访问一个网页的全过程

2017年12月08日 18:03:35 toumingren527 阅读数：22835

引言

打开浏览器，在地址栏输入URL，回车，出现网页内容。整个过程发生了什么？其中的原理是什么？以下进行整理和总结。

整个过程可以概括为几下几个部分：

1. 域名解析成IP地址；
2. 与目的主机进行TCP连接（三次握手）；
3. 发送与收取数据（浏览器与目的主机开始HTTP访问过程）；
4. 与目的主机断开TCP连接（四次挥手）；

正文

下面详细介绍其中的原理：

1. 域名解析成IP地址

访问目标地址有两种方式：

①使用目标IP地址访问。由于IP地址是一堆数字不方便记忆，于是有了域名这种字符型标识。

②使用域名访问。域名解析就是域名到IP地址的转换过程，域名的解析工作由DNS服务器完成。

DNS域名解析时用的是UDP协议。整个域名解析的过程如下：

1. 浏览器向本机DNS模块发出DNS请求，DNS模块生成相关的DNS报文；
2. DNS模块将生成的DNS报文传递给传输层的UDP协议单元；
3. UDP协议单元将该数据封装成UDP数据报，传递给网络层的IP协议单元；
4. IP协议单元将该数据封装成IP数据包，其目的IP地址为DNS服务器的IP地址；
5. 封装好的IP数据包将传递给数据链路层的协议单元进行发送；
6. 发送时在ARP缓存中查询相关数据，如果没有，就发送ARP广播（包含待查询的IP地址，收到广播的主机检查自己的IP，符合条件的主机将含有自的ARP包发送给ARP广播的主机）请求，等待ARP回应；
7. 得到ARP回应后，将IP地址与路由的下一跳MAC地址对应的信息写入ARP缓存表；
8. 写入缓存后，以路由下一跳的地址填充目的MAC地址，以数据帧形式转发；
9. 转发可能进行多次；
10. DNS请求到达DNS服务器的数据链路层协议单元；
11. DNS服务器的数据链路层协议单元解析数据帧，将内部的IP数据包传递给网络层IP协议单元；
12. DNS服务器的IP协议单元解析IP数据包，将内部的UDP数据报传递给传输层UDP协议单元；
13. DNS服务器的UDP协议单元解析收到的UDP数据报，将内部的DNS报文传递给DNS服务单元；
14. DNS服务单元将域名解析成对应IP地址，产生DNS回应报文；
15. DNS回应报文->UDP->IP->MAC->我的主机；
16. 我的主机收到数据帧，将数据帧->IP->UDP->浏览器；
17. 将域名解析结果以域名和IP地址对应的形式写入DNS缓存表。

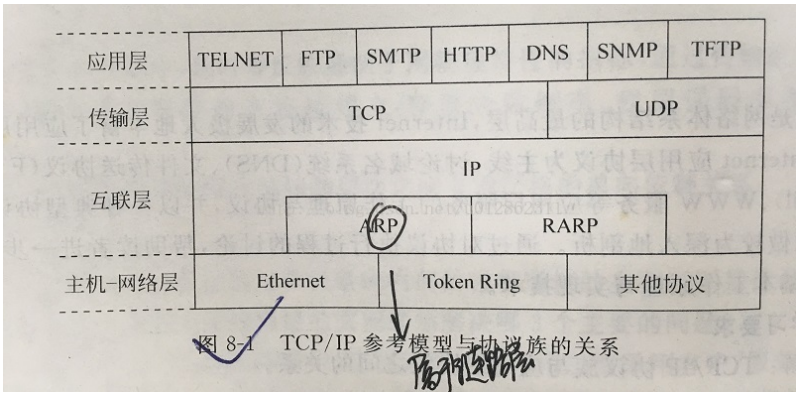
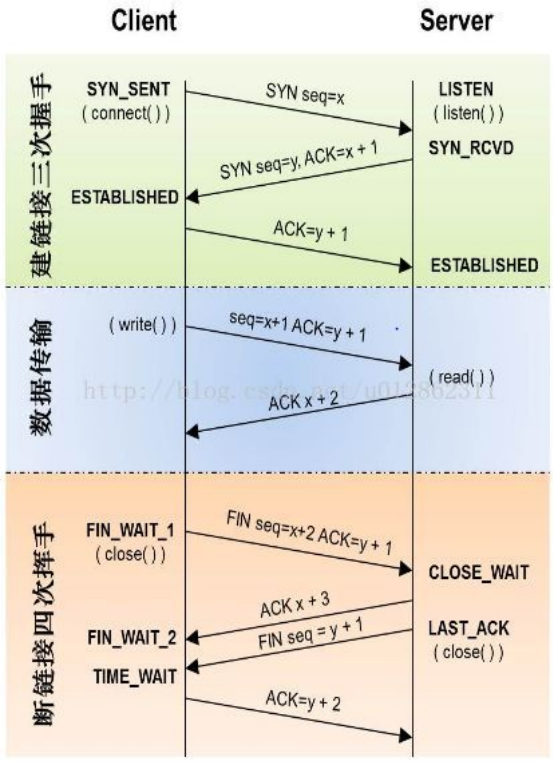


