中山大學
SUN YAT-SEN UNIVERSITY

计算机网络实验报告

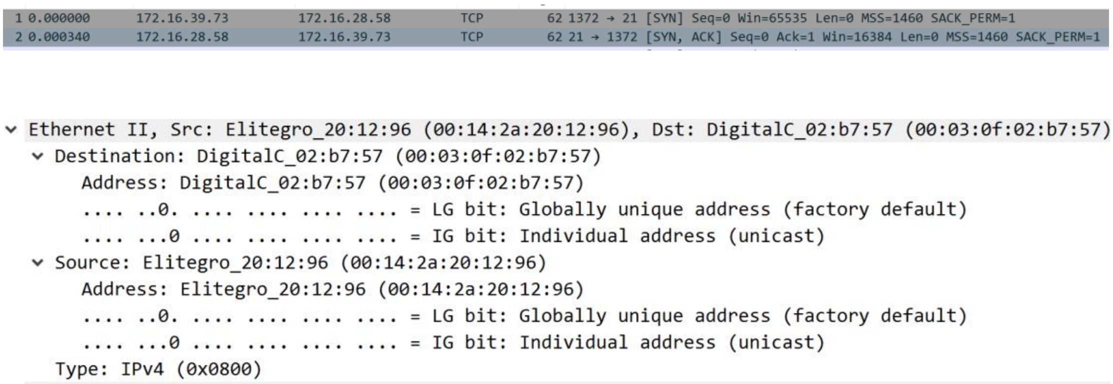
警示

1. 实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2. 当次小组成员成绩只计学号、姓名登录在下表中的。
3. 在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4. 实验报告文件以 PDF 格式提交。

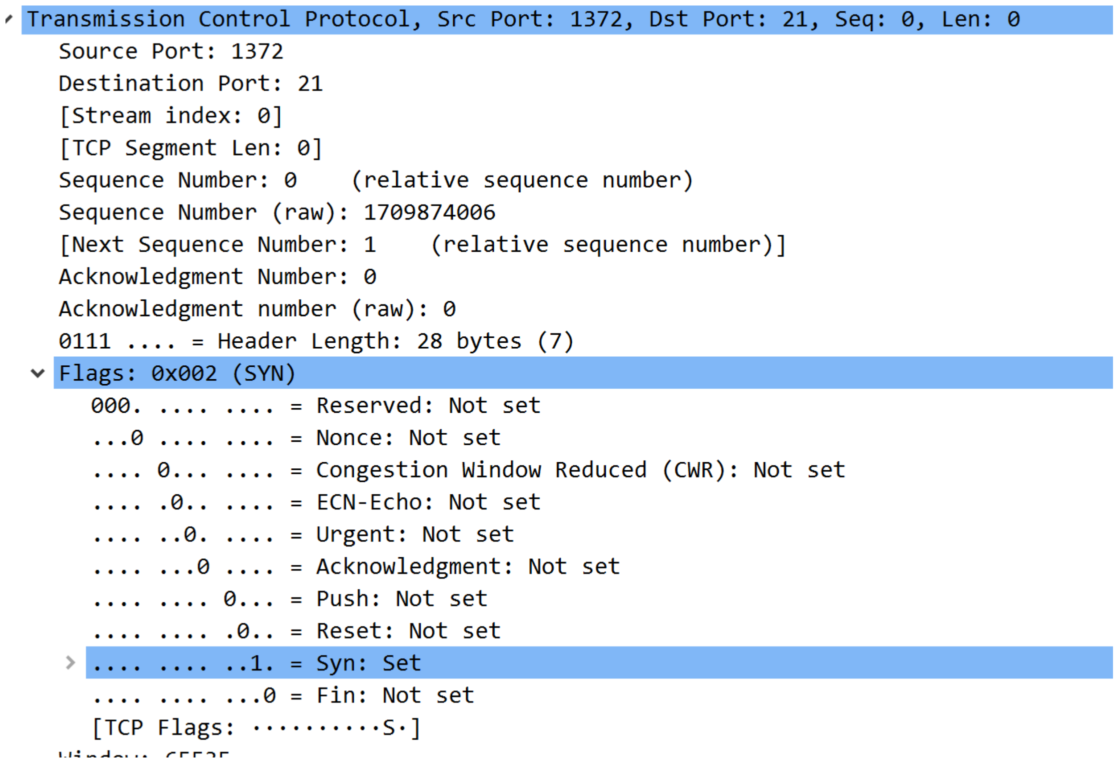
20337251 伍建霖

Ftp 协议分析实验

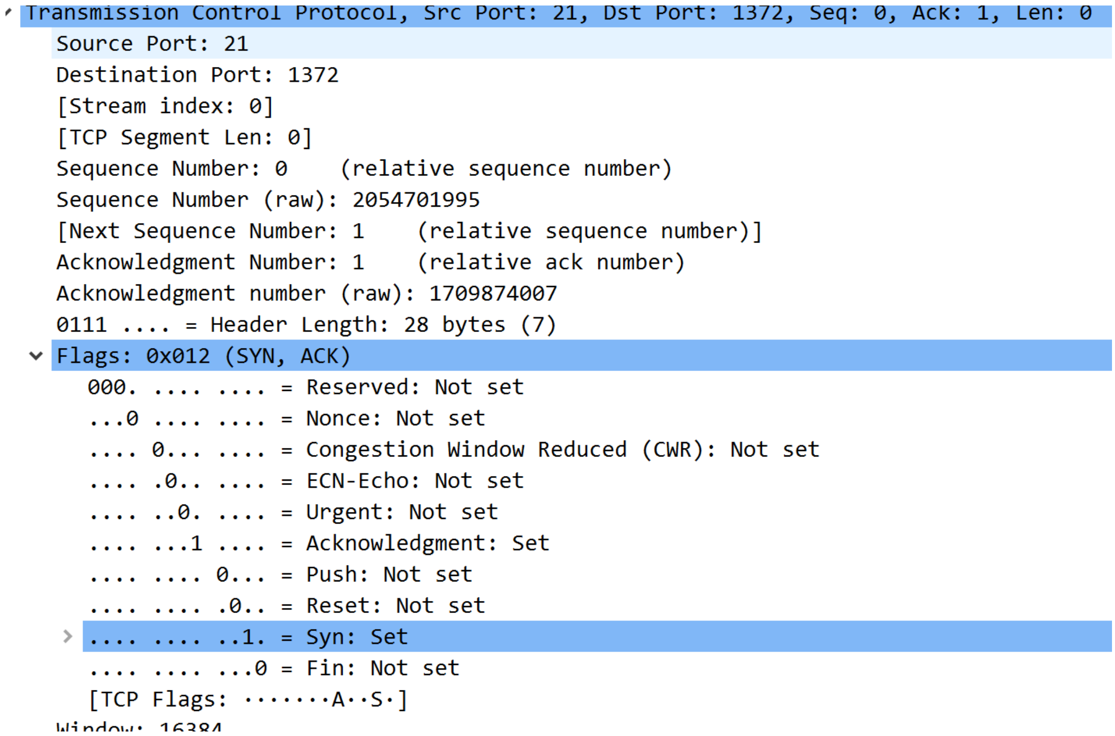
一、打开“FTP 数据包”的“ftp 例 1.cap”文件，进行观察分析，回答以下问题(见附件)

题号	
1	FTP 客户端的 mac 地址是多少?
答案	00:14:2a:20:12:96
截图	 1 0.000000 172.16.39.73 172.16.28.58 TCP 62 1372 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 2 0.000340 172.16.28.58 172.16.39.73 TCP 62 21 → 1372 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 Ethernet II, Src: Elitegro_20:12:96 (00:14:2a:20:12:96), Dst: DigitalC_02:b7:57 (00:03:0f:02:b7:57) Destination: DigitalC_02:b7:57 (00:03:0f:02:b7:57) Address: DigitalC_02:b7:57 (00:03:0f:02:b7:57) .... ..0. .... = LG bit: Globally unique address (factory default) ....0 .... = IG bit: Individual address (unicast) Source: Elitegro_20:12:96 (00:14:2a:20:12:96) Address: Elitegro_20:12:96 (00:14:2a:20:12:96) .... ..0. .... = LG bit: Globally unique address (factory default) ....0 .... = IG bit: Individual address (unicast) Type: IPv4 (0x0800)
分析	可以看出，前两号报文先由客户端发出请求连接，然后服务端响应。看第一号报文的数据链路层协议，找 src。
2	第 1、2、3 号报文的作用是什么?
答案	三次握手，建立连接。客户端先发出建立连接请求，服务端收到并确认，客户端再确认。

