



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	瞿久尧		院系	计算学部		
班级	2037101		学号	120L022314		
任课教师	李全龙		指导教师	李全龙		
实验地点	格物 207		实验时间	2022.10.28		
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						



计算机科学与技术学院 SINCE 1956...  
School of Computer Science and Technology

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：
<ul style="list-style-type: none"><li>● 学习 Wireshark 的使用</li><li>● 利用 Wireshark 分析 HTTP 协议</li><li>● 利用 Wireshark 分析 TCP 协议</li><li>● 利用 Wireshark 分析 IP 协议</li><li>● 利用 Wireshark 分析 Ethernet 数据帧</li></ul> 选做内容： <ul style="list-style-type: none"><li>● 利用 Wireshark 分析 DNS 协议</li><li>● 利用 Wireshark 分析 UDP 协议</li><li>● 利用 Wireshark 分析 ARP 协议</li></ul>
实验过程：
<div><h3>1. wireshark 的使用</h3></div>



```
> Frame 114: 549 bytes on wire (4392 bits), 549 bytes captured (4392 bits) on interface \Device\NPF{...}
> Ethernet II, Src: IntelCor_ef:fe:61 (c8:b2:9b:ef:fe:61), Dst: JuniperN_d2:ff:c2 (44:ec:ce:c2)
> Internet Protocol Version 4, Src: 172.20.157.186, Dst: 61.167.60.70
> Transmission Control Protocol, Src Port: 65415, Dst Port: 80, Seq: 1, Ack: 1, Len: 495
< Hypertext Transfer Protocol
  < GET / HTTP/1.1\r\n
    < [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
      Host: www.hit.edu.cn\r\n
      Connection: keep-alive\r\n
      Cache-Control: max-age=0\r\n
      Upgrade-Insecure-Requests: 1\r\n
```

- 你的浏览器向服务器指出它能接收何种语言版本的对象？  
中文

```
Host: www.hit.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/80.0.4012.91 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
\r\n
```

- 你的计算机的 IP 地址是多少？服务器 http://www.hit.edu.cn 的 IP 地址是多少？  
我的计算机：172.20.157.186  
服务器：61.167.60.70
- 从服务器向你的浏览器返回的状态代码是多少？  
200ok

240	7.580869	61.167.60.70	172.20.157.186	HTTP	1255	HTTP/1.1 200 OK (text/html)
246	7.589927	61.167.60.70	172.20.157.186	HTTP	917	HTTP/1.1 200 OK (text/css)

2) HTTP 条件 GET/response 交互

- 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？  
否

```

▼ Hypertext Transfer Protocol
  ▼ GET / HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: www.hit.edu.cn\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
    > Cookie: JSESSIONID=96B402819563598DACF5F71719B0E04A\r\n
    \r\n
    [Full request URI: http://www.hit.edu.cn/]
    [HTTP request 1/2]
    [Response in frame: 240]
    [Next request in frame: 524]

```

- 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？  
是，由content-length可知。

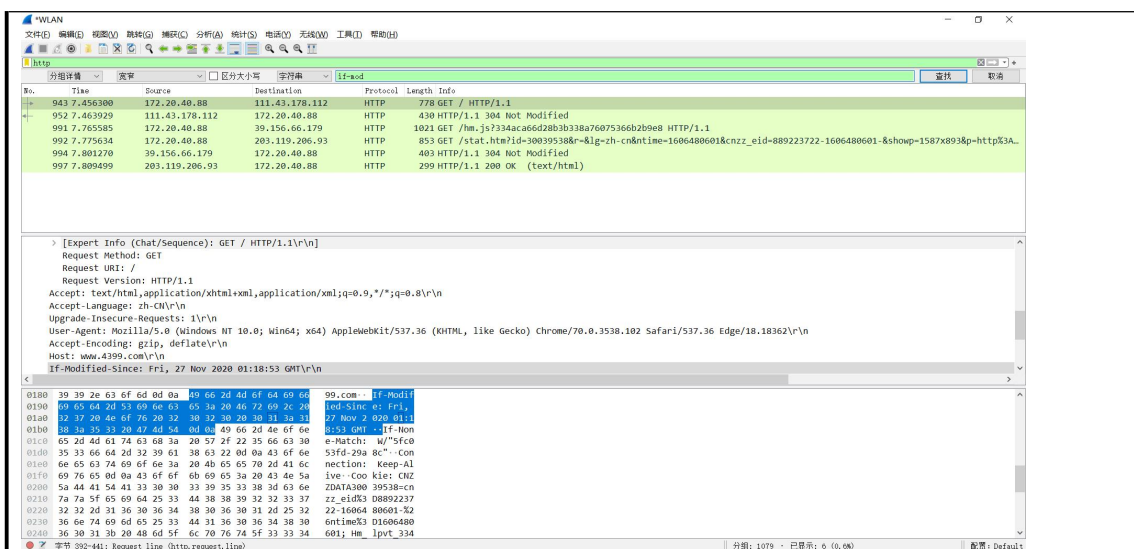
No.	Time	Source	Destination	Protocol	Length	Info
76	7.284670	172.20.157.186	61.167.60.70	HTTP	576	GET / HTTP/1.1
111	7.376944	172.20.157.186	61.167.60.70	HTTP	536	GET /_upload/tpl/02/df/735/template735/css/style.css?v=1666916522648 HTTP/1.1
240	7.588869	61.167.60.70	172.20.157.186	HTTP	1255	HTTP/1.1 200 OK (text/html)
246	7.589927	61.167.60.70	172.20.157.186	HTTP	917	HTTP/1.1 200 OK (text/css)
252	7.743847	2001:250:fe01:130:b...	2408:4002:1f10:141	HTTP	649	GET /xlibid=1&aid=1022&id=934&peerid=002867C878F93C8Q&userid=&referfrom=100001805-win&osversion=10.0.190448...
253	7.778016	172.20.157.186	61.167.60.70	HTTP	581	GET /_upload/article/videos/27/27/3a987ffb4e29b14490ab4ed22a1b/f103e716-e055-42ed-8010-882e2ec23fe8-B.mp4 HT...
256	7.783367	2001:250:fe01:130:b...	2408:4002:1f10:141	HTTP	658	GET /xlibid=1&aid=1022&id=916&peerid=002867C878F93C8Q&userid=&referfrom=100001805-win&osversion=10.0.190448...
268	7.818436	2408:4002:1f10:141	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)
285	7.857163	2408:4002:1f10:141	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)
524	8.461525	172.20.157.186	61.167.60.70	HTTP	589	GET /_upload/article/videos/27/27/3a987ffb4e29b14490ab4ed22a1b/f103e716-e055-42ed-8010-882e2ec23fe8-B.mp4 HT...
600	8.536672	61.167.60.70	172.20.157.186	HTTP	213	HTTP/1.1 206 Partial Content (video/mp4)
654	8.643252	2001:250:fe01:130:b...	2408:4002:1f10:141	HTTP	679	GET /xlibid=1&aid=1022&id=934&peerid=002867C878F93C8Q&userid=&referfrom=100001805-win&osversion=10.0.190448...
656	8.723068	2408:4002:1f10:141	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)
769	29.527662	172.20.157.186	39.156.66.18	HTTP	214	HEAD /robots.txt HTTP/1.1
771	30.500582	39.156.66.18	172.20.157.186	HTTP	151	HTTP/1.1 200 OK

