

从应用程序研发者的角度看,网络体系结构是固定的,并为应用程序提供了特定的服务集合。在另一方面,应用程序体系结构(application architecture)由应用程序研发者设计,规定了如何在各种端系统上组织该应用程序。在选择应用程序体系结构时,应用程序研发者很可能利用现代网络应用程序中所使用的两种主流体系结构之一:客户-服务器体系结构或对等(P2P)体系结构。

在客户-服务器体系结构(client-server architecture)中,有一个总是打开的主机称为服务器,它服务于来自许多其他称为客户的主机的请求。一个经典的例子是 Web 应用程序,其中总是打开的 Web 服务器服务于来自浏览器(运行在客户主机上)的请求。当 Web 服务器接收到来自某客户对某对象的请求时,它向该客户发送所请求的对象作为响应。值得注意的是利用客户-服务器体系结构,客户相互之间不直接通信;例如,在 Web 应用中两个浏览器并不直接通信。客户-服务器体系结构的另一个特征是该服务器具有固定的、周知的地址,该地址称为 IP 地址(我们将很快讨论它)。因为该服务器具有固定的、周知的地址,并且因为该服务器总是打开的,客户总是能够通过向该服务器的 IP 地址发送分组来与其联系。具有客户-服务器体系结构的非常著名的应用程序包括 Web、FTP、Telnet 和电子邮件。图 2-2a 中显示了这种客户-服务器体系结构。

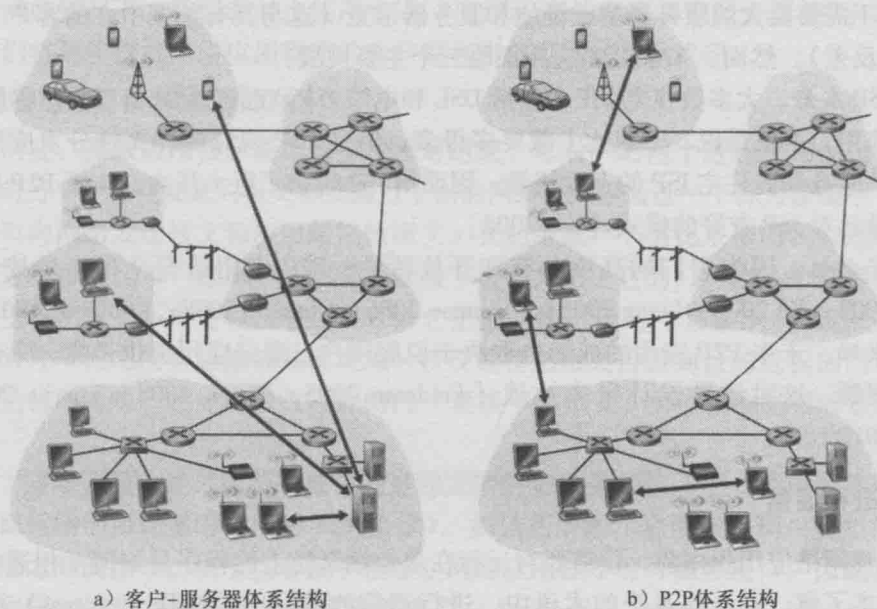


图 2-2 客户-服务器体系结构及 P2P 体系结构

在一个客户-服务器应用中,常常会出现一台单独的服务器主机跟不上它所有客户请求的情况。例如,一个流行的社交网站如果仅有一台服务器来处理所有请求,将很快变得不堪重负。为此,配备大量主机的数据中心常被用于创建强大的虚拟服务器。最为流行的因特网服务——如搜索引擎(如谷歌和 Bing)、因特网商务(如亚马逊和 e-Bay)、基于 Web 的电子邮件(如 Gmail 和雅虎邮件)、社交网络(如脸谱和推特),就应用了一个或多个数据中心。如在 1.3.3 节中所讨论的那样,谷歌有分布于全世界的 30~50 个数据中心,这些数据中心共同处理搜索、YouTube、Gmail 和其他服务。一个数据中心能够有数十万台服务器,它们必须要供电和维护。此外,服务提供商必须支付不断出现的互联和带