实验二 万维网运行原理分析

2.1 实验介绍

本实验通过对特定网站分析，了解网站运行原理和相关技术；通过使用抓包工具采集HTTP 协议包并进行分析。

2.2 实验目标

深入了解万维网结构、原理、技术

深入了解并掌握 WEB 页面组成

深入了解并掌握 HTTP 协议

2.3 实验原理与方法

万维网（www）运行原理

当输入一个网址进行访问，这中间其实是你的客户端浏览器与服务器端的通信过程,具体如下：

浏览器与网络上的域名对应的 Web 服务器建立 TCP 连接浏览器发出要求访问某个页面的 HTTP 请求，Web 服务器在接收到 HTTP 请求后，解析 HTTP 请求，然后发回包含目标页面的文件数据的 HTTP 响应浏览器接受到 HTTP 响应后，解析 HTTP 响应，并在其窗口中展示网页文件内容，浏览器与 Web 服务器之间的 TCP 连接关闭

服务器

接受来自浏览器的 TCP 的请求接收并解析 HTTP 请求创建并发送 HTTP 响应。常用的 Web 服务器有 IIS，Tomcat，Weblogic，jboss 等。

浏览器

请求与 Web 服务器建立 TCP 连接创建并发送 HTTP 请求接受并解析 HTTP 响应展示 html 文档

HTTP 客户程序(浏览器)和 HTTP 服务器分别由不同的软件开发商提供,目前最流行的浏览器 IE，Firefox，Google Chrome，Apple Safari 等。

HTTP 协议

简介

HTTP 协议（Hyper Text Transfer Protocol，超文本传输协议),是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。HTTP 基于 TCP/IP 通信协议来传递数据。HTTP 基于客户端/服务端（C/S）架构模型，通过一个可靠的链接来交换信息，是一个无状态的请求/响应协议。

特点

（1）HTTP 是无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。

（2）HTTP 是媒体独立的：只要客户端和服务器知道如何处理的数据内容，任何类型的数据都可以通过 HTTP 发送。客户端以及服务器指定使用适合的 MIME-type 内容类型。

（3）HTTP 是无状态：无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

HTTP 请求报文

请求行：

①是请求方法，GET 和 POST 是最常见的 HTTP 方法，除此以外还包括 DELETE、HEAD、OPTIONS、PUT、TRACE。

②为请求对应的 URL 地址，它和报文头的 Host 属性组成完整的请求 URL。

③是协议名称及版本号。

请求头：

④是 HTTP 的报文头，报文头包含若干个属性，格式为“属性名:属性值”，服务端据此获取客户端的信息。与缓存相关的规则信息，均包含在 header 中

请求体：

⑤是报文体，它将一个页面表单中的组件值通过 param1=value1&param2=value2 的键值对形式编码成一个格式化串，它承载多个请求参数的数据。不但报文体可以传递请求参数，请求 URL 也可以通过类似于“/chapter15/user.html?param1=value1&param2=value2”的方式传递请求参数。

HTTP 请求报文头属性

Accept

请求报文可通过一个“Accept”报文头属性告诉服务端 客户端接受什么类型的响应。Accept 属性的值可以为一个或多个 MIME 类型的值（描述消息内容类型的因特网标准， 消息能包含文本、图像、音频、视频以及其他应用程序专用的数据）

Cookie

客户端的 Cookie 就是通过这个报文头属性传给服务端的。服务端是如何？将 HTTP 请求报文头的 Cookie 属性的“jsessionid”的值关联起来，这样就知道客户端的多个请求是隶属于哪一个 Session。