> Transmission Control Protocol, Src Port: 80, Dst Port: 49164, Seq: 58481, Ack: 483, Len: 8<^	0000 c8 b2 9b ef fe 61 44 ec ce d2 ff c2 08 00 45 00 .....aD: .....E.
> [44 Reassembled TCP Segments (59343 bytes): #141(1360), #142(1360), #143(1360), #146(1360).	0010 03 87 7e 8f 40 00 7c 06 b9 25 3d a7 3c 46 0016 ...g. . .%=<cf
▼ Hypertext Transfer Protocol	0020 00 50 c0 0c c8 50 df 90 a1 8a be 0a 50 18 ...P.-p .....P.
> HTTP/1.1 200 OK\r\n	0030 1e 9c 93 54 00 00 7a 3a 31 35 70 78 3b 66 6f 6e ...T:t: 15px;fon
> [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]	0040 74 2d 73 69 7a 65 3a 31 34 70 78 7d 0d 0a 2e 6d t-size:1 4px;...m
Response Version: HTTP/1.1	0050 2d 4a 6f 75 72 6e 61 6c 69 73 6d 6c 20 2e 6d 6f -Journal isml .mo
Status Code: 200	0060 72 65 20 61 7b 66 6f 6e 74 2d 73 69 7a 65 3a 31 re a(fon t-size:1
[Status Code Description: OK]	0070 32 70 78 7d 0d 0a 2e 6d 2d 70 72 6f 6d 69 73 69 zpx). .m -promisi
Response Phrase: OK	0080 6e 67 6c 20 2e 70 69 63 7b 77 69 64 74 68 3a 32 ngl .pic [width:2
Content-Type: text/css\r\n	0090 30 30 70 78 3b 68 65 69 67 68 74 3a 33 30 30 70 00px;hei ght:300p
> Content-Length: 59025\r\n	00a0 78 3b 6d 61 72 67 69 6e 3a 36 30 70 78 20 61 75 x;margin :60px au
[Content length: 59025]	00b0 74 6f 20 32 30 70 78 7d 0d 0a 2e 6d 2d 6e 65 77 to 20px). .m-new
Connection: keep-alive\r\n	00c0 73 20 2e 70 69 63 7b 77 69 64 74 68 3a 36 30 25 s .pic[width:600
Server: \r\n	00d0 3b 20 68 65 69 67 68 74 3a 61 75 74 6f 3b 7d 0d ; height :auto;}
Date: Fri, 28 Oct 2022 00:17:18 GMT\r\n	00e0 0a 2e 6d 2d 6e 65 77 73 20 2e 74 69 74 7b 66 6f .m-news .tit(fo
Last-Modified: Wed, 26 Oct 2022 12:19:39 GMT\r\n	00f0 6e 74 2d 73 69 7a 65 3a 20 31 34 70 78 3b 20 6c nt-size: 14px; l
ETag: "e691-5ebef0e0eccc"\r\n	0100 69 6e 65 2d 68 65 69 67 68 74 3a 20 31 38 70 78 ine-heig ht: 18px
Accept-Ranges: bytes\r\n	0110 3b 20 6d 61 78 2d 68 65 69 67 68 74 3a 20 6e 6f ; max-he ight: no
Vary: Accept-Encoding\r\n	0120 6e 65 3b 7d 0d 0a 2e 6d 2d 6e 65 77 73 20 2e 64 ne);...m -news .s
	0130 65 74 7b 7d 0d 0a 2e 6d 2d 6e 65 77 73 20 2e 73 et();...m -news .s
	0140 77 69 70 65 72 2d 70 61 67 69 6e 61 74 69 6f 6e winner-na pination

