Assignment 2

DNS Resolution - Iterative and Recursive Lookup

Course: CS425: Computer Networks

Instructor: Adithya Vadapalli

TAs Incharge: Hrushikesh and Lokesh

Submission Deadline: 02.03.2025, EOD

Objective

This assignment aims to implement a DNS resolution system that supports both iterative and recursive lookups. Students will gain hands-on experience with DNS querying, network programming, and protocol understanding.

Background

DNS (Domain Name System) is a hierarchical system that translates human-readable domain names into IP addresses. The resolution process can be performed in two ways:

- **Iterative Resolution:** The resolver queries root servers, then top-level domain (TLD) servers, and finally authoritative servers to get the response.
- **Recursive Resolution:** The resolver delegates the entire resolution process to an external resolver, which retrieves the final result.

In this assignment, you will implement both iterative and recursive DNS lookup mechanisms using Python and the `dnspython` library.

Instructions

1. Clone the repository from: `https://github.com/privacy-iitk/cs425-2025.git`
2. Go to the `Homeworks/A2` directory as `cd cs425-2025/Homeworks/A2`
3. You will see `dnsresolver.py` and `README.md`
4. Your goal is to:
   - Complete the TODOs in the assignment.
   - provide detailed comments for the code that you wrote
5. This assignment can be solved in a group of at most three students.
6. Use only Piazza to ask for help in the assignment.

Tasks

Task 1: Iterative DNS Resolution

1. Implement a function to send a DNS query to a given nameserver.
2. Start resolution from the root DNS servers.
3. Extract and resolve the next nameservers at each stage (Root, TLD, and Authoritative servers).
4. Print the resolved IP address if found.

Task 2: Recursive DNS Resolution

1. Use a DNS resolver to perform recursive queries.
2. Fetch and display the IP address of a given domain.
3. Handle errors gracefully (e.g., non-existent domains, timeouts).

Deliverables

Students must submit the following:

- Source code implementing both iterative and recursive DNS lookup.
- A README file with instructions on how to run the code.

Implementation Guidelines

- Use Python and the `dnspython` library for DNS queries.
- Handle errors such as timeouts, incorrect domain names, and unreachable servers.
- Your script should accept command-line arguments.

```
python3 dns_resolver.py iterative example.com
python3 dns_resolver.py recursive example.com
```

An example output could look like:

```
python3 dns_sol.py iterative google.com
[Iterative DNS Lookup] Resolving google.com
[DEBUG] Querying ROOT server (198.41.0.4) - SUCCESS
Extracted NS hostname: l.gtld-servers.net.
Extracted NS hostname: j.gtld-servers.net.
Extracted NS hostname: h.gtld-servers.net.
Extracted NS hostname: d.gtld-servers.net.
Extracted NS hostname: b.gtld-servers.net.
Extracted NS hostname: f.gtld-servers.net.
Extracted NS hostname: k.gtld-servers.net.
Extracted NS hostname: m.gtld-servers.net.
Extracted NS hostname: i.gtld-servers.net.
Extracted NS hostname: g.gtld-servers.net.
Extracted NS hostname: a.gtld-servers.net.
Extracted NS hostname: c.gtld-servers.net.
Extracted NS hostname: e.gtld-servers.net.
Resolved l.gtld-servers.net. to 192.41.162.30
Resolved j.gtld-servers.net. to 192.48.79.30
Resolved h.gtld-servers.net. to 192.54.112.30
Resolved d.gtld-servers.net. to 192.31.80.30
Resolved b.gtld-servers.net. to 192.33.14.30
Resolved f.gtld-servers.net. to 192.35.51.30
Resolved k.gtld-servers.net. to 192.52.178.30
Resolved m.gtld-servers.net. to 192.55.83.30
Resolved i.gtld-servers.net. to 192.43.172.30
Resolved g.gtld-servers.net. to 192.42.93.30
Resolved a.gtld-servers.net. to 192.5.6.30
Resolved c.gtld-servers.net. to 192.26.92.30
Resolved e.gtld-servers.net. to 192.12.94.30
[DEBUG] Querying TLD server (192.41.162.30) - SUCCESS
Extracted NS hostname: ns2.google.com.
Extracted NS hostname: ns1.google.com.
Extracted NS hostname: ns3.google.com.
Extracted NS hostname: ns4.google.com.
Resolved ns2.google.com. to 216.239.34.10
Resolved ns1.google.com. to 216.239.32.10
Resolved ns3.google.com. to 216.239.36.10
Resolved ns4.google.com. to 216.239.38.10
[DEBUG] Querying AUTH server (216.239.34.10) - SUCCESS
[SUCCESS] google.com -> 142.250.194.78
Time taken: 0.597 seconds
```

```
python3 dns_sol.py recursive google.com
[Recursive DNS Lookup] Resolving google.com
[SUCCESS] google.com -> ns4.google.com.
[SUCCESS] google.com -> ns3.google.com.
[SUCCESS] google.com -> ns2.google.com.
[SUCCESS] google.com -> ns1.google.com.
[SUCCESS] google.com -> 172.217.167.206
Time taken: 0.014 seconds
```

Submission Instructions

- Submit a zip file containing the source code README
- The filename should be `A2Rollnumberofmember1Rollnumberofmember2Rollnumberofmember3`
- Upload your submission to HelloIITK before the deadline. Only one team member should submit the assignment.

Grading Rubric

- **Correctness (60%)**: The server works as expected and supports all required features.
- **Code Quality (15%)**: Comments in the code.
- **Documentation (25%)**: Clear instructions and explanation in the README.

Note: For any clarifications, post a message on Piazza. Ensure all submissions are made before the deadline. Late submissions will incur a penalty as per the course policy.

Instructor: **Adithya Vadapalli** TA Incharge: **Hrushikesh and Lokesh**