Introduction to Wireshark

OBJECTIVE

In this lab, the student shall work individually to:

1) Learn about packet sniffers and see how they capture and analyze network traffic.
2) Start to learn how Wireshark works.

THEORY: PACKET SNIFFERS

Packet sniffers are a basic tool for observing the messages on a network. 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.

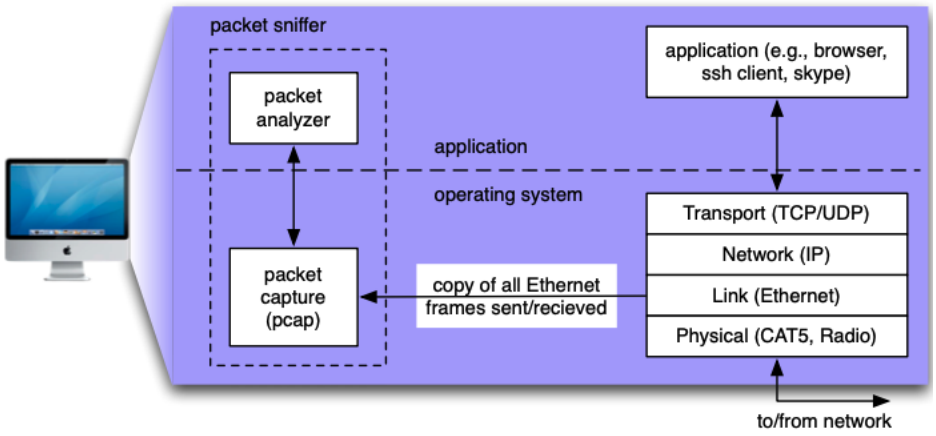


Figure 1: Understanding a Packet Sniffer

Figure 1 above shows the structure of a packet sniffer. At the right are the protocols (in this case, Internet protocols) and applications (such as a web browser or ssh client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle, is an addition to the usual software in your computer, and consists of two parts.

The first 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. Figure 2 below illustrates how the packet analyzer determines the HTTP message.

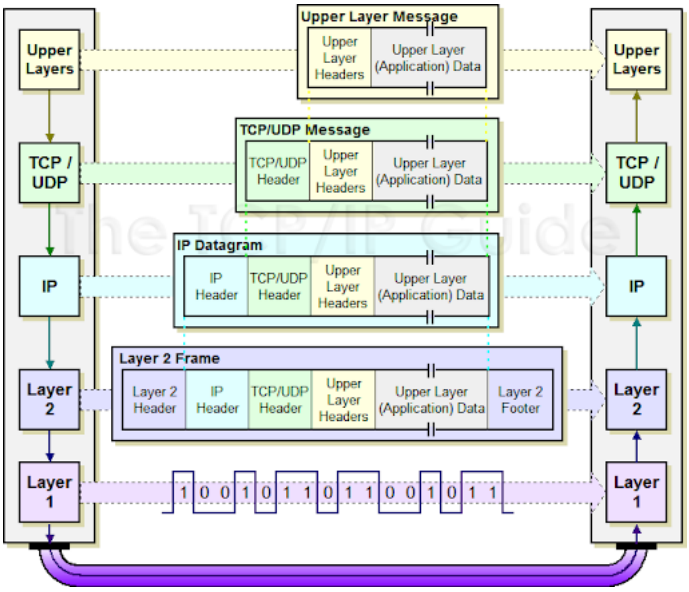


Figure 2: Understanding IP Datagram Encapsulation

The packet analyzer understands the format of the layer 2 Ethernet frames.

- Within the layer 2 Ethernet frame is the IP datagram and the packet analyzer understands the IP datagram format.
- Within the IP datagram is the TCP segment. The packet analyzer extracts the TCP segment.
- The packet analyzer understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment.
- Finally, the packet analyzer understands the HTTP protocol and it knows that the first bytes of an HTTP message will contain the string “GET,” “POST,” or “HEAD,” as shown in image below.

