CSCI3761 Lab #1 Spring 2019

Due: February, 27 (Wednesday), 2019

Max points: 30 points

Objective

1. Learn how to use packet sniffing tool, Wireshark
2. Learn the main protocol of DNS (Domain Name Services)

Note: This lab is adapted from DNS WireShark Lab, "Computer Networking: A Top-Down Approach", *7th ed,* J.F. Kurose and K.W. Ross.

Required software tool

1. Packet Sniffer

The purpose of lab assignments is to show how we can get a deeper understanding of the networking concepts by capturing and analyzing the packets sent and received from our host. One way to do so is to use a **packet sniffer**. A packet sniffer is a piece of soft- ware that should be running in parallel with the application whose packets needed to be analyzed. However, before running a packet sniffer, we need to interpret the term *packet*. As we discussed in Chapter 1 of the textbook, communication via the Internet is done using a five-layer suite. We can analyze the packets at four layers: application, transport, network, and data-link. There is no packet exchange at the physical layer; communication at this layer is done using bits.

In an outgoing situation, a packet created at any upper-layer is encapsulated in a *frame* (at the data-link layer); in an incoming situation, a packet intended for any layer is decapsulated from the received frame. This means we need to capture only outgoing or incoming frames; a packet-sniffer software can extract the packets at any layer desired to be analyzed from these frames. For this reason, a packet-sniffer software is normally has two components: a **packet-capturer** and a **packet-analyzer**. The packet-capturer captures a copy of all outgoing and incoming frames (at the data-link layer) and passes them to the packet-analyzer. The packet- analyzer can then extract different headers and the ultimate message for analysis.

2. Wireshark

In this and other lab assignments, we use a packet-sniffer called **Wireshark**. Wireshark (formerly known as ETHEREAL) is a free packet sniffer/analyzer which is available for both UNIX-like (Unix, Linux, Mac OS X, BSD, and Solaris) and Windows operating systems. It captures packets from a network interface and displays them with detailed protocol information. Wireshark, however, is a passive analyzer. It only captures packets without manipulate them; it neither sends packets to the network nor does other active operations. Wireshark is not an intrusion-detection tool either. It does not give warning about any network intrusion. It, nevertheless, can help network administrators to figure out what is going on inside a network and to troubleshoot network problems. In addition of being an indispensable tool for network administrators, Wire- shark is a valuable tool for protocol developers, who may use it to debug protocol implementations. It is also a great educational tool for computer-network students who can use it to see details of protocol operations in real time.

2.1 Downloading and Installing

To download the Wireshark software, connect to the Internet using the website:

http://www.wireshark.org/download.html

After the downloading is complete, install the software on your computer. If you have any problem in downloading or installing, you can consult the following site for more information:

http://wiki.wireshark.org/CaptureSetup

2.2 Main Window

The Wireshark main window is similar to other GUI tools as shown in Figure 2.1. The Wireshark window is made of seven sections: **title bar**, **menu bar**, **filter bar**, **packet list pane**, **packet detail pane**, **packet byte pane**, and **status bar**. We briefly discuss the functionality of each section below:

Title Bar

The *title bar* (like the one in any GUI) shows the title of the window, the closing, maximizing, and minimizing icons.

