



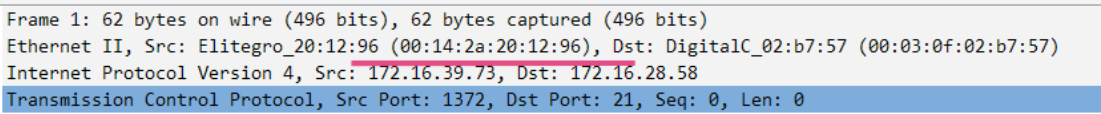
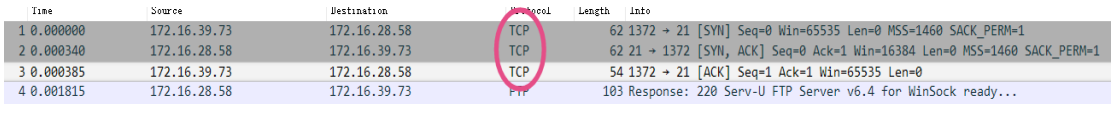
警示

- 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。

院系	数据科学与计算机学院	班 级	电子政务	组长	刘硕
学号	16340154	16340148	16340171		
学生	刘硕	刘虹奇	聂博业		

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	
分析	截图是 ftp 例 1.cap 的第一号报文，协议用的是 TCP，含义是使用 FTP 的客户端 172.16.39.72 即将与服务器 172.16.28.25 建立 TCP 连接，完成了第一次握手，所以 FTP 客户端的 mac 地址即为来源 Src 所显示的值。
2	第 1、2、3 号报文的作用是什么？
答案	第 1、2、3 号报文实现了 FTP 客户端与服务器的三次握手，并且成功建立 TCP 连接，准备传输数据。
截图	
分析	前三号报文分别代表了三次握手。第一号报文是 FTP 客户端发送 SYN 报文，置发送序号为 $x=0$ ；之后服务器发送 SYN+ACK 报文，置发送序号为 $y=0$ ，确认序号为 $x+1=1$ ；最后客户端发送 ACK 报文，置发送序号为 1，即 $x+1$ ，确认序号为 $y+1=1$ 。此时完成了三次握手过程，即成功建立了一条 TCP 连接，下一号报文即可将数据从服务器交给 FTP 客户端。
3	该数据包中共有多少个 TCP 流？
答案	5 个 TCP 流。



```
220 Serv-U FTP Server v6.4 for WinSock ready...
USER wlx2008
331 User name okay, need password.
PASS wlx2008
230 User logged in, proceed.
PORT 172,16,39,73,5,97
200 PORT Command successful.
NLST -l
150 Opening ASCII mode data connection for /bin/ls.
226-Maximum disk quota limited to 307200 kbytes
Used disk quota 0 kbytes, available 307143 kbytes
226 Transfer complete.
XMKD j]]
257 "/j]]" directory created.
RNFR j]]
350 File or directory exists, ready for destination name
RNT0 ppp
250 RNT0 command successful.
PORT 172,16,39,73,5,104
200 PORT Command successful.
RETR 888.xls
150 Opening ASCII mode data connection for 888.xls (57856 Bytes).
226-Maximum disk quota limited to 307200 kbytes
Used disk quota 56 kbytes, available 307143 kbytes
226 Transfer complete.
QUIT
221 Goodbye!
```

drw-rw-rw-	1 user	group	0 Nov 25 15:16 123456
drw-rw-rw-	1 user	group	0 Nov 25 15:19 9999

```
.....>.....
.....o.....n.....
.....
.....\p....sise
B.....a.....=.....:f!8.....X.@
1.....[S01.....[S01.....[
[S01.....%1.....[S01.....
%1.....[S01.....[S01.....
0;[.R.e.d.].....\..#.,#.0...7.....".#.,#.0...0.0.;".
0...0.0.;[.R.e.d.].....\..#.,#.0...0.0...i*.2...".*.*.
".*.*."-"._.;_..@_..(....).)_ * #,##0_ ;_ * \-#,#0_ ;_
0...0.0_ ;_..".*.*.\..#.,#.0...0.0_ ;_..".*.*.*.
\..#,#0.00_ ;_ * "-"?? ;_ @ .....\$#,#00_);\(\$#,#00\)...
\$#,#0.00\)...\$#,#0.00_);[Red]\(
\$#,#0.00\)... "Yes"; "Yes"; "No"..... "True"; "True"; "False"...
.#.,#.0...0.0_);[.R.e.d.].....\..[.$. -2.]..#.,#.0...0.0\
.....
.....
.....X.@ @ ..X.@ @ ..8.@ @
.....
..X..@ @ ..
```

