

# LAB-HTTP 2020.5.5

## Step 1: Manual GET with Telnet

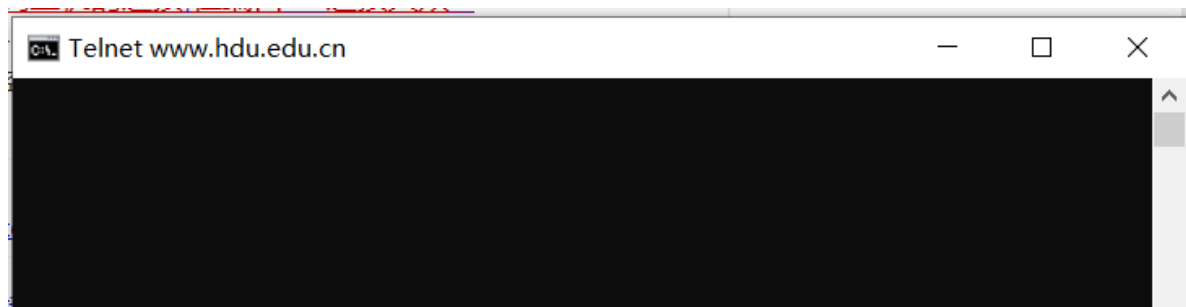
选用网页: <http://www.hdu.edu.cn/index.php> 航电的主页

<http://www.hdu.edu.cn/index.php> 的服务器名是 [www.hdu.edu.cn](http://www.hdu.edu.cn), 路径是/index.php

尝试用telnet获得网页:

### 1 用telnet连接服务器

```
telnet www.hdu.edu 80
```



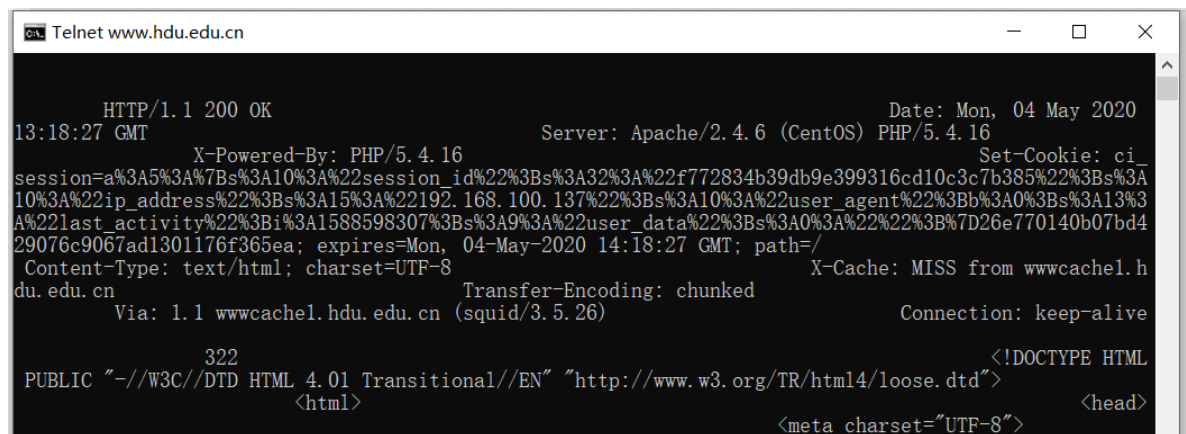
连接成功 (win10需要以管理员身份运行)

### 2 get网页

输入:

```
GET /index.php HTTP/1.1
```

```
Host: www.hdu.edu.cn
```



得到了一个html网页, 网页的内容都显示在了命令行中。

### 3 关闭

```
<div class="side-time">
ek"></div>
</div>
<div class="side-data"></div>
</div>
<script src="/asset/home/js/jquery-1.9.1.min.js"></script>
<script src="/asset/home/js/unslder-min.js"></script>
<script src="/asset/home/js/index.js"></script>
<script src="/asset/home/js/side.js"></script>
</body>
</html>

0

遗失对主机的连接。
C:\WINDOWS\system32>
```

