# 应用层协议

在传输层之上，便是应用层。传输层的 UDP 报文和 TCP 报文段的数据部分就是应用层交付的数据。

不同类型的网络应用有不同的通信规则，因此应用层协议是多种多样的，比如 DNS、FTP、Telnet、SMTP、HTTP、RIP、NFS 等协议都是用于解决其各自的一类问题。

本节实验，介绍 DNS、FTP、HTTP 三个常用的应用层协议。

## 一、DNS

DNS (Domain Name Service 域名服务) 协议基于 UDP，使用端口号 53。

由数字组成的 IP 地址很难记忆，所以我们上网使用网站 IP 地址的别名——域名。实际使用中，域名与 IP 地址是对应的，这种对应关系保存在**DNS 服务器**之中。

在浏览器中输入一个域名后，会有 DNS 服务器将域名解析为对应的 IP 地址。注意这和网络层的 ARP 协议的不同之处：DNS 提供的是域名与 IP 地址的对应关系，而 ARP 提供的是 IP 地址和 MAC 地址的对应关系。

**DNS 服务器**

DNS 服务器是个分层次的系统：

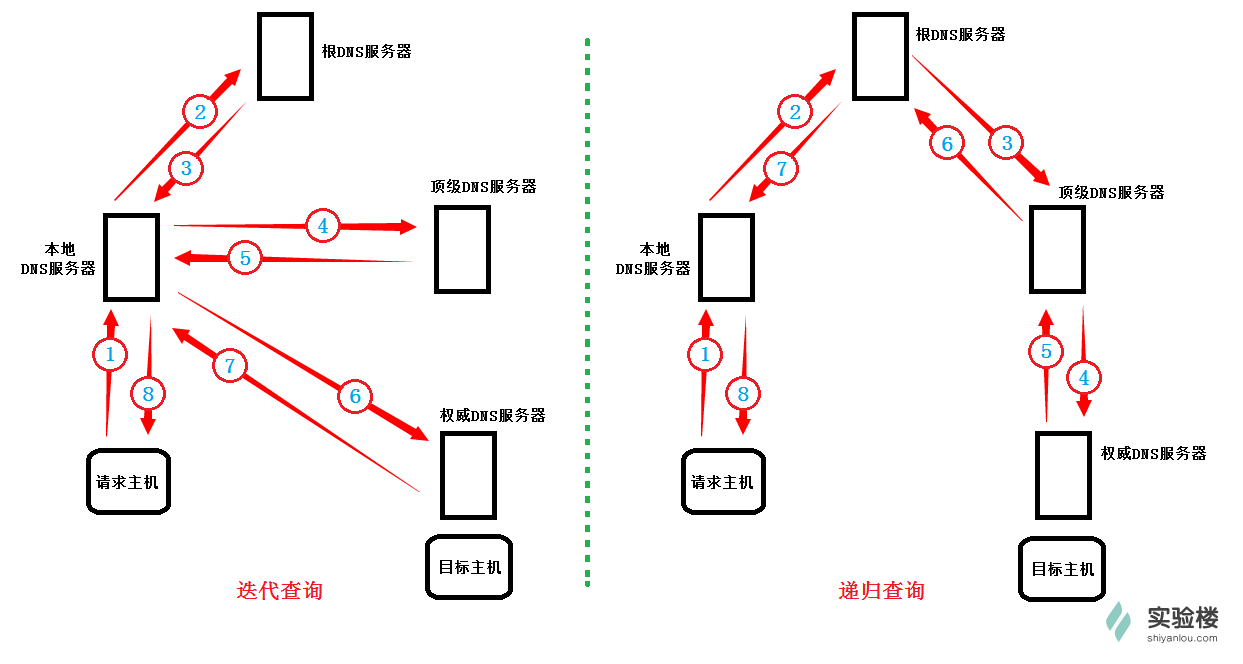
(1)**根 DNS 服务器** ：全世界共有 13 台根域名服务器，编号 A 到 M，其中大部分位于美国。