- 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？  
否，但如果有，跟的信息应该是最近一次修改时间。





- 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。

304not modified, 服务器没有明确返回文件的内容, 但是给出了ETag, 浏览器可以在此缓存中查找文件, 其值与第一次返回的报文的 ETag 一致。

### 3. TCP分析

A. 俘获大量的由本地主机到远程服务器的 TCP 分组

B. 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？

IP 地址: 172.20.157.186

端口号: 50448

128.119.245.12	172.20.157.186	TCP	56 80 → 50448 [ACK]
128.119.245.12	172.20.157.186	TCP	56 80 → 50448 [ACK]
128.119.245.12	172.20.157.186	TCP	56 80 → 50448 [ACK]

- 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？

IP 地址: 128.119.245.12

端口号: 80

172.20.157.186	128.119.245.12	TCP	1414 50448 → 80 [ACK]
172.20.157.186	128.119.245.12	TCP	1414 50448 → 80 [ACK]
172.20.157.186	128.119.245.12	TCP	1414 50448 → 80 [ACK]

C. TCP 基础

- 客户服务器之间用于初始化 TCP 连接的 TCPSYN 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？

初始 seq 为 0

通过设置 tcp 头部 flags 字段的 syn 标志位来标示该报文段是 SYN 报文段。

Sequence Number: 0 (relative sequence number)

```

v Flags: 0x002 (SYN)
  000. .... = Reserved: Not set
  ...0 .... = Accurate ECN: Not set
  .... 0... = Congestion Window Reduced: 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
  .... .... ...0 = Fin: Not set
  [TCP Flags: .....S.]
  
```

- 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？  
序列号：0

Acknowledgement 字段：1

服务器收到客户端发来的 syn 报文，此报文消耗一个序列号（0），因此服务器回复期望收到的下一个序列号的值为 1

通过设置 tcp 头部 flags 字段的 syn 标志位来标示该报文段是 SYNACK 报文段

```

Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 134351343
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3693856048
1000 .... = Header Length: 32 bytes (8)
v Flags: 0x012 (SYN, ACK)
  000. .... = Reserved: Not set
  ...0 .... = Accurate ECN: Not set
  .... 0... = Congestion Window Reduced: 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.]
  
```

- 你能从捕获的数据包中分析出 tcp 三次握手过程吗？

172.20.157.186 128.119.245.12 TCP 66 50448 → 80 [SYN] Seq=0

128.119.245.12 172.20.157.186 TCP 66 80 → 50448 [SYN, ACK] Seq=0

172.20.157.186 128.119.245.12 TCP 54 50448 → 80 [ACK] Seq=1

- 包含 HTTP POST 命令的 TCP 报文段的序号是多少？  
152741

No.	Time	Source	Destination	Protocol	Length	Info
57	4.164331	2001:250:fe01:130:b...	2008:4002:1f10::41	HTTP	658	GET /?xlbId=1&aid=1022&id=937&peerid=002B67C878F93CBQ&userid=&referrerFrom=100001805-win&OSversion=10.0.19044&pr...
100	4.240322	2008:4002:1f10::41	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)
6445	6.967515	2001:250:fe01:130:b...	2008:4002:1f10::41	HTTP	721	GET /?xlbId=1&aid=1022&id=934&peerid=002B67C878F93CBQ&userid=&referrerFrom=100001805-win&OSversion=10.0.19044&pr...
6456	7.057795	2008:4002:1f10::41	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)
6464	7.097209	172.20.157.186	128.119.245.12	HTTP	511	GET /favicon.ico HTTP/1.1
6542	7.398858	128.119.245.12	172.20.157.186	HTTP	539	HTTP/1.1 404 Not Found (text/html)
6890	12.900877	172.20.157.186	128.119.245.12	HTTP	839	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
6964	13.210595	128.119.245.12	172.20.157.186	HTTP	830	HTTP/1.1 200 OK (text/html)
6970	13.393097	2001:250:fe01:130:b...	2008:4002:1f10::41	HTTP	713	GET /?xlbId=1&aid=1022&id=934&peerid=002B67C878F93CBQ&userid=&referrerFrom=100001805-win&OSversion=10.0.19044&pr...
6975	13.467272	2008:4002:1f10::41	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)

