**02计算机网络面试核心**

**网络知识的考察点相对比较固定，更偏向理论**

**网络模型：当前主要存在4层、5层、7层协议**

**OSI开放式互联参考模型（7层协议模型）：**

1. 物理层

解决机器A与机器B之间发送接收比特流。定义物理设备标准（网线、光纤、接口类型）。主要任务传输010101...比特流，进行比特流与电流强弱之间的数模转化。本层数据：比特 本层工具：网卡

1. 数据链路层

定义如何格式化数据。通过建立数据链路连接，采用差错控制与流量控制的方法，以提供错误检测及纠正。本层数据：比特组成的帧 本层工具：交换机

1. 网络层

选择找到目标节点、经过多个节点完成点对点通信的最佳路径。如何将数据从发送方路由到接收方。本层数据：数据包或分组 本层工具：路由器 **本层关注IP协议**

1. 传输层（OSI模型最重要的一层）

传输大量文件需要切分为segment进行发送，解决segment的丢失重传，到达顺序等。进行流量控制，规定发送速率。本层数据：报文 **本层关注TCP、UDP协议**

1. 会话层

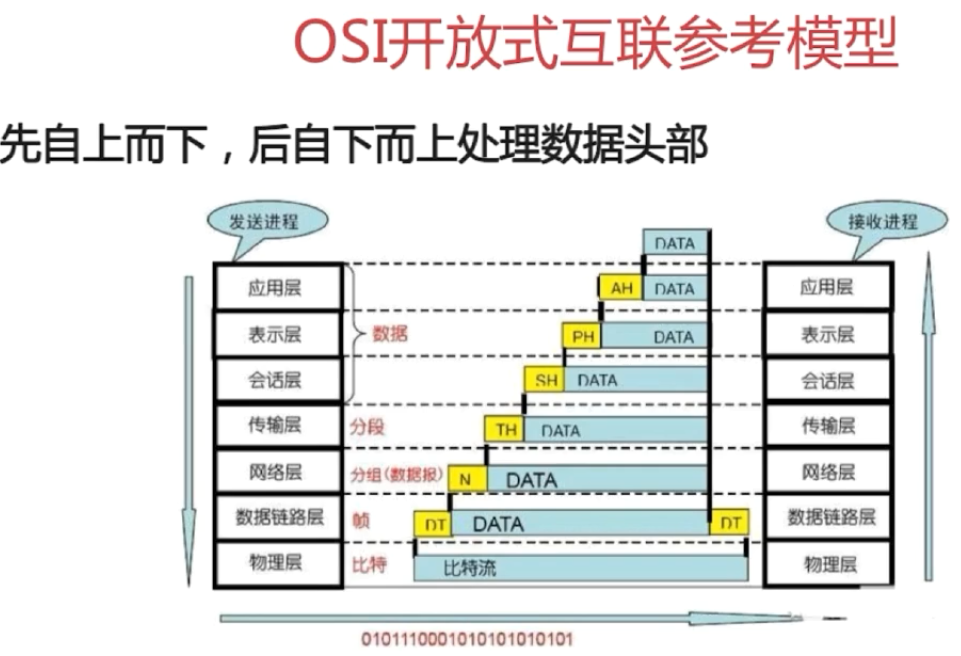
自动收发包，自动寻址。负责维护两个会话层主机之间连接的建立、管理和终止，以及数据的交换。

