

计算机网络第七次实验实验报告

舒文炫

2021 年 11 月 29 日

目录

1	实验内容	2
2	实验过程	3
2.1	1. Capturing and analyzing Ethernet frames	3
2.1.1	过程截图	3
2.1.2	小节思考题	4
2.2	2. The Address Resolution Protocol	6
2.2.1	ARP Caching	6
2.2.2	Observing ARP in action	7
2.2.3	小节思考题	8
2.3	Extra Credit	11

Chapter 1

实验内容

本次实验内容是以太网协议以及 ARP 协议，与前面实验相同，将会了解这些协议的报文格式

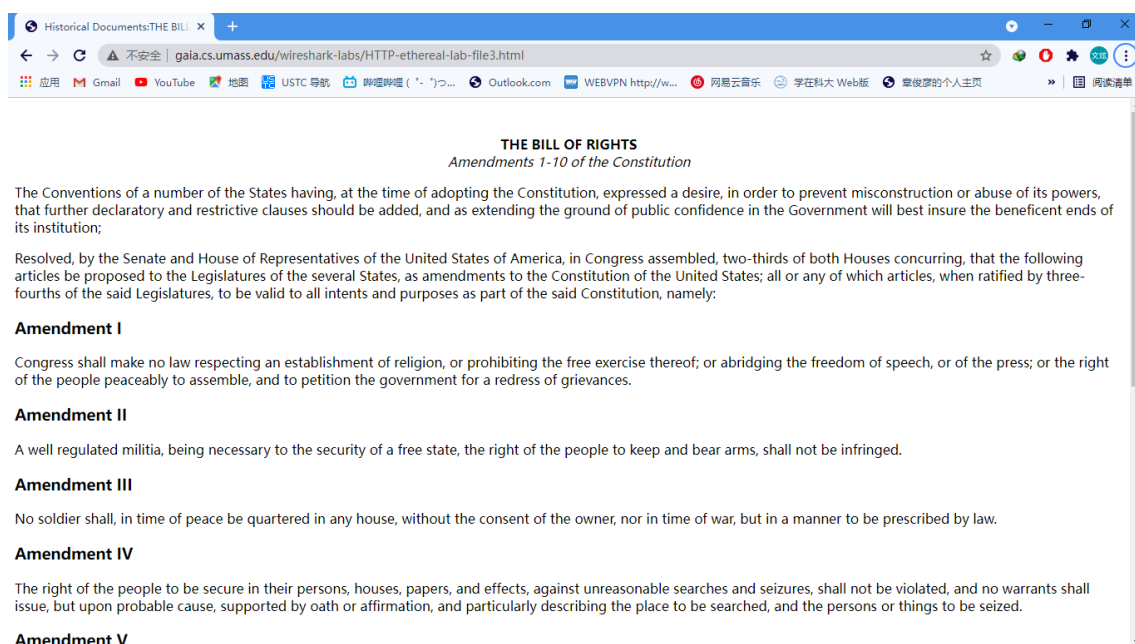
Chapter 2

实验过程

2.1 1. Capturing and analyzing Ethernet frames

2.1.1 过程截图

先清空浏览器的缓存,打开 wireshark,将网址 <http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html> 输入浏览器,将打开如下页面



这时停止 wireshark 捕获,要找到 HTTP 的 get 报文,这里我直接设置过滤器为 http,结果如下

No.	Time	Source	Destination	Protocol	Length	Info
1164	23.492802	192.168.43.193	128.119.245.12	HTTP	532	GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1
1174	23.755331	128.119.245.12	192.168.43.193	HTTP	835	HTTP/1.1 200 OK (text/html)
1187	23.876980	192.168.43.193	128.119.245.12	HTTP	478	GET /favicon.ico HTTP/1.1
1194	24.132997	128.119.245.12	192.168.43.193	HTTP	538	HTTP/1.1 404 Not Found (text/html)
1223	34.501730	192.168.43.193	54.222.224.191	HTTP	416	GET /message/updateTime?tags=commontags&updateTime=14488897091e6991483ef1056c7 HTTP/1.1
1225	34.559084	54.222.224.191	192.168.43.193	HTTP	227	HTTP/1.1 302
1230	34.590516	192.168.43.193	61.160.228.202	HTTP	368	GET /sprint/settings/updateTime/commontags HTTP/1.1
1232	34.630028	61.160.228.202	192.168.43.193	HTTP	975	HTTP/1.1 200 OK
1234	34.630176	61.160.228.202	192.168.43.193	HTTP	536	HTTP/1.1 400 Bad Request (text/html)

