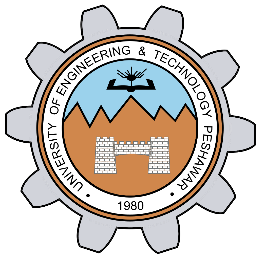
“**Introduction WireShark Lab**”

**LAB REPORT # 05**



# Spring 2023

**CSE 303L: Data Communication and Networks**

Submitted by: **Maaz Habib**

Registration # **20PWCSE1952**

Class Section: **C**

Submitted to:

**Engr. Yair Saleem Afridi**

17-05-2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar.

**CSE 303L: Data Communication and Computer Networks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Demonstration of**  **Concepts** | **Poor (Does not meet**  **expectation (1))**    The student failed to demonstrate a clear understanding of the  assignment concepts | **Fair (Meet Expectation**  **(2-3))**    The student demonstrated a clear understanding of some of the assignment concepts | **Good (Exceeds**  **Expectation (4-5)**    The student demonstrated a clear understanding of the assignment concepts | **Score**      **30%** |  |
| **Accuracy** | The student mis-configured enough network settings that the lab computer couldn't function properly on the network | The student configured enough network settings that the lab computer partially functioned on the network | The student configured the network settings that the lab computer fully functioned on the network | **30%** |  |
| **Following Directions** | The student clearly failed to follow the verbal and written instructions to successfully complete the lab | The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab | The student followed the verbal and written instructions to successfully complete  requirements of the lab | **20%** |  |
| **Time Utilization** | The student failed to complete even part of the lab in the allotted amount of time | The student failed to complete the entire lab in the allotted amount of time | The student completed the lab in its entirety in the allotted amount of time |  | **20%** |

# Lab 05

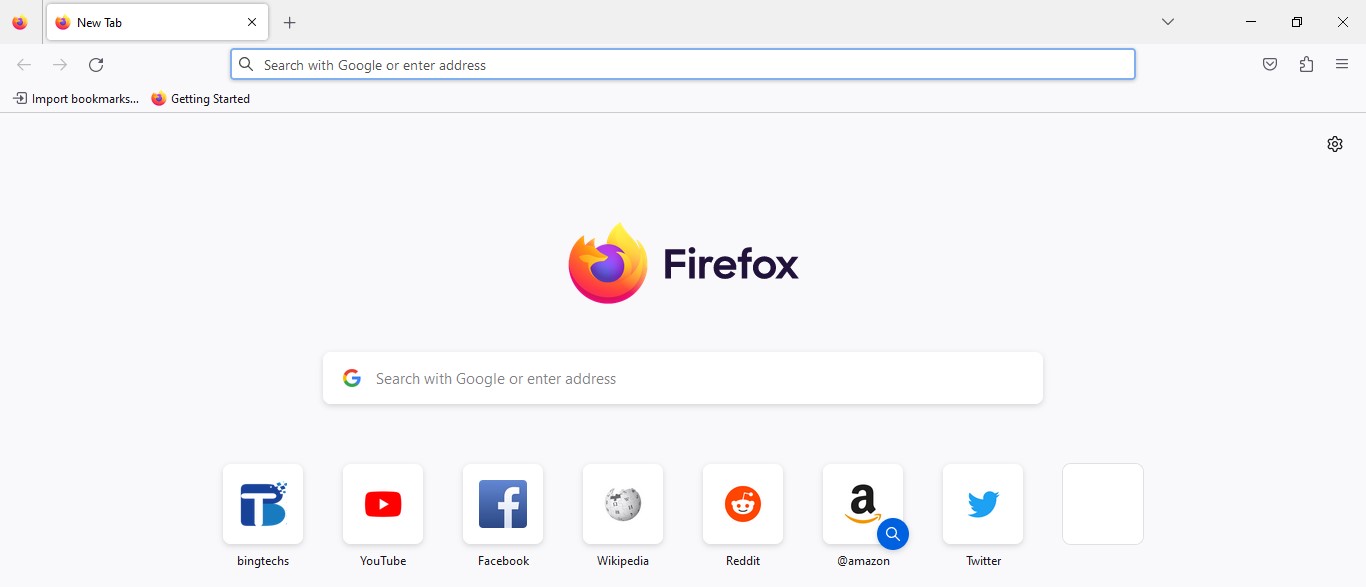
**Introduction WireShark Lab**

*“Tell me and I forget. Show me and I remember. Involve me and I understand.”* Chinese proverb

Taking Wireshark for a Test Run

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. Indeed, I recommend that you do this first lab on a computer that has a wired Ethernet connection, rather than just a wireless connection. Do the following

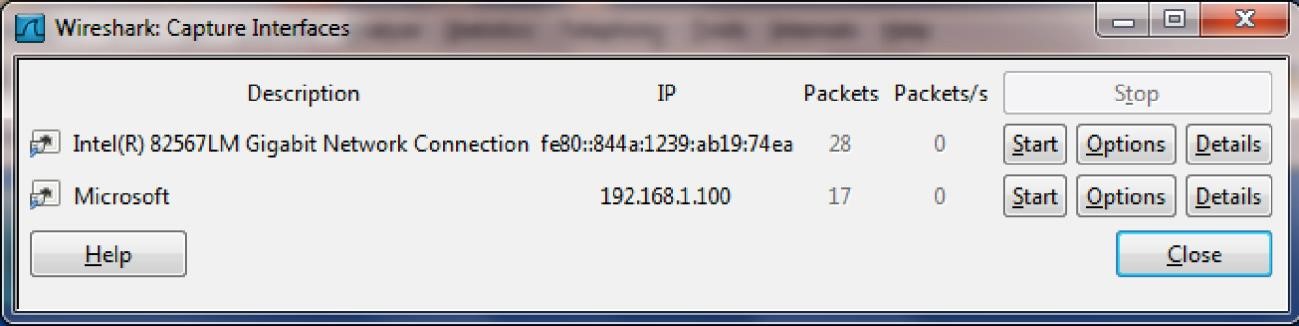
1. Start up your favorite web browser, which will display your selected homepage.