1. 表示层

数据加密解密、格式转换翻译、压缩解压缩等

1. 应用层

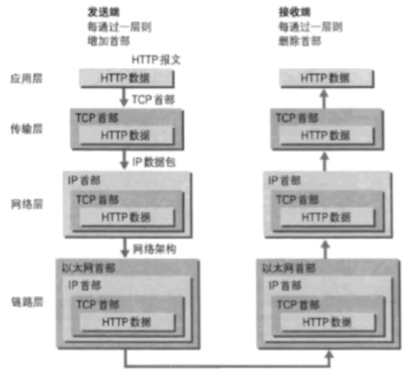
**本层关注HTTP协议**



**OSI只是一个概念性框架，没有提供实现方法，不是一个可用的标准——TCP/IP4层架构参考模型是事实上的标准**

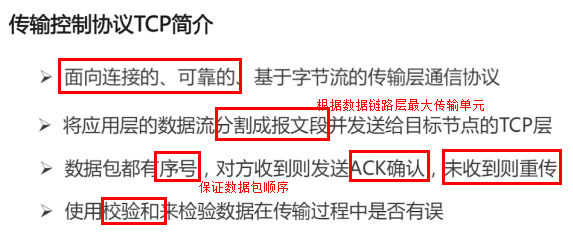


TCP/IP模型与OSI模型的层次对应关系



TCP/IP先自上而下，后自下而上处理数据头部

**TCP(传输控制协议)：**



IP地址唯一标识主机，TCP协议+端口号唯一标识进程

**Socket套接字：IP地址+协议+端口号——唯一标识网络中一个进程**

**TCP报文格式：《计算机网络第3版》p293 图7-14**

**TCP协议报文段结构：**

Seq\_num和ack\_num:

TCP连接中传送的字节流中的每个字节都被编号。若有一个报文段seq\_num=107,共携带100个字段，则下一个报文段seq\_num=207。Ack\_num为确认号，主机B收到主机A发送的seq\_num=107,长度为100的报文，则B发送给A的确认报文的ack\_num=207。

常见的TCP Flags(报文中的控制位)：

URG:紧急指针有效标志 ACK:确认序列号 PSH:push标志 RST:重置连接标志

SYN:同步序号，用于建立连接过程 FIN:finish标志，用于释放连接

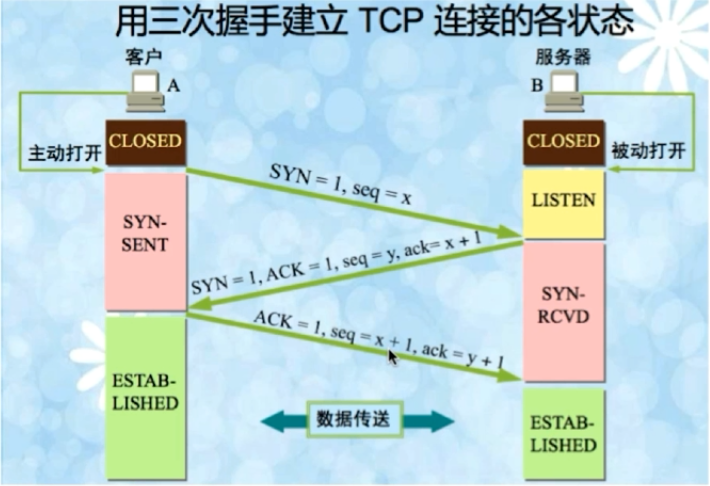
Window窗口大小：向发送端告知接收端的缓存大小，控制发送端发送数据的速率，从而达到流量控制。

