项目四　计算机网络的体系结构与协议

学习目标

知识目标

◆了解 OSI 参考模型的结构。

◆掌握 TCP/IP 参考模型的结构。

能力目标

◆能够进行网络结构设计。

◆能够进行 TCP/IP 的设置与管理。

素质目标

◆培养刻苦钻研与精益求精的精神。

◆培养良好的分析问题和解决问题能力。

思政目标

培养学生的系统思维和分析能力，增强国家意识和文化自信，培养学生的创新意识和实践能力，提高解决实际问题的能力。

项目描述

随着当今网络的迅速发展，通过网络进行公司文化宣传已经成为不可或缺的途径。现在 A 公司想要组建一个属于自己公司的网站，如果你是这个公司的网络管理员，请你搭建一个符合公司特点的网站。

知识准备

任务一　网络体系结构基本概念

一、网络体系结构的发展史

首先出现的计算机网络体系结构是 IBM 公司于 1974 年研制的 SNA（System Network Architecture，系统网络体系结构）。这种网络体系结构体现了分层的思想。SNA 不断发展，变更了几个版本。目前，它是世界上应用较为广泛的一种网络体系结构。之后，其他一些公司也相继推出不同的网络体系结构。

有了网络体系结构，一个公司生产的各种设备都能够按照协议互连成网。但是跨公司的产品因为分属不同的网络体系结构，采用的协议不同，就很难相互连通。这种情况方便了单个公司的垄断，因为用户一旦购买了某个公司的网络产品，那么后续扩展时就必须继续购买同一个公司的产品。而多个用户之间很有可能因为采用不同公司的网络产品而无法通信。然而全球经济一体化的迫切需求要求用户之间能够方便地交换信息。

1977 年，为了实现不同的计算机网络之间能够互连，国际标准化组织 ISO 成立了专门的研究机构。不久，他们提出了一个标准框架，叫做开放系统互连基本参考模型 OSI/RM（Open Systems Interconnection Reference Model），简称 OSI。“开放”是指：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、也遵循同一标准的其他任何系统进行通信。“系统”是指在现实的系统中与互连有关的各部分。历时 6 年，开放系统互连参考模型

OSI/RM 终于在 1983 年形成了正式的文件，即著名的 ISO7498 国际标准，它采用了七层协议的体系结构。

理想情况下，全世界的计算机网络都遵循 OSI 标准，那么任意计算机之间都将能够方便地进行互相通信。最初，许多大公司和政府机构纷纷表示支持 OSI，有人认为在未来全世界一定会按照 OSI 制订的标准来构造各自的计算机网络。然而，因特网抢先覆盖了全世界大部分的计算机网络，而这其中没有用到任何遵循 OSI 标准的网络产品，甚至，没有一家公司生产出符合 OSI 标准的商用产品。最终，OSI 仅仅获得了一些理论研究的成果，而没有实现市场化。现今规模最大的、覆盖全世界的计算机网络因特网采用的并非 OSI 标准，而是 TCP/IP 标准。

TCP/IP 标准是随着计算机网络的发展而生的，最初的计算机网络———ARPANET 就是采用 TCP/IP 标准，ARPANET 的继承者———因特网也是采用该标准。ARPANET 最初是租用电话线，将几百所大学和政府机构的计算机设备连接起来的。后来，卫星和无线电网络也加入了进来，原来的协议无法实现它们之间的互连。如何采用无缝的方式连接多种类型的网络是当时的主要研究目标。1974 年，Cerf 和 Kahn 发表了一篇文章，论述了一种体系结构，能够实现当时的需求，这就是采用 TCP/IP 标准的体系结构。1988 年，Clark 具体讨论了该体系结构背后的设计思想。

按照一般人的看法，任何技术与设备都必须符合相关国际标准才能得到大规模应用。然而，事实相反，官方制订的国际标准 OSI 并没有得到广泛的实际应用（甚至于几乎没有任何应用），而非官方形成的 TCP/IP 标准却成了事实上的国际标准。所以说，市场在一定程度上能够选择标准。

目前，因特网已经成了网络的代名词。当然，因特网的体系结构也在不断发展变化中。新一代网络体系结构也面临着一些挑战，主要有：

（1）大规模、可扩展；
（2）高性能；
（3）安全可信；
（4）移动性；
（5）异构共存（IPv4 到 IPv6 的过渡）；
（6）更好的管理；
（7）支持新应用；
（8）经济模式等等。

二、协议、层次、接口与体系结构的概念

协议，全称为网络协议，即在计算机网络中实现数据交换必须遵循的事先约定好的规则、标准或约定。网络协议主要由下面三个要素组成：

（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。

事实上，网络中任意两台计算机通信都必须有协议。它不同于计算机内部的数据通信，比如，我们经常在计算机上对文件进行读写操作，就不需要任何协议，除非读写的文件来自网络上另一台计算机。

协议都有两种形式。一种是采用文字描述的形式。另一种是使用计算机程序代码描述的形式。这两种形式的协议目的是一样的，都严格规定了网络上数据交换的规则。

计算机网络的设计经验告诉我们，对于非常复杂的计算机网络协议，必须采用层次化结构。层次化分解了协议的复杂度，它就像是将一件事情分为几个阶段来实现，每个阶段完成

部分工作，最终各阶段完毕后完成所有工作。我们以一个典型的网络软件———即时通信工具为例简单介绍一下层次化结构的工作原理。

如图 4-1 所示，现在假设用户 1 与用户 2 通过即时通信工具进行通信。这是一种很常用的工具，但其具体实现机制较为复杂。主要完成的工作可以划分为三类。

图 4-1　层次化结构举例

第一类工作负责信息传送，信息有多种类型，用户之间发送的短信息、文件、图片等等，都可以称之为信息。信息传送程序专门负责为信息分类、编码，确信对方的信息接收程序是否准备好，然后按照一定的协议对信息进行封装。接收方收到信息后，根据协议对信息进行分析解包，最终完成用户之间的信息交换。这些工作可以交给一个信息传送模块来完成。

如果让信息传送模块完成全部通信功能，将会使信息传送模块过于复杂。可以在信息传送模块下方再设立一个通信服务模块，用来确保信息传送命令在两个计算机之间正确交换。换句话说，信息传送模块利用下面的通信服务模块提供的服务完成自身的工作。可以看出，如果我们将信息传送模块换成文件传送模块，那么文件传送模块一样可以利用它下面的通信服务模块所提供的可靠通信的服务。

另外，还可以在通信服务模块下方再构造一个网络接入模块，让这个模块负责做与网络接口有关的工作，并向通信服务模块提供服务，使上层模块能够完成可靠通信的任务。

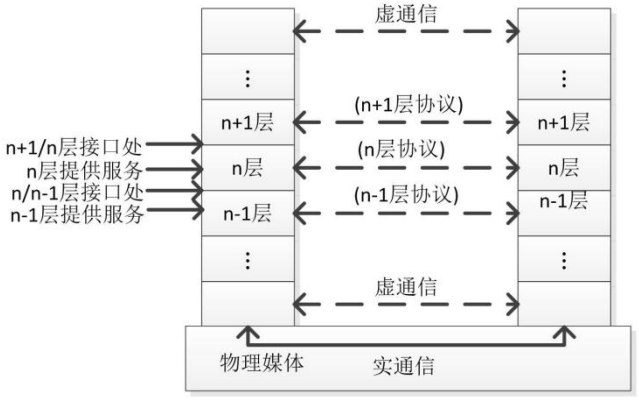


图 4-2　计算机网络的层次化模型

计算机网络的层次化模型示意图如图 4-2 所示：

层次化的要点：

（1）只有物理媒体上进行的是实通信，其余各对等实体间进行的都是虚通信。

（2）对等层的通信必须遵循该层的协议。

（3）n 层的虚通信是通过 n 层与 n－1 层间接口处 n－1 层提供的服务以及 n－1 层的通信来实现的。

层次与协议是相联系的，每一个层次都对应各自的协议。层次与协议组成了体系结构。

同时，层次化与体系结构一样，都属于抽象的概念。计算机网络的体系结构是这个计算

机网络及其部件所应完成的功能的精确定义。网络的体系结构的特点是：

（1）以功能作为划分层次的基础。

（2）第 n 层的实体在实现自身定义的功能时，只能使用第 n－1 层提供的服务。

（3）第 n 层在向第 n＋1 层提供的服务时，此服务不仅包含第 n 层本身的功能，还包含由下层服务提供的功能。

（4）仅在相邻层间有接口，且所提供服务的具体实现细节对上一层完全屏蔽。

接口负责相邻各层之间的交互，向上负责数据的解析，端口的分用；向下负责数据的封装，端口的复用。一个具有明确定义的接口便于各层的协议制订，各层实现时只需调用接口所提供的服务即可。

三、网络分层的意义与层次划分的原则

网络分层可以带来如下好处：

（1）功能独立。层与层之间相互独立，每一层都不用关心下一层如何实现，只需知道层间的接口提供的服务即可。由于各层实现的功能相对独立，所以复杂的网络通信问题就被划分成了多个相对简单的问题。因此，整个问题的复杂度就降低了。

（2）灵活。如果某一层由于技术进步，发生了一些改变，只要保持层间关系不变，就不会影响上下层。另外，还可以在某一层上增加功能，以满足新的服务需求，不需要时可随时删掉该功能，只要不影响层间接口即可，这就给各层的实现带来了极大的灵活性。

（3）隔离性好。各层都可以采用最合适的技术实现，互不影响。

（4）容易维护和实现。分层结构将一个完整的系统划分为若干个独立的子系统，这使得实现和调试一个庞大复杂的系统变得容易。

（5）促进标准化工作。因为各层功能与服务都必须有精确的说明，很容易进一步形成标准。分层的工作非常重要，它涉及网络协议的制定，每一层的具体功能必须非常明确。若层数太少，将会使各层协议趋于复杂。而层数过多又会导致在描述和综合各层功能并制定协议时遇到更多的困难。各层的功能概括如下：

（1）差错控制

使通信更加可靠。

（2）流量控制

限制发送端的发送速率，要使接收端来得及接收。

（3）分段和重装

发送端将数据划分为更小的单位，接收端收到后将其还原。

（4）复用和分用

发送端的多个高层应用可以复用一条低层的连接，在接收端必须进行分用。

（5）建立与释放连接

交换数据前必须先建立一条逻辑连接。数据传送结束后释放连接。

分层也有缺点，比如，有些功能可能在多个不同层次中重复出现，增加实现开销。所以，一个好的体系结构，必须兼顾这些问题，设计的层次既不能过少，以免实现起来过于繁琐、复杂；也不能过多，以免功能过度重叠，导致通信效率低下。

OSI 参考模型的分层原则如下：

（1）当需要一个新的抽象体时，应该创建一层。

（2）每一层的功能都必须是明确定义的。

（3）每一层功能的选择都必须同时考虑制订国际标准化协议。

（4）层边界的选择时必须考虑使跨边界所需要的信息越少越好。

（5）层数必须足够多，保证不同的功能由不同的层来实现，但也不能太多，以免体系

结构过于庞大。

任务二　开放系统互连（OSI）参考模型

一、OSI 参考模型

虽然 OSI 参考模型的实际应用意义不是很大，但它是由国际标准化组织中的网络、通信多个方面的顶级专家共同研究制订的，所以学习 OSI 参考模型对理解网络协议内部的运作非常有帮助，也为我们学习网络协议提供了一个很好的参考。OSI 参考模型共分 7 层，又称为 7 层模型，如图 4-3 所示：

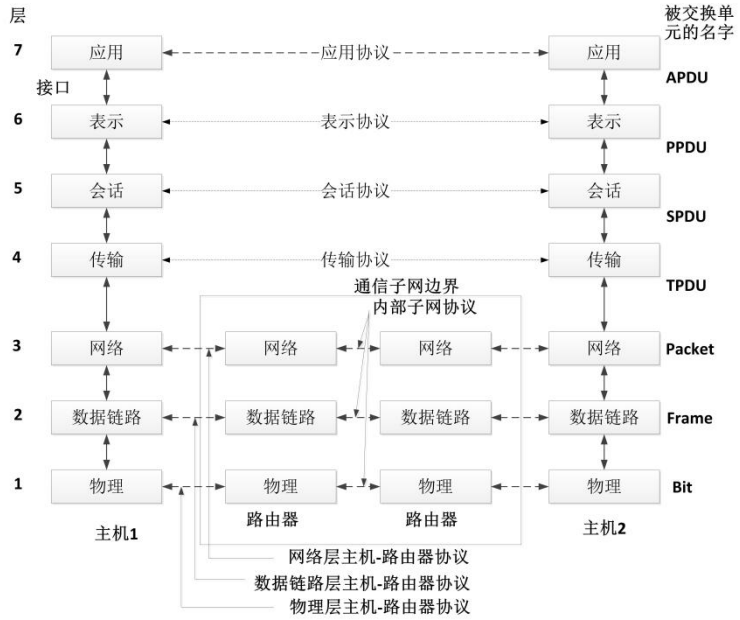


图 4-3　OSI 参考模型

下面从最底层开始介绍 OSI 参考模型的每一层。OSI 参考模型本身没有为每一层精确定义需要用到的服务和协议。它只是指明了每一层上的具体功能。当然，ISO 已经出台了相应的标准，只是这些标准并不属于参考模型本身。

