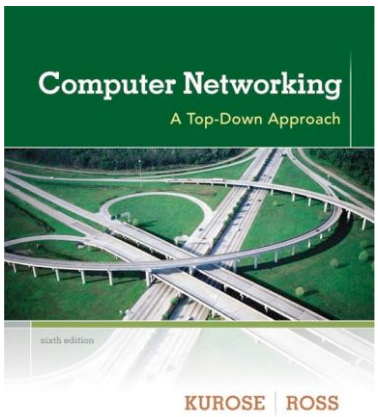
Assigned: Tuesday, July 11th, 2017
Lab2 has two parts DHCP and ARP.
Both parts are due on Thursday, July 20th at 11:59 PM (DEN Submission Link)
Late submissions are accepted for two days only, with a maximum penalty of 15% per day. For each day, submissions between 12-1am: 3%, 1-2 am: 7%, 2-3 am: 12%, and after 3am: 15%.

Part 1: DHCP

Wireshark Lab: DHCP v6.0

Supplement to *Computer Networking: A Top-Down Approach, 6th 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

---

In this lab, we'll take a quick look at DHCP. DHCP is covered in Section 4.4.2 of the text[1]. Recall that DHCP is used extensively in corporate, university and home-network wired and wireless LANs to dynamically assign IP addresses to hosts (as well as to configure other network configuration information).

This lab is brief, as we'll only examine the DHCP packets captured by a host. If you also have administrative access to your DHCP server, you may want to repeat this lab after making some configuration changes (such as the lease time). If you have a router at home, you most likely can configure your DHCP server. Because many linux/Unix machines (especially those that serve many users) have a static IP address and because manipulating DHCP on such machines typically requires super-user privileges, we'll only present a Windows version of this lab below.

[1] References to figures and sections are for the 6th edition of our text, *Computer Networks, A Top-down Approach, 6th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2012.*

DHCP Experiment

In order to observe DHCP in action, we'll perform several DHCP-related commands and capture the DHCP messages exchanged as a result of executing these commands. Do the following[2]:

1. Begin by opening the Windows Command Prompt application (which can be found in your Accessories folder). As shown in Figure 1, enter "*ipconfig /release*". The executable for *ipconfig* is in C:\windows\system32. This command releases your current IP address, so that your host's IP address becomes 0.0.0.0.
2. Start up the Wireshark packet sniffer, as described in the introductory Wireshark lab and begin Wireshark packet capture.
3. Now go back to the Windows Command Prompt and enter "*ipconfig /renew*". This instructs your host to obtain a network configuration, including a new IP address. In Figure 1, the host obtains the IP address 192.168.1.101
4. Wait until the "*ipconfig /renew*" has terminated. Then enter the same command "*ipconfig /renew*" again.
5. When the second *"ipconfig /renew"* terminates, enter the command "ipconfig/release" to release the previously-allocated IP address to your computer.
6. Finally, enter "*ipconfig /renew*" to again be allocated an IP address for your computer.
7. Stop Wireshark packet capture.

[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 and extract the file *dhcp-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 dhcp-ethereal-trace-1 trace file. You can then use this trace file to answer the questions in DHCP section.