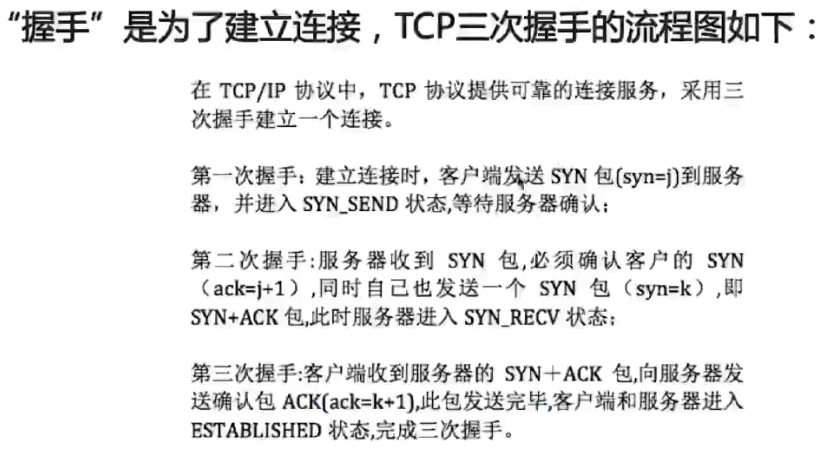
校验和：发送端对整个报文段进行16位奇偶校验和计算并存储，由接收端进行验证

紧急指针：指出本报文段中紧急数据的字节数

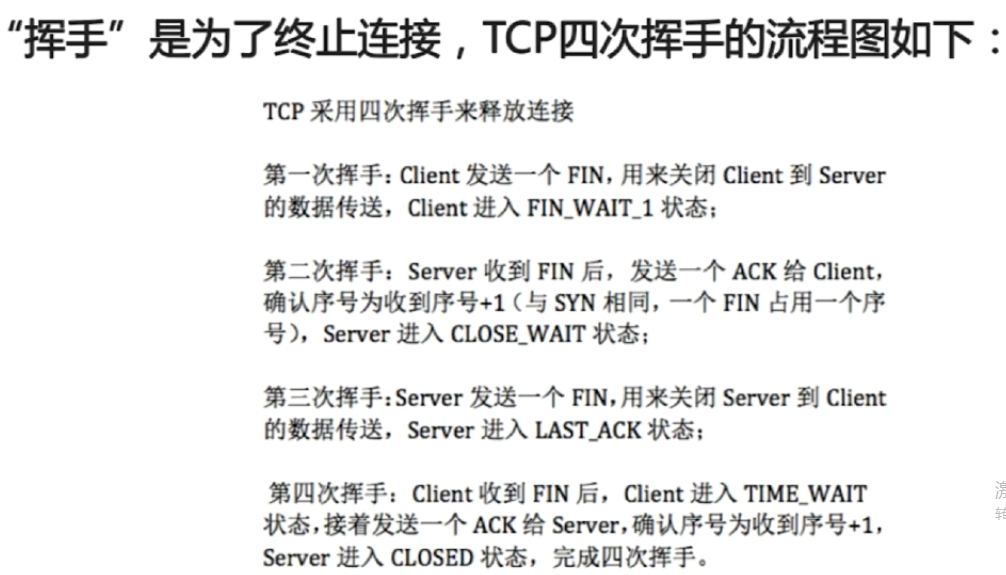
**全双工通信**：通信双方的应用程序均可以给对方发送数据——主机A可以给主机B发送信息，主机B也可以给主机A发送信息。双放都需要设置发送和接收缓冲区，接收端将接收到的数据放入接收缓冲区，高层应用程序从中读取数据。

**TCP工作原理示意图：《计算机网络第3版》p297 图7-15**









问题1：为什么会有2MSL的TIME\_WAIT状态？

答：（1）保证被动关闭方收到了最后一个ACK包，如果被动关闭放未收到最后一个ACK，就会触发FIN包重发，一来一去正好需要2MSL。（2）留出足够时间使本连接不会和后面的连接混在一起。如果连接被重用，延迟收到的包会和新连接混在一起。