Referer

表示这个请求是从哪个 URL 过来的。

Cache-Control

对缓存进行控制，例如一个请求希望响应返回的内容在客户端要被缓存一年或不希望被缓存，就可以通过这个报文头达到目的。

HTTP 响应报文

响应行：

①报文协议及版本；

②状态码及状态描述；

响应头：

③响应报文头，也是由多个属性组成；

响应体：

④响应报文体

响应状态码

和请求报文相比，响应报文多了一个“响应状态码”，它以“清晰明确”的语言告诉客户端本次请求的处理结果。

HTTP 的响应状态码由 5 段组成：

1xx	告诉客户端，请求已收到
2xx	处理成功
3xx	重定向，让客户端再发起一个请求
4xx	处理错误，一般是客户端异常
5xx	处理错误，一般是服务器异常

以下是几个常见的状态码：

200 OK	请求成功
303 See Other	查看其他地址
304 Not Modified	资源未修改，可读取本地缓存
403 Forbidden	服务器拒绝执行客户端请求
404 Not Found	服务器无法找到客户端的请求
500 Internal Server Error	服务器内部错误，无法完成请求
502 Bad Gateway	网关或者代理的服务器执行请求时，从远程服务器收到无效的响应
503 Service Unavailable	由于超载或者维护，服务器无法处理请求
504 Gateway Time-out	网关或者代理服务器未能及时从远程服务器获取请求

常见的 HTTP 响应报文头属性

Cache-Control

响应输出到客户端后，服务端通过该报文头属告诉客户端如何控制响应内容的缓存。常见的取值有 private、public、no-cache、max-age，no-store，默认为 private，缓存时间为 31536000 秒（365 天）也就是说，在 365 天内再次请求这条数据，都会直接获取缓存数据库中的数据，直接使用。

Private	客户端可以缓存
Public	客户端和代理服务器都可缓
Max-age=xxx	缓存的内容将在 xxx 秒后失效
No-cache	需要使用对比缓存来验证缓存数据
No-store	所有内容都不缓存

ETag

一个代表响应服务端资源（如页面）版本的报文头属性，如果某个服务端资源发生变化了，这个 ETag 就会相应发生变化。它是 Cache-Control 的有益补充，可以让客户端“更智能”地处理什么时候要从服务端取资源，什么时候可以直接从缓存中返回响应。

Location

我们在 JSP 中让页面 Redirect 到一个某个 A 页面中，其实是让客户端再发一个请求到 A 页面，这个需要 Redirect 到的 A 页面的 URL，其实就是通过响应报文头的 Location 属性告知客户端的。

Set-Cookie

服务端可以设置客户端的 Cookie，其原理就是通过这个响应报文头属性实现的：

客户端请求服务器，如果服务器需要记录该用户状态，就使用 response 向客户端浏览器颁发一个 Cookie。客户端浏览器会把 Cookie 保存起来。当浏览器再请求该网站时，浏览器把请求的网址连同该 Cookie 一同提交给服务器。服务器检查该 Cookie，以此来辨认用户状态。服务器还可以根据需要修改 Cookie 的内容。

Cookie 的 maxAge 决定着 Cookie 的有效期，单位为秒（Second）。Cookie 中通过 getMaxAge() 方法与 setMaxAge(int maxAge)方法来读写 maxAge 属性。如果 maxAge 属性为正数，则表示该 Cookie 会在 maxAge 秒之后自动失效。如果 maxAge 为负数，则表示该 Cookie 仅在本浏览器窗口以及本窗口打开的子窗口内有效，关闭窗口后该 Cookie 即失效。如果 maxAge 为 0，则表示删除该 Cookie。

Cookie 并不提供修改、删除操作。如果要修改某个 Cookie，只需要新建一个同名的 Cookie，添加到 response 中覆盖原来的 Cookie。

如果要删除某个 Cookie，只需要新建一个同名的 Cookie，并将 maxAge 设置为 0，并添加到 response 中覆盖原来的 Cookie。

2.4 实验步骤

1、选择一个网站；

2、登陆该网站，多次操作，请求页面，使用浏览器的开发者工具，查看 http 协议内容

2.5 实验要求

1、绘制分析网络拓扑和数据流向，按照浏览器、协议、服务器、数据等结构；

2、分析所选单个网页组成，按 HTML 组成要素；

3、分析所选网页的 HTTP 协议，包括请求报文、应答报文及报文关键点分析；

4、撰写实验报告。