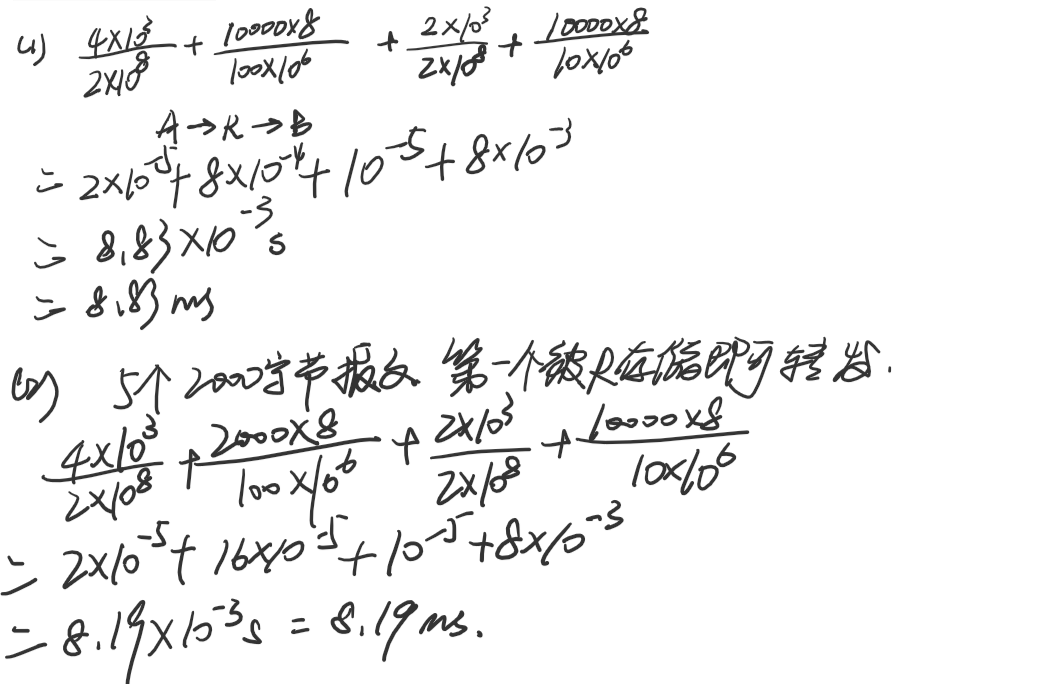
## 书面作业一

学号：2013018 姓名：许健 专业：2013018

第一章



影响端到端时延的主要因素包括传播时延PROP=L/C、传输时延TRANSP=P/R，单一链路上端端时延Latency = PROP + TRANSP

如果使用报文分组转发可以避免某条链路或路由器缓存被一个报文长时间占用，从而提高链路利用率。

在给定链路长度和传输速率的情况下，只有排队时延是变化量，会对网络性能产生较大的影响，因此网络中需要有拥塞控制机制。

第二章

1. **通过使用Windows命令行模式提供的nslookup命令查询www.baidu.com的IP地址，给出结果截图，并对返回的结果进行解释。同时，利用Wireshark捕获查询的交互过程，给出结果截图，并进行简要说明。**

**使用nslookup查询www.baidu.com的IP地址得到的结果**



服务器：本机DNS服务器信息

Address：当前计算机的DNS服务器地址

非权威应答：Non-authoritative answer，除了实际存储DNS Server中获得域名解析回答的，都称为非权威应答，也就是从DNS服务器缓存中获取域名解析结果。客户端接受非授权响应。如果注重准确性，客户机可以再联系授权服务器，验证映射是否仍有效。

