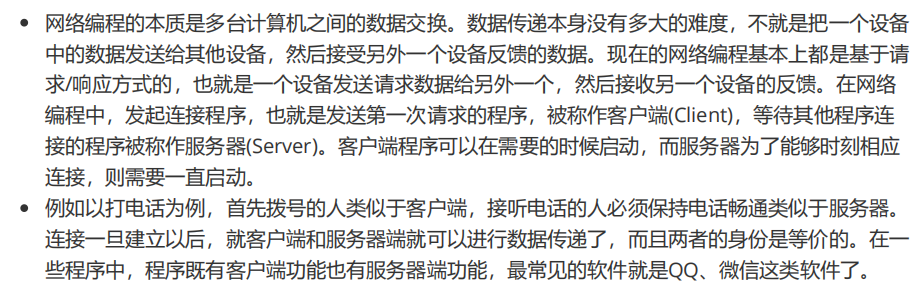
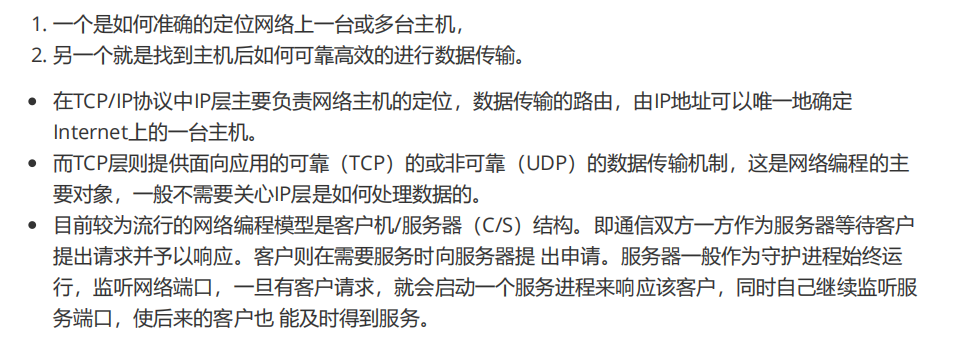
# 计网

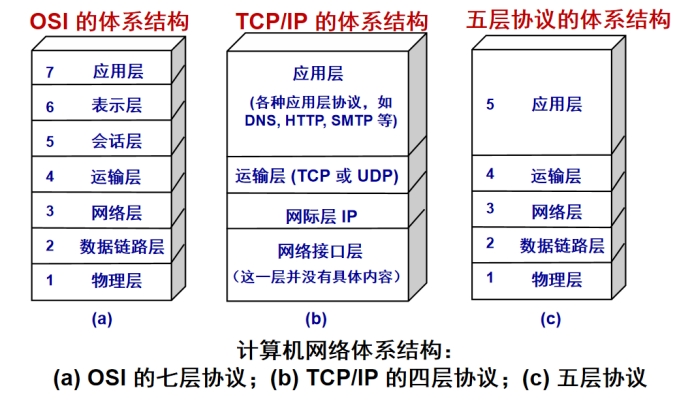
## 什么是网络编程



## 网络编程中两个主要问题



## 网络分层



应用层 应用层最靠近用户的一层，是为计算机用户提供应用接口，也为用户直接提供各种网络服务。我们常见应用层的网络服务协议有：HTTP，HTTPS，FTP，TELNET等。

传输层 建立了主机端到端的链接，传输层的作用是为上层协议提供端到端的可靠和透明的数据传输服务，包括处理差错控制和流量控制等问题。该层向高层屏蔽了下层数据通信的细

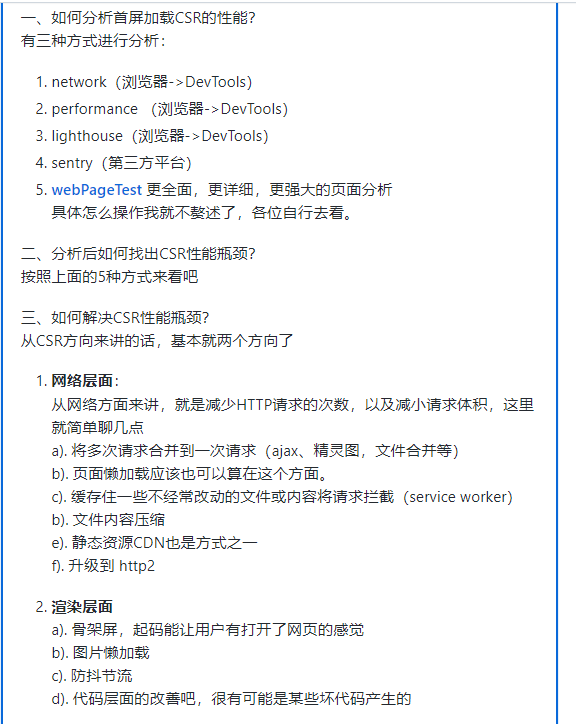
节，使高层用户看到的只是在两个传输实体间的一条主机到主机的、可由用户控制和设定的、可靠的数据通路。我们通常说的，TCP UDP就是在这一层。端口号既是这里的“端”。

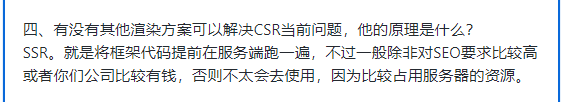
网络层 本层通过IP寻址来建立两个节点之间的连接，为源端的运输层送来的分组，选择合适的路由和交换节点，正确无误地按照地址传送给目的端的运输层。就是通常说的IP层。这一 层就是我们经常说的IP协议层。IP协议是Internet的基础。

数据链路层 通过一些规程或协议来控制这些数据的传输，以保证被传输数据的正确性。实现这些规程或协议的 硬件 和软件加到物理线路，这样就构成了数据链路，

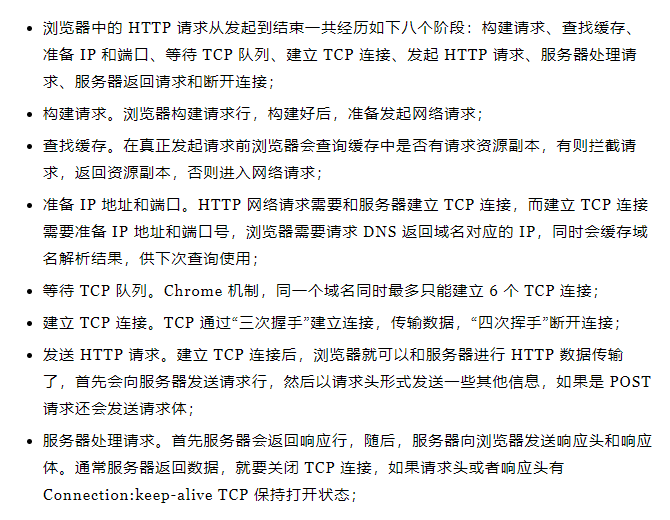
物理层 定义物理设备的标准，主要对物理连接方式，电气特性，机械特性制定统一标准，传输比特流。

