实验一 常用网络命令及 WireShark 使用

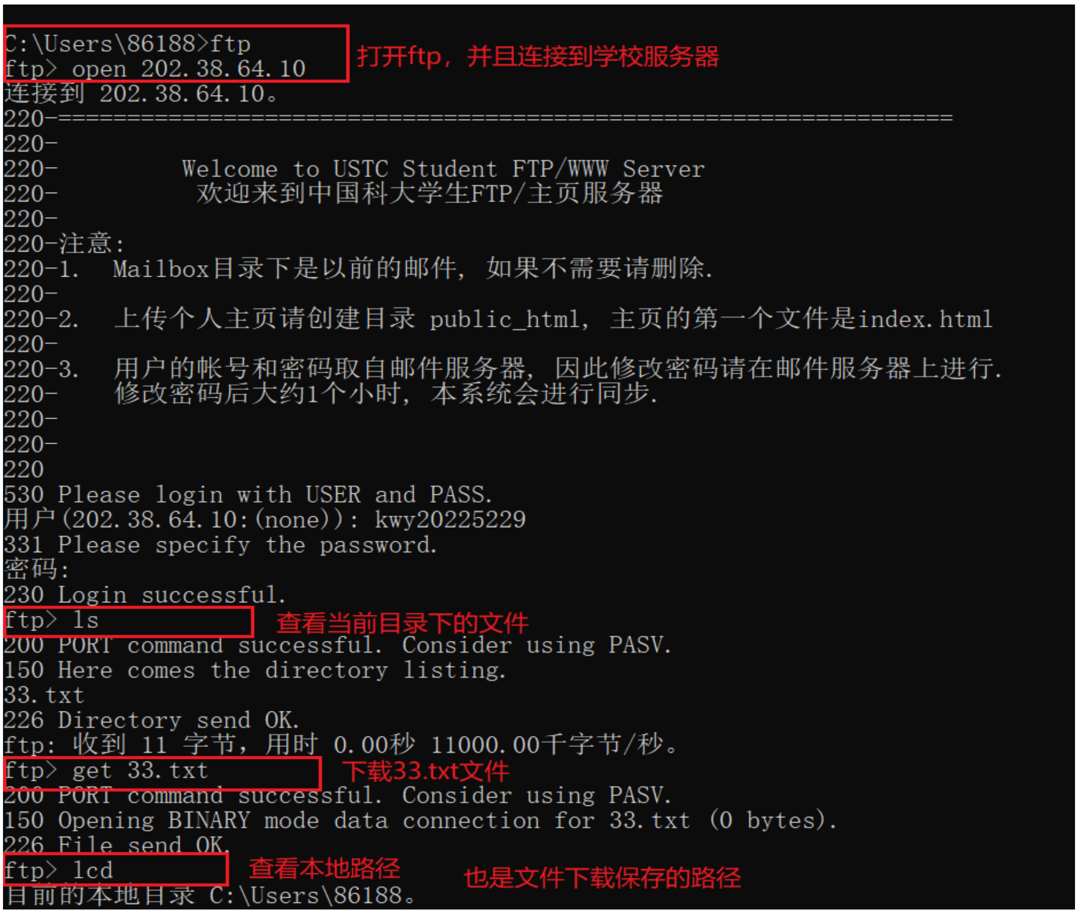
1.实验目的

- 掌握常用网络命令的使用方法 。
- 熟悉和掌握网络管理、网络维护的基本内容和方法 。
- 了解 WireShark 的使用方法和基本特点，掌握使用 WireShark 分析协议的方法。

2.实验内容

2.1 运行网络的基本命令，体会学习使用方法和相应的参数

2.1.1 FTP 的使用



```
C:\Users\86188>ftp
ftp> open 202.38.64.10
连接到 202.38.64.10。
220-=================================================================
220-
220-         Welcome to USTC Student FTP/WWW Server
220-          欢迎来到中国科大学生FTP/主页服务器
220-
220-注意:
220-1.  Mailbox目录下是以前的邮件, 如果不需要请删除.
220-
220-2.  上传个人主页请创建目录 public_html, 主页的第一个文件是index.html
220-
220-3.  用户的帐号和密码取自邮件服务器, 因此修改密码请在邮件服务器上进行.
220-    修改密码后大约1个小时, 本系统会进行同步.
220-
220-
220
530 Please login with USER and PASS.
用户(202.38.64.10:(none)): kwy20225229
331 Please specify the password.
密码:
230 Login successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
33.txt
226 Directory send OK.
ftp: 收到 11 字节，用时 0.00秒 11000.00千字节/秒。
ftp> get 33.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for 33.txt (0 bytes).
226 File send OK.
ftp> lcd
目前的本地目录 C:\Users\86188。
```

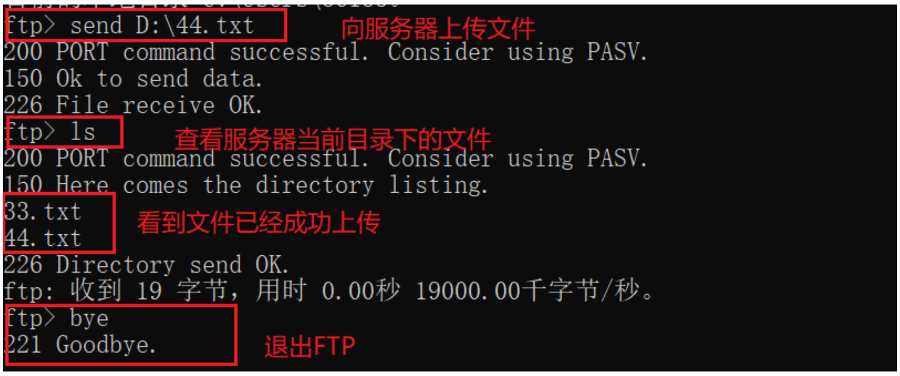
打开ftp，并且连接到学校服务器

查看当前目录下的文件

下载33.txt文件

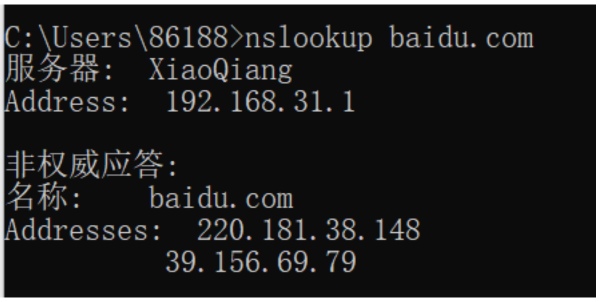
查看本地路径

也是文件下载保存的路径



```
ftp> send D:\44.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 File receive OK.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
33.txt
44.txt
226 Directory send OK.
ftp: 收到 19 字节，用时 0.00秒 19000.00千字节/秒。
ftp> bye
221 Goodbye.
```

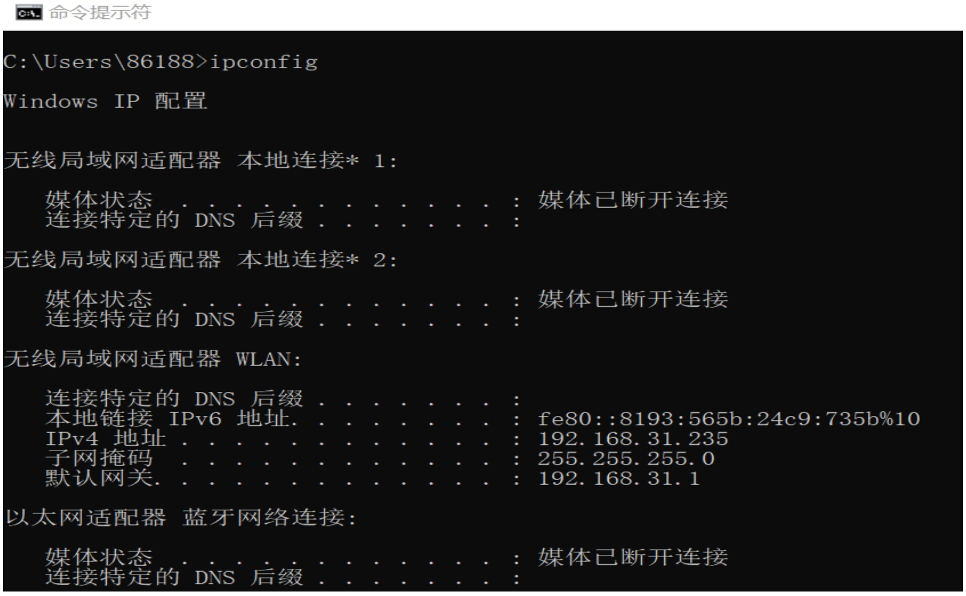
2.1.2 nslookup 的使用



```
C:\Users\86188>nslookup baidu.com
服务器:  XiaoQiang
Address:  192.168.31.1

非权威应答:
名称:    baidu.com
Addresses:  220.181.38.148
          39.156.69.79
```

