Lab4: The Domain Name System - Solutions

COM-208: Computer Networks

The goal of this lab is to get a sense of how the Domain Name System (DNS) essentially enables Internet operation.

The DNS protocol

The `dig` utility relies on the DNS protocol to provide information related to DNS names and IP addresses. It is similar to the `host` utility (that you used in Lab1), but provides more detailed information.

Start a Wireshak capture and configure the filter to catch DNS messages. Run `dig adelaide.edu.au`. Stop the capture and answer the following questions:

- Based on the captured packets, what transport-layer protocol does DNS use? what port number is associated with the DNS protocol?

> UDP.
>
> The DNS protocol uses port number `53`. You can verify this by looking at the Wireshark traces. DNS query messages have port number 53 as their destination port, and DNS response messages have port number 53 as their source port.

- Why do you think that DNS uses this transport-layer protocol? Summarize the advantages and disadvantages of this choice.

> The DNS query and DNS response messages are short and they can fit in one IP packet. By using UDP, we avoid the connection setup overhead of TCP, which involves a handshake that requires an extra round-trip time (RTT).

The disadvantage is that UDP does not provide reliable delivery of messages, thus DNS clients must keep track of the requests that were sent and wait for the replies. If the client receives no response within a particular amount of time, the client must either retransmit the original request, or time out and return an error.

For DNS, decreasing request latency to improve performance is considered an acceptable trade-off for the added complexity of having to deal with packet loss at the application layer.

DNS resource records, questions and answers

DNS servers store information in the form of DNS **resource records** (RRs), of different types. DNS clients and servers generate DNS **questions** (or "requests" or "queries"), while DNS servers provide DNS **answers** (or "responses") that contain RRs. A DNS message may carry multiple questions and/or answers.

- What kind of information do the following RR types provide: A, CNAME, PTR, MX, NS, and SOA? You can find the answer on Wikipedia and/or RFC1033 (just google it, and you will see what that is).

A = address record: stores the IP address for a hostname.

CNAME = canonical name record: a machine may have several names (aliases) associated with it; the CNAME record points from the aliases to the "main" one. The resolver would then query for the A record of the canonical name (but in practice the A record is usually returned together with the CNAME record to make the query faster). Alternatively, the domain's administrator could just create an A record for each alias, but then would have to make sure they are all modified consistently whenever the IP address is changed.

PTR = pointer record; stores the canonical name for an IP address.

MX = mail exchange record: stores the hostname of the e-mail server for the domain. These are servers that accept messages via SMTP.

NS = indicates the hostname of the nameservers that are responsible (authoritative) for the domain.

SOA = start-of-authority record: stores various administrative information about a domain: the name of the primary authoritative nameserver, the e-mail address of the administrator (note that the @ sign is replaced

with a dot), the serial number of the configuration file, how often the secondary authoritative nameservers should synchronize with the primary etc.

More types of DNS records, and corresponding details can be found on this Wikipedia Link.

- What is the IP address of `epfl.ch`? Which RR type stores the information needed to answer this question?

```
user@host:~$ dig epfl.ch

; <<>> DiG 9.10.6 <<>> epfl.ch
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26960
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL
    ↪ : 1

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

;; ANSWER SECTION:
epfl.ch.       42517432 IN A   128.178.222.108

;; Query time: 119 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Sep 28 00:36:27 CEST 2020
;; MSG SIZE  rcvd: 52
```

The IP address of epfl.ch is `128.178.222.108`. To get this answer we make a type A query. We can see this in both the dig output and in Wireshark. (use the filter "DNS" to show only DNS messages, then click on a query packet and show Domain Name System (query) > Queries > epfl.ch: Type A, class IN).

- What is the DNS name associated with IP address `128.178.222.108`? Which RR type stores the information needed to answer this question?

```
user@host:~$ dig -x 128.178.222.108

; <<>> DiG 9.10.6 <<>> -x 128.178.222.108
```

```
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52110
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL
    ↪ : 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;108.222.178.128.in-addr.arpa. IN  PTR

;; ANSWER SECTION:
108.222.178.128.in-addr.arpa. 86400 IN  PTR www-origin.epfl.ch.

;; AUTHORITY SECTION:
178.128.in-addr.arpa.  85661   IN  NS  stisun1.epfl.ch.
178.128.in-addr.arpa.  85661   IN  NS  stisun2.epfl.ch.
178.128.in-addr.arpa.  85661   IN  NS  scsnms.switch.ch.

;; ADDITIONAL SECTION:
scsnms.switch.ch.  9640    IN  A   130.59.31.26
scsnms.switch.ch.  227 IN  AAAA    2001:620:0:ff::a7
stisun1.epfl.ch.   73216   IN  A   128.178.15.8
stisun1.epfl.ch.   77086   IN  AAAA    2001:620:618:10f:1:80b2:
    ↪ f08:1
stisun2.epfl.ch.   73216   IN  A   128.178.15.7
stisun2.epfl.ch.   77086   IN  AAAA    2001:620:618:10f:1:80b2:
    ↪ f07:1

;; Query time: 45 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Sep 28 00:39:59 CEST 2020
;; MSG SIZE  rcvd: 293
```

The DNS name associated with `128.178.222.108` is `www-origin.epfl.ch`. We made a type PTR query for the name `108.222.178.128.in-addr.arpa`.

