University of Dhaka

Department of Computer Science and Engineering

CSE-3111 : Computer Networking Lab

Lab Report 4: Distributed Database Management, Implementation of Iterative, and Recursive Queries of DNS Records.

Submitted By:

Name : Abdullah Al Mahmud

Roll No : 15

Name : Zisan Mahmud

Roll No : 23

Submitted On:

February 15, 2024

Submitted To:

Dr. Md. Abdur Razzaque

Dr. Md Mamunur Rashid

Dr. Muhammad Ibrahim

Mr. Md. Redwan Ahmed Rizvee

1 Introduction

In this lab, our main goal is to observe how a distributed database system works and how DNS queries fetch data to clients. A distributed database is not limited to one system, it is spread over different sites that don't share physical components. It is required when a database needs to be accessed by various users globally. The DNS is a distributed database implemented in a hierarchy of DNS servers. A DNS query is a message that a client sends to the DNS server. It is send in two ways, iterative and recursive. We will implement this two types of queries in our lab and discover how they can be synergistically applied for effective network communication.

2 Objectives

The preliminary objective of this lab is to emulate the Domain Name Service (DNS) protocol and to understand the difference between iterative and recursive DNS resolution.. In this lab, we will:

- Design and implement local, root, top level domain and authoritative DNS server
- Design a client that will request IP address of his desired domain and the name-server hierarchy will use the DNS resolution to return the IP address of the corresponding domain to the client if the domain name is valid.
- Compare of the time required for resolving a request via iterative and recursive DNS resolution methods.

3 Theory

The DNS is a distributed database implemented in a hierarchy of DNS servers and an application-layer protocol that allows hosts to query the distributed database. DNS sues a large number of servers distributed around the world organized in a hierarchical manner. No single DNS server has all the mappings for all the hosts in the internet. Instead, the mappings are distributed across the DNS server. There are thee classes of DNS servers:

- **Root DNS servers:** A root server accepts a recursive resolver's query which includes a domain name, and the root nameserver responds by directing the recursive resolver to a TLD nameserver, based on the extension of that domain (.com, .net, .org, etc.).
- **Top-level domain(TLD) servers:** A TLD nameserver maintains information for all the domain names that share a common domain extension, such as .com, .net, or whatever comes after the last dot in a URL
- **Authoritative DNS servers:** The authoritative nameserver is usually the resolver's last step in the journey for an IP address. The authoritative nameserver contains information specific to the domain name it serves.

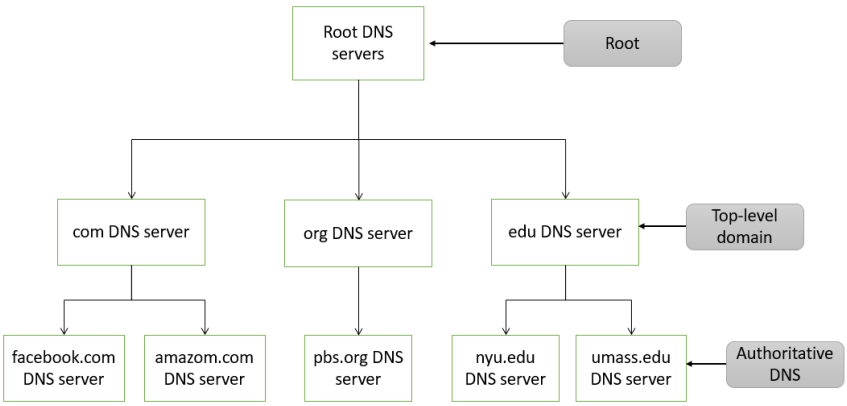


Figure 1: DNS Hierarchy

When we visit a domain such as google.com, our computer follows a series of steps to convert the human-readable web address into a machine-readable IP address. IP address retrieval process is as follows:

1. **Search in DNS cache:** When we ask our computer to resolve a hostname, the first place our computer looks is in its local DNS cache. If our computer doesn't already know the answer, it needs to perform a DNS query to find out.

2. **Request to ISP's DNS servers:** If the information is not stored locally, our computer queries our ISP's DNS servers.

3. **Request to root nameservers:** If those servers don't have the answer, they query the root nameservers. A nameserver is an always running computer that resolves queries about domain names, such as IP addresses.

4. **Request to Top Level Domain nameservers:** The root nameservers will look at the first part of our request, reading from right to left and direct our query to the Top-Level Domain (TLD) nameservers for .com (for google.com). Each TLD, such as .com, .org, and .bd, have their own set of nameservers, which act like a receptionist for each TLD.

5. **Request to authoritative DNS servers:** The TLD nameservers review the next part of our request and direct our query to the nameservers responsible for this specific domain. These authoritative nameservers are responsible for knowing all the information about a specific domain, which are stored in DNS records.

6. **Retrieve the record:** ISP's DNS server retrieves the record for google.com from the authoritative nameservers and stores the record in its local cache. If anyone else requests the host record for google.com, the ISP's servers will already have the answer and will not need to go through the lookup process again. All records have an expiration time. After a while, the server will need to ask for a new copy of the record to make sure the information doesn't become out-of-date.

