Birla Institute of Technology and Science Pilani (Rajasthan)

ECE F343 Communication Networks
Second Semester 2025-26
Tutorial 1
Wireshark Introduction

Objective: To get acquainted with Wireshark, which is a popularly used network protocol analyzer tool, 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.

Figure 0 shows the structure of a packet sniffer. At the right of Figure 0 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 0 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 WiFi). 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 WiFi 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.

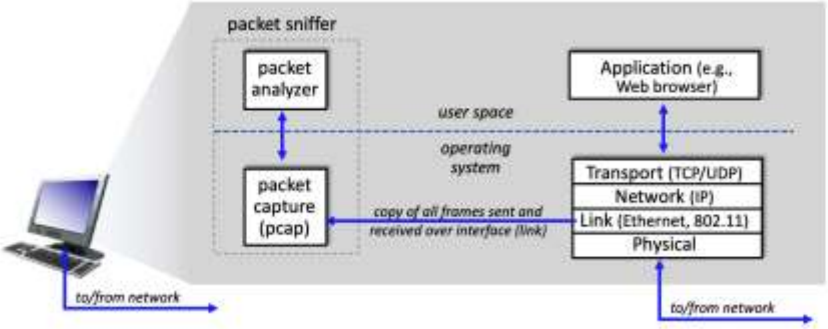


Figure 0: packet sniffer structure

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 0. 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".

In this lab, we will use the Wireshark packet sniffer [http://www.wireshark.org/], that allows 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. Also, technically speaking, Wireshark captures link-layer frames, but uses the generic term "packet" to refer to link-layer frames, network-layer datagrams, transport-layer segments, and application-layer messages, so we'll use the less-precise "packet" term here to go along with Wireshark convention).

Getting Wireshark: In order to run Wireshark, you'll 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 installed within your operating system, when you install Wireshark. See http://www.wireshark.org/download.html for a list of supported operating systems and download sites.

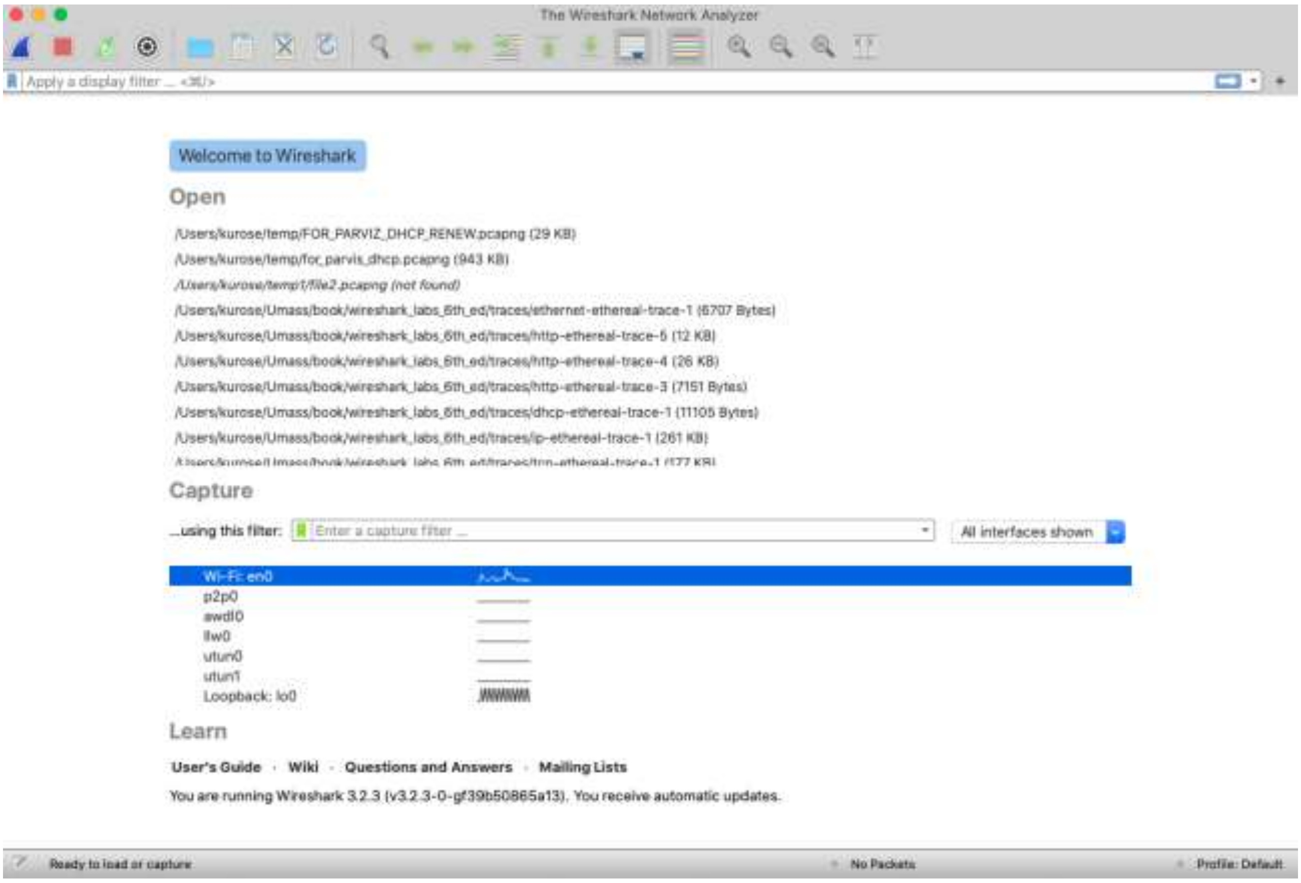
Download and install the Wireshark software: Go to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer.

