



三次握手

三次握手: (我要和你建立链接, 你真的要和我建立链接么, 我真的要和你建立链接, 成功)

第一次握手: 客户端发送 `syn` 包 (`syn=x`) 到服务器, 并进入 `SYN_SEND` 状态, 等待服务器确认;

第二次握手: 服务器收到 `syn` 包, 必须确认客户的 `SYN` (`ack=x+1`), 同时自己也发送一个 `SYN` 包 (`syn=y`), 即 `SYN+ACK` 包, 此时服务器进入 `SYN_RECV` 状态;

第三次握手: 客户端收到服务器的 `SYN+ACK` 包, 向服务器发送确认包 `ACK` (`ack=y+1`), 此包发送完毕, 客户端和服务器进入 `ESTABLISHED` 状态, 完成三次握手。

握手过程中传送的包里不包含数据, 三次握手完毕后, 客户端与服务器才正式开始传送数据。理想状态下, `TCP` 连接一旦建立, 在通信双方中的任何一方主动关闭连接之前, `TCP` 连接都将被一直保持下去。

四次挥手

四次挥手：(我要和你断开链接；好的，断吧。我也要和你断开链接；好的，断吧)：

第一次挥手：客户端主动关闭方发送一个 **FIN**，用来关闭客户端到服务端的数据传送，也就是客户端告诉服务端：我已经不会再给你发数据了 (当然，在 **fin** 包之前发送出去的数据，如果没有收到对应的 **ack** 确认报文，客户端依然会重发这些数据)，但是，此时客户端还可以接受数据。

第二次挥手：服务端收到 **FIN** 包后，发送一个 **ACK** 给客户端，确认序号为收到序号 + 1 (与 **SYN** 相同，一个 **FIN** 占用一个序号)。

第三次挥手：服务端发送一个 **FIN**，用来关闭服务端到客户端的数据传送，也就是告诉客户端，我的数据也发送完了，不会再给你发数据了。

第四次挥手：客户端收到 **FIN** 后，发送一个 **ACK** 给服务端，确认序号为收到序号 + 1，至此，完成四次挥手。

常见问题

1.第 3 次握手失败会怎么办？

第三次失败，只有客户端处于成功状态 (因为第 2 次服务器返回了 **ACK**)，服务器端没有接收到客户端的 **ACK**。

这要分几种情况讨论：

客户端发出的 **ACK** 丢失了，发出的 下一个数据包 没有丢失，则服务端接收到下一个数据包 (这个数据包里也会带上 **ACK** 信息)，能够进入正常的 **ESTABLISHED** 状态

如果服务端和客户端都没有数据发送，或者服务端想发送数据 (但是发不了，因为没有收到客户端的 **ACK**)，服务器都会有定时器发送第二步 **SYN+ACK** 数据包，如果客户端再次发送 **ACK** 成功，建立连接。

如果一直不成功，服务器肯定会有超时设置，超时之后会给客户端发 **RTS** 报文，进入 **CLOSED** 状态，防止 **SYN** 洪泛攻击。

2.为什么 TCP 链接需要三次握手，两次不可以么，为什么？

为了防止已失效的链接请求报文突然又传送到了服务端，因而产生错误。

客户端发出的连接请求报文并未丢失，而是在某个网络节点长时间滞留了，以致延误到链接释放以后的某个时间才到达 **Server**。这是，**Server** 误以为这是 **Client** 发出的一个新的链接请求，于是就向客户端发送确认数据包，同意建立链接。若不采用 “三次握手”，那么只要 **Server** 发出确认数据包，新的链接就建立了。由于 **client** 此时并未发出建立链接的请求，

所以其不会理睬 Server 的确认，也不与 Server 通信；而这时 Server 一直在等待 Client 的请求，这样 Server 就白白浪费了一定的资源。若采用“三次握手”，在这种情况下，由于 Server 端没有收到来自客户端的确认，则就会知道 Client 并没有要求建立请求，就不会建立链接。

3.为什么连接的时候是三次握手，关闭的时候却是四次握手？

TCP 是全双工模式，关闭连接时，当主机 B 收到主机 A 的 FIN 报文时，仅仅表示主机 A 不再发送数据了但是还能接收数据。此时，主机 B 也未必全部数据都发送给 A 了，所以 B 可以立即 close；也可以发送一些数据给 A 后，再发送 FIN 报文给对方来表示同意现在关闭连接，因此，主机 BACK 和 FIN 一般都会分开发送。