

CS78 Computer Networks Spring 2007

Ethereal Lab 1

March 31, 2007

This lab aims to make you more familiar with Ethereal. Ethereal is an open source software tool that we demonstrated in class that allows you to examine packets captured by any network interface on your machine.

Please send your answers for the lab to miluzzo@cs.dartmouth.edu by **5 PM Thursday 12 April 2007**. Emiliano will acknowledge receipt of your emailed submission. If you want to hand in your assignment give it directly to Emiliano before the due date (Room Sudikoff 147 or push it under the door if he is not there). Homework will be marked and returned to you one week after submission.

We recommend that you first install this software on your personal machine, as there are many interesting aspects of the program that require root access. CS department machines do not give you this privilege level. If you do not have access to your own machine we can provide you with already captured traces of packets that can be loaded into Ethereal (please email Nic - niclane@cs.dartmouth.edu for assistance). The loading of such traces does not require root access (and so can be done on these CS department machines). Using these traces you will find it possible to answer any questions in the lab.

You will find when you attempt to install Ethereal that the most recent version of this software now has a new name, Wireshark. There are not any significant changes between Ethereal and Wireshark. The Wireshark website observes this fact - "Same developers, same code, different name. The Ethereal network protocol analyzer has changed its name to Wireshark." You will find that either using Wireshark, or older versions of Ethereal will be suitable for performing both this and subsequent labs. The choice is entirely up to you.

Please work through each of the tasks discussed below. Each task will specify material you are required to hand in. For submission please tar all the material into a single file and submit it to via email as discussed above. A number of the questions will touch on concepts that we have not yet fully covered in class. Do not worry. Just answer the questions best you can. For a higher-level course like CS78 we expect you will consult the textbook, the web and your

other courses to guide you in your answers when needed. Please note, as always, citations must be provided with your answers if you consult any external source for information.

OK, let's start looking at packets!

1. To get started please read <http://www.cs.dartmouth.edu/~cs78/eth-intro.pdf> which is the handout that gives an introduction to Ethereal that comes with the text. Note, that this supplementary document is just to provide background to the lab and installation information. What we want you to do is discussed in the tasks below, which is not part of the book related Ethereal labs.
2. Start a capture session using Ethereal. Capture traffic when you are opening a web page in your browser. Open the web page <http://www.dartmouth.edu>. Provide this trace in your submission and use it to answer the following questions.
3. Examine the trace and find the exchange of packets between your machine and the web server (the host providing the web pages to your machine). Can you find an example packet in this exchange where the packet contains details about the type of your web browser (e.g., if it is Fire Fox, Internet Explorer, etc.) being used? If you can find this then what is the value of this attribute in your particular trace? Why is this information given? Examine the source html file of the Dartmouth web page (and provide it with your submission). Can you find the same attribute value in this file? If not why not? Where did it go?
4. In the trace you can see many protocols listed! There is a lot going on under the hood. Some of these protocols are called transport protocols. Which transport protocol is used between your machine and the web server? Why would you think this one is used instead of an alternative? You will see that other protocols are captured in your trace. One such protocol is HTTP. What is the relationship between the transport protocol you identified and HTTP? Both protocols are used to satisfy your browser's request for a web page. Why was more than one protocol used?
5. In the trace you will find IP addresses within the packets. Find an example packet in the trace where the IP address associated with your machine is present. Provide this example packet with your submission. Why is the IP address present in the particular packet you selected (what purpose does it serve)? How are IP addresses and port numbers used, to address what specifically?
6. We discussed protocol layers in class. Which layer is the IP associated with and why isn't it associated with say the application layer?
7. We didn't talk about the MAC (medium access control) address in class yet. But you can find the MAC address of your machine using "ifconfig -a" on unix, Linux, and OSX

machines (Why not read the manual pages of ifconfig). Each node has a unique Link Layer MAC address. Can you find the MAC address for your machine within this trace. What is the MAC address of your machine? Provide a trace of the packet in which you found it. Why do you think a MAC address is needed given that your machine has an IP address (it would be more precise to say that the IP address is associated with one of the network interfaces on your machine, this is true also of the MAC address).

8. Finally, there are an awful lot of protocols used by your machine that you were not aware of. Nor did you realize that lots of packets are being transmitted even when you thought your machine was idle. Consider the short trace you just captured. You will see many different protocols listed in the trace. Excluding the HTTP and TCP protocols identify one of these other protocols. Describe why this protocol is being used and what it is all about (Google it and summarize the service it offers. Recall a protocol is the core of a layer that provides a service to a higher layer – what is this service for one of the protocols in your list other than the obvious ones DNS, TCP, HTTP). Search to see if you can find an RFC for any of the protocols within your trace (one example RFC database is: <http://www.rfc-editor.org/rfc.html>). Can you find the RFCs for these protocols in your trace? Maybe dig into the RFC – is there a state machine? Furthermore, do a little research and find out more about the RFC process, what role does the RFC process perform? How does it work? OK we are done. Great job! You learnt a lot.