Assignment #4

Student Name: Binur Zhalenov

Student Id: bz1090

Github link: <https://github.com/bz1090/dns_app>

Table of Contents

[Introduction 3](#_Toc36655338)

[Description 3](#_Toc36655339)

[Implementation 4](#_Toc36655340)

[Conclusion 5](#_Toc36655341)

[Reference 6](#_Toc36655342)

# Introduction

A name server is a computer program that implements a network service against a directory service to provide answers to queries. It transforms an often humanly significant, text-based identifier into a system-internal, often numerical, or part addressing identity. The server conducts this function in response to a request from a database protocol.

An example of a name server is the Domain Name System (DNS) server component, one of the Internet's two principal namespaces. DNS servers 'most significant function is the conversion (resolution) of human-memorable domain names and hostnames into the corresponding numerical Internet Protocol (IP) addresses, the Internet's second main name space used to recognize and locate computer systems and services on the Internet.

# Description

An authoritative name server is a name server which provides answers to questions about names within a region. An authoritative-only name server returns answers only to queries about domain names which the administrator has explicitly configured. Name servers may also be designed to provide authoritative responses to queries in some areas while acting as a caching name server in all other areas

An authoritative name server may be either a primary (master) server, or a secondary (slave) server. The server that holds the authoritative copies of all records in that region is a primary server for a country. It is identified by start-of-authority (SOA) resource record. For a region, a secondary server uses an automated updating process to preserve an identical copy of a zone's primary server database. Examples of these mechanisms include DNS zone transfers and protocols for file transfer. DNS provides a mechanism by which the primary zone can notify all the known secondary’s for that zone when the zone contents have changed. The content of a zone is either configured manually by an administrator, or controlled using Dynamic DNS.

Examples of these mechanisms include DNS zone transfers, and protocols for file transfer. DNS provides a mechanism by which the primary zone can notify all the known secondaries for that zone when the zone contents have changed. The content of a zone is either configured manually by an administrator, or controlled using Dynamic DNS

When a domain is registered with a registrar of a domain name, the zone administrator provides the list of name servers (typically at least two, for redundancy) that are authoritative for the domain area. The registrar provides a domain registry with the names of certain servers for the top level domain containing the region. In addition, the domain registry configures the authoritative nameservers for that top-level domain with delegations for each zone server. When any name server's fully eligible domain name for a region exists within that region, the zone administrator supplies IP addresses for that name server, which are configured as glue records in the parent zone; otherwise the delegation consists of the list of NS records for that zone.

# Implementation

We have implemented an Authoritative Server which consist of the three major components:

**User Server**

A DNS server is a file server that maintains a database of public IP addresses and their related hostnames, and in most cases serves to resolve or convert those names as requested into IP addresses. DNS servers run special software, which use special protocols to communicate with each other.

The source code is included in userServer.py file.

**Fibonacci Server**

This will run the service on port 8080, and listen. The company is now in place and ready for business. This will provide you with a JSON blob (array) in the body [0, 1, 1, 2, 3] as well as a standard status code 200 in header. Answer is a status code in HTTP header with human readable message, and a JSON list in body text. Note body text can contain unexpected HTML entities, so before parsing the body as JSON, a client should always check HTTP status code.

The source code is included in fibonacciServer.py file.

**Authoritative Server**

I have implemented this to tell the customer whether they have configured the DNS of their domain for a specific service correctly. I want to search the authoritative DNS server for its domain. The source code is included in authoritativeServer.py file.

This uses the regular DNS server of the network to look up the root address for the top-level domain and to resolve the names of the different DNS servers in the chain, which I think is acceptable because such names will probably change very rarely.

# Conclusion

An authoritative name server offers a clear response to your DNS questions, such as – IP address of the mail server or IP address of the website. It provides initial and conclusive answers to DNS queries. It does not only have cached answers that were retrieved from another list of names. It therefore only returns answers to queries about domain names built in its configuration system.

# Reference

(n.d.). Retrieved from <https://www.w3schools.com/python/>

Python Tutorial. (n.d.). Retrieved from <https://www.tutorialspoint.com/python/index.htm>

Learn Python Tutorial - javatpoint. (n.d.). Retrieved from <https://www.javatpoint.com/python-tutorial>