

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： 蒋仕彪

学 院： 计算机学院

系： 计算机系

专 业： 计算机科学与技术

学 号： 3170102587

指导教师： 董玮

2020 年 6 月 4 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： 蒋仕彪 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

✧ 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？

	Time	Source	Destination	Protocol
17	2.500645	95.216.27.30	192.168.1.14	TCP
18	2.500855	192.168.1.14	95.216.27.30	TCP
19	2.763199	95.216.27.30	192.168.1.14	SSL
20	2.803556	192.168.1.14	95.216.27.30	TCP
21	6.485295	fe80::85dc:6b8c:cd3...	fe80::1	DNS
22	6.495127	fe80::1	fe80::85dc:6b8c:cd3...	DNS
23	6.495943	192.168.1.14	211.159.235.30	TCP
24	6.539807	211.159.235.30	192.168.1.14	TCP

不断有数据包被捕获，涉及各种协议，例如 TCP, SSL, DNS, ARP, MDNS, ICMP, HTTP……

- 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。

20	6.576536	127.0.0.1	127.0.0.1	TCP	44 54558 → 54530 [ACK] Seq=371 Ack=291 Win=10131 Len=0
21	7.945139	192.168.1.12	192.168.1.14	TCP	174 50809 → 54041 [PSH, ACK] Seq=1 Ack=1 Win=508 Len=120
22	7.993218	192.168.1.14	192.168.1.12	TCP	54 54041 → 50809 [ACK] Seq=1 Ack=121 Win=509 Len=0
23	8.189148	fe80::d7c:3ea2:5e02...	ff02::1:2	DHCPv6	157 Solicit XID: 0x668563 CID: 0001000121f8b1629cb6d0e2c5cb

> Frame 21: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}, id 8

▼ Ethernet II, Src: IntelCor_43:9f:63 (5c:80:b6:43:9f:63), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)

> Destination: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)

> Source: IntelCor_43:9f:63 (5c:80:b6:43:9f:63)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.14

> Transmission Control Protocol, Src Port: 50809, Dst Port: 54041, Seq: 1, Ack: 1, Len: 120

> Data (120 bytes)

这是个 TCP 协议。源地址和目标地址的 MAC 见图。

- 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。

> Frame 121: 282 bytes on wire (2256 bits), 282 bytes captured (2256 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}, id 8	
▼ Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a)	
> Destination: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a)	
> Source: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)	
Type: IPv4 (0x0800)	
▼ Internet Protocol Version 4, Src: 192.168.1.14, Dst: 121.51.0.176	
0100 = Version: 4	
.... 0101 = Header Length: 20 bytes (5)	
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
Total Length: 268	
Identification: 0xed2b (60715)	
> Flags: 0x4000, Don't fragment	
Fragment offset: 0	
Time to live: 128	
Protocol: TCP (6)	
Header checksum: 0xd126 [validation disabled]	
[Header checksum status: Unverified]	
Destination: 121.51.0.176	

这是个 IP 协议。源 IP 地址：192.168.1.14，目标 IP 地址：121.51.0.176

- 找一个 ARP 数据包，这是请求还是应答？标注发送者的 MAC 地址。

77	24.196315	fe80::d7c:3ea2:5e02::ff02::1:2	DHCPv6	157	Solicit	XID: 0x668563 CID: 0001000121f8b1629cb6d0e2c5cb
78	24.782389	HuaweiTe_fa:ae:6a	RivetNet_e2:c5:cb	ARP	42	Who has 192.168.1.14? Tell 192.168.1.1
79	24.782406	RivetNet_e2:c5:cb	HuaweiTe_fa:ae:6a	ARP	42	192.168.1.14 is at 9c:b6:d0:e2:c5:cb
80	24.783454	HuaweiTe_fa:ae:6a	RivetNet_e2:c5:cb	ARP	42	Who has 192.168.1.14? Tell 192.168.1.1
81	24.783470	RivetNet_e2:c5:cb	HuaweiTe_fa:ae:6a	ARP	42	192.168.1.14 is at 9c:b6:d0:e2:c5:cb
82	24.944810	fe80::1	ff02::1:ff2b:4bb8	ICMPv6	86	Neighbor Solicitation for fe80::f4f8:729b:cb2b:4bb8 from 14:30:04:fa:ae:6a
83	25.179388	fe80::d74:c5c7:7ad::ff02::1:2	DHCPv6	157	Solicit	XID: 0x864f24 CID: 0001000121f8b1629cb6d0e2c5cb
> Frame 78: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}, id 8						
> Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)						
▼ Address Resolution Protocol (request)						
Hardware type: Ethernet (1)						
Protocol type: IPv4 (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
Sender MAC address: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a)						
Sender IP address: 192.168.1.1						
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)						