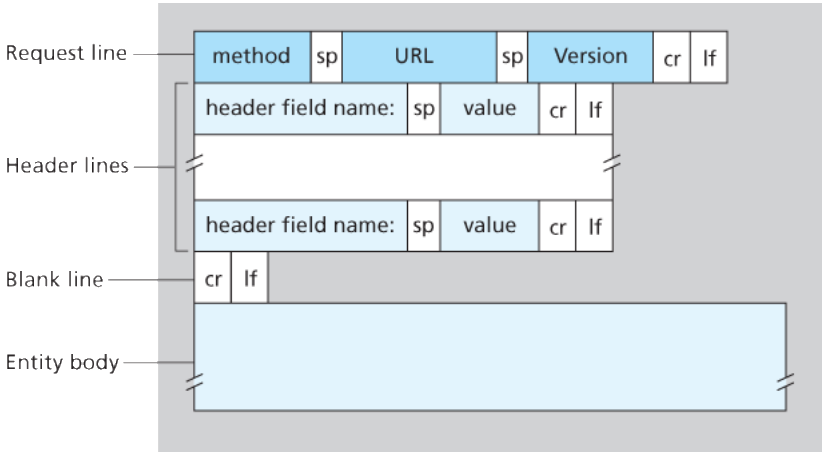


Figure 3: General HTTP Message Format

The second part of the packet sniffer is capturing the packets. The packet capture library receives a copy of every layer 2 data 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 data 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 data link-layer frames thus gives you all messages sent/ received from/by all protocols and applications executing in your computer.

The existence of the packet capture box in figure 1 illustrates two concepts. First, it shows that any packet in a shared medium (Ethernet, Wi-Fi, etc) can be captured and examined without notification of the sender or receiver. You cannot rely on common data link-layer protocols to protect your secrets or your privacy online. At a minimum, you should be using encryption protocols (generally buried in the application layer, though sometimes found elsewhere) to protect all network traffic you generate or receive. Second, you have the ability to act as the "bad guy" and capture the network traffic of other people, examine it and exploit what you find. You need to learn to use this tool in a responsible fashion. Remember the movie quote: "With great power comes great responsibility!" We will use a filter to ensure Wireshark doesn't display traffic other than your own, but this is purely a voluntary measure. Please act ethically and responsibly in your use of Wireshark.

PROCEDURES

1) Open Wireshark – When you installed GNS3 it automatically installed Wireshark 3.06.

- Make sure to also download the Wireshark user guide.
- The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.
- You may need to disable anti-virus protection software (McAffee) before your own IP address will show up in captured data.
- You should be connected to an Ethernet connection. If you only have WiFi, you'll need to figure out how to set your WiFi physical layer into monitor mode, which may be difficult or impossible, depending on your operating system. Failure to follow this instruction will mean you only see traffic originating or being sent to your own computer, which is sub-optimal for these labs.

2) When you run the Wireshark program, the Wireshark graphical user interface will be displayed. Initially, no data will be displayed in the various windows. By the way, the pictures I show in this lab guide may differ, perhaps substantially, from the interface you see on your computer, depending on your installed version and operating system. Be flexible.

3) The Wireshark interface has five major components.

