

Nicholas Everekyan, Ashley Kim, Mehjabeen Tasnim Khan
CS433: Networking
Professor Li
November 25th, 2024

Project 3 - DNS Client & Server

Introduction

This project involved designing and implementing a DNS system with a client and two server programs (localserver.py and amazoneserver.py). The client queries the servers over UDP sockets, retrieving resource records based on query types (A, AAAA, CNAME, NS). The servers maintain RR tables and manage TTL values for time-sensitive records. The system was designed to simulate core DNS functionality in a Python environment.

Development Process

System Design:

- Defined communication protocols and structured the RR table format.
- Identified query types and functionalities: A, AAAA, CNAME, and NS records.

Implementation

- **Utilized generative AI to aid in a skeleton of what functions were necessary to complete the task at hand.**
- Developed dnsclient.py for sending queries and parsing responses.
- Implemented localserver.py as the primary resolver for client queries.
- Implemented amazoneserver.py to provide authoritative records when queried by localserver.py.

- Introduced dynamic TTL handling to decrement the value with time, ensuring expired records are removed.

Testing:

- Performed the tests given in the project document, which contained querying `www.csusm.edu` and `shop.amazone.com` to check for the TTL, as well as utilizing a second client to test the local DNS and how it handles requests from a different client.

Functionality Testing

Below are screenshots showing the DNS Client, Local Server, and Amazone Server outputs to detail the 5 test cases. The fourth screenshot displays the second client that is being run to test querying `shop.amazone.com` to test the caching of the DNS.

```
ashleykim@MacBookAir CS436_project3 % python3 client.py
Enter the hostname (or type 'quit' to exit): www.csusm.edu
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,60,0
Enter the hostname (or type 'quit' to exit): www.csusm.edu
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,55,0
Enter the hostname (or type 'quit' to exit): shop.amazone.com
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,40,0
1,shop.amazone.com,A,3.33.147.88,60,0
Enter the hostname (or type 'quit' to exit): shop.amazone.com
record_no,name,type,result,ttl,static
0,shop.amazone.com,A,3.33.147.88,60,0
```

```
ashleykim@MacBookAir CS436_project3 % python3 localserver.py
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,None,1
1,my.csusm.edu,A,144.37.5.150,None,1
2,amazone.com,NS,dns.amazone.com,None,1
3,dns.amazone.com,A,127.0.0.1,None,1
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,None,1
1,my.csusm.edu,A,144.37.5.150,None,1
2,amazone.com,NS,dns.amazone.com,None,1
3,dns.amazone.com,A,127.0.0.1,None,1
4,shop.amazone.com,A,3.33.147.88,60,0
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,None,1
1,my.csusm.edu,A,144.37.5.150,None,1
2,amazone.com,NS,dns.amazone.com,None,1
3,dns.amazone.com,A,127.0.0.1,None,1
4,shop.amazone.com,A,3.33.147.88,60,0
record_no,name,type,result,ttl,static
0,www.csusm.edu,A,144.37.5.45,None,1
1,my.csusm.edu,A,144.37.5.150,None,1
2,amazone.com,NS,dns.amazone.com,None,1
3,dns.amazone.com,A,127.0.0.1,None,1
4,shop.amazone.com,A,3.33.147.88,44,0
```

```
ashleykim@MacBookAir CS436_project3 % python3 amazoneserver.py
record_no,name,type,result,ttl,static
0,shop.amazone.com,A,3.33.147.88,None,1
1,cloud.amazone.com,A,15.197.140.28,None,1
record_no,name,type,result,ttl,static
0,shop.amazone.com,A,3.33.147.88,None,1
1,cloud.amazone.com,A,15.197.140.28,None,1
```

```
ashleykim@MacBookAir CS436_project3 % python3 client.py
Enter the hostname (or type 'quit' to exit): shop.amazone.com
record_no,name,type,result,ttl,static
0,shop.amazone.com,A,3.33.147.88,60,0
```

Conclusion

This project provided experience with socket programming and DNS functionality. We learned the importance of precise time-based operations in dynamic systems and the challenges of inter-process communication. The utilization of generative AI allowed us to properly assess problem areas in our code to expedite and aid in our debugging process. Our areas for improvement are optimizing record retrieval for large datasets and extending functionality for additional DNS query types.