过了一会 自己关闭了。

## 问题1 服务器的HTTP版本

HTTP/1.1 (大部分服务器都是)

## 问题2 客户端如何识别服务器发送的内容的开头?

看内容的前几位是不是HTTP

## 问题3 客户端如何知道返回的内容类型?

看返回的Content-type字段, 例子中的就是text/html类型。

# Step 2: Capture a Trace

## 1 找到两个网站

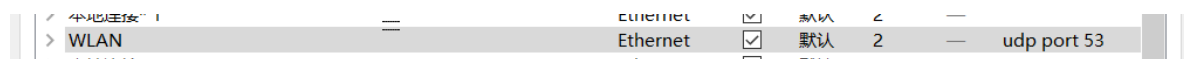
第一个: <http://www.hdu.edu.cn/asset/home/images/logo.png> 航电的logo

第二个: <http://www.hdu.edu.cn/index.php> 航电的主页

## 2 准备

关闭无关的网页。

## 3 设置filter

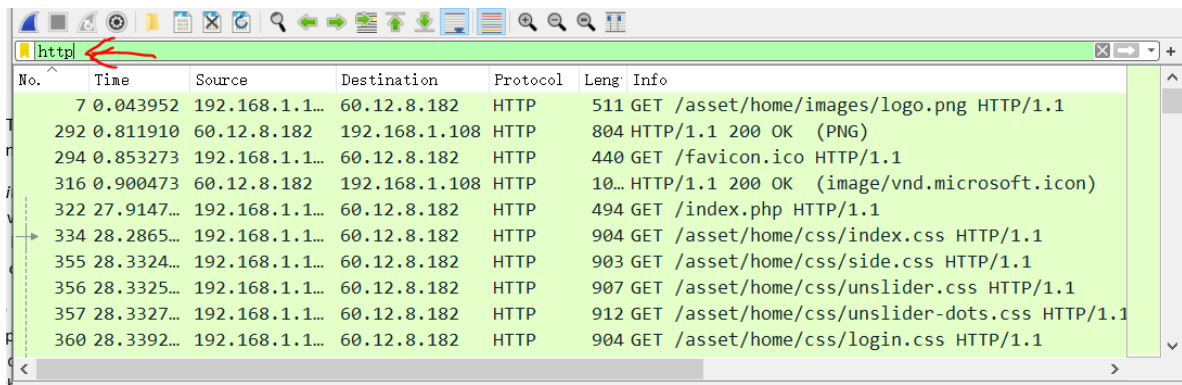


## 4 捕获

按步骤 先图片, 等待十秒, 再图片, 再等待十秒, 最后访问主页。

No.	Time	Source	Destination	Prot	Leng	Info
1	0.000000	192.168.1.1...	192.168.1.1	DNS	77	Standard query 0x0edd A dss0.bdstatic.com
2	0.002529	192.168.1.1...	192.168.1.1	DNS	77	Standard query 0x76bd A dss1.bdstatic.com
3	0.003305	192.168.1.1	192.168.1.108	DNS	364	Standard query response 0x0edd A dss0.bdstatic.com CNAME
4	0.004706	192.168.1.1...	192.168.1.1	DNS	76	Standard query 0x8c9e A ss1.bdstatic.com
5	0.005842	192.168.1.1	192.168.1.108	DNS	364	Standard query response 0x76bd A dss1.bdstatic.com CNAME
6	0.006433	192.168.1.1...	192.168.1.1	DNS	73	Standard query 0xe968 A sp0.baidu.com
7	0.009299	192.168.1.1	192.168.1.108	DNS	76	Standard query response 0x8c9e Refused A ss1.bdstatic.com
8	0.010167	192.168.1.1...	192.168.1.1	DNS	73	Standard query 0x9bda A sp1.baidu.com
9	0.012146	192.168.1.1	192.168.1.108	DNS	302	Standard query response 0xe968 A sp0.baidu.com CNAME www
10	0.014203	192.168.1.1	192.168.1.108	DNS	302	Standard query response 0x9bda A sp1.baidu.com CNAME www
11	0.017126	192.168.1.1...	192.168.1.1	DNS	73	Standard query 0x6bf2 A sp2.baidu.com
12	0.021744	192.168.1.1	192.168.1.108	DNS	135	Standard query response 0x6bf2 A sp2.baidu.com CNAME www
13	0.411713	192.168.1.1...	192.168.1.1	DNS	73	Standard query 0xc325 NS www.baidu.com
14	0.414522	192.168.1.1	192.168.1.108	DNS	73	Standard query response 0xc325 Refused NS www.baidu.com

