

Homework+Lab 2: Due at start of class on 01-Apr.

Please complete all five homework problems, and the twenty lab problems.

Homework Problems

1. Stallings Problem 2.4 from the “Problems” section, not the “Review Questions”.
2. Stallings Problem 2.7.
3. The signal $x(t) = \frac{1}{2} \sin(t)$ (in volts) undergoes an attenuation of 6 dB. What is the peak-to-peak voltage of the attenuated signal?
4. Suppose a communication system needs 11 dB SNR for reliable communication. Assume that the noise floor is -100 dBm, and that a fiber optic cable with an attenuation of 0.12 dB/km is chosen to connect the transmitter and receiver. What is the required transmit power (in watts) if there is 40 km of cable between the transmitter and receiver?
5. Stallings Problem 4.3.

Lab 2: Application Layer DNS

[adapted from J. Kurose and K.W. Ross]

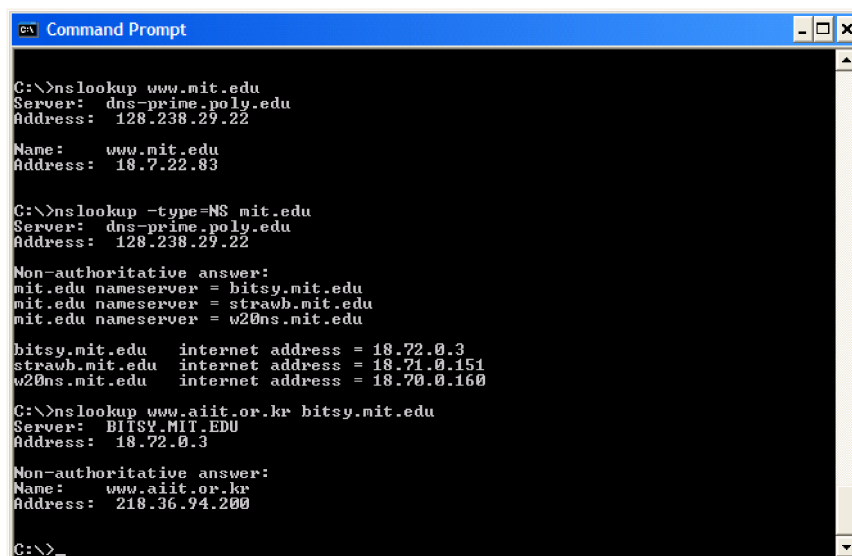
Before starting this lab, please make sure you have read section 24.2 in the textbook. As described in the textbook, the Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we'll take a closer look at the client side of 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 Figure 24.6 in the textbook, much can go on behind the scenes, 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.

What to hand in: *There are twenty questions below which you will encounter as you complete the lab. For each question, please provide the answer as well as necessary screenshots and packet printouts to support your answer.*

Part I: nslookup¹

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 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.



```

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 = straub.mit.edu
mit.edu nameserver = w20ns.mit.edu

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

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

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

C:\>
```

Figure 1: Screenshot demonstrating use of `nslookup`

¹OS X and Linux users (and Windows users who have installed Cygwin) may consider using `dig` rather than `nslookup` since `dig` is a more modern tool with more detailed outputs and is actively maintained.

The screenshot in Fig. 1 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 `dnsprime.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 24.2 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`.” 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 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 (remember, you must give the answers and provide the necessary screenshots to support your answers):

1. Run `nslookup` to obtain the IP address of a Web server in Europe.

2. Run `nslookup` to determine the authoritative DNS servers for a university in Asia.
3. Run `nslookup` so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail.

Part II: `ipconfig`

The `ipconfig` (for Windows) and `ifconfig` (for Linux/Unix/Mac) utilities are among the most useful little in your host, especially for debugging network issues. Here we'll only describe `ipconfig`, although the Linux/Unix/Mac `ifconfig` is very similar². `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 want to see all this information about your host, simply enter:

```
ipconfig /all
```

into the Command Prompt, as shown in the screenshot in Fig. 2.

