中国科学技术大学计算机学院

《计算机网络》实验报告

实验题目：lab2_WireShark Labs(Getting Started & HTTP)

学生姓名：胡毅翔

学生学号：PB18000290

专业：计算机科学与技术

指导老师：张信明

完成日期：2020 年 11 月 8 日

计算机实验教学中心制

2019 年 09 月

实验目的

1.掌握 Wireshark 网络分析工具。

2.捕获观察并分析 HTTP 报文结构。

3.回答本次实验指导中的问题。

4.分析 HTTP 中 GET 和 POST 请求方式的区别。

实验原理

本次实验使用 WireShark 工具。其中，用来观察执行协议实体之间交换的报文的基本工具称为分组嗅探器(packet sniffer)。分组嗅探器被动地拷贝(嗅探)由计算机发送和接收的报文；它也能显示出这些被捕获报文的各个协议字段的内容。分组嗅探器从不发送报文，同时接收到的报文也不会显式地发送到分组嗅探器。它接受的是发送/接收的报文的复制。

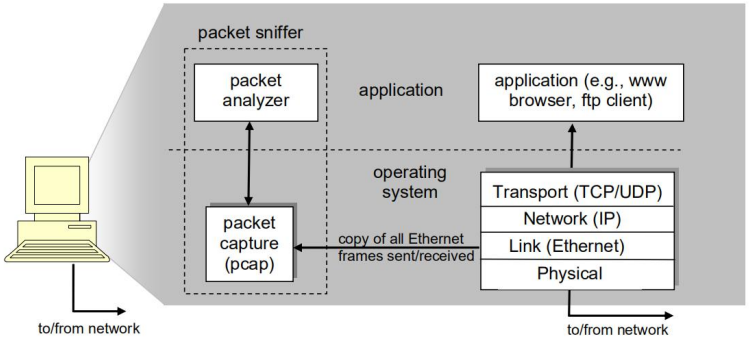

图 1

分组嗅探器的结构如图 1 所示。在图 1 的右边是运行在计算机上的协议和应用。分组嗅探器(图中画虚线框部分)是计算机中的附加软件(区别于上述协议和应用)，它包含两个部分。分组捕获库获取每一个链路层接收/发送的帧。第二部分是分组分析器，其中显示了协议所有字段的内容。为了实现这一目的，分组分析器必须理解所有协议所交换的信息的结构。比如，我们对图 1 中 HTTP 协议的各个字段信息感兴趣。分组分析器理解以太网帧的格式，所以可以从以太网帧中区分出 IP 数据报。同时，它还理解 IP 数据报格式，所以它能从 IP 数据报分离出 TCP 报文段。最后，它还理解 TCP 报文段格式，从中分离出 HTTP 报文。又因它理解 HTTP 协议，所以能在实现 WireShark 中显示 HTTP 协议各字段信息的功能。

实验环境

1.PC 一台

2.Windows 系统

3.WireShark 网络分析工具(版本 3.2.7)

4.Edge 浏览器(版本 86.0.622.56)

实验过程

WireShark Lab: Getting Started

WireShark 的安装

实验步骤

1.前往 http://www.wireshark.org/download.html 下载并安装 WireShark。

2.下载 WireShark 用户指南。

尝试运行 WireShark

实验步骤

1. 启动 WireShark，结合实验指导书，了解 WireShark 各个界面(命令菜单，显示过滤器，封包列表，封包详细信息以及 16 进制数据)的功能。

2. 根据实验指导书，对 WireShark 工具进行一系列设置。

3. 开始运行 WireShark，打开链接 http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html。