图 8-1 TCP/IP 参考模型与协议族的关系

与目的主机进行TCP连接（三次握手）

向目的主机发送TCP连接请求报文；

1. 该TCP报文中SYN标志位设为1，表示连接请求；
2. 该TCP报文通过IP（DNS）->MAC（ARP）->网关->目的主机；
3. 目的主机收到数据帧，通过IP->TCP，TCP协议单元回应请求应答报文；
4. 该报文中SYN和ACK标志设为1，表示连接请求应答；
5. 该TCP报文通过IP（DNS）->MAC（ARP）->网关->我的主机；
6. 我的主机收到数据帧，通过IP->TCP，TCP协议单元回应请求确认报文；
7. 该TCP报文通过IP（DNS）->MAC（ARP）->网关->目的主机；
8. 目的主机收到数据帧，通过IP->TCP，连接建立完成。



发送与收取数据（浏览器与目的主机开始HTTP访问过程）

只有建立连接后才能开始传输数据。

1. 浏览器向域名发出GET方法报文（HTTP请求）；
2. 该GET方法报文通过TCP->IP（DNS）->MAC（ARP）->网关->目的主机；
3. 目的主机收到数据帧，通过IP->TCP->HTTP，HTTP协议单元会回应HTTP协议格式封装好的HTML形式数据（HTTP响应）；[从请求信息中获得
问的主机名。从请求信息中获取客户机想要访问的web应用（web应用程序指提供浏览器访问的程序，简称web应用）。从请求信息中获取客户机

4. 该HTML数据通过TCP->IP（DNS）->MAC（ARP）->网关->我的主机；
5. 我的主机收到数据帧，通过IP->TCP->HTTP->浏览器，浏览器以网页形式显示HTML内容。

HTTP协议

HTTP请求：http请求由三部分组成，分别是：请求行、消息报头、请求正文

请求行以一个方法符号开头，以空格分开，后面跟着请求的URI和协议的版本，格式如下：Method Request-URI HTTP-Version CRLF
其中 Method表示请求方法；Request-URI是一个统一资源标识符；HTTP-Version表示请求的HTTP协议版本；CRLF表示回车和换行（除了作为结尾外，不允许出现单独的CR或LF字符）。

请求方法（所有方法全为大写）有多种，各个方法的解释如下：

- GET　请求获取Request-URI所标识的资源
- POST　在Request-URI所标识的资源后附加新的数据
- HEAD　请求获取由Request-URI所标识的资源的响应消息报头
- PUT　请求服务器存储一个资源，并用Request-URI作为其标识
- DELETE 请求服务器删除Request-URI所标识的资源
- TRACE 请求服务器回送收到的请求信息，主要用于测试或诊断
- CONNECT 保留将来使用
- OPTIONS 请求查询服务器的性能，或者查询与资源相关的选项和需求

HTTP响应也是由三个部分组成，分别是：状态行、消息报头、响应正文
状态行格式如下：HTTP-Version Status-Code Reason-Phrase CRLF
其中，HTTP-Version表示服务器HTTP协议的版本；Status-Code表示服务器发回的响应状态代码；Reason-Phrase表示状态代码的文本描述。
状态代码有三位数字组成，第一个数字定义了响应的类别，且有五种可能取值：

- 1xx：指示信息--表示请求已接收，继续处理
- 2xx：成功--表示请求已被成功接收、理解、接受
- 3xx：重定向--要完成请求必须进行更进一步的操作
- 4xx：客户端错误--请求有语法错误或请求无法实现
- 5xx：服务器端错误--服务器未能实现合法的请求