drw-rw-rw-	1 user	group	0 Nov 25 15:16 123456
drw-rw-rw-	1 user	group	0 Nov 25 15:19 9999
drw-rw-rw-	1 user	group	0 Nov 25 15:20 bbb
drw-rw-rw-	1 user	group	0 Nov 25 15:20 ppp
-rw-rw-rw-	1 user	group	57856 Nov 25 15:21 xs2009-9.xls

```
.....>.....
.....o.....n.....
.....
.....\p....sise
B.....a.....=.....:f!8.....X.
1.....[S01.....[S01.....[
[S01.....%1.....[S01.....
%1.....[S01.....[S01.....
0;[.R.e.d.].....\..#.,#.0...7.....".#.,#.0...0.0.;".
0...0.0.;[.R.e.d.].....\..#.,#.0...0.0...i*.2...".*.*.
".*.*."-"._.;_..@_..(....).)_ * #,##0_ ;_ * \-#,#0_ ;_
0...0.0_ ;_..".*.*.\..#.,#.0...0.0_ ;_..".*.*.*.
\..#,#0.00_ ;_ * "-"?? ;_ @ .....\$#,#00_);\(\$#,#00\)...
\$#,#0.00\)...\$#,#0.00_);[Red]\(
\$#,#0.00\)... "Yes"; "Yes"; "No"..... "True"; "True"; "False"...
.#.,#.0...0.0_);[.R.e.d.].....\..[.$. -2.]..#.,#.0...0.0\
.....
.....
.....X.@ @ ..X.@ @ ..8.@ @
.....
..X..@ @ ..
```

截图

分析

在抓包取到的数据包中选择需要进行查看的一条的数据包的内容,使用右键的方式来打开数据包中的数据流。右键一条数据包选择跟踪 TCP 流的选项,这是便可以通过打开的页面,来具体分



