**Lab - Using Wireshark to Capture Network Traffic**

1. Objectives

Capture and Analyze UDP Data in Wireshark

1. Background / Scenario

Wireshark is a Packet Sniffer. A Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures (“sniffs”) packets 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 packets. 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.

**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 every link-layer

frame that is sent from or received by your computer. Messages exchanged by higher

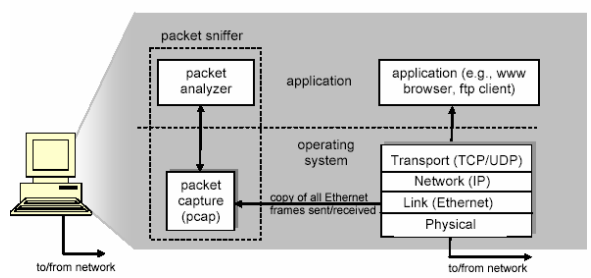
layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually

encapsulated in link-layer frames that are transmitted over physical media such as an

Ethernet cable.

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



**Figure 1: Packet Sniffer Structure**

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 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”.

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. (Technically speaking, Wireshark is a packet

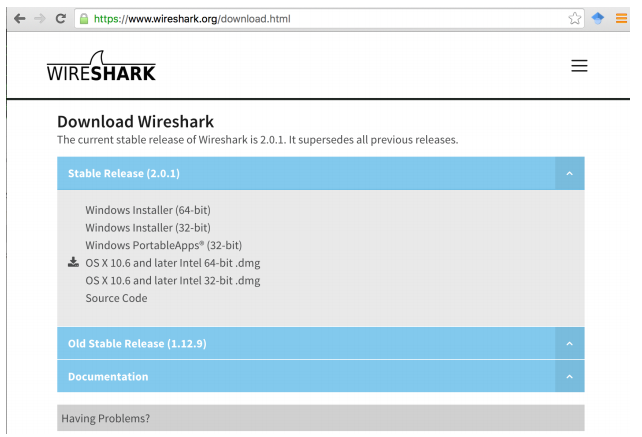
analyzer that uses a packet capture library in your computer). *Wireshark is a free*

*network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers.*

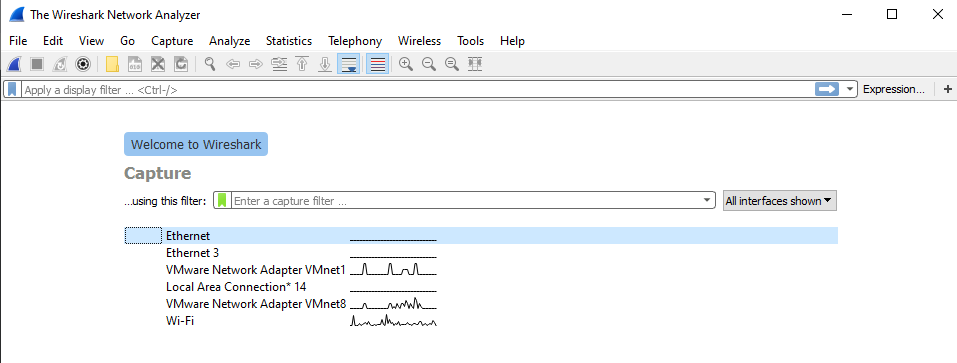
**Required Resources**

* 1 PC (Windows based system)
* Internet connection

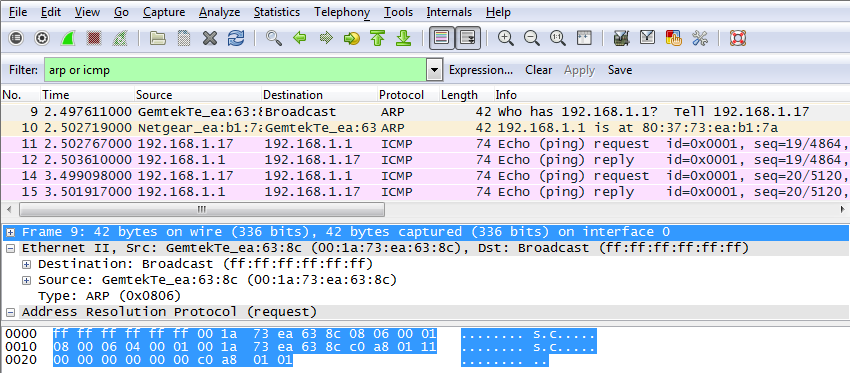
Wireshark can also be downloaded from here: <https://www.wireshark.org/download.html>

