CS-202 Exercises on web & DNS (L10 - L11)

Before You start

In today's exercise session, you will get a sense of what happens when you type a URL in your web browser. More specifically, you will learn about the application layer protocols involved: DNS and HTTP.

When you type in a URL, your web browser connects to the web server that stores the base file of the webpage you are trying to access. The two (web browser and web server) use the HTTP protocol to exchange messages. The web browser sends an HTTP GET request and the web server replies with (one or more) HTTP response message(s) that contains the requested file.

But before your machine can connect to the web server, it needs to know the IP address of that server. For this, it uses the Domain Name System (DNS).

To find the IP address of the target web server, the web browser extracts the DNS name from the URL and asks the DNS client for the corresponding IP address. The DNS client, running on your machine, then queries the local DNS server for the domain's IP address.

The DNS hierarchy

Every network has a local DNS server that your DNS client can send queries to. However, the local DNS does not always have an answer. In such cases, the local DNS asks other DNS servers, according to a DNS hierarchy that consists of three kinds of DNS servers:

- A root server knows the IP address of at least one (typically several) top-level- domain (TLD) servers for each TLD.
- A TLD server knows the IP address of at least one (typically several) authoritative servers for each domain that falls under its TLD.
- An authoritative server knows the IP address of every DNS name that falls under its domain

There are two ways in which the local DNS resolves the DNS query: **recursively** and **iteratively**. The two differ how a DNS server react when it receives the query but does not know the answer:

- If the query is resolved **recursively**: the DNS servers (of all levels) talk with each other to get the answer. It goes in the following order:
    1. The local DNS server talks with a root server.
    2. The root server talks with a TLD server.
    3. And the TLD server talks with an authoritative server.
- If a query is resolved **iteratively**: the local DNS server handles asking the other DNS servers to get the final answer. The order goes like this:
    1. The local DNS server asks the root server for the IP of *a* TLD server.
    2. The local DNS server asks the TLD server for the IP of *an* authoritative server.
    3. The local DNS server asks the authoritative server for the final answer

Paper and pencil exercises: DNS and HTTP

In these exercises assume that DNS and web-browser caches are initially empty. In some problems, you will be asked to fill in a table, stating all the messages that were transmitted or received as a result of some action. For each message, briefly describe the goal, e.g., is this message an HTTP GET request for a particular URL? Is it a DNS request for the IP address of a particular DNS name?

Exercise 1: DNS and HTTP message exchange [Basic]

You are working on an EPFL computer called workstation.epfl.ch. Your local DNS server is ns.epfl.ch. This DNS server knows the IP address of root server a.root-servers.net, which knows the IP address of .ch TLD server a.nic.ch, which knows the IP address of epfl.ch authoritative server ns.epfl.ch and unil.ch authoritative server ns.unil.ch. All these DNS servers perform *iterative* requests. Table 1 shows information about all the servers involved in this problem.

Server	DNS name	IP address
Root DNS server	a.root-servers.net	1.1.1.1
.ch TLD DNS server	a.nic.ch	2.2.2.2
EPFL DNS server	ns.epfl.ch	3.3.3.3
UNIL DNS server	ns.unil.ch	4.4.4.4
EPFL workstation	workstation.epfl.ch	5.5.5.5
UNIL web server	www.unil.ch	6.6.6.6

Table 1: Server DNS names and IP addresses.

- You open your web browser and type in http://www.unil.ch/index.html. This URL's base file does not reference any other URLs. In Table 2a, list all the DNS and HTTP messages that get transmitted as a result of your action.

Message	Source	Destination	Protocol	Purpose
1	5.5.5.5	3.3.3.3	DNS	query: A for www.unil.ch
2				
3				
4				
5				
6				
7				
8				
9				
10				

Table 2a: Transmitted DNS and HTTP messages.

- Immediately after retrieving this URL, you type in http://www.unil.ch/logo.png. In Table 2b, list all the DNS and HTTP messages that get transmitted as a result of your action.

Packet	Source	Destination	Application protocol	Purpose
11				
12				

Table 2b: Transmitted DNS and HTTP messages.

