中国科学技术大学计算机学院

计算机网络实验报告

实验二

利用Wireshark观察http报文

学　　号：PB17081504

姓　　名：廖洲洲

专　　业：计算机科学与技术

指导老师：张信明

中国科学技术大学计算机学院

2019年10月22日

一、 实验目的

1、了解HTTP 协议的层次结构；

2、熟悉HTTP 协议中用于消息传输的数据结构；

3、掌握HTTP 协议数据传输的原理。

二、 实验原理

1、Wireshark（前称Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是抓取网络封包，并尽可能显示出最为详细的网络封包资料。 Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换，监听共享网络上传送的数据包，但不能对其进行修改或者控制。

2、HTTP是一个简单的请求-响应协议，它通常运行在TCP之上。它指定了客户端可能发送给服务器什么样的消息以及得到什么样的响应。

三、 实验环境

1、 硬件

Lenovo XiaoXin CHAO7000
Intel® Core ™ i7-8500U CPU 1.80GHZ
内存：8G
显卡：NVIDIA GEFORCE 940MX

2、 软件

wireshark

四、 实验过程

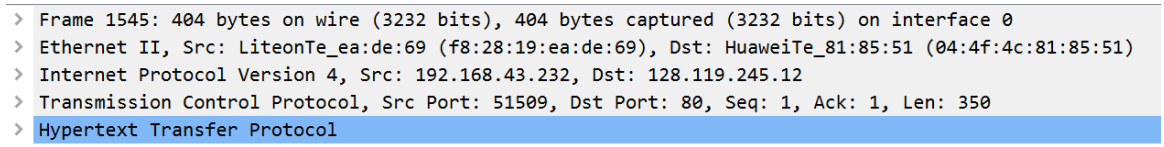
1）wireshark的安装

前往官网www.wireshark.org下载安装wireshark

2）实验分析

一、 第一题

1） http报文结构：



图中从上至下分别为：

Frame：物理层的数据帧概况

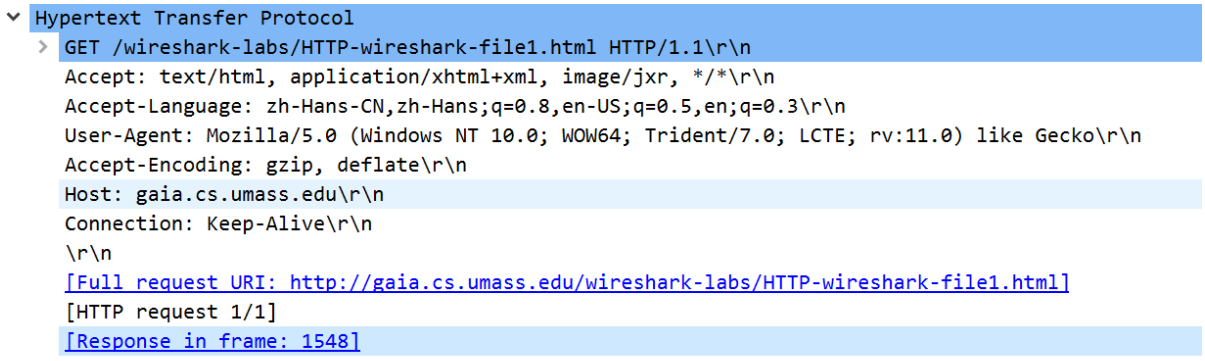
Ethernet II：数据链路层以太网帧头部信息

Internet Protocol Version 4：互联网层IP包头部信息

Transmission Control Protocol：传输层的数据段头部信息，此处是TCP

Hypertext Transfer Protocol：应用层的信息，此处是HTTP协议

http请求报文：



第一部分：请求行，用来说明请求类型,要访问的资源以及所使用HTTP版本.

第二部分：请求头部，紧接着请求行（即第一行）之后的部分，用来说明服务器要使用的附加信息。例如：HOST将指出请求的目的地。Connection是连接维持方式。Cache-Control控制缓存的行为等。

第三部分：空行，请求头部后面的空行是必须的。即使第四部分的请求数据为空，也必须有空行。

第四部分：请求数据也叫主体，可以添加任意的其他数据。

http响应报文：

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Date: Tue, 22 Oct 2019 12:37:41 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3\r\n
  Last-Modified: Tue, 22 Oct 2019 05:59:03 GMT\r\n
  ETag: "80-595797f538865"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 128\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.303249000 seconds]
  [Request in frame: 1545]
  [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
  File Data: 128 bytes
```

HTTP响应也由三个部分组成，分别是：状态行、响应头部、响应包体。状态行（status line）通过提供一个状态码来说明所请求的资源情况。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。状态代码由三位数字组成，第一个数字定义了响应的类别，且有五种可能取值。

2) 网页反馈的内容：