## 首屏渲染白屏问题

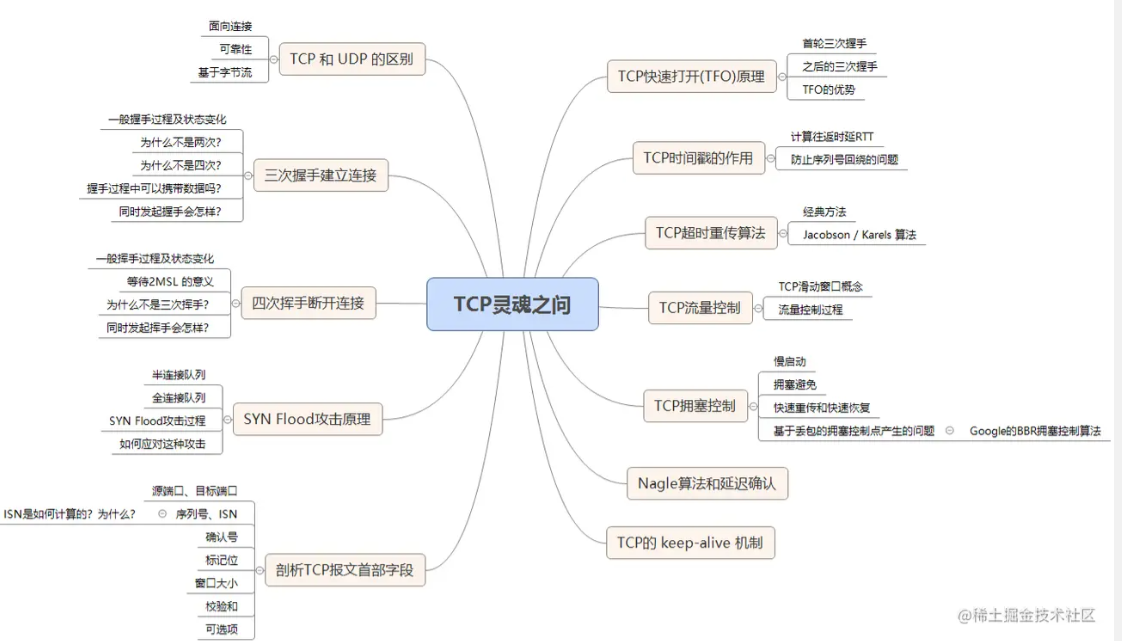




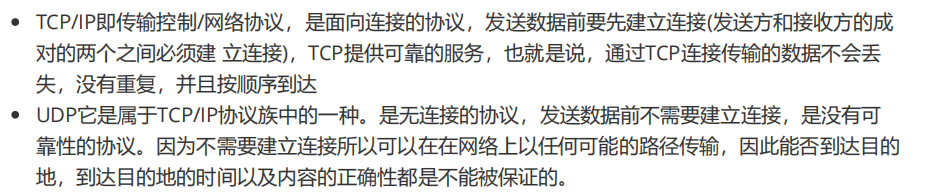
## http第二次打开速度很快



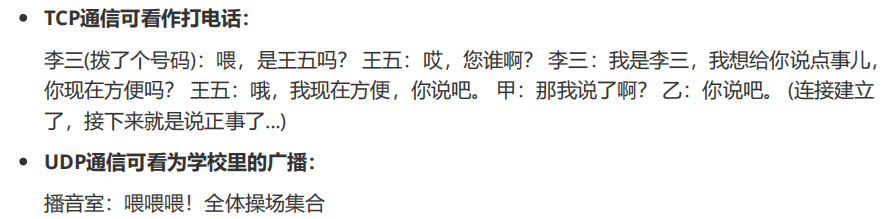
## TCP/IP和UDP

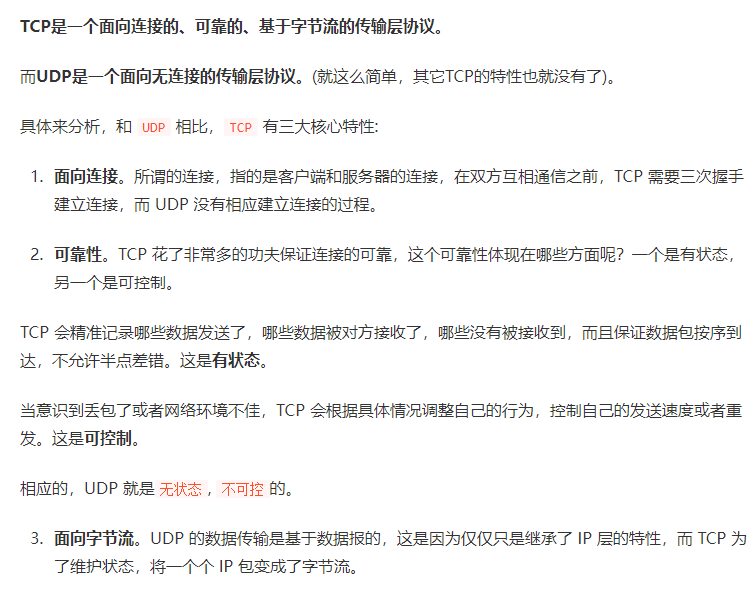


### 什么是TCP/IP和UDP

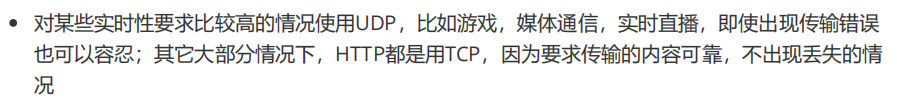


#### TCP和UDP区别





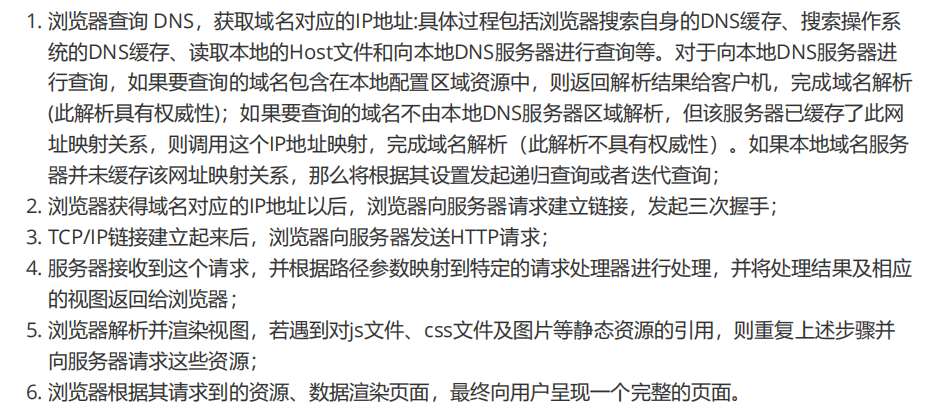
#### TCP和UDP应用场景



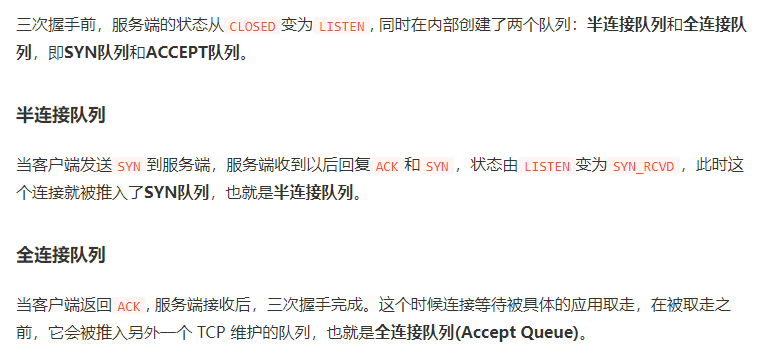
#### 运行在TCP和UDP应用层协议分析

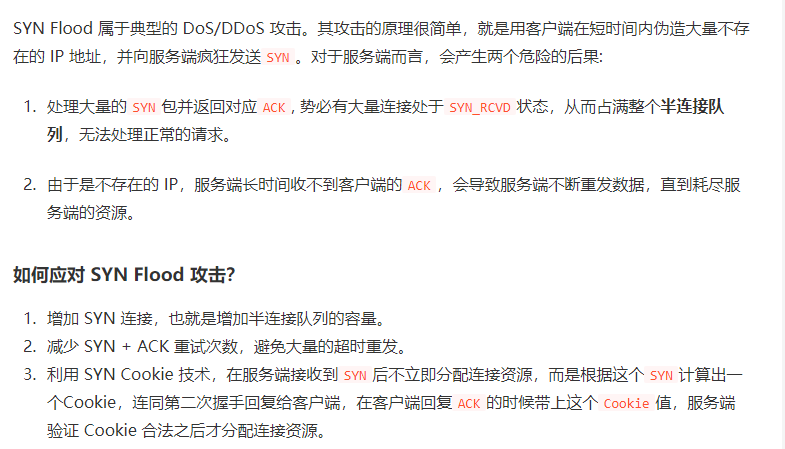


