是否正确，不能只看在正常情况下是否正确，还必须非常仔细地检查协议能否应付任何一种出现概率极小的异常情况。

下面是一个有关网络协议的非常著名的例子。

【例 1-1】占据东、西两个山顶的蓝军 1 和蓝军 2 与驻扎在山谷的白军作战。其力量对比是：单独的蓝军 1 或蓝军 2 打不过白军，但蓝军 1 和蓝军 2 协同作战则可战胜白军。现蓝军 1 拟于次日正午向白军发起攻击。于是用计算机发送电文给蓝军 2。但通信线路很不好，电文出错或丢失的可能性较大（没有电话可使用）。因此要求收到电文的友军必须送回一个确认电文。但此确认电文也可能出错或丢失。试问能否设计出一种协议使得蓝军 1 和蓝军 2 能够实现协同作战因而一定（即 100% 而不是 99.999...%）取得胜利？

【解】蓝军 1 先发送：“拟于明日正午向白军发起攻击。请协同作战和确认。”

假定蓝军 2 收到电文后发回了确认。

然而现在蓝军 1 和蓝军 2 都不敢下决心进攻。因为，蓝军 2 不知道此确认电文对方是否正确地收到了。如未正确收到，则蓝军 1 必定不敢贸然进攻。在此情况下，自己单方面发起进攻就肯定要失败。因此，必须等待蓝军 1 发送“对确认的确认”。

假定蓝军 2 收到了蓝军 1 发来的确认。但蓝军 1 同样关心自己发出的确认是否已被对方正确地收到。因此还要等待蓝军 2 的“对确认的确认的确认”。

这样无限循环下去，蓝军 1 和蓝军 2 都始终无法确定自己最后发出的电文对方是否已经收到（如图 1-19 所示）。因此，在本例题给出的条件下，没有一种协议可以使蓝军 1 和蓝军 2 能够 100% 地确保胜利。

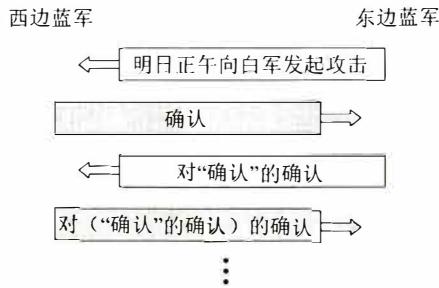


图 1-19 无限循环的协议

这个例子告诉我们，看似非常简单的协议，设计起来要考虑的问题还是比较多的。

1.7.5 TCP/IP 的体系结构

前面已经说过，TCP/IP 的体系结构比较简单，它只有四层。图 1-20 给出了这种四层协议表示方法的例子。

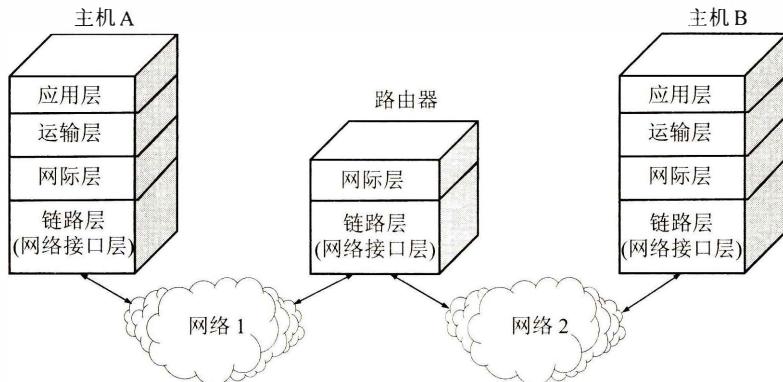


图 1-20 TCP/IP 四层协议的表示方法举例

应当指出，技术的发展并不遵循严格的 OSI 分层概念。实际上现在的互联网使用的 TCP/IP 体系结构有时已经演变成为图 1-21 所示的那样，即某些应用程序可以直接使用 IP 层，或直接使用最下面的链路层[PETE12]。虽然 TCP/IP 协议族得到了非常广泛的应用，但对 TCP/IP 体系结构的批评意见也有不少。例如，这个体系结构没有清晰地阐明区分开服务、接口和协议之间的关系，而链路层并非真正的一个层次，而仅仅是强调了 IP 层需要这样一个与网络的接口。这个体系结构没有把重要的物理层和链路层的内容包含进来。