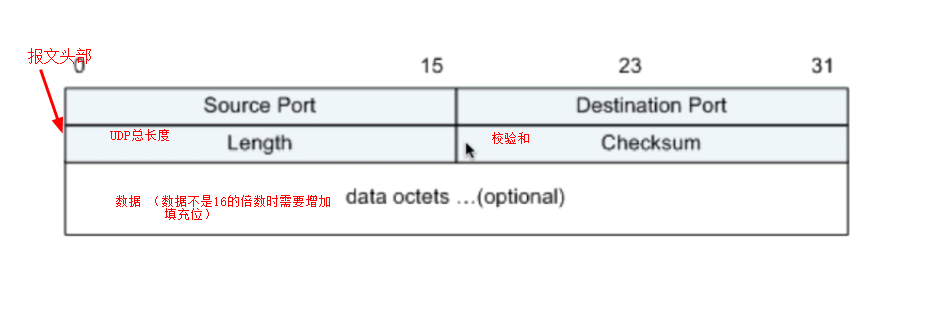
问题2：为什么需要四次握手才能断开连接？

答：（1）因为TCP协议是全双工的，发送方和接收方都需要FIN报文和ACK报文。

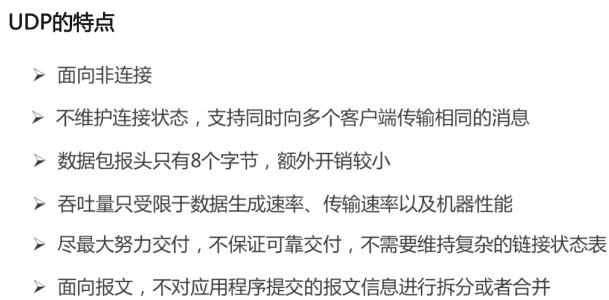
问题3:服务器出现大量CLOSEE\_WAIT状态的原因？

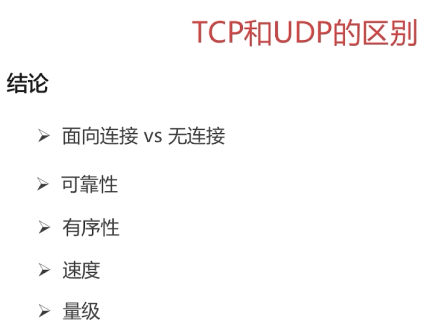
待解答

**UDP（用户数据报协议）：**



UDP报文格式



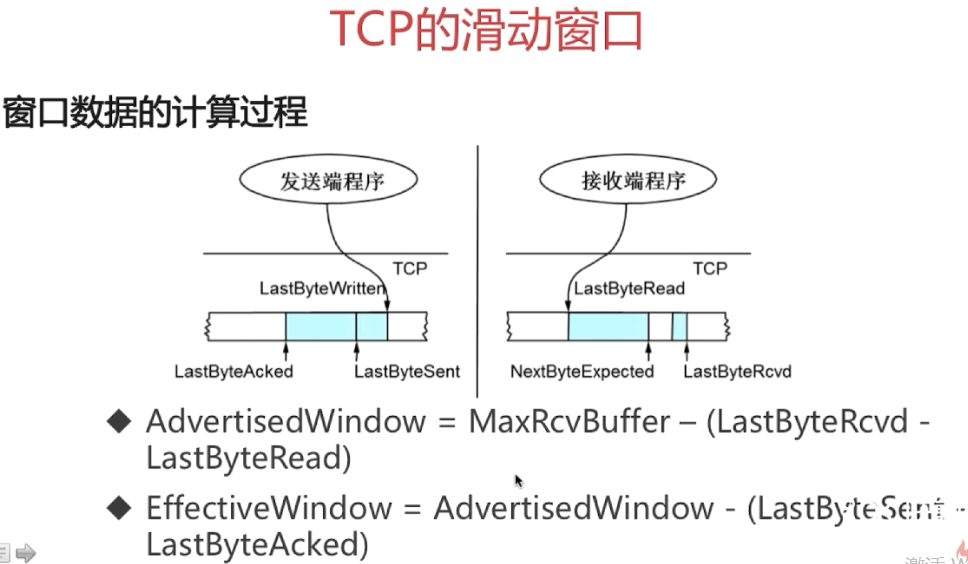


**RTT（Round-Trip Time）**：往返时延，发送一个数据包到收到对应的ACK，所花费的时间

**RTO（Retransmission Time-Out）**:超时重传时间，经过RTT计算。TCP每次发送后维护一个计时器，计时器超过RTO值则重传

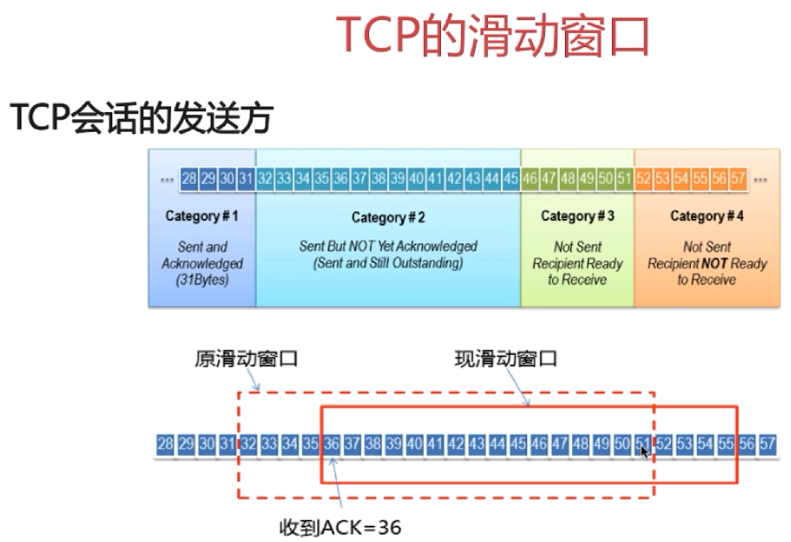