这里第 1164 个包是 HTTP GET 第 1174 个包是 HTTP response,但是实际上这时相应报文的最后一个报文,前面的报文是标记为 TCP 的,这在 TCP 实验里面见过。

然后将 ip 协议去掉,因为这里我们只需要看以太网协议,去掉之后得到下面的截图

No.	Time	Source	Destination	Protocol	Length	Info
1159	23.491479	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	66	IPv4
1160	23.491479	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	66	IPv4
1161	23.491840	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	54	IPv4
1162	23.491977	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	54	IPv4
1163	23.492107	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1164	23.492802	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	532	IPv4
1165	23.642166	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1166	23.681431	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x86dd	395	IPv6
1167	23.682653	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x86dd	917	IPv6
1168	23.754942	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	66	IPv4
1169	23.754942	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	54	IPv4
1170	23.755055	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	54	IPv4
1171	23.755130	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	1414	IPv4

> Frame 1164: 532 bytes on wire (4256 bits), 532 bytes captured (4256 bits) on interface \Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0

▼ Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

 Destination: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

 Address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

 ...1. = LG bit: Locally administered address (this is NOT the factory default)

 ...0. = IG bit: Individual address (unicast)

 Source: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)

 Address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)

 ...0. = LG bit: Globally unique address (factory default)

 ...0. = IG bit: Individual address (unicast)

 Type: IPv4 (0x0800)

> Data (518 bytes)

2.1.2 小节思考题

这里我们需要的是对应包含 HTTP GET 的报文，我将其打印出来，截图如下

No.	Time	Source	Destination	Protocol	Length	Info
1164	23.492802	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	532	IPv4

Frame 1164: 532 bytes on wire (4256 bits), 532 bytes captured (4256 bits) on interface \Device\NPF_{D7A10DBD-93D5-4F5B-95D5-A4C7F7828D05}, id 0

Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

 Destination: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

 Address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

 ...1. = LG bit: Locally administered address (this is NOT the factory default)

 ...0. = IG bit: Individual address (unicast)

 Source: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)

 Address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)

 ...0. = LG bit: Globally unique address (factory default)

 ...0. = IG bit: Individual address (unicast)

 Type: IPv4 (0x0800)

Data (518 bytes)

```

0000 45 00 02 06 ea f3 40 00 40 06 ec 10 c0 a8 2b c1 E....@.....+.
0010 80 7f f5 0c ec 5e 00 50 6b f5 a0 58 0b d7 9d bf .w...^..Pk..X...
0020 50 18 02 03 6e 05 00 00 47 45 54 20 2f 77 69 72 P...n...GET /wir
0030 65 73 68 61 72 6b 2d 6c 61 62 73 2f 48 54 54 50 eshark-labs/HTTP
0040 2d 65 74 68 65 72 65 61 6c 2d 6c 61 62 2d 66 69 -ethereal-lab-fi
0050 6c 65 33 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e le3.html HTTP/1.
0060 31 0d 0a 48 6f 73 74 3a 20 67 61 69 61 2e 63 73 1..Host: gaia.cs
0070 2e 75 6d 61 73 73 2e 65 64 75 0d 0a 43 6f 6e 6e .umass.edu..Conn
0080 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 ection: keep-ali
0090 76 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 ve..Upgrade-Inse
00a0 63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 cure-Requests: 1

```

1. What is the 48-bit Ethernet address of your computer?

截图里面有一行,Source: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5),表示我的电脑的以太网地址是 30:d1:6b:97:9f:c5

2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? (Hint: the answer is no). What device has this as its Ethernet address? [Note: this is an important question, and one that students sometimes get wrong. Re-read pages 468-469 in the text and make sure you understand the answer here.]