Authoritative and local DNS servers

Each lower-level domain, e.g., epfl.ch, has a set of **authoritative DNS servers**, which store all the latest information that the DNS system has about this domain.

When a DNS server provides a DNS answer that concerns a domain for which the server is authoritative, we say that the answer itself is **authoritative**.

- Which are the authoritative DNS servers for epfl.ch? What RR type stores the

information needed to answer this question?

```
user@host:~$ dig epfl.ch NS

; <<>> DiG 9.10.6 <<>> epfl.ch NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52162
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL
    ↪ : 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;epfl.ch.          IN  NS

;; ANSWER SECTION:
epfl.ch.       72075   IN  NS  stisun1.epfl.ch.
epfl.ch.       72075   IN  NS  stisun2.epfl.ch.

;; ADDITIONAL SECTION:
stisun1.epfl.ch.   72075   IN  A   128.178.15.8
stisun1.epfl.ch.   75945   IN  AAAA    2001:620:618:10f:1:80b2:
    ↪ f08:1
stisun2.epfl.ch.   72075   IN  A   128.178.15.7
stisun2.epfl.ch.   75945   IN  AAAA    2001:620:618:10f:1:80b2:
    ↪ f07:1

;; Query time: 493 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Sep 28 00:59:00 CEST 2020
;; MSG SIZE  rcvd: 168
```

The authoritative DNS servers for EPFL are `stisun1.epfl.ch` and `stisun2.epfl.ch`, and the type of resource records is `NS` as show in the answer section of the `dig` result above.

Your computer (like any Internet end-system in the world) knows the IP address(es) of one or more **local DNS servers**. When a DNS client process running in the application layer of your computer (e.g., `dig`) needs information from the DNS system, it sends a DNS question to one of these local DNS servers.

- Look carefully at the answers provided by `dig` so far. Can you identify in them the IP address of the local DNS server used by your computer? Are you using one of the authoritative DNS servers for epfl.ch as your local DNS server?

This is shown by dig in the output for any query that does not use a user-specified nameserver:

```
user@host:~$ dig epfl.ch

; <<>> DiG 9.10.6 <<>> epfl.ch
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26960
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL
    ↪ : 1

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

;; ANSWER SECTION:
epfl.ch.                42517432 IN     A       128.178.222.108

;; Query time: 119 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Sep 28 00:36:27 CEST 2020
;; MSG SIZE  rcvd: 52
```

In the above `dig` output, the IP address of local nameserver is `192.168.1.1`, and the port used is `53`, which is the default port for DNS.

The answer can also be found in the file /etc/resolv.conf. There are more entries in this file:

nameserver 192.168.1.1

A DNS client can send a DNS message to any DNS server in the world—it is not obligated to contact only the local DNS servers. For example, if you run `dig @*IP address* ...` then `dig` will send its DNS question to the DNS server that has the specified IP address.

- Ask the DNS server with IP address `8.8.8.8` for the mail servers that serve the epfl.ch domain. Did you get an authoritative answer?

```
user@host:~$ dig @8.8.8.8 epfl.ch MX

; <<>> DiG 9.10.6 <<>> @8.8.8.8 epfl.ch MX
; (1 server found)
;; global options: +cmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7905
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL
    ↪ : 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;epfl.ch.          IN  MX

;; ANSWER SECTION:
epfl.ch.       21567   IN  MX  50 mx3.epfl.ch.
epfl.ch.       21567   IN  MX  50 mx2.epfl.ch.
epfl.ch.       21567   IN  MX  50 mx1.epfl.ch.

;; Query time: 229 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Sep 28 01:01:54 CEST 2020
;; MSG SIZE  rcvd: 96
```

We see that Google's server cannot give us an authoritative answer, since the flags do not contain `aa`. This is because it does not have authority for the EPFL domain.

- What do you need to do to get an authoritative answer to your question?