计算机网络实验报告

	析数据流的内容。整个 TCP 流就会在一个单独的窗口中显示出来，我们注意到这个窗口中的文件以两种颜色显示，其中红色用来标明从源地址前往目的地址的流量，而蓝色用来区分出相反方向也就是从目的地址到源地址的流量。这里颜色的标记以哪方先开始通信为准，一般情况下都是由客户端主动发起与服务器的连接，所以大都是将客户端的通信显示为红色。																																																																											
4	用什么用户和密码登录成功？																																																																											
答案	用户：wlx2008；密码：wlx2008。																																																																											
截图	<table><tr><td>6 17.542571</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>68 Request: USER wlx2008</td></tr><tr><td>7 17.543205</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>90 Response: 331 User name okay, need password.</td></tr><tr><td>8 17.670704</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>54 1372 → 21 [ACK] Seq=15 Ack=86 Win=65450 Len=0</td></tr><tr><td>9 21.617636</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>68 Request: PASS wlx2008</td></tr><tr><td>10 21.618699</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 230 User logged in, proceed.</td></tr></table>	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.																																																		
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	当 TCP 连接成功后，服务器会向 FTP 主机回应请求用户名和密码，主机提供用户名请求 USER wlx2008，服务器返回正确；之后主机提供密码请求 PASS wlx2008，服务器返回正确，之后便可以正常与 FTP 服务器交互数据。																																																																											
5	该 FTP 的命令连接和数据连接分别是什么样的连接？																																																																											
答案	FTP 的控制链接使用端口号 21，是由客户端发起的；FTP 的数据连接是主动模式，使用 20 号端口，用于文件数据和目录数据的传输。																																																																											
截图	<table><tr><td>1 0.000000</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>62 1372 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_</td></tr><tr><td>2 0.000340</td><td>172.16.28.58</td><td>172.16.39.73</td><td>TCP</td><td>62 21 → 1372 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=</td></tr><tr><td>3 0.000385</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>54 1372 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0</td></tr><tr><td>4 0.001815</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>103 Response: 220 Serv-U FTP Server v6.4 for WinSock res</td></tr><tr><td>5 0.201287</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>54 1372 → 21 [ACK] Seq=1 Ack=50 Win=65486 Len=0</td></tr><tr><td>6 17.542571</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>68 Request: USER wlx2008</td></tr><tr><td>7 17.543205</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>90 Response: 331 User name okay, need password.</td></tr><tr><td>8 17.670704</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>54 1372 → 21 [ACK] Seq=15 Ack=86 Win=65450 Len=0</td></tr><tr><td>9 21.617636</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>68 Request: PASS wlx2008</td></tr><tr><td>10 21.618699</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 230 User logged in, proceed.</td></tr></table> <table><tr><td>13 31.306179</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 200 PORT Command successful.</td></tr><tr><td>14 31.308878</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>63 Request: NLST -l</td></tr><tr><td>15 31.309831</td><td>172.16.28.58</td><td>172.16.39.73</td><td>TCP</td><td>62 20 → 1377 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1</td></tr><tr><td>16 31.309871</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>62 1377 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_</td></tr><tr><td>17 31.310370</td><td>172.16.28.58</td><td>172.16.39.73</td><td>TCP</td><td>60 20 → 1377 [ACK] Seq=1 Ack=1 Win=65535 Len=0</td></tr></table>	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_	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=	3 0.000385	172.16.39.73	172.16.28.58	TCP	54 1372 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0	4 0.001815	172.16.28.58	172.16.39.73	FTP	103 Response: 220 Serv-U FTP Server v6.4 for WinSock res	5 0.201287	172.16.39.73	172.16.28.58	TCP	54 1372 → 21 [ACK] Seq=1 Ack=50 Win=65486 Len=0	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.	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	15 31.309831	172.16.28.58	172.16.39.73	TCP	62 20 → 1377 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1	16 31.309871	172.16.39.73	172.16.28.58	TCP	62 1377 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_	17 31.310370	172.16.28.58	172.16.39.73	TCP	60 20 → 1377 [ACK] Seq=1 Ack=1 Win=65535 Len=0
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_																																																																								
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=																																																																								
3 0.000385	172.16.39.73	172.16.28.58	TCP	54 1372 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0																																																																								
4 0.001815	172.16.28.58	172.16.39.73	FTP	103 Response: 220 Serv-U FTP Server v6.4 for WinSock res																																																																								
5 0.201287	172.16.39.73	172.16.28.58	TCP	54 1372 → 21 [ACK] Seq=1 Ack=50 Win=65486 Len=0																																																																								
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.																																																																								
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																																																																								
15 31.309831	172.16.28.58	172.16.39.73	TCP	62 20 → 1377 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1																																																																								
16 31.309871	172.16.39.73	172.16.28.58	TCP	62 1377 → 20 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_																																																																								
17 31.310370	172.16.28.58	172.16.39.73	TCP	60 20 → 1377 [ACK] Seq=1 Ack=1 Win=65535 Len=0																																																																								
分析	FTP 的控制链接有 FTP 控制命令完成工作，FTP 控制命令由 FTP 协议规定，以 ASCII 码方式传送。例如发送用户名的命令 USER，发送密码的命令是 PASS；FTP 的数据连接使用的主动模式即 PORT 方式，收到数据传送请求后，服务器主动与客户端建立连接。为此服务器必须获得客户端的端口号。传输数据时，服务器端通过自己的 TCP20 端口连接至客户端的指定端口发送数据。																																																																											
6	该 FTP 的连接模式是那种？为什么？																																																																											
答案	FTP 的连接模式是主动模式（PORT）。																																																																											
截图	<table><tr><td>12 31.305692</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>78 Request: PORT 172,16,39,73,5,97</td></tr><tr><td>13 31.306179</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 200 PORT Command successful.</td></tr></table> <table><tr><td>35 104.695575</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>79 Request: PORT 172,16,39,73,5,100</td></tr><tr><td>36 104.696037</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 200 PORT Command successful.</td></tr></table> <table><tr><td>106 105.017679</td><td>172.16.39.73</td><td>172.16.28.58</td><td>TCP</td><td>54 1372 → 21 [ACK] Seq=136 Ack=663 Win=64872</td></tr><tr><td>107 111.703852</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>79 Request: PORT 172,16,39,73,5,101</td></tr><tr><td>108 111.704411</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 200 PORT Command successful.</td></tr></table> <table><tr><td>127 149.968452</td><td>172.16.39.73</td><td>172.16.28.58</td><td>FTP</td><td>79 Request: PORT 172,16,39,73,5,104</td></tr><tr><td>128 149.968908</td><td>172.16.28.58</td><td>172.16.39.73</td><td>FTP</td><td>84 Response: 200 PORT Command successful.</td></tr></table>	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.	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.	106 105.017679	172.16.39.73	172.16.28.58	TCP	54 1372 → 21 [ACK] Seq=136 Ack=663 Win=64872	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.	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.																														
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.																																																																								
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.																																																																								
106 105.017679	172.16.39.73	172.16.28.58	TCP	54 1372 → 21 [ACK] Seq=136 Ack=663 Win=64872																																																																								
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.																																																																								
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.																																																																								
分析	FTP 的两个模式是主动模式（PORT）和被动模式（PASV）。主动模式是指服务器主动链接客户端的数据端口，连接成功后服务器便可以向 FTP 主机（从 20 端口）发送数据，并且在之后的每个传送到数据包中也有标识的 POST 字样。其中 POST 模式的过程：FTP 客户端从任意的																																																																											



