浏览器中输入URL（假如输入的是www.anvilliu.com）

本地DNS设置中配置本地DNS服务器地址，一般由ISP通过DHCP协议动态分配（路由器可能自己内置了DNS转发器，那么本地DNS可能是路由器的地址如192.168.1.1），也可以手动修改为公共DNS，如114.114.114.114，由路由系统根据任播网络选择最近的服务器。

13台根服务器：说是13台，由于任播网络其实是13组，设置为13组是由于历史原因，由于DNS协议大多数情况下是基于UDP实现的，当返回响应大于512字节时会采用TCP传输（或区域传输的时候），并且根据IPv4协议的最小重组缓冲区的限制即576字节，减去IP报头60字节，减去udp报头8字节，剩下508字节即为UDP的安全有效载荷，在这个大小的UDP包可以保证通过IP交付，因为可以避免被IP分片，大部分Intenet的网络接口MTU（Maximum Transmission Unit最大传输单元）>512字节，那么512字节的包目前13台域名服务器的DNS响应报文已经占了436字节，剩下76字节可以增加两台，但是由于保守的需求，需要保留一些扩展空间，而且目前来说13组根服务器完全可以满足需求。

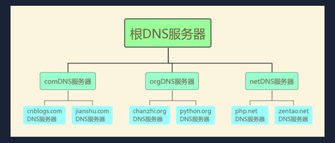
DNS为什么基于UDP：简单来说UDP通信只需要一个请求包，一个响应包，而TCP通信需要三次握手，一个请求包，一个响应包，四次挥手，UDP开销更小。但是随着网络的发展，这不绝对。

IP分片为什么不好：一个 UDP 报文如果因为 size > MTU，则会被 IP 层分成两片多片，但是只有一片有端口号，由于其它分片没有端口号，能否通过防火墙则完全看防火墙的脸色，所以对于能否通信成功是一个未知数。如果防火墙网开一面，不检查端口号，分片可以全部通行，到目的地再组装到一起，IP 层提交给 UDP/DNS，一点问题没有。但是防火墙的安全功能大打折扣，如何阻止非法的外来攻击包？如果防火墙严格检查端口号，则没有端口号的分片则统统丢弃，造成通信障碍。所以选择一个合适的 UDP size 至关重要，避免分片。

1. 查询浏览器缓存，如果有缓存则返回IP地址。否则进行2。

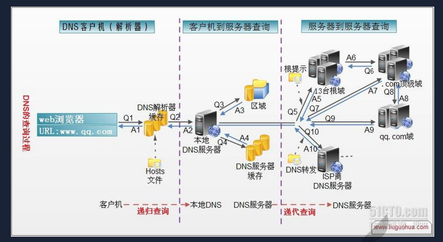
2. 查找系统缓存，如果本地的hosts文件中有域名对应的规则，则返回IP地址，否则就进行3。

3. 发送DNS请求到本地DNS服务器的地址，本地DNS服务器首先查询自己的本地DNS服务器区域解析和缓存记录，如果有记录，就返回域名对应的IP地址，否则根据本地DNS服务器的设置（是否设置转发器）进行查询。没设置转发则进行4，设置了转发进行5。



4. 非转发模式，本地DNS将请求发送至13台根服务器，根服务器会查询（.com）负责这个顶级域名的一个DNS服务器IP，并返回给本地DNS服务器，本地DNS服务器根据这个IP，联系负责.com域名的DNS服务器，该服务器收到请求后，如果无法自己解析，就会找到管理二级域名（anvilliu.com）的DNS服务器，将这个服务器的IP返回给本地DNS服务器，本地DNS服务器收到后再向该服务器发送请求，直到找到域名（www.anvilliu.com）所对应的IP。

5. 转发模式：本地DNS服务器会把请求转到上一级DNS服务器，由上一级服务器进行解析，若上一级服务器无法解析，就递归转发到上上级，直到找到对应IP，则返回给本地服务器。



DNS解析：从客户端到本地DNS服务器是属于递归查询，而DNS服务器之间就是的交互查询就是迭代查询。

建立TCP连接

以随机端口（1024~65535）向WEB服务器的80（http）/443（https）端口的服务器守护进程（httpd/nginx，守护进程可看TCP/IP协议的pdf）发起TCP连接请求，进行三次握手（三次握手的具体可看TCP/IP协议pdf）

建立TCP连接涉及到很多传输层知识点，其中除了上面提到的面试必考的三次握手外，还有，只有第一次SYN不包含ACK字段，握手后可能会生成对应的NAPT表，在TCP连接建立的过程中两端主机会计算出MSS（“最大消息长度”），即数据报的最大不会被IP分片的长度，MSS被包含在TCP首部。

