

# REDES DE COMPUTADORES

## 3<sup>RD</sup> LAB – DNS

*BASED ON THE LABS FROM THE BOOK*

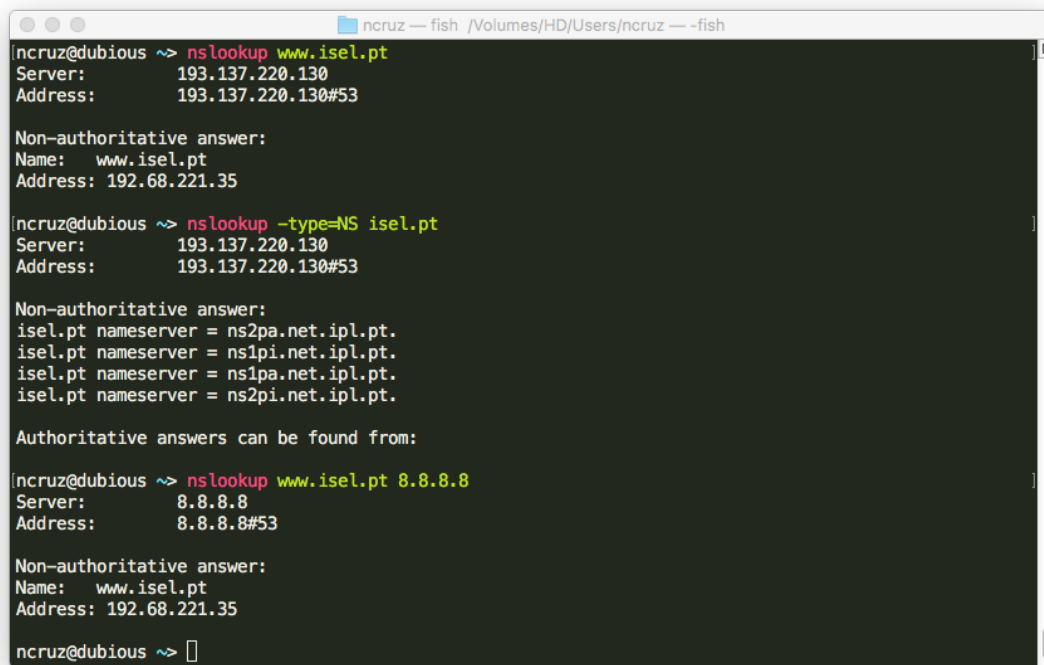
The Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we'll take a closer look at the client side of DNS. Recall that the client's role in the DNS is relatively simple – a client sends a *query* to its local DNS server, and receives a *response* back. Much can go on “under the covers,” invisible to the DNS clients, as the hierarchical DNS servers communicate with each other to either recursively or iteratively resolve the client's DNS query. From the DNS client's standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

Before beginning this lab, you'll probably want to review DNS by the textbook. In particular, you may want to review the material on **local DNS servers**, **DNS caching**, **DNS records and messages**, and the **TYPE field** in the DNS record.

### 1. NSLOOKUP

In this lab, we'll make extensive use of the *nslookup* tool, which is available in most Mac/Linux/Unix and Microsoft platforms today. To run *nslookup* in Mac/Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line. Alternatively, you may use *dig*.

In its most basic operation, *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.



```
ncruz — fish /Volumes/HD/Users/ncruz — -fish
ncruz@dubious ~> nslookup www.isel.pt
Server:      193.137.220.130
Address:     193.137.220.130#53

Non-authoritative answer:
Name:   www.isel.pt
Address: 192.68.221.35

ncruz@dubious ~> nslookup -type=NS isel.pt
Server:      193.137.220.130
Address:     193.137.220.130#53

Non-authoritative answer:
isel.pt nameserver = ns2pa.net.ipl.pt.
isel.pt nameserver = ns1pi.net.ipl.pt.
isel.pt nameserver = ns1pa.net.ipl.pt.
isel.pt nameserver = ns2pi.net.ipl.pt.

Authoritative answers can be found from:

ncruz@dubious ~> nslookup www.isel.pt 8.8.8.8
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
Name:   www.isel.pt
Address: 192.68.221.35

ncruz@dubious ~>
```

FIGURE 1: RUNNING NSLOOKUP

The above screenshot shows the results of three independent *nslookup* commands (displayed in the macOS Terminal Prompt). In this example, the client host is located on the campus of ISEL, where the default local DNS server is dnsf0.net.ipl.pt. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is 193.137.220.130 (or dnsf0.net.ipl.pt). Consider the first command:

```
nslookup www.isel.pt.
```

In words, this command is saying “please send me the IP address for the host *www.isel.pt*”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of *www.isel.pt*. Although the response came from the local DNS server, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in the textbook.

Now consider the second command:

```
nslookup -type=NS isel.pt.
```

Note: The “.” at the end marks this as a complete name and disables the auto complete that some operating systems (windows for example) may apply to the name.

