Wireshark Tutorial

INTRODUCTION

The purpose of this document is to introduce the packet sniffer WIRESHARK. WIRESHARK would be used for the lab experiments. This document introduces the basic operation of a packet sniffer, installation, and a test run of WIRESHARK.

PACKER SNIFFER

The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. As the name suggests, a packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a *copy* of packets that are sent / received from/by application and protocols executing on your machine.

Figure 1 shows the structure of a packet sniffer. At the right of Figure 1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in Figure 1 is an addition to the usual software in your computer, and consists of two parts. The **packet capture library** receives a copy of every link-layer frame that is sent from or received by your computer. Messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In Figure 1, the assumed physical media is an Ethernet, and so all upper layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

The second component of a packet sniffer is the **packet analyzer**, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string "GET," "POST," or "HEAD".

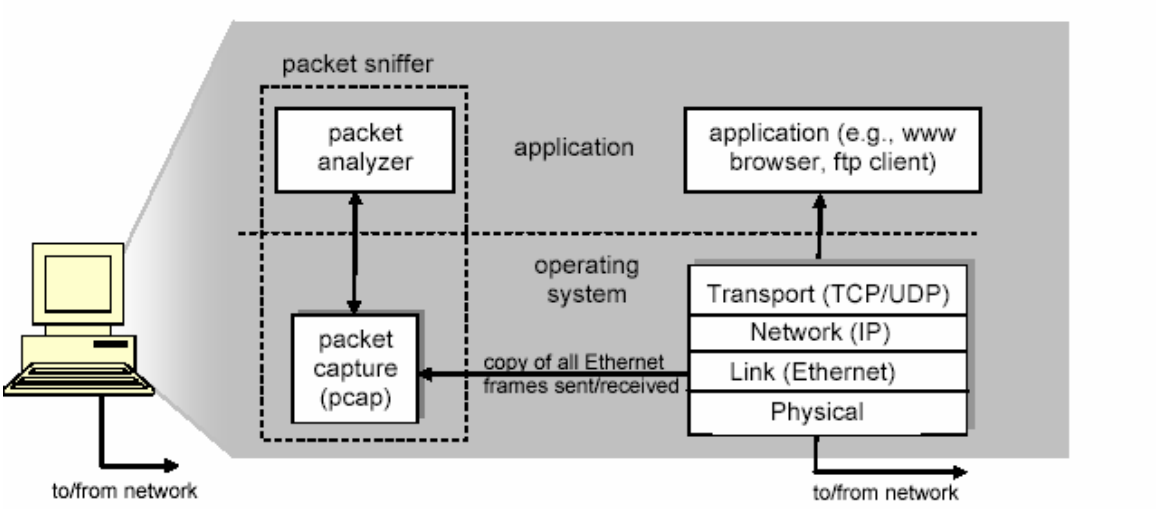

Figure 1: Packet sniffer structure

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers. It's an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a user-guide (http://www.wireshark.org/docs/wsug_html_chunked/), man pages (http://www.wireshark.org/docs/man-pages/), and a detailed FAQ (http://www.wireshark.org/faq.html), rich functionality that includes the capability to analyze hundreds of protocols, and a well-designed user interface. It operates in computers using Ethernet, Token-Ring, FDDI, serial (PPP and SLIP), 802.11 wireless LANs, and ATM connections (if the OS on which it's running allows Wireshark to do so).

Getting Wireshark

Wireshark has been installed on all machines in lab 237. Wireshark can be started on the PCs by executing the following steps:

Step 1 – Log on to the Linux PC in lab 237
Step 2 - Open a the terminal window
Step 3 – Enter the command "*sudo wireshark*".
Step 4 - Enter your account password

Running Wireshark

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will de displayed. Initially, no data will be displayed in the various windows.

