Lab 1: Packet analysis at application layer using Wireshark

SCSR1213 Network Communications

Universiti Teknologi Malaysia

Name : MADINA SURAYA BINTI ZHARIN

Matric No : A20EC0203

Section : 02

Objective:

1. Understanding of network protocols by 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.

2. To introduce student with Wireshark software tool for packet analyzer.

3. To analyze protocol used in application layer such as http and dns.

Reference material: Computer Networking: A Top-Down Approach, 7th ed., J.F. Kurose and K.W. Ross.

Mark

PART A: Wireshark Getting Started

1.0 Introduction

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 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 A.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 A.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. In Figure A.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.

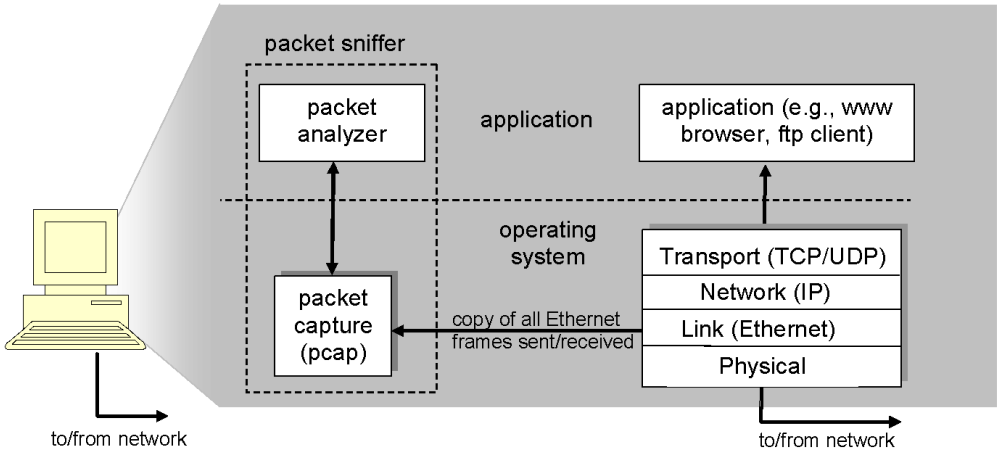


Figure A.1: Packet sniffer structure

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. 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".

2.0 Getting Wireshark Ready

- Download and install the Wireshark software
- Run Wireshark. Wireshark startup screen shown in Figure A.2.

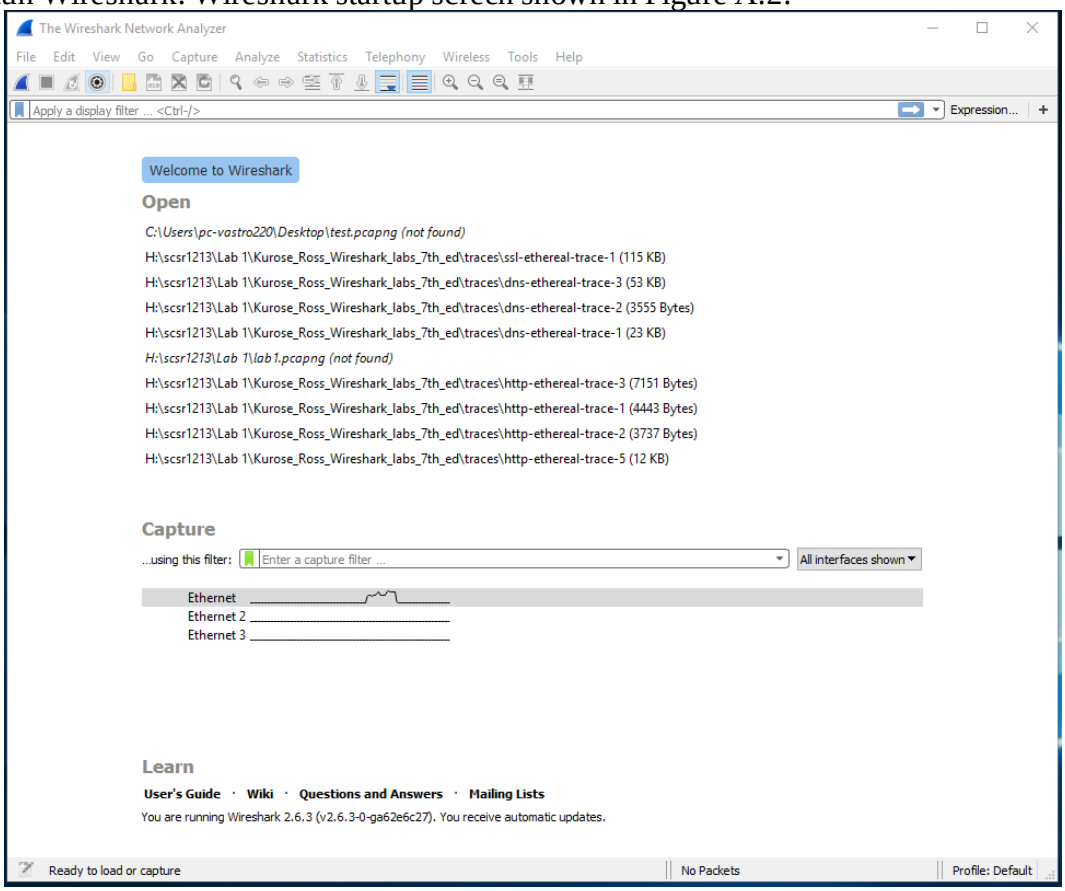


Figure A.2: Initial Wireshark startup screen

- The Wireshark interface has five major components as shown in Figure A.3.

