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。

院系	数据科学与计算机学院	班 级	周一班	组长	曾妮
学号	16340011	16340013		16340041	
学生	曾妮	曾翔		陈亚楠	

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。

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	Source: Elitegro_20:12:96 (00:14:2a:20:12:96)
分析	找到带有 request 信息的报文，确定为客户端，找到 MAC 地址所在层，找到 MAC 地址
2	第 1、2、3 号报文的作用是什么？
答案	建立 TCP 连接的三次握手
截图	TCP 62 1372 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TCP 62 21 → 1372 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 TCP 54 1372 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0

Wireshark · 分组 1 · ftp例1.cap

```
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)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 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
```

Wireshark · 分组 2 · ftp例1.cap

```
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)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    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·]
```

<table>
<tr><td></td><td>
Wireshark · 分组 3 · ftp例1.cap
<pre>
> Frame 3: 54 bytes on wire (432 bits), 54 bytes captured (432 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: 172.16.39.73, Dst: 172.16.28.58
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)
    [Next sequence number: 1   (relative sequence number)]
    Acknowledgment number: 1   (relative ack number)
    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····]
</pre>
</td></tr>
<tr><td>分析</td><td>一号报文由客服端发往服务端，发送顺序号 SEQ 为 0，SYN 标志位 set；

二号报文由服务端发往客户端， ACK 标志位 set，SYN 标志位 set；

三号报文由客户端发往服务端，发送顺序号 SEQ 为 1，ACK 标志位 set，三次握手建立完成。</td></tr>
<tr><td>3</td><td>该数据包中共有多少个 TCP 流？</td></tr>
<tr><td>答案</td><td>5 个</td></tr>
<tr><td>截图</td><td>
tcp.stream eq 4
<table>
<tr><th>No.</th><th>Time</th><th>Source</th><th>Destination</th><th>Protocol</th><th>Length</th><th>Info</th></tr>
<tr><td>130</td><td>149.974062</td><td>172.16.28.58</td><td>172.16.39.73</td><td>TCP</td><td>62</td><td>20 → 138</td></tr>
<tr><td>131</td><td>149.974102</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>62</td><td>1384 → 2</td></tr>
</table>
tcp.stream eq 5
<table>
<tr><th>No.</th><th>Time</th><th>Source</th><th>Destination</th><th>Protocol</th><th>Len</th></tr>
</table>
</td></tr>
<tr><td>分析</td><td>查找 TCP 流，从 0 开始一直到 4 都能查找到数据，到 5 就查找不到</td></tr>
<tr><td>4</td><td>用什么用户和密码登录成功？</td></tr>
<tr><td>答案</td><td>用户：wlx2008　　密码：wlx2008</td></tr>
<tr><td>截图</td><td>
<pre>
68 Request: USER wlx2008
90 Response: 331 User name okay, need password.
54 1372 → 21 [ACK] Seq=15 Ack=86 Win=65450 Len=0
68 Request: PASS wlx2008
84 Response: 230 User logged in, proceed.
</pre>
</td></tr>
<tr><td>分析</td><td>请求信息中附带了用户名和密码，回应信息表示用户登录成功</td></tr>
</table>

5	该 FTP 的命令连接和数据连接分别是什么样的连接？
答案	由客户端发起的“控制连接”（21），用来传输 FTP 命令，在整个会话期间一直保持打开 FTP 服务器端发起的“数据连接”（20），用来传输 FTP 数据 ；其中 21 端口与 20 端口都是在服务端
截图	Wireshark · 分组 12 · ftp例1.cap Frame 12: 78 bytes on wire (624 bits), 78 bytes captured (624 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: 172.16.39.73, Dst: 172.16.28.58 Transmission Control Protocol, Src Port: 1372, Dst Port: 21, Seq: 29, Ack: 116, Len: 24 Source Port: 1372 Destination Port: 21 [Stream index: 0] [TCP Segment Len: 24] Sequence number: 29 (relative sequence number) [Next sequence number: 53 (relative sequence number)] Acknowledgment number: 116 (relative ack number) 0101 .... = Header Length: 20 bytes (5) Wireshark · 分组 15 · ftp例1.cap Frame 15: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) Ethernet II, Src: DigitalC_02:b7:57 (00:03:0f:02:b7:57), Dst: Elitegro_20:12:96 (00:14:2a:20:12:96) Internet Protocol Version 4, Src: 172.16.28.58, Dst: 172.16.39.73 Transmission Control Protocol, Src Port: 20, Dst Port: 1377, Seq: 0, Len: 0 Source Port: 20 Destination Port: 1377 [Stream index: 1] [TCP Segment Len: 0] Sequence number: 0 (relative sequence number) [Next sequence number: 0 (relative sequence number)] Acknowledgment number: 0 0111 .... = Header Length: 28 bytes (7)
分析	客服端的 1372 端口向服务端的 21 端口发送命令，服务端的 20 端口向客户端的 1377 端口发送数据
6	该 FTP 的连接模式是那种？为什么？
答案	主动连接（PORT）
截图	12 31.305692 172.16.39.73 172.16.28.58 FTP 78 Request: PORT 172,16,39,73,5,97 13 31.306179 172.16.28.58 172.16.39.73 FTP 84 Response: 200 PORT Command successful. 14 31.308878 172.16.39.73 172.16.28.58 FTP 63 Request: NLST -1
分析	从 12 号报文可以看到，客户端主动告诉服务端 ip 地址以及端口号建立主动连接，所以连接模式为主动连接
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

