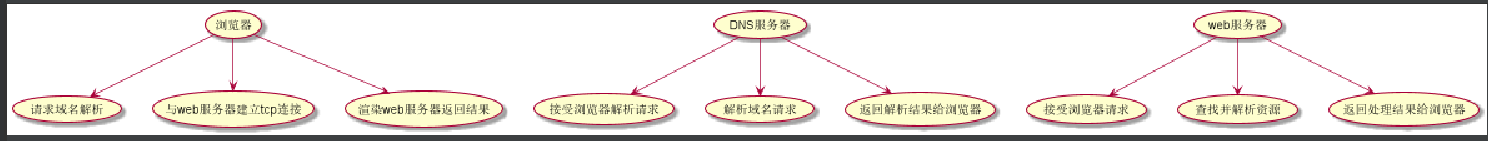
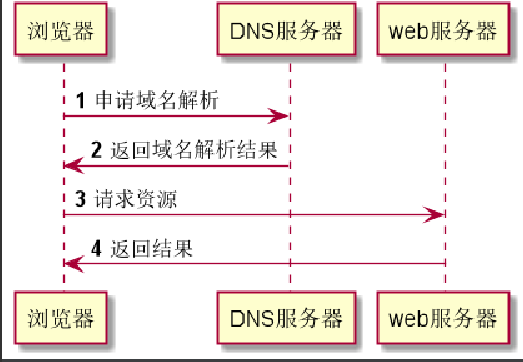
**键入 www.baidu.com流程探讨**

1. **整个流程总概叙：**
   1. **http的headr会给我们请求包装，比如AF中经常设置的可接受的Accept(text/html)-->域名解析，根据域名找到服务器的ip-->发起tcp的三次握手-->建立tcp链接后发起http请求-->服务器响应http请求-->浏览器得到html代码-->浏览器解析html并加载html代码中的资源（js,css,图片)-->浏览器对页面进行渲染呈现给用户。**
   2. **用图做如下描述**

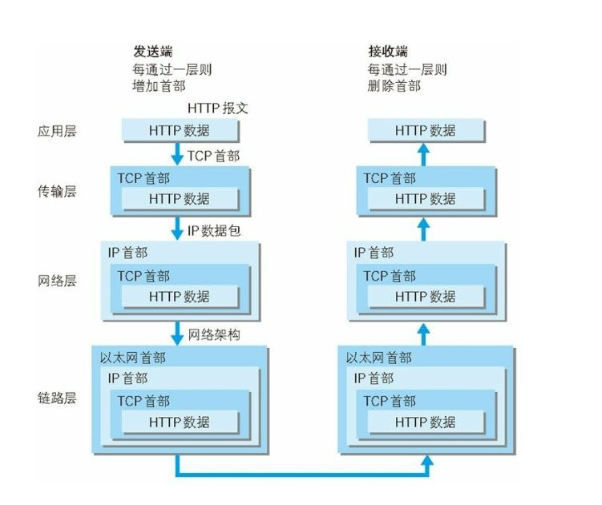




1. 从网络请求角度分析

每次请求的应用层http协议-->客户端的传输层（tcp协议）-->客户端的网络层（ip）协议-->客户端的链路层（网卡路由器）-->经过dns解析,穿越多个isp（互联网服务提供商，移动，联通，电信等）-->各种数据交换，找到了服务器------------------- 服务器的链路层 -->服务器的网络层 -->服务器的传输层 -->服务器的应用层

服务器响应相反



1. 里面关键详情探讨

3.1 域名解析（dns解析）：首先浏览器会解析www.baidu.com对应的ip。他需要做如下处理过程

3.1.1 Chrome浏览器 会首先搜索浏览器自身的DNS缓存（缓存时间比较短，大概只有1分钟，且只能容纳1000条缓存），看自身的缓存中是否有www.linux178.com 对应的条目，而且没有过期，如果有且没有过期则解析到此结束。 注：我们怎么查看Chrome自身的缓存？可以使用 chrome://net-internals/#dns 来进行查看

