TÉCNICO LISBOA

Computer Networks — 2021/22		Assignment:	Lab 4
Understanding and configuring DNS		Issued:	2021-12-09
Domain Name System		Submission Due:	2021-12-12
Authors:	João M. Tiago Prof. Luis D. Pedrosa	Comments Due:	2021-12-16
		Version:	1.0

Submission note. Answers to the questions in this assignment are to be submitted in Moodle (remember to store a local copy of the answers to avoid losing information). You should create a PDF file containing the answers, and submit the file to Moodle as an attachment to the submission form. Remember to include all relevant screenshots and diagrams in the PDF file itself.

1 Goals of the laboratory

- Explore different types of DNS (resource) records (SOA, A, AAAA, NS, CNAME, and MX).
- Learn the differences between an iterative and a recursive DNS query.
- Analyze how DNS caching impacts query time.
- Learn how to configure a DNS zone.

2 Exercises

2.1 Types of DNS records

In this part of the assignment, you will be using `dig`, a tool to perform DNS lookups, to learn more about the most important types of DNS records: SOA, A, AAAA, NS, CNAME, and MX. The tool supports multiple flags to control the DNS query, like `-t`, which lets you request a specific type of DNS record (possible values are `any`, `soa`, `a`, `aaaa`, `ns`, `cname`, `mx`, etc.).

Hint. To answer the questions at the bottom of the section, analyze the output of `dig`.

2.1.1 DNS record syntax

DNS records follow a simple syntax, as shown below (this is a simplified version).

```
<name> <ttl> <class> <type> ...
```

Fields are space-separated and some are optional (some types of DNS records may even have additional fields on top of optional ones):

- `<name>` is the domain;
- `<ttl>` stands for time-to-live and dictates the time, in seconds, that a record may be temporarily stored in cache;
- `<class>` refers to the class of DNS record (although, in practice, records always refer to the Internet, `IN`);
- `<type>` refers to the type of DNS record, as described in the following subsections.

2.1.2 Type SOA (start of authority)

A type SOA record stores important information about a domain or zone, such as the master authoritative DNS server for the domain, the email address of the administrator, and a serial number that can be used to detect changes in the zone. Read more about how this data is structured on Wikipedia.

Get the SOA record for the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t soa tecnico.ulisboa.pt
```

2.1.3 Type A (address)

A type A DNS record is a translator that maps a (fully qualified, or canonical) domain to an IPv4 address.

Get the IPv4 address of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t a tecnico.ulisboa.pt
```

It is important to mention that a DNS client may receive multiple records of the same type in the same DNS query response.

Get the IPv4 address of the domain `rtp.pt`.

```
:~$ dig -t a rtp.pt
```

This is useful for client-side load-balancing: the DNS client randomly picks one record, distributing network traffic across multiple servers.

2.1.4 Type AAAA (quad A)

A type AAAA DNS record is similar to a type A DNS record, but maps to an IPv6 address.

Get the IPv6 address of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t aaaa tecnico.ulisboa.pt
```

2.1.5 Type NS (name server)

A type NS DNS record indicates what DNS servers are authoritative for a given domain.

Get the list of authoritative servers of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t ns tecnico.ulisboa.pt
```

2.1.6 Type CNAME (canonical name)

A type CNAME DNS record maps an alias name to a canonical domain, but never to an IP address.

Get the canonical name of the domain `www.ulisboa.pt`.

```
:~$ dig -t cname www.ulisboa.pt
```

2.1.7 Type MX (mail exchange)

A type MX DNS record specifies the mail server responsible for handling email messages of a given domain.

