计算机网络与通信实验报告（二）

学　号	姓　名	班　级	报告日期
2022211939	杨涛	2211106	2024.10.30

实验内容	**利用分组嗅探器进行应用层协议分析**
实验目的	1. 了解传输层 TCP/UDP 协议构造。 2. 了解网络层 IP 协议构造。
实验预备知识	1. 详细掌握 TCP 段结构。 2. 详细掌握 UDP 段结构。 3. IP 数据报结构。 4. Ethereal 使用方式
实验过程描述	**一、利用工具分别对 TCP 套接字的实现及 UDP 套接字的实现捕包分析** 1. 利用工具对 TCP 套接字的实现捕包分析 a) 分别编写 TCP 客户端套接字程序、TCP 服务器端套接字程序 b) 将两个程序放置于两台设备上，并使其连接在同一局域网当中 c) 打卡捕包软件开始捕包；先运行服务器端，之后运行客户端 d) 当程序结束后终止捕包工作，并筛选出相关报文进行分析

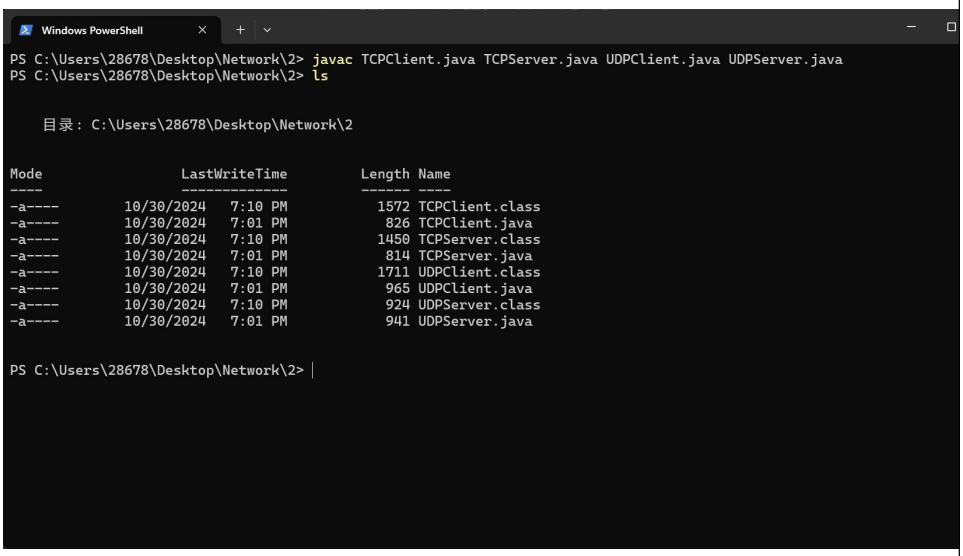


图 1 编译

图 2 启动 TCP 服务器

图 3 启动 TCP 客户端

No.	Time	Source	Destination	Protocol	Info
102	2.433224	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
103	2.433256	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SA
104	2.433280	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
282	7.737825	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=1
283	7.737856	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [ACK] Seq=1 Ack=2 Win=2161152 Len=0
284	7.737941	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [PSH, ACK] Seq=2 Ack=1 Win=2161152 Len=1
285	7.737950	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [ACK] Seq=1 Ack=3 Win=2161152 Len=0
286	7.737966	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [PSH, ACK] Seq=3 Ack=1 Win=2161152 Len=1
287	7.737972	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [ACK] Seq=1 Ack=4 Win=2161152 Len=0
288	7.737985	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [PSH, ACK] Seq=4 Ack=1 Win=2161152 Len=1
289	7.737992	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [ACK] Seq=1 Ack=5 Win=2161152 Len=0
290	7.738002	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [PSH, ACK] Seq=5 Ack=1 Win=2161152 Len=1
291	7.738008	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [ACK] Seq=1 Ack=6 Win=2161152 Len=0
292	7.738111	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [PSH, ACK] Seq=1 Ack=6 Win=2161152 Len=1
293	7.738141	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [ACK] Seq=6 Ack=2 Win=2161152 Len=0
294	7.738175	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [PSH, ACK] Seq=2 Ack=6 Win=2161152 Len=1
295	7.738187	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [ACK] Seq=6 Ack=3 Win=2161152 Len=0
296	7.738231	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [PSH, ACK] Seq=3 Ack=6 Win=2161152 Len=1
297	7.738251	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [ACK] Seq=6 Ack=4 Win=2161152 Len=0
298	7.738282	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [PSH, ACK] Seq=4 Ack=6 Win=2161152 Len=1
299	7.738298	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [ACK] Seq=6 Ack=5 Win=2161152 Len=0
300	7.738551	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [PSH, ACK] Seq=5 Ack=6 Win=2161152 Len=1
301	7.738570	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [ACK] Seq=6 Ack=6 Win=2161152 Len=0
302	7.741456	127.0.0.1	127.0.0.1	TCP	49179 → 6789 [FIN, ACK] Seq=6 Ack=6 Win=2161152 Len=0
303	7.741469	127.0.0.1	127.0.0.1	TCP	6789 → 49179 [ACK] Seq=6 Ack=7 Win=2161152 Len=0

图 4 捕包结果

2. 利用工具对 UDP 套接字的实现捕包分析
   a) 分别编写 UDP 客户端套接字程序、UDP 服务器端套接字程序
   b) 将两个程序放置于两台设备上，并使其连接在同一局域网当中
   c) 打卡捕包软件开始捕包；先运行服务器端，之后运行客户端
   d) 当程序结束后终止捕包工作，并筛选出相关报文进行分析

```
PS C:\Users\28678\Desktop\Network\out> java UDPServer
```

