



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	实验 4: 利用 Wireshark 进行协议分析					
姓名	张志路		院系	计算机学院		
班级	1603106		学号	1160300909		
任课教师	聂兰顺		指导教师	聂兰顺		
实验地点	格物 207		实验时间	2018 年 11 月 21 日		
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						

目 录

实验 4：利用 Wireshark 进行协议分析.....	3
1 实验目的.....	3
2 实验内容.....	3
3 实验环境.....	3
4 实验过程.....	4
4.1 Wireshark 的使用	4
4.2 分析 HTTP 协议	4
4.3 分析 TCP 协议	7
4.4 分析 IP 协议.....	11
4.5 抓取 ARP 数据包.....	16
4.6 抓取 UDP 数据包	17
4.7 分析 DNS 协议	19
4.8 分析 Ethernet 数据帧.....	19
5 心得体会.....	20

实验 4：利用 Wireshark 进行协议分析

1 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。

2 实验内容

必做内容：

- 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：

- ① 利用 Wireshark 分析 DNS 协议
- ② 利用 Wireshark 分析 UDP 协议
- ③ 利用 Wireshark 分析 ARP 协议。

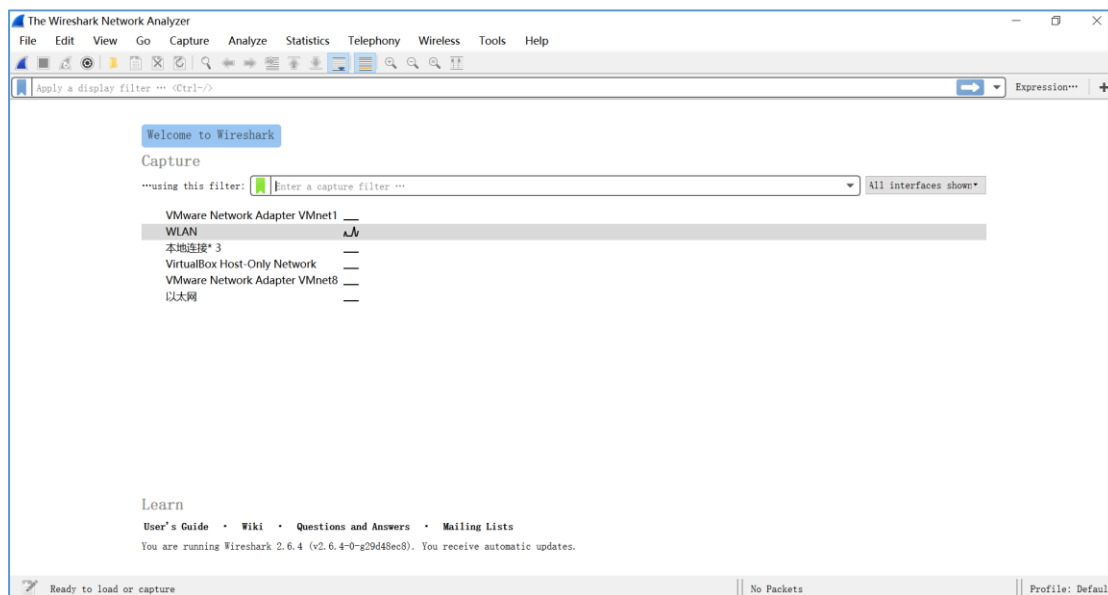
3 实验环境

- ① 操作系统：Windows10 64 位
- ② 与因特网连接的计算机网络系统
- ③ 工具：Wireshark

4 实验过程

4.1 Wireshark 的使用

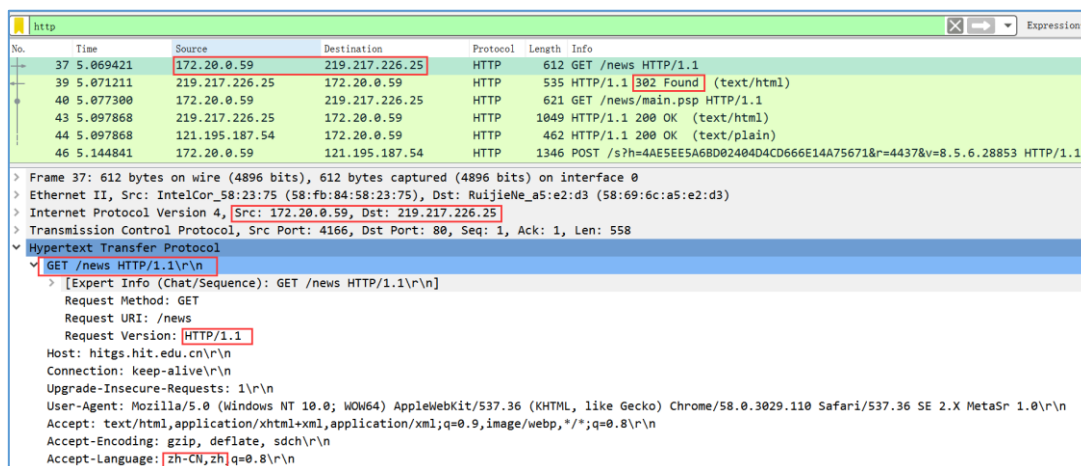
下载后安装即可。



4.2 分析 HTTP 协议

4.2.1 HTTPGET/response 交互

根据实验要求俘获窗口内容如下：



思考问题如下：

- (1) 你的浏览器运行的是 HTTP1.0，还是 HTTP1.1？你所访问的服务器所运行 HTTP 协议的版本号是多少？

如上图，由请求报文可知，我的浏览器运行的是 HTTP1.1，由响应报文可知，目标服务器所运行的 HTTP 协议的版本号是 HTTP1.1。

(2) 你的浏览器向服务器指出它能接收何种语言版本的对象？

如上图，语言版本的对象 Accept-Language: zh-CN,zh。

(3) 你的计算机的 IP 地址是多少？服务器 <http://hitgs.hit.edu.cn/news> 的 IP 地址是多少？

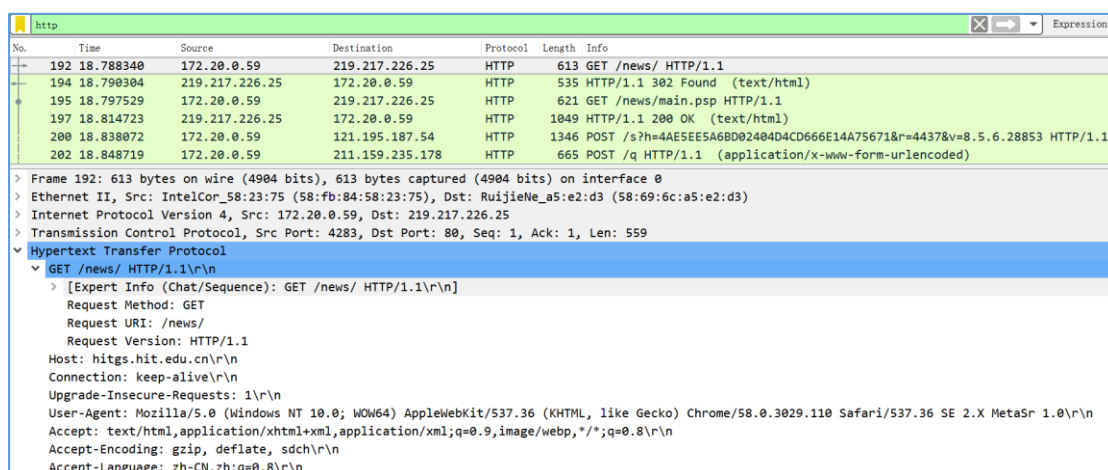
如上图，本机的 IP 地址是 172.20.0.59，服务器的 IP 地址是 219.217.226.25。

(4) 从服务器向你的浏览器返回的状态代码是多少？

如上图，状态码为 302。

4.2.2 HTTP 条件 GET/response 交互

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

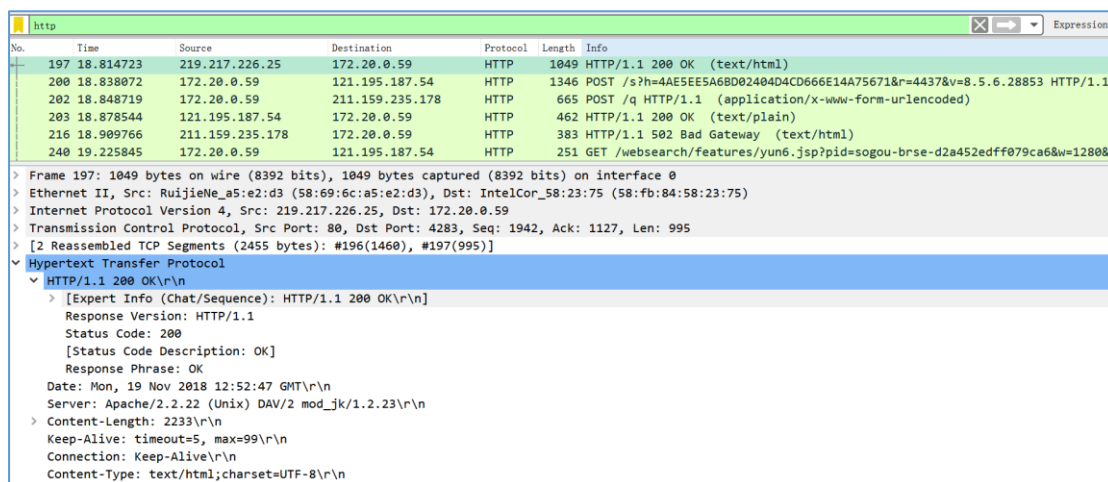


No.	Time	Source	Destination	Protocol	Length	Info
192	18.788340	172.20.0.59	219.217.226.25	HTTP	613	GET /news/ HTTP/1.1
194	18.790304	219.217.226.25	172.20.0.59	HTTP	535	HTTP/1.1 302 Found (text/html)
195	18.797529	172.20.0.59	219.217.226.25	HTTP	621	GET /news/main.psp HTTP/1.1
197	18.814723	219.217.226.25	172.20.0.59	HTTP	1049	HTTP/1.1 200 OK (text/html)
200	18.838072	172.20.0.59	121.195.187.54	HTTP	1346	POST /s?h=4AE5EE5A68D02404D4CD666E14A75671&r=4437&v=8.5.6.28853 HTTP/1.1
202	18.848719	172.20.0.59	211.159.235.178	HTTP	665	POST /q HTTP/1.1 (application/x-www-form-urlencoded)

Frame 192: 613 bytes on wire (4904 bits), 613 bytes captured (4904 bits) on interface 0
 Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
 Internet Protocol Version 4, Src: 172.20.0.59, Dst: 219.217.226.25
 Transmission Control Protocol, Src Port: 4283, Dst Port: 80, Seq: 1, Ack: 1, Len: 559
 Hypertext Transfer Protocol
 GET /news/ HTTP/1.1\r\n
 [Expert Info (Chat/Sequence): GET /news/ HTTP/1.1\r\n]
 Request Method: GET
 Request URI: /news/
 Request Version: HTTP/1.1
 Host: hitgs.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; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 SE 2.X MetaSr 1.0\r\n
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
 Accept-Encoding: gzip, deflate, sdch\r\n
 Accept-Language: zh-CN,zh;q=0.8\r\n

如上图，没有 IF-MODIFIED-SINCE 行。

(2) 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？



No.	Time	Source	Destination	Protocol	Length	Info
197	18.814723	219.217.226.25	172.20.0.59	HTTP	1049	HTTP/1.1 200 OK (text/html)
200	18.838072	172.20.0.59	121.195.187.54	HTTP	1346	POST /s?h=4AE5EE5A68D02404D4CD666E14A75671&r=4437&v=8.5.6.28853 HTTP/1.1
202	18.848719	172.20.0.59	211.159.235.178	HTTP	665	POST /q HTTP/1.1 (application/x-www-form-urlencoded)
203	18.878544	121.195.187.54	172.20.0.59	HTTP	462	HTTP/1.1 200 OK (text/plain)
216	18.909766	211.159.235.178	172.20.0.59	HTTP	383	HTTP/1.1 502 Bad Gateway (text/html)
240	19.225845	172.20.0.59	121.195.187.54	HTTP	251	GET /websearch/features/yun6.jsp?pid=sogou-brse-d2a452edff079ca6&w=1280&h=800 HTTP/1.1

Frame 197: 1049 bytes on wire (8392 bits), 1049 bytes captured (8392 bits) on interface 0
 Ethernet II, Src: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3), Dst: IntelCor_58:23:75 (58:fb:84:58:23:75)
 Internet Protocol Version 4, Src: 219.217.226.25, Dst: 172.20.0.59
 Transmission Control Protocol, Src Port: 80, Dst Port: 4283, Seq: 1942, Ack: 1127, Len: 995
 [2 Reassembled TCP Segments (2455 bytes): #196(1460), #197(995)]
 Hypertext Transfer Protocol
 HTTP/1.1 200 OK\r\n
 [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
 Response Version: HTTP/1.1
 Status Code: 200
 [Status Code Description: OK]
 Response Phrase: OK
 Date: Mon, 19 Nov 2018 12:52:47 GMT\r\n
 Server: Apache/2.2.22 (Unix) DAV/2 mod_jk/1.2.23\r\n
 Content-Length: 2233\r\n
 Keep-Alive: timeout=5, max=99\r\n
 Connection: Keep-Alive\r\n
 Content-Type: text/html; charset=UTF-8\r\n