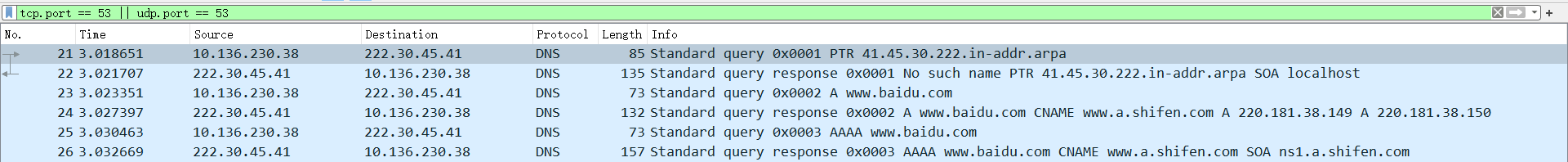
名称：域名（domain name）。

Addresses：目标域名对应物理IP可有多个。

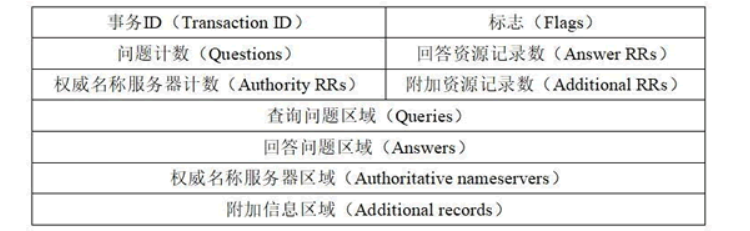
Aliases：别名

**使用wireshark捕获到查询的交互过程**

尽管DNS同时占用TCP和UDP的53号端口，但是DNS包长度小于512字节时还是使用的UDP。



DNS报文格式



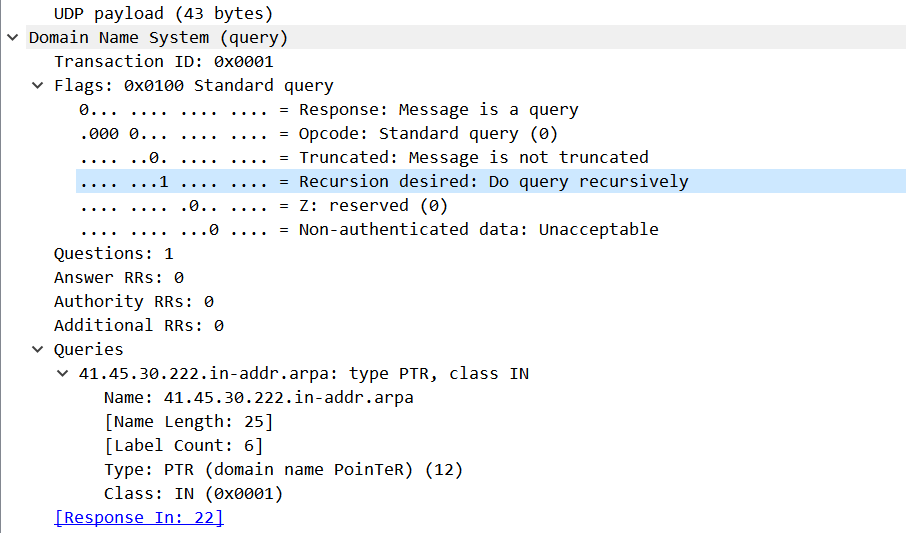
第一个报文的标志字段标识使用递归查询，包含一个查询问题

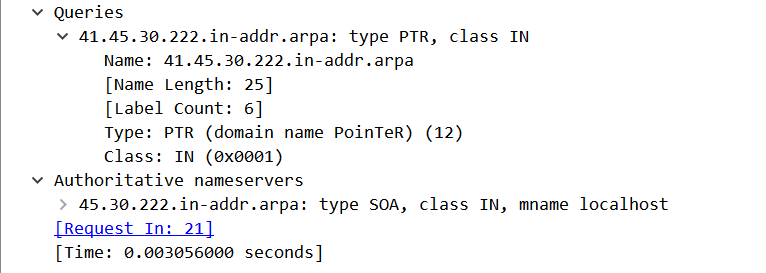
41.45.30.222.in-addr.arpa: type PTR, class IN