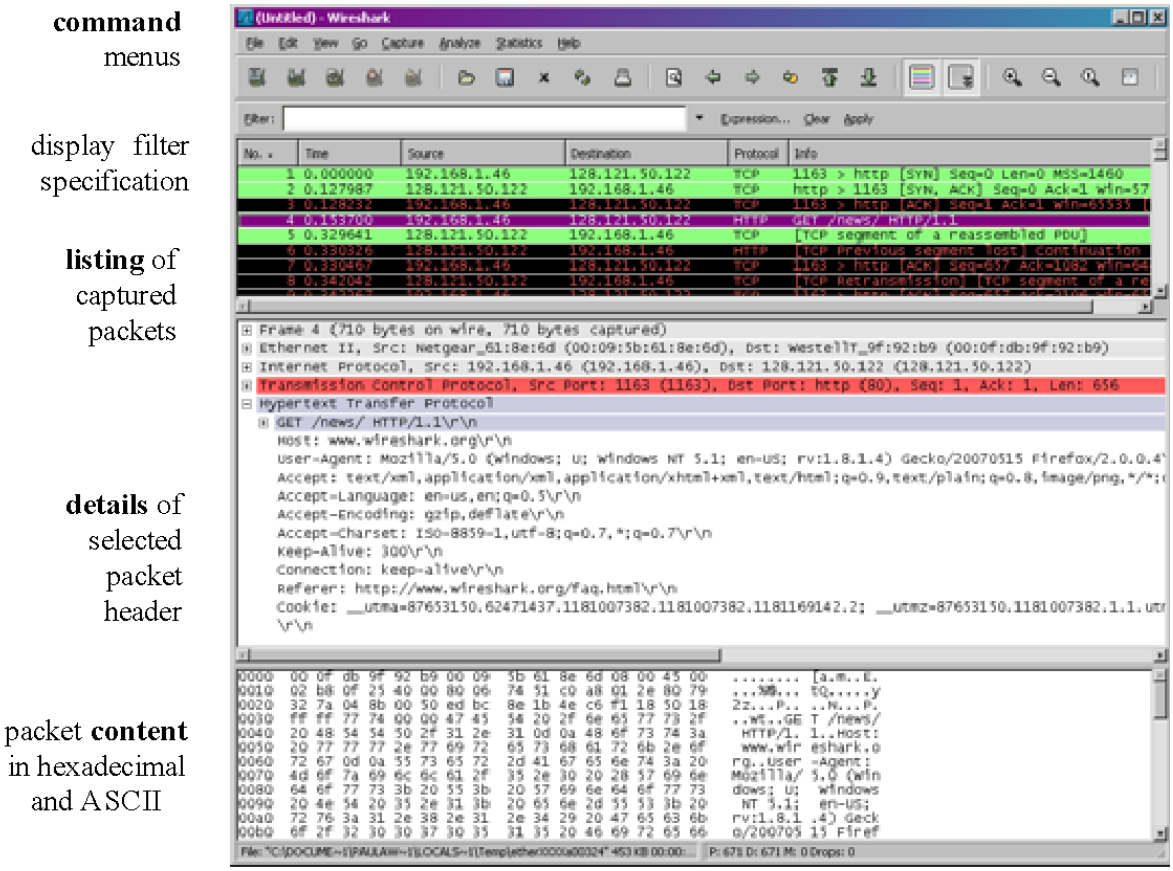


Figure A.3: Wireshark Graphical User Interface, during packet capture and

- The **command menus** are standard pulldown menus located at the top of the window.

- 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.

- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number, 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-header details window** provides details about the packet selected (highlighted) in the packet-listing window. These details include information about the Ethernet frame 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 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.

3.0 Test Run Wireshark

- Start up the Wireshark software.
- To begin packet capture, select the Capture pull down menu and pick Options menu. Select appropriate interfaces on your compute and click Start button to begin packet capture. Refer to Figure A.4

