

# Lab Exercise 1: Sniffing HTTP Traffic with Wireshark (10 pts.)

## What You Need

- A Computer running any OS. The instructions are written for Windows 7.

## Installing Wireshark

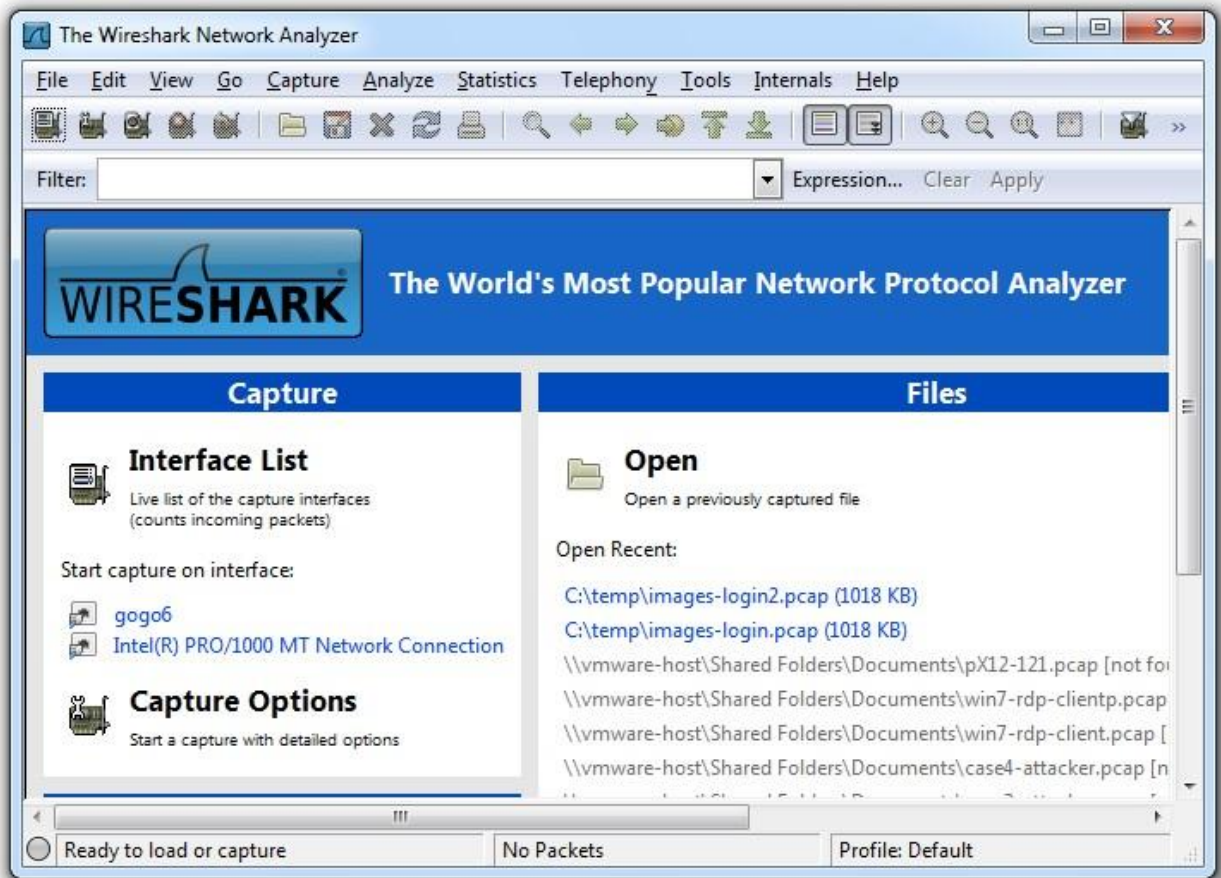
Click the **Start** button. In the Search box, type **WIRESHARK**. If Wireshark is found, that means it's already installed and you can skip the following steps. If it is not found, go to <http://www.wireshark.org/download.html> to download and install it. It will also install WinPCap.

## Capturing All Network Traffic With WireShark

Click the **Start** button. In the Search box, type **WIRE**

At the top of the menu, a Wireshark item appears. Right-click **Wireshark** and click "**Run as Administrator**". If a User Account Control box appears, allow the program to run.

Wiresharks opens, as shown below.



In the upper left of the Wireshark window, click "**Interface List**".

A list of network interfaces appears. Each interface has an IP address and a count of Packets, as shown below.