这是一个请求。发送者的 MAC 地址见图。

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。

4 层。Frame, Ethernet, IP, ICMP

426	188.578896	192.168.1.14	14.215.177.39	ICMP	74	Echo (ping) request	id=0x0001, seq=68/17408, ttl=64 (reply in 427)
427	188.616464	14.215.177.39	192.168.1.14	ICMP	74	Echo (ping) reply	id=0x0001, seq=68/17408, ttl=51 (request in 426)
428	189.333951	192.168.1.14	40.90.185.223	TCP	54	54835 → 443 [FIN, ACK]	Seq=5261 Ack=9147 Win=131584 Len=0
429	189.589902	192.168.1.14	14.215.177.39	ICMP	74	Echo (ping) request	id=0x0001, seq=69/17664, ttl=64 (reply in 430)
430	189.629409	14.215.177.39	192.168.1.14	ICMP	74	Echo (ping) reply	id=0x0001, seq=69/17664, ttl=51 (request in 429)
> Frame 426: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}, id 0							
> Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a)							
> Internet Protocol Version 4, Src: 192.168.1.14, Dst: 14.215.177.39							
▼ Internet Control Message Protocol							
Type: 8 (Echo (ping) request)							
Code: 0							
Checksum: 0x4d17 [correct]							
[Checksum Status: Good]							
Identifier (BE): 1 (0x0001)							
Identifier (LE): 256 (0x0100)							
Sequence number (BE): 68 (0x0044)							
Sequence number (LE): 17408 (0x4400)							

请求包

427	188.616464	14.215.177.39	192.168.1.14	ICMP	74	Echo (ping) reply	id=0x0001, seq=68/17408, ttl=51 (request in 426)
428	189.333951	192.168.1.14	40.90.185.223	TCP	54	54835 → 443 [FIN, ACK]	Seq=5261 Ack=9147 Win=131584 Len=0
429	189.589902	192.168.1.14	14.215.177.39	ICMP	74	Echo (ping) request	id=0x0001, seq=69/17664, ttl=64 (reply in 430)
430	189.629409	14.215.177.39	192.168.1.14	ICMP	74	Echo (ping) reply	id=0x0001, seq=69/17664, ttl=51 (request in 429)
> Frame 427: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}, id 0							
> Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)							
> Internet Protocol Version 4, Src: 14.215.177.39, Dst: 192.168.1.14							
▼ Internet Control Message Protocol							
Type: 0 (Echo (ping) reply)							
Code: 0							
Checksum: 0x5517 [correct]							
[Checksum Status: Good]							
Identifier (BE): 1 (0x0001)							
Identifier (LE): 256 (0x0100)							
Sequence number (BE): 68 (0x0044)							
Sequence number (LE): 17408 (0x4400)							

应答包

- 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？

4 层。分别是 Frame,Ethernet,IPv4,ICMP 协议。

3974 1151.870090 192.168.1.14 14.215.	3989 1155.545621 192.168.1.14 14.215.
3989 1155.545621 192.168.1.14 14.215.	4002 1159.536170 192.168.1.14 14.215.177.39
	4011 1163.546580 192.168.1.14 14.215.177.39
	4020 1167.536123 192.168.1.14 14.215.177.39

0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0) Total Length: 92 Identification: 0x4312 (17170) > Flags: 0x0000 Fragment offset: 0 Time to live: 12	Internet Protocol Version 4, Src: 192.168.1.14, Dst: 14.215.177.39 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: N/A) Total Length: 92 Identification: 0x4315 (17173) > Flags: 0x0000 Fragment offset: 0 Time to live: 13
---	---

每三个连续的请求包 TTL 相同。总体 TTL 会不断加一递增。

3989 1155.545621 192.168.1.14 14.215.177.39	ICMP	106 Echo (ping) request id=0x0001, seq=212/54272, ttl=12 (no response found!)
---	------	---

> Frame 3989: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 5 > Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a) > Internet Protocol Version 4, Src: 192.168.1.14, Dst: 14.215.177.39 > Internet Control Message Protocol Type: 8 (Echo (ping) request) Code: 0 Checksum: 0xf72a [correct] [Checksum Status: Good] Identifier (BE): 1 (0x0001) Identifier (LE): 256 (0x0100) Sequence number (BE): 212 (0x00d4) Sequence number (LE): 54272 (0xd400)
--

这是请求包。

4098 1187.585078 14.215.177.39 192.168.1.14	ICMP	106 Echo (ping) reply id=0x0001, seq=220/56320, ttl=51 (request in 4097)
---	------	--

> Frame 4098: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 5 > Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb) > Internet Protocol Version 4, Src: 14.215.177.39, Dst: 192.168.1.14 > Internet Control Message Protocol Type: 0 (Echo (ping) reply) Code: 0 Checksum: 0xff22 [correct] [Checksum Status: Good] Identifier (BE): 1 (0x0001) Identifier (LE): 256 (0x0100) Sequence number (BE): 220 (0x00dc) Sequence number (LE): 56320 (0xdc00) [Request frame: 4097] [Response time: 38.087 ms]

