

浙江大学

本科实验报告

课程名称： 计算机网络基础

实验名称： 网络协议分析

姓 名： 郑昱笙

专 业： 地理信息科学

学 号： 3180102760

指导教师： 高艺

2020 年 1 月 14 日

浙江大学实验报告

实验名称： 网络协议分析 实验类型： 分析实验

同组学生： 无 实验地点： 计算机网络实验室

一、 实验目的

- 进一步学习使用 Wireshark 抓包工具。
- 观察和理解常见网络协议的交互过程
- 理解数据包分层结构和格式。

二、 实验内容

- 熟练掌握网络协议分析软件 Wireshark 的使用
- 观察所在网络出现的各类网络协议，了解其种类和分层结构
- 观察捕获到的数据包格式，理解各字段含义
- 根据要求配置 Wireshark，捕获某一类协议的数据包，并分析解读

三、 主要仪器设备

- 联网的 PC 机
- WireShark 协议分析软件

四、 操作方法与实验步骤

- 配置网络包捕获软件，捕获所有机器的数据包
- 观察捕获到的数据包，并对照解析结果和原始数据包
- 配置网络包捕获软件，只捕获特定 IP 或特定类型的包
- 抓取以下通信协议数据包，观察通信过程和数据包格式
 - ✓ PING：测试一个目标地址是否可达（在实验一基础上）
 - ✓ TRACE ROUTE：跟踪一个目标地址的途经路由（在实验一基础上）
 - ✓ NSLOOKUP：查询一个域名（在实验一基础上）
 - ✓ HTTP：访问一个网页
 - ✓ FTP：上传或下载一个文件
 - ✓ SMTP：发送一封邮件
 - ✓ POP3/IMAP：接收一封邮件
 - ✓ RTP：抓取一段音频流