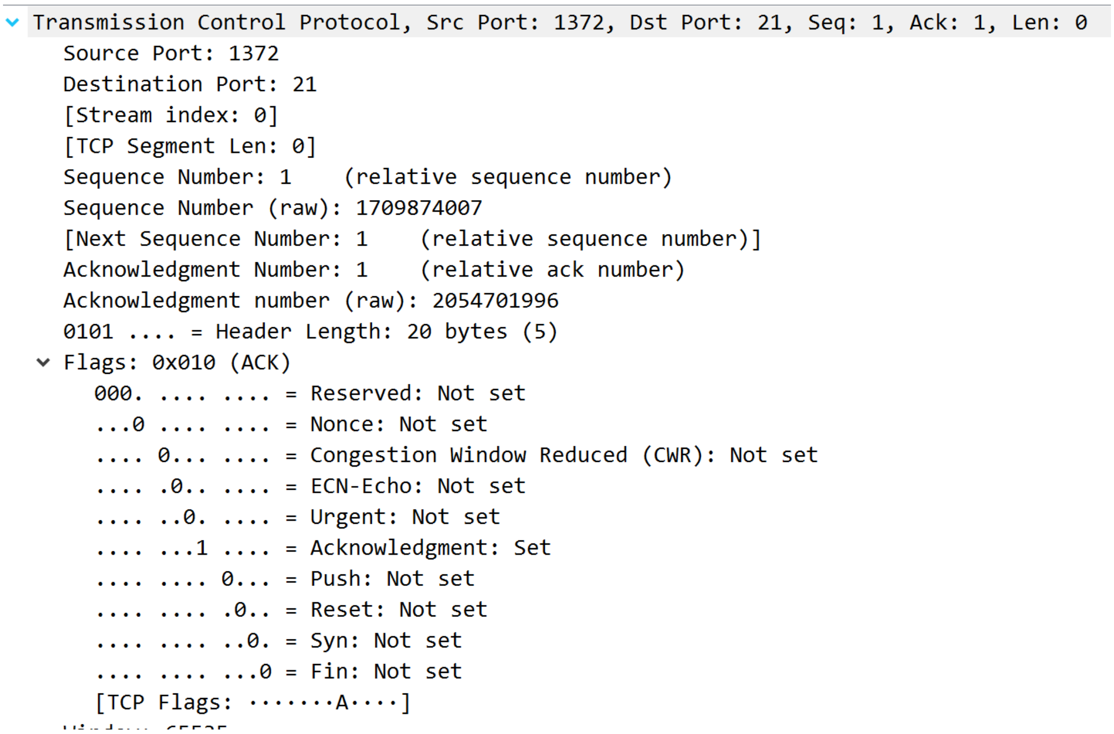
截图



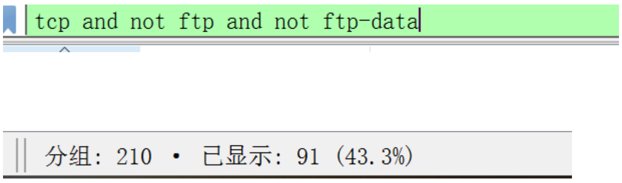
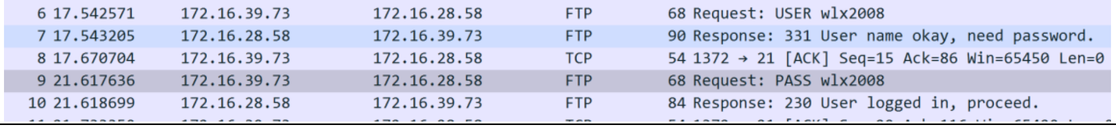
```
Transmission Control Protocol, Src Port: 1372, Dst Port: 21, Seq: 0, Len: 0
    Source Port: 1372
    Destination Port: 21
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 1709874006
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    0111 .... = Header Length: 28 bytes (7)
  Flags: 0x002 (SYN)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...0 .... = Acknowledgment: Not set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
      > .... .... ..1. = Syn: Set
        .... .... ...0 = Fin: Not set
        [TCP Flags: ··········S·]
    Window: 65535
```

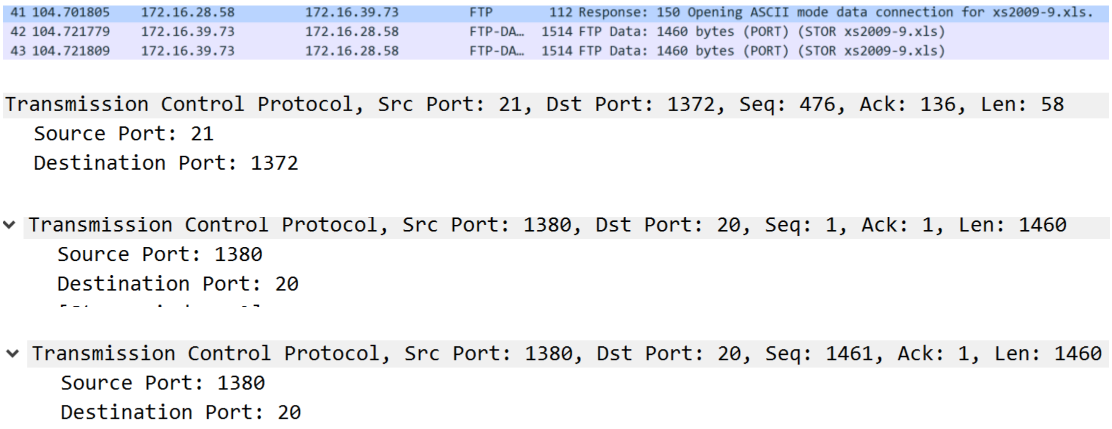
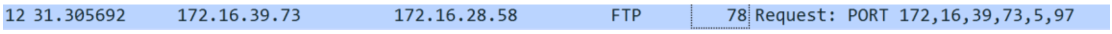
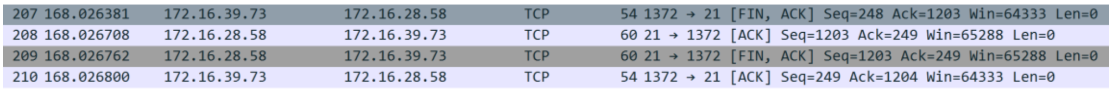
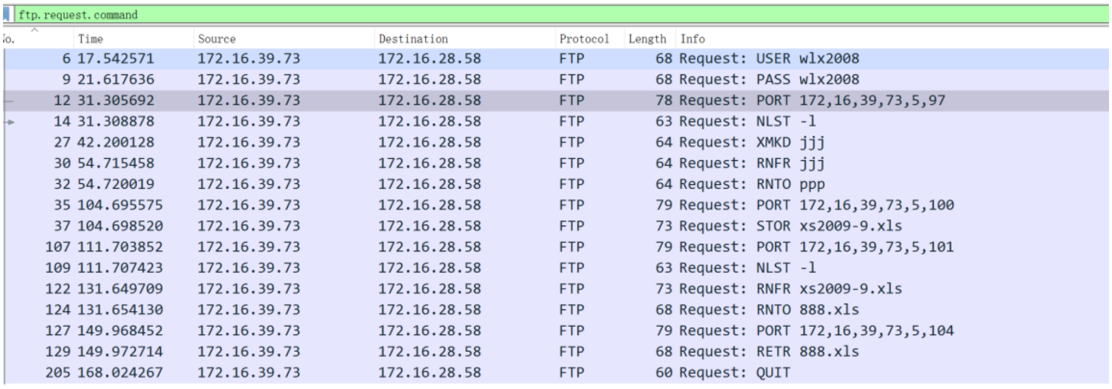


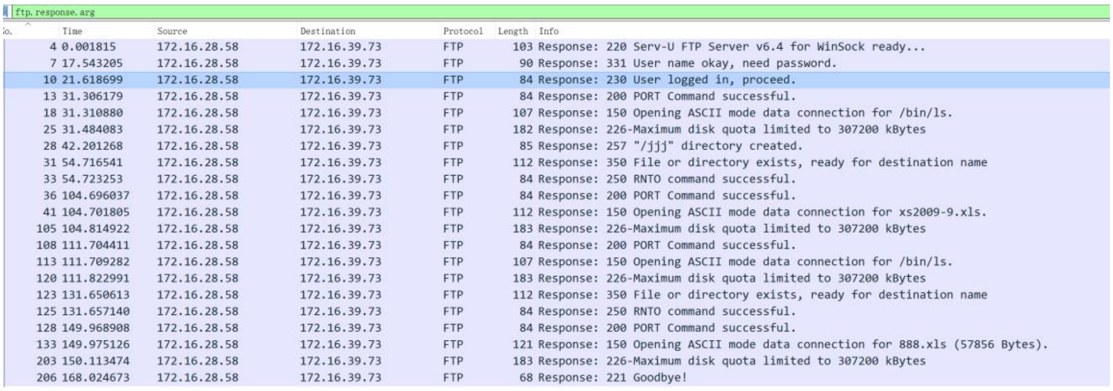
```
Transmission Control Protocol, Src Port: 21, Dst Port: 1372, Seq: 0, Ack: 1, Len: 0
    Source Port: 21
    Destination Port: 1372
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2054701995
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1709874007
    0111 .... = Header Length: 28 bytes (7)
  Flags: 0x012 (SYN, ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
      > .... .... ..1. = Syn: Set
        .... .... ...0 = Fin: Not set
        [TCP Flags: ·······A··S·]
    Window: 16384
```



```
v Transmission Control Protocol, Src Port: 1372, Dst Port: 21, Seq: 1, Ack: 1, Len: 0
    Source Port: 1372
    Destination Port: 21
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1709874007
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2054701996
    0101 .... = Header Length: 20 bytes (5)
  v 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····]
```

分析	第一号有 syn 信号，是客户端发送的请求连接的报文。第二号有 syn 和 ack 信号，是服务端接收到并发出请求连接的报文。第三号有 ack 信号，是客户端收到了服务端的请求的报文。这样就完成了连接过程。
3	该数据包中共有多少个 TCP 流？
答案	91
截图	 tcp and not ftp and not ftp-data 分组：210 • 已显示：91（43.3%）
分析	过滤器过滤条件如图，可以看出有 91 个 TCP 流。因为 FTP 和 FTP-data 是应用层，建立在 TCP 传输层的基础之上，故只过滤 TCP 的话仍然有 210 个。
4	用什么用户和密码登录成功？
答案	用户名：wlx2008，密码：wlx2008
截图	 6 17.542571 172.16.39.73 172.16.28.58 FTP 68 Request: USER wlx2008 7 17.543205 172.16.28.58 172.16.39.73 FTP 90 Response: 331 User name okay, need password. 8 17.670704 172.16.39.73 172.16.28.58 TCP 54 1372 → 21 [ACK] Seq=15 Ack=86 Win=65450 Len=0 9 21.617636 172.16.39.73 172.16.28.58 FTP 68 Request: PASS wlx2008 10 21.618699 172.16.28.58 172.16.39.73 FTP 84 Response: 230 User logged in, proceed.
分析	从 6.7 号报文看出客户端的用户名成功传给了服务端，服务端请求密码。9.10 号看出客户端的密码传送给了服务端，服务端校验成功。
5	该 FTP 的命令连接和数据连接分别是什么样的连接？
答案	命令连接服务端端口号为 21，客户端端口号是固定的，是客户端和服务器的指令交流。数据连接都是主动连接，服务器端口号为 20，客户端端口号不是固定的，是客户端和服务器的数据交流。