如上图，服务器返回了文件内容，因为有 HTTP/1.1 200 OK，状态码为 200 表示明确返回了文件。

- (3) 分析你的浏览器向服务器发出的较晚的“HTTPGET”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
989	7.459086	219.217.226.25	172.20.0.59	HTTP	259	HTTP/1.1 304 Not Modified
990	7.459124	172.20.0.59	219.217.226.25	HTTP	587	GET /_js/themes/icon.css HTTP/1.1
992	7.459224	172.20.0.59	219.217.226.25	HTTP	588	GET /_css/error/error.css HTTP/1.1
993	7.459745	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified
995	7.460033	219.217.226.25	172.20.0.59	HTTP	261	HTTP/1.1 304 Not Modified
996	7.460033	219.217.226.25	172.20.0.59	HTTP	261	HTTP/1.1 304 Not Modified
999	7.460385	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified
1001	7.461056	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified

Transmission Control Protocol, Src Port: 5705, Dst Port: 80, Seq: 1822, Ack: 44638, Len: 534						
Hypertext Transfer Protocol						
GET /_css/error/error.css HTTP/1.1\r\n						
[Expert Info (Chat/Sequence): GET /_css/error/error.css HTTP/1.1\r\n]						
Request Method: GET						
Request URI: /_css/error/error.css						
Request Version: HTTP/1.1						
Referer: http://hits.hit.edu.cn/news/main.psp\r\n						
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/17.17134\r\n						
Cache-Control: max-age=0\r\n						
Accept: text/css,*/*;q=0.1\r\n						
Accept-Language: zh-CN\r\n						
Accept-Encoding: gzip, deflate\r\n						
Host: hits.hit.edu.cn\r\n						
If-Modified-Since: Wed, 15 Nov 2017 07:37:36 GMT\r\n						
If-None-Match: "e2055b-1735-55e0093c93000"\r\n						

如上图，出现了 IF-MODIFIED-SINCE 行，其后面跟的内容是该文件被浏览器缓存时的最后修改时间。

- (4) 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。

No.	Time	Source	Destination	Protocol	Length	Info
993	7.459745	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified
995	7.460033	219.217.226.25	172.20.0.59	HTTP	261	HTTP/1.1 304 Not Modified
996	7.460033	219.217.226.25	172.20.0.59	HTTP	261	HTTP/1.1 304 Not Modified
999	7.460385	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified
1001	7.461056	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified
1003	7.461469	219.217.226.25	172.20.0.59	HTTP	260	HTTP/1.1 304 Not Modified
1005	7.523479	172.20.0.59	219.217.226.25	HTTP	327	GET /favicon.ico HTTP/1.1
1006	7.525359	219.217.226.25	172.20.0.59	HTTP	497	HTTP/1.1 404 Not Found (text/html)

Transmission Control Protocol, Src Port: 80, Dst Port: 5706, Seq: 28025, Ack: 2217, Len: 206						
Hypertext Transfer Protocol						
HTTP/1.1 304 Not Modified\r\n						
[Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]						
Response Version: HTTP/1.1						
Status Code: 304						
[Status Code Description: Not Modified]						
Response Phrase: Not Modified						
Date: Mon, 19 Nov 2018 13:43:15 GMT\r\n						
Server: Apache/2.2.22 (Unix) DAV/2 mod_jk/1.2.23\r\n						
Connection: Keep-Alive\r\n						
Keep-Alive: timeout=5, max=96\r\n						
ETag: "9a29e3-7e1f-55e0094bd5400"\r\n						

如上图，HTTP 状态代码是 304 Not Modified。服务器并没有明确返回文件的内容，因为文件自询问的时间起并没有被修改。

4.3 分析 TCP 协议

- (1) 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？

o.	Time	Source	Destination	Protocol	Length	Info
67	2.947067	172.20.0.59	65.52.171.231	TLv1...	468	Application Data
69	2.983688	128.119.245.12	172.20.0.59	TCP	60	80 → 5827 [ACK] Seq=1 Ack=15241 Win=59776 Len=0
70	2.983833	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=41521 Ack=1 Win=262144 Len=1460 [TCP segment of a re.
71	2.983887	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=42981 Ack=1 Win=262144 Len=1460 [TCP segment of a re.
72	2.986665	128.119.245.12	172.20.0.59	TCP	60	80 → 5827 [ACK] Seq=1 Ack=18161 Win=65536 Len=0
73	2.986787	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=44441 Ack=1 Win=262144 Len=1460 [TCP segment of a re.
74	2.986797	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=45901 Ack=1 Win=262144 Len=1460 [TCP segment of a re.
75	2.986802	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=47361 Ack=1 Win=262144 Len=1460 [TCP segment of a re.

Frame 71: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0	
Ethernet II, Src: IntelCor_S8:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)	
Internet Protocol Version 4, Src: 172.20.0.59, Dst: 128.119.245.12	
Transmission Control Protocol, Src Port: 5827, Dst Port: 80, Seq: 42981, Ack: 1, Len: 1460	
Source Port: 5827	
Destination Port: 80	
[Stream index: 3]	
[TCP Segment Len: 1460]	
Sequence number: 42981 (relative sequence number)	
[Next sequence number: 44441 (relative sequence number)]	
Acknowledgment number: 1 (relative ack number)	
0010 = Header Length: 20 bytes (5)	
Flags: 0x010 (ACK)	

如上图，客户端主机的 IP 地址为 172.20.0.59，TCP 端口号为 5827。

- (2) 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？

如上图，服务器的 IP 地址为 128.119.245.12，TCP 端口号为 80。

- (3) 客户服务器之间用于初始化 TCP 连接的 TCPSYN 报文段的序号 (sequencenumber) 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？

Time	Source	Destination	Protocol	Length	Info
3	1.325365	125.39.132.162	TCP	135	80 → 2212 [PSH, ACK] Seq=1 Ack=1 Win=30660 Len=81
4	1.367765	172.20.0.59	TCP	54	2212 → 80 [ACK] Seq=1 Ack=82 Win=68 Len=0
6	2.252624	172.20.0.59	TCP	54	5809 → 80 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
7	2.252765	172.20.0.59	TCP	54	5810 → 80 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
8	2.253195	172.20.0.59	TCP	66	5827 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
9	2.253216	172.20.0.59	TCP	66	5828 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
10	2.274538	172.20.0.59	TCP	66	5829 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
11	2.496436	128.119.245.12	TCP	60	80 → 5810 [ACK] Seq=1 Ack=2 Win=229 Len=0

