COM-407: TCP/IP NETWORKING

LAB EXERCISES (TP) 0
BASIC CONFIGURATION, IP SUITE, AND PACKET INSPECTION: PING(6), TRACEROUTE(6), NETSTAT, NSLOOKUP
With Solutions

September 22th, 2022
Deadline: October 5th, 2022 at 23:55

Abstract

In this lab you will practice some networking commands that enable you to obtain information about Internet machines and about the connectivity and the paths between them. You will also learn to use a GUI-based packet capture/inspection tool called Wireshark. You will use tshark (command-line version of Wireshark) for packet capture/inspection.

1 ORGANIZATION OF THE LAB

In this document, you will read the lab instructions. You will solve Moodle quizzes, which will be graded. Carefully follow this document while doing the lab.

First, we will guide you on installing and using the virtual environment that will be used for all the labs. After successful installation, you will be doing the graded part of the lab.

Important: If you only have macOS with an Mx chip (recent MacBook) where $x \in \mathbb{N}$, you should come in person to INF 019 during the lab session, where TAs will give you MINIX and the necessary equipment that you need for the labs. If you have a MacBook, to check what is your chip type: Click on the apple logo on the top-left of your screen and press "About This Mac". In the overview tab, inspect if "Mx" is written.

Indeed, for the lab, you will use Virtualbox, which is an x86 computer hardware simulator. Virtualbox uses x86 CPU instructions. Macs Mx use ARM-based CPUs, which is another technology, and Virtualbox is not available currently on Macs Mx. Virtualbox cannot translate ARM instructions (Mac Mx instructions) to an x86 computer, nor x86 instructions to an ARM computer. That is why you will work on Minix if you have Mx chip.

If you have any questions related to MINIX, please contact a TA (emi.sakamoto@epfl.ch).

2 VIRTUAL ENVIRONMENT

Most of the labs of this course will run in a virtualized environment that can be installed directly on your own computer. With the virtualized environment, you will be able to operate a network with several hosts, routers, and other communication equipment, all in your own machine. This considerably simplifies the operation of the labs and may prove useful outside this course whenever you have to test a communication system.

The virtualized environment is an emulated[1] environment, i.e., the virtual hosts and routers run the same code as real physical hosts and routers; only their hardware is replaced by the virtualized environment.

The virtualized environment is mainly composed of the following items:

- A virtualization software, like VirtualBox, provides hardware emulation to support virtual routers and standard PCs. Version `6.1` was tested for compatibility with the lab, but versions `6.0.x` should all work similarly.
- Next, we need to install the virtual machine in the virtual box. The virtual disk for the virtual machine is available on moodle.
- The virtual machine already has Mininet installed on it, Mininet provides emulation of networks and cables; it will be used to create network topologies from the next lab onwards.

2.1 YOUR FIRST VIRTUAL PC: INSTALLATION GUIDE

Download VirtualBox[2] and install it on your computer.

Download the virtual HD image `MininetVM-disk001.vmdk` from the link provided on Moodle.

We will now create a virtual machine.

1. Launch the VirtualBox software.
2. Create a new virtual machine, set the name to `MininetVM`, the type to Linux, choose version Other 64bit (and press next).
3. Assign 1GB of RAM (press next).
4. Choose "use an existing virtual hard disk file" and pick the uncompressed file `MininetVM-disk001.vmdk`.

Now that the machine is created, we need to add two features to it. First, we create a shared folder between the guest machine and the host machine (i.e., your computer). With this folder, you can easily backup and transfer configuration scripts, snapshots and any file you might find useful during your labs. Second, we will enable the "copy to clipboard" functionality between your host machine and the guest machine, and between guest machines as well. Note that you must do these changes when the VM is not launched.

1. **For the the shared folder:** Right-click on MininetVM and click on "settings", in the "Shared Folders" tab, click on the blue folder with a green cross to add a new folder. Point the "Folder Path" to a folder of your choice on your host machine, and in the "Name" field write `shared`. Check the box `Auto-mount`. Point the "Mount point" to `/media/lca2/shared`.
2. **For the clipboard copy**: Again from the same machine settings window, click on the "General" tab, then on the "Advanced" subtab tab and set the shared clipboard to "Bidirectional". Click on "OK" at the bottom right of the settings window.

[1] In contrast, a *simulated* environment such as ns3 replaces hosts and routers with simplified code.