```
Line-based text data: text/html (4 lines)
    <html>\n
    Congratulations.  You've downloaded the file \n
    http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html!\n
    </html>\n
```

3) `GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n`

我的浏览器版本：HTTP 1.1

`HTTP/1.1 200 OK\r\n`

服务器浏览器版本：HTTP 1.1

4) `Internet Protocol Version 4, Src: 192.168.43.232, Dst: 128.119.245.12`

我的IP地址：192.168.43.232

服务器IP地址：128.119.245.12

5) `Last-Modified: Tue, 22 Oct 2019 05:59:03 GMT\r\n`

6) File Data: 128 bytes

二、 第二题

1) 第一次请求报文：

```
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
  Accept: text/html, application/xhtml+xml, image/jxr, */*\r\n
  Accept-Language: zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; LCTE; rv:11.0) like Gecko\r\n
  Accept-Encoding: gzip, deflate\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: Keep-Alive\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  [HTTP request 1/1]
  [Response in frame: 4856]
```

第一次响应报文：

```
Frame 4911: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits) on interface 0
Ethernet II, Src: HuaweiTe_81:85:51 (04:4f:4c:81:85:51), Dst: LiteonTe_ea:de:69 (f8:28:19:ea:de:69)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.232
Transmission Control Protocol, Src Port: 80, Dst Port: 51631, Seq: 1, Ack: 233, Len: 485
Hypertext Transfer Protocol
Line-based text data: text/html (7 lines)
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Date: Tue, 22 Oct 2019 12:54:02 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3\r\n
  Last-Modified: Tue, 22 Oct 2019 05:59:03 GMT\r\n
  ETag: "173-595797f52e06c"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 371\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.328233000 seconds]
  [Request in frame: 4852]
  [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  File Data: 371 bytes
```

第二次请求报文：

```
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
  Accept: text/html, application/xhtml+xml, image/jxr, */*\r\n
  Accept-Language: zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; LCTE; rv:11.0) like Gecko\r\n
  Accept-Encoding: gzip, deflate\r\n
  Host: gaia.cs.umass.edu\r\n
  If-Modified-Since: Tue, 22 Oct 2019 05:59:03 GMT\r\n
  If-None-Match: "173-595797f52e06c"\r\n
  Connection: Keep-Alive\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  [HTTP request 1/1]
  [Response in frame: 5449]
```

第二次响应报文：

```
Frame 5449: 294 bytes on wire (2352 bits), 294 bytes captured (2352 bits) on interface 0
Ethernet II, Src: HuaweiTe_81:85:51 (04:4f:4c:81:85:51), Dst: LiteonTe_ea:de:69 (f8:28:19:ea:de:69)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.232
Transmission Control Protocol, Src Port: 80, Dst Port: 51650, Seq: 1, Ack: 437, Len: 240
Hypertext Transfer Protocol
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
  Date: Tue, 22 Oct 2019 12:56:40 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.10 Perl/v5.16.3\r\n
  Connection: Keep-Alive\r\n
  Keep-Alive: timeout=5, max=100\r\n
  ETag: "173-595797f52e06c"\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.327151000 seconds]
  [Request in frame: 5446]
  [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

“IF-MODIFIED-SINCE”产生在第二次请求报文中：

```
If-Modified-Since: Tue, 22 Oct 2019 05:59:03 GMT\r\n
```

2) 第一次显式地返回了文件内容，第二次没有。从两次响应报文的响应包体可以看出，第一次有Line-based text data，第二次没有。这是因为发送HTTP请求时，客户会把浏览器缓存页面的最后修改时间一起发到服务器，服务器把这个时间和实际文件最后修改时间进行比对。时间一致，返回http状态码304，客户端接收到后，就直接把本地缓存文件显示到浏览器中。

三、 第三题

1)

5077	281.820631	192.168.43.232	128.119.245.12	HTTP	404 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
5087	282.151773	128.119.245.12	192.168.43.232	HTTP	835 HTTP/1.1 200 OK (text/html)
5148	282.732255	192.168.43.232	128.119.245.12	HTTP	286 GET /favicon.ico HTTP/1.1
5178	283.072803	128.119.245.12	192.168.43.232	HTTP	539 HTTP/1.1 404 Not Found (text/html)

