New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Course Title: Data Communication & Networks
Course Number: CSCI-GA.2662-001
Instructor: Jean-Claude Franchitti
Session: 4

Lab #4 – Link Layer

I. Due

Thursday October 14, 2021 at the beginning of class.

II. Objectives

1. Understand the basic principles of the link layer and related protocols; in particular link layer addressing, ARP, and Ethernet.
2. Understand how to analyze link layer protocols

III. References

1. Slides and handouts posted on the course Web site
2. Textbook chapters as applicable
3. RFC 826 (ftp://ftp.rfc-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

IV. Software Required

1. Microsoft Word.
2. Win Zip as necessary.
3. Wireshark.

V. Lab Instructions

1. Problem 1 – Capturing and Analyzing Ethernet Frames:

Let's begin by capturing a set of Ethernet frames to study. Do the following[1]:

[1] If you are unable to run Wireshark live on a computer, you can extract the file *ethernet--ethereal-trace-1* from the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip. The traces in that file were collected by Wireshark while performing the steps indicated in this lab. You can load the file 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 lab questions.

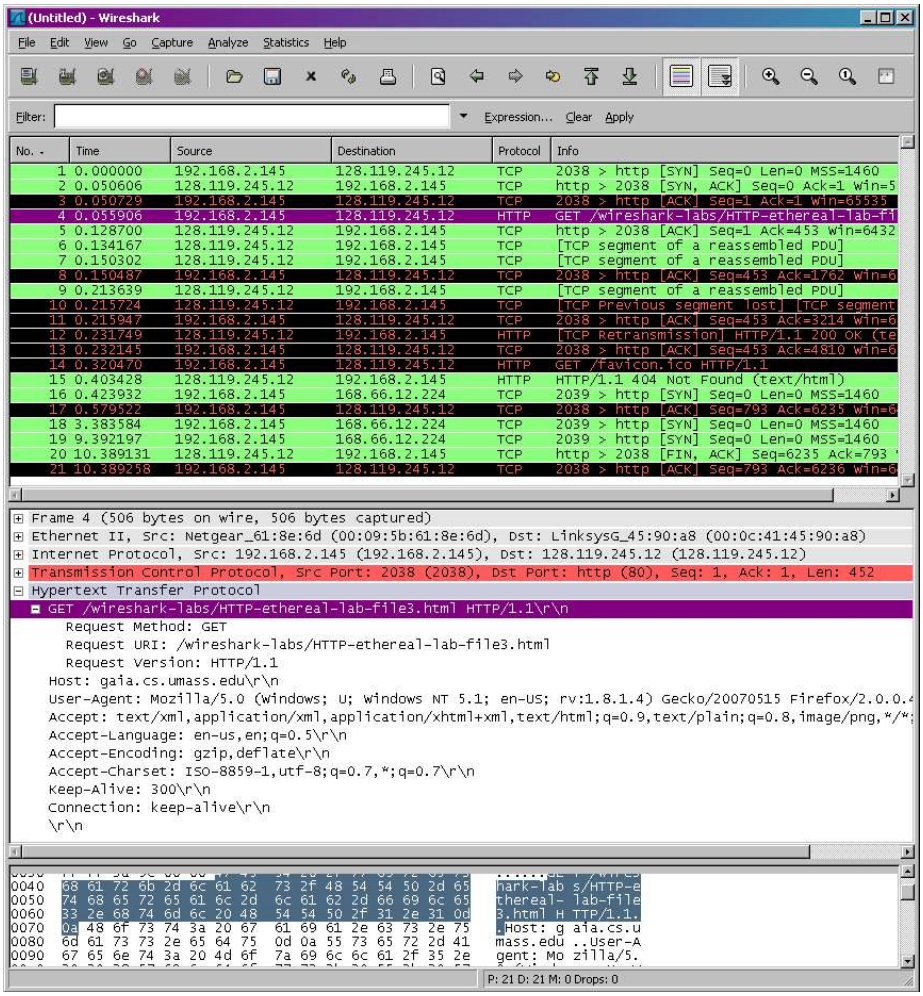
- First, make sure your browser's cache is empty. To do this under Mozilla Firefox V3, select *Tools->Clear Recent History* and check the box for Cache. For Internet Explorer, select *Tools->Internet Options->Delete Files.* Start up the Wireshark packet sniffer