#### 输入地址到获得页面

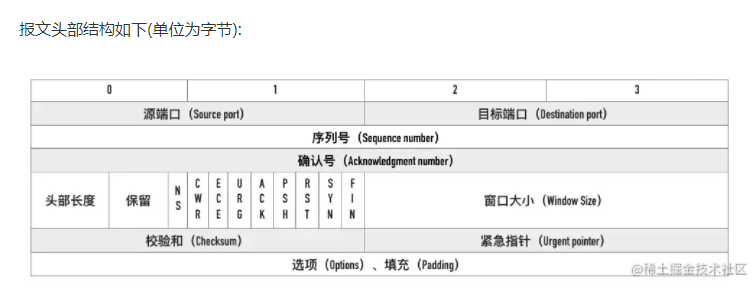


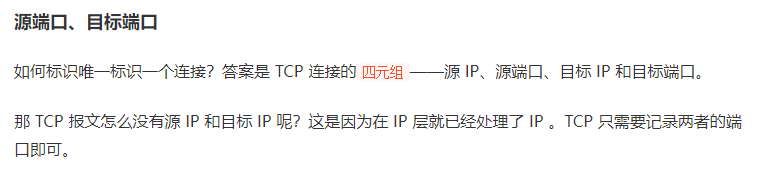
### 半连接队列和SYN Flood攻击原理

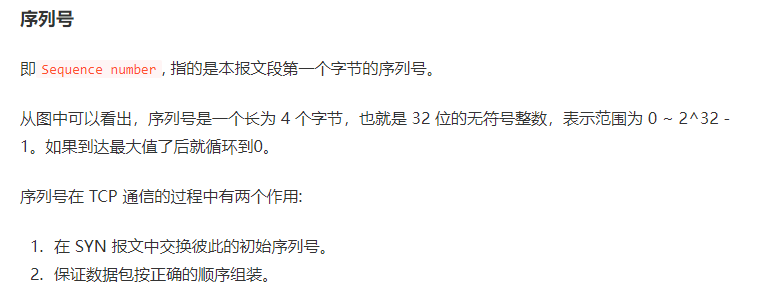


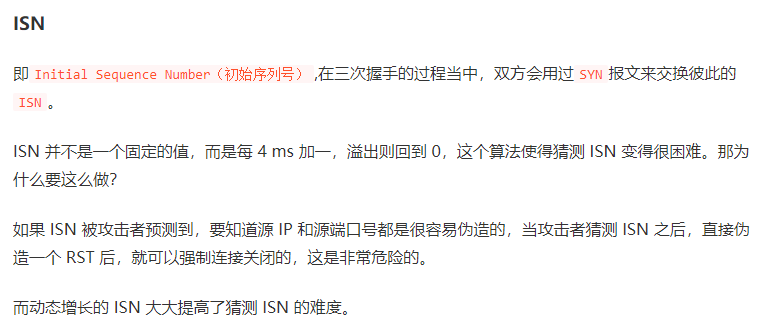


### TCP 报文头部的字段







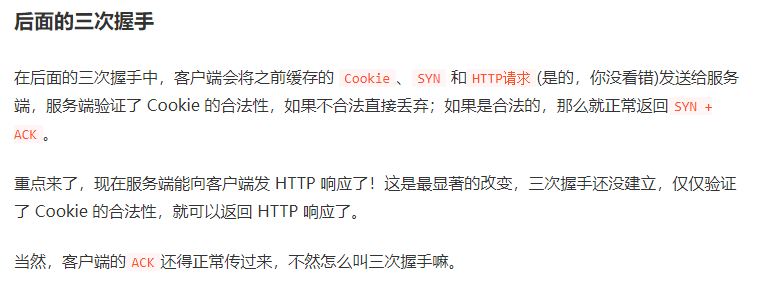


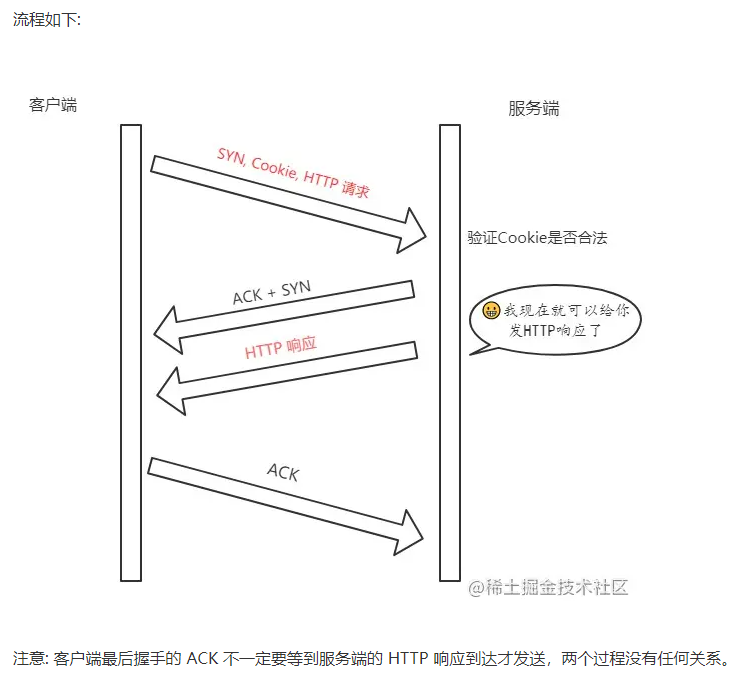


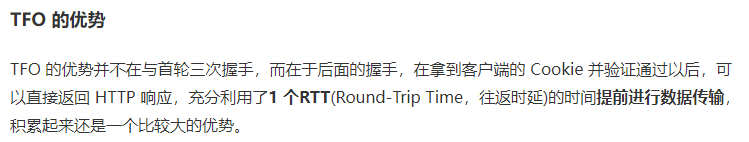


### TCP快速打开原理

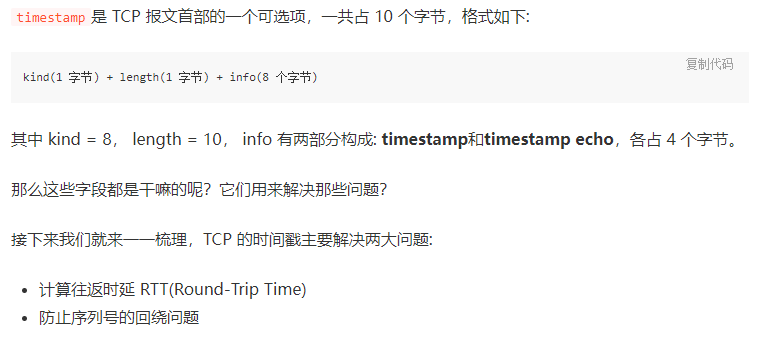








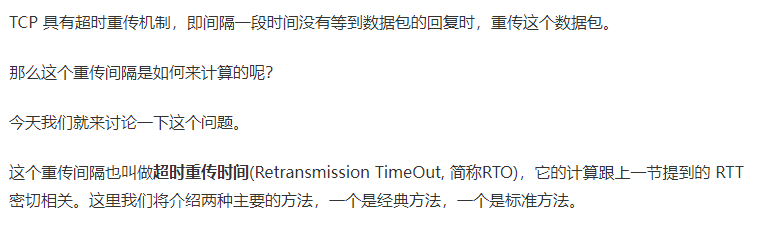
### 能不能说说TCP报文中时间戳的作用







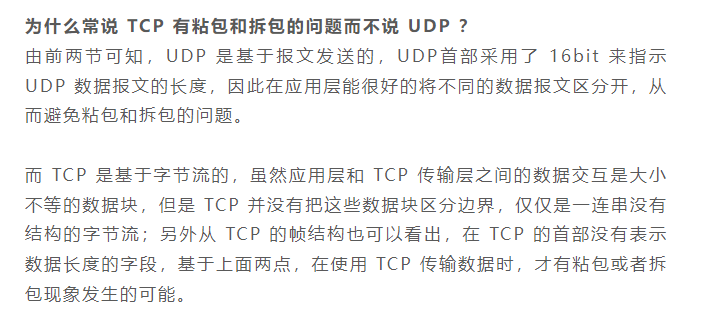
