# 复习超时重传

1. IP + PORT 的作用，以及约束
2. 五元组-传输层（端对端）通信通道的唯一标识（复合主键）
3. UDP
   1. 不可靠、无连接、面向数据包
   2. 有接收缓冲区，没有发送缓冲区
   3. Send 成功代表的意思，数据被成功发送到网络上
   4. Receive 会阻塞，一直到收到数据为止，除非接收缓冲区中有数据

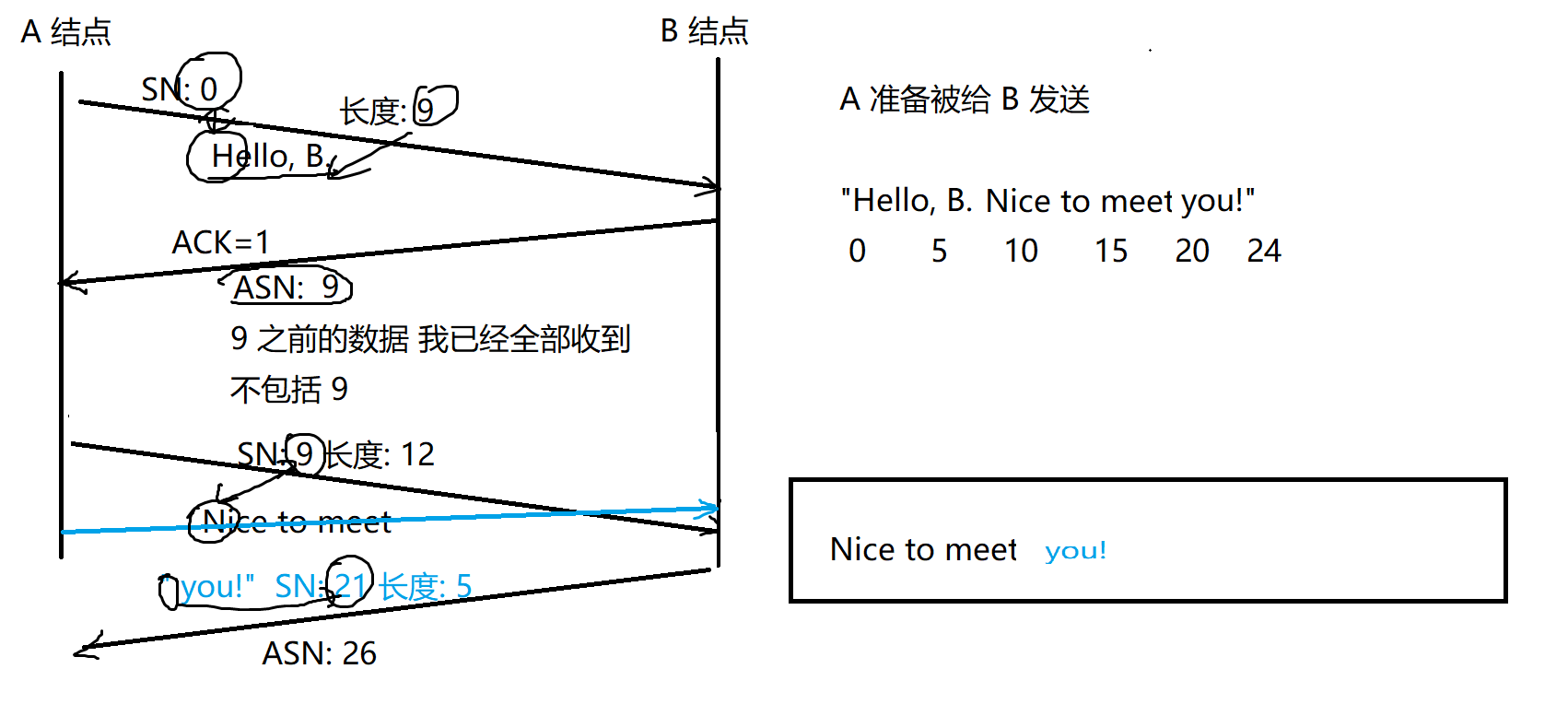
TCP 的目标是构建**可靠**的传输层协议

可靠的含义：

1. 数据保证正确 by 校验和
2. 发送完数据之后，希望知道对方有没有收到 By 确认应答机制（SN + ASN）
3. 接收方可以按序接收数据 By SN 进行排序
4. 接收方可以去重数据 By SN 进行去重

