Homework 1: All about DNS
CSE 534, Spring 2016
Instructor: Aruna Balasubramanian
Due date: 2/17/2016, 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. As a fun addition, you also get to write your own "dig" tool.

Part A (40 points)

First, you will be implementing a DNS resolver. In response to an input query, the resolver will first contact the root server, then the top-level domains, all the way down to the corresponding name server to resolve the DNS query.

You can assume that you have access to a library that can resolve individual DNS queries. The set of libraries that you may use are given in the Appendix. Of course, 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. Your server program must use this list to find a working root server. You may copy the server names in your local directory. To choose which root server to access, your code should use the round robin scheduling algorithm. In addition, if the request is not answered, move on to the next server in the list.
2. You should be able to run your program as "./mydnsresolver <domain name>". The output should be the IP address corresponding to the domain name.

PART B (20 points)

Build a "dig"-like tool called "mydig" using the DNS resolver from Part A. The mydig tool takes as input: (1) name and (2) type. The type represents the DNS query type and name depends on the query type.

Your tool should work for the following types: "A", "NS", "MX".

1. When run as "./mydig <name> <type>", your tool should display the results similar to the results from the dig tool *(only the answer section, not the additional section).*
2. Be sure to take care of cases when a resolution is not possible.
3. Along with the code, you need to submit an expected output file called "mydig_output.txt", that contains the expected output for running your mydig program for the 3 types mentioned above. Please specify the input to your program.

PART C (30 points)

Your next task is to measure the performance of your DNS resolver. Take 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 to resolve the DNS over the 25 websites.

Experiment 2: Now use your local DNS server and repeat the experiment. Find the average time to resolve the address for the 25 websites, computed as the average over 10 runs.

Experiment 3: Change the DNS resolver to Google's public DNS (IP address 4.4.4.4, or 8.8.8.8). Repeat the experiment one more time.

Draw a graph that shows the Cumulative Distribution Function (CDF) of the DNS resolution time for Experiment 1, Experiment 2, and Experiment 3. Your graph should have the correct x axis, y axis, units, and labels, and the correct legends. If you do not know how to draw a CDF, look it up.

Explain your result as best as you can.

PART D (10 points)

Draw the DNS packet format (as discussed in class) for the request and response packet for the query with name= www.cs.stonybrook.edu, and type="A". You should refer to the DNS lectures with the slide titled "DNS packet format" for this part. You should fill in the details of the resource records. You can use your DNS resolver to find the response. If there are fields that you cannot fill (such as flags), you can just leave them blank.

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 homework name, all separated by a dash ('-')
  e.g. lastname-firstname-HW1.zip
- The zip file should contain your DNS resolver code, the mydig code, and the code corresponding to Part C of this homework.
- Include the expected output file "mydig_output.txt".
- Include the text file containing the answers for Part C and Part D.
- You should provide a README.txt file describing:
  - which external libraries you used.
  - the high level view of your design.
  - how to run your programs.

APPENDIX

You may write your programs in the following languages: python, ruby, Java, C/C++, or Perl. For each language, you can use the corresponding DNS library to perform the single DNS resolution. The recommended libraries are:

python —> dnspython
ruby —> Resolv
Java —> dnsjava (org.xbill.DNS)
C/C++ —> DSNQuery
PERL —> Net::DNS::Resolver

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 recursive DNS resolution

Often, the type of DNS resolver you are writing is also called a recursive DNS resolver. It takes in a query, iteratively contacts each DNS name server in turn to resolve your query.

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.