常见状态代码、状态描述、说明：

- 200 OK　//客户端请求成功
- 400 Bad Request //客户端请求有语法错误，不能被服务器所理解
- 401 Unauthorized //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用
- 403 Forbidden //服务器收到请求，但是拒绝提供服务
- 404 Not Found //请求资源不存在，eg：输入了错误的URL
- 500 Internal Server Error //服务器发生不可预期的错误
- 503 Server Unavailable //服务器当前不能处理客户端的请求，一段时间后可能恢复正常

eg：HTTP/1.1 200 OK （CRLF）
消息报头：

常用的请求报头

Accept

Accept请求报头域用于指定客户端接受哪些类型的信息。eg：Accept：image/gif，表明客户端希望接受GIF图象格式的资源；Accept：text/html，表明客户端希望接受html文本。

Accept-Charset

Accept-Charset请求报头域用于指定客户端接受的字符集。eg：Accept-Charset:iso-8859-1,gb2312.如果在请求消息中没有设置这个域，缺省是任何字符集都可以接受。

Accept-Encoding

Accept-Encoding请求报头域类似于Accept，但是它是用于指定可接受的内容编码。eg：Accept-Encoding:gzip.deflate.如果请求消息中没有设置这个域服务器假定客户端对各种内容编码都可以接受。

Accept-Language

Accept-Language请求报头域类似于Accept，但是它是用于指定一种自然语言。eg：Accept-Language:zh-cn.如果请求消息中没有设置这个报头域

Authorization

Authorization请求报头域主要用于证明客户端有权查看某个资源。当浏览器访问一个页面时，如果收到服务器的响应代码为401（未授权），可以发送
Authorization请求报头域的请求，要求服务器对其进行验证。

Host（发送请求时，该报头域是必需的）

Host请求报头域主要用于指定被请求资源的Internet主机和端口号，它通常从HTTP URL中提取出来的，eg：
我们在浏览器中输入：http://www.guet.edu.cn/index.html
浏览器发送的请求消息中，就会包含Host请求报头域，如下：
Host：www.guet.edu.cn
此处使用缺省端口号80，若指定了端口号，则变成：Host：www.guet.edu.cn:指定端口号

User-Agent

User-Agent请求报头域允许客户端将它的操作系统、浏览器和其它属性告诉服务器。这个报头域不是必需的。

常用的响应报头

Location

Location响应报头域用于重定向接受者到一个新的位置。Location响应报头域常用在更换域名的时候。

Server

Server响应报头域包含了服务器用来处理请求的软件信息。与User-Agent请求报头域是相对应的。下面是
Server响应报头域的一个例子：
Server：Apache-Coyote/1.1

WWW-Authenticate

WWW-Authenticate响应报头域必须被包含在401（未授权的）响应消息中，客户端收到401响应消息时候，并发送Authorization报头域请求服务器
时，服务端响应报头就包含该报头域。
eg：WWW-Authenticate:Basic realm="Basic Auth Test!" //可以看出服务器对请求资源采用的是基本验证机制。

HTTP协议详解，可阅读：http://www.cnblogs.com/li0803/archive/2008/11/03/1324746.html

与目的主机断开TCP连接（四次挥手）

TCP连接释放过程：

1. 浏览器向目的主机发出TCP连接结束请求报文，此时进入FIN WAIT状态；
2. 该报文FIN标志位设为1，表示结束请求；
3. TCP结束请求报文通过IP（DNS）->MAC（ARP）->网关->目的主机；
4. 目的主机收到数据帧，通过IP->TCP，TCP协议单元回应结束应答报文；
5. 当前只是进行回应，因为目的主机可能还有数据要传，并不急着断开连接；
6. 该报文中ACK标志位设为1，表示收到结束请求；
7. 目的数据发送完所有数据后，向我的主机发出TCP连接结束请求报文；
8. 该报文FIN标志位设为1，表示结束请求；
9. TCP结束请求报文通过IP（DNS）->MAC（ARP）->网关->我的主机；
10. 我的主机收到数据帧，通过IP->TCP，TCP协议单元回应结束应答报文，此时进入TIME WAIT状态，因为不相信网络是可靠的，如果目的主机没
    发；
11. 该报文中的FIN标志位均设为1，表示结束应答；
12. 该TCP回应报文通过IP（DNS）->MAC（ARP）->网关->目的主机；
13. 目的主机关闭连接；
14. TIME WAIT等待结束后，没有收到回复，说明目的正常关闭了，我的主机也关闭连接。

总结：

URL访问网站时的网络传输全过程，可以归纳为：

首先通过域名找到IP，如果缓存里没有就要请求DNS服务器；得到IP后开始与目的主机进行三次握手来建立TCP连接；连接建立后进行HTTP访问，传
内容；传输完后与目的主机四次挥手来断开TCP连接。