**确认应答**

**对于 TCP 连接中任何 Segment 都需要应答，除了 ACK 本身（应答（无数据）本身不需要应答）**



超时重传-基于确认应答

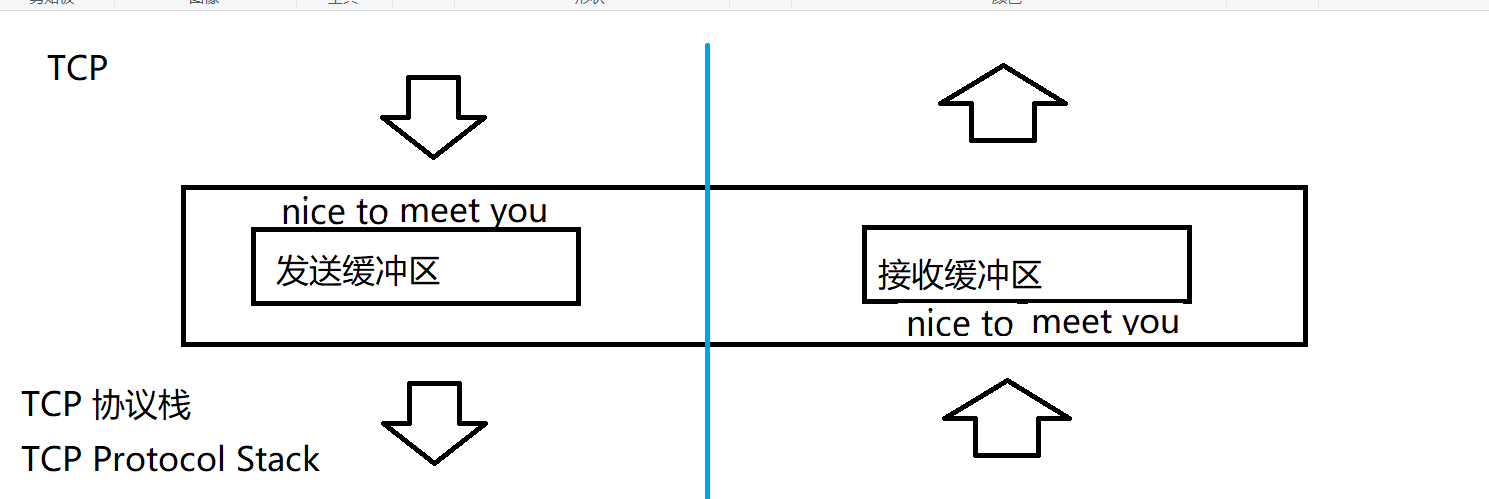
如果没有收到应答，对于接收方来说就进行重传

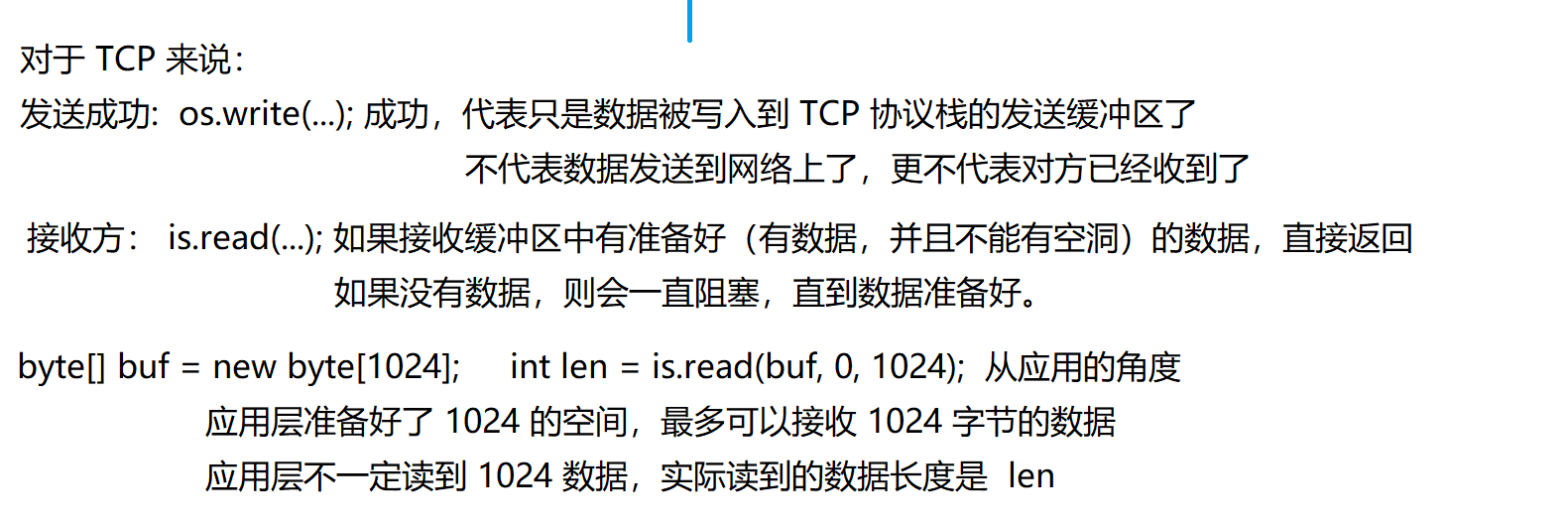
1. 超时时间多少合适？
2. 直接重传
3. 实际上可能是由两种情况引起 a) 对方没有收到 b) 应答没有收到；对于接收方来说不做区分

# 发送和接收缓冲区

TCP 有发送缓冲区 和 接收缓冲区

1. 为什么要有发送缓冲区，其中一个原因，就是我发送的数据有可能需要重传，所以TCP 需要暂时保存这些数据。
2. 为什么要有接收缓冲区，因为数据可能是无序到达的，还因为数据到达后，应用层不是马上取走。





# 连接管理-可靠性为基础

问题：

1. 为什么需要 TCP 连接?