计算机网络实验报告

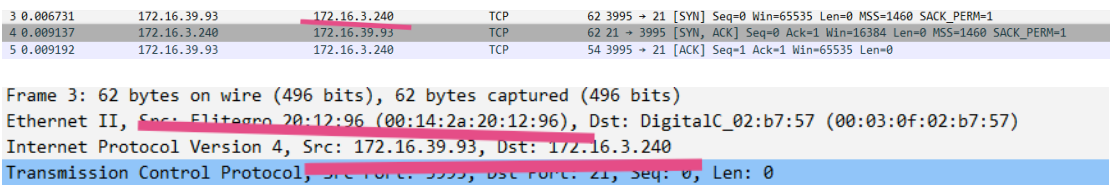
	非特殊的端口（ $N > 1204$ ）连入到 FTP 服务器的命令端口——21 端口，然后客户端在 $N+1$ 端口监听，并且通过该 $N+1$ 端口发送 PORT 命令给 FTP 服务器，接着服务器会从它自己的数据端口（20）连接到客户端指定的数据端口（ $N+1$ ）；PASV 模式的过程：当开启一个 FTP 连接时，客户端随机打开一个大于 1204 的本地端口 N 向服务器的 21 号端口发起连接，同时会开启 $N+1$ 号端口。然后向服务器提交 PASV 命令，通知服务器自己处于被动模式。那么服务器收到命令后就会开启一个任意的非特权端口（ $P > 1204$ ）监听，并发送 PASV P 命令给客户端通知自己的数据端口是 P 。然后客户端通过本地端口 $N+1$ 连接到服务器的端口 P 的连接用来传送数据。
7	最后四个报文的作用是什么？
答案	断开 TCP 连接。
截图	
分析	首先，FTP 客户端发送一个 FIN，用来关闭客户端到服务器的数据传送；之后服务器收到了这个 FIN，并且返回了一个 ACK，确认序号为 249（是上一号报文的收到序号 248+1），同 SYN 一样，一个 FIN 也将占用一个序号；倒数第二号报文是服务器发送一个 FIN 给 FTP 客户端，关闭与客户端之间的 TCP 连接；最后是 FTP 客户端返回 ACK 报文确认关闭，并且和倒数第三号报文一样，确认序号为 1204（是上一号报文的收到序号 1203+1）。
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 FTP 命令以及 21 个 FTP 应答，含义在分析中给出。
截图	



计算机网络实验报告

	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: RINTO 888.xls 125 131.657140 172.16.28.58 172.16.39.73 FTP 84 Response: 250 RINTO 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	第 4 号报文回应主机连接已经成功建立,可以进行数据传输;第 6 号命令输入用户名 wlx2008;第 7 号报文回应用户名正确,需要密码;第 9 号报文命令输入密码 wlx2008;第 10 号报文回应密码正确;第 12 号报文服务器主动链接客户端的数据端口;第 13 号报文回应连接成功;第 14 号报文命令服务器列出目录名称;第 18 号报文回应服务器在传输数据时不能连接到用户;第 25 号报文回应最大磁盘配额限制为 307200KB;第 27 号报文命令传建一个目录;第 28 号报文回应 FTP 目录被创建,为 jjj;第 30 号报文命令重命名目录 jjj;第 31 号报文回应准备好对其进行重命名;第 32 号报文命令将目录 jjj 重命名为 ppp;第 33 号报文回应重命名成功;第 35 号报文服务器主动链接客户端的数据端口;第 36 号报文回应连接成功;第 37 号报文命令接收数据并且在服务器站点保存文件 xs2009-9.xls;第 41 号报文回应服务器不能连接到文件 xs2009-9.xls;第 105 号报文回应最大磁盘配额限制为 307200KB;第 107 号报文主动链接客户端的数据端口;第 108 号报文回应连接成功;第 109 号报文返回指定目录的文件名列表;第 113 号报文回应服务器在传输数据时不能连接到用户;第 120 号报文回应最大磁盘配额限制为 307200KB;第 122 号报文命令重命名文件 xs2009-9.xls;第 123 号报文回应准备好对其进行重命名;第 124 号报文命令将文件 xs2009-9.xls 重命名为 888.xls;第 125 号报文回应重命名成功;第 127 号报文主动链接客户端的数据端口;第 128 号报文回应连接成功;第 129 号报文传输副本文件 888.xls;第 133 号报文回应服务器不能连接到文件 888.xls;第 203 号报文回应最大磁盘配额限制为 307200KB;第 205 号报文命令 FTP 客户端与服务器断开连接;第 206 号报文回应服务器就这样和您再见了嚟。

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	FTP 服务器的 ip 是 172.16.3.240; FTP 客户端的 mac 地址是 00:14:2a:20:12:96。
截图	
分析	第三号报文是 FTP 客户端请求与服务器建立 TCP 连接的第一次握手的过程,截图中报文来自 FTP 客户端(172.16.39.93),要发往终点服务器(172.16.3.240);同样可知,FTP 客户端的 mac 地址即为来源 Src 的值。
2	该数据包中共有多少个 TCP 流?
答案	9 个 TCP 流。



