

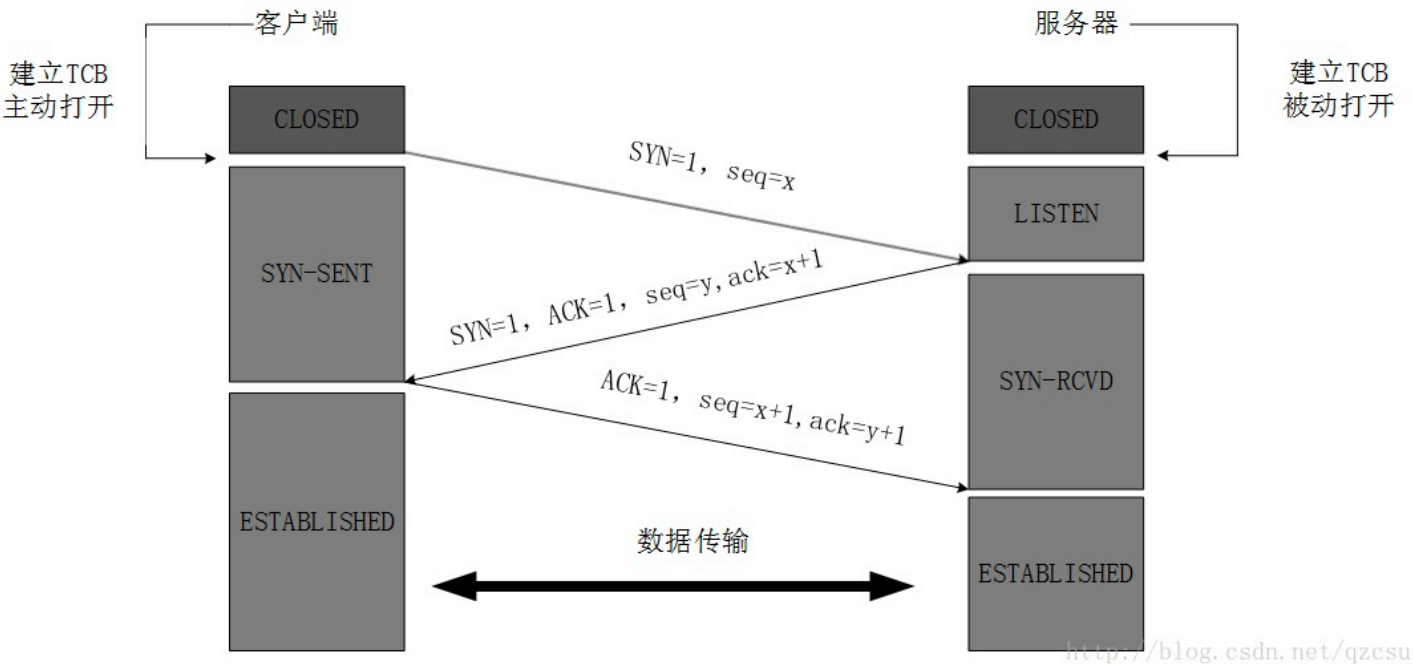
三次握手、四次挥手

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。

ACK=1 确认链接
 $ack=x+1$ 确认编号 提示客户端已经成功刚接手上一次所有数据

一、三次挥手：



- 1.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；
- 2.第二次消息发送时，B使用ack确认序号对A的数据包进行确认；
- 3.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。

为什么不是两次？

会有DOS攻击。（什么是DOS攻击）

为什么 A 还要发送一次确认呢？这主要是为了防止已失效的连接请求报文段突然又传送到了 B，因而产生错误。

所谓“已失效的连接请求报文段”是这样产生的。考虑一种正常情况。A 发出连接请求，但因连接请求报文丢失而未收到确认。于是 A 再重传一次连接请求。后来收到了确认，建立了连接。数据传输完毕后，就释放了连接。A 共发送了两个连接请求报文段，其中第一个丢失，第二个到达了 B。没有“已失效的连接请求报文段”。

现假定出现一种异常情况，即 A 发出的第一个连接请求报文段并没有丢失，而是在某些网络结点长时间滞留了，以致延误到连接释放以后的某个时间才到达 B。本来这是一个早已失效的报文段。但 B 收到此失效的连接请求报文段后，就误认为是 A 又发出一次新的连接请求。于是就向 A 发出确认报文段，同意建立连接。假定不采用三次握手，那么只要 B 发出确认，新的连接就建立了。

由于现在 A 并没有发出建立连接的请求，因此不会理睬 B 的确认，也不会向 B 发送数据。但 B 却以为新的运输连接已经建立了，并一直等待 A 发来数据。B 的许多资源就这样白白浪费了。