因为这套机制是以不同五元组描时的通道分别管理的。

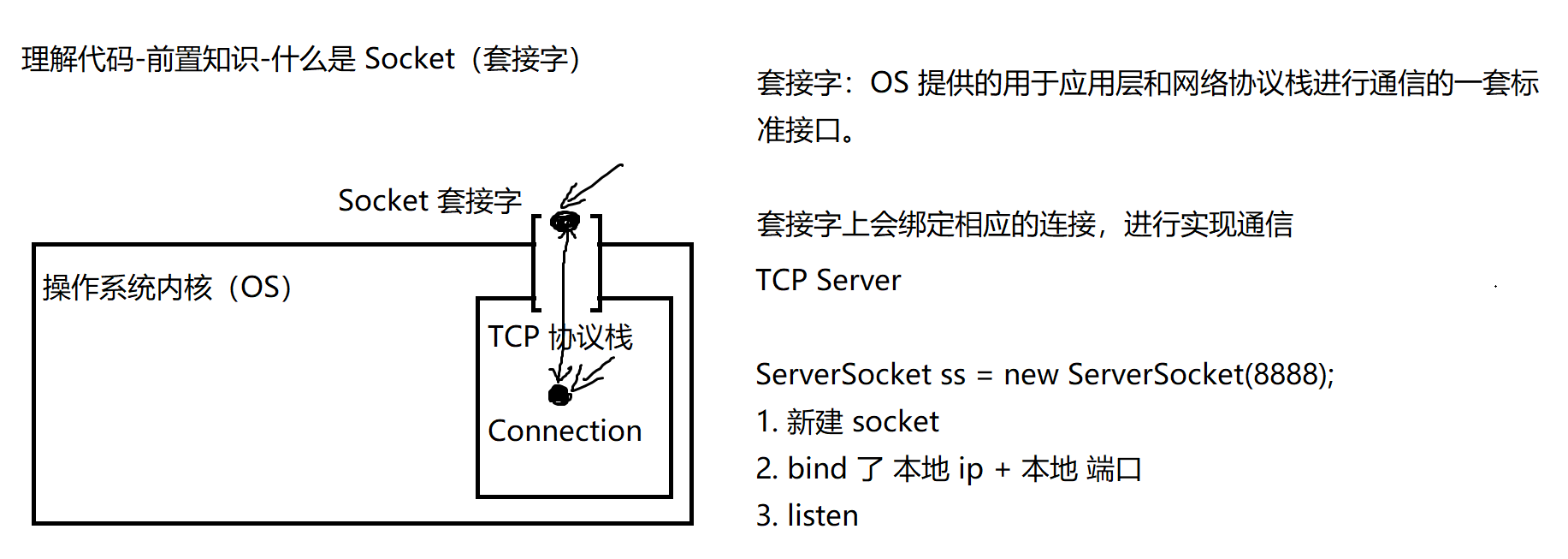
1. 什么是连接（Connection）

抽象层面上：五元组唯一标识的一条通信通道

OS 层面上：一个 Connection 对象，描述五元组信息 + 各种管理信息（SN/ASN + 缓冲区信息）

OS 通过 List / Map 把 Connection 对象组织起来

Socket 和连接的关系-从应用层代码角度理解