4. 加载完成后，在捕获窗口中停止 WireShark 捕获分组，利用“http”规则筛选出 HTTP 条目。

5. 分析 HTTP 报文结构。

请求报文

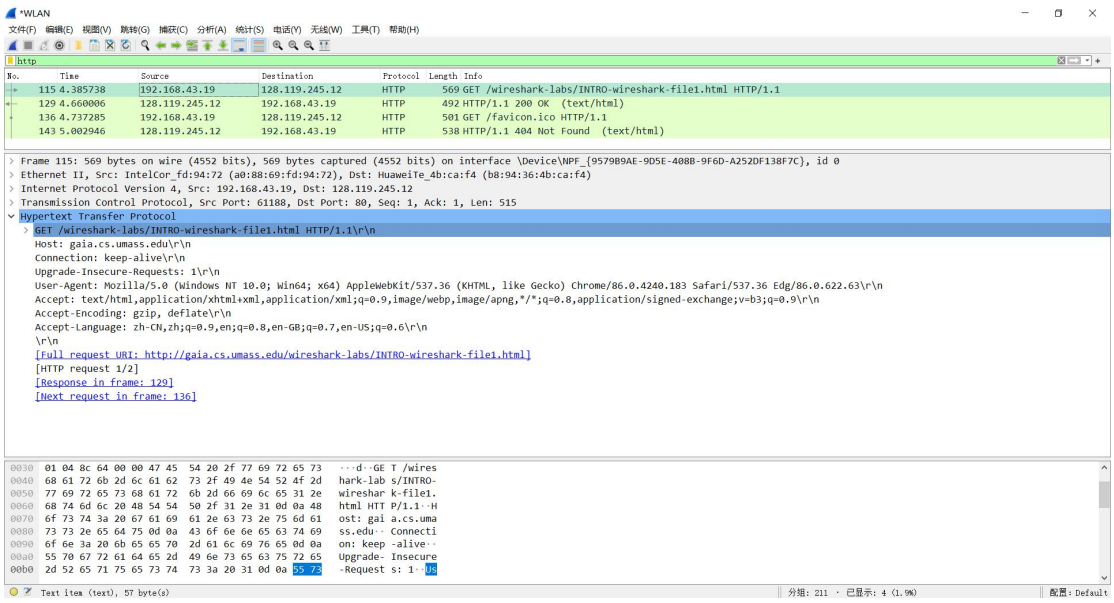

图 2

第一部分为请求行，说明请求类型(方法字段)为 GET，要访问的资源(URL 字段)为 /wireshark-labs/INTRO-wireshark-file1.html，所用的 HTTP 版本(HTTP 版本字段)为 1.1 版本(即采取持续连接，服务器在发送相应后保持该 TCP 连接打开，相同客户和服务器之间后续的请求和相应能通过相同的连接进行传送)。

第二部分为请求头部，或称首部行。Host 指明了对象所在的主机(gaia.cs.umass.edu)。Connection: keep-alive 表示使用持续连接。Upgrade-Insecure-Requests: 1 告诉服务器：浏览器可以处理 https 协议。User-Agent 用来指明用户代理，即向服务器发送请求的浏览器类型。这里浏览器类型是 Edg/86.0.622.63。Accpet 表明浏览器希望接收的数据类型。Accpet-Encoding 表明浏览器希望接收的数据编码格式。Accpet-Language 表明浏览器希望接收的对象的语言版本。这里 zh-CN 表示希望接收简体中文版本。