发送了两个GET报文，第一个报文即普通的网页请求报文，而第二个

GET报文是浏览器请求网站根目录的favicon.ico图标。

2) 收到4个TCP信息段完成这个HTTP回应

```
[Frame: 45070, payload: 0-1359 (1360 bytes)]
[Frame: 45071, payload: 1360-2719 (1360 bytes)]
[Frame: 45072, payload: 2720-4079 (1360 bytes)]
[Frame: 45090, payload: 4080-4860 (781 bytes)]
[Segment count: 4]
[Reassembled TCP length: 4861]
[Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a2057…]
```

3) `HTTP/1.1 200 OK\r\n`

状态码200 OK

4) 没有，HTTP中没有“Continuation”信息。

四、 第四题

1)

```
 404 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
1127 HTTP/1.1 200 OK  (text/html)
 464 GET /pearson.png HTTP/1.1
 478 GET /~kurose/cover_5th_ed.jpg HTTP/1.1
 945 HTTP/1.1 200 OK  (PNG)
```

发送了3个请求信息

前两个发送至gaia.cs.umass.edu:

`Host: gaia.cs.umass.edu\r\n`

后一个发送至manic.cs.umass.edu:

`Host: manic.cs.umass.edu\r\n`

2) 分别从两个网站下载并行图片

8781	101.478013	172.20.10.3	128.119.245.12	HTTP	464 GET /pearson.png HTTP/1.1
8813	101.756160	172.20.10.3	128.119.245.12	HTTP	478 GET /~kurose/cover_5th_ed.jpg HTTP/1.1
8817	101.758349	128.119.245.12	172.20.10.3	HTTP	945 HTTP/1.1 200 OK (PNG)

因为在第一个请求报文发送后，马上又发送第二个请求报文，而没有等第一个请求回复才继续第二次请求。

五、 第五题

1) `HTTP/1.1 401 Unauthorized\r\n`

状态码和短语401 Unauthorized,表示发送的请求需要有通过HTTP认证的认证信息。

2) 多了

`Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n`

六、 第六题

（1）GET请求将请求的数据附在URL之后，POST将请求的数据放在HTTP的实体体中。而实际中浏览器或者操作系统可能对url长度进行限制，所以GET，POST此时所能传输的信息有区别。

（2）GET案例：个人主页-home.ustc.edu.cn/~liaozz/liaozz.html

http GET报文：

```
GET /~liaozz/liaozz.html HTTP/1.1\r\n
Host: home.ustc.edu.cn\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7\r\n
If-None-Match: "91f0075-2657-6da56f40"\r\n
If-Modified-Since: Wed, 09 Oct 2019 14:52:05 GMT\r\n
\r\n
[Full request URI: http://home.ustc.edu.cn/~liaozz/liaozz.html]
[HTTP request 1/1]
```

