Introduction to Wireshark lab quiz

Due Mar 13 at 17:00 **Points** 6 **Questions** 6 **Time Limit** None **Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	956 minutes	6 out of 6

Score for this attempt: **6** out of 6
Submitted Mar 13 at 16:09
This attempt took 956 minutes.

Page 1 of 9

In this course, we will be using the Wireshark network analyzer to study network traffic. This program is installed in the labs. You can also download a copy for your home computer from www.wireshark.org (http://www.wireshark.org/), if you wish.

The same applies if you get an answer correct and don't understand why, ask your tutor or ask on the Canvas Discussion Forum. The lab work is examinable. We won't examine how to run wireshark; but we will examine protocol behaviour and networking concepts that you will study while using wireshark.

We expect you to work on the labs each week and complete the previous lab before the next one is scheduled. **The labs will remain open for one week after the expected submission time after this time you will be able to review your answers and the solutions but will not be able to improve your mark**.

Please note that we have already allowed an extra week after the expected submission time in the deadlines. Further extensions can not be given as the answers become available at the deadline.

Each lab has associated questions you need to complete for credit. You do not need to hand anything in for the labs. The on line questions take the place of the handins. You may also leave your lab session and return to it later.

You are allowed to re-attempt the quiz and your mark will be the average (mean) of your attempts.

It is important that you understand the material in these labs. So if you get an answer wrong, you need to work out why and what the correct answer is. Feedback is given for common misconceptions, but you can also use the Canvas Discussion forum and tutorials to ask questions.

During these labs you will sometimes be capturing your own traffic and sometimes be using pre-captured files. B**e sure to follow the instructions on whether to use the pre-captured file or live capture when answering the questions**. Pre-captured files are used where packets will differ based on different capture environments or times. Since the answers to questions in these cases could be different, we use a pre-captured file to allow the correct answer to be predictable. These labs can be done from any computer with an Internet connection.

If you have any questions or want to discuss the network traffic you see in this or other Wireshark labs, please post to the Canvas Discussion forum. You are encouraged to use Wireshark to explore and deepen your understanding of network and application behaviour.

To access Wireshark in the CS labs you should choose Wireshark from the "Applications:Internet" menu.

Time to get started!

Adapted from Version: 2.0

"Tell me and I forget. Show me and I remember. Involve me and I understand."
Chinese proverb

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. This can be done in simulated scenarios or in a

Important External links (Click to explore):

1. Wireshark packet sniffer (http://www.wireshark.org/)
2. Wireshark User's Guide (http://www.wireshark.org/docs/wsug_html_chunked/)
3. Wireshark Manual (http://www.wireshark.org/docs/man-pages/)
4. FAQ (http://www.wireshark.org/faq.html)
5. Download Wireshark (http://www.wireshark.org/download.html)

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 (used in web browsing) 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," as shown in Figure 2.8 in the text.

We will be using the Wireshark packet sniffer [http://www.wireshark.org/ (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,

"real" network environment such as the Internet. The Java applets that accompany this text take the first approach. In these Wireshark labs[1], we'll take the latter approach. You'll be running various network applications in different scenarios using a computer on your desk, at home, or in a lab. You'll observe the network protocols in your computer "in action," interacting and exchanging messages with protocol entities executing elsewhere in the Internet. Thus, you and your computer will be an integral part of these "live" labs. You'll observe, and you'll learn, by doing.

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

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/** (http://www.wireshark.org/docs/wsug_html_chunked/)), man pages (**http://www.wireshark.org/docs/man-pages/** (http://www.wireshark.org/docs/man-pages/)), and a detailed FAQ (**http://www.wireshark.org/faq.html** (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

In order 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 installed within your operating system, when you install Wireshark. See **http://www.wireshark.org/download.html** (http://www.wireshark.org/download.html) for a list of supported operating systems and download sites

Wireshark is already installed in the school computing labs. If you want to download and install the Wireshark software on your own computer:

- Go to **http://www.wireshark.org/download.html** (http://www.wireshark.org/download.html) and download and install the Wireshark binary for your computer.
- 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 can also use the **Wireshark Lab forums** (%24CANVAS_OBJECT_REFERENCE%24/discussion_topics/ifd8fb4bfd6ae79fa3a2d7ee41a920d8d)

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

to see if other students are using the same platform as you. If all else fails, use the CS labs, where wireshark is supported.