第二个报文标识支持递归查询，并给出回复

这两个报文和DNS服务器的IP地址有关，涉及类型PTR和SOA

与我们查询百度的IP无关，不作为重点

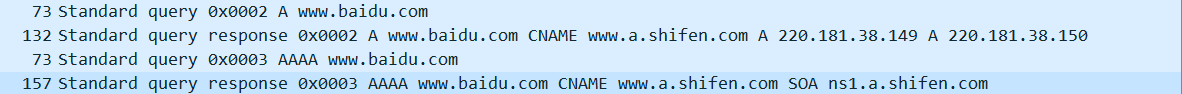


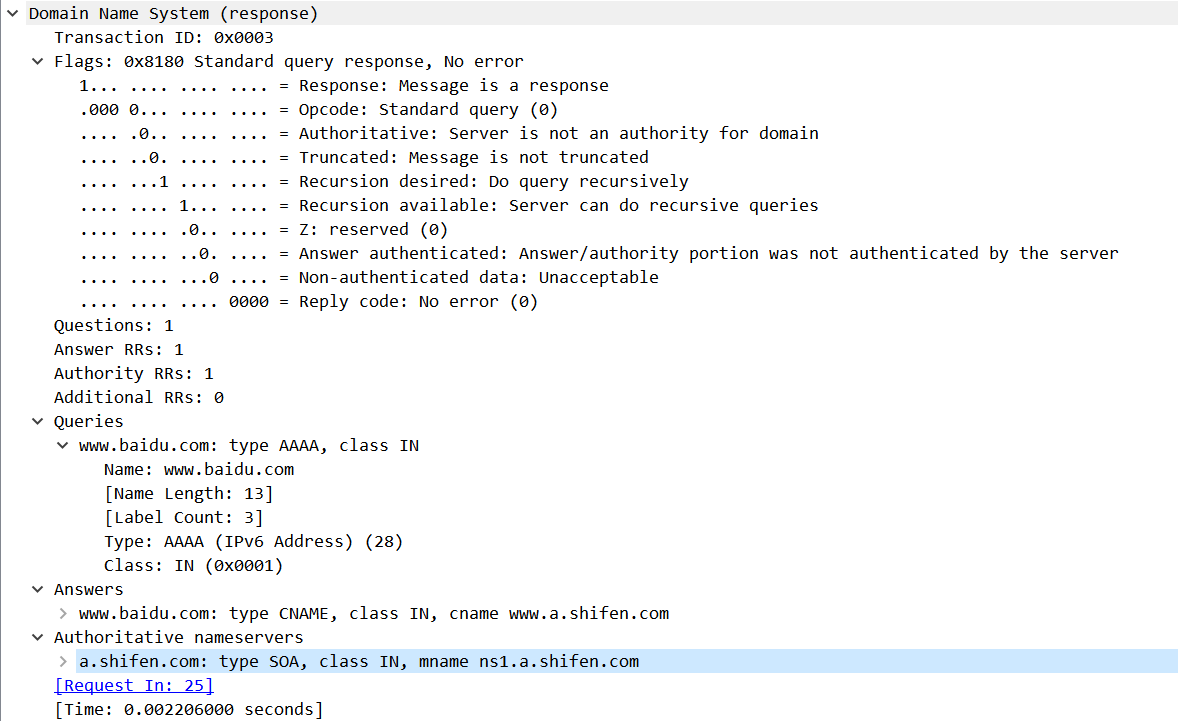


接下来的四个报文请求www.baidu.com的IPv4地址（type A）和IPv6地址(type AAAA)

本地DNS服务器给出回应，cname标识对应主机的规范主机名，为www.shifen.com。并返回两个可以的IP地址。

当查询IPv6时（type AAAA），本地DNS服务器返回SOA记录，表示此域名的权威解析服务器地址。当要查询的域名在所有递归解析服务器中没要域名解析的缓存时，就会回源来请求此域名的SOA记录，也叫权威解析记录。



