

Lab1 实验报告

李文彬, 1120173001

2020 年 2 月 27 日

1 实验内容

实验报告中记录了根据实验指导进行实现的操作过程, 展示实验各关键部分的截图, 绘制要求图表, 并在相应阶段回答问题。

1.1 实验环境

Macbook pro, macOS Catalina
Wireshark for Mac

2 实验流程

2.1 Step 1 : Capture a trace

1. 打开 Wireshark, 根据实验指导, 在 Wireshark 中 filter 栏输入 “icmp”
2. 打开 Terminal, 在终端中输入 ping ”www.baidu.com” 进行抓包
3. 按 “control+c” 快捷键停止 Terminal 中的抓包

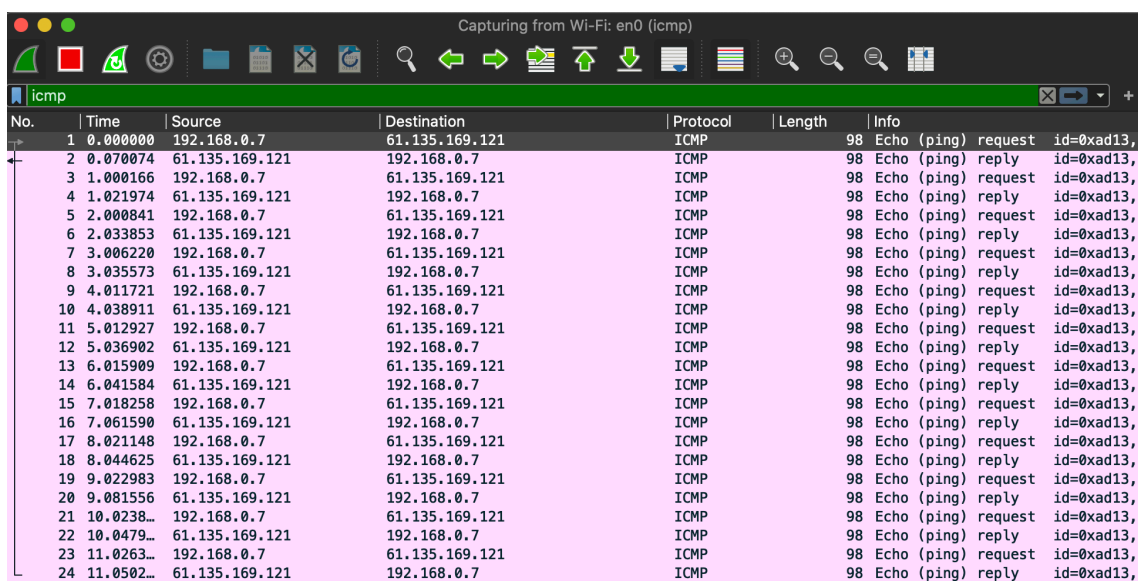
2.1.1 过程分析

实验过程中, 在 Terminal 中运行 ping www.baidu.com 命令, 结果如下

```
1. zsh
Last login: Thu Feb 27 18:20:26 on console
leslie@bogon ~ % ping www.baidu.com
PING www.a.shifen.com (61.135.169.121): 56 data bytes
64 bytes from 61.135.169.121: icmp_seq=0 ttl=55 time=70.199 ms
64 bytes from 61.135.169.121: icmp_seq=1 ttl=55 time=21.949 ms
64 bytes from 61.135.169.121: icmp_seq=2 ttl=55 time=33.178 ms
64 bytes from 61.135.169.121: icmp_seq=3 ttl=55 time=29.611 ms
64 bytes from 61.135.169.121: icmp_seq=4 ttl=55 time=27.341 ms
64 bytes from 61.135.169.121: icmp_seq=5 ttl=55 time=24.111 ms
64 bytes from 61.135.169.121: icmp_seq=6 ttl=55 time=25.854 ms
64 bytes from 61.135.169.121: icmp_seq=7 ttl=55 time=43.503 ms
64 bytes from 61.135.169.121: icmp_seq=8 ttl=55 time=23.619 ms
64 bytes from 61.135.169.121: icmp_seq=9 ttl=55 time=58.713 ms
64 bytes from 61.135.169.121: icmp_seq=10 ttl=55 time=24.229 ms
64 bytes from 61.135.169.121: icmp_seq=11 ttl=55 time=24.002 ms
^C
--- www.a.shifen.com ping statistics ---
12 packets transmitted, 12 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 21.949/33.859/70.199/14.967 ms
leslie@bogon ~ %
```

