Yengkong

Sayaovong

IFT 259 Introduction to Internet Networking

Lab 8
Use Wireshark to view and examine traffic

After you complete each step, put a '√' or 'x' in the completed (yes/no) box

One's understanding of network protocols can often be greatly deepened by "seeing protocols in action" and by "playing around with protocols" – observing the sequence of messages exchanged between two protocol entities, delving down into the details of protocol operation, and causing protocols to perform certain actions and then observing these actions and their consequences.

In this first Wireshark lab, you'll get acquainted with Wireshark, and make some simple packet captures and observations.

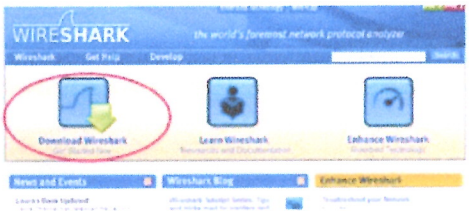
The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. As the name suggests, a packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various 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.

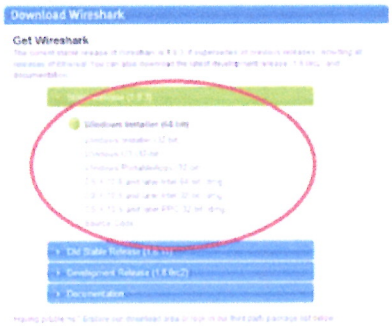
We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and 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/),

Note: New versions of Wireshark may appear different than the screenshots below.

Download & Install Wireshark

- You are going to install and use a free packet sniffer (Wireshark) to analyze traffic on the network.
- Go to www.wireshark.org, download Wireshark from the Internet and install it.

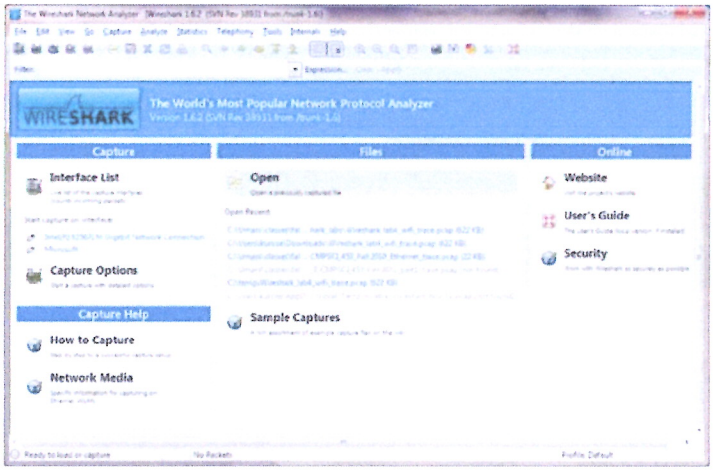




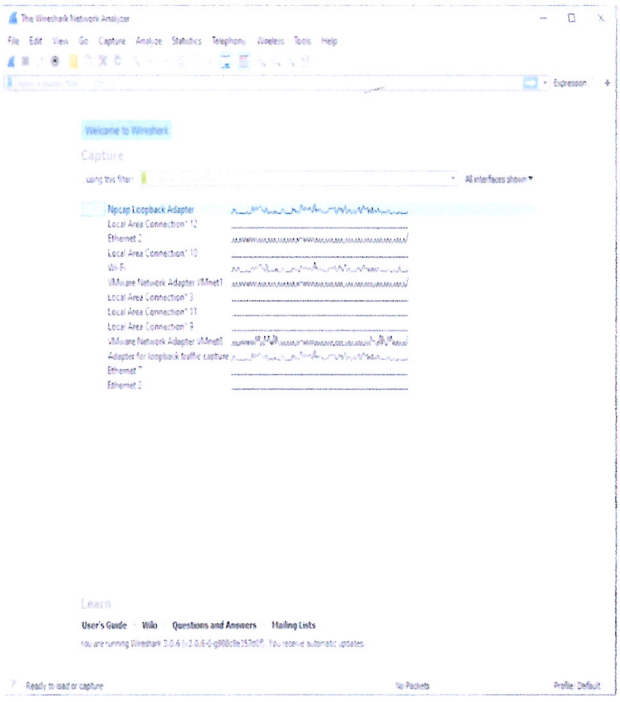
Completed ☒

Using Wireshark (a brief usage)

1. Start Wireshark (you may need to run it in Administrator mode, right click the Wireshark icon and select 'Run as administrator') and you'll get a startup screen, as shown below

