Lab 3: Link Layer

Objective

In this lab, you will investigate Ethernet and the ARP protocol. You will also prove you are a Wireshark Ninja by dissecting an unknown protocol. Knowledge from Lecture 20 and textbook sections 5.4 and 5.5 will be key. RFC 826 contains all the interesting details of the ARP protocol.

Administrative Details

Turn in a pdf file to Gradescope by the deadline. No late work will be accepted.

Obviously, it would be better for you to accomplish the lab prior to the Final Exam, as it will help you understand topics that you may be asked about on the Final. However, if you are looking at this in the early hours of the night/morning before the Final, please put it aside and get some sleep -- in such a case, completing it probably won't help any more than not getting a good night's sleep will hurt.

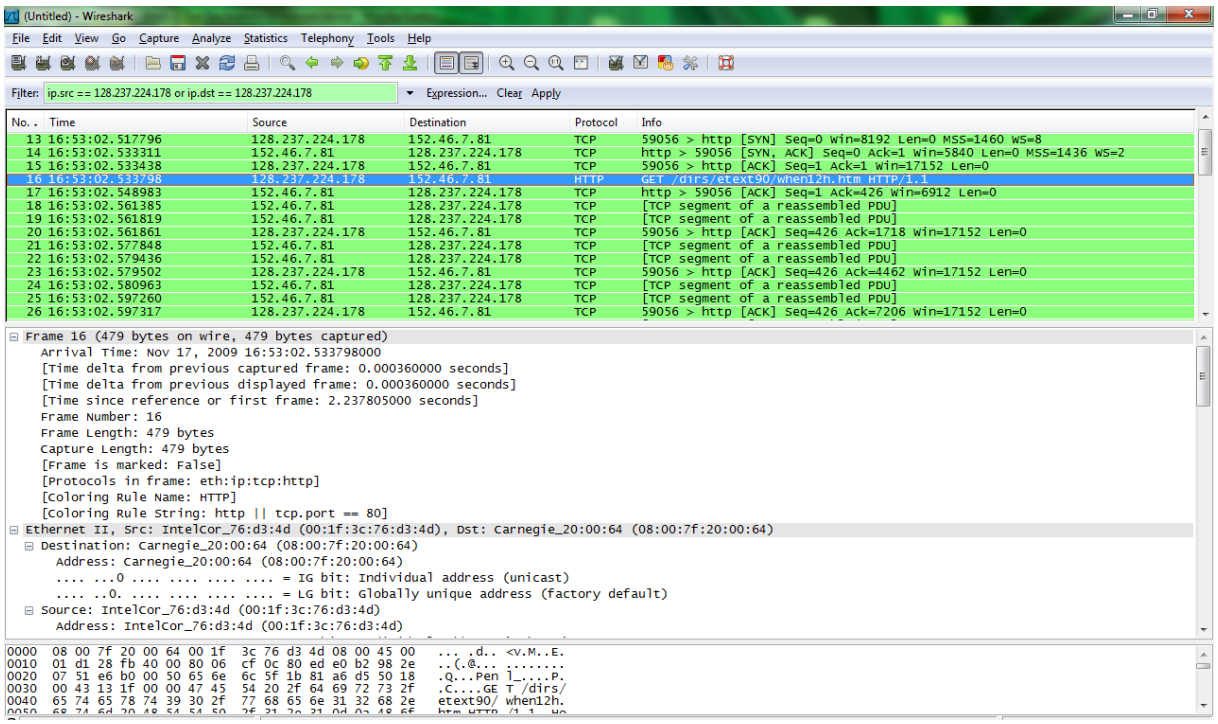
Capturing and Analyzing Ethernet Frames

Let's begin by capturing a set of Ethernet frames to study. It should go without saying that your computer should be hooked up to an Ethernet or WiFi connection for this exercise. Do the following:

- First, make sure your browser's cache is empty. In Safari, type ⌥⌘E. In Firefox, I believe it is in *Tools* → *Clear Private Data*. For Internet Explorer, it is *Start* → *Control Panel* → *Add/Remove Programs* → *Remove Internet Explorer* (just kidding!)
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser

  http://www.ietf.org/rfc/rfc894.txt
- Your browser should display the RFC for IP over Ethernet.
- 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 www.ietf.org, as well as the beginning of the HTTP response message sent to your computer by www.ietf.org. You should see a screen that looks something like this (where packet 4 in the screen shot below contains the HTTP GET message).