The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.

Running Wireshark: Before you run Wireshark, here are two things to check in your network connection and browser configuration:

- Make sure you are not running a VPN (virtual private network) service. When you're running a VPN service, upper-layer protocol information (HTTP, TCP) that is sent by your computer may be encrypted. This makes it impossible to use Wireshark to look at what's going on inside these protocols!
- Make sure your browser program is not, by default, using the HTTP/3 protocol or the QUIC protocol. When you're running HTTP/3 or QUIC, upper-layer protocol information (HTTP, TCP) sent by your computer will be encrypted. By 2024, most popular browsers have adopted HTTP/3 and QUIC as defaults. A document that describes how to disable HTTP/3 AND QUIC as the defaults for your browser is here: https://techysnoop.com/disable-quic-protocol-in-chrome-edge-firefox/ .

When you run the Wireshark program, you'll get a startup screen that looks something like the screen below. Different versions of Wireshark will have different startup screens – so don't panic if yours doesn't look exactly like the screen below.



The Wireshark Network Analyzer
Apply a display filter ... <⌘/>
Welcome to Wireshark

Open

/Users/kurose/temp/FOR_PARVIZ_DHCP_RENEW.pcapng (29 KB)
/Users/kurose/temp/for_parvis_dhcp.pcapng (943 KB)
/Users/kurose/temp1/file2.pcapng (not found)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/ethernet-ethereal-trace-1 (6707 Bytes)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/http-ethereal-trace-5 (12 KB)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/http-ethereal-trace-4 (26 KB)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/http-ethereal-trace-3 (7151 Bytes)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/dhcp-ethereal-trace-1 (11105 Bytes)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/ip-ethereal-trace-1 (261 KB)
/Users/kurose/Umass/book/wireshark_labs_6th_ed/traces/tcp-ethereal-trace-1 (177 KB)

Capture

...using this filter: Enter a capture filter ... All interfaces shown

Wi-Fi: en0
p2p0
awdl0
llw0
utun0
utun1
Loopback: lo0

Learn

You are running Wireshark 3.2.3 (v3.2.3-0-gf39b50865a13). You receive automatic updates.