(2)**顶级(TLD)DNS 服务器** ：负责如 com 、org 、edu 等顶级域名和所有国家的顶级域名(如 cn 、uk 、jp )。

(3)**权威 DNS 服务器** ：大型组织、大学、企业的域名解析服务。

(4)**本地 DNS 服务器** ：通常与我们主机最近的 DNS 服务器。

而域名解析的过程，有迭代查询和递归查询两种方式：

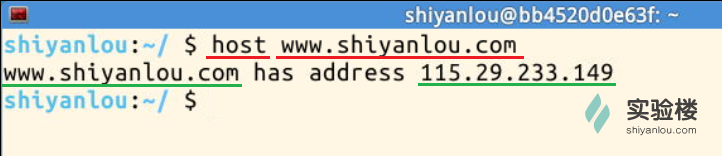


**host 命令**

在 linux 系统中，可以用 **host 命令** 进行 DNS 查询，查看一个指定域名的 IP，比如要查询[实验楼 GIT 服务](https://www.shiyanlou.com/courses/98/labs/565/git.shiyanlou.com" \t "/home/jason/文档\\x/_blank)的 IP 地址：

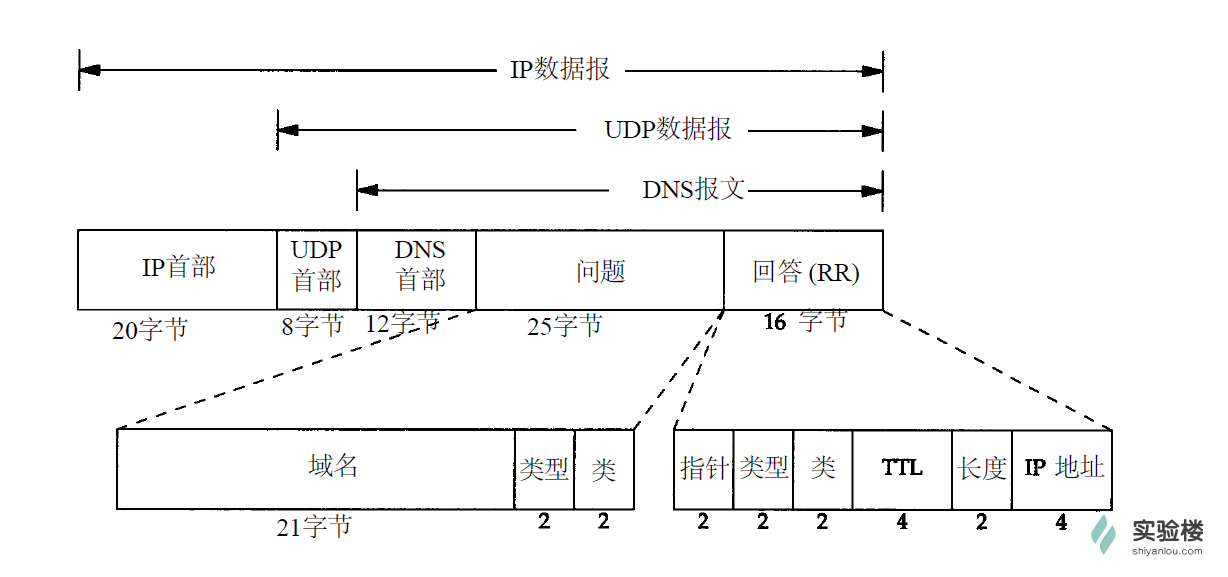
host mirrors.aliyuncs.com

下图示例查询www.shiyanlou.com的操作，由于实验楼网络限制，只有mirrors.aliyuncs.com域名可以查询。



**DNS 报文**

主机向 DNS 服务器发出的查询叫做**DNS 报文**，大致结构：



DNS 问答报文的内容，都是 IP 和域名的对应信息，关于 DNS 首部和内容 各字段这里不做详细介绍。深入了解，可以先 host 一个域名，再使用 tcpdump 抓取报文并解读。

**DNS 缓存和 hosts 文件**

刚才 DNS 解析查询过程的图中，共发出了 8 份 DNS 报文，这是非常消耗时间的，所以实际应用上使用 **DNS 缓存** ：当一个 DNS 服务器接收到一个 DNS 回答后，会将其信息缓存一段时间，当再有一个对相同域名的查询时，便可直接回复。

通过 DNS 缓存，其实很多查询都只需要本地 DNS 服务器便可完成。

有“翻墙”爱好的同学应该知道 hosts 文件，其实 hosts 文件可以看作是一个小型的 DNS 服务器。

使用命令打开 hosts 文件：

sudo gedit /etc/hosts

查看文件内容，可以发现里面全是类似下图中的 IP 和域名对应记录：



在实际上网过程中，域名解析的的优先顺序是：先在 DNS 缓存查询，若没有找到记录，再查询 hosts 文件，若还是没找到记录，再向 DNS 服务器发出 DNS 查询报文。

