ELEC8560-1-R-2023F|Computer Networks

AT

Assignments › Lab 1: Introduction to Wireshark

Lab 1: Introduction to Wireshark

Hide Assignment Information

Instructions

University of Windsor
Department of Electrical and Computer Engineering

ELEC 8560 - Computer Networks

Semester: Fall 2023

Lab 1: Introduction to Wireshark

Instructions:

- Support all answers by a screenshot of your Wireshark and Command Prompt windows. Annotate screenshots to explain your answer.
- Submissions must be through Brightspace.
- There is a 24-hour grace period after the due date without a penalty. Late submissions will not be accepted.

Note: This lab is mostly adapted from materials provided by the authors of *Computer Networking: A Top-Down Approach*.

Introduction

Your understanding of network protocols can often be greatly deepened by "seeing protocols in action" and by "playing around with protocols". This includes 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 the Wireshark labs you will be doing in this course, you will be running various network applications in different scenarios using your own computer. You will observe the network protocols in your computer in action, interacting and exchanging messages with protocol entities executing elsewhere in the Internet.

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 applications and protocols running on your computer, but never sends

Cancel

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.

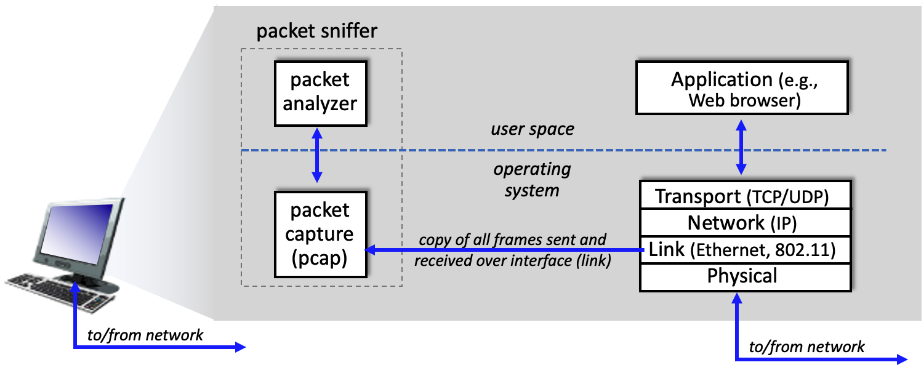

Figure 1: packet sniffer structure

At the right of Figure 1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or email 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 over a given interface (link layer, such as Ethernet or Wi-Fi). Recall that 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 or an 802.11 Wi-Fi radio. Capturing all link-layer frames thus gives you all messages sent/received across the monitored link 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. To do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols. For example, suppose you 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 what the first bytes of an HTTP message will contain.

You will be using the Wireshark packet sniffer for the 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 captures link-layer frames as shown in Figure 1 but uses the generic term "packet" to refer to link-layer frames, network-layer datagrams, transport-layer segments, and application-layer messages, so the lab manuals will use the less-precise "packet" term to go along with Wireshark convention.

Wireshark is a free network protocol analyzer that runs on Windows, Mac, and Linux/Unix computers. It is an ideal packet analyzer for our labs - it is stable, has a large user base and well-documented support that includes a user-guide, man pages, and a detailed FAQ, rich functionality that includes the capability to analyze hundreds of protocols, and a well-designed user interface. Wireshark operates in computers using Ethernet, serial (PPP), 802.11 (Wi-Fi) wireless LANs, and many other link-layer technologies.

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

Getting Wireshark

Download and install the Wireshark software by going to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer. To run Wireshark, you will need to have access to a computer that supports both Wireshark and the *libpcap* or *WinPCap* packet capture library. The libpcap software will be installed for you if it is not already installed within your operating system when you install Wireshark.

Running Wireshark

When you run the Wireshark program, you will get a startup screen that looks something like the screen below.

