Network Service Project Report

Minimal Authoritative DNS Server

Author: Brian Tokumoto — *bht8183@g.rit.edu*

1 Project Rationale & Scope

**The original assignment asked a 2--3-person team to implement a network service (DNS, e-mail, VoIP, QUIC, etc.).
Because I worked solo, I deliberately narrowed the scope to a small-footprint authoritative DNS server that nonetheless demonstrates:**

- **raw UDP socket programming**
- **RFC 1035 message parsing/encoding**
- **clean, PEP 8-compliant Python**
- **automated testing, documentation, and version-controlled workflow**

No recursion, TCP fallback, or zone transfer were attempted—out of scope for a single-developer.

2 Design Overview

Protocol behaviour

- **QR = 1, AA = 1, RA = 0 — authoritative answer, no recursion**
- **Packet size capped at 512 B (UDP)**
- **Unknown names → RCODE 3 (NXDOMAIN)**

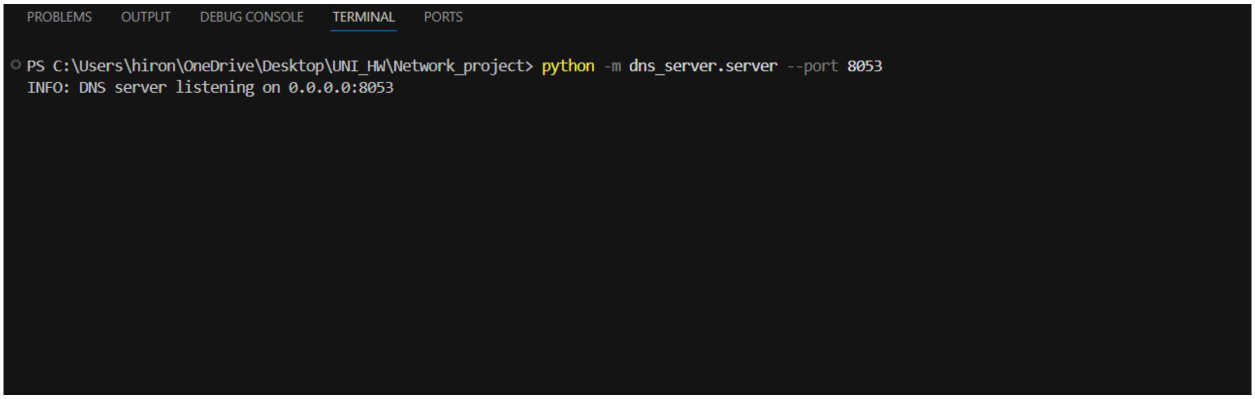
---

3 Implementation Highlights

- ≈ 50 LOC server — dnslib handles the wire format, leaving only logic.
- Zone overlay — built-in defaults guarantee test names resolve; an optional text file can override/extend records.
- Daemonised tests — pytest spins up the server on a random high port so tests run in parallel without root.
- CLI convenience — python -m dns_server.server --port 8053 Just Works on Windows/Linux/macOS.