截图

```
220-FTP Server ready...
220-..FTP.....
220-.....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220
USER anonymous
331 User name okay, please send complete E-mail address as password.
PASS IEUser@
530 .....
```

```
220-FTP Server ready...
220-..FTP.....
220-.....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220
USER anonymous
331 User name okay, please send complete E-mail address as password.
PASS IEUser@
530 .....
```

```
220-FTP Server ready...
220-..FTP.....
220-.....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220
USER xxxx
331 User name okay, need password.
PASS yyyy
530 Not logged in.
```

```
220-FTP Server ready...
220-..FTP.....
220-.....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220
USER kjdown
```




```
220-FTP Server ready...
220-..FTP.....
220-.....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-.....
220-(..).....
220-(..).....
220-(..).....
220-.....
USER kjdown
331 User name okay, need password.
PASS kjdown
230 User logged in, proceed.
opts utf8 on
501 Invalid option.
syst
215 UNIX Type: L8
site help
501 SITE option not supported.
PWD
257 "/" is current directory.
TYPE A
200 Type set to A.
PASV
227 Entering Passive Mode (172,16,3,240,18,44)
LIST
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.
noop
200 Command okay.
```

drw-rw-rw-	1	user	group	0 Sep 9 13:08 .
drw-rw-rw-	1	user	group	0 Sep 9 13:08 ..
drw-rw-rw-	1	user	group	0 Oct 28 09:32 2008.....
drw-rw-rw-	1	user	group	0 Jul 11 19:20 java.....
drw-rw-rw-	1	user	group	0 Oct 13 13:29
drw-rw-rw-	1	user	group	0 Dec 15 2007
drw-rw-rw-	1	user	group	0 Sep 16 16:33
drw-rw-rw-	1	user	group	0 Oct 20 09:36
drw-rw-rw-	1	user	group	0 Feb 26 2009
drw-rw-rw-	1	user	group	0 Sep 30 14:53
drw-rw-rw-	1	user	group	0 Mar 10 2009
drw-rw-rw-	1	user	group	0 Sep 28 14:33
drw-rw-rw-	1	user	group	0 Sep 2 2008
drw-rw-rw-	1	user	group	0 Oct 28 09:33
drw-rw-rw-	1	user	group	0 Sep 7 17:01
drw-rw-rw-	1	user	group	0 Oct 28 16:04
drw-rw-rw-	1	user	group	0 Oct 20 11:09
drw-rw-rw-	1	user	group	0 Oct 21 09:42
drw-rw-rw-	1	user	group	0 Sep 25 11:13
drw-rw-rw-	1	user	group	0 Sep 2 2008
drw-rw-rw-	1	user	group	0 Jan 14 2009
drw-rw-rw-	1	user	group	0 Oct 28 15:07
drw-rw-rw-	1	user	group	0 Dec 15 2007



	
分析	在软件中点击菜单中的 Analyze 选项，这样就会弹出了下拉菜单，选择为跟踪 TCP 流的选项。红色带表 FTP 客户端发送数据流，蓝色代表服务器端返回数据流， Wireshark 下面有提示， TCP 模拟封包发送的数据包，直接发送红色数据，返回服务端字节集。
3	最后用什么用户和密码登录成功？
答案	最后登陆成功的用户名为 kjdown ；密码为 kjdown 。
截图	
分析	当三次握手完成之后， FTP 客户端便成功与服务器建立了 TCP 连接， FTP 服务器需要客户端发送用户名和密码来验证才可以传输数据。之后用户命令了几次用户名和密码，但服务器都会迎来错误，最终到了第二百零五号报文的命令是告知服务器用户名为 kjdown ，第二百零七号报文的命令是告知服务器密码为 kjdown ，服务器回应登录成功。
4	该 FTP 的命令连接和数据连接分别是什么？
答案	FTP 的控制链接由客户端发起，使用端口号 21 ； FTP 的数据连接是被动模式，使用端口号 20 ，用于文件数据和目录数据的传输。
截图	