[Stream index: 3]	
[TCP Segment Len: 0]	
Sequence number: 0 (relative sequence number)	
[Next sequence number: 0 (relative sequence number)]	
Acknowledgment number: 0	
1000 = Header Length: 32 bytes (8)	
Flags: 0x002 (SYN)	
0000 = 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	
....0... = Syn: Set	

如上图，SYN 报文段的序号是 0。在该报文段中是使 Flags 中的 Syn 标志位为 1 来表示该报文段是 SYN 报文段。

- (4) 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？

Time	Source	Destination	Protocol	Length	Info
13 2.496755	128.119.245.12	172.20.0.59	TCP	66	80 → 5827 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS...
14 2.496849	172.20.0.59	65.52.171.231	TCP	54	5829 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
15 2.496950	172.20.0.59	128.119.245.12	TCP	54	5827 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
16 2.497054	128.119.245.12	172.20.0.59	TCP	66	80 → 5828 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS...
17 2.497093	172.20.0.59	128.119.245.12	TCP	694	5827 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=640 [TCP segment of a re...

Transmission Control Protocol, Src Port: 80, Dst Port: 5827, Seq: 0, Ack: 1, Len: 0

Source Port: 80
Destination Port: 5827
[Stream index: 3]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
1000 = Header Length: 32 bytes (8)
▼ Flags: 0x012 (SYN, ACK)
0000 = 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

如上图，服务器向客户端发送的 SYNACK 报文段序号是 0，该报文 Acknowledgement 字段的值是 1。

gaia.cs.umass.edu 服务器是根据上一个报文的段的 seq 加 1 来决定这个 ACK 的值的。

在该报文段中，Flags 中的 Syn 和 Acknowledgement 标志都被设置成为 1，以此来表示该报文段是 SYNACK 报文段。

- (5) 你能从捕获的数据包中分析出 tcp 三次握手过程吗？

8 2.253195	172.20.0.59	128.119.245.12	TCP	66	5827 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
13 2.496755	128.119.245.12	172.20.0.59	TCP	66	80 → 5827 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1
15 2.496950	172.20.0.59	128.119.245.12	TCP	54	5827 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0

第一次握手：Client 将标志位 SYN 置为 1，随机产生一个值 seq=J，并将该数据包发送给 Server，Client 进入 SYN_SENT 状态，等待 Server 确认。

第二次握手：Server 收到数据包后由标志位 SYN=1 知道 Client 请求建立连接，Server 将标志位 SYN 和 ACK 都置为 1，ack=J+1，随机产生一个值 seq=K，并将该数据包发送给 Client 以确认连接请求，Server 进入 SYN_RCVD 状态。

第三次握手：Client 收到确认后，检查 ack 是否为 J+1，ACK 是否为 1，如果正确则将标志位 ACK 置为 1，ack=K+1，并将该数据包发送给 Server，Server 检查 ack 是否为 K+1，ACK 是否为 1，如果正确则连接建立成功，Client 和 Server 进入 ESTABLISHED 状态，完成三次握手，随后 Client 与 Server 之间可以开始传输数据了。

(6) 包含 HTTP POST 命令的 TCP 报文段的序号是多少？

No.	Time	Source	Destination	Protocol	Length	Info
170	3.233384	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=149561 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (149456 - 149607) = 149561:149607]
171	3.233389	172.20.0.59	128.119.245.12	TCP	1514	5827 → 80 [ACK] Seq=151021 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (150913 - 151064) = 151021:151064]
172	3.233396	172.20.0.59	128.119.245.12	HTTP	539	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
173	3.237353	128.119.245.12	172.20.0.59	TCP	60	80 → 5827 [ACK] Seq=1 Ack=73641 Win=176512 Len=0
174	3.237354	128.119.245.12	172.20.0.59	TCP	60	80 → 5827 [ACK] Seq=1 Ack=75101 Win=179456 Len=0
175	3.237355	128.119.245.12	172.20.0.59	TCP	60	80 → 5827 [ACK] Seq=1 Ack=76561 Win=182400 Len=0

> Frame 172: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits) on interface 0
 > Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
 > Internet Protocol Version 4, Src: 172.20.0.59, Dst: 128.119.245.12
 > Transmission Control Protocol, Src Port: 5827, Dst Port: 80, Seq: 152481, Ack: 1, Len: 485
 Source Port: 5827
 Destination Port: 80
 [Stream index: 3]
 [TCP Segment Len: 485]
 Sequence number: 152481 (relative sequence number)
 [Next sequence number: 152966 (relative sequence number)]
 Acknowledgment number: 1 (relative ack number)
 0101 = Header Length: 20 bytes (5)

如上图，包含 HTTP POST 命令的 TCP 报文段的序号是 152481。

(7) 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？

> Transmission Control Protocol, Src Port: 5827, Dst Port: 80,
 > [106 Reassembled TCP Segments (152965 bytes): #17(640), #20(
 [Frame: 17, payload: 0-639 (640 bytes)]
 [Frame: 20, payload: 640-2099 (1460 bytes)]
 [Frame: 21, payload: 2100-3559 (1460 bytes)]
 [Frame: 22, payload: 3560-5019 (1460 bytes)]
 [Frame: 23, payload: 5020-6479 (1460 bytes)]
 [Frame: 24, payload: 6480-7939 (1460 bytes)]
 [Frame: 25, payload: 7940-9399 (1460 bytes)]

Time	Source	Destination	Protocol	Length	Info
24	2.497320	172.20.0.59	TCP	1514	5827 → 80 [ACK] Seq=6481 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (6400 - 7939) = 6481:7939]
25	2.497324	172.20.0.59	TCP	1514	5827 → 80 [ACK] Seq=7941 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (7940 - 9399) = 7941:9399]
26	2.497328	172.20.0.59	TCP	1514	5827 → 80 [ACK] Seq=9401 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (9400 - 10859) = 9401:10859]

Destination Port: 80
 [Stream index: 3]
 [TCP Segment Len: 1460]
 Sequence number: 6481 (relative sequence number)
 [Next sequence number: 7941 (relative sequence number)]
 Acknowledgment number: 1 (relative ack number)
 0101 = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
 Window size value: 1024
 [Calculated window size: 262144]
 [Window size scaling factor: 256]
 Checksum: 0x5c17 [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 > [SEQ/ACK analysis]
 > [Timestamps]
 [Time since first frame in this TCP stream: 0.244125000 seconds]
 [Time since previous frame in this TCP stream: 0.00004000 seconds]

