第2章

计算机网络参考模型

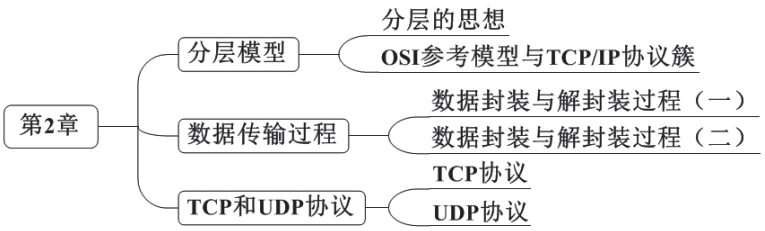
技能目标

- 掌握 OSI 和 TCP/IP 分层模型的结构
- 理解 OSI 各层功能
- 掌握数据传输过程
- 理解 TCP 和 UDP 协议

本章导读

本章将学习网络参考模型，它是理解网络这个全新世界的关键所在。本章的主要内容有三部分：各层的名称、功能，数据在各层之间的传输过程，TCP/IP 协议簇。TCP/IP 协议簇的传输层有两个重要的协议：TCP 协议和 UDP 协议，本章将详细介绍它们的首部格式、TCP 连接建立与终止的过程。

知识服务



2.1 分层模型

我们对现实世界的认识往往只是冰山一角，大部分的“真相”都掩藏在海平面以下，网络世界更是如此。平时在家里访问各种网页或者聊 QQ 时，我们的操作无外乎点击图标，打几个字而已，但对于计算机和网络中转设备来说，却是一个相当复杂的过程。就好像邮寄一份礼物给远方的朋友，我们需要做的只是将这份礼物交给邮局并写明正确的地址，如果不出意外，这位朋友将会顺利收到，但是这份礼物在中间经历了哪些复杂的过程，传递礼物的双方就不得而知了。对于网络的最终用户，了解到这个层次已经足够了，但如果想成为一名网络技术人员，就必须对这个过程了如指掌，这样才能分析排查网络的常见故障。

2.1.1 分层的思想

下面将开始研究网络传输的真正过程，这个过程非常复杂，因此应首先建立分层模型的概念。分层模型是一种用于开发网络协议的设计方法。而分层思想本质上讲就是把节点间通信这个复杂问题分成若干相对简单的问题逐一解决每个问题对应一层。每一层实现一定的功能，相互协作即可实现数据通信这个复杂任务。

让我们试想一下，早上时间比较紧张的时候，冲一杯牛奶是一种不错的早餐方案。作为最终用户，我们并没有感受到喝一杯奶有多难，因为我们只是把奶粉从超市买回家，用水冲开而已。但奶粉的生产者面临一系列复杂的问题，如何选择物美价廉的奶源，如何将牛奶运送到奶粉厂且保证牛奶不变质，如何安排奶粉的整个生产工艺（包括检验），如何包装才能更吸引客户，如何与各大超市洽谈，如何与物流公司沟通，等等。作为一名奶粉厂的管理者，应该如何应对这么复杂的事情呢？最好的方法就是用分层的思想，将整个生产销售流程分成几个不同的管理模块，每个模块由专门的负责人管理协调。于是奶粉厂就会出现各个部门：原料采购部、奶源加工车间、奶粉生产车间、奶粉包装车间、销售部门等。如图 2.1 所示。

这样奶粉加工生产的整个过程就变得很清晰了，更重要的是，如果出现各种问题，如奶粉质量问题等，管理者可以很快确定问题的原因，从而针对性地解决问题。这些部门有着各自相对独立的职责，彼此又是相关联的，处于流程前端的部门为后续部门服务，后续部门也需要在前端部门的基础上实现其功能。例如，原料采购部门为奶源

加工车间服务，因为只有优质的奶源才能保证加工的半成品的质量，奶源加工车间的工作又是在原料采购部的基础上完成的。一旦在最后的成品中发现细菌超标，可以很容易确定是奶源加工车间出了问题。

部门	职责
原料采购部	选购优质奶源，与农场签订合同，保质保量运输奶源
奶源加工车间	原料验收，杀菌处理，储藏
奶粉生产车间	浓缩、喷雾干燥、冷却筛粉
奶粉包装车间	奶粉包装、奶粉装箱，质检
销售部门	联系各大销售渠道，联系物流运输

图 2.1　奶粉厂生产流程

现在让我们从现实世界回到网络世界，网络节点间通信也体现了这种思想。赋予每一层一定的功能，相邻层之间通过接口来通信，下层为上层提供服务。一旦网络发生故障，很容易确定问题是由哪一层的功能没有实现而导致的，将故障产生的原因聚焦于一点，有助于更加清晰明了地分析问题、解决问题。另外，对于还处于学习阶段的我们，将网络最终的通信目标分解成各个子层的目标，然后逐一研究每一层的功能是如何实现的，这种思想有助于将复杂问题简单化、清晰化。

2.1.2　OSI 参考模型与 TCP/IP 协议簇

1. OSI 参考模型

由之前的例子，应该可以理解分层模型对于网络管理而言就像是企业组织架构对于企业管理一样具有至关重要的地位。由于各个计算机厂商都采用私有的网络模型，因此给通信带来诸多麻烦，国际标准化组织（International Standard Organization，ISO）于 1984 年颁布了开放系统互联（Open System Interconnection，OSI）参考模型。OSI 参考模型是一个开放式体系结构，它规定将网络分为七层，从下往上依次是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层，如图 2.2 所示。

分层	功能
应用层	网络服务与最终用户的一个接口
表示层	数据的表示、安全、压缩
会话层	建立、管理、中止会话
传输层	定义传输数据的协议端口号，以及流量控制和差错校验
网络层	进行逻辑地址寻址，实现不同网络之间的路径选择
数据链路层	建立逻辑连接、进行硬件地址寻址、差错校验等功能
物理层	建立、维护、断开物理连接

图 2.2　OSI 七层模型

（1）物理层

物理层（Physical Layer）的主要功能是完成相邻节点之间原始比特流的传输。

物理层协议关心的典型问题是使用什么样的物理信号来表示数据 1 和 0，一位数据持续的时间有多长，数据传输是否可以同时在两个方向上进行，最初的连接如何建立以及完成通信后连接如何终止，物理接口（插头和插座）有多少针以及各针的用处。物理层的设计主要涉及物理层接口的机械、电气、功能和过程特性，以及物理层接口连接的传输介质等问题。物理层的设计还涉及通信工程领域内的一些问题。

