**TCP有限状态机**

2014220202021 李家兴

（电子科技大学 四川省成都市成华区一环路东一段240号 610054）

【摘要】有限状态机是一种用来进行对象行为建模的工具，其作用主要是描述对象在它的生命周期内所经历的状态序列，以及如何响应来自外界的各种事件。在计算机科学中，有限状态机被广泛用于建模应用行为、硬件电路系统设计、软件工程，编译器、网络协议、和计算与语言的研究。网络通信中，一个健壮的套接字应用程序必须能够处理网络中可能出现的各种状态，必须对TCP的有限状态机有所了解。TCP从建立到终止整个过程中，存在11种状态，TCP的有限状态机给出了TCP链接从一个状态转换到另一个状态的规则。

关 键 词 有限状态机; 生命周期; 网络通信; 套接字; 状态转换

**TCP finite state machine**

2014220202021 LI Jiaxing

(UESTC Chengdu City, Sichuan Province, Chenghua District, a loop of the East Section 240 610054)

**Abstract** Finite state machine is a tool used to model the behavior of the object, its role is to describe the object in its life cycle experienced by the state sequence, and how to respond to events from the outside world. In computer science, finite state machines are widely used in modeling application behavior, hardware circuit system design, software engineering, compilers, network protocols, and computational and linguistic research. In network communication, a robust socket application must be able to handle the various states that may be present in the network and must have an understanding of TCP's finite state machines. TCP from the establishment to the termination of the whole process, there are 11states, TCP finite state machine gives a TCP link from one state transition to another state rules.

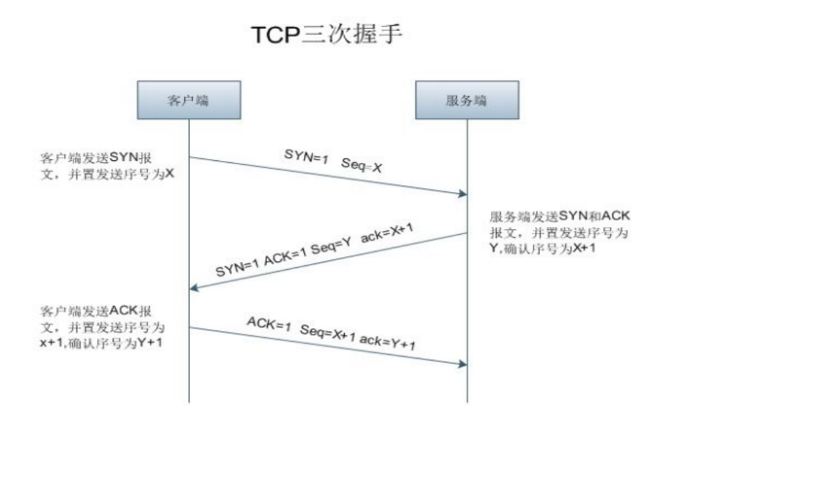
**Key words** Finite state machine; Life cycle; Telecommunication; Sockets; State transition

**引言** 分析思考并总结三次握手及关闭连接所会遇到的全部边界状况

## 1 对TCP有限状态机三次握手的思考与总结

### 1.1 三次握手

三次握手图解：



三次握手详解：

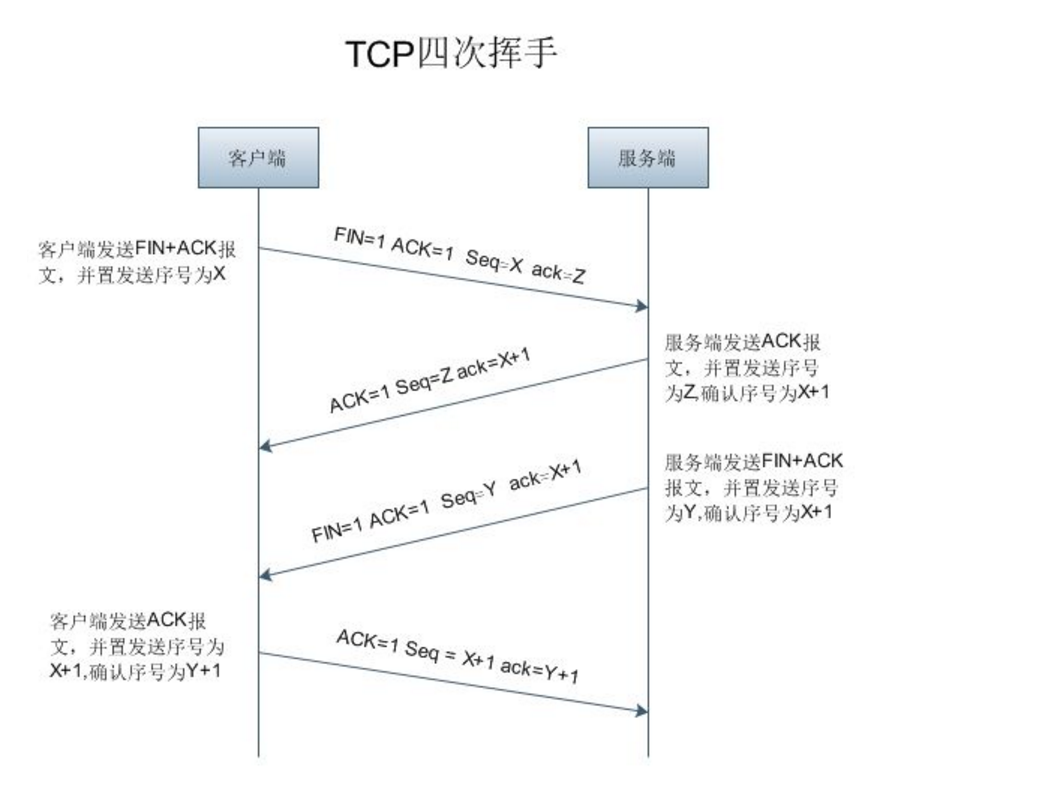
①第一次握手：客户端向服务器端发送连接请求包，标志位SYN（同步序号）置为1，产生序列号为X=0；