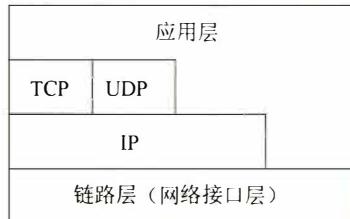


图 1-21 TCP/IP 体系结构的另一种表示方法

还有另一种方法用来表示 TCP/IP 协议族（如图 1-22 所示），它的特点是上下两头大而中间小：应用层和网络接口层都有多种协议，而中间的 IP 层很小，上层的各种协议都向下

汇聚到一个 IP 协议中。这种很像沙漏计时器形状的 TCP/IP 协议族表明：IP 层可以支持多种运输层协议（虽然这里只画出了最主要的两种），而不同的运输层协议上面又可以有多种应用层协议（所谓的 everything over IP），同时 IP 协议也可以在多种类型的网络上运行（所谓的 IP over everything）。正因为如此，互联网才会发展到今天的这种全球规模。从图 1-22 不难看出 IP 协议在互联网中的核心作用。

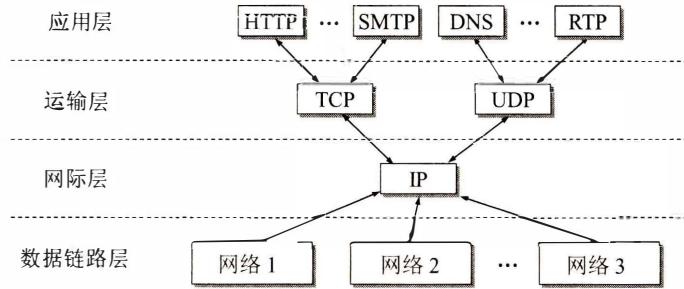


图 1-22 沙漏计时器形状的 TCP/IP 协议族示意

实际上，图 1-22 还反映出互联网的一个十分重要的设计理念，这就是网络的核心部分越简单越好，把一切复杂的部分让网络的边缘部分去实现。

【例 1-2】利用协议栈的概念，说明在互联网中常用的客户-服务器工作方式。

【解】图 1-23 中的主机 A 和主机 B 都各有自己的协议栈。主机 A 中的应用进程（即客户进程）的位置在最高的应用层。这个客户进程向主机 B 应用层的服务器进程发出请求，请求建立连接（图中的①）。然后，主机 B 中的服务器进程接受 A 的客户进程发来的请求（图中的②）。所有这些通信，实际上都需要使用下面各层所提供的服务。但若仅仅考虑客户进程和服务器进程的交互，则可把它们之间的交互看成是图 1-23 中的水平虚线所示的那样。

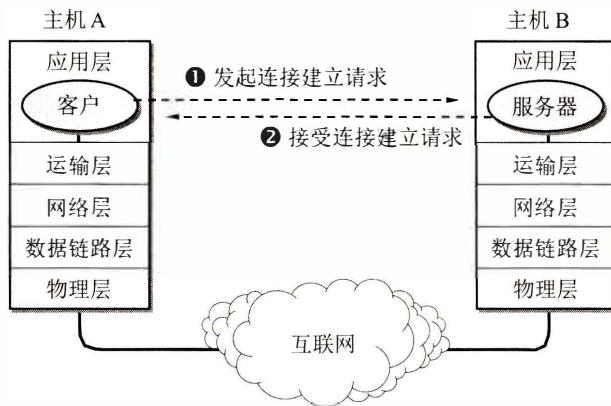


图 1-23 在应用层的客户进程和服务器进程的交互

图 1-24 画出了三台主机的协议栈。主机 C 的应用层中同时有两个服务器进程在通信。服务器 1 在和主机 A 中的客户 1 通信，而服务器 2 在和主机 B 中的客户 2 通信。有的服务器进程可以同时向几百个或更多的客户进程提供服务。