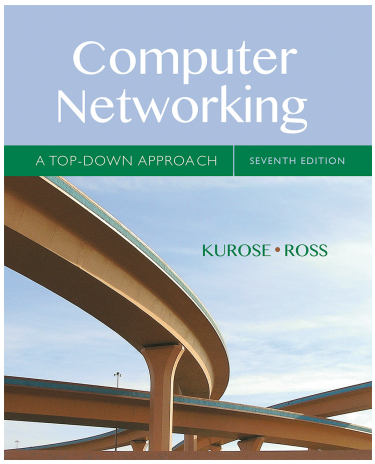
Wireshark Lab: DNS v7.0

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



"Tell me and I forget. Show me and I remember. Involve me and I understand." Chinese proverb

---

As described in Section 2.4 of the text[1], 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.

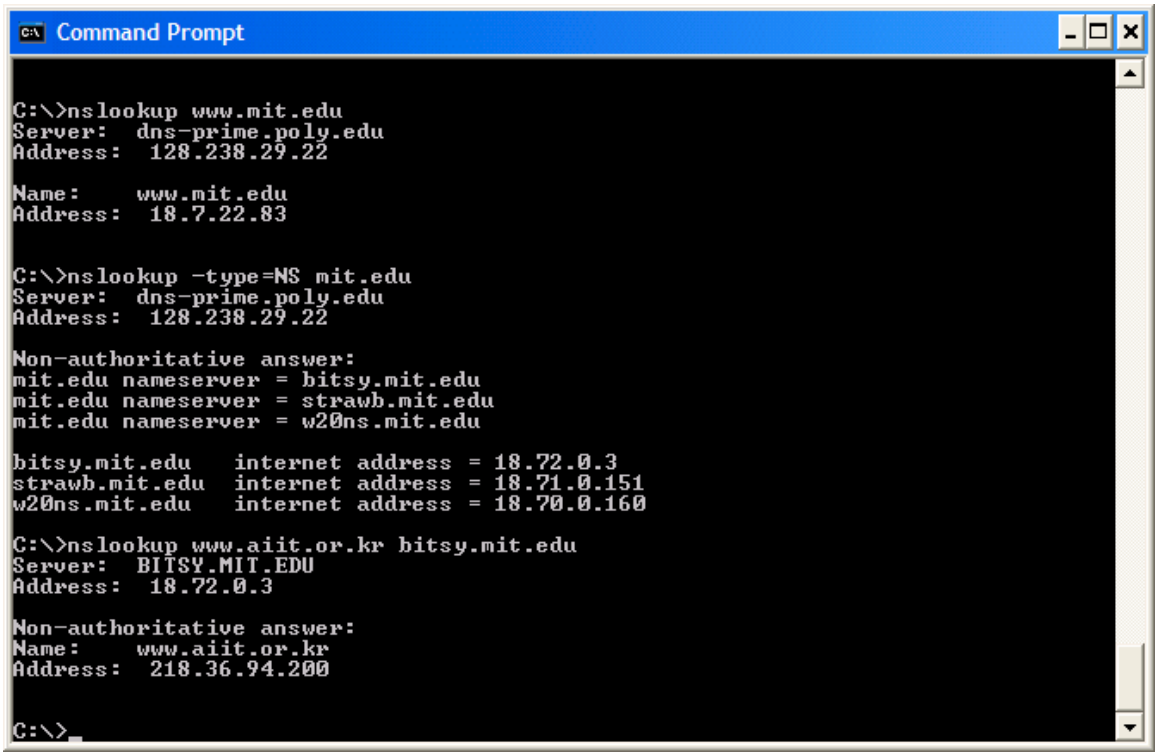
1. nslookup

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.

---

[1] References to figures and sections are for the 7th edition of our text, *Computer Networks, A Top-down Approach, 7th ed.,* J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2016.



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 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 (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?