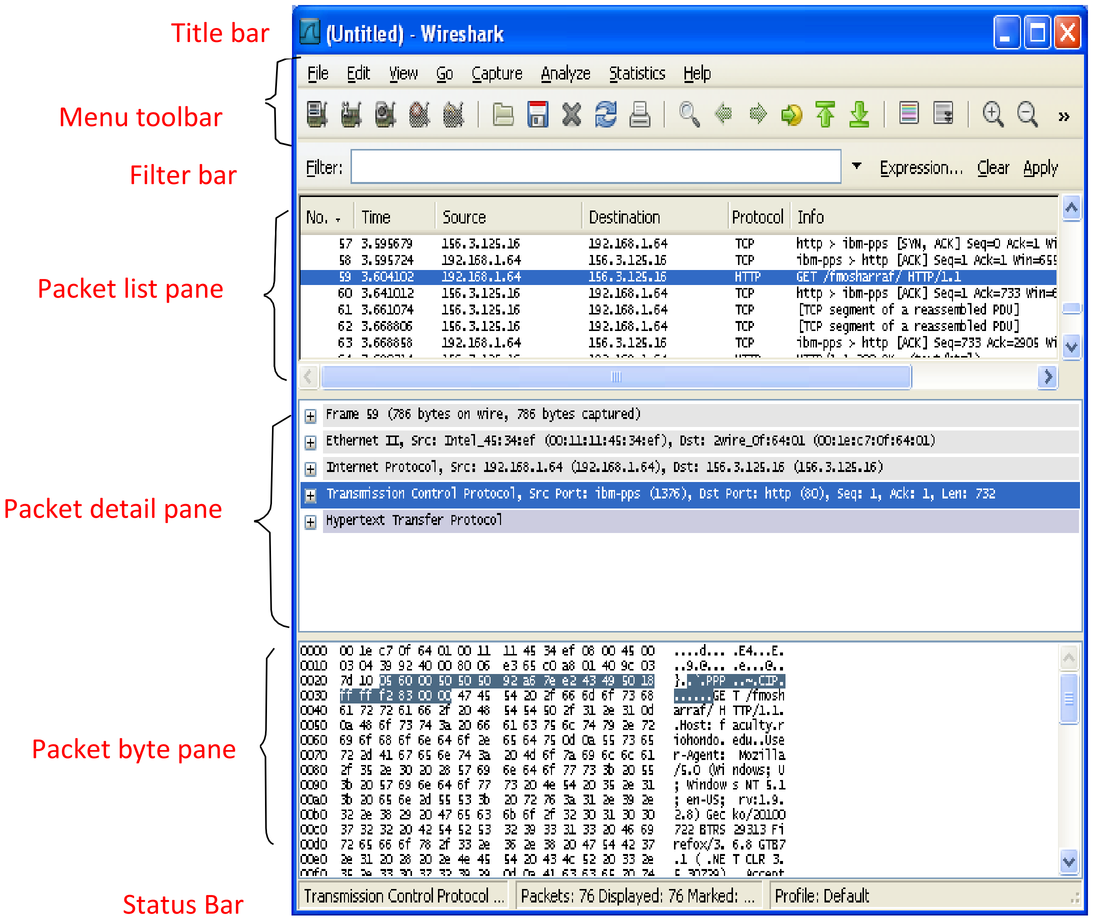
Menu Bar

The *menu bar* is made of several pulldown menus and tool bars used in most GUIs. We will using some of these menus in our lab assignments. We can use the **File** menu to perform some actions on the file itself such as saving and printing. The **Capture** menu is used to start and capturing frames. The **View** menu is useful to show or hide some of the sections in the window.

Filter Bar

The *filter bar* allows us to display packet we are interested in while hiding the rest. As we see later in this document, when we start capturing frames, Wireshark captures and analyze any outgoing and incoming frame no matter what is the source or sink proto- col. Sometimes, this is not what we want. We may want to limit the analysis to a specific source or sink protocol. For example, we may want to analyze only packets sent or receive by the HTTP protocol at the application layer or the ARP protocol at the net- work layer. This is called filtering in the parlance of packet sniffing. After packets have been captured, we can type the name of the protocol in lowercase and click **Apply**.

Figure 2.1 *Main window of Wireshark*



```
No. -  Time        Source          Destination     Protocol  Info
    57 3.595679    156.3.125.16    192.168.1.64    TCP       http > ibm-pps [SYN, ACK] Seq=0 Ack=1 Wi
    58 3.595724    192.168.1.64    156.3.125.16    TCP       ibm-pps > http [ACK] Seq=1 Ack=1 Win=65!
    59 3.604102    192.168.1.64    156.3.125.16    HTTP      GET /fmosharraf/ HTTP/1.1
    60 3.641012    156.3.125.16    192.168.1.64    TCP       http > ibm-pps [ACK] Seq=1 Ack=733 Win=6
    61 3.661074    156.3.125.16    192.168.1.64    TCP       [TCP segment of a reassembled PDU]
    62 3.668806    156.3.125.16    192.168.1.64    TCP       [TCP segment of a reassembled PDU]
    63 3.668858    192.168.1.64    156.3.125.16    TCP       ibm-pps > http [ACK] Seq=733 Ack=2905 Wi
```

```
+ Frame 59 (786 bytes on wire, 786 bytes captured)
+ Ethernet II, Src: Intel_45:34:ef (00:11:11:45:34:ef), Dst: 2wire_0f:64:01 (00:1e:c7:0f:64:01)
+ Internet Protocol, Src: 192.168.1.64 (192.168.1.64), Dst: 156.3.125.16 (156.3.125.16)
+ Transmission Control Protocol, Src Port: ibm-pps (1376), Dst Port: http (80), Seq: 1, Ack: 1, Len: 732
+ Hypertext Transfer Protocol
```

```
0000  00 1e c7 0f 64 01 00 11  11 45 34 ef 08 00 45 00   ....d... .E4...E.
0010  03 04 39 92 40 00 80 06  e3 65 c0 a8 01 40 9c 03   ..9.@... .e...@..
0020  7d 10 05 60 00 50 50 50  92 a6 7e e2 43 49 50 18   }..`.PPP ..~.CIP.
0030  ff ff f2 83 00 00 47 45  54 20 2f 66 6d 6f 73 68   ......GE T /fmosh
0040  61 72 72 61 66 2f 20 48  54 54 50 2f 31 2e 31 0d   arraf/ H TTP/1.1.
0050  0a 48 6f 73 74 3a 20 66  61 63 75 6c 74 79 2e 72   .Host: f aculty.r
0060  69 6f 68 6f 6e 64 6f 2e  65 64 75 0d 0a 55 73 65   iohondo. edu..Use
0070  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
0080  2f 35 2e 30 20 28 57 69  6e 64 6f 77 73 3b 20 55   /5.0 (Wi ndows; U
0090  3b 20 57 69 6e 64 6f 77  73 20 4e 54 20 35 2e 31   ; Window s NT 5.1
00a0  3b 20 65 6e 2d 55 53 3b  20 72 76 3a 31 2e 39 2e   ; en-US;  rv:1.9.
00b0  32 2e 38 29 20 47 65 63  6b 6f 2f 32 30 31 30 30   2.8) Gec ko/20100
00c0  37 32 32 20 42 54 52 53  32 39 33 31 33 20 46 69   722 BTRS 29313 Fi
00d0  72 65 66 6f 78 2f 33 2e  36 2e 38 20 47 54 42 37   refox/3. 6.8 GTB7
00e0  2e 31 20 28 20 2e 4e 45  54 20 43 4c 52 20 33 2e   .1 ( .NE T CLR 3.
```

```
Transmission Control Protocol ... | Packets: 76 Displayed: 76 Marked: ... | Profile: Default
```