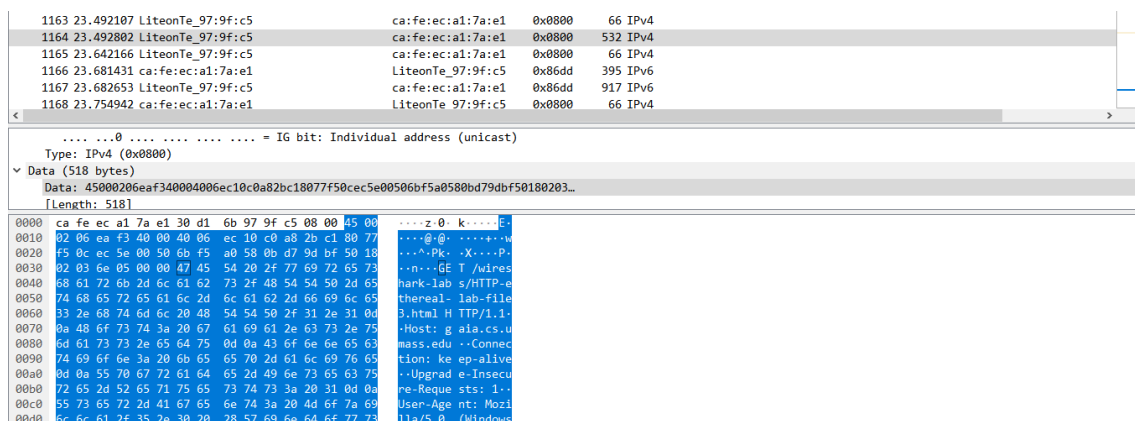
截图里面有一行 Destination: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1), 表示目的地址为 ca:fe:ec:a1:7a:e1, 这个不是 gaia.cs.umass.edu 的地址, 实际上这是通往目的地路径上第一跳路由器接口对应的适配器地址, 如果真是 gaia.cs.umass.edu 对应的主机的地址, 在这个子网上无法匹配该目的地址, 那就没办法发送出去。

3. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

截图里面有一行 Type: IPv4 (0x0800), 值是 0x0800, 表示这个上层协议是 ipv4

4. How many bytes from the very start of the Ethernet frame does the ASCII “G” in “GET” appear in the Ethernet frame?

我在 data 里面找到了这个 GET，截图如下



选择了 G 这个字母，找到其位置，是第 55 个字节

下面需要 HTTP response 报文，我将其打印出来截图如下



5. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu (Hint: the answer is no). What device has this as its Ethernet address?

截图里面有一行，Source: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)，表示源地址为 ca:fe:ec:a1:7a:e1，这既不是我电脑的地址也不是 gaia.cs.umass.edu 的地址，实际上这是对于我的电脑第一跳路由器接口的地址。

6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

截图里面有一行，Destination: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)，表示目的地址为 30:d1:6b:97:9f:c5，这是我电脑的以太网地址

7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

截图里面有一行 Type: IPv4 (0x0800)，表示其值为 0x0800，表示上层协议为 ipv4

8. How many bytes from the very start of the Ethernet frame does the ASCII “O” in “OK” (i.e., the HTTP response code) appear in the Ethernet frame?

在 data 里面找到了 OK，选择了 O，截图如下

No.	Time	Source	Destination	Protocol	Length	Info
1171	23.755139	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	1414	IPv4
1172	23.755331	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	1414	IPv4
1173	23.755331	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	1414	IPv4
1174	23.755331	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	835	IPv4
1175	23.755401	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	54	IPv4
1176	23.802733	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	197	IPv4
1177	23.803103	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	1414	IPv4

Type: IPv4 (0x0800)
▼ Data (1400 bytes)
Data: 450005787e2c400029066c668077f50cc0a82bc10050ec5e0bd79dbf6bf5a236501000ed...
[Length: 1400]

