Ethereal Lab: Ethernet and ARP

In this last lab, we'll investigate the Ethernet protocol and the ARP protocol. Before beginning this lab, you'll probably want to review sections 5.5 (Ethernet), 5.4.1 (link-layer addressing) and 5.4.2 (ARP) in the text. RFC 826 (<u>ftp://ftp.rfc-editor.org/in-notes/std/std37.txt</u>) 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.

1. Capturing and analyzing Ethernet frames

The trace file that you will use for this lab was created by the following steps:

- The browser's cache where ethereal was running was emptied. (This can be done on Netscape 7.0 by selecting *Edit->Preferences->Advanced->Cache* and clearing the memory and disk cache. For Internet Explorer, select *Tools->Internet Options->Delete Files.*)
- The Ethereal packet sniffer was started.
- The following URL was entered into the browser: http://gaia.cs.umass.edu/ethereal-labs/ HTTP-ethereal-lab-file3.html
- The Ethereal packet capture was stopped.

Steps for you to take:

• First, find the packet numbers (the leftmost column in the upper Ethereal window) of the HTTP GET message that was sent from the client computer to gaia.cs.umass.edu, as well as the beginning of the HTTP response message sent to the client computer by gaia.cs.umass.edu. You should see a screen that looks like this (where packet 10 in the screen shot below contains the HTTP GET message)

ⓒ ethernet-ethereal-lab-trace-1 - Ethereal			
Elle Edit View Go Capture Analyze Statistics Help			
🕒 Elter: 🍡 🛧 Expression 🗞 Qear 🗸 Apply			
No Time Source Destination	Protocol Info		
1 0.000000 AmbitMic_a9:3d:68 Broadcast 2 0.001018 Linksysg_da:af:73 AmbitMic_a9:3d:6 3 0.001028 192:168.1.105 199.2.53.206 4 2.962850 192:168.1.105 199.2.53.206 5 8.971488 192:168.1.105 199.2.53.206 6 13.542974 Telebit_73:8d:ce Broadcast 7 17.444423 192:168.1.105 128.119.245.12 9 17.465927 192:168.1.105 128.119.245.12 10 17.465645 192:168.1.105 128.119.245.12 10 17.465645 192.168.1.105 128.119.245.12 11 17.465645 192.168.1.105 128.119.245.12 12 17.494766 128.119.245.12 192:168.1.105 13 17.500025 128.119.245.12 192:168.1.105 14 17.500069 192:168.1.105 128.119.245.12 15 17.527057 128.119.245.12 192:168.1.105 16 17.527457 192:168.1.105 128.119.245.12 16 17.527457 192:168.1.105 128.119.245.12 17 7.527457 192:168.1.105 128.119.245.12 17 7.527457 192:168.1.105 128.119.245.12	ARP who has 192.168.1.17 rell 192.168.1.105 8 ARP 192.168.1.1 is at 00:06:25:46:a;af:73 TCP 1057 > 631 [SYN] seq=1691450891 Ack=0 win=64240 Len=0 MSS=1460 TCP 1057 > 631 [SYN] seq=1691450891 Ack=0 win=64240 Len=0 MSS=1460 TCP 1057 > 631 [SYN] seq=1691450891 Ack=0 win=64240 Len=0 MSS=1460 ARP who has 192.168.1.117 rell 192.168.1.104 TCP 1058 > http [SYN] seq=1691848870 Ack=0 win=64240 Len=0 MSS=1460 TCP https > 1058 [SYN, Seq=2896510899 Ack=1695848871 win=5840 Len=0 TCP https > 1058 [Ack] seq=2896510900 Ack=2896510900 win=64240 Len=0 HTTP GET //ethereal labs/HTTP=ethereal-labs-file3.html HTTP/1.1 TCP https > 1058 [Ack] seq=2896510900 Ack=1695849503 win=6952 Len=0 HTTP HTTP/1.1 200 0K (text/html) HTTP continuation TCP 1058 > http [Ack] seq=1695849503 Ack=2896513820 win=64240 Len=0 HTTP continuation TCP 1058 > http [Ack] seq=1695849503 Ack=2896513715 win=64240 Len=0		
P Frame 10 (686 bytes on wire, 686 bytes captured) > Ethernet II, Src: 00:d0:59:a9:3d:68, Dst: 00:06:25 Destination: 00:06:25:da:af:73 (LinksysG_da:af:73 Source: 00:d0:59:a9:3d:68 (AmbitMic_a9:3d:68) Type: IP (0x0800) > Internet Protocol, Src Addr: 192.168.1.105 (192.166 > Transmission Control Protocol, Src Port: 1058 (1055 > Hypertext Transfer Protocol 0010 02 a0 00 fa 40 00 80 06 bf c8 c0 a8 01 69 80 7 0020 f5 0c 04 22 00 50 65 14 99 a7 ac a5 37 b4 50 7 0030 fa 10 72 ef 00 00 47 45 54 20 27 65 74 68 65 7 0040 65 61 6c 2d 6c 61 62 73 2f 48 54 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 2 73 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 2 73 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 2 73 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 2 73 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 2 73 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 56 16 6c 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 54 50 2d 65 7 0050 68 57 26 57 16 16 2d 6c 66 16 27 3 2f 48 54 54 54 54 54 54 54 54 54 54 54 54 54	<pre> 3.1.105), Dst Addr: 128.119.245.12 (128.119.245.12) 3), Dst Port: http (80), Seq: 1695848871, Ack: 2896510900, Len: 632 </pre>		