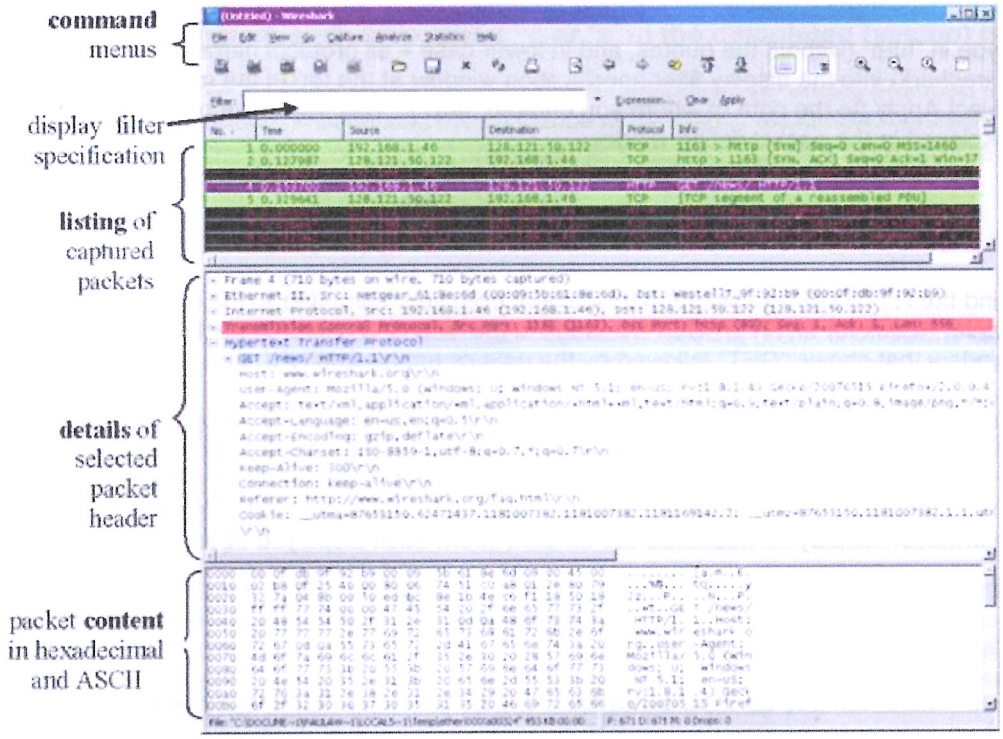


2. Open **Wireshark** and double click on the **network interface with traffic** that you primarily use to connect to the Internet (Ethernet, Ethernet 2, Wi-Fi). In this example, Wi-Fi is the primary network interface with traffic that I am currently connected to.



Completed ☒

3. A screen like the one below will be displayed, showing information about the packets being captured on this interface. Once you start packet capture, you can stop it by using the Capture pull down menu and selecting Stop.



Completed ☒

4. Let's capture some interesting packets. To do so, we'll need to generate some network traffic. Let's do so using a web browser, which will use the HTTP protocol that we will study in detail in class to download content from a website.
5. While Wireshark is capturing packets, open a browser and go to http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html.

Completed ☒

6. In order to display this page, your browser will contact the HTTP server at gaia.cs.umass.edu and exchange HTTP messages with the server in order to download this page. The Ethernet frames containing these HTTP messages (as well as all other frames passing through your Ethernet adapter) will be captured by Wireshark.
7. After about 30 seconds (after your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), 'stop' the capturing of packets (Capture – Stop)
8. 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 gaia.cs.umass.edu 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). 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.

9. Scroll through the Source column in the packet list pane to view the list of devices for which the Wireshark analyzer captured packets. Do you recognize your IP addresses?

yes ☒ no ☐

10. 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.

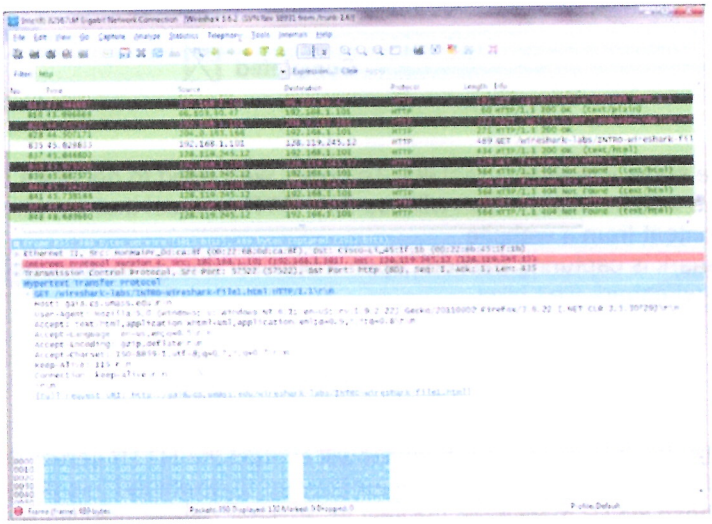
Completed ☒

11. Find the HTTP GET message that was sent from your computer to the gaia.cs.umass.edu HTTP server. (Look for an HTTP GET message in the "listing of captured packets" portion of the Wireshark window that shows "GET" followed by the gaia.cs.umass.edu URL that you entered).

12. 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 '+' and '-' right-pointing and down-pointing arrowheads 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.

Completed ☒

13. Your Wireshark display should now look roughly as shown in the figure below.



Completed ☒