**TCP滑动窗口策略（待完善）：**

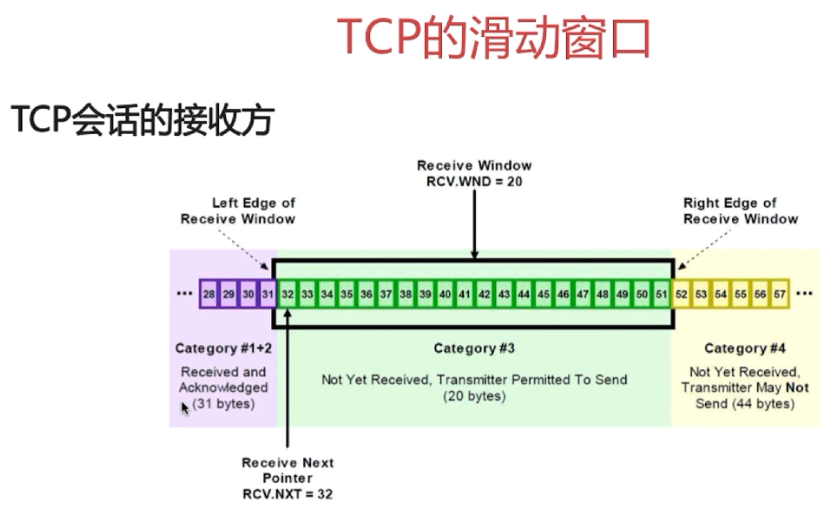
TCP使用**滑动窗口**做流量控制与乱序重排。用以保证TCP的**可靠性**和**流量控制**。



由接收方数据报window字段计算

接收方当前可接收的最大数据量和发送方可发送的最大数据量



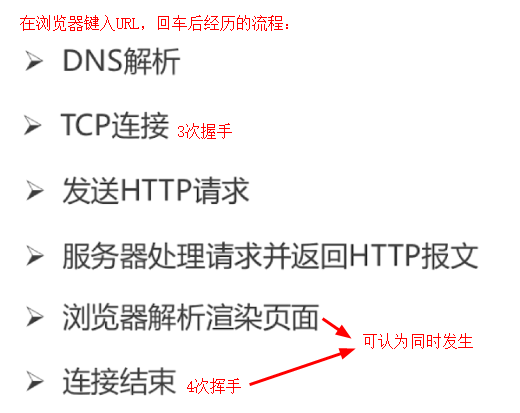


TCP最基本的传输可靠性来自于确认重传机制，TCP滑动窗口的可靠性建立在确认重传基础上。

**HTTP协议——工作在应用层，基于请求与相应的的无状态协议**

特点：

1. 支持客户/服务器模式——client请求，server相应
2. 简单快速，程序规模小，通信速度快——client向server请求服务只需传送请求方法和路径，常用请求方法get、post
3. 灵活——允许传播任意类型的的数据对象
4. 无连接——限制每次连接只处理一个请求（HTTP1.1采用keep-alive长连接）
5. 无状态