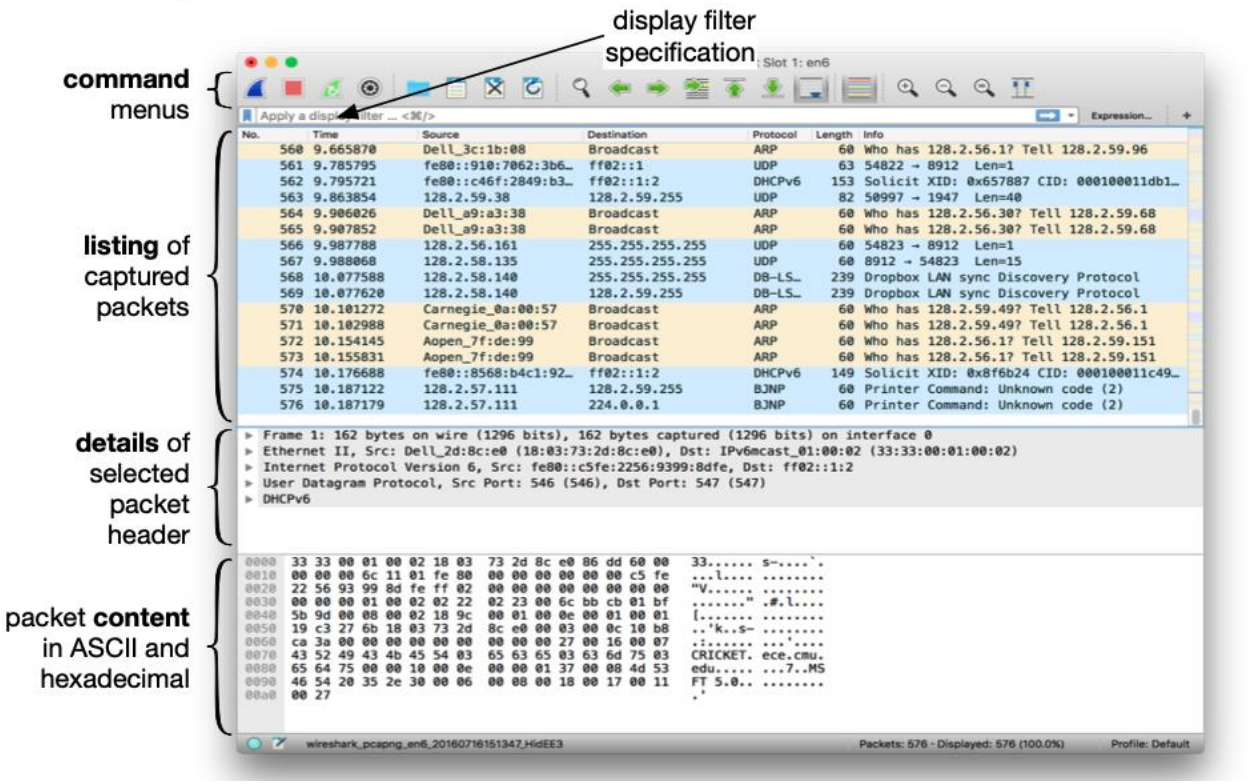


Figure 4: Wireshark Display

1. The **command menus** are standard pulldown menus located at the top of the window or in your menu-bar (not shown figure 4). Included is a toolbar (shown in figure 4). 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. The Capture menu allows you to begin packet capture.

2. 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).

3. The **packet-header details window** provides details about the packet selected 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-or-minus boxes 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. Details about the highest-level protocol that sent or received this packet are also provided.

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

5. 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).

4) Take Wireshark for a "Test Run." The best way to learn about any new piece of software is to try it out! Do the following:

1. Start up your favorite web browser, which will display your selected homepage. If you are using a proxy (especially a host-based one), disable it if possible. You want to examine uncached network traffic.