图 1

通过 wireshark 软件捕获 ping 命令后发送接收的包，结果显示如下：



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
2	0.070074	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
3	1.000166	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
4	1.021974	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
5	2.000841	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
6	2.033853	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
7	3.006220	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
8	3.035573	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
9	4.011721	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
10	4.038911	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
11	5.012927	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
12	5.036902	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
13	6.015909	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
14	6.041584	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
15	7.018258	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
16	7.061590	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
17	8.021148	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
18	8.044625	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
19	9.022983	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
20	9.081556	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
21	10.0238...	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
22	10.0479...	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,
23	11.0263...	192.168.0.7	61.135.169.121	ICMP	98	Echo (ping) request id=0xad13,
24	11.0502...	61.135.169.121	192.168.0.7	ICMP	98	Echo (ping) reply id=0xad13,

图 2

2.2 Step 2 : Ethernet Frame Structure

随机选取图 2 中的一帧，得到 wireshark 的自动进行解析，显示如下：

```
▶ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
▼ Ethernet II, Src: Apple_9f:52:7b (78:4f:43:9f:52:7b), Dst: ZioncomE_20:47:a0 (f4:28:53:20:47:a0)
  ▶ Destination: ZioncomE_20:47:a0 (f4:28:53:20:47:a0)
  ▶ Source: Apple_9f:52:7b (78:4f:43:9f:52:7b)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.0.7, Dst: 61.135.169.121
▶ Internet Control Message Protocol

0000 f4 28 53 20 47 a0 78 4f 43 9f 52 7b 08 00 45 00  (S G·x0 C·R{··E·
0010 00 54 eb fb 00 00 40 01 e6 fd c0 a8 00 07 3d 87  T····@· ····=-
0020 a9 79 08 00 ca e5 ad 13 00 00 5e 57 d2 02 00 00  y·····^W····
0030 64 a9 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  d····· ····
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
```

图 3

下部显示的信息共 0x62 字节，与描述 98bytes 匹配（未计算尾部 checksum）。图 3 中选中展开 Ethernet 部分，Wireshark 自动显示其 header 占 14 字节。

2.2.1 Results

根据描述信息可知，实验中采用的是 Ethernet II 协议，由 header、data 和 checksum 组成，Ethernet II 协议的帧结构如下图：

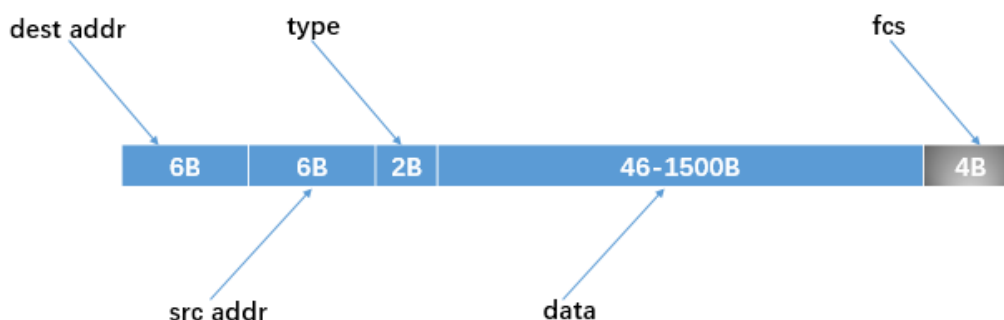


图 4

2.3 Step 3 : Scope of Ethernet Addresses

```
▶ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
▼ Ethernet II, Src: Apple_9f:52:7b (78:4f:43:9f:52:7b), Dst: ZioncomE_20:47:a0 (f4:28:53:20:47:a0)
  ▶ Destination: ZioncomE_20:47:a0 (f4:28:53:20:47:a0)
  ▶ Source: Apple_9f:52:7b (78:4f:43:9f:52:7b)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.0.7, Dst: 61.135.169.121
▶ Internet Control Message Protocol
```

