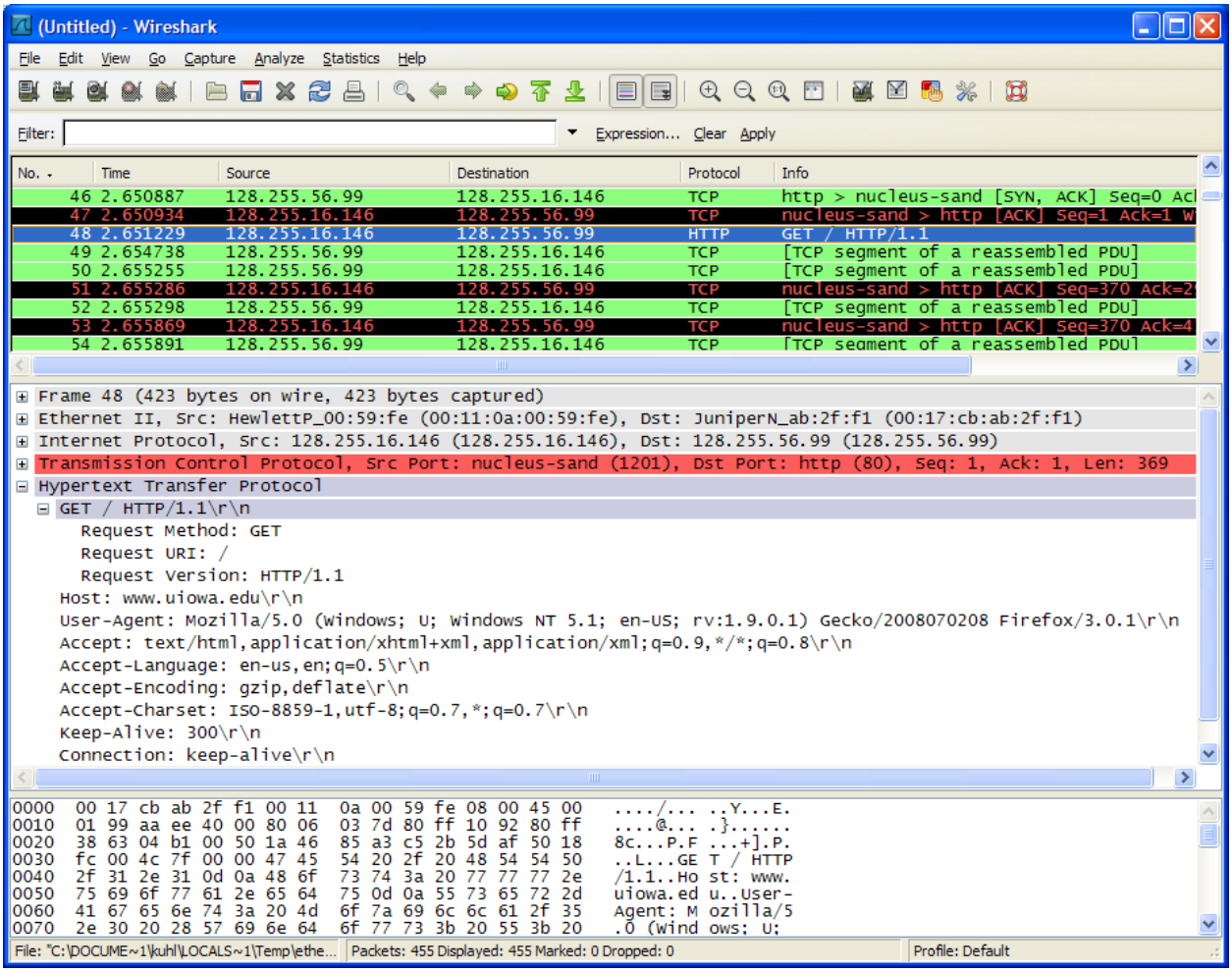
Wireshark Lab 2: Ethernet and ARP

In this lab, we will investigate the Ethernet protocol and the ARP protocol. Before beginning this lab, you should review sections 5.5 (Ethernet), 5.4.1 (link layer addressing) and 5.4.2 (ARP) in the text. RFC 826 (ftp://ftp.rfc-editor.org/innotes/std/std37.txt) contains the gory details of the ARP protocol, which is used by an IP device to determine the link layer address of a remote interface (on the same subnet) whose IP address is known.