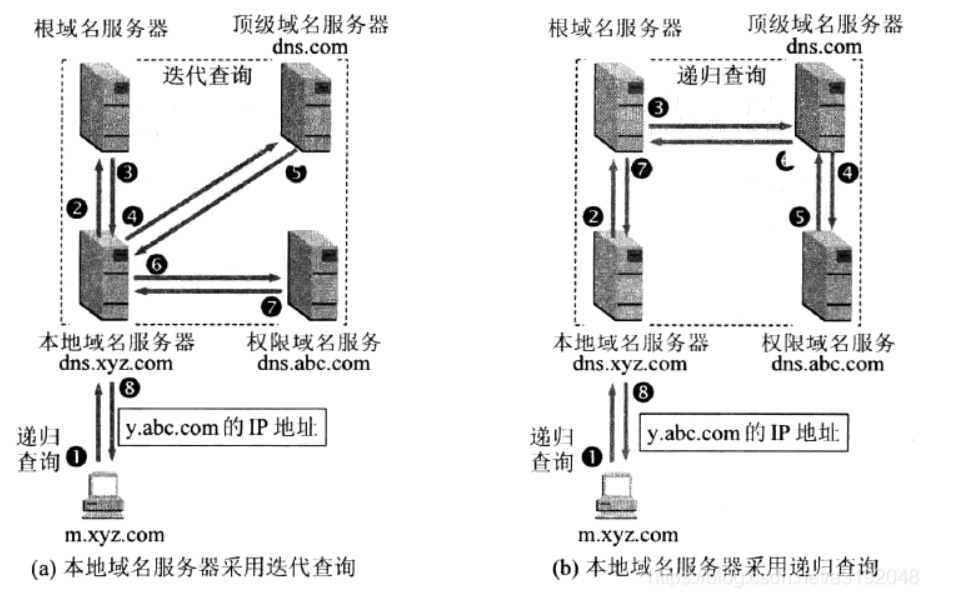


1. **以反复解析为例，说明域名解析的基本工作过程（可以结合图例）。给出内容分发网络（CDN）中DNS重定向的基本方法，说明原始资源记录应该如何修改，并描述重定向过程。**

