TCP FIN\_WAIT\_2探究(2)

我们知道，在TCP\_WAIT\_2阶段，如果收到fin 置位，ack num 为snd\_nxt,seq num为rcv\_next的数据长度为零的tcp报文，则进入time-wait阶段，但是如果TCP\_WAIT\_2阶段收到一个非上述条件的报文，应该怎么处理呢，这个问题在rfc793和rfc1121以及rfc5961中都有答案，但是rfc793和rfc1121和rfc5961在某些点上存在冲突，另外，linux是采用哪种标准执行呢，这些问题我们依次分情况说明。

一个tcp 报文，主要元素如下，seq\_num、ack\_ num、rst flag、syn flag和fin flag和用户数据长度，这些元素和tcp的状态变量rcv\_nxt，rcv\_wnd,snd\_nxt互相作用决定当前tcp连接的处理结果。

首先看一下rfc793在FIN\_WAIT\_2阶段对收到的tcp报文处理情况:

1. 检查tcp报文的seq num、用户数据长度(data len)、tcp状态变量rcv\_nxt,rcv\_wnd的关系，分为如下几种情况：

|  |  |  |  |
| --- | --- | --- | --- |
| data len | rcv\_wnd | seq\_num和 rcv\_nxt关系 | 处理结果 |
| 等于0 | 等于0 | (seq\_num==rcv\_nxt) | 进入下一阶段 |
| 等于0 | 等于0 | (seq\_num ! =rcv\_nxt) | 回复ack，discard，return； |
| 等于0 | 大于0 | (seq\_num == rcv\_nxt) | 下一阶段 |
| 等于0 | 大于0 | (rcv\_nxt<seq\_num)&&  (seq\_num<=rcv\_nxt+rcv\_wnd) | 进入out of order，ignore； |
| 等于0 | 大于0 | (seq\_num <rcv\_nxt) ||  ( seq\_num > rcv\_nxt+rcv\_wnd) | 回复ack，discard,return |
| 大于0 | 等于0 | any | 回复ack,  discard,  Return； |
| 大于0 | 大于0 | (seq\_num <=rcv\_nxt+rcv\_wnd)&&  (seq\_num+data\_len >=rcv\_nxt)&&  (seq\_num==rcv\_nxt) | 进入下一阶段 |
| 大于0 | 大于0 | (Seq\_num <=rcv\_nxt+rcv\_wnd)&&  (seq\_num+data\_len >=rcv\_nxt )&&  (seq\_num !=rcv\_nxt) | 进入out of order |
| 大于0 | 大于0 | (seq\_num +data\_len<rxv\_nxt) ||  ( seq\_num > rcv\_nxt+rcv\_wnd) | 回复ack，  discard,  Return; |

1. 检查 RST bit是否置位

如果置位，则关闭tcp连接

1. 检查 SYN是否置位

如果置位，给对端发送reset。

1. 检查ACK是否置位

如果未置位，discard 、return。

如果置位，会有如下几种情况:

1. unack num < ack num <=snd\_nxt,则设置unack num为ack num，并释放对应的重传包
2. ack num < unack num，则忽略
3. ack num > snd\_nxt ，则回复ack ，discard ,return.
4. 检查FIN 是否置位

进入time-wait状态

rfc1122处理情况:

rfc1122主要是对rfc793的修正，rfc1172要求在fin-wait-2阶段如果是全双工关闭并且收到带有用户数据的tcp报文后必须给对端发送rst。

rfc5961处理情况:

rfc5961也是对rfc794的修正，rfc5961要求在fin-wait-2阶段如果收到in window的syn报文回复ack，而不是reset

linux处理情况:

linux基本按照rfc793规范处理，但是涉及到rfc1122和rfc5961的点，又会遵循后者的规范来实施