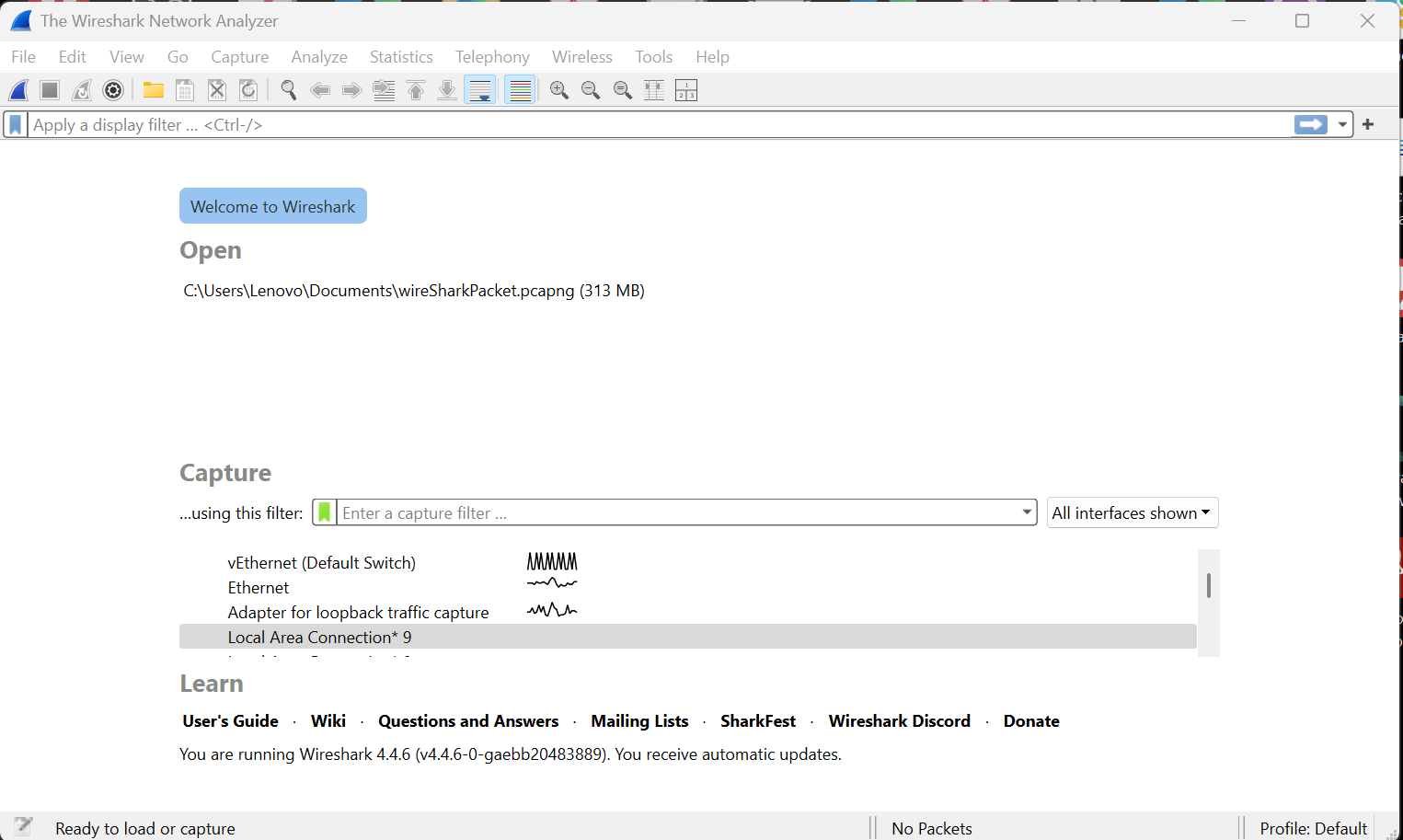
**ID:\_\_\_\_\_\_\_\_\_\_\_\_ Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