这是应答包。

- 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。

5 层。Frame, Ethernet, IP, UDP, DNS

69	13.993168	117.185.116.142	192.168.1.14	UDP	404 8000 → 52000 Len=362
70	14.046482	fe80::85dc:6b8c:cd3...	fe80::1	DNS	152 Standard query 0x0001 PTR 1
71	14.061220	fe80::1	fe80::85dc:6b8c:cd3...	DNS	201 Standard query response 0x0001
72	14.064627	fe80::85dc:6b8c:cd3...	fe80::1	DNS	93 Standard query 0x0002 A www
73	14.083203	fe80::1	fe80::85dc:6b8c:cd3...	DNS	152 Standard query response 0x0002
74	14.086700	fe80::85dc:6b8c:cd3...	fe80::1	DNS	93 Standard query response 0x0003 AAAA

> Frame 69: 404 bytes on wire (3232 bits), 404 bytes captured (3232 bits) on interface \Device\NPF_{401BD6...}

> Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)

> Internet Protocol Version 4, Src: 117.185.116.142, Dst: 192.168.1.14

▼ User Datagram Protocol, Src Port: 8000, Dst Port: 52000

Source Port: 8000

Destination Port: 52000

Length: 370

这是 UDP 访问的端口字段标记。

Type: IPv6 (0x86dd)
> Internet Protocol Version 6, Src: fe80::85dc:6b8c:cd3e:5128, Dst: fe80::1
> User Datagram Protocol, Src Port: 62012, Dst Port: 53
▼ Domain Name System (query)
Transaction ID: 0x0002
> 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: 73]

这是一个请求包。

Type: IPv6 (0x86dd)
> Internet Protocol Version 6, Src: fe80::1, Dst: fe80::85dc:6b8c:cd3e:5128
> User Datagram Protocol, Src Port: 53, Dst Port: 62012
▼ Domain Name System (response)
Transaction ID: 0x0002
> Flags: 0x8180 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: 72]

这是一个响应包。

✧ 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。

五层。Frame, Ethernet, IPv4, TCP, HTTP

```
558 5.258493 211.159.235.146 192.168.1.14 HTTP 650 HTTP/1.1 200 OK (text/html)
> Frame 558: 650 bytes on wire (5200 bits), 650 bytes captured (5200 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DDD369024}, id 0
> Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)
> Internet Protocol Version 4, Src: 211.159.235.146, Dst: 192.168.1.14
> Transmission Control Protocol, Src Port: 80, Dst Port: 55040, Seq: 1, Ack: 910, Len: 596
> Hypertext Transfer Protocol
  Line-based text data: text/html (1 lines)
    policyNo=1&brwType=1&pingbackTimes=7&baidu_pingbackTimes=25&sogou_pingbackTimes=25&isFullHtml=1&bd_query_min_len=2&bd_query_max_len=10&bd_qu
```

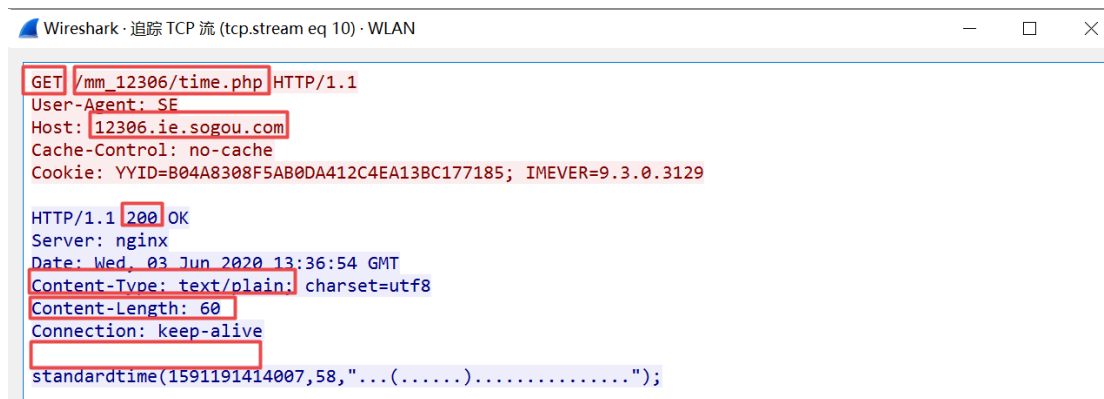
- 找到建立 TCP 连接的三个数据包（称为三次握手），展开 TCP 协议层的 Flags 字段，分别标记三个数据包的 SYN 标志位和 ACK 标志位。

```
539 5.178977 192.168.1.14 211.159.235.146 TCP 66 55040 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
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.]
```

```
546 5.215957 211.159.235.146 192.168.1.14 TCP 66 80 → 55040 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400 SACK_PERM=1 W
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.]
```

```
547 5.216420 192.168.1.14 211.159.235.146 TCP 54 55040 → 80 [ACK] Seq=1 Ack=1 Win=131584 Len=0
Flags: 0x010 (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
  ....0 .... = Syn: Not set
  ....0 .... = Fin: Not set
  [TCP Flags: .....A....]
```