## 5 停止



No.	Time	Source	Destination	Protocol	Leng	Info
7	0.043952	192.168.1.1...	60.12.8.182	HTTP	511	GET /asset/home/images/logo.png HTTP/1.1
292	0.811910	60.12.8.182	192.168.1.108	HTTP	804	HTTP/1.1 200 OK (PNG)
294	0.853273	192.168.1.1...	60.12.8.182	HTTP	440	GET /favicon.ico HTTP/1.1
316	0.900473	60.12.8.182	192.168.1.108	HTTP	10...	HTTP/1.1 200 OK (image/vnd.microsoft.icon)
322	27.9147...	192.168.1.1...	60.12.8.182	HTTP	494	GET /index.php HTTP/1.1
334	28.2865...	192.168.1.1...	60.12.8.182	HTTP	904	GET /asset/home/css/index.css HTTP/1.1
355	28.3324...	192.168.1.1...	60.12.8.182	HTTP	903	GET /asset/home/css/side.css HTTP/1.1
356	28.3325...	192.168.1.1...	60.12.8.182	HTTP	907	GET /asset/home/css/unslider.css HTTP/1.1
357	28.3327...	192.168.1.1...	60.12.8.182	HTTP	912	GET /asset/home/css/unslider-dots.css HTTP/1.1
360	28.3392...	192.168.1.1...	60.12.8.182	HTTP	904	GET /asset/home/css/login.css HTTP/1.1

只看HTTP协议的内容。

## Step 3: Inspect the Trace

filter的添加和上一张图一样。

### 第一个GET

No.	Time	Source	Destination	Prot	Leng	Info
1	0.000000	192.168.1.1...	61.135.165.2...	DNS	87	Standard query 0x8fed A www.a.shifen.com OPT
2	0.041813	61.135.165.2...	192.168.1.108	DNS	278	Standard query response 0x8fed A www.a.shifen.com A 61.135.165.2...

可以看到：

Host: [www.hdu.edu.cn](http://www.hdu.edu.cn) 服务器及其端口的域名（航电服务器）

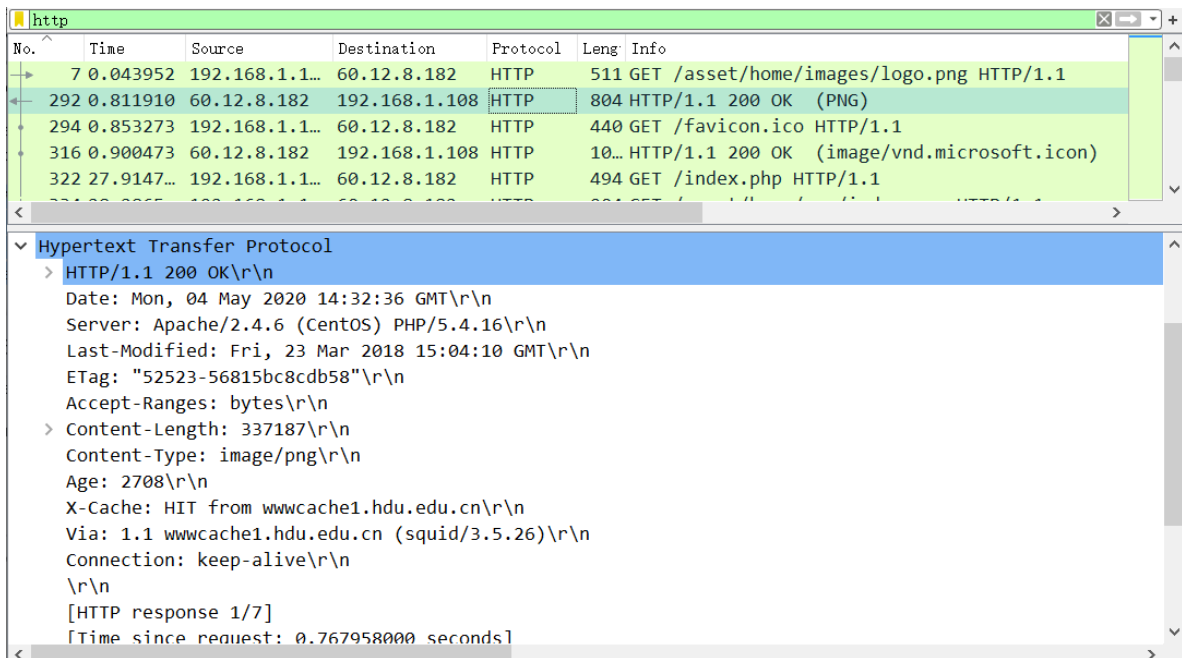
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/79.0.3945.130 Safari/537.36 浏览器的类型及其功能（chrome浏览器）