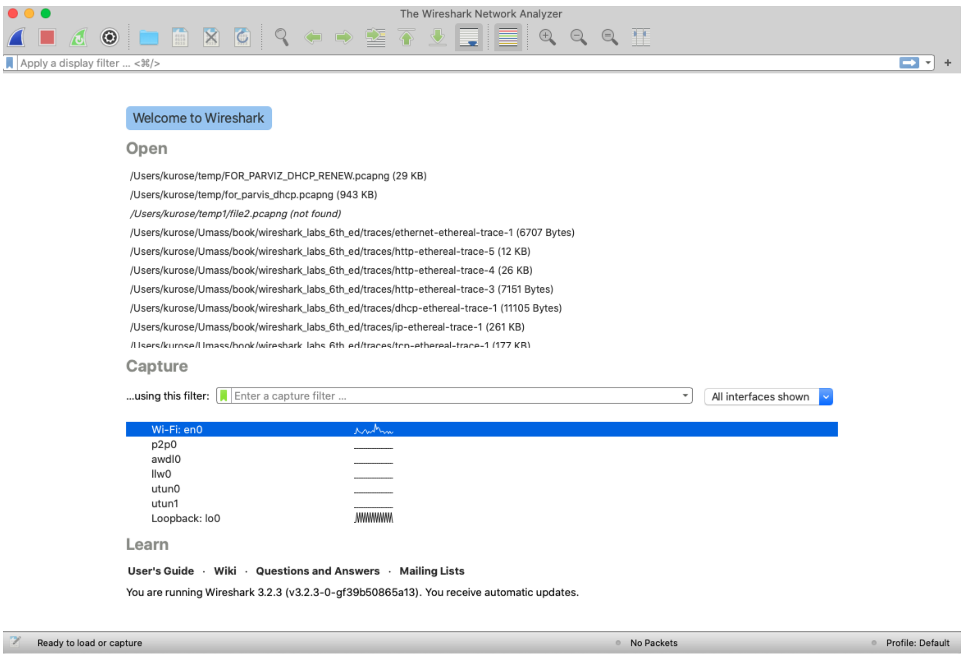

Figure 2: Initial Wireshark Screen

Note: Different versions of Wireshark will have different startup screens, so it is possible that your screen does not look exactly like the screen below.

There is not much that is very interesting on this screen. But note that under the Capture section, there is a list of so-called interfaces. The computer these screenshots are taken from has just one interface - "Wi-Fi en0," (shaded in blue in Figure 2) which is the interface for Wi-Fi access. All packets to/from this computer will pass through the Wi-Fi interface, so it is here where we will want to capture packets. Locate the interface(s) on your startup page through which you are getting Internet connectivity, e.g., mostly likely a Wi-Fi or Ethernet interface, and select that interface.

When you double click on one of these interfaces to start packet capture (i.e., for Wireshark to begin capturing all packets being sent to/from that interface), a screen like the one below will be displayed, showing information about the packets being captured. Once you start packet capture, you can stop it by using the Capture pull down menu and selecting Stop (or by clicking on the red square button next to the Wireshark fin).

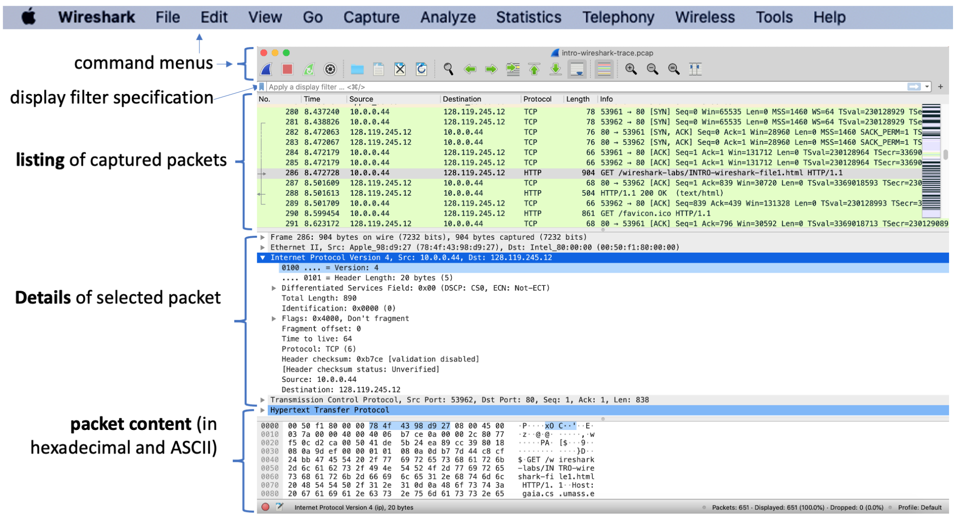

Figure 3: Wireshark window, during and after capture

The Wireshark interface in Figure 3 has five major components:

1. The **menus** are standard pulldown menus located at the top of the Wireshark 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.

2. The **filter toolbar** allows users to set display filters to filter which packets are displayed. That is, a protocol name or other information can be entered to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we will use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

3. The **packet list pane** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; note that 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.

4. The **packet details pane** 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 (assuming the packet was sent/received over an Ethernet interface) 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 plus/minus boxes or right/downward-pointing triangles 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.

5. The **packet bytes pane** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

Testing Wireshark

Make sure that your computer is connected to the Internet via a wired Ethernet interface or a wireless 802.11 Wi-Fi interface, then do the following:

1. Start up your favorite web browser, which will display your selected homepage.

2. Start up the Wireshark software. You will initially see a window similar to that shown in Figure 2. Wireshark has not yet begun capturing packets.

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 (on a PC) or you can choose Options on a Mac.

4. You will see a list of the interfaces on your computer as well as a count of the packets that have been observed on that interface so far. On a Windows machine, click on *Start* for the interface on which you want to begin packet capture. On a Windows machine, select the interface and click Start on the bottom of the window. Packet capture will now begin - Wireshark is now capturing all packets being sent/received from/by your computer!

5. Once you begin packet capture, a window similar to that shown in Figure 3 will appear. This window shows the packets being captured. By selecting *Capture* pulldown menu and selecting *Stop*, or by click on the red Stop square, you can

stop packet capture. But do not stop packet capture yet. Let's capture some interesting packets first. To do so, we will 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.