7. **Receive the answer:** Armed with the answer, ISP's server returns the record back to our computer. Computer stores the record in its cache, reads the IP address from the record, then passes this information to browser. The browser then opens a connection to the webserver and receives the website.

The clients request is processed in two methods. These methods are known as DNS queries. They are:

1. **Iterative Query:** In an iterative query the client send message to the local DNS server. The local server forward the message to the root server. The root server checks for the IP address in it database, if the IP address is found, then root returns the address to the local server. If the address is not found, then local server forward the query to the TLD and finally to the the authoritative domain server.

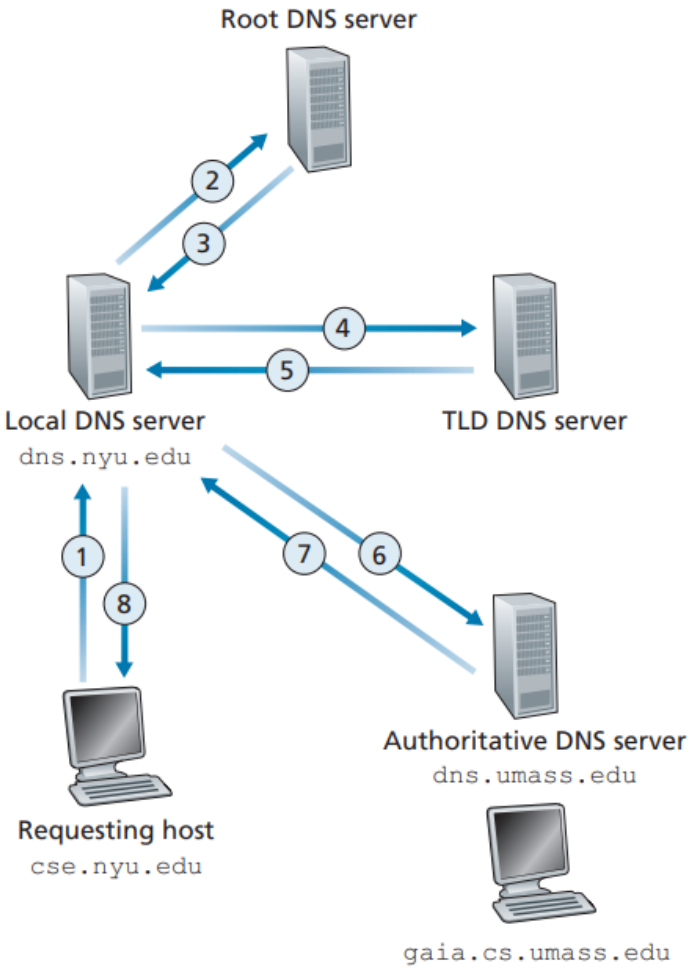


Figure 2: Iterative queries in DNS

2. **Recursive Query:** A recursive query tells the querying DNS server or resolve that it should provide an answer for the send questions. If the server itself serves as an authoritative server for this domain, it should respond with the normal response. If however, the server is not an authoritative server for this domain, it should start a recursive query process to resolve the domain name itself and then return the resulting records.