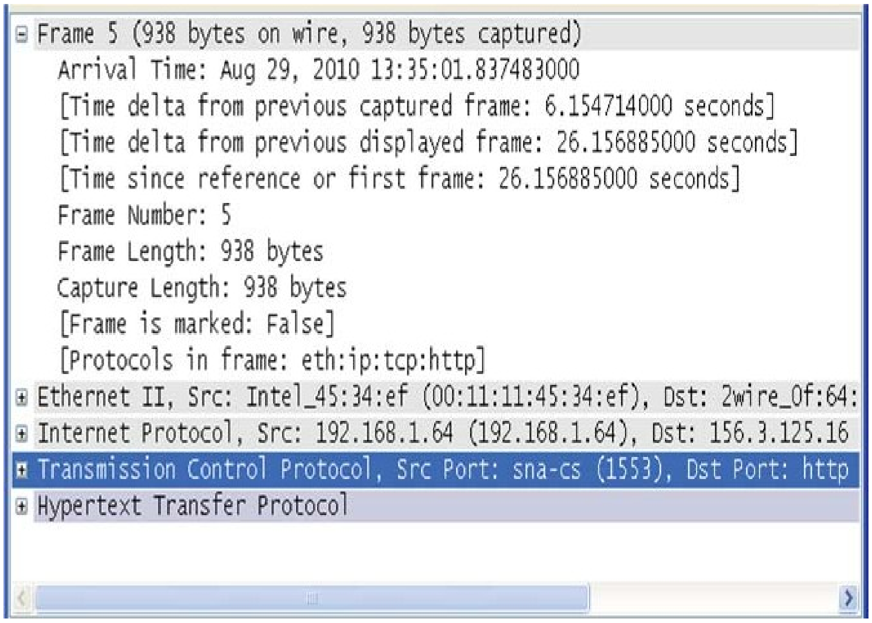
Packet List Pane

The *packet list pane* displays a one-line summary for each captured packet (actually frame). The summary includes the packet (frame) number (added by the Wireshark and not part of the packet), the time when the packet was captured, the source and destination IP addresses of the packet (at the network layer), the packet source or sink proto- col, and the additional information about the packet contents. In other words, this pane shows the captured frames that will be passed for analyzing to the packet analyzer.

Packet Detail Pane

The *packet detail pane* shows the detailed analysis for each frame (Figure 2.2). The information is limited to one frame, which means we need to select one of the frames in the packet list pane for analysis. This can be done by clicking on the corresponding frame in the packet list pane. Clicking on any frame in the *packet list pane* highlights the frame and shows the details of the frame in the packet details pane. Information exhibited in this pane for each frame is made of a tree structure. However, each top branch of the tree is shown as one line as it is common in GUI trees. We can expand the branch (to see subbranches) by clicking on the plus box at the leftmost part of the line, which changes the plus sign to a minus sign; the branch can be collapsed again, which changes the minus sign to the plus sign. Note that the analyzer first shows a general information at the data-link layer (frame). It then displays the information contained in each header from the data-link layer (H2) up to the source or sink protocol. It finally shows the whole message at the source or sink layer. Figure 2.2 shows an example of a packet details pane when the frame is expanded. It shows some general information and names of all protocols used in the frame (intermediate and source or sink).

Figure 2.2 *Packet detail pane*

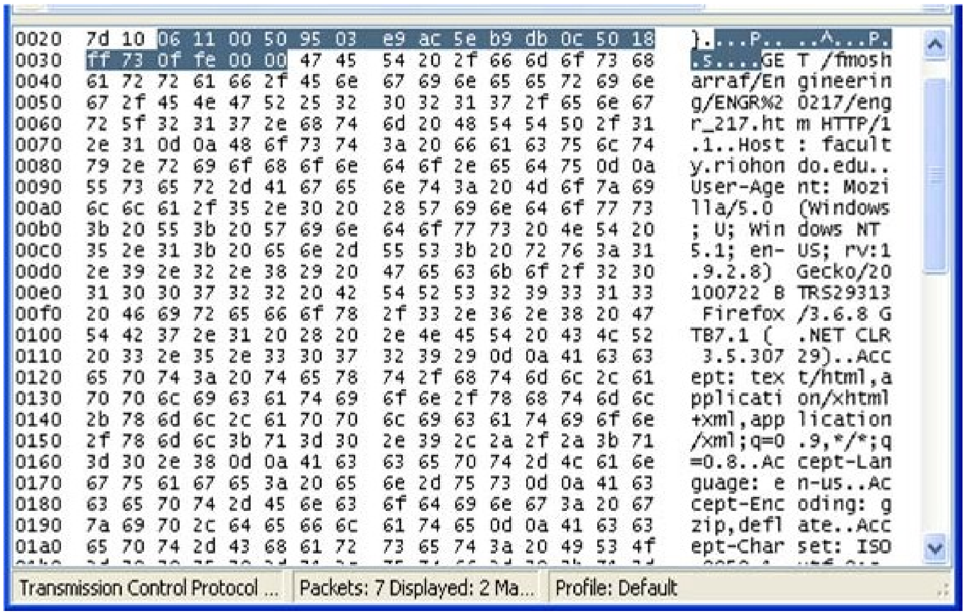


```
⊟ Frame 5 (938 bytes on wire, 938 bytes captured)
    Arrival Time: Aug 29, 2010 13:35:01.837483000
    [Time delta from previous captured frame: 6.154714000 seconds]
    [Time delta from previous displayed frame: 26.156885000 seconds]
    [Time since reference or first frame: 26.156885000 seconds]
    Frame Number: 5
    Frame Length: 938 bytes
    Capture Length: 938 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:tcp:http]
⊞ Ethernet II, Src: Intel_45:34:ef (00:11:11:45:34:ef), Dst: 2wire_0f:64:
⊞ Internet Protocol, Src: 192.168.1.64 (192.168.1.64), Dst: 156.3.125.16
⊞ Transmission Control Protocol, Src Port: sna-cs (1553), Dst Port: http
⊞ Hypertext Transfer Protocol
```

Packet byte pane

The *packet byte pane* shows the entire current frame (selected in the packet list pane) in hexdump format (hexadecimal view of data) and ASCII format. The number in the left field shows the offset in the packet data; the hexdump of the packet is shown in the middle field; the corresponding ASCII characters are shown in the right field. If we need the byte (or ASCII equivalent) of any line in the packet detail pane, we can click on the line in the packet detail pane and the byte contents will be highlighted. Figure 2.3 shows an example of a packet byte pane. It shows all the bytes in the frame, but we can select the bytes in any protocol header by highlighting it in the *packet detail pane* section.