6. While Wireshark is running, enter the URL:

   http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html

   and have that page displayed in your browser. To display this page, your browser will contact the HTTP server at gaia.cs.umass.edu and exchange HTTP messages with the server to download this page. The Ethernet or Wi-Fi frames containing these HTTP messages (as well as all other frames passing through your Ethernet or Wi-Fi adapter) will be captured by Wireshark.

7. After your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), stop Wireshark packet capture by selecting stop in the Wireshark capture window. The main Wireshark window should now look similar to Figure 3. 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 in Figure 3). 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. We will learn much more about these protocols as we progress through the course! For now, you should just be aware that there is often much more going on.

8. Type in "http" (without the quotes, and *in lower case* -- all protocol names are in lower case in Wireshark, and make sure to press enter/return key) 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") or just hit return. This will cause only HTTP message to be displayed in the packet-listing window. Figure 4 below shows a screenshot after the http filter has been applied to the packet capture window shown earlier in Figure 3. Also note that in the Selected packet details window, we have chosen to show detailed content for the Hypertext Transfer Protocol application message that was found within the TCP segment, that was inside the IPv4 datagram that was inside the Ethernet II (Wi-Fi) frame. Focusing on content at a specific message, segment, datagram and frame level lets us focus on just what we want to look at (in this case HTTP messages).

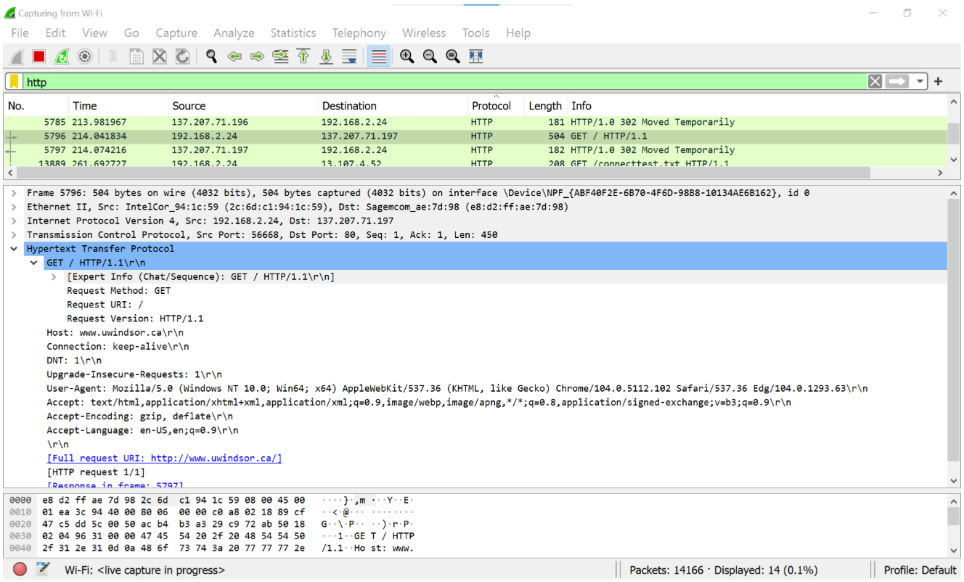

Figure 4: looking at the details of the HTTP message that contained a GET of http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html

3. 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 (see Figures 3 and 4) that shows "GET" followed by the gaia.cs.umass.edu URL that you entered. 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 window.

   Note: The HTTP GET message that is sent to the gaia.cs.umass.edu web server is contained within a TCP segment, which is contained (encapsulated) in an IP datagram, which is encapsulated in an Ethernet frame.

4. By clicking on '+' and '-' 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. Your Wireshark display should now look roughly as shown in Figure 4. (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).

5. Exit Wireshark

Congratulations! You have now completed the first lab!

Questions

Now answer the following questions:

1. Which of the following protocols appear (i.e., are listed in the Wireshark "protocol" column) in your trace file: TCP, QUIC, HTTP, DNS, UDP, TLSv1.2?

2. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received?

    Note: By default, the value of the Time column in the packet list pane is the amount of time, in seconds since Wireshark tracing began. If you want 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.*

3. What is the Internet address of the gaia.cs.umass.edu (also known as www-net.cs.umass.edu)? What is the Internet address of your computer (the computer that sent the HTTP GET message)?

To answer the following two questions, you will need to select the TCP packet containing the HTTP GET request (i.e., similar to Packet 286 in Figure 3).

4. Expand the information on the HTTP message in the packet details pane so you can see the fields in the HTTP GET request message. What type of Web browser issued the HTTP request? The answer is shown at the right end of the information following the "User-Agent" field in the expanded HTTP message display. This field value in the HTTP message is how a web server learns what type of browser you are using.

5. Expand the information on the Transmission Control Protocol for this packet in the packet details pane so you can see the fields in the TCP segment carrying the HTTP message. What is the destination port number (i.e., the number following "Dest Port" for the TCP segment containing the HTTP request) to which this HTTP request is being sent?

Due on Sep 19, 2023 11:59 PM

Available until Sep 20, 2023 11:59 PM. **Submission restricted after availability ends.**

Submit Assignment

Submission is restricted outside of availability dates.