---

[1]Earlier versions of these labs used the Ethereal packet analyzer. In May 2006, the developer of Ethereal joined a new company, and had to leave the Ethereal(r) trademarks behind. He then created the Wireshark network protocol analyzer, a successor to Ethereal(r). Since Ethereal(r) is no longer being actively maintained or developed, we have thus switched these labs over to Wireshark with the 4th edition of our text.

every link-layer frame that is sent from or received by your computer. Recall from the discussion from section 1.5 in the text (Figure 1.24[2]) 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. 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.

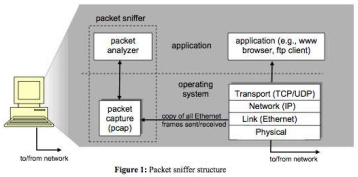


Figure 1: Packet sniffer structure

Page 3 of 9

Install Troubleshooting

If you are having difficulty running wireshark in the lab, please double check you have followed the instructions (just typing wireshark will not work). If you have followed the instructions, e-mail or see you instructor with as much information as possible (student number, exactly what you tried and the exact result including any error messages - no matter how cryptic).

If you are installing on your own computer, the wireshark website **http://www.wireshark.org** (http://www.wireshark.org/) has frequently asked question (FAQ) pages and forums where you can get assistance. You may also want to ask on the **Wireshark Lab forums** (%24CANVAS_OBJECT_REFERENCE%24/discussion_topics/ifd8fb4bfd6ae79fa3a2d7ee41a920d8d) to see if others have successfully installed on your hardware type or are experiencing similar problems.

Page 4 of 9

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will be displayed.

The **Open** section lists recently opened packet capture files, in case you want to re-open them. The Capture section lists the network interfaces on your computer. You may have more than one if there is more than one interface that your computer can receive network traffic on (in the example above, I have 4). The line next to them is an indication of the network traffic arriving on the interface. You can see that although I have 4 interfaces, only one (my wi-fi connection) is actually active and receiving network traffic. Double click on interface that is receiving traffic and this will start wireshark recording the traffic on this interface.