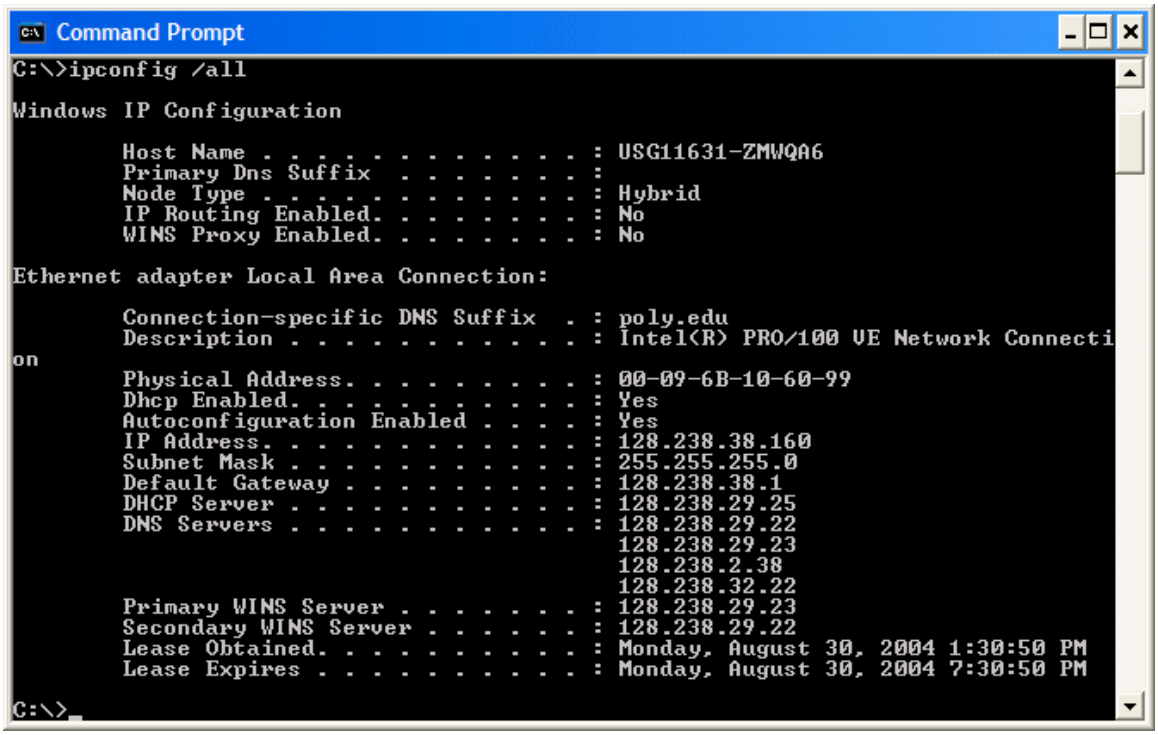
2. ipconfig

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.



ipconfig is also very useful for managing the 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 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.

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author's computers[2]. 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[3] 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.

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?
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?

---

[2] Download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zipand extract the file dns-ethereal-trace-1. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the dns-ethereal-trace-1 trace file.