图 5 启动 UDP 服务器

```
PS C:\Users\28678\Desktop\Network\out> java UDPClient
fuck you
FROM SERVER:FUCK YOU
```

图 6 启动 UDP 客户端

No.	Time	Source	Destination	Protocol	Info
593	41.581488	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
594	41.581525	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SAC
595	41.581547	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
614	44.300997	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=1
615	44.301028	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [ACK] Seq=1 Ack=2 Win=2161152 Len=0
616	44.301277	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [PSH, ACK] Seq=2 Ack=1 Win=2161152 Len=1
617	44.301296	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [ACK] Seq=1 Ack=3 Win=2161152 Len=0
618	44.301319	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [PSH, ACK] Seq=3 Ack=1 Win=2161152 Len=1
619	44.301327	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [ACK] Seq=1 Ack=4 Win=2161152 Len=0
620	44.301342	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [PSH, ACK] Seq=4 Ack=1 Win=2161152 Len=1
621	44.301350	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [ACK] Seq=1 Ack=5 Win=2161152 Len=0
622	44.301363	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [PSH, ACK] Seq=5 Ack=1 Win=2161152 Len=1
623	44.301371	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [ACK] Seq=1 Ack=6 Win=2161152 Len=0
624	44.303001	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [PSH, ACK] Seq=1 Ack=6 Win=2161152 Len=1
625	44.303027	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [ACK] Seq=6 Ack=2 Win=2161152 Len=0
626	44.303061	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [PSH, ACK] Seq=2 Ack=6 Win=2161152 Len=1
627	44.303075	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [ACK] Seq=6 Ack=3 Win=2161152 Len=0
628	44.303107	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [PSH, ACK] Seq=3 Ack=6 Win=2161152 Len=1
629	44.303120	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [ACK] Seq=6 Ack=4 Win=2161152 Len=0
630	44.303144	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [PSH, ACK] Seq=4 Ack=6 Win=2161152 Len=1
631	44.303155	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [ACK] Seq=6 Ack=5 Win=2161152 Len=0
632	44.303176	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [PSH, ACK] Seq=5 Ack=6 Win=2161152 Len=1
633	44.303186	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [ACK] Seq=6 Ack=6 Win=2161152 Len=0
634	44.304296	127.0.0.1	127.0.0.1	TCP	50202 → 6789 [FIN, ACK] Seq=6 Ack=6 Win=2161152 Len=0
635	44.304317	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [ACK] Seq=6 Ack=7 Win=2161152 Len=0
860	90.342117	127.0.0.1	127.0.0.1	TCP	6789 → 50202 [RST, ACK] Seq=6 Ack=7 Win=0 Len=0

图 7 捕包结果

二、利用工具捕包分析协议 HTTP、FTP 和 DNS

1、 HTTP GET/response 交互

（1）启动浏览器，清空浏览器的缓存。

（2）启动分组俘获器，在窗口的显示过滤说明处输入“http”。

（3）一分钟以后，开始分组俘获。

（4）在浏览器的地址栏中输入以下 URL:

http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file1.html

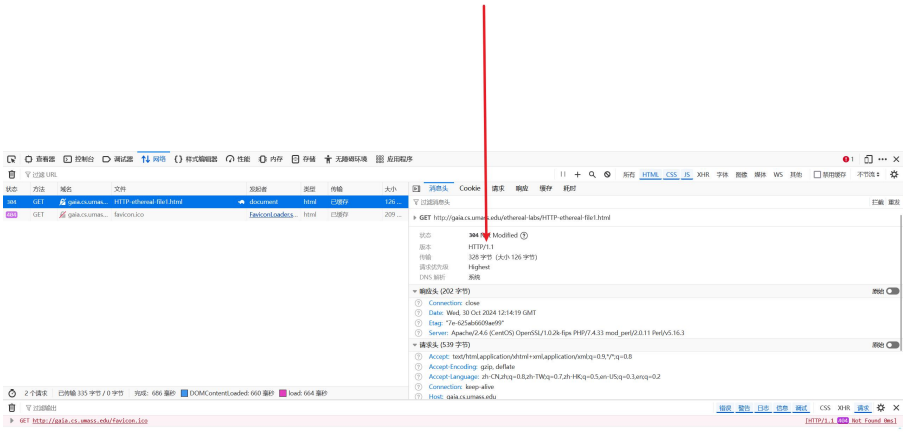
（5）停止分组俘获。

2、 HTTP 条件 GET/response 交互

（1） 启动浏览器，清空浏览器的缓存。

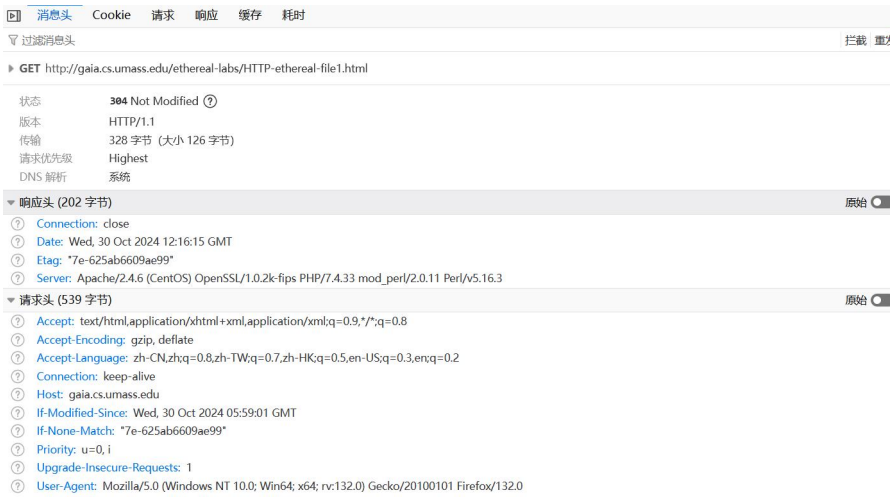
	（2） 启动分组俘获器，开始分组俘获。 （3） 在浏览器的地址栏中输入以下 URL: http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file2.html, （4） 在浏览器中重新输入相同的 URL 或单击浏览器中的“刷新”按钮。 （5） 停止分组俘获，在显示过滤筛选说明处输入“http”。 3、 获取长文件 （1） 启动浏览器，将浏览器的缓存清空。 （2） 启动分组俘获器，开始分组俘获。 （3） 在浏览器的地址栏中输入以下 URL: http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file3.html （4） 停止 Ethereal 分组俘获，在显示过滤筛选说明处输入“http”。 4、 嵌有对象的 HTML 文档 （1） 启动浏览器，将浏览器的缓存清空。 （2） 启动分组俘获器，开始分组俘获。 （3） 在浏览器的地址栏中输入以下 URL: http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file4.html， （4） 停止分组俘获，在显示过滤筛选说明处输入“http”。 5、 HTTP 认证 （1） 启动浏览器，将浏览器的缓存清空。 （2） 启动分组俘获器。开始分组俘获。 （3） 在浏览器的地址栏中输入以下 URL: http://gaia.cs.umass.edu/ethereal-labs/protected_pages/HTTP-ethereal-file5.html 并输入所需要的用户名和密码(用户名：eth-students,密码:network)。 （4） 停止分组俘获，在显示过滤筛选说明处输入“http”。 6、 跟踪 DNS （1） 开始分组俘获。 （2） 在浏览器的地址栏中输入：http://www.ietf.org （3） 停止分组俘获。 （7） 开始分组俘获。 （5） 执行命令：nslookup www.mit.edu （6） 停止分组俘获。 （7） 重复上面的实验，将命令替换为：nslookup –type=NS mit.edu （8） 重复上面的实验，将命令替换为：nslookup www.aiit.or.kr bitsy.mit.edu
实验结果	实验操作后的问题及相关回答: （1）你的浏览器运行的是 HTTP1.0，还是 HTTP1.1？你所访问的服务器所运行的 HTTP 版本号是多少？ **版本是 HTTP1.1（Firefox）**

Congratulations. You've downloaded the file http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file1.html!



（2）你的浏览器向服务器指出它能接收何种语言版本的对象？

在 accept-Language 中指明为 Zh-CN 语言



（3）你的计算机的 IP 地址是多少？服务器 gaia.cs.umass.edu 的 IP 地址是多少？

```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   IPv4 地址 . . . . . . . . . . . . : 192.168.15.161
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.15.96
```

```
PS C:\Users\28678\Desktop> nslookup gaia.cs.umass.edu
服务器:  UnKnown
Address:  192.168.15.96

非权威应答:
名称:    gaia.cs.umass.edu
Address:  128.119.245.12
```

（4）从服务器向你的浏览器返回的状态代码是多少？

状态	方法	域名	文件
200	GET	gaia.cs.umass.edu	HTTP-ethereal-file1.html
404	GET	gaia.cs.umass.edu	favicon.ico

（5）你从服务器上所获取的 HTML 文件的最后修改时间是多少？

GET http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file1.html

状态	200 OK
版本	HTTP/1.1
传输	447 字节（大小 126 字节）
请求优先级	Highest
DNS 解析	系统

响应头 (321 字节)　原始

Accept-Ranges: bytes
Connection: close
Content-Length: 126
Content-Type: text/html; charset=UTF-8
Date: Wed, 30 Oct 2024 12:22:39 GMT
Etag: "7e-625ab6609ae99"
Last-Modified: Wed, 30 Oct 2024 05:59:01 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3

请求头 (454 字节)　原始

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Connection: keep-alive
Host: gaia.cs.umass.edu
Priority: u=0, i
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gecko/20100101 Firefox/132.0