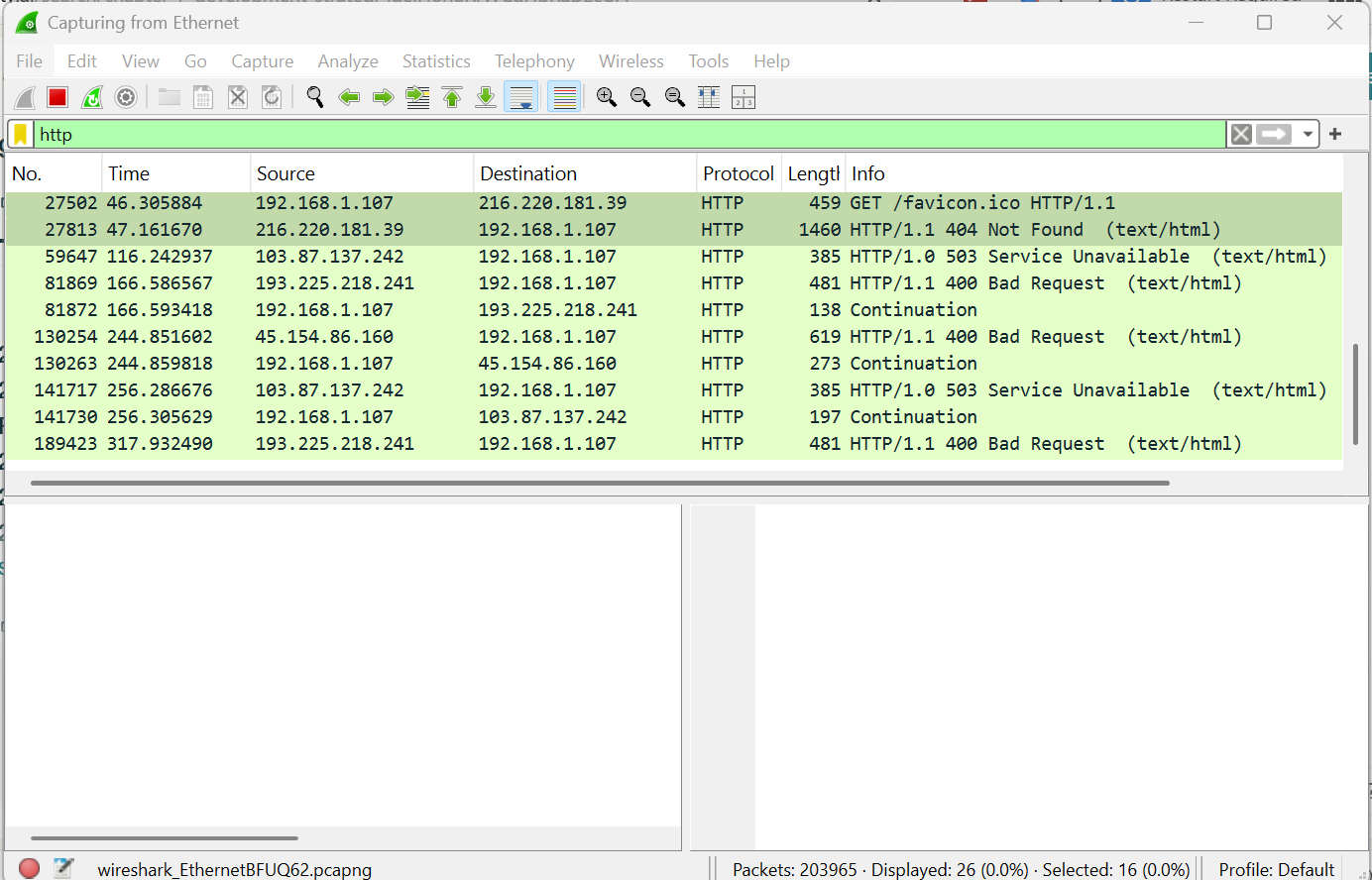
**DCAN202 Week 9 Tutorial – Data Communication and Networking**

**In this lab you will:**

1. Execute Wireshark and practice capturing data packets

Ans:





1. Analyze the results from capturing packets for a file download from a web server.

26586 44.507522 192.168.1.107 216.220.181.39 HTTP 509 GET /cwhite HTTP/1.1

26767 44.771540 216.220.181.39 192.168.1.107 HTTP 442 HTTP/1.1 301 Moved Permanently (text/html)

26770 44.774260 192.168.1.107 216.220.181.39 HTTP 510 GET /cwhite/ HTTP/1.1

26903 45.037923 216.220.181.39 192.168.1.107 HTTP 727 HTTP/1.1 200 OK (text/html)

26911 45.098853 192.168.1.107 216.220.181.39 HTTP 467 GET /cwhite/logo\_CDM.png HTTP/1.1

26967 45.117300 192.168.1.107 216.220.181.39 HTTP 463 GET /cwhite/tdot.gif HTTP/1.1

26970 45.124026 192.168.1.107 216.220.181.39 HTTP 464 GET /cwhite/white.jpg HTTP/1.1

27089 45.364713 216.220.181.39 192.168.1.107 HTTP 1276 HTTP/1.1 200 OK (PNG)

27098 45.369470 192.168.1.107 216.220.181.39 HTTP 466 GET /cwhite/bg\_home.gif HTTP/1.1

27099 45.380036 216.220.181.39 192.168.1.107 HTTP 342 HTTP/1.1 200 OK (GIF89a)

27100 45.381445 192.168.1.107 216.220.181.39 HTTP 475 GET /cwhite/bullet\_blue\_down.gif HTTP/1.1

27110 45.388498 216.220.181.39 192.168.1.107 HTTP 1175 HTTP/1.1 200 OK (JPEG JFIF image)

27166 45.642134 216.220.181.39 192.168.1.107 HTTP 1321 HTTP/1.1 200 OK (GIF89a)

27168 45.646581 216.220.181.39 192.168.1.107 HTTP 1223 HTTP/1.1 200 OK (GIF89a)

27502 46.305884 192.168.1.107 216.220.181.39 HTTP 459 GET /favicon.ico HTTP/1.1

27813 47.161670 216.220.181.39 192.168.1.107 HTTP 1460 HTTP/1.1 404 Not Found (text/html)

**Above packets were captured using Wireshark.**

**Wireshark HTTP Packet Capture Analysis**

**1. Introduction**

This report documents the capture and analysis of HTTP traffic during a web page download using Wireshark. The objective was to observe the interaction between a client (192.168.1.107) and a web server (216.220.181.39) when accessing the URL http://facweb.cs.depaul.edu/cwhite.

**2. Methodology**

1. **Setup**:
   * Determined the client’s MAC address (00-2B-67-3C-E9-E5) and IPv4 address (192.168.1.107).
   * Launched Wireshark and initiated packet capture on the active Ethernet interface.
   * Accessed the target URL using a web browser.
   * Stopped the capture after the page loaded and filtered results using http to isolate HTTP traffic.
2. **Verification**:
   * Confirmed successful capture of HTTP packets, including requests and responses.
   * Cleared browser cache to ensure fresh downloads (avoiding 304 Not Modified responses).

**3. Results**

The capture revealed the following sequence of HTTP transactions:

**Key Events**

1. **Initial Request**:
   * **Packet 26586**: Client sends GET /cwhite HTTP/1.1 to the server.
   * **Purpose**: Requests the root page /cwhite.
2. **Server Redirection**:
   * **Packet 26767**: Server responds with HTTP/1.1 301 Moved Permanently.
   * **Purpose**: Informs the client that the resource has moved to /cwhite/.
3. **Follow-Up Request**:
   * **Packet 26770**: Client sends GET /cwhite/ HTTP/1.1.
   * **Purpose**: Requests the redirected page.
4. **Page Delivery**:
   * **Packet 26903**: Server responds with HTTP/1.1 200 OK, delivering the HTML content.
5. **Resource Loading**:
   * Subsequent GET requests for images (e.g., logo\_CDM.png, tdot.gif, white.jpg) and server responses (200 OK).
   * Example:
     + **Packet 26911**: GET /cwhite/logo\_CDM.png HTTP/1.1
     + **Packet 27089**: Server responds with 200 OK (PNG image).
6. **Favicon Request**:
   * **Packet 27502**: Client requests /favicon.ico.
   * **Packet 27813**: Server responds with HTTP/1.1 404 Not Found (favicon unavailable).

