BRNO UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

NETWORK APPLICATIONS AND NETWORK ADMINISTRATION

Semester project

DNS resolver

November 6, 2023 Vadim Goncearenco (xgonce00@stud.fit.vutbr.cz)

Contents

1 Theory

1.1 Domain Name System

DNS (Domain Name System) is a hierarchical naming system that maps domain names to IP addresses. It is used to translate human-readable domain names, such as *www.example.com*, into IP addresses, such as *192.0.2.1*, which are used to identify and locate computer services and devices on the internet.[2]

A DNS resolver is a program that queries DNS servers to resolve domain names into IP addresses. When a user enters a domain name into a web browser, the browser sends a DNS query to a DNS resolver, which then sends a query to a DNS server to obtain the IP address associated with the domain name.

DNS queries are used to obtain information about domain names, such as IP addresses, mail server information, and other resource records. A DNS query consists of a header and a question section, which contains the domain name being queried and the type of information being requested.

When a DNS query is sent, DNS resolver first checks the cache on a local device to see if it has the requested information. If the information is not in the cache, the resolver sends the query to a DNS server.

The DNS server then checks its cache to see if it has the requested information. If the information is not in the cache, there are two possibilities how the DNS server can act depending on the type of query.

There are two types of DNS queries: recursive and iterative.

- In a recursive query, the DNS server is responsible for obtaining the requested information and returning it to the client. Which means that the DNS server communicates with other DNS servers, if necessary, to fully resolve the domain name. Once the information is obtained, the DNS server returns the final result to the client.

- In an iterative query, the DNS server returns the best information it has available at the time of the query. The DNS server may provide referrals to other DNS servers that might have more specific information about the domain. The client is responsible for sending additional queries to these referred DNS servers if needed to continue the resolution process.

DNS queries can also be performed over UDP or TCP. UDP is used for small queries, while TCP is used for larger queries that require more data to be transmitted or for example in case of DNSSEC.

This section was partially adopted from `RFC 1035` [2].

2 Application design

The `dns` program is implemented in *C programming language* using *c99* language standard. For network operations the program utilizes *BSD socket libraries*. The following network libraries are used:

- `arpa/inet.h` - Contains functions for converting between network and host byte order.

- `netinet/in.h` - Contains constants and structures needed for internet domain addresses.

- `sys/socket.h` - Contains definitions of socket data types.

- `netdb.h` - Contains definitions of network database operations.

2.1 Project files

Project directory contains the following files:

- `dns.c` - Main file. Contains the `main` function and calls functions from other modules.

- `base.h` - Common header includes, macros and constants.

- `args.c` - Functions for parsing command line arguments.
- `args.h` - Header file for `args.c`.
- `dns_packet.c` - Contains functions for creating and parsing DNS requests.
- `dns_packet.h` - Header file for `dns_packet.c`.
- `Makefile` - Makefile for building the application and running the tests.
- `README.md` - Application usage.
- `test/test.py` - Python script for running the tests.
- `test/test_cases.json` - Test cases for the `test.py` script.
- `doc/manual.pdf` - This document.

3 Program usage

The following is the usage of the `dns` program:

```
NAME
    dns - DNS resolver

SYNOPSIS
    dns [-r] [-x|-6] -s server [-p port] domain|address
    dns -h

DESCRIPTION
    dns is a simple DNS resolver that can handle both IPv4 and IPv6
    addresses. Additionally, it has the capability to perform
    reverse queries and can query any DNS server. It supports
    recursive queries and can also communicate with DNS servers using
    IPv6 or a non-standard port.

OPTIONS
    -r
        Recursive query. If the DNS server does not have the answer,
        it will recursively query other DNS servers.

    -x
        Reverse DNS lookup. The address is interpreted as an
        IPv4/IPv6 address and a PTR query is sent.

    -6
        Send AAAA query to receive IPv6 address.

    -s server
        DNS server domain name or IPv4/IPv6 address to send a query to.

    -p port
        Port to use when querying the DNS server. Default is 53.
    -h
        Print help and exit.

    domain|address
        Domain name to query or IPv4 address to reverse query.
```