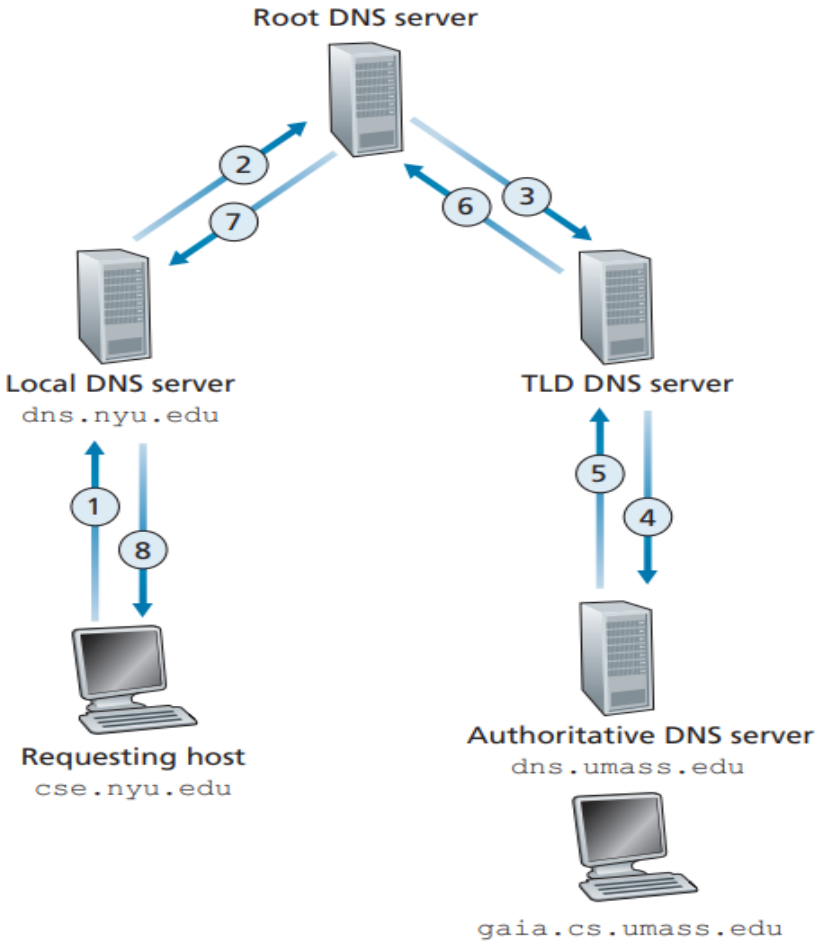


Figure 3: Recursive queries in DNS

The DNS servers that together implement the DNS distributed database store resource records (RRs), including RRs that provide hostname-to-IP address mappings. Each DNS reply message carries one or more resource records. A resource record is a four-tuple that contains the following fields:

(Name, Value, Type, TTL)

Here, TTL is the time to live value of the record which determines the time when a resource should be removed from a cache. The meaning of Name and Value depend on Type:

- Type = A: *Name* is a hostname and *Value* is the IP address for the hostname.
- Type = NS: *Name* is a domain (such as abc.com) and *Value* is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain
- Type = CNAME: *Value* is a canonical hostname for the alias hostname *Name*.
- Type = MX: *Value* is the canonical name of a mail server that has an alias hostname *Name*.

4 Methodology

1. We create a thread for each DNS server.
2. We create a tree of DNS servers with a root node.
3. Each server is provided with the reference to its parent server and child servers.
4. The server awaits for request from client
5. After receiving request, server looks for the requested domain in its current location and the child (if any). if not found, it will forward the request to the parent.
6. Parent does a similar search, if not found, it sends the request to its parents.
7. If the domain can't be found in any of the DNS servers, then the client will be prompted with an error message.

8. If the domain is found in any of the servers, they will return to the requesting client. (Iteratively/recursively)

5 Experimental result

5.1 Part 1: Setting up the DNS server

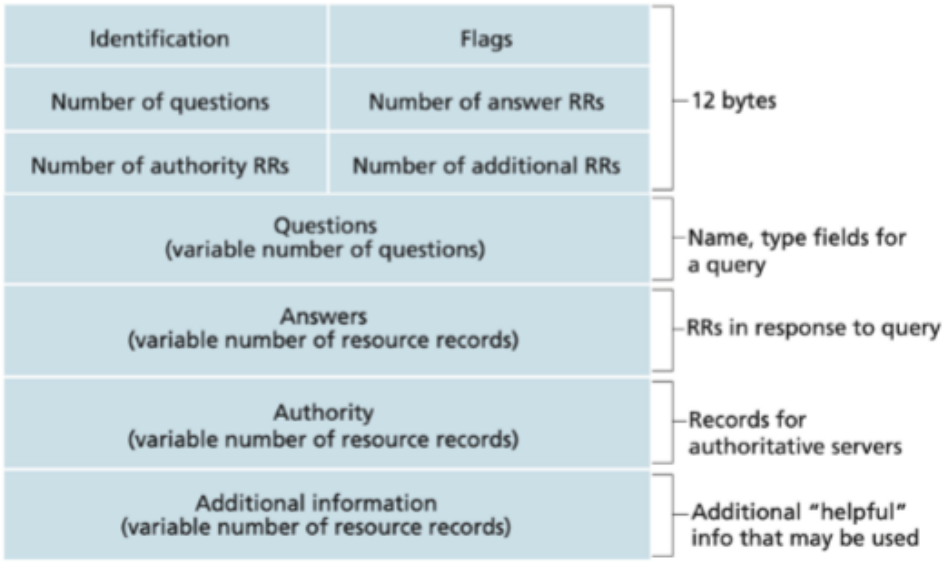
5.1.1 Server :

Server receives request from client and sends the headers and IP address of the requested domain.

```
araf@Arafs-MacBook-Air Task1 % python3 dns_server.py
[STARTING] Server is starting
[LISTENING] Server is listening on :9000.
[RECEIVED MESSAGE] cse.du.ac.bd. A from ('127.0.0.1', 63626).
[ACTIVE CONNECTIONS] 1
cse.du.ac.bd.
A
IP address found in the file
192.0.2.3
192.0.2.3
[RECEIVED MESSAGE] ns1.cse.du.ac.bd.  AAAA from ('127.0.0.1', 64682).
ns1.cse.du.ac.bd.
AAAA
[ACTIVE CONNECTIONS] 1
IP address found in the file
ns1.cse.du.ac.bd.
IP address found in the file
192.0.2.1
192.0.2.1
[RECEIVED MESSAGE] www.cse.du.ac.bd. CNAME from ('127.0.0.1', 59232).
www.cse.du.ac.bd.
CNAME
[ACTIVE CONNECTIONS] 1
IP address found in the file
cse.du.ac.bd.
IP address found in the file
192.0.2.3
192.0.2.3
[RECEIVED MESSAGE] cse.du.ac.bd. NS from ('127.0.0.1', 58687).
cse.du.ac.bd.
NS
[ACTIVE CONNECTIONS] 1
IP address found in the file
ns1.cse.du.ac.bd.
IP address found in the file
192.0.2.1
192.0.2.1
```

Figure 4: Part-1 Server

The response message consists of following sections:



5.1.2 Client :

Client request with the domain name for the IP address. After receiving the response, the header and IP will be available at he client side.