1．物理层

物理层主要考虑在通信信道上原始比特流的传输。设计时必须保证，发送方发送比特 1 时，接收方不能接收为比特 0，必须保持原来的值。因此物理层的设计就需要考虑：比特 1 和 0 用多大的电压来表示；每个比特的传输应该持续多长时间；传输通道是否采用双向的；传输通道如何建立和撤销；网卡物理接口如何设计与实现。设计时需要考虑机械、电气和物理传输介质等问题。

2．数据链路层

数据链路层的作用主要是将原始传输设备上没有被发现的错误呈现给网络层。它的做法是：将发送的数据分成若干个数据帧（dataframe，通常由几百个或上千个字节组成），并按照顺序将这些数据帧发送出去。如果数据链路层采用可靠服务，那么接收者在接收到每个发送帧后还必须给发送方一个确认帧（acknowledgement frame，字节数较少）。

数据链路层的另一个职能是保证一个快速的发送方不能淹没一个慢速的接收方。通常采用流量调节机制来完成这个功能，即接收方要设法通知发送方自身的网络接收缓存的大小以及剩余空间。通常情况下，流量调节功能和错误处理功能是整合在一起的。

数据链路层在广播式网络上还需解决一个问题：如何控制共享信道的访问。数据链路层的一个子层是专门负责解决这个问题的，即介质访问控制子层。

3．网络层

网络层负责控制整个子网的运转。其中一个关键功能就是路由功能，即如何将一个包从源端路由到目标端。路由表可以是静态的、很少改变的。这些路由可能在每次通信开始时都是确定的，就像终端会话，比如登录到一台远端机器上。当然，网络层也可以采用动态路由，它可以在每个包发送时都可以根据网络负载的状态来重新选择路由。

网络拥塞控制是网络层需要实现的功能，所谓网络拥塞是指：同一时间，在子网上出现的包过多，就会造成传输的瓶颈。另外，网络服务质量（QoS，the quality of service）也属于网络层的功能部分，QoS 包括网络延迟、传输时间和抖动等。

当一个包是从一个网络传到另外一个网络，这中间可能会发生很多问题。比如，第二个网络的编址可能不同于第一个网络；第二个网络可能根本无法接收第一个网络发送的包，因为包太大了；这两个网络之间的协议可能完全不同，等等一系列问题。网络层必须解决这些问题，保证异构的网络能够互连互通。

在一个广播式网络中，路由问题相对简单，所以网络层比较薄，甚至没有网络层也可以工作。

4．传输层

传输层的作用是将上一层传下来的数据拆分成更小的单元传给网络层，最终保证网络另一端的计算机的传输层能够正确接收数据。另外，还必须保证这些工作的高效，并且为上一层屏蔽硬件实现细节，即硬件技术的变化不会对上层有任何影响。

传输层为会话层（最终是向网络用户）提供服务。其中采用的最流行的服务类型是，传输层上的传输连接是无错的点到点信道，按照原始报文的顺序依次发送数据。传输层也可能采用其他服务类型，比如，传送独立的不按顺序的报文，将报文广播给多个接收端。服务的类型是在连接建立一开始所决定的。真正的无错误连接是不可能实现的，之所以提到这种服务类型，主要是说它的错误出现情况非常小，以致在实际应用中可以忽略不计。

对用户来说，传输层是真正的端到端的层，一个用户向另一个用户发起通信，最终就是在传输层上建立的端到端的连接。换句话说，源端与目的端的通信程序是类似的（甚至是一模一样的），并且使用同样的报文头部和控制报文。在底层，与源端机器直接相连的机器通常不是真正的目的端机器，传输层意义上的两端之间可能被很多路由器分隔开来。图 3－3 所示的第 1 层到第 3 层中的机器之间是直接串联起来的，而第 4 层到第 7 层是端到端的。

5．会话层

会话层允许不同机器上的用户之间建立会话。会话层提供的服务包括：

对话控制，记录对话方的轮转；

令牌管理，禁止双方同时执行一个关键操作；

同步，在传输过程中设立检查点，这样，如果在一次较长的数据传输过程中系统崩溃，那么恢复后仍然能够继续上一个检查点继续传输数据。

6．表示层

表示层与其它层不同，其它层主要关心数据位的传输，表示层关心的是信息中的语法和语义的传输。不同的计算机的数据表示法可能有所不同，表示层的功能就是要保证具有不同数据表示法的计算机之间能够正常通信。表示层采用的方法通常是定义一种抽象的数据结构，数据按照这种标准编码后再进行通信传输。表示层负责管理这些抽象的数据结构，并允许高层数据结构的定义与交换。

7．应用层

应用层定义的各种协议都是针对用户的实际需要。例如，HTTP（Hyper Text Transfer Protocol，超文本传输协议）是一种广泛使用的协议，它是万维网（World Wide Web，WWW）的基础。当用户使用浏览器访问页面时，必须采用 HTTP 协议向服务器发送页面的名字。

服务器收到请求后将相应的页面发给用户。其它的应用层协议实现的功能包括：文件传输、电子邮件、网络新闻以及即时通信等等。

二、OSI 环境中的数据传输过程

OSI 环境中数据的实际传递过程如图 4-4 所示：

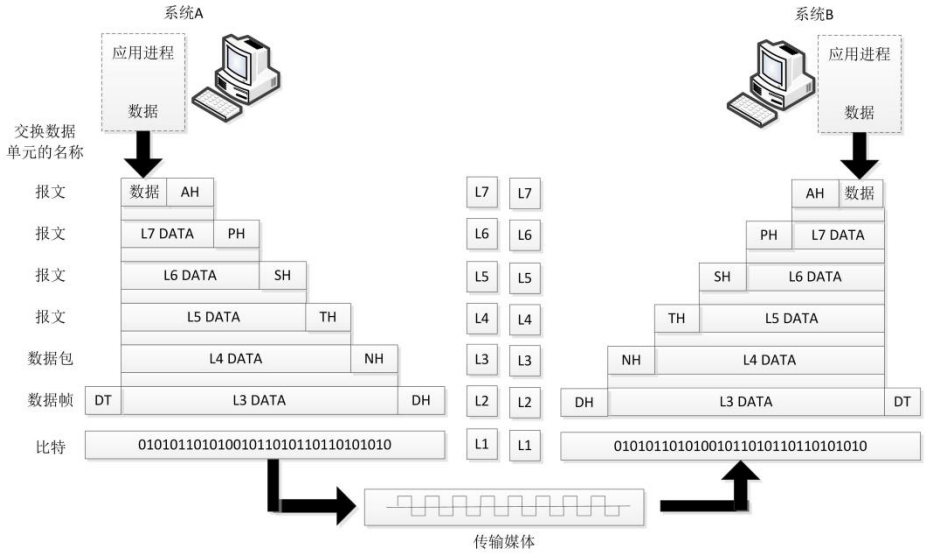


图 4-4　OSI 网络结构中的数据传递

用户打开程序，准备数据通信，对操作系统而言将建立一个应用进程。应用层协议按照一定的通信规则，将待传送的数据加上协议首部，协议首部中通常需要存放的信息有：通信数据的长度、报文的编号、校验和等；应用层将封装好的报文传递给表示层。同理，表示层也按照自己的通信规则，将上层传下来的报文加上协议首部，并将其传递给会话层。以此类推，最终报文到达网络层后，将按照网络上的最大协议单元长度规则，报文被拆分成多个数据包，每个数据包都需要加上首部并传递给数据链路层。数据链路层将数据包变成数据帧，并传递给物理层。物理层最终通过数模转换，将数据转换成模拟信号发送出去。

任务三　网络各层的功能和设计要点

一、物理层

物理层的功能是在网络中各计算机之间的传输媒介上传输数据比特流，它不必考虑具体的物理设备和传输媒介本身。然而，现有的计算机网络中存在各种类型的物理设备和纷繁复杂的传输媒介，比如磁介质、双绞线、同轴电缆、光纤等等，通信手段也多种多样，比如无线传输、有线传输，无线传输又分为电磁波谱、无线电传输、微波传输、红外线和毫米波、光波传输等等，有线传输可以采用有线电视、公共交换电话网络等等。

物理层的作用就是要尽可能屏蔽这些差异，使物理层上面的数据链路层无需考虑底层的具体技术细节，使得数据链路层只需完成本层的协议和服务。通常，使用物理层的协议也常称为物理层规程，实质上就是物理层协议。物理层的主要功能是确定与传输媒介的接口的一些特性，即：

（1）机械特性

指明接口所用接线器的形状和尺寸、引线数目和排列、固定和锁定装置等等。

（2）电气特性

指明在接口电缆的各条线上出现的电压的范围。

（3）功能特性

指明某条线上出现的某一电平的电压表示何种意义。

（4）规程特性

指明对于不同功能的各种可能事件的出现顺序。

计算机网络的物理层上通常都采用串行传输，因为串行传输对时序和电磁干扰要求较低。当然，短距离范围有时也可以采用多个比特的并行传输方式。出于经济上的考虑，远距离的传输通常都采用串行传输。

具体的物理层协议是非常复杂的。因为物理连接方式多种多样，比如，可以是点对点连接，也可以是多点连接，或者采用广播连接等等。

最常用的物理层标准有两种：

1．EIA－232－E 接口标准

EIA－232－E 是美国电子工业协会 EIA 制订的物理层异步通信接口标准。它最早是 1962 年制订的标准 RS－232。这里的 RS 表示是 EIA 的一种“推荐标准”，232 是个编号。

在 1969 年修订为 RS－232－C，C 是标准 RS－232 以后的第三个修订版本。1987 年 1 月，修订为 EIA－232－D。1991 年又修订你给为 EIA－232－E。由于标准修改得并不多，因此现在很多厂商仍然使用旧名称。有时简称为 EIA－232，甚至说得更简单些：“提供 232 接口”。

EIA－232 是 DTE 与 DCE 之间的接口标准。因此下面先介绍什么是 DTE 和 DCE。

DTE（Data Terminal Equipment，数据终端设备）是具有一定的数据处理能力以及发送和接收数据能力的设备。大家知道，大多数的数字数据处理设备的数据传输能力是很有限的。直接将相隔很远的两个数据处理设备连接起来，是不能进行通信的。必须在数据处理设备和传输线路之间，加上一个中间设备。这个中间设备就是数据通信设备（DCE，Data Communication Equipment）。DCE 的作用是在 DTE 和传输线路之间提供信号变换和编码的功能，并且负责建立、保持和释放数据链路的连接。如图 4-5 所示，数据终端设备通过 DCE 之后连接到通信链路上。

DTE 可以是计算机，也可以是终端设备或者各种 I/O 设备。典型的 DCE 则是一个与模拟电话线路相连接的调制解调器。从图 3－5 中可以看出，DCE 虽然处于通信环境内，但它和 DTE 均属于用户设施。用户环境只有 DTE。

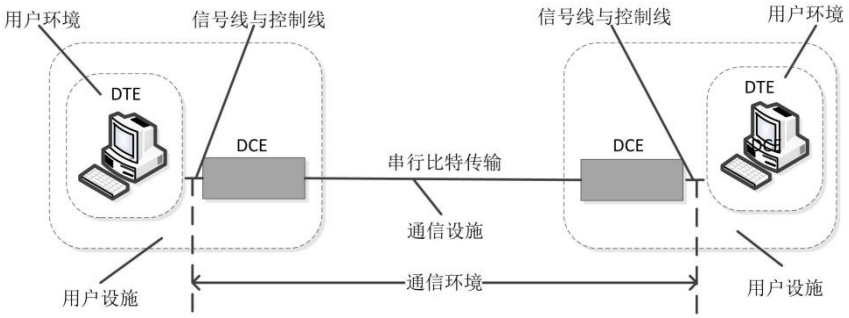


图 4-5　数据终端设备通过 DCE 与通信链路相连

DTE 与 DCE 之间的接口由许多信号线和控制线组成。DCE 将 DTE 传过来的数据，按照比特先后顺序逐个发往传输线路，或者反过来，从传输线路上接收串行的比特流，然后再交给 DTE。很明显，这里需要高度协调地工作。为了减轻数据处理设备用户的负担，就必须对 DTE 和 DCE 的接口进行标准化。这种接口标准就是所谓的物理层协议。

