



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	郭子阳		院系	计算机学院		
班级	1703101		学号	1170300520		
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	2019.11.9		
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：

- 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

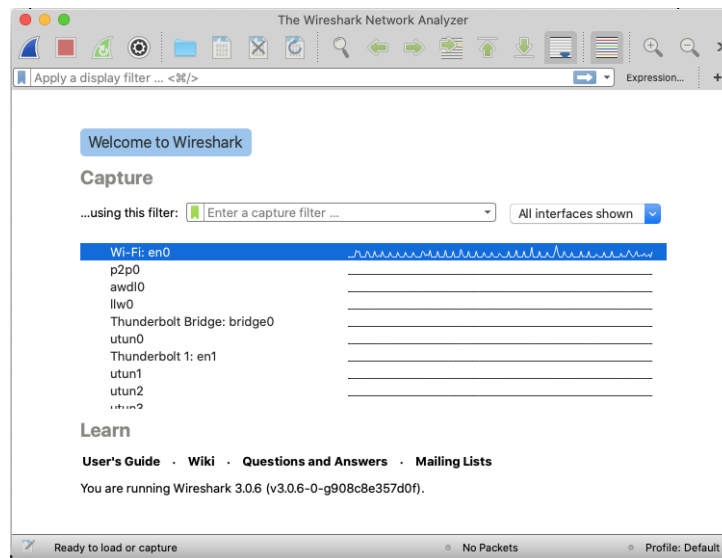
选做内容：

- a) 利用Wireshark分析DNS协议
- b) 利用Wireshark分析UDP协议
- c) 利用Wireshark分析ARP协议

实验过程与结果：

1. 学习Wireshark的使用

选择网卡后即可自动开始抓取数据



2. 利用Wireshark分析HTTP协议

1) HTTP GET/response 交互

访问<http://hitgs.hit.edu.cn/news>

结果如下：

Capturing from Wi-Fi: en0

No.	Time	Source	Destination	Protocol	Length	Info
73	7.099844	2001:250:fe01:130:...	2001:da8:b800:253:...	HTTP	621	GET /news/ HTTP/1.1
75	7.121445	2001:da8:b800:253:...	2001:250:fe01:130:...	HTTP	527	HTTP/1.1 302 Found
81	7.127773	2001:250:fe01:130:...	2001:da8:b800:253:...	HTTP	629	GET /news/main
83	7.150281	2001:da8:b800:253:...	2001:250:fe01:130:...	HTTP	1273	HTTP/1.1 200 OK
100	7.197310	2001:250:fe01:130:...	2001:da8:b800:253:...	HTTP	563	GET /_js/theme
101	7.197538	2001:250:fe01:130:...	2001:da8:b800:253:...	HTTP	553	GET /_js/theme
102	7.197661	2001:250:fe01:130:...	2001:da8:b800:253:...	HTTP	543	GET /_js/jquery

Frame 73: 621 bytes on wire (4968 bits), 621 bytes captured (4968 bits) on interface 0

Ethernet II, Src: Apple_c4:29:b8 (d0:81:7a:c4:29:b8), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)

Internet Protocol Version 6, Src: 2001:250:fe01:130:7536:656a:77cd:df71, Dst: 2001:da8:b800:253:656a:77cd:df71

Transmission Control Protocol, Src Port: 62646, Dst Port: 80, Seq: 1, Ack: 1, Len: 535

Hypertext Transfer Protocol

GET /news/ HTTP/1.1\r\n

Host: hitgs.hit.edu.cn\r\n

Connection: keep-alive\r\n

0000 58 69 6c a5 e2 d3 d0 81 7a c4 29 b8 86 dd 60 04 Xl...z...
 0010 cb 20 02 37 06 40 20 01 02 50 fe 01 01 30 75 36 .7@...P...0u6
 0020 65 6a 77 cd df 71 20 01 0d a8 b8 00 02 53 00 00 ejw...q...S...
 0030 00 00 db d9 e2 19 f4 b6 00 50 53 45 7f 68 21 17PSE:h!
 0040 1e 95 80 18 08 04 c9 f1 00 00 01 01 08 0a 42 22B"
 0050 27 3a 2a 0f 51 45 47 45 54 20 2f 6e 65 77 73 2f '*:QEGE T /news/
 0060 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a HTTP/1.1:Host
 0070 20 68 69 74 67 73 2e 68 69 74 2e 65 64 75 2e 63 hitgs.h it.edu.c
 0080 6e 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b n...Conne ction: k

Wi-Fi: en0: <live capture in progress> Packets: 3817 · Displayed: 275 (7.2%) Profile: Default

HTTP协议请求头如下

```

Hypertext Transfer Protocol
  GET /news/ HTTP/1.1\r\n
    Host: hitgs.hit.edu.cn\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_1) 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: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7\r\n
    Cookie: _ga=GA1.3.1995953044.1554477947; JSESSIONID=99C9D9F83987910B3EF04AEFE287D1A5\r\n
    [Full request URI: http://hitgs.hit.edu.cn/news/]
    [HTTP request 1/1]
    [Response in frame: 75]
  