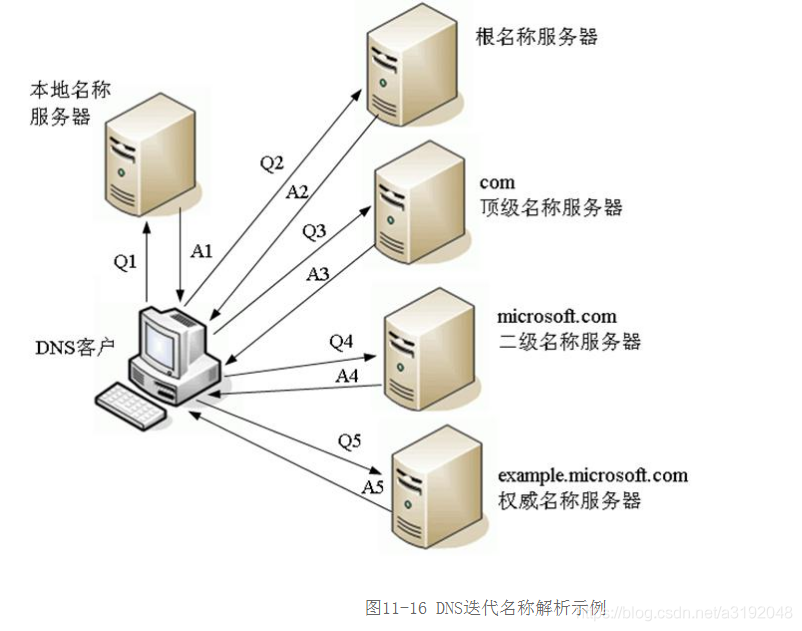
**DNS递归解析，可以分2种情况（老师上课讲的），即：**

1. DNS客户端和本地名称服务器是**递归**，而本地名称服务器和其他名称服务器之间是**迭代**。
2. DNS客户端和本地名称服务器是**递归**，而本地名称服务器和其他名称服务器之间也是**递归**。

**DNS默认是递归查询，可能是DNS客户端为了节省自己的CPU，内存等各方面资源。**



**迭代（反复）解析的流程**



****这里的迭代指DNS客户端向本地名称服务器（也叫本地域名服务器）查询是迭代。****

1. 客户端向本机配置的本地名称服务器发出DNS域名查询请求。
2. 本地名称服务器收到请求后，先查询本地的缓存，如果有该域名的记录项，则本地名称服务器就直接把查询的结果返回给客户端；如果本地缓存中没有该域名的记录，则向DNS客户端返回一条DNS应答报文，报文中会给出一些参考信息，如本地名称服务器上的根名称服务器地址等。
3. DNS客户端在收到本地名称服务器的应答报文后，会根据其中的根名称服务器地址信息，向对应的根名称服务器再次发出与前面一样的DNS查询请求报文。
4. 根名称服务器在收到DNS查询请求报文后，通过查询自己的DNS数据库得到请求DNS域名中顶级域名所对应的顶级名称服务器信息，然后以一条DNS应答报文返回给DNS客户端。
5. DNS客户端根据来自根名称服务器应答报文中的对应顶级名称服务器地址信息，向该顶级名称服务器发出与前面一样的DNS查询请求报文。
6. 顶级名称服务器在收到DNS查询请求后，先查询自己的缓存，如果有所请求的DNS域名的记录项，则相接把对应的记录项返回给DNS客户端，否则通过查询后把对应域名中二级域名所对应的二级名称服务器地址信息以一条DNS应答报文返回给DNS客户端。
7. 然后DNS客户端继续按照前面介绍的方法一次次地向三级、四级名称服务器查询，直到最终的权威名称服务器返回到最终的记录。

**两种情况下使用迭代查询：**

1. 在查询本地名称服务器时，如果客户端的请求报文中没有申请使用递归查询，即在DNS请求报头部的RD字段没有置1。
2. 客户端在DNS请求报文中申请使用的是递归查询（也就是RD字段置1了），但在所配置的本地名称服务器上是禁用递归查询（DNS服务器一般默认支持递归查询的），即在应答DNS报文头部的RA字段置0。

**内容分发网络（CDN）中DNS重定向的基本方法**

对于包含CDN的域名解析过程：

1.浏览器要将域名解析为 IP 地址，所以先向本地 DNS 发出请求。

2.本地 DNS 依次向根服务器、顶级域名服务器、二级域名服务器、三级域名服务器发出请求，得到全局负载均衡系统（GSLB）的 IP 地址。

3.本地 DNS 再向 GSLB 发出请求，GSLB 可以根据本地 DNS 服务器的 IP 地址判断用户的位置，筛选出距离用户较近的本地负载均衡系统（SLB），并将该 SLB 的 IP 地址作为结果返回给本地 DNS。

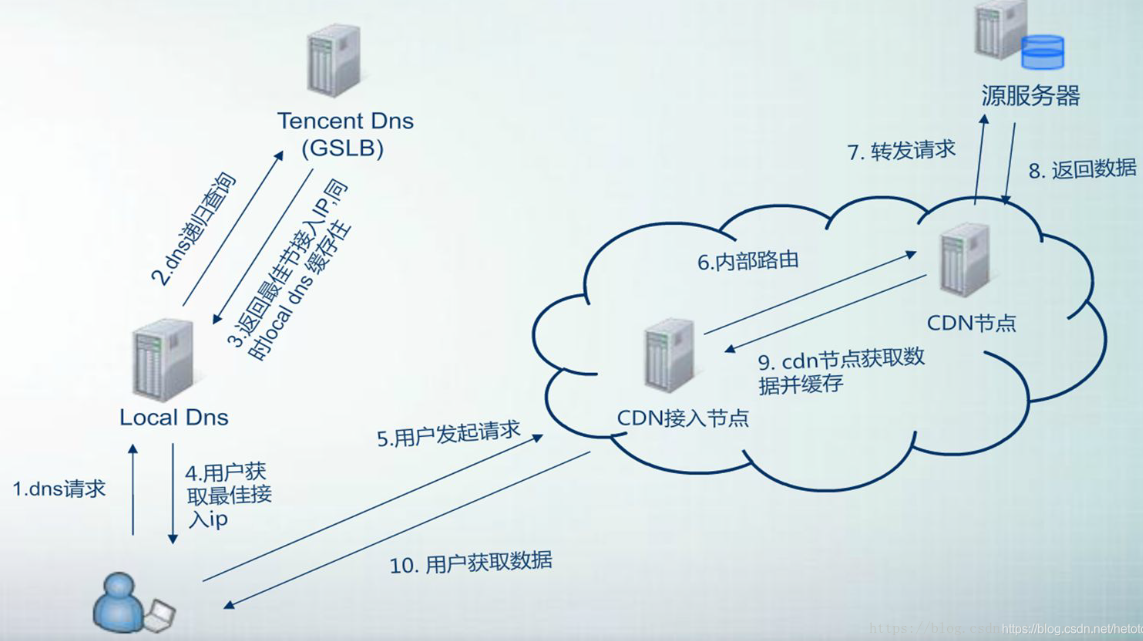
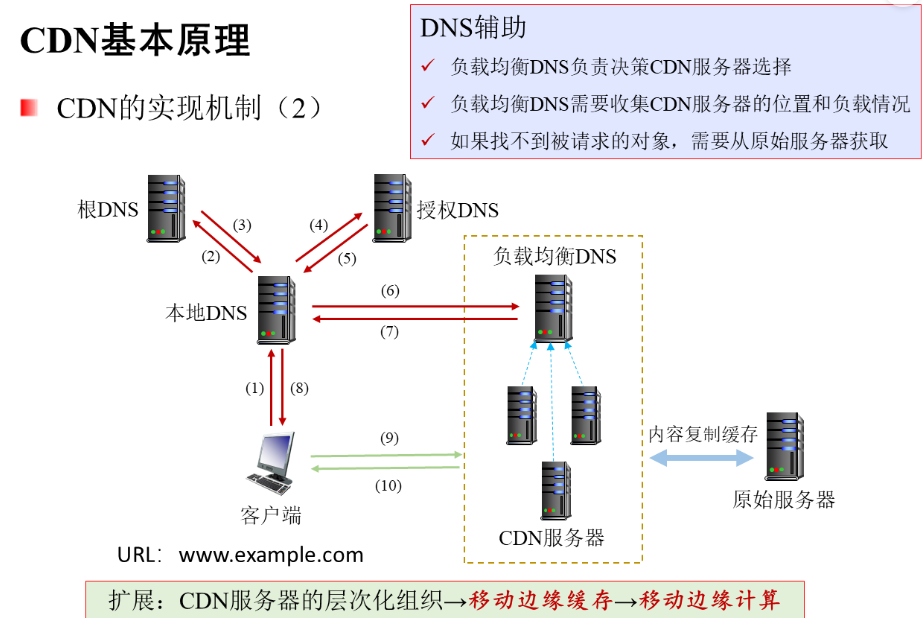
4.本地 DNS 将 SLB 的 IP 地址发回给浏览器，浏览器向 SLB 发出请求。