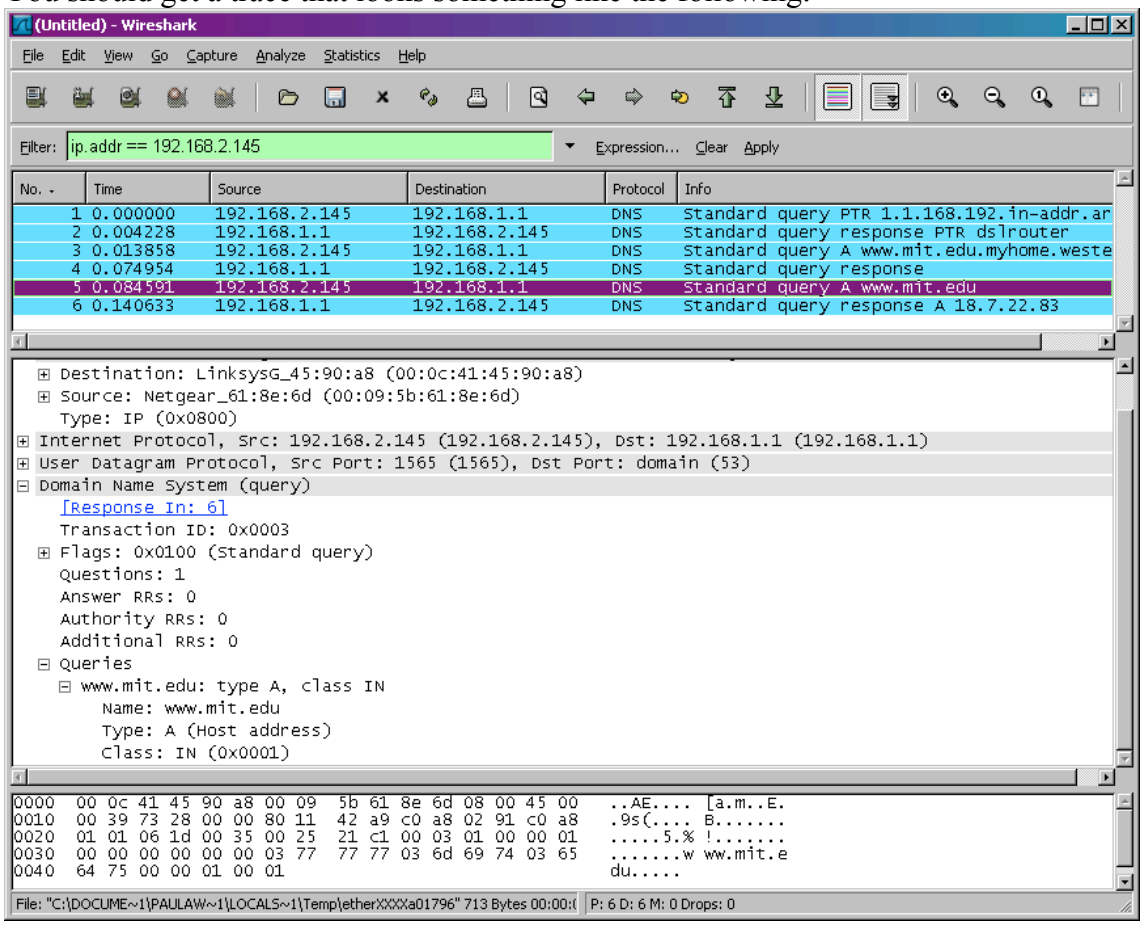
[3] What do we mean by "annotate"? If you hand in a paper copy, please highlight where in the printout you've found the answer and add some text (preferably with a colored pen) noting what you found in what you 've highlight. If you hand in an electronic copy, it would be great if you could also highlight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

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?

Now let's play with *nslookup*[4].