```

HTTP响应头如下:

```

Hypertext Transfer Protocol
  HTTP/1.1 302 Found\r\n
    Date: Thu, 07 Nov 2019 12:47:03 GMT\r\n
    Server: Apache\r\n
    X-Frame-Options: SAMEORIGIN\r\n
    Frame-Options: SAMEORIGIN\r\n
    Location: /news/main.psp\r\n
    Content-Length: 198\r\n
    Connection: close\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    [HTTP response 1/1]
    [Time since request: 0.021601000 seconds]
    [Request in frame: 73]
    [Request URI: http://hitgs.hit.edu.cn/news/]
    File Data: 198 bytes
  Line-based text data: text/html (7 lines)
  
```

可以看出，浏览器运行的HTTP版本为HTTP 1.1，访问的服务器所运行的HTTP协议版本为HTTP 1.1

请求头中有Accept-Language字段，表明可接受en-US版本的对象
服务器返回的状态码为302

IP协议报文段如下:

```

▼ Internet Protocol Version 6, Src: 2001:250:fe01:130:7536:656a:77cd:df71, Dst: 2001:da8:b800:
  0110 .... = Version: 6
  ► .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 0100 1100 1011 0010 0000 = Flow Label: 0x4cb20
  Payload Length: 567
  Next Header: TCP (6)
  Hop Limit: 64
  Source: 2001:250:fe01:130:7536:656a:77cd:df71
  Destination: 2001:da8:b800:253::dbd9:e219
  
```

表明本机的IP为2001:250:fe01:130:7536:656a:77cd:df71，服务器的IP地址为2001:da8:b800:253::dbd9:e219，都使用IPv6

2) HTTP 条件 GET/response 交互

由于<http://hitgs.hit.edu.cn/news>已无内容，改为访问<http://www.people.com.cn>情况浏览器缓存后，首次访问网页时，请求头如下：

```

▼ Hypertext Transfer Protocol
  ► GET / HTTP/1.1\r\n
    Host: www.people.com.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 (Macintosh; Intel Mac OS X 10_15_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/si
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7\r\n
    ► Cookie: ALLYESID4=127105DB47B0718F; sso_c=0; sfr=1; wdcid=6a148fbb437bd694; _ma_tk=cyxbcbs0uolenxigd995ugc0e
      \r\n
    [Full request URI: http://www.people.com.cn/]
    [HTTP request 1/22]
    [Response in frame: 55]
    [Next request in frame: 58]
  
```

请求头中并没有IF-MODIFIED-SINCE字段
服务器响应如下：

```

▼ Hypertext Transfer Protocol
  ► HTTP/1.1 200 OK\r\n
    Content-Type: text/html\r\n
    Connection: keep-alive\r\n
    X-Cache: HIT from PDcache-42 :www.people.com.cn\r\n
    Vary: Accept-Encoding\r\n
    Powered-By-ChinaCache: HIT from BGP-YZ-b-D72\r\n
    Powered-By-ChinaCache: HIT from CHN-SH-a-3EJ\r\n
    ETag: W/"5dc414a0-28071"\r\n
    ► Content-Length: 39277\r\n
    Server: nginx/1.14.2\r\n
    Content-Encoding: gzip\r\n
    X-Cache-Hits: 18\r\n
    Expires: Thu, 07 Nov 2019 12:59:58 GMT\r\n
    Date: Thu, 07 Nov 2019 12:58:58 GMT\r\n
    Last-Modified: Thu, 07 Nov 2019 12:57:04 GMT\r\n
    Age: 9\r\n
    Accept-Ranges: bytes\r\n
    CACHE: TCP_HIT\r\n
    CC_CACHE: TCP_REFRESH_HIT\r\n
    \r\n
    [HTTP response 1/23]
    [Time since request: 0.178612000 seconds]
    [Request in frame: 9]
    [Next request in frame: 58]
    [Next response in frame: 117]
    [Request URI: http://www.people.com.cn/favicon.ico]
    Content-encoded entity body (gzip): 39277 bytes -> 163953 bytes
    File Data: 163953 bytes
  ► Line-based text data: text/html (2165 lines)
  
```

响应的状态码为200，响应头中包含了LAST-MODIFIED字段，值为Thu, 07 Nov 2019 12:57:04 GMT，响应包含了文件的所有内容，File Data为163953字节。

当再次刷新网页时，浏览器向服务器发送的请求头如下：

```

▼ Hypertext Transfer Protocol
▶ GET / HTTP/1.1\r\n
Host: www.people.com.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 (Macintosh; Intel Mac OS X 10_15_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.101 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7\r\n
▶ Cookie: ALLYESID4=127105DB47B0718F; sso_c=0; sfr=1; wdcid=6a148fbb437bd694; _ma_tk=cyxbcbs0uolenxigd995ugc0e\r\n
If-None-Match: W/"5dc414a0-28071"\r\n
If-Modified-Since: Thu, 07 Nov 2019 12:57:04 GMT\r\n
\r\n
[Full request URI: http://www.people.com.cn/]
[HTTP request 1/3]
[Response in frame: 72]
[Next request in frame: 171]

```

其中包含了If-Modified-Since字段，值和上一次返回的Last-Modified的值相同，都是Thu, 07 Nov 2019 12:57:04 GMT。

服务器响应头为：

```

▼ Hypertext Transfer Protocol
▶ HTTP/1.1 304 Not Modified\r\n
Server: nginx\r\n
Connection: keep-alive\r\n
Date: Thu, 07 Nov 2019 12:59:58 GMT\r\n
ETag: W/"5dc414a0-28071"\r\n
Last-Modified: Thu, 07 Nov 2019 12:57:04 GMT\r\n
Expires: Thu, 07 Nov 2019 13:00:58 GMT\r\n
Age: 59\r\n
CC_CACHE: TCP_REFRESH_HIT\r\n
\r\n
[HTTP response 1/3]
[Time since request: 0.098501000 seconds]
[Request in frame: 49]
[Next request in frame: 171]
[Request URI: http://www.people.com.cn/]

