

当你最终要去读取数据的时候，会得到一个连接被对端重置的错误，已缓存的未读响应数据都丢失了，尽管其中的大部分都已经成功抵达你的机器了。

### 3. 正常关闭

HTTP 规范建议，当客户端或服务器突然要关闭一条连接时，应该“正常地关闭传输连接”，但它并没有说明应该如何去做。

总之，实现正常关闭的应用程序首先应该关闭它们的输出信道，然后等待连接另一端的对等实体关闭它的输出信道。当两端都告诉对方它们不会再发送任何数据（比如关闭输出信道）之后，连接就会被完全关闭，而不会有重置的危险。

但不幸的是，无法确保对等实体会实现半关闭，或对其进行检查。因此，想要正常关闭连接的应用程序应该先半关闭其输出信道，然后周期性地检查其输入信道的状态（查找数据，或流的末尾）。如果在一定的时间区间内对端没有关闭输入信道，应用程序可以强制关闭连接，以节省资源。

## 4.8 更多信息

到这里我们对 HTTP 通道事务处理的介绍就结束了。更多有关 TCP 性能和 HTTP 连接管理功能的内容请参见下列参考资源。

### 4.8.1 HTTP连接

- <http://www.ietf.org/rfc/rfc2616.txt>  
RFC 2616，“超文本传输协议——HTTP/1.1”是 HTTP/1.1 的官方规范，解释了并行、持久和管道式 HTTP 连接的使用，以及用于实现这些连接的 HTTP 首部字段。此文档并未涵盖对底层 TCP 连接的正确使用。
- <http://www.ietf.org/rfc/rfc2068.txt>  
RFC 2068 是 HTTP/1.1 协议的 1997 年的版本。其中包含了 RFC 2616 中没有的、对 HTTP/1.0+ keep-alive 连接的解释。
- <http://www.ics.uci.edu/pub/ietf/http/draft-ietf-http-connection-00.txt>  
这个过期的因特网草案“HTTP Connection Management”，（“HTTP 连接管理”）探讨了 HTTP 连接管理面临的问题。

### 4.8.2 HTTP性能问题

- <http://www.w3.org/Protocols/HTTP/Performance/>