tcp报文段头部结构再次不赘述。（详见《tcp报文段头部》）

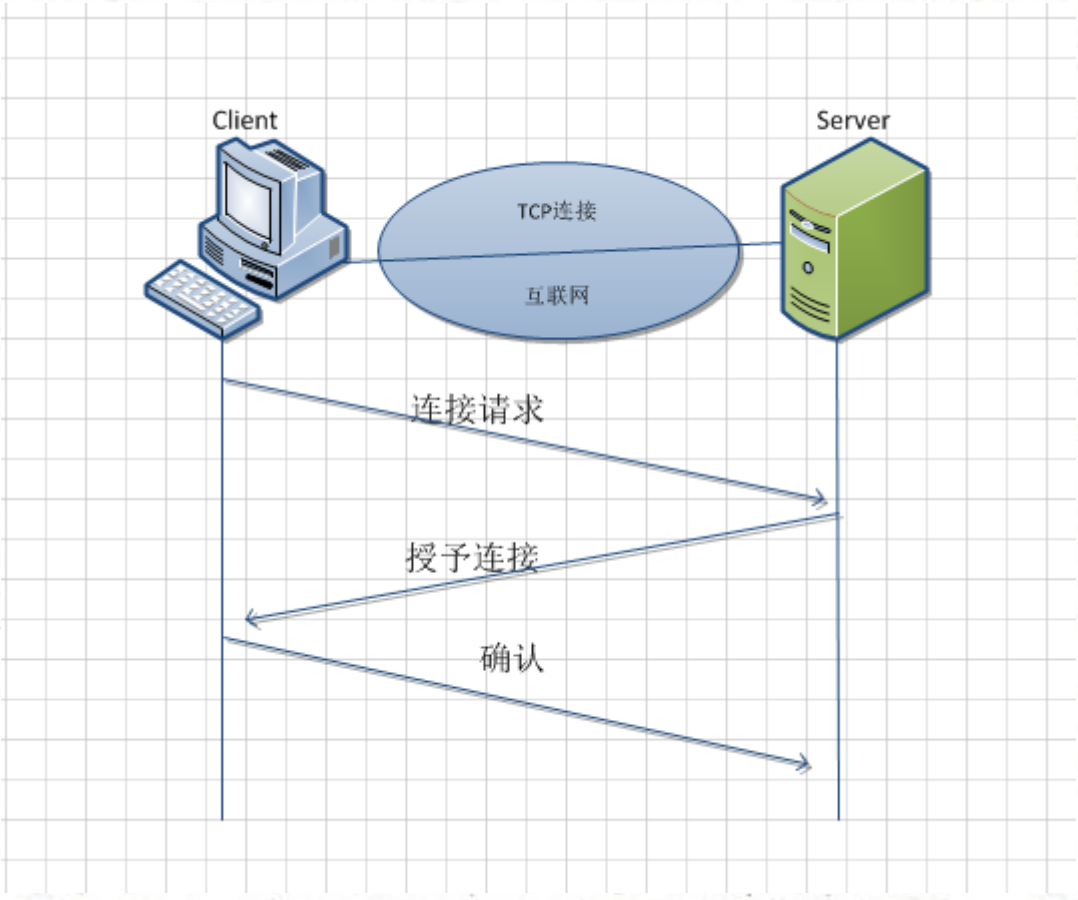
连接建立：

tcp连接建立需要经过“三次握手”：

1、A向B发送连接请求，tcp报文段中标志位SYN设值为1，同时设置序号基准值为X（该值为随机值）

2、B进行回复，tcp报文中标志位SYN设值为1，标志位ACK设置为1（表示对A发送请求的回复），确认号为X+1，同时设置序号为Y（随机值）

3、A对B的响应进行回复，其中标志位ACK设值为1，标志位SYN设值为0（此后均设置为0），确认号为Y+1，顺序号为X+1。三次握手完成。



连接断开：

tcp连接断开需要“四次握手”。因为tcp时全双工的协议，