Accept、Accept-Encoding等 对响应中可接受的格式(如文本/html)的描述，包括其编码(如gzip)和语言

无Cookie 估计这个访问用不到cookie，涉及到登陆的网站会用cookie

### 第一个RESPONSE



No.	Time	Source	Destination	Protocol	Leng	Info
7	0.043952	192.168.1.1...	60.12.8.182	HTTP	511	GET /asset/home/images/logo.png HTTP/1.1
292	0.811910	60.12.8.182	192.168.1.108	HTTP	804	HTTP/1.1 200 OK (PNG)
294	0.853273	192.168.1.1...	60.12.8.182	HTTP	440	GET /favicon.ico HTTP/1.1
316	0.900473	60.12.8.182	192.168.1.108	HTTP	10...	HTTP/1.1 200 OK (image/vnd.microsoft.icon)
322	27.9147...	192.168.1.1...	60.12.8.182	HTTP	494	GET /index.php HTTP/1.1

Hypertext Transfer Protocol	
>	HTTP/1.1 200 OK\r\n
	Date: Mon, 04 May 2020 14:32:36 GMT\r\n
	Server: Apache/2.4.6 (CentOS) PHP/5.4.16\r\n
	Last-Modified: Fri, 23 Mar 2018 15:04:10 GMT\r\n
	ETag: "52523-56815bc8cdb58"\r\n
	Accept-Ranges: bytes\r\n
>	Content-Length: 337187\r\n
	Content-Type: image/png\r\n
	Age: 2708\r\n
	X-Cache: HIT from wwwcache1.hdu.edu.cn\r\n
	Via: 1.1 wwwcache1.hdu.edu.cn (squid/3.5.26)\r\n
	Connection: keep-alive\r\n
	\r\n
	[HTTP response 1/7]
	[Time since request: 0.767958000 seconds]

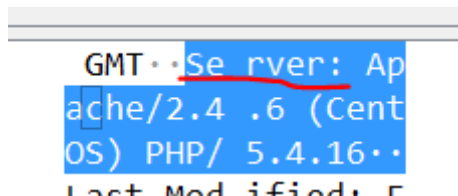
可以看到200 OK的表头

Server: Apache/2.4.6 (CentOS) PHP/5.4.17 服务端的类型及其功能

Date, Last-Modified 响应时间和内容最后修改时间

ETag: "52523-56815bc8cdb58"

问题1 header line的格式是什么?给出一个符合你所看到的标题的简单描述。



GMT Server: Apache/2.4.6 (CentOS) PHP/5.4.16

每一行header包涵了一个字段的信息、冒号和 值。header之间以\r\n分开。

问题2 使用什么标头来指示响应中返回的内容的种类和长度?