```

服务器响应的状态码为304，并没有包含文件信息，因为浏览器缓存的内容并未过期，服务器返回的字段里有ETag字段，指示缓存编号，浏览器可以根据ETag直接在缓存中查找文件，该ETag与第一次访问时返回的ETag字段内容一致。

3. TCP分析

上传Alice.txt后，wireshark获取到的第一个TCP包如下：

No.	Time	Source	Destination	Protocol	Length	Info
9	0.696245	172.20.67.203	58.251.80.219	TCP	54	64682 → 8080 [ACK] Seq=1 Ack=
12	0.701904	172.20.67.203	52.17.172.5	TCP	78	64990 → 80 [SYN] Seq=0 Win=65
13	0.980842	172.20.67.203	52.17.172.5	TCP	78	64991 → 80 [SYN] Seq=0 Win=65
14	1.015307	52.17.172.5	172.20.67.203	TCP	74	80 → 64990 [SYN, ACK] Seq=0 A
15	1.015429	172.20.67.203	52.17.172.5	TCP	66	64990 → 80 [ACK] Seq=1 Ack=1
16	1.015979	172.20.67.203	52.17.172.5	TCP	755	64990 → 80 [PSH, ACK] Seq=1 Ack=1 W:

▶	Frame 12: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
▶	Ethernet II, Src: Apple_c4:29:b8 (d0:81:7a:c4:29:b8), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
▶	Internet Protocol Version 4, Src: 172.20.67.203, Dst: 52.17.172.5
▼	Transmission Control Protocol, Src Port: 64990, Dst Port: 80, Seq: 0, Len: 0
	Source Port: 64990
	Destination Port: 80
	[Stream index: 4]
	[TCP Segment Len: 0]
	Sequence number: 0 (relative sequence number)
	[Next sequence number: 0 (relative sequence number)]
	Acknowledgment number: 0
	1011 = Header Length: 44 bytes (11)
▼	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
0 = Fin: Not set
	[TCP Flags:S.]
	Window size value: 65535
	[Calculated window size: 65535]

可以看出，客户端的IP地址为172.20.67.203，TCP端口号为64990，服务器的IP地址为52.17.172.5，端口号为80

用于初始化TCP SYN报文段的序号（Seq）为0，在Flags中，Syn位被置为1，表明该报文段为SYN报文段。

服务器向客户端返回的SYNACK报文段如下，该报文段序号为0，Acknowledgment序号为1，因为客户端向服务器发送的报文段序号为0，所以服务器期望获得的后续报文段序号为1。Flags中将Acknowledgment和Syn字段设置为1，表明该报文段为SYNACK报文段。

▼	Transmission Control Protocol, Src Port: 80, Dst Port: 64990, Seq: 0, Ack: 1, Len: 0
	Source Port: 80
	Destination Port: 64990
	[Stream index: 4]
	[TCP Segment Len: 0]
	Sequence number: 0 (relative sequence number)
	[Next sequence number: 0 (relative sequence number)]
	Acknowledgment number: 1 (relative ack number)
	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.]
	Window size value: 26847

TCP三次握手的报文段如下（13、14、15号）

13	0.980842	172.20.67.203	52.17.172.5	TCP	78	64991 → 80 [SYN] Seq=0 Win=65
14	1.015307	52.17.172.5	172.20.67.203	TCP	74	80 → 64990 [SYN, ACK] Seq=0 A
15	1.015429	172.20.67.203	52.17.172.5	TCP	66	64990 → 80 [ACK] Seq=1 Ack=1

包含HTTP POST命令的报文段如下，Seq为152730

No.	Time	Source	Destination	Protocol	Length	Info
169	2.742148	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80 [ACK] Seq=149834 Ack=1 Win=131712
170	2.742148	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80 [ACK] Seq=151282 Ack=1 Win=131712
171	2.742150	172.20.67.203	52.17.172.5	HTTP	347	POST /v2/5dc4370f300000575a347a2f HTTP/1.1
172	2.743534	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=84674 Win=195840
173	2.747452	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=87570 Win=197376
174	2.747457	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=94810 Win=197376
175	2.747458	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=102050 Win=197376
176	2.747459	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=104946 Win=197376
177	2.749542	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=107842 Win=197376
178	2.753063	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=110738 Win=197376
179	3.081039	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=113634 Win=197376
180	3.081044	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=117978 Win=197376
181	3.081047	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=120874 Win=197376
182	3.081048	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=123770 Win=197376 Len=0

▶ Frame 171: 347 bytes on wire (2776 bits), 347 bytes captured (2776 bits) on interface 0
 ▶ Ethernet II, Src: Apple_c4:29:b8 (d0:81:7a:c4:29:b8), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
 ▶ Internet Protocol Version 4, Src: 172.20.67.203, Dst: 52.17.172.5
 ▶ Transmission Control Protocol, Src Port: 64990, Dst Port: 80, Seq: 152730, Ack: 1, Len: 281
 Source Port: 64990
 Destination Port: 80
 [Stream index: 4]
 [TCP Segment Len: 281]
 Sequence number: 152730 (relative sequence number)
 [Next sequence number: 153011 (relative sequence number)]
 Acknowledgment number: 1 (relative ack number)
 1000 = Header Length: 32 bytes (8)
 ▶ Flags: 0x018 (PSH, 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
 Frame (347 bytes) Reassembled TCP (153010 bytes)
 Acknowledgment (tcp.flags.ack), 1 byte Packets: 1012 · Displayed: 336 (33.2%) · Dropped: 0 (0.0%) · Profile: Default