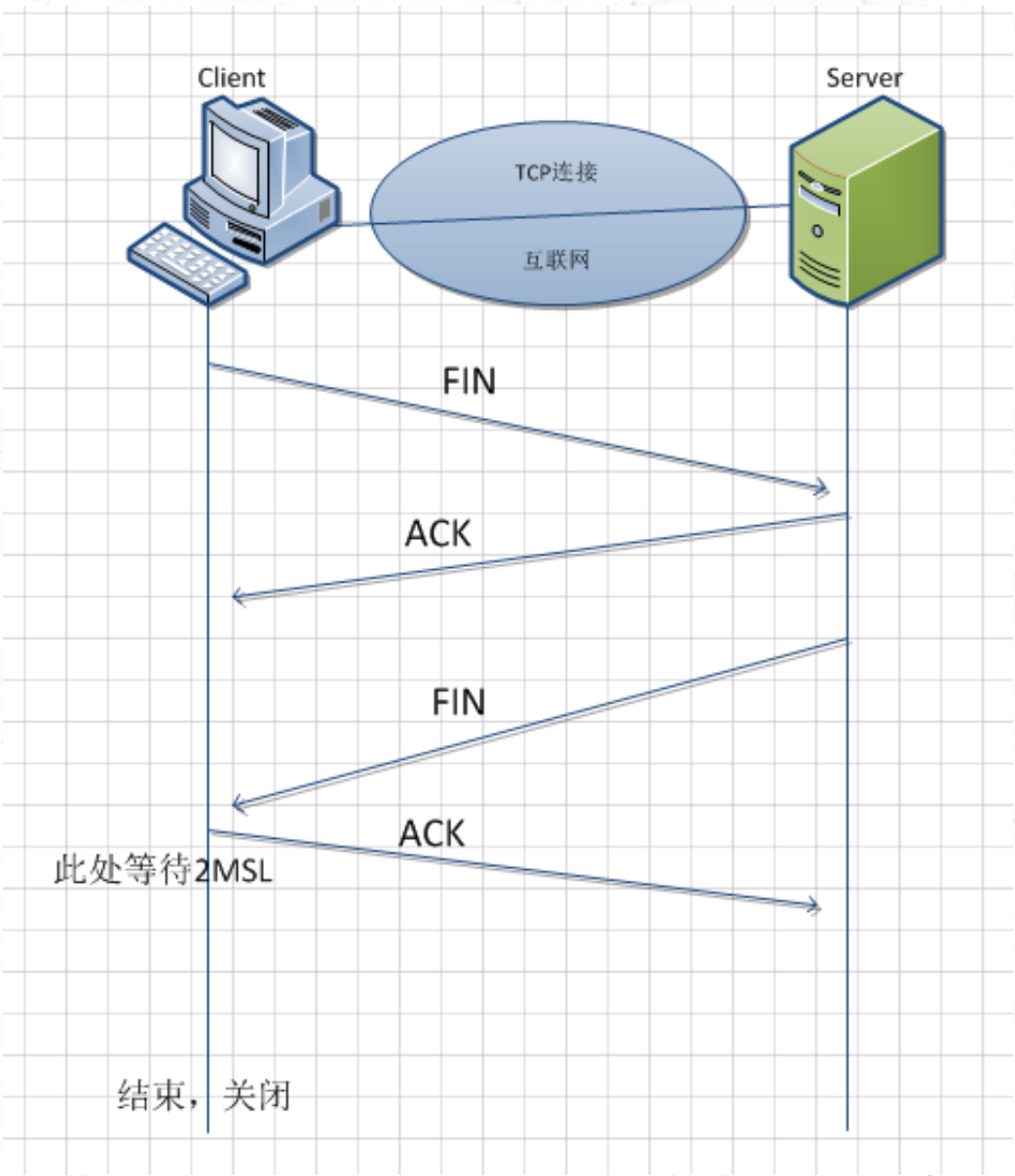
1、主机A发送tcp报文段，标志位FIN设值为1，序号为X（为最后一个数据包的序号+1）。且停止发送数据

2、主机B收到报文回复为A，ACK的值为1，确认号为X+1

3、主机B发送tcp报文段，标志位FIN设值为1，序号为Y，确认号为X+1，断开B到A的连接

4、主机A回复B，ACK字段设值为1，序号为Y+1 。

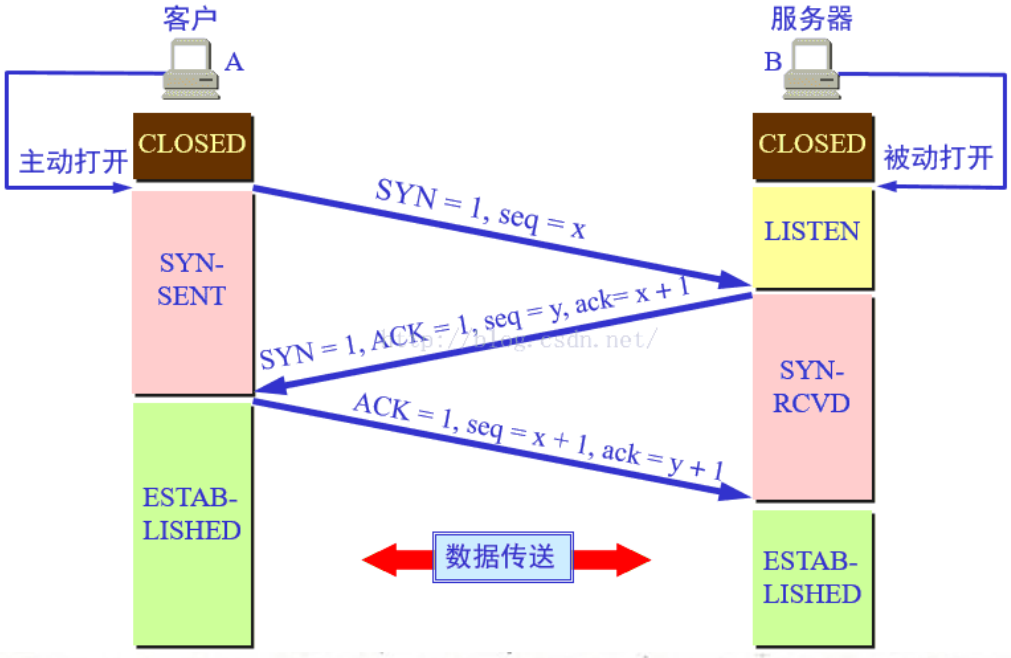
此时client端等待2msl（msl报文最大生存时间），若未再次收到server端FIN请求，则证明两端都正确关闭