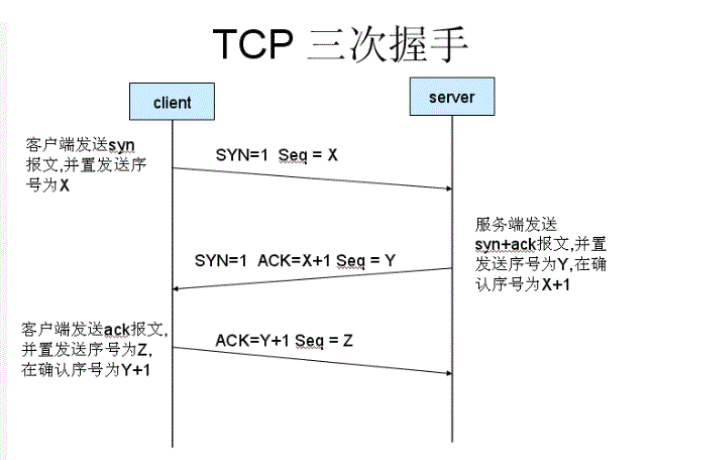
3.1.2 如果浏览器自身的缓存里面没有找到对应的条目，那么Chrome会搜索操作系统自身的DNS缓存,如果找到且没有过期则停止搜索解析到此结束.注：怎么查看操作系统自身的DNS缓存，以Windows系统为例，可以在命令行下使用 ipconfig /displaydns 来进行查看

3.1.3 如果在Windows系统的DNS缓存也没有找到，那么尝试读取hosts文件（位于C:\Windows\System32\drivers\etc），看看这里面有没有该域名对应的IP地址，如果有则解析成功

3.1.4 如果在hosts文件中也没有找到对应的条目，浏览器就会发起一个DNS的系统调用，就会向本地配置的首选DNS服务器（一般是电信运营商提供的，也可以使用像Google提供的DNS服务器）发起域名解析请求（通过的是UDP协议向DNS的53端口发起请求，这个请求是递归的请求，也就是运营商的DNS服务器必须得提供给我们该域名的IP地址），运营商的DNS服务器首先查找自身的缓存，找到对应的条目，且没有过期，则解析成功。如果没有找到对应的条目，则有运营商的DNS代我们的浏览器发起迭代DNS解析请求，它首先是会找根域的DNS的IP地址（这个DNS服务器都内置13台根域的DNS的IP地址），找打根域的DNS地址，就会向其发起请求（请问www.baidu.com这个域名的IP地址是多少啊？），根域发现这是一个顶级域com域的一个域名，于是就告诉运营商的DNS我不知道这个域名的IP地址，但是我知道com域的IP地址，你去找它去，于是运营商的DNS就得到了com域的IP地址，又向com域的IP地址发起了请求（请问www.baidu.com这个域名的IP地址是多少?）,com域这台服务器告诉运营商的DNS我不知道www.baidu.com这个域名的IP地址，但是我知道linux178.com这个域的DNS地址，你去找它去，于是运营商的DNS又向linux178.com这个域名的DNS地址（这个一般就是由域名注册商提供的，像万网，新网等）发起请求（请问www.baidu.com这个域名的IP地址是多少？），这个时候linux178.com域的DNS服务器一查，诶，果真在我这里，于是就把找到的结果发送给运营商的DNS服务器，这个时候运营商的DNS服务器就拿到了www.baidu.com这个域名对应的IP地址，并返回给Windows系统内核，内核又把结果返回给浏览器，终于浏览器拿到了www.baidu.com 对应的IP地址，该进行一步的动作了

3.2 发起tcp的3次握手

3.2.1 拿到域名对应的IP地址之后，User-Agent（一般是指浏览器）会以一个随机端口（1024 < 端口 < 65535）向服务器的WEB程序（常用的有httpd,nginx等）80端口发起TCP的连接请求。这个连接请求（原始的http请求经过TCP/IP4层模型的层层封包）到达服务器端后（这中间通过各种路由设备，局域网内除外），进入到网卡，然后是进入到内核的TCP/IP协议栈（用于识别该连接请求，解封包，一层一层的剥开），还有可能要经过Netfilter防火墙（属于内核的模块）的过滤，最终到达WEB程序（本文就以Nginx为例），最终建立了TCP/IP的连接。