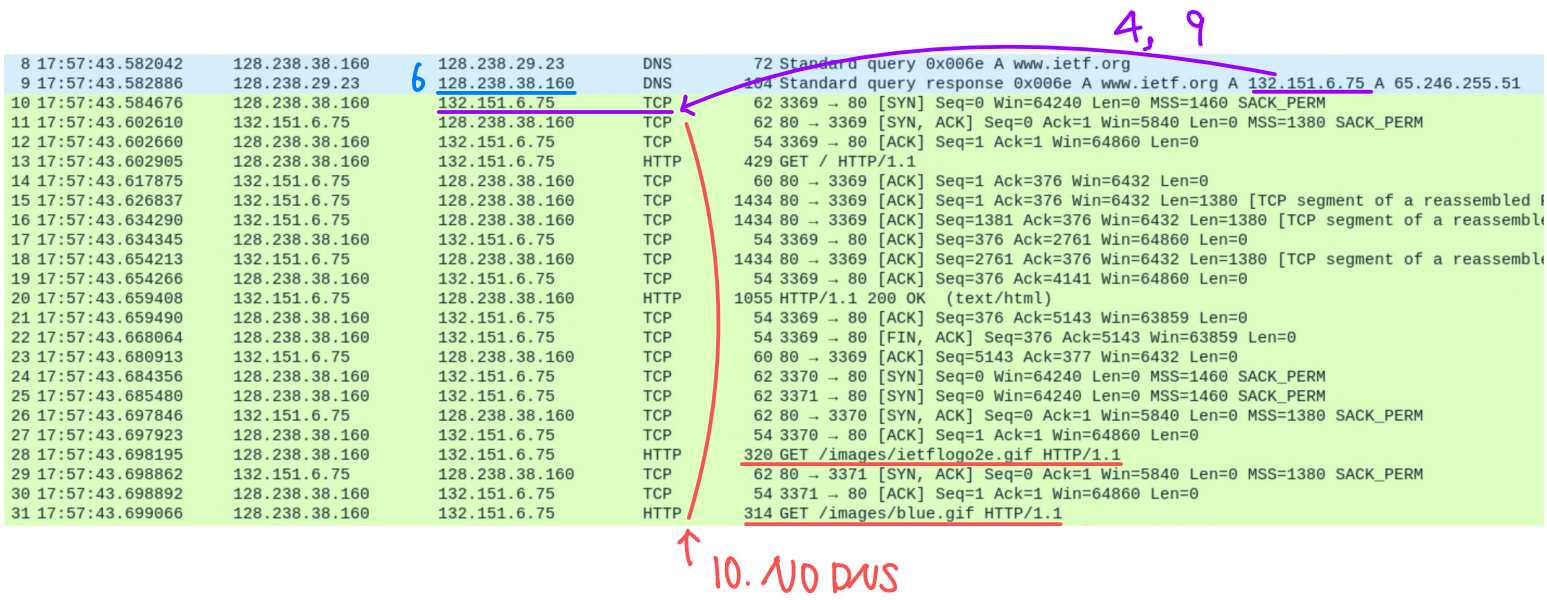
- Start packet capture.
- Do an *nslookup* on www.mit.edu
- Stop packet capture.