三次握手后，客户端开始传送TCP报文段，共分为107个报文段：

[Frame: 159, payload: 135353-136800 (1448 bytes)]
 [Frame: 160, payload: 136801-138248 (1448 bytes)]
 [Frame: 161, payload: 138249-139696 (1448 bytes)]
 [Frame: 162, payload: 139697-141144 (1448 bytes)]
 [Frame: 163, payload: 141145-142592 (1448 bytes)]
 [Frame: 164, payload: 142593-144040 (1448 bytes)]
 [Frame: 165, payload: 144041-145488 (1448 bytes)]
 [Frame: 166, payload: 145489-146936 (1448 bytes)]
 [Frame: 167, payload: 146937-148384 (1448 bytes)]
 [Frame: 168, payload: 148385-149832 (1448 bytes)]
 [Frame: 169, payload: 149833-151280 (1448 bytes)]
 [Frame: 170, payload: 151281-152728 (1448 bytes)]
 [Frame: 171, payload: 152729-153009 (281 bytes)]
 [Segment count: 107]
 [Reassembled TCP length: 153010]
 [Reassembled TCP Data: 504f5354202f76322f3564633433373066333030303035...]

第六个报文段如下：

26	1.441152	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80 [ACK] Seq=6482 Ack=1 Win=131712
27	1.441153	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80 [ACK] Seq=7930 Ack=1 Win=131712
28	1.441153	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80 [ACK] Seq=9378 Ack=1 Win=131712
29	1.656238	172.20.67.203	185.85.13.155	TCP	78	[TCP Retransmission] 64989 → 80 [SYN] Seq=0
30	1.762772	52.17.172.5	172.20.67.203	TCP	66	80 → 64990 [ACK] Seq=1 Ack=6482 Win=51968

[TCP Segment Len: 1448]
 Sequence number: 6482 (relative sequence number)
 [Next sequence number: 7930 (relative sequence number)]
 Acknowledgment number: 1 (relative ack number)
 1000 = Header Length: 32 bytes (8)
 ▶ Flags: 0x010 (ACK)
 Window size value: 2058
 [Calculated window size: 131712]
 [Window size scaling factor: 64]
 Checksum: 0x5edc [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 ▶ [SEQ/ACK analysis]
 ▶ [Timestamps]
 TCP payload (1448 bytes)
 [Reassembled PDU in frame: 171]
 TCP segment data (1448 bytes)

序号为6482，是在POST发送之前发送的，对应的ACK是服务器返回的第六个ACK。
前六个TCP报文段信息如下：

```
▼ [107 Reassembled TCP Segments (153010 bytes): #16(689)]
[Frame: 16, payload: 0-688 (689 bytes)]
[Frame: 17, payload: 689-2136 (1448 bytes)]
[Frame: 18, payload: 2137-3584 (1448 bytes)]
[Frame: 24, payload: 3585-5032 (1448 bytes)]
[Frame: 25, payload: 5033-6480 (1448 bytes)]
[Frame: 26, payload: 6481-7928 (1448 bytes)]
[Frame: 27, payload: 7929-9376 (1448 bytes)]
[Frame: 28, payload: 9377-10824 (1448 bytes)]
[Frame: 33, payload: 10825-12272 (1448 bytes)]
[Frame: 34, payload: 12273-13720 (1448 bytes)]
[Frame: 35, payload: 13721-15168 (1448 bytes)]
[Frame: 36, payload: 15169-16616 (1448 bytes)]
[Frame: 37, payload: 16617-18064 (1448 bytes)]
[Frame: 38, payload: 18065-19512 (1448 bytes)]
```

大小分别为689字节、1448字节、1448字节、1448字节、1448字节、1448字节。
接收端公示的最小可用缓存空间为第二个ACK报文显示的窗口大小，为111：

20	1.441051	52.17.172.5	172.20.67.203	TCP	66	80 → 64990	[ACK] Seq=1 Ack=690 Win=28416 Len=0
21	1.441052	52.17.172.5	172.20.67.203	TCP	66	80 → 64990	[ACK] Seq=1 Ack=2138 Win=31232 Len=0
22	1.441053	52.17.172.5	172.20.67.203	TCP	66	80 → 64990	[ACK] Seq=1 Ack=3586 Win=34048 Len=0
23	1.441151	172.20.67.203	52.17.172.5	TCP	66	64991 → 80	[ACK] Seq=1 Ack=1 Win=131712 Len=0
24	1.441151	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80	[ACK] Seq=3586 Ack=1 Win=131712 Len=0
25	1.441152	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80	[ACK] Seq=5034 Ack=1 Win=131712 Len=144

```

[Stream index: 4]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 690 (relative ack number)
1000 .... = Header Length: 32 bytes (8)
  ► Flags: 0x010 (ACK)
    Window size value: 111
    [Calculated window size: 28416]
    [Window size scaling factor: 256]
    Checksum: 0x4f35 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    
```