[2] You can also use a different virtualization software, but we will provide support and instructions only for VirtualBox.

Lastly, we will attach a NAT to the VM for it to be able to access the internet via the host connection. This should be done by default, but in case it is not, follow the steps below.

In VirtualBox, click on "VirtualBox" and "Preferences", then go to "Network" tab and add a new NAT networks by clicking on green cross. Then right click and select "edit NAT Network", please select all the parameter, except "port forwarding".

Then go back to the machine settings by right-clicking on MininetVM and selecting "Settings", go to "Network" and then check "Enable Network Adapter" and choose "NAT Network". Finally, select the network you have created before: "NatNetwork" subtab. Click on "OK" at the bottom right of the settings window.

Congratulations, you have just created a virtual machine!

2.2 RUNNING THE VIRTUAL MACHINE

In VirtualBox select `MininetVM` and run it (Double-click on MininetVM).

> Advice: If you have experienced some issues running the virtual machines, such as Kernel problems when running the virtual machine, ensure that you have downloaded VirtualBox from the official website. If not, uninstall your version, and download a version directly from it. You can use a recent version of VirtualBox that is not the latest release: uninstall your version of VirtualBox and install version 6.1.4 for example.
> Advice: For Mac users, the installation requires you first to accept the software in the security preferences of the system: if this is the case uninstall VirtualBox, download it again, accept it in the security preferences and then install it again.
> Advice: In case you have issues launching your virtual machine, you may need to activate hardware virtualization. To do this you need to access the `BIOS` menu of your computer. You might need to power off your computer after enabling this option (not just a simple reboot!).

Login using the username `lca2` and password `lca2`. (Password for azerty keyboards on the login page: `lcqé`).

At the first launch of the VM, there could have a warning about "A new version of Ubuntu is available. Would you like to upgrade ?". Click on "Don't upgrade".

Also, there could be updates to be done, accept them. This step may take some time depending on the amount of updates and on your internet connection. If you encountered this error message: "Failed to download package files Check your internet connection", please enter in a terminal the following command: *sudo apt -y update && sudo apt -y upgrade && sudo apt -y autoremove*

The virtual machine should already be connected to the Internet via the host's connection. Open the Firefox web browser and go to a webpage to check that you are indeed connected.

Check that the shared folder works: outside of virtual box on your machine, copy files to the shared folder. Now, on your virtual machine, find the shared folder and check that the files are there.

If you have any issues at this point (no Internet connection, cannot launch VM, cannot copy paste, cannot access shared folder) that you cannot resolve yourself, contact a TA for help.

2.3 LINUX CRASH COURSE (OPTIONAL)

This section is meant to provide a brief introduction to Linux commands and best practices. Feel free to skip this section and go directly to Section 3, if you are familiar with Linux. If you decide to skip this section, we encourage you to do the non-mandatory research exercise at the end of the lab. Note however, that these are the basic commands that you will need to be familiar with for all the remaining labs.

2.3.1 KEYBOARD LAYOUT

The keyboard layout on your virtual machine is `us` by default. To change it to a swiss one, open a terminal (LXTerminal, available on the desktop) and type the command:

```
$ sudo setxkbmap ch
```

However, you will have to type this command every time you reboot your system. If you want to make this command executed automatically upon booting the system, you need to write a configuration script (for example `keyboard_conf.sh`) and place it in the folder `/etc/profile.d`. To do so, go to the folder by typing the command

```
$ cd /etc/profile.d
```

then create the configuration file and open it with a text editor

```
$ sudo leafpad keyboard_conf.sh
```

and write the following code in it:

```
#!/bin/sh
setxkbmap ch
```

Then save the file and check that your newly created file exists: `ls` (this command shows you all documents in the folder that you are in). You can check its content in the terminal using the command `cat keyboard_conf.sh`. You have to set your file to executable with the following command

```
$ sudo chmod +x keyboard_conf.sh
```

Now if you check it again with `ls`, the name of the file should be in a different colour than before, this shows that it has been transformed into an executable.

You can test that it works by restarting your VM and making sure that after the reboot, the keyboard layout is in the desired language.

2.3.2 LINUX COMMANDS

The Linux distribution of the virtual machine comes with a friendly graphical interface. For configurating network interfaces however, we will use the terminal. Here are a few things you need to know:

Linux is a multi-user system that uses the Extended file system (e.g., ext2, ext3, ext4) to store files. In extfs each file has a unique owner (a user), belongs to a group of users, and has a set of permissions which define access rights to the file (read, write, and/or execute) for the owner, the group, and everyone else.

