



哈尔滨工业大学  
Harbin Institute of Technology

# 计算机网络 课程实验报告

实验名称	利用 Wireshark 进行协议分析					
姓名	袁野		院系	计算学部计算机科学与技术专业		
班级	1903102		学号	1190200122		
任课教师	刘亚维		指导教师	刘亚维		
实验地点	格物 207		实验时间	2021.11.21		
实验课表现	出勤、表现得分(10)		实验报告 得分(40)		实验总分	
	操作结果得分(50)					
教师评语						



## 实验目的:

熟悉并掌握 Wireshark 的基本操作, 了解网络协议实体间进行交互以及报文交换的情况。

## 实验内容:

概述本次实验的主要内容, 包含的实验项等。

- 1) 学习 Wireshark 的使用
- 2) 利用 Wireshark 分析 HTTP 协议
- 3) 利用 Wireshark 分析 TCP 协议
- 4) 利用 Wireshark 分析 IP 协议
- 5) 利用 Wireshark 分析 Ethernet 数据帧
- 6) 利用 Wireshark 分析 DNS 协议
- 7) 利用 Wireshark 分析 UDP 协议
- 8) 利用 Wireshark 分析 ARP 协议

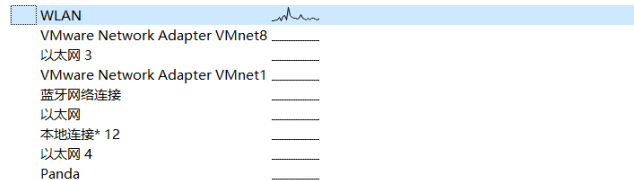
## 实验过程与结果:

### 1. Wireshark 的使用

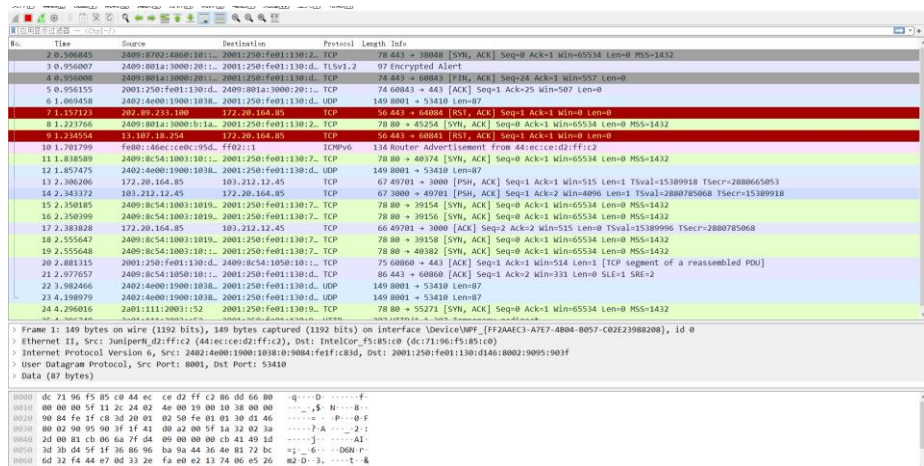
首先在Wireshark官网: <https://www.wireshark.org/download.html> 下载Wireshark, 之后捕获器选择接口进行捕获。

捕获

... 使用这个过滤器:  显示所有接口



抓包界面如下所示



我们在浏览器中访问<http://www.hit.edu.cn>，待网页加载完成后，停止分组捕获。

No.	Time	Source	Destination	Protocol	Length	Info
32135	345.997527	240e:928:1400:10:123	2001:250:fe01:130:d...	TCP	86	80 → 56268 [SYN, ACK] Seq=0 Ack=1 Win=28400 Len=0 MSS=1420 SACK_PERM=1 WS=1024
32136	345.997583	2001:250:fe01:130:d...	240e:928:1400:10:123	TCP	74	56268 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
32137	345.997641	2001:250:fe01:130:d...	240e:928:1400:10:123	TCP	356	56268 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=282 [TCP segment of a reassembled PDU]
32138	346.015926	240e:928:1400:10:123	2001:250:fe01:130:d...	TCP	78	443 → 57472 [SYN, ACK] Seq=0 Ack=1 Win=65534 Len=0 MSS=1432
32139	346.091076	240e:928:1400:10:123	2001:250:fe01:130:d...	TCP	74	80 → 56268 [ACK] Seq=1 Ack=283 Win=29696 Len=0
32140	346.091144	2001:250:fe01:130:d...	240e:928:1400:10:123	HTTP	294	POST /cgi-bin/httpconn HTTP/1.1
32141	346.118220	240e:928:1400:10:123	2001:250:fe01:130:d...	UDP	149	8001 → 53410 Len=87
32142	346.176551	240e:928:1400:10:123	2001:250:fe01:130:d...	TCP	74	80 → 56268 [ACK] Seq=1 Ack=503 Win=30720 Len=0
32143	346.188865	240e:928:1400:10:123	2001:250:fe01:130:d...	HTTP	336	HTTP/1.1 200 OK (text/plain)
32144	346.231528	2001:250:fe01:130:d...	240e:928:1400:10:123	TCP	74	56268 → 80 [ACK] Seq=503 Ack=263 Win=65274 Len=0
32145	346.280248	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32146	346.310801	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32147	346.342595	2001:250:fe01:130:d...	240e:928:1400:10:123	UDP	109	53410 → 8001 Len=47
32148	346.450993	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32149	346.464390	240e:928:1400:10:123	2001:250:fe01:130:d...	UDP	709	8001 → 53410 Len=647
32150	346.465368	2001:250:fe01:130:d...	240e:928:1400:10:123	UDP	109	53410 → 8001 Len=47
32151	346.512775	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32152	346.512777	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32153	346.512932	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32154	346.501714	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
32155	346.506190	240e:928:1400:10:123	2001:250:fe01:130:d...	UDP	117	8001 → 53410 Len=55
32156	346.856731	172.20.164.85	216.58.200.46	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 56268 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

在过滤器中输入http进行过滤：

No.	Time	Source	Destination	Protocol	Length	Info
23821	332.636406	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23828	332.642126	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23829	332.642128	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23844	332.649909	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23846	332.649910	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23847	332.649910	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23848	332.649911	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23849	332.649911	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23856	332.653313	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23858	332.653314	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23863	332.653315	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23864	332.653316	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23865	332.653316	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	[TCP Previous segment not captured] Continuation
23867	332.653316	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23878	332.653319	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23879	332.653319	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	Continuation
23881	332.653320	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	765	Continuation
24187	332.795231	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1483	HTTP/1.1 200 OK (PNG)
24288	332.838884	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	490	GET /upload/tpl/02/c8/712/template712/images/favicon.ico HTTP/1.1
24311	332.851076	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1514	[TCP Previous segment not captured] Continuation
32140	346.091144	2001:250:fe01:130:d...	240e:928:1400:10:123	HTTP	294	POST /cgi-bin/httpconn HTTP/1.1
32143	346.188865	240e:928:1400:10:123	2001:250:fe01:130:d...	HTTP	336	HTTP/1.1 200 OK (text/plain)