Figure 2.3 *Packet byte pane*



```
0020  7d 10 06 11 00 50 95 03  e9 ac 5e b9 db 0c 50 18   }....P.. ..^...P.
0030  ff 73 0f fe 00 00 47 45  54 20 2f 66 6d 6f 73 68   .s....GE T /fmosh
0040  61 72 72 61 66 2f 45 6e  67 69 6e 65 65 72 69 6e   arraf/En gineerin
0050  67 2f 45 4e 47 52 25 32  30 32 31 37 2f 65 6e 67   g/ENGR%2 0217/eng
0060  72 5f 32 31 37 2e 68 74  6d 20 48 54 54 50 2f 31   r_217.ht m HTTP/1
0070  2e 31 0d 0a 48 6f 73 74  3a 20 66 61 63 75 6c 74   .1..Host : facult
0080  79 2e 72 69 6f 68 6f 6e  64 6f 2e 65 64 75 0d 0a   y.riohon do.edu..
0090  55 73 65 72 2d 41 67 65  6e 74 3a 20 4d 6f 7a 69   User-Age nt: Mozi
00a0  6c 6c 61 2f 35 2e 30 20  28 57 69 6e 64 6f 77 73   lla/5.0  (Windows
00b0  3b 20 55 3b 20 57 69 6e  64 6f 77 73 20 4e 54 20   ; U; Win dows NT
00c0  35 2e 31 3b 20 65 6e 2d  55 53 3b 20 72 76 3a 31   5.1; en- US; rv:1
00d0  2e 39 2e 32 2e 38 29 20  47 65 63 6b 6f 2f 32 30   .9.2.8)  Gecko/20
00e0  31 30 30 37 32 32 20 42  54 52 53 32 39 33 31 33   100722 B TRS29313
00f0  20 46 69 72 65 66 6f 78  2f 33 2e 36 2e 38 20 47    Firefox /3.6.8 G
0100  54 42 37 2e 31 20 28 20  2e 4e 45 54 20 43 4c 52   TB7.1 (  .NET CLR
0110  20 33 2e 35 2e 33 30 37  32 39 29 0d 0a 41 63 63    3.5.307 29)..Acc
0120  65 70 74 3a 20 74 65 78  74 2f 68 74 6d 6c 2c 61   ept: tex t/html,a
0130  70 70 6c 69 63 61 74 69  6f 6e 2f 78 68 74 6d 6c   pplicati on/xhtml
0140  2b 78 6d 6c 2c 61 70 70  6c 69 63 61 74 69 6f 6e   +xml,app lication
0150  2f 78 6d 6c 3b 71 3d 30  2e 39 2c 2a 2f 2a 3b 71   /xml;q=0 .9,*/*;q
0160  3d 30 2e 38 0d 0a 41 63  63 65 70 74 2d 4c 61 6e   =0.8..Ac cept-Lan
0170  67 75 61 67 65 3a 20 65  6e 2d 75 73 0d 0a 41 63   guage: e n-us..Ac
0180  63 65 70 74 2d 45 6e 63  6f 64 69 6e 67 3a 20 67   cept-Enc oding: g
0190  7a 69 70 2c 64 65 66 6c  61 74 65 0d 0a 41 63 63   zip,defl ate..Acc
01a0  65 70 74 2d 43 68 61 72  73 65 74 3a 20 49 53 4f   ept-Char set: ISO
```

Transmission Control Protocol ... | Packets: 7 Displayed: 2 Ma... | Profile: Default

Status Bar

The last section of the window (at the bottom) is the *status bar* which shows the current protocol, the total number of packets captured, and so on.

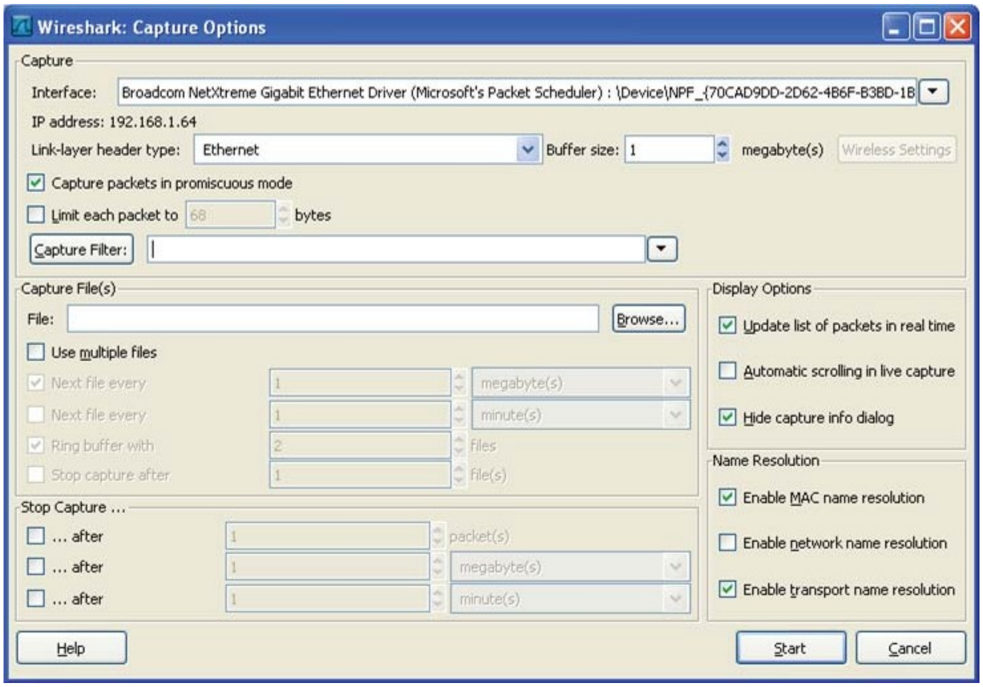
2.3 Working With Wireshark

When we work with Wireshark in this and other labs, there are some actions that we need to repeat over and over. We mention the details of some of this action to avoid re-mentioning them.

