

如果一个事务，不管是执行一次还是很多次，得到的结果都相同，这个事务就是幂等的。实现者们可以认为 GET、HEAD、PUT、DELETE、TRACE 和 OPTIONS 方法都共享这一特性。¹⁹ 客户端不应该以管道化方式传送非幂等请求（比如 POST）。否则，传输连接的过早终止就会造成一些不确定的后果。要发送一条非幂等请求，就需要等待来自前一条请求的响应状态。

尽管用户 Agent 代理可能会让操作员来选择是否对请求进行重试，但一定不能自动重试非幂等方法或序列。比如，大多数浏览器都会在重载一个缓存的 POST 响应时提供一个对话框，询问用户是否希望再次发起事务处理。

4.7.4 正常关闭连接

如图 4-19 所示，TCP 连接是双向的。TCP 连接的每一端都有一个输入队列和一个输出队列，用于数据的读或写。放入一端输出队列中的数据最终会出现在另一端的输入队列中。

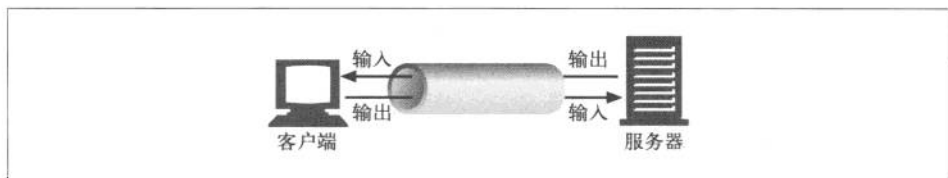


图 4-19 TCP 连接是双向的

1. 完全关闭与半关闭

应用程序可以关闭 TCP 输入和输出信道中的任意一个，或者将两者都关闭了。套接字调用 `close()` 会将 TCP 连接的输入和输出信道都关闭了。这被称作“完全关闭”，如图 4-20a 所示。还可以用套接字调用 `shutdown()` 单独关闭输入或输出信道。这被称为“半关闭”，如图 4-20b 所示。

2. TCP 关闭及重置错误

简单的 HTTP 应用程序可以只使用完全关闭。但当应用程序开始与很多其他类型的 HTTP 客户端、服务器和代理进行对话且开始使用管道化持久连接时，使用半关闭来防止对等实体收到非预期的写入错误就变得很重要了。

注 19：基于 GET 构建动态表单的管理者们要确保这些表单是幂等的。