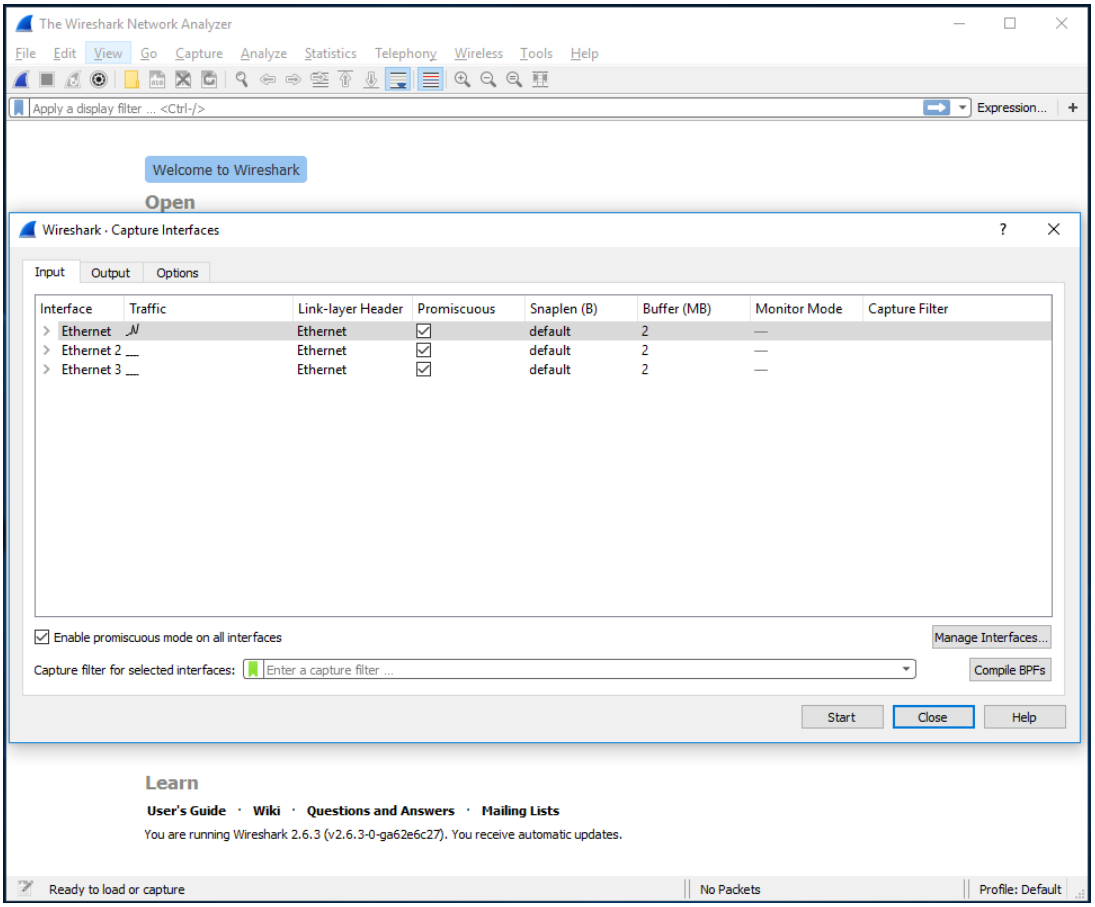


Figure A.4: Capture and Options Menu

- Once you begin packet capture, the result will be shown as in Figure A.5.