2. Start up the Wireshark software. You will initially see a window similar to that shown In figure 5 below.

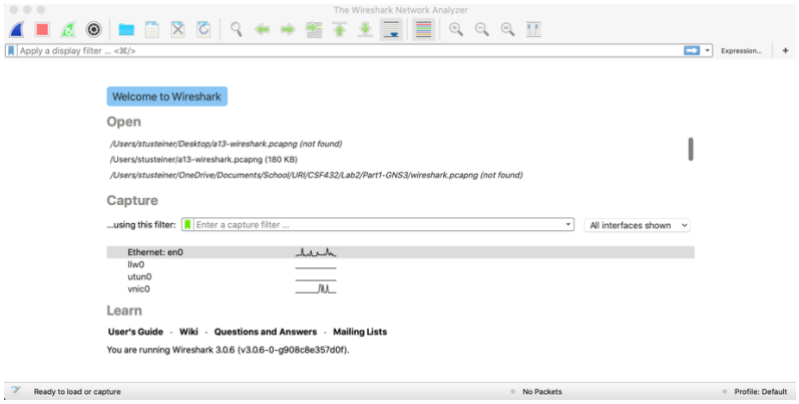
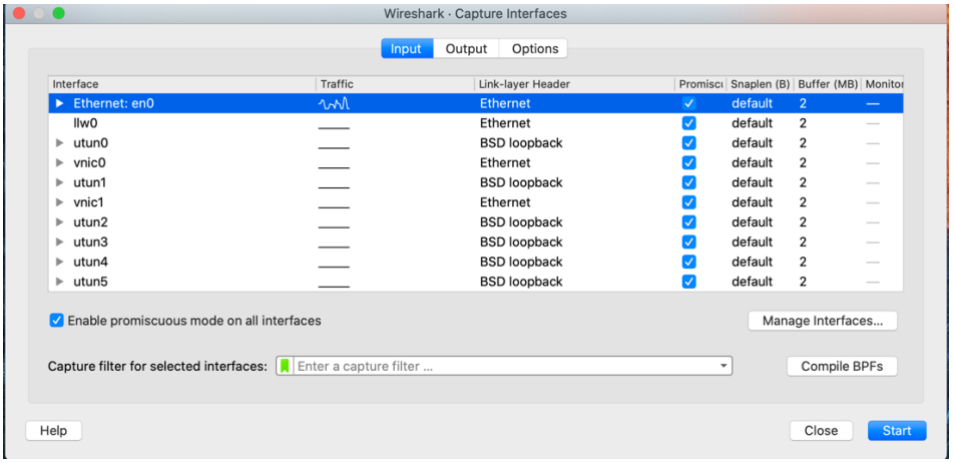
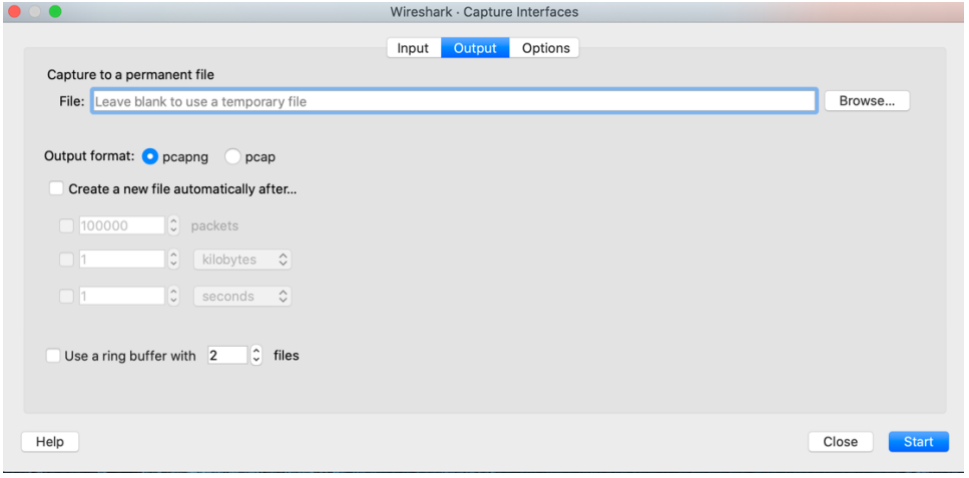