截图	 41 104.701805 172.16.28.58 172.16.39.73 FTP 112 Response: 150 Opening ASCII mode data connection for xs2009-9.xls. 42 104.721779 172.16.39.73 172.16.28.58 FTP-DA… 1514 FTP Data: 1460 bytes (PORT) (STOR xs2009-9.xls) 43 104.721809 172.16.39.73 172.16.28.58 FTP-DA… 1514 FTP Data: 1460 bytes (PORT) (STOR xs2009-9.xls) Transmission Control Protocol, Src Port: 21, Dst Port: 1372, Seq: 476, Ack: 136, Len: 58 Source Port: 21 Destination Port: 1372 Transmission Control Protocol, Src Port: 1380, Dst Port: 20, Seq: 1, Ack: 1, Len: 1460 Source Port: 1380 Destination Port: 20 Transmission Control Protocol, Src Port: 1380, Dst Port: 20, Seq: 1461, Ack: 1, Len: 1460 Source Port: 1380 Destination Port: 20
分析	根据服务端端口号和协议类型 FTP 或 FTP-data 可以看出 41 号是服务器的应答指令，42 和 43 是客户端上传数据的报文。
6	该 FTP 的连接模式是那种？为什么？
答案	主动连接
截图	 12 31.305692 172.16.39.73 172.16.28.58 FTP 78 Request: PORT 172,16,39,73,5,97
分析	根据客户端 FTP 请求的端口为 PORT 方式，可以看出是主动连接。
7	最后四个报文的作用是什么？
答案	断开连接
截图	 207 168.026381 172.16.39.73 172.16.28.58 TCP 54 1372 → 21 [FIN, ACK] Seq=248 Ack=1203 Win=64333 Len=0 208 168.026708 172.16.28.58 172.16.39.73 TCP 60 21 → 1372 [ACK] Seq=1203 Ack=249 Win=65288 Len=0 209 168.026762 172.16.28.58 172.16.39.73 TCP 60 21 → 1372 [FIN, ACK] Seq=1203 Ack=249 Win=65288 Len=0 210 168.026800 172.16.39.73 172.16.28.58 TCP 54 1372 → 21 [ACK] Seq=249 Ack=1204 Win=64333 Len=0
分析	207 是客户端向服务端发送断开连接请求 FIN，208 是服务端收到该请求，209 是服务端向客户端发送断开连接请求 FIN，210 是客户端收到该请求。之后就断开了连接。
8	该数据包中有多少个 ftp 的命令及应答，其含义分别是什么？
答案	16 个命令，21 个应答
截图	 ftp.request.command No. Time Source Destination Protocol Length Info 6 17.542571 172.16.39.73 172.16.28.58 FTP 68 Request: USER wlx2008 9 21.617636 172.16.39.73 172.16.28.58 FTP 68 Request: PASS wlx2008 12 31.305692 172.16.39.73 172.16.28.58 FTP 78 Request: PORT 172,16,39,73,5,97 14 31.308878 172.16.39.73 172.16.28.58 FTP 63 Request: NLST -l 27 42.200128 172.16.39.73 172.16.28.58 FTP 64 Request: XMKD jjj 30 54.715458 172.16.39.73 172.16.28.58 FTP 64 Request: RNFR jjj 32 54.720019 172.16.39.73 172.16.28.58 FTP 64 Request: RNTO ppp 35 104.695575 172.16.39.73 172.16.28.58 FTP 79 Request: PORT 172,16,39,73,5,100 37 104.698520 172.16.39.73 172.16.28.58 FTP 73 Request: STOR xs2009-9.xls 107 111.703852 172.16.39.73 172.16.28.58 FTP 79 Request: PORT 172,16,39,73,5,101 109 111.707423 172.16.39.73 172.16.28.58 FTP 63 Request: NLST -l 122 131.649709 172.16.39.73 172.16.28.58 FTP 73 Request: RNFR xs2009-9.xls 124 131.654130 172.16.39.73 172.16.28.58 FTP 68 Request: RNTO 888.xls 127 149.968452 172.16.39.73 172.16.28.58 FTP 79 Request: PORT 172,16,39,73,5,104 129 149.972714 172.16.39.73 172.16.28.58 FTP 68 Request: RETR 888.xls 205 168.024267 172.16.39.73 172.16.28.58 FTP 60 Request: QUIT



```
ftp.response.arg
No.  Time        Source         Destination    Protocol  Length  Info
  4 0.001815     172.16.28.58   172.16.39.73   FTP       103 Response: 220 Serv-U FTP Server v6.4 for WinSock ready...
  7 17.543205    172.16.28.58   172.16.39.73   FTP        90 Response: 331 User name okay, need password.
 10 21.618699    172.16.28.58   172.16.39.73   FTP        84 Response: 230 User logged in, proceed.
 13 31.306179    172.16.28.58   172.16.39.73   FTP        84 Response: 200 PORT Command successful.
 18 31.310880    172.16.28.58   172.16.39.73   FTP       107 Response: 150 Opening ASCII mode data connection for /bin/ls.
 25 31.484083    172.16.28.58   172.16.39.73   FTP       182 Response: 226-Maximum disk quota limited to 307200 kBytes
 28 42.201268    172.16.28.58   172.16.39.73   FTP        85 Response: 257 "/jjj" directory created.
 31 54.716541    172.16.28.58   172.16.39.73   FTP       112 Response: 350 File or directory exists, ready for destination name
 33 54.723253    172.16.28.58   172.16.39.73   FTP        84 Response: 250 RNTO command successful.
 36 104.696037   172.16.28.58   172.16.39.73   FTP        84 Response: 200 PORT Command successful.
 41 104.701805   172.16.28.58   172.16.39.73   FTP       112 Response: 150 Opening ASCII mode data connection for xs2009-9.xls.
105 104.814922   172.16.28.58   172.16.39.73   FTP       183 Response: 226-Maximum disk quota limited to 307200 kBytes
108 111.704411   172.16.28.58   172.16.39.73   FTP        84 Response: 200 PORT Command successful.
113 111.709282   172.16.28.58   172.16.39.73   FTP       107 Response: 150 Opening ASCII mode data connection for /bin/ls.
120 111.822991   172.16.28.58   172.16.39.73   FTP       183 Response: 226-Maximum disk quota limited to 307200 kBytes
123 131.650613   172.16.28.58   172.16.39.73   FTP       112 Response: 350 File or directory exists, ready for destination name
125 131.657140   172.16.28.58   172.16.39.73   FTP        84 Response: 250 RNTO command successful.
128 149.968908   172.16.28.58   172.16.39.73   FTP        84 Response: 200 PORT Command successful.
133 149.975126   172.16.28.58   172.16.39.73   FTP       121 Response: 150 Opening ASCII mode data connection for 888.xls (57856 Bytes).
203 150.113474   172.16.28.58   172.16.39.73   FTP       183 Response: 226-Maximum disk quota limited to 307200 kBytes
206 168.024673   172.16.28.58   172.16.39.73   FTP        68 Response: 221 Goodbye!
```

分析

4 服务端说准备就绪

6 客户端发送用户名

7 服务端说用户名正确

9 客户端发送密码

10 服务端说密码正确，登陆成功

12 客户端发送端口号，请求数据连接

13 服务端说成功

14 客户端请求打开那个文件夹

18 服务端打开连接

25 服务端结束数据连接

27 服务端请求：XMKD jjj，表示在服务器上创建指定的目录，目录名为 jjj

28 服务端回应：路径名创建（对应响应码 257）

30 客户端请求：RNFR jjj，表示对 jjj 文件夹进行重命名；

31 服务端回应：先将文件夹内的文件行为关闭（对应响应码 350）

32 客户端请求：RNTO ppp，请求将 jjj 文件夹改名为 ppp

33 服务端回应：文件（改名）行为完成（对应响应码 250）

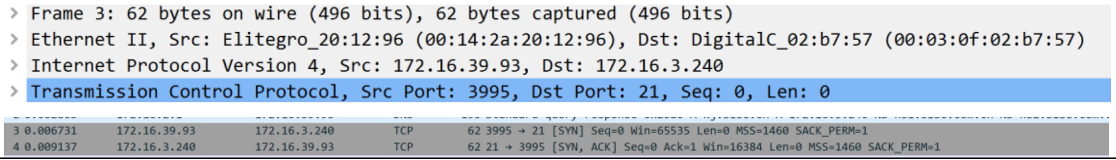
35 客户端请求：PORT 声明当前 IP 地址和端口号：172，16，39，73，5，100

	36 服务端回应：成功 37 客户端请求：STOR 将文件 xs2009-9.xls 上传到服务器上 41 105 服务端回应：成功以 ASCII 编码模式打开文件并创建连接，（完成上传后）并提示服务器磁盘容量 107 客户端请求：PORT 声明当前 IP 地址和端口号：172，16，39，73，5，101 108 服务端回应：PORT 请求成功 109 行命令 NSTL -l:列出目录内容 113 行应答：用 ASCII 的模式打开/bin/ls 文件夹 120 行应答：磁盘还有 307200kBytes 122 行命令：重命名 xs2009-9.xls 文件 123 行应答：350 个文件或目录存在，准备接收目的名字 124 行命令：重命名为 888.xls 125 行应答：重命名成功 127 行命令：向服务器发送客户端 IP 地址和两字节的端口 ID(172,16,39,73,5,104) 128 行应答：发送成功 129 行命令：从服务器上复制文件 888.xls 133 行应答：用 ASCII 的模式连接 888.xls(57856Bytes)文件 203 行应答：磁盘还有 307200kBytes 205 行命令：从 FTP 服务器上退出登录 206 行应答：退出网络，服务器回复：再见

二、打开“FTP 数据包”的“ftp 例 2.cap”文件，进行观察分析，回答以下问题

题号	
1	FTP 服务器的 ip 是多少？FTP 客户端的 mac 地址是多少？
答案	服务器 ip：172.16.3.240 客户端 mac：00:14:2a:20:12:96

截图



```
> Frame 3: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
> Ethernet II, Src: Elitegro_20:12:96 (00:14:2a:20:12:96), Dst: DigitalC_02:b7:57 (00:03:0f:02:b7:57)
> Internet Protocol Version 4, Src: 172.16.39.93, Dst: 172.16.3.240
> Transmission Control Protocol, Src Port: 3995, Dst Port: 21, Seq: 0, Len: 0
3 0.006731   172.16.39.93   172.16.3.240   TCP   62 3995 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
4 0.009137   172.16.3.240   172.16.39.93   TCP   62 21 → 3995 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
```

分析

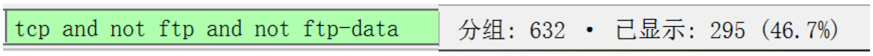
由三号报文，客户端向服务端发送连接请求，得知服务器 ip 地址。打开三号报文，看数据链路层的 src 即为客户端 mac 地址