Wireshark · 分组 207 · ftp例1.cap

```
Transmission Control Protocol, Src Port: 1372, Dst Port: 21, Seq: 248, Ack: 1203, Len: 0
    Source Port: 1372
    Destination Port: 21
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 248    (relative sequence number)
    [Next sequence number: 248    (relative sequence number)]
    Acknowledgment number: 1203    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x011 (FIN, 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
        .... .... ...1 = Fin: Set
        [TCP Flags: ·····A···F]
```

Wireshark · 分组 208 · ftp例1.cap

```
Transmission Control Protocol, Src Port: 21, Dst Port: 1372, Seq: 1203, Ack: 249, Len: 0
    Source Port: 21
    Destination Port: 1372
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1203    (relative sequence number)
    [Next sequence number: 1203    (relative sequence number)]
    Acknowledgment number: 249    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
    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····]
```

Wireshark · 分组 209 · ftp例1.cap

```
Transmission Control Protocol, Src Port: 21, Dst Port: 1372, Seq: 1203, Ack: 249, Len: 0
    Source Port: 21
    Destination Port: 1372
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1203    (relative sequence number)
    [Next sequence number: 1203    (relative sequence number)]
    Acknowledgment number: 249    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x011 (FIN, 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
        .... .... ...1 = Fin: Set
        [TCP Flags: ·····A···F]
```