Time	Source	Destination	Protocol	Length	Info
43	2.742103	172.20.0.59	TCP	1514	5827 → 80 [ACK] Seq=19621 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (19425 - 19845) = 19621:19845]
44	2.742241	128.119.245.12	TCP	60	80 → 5827 [ACK] Seq=1 Ack=7941 Win=45184 Len=0
45	2.742241	128.119.245.12	TCP	60	80 → 5827 [ACK] Seq=1 Ack=9401 Win=48000 Len=0
46	2.742288	172.20.0.59	TCP	1514	5827 → 80 [ACK] Seq=21081 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembled TCP segment (20881 - 21281) = 21081:21281]

Transmission Control Protocol, Src Port: 80, Dst Port: 5827, Seq: 1, Ack: 7941, Len: 0
 Source Port: 80
 Destination Port: 5827
 [Stream index: 3]
 [TCP Segment Len: 0]
 Sequence number: 1 (relative sequence number)
 [Next sequence number: 1 (relative sequence number)]
 Acknowledgment number: 7941 (relative ack number)
 0101 = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
 Window size value: 353
 [Calculated window size: 45184]
 [Window size scaling factor: 128]
 Checksum: 0x8691 [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 > [SEQ/ACK analysis]
 > [Timestamps]
 [Time since first frame in this TCP stream: 0.489046000 seconds]
 [Time since previous frame in this TCP stream: 0.000138000 seconds]

如上图，TCP 连接上的第六个报文段的序号是 6481，在 0.244125s 发送，该报文段所对应的 ACK 是在 0.489046s 接收。

(8) 前六个 TCP 报文段的长度各是多少？

```
[106 Reassembled TCP Segments (152965 bytes): #17]
[Frame: 17, payload: 0-639 (640 bytes)]
[Frame: 20, payload: 640-2099 (1460 bytes)]
[Frame: 21, payload: 2100-3559 (1460 bytes)]
[Frame: 22, payload: 3560-5019 (1460 bytes)]
[Frame: 23, payload: 5020-6479 (1460 bytes)]
[Frame: 24, payload: 6480-7939 (1460 bytes)]
```

如上图，前六个 TCP 报文段的长度分别为 640、1460、1460、1460、1460、1460 字节。

(9) 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？

```
13 2.496755 128.119.245.12 172.20.0.59 TCP 66 80 → 5827 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS
14 2.496849 172.20.0.59 65.52.171.231 TCP 54 5829 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0

1000 .... = Header Length: 32 bytes (8)
> Flags: 0x012 (SYN, ACK)
Window size value: 29200
[Calculated window size: 29200]
```

如上图，接收端公示的最小的可用缓存空间是 29200。限制发送端的传输以后，接收端的缓存够用。

(10) 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，因为客户端发送的报文序列号没有出现重复。

(11) TCP 连接的 throughput(bytes transferred per unit time)是多少？请写出你的计算过程。

```
▼ [106 Reassembled TCP Segments (152965 bytes): #17(640),
[Frame: 17, payload: 0-639 (640 bytes)]
[Frame: 20, payload: 640-2099 (1460 bytes)]
[Frame: 21, payload: 2100-3559 (1460 bytes)]
[Frame: 22, payload: 3560-5019 (1460 bytes)]
[Frame: 23, payload: 5020-6479 (1460 bytes)]
[Frame: 24, payload: 6480-7939 (1460 bytes)]
[Frame: 25, payload: 7940-9399 (1460 bytes)]
[Frame: 26, payload: 9400-10859 (1460 bytes)]
```

```
20 2.497299 172.20.0.59 128.119.245.12 TCP 1514 5827 → 80 [ACK] Seq=641 Ack=1 Win=262144 Len=1460 [TCP segment of
21 2.497303 172.20.0.59 128.119.245.12 TCP 1514 5827 → 80 [ACK] Seq=641 Ack=1 Win=262144 Len=1460 [TCP segment of
Sequence number: 641 (relative sequence number)
[Next sequence number: 2101 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
Window size value: 1024
[Calculated window size: 262144]
[Window size scaling factor: 256]
Checksum: 0x8609 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
> [SEQ/ACK analysis]
▼ [Timestamps]
[Time since first frame in this TCP stream: 0.244104000 seconds]
[Time since previous frame in this TCP stream: 0.000206000 seconds]
```

202	3.480233	172.20.0.59	128.119.245.12	TCP	54	5827 → 80 [ACK] Seq=152966 Ack=778 Win=261120 Len=0
Destination Port: 80 [Stream index: 3] [TCP Segment Len: 0] Sequence number: 152966 (relative sequence number) [Next sequence number: 152966 (relative sequence number)] Acknowledgment number: 778 (relative ack number) 0101 = Header Length: 20 bytes (5) > Flags: 0x010 (ACK) Window size value: 1020 [Calculated window size: 261120] [Window size scaling factor: 256] Checksum: 0x4a6a [unverified] [Checksum Status: Unverified] Urgent pointer: 0 > [SEQ/ACK analysis] > [Timestamps] [Time since first frame in this TCP stream: 1.227038000 seconds] [Time since previous frame in this TCP stream: 0.000158000 seconds]						

如上图，发送数据的总长度为 $152965\text{B} + 106 \times 54\text{B} = 158689\text{B}$

发送时间间隔约为 $1.227038\text{s} - 0.244104\text{s} = 0.982934\text{s}$

因此吞吐量为 $158689\text{B} / 0.982934\text{s} = 1291554\text{bps}$

4.4 分析 IP 协议

4.4.2 捕获数据包，选择第一个你本机发出的 ICMP Echo Request 消息，在 packet details 窗口展开数据包的 Internet Protocol 部分，回答(1)-(5)题。

No.	Time	Source	Destination	Protocol	Length	Info
25	6.873472	172.20.0.59	61.167.60.70	ICMP	70	Echo (ping) request id=0x0001, seq=845/19715, ttl=128 (reply in 26)
26	6.875221	61.167.60.70	172.20.0.59	ICMP	70	Echo (ping) reply id=0x0001, seq=845/19715, ttl=249 (request in 25)
27	6.878031	172.20.0.59	61.167.60.70	TCP	70	Echo (ping) request id=0x0001, seq=845/19715, ttl=128 (no response found)
> Frame 25: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0 > Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3) > Internet Protocol Version 4, Src: 172.20.0.59, 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: 0x2f65 (12133) > 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: 128 Protocol: ICMP (1) Header checksum: 0xe523 [validation disabled] [Header checksum status: Unverified] Source: 172.20.0.59 Destination: 61.167.60.70 > Internet Control Message Protocol						
0010 00 38 2f 65 00 00 00 01 e5 23 ac 14 00 3d a7 -8/e... -#...;=						