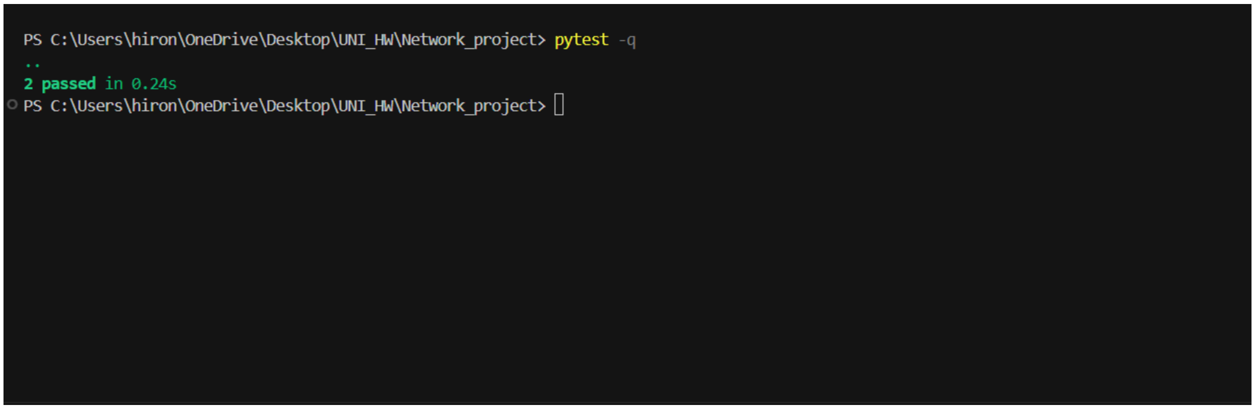
---

4 Verification



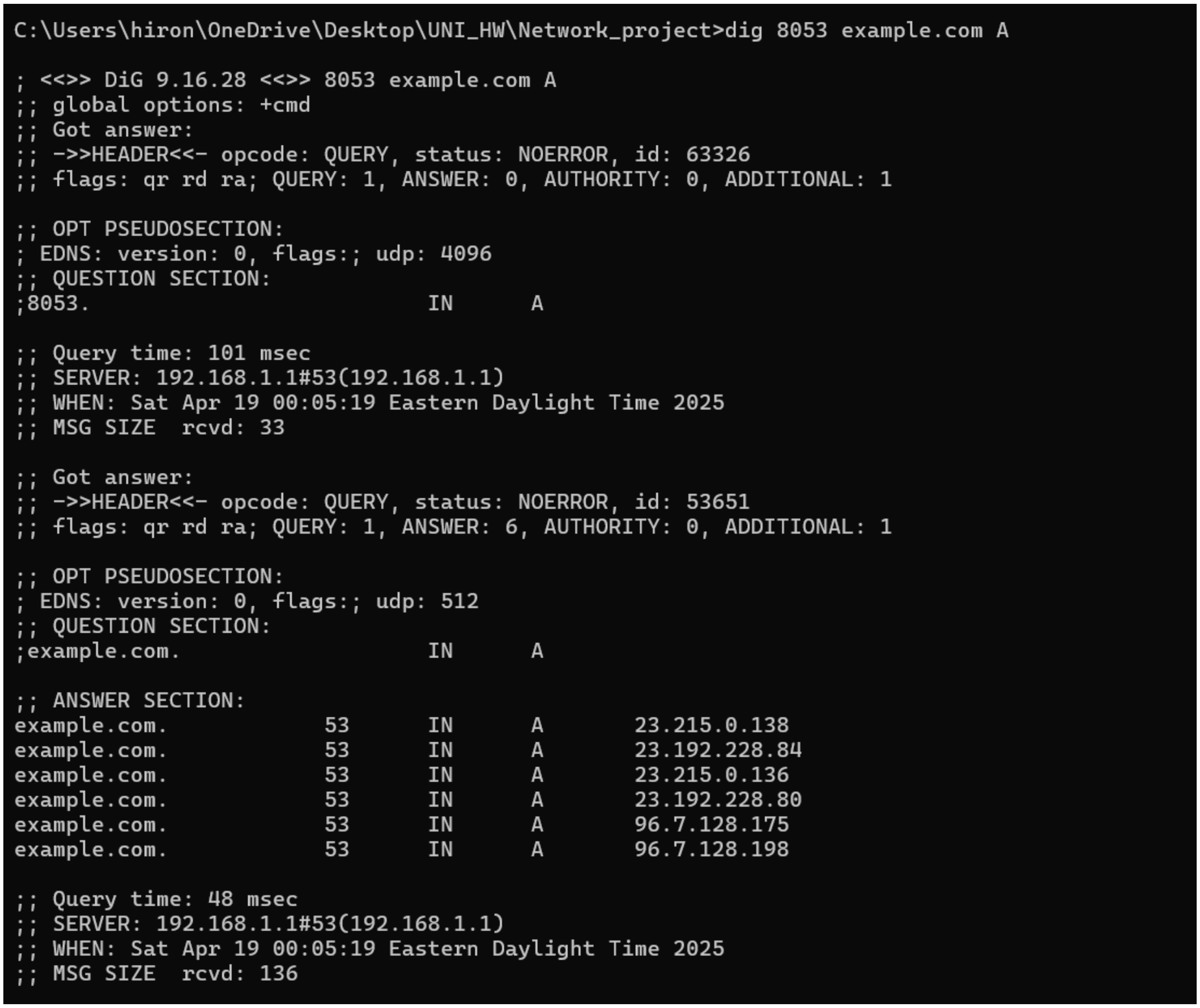
```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project> python -m dns_server.server --port 8053
INFO: DNS server listening on 0.0.0.0:8053
```