Each normal user has a home directory, that is referred to by the symbol $\sim$ (tilde). To change directories in the terminal use the command `cd` followed by the name of the directory. This can be a relative name,

such as `Documents`, or an absolute name, such as `/etc/init.d` (i.e., beginning with the `/`, which is the root of the filesystem). It can also be the home directory (i.e., `cd ~`). To move up in the tree, i.e., out of a directory, use `cd ..` (two dots). The current directory is always represented by a single dot, so `cd .` does nothing. To display the current directory use `pwd` (print working directory).

Try it yourself: open a terminal [3]. Use `pwd` to show the current directory. Then `cd` to `~/Tutorial/`. Use the Tab key for auto-complete. If you cannot find the ~ tab on your keyboard, try using the F6 key.

In the terminal, you can list the files in a directory by using the command `ls`. You can add switches to a command. For example, if you want to see detailed attributes of all the files in a directory (including the permissions), you can use the `-l` switch:

```
$ ls -l
```

You should see something like this:

```
lca2@lca2:~/Tutorial$ ls -l
total 4
-rw-r-r-- 1 lca2 lca2   0 Aug 18 12:14 emptyFile.txt
-rw-r-r-- 1 lca2 lca2 12 Aug 18 12:16 helloWorld.txt
```

You can output the contents of a file to the terminal by using commands such as `cat` or `less`:

```
$ cat helloWorld.txt
```

You can see above that the permissions of the `helloWorld.txt` file are `-rw-r--r--`, that it belongs to the user `lca2` and the group `lca2`, and that it is `12` bytes long. The permissions string is 10 characters long. The first character is either a dash `-` for regular files, or other letters for special files (a `d` for directories, etc.). The next three characters give the permissions for the owner of the file, in this case `rw-`. This means that the owner has the right to read the file (`r`), to write/modify the file (`w`), but not to execute the file. If the file was executable, in the third position there would be an `x`. The next three characters describe the permissions of the group, and the last three characters the permissions of all the other users in the system. In this case, the file is read-only for the group and for everyone else.

A file that the user `lca2` does not want anyone to see but herself would have permissions `-rw-------`, whereas a file with full rights for everybody would have permissions `-rwxrwxrwx`.

A file's permissions can be changed by using the command `chmod`. You need to specify whose access rights to the file you want to alter: of the user who owns it (`u`), of the group (`g`), or of others (`o`), whether you want to add (`+`), or remove (`-`) a right, and which right you mean (`r`, `w`, or `x`).

For example,

```
$ chmod o-r,g+w emptyFile.txt
```

[3] Tip: The shortcut to open a terminal is `Ctrl+Alt+T`.

removes the reading right for other users than the owner or the group and adds writing for the group. To change the ownership of the file, use `chown`.

When you issue a command in the terminal, you are in fact running a certain executable file. The command interpreter (or the shell) looks for these executable files in one of the several directories specified in the `PATH` environment variable. To list the contents of this variable, run

```
$ echo $PATH
```

The character `$` before `PATH` indicates that we want to display the contents of the variable `PATH`; without it, the command would simply display the string `PATH`. The directories are separated by semicolons, and they are searched in order. To see which executable you are running, use the command `which` followed by the name of the executable. For example, `which ls` displays `/bin/ls`, the location of the `ls` executable.

Note that the current directory (`.`) is not in the `PATH` for security reasons (a miscreant user might create an executable called `ls` in some directory, which in fact erases the given directory instead of listing it). Therefore, if you really want to execute a file in the current directory, you need to specify the path (the current directory), i.e., to type `./some_script` instead of simply typing `some_script` (the latter results in a "file not found" error).

Normal users cannot alter system configurations files (they do not have permission). For this reason it is safer to use a Linux machine as a normal user, and not as an administrator. This way, you cannot do too much harm.

There is a super-user (administrator) called `root` that has absolute rights (i.e., can do **anything**). In the terminal, the command prompt for a normal user ends with a dollar sign `$`, whereas for the `root` the prompt ends with a hash `#`.

IMPORTANT In these labs, whenever you see the hash `#` sign in front of a command that you are supposed to type, it means that you need `root` access.