Wireshark · 分组 210 · ftp例1.cap

```
> Frame 210: 54 bytes on wire (432 bits), 54 bytes captured (432 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: 172.16.39.73, Dst: 172.16.28.58
v Transmission Control Protocol, Src Port: 1372, Dst Port: 21, Seq: 249, Ack: 1204, Len: 0
    Source Port: 1372
    Destination Port: 21
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 249    (relative sequence number)
    [Next sequence number: 249    (relative sequence number)]
    Acknowledgment number: 1204    (relative ack number)
    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

分组 207 为客户端发往服务端通知服务器关闭连接，其中 FIN 标志位 set，ACK 标志位 set，并且发送顺序号 SEQ 为 248，确认序号 ACK 为 1203。

分组 208 为服务端发往客户端，确认收到了关闭通知报文，ACK 标志位 set，发送序号为 1203，确认序号为 249。

分组 209 为服务端发往客户端，通知客户端连接已关闭，FIN 标志位 set，ACK 标志位 set，发送序号为 1203，确认序号为 249。

分组 210 为客户端收到服务端发来的关闭报文后发送报文确认，ACK 标志位 set，发送序号为 249。

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

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.
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.
12	31.305692	172.16.39.73	172.16.28.58	FTP	78	Request: PORT 172,16,39,73,5,97
13	31.306179	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
14	31.308878	172.16.39.73	172.16.28.58	FTP	63	Request: NLST -l
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
27	42.200128	172.16.39.73	172.16.28.58	FTP	64	Request: XMKD jjj
28	42.201268	172.16.28.58	172.16.39.73	FTP	85	Response: 257 "/jjj" directory created.
30	54.715458	172.16.39.73	172.16.28.58	FTP	64	Request: RNFR jjj
31	54.716541	172.16.28.58	172.16.39.73	FTP	112	Response: 350 File or directory exists, ready for destination name
32	54.720019	172.16.39.73	172.16.28.58	FTP	64	Request: RNTO ppp
33	54.723253	172.16.28.58	172.16.39.73	FTP	84	Response: 250 RNTO command successful.
35	104.695575	172.16.39.73	172.16.28.58	FTP	79	Request: PORT 172,16,39,73,5,100