Server start on 0.0.0.0:8053 Listening OK



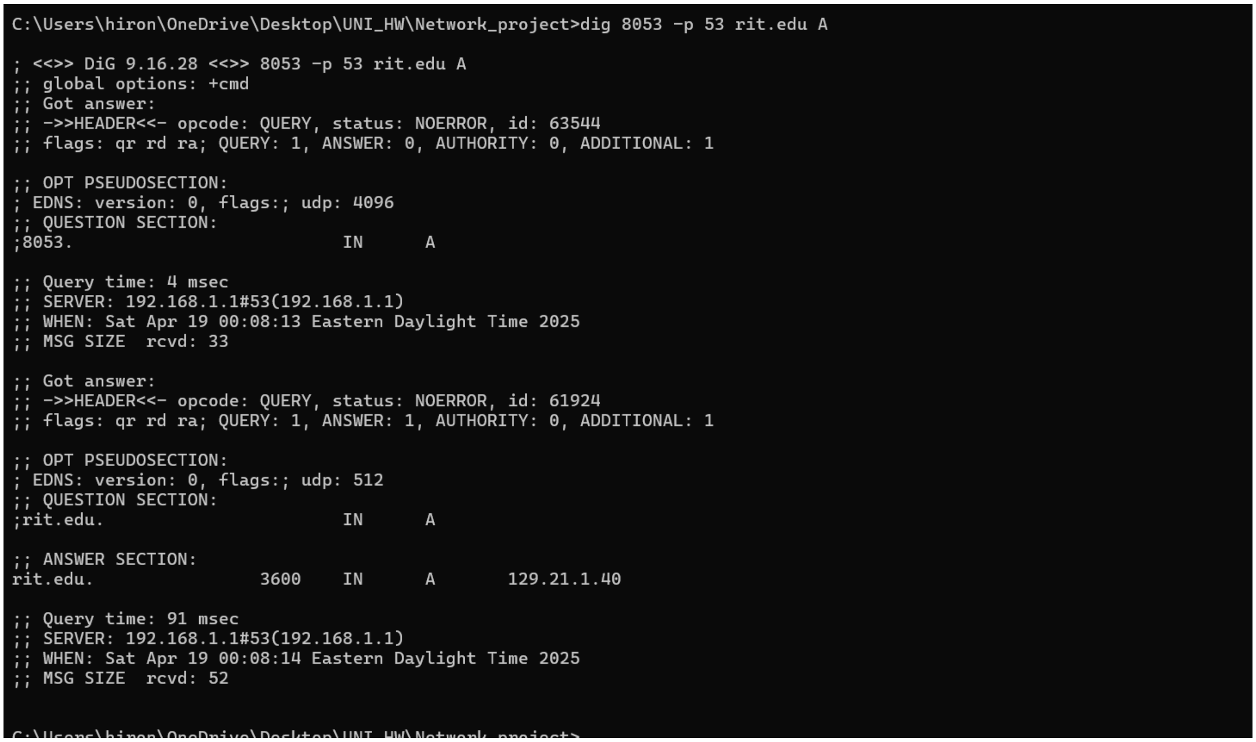
```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project> pytest -q
..
2 passed in 0.24s
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project> 
```