后续接收的时候，接收端会不断增加窗口大小，故接收端缓存够用
整个发送期间没有重传现象发生，因为所有发送的报文段的序号Seq都不相同
TCP连接的throughput:

寻找len为0的报文，可获取头部长度为66 bytes:

15	1.015429	172.20.67.203	52.17.172.5	TCP	66	64990 → 80	[ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=112303
16	1.015979	172.20.67.203	52.17.172.5	TCP	755	64990 → 80	[PSH, ACK] Seq=1 Ack=1 Win=131712 Len=689 TSval=112303
17	1.016184	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80	[ACK] Seq=690 Ack=1 Win=131712 Len=1448 TSval=112303
18	1.016185	172.20.67.203	52.17.172.5	TCP	1514	64990 → 80	[ACK] Seq=2138 Ack=1 Win=131712 Len=1448 TSval=112303

共分为107个段传送，TCP报文段总头部长度为66 * 107 = 7062 bytes

又因为：

```
▼ [107 Reassembled TCP Segments (153010 bytes): #16(689)]
[Frame: 16, payload: 0-688 (689 bytes)]
[Frame: 17, payload: 689-2136 (1448 bytes)]
[Frame: 18, payload: 2137-3584 (1448 bytes)]
[Frame: 24, payload: 3585-5032 (1448 bytes)]
[Frame: 25, payload: 5033-6480 (1448 bytes)]
[Frame: 26, payload: 6481-7928 (1448 bytes)]
[Frame: 27, payload: 7929-9376 (1448 bytes)]
[Frame: 28, payload: 9377-10824 (1448 bytes)]
[Frame: 33, payload: 10825-12272 (1448 bytes)]
[Frame: 34, payload: 12273-13720 (1448 bytes)]
[Frame: 35, payload: 13721-15168 (1448 bytes)]
[Frame: 36, payload: 15169-16616 (1448 bytes)]
[Frame: 37, payload: 16617-18064 (1448 bytes)]
[Frame: 38, payload: 18065-19512 (1448 bytes)]
```

于是TCP总传输数据大小为153010 + 7062 = 160072 bytes

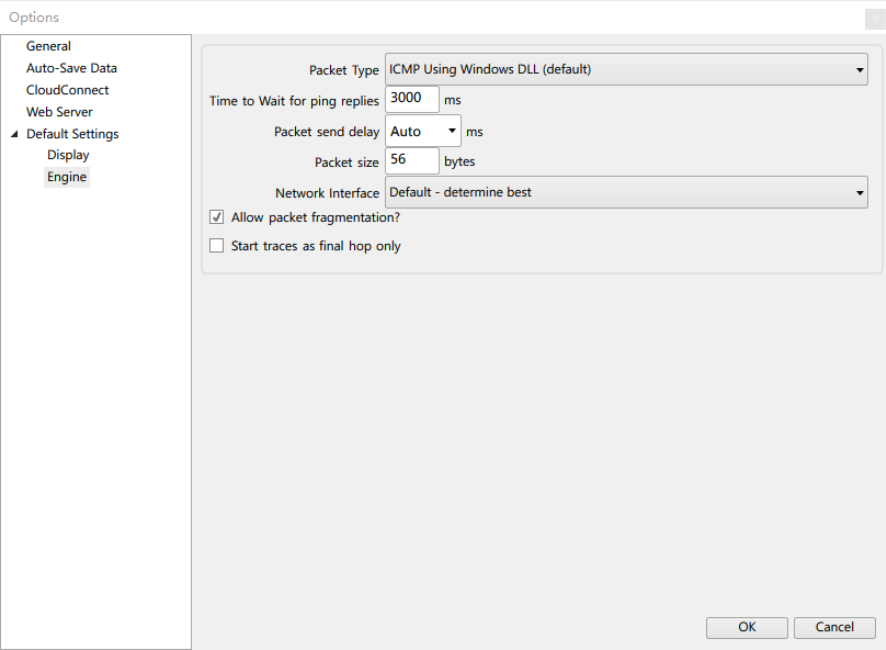
最后一个段中，wireshark提供了Timestamps字段，记录了从第一个段到最后一个段所花费的时间：

▼ [Timestamps]
[Time since first frame in this TCP stream: 2.040246000 seconds]
[Time since previous frame in this TCP stream: 0.000002000 seconds]
TCP payload (281 bytes)
TCP segment data (281 bytes)

共花费2.040246秒
于是throughput为160072 bytes / 2.040246 s = 78457.2057 bytes/s, 约为78.5 KB/s

4. IP分析

使用pingplotter，首先在设置中将packet size设置为56 bytes



主机的IP地址如下，为172.20.77.246:

→	263	27.809741	172.20.77.246	61.167.60.70	ICMP	70 Ect
←	264	27.812768	61.167.60.70	172.20.77.246	ICMP	70 Ect
	265	27.846935	172.20.77.246	61.167.60.70	ICMP	70 Ect

在IP数据包头中，上层协议号为1，如下:

▼ Internet Protocol Version 4, Src: 172.20.77.246, Dst: 61.167.60.70

0100 = Version: 4

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

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

Total Length: 56

Identification: 0x2d5c (11612)

> Flags: 0x0000

...0 0000 0000 0000 = Fragment offset: 0

Time to live: 255

Protocol: ICMP (1)

Header checksum: 0x1a71 [validation disabled]

[Header checksum status: Unverified]