多数的物理层协议使用如图 4-5 所示的模型。但也有不同的物理层协议。例如，在局域网中，物理层协议多定义的是一个数据终端设备和链路的传输媒介的接口，而并没有使用

DTE/DCE 模型。有一种常用的 DCE 类型叫做 CSU/DSU，意思是信道服务单元/数据服务单元（Channel Service Unit/Data Service Unit），它将 DTE/DCE 接口转换为通常的电话接口。

物理层标准 EIA－232 的主要特点如下。

机械特性：EIA－232 使用 ISO2110 关于插头座的标准。即使用 25 根引脚的 DB－25 插头座。引脚分为上、下两排，分别有 13 和 12 根引脚，其编号分别规定为 1 至 13 和 14 至 25，都是从左到右。

电气特性：EIA－232 与 CCITT 的 V．28 建议书相同。EIA－232 采用负逻辑，即逻辑 0 相当于对信号地线有＋3V 或更高的电压；而逻辑 1 相当于对信号地线有－3V 或更负的电压。逻辑 0 相当于数据的“0”（空号）或控制线的“接通”状态，而逻辑 1 则相当于数据的“1”（传号）或控制线的“断开”状态。当连接电缆线的长度不超过 15m 时，允许数据传输速率不超过 20kb/s。但是当连接电缆长度较短时，数据传输速率就可以大大提高。

EIA－232 的功能特性与 CCITT 的 V．24 建议书一致。它规定了什么电路应当连接到 25 根引脚中的哪一根以及该引脚的作用。图 3－6 中描述了最常用的 10 个引脚的作用，括弧中的数目为引脚的编号。其余的一些引脚可以空着不用。图中引脚 7 是信号地，即公共回线。引脚 1 是保护地（即屏蔽地），有时可不用。引脚 2 和引脚 3 都是传送数据的数据线。

“发送”和“接收”都是对 DTE 而言。有时只用图中的 9 个引脚（将“保护地”除外）制成专用的 9 芯插头，供计算机与调制解调器的连接使用。

EIA－232 的规程特性规定了在 DTE 与 DCE 之间所发生的事件的合法序列。这部分内容与 CCITT 的 V．24 建议书相同。

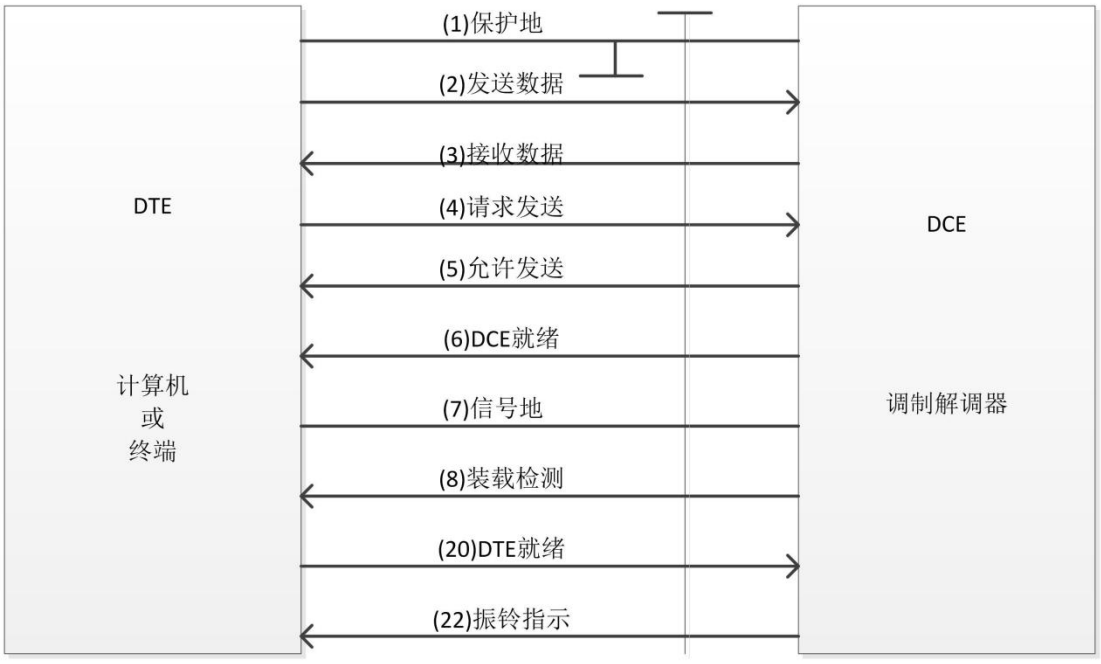


图 4-6　EIA－232/V．24 的信号定义

图 3-7 所示的例子描述了 DTE－A 向 DTE－B 发送数据的过程。

图 4-7　两个 DTE 通过 DCE 进行通信的例子

（1）当 DTE－A 要和 DTE－B 进行通信时，就将引脚 20“DTE 就绪”置为 ON，同时通过引脚 2“发送数据”向 DCE－A 传送电话号码信号。

（2）DCE－B 将引脚 22“振铃指示”置为 ON，表示通知 DTE－B 有入呼叫信号到达（在

振铃的间隙以及其他时间，振铃指示均为 OFF 状态）。DTE－B 就将引脚 20“DTE 就绪”置为 ON。DCE－B 接着产生载波信号，并将引脚 6“DCE 就绪”置为 ON，表示已准备好接收数据。

（3）当 DCE－A 检测到载波信号时，将引脚 8“载波检测”和引脚 6“DCE 就绪”都置为 ON，以便使 DTE－A 知道通信电路已经建立。DCE－A 还可通过引脚 3“接收数据”向 DTE－A 发送在其屏幕上显示的信息。

（4）DCE－A 接着向 DCE－B 发送其载波信号，DCE－B 将其引脚 8“载波检测”置为 ON。

（5）当 DTE－A 要发送数据时，将其引脚 4“请求发送”置为 ON。引脚 5“允许发送”置为 ON。然后 DTE－A 通过引脚 2“发送数据”来发送其数据。DCE－A 将数字信号转换为模拟信号箱 DCE－B 发送过去。

（6）DCE－B 将收到的模拟信号转换为数字信号经过引脚 3“接收数据”向 DTE－B 发送。

其他的一些引脚的作用是：选择数据的发送速率，测试调制解调器，传送数据的码元定时信号，以及从另一个辅助信道反向发送数据等等。但是这些引脚在实际中很少使用。

许多产品都声称自己的串行接口与 EIA－232 标准兼容。这句话的意思只是说明该产品的接口的电气特性和机械特性与 EIA－232 接口标准没有矛盾。并不能说明该接口是否能够支持 EIA－232 的全部功能。这是因为，很多厂商出售的调制解调器只使用了接口的 25 根引脚中的 412 根。因此他们所实现的很可能只是整个 EIA－232 标准的一个子集。因此应弄清你所需要的性能是否已包括在这个子集之中。

EIA 还规定了插头应装在 DTE 上，插座应装在 DCE 中。因此当终端或计算机与调制解调器相连时就非常方便。然而有时却需要将两台计算机通过 EIA－232 串行接口直接相连。这显然有点麻烦。例如，这台计算机通过引脚 2 发送数据，但仍然传送到另一台计算机的引脚 2，这就使对方无法接收。为了不改动计算机内标准的串行接口线路，可以采用虚调制解调器的方法。所谓虚调制解调器就是一段电缆，具体的连接方法如图 4-8 所示。这样对每一台计算机来说，都好像是与一个调制解调器相连，但实际上并没有真正的调制解调器存在。

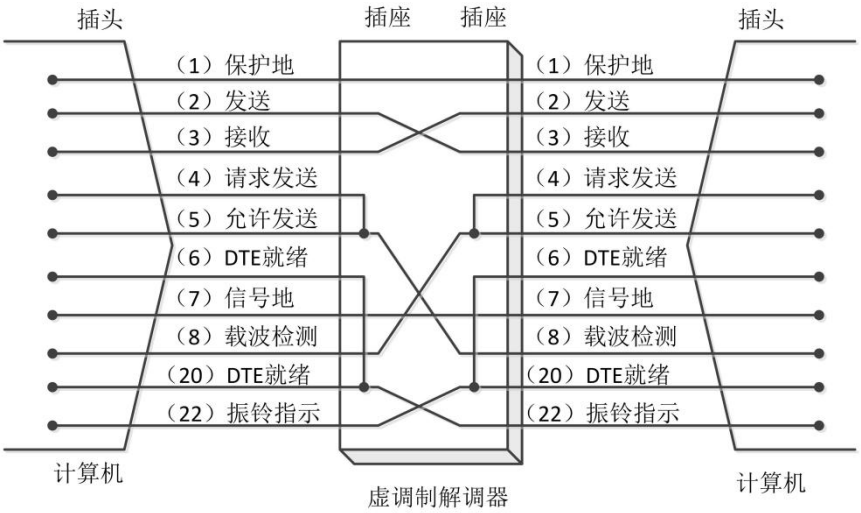


图 4-8　利用虚调制解调器与两台计算机相连

2．RS－449 接口标准

EIA－232 接口标准有两个较大的缺点，即：

◆数据的传输速率最高为 20kb/s；

◆连接电缆的最大长度不超过 15m。

这就促使人们制订性能更好的接口标准。处于这种考虑，EIA 于 1977 年又制定了一个新的标准 RS－449，以便逐渐取代旧的 RS－232。

实际上，RS－449 由 3 个标准组成。即：

（1）RS－449

规定接口的机械特性、功能特性和过程特性。RS－449 采用 37 根引脚的插头座。在 CCITT 的建议书中，RS－449 相当于 V．35。

（2）RS－423－A

规定在采用非平衡传输时（即所有的电路公用一个公共地）的电气特性。当连接电缆长度为 10m 时，数据的传输速率可达 300kb/s。

（3）RS－422－A

规定在采用平衡传输时（即所有的电路没有公共地）的电气特性。它可将传输速率提高到 2Mb/s，而连接电缆长度可超过 60m。当连接电缆长度更短时（如 10m），则传输速率还可以更高些（如达到 10Mb/s）。

通常 EIA－232/V．24 用于标准电话线路（一个话路）的物理层接口，而 RS－449/V．35 则用于宽带电路（一般都是租用电路），其典型的传输速率为 48168kb/s，都是用于点到点的同步传输。

二、数据链路层

从数据链路层开始到上层，我们通常采用虚通道的方式来介绍对等层之间的通信，因为大家知道，实际的通信只能在物理层才能实现，真正传输的信号是在物理媒介上传送电信号或者光信号。在数据链路层上传送的数据单元，我们称之为帧。这种说法就已经屏蔽了底层的技术实现细节。

数据链路层的主要功能如下：

1．给网络层提供服务

数据链路层的主要功能是为网络层提供服务。它的主要任务就是将源机器网络层的数据传送到目标机器的网络层之上。在源机器端，网络层中一个实体进程将一些数据位交给数据链路层并将其发送到目标机器。对数据链路层来说，它的工作就是建立一条虚拟路径，将数据位直接传递到目标机器中，如图 4-9（a）所示。但是实际的数据传输还是如图 4-9（b）所示。学习中，我们经常使用图 4-9（a）所示的虚拟通信模型作为参考，而不采用图 4-9（b）的实际通信模型，因为这样更直观一些。

数据链路层提供的服务类型多种多样，与系统实现有关，大体上可分为三类：

（1）无确认的无连接服务

这种服务类型是指，源机器向目标机器发送独立的帧，目标机器不用向源机器发送应答信息。通信过程中无需建立逻辑连接，当然，也不需要释放逻辑连接。当发生干扰而丢帧时，采用这种服务类型的数据链路层没有检测和恢复丢帧的功能，它将这种服务留给上一层来完成，所以这种服务类型仅适用于错误发生很少的网络环境。这种场景比较适合于实时通信，比如，语音通信。语音通信中数据延迟比数据丢失更不能容忍。许多局域网的数据链路层都采用这种服务类型。