To get an authoritative answer we can just query one of the authoritative nameservers for the domain epfl.ch:

```
user@host:~$ dig @stisun1.epfl.ch epfl.ch MX

; <<>> DiG 9.10.6 <<>> @stisun1.epfl.ch epfl.ch MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26082
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL
    ↪ : 11
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;epfl.ch.                       IN      MX

;; ANSWER SECTION:
epfl.ch.                86400   IN      MX      50 mx1.epfl.ch.
```

```
epfl.ch.                86400   IN      MX      50 mx3.epfl.ch.
epfl.ch.                86400   IN      MX      50 mx2.epfl.ch.

;; AUTHORITY SECTION:
epfl.ch.                86400   IN      NS      stisun1.epfl.ch.
epfl.ch.                86400   IN      NS      stisun2.epfl.ch.

;; ADDITIONAL SECTION:
mx1.epfl.ch.            86400   IN      A       128.178.166.20
mx2.epfl.ch.            86400   IN      A       128.178.166.21
mx3.epfl.ch.            86400   IN      A       128.178.166.22
stisun1.epfl.ch.        86400   IN      A       128.178.15.8
stisun2.epfl.ch.        86400   IN      A       128.178.15.7
mx1.epfl.ch.            86400   IN      AAAA    2001:620:618:1a6
    ↪ :1:80b2:a614:1
mx2.epfl.ch.            86400   IN      AAAA    2001:620:618:1a6
    ↪ :1:80b2:a615:1
mx3.epfl.ch.            86400   IN      AAAA    2001:620:618:1a6
    ↪ :1:80b2:a616:1
stisun1.epfl.ch.        86400   IN      AAAA    2001:620:618:10f
    ↪ :1:80b2:f08:1
stisun2.epfl.ch.        86400   IN      AAAA    2001:620:618:10f
    ↪ :1:80b2:f07:1

;; Query time: 73 msec
;; SERVER: 128.178.15.8#53(128.178.15.8)
;; WHEN: Mon Sep 28 01:10:46 CEST 2020
;; MSG SIZE  rcvd: 360
```

DNS caching and time-to-live (TTL)

DNS clients and servers – at all levels of the DNS hierarchy – **cache** the RRs they receive. To prevent inconsistency between authoritative and cached RRs, each RR is associated with a **time to live** (TTL), which indicates until when the RR is expected to be valid, hence until when it can be safely cached.

Imagine that the EPFL sysadmins need to urgently change the names of the mail servers that serve epfl.ch. Hence, they login to the authoritative DNS servers for epfl.ch and change the RR that specifies the mail-server names, before the RR's TTL has expired.

- What will happen now if a DNS client asks `8.8.8.8` for the mail servers that serve epfl.ch? How long will it take until `8.8.8.8` can answer this question correctly?
- What could the EPFL sysadmins do to make the change as quickly as possible without causing any inconsistency in the DNS system?

a) Since we query the DNS server of Google (`8.8.8.8`), the answer is not authoritative, thus not guaranteed to be correct. This is because Google may already have the old record in its cache; until the record expires (remaining TTL value), it keeps serving the old value without knowing that the value is incorrect.

b) From the query results (`dig @stisun1.epfl.ch epfl.ch MX`), we can see that TTL value EPFL RRs are set to 86400 seconds, that is, 24 hours or 1 day. In the worst case scenario, where a DNS server would cache the EPFL RR just before the configuration change, it will take 24 hours for that cache to invalidate.

In order for EPFL sysadmins to make change as quickly as possible, they should inspect the TTL of the MX records, in this case 24 hours. We change it to a small value (e.g. 1 second or even zero). Then we wait for 24 hours until the new record propagates to all servers that may have cached the old one. Now we can change the content of the MX record and set the TTL back to the old value. The downside is that for 24 hours EPFL's DNS server will get much higher DNS traffic due to those queries (since the caches of all the other resolvers on the Internet expire quickly). To avoid that, we can make the TTL change in 2 phases: first we change it to a few minutes, and after 6 hours we change it to 1 or 0 seconds. In this case we will get a very high volume of traffic only for a few minutes.

Iterative DNS queries

There are two ways to resolve a DNS query: **recursively** and **iteratively**. They differ in what a DNS server does when it receives the query but does not know the answer:

If the query is resolved recursively, the DNS server asks another DNS server that may know the answer; so, a root server asks a TLD server, and a TLD server asks an authoritative server.

If a query is resolved iteratively, the DNS server returns the IP address of another DNS server that may know the answer; so, a root server returns the IP address of a TLD server, and a TLD server returns the IP address of an authoritative server.

You will now pretend your computer is a local DNS server that is resolving a query iteratively. Your goal is to find the IP address of `aladdin.planetlab.extranet.uni-passau` ↪ `.de.` The root DNS server that you will use is `a.root-servers.net (198.41.0.4)`. If you run `dig +norecurse ...`, `dig` will ask the DNS server that it contacts to not recurse.

- How many DNS servers do you think you will have to ask before you get the final

answer?

- Ask the root DNS server for the target IP address. Based on the answer, decide which DNS server you need to ask next. Coninue asking until you get the final answer.
- How many DNS servers did you end up asking? Is this the number you expected?

a) We have to query the root server to find the DE TLD server, then the TLD server to find the DNS server of uni- passau.de. Afterwards we have to query at least the DNS server of uni-passau.de; whether we have to make more queries depends on the hierarchy of the DNS servers of uni-passau.de.