（2）数据链路层

数据链路层（Data Link Layer）负责将上层数据封装成固定格式的帧，在数据帧内封装发送端和接收端的数据链路层地址（在以太网中为 MAC 地址，MAC 地址是用来标识网卡的物理地址；在广域网中点到多点的连接情况下，可以是一个链路的标识），并且为了防止在数据传输过程中产生误码，要在帧尾部加上校验信息，发现数据错误时，可以重传数据帧。

（3）网络层

网络层（Network Layer）的主要功能是实现数据从源端到目的端的传输。在网络层，使用逻辑地址来标识一个点，将上层数据封装成数据包，在数据包的头部封装源和目的端的逻辑地址。网络层根据数据包头部的逻辑地址选择最佳的路径，将数据送达目的端。

（4）传输层

传输层（Transport Layer）的主要功能是实现网络中不同主机上用户进程之间的数据通信。

网络层和数据链路层负责将数据送达目的端主机，而这个数据需要什么用户进程去处理，就需要传输层帮忙了。

例如，用 QQ 发送消息，网络层和数据链路层负责将消息转发到接收人的主机，而接收人应该用 QQ 程序来接收还是用 IE 浏览器来接收，就是在传输层进行标识。

传输层要决定对会话层用户（最终的网络用户）提供什么样的服务。因此，我们经常把 1 ～ 3 层的协议称为点到点的协议，而把 4 ～ 7 层的协议叫作端到端的协议。

由于绝大多数主机都支持多进程操作，机器上会同时有多个程序访问网络，这就意味着将有多条连接进出这台主机，需要以某种方式区别报文属于哪条连接。识别这些连接的信息可以放在传输层的报文头中。除了将几个报文流多路复用到一条通道上，传输层还必须管理跨网连接的建立与拆除。这就需要某种命名机制，使机器内的进程能够说明其希望交谈的对象。

（5）会话层

会话层（Session Layer）允许不同机器上的用户之间建立会话关系。会话层允许进行类似传输层的普通数据的传送，在某些场合还提供了一些有用的增强型服务；也允许用户利用一次会话在远端的分时系统上登录，或者在两台机器间传递文件。

会话层提供的服务之一是进行会话控制。会话层允许信息同时双向传输，或任

意一个时刻只能单向传输。如果属于后者，则类似于物理信道上的半双工模式，会话层将记录此时该轮到哪一方。一种与对话控制有关的服务是令牌管理（Token Management）。有些协议会保证双方不能同时进行同样的操作，这一点很重要。为管理这些活动，会话层提供了令牌，令牌可以在会话双方之间移动，只有持有令牌的一方可以执行某种关键性操作。另一种会话层服务是提供同步。如果在平均每小时出现一次大故障的网络上，两台机器间要进行一次两小时的文件传输，那么在每一次传输中途失败后，都不得不重新传送这个文件。为解决这个问题，会话层提供了一种方法，即在数据中插入同步点。当每次网络出现故障后，仅需重传最后一个同步点以后的数据。

（6）表示层

表示层（Presentation Layer）用于完成某些特定功能，对这些功能人们常常希望找到普遍的解决方法，而不必由每个用户自己来实现。值得一提的是，表示层以下各层只关心从源端机到目标机可靠地传送比特，而表示层关心的是所传送信息的语法和语义。表示层服务的一个典型例子是用大家一致选定的一种标准方法对数据进行编码。大多数用户程序之间并非交换随机比特，而是交换诸如人名、日期、货币数量和发票之类的信息。这些对象是采用字符串、整型数、浮点数的形式，以及由几种简单类型组成的数据结构来表示的。

在网络上，计算机可能采用不同的数据表示法，所以在数据传输时需要进行数据格式转换。例如，在不同的机器上常用不同的代码来表示字符串（ASCII 和 EBCDIC）、整型数（二进制反码或补码）以及机器字的不同字节顺序等。为了让采用不同数据表示法的计算机之间能够相互通信并交换数据，我们在通信过程中使用抽象的数据结构（如抽象语法表示 ASN.1）来表示所传送的数据，而在机器内部仍然采用各自的标准编码。管理这些抽象数据结构，并在发送方将机器的内部编码转换为适合网上传输的传送语法以及在接收方做相反的转换等工作都是由表示层来完成的。另外，表示层还涉及数据压缩和解压、数据加密和解密等工作。

（7）应用层

应用层（Application Layer）包含大量人们普遍需要的协议。显然，对于需要通信的不同应用来说，应用层的协议都是必需的。例如，PC 用户操作仿真终端软件通过网络使用远程主机的资源。这个仿真终端软件使用虚拟终端协议，将键盘输入的数据传送到主机的操作系统并接收显示于屏幕的数据。又如，当用户想要获得远程计算机上的一个文件副本时，他要向本机的文件传输软件发出请求，这个软件与远程计算机上的文件传输进程通过文件传输协议进行通信，协议主要处理文件名、用户许可状态和其他请求细节的通信。远程计算机上的文件传输进程则使用其他进程则来传输文件内容。

由于每个应用有不同的要求，因此应用层的协议集在 OSI 模型中并没有定义。但是，有些确定的应用层协议，包括虚拟终端、文件传输和电子邮件等都可作为标准化的候选。

2. TCP/IP 参考模型

另外一个著名的模型是 TCP/IP 模型。TCP/IP 是传输控制协议 / 网络互联协议（Transmission Control Protocol/Internet Protocol）的简称。早期的 TCP/IP 模型是一个四层结构，从下往上依次是网络接口层、互联网层、传输层和应用层。在后来的使用过程中，借鉴 OSI 的七层参考模型，又将网络接口层划分为物理层和数据链路层，形成了一个新的五层结构。TCP/IP 是一系列协议的集合，所以严格的称呼应该是 TCP/IP 协议簇。