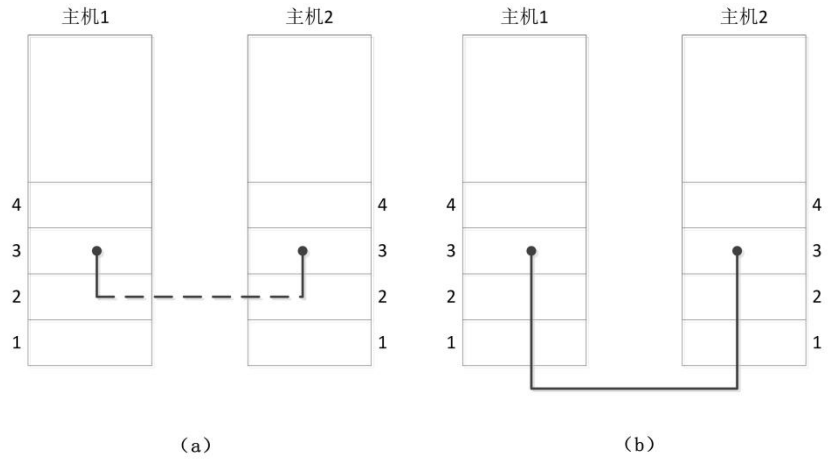


图 4-9　数据链路层

（a）虚拟通信（b）真实通信

（2）有确认的无连接服务

有确认的服务能够提供更高的可靠性。这类服务没有使用逻辑连接，所以发送的仍然是独立的帧。由于接收方对每个帧都会发送应答信息，所以发送方能够确认每个帧是否发送到了接收方。如果在特定的一段时间内没有收到应答信息，发送方将再次发送该帧。通常在一些不可靠的信道上采用这种服务类型，比如无线环境。

应该说，数据链路层上的确认属于整个网络系统的优化措施，而不是必须的。比如，在网络层上采用的是有确认的服务，那么在网络层上发送一个数据包，必须得到接收方的应答才有效。如果网络层上的数据包很长，传到数据链路层后必须分割为 10 帧，那么在数据链路层上即使丢帧率为 20%（假设数据链路层采用无确认的服务），那么网络层上的数据包几乎每次都要重传，并且重传成功的概率也很低。为此，数据链路层中，在不可靠的信道上采用有确认的服务将会优化整个系统的工作效率。当然，在诸如光纤等可靠地信道上大可不必采用有确认的服务。

（3）有确认的面向连接服务

这类服务是最复杂的。因为，它在传送数据之前都必须建立通道连接。发送过程中，每一帧都被编号，并在数据链路层中保证发送的成功。另外，它还需要保证按序接收。可以想象，采用这种服务类型的数据链路层，在发送网络层的一个数据包时将会收到很多条应答信息，并且它能够提供相对可靠地比特流。

面向连接的服务发送数据需要经历三个阶段。第一阶段，建立连接，初始化计数器和变量，用来记录帧的发送与应答的接收。第二阶段，传输数据帧。第三阶段，释放连接，释放变量、缓冲区和各种资源。

下面分析一个例子，如图 4-10 所示：

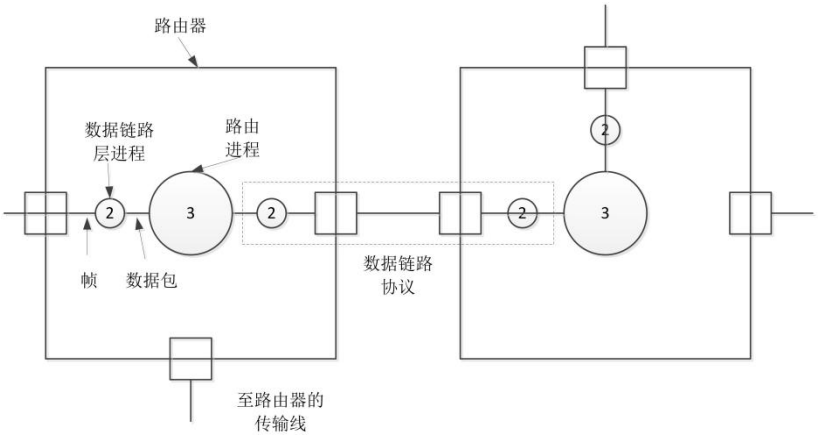


图 4-10　数据链路层的位置

在一个广域网中，有一个使用电话线连接多个路由器的子网。当一个数据帧到达一个路由器，首先硬件检查一下是否有错，接着将其传送给数据链路层的软件（可能是网卡芯片中的固件程序）。数据链路层检验一下接收的是否为期望的帧，如果是，将该帧内的数据包信息解析出来交给路由软件，路由软件选择合适的路由线路，将数据包再传给数据链路层软件，最终由其发送。

2．数据帧

数据链路层使用物理层提供的服务来完成为网络层服务。然而，物理层只接收原始的比特流并尝试将其发送到目的地。这个比特流无法保证正确无误。甚至于，数据位的个数可能比实际应该发送的不一致，并且值可能不同。这些都需要靠数据链路层来检测和错误校验。

通常，数据链路层会将比特流切分为若干离散的数据帧，并为每一帧计算校验和。当这一帧到达目的地后，校验和被重新计算，如果与原来的不一致，则说明数据在传送的过程中发生了错误。数据链路层负责进行校验并处理发生的错误（丢弃错误帧，也可能会回执一个错误报告）。

将比特流切分成数据帧可能不像说起来那么容易。一种形成数据帧的方法是在中间插入时间间隔（time gaps），就像在文章中每个单词间插入空格一样。然而这种方法的风险很大，因为网络中无法保证时间的正确性。

下面有四种形成数据帧的方法：

（1）字符计数法

在每一帧的开头填充该帧的实际长度，以便接收方知晓该帧的开头和结尾处。然而数据在传输中可能会出错，比如正好错误发生在该帧的开头长度域。实际长度 100 变成了 255，即便通过校验和可以确定该帧是错误的，但仍然无法确定到哪里结束接收该帧。所以这种方法已经很少使用了。

（2）含字节填充的分界符法

这种方法在帧的开始和结束处都加上相同内容的字节，也叫做标志字节（flag byte）。

这样，即便数据传输过程中发生了错误，也可以通过搜索匹配标志字节来重新同步。如果数、据中出现的与标志字节相同的字符，数据链路层协议负责将其前面加上一个转义字符，以免发生错误分界。同理，如果数据中出现了与转义字符相同的字符，则再加上一个转义字符。

这种方法的缺点是假定了传输的数据都是 8 位字节码，然而，这种假定不一定合适。

（3）含位填充的分界标志法

这种方法允许字符有任意长度的位。它的工作方式是，每一帧的开始和结尾都有一个特殊的比特串———01111110。当发送方的数据链路层经过的数据中有连续 5 个“1”时，它

会自动在其后填充一个“0”。接收时自动去掉连续 5 个“1”后面的“0”再将数据送往上层。采用这种方法，数据流中出现连续 6 个“1”时说明是帧的开始或结束，不可能是数据本身。同步过程很容易，只需扫描数据流，发现有连续 6 个“1”时则说明到了帧的边界处。

（4）物理层编码违例法

只有物理层上的编码中包含冗余信息才能使用该法。比如，物理介质上采用“高－低”两个电平组合表示“1”，“低－高”两个电平组合表示“0”。这时，可以使用“高－高”或“低－低”这样的电平组合确定帧的边界。

实际上，很多数据链路协议综合采用多种方法形成数据帧，确保安全可靠。过程中，只有发现数据帧的分界符，计算数据帧校验和无误才能确定该帧是有效的。

3．错误控制

仅仅在数据接收端确定帧的有效性是不够的，有时候发送方还需要确认帧是否按顺序发送到目标机器。当然，对于无确认的无连接服务，即使发送方只管发送数据，不顾对方是否收到数据也是能够满足要求。但是，对于有确认的面向连接的服务，这样做是不合适的。

为了确保数据的可靠交付呢，通常做法是，接收方向发送方回执信息。另外还需要额外的辅助设计，定时器和数据帧序号。一旦发送方在一段时间内没有接收到回执信息，则重传原数据帧。为了避免接收方重复接收数据帧，需要为每个数据帧编号，以便接收方能够区别重复帧。

4．流控制

如果发送方的帧发送速度超过接收方，将有可能淹没接收方，使得很多数据帧没来得及接收就丢掉了。这种情况是很多的，比如接收方的机器很慢，而发送方的机器很快，就可能造成这种问题的发生，数据链路层需要解决这个问题。通常，数据链路层通过流控制来限制上面问题的发生。通常采用两种方法来解决这个问题。

（1）基于反馈的流控制

接收方会向发送方提供反馈信息，比如告诉发送方自身的状态。采用这种方法的协议通常规定了发送方什么时候可以发送后面的数据帧，在没有得到接收方许可的情况下，发送方不能继续发送数据。比如，数据传送过程中，接收方可能会向发送方提供这样的信息：“你现在可以发送 n 个数据帧，发完后停止，等待我的进一步反馈。”

（2）基于速率的流控制

这种方法主要是通过限制发送方的发送速率来实现的，无需接收方的反馈信息。由于限制了网络传输速率，数据链路层的协议通常不会采用这种方法。

三、网络层

物理层和数据链路层只需考虑与物理上相邻的机器之间的数据传输，网络层与它们不同，它必须保证数据包从源端发送到目标端（数据最终终止传输的接收端）。为此，它需要提供以下服务。

1．存储－转发数据包交换

图 4-11 描述了网络层协议的运行环境。系统主要由客户设备和网络运营商两部分组成，其中阴影部分属于网络运营商的设备。主机 H1 通过一条租用线路直接连到网络运营商的路由器 A。主机 H2 通过公司内部 LAN 先连接到公司的出口路由器 F 上，F 再通过一条租用线路连到网络运营商的路由器 E 上。

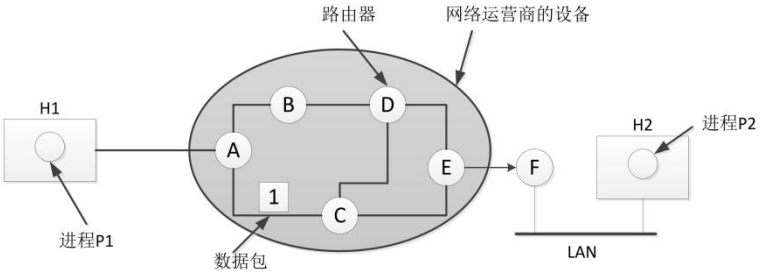


图 4-11　网络层协议的环境

如果主机 A 需要发送数据，那么工作过程大致如下：

（1）首先将数据包发送给最近的路由器 A；

（2）路由器 A 将数据包接收并存储在其上，直到接收完毕并校验无误才将其沿路发送给下一个路由器，直到到达目标主机为止；

（3）目标主机接收到数据包后将其递交给相应的进程。

这就是所谓的存储－转发数据包交换过程。

2．向传输层提供服务

网络层需要为传输层提供服务，它提供的服务需要基于以下几条原则：

◆服务必须独立于路由器技术。

◆对传输层来说，路由器的数量、类型和拓扑结构应该是透明的。

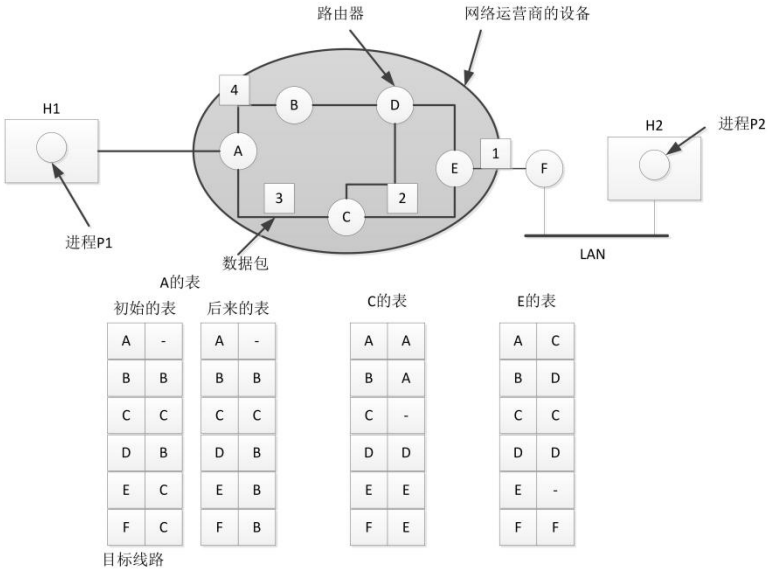
◆传输层必须使用统一的网络编址，以便跨越局域网甚至广域网。

这些原则对于网络层设计者来说是比较自由的，网络层可以为上层提供两大类服务，一种是面向连接服务，一种是无连接的服务。下面依次介绍两类服务。

3．无连接服务

和数据链路层一样，网络层的无连接服务也无需在数据传输前建立网络连接。所有数据包都独立于路由进行传输。在这种环境中，数据包（packet）通常称为数据报（data gram，类似于电报 telegram），子网称为数据报子网（data gram sub net）。