Nslookup 用于查询 DNS 的记录和域名解析是否正常，如图所示，默认的 DNS 服务器是 XiaoQiang,ip 地址是 192.168.31.1。百度的 ip 地址是 220.181.38.148 和 39.156.69.79。

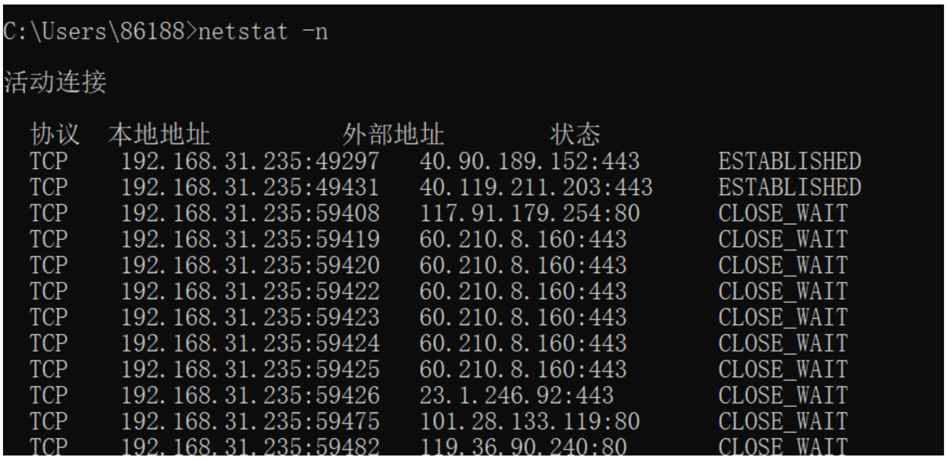
2.1.3 ipconfig 的使用



```
C:\Users\86188>ipconfig

Windows IP 配置

无线局域网适配器 本地连接* 1:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

无线局域网适配器 本地连接* 2:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::8193:565b:24c9:735b%10
   IPv4 地址 . . . . . . . . . . . . : 192.168.31.235
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.31.1

以太网适配器 蓝牙网络连接:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
```

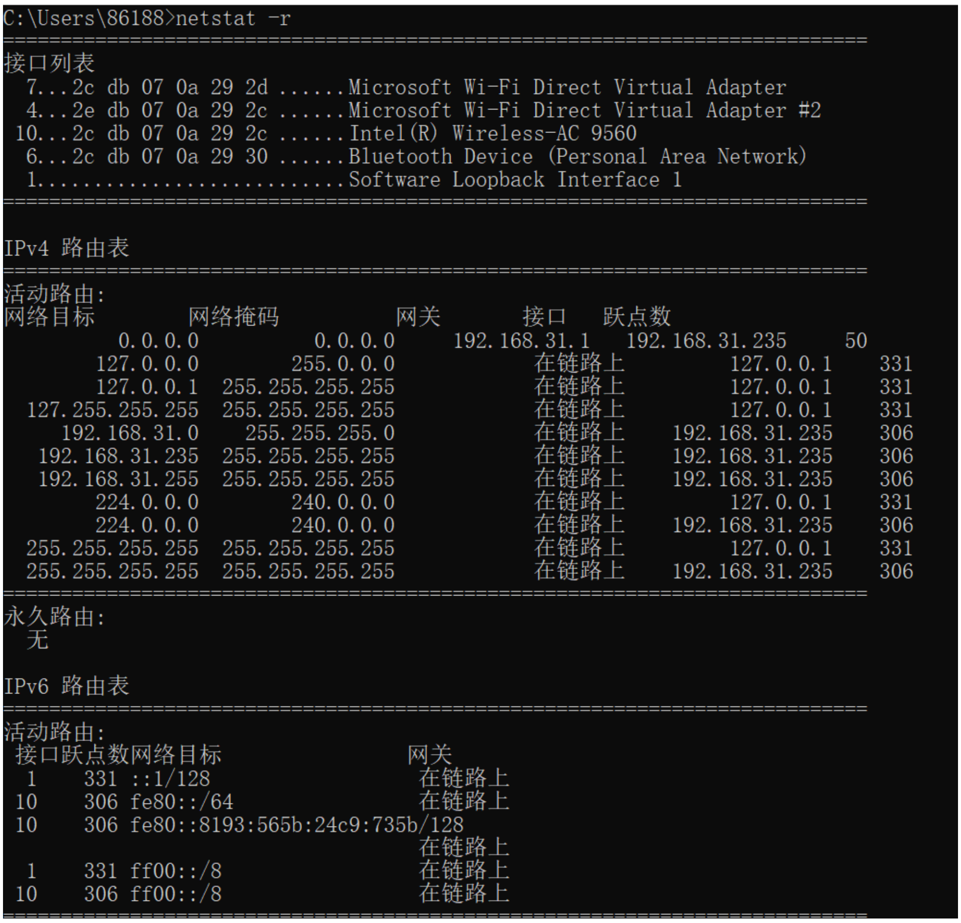
ipconfig 可查看电脑 ip 参数配置信息，如 ip 地址、默认网关、子网掩码、DNS（域名服务)、WINS 服务器等地址。如图所示本机的 ip 地址是 192.168.31.235，子网掩码是 255.255.255.0,默认网关 shi192.168.31.1。

2.1.4 netstat 的使用



```
C:\Users\86188>netstat -n

活动连接

  协议  本地地址          外部地址        状态
  TCP    192.168.31.235:49297   40.90.189.152:443      ESTABLISHED
  TCP    192.168.31.235:49431   40.119.211.203:443     ESTABLISHED
  TCP    192.168.31.235:59408   117.91.179.254:80      CLOSE_WAIT
  TCP    192.168.31.235:59419   60.210.8.160:443       CLOSE_WAIT
  TCP    192.168.31.235:59420   60.210.8.160:443       CLOSE_WAIT
  TCP    192.168.31.235:59422   60.210.8.160:443       CLOSE_WAIT
  TCP    192.168.31.235:59423   60.210.8.160:443       CLOSE_WAIT
  TCP    192.168.31.235:59424   60.210.8.160:443       CLOSE_WAIT
  TCP    192.168.31.235:59425   60.210.8.160:443       CLOSE_WAIT
  TCP    192.168.31.235:59426   23.1.246.92:443        CLOSE_WAIT
  TCP    192.168.31.235:59475   101.28.133.119:80      CLOSE_WAIT
  TCP    192.168.31.235:59482   119.36.90.240:80       CLOSE_WAIT
```

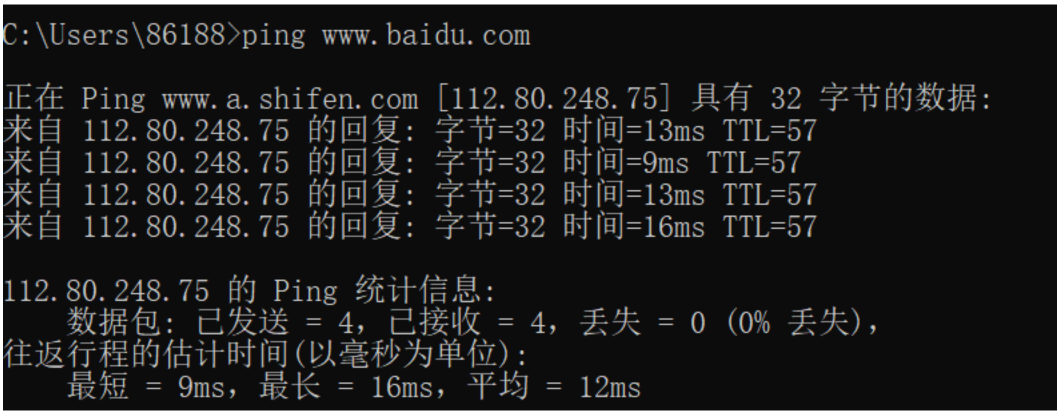
netstat 命令用于查看网络连接，路由表，网络接口统计数据，虚拟连接等信息。



```
C:\Users\86188>netstat -r
===========================================================================
接口列表
  7...2c db 07 0a 29 2d ......Microsoft Wi-Fi Direct Virtual Adapter
  4...2e db 07 0a 29 2c ......Microsoft Wi-Fi Direct Virtual Adapter #2
 10...2c db 07 0a 29 2c ......Intel(R) Wireless-AC 9560
  6...2c db 07 0a 29 30 ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0     192.168.31.1   192.168.31.235     50
        127.0.0.0        255.0.0.0            在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255            在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
     192.168.31.0    255.255.255.0            在链路上    192.168.31.235    306
   192.168.31.235  255.255.255.255            在链路上    192.168.31.235    306
   192.168.31.255  255.255.255.255            在链路上    192.168.31.235    306
        224.0.0.0        240.0.0.0            在链路上         127.0.0.1    331
        224.0.0.0        240.0.0.0            在链路上    192.168.31.235    306
  255.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
  255.255.255.255  255.255.255.255            在链路上    192.168.31.235    306
===========================================================================
永久路由:
  无

IPv6 路由表
===========================================================================
活动路由:
 接口跃点数网络目标                网关
  1    331 ::1/128                  在链路上
 10    306 fe80::/64                在链路上
 10    306 fe80::8193:565b:24c9:735b/128
                                    在链路上
  1    331 ff00::/8                 在链路上
 10    306 ff00::/8                 在链路上
===========================================================================
```