Figure 1: Initial Wireshark Screen

Under the Capture section, there is a list of so-called interfaces. The Mac computer from where this screenshot is taken has just one interface – "Wi-Fi en0," (shaded in blue in Figure 1) which is the interface for Wi-Fi access. All packets to/from this computer will pass through the Wi-Fi interface, so it's here where we'll want to capture packets. On a Mac, double click on this interface (or on another computer locate the interface on startup page through which you are getting Internet connectivity, e.g., mostly likely a WiFi or Ethernet interface, and select that interface in the Wireshark screen where you specify the packet capture interface).

If you click on one of the 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 in Figure 1).

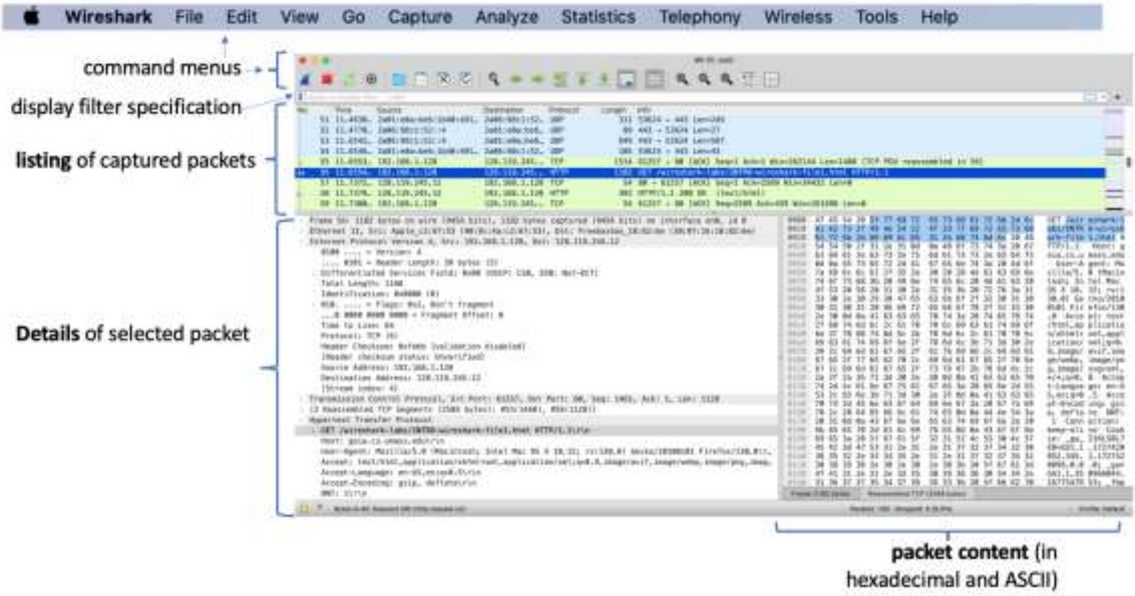

Wireshark
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
command menus
display filter specification
listing of captured packets
Details of selected packet
packet content (in hexadecimal and ASCII)

Figure 2: Wireshark window, during and after capture

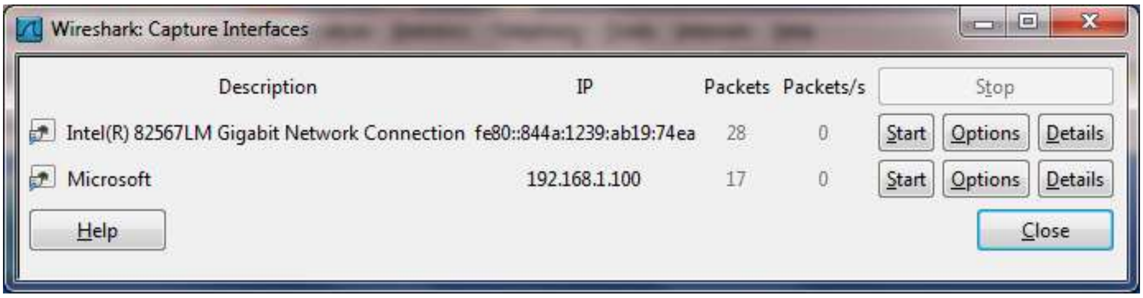
The Wireshark interface has five major components:

- The **command menus** are standard pulldown menus located at the top of the Wireshark window (and on a Mac at the top of the screen as well; the screenshot in Figure 2 is from a Mac). 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; 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 IP addresses, the upper-layer 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 (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.