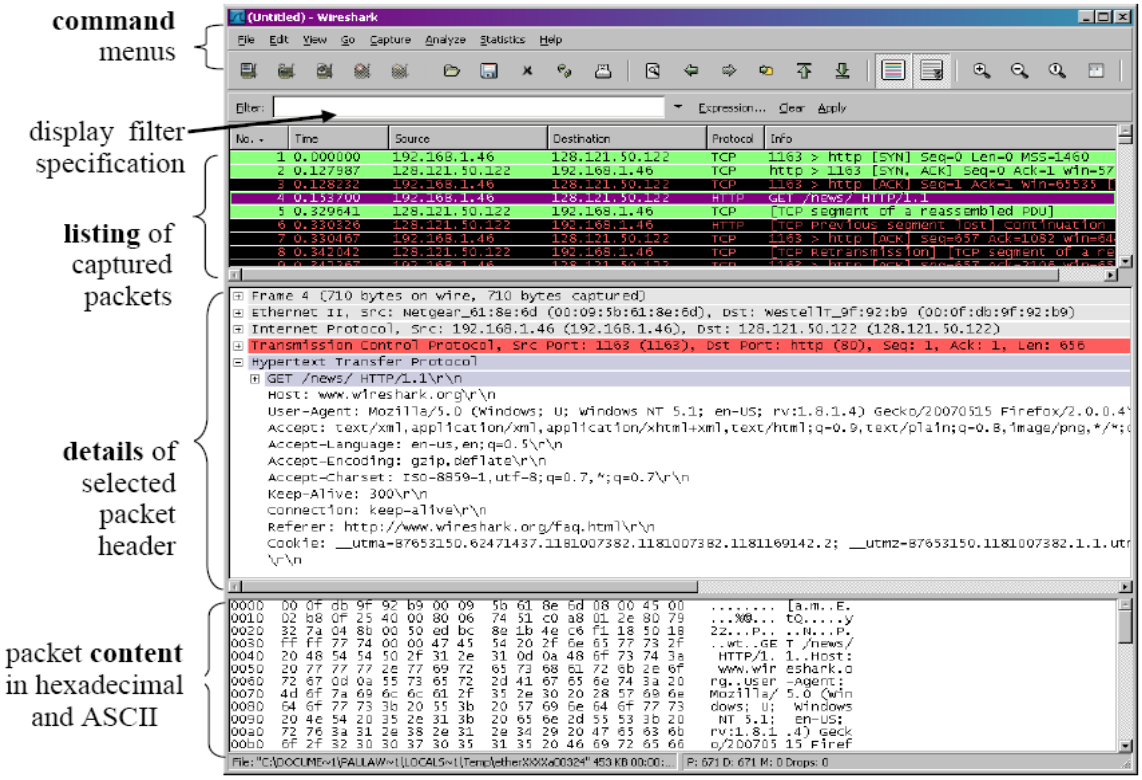

Figure 2: Wireshark Graphical User Interface

The Wireshark interface has five major components:

- The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is *not* a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The

protocol type field lists the highest level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

- The **packet-header details window** provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing window, place the cursor over the packet's one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest level protocol that sent or received this packet are also provided.

- The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

- Towards the top of the Wireshark graphical user interface, is the **packet display filter field,** into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

Test Run

The best way to learn about any new piece of software is to try it out! First, you need to know the network interconnections in the lab. The IP addresses are shown in Table 1 The 11 PCs are connected in the following fashion. **(1 ↔ 2),(3 ↔ 4),(5 ↔ 6),(7 ↔ 8),(9 ↔ 2),(9 ↔ 1),(10 ↔ 3),(10 ↔ 4),(11 ↔ 5),and (11 ↔ 6)**. For ex (1 ↔ 2) means Pc1 and Pc2 are connected to the same switch. So PC1 and PC2 can communicate with each other. To perform the following steps, identify the two PCs for your test run.

Do the following

1. Start up your favorite web browser.

2. Start up the Wireshark software. You will initially see a window similar to that shown in Figure 3, except that no packet data will be displayed in the packet listing, packet-header, or packet-contents window, since Wireshark has not yet begun capturing packets. Make sure you check "Don't show this message again" and press "ok" on the small dialog box that pops up.

3. To begin packet capture, select the Capture pull down menu and select Interfaces. This will cause the “Wireshark: Capture Interfaces” window to be displayed, as shown in Figure 4.

Table 1- IP Address Assignment

PC	IP Addresses(eth0 and eth1)
1	10.0.1.1 and 10.0.1.2
2	10.0.1.3 and 10.0.1.4
3	10.0.1.5 and 10.0.1.6
4	10.0.1.7 and 10.0.1.8
5	10.0.1.9 and 10.0.1.10
6	10.0.1.11 and 10.0.1.12
7	10.0.1.13 and 10.0.1.14
8	10.0.1.15 and 10.0.1.16
9	10.0.1.17 and 10.0.1.18
10	10.0.1.19 and 10.0.1.20
11	10.0.1.21 and 10.0.1.22

