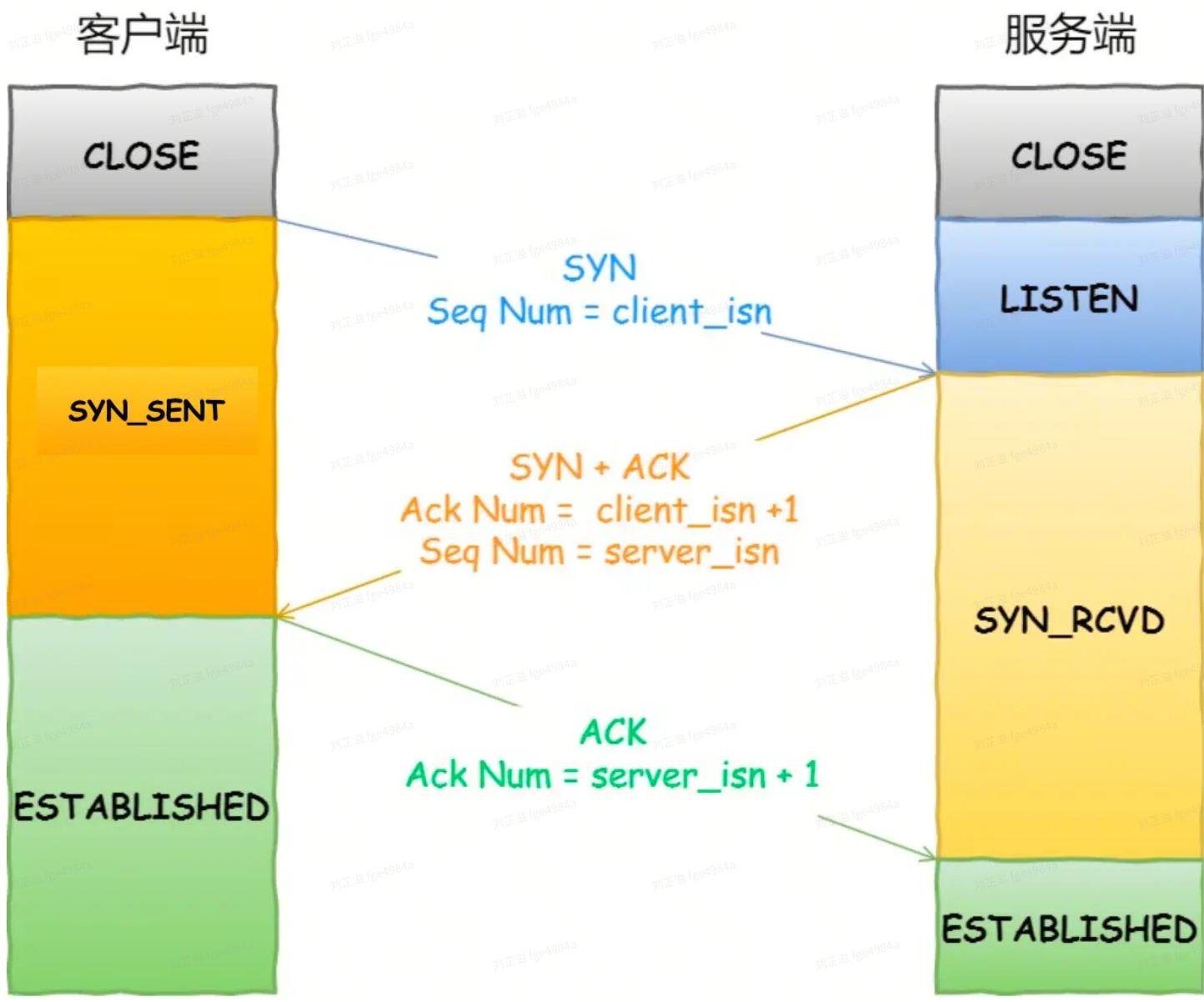


# TCP三次握手与四次挥手

## TCP三次握手与四次挥手

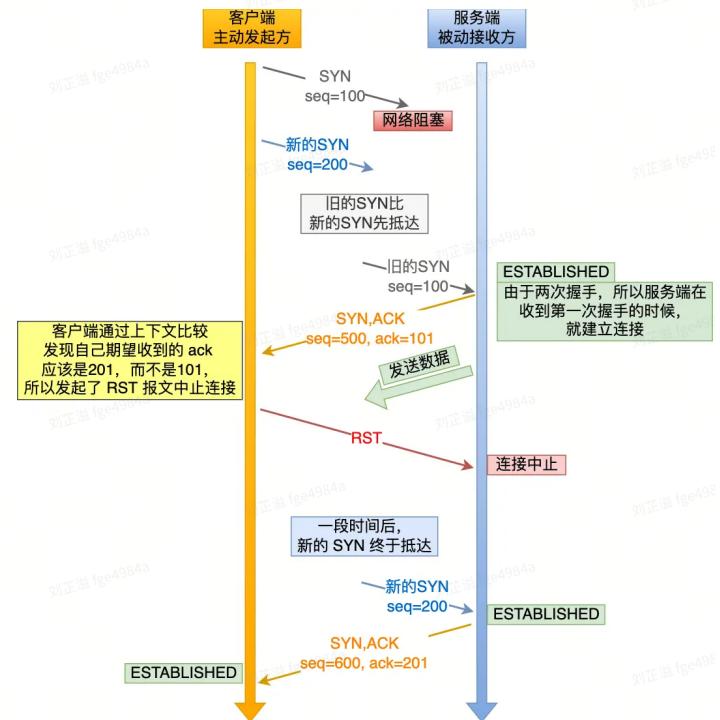
### TCP三次握手

- 第三次握手是可以携带数据的，前两次握手是不可以携带数据的。
- 一旦完成三次握手，双方都处于 **ESTABLISHED** 状态，此时连接就已建立完成，客户端和服务端就可以相互发送数据了。
- 服务器发送完 SYN-ACK 包，如果未收到客户端响应的确认包，也即第三次握手丢失。那么服务器就会进行首次重传，若等待一段时间仍未收到客户确认包，就进行第二次重传。**如果重传次数超过系统规定的最大重传次数，则系统将该连接信息从半连接队列中删除。**



