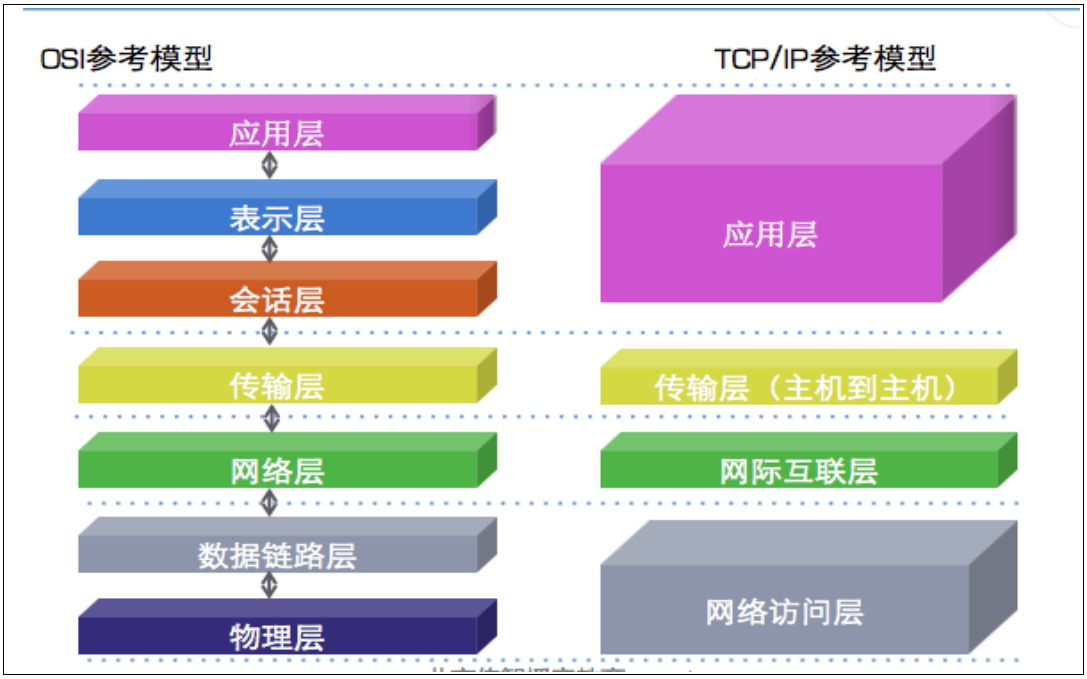
Socket

一、网络模型

●网络模型的出现，由于网络是一个比较复杂的系统，所以需要将网络进行分层，每一层规范相应的职能。正如一个公司越来越大时，会进行改革，也就是部门划分，员工制度规范。网络也是一样，为了维持网络的可持续发展，所以网络模型就出来。

●网络模型有OSI参考模型和TCP/IP参考模型



●OSI参考模型各层职能

1.物理层：主要定义物理设备标准，如网线的接口类型、各种传输介质的传输速率等。主要作用是传输比特流（就是由1、0转化为电流强弱来进行传输，到达目的地后再转化为1、0，也就是常说的数模与模数转换）。这一层的数据叫做比特（bit），主要设备：集线器、网线

2.数据链路层：主要将从物理层接收的数据进行MAC地址的封装与解封装。常把这一层的数据叫做帧，主要设备：网卡，交换机

3.网络层：选择合适的网间路由和交换结点， 确保数据及时传送，将从下层接收到的数据进行IP地址的封装与解封装，称为**IP协议**。常把这一层数据叫做数据包，主要设备：路由器。

4.传输层：定义了一些传输数据的协议和端口，如TCP、UDP协议，主要将从下层接收的数据进行分段和传输，到达目的地址后再进行重组，以往把这一层数据叫做段。

5.会话层：通过传输层建立数据传输通路。在系统之间发起会话或者接受会话请求（设备之间需要互相认识）

6.表示层：主要是进行对接收的数据进行解释、压缩与解压缩等，即把计算机能够识别的东西转化成人能够识别的东西（如图片、声音等）

7.应用层：主要是一些终端的应用，比如说FTP（各种文件下载）、浏览器、QQ等，可以将其理解为在电脑屏幕上可以看到的东西，也就是终端应用。

二、TCP/IP协议

网络协议即网络中（包括互联网）传递、管理信息的一些规范。如同人与人之间相互交流是需要遵循一定的规矩一样，计算机之间的相互通信需要共同遵守一定的规则，这些规则就称为网络协议。

TCP/IP协议是网络的基础，是Internet的语言，可以说没有TCP/IP协议就没有互联网的今天。

IP是网络层,TCP是传输层,UDP也是传输层的

●什么是 TCP 和 UDP 及区别

>TCP（传输控制协议）

建立连接，形成传输数据的通道

在连接中进行大数据传输（数据大小不收限制）

通过三次握手完成连接，是可靠协议，安全送达

必须建立连接，效率会稍低

>UDP（用户数据报协议）

将数据及源和目的封装成数据包中，不需要建立连接

每个数据报的大小限制在 64K 之内

因为无需连接，因此是不可靠协议

不需要建立连接，速度快

三、网络通讯要素

■IP地址（唯一标示网络设备的）：

- 网络中设备的标示
- 不易记忆，可以用主机名
- 本地回环地址：127.0.0.1 主机名：localhost

■端口号(定位程序)