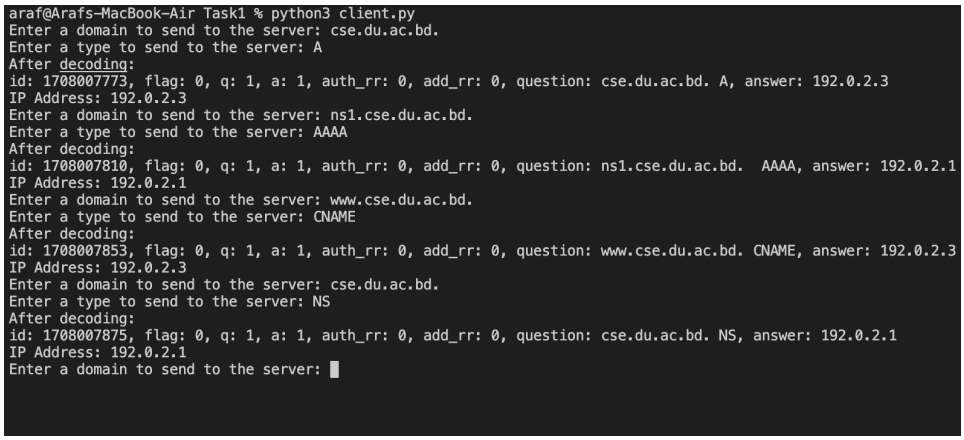


Figure 5: Part-1 Client

5.2 Part 2: Iterative DNS resolution

5.2.1 Server :

- **Local server:** Client first requests to the local server. If the local server does not have the requested IP in its cache, then it forward the request to root server.

```
araf@Arafs-MacBook-Air Iterative % python3 local_server.py
Local Server listening on localhost:8000
Local Server received query: cse.du.ac.bd from ('127.0.0.1', 63445)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 192.0.2.3 from 8003
IP address sent to ('127.0.0.1', 63445)
Local Server received query: google.com from ('127.0.0.1', 57033)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 142.250.193.110 from 8003
IP address sent to ('127.0.0.1', 57033)
Local Server received query: mail.cse.du.ac.bd from ('127.0.0.1', 63490)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 192.0.2.4 from 8003
IP address sent to ('127.0.0.1', 63490)
Local Server received query: mail.cse.du.ac.bd from ('127.0.0.1', 51274)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 192.0.2.4 from 8003
IP address sent to ('127.0.0.1', 51274)
Local Server received query: google.com from ('127.0.0.1', 51728)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 142.250.193.110 from 8003
IP address sent to ('127.0.0.1', 51728)
Local Server received query: ns1.cse.du.ac.bd from ('127.0.0.1', 64208)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 192.0.2.1 from 8003
IP address sent to ('127.0.0.1', 64208)
Local Server received query: ns2.cse.du.ac.bd from ('127.0.0.1', 54574)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 192.0.2.2 from 8003
IP address sent to ('127.0.0.1', 54574)
Local Server received query: cse.du.ac.bd from ('127.0.0.1', 64337)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: 192.0.2.3 from 8003
IP address sent to ('127.0.0.1', 64337)
Local Server received query: amazon.com from ('127.0.0.1', 55444)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: Not Found from 8003
Local Server sending response: Not Found
Local Server received query: yahoo.com from ('127.0.0.1', 60279)
Local Server received response: 8002
Local Server received response: 8003 from 8002
Local Server received response: Not Found from 8003
Local Server sending response: Not Found
```

Figure 6: Part-2 Local Server

- **Root Server:** The root server responds to the query with a referral to the TLD server for the TLD associated with the domain name being resolved.

```
araf@Arafs-MacBook-Air Iterative % python3 root_server.py
Root Server listening on localhost:8001
Root Server received query: bd
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: com
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: bd
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: bd
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: com
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: bd
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: bd
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: bd
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: com
Root Server sending response: 8002 to ('127.0.0.1', 8000)
Root Server received query: com
Root Server sending response: 8002 to ('127.0.0.1', 8000)
```