HTTP状态码5种可能的取值：

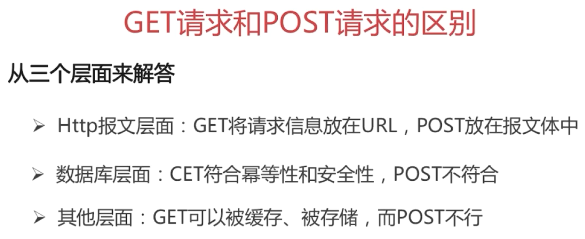
1xx:指示信息——表示请求已接收，继续处理

2xx:成功——表示请求已被成功接收、理解、接受

3xx:重定向——要完成请求必须进行更进一步操作（一般与跳转有关）

4xx:客户端错误——请求有语法错误或无法实现

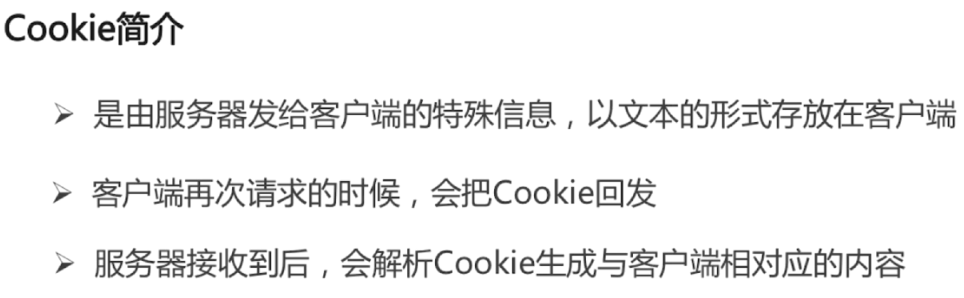
5xx:服务器端错误——服务器未能实现合法的请求



Cookie和Session的区别？

因为HTTP是无状态的，因此我们每次访问有登录需求的的页面时，为了避免多次输入登录的账号密码，引入Cookie和Session机制。

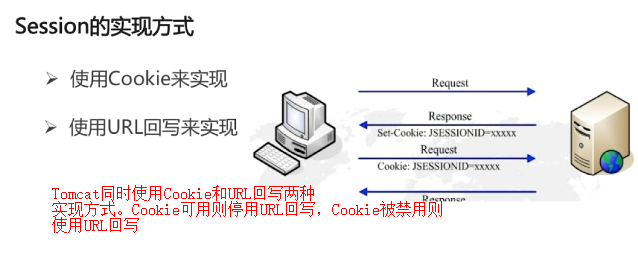
Cookie——**客户端**的解决方案



Session——服务器端的解决方案

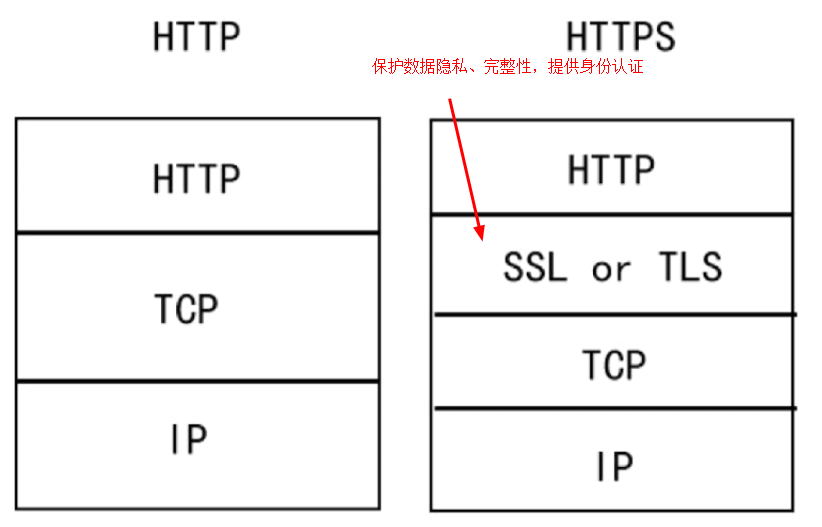
Session机制中，服务器使用了一个类似散列表的结构保存信息。当程序需要为某个client的请求创建Session时 ，首先检查请求内是否包含Session id，如果包含Session id，则说明服务器之前为client创建过Session,根据Session id检索出Session使用，检索不出则新建一个；如果请求不包含Session id，则为此client创建一个Session并生成Session id。

Session id——既不会重复又不容易找到规律以防捏造。Session id回发给Client保存。





HTTPS（超文本传输安全协议）——以计算机网络**安全**通信为目的的传输协议，安全版的HTTP。

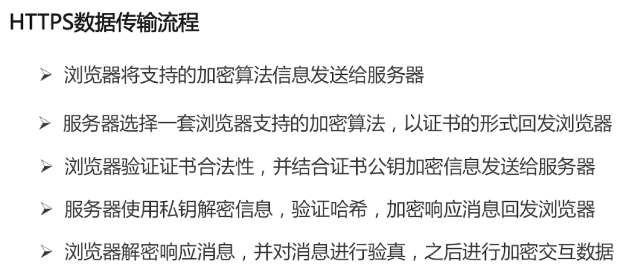


SSL（Security Sockets Layer，安全套接层）——为网络通信提供安全及数据完整性的协议。SSL3.0更名为TLS。采用（1）身份认证和（2）数据加密提供安全和数据完整性。