## 连接管理（主要是 TCP 协议栈的视角，结合应用层代码理解）

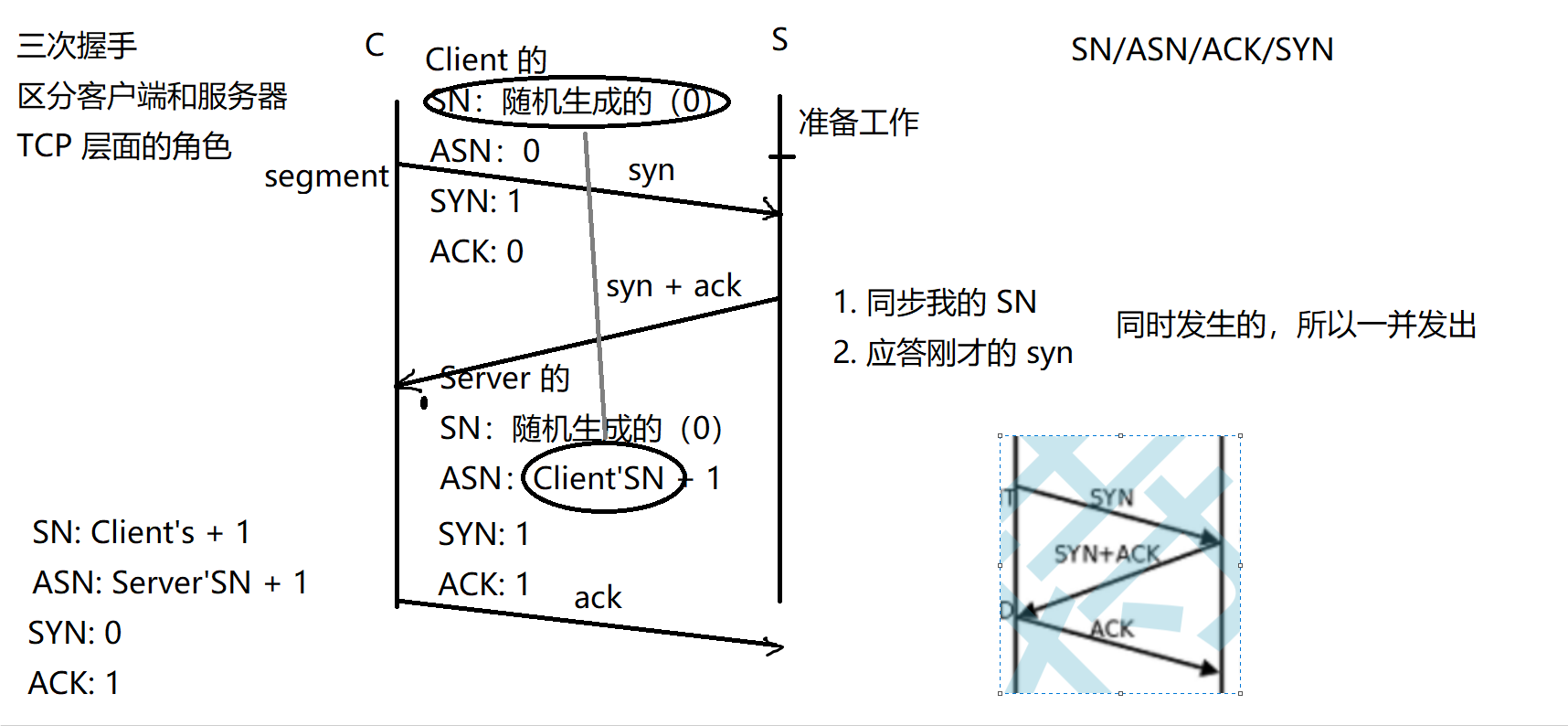
需要有创建连接的过程和销毁连接的过程

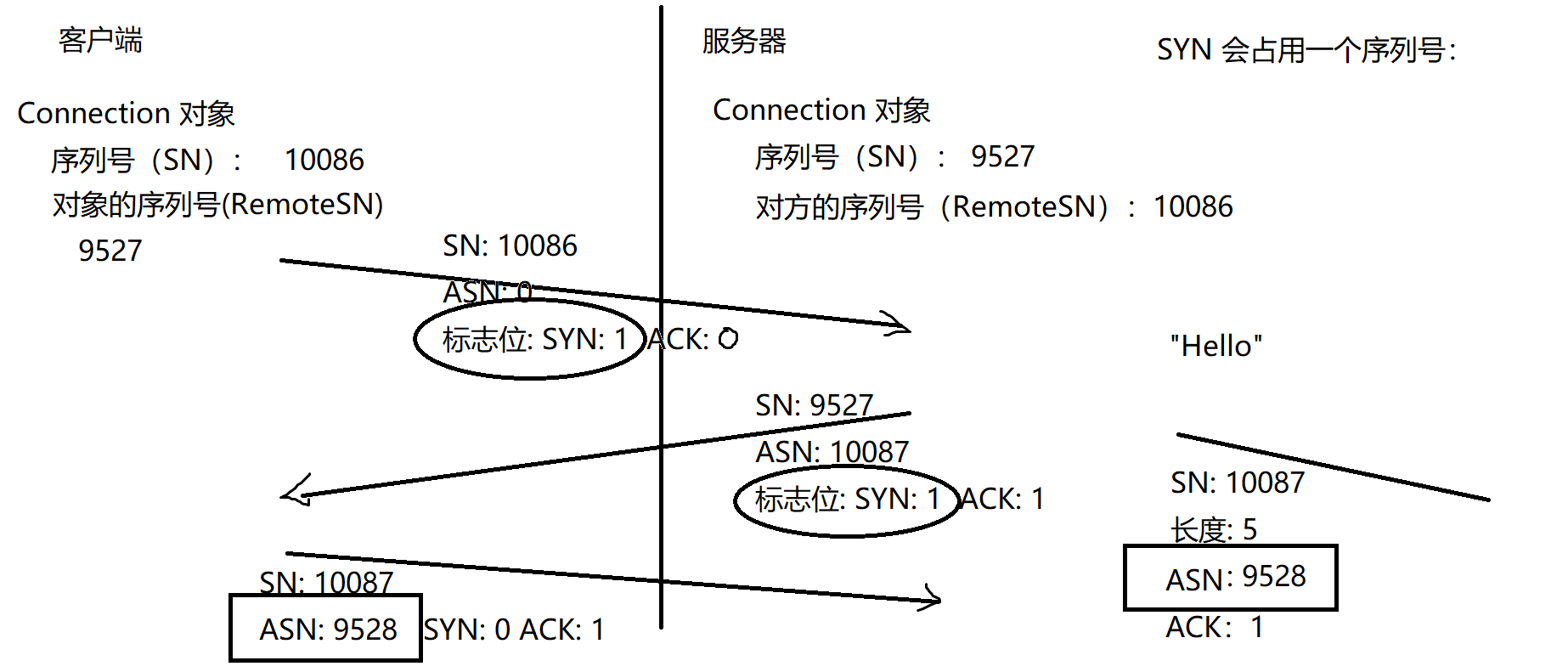
### 三次握手建立 TCP 连接

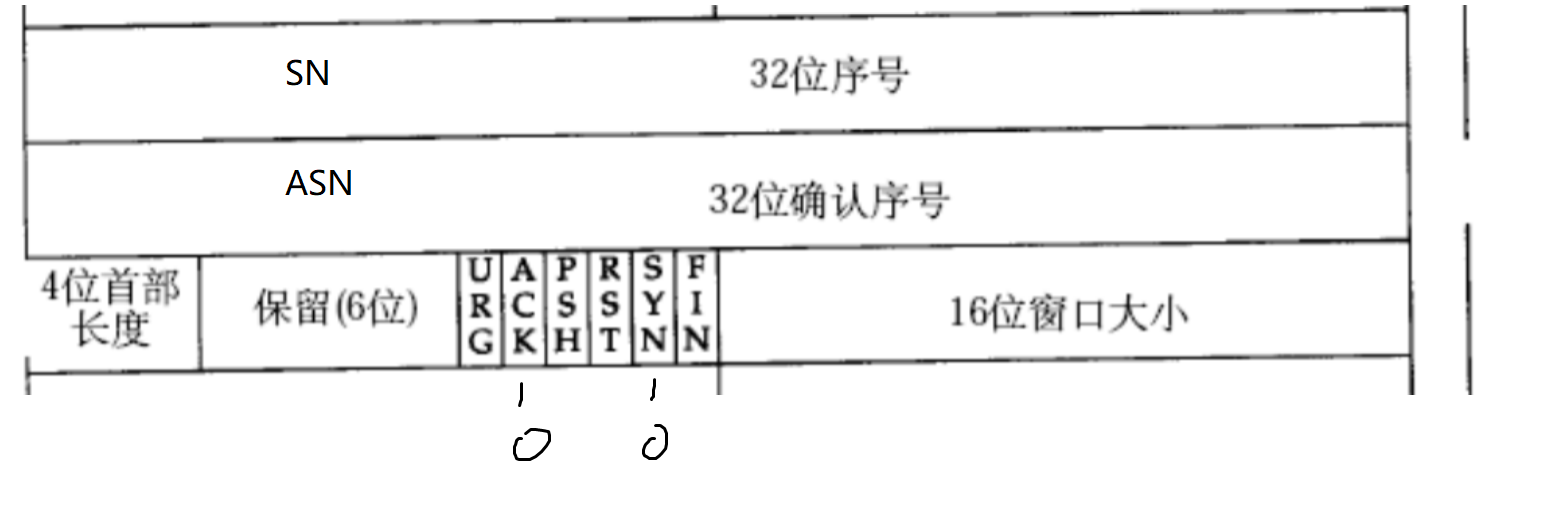
SYN：表示想建立连接，进行信息的同步

ACK：确认应答用的，表示确认之前收到的 segment，配合确认序列号（ASN）使用

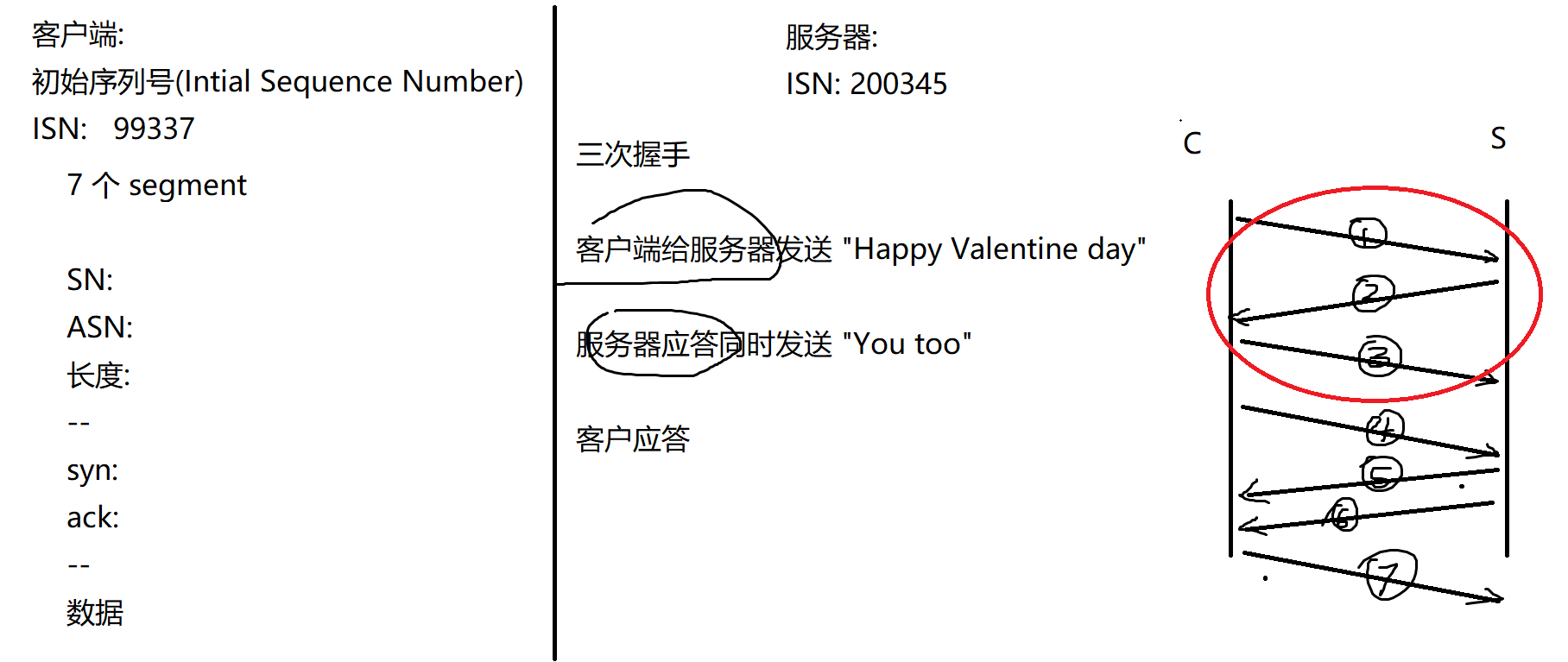