0000	30 d1 6b 97 9f c5 ca fe ec a1 7a e1 08 00 45 00	0-k-----z---E-
0010	05 78 7e 2c 40 00 29 06 6c 66 80 77 f5 0c c0 a8	·x~.@·)·lf·w····
0020	2b c1 00 50 ec 5e 0b d7 9d bf 6b f5 a2 36 50 10	+..P.^...·k..6P·
0030	00 ed 32 67 00 00 48 54 54 50 2f 31 2e 31 20 32	..2g..HT TP/1.1 2
0040	30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 53 75 6e	00 OK..D ate: Sun
0050	2c 20 32 38 20 4e 6f 76 20 32 30 32 31 20 30 39	, 28 Nov 2021 09
0060	3a 30 38 3a 33 37 20 47 4d 54 0d 0a 53 65 72 76	:08:37 G MT..Serv
0070	65 72 3a 20 41 70 61 63 68 65 2f 32 2e 34 2e 36	er: Apac he/2.4.6
0080	20 28 43 65 6e 74 4f 53 29 20 4f 70 65 6e 53 53	(CentOS) OpenSS
0090	4c 2f 31 2e 30 2e 32 6b 2d 66 69 70 73 20 50 48	L/1.0.2k -fips PH
00a0	50 2f 37 2e 34 2e 32 35 20 6d 6f 64 5f 70 65 72	P/7.4.25 mod_per
00b0	6c 2f 32 2e 30 2e 31 31 20 50 65 72 6c 2f 76 35	l/2.0.11 Perl/v5
00c0	2e 31 36 2e 33 0d 0a 4c 61 73 74 2d 4d 6f 64 69	.16.3..L ast-Modi

可以看到这里对应的是第 68 个字节

2.2 2. The Address Resolution Protocol

2.2.1 ARP Caching

过程截图

windows 上由 arp 命令可以进行一些 arp 的基本操作，比如表项的查看，增加，删除之类的
运行 arp -a 得到了我这里缓存了一系列 arp 表项

```
C:\Users\SWX>arp -a

接口: 192.168.11.1 --- 0x4
Internet 地址      物理地址      类型
192.168.11.254     00-50-56-fa-d9-ca      动态
192.168.11.255     ff-ff-ff-ff-ff-ff      静态
224.0.0.22         01-00-5e-00-00-16      静态
224.0.0.251        01-00-5e-00-00-fb      静态
224.0.0.252        01-00-5e-00-00-fc      静态
239.255.255.250    01-00-5e-7f-ff-fa      静态
255.255.255.255    ff-ff-ff-ff-ff-ff      静态

接口: 192.168.43.193 --- 0x11
Internet 地址      物理地址      类型
192.168.43.1       ca-fe-ec-a1-7a-e1      动态
192.168.43.255     ff-ff-ff-ff-ff-ff      静态
224.0.0.22         01-00-5e-00-00-16      静态
224.0.0.251        01-00-5e-00-00-fb      静态
224.0.0.252        01-00-5e-00-00-fc      静态
239.255.255.250    01-00-5e-7f-ff-fa      静态
255.255.255.255    ff-ff-ff-ff-ff-ff      静态

接口: 192.168.40.1 --- 0x14
Internet 地址      物理地址      类型
192.168.40.255     ff-ff-ff-ff-ff-ff      静态
224.0.0.22         01-00-5e-00-00-16      静态
224.0.0.251        01-00-5e-00-00-fb      静态
224.0.0.252        01-00-5e-00-00-fc      静态
239.255.255.250    01-00-5e-7f-ff-fa      静态
```

但是运行 arp -d 命令，这个时候需要有管理员权限，下面是运行 arp -d * 命令的结果

```

C:\windows\system32>arp -d *
C:\windows\system32>arp -a

接口: 192.168.11.1 --- 0x4
Internet 地址      物理地址      类型
224.0.0.22        01-00-5e-00-00-16 静态

接口: 192.168.43.193 --- 0x11
Internet 地址      物理地址      类型
192.168.43.1       ca-fe-ec-a1-7a-e1 动态
224.0.0.22         01-00-5e-00-00-16 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

接口: 192.168.40.1 --- 0x14
Internet 地址      物理地址      类型
224.0.0.22        01-00-5e-00-00-16 静态

```

小节思考题

9. Write down the contents of your computer's ARP cache. What is the meaning of each column value?

上面的截图就是我电脑里面的 arp 缓存，这里第一列为 IP 地址，第二列为其对应的 mac 地址，第三列为类型，分为静态和动态

2.2.2 Observing ARP in action

过程截图

这里首先要清空 arp 缓存，使用 `arp -d *` 命令，然后清空浏览器缓存，打开 wireshark 开始捕获，在浏览器输入与之前相同的网址，打开页面后，停止捕获，这时我得到了下面的结果