In this example, we have provided the option “-type=NS” and the domain “isel.pt.”. This causes *nslookup* to send a query for a type-NS record to the default local DNS server. In words, the query is saying, “please send me the host names of the authoritative DNS for isel.pt”. (When the -type option is not used, *nslookup* uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer (which is the default local DNS server) along with four ISEL nameservers. Each of these servers is indeed an authoritative DNS server for the hosts on the ISEL campus. However, *nslookup* also indicates that the answer is “non-authoritative,” meaning that this answer came from the cache of some server rather than from an authoritative ISEL DNS server. Sometimes, the answer also includes the IP addresses of the authoritative DNS servers. (Even though the type-NS query generated by *nslookup* did not explicitly ask for the IP addresses, the local DNS server returned these “for free” and *nslookup* displays the result.)

Now finally consider the third command:

```
nslookup www.ipl.pt. 8.8.8.8
```

In this example, we indicate that we want the query sent to the DNS server 8.8.8.8 rather than to the default DNS server (193.137.220.130). Thus, the query and reply transaction takes place directly between our querying host and 8.8.8.8. In this example, the DNS server 8.8.8.8 provides the IP address of the host www.ipl.pt.

Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of *nslookup* commands. The syntax is:

```
nslookup -option1 -option2 host-to-find dns-server
```

In general, *nslookup* can be run with zero, one, two or more options. And as we have seen in the above examples, the dns-server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself. Do the following (and write down the results):

1. Run *nslookup* to obtain the IP address of a Web server in USA. What is the IP address of that server?
2. Run *nslookup* to determine the authoritative DNS servers for a university in USA.
3. Run *nslookup* so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Gmail. What is its IP address?

## 2. IPCONFIG

*ipconfig* (for Windows) and *ifconfig* (for Linux/Unix/Mac) are among the most useful little utilities in your host, especially for debugging network issues. Here we'll only describe *ipconfig*, although the Linux/Unix *ifconfig* is very similar. *ipconfig* can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you all this information about your host simply by entering

```
ipconfig /all
```

into the Command Prompt, as shown in the following screenshot.

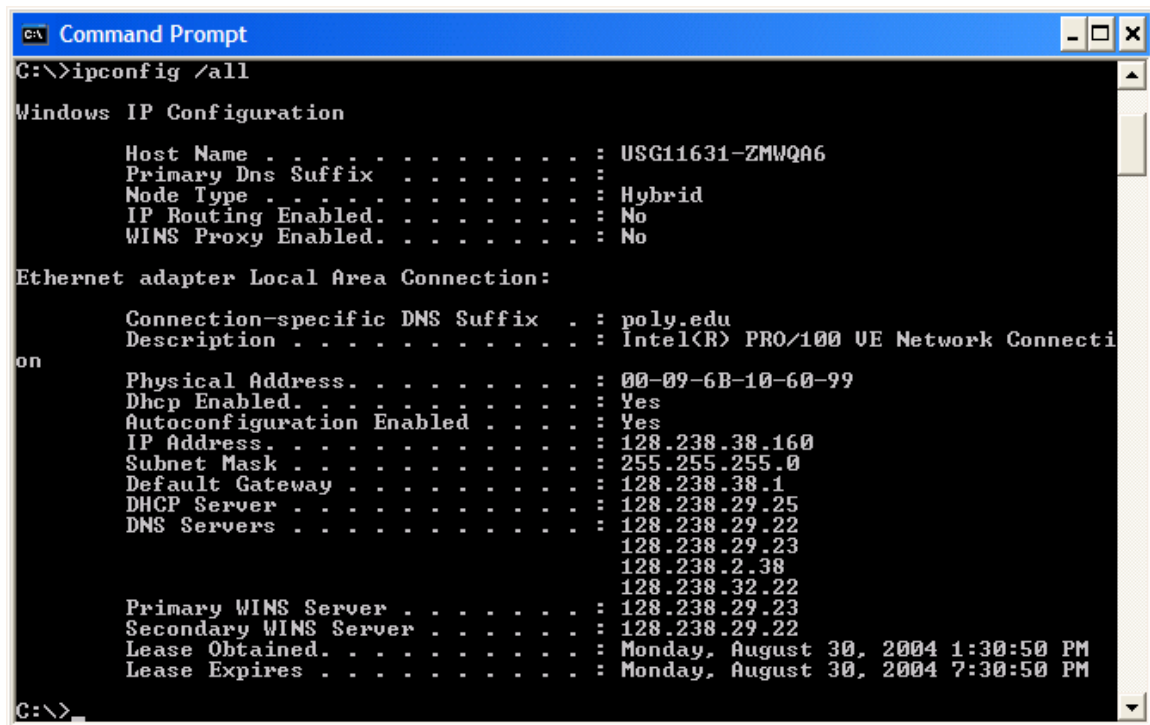


FIGURE 2: RUNNING IPCONFIG

*ipconfig* is also very useful for managing the DNS information stored in your host. We learned that a host can cache DNS records it recently obtained. To see these cached records, after the prompt C:\> provide the following command:

```
ipconfig /displaydns
```

Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter

```
ipconfig /flushdns
```

Flushing the DNS cache clears all entries and reloads the entries from the hosts file.

