Ethereal Lab: Ethernet and ARP



Computer Networking: A Topdown Approach Featuring the Internet, 3rd edition.

Version: 1.0 © 2005 J.F. Kurose, 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 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

Let's begin by capturing a set of Ethernet frames to study. Do the following¹:

- First, make sure your browser's cache is empty. (To do this under Netscape 7.0, select *Edit->Preferences->Advanced->Cache* and clear the memory and disk cache. For Internet Explorer, select *Tools->Internet Options->Delete Files*
- Start up the Ethereal packet sniffer
- Enter the following URL into your browser http://gaia.cs.umass.edu/ethereal-labs/ HTTP-ethereal-lab-file3.html Your browser should display the rather lengthy US Bill of Rights.
- Stop Ethereal packet capture. First, find the packet numbers (the leftmost column in the upper Ethereal 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 10 in the screen shot below contains the HTTP GET message)

¹ If you are unable to run Ethereal live on a computer, you can download the zip file <u>http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip</u> and extract the file *ethernet--ethereal-trace-1*. The traces in this zip file were collected by Ethereal running on one of the author's computers, while performing the steps indicated in the Ethereal lab. Once you have downloaded the trace, you can load it into Ethereal 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.

© ethernet-ethereal-lab-trace-1 - EtherealX		
Elle Edit View Go Capture Analyze Statistics Help		
	[•, •, •, •, • ▶ ▶ ₩ 💥 💆	
D fiker:	▼ Apply	
No Time Source Destination	Protocol Info	
1 0.00000 Amb1M1c_49:3d:68 Broadcast 2 0.001018 linksysc_da:af:73 Amb1tM1c_49:3d:68 3 0.001028 192.168.1.105 199.2.53.206 4 2.962850 192.168.1.105 199.2.53.206 6 13.542974 Teleb1t_73:8d:ce Broadcast 7 17.444423 192.168.1.105 128.119.245.12 8 17.465902 128.119.245.12 192.168.1.105 9 17.465927 192.168.1.105 128.119.245.12 10 17.466468 192.168.1.105 128.119.245.12 11 17.494766 128.119.245.12 192.168.1.105 12 17.498935 128.119.245.12 192.168.1.105 13 17.500025 128.119.245.12 192.168.1.105 13 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 17 17.527457 192.168.1.105 128.119.245.12 192.168.1.105 17 17.527457 192.168.1.105 128.119.245.12 192.168.1.105	ARP who has 192.168.1.1? Tell 192.168.1.105 ARP 192.168.1.1 is at 00:06:25:da: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 ARP who has 192.168.1.117? Tell 192.168.1.104 TCP 1058 > http [SYN] Seq=1691450891 Ack=0 win=64240 Len=0 MSS=1460 http > 1058 > http [SYN] Seq=1693450870 Ack=0 win=64240 Len=0 MSS=1460 tCP 1058 > http [SYN] Seq=1693450870 Ack=0 win=64240 Len=0 MSS=1460 tCP 1058 > http [ACK] Seq=2896510899 Ack=1695848871 win=5840 Len=1 TCP 1058 > http [ACK] Seq=2896510900 Ack=1695849510900 win=64240 Len=0 HTTP Get /etnereal=laos/HTTP-etnereal=lab=71163,html HTTP/1.1 TCP 1058 > http [ACK] Seq=1695849503 Ack=2896513820 win=64240 Len=0 HTTP Continuation TCP 1058 > http [ACK] Seq=1695849503 Ack=2896513820 win=64240 Len=0 HTTP Continuation TCP 1058 > http [ACK] Seq=1695849503 Ack=2896515715 win=64240 Len=0 HTTP Continuation	
▷ Frame 10 (686 bytes on wire, 686 bytes captured) ▷ Ethernet II, Src: 00:00:59:a9:30:68, Dst: 00:06:25:da Destination: 00:06:25:da:af:73 (inksysc_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 (192.168.1 ▷ Transmission Control Protocol, Src Port: 1058 (1058), ▷ Hypertext Transfer Protocol ♥ 0010 02 a0 00 fa 40 00 80 06 bf c8 c0 a8 01 69 80 77 0020 f5 0c 04 22 00 50 65 14 99 a7 ac a5 3f b4 50 18 0030 fa f0 7e 4f 00 00 47 55 42 02 f6 57 46 86 57 22 0040 65 61 6c 2d 6c 61 62 73 2f 48 54 54 50 2d 65 74 0050 68 65 72 65 61 6c 2d 6c 61 62 2d 66 69 66 65 33 Charlen the state of the form of the state of the form of the state of the state of the form of the form of the state of the form of the state of the form of the form of the form of the form of the state of the form	<pre>:af:73 .105), Dst Addr: 128.119.245.12 (128.119.245.12) 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		
Elle Edit View Go Capture Analyze Statistics Help		
$\textcircled{\begin{tabular}{lllllllllllllllllllllllllllllllllll$		
Elter:		
No Time Source Destination Protocol Info		
1 0.000000 AmbitMic_a9:3d:68 Broadcast ARP who has 192.168.1.17 Tell 192.168.1.105 2 0.001018 LinksysG_da:af:73 AmbitMic_a9:3d:68 ARP 192.168.1.17 Tell 192.168.1.105 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 8.971488 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP 6 13.542974 Telebit_73:8d:ce Broadcast ARP who has 192.168.1.117? Tell 192.168.1.104 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 17.465492 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP 10 17.466468 AmbitMic_a9:3d:68 No1800 IP InksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP 12 17.498935 LinksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP InksysG_da:af:73 AmbitMic_a9:3d:68 0x0800 IP 13 17.500069 <td></td>		
<pre> Frame 10 (686 bytes on wire, 686 bytes captured) Ethernet II, src: 00:d0:59:a9:3d:68, pst: 00:06:25:da:af:73 Destination: 00:06:25:da:af:73 (LinksysG_da:af:73) Source: 00:d0:59:a9:3d:68 (Ambitwic_a9:3d:68) Type: IP (0x0800) Data (672 bytes) </pre>		
Data (6/2 bytes) 0000 00 06 25 da af 73 00 d0 59 a9 3d 68 08 00 45 00%s Y.=hE. 0010 02 a0 00 fa 40 00 80 06 bf c8 c0 a8 01 69 80 77i.w 0020 f5 0c 04 22 00 50 65 14 99 a7 ac a5 3f b4 50 18pe		
Lose το το 60 74 64 66 το 0.40 64 54 56 το 76 74 το 7		