下到网络层，TCP包封装好后来到IP协议，IP协议和UDP类似是无连接不可靠协议，IP协议要打上自己的包装，IP首部（本机IP地址由DHCP服务器如何分配），但是目前只有目标IP地址的话，还无法送到目的地，因此要根据ARP协议查询对应的主机的MAC地址（MAC地址在网络层查询，数据链路层使用），ARP的查询包不知道MAC地址，因此是用广播的形式发出（广播是什么形式？多播呢？）其中又涉及到同一链路和不同链路下ARP包的传递，在路由器中会形成ARP表负责转发ARP包，还可以设置代理ARP服务转发被隔离的ARP广播到邻近网段。（谈到ARP协议就还有RARP协议用来对MAC地址分配IP地址），在网络层，是将IP包转发到不同的数据链路中，因此还涉及到路由协议、路由选择算法和路由选择表。不同的网段之间，可能使用的IP协议版本不同，有的是IPv4有的是IPv6，这就需要用IP隧道传输。还有移动IP，IP多播，任播等技术应对不同的传输要求和场景。IPv6是根本上解决IP地址不够用的办法，IPv4也提供了子网掩码，NAT，NAPT来解决IP地址耗尽的问题（一个全局IP地址对应多个子网），如果传输遇到问题或者需要检验异常，ICMP协议会告知发送主机（ICMP有很多用处），详情可看TCP和IP协议簇pdf。（此处讲解多以IPv4为主）

当获取到了目标MAC地址后，IP包来到数据链路层，在这里要打上以太网首部和数据链路层尾部（FCS位，用于CRC检测传输干扰对数据是否损坏），数据链路层中的网桥可以将帧对应MAC地址转发表转发到对应端口的主机或者路由器。网桥会对数据进行校验，若数据帧被破坏则会丢弃。通过物理层的传输介质传给下一跳。

比特经过物理层输送，在接收端最终收到数据帧，接收端会首先判断是否是自己的包，是则传给对应的网络层设备或者主机，最终经过转发会来到接收端主机的网络层，以IP协议为例子，接收端首先判断IP地址是否匹配，然后传给对应的传输层协议处理，TCP协议根据首部校验和判断数据有效性，最终可以给发送端主机以TCP确认响应（TCP三次握手的第二次），再经历上述过程。

HTTPS/HTTP请求

HTTPS比HTTP在正式第一次请求之前，要多一个建立SSL安全隧道的过程，保证HTTP报文是加密传输的。（具体步骤可看HTTP协议pdf）

HTTP协议进行了多次改进，由最初HTTP1.0-HTTP1.1-结合SPDY-HTTP2.0，最新的HTTP2.0协议相当于将SPDY并入HTTP1.1并有些改进（具体可看HTTP协议pdf）

当TCP连接（SSL隧道）建立后，会发出第一个HTTP请求，HTTP编码数据后根据HTTP协议进行必要包装，加上HTTP请求头，就开始进入上一步的发送步骤，但是这一次带有数据，首次数据大小是以MSS确定，之后的发送大小首先以一次ICMP协议进行路径MTU发现，避免数据遭到IP分片，减轻路由器负担（IPv6会在主机分段）如果TCP建立连接时就已经计算了MSS则可以保证大小不会再被IP分片。往往一个报文很大，远超过一次TCP能传输的大小，因此要很多次TCP传输才能传完，TCP有慢启动机制（拥塞窗口）避免激增导致的网络拥塞，有滑动窗口机制，窗口控制，重发控制机制保证网络利用率和数据吞吐量。后面的步骤和TCP连接请求的传输一样。

HTTPS/HTTP响应

服务端应用程序收到请求并进行解码，对请求内容响应并编码为HTTP报文后，返回响应。HTTPS响应同样在SSL隧道中被加密，响应与请求在HTTP1.0中是一对一的，即一个请求后必须一个响应再请求，这很明显没有利用好网络资源，因此HTTP1.1中加入了pipelining，把多个HTTP请求方到一个TCP中发送，但是客户端还是以请求顺序接受响应，因此会出现线头阻塞（或者叫队头阻塞）的问题，因此HTTP2.0中引入了新的编码结构，多路复用方式来解决这个问题。（HTTP协议不同版本的不同可看HTTP协议pdf）

连接维持

HTTP1.1开始默认连接为持久连接，这使得pipelining得以实现，这样的好处有如避免了TCP慢启动阶段，节约了带宽和时间等（具体可看HTTP协议pdf）再HTTP1.0+中可以通过Connection:Keep-Alive首部支持长连接。

浏览器会发送TCP keep-alive探测包，判断TCP连接状况，在服务端Nginx会监听连接时长，时长可自行设置。

断开连接

HTTPS连接中客户端要发送close_notify报文来断开连接，之后进行四次挥手（可看TCP和IP协议pdf）