Exercise 2: Adding referenced files [Basic]

Now, suppose instead that http://www.unil.ch/index.html is an HTML page that references two image files: http://www.unil.ch/logo.png and http://www.unil.ch/banner.jpg.

- You open your web browser and type in http://www.unil.ch/index.html. In Table 3, list all the DNS and HTTP messages that would be transmitted as a result of your action. Assume the same DNS servers are involved as listed in Table 1 and all of these perform iterative requests.

Message	Source	Destination	Protocol	Purpose
1	5.5.5.5	3.3.3.3	DNS	query: A for www.unil.ch

Table 3: Transmitted DNS and HTTP messages.

Exercise 3: Adding a security twist [Advanced]

Three users, Alice, Bob, and Persa, are logged into their computers, all located inside ETHZ's network.

ETHZ has a web server www.ethz.ch and local DNS server ns.ethz.ch, which is also the authoritative server for the ethz.ch domain.

EPFL has web server www.epfl.ch and local DNS server ns.epfl.ch, which is also the authoritative server for the epfl.ch domain.

All DNS servers perform *recursive* requests.

Figure 2 illustrates the setup for this problem.

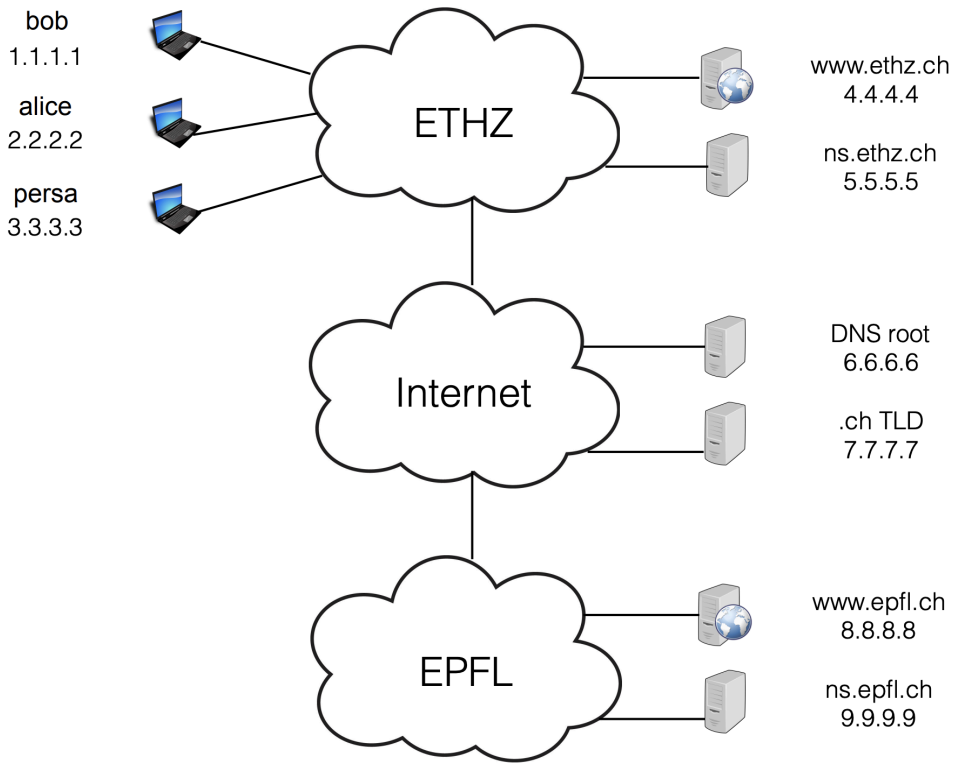

Figure 2: Question Setup

- Alice types in her web browser http://www.epfl.ch/index.html. This URL's base file references two other URLs: http://www.epfl.ch/image.jpg and http://www.ethz.ch/file.html (which does not reference any other URL).

  In Table 4, list all the application-layer HTTP and DNS packets that are transmitted as a result of this action.

Message	Source IP	Dest. IP	Protocol	Purpose
ex.1	1.0.0.1	1.0.0.2	HTTP	HTTP reply with image.jpg

Table 4: Transmitted packets.