使用-r(route)参数可以查看路由相关信息。

2.1.5 ping 的使用（具体介绍参考 2.2.1）



```
C:\Users\86188>ping www.baidu.com

正在 Ping www.a.shifen.com [112.80.248.75] 具有 32 字节的数据:
来自 112.80.248.75 的回复: 字节=32 时间=13ms TTL=57
来自 112.80.248.75 的回复: 字节=32 时间=9ms TTL=57
来自 112.80.248.75 的回复: 字节=32 时间=13ms TTL=57
来自 112.80.248.75 的回复: 字节=32 时间=16ms TTL=57

112.80.248.75 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 = 9ms，最长 = 16ms，平均 = 12ms
```

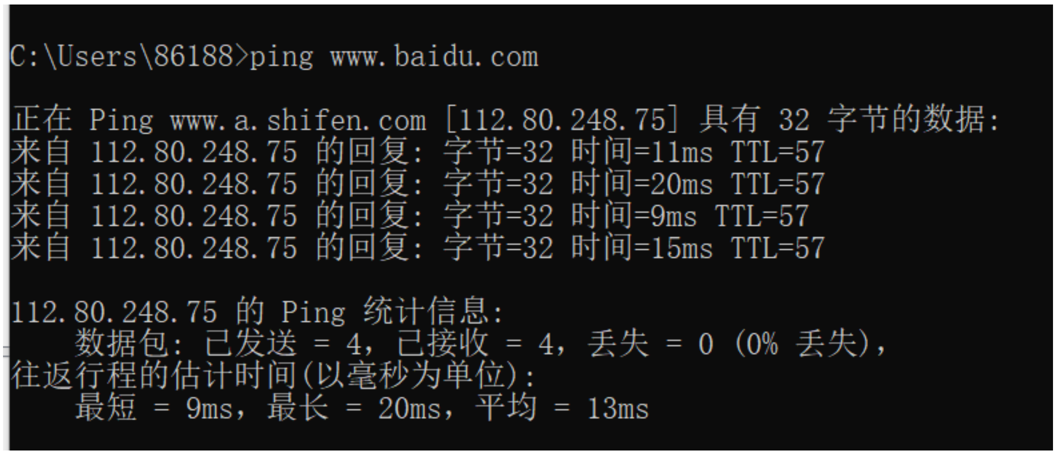
2.2 使用 WireShark 抓取数据包

2.2.1

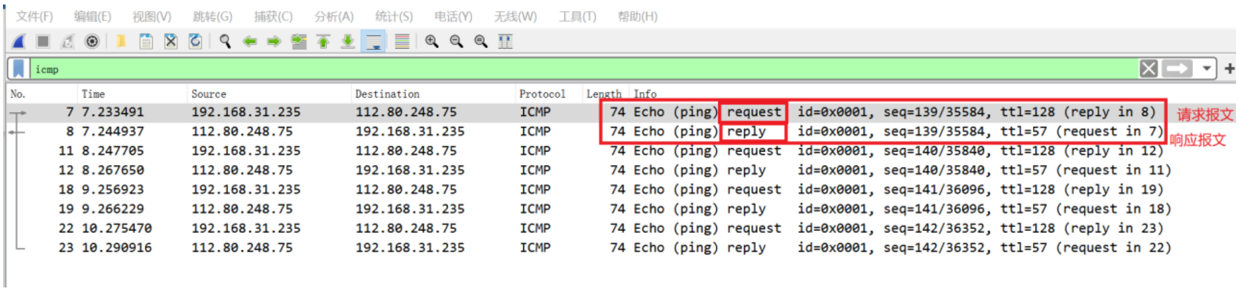
实验要求：使用 ping 命令，抓取 ping www.baidu.com 后的数据包进行分析。写出源 ip 地 址、目的 ip 地址、IP 标识、总长度、TTL 值这几项。

ping 的原理是：

用来测试两个主机之间的连通性。Ping 命令属于应用层程序，但它直接使用网络层的 ICMP 协议，首先由源主机发送一个请求报文，然后目的主机收到后发送一个响应报文。在 ping 中会发送四次这样的请求，如果两个主机联通，就会收到四次响应报文。

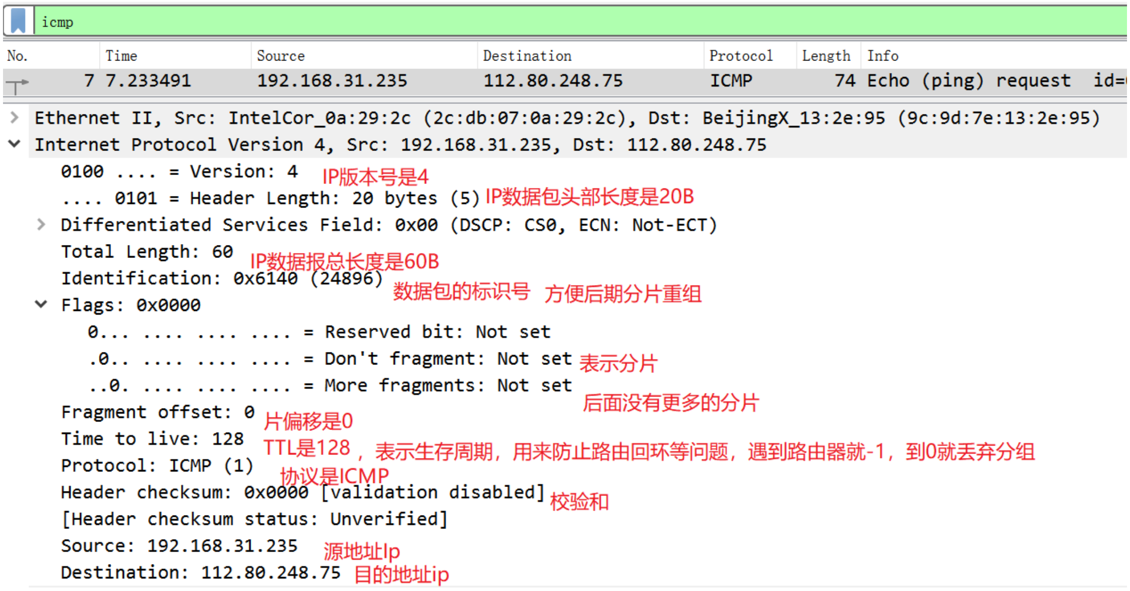


```
C:\Users\86188>ping www.baidu.com

正在 Ping www.a.shifen.com [112.80.248.75] 具有 32 字节的数据:
来自 112.80.248.75 的回复: 字节=32 时间=11ms TTL=57
来自 112.80.248.75 的回复: 字节=32 时间=20ms TTL=57
来自 112.80.248.75 的回复: 字节=32 时间=9ms TTL=57
来自 112.80.248.75 的回复: 字节=32 时间=15ms TTL=57

112.80.248.75 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 = 9ms，最长 = 20ms，平均 = 13ms
```



```
icmp
No.    Time       Source           Destination      Protocol  Length  Info
     7 7.233491   192.168.31.235   112.80.248.75    ICMP        74 Echo (ping) request  id=0x0001, seq=139/35584, ttl=128 (reply in 8)
     8 7.244937   112.80.248.75    192.168.31.235   ICMP        74 Echo (ping) reply    id=0x0001, seq=139/35584, ttl=57 (request in 7)
    11 8.247705   192.168.31.235   112.80.248.75    ICMP        74 Echo (ping) request  id=0x0001, seq=140/35840, ttl=128 (reply in 12)
    12 8.267650   112.80.248.75    192.168.31.235   ICMP        74 Echo (ping) reply    id=0x0001, seq=140/35840, ttl=57 (request in 11)
    18 9.256923   192.168.31.235   112.80.248.75    ICMP        74 Echo (ping) request  id=0x0001, seq=141/36096, ttl=128 (reply in 19)
    19 9.266229   112.80.248.75    192.168.31.235   ICMP        74 Echo (ping) reply    id=0x0001, seq=141/36096, ttl=57 (request in 18)
    22 10.275470  192.168.31.235   112.80.248.75    ICMP        74 Echo (ping) request  id=0x0001, seq=142/36352, ttl=128 (reply in 23)
    23 10.290916  112.80.248.75    192.168.31.235   ICMP        74 Echo (ping) reply    id=0x0001, seq=142/36352, ttl=57 (request in 22)
```

抓包到八个 ICMP 报文，共有四组，每一组包括一个请求报文和一个响应报文。

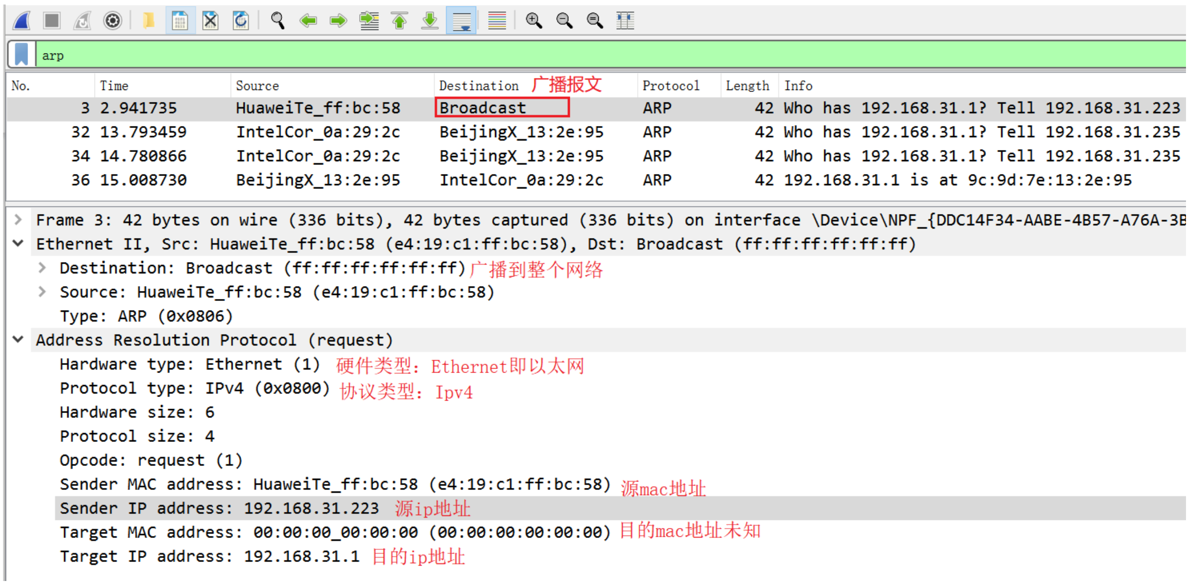


```
icmp
No.    Time        Source           Destination      Protocol  Length  Info
     7 7.233491    192.168.31.235   112.80.248.75    ICMP        74 Echo (ping) request  id=
> Ethernet II, Src: IntelCor_0a:29:2c (2c:db:07:0a:29:2c), Dst: BeijingX_13:2e:95 (9c:9d:7e:13:2e:95)
v Internet Protocol Version 4, Src: 192.168.31.235, Dst: 112.80.248.75
    0100 .... = Version: 4  IP版本号是4
    .... 0101 = Header Length: 20 bytes (5) IP数据包头部长度是20B
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60  IP数据报总长度是60B
    Identification: 0x6140 (24896)  数据包的标识号  方便后期分片重组
  v Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set 表示分片
      ..0. .... .... .... = More fragments: Not set
                                                   后面没有更多的分片
    Fragment offset: 0  片偏移是0
    Time to live: 128  TTL是128，表示生存周期，用来防止路由回环等问题，遇到路由器就-1，到0就丢弃分组
    Protocol: ICMP (1)  协议是ICMP
    Header checksum: 0x0000 [validation disabled] 校验和
    [Header checksum status: Unverified]
    Source: 192.168.31.235  源地址Ip
    Destination: 112.80.248.75  目的地址ip
```

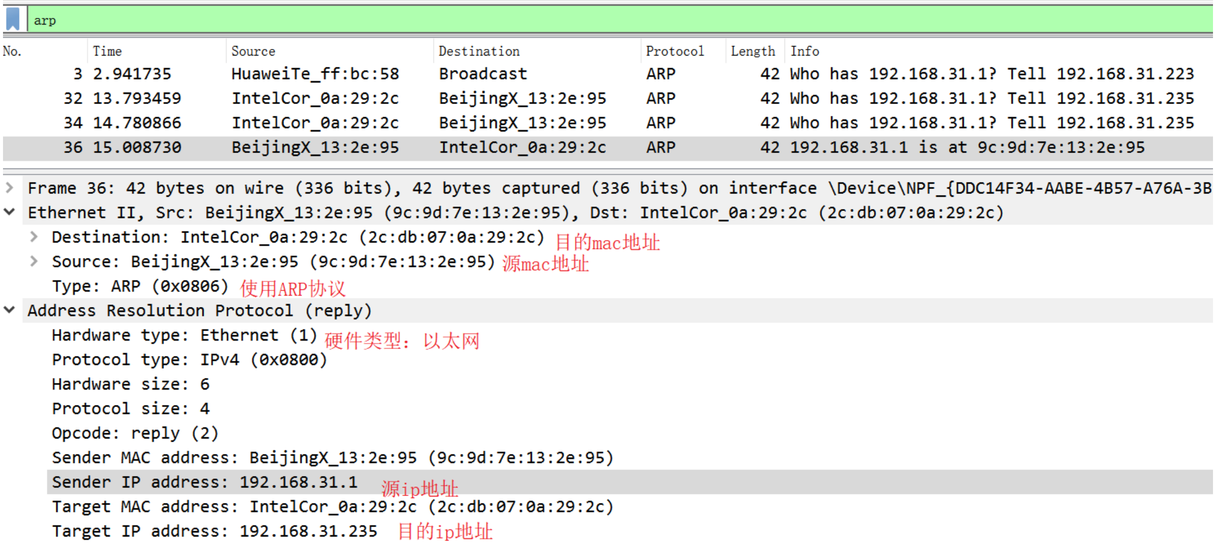
2.2.2

实验要求：通过筛选得到 arp 数据包，分析某个 arp 数据包的硬件类型、协议类型、发 送源 ip 地址、发送源 MAC 地址、目的主机 ip 地址、目的 MAC 地址。

ARP 原理：ARP 是网络层协议，用来获得某已知 IP 主机的 mac 地址。首先源主机会向网络发送一个广播请求报文，当目的主机接收到广播请求报文之后，匹配 ip,之后发送响应报文（单播）,在报文里有 Mac 地址。



```
arp
No.   Time        Source              Destination  广播报文   Protocol  Length  Info
    3 2.941735    HuaweiTe_ff:bc:58   Broadcast              ARP           42 Who has 192.168.31.1? Tell 192.168.31.223
   32 13.793459   IntelCor_0a:29:2c   BeijingX_13:2e:95      ARP           42 Who has 192.168.31.1? Tell 192.168.31.235
   34 14.780866   IntelCor_0a:29:2c   BeijingX_13:2e:95      ARP           42 Who has 192.168.31.1? Tell 192.168.31.235
   36 15.008730   BeijingX_13:2e:95   IntelCor_0a:29:2c      ARP           42 192.168.31.1 is at 9c:9d:7e:13:2e:95

> Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{DDC14F34-AABE-4B57-A76A-3B
v Ethernet II, Src: HuaweiTe_ff:bc:58 (e4:19:c1:ff:bc:58), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff) 广播到整个网络
  > Source: HuaweiTe_ff:bc:58 (e4:19:c1:ff:bc:58)
    Type: ARP (0x0806)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)  硬件类型：Ethernet即以太网
    Protocol type: IPv4 (0x0800) 协议类型：Ipv4
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HuaweiTe_ff:bc:58 (e4:19:c1:ff:bc:58) 源mac地址
    Sender IP address: 192.168.31.223  源ip地址
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00) 目的mac地址未知
    Target IP address: 192.168.31.1  目的ip地址
```



```
arp
No.   Time        Source              Destination         Protocol  Length  Info
    3 2.941735    HuaweiTe_ff:bc:58   Broadcast           ARP           42 Who has 192.168.31.1? Tell 192.168.31.223
   32 13.793459   IntelCor_0a:29:2c   BeijingX_13:2e:95   ARP           42 Who has 192.168.31.1? Tell 192.168.31.235
   34 14.780866   IntelCor_0a:29:2c   BeijingX_13:2e:95   ARP           42 Who has 192.168.31.1? Tell 192.168.31.235
   36 15.008730   BeijingX_13:2e:95   IntelCor_0a:29:2c   ARP           42 192.168.31.1 is at 9c:9d:7e:13:2e:95

> Frame 36: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{DDC14F34-AABE-4B57-A76A-3B
v Ethernet II, Src: BeijingX_13:2e:95 (9c:9d:7e:13:2e:95), Dst: IntelCor_0a:29:2c (2c:db:07:0a:29:2c)
  > Destination: IntelCor_0a:29:2c (2c:db:07:0a:29:2c) 目的mac地址
  > Source: BeijingX_13:2e:95 (9c:9d:7e:13:2e:95) 源mac地址
    Type: ARP (0x0806) 使用ARP协议
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1) 硬件类型：以太网
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: BeijingX_13:2e:95 (9c:9d:7e:13:2e:95)
    Sender IP address: 192.168.31.1  源ip地址
    Target MAC address: IntelCor_0a:29:2c (2c:db:07:0a:29:2c)
    Target IP address: 192.168.31.235  目的ip地址
```

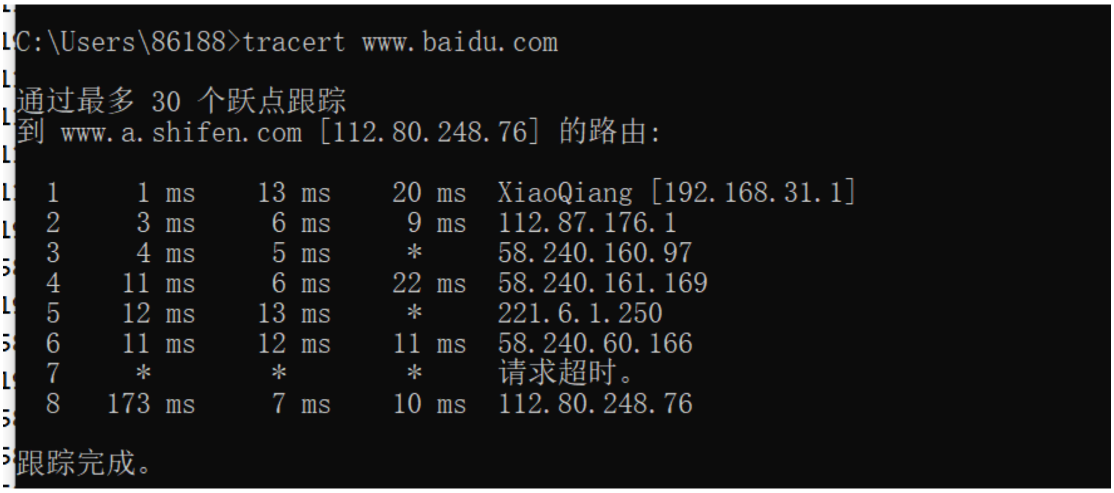
2.2.3

实验要求：tracert 跟踪百度，捕捉 tracert 使用的 icmp 报文，显示该数据包的 TTL 域 。

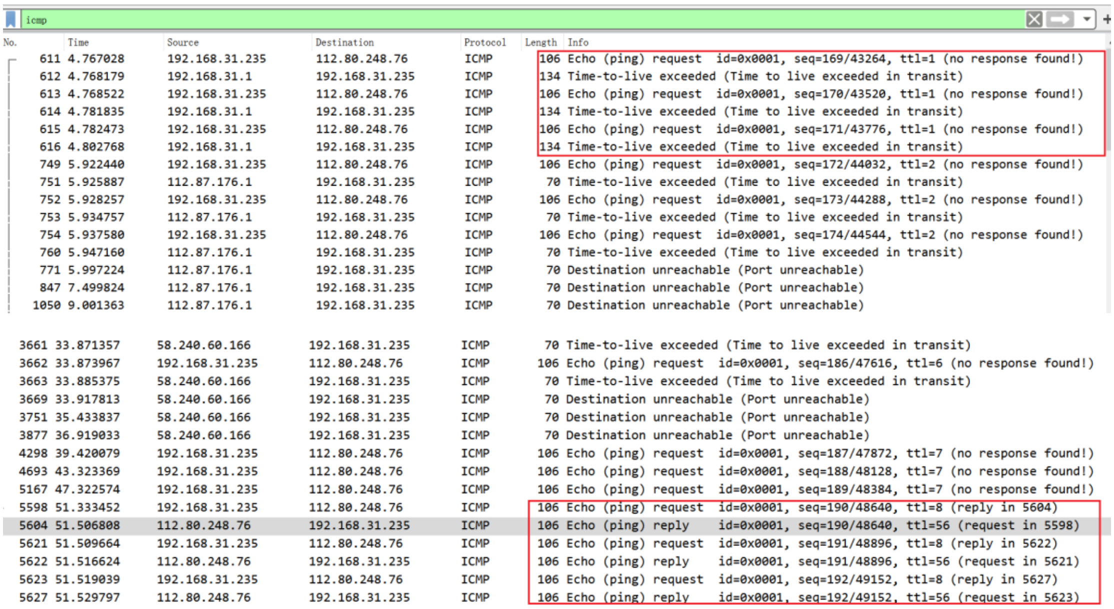
Tracert 原理：tracert 会追踪最多 30 个跃点追踪到目的地址，每次追踪会发送三个 ICMP 报文（为了计算平均时延），第一次设置 TTL 为 1，后面每次增加 1，直到追踪到目的地址。

tracert 命令是基于 ICMP 协议实现的，即直接发送一个 ICMP 回显请求（echo request）数据包，服务器在收到回显请求的时候会向客户端发送一个 ICMP 回显应答（echo reply）数据包，tracert 跟踪路由时，每当 TTL 减为 0 时，路

由就会往源主机发送一个 ICMP 超时报文，当到达目的主机时，目的主机回向源主机发送一个 ICMP 回显应答（echo reply）数据包，并将 TTL 设为较大的默认值，防止包丢失；通过 tracert 命令我们可以知道 ip 分组到达目的主机经过了哪些路由器，以及经过每一跳的网络延迟。



```
C:\Users\86188>tracert www.baidu.com

通过最多 30 个跃点跟踪
到 www.a.shifen.com [112.80.248.76] 的路由:

  1     1 ms    13 ms    20 ms  XiaoQiang [192.168.31.1]
  2     3 ms     6 ms     9 ms  112.87.176.1
  3     4 ms     5 ms     *     58.240.160.97
  4    11 ms     6 ms    22 ms  58.240.161.169
  5    12 ms    13 ms     *     221.6.1.250
  6    11 ms    12 ms    11 ms  58.240.60.166
  7     *        *        *     请求超时。
  8   173 ms     7 ms    10 ms  112.80.248.76

跟踪完成。
```

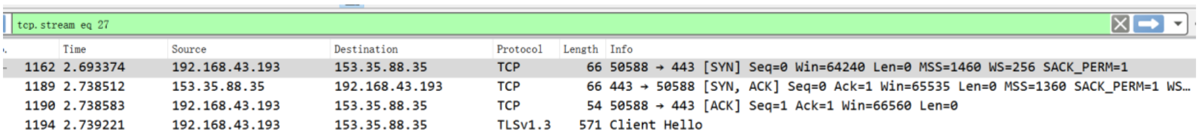


```
icmp
No.     Time       Source          Destination     Protocol Length Info
    611 4.767028   192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=169/43264, ttl=1 (no response found!)
    612 4.768179   192.168.31.1    192.168.31.235  ICMP     134 Time-to-live exceeded (Time to live exceeded in transit)
    613 4.768522   192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=170/43520, ttl=1 (no response found!)
    614 4.781835   192.168.31.1    192.168.31.235  ICMP     134 Time-to-live exceeded (Time to live exceeded in transit)
    615 4.782473   192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=171/43776, ttl=1 (no response found!)
    616 4.802768   192.168.31.1    192.168.31.235  ICMP     134 Time-to-live exceeded (Time to live exceeded in transit)
    749 5.922440   192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=172/44032, ttl=2 (no response found!)
    751 5.925887   112.87.176.1    192.168.31.235  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
    752 5.928257   192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=173/44288, ttl=2 (no response found!)
    753 5.934757   112.87.176.1    192.168.31.235  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
    754 5.937580   192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=174/44544, ttl=2 (no response found!)
    760 5.947160   112.87.176.1    192.168.31.235  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
    771 5.997224   112.87.176.1    192.168.31.235  ICMP      70 Destination unreachable (Port unreachable)
    847 7.499824   112.87.176.1    192.168.31.235  ICMP      70 Destination unreachable (Port unreachable)
   1050 9.001363   112.87.176.1    192.168.31.235  ICMP      70 Destination unreachable (Port unreachable)

   3661 33.871357  58.240.60.166   192.168.31.235  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
   3662 33.873967  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=186/47616, ttl=6 (no response found!)
   3663 33.885375  58.240.60.166   192.168.31.235  ICMP      70 Time-to-live exceeded (Time to live exceeded in transit)
   3669 33.917813  58.240.60.166   192.168.31.235  ICMP      70 Destination unreachable (Port unreachable)
   3751 35.433837  58.240.60.166   192.168.31.235  ICMP      70 Destination unreachable (Port unreachable)
   3877 36.919033  58.240.60.166   192.168.31.235  ICMP      70 Destination unreachable (Port unreachable)
   4298 39.420079  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=187/47872, ttl=7 (no response found!)
   4693 43.323369  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=188/48128, ttl=7 (no response found!)
   5167 47.322574  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=189/48384, ttl=7 (no response found!)
   5598 51.333452  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=190/48640, ttl=8 (reply in 5604)
   5604 51.506808  112.80.248.76   192.168.31.235  ICMP     106 Echo (ping) reply    id=0x0001, seq=190/48640, ttl=56 (request in 5598)
   5621 51.509664  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=191/48896, ttl=8 (reply in 5622)
   5622 51.516624  112.80.248.76   192.168.31.235  ICMP     106 Echo (ping) reply    id=0x0001, seq=191/48896, ttl=56 (request in 5621)
   5623 51.519039  192.168.31.235  112.80.248.76   ICMP     106 Echo (ping) request  id=0x0001, seq=192/49152, ttl=8 (reply in 5627)
   5627 51.529797  112.80.248.76   192.168.31.235  ICMP     106 Echo (ping) reply    id=0x0001, seq=192/49152, ttl=56 (request in 5623)
```

2.2.4

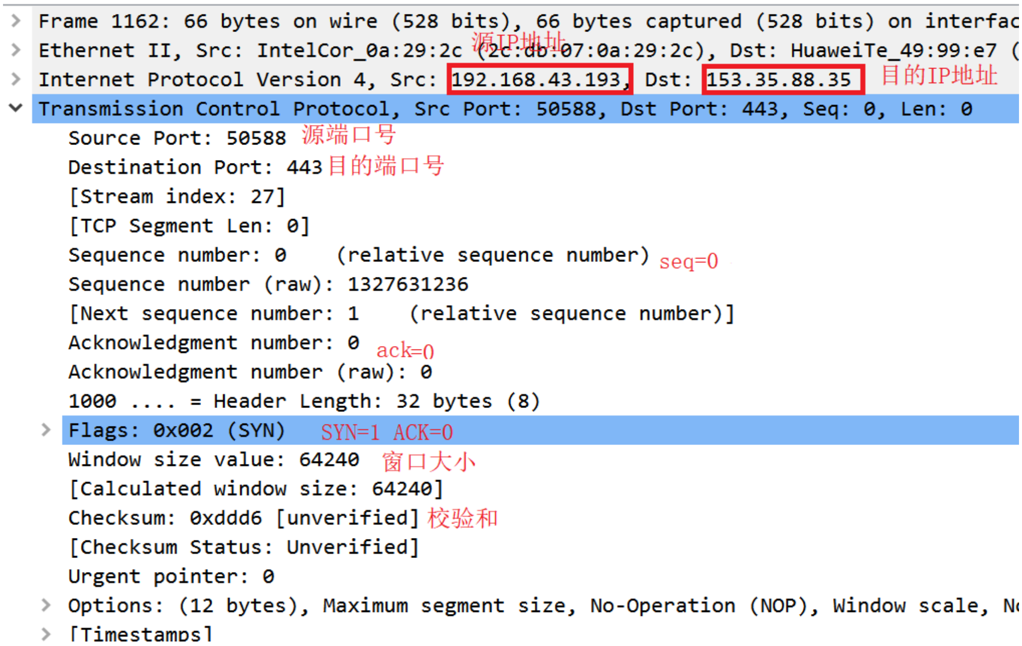
实验要求：对 TCP 三次握手、四次挥手过程进行抓包分析，并通过抓取的包进行握手 与挥手过程，通过截图体现传输内容。写出某 TCP 数据包的源 ip 地址、目 的 ip 地址、源端口、目的端口、窗口大小，以及三次握手和四次挥手中 SYN、ACK、seq 和 ack 等值。

TCP 三次握手：



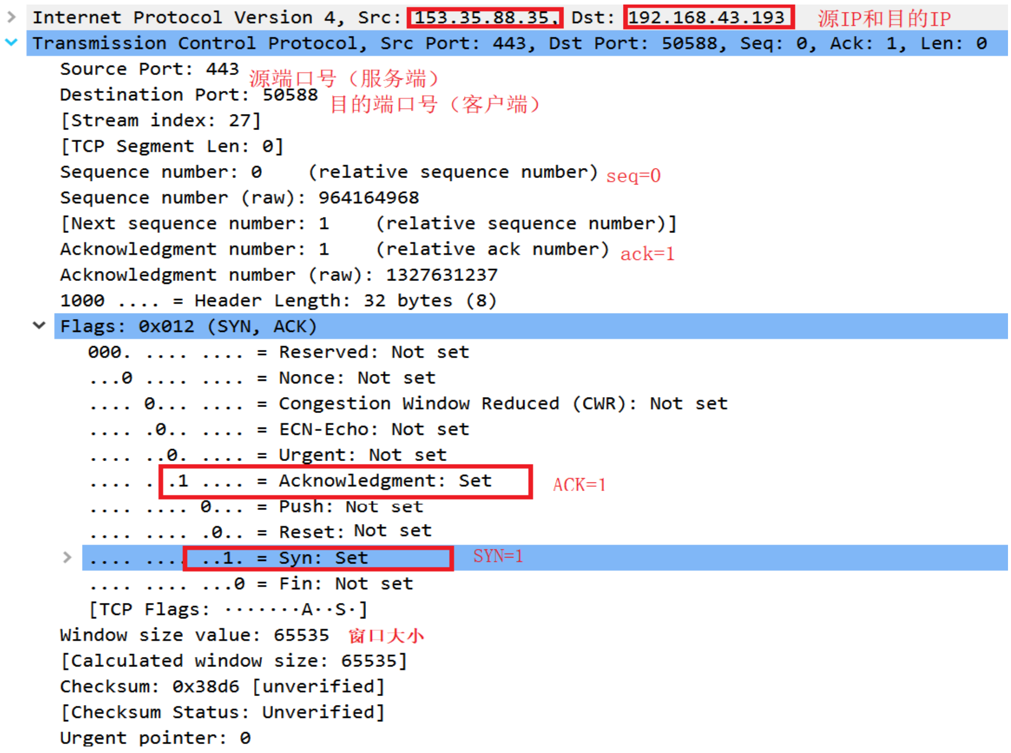
```
tcp.stream eq 27
     Time        Source          Destination    Protocol Length Info
1162 2.693374    192.168.43.193  153.35.88.35   TCP         66 50588 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1189 2.738512    153.35.88.35    192.168.43.193 TCP         66 443 → 50588 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1360 SACK_PERM=1 WS…
1190 2.738583    192.168.43.193  153.35.88.35   TCP         54 50588 → 443 [ACK] Seq=1 Ack=1 Win=66560 Len=0
1194 2.739221    192.168.43.193  153.35.88.35   TLSv1.3    571 Client Hello
```

•第一次握手：客户端发送一个 SYN=1 的报文给服务器，要求建立数据连接。



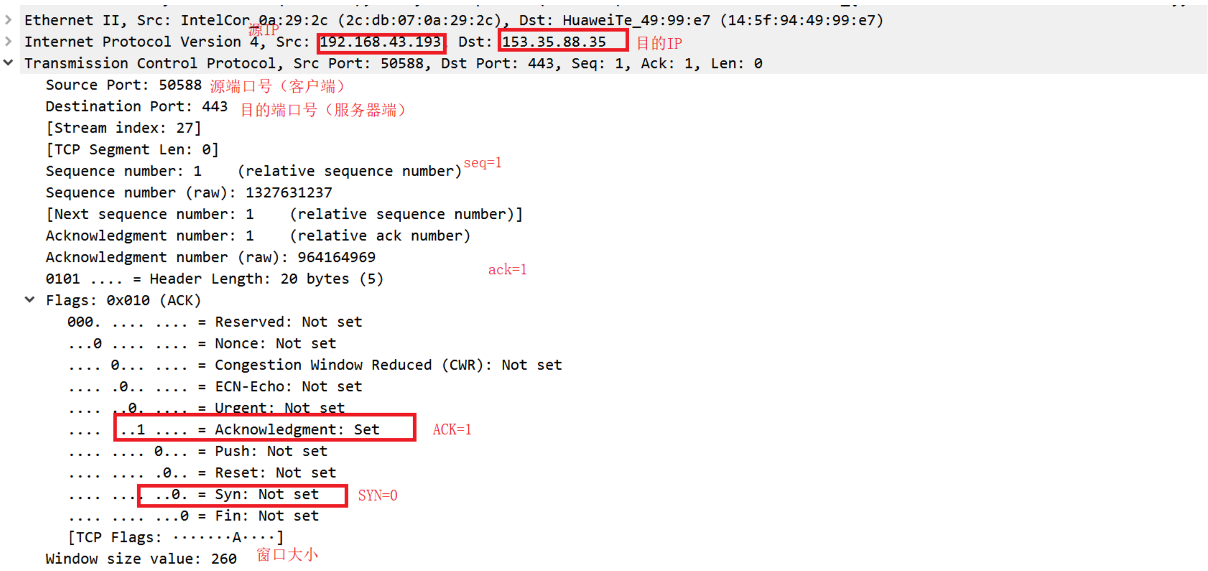
```
> Frame 1162: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interfac
> Ethernet II, Src: IntelCor_0a:29:2c (2c:db:07:0a:29:2c), Dst: HuaweiTe_49:99:e7 (
> Internet Protocol Version 4, Src: 192.168.43.193, Dst: 153.35.88.35
v Transmission Control Protocol, Src Port: 50588, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 50588
    Destination Port: 443
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 1327631236
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0xddd6 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, N
  > [Timestamps]
```

•第二次握手：服务器发送 SYN+ACK 确认报文给客户端， 此时 seq(序列号)=0，ack=1(上一个报文的 seq+1,表示想要收到 seq=1 的报文)。



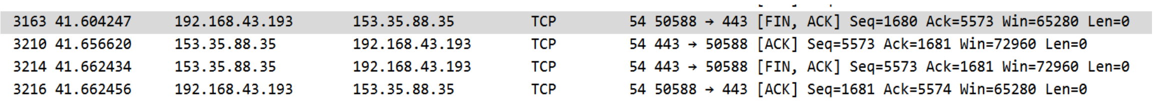
```
> Internet Protocol Version 4, Src: 153.35.88.35, Dst: 192.168.43.193
v Transmission Control Protocol, Src Port: 443, Dst Port: 50588, Seq: 0, Ack: 1, Len: 0
    Source Port: 443
    Destination Port: 50588
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 964164968
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 1327631237
    1000 .... = Header Length: 32 bytes (8)
  v 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 size value: 65535
    [Calculated window size: 65535]
    Checksum: 0x38d6 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
```

•第三次握手：客户端再次发送 ACK 向服务器，服务器验证 ACK 没有问题，则建立起连接。此时 seq=1, ack=1(上次报文的 seq+1)。



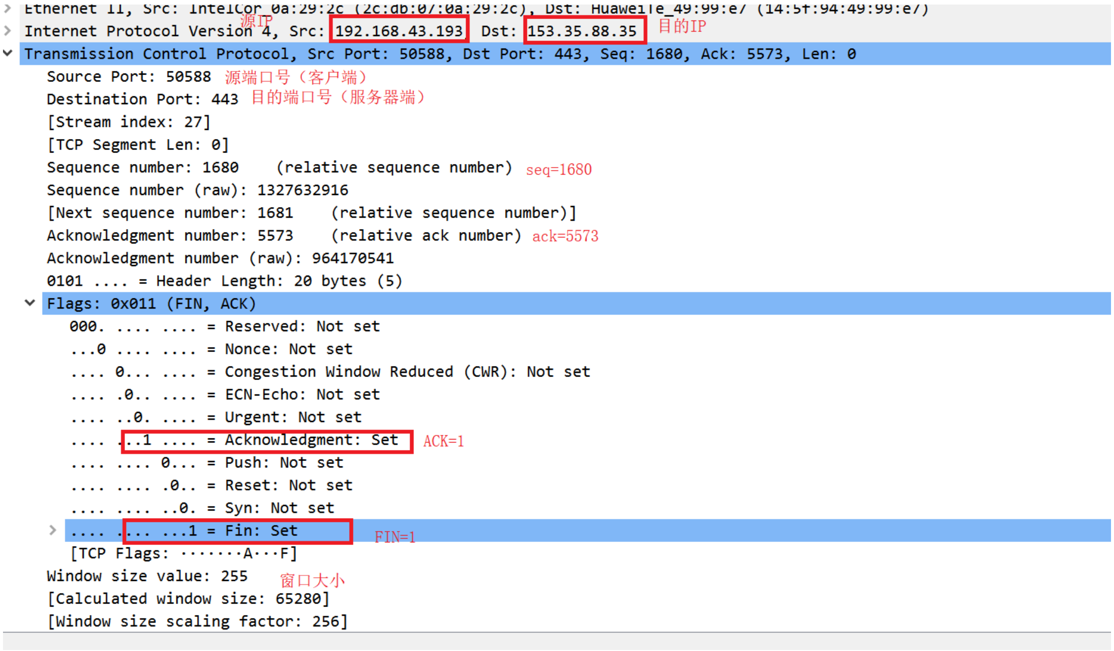
```
> Ethernet II, Src: IntelCor_0a:29:2c (2c:db:07:0a:29:2c), Dst: HuaweiTe_49:99:e7 (14:5f:94:49:99:e7)
> Internet Protocol Version 4, Src: 192.168.43.193, Dst: 153.35.88.35
v Transmission Control Protocol, Src Port: 50588, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
    Source Port: 50588
    Destination Port: 443
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 1327631237
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 964164969
    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····]
    Window size value: 260
```

四次挥手



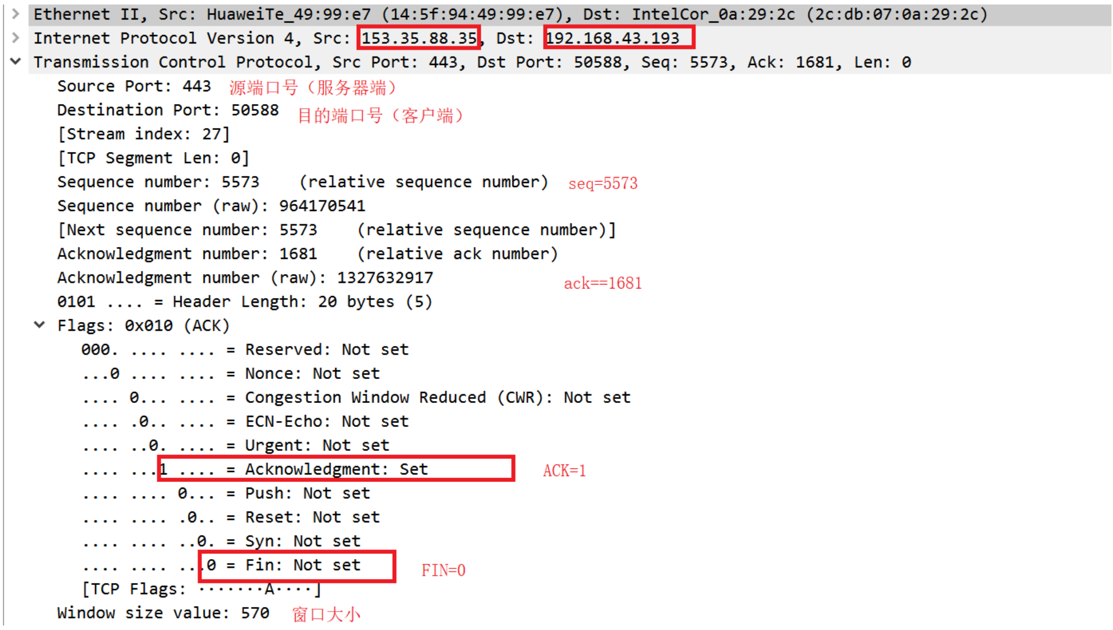
```
3163 41.604247    192.168.43.193    153.35.88.35      TCP    54 50588 → 443 [FIN, ACK] Seq=1680 Ack=5573 Win=65280 Len=0
3210 41.656620    153.35.88.35      192.168.43.193    TCP    54 443 → 50588 [ACK] Seq=5573 Ack=1681 Win=72960 Len=0
3214 41.662434    153.35.88.35      192.168.43.193    TCP    54 443 → 50588 [FIN, ACK] Seq=5573 Ack=1681 Win=72960 Len=0
3216 41.662456    192.168.43.193    153.35.88.35      TCP    54 50588 → 443 [ACK] Seq=1681 Ack=5574 Win=65280 Len=0
```

•第一次挥手：客户端发送 FIN(结束)报文，通知服务器数据已经传输完毕；



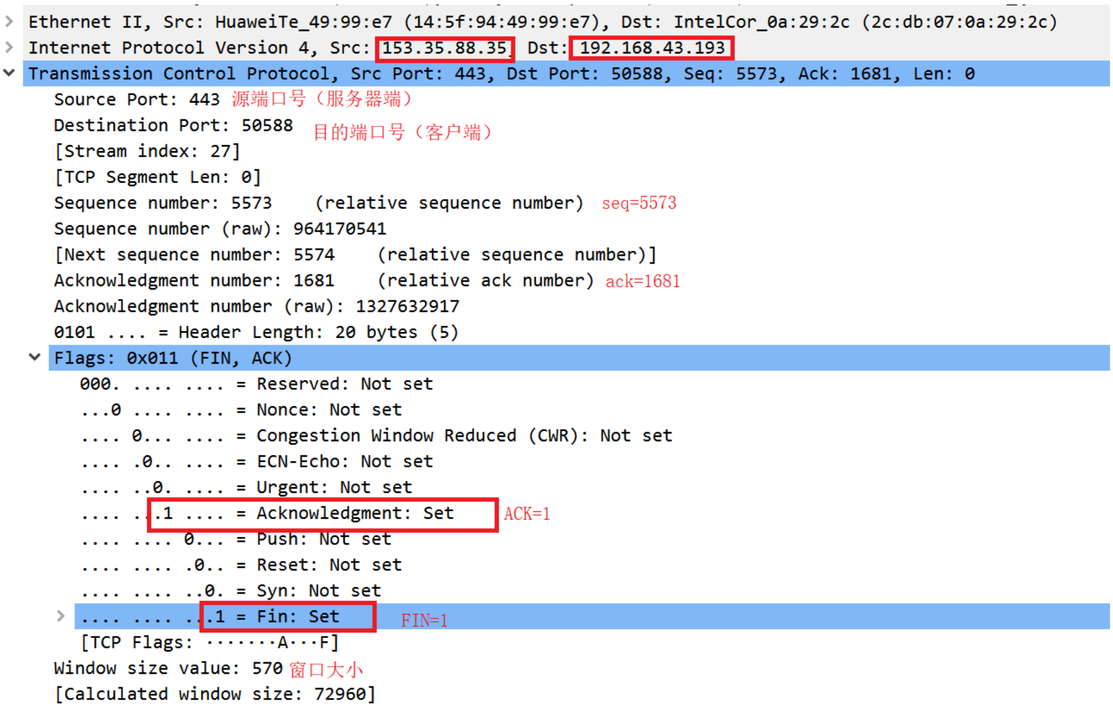
```
> Ethernet II, Src: IntelCor_0a:29:2c (2c:db:07:0a:29:2c), Dst: HuaweiTe_49:99:e7 (14:5f:94:49:99:e7)
> Internet Protocol Version 4, Src: 192.168.43.193, Dst: 153.35.88.35
v Transmission Control Protocol, Src Port: 50588, Dst Port: 443, Seq: 1680, Ack: 5573, Len: 0
    Source Port: 50588
    Destination Port: 443
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 1680    (relative sequence number)
    Sequence number (raw): 1327632916
    [Next sequence number: 1681    (relative sequence number)]
    Acknowledgment number: 5573    (relative ack number)
    Acknowledgment number (raw): 964170541
    0101 .... = Header Length: 20 bytes (5)
  v 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]
    Window size value: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
```

•第二次挥手：服务器接收到之后,发送 ACK(确认)给客户端，表示单项断开连接，而服务器这端数据还没有传输完成。



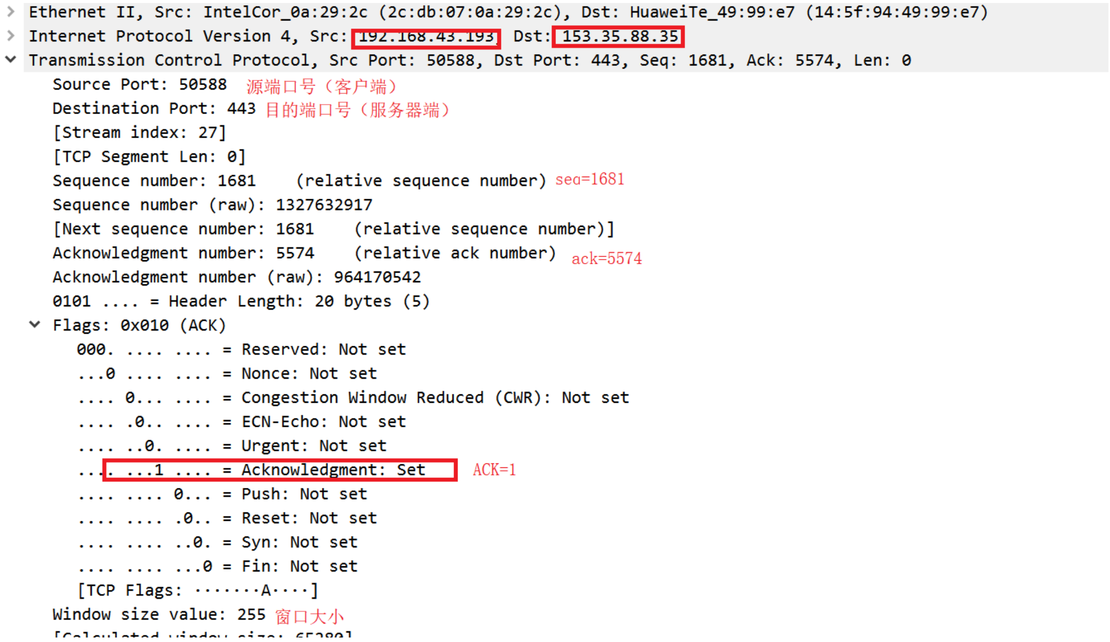
```
Ethernet II, Src: HuaweiTe_49:99:e7 (14:5f:94:49:99:e7), Dst: IntelCor_0a:29:2c (2c:db:07:0a:29:2c)
Internet Protocol Version 4, Src: 153.35.88.35, Dst: 192.168.43.193
Transmission Control Protocol, Src Port: 443, Dst Port: 50588, Seq: 5573, Ack: 1681, Len: 0
    Source Port: 443  源端口号（服务器端）
    Destination Port: 50588  目的端口号（客户端）
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 5573    (relative sequence number)  seq=5573
    Sequence number (raw): 964170541
    [Next sequence number: 5573    (relative sequence number)]
    Acknowledgment number: 1681    (relative ack number)
    Acknowledgment number (raw): 1327632917              ack==1681
    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           ACK=1
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set          FIN=0
        [TCP Flags: ·······A····]
    Window size value: 570  窗口大小
```

•第三次挥手：服务器已经传输完毕，发送 FIN 通知客户端，数据已经传输完毕。



```
Ethernet II, Src: HuaweiTe_49:99:e7 (14:5f:94:49:99:e7), Dst: IntelCor_0a:29:2c (2c:db:07:0a:29:2c)
Internet Protocol Version 4, Src: 153.35.88.35, Dst: 192.168.43.193
Transmission Control Protocol, Src Port: 443, Dst Port: 50588, Seq: 5573, Ack: 1681, Len: 0
    Source Port: 443 源端口号（服务器端）
    Destination Port: 50588  目的端口号（客户端）
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 5573    (relative sequence number)  seq=5573
    Sequence number (raw): 964170541
    [Next sequence number: 5574    (relative sequence number)]
    Acknowledgment number: 1681    (relative ack number) ack=1681
    Acknowledgment number (raw): 1327632917
    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       ACK=1
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...1 = Fin: Set          FIN=1
        [TCP Flags: ·······A···F]
    Window size value: 570 窗口大小
    [Calculated window size: 72960]
```

•第四次挥手：客户端再次发送 ACK,进入 TIME_WAIT 状态；服务器和客户端关闭连接；



```
> Ethernet II, Src: IntelCor_0a:29:2c (2c:db:07:0a:29:2c), Dst: HuaweiTe_49:99:e7 (14:5f:94:49:99:e7)
> Internet Protocol Version 4, Src: 192.168.43.193  Dst: 153.35.88.35
v Transmission Control Protocol, Src Port: 50588, Dst Port: 443, Seq: 1681, Ack: 5574, Len: 0
    Source Port: 50588  源端口号（客户端）
    Destination Port: 443 目的端口号（服务器端）
    [Stream index: 27]
    [TCP Segment Len: 0]
    Sequence number: 1681    (relative sequence number) seq=1681
    Sequence number (raw): 1327632917
    [Next sequence number: 1681    (relative sequence number)]
    Acknowledgment number: 5574    (relative ack number)  ack=5574
    Acknowledgment number (raw): 964170542
    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     ACK=1
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A····]
    Window size value: 255 窗口大小
    [Calculated window size: 65280]
```

3.实验总结

• 掌握了 ftp、nslookup、ipconfig、ping 等网络命令的使用方法 。

•掌握使用 WireShark 抓包的方法，并且可以针对 ICMP、TCP、ARP 等不同协议的数据包进行分析，对三次握手和四次挥手也进行了报文分析，借由实验对其原理有了更好的理解。