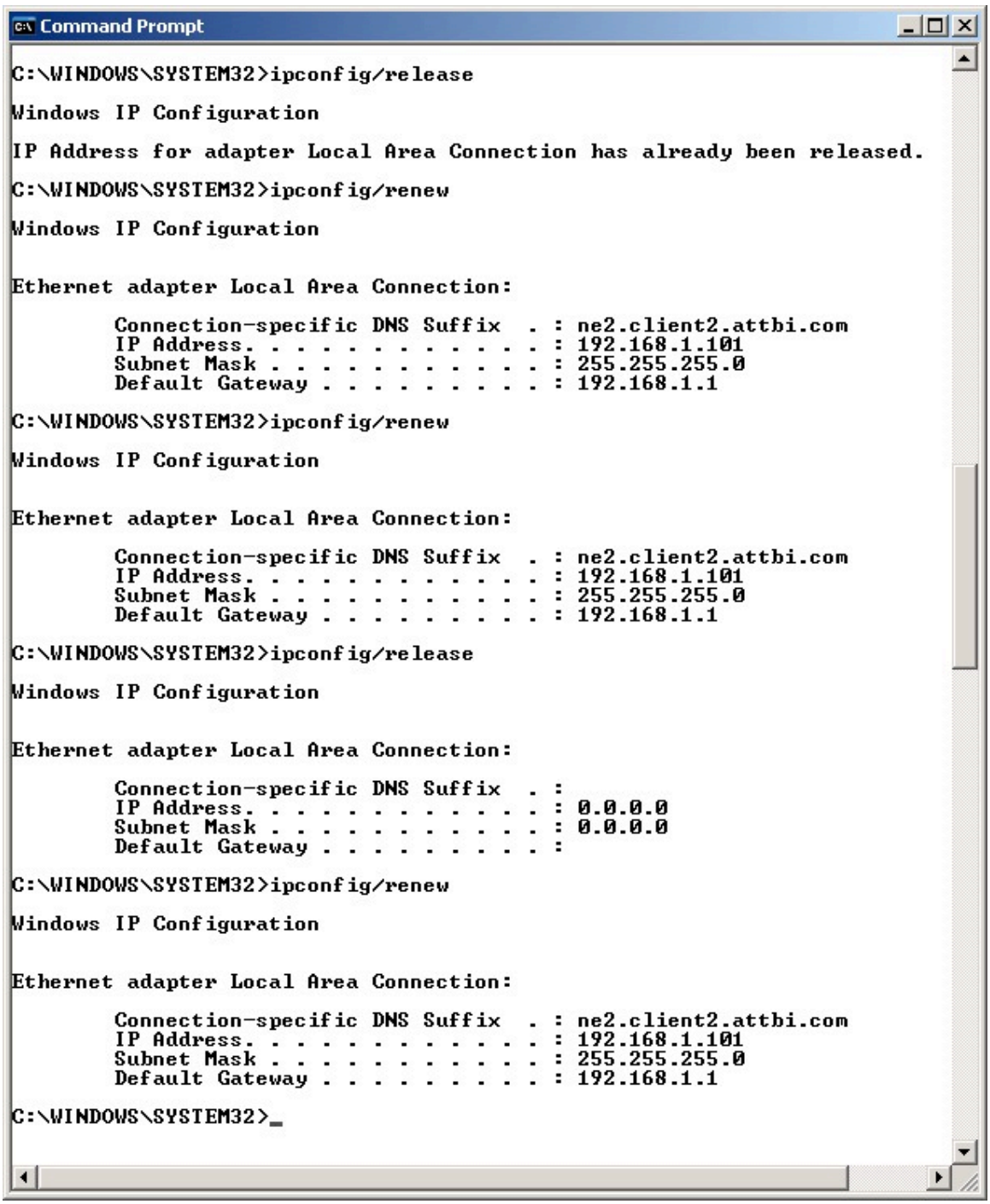


Figure 1 Command Prompt window showing sequence of *ipconfig* commands that you should enter.

Now let's take a look at the resulting Wireshark window. To see only the DHCP packets, enter into the filter field "bootp". (DHCP derives from an older protocol called BOOTP. Both BOOTP and DHCP use the same port numbers, 67 and 68. To see DHCP packets in the current version of Wireshark, you need to enter "bootp" and not "dhcp" in the filter.) We see from Figure 2 that the first *ipconfig* renew command caused four DHCP packets to be generated: a DHCP Discover packet, a DHCP Offer packet, a DHCP Request packet, and a DHCP ACK packet.