2 该数据包中共有多少个 TCP 流？

答案

295

截图



```
tcp and not ftp and not ftp-data    分组：632 • 已显示：295（46.7%）
```

分析

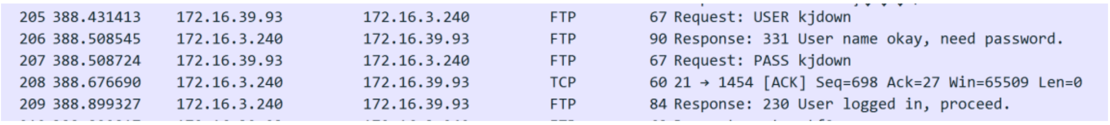
根据此次筛选，可以看出有 295 个 TCP。但是如果只筛选 TCP，则有 630 个。

3 最后用什么用户和密码登录成功？

答案

用户名和密码都是 kjdown

截图



```
205 388.431413   172.16.39.93   172.16.3.240   FTP   67 Request: USER kjdown
206 388.508545   172.16.3.240   172.16.39.93   FTP   90 Response: 331 User name okay, need password.
207 388.508724   172.16.39.93   172.16.3.240   FTP   67 Request: PASS kjdown
208 388.676690   172.16.3.240   172.16.39.93   TCP   60 21 → 1454 [ACK] Seq=698 Ack=27 Win=65509 Len=0
209 388.899327   172.16.3.240   172.16.39.93   FTP   84 Response: 230 User logged in, proceed.
```

分析

可以看出只有这个用户名和密码，服务器端才显示登录成功。

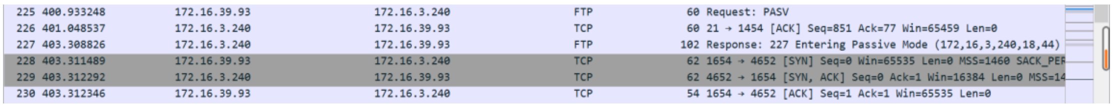
4 该 FTP 的命令连接和数据连接分别是什么？

答案

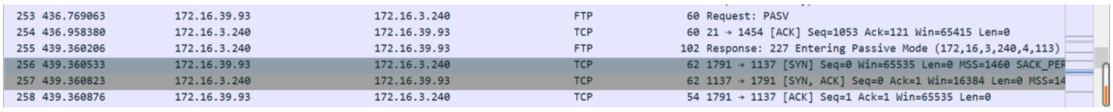
控制连接：客户端准备与 FTP 服务器建立数据传输时,它首先向服务器的 TCP 21 端口发起一个建立连接的请求（参考 225 行报文）,FTP 服务器接受来自客户端的请求（226，227 行报文）,完成连接的建立过程。

数据连接：FTP 控制连接建立之后（需要三次握手，参考 228，229，230 行报文）,即可开始接受文件处理指令和传输文件。

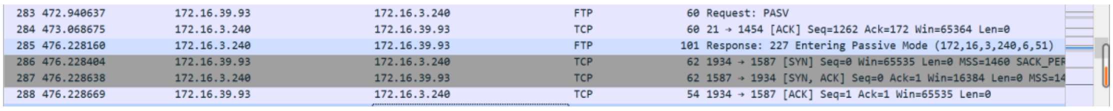
截图



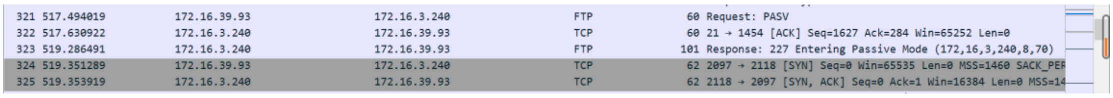
```
225 400.933248   172.16.39.93   172.16.3.240   FTP    60 Request: PASV
226 401.048537   172.16.3.240   172.16.39.93   TCP    60 21 → 1454 [ACK] Seq=851 Ack=77 Win=65459 Len=0
227 403.308826   172.16.3.240   172.16.39.93   FTP   102 Response: 227 Entering Passive Mode (172,16,3,240,18,44)
228 403.311489   172.16.39.93   172.16.3.240   TCP    62 1654 → 4652 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PER
229 403.312292   172.16.3.240   172.16.39.93   TCP    62 4652 → 1654 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=14
230 403.312346   172.16.39.93   172.16.3.240   TCP    54 1654 → 4652 [ACK] Seq=1 Ack=1 Win=65535 Len=0
```



```
253 436.769063   172.16.39.93   172.16.3.240   FTP    60 Request: PASV
254 436.958380   172.16.3.240   172.16.39.93   TCP    60 21 → 1454 [ACK] Seq=1053 Ack=121 Win=65415 Len=0
255 439.360206   172.16.3.240   172.16.39.93   FTP   102 Response: 227 Entering Passive Mode (172,16,3,240,4,113)
256 439.360533   172.16.39.93   172.16.3.240   TCP    62 1791 → 1137 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PER
257 439.360823   172.16.3.240   172.16.39.93   TCP    62 1137 → 1791 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=14
258 439.360876   172.16.39.93   172.16.3.240   TCP    54 1791 → 1137 [ACK] Seq=1 Ack=1 Win=65535 Len=0
```



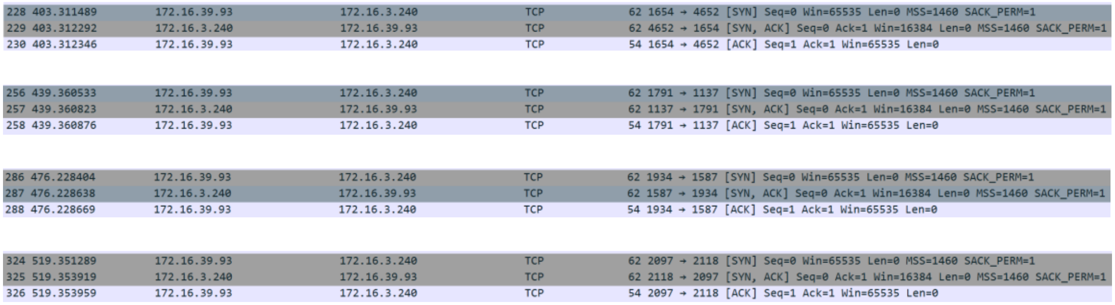
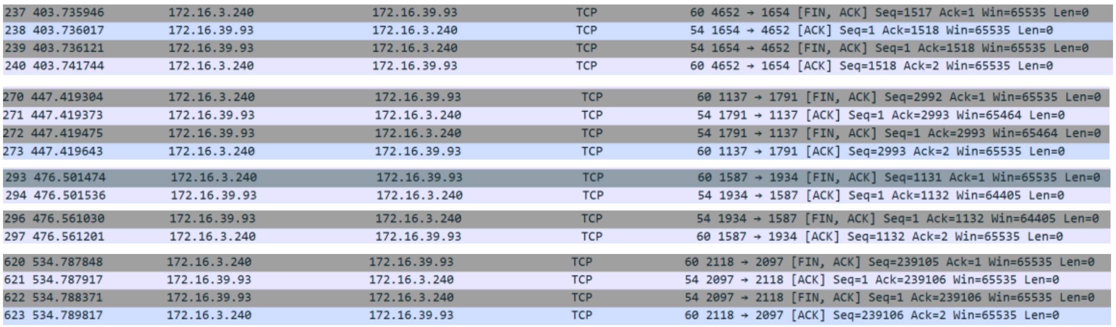
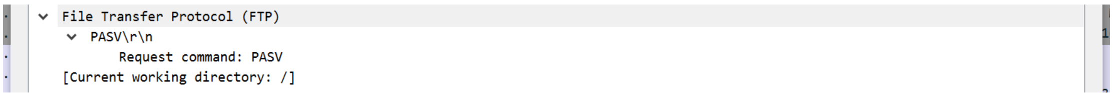
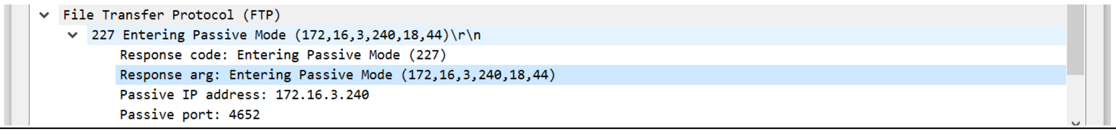
```
283 472.940637   172.16.39.93   172.16.3.240   FTP    60 Request: PASV
284 473.068675   172.16.3.240   172.16.39.93   TCP    60 21 → 1454 [ACK] Seq=1262 Ack=172 Win=65364 Len=0
285 476.228160   172.16.3.240   172.16.39.93   FTP   101 Response: 227 Entering Passive Mode (172,16,3,240,6,51)
286 476.228404   172.16.39.93   172.16.3.240   TCP    62 1934 → 1587 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PER
287 476.228638   172.16.3.240   172.16.39.93   TCP    62 1587 → 1934 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=14
288 476.228669   172.16.39.93   172.16.3.240   TCP    54 1934 → 1587 [ACK] Seq=1 Ack=1 Win=65535 Len=0
```



```
321 517.494019   172.16.39.93   172.16.3.240   FTP    60 Request: PASV
322 517.630922   172.16.3.240   172.16.39.93   TCP    60 21 → 1454 [ACK] Seq=1627 Ack=284 Win=65252 Len=0
323 519.286491   172.16.3.240   172.16.39.93   FTP   101 Response: 227 Entering Passive Mode (172,16,3,240,8,70)
324 519.351289   172.16.39.93   172.16.3.240   TCP    62 2097 → 2118 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PER
325 519.353919   172.16.3.240   172.16.39.93   TCP    62 2118 → 2097 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=14
```