Figure 7: Part-2 Root Server

- **TLD Server:** It provide a referral to the authoritative name server for the domain name being resolved.

```
araf@Arafs-MacBook-Air Iterative % python3 tld_server.py
TLD Server listening on localhost:8002
TLD Server received query: cse.du.ac.bd
TLD Server sending response: 8003 to ('127.0.0.1', 8000)
TLD Server received query: google.com
TLD Server sending response: 8003 to ('127.0.0.1', 8000)
TLD Server received query: mail.cse.du.ac.bd
Record is not found redirecting to Authoritative Server 8003
TLD Server received query: mail.cse.du.ac.bd
Record is not found redirecting to Authoritative Server 8003
TLD Server received query: google.com
TLD Server sending response: 8003 to ('127.0.0.1', 8000)
TLD Server received query: ns1.cse.du.ac.bd
Record is not found redirecting to Authoritative Server 8003
TLD Server received query: ns2.cse.du.ac.bd
Record is not found redirecting to Authoritative Server 8003
TLD Server received query: cse.du.ac.bd
TLD Server sending response: 8003 to ('127.0.0.1', 8000)
TLD Server received query: amazon.com
Record is not found redirecting to Authoritative Server 8003
TLD Server received query: yahoo.com
Record is not found redirecting to Authoritative Server 8003
```

Figure 8: Part-2 Top-level Domain(TLD) Server

- **Authoritative Server:** This server holds the authoritative information for a specific domain.

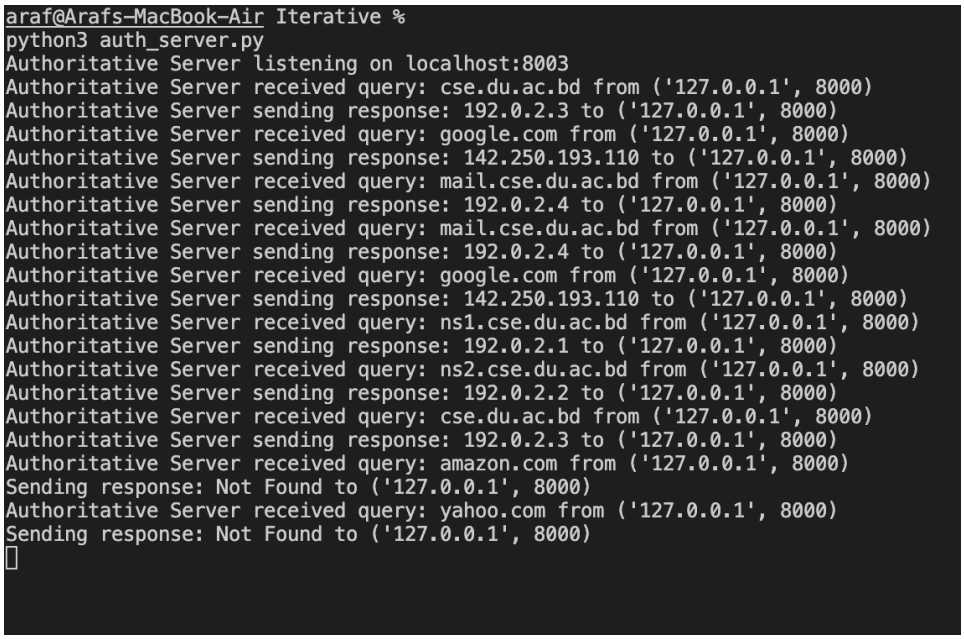


Figure 9: Part-2 Authoritative Server

5.2.2 Client :

The client sends request message to the local DNS server and get the Header and IP address of the requested domain name if it exists.

```
araf@Arafs-MacBook-Air Iterative % python3 client.py
Enter your DNS query: cse.du.ac.bd
IP address from server: 192.0.2.3
Time taken: 6.55 milliseconds
Enter your DNS query: google.com
IP address from server: 142.250.193.110
Time taken: 2.01 milliseconds
Enter your DNS query: mail.cse.du.ac.bd
IP address from server: 192.0.2.4
Time taken: 2.84 milliseconds
Enter your DNS query: mail.cse.du.ac.bd
IP address from server: 192.0.2.4
Time taken: 2.71 milliseconds
Enter your DNS query: google.com
IP address from server: 142.250.193.110
Time taken: 2.51 milliseconds
Enter your DNS query: ns1.cse.du.ac.bd
IP address from server: 192.0.2.1
Time taken: 2.73 milliseconds
Enter your DNS query: ns2.cse.du.ac.bd
IP address from server: 192.0.2.2
Time taken: 2.31 milliseconds
Enter your DNS query: cse.du.ac.bd
IP address from server: 192.0.2.3
Time taken: 2.03 milliseconds
Enter your DNS query: amazon.com
IP address from server: Not Found
Time taken: 2.63 milliseconds
Enter your DNS query: yahoo.com
IP address from server: Not Found
Time taken: 2.75 milliseconds
Enter your DNS query:
```

Figure 10: Part-2 Client

5.2.3 Advantages :

1. In an iterative query, request made by the local DNS server to the root, TLD and authoritative servers can be cached inside the local DNS. So it is faster.