	<table><tr><td>225</td><td>400.933248</td><td>172.16.39.93</td><td>172.16.3.240</td><td>FTP</td><td>60 Request: PASV</td></tr><tr><td>226</td><td>401.048537</td><td>172.16.3.240</td><td>172.16.39.93</td><td>TCP</td><td>60 21 → 1454 [ACK] Seq=851 Ack=77 Win=65459 Len=0</td></tr><tr><td>227</td><td>403.308826</td><td>172.16.3.240</td><td>172.16.39.93</td><td>FTP</td><td>102 Response: 227 Entering Passive Mode (172,16,3,240,18,44)</td></tr></table>	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)						
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)																				
分析	FTP 的控制链接有 FTP 控制命令完成工作，FTP 控制命令由 FTP 协议规定，以 ASCII 码方式传送。例如发送用户名的命令 USER，发送密码的命令是 PASS；FTP 的数据连接使用的被动模式即 PASV 方式，FTP 客户端发送 PASV 命令到 FTP 服务器。FTP 服务器收到 PASV 命令后，随机打开一个高端端口（端口号大于 1024），并且通知该客户端在该端口上传数据的请求。客户端连接 FTP 服务器端口，然后 FTP 服务器将很多通过该端口数据的传送，此时 FTP 服务器不再需要和客户端建立一个新连接。																								
5	哪几个报文是 FTP 数据连接的三次握手机文？																								
答案	第 3、4、5 号报文是 FTP 数据连接的三次握手机文。																								
截图	<table><tr><td>3</td><td>0.006731</td><td>172.16.39.93</td><td>172.16.3.240</td><td>TCP</td><td>62 3995 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1</td></tr><tr><td>4</td><td>0.009137</td><td>172.16.3.240</td><td>172.16.39.93</td><td>TCP</td><td>62 21 → 3995 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1</td></tr><tr><td>5</td><td>0.009192</td><td>172.16.39.93</td><td>172.16.3.240</td><td>TCP</td><td>54 3995 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0</td></tr></table> <p>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</p> <p>Frame 4: 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: 172.16.3.240, Dst: 172.16.39.93 Transmission Control Protocol, Src Port: 21, Dst Port: 3995, Seq: 0, Ack: 1, Len: 0</p> <p>Frame 5: 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: 172.16.39.93, Dst: 172.16.3.240 Transmission Control Protocol, Src Port: 3995, Dst Port: 21, Seq: 1, Ack: 1, Len: 0</p>	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	5	0.009192	172.16.39.93	172.16.3.240	TCP	54 3995 → 21 [ACK] Seq=1 Ack=1 Win=65535 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																				
5	0.009192	172.16.39.93	172.16.3.240	TCP	54 3995 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0																				
分析	客户端向服务器发送一个连接请求包，标志位 SYN 置为 1，序号置为 0。服务器收到客户端发来的报文，有 SYN=1 知道客户端是需要建立连接，于是向客户端发送一个确认包，置 SYN=1，ACK=1，第一次握手结束。之后设置初始序号 Y=0，将确认序号 ACK 设置为客户的序列号加 1，即 X+1=1。客户端收到服务器发来的包后检查确认序号 ACK 是否正确，即第一次发送的序号加 1，即 X+1=1。以及标志位 ACK 是否为 1。若正确，客户端再次发送确认包，ACK 标志位为 1，SYN 标志位为 0。确认序号 ACK=Y+1=1，发送序号为 X+1=1。服务器收到后确认序号值 ACK=1，则完成三次握手，连接建立成功。																								
6	哪几个报文是 FTP 数据连接的挥手报文（结束报文）？																								
答案	最后的四个报文（629 号到 632 号）是 FTP 数据连接的挥手报文（结束报文）。																								
截图	<table><tr><td>629</td><td>565.983884</td><td>172.16.39.93</td><td>172.16.3.240</td><td>TCP</td><td>54 1454 → 21 [FIN, ACK] Seq=375 Ack=1843 Win=65161 Len=0</td></tr><tr><td>630</td><td>565.988017</td><td>172.16.3.240</td><td>172.16.39.93</td><td>TCP</td><td>60 21 → 1454 [ACK] Seq=1843 Ack=376 Win=65161 Len=0</td></tr><tr><td>631</td><td>566.203149</td><td>172.16.3.240</td><td>172.16.39.93</td><td>TCP</td><td>60 21 → 1454 [FIN, ACK] Seq=1843 Ack=376 Win=65161 Len=0</td></tr><tr><td>632</td><td>566.203215</td><td>172.16.39.93</td><td>172.16.3.240</td><td>TCP</td><td>54 1454 → 21 [ACK] Seq=376 Ack=1844 Win=65161 Len=0</td></tr></table> <p>Frame 629: 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: 172.16.39.93, Dst: 172.16.3.240 Transmission Control Protocol, Src Port: 1454, Dst Port: 21, Seq: 375, Ack: 1843, Len: 0</p> <p>Frame 630: 60 bytes on wire (480 bits), 60 bytes captured (480 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: 172.16.3.240, Dst: 172.16.39.93 Transmission Control Protocol, Src Port: 21, Dst Port: 1454, Seq: 1843, Ack: 376, Len: 0</p>	629	565.983884	172.16.39.93	172.16.3.240	TCP	54 1454 → 21 [FIN, ACK] Seq=375 Ack=1843 Win=65161 Len=0	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	632	566.203215	172.16.39.93	172.16.3.240	TCP	54 1454 → 21 [ACK] Seq=376 Ack=1844 Win=65161 Len=0
629	565.983884	172.16.39.93	172.16.3.240	TCP	54 1454 → 21 [FIN, ACK] Seq=375 Ack=1843 Win=65161 Len=0																				
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																				
632	566.203215	172.16.39.93	172.16.3.240	TCP	54 1454 → 21 [ACK] Seq=376 Ack=1844 Win=65161 Len=0																				