②第二次握手：服务器端收到客户端发过来报文，由SYN=1知道客户端要求建立联机。于是向客户端发送一个SYN和ACK都置为1的TCP报文，设置自己的初始序列号Y=0，将确认序号(ACK)设置为客户端的序列号加1，即X+1 = 0+1=1；

③第三次握手：客户端收到服务器发来的包后检查确认序号(ACK)是否正确，即第一次发送的序号加1（X+1=1）。以及标志位ACK是否为1。若正确，服务器再次发送确认包，ACK标志位为1，SYN标志位为0。确认序号(ACK)=Y+1=0+1=1，发送序号为X+1=1。客户端收到后确认序号值与ACK=1则连接建立成功，可以传送数据了。

### 1.2 提出问题（为什么三次握手，四次挥手）

四次挥手图解：



四次挥手详解：

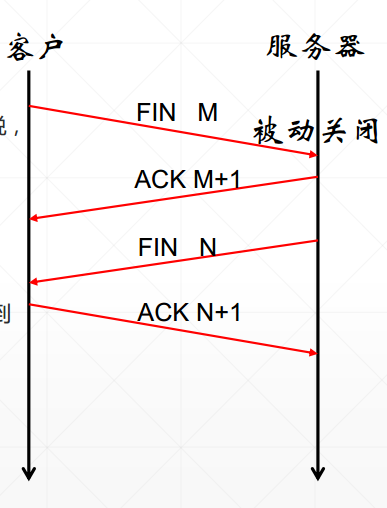
①第一次挥手：客户端给服务器发送TCP包，用来关闭客户端到服务器的数据传送。将标志位FIN和ACK置为1，序号为X=1，确认序列号为Z=1；

②第二次挥手：服务器收到FIN之后，发回一个ACK(标志位ACK=1),确认序号为收到的序号加1，即X=X+1=2。序号为收到的确认序号=Z；

③第三次挥手：服务器关闭与客户端的连接，发送一个FIN。标志位FIN和ACK置为1，序号为Y=1，确认序号为X=2；

④第四次挥手：客户端收到服务器端发送的FIN之后，发回ACK确认(标志位ACK=1),确认序号为收到的序号加1，即Y+1=2。序号为收到的确认序号X=2。

## 2 分析TCP有限状态机关闭连接会遇到的边界情况



### 2.1 连接未断开

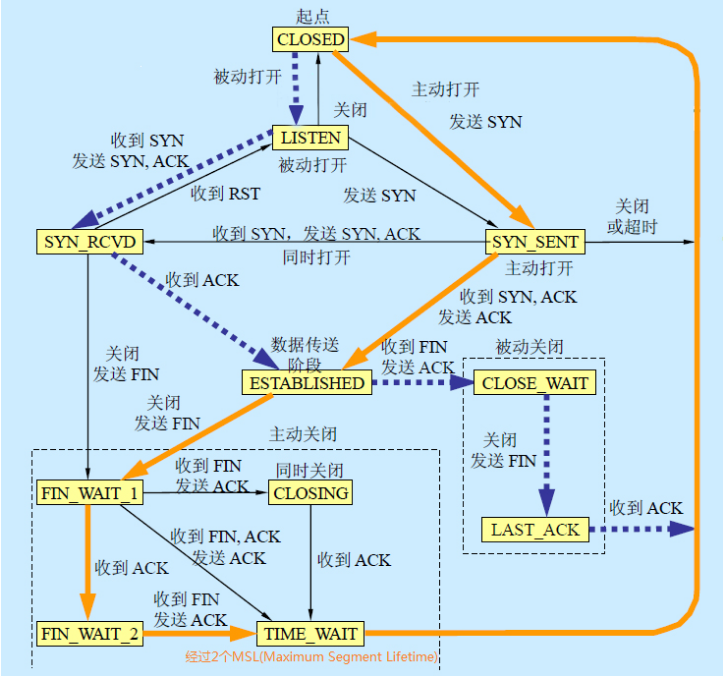
① 客户端没有收到FIN应答，超时，然后重发；

② 当前连接还未终止，但是服务器端收到多份FIN，服务器端对每个FIN都回复ACK确认序号，导致断开连接失败。

### 2.2 连接已经断开

1. 主动关闭端处于TIME\_WAIT状态，收到FIN则回复ACK确认序号；
2. 被动关闭端若已经断开连接，则回复RST，重置连接。
3. 主动关闭端处于CLOSE状态，收到FIN，回复RST，重置连接。