图 5

以上图的 Ethernet frame 为例, src 与 dest 的物理地址分别为 (78:4f:43:9f:52:7b) (18:31:bf:4a:be:80), 本机 ip 为 (192.168.0.7), remote server 的 ip 为 (61.135.169.121)。因此 my computer, router 和 remote server 的相对位置如图所示:

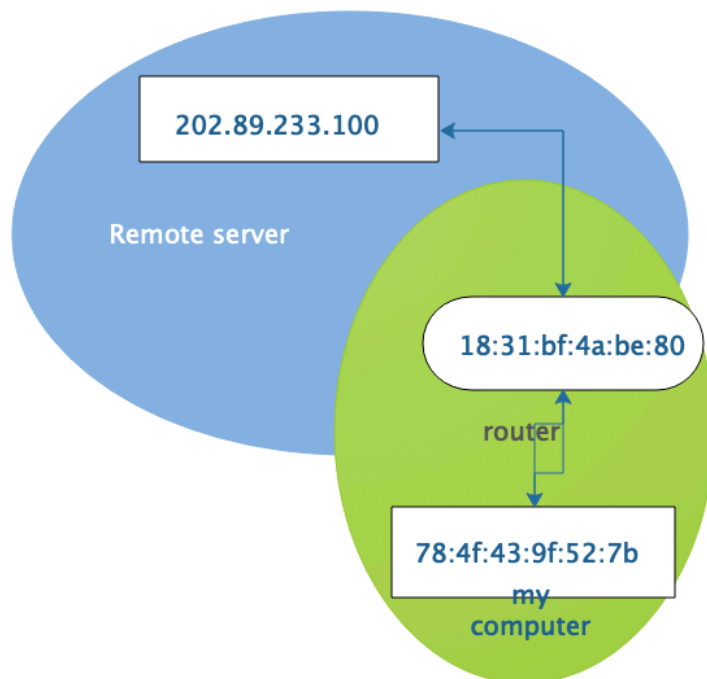


图 6

3 Broadcast Frames

通过如下所示 filter, 只显示以太网的多播 multicast 或广播 broadcast 消息。

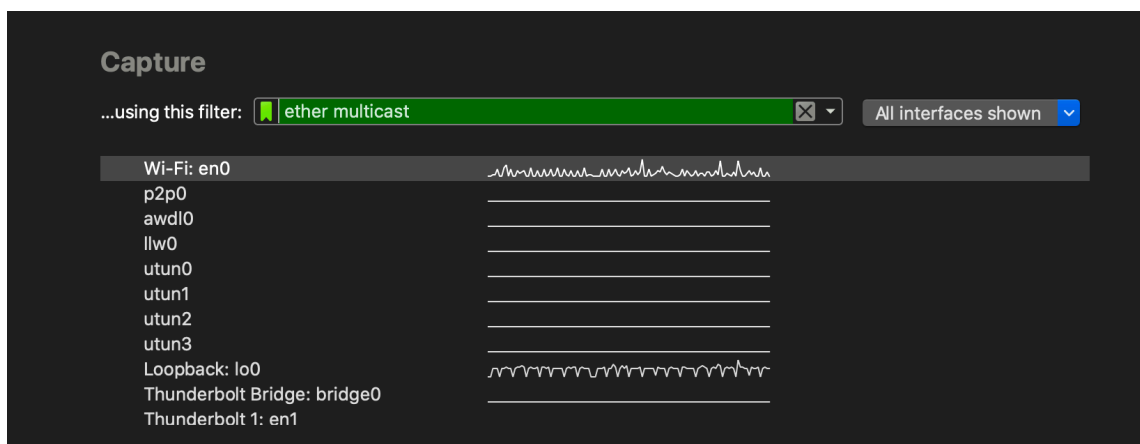


图 7

以下是截获到的 broadcast 和 multicast 样例。

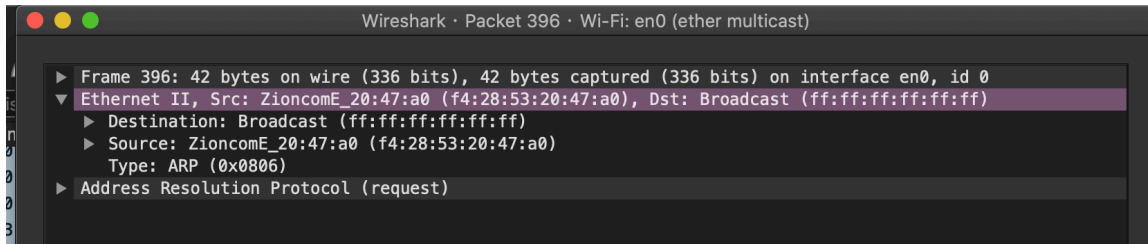