- 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 that do not correspond to HTTP messages.

Test Run: Connect your computer to the Internet via a wired Ethernet interface or a wireless 802.11 WiFi interface. 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 1. 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. You should see a list of interfaces, as shown in Figures 3 (on Windows).



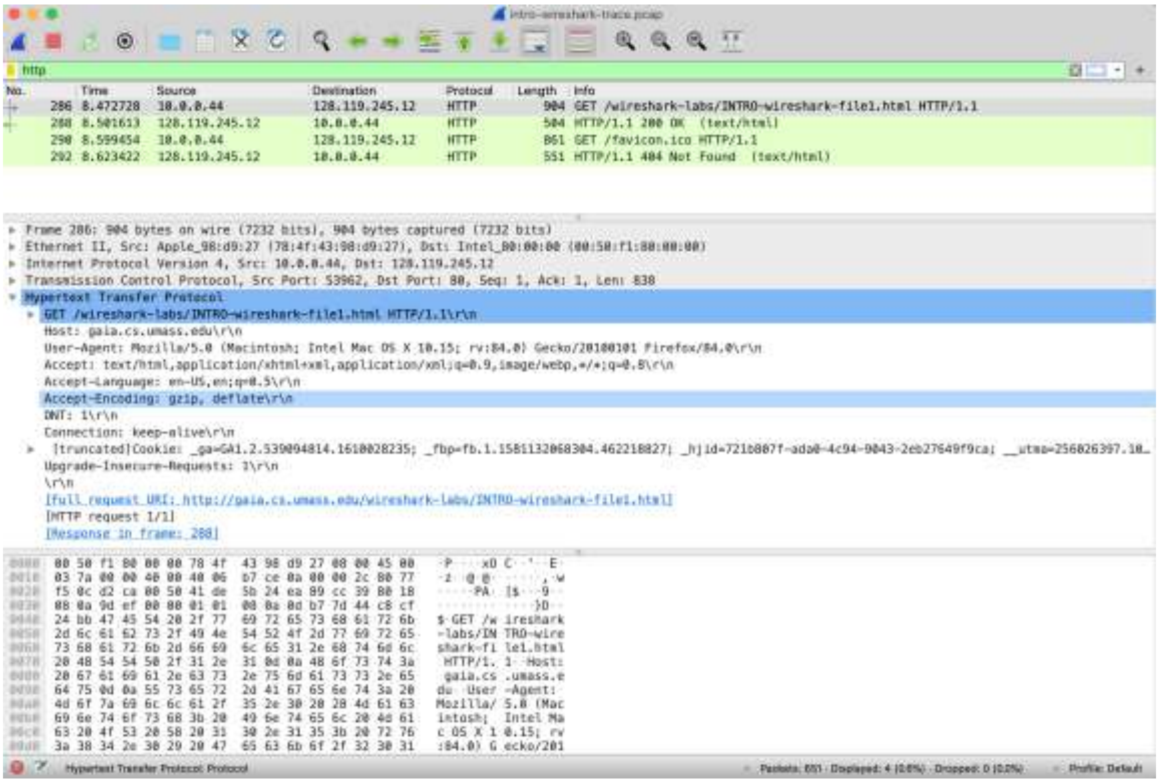
Description	IP	Packets	Packets/s
Intel(R) 82567LM Gigabit Network Connection	fe80::844a:1239:ab19:74ea	28	0
Microsoft	192.168.1.100	17	0

Figure 3: Wireshark Capture interface window, on a Windows computer

4. You'll 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. 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 2 will appear. This window shows the packets being captured. By selecting the Capture pulldown menu and selecting Stop, or by click on the red Stop square, you can stop packet capture. But don't stop packet capture yet. Let's capture some interesting packets first. 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 class later) 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. 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 or WiFi frames containing these HTTP messages (as well as all other frames passing through your Ethernet or WiFi 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 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 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 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.