### 3. TRACING DNS WITH WIRESHARK

Now that we are familiar with *nslookup* and *ipconfig*, we're ready to get down to some serious business. Let's first capture the DNS packets that are generated by ordinary Web-surfing activity.

- Use *ipconfig* to empty the DNS cache in your host.
- Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)
- Open Wireshark and enter "ip.addr == your\_IP\_address" into the filter, where you obtain your\_IP\_address with ipconfig. This filter removes all packets that neither originate nor are destined to your host.
- Start packet capture in Wireshark.
- With your browser, visit the Web page: <http://www.ietf.org>
- Stop packet capture.

Answer the following questions.

4. Locate the DNS query and response messages. Are then sent over UDP or TCP?
5. What is the destination port for the DNS query message? What is the source port of DNS response message?
6. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
7. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
8. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?
9. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?
10. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Now let's play with *nslookup*.

- Start packet capture.
- Do an *nslookup* on [www.isel.pt](http://www.isel.pt)

- Stop packet capture.

You should get a trace that looks something like the following (you may use the keyword dns also as a filter):

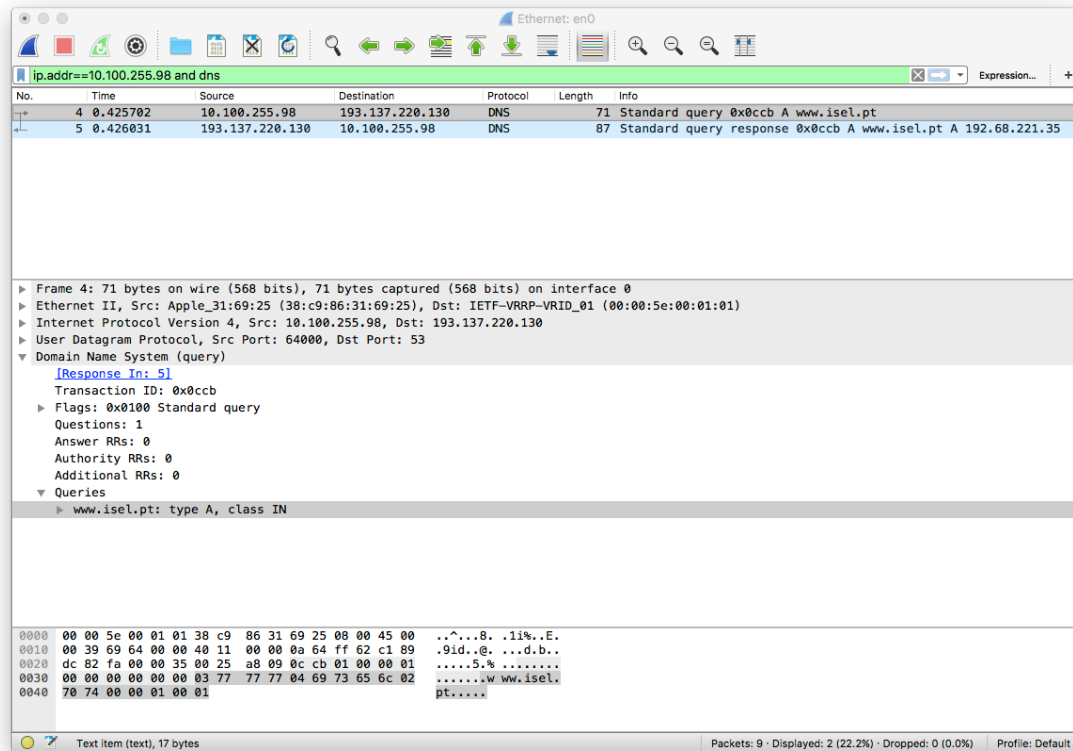


FIGURE 3: WIRESHARK CAPTURING DNS

If you got multiple DNS queries you should focus on the correct query and response messages. Please answer the following questions:

11. What is the destination port for the DNS query message? What is the source port of DNS response message?
12. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
13. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
14. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?

Now repeat the previous experiment, but instead issue the command:

```
nslookup -type=NS isel.pt.
```

Answer the following questions:

15. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
16. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
17. Examine the DNS response message. What ISEL nameservers does the response message provide? Does this response message also provide the IP addresses of the ISEL nameservers?

Now repeat the previous experiment, but instead issue the command:

```
nslookup www.sapo.pt. 8.8.8.8
```

Answer the following questions:

18. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?
19. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
20. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?