提醒：为了避免捕获到大量无关数据包，影响实验观察，建议关闭所有无关软件。

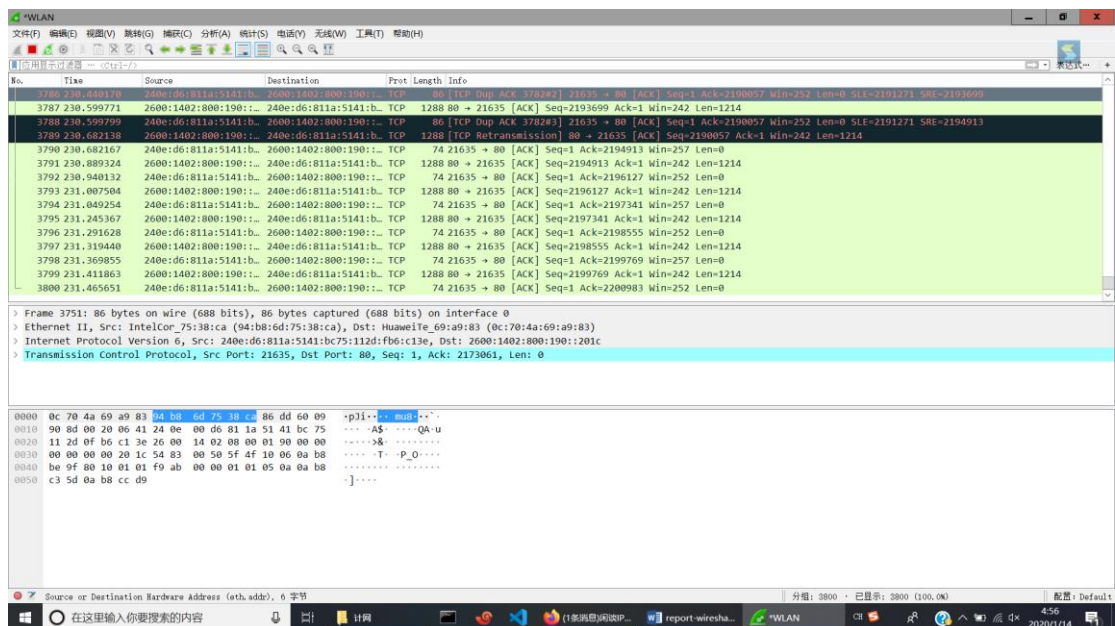
五、 实验数据记录和处理

以下实验记录均需结合屏幕截图，进行文字标注和描述，图片应大小合适、关键部分清

晰可见，可直接在图片上进行标注，也可以单独用文本进行描述。（看完请删除本句）。

✧ Part One

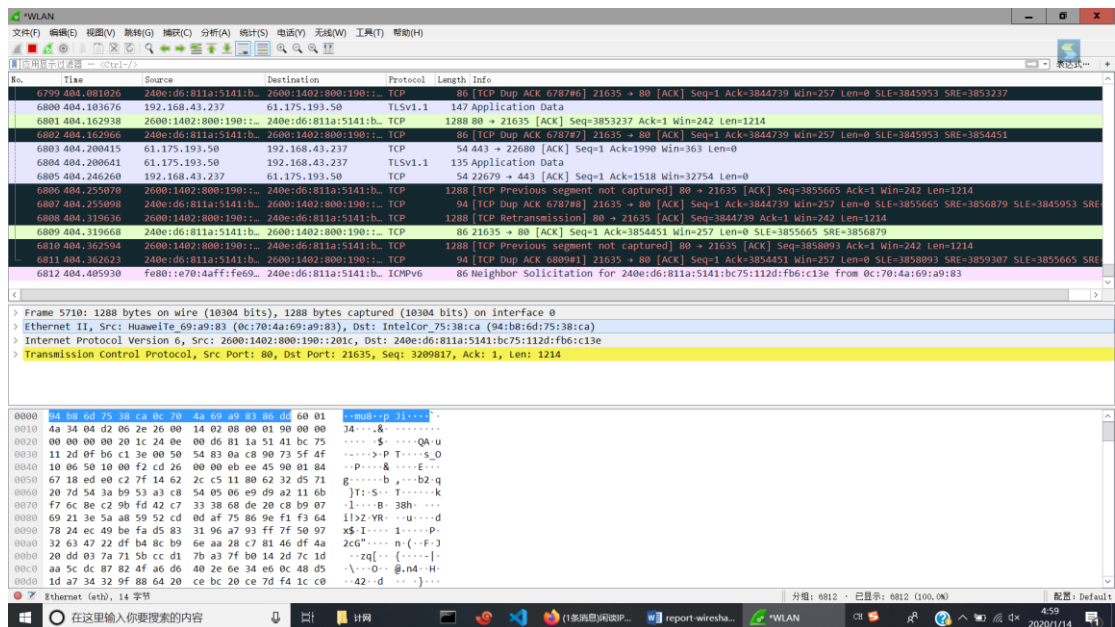
- 打开 WireShark，开始捕获网络数据包后，你看到了什么？有哪些协议？



看到了一大堆乱七八糟不停滚动的数据包列表，以及随后的数据包内容和数据包解析。

有 ICMP、ARP、HTTP、TCP 等协议。

- 找一个包含 Ethernet 的数据包，这是什么协议？标出源和目标 MAC 地址。

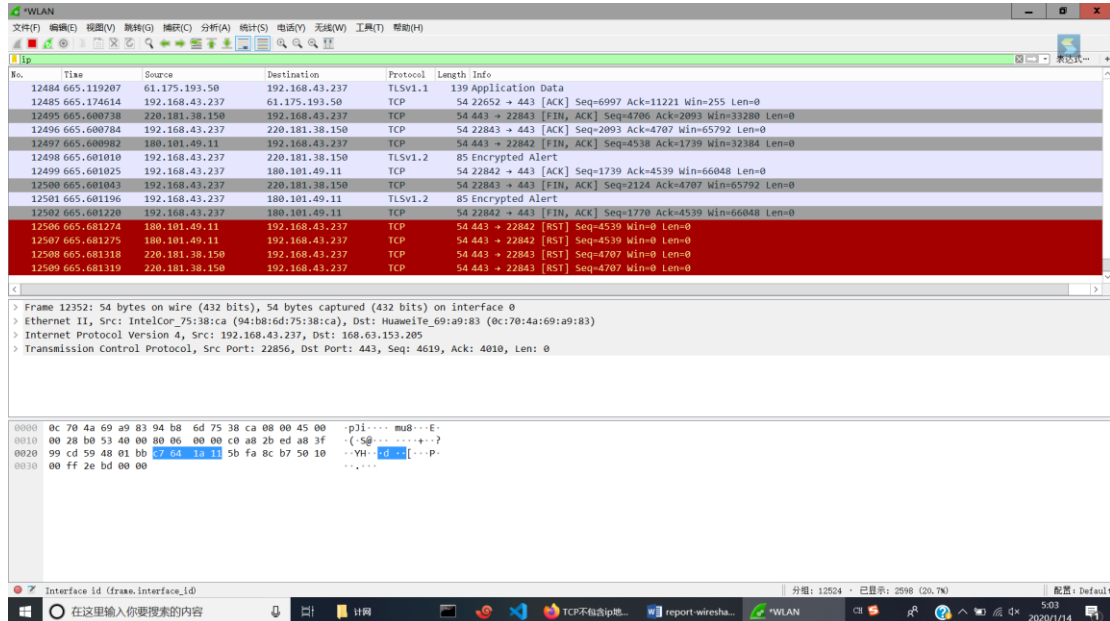


这是 TCP 协议。

源地址和目标地址：

- ▼ Ethernet II, Src: HuaweiTe_69:a9:83 (0c:70:4a:69:a9:83), Dst: IntelCor_75:38:ca (94:b8:6d:75:38:ca)
 - ▼ Destination: IntelCor_75:38:ca (94:b8:6d:75:38:ca)
 - Address: IntelCor_75:38:ca (94:b8:6d:75:38:ca)
 - 0. = LG bit: Globally unique address (factory default)
 - 0. = IG bit: Individual address (unicast)
 - ▼ Source: HuaweiTe_69:a9:83 (0c:70:4a:69:a9:83)
 - Address: HuaweiTe_69:a9:83 (0c:70:4a:69:a9:83)

- 找一个包含 IP 的数据包，这是什么协议？标出源 IP 地址、目标 IP 地址。



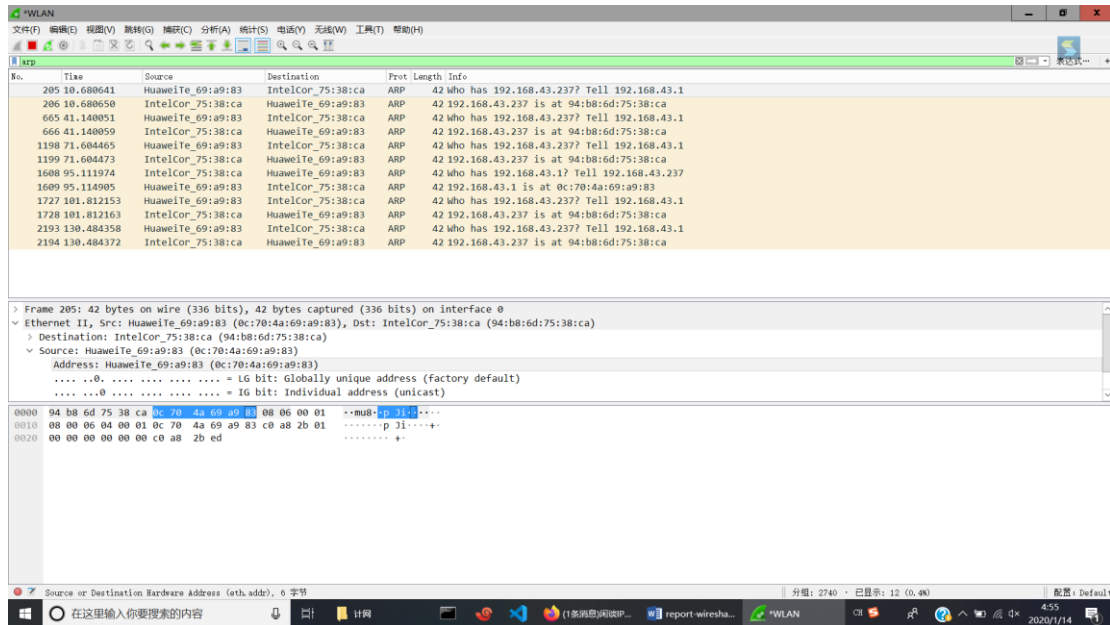
还是 TCP 协议。

源 IP 地址、目标 IP 地址：

- ▼ Internet Protocol Version 4, Src: 192.168.43.237, Dst: 168.63.153.205
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
 - > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 40
 - Identification: 0xb053 (45139)
 - > Flags: 0x4000, Don't fragment
 - Time to live: 128
 - Protocol: TCP (6)
 - Header checksum: 0x0000 [validation disabled]
 - [Header checksum status: Unverified]
 - Source: 192.168.43.237
 - Destination: 168.63.153.205

- 找一个 ARP 数据包，这是请求还是应答？标注发送者的 MAC 地址。

是 ARP 请求。



请在下面的每次捕获任务完成后，保存 Wireshark 抓包记录（.pcap 格式），随报告一起提交。每一个协议一个单独文件，文件名请取得便于理解。

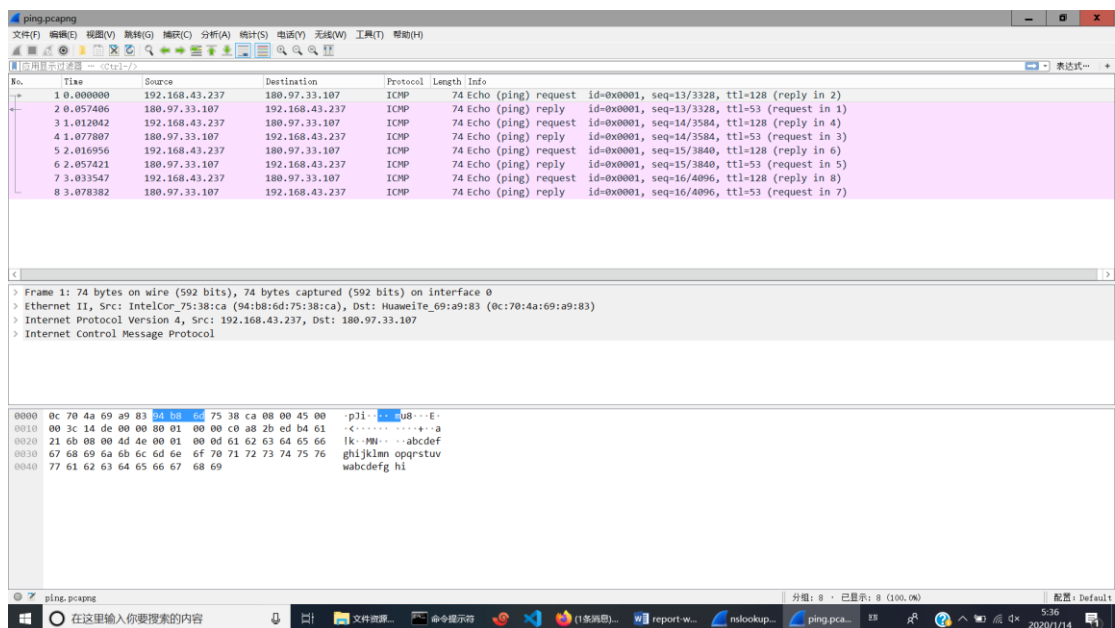
✧ Part Two

- 使用 Ping 命令，测试某个 IP 地址的连通性，并捕获这次的数据包。数据包由几层协议构成？分别是什么协议？选择一个请求包和一个响应包，展开最高层协议的详细内容，标出请求包和应答包、类型、序号。

```
C:\Users\云微>ping 180.97.33.107

正在 Ping 180.97.33.107 具有 32 字节的数据:
来自 180.97.33.107 的回复: 字节=32 时间=49ms TTL=53
来自 180.97.33.107 的回复: 字节=32 时间=79ms TTL=53
来自 180.97.33.107 的回复: 字节=32 时间=42ms TTL=53
来自 180.97.33.107 的回复: 字节=32 时间=41ms TTL=53

180.97.33.107 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 41ms, 最长 = 79ms, 平均 = 52ms
```