No.	Time	Source	Destination	Protocol	Length	Info
1005	6.809450	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	66	IPv4
1006	6.809647	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	54	IPv4
1007	6.810357	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	571	IPv4
1008	6.848097	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	0x0800	54	IPv4
1009	7.244228	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1010	7.518078	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1011	7.533108	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1012	8.144345	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	ARP	42	Who has 192.168.43.193? Tell 192.168.43.1
1013	8.144508	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	ARP	42	192.168.43.193 is at 38:d1:6b:97:9f:c5
1014	8.664247	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1015	8.664839	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1016	8.684165	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1017	8.684945	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1018	8.793672	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4
1019	8.894441	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	0x0800	66	IPv4

我将需要的 ARP 请求和响应报文打印出来，截图如下，我这里和实验指导书上面的有些许不同，没有看到广播地址，按照上面流程重试了很多次都是这样，不过不影响做题

这是请求报文

```

No.    Time          Source           Destination      Protocol Length Info
1012   8.144345      ca:fe:ec:a1:7a:e1 LiteonTe_97:9f:c5 ARP              42      Who has 192.168.43.193? Tell 192.168.43.1
Frame 1012: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
\Device\NPF_{D7A180B0-93D5-4F5B-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1), Dst: LiteonTe_97:9f:c5 (38:d1:6b:97:9f:c5)
Destination: LiteonTe_97:9f:c5 (38:d1:6b:97:9f:c5)
...0. .... = LG bit: Globally unique address (factory default)
...0. .... = IG bit: Individual address (unicast)
Source: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
...1. .... = LG bit: Locally administered address (this is NOT the factory default)
...0. .... = IG bit: Individual address (unicast)
Type: ARP (0x0806)
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Sender IP address: 192.168.43.1
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.43.193

```

这是响应报文


```

No.      Time      Source      Destination  Protocol Length Info
1013 8.144360 LiteonTe_97:9f:c5 ca:fe:ec:a1:7a:e1 ARP 42 192.168.43.193 is at 30:d1:6b:
97:9f:c5
Frame 1013: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
\Device\NPF_{D7A1008D-93D5-4F58-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Destination: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
.... 01. .... = LG bit: Locally administered address (this is NOT the factory default)
.... 0 .... = IG bit: Individual address (unicast)
Source: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
.... 0 .... = LG bit: Globally unique address (factory default)
.... 0 .... = IG bit: Individual address (unicast)
Type: ARP (0x0806)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Sender IP address: 192.168.43.193
Target MAC address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Target IP address: 192.168.43.1

```

2.2.3 小节思考题

10. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?

在包含请求报文的帧里面源地址 Source: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)，目的地址 Destination: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)

11. Give the hexadecimal value for the two-byte Ethernet Frame type field. What upper layer protocol does this correspond to?

请求报文截图了有一行 Type: ARP (0x0806)，表示上层协议是 ARP

12. Download the ARP specification from <ftp://ftp.rfc-editor.org/in-notes/std/std37.txt>. A readable, detailed discussion of ARP is also at <http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html>. 下载的文件如下

```

std37.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Network Working Group          David C. Plummer
Request For Comments: 826      (DCP@MIT-MC)
                               November 1982

An Ethernet Address Resolution Protocol
-- OF --
Converting Network Protocol Addresses
to 48.bit Ethernet Address
for Transmission on
Ethernet Hardware

Abstract

The implementation of protocol P on a sending host S decides,
through protocol P's routing mechanism, that it wants to transmit
to a target host T located some place on a connected piece of
10Mbit Ethernet cable. To actually transmit the Ethernet packet
a 48.bit Ethernet address must be generated. The addresses of
hosts within protocol P are not always compatible with the
corresponding Ethernet address (being different lengths or
values). Presented here is a protocol that allows dynamic
distribution of the information needed to build tables to
translate an address A in protocol P's address space into a
48.bit Ethernet address.

Generalizations have been made which allow the protocol to be
used for non-10Mbit Ethernet hardware. Some packet radio
networks are examples of such hardware.

```

或者可以打开网址如下

Address Resolution Protocol (arp)

The address resolution protocol (arp) is a protocol used by the [Internet Protocol \(IP\)](#) [RFC826], specifically IPv4, to map [IP network addresses](#) to the hardware addresses used by a data link protocol. The protocol operates below the network layer as a part of the interface between the OSI network and OSI link layer. It is used when [IPv4 is used over Ethernet](#).