TCP/IP 协议簇的前四层与 OSI 参考模型的前四层相对应，其功能也非常类似，而应用层则与 OSI 参考模型的最高三层相对应，如图 2.3 所示。

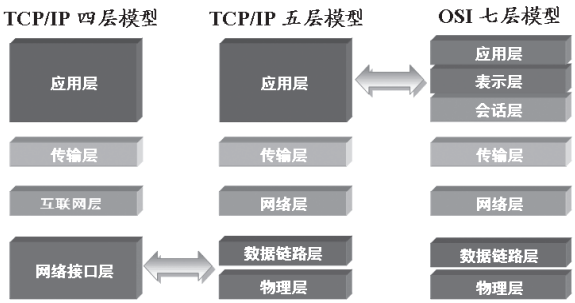


图 2.3　OSI 参考模型与 TCP/IP 协议簇

值得注意的是，OSI 参考模型没有考虑任何一组特定的协议，因此 OSI 更具通用性；而 TCP/IP 参考模型与 TCP/IP 协议簇吻合得很好，虽然该模型不适用于其他任何协议栈，但如今的网络多以 TCP/IP 协议簇作为基础，在分层设计上没有过多考虑协议的 OSI 分层理念，故 OSI 模型没有广泛地应用于实际工作中。相反，人们更多地应用 TCP/IP 分层模型在实际工作中分析问题、解决问题。

TCP/IP 五层模型应用得更广泛，本书及以后的内容在讨论问题时一律采用五层模型。下面是该模型对应的一些常见协议，如图 2.4 所示。

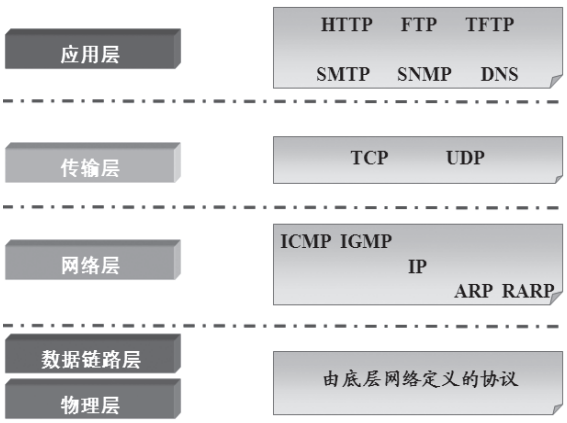


图 2.4　TCP/IP 五层模型常见协议

（1）物理层和数据链路层

在物理层和数据链路层，TCP/IP 并没有定义任何特定的协议。它支持所有标准的和专用的协议，网络可以是局域网（如广泛使用的以太网）、城域网或广域网。所以，TCP/IP 实际上只有三个层次。

（2）网络层

在网络层，TCP/IP 定义了网络互联协议（Internet Protocol，IP），而 IP 又由四个支撑协议组成：ARP（地址解析协议）、RARP（逆地址解析协议）、ICMP（网际控制报文协议）和 IGMP（网际组管理协议）。

（3）传输层

传统上，TCP/IP 有两个传输层协议：TCP（传输控制协议）和 UDP（用户数据报协议）。TCP 协议传输更加稳定可靠，UDP 协议传输效率更高。

（4）应用层

在应用层，TCP/IP 定义了许多协议，如 HTTP（超文本传输协议）、FTP（文件传输协议）、SMTP（简单邮件传输协议）、DNS（域名系统）等。

上述这些协议将在后续课程中具体讲解，这里只要明确协议与各层的对应关系即可。当我们研究具体协议的应用时，结合该协议所在层功能来理解和分析问题将事半功倍。

2.2　数据传输过程

2.2.1　数据封装与解封装过程（一）

下面我们将以 TCP/IP 五层结构为基础来学习数据在网络中传输的“真相”。由于这个过程比较抽象，我们可以类比给远在美国的朋友邮寄圣诞节礼物的过程。

如图 2.5 所示，当给朋友写一封信时，一定会遵照一个约定俗成的信件格式去写信。例如，在开头写对收信人的称呼，接下来是问候语“你好”等，中间是信的内容，最后落款写自己的姓名、日期等。那么，这个信件格式以及通信采用的语言实际上就是和朋友之间的协议，只有遵照这个协议，对方才能读懂信件。

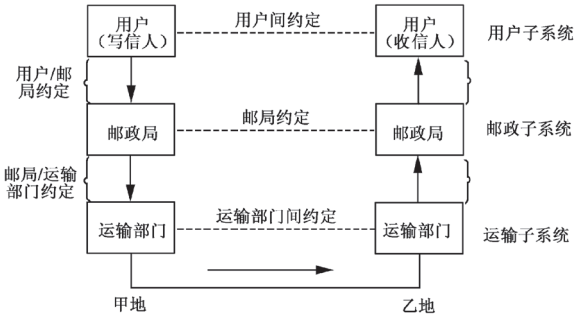


图 2.5　邮政系统分层模型

写好了信，要将信装在信封中。在信封上，要书写收信人的地址和姓名等。再将信交给邮局。

邮局根据收信人的地址，将信件再次封装成大的包裹，通过运输部门发往目的城市。

运输部门会将信件的包裹送达目的地的邮局。目的地的邮局会将信件送达收信人手中。

在这个寄信的例子中，一封信的传输需要经过三个层次，首先发信和收信的双方是这个过程中的最高层，位于下层的邮局和运输部门都是为了最高层之间的通信服务。寄信人与收信人之间要有一个协议，这个协议保证收信人能读懂寄信人的信件。两地的邮局和运输部门之间也有约定，如包裹的大小、地址的书写方式、运输到站的时间等。

邮局是寄信人和收信人的下一层，为上一层提供服务，邮局为寄信人提供服务时，邮筒就是两个层之间的“接口”。

1. 数据封装过程

正如前一节内容所讲，在计算机网络中层次的划分要比上述的例子细致，每一层实现的功能也更为复杂。为了能够更明确地说明此过程，我们将以两台主机的通信为实例进行分析讲解，如图 2.6 所示。

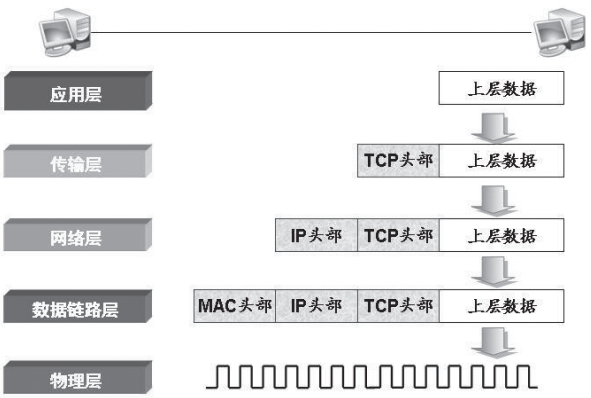


图 2.6　数据封装过程

（1）应用层传输过程

在应用层，数据被“翻译”为网络世界使用的语言——二进制编码数据。大家可以试想一下，人们需要通过计算机传输的数据形式千变万化、各式各样，有字母、数字、汉字、图片、声音等。这些信息对于单纯通过弱电流传输的计算机来说太过于“复杂”，因此这些方便人类识别的信息被应用层通过各种特殊的编码过程转换成二进制数据。这就是上面所描述的“翻译”过程，也是应用层在网络数据传输过程中最为核心的贡献。