- Enter the following URL into your browser:

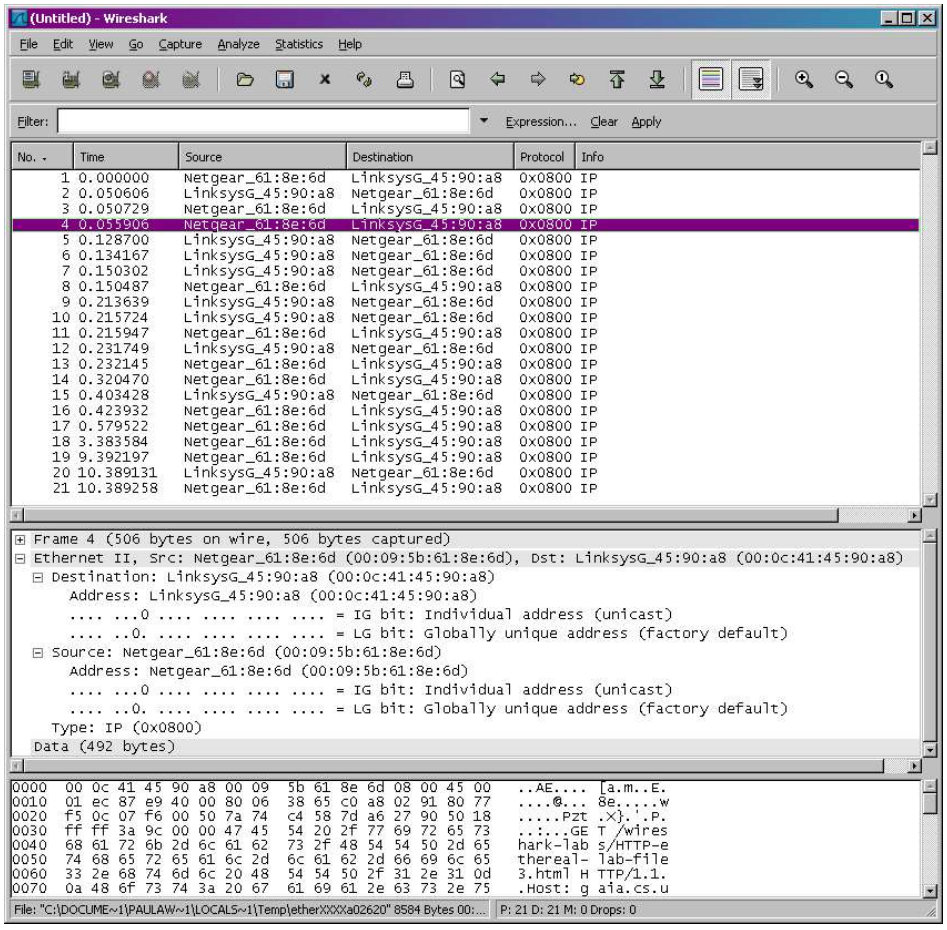
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-ethereal-lab-file3.html

  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 are not interested in IP or higher-layer protocols. So let us 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 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 encapsulated 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 your electronic submission[2] 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?
3. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?
4. How many bytes from the very start of the Ethernet frame does the ASCII "G" 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:

5. 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?
6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?
7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?
8. 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?

2. Problem 2 – Observing 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 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:

[2] What do we mean by "annotate"? Please highlight where in the electronic copy you submit you have found the answer and add some text (preferably in color) noting what you found in what you have highlighted.

- **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. Run the *arp* command.

9. 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 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.

- **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* * 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 Wireshark on a Windows machine, you can skip the trace collection part of this lab and just use the trace discussed in the earlier footnote.