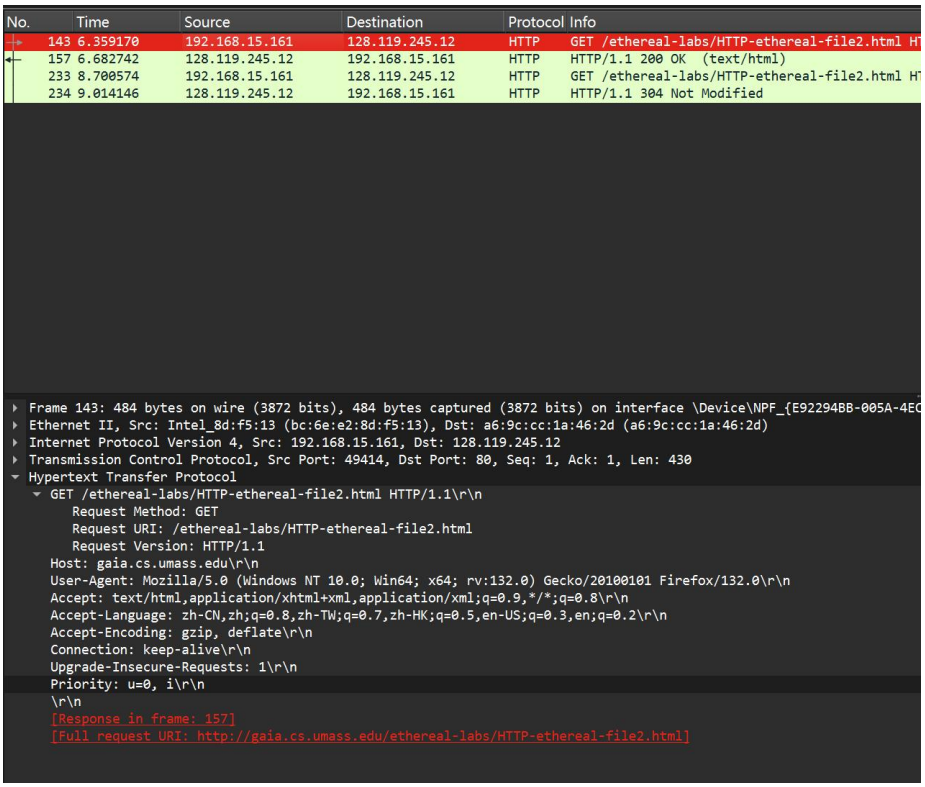
（6）返回到你的浏览器的内容一共多少字节？

126B

状态	方法	域...	文件	发起者	类型	传输	大小
200	GET	...	HTTP-et	document	html	447 字节	126 字节
404	GET	...	favicon.i	FaviconLoader.sys.mjs:175 (img)	html	448 字节	209 字节

（7）分析你的浏览器向服务器发出的第一个 HTTP GET 请求的内容，在该请求报文中，是否有一行是：IF-MODIFIED-SINCE？

没有



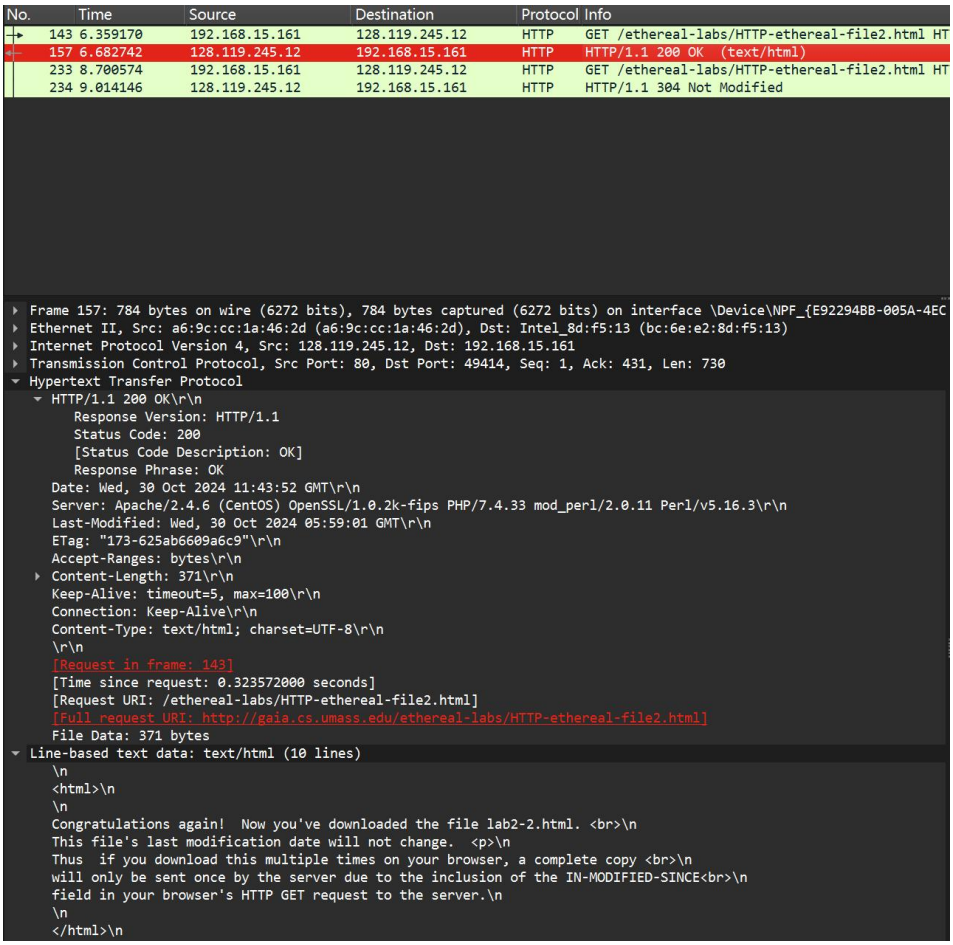
（8）分析服务器响应报文的内容，服务器是否明确返回了文件的内容？如何获知？

服务器返回了文件的内容。可以通过以下几方面来确认：

a. HTTP 状态码：在“Hypertext Transfer Protocol”部分，可以看到状态码 `200 OK`。HTTP 200 状态码表示服务器成功处理了请求并返回了所请求的资源。

b. Content-Type：响应头中包含了 `Content-Type: text/html; charset=UTF-8`，指明返回的内容类型为 HTML 文件。

c. 响应主体内容：在响应数据部分，展示了文件内容（HTML 代码），例如“Congratulations again!” 等字样。这部分内容是服务器返回的文件的内容。



（9）分析你的浏览器向服务器发出的第二个“HTTP GET”请求，在该请求报文中是否有一行是：IF-MODIFIED-SINCE？如果有，在该首部行后面跟着的信息是什么？

报文中有 IF-MODIFIED-SINCE。日期表示资源最后修改的时间。如果服务器上的资源在这个时间之后没有被修改，它会返回一个 304 Not Modified 状态，表示客户端可以使用缓存的版本，而不需要重新下载资源。如果资源在这个时间之后被修改，服务器会返回最新的资源和一个 200 OK 状态。这样可以减少不必要的数据传输，提高效率。

