Bialystok University of Technology
Faculty of Electrical Engineering

LABORATORY REPORT

Computer Networks
IS-FEE-10082S

Subject:
DNS Testing and Configuration

Instructor: Andrzej Zankiewicz, PhD

Students:
Bahadır Emre Yıldırım
Burak Düzenli

Białystok, May 2025

Contents

1 Objective

The main objective of this laboratory exercise is to develop a practical understanding of the Domain Name System (DNS), which is a fundamental component of modern computer networks.

This lab is designed to familiarize students with:

- The structure and functioning of the DNS system, including the concepts of root servers, top-level domains (TLDs), authoritative name servers, and recursive resolution.
- Various types of DNS records such as `A`, `NS`, `MX`, `CNAME`, and `SOA`, and their roles in name resolution, mail routing, and alias management.
- The use of the `nslookup` command-line tool for querying DNS servers and analyzing the DNS resolution process at different levels of detail.
- Interpreting the DNS responses in both standard and debug modes to identify and understand the behavior of the DNS infrastructure.
- Diagnosing network configurations related to DNS, including IP addressing, gateway routing, and DNS server assignments using system utilities such as `ip`, `resolv.conf`, and `nslookup`.

The exercise is also a step toward understanding more advanced DNS-related operations such as configuring a custom DNS server, setting up new DNS zones, and analyzing the Start of Authority (SOA) and authoritative name records in a controlled environment.

By the end of the session, students should be capable of performing DNS diagnostics, interpreting the results, and understanding how DNS plays a critical role in the functioning and performance of networked systems.

2 Network Configuration

```
>ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 ...
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
    inet 10.1.0.113/24 brd 10.1.0.255 scope global dynamic ...
    inet6 fe80::9775:ab09:716f:56a6/64 scope link ...
```

Listing 1: Output of ‘ip a‘

The `ip a` command displays all network interfaces and their assigned IP addresses. In this case, the loopback interface (`lo`) is configured with the standard local IP address `127.0.0.1`, and the Ethernet interface (`eth0`) is dynamically assigned the IP address `10.1.0.113/24` via DHCP, along with a link-local IPv6 address.

```
>ip r
default via 10.1.0.1 dev eth0 proto dhcp src 10.1.0.113
10.1.0.0/24 dev eth0 proto kernel scope link src 10.1.0.113
```

Listing 2: Output of 'ip r'

The routing table shown by the `ip r` command indicates that the default gateway for accessing external networks is `10.1.0.1`. This means all traffic destined outside the local `10.1.0.0/24` network will be forwarded to this gateway.

```
>cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 212.33.64.2
nameserver 212.33.64.18
```

Listing 3: Contents of /etc/resolv.conf

The DNS resolution settings in `/etc/resolv.conf` specify two nameservers: `212.33.64.2` and `212.33.64.18`. These addresses belong to external DNS servers that are responsible for resolving domain names into IP addresses. These servers will be used by the system's resolver library to perform all DNS lookups unless overridden by a manual configuration or tool-specific setting (such as changing the server in `nslookup`).

Together, these configurations ensure that the system can correctly identify itself on the local network, route packets to external destinations, and resolve domain names—thus enabling full participation in network communication.

3 Launching `nslookup`

```
>nslookup
> server 8.8.8.8
Default server: 8.8.8.8
Address: 8.8.8.8#53
```

Listing 4: Setting DNS server to 8.8.8.8

The `nslookup` utility is a command-line tool used to query Internet domain name servers. It is primarily used for troubleshooting DNS-related problems and verifying DNS records. When launched, `nslookup` automatically uses the system's default DNS server, as configured in `/etc/resolv.conf`. In this case, the initial output confirms that the default DNS server is correctly recognized.

However, `nslookup` also allows users to change the DNS server manually using the `server` command. This feature is particularly useful when testing alternative name servers or bypassing potentially misconfigured local resolvers. In the example provided, the DNS server is switched to `8.8.8.8` — which is a public DNS server provided by Google — to ensure reliable and independent name resolution. All subsequent queries in the session will be sent to this server.