# 原因

1. 避免历史连接：如果只有两次握手，接收到 SYN后，服务器就开始向客户端传递数据，直到客户端返回RST，才中断历史连接；浪费服务器资源。

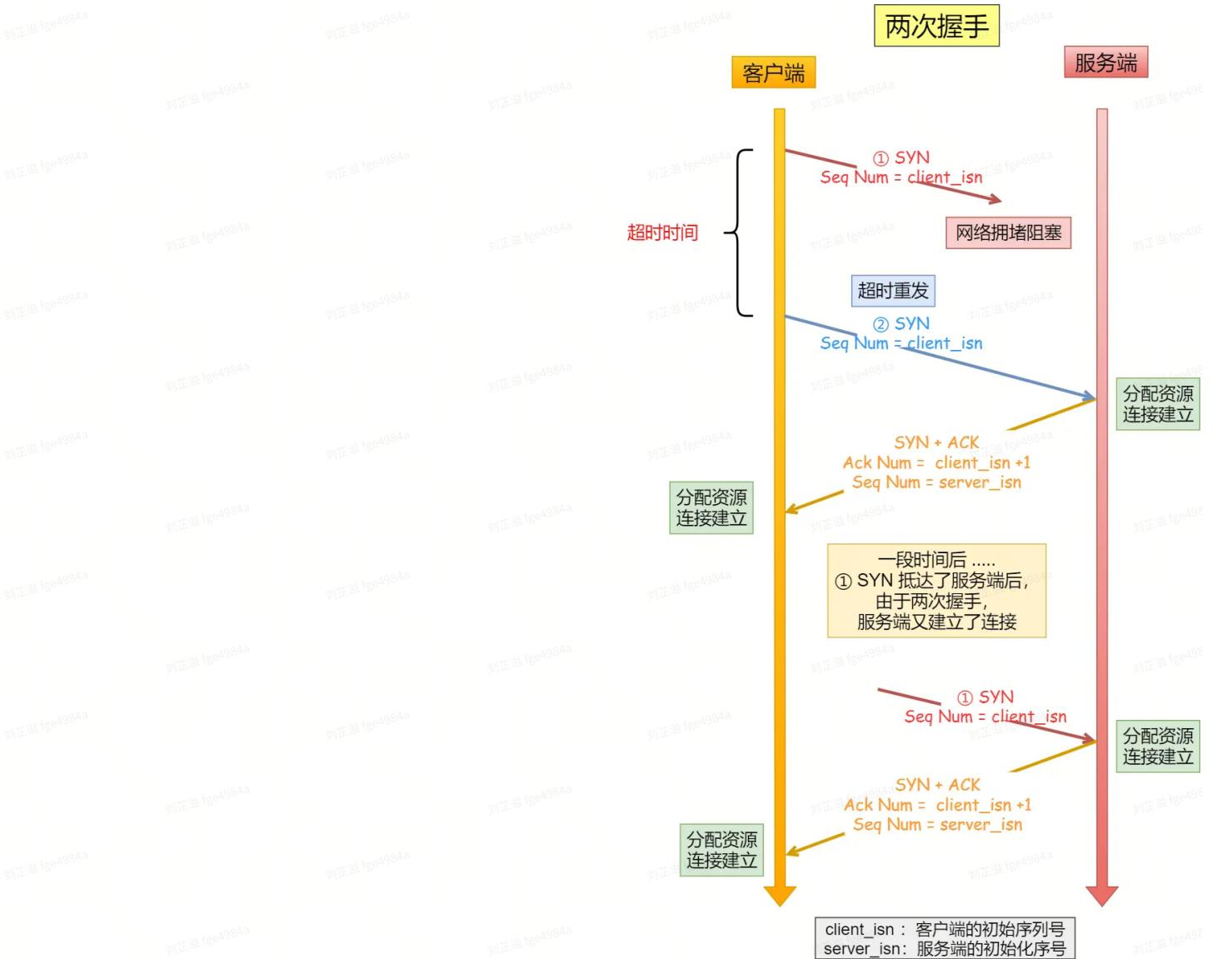


1. 同步双方的序列号：两次握手只保证了一方的初始序列号能被对方成功接收，没办法保证双方的初始序列号都能被确认接收。

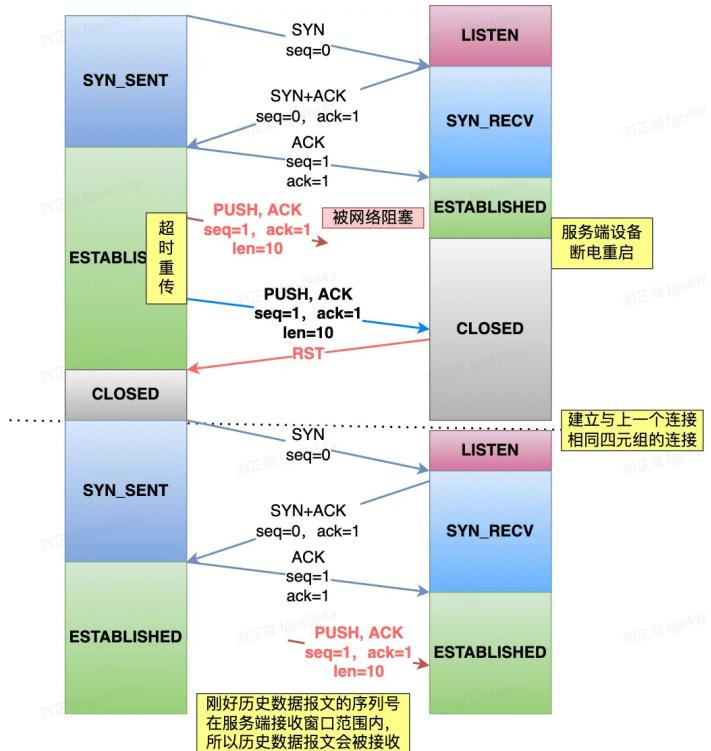
2. 序列号作用：

- 接收方可以去除重复的数据；
- 接收方可以根据数据包的序列号按序接收；
- 可以标识发送出去的数据包中，哪些是已经被对方收到的（通过 ACK报文中的序列号知道）；