分析

如上截图，整个过程中，以被动模式连接数据了 4 次，并且只有在最后一次才进行了文件的下载，前三次以文件查询等操作为主。

5	哪几个报文是 FTP 数据连接的三次握手报文?
答案	1.228、229、230 行；2.256、257、258 行；3.286、287、288 行；4.324、325、326 行
截图	 228 403.311489 172.16.39.93 172.16.3.240 TCP 62 1654 → 4652 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 229 403.312292 172.16.3.240 172.16.39.93 TCP 62 4652 → 1654 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 230 403.312346 172.16.39.93 172.16.3.240 TCP 54 1654 → 4652 [ACK] Seq=1 Ack=1 Win=65535 Len=0 256 439.360533 172.16.39.93 172.16.3.240 TCP 62 1791 → 1137 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 257 439.360823 172.16.3.240 172.16.39.93 TCP 62 1137 → 1791 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 258 439.360876 172.16.39.93 172.16.3.240 TCP 54 1791 → 1137 [ACK] Seq=1 Ack=1 Win=65535 Len=0 286 476.228404 172.16.39.93 172.16.3.240 TCP 62 1934 → 1587 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 287 476.228638 172.16.3.240 172.16.39.93 TCP 62 1587 → 1934 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 288 476.228669 172.16.39.93 172.16.3.240 TCP 54 1934 → 1587 [ACK] Seq=1 Ack=1 Win=65535 Len=0 324 519.351289 172.16.39.93 172.16.3.240 TCP 62 2097 → 2118 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 325 519.353919 172.16.3.240 172.16.39.93 TCP 62 2118 → 2097 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 326 519.353959 172.16.39.93 172.16.3.240 TCP 54 2097 → 2118 [ACK] Seq=1 Ack=1 Win=65535 Len=0
分析	四次数据连接都是被动连接，客户端端口号分别是 1654、1791、1934、2097，服务器端口号分别是 4652、1137、1587、2118
6	哪几个报文是 FTP 数据连接的挥手报文（结束报文）？
答案	1.237、238、239、240 行；2.270、271、272、273 行；3.293、295、296、297 行；4.620、621、622、623 行
截图	 237 403.735946 172.16.3.240 172.16.39.93 TCP 60 4652 → 1654 [FIN, ACK] Seq=1517 Ack=1 Win=65535 Len=0 238 403.736017 172.16.39.93 172.16.3.240 TCP 54 1654 → 4652 [ACK] Seq=1 Ack=1518 Win=65535 Len=0 239 403.736121 172.16.39.93 172.16.3.240 TCP 54 1654 → 4652 [FIN, ACK] Seq=1 Ack=1518 Win=65535 Len=0 240 403.741744 172.16.3.240 172.16.39.93 TCP 60 4652 → 1654 [ACK] Seq=1518 Ack=2 Win=65535 Len=0 270 447.419304 172.16.3.240 172.16.39.93 TCP 60 1137 → 1791 [FIN, ACK] Seq=2992 Ack=1 Win=65535 Len=0 271 447.419373 172.16.39.93 172.16.3.240 TCP 54 1791 → 1137 [ACK] Seq=1 Ack=2993 Win=65464 Len=0 272 447.419475 172.16.39.93 172.16.3.240 TCP 54 1791 → 1137 [FIN, ACK] Seq=1 Ack=2993 Win=65464 Len=0 273 447.419643 172.16.3.240 172.16.39.93 TCP 60 1137 → 1791 [ACK] Seq=2993 Ack=2 Win=65535 Len=0 293 476.501474 172.16.3.240 172.16.39.93 TCP 60 1587 → 1934 [FIN, ACK] Seq=1131 Ack=1 Win=65535 Len=0 294 476.501536 172.16.39.93 172.16.3.240 TCP 54 1934 → 1587 [ACK] Seq=1 Ack=1132 Win=64405 Len=0 296 476.561030 172.16.39.93 172.16.3.240 TCP 54 1934 → 1587 [FIN, ACK] Seq=1 Ack=1132 Win=64405 Len=0 297 476.561201 172.16.3.240 172.16.39.93 TCP 60 1587 → 1934 [ACK] Seq=1132 Ack=2 Win=65535 Len=0 620 534.787848 172.16.3.240 172.16.39.93 TCP 60 2118 → 2097 [FIN, ACK] Seq=239105 Ack=1 Win=65535 Len=0 621 534.787917 172.16.39.93 172.16.3.240 TCP 54 2097 → 2118 [ACK] Seq=1 Ack=239106 Win=65535 Len=0 622 534.788371 172.16.39.93 172.16.3.240 TCP 54 2097 → 2118 [FIN, ACK] Seq=1 Ack=239106 Win=65535 Len=0 623 534.789817 172.16.3.240 172.16.39.93 TCP 60 2118 → 2097 [ACK] Seq=239106 Ack=2 Win=65535 Len=0
分析	四次数据连接都是被动连接，客户端端口号分别是 1654、1791、1934、2097，服务器端口号分别是 4652、1137、1587、2118
7	该 FTP 的连接模式是那种？为什么？
答案	FTP 的连接模式是被动模式（Pasv 模式），因为客户端向服务端发送了 Pasv 命令（例如第 225 行报文），并于 227 行报文，服务器向客户端发送了“Entering Passive Mode (172,16,3,240,18,44)”，表明进入了被动连接模式。
截图	第 225 例报文：  File Transfer Protocol (FTP) PASV\r\n Request command: PASV [Current working directory: /] 第 227 例报文：  File Transfer Protocol (FTP) 227 Entering Passive Mode (172,16,3,240,18,44)\r\n Response code: Entering Passive Mode (227) Response arg: Entering Passive Mode (172,16,3,240,18,44) Passive IP address: 172.16.3.240 Passive port: 4652
分析	我们选取其中一个例子： 225 行报文：客户端向服务端发送了 Pasv 命令，请求开启被动方式的数据连接

	226 行报文：服务器确认，并随即打开了一个高级端口：1454. 227 行报文：服务器向客户端发送“Entering Passive Mode”，进入被动连接模式

