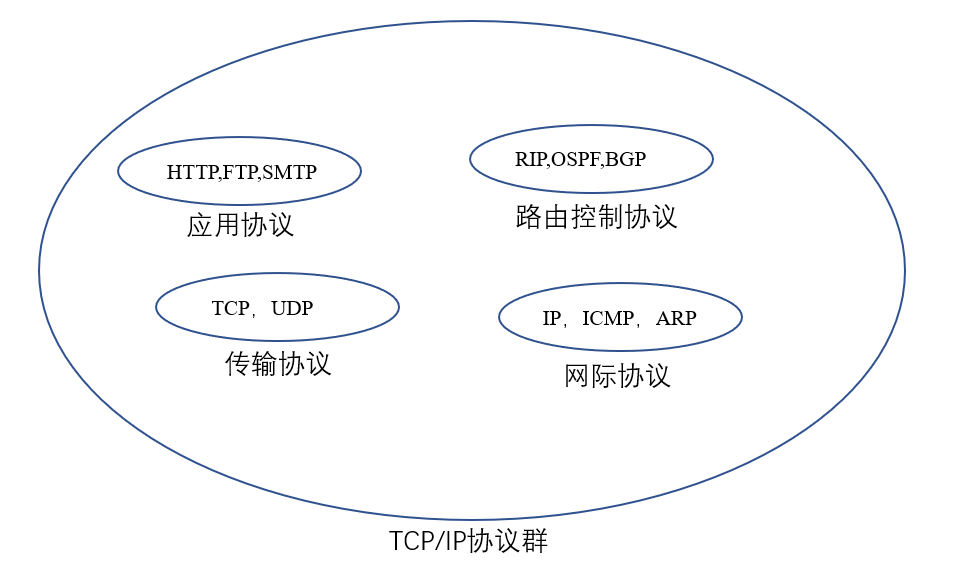
1.TCP/IP协议

协议：计算机与计算机之间通过网络实现通信达成的 “约定”。

两台计算机之间必须支持相同的协议，并遵循相同协议进行处理，这样才能进行通信。

TCP/IP是一个协议群，如下图。



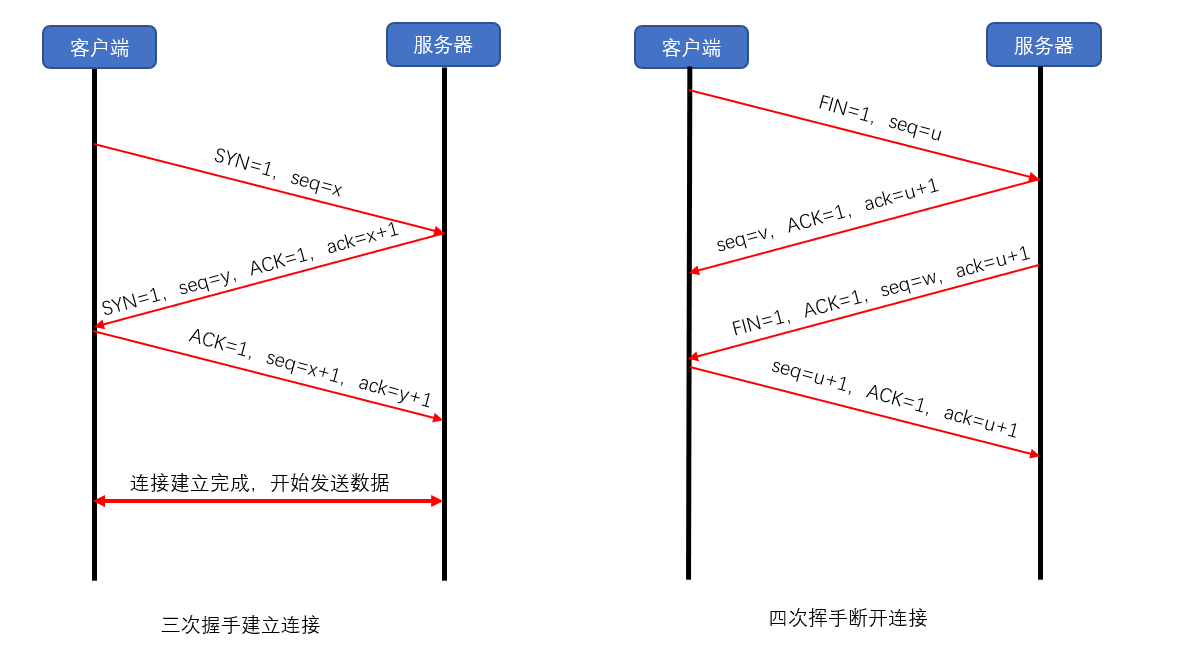
分层：各层之间相互独立，灵活性好，易于实现和维护，促进标准化工作。



TCP三次握手和四次挥手的理解

三次握手指的是建立一个TCP连接时，需要客户端和服务器总共发送三个包，进行三次握手的目的就是确定双方的接收与发送都是正常的，并且同步连接双方的序列号和确认号，交换TCP窗口大小信息。