36	104.696037	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
37	104.698520	172.16.39.73	172.16.28.58	FTP	73	Request: STOR xs2009-9.xls
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
107	111.703852	172.16.39.73	172.16.28.58	FTP	79	Request: PORT 172,16,39,73,5,101
108	111.704411	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
109	111.707423	172.16.39.73	172.16.28.58	FTP	63	Request: NLST -l

109	111.707423	172.16.39.73	172.16.28.58	FTP	63	Request: NLST -l
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
122	131.649709	172.16.39.73	172.16.28.58	FTP	73	Request: RNFR xs2009-9.xls
123	131.650613	172.16.28.58	172.16.39.73	FTP	112	Response: 350 File or directory exists, ready for destination name
124	131.654130	172.16.39.73	172.16.28.58	FTP	68	Request: RNTO 888.xls
125	131.657140	172.16.28.58	172.16.39.73	FTP	84	Response: 250 RNTO command successful.
127	149.968452	172.16.39.73	172.16.28.58	FTP	79	Request: PORT 172,16,39,73,5,104
128	149.968908	172.16.28.58	172.16.39.73	FTP	84	Response: 200 PORT Command successful.
129	149.972714	172.16.39.73	172.16.28.58	FTP	68	Request: RETR 888.xls
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
205	168.024267	172.16.39.73	172.16.28.58	FTP	60	Request: QUIT
206	168.024673	172.16.28.58	172.16.39.73	FTP	68	Response: 221 Goodbye!

分析

命令分别为：

1 USER wlx2008　　发送用户名

2 PASS wlx2008　　发生密码

3 PORT 172,16,39,73,5,97　　发送 ip 地址与端口号，让服务端建立数据连接

4 NLST –l　　获取当前工作目录的信息

5 XMKD jjj　　创建 jjj 目录

6 RNFR jjj　　重命名

7 RNTO ppp　　重命名为 ppp

8 PORT 172,16,39,73,5,100　　发送 ip 地址与端口号，让服务端建立数据连接

9 STOR xs2009-9.xls　　接收数据并且在服务器站点保存为文件

10 PORT 172,16,39,73,5,101　　发送 ip 地址与端口号，让服务端建立数据连接

11 NLST –l　　获取当前工作目录的信息

12 RNFR xs2009-9.xls　　重命名

13 RNTO 888.xls　　重命名为 888.xls

14 PORT 172,16,39,73,5,104　　发送 ip 地址与端口号，让服务端建立数据连接

15 RETR 888.xls　　传输文件副本

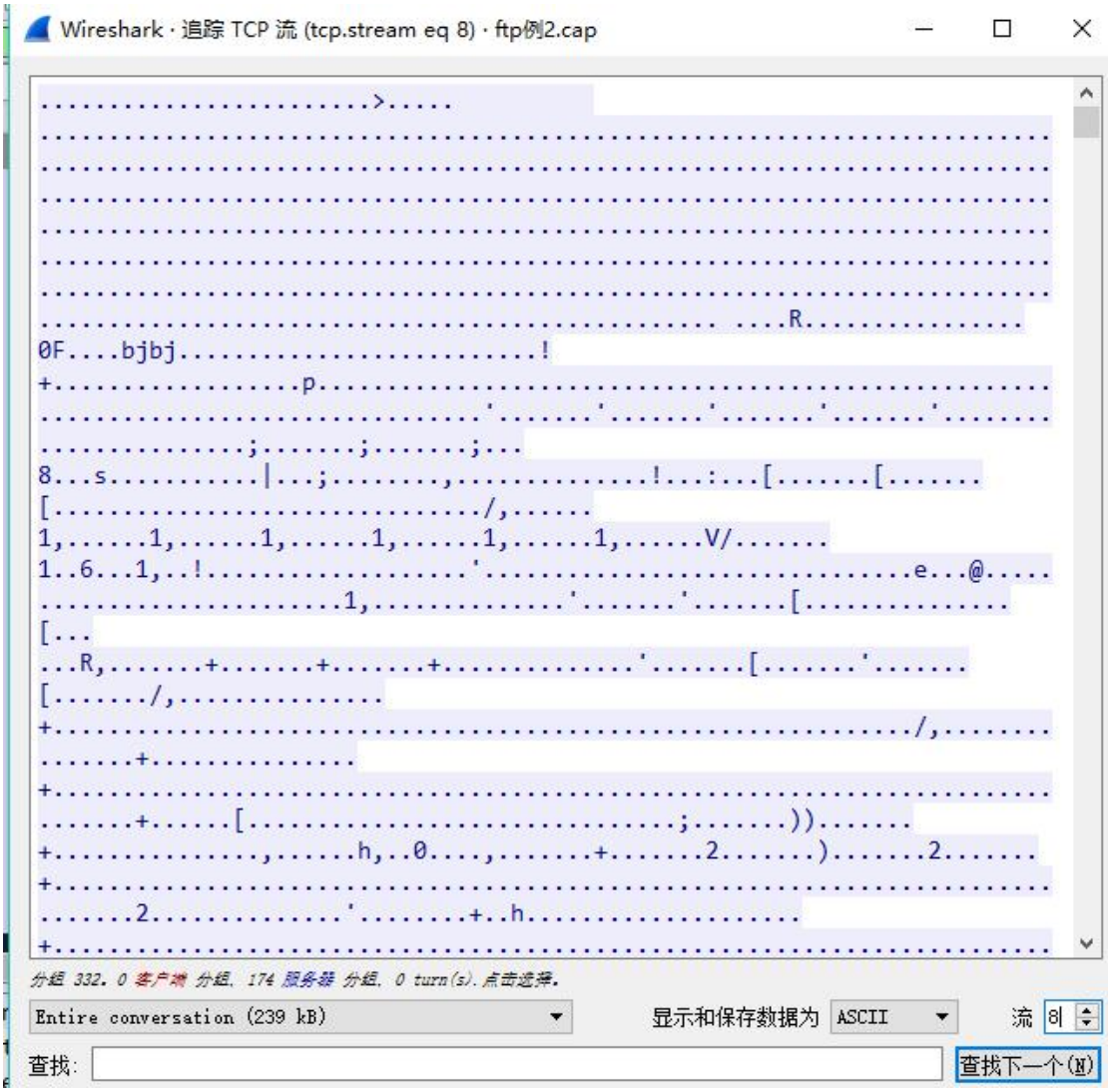
16 QUIT　　断开连接

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	客户端 mac 地址：00:14:2a：20:12:96 ftp 服务器地址：172.16.3.240