图 4-12 描述了数据报子网的工作过程。



目标线路	初始的表
A	-
B	B
C	C
D	B
E	C
F	C

目标线路	后来的表
A	-
B	B
C	C
D	B
E	B
F	B

目标线路	C的表
A	A
B	A
C	-
D	D
E	E
F	E

目标线路	E的表
A	C
B	D
C	C
D	D
E	-
F	F

图 4-12　数据报子网络由

假设进程 P1 要向 P2 发送大量的数据（不可能放在一个数据报中）。进程将数据递交给传输层，并告知传输层相应的程序，将其发送到远端机器 H2 中的进程 P2 中。传输层程序在数据前加上一个传输头，并将其递交给网络层相应的程序。

假设数据的长度是最大数据包长度的 4 倍，网络层必须将数据划分成 4 个数据包，并编上序号，使用点到点的协议（如 PPP）将其依次发往路由器 A。此时，将由网络运营商接管余下工作。每个路由器都有一个路由表，记录着针对每个可能的地址，数据包发送的路径（下一跳路由器）。即，路由表的表项包含两个元素：目标地址，下一跳路由器地址（与路由表所在路由器直接相连的路由器）。路由器 A 的初始表如图 4-12 所示，对于任意一个目标地址，A 的路由表中下一跳路由器地址只能是 B、C 中的一个。

数据包 1、2、3 到达路由器 A 后被暂时保存下来，并按照 A 中的初始表查出下一跳路由器地址，将三个包依次发送给路由器 C。路由器 C 又将它们存储转发给路由器 E。路由器 E 接着将其存储转发给 F。当数据包到达 F 后将被封装到一个数据链路层的帧中，通过 LAN 发送给 H2。

然而，数据包 4 的发送有所不同。当它到达路由器 A 之后，A 上的路由表发生了变化，出于某种原因，A 将目标地址为路由器 F 的下一跳路由器地址设置为路由器 B。这可能是由于 A 发现往路由器 C 的路径上发生了流量拥塞，因此更新了路由表。如图 4-12 中所示的“后来的表”。

4．面向连接的服务

如果系统采用面向连接的服务，每次通信时都必须首先建立连接。连接建立后，从源机器到目标机器之间将形成一条路径，途经路由器将为该路径设置好一个固定的路由表，该路由表在连接未释放前不会修改。这种服务方式与电话系统的工作方式完全一致。我们通常称这个连接为一个虚电路，当连接被释放之后，虚电路也随之消失。采用面向连接的服务中，每个数据包都携带一个标识符，指明自己属于哪个虚电路。

图 4-13 描述了一个面向连接服务的例子。

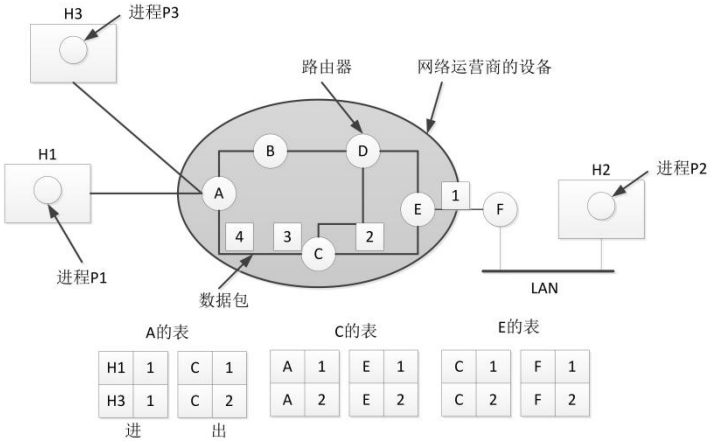


图 4-13　虚电路子网内的路由

主机 H1 与 H2 之间已经建立好了一个连接，这个连接的标识符为 1。路径中的路由器将相应的路由信息记录在其路由表的第 1 项。如果一个数据包来自主机 H1，连接标识符为 1，则将其存储转发给路由器 C，并赋予标识符为 1。同样，路由器 C 的路由表中第 1 项是该连接对应的路由信息，它指示下一跳路由器为 E，标识符仍为 1。

这时，主机 H3 想与 H2 建立连接，它也设置标识符为 1（因为这是主机 H3 发起的第一条连接），并告知子网它需要建立一条虚电路。它将路由信息记录在各个路由表的第 2 项中。路由器 A 第 2 项的输入是主机 H3，连接标识符为 1；假设输出仍为路由器 C，但是标识符必须要设置为 2，否则会产生冲突。避免冲突是路由器必备的功能之一，通常称为标签交换（label switching）。

四、传输层

传输层与其它层不同，它是整个协议架构的核心。

1．向上层提供的服务

传输层的最终目标是向它的用户（通常是应用层的进程）提供高效的、可靠的和性价比合理的服务。当然，为了达到这个目标，传输层必须利用网络层提供的服务。传输层中做这项工作的可以是硬件，也可以是软件程序，通常称为传输实体（transportentity）。传输实体可能被放在操作系统核心中，也可能属于一个用户进程，或者以一个二进制包绑定到网络应用程序中，或者放在一个可信的网卡上。网络层、传输层和应用层之间的关系如图 4-14 所示。

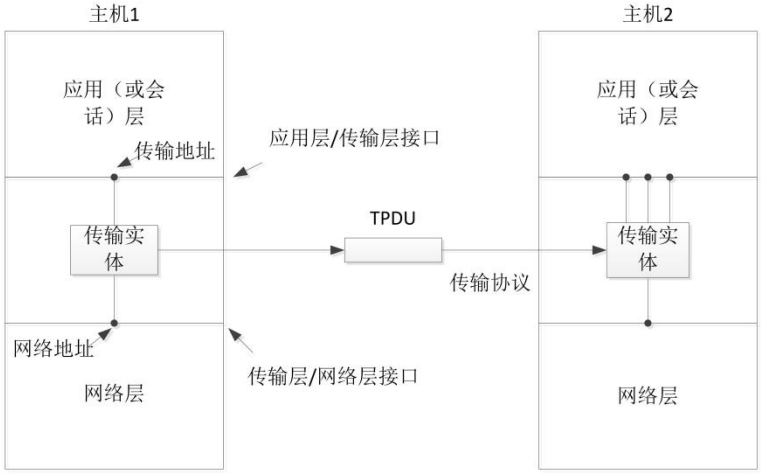


图 4-14　网络层、传输层和应用层

与网络层一样，传输层也有两类服务：面向连接的和无连接的服务。面向连接的传输层服务与面向连接的网络层服务在很多方面都相似。它们的连接都分为三个阶段：建立、传输数据、释放。地址编码和流量控制也很相似。另外，传输层中的无连接服务与网络层的无连接服务也比较相似。

随之而来的问题是，既然二者的服务如此相似，为何还分为两层呢？合为一层可以吗？

为了回答这个问题，可以先看图 4-15 所示内容。

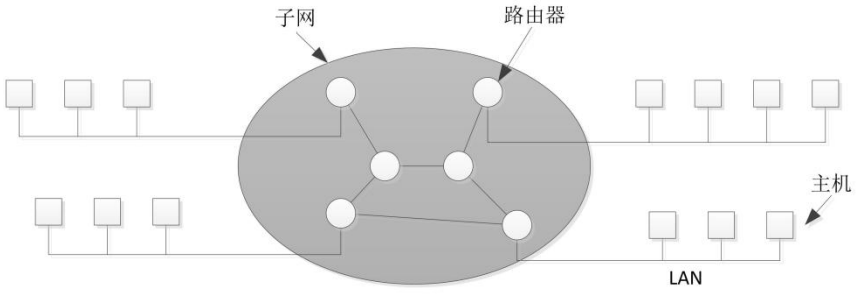


图 4-15　LAN 上的主机和子网之间的关系

传输层的代码只能运行在用户机器上，路由器中运行的最高层程序只能到网络层。如果网络层无法提供足够多的服务，此时子网中频繁丢包或者路由器时不时的崩溃，那么该怎么办呢？由于用户无法控制路由器的网络层，所以无法解决路由器的诸多问题。唯一的解决方法就是在网络层上再加一层来提高网络的服务质量，这就是网络层提供足够多的服务的原因。

如果网络层能够提供面向连接的服务，考虑这样一个例子，一个传输实体在数据传输过程中网络突然中断，连接因此被断开，并且没有任何关于最后一个成功传送的数据包是哪一

个的指示信息。当网络连接重新建立后，发送端可以使用这条新的网络连接向网络另一端发送查询请求，询问哪些数据包已经被接收，哪些还没有收到，根据这些信息就可以从上次中断的地方重新开始数据传输。

本质上，传输层可以比网络层提供更可靠的服务。丢失的数据包和毁坏的数据都可以在传输层上检测并重传。另外，传输层提供的服务可以利用网络层的服务原语来实现，而传输层提供的服务原语可以独立于网络层的服务原语。这样，更改网络层的服务原语并不会影响传输层的服务原语提供的服务功能，比如，无连接的 LAN 与面向连接的 WAN 的网络层服务原语完全不同，但是这上面的传输层服务原语提供的功能不会有任何变化。

正是因为有了传输层，应用程序编写者才可以根据标准编写程序代码，并运行在各种各样的网络中，而不用去处理不同的子网接口，也不同担心不可靠的传输方式。当然，如果所有的网络都是完美的、可靠的，传输层是可以去掉的。然而，现实情况并非如此，所以传输层在屏蔽子网的诸多问题上起到了关键的作用。

由于上面两方面的原因，传输层和网络层都是需要的。

基于传输层的特殊性，许多人习惯于将整个网络分成两个部分：第 1 层到第 4 层为一部分，其余为另一部分。从物理层到传输层可以被看成传输服务提供者（transport service provider），其余的层称为传输服务用户（transport service user）。

2．传输服务原语

传输层为应用程序提供了一个传输服务接口，用户可以利用这个接口来使用传输层提供的服务。每个传输层的服务都有自己的接口。

传输层的服务与网络层的服务比较相似，但二者有着重要的区别。二者最主要的区别是，网络层按照真实的网络情况建立服务模型。网络在现实情况中可能会丢包，所以通常是不可靠的。相比较而言，面向连接的传输层服务是可靠的。当然，真实的网络不是无错的，那么传输层的目标就是在不可靠的网络基础上提供可靠服务。

打个比方，对于 UNIX 系统上的两个进程，它们通过管道进行通信，通常认为通信过程是完美无缺的，进程本身不用考虑确认、丢包、拥塞等问题。它们希望的是 100％可靠的连接。进程 A 将数据放在管道的一端，进程 B 从另一端取出数据。面向连接的传输层服务就是这样的，它对进程屏蔽了很多网络层状况，并假定存在一条无错的比特流。

二者在服务目标上也存在区别。网络层只为传输实体提供服务。很少有人自己编写传输实体，并且很少有程序或用户直接使用网络层服务。相反，它们直接看到和用到的是传输层的服务原语。因此，传输层的服务必须方便易用。

表 4-1 给出了 5 个基本的服务原语。这是一个精简的传输层服务接口，描述了面向连接的传输服务接口的本质。它允许应用程序建立、使用和释放连接，对大部分应用程序来说已经足够使用。

表 4-1 一个简单传输服务的原语

原语	发送的分组	含义
LISTEN	（无）	阻塞，直到有某个进程试图与它建立连接
CONNECT	CONNECTION REQ.	主动地尝试建立一个连接
SEND	DATA	发送信息
RECEIVE	（无）	阻塞，直到一个数据包到来
DISCONNECT	DISCONNECTION REQ.	希望释放已经建立的连接

为了说明这些服务原语是如何工作的，考虑这样一个应用场景，它包含一个服务器端和若干个远端的客户端。

首先，服务器端执行一个监听（LISTEN）的服务原语，典型的做法是调用一个库程序，它将执行一个系统调用，进而阻塞该服务器程序，直到有客户端发来连接请求。客户端为了

和服务器端通信，它将执行 CONNECT 服务原语。传输实体将阻塞客户端并向服务器端发送数据包。该数据包内存放的数据属于传输层消息。

我们经常使用 TPDU（Transport Protocol Data Unit，传输协议数据单元）来表示传输实体之间传递的消息。所以，TPDUs 通常存放在网络层中传输的数据包中。同理，也存放在数据链路层中传递的数据帧中。当接收到一个数据帧时，数据链路层将对帧头进行处理，然后将内容（网络层的数据包，packets）递交给网络层。网络实体对数据包的头部进行处理，然后将内容递交给传输层。这种嵌套关系如图 4-16 所示：