（2）传输层传输过程

在传输层，上层数据被分割成小的数据段，并为每个分段后的数据封装 TCP 报文头部。应用层将人们需要传输的信息转换成计算机能够识别的二进制数据后，这

些数据往往都是海量的。例如，一张高清晰的图片转换成二进制数据可能会有几百万甚至几千万位比特，一次性传输如此庞大的数据，一旦网络出现问题而导致数据出错就要重新传输，数据量过大也会增加出错的概率，最终可能导致网络资源耗尽。因此，将数据先分割成小段再逐段传输，一旦数据传输出现错误只需重传这一小段数据即可。

在 TCP 头部有一个关键的字段信息——端口号，它用于标识上层的协议或应用程序，确保上层应用数据的正常通信。计算机是可以多进程并发运行的，如图2.6中的例子，左边的计算机在通过 QQ 发送信息的同时也可以通过 IE 浏览右边主机的 Web 页面，对于右边的主机就必须搞清左边主机发送的数据要对哪个应用程序实施通信。但是对于传输层而言，它是不可能“看懂”应用层传输数据的具体内容的，因此只能借助一种标识来确定接收到的数据对应的应用程序，这种标识就是端口号。

（3）网络层传输过程

在网络层，上层数据被封装上新的报文头部——IP 头部。值得注意的是，这里所说的上层数据包括 TCP 头部，也就是说，这里的上层是指传输层。对于网络层而言，它是“看不懂”TCP 包头中的内容的，无论是应用层的应用数据，还是 TCP 头部信息都属于上层数据。

在 IP 头部中有一个关键的字段信息——IP 地址，它是由一组 32 位的二进制数组成的，用于标识网络的逻辑地址。回想刚才寄信的例子，我们在信封上填写了对方的详细地址和本地的详细地址，以保证收件人能够顺利收到信件。网络层的传输过程与其类似，在 IP 头部中包含目标 IP 地址和源 IP 地址，在网络传输过程中的一些中间设备，如路由器，会根据目标 IP 地址来进行逻辑寻址，找到正确的路径将数据转发到目的端主机。如果中间的路由设备发现目标的 IP 地址是不可能到达的，它将会把该消息传回发送端主机，因此在网络层需要同时封装目标 IP 和源 IP。

（4）数据链路层传输过程

在数据链路层，上层数据被封装一个 MAC 头部，其内部有一个关键的字段信息——MAC 地址，它由一组 48 位的二进制数组成。在目前阶段，我们可以先把它理解为固化在硬件设备中的物理地址，具有全球唯一性。例如，之前讲解的网卡就有属于自己的唯一的 MAC 地址。和 IP 头部类似，在 MAC 头部也同时封装着目标 MAC 地址和源 MAC 地址。其实，二层封装还涉及尾部的封装，考虑大家目前的学习层次，不再详述，后续会讲解相关内容。

（5）物理层传输过程

无论在之前封装的报文头部还是上层的数据信息都是由二进制数组成的，在物理层，将这些二进制数字组成的比特流转换成电信号在网络中传输。

2. 数据解封装过程

数据被封装完毕通过网络传输到接收方后，将进入数据的解封装过程，这将是封装过程的一个逆过程，如图 2.7 所示。

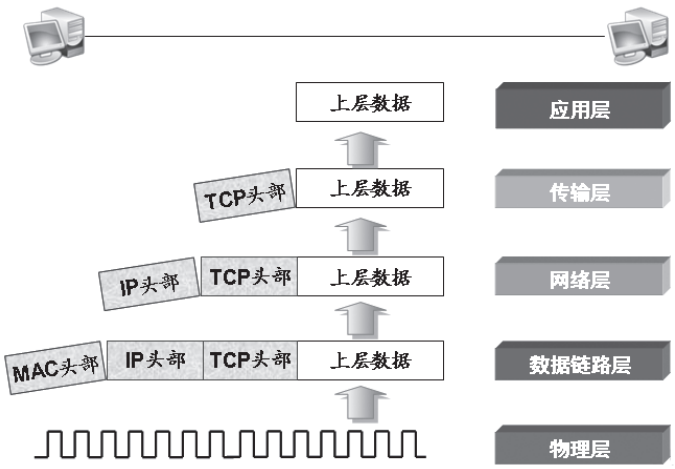


图 2.7　数据解封装过程

在物理层，首先将电信号转换成二进制数据，并将数据送至数据链路层。在数据链路层，将查看目标 MAC 地址，判断其是否与自己的 MAC 地址吻合，并据此完成后续处理。如果数据报文的目标 MAC 地址就是自己的 MAC 地址，数据的 MAC 头部将被“拆掉”，并将剩余的数据送至上一层；如果目标 MAC 地址不是自己的 MAC 地址，对于终端设备来说，它将会丢弃数据。网络层与数据链路层类似，目标 IP 地址将被核实是否与自己的 IP 地址相同，从而确定是否送至上一层。到了传输层，首先要根据 TCP 头部判断数据段送往哪个应用层协议或应用程序，然后将之前被分组的数据段重组，再送往应用层；在应用层。这些二进制数据将经历复杂的解码过程，以还原成发送者所传输的最原始的信息。

3. 数据传输的一些基本概念

（1）PDU

对于 OSI 参考模型而言，每一层都是通过协议数据单元来进行通信的；而对于 TCP/IP 五层结构，也可以沿用这个概念。PDU（Protocol Data Unit，协议数据单元）是指同层之间传递的数据单位。例如：TCP/IP 五层结构体系中，上层数据被封装了 TCP 头部后，这个单元称为段（Segment）；数据段向下传到网络层，被封装了 IP 头部后，这个单元称为包（Packet）；数据包继续向下传送到数据链路层，被封装了 MAC 头部后，这个单元称为帧（Frame）；最后帧传送到物理层，帧数据变成比特（Bits）流；比特流通过物理介质传送出去，如图 2.8 所示。