截图	Wireshark · 分组 3 · ftp例2.cap 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) 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
分析	由图中可以看出目的端口是 21，所以这是由客户端发往服务端的，所以能够知道客户端的 mac 地址与服务端的 ip 地址。
2	该数据包中共有多少个 TCP 流？
答案	9 个

截图



分析	在追踪 TCP 流窗口中，流数最大为 8，即 0 到 8，所以由 9 个流。
3	最后用什么用户和密码登录成功？
答案	用户名：kjdown 密码：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
分析	从下往上找，发现最后是以图中的用户名与密码登陆成功的
4	该 FTP 的命令连接和数据连接分别是什么？
答案	命令连接有 5 次，即与服务端 21 端口建立连接的 5 次，分别为 1454、3995、4218、4685、1123 数据连接有 4 次，被动模式下的建立数据连接的端口不固定，如下图

截图

top.seq

No.	Time	Source	Destination	Protocol	Length	Info
630	565.988017	172.16.3.240	172.16.39.93	TCP	60	21 → 1454 [ACK] Seq=1843 Ack=376 Win=65161 Len=0
631	566.203149	172.16.3.240	172.16.39.93	TCP	60	21 → 1454 [FIN, ACK] Seq=1843 Ack=376 Win=65161 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
45	54.561498	172.16.39.93	172.16.3.240	TCP	62	4218 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
46	54.571096	172.16.3.240	172.16.39.93	TCP	62	21 → 4218 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
89	177.671981	172.16.39.93	172.16.3.240	TCP	62	4685 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
90	177.672313	172.16.3.240	172.16.39.93	TCP	62	21 → 4685 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
133	267.933915	172.16.39.93	172.16.3.240	TCP	62	1132 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
134	267.935597	172.16.3.240	172.16.39.93	TCP	62	21 → 1132 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
171	346.347532	172.16.39.93	172.16.3.240	TCP	62	1454 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
172	346.347757	172.16.3.240	172.16.39.93	TCP	62	21 → 1454 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1
222	398.483654	172.16.39.93	172.16.3.240	FTP	62	Request: TYPE A
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