2. The client is responsible for sending the query to successive servers, until the query is resolved by hitting a server that is authorized for the domain name (or until an error or time-out)2. This gives the client more control over the process.

5.2.4 Disadvantages :

1. Each iterative query results in network traffic between the client and each DNS server involved in the resolution process.

2. It has higher risk of errors, as there is a chance being incorrect referral.

3. For the security implications, iterative queries could potentially expose the client to certain risks. For example, since the client communicates directly with each DNS server involved in the lookup, it could be exposed to malicious servers.

5.3 Part 3: Recursive DNS resolution

5.3.1 Server :

- **Local server:** Client first requests to the local server. If the local server does not have the requested IP in its cache, then it forward the request to root server.

```
araf@Arafs-MacBook-Air Recursive % python3 local_server.py
Local Server listening on localhost:8000
Local Server received query: cse.du.ac.bd from ('127.0.0.1', 54038)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 54038)
Local Server received query: google.com from ('127.0.0.1', 62330)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 62330)
Local Server received query: mail.cse.du.ac.bd from ('127.0.0.1', 52583)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 52583)
Local Server received query: cse.du.ac.bd from ('127.0.0.1', 50798)
IP address found in local records
IP address sent to ('127.0.0.1', 50798)
Local Server received query: google.com from ('127.0.0.1', 50933)
IP address found in local records
IP address sent to ('127.0.0.1', 50933)
Local Server received query: ns1.cse.du.ac.bd from ('127.0.0.1', 65065)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 65065)
Local Server received query: amazon.com from ('127.0.0.1', 49432)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address Not Found
Local Server received query: google.com from ('127.0.0.1', 49613)
IP address found in local records
IP address sent to ('127.0.0.1', 49613)
Local Server received query: ns1.cse.du.ac.bd from ('127.0.0.1', 59288)
IP address found in local records
IP address sent to ('127.0.0.1', 59288)
Local Server received query: ns2.cse.du.ac.bd from ('127.0.0.1', 54606)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 54606)
```

Figure 11: Part-3 Local Server

- **Root Server:** The root server responds to the query with a referral to the TLD server for the TLD associated with the domain name being resolved.

```
araf@Arafs-MacBook-Air Recursive % python3 root_server.py
Root Server listening on localhost:8001
Root Server received query: cse.du.ac.bd
Record not found redirecting to TLD Server 8002
Root Server received response from TLD Server 8002
Root Server sent response to Local Server
Root Server received query: google.com
Record not found redirecting to TLD Server 8002
Root Server received response from TLD Server 8002
Root Server sent response to Local Server
Root Server received query: mail.cse.du.ac.bd
Record not found redirecting to TLD Server 8002
Root Server received response from TLD Server 8002
Root Server sent response to Local Server
Root Server received query: ns1.cse.du.ac.bd
Record not found redirecting to TLD Server 8002
Root Server received response from TLD Server 8002
Root Server sent response to Local Server
Root Server received query: amazon.com
Record not found redirecting to TLD Server 8002
Root Server received response from TLD Server 8002
Root Server sent response to Local Server
Root Server received query: ns2.cse.du.ac.bd
Record not found redirecting to TLD Server 8002
Root Server received response from TLD Server 8002
Root Server sent response to Local Server
```

Figure 12: Part-3 Root Server

- **TLD Server:** It provide a referral to the authoritative name server for the domain name being resolved.

```
araf@Arafs-MacBook-Air Recursive % python3 tld_server.py
TLD Server listening on localhost:8002
TLD Server received query: cse.du.ac.bd
Record not found redirecting to Authoritative Server 8003
TLD Server received response from Authoritative Server 8003
TLD Server sent response to Root Server
TLD Server received query: google.com
Record not found redirecting to Authoritative Server 8003
TLD Server received response from Authoritative Server 8003
TLD Server sent response to Root Server
TLD Server received query: mail.cse.du.ac.bd
Record not found redirecting to Authoritative Server 8003
TLD Server received response from Authoritative Server 8003
TLD Server sent response to Root Server
TLD Server received query: ns1.cse.du.ac.bd
Record not found redirecting to Authoritative Server 8003
TLD Server received response from Authoritative Server 8003
TLD Server sent response to Root Server
TLD Server received query: amazon.com
Record not found redirecting to Authoritative Server 8003
TLD Server received response from Authoritative Server 8003
TLD Server sent response to Root Server
TLD Server received query: ns2.cse.du.ac.bd
Record not found redirecting to Authoritative Server 8003
TLD Server received response from Authoritative Server 8003
TLD Server sent response to Root Server
```

Figure 13: Part-3 Top-level Domain(TLD) Server

- **Authoritative Server:** This server holds the authoritative information for a specific domain.