- After Alice has retrieved http://www.epfl.ch/index.html, Bob wants to access the same URL.

  Persa is a malicious user who guesses exactly when Bob tries to access http://www.epfl.ch/index.html. She wants to trick Bob and make him access a web server running on her own computer, thinking that he is accessing the EPFL web server.

  How can Persa do that by sending DNS traffic to Bob?

Mini-lab: IP addresses and process names

Exercise 4: Find your own IP address using the ifconfig utility [Basic]

Every computer in the world has at least one **network interface**. Whenever an entity outside the computer wants to communicate with the computer, it needs to name one of its network interfaces. Different entities use different names to refer to a computer's network interface: the network layer uses IP addresses, the link layer uses MAC addresses, the computer's operating system (OS) uses local interface names.

The **ifconfig** utility lists a computer's network interfaces and displays or updates their configuration.

Type ifconfig in a terminal command line and answer the following questions:

- How many active network interfaces does your computer have? (Hint: look at 'status' under your interfaces, usually named something like eth0, en0, or wlan0.)
- What is the IP address of each active interface? (Hint: Look for something like inet or inet6)
- Can you guess why it has more than one?

If you are curious and want to learn more about a command, you can use your environment's man pages (e.g. if you type man ifconfig in the command line, it will display everything you could possibly want to know about the ifconfig command), or you can turn to online resources (e.g. Wikipedia).

Exercise 5: See local and remote addresses using the netstat utility [Basic]

The **netstat** utility displays the contents of various network-related data structures that are stored in your computer. E.g., if you type netstat -t in the command line, that will display the list of "communication sessions" that are active between your computer and remote computers.

- The "Local Address" column lists processes that are running in the application layer of your computer. Notice that the names of all (or most of) these processes share a common prefix. Why is that? What does this prefix correspond to?
- The "Foreign Address" column lists all the processes that are running in the application layer of a remote computer that your computer is communicating with. Login to Moodle in your browser and then run the netstat utility. Can you tell which foreign address(es) correspond to EPFL server(s)?

Mini-lab: Playing with DNS

The dig utility

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 Exercise Session 1), but provides more detailed information.

Run "dig adelaide.edu.au" and answer the following questions:

- What is the primary IP address (A record) associated with adelaide.edu.au according to the output?
- What DNS server provided the response to your query, and how long did the query take to complete?

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

- 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 (or you can just google it, and you will see what that is).

Now you know the kind of information different RR types provide. Use this information to answer the next parts of this lab exercise.

Exercise 6: DNS Lookup [Basic]

- 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 the IP address obtained in the previous question? Which RR type stores the information needed to answer this question?

  To do reverse lookup, use the '-x' option; you can view more details about the option using "man dig".

Exercise 7: Authoritative and local DNS servers [Basic]

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 query 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. If you run: "dig @<IP address> …" then dig will send its DNS query 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? Hint: look at the HEADER flags.

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

Exercise 8: DNS caching and time-to-live (TTL) [Advanced]

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 should be 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?

Mini-lab: Cookies [Optional]

Cookies enable a web server to **link subsequent HTTP requests** to the same web browser: if you send 10 HTTP GET requests, for 10 different resources, to the same web server, the web server can use cookies to figure out that these 10 requests came from the same web browser, even if you did not explicitly provide any identification information (e.g., you did not login).

Before you start, figure out how to control cookie settings in your browser. In Firefox:

- To view or delete the cookies that have been stored on your computer: ≡ → Settings → Privacy & Security → Cookies and Site Data → Manage Data or Clear Data...
- To view all the cookies stored due to visiting the web page: Go to Storage from the top panel, and then select Cookies.

Let us see cookies in action:

- Allow your browser to exchange cookies. Delete existing cookies. Open Moodle. Did the EPFL web server send you any cookies? And are they all from the same domain?
- Login to your moodle account. Restart your web browser and re-open Moodle. Does it ask you to login again? Explain your browser's behavior.
- Delete existing cookies. Restart your web browser and re-open Moodle. Does it ask you to login again? Explain your browser's behavior.