### TCP 的超时重传时间是如何计算的？



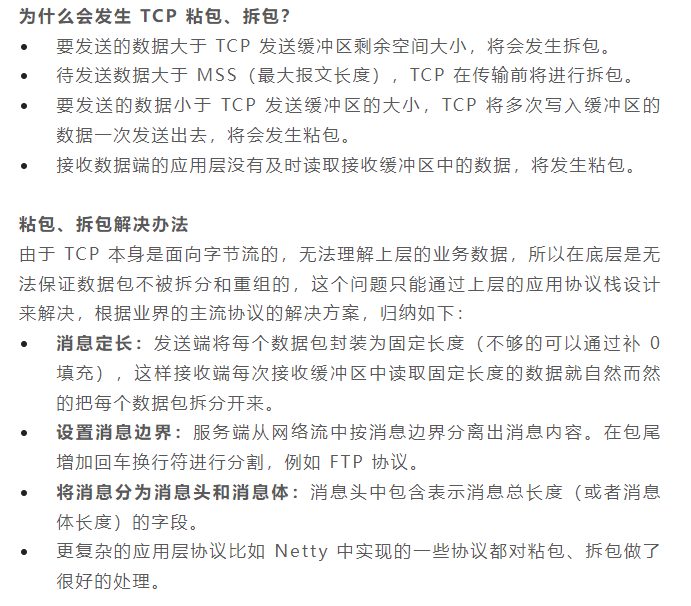




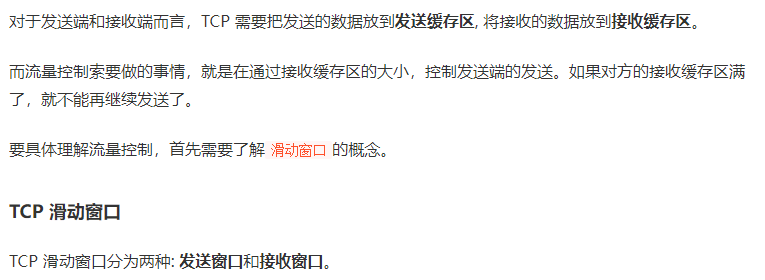
### 粘包、拆包

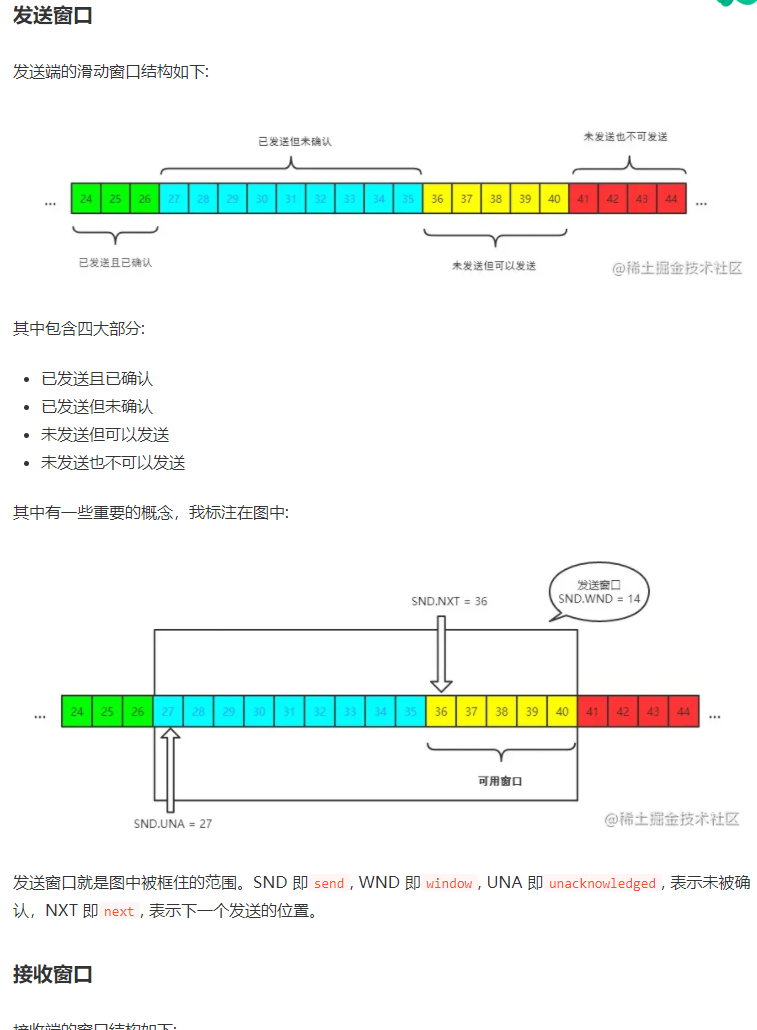


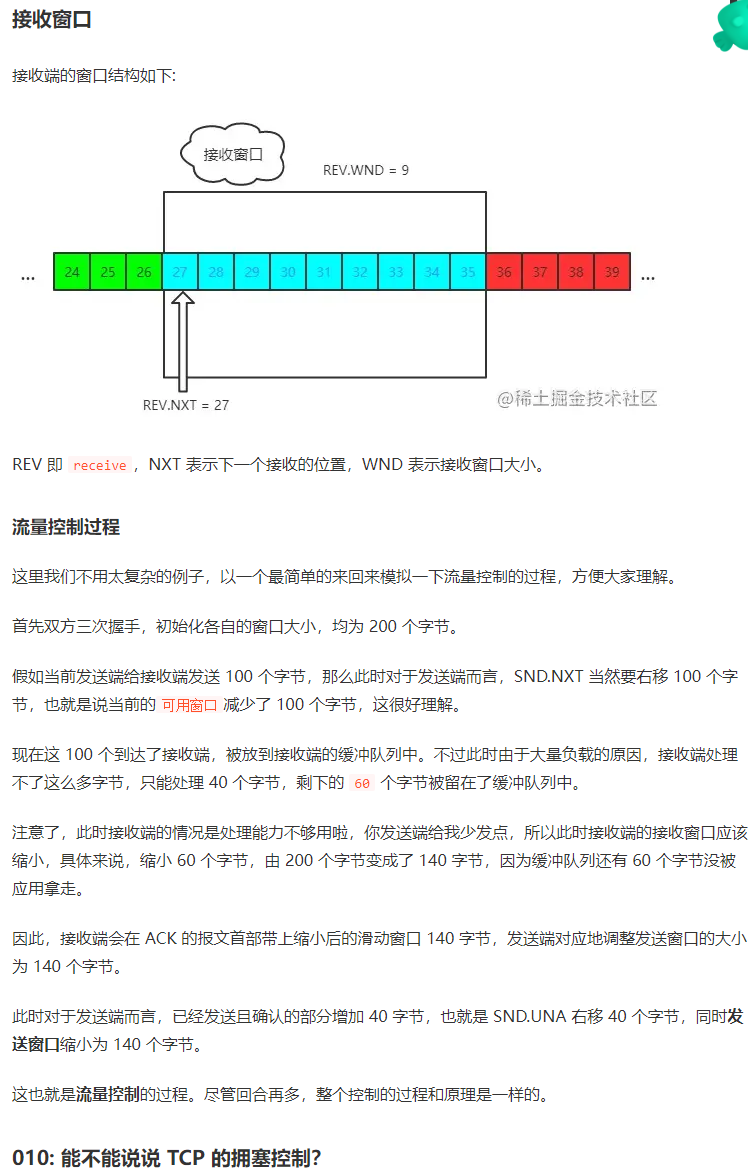




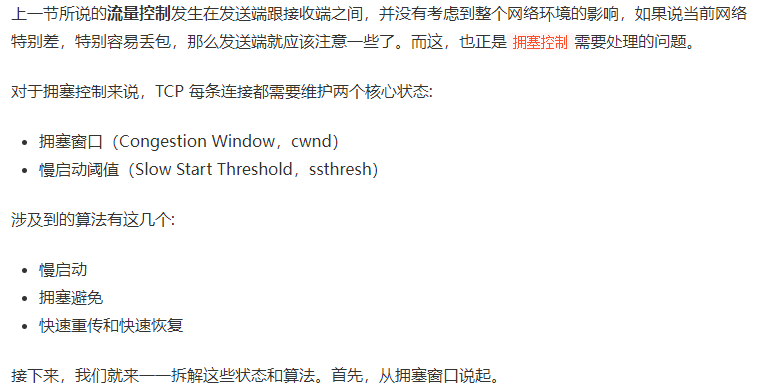
### 流量控制

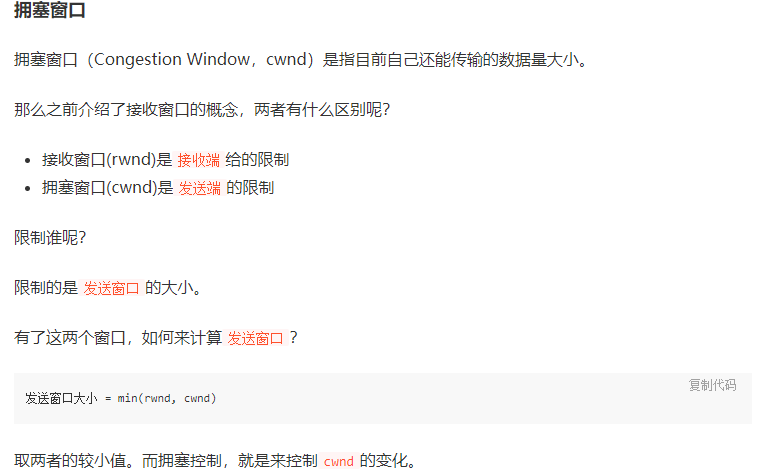


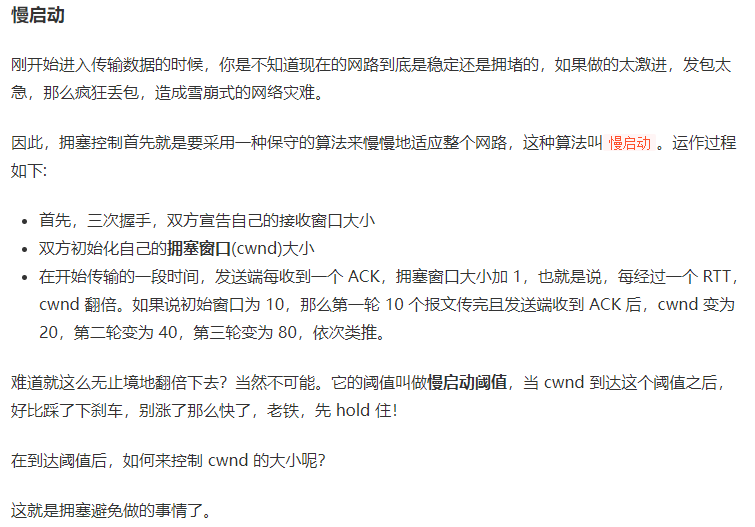


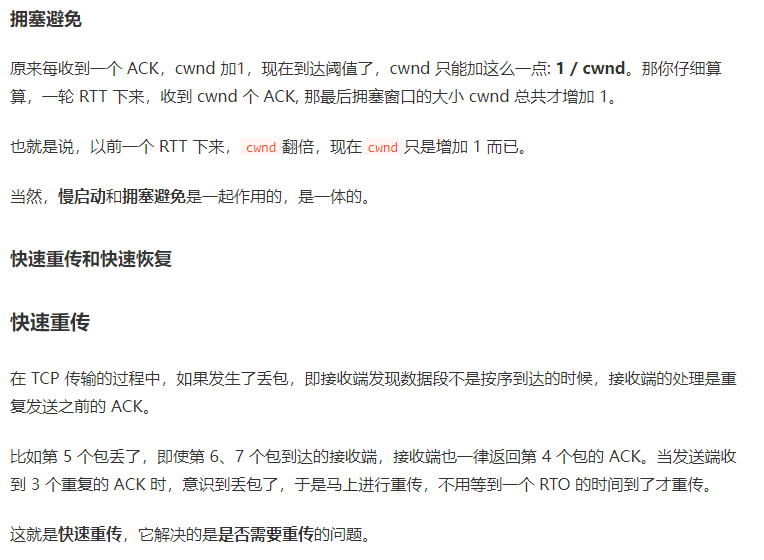


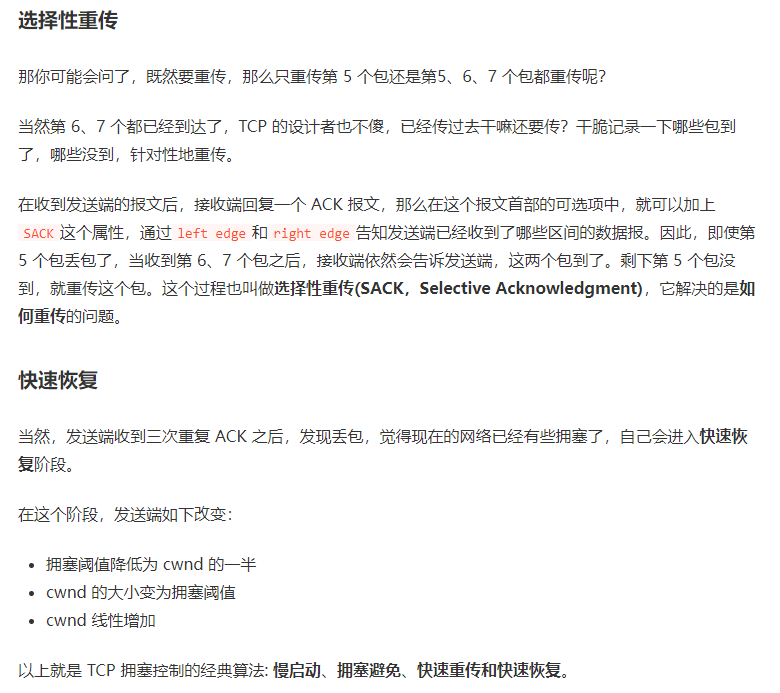