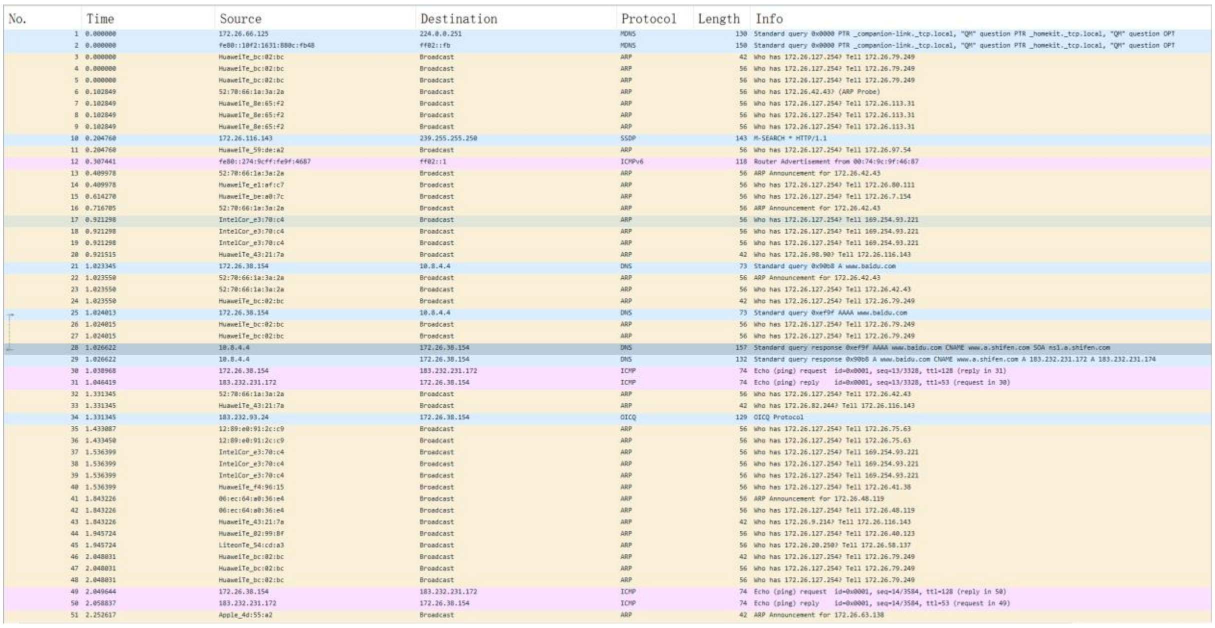
三、在线捕获数据包实验

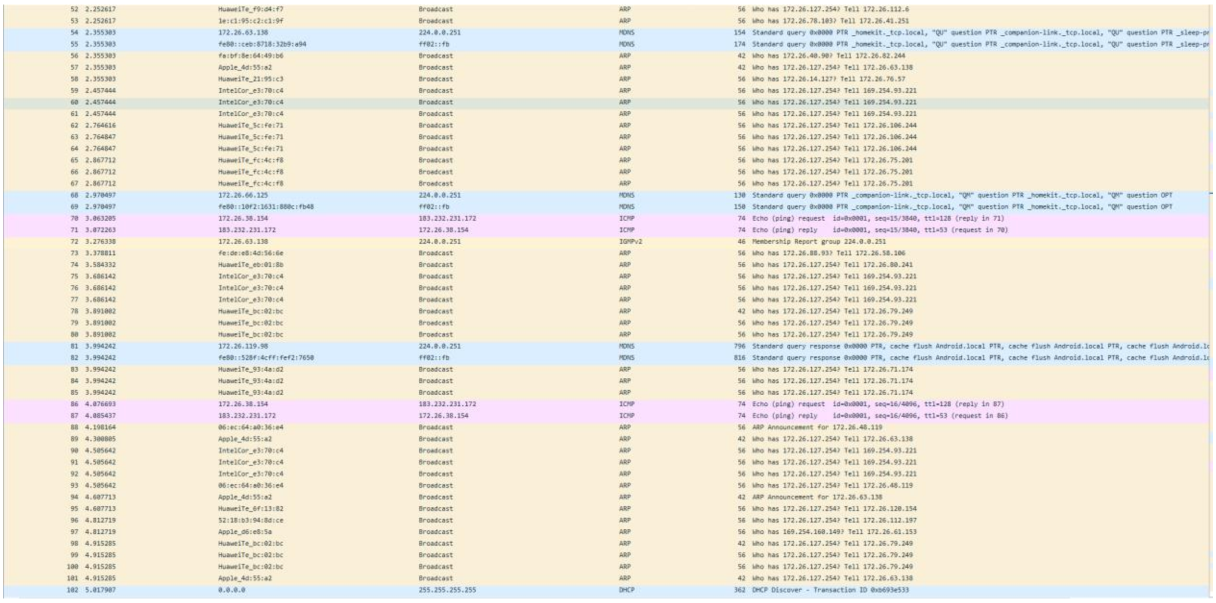
1. 阅读教材 P64-69 内容，熟悉 FTP 协议。

2. 完成 P51 的实例 2-1。

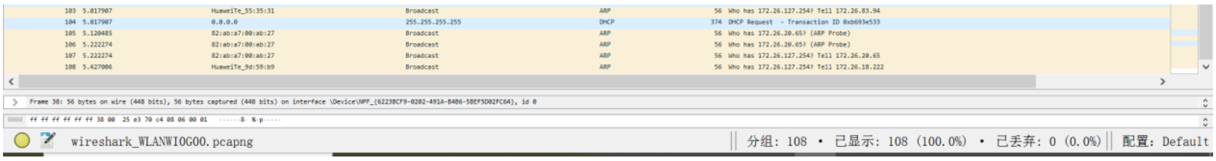
实验内容：

1.侦听捕获的数据量：





No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.26.66.125	224.0.0.251	MDNS	130	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _homekit._tcp.local, "QM" question OPT
2	0.000000	fe80::10f2:1631:880c:fb48	ff02::fb	MDNS	150	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _homekit._tcp.local, "QM" question OPT
3	0.000000	HuaweiTe_bc:02:bc	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.79.249
4	0.000000	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
5	0.000000	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
6	0.102849	52:70:66:1a:3a:2a	Broadcast	ARP	56	Who has 172.26.42.43? (ARP Probe)
7	0.102849	HuaweiTe_8e:65:f2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.113.31
8	0.102849	HuaweiTe_8e:65:f2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.113.31
9	0.102849	HuaweiTe_8e:65:f2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.113.31
10	0.204760	172.26.116.143	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
11	0.204760	HuaweiTe_59:de:a2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.97.54
12	0.307441	fe80::274:9cff:fe9f:4687	ff02::1	ICMPv6	118	Router Advertisement from 00:74:9c:9f:46:87
13	0.409978	52:70:66:1a:3a:2a	Broadcast	ARP	56	ARP Announcement for 172.26.42.43
14	0.409978	HuaweiTe_e1:af:c7	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.80.111
15	0.614270	HuaweiTe_be:a0:7c	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.7.154
16	0.716705	52:70:66:1a:3a:2a	Broadcast	ARP	56	ARP Announcement for 172.26.42.43
17	0.921298	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
18	0.921298	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
19	0.921298	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
20	0.921515	HuaweiTe_43:21:7a	Broadcast	ARP	42	Who has 172.26.98.90? Tell 172.26.116.143
21	1.023345	172.26.38.154	10.8.4.4	DNS	73	Standard query 0x90b8 A www.baidu.com
22	1.023550	52:70:66:1a:3a:2a	Broadcast	ARP	56	ARP Announcement for 172.26.42.43
23	1.023550	52:70:66:1a:3a:2a	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.42.43
24	1.023550	HuaweiTe_bc:02:bc	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.79.249
25	1.024013	172.26.38.154	10.8.4.4	DNS	73	Standard query 0xef9f AAAA www.baidu.com
26	1.024015	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
27	1.024015	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
28	1.026622	10.8.4.4	172.26.38.154	DNS	157	Standard query response 0xef9f AAAA www.baidu.com CNAME www.a.shifen.com SOA ns1.a.shifen.com
29	1.026622	10.8.4.4	172.26.38.154	DNS	132	Standard query response 0x90b8 A www.baidu.com CNAME www.a.shifen.com A 183.232.231.172 A 183.232.231.174
30	1.038968	172.26.38.154	183.232.231.172	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 31)
31	1.046419	183.232.231.172	172.26.38.154	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=53 (request in 30)
32	1.331345	52:70:66:1a:3a:2a	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.42.43
33	1.331345	HuaweiTe_43:21:7a	Broadcast	ARP	42	Who has 172.26.82.244? Tell 172.26.116.143
34	1.331345	183.232.93.24	172.26.38.154	OICQ	129	OICQ Protocol
35	1.433087	12:89:e0:91:2c:c9	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.75.63
36	1.433450	12:89:e0:91:2c:c9	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.75.63
37	1.536399	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
38	1.536399	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
39	1.536399	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
40	1.536399	HuaweiTe_f4:96:15	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.41.38
41	1.843226	06:ec:64:a0:36:e4	Broadcast	ARP	56	ARP Announcement for 172.26.48.119
42	1.843226	06:ec:64:a0:36:e4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.48.119
43	1.843226	HuaweiTe_43:21:7a	Broadcast	ARP	42	Who has 172.26.9.214? Tell 172.26.116.143
44	1.945724	HuaweiTe_02:99:8f	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.40.123
45	1.945724	LiteonTe_54:cd:a3	Broadcast	ARP	56	Who has 172.26.20.250? Tell 172.26.58.137
46	2.048031	HuaweiTe_bc:02:bc	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.79.249
47	2.048031	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
48	2.048031	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
49	2.049644	172.26.38.154	183.232.231.172	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 50)
50	2.058837	183.232.231.172	172.26.38.154	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=53 (request in 49)
51	2.252617	Apple_4d:55:a2	Broadcast	ARP	42	ARP Announcement for 172.26.63.138
52	2.252617	HuaweiTe_f9:d4:f7	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.112.6
53	2.252617	1e:c1:95:c2:c1:9f	Broadcast	ARP	56	Who has 172.26.78.103? Tell 172.26.41.251
54	2.355303	172.26.63.138	224.0.0.251	MDNS	154	Standard query 0x0000 PTR _homekit._tcp.local, "QU" question PTR _companion-link._tcp.local, "QU" question PTR _sleep-pr
55	2.355303	fe80::ceb:8718:32b9:a94	ff02::fb	MDNS	174	Standard query 0x0000 PTR _homekit._tcp.local, "QU" question PTR _companion-link._tcp.local, "QU" question PTR _sleep-pr
56	2.355303	fa:bf:8e:64:49:b6	Broadcast	ARP	42	Who has 172.26.40.90? Tell 172.26.82.244
57	2.355303	Apple_4d:55:a2	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.63.138
58	2.355303	HuaweiTe_21:95:c3	Broadcast	ARP	56	Who has 172.26.14.127? Tell 172.26.76.57
59	2.457444	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
60	2.457444	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
61	2.457444	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
62	2.764616	HuaweiTe_5c:fe:71	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.106.244
63	2.764847	HuaweiTe_5c:fe:71	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.106.244
64	2.764847	HuaweiTe_5c:fe:71	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.106.244
65	2.867712	HuaweiTe_fc:4c:f8	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.75.201
66	2.867712	HuaweiTe_fc:4c:f8	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.75.201
67	2.867712	HuaweiTe_fc:4c:f8	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.75.201
68	2.970497	172.26.66.125	224.0.0.251	MDNS	130	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _homekit._tcp.local, "QM" question OPT
69	2.970497	fe80::10f2:1631:880c:fb48	ff02::fb	MDNS	150	Standard query 0x0000 PTR _companion-link._tcp.local, "QM" question PTR _homekit._tcp.local, "QM" question OPT
70	3.063205	172.26.38.154	183.232.231.172	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 71)
71	3.072263	183.232.231.172	172.26.38.154	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=53 (request in 70)
72	3.276338	172.26.63.138	224.0.0.251	IGMPv2	46	Membership Report group 224.0.0.251
73	3.378811	fe:de:e8:4d:56:6e	Broadcast	ARP	56	Who has 172.26.88.93? Tell 172.26.58.106
74	3.584332	HuaweiTe_eb:01:8b	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.80.241
75	3.686142	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
76	3.686142	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
77	3.686142	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
78	3.891002	HuaweiTe_bc:02:bc	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.79.249
79	3.891002	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
80	3.891002	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
81	3.994242	172.26.119.98	224.0.0.251	MDNS	796	Standard query response 0x0000 PTR, cache flush Android.local PTR, cache flush Android.local PTR, cache flush Android.lo
82	3.994242	fe80::528f:4cff:fef2:7650	ff02::fb	MDNS	816	Standard query response 0x0000 PTR, cache flush Android.local PTR, cache flush Android.local PTR, cache flush Android.lo
83	3.994242	HuaweiTe_93:4a:d2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.71.174
84	3.994242	HuaweiTe_93:4a:d2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.71.174
85	3.994242	HuaweiTe_93:4a:d2	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.71.174
86	4.076693	172.26.38.154	183.232.231.172	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 87)
87	4.085437	183.232.231.172	172.26.38.154	ICMP	74	Echo (ping) reply id=0x0001, seq=16/4096, ttl=53 (request in 86)
88	4.198164	06:ec:64:a0:36:e4	Broadcast	ARP	56	ARP Announcement for 172.26.48.119
89	4.300805	Apple_4d:55:a2	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.63.138
90	4.505642	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
91	4.505642	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
92	4.505642	IntelCor_e3:70:c4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 169.254.93.221
93	4.505642	06:ec:64:a0:36:e4	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.48.119
94	4.607713	Apple_4d:55:a2	Broadcast	ARP	42	ARP Announcement for 172.26.63.138
95	4.607713	HuaweiTe_6f:13:82	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.120.154
96	4.812719	52:18:b3:94:8d:ce	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.112.197
97	4.812719	Apple_d6:e8:5a	Broadcast	ARP	56	Who has 169.254.160.149? Tell 172.26.61.153
98	4.915285	HuaweiTe_bc:02:bc	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.79.249
99	4.915285	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
100	4.915285	HuaweiTe_bc:02:bc	Broadcast	ARP	56	Who has 172.26.127.254? Tell 172.26.79.249
101	4.915285	Apple_4d:55:a2	Broadcast	ARP	42	Who has 172.26.127.254? Tell 172.26.63.138
102	5.017907	0.0.0.0	255.255.255.255	DHCP	362	DHCP Discover - Transaction ID 0xb693e533



```
103 5.017907  HuaweiTe_55:35:31  Broadcast        ARP   56  Who has 172.26.127.254? Tell 172.26.83.94
104 5.017907  0.0.0.0            255.255.255.255  DHCP  374 DHCP Request  - Transaction ID 0xb693e533
105 5.120485  82:ab:a7:00:ab:27  Broadcast        ARP   56  Who has 172.26.20.65? (ARP Probe)
106 5.222274  82:ab:a7:00:ab:27  Broadcast        ARP   56  Who has 172.26.20.65? (ARP Probe)
107 5.222274  82:ab:a7:00:ab:27  Broadcast        ARP   56  Who has 172.26.127.254? Tell 172.26.20.65
108 5.427006  HuaweiTe_9d:59:b9  Broadcast        ARP   56  Who has 172.26.127.254? Tell 172.26.18.222

> Frame 38: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{62238CF9-0202-491A-B486-5BEF5D02FC64}, id 0
0000  ff ff ff ff ff ff 38 00  25 e3 70 c4 08 06 00 01   ······8· %·p·····

wireshark_WLANWIOGOO.pcapng    分组: 108 · 已显示: 108 (100.0%) · 已丢弃: 0 (0.0%)    配置: Default
```

总共捕获 108 个分组。

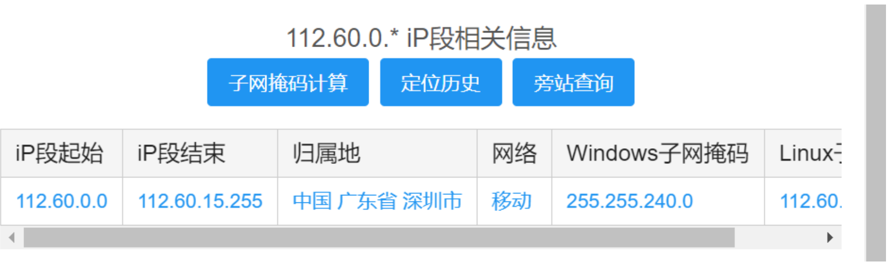
2.