常见加密方式：

1. 对称加密：加密解密都使用同一种密钥——效率较高，安全性较差
2. 非对称加密：加密用密钥（公钥）和解密用密钥（私钥）不相同——效率较低，安全性强，能加密的数据长度有限
3. 哈希算法：将任意长度的信息转换为固定长度值，算法不可逆
4. 数字签名：在内容后面加上一段哈希过后的值，值可证明信息没有被修改过。哈希值和内容信息一起发送。

HTTPS使用证书+加密手段



**HTTP和HTTPS的区别？**

答：

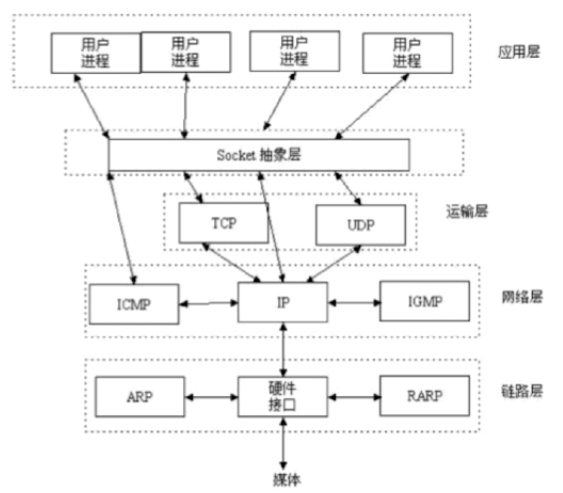
1. HTTPS需要到CA获取证书，免费证书较少，需要一定费用。HTTP不需要
2. HTTPS是SSL加密传输，HTTP明文传输
3. 连接方式不同，HTTPS默认使用443端口，HTTP使用80端口
4. HTTPS=HTTP+加密+认证+完整性保护，较HTTP安全。HTTP无状态，HTTPS中的SSL有状态



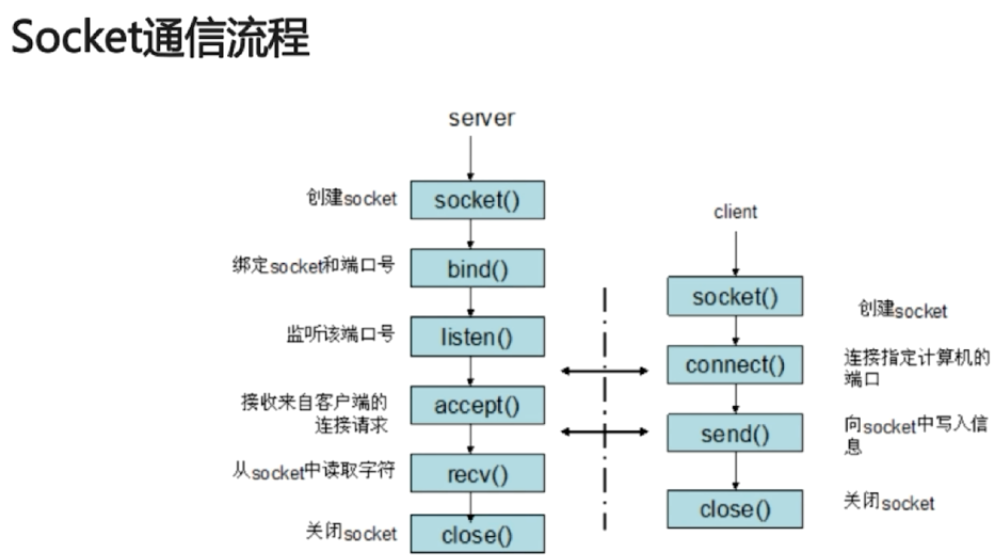
**增加问题：IP协议、Ipv4和Ipv6的区别以及它们对地址的定义**

IP地址唯一标识主机，TCP协议+端口号唯一标识进程

**Socket套接字：IP地址+协议+端口号——唯一标识网络中一个进程，使得网络间的进程能够通信**



Socket是对TCP/IP协议的抽象，是操作系统对外开放的接口



Socket面试题（考察较少）：见视频