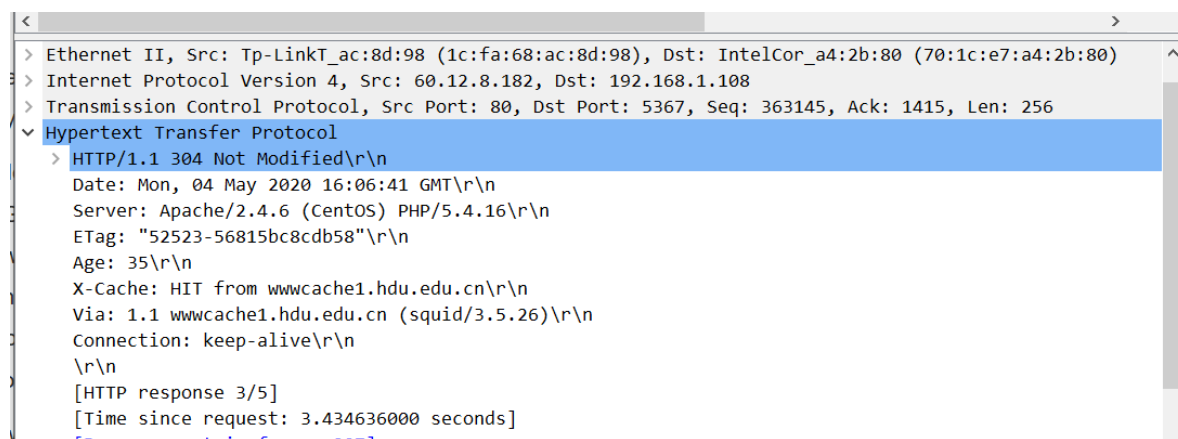
种类: Content-Type 数量: Content-Length

## Step 4: Content Caching

由于刚才没有能够对图片进行成功的第二次访问,所以在清除数据后,再做了一次捕获。

No.	Time	Source	Destination	Protocol	Length	Info
5	0.047562	192.168.1.1...	60.12.8.182	HTTP	511	GET /asset/home/images/logo.png HTTP/1.1
295	0.712198	60.12.8.182	192.168.1.108	HTTP	292	HTTP/1.1 200 OK (PNG)
297	0.768359	192.168.1.1...	60.12.8.182	HTTP	440	GET /favicon.ico HTTP/1.1
318	0.818781	60.12.8.182	192.168.1.108	HTTP	10...	HTTP/1.1 200 OK (image/vnd.microsoft.icon)
336	44.1012...	192.168.1.1...	123.59.22.210	HTTP	166	POST /api/v1/update HTTP/1.1
342	46.3696...	123.59.22.2...	192.168.1.108	HTTP	393	HTTP/1.1 200 OK (text/html)
352	53.3023...	192.168.1.1...	60.12.8.182	HTTP	625	GET /asset/home/images/logo.png HTTP/1.1
366	56.7370...	60.12.8.182	192.168.1.108	HTTP	310	HTTP/1.1 304 Not Modified
372	60.4086...	192.168.1.1...	61.135.217.24	HTTP	597	GET /fsearch?keyfrom=deskdict.screentrans.http
375	60.5064...	61.135.217....	192.168.1.108	HTTP/X...	224	HTTP/1.1 200