This task introduces a foundational concept in DNS diagnostics: different DNS servers may return different results due to cache status, propagation delays, or configuration differences. The ability to direct queries to specific servers is therefore an essential skill in network administration and troubleshooting.

4 IP Address Lookup of Hosts

```
> www.google.com
Address: 142.250.179.164, 2a00:1450:400e:802::2004

> www.pb.edu.pl
Address: 212.33.68.5

> www.bialystok.pl
Address: 212.33.91.214

> www.microsoft.com
Address: 23.52.185.219 and multiple IPv6 addresses

> www.cisko.com
Address: 104.143.9.110, 104.143.9.111
```

Listing 5: IP lookup results

DNS queries were made for several domain names using the `nslookup` tool. These queries typically return `A` records (IPv4 addresses) and optionally `AAAA` records (IPv6 addresses) associated with the requested hostnames.

For example, querying `www.google.com` returned both IPv4 (`142.250.179.164`) and IPv6 (`2a00:1450:400e:802::2004`) addresses, demonstrating how modern DNS systems support dual-stack networking. Similarly, queries for `www.pb.edu.pl` and `www.bialystok.pl` returned IPv4 addresses corresponding to their web servers.

More complex resolution chains can also occur. For instance, the result for `www.microsoft.com` includes a sequence of `CNAME` (Canonical Name) records that eventually resolve to a content delivery network (CDN) endpoint (e.g., `e13678.dscb.akamaiedge.net`). This illustrates the use of aliasing in DNS to support load balancing and geographic distribution of content.

The `nslookup` results confirm that each domain is correctly registered and associated with one or more IP addresses, validating the functionality of the name resolution process. This step is essential for ensuring that end users can access web services by name, without needing to remember numeric IP addresses.

5 NS Records

```
> set querytype=ns
> pb.edu.pl
dns.man.bialystok.pl, master.man.bialystok.pl

```

```
> pl
a-dns.pl, b-dns.pl, h-dns.pl, f-dns.pl, d-dns.pl, j-dns.pl

> google.com
ns1.google.com, ns2.google.com, ns3.google.com, ns4.google.com

> bialystok.pl
a-dns.pl, b-dns.pl, d-dns.pl, f-dns.pl, h-dns.pl, j-dns.pl
```

Listing 6: NS Records

The `nslookup` tool was used to query `NS` (Name Server) records for several domains. `NS` records indicate which authoritative DNS servers are responsible for a given domain. These servers store the official DNS information for that domain and are queried directly during the resolution process.

For example, querying `pb.edu.pl` returned two authoritative name servers: `dns.man.bialysto` and `master.man.bialystok.pl`. Similarly, querying the top-level `.pl` domain yielded multiple root-level DNS servers (e.g., `a-dns.pl`, `b-dns.pl`, etc.), reflecting the distributed and redundant design of DNS.

Queries for domains like `google.com` and `bialystok.pl` revealed multiple authoritative name servers as well, enhancing fault tolerance and availability. By using geographically distributed name servers, DNS providers ensure faster response times and higher resilience against failures or attacks.

Understanding NS records is essential because they define the trust boundaries within the DNS hierarchy. When a recursive resolver attempts to answer a query, it follows these NS records up and down the domain tree to find the server that holds the authoritative answer. This mechanism is at the core of how the DNS system operates globally.

6 MX Records

```
> set querytype=mx
> gmail.com
5 gmail-smtp-in.l.google.com.
10 alt1.gmail-smtp-in.l.google.com.
20 alt2.gmail-smtp-in.l.google.com.
30 alt3.gmail-smtp-in.l.google.com.
40 alt4.gmail-smtp-in.l.google.com.

> pb.edu.pl
0 poczta.pb.edu.pl

> student.pb.edu.pl
0 student-pb-edu-pl.mail.eo.outlook.com
```

Listing 7: MX Records

This task involved querying the `MX` (Mail Exchanger) records for several domains to identify the servers responsible for handling their email traffic. MX records are a fundamental part of DNS used by mail servers to determine how to route emails sent to a domain.