图 8

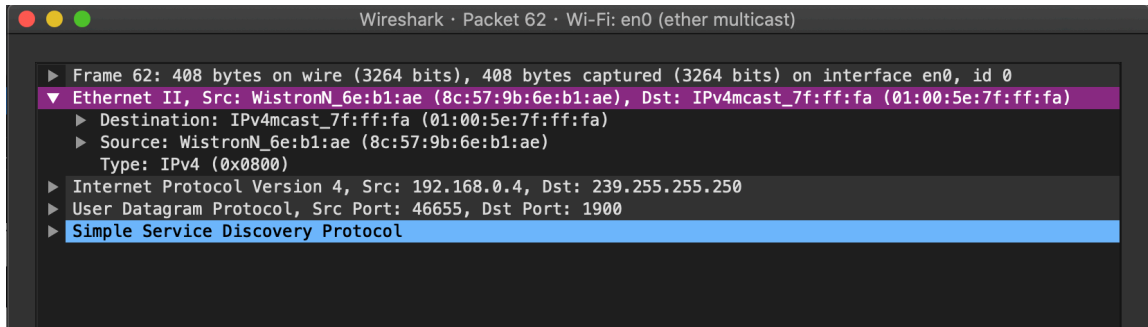


图 9

4 Answering the following question

- i What is the broadcast Ethernet address, written in standard form as Wireshark displays it?

图 8 可见，broadcast 的目的地址为 (ff:ff:ff:ff:ff:ff)，即 48bit 全为 1。

- ii whether it is unicast or multicast/broadcast?

分析 multicast 的目的物理地址段，wireshark 软件显示 Group address 标志位为第 8 位，如下图所示，其地址的第一字节为 01H，标识位为 1 表示多播。

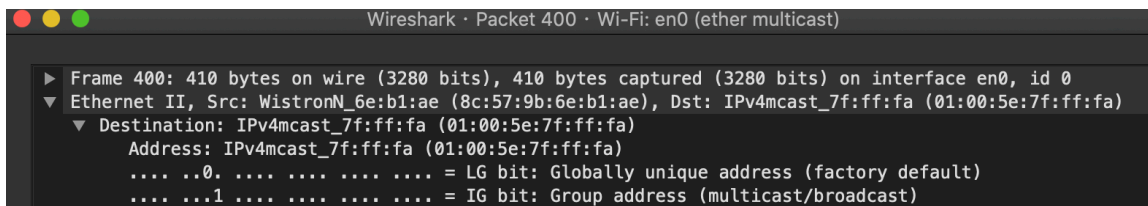


图 10