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 your computer?
- 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 450-451 in the text and make sure you understand the answer here.]

- 3. Give the hexadecimal value for the two-byte Frame type field. What do the bit(s) whose value is 1 mean within the flag field?
- 4. How many bytes from the very start of the Ethernet frame does the ASCII "G" in "GET" appear in the Ethernet frame?
- 5. What is the hexadecimal value of the CRC field in this Ethernet frame?

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

- 6. 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?
- 7. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?
- 8. Give the hexadecimal value for the two-byte Frame type field. What do the bit(s) whose value is 1 mean within the flag field?
- 9. 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?
- 10. What is the hexadecimal value of the CRC field in this Ethernet frame?

2. The Address Resolution Protocol

In this section, we'll observe the ARP protocol in action. We strongly recommend that you 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 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.

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

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.

- **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.
- Linux/Unix. 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 Ethereal on a Windows machine, you can skip the trace collection part of this lab and just use the trace discussed in footnote 1.

Observing ARP in action

Do the following²:

- Clear your ARP cache, as described above.
- Next, make sure your browser's cache is empty. (To do this under Netscape 7.0, select *Edit->Preferences->Advanced->Cache* and clear the memory and disk cache. For Internet Explorer, select *Tools->Internet Options->Delete Files.*)
- Start up the Ethereal packet sniffer
- Enter the following URL into your browser http://gaia.cs.umass.edu/ethereal-labs/ HTTP-ethereal-lab-file3.html Your browser should again display the rather lengthy US Bill of Rights.
- Stop Ethereal packet capture. 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:

² The *ethernet--ethereal-trace-* trace file in <u>http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip</u> was created using the steps below (in particular after the ARP cache had been flushed).

🔇 ethernet-ethereal-lab-trace-1 - Ethereal		
Eile Edit <u>Vi</u> ew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics <u>H</u> elp		
Eilter: Succession Succession & Apply		
No Time Source Destination Protocol Info		
1 0.000000 AmbitMic_a9:3d:68 Broadcast ARP Who has 192.168.1.17 TELL 192.168.1.105		
2 0.001018 LinksysG_da:at:73 AmbitMic_a9:3d:68 ARP 192.168.1.1 is at 00:06:25:da:at:73		
4 2.962850 AmbitMic_a9:3d:68 LinksysG_da:af:73 0X0800 IP		
5 8.971488 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP		
6 13.542974 Telebit_73:80:ce Broadcast ARP Who has 192.168.1.117? Tell 192.168.1.104		
8 17.465902 Linksys_daiaf:73 AmbitMic_a9:3d:68 0x0800 IP		
9 17.465927 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP		
10 17.466468 AmbitM1c_a9:3d:68 LinksysG_da:at:73 0X0800 IP		
12 17.498935 Linksysg_da:af:73 Ambitwic_a9:3d:68 0x0800 IP		
13 17.500025 Linksýsg_da:af:73 AmbitMic_a9:3d:68 0x0800 IP		
14 17.500069 AmbitMic_a9:3d:68 LinksysG_da:at:73 0x0800 IP		
16 17.527622 LinksysG_da:af:73 AmbitMic_39:3d:68 0X0800 IP		
17 17.527457 AmbitMic_a9:3d:68 LinksysG_da:af:73 0x0800 IP		
Packet Length: 42 bytes Canture Length: 42 bytes		
Ethernet II, Src: 00:d0:59:a9:3d:68, Dst: ff:ff:ff:ff:ff		
Destination: ff:ff:ff:ff:ff (Broadcast)		
Source: 00:d0:59:a9:3d:68 (AmbitMic_a9:3d:68)		
IVPE: ARP (UXU806)		
0010 TT TT TT TT TT TT 00 d0 59 a9 36 68 08 06 00 01		
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:

- 12. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?
- 13. Give the hexadecimal value for the two-byte Ethernet Frame type field. What do the bit(s) whose value is 1 mean within the flag field?
- 14. 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) How many bytes from the very beginning of the Ethernet frame does the ARP *opcode* field begin?
 - 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?
 - c) Does the ARP message contain the IP address of the sender?
 - d) Where in the ARP request does the "question" appear the Ethernet address of the machine whose corresponding IP address is being queried?
- 15. 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?
- 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?
- 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?
- 16. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?
- 17. Open the *ethernet--ethereal-trace-* trace file in <u>http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip</u>. The first and second ARP packets in this 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 indiated 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?

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?

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.