The term address resolution refers to the process of finding an address of a computer in a network. The address is "resolved" using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer. The information received by the server allows the server to uniquely identify the network system for which the address was required and therefore to provide the required address. The address resolution procedure is completed when the client receives a response from the server containing the required address.

An [Ethernet](#) network uses two hardware addresses which identify the source and destination of each frame sent by the [Ethernet](#). The destination address (all 1's) may also identify a [broadcast](#) packet (to be sent to all connected computers). The hardware address is also known as the [Medium Access Control \(MAC\) address](#), in reference to the standards which define [Ethernet](#). Each computer [network interface card](#) is allocated a globally unique 6 byte link address when the factory manufactures the card (stored in a PROM). This is the normal link source address used by an interface. A computer sends all packets which it creates with its own hardware source link address, and receives all packets which match the same hardware address in the destination field or one (or more) pre-selected broadcast/multicast addresses.

The Ethernet address is a link layer address and is dependent on the interface card which is used. IP operates at the network layer and is not concerned with the [link addresses](#) of individual nodes which are to be used. The address resolution protocol (arp) is therefore used to translate between the two types of address. The arp client and server processes operate on all computers using [IP over Ethernet](#). The processes are normally implemented as part of the software driver that drives the [network interface card](#).

There are four types of arp messages that may be sent by the arp protocol. These are identified by four values in the "operation" field of an arp message. The types of message are:

1. ARP-Request (Broadcast, source IP address of the requester)
2. ARP-Reply (Unicast to requester, the target)

The format of an arp message is shown below:

0 2 16 32

这两个文件都简略的介绍了 ARP 协议的相关内容

(a) How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

```
Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
  Sender IP address: 192.168.43.1
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.43.193

0000 30 d1 6b 97 9f c5 ca fe ec a1 7a e1 08 06 00 01 0-k-...z-...
0010 08 00 06 04 00 01 ca fe ec a1 7a e1 c0 a8 2b 01 -...-z-...+
0020 00 00 00 00 00 00 c0 a8 2b c1 -...-+...
```

我选中了 opcode 项，这里可以看到是从第 21 个字节开始，也就是距离这个以太网帧开头 20 个字节之后。

(b) What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP request is made?

截图里面有一行 Opcode: request (1)，表示这个 opcode 项的值为 1

(c) Does the ARP message contain the IP address of the sender?

里面有一行 Sender IP address: 192.168.43.1，表示这个 ARP 报文发送者的 IP 地址为 192.168.43.1

(d) Where in the ARP request does the “question” appear –the Ethernet address of the machine whose corresponding IP address is being queried?

截图里面可以看到 Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00) 以及 Target IP address: 192.168.43.193，表示我需要 IP 地址 192.168.43.193 对应的物理地址，但是这里不知道，物理地址用 00:00:00:00:00:00。从这里可以看到在询问这个 MAC 地址

当然如果看这个 info

```
1012 8.144345 ca:fe:ec:a1:7a:e1 LiteonTe_97:9f:c5 ARP 42 Who has 192.168.43.193? Tell 192.168.43.1
1013 8.144360 LiteonTe_97:9f:c5 ca:fe:ec:a1:7a:e1 ARP 42 192.168.43.193 is at 30:d1:6b:97:9f:c5
```

请求报文的 info 里面有一个问题，询问目的 ip 的物理地址，Who has 192.168.43.193? Tell 192.168.43.1

13. Now find the ARP reply that was sent in response to the ARP request. (a) How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

```
No.    Time    Source          Destination      Protocol Length Info
1013  8.144360 LiteonTe_97:9f:c5 ca:fe:ec:a1:7a:e1 ARP 42 192.168.43.193 is at 30:d1:6b:97:9f:c5
97:9f:c5
Frame 1013: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
\Device\NPF_{D7A1008D-93D5-4F58-95D5-A4C7F7828D05}, id 0
Ethernet II, Src: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5), Dst: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Destination: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
.....1. .... = IG bit: Locally administered address (this is NOT the factory default)
.....0 .... = IG bit: Individual address (unicast)
Source: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
.....0 .... = IG bit: Globally unique address (factory default)
.....0 .... = IG bit: Individual address (unicast)
Type: ARP (0x0806)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)
Sender IP address: 192.168.43.193
Target MAC address: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)
Target IP address: 192.168.43.1
```

我选中了 opcode 项，这里可以看到是从第 21 个字节开始，也就是距离这个以太网帧开头 20 个字节之后。

(b) What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP response is made?

