

Introduction to Computer Networks

Project #3 15 points

Due: October 23, 15:65

Submit: (1) a hardcopy in class

Note: You are expected to submit a well-organized hardcopy report in class. The report should include

- your names, the project number, date, and etc.;
- the original questions and your answers right under the questions.
- Remember to include supporting materials for your answers. You have to explain how you get each answer by showing diagrams/screenshots (0.5pts) when appropriate and highlighting the related fields (0.5pts).
- This is a group project, and two students a group.

Before beginning this lab, you'll probably want to review DNS by reading the text. 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 Linux/Unix and Microsoft platforms today. To run *nslookup* in 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.

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.

```
veronicas-iMac:~ veronicaxu$ nslookup www.zju.edu.cn
Server:      10.10.0.21
Address:     10.10.0.21#53

Name:   www.zju.edu.cn
Address: 10.203.6.101

veronicas-iMac:~ veronicaxu$ nslookup -type=NS zju.edu.cn
Server:      10.10.0.21
Address:     10.10.0.21#53

zju.edu.cn      nameserver = dns1.zju.edu.cn.

veronicas-iMac:~ veronicaxu$ nslookup www.mit.edu
Server:      10.10.0.21
Address:     10.10.0.21#53

Non-authoritative answer:
www.mit.edu      canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dscb.akamaiedge.net.
Name:   e9566.dscb.akamaiedge.net
Address: 184.30.185.198

veronicas-iMac:~ veronicaxu$ █
```

The above screenshot shows the results of three independent *nslookup* commands (displayed in the terminal in MacOS). In this example, the client host is located on the campus of Zhejiang University in Hangzhou, where the default local DNS server is 10.10.0.21. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is 10.10.0.21. Consider the first command:

```
nslookup www.zju.edu.cn
```

In words, this command is saying “Please send me the IP address for the host www.zju.edu.cn” 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.zju.edu.cn. Although the response came from the local DNS server at Zhejiang University, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 2.5 of the textbook.

Now consider the second command:

```
nslookup -type=NS zju.edu.cn
```

In this example, we have provided the option “-type=NS” and the domain “zju.edu.cn”. 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 zju.edu.cn.” (When the -type option is not used, *nslookup* uses the default, which is to query for type A records; see Section 2.5.3 in the text.) 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 the authoritative ZJU name servers (dns1.zju.edu.cn).

Now finally consider the third command:

```
nslookup www.mit.edu
```

In this example, we indicate that we want to query the IP address of www.mit.edu. The answer indicates that the answer is “non-authoritative,” meaning that this answer came from the cache of some server rather than from an authoritative MIT DNS server. Finally, the answer also includes the canonical name of www.mit.edu and the IP addresses of host.

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):

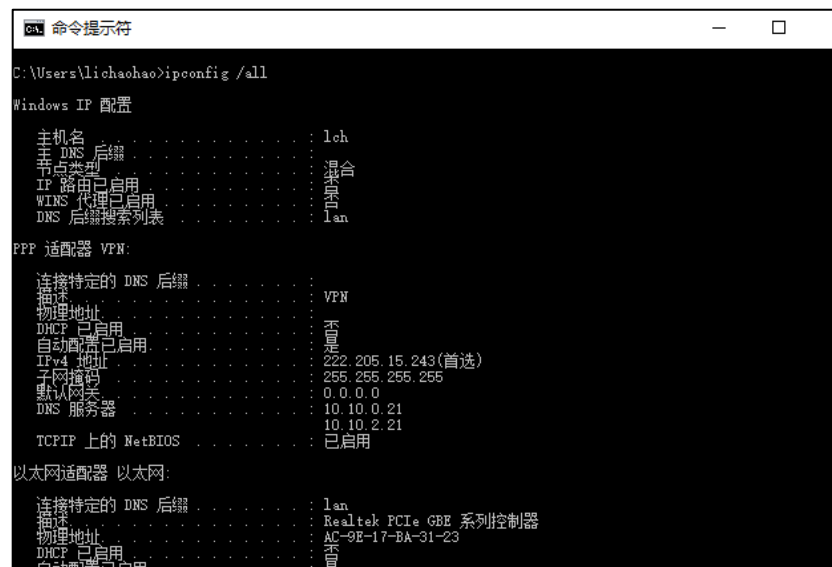
1. Run *nslookup* to obtain the IP address of a Web server in Asia.
2. Run *nslookup* to determine the authoritative DNS servers for a university in Europe.
3. Run *nslookup* so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail.