No.	Time	Source	Destination	Protocol	Info
143	6.359170	192.168.15.161	128.119.245.12	HTTP	GET /ethereal-labs/HTTP-ethereal-file2.html HTTP/1.1
157	6.682742	128.119.245.12	192.168.15.161	HTTP	HTTP/1.1 200 OK (text/html)
233	8.700574	192.168.15.161	128.119.245.12	HTTP	GET /ethereal-labs/HTTP-ethereal-file2.html HTTP/1.1
234	9.014146	128.119.245.12	192.168.15.161	HTTP	HTTP/1.1 304 Not Modified

```
Frame 233: 570 bytes on wire (4560 bits), 570 bytes captured (4560 bits) on interface \Device\NPF_{E92294BB-005A-4EC7-832E
Ethernet II, Src: Intel_8d:f5:13 (bc:6e:e2:8d:f5:13), Dst: a6:9c:cc:1a:46:2d (a6:9c:cc:1a:46:2d)
Internet Protocol Version 4, Src: 192.168.15.161, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 49414, Dst Port: 80, Seq: 431, Ack: 731, Len: 516
Hypertext Transfer Protocol
  GET /ethereal-labs/HTTP-ethereal-file2.html HTTP/1.1\r\n
      Request Method: GET
      Request URI: /ethereal-labs/HTTP-ethereal-file2.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gecko/20100101 Firefox/132.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Wed, 30 Oct 2024 05:59:01 GMT\r\n
    If-None-Match: "173-625ab6609a6c9"\r\n
    Priority: u=0, i\r\n
    \r\n
    [Response in frame: 234]
    [Full request URI: http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file2.html]
```

（10）服务器对第二个 HTTP GET 请求的响应中的 HTTP 状态代码是多少？服务器是否明确返回了文件的内容？请解释。

服务器对第二个 HTTP GET 请求的响应状态码是 **304 Not Modified**。这表明服务器没有返回文件的内容，因为资源自上次客户端请求的时间以来并未改变。

在 304 Not Modified 状态下，服务器会告诉客户端资源未更改，因此不需要重新传输文件内容。客户端可以使用本地缓存的文件副本，而不必下载新的数据，有助于减少网络带宽的使用。

（11）你的浏览器一共发出了多少个 HTTP GET 请求？

1 个（捕了十来次😢）

No.	Time	Source	Destination	Protocol	Info
173	2.708632	192.168.105.161	128.119.245.12	HTTP	GET /ethereal-labs/HTTP-ethereal-file3.html HTTP/1.1
181	3.166380	128.119.245.12	192.168.105.161	HTTP	HTTP/1.1 200 OK (text/html)

（12）承载这一个 HTTP 响应报文一共需要多少个 data-containing TCP 报文段？

4 个

```
[4 Reassembled TCP Segments (4861 bytes): #176(1360), #177(1360), #180(1360), #181(781)]
    [Frame: 176, payload: 0-1359 (1360 bytes)]
    [Frame: 177, payload: 1360-2719 (1360 bytes)]
    [Frame: 180, payload: 2720-4079 (1360 bytes)]
    [Frame: 181, payload: 4080-4860 (781 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 4861]
```

（13）与这个 HTTP GET 请求相对应的响应报文的状态代码和状态短语是什么？

200ok

```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 OK\r\n
    Date: Sun, 03 Nov 2024 0[illegible]:15:43 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Sat, 02 Nov 2024 05:[illegible]:02 GMT\r\n
    ETag: "1194-625e7bf94d701"\r\n
    Accept-Ranges: bytes\r\n
  ▸ Content-Length: 4500\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
    [Request in frame: 173]
    [Time since request: 0.457748000 seconds]
    [Request URI: /ethereal-labs/HTTP-ethereal-file3.html]
    [Full request URI: http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file3.html]
    File Data: 4500 bytes
```

（14）你的浏览器一共发出了多少个 HTTP GET 请求？这些请求被发送到的目的地的 IP 地址是多少？

共有 3 个 ：

第一个是 128.119.245.12，

第二个是 52.51.131.59，

第三个是 128.117.245.12

No.	Time	Source	Destination	Protocol	Info
118	6.015426	192.168.105.161	128.119.245.12	HTTP	GET /ethereal-labs/HTTP-ethereal-file4.html HTTP/1.1
128	6.315877	128.119.245.12	192.168.105.161	HTTP	HTTP/1.1 200 OK (text/html)
142	6.546249	192.168.105.161	52.51.131.59	HTTP	GET /catalog/images/pearson-logo-footer.gif HTTP/1.1
148	6.646314	192.168.105.161	128.119.245.12	HTTP	GET /~kurose/cover.jpg HTTP/1.1
151	6.736730	52.51.131.59	192.168.105.161	HTTP	HTTP/1.1 403 Forbidden (text/html)
278	8.076191	128.119.245.12	192.168.105.161	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)

（15）浏览器在下载这两个图片时，是串行下载还是并行下载？请解释。

是并行下载。原因是：

从时间戳上看，第一个请求在 6.315877 秒处得到了响应（200 OK），而第二个请求紧接在 6.546249 秒处发出。由于两个请求在非常短的时间间隔内发起，可以判断浏览器是并行下载这两个图片的。

（16）对于浏览器发出的最初的 HTTP GET 请求，服务器的响应是什么(状态代码和状态短语)?