There are users called "sudoers" that are allowed to run a single command as `root` (the user `lca2` in our virtual machine is such a user). This is achieved by typing `sudo` followed by the desired command. You will then be prompted for the password of the user.

If you want to run a terminal in `root` mode, type the command `sudo su`. You will then be prompted for the `root` password and you will switch to `root` mode. the password for root is `lca2`.

OTHER USEFUL COMMANDS: if you launch an application using the terminal ex: `$ leafpad`, then the application will open but the terminal you used will be dedicated to the application and you won't be able to type other commands in it, to prevent the use of too many terminal windows simultaneously, you can detach a command using `&` at the end of your command: `leafpad &`, this will launch the application and allow you to type other commands in the same terminal afterwards. Note that if `sudo` is needed to launch the application, the use of `&` may not work because the detachment of the command does not allow you to type the password required by sudo. Another useful command allows you to search for files on the entire machine or in specific branches of the arborescence:

```
sudo find <branch> -iname <file>
```

you can replace `<branch>` by `/` if you want to search everywhere or with the path to where you want to search e.g. `/etc/`, you can replace `<file>` with the name of the file you are searching for `keayboard_conf.sh`

or if you don't remember the exact name you can write elements of it and use * to signify anything: *board*. To sum up, if you want to find your keyboard configuration file but only remember it is somewhere below /etc/ and that the word "board" is in it you can type

```
sudo find /etc/ -iname *board*
```

This will output the path to your file and potentially other files that also satisfy this description.

From the command line, you can write several commands at a time using |. For example if you want to run command2 on the output of command1 you can type

```
command1 | command2
```

For example, if you place yourself in the folder /etc/init.d and run

```
ls | grep key
```

then the first part ls outputs all documents and on this output, the second part only outputs the filenames that include the string key, thus it should output your keyboard_conf.sh file and no filenames that do not contain the string key. Notice the use of grep, this command enables you to search for specific strings of characters in an output. Finally, you can write the output of a command to a file with >:

```
command > file.txt
```

or you can append the output of your command to a file which already contains other information using >>:

```
command >> file.txt
```

2.3.3 BEST PRACTICES

In these labs you will often type configuration commands in the terminal, usually one by one, to observe and understand their effects. However, after a reboot, the effects of these commands are usually lost, and you need to type them again, which is cumbersome.

We recommend the following practice:

Keep a text editor open in the virtual machine (for example "Leafpad", located in Accessories, or nano in another terminal). Whenever you type a configuration command in the terminal, paste it in the editor. In Linux it suffices to select a text to copy it in the clipboard. For pasting use the middle mouse button. Otherwise use the standard "right-click" + Copy (but be warned that this might not work in all terminals). The shortcuts for copy and paste on the terminal are Ctrl+Shift+C and Ctrl+Shift+V, respectively.

Save the resulting file in your home directory (for example as conf.sh). When you reboot, you can run all the commands in the file as root by

```
# sh conf.sh
```

or as a regular user via `sudo` by

```
$ sudo sh conf.sh
```

2.3.4 ADDITIONAL INFO

There are two main software packages that provide tools for configuring the network: the older, standard `net-tools` (provides `ifconfig`, `route`, `netstat`), and the newer and more powerful `iproute2` (provides `ip`, `ss`). Both are installed on the virtual machine, but we will focus primarily on the second set of tools (here is an angrily argumented viewpoint `http://inai.de/2008/02/19`).

3 THE IPV4 INTERNET AND NETWORK PACKET INSPECTION

Launch MininetVM and do the lab in there.

This document will guide you through the Moodle quizzes, which will be graded. The grading is indicated in Moodle quizzes. Notice that some questions have feedback that can help you.

Q1/ Answer **Lab 0 - Part 1** on Moodle

Solution.

3.1 FINDING THE INFORMATION ON THE CONNECTION

Connect to the Internet in IPv4 and disable IPv6 connectivity, if needed.