## 2. HTTP分析

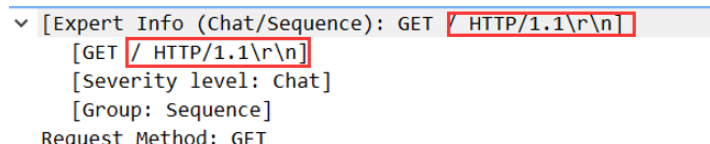
### 2.1 HTTP GET/response 交互

首先清除浏览器的缓存，然后过滤出HTTP后重新开始捕获，在浏览器访问 <http://jwes.hit.edu.cn/>

No.	Time	Source	Destination	Protocol	Length	Info
306	4.281328	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	375	GET / HTTP/1.1
404	4.460033	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	576	HTTP/1.1 200 OK (text/html)
411	4.568884	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	445	GET /upload/tpl/02/c8/712/template712/extends/extends.js HTTP/1.1
412	4.570923	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	426	GET /upload/tpl/02/c8/712/template712/css/animate.min.css HTTP/1.1
422	4.587882	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	519	GET /upload/article/images/e9/72/fd45395c417bfc6e468da810/a955d7be-44d6-4d00-95de-f28371275ad9.png HTTP/1.1
425	4.593133	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	452	GET /upload/site/00/ee/238/logo.png HTTP/1.1
428	4.595536	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	423	GET /upload/tpl/02/c8/712/template712/css/iconfont.css HTTP/1.1
431	4.596718	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	420	GET /upload/tpl/02/c8/712/template712/css/slick.css HTTP/1.1
433	4.613924	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	370	HTTP/1.1 200 OK (application/javascript)
447	4.617190	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	443	GET /upload/tpl/02/c8/712/template712/js/jquery.min.js HTTP/1.1
457	4.627310	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	872	HTTP/1.1 200 OK (text/css)
479	4.630017	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	426	GET /upload/tpl/02/c8/712/template712/extends/extends.css HTTP/1.1
484	4.631898	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	717	HTTP/1.1 200 OK (text/css)
487	4.633346	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	453	GET /upload/tpl/02/c8/712/template712/extends/lib/jquery.min.js HTTP/1.1
506	4.698026	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	798	HTTP/1.1 200 OK (PNG)
526	4.698334	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	440	GET /upload/tpl/02/c8/712/template712/js/brccheck.js HTTP/1.1
528	4.708158	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1204	HTTP/1.1 200 OK (text/css)
530	4.708469	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	436	GET /upload/tpl/02/c8/712/template712/js/lib.js HTTP/1.1
602	4.793896	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1294	HTTP/1.1 200 OK (application/javascript)
605	4.793897	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	939	HTTP/1.1 200 OK (application/javascript)
625	4.797459	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	452	GET /upload/tpl/02/c8/712/template712/js/jquery.lazyload.min.js HTTP/1.1
626	4.798136	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	442	GET /upload/tpl/02/c8/712/template712/js/slick.min.js HTTP/1.1
646	4.814610	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	773	HTTP/1.1 200 OK (text/css)
659	4.814996	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	442	GET /upload/tpl/02/c8/712/template712/css/iconfont.js HTTP/1.1
686	4.874398	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	828	HTTP/1.1 200 OK (application/javascript)
696	4.874677	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	439	GET /upload/tpl/02/c8/712/template712/js/common.js HTTP/1.1

根据俘获窗口内容，思考以下问题：

- 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 HTTP 协议的版本号是多少？



由上截图内容可知，我的浏览器运行的是HTTP1.1，访问的服务器运行的是HTTP1.1。

- 你的浏览器向服务器指出它能接收何种语言版本的对象？

Accept-Language: zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3\r\n

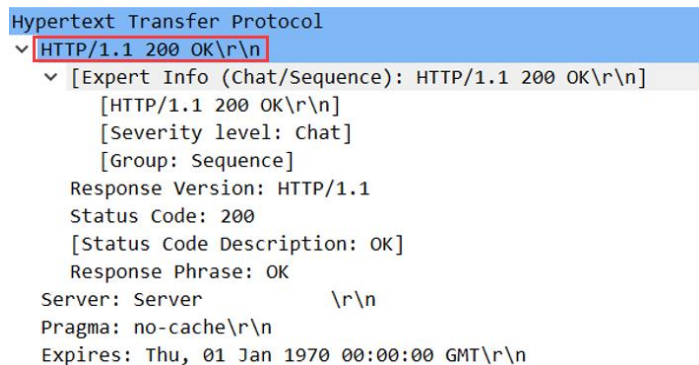
由上截图内容可知，简体中文，美国英语，通用英语

- 你的计算机的 IP 地址是多少？服务器 <http://jwes.hit.edu.cn/> 的 IP 地址是多少？

1604 48.185660	219.217.226.139	172.20.113.24	HTTP	176 HTTP/1.1 200 OK (text/html)
1606 48.199918	172.20.113.24	219.217.226.139	HTTP	479 GET /resources/js/jquery/jquery-1.7.2.min.js HTTP/1.1
1607 48.201031	172.20.113.24	219.217.226.139	HTTP	451 GET /resources/css/common/style1.css HTTP/1.1

由以上截图内容可知，本机IP为172.20.113.24，服务器IP为219.217.226.139

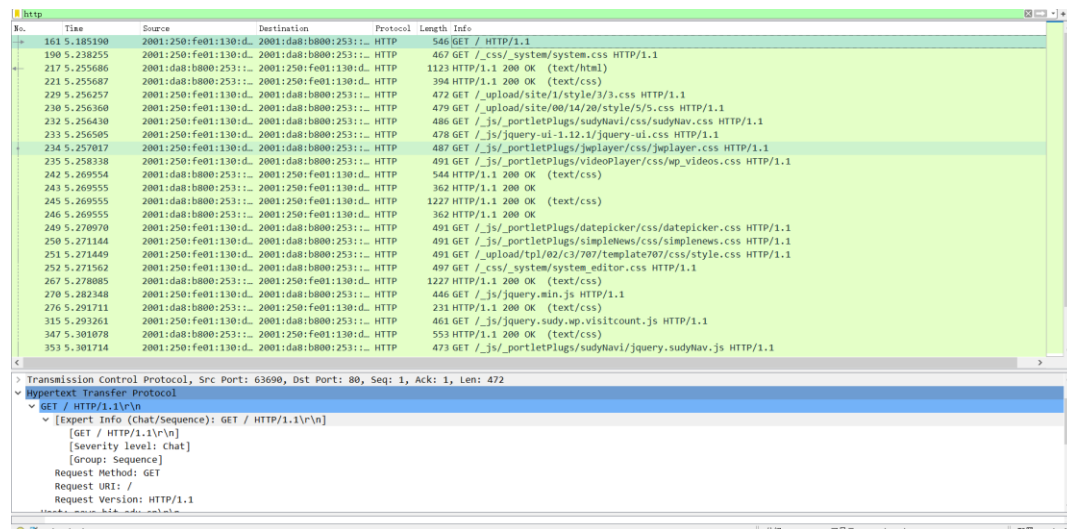
- 从服务器向你的浏览器返回的状态代码是多少？



由图中内容可知，返回的状态码为200

## 2.2 HTTP 条件 GET/response 交互

清除浏览器缓存后，重新开始捕获，并将过滤器设置为HTTP，通过浏览器<http://news.hit.edu.cn/>，待加载完成后再次访问该网址，得到报文。



- 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?

```

    Hypertext Transfer Protocol
    GET /_js/_portletPlugs/sudyNavi/css/sudyNav.css HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /_js/_portletPlugs/sudyNavi/css/sudyNav.css HTTP/1.1\r\n]
    [GET /_js/_portletPlugs/sudyNavi/css/sudyNav.css HTTP/1.1\r\n]
    [Severity level: Chat]
    [Group: Sequence]
    Request Method: GET
    Request URI: /_js/_portletPlugs/sudyNavi/css/sudyNav.css
    Request Version: HTTP/1.1
    Accept: text/css, */*\r\n
    Referer: http://news.hit.edu.cn/\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; rv:11.0) like Gecko\r\n
    Accept-Encoding: gzip, deflate\r\n
    Host: news.hit.edu.cn\r\n
    Connection: Keep-Alive\r\n
    Cookie: JSESSIONID=A8A36B386E7A565A4F03ED2372B25417\r\n
    Cookie pair: JSESSIONID=A8A36B386E7A565A4F03ED2372B25417
  
