Programming Assignment 1: All about DNS
CSE 310, Spring 2020
Instructor: Aruna Balasubramanian
Due date: 02/24/2020, 9.00pm

Almost everything on the Internet involves DNS resolution. If DNS is slow, the performance of your application is going to suffer. The goal of this homework is write your own DNS resolver, and compare its performance with other existing DNS resolvers.

Part A (70 points)

You will be implementing a DNS resolver. The resolver takes as input a domain name. Your resolver resolves this query by first contacting the root server, the top-level domain, all the way until the authoritative name server.

You can assume that you have access to a library that can resolve a single iterative DNS query. The set of libraries that you may use are given in the Appendix. The libraries also perform complete DNS resolution, but you are *not* allowed to use that.

1. You can access the IP address of the root servers from https://www.iana.org/domains/root/servers.

2. Build a "dig"-like tool called "mydig". The mydig tool takes as inpt the name of the domain you want to resolve. You should resolve the "A" record for the query.

   When run as "mydig www.cnn.com", your tool should display the following

```
QUESTION SECTION:
www.cnn.com.          IN   A

ANSWER SECTION:
www.cnn.com.      262  IN   A    151.101.209.67

Query time: How much time it took to resolve the query
WHEN: Data and time of request
```

Along with the code, you need to submit an output file called "mydig_output", that contains the expected output for running your mydig program. Please specify the input to your program.

In some cases, you will not be able to resolve the query to the complete IP address, but only get a "CNAME". In this case, simply output the CNAME in the answer section.

You will use two APIs to create a DNS request to each individual server. The first is a "make_query" In python the API is dns.message.make_query. The second is to actually send this query over UDP. In python, the API is dns.query.udp. The query you will send to each server.

Bonus (5%): In some cases, you may need additional resolution. For example, google.co.jp will often not resolve to an IP address in one pass and will resolve to a CNAME. For bonus points, you can handle this case.

PART B (30 points)

Your next task is to measure the performance of your DNS resolver from Part A. Pick 10 out of the top 25 Websites from alexa.com (http://www.alexa.com/topsites.)

Experiment 1: Run your DNS resolver on each website 10 times and find the average time, 25th percentile, and 75th percentile to resolve the DNS for each of the 25 websites.

Experiment 2: Now use your local DNS resolver and repeat the experiment(call this Local DNS). Find the average time to resolve the address for the 10 websites.

Experiment 3: Change the DNS resolver to Google's public DNS (The IP address of this public DNS is often 8.8.8.8, or 8.8.4.4, but you need to verify). Repeat the experiment one more time and call this result "Google DNS"

You can use the dig command for experiments 2 and 3.

For each of the 10 Website, plot the average, 25th percentile and 75th percentile values over the 10 runs and draw a graph. The x axis is the website, named 1 to 10. The y axis is the time taken to resolve the query. Each point will have three bars corresponding to the three experiments.

Explain your result as best as you can.

Submission instruction

You need to submit your homework in a single zip file as follows:

- The zip file and (the root folder inside) should be named using your last name, first name, and the assignment name, all separated by a dash ('-')
  e.g. lastname-firstname-assignment2.zip
- The zip file should contain your code corresponding to Part A and your answer in Part B. Please be sure to put code in the root folder rather than in separate folders. Provide sufficient comments to your code.
- Include the expected output file "mydig_output" as specified in Part A.
- You should provide a README file describing:
  - External libraries used.
  - Instructions on how to run your programs.

All README and your answer to Part B should be in pdf.

APPENDIX

A. You may write your programs in the following languages: python, Java, and C/C++. For each language, you can use the corresponding DNS library to perform the single DNS resolution. The recommended libraries are:
python —> dnspython
C/C++ —> DSNQuery

If you choose to write your program using any other language or using a different DNS library, you should get permission from the instructor first. Remember that you cannot use these libraries to perform the entire resolution.

A note on running your code using the Universities network

If you are not able to connect to the root servers on campus, this is because the campus network blocks access to the root servers (but the CS department network does not do so). You can work around this by using a VPN. Please note that **TunnelBear VPN does not work. It resolves the complete query for you** even if you are trying to do it iteratively.