Start Capturing

To begin capturing, select the **Capture** from the pull down menu and click **Options** to open the Wireshark **capture options dialog box** (Figure 2.5).

Figure 2.5 *Caption options dialog box*



Wireshark: Capture Options

Capture

Interface: Broadcom NetXtreme Gigabit Ethernet Driver (Microsoft's Packet Scheduler) : \Device\NPF_{70CAD9DD-2D62-4B6F-B3BD-1B

IP address: 192.168.1.64

Link-layer header type: Ethernet Buffer size: 1 megabyte(s) Wireless Settings

✓ Capture packets in promiscuous mode

☐ Limit each packet to 68 bytes

Capture Filter:

Capture File(s)

File: Browse...

☐ Use multiple files

✓ Next file every 1 megabyte(s)

☐ Next file every 1 minute(s)

✓ Ring buffer with 2 files

☐ Stop capture after 1 file(s)

Stop Capture ...

☐ ... after 1 packet(s)

☐ ... after 1 megabyte(s)

☐ ... after 1 minute(s)

Display Options

✓ Update list of packets in real time

☐ Automatic scrolling in live capture

✓ Hide capture info dialog

Name Resolution

✓ Enable MAC name resolution

☐ Enable network name resolution

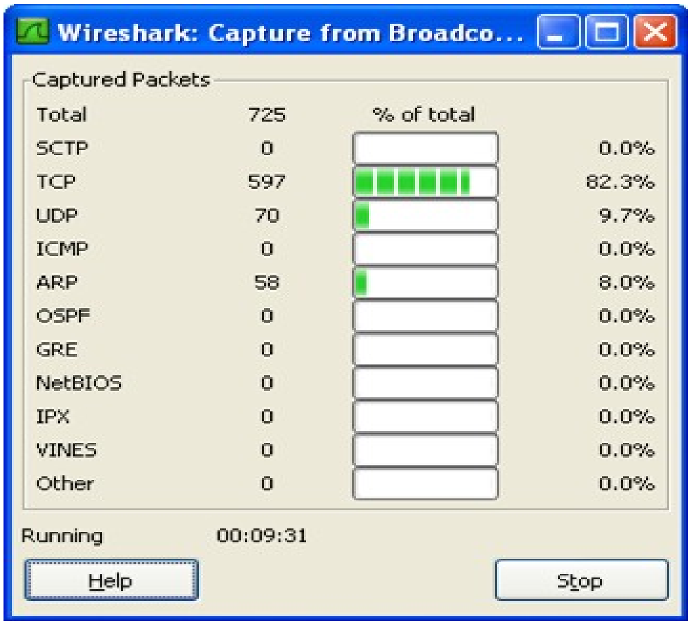
✓ Enable transport name resolution

Help Start Cancel

There are several steps that you need to follow before you start capturing:

- You normally will use the default values in the *capture options dialog box*, but there are some options that you may need to override the default. In particular, you may want to uncheck "Hide capture info dialog." and open capture information window (Figure 2.6).
- The network interfaces are shown in the Interface drop-down list at the top of the dialog box. Select the network interface (or use the default interface chosen by Wireshark). If the IP address in the dialog box is unknown, you must select a different interface; otherwise, the Wireshark will not capture any packet

Figure 2.6 *Capture information dialog*



Captured Packets

Total	725	% of total
SCTP	0	0.0%
TCP	597	82.3%
UDP	70	9.7%
ICMP	0	0.0%
ARP	58	8.0%
OSPF	0	0.0%
GRE	0	0.0%
NetBIOS	0	0.0%
IPX	0	0.0%
VINES	0	0.0%
Other	0	0.0%

Running 00:09:31

.

- After the above two steps, click Start. Wireshark starts to captures packets that are exchanged between your computer and the network. If, after a minute, Wireshark does not capture any packet, there must be a problem; check for possible reason and troubleshooting.

Stop Capturing

Whenever you feel you have captured all the packets (frames) that you need to do your lab report, you can stop capturing. To do so, you need to use the Capture pulldown menu and click **Stop**. Wireshark stops capturing the frames

Saving the Captured File

After you have stopped capturing, you may want to save the captured file for future use.

2.4 **Incoming and Outgoing Frames**

When we see the list of the captured frames, we often wonder which frames are the incoming and which ones are outgoing. This can be found by looking at the frame in packet list pane. The packet list pane shows the source and destination addresses of the frame (generated and inserted at the network layer). If the source address is the address of the host you are working with (shown on the Capture window when you start capturing), the frame is the outgoing frame; if the destination address is the address of your host, and the frame is the incoming frame.

2.5 **Lab-Report Sheets**

To make the report of your observation in each chapter easier and consistent, we have created lab report sheets for each lab assignment. Each chapter has as many lab report sheets as the number of lab assignments for that chapter. Each lab report sheet has been prepared as a Word documents to let the student fill up and turn in.

2.6 **Printing the Captured Information**

As a supporting document for each lab assignment, you need to turn in a printout of the captured information. You can do this by selecting the packet and expanding it in the packet detailed pane, selecting **Print** from the **File** menu, selecting the necessary printing options in the Wireshark print window, and finally clicking the Print button.

3. LAB1 : DNS + Wireshark

The lab assignment for this chapter is a warm-up testing of the Wireshark software. In this lab, we retrieve a web page and then, using Wireshark, capture packets.

There are several network administration tools for Microsoft Windows and UNIX-like operating systems that are useful for network troubleshooting as well as for educational purposes. Among these tools are the following:

- **dig** (Domain Information Groper) is used for querying DNS servers. This utility replaces older tools such as *nslookup*.
- **ipconfig** (Internet Protocol Configuration) for Windows or **ifconfig** (Interface Configuration) for UNIX-like operating systems is used to configure, control, and query TCP/IP network interface parameters.

