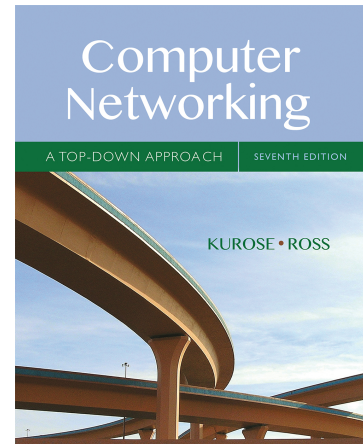


# Wireshark Lab: Ethernet and ARP v7.0

Supplement to *Computer Networking: A Top-Down Approach*, 7<sup>th</sup> ed., J.F. Kurose and K.W. Ross

“Tell me and I forget. Show me and I remember. Involve me and I understand.” Chinese proverb

© 2005-2016 J.F Kurose and K.W. Ross, All Rights Reserved



In this lab, we'll investigate the Ethernet protocol and the ARP protocol. Before beginning this lab, you'll probably want to review sections 6.4.1 (Link-layer addressing and ARP) and 6.4.2 (Ethernet) in the text<sup>1</sup>. RFC 826 ([ftp://ftp.rfc-editor.org/in-notes/std/std37.txt](http://ftp.rfc-editor.org/in-notes/std/std37.txt)) contains the gory details of the ARP protocol, which is used by an IP device to determine the IP address of a remote interface whose Ethernet address is known.

在本实验中，我们将研究以太网协议和 ARP 协议。在开始实验之前，您可以查看课本的 6.4.1 节（链路层地址和 ARP）和 6.4.2（以太网），您也可以去看 RFC 826 ([ftp://ftp.rfc-editor.org/in-notes/std/std37.txt](http://ftp.rfc-editor.org/in-notes/std/std37.txt)) 了解关于 ARP 的协议详细信息，该协议可以根据 IP 地址获取远程主机的物理地址。

## 1. Capturing and analyzing Ethernet frames 捕获和分析以太网帧

Let's begin by capturing a set of Ethernet frames to study. Do the following<sup>2</sup>:  
让我们从捕获一组以太网帧开始研究。请执行下列操作：

- First, make sure your browser's cache is empty. To do this under Mozilla Firefox V3, select *Tools->Clear Recent History* and check the box for Cache. For Internet

---

<sup>1</sup> References to figures and sections are for the 7<sup>th</sup> edition of our text, *Computer Networks, A Top-down Approach*, 7<sup>th</sup> ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2016.

<sup>2</sup> If you are unable to run Wireshark live on a computer, you can download the zip file <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip> and extract the file *ethernet--ethereal-trace-1*. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the *ethernet-ethereal-trace-1* trace file. You can then use this trace file to answer the questions below.  
如果您无法抓包，可以使用作者抓包结果 ethernet-ethereal-trace-1

Explorer, select *Tools->Internet Options->Delete Files*. Start up the Wireshark packet sniffer

首先，确保浏览器的缓存为空(清除浏览器缓存)。要在 Mozilla Firefox V3 下执行此操作，请选择工具 ->清除最近历史记录，然后选中缓存框。对于 Internet Explorer，选择工具 -> Internet 选项 ->删除文件。然后启动 Wireshark 数据包嗅探器

- Enter the following URL into your browser  
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html>  
Your browser should display the rather lengthy US Bill of Rights.  
打开以下 URL  
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html>  
您的浏览器应显示相当冗长的美国权利法案。

Stop Wireshark packet capture. First, find the packet numbers (the leftmost column in the upper Wireshark window) of the HTTP GET message that was sent from your computer to gaia.cs.umass.edu, as well as the beginning of the HTTP response message sent to your computer by gaia.cs.umass.edu. You should see a screen that looks something like this (where packet 4 in the screen shot below contains the HTTP GET message)