SIMPLE USAGE

```
    $ ./dns -r -s dns.google www.github.com
    Authoritative: No, Recursive: Yes, Truncated: No
    Question section (1)
        www.github.com., A, IN
    Answer section (2)
        www.github.com., CNAME, IN, 3600, github.com.
    github.com., A, IN, 60, 140.82.121.3
    Authority section (0)
    Additional section (0)

    $ ./dns -r -x -s dns.google 140.82.121.3
    Authoritative: No, Recursive: Yes, Truncated: No
    Question section (1)
        3.121.82.140.in-addr.arpa., PTR, IN
    Answer section (1)
        3.121.82.140.in-addr.arpa., PTR, IN, 2004,
    lb-140-82-121-3-fra.github.com.
    Authority section (0)
    Additional section (0)

    $ ./dns -r -s kazi.fit.vutbr.cz www.fit.vut.cz
    Authoritative: No, Recursive: Yes, Truncated: No
    Question section (1)
        www.fit.vut.cz., A, IN
    Answer section (1)
        www.fit.vut.cz., A, IN, 14400, 147.229.9.26
    Authority section (0)
    Additional section (0)

    $ ./dns -r -s kazi.fit.vutbr.cz www.github.com
    Authoritative: No, Recursive: Yes, Truncated: No
    Question section (1)
        www.github.com., A, IN
    Answer section (2)
        www.github.com., CNAME, IN, 3600, github.com.
        github.com., A, IN, 60, 140.82.121.3
    Authority section (0)
    Additional section (0)

    $ ./dns -r -x -s 2001:4860:4860::8844 2a00:1450:400d:80e::2004
    Question section (1)
        4.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.e.0.8.0.d.0.0.4.0.5.4.1.0.
```

```
0.a.2.ip6.arpa., PTR, IN
Authoritative: No, Recursive: Yes, Truncated: No
Answer section (1)
    4.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.e.0.8.0.d.0.0.4.0.5.4.1.0.
0.a.2.ip6.arpa., PTR, IN, 14261, bud02s39-in-x04.1e100.net.
Authority section (0)
Additional section (0)
```

The description of program usage can also be found in the `README.md` file.

4 Testing

Testing is done using the `test.py` script. The script reads test cases from the `test_cases.json` file and runs the program with the given arguments. The output of the program is then compared with the output of `dig` Linux utility program [3].

The script can be run using the following `Makefile` target:

```
make test
```

or by running the Python script directly:

```
usage:
    python3 test.py [-h] [-d] [-6] [-v] input_file

positional arguments:
    input_file          input JSON file

options:
    -h, --help          show this help message and exit
    -d, --debug         run in debug mode
                         (print all queries and responses)
    -6, --ignore-ipv6   ignore tests that required IPv6 support
                         on current machine
    -v, --ignore-vpn    ignore tests that require VUT FIT network VPN
                         (e.g. require querying kazi.fit.vutbr.cz)
```

Be aware though, that because the test script compares output of the `dns` program with output of the `dig` command, some tests may occasionally fail because both programs may resolve a domain name to a different address depending on the exact time of execution. For example, the domain name `www.github.com` can be resolved to either `140.82.121.3` or `140.82.121.4` IPv4 address. It may be because of *load-balancing*, which is used to distribute traffic across multiple servers by returning different IP addresses in response to DNS queries [1].

So to ensure that the program works correctly, it may be necessary to run the tests multiple times.

5 Project task extensions and ambiguities

1. Project task does not explicitly state the program behavior when combination of flags `-x` and `-6` is provided.

   In my implementation I decided not to support this combination of flags because this combination is ambiguous or does not make sense.

2. Project task does not explicitly state whether the program should support passing the DNS server name as an IP address.

   My implementation supports passing the server address (`-s server`) as both IPv4 and IPv6 addresses.

3. Project task does not explicitly state whether the program should support reverse queries (`PTR`) for IPv6 addresses.

   My implementation supports reverse queries for both IPv4 and IPv6 addresses.

References

[1] Thomas P. Brisco. *DNS Support for Load Balancing*. Tech. rep. RFC 1794. Apr. 1995. DOI: `10.17487/RFC1794`. URL: `https://datatracker.ietf.org/doc/rfc1794` (visited on 10/20/2023).

[2] *Domain names - implementation and specification*. Tech. rep. RFC 1035. Nov. 1987. DOI: `10.17487/RFC1035`. URL: `https://datatracker.ietf.org/doc/rfc1035` (visited on 10/20/2023).

[3] *IBM Documentation*. en-US. online. Mar. 2023. URL: `https://www.ibm.com/docs/en/aix/7.1?topic=d-dig-command` (visited on 10/20/2023).