```

由以上截图中观察报文内容可知，报文中并没有 IF-MODIFIED-SINCE

- 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？

```

GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, image/jxr, */*
Accept-Language: zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: news.hit.edu.cn
Connection: Keep-Alive
Cookie: JSESSIONID=A8A36B386E7A565A4F03ED2372B25417

HTTP/1.1 200 OK
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Server:
Date: Fri, 26 Nov 2021 17:33:18 GMT
Accept-Ranges: bytes
Vary: Accept-Encoding
X-Frame-Options: ALLOW-FROM http://homepage.hit.edu.cn/

<!DOCTYPE html>
<html lang="zh-CN">

<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1" />
  <meta name="renderer" content="webkit" />
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-scalable=no" />
  <meta name="format-detection" content="telephone=no" />
  <title>.....</title>
  <meta name="description" content="....." />
  <meta name="keywords" content="....." />
  <meta name="keywords" content="....." />
  <meta name="description" content="....." />

  <link type="text/css" href="/_css/_system/system.css" rel="stylesheet"/>
  <link type="text/css" href="/_upload/site/1/style/3/3.css" rel="stylesheet"/>
  <link type="text/css" href="/_upload/site/00/14/20/style/5/5.css" rel="stylesheet"/>
  <link type="text/css" href="/_js/_portletPlugs/sudyNavi/css/sudyNav.css" rel="stylesheet" />
  
```

68	11.951213	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	429	GET / HTTP/1.1
84	11.978251	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	470	GET /_js/_portletPlugs/sudyNavi/css/sudyNav.css HTTP/1.1
109	11.993018	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	544	HTTP/1.1 200 OK (text/css)
112	11.994179	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	462	GET /_js/jquery-ui-1.12.1/jquery-ui.css HTTP/1.1
130	12.019482	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	471	GET /_js/_portletPlugs/jwplayer/css/jwplayer.css HTTP/1.1
131	12.019592	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	475	GET /_js/_portletPlugs/videoPlayer/css/wp_videos.css HTTP/1.1
136	12.029957	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	475	GET /_js/_portletPlugs/datepicker/css/datepicker.css HTTP/1.1
137	12.029972	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	458	GET /_css/_system/system.editor.css HTTP/1.1
143	12.031028	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	1132	HTTP/1.1 200 OK (text/html)
147	12.031616	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	475	GET /_js/_portletPlugs/simpleNews/css/simpleNews.css HTTP/1.1
180	12.070284	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	1227	HTTP/1.1 200 OK (text/css)
182	12.070476	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	475	GET /_upload/tp1/02/c3/707/template207/css/style.css HTTP/1.1
187	12.076747	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	1227	HTTP/1.1 200 OK (text/css)
192	12.076748	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	231	HTTP/1.1 200 OK (text/css)
195	12.078502	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	475	GET /_upload/tp1/02/c3/707/template207/css/slick.css HTTP/1.1
196	12.078762	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	465	GET /_js/jquery.min.js HTTP/1.1
221	12.086245	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	553	HTTP/1.1 200 OK (text/css)
224	12.087604	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	480	GET /_js/jquery.sudy.up.visitcount.js HTTP/1.1
231	12.089205	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	150	HTTP/1.1 200 OK (text/css)
235	12.096373	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	402	GET /_js/_portletPlugs/sudyNavi/jquery.sudyNav.js HTTP/1.1
263	12.146192	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	233	HTTP/1.1 200 OK (text/css)
270	12.146594	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	485	GET /_js/jquery-ui-1.12.1/jquery-ui.min.js HTTP/1.1
298	12.161571	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	171	HTTP/1.1 200 OK (text/css)
302	12.161737	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	497	GET /_js/_portletPlugs/videoPlayer/player/swfobject.js HTTP/1.1
309	12.169889	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	680	HTTP/1.1 200 OK (application/javascript)
318	12.170143	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	486	GET /_js/_portletPlugs/jwplayer/jwplayer.js HTTP/1.1
330	12.180063	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	145	HTTP/1.1 200 OK (text/css)
332	12.181162	2001:250:f001:130:d::	2001:dab:b000:253::	HTTP	489	GET /_js/_portletPlugs/jwplayer/jwplayerIE8.js HTTP/1.1
350	12.188131	2001:dab:b000:253::	2001:250:f001:130:d::	HTTP	1367	HTTP/1.1 200 OK (application/javascript)

追踪该HTTP流可知服务器返回的所有json文件的状态码均为200，因此服务器明确返回了所有内容。

- 分析你的浏览器向服务器发出的较晚的“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？



No.	Time	Source	Destination	Protocol	Length	Info
183	5.580163	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	533	GET / HTTP/1.1
150	5.649061	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	1123	HTTP/1.1 200 OK (text/html)
196	8.584071	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	626	GET /upload/article/images/6c/6e/cdc9af542beb5aeb88735dec72/dbb000e7-3bd3-44f0-b712-61f5cfd6a6df.jpg HTTP/1.1
197	8.593474	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	333	HTTP/1.1 304 Not Modified
198	8.609120	2001:250:fe01:130:d...	2001:da8:b800:253::...	HTTP	627	GET /upload/article/images/3c/30/f1a328604b34ab858687cb3a7723/9b7811aa-fd53-4f4a-86a5-eb5a73c7b5b6_s.jpg HTTP/1.1
199	8.620242	2001:da8:b800:253::...	2001:250:fe01:130:d...	HTTP	332	HTTP/1.1 304 Not Modified

Frame 196: 626 bytes on wire (5008 bits), 626 bytes captured (5008 bits) on interface \Device\NPF_{FF2AAE3-A7E7-4B04-B057-C02E23988208}, id 0 Ethernet II, Src: IntelCor_f5:85:c0 (dc:71:96:f5:85:c0), Dst: JuniperW_d2:ff:c2 (44:ec:ce:d2:ff:c2) Internet Protocol Version 6, Src: 2001:250:fe01:130:d146:8002:9095:903f, Dst: 2001:da8:b800:253::c0a8:3209 Transmission Control Protocol, Src Port: 60533, Dst Port: 80, Seq: 460, Ack: 57210, Len: 552						
Hypertext Transfer Protocol						
GET /upload/article/images/6c/6e/cdc9af542beb5aeb88735dec72/dbb000e7-3bd3-44f0-b712-61f5cfd6a6df.jpg HTTP/1.1\r\n						
[Expert Info (Chat/Sequence): GET /upload/article/images/6c/6e/cdc9af542beb5aeb88735dec72/dbb000e7-3bd3-44f0-b712-61f5cfd6a6df.jpg HTTP/1.1\r\n]						
[GET /upload/article/images/6c/6e/cdc9af542beb5aeb88735dec72/dbb000e7-3bd3-44f0-b712-61f5cfd6a6df.jpg HTTP/1.1\r\n]						
[Severity level: Chat]						
[Group: Sequence]						
Request Method: GET						
Request URI: /upload/article/images/6c/6e/cdc9af542beb5aeb88735dec72/dbb000e7-3bd3-44f0-b712-61f5cfd6a6df.jpg						
Request Version: HTTP/1.1						
Host: news.hit.edu.cn\r\n						
Connection: keep-alive\r\n						
If-Modified-Since: Fri, 26 Nov 2021 08:10:07 GMT\r\n						
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36\r\n						
If-None-Match: "f355-5d1ac9e7b15c0"\r\n						
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n						
Referer: http://news.hit.edu.cn/\r\n						
Accept-encoding: gzip, deflate\r\n						
Accept-Language: zh-CN,zh;q=0.9\r\n						
\r\n						

IF-MODIFIED-SINCE位置如图所示，后面跟着的是当前缓存最后一次更新的时间。

- 服务器对较晚的 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。

Hypertext Transfer Protocol	
HTTP/1.1 304 Not Modified\r\n	
[Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]	
[HTTP/1.1 304 Not Modified\r\n]	
[Severity level: Chat]	
[Group: Sequence]	
Response Version: HTTP/1.1	
Status Code: 304	
[Status Code Description: Not Modified]	
Response Phrase: Not Modified	
Connection: keep-alive\r\n	
Server: \r\n	
Date: Fri, 26 Nov 2021 17:33:34 GMT\r\n	
Last-Modified: Thu, 07 Sep 2017 01:14:32 GMT\r\n	
ETag: "1e-5588f2ec84e00"\r\n	
Accept-Ranges: bytes\r\n	
X-Frame-Options: ALLOW-FROM http://homepage.hit.edu.cn/\r\n	
\r\n	
[HTTP response 7/35]	
[Time since request: 0.016141000 seconds]	
[Prev request in frame: 2459]	
[Request in frame: 37821]	
[Next request in frame: 37859]	
[Next response in frame: 37870]	
[Request URI: http://news.hit.edu.cn/_js/jquery.min.js]	