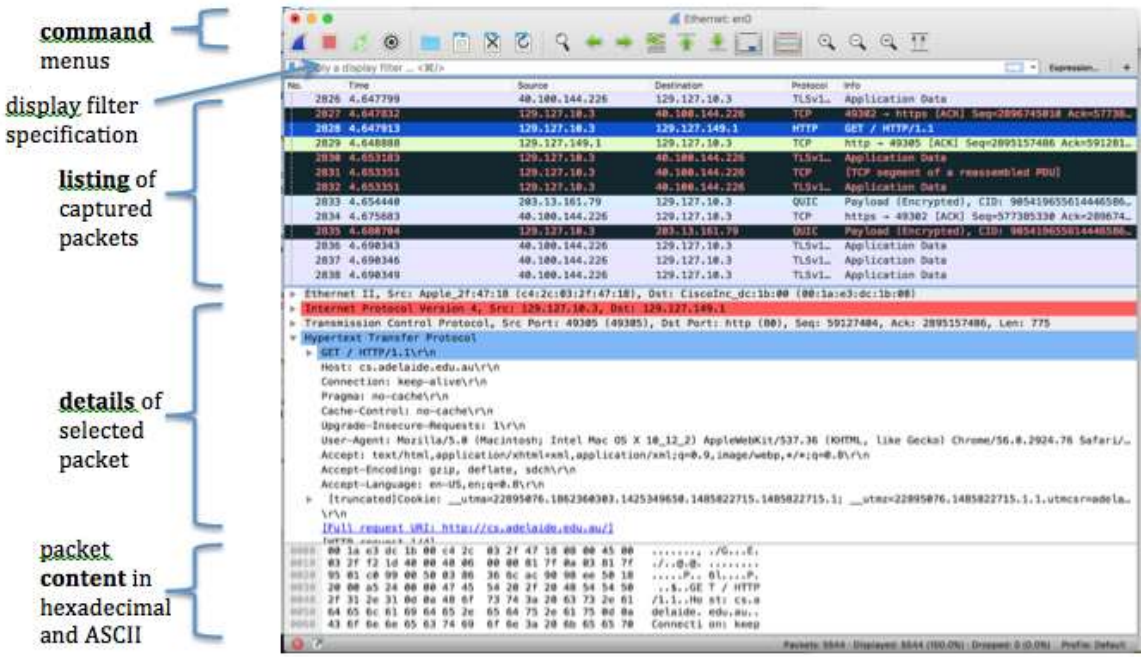


Figure 2: Wireshark Graphical User Interface

The Wireshark interface has five major components:

- The **command menus** include commands for starting/stopping captures, editing preferences, saving files, etc. Of interest to us now are the File (blue folder) and Capture (shark fin) 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.
- 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 (assuming the packet was sent/receiverd 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. 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.

The best way to learn about any new piece of software is to try it out! We'll assume that your computer is connected to the Internet via a wired Ethernet interface. Do the following

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

2. Start up the Wireshark software. Wireshark is in the "Applications" menu in the labs under the "Internet" sub menu.

You will initially see a window with links to documentation, tutorials and capture options. The first thing you need to do is select which network interface you want to capture packets from. A computer may have multiple network interfaces, for example, a wireless network interface and a wired ethernet interface. You can see the activity of the different interfaces by looking at the line graphs next to the interfaces. This will show you how many packets of data are being received on the different interfaces. Choose one of the interfaces that is receiving traffic and press the "Start" button (the control button with the blue shark fin). You will see a window similar to that shown in Figure 2. - all packets being sent/received from/by your computer are now being captured by Wireshark!