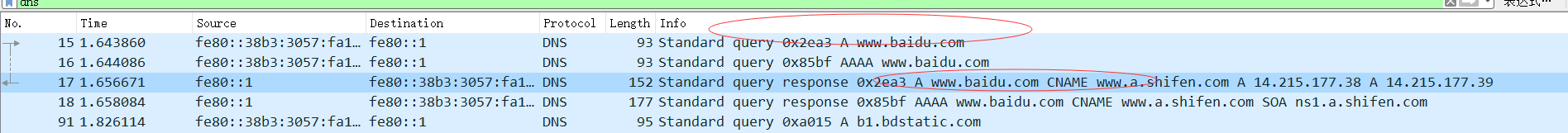
1） Client首先发送一个连接试探，ACK=0 表示确认号无效，SYN = 1 表示这是一个连接请求或连接接受报文，同时表示这个数据报不能携带数据，seq = x 表示Client自己的初始序号（seq = 0 就代表这是第0号包），这时候Client进入syn\_sent状态，表示客户端等待服务器的回复

2） Server监听到连接请求报文后，如同意建立连接，则向Client发送确认。TCP报文首部中的SYN 和 ACK都置1 ，ack = x + 1表示期望收到对方下一个报文段的第一个数据字节序号是x+1，同时表明x为止的所有数据都已正确收到（ack=1其实是ack=0+1,也就是期望客户端的第1个包），seq = y 表示Server 自己的初始序号（seq=0就代表这是服务器这边发出的第0号包）。这时服务器进入syn\_rcvd，表示服务器已经收到Client的连接请求，等待client的确认。

1. Client收到确认后还需再次发送确认，同时携带要发送给Server的数据。ACK 置1 表示确认号ack= y + 1 有效（代表期望收到服务器的第1个包），Client自己的序号seq= x + 1（表示这就是我的第1个包，相对于第0个包来说的），一旦收到Client的确认之后，这个TCP连接就进入Established状态，就可以发起http请求了。
2. 建立TCP连接后发起http请求

附加：实际操作模拟dns解析需要执行一下操作

1. 执行ipconfig/flushdns （刷新本地dns缓存）
2. 执行ipconfig /displaydns （查看dns列表）
3. 浏览器键入 [www.baidu.com](http://www.baidu.com) 查看抓包信息

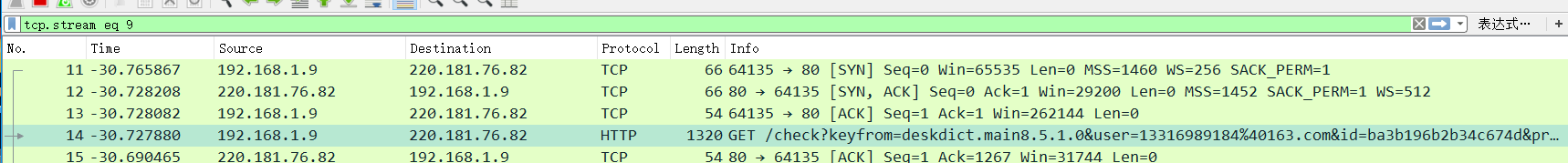


4.查看tcp三次握手信息

4.1 打开抓包软件，并键入www.baidu.com

4.2 过滤http协议

4.3 点击跟随 tcp流会执行如下操作



此处可以证明是基于tcp建立http请求

相关文档

1. 抓包软件官网文档：<https://www.wireshark.org/docs/wsdg_html_chunked/>
2. 抓包软件官网：<https://www.wireshark.org/>
3. 相关博客：<https://blog.csdn.net/u014600626/article/details/78720763>