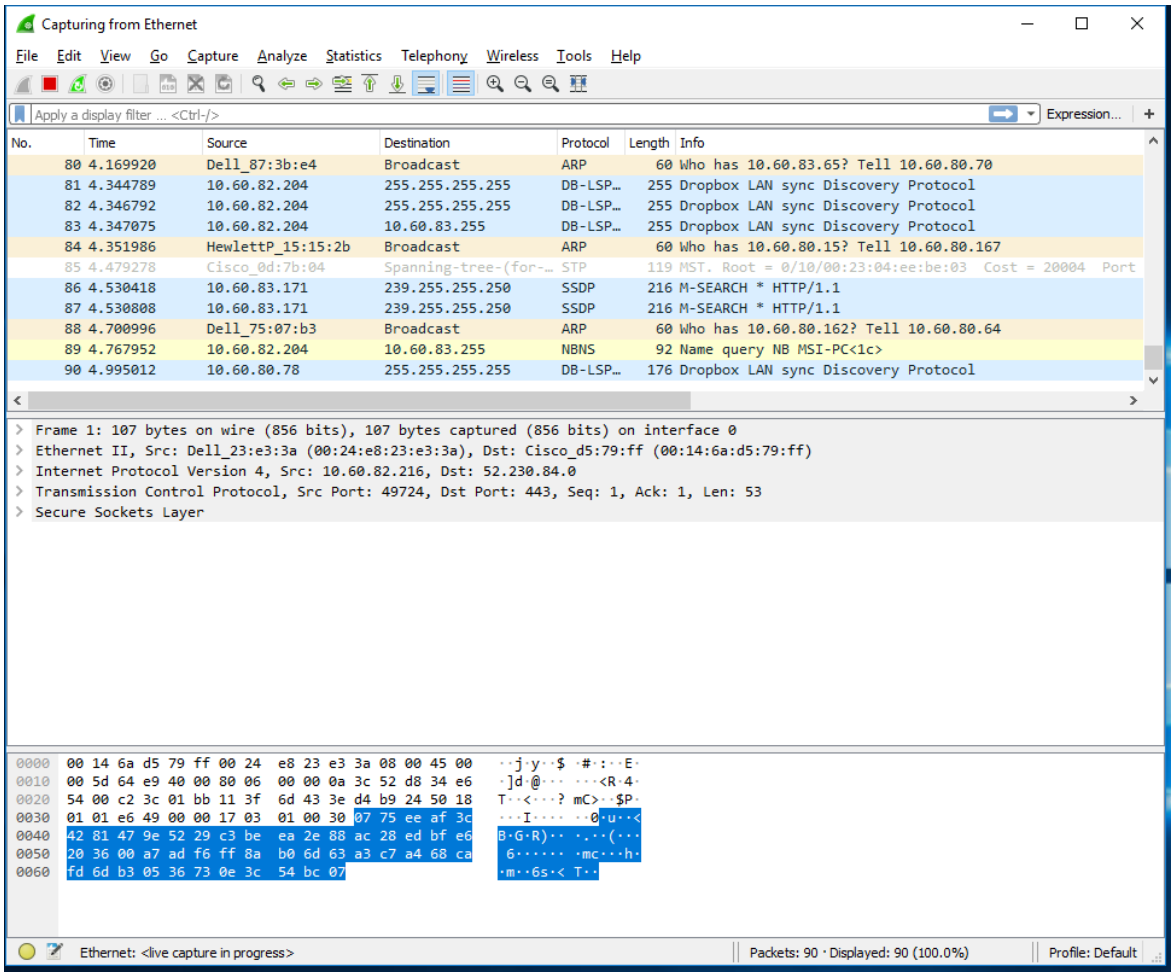


Figure A.5: Wireshark packet capture result

- By selecting Capture pulldown menu and selecting Stop, you can stop packet capture.

- Type “arp” in packet display filter field and press Enter key. This will cause only ARP message to be displayed in the packet-listing window as shown in Figure A.6.

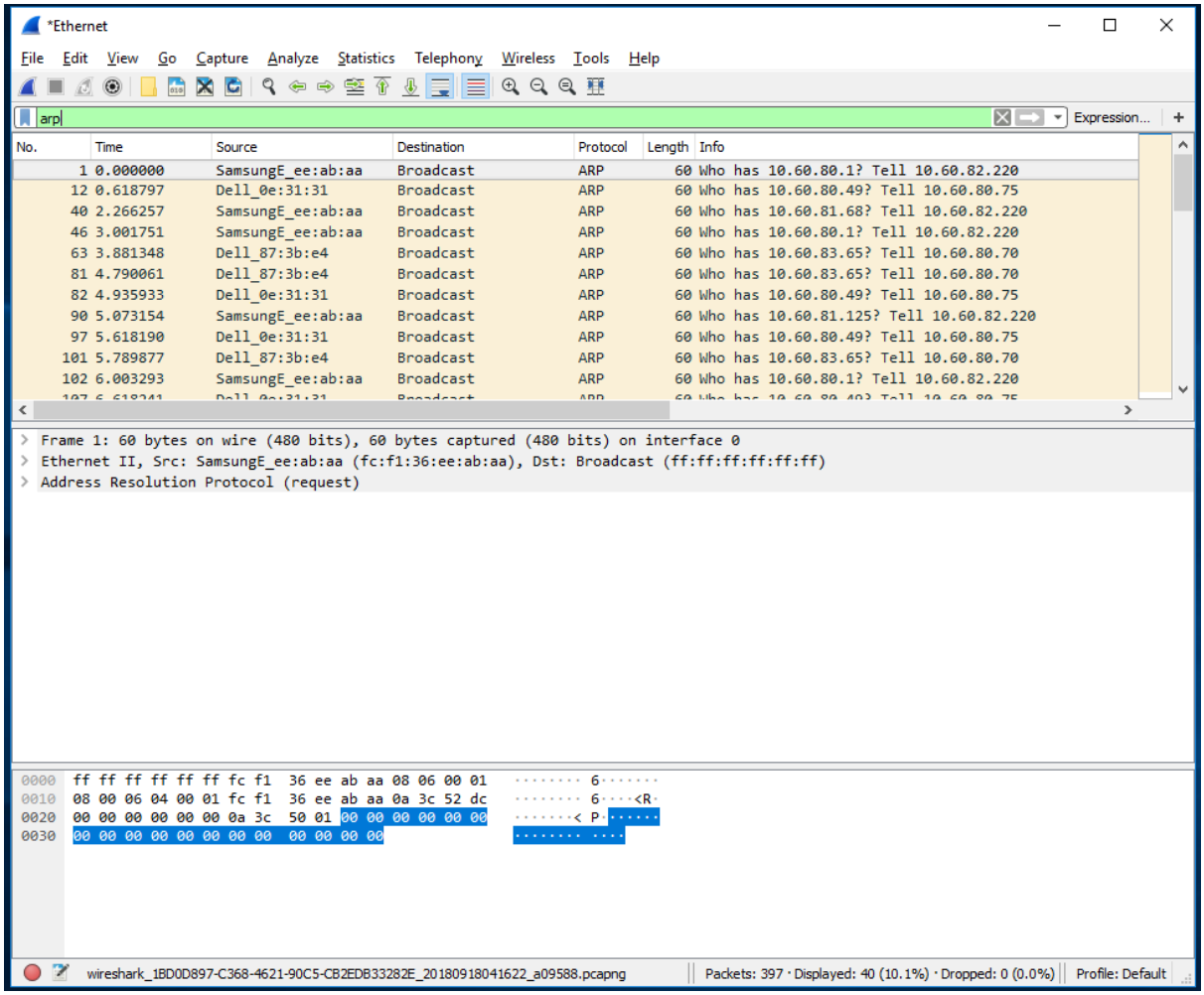


Figure A.6: ARP packet capture

- To save the trace result, use File pulldown menu and select Save function as shown in Figure A.7.