该报文向主机home.ustc.edu.cn请求对象/~liaozz/liaozz.html,同时告诉服务器使用持续连接。

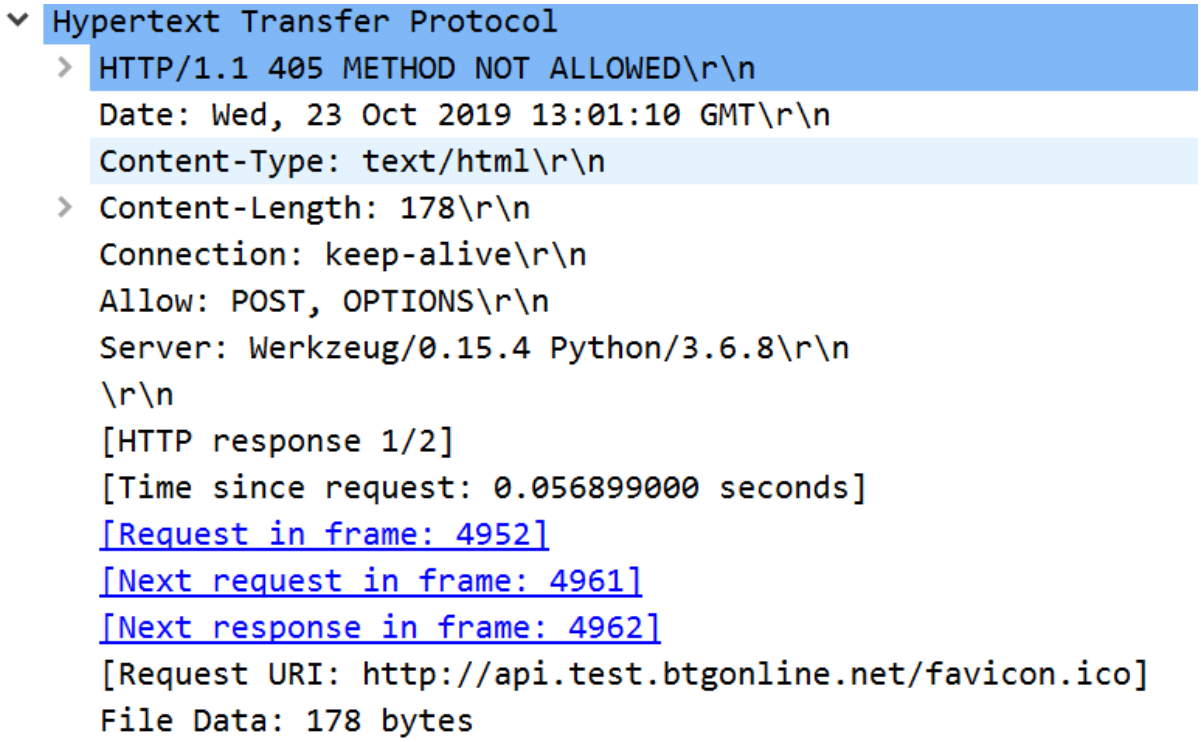
（3）POST案例：使用科大邮箱发送电子邮件

```
POST /coremail/XT3/pab/savercpt.jsp?sid=BAwUBdMMiQGCDpApIjMMilsoDyGzgEep HTTP/1.1\r\n
Host: mail.ustc.edu.cn\r\n
Connection: keep-alive\r\n
Content-Length: 268\r\n
Cache-Control: max-age=0\r\n
Origin: http://mail.ustc.edu.cn\r\n
Upgrade-Insecure-Requests: 1\r\n
Content-Type: application/x-www-form-urlencoded\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
Referer: http://mail.ustc.edu.cn/coremail/XT3/compose/composeAction.jsp?sid=BAwUBdMMiQGCDpApIjMMilsoDyGzgEep&urlfrom=&action=deliver&needAudit=undefin
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7\r\n
 [truncated]Cookie: uid=liaozz; domain=mail.ustc.edu.cn; uid=liaozz; domain=ustc.edu.cn; Coremail=e19725d4e329876f4a832384e9b014d0; CoremailReferer=ht
\r\n
[Full request URI: http://mail.ustc.edu.cn/coremail/XT3/pab/savercpt.jsp?sid=BAwUBdMMiQGCDpApIjMMilsoDyGzgEep]
[HTTP request 2/3]
[Prev request in frame: 24447]
[Response in frame: 24483]
[Next request in frame: 24485]
File Data: 268 bytes
```

该报文通过post方法向主机mail.ustc.edu.cn的/coremail/XT3/compose/composeAction.jsp?sid=BAwUBdMMiQGCDpApIjMMilsoDyGzgEep&urlfrom=&action=deliver&needAudit=undefined&smsAddrs= 发送邮

七、 第七题

在浏览器中输入下列网址：api.test.btgonline.net/chain/create_company_account
会出现405 method not allowed响应报文，这是因为浏览器发送了get请求，而方法只能处理post请求。与正常报文相比它出现了allow、server字段。



五、 实验总结

1. 分析HTTP协议的功能

（1）HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从WWW服务器传输超文本到本地浏览器的传输协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以哪部分内容首先显示等。

2. HTTP的原理

（1）HTTP由两个程序实现：一个客户程序和一个服务器程序。客户程序和服务器程序运行在不同的端系统中，通过交换HTTP报文进行会话。

（2）HTTP使用TCP作为它的支撑运输协议。HTTP客户首先发起与一个服务器的TCP连接，连接建立，浏览器可以和服务器进程就可以通过套接字接口访问TCP。

（3）HTTP是一个无状态协议。

3. HTTP报文

（1）请求报文

第一部分：请求行，用来说明请求类型,要访问的资源以及所使用HTTP版本.

第二部分：请求头部，紧接着请求行（即第一行）之后的部分，用来说明服务器要使用的附加信息。例如：HOST将指出请求的目的地。Connection是连接维持方式。Cache-Control控制缓存的行为等。

第三部分：空行，请求头部后面的空行是必须的。即使第四部分的请

求数据为空，也必须有空行。
第四部分：请求数据也叫主体，可以添加任意的其他数据。

（2）响应报文

HTTP响应也由三个部分组成，分别是：状态行、响应头部、响应包体。状态行（status line）通过提供一个状态码来说明所请求的资源情况。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。状态代码由三位数字组成，第一个数字定义了响应的类别，且有五种可能取值。