Wireshark - Ethernet - 19
(gdocs source)

This Lab is a combination of:
Wireshark Lab: Ethernet & Arp by KR
Erlinger's old Ethernet lab.

1. Capturing and analyzing Ethernet frames

Let's 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 Mozilla Firefox V3, select *Tools->Clear Recent History 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)***
- **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 contains the HTTP GET message.

Since this lab is about Ethernet, we are 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 and select *OK*. You should now see a different Wireshark window.

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. 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 (see labfaq for details) 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?**
3. **Is this the Ethernet address of gaia.cs.umass.edu? (Hint: the answer is *no*).**
4. **What device has this as its Ethernet address? [Note: this is an important question, and one that students sometimes get wrong. Re-read pages 468-469 in the text and make sure you understand the answer here.]**
5. **Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?**
6. **How many bytes from the very start of the Ethernet frame does the ASCII "G" in "GET" appear in the Ethernet frame?**
7. **What is the meaning of any 1 bits in the Flag Field.**
8. **So which C struct below would you use in dealing with an Ethernet header and why? (If you do not like any of them, create one and justify yours)**

struct 1

```
 struct ethhdr {
   unsigned int    h_dest[ETH_ALEN];   /* destination eth addr */
   unsigned int    h_source[ETH_ALEN]; /* source ether addr    */
   unsigned int    h_proto;        /* packet type ID field */
    }     ;
```

Struct 2

```
struct ethhdr {
   unsigned char   h_dest[ETH_ALEN];   /* destination eth addr */
   unsigned char   h_source[ETH_ALEN]; /* source ether addr    */
   unsigned short  h_proto;        /* packet type ID field */
 };
```

Struct 3

```
struct ethhdr {
   unsigned short  h_dest[ETH_ALEN];   /* destination eth addr */
```

```
  unsigned long   h_source[ETH_ALEN]; /* source ether addr    */
  unsigned int    h_proto;        /* packet type ID field */
};
```

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

7. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu. What device has this as its Ethernet address?

8. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

9. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

10. 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?

11. How much time did you spend on this lab.