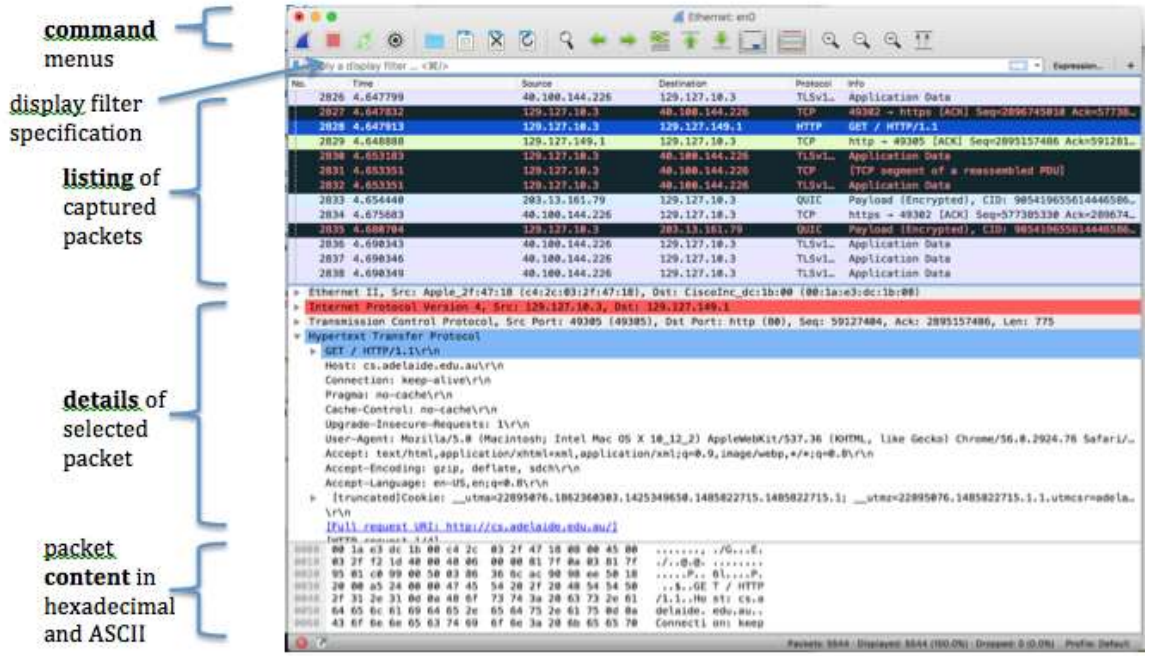


Figure 2: Wireshark Graphical User Interface

3. While Wireshark is running, got to the URL:

http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html (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, as discussed in section 2.2 of the text. The Ethernet frames containing these HTTP messages will be captured by Wireshark.

4. After your browser has displayed the INTRO-wireshark-file1.html page, stop Wireshark packet capture by selecting stop icon (red square) in the Wireshark command menus. The Wireshark window will display all packets captured since you started 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 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. Even though the only action you took was to download a web page, there were evidently many other network protocols running on your computer that are unseen by the user. 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”!

Question 1

1 / 1 pts

In the packet listing window under the "Protocol" column you will see several acronyms (often ending in 'P' for 'protocol) which is the protocol being used for each of the captured packets. Internet applications define their own protocols for communication. Here are a few examples you're likely to come across:

POP (Post Office Protocol) is used to download e-mail

IMAP (Internet Message Access Protocol) is the other protocol commonly used to download e-mail. It offers many more options for managing e-mails than POP.

TCP (transmission control protocol) is used in the Internet to reliably transfer data between end systems (ie the source and destination computers)

HTTP (hypertext transfer protocol) is the protocol used by web clients and servers to talk to each other

NFS (network file system) is a protocol that allows users to access and update files from remote computers

You'll see at least some of the above protocols in your packet capture.

The Internet Engineering Task Force (IETF) maintains the specifications of Internet protocols. You can search for these specifications at **www.ietf.org** **(http://www.ietf.org/)** They are written up as Request For Comments (RFCs).

Wikipedia is also generally accurate in Internet protocol information and provides links to the relevant RFCs. So **http://en.wikipedia.org/wiki/List_of_network_protocols** **(http://en.wikipedia.org/wiki/List_of_network_protocols)** is a good place to start for an overview of a protocols.

Let's look at what is happening in the CS labs.....

Download the file **intro-capture-1.pcapng** which is a wireshark capture from the EM108/109. Open this file in wireshark by selecting the folder icon in the wireshark control menu. Look for the protocols listed above. You can get wireshark to show only packets from a particular protocol by typing the protocol name (lower case) in the "Filter:" box. For example, type http in the filter box and click 'Apply' (the arrow at the end of the filter box). Only the http packets will appear and all others will be hidden. Click 'Clear' (the X at the end of the filter box) to remove the filter.

Explore some of the protocols that have been captured and what are they used for. Remember you can filter by typing the protocol you are looking for in the display filter (protocol must be typed in lower case).

Which of the following protocols does ***not*** appear in the packet-listing window in the packet capture file intro-capture-1.pcapng?

- [ ] Network File System Protocol

Correct!

- [x] Network Time Protocol

NTP (Network Time Protocol) is very commonly found on networks and is used by hosts to get the current time and keep their clocks in sync. However, it doesn't appear in this short capture.

- [ ] Transmission Control Protocol
- [ ] Hypertext Transfer Protocol

Question 2

1 / 1 pts

Which of the three protocols: NFS, HTTP or DNS is generating the most packet traffic in the capture?

Correct!

NFS

NFS is an open standard remote file access protocol similar (SMB (Samba) and AFP (Apple) are other file access protocols you may have come across). Managing remote changes to files needs frequent communication to keep local and remote copies synchronised.