Note: naively, since the queries are non-recursive, we might think that we must send one message for each part of the name (root server, de TLD, uni-passau.de, extranet.uni-passau.de, planetlab.extranet.uni-passau.de and aladdin.planetlab.extranet.uni- passau.de) for a total of 5 queries. This is actually a good heuristic but we will see it is not always the case.

b)

```
user@host:~$ dig @198.41.0.4 aladdin.planetlab.extranet.uni-
    ↪ passau.de +norecurse

; <<>> DiG 9.10.6 <<>> @198.41.0.4 aladdin.planetlab.extranet.uni
    ↪ -passau.de +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46994
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 13

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;aladdin.planetlab.extranet.uni-passau.de. IN A

;; AUTHORITY SECTION:
de.             172800  IN  NS  a.nic.de.
de.             172800  IN  NS  f.nic.de.
de.             172800  IN  NS  l.de.net.
de.             172800  IN  NS  n.de.net.
de.             172800  IN  NS  s.de.net.
de.             172800  IN  NS  z.nic.de.

;; ADDITIONAL SECTION:
a.nic.de.       172800  IN  A   194.0.0.53
```

```
f.nic.de.       172800  IN  A   81.91.164.5
l.de.net.       172800  IN  A   77.67.63.105
n.de.net.       172800  IN  A   194.146.107.6
s.de.net.       172800  IN  A   195.243.137.26
z.nic.de.       172800  IN  A   194.246.96.1
a.nic.de.       172800  IN  AAAA    2001:678:2::53
f.nic.de.       172800  IN  AAAA    2a02:568:0:2::53
l.de.net.       172800  IN  AAAA    2001:668:1f:11::105
n.de.net.       172800  IN  AAAA    2001:67c:1011:1::53
s.de.net.       172800  IN  AAAA    2003:8:14::53
z.nic.de.       172800  IN  AAAA    2a02:568:fe02::de

;; Query time: 22 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Mon Oct 05 17:42:09 CEST 2020
;; MSG SIZE  rcvd: 439
```

Note that although we have queried the root server with an A query for aladdin.planetlab.extranet.uni-passau.de, the root server does not have the answer. Instead, it returns the information required to query the DE top-level domain server: NS records of the TLD server with the names of the server followed by the A records with the IP addresses for these names. We choose one, a.nic.de, to continue the query.

c) Query `a.nic.de`

```
user@host:~$ dig @194.0.0.53 aladdin.planetlab.extranet.uni-
    ↪ passau.de +norecurse

; <<>> DiG 9.10.6 <<>> @194.0.0.53 aladdin.planetlab.extranet.uni
    ↪ -passau.de +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7258
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;aladdin.planetlab.extranet.uni-passau.de. IN A

;; AUTHORITY SECTION:
uni-passau.de.      86400   IN  NS  ns.rz.uni-passau.de.
uni-passau.de.      86400   IN  NS  dns-1.dfn.de.
uni-passau.de.      86400   IN  NS  ns.forwiss.uni-passau.de.
uni-passau.de.      86400   IN  NS  ns.fim.uni-passau.de.
```

```
;; ADDITIONAL SECTION:
ns.rz.uni-passau.de.   86400   IN  A   132.231.51.4
ns.fim.uni-passau.de.  86400   IN  A   132.231.1.24
ns.forwiss.uni-passau.de. 86400    IN  A   132.231.20.100

;; Query time: 29 msec
;; SERVER: 194.0.0.53#53(194.0.0.53)
;; WHEN: Mon Oct 05 17:53:34 CEST 2020
;; MSG SIZE  rcvd: 207
```

Query `ns.rz.uni-passau.de.`

```
user@host:~$ dig @132.231.51.4 aladdin.planetlab.extranet.uni-
    ↪ passau.de +norecurse

; <<>> DiG 9.10.6 <<>> @132.231.51.4 aladdin.planetlab.extranet.
    ↪ uni-passau.de +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57459
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;aladdin.planetlab.extranet.uni-passau.de. IN A

;; ANSWER SECTION:
aladdin.planetlab.extranet.uni-passau.de. 7200 IN A 195.37.16.121

;; Query time: 32 msec
;; SERVER: 132.231.51.4#53(132.231.51.4)
;; WHEN: Mon Oct 05 17:56:23 CEST 2020
;; MSG SIZE  rcvd: 85
```

Finally, resolved A record for the request query.

d) To get an answer, we had to ask only 3 DNS servers. Note that although recursion was forbidden, ns.rz.uni-passau.de. had the answer already cached, and replied directly. This has lead to a number queries suprisingly small given that the request name has many parts.