接下来停止 Wireshark 数据包捕获，找到您向 gaia.cs.umass.edu 的 HTTP GET 消息的数据包编号以及 gaia.cs.umass.edu 相应您的 HTTP 回应。您的抓包结果应看起来向下面一样：

The screenshot shows the Wireshark interface with a packet list, packet details, and packet bytes pane. The packet list shows 21 packets. Packet 4 is selected, showing an HTTP GET request. The packet details pane shows the structure of the packet, including Ethernet II, Internet Protocol, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.145	128.119.245.12	TCP	2038 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.050606	128.119.245.12	192.168.2.145	TCP	http > 2038 [SYN, ACK] Seq=0 Ack=1 win=5
3	0.050729	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=1 Ack=1 win=65535
4	0.055906	192.168.2.145	128.119.245.12	HTTP	GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1
5	0.128700	128.119.245.12	192.168.2.145	TCP	http > 2038 [ACK] Seq=1 Ack=453 win=6432
6	0.134167	128.119.245.12	192.168.2.145	TCP	[TCP segment of a reassembled PDU]
7	0.150302	128.119.245.12	192.168.2.145	TCP	[TCP segment of a reassembled PDU]
8	0.150487	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=453 Ack=1762 win=6
9	0.213639	128.119.245.12	192.168.2.145	TCP	[TCP segment of a reassembled PDU]
10	0.213724	128.119.245.12	192.168.2.145	TCP	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
11	0.215947	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=453 Ack=3214 win=6
12	0.231749	128.119.245.12	192.168.2.145	HTTP	[TCP Retransmission] HTTP/1.1 200 OK (text/html)
13	0.232145	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=453 Ack=4810 win=6
14	0.320470	192.168.2.145	128.119.245.12	HTTP	GET /favicon.ico HTTP/1.1
15	0.403428	128.119.245.12	192.168.2.145	HTTP	HTTP/1.1 404 Not Found (text/html)
16	0.423932	192.168.2.145	168.66.12.224	TCP	2039 > http [SYN] Seq=0 Len=0 MSS=1460
17	0.570922	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=793 Ack=6235 win=6
18	3.383584	192.168.2.145	168.66.12.224	TCP	2039 > http [SYN] Seq=0 Len=0 MSS=1460
19	9.392197	192.168.2.145	168.66.12.224	TCP	2039 > http [SYN] Seq=0 Len=0 MSS=1460
20	10.389131	128.119.245.12	192.168.2.145	TCP	http > 2038 [FIN, ACK] Seq=6235 Ack=793
21	10.389258	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=793 Ack=6236 win=6

Frame 4 (506 bytes on wire, 506 bytes captured)

Ethernet II, Src: Netgear\_61:8e:6d (00:09:5b:61:8e:6d), Dst: LinksysG\_45:90:a8 (00:0c:41:45:90:a8)

Internet Protocol, Src: 192.168.2.145 (192.168.2.145), Dst: 128.119.245.12 (128.119.245.12)

Transmission Control Protocol, Src Port: 2038 (2038), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452

