Lab4: The Domain Name System

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

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

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?

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?

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

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?

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?