1. 初始序列号不同，防止历史报文被下一个相同四元组的连接接收

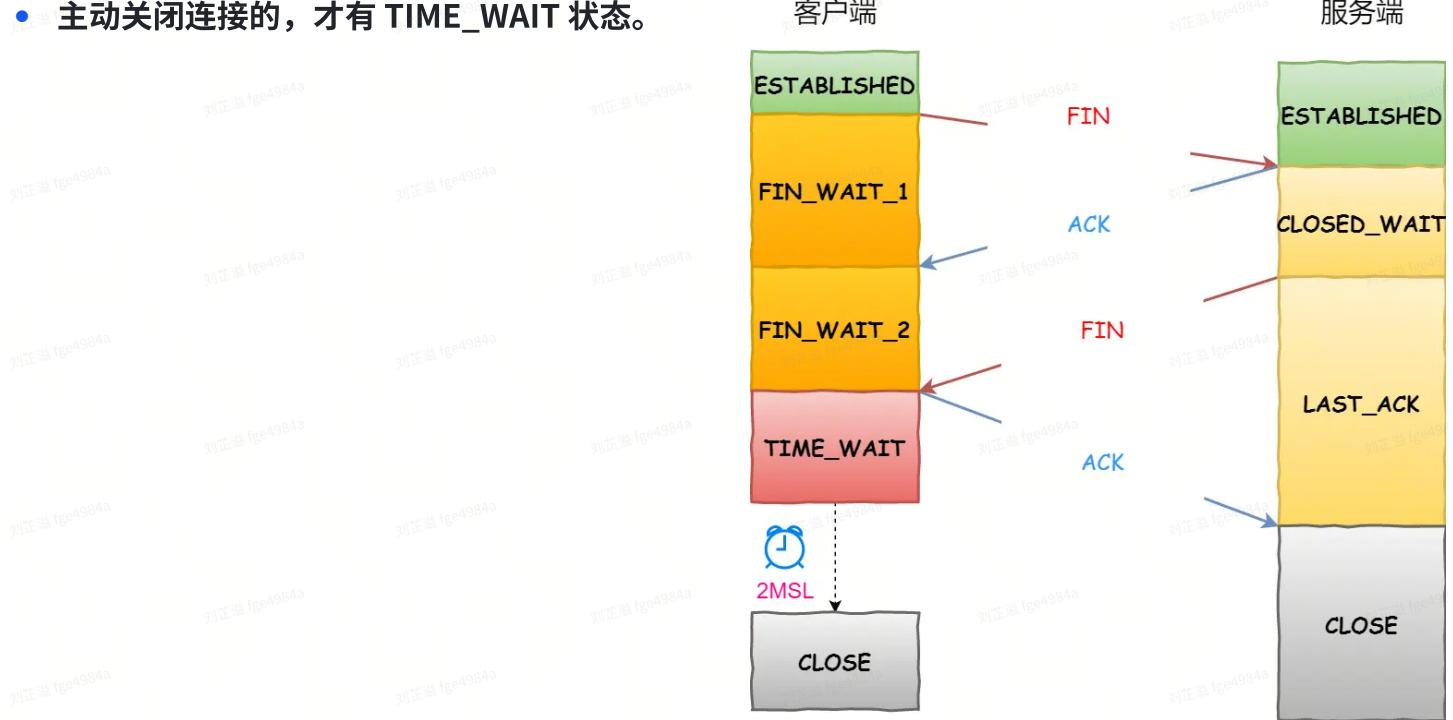


**1. 避免资源浪费：**两次握手时，服务器收到SYN即建立连接。如果客户端发送的SYN报文在网络中阻塞了，重复发送多次SYN报文，那么服务端在收到请求后就会建立多个冗余的无效链接，造成不必要的资源浪费。



## TCP四次挥手

- 主动关闭连接的，才有 TIME\_WAIT 状态。

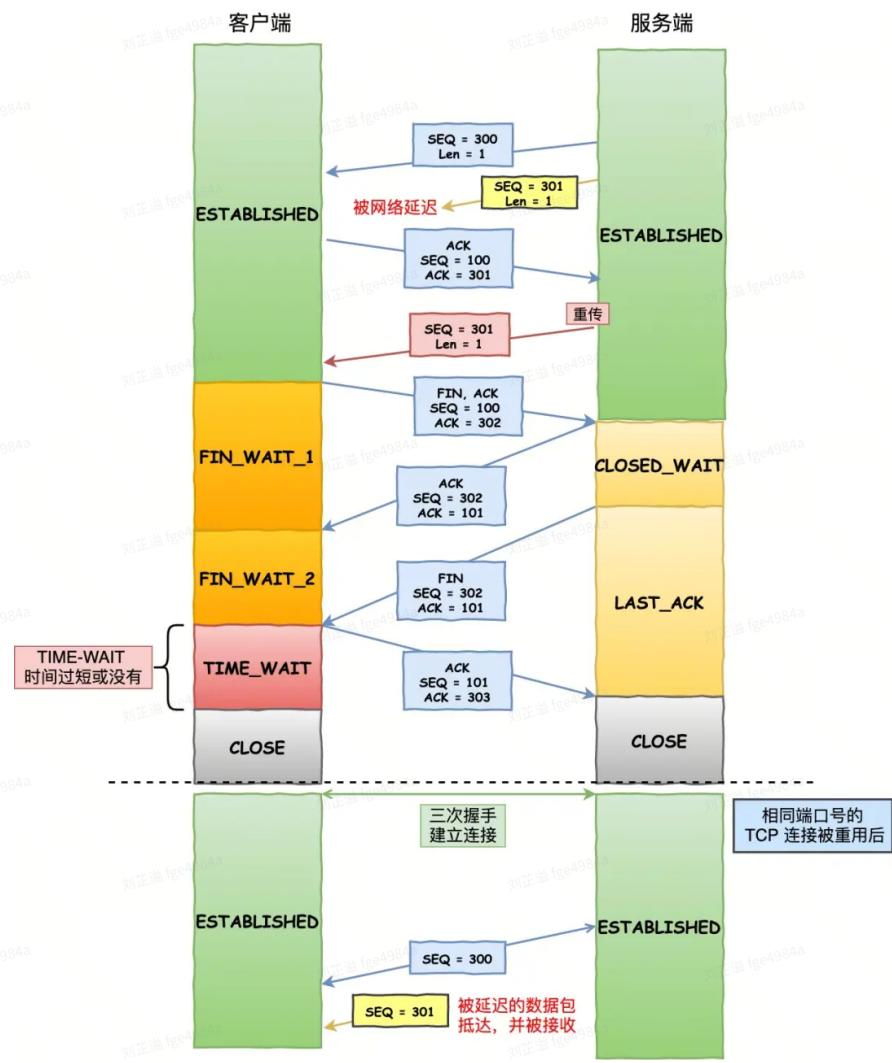


- 由于 TCP 的半关闭 (half-close) 特性，TCP 提供了连接的一端在结束它的发送后还能接收来自另一端数据的能力。
- 任何一方都可以在数据传送结束后发出连接释放的通知，待对方确认后进入半关闭状态。当另一方也没有数据再发送的时候，则发出连接释放通知，对方确认后就完全关闭了TCP连接。