(1) 你主机的 IP 地址是什么？

如上图，本机的 IP 地址是 172.20.0.59。

(2) 在 IP 数据包头中，上层协议（upperlayer）字段的值是什么？

如上图，Protocol 为 ICMP(1)，值为 01。

(3) 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？

如上图，IP 头有 20 字节，IP 数据包的总大小为 56 字节。

所以 IP 数据包的净载为总大小减去头部的 20 字节，可得净载为 36 字节。

(4) 该 IP 数据包分片了吗？解释你是如何确定该 IP 数据包是否进行了分片

如上图，IP 数据包没有分片。因为 Fragment offset 的值为 0，More fragments 的值为 0，表示后面并没有分段。

4.4.2 单击 Source 列按钮，这样将对捕获的数据包按源 IP 地址排序。选择第一个你的主机发出的 ICMP Echo Request 消息，在 packet details 窗口展开数据包的 Internet Protocol 部分。在“listing of captured packets”窗口，你会看到许多后续的 ICMP 消息（或许还有你主机上运行的其他协议的数据包）。回答(6)-(8)题。

(5) 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？

IP 数据报中总改变的是 Header Checksum、Identification 和 Time to live 字段。

(6) 哪些字段必须保持常量？哪些字段必须改变？为什么？

必须保持常量的字段：

- ① Version 字段：IP 的版本号通常都采用 IPv4。
- ② Header Length 字段：因为这些都是 ICMP 报文。
- ③ Source IP 字段：因为数据都是从自己的主机发送。
- ④ Destination IP 字段：因为数据都是发送向同一主机 www.hit.edu.cn。
- ⑤ Differentiated Services Field 字段：因为所有的包都是 ICMP 包，他们使用相同服务。

⑥ Protocol 字段：因为这些都是 ICMP 数据报。

必须改变的字段有：

- ① Header Checksum 字段：头部存在一直再变的字段，因而校验和一定变。
- ② Identification 字段：IP 数据包有不同的 ID。
- ③ Time to live 字段：来自 traceroute 的要求，用来测试路径上的路由信息。

(7) 描述你看到的 IP 数据包 Identification 字段值的形式。

Identification: 0x2f65 (12133)

如上图，Identification 字段值为 16 位格式且每次递增 1。

4.4.3 找到由最近的路由器(第一跳)返回给你主机的 ICMP Time-to-live exceeded 消息, 回答(9)-(10)题。

(8) Identification 字段和 TTL 字段的值是什么?

No.	Time	Source	Destination	Protocol	Length	Info
567	51.038097	163.177.68.200	172.20.0.59	TCP	60	[TCP Keep-Alive ACK] 443 → 8597 [ACK] Seq=33 Ack=1 Win=138 Len=0
28	6.925655	172.20.0.1	172.20.0.59	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
36	6.978941	172.20.0.1	172.20.0.59	ICMP	120	Destination unreachable (Port unreachable)
69	9.383530	172.20.0.1	172.20.0.59	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)

> Frame 28: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0 > Ethernet II, Src: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3), Dst: IntelCor_58:23:75 (58:fb:84:58:23:75) > Internet Protocol Version 4, Src: 172.20.0.1, Dst: 172.20.0.59 > 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT) Total Length: 84 Identification: 0x1b28 (6952) Flags: 0x0000 Time to live: 64 Protocol: ICMP (1) Header checksum: 0x065d [validation disabled] [Header checksum status: Unverified] Source: 172.20.0.1 Destination: 172.20.0.59 > Internet Control Message Protocol	
---	--

如上图, Identification 字段为 0x1b28, TTL 字段值为 64。

(9) 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?

Identification 字段改变, 因为不同的 IP 数据报, ID 不同。

TTL 字段不变, 因为是同一个路由器, 在一段时间内设置不变, 初始的 TTL 不变, 经过一跳之后也不变。

4.4.4 单击 Time 列按钮, 对捕获的数据包按时间排序。找到在将包大小改为 2000 字节后你的主机发送的第一个 ICMP Echo Request 消息。回答(11)-(12)题。

(10)该消息是否被分解成不止一个 IP 数据报?

188	21.833806	172.20.0.59	61.167.60.70	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=2f95)
189	21.833822	172.20.0.59	61.167.60.70	ICMP	534	Echo (ping) request id=0x0001, seq=893/32003, ttl=128
190	21.836037	61.167.60.70	172.20.0.59	ICMP	1514	Echo (ping) reply id=0x0001, seq=893/32003, ttl=249

> Frame 189: 534 bytes on wire (4272 bits), 534 bytes captured (4272 bits) on interface 0 > Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3) > Internet Protocol Version 4, Src: 172.20.0.59, 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: 520 Identification: 0x2f95 (12181) Flags: 0x00b9 Time to live: 128 Protocol: ICMP (1) Header checksum: 0xe26a [validation disabled] [Header checksum status: Unverified] Source: 172.20.0.59 Destination: 61.167.60.70 > [2 IPv4 Fragments (1980 bytes): #188(1480), #189(500)] [Frame: 188, payload: 0-1479 (1480 bytes)] [Frame: 189, payload: 1480-1979 (500 bytes)] [Fragment count: 2]	
---	--

如上图, 该消息被分解成不止一个 IP 数据报。

(11) 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？

188	21.833806	172.20.0.59	61.167.60.70	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=2f95)
189	21.833822	172.20.0.59	61.167.60.70	ICMP	534	Echo (ping) request id=0x0001, seq=893/32003, ttl=128
190	21.836037	61.167.60.70	172.20.0.59	ICMP	1514	Echo (ping) reply id=0x0001, seq=893/32003, ttl=249

Internet Protocol Version 4, Src: 172.20.0.59, 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: 0x2f95 (12181)	
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: 128	
Protocol: ICMP (1)	
Header checksum: 0xbfd4f [validation disabled]	
[Header checksum status: Unverified]	
Source: 172.20.0.59	
Destination: 61.167.60.70	
Reassembled IPv4 in frame: 189	
> Data (1480 bytes)	

如上图，MF=1 表示已经分片，且不是最后一块。该分片的长度为 1500B。

4.4.5 找到在将包大小改为 3500 字节后你的主机发送的第一个 ICMP Echo Request 消息。回答(13)-(14)题。

(12) 原始数据包被分成了多少片？

No.	Time	Source	Destination	Protocol	Length	Info
716	64.339062	172.20.0.59	61.167.60.70	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=301d)
717	64.339068	172.20.0.59	61.167.60.70	ICMP	554	Echo (ping) request id=0x0001, seq=1029/1284, ttl=128
718	64.341488	61.167.60.70	172.20.0.59	ICMP	1514	Echo (ping) reply id=0x0001, seq=1029/1284, ttl=249