空的 ACK 不需要再次应答





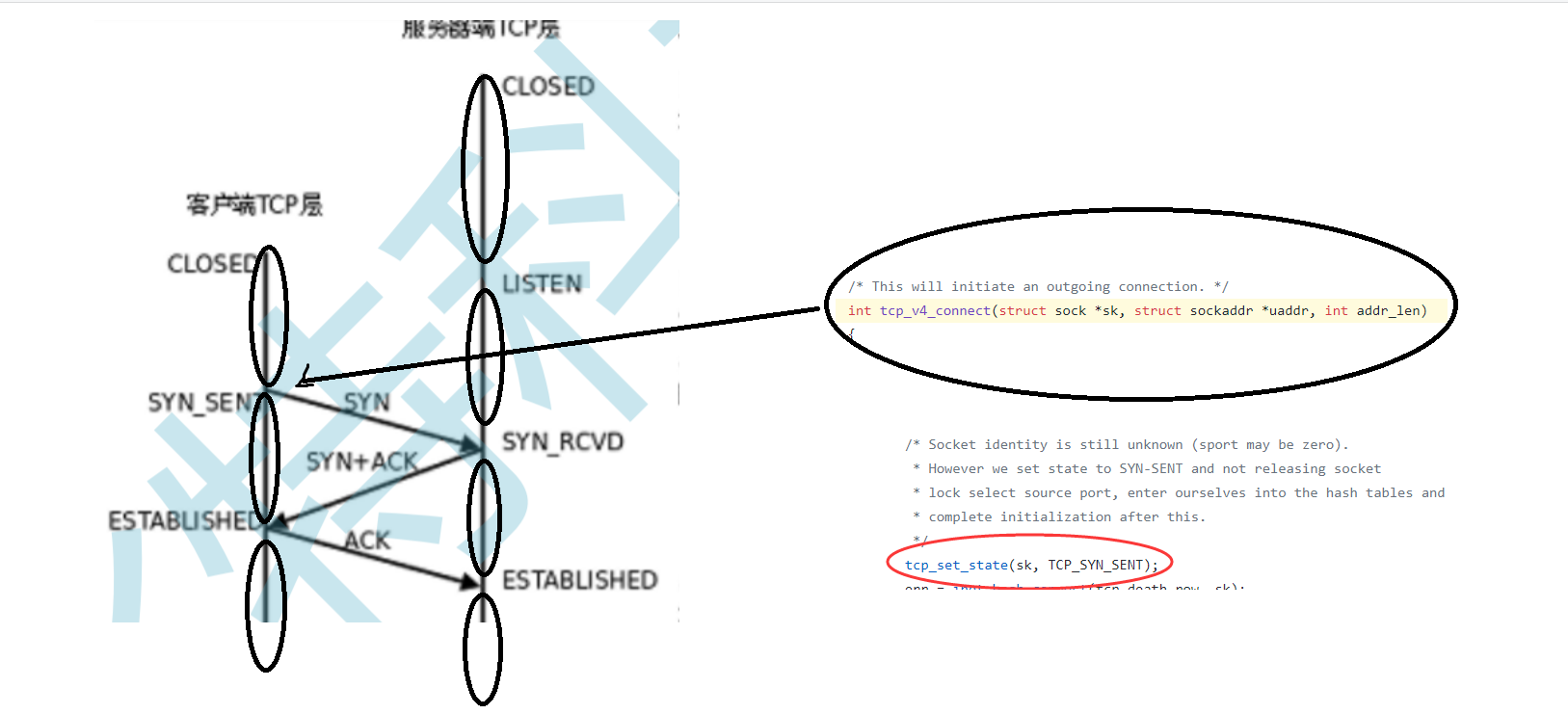


### 练习

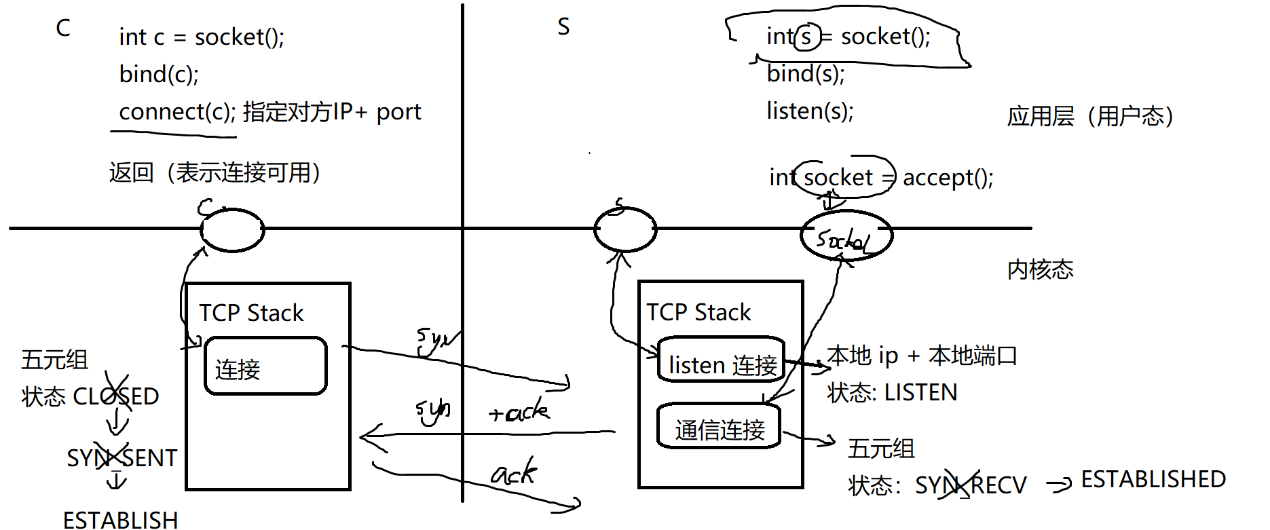


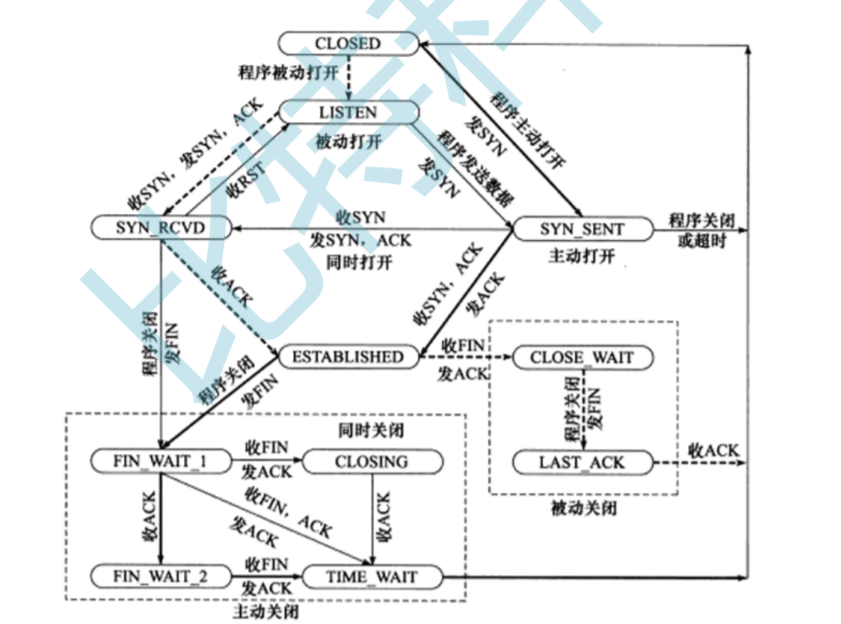
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C' ISN | 99337 |  |  |  |  |  |  |  |
| S' ISN | 200345 |  |  |  |  |  |  |  |
| 编号 | 方向 | SN | ASN | 长度 | SYN | ACK | 数据 |  |
| 1 | C->S | 99337 | 0 | 1 | 1 | 0 | "" | 三次握手 |
| 2 | S->C | 200345 | 99338 | 1 | 1 | 1 | "" |
| 3 | C->S | 99338 | 200346 | 0 | 0 | 1 | "" |
|  |  |  |  |  |  |  |  |  |
| 4 | C->S | 99338 | 200346 | 19 | 0 | 1 | "Happy Valentine day" |  |
| 5 | S->C | 200346 | 99357 | 0 | 0 | 1 | "" |  |
|  |  |  |  |  |  |  |  |  |
| 6 | S->C | 200346 | 99357 | 7 | 0 | 1 | "You too" |  |
| 7 | C->S | 99357 | 200353 | 0 | 0 | 1 | "" |  |