- 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）。



- 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。

一共有 0~32 共 33 个 TCP 流。

第 9 个 TCP 流对应 4 组，第 10 个对应 1 组，第 15 个对应 1 组（其余失败），第 16, 18, 20, 21, 24, 26, 30 对应 1 组。以下的例子是第 9 个流的截图。

No.	Time	Source	Destination	Protocol	Length	Info
335	3.151754	192.168.1.14	109.244.23.123	TCP	66	55031 → 80 [SVN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
336	3.189541	109.244.23.123	192.168.1.14	TCP	58	80 → 55031 [SVN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400
337	3.189709	192.168.1.14	109.244.23.123	TCP	54	55031 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
338	3.194448	192.168.1.14	109.244.23.123	HTTP	292	GET /query?v=8.6.1.31812&m=sogouexplorer&f=P011dGFibGVQaw5nYmFja0hpc
341	3.232700	109.244.23.123	192.168.1.14	TCP	54	80 → 55031 [ACK] Seq=1 Ack=239 Win=26800 Len=0
342	3.238795	109.244.23.123	192.168.1.14	HTTP	338	HTTP/1.1 200 OK
343	3.238996	192.168.1.14	109.244.23.123	TCP	54	55031 → 80 [ACK] Seq=239 Ack=285 Win=65535 Len=0
344	3.253259	192.168.1.14	109.244.23.123	HTTP	320	GET /query?v=8.6.1.31812&m=sogouexplorer&f=P1N1dFVJQwN0aw9uUGluZ2Jh
345	3.298941	109.244.23.123	192.168.1.14	HTTP	212	HTTP/1.1 200 OK
346	3.299143	192.168.1.14	109.244.23.123	TCP	54	55031 → 80 [ACK] Seq=505 Ack=443 Win=65535 Len=0
347	3.305278	192.168.1.14	109.244.23.123	HTTP	338	GET /query?v=8.6.1.31812&m=sogouexplorer&f=P01vZGlmeVNIYXJjaEVuZ2lu
348	3.349090	109.244.23.123	192.168.1.14	HTTP	212	HTTP/1.1 200 OK
349	3.349323	192.168.1.14	109.244.23.123	TCP	54	55031 → 80 [ACK] Seq=789 Ack=601 Win=65535 Len=0
350	3.352839	192.168.1.14	109.244.23.123	HTTP	326	GET /query?v=8.6.1.31812&m=sogouexplorer&f=P0d1dEnvbmZp20ludEBDb25m
351	3.397862	109.244.23.123	192.168.1.14	HTTP	212	HTTP/1.1 200 OK
352	3.398075	192.168.1.14	109.244.23.123	TCP	54	55031 → 80 [ACK] Seq=1061 Ack=759 Win=65535 Len=0

✧ 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。

```
43 2.685975 192.168.1.14 220.181.12.14 SMTP/... 1015 from: "jiangshibiao001@163.com" <jiangshibiao001@163.com>, subject: Test...
> Frame 43: 1015 bytes on wire (8120 bits), 1015 bytes captured (8120 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}, id 0
> Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a)
> Internet Protocol Version 4, Src: 192.168.1.14, Dst: 220.181.12.14
> Transmission Control Protocol, Src Port: 55517, Dst Port: 25, Seq: 609, Ack: 381, Len: 961
> Simple Mail Transfer Protocol
  Internet Message Format
    Date: Wed, 3 Jun 2020 22:29:10 +0800
    From: "jiangshibiao001@163.com" <jiangshibiao001@163.com>, 1 item
    To: 3170102587 <3170102587@zju.edu.cn>, 1 item
    Subject: Test for Computer Network
    Unknown-Extension: X-Priority: 3 (Contact Wireshark developers if you want this supported.)
    Unknown-Extension: X-GUID: FB4593CF-548B-4C8A-85BC-FE8A6D0AE12C (Contact Wireshark developers if you want this supported.)
    Unknown-Extension: X-Has-Attach: no (Contact Wireshark developers if you want this supported.)
    X-Mailer: Foxmail 7.2.16.188[cn]
    MIME-Version: 1.0
```

