

H1 第九周-web页面解析的流程学习

看完这个任务，就想起了去年学计算机网络的时候的那些东西，推荐大家去看《计算机网络》-第七版-谢希仁著

H2 域名解析的过程

H3 简单介绍域名服务器的类型

- 根域名服务器：最高层次的域名服务器，域名解析(IP地址转换)，都可能会借助根域名服务器
- 顶级域名服务器（TLD服务器）：负责管理在该顶级域名服务器注册的所有二级域名，当收到DNS查询时给出相应应答
- 权限域名服务器：负责一个区的服务器
- 本地域名服务器：当一台主机发送DNS查询请求时，这个查询请求报文就发送给本地域名服务器，我们的电脑里IPV4的DNS服务器就是指的本地域名服务器

H3 查询步骤

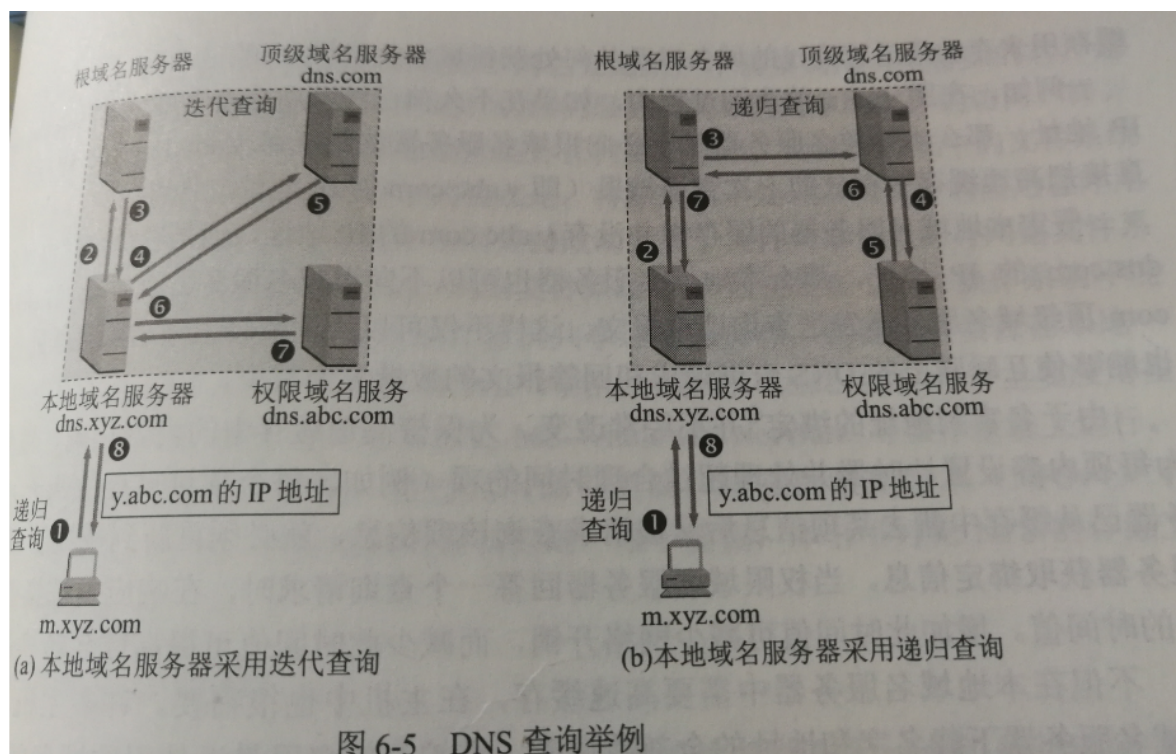


图 6-5 DNS 查询举例

这里直接借用书上的例子来说DNS解析的过程，域名解析过程有两种方式，一种是 **主机向本地域名服务器的查询-----递归查询**，另一种是 **本地域名服务器向根域名服务器的查询-----迭代查询**

可以看到这里本地域名服务器为：dns.xyz.com。当主机m.xyz.com想知道另一台域名为y.abc.com的IP地址时，那么

- 主机 m.xyz.com 先向本地域名服务器进行递归查询
- 本地域名服务器采用迭代查询，先向一个根域名服务器进行查询
- 然后根域名服务器告诉本地域名服务器，下一个应该查询的顶级域名服务器 dns.com 的IP地址
- 然后本地域名服务器就向这个顶级域名服务器查询
- 这个时候顶级域名服务器告诉本地服务器，下一个应该查询的权限域名服务器的IP地址
- 然后本地域名服务器就去查权限域名服务器

