

Wireshark Ethernet lab quiz

Due 2 Jun at 17:00 **Points** 13 **Questions** 13
Available until 9 Jun at 17:00 **Time limit** None **Allowed attempts** 5

This quiz was locked 9 Jun at 17:00.

Attempt history

	Attempt	Time	Score
LATEST	Attempt 1	27,181 minutes	13 out of 13

Score for this attempt: **13** out of 13

Submitted 29 May at 20:06

This attempt took 27,181 minutes.

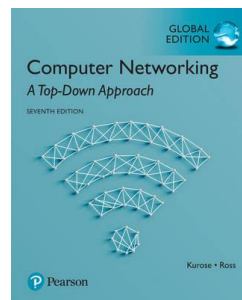
Question 1

1 / 1 pts

Wireshark Lab: Ethernet and ARP v7.0

Adapted from Supplement to
Computer Networking: A Top-Down Approach, 7th 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[1].

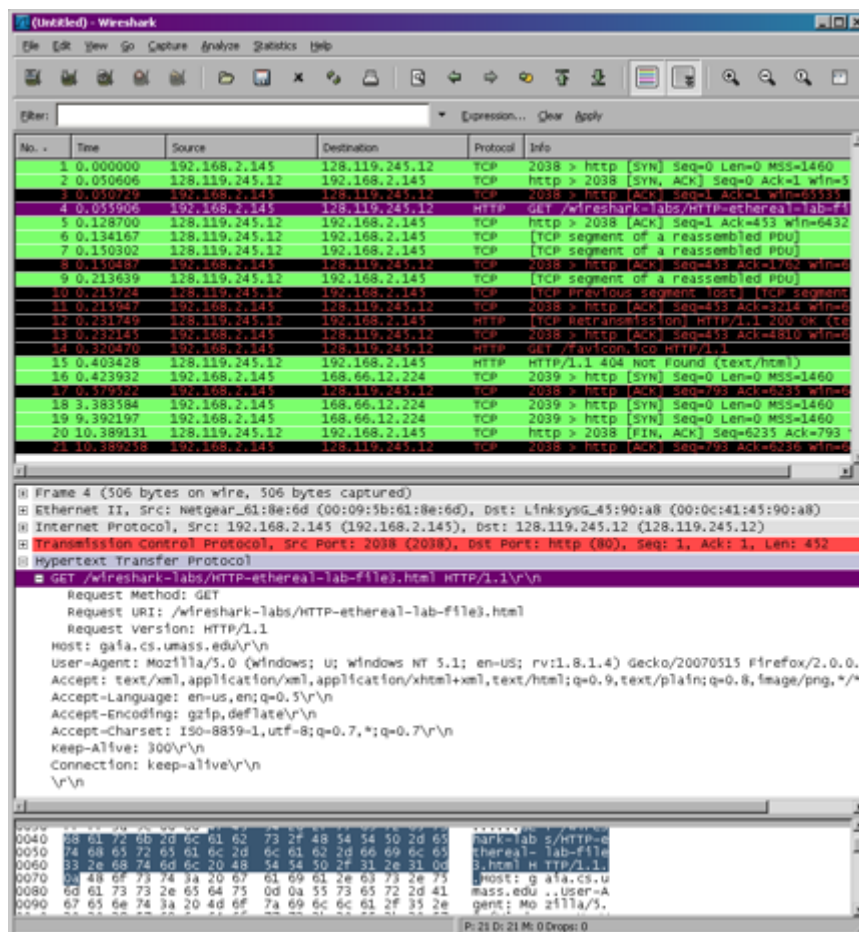
RFC 826 (<http://ftp.rfc-editor.org/in-notes/std/std37.txt> ↗ (<http://ftp.rfc-editor.org/in-notes/std/std37.txt> ↗

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.

1. Capturing and analyzing Ethernet frames

Let's begin by capturing a set of Ethernet frames to study. Do the following[2]:

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



- 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 (IPv4 and IPv6) and select OK. You should now see an Wireshark window that looks like:


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).

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.

Answer the following questions, based on the contents of the Ethernet frame containing the HTTP GET message.

Which of the following are ethernet addresses?

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

[2] 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>  (<http://gaia.cs.umass.edu/ethereal-labs/ethereal-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.

Correct!☐ 192.168.141.22☒ 00:11:24:d9:01:46☐ 00:04:d9:34:zz:72☐ 00:04:ed:42:56**Question 2****1 / 1 pts**

A computer on an Ethernet will always have exactly one Ethernet address

☐ True☒ False**Correct!****Question 3****1 / 1 pts**

"In the frame containing the HTTP GET message, the Ethernet destination is the address of"

☐ gaia.cs.umass.edu☒ the gateway router

correct

☐ the DNS server☐ the proxy web server**Correct!**

Question 4**1 / 1 pts**

What is the hexadecimal value of the 'type' field in the Ethernet frame containing the GET message?

Correct!**Incorrect Answers**

0x0800

800

0800

Question 5**1 / 1 pts**

"The ASCII letter ""G"" of the "GET /wireshark...." appears in the ____th byte of the Ethernet frame."

