# Network Programming Laboratory

*13 June 2025*

## Exercise

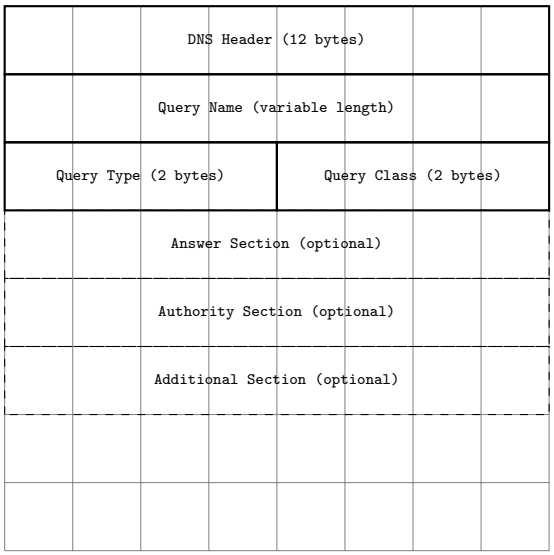
The **Domain Name System (DNS)** is a distributed hierarchical system that resolves human-readable domain names (e.g., `www.example.com`) into IP addresses (e.g., `93.184.216.34`). A DNS query typically involves:

- A client sending a query to a recursive DNS server.
- UDP port 53 is used for most DNS queries (TCP is used for larger packets or DNSSEC).
- The server replies with an answer or a referral.

A DNS packet contains a header, questions, and (optionally) answers, authorities, and additional information.
In any case, each query contains:

- A domain name
- A query type (QTYPE): e.g., `A`, `CNAME`, `NS`, `MX`, `AAAA`, etc.

### DNS Query Packet Format (for reference)




Write a Rust program that captures or analyzes DNS queries from network traffic and logs detailed information and statistics.

### Command-Line Interface

Use the `clap` crate to implement the CLI with:

- `--interface` or `-i`: Capture live packets on the selected interface (e.g., `eth0`)
- `--pcap-file` or `-f`: Analyze packets from an offline PCAP file
- `--interval` or `-n`: Time interval in seconds to print stats (default: 5)

Only one of `--interface` or `--pcap-file` may be used at a time. The program should print:

- For live mode: `[*] Reading live packets from interface 'eth0'...`
- For offline mode: `[*] Reading packets from file 'trace.pcap'...`

## DNS Query Packet Processing

Use the `pcap` crate to handle packets:

- In live mode: Open the interface in promiscuous mode and apply the proper BPF filter
- In offline mode: Open the PCAP file and apply the same filter
- For each packet:
  - Parse the UDP payload as a DNS packet using `simple-dns`
    - The `Packet::parse()` method requires the DNS payload only as input
  - For each question:
    - Collect *query name* and *query type* for logging and statistics

## Output and Statistics

In live mode, the program must print the statistics every `n` seconds (from `--interval`). In the offline mode, stats are printed once at the very end, when processing is completed. The output should include:

- Current timestamp
- **Query Type Counts** (only A, CNAME, MX, NS, Other)
- **Top 5 Domain Names** queried, with counts, in descending order

Example:

```
=== DNS Stats at 14:52:03 ===

Query Type Statistics:
A: 15
CNAME: 8

Top 5 Queried Domains:
1. www.google.com - 6
2. api.github.com - 4
3. www.example.com - 2
4. ftp.example.org - 1
5. docs.rust-lang.org - 1
```