（2）常见硬件设备与五层模型的对应关系

常见的设备属于哪一层并没有严格的定义或者是官方的 RFC 文档说明，但是了解网络设备属于哪一层对于后续的网络硬件课程学习具有很好的指导意义。

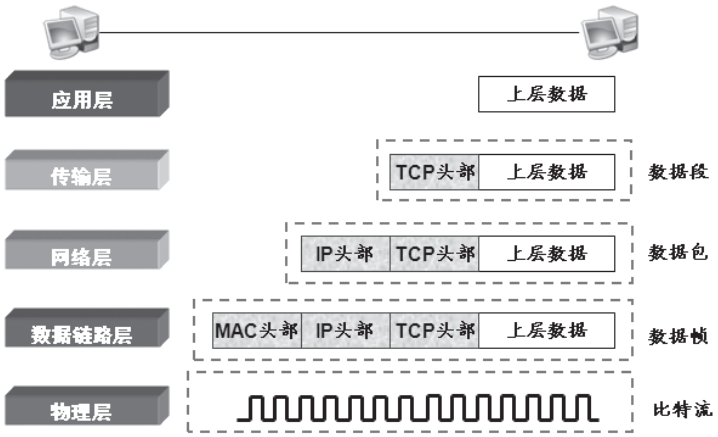


图 2.8　PDU 协议数据单元

设备属于哪一层要看这个设备主要工作在哪一层。一般来说，常用的个人计算机和服务器都属于应用层设备，因为计算机包含所有各层的功能。由器属于网络层设备，因为路由器的主要功能是网络层的逻辑寻址。传统的交换机属于数据链路层设备（这里之所以说传统，是因为如今三层、四层的交换机已经非常普遍了），因为交换机的主要功能是基于 MAC 地址的二层数据帧交换。网卡一般意义上定义在物理层，虽然目前有些高端的网卡甚至涵盖防火墙的功能，但其最主要、最基本的功能仍是物理层通信。还有就是硬件防火墙，从理论上讲，属于传输层设备，因为它主要基于传输层端口号来过滤上层应用数据的传输，但是需求永远是网络行业发展的源动力，如今的防火墙更注重整体解决方案的实现。对于病毒、木马、垃圾邮件的过滤已经成为防火墙的附属功能，而且在企业中也已经广泛应用，因此，很多人愿意将防火墙归属于应用层。如表 2-1 所示为网络中各层典型的硬件设备。

表 2-1　网络中各层典型的硬件设备

层名称	应用层	传输层	网络层	数据链路层	物理层
典型设备	计算机	防火墙	路由器	交换机	网卡

2.2.2　数据封装与解封装过程（二）

如果网络世界只有终端设备，那将不能称之为网络。正因为有很多中转设备才形成了复杂的 Internet，只不过作为网络用户的我们没有机会感知它们的存在，这都是传输层的“功劳”。由于传输层通过端口号辅助上层建立最终用户间的端到端会话，因此对于最终用户而言，数据的真实传输过程都被隐藏起来。例如，通过 QQ 软件即时通信时，用户感觉好像在和对方面对面沟通，全然不知自己说的内容经过了多少交换机和路由器才到达对方那里，但这些过程是真实存在的。下面我们就结合封装过程具

体介绍一下这个过程。

首先需要明确一个问题，发送方与接收方各层之间必须采用相同的协议才能建立连接，实现正常的通信，如图 2.9 所示。例如，应用层之间必须采用相同的编码解码规则，才能保证用户信息传输的正确性；传输层之间必须采用相同的端口号与协议应关系，才能保证上层应用进程间的通信；网络层之间必须采用相同的逻辑寻址过程才能保证数据不会传输到错误的目的地。如果数据链路层采用的协议不同，接收方甚至都不能“理解”数据的内容；如果物理层的硬件接口规格不同，接收方甚至都无法接收到信号。

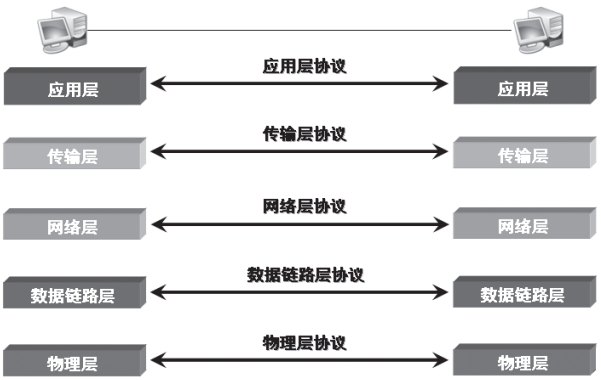


图 2.9　TCP/IP 五层模型各层间通信（一）

在实际的网络环境中，发送方和接收方往往相隔千山万水，中间会有很多的硬件设备起到中转的作用。为了说明整个过程，我们假设了一种通信结构，如图 2.10 所示。在两台通信的计算机之间增加了两台交换机和路由器，发送主机的数据将会经过这些“中间设备”才能到达接收主机。

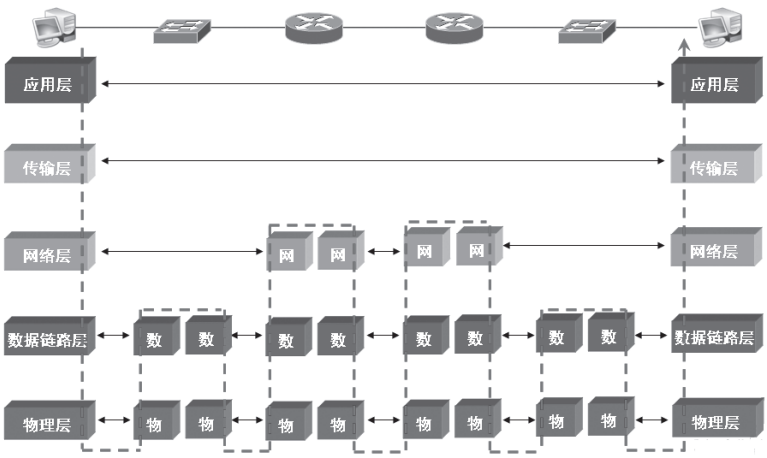


图 2.10　TCP/IP 五层模型各层间通信（二）

1）发送主机按照之前讲解的内容进行数据封装，这里不再赘述了。

2）从发送主机物理网卡发出的电信号通过网线到达交换机，交换机将电信号转换成二进制数据送往交换机的数据链路层。因为交换机属于数据链路层的设备，所以它将可以查看数据帧头部的内容，但不会进行封装和解封装过程。当交换机发现数据帧头部封装的 MAC 地址不属于自己，它不会像终端设备那样将数据帧丢弃，而是根据该 MAC 地址将数据帧智能地转发到路由器设备，在转发前要重新将二进制数据转换成物理的电信号。

