COM-407: TCP/IP NETWORKING

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

September 12th, 2024
Deadline: September 25th, 2024 at 23:55 PM

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: The installation instructions will differ depending on your operating system (Windows/Linux or Mac). If you have a MacBook, the instructions also differ whether you have macOS with an M1 chip (recent MacBook) or not. 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 "M1" written.

For the lab, you will use VMware, which provides different solutions depending on whether your hardware runs on ARM (Mac M1) or x86 instructions.

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.

2.1 YOUR FIRST VIRTUAL PC: INSTALLATION GUIDE

2.1.1 VMWARE AND VM DOWNLOAD

If your host machine runs on **Windows or Linux**, download VMware Workstation Player for your operating system. The labs were tested on version 17.0.2. Then, download the virtual HD image `MininetVM-x86.ova` from the link provided on Moodle.

If your host machine runs on **Mac**, download VMware Fusion Player. You will need to register for a personal use license, which is free. The labs were tested on version 13.

If you have a **Mac with M1 processor**, download `MininetVM-arm.zip` from the link provided on Moodle and unzip it. If you have an **older Mac**, download `MininetVM-x86.ova`.

2.1.2 INSTALLATION INSTRUCTIONS

If you have an **x86 processor**, open VMware, click on ”file” and ”Open a Virtual Machine”. Select the `MininetVM-x86.ova` file, name your VM as ”MininetVM” and click ”import”. You might get an error message that the import did not pass OVF specification. Click ”Retry” and it should work. If not, please contact the TAs.

If you have an **Arm processor**, open VMware, click on ”file” and ”Open” and select the file `MininetVM-arm`.

2.1.3 VM CONFIGURATION

Now that the machine is created, we need to add some 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.

Right-click on your VM and click on ”Virtual Machine Settings”, in the ”Options” tab, select ”shared Folders” and select the option ”Always enabled”. Then, click on ”Add”, name the folder ”shared” and click on ”Browse” to select a folder from your host machine. Note that the shared folder is not yet available on the VM. To do this, we need to mount the shared folder to the VM, as explained in Section 2.4.

We also need to 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.

If you are on **Windows or Linux**, right-click on your VM, and select ”Virtual Machine Settings”. Select the ”Network Adapter” tab and make sure the options ”Connect at power on” and ”NAT: Used to share the host’s IP address” are connected.

If you are on **Mac**, click on your VM, then click on the ”Virtual Machine” tab from the VMware menu (top of the screen), and under ”Network Adapter”, select the option ”NAT”.

Congratulations, you have just created a virtual machine!

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

2.2 RUNNING THE VIRTUAL MACHINE

In VMware run your VM by double-clicking it.

Login using the username `lca2` and password `lca2`.

Please note the following: The **x86** VM was configured with an American QWERTY keyboard, while the **Arm** VM was configured with a Swiss QWERTZ keyboard. If the keyboard on your host machine is different from that of the one used by the VM, the characters you type will be different from the ones registered. This can be fixed with the instructions on Section 2.3. For now, note that if you use a French AZERTY keyboard, the password you need to type is `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.

Note for students using the Arm VM: This is the first year that we provide a separate VM for students with an Arm processor. Unfortunately we did not get to test all the lab exercises on this VM, as we did not have a Mac with M1 processor. Therefore, it is possible that some packages needed for the labs are not pre-installed on your VM. If you cannot run a specific command, you can install the relevant package by typing the command

```
sudo apt install <package-name>
```

If you are not sure which package you need to install, please contact a TA. We will post an announcement on moodle, as soon as we notice that some package is missing from the VM.

2.3 KEYBOARD LAYOUT

The keyboard layout on your virtual machine is `us` for **x86** machines and `ch` for **Arm** machines by default. Below, we give instructions on how to change it to the one used by your machine using the Linux command line interface (CLI). However, we noticed that some commands did not work with the Arm VM. In that case, you can also change the keyboard layout using the graphical user interface (GUI).

Open a terminal (LXTerminal, available on the desktop) and type the command (for ch keyboards):

```
$ 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.

2.4 MOUNTING THE SHARED FOLDER

If you have issues accessing the shared folder or copy pasting to/from the VM, type the following commands

```
$ sudo apt update
$ sudo apt install open-vm-tools open-vm-tools-desktop
```

At this point, you should be able to use copy paste; if it does not work, try restarting the VM.

To mount the shared folder, create the `/mnt/hgfs` directory if it does not exist:

```
$ sudo mkdir -p /mnt/hgfs
```

Then, you can mount the shared folder using the command

```
$ sudo mount -t fuse.vmhgfs-fuse .host:/ /mnt/hgfs -o allow_other
```

Now you should be able to access the shared folder (it will appear as a subdirectory of `/mnt/hgfs`). **The shared folder will only be available during this session: you need to mount the shared folder every time you restart the VM.** To make the shared folder persist across sessions add the following line at the end of the `/etc/fstab` file:

```
.host:/ /mnt/hgfs fuse.vmhgfs-fuse auto,allow_other 0 0
```

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

2.5 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.

2.5.1 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 [2]. Use `pwd` to show the current directory. Then `cd` to `~/Tutorial/`. Use the Tab key for auto-complete. If you cannot find the $\sim$ 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
```

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

```
-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
```

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
```

2.5.2 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
```

2.5.3 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 the Mininet VM 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

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

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

RESEARCH EXERCISES (OPTIONAL)

4 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.