1. 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.

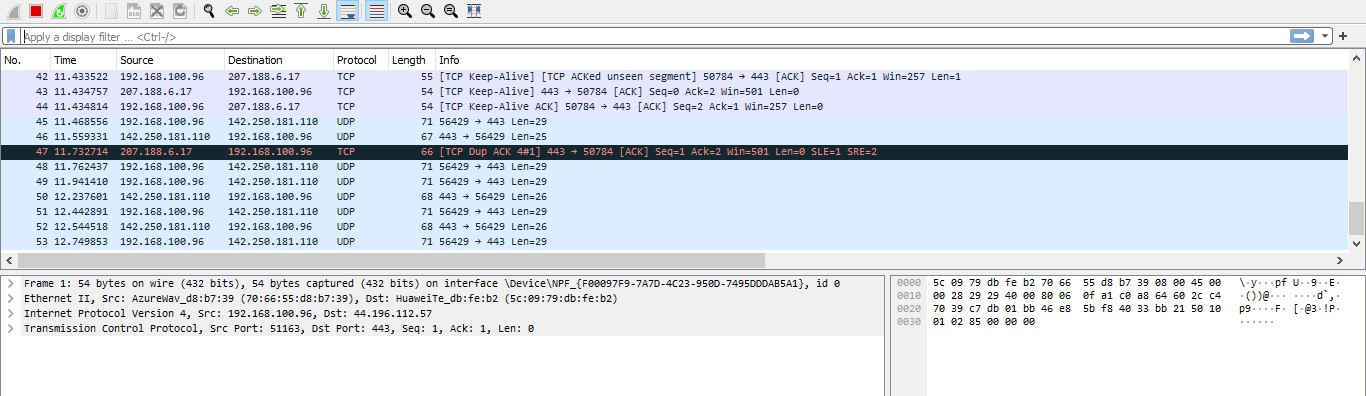
1. To begin packet capture, select the Capture pull down menu and select *Interfaces.*

This will cause the “Wireshark: Capture Interfaces” window to be displayed, as shown in Figure 4 below.



**Figure 4:** Wireshark Capture Interface Window

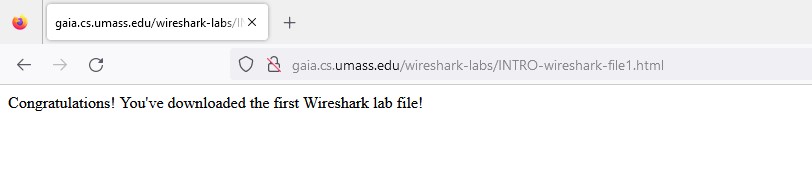
1. You’ll see a list of the interfaces on your computer as well as a count of the packets that have been observed on that interface so far. Click on *Start* for the on which you want to begin packet capture (in the case, the Gigabit network Connection). Packet capture will now begin - Wireshark is now capturing all packets being sent/received from/by your computer



1. 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*, 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

1. 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 frames containing these HTTP messages (as well as all other frames passing through your Ethernet adapter) will be captured by Wireshark.



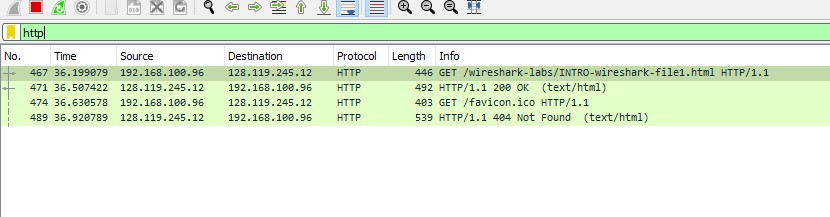
1. After your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), stop Wireshark packet capture by selecting stop in the Wireshark capture window. The main Wireshark window should now look similar to Figure 3. You now have live packet data that contains 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.

For now, you should just be aware that there is often much more going on than “meet’s the eye”!

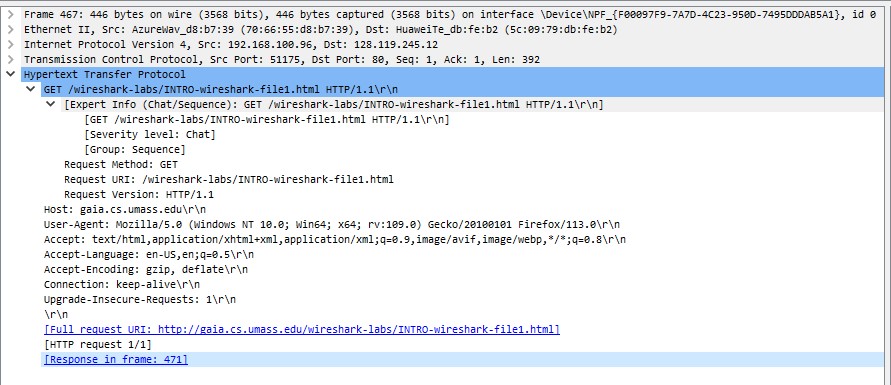
1. Type in “http” (without the quotes, and in lower case – all protocol names are in lower case in

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.



1. 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 Figure 3) 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 window2. By clicking on ‘+’ 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 5. (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).

1. Exit Wireshark