3）当路由器收到数据后，会拆掉数据链路层的 MAC 头部信息，将数据送达网络层，这样 IP 头部信息就“暴露”在最外面。路由器将检测数据包头部的目标 IP 地址信息，并根据该信息进行路由转发，智能地将数据报文转发到下一跳路由器上，在转发前要重新封装新的 MAC 头部信息，并将数据转换成二进制。

4）之后的过程有点大同小异了……

从这个过程我们可以看出，数据在传输过程中不断地进行着封装和解封装的过程，中间设备属于哪一层就在哪一层对数据进行相关的处理，以实现设备的主要功能。也正因如此，我们称 TCP/IP 五层模型为“参考”模型，参考这五层模型可以帮助我们很好地研究网络中的设备以及设备工作过程中遵守的协议。

2.3　TCP 和 UDP 协议

TCP/IP 协议簇的传输层协议主要有两个：TCP（Transmission Control Protocol，传输控制协议）和 UDP（User Datagram Protocol，用户数据报协议）。

下面首先对 TCP 协议进行详细介绍，然后简单介绍 UDP 协议。

2.3.1　TCP 协议

TCP 是面向连接的、可靠的进程到进程通信的协议。TCP 提供全双工服务，即数据可在同一时间双向传输，每一个 TCP 都有发送缓存和接收缓存，用来临时存储数据。

1. TCP 报文段

TCP 将若干个字节构成一个分组，称为报文段（Segment）。TCP 报文段封装在 IP 数据报中，如图 2.11 所示。

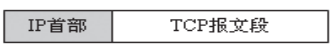


图 2.11　TCP 报文段的封装

TCP 报文段的首部格式如图 2.12 所示。

首部长度为 20 ～ 60 字节，以下是各字段的含义。

- 源端口号：它是 16 位字段，为发送方进程对应的端口号。

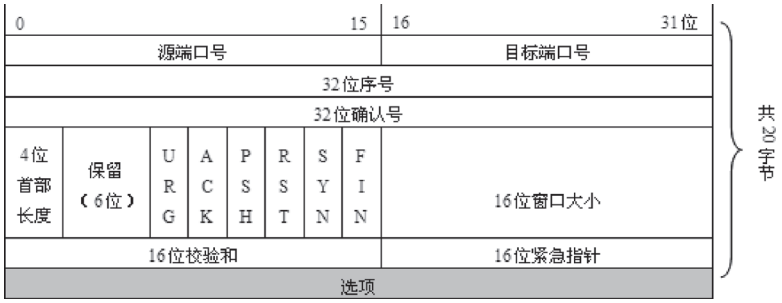


图 2.12　TCP 报文段的首部格式

- 目标端口号：它是 16 位字段，对应的是接收端的进程，接收端收到数据段后，根据这个端口号来确定把数据送给哪个应用程序的进程。
- 序号：当 TCP 从进程接收数据字节时，就把它们存储在发送缓存中，并对每一个字节进行编号。编号的特点如下。
  - 编号不一定从 0 开始，一般会产生一个随机数作为第 1 个字节的编号，称为初始序号（ISN），范围是 0 ～ 2^{32}-1。
  - TCP 每个方向的编号都是互相独立的。
  - 当字节都被编上号后，TCP 就给每个报文段指派一个序号，序号就是该报文段中第一个字节的编号。

当数据到达目的地后，接收端会按照这个序号把数据重新排列，保证数据的正确性。

- 确认号：确认号是对发送端的确认信息，用它来告诉发送端这个序号之前的数据段都已经收到，如确认号是 X，就表示前 X-1 个数据段都已经收到。
- 首部长度：用它可以确定首部数据结构的字节长度。一般情况下 TCP 首部是 20 字节，但首部长度最大可以扩展为 60 字节。
- 保留：这部分保留位作为今后扩展功能用，现在还没有使用到。
- 控制位：这六位有很重要的作用，TCP 的连接、传输和断开都受这六个控制位的指挥。各位的含义如下。
  - URG：紧急指针有效位。
  - ACK：只有当 ACK ＝ 1 时，确认序列号字段才有效；当 ACK ＝ 0 时，确认序列号字段无效。
  - PSH：标志位为 1 时要求接收方尽快将数据段送达应用层。
  - RST：当 RST 值为 1 时通知重新建立 TCP 连接。
  - SYN：同步序号位，TCP 需要建立连接时将这个值设为 1。
  - FIN：发送端完成发送任务位，当 TCP 完成数据传输需要断开连接时，提出断开连接的一方将这个值设为 1。
- 窗口值：说明本地可接收数据段的数目，这个值的大小是可变的，当网络通畅时将这个窗口值变大可加快传输速度，当网络不稳定时减小这个值可保证网络数据的可靠传输。TCP 协议中的流量控制机制就是依靠变化窗口值的大

小实现的。

- 校验和：用来做差错控制，与 IP 的校验和不同，TCP 校验和的计算包括 TCP 首部、数据和其他填充字节。在发送 TCP 数据段时，由发送端计算校验和，当到达目的地时再进行一次校验和计算。若两次的校验和一致，则说明数据基本是正确的，否则将认为数据已被破坏，接收端将丢弃数据。
- 紧急指针：和 URG 配合使用，当 URG ＝ 1 时有效。
- 选项：在 TCP 首部可以有多达 40 字节的可选信息。

2. TCP 连接

TCP 是面向连接的协议，它在源点和终点之间建立一条虚连接。大家可能会感到奇怪，为什么使用 IP（无连接协议）服务的 TCP 却是面向连接的？关键点是 TCP 的连接是虚连接，而不是物理连接。TCP 报文段封装成 IP 数据报后，每一个 IP 数据报可以走不同的路径到达终点，因此收到的 IP 数据报可能不按顺序到达，甚至可能损坏或丢失。如果一个报文段没有按顺序到达，那么 TCP 保留它，然后等待之前的报文段到达；如果一个报文段损坏或丢失，那么 TCP 就要重传。总之，TCP 会保证报文段是有序的。

在数据通信之前，发送端与接收端要先建立连接，等数据发送结束后，双方再断开连接。TCP 连接的每一方都是由一个 IP 地址和一个端口号组成的。

（1）连接建立

TCP 建立连接的过程称为三次握手。下面通过 Sniffer 抓包来分析三次握手的过程。实验环境由两台 Windows 主机 PC1 和 PC2 组成，确保两台主机通信正常，在 PC2 上搭建 Web 站点并安装 Sniffer Pro 软件，如图 2.13 所示。