- 用于标示进程的逻辑地址，不同进程的标示
- 有效端口：0~65535，其中 0~1024 由系统使用或者保留端口，开发中不要使用 1024 以下的端口

■传输协议（用什么样的方式进行交互）

- 通讯的规则
- 常见协议：TCP、UDP

■URL（统一资源定位） http://ip:80/文件路径

http 是数据传输格式协议，tcp 是数据传输方式，tcp 相当于邮寄信封或者是打电话,http 相当于信或者打电话是英语还是国语沟通

●Telnet 的使用可以查看服务是否开启

>telnet 127.0.0.1 8888

三、socket(套接字)

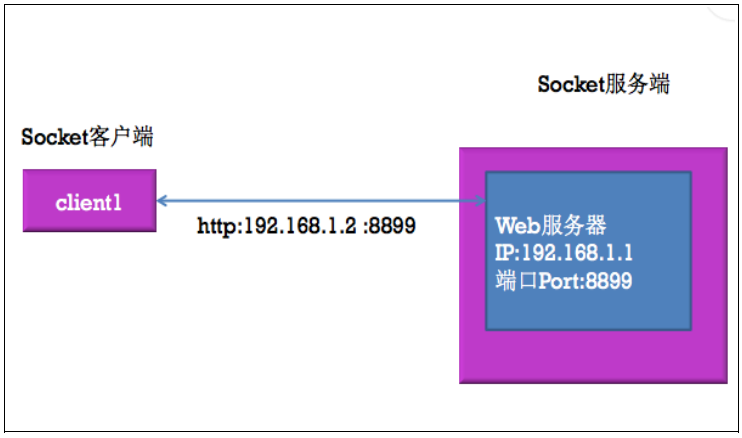
●Socket 就是为网络服务提供的一种机制

●通信的两端都是 Socket

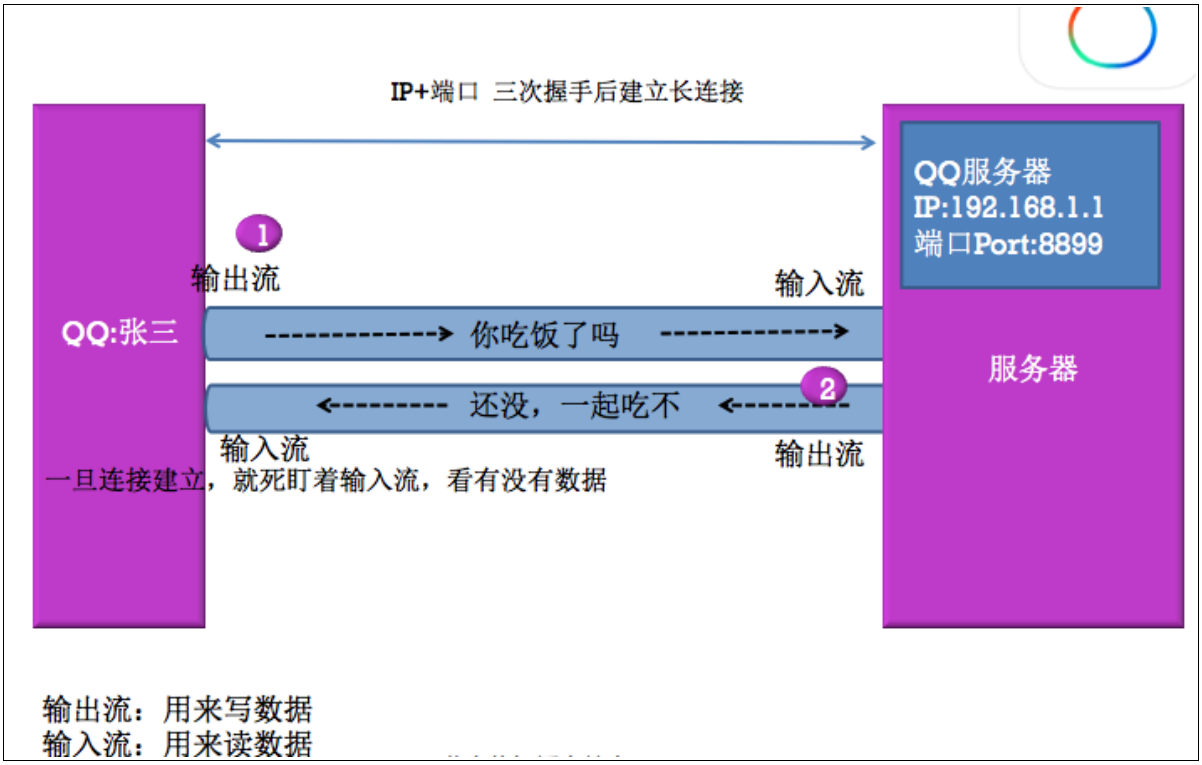
●网络通信其实就是 Socket 间的通信

●数据在两个 Socket 间通过 IO 传输

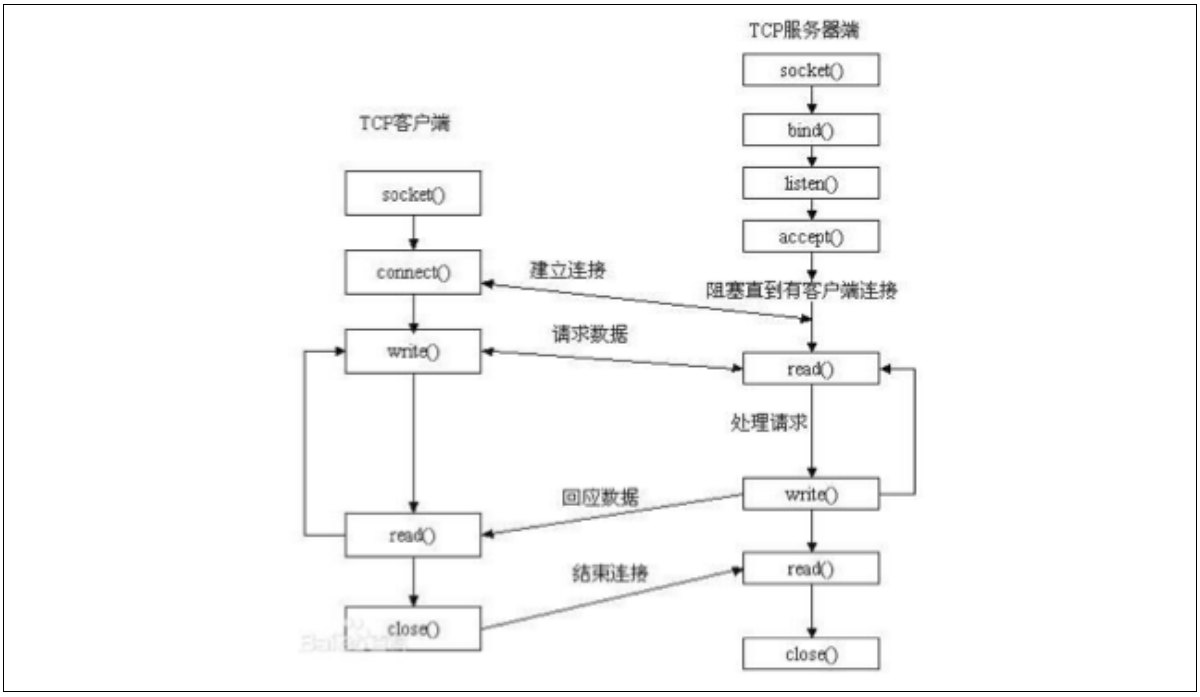
●HTTP 协议的传输实质就是 Socket 通信



●模拟QQ聊天实现的socket流程，详情看看PPT演示



四、Socket通信流程图



五、案例：简单聊天室

//开发过程中一般服务器不用自己写，这里自己使用已经写好的

//ios在要实现socket通过使用C语言

步骤：

>启动聊天服务器 Python chatserver.py

>连接到主机，分配输入输出流空间

>建立长连接CFStreamCreatePairWithSocketToHost

>将C语言的输入输出流转成OC对象

>设置代理监听输入输出流的状态
>添加到主运行循环，否理代理不工作

```
[_inputStream scheduleInRunLoop:[NSRunLoop mainRunLoop] forMode:NSDefaultRunLoopMode];