- 权限域名服务器就告诉本地域名服务器所要查询的主机的IP地址
- 最后本地域名服务器就告诉请求主机所要查询主机的IP地址

而递归查询就是右边那个图，就不再过多解释。不过要说的是，上面的递归查询总共8个步骤，总共要使用8个UDP报文。

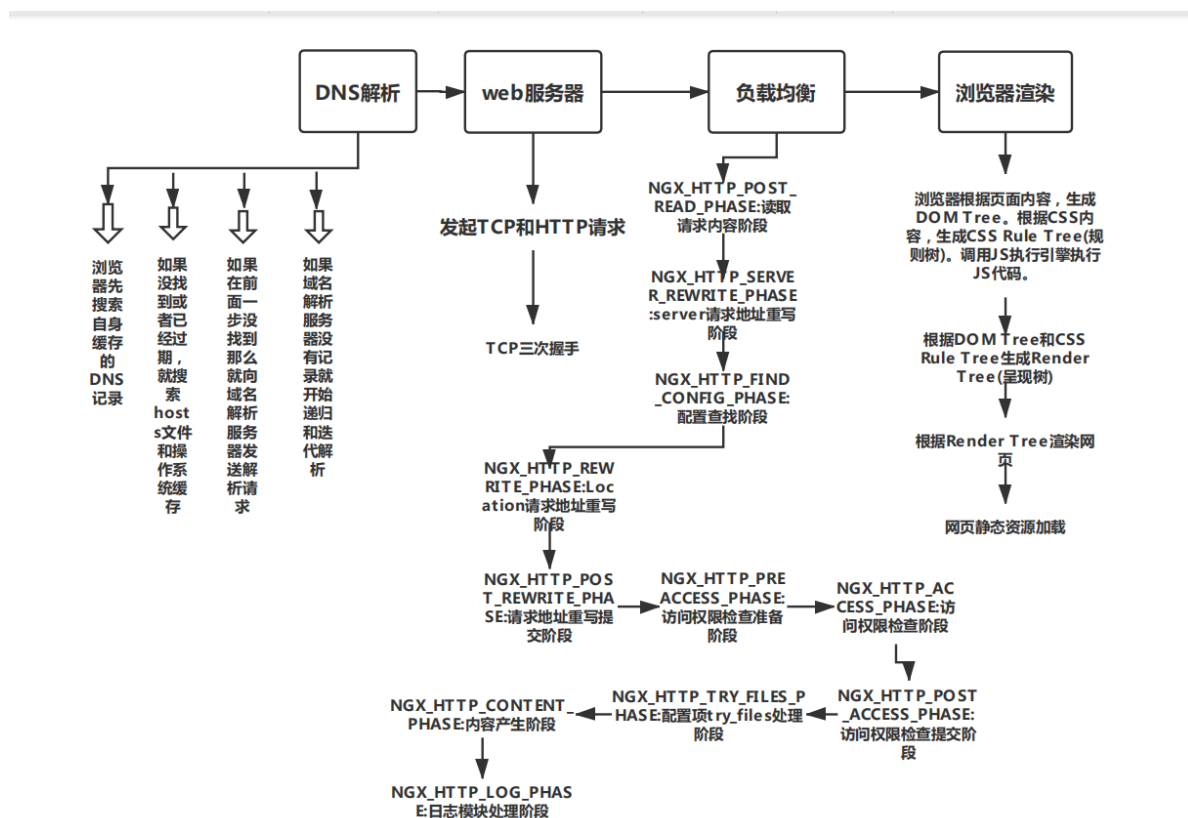
H3 域名解析的方式

域名解析记录只要分为A记录、MX记录、CNAME记录、NS记录、TXT记录

- A (Address) 记录：用来指定域名对应的IP地址，可以将多个域名地址解析到一个IP地址，但是不能讲一个域名解析到多个IP地址
- MX (Mail Exchange) 记录：讲某个域名下的邮件服务器指向自己的Mail Server
- CNAME (Canonical Name) 记录：别名解析就是可以为一个域名设置一个或多个别名
- NS记录：为某个域名指定DNS解析服务器
- TXT记录：为某个主机名或域名设置说明