状态码为304，不会明确返回文件内容，因为服务器经过比对发现本地缓存文件最后更新时间与服务器的文件最后更新时间一致，因此会认为No Modified，表示本地的缓存未过期。

### 3. TCP分析

通过向 gaia.cs.umass.edu 发送文件可以俘获大量的TCP分组

No.	Time	Source	Destination	Protocol	Length	Info
74	4.982227	2001:da8:b800:11:f6:2001:250:fe01:130:d...	2001:da8:b800:11:f6:2001:250:fe01:130:d...	TCP	78	443 → 4080 [SYN, ACK] Seq=0 Ack=1 Win=0 MSS=1432 Len=0
75	4.132269	20.42.73.26	172.20.113.24	TLSv1.2	538	Application Data
76	4.132370	172.20.113.24	20.42.73.26	TCP	54	59143 → 443 [ACK] Seq=4521 Ack=5423 Win=261120 Len=0
77	4.225003	2001:da8:b800:11:f6:2001:250:fe01:130:d...	2001:da8:b800:11:f6:2001:250:fe01:130:d...	TCP	78	443 → 4080 [SYN, ACK] Seq=0 Ack=1 Win=0 MSS=1432 Len=0
78	4.373598	60.28.172.14	172.20.113.24	TCP	56	80 → 59116 [FIN, ACK] Seq=1 Ack=1 Win=19 Len=0
79	4.373708	172.20.113.24	60.28.172.14	TCP	54	59116 → 80 [ACK] Seq=1 Ack=2 Win=32407 Len=0
80	4.377050	60.28.172.14	172.20.113.24	TCP	56	80 → 59115 [FIN, ACK] Seq=1 Ack=1 Win=22 Len=0
81	4.377105	172.20.113.24	60.28.172.14	TCP	54	59115 → 80 [ACK] Seq=1 Ack=2 Win=32652 Len=0
82	4.377797	60.28.172.14	172.20.113.24	TCP	56	80 → 59117 [FIN, ACK] Seq=1 Ack=1 Win=19 Len=0
83	4.377876	172.20.113.24	60.28.172.14	TCP	54	59117 → 80 [ACK] Seq=1 Ack=2 Win=32131 Len=0
84	4.384793	172.20.113.24	111.13.34.176	TLSv1.2	239	Application Data
85	4.385113	172.20.113.24	111.13.34.176	TLSv1.2	401	Application Data
86	4.417373	111.13.34.176	172.20.113.24	TCP	56	443 → 59103 [ACK] Seq=312 Ack=1065 Win=331 Len=0
87	4.422357	111.13.34.176	172.20.113.24	TLSv1.2	365	Application Data
88	4.472698	172.20.113.24	111.13.34.176	TCP	54	59103 → 443 [ACK] Seq=1065 Ack=623 Win=511 Len=0
89	5.547241	172.20.113.24	128.119.245.12	TCP	74	59145 → 80 [SYN] Seq=0 Win=0 MSS=1460 WS=256 SACK_PERM=1 Tsv=44225862 TSecr=0
90	5.547241	172.20.113.24	128.119.245.12	TCP	74	59145 → 80 [SYN] Seq=0 Win=0 MSS=1460 WS=256 SACK_PERM=1 Tsv=44225862 TSecr=0
91	5.626979	128.119.245.12	172.20.113.24	TCP	74	80 → 59145 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1360 SACK_PERM=1 Tsv=1937017182 TSecr=44225862 WS=1
92	5.827008	172.20.113.24	128.119.245.12	TCP	66	59145 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0 Tsv=44226144 TSecr=1937017182
93	5.827353	172.20.113.24	128.119.245.12	TCP	537	59145 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=471 Tsv=44226144 TSecr=1937017182 [TCP segment of a reas
94	5.828946	128.119.245.12	172.20.113.24	TCP	74	80 → 59144 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1360 SACK_PERM=1 Tsv=1937017182 TSecr=44225862 WS=1
95	5.829052	172.20.113.24	128.119.245.12	TCP	66	59144 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0 Tsv=44226144 TSecr=1937017182
96	5.829703	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=472 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass
97	5.829800	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=1820 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass
98	5.829811	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=3168 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass
99	5.829821	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=4516 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass
100	5.829831	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=5864 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass
101	5.829840	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=7212 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass
102	5.829850	172.20.113.24	128.119.245.12	TCP	1414	59145 → 80 [ACK] Seq=8560 Ack=1 Win=262144 Len=1348 Tsv=44226144 TSecr=1937017182 [TCP segment of a reass

### 3.1 浏览追踪信息

- 向 gaia.cs.umass.edu 服务器传送文件的客户端主机的 IP 地址和TCP 端口号是多少？

```
> Internet Protocol Version 4, Src: 172.20.113.24, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 59145, Dst Port: 80, Seq: 0, Len: 0
```

客户端主机的IP为172.20.113.24, TCP端口号为59145

- gaia.cs.umass.edu 服务器的 IP 地址是多少？对这一连接，它用来发送和接收 TCP 报文的端口号是多少？

服务器IP地址为128.119.245.12, TCP端口号为80

### 3.2 TCP基础

- 客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号（sequence number）是多少？在该报文段中，是用什么来标示该报文段是 SYN 报文段的？

```
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 3050453715
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1010 .... = Header Length: 40 bytes (10)
▼ Flags: 0x002 (SYN)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... 0... = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...0 = Acknowledgment: Not set
  .... .... 0... = Push: Not set
  .... .... ..0. = Reset: Not set
  ▼ .... .... ..1. = Syn: Set
    ▼ [Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]
      [Connection establish request (SYN): server port 80]
      [Severity level: Chat]
      [Group: Sequence]
      .... .... ...0 = Fin: Not set
```

初始化TCP连接的TCP SYN报文段的序号是0；通过Flags标志位来标示该报文段是SYN报文段的

- 服务器向客户端发送的 SYNACK 报文段序号是多少？该报文段中，Acknowledgement 字段的值是多少？Gaia.cs.umass.edu 服务器是如何决定此值的？在该报文段中，是用什么来标示该报文段是SYNACK 报文段的？

```
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 1932071241
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3329663186
1010 .... = Header Length: 40 bytes (10)
Flags: 0x012 (SYN, ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... 0... = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 .... = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... .... ..0. = Reset: Not set
  > .... .... ..1. = Syn: Set
  .... .... ...0 = Fin: Not set
[TCP Flags: .....A..S.]
```

服务器向客户端发送的SYNACK报文段序号是1, Acknowledgment字段的值是1, 服务器根据用户上一次发送的报文中的seq+1得到Acknowledgment的值, 通过Flags中Syn和Acknowledgment位的值为1来确定该报文段是SYN ACK报文段