## 3 结 论或结 束 语



**CLOSED:** 表示初始状态。   
**LISTEN:** 表示服务器端的某个SOCKET处于监听状态，可以接受连接了。   
**SYN\_RCVD:** 这个状态表示接受到了SYN报文，在正常情况下，这个状态是服务器端的SOCKET在建立TCP连接时的三次握手会话过程中的一个中间状态，很短暂，基本上用netstat很难看到这种状态的，除非特意写一个客户端测试程序，故意将三次TCP握手过程中最后一个ACK报文不予发送。因此该状态下收到客户端的ACK报文后，它会进入到ESTABLISHED状态。   
**SYN\_SENT:** 表示客户端已发送SYN报文，当客户端SOCKET执行CONNECT连接时，它首先发送SYN报文，因此也随即它会进入到了SYN\_SENT状态，并等待服务端的发送三次握手中的第2个报文。   
**ESTABLISHED：**表示连接已经建立了。   
**FIN\_WAIT\_1:** 其实FIN\_WAIT\_1和FIN\_WAIT\_2状态都表示等待对方的FIN报文。   
\*\*而这两种状态的区别是：\*\*FIN\_WAIT\_1状态实际上是当SOCKET在ESTABLISHED状态时，它想主动关闭连接，向对方发送了FIN报文，此时该SOCKET即进入到FIN\_WAIT\_1状态。而当对方回应ACK报文后，则进入到FIN\_WAIT\_2状态，当然在实际的正常情况下，无论对方何种情况下，都应该马上回应ACK报文，所以FIN\_WAIT\_1状态一般是比较难见到的，而FIN\_WAIT\_2状态还有时常常可以用netstat看到。   
**FIN\_WAIT\_2：**实际上FIN\_WAIT\_2状态下的SOCKET，表示半连接，即有一方要求close连接，但另外还告诉对方，我暂时还有点数据需要传送给你，稍后再关闭连接。   
**TIME\_WAIT:** 表示收到了对方的FIN报文，并发送出了ACK报文，就等2MSL后即可回到CLOSED可用状态了。在FIN\_WAIT\_1状态下，收到了对方同时带FIN标志和ACK标志的报文时，可以直接进入到TIME\_WAIT状态，而无须经过FIN\_WAIT\_2状态。   
**CLOSING:** 表示双方都正在关闭SOCKET连接，属于一种比较罕见的例外状态。正常情况下，当发送FIN报文后，按理来说是应该先收到（或同时收到）对方的ACK报文，再收到对方的FIN报文。但是CLOSING状态表示发送FIN报文后，并没有收到对方的ACK报文，反而却收到了对方的FIN报文。如果双方几乎在同时close一个SOCKET的话，那么就出现了双方同时发送FIN报文的情况，即出现CLOSING状态。   
**CLOSE\_WAIT:** 表示在等待关闭。当对方close一个SOCKET后发送FIN报文给自己，你毫无疑问地会回应一个ACK报文给对方，此时则进入到CLOSE\_WAIT状态。接下来真正需要考虑的事情是查看是否还有数据发要送给对方，如果没有的话，那么就可以close这个SOCKET，发送FIN报文给对方，即关闭连接。所以在CLOSE\_WAIT状态下，需要完成的事情是等待你去关闭连接。   
**LAST\_ACK:** 这个状态还是比较容易好理解的，它是被动关闭一方在发送FIN报文后，最后等待对方的ACK报文。当收到ACK报文后，也即可以进入到CLOSED可用状态了。

参 考 文 献

[1] SONG B S, GRAY P R. A precision curvature-compensated CMOS bandgap reference[J]. IEEE Journal of Solid-State Circuits, 1983, 18(6): 634-643.

[2] AVOINNE C, RASHID T, CHOWDHURY V, et al. Second-order compensated bandgap reference with convex correction[J]. Electronic Letters, 2005, 41(5): 276-277.

[3] KER Ming-dou, CHEN Jung-sheng, CHU Ching-yun. New curvature-compensation technique for CMOS bandgap reference with sub-1-V operation[C]//The 2005 IEEE International Symposium on Circuits and Systems. Newyork: IEEE, 2005, 4: 3861-3864.

[4] http://blog.csdn.net/Celeste7777/article/details/47728869