For example, the domain `gmail.com` returned multiple MX records, each with a different priority value. These records include servers such as `gmail-smtp-in.l.google.com` and `altX.gmail-smtp-in.l.google.com`. Lower priority values (e.g., 5 or 10) indicate higher preference, meaning mail servers will attempt delivery to these addresses first. This redundancy ensures reliability and load balancing across Google's mail infrastructure.

The domain `pb.edu.pl` returned a single MX record pointing to `poczta.pb.edu.pl`, which suggests a simpler mail setup typically found in smaller or institutional networks. Similarly, the subdomain `student.pb.edu.pl` is handled by `student-pb-edu-pl.mail.eo.outlook.` indicating that the university leverages Microsoft's cloud mail service for students.

These queries demonstrate how mail delivery relies on the DNS system, and how organizations can structure their email infrastructure using MX records to balance load, ensure redundancy, or outsource services to external providers such as Outlook or Google Workspace.

7 Debug and d2 Options in nslookup

The `debug` and `d2` options in `nslookup` are used to gain more insight into the DNS query process by increasing the verbosity of the output.

```
> set debug
> google.com

------------
    QUESTIONS:
        google.com, type = A, class = IN
    ANSWERS:
    ->  google.com
        internet address = 142.251.36.14
        ttl = 169
------------
    QUESTIONS:
        google.com, type = AAAA, class = IN
    ANSWERS:
    ->  google.com
        has AAAA address 2a00:1450:400e:80f::200e
        ttl = 264
------------
```

Listing 8: Using 'set debug' option

The `set debug` option enhances the visibility of the DNS response by explicitly printing sections such as `QUESTIONS`, `ANSWERS`, `AUTHORITY RECORDS`, and `ADDITIONAL RECORDS`. Each resource record includes detailed information such as the record type (e.g., A or

AAAA), class (usually IN for Internet), time-to-live (TTL), and resolved addresses. This level of detail is helpful when verifying how the DNS server structures and returns data.

```
> set d2
> google.com

addlookup()
make_empty_lookup()
looking up google.com
setup_lookup()
cloning server list
make_server(212.33.64.2)
recursive query
add_question()
create query 0x7f2b5f26bc00 linked to lookup 0x7f2b5f378800
sendcount=1
recvcount=0
before parse starts
after parse
printmessage()
...
dighost.c:2734:lookup_attach(0x7f2b5f378800) = 4
canceling pending query 0x7f2b5f26bc00
destroy_lookup
freeing server 0x7f2b5f2b6a00
```

Listing 9: Using ‘set d2‘ option (partial output)

The `set d2` (debug level 2) provides a significantly deeper view. It reveals the internal workings of the DNS query mechanism implemented in `nslookup`. The output logs function calls like `make_empty_lookup()`, `clone_server_list()`, `start_udp()`, and `recv_done()`, along with memory addresses of query and lookup objects. It also shows the creation, transmission, and cancellation of DNS queries at a low level. These internal logs are essential for developers or advanced users debugging the DNS client implementation itself, rather than the external DNS behavior.

Both options serve different purposes: `debug` is mainly for DNS record inspection, while `d2` is used for analyzing the internal logic of query construction and protocol-level transactions.