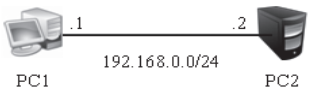


图 2.13　实验拓扑图

在 PC1 上启动 IE 浏览器访问 192.168.0.2，在 PC2 上用 Sniffer 抓到了很多数据包，只分析前三个数据包，如图 2.14 至图 2.16 所示。

- 第一次握手

PC1 使用一个随机的端口号向 PC2 的 80 端口发送建立连接的请求，此过程的典型标志就是 TCP 的 SYN 控制位为 1，其他五个控制位全为 0。

在图 2.14 中，源地址（Source Address）为 192.168.0.1，源端口号（Source Port）为 1276，目的地址（Dest Address）为 192.168.0.2，目的端口号（Destination Port）为 80，初始序列号（Initial Sequence Number）为 1552649478，标志位（Flags）中的 SYN 为 1。

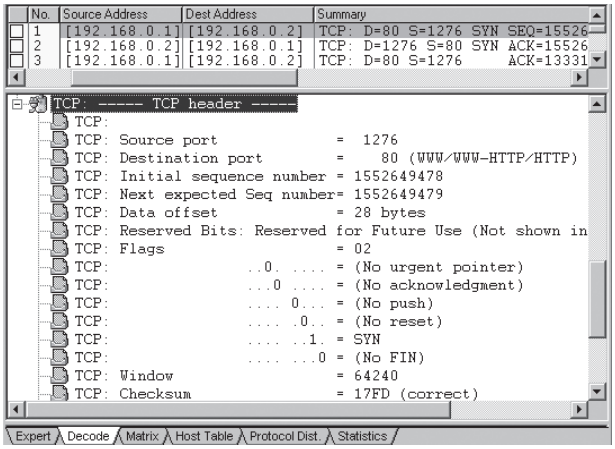


图 2.14　TCP 三次握手（1）

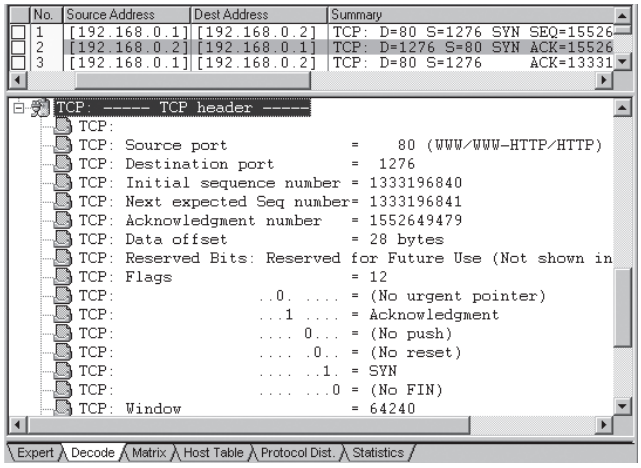


图 2.15　TCP 三次握手（2）

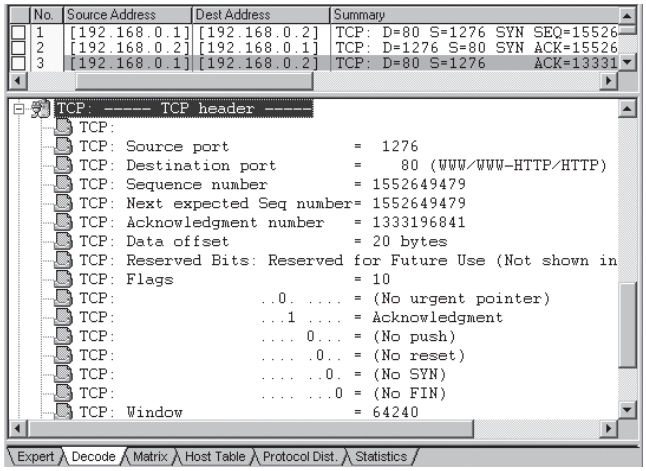


图 2.16　TCP 三次握手（3）

● 第二次握手

这一次握手实际上是分两部分来完成的。

1）PC2 收到了 PC1 的请求，向 PC1 回复一个确认信息，此过程的典型标志就是 TCP 的 ACK 控制位为 1，其他五个控制位全为 0，而且确认序列号是 PC1 的初始序列号加 1。

2）PC2 也向 PC1 发送建立连接的请求，此过程的典型标志和第一次握手一样，即 TCP 的 SYN 控制位为 1，其他五个控制位全为 0。

为了提高效率，一般将这两部分合并在一个数据包里实现。

在图 2.15 中，源地址（Source Address）为 192.168.0.2，源端口号（Source Port）为 80，目的地址（Dest Address）为 192.168.0.1，目的端口号（Destination Port）为 1276，确认序列号（Acknowledgment Number）为 1552649479，初始序列号（Initial Sequence Number）为 1333196840，标志位（Flags）中的 SYN 为 1，ACK 为 1。

● 第三次握手

PC1 收到了 PC2 的回复（包含请求和确认），也要向 PC2 回复一个确认信息，此过程的典型标志就是 TCP 的 ACK 控制位为 1，其他五个控制位全为 0，而且确认序列号是 PC2 的初始序列号加 1。

在图 2.16 中，源地址（Source Address）为 192.168.0.1，源端口号（Source Port）为 1276，目的地址（Dest Address）为 192.168.0.2，目的端口号（Destination Port）为 80，确认序列号（Acknowledgment Number）为 1333196841，标志位（Flags）中的 ACK 为 1。