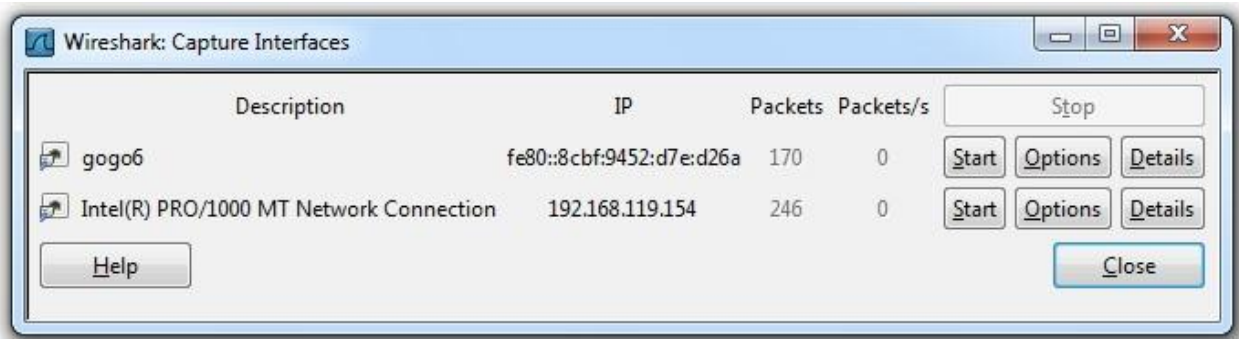


At first, all the IP addresses start with fe80: -- these are "Link-Local IPv6 Addresses", and they are very useful.

Find the network interface with the most rapidly increasing number of packets--this is the interface that connects to the Internet. Click its IP address.

Wireshark will show the other addresses of this interface.

After one or more clicks, you should see the IPv4 address of the interface, which is four values separated by periods, as shown below:

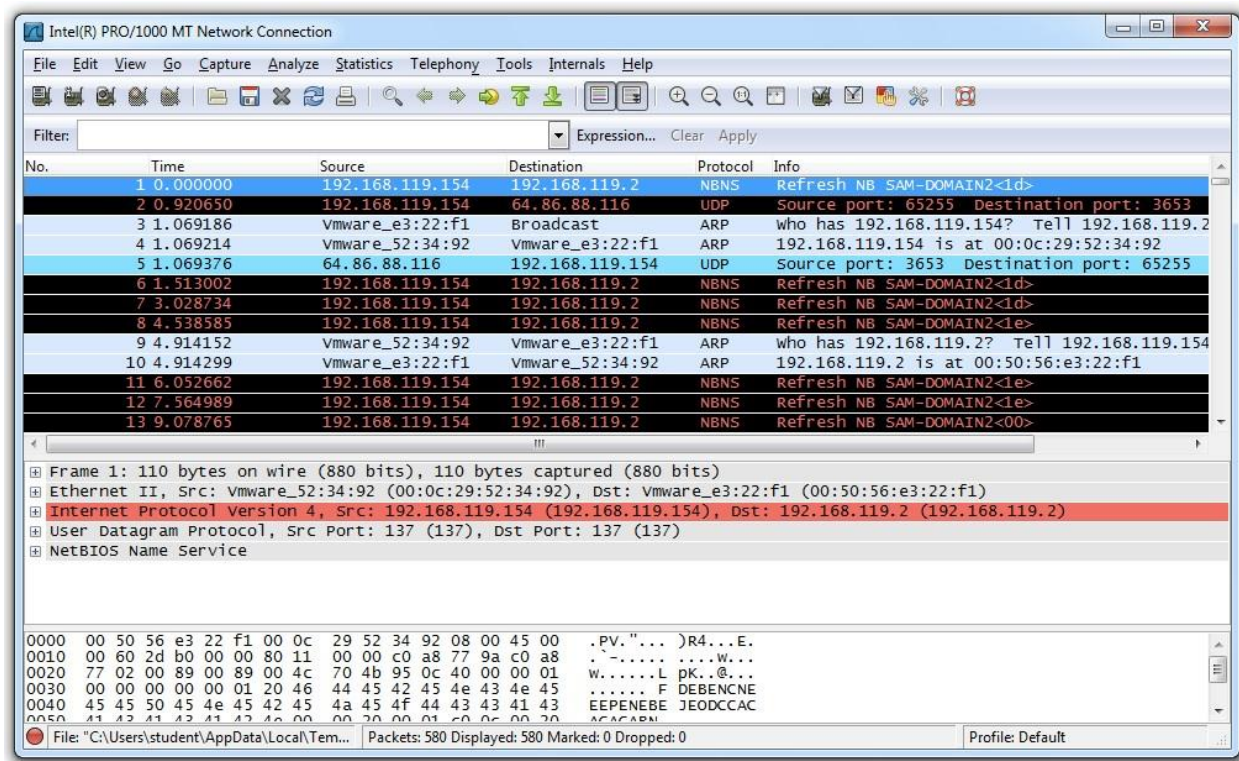


Click the **Start** button next to the interface that connects to the Internet.