Correct Answers

nfs

NFS

Question 3

1 / 1 pts

Page 7 of 9

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.

Look at the elapsed time from when the ***second*** HTTP GET message was sent until the HTTP OK reply was received in the file capture **intro-capture-1.pcapng** The first HTTP GET is not forwarded. It is responded to with a request for authentication. The second HTTP GET request is the one actually sent to the gaia web server to download the page.

Consider where network delays come from - distance to get to gaia.cs.umass.edu.au (in Massachusetts, East Coast USA) (***propagation delay***), competition with other packets for network bandwidth and routers (***queuing delay***), time to process the packets at the end hosts and intermediate routers - determining where to send, etc. (***processing delay***) and the time needed for the sending host to transmit/signal the bits of the packet into the network (***transmission delay***). These 4 delays make up the overall delay a packet experiences on a network.

Using a car analogy, ***propagation delay*** depends on how fast you can drive on the road (***propagation speed***) and how far you have to drive. ***queuing delay*** depends on how many other cars are on the road and how many lanes the road has (***bandwidth***) and whether any of them choose to go through the same tollbooth at the same time as you (***router queuing***) and how fast the toll operator is able to take the toll and give change or the electronic payment system is able to debit your card (***processing delay***) and finally, how long it takes for you to get your entire car onto the road from the time you start to pull out into the road (***transmission delay)***

(https://forums.cs.adelaide.edu.au/forums/draftfile.php/26897/user/draft/496807316/intro-capture-1.pcapng) Adelaide is approximately 18,000 kilometers from Amherst Massachusetts, U.S.A. (where the gaia server is). The propagation rate through optical fibre and copper cable is approximately 2 x 10^8 meters/sec. Approximately what percentage of the delay from sending the request to gaia to receiving the response is due to the time for the signal to propagate (travel) between Adelaide and Amherst (request) and back (response)?

Correct!

- [x] 35%

  Correct! Is this delay typical? Have a look at the delays you see in your own web downloads at Uni and at home to different countries and states. Do other delays (queuing, processing, transmission) typically make up 65% of the total delay.