(1)

从图中可以看到，本机 IP 地址为：172.26.38.154，其中第 30 行报文、第 49 行报文、第 70 行报文和第 86 行报文是本机发出去的， 其中第 28 行报文、第 29 行报文、第 31 行报文、第 50 行报文、第 71 行报文和第 87 行报文是本机接收到的报文。

(2)

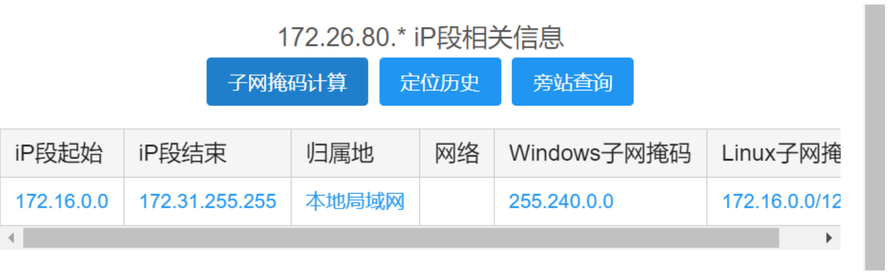
选择 ip 地址：112.60.0.199，通过 IP138 网站进行查询，得到该 IP 地址的地理位置为中国广东省深圳市。



112.60.0.* iP段相关信息

iP段起始	iP段结束	归属地	网络	Windows子网掩码	Linux子
112.60.0.0	112.60.15.255	中国 广东省 深圳市	移动	255.255.240.0	112.60.

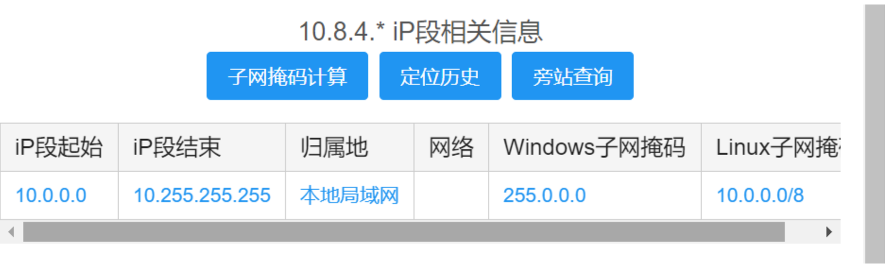
选择 ip 地址：172.26.80.184，通过 IP138 网站进行查询，得到该 IP 地址的地理位置为本地局域网



172.26.80.* iP段相关信息

iP段起始	iP段结束	归属地	网络	Windows子网掩码	Linux子网掩
172.16.0.0	172.31.255.255	本地局域网		255.240.0.0	172.16.0.0/12

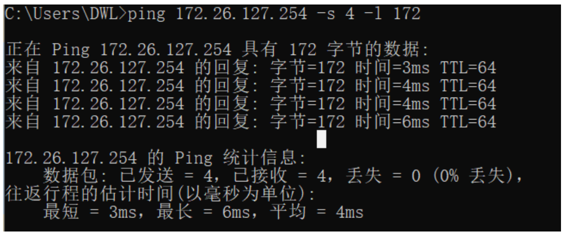
选择 ip 地址：10.8.4.4，通过 IP138 网站进行查询，得到该 IP 地址的地理位置为本地局域网



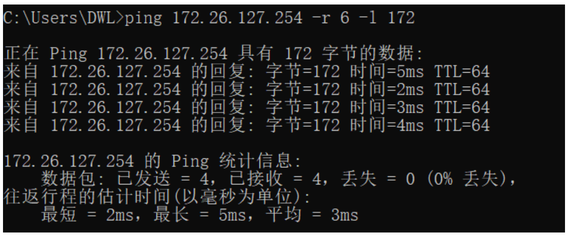
10.8.4.* iP段相关信息

iP段起始	iP段结束	归属地	网络	Windows子网掩码	Linux子网掩
10.0.0.0	10.255.255.255	本地局域网		255.0.0.0	10.0.0.0/8

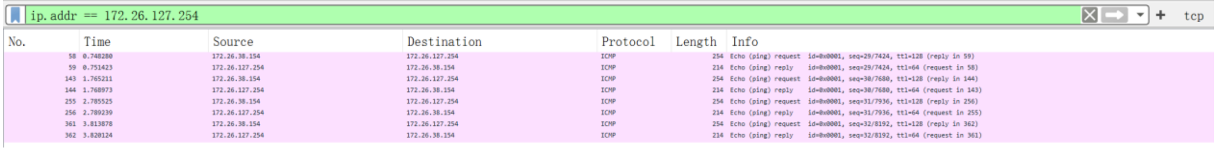
3.(4.) 运行结果截屏：



```
C:\Users\DWL>ping 172.26.127.254 -s 4 -l 172

正在 Ping 172.26.127.254 具有 172 字节的数据:
来自 172.26.127.254 的回复: 字节=172 时间=3ms TTL=64
来自 172.26.127.254 的回复: 字节=172 时间=4ms TTL=64
来自 172.26.127.254 的回复: 字节=172 时间=4ms TTL=64
来自 172.26.127.254 的回复: 字节=172 时间=6ms TTL=64

172.26.127.254 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 = 3ms, 最长 = 6ms, 平均 = 4ms
```

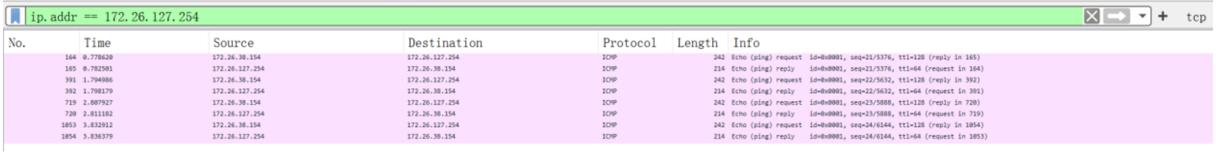


```
C:\Users\DWL>ping 172.26.127.254 -r 6 -l 172

正在 Ping 172.26.127.254 具有 172 字节的数据:
来自 172.26.127.254 的回复: 字节=172 时间=5ms TTL=64
来自 172.26.127.254 的回复: 字节=172 时间=2ms TTL=64
来自 172.26.127.254 的回复: 字节=172 时间=3ms TTL=64
来自 172.26.127.254 的回复: 字节=172 时间=4ms TTL=64

172.26.127.254 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 = 2ms, 最长 = 5ms, 平均 = 3ms
```



ip.addr == 172.26.127.254

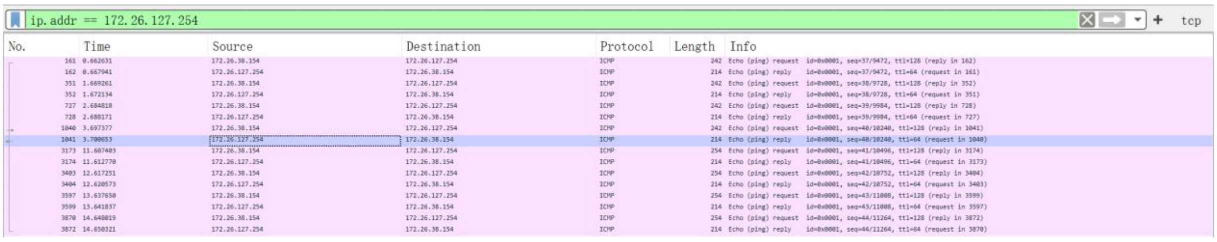
No.	Time	Source	Destination	Protocol	Length	Info
58	0.748280	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=29/7424, ttl=128 (reply in 59)
59	0.751423	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=29/7424, ttl=64 (request in 58)
143	1.765211	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=30/7680, ttl=128 (reply in 144)
144	1.768973	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=30/7680, ttl=64 (request in 143)
255	2.785525	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=31/7936, ttl=128 (reply in 256)
256	2.789239	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=31/7936, ttl=64 (request in 255)
361	3.813878	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=32/8192, ttl=128 (reply in 362)
362	3.820124	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=32/8192, ttl=64 (request in 361)



ip.addr == 172.26.127.254

No.	Time	Source	Destination	Protocol	Length	Info
164	0.778620	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=21/5376, ttl=128 (reply in 165)
165	0.782501	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=21/5376, ttl=64 (request in 164)
391	1.794986	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=22/5632, ttl=128 (reply in 392)
392	1.798179	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=22/5632, ttl=64 (request in 391)
719	2.807927	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=23/5888, ttl=128 (reply in 720)
720	2.811182	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=23/5888, ttl=64 (request in 719)
1053	3.832912	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=24/6144, ttl=128 (reply in 1054)
1054	3.836379	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=24/6144, ttl=64 (request in 1053)

5.