Figure 5: Wireshark Start Window

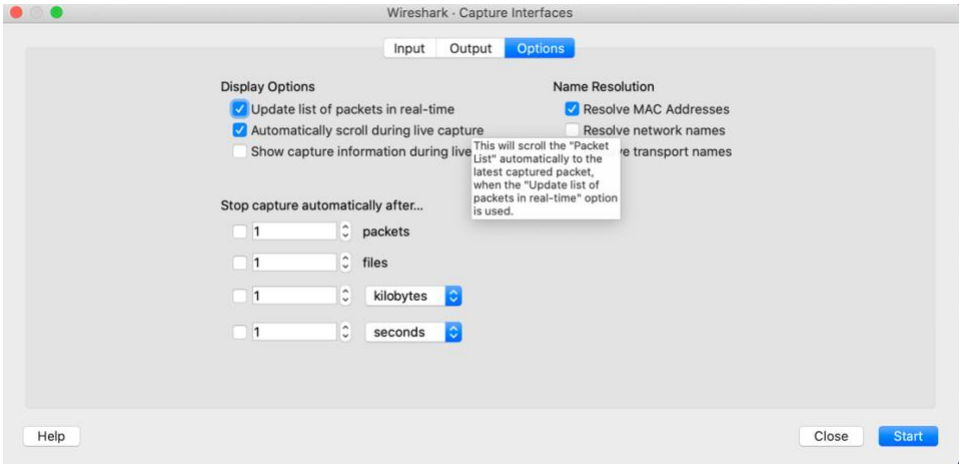
3. To begin packet capture, select the Capture pull down menu and select Options. This will cause the “Wireshark: Capture Interfaces” window to be displayed. There are three sections to this window: Input, Output and Options, as shown below.
    - The Input window allows you to select which interface you will use for capture. You can see that the computer where I took this screenshot has Wi-Fi and a bunch of Ethernet interfaces, as well as the loopback interface. Only one of them is in use, so I'll pick that one.



- The Output window lets you choose to dump all the collected packets into a file. This is handy for scripting. Note: you can limit the file sizes. I generally don't touch anything in this window.



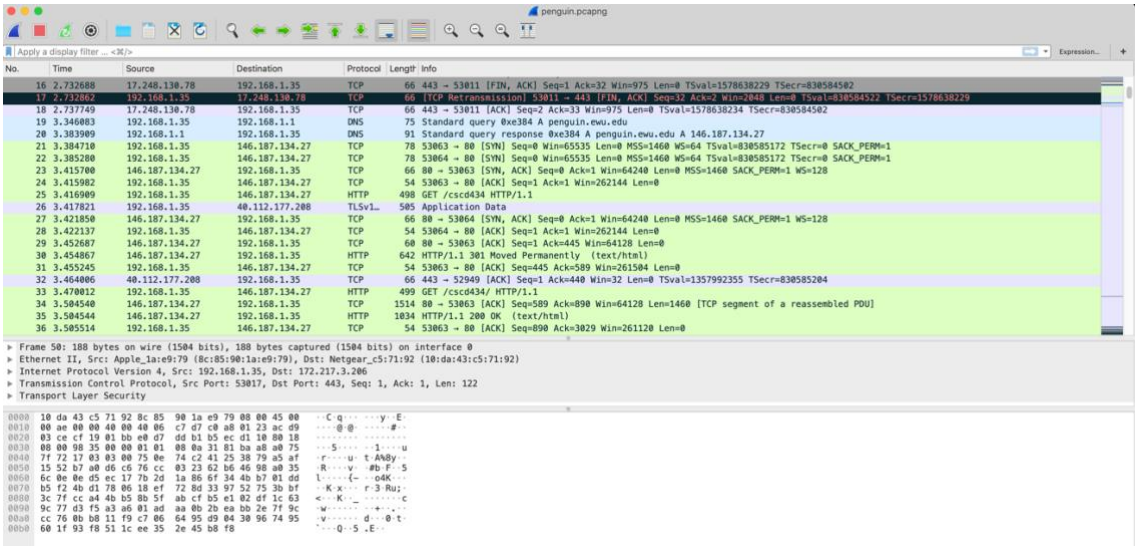
- The Options window lets you specify when the capture should quit (in packets, files, size or time), controls the listing section of the main window during the capture (update or not? Scroll or not?) as well as choose to resolve names or not.



- You can use most of the default values in the Options window, but check "Show extra capture information dialog." The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the Interface pull down menu at the top of the Capture Options window. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets. After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - all packets visible to your network interface (including those being sent/received from/ by your computer) are now being captured by Wireshark!

4. Once you begin packet capture, a packet capture summary window will appear. This window summarizes the number of packets of various types that are being captured, and (importantly!) contains the Stop button that will allow you to stop packet capture. Don't stop packet capture yet.
5. While Wireshark is running, enter the URL http://penguin.ewu.edu/cscd434 and have that page displayed in your browser. Make sure to clear your browser cache if you have previously displayed this webpage (you want to get it across the internet, not from your cache). In order to display this page, your browser will contact the HTTP server at penguin.ewu.edu 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 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 the figure below.