No.	Time	Source	Destination	Protocol	Info
19	1.745925	192.168.105.161	128.119.245.12	HTTP	GET /ethereal-labs/protected_pages/HTTP-ethereal-file5.html HTTP/1.1
42	2.075582	128.119.245.12	192.168.105.161	HTTP	HTTP/1.1 401 Unauthorized (text/html)
111	9.824258	192.168.105.161	128.119.245.12	HTTP	GET /ethereal-labs/protected_pages/HTTP-ethereal-file5.html HTTP/1.1
115	10.121774	128.119.245.12	192.168.105.161	HTTP	HTTP/1.1 200 OK (text/html)
116	10.154131	192.168.105.161	128.119.245.12	HTTP	GET /favicon.ico HTTP/1.1
150	10.442819	128.119.245.12	192.168.105.161	HTTP	HTTP/1.1 404 Not Found (text/html)

401 Unauthorized

```
▸ Ethernet II, Src: a6:9c:cc:1a:46:2d (a6:9c:cc:1a:46:2d), Dst: Intel_8d:f5:13 (bc:6e:e2:8d:f5:13)
▸ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.105.161
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 55679, Seq: 1, Ack: 487, Len: 711
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 401 Unauthorized\r\n
      Response Version: HTTP/1.1
      Status Code: 401
      [Status Code Description: Unauthorized]
      Response Phrase: Unauthorized
    Date: Sun, 03 Nov 2024 02:50:18 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    WWW-Authenticate: Basic realm="eth-students only"\r\n
  ▸ Content-Length: 381\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    \r\n
    [Request in frame: 19]
    [Time since request: 0.329657000 seconds]
    [Request URI: /ethereal-labs/protected_pages/HTTP-ethereal-file5.html]
    [Full request URI: http://gaia.cs.umass.edu/ethereal-labs/protected_pages/HTTP-ethereal-file5.html]
    File Data: 381 bytes
▸ Line-based text data: text/html (12 lines)
```

（17）当浏览器发出第二个 HTTP GET 请求时，在 HTTP GET 报文中包含了哪些新的字段？

用户名-密码 – Authorization 行

```
▾ Hypertext Transfer Protocol
  ▾ GET /ethereal-labs/protected_pages/HTTP-ethereal-file5.html HTTP/1.1\r\n
      Request Method: GET
      Request URI: /ethereal-labs/protected_pages/HTTP-ethereal-file5.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
  ▾ Authorization: Basic ZXRoLXN0dWRlbnRzOm5ldHdvcms=\r\n
      Credentials: eth-students:network
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Saf
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9\r\n
    \r\n
    [Response in frame: 115]
    [Full request URI: http://gaia.cs.umass.edu/ethereal-labs/protected_pages/HTTP-ethereal-file5.html]
```

（18）定位到 DNS 查询报文和查询响应报文，这两种报文的发送是基于 UDP 还是基于 TCP 的？

UDP

```
▸ Frame 743: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{E92294BB
▸ Ethernet II, Src: Intel_8d:f5:13 (bc:6e:e2:8d:f5:13), Dst: RuijieNetwor_a6:8f:09 (58:69:6c:a6:8f:09)
▸ Internet Protocol Version 4, Src: 10.236.248.40, Dst: 172.26.26.3
▾ User Datagram Protocol, Src Port: 58223, Dst Port: 53
    Source Port: 58223
    Destination Port: 53
    Length: 44
    Checksum: 0xc96f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 14]
    [Stream Packet Number: 1]
  ▸ [Timestamps]
    UDP payload (36 bytes)
▸ Domain Name System (query)
```

（19）DNS 查询报文的目的端口号是多少？DNS 查询响应报文的源端口号是多少？

查询报文的目的端口-53；应报文的源端口-53

```
▸ Frame 743: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{E922
▸ Ethernet II, Src: Intel_8d:f5:13 (bc:6e:e2:8d:f5:13), Dst: RuijieNetwor_a6:8f:09 (58:69:6c:a6:8f:09
▸ Internet Protocol Version 4, Src: 10.236.248.40, Dst: 172.26.26.3
▾ User Datagram Protocol, Src Port: 58223, Dst Port: 53
    Source Port: 58223
    Destination Port: 53
    Length: 44
    Checksum: 0xc96f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 14]
    [Stream Packet Number: 1]
  ▸ [Timestamps]
    UDP payload (36 bytes)
▸ Domain Name System (query)
```

```
Frame 798: 553 bytes on wire (4424 bits), 553 bytes captured (4424 bits) on interface \Device\NPF_{E92294BB-005A-4
Ethernet II, Src: RuijieNetwor_a6:8f:09 (58:69:6c:a6:8f:09), Dst: Intel_8d:f5:13 (bc:6e:e2:8d:f5:13)
Internet Protocol Version 4, Src: 172.26.26.3, Dst: 10.236.248.40
User Datagram Protocol, Src Port: 53, Dst Port: 58223
   Source Port: 53
   Destination Port: 58223
   Length: 519
   Checksum: 0xdc2d [unverified]
   [Checksum Status: Unverified]
   [Stream index: 14]
   [Stream Packet Number: 2]
 ▸ [Timestamps]
   UDP payload (511 bytes)
Domain Name System (response)
```

（20）DNS 查询报文发送的目的地的 IP 地址是多少？利用 ipconfig 命令（ipconfig/all）决定你主机的本地 DNS 服务器的 IP 地址。这两个地指相同吗？

查询报文发送目的 IP：172.26.26.3

```
▸ Frame 743: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{E9229
▸ Ethernet II, Src: Intel_8d:f5:13 (bc:6e:e2:8d:f5:13), Dst: RuijieNetwor_a6:8f:09 (58:69:6c:a6:8f:09)
▸ Internet Protocol Version 4, Src: 10.236.248.40, Dst: 172.26.26.3
▾ User Datagram Protocol, Src Port: 58223, Dst Port: 53
    Source Port: 58223
    Destination Port: 53
    Length: 44
    Checksum: 0xc96f [unverified]
    [Checksum Status: Unverified]
    [Stream index: 14]
    [Stream Packet Number: 1]
  ▸ [Timestamps]
    UDP payload (36 bytes)
▸ Domain Name System (query)
```

