

应用程序中，一个客户浏览器进程与一台 Web 服务器进程交换报文。在一个 P2P 文件共享系统中，文件从一个对等方中的进程传输到另一个对等方中的进程。对每对通信进程，我们通常将这两个进程之一标识为**客户**（client），而另一个进程标识为**服务器**（server）。对于 Web 而言，浏览器是一个客户进程，Web 服务器是一台服务器进程。对于 P2P 文件共享，下载文件的对等方标识为客户，上载文件的对等方标识为服务器。

你或许已经观察到，如在 P2P 文件共享的某些应用中，一个进程能够既是客户又是服务器。在 P2P 文件共享系统中，一个进程的确既能上载文件又能下载文件。无论如何，在任何给定的一对进程之间的通信会话场景中，我们仍能将一个进程标识为客户，另一个进程标识为服务器。我们定义客户和服务器进程如下：

在给定的一对进程之间的通信会话场景中，发起通信（即在该会话开始时发起与其他进程的联系）的进程被标识为**客户**，在会话开始时等待联系的进程是**服务器**。

在 Web 中，一个浏览器进程向一台 Web 服务器进程发起联系，因此该浏览器进程是客户，而该 Web 服务器进程是服务器。在 P2P 文件共享中，当对等方 A 请求对等方 B 发送一个特定的文件时，在这个特定的通信会话中对等方 A 是客户，而对等方 B 是服务器。在不致混淆的情况下，我们有时也使用术语“应用程序的客户端和服务端”。在本章的结尾，我们将逐步讲解网络应用程序的客户端和服务端的简单代码。

2. 进程与计算机网络之间的接口

如上所述，多数应用程序是由通信进程对组成，每对中的两个进程互相发送报文。从一个进程向另一个进程发送的报文必须通过下面的网络。进程通过一个称为**套接字**（socket）的软件接口向网络发送报文和从网络接收报文。我们考虑一个类比来帮助理解进程和套接字。进程可类比于一座房子，而它的套接字可以类比于它的门。当一个进程想向位于另外一台主机上的另一个进程发送报文时，它把报文推出该门（套接字）。该发送进程假定该门到另外一侧之间有运输的基础设施，该设施将把报文传送到目的进程的门口。一旦该报文抵达目的主机，它通过接收进程的门（套接字）传递，然后接收进程对该报文进行处理。

图 2-3 显示了两个经过因特网通信的进程之间的套接字通信（图 2-3 中假定由该进程使用的下面运输层协议是因特网的 TCP 协议）。如该图所示，套接字是同一台主机内应用层与运输层之间的接口。由于该套接字是建立网络应用程序的可编程接口，因此套接字也称为应用程序和网络之间的**应用程序编程接口**（Application Programming Interface, API）。应用程序开发者可以控制套接字在应用层端的一切，但是对该套接字的运输层端几乎没有控制权。应用程序开发者对于运输层的控制仅限于：①选择运输层协议；②也许能设定几个运输层参数，如最大缓存和最大报文段长度等（将在第 3 章中涉及）。一旦应用程序开发者选择了一个运输层协议（如果可供选择的话），则应用程序就建立在由该协议提供的运输层服务之上。我们将在 2.7 节中对套接字进行更为详细的探讨。

3. 进程寻址

为了向特定目的地发送邮政邮件，目的地需要有一个地址。类似地，在一台主机上运行的进程为了向在另一台主机上运行的进程发送分组，接收进程需要有一个地址。为了标识该接收进程，需要定义两种信息：①主机的地址；②定义在目的主机中的接收进程的标识符。