250	434.054849	172.16.39.93	172.16.3.240	FTP	62	Request: TYPE A
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
280	472.484679	172.16.39.93	172.16.3.240	FTP	62	Request: TYPE A
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
318	515.616639	172.16.39.93	172.16.3.240	FTP	62	Request: TYPE I
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

分析：由上图可知一共有 5 次命令连接，4 次数据连接

5：哪几个报文是 FTP 数据连接的三次握手报文？

答案：如下图

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：由上题可知 4 次数据连接，找到相应的序号，得到三次握手报文

6：哪几个报文是 FTP 数据连接的挥手报文（结束报文）？

答案：如下图

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

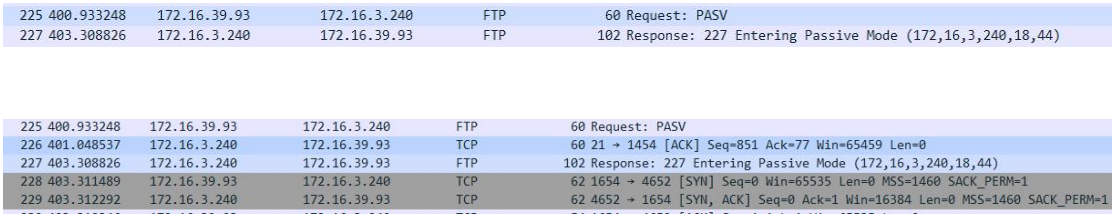
270	447.419304	172.16.3.240	172.16.39.93	TCP	60	1137 → 1791 [FIN, ACK] Seq=2992 Ack=1 Win=65535 Len=0
271	447.419373	172.16.39.93	172.16.3.240	TCP	54	1791 → 1137 [ACK] Seq=1 Ack=2993 Win=65464 Len=0
272	447.419475	172.16.39.93	172.16.3.240	TCP	54	1791 → 1137 [FIN, ACK] Seq=1 Ack=2993 Win=65464 Len=0
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
295	476.541711	172.16.39.93	172.16.3.240	TCP	54	1454 → 21 [ACK] Seq=178 Ack=1362 Win=64174 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

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：由前两题已知数据连接的三次握手报文，向下找到相对应的结束报文

7：该 FTP 的连接模式是那种？为什么？

答案	被动模式
截图	
分析	如图所示，客户端先向服务端发送“PASV”请求。然后服务端接受请求并返回报文，然后进入被动模式“Entering Passive Mode”。

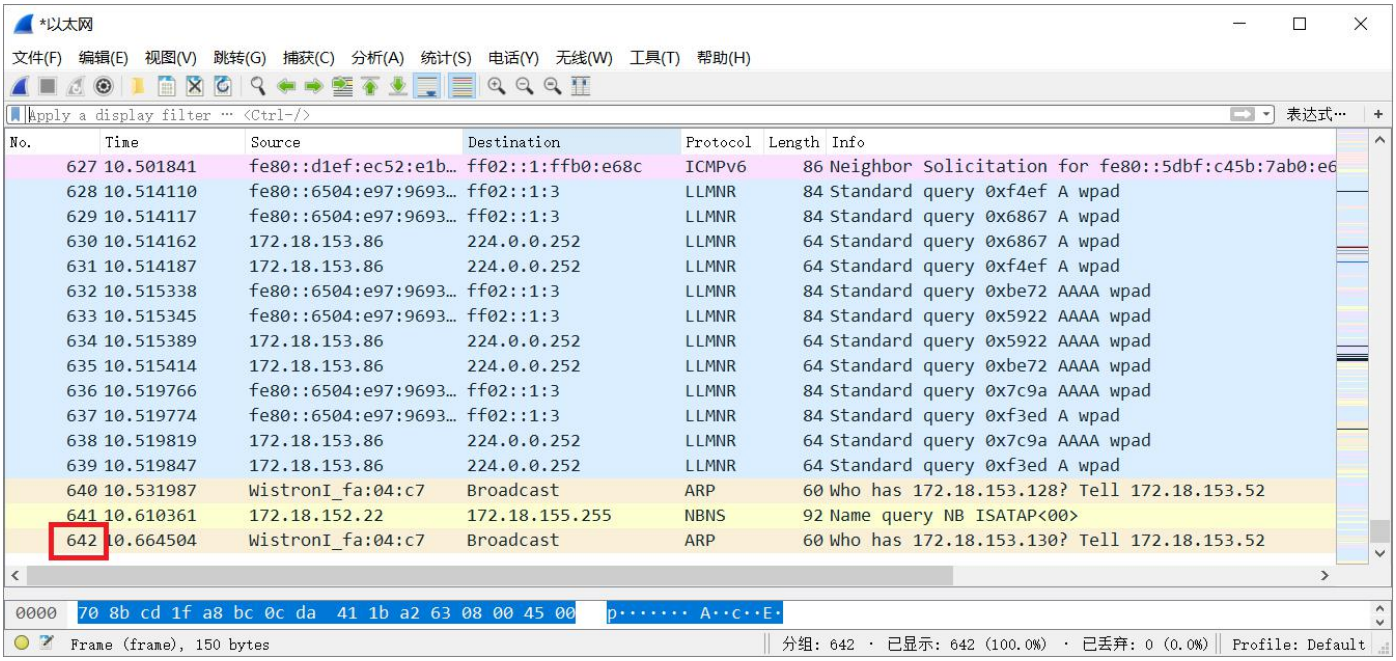
三、在线捕获数据包实验

1. 阅读教材 P64-69 内容，熟悉 FTP 协议。

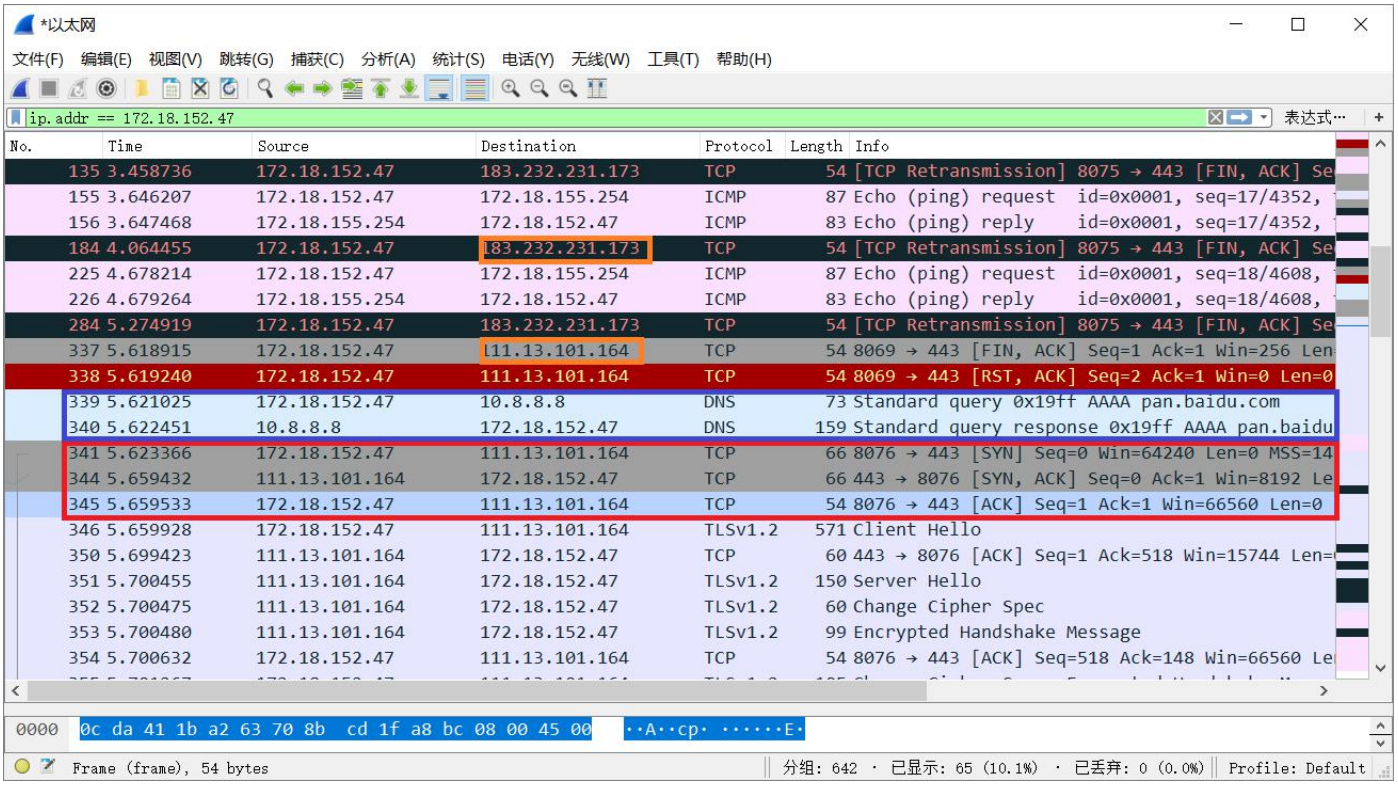
2. 完成 P51 的实例 2-1。

【实验内容】

（1）一共捕获了 642 个分组数据：



（2）既有发出去的，也有发过来的，如下图：



1. 蓝色方框标出的即为本机与 DNS 服务机的交互，本机发出请求，服务端做出响应。

2. 红色方框中表示本机与 111.13.101.164 的 TCP 连接的三次握手，也体现了这些数据既有发出去的，也有发过来的。

通过网站 www.ip138.com 查询橘色方框标出的 ip 地址的地理位置，如下图：

您查询的IP:183.232.231.173

- 本站数据：广东省广州市 移动
- 参考数据1：广东广州 移动
- 参考数据2：广东省 移动
- 兼容IPv6地址：::B7E8:E7AD
- 映射IPv6地址：::FFFF:B7E8:E7AD

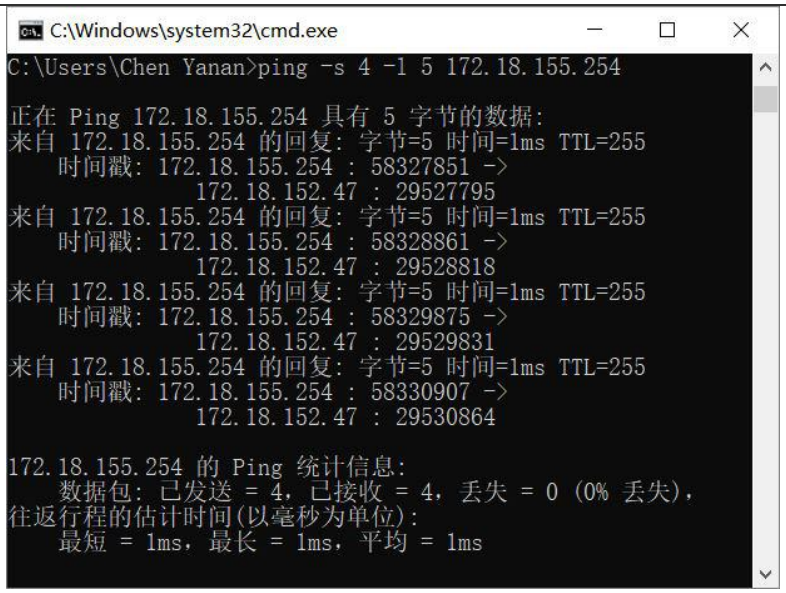
（3）网关 ip 地址可以通过命令行命令 ipconfig 查询到，如下图红框：