5.SLB 根据浏览器请求的资源和地址，选出最优的缓存服务器发回给浏览器。

6.浏览器再重定向到缓存服务器。

7.如果缓存服务器有浏览器需要的资源，就将资源发回给浏览器。

8.如果没有，就向更高一级的CDN缓存服务器请求，（直到）向源服务器请求资源，再一级一级发回给浏览器并缓存在本地。



**对于原始记录的修改**

1、要加入CDN服务的网站，需要域名(如www.linuxaid.com.cn, 地址202.99.11.120)解析权提供给CDN运营商，Linuxaid的域名解析记录只要把www主机的A记录改为CNAME并指向cache.cdn.com即可。cache.cdn.com是CDN网络自定义的缓存服务器的标识。在/var/named/linuxaid.com.cn域名解析记录中，由：

www IN A 202.99.11.120  
改为  
www IN CNAME cache.cdn.com.

2、CDN运营商得到域名解析权以后，得到域名的CNAME记录，指向CDN网络属下缓存服务器的域名，如cache.cdn.com，CDN网络的全局负载均衡DNS，需要把CNAME记录根据策略解析出IP地址，一般给出就近访问的Cache地址。

1. **在DNS域名系统中，域名解析时使用UDP协议提供的传输层服务（DNS服务器使用UDP的53端口），而UDP提供的是不可靠的传输层服务，请你解释DNS协议应如何保证可靠机制。**

**关于DNS的安全性问题**

起初，DNS (Domain Name System) 在设计上面并没有过多的去考虑安全的问题，也没有去考虑解析结果会不会被挟持的问题。当时的网络环境比较良好、安全，况且能接入互联网的人数并不多。但是随着科技的日益发展，接入互联网的人数也越来越多，网络环境也开始变得复杂起来，一些不怀好意的人出于经济目的抑或是政治目的，对互联网实施各种的挟持攻击，其中就包括了 DNS 的挟持。