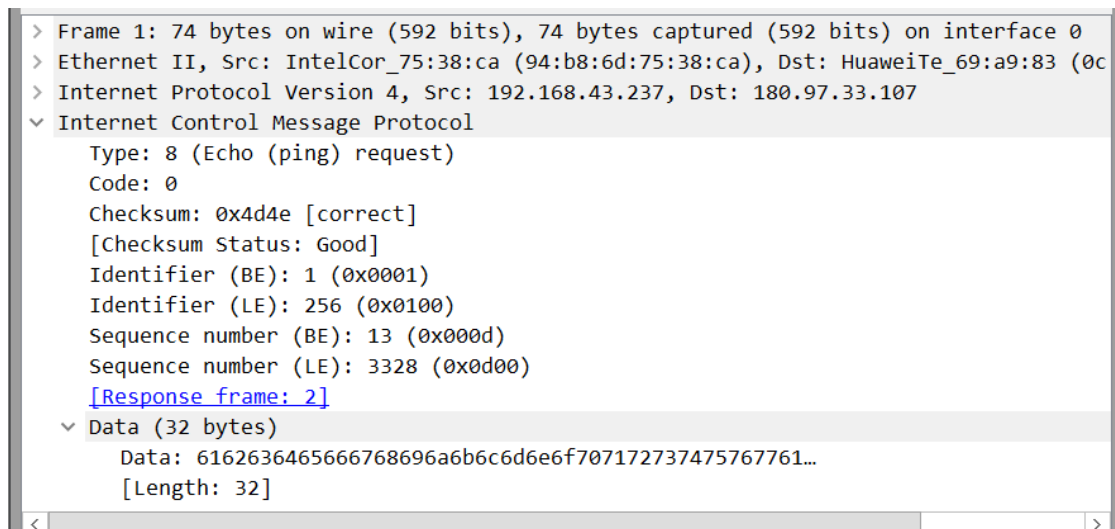


由三层协议构成：以太网协议、ipv4 协议和 icmp 协议

请求包：

Type: 8 (Echo (ping) request)

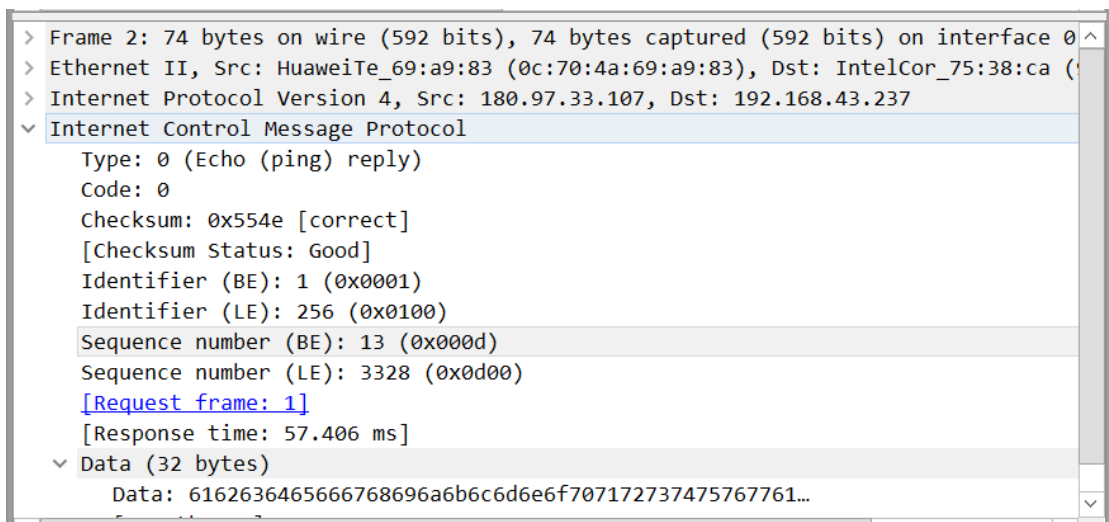
Sequence number (BE): 13 (0x000d)



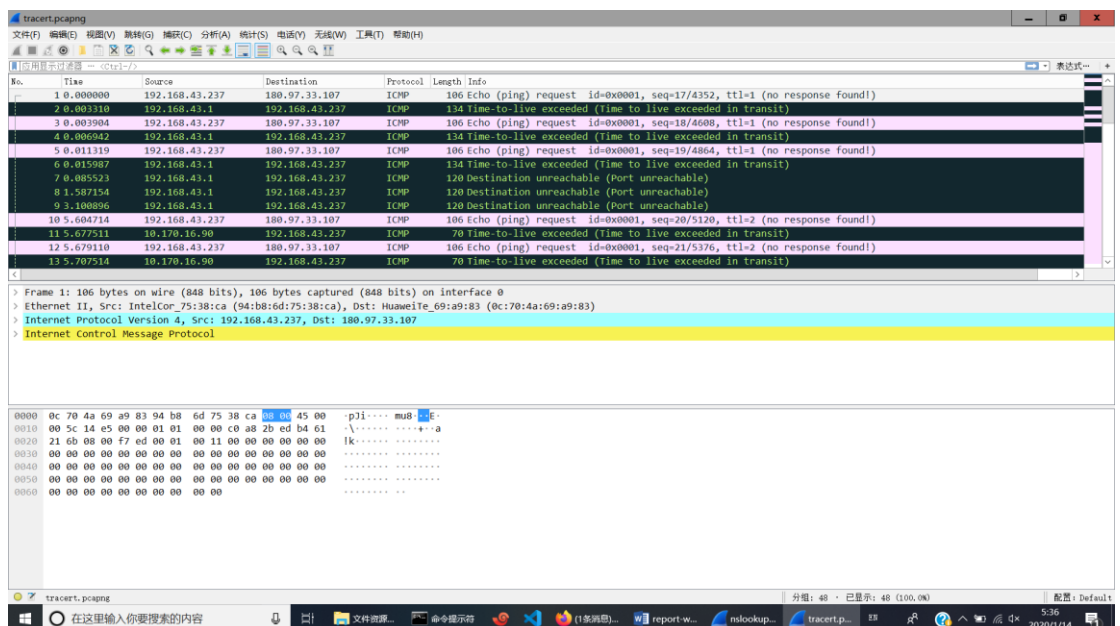
应答包：

Type: 0 (Echo (ping) reply)

Sequence number (BE): 13 (0x000d)



- 使用 Tracert 命令（Mac 下使用 Traceroute 命令），跟踪某个外部 IP 地址的路由，并捕获这次的数据包。数据包由几层协议构成？分别是什么协议？查看并标记多个请求包的 IP 协议层的 TTL 字段，发现了什么规律？选择一个请求包和一个响应包，展开最高层协议的详细内容，标出类型、序号等关键字段。与 Ping 命令的数据包有什么不同？



由三层协议构成：以太网协议、ipv4 协议和 icmp 协议

TTL 字段从 1 开始递增；

请求包：

Type: 8 (Echo (ping) request)

Sequence number (BE): 17 (0x0011)


```

> Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
> Ethernet II, Src: IntelCor_75:38:ca (94:b8:6d:75:38:ca), Dst: HuaweiTe_69:a9:83 (0c
> Internet Protocol Version 4, Src: 192.168.43.237, Dst: 180.97.33.107
  > Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf7ed [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 17 (0x0011)
    Sequence number (LE): 4352 (0x1100)
  > [No response seen]
  > Data (64 bytes)
    Data: 0000000000000000000000000000000000000000000000000000000000000000...
    [Length: 64]

```

应答包:

Type: 11 (Time-to-live exceeded)

Sequence number (BE): 17 (0x0011)

```

  > Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
    Unused: 00000000
  > Internet Protocol Version 4, Src: 192.168.43.237, Dst: 180.97.33.107
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 92
    Identification: 0x14e5 (5349)
  > Flags: 0x0000
  > Time to live: 1
    > [Expert Info (Note/Sequence): "Time To Live" only 1]
    Protocol: ICMP (1)
    Header checksum: 0xe25a [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.43.237
    Destination: 180.97.33.107
  > Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf7ed [unverified] [in ICMP error packet]
    [Checksum Status: Unverified]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 17 (0x0011)
    Sequence number (LE): 4352 (0x1100)
  > Data (64 bytes)

```

和 ping 命令相比, tracert 发送的 ICMP 数据包 data 字段总为 0, 且应答包中包含了 TTL 降为 0 超时的错误信息。

- 使用 `nslookup` 命令，查询某个域名，并捕获这次的数据包。数据包由几层协议构成？分别是什么协议？标记 UDP 协议层的端口字段。选择一个请求包和一个响应包，展开最高层协议的详细内容，标出类型、序号、域名信息。

The first screenshot shows the output of the `nslookup www.baidu.com` command in a Windows command prompt. It displays the server address (192.168.43.1) and a non-authoritative response for the domain `www.a.shifen.com` with multiple IP addresses.