Internet Protocol Version 4, Src: 172.20.0.59, 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: 540	
Identification: 0x301d (12317)	
> Flags: 0x0172	
Time to live: 128	
Protocol: ICMP (1)	
Header checksum: 0xe115 [validation disabled]	
[Header checksum status: Unverified]	
Source: 172.20.0.59	
Destination: 61.167.60.70	
> [3 IPv4 Fragments (3480 bytes): #715(1480), #716(1480), #717(520)]	
[Frame: 715, payload: 0-1479 (1480 bytes)]	
[Frame: 716, payload: 1480-2959 (1480 bytes)]	
[Frame: 717, payload: 2960-3479 (520 bytes)]	
[Fragment count: 3]	
[Reassembled IPv4 length: 3480]	
[Reassembled IPv4 data: 0800b45a0001040532334550696e67506c6f74746572342e...]	
> Internet Control Message Protocol	

如上图，原始数据包被分成了 3 片。

(13) 这些分片中 IP 数据报头部哪些字段发生了变化？

715	64.339048	172.20.0.59	61.167.60.70	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=301d) [R
716	64.339062	172.20.0.59	61.167.60.70	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=301d)
717	64.339068	172.20.0.59	61.167.60.70	ICMP	554	Echo (ping) request id=0x0001, seq=1029/1284, ttl=128 (

> Frame 715: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

> Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)

▼ Internet Protocol Version 4, Src: 172.20.0.59, 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: 0x301d (12317)

▼ 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: 128

Protocol: ICMP (1)

Header checksum: 0xbec7 [validation disabled]

[Header checksum status: Unverified]

Source: 172.20.0.59

Destination: 61.167.60.70

> Frame 716: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

> Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)

▼ Internet Protocol Version 4, Src: 172.20.0.59, 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: 0x301d (12317)

▼ Flags: 0x20b9, More fragments

0... .. = Reserved bit: Not set

.0... .. = Don't fragment: Not set

..1... .. = More fragments: Set

...0 0000 1011 1001 = Fragment offset: 185

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0xbe0e [validation disabled]

[Header checksum status: Unverified]

Source: 172.20.0.59

Destination: 61.167.60.70

> Frame 717: 554 bytes on wire (4432 bits), 554 bytes captured (4432 bits) on interface 0

> Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)

▼ Internet Protocol Version 4, Src: 172.20.0.59, 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: 540

Identification: 0x301d (12317)

▼ Flags: 0x0172

0... .. = Reserved bit: Not set

.0... .. = Don't fragment: Not set

..0... .. = More fragments: Not set

...0 0001 0111 0010 = Fragment offset: 370

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0xe115 [validation disabled]

[Header checksum status: Unverified]

Source: 172.20.0.59

Destination: 61.167.60.70

如上图, IP 数据报头部 Total length、More Fragments、Fragment offset 和 Header Checksum 字段发生了变化。

4.5 抓取 ARP 数据包

- (1) 利用 MS-DOS 命令：arp 或 c:\windows\system32\arp 查看主机上 ARP 缓存的内容。说明 ARP 缓存中每一列的含义是什么？

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

PS C:\WINDOWS\system32> arp -a

接口: 172.20.0.59 --- 0xf
Internet 地址      物理地址      类型
172.20.0.1         58-69-6c-a5-e2-d3 动态
172.20.0.186       58-69-6c-a5-e2-d3 动态
172.20.1.75        58-69-6c-a5-e2-d3 动态
172.20.7.67        58-69-6c-a5-e2-d3 动态
172.20.9.198       58-69-6c-a5-e2-d3 动态
172.20.10.33       58-69-6c-a5-e2-d3 动态
172.20.13.16       58-69-6c-a5-e2-d3 动态
172.20.14.154      58-69-6c-a5-e2-d3 动态
172.20.15.28       58-69-6c-a5-e2-d3 动态
172.20.15.192      58-69-6c-a5-e2-d3 动态
```

如上图，ARP 缓存中的每一列分别表示 IP 地址和其所对应的物理地址以及类型（动态配置或静态配置）。

“arp -d”命令清除主机上 ARP 缓存的内容，抓取 ping 命令时的数据包。分析数据包，回答下面(2)~(4)题：

- (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？



如上图，ARP 数据包由 9 部分构成，分别是硬件类型（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）。

- (3) 如何判断一个 ARP 数据是请求包还是应答包？

```
3280 261.764611 IntelCor_58:23:75 RuijieNe_a5:e2:d3 ARP 42 Who has 172.20.0.1? Tell 172.20.0.59
3281 261.766762 RuijieNe_a5:e2:d3 IntelCor_58:23:75 ARP 60 172.20.0.1 is at 58:69:6c:a5:e2:d3