**4. Analysis**

* **HTTP Status Codes**:
  + 301 Moved Permanently: Indicates URL redirection.
  + 200 OK: Confirms successful retrieval of resources.
  + 404 Not Found: Signifies a missing resource (favicon).
* **Cache Behavior**:
  + The absence of 304 Not Modified confirms the page was downloaded freshly from the server, not from the browser cache.
* **Automatic Resource Loading**:
  + Modern browsers automatically request embedded resources (images, stylesheets), resulting in multiple GET-200 OK pairs.

**5. Conclusion**

The Wireshark capture successfully recorded all HTTP transactions during the web page download:

* The client initially requested /cwhite, was redirected to /cwhite/, and received the HTML page.
* Embedded resources (images) were fetched automatically, with the server returning 200 OK for valid files and 404 Not Found for the missing favicon.
* The absence of cached responses (304 Not Modified) validates that the capture reflects a full download from the server.

This exercise demonstrates the use of Wireshark to analyze HTTP traffic, confirm client-server interactions, and verify successful resource retrieval.

**End of Report**

Please go through the videos to learn about Wireshark software:

What is wireshark

<https://www.youtube.com/watch?v=PIwk1gWLN2I&t=22s>

<https://www.youtube.com/watch?v=f4zqMDzXt6k>

How to install wireshark in windows 10

<https://www.youtube.com/watch?v=fpeMCuCKgHA&t=258s>

**Part 1: Check computer addresses and start up Wireshark Capture**

1. Determine the Physical MAC address and the IP address for the Ethernet interface of the computer you are using.

2. Start up Wireshark, click Capture! Interfaces and click the Capture button corresponding to your active Ethernet interface. Wireshark will begin running in Capture Mode and will open up a Wireshark Capture window showing you how many packets have been captured in real time.

**Part 2: Download Web Page**

1. With Wireshark still running in Capture Mode, start up your favorite browser (Internet Explorer, Mozilla, Firefox or whatever)
2. Enter the following address: **http://facweb.cs.depaul.edu/cwhite**
3. A web page should appear in your browser from the author’s home page
4. Close your browser window
5. Go back to the Wireshark Capture window and click the ***Stop*** button to stop the packet capture.

# **Part 3: Verify that the Web Page Download has Been Captured**

1. Back in the Wireshark window, you should now see lots of packets in the top summary pane. You can filter out all packets except HTTP packets by typing the word “http” into the Filter box (click View !Filter Toolbar if you don’t see a Filter box at the top). This will make things much easier to read.
2. You should see a packet containing something like “GET /facweb.cs.depaul.edu/ cwhite” sent by your PC to request the web page download.
3. If the next packet listed (containing the reply from the web server to your PC) contains “HTTP/1.1 200 OK” and the next 3 are “Continuation” packets, then you have successfully captured the packets containing the lab html web page. Skip the following step.
4. On the other hand, if the reply from the web server contains “HTTP/1.1 304 Not Modified”, then this means Wireshark **did not capture** the packets from the web site because the web page was already stored (cached) in your browser. In this case you must clear your browser cache and then go back and re-do the capture as follows:
   1. First, you must clear the web cache in your browser.
      1. For Internet Explorer, click ***Tools*** ! ***Internet Options,*** then, under the General tab, click the ***Delete Files*** button within the ***Temporary Internet Files*** box area***.***
      2. For Firefox, click ***Tools*** ! ***Options***, then ***Privacy, Cache*** and click ***Clear***. iii. For other browsers, you’re on your own.
   2. Now in your Wireshark window, again select ***Capture***!***Interfaces*** and click the ***Capture*** button corresponding to the Ethernet interface. When prompted whether to save the previous capture, click ***Continue without saving.***
   3. Go back to Part 2, step 1 above to download the web page again while Wireshark is capturing packets.