```

>打开输入输出流
>登录发送

```
NSString *loginStr = [NSString stringWithFormat:@"iam:zhangsan"];
NSData *data = [loginStr dataUsingEncoding:NSUTF8StringEncoding];
[_outputStream write:data.bytes maxLength:data.length];
```

>数据读取

```
//建立个缓冲区来保存读取到的数据
uint8_t buffer[1024];
//返回实际读取的数据长度
NSInteger len = [_inputStream read:buffer maxLength:sizeof(buffer
//有数据
if (len > 0) {
    NSString *receiver =  [[NSString alloc] initWithBytes:buffer length:len
encoding:NSUTF8StringEncoding];
}
```

>断开连接要关闭输入输出流并移除主运行循环

补充：netstat -an -p tcp |grep 12345 **查看所有开放的端口查看连接状态**

六、GCDAsyncSoocket 框架

- GCDAsyncSoocket 是能 c 语言 socket 编程的封装，是面向对象
- 学习 GCDAsyncSoocket 客户端的连接与数据发送
- 学习 GCDAsyncSoocket 服务端的监听客户端的连接与数据响应

去除回车和换行符

```
readStr = [readStr stringByReplacingOccurrencesOfString:@"\r" withString:@""];
readStr = [readStr stringByReplacingOccurrencesOfString:@"\n" withString:@""];
```