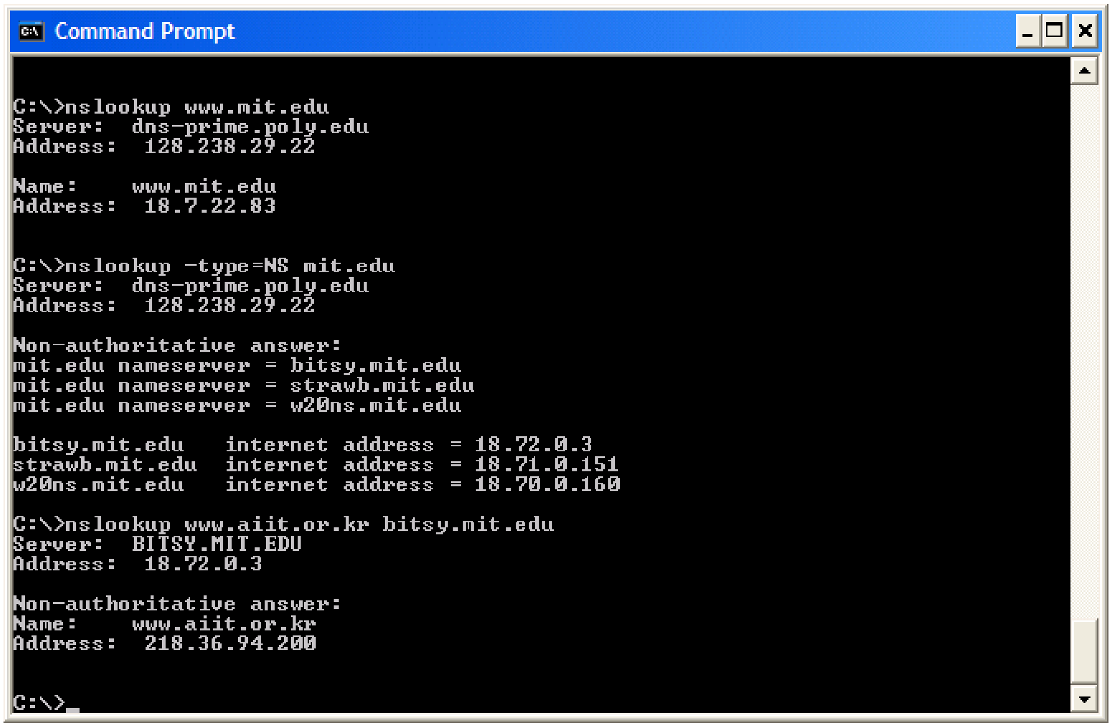
3.1 Assignment

This lab is **made of three parts**. In Part I, we use the nslookup. In Part II, we use ipcon- fig utility. Finally, in Part III, we use Wireshark to find more information about the packets exchanged by the DNS protocol. Pick **an educational institute outside of the state of Colorado**, and use its Hostname or IP address when you run “dig” using both CLI based and Web based.

3.1 nslookup

In this lab, we’ll make extensive use of the *nslookup* tool, which is available in most Linux/Unix and Microsoft platforms today. To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line.

In it is most basic operation, *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 (see the textbook for definitions of these terms). 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.



```
C:\>nslookup www.mit.edu
Server:  dns-prime.poly.edu
Address:  128.238.29.22

Name:    www.mit.edu
Address:  18.7.22.83


C:\>nslookup -type=NS mit.edu
Server:  dns-prime.poly.edu
Address:  128.238.29.22

Non-authoritative answer:
mit.edu nameserver = bitsy.mit.edu
mit.edu nameserver = strawb.mit.edu
mit.edu nameserver = w20ns.mit.edu

bitsy.mit.edu   internet address = 18.72.0.3
strawb.mit.edu  internet address = 18.71.0.151
w20ns.mit.edu   internet address = 18.70.0.160

C:\>nslookup www.aiit.or.kr bitsy.mit.edu
Server:  BITSY.MIT.EDU
Address:  18.72.0.3

Non-authoritative answer:
Name:    www.aiit.or.kr
Address:  218.36.94.200


C:\>_
```

The above screenshot shows the results of three independent *nslookup* commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of Polytechnic University in Brooklyn, where the default local DNS server is dns-prime.poly.edu. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is dns-prime.poly.edu. Consider the first command:

```
nslookup www.mit.edu
```

In words, this command is saying “please send me the IP address for the host www.mit.edu”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of www.mit.edu. Although the response came from the local DNS server at Polytechnic University, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 4.17 of the textbook.

Now consider the second command:

```
nslookup –type=NS mit.edu
```

In this example, we have provided the option "-type=NS" and the domain "mit.edu". This causes *nslookup* to send a query for a type-NS record to the default local DNS server. In words, the query is saying, "please send me the host names of the authoritative DNS for mit.edu". (When the –type option is not used, *nslookup* uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer (which is the default local DNS server) along with three MIT nameservers. Each of these servers is indeed an authoritative DNS server for the hosts on the MIT campus. However, *nslookup* also indicates that the answer is "non-authoritative," meaning that this answer came from the cache of some server rather than from an authoritative MIT DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers at MIT. (Even though the type-NS query generated by *nslookup* did not explicitly ask for the IP addresses, the local DNS server returned these "for free" and *nslookup* displays the result.)

Now finally consider the third command:

```
nslookup www.snu.ac.kr google-piblic-dns-a.google.com
```

In this example, we indicate that we want to the query sent to the DNS server bitsy.mit.edu rather than to the default DNS server (dns-prime.poly.edu). Thus, the query and reply transaction takes place directly between our querying host and bitsy.mit.edu. In this example, the DNS server bitsy.mit.edu provides the IP address of the host www.aiit.or.kr, which is a web server at the Advanced Institute of Information Technology (in Korea).

Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of *nslookup* commands. The syntax is:

```
nslookup –option1 –option2 host-to-find dns-server
```

In general, *nslookup* can be run with zero, one, two or more options. And as we have seen in the above examples, the dns-server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself.

Questions (Do the following (and write down the results)):

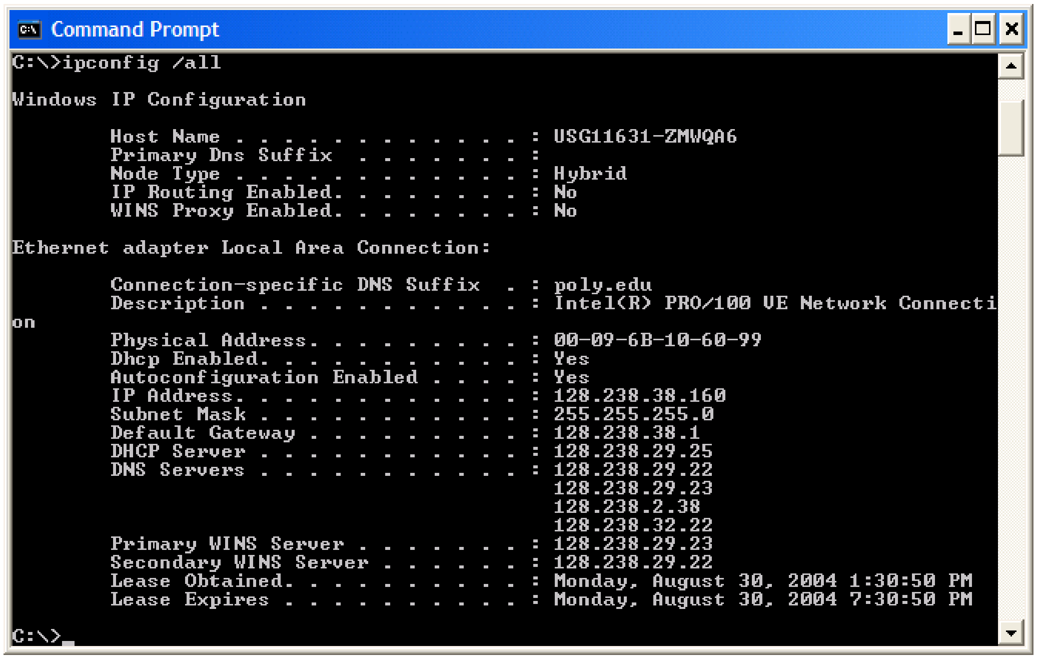
1. Run *nslookup* to obtain the IP address of a Web server in Asia. What is the IP address of that server?
2. Run *nslookup* to determine the authoritative DNS servers for a university in Europe.
3. Run *nslookup* so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail.   What is its IP address?

3.2 Using ipconfig

ipconfig (for Windows) and *ifconfig* (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues. Here we'll only describe *ipconfig*, although the Linux/Unix *ifconfig* is very similar. *ipconfig* can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you all this information about your host simply by entering

```
ipconfig \all
```

into the Command Prompt, as shown in the following screenshot.



```
C:\>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : USG11631-ZMWQA6
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : poly.edu
        Description . . . . . . . . . . . : Intel(R) PRO/100 VE Network Connecti
on
        Physical Address. . . . . . . . . : 00-09-6B-10-60-99
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 128.238.38.160
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 128.238.38.1
        DHCP Server . . . . . . . . . . . : 128.238.29.25
        DNS Servers . . . . . . . . . . . : 128.238.29.22
                                            128.238.29.23
                                            128.238.2.38
                                            128.238.32.22
        Primary WINS Server . . . . . . . : 128.238.29.23
        Secondary WINS Server . . . . . . : 128.238.29.22
        Lease Obtained. . . . . . . . . . : Monday, August 30, 2004 1:30:50 PM
        Lease Expires . . . . . . . . . . : Monday, August 30, 2004 7:30:50 PM

C:\>_
```

ipconfig is also very useful for managing the DNS information stored in your host. In Section 4.17 we learned that a host can cache DNS records it recently obtained. To see these cached records, after the prompt C:\> provide the following command:

```
ipconfig /displaydns
```

Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter

```
ipconfig /flushdns
```

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

Tracing DNS with Wireshark

Now that we are familiar with *nslookup* and *ipconfig*, we're ready to get down to some serious business. Let's first capture the DNS packets that are generated by ordinary Web-surfing activity.

- Use *ipconfig* to empty the DNS cache in your host.
- Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)
- Open Wireshark and enter "ip.addr == your_IP_address" into the filter, where you obtain your_IP_address with ipconfig. This filter removes all packets that neither originate nor are destined to your host.
- Start packet capture in Wireshark.
- With your browser, visit the Web page: http://www.ietf.org
- Stop packet capture.

Answer the following questions. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout[1] to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

Questions

Using the result of running *ipconfig*, answer the following question in your lab report.

4. Locate the DNS query and response messages. Are then sent over UDP or TCP?
5. What is the destination port for the DNS query message? What is the source port of DNS response message?