The second screenshot shows a Wireshark packet capture of the DNS query and response. The packet list shows a query (No. 1) and a response (No. 2). The packet details pane shows the Ethernet II, Internet Protocol Version 4, and User Datagram Protocol (UDP) layers. The packet bytes pane shows the raw data of the DNS message.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.237	192.168.43.1	DNS	85	Standard query 0x0001 PTR 1.43.168.192.in-addr.arpa
2	0.004102	192.168.43.1	192.168.43.237	DNS	85	Standard query response 0x0001 No such name PTR 1.43.168.192.in-addr.arpa
3	0.005293	192.168.43.237	192.168.43.1	DNS	73	Standard query 0x0002 A www.baidu.com
4	0.008302	192.168.43.1	192.168.43.237	DNS	199	Standard query response 0x0002 A www.baidu.com CNAME www.a.shifen.com A 220.181.38.149 A 180.97.33.108 A 180.101.49.12 A 180.97.33.107 A 180.101.49.11 A 220.181.38.150
5	0.008862	192.168.43.237	192.168.43.1	DNS	73	Standard query 0x0003 AAAA www.baidu.com
6	0.011604	192.168.43.1	192.168.43.237	DNS	103	Standard query response 0x0003 AAAA www.baidu.com CNAME www.a.shifen.com

Frame 6: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface 0
Ethernet II, Src: HuaweiTe_69:a9:83 (0c:70:4a:69:a9:83), Dst: IntelCor_75:38:ca (94:b8:6d:75:38:ca)
Internet Protocol Version 4, Src: 192.168.43.1, Dst: 192.168.43.237
User Datagram Protocol, Src Port: 53, Dst Port: 61658
Domain Name System (response)

0000 94 b8 6d 75 38 ca 0c 70 4a 69 a9 83 00 00 45 00 - mu8 - p 3l ---- E
0010 00 59 3f cf 40 00 40 11 22 86 c0 a8 2b 01 c0 a8 - v7 @ @ " - - - - -
0020 2b ed 00 35 f0 da 00 45 b3 3d 00 03 81 80 00 01 - + - 5 - - - - E - - - - -
0030 00 01 00 00 00 00 03 77 77 77 05 62 61 69 64 75 - - - - - w w w - baidu
0040 03 63 6f 6d 00 00 1c 00 01 c0 0c 00 05 00 01 00 - - com - - - - -
0050 00 02 02 00 12 03 77 77 77 01 61 06 73 68 69 66 - - - - - w w w - a - shif
0060 65 6e 03 63 6f 6d 00 - en - com -

由四层协议构成，分别是以太网、ipv4、UDP、DNS

UDP 协议层的端口字段：

<		
User Datagram Protocol, Src Port: 53, Dst Port: 61658		
Source Port: 53		
Destination Port: 61658		
Length: 69		
Checksum: 0xb33d [unverified]		
[Checksum Status: Unverified]		
<		
0000	94 b8 6d 75 38 ca 0c 70 4a 69 a9 83 08 00 45 00	..mu8..p Ji...E.
0010	00 59 3f cf 40 00 40 11 22 86 c0 a8 2b 01 c0 a8	.Y?..@.. "....+
0020	2b ed 00 35 f0 da 00 45 b3 3d 00 03 81 80 00 01	+..5...E +=.....
0030	00 01 00 00 00 00 03 77 77 77 05 62 61 69 64 75w ww·baidu
0040	03 63 6f 6d 00 00 1c 00 01 c0 0c 00 05 00 01 00	·com.....
0050	00 02 02 00 12 03 77 77 77 01 61 06 73 68 69 66ww w·a·shif
0060	65 6e 03 63 6f 6d 00	en·com·

请求包:

> User Datagram Protocol, Src Port: 61657, Dst Port: 53	
Domain Name System (query)	
Transaction ID: 0x0002	
Flags: 0x0100 Standard query	
0... .. = Response: Message is a query	
.000 0... .. = Opcode: Standard query (0)	
... ..0. = Truncated: Message is not truncated	
... ..1 = Recursion desired: Do query recursively	
... ..0. = Z: reserved (0)	
... ..0 = Non-authenticated data: Unacceptable	
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: 4]	
> TRANSUM RTE Data	

类型: Flags: 0x0100 Standard query

序号: Transaction ID: 0x0002

域名信息: www.baidu.com: type A, class IN

响应包:

```
Domain Name System (response)
Transaction ID: 0x0002
Flags: 0x8180 Standard query response, No error
  1... .. = Response: Message is a response
  .000 0... .. = Opcode: Standard query (0)
  .... 0... .. = Authoritative: Server is not an authority for domain
  .... ..0... .. = Truncated: Message is not truncated
  .... ..1... .. = Recursion desired: Do query recursively
  .... ..1... .. = Recursion available: Server can do recursive queries
  .... ..0... .. = Z: reserved (0)
  .... ..0... .. = Answer authenticated: Answer/authority portion was not authenticated by
  .... ..0... .. = Non-authenticated data: Unacceptable
  .... ..0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 7
Authority RRs: 0
Additional RRs: 0
> Queries
> Answers
  > www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
  > www.a.shifen.com: type A, class IN, addr 220.181.38.149
  > www.a.shifen.com: type A, class IN, addr 180.97.33.108
  > www.a.shifen.com: type A, class IN, addr 180.101.49.12
  > www.a.shifen.com: type A, class IN, addr 180.97.33.107
  > www.a.shifen.com: type A, class IN, addr 180.101.49.11
  > www.a.shifen.com: type A, class IN, addr 220.181.38.150
[Request In: 3]
[Time: 0.003009000 seconds]
```

类型: Flags: 0x8180 Standard query response, No error

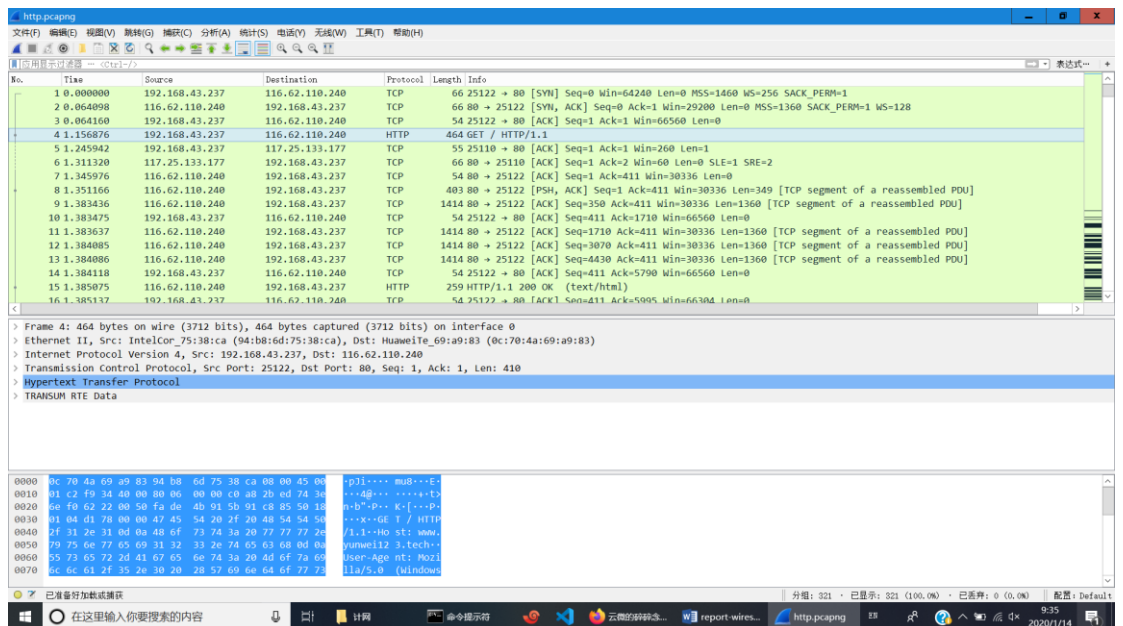
序号: Transaction ID: 0x0002

域名信息:

```
Answers
> www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
> www.a.shifen.com: type A, class IN, addr 220.181.38.149
> www.a.shifen.com: type A, class IN, addr 180.97.33.108
> www.a.shifen.com: type A, class IN, addr 180.101.49.12
> www.a.shifen.com: type A, class IN, addr 180.97.33.107
> www.a.shifen.com: type A, class IN, addr 180.101.49.11
> www.a.shifen.com: type A, class IN, addr 220.181.38.150
[Request In: 3]
```

✧ Part Three

- 运行 `ipconfig /flushdns` 命令清空 DNS 缓存, 然后打开浏览器, 访问一个网页, 并捕获这次的数据包 (网页完全打开后, 停止捕获)。数据包由几层协议构成? 分别是什么协议? 标出数据包的源和目标 IP 地址、源和目标端口。