Source: 172.20.77.246

Destination: 61.167.60.70

> Internet Control Message Protocol

<

0010	00 38 2d 5c 00 00 ff 01	1a 71 ac 14 4d f6 3d a7	·8- \ ···· ·q· ·M· =·
0020	3c 46 08 00 36 3c 00 01	00 01 20 20 20 20 20 20	<F· ·6<· ··
0030	20 20 20 20 20 20 20 20	20 20 20 20 20 20 20 20	
0040	20 20 20 20 20 20 20 20		

由上图可以看出，IP头有20个字节，IP数据包大小为56字节，于是净载为36字节
展开flags字段，可以看到分段信息。可以看到该段偏移为0，More fragments为0，所以没有分片

```

    v Flags: 0x0000
      0... .. = 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
    
```

通过分析多个ICMP包，总是在变的字段有Identification、Time to live和Header checksum
Identification用于鉴别不同的数据包，Time to live用来测试路由信息，Header checksum为校验和，这三个字段必须改变，其他字段保持常量

Identification为16位二进制数，按1递增

第一跳返回的TTL exceeded消息中，Identification为0，TTL为254

```

    v Internet Protocol Version 4, Src: 192.168.80.1, Dst: 172.20.77.246
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
      > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 56
      Identification: 0x0000 (0)
      > Flags: 0x0000
      ...0 0000 0000 0000 = Fragment offset: 0
      Time to live: 254
      Protocol: ICMP (1)
      Header checksum: 0xb210 [validation disabled]
      [Header checksum status: Unverified]
      Source: 192.168.80.1
      Destination: 172.20.77.246
      > Internet Control Message Protocol
    
```

不变，因为相同的Identification是为了分段之后组装时为同一段，给同一个主机返回的ICMP，标识不代表序号，所以Identification不变，因为是第一跳路由器发送的数据报，所以TTL为最大值减一，总是为254。

当包大小改变为2000字节时，第一个Echo request被分片发送，第一个IP分片信息如下：

```

    v Internet Protocol Version 4, Src: 172.20.77.246, Dst: 61.167.60.70
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
      > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 1500
      Identification: 0x5588 (21896)
      v Flags: 0x2000, 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: 1
    
```

段偏移为0，More fragments字段设为1，表示该段为第一段，后续还有段，分片长度为1500字节

当包大小改为3000字节时，如下：

```

    [3 IPv4 Fragments (2980 bytes): #4(1480), #5(1480), #6(20)]
    [Frame: 4, payload: 0-1479 (1480 bytes)]
    [Frame: 5, payload: 1480-2959 (1480 bytes)]
    [Frame: 6, payload: 2960-2979 (20 bytes)]
    [Fragment count: 3]
    [Reassembled IPv4 length: 2980]
    [Reassembled IPv4 data: 08003f7000013e1420202020202020202020202020202020...]

```

分成了三个段发送，前两个分片More fragments字段为1，后两个分片的offset分别为1480和2960

5. 抓取ARP数据包

在CMD中输入arp -a命令，结果如下：

```

C:\Users\guo>arp -a

接口: 172.20.77.246 --- 0xa
Internet 地址      物理地址      类型
172.20.0.1        58-69-6c-a5-e2-d3 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
239.255.255.250   01-00-5e-7f-ff-fa 静态
255.255.255.255   ff-ff-ff-ff-ff-ff 静态

```

其中，第一列为IP地址，第二列MAC地址，最后一列为类型，动态说明一定时间后记录会被删除

arp -d清除arp缓存后，ping 172.20.77.246后可过滤到arp数据包，如下：

No.	Time	Source	Destination	Protocol	Length	Info
91	14.003840	IntelCor_09:fb:0a	Broadcast	ARP	42	Who has 172.20.0.1? Tell 172.20.77.246
92	14.005056	RuijieNe_a5:e2:d3	IntelCor_09:fb:0a	ARP	60	172.20.0.1 is at 58:69:6c:a5:e2:d3
154	17.950622	RuijieNe_a5:e2:d3	Broadcast	ARP	64	Gratuitous ARP for 172.20.0.1 (Request)
184	19.896729	RuijieNe_a5:e2:d3	Broadcast	ARP	64	Gratuitous ARP for 172.20.0.1 (Request)
195	20.877464	RuijieNe_a5:e2:d3	Broadcast	ARP	64	Gratuitous ARP for 172.20.0.1 (Request)
207	21.873339	RuijieNe_a5:e2:d3	Broadcast	ARP	64	Gratuitous ARP for 172.20.0.1 (Request)

```

<
> Frame 91: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Ethernet II, Src: IntelCor_09:fb:0a (d0:57:7b:09:fb:0a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Destination: Broadcast (ff:ff:ff:ff:ff:ff)
> Source: IntelCor_09:fb:0a (d0:57:7b:09:fb:0a)
  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_09:fb:0a (d0:57:7b:09:fb:0a)
    Sender IP address: 172.20.77.246
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 172.20.0.1

```

Arp数据包格式如下：

广播 Mac 地址(全 1) ◊		
目标 Mac 地址(广播 Mac 地址) ◊		源 Mac 地址 ◊
源 Mac 地址 ◊		
协议类型 ◊		
硬件类型 ◊		协议类型 ◊
硬件地址长度 ◊	协议长度 ◊	操作(请求 1) ◊
发送方硬件地址(前 32 位) ◊		
发送方硬件地址(后 16 位) ◊		发送方 IP 地址(前 16 位) ◊
发送方 IP 地址(后 16 位) ◊		目标硬件地址(前 16 位) ◊
目标硬件地址(后 32) ◊		
目标 IP 地址(32 位) ◊		@51CTO博客