5 层。Fram, Ethernet, IP, TCP, SMTP

- 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）。

EHLO

```
22 2.266601 192.168.1.14 220.181.12.14 SMTP 76 C: EHLO DESKTOP-7HNC82T
23 2.303245 220.181.12.14 192.168.1.14 TCP 60 25 → 55517 [ACK] Seq=66 Ack=23 Win=14720 Len=0
24 2.303245 220.181.12.14 192.168.1.14 SMTP 239 S: 250-mail | PIPELINING | AUTH LOGIN PLAIN | AUTH
25 2.303753 192.168.1.14 220.181.12.14 SMTP 66 C: AUTH LOGIN
26 2.339857 220.181.12.14 192.168.1.14 SMTP 72 S: 334 dXNlcm5hbWU6
28 2.364788 192.168.1.14 220.181.12.14 SMTP 88 C: User: am1hbmDzaGliaWFVMDAxQDE2My5jb20=
29 2.400894 220.181.12.14 192.168.1.14 SMTP 72 S: 334 UGFzc3dvcmQ6
30 2.401434 192.168.1.14 220.181.12.14 SMTP 76 C: Pass: am1hbmDxOTk5MDEyMA==
> Frame 22: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{4018D675-1267-4033-9771-E61DD0369024}
> Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a)
> Internet Protocol Version 4, Src: 192.168.1.14, Dst: 220.181.12.14
> Transmission Control Protocol, Src Port: 55517, Dst Port: 25, Seq: 1, Ack: 66, Len: 22
> Simple Mail Transfer Protocol
  Command Line: EHLO DESKTOP-7HNC82T\r\n
  Command: EHLO
  Request parameter: DESKTOP-7HNC82T

33 2.498888 192.168.1.14 220.181.12.14 SMTP 92 C: MAIL FROM: <jiangshibiao001@163.com>
34 2.534966 220.181.12.14 192.168.1.14 TCP 60 25 → 55517 [ACK] Seq=318 Ack=129 Win=14720 Len=0
35 2.537644 220.181.12.14 192.168.1.14 SMTP 67 S: 250 Mail OK
36 2.538154 192.168.1.14 220.181.12.14 SMTP 88 C: RCPT TO: <3170102587@zju.edu.cn>
37 2.574251 220.181.12.14 192.168.1.14 SMTP 67 S: 250 Mail OK
38 2.574970 192.168.1.14 220.181.12.14 SMTP 60 C: DATA
40 2.610371 220.181.12.14 192.168.1.14 SMTP 91 S: 354 End data with <CR><LF>.<CR><LF>
41 2.610812 192.168.1.14 220.181.12.14 SMTP 494 C: DATA fragment, 440 bytes
42 2.685729 220.181.12.14 192.168.1.14 TCP 60 25 → 55517 [ACK] Seq=381 Ack=609 Win=15744 Len=0
43 2.685975 192.168.1.14 220.181.12.14 SMTP/... 1015 from: "jiangshibiao001@163.com" <jiangshibiao001@163.com>
  To: 3170102587 <3170102587@zju.edu.cn>, 1 item
  Subject: Test for Computer Network
  Unknown-Extension: X-Priority: 3 (Contact Wireshark developers if you want this supported.)
  Unknown-Extension: X-GUID: FB4593CF-548B-4C8A-85BC-FE8A6D0AE12C (Contact Wireshark developers if you want this supported.)
  Unknown-Extension: X-Has-Attach: no (Contact Wireshark developers if you want this supported.)
  X-Mailer: Foxmail 7.2.16.188[cn]
  MIME-Version: 1.0
  Message-ID: <202006032229092904560@163.com>
  Content-Type: multipart/alternative;\r\n\tboundary="-----_001_NextPart205725204067_-----"
  MIME Multipart Media Encapsulation, Type: multipart/alternative, Boundary: "-----_001_NextPart205725204067_-----"
```

- 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。

5 层。Frame, Ethernet, IP, TCP, IMAP

No.	Time	Source	Destination	Protocol	Length	Info
103	2.803768	192.168.1.14	123.126.97.78	IMAP	75	Request: C17 FETCH 1:1 (UID)
104	2.848397	123.126.97.78	192.168.1.14	IMAP	106	Response: C17 OK Fetch completed
105	2.849396	192.168.1.14	123.126.97.78	IMAP	122	Request: C18 UID FETCH 1405244917 (UID RFC822.SIZE FLAGS BODY.PEEK[HEADERS])
106	2.897041	123.126.97.78	192.168.1.14	TCP	1454	143 → 55575 [ACK] Seq=1754 Ack=833 Win=1400 [TCP segment of
107	2.897041	123.126.97.78	192.168.1.14	IMAP/...	1454	from: "<?gb18030?B?0KHA9b201DmyMw==?>" <jiangshibiao1999@qq.com>, subj
108	2.897042	123.126.97.78	192.168.1.14	IMAP	79	Response:)
109	2.897312	192.168.1.14	123.126.97.78	TCP	54	55575 → 143 [ACK] Seq=833 Ack=4554 Win=131584 Len=0
112	2.936575	192.168.1.14	123.126.97.78	TCP	54	55575 → 143 [ACK] Seq=833 Ack=4579 Win=131328 Len=0
124	3.199939	192.168.1.14	123.126.97.78	IMAP	64	Request: C19 NOOP
127	3.244841	123.126.97.78	192.168.1.14	IMAP	77	Response: C19 OK NOOP completed
128	3.246151	192.168.1.14	123.126.97.78	IMAP	70	Request: C20 CAPABILITY