这个如果有人买过域名肯定不陌生

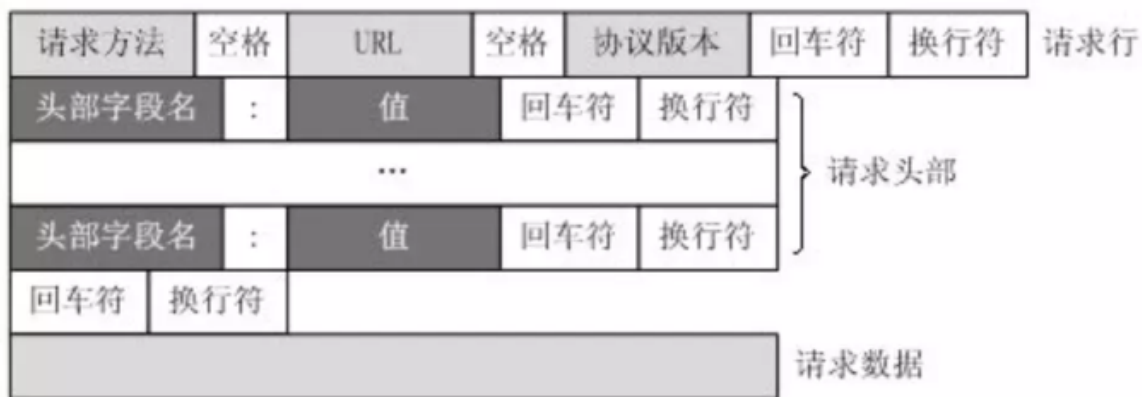
H2 web页面请求的流程（流程图、nginx处理的11个过程）



这是用processOn做的 如果流程这个地方有什么不对的请各位师傅指正

H2 http协议中的字段及其含义

H3 报文结构



请求行：声明请求方法、主机域名、资源路径、协议版本

请求头：声明客户端、服务器、报文的部分信息

请求体：存放发送的数据信息

重点放在请求头这里！

H3 HTTP首部字段

HTTP首部字段根据实际用途被分为以下4种类型：

- 通用首部字段

请求报文和响应报文两方都会使用的首部。

- 请求首部字段

从客户端向服务器端发送请求报文时使用的首部。补充了请求的附加内容、客户端信息、响应内容相关优先级等信息。

- 响应首部字段

从服务器端向客户端返回报文时使用的首部。补充了响应的附加内容，也会要求客户端附加额外的内容信息。

- 实体首部字段

针对请求报文和响应报文的实体部分使用的首部。补充了资源内容更新时间等与实体有关的信息。

HTTP/1.1规范定义了如下47种首部字段

H4 通用首部字段（9个）

Cache-Control: 控制缓存的行为
Connection: 逐跳首部、连接的管理
Date: 创建报文的日期时间
Program: 报文指令
Trailer: 报文末端的首部一览
Transfer-Encoding: 指定报文主体的传输编码方式
Upgrade: 升级为其他协议
Via: 代理服务器的相关信息
Warning: 错误通知

H4 请求首部字段 (19个)

