Project 1: Recursive DNS client and DNS servers

==========

Overview

The goal of this project is to implement a simplified DNS system consisting of a client program and two server programs: RS (a simplified root DNS server) and TS (a simplified top-level DNS server). In project 0 (your first HW) , you have already seen a client-server program with one socket each in the client and the server. In this project, you will extend that implementation to have two sockets in the client program. One socket will be used to communicate with RS and the other with TS.

The RS and TS programs each maintain a DNS_table consisting of three fields:

- Hostname

- IP address

- Flag (A or NS)

You need to choose the appropriate data structure to store the values for each entry. The client always connects first to RS, sending the queried hostname as a string. The RS program does a look up in its DNS_table, and if there is a match, sends the entry as a string

Hostname IPaddress A

If there is no match, RS sends the string

TSHostname - NS

where TShostname is the name of the machine on which the TS program is running.

If the client receives a string with "A" field, it outputs the received string as is. On the other hand, if the client receives a string with "NS" field, it uses the TSHostname part of the received string to determine the IP address of the machine running the TS program and connects to the TS program using a second socket. The client then sends the queried hostname as a string to TS. The TS program looks up the hostname in its DNS_table, and if there is a match, sends the entry as a string

Hostname IP address A to the client. Otherwise, it sends an error string

Hostname - Error:HOST NOT FOUND

In the TS outputs above, Hostname is the queried hostname. The client outputs

the string received from TS as is.

Note that all DNS lookups are case-insensitive. If there is a hit in the local

DNS table, the server programs must respond with the version of the string that

is in their local DNS table.

How will we test your programs?

As part of your submission, you will turn in three programs: rs.py, ts.py, and client.py, and one README file (more on this below). We will be running the three programs on the ilab machines with Python 2.7.

Please do not assume that all programs will run on the same machine or that all connections are made to the local host. We reserve the right to test your programs with local and remote socket connections, for example with client.py, ts.py, and rs.py each running on a different machine. You are welcome to simplify the initial development and debugging of your project, and get off the ground by running all programs on one machine first. However, you must eventually ensure that the programs can work across multiple machines.

The programs must work with the following command lines:

python ts.py tsListenPort

python rs.py rsListenPort

python client.py rsHostname rsListenPort tsListenPort

Here tsListenPort (rsListenPort) is the port on which TS (RS) listens for requests and rsHostname is the hostname of the machine running the RS program. The hostname strings to be queried will be given one per line in a file

PROJI-HNS.txt.

The entries of the local DNS tables, one each for RS and TS, will be strings with fields separated by spaces. There will be one entry per line. You can see the format in PROJI-DNSRS.txt and PROJI-DNSTS.txt. Your server programs should populate the DNS table by reading the entries from the corresponding files.Your client program should output the results to a file RESOLVED.txt, with one line per result.

See the samples attached in this folder.

We will test your programs by running them with the hostnames and tables in the attached input files (*.txt) as well as with new hostnames and table configurations. You will be graded based on the outputs in RESOLVED.txt. Your programs should not crash on correct inputs.

README file

In addition to your programs, you must also submit a README file with clearly

dilenated sections for the following.

0. Please write down the full names and netids of both your team members.

1. Briefly discuss how you implemented your recursive client functionality.

2. Are there known issues or functions that aren't working currently in your attached code? If so, explain.

3. What problems did you face developing code for this project?

4. What did you learn by working on this project?

Submission

Turn in your project on Sakai assignments. Only one team member needs to submit. Please upload a single zip file consisting of client.py, rs.py, ts.py, and README.

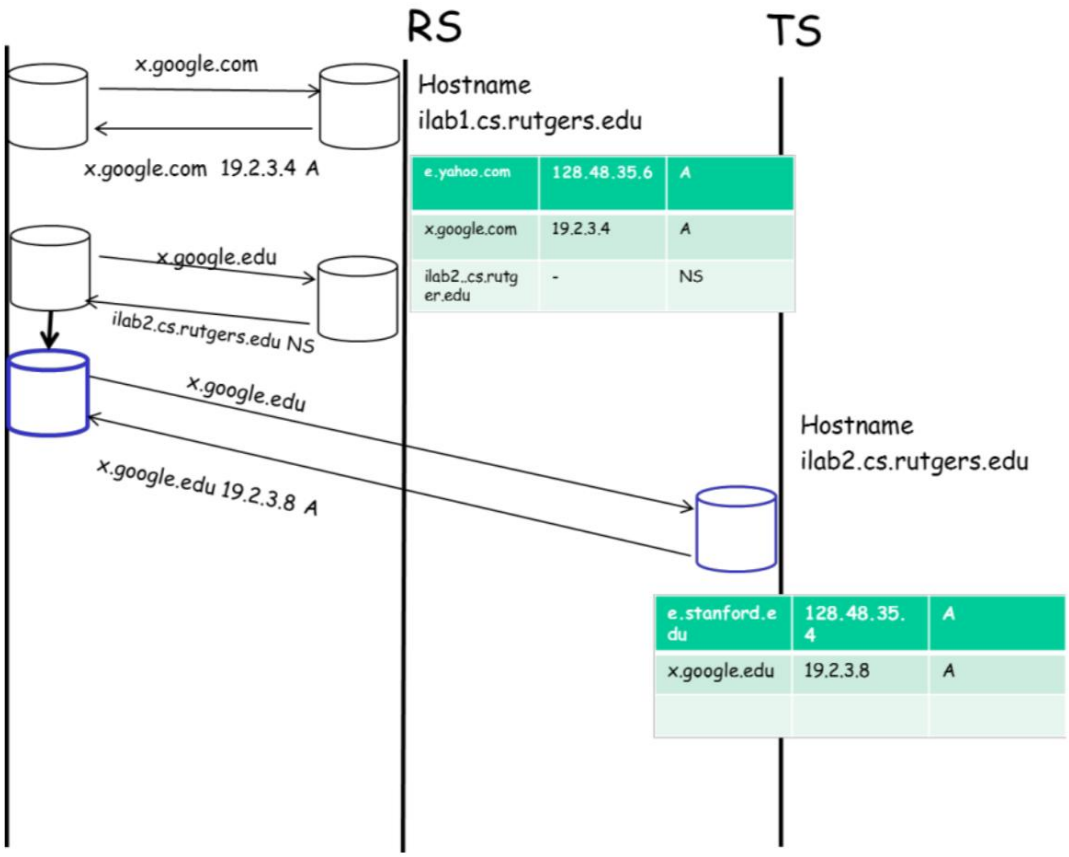
Some tips

Run your programs by first starting the TS program, the RS program, and then the client program. A brief sketch of the interaction among the programs is attached in this folder.

DNS lookups are case-insensitive.

It is okay to assume that each DNS entry or host name is smaller than 200 characters.

START EARLY to allow plenty of time for questions on Piazza should you run into difficulties.


RS
TS
x.google.com
Hostname
ilab1.cs.rutgers.edu
x.google.com 19.2.3.4 A
e.yahoo.com
128.48.35.6
A
x.google.com
19.2.3.4
A
ilab2..cs.rutger.edu
-
NS
x.google.edu
ilab2.cs.rutgers.edu NS
x.google.edu
Hostname
ilab2.cs.rutgers.edu
x.google.edu 19.2.3.8 A
e.stanford.edu
128.48.35.4
A
x.google.edu
19.2.3.8
A