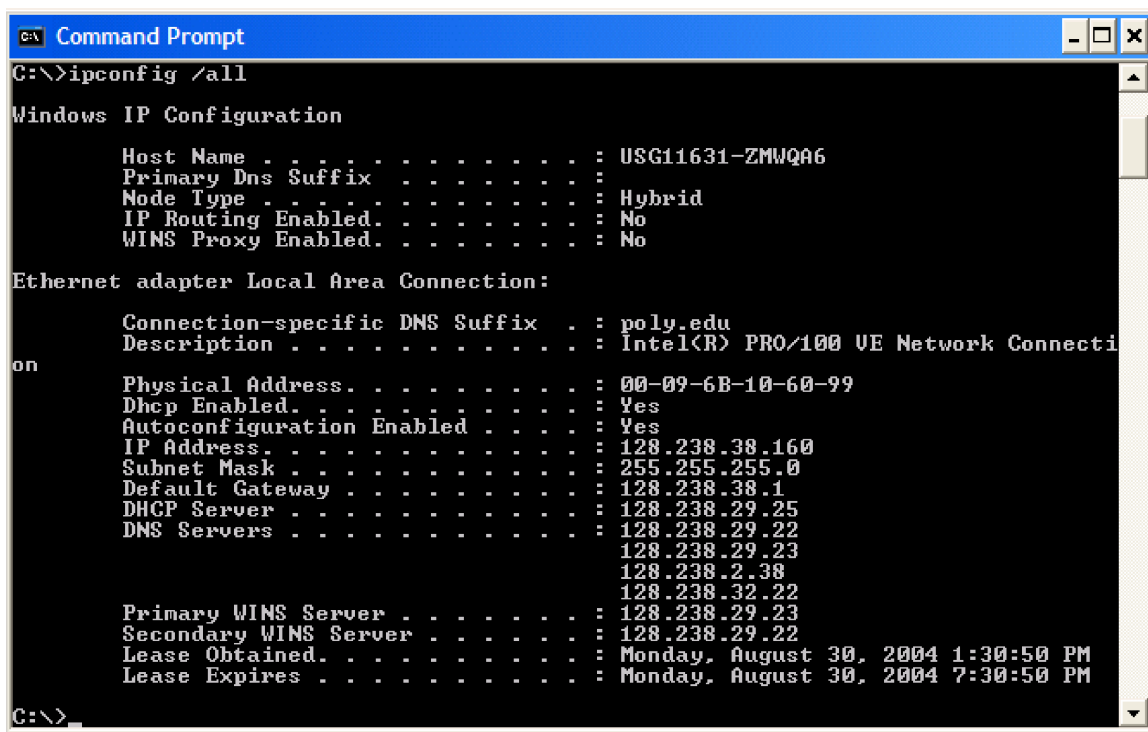


Figure 2: Screenshot demonstrating use of `ipconfig`

`ipconfig` is also very useful for managing the DNS information stored in your host. As mentioned in section 24.2 of the text, a host can cache DNS records it recently obtained. To see these cached records, after the command prompt 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.

²Type `man ifconfig` for help with this utility in Linux/Unix

Part III: Tracing DNS with Wireshark (and stuff to hand in)

Now that we are familiar with `nslookup` and `ipconfig`, we're ready to put it to use. Let's first capture the DNS packets that are generated by ordinary web browsing activity.

- Use `ipconfig` to empty/flush 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. With Firefox, go to Tools and select Clear Private Data)
- Open Wireshark and enter `ip.addr == your_IP_address` into the filter, where you obtain `your_IP_address` (the IP address for the computer on which you are running Wireshark) 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:

4. Locate the DNS query and response messages. Are they sent using the UDP or TCP protocol?
5. What is the destination port for the DNS query message? What is the source port of DNS response message?
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 it³? Does the query message contain any “answers”?
8. Examine the DNS response message. How many “answers” are provided? What does 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?

³A table of all the different “types” is available in your textbook (see “resource record types”).