本机 DNS：172.26.26.3

两个地址是相同的。

```
Windows PowerShell
   DHCP 已启用 . . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . . : 是
   IPv4 地址 . . . . . . . . . . . . . : 10.236.248.40(首选)
   子网掩码 . . . . . . . . . . . . . : 255.255.0.0
   获得租约的时间 . . . . . . . . . . : 2024年11月3日星期日 上午11:21:18
   租约过期的时间 . . . . . . . . . . : 2024年11月3日星期日 下午2:24:20
   默认网关. . . . . . . . . . . . . : 10.236.255.254
   DHCP 服务器 . . . . . . . . . . . : 10.236.255.254
   DNS 服务器 . . . . . . . . . . . : 172.26.26.3
                                       218.201.96.130
   TCPIP 上的 NetBIOS . . . . . . . . : 已启用

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
   物理地址. . . . . . . . . . . . . : BC-6E-E2-8D-F5-17
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   物理地址. . . . . . . . . . . . . : 6C-24-08-28-EA-FE
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
PS C:\Users\28678\Desktop>
```

（21）检查 DNS 查询报文，它是哪一类型的 DNS 查询？该查询报文中包含“answers”吗？

属于 A 类型和 HTTP 类型,报文中不包含 answers。

```
    UDP payload (36 bytes)
▾ Domain Name System (query)
    Transaction ID: 0x0829
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ analytics.ietf.org: type A, class IN
        Name: analytics.ietf.org
        [Name Length: 18]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
    [Response In: 799]
```

```
Domain Name System (query)
    Transaction ID: 0x6929
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ analytics.ietf.org: type HTTPS, class IN
        Name: analytics.ietf.org
        [Name Length: 18]
        [Label Count: 3]
        Type: HTTPS (65) (HTTPS Specific Service Endpoints)
        Class: IN (0x0001)
    [Response In: 798]
```

（22）检查 DNS 查询响应报文，其中提供了多少个“answers”？每个 answers 包含哪些内容？

A 的查询响应提供了 2 个 answers

```
Answers
  ▾ analytics.ietf.org: type A, class IN, addr 104.16.44.99
      Name: analytics.ietf.org
      Type: A (1) (Host Address)
      Class: IN (0x0001)
      Time to live: 62 (1 minute, 2 seconds)
      Data length: 4
      Address: 104.16.44.99
  ▾ analytics.ietf.org: type A, class IN, addr 104.16.45.99
      Name: analytics.ietf.org
      Type: A (1) (Host Address)
      Class: IN (0x0001)
      Time to live: 62 (1 minute, 2 seconds)
      Data length: 4
      Address: 104.16.45.99
```

HTTP 的查询响应提供了 1 个 answers

```
Answers
  ▾ analytics.ietf.org: type HTTPS, class IN
      Name: analytics.ietf.org
      Type: HTTPS (65) (HTTPS Specific Service Endpoints)
      Class: IN (0x0001)
      Time to live: 6 (6 seconds)
      Data length: 61
      SvcPriority: 1
      TargetName: <Root>
    ▸ SvcParam: alpn=h3,h2
    ▸ SvcParam: ipv4hint=104.16.44.99,104.16.45.99
    ▸ SvcParam: ipv6hint=2606:4700::6810:2c63,2606:4700::6810:2d63
```

（23）考虑一下你的主机发送的 subsequent(并发)TCP SYN 分组， SYN 分组的目的 IP 地址是否与在 DNS 查询响应报文中提供的某个 IP 地址相对应？

是的。SYN 分组的目的 IP 地址是否与在 DNS 查询响应报文中提供的某个 IP 地址。

（24）打开的 WEB 页中包含图片，在获取每一个图片之前，你的主机发出新的 DNS 查询了吗？

没有

（25）DNS 查询报文的目的端口号是多少？DNS 查询响应报文的源端口号是多少？

目的：53　　响应的源端口：53

```
User Datagram Protocol, Src Port: 65098, Dst Port: 53
    Source Port: 65098
    Destination Port: 53
    Length: 37
    Checksum: 0xc968 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 16]
    [Stream Packet Number: 1]
  [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (29 bytes)
```

```
User Datagram Protocol, Src Port: 53, Dst Port: 65098
    Source Port: 53
    Destination Port: 65098
    Length: 126
    Checksum: 0x8d22 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 16]
    [Stream Packet Number: 3]
  [Timestamps]
      [Time since first frame: 0.007308000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (118 bytes)
```

（26）DNS 查询报文发送的目的地的 IP 地址是多少？这个地址是你的默认本地 DNS 服务器的地址吗？

是

```
Internet Protocol Version 4, Src: 10.236.248.40, Dst: 172.26.26.3
User Datagram Protocol, Src Port: 65098, Dst Port: 53
    Source Port: 65098
    Destination Port: 53
    Length: 37
    Checksum: 0xc968 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 16]
    [Stream Packet Number: 1]
  [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (29 bytes)
Domain Name System (query)
```

（27）检查 DNS 查询报文，它是哪一类型的 DNS 查询？该查询报文中包含“answers”吗？

A 类型，没有 answer

```
Domain Name System (query)
    Transaction ID: 0x0002
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▸ Queries
    [Response In: 701]
```