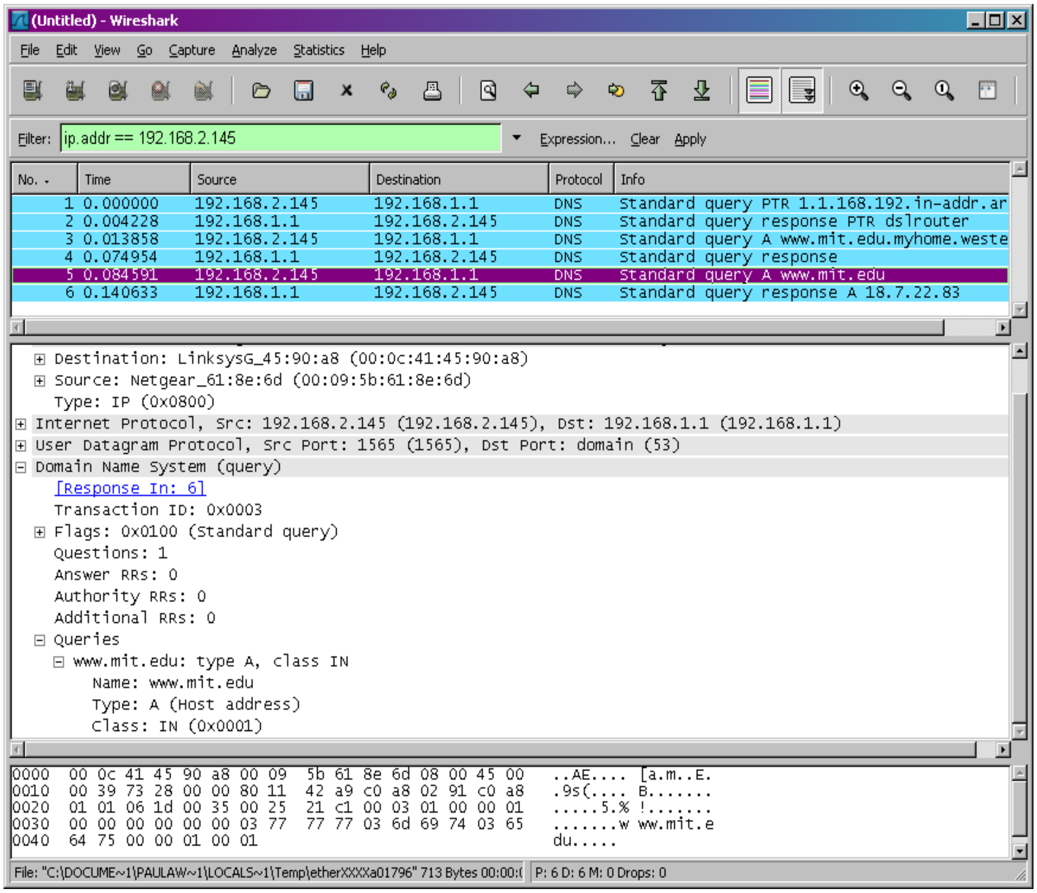
[1] What do we mean by "annotate"? If you hand in a paper copy, please highlight where in the printout you've found the answer and add some text (preferably with a colored pen) noting what you found in what you 've highlight. If you hand in an electronic copy, it would be great if you could also highlight and annotate.

6. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
7. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
8. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
9. 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?
10. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

3,.3 Using Wireshark to Capture DNS Packets

Now let’s play with *nslookup*.

- Start packet capture.
- Do an *nslookup* on www.mit.edu
- Stop packet capture.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: ip.addr == 192.168.2.145
Expression... Clear Apply

```
No. .  Time      Source         Destination    Protocol  Info
  1 0.000000  192.168.2.145  192.168.1.1    DNS  Standard query PTR 1.1.168.192.in-addr.ar
  2 0.004228  192.168.1.1    192.168.2.145  DNS  Standard query response PTR dslrouter
  3 0.013858  192.168.2.145  192.168.1.1    DNS  Standard query A www.mit.edu.myhome.weste
  4 0.074954  192.168.1.1    192.168.2.145  DNS  Standard query response
  5 0.084591  192.168.2.145  192.168.1.1    DNS  Standard query A www.mit.edu
  6 0.140633  192.168.1.1    192.168.2.145  DNS  Standard query response A 18.7.22.83
```

```
  ⊞ Destination: LinksysG_45:90:a8 (00:0c:41:45:90:a8)
  ⊞ Source: Netgear_61:8e:6d (00:09:5b:61:8e:6d)
    Type: IP (0x0800)
⊞ Internet Protocol, Src: 192.168.2.145 (192.168.2.145), Dst: 192.168.1.1 (192.168.1.1)
⊞ User Datagram Protocol, Src Port: 1565 (1565), Dst Port: domain (53)
⊟ Domain Name System (query)
    [Response In: 6]
    Transaction ID: 0x0003
  ⊞ Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ www.mit.edu: type A, class IN
        Name: www.mit.edu
        Type: A (Host address)
        Class: IN (0x0001)
```

```
0000  00 0c 41 45 90 a8 00 09  5b 61 8e 6d 08 00 45 00   ..AE.... [a.m..E.
0010  00 39 73 28 00 00 80 11  42 a9 c0 a8 02 91 c0 a8   .9s(.... B.......
0020  01 01 06 1d 00 35 00 25  21 c1 00 03 01 00 00 01   .....5.% !.......
0030  00 00 00 00 00 00 03 77  77 77 03 6d 69 74 03 65   .......w ww.mit.e
0040  64 75 00 00 01 00 01                               du.....
```

```
File: "C:\DOCUME~1\PAULAW~1\LOCALS~1\Temp\etherXXXXa01796" 713 Bytes 00:00:( | P: 6 D: 6 M: 0 Drops: 0
```

You should get a trace that looks something like the following:

We see from the above screenshot that *nslookup* actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, 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.

Questions

11. What is the destination port for the DNS query message? What is the source port of DNS response message?
12. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
13. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
14. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?
15. Provide a screenshot.

Now repeat the previous experiment, but instead issue the command:

```
nslookup –type=NS mit.edu
```

Questions

16. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
17. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
18. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?
19. Provide a screenshot.

Now repeat the previous experiment, but instead issue the command:

```
nslookup www.snu.ac.kr  google-public-dns-a.google.com
```

Answer the following questions:

20. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?
21. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
22. Examine the DNS response message. How many "answers" are provided? What does each of these answers contain?
23. Provide a screenshot.

4. Documents to Turn in

Turn in the following documents:

A hard copy of the Lab1-DNS report sheet that contains answered questions and the screen captures of all three parts for your answers.