Hypertext Transfer Protocol

GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1\r\n

Request Method: GET

Request URI: /wireshark-labs/HTTP-ethereal-lab-file3.html

Request Version: HTTP/1.1

Host: gaia.cs.umass.edu\r\n

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4\r\n

Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5\r\n

Accept-Language: en-us,en;q=0.5\r\n

Accept-Encoding: gzip,deflate\r\n

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7\r\n

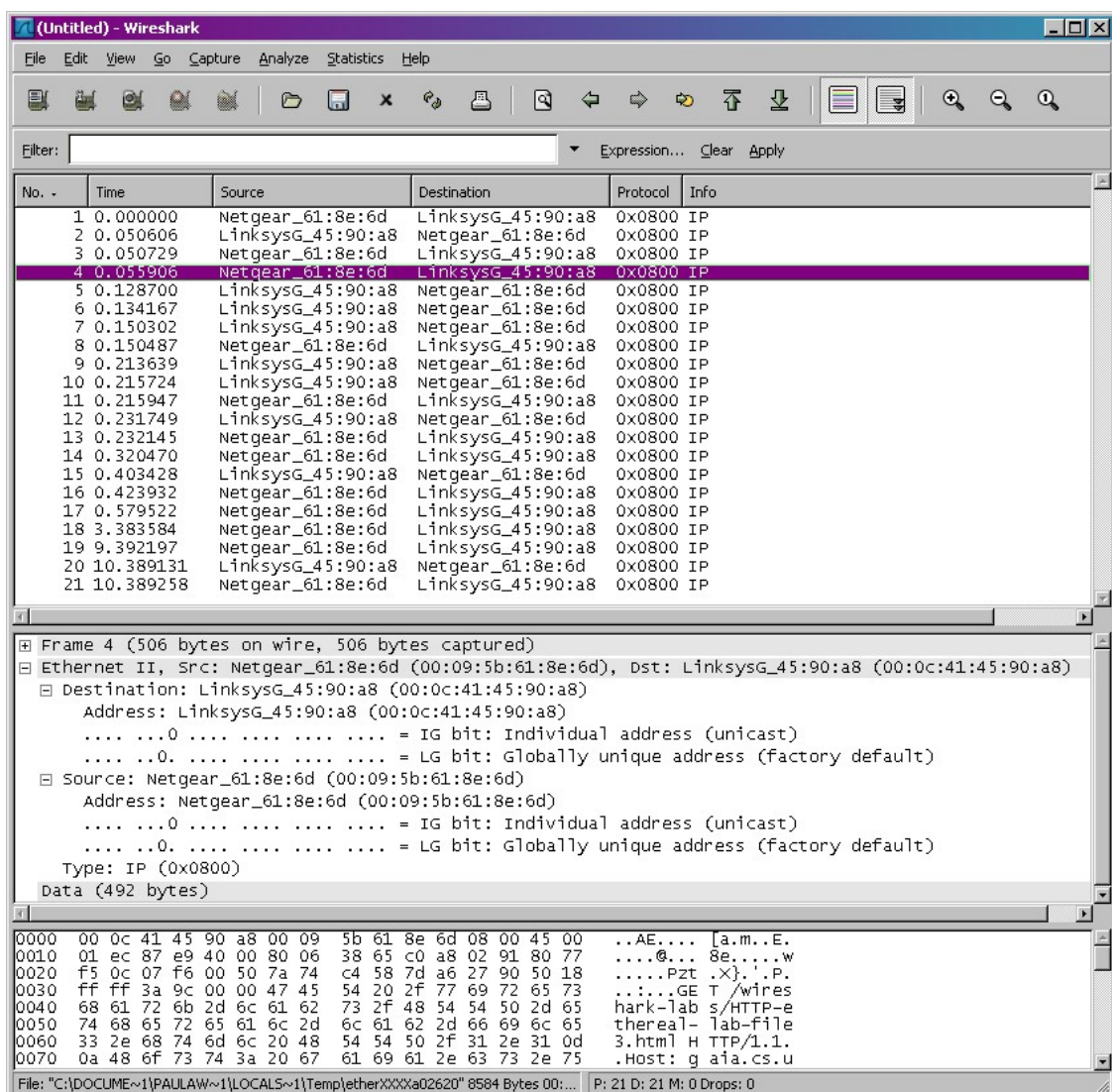
Keep-Alive: 300\r\n

Connection: keep-alive\r\n\r\n

P: 21 D: 21 M: 0 Drops: 0

- Since this lab is about Ethernet and ARP, we're not interested in IP or higher-layer protocols. So let's change Wireshark's "listing of captured packets" window so that it shows information only about protocols below IP. To have Wireshark do this, select *Analyze->Enabled Protocols*. Then uncheck the IP box and select *OK*. You should now see an Wireshark window that looks like:

由于本实验是关于以太网和 ARP 的，我们对 IP 或更高层协议不感兴趣。因此，让我们更改 Wireshark 的“捕获数据包列表”窗口，以便它仅显示有关 IP 以下协议的信息。要让 Wireshark 执行此操作，请选择 Analyze-> Enabled Protocols(分析-启用的协议)。然后取消选中 IP 框(译者注：这里指的 **IPV4 协议**，下面有搜索)并选择确定。您现在 Wireshark 窗口应该如下所示：



In order to answer the following questions, you'll need to look into the packet details and packet contents windows (the middle and lower display windows in Wireshark).

