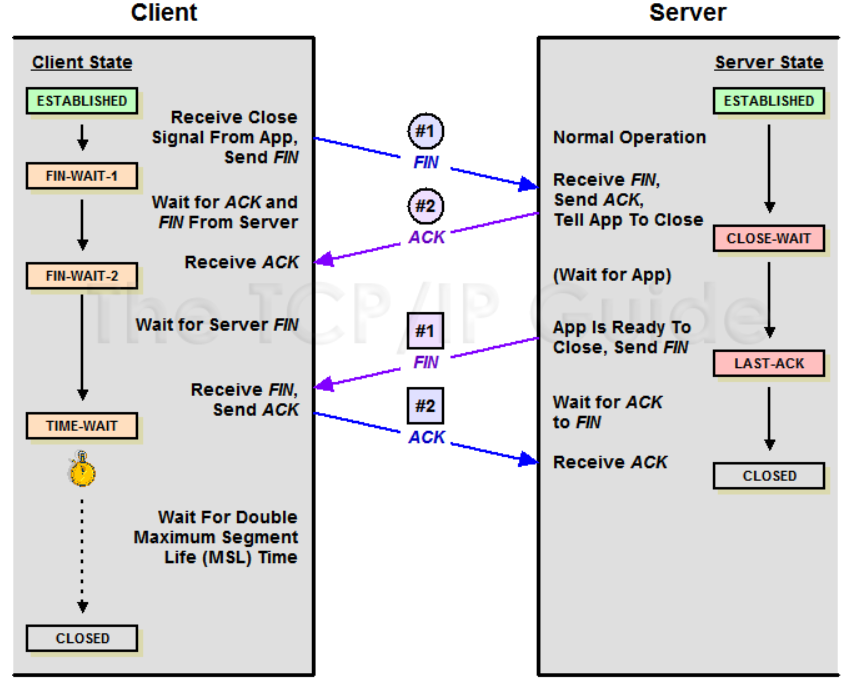


三次握手：

1. 客户端发送SYN包。此包中SYN设置为1，32位序号假设为x。客户端进入SYN\_SEND状态。
2. 服务端收到后，发送ACK+SYN包。也就是ACK设置为1，说明我收到了，32位确认序号为x+1；SYN设置为1说明，服务器端也想连接。同时32为序号假设为y。此时服务器端进入SYN\_RCVD。
3. 客户端收到后，发送ACK包。32位确认序号为y+1.此时客户端进入ESTABLISHED状态，当服务器端接收到报文后，也进入ESTABLISHED。  
     
   三次握手的目的是建立一个全双工的可信可靠的通信的信道。两次握手不行的原因？  
   现在做出一种假设：假设A发出一个请求连接的报文，此报文没有丢失，但是因为在某个网络节点长时间滞留了，导致延误到连接释放才到达B。此时，这个报文已经没有意义了，已经失效。但是B不知道，他不管什么样的报文，他还以为是新发出的报文，他都会发送ACK+SYN包。  
   如果只有两次握手，那么连接就已经建立了，B会等待A发来数据，他去响应，但是实际上A此时可能已经并不需要发出请求了，B的很多资源就白白浪费了。

三次握手就可以解决上述问题。当B发过来的ACK+SYN包到达A时，A知道已过期，不回去确认，那么B收不到A的ACK报文，连接不会建立。

四次挥手：

1. 客户端发送FIN，32位序号假设为m。此时客户端进入FIN\_WAIT\_1.
2. 服务端收到后，发送一个ACK包（ack=m+1）。此时服务端进入CLOSE\_WAIT。客户端收到ACK包后，进入FIN\_WAIT\_2状态，等待服务端发送FIN包。
3. 服务端发送FIN包，进入LAST\_ACK，等待客户端的ACK包。
4. 客户端发送ACK包。客户端先是TIME\_WAIT状态，在2MSL后进入CLOSED。服务器端收到ACK报之后，进入CLOSED状态。  
   

为什么客户端发送最后一个ACK包，先是TIME\_WAIT，还要等2MLS？

MSL为 Maximum Segment Lifetime，即“报文最大生存时间”。

1. 设置这个的原因主要是怕最后一个ACK对方没收到。那么对方在超时后会重发第三次握手的FIN，主动关闭端收到重发的FIN包后，可以再发一个ACK应答包。
2. 还有就是等待该连接在网络上的所有报文都传输完毕，在处于TIME\_WAIT状态的时候，两端的端口都是不可用的，迟到的报文都会被弃用。这样可以使下一个连接中不会出现旧连接的报文段。