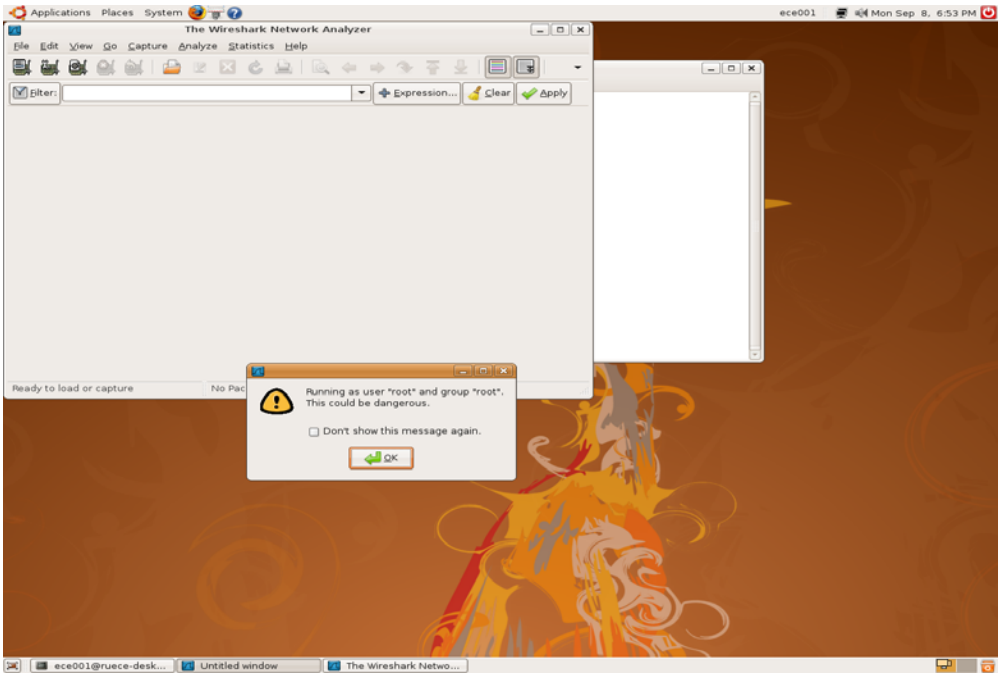

Fig. 3 Wireshark GUI

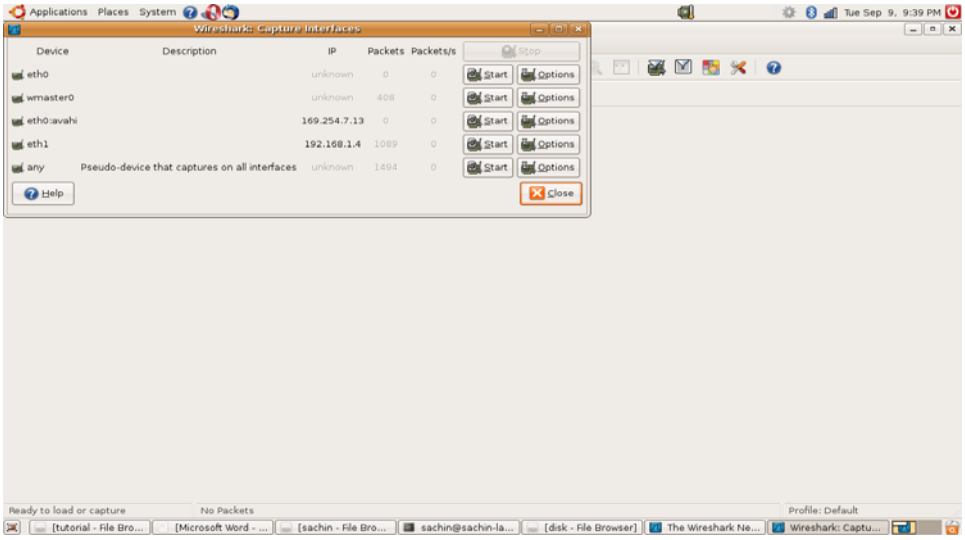

Figure 4: Wireshark Capture Interfaces Window

4. The network interfaces (i.e., the physical connections) that your computer has to the network are shown. The attached snapshot was taken from my computer. You may not see the exact same entries when you perform a capture in the 237 Lab. You will notice that eth0 and eth1 will be displayed. Click "Start" for interface eth0. Packet capture will now begin - all packets being sent / received from/by your computer are now being captured by Wireshark!