> Frame 6890: 839 bytes on wire (6712 bits), 839 bytes captured (6712 bits) on interface \Device\NPF{...}	0020 f5 0c 5 10 00 50 ca b5 2a 30 d7 f8 4d 34 50 18	.....P...-MAP-
> Ethernet II, Src: IntelCor_ef:fe:61 (c8:b2:9b:ef:fe:61), Dst: JuniperM_d2:ff:c2 (44:ec:c6:ce:)	0030 02 01 c2 7e 00 00 75 72 20 6f 66 20 74 68 65 20	.....up of the
> Internet Protocol Version 4, Src: 172.20.157.186, Dst: 128.119.245.12	0040 62 75 73 79 20 66 61 72 6d 2d 79 61 72 64 2d 20	.....busy fan m-yard--
> Transmission Control Protocol, Src Port: 50448, Dst Port: 80, Seq: 152741, Ack: 486, Len: 839	0050 77 68 69 6c 65 20 74 68 65 20 6c 6f 77 69 6e 67	.....while the e lowing
Source Port: 50448	0060 20 6f 66 20 74 68 65 0d 0a 63 61 74 74 6c 65 20	.....of the cattle
Destination Port: 80	0070 69 6e 20 74 68 65 20 64 69 73 74 61 6e 63 65 20	.....in the d istance
[Stream index: 52]	0080 77 6f 75 6c 64 20 74 61 6b 65 20 74 68 65 20 70	.....would ta ke the p
[Conversation completeness: Incomplete, DATA (15)]	0090 6c 61 63 65 20 6f 66 20 74 68 65 20 4d 6f 63 6b	.....lace of the hock
[TCP Segment Len: 785]	00a0 20 54 75 72 74 6c 65 27 73 0d 0a 68 65 61 76 79	.....Turtle! s'-heavy
Sequence Number: 152741 (relative sequence number)	00b0 20 73 6f 62 73 2e 0d 0a 0d 0a 20 20 4c 61 73 74	.....sobs... Last
Next Sequence Number: 153226 (relative sequence number)	00c0 6c 79 2c 20 73 68 65 20 70 69 63 74 75 72 65 64	.....ly, she pictured
Acknowledgment Number: 486 (relative ack number)	00d0 20 74 6f 20 68 65 72 73 65 6c 66 20 68 6f 77 20	.....to hers elf how
Acknowledgment number (raw): 3623374132	00e0 74 68 69 73 20 73 61 6d 65 20 6c 69 74 74 6c 65	.....this sam e little
0101 .... = Header Length: 20 bytes (5)	00f0 20 73 69 73 74 65 72 20 6f 66 0d 0a 68 65 72 73	.....sister of--hers
	0100 20 77 6f 75 6c 64 2c 20 69 6e 20 74 68 65 20 61	.....would, in the a
	0110 66 74 6f 72 2d 74 69 6d 65 2c 20 62 65 20 68 65	.....fter-tim e, be he
	0120 73 73 65 6c 66 20 61 20 67 73 6f 77 6a 20 72 6f	.....realiz e a dream wa

- 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？  
序号：6187  
在 http post 发送之前发送  
对应的ack是服务器发送的第6个ack

190	2.045053	172.20.157.186	128.119.245.12	HTTP	903	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
233	2.529524	128.119.245.12	172.20.157.186	HTTP	831	HTTP/1.1 200 OK (text/html)
259	2.970857	2001:250:fe01:130:b...	2008:4002:1f10::41	HTTP	649	GET /?xlbId=1&aid=1022&id=928&peerid=002B67C878F93CBQ&userid=&referrerFrom=100001805-win&OSversion=10.0.19044&pr...
260	2.970915	2001:250:fe01:130:b...	2008:4002:1f10::41	HTTP	650	GET /?xlbId=1&aid=1022&id=916&peerid=002B67C878F93CBQ&userid=&referrerFrom=100001805-win&OSversion=10.0.19044&pr...
262	3.051256	2008:4002:1f10::41	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)
264	3.051256	2008:4002:1f10::41	2001:250:fe01:130:b...	HTTP	288	HTTP/1.1 200 OK (GIF89a)