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 your 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 2. Note also that in the Selected packet details window, we've 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 (WiFi) 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).



```
http
```

```
No.     Time      Source          Destination      Protocol  Length  Info
   286  8.472728  10.0.0.44       128.119.245.12   HTTP         904  GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
   288  8.501613  128.119.245.12  10.0.0.44        HTTP         504  HTTP/1.1 200 OK  (text/html)
   290  8.599454  10.0.0.44       128.119.245.12   HTTP         861  GET /favicon.ico HTTP/1.1
   292  8.623422  128.119.245.12  10.0.0.44        HTTP         551  HTTP/1.1 404 Not Found  (text/html)
```

```
▸ Frame 286: 904 bytes on wire (7232 bits), 904 bytes captured (7232 bits)
▸ Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Intel_80:00:00 (00:50:f1:80:00:00)
▸ Internet Protocol Version 4, Src: 10.0.0.44, Dst: 128.119.245.12
▸ Transmission Control Protocol, Src Port: 53962, Dst Port: 80, Seq: 1, Ack: 1, Len: 838
▾ Hypertext Transfer Protocol
  ▸ GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:84.0) Gecko/20100101 Firefox/84.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
  ▸ [truncated]Cookie: _ga=GA1.2.539094814.1610028235; _fbp=fb.1.1581132068304.462218827; _hjid=721b807f-ada0-4c94-9043-2eb27649f9ca; __utma=256026397.10…
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    [HTTP request 1/1]
    [Response in frame: 288]
```

```
0000  00 50 f1 80 00 00 78 4f  43 98 d9 27 08 00 45 00   ·P····xO C··'··E·
0010  03 7a 00 00 40 00 40 06  b7 ce 0a 00 00 2c 80 77   ·z··@·@· ·····,·w
0020  f5 0c d2 ca 00 50 41 de  5b 24 ea 89 cc 39 80 18   ·····PA· [$···9··
0030  08 0a 9d ef 00 00 01 01  08 0a 8d b7 7d 44 c8 cf   ········ ····}D··
0040  24 bb 47 45 54 20 2f 77  69 72 65 73 68 61 72 6b   $·GET /w ireshark
0050  2d 6c 61 62 73 2f 49 4e  54 52 4f 2d 77 69 72 65   -labs/IN TRO-wire
0060  73 68 61 72 6b 2d 66 69  6c 65 31 2e 68 74 6d 6c   shark-fi le1.html
0070  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1··Host:
0080  20 67 61 69 61 2e 63 73  2e 75 6d 61 73 73 2e 65    gaia.cs .umass.e
0090  64 75 0d 0a 55 73 65 72  2d 41 67 65 6e 74 3a 20   du··User -Agent: 
00a0  4d 6f 7a 69 6c 6c 61 2f  35 2e 30 20 28 4d 61 63   Mozilla/ 5.0 (Mac
00b0  69 6e 74 6f 73 68 3b 20  49 6e 74 65 6c 20 4d 61   intosh;  Intel Ma
00c0  63 20 4f 53 20 58 20 31  30 2e 31 35 3b 20 72 76   c OS X 1 0.15; rv
00d0  3a 38 34 2e 30 29 20 47  65 63 6b 6f 2f 32 30 31   :84.0) G ecko/201
```

```
Hypertext Transfer Protocol: Protocol          Packets: 651 · Displayed: 4 (0.6%) · Dropped: 0 (0.0%)          Profile: Default
```

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

9. 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 2 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. 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).

10. Exit Wireshark

---END---