TCP中关于不正常连接的请求释放

**学号： 姓名： 班级：**

如图1所示，为正常的TCP连接关闭。

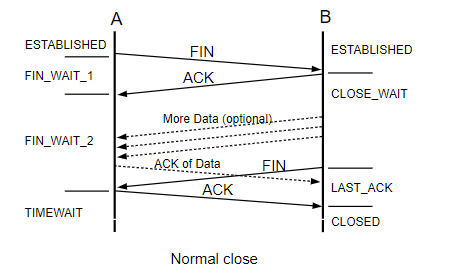


图1 TCP正常连接释放

TCP / IP会话的异常终止可能是由于传输或协议错误或数据错误导致接收消息处理程序不接受该数据而导致的。

由于TCP会话涉及两个对等级别的系统，因此错误恢复处理和异常会话终止过程可能会有所不同，具体取决于启动连接的系统。

不正常的连接请求释放可能是由于下列几个原因：

* 缺乏资源或网络中断（A lack of resources or network interruption）
* 会话期间的崩溃/错误（A crash/bug during the session）
* 其中一对已经关闭了连接的一部分，另一部分则继续发送数据（While one pair has already closed its part of the connexion, the other part continues to send data）
* 服务器拒绝打开到客户端的连接（The server refuses to open a connection to the client）

如果每个连接的SYN速率和服务器RST都很高，这意味着服务器拒绝客户端连接请求。有时，RST包相当“正常”。例如，当用户手动中断一个巨大的数据传输时。TCP会话正在尽可能快地发送数据包，因此当客户机发送FIN并关闭它的部分时，服务器仍在发送大量数据。在这种情况下，客户端发送RST包，直到服务器停止发送数据。

必须承认的是，关闭一个TCP连接可能比打开一个更复杂。会话可以通过双FIN、FIN + RST组合或仅通过RST包来关闭。但是，也可以在没有任何连接的情况下发送RST包。

通过查阅资料，发现了较为详细的原因列举，如表1所示。

表 1 TCP连接异常中止原因分析

|  |  |
| --- | --- |
| **Reason** | **Description** |
| Reset-I | TCP reset was from the inside |
| Reset-O | TCP reset was from the outside |
| TCP FINs | Normal shutdown sequence |
| FIN Timeout | Forced termination after 15 seconds awaiting                     last ACK |
| SYN Timeout | Forced termination after two minutes                      awaiting three-way handshake completion |
| Xlate Clear | Command-line removal |
| Deny | Terminated by application inspection |
| SYN Control | Back channel initiation from wrong side |
| Uauth Deny | Denied by URL filter |
| Unknown | Catch-all error |
| Conn-timeout | Connection was torn down because it was idle longer than the configured idle timeout |

发现一篇写得不错的博文（<https://blog.csdn.net/yusiguyuan/article/details/22515921>），我在此列举其中的4点。

服务端情景1：

* 原因：客户端程序正常运行的情况下，拔掉网线，杀掉客户端程序。
* 后果 ：这种情况下服务器程序没有检测到任何异常，并最后等待“超时”才断开TCP连接。

服务端情景2：

* 原因：客户端程序发送很多数据包后正常关闭Socket并exit进程(或不退出进程)。
* 后果：这种情况下服务器程序能够成功接收完所有消息，并最后收到“对端关闭”（Recv返回零）消息。

客户端情景1：

* 原因：服务器端在TCP的接收缓冲区中还有未接收数据的情况下close了Socket，客户端再收包。
* 后果：这种情况服务器端就会向对端发送RST包，而不是正常的FIN包（已经抓包证明），这就会导致客户端提前（RST包比正常数据包先被收到）。

客户端情景2：

* 原因：服务器端已经close了Socket，客户端再发送数据。
* 后果：第一包可以发送成功，但第二包发送失败。