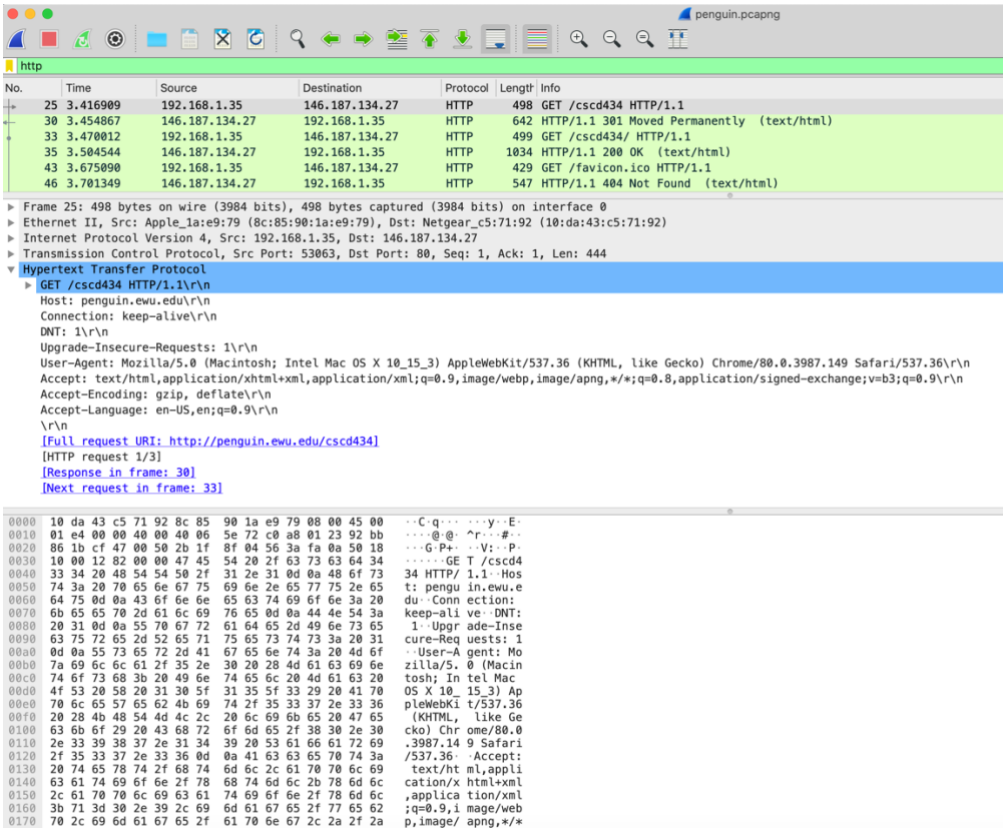


You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the penguin.ewu.edu web server should appear somewhere in the listing of packets captured. There are many packets displayed as well. Even though 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 (as well as data sent via various protocols by other computers on your network). We'll 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 than "meet's the eye!"

7. Type in http (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" in the filter toolbar. This will cause only HTTP message to be displayed in the packet-listing window. Add the filter ip.src == <your IP address> || ip.dst == <your IP address> to filter out traffic that isn't going to or from your computer. This will keep other people's traffic private and get rid of lots of HTTP exchanges from other computers that you don't care about. Filters are combined with C operators.

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 to the penguin.ewu.edu 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 window. By clicking the triangles 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 like the figure below.



