NEW YORK UNIVERSITY

CS-GY 6843

Python Programming: DNS Part 1

Part 1: What to Hand In

- Your code must be saved in a program file named **DNSClient.py**
- Submit the code to GradeScope **using your GitHub repository specifically created for this assignment.**
- Assignment submissions with **multiple files in the same GitHub repository will not be accepted**. Only one file per assignment is to be submitted.
- Multiple submission attempts are allowed up until the deadline or due date.

IMPORTANT
❗ Use the skeleton code as a starting point, but make sure you understand what each part of it does.
❗ **Don't modify outside of the designated areas**.

Part 2: Assignment Expectations and Instructions

Objective: In this assignment, you will:

- Learn about the Domain Name System (DNS) and how it works.
- Learn how to write a simple DNS test client in Python to verify your DNS server code in Part 2.

✅ This first part is not meant to be challenging,it helps you get a test client up and running while you work on the DNS Server itself in Part 2.

Introduction to test clients and DNS

A test client is a program that sends requests to a server, and the 'test part' is us verifying that the responses are correct. In the context of a DNS server, a test client sends DNS queries to the server and checks that the responses contain the expected IP addresses. In this case, we will query our local DNS server and a Public DNS server to see if the responses are the same!

The provided code is an example of a simple test client written in Python using the dnspython library. This library provides both high- and low-level access to DNS. The high-level classes perform queries for data of a given name, type, and class, and return an answer set.

The code defines two functions: query_local_dns_server and query_dns_server. The first function sends a DNS query to a local DNS server running on the same machine as the test client. The second function sends a DNS query to a public DNS server.

Both functions take as input a domain name and return the IP address associated with that domain name according to the respective DNS server. There are a few extra functions there for you to use as you see fit.

Assignment Instructions:

- **Writing a Test Client for a DNS Server in Python**
  In this part of the assignment, you will write your own test client for a DNS server in Python. You can use the provided skeleton code as a starting point, you are to fill in the blanks denoted by '?'. Do not edit outside of the designated blocks.

✅ **The skeleton code can be found at this link:**
https://github.com/NYUCyberFellows-CSGY6843/DNSClient

💡 **Hint**: Many sections have already been handled in the skeleton code. Just complete the missing parts.

- **Steps for writing the test client**

  1. Import the necessary modules, including dns.resolver from the dnspython library.
  2. Provide the needed variable values for the local_host (what IP is reserved for local servers on the local machine?), a real external DNS server, and question_type.
  3. Define two functions: one for querying your local DNS server and one for querying a public DNS server. Both functions should take as input a domain name and return the IP address associated with that domain name according to the respective DNS server.
  4. In each function, create an instance of dns.resolver.Resolver and set its nameservers attribute to the IP address of the respective DNS server.
  5. Use the resolve method of the Resolver instance to send a DNS query for the given domain name and record type (e.g., A for IPv4 addresses).
  6. Extract the IP address from the response using the to_text method of the first answer record.
  7. Return the IP address from each function.

8. Write additional code to call your functions with different domain names and verify that they return the expected IP addresses.

9. Test the results from your local DNS server against your chosen public DNS server, by writing a comparison function that returns a Boolean.

💡 **Hint:** See the compare_dns_servers() method.