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). 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.

1. [4 points] What is the 48-bit Ethernet address of your computer? Where did you find this information?
2. [4 points] What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of www.ietf.org? (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 section 5.4.1 in the text and make sure you understand the answer here.] Again, describe where you got this information.
3. [4 points] Who is the manufacturer of the Ethernet adapter (or computer) for both the source and destination Ethernet adapters. Determine this from the Ethernet address, not by looking at the label on your laptop. Use the document at standards.ieee.org/regauth/oui/index.shtml to determine this, don't just rely on Wireshark.
4. [4 points] Give the hexadecimal value for the two-byte TYPE field. Interpret the meaning of this value.
5. [4 points] How many bytes from the very start of the Ethernet frame does the ASCII "T" in "GET" appear in the 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. [4 points] What is the value of the Ethernet source address? Is this the address of your computer, or of www.ietf.org? (Hint: the answer is no). What device has this as its Ethernet address?

7. [4 points] What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

8. [4 points] Give the hexadecimal value for the two-byte Frame type field. Interpret the meaning of this value.

9. [4 points] 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?

The Address Resolution Protocol

In this section, you will observe ARP in action. I 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 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. The *arp* command will display the contents of the ARP cache on your computer. For Windows[1] and Mac OSX, use *arp -a*, for some variants of Linux, just use *arp*. Run the *arp* or *arp -a* command and keep a copy of the contents of your computer's ARP cache.

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. The -d flag to the arp command deletes an entry from the ARP cache. Use *arp -ad* to clear all entries from the cache.

Observing ARP in action

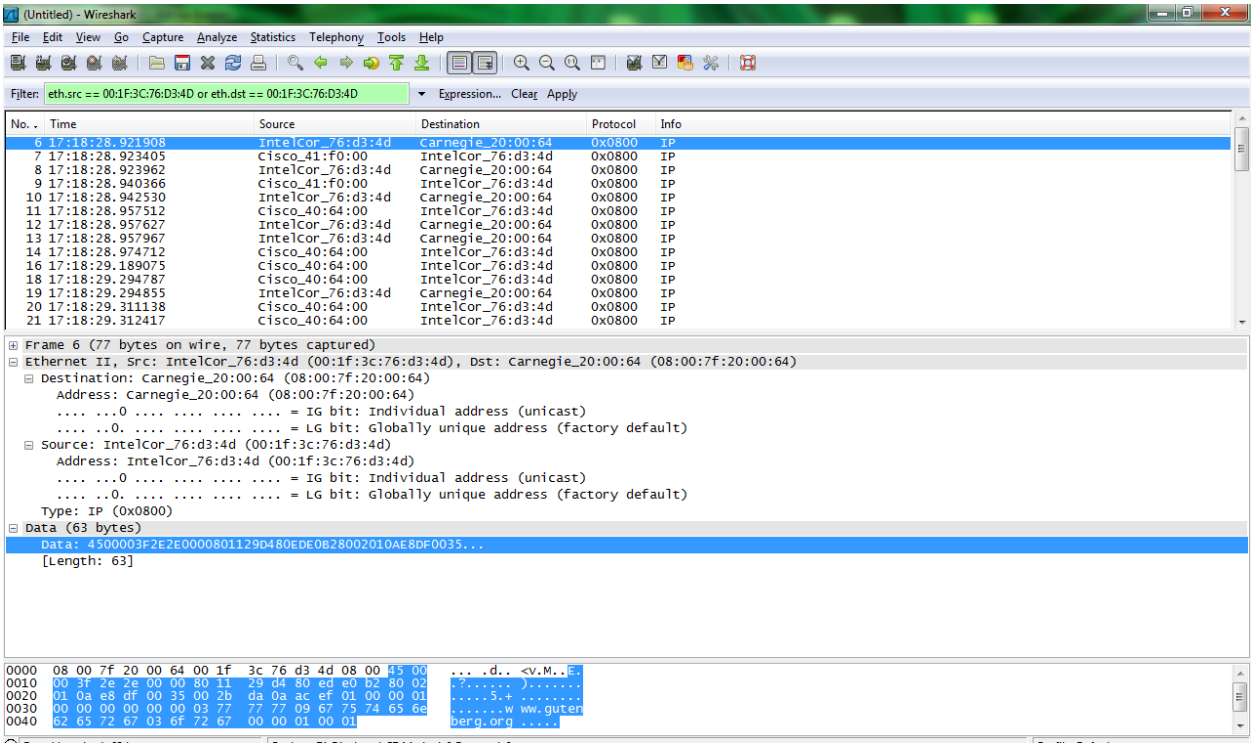
Do the following:

- Clear your ARP cache, as described above.
- Next, make sure your browser's cache is empty.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser

  http://www.ietf.org/rfc/rfc826.txt
- Your browser should display the RFC for ARP.
- Stop Wireshark packet capture. To get rid of other packets, you can filter on your ip address with ip.addr == <ip> or use the eth.add == <mac> to filter on ethernet frames.

You should now see an Wireshark window that looks like:

[1] Apparently, on Windows 7+, you need to use the netsh command instead. I don't have a machine to test on but it's something like: **`netsh interface ip add neighbors "Local Area Connection" <ip address> <MAC address>`**



In the example above, the first two frames in the trace contain ARP messages (as does the 6th message).

Answer the following questions:

10. [4 points] What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?

11. [4 points] Give the hexadecimal value for the two-byte Ethernet Frame type field. How is this value different than you saw in questions 4 and 8?

12. You may need to refer to the ARP specification (in the RFC you just downloaded)x. A readable, detailed discussion of ARP is also at http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html.

- **a.** [2 points] How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?
- **b.** [2 points] What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP request is made? What does it mean?
- **c.** [2 points] Does the ARP message contain the IP address of the sender? Why or why not?
- **d.** [2 points] Where in the ARP request does the "question" appear – the Ethernet address of the machine whose corresponding IP address is being queried?

13. Now find the ARP reply that was sent in response to the ARP request.

- **a.** [2 points] How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

b. [2 points] What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP response is made? What does it mean?

c. [2 points] Where in the ARP message does the "answer" to the earlier ARP request appear – the MAC address of the machine IP address is being queried?

14. [4 points] What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?

15. [12 points] The arp command:

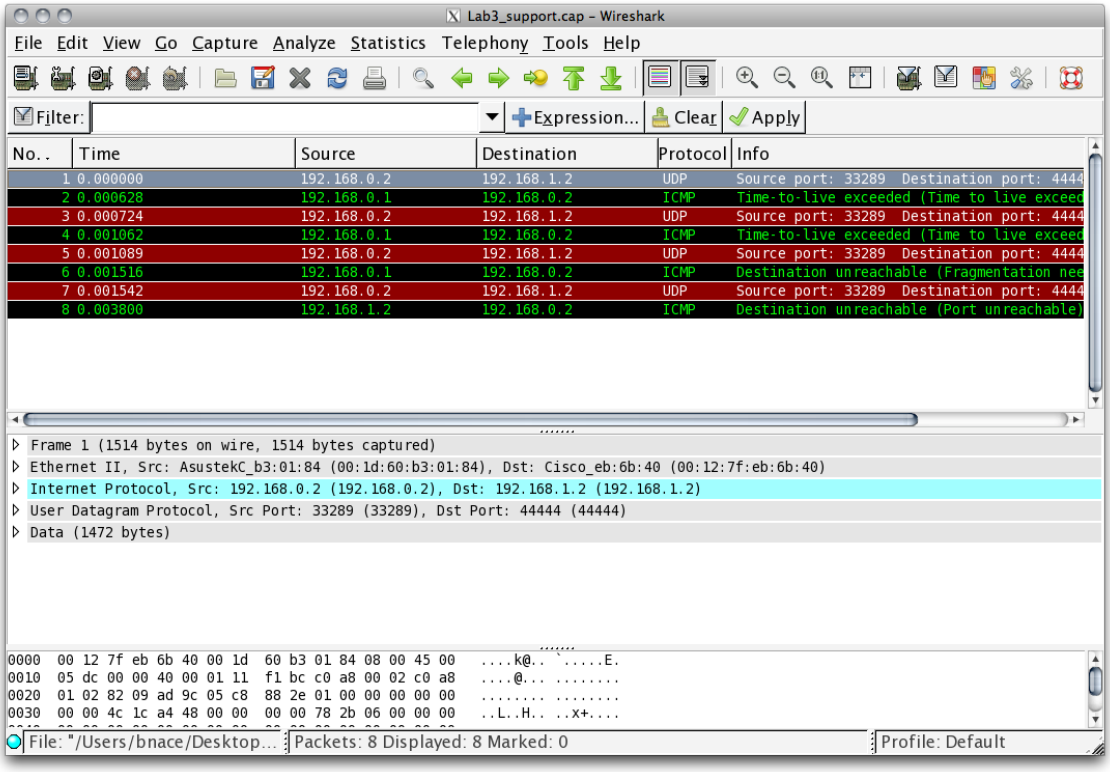
arp -s InetAddr EtherAddr

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? Try it and report on your findings.

16. [6 points] 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.

Hmmmm...

Download the Lab3_support.cap file from the course website. This file is a Wireshark capture file with some interesting data in it. Open it in Wireshark and take a look.



There are only 8 packets here, so this shouldn't take too long to examine them and figure out what is going on.

17. [20 points] For each of the packets, write a short description of what the *purpose* of the packet is. Back your assertion up with data from the packet. List anything else interesting in the packet. I'm not looking for a straight recital of what the packet contents are, I'm looking for the deeper meaning behind the packet. So, don't say "This is a UDP packet sent to port 67." Instead say "This is the DHCP OFFER message from the server. You can see the XID field is the same as the DHCP DISCOVER message..."

If you've gotten inspired to use Wireshark to dig into the protocols you've learned about in class, you might want to check out www.packetlife.net/captures and wiki.wireshark.org/SampleCaptures for some sample capture files that you can dissect.

Course Evaluations

I value your honest and critical evaluation of the course. While I'd love for your experience in the course to have been absolutely perfect in every way, I know realistically that didn't happen. Please help me improve the course and the experience for future students by filling out a course evaluation.

While the numbers in the course evaluation are helpful, the really good information that will help me improve the most is to be found in comments. **If there is any question on the course evaluation that you have not given me perfect marks for, please leave me a comment explaining what I could do to improve and earn perfect marks.**

If you attach the printout from the end of the evaluation to this lab, proving that you filled out a course evaluation for this course (but without your answers, obviously), then you will receive extra credit worth 25% of a lab grade.

If you also attach the following signed statement (obviously, which should be true. Don't sign if you didn't provide a useful evaluation), you will receive an additional 25% extra credit.

> I HAVE PROVIDED A USEFUL COURSE EVALUATION. FOR EACH QUESTION ON THE COURSE EVALUATION, I EITHER GAVE PERFECT MARKS OR LEFT A HELPFUL COMMENT EXPLAINING WHAT COULD BE DONE TO IMPROVE THE COURSE TO THAT POINT.
>
> _____
>
> SIGNED NAME AND DATE