COMP 3331/9331: Computer Networks and Applications

Lab Exercise 5b: Ethernet (Solutions)

EXPERIMENT 1: Studying Ethernet using Wireshark

Tools

For this experiment, we will use the *Wireshark* packet analyser that we used extensively in the previous lab. Before you begin go to the "Trace Files" link on the course webpage and download the traces for the Link Layer lab.

Reference

Before you begin this lab, you'll probably want to review sections 5.5 (Ethernet) and 5.4.1 (link layer addressing) in the textbook.

Exercise

Follow the steps described below. You will notice certain questions as you attempt the exercise. Write down the answers for your own reference. The solutions will be put up on the webpage at the end of the week. If you have any questions or are experiencing difficulty with executing the lab please consult your lab instructor.

Step 1: Open an xterm and run Wireshark.

Step 2: Load the trace file *ethernet-ethereal-trace-1* by using the *File* pull down menu, choosing *Open* and selecting the appropriate trace file. This file captures the sequence of HTTP request and response messages exchanged between a browser and a web server (gaia.cs.umass.edu). The web server response contains the rather lengthy US Bill of Rights.

Step 3: Locate the packet containing the HTTP GET request. Since this lab is about Ethernet and ARP, 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.

Step 4: 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 1.7.2 in the text if you find this nesting a bit confusing). To find this packet you will need to look inside the Data part of the Ethernet frames. (Hint: you will find this packet midway through the trace) 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.

Step 5: Answer the following questions, based on the contents of the Ethernet frame containing the HTTP GET message.

Question 1. What is the 48-bit Ethernet address of the source host of this packet?

Answer: The source Ethernet address is 00:d0:59:a9:3d:68.

Question 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? (Note: this is an important question, and one that students sometimes get wrong. Re-read pages 450-451 in the text and make sure you understand the answer here)

Answer: The destination address is 00:06:25:da:af:73. The source host sending the GET request and the web server (the intended recipient of the GET message) do not belong to the same subnet and are in fact separated by several routers in between. So the destination address here is the MAC address of the router connected to the LAN segment of which the source host is a part of (i.e. the first hop router).

Question 3. Give the hexadecimal value for the two-byte Frame type field.

Answer: The 16 bit hexadecimal value for the Frame type is 0x0800 indicating the IP protocol.

Question 4. How many bytes from the very start of the Ethernet frame does the ASCII "G" in "GET" appear in the Ethernet frame? Note that when you examine the Data portion of this frame, it actually consists of both the Ethernet frame headers as well as the payload (i.e. bottom window in Wireshark shows the entire 686 byte frame that is captured, not just 672 bytes as the *Data(672)* line seems to indicate. Of the bytes preceding the G, the first some number are the Ethernet frame header. Does this include the preamble bytes, or are those bytes omitted from the capture? Given this, how many bytes of frame header are present? What are the remainder of the bytes before the G?

Answer: "G" appears 54 bytes after the start of the frame. The reason for this is that the first 14 bytes represent the Ethernet frame header. The next 20 bytes represent the 20 byte IP header and the 20 bytes following that consist of the TCP headers. Note that the

HTTP GET request is encapsulated in a TCP segment, which in turn is encapsulated in an IP datagram, which finally is encapsulated in an Ethernet frame. The preamble bytes are not captured by Wireshark.

Step 6: Next, answer the following questions, based on the contents of the Ethernet frame containing the first byte of the HTTP response message.

Question 5. What is the value of the Ethernet source address? Is this the address of the host that sent the GET HTTP request, or of gaia.cs.umass.edu (Hint: the answer is no). What device has this as its Ethernet address?

Answer: The source Ethernet address for this frame is 00:06:25:da:af:73. This is neither the Ethernet address of gaia.cs.umass.edu nor the source host. This refers to the Ethernet address of the first-hop router from the source host. This address corresponds to the answer of Question 2 above.

Question 6. What is the destination address in the Ethernet frame? Is this the Ethernet address of the source host that sent the earlier GET HTTP request?

Answer: The destination address of the frame is 00:d0:59:a9:3d:68 which is indeed the Ethernet address of the source host that sent the earlier GET HTTP request.

Question 7. Give the hexadecimal value for the two-byte Frame type field.

Answer: The value of the Frame type field is again 0x0800 indication IP.

Question 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?

Answer: In this case "O" starts 67 bytes from the start of the Ethernet frame. This includes 14 bytes of the Ethernet frame header, 20 bytes of IP header, 20 bytes of TCP header and 13 bytes for the "HTTP /1.1 200 " part of the HTTP response status line.

END OF LAB