..... .0. = Syn: Not set	00001820 6e 6c 79 2c 20 74 68 65 6d 70 21 20 74 68 75 6d	nly, thu mpl thum
..... .0. = Fin: Not set	00001840 20 75 70 6f 6e 20 61 20 68 65 61 70 20 6f 6e 6d	pit down she came
[TCP flags: .....AP...]	00001860 0a 73 74 69 63 6b 73 20 61 6e 64 20 64 72 79 20	upon a heap of
Window: 515	00001880 6c 65 61 76 65 73 2c 20 61 6e 64 20 74 68 65 20	sticks and dry
[calculated window size: 515]	000018a0 66 61 6e 6c 20 77 61 73 20 6f 76 65 72 2e 0d 0a	leaves, and the
[window size scaling factor: -1 (unknown)]	000018c0 74 20 61 20 62 69 74 20 68 75 72 74 2c 20 61 6e	fall was over...
Checksum: 0xc2be [unverified]	000018e0 64 20 61 20 62 69 74 20 68 75 72 74 2c 20 61 6e	Alic e was no
[checksum status: unverified]	00001900 64 20 73 68 65 20 6a 75 6d 70 65 64 20 75 70 20	t a hit hurt, an
Urgent Pointer: 0	00001920 6f 6e 20 74 6f 20 68 65 72 20 66 65 65 74 20 69	d she jum ped up
> [Timestamps]	00001940 6e 20 61 0d 0a 6d 6f 6d 65 6e 74 3a 20 20 73 68	on to he r feet i
> [Seq/Ack analysis]	00001960 65 20 6c 6f 6b 65 64 20 75 70 2c 20 62 65 74	in a mom ent: sh
TCP payload (849 bytes)	00001980 20 69 74 20 77 61 73 20 61 6c 6c 20 64 61 72 6b	e looked up, but
TCP segment data (849 bytes)	000019a0 20 6f 76 65 72 68 65 61 64 3b 20 62 65 66 6f 72	it was all dark
> [14 Reassembled TCP segments (153068 bytes): #14(747), #15(1360), #16(1360), #17(1360), #18(1360), #19(1360), #20(1360), #21(1360), #22(1360), #23(1360), #24(1360), #25(1360), #26(1360), #27(1360), #28(1360), #29(1360), #30(1360), #31(1360), #32(1360), #33(1360), #34(1360), #35(1360), #36(1360), #37(1360), #38(1360), #39(1360), #40(1360), #41(1360), #42(1360), #43(1360), #44(1360), #45(1360), #46(1360), #47(1360), #48(1360), #49(1360), #50(1360), #51(1360), #52(1360), #53(1360), #54(1360), #55(1360), #56(1360), #57(1360), #58(1360), #59(1360), #60(1360), #61(1360), #62(1360), #63(1360), #64(1360), #65(1360), #66(1360), #67(1360), #68(1360), #69(1360), #70(1360), #71(1360), #72(1360), #73(1360), #74(1360), #75(1360), #76(1360), #77(1360), #78(1360), #79(1360), #80(1360), #81(1360), #82(1360), #83(1360), #84(1360), #85(1360), #86(1360), #87(1360), #88(1360), #89(1360), #90(1360), #91(1360), #92(1360), #93(1360), #94(1360), #95(1360), #96(1360), #97(1360), #98(1360), #99(1360), #100(1360), #101(1360), #102(1360), #103(1360), #104(1360), #105(1360), #106(1360), #107(1360), #108(1360), #109(1360), #110(1360), #111(1360), #112(1360), #113(1360), #114(1360), #115(1360), #116(1360), #117(1360), #118(1360), #119(1360), #120(1360), #121(1360), #122(1360), #123(1360), #124(1360), #125(1360), #126(1360), #127(1360), #128(1360), #129(1360), #130(1360), #131(1360), #132(1360), #133(1360), #134(1360), #135(1360), #136(1360), #137(1360), #138(1360), #139(1360), #140(1360), #141(1360), #142(1360), #143(1360), #144(1360), #145(1360), #146(1360), #147(1360), #148(1360), #149(1360), #150(1360), #151(1360), #152(1360), #153(1360), #154(1360), #155(1360), #156(1360), #157(1360), #158(1360), #159(1360), #160(1360), #161(1360), #162(1360), #163(1360), #164(1360), #165(1360), #166(1360), #167(1360), #168(1360), #169(1360), #170(1360), #171(1360), #172(1360), #173(1360), #174(1360), #175(1360), #176(1360), #177(1360), #178(1360), #179(1360), #180(1360), #181(1360), #182(1360), #183(1360), #184(1360), #185(1360), #186(1360), #187(1360), #188(1360), #189(1360), #190(1360), #191(1360), #192(1360), #193(1360), #194(1360), #195(1360), #196(1360), #197(1360), #198(1360), #199(1360), #200(1360), #201(1360), #202(1360), #203(1360), #204(1360), #205(1360), #206(1360), #207(1360), #208(1360), #209(1360), #210(1360), #211(1360), #212(1360), #213(1360), #214(1360), #215(1360), #216(1360), #217(1360), #218(1360), #219(1360), #220(1360), #221(1360), #222(1360), #223(1360), #224(1360), #225(1360), #226(1360), #227(1360), #228(1360), #229(1360), #230(1360), #231(1360), #232(1360), #233(1360), #234(1360), #235(1360), #236(1360), #237(1360), #238(1360), #239(1360), #240(1360), #241(1360), #242(1360), #243(1360), #244(1360), #245(1360), #246(1360), #247(1360), #248(1360), #249(1360), #250(1360), #251(1360), #252(1360), #253(1360), #254(1360), #255(1360), #256(1360), #257(1360), #258(1360), #259(1360), #260(1360), #261(1360), #262(1360), #263(1360), #264(1360), #265(1360), #266(1360), #267(1360), #268(1360), #269(1360), #270(1360), #271(1360), #272(1360), #273(1360), #274(1360), #275(1360), #276(1360), #277(1360), #278(1360), #279(1360), #280(1360), #281(1360), #282(1360), #283(1360), #284(1360), #285(1360), #286(1360), #287(1360), #288(1360), #289(1360), #290(1360), #291(1360), #292(1360), #293(1360), #294(1360), #295(1360), #296(1360), #297(1360), #298(1360), #299(1360), #300(1360), #301(1360), #302(1360), #303(1360), #304(1360), #305(1360), #306(1360), #307(1360), #308(1360), #309(1360), #310(1360), #311(1360), #312(1360), #313(1360), #314(1360), #315(1360), #316(1360), #317(1360), #318(1360), #319(1360), #320(1360