2. ipconfig

ipconfig (for Windows) and ifconfig (for Linux/Unix) 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 want to see all this information about your host, simply enter:

```
ipconfig /all
```

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



```
命令提示符
C:\Users\lichao>ipconfig /all

Windows IP 配置

主机名 . . . . . : lch
主 DNS 后缀 . . . . . :
节点类型 . . . . . : 混合
IP 路由 . . . . . : 已启用
WINS 代理 . . . . . : 已启用
DNS 后缀搜索列表 . . . . . : lan

PPP 适配器 VPN:

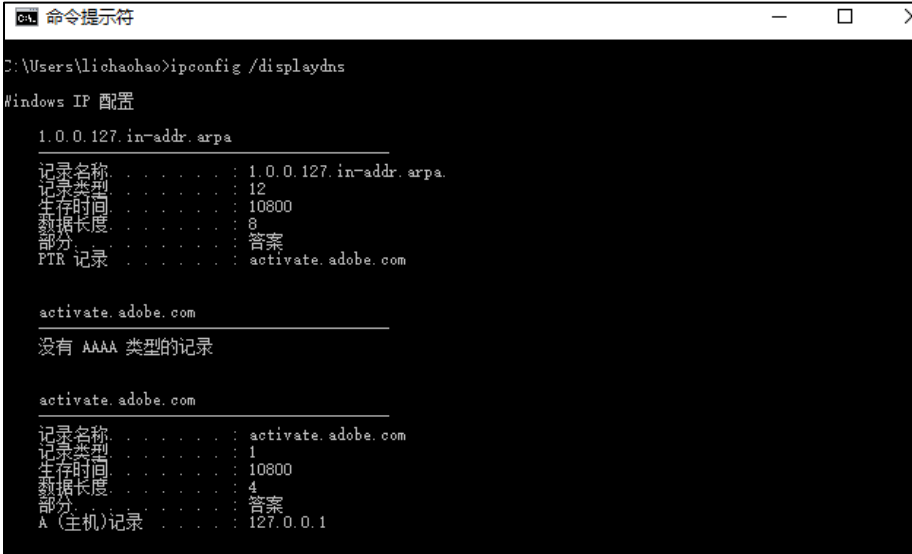
连接特定的 DNS 后缀 . . . . . : VPN
描述 . . . . . :
物理地址 . . . . . :
DHCP 已启用 . . . . . : 否
自动配置已启用 . . . . . : 是
IPv4 地址 . . . . . : 222.205.15.243(首选)
子网掩码 . . . . . : 255.255.255.255
默认网关 . . . . . : 0.0.0.0
DNS 服务器 . . . . . : 10.10.0.21
                        10.10.2.21
TCP/IP 上的 NetBIOS . . . . . : 已启用

以太网适配器 以太网:

连接特定的 DNS 后缀 . . . . . : lan
描述 . . . . . : Realtek PCIe GBE 系列控制器
物理地址 . . . . . : AC-9E-17-BA-31-23
DHCP 已启用 . . . . . : 否
自动配置已启用 . . . . . : 是
```

ipconfig is also very useful for managing the DNS information stored in your host. In Section 2.5 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
```



```
命令提示符
C:\Users\lichaozhao>ipconfig /displaydns

Windows IP 配置

1.0.0.127.in-addr.arpa
-----
记录名称 . . . . . : 1.0.0.127.in-addr.arpa.
记录类型 . . . . . : 12
生存时间 . . . . . : 10800
数据长度 . . . . . : 8
部分 . . . . . : 答案
PTR 记录 . . . . . : activate.adobe.com