### TCP 的状态转移

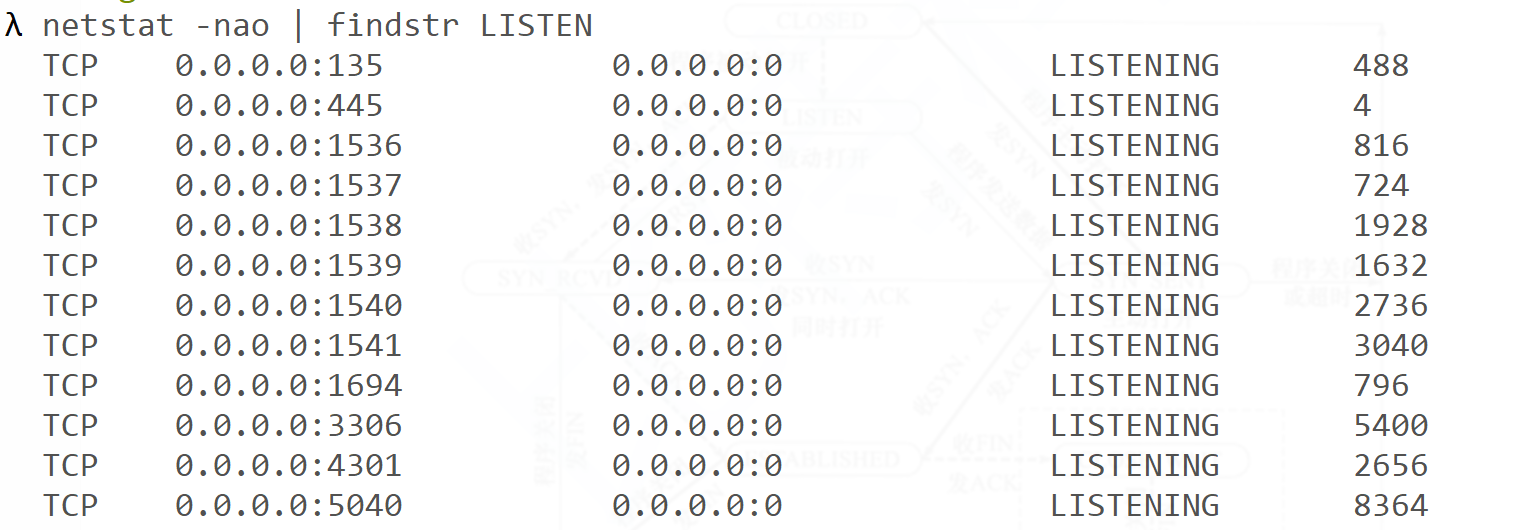


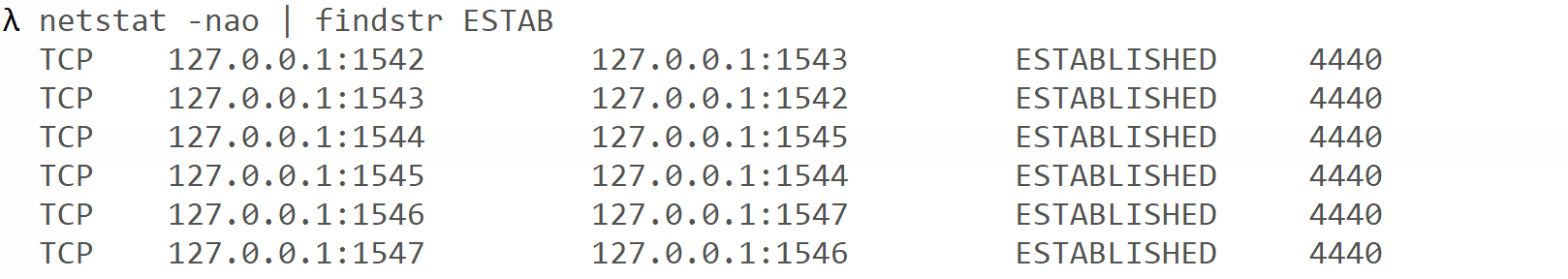
### 和应用层代码结合





### 工具





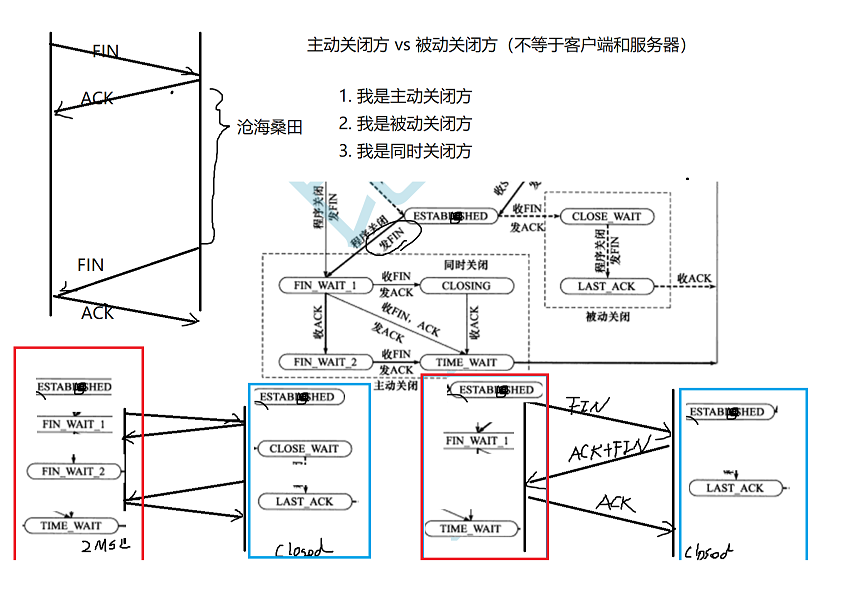
### 总结下常见的面试题：

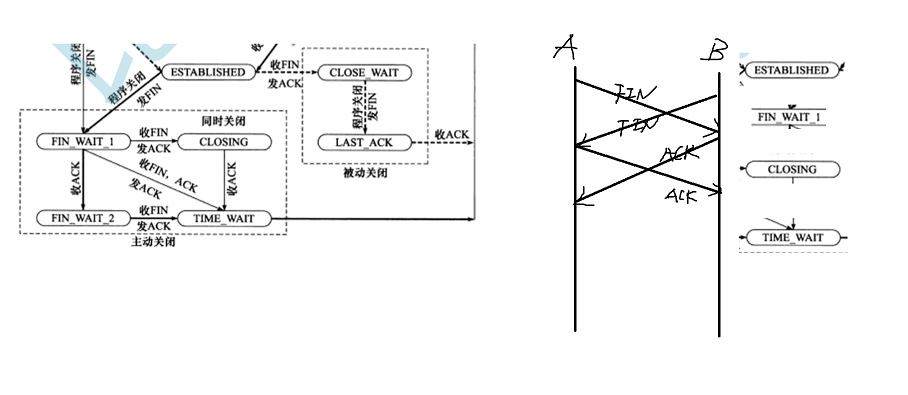
1. 描述三次握手的过程？
   1. 标志位：SYN + ACK
   2. 状态转移
   3. SN + ASN
   4. 结合应用层代码
2. 为什么是三次？
   1. 为什么不能两次？C 发 SYN，S 回 ACK，S 发 SYN，C 回 ACK。不满足确认应答
   2. 为什么不想四次？能少尽量少，因为 S 回 ACK 和 S 发 SYN 肯定是同时的，所以合并了
3. 建立连接的目的是什么？
4. 为什么要有连接？
5. 什么是连接？
6. 建立连接主要是为了同步 ISN（初始序列号）
7. 为什么 ISN 序列号必须是随机的？如果 ISN 都是 0，是不是就不需要同步了？

ISN 如果是确定的，可能会被伪造 segment 进行攻击

### 四次挥手释放 TCP 连接

A 结点 和 B 结点（没有客户端服务器之说）





# 总结

连接管理（重点）：

0. 什么是连接，为什么要有连接。核心点都是可靠性

1. 三次握手（重点）

1. 发送 segment 中的 tcp header 字段如何填充

1. 标志位（重点）

2. 序列号 + 确认序列号（次重点）

2. 状态转移过程（次重点）

3. 和应用层代码的结合（重在理解）

2. 四次挥手（次重点）