接收方MAC	6字节
发送方MAC	6字节
Ethertype	2字节
硬件类型 hdtype	2字节
上层协议类型protyp	2字节
MAC地址长度hdsiz	1字节
IP地址长度prosize	1字节
操作码 op	2字节
发送方MAC smac[6]	6字节
发送方IP sip[4]	4字节
接收方MAC dmac[6]	6字节
接收方IP dip	4字节
填充数据	18字节

判断arp包是请求包还是应答包可以根据opcode，opcode为1的是请求包，2为应答包
由于请求时，源主机不知道目的主机的mac地址，故无法在链路层封装该IP的mac帧，于是采用广播的模式，而当应答时，主机可以通过arp帧获取到源主机的mac地址，可以对特定主机应答。

6. 抓取UDP数据包

从QQ中发送消息后，捕获到的数据包如下：

2183	11.364955	125.39.132.99	172.20.77.246	UDP	81 8000 → 4024 Len=39
2184	11.364955	125.39.132.99	172.20.77.246	UDP	81 8000 → 4024 Len=39
2185	11.367418	172.20.77.246	125.39.132.99	UDP	489 4024 → 8000 Len=447
2186	11.858567	172.20.77.246	125.39.132.99	UDP	193 4024 → 8000 Len=151
2190	12.777250	172.20.77.246	125.39.132.99	UDP	81 4024 → 8000 Len=39
2192	12.789270	172.20.77.246	125.39.132.99	UDP	380 4024 → 8000 Len=247
> Frame 2183: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0					
Ethernet II, Src: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3), Dst: IntelCor_09:fb:0a (d0:57:7b:09:fb:0a)					
> Destination: IntelCor_09:fb:0a (d0:57:7b:09:fb:0a)					
> Source: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)					
Type: IPv4 (0x0800)					
> Internet Protocol Version 4, Src: 125.39.132.99, Dst: 172.20.77.246					
> User Datagram Protocol, Src Port: 8000, Dst Port: 4024					
Source Port: 8000					
Destination Port: 4024					
Length: 47					
Checksum: 0x56f9 [unverified]					
[Checksum Status: Unverified]					
[Status: Index: 0]					
0000	d0 57 7b 09 fb 0a 58 69	6c a5 e2 d3 08 00 45 00	.W{...Xi 1....E.		
0010	00 43 d9 32 40 00 36 11	6f e2 7d 27 84 63 ac 14	.C.2@.6. o.}.c..		
0020	4d f6 1f 40 0f b8 00 2f	56 f9 02 38 3b 01 52 4c	M.@.../ V.8;.RL		

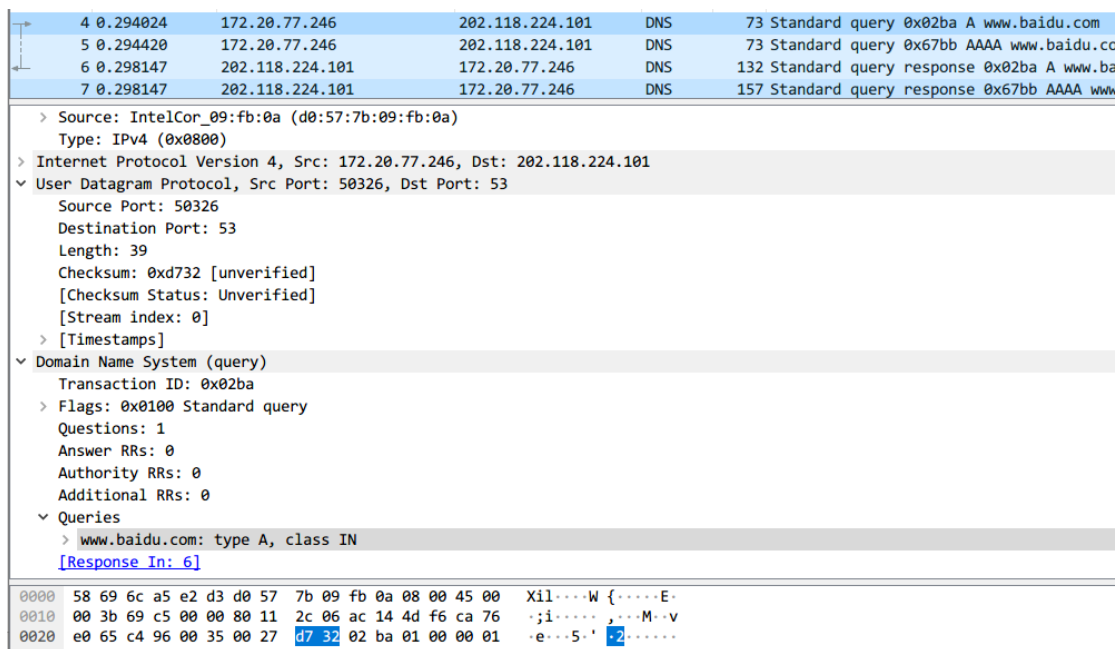
该消息基于UDP，本机IP为172.20.77.246，目的IP为125.39.132.99，本机QQ端口为4024，目的服务器的端口为8000

消息包含字段：源端口，目的端口，长度，校验和，各占16个字节

没发送一个ICQ数据包，服务器就会返回一个ICQ数据包，返回的是接受结果，UDP是不可靠的数据传输，仅仅返回一个简单的接收状态，无重传等机制，UDP数据包是没有序列号的，于是数据是乱序的无连接的。

7. 利用DNS进行DNS协议分析

请求：



4 0.294024 172.20.77.246 202.118.224.101 DNS 73 Standard query 0x02ba A www.baidu.com

5 0.294420 172.20.77.246 202.118.224.101 DNS 73 Standard query 0x67bb AAAA www.baidu.com

6 0.298147 202.118.224.101 172.20.77.246 DNS 132 Standard query response 0x02ba A www.ba

7 0.298147 202.118.224.101 172.20.77.246 DNS 157 Standard query response 0x67bb AAAA www

> Source: IntelCor_09:fb:0a (d0:57:7b:09:fb:0a)
Type: IPv4 (0x0800)

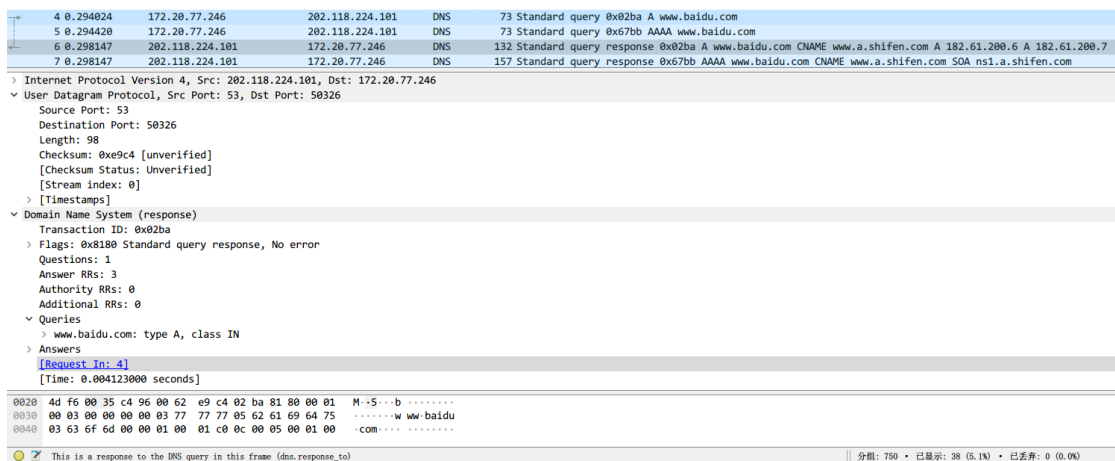
> Internet Protocol Version 4, Src: 172.20.77.246, Dst: 202.118.224.101

> User Datagram Protocol, Src Port: 50326, Dst Port: 53
Source Port: 50326
Destination Port: 53
Length: 39
Checksum: 0xd732 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
> [Timestamps]

> Domain Name System (query)
Transaction ID: 0x02ba
> Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
> Queries
> www.baidu.com: type A, class IN
[Response In: 6]

0000 58 69 6c a5 e2 d3 d0 57 7b 09 fb 0a 08 00 45 00 Xil...W {.....E:
0010 00 3b 69 c5 00 00 80 11 2c 06 ac 14 4d f6 ca 76 ;i.....,...M...v
0020 e0 65 c4 96 00 35 00 27 d7 32 02 ba 01 00 00 01 e...5...' +2.....

响应：



4 0.294024 172.20.77.246 202.118.224.101 DNS 73 Standard query 0x02ba A www.baidu.com

5 0.294420 172.20.77.246 202.118.224.101 DNS 73 Standard query 0x67bb AAAA www.baidu.com

6 0.298147 202.118.224.101 172.20.77.246 DNS 132 Standard query response 0x02ba A www.baidu.com CNAME www.a.shifen.com A 182.61.200.6 A 182.61.200.7

7 0.298147 202.118.224.101 172.20.77.246 DNS 157 Standard query response 0x67bb AAAA www.baidu.com CNAME www.a.shifen.com SOA ns1.a.shifen.com

> Internet Protocol Version 4, Src: 202.118.224.101, Dst: 172.20.77.246

> User Datagram Protocol, Src Port: 53, Dst Port: 50326
Source Port: 53
Destination Port: 50326
Length: 98
Checksum: 0xe9c4 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
> [Timestamps]

> Domain Name System (response)
Transaction ID: 0x02ba
> Flags: 0x8100 Standard query response, No error
Questions: 1
Answer RRs: 3
Authority RRs: 0
Additional RRs: 0
> Queries
> www.baidu.com: type A, class IN
> Answers
[Request In: 4]
[Time: 0.004123000 seconds]

0020 4d f6 00 35 c4 96 00 62 e9 c4 02 ba 01 00 00 01 M..S...b
0030 00 03 00 00 00 00 03 77 77 77 05 62 61 69 64 75w ww baidu
0040 03 63 6f 6d 00 00 01 00 01 c0 0c 00 05 00 01 00 .com.....

This is a response to the DNS query in this frame (dns.response_to) | 分组: 750 • 已显示: 38 (5.1%) • 已丢弃: 0 (0.0%)

心得体会：

学会了使用Wireshark进行协议分析，深入理解了各个协议的实现