### 拥塞控制







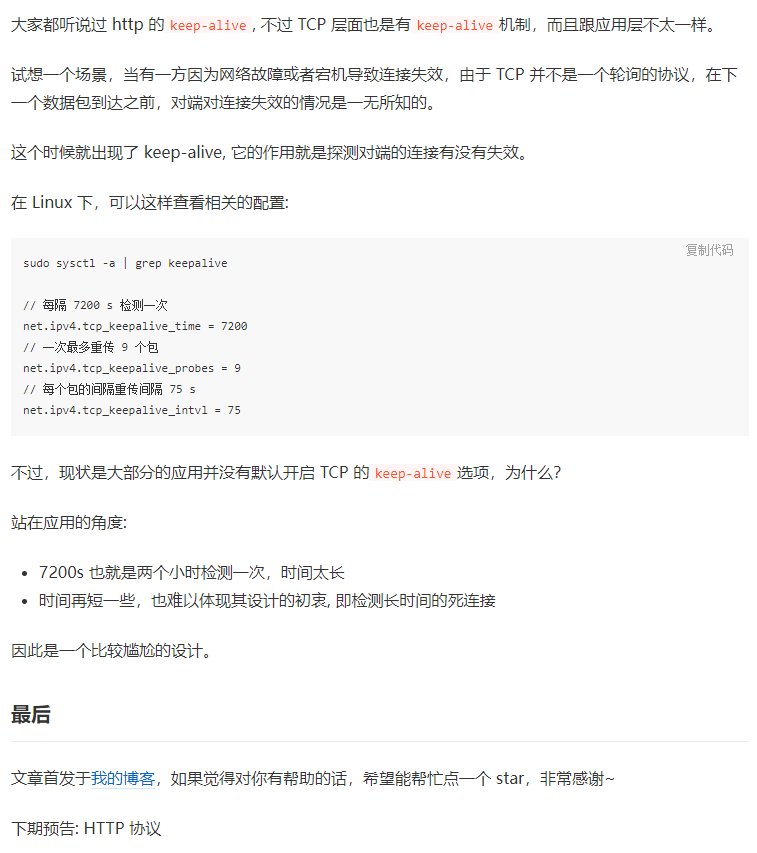




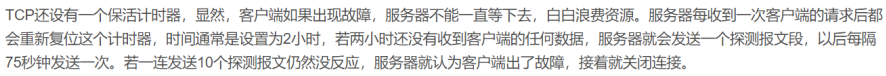
### Nagle 算法和延迟确认？



### TCP 的 keep-alive？



### 如果已经建立连接但客户端突然出现故障怎么办？



### TCP怎么保证可靠性

TCP协议保证数据传输可靠性的方式主要有：

校验和 序列号 确认应答 超时重传 连接管理 流量控制 拥塞控制 （1）校验和

发送方：在发送数据之前计算检验和，并进行校验和的填充。 接收方：收到数据后，对数据以同样的方式进行计算，求出校验和，与发送方的进行比对。

注意：如果接收方比对校验和与发送方不一致，那么数据一定传输有误。但是如果接收方比对校验和与发送方一致，数据不一定传输成功。

（2）确认应答、序列号、超时重传

TCP传输的过程中，每次接收方收到数据后，都会对传输方进行确认应答。也就是发送ACK报文，表示已经收到该数据段，这个ACK报文当中带有对应的确认序列号，告诉发送方，接收到了哪些数据，下一次的数据从哪里发。