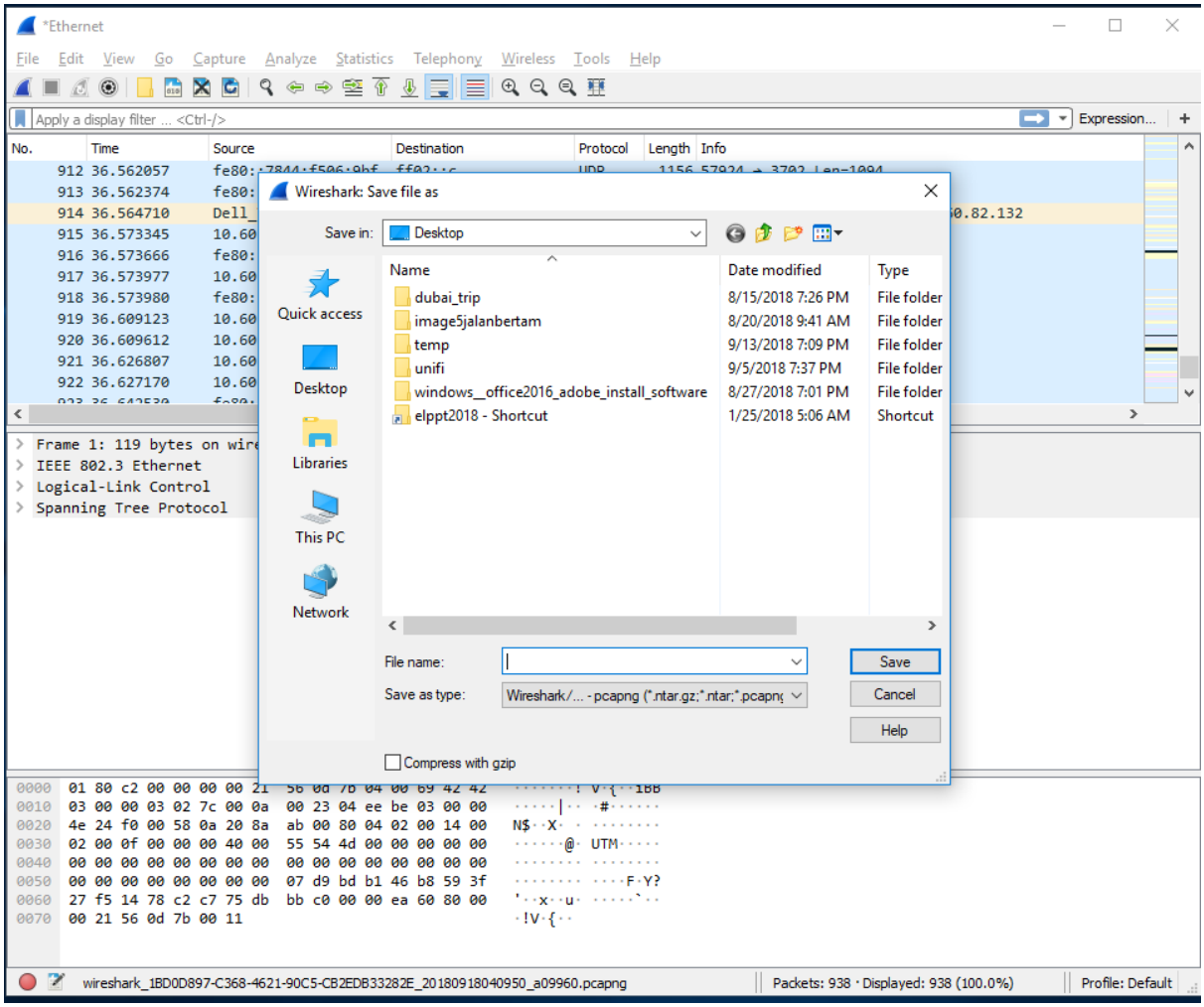


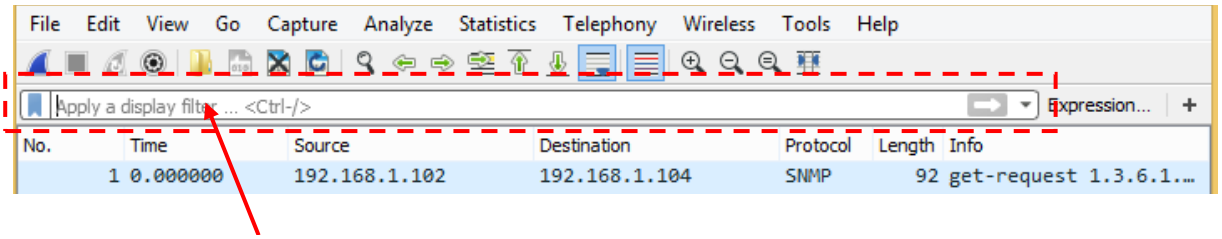
Figure A.7: Save Wireshark trace result

PART B: HTTP Trace

In this part, we'll explore several aspects of the HTTP protocol: the basic GET/response interaction, HTTP message formats and retrieving HTML files with embedded objects. Before beginning these labs, you might want to review Section 2.2 of the textbook.