Get the mail servers of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t mx tecnico.ulisboa.pt
```

2.1.8 Questions

Question 1. What DNS servers are authoritative for the domain `tecnico.ulisboa.pt`?

Question 2. Request the type A DNS record for the domain `www.ulisboa.pt`. Why does the «Answer section» contain two records of different types?

Question 3. What is the email address of the administrator of the domain `tecnico.ulisboa.pt`?

Question 4. In the scope of the domain `tecnico.ulisboa.pt`, what servers will be contacted for HTTP(S)? And for SMTP?

2.2 Iterative vs. recursive DNS queries

In this part of the assignment, you will learn the differences between iterative and recursive queries, two strategies used to resolve a domain. You will mainly be looking at the number and content of messages exchanged between the DNS client and the DNS server(s) involved.

> **Hint.** To capture DNS traffic, use Wireshark with an adequate filter (e.g. `dns and udp`). To answer the questions at the bottom of the section, inspect the most relevant packets that you captured (do not overlook the «Additional records» section of some responses).

Before proceeding, try to answer this question: in the previous exercises, how did `dig` know which DNS server(s) to contact if you didn't specify any? The truth is that your operating system keeps a list of DNS servers to reach out to when resolving a domain (on Linux, you can find the list at `/etc/resolv.conf`.) Each server on that list is known as a

resolver, and it is the first stop in the DNS lookup, responsible for handling the client that made the initial query. In other words, for instance, when you `ping google.com`, `ping` asks the first resolver on the list (the resolver is the DNS client) to resolve the domain `google.com`.

The class VM has been configured to use a single DNS resolver that runs locally on the virtual machine: BIND.

> **Note.** BIND is an open source implementation of a DNS server (by default, assume that BIND resolves names in an iterative fashion). For performance and efficiency reasons, DNS responses may be cached for a certain period of time (dictated by the responses's time-to-live, or TTL). A second query for the same domain will, then, be first answered from the cache.

2.2.1 Iterative query

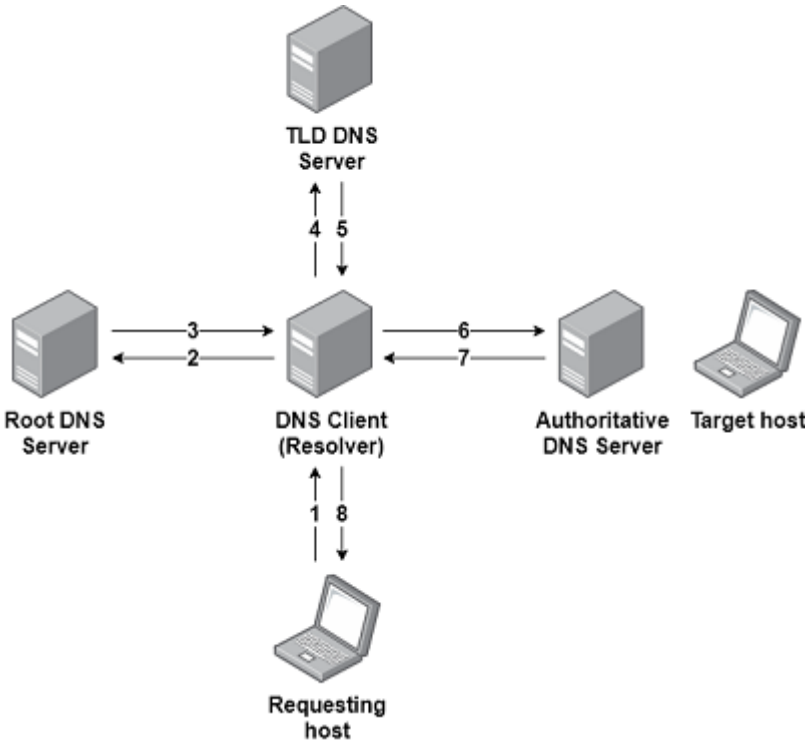

Figure 1: Iterative query

Figure 1 shows an iterative query: to resolve a name, the DNS client communicates directly with each DNS server involved in the lookup (typically, a root DNS server, a top-level domain, TLD, DNS server, and an authoritative DNS server for the domain).

Start a new packet capture and perform an iterative DNS query to resolve the domain `tecnico.ulisboa.pt`.

> **Hint.** To force `dig` to communicate directly with each DNS server in an iterative fashion, use `+trace` (in this particular case, `dig` becomes the DNS client). Moreover, consider using `+nodnssec` (to prevent the tool from requesting DNSSEC records) and `-4` (to use IPv4 query transport only) to improve the readibility of your packet captures.

```
:~$ dig +trace tecnico.ulisboa.pt +nodnssec -4
```

2.2.1.1 Questions

Question 5. How many DNS servers does `dig` contact to resolve the domain? What are their types?

Question 6. Using draw.io, elaborate a diagram, similar to Figure 1, where you include **the most relevant sequence of DNS messages** exchanged between `dig` and the DNS servers that took part in the name resolution. For each DNS message, indicate the type(s) of DNS records involved, and do not forget to mention the type and IPv4 address of each DNS server. To validate your diagram, attach a screenshot with the output of `dig`.

Question 7. How can additional records, contained in some DNS query responses, help reduce query time?

Question 8. Is the last DNS query response authoritative? Show how you found that out.

2.2.2 Recursive query

Figure 2 shows a recursive query: to resolve a name, the DNS client delegates the task to a different DNS server by forwarding the query to that server. In turn, that server may use any strategy for name resolution.

Start a new packet capture and perform a recursive DNS query to resolve the domain `tecnico.ulisboa.pt`.