To disable IPv6, use the following commands from the Terminal *app*

```
$ sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
$ sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

You can check that your command worked by checking the value of the variable that you just tried to set to 1:

```
sudo sysctl -a | grep disable_ipv6
```

After disabling the IPv6 connectivity, we now want to determine the following information:

- *the IP address(es) of your virtual machine* `<my_ip>`,
- *the netmask* `<my_netmask>`, *and*
- *the default gateway of your virtual machine* `<my_gateway>`.

Use the following commands in the Terminal *app*

```
$ ip addr show
$ ip route show
```

***Part 1 Question 1**:*

- *<my_ip>=* ***10.0.2.****, several IP addresses are accepted, depends on your host, but belongs to 10.0.2.0/24.*
- *Your IP address is* ***private***
- *<my_netmask>=* ***255.255.255.0***
- *<my_gateway>=* ***10.0.2.****

***Part 1 Question 2**:*

Possible way you can infer is your IP address public or private is by: ***confirming on the internet that the obtained address belongs to private IP address range****.*

***Part 1 Question 3**:*

The netmask or prefix is used to distinguish the ***network*** *and the* ***host*** *parts of the IP address.*

3.2 NETWORK PACKET INSPECTION. WIRESHARK

We need to see or inspect the packets leaving or coming to our computer or other computers for various reasons. These reasons vary depending on the person and his motivations. For example, network administrators need this for troubleshooting network-related problems, software developers for debugging network-related code and network protocol implementations, and security engineers for analyzing the network traffic for security purposes. In general, we all can use these tools to understand how machines actually communicate with each other, i.e., to understand the internals of the network protocols.

There exists many tools for network packet inspection. Under the hood, all these tools use packet capture libraries such as libpcap, winpcap or npcap but they differ in the way users can interface with them and the features they provide. For example, Wireshark is a powerful sniffer which can decode lot of protocols. It provides a nice GUI to make usage more user friendly.

Since there are a lot of packets generated by the applications running on your machine, you may want to use filters, for more details see `https://wiki.wireshark.org/DisplayFilters`. *Please note that there are two types of filters:* capture *and* display. *Capture filters are used to selectively capture the traffic whereas with display filters, you capture all the traffic but the traffic is displayed as per the filter rules.*

Wireshark is already installed on your virtual machine. Start it (as administrator) by typing `sudo wireshark` *on the terminal. On the main page, under "Capture" you have a list of interfaces that you can select to observe its passing traffic. Next to the interface name, the amount of traffic on it is illustrated by a signal. Select the interface that is currently used for internet connectivity and capture it by clicking on the blue shark fin.*

Write a filter command that displays only the packets with destination IP address of your default gateway: ***ip.dst==10.0.2.****

Next, navigate to a webpage through your browser. Do you see any packet captured (other than DNS)? ***No.***

This is because ***in IP communication is done end-to-end.***

3.3 PING

PONG

The ping command uses the ICMP protocol to probe whether a host is up:

```
$ ping <host_name>
```

Part 1 Question 5*:*

When we ping an address for the first time, for example EPFL's address: First, to find IP address of EPFL, we expect that ***DNS query*** *is sent, after which* ***DNS query response*** *is received. Next, a* ***ping request*** *is sent to the IP address of epfl, which is followed by* ***ping reply****. The protocol that is used for ping request/reply is* ***ICMP****.*

***Part 1 Question 6**:*

Start a new capture with Wireshark and then ping www.epfl.ch. *Do you see the messages from the previous question in the Wireshark output?* ***Yes***.

***Part 1 Question 7**:*

Stop the ping and start the second ping again after a couple of seconds. Compare the packets in Wireshark with the packets that were captured during the first ping. What do you observe? ***The second time DNS request is typically not performed because the IP address was cached***.

***Part 1 Question 8**:*

In a browser open www.netflix.com. *Is the server hosting the website up?* ***Yes***.

***Part 1 Question 9**:*

Now ping www.netflix.com. *Does it work?* ***No***.

***Part 1 Question 10**:*

Explain the findings from the previous 2 questions. ***The server is configured not to respond to ping***.

***Part 1 Question 11**:*

Ping www.canterbury.ac.nz *and* www.newzealand.com. *RTT to* www.canterbury.ac.nz *is around* ***350*** *ms, while RTT to* www.newzealand.com *is around* ***20*** *ms. Based on this observation, server* ***www.canterbury.ac.nz*** *may be located in New Zealand.*

Q2/ Answer **Lab 0 - Part 2** on Moodle.

Solution.

3.4 TRACEROUTE AND NETSTAT

***traceroute** is a tool for displaying the route to a destination. For example, to display the route to* `www.grimper.ch` *use the following command in the Terminal app:*