四次挥手指的是TCP终止一个连接需要发送四个包，收到FIN后不会立即关闭，需要先回复ACK，然后在发送FIN请求关闭；这是由于TCP的半关闭造成的，半关闭是TCP提供了结束了发送还具有接收能力

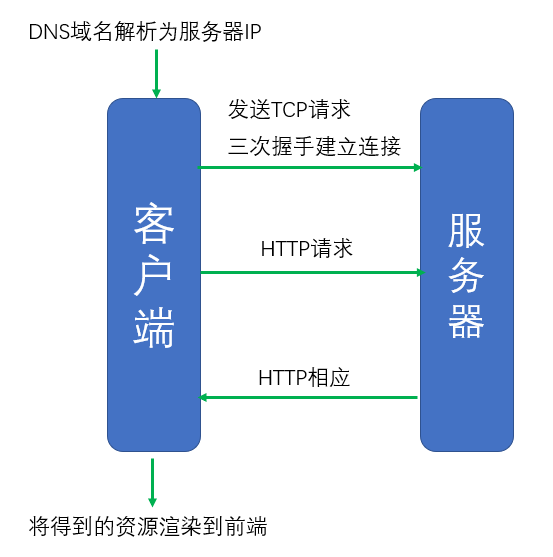
ack确认号，seq序列号，SYN请求连接报文，FIN请求断开连接报文，ACK确认报文。

**如果已经建立了连接，但是客户端突然出现故障了怎么办？**

TCP设有一个保活计时器，客户端如果出现故障，服务器不能一直等下去，白白浪费资源。服务器每收到一次客户端的请求后都会重新复位这个计时器，时间通常是设置为2小时，若两小时还没有收到客户端的任何数据，服务器就会发送一个探测报文段，以后每隔75秒钟发送一次。若一连发送10个探测报文仍然没反应，服务器就认为客户端出了故障，接着就关闭连接。

2.HTTP

HTTP（Hyper TextTransfer Protocol）超文本传输协议，用于客户端和服务器端之间的通信。现在常用的是HTTP1。



特点：无连接，每次连接只处理一个请求；简单快速，客户端向服务端请求服务的时候，只需要传送请求方法和路径；无状态，对于事物的处理没有记忆，后续需要前面的信息只能重传。

HTTP2的多路复用就解决了上面的性能问题，采用单一长连接，只会创建一条TCP连接，能减少TCP握手带来的延时，提高吞吐量；多路是因为一个TCP连接中可以存在多个流，能发送多个请求，对端可以通过帧中的标识知道属于哪个请求，并且HTTP2还能对这些流指定优先级，极大提高了传输性能。

