Homework2: The Interplay between Web and DNS

COM-208: Computer Networks

For most of us, the most common operation we do with our computer is type a URL in the web browser. In this homework, you will explore the underlying messages that this operation triggers.

Whenever you start a new problem, assume that all the DNS, web-browser, and web-server-proxy caches are initially empty.

Also, assume that web browsers and web servers communicate over *persistent* TCP connections, i.e., they use the same TCP connection to exchange multiple HTTP messages (so that they do not have to pay the TCP connection-setup cost for every new HTTP message they exchange).

In most 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?

Before you start, a quick recap:

When a process running in the application layer of your computer wants to communicate with another, remote process, it must form the remote process's name; for that, it needs to know the IP address of the network interface behind which the remote process is running.

For example, when you type a URL in your web browser, the latter must form the process name of the web server that stores that URL; for that, it needs to know the IP address of the network interface behind which the web server is running.

To form the process name of the target web server, the web browser extracts the DNS name from the URL and asks a DNS client for the corresponding IP address; the DNS client then asks a local DNS server.

If the local DNS server does not know the answer, it asks another DNS server, 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.

DNS clients and servers cache the DNS answers they receive, so that they do not need to send the same DNS questions multiple times. DNS caching significantly affects the amount of DNS traffic on the Internet.

DNS and HTTP messages

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 `www.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 2, list all the DNS and HTTP messages that get transmitted as a result of your action.

Packet	Source	Destination	Application protocol	Message
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 2: Transmitted DNS and HTTP messages.

- Immediately after retrieving this URL, you type in `http://www.unil.ch/logo.png`. In Table 3, list all the DNS and HTTP messages that get transmitted as a result of your action.

Packet	Source	Destination	Application protocol	Message
11				
12				

Table 3: Transmitted DNS and HTTP messages.

Adding a proxy web server

Two users are logged into their respective computers, `user1.epfl.ch` and `user2.epfl.ch`, both located inside the EPFL network. Each user's web browser uses a single persistent TCP connection to communicate with a web server.

EPFL has local DNS server `ns.epfl.ch`, web server `www.epfl.ch`, and proxy web server `proxy.epfl.ch`.

All the computers inside the EPFL network use `ns.epfl.ch` as their local DNS server, which is also the authoritative DNS server for the `epfl.ch` domain. All DNS servers perform *iterative* requests.

- User1 types `http://www.epfl.ch/index.html` in her web browser. This URL's base file references only one other URL, `http://www.epfl.ch/image.png`. User1's web browser does not use any proxy web server.

  In Table 4, list all the application-layer messages and TCP connection-setup packets that are transmitted as a result of this action.

Packet	Source	Destination	Transport protocol	Application protocol	Message
1					
2					

Table 4: Transmitted messages and connection-setup packets.

- User2 types in his web browser `http://www.epfl.ch/help.html`. This URL's base file does not reference any other URLs. User2's web browser uses the EPFL proxy web server.

  In Table 5, list all the application-layer messages and connection setup packets that are transmitted as a result of this action.

Packet	Source	Destination	Transport protocol	Application protocol	Message
1					
2					

Table 5: Transmitted messages and connection-setup packets.

Adding a security twist

Three users, Alice, Bob, and Persa, are logged into their computers, respectively called `alice.ethz.ch`, `bob.ethz.ch`, and `persa.ethz.ch`, all located inside ETHZ's network.

ETHZ has 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.

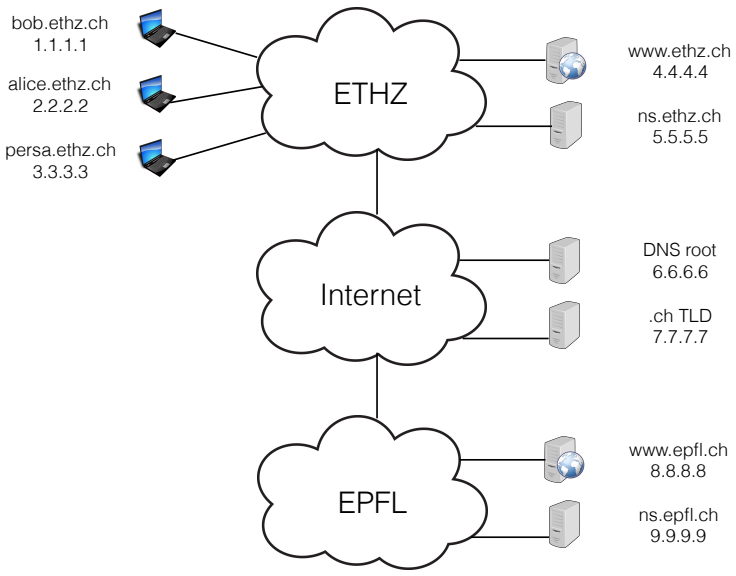

Figure 2: Question Setup

- 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 6, list all the application-layer messages and connection-setup packets that are transmitted as a result of this action.

Packet	Source IP	Dest. IP	Transport protocol	Application protocol	Purpose
ex.1	`1.0.0.1`	`1.0.0.2`	TCP	HTTP	HTTP reply with `image.jpg`

Table 6: Transmitted messages and connection-setup 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?

Thinking creatively about DNS

You are an ordinary user (not a network/system administrator), and your computer is inside the EPFL network. All computers in this network use the same local DNS server.

Can you find out whether a given external URL, e.g., `www.mit.edu`, was recently accessed by another EPFL user?

Web page retrieval time

You type a URL in your web browser, and this causes your DNS client to send a DNS query. The number of DNS servers that participate in answering this DNS query is n. The propagation delay between any two DNS servers or a DNS server and client is constant and equal to D_1; the corresponding round-trip time (RTT) is $RTT_1 = 2D_1$. There are no queuing or processing delays.

How much time elapses from the moment your computer transmits the DNS query until it receives a response, in each of the following scenarios:

- All DNS servers perform recursive requests.
- The local DNS server performs iterative requests.
- The local DNS server tries to perform recursive requests, but some of the other DNS servers do not accept such requests (they send an iterative response instead).

After your computer receives the DNS response, your web browser retrieves the target URL, whose base file references m other URLs, all stored on the same web server. Each retrieved object is small enough that the transmission delays are negligible relative to the propagation delays. The RTT between your computer and the web server is constant and equal to RTT_2. There are no queuing or processing delays.

How much time elapses from the moment you type in the URL until the web browser finishes downloading the entire web page, in each of the following scenarios:

- The web browser and server communicate over a single persistent TCP connection.
- The web browser and server communicates over m parallel TCP connections.