由四层协议构成，分别以太网、ipv4、TCP HTTP

Source Port: 25122

Destination Port: 80

Source: 192.168.43.237

Destination: 116.62.110.240

- 找到建立 TCP 连接的三个数据包（称为三次握手），展开 TCP 协议层的 Flags 字段，分别标记三个数据包的 SYN 标志位和 ACK 标志位。

第一次：

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.237	116.62.110.240	TCP	66	25122 → 80
2	0.064098	116.62.110.240	192.168.43.237	TCP	66	80 → 25122
3	0.064160	192.168.43.237	116.62.110.240	TCP	54	25122 → 80
< >						
> Ethernet II, Src: IntelCor_75:38:ca (94:b8:6d:75:38:ca), Dst: HuaweiTe_69:a9:83 (0c:70:42:69:a9:83) > Internet Protocol Version 4, Src: 192.168.43.237, Dst: 116.62.110.240 > Transmission Control Protocol, Src Port: 25122, Dst Port: 80, Seq: 0, Len: 0						
Source Port: 25122 Destination Port: 80 [Stream index: 0] [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) <ul style="list-style-type: none"> 000. = Reserved: Not set ...0 = Nonce: Not set 0... = Congestion Window Reduced (CWR): Not set 0.. = ECN-Echo: Not set0. = Urgent: Not set0 = Acknowledgment: Not set 0... = Push: Not set0.. = Reset: Not set1. = Syn: Set > [Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]0 = Fin: Not set [TCP Flags:S.] Window size value: 64240 [Calculated window size: 64240]						
< >						
0010	00 34 f9 32 40 00 80 06	00 00 c0 a8 2b ed 74 3e	.4.2@... ..+t>			
0020	6e f0 62 22 00 50 fa de	4b 90 00 00 00 00 80 02	n·b"·P·· K.....			
0030	fa f0 cf ea 00 00 02 04	05 b4 01 03 03 08 01 01			
0040	04 02		..			

第二次

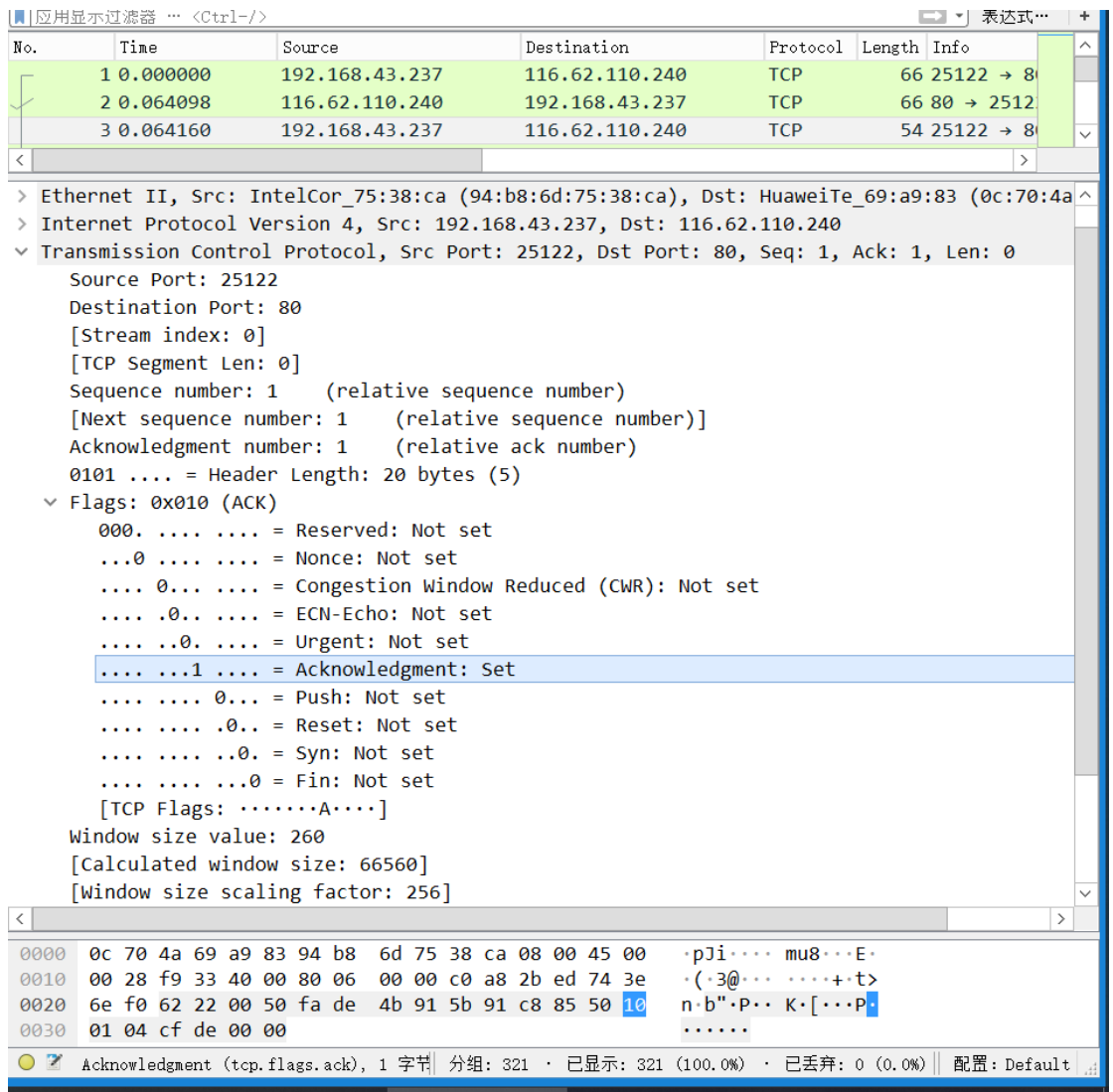
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.43.237	116.62.110.240	TCP	66	25122 → 80
2	0.064098	116.62.110.240	192.168.43.237	TCP	66	80 → 25122
3	0.064160	192.168.43.237	116.62.110.240	TCP	54	25122 → 80

>	Ethernet II, Src: HuaweiTe_69:a9:83 (0c:70:4a:69:a9:83), Dst: IntelCor_75:38:ca (94:b8:6d:75:38:ca)
>	Internet Protocol Version 4, Src: 116.62.110.240, Dst: 192.168.43.237
▼	Transmission Control Protocol, Src Port: 80, Dst Port: 25122, Seq: 0, Ack: 1, Len: 0
	Source Port: 80
	Destination Port: 25122
	[Stream index: 0]
	[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)
	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
	> [Expert Info (Chat/Sequence): Connection establish acknowledge (SYN+ACK): server p
0 = Fin: Not set
	[TCP Flags:A..S.]
	Window size value: 29200
	[Calculated window size: 29200]

0010	00 34 00 00 40 00 31 06	7a 00 74 3e 6e f0 c0 a8	4..@.1. z.t>n..
0020	2b ed 00 50 62 22 5b 91	c8 84 fa de 4b 91 80 12	+..Pb"[.K...
0030	72 10 60 98 00 00 02 04	05 50 01 01 04 02 01 03	r.`.....P.....
0040	03 07		..