```
# sudo traceroute -I www.grimper.ch
```

Note that the traceroute command on VirtualBox is sometimes unstable; we have observed that students with Windows machines sometimes encountered issues. If this is the case for you, you can perform the traceroutes directly on your host machine (i.e., not in the virtual machine but directly on your machine). We observed this issue with IPv4. Don't hesitate to ask a TA for help.

Start Wireshark and do traceroute to www.grimper.ch. *One of IP addresses of grimper.ch is* ***104.26.11.137***. *Here* ***ICMP*** *is used for traceroute.*

We need to filter the packets with the destination address of the grimper server: `ip.dst == grimper_ip_address`. The IP address of grimper that we saw is 104.26.11.137 but they may have several (104.26.10.137, 172.67.73.241 for example). You should see ICMP packets due to the option -I used for traceroute.

netstat** is a tool for displaying TCP connections, routing table, interfaces and network statistics. On Linux, netstat (part of net-tools) is superseded by* ***ss *(part of iproute2).*

Open a web browser, go to `www.epfl.ch`*, and leave the browser open for a moment.*

Look at the active TCP connections:

```
# ss -t -n
```

Remarks *: The option -t displays TCP sockets; that is how you get only TCP connections. The -n switch prevents name resolving and makes netstat/ss display results faster (but obviously without the names of the hosts).*

Identify the TCP connections where the destination IP address is the IP address of the www.epfl.ch webpage. Is there one, or are there several such connections? ***Several***

3.5 MAC ADDRESSES

A MAC address (media access control address) of a device is a unique identifier assigned to a network interface controller (NIC). MAC addresses are linked to the hardware of network adapters. A MAC address is given to a network adapter when it is manufactured. It is hardwired or hard-coded onto your computer's network interface card (NIC) and is unique to it.

Remark: The format of a MAC address is XX:XX:XX:XX:XX:XX . Please use this format when answering the questions in the labs.

Part 2 Question 3*:*

What is the MAC address of your ethernet interface? ***08:00:27:e0:52:e4***. It depends on the machine.

Part 2 Question 4*:*

The protocol for resolving a MAC address with IPv4 is called ***ARP****. Suppose that we know IPv4 of the default gateway and we want to find out the MAC address of the deafult gateway. In Terminal, we can achieve that with command* ***arp*** *with option -a.*

Part 2 Question 5*:*

What is the MAC address of your default gateway? ***52:54:00:12:35:*****. *In order to find the default MAC address of the default gateway, run the command "ip route show" to see the IP address of your gateway. Finally, run the command "arp -a" and find the gateway IP address, you'll see the MAC address on the same line. If you want to do it with Wireshark, ping the gateway and capture the ICMP packets. Look at the ethernet layer for the echo request destination MAC.*

Part 2 Question 6*:*

You will describe what happens when resolving the MAC address with IPv4:

First, the request is ***broadcast to all the machines in the same subnet****. Next,* ***the machine whose MAC address is being resolved replies*** *to the machine that sent the request. Since the whole concept works on MAC layer, we confirm that it is possible to find out MAC addresses of* ***all the machines in the same subnet****.*

Part 2 Question 7*:*

Ping www.newzealand.com*. The MAC address of the packet received from it while pinging is* ***52:54:00:12:35:*****. *This is the MAC address of the* ***default gateway****.*

Q3/ Answer **Lab 0 - Part 3** on Moodle.

Solution.

4 NAMES IN THE INTERNET

Juliet: *[...]*
What's in a name? That which we call a rose
By any other name would smell as sweet.

W.S.

Replace your DNS servers by an inexisting IP address, say `1.2.3.4`*. If you configured statically your DNS servers, don't forget to write them down somewhere before changing them to* `1.2.3.4`*.*

To change your DNS server configuration, edit `/etc/resolv.conf` *file. Comment out the lines that begin with* `nameserver` *(precede them with the* `#` *character) and add one line* `nameserver 1.2.3.4`

Start capturing with Wireshark and ping www.grimper.ch.

Do you get the reply from DNS server? ***No****.*

Try pinging the IP address of grimper (that you discovered in the quiz Lab 0 part 2 under traceroute section). Does it work? ***Yes****.*

Restore now your initial DNS configuration*.*

nslookup *is a command-line tool for querying Domain Name System (DNS) name servers. Run* `nslookup` *with the address of the Google public DNS server.*

```
# nslookup - 8.8.8.8
```

For IPv4, in the > prompt, type:

```
>set type=A
```

For IPv6, in the > prompt, type:

```
>set type=AAAA
```

Then type epfl.ch.

IPv4 address is ***128.178.222.****.*

IPv6 address is ***2001:620:618:1de:1:80b2:*:****.

Part 3 Question 3*:*

Start a capture in Wireshark and do an IPv4 traceroute to www.grimper.ch. Focus on one line of the form:

name (IPv4address) time ms time ms ... time ms

Filter the DNS packets in Wireshark. Look at the capture and identify the packet in which you see the same name as in the terminal.

The observed packets differ from the typical DNS, since we ask the DNS server what is the ***name*** *of the server with a given* ***IP address****. Therefore, in case of traceroute, reverse DNS query is performed.*

Part 3 Question 4*:*

Based on the observed difference and on the theory, explain how traceroute works:

*The traceroute tool works by sending the **UDP** packet (in case of Linux and MacOSX) and **ICMP** packet (in case of Windows) with **increasing** TTL values until it reaches the destination. If n-th packet expires on the router, that means that this router is the n-th hop to our destination. When TTL expires on the intermediate router, the intermediate router **replies to initiator of traceroute** and that is how the initiator finds out about **IP addresses of intermediate routers**. Finally, the initiator of the traceroute **performs reverse DNS query asking for intermediate routers' names**.*

5 IPv4 AND IPv6

*Now let's examine the situation when IPv6 connectivity is also present. For this part, You should have **an IPv6 connectivity, which is the case for the EPFL Wifi**.*

To restart the IPv6 connectivity:

```
$ sudo sysctl -w net.ipv6.conf.all.disable_ipv6=0
$ sudo sysctl -w net.ipv6.conf.default.disable_ipv6=0
```

*and **reboot the machine**. As previously, you can check that your command worked using*

```
sudo sysctl -a | grep disable_ipv6
```

To ping over IPv6:

```
$ ping6 <host_name>
```

To traceroute over IPv6:

```
$ sudo traceroute -I -6 <host_name>
```

Note that the ping6 and traceroute for Ipv6 command on VirtualBox is sometimes unstable; we have observed that students sometimes encountered issues. If this is the case for you, you can perform the traceroutes directly on your host machine (i.e., not in the virtual machine but directly on your machine). We only observed this issue with IPv6. For MAC users, the command on your machine are ping6 and traceroute6. For Windows users, the command is ping -6 and tracert -6. Don't hesitate to ask a TA for help.

For Minix users, to access IPv6, either you connect to the Minix by the epfl WiFi (do not forget to disconnect the wired connexion (ethernet cable)). You will find on the Minix desktop a setup file if needed. Or you can also do this part on your own computer.

***Part 3 Question 5**:*

Ping grimper.ch over IPv4 and IPv6. Write down the average RTT that you obtain in case of ping and ping6.

Results on my machine *: The average RTT when using IPv6 is **17** ms, while with IPv4 it is **15** ms.*

***Part 3 Question 6**:*

Use wireshark to observe the traffic. On your computer type: ping6 www.grimper.ch

Describe some differences in the observed traffic compared to the IPv4 case. ***IPv6 and IPv4 packets may take different paths to reach the destination host, also at any given moment we could experience congestion in the network, thus RTT may be different. In the IPv4 case, ICMP packets are exchanged. On the other hand, in the IPv6 case we see that ICMPv6 packets are exchanged, so the difference is also in the protocol used. Differences are also in packet length.***

Part 3 Question 7*:*

Repeat the test with the traceroute6 *command. Does the path to grimper in the IPv6 Internet cross the exact same routers as in IPv4?* ***No/Yes, both are accepted****.*

However, it is possible that some routers in the paths are ***dual-stack routers****, so they can forward both IPv4 and IPv6 packets.*

Part 3 Question 8*:*

If you access a webpage via an IPv4 connection and an IPv6 connection, do you think it has to be the exact same page? ***No****.*

Part 3 Question 9*:*

Can you imagine by which mechanism a difference may occur? The ***web server*** *itself, when it is contacted by a client, knows on which network (IPv4 or IPv6) the HTTP request arrives (based on sockets, as we will see later in the course). The web server then runs scripts with different instructions depending on whether the request arrived over IPv4 or IPv6. Intermediate systems are, of course, not involved in this.*

Part 3 Question 10*:*

Do a traceroute in IPv4 and IPv6 to www.switch.ch.

Does traceroute in IPv4 work? ***Yes***

Does traceroute in IPv6 work? ***Yes***

Do they traverse different routers? ***Yes/No, see Part 3 Question 12***

Part 3 Question 11*:*

Now, start a new Wireshark capture, open a browser and type www.switch.ch.

Your connection to the webpage done with: ***It depends on your operating system****.*

Part 3 Question 12*:*

Explain how do you think a machine could decide whether it uses IPv4 or IPv6, assuming that a target host has both IPv4 and IPv6 addresses.

Typically, ***it depends on the machine (e.g. configuration, decision-making algorithms of vendors****.*

RESEARCH EXERCISES (OPTIONAL)

6 WIRESHARK VS TSHARK

You already have experience of Wireshark usage. There also exists a command line version of wireshark, called tshark. Depending on one's needs, abilities, and familiarity, one may sometimes find tshark more handy than wireshark or vice-versa. In the research exercise you will compare tshark and wireshark and see in which cases one tool is better than the other.

In the bonus part, we introduce you with tshark.

Q4/ Answer **Lab 0 - Bonus** on Moodle.

Solution.

6.1 TSHARK

tshark lets you capture packet data from a live network, or read packets from a previously saved capture file. The captured packets are decoded by tshark and then, can either be printed to the standard output or written to a file. tshark's native capture file format is pcap format, which is also the format used by wireshark and tcpdump.

6.1.1 A SHORT TUTORIAL ON TSHARK

To capture all the traffic passing through a certain interface and save it in `captured_packets.pcap` *file, you first need to create the file and change its permissions such that* "Anyone" *can* "Change content". *Then use the following command:*