```
:~$ dig tecnico.ulisboa.pt -4
```

2.2.2.1 Questions

Question 9. This time, how many servers does `dig` contact? And the resolver?

Question 10. Is the last DNS query response authoritative? Show how you found that out.

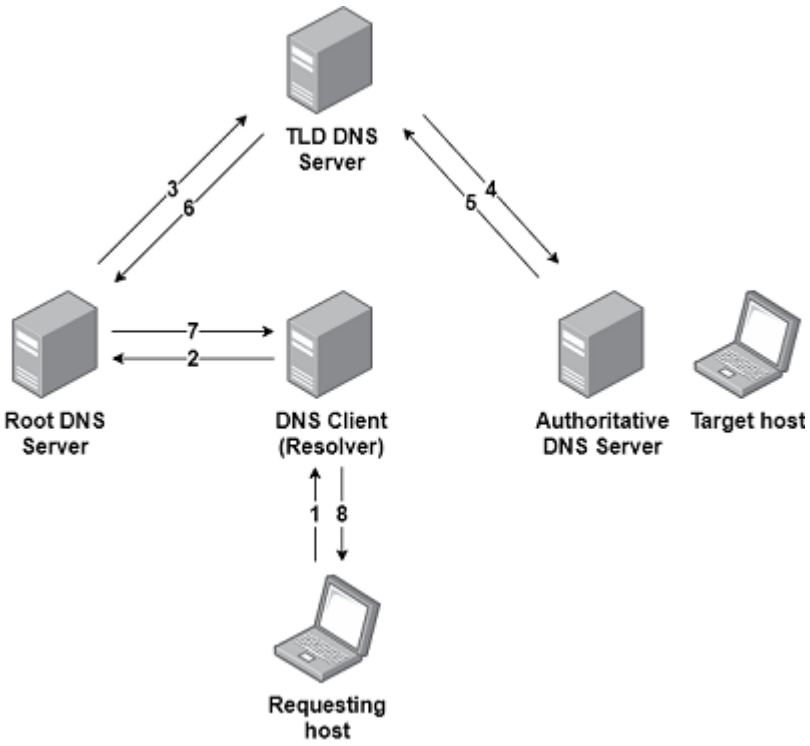

Figure 2: Recursive query

2.3 Caching effects

You will learn how DNS caching can drastically reduce the time required by a DNS client to perform a DNS query.

2.3.1 Perform a DNS query with no caching

For educational purposes, BIND caching mechanism has been disabled in our VM.

Use `dig` to get the IPv4 address of the domain `tecnico.ulisboa.pt`. Repeat the experiment a few times and write down the average query time.

```
:~$ dig -t a tecnico.ulisboa.pt
```

2.3.2 Activate DNS caching

Configuration files for BIND can be found at `/etc/bind`, and `named.conf.options` is the file that hosts all configuration options. In line 9, you can find that the TTL of cache entries has been set to 0 (`max-cache-ttl 0`), i.e. cache entries expire as soon as they are created, disabling the cache.

To enable caching, comment that line using a text editor of your preference, and restart BIND.

Hint. To restart BIND, run `sudo systemctl restart bind9`.

2.3.3 Perform a DNS query with caching

To see caching effects, the cache must be in a clean state. First, you must learn how to inspect the contents of the DNS cache. Know that the cache is held only in memory, which is why it must dumped into a file first.

```
:~$ sudo rndc dumpdb -cache
```

This operation dumps the cache into a new file located at `/var/cache/bind/named_dump.db`. To look for a particular string, you can use `grep`. For instance, if you want to look for a record that contains the domain `tecnico.ulisboa.pt`, you can proceed as follows.

```
:~$ cat /var/cache/bind/named_dump.db | grep tecnico.ulisboa.pt
```

Clear the cache by restarting BIND, repeat the initial DNS query a few times, and write down the average query time (do not consider the first query, which will populate the cache).

2.3.4 Questions

Question 11. Clear the DNS cache and, using `dig`, perform two queries (mind the order!): request the type CNAME DNS record for the domain `www.ulisboa.pt`, and request the type A DNS record for the same domain (write down the query time of this one). Clear the DNS cache one more time, and repeat the second query only. How do you compare query times? Justify your answer.

Question 12. Suppose Instituto Superior Técnico gets a new server with a new IPv4 address. Can administrators shut down the old server immediately? Justify your answer.

Question 13. Is 30 seconds a reasonable value for the TTL of cache entries? Justify your answer.

2.4 Configuring a new DNS zone

A DNS zone is a portion of the DNS namespace delegated to a specific entity, like a person or an organization. DNS zones are described by text files, known as *zone files*, that include a collection of DNS records of various types. In BIND, zone files are declared in a special file, located at `/etc/bind/named.conf.local`, and its contents are fetched every time BIND starts.

Hypothetically, consider that the TLD `.rc.leic` is added to the DNS namespace (the zone file is located at `/etc/bind/db.rc.leic`). As an administrator, you bought a server to offer a new service to your clients, and you want DNS to resolve the domain `<id>.rc.leic` (`<id>` is your IST ID) to your server's IPv4 address.

Without delving into details, you have to:

- Create and declare the zone file for your new domain `<id>.rc.leic`;
- Update the zone file for the TLD `.rc.leic` to include the authoritative servers for your domain.

Bear in mind that you must restart BIND every time you create or modify a zone. Before answering the questions at the bottom of the section, go through the next subsections to learn more.

2.4.1 Declare a DNS zone

To declare a new zone, you must add a new entry to `/etc/bind/named.conf.local`. A zone declaration looks like this.