B.1 The Basic HTTP GET/response interaction

- Open packet trace file **lab1-http-B01.pcapng**.
- Enter "**http**" (just the letters, not the quotation marks) in the **packet display filter field**, so that only captured HTTP messages will be displayed later in the packet-listing window. Refer to figure below:



packet display filter

- By looking at the information in the HTTP GET and response messages, answer the following questions:

1. What version of HTTP is the server running?
   VERSION 4

2. What is the IP address of the client computer?
   192.168.1.102

3. What is the IP address of the gaia.cs.umass.edu server?
   128.119.245.12

4. How many bytes of content are being returned to the client browser?

   73 bytes

5. What is the status code returned from the server to the client browser?

   200 OK

B.2 The HTTP CONDITIONAL GET/response interaction

- Open packet trace file **lab1-http-B02.pcapng**.
- By looking at the information in the HTTP GET and response messages, answer the following questions:

1. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

   No

2. Inspect the contents of the server response after the first GET request from the client. Did the server explicitly return the contents of the file? How can you tell?

   Yes. It displays the content of the http.

```
Line-based text data: text/html (10 lines)
    \n
    <html>\n
    \n
    Congratulations again!  Now you've downloaded the file lab2-2.html. <br>\n
    This file's last modification date will not change.  <p>\n
    Thus  if you download this multiple times on your browser, a complete copy <br>\n
    will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
    field in your browser's HTTP GET request to the server.\n
    \n
    </html>\n
```

3. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header?

   The day, date and time.

```
If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n
```

4. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

   304 Not Modified.

   The server did not explicitly return the contents of the file as the requested resource has not been modified since the previous transmission, so there is no need to retransmit the requested resource to the client. It is also the redirection to a cached version of the requested resource.

B.3 HTML Documents with Embedded Objects

- Open packet trace file **lab1-http-B03.pcapng**.
- By looking at the information in the HTTP GET and response messages, answer the following questions:

1. How many HTTP GET request messages did the client browser send?
   3 times.

2. To which Internet addresses were these GET requests sent?
   - 128.119.245.12
   - 165.193.123.218
   - 134.241.6.82

3. How many bytes of content are being returned to the client browser for the **pearson-logo-footer.gif** image file?
   3357 bytes

4. How many bytes of content are being returned to the client browser for the **cover.jpg** image file?

   15642 bytes

PART C: DNS Trace

1.0 nslookup

nslookup tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server. To accomplish this task, nslookup sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.

- To run it in Windows, open the Command Prompt (cmd) and run nslookup on the command line as shown in Figure C.1 and Figure C.2

