

奇怪的地方关闭连接。

对管道化持久连接来说，这种情形是很常见的。HTTP 应用程序可以在经过任意一段时间之后，关闭持久连接。比如，在持久连接空闲一段时间之后，服务器可能会决定将其关闭。

但是，服务器永远都无法确定在它关闭“空闲”连接的那一刻，在线路那一头的客户端有没有数据要发送。如果出现这种情况，客户端就会在写入半截请求报文时发现出现了连接错误。

4.7.2 Content-Length及截尾操作

每条 HTTP 响应都应该有精确的 Content-Length 首部，用以描述响应主体的尺寸。一些老的 HTTP 服务器会省略 Content-Length 首部，或者包含错误的长度指示，这样就要依赖服务器发出的连接关闭来说明数据的真实末尾。

客户端或代理收到一条随连接关闭而结束的 HTTP 响应，且实际传输的实体长度与 Content-Length 并不匹配（或没有 Content-Length）时，接收端就应该质疑长度的正确性。

如果接收端是个缓存代理，接收端就不应该缓存这条响应（以降低今后将潜在的错误报文混合起来的可能）。代理应该将有问题的报文原封不动地转发出去，而不应该试图去“校正” Content-Length，以维护语义的透明性。

4.7.3 连接关闭容限、重试以及幂等性

即使在非错误情况下，连接也可以在任意时刻关闭。HTTP 应用程序要做好正确处理非预期关闭的准备。如果在客户端执行事务的过程中，传输连接关闭了，那么，除非事务处理会带来一些副作用，否则客户端就应该重新打开连接，并重试一次。对管道化连接来说，这种情况更加严重一些。客户端可以将大量请求放入队列中排队，但源端服务器可以关闭连接，这样就会留下大量未处理的请求，需要重新调度。

101

副作用是很重要的问题。如果在发送出一些请求数据之后，收到返回结果之前，连接关闭了，客户端就无法百分之百地确定服务器端实际激活了多少事务。有些事务，比如 GET 一个静态的 HTML 页面，可以反复执行多次，也不会有什么变化。而其他一些事务，比如向一个在线书店 POST 一张订单，就不能重复执行，不然会有下多张订单的危险。

注 18：除非服务器怀疑出现了客户端或网络故障，否则就不应该在请求的中间关闭连接。