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 what is shown in Fig. 3.

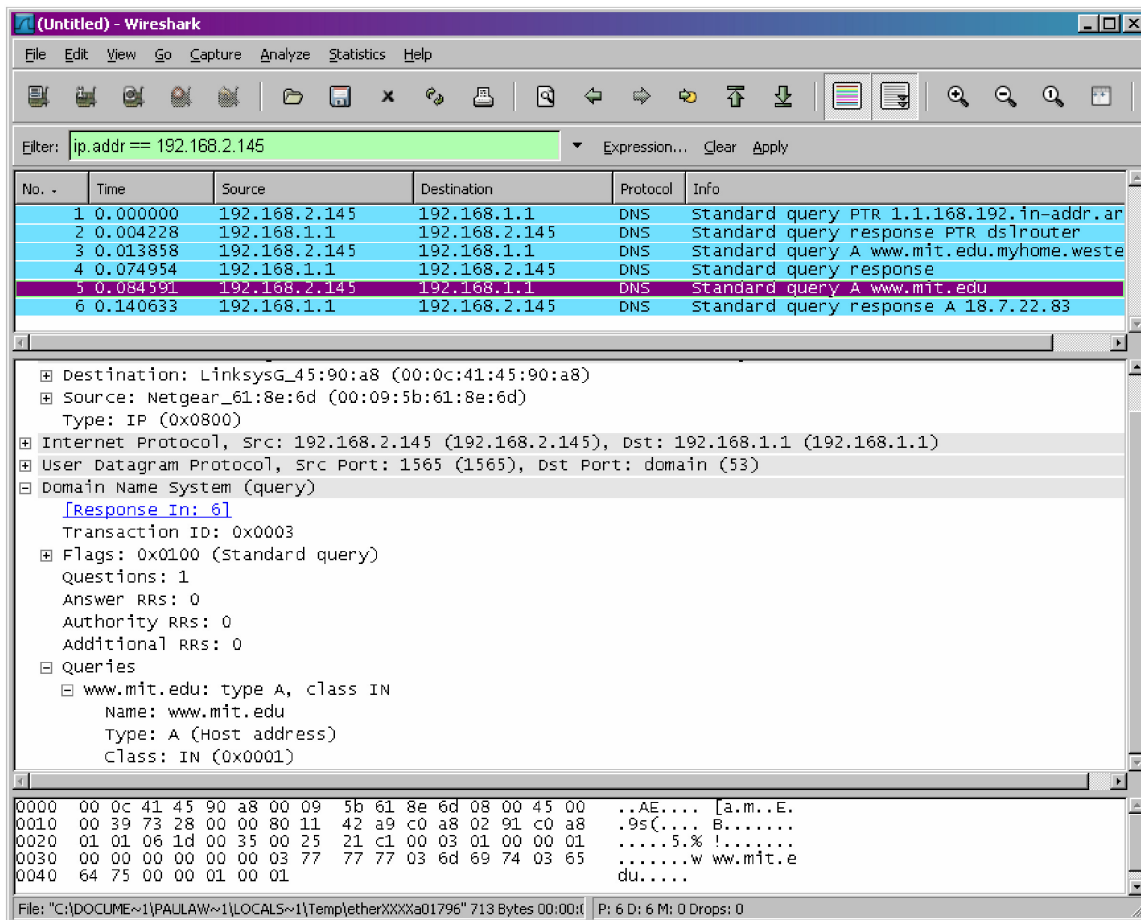


Figure 3: Example Wireshark trace of DNS requests

We see from Fig. 3 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.

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 does each of these answers contain? Provide a screenshot.

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

```
nslookup -type=NS mit.edu
```

Answer the following questions:

15. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
16. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
17. Examine the DNS response message. What MIT name servers does the response message provide? Does this response message also provide the IP addresses of the MIT name servers? Provide a screenshot.

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

```
nslookup www.aiit.or.kr bitsy.mit.edu
```

Answer the following questions:

18. 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?
19. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
20. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain? Provide a screenshot.