Correct!**Incorrect Answers**

55

67

54

Question 6**1 / 1 pts**

The last 4 bytes of the Ethernet frame contain

☐ synchronization code☒ CRC☐ Flags**Correct!**

☐ data

Question 7

1 / 1 pts

2. The Address Resolution Protocol

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.

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/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).

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

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

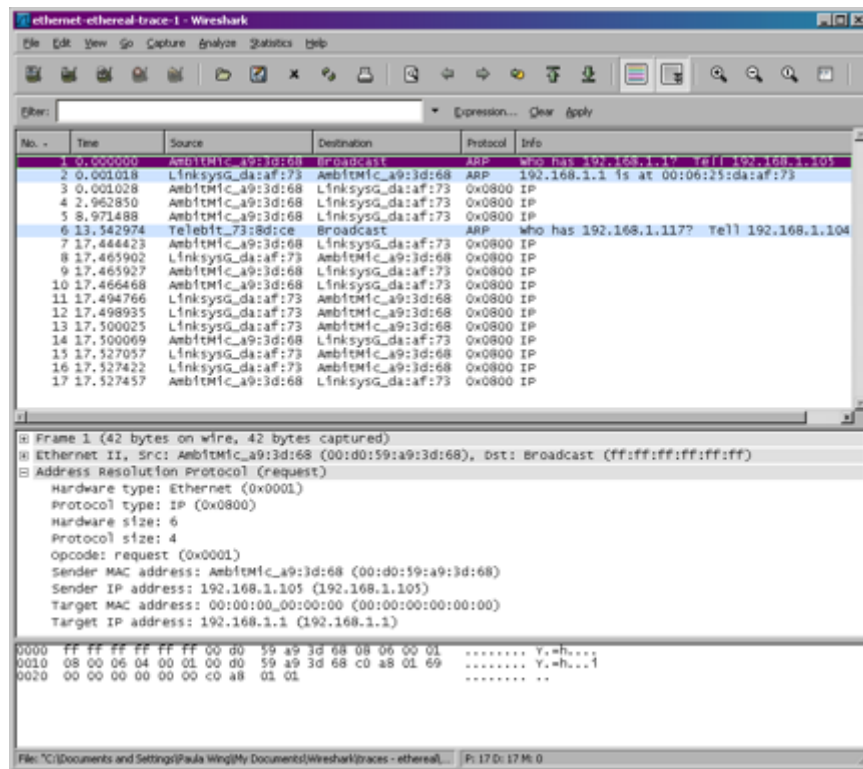
In order to observe your computer sending and receiving ARP messages, we'll need to clear the ARP cache, since otherwise your computer is likely to find a needed IP-Ethernet address translation pair in its cache and consequently not need to send out an ARP message.

- **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/MacOS.** The `arp -d -a` will clear your ARP cache. In order to run this command you'll need root privileges (ie run as `sudo arp -d -a`). If you don't have root privileges, you can skip the trace collection part of this lab and just use the trace discussed in the earlier footnote.

Observing ARP in action

Do the following^[1]:

- Clear your ARP cache, as described above.
- 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*.
- Start up the Wireshark packet sniffer
- 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.
- 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 (IPv4 and IPv6) and select *OK*. You should now see an Wireshark window that looks like:



In the example above, the first two frames in the trace contain ARP messages (as does the 6th message). The screen shot above corresponds to the trace referenced in footnote 1. You can find the ARP messages by putting *arp* into the filter.

Answer the following questions:

What is the Ethernet destination address in the Ethernet frame containing the ARP request message?

[1] The *ethernet-ethereal-trace-1* trace file in <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip> (<http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip>) was created using the steps below (in particular after the ARP cache had been flushed).

Correct!

ff:ff:ff:ff:ff:ff

Correct Answers

ff:ff:ff:ff:ff:ff

Question 8**1 / 1 pts**

What is the hexadecimal value in the Ethernet frame 'type' field for the ARP request?

Correct!**Incorrect Answers**

0x0806

0806

806

Question 9**1 / 1 pts**

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

Correct!**Incorrect Answers**

1

0x0001

Question 10**1 / 1 pts**

What is the hexadecimal value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP reply is made? For this you will need to locate a reply. Look for an ARP response "x.y.z.w is at a:b:c:d:e:f" (ip address is at ethernet address).

Correct!**Incorrect Answers**

0x0002

2

0002

Question 11**1 / 1 pts**

"In the tracefile, there is no reply for the second ARP request (packet 6). This is most likely because"

☐ No host on the network had the requested IP address

☐ The host with that IP address was not responding

☒

The reply was not destined for Broadcast or the host running ethereal

correct

☐ The host running ethereal had cached the answer already

Correct!**Question 12****1 / 1 pts**

The ARP request message contains the values of

☒ the MAC address of the sender

☒ the IP address of the sender

☒ the target IP address

☐ the target MAC address

Correct!**Correct!****Correct!****Question 13****1 / 1 pts**

The ARP reply message contains

Correct!

☒ the IP address of the sender

Correct!

☒ the MAC address of the sender

Correct!

☒ the target IP address

Correct!

☒ the target MAC address

Quiz score: **13** out of 13