

天下没有不散的宴席，对于 TCP 连接也是这样。参与一条 TCP 连接的两个进程中的任何一个都能终止该连接。当连接结束后，主机中的“资源”（即缓存和变量）将被释放。举一个例子，假设某客户打算关闭连接，如图 3-40 所示。客户应用程序发出一个关闭连接命令。这会引客户 TCP 向服务器进程发送一个特殊的 TCP 报文段。这个特殊的报文段让其首部中的一个标志位即 FIN 比特（参见图 3-29）被设置为 1。当服务器接收到该报文段后，就向发送方回送一个确认报文段。然后，服务器发送它自己的终止报文段，其 FIN 比特被置为 1。最后，该客户对这个服务器的终止报文段进行确认。此时，在两台主机上用于该连接的所有资源都被释放了。

在一个 TCP 连接的生命周期内，运行在每台主机中的 TCP 协议在各种 TCP 状态（TCP state）之间变迁。图 3-41 说明了客户 TCP 会经历的一系列典型 TCP 状态。客户 TCP 开始时处于 CLOSED（关闭）状态。客户的应用程序发起一个新的 TCP 连接（可通过在第 2 章讲过的 Python 例子中创建一个 Socket 对象来完成）。这引起客户中的 TCP 向服务器中的 TCP 发送一个 SYN 报文段。在发送过 SYN 报文段后，客户 TCP 进入了 SYN\_SENT 状态。当客户 TCP 处在 SYN\_SENT 状态时，它等待来自服务器 TCP 的对客户所发报文段进行确认且 SYN 比特被置为 1 的一个报文段。收到这样一个报文段之后，客户 TCP 进入 ESTABLISHED（已建立）状态。当处在 ESTABLISHED 状态时，TCP 客户就能发送和接收包含有效载荷数据（即应用层产生的数据）的 TCP 报文段了。

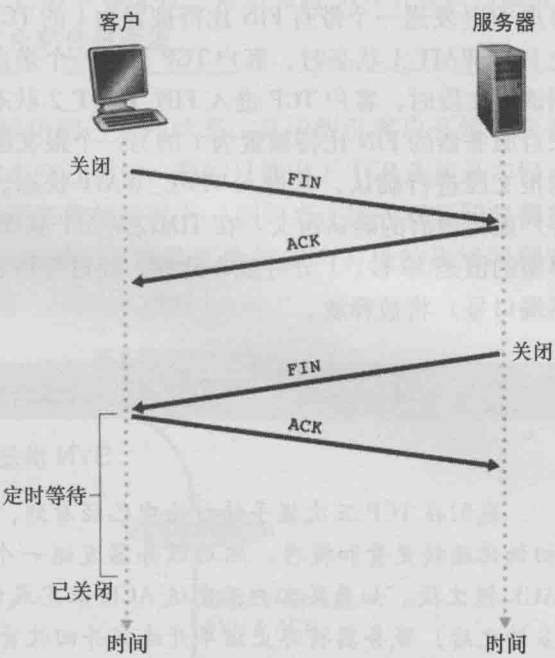


图 3-40 关闭一条 TCP 连接

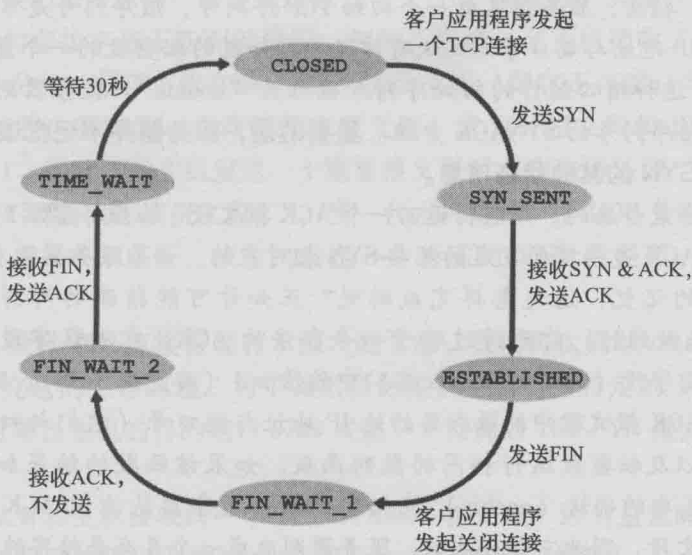


图 3-41 客户 TCP 经历的典型的 TCP 状态序列