), #321(1360), #322(1360), #323(1360), #324(1360), #325(1360), #326(1360), #327(1360), #328(1360), #329(1360), #330(1360), #331(1360), #332(1360), #333(1360), #334(1360), #335(1360), #336(1360), #337(1360), #338(1360), #339(1360), #340(1360), #341(1360), #342(1360), #343(1360), #344(1360), #345(1360), #346(1360), #347(1360), #348(1360), #349(1360), #350(1360), #351(1360), #352(1360), #353(1360), #354(1360), #355(1360), #356(1360), #357(1360), #358(1360), #359(1360), #360(1360), #361(1360), #362(1360), #363(1360), #364(1360), #365(1360), #366(1360), #367(1360), #368(1360), #369(1360), #370(1360), #371(1360), #372(1360), #373(1360), #374(1360), #375(1360), #376(1360), #377(1360), #378(1360), #379(1360), #380(1360), #381(1360), #382(1360), #383(1360), #384(1360), #385(1360), #386(1360), #387(1360), #388(1360), #389(1360), #390(1360), #391(1360), #392(1360), #393(1360), #394(1360), #395(1360), #396(1360), #397(1360), #398(1360), #399(1360), #400(1360), #401(1360), #402(1360), #403(1360), #404(1360), #405(1360), #406(1360), #407(1360), #408(1360), #409(1360), #410(1360), #411(1360), #412(1360), #413(1360), #414(1360), #415(1360), #416(1360), #417(1360), #418(1360), #419(1360), #420(1360), #421(1360), #422(1360), #423(1360), #424(1360), #425(1360), #426(1360), #427(1360), #428(1360), #429(1360), #430(1360), #431(1360), #432(1360), #433(1360), #434(1360), #435(1360), #436(1360), #437(1360), #438(1360), #439(1360), #440(1360), #441(1360), #442(1360), #443(1360), #444(1360), #445(1360), #446(1360), #447(1360), #448(1360), #449(1360), #450(1360), #451(1360), #452(1360), #453(1360), #454(1360), #455(1360), #456(1360), #457(1360), #458(1360), #459(1360), #460(1360), #461(1360), #462(1360), #463(1360), #464(1360), #465(1360), #466(1360), #467(1360), #468(1360), #469(1360), #470(1360), #471(1360), #472(1360), #473(1360), #474(1360), #475(1360), #476(1360), #477(1360), #478(1360), #479(1360), #480(1360), #481(1360), #482(1360), #483(1360), #484(1360), #485(1360), #486(1360), #487(1360), #488(1360), #489(1360), #490(1360), #491(1360), #492(1360), #493(1360), #494(1360), #495(1360), #496(1360), #497(1360), #498(1360), #499(1360), #500(1360), #501(1360), #502(1360), #503(1360), #504(1360), #505(1360), #506(1360), #507(1360), #508(1360), #509(1360), #510(1360), #511(1360), #512(1360), #513(1360), #514(1360), #515(1360), #516(1360), #517(1360), #518(1360), #519(1360), #520(1360), #521(1360), #522(1360), #523(1360), #524(1360), #525(1360), #526(1360), #527(1360), #528(1360), #529(1360), #530(1360), #531(1360), #532(1360), #533(1360), #534(1360), #535(1360), #536(1360), #537(1360), #538(1360), #539(1360), #540(1360), #541(1360), #542(1360), #543(1360), #544(1360), #545(1360), #546(1360), #547(1360), #548(1360), #549(1360), #550(1360), #551(1360), #552(1360), #553(1360), #554(1360), #555(1360), #556(1360), #557(1360), #558(1360), #559(1360), #560(1360), #561(1360), #562(1360), #563(1360), #564(1360), #565(1360), #566(1360), #567(1360), #568(1360), #569(1360), #570(1360), #571(1360), #572(1360), #573(1360), #574(1360), #575(1360), #576(1360), #577(1360), #578(1360), #579(1360), #580(1360), #581(1360), #582(1360), #583(1360), #584(1360), #585(1360), #586(1360), #587(1360), #588(1360), #589(1360), #590(1360), #591(1360), #592(1360), #593(1360), #594(1360), #595(1360), #596(1360), #597(1360), #598(1360), #599(1360), #600(1360), #601(1360), #602(1360), #603(1360), #604(1360), #605(1360), #606(1360), #607(1360), #608(1360), #609(1360), #610(1360), #611(1360), #612(1360), #613(1360), #614(1360), #615(1360), #616(1360), #617(1360), #618(1360), #619(1360), #620(1360), #621(1360), #622(1360), #623(1360), #624(1360), #625(1360), #626(1360), #627(1360), #628(1360), #629(1360), #630(1360), #631(1360), #632(1360), #633(1360), #634(1360), #635(1360), #636(1360), #637(1360), #638(1360), #639(1360), #640(1360), #641(1360), #642(1360), #643(1360), #644(1360), #645(1360), #646(1360), #647(1360), #648(1360), #649(1360), #650(1360), #651(1360), #652(1360), #653(1360), #654(1360), #655(1360), #656(1360), #657(1360), #658(1360), #659(1360), #660(1360), #661(1360), #662(1360), #663(1360), #664(1360), #665(1360), #666(1360), #667(1360), #668(1360), #669(1360), #670(1360), #671(1360), #672(1360), #673(1360), #674(1360), #675(1360), #676(1360), #677(1360), #678(1360), #679(1360), #680(1360), #681(1360), #682(1360), #683(1360), #684(1360), #685(1360), #686(1360), #687(1360), #688(1360), #689(1360), #690(1360), #691(1360), #692(1360), #693(1360), #694(1360), #695(1360), #696(1360), #697(1360), #698(1360), #699(1360), #700(1360), #701(1360), #702(1360), #703(1360), #704(1360), #705(1360), #706(1360), #707(1360), #708(1360), #709(1360), #710(1360), #711(1360), #712(1360), #713(1360), #714(1360), #715(1360), #716(1360), #717(1360), #718(1360), #719(1360), #720(1360), #721(1360), #722(1360), #723(1360), #724(1360), #725(1360), #726(1360), #727(1360), #728(1360), #729(1360)		