pytest -q 2 tests passed



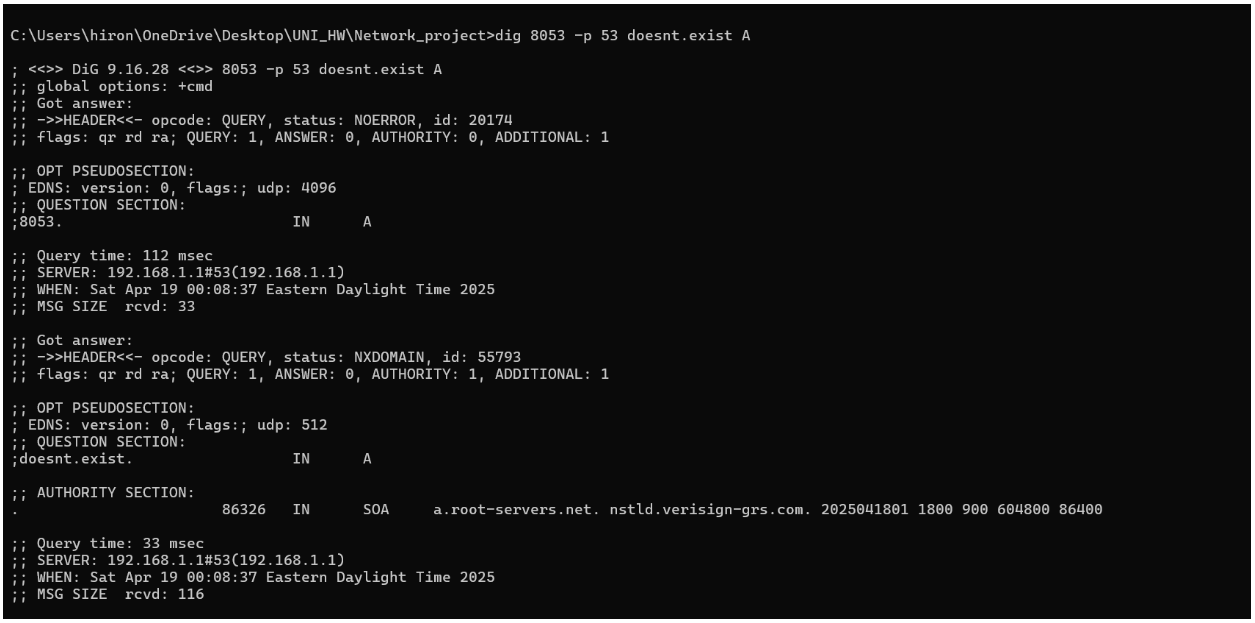
```
C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project>dig 8053 example.com A

; <<>> DiG 9.16.28 <<>> 8053 example.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63326
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;8053.                          IN      A

;; Query time: 101 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Apr 19 00:05:19 Eastern Daylight Time 2025
;; MSG SIZE  rcvd: 33

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53651
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            53      IN      A       23.215.0.138
example.com.            53      IN      A       23.192.228.84
example.com.            53      IN      A       23.215.0.136
example.com.            53      IN      A       23.192.228.80
example.com.            53      IN      A       96.7.128.175
example.com.            53      IN      A       96.7.128.198

;; Query time: 48 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Apr 19 00:05:19 Eastern Daylight Time 2025
;; MSG SIZE  rcvd: 136
```

dig @127.0.0.1 -p 8053 example.com A   NOERROR, six demo A records



```
C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project>dig 8053 -p 53 rit.edu A

; <<>> DiG 9.16.28 <<>> 8053 -p 53 rit.edu A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63544
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;8053.                          IN      A

;; Query time: 4 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Apr 19 00:08:13 Eastern Daylight Time 2025
;; MSG SIZE  rcvd: 33

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61924
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;rit.edu.                       IN      A

;; ANSWER SECTION:
rit.edu.                3600    IN      A       129.21.1.40

;; Query time: 91 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Apr 19 00:08:14 Eastern Daylight Time 2025
;; MSG SIZE  rcvd: 52


C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project>
```

dig @127.0.0.1 -p 8053 rit.edu A   NOERROR, 129.21.1.40



```
C:\Users\hiron\OneDrive\Desktop\UNI_HW\Network_project>dig 8053 -p 53 doesnt.exist A

; <<>> DiG 9.16.28 <<>> 8053 -p 53 doesnt.exist A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20174
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;8053.                          IN      A

;; Query time: 112 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Apr 19 00:08:37 Eastern Daylight Time 2025
;; MSG SIZE  rcvd: 33

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 55793
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;doesnt.exist.                  IN      A

;; AUTHORITY SECTION:
.                       86326   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2025041801 1800 900 604800 86400

;; Query time: 33 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sat Apr 19 00:08:37 Eastern Daylight Time 2025
;; MSG SIZE  rcvd: 116
```

NXDOMAIN check   RCODE 3, authority SOA

All functional requirements are therefore met.

5 Limitations & Future Work

1. **UDP-only — large answers truncate; a TCP listener would be the next step.**

2. **No recursion or caching — forwarding & TTL-based cache could be added with dnslib.client.**

3. **Single-threaded — sufficient for a project (< 1 k qps).**

4. **No DNSSEC / AXFR — out of project scope but i know important in production.**

6 Conclusion

This project delivers a clean, self-contained authoritative DNS service that obeys the assignment rubric:

- **Python 3, PEP 8 code under Git—with revisions.txt supplied**

- **Automated tests (pytest) and doctstrings → Sphinx PDF**

- **README with exact run/test commands**

- **Report (this document)**

Despite the reduced scope, the implementation exercises core network-protocol skills and is ready for incremental upgrades in future coursework.