第三部分为实体体。使用 GET 方法时实体体为空。

响应报文

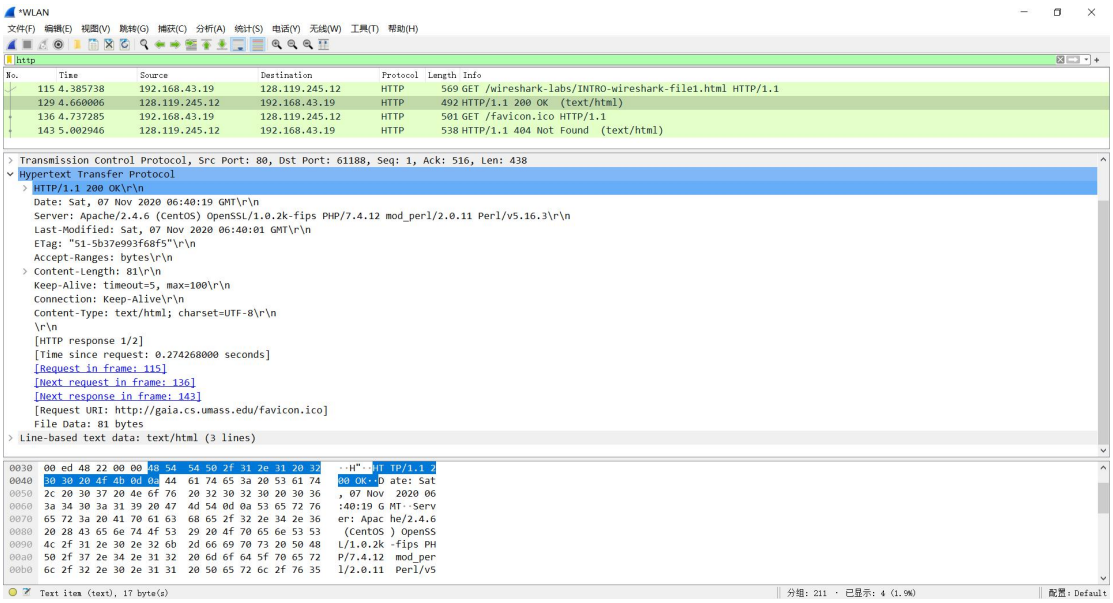

图 3

第一部分为状态行，有 3 个字段：协议版本字段、状态码和相应状态信息。这里指示服务器正在使用 HTTP/1.1，并且一切正常。

第二部分为首部行。Date: 指示服务器产生并发送该响应报文的日期和时间。Server: 指示该报文是由 Apache Web 服务器产生的。Last-Modified: 指示了对象创建或最后修改的日期和时间。Etag: 指示了对象的标记，可以视为版本标记，主要是为了解决一些 Last-Modified 无法解决的问题(1.一些文件也许会周期性的更改，但是他的内容并不改变(仅仅改变的修改时间)，这个时候我们并不希望客户端认为这个文件被修改了，而重新 GET; 2.某些文件修改非常频繁，比如在秒以下的时间内进行修改，(比方说 1s 内修改了 N 次)，If-Modified-Since 能检查到的粒度是 s 级的，这种修改无法判断(或者说 UNIX 记录 MTIME 只能精确到秒) ;3.某些服务器不能精确的得到文件的最后修改时间；)。响应头 Accept-Ranges 标识自身支持范围请求(partial requests)。字段的具体值用于定义范围请求的单位。Content-Length: 指示了被发送对象中的字节数。Keep-Alive: 中 timeout 指示了一个空闲连接需要保持打开状态的最小时长（以秒为单位），而 max 指示了在连接关闭之前，在此连接可以发送的请求的最大值。Connection: Keep-Alive 指示了 持续连接。 Content-Type: 指示了实体体的对象是 html，编码格式为 utf-8。

6.退出 WireShark

问题

Q1:List up to 10 different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above. TCP、DNS TLSv1.2 ICMP NBNS HTTP UDP ARP MDNS BROWSER TLSv1.3 2. How long did it take from

when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

A1:发送请求报文的相对时间是 4.385738 秒，接收响应报文的相对时间是 4.660006，故这之间的用时是 0.274268 秒

Q2:What is the Internet address of the gaia.cs.umass.edu (also known as www net.cs.umass.edu)? What is the Internet address of your computer?

A2:gaia.cs.umass.edu 的因特网地址是 128.119.245.12。我的计算机的因特网地址是 192.128.43.19。

Q3:Print the two HTTP messages displayed in step 9 above. To do so, select Print from the Wireshark File command menu, and select “Selected Packet Only” and “Print as displayed” and then click OK.

A3:HTTP 请求和接收报文截图已包含在实验步骤中。

WireShark Lab: HTTP

The Basic HTTP GET/response interaction

实验步骤

1.启动浏览器。

2.启动 WireShark 分组嗅探器，在显示过滤器中输入 http。

3.等一分多钟，然后开始捕获分组。在浏览器中输入 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

4.停止 WireShark 捕获分组。

问题

Q1:Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

A1:我的浏览器运行的 HTTP 版本是 HTTP/1.1，从请求行”GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n” 可以看出。服务器的 HTTP 版本是 HTTP/1.1，从响应行”HTTP/1.1 200 OK\r\n”可以看出。

Q2:What languages (if any) does your browser indicate that it can accept to the server?