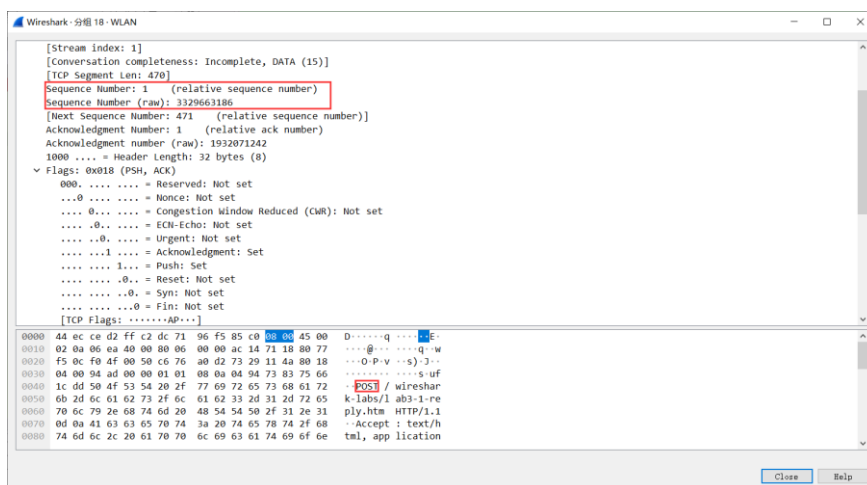
- 你能从捕获的数据包中分析出 tcp 三次握手过程吗？

10 0.754046	172.20.113.24	128.119.245.12	TCP	74 61519 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=76837456 TSecr=0
16 1.060043	128.119.245.12	172.20.113.24	TCP	74 80 → 61519 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1360 SACK_PERM=1 TSval=1969626333 TSecr=76837456 WS=1
17 1.060089	172.20.113.24	128.119.245.12	TCP	66 61519 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0 TSval=76837762 TSecr=1969626333

客户端先向服务器发送 seq=0 的建立连接的请求；然后服务器向客户端返回 seq=0, ack=1 的响应；最后客户端向服务器返回 seq=1, ack=1 的确认报文。

- 包含 HTTP POST 命令的 TCP 报文段的序号是多少？

1.754046	172.20.113.24	128.119.245.12	TCP	74 61519 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=76837456 TSecr=0
..060043	128.119.245.12	172.20.113.24	TCP	74 80 → 61519 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1360 SACK_PERM=1 TSval=1969626333 TSecr=76837456 WS=1
..060089	172.20.113.24	128.119.245.12	TCP	66 61519 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0 TSval=76837762 TSecr=1969626333
..060696	172.20.113.24	128.119.245.12	TCP	536 61519 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=470 TSval=76837763 TSecr=1969626333
..062779	172.20.113.24	128.119.245.12	TCP	1414 61519 → 80 [ACK] Seq=471 Ack=1 Win=262144 Len=1348 TSval=76837765 TSecr=1969626333
..062787	172.20.113.24	128.119.245.12	TCP	1414 61519 → 80 [ACK] Seq=1819 Ack=1 Win=262144 Len=1348 TSval=76837765 TSecr=1969626333
..062790	172.20.113.24	128.119.245.12	TCP	1414 61519 → 80 [ACK] Seq=3167 Ack=1 Win=262144 Len=1348 TSval=76837765 TSecr=1969626333
..062817	172.20.113.24	128.119.245.12	TCP	1414 61519 → 80 [ACK] Seq=4515 Ack=1 Win=262144 Len=1348 TSval=76837765 TSecr=1969626333



序列号为1，绝对序号为3329663186

- 如果将包含 HTTP POST 命令的 TCP 报文段看作是 TCP 连接上的第一个报文段，那么该 TCP 连接上的第六个报文段的序号是多少？是何时发送的？该报文段所对应的 ACK 是何时接收的？

172.20.113.24	TCP	74 80 → 63674 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=76837456 TSecr=0
128.119.245.12	TCP	66 63673 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	66 63674 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	669 63674 → 80 [PSH, ACK] Seq=1 Ack=1 Win=132096 Len=603 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=604 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=1952 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=3300 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=4648 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=5996 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=7344 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=8692 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=10040 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
128.119.245.12	TCP	1414 63674 → 80 [ACK] Seq=11388 Ack=1 Win=132096 Len=1348 TSval=89 TSecr=76837456 WS=1
172.20.113.24	TCP	66 443 → 63650 [ACK] Seq=1 Ack=2 Win=253 Len=0 TSval=19819 TSecr=76837456 WS=1
172.20.113.24	TCP	66 80 → 63649 [ACK] Seq=1 Ack=2 Win=235 Len=0 TSval=19819 TSecr=76837456 WS=1
13.67.9.5	TCP	55 63618 → 443 [ACK] Seq=1 Ack=1 Win=511 Len=1 [TCP segment of data length 0 bytes] TSval=19819 TSecr=76837456 WS=1
2:.. 2001:250:fe01:130:4::	TCP	78 443 → 38934 [SYN, ACK] Seq=0 Ack=1 Win=65534 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=249 TSecr=76837456 WS=1
172.20.113.24	TCP	74 80 → 63675 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=249 TSecr=76837456 WS=1
128.119.245.12	TCP	66 63675 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0 TSval=89 TSecr=76837456 WS=1
172.20.113.24	TCP	66 80 → 63674 [ACK] Seq=1 Ack=604 Win=30208 Len=0 TSval=19819 TSecr=76837456 WS=1
172.20.113.24	TCP	66 [TCP Dup ACK 46#1] 80 → 63674 [ACK] Seq=1 Ack=604 Win=30208 Len=0 TSval=19819 TSecr=76837456 WS=1
172.20.113.24	TCP	66 80 → 63674 [ACK] Seq=1 Ack=4648 Win=38272 Len=0 TSval=19819 TSecr=76837456 WS=1
172.20.113.24	TCP	66 80 → 63674 [ACK] Seq=1 Ack=5996 Win=41215 Len=0 TSval=19819 TSecr=76837456 WS=1

Frame 34: 1414 bytes on wire (11312 bits), 1414 bytes captured (11312 bits) on interface 0 (Device\NPF\_{FF2AAEC3-A7E7-4B04-B057-C02E23988208})  
 Encapsulation type: Ethernet (1)  
 Arrival Time: Nov 27, 2021 14:17:04.634001000 中国标准时间  
 [Time shift for this packet: 0.000000000 seconds]  
 Epoch Time: 1637993824.634001000 seconds  
 [Time delta from previous captured frame: 0.000050000 seconds]  
 [Time delta from previous displayed frame: 0.000050000 seconds]  
 [Time since reference or first frame: 0.315307000 seconds]  
 Error Number: 0



```

Frame 49: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on in
> Interface id: 0 (\Device\NPF_{FF2AAEC3-A7E7-4B04-B057-C02E23988208})
  Encapsulation type: Ethernet (1)
  Arrival Time: Nov 27, 2021 14:17:04.948230000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1637993824.948230000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.629536000 seconds]
    
```

报文序号是5996，发送时间是 Nov 27, 2021 14:17:04.634001000 中国标准时间，对应ACK的接收时间为 Nov 27, 2021 14:17:04.948230000 中国标准时间。

- 前六个 TCP 报文段的长度各是多少？

```

TCP      669 63674 → 80 [PSH, ACK] Seq=1 Ack=1 Win=132096 Len=603 TSval=8918288
TCP      1414 63674 → 80 [ACK] Seq=604 Ack=1 Win=132096 Len=1348 TSval=89182883
TCP      1414 63674 → 80 [ACK] Seq=1952 Ack=1 Win=132096 Len=1348 TSval=89182883
TCP      1414 63674 → 80 [ACK] Seq=3300 Ack=1 Win=132096 Len=1348 TSval=89182883
TCP      1414 63674 → 80 [ACK] Seq=4648 Ack=1 Win=132096 Len=1348 TSval=89182883
TCP      1414 63674 → 80 [ACK] Seq=5996 Ack=1 Win=132096 Len=1348 TSval=89182883
TCP      1414 63674 → 80 [ACK] Seq=7344 Ack=1 Win=132096 Len=1348 TSval=89182883
TCP      1414 63674 → 80 [ACK] Seq=8692 Ack=1 Win=132096 Len=1348 TSval=89182883
    
```