（28）检查 DNS 查询响应报文，其中提供了多少个“answers”？每个 answers 包含哪些内容？

3 个 answer：

```
▾ Answers
  ▾ www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
      Name: www.mit.edu
      Type: CNAME (5) (Canonical NAME for an alias)
      Class: IN (0x0001)
      Time to live: 81535 (22 hours, 38 minutes, 55 seconds)
      Data length: 25
      CNAME: www.mit.edu.edgekey.net
  ▾ www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
      Name: www.mit.edu.edgekey.net
      Type: CNAME (5) (Canonical NAME for an alias)
      Class: IN (0x0001)
      Time to live: 81535 (22 hours, 38 minutes, 55 seconds)
      Data length: 24
      CNAME: e9566.dscb.akamaiedge.net
  ▾ e9566.dscb.akamaiedge.net: type A, class IN, addr 184.84.55.33
      Name: e9566.dscb.akamaiedge.net
      Type: A (1) (Host Address)
      Class: IN (0x0001)
      Time to live: 81535 (22 hours, 38 minutes, 55 seconds)
      Data length: 4
      Address: 184.84.55.33
  [Retransmitted response. Original response in: 701]
  [Retransmission: True]
```

（29）DNS 查询报文发送的目的地的 IP 地址是多少？这个地址是你的默认本地 DNS 服务器的地址吗？

172.26.26.3 是本地 DNS 服务器

```
› Internet Protocol Version 4, Src: 192.168.128.63, Dst: 192.168.128.237
```

```
▸ Internet Protocol Version 4, Src: 10.236.248.40, Dst: 172.26.26.3
▾ User Datagram Protocol, Src Port: 61844, Dst Port: 53
    Source Port: 61844
    Destination Port: 53
    Length: 33
    Checksum: 0xc964 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
  ▾ [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (25 bytes)
```

（30）检查 DNS 查询报文，它是哪一类型的 DNS 查询？该查询报文中包含“answers”吗？

A 类型查询 不包含 answers

```
▼ Domain Name System (query)
    Transaction ID: 0x07d8
  ▶ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ mit.edu: type A, class IN
        Name: mit.edu
        [Name Length: 7]
        [Label Count: 2]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
    [Response In: 151]
```

（31）检查 DNS 查询响应报文，其中响应报文提供了哪些 MIT 名称服务器？响应报文提供这些 MIT 名称服务器的 IP 地址了吗？q

提供了 3 个名称服务器，没提供 IP 地址

```
▼ Domain Name System (response)
  ▶ Transaction ID: 0xcab3
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ www.baidu.com: type A, class IN
        Name: www.baidu.com
        [Name Length: 13]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
  ▼ Answers
    ▼ www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
        Name: www.baidu.com
        Type: CNAME (5) (Canonical NAME for an alias)
        Class: IN (0x0001)
        Time to live: 1107 (18 minutes, 27 seconds)
        Data length: 15
        CNAME: www.a.shifen.com
    ▼ www.a.shifen.com: type A, class IN, addr 39.156.66.18
        Name: www.a.shifen.com
        Type: A (1) (Host Address)
        Class: IN (0x0001)
        Time to live: 52 (52 seconds)
        Data length: 4
        Address: 39.156.66.18
    ▼ www.a.shifen.com: type A, class IN, addr 39.156.66.14
        Name: www.a.shifen.com
        Type: A (1) (Host Address)
        Class: IN (0x0001)
        Time to live: 52 (52 seconds)
        Data length: 4
        Address: 39.156.66.14
    [Retransmitted response. Original response in: 323]
    [Retransmission: True]
```

（32）DNS 查询报文发送的目的地的 IP 地址是多少？这个地址是你的默认本地 DNS 服务器的地址吗？如果不是，这个 IP 地址相当于什么？

```
▸ Internet Protocol Version 4, Src: 10.236.248.40, Dst: 172.26.26.3
▾ User Datagram Protocol, Src Port: 60964, Dst Port: 53
    Source Port: 60964
    Destination Port: 53
    Length: 39
    Checksum: 0xc96a [unverified]
    [Checksum Status: Unverified]
    [Stream index: 2]
    [Stream Packet Number: 1]
  ▾ [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (31 bytes)
▾ Domain Name System (query)
    Transaction ID: 0x8917
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
```

（33）检查 DNS 查询报文，它是哪一类型的 DNS 查询？该查询报文中包含“answers”吗？

A 类型的 DNS 查询，不包含 answers

```
▾ Domain Name System (query)
    Transaction ID: 0x8917
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ bitsy.mit.edu: type A, class IN
        Name: bitsy.mit.edu
        [Name Length: 13]
        [Label Count: 3]
        Type: A (1) (Host Address)
        Class: IN (0x0001)
    [Response In: 107]
```

（34）检查 DNS 查询响应报文，其中提供了多少个“answers”？每个 answers 包含哪些内容？

1 个包含域名和地址

<table>
<tr><td></td><td>

```
Domain Name System (response)
    Transaction ID: 0x8917
    Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 8
    Additional RRs: 9
    Queries
    Answers
        bitsy.mit.edu: type A, class IN, addr 18.0.72.3
            Name: bitsy.mit.edu
            Type: A (1) (Host Address)
            Class: IN (0x0001)
            Time to live: 7 (7 seconds)
            Data length: 4
            Address: 18.0.72.3
    Authoritative nameservers
    Additional records
    [Retransmitted response. Original response in: 107]
    [Retransmission: True]
```

</td></tr>
<tr><td>实验当中问题及解决方法</td><td>1、TCP 套接字分组捕获时，服务器端无法获取正确的 TCP 报文，所有来自 Client 的报文均是坏包？
解决：分析过后，是客户端访问的服务器不可访问，手动修改为 127.0.0.1 即可正常访问。

2、在访问某些指定网页时捕获不到任何包？
解决：将 https 转为 http 并且不使用无痕模式，即可捕到包。

3、有些时候捕包会经常返回奇怪的 403，取不到数据并且会神奇的发送多次 get 请求？
解决：打开手机热点，不要使用校园网！</td></tr>
<tr><td>成绩（教师打分）</td><td>优秀</td><td>良好</td><td>及格</td><td>不及格</td></tr>
</table>