(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. To use Wireshark effectively, you need to learn how to filter the results, so you aren't wading through too much data. Wireshark uses two different filters, one to filter the results that get captured and another to filter the results that are displayed. Unfortunately, both use different languages to specify the filter. You've already been introduced to display filters, which use a C-like set of operators. Another powerful operator you should know about is "contains" which does a substring match. The actual values being combined can come from any of the protocols and any of the protocol fields that

The capture time of each packet is quite important, so is displayed in the packet listing area as the second column. By default, this time is "number of seconds since the beginning of capture." However, you have control over what is displayed. Explore the View → Time Display Format menu to see display formats as well as precision choices. Also, of interest is the ability to change the time reference so that all times are displayed relative to the capture time of a chosen packet. First, chose a packet from the display list by clicking on it. Then, go to the Edit → Set/Unset Time Reference, which will toggle

your choice to use the chosen packet as the reference. When set, you will see the time for that packet changed to "*REF*" All other packet's time has been changed to seconds before or after the capture of that reference packet. This is a particularly handy way to figure out round-trip-time. Set the request packet as the reference, then find the reply packet. The time given on that packet will be the number of seconds it took from the request packet for it to arrive. No arithmetic necessary!

The display filter language is also used to define rules that Wireshark uses to assign colors to particular packets in the user interface. Using the captured packets, practice temporary color changes by selecting a packet and then pressing <ctrl> 1, <ctrl> 2, etc. Also, examine the coloring rules dialog and experiment with defining permanent coloring rules (you might want to export the default set of coloring rules before messing around with them).

Capture filters are also quite useful. They let you restrict the amount of data you collect in the first place. Whereas display filters don't actually change the contents of the data that Wireshark collects, merely which of the packets that have been captured are displayed. Capture filters are entered in the "Filter" field of the "Capture Options" dialog box. The capture language is based on tcpdump and requires a bit more protocol knowledge to use. For now, simply experiment with host <ip address> to ensure you don't capture data from other network users.

10. Exit Wireshark

Congratulations! You've now completed setting up an important network engineering tool and learning a bit about its operation.

WHAT TO TURN-IN

The goal of this first lab was primarily to introduce you to Wireshark. The following questions will demonstrate that you've been able to get Wireshark up and running and have explored some of its capabilities. Answer the following questions, based on your Wireshark experimentation.

1. List up to 10 different protocols that appear in the protocol column in the unfiltered packet-listing window. As I don't have control over the data flowing over your network at the time of your lab, I don't know exactly how many and what protocols those will be. I do expect that you have a bunch (if less than 5, please look harder). Just list out those that you see, but don't bother to list more than 10. (5 Points)

2. Open your web browser and use the following address https://cse.wwu.edu/computer-science. Once the page loads wait 3 seconds and stop the capture. 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.) Include a screenshot and describe where you got the data to answer this question. (15 Points)

3. What is the Internet address (IP address) of cse.wwu.edu/computer-science? What is the Internet address of your computer (This might be a private address, if you are behind a NAT device. No worries, we'll learn about that later)? Include a screenshot and describe where you got the data to answer this question. (15 Points)

4. How many packets did you capture (total of all protocols, not just HTTP)? Now, use display filters to determine how many packets contain your ip address (hint: Use ip.addr instead of the clumsy ip.src or ip.dst format). What is this filter you used? Now, reverse the filter to determine how many packets don't contain your ip address. See any problems here? If not, you've already figured out the point of this question, so explain how you did so. If so, how can this problem be fixed? What are the appropriate display filters to use? How does Wireshark warn you of such a problem? (This is an important detail to remember about Wireshark. Please ensure you've discussed the problem well enough so that the grader can ensure you explored it thoroughly. If your numbers show you don't have a problem, then figure out how you might reverse the filter in such a way as to cause a problem.) (25 Points)

5. Using the address http://www.ini.cmu.edu. Once the page loads wait 3 seconds and stop the capture. Use your newly acquired Wireshark skills to capture the process when your browser loads the front page of INI's website. How many packets did you capture? Were all of them HTTP? How many HTTP requests did you make? Were all the replies "200 OK"? Did you find anything else interesting? Please ensure you have examined this packet capture in detail, using appropriate Wireshark functionality. Write up what you observed (include screen captures to justify your answers). (40 Points)

6. Turn in your answers in a single PDF file named your last name first letter of your first name wslab1.pdf (Example: steinerswslab1.pdf)