前六个TCP报文段长度分别为603，1348，1348，1348，1348，1348

- 在整个跟踪过程中，接收端公示的最小的可用缓存空间是多少？限制发送端的传输以后，接收端的缓存是否仍然不够用？

```

TCP      74 80 → 63673 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=13
TCP      74 80 → 63674 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=13
TCP      66 63673 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0 TSval=8918
TCP      66 63674 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0 TSval=8918
TCP      669 63674 → 80 [PSH, ACK] Seq=1 Ack=1 Win=132096 Len=603 Tsv
TCP      1414 63674 → 80 [ACK] Seq=604 Ack=1 Win=132096 Len=1348 TSva
TCP      1414 63674 → 80 [ACK] Seq=1952 Ack=1 Win=132096 Len=1348 TSva
TCP      1414 63674 → 80 [ACK] Seq=3300 Ack=1 Win=132096 Len=1348 TSva
TCP      1414 63674 → 80 [ACK] Seq=4648 Ack=1 Win=132096 Len=1348 TSva
TCP      1414 63674 → 80 [ACK] Seq=5996 Ack=1 Win=132096 Len=1348 TSva
TCP      1414 63674 → 80 [ACK] Seq=7344 Ack=1 Win=132096 Len=1348 TSva
TCP      1414 63674 → 80 [ACK] Seq=8692 Ack=1 Win=132096 Len=1348 TSva
    
```

最小可用缓存空间为132096，发送端的传输以后接收端的缓存够用。

- 在跟踪文件中是否有重传的报文段？进行判断的依据是什么？  
没有重传的报文段，客户端发送的序列号一直递增。
- TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。

```

[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
> Ethernet II, Src: IntelCor_f5:85:c0 (dc:71:96:f5:85:c0), Dst: JuniperN_d2:ff:c2 (44:ec:ce:d2:f
> Internet Protocol Version 4, Src: 172.20.113.24, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 63674, Dst Port: 80, Seq: 151896, Ack: 1, Len: 1029
v [115 Reassembled TCP Segments (152924 bytes) #29(603), #30(1348), #31(1348), #32(1348), #33(1
  [Frame: 29, payload: 0-602 (603 bytes)]
  [Frame: 30, payload: 603-1950 (1348 bytes)]
  [Frame: 31, payload: 1951-3298 (1348 bytes)]
  [Frame: 32, payload: 3299-4646 (1348 bytes)]
  [Frame: 33, payload: 4647-5994 (1348 bytes)]
  [Frame: 34, payload: 5995-7342 (1348 bytes)]
  [Frame: 35, payload: 7343-8690 (1348 bytes)]
    
```

▼ [Timestamps]

```

[Time since first frame in this TCP stream: 1.299152000 seconds]
[Time since previous frame in this TCP stream: 0.000002000 seconds]
    
```

发送数据的大小一共为152924bytes，时间为1.229152000-0.000002000=1.229150000s，

#### 4. IP分组

```

> Frame 1: 78 bytes on wire (624 bits). 78 bytes captured (624 bits) on interface \Device\NPF {FF2AAEC3-A7E7-4B04-B057-C02E23988208}. id 0

```

#### 4.1 对捕获的数据包进行分析

```

Name: 14: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF{...}
Ethernet II, Src: IntelCor_f5:85:c0 (dc:71:96:f5:85:c0), Dst: JuniperN_d2:ff:c2 (44:ec:ce:f5:85:c0)
Internet Protocol Version 4, Src: 172.20.128.87, Dst: 219.217.226.139
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 56
Identification: 0x1574 (5492)
Flags: 0x00

```

```
Time to Live: 255
Protocol: ICMP (1)
Header Checksum: 0x0000
```

- IP头有多少字节？该IP数据包的净载为多少字节？并解释你是怎样确定该IP数据包的净载大小的？

```
Internet Protocol Version 4, Src: 172.20.128.87, Dst: 219.217.226.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 56
  Identification: 0x1574 (5492)
> Flags: 0x00
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 255
  Protocol: ICMP (1)
```

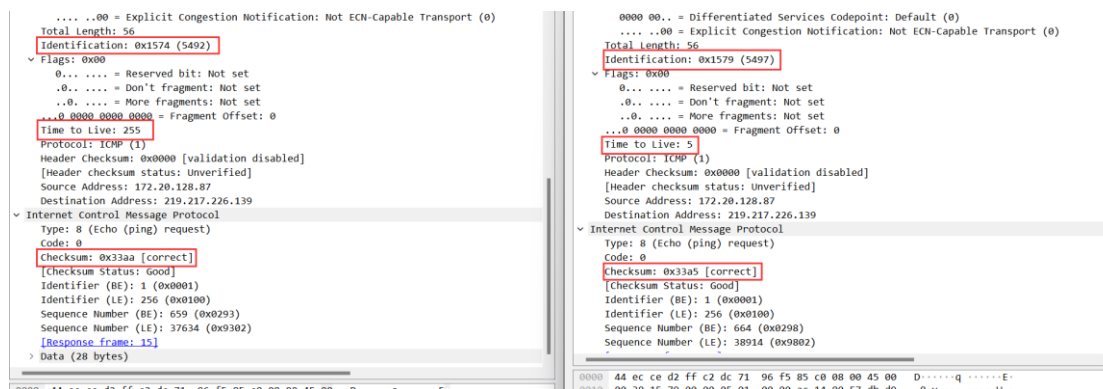
- 该IP数据包分片了吗？解释你是如何确定该P数据包是否进行了分片

```

    ... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 56
    Identification: 0x1574 (5492)
    Flags: 0x00
    0... ..00 = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 255
    
```

查看more fragments位可知其位数为0，且偏移量为0，故当前数据包并未分片。

- 你主机发出的一系列ICMP消息中IP数据报中哪些字段总是发生改变？



Identification、TTL和Checksum总在发生变化

- 哪些字段必须保持常量？哪些字段必须改变？为什么？

必须保持常量的是版本号、首部长、Differentiated Services Field 以及协议（始终为ICMP）。必须改变的是 TTL、Checksum 和 Identification，TTL 为生存时间，每次转发 必然改变；由于 TTL 的改变，Checksum 自然也会改变；Identification 则是用于区分不同的 ICMP 报文。

- 描述你看到的IP数据包Identification字段值的形式。

四位16进制，每个包的Identification每次加一。

- Identification字段和TTL字段的值是什么？

```

    Total Length: 56
    Identification: 0x0000 (0)
    Flags: 0x00
    0... ..00 = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 254
    Protocol: ICMP (1)
    Header Checksum: 0x7daf [validation disabled]
    [Header checksum status: Unverified]
    
```

Identification是0，TTL字段为254。

- 最近的路由器（第一跳）返回给你主机的ICMP Time-to-live exceeded消息中这些值是否保持不变？为什么？

不变，因为是第一跳路由器发回的报文，TTL不变；IP是无连接服务，标识不是序列号，相同的标识是为了分段后重组，给同一个主机发送的ICMP报文，TTL不变，则Identification字段不变。

- 该消息是否被分解成不止一个IP数据报？

被分成了两个IP数据包。

- 观察第一个IP分片，IP头部的哪些信息表明数据包被进行了分片？IP头部的哪些信息表明数据包是第一个而不是最后一个分片？该分片的长度是多少？

```
.... ..00 = Explicit Congestion Notification: Not set
Total Length: 1500
Identification: 0x1593 (5523)
Flags: 0x20, More fragments
 0... .. = Reserved bit: Not set
 .0.. .. = Don't fragment: Not set
 ..1. .... = More fragments: Set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 255
Protocol: ICMP (1)
Header Checksum: 0x0000 [validation disabled]
```

More fragments位值为1，表明当前信息分片，且当前分片不是最后一块，该分片长度为1500。

- 原始数据包被分成了多少片？

```
[Header checksum status: Unverified]
Source Address: 172.20.128.87
Destination Address: 219.217.226.139
> [3 IPv4 Fragments (3480 bytes): #323(1480), #324(1480), #325(520)]
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x8b33 [validation disabled]
```