> Frame 107: 1454 bytes on wire (11632 bits), 1454 bytes captured (11632 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 0
 > Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)
 > Internet Protocol Version 4, Src: 123.126.97.78, Dst: 192.168.1.14
 > Transmission Control Protocol, Src Port: 143, Dst Port: 55575, Seq: 3154, Ack: 833, Len: 1400
 > [2 Reassembled TCP Segments (2800 bytes): #106(1400), #107(1400)]
 > Internet Message Access Protocol
 > Internet Message Format

- 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）。

35	2.046754	123.126.97.78	192.168.1.14	IMAP	182	Response: C1 OK CAPABILITY completed
36	2.047531	192.168.1.14	123.126.97.78	IMAP	208	Request: C2 ID ("name" "com.tencent.foxmail" "version" "7.2.16.188"
39	2.092632	123.126.97.78	192.168.1.14	IMAP	154	Response: C2 OK ID completed
40	2.093168	192.168.1.14	123.126.97.78	IMAP	104	Request: C3 LOGIN jiangshibiao001@163.com "
45	2.178057	123.126.97.78	192.168.1.14	TCP	60	143 → 55575 [ACK] Seq=311 Ack=220 Win=15744 Len=0
48	2.191035	123.126.97.78	192.168.1.14	IMAP	77	Response: C3 OK LOGIN completed
49	2.191606	192.168.1.14	123.126.97.78	IMAP	69	Request: C4 CAPABILITY
51	2.235335	123.126.97.78	192.168.1.14	TCP	60	143 → 55575 [ACK] Seq=334 Ack=235 Win=15744 Len=0

> From: "<?gb18030?B?0KHA9b201DmyMw==?>" <jiangshibiao1999@qq.com>, 1 item
 > To: "<?gb18030?B?am1hbmZaGliaWVMDAX?>" <jiangshibiao001@163.com>, 1 item
 > Subject: Test
 MIME-Version: 1.0
 > Content-Type: multipart/alternative;\n\n\tboundary="-----_NextPart_5ED787F7_10693668_770A6E8E"
 Content-Transfer-Encoding: 8Bit
 Date: Wed, 3 Jun 2020 22:47:19 +0800
 > Unknown-Extension: X-Priority: 3 (Contact Wireshark developers if you want this supported.)
 Message-ID: <tencent_5144C85C128E5456CDA3968E11D57104CB0A@qq.com>
 > Unknown-Extension: X-QQ-MIME: TCtime 1.0 by Tencent (Contact Wireshark developers if you want this supported.)
 X-Mailer: QQMail 2.x

✧ Part Five

本部分需要边操作，边捕获，请在每次操作后暂停捕获，或者使用过滤器。建议通过 FTP 命令行进行实验，也可以使用 FTP 图形客户端。

- 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。

我登陆的是一个同学的带密码服务器。

```
Windows PowerShell
PS C:\Users\jiang\Desktop> ftp 120.78.125.201
连接到 120.78.125.201.
220 Microsoft FTP Service
501 Option not supported.
用户(120.78.125.201:(none)): Administrator
331 Password required
密码:
230 User logged in.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
a.exe
FileZilla_Server-0_9_60_2.exe
lrw-vl-partaf
numpy-1.16.2+mkl-cp27-cp27m-win_amd64.whl
python-3.7.2-amd64.zip
sogou_pinyin_94a.exe
test.zip
```

5 层。Frame, Ethernet, IP, TCP, FTP。

6	2.052920	120.78.125.201	192.168.1.14	FTP	81	Response: 220 Microsoft FTP Service
7	2.069975	192.168.1.14	120.78.125.201	FTP	68	Request: OPTS UTF8 ON

>	Frame 6: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 6
>	Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)
>	Internet Protocol Version 4, Src: 120.78.125.201, Dst: 192.168.1.14
>	Transmission Control Protocol, Src Port: 21, Dst Port: 57276 Seq: 1, Ack: 1, Len: 27
>	File Transfer Protocol (FTP)
>	[Current working directory:]

IP 地址和端口见图。

- 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。

5	2.014278	192.168.1.14	120.78.125.201	TCP	54	57276 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
6	2.052920	120.78.125.201	192.168.1.14	FTP	81	Response: 220 Microsoft FTP Service
7	2.069975	192.168.1.14	120.78.125.201	FTP	68	Request: OPTS UTF8 ON
8	2.108737	120.78.125.201	192.168.1.14	FTP	81	Response: 501 Option not supported.
9	2.154805	192.168.1.14	120.78.125.201	TCP	54	57276 → 21 [ACK] Seq=15 Ack=55 Win=8138 Len=0
17	10.173767	192.168.1.14	120.78.125.201	FTP	74	Request: USER Administrator
18	10.216516	120.78.125.201	192.168.1.14	FTP	77	Response: 331 Password required
19	10.262723	192.168.1.14	120.78.125.201	TCP	54	57276 → 21 [ACK] Seq=35 Ack=78 Win=8115 Len=0
34	19.064933	192.168.1.14	120.78.125.201	FTP	75	Request: PASS li*****
35	19.107221	120.78.125.201	192.168.1.14	FTP	75	Response: 230 User logged in.