序列号的作用不仅仅是应答的作用，有了序列号能够将接收到的数据根据序列号排序，并且去掉重复序列号的数据。这也是TCP传输可靠性的保证之一。

如果发送方迟迟未收到确认应答，那么可能是发送的数据丢失，也可能是确认应答丢失，这时发送方在等待一定时间后会进行重传。

（3）通过三次握手与四次挥手的过程保证可靠的连接。

（4）流量控制

接收端在接收到数据后，对其进行处理。如果发送端的发送速度太快，导致接收端的接收缓冲区很快的填充满了。

此时如果发送端仍旧发送数据，那么接下来发送的数据都会丢包，继而导致丢包的一系列连锁反应，超时重传呀什么的。

而TCP会根据接收端对数据的处理能力，决定发送端的发送速度，这个机制就是流量控制。

如何控制：在TCP协议的报头信息当中，有一个16位字段的窗口大小。在介绍这个窗口大小时我们知道，窗口大小的内容实际上是接收端接收数据缓冲区的剩余大小。这个数字越大，证明接收端接收缓冲区的剩余空间越大，网络的吞吐量越大。接收端会在确认应答发送ACK报文时，将自己的即时窗口大小填入，并跟随ACK报文一起发送过去。而发送方根据ACK报文里的窗口大小的值的改变进而改变自己的发送速度。如果接收到窗口大小的值为0，那么发送方将停止发送数据。并定期的向接收端发送窗口探测数据段，让接收端把窗口大小告诉发送端。

（5）拥塞控制

TCP传输的过程中，发送端开始发送数据的时候，如果刚开始就发送大量的数据，那么就可能造成一些问题。网络可能在开始的时候就很拥堵，如果给网络中在扔出大量数据，那么这个拥堵就会加剧。拥堵的加剧就会产生大量的丢包，就对大量的超时重传，严重影响传输。

所以TCP引入了慢启动的机制，在开始发送数据时，先发送少量的数据探路。探清当前的网络状态如何，再决定多大的速度进行传输。这时候就引入一个叫做拥塞窗口的概念。发送刚开始定义拥塞窗口为 1，每次收到ACK应答，拥塞窗口加 1。在发送数据之前，首先将拥塞窗口与接收端反馈的窗口大小比对，取较小的值作为实际发送的窗口。

如果把窗口定的很大，发送端连续发送大量的数据，可能会造成网络的拥堵（大家都在用网，你在这狂发，吞吐量就那么大，当然会堵），甚至造成网络的瘫痪。所以TCP在为了防止这种情况而进行了拥塞控制。

慢启动：定义拥塞窗口，一开始将该窗口大小设为1，之后每次收到确认应答（经过一个rtt），将拥塞窗口大小\*2。

拥塞避免：设置慢启动阈值，一般开始都设为65536。拥塞避免是指当拥塞窗口大小达到这个阈值，拥塞窗口的值不再指数上升，而是 加法增加（每次确认应答/每个rtt，拥塞窗口大小+1），以此来避免拥塞。

将报文段的超时重传看做拥塞，则一旦发生超时重传，我们需要先将阈值设为当前窗口大小的一半，并且将窗口大小设为初值1，然后重新进入慢启动过程。

快重传：在遇到3次重复确认应答（高速重发控制）时，代表收到了3个报文段，但是这之前的1个段丢失了，便对它进行立即重传。 ​ 快恢复：然后，先将阈值设为当前窗口大小的一半，然后将拥塞窗口大小设为慢启动阈值+3的大小。

这样可以达到：在TCP通信时，网络吞吐量呈现逐渐的上升，并且随着拥堵来降低吞吐量，再进入慢慢上升的过程，网络不会轻易的发生瘫痪。

### IMG_37114.DNS详细过程

（1）检查浏览器缓存、检查本地hosts文件是否有这个网址的映射，如果有，就调用这个IP地址映射，解析完成。 （2）如果没有，则查找本地DNS解析器缓存是否有这个网址的映射，如果有，返回映射，解析完成。 （3）如果没有，则查找填写或分配的首选DNS服务器，称为本地DNS服务器。服务器接收到查询时： 如果要查询的域名包含在本地配置区域资源中，返回解析结果，查询结束，此解析具有权威性。 如果要查询的域名不由本地DNS服务器区域解析，但服务器缓存了此网址的映射关系，返回解析结果，查询结束，此解析不具有权威性。 （4）如果本地DNS服务器也失效： 如果未采用转发模式（迭代），本地DNS就把请求发至13台根DNS，根DNS服务器收到请求后，会判断这个域名（如.com）是谁来授权管理，并返回一个负责该顶级域名服务器的IP，本地DNS服务器收到顶级域名服务器IP信息后，继续向该顶级域名服务器IP发送请求，该服务器如果无法解析，则会找到负责这个域名的下一级DNS服务器（如<http://baidu.com>）的IP给本地DNS服务器，循环往复直至查询到映射，将解析结果返回本地DNS服务器，再由本地DNS服务器返回解析结果，查询完成。

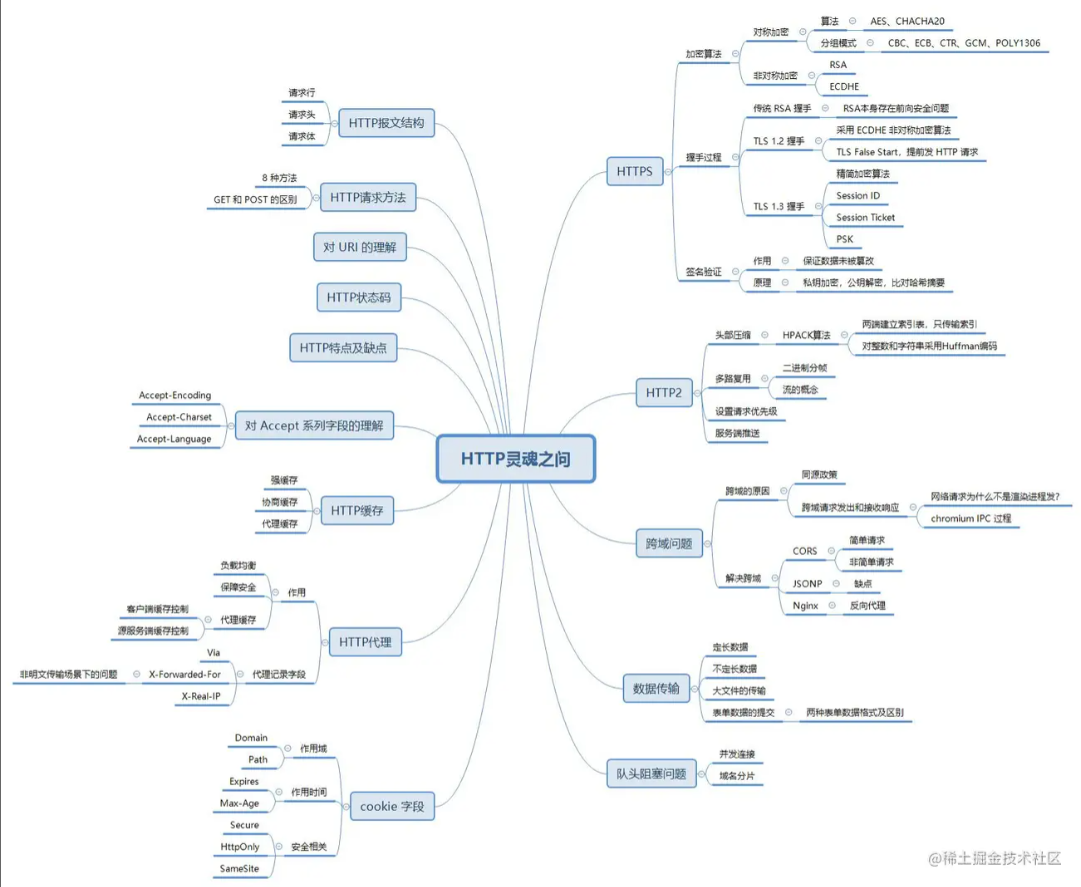
如果采用转发模式（递归），则此DNS服务器就会把请求转发至上一级DNS服务器，如果上一级DNS服务器不能解析，则继续向上请求。最终将解析结果依次返回本地DNS服务器，本地DNS服务器再返回给客户机，查询完成。