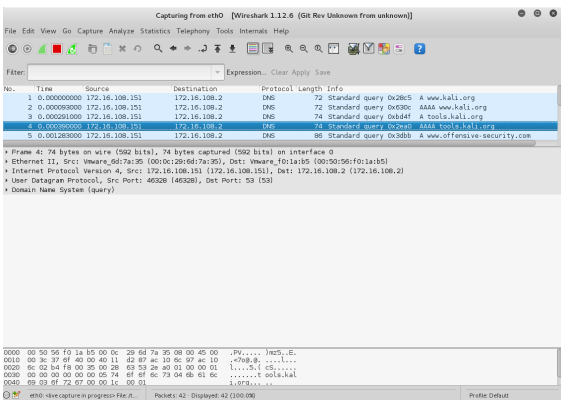


**Figure 1: Download Page of Wireshark**



**Figure 2: Initial Graphic interface for Wireshark**





**Figure 3: Versions of Wireshark Capture Interface**

Capture and Analyze UDP Data in Wireshark

You will need to download and install Wireshark for your Operating System at the following website: <https://www.wireshark.org/download.html>

Please watch following YouTube videos for quick introduction into Wireshark, the most famous open source traffic/protocol analyzer.

<https://www.youtube.com/watch?v=jvuiI1Leg6w&t=802s>

<https://www.youtube.com/watch?v=0ELCdQaHELs>

I am also including resources in Canvas that briefly talks about the way Wireshark is typically used in analyzing VoIP protocols.

You will use Wireshark to capture packets from your Network Interface Card. Here is what you need to do:

* Close all opened running windows and applications (web browser, email program and other

applications that typically send data to the Internet). This will minimize the amount of traffic you

are going to capture.

* Open Wireshark program and wait until you are through with your next step.
* Next step: Open any web browser you might have installed on your computer.
* Now, you can start capturing packets on your active network card.
* Type www.google.com In the address bar (URL bar) of your web browser and leave it open for

about 5-10 seconds.

* Stop Wireshark capture and save it as Assignment1\_wireshark.pcap file.

(Kaltura screen capture illustrates this process using my WiFi network interface. You can find it

in the Blackboard under this document.)

* At this point in time you will be able to start analyzing packets and protocols for this assignment.

Answer following questions as part of your homework delivery. Go to the end of the document for submission details:

1. What is the duration of your capture in seconds? What about the start and end time of the

capture expressed in hh:mm:ss? **(Take a Screenshot of this screen)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. How many protocols do you see in the protocol window? Name some of these for me? You can get these info from the “protocol” field. You can sort on this or any other field in the window.

You can also add or delete fields from the list.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. How many IPv4 or IPv6 conversations do you have in your capture? You can get these if you investigate

Statistics -> Conversations.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. What is the IP address of the DNS server you are connecting to?

To minimize the search time you should search for a specific string, in this case “google” since

we ended up typing www.google.com in the web browser and it is what the system needs to

resolve with DNS to get to the appropriate IP address of the Google server servicing your search

request. To find a string within a packet, click on Edit > Find Packet. Under "Find By:" select

"string" and enter your search string in the text entry box.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. What is the IP address of the Google server?

Once you locate DNS query within all captured packets, you will be able to easily find this

address is well.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Type udp.port in *Apply a display filter … <Ctrl-/>?* field and click Enter.

List protocols in ”Protocol” field that you see now.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now type udp.port in *Apply a display filter … <Ctrl-/>?* field and click Enter.

List protocols in ”Protocol” field that you see now.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. What is Checksum in UDP header used for and can it be used for reliable data delivery?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. What is Checksum field in UDP header used for and can it be used for reliable data delivery?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9. What is TOS field in IP header used for and can it be used for reliable data delivery?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10. What is Sequence Number field in TCP header used for?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

11. What is timestamp field in UDP header used for?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

12. Elaborate how router uses TCP acknowledgment for reliable packet delivery?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please return your completed assignments using Blackboard.

**You should answer these questions in the Answer Template that is provided in your Assignment folder.**

Save your Answer Template using the convention of [your first initial] + [your last name] + “\_Lab3”.

For example: Joe Smith will save his file template as JSmith\_Lab3.doc .

The following documents are required as part of your submission.

* **This answer Template completed**
* **Wireshark Capture Assignment1\_wireshark.pcap**