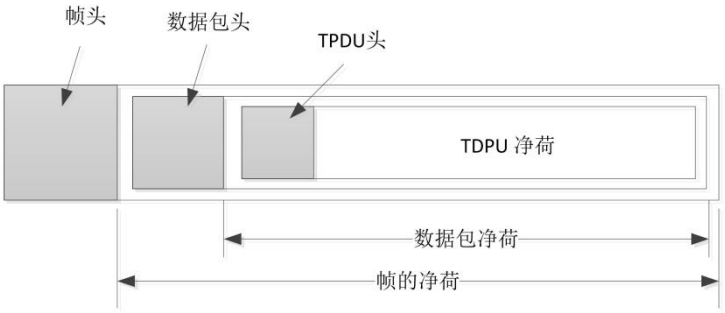


图 4-16　TPDU、数据包和帧的嵌套关系

回到刚才的客户－服务器通信的应用场景中，客户端执行的 CONNECT 服务原语将发送一个称为连接请求的 TPDU（CONNECTIONREQUESTTPDU），并将其发送给服务器端，客户端也进入阻塞状态。当连接请求到达服务器后，传输实体将检查服务器是否在监听（LISTEN），如果是，它将接触阻塞，并给客户端发送（回执）一个允许连接的数据单元（CONNECTIONACCEPTEDTPDU）。当客户端收到这个 TPDU 后，将解除客户端的阻塞。此时，客户－服务器连接已经建好。

数据的传送通常使用 SEND 和 RECEIVE 服务原语。任何一方都可以执行 RECEIVE 原语阻塞自己，等待另一方执行 SEND 原语发送数据。当接收方接收到一个 TPDU，它将解除自己的阻塞。它需要对这个 TPDU 进行处理，并回送一个应答信息。只要通信双方能够轮流的发送，工作就可以继续下去，不会发生冲突。

需要注意的是，传输层中即使再简单的传输，也要比网络层的通信要复杂的多。发送的每一个数据包都需要被确认。携带控制信息的 TPDU 的数据包也需要确认，可以是隐式的或者显示的。这些确认信息是由传输实体直接管理的，它利用了网络层协议，对传输层的用户来说是透明的。另外，传输实体还需要考虑定时器和重传的问题，但是对传输层的用户仍却是透明的。对传输层用户来说，一个连接就是一个可靠的管道，用户在一端放数据，另一端就一定会出现数据。这种具备隐藏底层复杂性的能力使得分层协议称为一种功能强大的工具。

连接最终必须被释放。释放连接有两种方式：对称性的和非对称性的。在非对称方式中，任何一方都可以提出断开连接（执行 DISCONNECT 原语），结果是本地机器发送一个 DISCONNECTTPDU 到远端的传输实体中。另一端接收到该 TPDU 后，即释放连接。

在对称方式中，每一个方向都需要单独的关闭———即都需要执行 DISCONNECT 原语。当一方执行了 DISCONNECT 原语，那么这一方就不能再发送数据了，但是可以接收数据。

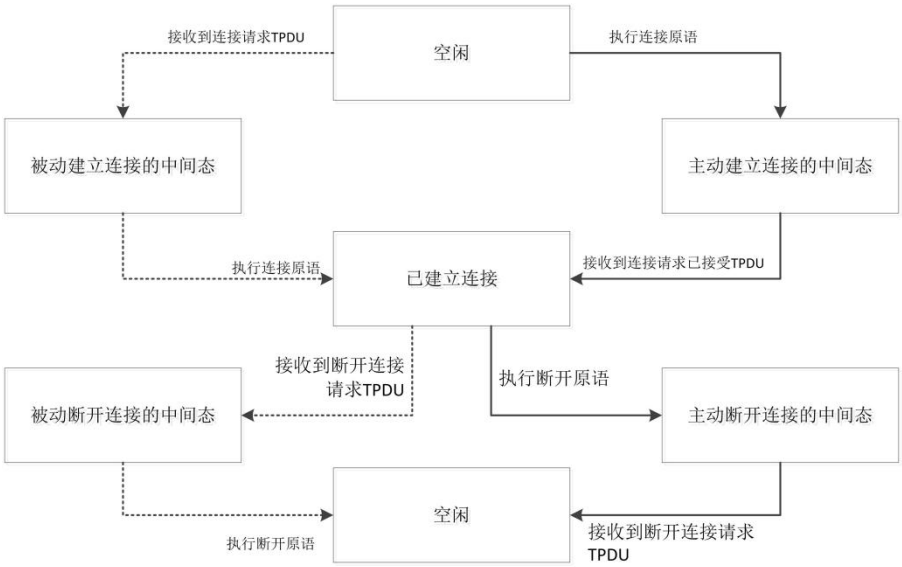


图 4-17　一个简单的连接管理方案

图 4-17 给出了一个传输层上的连接状态图。从图中可以看出，每一次状态的转换都是有一些事件所触发的，可能是由于传输层用户执行一个原语操作，或者是由于接收到网络上发送的数据包。为了简单起见，我们假定每个 TPDU 都需要单独的确认。

五、会话层

会话层是 OSI 模型中提出的，它在传输层提供的服务之上，给表示层提供服务，加强了会话管理、同步和活动管理等功能。

会话层的主要特点如下：

1．实现会话连接到传输连接的映射

会话层的主要功能是提供建立连接并有序传输数据的一种方法。这种连接就叫做会话（Session）。会话可以使一个远程终端登录到远地的计算机，进行文件传输或进行其它的应用。

会话连接建立的基础是建立传输连接。只有当传输连接建立好之后，会话连接才能依赖于它而建立。会话与传输层的连接有三种对应关系：

（1）一对一的关系，在会话层建立会话时，必须建立一个传输连接。当会话结束时，这个传输连接也释放了。

（2）多会话连接对单个传输连接。例如在航空订票系统中，为一个顾客订票则代理点终端与主计算机的订票数据库建立一个会话，订票结束则结束这一次会话。然后又有另一顾客要求订票，于是又建立另一个会话。但是，运载这些会话的传输连接没有必要不停地建立和释放。但多个会话不可同时使用一个传输连接。在同一时刻，一个传输连接只能对应一个会话连接。

（3）单会话连接对多个传输连接，这种情况是指传输连接在连接建立后中途失效了，这时会话层可以重新建立一个传输连接而不用废弃原有的会话。当新的传输连接建立后，原来的会话可以继续下去。

2．会话连接的释放

会话连接的释放不同于传输连接的释放，它采用有序释放方式，使用完全的握手，包括请求、指示、响应和确认原语，只有双方同意会话才终止。这种释放方式不会丢失数据：由于异常原因，会话层可以不经协商立即释放。但这样可能会丢失数据。

3．会话层管理

与其它各层一样，两个会话实体之间的交互活动都需协调、管理和控制的。会话服务的获得是执行会话层协议的结果，会话层协议支持并管理同等会话实体之间的数据交换。

由于会话往往是由一系列交互对话组成，所以对话的次序，对话的进展情况必须加以控制和管理。在会话层管理中考虑了令牌与对话管理、活动与对话单元以及同步与重新同步的措施。

（1）令牌和对话管理

在原理上，所有 OSI 的连接都是全双工的。然而，在许多情况下，高层软件为方便往往设计成半双工那样交互式通信。例如，远程终端访问一个数据库管理系统，往往是发出一个查询，然后等待回答，要么轮到用户发送，要么轮到数据库发送，保持这些轮换的轨迹并强制实行轮换，就叫做对话管理。实现对话管理的方法是使用数据令牌（data－token）。令牌是会话连接的一个属性，它表示了会话服务用户对某种服务的独占使用权，只有持有令牌的用户可以发送数据，另一方必须保持沉默。令牌可在某一时该动态地分配给一个会话服务用户，该用户用完后又可重新分配。所以，令牌是一种非共享的 OSI 资源。会话层中还定义了次同步令牌和主同步令牌，这两种用于同步机制的令牌将与下面的同步服务一起介绍。

（2）活动与对话单元

会话服务用户之间的合作可以划分为不同的逻辑单位，每一个逻辑单位称为一个活动（activity）。每个活动的内容具有相对的完整性和独立性。因此也可以将活动看成是为了保持应用进程之间的同步而对它们之间的数据传输进行结构化而引入的一个抽象概念。在任一时刻，一个会话连接只能为一个活动所使用，但允许某个活动跨越多个会话连接。另外，可以允许有多个活动顺序地使用一个会话连接，但在使用上不允许重叠。

例如：一对拨通的电话相当于一个会话连接，使用这对电话通话的用户进行的对话相当于活动。显然一个电话只能一个人使用，即支持一个活动。然而，当一对用户通完话后可不挂断电话，让后续需要同一电话线路连接的人接着使用，这就相当于一个会话连接供多个活动使用。若在通话过程中线路出现故障引起中断，则需要重新再接通电话继续对话，则就相当于一个活动跨越多个连接。

对话单元又是一个活动中数据的基本交换单元，通常代表逻辑上重要的工作部分。在活动中，存在一系列的交互通话，每个单向的连接通信动作所传输的数据就构成一个对话单元。

（3）同步与重新同步

会话层的另一个服务是同步。所谓同步就是使会话服务用户对会话的进展情况有一致的了解。在会话被中断后可以从中断处继续下去，而不必从头恢复会话。这种对会话进程的了解是通过设置同步点来获得的。会话层允许会话用户在传输的数据中自由设置同步点、并对每个同步点赋予同步序号，以识别和管理同步点。这些同步点是插在用户数据流中一起传输给对方的。当接收方通知发送方，它收到一个同步点。发送方就可确信接收方已将此同步点之前发送的数据全部接收完毕。会话层中定义了两类同步点：

◆主同步点：它用于在连续的数据流中划分出对话单元，一个主同步点是一个对话单元的结束和下一个对话单元的开始。只有持有主同步令牌的会话用户才能有权申请设置主同步点。

◆次同步点：次同步点用于在一个对话单元内部实现数据结构化，只有持有次同步点令牌的会话用户才有权申请设置次同步点。

主同步点与次同步点有一些不同。在重新同步时，只可能回到最近的主同步点。每一个插入数据流中的主同步点都被明确地确认。次同步点不被确认。

活动与同步点密切相关。当一个活动开始的时候，同步顺序号复位到 1 并设置一个主同步点。在一个活动内有可能设置另外的主同步点或次同步点。

（4）异常报告

会话层的另一个特点是报告非期待差错的通用机构。在会话期间报告来自下面网络的异常情况。

会话层可以向用户提供许多服务，为使两个会话服务用户在会话建立阶段，能协商所需的确切的服务，将服务分成若干个功能单元。

通用的功能单元包括：

◆核心功能单元：提供连接管理和全双工数据运输的基本功能。

◆协商释放功能单元：提供有次序的释放服务。

◆半双工功能单元：提供单向数据运输。

◆同步功能单元：在会话连接期间提供同步或重新同步。

◆活动管理功能单元：提供对话活动的识别、开始、结束，暂停和重新开始等管理功能。

◆异常报告功能单元：在会话连接期间提供异常情况报告。

上述所有功能的执行均有相应的用户服务原语。每一种原语类型都可能具有 request（请求）、indication（指示）、response（响应）和 confirm（确认）四种形式。然而，并非所有的组合都有效。

面向连接的 OSI 会话服务原语有 58 条，划分成 7 组：

◆连接建立。

◆连接释放。

◆数据运输。

◆令牌管理。

◆同步。

◆活动管理。

◆例外报告。

六、表示层

OSI 模型中，表示层以下的各层主要负责数据在网络中传输时不要出错。但数据的传输没有出错，并不代表数据所表示的信息不会出错。例如你想下午两点从杭州出发去上海，于是你对上海的朋友说，“我下午两点来”，可是你的朋友却理解为两点钟到达上海。所以这句话虽然没有听错，却因为不同的理解，产生了完成不同的结果。表示层就专门负责这些有关网络中计算机信息表示方式的问题。表示层负责在不同的数据格式之间进行转换操作，以实现不同计算机系统间的信息交换。

表示层（Presentation Layer）是 OSI 模型的第六层，它对来自应用层的命令和数据进行解释，对各种语法赋予相应的含义，并按照一定的格式传送给会话层。其主要功能是“处理用户信息的表示问题，如编码、数据格式转换和加密解密”等。表示层的具体功能如下：

数据格式处理：协商和建立数据交换的格式，解决各应用程序之间在数据格式表示上的差异。数据的编码：处理字符集和数字的转换。例如由于用户程序中的数据类型（整型或实型、有符号或无符号等）、用户标识等都可以有不同的表示方式，因此，在设备之间需要具有在不同字符集或格式之间转换的功能。