截图里面有一行 Opcode: reply (2)，表示这个 opcode 项的值为 2

(c) Where in the ARP message does the “answer” to the earlier ARP request appear –the IP address of the machine having the Ethernet address whose corresponding IP address is being queried?

1012 8.144345	ca:fe:ec:a1:7a:e1	LiteonTe_97:9f:c5	ARP	42	Who has 192.168.43.193? Tell 192.168.43.1
1013 8.144360	LiteonTe_97:9f:c5	ca:fe:ec:a1:7a:e1	ARP	42	192.168.43.193 is at 30:d1:6b:97:9f:c5

可以看这个 info,响应报文的 info 里面有这个答案,192.168.43.193 is at 30:d1:6b:97:9f:c5,不过这个 info 不知道是在那里储存的 orz, 不过也可以看 Sender MAC address: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5)Sender IP address: 192.168.43.193, 也能体现这个 ip 地址对应的物理地址

14. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?

源地址 Source: LiteonTe_97:9f:c5 (30:d1:6b:97:9f:c5),目的地址 Destination: ca:fe:ec:a1:7a:e1 (ca:fe:ec:a1:7a:e1)

15. Open the ethernet-ethereal-trace-1 trace file in <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip>. The first and second ARP packets in this trace correspond to an ARP request sent by the computer running Wireshark, and the ARP reply sent to the computer running Wireshark by the computer with the ARP-requested Ethernet address. But there is yet another computer on this network, as indicated by packet 6 –another ARP request. Why is there no ARP reply (sent in response to the ARP request in packet 6) in the packet trace?

下面是这个作者提供的 trace file 截图

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	AmbitMtc_a9:3d:68	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.105
2	0.000010	LinksysG_da:af:73	AmbitMtc_a9:3d:68	ARP	68	192.168.1.1 is at 08:00:25:da:af:73
3	0.001020	AmbitMtc_a9:3d:68	LinksysG_da:af:73	0x0800	62	IPv4
4	2.962050	AmbitMtc_a9:3d:68	LinksysG_da:af:73	0x0800	62	IPv4
5	0.971480	AmbitMtc_a9:3d:68	LinksysG_da:af:73	0x0800	62	IPv4
6	13.542974	CnetTech_73:8d:c6	Broadcast	ARP	68	Who has 192.168.1.117? Tell 192.168.1.104
7	17.444423	AmbitMtc_a9:3d:68	LinksysG_da:af:73	0x0800	62	IPv4

这里第二次没有收到 reply 消息是因为，广播消息是所有这个网络上的主机都能接受到的，但是响应消息是单播的，只有发出该请求的主机才能接收到。

2.3 Extra Credit

EX-1. The arp command: `arp -s InetAddr EtherAddr` allows you to manually add an entry to the ARP cache that resolves the IP address `InetAddr` to the physical address `EtherAddr`. What would happen if, when you manually added an entry, you entered the correct IP address, but the wrong Ethernet address for that remote interface?

这里我尝试了对已有的项添加另一个物理地址，那这个物理地址肯定是不对的，得到下面的结果

```
C:\windows\system32>arp -d *
C:\windows\system32>arp -a

接口: 192.168.11.1 --- 0x4
Internet 地址      物理地址      类型
224.0.0.22         01-00-5e-00-00-16 静态

接口: 192.168.43.193 --- 0x11
Internet 地址      物理地址      类型
192.168.43.1       ca-fe-ec-a1-7a-e1 动态
224.0.0.22         01-00-5e-00-00-16 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

接口: 192.168.40.1 --- 0x14
Internet 地址      物理地址      类型
224.0.0.22         01-00-5e-00-00-16 静态

C:\windows\system32>arp -s 192.168.43.1 10-10-10-10-10-10 192.168.43.193
ARP 项添加失败: 拒绝访问。
```