Acknowledgment (tcp.flags.ack), 1 字节 | 分组: 321 · 已显示: 321 (100.0%) · 已丢弃: 0 (0.0%) | 配置: Default

第三次:



- 选择一个包，点击右键，选择跟踪一个 TCP 流，截取完整的 HTTP 请求消息和部分响应消息，标记 HTTP 请求头部的 Method 字段、URI 字段和 Host 字段，标记 HTTP 响应头部的 Status Code 字段、Content-Type 和 Content-Length 字段，以及区分响应头部和体部的标记（单独的回车换行符）。

Method 字段、URI 字段和 Host 字段：

GET

/wp-content/themes/kratos-pjax-master/static/css/kratos.min.cs

s?ver=0.4.0 HTTP/1.1

Host: www.yunweil23.tech

HTTP 响应头部的 Status Code 字段：

HTTP/1.1 200 OK

Content-Type 和 Content-Length 字段：

Content-Type: text/css

Content-Length: 15522

```
GET /wp-content/themes/kratos-pjax-master/static/css/kratos.min.css?ver=0.4.0 HTTP/1.1
Host: www.yunwei123.tech
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/css,*/*;q=0.1
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.yunwei123.tech/
If-Modified-Since: Sun, 13 Oct 2019 05:12:17 GMT
If-None-Match: "1471d-594c3cb879a41-gzip"
Cache-Control: max-age=0

HTTP/1.1 200 OK
Date: Tue, 14 Jan 2020 01:34:43 GMT
Server: Apache/2.4.25 (Debian)
Last-Modified: Sun, 13 Oct 2019 05:12:17 GMT
ETag: "1471d-594c3cb879a41-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 15522
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/css

.....}...H..{..T..Z^u..v=.....<.....F.....(Q+J..#$......
0.3~....q... 'H*..wm.h.Z"#N.....?.....f .....h.n...D..Z...S{.....e..VO.....WR..Z../.
```

- 使用过滤器 `tcp.stream eq X`，让 `X` 从 0 开始变化，直到没有数据。观察总共捕获到了几个 TCP 连接（一个 TCP 流对应一个 TCP 连接）？存在几个 HTTP 会话（一对 HTTP 请求和响应对应一次 HTTP 会话）？注意：一个 TCP 流上可能存在多个 HTTP 会话。

总共捕获到了 11 个 TCP 连接

`tcp.stream eq 0` 6 个

2 一个

3 2 个

4 一个

5 一个

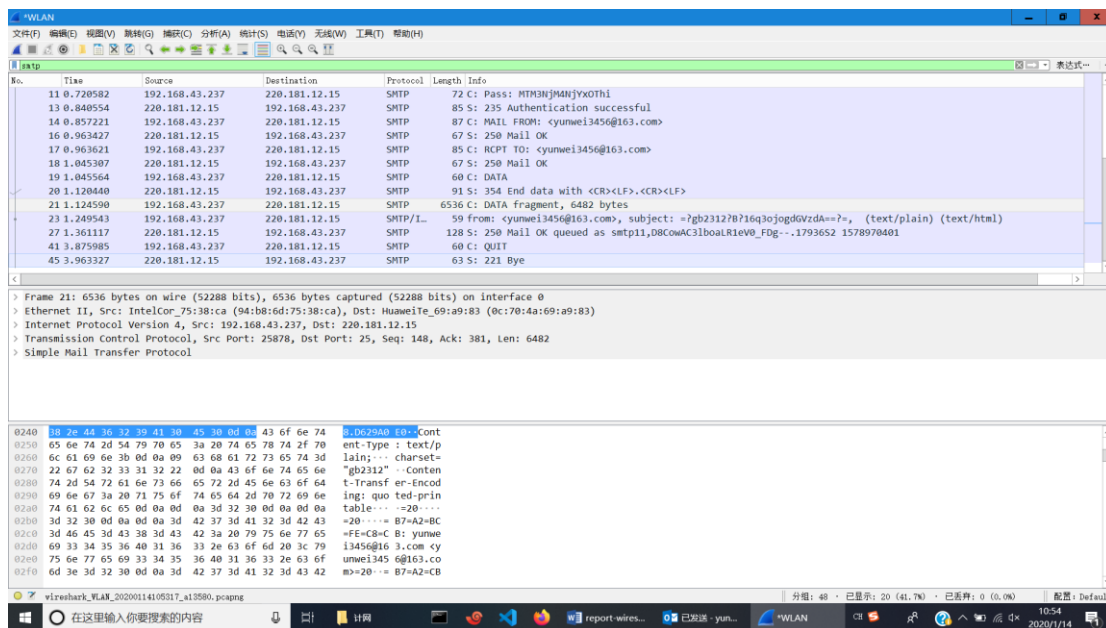
6 一个

总共 12 个

✧ Part Four

- 打开邮件客户端 Foxmail 或 Outlook，写一封电子邮件（建议采用直接送达方式），并捕获这次的数据包。捕获到的数据包由几层协议构成？分别是什么协议？标出

数据包的源和目标 IP 地址、源和目标端口。



由四层协议构成，分别是以太网、ipv4、TCP SMTP

Source: 192.168.43.237

Destination: 220.181.12.15

Source Port: 25878

Destination Port: 25

- 跟踪 TCP 流，查看 SMTP 握手消息采用的是什么（HELO 还是 EHLO）？标出 SMTP 协议层中的客户端机器名、发件人地址、收件人地址、认证的用户名和密码（如果是 EHLO 握手方式）、邮件正文（内容过长可截取关键部分）。

SMTP 握手消息采用的是 EHLO

客户端机器名 LAPTOPFSMLGFU0

发件人地址 MAIL FROM: yunwei3456@163.com

收件人地址 TO: yunwei3456@163.com

认证的用户名和密码：

AUTH LOGIN

334 dXNlcm5hbWU6

eXVud2VpMzQ1NkAxNjMuY29t

334 UGFzc3dvcmQ6

MTM3NjM4NjYxOThi

235 Authentication successful

```
220 163.com Anti-spam GT for Coremail System (163com[20141201])
EHLO LAPTOPFSMLGFU0
250-mail
250-PIPELINING
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-coremail 1Uxr2xKj7kG0xkI17xGrU7I0s8FY2U3Uj8Cz28x1UUUUU7Ic2I0Y2Ur1fA5mUCa0xDrUUUUj
250-STARTTLS
250 8BITIME
AUTH LOGIN
334 dXNlcm5hbWU6
eXVud2VpMzQ1bkAxNjMuY29t
334 UGFzc3dvcmQ6
MTM3NjM4NjYxOTI0
235 Authentication successful
MAIL FROM: <yunwei3456@163.com>
250 Mail OK
RCPT TO: <yunwei3456@163.com>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
From: <yunwei3456@163.com>
To: <yunwei3456@163.com>
References:
In-Reply-To:
Subject: =?gb2312?B?16q3ojogdGVzdA==?=
Date: Tue, 14 Jan 2020 10:53:13 +0800
Message-ID: <002701d5ca85$c805ebb0$5811c310$@163.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="-----_NextPart_000_0028_01D5CAC8.D629A0E0"
X-Mailer: Microsoft Outlook 16.0
Thread-Index: AQINVCN7anIznCkaMAqDBm3kQsDSjqd5kz0A
Content-Language: zh-cn

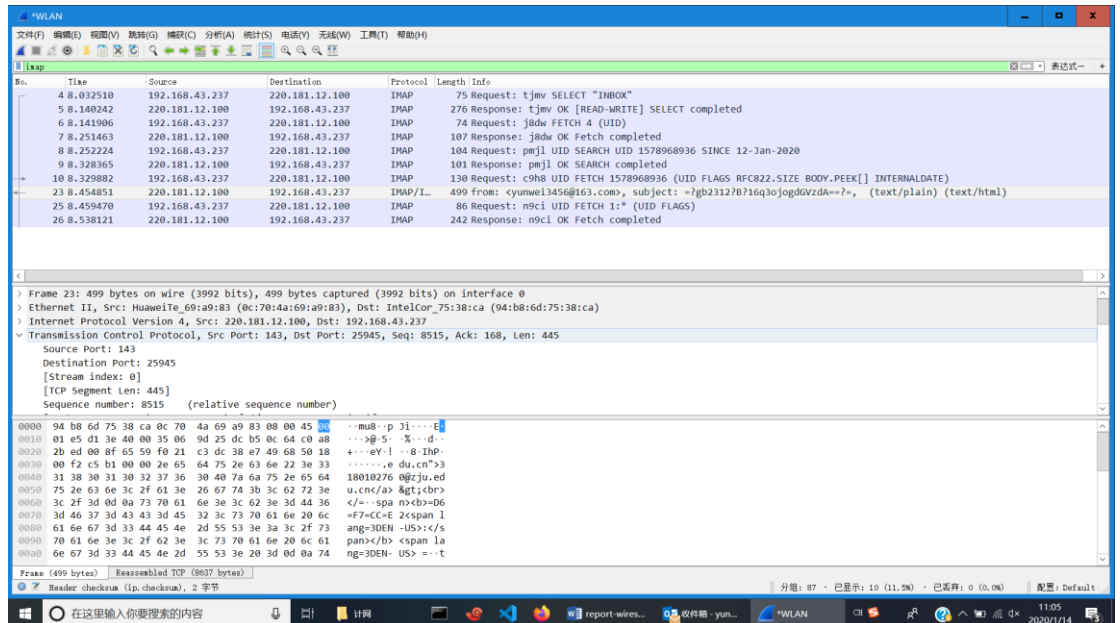
This is a multipart message in MIME format.

-----_NextPart_000_0028_01D5CAC8.D629A0E0
10 客户端 分组, 10 服务器 分组, 10 turn(s).
```