压缩和解压缩：为了减少数据的传输量，这一层还负责数据的压缩与恢复。数据的加密和解密：可以提高网络的安全性。

七、应用层

应用层（Application Layer）是 OSI 参考模型的最高层。它是计算机用户，以及各种应用程序和网络之间的接口，其功能是直接向用户提供服务，完成用户希望在网络上完成的各

种工作。它在其它 6 层工作的基础上，负责完成网络中应用程序与网络操作系统之间的联系，建立与结束使用者之间的联系，并完成网络用户提出的各种网络服务及应用所需的监督、管理和服务等各种协议。此外，该层还负责协调各个应用程序间的工作。

应用层为用户提供的服务和协议有：文件服务、目录服务、文件传输服务（FTP）、远程登录服务（Telnet）、电子邮件服务（E-mail）、打印服务、安全服务、网络管理服务、数据库服务等。上述的各种网络服务由该层的不同应用协议和程序完成，不同的网络操作系统之间在功能、界面、实现技术、对硬件的支持、安全可靠性以及具有的各种应用程序接口等各个方面的差异是很大的。应用层的主要功能如下：

用户接口：应用层是用户与网络，以及应用程序与网络间的直接接口，使得用户能够与网络进行交互式联系。

实现各种服务：该层具有的各种应用程序可以完成和实现用户请求的各种服务。

任务四　TCP/IP 体系结构

一、TCP/IP 参考模型的发展

实际上，TCP/IP 本身是一套协议栈。

ARPA 在 20 世纪 70 年代中期就开始研究互联网技术，到了 70 年代末期形成了基本的框架结构，与今天的形式大致相同。那时，ARPA 是分组交换网络研究的主要资助机构，ARPANET 已经初具规模。最初 ARPANET 主要租用传统的点到点通信线路进行互连。

陆续的，ARPA 开始研究在无线、卫星等通信线路上实施分组交换思想。事实上，网络硬件技术的多样性迫使 ARPA 必须去研究网络之间的互联互通问题。

ARPA 资助了很多研究机构去研究这些网际互联的问题，这些研究机构大多在前期参与过研制 ARPANET，具备了分组交换的思想。ARPA 定期的开会讨论研究者的思想和实验的研究成果，并成立了一个非正式的组织（group）———互联网研究组（Internet Research Group）。到了 1979 年，更多的研究者参与了进来，于是，ARPA 成立了一个非正式的委员会（committee）———互联网控制与配置委员会（Internet Control and Configuration Board，ICCB），该组织定期召开会议，直到 1983 年被重新组建。

全球性的互联网开始于 1980 年，从那时开始，ARPA 开始在网络内的机器上绑定新的 TCP/IP 协议。这时，ARPANET 变成了互联网的核心网络，它属于最早开展 TCP/IP 实验的网络。到了 1983 年 1 月，国防部长办公室要求所有连接到网络中的计算机必须使用 TCP/IP 协议。同时，国防通信局（Defense Communication Agency）将 ARPANET 拆分成两个独立的网络，一个用来研究，一个用来军事通信。用来研究的那部分网络仍然沿用 AR-PANET 的名字，做为军事通信部分的网络便成了军用网络，改名为 MILNET（militarynet-work）。

为了鼓励大学研究者采用新的网络协议，ARPA 将使用费用降到了最低。那时，美国许多大学的计算机系都在使用加利福尼亚大学的伯克利软件发布机构研制的 UNIX 操作系统———通常称为 Berkeley UNIX 或 BSDUNIX。ARPA 资助 BBN 有限公司在 UNIX 操作系统上实现了 TCP/IP 协议，接着通过给 Berkeley 投资的方式，使其在 UNIX 发布版中集成了 TCP/IP 协议。这样一来，使用 TCP/IP 协议进行通信的大学达到了 90%。恰在此时，许多大学都开始采购第二台、第三台计算机，急需将这些计算机互联成局域网，通信协议软件成了必须品。

BSDUNIX 也因为集成了 TCP/IP 通信协议而变得更加实用，从而名气大增，促进了这种操作系统的流行。除了标准的 TCP/IP 通信协议软件，Berkeley 还提供了一套网络服务工具，类似于传统的 UNIX 单机服务。Berkeley 的优势在于它与标准 UNIX 非常相似。比如，有一定经验的 UNIX 用户很快能够学会使用 Berkeley 的远程文件复制工具（remote file copy，

rcp），因为它的操作方式与 UNIX 中的文件复制工具几乎相同，除了它能够将文件复制到远程机器。

BSDUNIX 除了提供大量的网络通信工具外，它还提供了一套编程接口———socket，能够允许程序开发人员访问通信协议。另外，套接字接口除了可以使用 TCP/IP 协议之外，还可以选用其他网络协议。套接字这种设计从使用开始，就引起了很多争论，很多操作系统设计者都提出了各自的设计方案。但是由于它不仅有着整体性的优点，而且它允许程序员花费较少的代价就能掌握套接字的开发方法。因此，它激发了更多的研究者参与到了 TCP/IP 的使用中。

TCP/IP 技术的成功和互联网技术的研究吸引了更多的人开始使用它们。意识到网络通信将称为未来科学研究的重点，国家科学基金会（National Science Foundation，NSF）促进了 TCP/IP 互联网的发展。20 世纪 70 年代末，NSF 成立了 CSNET（Computer Science NET work），它的目的是将所有计算机科学家用网络连接起来。从 1985 年开始，它规划使用 6 个超级计算机作为核心形成一个计算机网络。1986 年，NSF 扩展了该网络，资助了一个新的广域主干网，被叫做 NSFNET，它能够与所有的大型计算机通信，并将其连到 AR-PANET。最终，在 1986 年，NSF 为各个地区的网络提供种子资金，使得他们可将该地区的主要研究机构连到互联网上。所有 NSF 赞助的网络都采用 TCP/IP 协议，都变成了互联网的一部分。

TCP/IP 互联网协议族不是从特定厂家或被公认的专家组中产生出来的，它是由 Internet 结构委员会（Internet Architecture Board，IAB）协调并制订的。

1983 年，ARPA 组建了 IAB，它的最初目标是鼓励参与了 TCP/IP 和 Internet 研究的核心人员交流思想，然后促使人们集中于一个共同的目标。在前 6 年中，IAB 从一个由 ARPA 指定的研究组织演化成为一个自治的团体。在这些年里，IAB 的每个成员都是一个 Internet 任务组（Internet Task Force）的主持者，分管研究某个重要课题。IAB 大约由 10 个任务组组成，它们章程的变化范围很大，从研究不同应用程序引入的通信量负载如何影响 Internet，到研究如何处理短期 Internet 工程问题。IAB 每年开几次会议，听取每个任务组的态势报告、评审和修改技术方向、讨论政策、与 ARPA 和 NSF 之类的资助 Internet 运行及研究的机构代表交换信息等。

IAB 的主席有“Internet 设计师（Internet Architect）”的头衔，他负有建议技术方向和协调各任务组活动的责任。在 IAB 的建议下，IAB 主席建立新的任务组，对其他机构来说，他也代表 IAB。

经过最初的 7 年，互联网已经发展了遍布美国和欧洲的数百个区域性网络。它连接了将近 20000 台计算机，这些计算机遍布了大学、政府机构和科研实验室。互联网的大小和范围一直在以惊人的速度在增长。到了 1987 年底，它每个月的增长速度约为 15％。到了 2000 年，互联网已经连接了全球 209 个国家，超过 50 万台计算机。

互联网的发展已经不再局限于政府投资的项目。很多大公司也像计算机公司那样连到了互联网，如，石油公司、汽车工业、电子工厂、制药公司、电信公司等等。1990 年开始，中小规模的公司也开始接入互联网。虽然有些公司内部没有接入互联网，但是公司的局域网采用的仍然是 TCP/IP 协议。

互联网快速的扩张也带来了许多设计之初没有考虑到的问题，科学家们开始研究如何管理规模巨大的分布式的网络资源。在最初的设计中，计算机在网络中使用的名字和地址是一个手工编写的文件，发布到互联网的每个站点上。到了 20 世纪 80 年代中期，数据存放在一个中心数据库显然不够了。首先，越来越多的计算机要连到互联网上，工作人员已经无法满足更新文件的需要。其次，即使中心文件能够及时更新并且准确，网络也没有足够的流量来满足遍布各地的站点来访问这个文件。

为了解决这些问题，科研人员开发了新的协议，并设计了命名系统，使得用户能够自动

解析互联网上任意一台机器的名字。这就是著名的域名系统（Domain Name System，DNS），它主要由域名服务器组成，域名服务器负责回答关于名字的问题。名字不再需要集成存放在一台计算机上，它们被分布各个区域中，可以使用 TCP/IP 协议向它们发送查询信息。

二、TCP/IP 的体系结构

参照 ISO 参考模型，TCP/IP 体系结构可以划分为 5 个概念层次———4 个软件层和一个硬件层。如图 4-18 所示。

图 4-18　TCP/IP 概念层次

图 4-19 描述了 TCP/IP 模型中早期的协议和网络。TCP/IP 中通常不涉及硬件层内容，其它四个软件层的概念如下所述。

1．应用层

应用层处在最高层，用户调用应用层软件来访问 TCP/IP 互联网提供的服务。应用层软件利用传输层接口发送和接收数据。每个应用层软件都可以选择适合自己的传输服务类型———独立的报文序列和连续字节流两种类型。应用程序将数据递交给传输层。

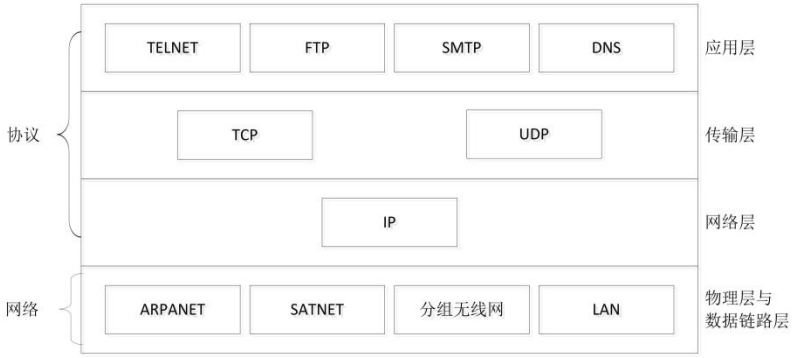


图 4-19　TCP/IP 模型中早期的协议和网络

TCP/IP 模型没有会话层和表示层。因为 TCP/IP 协议设计者认为不需要这两层，所以设计中没有包含这两层。从 OSI 模型的经验来看，这么设计是正确的，因为对大多应用程序来说是没有任何用处的。

应用层在传输层之上。它包含很多高层协议。比如，早期出现的有虚拟终端———TEL-NET，文件传输协议———FTP，电子邮箱协议———SMTP。虚拟终端协议允许本地

机器上的用户登录到远程机器上并操作其程序。文件传输协议提供了一种高效的方式，可将文件数据从一台机器传递到另一台机器上。起初，电子邮箱也是采用文件传输的方式来实现，后来，为收发电子邮件专门开发一种协议———SMTP。经过这么多年的发展，应用层协议变得丰富多彩。

2．传输层

传输层主要提供端到端的通信服务。传输层必须可以管控数据流，能够提供可靠的连接服务，保证数据没有错误并且按序到达。为了满足这种服务要求，传输层设计了应答和重传机制。传输层软件将数据流分解为多片（通常称为数据包），然后在每个数据包前加上目的地址后递交给下一层协议。

图 4-18 画的应用层只有一个方框，但实际上，一个通用的计算机可能会在同一时间有很多应用程序访问互联网。所以传输层必须能够接收多个应用层程序递交的数据，处理后递交给下一层。为此，它为每个数据包增加一些额外的信息和代码，以便能够区分不同的应用层软件，同时还要加上校验码。接收端根据校验和保证数据的完整性和正确性，根据目的地信息确认该交给哪个应用层软件。

TCP/IP 的传输层与 OSI 的传输层相似。它设计了两种端到端的传输协议。第一个是 TCP（Transmission Control Protocol，传输控制协议），它是一种可靠的面向连接的协议，它允许互联网中一台机器将字节流无错的传送给另一台机器。TCP 需要进行流量控制，确保高速的发送端不会淹没慢速的接收端。

传输层上另一个协议是 UDP（User Datagram Protocol，UDP），它是一种不可靠的、无连接的协议，主要用在那些自身能够提供 TCP 的各种功能（比如，流量控制、序列化）的应用程序。它通常用于那些只需要发送一次信息的应用场合，和那些及时交付要比准备交付更重要的场合，比如视频和语音传输。