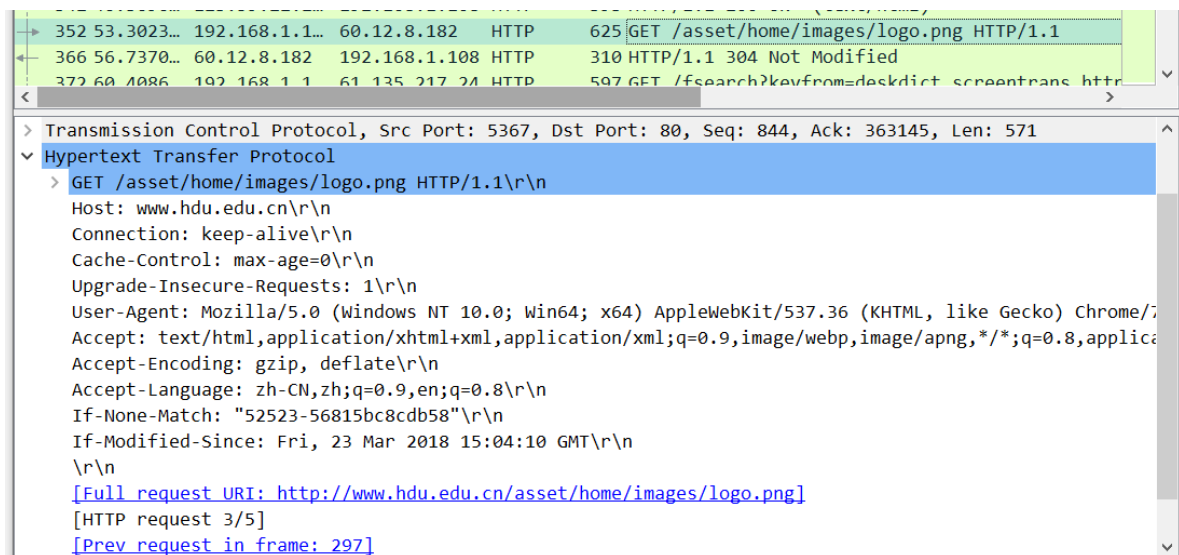
这一次捕获到了对于图片再次请求时得到的304 Not Modified, 现在打开这个包看看:



> Ethernet II, Src: Tp-LinkT\_ac:8d:98 (1c:fa:68:ac:8d:98), Dst: IntelCor\_a4:2b:80 (70:1c:e7:a4:2b:80)  
> Internet Protocol Version 4, Src: 60.12.8.182, Dst: 192.168.1.108  
> Transmission Control Protocol, Src Port: 80, Dst Port: 5367, Seq: 363145, Ack: 1415, Len: 256  
▼ Hypertext Transfer Protocol  
    > HTTP/1.1 304 Not Modified\r\n    Date: Mon, 04 May 2020 16:06:41 GMT\r\n    Server: Apache/2.4.6 (CentOS) PHP/5.4.16\r\n    ETag: "52523-56815bc8cdb58"\r\n    Age: 35\r\n    X-Cache: HIT from wwwcache1.hdu.edu.cn\r\n    Via: 1.1 wwwcache1.hdu.edu.cn (squid/3.5.26)\r\n    Connection: keep-alive\r\n    \r\n    [HTTP response 3/5]  
    [Time since request: 3.434636000 seconds]  
    [New request in frame: 307]

问题1 浏览器发送来让服务器决定是否发送新内容的标题的名称是什么?

打开浏览器发送的那个包:



发现三个新加的字段： Upgrade-Insecure-Requests、 If-None-Match 和 If-Modified-Since。

应该时第一个字段和第三个字段配合来让服务器决定。其中第一个字段是让服务器知道要去检查一下，第三个字段是给服务器提供是否返回新图片的线索（如果新修改的时间晚于第三个字段的时间，那么就要传新的。否则不传）