> Frame 3280: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
v 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_58:23:75 (58:fb:84:58:23:75)
  Sender IP address: 172.20.0.59
  Target MAC address: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
  Target IP address: 172.20.0.1
```

通过 OP 字段。当 OP 字段值为 0x0001 时是请求包（如上图），当 OP 字段值为 0x0002 时是应答包（如下图）。

3280	261.764611	IntelCor_58:23:75	RuijieNe_a5:e2:d3	ARP	42 Who has 172.20.0.1? Tell 172.20.0.59
3281	261.766762	RuijieNe_a5:e2:d3	IntelCor_58:23:75	ARP	60 172.20.0.1 is at 58:69:6c:a5:e2:d3

Frame 3281: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0	
Ethernet II, Src: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3), Dst: IntelCor_58:23:75 (58:fb:84:58:23:75)	
Address Resolution Protocol (reply)	
Hardware type: Ethernet (1)	
Protocol type: IPv4 (0x0800)	
Hardware size: 6	
Protocol size: 4	
Opcode: reply (2)	
Sender MAC address: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)	
Sender IP address: 172.20.0.1	
Target MAC address: IntelCor_58:23:75 (58:fb:84:58:23:75)	
Target IP address: 172.20.0.59	

- (4) 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？

因为主机进行 ARP 查询时并不知道目的 IP 地址对应的 MAC 地址，所以需要进行广播。

ARP 响应时通过查询主机发出的查询报文已经获得了查询主机的 MAC 地址，且局域网中的其他主机并不需要本次查询的结果，所以 ARP 响应要在一个有着明确目的局域网地址的帧中传送。

4.6 抓取 UDP 数据包

- (1) 消息是基于 UDP 的还是 TCP 的？

107	23.039970	172.20.0.59	182.254.79.23	UDP	252 63576 → 8000 Len=210
108	23.112731	182.254.79.23	172.20.0.59	UDP	308 8000 → 63576 Len=266

Frame 107: 252 bytes on wire (2016 bits), 252 bytes captured (2016 bits) on interface 0	
Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)	
Internet Protocol Version 4, Src: 172.20.0.59, Dst: 182.254.79.23	
User Datagram Protocol, Src Port: 63576, Dst Port: 8000	
Data (210 bytes)	

如上图，消息是基于 UDP 的。

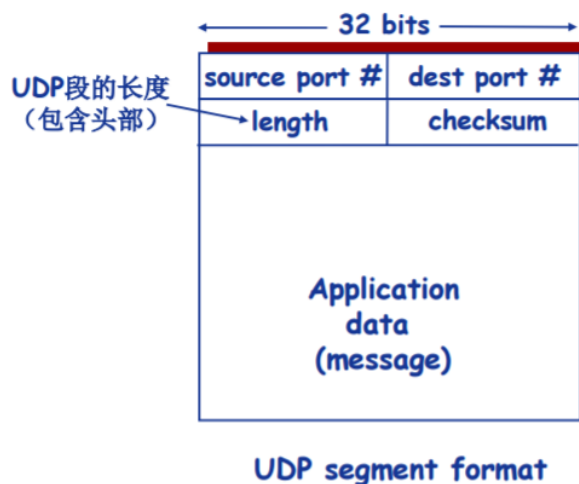
- (2) 你的主机 ip 地址是什么？目的主机 ip 地址是什么？

如上图，我的主机 ip 地址是 172.20.0.59，目的主机 ip 地址是 182.254.79.23。

- (3) 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？

如上图，本机发送 QQ 消息的端口号为 63576，QQ 服务器的端口号为 8000。

(4) 数据报的格式是什么样的？都包含哪些字段，分别占多少字节？



如上图，UDP 数据报格式有首部和数据两个部分。

首部很共 8 字节。包括：

- ① 源端口号：2 字节；
- ② 目的端口号：2 字节；
- ③ 长度：2 字节，UDP 用户数据报的总长度，以字节为单位；
- ④ 校验和：2 字节，用于校验 UDP 数据报的数字段和包含 UDP 数据报首部的“伪首部”。其校验方法与 IP 分组首部的校验方法相同。

(5) 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？

① 服务器返回一个 ICQ 数据包用于通知发送方数据包已经收到。

② 因为 UDP 是不可靠的数据传输，通过这种机制可以确保数据的正确传送，相当于 ACK 报文的作用。但服务器只提供了一次返回的 ACK，因此不保证数据一定完全正确送达。

③ 可以看出 UDP 是无连接的。因为 UDP 在数据传输时没有连接的建立与断开过程，不能像 TCP 协议那样先握手再发送数据。另外，UDP 每次只发送一个数据报，然后等待服务器响应，不像 TCP 协议那样分组发送。

4.7 分析 DNS 协议

打开浏览器，输入 www.baidu.com，用 Wireshark 捕获的 DNS 报文。

No.	Time	Source	Destination	Protocol	Length	Info
12	4.104011	172.20.0.59	202.118.224.101	DNS	73	Standard query 0x42ae A www.baidu.com
13	4.104312	172.20.0.59	202.118.224.101	DNS	73	Standard query 0xc74a AAAA www.baidu.com
14	4.106923	202.118.224.101	172.20.0.59	DNS	132	Standard query response 0x42ae A www.baidu.com
15	4.107170	202.118.224.101	172.20.0.59	DNS	157	Standard query response 0xc74a AAAA www.baidu.com

> Frame 12: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0

> Ethernet II, Src: IntelCor_58:23:75 (58:fb:84:58:23:75), Dst: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)

> Internet Protocol Version 4, Src: 172.20.0.59, Dst: 202.118.224.101

> User Datagram Protocol, Src Port: 50711, Dst Port: 53

> Domain Name System (query)

Transaction ID: 0x42ae

> Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

> Queries

> www.baidu.com: type A, class IN

Name: www.baidu.com

[Name Length: 13]

[Label Count: 3]

Type: A (Host Address) (1)

Class: IN (0x0001)

[Response In: 14]

上图为 DNS 查询报文。

可知，本机 IP 地址为 172.20.0.59，本地域名服务器 IP 地址为 202.118.224.101。
报文的源端口号 50711，目的端口号 53。

No.	Time	Source	Destination	Protocol	Length	Info
12	4.104011	172.20.0.59	202.118.224.101	DNS	73	Standard query 0x42ae A www.baidu.com
13	4.104312	172.20.0.59	202.118.224.101	DNS	73	Standard query 0xc74a AAAA www.baidu.com
14	4.106923	202.118.224.101	172.20.0.59	DNS	132	Standard query response 0x42ae A www.baidu.com CNAME www.a.shifen.com A ..
15	4.107170	202.118.224.101	172.20.0.59	DNS	157	Standard query response 0xc74a AAAA www.baidu.com CNAME www.a.shifen.com ..

> Frame 14: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface 0

> Ethernet II, Src: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3), Dst: IntelCor_58:23:75 (58:fb:84:58:23:75)

> Internet Protocol Version 4, Src: 202.118.224.101, Dst: 172.20.0.59

> User Datagram Protocol, Src Port: 53, Dst Port: 50711

> Domain Name System (response)

Transaction ID: 0x42ae

> 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

> www.baidu.com: type CNAME, class IN, cname www.a.shifen.com

> www.a.shifen.com: type A, class IN, addr 119.75.217.109

> www.a.shifen.com: type A, class IN, addr 119.75.217.26

[Request In: 12]

[Time: 0.002912000 seconds]

上图为 DNS 应答报文。

由上图可知 www.baidu.com 的 ip 地址为 119.75.217.26 或 119.75.217.109。

4.8 分析 Ethernet 数据帧

俘获窗口内容如下：

> Ethernet II, Src: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3), Dst: IntelCor_58:23:75 (58:fb:84:58:23:75)
> Destination: IntelCor_58:23:75 (58:fb:84:58:23:75)
Address: IntelCor_58:23:75 (58:fb:84:58:23:75)
.... 0. = LG bit: Globally unique address (factory default)
.... 0. = IG bit: Individual address (unicast)
> Source: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
Address: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
.... 0. = LG bit: Globally unique address (factory default)
.... 0. = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

由上图可知,该 Ethernet 数据帧源 MAC 地址为 58:69:6c:a5:e2:d3,目的 MAC 地址为 58:fb:84:58:23:75, 且可知帧中封装的是 IPv4 协议的分组。

5 心得体会

通过本次实验,我熟悉并掌握了 Wireshark 的基本操作,了解了网络协议实体间进行交互以及报文交换的情况,并且对 TCP、UDP、IP、ARP、DNS 等协议的原理有了更深刻的认识。

纸上得来终觉浅,绝知此事要躬行。通过实验也使得我对该部分的理论内容加深了理解,一些原来理解有偏差的点得到纠正,真正起到了巩固与提升的作用,真正做到了学以致用。