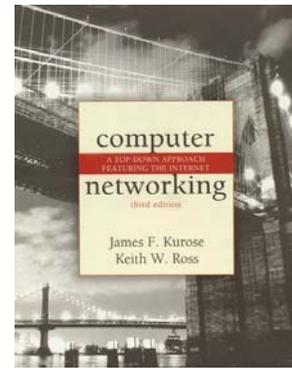


Ethereal Lab: DNS



Computer Networking: A Top-down Approach Featuring the Internet, 3rd edition.

Version: July 2005

© 2005 J.F. Kurose, K.W. Ross. All Rights Reserved

As described in Section 2.5 of 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 Figures 2.26 and 2.18 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 2.5 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.

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 its most basic operation, the 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.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:\>
```

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 2.5 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.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:

- 1) Run nslookup to obtain the IP address of a Web server in Asia.
- 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.

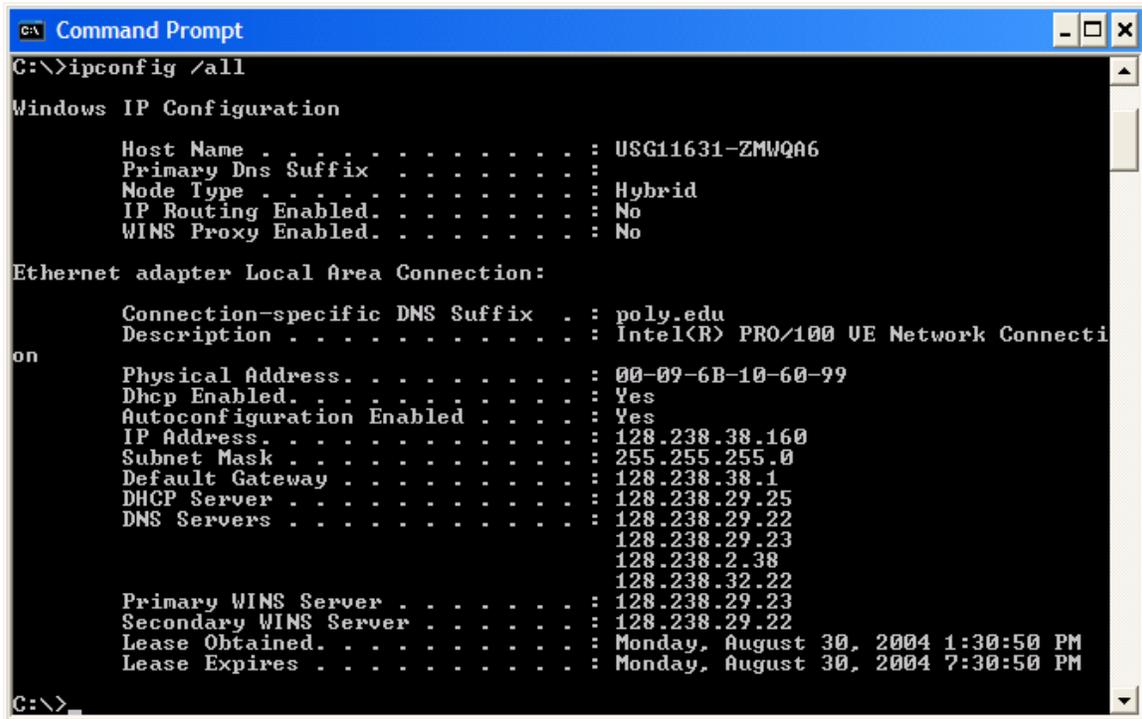
2. ipconfig

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 DNS information stored in your host. In Section 2.5 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.

3. Tracing DNS with Ethereal

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.

1. Use ipconfig to empty the DNS cache in your host.
2. 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.)
3. Open Ethereal 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.
4. Start packet capture in Ethereal.
5. With your browser, visit the Web page: <http://www.ietf.org>
6. Stop packet capture.

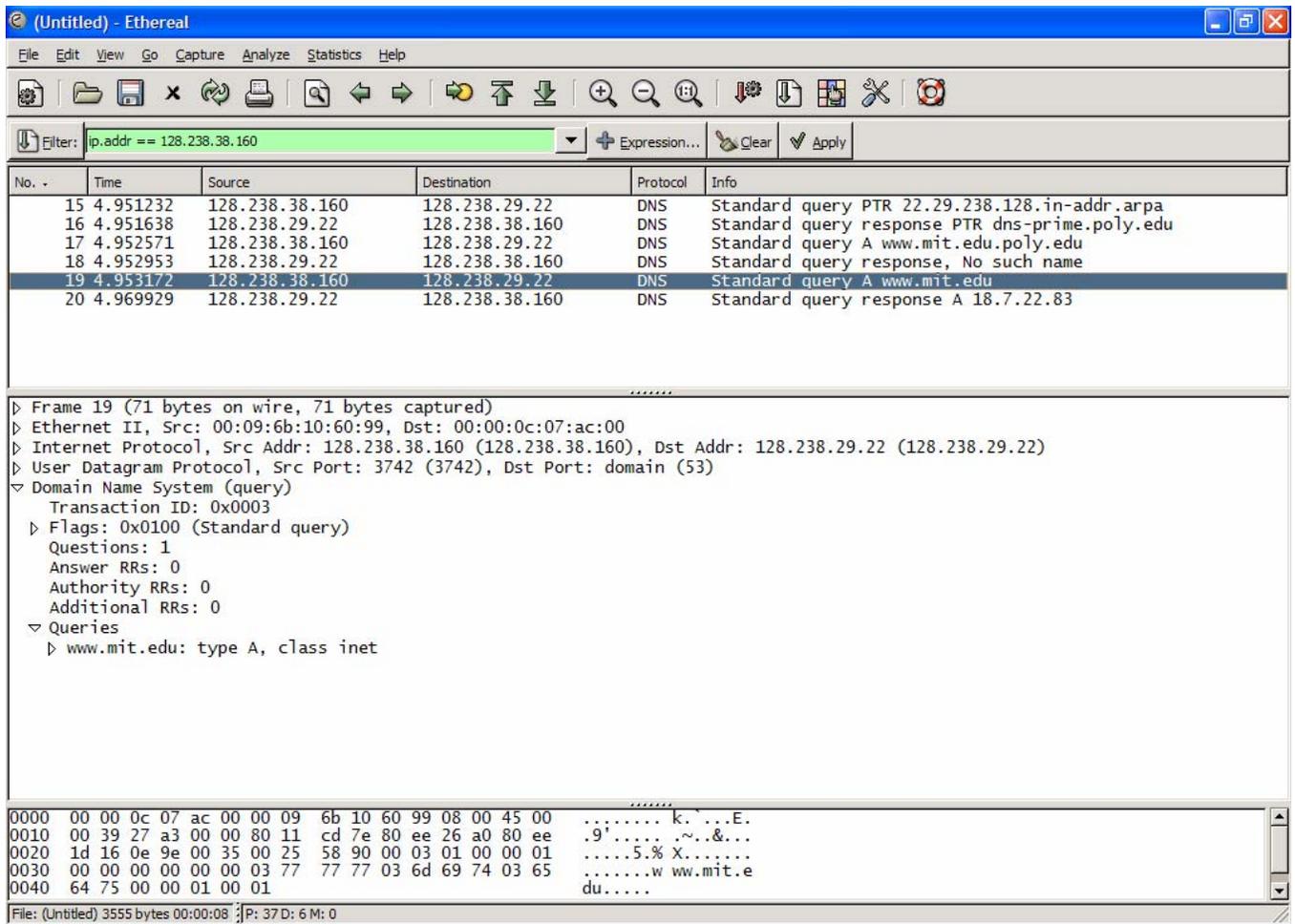
Answer the following questions:

1. Locate the DNS query and response messages. Are they sent over UDP or TCP?
2. What is the destination port for the DNS query message? What is the source port of DNS response message?
3. 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?
4. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
5. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
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?
7. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Now let’s play with nslookup.

1. Start packet capture.
2. Do an nslookup on www.mit.edu
3. Stop packet capture.

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.

1. What is the destination port for the DNS query message? What is the source port of DNS response message?
2. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
3. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
4. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
5. Provide a screenshot.

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

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

Answer the following questions:

1. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
2. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
3. 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 nameservers?
4. Provide a screenshot.

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

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

Answer the following questions:

1. 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?
2. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
3. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?
4. Provide a screenshot.
5. If you instead issue the command `nslookup www.aiit.or.kr` after flushing the dns cache, do you pass through the same nameservers to get the answer?