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

🕝 ethernet-ethereal-lab-trace-1 - Ethereal			
<u>File E</u> dit <u>Vi</u> ew <u>Go</u> <u>Capture</u> <u>A</u> nalyze <u>S</u> tatistics <u>H</u> elp			
	♥ 7 ₺ 0 0 0 10 11 11 11 10		
Eiter:			
No Time Source Destinati	on Protocol Info		
3 0.001028 AmbitMic_a9:3d:68 Links; 4 2.962850 AmbitMic_a9:3d:68 Links; 5 8.971488 AmbitMic_a9:3d:68 Links; 6 13.542974 Telebit_73:8d:ce Broad 7 17.444423 AmbitMic_a9:3d:68 Links; 8 17.465902 LinksysG_da:af:73 AmbitM 9 17.465458 AmbitMic_a9:3d:68 Links; 10 17.465468 AmbitMic_a9:3d:68 Links; 11 17.494766 LinksysG_da:af:73 AmbitM 12 17.498985 LinksysG_da:af:73 AmbitM 13 17.500025 LinksysG_da:af:73 AmbitM 14 17.500069 AmbitMic_a9:3d:68 Links; 15 17.527057 LinksysG_da:af:73 AmbitM	<pre>Hic_a9:3d:68 ARP 192.168.1.1 1s at 00:06:25:da:af:73 /sG_da:af:73 0x0800 IP /sG_da:af:</pre>		
0010 02 a0 00 fa 40 00 80 06 bf c8 c0 a8 0020 f5 0c 04 22 00 50 65 14 99 a7 ac a5 0030 fa f0 7e 4f 00 00 47 45 54 20 2f 65 0040 65 61 6c 2d 6c 61 62 73 2f 48 54 54 0050 68 65 72 65 61 6c 2d 6c 61 62 7d 66	: 00:06:25:da:af:73 sG_da:af:73) :3d:68) 08 00 45 00%s Y.=hE. 01 69 80 77		

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 Ethereal).

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 1.7.2 in the text if you find this nesting 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.

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 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.

- 1. What is the 48-bit Ethernet address of the client computer?
- 2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu?

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

- 3. What is the value of the Ethernet source address? What device has this as its Ethernet address?
- 4. What is the destination address in the Ethernet frame? Is this the Ethernet address of the client computer?

2. The Address Resolution Protocol

In this section, we'll observe the ARP protocol in action. You should re-read section 5.4.2 in the text before proceeding.

ARP Caching

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.

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

- **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).
- Linux/Unix. 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).

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

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

Observing ARP in action

The trace you are analyzing was actually generated by the following steps:

- The ARP cache at the client computer was cleared (through the command arp -d *, which you need root privileges to run).
- The browser's cache was emptied.
- The Ethereal packet sniffer was started.
- The following URL was entered into the browser: http://gaia.cs.umass.edu/ethereal-labs/ HTTP-ethereal-lab-file3.html
- Ethereal packet capture was stopped.

Steps for you to take:

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

<u>File Edit Vi</u> ew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics <u>H</u> elp			
Eilter:			
No Time Source Destination Protocol Info			
1 0.000000 AmbitMic_a9:3d:68 Broadcast ARP Who has 192.168.1.1? Tell 192.168.1.1	L05		
2 0.001018 Linksysc_da:af:73 AmbitMic_a9:36:68 APP 192.168.1.1 is at 00:06:25:da:af:73			
3 0.001028 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP 4 2.962850 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP			
5.971488 AmbitMic_a013d:68 LinksySG_da:af:73 0X0800 IP			
6 13.542974 Telebit_73:8d:ce Broadcast ARP Who has 192.168.1.117? Tell 192.168.1	L.104		
7 17.444423 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP			
8 17.465902 LinksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP 9 17.465927 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP			
9 17.465927 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP 10 17.466468 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP			
11 17.494766 Linksysc_da:af:73 AmbitMic_a0:3d:68 0x0800 IP			
12 17.498935 LinksýsG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP			
13 17.500025 LinksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP			
14 17.500069 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP 15 17.527057 LinksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP			
15 17.527057 LinksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP 16 17.527422 LinksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP			
17 17.527457 AmbitMic_39:3d:68 Linksysg_da:af:73 0x0800 IP			
Packet Length: 42 bytes			
Capture Length: 42 bytes			
▽ Ethernet II, Src: 00:d0:59:a9:3d:68, Dst: ff:ff:ff:ff:ff			
Destination: ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff			
Source: 00:d0:59:a9:3d:68 (AmbitMic_a9:3d:68)			
Type: ARP (0x0806) Address Resolution Protocol (request)			
0000 ff ff ff ff ff 00 d0 59 a9 3d 68 08 06 00 01			
1			
L			
File: ethernet-ethereal-lab-t ∬P: 17 D: 17 M: 0	/		

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

Answer the following questions:

- 6. What are the hexadecimal values for the source and destination addresses in the first Ethernet frame containing the ARP request message?
- 7. Download the ARP specification from <u>ftp://ftp.rfc-editor.org/in-notes/std/std37.txt</u>. A readable, detailed discussion of ARP is also at <u>http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html</u>.
 - a) Does the ARP message contain the IP address of the sender?
 - b) Where in the ARP request does the "question" appear the Ethernet address of the machine whose corresponding IP address is being queried?
- 8. Now find the ARP reply that was sent in response to the ARP request.
 - a) 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?
- 9. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?

10. Note that the first and second ARP packets in the trace correspond to an ARP request sent by the computer running Ethereal, and the ARP reply sent to the computer running Ethereal 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?