1. TCP头部

六个二进制位：

ACK：确认位，1 🡪 确认包 0 🡪 非确认包

SYN：请求位，1 🡪 建立连接的请求包 0 🡪 其它的包

FIN： 完成位 1 🡪 释放链接的包

1. 三次握手

第一次握手：客户端发送一个SYN包

一个 SYN包就是仅SYN标记设为1的TCP包

第二次握手，server 端返回的包：SYN/ACK包

SYN/ACK包是仅SYN 和 ACK 标记为1的包.

第三次握手：ACK包

ACK包就是仅ACK 标记设为1的TCP包

连接建立后TCP连接的每个包都会设置ACK位。

1. 四次挥手断开连接

第一次：客户端发起一个ACK/FIN包

ACK/FIN 包, ACK 和FIN 标记设为1

由于连接还没有关闭, FIN包总是打上ACK标记. 没有ACK标记而仅有FIN标记的包不是合法的包，并且通常被认为是恶意的。

第二次：Server端发送一个ACK包  
(Client) <– ACK <– (Server)

Server端收到FIN报文时，很可能并不会立即关闭SOCKET（套接字），所以只能先回复一个ACK报文，告诉Client端，"你发的FIN报文我收到了"。

第三次：Server端断开连接，发送一个ACK/FIN包  
 (Client) <– ACK/FIN <– (Server)  
第四次：客户端发送一个ACK包给Server ，服务器就关闭连接，客户端等待一段时间后没有收到回复说明Server端关闭连接，此时也关闭连接。

(Client) –> ACK –> (Server)

1. 四次挥手的状态变化

FIN\_WAIT\_1：向对方发送了FIN报文，此时该SOCKET即进入到FIN\_WAIT\_1状态

FIN\_WAIT\_2：而当对方回应ACK报文后，则进入到FIN\_WAIT\_2状态

TODO： netstat

TODO： socket

TODO：tcp的流量控制

CLOSE\_WAIT：服务器端回应一个ACK报文给对方，此时则进入到CLOSE\_WAIT状态。接下来需要考虑的事情是查看服务器端是否还有数据发送给对方

LAST\_ACK：服务器端发送FIN报文后等待对方发送一个ACK报文时的状态

TIME\_WAIT：表示收到了对方的FIN报文，并发送出了ACK报文

CLOSED: 表示连接中断。