为了回答以下问题，您需要查看数据包详细信息和数据包内容窗口（Wireshark 中的中间和下部显示窗口，译者注：若您看到的不是如此-建议您重置布局（视图-重置布局））。

Select the Ethernet frame containing the HTTP GET message. (Recall that the HTTP GET message is carried inside of a TCP segment, which is carried inside of an IP datagram, which is carried inside of an Ethernet frame; reread section 1.5.2 in the text if you find this encapsulation a bit confusing). Expand the Ethernet II information in the packet details window. Note that the contents of the Ethernet frame (header as well as payload) are displayed in the packet contents window.

选择包含 HTTP GET 消息的以太网帧。（回想一下，HTTP GET 请求是被加上 TCP 头封装到 TCP 段进行传输，TCP 段加上 IP 头被封装到 IP 数据报进行传输，IP 数据报又被加上以太网头封装成以太网帧进行传输；如果你发现这个封装有点令人困惑，请重读文本中的第 1.5.2 节）。在数据包详细信息窗口中展开以太网 II 信息。请注意，以太网帧的内容（标题以及有效负载）显示在数据包内容窗口中

Answer the following questions, based on the contents of the Ethernet frame containing the HTTP GET message. Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout<sup>3</sup> to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

根据包含 HTTP GET 消息的以太网帧进行分析，如果有可能建议您使用标记的方式展现您的答案。

1. What is the 48-bit Ethernet address of your computer?  
你的电脑 48 位的地址是多少
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.]  
以太网帧中的 48 位目标地址是什么？这是 gaia.cs.umass.edu 的以太网地址吗？（提示：答案是否定的）。那么它是什么？注意这一题可能会犯错，请阅读 468-469(中文版 305-308 页)然后理解它
3. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

---

<sup>3</sup> What do we mean by “annotate”? If you hand in a paper copy, please highlight where in the printout you’ve found the answer and add some text (preferably with a colored pen) noting what you found in what you’ve highlight. If you hand in an electronic copy, it would be great if you could also highlight and annotate.

以太网帧上层协议 16 进制值是什么?这对应的上层协议是什么?

4. How many bytes from the very start of the Ethernet frame does the ASCII “G” in “GET” appear in the Ethernet frame?  
从以太帧的开始，一直到“GET”中的 ASCII“G”出现在以太网帧中为止，有多少字节?

Next, answer the following questions, based on the contents of the Ethernet frame containing the first byte of the HTTP response message.

接下来，根据包含 HTTP 响应消息的第一个字节的以太网帧的内容，回答以下问题。

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?  
这个以太网帧中，以太网源地址的值是多少？这是你的计算机的地址，还是 gaia.cs.umass.edu 的地址（提示：答案是否定的）。拥有这个以太网地址的设备是什么？
6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?  
以太网帧中的目的地址是什么？这是您的计算机的以太网地址吗？
7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?  
以太网帧上层协议 16 进制值是什么?这对应的上层协议是什么？
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?  
从以太帧的开始，一直到“OK”中的 ASCII“O”出现在以太网帧中为止，有多少字节？



## 2. The Address Resolution Protocol 地址解析协议

In this section, we'll observe the ARP protocol in action. We strongly recommend that you re-read section 6.4.1 in the text before proceeding.