A2:根据首部行”Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n”，我的浏览器指示希望接收的语言版本有简体中文、中文、英语、英式英语和美式英语，且根据 q 值的不同表明的对不同语言版本的偏好程度(q 值越大，表示越希望收到该语言版本)。

Q3:What is the IP address of your computer? Of the gaia.cs.umass.edu server?

A3:gaia.cs.umass.edu 服务器的 IP 地址是 128.119.245.12。我的计算机的 IP 地址是 192.128.43.19。

Q4:What is the status code returned from the server to your browser?

A4:根据响应行”HTTP/1.1 200 OK\r\n”,可知返回的状态码是 200。

Q5:When was the HTML file that you are retrieving last modified at the server?

A5:根据首部行"Last-Modified: Sat, 07 Nov 2020 06:59:01 GMT\r\n"，可知最后一次修改的时间是格林威治时间 2020 年 11 月 7 日 06:59:01。

Q6:How many bytes of content are being returned to your browser?

A6:根据首部行"Content-Length: 128\r\n"，可知返回的内容程度是 128 字节。

Q7:By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

A7:通过观察，请求报文 packeting-list window 中有首部行：Host,Connection,Upgrade-Insecure-Request,User_Agent,Accept,Accept-Encoding,Accept-Lauguage，与 packet content window 中所示相同。通过观察，响应报文 packeting-list window 中有首部行：Date,Server,Last-Modified、ETag、Accept-Ranges、Content-Length、Keep-Alive,Connection,Content-Type，与 packet content window 中所示相同。

The HTTP CONDITIONAL GET/response interaction

实验步骤

1.启动 Edge 浏览器，清除缓存。

2.启动 WireShark 分组嗅探器。

3.进入 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html。

4.刷新页面。

5.停止 WireShark 捕获分组

6.在显示过滤器上输入 http。

问题

Q1:Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

A1:观察第一个 GET 请求报文的首部行，并没有看到"IF-MODIFIED-SINCE"。

Q2:Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

A2:是的。在实体体中有"HTTP-wireshark-file2.html"这一文件的内容。

Q3:Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header?

A3:在第二个 GET 请求报文中有首部行"If-Modified-Since: Sat, 07 Nov 2020 06:59:01 GMT"。

Q4:What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

A4:对应第二个 GET 请求报文的响应报文的响应行为"HTTP/1.1 304 Not Modified\r\n"，状态码为 304，状态信息 Not Modified。在该响应报文中没有返回所请求的 html 文件的内容，因为服务器中的该文件相较于上次请求未发生修改，说

明本地缓存的该文件于所请求的内容相同，那么服务器只需告诉客户端未发生修改这一信息，便可实现其功能，这有利于减少网络负载。

Retrieving Long Documents

实验步骤

1.启动 Edge 浏览器，清除缓存。

2.启动 WireShark 分组嗅探器。

3.进入 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html。

4.刷新页面。

5.停止 WireShark 捕获分组。

6.在显示过滤器上输入 http。

问题

Q1:How many HTTP GET request messages were sent by your browser?