```
# tshark -i interface_name -w captured_packets.pcap
```

where `-i` *should be followed by the name of the interface and* `-w` *with the name of the file for captured data. In order to get the names of interfaces you can use the* `-D` *option:*

```
# tshark -D
```

Now, using a web browser, visit few web pages like facebook.com or cnn.com. Once you're done, stop the packet capture by pressing Ctrl + C.

To read the packets captured in `captured_packets.pcap` *file, use the* `-r` *option. Following should read all the packets captured in the* `captured_packets.pcap` *file:*

```
# tshark -r captured_packets.pcap
```

If you want only http request packets to be displayed, please do:

```
# tshark -r captured_packets.pcap -Y http.request
```

where `-Y` option lets you specify display filters (using the same syntax as in Wireshark).

Now, let's display the hosts you connected through http. To specify that, you need to use `-T` option to specify that we want to extract fields and `-e` option to specify the field you want to be displayed. Therefore, the whole commands becomes:

```
# tshark -r capture.pcp -Y http.request -T fields -e http.host
```

If you want to check whole list of available options in tshark, you can do:

```
# tshark -help
```

or the help page can be accessed through web with this link

https://www.wireshark.org/docs/man-pages/tshark.html

The capture and display filters used in tshark are the same as in Wireshark and can be accessed with below links.

Capture Filters: https://wiki.wireshark.org/CaptureFilters

Display Filters: https://wiki.wireshark.org/DisplayFilters

6.1.2 EXERCISE

Alice is soon going to have her holidays. She is searching for holiday offers on the web. She finds a very interesting and inexpensive offer at a website and therefore, she hurries up to book it. She enters all her details in a html form, including her name, date of birth, phone numbers, email addresses, home address, and registers for this offer. After registration, when she wants to pay for this offer, she realizes that her connection to this website (until now) is not encrypted. So she stops the online payment.

The pcap file, named `alice.pcap`, stores all the above-mentioned activities of Alice, captured by tshark at her network interface. It can be found on Moodle (In folder Lab 0).

Now, your job is to find the packet in the pcap file that contains all her information. Ultimately, we are interested in Alice's birthday.

Use the shared folder to place the provided `alice.pcap` file on the Desktop of your virtual machine. In the terminal, place yourself on the Desktop. You should come up with a tshark command to get hold of all her details she typed in for reserving this trip.

Hint1: The details are filled by Alice in a html form. Therefore, an http post request body should contain her details.

Hint2: Rely on the documentation!

***Part Bonus Question 1**:*

*What http filter you can use to see only post request packets? **http.request.method==POST**.*

***Part Bonus Question 2**:*

*What is the name of http field that displays the data of the file? **http.file_data**.*

***Part Bonus Question 3**:*

What is Alice's birth date? ***26/08/1992 with for example command***

```
tshark -r alice.pcap -Y 'http.request.method == POST' -T fields -e text
```

.