### 15.ip寻址过程

互联网是一个多级的网络，我们的个人电脑位于网络最底层，上层还有互联网服务提供商ISP，再上层是网络服务提供商NSP，NSP网络通过网络访问点NAP相连，来交换数据包流量。

互联网的每一级都会有路由器，路由器上有个路由表，记录了其子网络的所有IP地址。然而它并不知道上层网络所包含的IP地址，当数据包到达路由器，路由器检查路由表上是否有目的地的IP地址，如果有则直接发送给那个网络，否则就向上发送数据，在更高的层级去寻找拥有该IP的路由器，如果依然没有找到，则再次将数据包向上路由，直到到达NSP主干网为止，连接到NSP骨干网的路由器拥有最大的路由表，通过这张表可以将数据包路由到正确的骨干网，然后他将开始向下传播，进入越来越小的网络，直到找到目的地为止。

## Http



### HTTP 报文结构



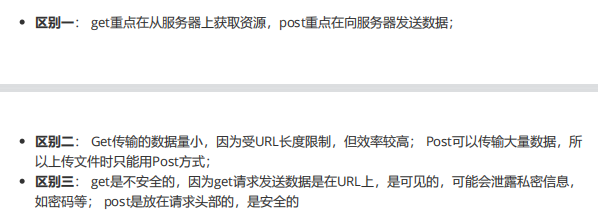


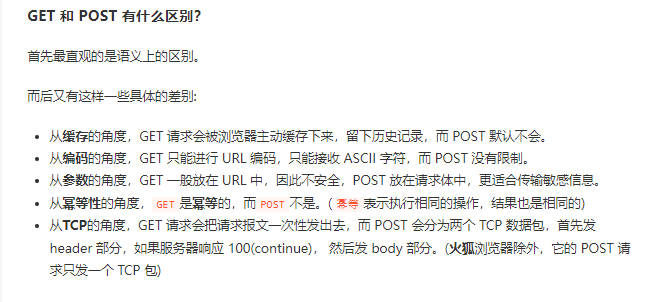


### HTTP请求所经历几个步骤



### GET和POST区别





### 状态码



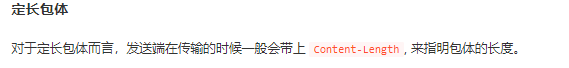


### URI

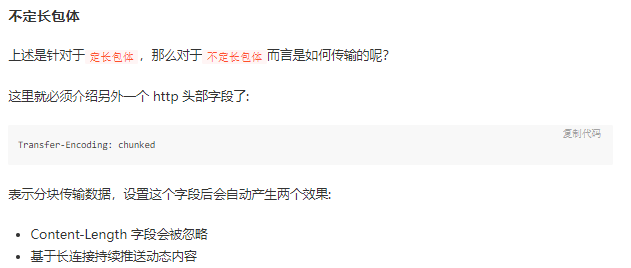




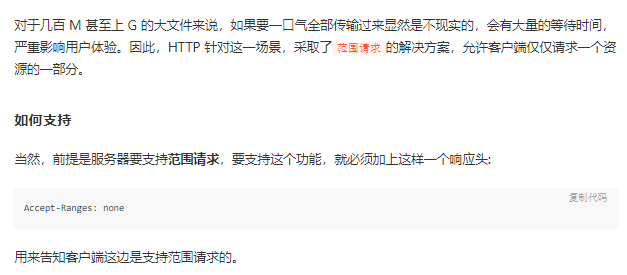
### 对于定长和不定长数据，HTTP怎么传输



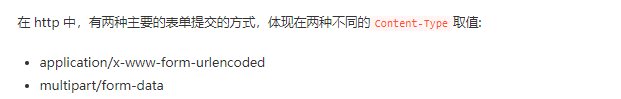
设置不当直接导致传输失败。

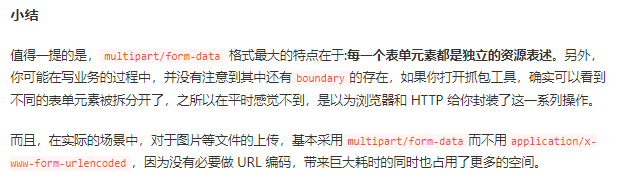


### Http如何处理大文件传输



### Http如何处理表单数据的提交

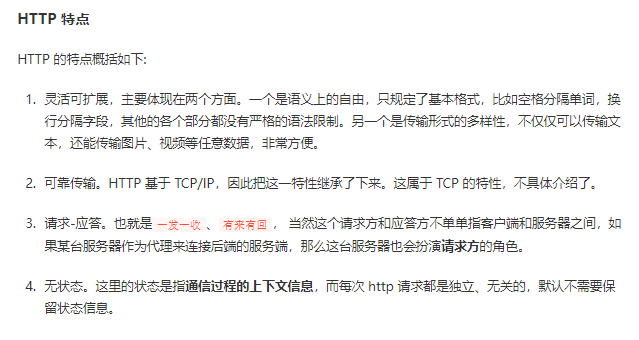


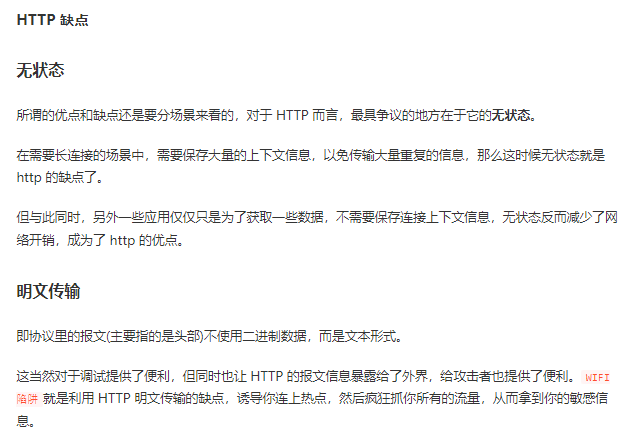


### 队头阻塞

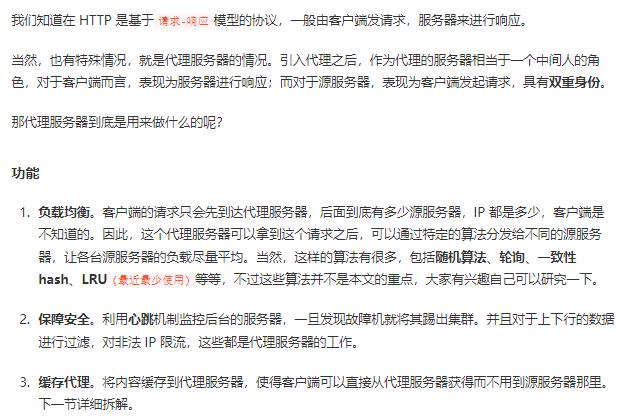


### Http特点

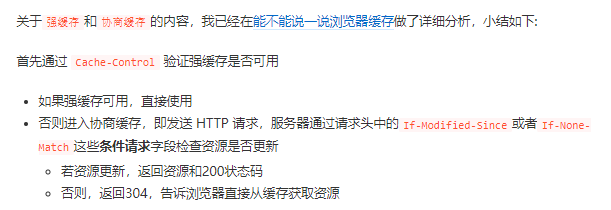


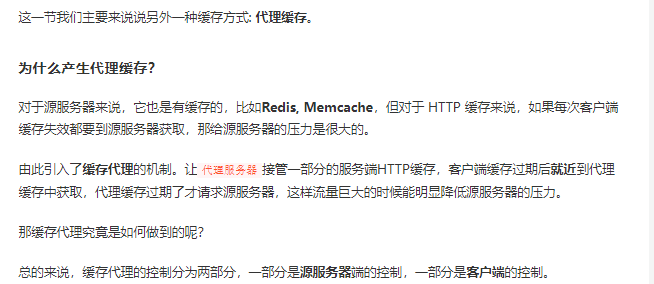


### Http代理



### HTTP 缓存及缓存代理

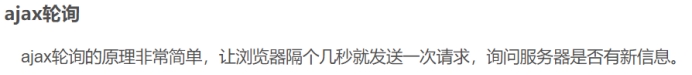


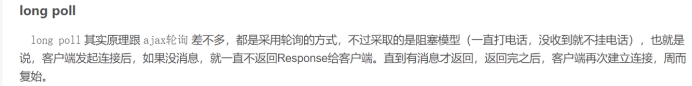


### websocket

IMG_392

IMG_393



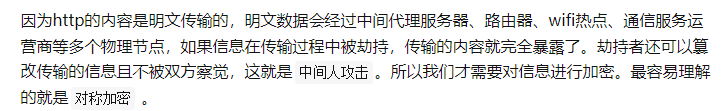


这两种方式都是非常消耗资源的

当服务器完成协议升级成Websocket后，服务端就可以主动推送信息给客户端（可以实现全双工通信），只需要经过一次HTTP请求，服务端就可以源源不断得向客户端传送信息。