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

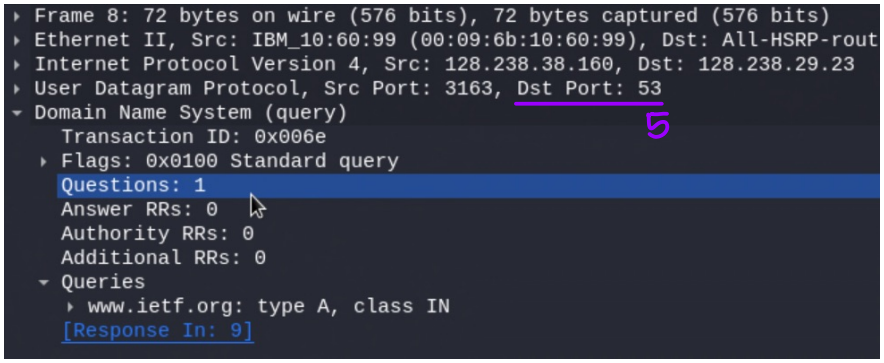


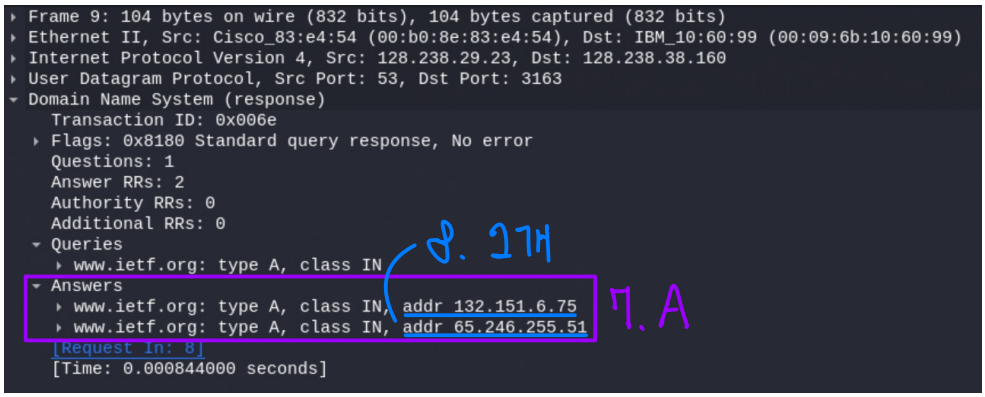
We see from the above screenshot that *nslookup* actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first two sets of queries/responses, as they are specific to *nslookup* and are not normally generated by standard Internet applications. You should instead focus on the last query and response messages.

[4] If you are unable to run Wireshark and capture a trace file, use the trace file dns-ethereal-trace-2 in the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip



8	17:57:43.582042	128.238.38.160	128.238.29.23	DNS	72 Standard query 0x006e A www.ietf.org
9	17:57:43.582886	128.238.29.23	128.238.38.160	DNS	104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
10	17:57:43.584676	128.238.38.160	132.151.6.75	TCP	62 3369 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
11	17:57:43.602610	132.151.6.75	128.238.38.160	TCP	62 80 → 3369 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM
12	17:57:43.602660	128.238.38.160	132.151.6.75	TCP	54 3369 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
13	17:57:43.602905	128.238.38.160	132.151.6.75	HTTP	429 GET / HTTP/1.1
14	17:57:43.617875	132.151.6.75	128.238.38.160	TCP	60 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=0
15	17:57:43.626837	132.151.6.75	128.238.38.160	TCP	1434 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled P
16	17:57:43.634290	132.151.6.75	128.238.38.160	TCP	1434 80 → 3369 [ACK] Seq=1381 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembl
17	17:57:43.634345	128.238.38.160	132.151.6.75	TCP	54 3369 → 80 [ACK] Seq=376 Ack=2761 Win=64860 Len=0
18	17:57:43.654213	132.151.6.75	128.238.38.160	TCP	1434 80 → 3369 [ACK] Seq=2761 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembl
19	17:57:43.654266	128.238.38.160	132.151.6.75	TCP	54 3369 → 80 [ACK] Seq=376 Ack=4141 Win=64860 Len=0
20	17:57:43.659408	132.151.6.75	128.238.38.160	HTTP	1055 HTTP/1.1 200 OK (text/html)
21	17:57:43.659490	128.238.38.160	132.151.6.75	TCP	54 3369 → 80 [ACK] Seq=376 Ack=5143 Win=63859 Len=0
22	17:57:43.668064	128.238.38.160	132.151.6.75	TCP	54 3369 → 80 [FIN, ACK] Seq=376 Ack=5143 Win=63859 Len=0
23	17:57:43.680913	132.151.6.75	128.238.38.160	TCP	60 80 → 3369 [ACK] Seq=5143 Ack=377 Win=6432 Len=0
24	17:57:43.684356	128.238.38.160	132.151.6.75	TCP	62 3370 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
25	17:57:43.685480	128.238.38.160	132.151.6.75	TCP	62 3371 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
26	17:57:43.697846	132.151.6.75	128.238.38.160	TCP	62 80 → 3370 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM
27	17:57:43.697923	128.238.38.160	132.151.6.75	TCP	54 3370 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
28	17:57:43.698195	128.238.38.160	132.151.6.75	HTTP	320 GET /images/ietflogo2e.gif HTTP/1.1
29	17:57:43.698862	132.151.6.75	128.238.38.160	TCP	62 80 → 3371 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM
30	17:57:43.698892	128.238.38.160	132.151.6.75	TCP	54 3371 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
31	17:57:43.699066	128.238.38.160	132.151.6.75	HTTP	314 GET /images/blue.gif HTTP/1.1