747, 1360, 1360, 1360, 1360, 1360

[Frame: 14, payload: 0-746 (747 bytes)]  
 [Frame: 15, payload: 747-2106 (1360 bytes)]  
 [Frame: 16, payload: 2107-3466 (1360 bytes)]  
 [Frame: 17, payload: 3467-4826 (1360 bytes)]  
 [Frame: 18, payload: 4827-6186 (1360 bytes)]  
 [Frame: 19, payload: 6187-7546 (1360 bytes)]

- 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？  
最小 239，够用，因为该窗口大小会一直增加

```
> Flags: 0x010 (ACK)
  Window size value: 239
  [Calculated window size: 30592]
```

- 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？  
否，应为客户端发送的 seq 并不重复
- TCP 连接的 throughput (bytes transferred per unit time) 是多少？请写出你的计算过程。  
头部为 54B，共 106 个包， $106 \times 54 = 5724\text{B}$   
总传送数据为  $152982 + 5724 = 158706\text{B}$   
时间间隔： $5.073677 - 4.329848 = 0.743829\text{s}$   
 $\text{throughput} = 158706\text{B} / 0.743829\text{s} = 213363.555\text{Bps}$

实验结果：

#### 4. IP分析

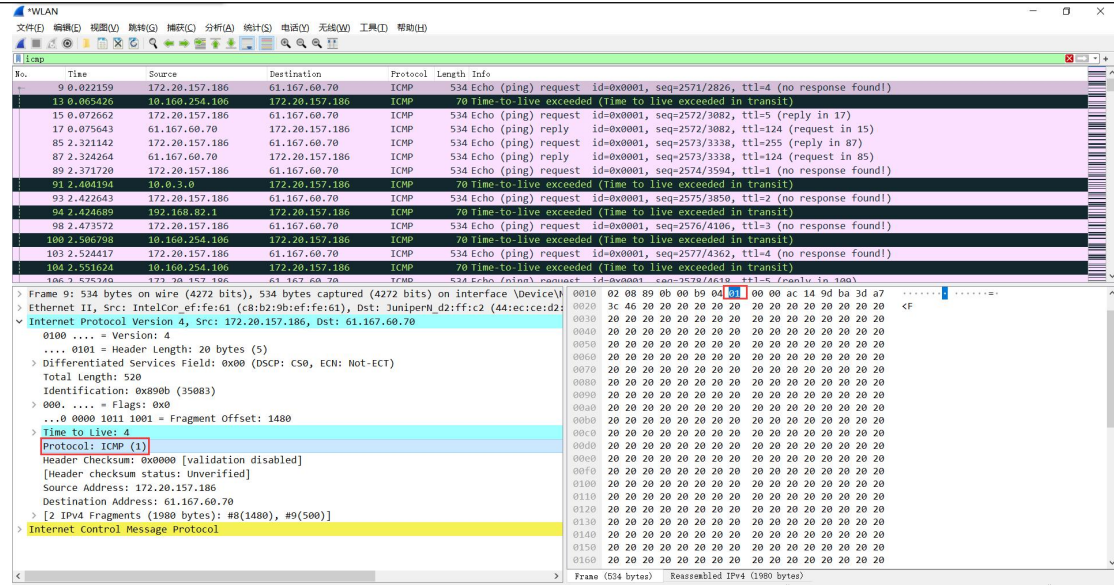
- 通过执行 traceroute 执行捕获数据包
- 对捕获的数据包进行分析

1)

- 你主机的 IP 地址是什么？  
172.20.157.186

Source	Destination
172.20.157.186	61.167.60.70
10.160.254.106	172.20.157.186
172.20.157.186	61.167.60.70
61.167.60.70	172.20.157.186
172.20.157.186	61.167.60.70
61.167.60.70	172.20.157.186

- 在 IP 数据包头中，上层协议 (upper layer) 字段的值是什么？  
01



- 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？  
IP 头部有 20 字节  
净载为 36 字节（净载= total length - 头部大小）

Internet Protocol Version 4, Src: 61.167.60.70, Dst: 172.20.157.186

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 56

Identification: 0xc414 (50196)

> 010. .... = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 124

Protocol: ICMP (1)

Header checksum: 0x76f4 [validation disabled]

[Header checksum status: Unverified]

Source Address: 61.167.60.70

Destination Address: 172.20.157.186

- 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？  
没有，因为 Don't fragment 为 1, More segments 为 0.

```

Internet Protocol Version 4, Src: 61.167.60.70, Dst: 172.20.157.186
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 56
  Identification: 0xc414 (50196)
  ✓ 010. .... = Flags: 0x2, Don't fragment
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 124
  Protocol: ICMP (1)
  Header Checksum: 0x76f4 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 61.167.60.70
  Destination Address: 172.20.157.186
  
```

2)

- 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？  
ID, TTL, Header checksum
- 哪些字段必须保持常量？哪些字段必须改变？为什么？  
ID 必须改变，因为其为鉴别码，用以区分不同的数据包  
TTL 必须改变，来自 traceroute 的要求，用来测试路径上的路由信息  
Header checksum 必须改变，这是首部校验和，前面的字段改变该值会随之改变。
- 描述你看到的 IP 数据包 Identification 字段值的形式。  
16bit，在某一范围内递增。

3)

- Identification 字段和 TTL 字段的值是什么？  
Identification: 50196  
TTL: 124



```

v Internet Protocol Version 4, Src: 61.167.60.70, Dst: 172.20.157.186
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 56
  Identification: 0xc414 (50196)
v 010. .... = Flags: 0x2, Don't fragment
  0... .... = Reserved bit: Not set
  .1.. .... = Don't fragment: Set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 124
  Protocol: ICMP (1)
  Header Checksum: 0x76f4 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 61.167.60.70
  Destination Address: 172.20.157.186
  
```

- 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？  
保持不变。原因：IP 是无连接服务，相同的标识是为了分段后组装成同一段，给同一个主机返回的标识不代表序号，因此 Identification 字段不变；又因为 是第一跳路由器发回的数据报，所以 TTL 字段是最大值-1。

4)

- 该消息是否被分解成不止一个 IP 数据报？  
是，分解为了 2 个

```

v 000. .... = Flags: 0x0
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0000 1011 1001 = Fragment Offset: 1480
  Time to Live: 124
  Protocol: ICMP (1)
  Header Checksum: 0x7082 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 61.167.60.70
  Destination Address: 172.20.157.186
v [2 IPv4 Fragments (1980 bytes): #19964(1480), #19965(500)]
  [Frame: 19964, payload: 0-1479 (1480 bytes)]
  [Frame: 19965, payload: 1480-1979 (500 bytes)]
  [Fragment count: 2]
  
```

- 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？

more segment 为 1，表明被分片了，且当前不是最后一块，该分片长度为 1500Byte。



C.