```
以太网适配器 以太网:

   连接特定的 DNS 后缀 . . . . . . . : sysu.edu.cn
   IPv6 地址 . . . . . . . . . . . . : 2001:250:3002:4600:dd9c:f397:b3cd:b4fa
   临时 IPv6 地址. . . . . . . . . . : 2001:250:3002:4600:9490:ed9:e27b:eaaf
   本地链接 IPv6 地址. . . . . . . . : fe80::dd9c:f397:b3cd:b4fa%5
   IPv4 地址 . . . . . . . . . . . . : 172.18.152.47
   子网掩码 . . . . . . . . . . . . : 255.255.252.0
   默认网关. . . . . . . . . . . . . : fe80::eda:41ff:fe1b:a263%5
                                       172.18.155.254
```

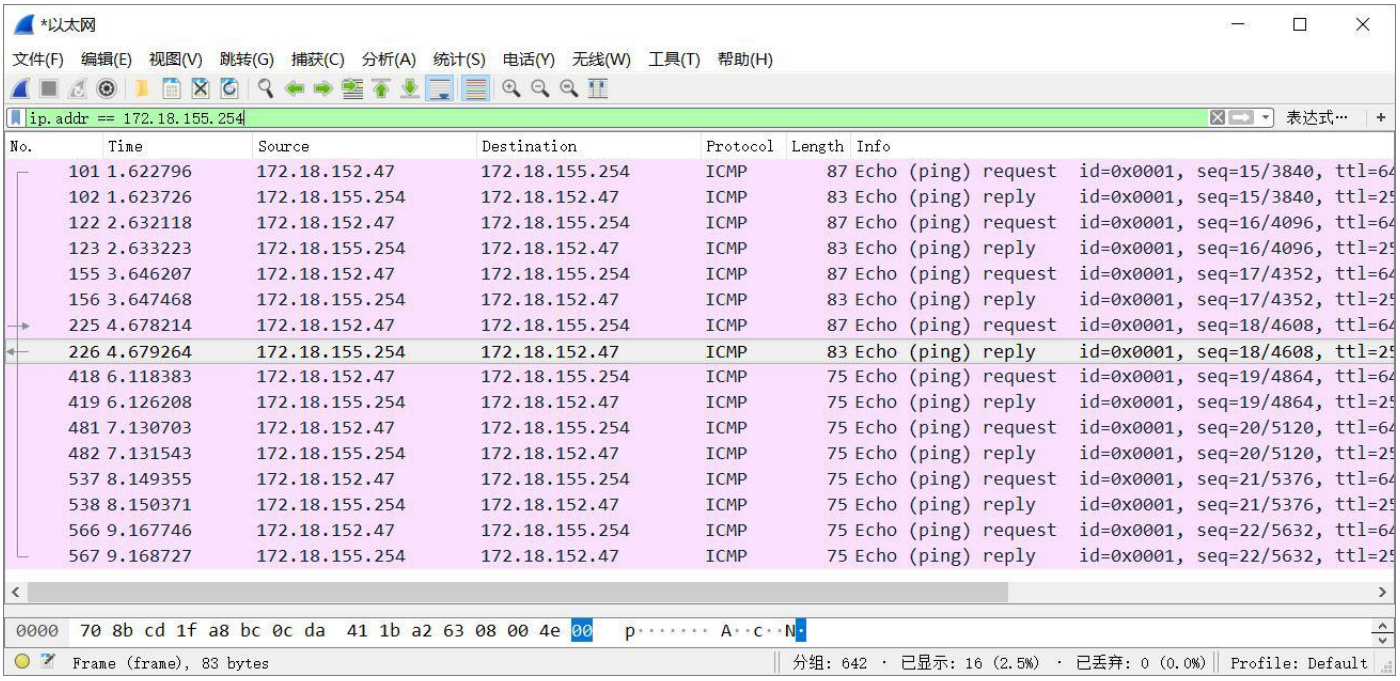
执行 ping –r 6 –l 5 172.18.155.254 命令，其中，-r 6 指的是记录计数跃点的路由，-l 5 是发送的缓冲区的大小。