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
```

Answer the following questions[5] :

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

Answer the following questions[6]:

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.

[5] If you are unable to run Wireshark and capture a trace file, use the trace file dns-ethereal-trace-3 in the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

[6] If you are unable to run Wireshark and capture a trace file, use the trace file dns-ethereal-trace-4 in the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

No.	Time	Source	Destination	Protocol	Length	Info
15	17:06:12.186083	128.238.38.160	128.238.29.22	DNS	86	Standard query 0x0001 PTR 22.29.238.128.in-addr.arpa
16	17:06:12.186489	128.238.29.22	128.238.38.160	DNS	118	Standard query response 0x0001 PTR 22.29.238.128.in-addr.arpa PTR dns-prime.poly.edu
17	17:06:12.187422	128.238.38.160	128.238.29.22	DNS	80	Standard query 0x0002 A www.mit.edu.poly.edu
18	17:06:12.187804	128.238.29.22	128.238.38.160	DNS	139	Standard query response 0x0002 No such name A www.mit.edu.poly.edu SOA dns-prime.poly.edu
19	17:06:12.188023	128.238.38.160	128.238.29.22	DNS	71	Standard query 0x0003 A www.mit.edu
20	17:06:12.204780	128.238.29.22	128.238.38.160	DNS	196	Standard query response 0x0003 A www.mit.edu A 18.7.22.83 NS BITSY.mit.edu NS STRAWB.mit.e

```
▸ Frame 15: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
▾ Ethernet II, Src: IBM_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
  ▸ Destination: All-HSRP-routers_00 (00:00:0c:07:ac:00)
  ▸ Source: IBM_10:60:99 (00:09:6b:10:60:99)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.22
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
    Identification: 0x27a1 (10145)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0xcd71 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 128.238.38.160
    Destination Address: 128.238.29.22
▸ User Datagram Protocol, Src Port: 3740, Dst Port: 53
▸ Domain Name System (query)
```

12. 로컬 DNS : 라우터

11

15번 패킷

```
▾ Domain Name System (query)
    Transaction ID: 0x0001
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ 22.29.238.128.in-addr.arpa: type PTR, class IN
    [Response In: 16]
```

17번 패킷

```
▾ Domain Name System (query)
    Transaction ID: 0x0002
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ www.mit.edu.poly.edu: type A, class IN
    [Response In: 18]
```

13. PTR, A, A
Answer 포함X

19번 패킷

```
▾ Domain Name System (query)
    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
    [Response In: 20]
```

16번

```
Domain Name System (response)
    Transaction ID: 0x0001
  ▸ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ 22.29.238.128.in-addr.arpa: type PTR, class IN
  ▾ Answers
    ▸ 22.29.238.128.in-addr.arpa: type PTR, class IN, dns-prime.poly.edu
    [Request In: 15]
    [Time: 0.000406000 seconds]
```

14. 1개
15.

18번

```
Domain Name System (response)
    Transaction ID: 0x0002
  ▸ Flags: 0x8583 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▾ Queries
    ▸ www.mit.edu.poly.edu: type A, class IN
  ▾ Authoritative nameservers
    ▸ poly.edu: type SOA, class IN, mname dns-prime.poly.edu
    [Request In: 17]
    [Time: 0.000382000 seconds]
```

14. 0개

20번