用户名密码、服务器响应见图。

- 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)。

好像依然是在原来的 TCP 流里。

ls 命令被翻译成为 NLST 命令。

35	24.252084	192.168.1.14	120.78.125.201	FTP	60	Request: NLST
41	24.290643	120.78.125.201	192.168.1.14	FTP	108	Response: 125 Data connection already open; Transfer starting.
42	24.290896	120.78.125.201	192.168.1.14	FTP	78	Response: 226 Transfer complete.

> Frame 35: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 0 > Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a) > Internet Protocol Version 4, Src: 192.168.1.14, Dst: 120.78.125.201 > Transmission Control Protocol, Src Port: 57361, Dst Port: 21, Seq: 82, Ack: 129, Len: 6 > File Transfer Protocol (FTP) NLST\r\n Request command: NLST						
---	--	--	--	--	--	--

- 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。

50	31.489205	192.168.1.14	120.78.125.201	FTP	61	Request: CWD a
51	31.540408	120.78.125.201	192.168.1.14	FTP	103	Response: 550 The system cannot find the file specified.

> Frame 50: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 0 > Ethernet II, Src: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb), Dst: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a) > Internet Protocol Version 4, Src: 192.168.1.14, Dst: 120.78.125.201 > Transmission Control Protocol, Src Port: 57361, Dst Port: 21, Seq: 88, Ack: 207, Len: 7 > File Transfer Protocol (FTP) CWD a\r\n Request command: CWD Request arg: a [Current working directory:]						
--	--	--	--	--	--	--

cd 命令被翻译成 CWD 命令。

服务器上没有 a 这个文件夹, 所以操作失败了。

- 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)。

在原来的 TCP 流里能看到建立和成功的信息, 但是看不到具体的传输信息。

59	38.713588	192.168.1.14	120.78.125.201	FTP	69	Request: RETR test.zip
67	38.753891	120.78.125.201	192.168.1.14	FTP	108	Response: 125 Data connection already open; Transfer starting.
80	38.801263	192.168.1.14	120.78.125.201	TCP	54	57361 → 21 [ACK] Seq=136 Ack=340 Win=7853 Len=0
641	39.178072	120.78.125.201	192.168.1.14	FTP	78	Response: 226 Transfer complete.

> Frame 641: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{401BD675-1267-4033-9771-E61DD0369024}, id 0 > Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb) > Internet Protocol Version 4, Src: 120.78.125.201, Dst: 192.168.1.14 > Transmission Control Protocol, Src Port: 21, Dst Port: 57361, Seq: 340, Ack: 136, Len: 24 > File Transfer Protocol (FTP) 226 Transfer complete.\r\n Response code: Closing data connection (226) Response arg: Transfer complete. [Current working directory:]						
--	--	--	--	--	--	--

而且他们之间的序号相差很多，猜测在这之间有具体的传输细节。

```

516 39.044282 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
517 39.044283 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
518 39.044284 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
519 39.044285 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
520 39.044404 192.168.1.14 120.78.125.201 TCP 54 57363 + 20 [ACK] Seq=1 Ack=404417 Win=131584 Len=0
521 39.044864 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
522 39.044865 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
523 39.044885 192.168.1.14 120.78.125.201 TCP 54 57363 + 20 [ACK] Seq=1 Ack=407217 Win=131584 Len=0
524 39.044970 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
525 39.045050 192.168.1.14 120.78.125.201 TCP 54 57363 + 20 [ACK] Seq=1 Ack=410017 Win=131584 Len=0
526 39.045059 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
527 39.045060 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
528 39.045062 120.78.125.201 192.168.1.14 FTP-D... 1454 FTP Data: 1400 bytes (PORT) (RETR test.zip)
529 39.045252 192.168.1.14 120.78.125.201 TCP 54 57363 + 20 [ACK] Seq=1 Ack=412817 Win=131584 Len=0
530 39.045437 192.168.1.14 120.78.125.201 TCP 54 57363 + 20 [ACK] Seq=1 Ack=415617 Win=131584 Len=0

```

Frame 526: 1454 bytes on wire (11632 bits), 1454 bytes captured (11632 bits) on interface \Device\NPF_{408BD675-1267-4033-9771-E61DD369024} Ethernet II, Src: HuaweiTe_fa:ae:6a (14:30:04:fa:ae:6a), Dst: RivetNet_e2:c5:cb (9c:b6:d0:e2:c5:cb)