整个对话 (7103 bytes) 显示和保存数据为 ASCII 流 0

查找: 查找下一个(N)

- 打开邮件客户端 Foxmail 或 Outlook, 收取自己邮箱中的邮件 (请在邮件服务器中设置允许 POP3 或者 IMAP), 并捕获这次的数据包。捕获到的数据包由几层协议构成? 分别是什么协议? 标出数据包的源和目标 IP 地址、源和目标端口。



由四层协议构成，分别是以太网、ipv4、TCP IMAP

Source Port: 143

Destination Port: 25945

Source: 220.181.12.100

Destination: 192.168.43.237

- 跟踪 TCP 流，标出 POP3 或 IMAP 协议层中的认证用户名和密码、以及接收的邮件正文（内容过长可截取关键部分）。

UID 1578968936

```
tjmv SELECT "INBOX"
* 4 EXISTS
* 1 RECENT
* OK [UIDVALIDITY 1] UIDs valid
* FLAGS (\Answered \Seen \Deleted \Draft \Flagged)
* OK [PERMANENTFLAGS (\Answered \Seen \Deleted \Draft \Flagged)] Limited
tjmv OK [READ-WRITE] SELECT completed
j8dw FETCH 4 (UID)
* 4 FETCH (UID 1578968936)
j8dw OK Fetch completed
pmjl UID SEARCH UID 1578968936 SINCE 12-Jan-2020
* SEARCH 1578968936
pmjl OK SEARCH completed
c9h8 UID FETCH 1578968936 (UID FLAGS RFC822.SIZE BODY.PEEK[] INTERNALDATE)
* 4 FETCH (UID 1578968936 INTERNALDATE "14-Jan-2020 11:04:44 +0800" FLAGS () RFC822.SIZE
8502 BODY[] {8500}
Received: from LAPTOPFSMLGFU0 (unknown [140.243.192.213])
by smtp9 (Coremail) with SMTP id DcCowACXETDKLx1e7lZDBQ--.15327S2;
Tue, 14 Jan 2020 11:04:43 +0800 (CST)
From: <yunwei3456@163.com>
```

正文：

```

-----=_NextPart_000_0039_01D5CACA.6CD6AAF0
Content-Type: text/plain;
      charset="gb2312"
Content-Transfer-Encoding: quoted-printable

=20

=20

=B7=A2=BC=FE=C8=CB: yunwei3456@163.com <yunwei3456@163.com>=20
=B7=A2=CB=CD=CA=B1=BC=E4: 2020=C4=EA1=D4=C214=C8=D5 10:53
=CA=D5=BC=FE=C8=CB: 'yunwei3456@163.com' <yunwei3456@163.com>
=D6=F7=CC=E2: =D7=AA=B7=A2: test

=20

=20

=20

=B7=A2=BC=FE=C8=CB: yunwei3456@163.com <yunwei3456@163.com>=20
=B7=A2=CB=CD=CA=B1=BC=E4: 2020=C4=EA1=D4=C214=C8=D5 10:49
=CA=D5=BC=FE=C8=CB: '3180102760@zju.edu.cn' <3180102760@zju.edu.cn>
=D6=F7=CC=E2: =D7=AA=B7=A2: test

=20

=20

=20

=B7=A2=BC=FE=C8=CB: yunwei3456@163.com <mailto:yunwei3456@163.com> =
<yunwei3456@163.com
<mailto:yunwei3456@163.com> >=20
=B7=A2=CB=CD=CA=B1=BC=E4: 2020=C4=EA1=D4=C214=C8=D5 10:47
=CA=D5=BC=FE=C8=CB: '3180102760@zju.edu.cn' <3180102760@zju.edu.cn
<mailto:3180102760@zju.edu.cn> >

```

✧ Part Five

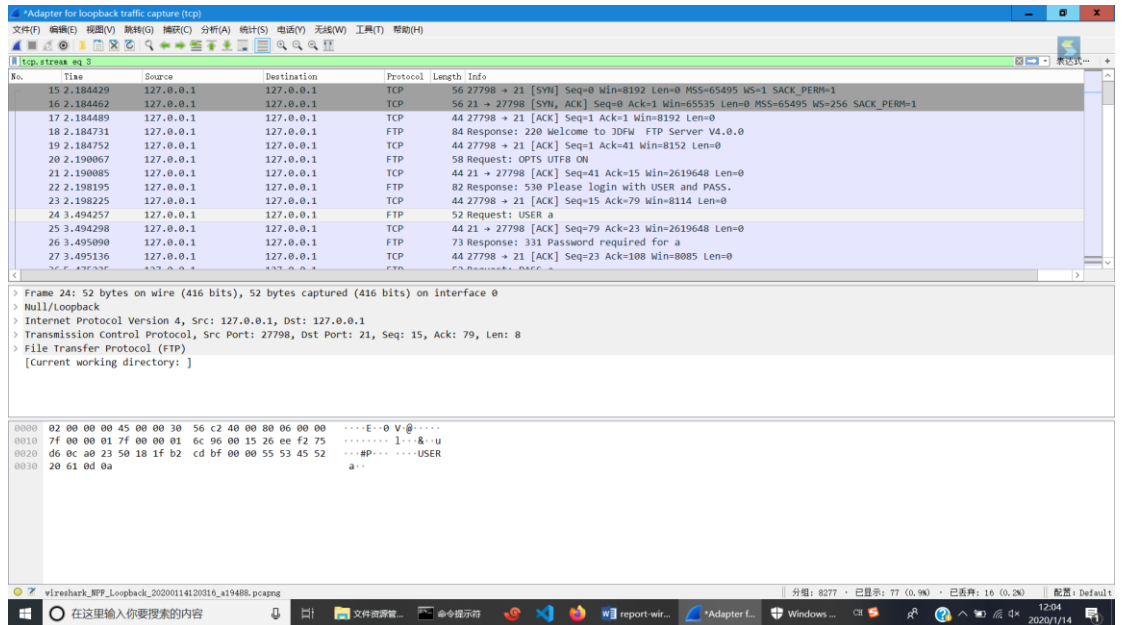
- 运行 FTP xxx.com 命令，连接并登录服务器，输入用户名和帐号（如果是免费服务器，可以使用匿名帐号 Anonymous，密码是任意的邮箱）。捕获到的数据包由几层协议构成？分别是什么协议？标出数据包的源和目标 IP 地址、源和目标端口。
- 由四层协议构成，分别是以太网、ipv4、TCP FTP

Source Port: 21

Destination Port: 27798

Source: 127.0.0.1

Destination: 127.0.0.1



- 跟踪 TCP 流，标注客户端发出的登录命令、用户名、密码以及服务器的响应。

登录命令

530 Please login with USER and PASS.

用户名

USER a

331 Password required for a

密码

PASS a

230 Client :a successfully logged in. Client IP :127.0.0.1

```
220 Welcome to JDFW FTP Server V4.0.0
OPTS UTF8 ON
530 Please login with USER and PASS.
USER a
331 Password required for a
PASS a
230 Client :a successfully logged in. Client IP :127.0.0.1
PORT 127,0,0,1,108,151
200 Port command successful.
NLST
150 Opening ASCII mode data connection for directory list.
226 Transfer complete.
CWD qq
250 "//qq" is current directory.
PORT 127,0,0,1,108,154
200 Port command successful.
NLST
150 Opening ASCII mode data connection for directory list.
226 Transfer complete.
CWD 1067
550 "1067": Directory not found.
CWD 1067852565
250 "//qq/1067852565" is current directory.
PORT 127,0,0,1,108,161
200 Port command successful.
NLST
150 Opening ASCII mode data connection for directory list.
226 Transfer complete.
PORT 127,0,0,1,108,163
200 Port command successful.
RETR Info.db
150 Opening BINARY mode data connection for file transfer.
226 Transfer complete.
QUIT
220 Bye
```

- 执行列目录操作 (ls), 在新捕获的数据包中跟踪 TCP 流, 标注客户端发出的命令、以及服务器的响应。查看是否建立了一个新的 TCP 连接, 跟踪该连接的 TCP 流。建议连接校内服务器, 如果服务器在校外, 可能需要先执行 passive 命令 (下同)。建立了一个新的 TCP 连接。

```
NLST
150 Opening ASCII mode data connection for directory list.
226 Transfer complete.
```



```
$RECYCLE.BIN
.Trash-1000
3DMGAME
Android
arcgis
BaiduNetdiskDownload
C
CloudMusic
Config.Msi
debug.exe
Downloads
eclipse-workspace
GACacheV1
Game
msdownld.tmp
MSOCache
music
MyDownloads
Program Files
Program Files (x86)
ProgramData
qq
Recovery
Recovery.txt
result.txt
steam
SuperMap
System Volume Information
Users
VC_RED.cab
VC_RED.MSI
Virtual Machines
WeChat Files
win32-loader
Windows Kits
```