## 二、FTP

FTP (File Transfer Protocol 文件传输协议) 基于 TCP，使用端口号 20(数据)和 21(控制)。

它的主要功能是减少或消除在不同操作系统下处理文件的不兼容性，以达到便捷高效的文件传输效果。

FTP 只提供文件传输的基本服务，它采用 客户端—服务器 的方式，一个 FTP 服务器可同时为多个客户端提供服务。

在进行文件传输时，FTP 的客户端和服务器之间会建立两个 TCP 连接：21 号端口建立**控制连接**，20 号端口建立**数据连接**。

FTP 的传输有两种方式：ASCII 传输模式和二进制数据传输模式。

## 三、HTTP

HTTP (HyperText Transfer Protocol 超文本传输协议) 基于 TCP，使用端口号 80 或 8080。

每当你在浏览器里输入一个网址或点击一个链接时，浏览器就通过 HTTP 协议将网页信息从服务器提取再显示出来，这是现在使用频率最大的应用层协议。

这个原理很简单：

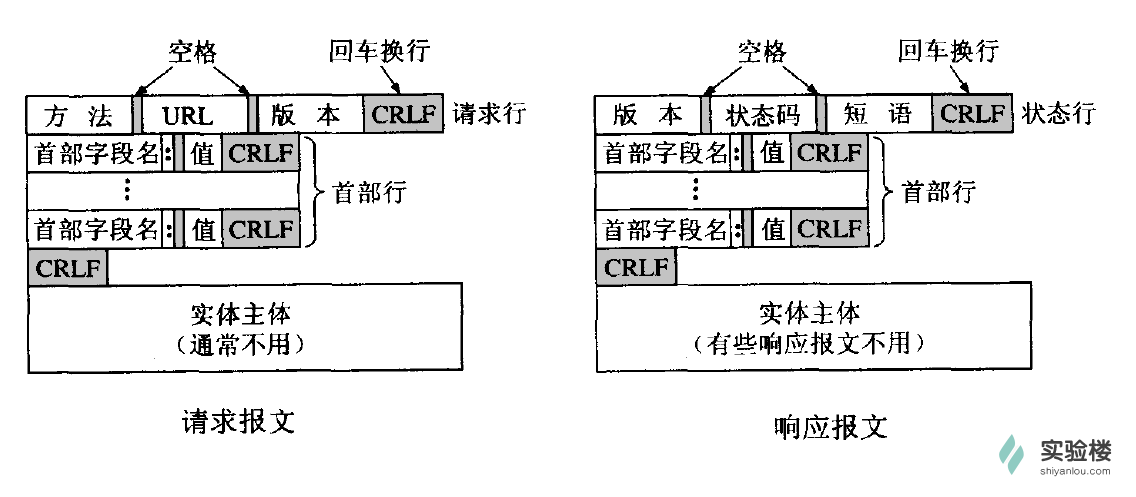
点击一个链接后，浏览器向服务器发起 TCP 连接；

连接建立后浏览器发送 HTTP 请求报文，然后服务器回复响应报文；

浏览器将收到的响应报文内容显示在网页上；

报文收发结束，关闭 TCP 连接。

HTTP 报文会被传输层封装为 TCP 报文段，然后再被 IP 层封装为 IP 数据报。HTTP 报文的结构：



可见报文分为 3 部分：

(1)开始行：用于区分是请求报文还是响应报文，请求报文中开始行叫做**请求行**，而响应报文中，开始行叫做**状态行**。在开始行的三个字段之间都用空格分开，结尾处 CRLF 表示回车和换行。

(2)首部行：用于说明浏览器、服务器或报文主体的一些信息。

(3)实体主体：请求报文中通常不用实体主体。

请求报文的**方法字段**是对所请求对象进行的操作，而响应报文的**状态码**是一个 3 位数字，分为 5 类 33 种：

**1xx** 表示通知信息，如收到或正在处理。

**2xx** 表示成功接收。

**3xx** 表示重定向。

**4xx** 表示客户的差错，如 404 表示网页未找到。

**5xx**表示服务器的差错，如常见的 502 Bad Gateway。

### 四、作业

1、使用 host 命令查看常见网站的 IP 地址，用 tcpdump 抓取 DNS 报文做解读，并截图。

参考：tcpdump 抓取命令：sudo tcpdump -vvv -X udp port 53

2、查看 linux 的 hosts 文件，并截图。

3、使用 tcpdump 抓取 HTTP 报文做解读，并截图。

参考：浏览器输入一个不存在的网址，抓取状态码为 404 报文。