```
zone "rc.leic" {
  type master;
  file "/etc/bind/db.rc.leic";
}
```

In this example, the zone file for `rc.leic` is located at `/etc/bind/db.rc.leic`.

2.4.2 Structure of zone files

Zone files consist of directives (a sort of variables) and DNS records. To help you understand better, take a look at the beginning of the zone file for the TLD `.rc.leic`.

```
$ORIGIN     rc.leic.        ; The domain
$TTL        604800            ; Default TTL (in seconds)

@   IN  SOA  ns1.rc.leic. admin.rc.leic. (
                202111284   ; Serial (today's date + today's serial)
                   604800   ; Refresh (in seconds)
                    86400   ; Retry (in seconds)
                  2419200   ; Expire (in seconds)
                   604800 ) ; Negative Cache TTL (in seconds)
```

For instance, `$ORIGIN` and `$TTL` are directives that define, respectively, the domain the zone file refers to, and the default TTL for all DNS records in the zone.

Next, you have a type SOA DNS record that specifies, among other things:

- `ns1.rc.leic` as the master authoritative DNS server for the domain `rc.leic`;
- `admin@rc.leic` as the email address of the administrator (in this syntax, the first `.` is replaced by a `@`).

Keep looking at the contents of the zone file to find other types of DNS records.

2.4.3 Checking zone files for errors

To validate the syntax and integrity of zone files, you can use `named-checkzone`, a tool that takes the zone name and the path to the zone file as arguments. Invoke it as follows.

```
:~$ named-checkzone rc.leic \
  /etc/bind/db.rc.leic
```

```
zone rc.leic/IN: loaded serial 202111281
OK
```

The tool will raise an error if the zone file is misconfigured, and provide a hint on how to fix the issue.

2.4.4 Questions

Question 14. Configure a new DNS zone for your domain following the steps from above. Your domain should have two authoritative DNS servers (`ns1.<id>.rc.leic` and `ns2.<id>.rc.leic`), one mail server (`mail.<id>.rc.leic`), and support the alias `www.<id>.rc.leic` that maps to `<id>.rc.leic` – assume that everything happens locally, i.e. the IPv4 addresses of all servers involded must be `127.0.0.1`. Include in your answer all configuration files that you consider relevant.

To validate your configuration, you can find the expected output of some commands below.

```
:~$ dig -t ns www.<id>.rc.leic
```

```
...
;; ANSWER SECTION:
www.<id>.rc.leic.   604800  IN      CNAME   <id>.rc.leic.
<id>.rc.leic.       604800  IN      NS      ns2.<id>.rc.leic.
<id>.rc.leic.       604800  IN      NS      ns1.<id>.rc.leic.

;; ADDITIONAL SECTION:
ns1.<id>.rc.leic.   604800  IN      A       127.0.0.1
ns2.<id>.rc.leic.   604800  IN      A       127.0.0.1
...
```

Computer Networks — 2021/22		Assignment:	Lab 4
Understanding and configuring DNS		Issued:	2021-12-09
Domain Name System		Submission Due:	2021-12-12
Authors:	João M. Tiago Prof. Luis D. Pedrosa	Comments Due:	2021-12-16
		Version:	1.0

```
:~$ dig -t mx www.<id>.rc.leic
```

```
...
;; ANSWER SECTION:
www.<id>.rc.leic.  604800  IN      CNAME   <id>.rc.leic.
<id>.rc.leic.      604800  IN      MX      10 mail.<id>.rc.leic.

;; ADDITIONAL SECTION:
mail.<id>.rc.leic. 604800  IN      A       127.0.0.1
...
```

```
:~$ ping www.<id>.rc.leic
```

```
PING <id>.rc.leic (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 ...
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 ...
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 ...
...
```

Question 15. Suppose you request a type A DNS record for the fictional domain `example.rc.leic`. The «Answer section» is empty, but the «Authority section» is not. Why?