在本节中，我们将观察 ARP 协议的作用。我们强烈建议您在继续实验之前重读课文 6.4.1 节

### ARP Caching(ARP 缓存)

Recall that the ARP protocol typically maintains a cache of IP-to-Ethernet address translation pairs on your computer. The *arp* command (in both MSDOS and Linux/Unix) is used to view and manipulate the contents of this cache. Since the *arp* command and the ARP protocol have the same name, it's understandably easy to confuse them. But keep in mind that they are different - the *arp* command is used to view and manipulate the ARP cache contents, while the ARP protocol defines the format and meaning of the messages sent and received, and defines the actions taken on message transmission and receipt.

回想一下，ARP 协议通常在您的计算机上维护 IP 到以太网地址转换对的缓存。*arp* 命令（在 MSDOS 和 Linux / Unix 中）用于查看和操作此缓存的内容。由于 *arp* 命令和 ARP 协议具有相同的名称，因此很容易混淆它们。但请记住，它们是不同的：*arp* 命令用于查看和操作 ARP 缓存内容，而 ARP 协议定义了发送和接收的消息的格式和含义，并定义了对消息传输和接收所采取的操作。

Let's take a look at the contents of the ARP cache on your computer:

我们来看看您计算机上 ARP 缓存的内容：

- **MS-DOS.** The *arp* command is in `c:\windows\system32`, so type either "*arp*" or "`c:\windows\system32\arp`" in the MS-DOS command line (without quotation marks).  
MS-DOS: *arp* 命令位于 `c:\windows\system32` 中，因此在 MS-DOS 命令行中输入 "*arp*" 或 "`c:\windows\system32\arp`"（没有引号）
- **Linux/Unix/MacOS.** The executable for the *arp* command can be in various places. Popular locations are `/sbin/arp` (for linux) and `/usr/etc/arp` (for some Unix variants).  
**Linux/Unix/MacOS.** 根据安装位置不同路径而不同，一般有 `/sbin/arp` (linux) 和 `/usr/etc/arp` (Unix)

The Windows *arp* command with no arguments will display the contents of the ARP cache on your computer. Run the *arp* command.

没有参数的 Windows *arp* 命令将显示计算机上 ARP 缓存的内容。运行 ARP 命令。

(译者注:我在做实验发现单单一个命令并不能显示 ARP 表，应该运行 `arp -a` 才对)

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

写下计算机 ARP 缓存的内容。每个列值的含义是什么？

In order to observe your computer sending and receiving ARP messages, we'll need to clear the ARP cache, since otherwise your computer is likely to find a needed IP-Ethernet address translation pair in its cache and consequently not need to send out an ARP message.

为了观察您的计算机发送和接收 ARP 消息，我们需要清除 ARP 缓存，否则您的计算机很可能在其缓存中找到所需的 IP-Ethernet 地址转换关系，因此不会发送 ARP 消息。

- **MS-DOS.** The MS-DOS *arp -d \** command will clear your ARP cache. The *-d* flag indicates a deletion operation, and the *\** is the wildcard that says to delete all table entries.  
MS-DOS: MS-DOS *arp -d \**命令将清除 ARP 缓存。*-d* 标志指示删除操作，*\** 是表示删除所有表项的通配符。
- **Linux/Unix/MacOS.** The *arp -d \** will clear your ARP cache. In order to run this command you'll need root privileges. If you don't have root privileges and can't run Wireshark on a Windows machine, you can skip the trace collection part of this lab and just use the trace discussed in the earlier footnote.  
**Linux/Unix/MacOS: 清除 arp 缓存的** *arp -d \**需要 root 权限，如果您没有也没办法使用 Windows 系统进行实验，请下载作者的抓包结果。