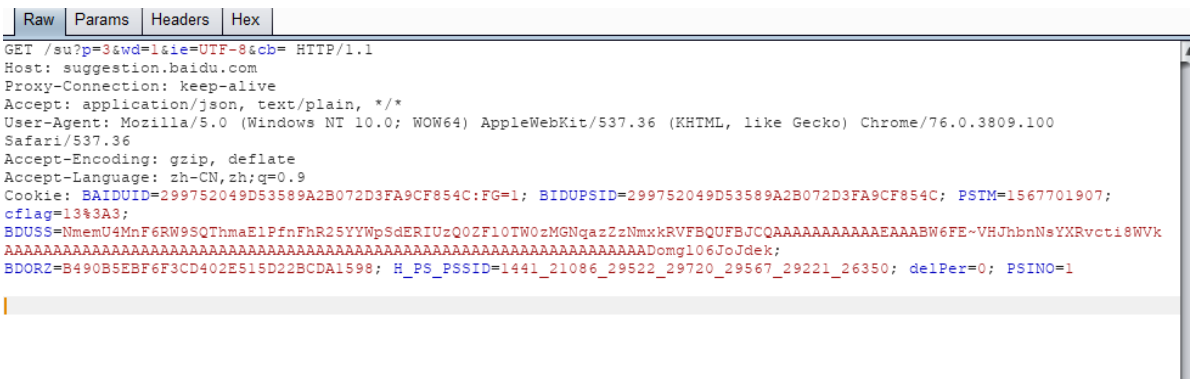
Accept: 用户代理可处理的媒体类型
Accept-Charset: 优先的字符集
Accept-Encoding: 优先的内容编码
Accept-Language: 优先的语言 (自然语言)
Authorization: Web认证信息
Expect: 期待服务器的特定行为
From: 用户的电子邮箱地址
Host: 请求资源所在的服务器
If-Match: 比较实体标记 (ETag)
If-Modified-Since: 比较资源的更新时间
If-None-Match: 比较实体标记 (与If-Match相反)
If-Range: 资源未更新时发送实体Byte的范围请求
If-Unmodified-Since: 比较资源的更新时间 (与If-Modified-Since相反)
Max-Forwards: 最大传输逐跳数
Proxy-Authorization: 代理服务器要求客户端的认证信息
Range: 实体的字节范围请求
Referer: 对请求中的URI的原始获取方
TE: 传输编码的优先级
User-Agent: HTTP客户端程序的信息

H4 响应首部字段 (9个)

Accept-Ranges: 是否接受字节范围请求
Age: 推算资源创建经过时间
Content-Disposition: 可以控制返回的资源是下载还是预览 (图片)
ETag: 资源的匹配信息
Location: 令客户端重定向至指定URI
Proxy-Authenticate: 代理服务器对客户端的认证信息
Retry-After: 对再次发起请求的时机要求
Server: HTTP服务器的安装信息
Vary: 代理服务器缓存的管理信息
WWW-Authenticate: 服务器对客户端的认证信息

H4 实体首部字段（10个）

Allow: 资源可支持的HTTP方法
Content-Encoding: 实体主体试用的编码方式
Content-Language: 实体主体的自然语言
Content-Length: 实体主体的大小（单位：字节）
Content-Location: 替换对应资源的URI
Content-MD5: 实体主体的报文摘要
Content-Range: 实体主体的位置范围
Content-Type: 实体主体的媒体类型
Expires: 实体主体过期的日期时间
Last-Modified: 资源的最后修改日期时间



这是我用bp随便抓的一个包

H2 HTTP请求方式以及返回状态码的类型和含义

根据HTTP的标签，在HTTP1.0版本中，有三种请求方法，分别是POST、GET和HEAD方法；在HTTP1.1版本中，新增了六种请求方法，分别是OPTIONS、PUT、PATCH、DELETE、TRACE和CONNECT方法。

H3 请求方式

H4 0x01-GET请求

get方式是最常见的一种请求方式之一了，它是请求读取由URL所标志的信息

H4 0x02-POST

post方式是另外一种最常见的方式之一，向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。

H4 0x03-OPTION

返回服务器针对特定资源所支持的HTTP请求方法，也可以利用向web服务器发送*的请求来测试服务器的功能性，请求一些选项的信息

H4 0x04-HEAD

请求读取由URL所标志的信息的首部

H4 0x05-PUT

在指明的URL下存储一个文档

H4 0x06-DELETE

请求服务器删除Request-URL所标识的资源

H4 0x07-TRACE

用来进行环回测试的请求报文

H4 0x08-CONNECT

HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。

H3 状态码

状态码	说明
200	请求成功并返回数据
201	创建资源成功并返回组员数据
202	接受请求，但无法立即完成创建行为，返回的实体中应该包含当前状态的信息，以及指向处理状态监视器或状态预测的指针，以便客户端能够获取最新状态
204	请求成功不返回数据，通常用于 delete
301	被请求的资源已永久移动到新位置
302	请求的资源现在临时从不同的 URI 响应请求
303	对应当前请求的响应可以在另一个 URI 上被找到，客户端应该使用 GET 方法进行请求。比如在创建已经被创建的资源时，可以返回 303；
400	请求体包含语法错误
401	需要验证用户身份或认证失败
403	服务器拒绝执行
404	找不到目标资源
500	服务器遇到了一个未曾预料的状态，导致了它无法完成对请求的处理
501	服务器不支持当前请求所需要的某个功能
502	作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应
503	服务器维护或者过载，服务器当前无法处理请求