	<div>Frame 631: 60 bytes on wire (480 bits), 60 bytes captured (480 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: 172.16.3.240, Dst: 172.16.39.93 Transmission Control Protocol, Src Port: 21, Dst Port: 1454, Seq: 1843, Ack: 376, Len: 0</div> <div>Frame 632: 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: 172.16.39.93, Dst: 172.16.3.240 Transmission Control Protocol, Src Port: 1454, Dst Port: 21, Seq: 376, Ack: 1844, Len: 0</div>
分析	客户端发出连接释放报文，并且停止发送数据。释放数据报文首部，FIN=1，其序列号为SEQ=375，此时，客户端进入终止等待 1 状态。之后服务器收到连接释放报文，发出确认报文，ACK=375+1=376，并且带上自己的序列号 SEQ=1843，此时，服务端就进入了关闭等待状态。客户端收到服务器的确认请求后，此时，客户端就进入终止等待 2 状态，等待服务器发送连接释放报文。服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，ACK=375+1=376，此时，服务器就进入了最后确认状态，等待客户端的确认。客户端收到服务器的连接释放报文后，必须发出确认，此时，客户端就进入了时间等待状态。服务器只要收到了客户端发出的确认，立即进入关闭状态。此时就完成了四次挥手，就结束了这次的 TCP 连接。
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，均是由客户端从 N>1204 的接口连入到 FTP 服务器的 21 号命令端口，然后客户端收到服务器的数据连接请求后发送 PASV 命令到服务器，后者会从它自己的一个任意的非特权端口监听，等待客户端通过本地端口 N+1 连接到服务器的端口的连接用来传送数据。
截图	<div>224 400.851141 172.16.3.240 172.16.39.93 FTP 74 Response: 200 Type set to A. 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) 228 403.311400 172.16.3.240 172.16.3.240 TCP 60 21 → 1454 [ACK] Seq=1053 Ack=115 Win=65421 Len=0 251 434.242603 172.16.3.240 172.16.39.93 TCP 60 21 → 1454 [ACK] Seq=1053 Ack=121 Win=65415 Len=0 252 436.768635 172.16.3.240 172.16.39.93 FTP 74 Response: 200 Type set to A. 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)</div>
分析	PASV 模式的过程中，当开启一个 FTP 连接时，客户端随机打开一个大于 1204 的本地端口 N 向服务器的 21 号端口发起连接，同时会开启 N+1 号端口。然后向服务器提交 PASV 命令，通知服务器自己处于被动模式。那么服务器收到命令后就会开启一个任意的非特权端口(P>1204)监听，并发送 PASV P 命令给客户端通知自己的数据端口是 P。然后客户端通过本地端口 N+1 连接到服务器的端口 P 的连接用来传送数据。在被动模式下，FTP 的数据连接和控制连接的方向都是一致的，也就是说：是客户端向服务器发起一个用于数据传输的连接，客户端的连接端口是发起这个数据连接请求时使用的端口。客户端的控制连接和数据连接的端口号是大于 1024 的两个端口号（临时端口），而服务器端的数据端口是临时端口，而不一定是常规的 20。

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。

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）

- 1) 单击 Wireshark 工具栏左起第一个图标在接口上开始侦听，片刻后停止倾听。这时截获的数据量有多少？



截获的数据包为 541，见下图：

No.	Time	Source	Destination	Protocol	Length	Info
530	11.569542	172.18.34.177	255.255.255.255	UDP	76	57101 → 7533 Len=34
531	11.570477	AsustekC_2d:94:a6	Broadcast	ARP	60	Who has 172.18.35.254? Tell 172.18.32.3
532	11.588739	172.18.32.64	172.18.35.255	NBNS	92	Name query NB WPAD<00>
533	11.599734	fe80::cde8:1a64:6b2...	ff02::c	SSDP	208	M-SEARCH * HTTP/1.1
534	11.669356	fe80::3a22:d6ff:fee...	fe80::e9ec:b8b3:b2f...	ICMPv6	86	Neighbor Solicitation for fe80::e9ec:b8
535	11.669467	fe80::e9ec:b8b3:b2f...	fe80::3a22:d6ff:fee...	ICMPv6	86	Neighbor Advertisement fe80::e9ec:b8b3:
536	11.756904	172.18.34.179	172.18.35.255	UDP	305	54915 → 54915 Len=263
537	11.817109	172.18.35.138	172.18.35.255	UDP	305	54915 → 54915 Len=263
538	11.842005	172.18.35.168	225.0.0.222	UDP	216	53706 → 54997 Len=174
539	11.926503	172.18.34.246	172.18.35.255	UDP	119	7083 → 7083 Len=77
540	11.927281	172.18.34.246	172.18.35.255	UDP	119	7083 → 7083 Len=77
541	12.019204	fe80::2583:ba65:3e0...	ff02::1:3	LLMNR	152	Standard query 0x6f3c PTR 3.0.0.0.1.0.0

2) 观察截获数据源 IP 地址和目的 IP 地址，这些数据是发出的还是到达的？选择几个 IP 地址，通过网站 www.ip138.com 查询这些 IP 地址的地理位置。

您查询的IP:172.18.34.246

- 本站数据：本地局域网
- 参考数据1：局域网局域网
- 参考数据2：本地局域网
- 兼容IPv6地址：::AC12:22F6
- 映射IPv6地址：::FFFF:AC12:22F6