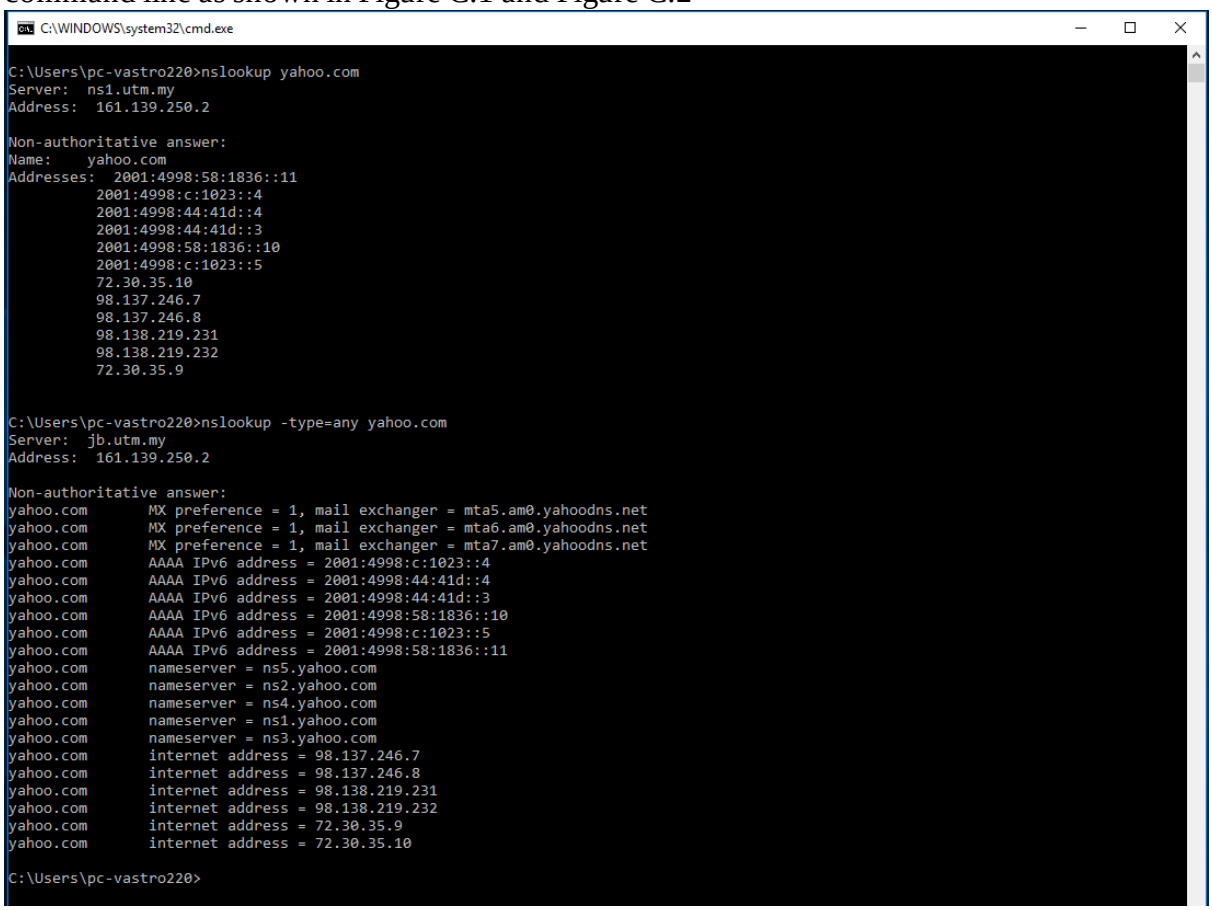


Figure C.1: nslookup result

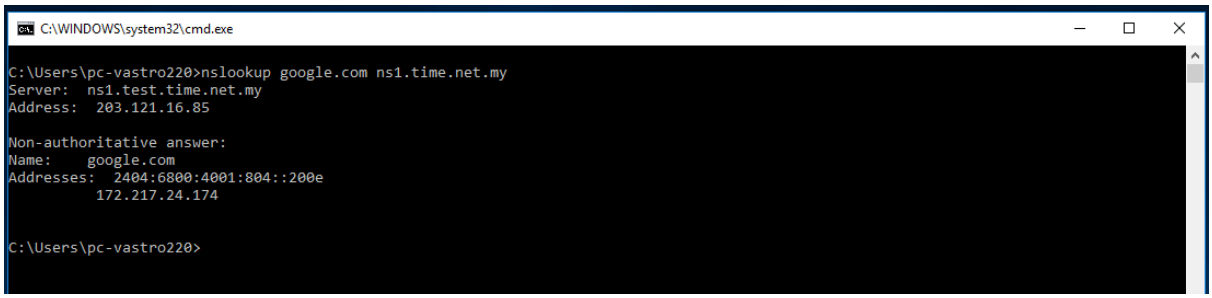


Figure C.2: nslookup result

1. Run nslookup to obtain the IP address of a www.microsoft.com server. What is the IP address of that server? Add screenshot to your answer.

2. Run nslookup to determine the non-authoritative DNS servers for domain microsoft.com. Add screenshot to your answer.

2.0 ipconfig

ipconfig can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on.