A1:如下图 4 可见，发送了 3 个 GET 请求报文。

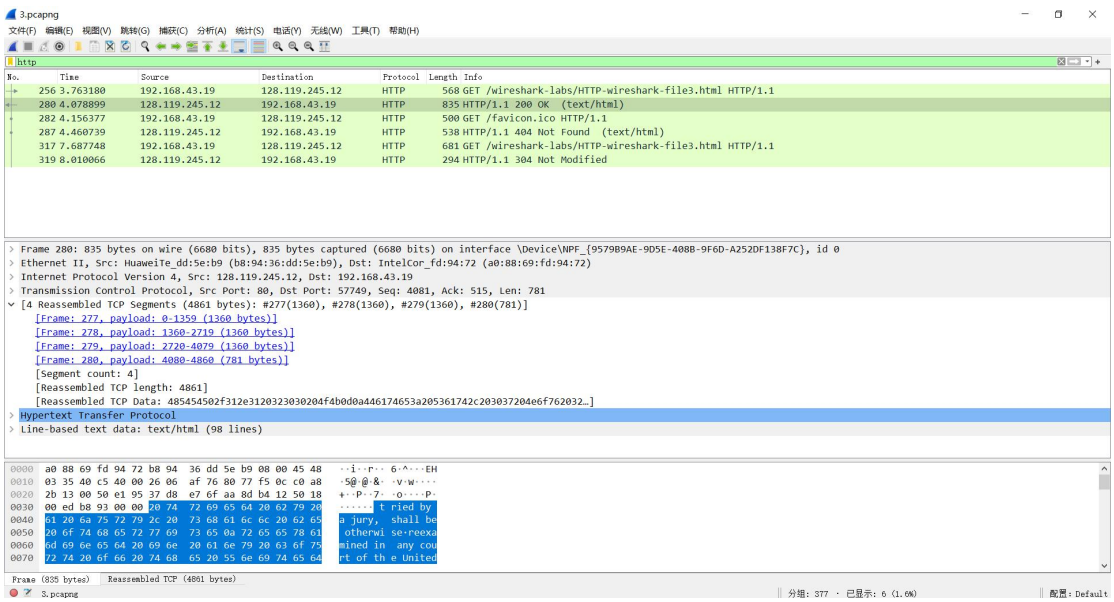

图 4

Q2:How many data-containing TCP segments were needed to carry the single HTTP response?

A2:如图 4 中 TCP 详细信息可见，这一 HTTP 响应报文分为了 4 个 TCP 报文段。

Q3:What is the status code and phrase associated with the response to the HTTP GET request?

A3:第一个 GET 请求报文(请求 html 文件)对应的响应报文的首部行为"HTTP/1.1 200 OK\r\n"。第二个 GET 请求报文(请求网站图标)对应的响应报文的首部行为"HTTP/1.1 404 Not Found\r\n"。第三个 GET 请求报文(再次请求同一 html 文件)对应的响应报文的首部行为"HTTP/1.1 304 Not Modified\r\n"。

Q4:Are there any HTTP status lines in the transmitted data associated with a TCP induced "Continuation"?

A4:根据封包详细信息，并没有在 HTTP 中发现和 TCP 引起的持续的相关信息。

HTML Documents with Embedded Objects

实验步骤

1.启动 Edge 浏览器，清除缓存。

2.启动 WireShark 分组嗅探器。

3.进入 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html。

4.停止 WireShark 捕获分组。

5.在显示过滤器上输入 http。

问题

Q1:How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?

A1:如图 5，浏览器发送了 3 个 GET 请求报文。第一个是发往 gaia.cs.umass.edu;第二个是发往 gaia.cs.umass.edu;第三个是发往 manic.cs.umass.edu。