被分成了三片。

- 这些分片中IP数据报头部哪些字段发生了变化？

<pre>Flags: 0x20, More fragments  0... .. = Reserved bit: Not set  .0.. .. = Don't fragment: Not set  ..1. .... = More fragments: Set ...0 0000 0000 0000 = Fragment Offset: 0 Time to Live: 60 Protocol: ICMP (1) Header Checksum: 0x8a33 [validation disabled]</pre>	<pre>Flags: 0x20, More fragments  0... .. = Reserved bit: Not set  .0.. .. = Don't fragment: Not set  ..1. .... = More fragments: Set ...0 0101 1100 1000 = Fragment Offset: 1480 Time to Live: 60 Protocol: ICMP (1) Header Checksum: 0x897a [validation disabled]</pre>	<pre>Flags: 0x01  0... .. = Reserved bit: Not set  .0.. .. = Don't fragment: Not set  ..0. .... = More fragments: Not set ...0 1011 1001 0000 = Fragment Offset: 2960 Time to Live: 60 Protocol: ICMP (1) Header Checksum: 0xac81 [validation disabled] [Header checksum status: Unverified]</pre>
--	---	--

标志位部分、偏移量和 Checksum 部分发生了变化

## 5. 抓取 ARP 数据包

查看arp缓存内容：

```
C:\Users\Youngsc>arp -a

接口: 172.20.1.6 --- 0x18
Internet 地址      物理地址          类型
172.20.0.1         44-ec-ce-d2-ff-c2 动态
172.20.23.75       44-ec-ce-d2-ff-c2 动态
172.20.28.217      44-ec-ce-d2-ff-c2 动态
172.20.29.137      44-ec-ce-d2-ff-c2 动态
172.20.41.68       44-ec-ce-d2-ff-c2 动态
172.20.41.102      44-ec-ce-d2-ff-c2 动态
172.20.41.165      44-ec-ce-d2-ff-c2 动态
172.20.51.69       44-ec-ce-d2-ff-c2 动态
172.20.52.196      44-ec-ce-d2-ff-c2 动态
172.20.63.206      44-ec-ce-d2-ff-c2 动态
172.20.66.144      44-ec-ce-d2-ff-c2 动态
172.20.83.239      44-ec-ce-d2-ff-c2 动态
172.20.98.111      44-ec-ce-d2-ff-c2 动态
172.20.118.162     44-ec-ce-d2-ff-c2 动态
172.20.132.195     44-ec-ce-d2-ff-c2 动态
172.20.204.188     44-ec-ce-d2-ff-c2 动态
172.20.229.134     44-ec-ce-d2-ff-c2 动态
172.20.250.97      44-ec-ce-d2-ff-c2 动态
172.20.255.255     ff-ff-ff-ff-ff-ff 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
224.0.0.252        01-00-5e-00-00-fc 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态
```



在命令行模式下输入：ping 172.17.42.105

```
C:\Users\Youngsc>ping 172.17.42.105

正在 Ping 172.17.42.105 具有 32 字节的数据:
来自 172.17.42.105 的回复: 字节=32 时间=54ms TTL=61
来自 172.17.42.105 的回复: 字节=32 时间=7ms TTL=61
来自 172.17.42.105 的回复: 字节=32 时间=10ms TTL=61
来自 172.17.42.105 的回复: 字节=32 时间=13ms TTL=61

172.17.42.105 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 7ms, 最长 = 54ms, 平均 = 21ms
```

启动wireshark进行捕获

No.	Time	Source	Destination	Protocol	Length	Info
51	5.450546	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.83.239? Tell 172.20.1.6
52	5.461857	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	56	172.20.83.239 is at 44:ec:ce:d2:ff:c2
56	5.954663	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.0.1? Tell 172.20.1.6
57	6.071490	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	60	172.20.0.1 is at 44:ec:ce:d2:ff:c2
148	23.455028	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.51.69? Tell 172.20.1.6
149	23.471000	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	56	172.20.51.69 is at 44:ec:ce:d2:ff:c2
191	36.463616	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.132.195? Tell 172.20.1.6
192	36.478412	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	56	172.20.132.195 is at 44:ec:ce:d2:ff:c2
211	41.950501	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.83.239? Tell 172.20.1.6
212	42.050744	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	56	172.20.83.239 is at 44:ec:ce:d2:ff:c2
283	46.959327	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.0.1? Tell 172.20.1.6
285	47.037909	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	60	172.20.0.1 is at 44:ec:ce:d2:ff:c2
445	63.456519	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.229.134? Tell 172.20.1.6
446	63.470034	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	56	172.20.229.134 is at 44:ec:ce:d2:ff:c2
455	69.453950	IntelCor_f5:85:c0	JuniperN_d2:ff:c2	ARP	42	Who has 172.20.51.69? Tell 172.20.1.6
456	69.469557	JuniperN_d2:ff:c2	IntelCor_f5:85:c0	ARP	56	172.20.51.69 is at 44:ec:ce:d2:ff:c2

5.1

ARP表的格式如下。在ARP表中，每一项表示一个IP地址到物理地址的映射。每一项第一列是IP地址，第二列是物理地址，第三列是类型。

```
C:\Users\Youngsc>arp -a

接口: 172.20.1.6 --- 0x18
Internet 地址          物理地址              类型
172.20.0.1             44-ec-ce-d2-ff-c2     动态
172.20.23.75           44-ec-ce-d2-ff-c2     动态
172.20.28.217          44-ec-ce-d2-ff-c2     动态
172.20.29.137          44-ec-ce-d2-ff-c2     动态
172.20.41.68           44-ec-ce-d2-ff-c2     动态
172.20.41.102          44-ec-ce-d2-ff-c2     动态
172.20.41.165          44-ec-ce-d2-ff-c2     动态
172.20.51.69           44-ec-ce-d2-ff-c2     动态
172.20.52.196          44-ec-ce-d2-ff-c2     动态
172.20.63.206          44-ec-ce-d2-ff-c2     动态
172.20.66.144          44-ec-ce-d2-ff-c2     动态
172.20.83.239          44-ec-ce-d2-ff-c2     动态
172.20.98.111          44-ec-ce-d2-ff-c2     动态
172.20.118.162         44-ec-ce-d2-ff-c2     动态
172.20.132.195         44-ec-ce-d2-ff-c2     动态
172.20.204.188         44-ec-ce-d2-ff-c2     动态
172.20.229.134         44-ec-ce-d2-ff-c2     动态
172.20.250.97          44-ec-ce-d2-ff-c2     动态
172.20.255.255         ff-ff-ff-ff-ff-ff     静态
224.0.0.22             01-00-5e-00-00-16     静态
224.0.0.251            01-00-5e-00-00-fb     静态
224.0.0.252            01-00-5e-00-00-fc     静态
239.255.255.250        01-00-5e-7f-ff-fa     静态
255.255.255.255        ff-ff-ff-ff-ff-ff     静态
```

5.2

- ARP数据包的格式是怎样的？由几部分构成，各个部分所占的字节数是多少？



格式如上图所示，共由九部分组成。硬件类型（2 字节），协议类型（2 字节），硬件地址长度（1 字节），协议地址长度（1 字节），OP（2 字节），发送端 MAC 地址（6 字节），发送端 IP 地址（4 字节），目的 MAC 地址（6 字节），目的 IP 地址（4字节）

- 如何判断一个ARP数据是请求包还是应答包？

可以通过 Opcode 字段判断，若为 1 则是请求包；若为 2 则是应答包。

<pre> Type: ARP (0x0806) Address Resolution Protocol (request) Hardware type: Ethernet (1) Protocol type: IPv4 (0x0800) Hardware size: 6 Protocol size: 4 Opcode: request (1) Sender MAC address: IntelCor_f5:85:c0 (dc:71: Sender IP address: 172.20.1.6 Target MAC address: JuniperN_d2:ff:c2 (44:ec: Target IP address: 172.20.83.239         </pre>	<pre> Type: ARP (0x0806) Trailer: 00000000000000000000000000000000 Address Resolution Protocol (reply) Hardware type: Ethernet (1) Protocol type: IPv4 (0x0800) Hardware size: 6 Protocol size: 4 Opcode: reply (2) Sender MAC address: JuniperN_d2:ff:c2 (44:ec:ce:d2:ff:c2) Sender IP address: 172.20.83.239 Target MAC address: IntelCor_f5:85:c0 (dc:71:96:f5:85:c0) Target IP address: 172.20.1.6         </pre>
---	---