采用三次握手的办法可以防止上述现象的发生。例如在刚才的情况下，A 不会向 B 的确认发出确认。B 由于收不到确认，就知道 A 并没有要求建立连接。

二、四次挥手

由于TCP连接时全双工的（客户端和服务端不停地相互交流），因此，每个方向都必须单独进行关闭，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。

1.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

客户端进程发出连接释放报文，并且停止发送数据。释放数据报文首部，FIN=1，其序列号为seq=u（等于前面已经传送过来的数据的最后一个字节的序号加1），此时，客户端进入FIN-WAIT-1（终止等待1）状态。TCP规定，FIN报文段即使不携带数据，也要消耗一个序号。

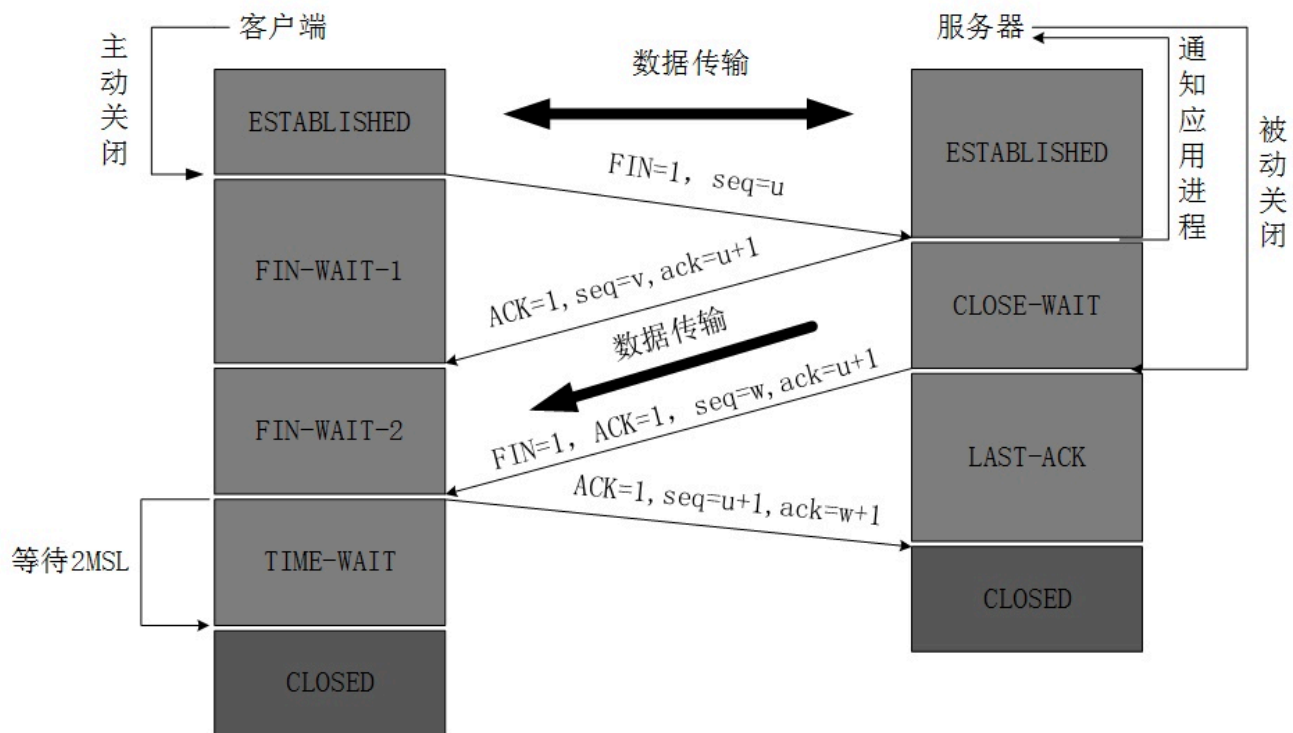
2.服务器收到连接释放报文，发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号seq=v，此时，服务端就进入了CLOSE-WAIT（关闭等待）状态。TCP服务器通知高层的应用进程，客户端向服务器的方向就释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。

客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的的数据）。

3.服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，ack=u+1，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq=w，此时，服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。

4.客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是seq=u+1，此时，客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2*MSL（最长报文段寿命）的时间后，当客户端撤销相应的TCB后，才进入CLOSED状态。

服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的TCP连接。可以看到，服务器结束TCP连接的时间要比客户端早一些。



<http://blog.csdn.net/qzcsu>

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server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？

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。