## 为什么存在TIME\_WAIT状态

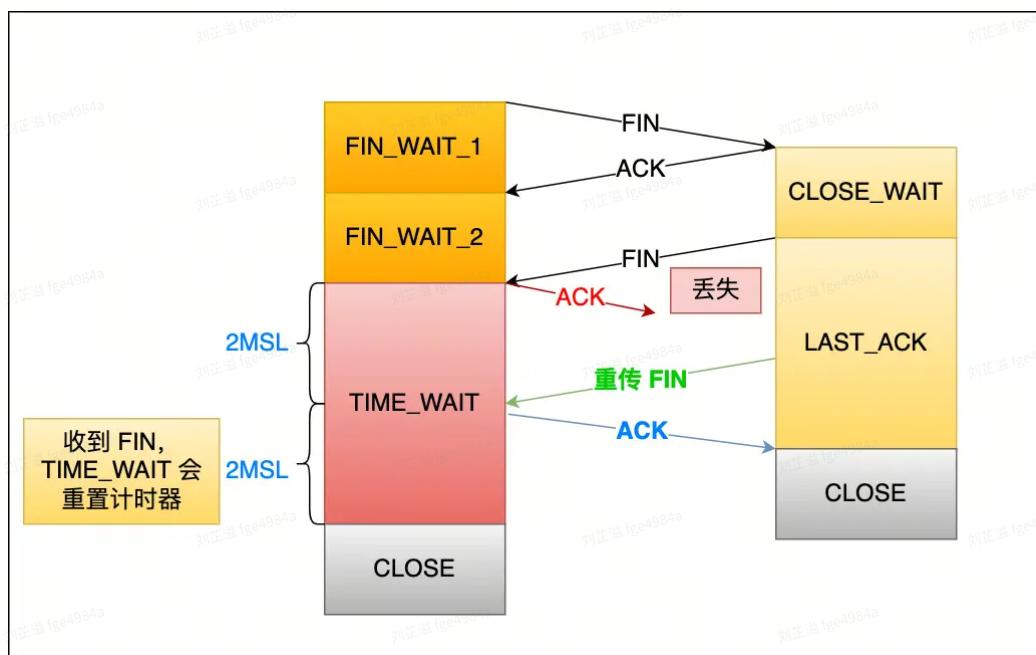
### 1. 防止历史连接中的数据，被后面相同四元组的连接错误的接收

- TIME\_WAIT 状态会持续 2MSL 时长，这个时间足以让两个方向上的数据包都被丢弃，使得原来连接的数据包在网络中都自然消失，再出现的数据包一定都是新建立连接所产生的。



## 1. 保证被动关闭连接的一方可以正确被关闭

- 如果TIME\_WAIT时间过短，服务端重传ACK，此时客户端已经关闭返回RST报文，服务端收到这个RST并将其解释为一个错误（Connection reset by peer）
- 等待足够的时间以确保最后的ACK能让被动关闭方接收，从而帮助其正常关闭。



# accept、connect发生在三次握手的时机

- 客户端 connect 成功返回是在第二次握手，服务端 accept 成功返回是在三次握手成功之后。