5. If you started your Web browser on PC1, you can only connect to PC2 and PC9 (refer to the interconnections listed at the start of this section). If you want to connect to PC2, refer to Table 1, and identify the IP address of eth0. The IP address is 10.0.1.3. If you wanted to connect to PC9, the IP address would be 10.0.1.17. While Wireshark is running, enter the URL: http://10.0.1.3/INTRO.htm to connect to the web server in PC2 and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at 10.0.1.3(PC2) and exchange HTTP messages with the server in order to download this page. The Ethernet frames containing these HTTP messages will be captured by Wireshark.

6. After your browser has displayed the intro.htm page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to Figure 2. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the PC2 web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the *Protocol* column in Figure 2). Even though the only action you took

was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user.

7. Type in “http” (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select *Apply* (to the right of where you entered “http”). This will cause only HTTP message to be displayed in the packet-listing window.

8. Select the first http message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer(ex. PC1) to the PC2 HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window2. By clicking on right pointing and down-pointing arrows heads to the left side of the packet details window, *minimize* the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. *Maximize* the amount information displayed about the HTTP protocol. Your Wireshark display should now look roughly as shown in Figure 5 (Note, in particular, the minimized amount of protocol information for all protocols except HTTP, and the maximized amount of protocol information for HTTP in the packet-header window).

9. Exit Wireshark

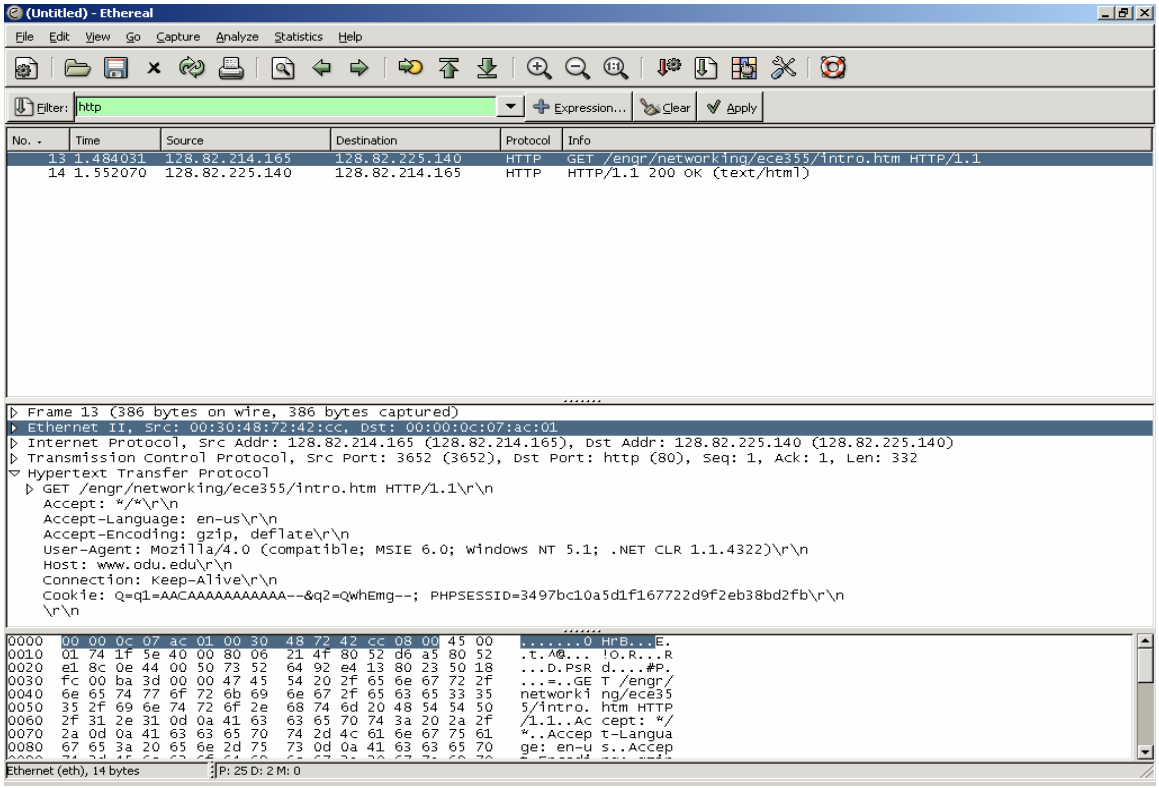

Figure 5: Wireshark display after step 9

Review Questions

1. List the different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.

2. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet listing window is the amount of time, in seconds, since Wireshark tracing began.
To display the Time field in time-of-day format, select the Wireshark *View* pull down menu, then select Time *Display Format*, then select *Time-of-day*.)