- 原始数据包被分成了多少片？

3

▼ [3 IPv4 Fragments (3480 bytes): #7687(1480), #7688(1480), #7689(520)]  
[Frame: 7687, payload: 0-1479 (1480 bytes)]  
[Frame: 7688, payload: 1480-2959 (1480 bytes)]  
[Frame: 7689, payload: 2960-3479 (520 bytes)]  
[Fragment count: 3]

- 这些分片中 IP 数据报头部哪些字段发生了变化？  
前两个分片 more segment 为 1，最后一个为 0。  
第一个分片偏移为 0，第二个为 1480，第三个为 2960。

## 5. 抓取 ARP 数据包

- 说明 ARP 缓存中每一列的含义是什么？  
第一列：IP 地址  
第二列：MAC 地址  
第三列：类型（静态不变，动态超过一定时间，记录会被删除）

```
C:\Windows\System32>arp -a

接口: 172.20.157.186 --- 0x3
Internet 地址      物理地址      类型
172.20.0.1         44-ec-ce-d2-ff-c2 动态
172.20.157.148     b8-08-d7-66-ff-7a 动态
172.20.194.20      ec-0e-c4-4d-9b-67 动态
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
239.11.20.1        01-00-5e-0b-14-01 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

接口: 192.168.237.1 --- 0x4
Internet 地址      物理地址      类型
192.168.237.254    00-50-56-eb-86-df 动态
192.168.237.255    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
239.11.20.1        01-00-5e-0b-14-01 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

接口: 192.168.126.1 --- 0xe
Internet 地址      物理地址      类型
192.168.126.254    00-50-56-e6-fe-23 动态
192.168.126.255    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
239.11.20.1        01-00-5e-0b-14-01 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

接口: 192.168.137.1 --- 0x11
Internet 地址      物理地址      类型
192.168.137.255    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
239.11.20.1        01-00-5e-0b-14-01 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
```

- 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？

查看 arp 解析即可知格式，查看每一部分所占位数即可知所占字节数

hardware type:16bit

protocol type:16bit

hardware size:8bit

protocol size:8bit

opcode:16bit

sender mac address:48bit

```

sender ip address:32bit
target mac address:48bit
target ip address:32bit
▼ 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_ef:fe:61 (c8:b2:9b:ef:fe:61)
  Sender IP address: 172.20.157.186
  Target MAC address: HuaweiTe_66:ff:7a (b8:08:d7:66:ff:7a)
  Target IP address: 172.20.157.148

```

- 如何判断一个ARP数据是请求包还是应答包？

根据 opcode 值：

请求：opcode = 1

应答：opcode = 2

Opcode: request (1)

Opcode: reply (2)

- 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？

查询时没有相应的mac地址，即无法在链路层装配该ip地址的mac帧，因而采用广播的方式；应答时，主机可以从arp请求中知道源主机的mac地址，因此对特定主机应答，从而减少网络流量的耗费。

## 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？

我的：172.20.157.186

目的：61.149.23.29

6 0.361722 192.168.43.172 223.166.151.63 UDP 337 4023 → 8000 Len=2

- 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？

我的：4007

目的：8000

6 0.361722 192.168.43.172 223.166.151.63 UDP 337 4023 → 8000 Len=2

- 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: 4007, Dst Port: 8000

Source Port: 4007

Destination Port: 8000

Length: 103

Checksum: 0x9ef9 [unverified]

[Checksum Status: Unverified]

[Stream index: 0]

> [Timestamps]

UDP payload (95 bytes)

source port:16bit

destination port:16bit

length:16bit

checksum:16bit

- 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？

因为服务器应返回接受的结果给客户端。

可以看出 udp 的不可靠数据传输，因为只提供了一次返回的 ack，没有保证数据一定送达。

还可以看出 udp 数据包没有序列号，因此不能像 tcp 协议一样先握手再发送数据，因此发送的数据是乱序的。

## 7. 利用 WireShark 进行 DNS 协议分析

The screenshot shows the Wireshark interface with a list of network packets. The selected packet is Frame 1116, which is a DNS Standard query response. The packet details pane shows the following structure:

- Frame 1116: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface \Device\NPF...
- Ethernet II, Src: IntelCor\_ef:fe:61 (c8:b2:9b:ef:fe:61), Dst: JuniperH\_d2:ff:c2 (44:ec:cce2:...
- Internet Protocol Version 4, Src: 172.20.157.186, Dst: 10.128.1.114
- 0100 .... = Version: 4
- .... 0101 = Header Length: 20 bytes (5)
- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 59
- Identification: 0x2b5b (11099)
- 0000 .... = Flags: 0x0
- 0... .... = Reserved bit: Not set
- .0... .... = Don't fragments: Not set
- ..0... .... = More fragments: Not set
- ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 128
- Protocol: UDP (17)
- Header Checksum: 0x0000 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 172.20.157.186
- Destination Address: 10.128.1.114
- User Datagram Protocol, Src Port: 61871, Dst Port: 53
- Source Port: 61871
- Destination Address (ip.dst), 4 byte(s)

The packet bytes pane shows the hex data for the DNS response, including the query ID, flags, and the response data.

问题讨论：

无。

心得体会：



通过本次实验，我对http, tcp, udp, ip, arp, dns等报文的结构有了全面的了解，并对tcp连接的建立过程和数据传输过程有了更为清晰的认识。