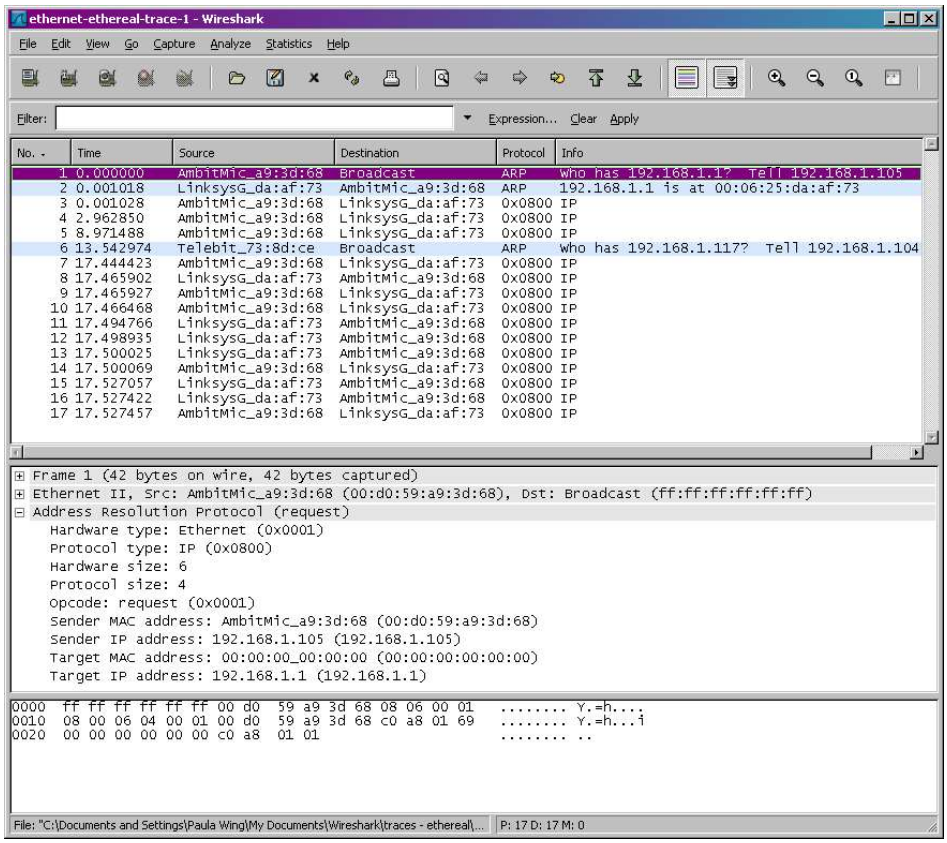
Observing ARP in action

Do the following[3]:

- 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

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

Analyze->Enabled Protocols. Then uncheck the IP box 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.

Answer the following questions:

1. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?
2. Give the hexadecimal value for the two-byte Ethernet Frame type field. What upper layer protocol does this correspond to?
3. Download the ARP specification from:
   ftp://ftp.rfc-editor.org/in-notes/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 – the Ethernet address of the machine whose corresponding IP address is being queried?

4. 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?
5. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?
6. Open the *ethernet-ethereal-trace-1* trace file in:
   http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip
   The first and second ARP packets in this trace correspond to an ARP request sent by the computer running Wireshark, and the ARP reply sent to the computer running Wireshark 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?

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.

VI. Deliverables

1. Electronic:

   Your lab assignment files must be submitted via NYU Brightspace.

   For part V.1, your lab assignment file must be submitted via NYU Brightspace. Name the file "**firstname_lastname_lab_#.docx**" (e.g., "john_doe_lab_4.docx").

The file must be created and sent by the beginning of class. After the class period, the homework is late. The email clock is the official clock.

2. Cover page and other formatting requirements:

   The cover page supplied on the next page must be the first page of your lab assignment file.

   Fill in the blank area for each field.

   NOTE:

   The sequence of the electronic submission is:

   1. **Cover sheet**
   2. **Lab Assignment Answer Sheet(s)**

3. Grading guidelines:

   Assignment Layout (15%)
   - o Lab Assignment is neatly assembled on 8 1/2 by 11 layout
   - o Cover page with your name (last name first followed by a comma then first name), username and section number with a signed statement of independent effort is included.
   - o File name is correct.

   Answers to Individual Questions (85%):
   - o Answers to all questions are complete and correct.
   - o Assumptions provided as required.

   (100 points total, all questions weighted equally)

VII. Sample Cover Sheet:

Name _____ Date: _____
(last name, first name)
Section: _____

Lab #4 – Link Layer

Total in points (100 points total): _____

Professor's Comments:

Affirmation of my Independent Effort: _____
(Sign here)