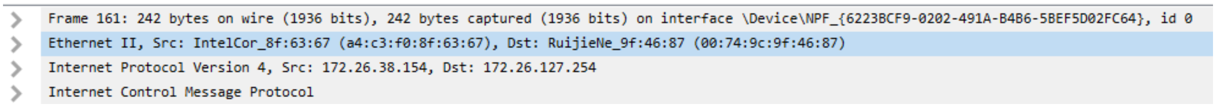


ip.addr == 172.26.127.254

No.	Time	Source	Destination	Protocol	Length	Info
161	0.662631	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=37/9472, ttl=128 (reply in 162)
162	0.667941	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=37/9472, ttl=64 (request in 161)
351	1.669261	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=38/9728, ttl=128 (reply in 352)
352	1.672134	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=38/9728, ttl=64 (request in 351)
727	2.684818	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=39/9984, ttl=128 (reply in 728)
728	2.688171	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=39/9984, ttl=64 (request in 727)
1040	3.697377	172.26.38.154	172.26.127.254	ICMP	242	Echo (ping) request id=0x0001, seq=40/10240, ttl=128 (reply in 1041)
1041	3.700653	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=40/10240, ttl=64 (request in 1040)
3173	11.607403	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=41/10496, ttl=128 (reply in 3174)
3174	11.612770	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=41/10496, ttl=64 (request in 3173)
3403	12.617251	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=42/10752, ttl=128 (reply in 3404)
3404	12.620573	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=42/10752, ttl=64 (request in 3403)
3597	13.637650	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=43/11008, ttl=128 (reply in 3599)
3599	13.641837	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=43/11008, ttl=64 (request in 3597)
3870	14.648019	172.26.38.154	172.26.127.254	ICMP	254	Echo (ping) request id=0x0001, seq=44/11264, ttl=128 (reply in 3872)
3872	14.650321	172.26.127.254	172.26.38.154	ICMP	214	Echo (ping) reply id=0x0001, seq=44/11264, ttl=64 (request in 3870)

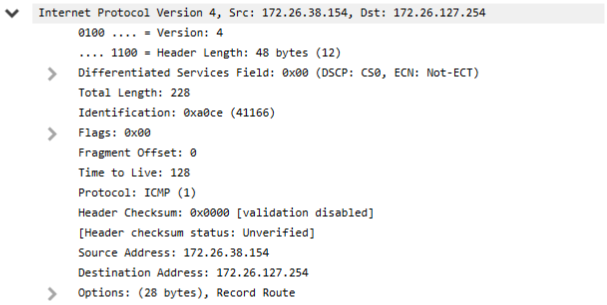
如上图，捕获中的数据的协议都是 ICMP；上图是 Echo 的请求（request）和响应（reply），可以在每一行报文的描述（info）中得到。

下面给出一个报文的截图并分析这个报文的信息：

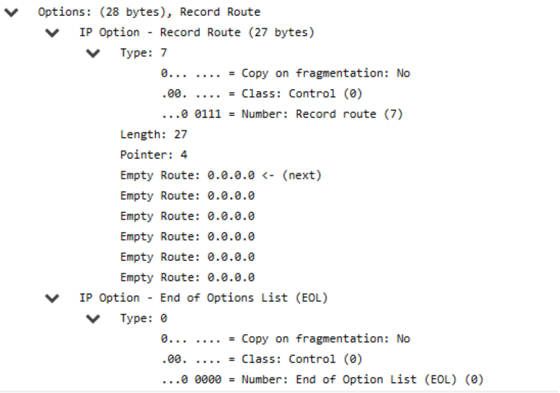


```
> Frame 161: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits) on interface \Device\NPF_{6223BCF9-0202-491A-B4B6-5BEF5D02FC64}, id 0
> Ethernet II, Src: IntelCor_8f:63:67 (a4:c3:f0:8f:63:67), Dst: RuijieNe_9f:46:87 (00:74:9c:9f:46:87)
> Internet Protocol Version 4, Src: 172.26.38.154, Dst: 172.26.127.254
> Internet Control Message Protocol
```

我们可以从截图中得到数据的总长度(242 字节)、源 IP 地址(172.26.38.154)、目的 IP 地址(172.26.127.254)、网络协议信息（ICMP: Internet 控制报文协议, TCP/IP 协议簇的一个子协议，用于在 IP 主机、路由器之间传递控制消息）。



```
v Internet Protocol Version 4, Src: 172.26.38.154, Dst: 172.26.127.254
      0100 .... = Version: 4
      .... 1100 = Header Length: 48 bytes (12)
  >   Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 228
      Identification: 0xa0ce (41166)
  >   Flags: 0x00
      Fragment Offset: 0
      Time to Live: 128
      Protocol: ICMP (1)
      Header Checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 172.26.38.154
      Destination Address: 172.26.127.254
  >   Options: (28 bytes), Record Route
```



```
v Options: (28 bytes), Record Route
    v IP Option - Record Route (27 bytes)
        v Type: 7
              0... .... = Copy on fragmentation: No
              .00. .... = Class: Control (0)
              ...0 0111 = Number: Record route (7)
          Length: 27
          Pointer: 4
          Empty Route: 0.0.0.0 <- (next)
          Empty Route: 0.0.0.0
          Empty Route: 0.0.0.0
          Empty Route: 0.0.0.0
          Empty Route: 0.0.0.0
          Empty Route: 0.0.0.0
    v IP Option - End of Options List (EOL)
        v Type: 0
              0... .... = Copy on fragmentation: No
              .00. .... = Class: Control (0)
              ...0 0000 = Number: End of Option List (EOL) (0)
```

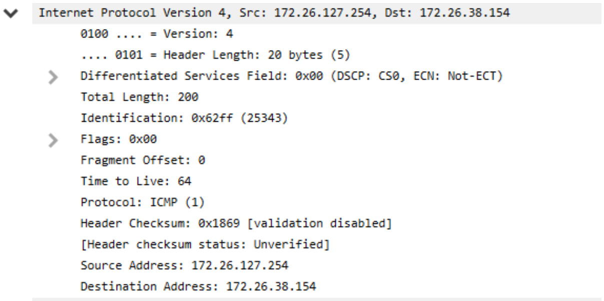
(上图是请求的路由信息)

主要字段含义分析：从图中可以看到

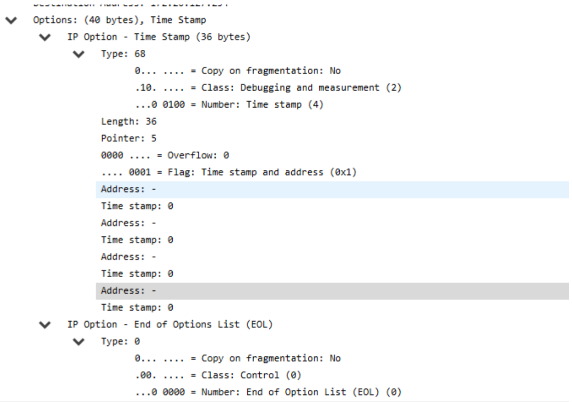
（1）、Echo 请求分组的 IP 版本（Version）为 4，即使用的 IPV4 地址，包头（Header Length）长度为 48 字节；片偏移（Fragment Offset）为 0，表示该 IP 包在该组分片包中位置的 0，接收端靠此来组装还原 IP 包；生存时间（TTL）为 128。

（2）、其源 IP 地址为 172.26.38.154，其目的 IP 地址为 172.26.127.254

（3）、Echo 请求分组的选项（Option）包含记录路由（Record Route）和 End of Options List（用于指示 IP 报头中选项列表的末尾。），而记录路由的目的是让沿途的路由器都将 IP 地址加到可选字段之后，以便跟踪路由选择算法的错误。关于该选项的组成：其 Class 为 0 表示控制，Number 为 7 表示记录路由。



```
Internet Protocol Version 4, Src: 172.26.127.254, Dst: 172.26.38.154
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 200
    Identification: 0x62ff (25343)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x1869 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.26.127.254
    Destination Address: 172.26.38.154
```



```
Options: (40 bytes), Time Stamp
  IP Option - Time Stamp (36 bytes)
    Type: 68
        0... .... = Copy on fragmentation: No
        .10. .... = Class: Debugging and measurement (2)
        ...0 0100 = Number: Time stamp (4)
    Length: 36
    Pointer: 5
    0000 .... = Overflow: 0
    .... 0001 = Flag: Time stamp and address (0x1)
    Address: -
    Time stamp: 0
    Address: -
    Time stamp: 0
    Address: -
    Time stamp: 0
    Address: -
    Time stamp: 0
  IP Option - End of Options List (EOL)
    Type: 0
        0... .... = Copy on fragmentation: No
        .00. .... = Class: Control (0)
        ...0 0000 = Number: End of Option List (EOL) (0)
```

（上图是响应的路由信息）

主要字段含义分析：从图中可以看到

（1）、与 Echo 请求分组相同，Echo 响应分组的 IP 版本（Version）同样为 4，包头（Header Length）长度为 20 字节；片偏移（Fragment Offset）为 0，生存时间（TTL）为 64。

（2）、其源 IP 地址为 172.26.127.254，其目的 IP 地址为 172.26.38.154，与 Echo 请求分组正好相反；

（3）、Echo 请求分组的选项（Option）包含时间戳（Time Stamp）和 End of Options List，而时间戳选项使每台路由器都附上它的 IP 地址和时间标记，在用途上有测量 TCP 连接两端通讯的延迟和处理 Sequence 号反转的问题两种用途。关于该选项的组成：其 Class 为 10 和 Number 为 4 对应网络时间戳。

（4）、观察时间戳（Time Stamp）信息，当前 Flag 为 1，表示每台路由器都有记录它的 IP 地址和时间戳，而四对存放地址和时间戳的空间为空，目前的还没有记录 Address，时间戳都为零。

【交实验报告】

上传实验报告：ftp://172.18.187.1/　　用户名/口令：netjob/d502　　截止日期（不迟于）：1 周之内

上传包括两个文件：

（1）小组实验报告。上传文件名格式：小组号_ Ftp 协议分析实验.pdf　（由组长负责上传）

例如：文件名“10_ Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告,，视频文件与小组文件相同，扩展名是 mp4

（2）小组成员实验体会。每个同学单独交一份只填写了实验体会的实验报告。只需填写自己的学号和姓名。

文件名格式：小组号_学号_姓名_ Ftp 协议分析实验.pdf　（由组员自行上传）

例如：文件名“10_05373092_张三_ Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告。

注意：不要打包上传！