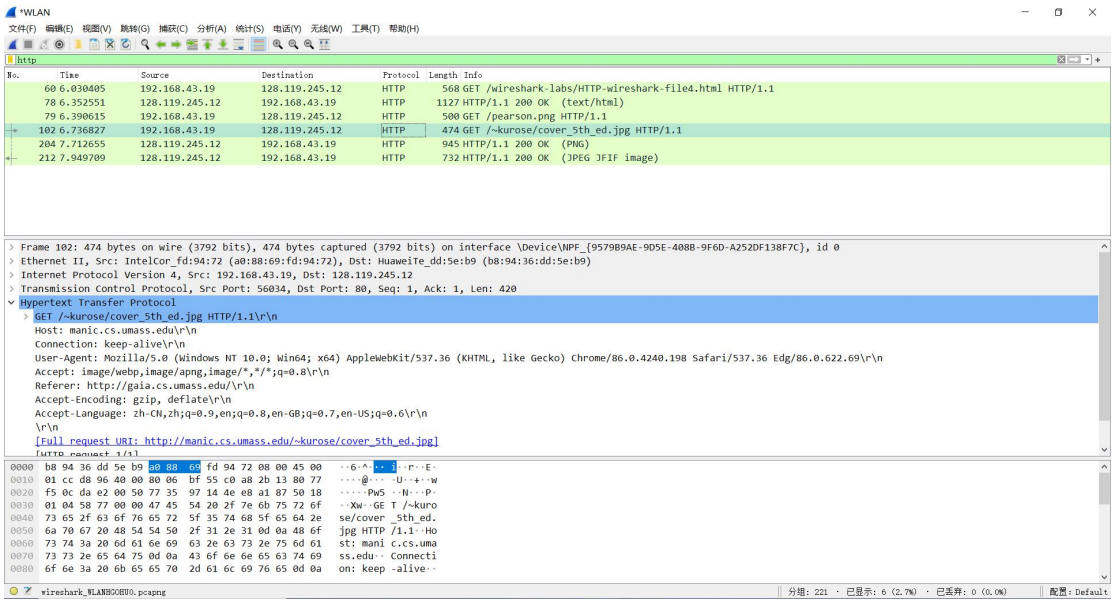

图 5

Q2:Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

A2:根据图 5，是并行下载。因为在发送请求 pearson.png 文件，接收对应响应报文之前，就发送了 cover_5th_ed.jpg 文件的请求报文。

HTTP Authentication

实验步骤

1.清除缓存，关闭浏览器，启动 Edge 浏览器。

2.启动 WireShark 分组嗅探器。

3.进入 http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html。

4.停止 WireShark 捕获分组。

5.在显示过滤器上输入 http。

问题

Q1:What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

A1:第一个 GET 请求报文对应的响应报文"HTTP/1.1 401 Unauthorized\r\n" 。

Q2:When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

A2:在第二次发送的 GET 请求报文中，新增首部行"Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n"。

结果分析

HTTP 中 GET 和 POST 请求方式的区别

HTTP 中 GET 方法请求指定的资源。使用 GET 的请求应该只用于获取数据。

HTTP 中 POST 方法发送数据给服务器。请求主体的类型由 Content-Type 首部指定。

一个 POST 请求通常是通过 HTML 表单发送，并返回服务器的修改结果。在这种情况下, content type 是通过在 <form> 元素中设置正确的 enctype 属性, 或是在 <input> 和 <button> 元素中设置 formenctype 属性来选择的:

- application/x-www-form-urlencoded: 数据被编码成以 '&' 分隔的键-值对，同时以 '=' 分隔键和值. 非字母或数字的字符会被 percent-encoding: 这也就是为什么这种类型不支持二进制数据(应使用 multipart/form-data 代替).
- multipart/form-data
- text/plain

当 POST 请求是通过除 HTML 表单之外的方式发送时，例如使用 XMLHttpRequest，那么请求主体可以是任何类型。按 HTTP 1.1 规范中描述，POST 为了以统一的方法来涵盖以下功能：

- 注释已有的资源

- 在公告板，新闻组，邮件列表或类似的文章组中发布消息;
- 通过注册新增用户;
- 向数据处理程序提供一批数据，例如提交一个表单;
- 通过追加操作，扩展数据库数据.

	GET 方法	POST 方法
请求是否有主体	否	是
成功的响应是否有主体	是	是
安全	是	否
幂等	是	否
可缓存	是	Only if freshness information is included
HTML 表单是否支持	是	是

表 1

总结

本次实验通过 WireShark 分组嗅探器捕获并分析 HTTP 报文，深入了解了 HTTP 请求报文、响应报文的结构，对报文中的内容有了深刻的理解，在回顾教材内容的同时又有所提升。各个小实验中体分别现了不同首部行的实际含义，深入浅出，为今后的理论学习和实验打下基础。

附：本次实验使用 WireShark 分组嗅探器获得的数据部分以截图形式体现于实验报告中，完整数据在 Github 仓库中。