```
Domain Name System (response)
    Transaction ID: 0x0003
  ▸ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 3
  ▾ Queries
    ▸ www.mit.edu: type A, class IN
  ▾ Answers
    ▸ www.mit.edu: type A, class IN, addr 18.7.22.83
  ▾ Authoritative nameservers
    ▸ mit.edu: type NS, class IN, ns BITSY.mit.edu
    ▸ mit.edu: type NS, class IN, ns STRAWB.mit.edu
    ▸ mit.edu: type NS, class IN, ns W20NS.mit.edu
  ▾ Additional records
    ▸ BITSY.mit.edu: type A, class IN, addr 18.72.0.3
    ▸ STRAWB.mit.edu: type A, class IN, addr 18.71.0.151
    ▸ W20NS.mit.edu: type A, class IN, addr 18.70.0.160
    [Request In: 19]
    [Time: 0.016757000 seconds]
```

14. 1개
15

No.	Time	Source	Destination	Protocol	Length	Info
488	17:20:35.848640	128.238.38.160	128.238.29.22	DNS	86	Standard query 0x0001 PTR 22.29.238.128.in-addr.arpa
489	17:20:35.849007	128.238.29.22	128.238.38.160	DNS	118	Standard query response 0x0001 PTR 22.29.238.128.in-addr.arpa PTR dns-prime.poly.edu
490	17:20:35.849848	128.238.38.160	128.238.29.22	DNS	76	Standard query 0x0002 NS mit.edu.poly.edu
491	17:20:35.850192	128.238.29.22	128.238.38.160	DNS	135	Standard query response 0x0002 No such name NS mit.edu.poly.edu SOA dns-prime.poly.edu
492	17:20:35.850423	128.238.38.160	128.238.29.22	DNS	67	Standard query 0x0003 NS mit.edu
493	17:20:35.850784	128.238.29.22	128.238.38.160	DNS	176	Standard query response 0x0003 NS mit.edu NS bitsy.mit.edu NS strawb.mit.edu NS w20ns.mit.edu

16

488번

```
▾ Domain Name System (query)
    Transaction ID: 0x0001
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ 22.29.238.128.in-addr.arpa: type PTR, class IN
    [Response In: 489]
```

490번

```
▸ User Datagram Protocol, Src Port: 3745, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x0002
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ mit.edu.poly.edu: type NS, class IN
    [Response In: 491]
```

17 No answer

492번

```
▸ User Datagram Protocol, Src Port: 3746, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x0003
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ mit.edu: type NS, class IN
    [Response In: 493]
```

493번

```
▸ User Datagram Protocol, Src Port: 53, Dst Port: 3746
▾ Domain Name System (response)
    Transaction ID: 0x0003
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 0
    Additional RRs: 3
  ▾ Queries
    ▸ mit.edu: type NS, class IN
  ▾ Answers
    ▸ mit.edu: type NS, class IN, ns bitsy.mit.edu
    ▸ mit.edu: type NS, class IN, ns strawb.mit.edu
    ▸ mit.edu: type NS, class IN, ns w20ns.mit.edu
  ▾ Additional records
    ▸ bitsy.mit.edu: type A, class IN, addr 18.72.0.3
    ▸ strawb.mit.edu: type A, class IN, addr 18.71.0.151
    ▸ w20ns.mit.edu: type A, class IN, addr 18.70.0.160
    [Request In: 492]
    [Time: 0.000361000 seconds]
```

Name servers of MIT

IPs of MIT

489번

```
▸ User Datagram Protocol, Src Port: 53, Dst Port: 3744
▾ Domain Name System (response)
    Transaction ID: 0x0001
  ▸ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ 22.29.238.128.in-addr.arpa: type PTR, class IN
  ▾ Answers
    ▸ 22.29.238.128.in-addr.arpa: type PTR, class IN, dns-prime.poly.edu
    [Request In: 488]
    [Time: 0.000367000 seconds]
```

491번

```
▸ User Datagram Protocol, Src Port: 53, Dst Port: 3745
▾ Domain Name System (response)
    Transaction ID: 0x0002
  ▸ Flags: 0x8583 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▾ Queries
    ▸ mit.edu.poly.edu: type NS, class IN
  ▾ Authoritative nameservers
    ▸ poly.edu: type SOA, class IN, mname dns-prime.poly.edu
    [Request In: 490]
    [Time: 0.000344000 seconds]
```