- [ ] 50%
- [ ] 17%
- [ ] 1%

Question 4

1 / 1 pts

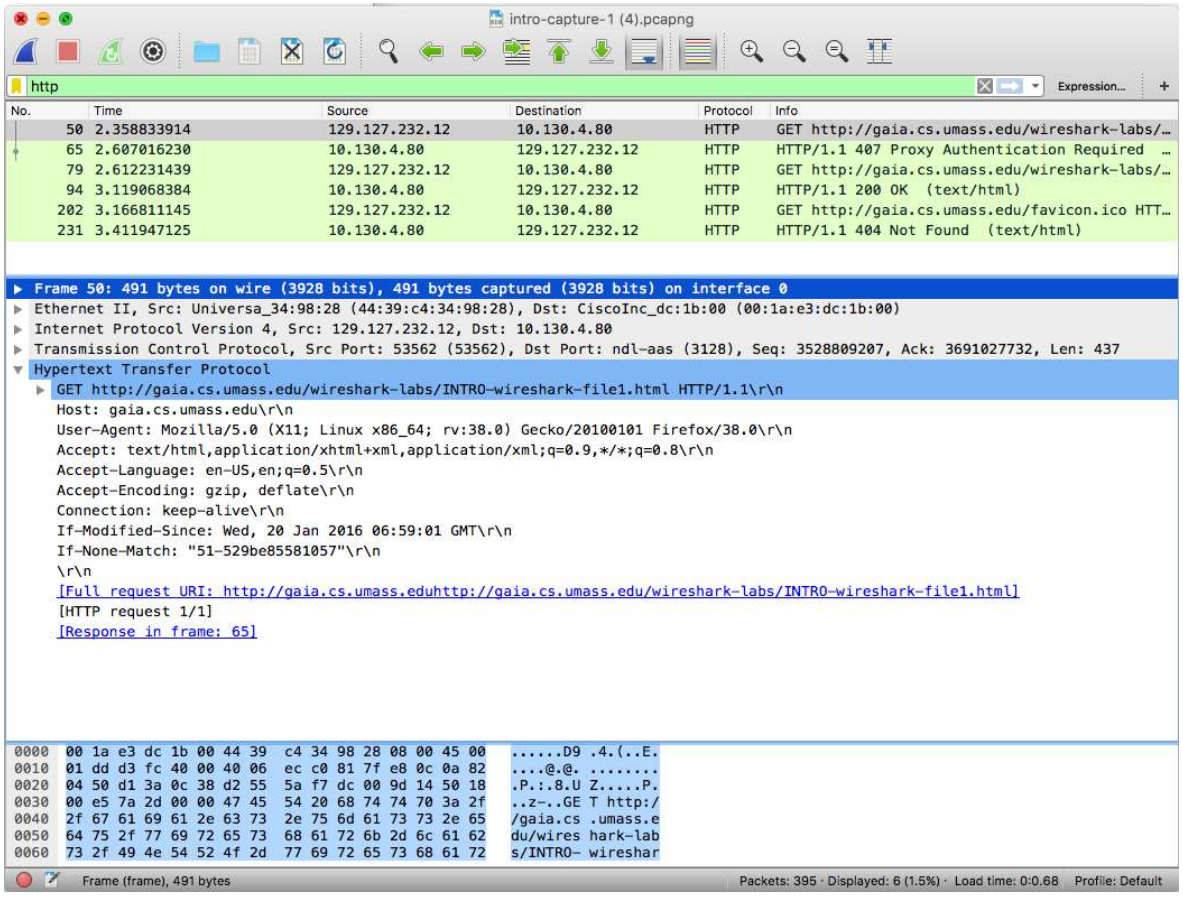
Page 8 of 9

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 gaia.cs.umass.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 triangles to the left side of the packet details window, you can hide or show the details of the Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information carried inside the packet.

Show the details of the HTTP protocol for the second HTTP GET request. Your Wireshark display should now look roughly as shown in Figure 5. (Note, in particular, the hidden information for all protocols except HTTP, and the detailed protocol information for HTTP in the packet detail window).



Q1) Looking at the Source and Destination addresses in the listing of packets captured (or alternatively in the Internet Protocol section of the packet details), what is the Internet address (in '[dotted decimal]

(http://en.wikipedia.org/wiki/Dot-decimal_notation) ' format) of the computer that made the web page request in the captured file intro-capture-1.pcapng?

Correct!

129.127.232.12

Correct Answers

129.127.232.12

Question 5

1 / 1 pts

Page 9 of 9

HTTP requests from most University computers are required to use a ***Web Proxy Server***. We'll discuss proxies later in the course but they serve as a cache for frequently accessed web pages and can also monitor web usage.

Look at the Internet address where the response was sent from.

This Internet address is a ***private address*** (https://tools.ietf.org/html/rfc1918) (see section 3 for the Internet addresses which are private), which means it must be part of an internal network that the lab computer is on. So this address is part of the University of Adelaide network, not University of Massachusetts where gaia is. This response is coming from the university web proxy.

Q2) What is the private Internet address of the proxy that returns the reply from gaia.cs.umass.edu?

Type the IP address in 'dotted decimal' format.

Correct!

10.130.4.80

Correct Answers

10.130.4.80

Question 6

1 / 1 pts

Q3) Look at the HTTP ok response packet. Open the Hypertext Transfer Protocol details. What web server is running on the web proxy? Type in the value in the server header. You only need to type the **server name** and **version number** as seen in wireshark after the colon, for example: IIS/1.2

Correct!

Apache/2.4.6

Correct Answers

Apache/2.4.6

Apache

You have seen all questions.

You can now select submit to mark your answers.

Remember you can retake the lesson as many times as you like and your average mark will be recorded. Just click on the quiz again. So if you got any questions incorrect, have a look at the feedback and have another go.

Quiz Score: **6** out of 6