You should see a lot of text scrolling by, as shown below on this page. Each line in the upper pane summarizes one frame (or packet).

Find these columns in the Wireshark window:

- **No** Frame Number
- **Time** Time in seconds that the frame was captured
- **Source** Source address of the frame
- **Destination** Destination address of the frame
- **Protocol** Protocol of the frame
- **Info** Other information



Notice that some lines show Broadcast in the Destination column. Broadcast traffic is common on networks as network devices alert one another of their presence. But it's usually not very interesting. To make Wireshark easier to use, you can Filter the traffic, to see only the interesting packets.

## Capturing HTTP Traffic With WireShark

At the upper left of the Wireshark window, in the "Filter" bar, type `http`  
 Press the Enter key on the keyboard.

Wireshark now just sits there, with little or no visible traffic, because it is ignoring all the nonHTTP packets.

## Loading the CCSF Web Page

In a Web browser, go to [www.ccsf.edu](http://www.ccsf.edu)

You should see a lot of text scroll by in the Wireshark window.

From the Wireshark menu bar, click **Capture, Stop**.

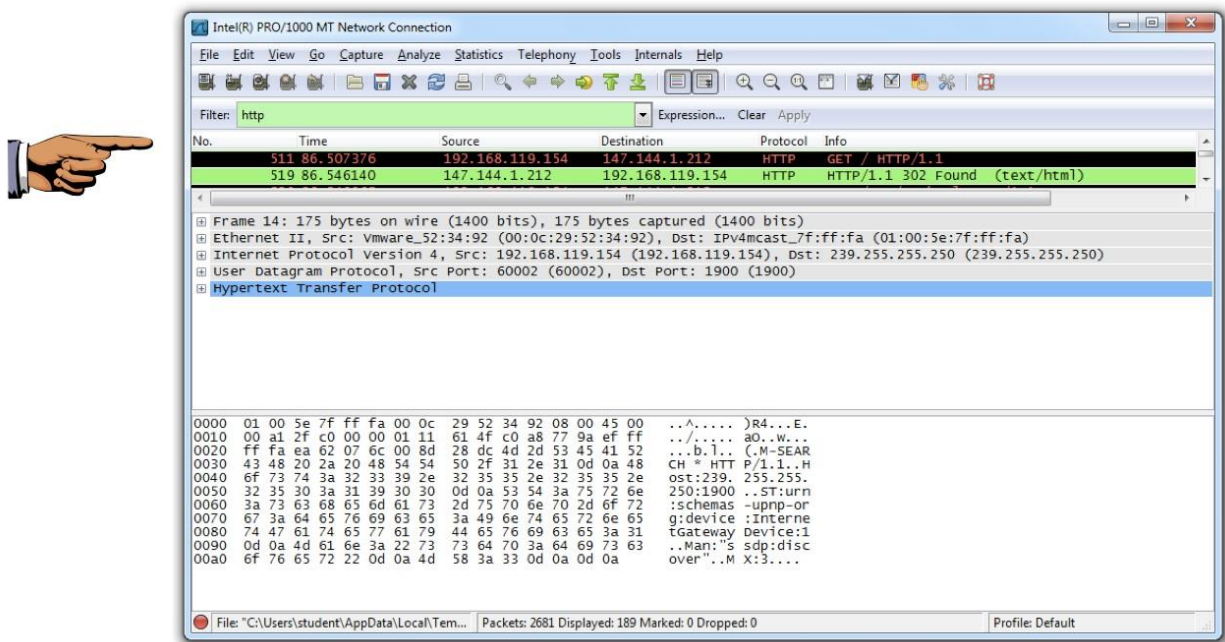
In the Wireshark window, scroll back to the top of the packet list.

# Understanding HTTP GET Packets

The CCSF Web server has an IPv4 address of 147.144.1.212.

Find a packet with a Destination of 147.144.1.212 (the CCSF Web server), and "**GET / HTTP/1.1**" in the Info column. In the example below, it is packet number 511.

Find the response to that packet, with a Source of 147.144.1.212, and "**HTTP/1.1 302 Found (text/html)**" in the Info column. In the example below, it is packet number 519.



## Saving the Screen Image

Resize the panes in Wireshark so that only these two packets are visible:

"**GET / HTTP/1.1**" and "**HTTP/1.1 302 Found (text/html)**", as shown above.

On your keyboard, press the PrntScr key.

Click **Start**, type in **PAINT**, and open Paint.