- 执行更换目录操作 (**cd**)，在新捕获的数据包中跟踪 TCP 流，标注客户端发出的命令、以及服务器的响应。

```
CWD qq
```

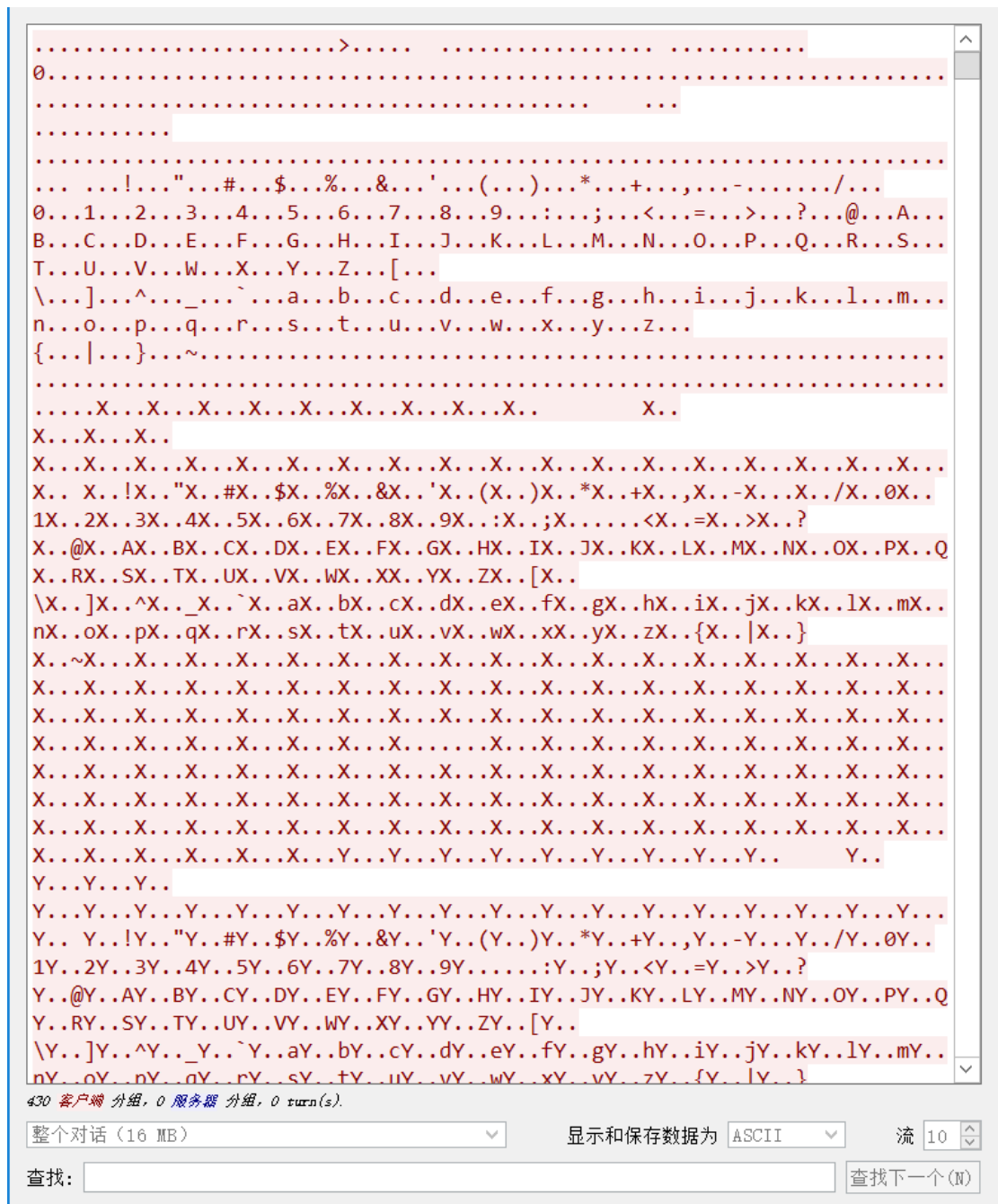
```
250 "/"qq" is current directory.
```

- 执行下载文件操作 (**get filename**)，如果是二进制文件，先执行 **binary** 命令。在新捕获的数据包中跟踪 TCP 流，标注客户端发出的命令、以及服务器的响应。查看是否建立了一个新的 TCP 连接，跟踪该连接的 TCP 流（内容较长时截取部分关键内容）。

```
RETR Info.db
```

```
150 Opening BINARY mode data connection for file transfer.
```

226 Transfer complete.



六、实验结果与分析

- Ping 发送的是什么类型的协议数据包？什么时候会出现 ARP 消息？Ping 一个域名和 Ping 一个 IP 地址出现的数据包有什么不同？

Ping 发送的是 ICMP 的协议数据包

Ping 一个域名和 Ping 一个 IP 地址出现的数据包相比还要出现 DNS 解析相关的数据包

- Tracert/Traceroute 发送的是什么类型的协议数据包，整个路由跟踪过程是如何进行

的？

是 ICMP 类型的。

通过 TTL 字段逐步增加来实现，发送的 TTL 字段为 1 的时候到达第一跳的路由器数据包超时，本机就可以收到 ICMP error 的信息知道路由器地址，字段为 2 的知道第二跳，以此类推

- 建立 TCP 连接的数据包由几个构成？各自的 SYN 和 ACK 标志字段是什么？

三个

SYN=1

SYN=1, ACK=1

ACK=1

- 浏览器打开一个网页，可能会看到多个 TCP 连接，多次 HTTP 会话。一个 TCP 连接上是否会存在多个 HTTP 会话？什么情况下会出现 DNS 数据包？

会的。

当使用域名访问网页的时候，而本地又没有相关信息就会出现 DNS 数据包

- 邮件客户端发送一封电子邮件，需要几次请求、响应消息的交互？消息的一般格式是什么？邮件正文结束的标记是什么？

使用 smtp 协议发送邮件给邮件服务器时需要六次请求、响应消息的交互

1、使用"ehlo"命令和连接上的 smtp 服务器打声招呼，例如：

2、使用"auth login"命令登录到 Smtplib 服务器

3、指明邮件的发件人和收件人

mail from:<gacl@sohu.com>

rcpt to:<xdp_gacl@sina.cn>

4、编写要发送的邮件内容，邮件的编写格式是有一定的规则的，一封格式良好的邮件应该包含邮件头和邮件的主体内容。

邮件头使用下面的三个字段来指明

from 字段用于指明邮件的发送人

to 字段用于指明邮件的收件人

subject 字段用于指明邮件的主题

邮件的内容包含了这些信息之后才是一封格式良好的邮件。

①、输入"data"命令

data

②、编写邮件内容

from:<gacl@sohu.com> ----邮件头

to:<xdp_gacl@sina.cn> ----邮件头

subject:hello ----邮件头

----空行

hello gacl ----邮件的具体内容

5、输入一个.告诉邮件服务器邮件内容已经写完了

6、输入 quit 命令断开与邮件服务器的连接

quit

- 邮件客户端接收一封电子邮件，需要几次请求、响应消息的交互？消息的一般格式是什么？用户名和密码是否经过了加密处理？

大致经历了以下几个步骤：

1、LOGIN

2、SELECT INBOX

3、SEARCH NEW

4、FETCH

5、LOGOUT

一般格式：

邮件头

Received: from

by smtp9 (Coremail) with SMTP id DcCowACXETDKLx1e7lZDBQ--.15327S2;

Tue, 14 Jan 2020 11:04:43 +0800 (CST)

From:

To:

References:

In-Reply-To:

Subject:

Date:

Message-ID:

MIME-Version: 1.0

Content-Type: multipart/alternative;

正文

用户名和密码均经过了加密处理。

- 登录 FTP 服务器时，会产生几个 TCP 连接？列目录和上传或者下载文件时，会产生几个 TCP 连接？

登录时产生一个控制 TCP 连接，

之后列目录再产生一个连接

下载文件再产生一个连接

七、 讨论、心得

FTP 那边使用计网的 `ftp` 一开始得到的数据都是经过 TLS 层加密的，就先自己配了一个简单的 `ftp` 服务器然后抓本地回环接口上的包。