Internet Protocol Version 4, Src: 120.78.125.201, Dst: 192.168.1.14

Transmission Control Protocol, Src Port: 20, Dst Port: 57363, Seq: 439417, Ack: 1, Len: 1400

FTP Data (1400 bytes data)

[\[Setup frame: 55\]](#)

[Setup method: PORT]

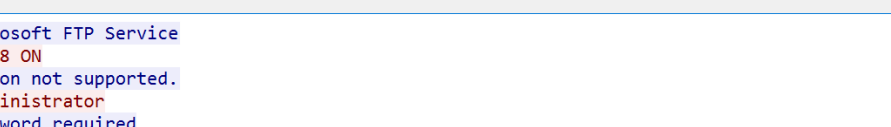
[Command: RETR test.zip]

果然，在此之前全是传输，每次传输只传 1400 字节。

这是总体的终端命令。

```
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
a.exe
FileZilla_Server-3.8.0_3.exe
firefox-portable
mozilla-3.6.3-win32-xpSP2-sp3To-win_usd94.msi
python-2.7.3-amd64.zip
python_x64_python_64a.exe
test.zip
total 120, zip
total 120, zip
total size: 7 (local, portable)
本地文件与远程文件一致 4. 本地文件比远程 -120, zip
本地文件与远程文件不一致 4. 本地文件比远程 -120, zip
本地文件与远程文件不一致 4. 本地文件比远程 -120, zip
本地文件与远程文件不一致 4. 本地文件比远程 -120, zip
226 Transfer complete.
ftp: 收到 380 字节, 用时 0.05秒 7.17千字节/秒。
ftp> cd a
550 The system cannot find the file specified.
ftp> get test.zip
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp: 收到 533953 字节, 用时 0.39秒 1355.21千字节/秒。
```

可以通过追踪 TCP 流看到交互信息。



Wireshark · 追踪 TCP 流 (tcp.stream eq 2) · WLAN

220 Microsoft FTP Service

OPTS UTF8 ON

501 Option not supported.

USER Administrator

331 Password required

PASS li*****

230 User logged in.

PORT 192,168,1,14,224,18

200 PORT command successful.

NLST

125 Data connection already open; Transfer starting.

226 Transfer complete.

CWD a

550 The system cannot find the file specified.

PORT 192,168,1,14,224,19

200 PORT command successful.

RETR test.zip

125 Data connection already open; Transfer starting.

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？

ICMP

请求地址解析时。

Ping IP 时只有 ICMP 和 ARP 包，ping 域名时 ICMP、ARP、DNS 都存在。

- 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？

ICMP 类型

Tracert 先向目标服务器送出一个 TTL=1 的数据包，由于路径上第一个路由器接收后 TTL 变为 0，因此路由器会丢弃该数据包，并返回 Time-to-live exceeded，由此得到路径上第一个路由器地址。再送出 TTL=2,3...的数据包即可，每次连续送三个。最后，若有数据包到达目标服务器，目标服务器正常返回 reply，即可知路径到达。

- 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？

3 个（三次握手）。

SYN 表示开始建立连接。

ACK 指的是 ack（acknowledge）是否有效。

- 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？

我认为不会，因为 HTTP 是建立在 TCP 上的。

当需要域名解析的时候。

- 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？

大概是三个阶段，先是发送 EHLP 建立连接，再是登陆阶段（用户名和密码的请求和相应），最后才是邮件数据的发送。

Date:

From:

To:

Subject:

Content-Type:

Content:

正文结束标记是. (即\r\n.\r\n)

- 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？
也分为三个阶段：连接，用户登陆，收取邮件。

Date:

From:

To:

Subject:

Content-Type:

Content:

没有经过加密处理（所以我在上面的截图对密码打了码）。

- 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？
1 个。
2 个，一个 TCP 连接用于控制，另外一个 TCP 连接用来传输数据。

七、 讨论、心得

我觉得本次实验设计得颇为合理，步骤不是很多，wireshark 的抓包也很有趣，在实践中获得知识。不过截图保存这一步还是有点麻烦的。

我在第五步（FTP）实验时有点卡住了：因为疫情原因不在学校，一时间找不到可以连的 FTP。而且我发现，如果登陆了 VPN 再实验就抓不到 FTP 的协议了。最后还是登陆了一个同学个人的 FTP 服务器才解决了这个问题。

实验报告里的后面几个问题有点难，要是比较有专业的讲解就好了（wireshark 的记录有点凌乱，分析起来不是很确定）。

还有一个想法：光是免费的 Wireshark 软件功能都很强大（像 FTP, SMTP, IMAP 这种协议都可以明文截取），更别说一些更专业的软件了。所以我十分担心公民的隐私安全。