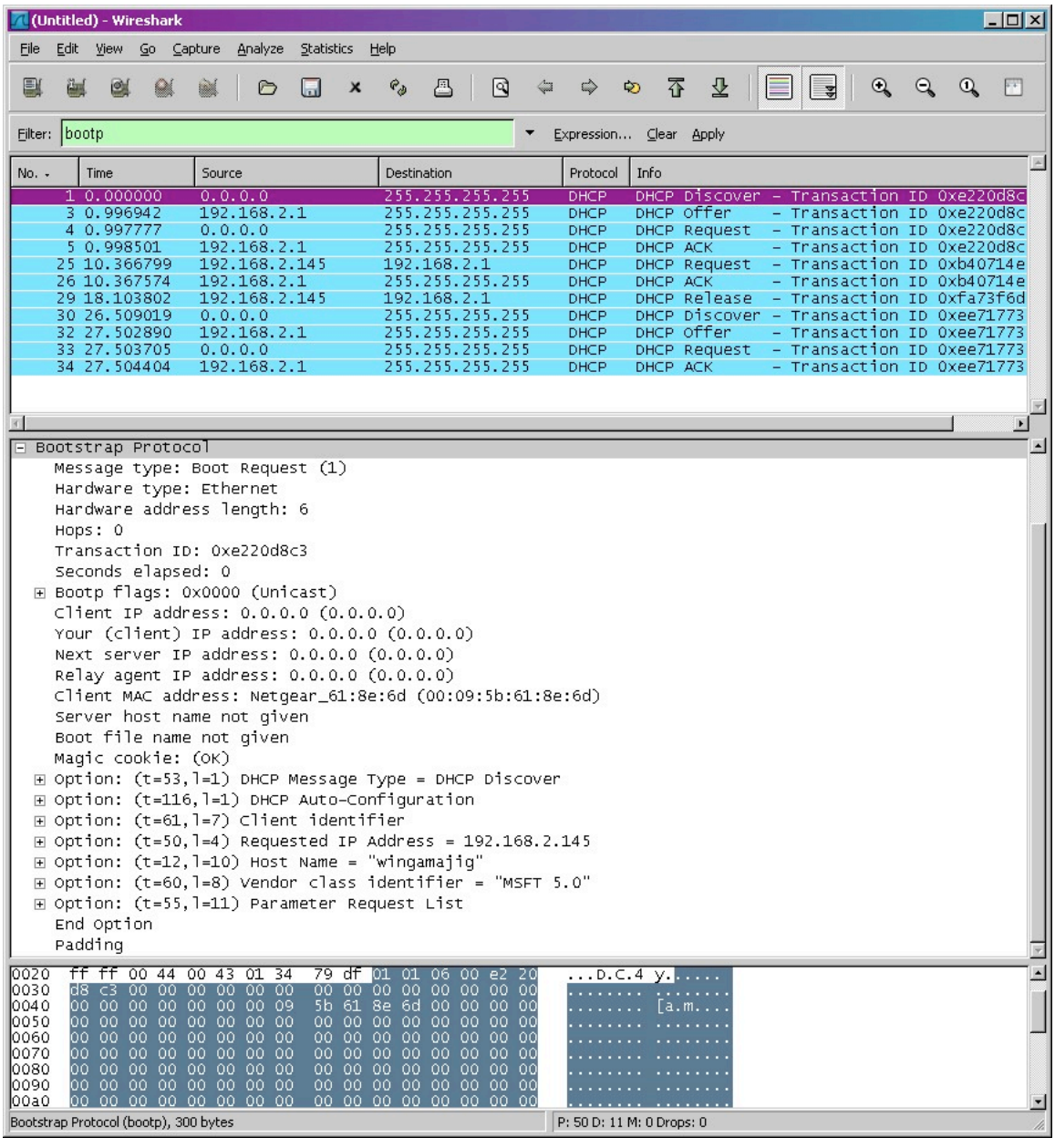


Figure 2 Wireshark window with first DHCP packet – the DHCP Discover packet – expanded.

What to Hand In:

You should hand in a screen shot of the Command Prompt window similar to Figure 1 above. Whenever possible, when answering a question below, you should hand in a screenshot of the packet(s) within the trace that you used to answer the question asked. Annotate the screenshot[3] 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.

Answer the following questions:

1. Are DHCP messages sent over TCP or UDP? Provide a snapshot.
2. Does DHCP use client-server or peer to peer architecture? No snapshot needed.
3. Draw a timing datagram illustrating the sequence of the first four-packet Discover/Offer/Request/ACK DHCP exchange between the client and server. For each packet, indicated the source and destination port numbers and IP addresses.
4. What is the link-layer (e.g., Ethernet) address of your host in hex format?
5. What values in the DHCP Discover message differentiate this message from the DHCP Request message?
6. What is the value of the Transaction-ID in each of the first four DHCP messages? What are the values of the Transaction-ID in the second set (Request/ACK) set of DHCP messages? Why do we need the Transaction-ID field?
7. A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the first four DHCP messages (Discover/Offer/Request/ACK DHCP), indicate the source and destination IP addresses that are carried in the encapsulating IP packet.
8. What is the IP address of your DHCP server?
9. What IP address is the DHCP server offering to your host in the DHCP Offer message and what is the lease time? Which DHCP messages have this IP Address in them?
10. Apart from IP Address, what other information does DHCP server provide to the client?
11. In the example screenshot in this assignment, there is no relay agent between the host and the DHCP server. What values in the trace indicate the absence of a relay agent? Is there a relay agent in your experiment? If so what is the IP address of the agent?

---

[3] What do we mean by "annotate"? In an electronic copy, you should highlight the information in the snapshot which helped you to get the answer.

[4] 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 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 in ARP section.

12. Explain the purpose of the router and subnet mask lines in the DHCP offer message and indicate the IP address of the default gateway (router).
13. In the client's response to the first server DHCP Offer message, does the client accept this IP address? Where in the DHCP Request is the client's requested IP address?
14. What is the purpose of the DHCP Release message? Does the DHCP server issue an acknowledgment of receipt of the client's DHCP Release message? What would happen if the client's DHCP Release message is lost?
15. Clear the *bootp* filter from your Wireshark window. Were any ARP packets sent or received during the DHCP packet-exchange period? If so, explain the purpose of those ARP packets.

Part 2: Address Resolution Protocol

In this section, we'll observe the ARP protocol in action. We strongly recommend that you re-read section 5.4.1 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.** 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* command with no arguments will display the contents of the ARP cache on your computer. Run the *arp* command.