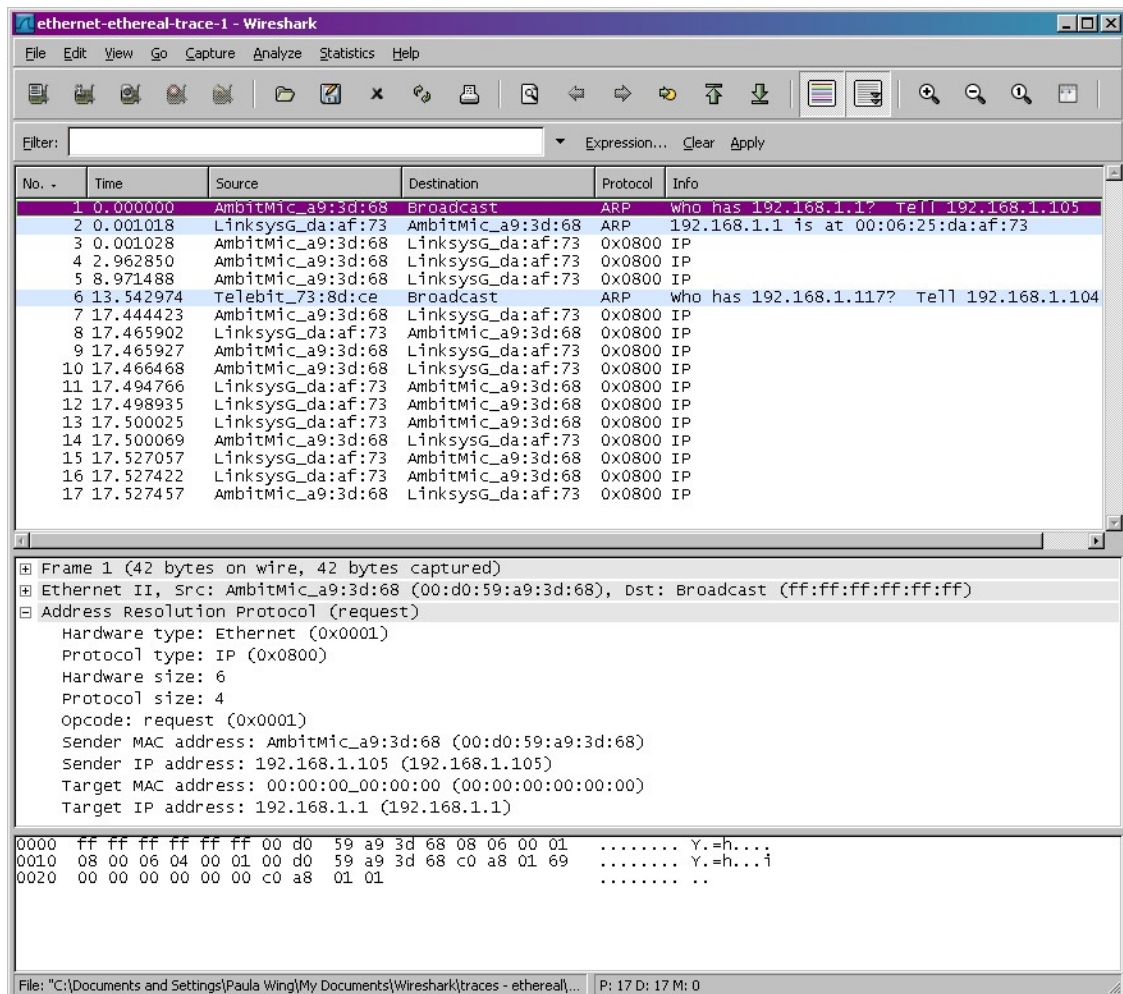
## Observing ARP in action 抓取 ARP 消息

Do the following<sup>4</sup>: 请进行以下操作

- Clear your ARP cache, as described above.  
清除你的 ARP 缓存，如上所述。
- Next, make sure your browser's cache is empty. To do this under Mozilla Firefox V3, select *Tools->Clear Recent History* and check the box for Cache. For Internet Explorer, select *Tools->Internet Options->Delete Files*.  
接下来，确保浏览器的缓存是空的。要在 Mozilla Firefox V3 下执行此操作，请选择工具->清除最近的历史并检查缓存的方框。对于 Internet Explorer，选择工具-> Internet 选项>删除文件
- Start up the Wireshark packet sniffer  
启动 Wireshark 捕捉封包
- Enter the following URL into your browser  
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-lab-file3.html>  
Your browser should again display the rather lengthy US Bill of Rights.
- 打开以下 URL， <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-lab-file3.html>。你的浏览器应该再次显示相当长的美国权利法案
- Stop Wireshark packet capture. Again, we're not interested in IP or higher-layer protocols, so change Wireshark's "listing of captured packets" window so that it shows information only about protocols below IP. To have Wireshark do this, select *Analyze->Enabled Protocols*. Then uncheck the IP box and select *OK*.  
You should now see an Wireshark window that looks like:  
同样设置不显示 IP 和更高层协议，请选择 Analyze-> Enabled Protocols(分析-启用的协议)。然后取消选中 IP 框(译者注：这里指的 **IPV4 协议**，下面有搜索)并选择确定。您现在 Wireshark 窗口应该如下所示：

---

<sup>4</sup> The *ethernet-ethereal-trace-1* trace file in <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip> was created using the steps below (in particular after the ARP cache had been flushed).  
如果您无法抓包，建议使用作者的抓包结果 *ethernet-ethereal-trace-1*



In the example above, the first two frames in the trace contain ARP messages (as does the 6<sup>th</sup> message). The screen shot above corresponds to the trace referenced in footnote 1.

图示是作者的抓包结果截图，您可以发现第 1，2，6 帧都包含 ARP 消息。

Answer the following questions: (回答下列问题)

