You’ll learn the most by doing the two parts of this lab on your own computer first – looking at the DNS messages and UDP datagrams sent and received in response to what you do. You can then complete the Moodle quiz for this Lab, which will ask you to answer the same questions, but using a pcap packet trace file.

1. The DNS

As described in Section 2.4 of the text1, 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.

1.1 Accessing the DNS via the command line

For this lab, it’ll be helpful to know a couple of DNS-related commands that you can issue on the command line of your computer (MAC, PC, Linux).

---

The `nslookup` command (entered on your terminal command line), allows your host 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. To do this, `nslookup` sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result. Let’s try out `nslookup`.

```
[kurose@macbook-pro-6 ~] sudo killall -HUP mDNSResponder
[kurose@macbook-pro-6 ~] nslookup gaia.cs.umass.edu
Server:    75.75.75.75
Address:  75.75.75.76#53

Non-authoritative answer:
Name: gaia.cs.umass.edu
Address: 128.119.246.12

[kurose@macbook-pro-6 ~] nslookup -type=NS cs.umass.edu
Server:    75.75.75.75
Address:  75.75.75.76#53

Non-authoritative answer:
cs.umass.edu nameserver = ns2.umass.edu.
cs.umass.edu nameserver = ns3.umass.edu.
cs.umass.edu nameserver = ns1.umass.edu.
cs.umass.edu nameserver = unix1.cs.umass.edu.

Authoritative answers can be found from:
ns3.umass.edu internet address = 128.103.36.68
ns1.umass.edu internet address = 128.119.10.27
ns2.umass.edu internet address = 128.119.10.28

[kurose@macbook-pro-6 ~]
```

Figure 1: command line interactions with the DNS

Let’s look at the DNS-related command line interactions shown in Figure 1:

- We start by flushing out any entries in my computer’s host-local DNS cache. That’s what’s done by the yellow-highlighted `sudo killall -HUP mDNSResponder` command. You can learn how to clear any device DNS cache [here](#).
- The first `nslookup` command, `nslookup gaia.cs.umass.edu` is saying “please send me the IP address for the host gaia.cs.umass.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 gaia.cs.umass.edu. Although the response came from the local DNS within Comcast\(^2\), 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.

\(^2\) How did I know my local DNS server was in Comcast? I did a `whois 75.75.75.75` on the command line, and learned that the address range 75.64.0.0 to 75.75.191.255 is owned by Comcast Cable Communications, since 2006.
• In the second *nslookup* command: `% nslookup -type=NS cs.umass.edu` we have provided the option “-type=NS” and the domain “cs.umass.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* domain name servers for cs.umass.edu”. (When the –type option is not used, *nslookup* defaults to querying for type A records.) The answer, displayed in Figure 1, first indicates my local DNS server 75.75.75.75 is providing a nonauthoritative answer (since it is not the authoritative name server for cs.umass.edu). There are four nameservers shown for cs.umass.edu. Three of them, with addresses shown are authoritative name servers for cs.umass.edu.

Note that different ISPs may have different policies about returning authoritative name server information, so YMMV - you may or may not get information back from this *nslookup* command.

Let’s now try this on your computer. Do the following:

• Clear any DNS cache information on the command line
  o on a Mac, type `sudo killall -HUP mDNSResponder` and hit enter.
  o or on a PC, type `ipconfig /flushdns` and hit enter.
  o or on other devices, check the link [here](#).
• Start up your browser and clear your browser’s history.
• Start up Wireshark, and begin packet capture. Well now want to capture some DNS messages, which conveniently enough are carried in UDP datagrams – so we can do a DNS and a UDP Wireshark lab at once!
• On the terminal command line for your computer, type the following:
  o `nslookup nyu.edu`
  o `nslookup -type=NS nyu.edu`

Evidently, we’re interested in the IP address of nyu.edu as well as the identify of NYU’s authoritative name server.

• Then go to your web browser and display the web page at `http://theonion.com`
• You can now stop Wireshark, as we’ve got a good number of DNS packets to take a look at.

Your Wireshark screen should look similar to as shown in Figure 2. Note that we have used the “dns” (no quotes) filter in the filter box just under the (wire)shark fin, so we can just display and focus on DNS messages. If you don’t do this your DNS messages will be lost in a sea of other frames, datagrams, segments and messages.
Let first focus on the DNS message (and its reply) that was sent to determine the IP address of nyu.edu. Answer the following questions (note: to hand in this assignment, you will later answer these questions via Moodle, but using a pcap file trace rather than the trace you have captured live here. But you should answer these questions for yourself here, using your trace).

1. Locate the DNS query and response messages that are used to resolve the hostname nyu.edu. 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?
4. Examine the DNS query message. What “Type” of DNS query is it? Does the query message itself contain any “answers”?
5. Examine the DNS response message. What type of record is being returned?
6. What is the IP address associated with nyu.edu?
7. How long is the local DNS allowed to cache this record before timing it out of its local cache?

Now let’s focus on the DNS message that was sent to determine the NYU authoritative name server(s). Answer the following questions:
8. How many authoritative name servers are returned in the response?
9. How long may the local name server cache this record?
10. What additional information is contained in the “Additional records” that are returned.

Finally, let’s look at the DNS query that was name for theonion.com.
11. How many IP addresses are returned for thenion.com?

Since we’ve got some UDP segments on our screen, let also take a look at the UDP protocol in this lab. Because UDP is simple and sweet, we’ll be able to cover it pretty quickly in this lab. Don’t worry - if you’ve got something to do in 30 minutes, don’t west sweat it, you should be able to finish this lab with ample time to spare.

12. Let’s focus just on the single UDP segment that contains (as its payload) the DNS query to find out the IP address (A record) of nyu.edu.
13. What are the source and destination port numbers for the is UDP segment?
14. How many bytes long is this UDP segment (header plus payload).
15. This question is really a network-layer question. Using Wireshark, look into the header fields of the IP datagram containing this UDP segment. Find the “upper layer protocol field”. The network layer will typically demultiplex up a segment to wither TCP (which has a protocol number of X) or UDP. What is the protocol number for UDP, contained in the IP datagram header?

That’s it! (See that wasn’t hard!).

What to hand in

Hopefully, you’ve done all of the steps above, and pondered the questions asked. To complete and “hand-in” this lab assignment, you need to go to the class Moodle site, where you’ll the find, and should complete, the DNS/UDP Wireshark Lab quiz.

In that Moodle quiz, you’ll be asked to load a trace file (http://gaia.cs.umass.edu/cs453_fall_2020/files/DNS_UDP_Wireshark_Lab_2020.pcapng) into Wireshark that was taken while the GETs in this Lab writeup were made, and you’ll answer the same questions as in this Lab writeup based on the loaded trace file (i.e., not using the packet capture you just performed on your own computer).