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

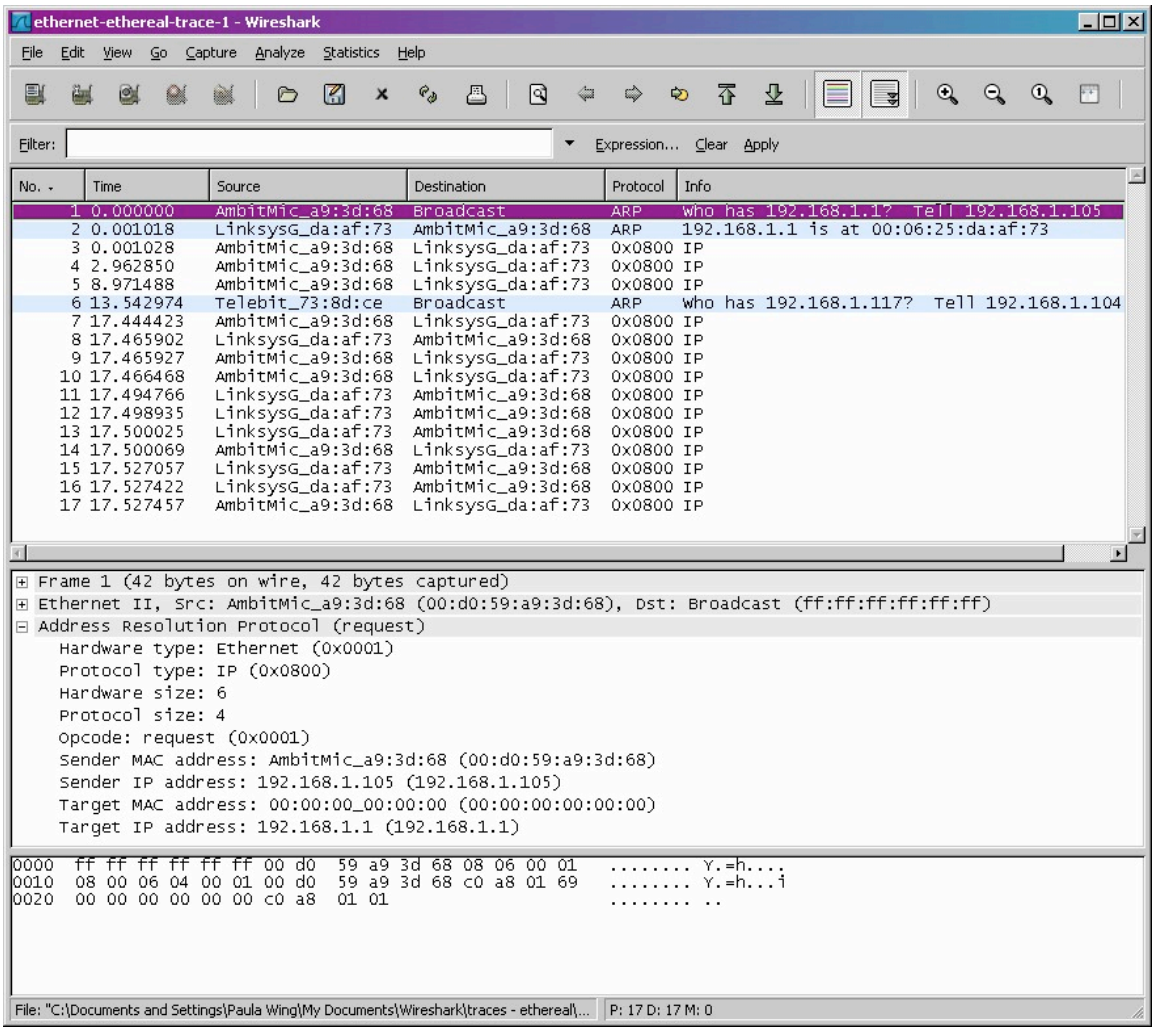
In order to observe your computer sending and receiving ARP messages, we'll need to clear the ARP cache, since otherwise your computer is likely to find a needed IP-Ethernet address translation pair in its cache and consequently not need to send out an ARP message.

- **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.** 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 the machine, you can skip the trace collection part of this lab and just use the trace discussed in footnote 4.

Observing ARP in action

Do the following:

- Clear your ARP cache, as described above.
- Next, make sure your browser's cache is empty. (To do this under 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 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 4. Answer the following questions:

17. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?
18. Give the hexadecimal value for the two-byte Ethernet Frame type field. What do the bit(s) whose value is 1 mean within the flag field?
19. 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 – the Ethernet address of the machine whose corresponding IP address is being queried?

20. 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?
21. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?
22. 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 indiated 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?