2.1常见的请求方法

GET：请求访问已经被URL识别的资源；

POST：向指定资源提交数据进行处理请求（提交表单或者上传文件）；

PUT：与POST基本相同，PUT是幂等的，POST不是幂等的；

HEAD：请求获得报文首部；

DELETE：用来删除指定文件；

2.2服务端返回的常见状态码

2××（请求被正常处理）

200 OK：请求成功，资源成功返回。

202 Accepted：接收请求，但处理未完成。

204 No content：请求成功，但没有资源返回。

206 Partial Content：请求成功，返回指定内容。

3××（重定向）

301 Moved Permanently：永久重定向

302 Found：临时重定向

304 Not Modified：服务器资源未修改，可直接使用客户端未过期缓存

4××（客户端错误）

400 Bad Request：请求报文中存在错误语法；

401 Unauthorized：发送的请求需要通过HTTP认证；

403 Forbidden：服务器拒绝执行

404 Not Found：服务器上没有请求的资源

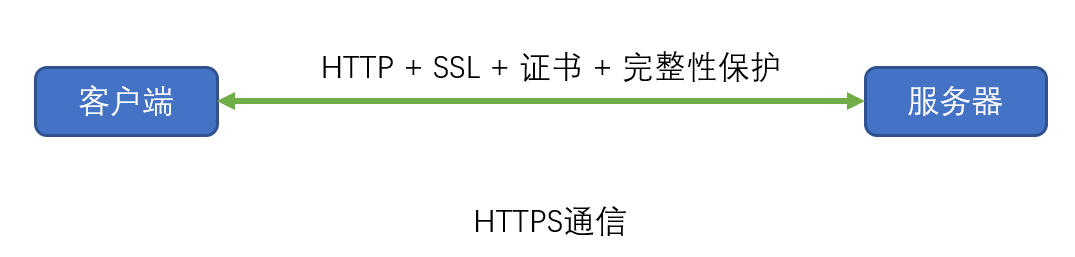
5××（服务器错误）

500 Internal Server Error：服务器在执行请求时发生了错误。

503 Server Unavailable:服务器当前不能处理客户端的请求，一段时间后可能恢复。

3.HTTPS

我们把添加了加密认证机制的HTTP称为HTTPS



HTTP的不足：通信使用明文，内容可能会被窃听；

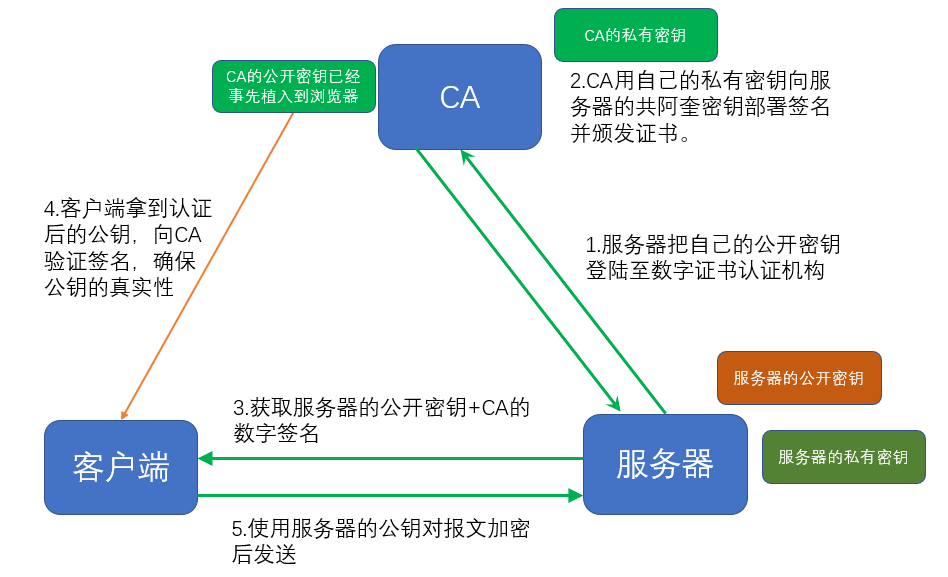
不验证通信方的身份，因此可能会遭遇伪装；

无法证明报文的完整性，所以有可能已经遭到篡改；

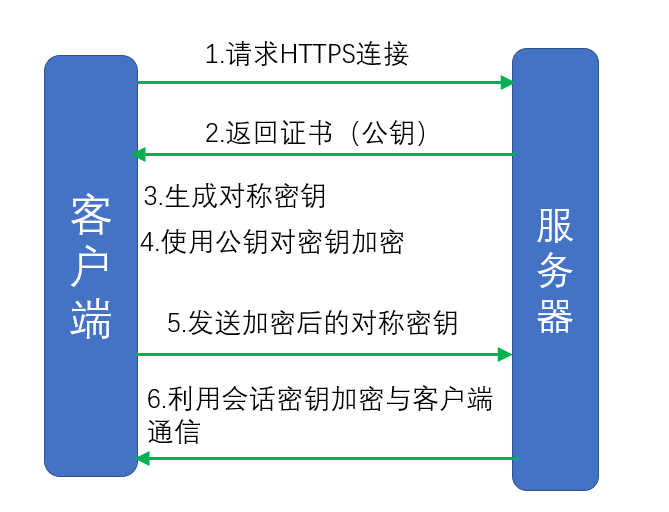
这时候HTTPS的出现就是为了弥补不足，HTTPS并非是一种新的协议，只是HTTP通信接口部分用SSL和TLS协议代替。

SSL（Transport Socket Layer）安全套接字层，它采用公钥加密技术：发送密文的一方使用对方的公开密钥进行加密处理，对方收到加密信息后，再使用自己的私有密钥进行解密。

但是公开密钥加密方式还是有一些问题，无法证明公开密钥的货真价实，这时候就需要数字证书认证机构（CA，Certificate Authority）和其相关机构颁发的公开密钥证书。



这时候HTTPS安全通信机制：



不能一直使用HTTPS，非敏感数据使用HTTP：

HTTPS通信会消耗更多的CPU以及内存资源；

连接缓存不如HTTP高效，会增加数据开销和功耗；

申请SSL证书需要钱；

SSL安全算法会消耗CPU资源。