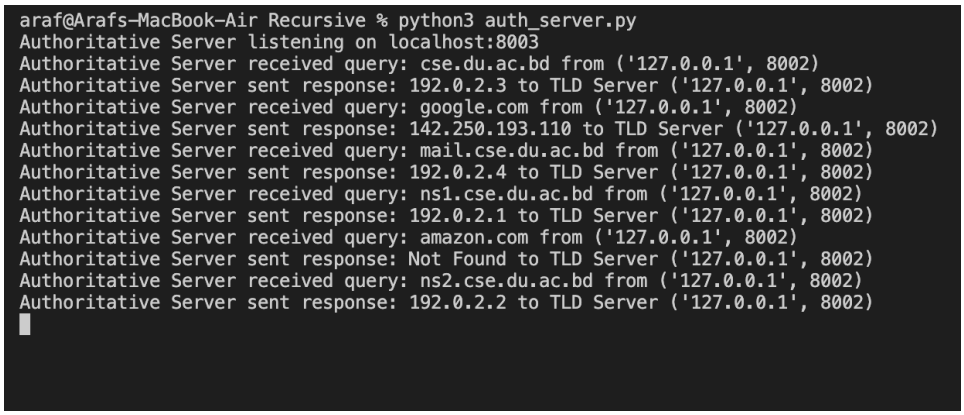


Figure 14: Part-3 Authoritative Server

The key difference between iterative and recursive is that a recursive DNS lookup is where one DNS server communicates with several other DNS servers to hunt down an IP address and return it to the client. This is in contrast to an iterative DNS query, where the client communicates directly with each DNS server involved in the lookup.

5.3.2 Client :

The client sends request message to the local DNS server and get the Header and IP address of the requested domain name if it exists.

```
araf@Arafs-MacBook-Air Recursive % python3 client.py
Enter your DNS query: cse.du.ac.bd
IP address from server: 192.0.2.3
Time taken: 5.83 milliseconds
Enter your DNS query: google.com
IP address from server: 142.250.193.110
Time taken: 2.56 milliseconds
Enter your DNS query: mail.cse.du.ac.bd
IP address from server: 192.0.2.4
Time taken: 2.80 milliseconds
Enter your DNS query: cse.du.ac.bd
IP address from server: 192.0.2.3
Time taken: 1.11 milliseconds
Enter your DNS query: google.com
IP address from server: 142.250.193.110
Time taken: 1.27 milliseconds
Enter your DNS query: ns1.cse.du.ac.bd
IP address from server: 192.0.2.1
Time taken: 1.94 milliseconds
Enter your DNS query: amazon.com
IP address from server: Not Found
Time taken: 2.41 milliseconds
Enter your DNS query: google.com
IP address from server: 142.250.193.110
Time taken: 1.10 milliseconds
Enter your DNS query: ns1.cse.du.ac.bd
IP address from server: 192.0.2.1
Time taken: 1.33 milliseconds
Enter your DNS query: ns2.cse.du.ac.bd
IP address from server: 192.0.2.2
Time taken: 1.99 milliseconds
Enter your DNS query:
```

Figure 15: Part-3 Client

5.3.3 Advantages :

1. It is a simple process and the client has less load compared to iterative process.

2. It is more reliable because the server handles the error checking itself.

5.3.4 Disadvantages :

1. It is a slower process.

2. Server faces high network traffic and implementation is complex because it has to handle same request for multiple client simultaneously.

3. Cache poisoning results from someone tricking a DNS server into believing that a fake DNS query response is authentic. Because responses are normally cached, this false information can be distributed to users of that server.

4. With recursive DNS queries enabled, a server is more easily hijacked and its performance compromised.

5.4 Part 4: Extending the System

DNS cache stores the most recent request of the client in local, root and TLD servers which makes the response faster. TTL value is the time which determines when the cache will be deleted.

```
araf@Arafs-MacBook-Air Recursive % python3 local_server.py
Local Server listening on localhost:8000
Local Server received query: cse.du.ac.bd from ('127.0.0.1', 62195)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 62195)
Record cse.du.ac.bd expired after 3 seconds. Removed from Local records.
Local Server received query: google.com from ('127.0.0.1', 52892)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 52892)
Record google.com expired after 3 seconds. Removed from Local records.
Local Server received query: go.com from ('127.0.0.1', 61921)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address Not Found
Local Server received query: ns1.cse.du.ac.bd from ('127.0.0.1', 52415)
Local Server sending query to Root Server 8001
Local Server received response from Root Server
IP address added to local records
IP address sent to ('127.0.0.1', 52415)
Record ns1.cse.du.ac.bd expired after 3 seconds. Removed from Local records.
```