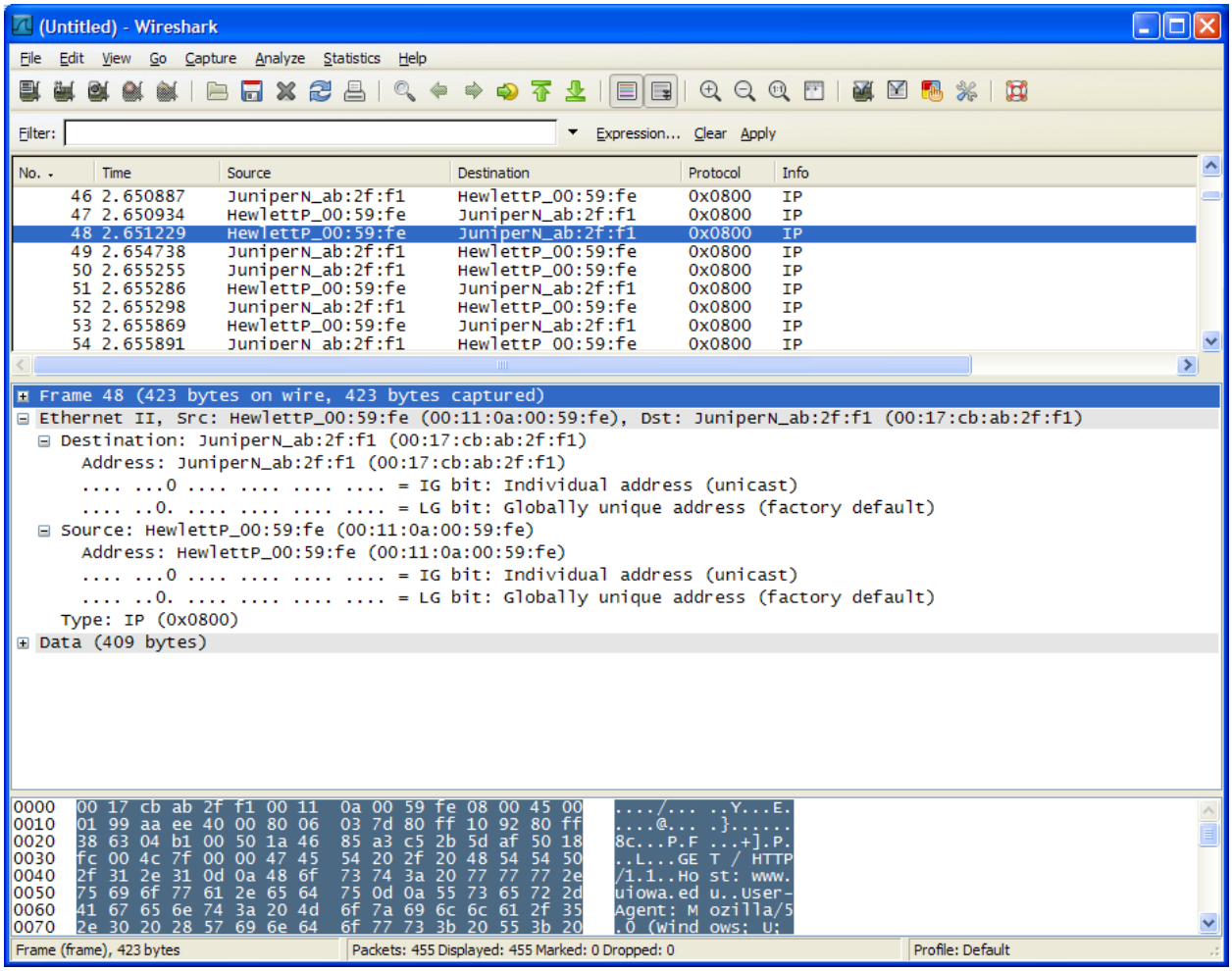
1. Capturing and analyzing Ethernet frames

Begin by capturing a set of Ethernet frames to study. Do the following:

- 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.* For Firefox select *Tools->Clear Private Data.*
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser: http://www.uiowa.edu
- Your browser should display the main University of Iowa web page..
- Stop Wireshark packet capture.
- 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 www.uiowa.edu. You should see a screen that looks something like this (where frame 48 in the screen shot below contains the HTTP GET message)



Since this lab is about Ethernet and ARP, we are 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 do this, select *Analyze->Enabled Protocols.* Then uncheck the IP box and select *OK*. You should now see a Wireshark window that looks like:



In order to answer the following questions, you will 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. 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 www.uiowa.edu? If not, what device has this as its Ethernet address?
3 Give the hexadecimal value for the two-byte Frame type field. What type of frame does this specify?
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 www.uiowa.edu? If not, 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 type of frame does this specify?
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 will observe the ARP protocol in action.

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 Windows 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 is 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.

Take a look at the contents of the ARP cache on your computer:

- **Windows.** The *arp* command is in c:\windows\system32, so type either "*arp - a*" or "*c:\windows\system32\arp -a*" in the Windows 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 –a* command will display the contents of the ARP cache on your computer.