Press **Ctrl+V** to paste in the image of your desktop.

**YOU MUST SUBMIT WHOLE-DESKTOP IMAGES TO GET FULL CREDIT.**

Save the image with a filename of "**LabExer1\_[YOUR NAME]**".



# Example: LabExer1\_DonErickBonus Turning In Your Project

Upload the Lab Exercise to Canvas.

The image displays a Wireshark packet capture and a corresponding web browser window. The Wireshark interface shows a list of captured packets, with the selected packet (No. 8127) being an HTTP GET request to `/jru_wp/wp-content/uploads/2019/02/jru-faviicon.png`. The packet details pane shows the full HTTP request, including headers like `Host: jru.edu`, `Connection: keep-alive`, `Pragma: no-cache`, `Cache-Control: no-cache`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36`, `Referer: http://www.jru.edu/\`, `Accept-Encoding: gzip, deflate`, `Accept-Language: en-US,en;q=0.9`, and a `Cookie` string.

The browser window shows the Jose Rizal University website with a banner for **CORONAVIRUS (COVID-19)** and a button labeled **Updates**. Below the banner, there are sections for **PROGRAM OFFERINGS**, **APPLY TO JRU**, **INTERNATIONAL**, and **EXPLORE JRU**.

No.	Time	Source	Destination	Protocol	Length	Info
8096	38.707326	192.168.254.103	23.215.177.67	HTTP	385	GET /pr/492350f6-3a01-4f97-b9c0-c7c6ddf67/
8127	38.897345	192.168.254.103	216.92.24.250	HTTP	558	GET /jru_wp/wp-content/uploads/2019/02/jru-faviicon.png HTTP/1.1 200 OK (PNG)
8242	39.381395	216.92.24.250	192.168.254.103	HTTP	978	HTTP/1.1 200 OK (PNG)
8273	39.537373	23.215.177.67	192.168.254.103	HTTP	1357	HTTP/1.1 206 Partial Content

Frame 8127: 558 bytes on wire (4464 bits), 558 bytes captured (4464 bits) on interface \Device\NPF\_{75F01D31-B5FA-4666-B...}

Ethernet II, Src: LiteonTe\_5f:5c:71 (30:52:cb:5f:5c:71), Dst: Shenzhen\_b1:74:50 (18:c5:01:b1:74:50)

Internet Protocol Version 4, Src: 192.168.254.103, Dst: 216.92.24.250

Transmission Control Protocol, Src Port: 49969, Dst Port: 80, Seq: 1, Ack: 1, Len: 504

Hypertext Transfer Protocol

GET /jru\_wp/wp-content/uploads/2019/02/jru-faviicon.png HTTP/1.1\r\n

Host: jru.edu\r\n

Connection: keep-alive\r\n

Pragma: no-cache\r\n

Cache-Control: no-cache\r\n

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36\r\n

Accept: image/webp,image/apng,image/\*,\*/\*;q=0.8\r\n

Referer: http://www.jru.edu/\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: en-US,en;q=0.9\r\n

Cookie: \_ga=GA1.2.936935648.1598136190; \_gid=GA1.2.908704427.1598136190; \_gat=1\r\n

0030 02 02 5e 7c 00 00 47 45 54 20 2f 6a 72 75 5f 77 ..^|...GE T /jru\_w

0040 70 2f 77 70 2d 63 6f 6e 74 65 6e 74 2f 75 70 6c p/wp-con tent/upl

0050 6f 61 64 73 2f 32 30 31 39 2f 30 32 2f 6a 72 75 oads/201 9/02/jru

0060 2d 66 61 76 69 69 63 6f 6e 2e 70 6e 67 20 48 54 -faviico n.png HT

0070 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 6a 72 TP/1.1.. Host: jr

0080 75 2e 65 64 75 0d 0a 43 6f 6e 6e 65 63 74 69 6f u.edu..C connectio

0090 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 50 n: keep- alive .P

00a0 72 61 67 6d 61 3a 20 6e 6f 2d 63 61 63 68 65 0d ragma: n o-cache-

00b0 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 -Cache-C ontrol:

00c0 6e 6f 2d 63 61 63 68 65 0d 0a 55 73 65 72 2d 41 no-cache ..User-A

00d0 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e gent: Mo zilla/5.

00e0 30 20 28 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 0 (Windo ws NT 10

00f0 2e 30 3b 20 57 69 6e 36 34 3b 20 78 36 34 29 20 .0; Win6 4; x64)

Hypertext Transfer Protocol (http), 504 bytes

Packets: 9480 · Displayed: 264 (2.8%) · Dropped: 0 (0.0%) · Profile: Default