```
C:\Windows\system32\cmd.exe

C:\Users\Chen Yanan>ping -r 6 -l 5 172.18.155.254

正在 Ping 172.18.155.254 具有 5 字节的数据:
来自 172.18.155.254 的回复: 字节=5 时间=7ms TTL=255
    路由: 172.18.155.254
来自 172.18.155.254 的回复: 字节=5 时间<1ms TTL=255
    路由: 172.18.155.254
来自 172.18.155.254 的回复: 字节=5 时间=1ms TTL=255
    路由: 172.18.155.254
来自 172.18.155.254 的回复: 字节=5 时间=1ms TTL=255
    路由: 172.18.155.254

172.18.155.254 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 = 0ms，最长 = 7ms，平均 = 2ms

C:\Users\Chen Yanan>
```

执行 ping –s 4 –l 5 172.18.155.254 命令，其中-s 4 指的是计数跃点的时间戳。



（4）查看捕获的数据包中 ip 地址为 172.18.155.254（网关）的部分：



（5）捕获的只有 ICMP 协议，是 TCP/IP 协议族的一个子协议，用于在 IP 主机、路由器之间传递控制消息。这里捕获到的是上一题中我们 ping 网关时的数据，故而只有 Echo 的请求和相应，由于只是主机随机发送出去的 5 个字节的 ping 操作，这些字段并没有什么特别的含义。

【实验思考】

（1） 对网络嗅探行为的检测主要是检测网络接口设备是否工作在混杂模式,一般有以下几种方式：

a) 采用 ARP 技术检测网络嗅探行为

b) 采用 DNS 技术检测网络嗅探行为

c) 采用网络和主机响应时间测试的方法检测网络嗅探行为

学号	学生	自评分

（2） 嗅探防范措施主要有：

a) 采用主动式集线器或交换机

b) 加密传输

c) 一次性口令

d) 使用不支持混杂工作模式的网卡

本次实验完成后，请根据组员在实验中的贡献，请实事求是，自评在实验中应得的分数。（按百分制）

【交实验报告】

上传实验报告：ftp://222.200.180.109/ 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

（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！