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

包含 ARP 请求消息的以太网帧中源和目标地址的十六进制值是什么？

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

以太网帧上层协议 16 进制值是什么？

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 ARP 规范(<ftp://ftp.rfc-editor.org/in-notes/std/std37.txt>),讨论请移步 (<http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html>)
- a) How many bytes from the very beginning of the Ethernet frame does the ARP *opcode* field begin?  
ARP 操作码字段开始从以太网帧的最开始有多少字节?
  - 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?  
在进行 ARP 请求的以太网帧的 ARP 负载部分中, 操作码字段的值是多少?
  - c) Does the ARP message contain the IP address of the sender?  
ARP 消息是否包含发送方的 IP 地址?
  - d) Where in the ARP request does the “question” appear – the Ethernet address of the machine whose corresponding IP address is being queried?  
在 ARP 请求中从哪里看出我们要查询相应 IP 的以太网地址?
13. Now find the ARP reply that was sent in response to the ARP request.  
找到相应 ARP 请求的而发送 ARP 回复
- a) How many bytes from the very beginning of the Ethernet frame does the ARP *opcode* field begin?  
ARP 操作码字段开始从以太网帧的最开始有多少字节?
  - 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?  
在进行 ARP 响应的以太网帧的 ARP 负载部分中, 操作码字段的值是多少?
  - 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?  
在响应 ARP 中从哪里看出早期 ARP 请求的答案?
14. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?
- 包含 ARP 回复消息的以太网帧中的源地址和目标地址的十六进制值是多少?
15. Open the *ethernet-etherreal-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?

在作者抓包结果中，他有两台电脑，一台运行 wireshark 进行抓包，一台没有，那么为什么运行 wireshark 那台电脑发送 ARP 请求得到了应答，另外一台电脑的 ARP 请求没有得到应答?(没有相应第 6 帧的 ARP 的请求)

### Extra Credit 额外实验

EX-1. The *arp* command: *arp* 命令:

```
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?

这个命令允许你手动添加 arp 记录到缓存表中。它会把您输入的 IP 地址 (*InetAddr*) 解析为物理地址 (*EtherAddr*)，请问您输入正确 IP 地址但是物理地址错误会发生什么。

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.

在删除 ARP 缓存之前，请问它们默认的有效时间是多少，您可以通过不定时的查看缓存内容得出结论或者查询相应的操作系统文档。