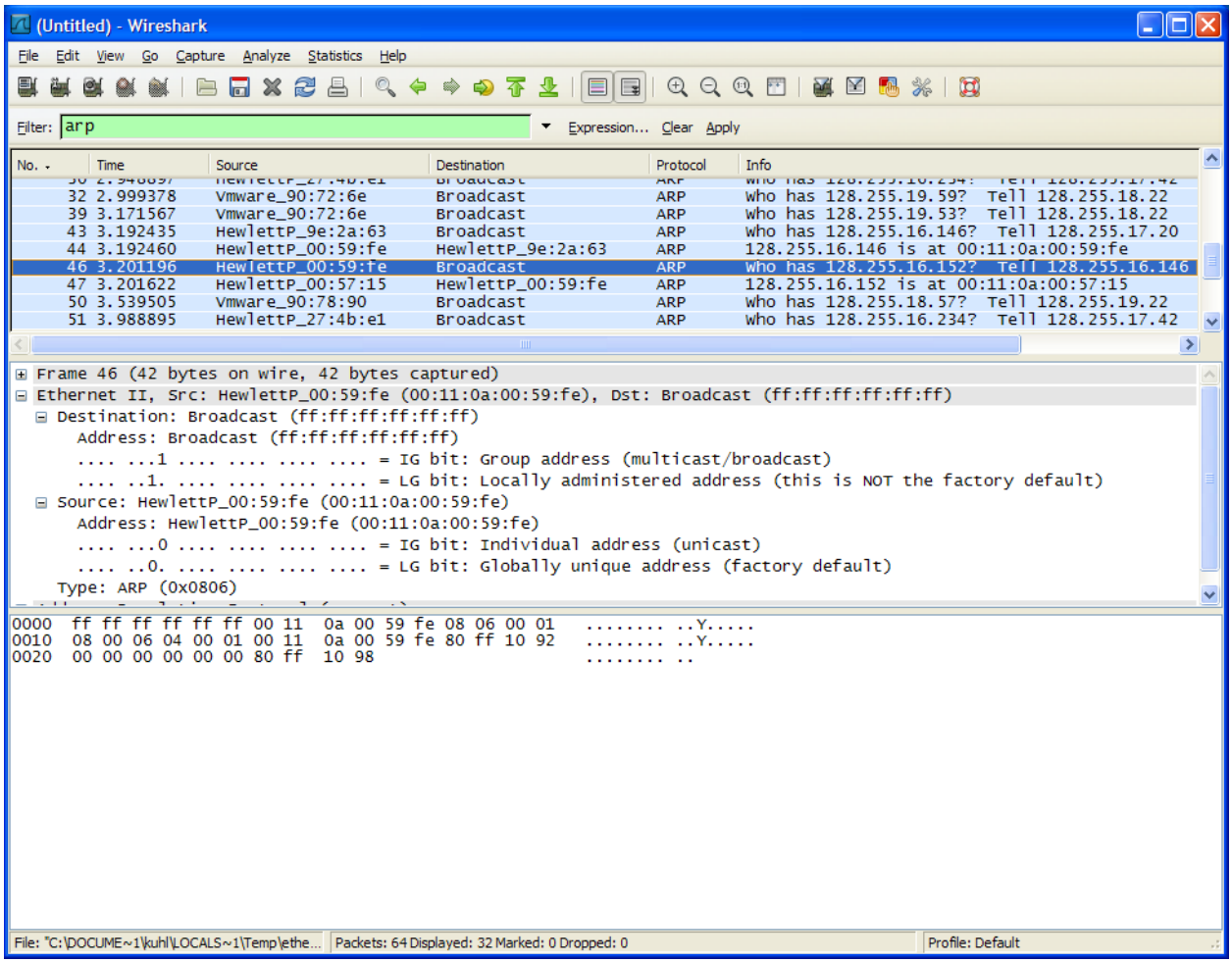
11. Write down the contents of your computer's ARP cache. What is the meaning of each column value?

In order to observe your computer sending and receiving ARP messages, you will 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. The Windows command "arp –d *" will clear the ARP cache. However, this command requires administrator privileges on the computers in the lab. So, instead, simply restart your computer. After restarting the computer, do another "arp –a" command. You will note that there are already a number of entries in the ARP cache. These are the Ethernet addresses of switches and servers with which your computer communicated during the start-up process

Observing ARP in action

Do the following:

- Restart your computer to clear the ARP cache, as described above.
- Start up the Wireshark packet sniffer.
- From the Windows command line, *ping* one of the other computers in the lab. (The Windows computers in 2245 SC have hostnames l-ece000 – l-ece007). Make sure that the computer you are attempting to ping is turned on (you do not need to be logged into it.). Since all of the computers in 2245 SC are on the same logical subnet, the ping request will generate direct frame transfers from your computer to the one being pinged. This will require your computer to issue an ARP request to determine the Ethernet address of the computer you are pinging, assuming that the information is not already present in your computer's ARP cache.
- 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. You should now see an Wireshark window that looks like:



In the example above, the Wireshark "arp" filter has been applied to display only ARP packets.

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 type of frame does this specify?

14 Download the ARP specification from ftp://ftp.rfc-editor.org/innotes/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.

   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 – i.e. the IP address of the machine whose corresponding Ethernet address is being sought?

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 – i.e. the Ethernet address of the machine with the IP address specified in the ARP request?
16. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?
17. 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). Indicate how you determined this value.