众所周知，通用情况下用户与 DNS 服务器之间采用 UDP (用户数据报协议) 进行通讯。 UDP 并不像 TCP 那样有三次握手，也没有结束时候的握手，这在一定程度上面给解析带来的效率，但是这却留下了安全隐患。DNS 通讯时使用明文通讯，这就意味着中间设备都可以轻松地抓到你的 DNS 请求以及回复，也可以轻松地对此进行修改，伪造一份来自 DNS 服务器的查询结果（因为 UDP 协议的天性），并且将正确结果的包给 DROP 掉，这样用户就到达不了他真正想去的网站了。

域名系统安全扩展（Domain Name System Security Extensions，缩写为 DNSSEC）对 DNS提供给DNS客户端（解析器）的DNS数据来源进行认证，并验证不存在性和校验数据完整性验证，**但不提供机密性和可用性。**

简单来说，DNSSEC 就是一个对现有DNS协议进行安全完善的拓展。它在现有的DNS协议的基础上，增加了几个新的资源记录来达到这个目的。其主要改动包括资源记录签名(RRSIG)、DNS公钥(DNSKEY)、授权签名者(DS)和Next Secure (NSEC)。它还添加了两个新的消息头位:检查禁用数据(CD)和验证数据(AD)。最后，DNSSEC需要支持DNSSEC OK (DO) EDNS头位([RFC3225])，以便安全感知解析器可以在其查询中指出它希望在响应消息中接收DNSSEC RRs。

DNSSEC存在以下缺点：

1. 无法保证私密性。DNSSEC 并没有改变 DNS 基于 UDP 的通讯方式，数据流也都是明文传输，他所做的只是加上了一个数字签名，而中间人依然可以看到你请求了什么、结果是什么
2. 挟持发生时不能告诉用户真正的记录。当用户的 DNS 被挟持的时候，用户通过检查 DNSSEC 签名，可以知道自己得到的并不是真正的解析结果，而是得到了一个被伪造的地址。但是，用户并不知道真正的解析结果是什么。

**使用TCP传输**

DNS 查询由于 DNSSEC 和 IPv6 的引入迅速膨胀，导致 DNS 响应经常超过 MTU 造成数据的分片和丢失，我们需要依靠更加可靠的 TCP 协议完成数据的传输。DNS同时占用[UDP](https://so.csdn.net/so/search?q=UDP&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/ttyy1112/article/details/_blank)和TCP的53号端口。客户端默认通过 UDP 协议进行通讯，但是由于广域网中不适合传输过大的 UDP 数据包，因此规定当报文长度超过了 512 字节时，应转换为使用 TCP 协议进行数据传输。

可能会出现如下的两种情况：

1. 客户端认为 UDP 响应包长度可能超过 512 字节，主动使用 TCP 协议；
2. 客户端使用 UDP 协议发送 DNS 请求，服务端发现响应报文超过了 512 字节，在截断的 UDP 响应报文中将 TC 设置为 1 ，以通知客户端该报文已经被截断，客户端收到之后再发起一次 TCP 请求。

**DNS加密**

由于 DNS 查询请求和响应在默认情况下以明文方式传输，很容易带来安全隐患，因此未来势必会通过对DNS加密增加安全性。对DNS的加密有两种选择：侧重隐私性的 DoH(DNS over HTTPS,超文本传输协议安全）或侧重安全性的 DoT(DNSover TLS，传输层安全）。

DoT 使用 TLS 协议来传输 DNS 协议。TLS 协议是目前互联网最常用的安全加密协议之一，我们访问 HTTPS 的安全基础就是基于 TLS 协议的。相比于之前使用无连接无加密的 UDP 模式， TLS 本身已经实现了保密性与完整性。DoH 使用 HTTPS 来传输 DNS 协议，其安全原理与 DoT 一样，他们之间的区别只在于：DoH 有了HTTP格式封装，更加通用。

**DNSDEA（2019）**

DNSDEA(DNS dataencryption algorithm)。DNSDEA 采用非对称加密算法的密文方式通信，通过 PKI 加密体系与 DNS协议相融合，弥补了传统 DNS 体系域名隐私保护的问题，且与传统 DNS 体系有良好的兼容性。客户端的公钥通过 DNS 查询传输，降低了基于 TLS 等方法建立链路的开销，有效保持了 DNS 低延迟的特性。