# 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: Md. Emon Khan

Roll No : 30

Name: Mahmudul Hasan

Roll No : 60

**Submitted To :**

Dr. Md. Abdur Razzaque

Dr. Md. Mamun Or Rashid

Dr. Muhammad Ibrahim

Md. Redwan Ahmed Rizvee

**Submitted On :** February 15, 2024

# 1 Introduction

The goal of lab-4 is to investigate the Domain Name System (DNS) through the implementation and analysis of the iterative and recursive resolution methods. We simulate a DNS client and server to understand their functionalities and compare their message flow, performance, and suitability for various scenarios. By testing and evaluating the implemented system, we aim to gain practical insights into the inner workings of DNS and contribute to a deeper understanding of this fundamental internet technology.

## 1.1 Objectives

Some of the preliminary objectives of this lab experiment are:

- To emulate the Domain Name Service (DNS) protocol
- To implement and understand the functionality of an iterative DNS resolution
- To implement and understand the functionality of a recursive DNS resolution
- To understand the differences between the iterative and recursive DNS resolutions
- To emulate caching mechanism of DNS servers and subsequent error handling

# 2 Theory

The Domain Name System (DNS) is a hierarchical distributed system that translates human readable domain names like ”www.example.com” into machine readable IP addresses. When a user tries to access a website, their computer follows a series of steps to resolve the domain name to an IP address:

- **Search DNS Cache:** The computer first checks its local DNS cache to see if the IP address for the domain name is already stored. If it is, the process is complete and the user is directed to the website.
- **Query ISP's DNS Servers:** If the IP address is not found in the cache, the computer queries its Internet Service Provider's (ISP) DNS

servers. These servers have a larger cache of domain names and IP addresses, and they may be able to resolve the query without needing to contact other servers.

- **Query Root Nameservers:** If the ISP's DNS servers do not have the answer, they query the root nameservers. The root nameservers are the top level of the DNS hierarchy, and they know the location of the Top-Level Domain (TLD) nameservers for all domains, e.g., .com, .org, .net etc.
- **Query TLD Nameservers:** The root nameservers direct the query to the TLD nameservers for the specific domain. These nameservers are responsible for a particular TLD, and they know the location of the authoritative nameservers for that domain.
- **Query Authoritative Nameservers:** The TLD nameservers direct the query to the authoritative nameservers for the specific domain. These nameservers are the ultimate source of information for the domain name, and they store the IP address for the domain.
- **Receive the Answer:** The authoritative nameservers send the IP address back to the ISP's DNS servers, which then cache the information and send it back to the user's computer.
- **Store the Answer:** The user's computer stores the IP address in its DNS cache for future reference.

This process is typically very fast, and users rarely notice the steps involved. However, it is important to understand how DNS works, as it is a fundamental part of the internet infrastructure.

# 3 Methodology

We use UDP connection to simulate a DNS server

## 3.1 Setting up the DNS Server

- We create a single server that acts as an authoritative DNS server
- The server awaits a client's request
- Upon receiving the request the server looks for the requested domain

- If the domain is found the server replies with the IP
- If the domain is not found the server replies with an error message

### 3.2 Iterative DNS resolution

- We create a rooted tree structure of DNS servers and have then run in different threads
- The client first sends a request to the root DNS server
- If the requested domain is found, the rooted server sends the IP and the query ends
- If the request is not found, the rooted server replies with a referral to a top level domain (TLD) DNS server
- The client then sends the request to that TLD server
- If the requested domain is found, the TLD server sends the reply and the query ends
- Else the TLD server replies with a referral to some authoritative DNS server
- Then the client sends the request to that authoritative DNS server
- The authoritative DNS server then replies with an answer if the domain is found, or else an error, anyway, ending the query

### 3.3 Recursive DNS resolution

- We create a rooted tree structure of DNS servers and have then run in different threads
- The client sends a recursive DNS query to recursive DNS resolver
- The recursive DNS resolver will send query to the root DNS server on behalf of the client
- If the root has the answer it'll send the answer to the resolver or else forward it to a TLD server
- If the TLD server finds the domain it'll reply with the IP to the root, otherwise forward to an authoritative DNS server

- If the authoritative server finds the domain it'll reply to the TLD with the IP or an error otherwise
- The replies will be backtracked and finally be received by the client through the resolver

### 3.4 Extending the System

- We try extending the system using a short TTL value
- Each server deletes a record when the TTL expires
- We implement DNS caching by saving successful query results
- We test the DNS server process failure

## 4 Experimental result

### 4.1 Setting up the DNS Server

#### Authoritative DNS Server

#### Client

### 4.2 Iterative DNS resolution

#### DNS Servers

#### Client

### 4.3 Recursive DNS resolution

#### DNS Servers

#### Client

### 4.4 Extending the System

## 5 Experience

- We had to look up how to use the dnslib package in Python
- We had a clearer look into the DNS message format

## References

[1] https://pypi.org/project/dnslib/

[2] https://github.com/paulc/dnslib

[3] https://aws.amazon.com/route53/what-is-dns

[4] https://constellix.com/news/dns-record-types