这样就完成了三次握手，在 PC1 与 PC2 之间建立了 TCP 连接。

从以上的演示中可以将 TCP 三次握手总结为如图 2.17 所示的过程，图中 Seq 表示请求序列号，Ack 表示确认序列号，SYN 和 ACK 为控制位。

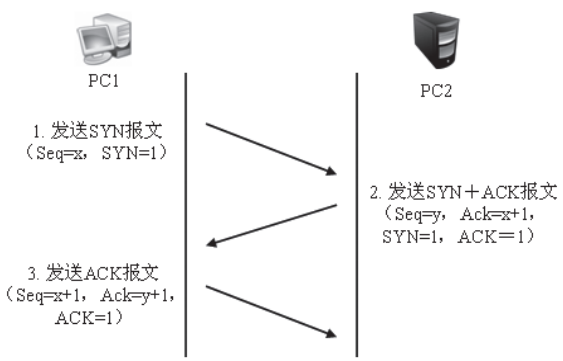


图 2.17　TCP 三次握手示意图

可以看出，SYN 控制位只有在请求建立连接时才被置为 1。

TCP 使用面向连接的通信方式，大大提高了数据传输的可靠性，使发送端和接收端在数据传输之前就有了交互，为正式数据传输打下了坚实的基础。

（2）连接终止

参加数据交换的双方中的任何一方（客户或服务器）都可以关闭连接。TCP 断开连接分四步，也称为四次握手，如图 2.18 所示。

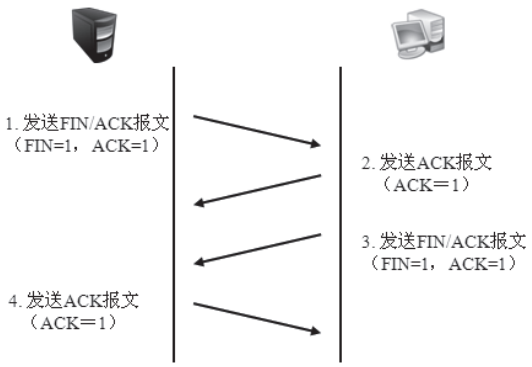


图 2.18　TCP 断开连接示意图

1）服务器向客户端发送 FIN 和 ACK 位置 1 的 TCP 报文段。

2）客户端向服务器返回 ACK 位置 1 的 TCP 报文段。

3）客户端向服务器发送 FIN 和 ACK 位置 1 的 TCP 报文段。

4）服务器向客户端返回 ACK 位置 1 的 TCP 报文段。

在 TCP 断开连接过程中，有一个半关闭的概念。TCP 一方（通常是客户端）可以终止发送数据，但仍然可以接收数据，称为半关闭。具体描述如下：

1）客户端发送 FIN 报文段，半关闭了这个连接，服务器发送 ACK 报文段接受半关闭。

2）服务器继续发送数据，而客户端只发送 ACK 确认，不再发送任何数据。

3）当服务器把所有数据都发送完毕时，就发送 FIN 报文段，客户再发送 ACK 报文段，这样就关闭了 TCP 连接。

请思考：

TCP 建立连接需要三次握手，为什么终止连接需要四次握手？

TCP 在网络中的应用范围很广，主要用在对数据传输可靠性要求高的环境中，如网页浏览，它使用的 HTTP 协议就是依赖 TCP 提供可靠性的。在使用 TCP 协议时，通信方对数据的可靠性要求高，即使降低了一些数据传输率也是可以接受的。

这样的例子有很多，如表 2-2 所示列出了一些常用的端口号及其功能。

2.3.2　UDP 协议

UDP 是一个无连接、不保证可靠性的传输层协议，也就是说发送端不关心发送的

数据是否到达目标主机、数据是否出错等，收到数据的主机也不会告诉发送方是否收到了数据，它的可靠性由上层协议来保障。既然 UDP 有这样的缺点，那为什么进程还愿意使用它呢？因为 UDP 也有优点，UDP 的首部结构简单，在数据传输时能实现最小的开销，如果进程想发送很短的报文而不关心可靠性，就可以使用 UDP。使用 UDP 发送很短的报文时，在发送端和接收端之间的交互要比使用 TCP 时少得多。

表 2-2　TCP 端口及其应用

端口	协议	说明
21	FTP	FTP 服务器所开放的控制端口
23	TELNET	用于远程登录，可以远程控制管理目标计算机
25	SMTP	SMTP 服务器开放的端口，用于发送邮件
80	HTTP	超文本传输协议

UDP 首部的格式如图 2.19 所示。

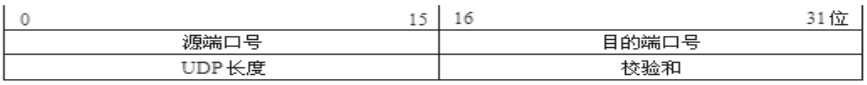


图 2.19　UDP 首部的格式

各字段的含义如下：

- 源端口号：用来标识数据发送端的进程，和 TCP 的端口号类似。
- 目的端口号：用来标识数据接收端的进程，和 TCP 的端口号类似。
- UDP 长度：用来指出 UDP 的总长度，为首部加上数据。
- 校验和：用来完成对 UDP 数据的差错检验，它的计算与 TCP 校验和类似。这是 UDP 提供的唯一可靠机制。

UDP 在实际工作中的应用范围很广。例如，聊天工具 QQ 在发送短消息时就是使用了 UDP 的方式。不难想象，如果发送十几个字的短消息也使用 TCP 进行一系列的验证，将导致传输率大大下降。有谁愿意用一个“反应迟钝”的软件进行网络聊天呢？在网络飞速发展的今天，网络技术日新月异，对于常用的简单数据传输来说，UDP 不失为一个很好的选择。在网络服务中也有用到 UDP 的，如表 2-3 所示列出了 UDP 常用的一些端口。

表 2-3　UDP 常用的一些端口

端口	协议	说明
69	TFTP	简单文件传输协议
111	RPC	远程过程调用
123	NTP	网络时间协议

本章总结

- OSI 参考模型的七层由低到高分别为物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。
- 早期的 TCP/IP 模型是一个四层结构，从下往上依次是网络接口层、互联网层、传输层和应用层。在后来的使用过程中，借鉴 OSI 的七层参考模型，将网络接口层又划分为物理层和数据链路层，形成了一个新的五层结构。
- TCP 报文段首部长度为 20 ～ 60 字节，其首部格式中有六个重要的控制位。而 UDP 的首部格式要简单得多。
- TCP 建立连接需要三次握手，而断开连接需要四次握手。

本章作业

1．简述 OSI 七层模型的各层功能。

2．简述 TCP/IP 五层模型的封装和解封装过程。

3．在客户端主机安装 Sniffer 软件，访问 Web 网站，在客户端通过抓包观察 TCP 建立连接的过程。

推荐步骤：

Step 1 准备工作

在客户端运行 Sniffer 软件开始抓包，然后通过 IE 浏览器访问 Web 服务器。

Step 2 过滤数据

过滤范围：在 Web 服务器和客户端主机之间。

Step 3 分析数据

分析三次握手的数据报文内容。

- 源和目标端口号。
- 初始序列号、确认号。
- 六个控制位。

4．用课工场 APP 扫一扫，完成在线测试，快来挑战吧！