Figure 16: Part-4 TTL for Local server

Figure 17: Part-4 TTL for Root server

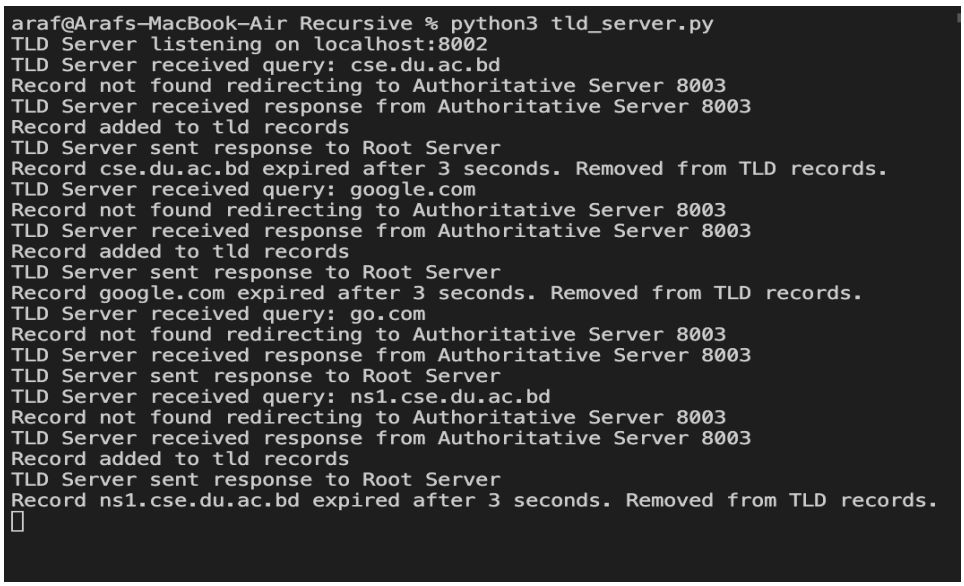


Figure 18: Part-4 TTL for Top-level Domain(TLD) server

5.5 Time Comparison: Iterative vs Recursive DNS resolution methods

The following graphs shows the time taken by each method to process same request message:

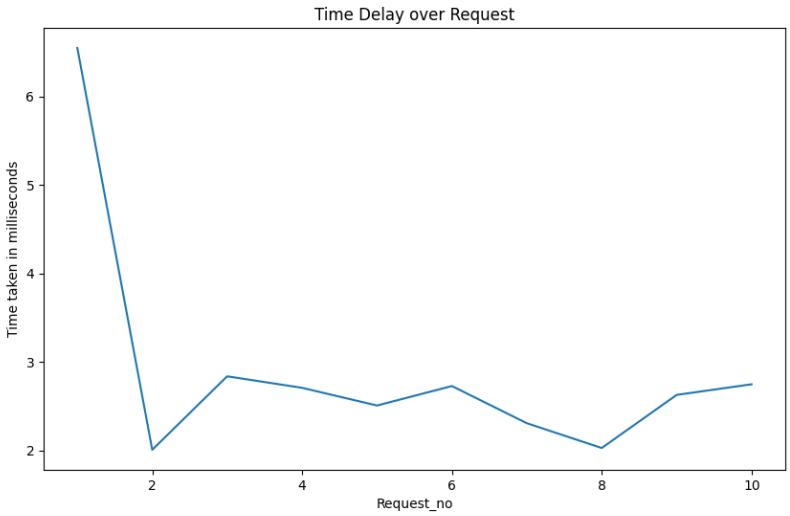


Figure 19: Time required for resolving a request via iterative DNS resolution method

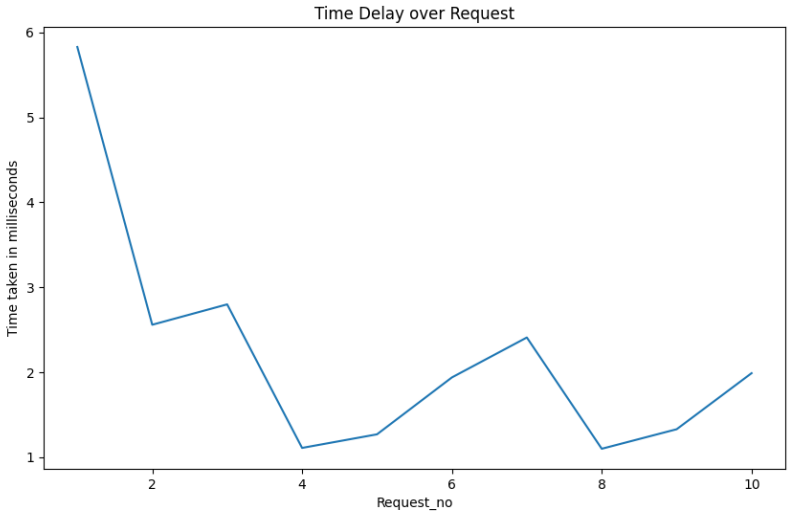


Figure 20: Time required for resolving a request via recursive DNS resolution method

References

[1] Amazon, *What is DNS?*, Available at: `https://aws.amazon.com/route53/what-is-dns/`

[2] ConstellixLAB, *DNS Record Types*, Available at: `https://constellix.com/news/dns-record-types`

[3] NETWORKWORLD, *What is DNS and how does it work?*, Available at: `https://www.networkworld.com/article/965540/what-is-dns-and-how-does-it-work.html`

[4] Cloudflare, *DNS server types*, Available at: `https://www.cloudflare.com/learning/dns/dns-server-types/`

[5] James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, 8th Edition, Pearson, 2022.