Wireshark Lab: DNS (15 points)

This lab is adapted from “Wireshark Lab: DNS v7.0”, a supplement to Computer Networking: A Top-Down Approach, 7th ed., J.F. Kurose and K.W. Ross

As described in Section 2.4 of the text, the Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we’ll take a closer look at the client side of DNS. Recall that the client’s role in the DNS is relatively simple – a client sends a query to its local DNS server, and receives a response back. As shown in Figures 2.19 and 2.20 in the textbook, much can go on “under the covers,” invisible to the DNS clients, as the hierarchical DNS servers communicate with each other to either recursively or iteratively resolve the client’s DNS query. From the DNS client’s standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

Before beginning this lab, you’ll probably want to review DNS by reading Section 2.4 of the text. In particular, you may want to review the material on local DNS servers, DNS caching, DNS records and messages, and the TYPE field in the DNS record.

If you haven’t used Wireshark before, follow the instructions in “Wirehshark Lab: Getting Started v7.0”.

When answering the following questions, provide screenshots whenever possible.

1. nslookup

In this lab, we’ll make extensive use of the nslookup tool, which is available in most Linux/Mac/Windows platforms today. To run it in Windows, open the Command Prompt and run nslookup on the command line. To run it in Mac, open a terminal window and run nslookup on the command line.

In its 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.

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 2.4 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 name servers. 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.aiit.or.kr bitsy.mit.edu
```

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. 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? What is the IP address of your local DNS server?
2. Run nslookup to determine the authoritative DNS servers for a university in Europe.
2. Tracing DNS with Wireshark

Now that we are familiar with *nslookup*, 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.

- Open your browser and empty your browser cache.
- Open Wireshark and enter “ip.addr == your_IP_address” into the filter. You can obtain your_IP_address by running ipconfig (for Windows) or ifconfig (for Mac) on the command line. 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.

Enter “dns” into the filter to show DNS packets only. 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 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.

3. Locate the DNS query and response messages. Are they sent over UDP or TCP?
4. What is the destination port for the DNS query message? What is the source port of DNS response message?
5. To what IP address is the DNS query message sent? Is it the same as your local DNS server?
6. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”? 
7. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
8. Provide a screenshot.

Now let’s play with *nslookup*.

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

You should get a trace that looks something like the following (use the “dns” filter):
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.

9. What is the destination port for the DNS query message? What is the source port of DNS response message?

10. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

11. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

12. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?

13. Provide a screenshot.