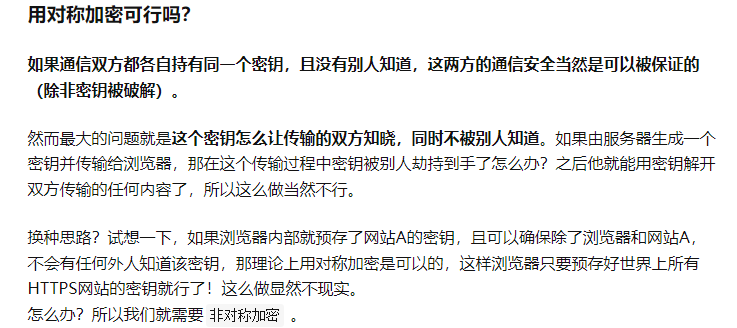
## HTTPS

### 为什么需要加密

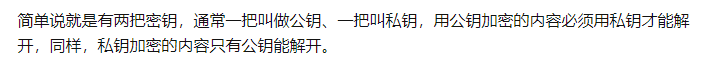


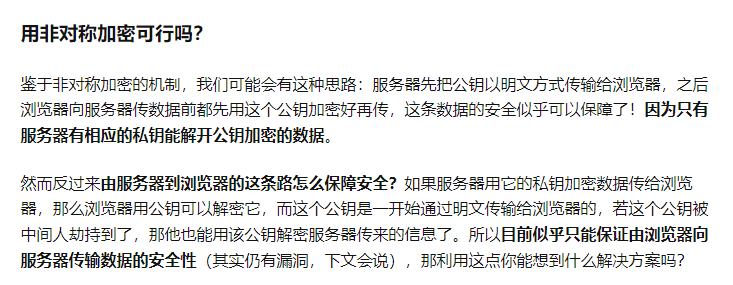
### 什么是对称加密

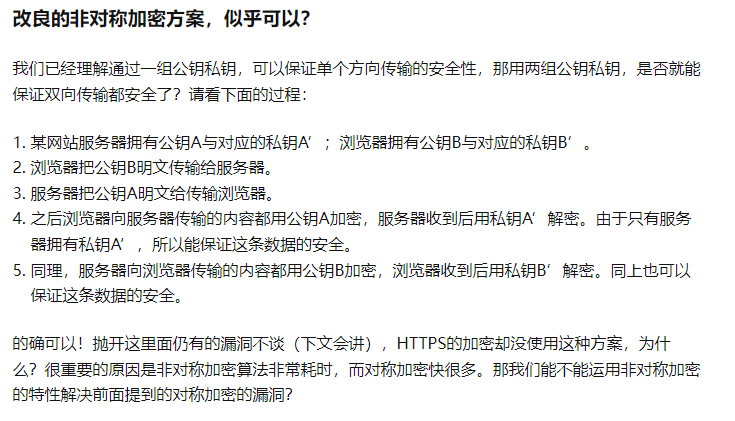




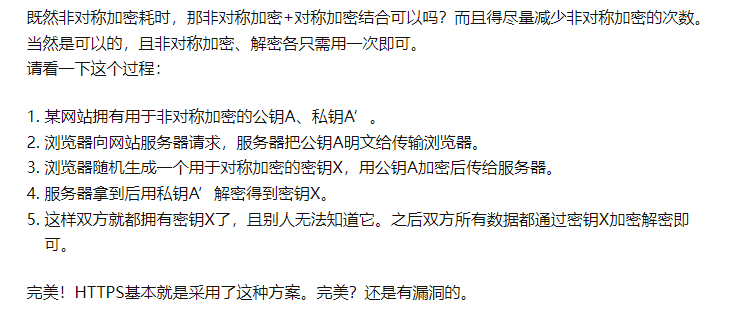
### 什么是非对称加密

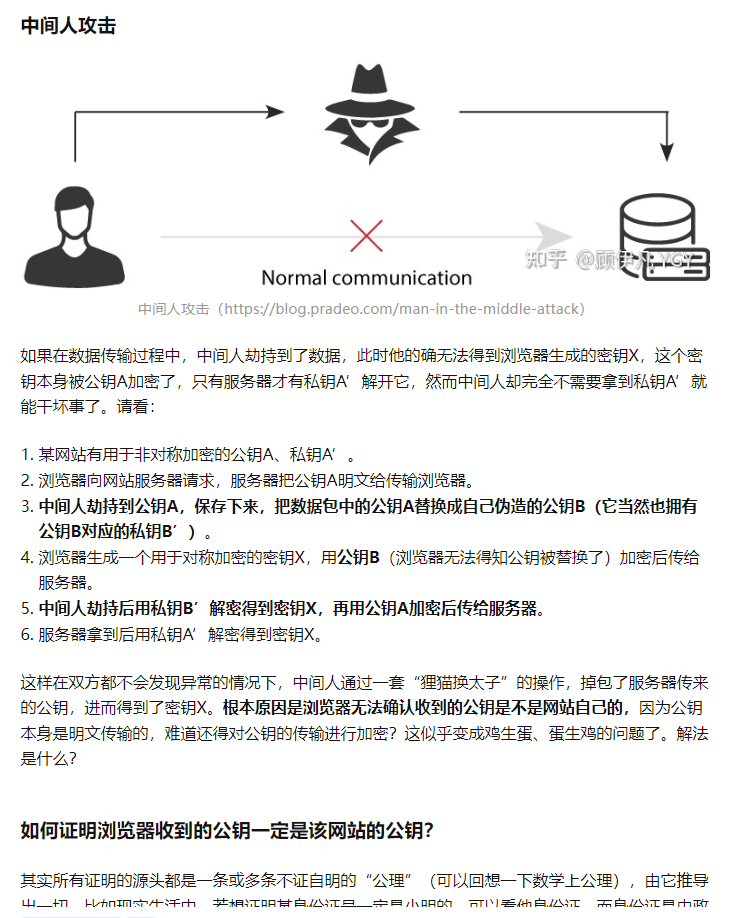




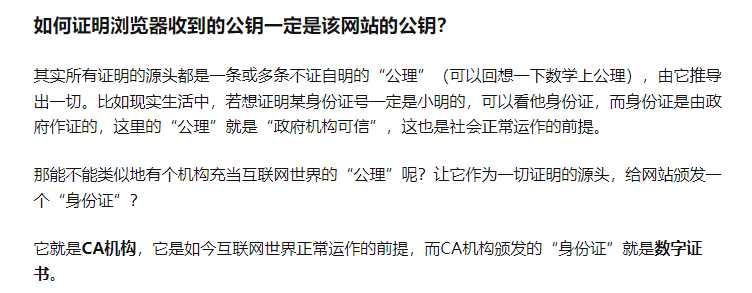


### 非对称+对称加密





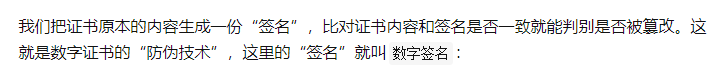
### 如何证明浏览器收到的公钥一定是该网站的公钥？



### 数字证书



### 防止数字证书被篡改



### 数字签名

颁发机构自有一套公钥和私钥