8 DNS Server Installation and SOA Record Interpretation

To begin the lab, the BIND9 DNS server was installed and started on a Linux system (Kali). The status of the DNS service was verified using the command `sudo systemctl status bind9`, which confirmed that the service was active and running. The DNS configuration file was modified to include a new primary zone named `"lab"`:

```
zone "lab" {
    type master;
    file "/etc/bind/forward.lab";
};
```

The corresponding zone file (`forward.lab`) was created with a Start of Authority (SOA) record. The SOA record defines authoritative information about the zone:

```
@ IN SOA lab. admin.lab. (
    1         ; Serial
    604800    ; Refresh
    86400     ; Retry
    2419200   ; Expire
    604800    ; Negative Cache TTL
)
```

Each parameter in the SOA record has a specific role:

- **Serial**: Version number of the zone file; used by secondary DNS servers to detect updates.
- **Refresh**: Time interval (in seconds) for the secondary server to check for zone updates (7 days).
- **Retry**: Time interval before retrying a failed zone transfer (1 day).
- **Expire**: Maximum time the zone is considered valid if not refreshed (28 days).
- **Negative Cache TTL**: Time-to-live for negative responses (e.g., non-existent domain replies) (7 days).

9 Adding and Verifying an A Record

To complete the second task, a type A record was added to the `lab` zone in the `forward.lab` file as shown:

```
test IN A 1.2.3.4
```

This entry maps the hostname `test.lab` to the IP address `1.2.3.4`. After reloading the DNS service, the record was verified using two methods:

- **dig command**:

  ```
  $ dig @10.1.0.129 test.lab
  ...
  ;; ANSWER SECTION:
  test.lab.   604800  IN  A   1.2.3.4
  ```

- **nslookup command**:

```
> server 10.1.0.129
> test.lab
Name:   test.lab
Address: 1.2.3.4
```

Both methods confirmed that the DNS server successfully resolved `test.lab` to the correct IP address, demonstrating that the A record was properly configured and functioning as expected.

10 Conclusion

The laboratory exercise provided a comprehensive hands-on experience with the Domain Name System (DNS), covering both theoretical concepts and practical implementations. Through this lab, we gained a deeper understanding of DNS operations, including name resolution, different types of DNS records (A, NS, MX, SOA), and the role of authoritative and recursive DNS servers.

Key takeaways from this session include:

- The ability to use `nslookup` and `dig` to query DNS records and analyze responses in both standard and debug modes, which is crucial for troubleshooting DNS-related issues.
- The importance of DNS record types such as `A` (address mapping), `NS` (name server delegation), and `MX` (mail server routing) in ensuring proper network functionality.
- The role of the Start of Authority (SOA) record in defining zone parameters, including refresh intervals and TTL values, which are essential for DNS zone management.
- Practical experience in configuring a local DNS server (BIND9) and adding custom DNS records, reinforcing the process of DNS zone file management.

Additionally, the lab highlighted the distributed and hierarchical nature of DNS, demonstrating how queries propagate from root servers to authoritative name servers. The successful resolution of manually configured records (e.g., `test.lab`) confirmed the correctness of our DNS server setup.

This exercise not only enhanced our technical skills in DNS configuration and diagnostics but also emphasized the critical role DNS plays in modern networking. The knowledge gained will be invaluable for future network administration tasks, particularly in troubleshooting, optimizing DNS performance, and ensuring reliable name resolution in complex network environments.

In conclusion, the lab achieved its objectives by bridging theoretical DNS concepts with real-world implementation, providing a solid foundation for further exploration of advanced DNS configurations and security considerations.

11 References

1. Mockapetris, P. (1987). Domain names - implementation and specification (RFC 1035). Network Working Group. https://tools.ietf.org/html/rfc1035
2. Liu, C., Albitz, P. (2006). DNS and BIND (5th ed.). O'Reilly Media.
3. BIND9 Administrator Reference Manual. (n.d.). Internet Systems Consortium. https://www.isc.org/bind/
4. DNS for Rocket Scientists. (n.d.). Zytrax. https://www.zytrax.com/books/dns/
5. Linux man pages: dig, host, nslookup. (n.d.). https://linux.die.net/man/
6. DNS Root Servers. (n.d.). IANA. https://www.iana.org/domains/root/servers