3．网际层

如前所述，网际层负责网络通信问题。该层是 TCP/IP 体系结构中至关重要的一层。

网际层定义了一种数据包交换协议———IP（Internet Protocol，网际协议）。它负责接收传输层的请求，传输层将数据包递交给网际层，由网际层负责发送。它将数据包封装到 IP 数据报（datagram）中，填写数据包首部，使用选路算法来决定是直接交付数据报还是把它发送给路由器，接着将数据包发送给相应的网卡进行传输。网际层也需要处理到来的数据报，负责检查其有效性，然后使用选路算法来决定是在本地处理还是继续转发。如果目标就是本机，则需本地处理，网际层软件删除数据报首部，递交给相应的传输层软件来处理该数据包。最后，网际层还需要有能力发送和接收互联网控制报文协议（ICMP）的差错和控制报文。

网际层主要负责处理分组路由和拥塞避免等问题，它与 OSI 模型中的网络层类似。

4．网络接口层

这是 TCP/IP 软件层次的最底层。该层负责接收 IP 数据报，并将其发给指定的网络。一个网络接口可能包含一个设备驱动（比如，计算机与局域网相连时所需的网卡驱动），也可能包含一个复杂的使用自己的数据链路协议的子系统（比如，由使用高级链路控制（high level data link control，HDLC）协议的主机构成的分组交换网络）。

事实上，TCP/IP 中并没有明确规定网络接口层应该包含哪些功能，它只是规定主机必须通过某种协议连接到网络上，以便可以将 IP 数据包发送到网络中。关于 TCP/IP 体系结构的书很少描述这一层内容。

任务五　OSI 参考模型与 TCP/IP 参考模型的比较

OSI 和 TCP/IP 参考模型有很多共同点。两者都以协议栈的概念为基础，并且协议栈中

的协议彼此相互独立。而且，两个模型中各个层的功能也大体相似。例如，在两个模型中，传输层以及传输层以上的各层都为希望进行通信的进程提供了一种端到端的、与网络无关的服务。这些层形成了传输提供方。另外，在这两个模型中，传输层之上的各层也都是传输服务的用户，并且是面向应用的用户。当然，两个模型之间也有很多不同的地方。

一、对 OSI 参考模型的评价

OSI 模型，它的核心在于 3 个概念：服务、接口和协议。

OSI 模型最大的贡献是使这 3 个概念之间的区别变得清晰，明了。每一层都为它的上一层提供一些服务。服务的定义指明该层做些什么，而不是上一层的实体如何访问这一层，或这一层是如何工作的。它定义了这一层的语义每一层的接口告诉它上面的进程应该如何访问本层。它规定有哪些参数以及结果是什么。但是它并未说明本层内部是如何工作的。

最后，每一层所用的对等协议是本层内部的事情。它可以使用任何协议，只要它能够完成任务就行（也就是说提供所承诺的服务）。它也可以随意改变协议，而不会影响它上面的各层。

这些思想与现代的面向对象程序设计思想非常吻合，一个对象就如同一个层次，它有一组方法（或称操作），对象之外的过程可以调用这些方法。这些方法的语义规定该对象所提供的服务的集合。方法的参数和结果构成对象的接口。对象的内部代码是它的协议，对于外部而言是不可见的，也不需要被外界关心。

TCP/IP 最初并未明确的区分服务、接口和协议三者之间的差异，但是在他成熟之后，人们人已经努力对它做了改进，以便更加接近于 OSI 模型。例如，网际层提供的服务其实只有发送 IP 分组和接收 IP 分组。

因此，OSI 模型中的协议比 TCP/IP 模型中的协议有更好的隐蔽性，当技术发生变化的时候 OSI 中的协议相对更容易被替换为新的协议。最初采用的分层协议的主要目的之一就是能够做这样的替换。

OSI 参考模型是在协议发明之前就已经产生的。这种顺序关系意味着 OSI 模型不会偏向于任何一种特定的协议，因而该模型更加具有通用性。这种做法存在的缺点是，设计者有太多的经验可以参考，因此不知道那些功能应该放在哪一层上。例如，数据链路层最初只处理点到点网络。当广播式网络出现后，必须在模型中嵌入一个新的子层，当人们使用 OSI 模型和已有的协议来建立实际网络时，才发现这些网络并不能很好地匹配所要求的服务规范，这使得当初的设计者感到万分惊讶！因此不得不在模型中加入一些子层，以便可以提供足够的空间来弥补这些差异。还有，标准委员会最初期望每一个国家都有一个由政府运行的网络并使用 OSI 协议，所以根本不考虑网络互连的问题。然而事情并非像预期的那样。

OSI 参考模型描述了信息或数据通过网络，是如何从一台计算机的一个应用程序到达网络中另一台计算机的另一个应用程序的。当信息在 OSI 参考模型内逐层传送的时候，它越来越不像人类的语言，变为只有计算机才能明白的数字（0 或 1）。

在 OSI 参考模型中，计算机之间传送信息的问题分为 7 个较小且更容易管理和解决的小问题。每一个小问题都由模型中的一层来解决。之所以划分为 7 个小问题，是因为它们中的任何一个都囊括了问题本身，不需要太多的额外信息就能很容易地解决。

OSI 参考模型的主要问题是定义复杂、实现困难，有些同样的功能（如流量控制与差错控制等）在多层重复出现，效率低下等。而 TCP/IP 体系结构的缺陷包括网络接口层本身并不是实际的一层，每层的功能定义与其实现方法没能区分开来，使 TCP/IP 体系结构不适合于其他非 TCP/IP 协议族等。

人们普遍希望网络标准化，但 OSI 迟迟没有成熟的网络产品。因此，OSI 参考模型与协议没有像专家们所预想的那样风靡世界。

二、对 TCP/IP 参考模型的评价

TCP/IP 参考模型是因特网（Internet）的基础。和 OSI 的 7 层协议比较，TCP/IP 参考模型中没有会话层和表示层。通常说的 TCP/IP 是一组协议的总称，TCP/IP 实际上是一个协议族（或协议包），包括 100 多个相互关联的协议，其中 IP（Internet Protocol，网际协议）是网络层最主要的协议；TCP（Transmission Control Protocol，传输控制协议）和 UDP（User Datagram Protocol，用户数据报协议）是传输层中最主要的协议。一般认为 IP、TCP、UDP 是最根本的三种协议，是其它协议的基础。

TCP/IP 也是使用协议栈来工作，栈是所有用来在两台机器间完成一个传输的所有协议的几个集合。数据通过栈，从一台机器到另一台机器，在这过程中，一个复杂的查错系统会在起始机器和目的机器中执行。栈分成五个层，每一层都能从相邻的层中接收或发送数据，每一层都与许多协议相联系。

OSI 和 TCP/IP 参考模型有很多共同点。两者都以协议栈的概念为基础，并且协议栈中的协议相互独立。同时，两个模型中的各个层次的功能也大体相似。例如，在两个模型中，传输层以及传输层以上的各层都为进行通信的进程提供一种端到端的、与网络无关的传输服务，这些层形成了传输提供方。另外，在这两个模型中，传输层之上的各层都是传输服务的用户，并且都是面向应用的用户。

TCP/IP 与 OSI 不同，它是先出现协议，TCP/IP 模型只是这些已有协议的一个描述而已。所以协议一定会符合模型，这肯定没有问题。而且两者确实吻合得很好，唯一的问题在于，TCP/IP 模型不适合其他的任何协议栈，因此，想要描述其他非 TCP/IP 网络，该模型并不是很有用。

TCP/IP 一开始就对面向连接服务和无连接服务并重，而 OSI 在开始时只强调面向连接这一种服务。

TCP/IP 较早就有较好的网络管理功能，而 OSI 到后来才开始考虑这个问题。

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在大范围获得工程上的应用。但现在情况却反过来了，得到最广泛应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。

尽管 TCP/IP 体系结构与 OSI 参考模型在层次划分及使用的协议上有很大区别，但它们在设计中都采用了层次结构的思想。无论是 OSI 参考模型还是 TCP/IP 体系结构都不是完美的，对二者的评论与批评都很多。但是 TCP/IP 体系结构与协议在 Internet 中经受了几十年的风风雨雨，得到了 IBM、Microsoft、Novell 及 Oracle 等大型网络公司的支持，成为计算机网络中的主要标准体系。

项目实训

实训任务　在 Windows10 下配置和管理 TCP/IP

【实训目标】

1.了解网络基本配置中包含的协议、服务、客户端。

2.了解 Windows 支持的网络协议及参数设置方法。

3.掌握 TCP/IP 的配置。

【实训环境】

硬件环境：每人 1 台计算机，能够接入 Internet 的局域网络。

【实训内容】

公司给新来的领导配备了一台新电脑，需要配置相应的 IP 地址才能上网，领导让你过去帮忙配置一下。配置过程中，查看所在机器的主机名称和网络属性，了解网络基本配置中

包含的协议、服务、客户端等信息，完成 TCP/IP 的参数配置，实现使用新电脑能正常上网。

【实训步骤】

步骤 1

打开“开始”→“设置”→“网络和 Internet”→“以太网”→“网络和共享中心”，如图 4-20 所示。也可以通过 Windows10 桌面右下角通知栏里的网络图标或右键单击桌面“网.上邻居”，打开“网络和共享中心”。

图 4-20　“网络和共享中心”窗口

步骤 2

在“网络和共享中心”窗口中单击“以太网”，打开“以太网状态”窗口，如图 4-21 所示。单击“属性”按钮，打开如图 4-22 所示“以太网属性”窗口。通过“以太网属性”窗口，了解网络基本配置中包含的协议、服务、客户端等信息。

图 4-21　“以太网状态”窗口

图 4-22　“以太网属性”窗口

步骤 3

在“以太网属性”窗口中，选择“Internet 协议版本 4（ TCP/IPv4）”，然后单击“属性”按钮，打开“Internet 协议版本 4（ TCP/IPv4）属性”窗口。选中“使用下面的 IP 地址（S）”及“使用下面的 DNS 服务器地址（E）”单选按钮，在 IP 地址栏里输入管理员分配的 IP 地址及相关参数，单击“确定”按钮即可，如图 4-23 所示。

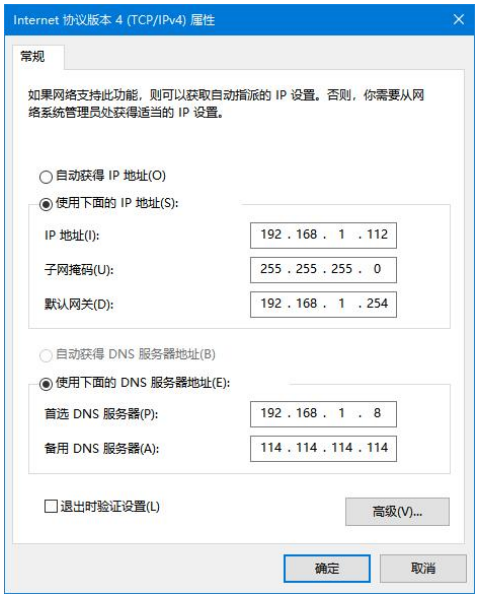


图 4-23　“Internet 协议版本 4（ TCP/IPv4 ）属性”窗口

项目小结

在本项目，我们从网络的分层体系结构出发，主要学习了 TCP/IP 基础，比较了 OSI 参考模型与 TCP/IP 参考模型；介绍了网络接口层、网络层、传输层及应用层的相关功能及对应的网络协议；并重点学习了 IP 地址规划与子网划分。

通过项目实训，读者应该对 TCP/IP 基础有一定的了解，能够充分理解 TCP/IP 各层的基本原理和常见应用，理解网络协议的工作原理。

思考与练习

一、填空题

1．网络协议为在计算机网络中实现数据交换必须遵循的事先约定好的_____、_____或_____。网络协议主要由下面三个要素组成：_____、_____、_____。

2._____的作用主要是将原始传输设备上没有被发现的错误呈现给网络层。_____负责控制整个子网的运转。_____的作用是将上一层传下来的数据拆分成更小的单元传给网络层，最终保证网络另一端的计算机的传输层能够正确接收数据。

3．物理层的主要功能是确定与传输媒介的接口的一些特性，即：_____、_____、_____、_____。

4．每一种原语类型都可能具有_____、_____、_____和_____四种形式。

5．OSI 模型，它的核心在于 3 个概念：_____、_____和_____。

二、简答题

1．计算机网络协议采用层次结构有何好处？

2．简述协议、层次、接口的概念？

3．简述 OSI 参考模型的主要特征，体系结构分几层，各层有什么作用？

4．简述网络层的主要功能？