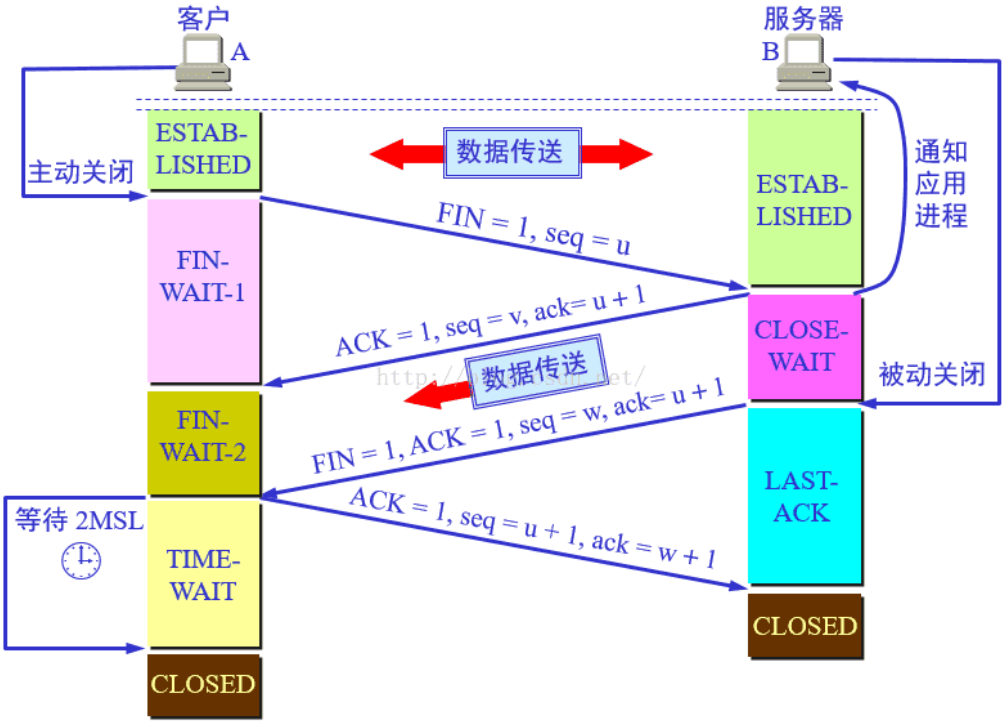
tcp状态转换：

TCP状态转换图如下所示：

连接建立



连接断开



tcp三次握手中状态转化

刚开始, 客户端和服务器都处于 CLOSE 状态.

此时, 客户端向服务器主动发出连接请求, 服务器被动接受连接请求.

1, TCP服务器进程先创建传输控制块TCB, 时刻准备接受客户端进程的连接请求, 此时服务器就进入了 LISTEN（监听）状态
2, TCP客户端进程也是先创建传输控制块TCB, 然后向服务器发出连接请求报文，此时报文首部中的同步标志位SYN=1, 同时选择一个初始序列号 seq = x, 此时，TCP客户端进程进入了 SYN-SENT（同步已发送状态）状态。TCP规定, SYN报文段（SYN=1的报文段）不能携带数据，但需要消耗掉一个序号。
3, TCP服务器收到请求报文后, 如果同意连接, 则发出确认报文。确认报文中的ACK=1, SYN=1, 确认序号是 x+1, 同时也要为自己初始化一个序列号 seq = y, 此时, TCP服务器进程进入了SYN-RCVD（同步收到）状态。这个报文也不能携带数据, 但是同样要消耗一个序号。
4, TCP客户端进程收到确认后还, 要向服务器给出确认。确认报文的ACK=1，确认序号是 y+1，自己的序列号是 x+1.
5, 此时，TCP连接建立，客户端进入ESTABLISHED（已建立连接）状态。当服务器收到客户端的确认后也进入ESTABLISHED状态，此后双方就可以开始通信了。

tcp四次挥手中状态的转化

数据传输完毕后，双方都可以释放连接.
此时客户端和服务器都是处于ESTABLISHED状态，然后客户端主动断开连接，服务器被动断开连接.

1, 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，确认序号为 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。

3, 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终数据）

4, 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，确认序号为v+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。

5, 客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，确认序号为w+1，而自己的序列号是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。

6, 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。