- Information about host, use the following command: ipconfig /all

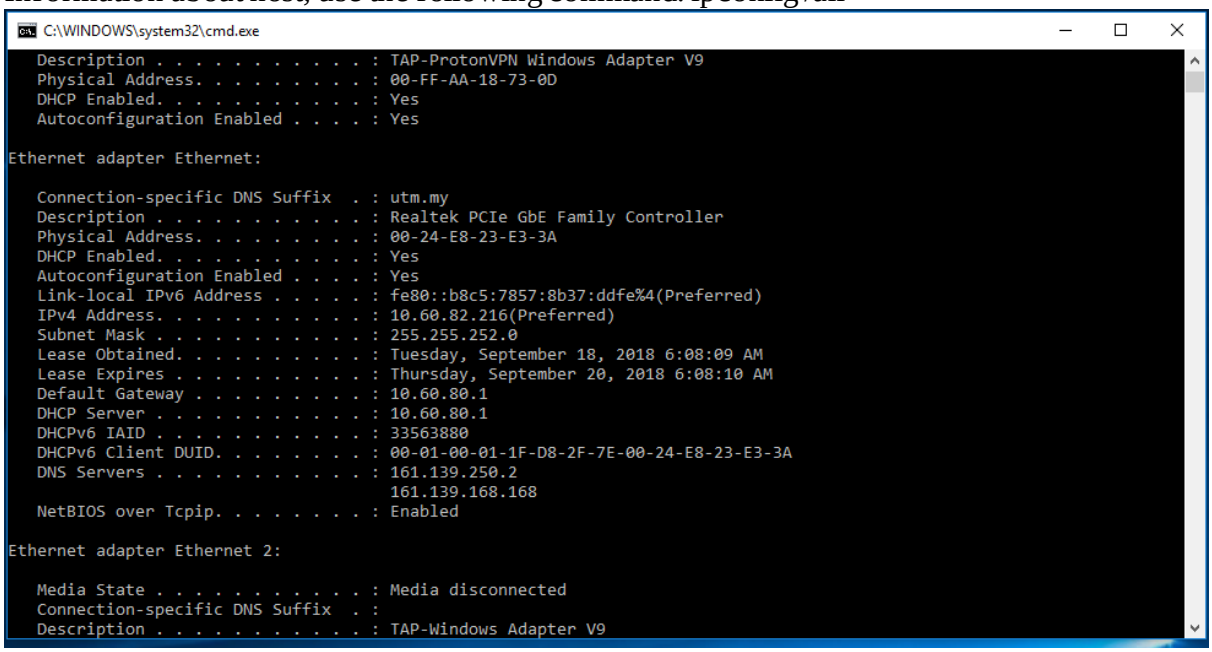


Figure C.3: ipconfig /all result

- ipconfig is also very useful for managing the DNS information stored in your host. Each entry shows the remaining Time to Live (TTL) in seconds.
  Command: ipconfig /displaydns

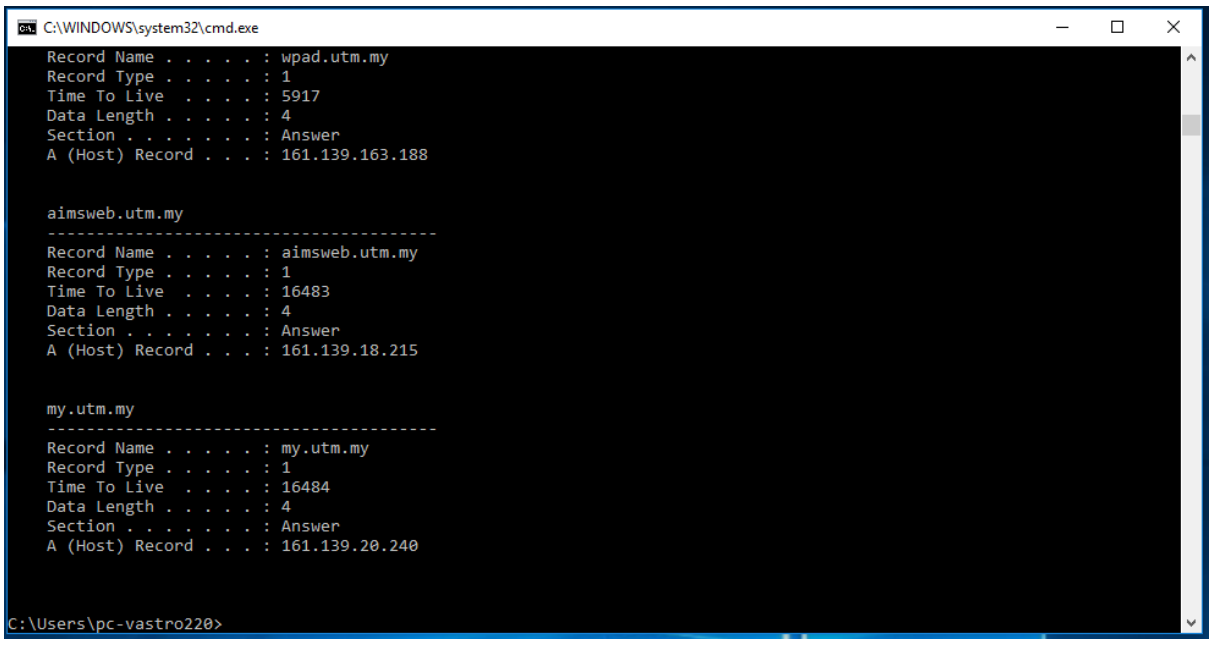


Figure C.4: ipconfig /displaydns result

- Flushing the DNS cache clears all entries and reloads the entries from the hosts file.

  Command: ipconfig /flushdns

Figure C.5: ipconfig /flushdns result

3.0 Tracing DNS with Wireshark

- Open packet trace file dns-trace-1. Answer the following questions.

1. Locate the DNS query and response messages. Are then sent over UDP or TCP? Add screenshots in your answer.

2. What is the destination port for the DNS query message? What is the source port of DNS response message? Add screenshots in your answer.

3. To what IP address is the DNS query message sent? Add screenshots in your answer.

4. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"? Add screenshots in your answer.

5. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain? Add screenshots in your answer.

6. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message? Add screenshots in your answer.

7. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

- Open packet trace file dns-trace-2 for nslookup.
- We see from Wireshark that nslookup actually sent three DNS queries and received three DNS responses. For the purpose of this lab, ignore the first two sets of queries/responses, as they are specific to nslookup and are not normally generated by standard Internet applications. You should instead focus on the last query and response messages.
- Answer the following questions.

8. What is the destination port for the DNS query message? What is the source port of DNS response message? Add screenshots in your answer.

9. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? Add screenshots in your answer.

10. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"? Add screenshots in your answer.

11. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain? Add screenshots in your answer.