activate.adobe.com
-----
没有 AAAA 类型的记录

activate.adobe.com
-----
记录名称 . . . . . : activate.adobe.com
记录类型 . . . . . : 1
生存时间 . . . . . : 10800
数据长度 . . . . . : 4
部分 . . . . . : 答案
A (主机)记录 . . . . : 127.0.0.1
```

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 (the IP address for the computer on which you are running Wireshark) 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.

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author's computers¹. Answer the following questions:

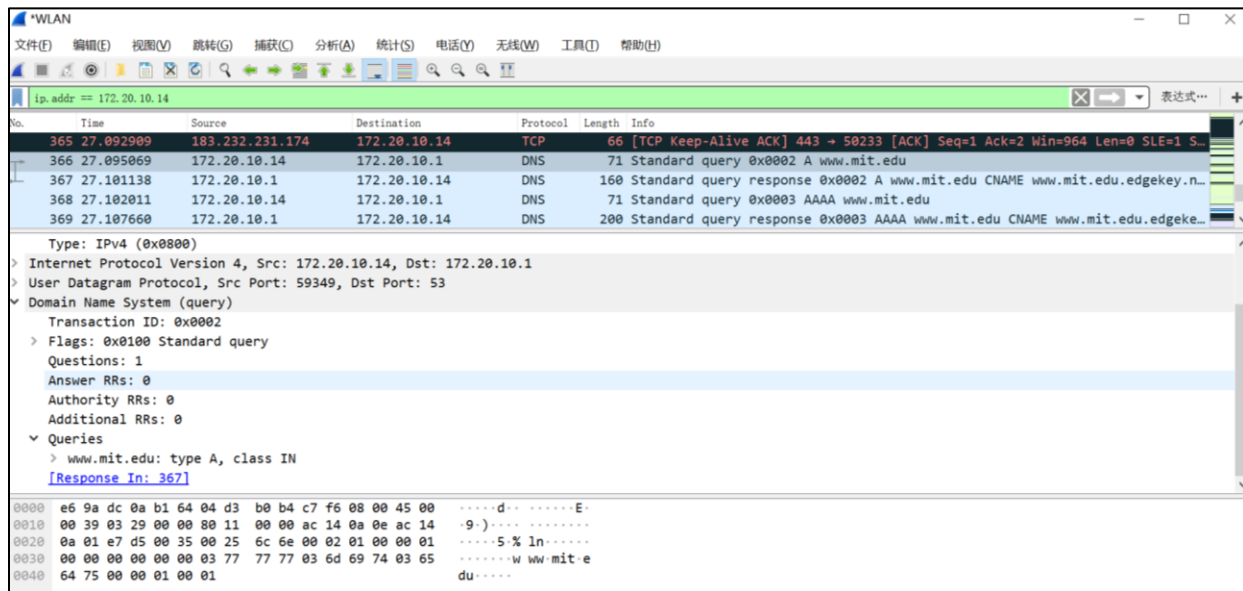
4. Locate the DNS query and response messages. Are they sent over UDP or TCP?
5. What is the destination port for the DNS query message? What is the source port of DNS response message?
6. To what IP address is the DNS query message sent? Use *ipconfig* to determine the IP address of your local DNS server. 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 does 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?

¹ Download the zip file (appendix.trace1) and extract the file dns-ethereal-trace-1. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the dns-ethereal-trace-1 trace file.

Now let's play with *nslookup*².

- Start packet capture.
- Do an *nslookup* on *www.mit.edu*
- Stop packet capture.

You should get a trace that looks something like the following:



We see from the above screenshot that *nslookup* actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first two sets of queries/responses, as they are specific to *nslookup* and are not normally generated by standard Internet applications. You should instead focus on the last query and response messages.

² If you are unable to run Wireshark and capture a trace file, use the trace file *dns-ethereal-trace-2* in the zip file (*appendix.trace2*)

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 does each of these answers contain?
15. Provide a screenshot.

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

```
nslookup -type=NS mit.edu
```

Answer the following questions³ :

16. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
17. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
18. Examine the DNS response message. What MIT name servers does the response message provide? Does this response message also provide the IP addresses of the MIT name servers?
19. Provide a screenshot.

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

```
nslookup www.sc.edu
```

Answer the following questions⁴:

20. 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?
21. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
22. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?
23. Provide a screenshot.

³ If you are unable to run Wireshark and capture a trace file, use the trace file dns-ethereal-trace-3 in the zip file (appendix.trace3)

⁴ If you are unable to run Wireshark and capture a trace file, use the trace file dns-ethereal-trace-4 in the zip file (appendix.trace4)