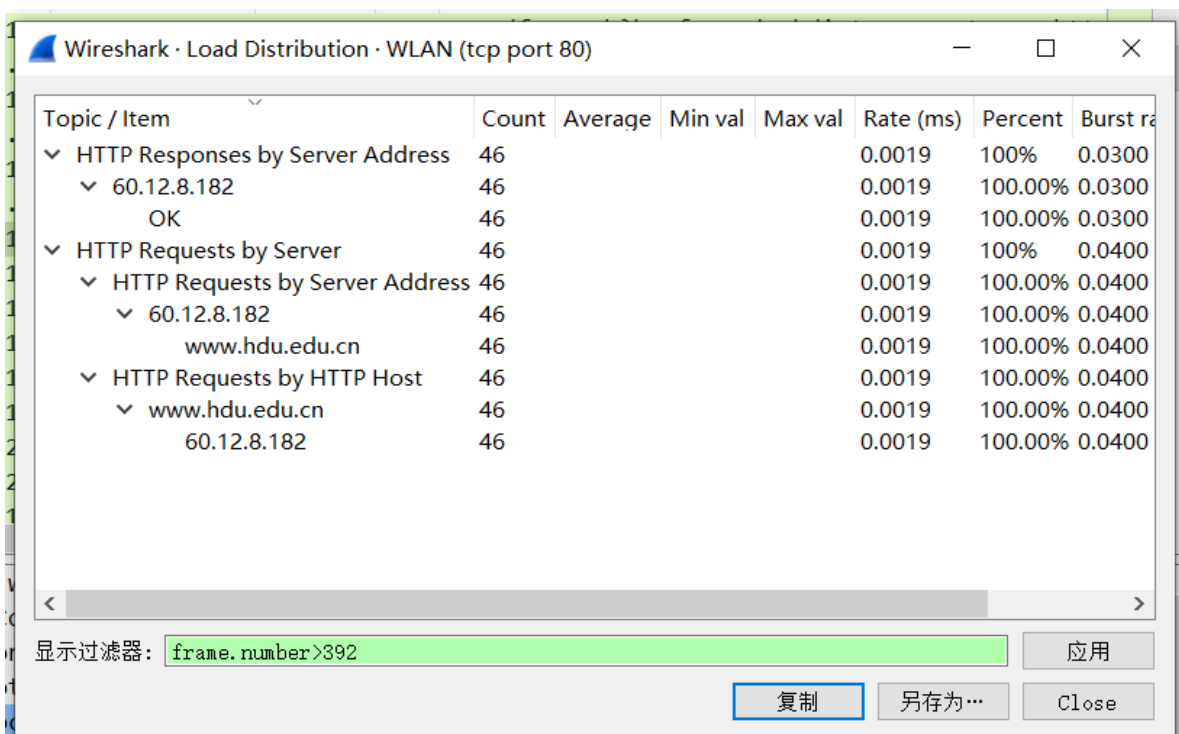
## 问题2 报头携带的时间戳值究竟来自何处？

时间戳值来自最近的下载的内容的“Last-Modified”字段。它是最后一次更改内容时的服务器时间，它不是下载时的时间。

## Step 5: Complex Pages

是从第393个包开始的http 是找[www.hdu.edu.cn](http://www.hdu.edu.cn)的内容

然后就以 frame.number>392为条件 创建一个HTTP/Load Distribution:



创建一个 Packet Counter panel:

Wireshark · Packet Counter · WLAN (tcp port 80)

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst
▼ Total HTTP Packets	92				0.0038	100%	0.0600	63.174
Other HTTP Packets	0				0.0000	0.00%	-	-
▼ HTTP Response Packets	46				0.0019	50.00%	0.0300	63.374
??? : broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	0				0.0000	0.00%	-	-
▼ 2xx: Success	46				0.0019	100.00%	0.0300	63.374
200 OK	46				0.0019	100.00%	0.0300	63.374
1xx: Informational	0				0.0000	0.00%	-	-
▼ HTTP Request Packets	46				0.0019	50.00%	0.0400	63.174
GET	46				0.0019	100.00%	0.0400	63.174

显示过滤器: `frame.number>392`      应用      复制      另存为...      Close

创建一个请求面板:

Wireshark · Requests · WLAN (tcp port 80)

Topic / Item	Count	Average	Min val	Max val
▼ HTTP Requests by HTTP Host	46			
▼ www.hdu.edu.cn	46			
/uploads/images/20200430/202004301509481000.jpg	1			
/uploads/images/20200430/202004301002541000.jpg	1			
/uploads/images/20200428/202004281600511000.jpg	1			
/uploads/images/20200426/202004261422081000.jpg	1			
/uploads/images/20200424/202004241402161000.JPG	1			
/uploads/images/20200423/202004231246411000.jpg	1			
/uploads/images/20200416/202004161536411000.png	1			
/uploads/images/20190912/201909121632111000.jpg	1			
/uploads/images/20190401/201904011018051000.jpg	1			
/uploads/images/20181214/201812141531321000.jpg	1			
/uploads/images/20170329/201703290113271000.jpg	1			
/uploads/images/20170329/201703290053281000.jpg	1			

显示过滤器: `frame.number>392`      应用      复制      另存为...      Close

谷歌浏览器自带的查看请求的窗口:

