

实验二报告

李国楷 2201110101026



实验目的：

学习捕获和分析网络数据包

掌握以太网 MAC 帧、802.11 数据帧和 IPv4 数据包的构成，了解各字段的含义

掌握 ICMP 协议，ping 和 tracert 指令的工作原理

掌握 ARP 协议的请求/响应机理

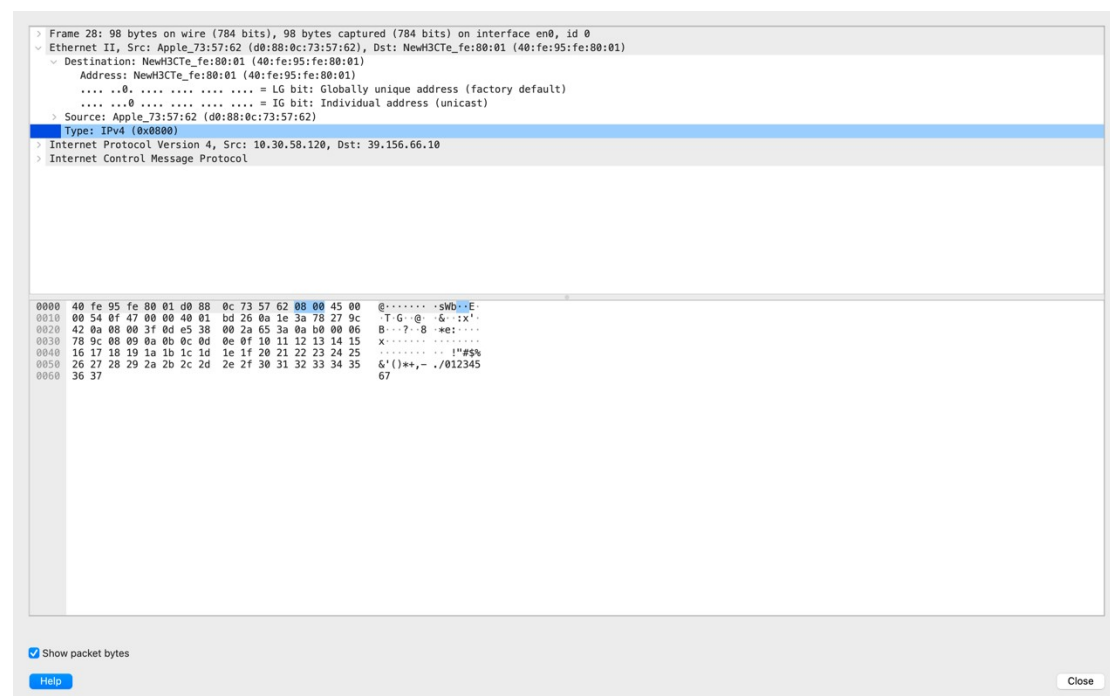
实验内容、结果、分析：

任务 1：捕获和分析有线以太网数据包

Ping www.baidu.com

```
Ken-Lees-MacBook-Air:~ apple$ ping baidu.com
PING baidu.com (39.156.66.10): 56 data bytes
64 bytes from 39.156.66.10: icmp_seq=0 ttl=50 time=58.295 ms
64 bytes from 39.156.66.10: icmp_seq=1 ttl=50 time=55.807 ms
64 bytes from 39.156.66.10: icmp_seq=2 ttl=50 time=54.006 ms
64 bytes from 39.156.66.10: icmp_seq=3 ttl=50 time=57.583 ms
64 bytes from 39.156.66.10: icmp_seq=4 ttl=50 time=57.377 ms
64 bytes from 39.156.66.10: icmp_seq=5 ttl=50 time=56.913 ms
64 bytes from 39.156.66.10: icmp_seq=6 ttl=50 time=63.065 ms
64 bytes from 39.156.66.10: icmp_seq=7 ttl=50 time=64.164 ms
64 bytes from 39.156.66.10: icmp_seq=8 ttl=50 time=64.878 ms
64 bytes from 39.156.66.10: icmp_seq=9 ttl=50 time=57.153 ms
64 bytes from 39.156.66.10: icmp_seq=10 ttl=50 time=57.515 ms
64 bytes from 39.156.66.10: icmp_seq=11 ttl=50 time=66.872 ms
64 bytes from 39.156.66.10: icmp_seq=12 ttl=50 time=59.652 ms
64 bytes from 39.156.66.10: icmp_seq=13 ttl=50 time=62.021 ms
64 bytes from 39.156.66.10: icmp_seq=14 ttl=50 time=56.255 ms
```

1.1 观察 MAC 帧格式



EUI-48 解读：

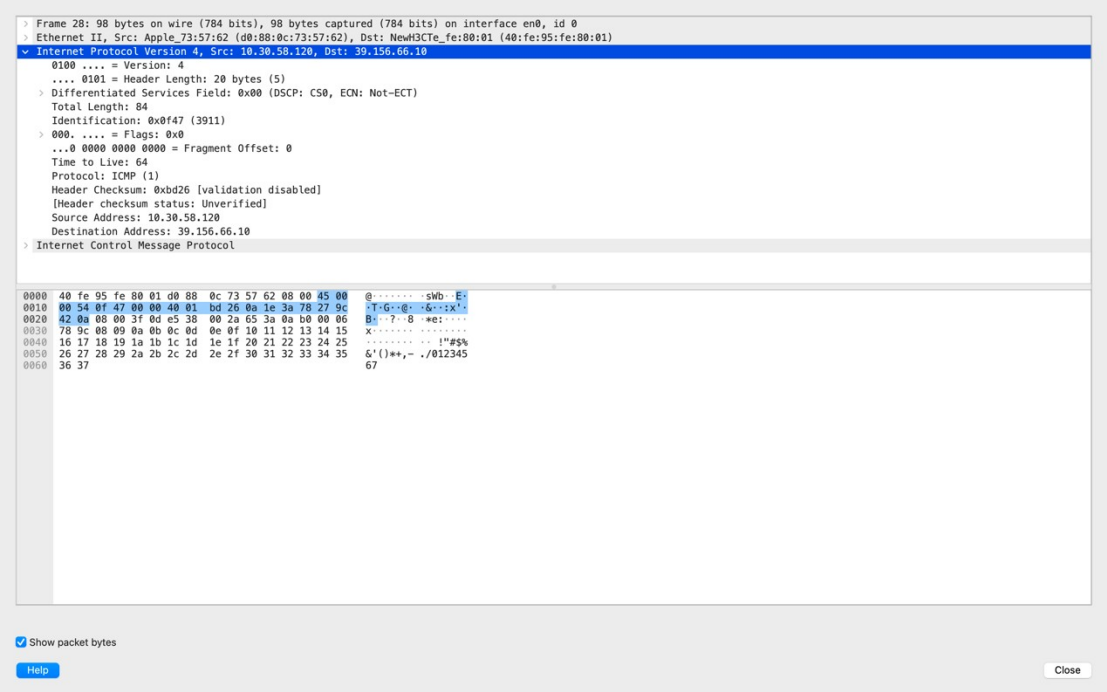
共有 48 位即 6 个字节，其中

前 3 个字节（24 位）：这部分通常用于标识制造商。这三个字节被称为组织唯一标识符。这个部分的前 6 位通常表示制造商标识，后 2 位则用于版本号或其他用途。

后 3 个字节（24 位）：这部分通常由制造商分配给其生产的具体设备。这个部分的前 1 位通常用于指示地址的类型（全球唯一的或本地管理的），后 23 位用于具体的设备识别。

1.2观察 IP 数据报首部结构

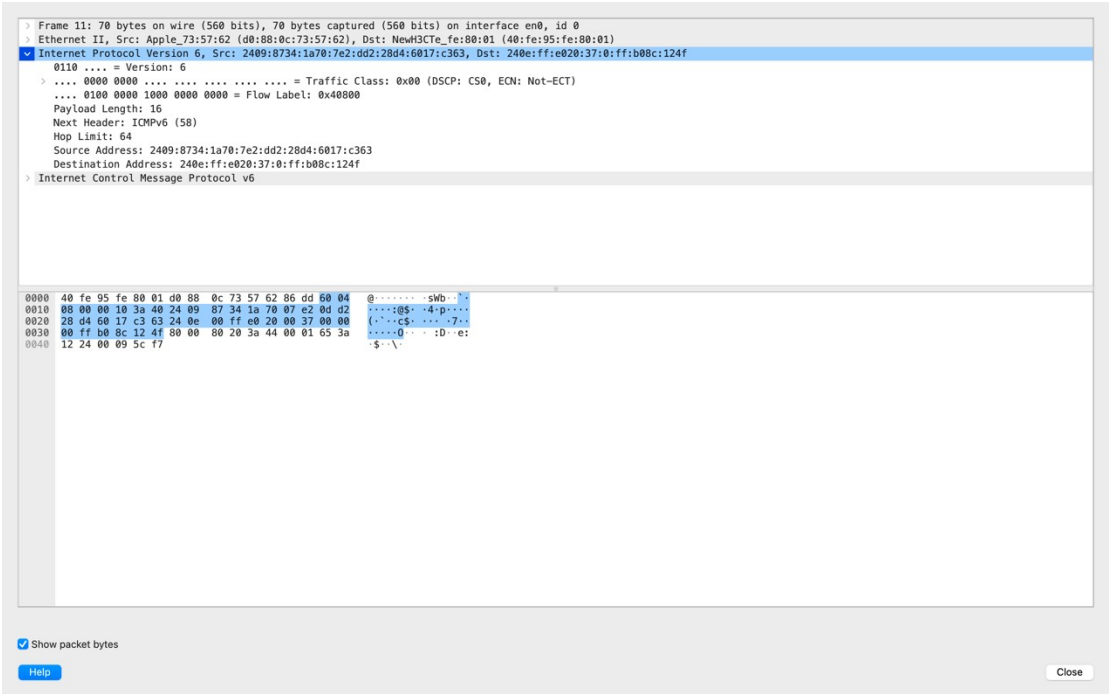
IPV4:



- 1.版本：4 位字段，指示使用的 IPv4 版本号。
- 2.头部长度的：4 位字段，指示 IPv4 首部的长度。IPv4 首部的最小长度为 20 字节，最大长度为 60 字节。
- 3.服务类型：8 位字段，用于指示数据包的服务质量（Quality of Service）和优先级。
- 4.总长度：16 位字段，指示整个 IPv4 数据包的长度，包括首部和数据部分。
- 5.标识：16 位字段，用于标识数据包的唯一性，通常由发送方生成。
- 6.标志：3 位字段，包括 DF（Don't Fragment）、MF（More Fragments）和保留位。DF 位指示该数据包是否可以分片，MF 位指示是否还有更多分片，保留位为将来使用保留。
- 7.片偏移：13 位字段，用于指示分片相对于原始数据报开始位置的偏移量，以 8 字节为单位。
- 8.生存时间：8 位字段，指示数据包在网络中可以传播的最大跳数。每经过一个路由器，该值减 1，当生存时间为 0 时，数据包被丢弃。

- 9.协议：8 位字段，指示数据部分使用的协议（例如，TCP、UDP、ICMP 等）。
- 10.首部校验和：16 位字段，用于检测首部中的错误，确保数据包在传输过程中没有被损坏。
- 11.源 IP 地址：32 位字段，指示数据包的发送者的 IP 地址。
- 12.目标 IP 地址：32 位字段，指示数据包的接收者的 IP 地址。

IPV6:



- 1.版本：4 位字段，表示 IPv6 协议的版本号，固定为 6。
- 2.流量等级：12 位字段，用于指示流量等级
- 3.ENC：用来实现拥塞管理，当网络中出现拥塞时，可以通过设置 ECN 字段来通知发送方减少发送速率。
- 4.流标签：20 位字段，用于在源主机和目标主机之间标识数据流，以便路由器可以提供特殊服务质量。
- 5.有效负载长度：16 位字段，表示 IPv6 数据报文中有效负载（不包括首部）的长度。
- 6.下一首部：8 位字段，表示紧随 IPv6 首部的下一个报文头的类型，类似于 IPv4 的协议字段。
- 7.跳数限制：8 位字段，类似于 IPv4 中的生存时间（Time to Live，TTL），用于限制数据报文在网络中可以传递的最大跳数。

- 8.源 IPv6 地址：128 位字段，表示数据报文的源 IPv6 地址。
- 9.目标 IPv6 地址：128 位字段，表示数据报文的目標 IPv6 地址。
- 10.下一首部：8 位
- 11.首部长度：8 位
- 12 拓展首部：若需要
- 13.数据内容

比较：

相比之下，IPv4 数据报的首部包括版本、头部长度、服务类型、总长度、标识、标志、片偏移、生存时间、协议、首部校验和、源 IPv4 地址和目标 IPv4 地址。IPv4 数据报支持分片，因此有关分片的字段在 IPv4 首部中。IPv4 首部的最小长度为 20 字节。

1.3IP 数据包分片

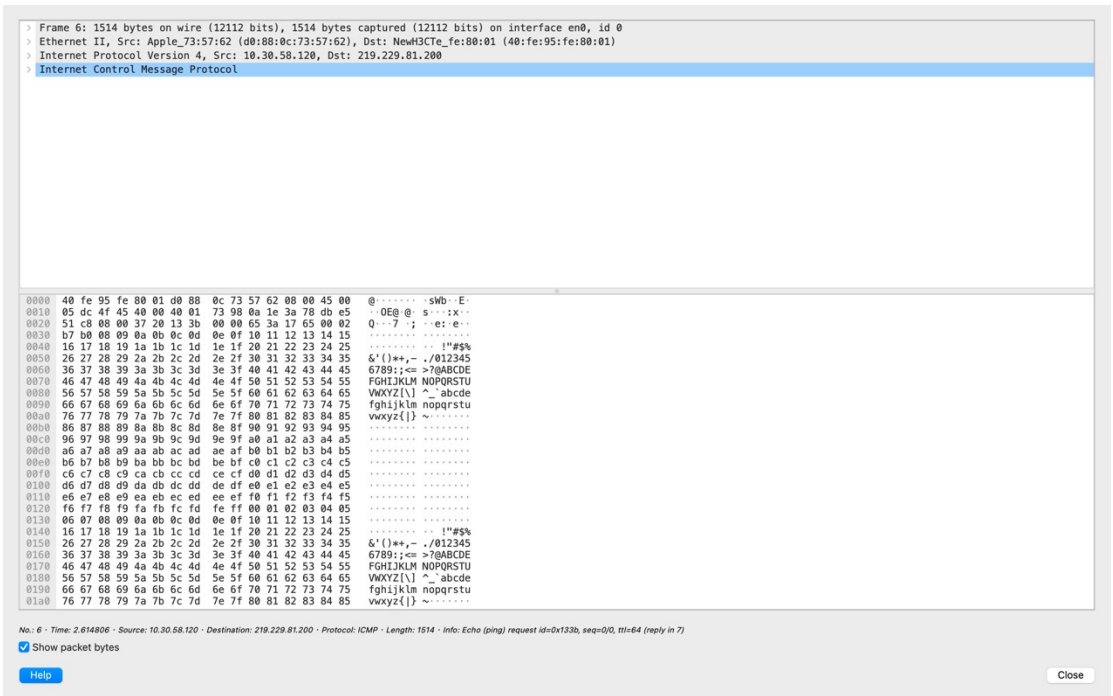
A

```
> Frame 69: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Apple_73:57:62 (d0:88:0c:73:57:62), Dst: NewH3CTe_fe:80:01 (40:fe:95:fe:80:01)
> Internet Protocol Version 4, Src: 10.30.58.120, Dst: 219.229.81.200
> Internet Control Message Protocol
```

| | | | |
|------|-------------------------|-------------------------|-------------------|
| 0000 | 40 fe 95 fe 80 01 d0 88 | 0c 73 57 62 08 00 45 00 | @.....sWb..E. |
| 0010 | 00 54 7f 40 00 00 40 01 | 89 25 0a 1e 3a 78 db e5 | .T.@..@. %.:x.. |
| 0020 | 51 c8 08 00 03 27 f8 3a | 00 15 65 3a 16 77 00 00 | Q.....: ..e:w.. |
| 0030 | 95 d4 08 09 0a 0b 0c 0d | 0e 0f 10 11 12 13 14 15 | |
| 0040 | 16 17 18 19 1a 1b 1c 1d | 1e 1f 20 21 22 23 24 25 | !"#\$\$% |
| 0050 | 26 27 28 29 2a 2b 2c 2d | 2e 2f 30 31 32 33 34 35 | &'()*+,- ./012345 |
| 0060 | 36 37 | | 67 |

这个命令使用 IPv4 地址（不使用 IPv6），向指定的 www.xmu.edu.cn 主机发送 ping 请求。这是一个标准的 IPv4 ping 请求。

B



向 www.xmu.edu.cn 发送一个大小为 1472 字节（不包括 IP 头部的字节）的 ping 请求（通过 -l 1472 设置）。-f 选项表示在 ping 请求中设置“不分段”标志位，确保整个 ping 请求以一个完整的数据包发送。-n 1 表示只发送一个 ping 请求。

C.

```
[Ken-Lees-MacBook-Air:~ apple$ ping -c 1 -s 1473 -D www.xmu.edu.cn
PING csmn1.xmu.edu.cn (219.229.81.200): 1473 data bytes
ping: sendto: Message too long

--- csmn1.xmu.edu.cn ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

同 B，但是因为包太大而没有数据返回

d.

| | Time | Source | Destination | Protocol | Length | Info | |
|---|--|----------------|----------------|----------|--------|--|---|
| → | 127.2.744392 | 10.125.125.47 | 219.229.81.200 | ICMP | 1514 | Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (reply in 129) | |
| → | 128.2.744392 | 10.125.125.47 | 219.229.81.200 | IPv4 | 35 | Fragmented IP protocol (proto=ICMP, 1, off=1480, ID=d2b1) | |
| → | 129.2.748358 | 219.229.81.200 | 10.125.125.47 | ICMP | 1514 | Echo (ping) reply id=0x0001, seq=7/1792, ttl=59 (request in 127) | |
| → | 130.2.749778 | 219.229.81.200 | 10.125.125.47 | IPv4 | 60 | Fragmented IP protocol (proto=ICMP, 1, off=1480, ID=2009) | |
| ↗ | Internet Protocol Version 4, Src: 10.125.125.47, Dst: 219.229.81.200 | | | | | | 00020 51 c8 00 d0 df 3f 00 01 00 07 61 62 63 64 65 66 Q....?..-..abcd |
| | 0100 = Version: 4 | | | | | | 00030 67 68 69 6a 6b 6c 6d 6e 6f 68 69 71 72 73 74 75 fghijklm opqrstu |
| | 0101 = Header Length: 20 bytes (5) | | | | | | 00040 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f wabdefgh hijklmn |
| → | Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) | | | | | | 00050 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 prstuvw abcdefgh |
| | Total Length: 1500 | | | | | | 00060 69 6a 6b 6c 6d 6e 6f 68 69 71 72 73 74 75 76 77 ijklmnop qrstuvw |
| | Identification: 0xd2b1 (53937) | | | | | | 00070 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 68 bcd efghijklmnopq |
| | 001. = Flags: 0x1, More Fragments | | | | | | 00080 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a rstuvwab cdefghij |
| | 000 0000 0000 0000 = Fragment Offset: 0 | | | | | | 00090 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 klmnopqr stuvwabcr |
| | Time to Live: 64 | | | | | | 000a0 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 defghijkl mnopqrst |
| | Protocol: ICMP (1) | | | | | | 000b0 74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c tuvabcd efghijkl |
| | Header Checksum: 0xcd15 [validation disabled] | | | | | | 000c0 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 65 mnopqrst uvwabcde |
| | [Header checksum status: Unverified] | | | | | | 000d0 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 fghijkla nopqrstu |
| | Source Address: 10.125.125.47 | | | | | | 000e0 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e vwabcde fghijklm |
| | Destination Address: 219.229.81.200 | | | | | | 000f0 67 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 opqrstu vwabcde fgh |
| ↗ | Internet Control Message Protocol | | | | | | 01000 68 69 6a 6b 6c 6d 6e 6f 68 69 71 72 73 74 75 76 hijklmno prstuvw |
| | Type: 8 (Echo (ping) request) | | | | | | 01100 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 abcdefgh ijklnmop |
| | Code: 0 | | | | | | 01200 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvw bdefghij |
| | Checksum: 0xdf3f [unverified] [fragmented datagram] | | | | | | 01300 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 jklmnopq rstuvwab |
| | [Checksum Status: Unverified] | | | | | | 01400 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 cdefghij klmnopqr |
| | Identifier (BE): 1 (0x0001) | | | | | | 01500 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b stuvwabc defghijk |
| | Identifier (LE): 256 (0x0100) | | | | | | 01600 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 lmnopqrs tuvwabcd |
| | Sequence Number (BE): 7 (0x0007) | | | | | | 01700 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 efghijkl mnopqrst |
| | Sequence Number (LE): 1792 (0x0700) | | | | | | 01800 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d uvwabcde fghijklm |
| | [Response frame: 129] | | | | | | 01900 67 68 69 71 72 73 74 75 76 77 61 62 63 64 65 66 nopqrstu vwabcde |
| | [Data (1472 bytes)] | | | | | | 01a00 68 69 6a 6b 6c 6d 6 |

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|----------------|----------------|----------|--------|--|
| 127 | 2.744392 | 10.125.125.47 | 219.229.81.200 | ICMP | 1514 | Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (reply in 129) |
| 128 | 2.744392 | 10.125.125.47 | 219.229.81.200 | IPv4 | 35 | Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d2b1) |
| 129 | 2.748358 | 219.229.81.200 | 10.125.125.47 | ICMP | 1514 | Echo (ping) reply id=0x0001, seq=7/1792, ttl=59 (request in 127) |
| 130 | 2.749778 | 219.229.81.200 | 10.125.125.47 | IPv4 | 60 | Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d209) |

> Frame 128: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface 0
Ethernet II, Src: LiteonTe_14:e9:11 (14:5a:fc:14:e9:11), Dst: HuaweiTe_6e:00:00:00:00:00 (08:00:27:00:00:00)
Internet Protocol Version 4, Src: 10.125.125.47, Dst: 219.229.81.200
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 21
Identification: 0xd2b1 (53937)
000. = Flags: 0x0
...0 0000 1011 1001 = Fragment Offset: 1480
Time to Live: 64
Protocol: ICMP (1)
Header Checksum: 0xf223 [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.125.125.47
Destination Address: 219.229.81.200
Data (1 byte)
Data: 61
[Length: 1]

这个命令取消了f，数据可以分段，得到两个报文 IPv4 和 ICMP

1.4 解释 ICMP 报文

| | | | | | | | |
|----|-----------|----------------|----------------|------|----|---------------------|--|
| 32 | 12.856141 | 10.30.58.120 | 219.229.81.200 | ICMP | 98 | Echo (ping) request | id=0xd83b, seq=0/0, ttl=64 (reply in 33) |
| 33 | 12.867355 | 219.229.81.200 | 10.30.58.120 | ICMP | 98 | Echo (ping) reply | id=0xd83b, seq=0/0, ttl=59 (request in 32) |

| | |
|---|--|
| > | Frame 32: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0 |
| > | Ethernet II, Src: Apple_73:57:62 (d0:88:0c:73:57:62), Dst: NewH3CTe_fe:80:01 (40:fe:95:fe:80:01) |
| > | Internet Protocol Version 4, Src: 10.30.58.120, Dst: 219.229.81.200 |
| ▼ | Internet Control Message Protocol |
| | Type: 8 (Echo (ping) request) |
| | Code: 0 |
| | Checksum: 0x52aa [correct] |
| | [Checksum Status: Good] |
| | Identifier (BE): 55355 (0xd83b) |
| | Identifier (LE): 15320 (0x3bd8) |
| | Sequence Number (BE): 0 (0x0000) |
| | Sequence Number (LE): 0 (0x0000) |
| | [Response frame: 33] |
| | Timestamp from icmp data: Oct 26, 2023 16:04:57.352046000 CST |
| | [Timestamp from icmp data (relative): 0.000154000 seconds] |
| > | Data (48 bytes) |
| ▼ | Internet Control Message Protocol |
| | Type: 0 (Echo (ping) reply) |
| | Code: 0 |
| | Checksum: 0x5aaa [correct] |
| | [Checksum Status: Good] |
| | Identifier (BE): 55355 (0xd83b) |
| | Identifier (LE): 15320 (0x3bd8) |
| | Sequence Number (BE): 0 (0x0000) |
| | Sequence Number (LE): 0 (0x0000) |
| | [Request frame: 32] |
| | [Response time: 11.214 ms] |
| | Timestamp from icmp data: Oct 26, 2023 16:04:57.352046000 CST |
| | [Timestamp from icmp data (relative): 0.011368000 seconds] |

Ping 1 次得到一个请求报文和一个回应报文
 差别：请求帧 type=8，回应帧 type=0

1.5tracert 命令

| No. | Time | Source | Destination | Protocol | Length | Info |
|--|----------|--------------|--------------|----------|--------|---|
| 16 | 1.479684 | 10.30.58.120 | 10.30.32.1 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 17) |
| 17 | 1.503759 | 10.30.32.1 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=255 (request in 16) |
| 18 | 1.509842 | 10.30.58.120 | 10.30.32.1 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 19) |
| 19 | 1.532656 | 10.30.32.1 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=255 (request in 18) |
| 20 | 1.535995 | 10.30.58.120 | 10.30.32.1 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 21) |
| 21 | 1.545965 | 10.30.32.1 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=255 (request in 20) |
| 22 | 1.549785 | 10.30.58.120 | 10.30.32.1 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 23) |
| 23 | 1.559147 | 10.30.32.1 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=255 (request in 22) |
| 24 | 1.561358 | 10.30.58.120 | 23.77.60.217 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 25) |
| 25 | 1.625353 | 23.77.60.217 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=49 (request in 24) |
| 26 | 1.628239 | 10.30.58.120 | 23.77.60.217 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 27) |
| 27 | 1.683632 | 23.77.60.217 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=49 (request in 26) |
| 28 | 1.685622 | 10.30.58.120 | 23.77.60.217 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 29) |
| 29 | 1.745623 | 23.77.60.217 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=49 (request in 28) |
| 30 | 1.747613 | 10.30.58.120 | 23.77.60.217 | ICMP | 1382 | Echo (ping) request id=0x052c, seq=0/0, ttl=64 (reply in 31) |
| 31 | 1.804217 | 23.77.60.217 | 10.30.58.120 | ICMP | 1382 | Echo (ping) reply id=0x052c, seq=0/0, ttl=49 (request in 30) |
| 70 | 6.867884 | 10.30.32.1 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 80 | 6.874662 | 10.30.32.1 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 82 | 6.882892 | 10.30.32.1 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 84 | 6.888787 | 172.31.10.41 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 86 | 6.897092 | 172.31.10.37 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 88 | 6.904517 | 172.31.10.37 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 90 | 6.911817 | 210.34.1.57 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| 92 | 6.916040 | 230.34.1.135 | 10.30.58.120 | ICMP | 70 | Time-to-live exceeded (Time to live exceeded in transit) |
| Frame 16: 1382 bytes on wire (11056 bits), 1382 bytes captured (11056 bits) on interface en0, id 0 | | | | | | |
| Ethernet II, Src: Apple_73:57:62 (d0:88:0c:73:57:62), Dst: NewH3CTe_fe:80:01 (40:fe:95:fe:80:01) | | | | | | |
| Internet Protocol Version 4, Src: 10.30.58.120, Dst: 10.30.32.1 | | | | | | |
| Internet Control Message Protocol | | | | | | |
| Type: 8 (Echo (ping) request) | | | | | | |
| Code: 0 | | | | | | |
| Checksum: 0xc98e [correct] | | | | | | |
| [Checksum Status: Good] | | | | | | |
| Identifier (BE): 1324 (0x052c) | | | | | | |
| Identifier (LE): 11269 (0xc205) | | | | | | |
| Sequence Number (BE): 0 (0x0000) | | | | | | |
| Sequence Number (LE): 0 (0x0000) | | | | | | |
| [Response frame: 17] | | | | | | |
| Data (1340 bytes) | | | | | | |


```

[Ken-Lees-MacBook-Air:~ apple$ traceroute xmu.edu.cn
traceroute to xmu.edu.cn (219.229.81.200), 64 hops max, 52 byte packets
 1  10.30.32.1 (10.30.32.1)  15.642 ms  6.000 ms  8.231 ms
 2  172.31.10.41 (172.31.10.41)  5.884 ms
   172.31.10.37 (172.31.10.37)  7.317 ms  6.680 ms
 3  210.34.1.57 (210.34.1.57)  7.288 ms
   210.34.1.125 (210.34.1.125)  6.570 ms
   210.34.1.61 (210.34.1.61)  7.552 ms
 4  210.34.1.218 (210.34.1.218)  7.031 ms  6.155 ms  6.971 ms
 5  210.34.1.54 (210.34.1.54)  6.290 ms  6.721 ms  7.369 ms
 6  219.229.81.200 (219.229.81.200)  8.911 ms !Z  7.608 ms !Z  6.305 ms !Z

```

下面是 Tracert 的工作原理：

发送初始数据包：Tracert 从本地计算机向目标主机发送第一个数据包，通常是一个 ICMP Echo Request 数据包。

第一跳：第一个路由器（第一跳）接收到数据包，它将数据包传递到下一个路由器，然后返回一个 ICMP Echo Reply 数据包，指示它已经接收到数据包。

测量延迟：Tracert 记录了第一跳的响应时间，以便后续分析。

递增 TTL：Tracert 逐一增加数据包的 Time-to-Live (TTL) 或 Hop Limit 字段，然后重新发送数据包。TTL 是数据包在网络中传播的时候逐跳递减的，当 TTL 减为 0 时，路由器会丢弃数据包并向发送方返回 ICMP Time Exceeded 消息。这个递增 TTL 的过程使 Tracert 能够追踪数据包在网络中的跳数。

追踪每一跳：Tracert 持续发送数据包并增加 TTL，每一跳路由器都会接收并传递数据包，然后返回 ICMP Echo Reply 或 Time Exceeded 消息。这样，Tracert 会记录每一跳的 IP 地址、响应时间以及可能的主机名（如果可用）。

终止条件：Tracert 会持续增加 TTL 并收集每一跳的信息，直到它达到指定的跳数限制或直到它到达目标主机。一旦 Tracert 到达目标主机，它会显示目标主机的信息，包括 IP 地址和响应时间。

分析结果：Tracert 会将收集到的每一跳的信息以列表形式显示，供用户分析。用户可以根据 Tracert 的输出来诊断网络问题，了解数据包的路径，找出网络瓶颈等信息

1.6 ARP 协议分析

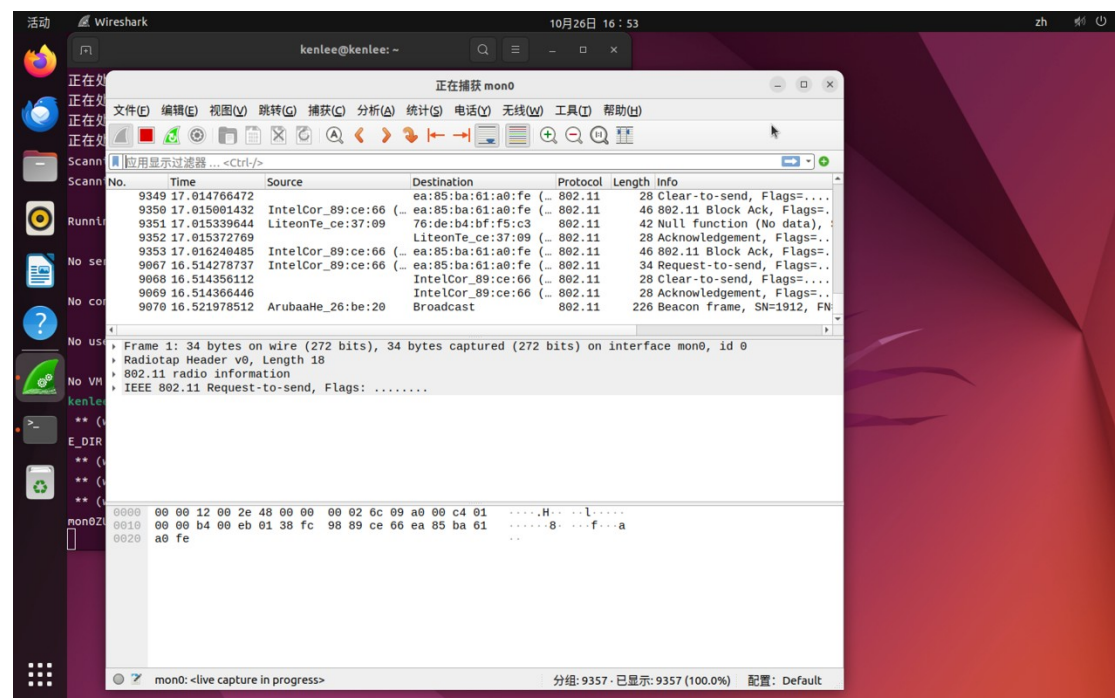
| | | | | | | |
|----|----------|-------------------|-------------------|------|----|---|
| 25 | 4.555665 | LiteonTe_14:e9:11 | Broadcast | ARP | 42 | Who has 192.168.43.122? Tell 192.168.43.122 |
| 26 | 4.745471 | Apple_73:57:62 | LiteonTe_14:e9:11 | ARP | 42 | 192.168.43.181 is at d0:88:0c:73:57:62 |
| 27 | 4.745517 | 192.168.43.122 | 192.168.43.181 | ICMP | 74 | Echo (ping) request id=0x0001, seq=69/17664, ttl=64 (reply in 6.167649s) |
| 28 | 4.752092 | 192.168.43.181 | 192.168.43.122 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=69/17664, ttl=64 (request id=0x0001, seq=70/17920, ttl=64) |
| 29 | 5.561321 | 192.168.43.122 | 192.168.43.181 | ICMP | 74 | Echo (ping) request id=0x0001, seq=70/17920, ttl=64 (reply in 6.167649s) |
| 30 | 5.653867 | 192.168.43.181 | 192.168.43.122 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=70/17920, ttl=64 (request id=0x0001, seq=71/18176, ttl=64) |
| 34 | 6.167649 | 76:87:2c:be:d4:f8 | LiteonTe_14:e9:11 | ARP | 42 | Who has 192.168.43.122? Tell 192.168.43.1 |
| 35 | 6.167677 | LiteonTe_14:e9:11 | 76:87:2c:be:d4:f8 | ARP | 42 | 192.168.43.122 is at 14:5a:fc:14:e9:11 |

局域网内 apr 会询问谁有目标 ip 告诉本机，然后会得到回应，路由器也会向本机询问地址

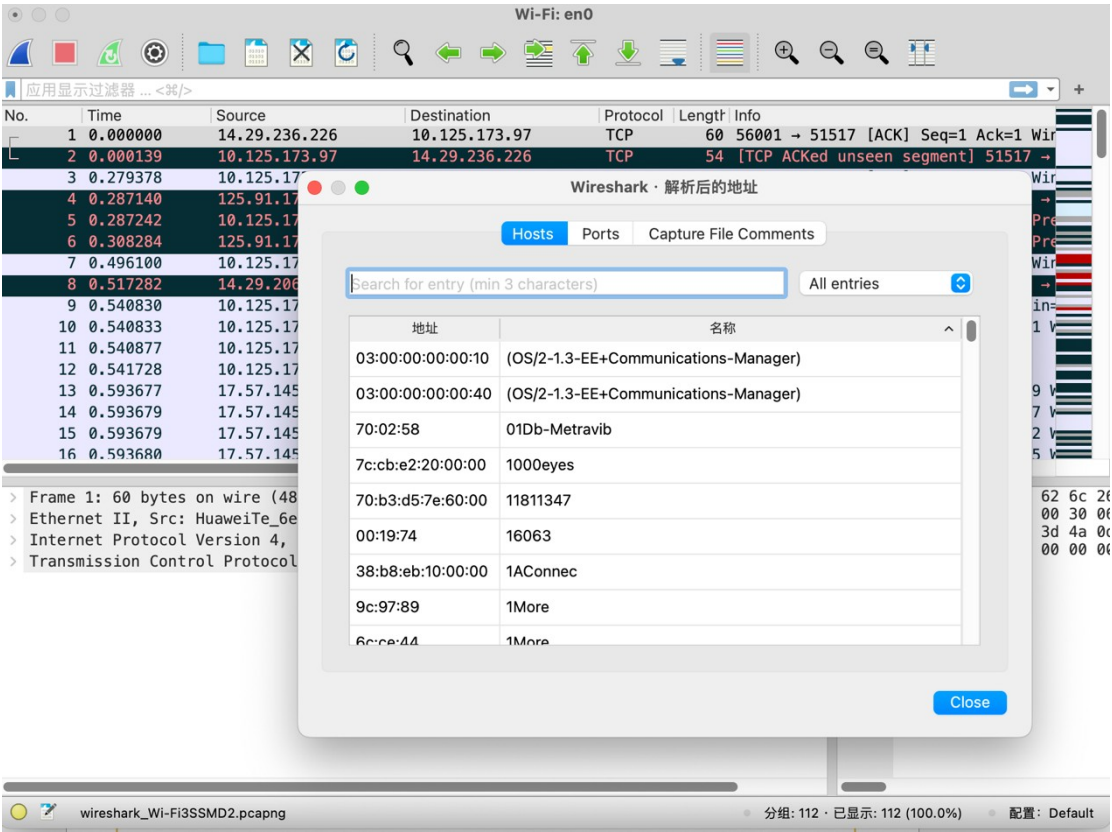
| | | | | | | |
|-----|-----------|-------------------|-------------------|------|----|---|
| 81 | 10.951647 | LiteonTe_14:e9:11 | Broadcast | ARP | 42 | Who has 192.168.43.1? Tell 192.168.43.122 |
| 82 | 10.953475 | 76:87:2c:be:d4:f8 | LiteonTe_14:e9:11 | ARP | 42 | 192.168.43.1 is at 76:87:2c:be:d4:f8 |
| 107 | 17.877191 | 192.168.43.122 | 157.148.69.74 | ICMP | 74 | Echo (ping) request id=0x0001, seq=73/18688, ttl=64 (reply ir |
| 108 | 17.931826 | 157.148.69.74 | 192.168.43.122 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=73/18688, ttl=51 (request |
| 109 | 18.884632 | 192.168.43.122 | 157.148.69.74 | ICMP | 74 | Echo (ping) request id=0x0001, seq=74/18944, ttl=64 (reply ir |
| 110 | 18.964239 | 157.148.69.74 | 192.168.43.122 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=74/18944, ttl=51 (request |
| 111 | 19.888806 | 192.168.43.122 | 157.148.69.74 | ICMP | 74 | Echo (ping) request id=0x0001, seq=75/19200, ttl=64 (reply ir |
| 112 | 19.931329 | 157.148.69.74 | 192.168.43.122 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=75/19200, ttl=51 (request |
| 113 | 20.905429 | 192.168.43.122 | 157.148.69.74 | ICMP | 74 | Echo (ping) request id=0x0001, seq=76/19456, ttl=64 (reply ir |
| 114 | 20.942422 | 157.148.69.74 | 192.168.43.122 | ICMP | 74 | Echo (ping) reply id=0x0001, seq=76/19456, ttl=51 (request |

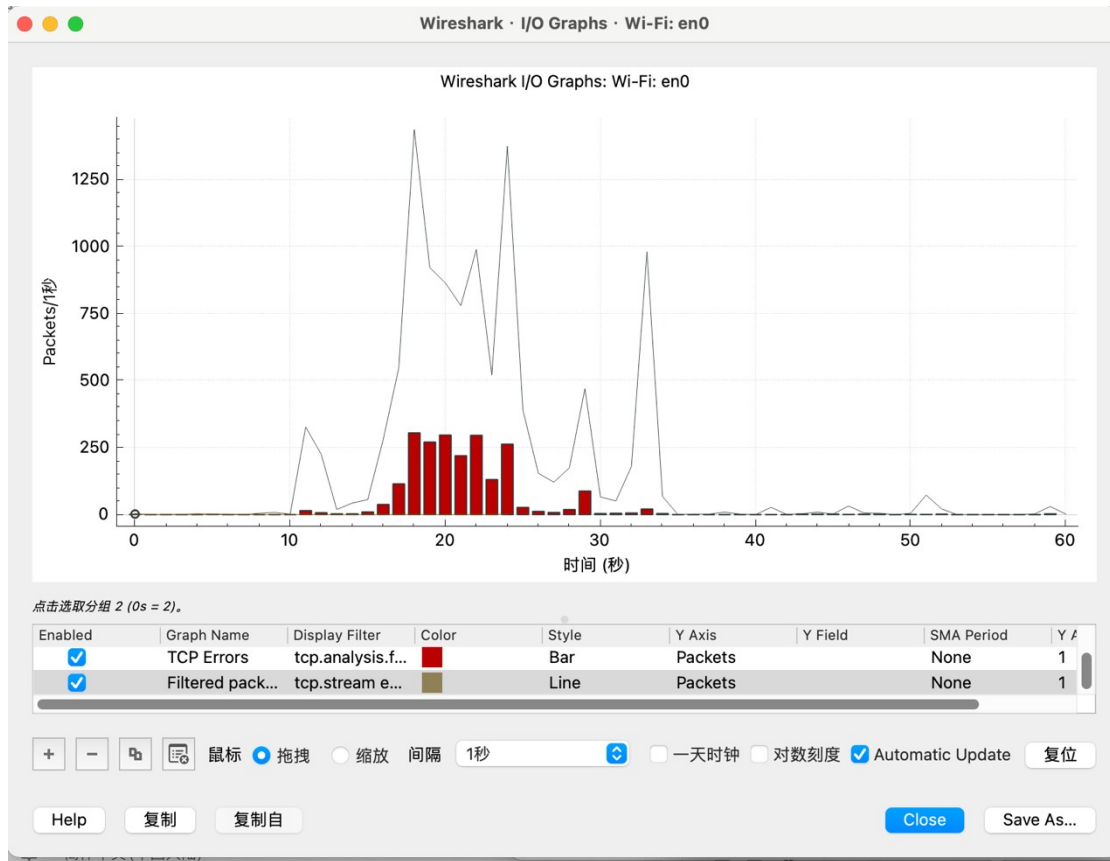
局域网外 apr 会询问谁有路由器 ip 而不是询问目标 ip，得到的是路由器的回应

任务 2：捕获和分析 802.11 数据



3.1 : 探索 Wireshark 更丰富的功能





Wireshark · 协议分级统计 · Wi-Fi: en0

| 协议 | 按分组百分比 | 分组 | 按字节百分比 | 字节 | 比特/秒 |
|-------------------------------|--------|----|--------|-----|------|
| Frame | 100.0 | 5 | 100.0 | 282 | 93 |
| Ethernet | 100.0 | 5 | 29.1 | 82 | 27 |
| Internet Protocol Version 4 | 100.0 | 5 | 35.5 | 100 | 33 |
| Transmission Control Protocol | 100.0 | 5 | 35.5 | 100 | 33 |

显示过滤器: tcp.stream eq 0

Help 复制 Close

Wireshark 还可以绘制 io 图标，解析地址，进行协议分级统计等等

3.2 : 探索其它抓包工具

```
[Ken-Lees-MacBook-Air:~ apple$ sudo tcpdump
tcpdump: data link type PKTAP
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on pktap, link-type PKTAP (Apple DLT_PKTAP), snapshot length 524288 bytes
14:29:03.574347 IP 10.125.173.97.49169 > 157.148.55.96.14000: Flags [P.], seq 570392393:570392510, ack 2684372800, win 4096, length 117
14:29:03.608399 IP 157.148.55.96.14000 > 10.125.173.97.49169: Flags [P.], seq 1:81, ack 117, win 251, length 80
14:29:03.608471 IP 10.125.173.97.49169 > 157.148.55.96.14000: Flags [.], ack 81, win 4094, length 0
14:29:03.636248 IP 10.125.173.97.50403 > 192.168.31.1.domain: 31759+ PTR? 97.173.125.10.in-addr.arpa. (44)
14:29:04.683135 IP 10.125.173.97.50403 > 192.168.31.1.domain: 31759+ PTR? 97.173.125.10.in-addr.arpa. (44)
14:29:05.610693 IP 10.125.173.97.52497 > 64.87.36.59.broad.dg.gd.dynamic.163data.com.cn.56001: Flags [.], ack 1271063307, win 20612, length 0
14:29:05.633790 IP 64.87.36.59.broad.dg.gd.dynamic.163data.com.cn.56001 > 10.125.173.97.52497: Flags [.], ack 1, win 21, length 0
```

Tcpdump 是一个命令行网络抓包工具，它也可以在 macOS 上运行。你可以通过终端使用 Tcpdump 来捕获和分析网络数据包。

这里在苹果终端直接用 `sudo tcpdump` 命令即可进行抓包。

两个软件的区别：Wireshark 有自己 ui 交互界面，用户体验很友好，Tcpdump 因为是命令行工具，需要以命令为基础，使用门槛高，突出专业性。

实验小结、感想：

Wireshark 是一个很强大的软件，通过学习 Wireshark 的使用，使我对 mac 帧，网络协议等有了进一步认识，让我们理解了信息流动的实质，还掌握了各种终端命令的工作原理。

相关代码文档和文件记录

本次课不要求保存文件，切不包含代码工作，截图记录详情见“实验内容、结果、分析”部分。