您查询的IP:211.159.235.146

- 本站数据：北京市北京市 腾讯云计算（北京）有限责任公司 腾讯云
- 参考数据1：北京北京 tencent.com 电信/联通/移动
- 参考数据2：北京市 联通
- 兼容IPv6地址：::D39F:EB92
- 映射IPv6地址：::FFFF:D39F:EB92

您查询的IP:221.179.183.59

- 本站数据：北京市北京市 移动
- 参考数据1：北京北京 移动
- 参考数据2：重庆市 移动
- 兼容IPv6地址：::DDB3:B73B
- 映射IPv6地址：::FFFF:DDB3:B73B



您查询的IP:157.147.36.45

- 本站数据: 美国
- 参考数据1: 日本日本 so-net.ne.jp
- 参考数据2: 美国
- 兼容IPv6地址: ::9D93:242D
- 映射IPv6地址: ::FFFF:9D93:242D

3) 在命令窗口运行 ipconfig /all 查看 IP 地址, 运行 ping -r 6 -l 200 172.18.34.180 和 ping -s 4 -l 200 172.18.34.180

```
C:\Users\Nby>ping -r 6 -l 200 172.18.34.180

正在 Ping 172.18.34.180 具有 200 字节的数据:
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128

172.18.34.180 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 = 0ms, 平均 = 0ms

C:\Users\Nby>ping -s 4 -l 200 172.18.34.180

正在 Ping 172.18.34.180 具有 200 字节的数据:
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128
来自 172.18.34.180 的回复: 字节=200 时间<1ms TTL=128

172.18.34.180 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 = 0ms, 平均 = 0ms
```

4) 执行 filter: ip.addr==172.18.34.180 命令查看

[ip.addr==172.18.34.180]									
No.	Time	Source	Destination	Protocol	Length	Info			
38	0.303887	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=33/8448, ttl=64 (reply in 41)			
41	0.342517	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=33/8448, ttl=46 (request in 38)			
75	1.310831	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=34/8704, ttl=64 (reply in 78)			
78	1.349019	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=34/8704, ttl=46 (request in 75)			
109	2.319574	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=35/8960, ttl=64 (reply in 114)			
114	2.359157	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=35/8960, ttl=46 (request in 109)			
172	3.325275	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=36/9216, ttl=64 (reply in 174)			
174	3.363470	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=36/9216, ttl=46 (request in 172)			
188	3.960022	172.18.34.180	112.47.5.99	TCP	55	2308 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=1 [TCP segment of a reassembled PDU]			
189	3.985103	112.47.5.99	172.18.34.180	TCP	66	443 → 2308 [ACK] Seq=1 Ack=2 Win=33 Len=0 SLE=1 SRE=2			
222	4.579950	172.18.34.180	180.163.251.137	TCP	55	2309 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=1 [TCP segment of a reassembled PDU]			
223	4.580436	172.18.34.180	180.163.251.137	TCP	55	2310 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=1 [TCP segment of a reassembled PDU]			
226	4.664032	180.163.251.137	172.18.34.180	TCP	66	443 → 2309 [ACK] Seq=1 Ack=2 Win=136 Len=0 SLE=1 SRE=2			

5) 捕获的数据中有 tcp, http, icmp 等不同的协议;

以上截图均为含有 Echo 和 Stamp 的请求和相应分组。字段从左到右的意思依次为: 数据传输发起地址、



目的地址、使用协议、响应时间和延时等。

No.	Time	Source	Destination	Protocol	Length	Info
38	0.303887	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=33/8448, ttl=64 (reply in 41)
41	0.342517	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=33/8448, ttl=46 (request in 38)
75	1.310831	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=34/8704, ttl=64 (reply in 78)
78	1.349019	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=34/8704, ttl=46 (request in 75)
109	2.319574	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=35/8960, ttl=64 (reply in 114)
114	2.359157	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=35/8960, ttl=46 (request in 109)
172	3.325275	172.18.34.180	123.125.115.110	ICMP	74	Echo (ping) request id=0x0001, seq=36/9216, ttl=64 (reply in 174)
174	3.363470	123.125.115.110	172.18.34.180	ICMP	74	Echo (ping) reply id=0x0001, seq=36/9216, ttl=46 (request in 172)

> Frame 38: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: HewlettP_40:ab:14 (ec:8e:b5:40:ab:14), Dst: Hangzhou_e5:b2:d4 (38:22:d6:e5:b2:d4)
> Internet Protocol Version 4, Src: 172.18.34.180, Dst: 123.125.115.110
v Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d3a [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence number (BE): 33 (0x0021)
Sequence number (LE): 8448 (0x2100)
[\[Response frame: 41\]](#)
> Data (32 bytes)

Type 为 echo 的类型，seq 表示数据总体的大小，其中 be 和 le 分别表示其最大和最小的值，ttl 指一个网络层的数据包的生存周期

学号	学生	自评分
<u>16340154</u>	刘硕	<u>99</u>
<u>16340148</u>	刘虹奇	<u>98</u>
<u>16340171</u>	聂博业	<u>98</u>

【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！