是拒绝访问

同时我还尝试了，对尚未添加的项，进行添加，并给出错误的物理地址，结果如下

```
C:\windows\system32>arp -s 224.0.0.251 10-10-10-10-10-10 192.168.43.193
C:\windows\system32>arp -a

接口: 192.168.11.1 --- 0x4
Internet 地址      物理地址      类型
224.0.0.22         01-00-5e-00-00-16 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态

接口: 192.168.43.193 --- 0x11
Internet 地址      物理地址      类型
192.168.43.1       ca-fe-ec-a1-7a-e1 动态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

接口: 192.168.40.1 --- 0x14
Internet 地址      物理地址      类型
224.0.0.22         01-00-5e-00-00-16 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
```

添加的项里面直接把我给的错误物理地址修改为正确的了。

综上，如果插入正确的 ip 地址和错误的 mac 地址，这个会在访问那个 ip 地址时进行更新，修改为正确的

EX-2. What is the default amount of time that an entry remains in your ARP cache before being removed. You can determine this empirically (by monitoring the cache contents) or by looking this up in your operation system documentation. Indicate how/where you determined this value.

这个我在 win10 系统里面不知道在哪找这个配置文件，但是 Ubuntu 里面比较好找，下面是对应的截图

```
swx@ubuntu: /proc/sys/net/ipv4/neigh/default
File Edit View Search Terminal Help
swx@ubuntu:~$ cd /proc/sys/net/ipv4/neigh/default
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ ls
anycast_delay      gc_interval      locktime          retrans_time
app_solicit        gc_stale_time    mcast_resolicit   retrans_time_ms
base_reachable_time gc_thresh1        mcast_solicit     ucast_solicit
base_reachable_time_ms gc_thresh2      proxy_delay       unres_qlen
delay_first_probe_time gc_thresh3      proxy_qlen        unres_qlen_bytes
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ gedit base_reachable_time
base_reachable_time [Read-Only]
/proc/sys/net/ipv4/neigh/default
30
```

这个 `base_reachable_time` 就是这个 arp 项从 reachabe 变成 stale 状态的时间,具体来说就是 `base_reachable_time / 2` 到 `3 * base_reachable_time / 2` 之间如果该项变成了 stale 状态,只要我们在对这个 ip 发送请求,又会变成

```
swx@ubuntu: /proc/sys/net/ipv4/neigh/default
File Edit View Search Terminal Help
swx@ubuntu:~$ cd /proc/sys/net/ipv4/neigh/default
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ ls
anycast_delay      gc_interval      locktime          retrans_time
app_solicit        gc_stale_time    mcast_resolicit   retrans_time_ms
base_reachable_time gc_thresh1        mcast_solicit     ucast_solicit
base_reachable_time_ms gc_thresh2      proxy_delay       unres_qlen
delay_first_probe_time gc_thresh3      proxy_qlen        unres_qlen_bytes
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ gedit base_reachable_time
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ gedit gc_stale_time
gc_stale_time [Read-Only]
/proc/sys/net/ipv4/neigh/default
60
```

这个 `gc_stale_time`, 该项是变成 stale 状态后, 过多久可以被 gc(garbage collection) 回收

```
swx@ubuntu: /proc/sys/net/ipv4/neigh/default
File Edit View Search Terminal Help
swx@ubuntu:~$ cd /proc/sys/net/ipv4/neigh/default
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ ls
anycast_delay      gc_interval      locktime          retrans_time
app_solicit        gc_stale_time    mcast_resolicit   retrans_time_ms
base_reachable_time gc_thresh1        mcast_solicit     ucast_solicit
base_reachable_time_ms gc_thresh2      proxy_delay       unres_qlen
delay_first_probe_time gc_thresh3      proxy_qlen        unres_qlen_bytes
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ gedit base_reachable_time
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ gedit gc_stale_time
swx@ubuntu:/proc/sys/net/ipv4/neigh/default$ gedit gc_interval
gc_interval [Read-Only]
/proc/sys/net/ipv4/neigh/default
30
```

这个 `gc_interval` 是 gc 进程运行的间隔, 每隔 `gc_interval` 扫描一次。

这些是基本的一些参数, 不过实际上这个 ARP 缓存的项的删除还涉及到其他一些复杂的参数和机制, 也取决于当时的环境如何, 比较复杂 orz