- 为什么ARP查询要在广播帧中传送，而ARP响应要在一个有着明确目的局域网地址的帧中传送？

因为进行 ARP 查询时并不知道目的 IP 地址对应的 MAC 地址，所以需要广播查询；而 ARP 响应报文知道查询主机的 MAC 地址（通过查询主机发出的查询报文获得），且局域网中的其他主机不需要此次查询的结果，因此 ARP 响应要在一个有着明确目的局域网地址的帧中传送。

## 6. 抓取UDP数据包

- 消息是基于UDP的还是TCP的？

UDP

- 你的主机ip地址是什么？目的主机ip地址是什么？

```

> Source: IntelCor_f5:85:c0 (dc:71:96:f5:85:c0)
Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: 2001:250:fe01:130:83:34a5:1f76:8aa0, Dst: 2402:4e00:1830:1039:0:9084:c0ed:4e85
0110 .... = Version: 6
> .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
.... 1010 0001 1000 0011 1110 = Flow Label: 0xa183e
Payload Length: 47
Next Header: UDP (17)
Hop Limit: 64
Source Address: 2001:250:fe01:130:83:34a5:1f76:8aa0
Destination Address: 2402:4e00:1830:1039:0:9084:c0ed:4e85
    
```

我的主机IP为2001:250:fe01:130:83:34a5:1f76:8aa0，目的主机IP为2402:4e00:1830:1039:0:9084:c0ed:4e85

- 你的主机发送QQ消息的端口号和QQ服务器的端口号分别是多少？

```

Limit: 64
Source Address: 2001:250:fe01:130:83:34a5:1f76:8aa0
Destination Address: 2402:4e00:1830:1039:0:9084:c0ed:4e85
Telegram Protocol, Src Port: 56575, Dst Port: 8001
Source Port: 56575
Destination Port: 8001
Length: 47
Checksum: 0x3b65 [unverified]
Checksum Status: Unverified
    
```

主机的QQ消息端口号为56575，服务器的端口号为8001

- 数据包的格式是什么样的？都包含哪些字段，分别占多少字节？

```

    User Datagram Protocol, Src Port: 56575, Dst Port: 8001
    Source Port: 56575
    Destination Port: 8001
    Length: 47
    Checksum: 0x3b65 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    > [Timestamps]
    UDP payload (39 bytes)
  
```

UDP 数据报由五部分构成，分别是源端口号（4 字节），目的端口号（4 字节），长度（4 字节），校验和（4 字节）和应用层数据。

- 为什么你发送一个ICQ数据包后，服务器又返回给你的主机一个ICQ数据包？这UDP的不可靠数据传输有什么联系？对比前面的TCP协议分析，你能看出UDP是无连接的吗？

因为 UDP 是不可靠的数据传输，需要上层协议来实现可靠数据传输，因此每次发送 ICQ 报文后又回复一个 ICQ 数据包来确认。UDP 是无连接的，因为可以看到发送数据之前没有连接的建立过程（如 TCP 的三次握手），没有序列号，因此为无连接数据传输。

## 7. 利用 Wireshark 进行 DNS 协议分析

利用 Wireshark 进行 DNS 协议抓包的结果如下。

No.	Time	Source	Destination	Protocol	Length	Info
13	1.143150	172.20.1.6	223.5.5.5	DNS	76	Standard query 0x65bf A pss.bdstatic.com
14	1.143155	172.20.1.6	223.5.5.5	DNS	73	Standard query 0x9863 A www.baidu.com
21	1.183840	172.20.1.6	223.6.6.6	DNS	73	Standard query 0x9863 A www.baidu.com
22	1.183840	172.20.1.6	223.6.6.6	DNS	76	Standard query 0x65bf A pss.bdstatic.com
24	1.212714	223.5.5.5	172.20.1.6	DNS	132	Standard query response 0x9863 A www.baidu.com CNAME www.a.shifen.com A 39.156.66.18 A 39.156.66.14
25	1.213028	172.20.1.6	223.5.5.5	DNS	73	Standard query 0xd5a7 AAAA www.baidu.com
33	1.221372	223.6.6.6	172.20.1.6	DNS	132	Standard query response 0x9863 A www.baidu.com CNAME www.a.shifen.com A 39.156.66.18 A 39.156.66.14
34	1.224073	223.6.6.6	172.20.1.6	DNS	314	Standard query response 0x65bf A pss.bdstatic.com CNAME pss.bdstatic.com A bdydns.com CNAME opendinglobalv6
35	1.224187	172.20.1.6	223.6.6.6	DNS	76	Standard query 0x58e5 AAAA pss.bdstatic.com
38	1.248801	223.5.5.5	172.20.1.6	DNS	157	Standard query response 0xd5a7 AAAA www.baidu.com CNAME www.a.shifen.com SDA ns1.a.shifen.com
44	1.250866	172.20.1.6	223.5.5.5	DNS	82	Standard query 0xf297 A hectorstatic.baidu.com
46	1.250311	172.20.1.6	223.5.5.5	DNS	73	Standard query 0x38b2 A sp1.baidu.com
48	1.254774	223.6.6.6	172.20.1.6	DNS	182	Standard query response 0x58e5 AAAA pss.bdstatic.com CNAME pss.bdstatic.com A bdydns.com CNAME opendinglobalv6
50	1.256286	172.20.1.6	223.5.5.5	DNS	76	Standard query 0x0ba8 A ssl.bdstatic.com
54	1.280260	223.5.5.5	172.20.1.6	DNS	314	Standard query response 0x65bf A pss.bdstatic.com CNAME pss.bdstatic.com A bdydns.com CNAME opendinglobalv6
57	1.280261	223.5.5.5	172.20.1.6	DNS	176	Standard query response 0xf297 A hectorstatic.baidu.com CNAME hectorstatic.baidu.com A bdydns.com CNAME ope
60	1.280587	172.20.1.6	223.5.5.5	DNS	82	Standard query 0x2248 AAAA hectorstatic.baidu.com
63	1.291325	172.20.1.6	223.6.6.6	DNS	73	Standard query 0x38b2 A sp1.baidu.com
64	1.291326	172.20.1.6	223.6.6.6	DNS	76	Standard query 0x0ba8 A ssl.bdstatic.com
67	1.294355	223.5.5.5	172.20.1.6	DNS	132	Standard query response 0x38b2 A sp1.baidu.com CNAME www.a.shifen.com A 39.156.66.18 A 39.156.66.14
69	1.294355	223.5.5.5	172.20.1.6	DNS	126	Standard query response 0x0ba8 A ssl.bdstatic.com CNAME ssl.bdstatic.com A 111.40.186.32
73	1.294625	172.20.1.6	223.5.5.5	DNS	73	Standard query 0xb755 AAAA sp1.baidu.com
74	1.294626	172.20.1.6	223.5.5.5	DNS	76	Standard query 0xc0d7 AAAA ssl.bdstatic.com
89	1.326201	223.5.5.5	172.20.1.6	DNS	188	Standard query response 0x2248 AAAA hectorstatic.baidu.com CNAME hectorstatic.baidu.com A bdydns.com CNAME

### 问题讨论：

问题1：用Edge、谷歌浏览器多次访问各种网址，但报文中始终没有出现If-Modified-Since字段，最终将浏览器更换为IE之后才成功。

问题2：实验指导书中给出的网址，访问后使用wireshark分析后发现ipv6地址，造成难以按照实验步骤进行分析。解决方案：更改访问网站为IPv4网站（http://jwes.hit.edu.cn/或http://today.hit.edu.cn/等）

### 心得体会：

- 1、本次实验学会了如何使用Wireshark进行抓包。
- 2、本次实验对于各个协议之间进行报文格式和报文交换有了进一步的了解。