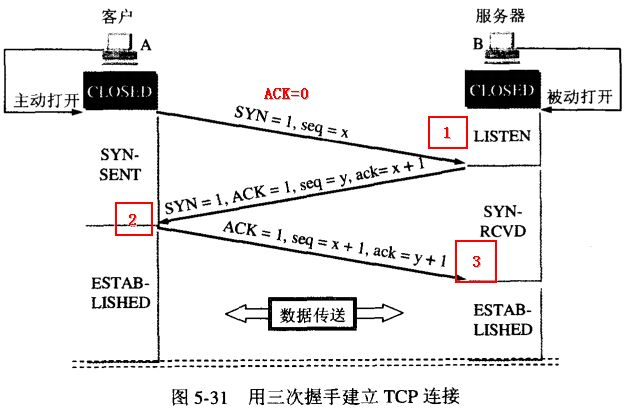
[**TCP三次握手详解及释放连接过程**](https://www.cnblogs.com/laowz/p/6947539.html)

TCP在传输之前会进行三次沟通，一般称为“三次握手”，传完数据断开的时候要进行四次沟通，一般称为“四次挥手”。

两个序号和三个标志位：

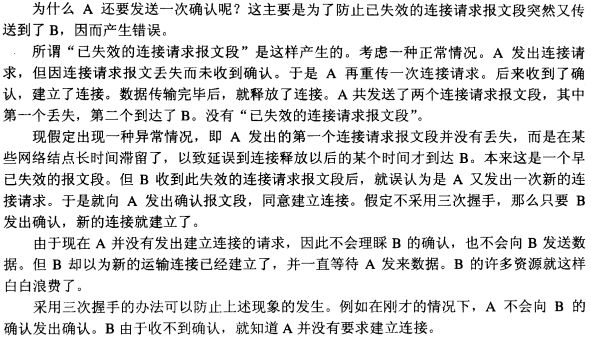
  （1）序号：seq序号，占32位，用来标识从TCP源端向目的端发送的字节流，发起方发送数据时对此进行标记。  
  （2）确认序号：ack序号，占32位，只有ACK标志位为1时，确认序号字段才有效，ack=seq+1。  
  （3）标志位：共6个，即URG、ACK、PSH、RST、SYN、FIN等，具体含义如下：  
  （A）URG：紧急指针（urgent pointer）有效。  
  （B）ACK：确认序号有效。  
  （C）PSH：接收方应该尽快将这个报文交给应用层。  
  （D）RST：重置连接。  
  （E）SYN：发起一个新连接。  
  （F）FIN：释放一个连接。  
  
 需要注意的是：  
  （A）不要将确认序号ack与标志位中的ACK搞混了。  
  （B）确认方ack=发起方req+1，两端配对。



在第一次消息发送中，A随机选取一个序列号作为自己的初始序号发送给B；第二次消息B使用ack对A的数据包进行确认，

因为已经收到了序列号为x的数据包，准备接收序列号为x+1的包，所以ack=x+1，同时B告诉A自己的初始序列号，就是seq=y；

第三条消息A告诉B收到了B的确认消息并准备建立连接，A自己此条消息的序列号是x+1，所以seq=x+1，而ack=y+1是表示A正准备接收B序列号为y+1的数据包。

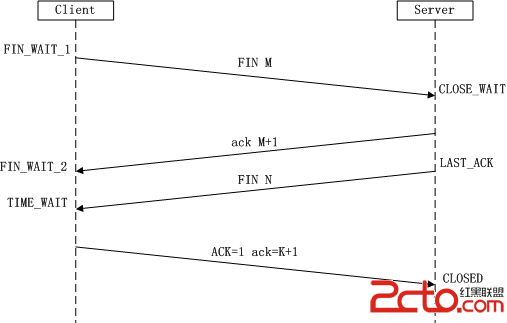


四次挥手：

由于TCP连接时全双工的，因此，每个方向都必须要单独进行关闭，这一原则是当一方完成数据发送任务后，发送一个FIN来终止这一方向的连接，

收到一个FIN只是意味着这一方向上没有数据流动了，即不会再收到数据了，但是在这个TCP连接上仍然能够发送数据，直到这一方向也发送了FIN。

首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭，上图描述的即是如此。  
 （1）第一次挥手：Client发送一个FIN，用来关闭Client到Server的数据传送，Client进入FIN\_WAIT\_1状态。  
  （2）第二次挥手：Server收到FIN后，发送一个ACK给Client，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号），Server进入CLOSE\_WAIT状态。  
 （3）第三次挥手：Server发送一个FIN，用来关闭Server到Client的数据传送，Server进入LAST\_ACK状态。  
  （4）第四次挥手：Client收到FIN后，Client进入TIME\_WAIT状态，接着发送一个ACK给Server，确认序号为收到序号+1，Server进入CLOSED状态，完成四次挥手。



（1）三次握手是什么或者流程？四次握手呢？答案前面分析就是。  
（2）为什么建立连接是三次握手，而关闭连接却是四次挥手呢？  
  这是因为服务端在LISTEN状态下，收到建立连接请求的SYN报文后，把ACK和SYN放在一个报文里发送给客户端。而关闭连接时，收到对方的FIN报文时，

仅仅表示对方不再发送数据了但是还能接收数据，己方也未必全部数据都发送给对方了，所以己方可以立即close，也可以发送一些数据给对方后，

再发送FIN报文给对方来表示同意现在关闭连接，因此，己方ACK和FIN一般都会分开发送。

**为什么TIME\_WAIT状态需要经过2MSL(最大报文段生存时间)才能返回到CLOSE状态？**

不应该是为了防止B发送的FIN=1的包的丢失，因为如果A没有收到来自B的释放连接请求，是不会进入TIME-WAIT状态的。

所以正确的解释是：A发送的确认释放连接信息B没有收到，这时候B会再次发送一个FIN=1的释放连接请求，而这个时候A还处于TIME-WAIT，所以可以再次发送确认信息