POLITECNICO DI TORINO

Lab # 1 on SDN

“Computer network design and control” module of
Communication and network systems

Academic year 2023/24

Andrea Bianco, Paolo Giaccone,
Alessandro Cornacchia, Iman Lotfimahyari, Francesco Malandrino, Matteo Sacchetto

Version: December 1, 2023

Chapter 1

Laboratory #1

The aim of the lab is to experiment using a Mininet SDN network emulator on a Virtual Machine in Crownlabs. Open vSwitch will be used as a software switch within Mininet.

The aim of the lab is to first become familiar with the emulated environment, and then to practice basic and advanced routing problems in a SDN scenario. No SDN controller will be used and forwarding tables will be installed manually.

The lab follows a sequence of incremental steps. Each step is propedeutic to the sequent ones.

1.1 Starting the lab

1. Navigate to the Crownlabs website: https://crownlabs.polito.it/
2. Click on the button ”Login @Polito”
3. On the web page with the login form click on the button ”Polito SSO”, that you can find at the bottom of the form
4. Login in using your Polito credentials (the same credentials you use to access the ”Portale della didattica”).
5. You will now be logged-in in Crownlabs. Here you will find yourself on the ”Dashboard” tap, and on the UI you should see the ”Communication and Network Systems” workspace on the left of the user interface (UI).
6. Click on that workspace and you will see that on the UI a new section appears, where you will find the VM called ”Lab1”
7. Click on the ”Create” button and a new instance of such VM will be created. Once the creation is complete you will see that the ”Connect” button becomes active
8. When the ”Connect” button is active, click on it and a new web browser tab will open up where you will be connected to the VM and you will be able to see the VM desktop.

1.1.1 What to do if you close Crownlabs by mistake

If for some reasons you end up closing the web browser or the web browser tab by mistake, **don’t panic**, the VM instance is still running and you can simply connect again to it.

1. Follow the steps illustrated in Section 1.1 up to the step 6.

2. Once you reach the VM list, under the ”Lab1” VM you will find the VM instance previously created.

3. Click again on the ”Connect” button and you should find yourself back to where you left.

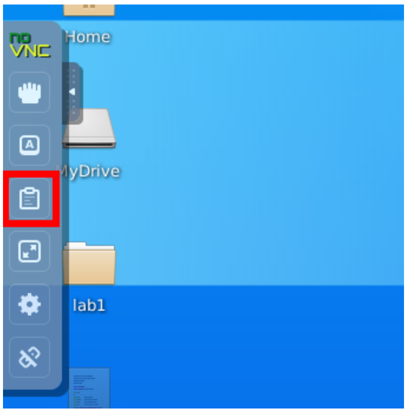
1.1.2 How to copy-paste between Crownlabs and you PC

In order to copy-paste between your PC and the VM instance, it is not as easy as doing CTRL+C and CTRL+V, but some additional step need to be followed.

1. On the left side of the UI you should see a small tab with an arrow. It may be already open, but in case it is not you can open it by clicking on that tab.

2. Once its open you should see a small menu with at the top the title ”no VNC” and below a list of icons

3. The icon illustrated in the picture below is the clipdoard (may be the second or third icon)



4. By clicking on it a text area opens up (click it again to close it). This text area is the mean through which you can copy-paste between Crownlabs and your PC

5. When you select some text in Crownlabs it will appear also in such clipboard, and from that clipboard you will be able to copy-paste to your PC as usual

6. Instead, if you want to copy some text from your PC and paste it in the VM, you need to first to paste it to such clipboad. Once its there, you can then proceed to paste it where you need within Crownlabs

1.1.3 Sharing files between Crownlabs and your PC

If you need to upload a file to the VM or download a file from the VM, you can do so through the drive functionality of Crownlabs. You have 1GB of drive space which can be accessed

- **From the VM**: by opening the file manager and selecting the ”MyDrive” device
- **From the Crownlabs UI**: by selecting the ”Drive” tab at the top of the UI

The files you upload to the drive will be available from within the VM and viceversa. The files you upload to such drive are the only ones which will be stored persistently. Instead, all the files you create and modify within the VM in any other location will be deleted once you delete the VM.

1.1.4 What to do at the end of the lab

Once you finish the lab, we kindly ask you to stop the VM instance, in order to not waste crownlab resources. Before deleting the VM please transfer all the files that you wish to keep in the ”MyDrive” directory (as illustrated in the previous subsection).

1. Close the tab with the VM desktop
2. Navigate back to Crownlabs (repeat all the steps illustrated in Section 1.1 up to the step 6)
3. Near the ”Connect” button of your VM instance you should see the icon of a trash bin. Click on it, and a popup windows will appear
4. In this popup window confirm the deletion of the VM instance by clicking the ”Delete” button

Before beginning the lab, recall that you have a list of the most useful Linux and Mininet commands reference file. Have the reference handy while doing the lab.

1.2 Overview of shell commands

As a preliminary requirement, you must become familiar with both the Linux terminal and the gedit editor, both available from the menu which opens up once you click on the icon you can find at the top left of the desktop.

The most useful terminal commands are reported in the following list. You must practice with each of them at least once before proceeding with the next steps.

Open a terminal. You can either

- select the top left icon and when the window opens select the Icon terminal emulator.
- right click on the destop. In the pop up menu select Open Terminal here.

Recall two important hints when issuing commands in the terminal:

- Use the TAB key to complete file names and commands.
- Use the UP and DOWN arrow to scroll across the previously written commands.

Try the following commands:

1. `man` to get the manual of a command. E.g., `man ls`
2. `pwd` to identify the current directory `pwd`
3. `cd` to change directory. E.g., `cd ~/Desktop/lab1`
4. `ls` to list folder contents. E.g., `ls -lrt`

5. `gedit` to create/edit a file. Type `gedit readme.txt` write something in the file, save and close it.
6. `cat` or `more` to print the file content. E.g., `cat readme.txt` or `more readme.txt` (press q to close it)
7. Type `command1 && command2` (i.e., adding `&&` between two commands) to run two commands in sequence. `command2` will be executed if and only if `command1` finished successfully. E.g., `pwd && ls`.
8. `ifconfig` to show the configuration of the network interfaces.

1.3 Step 1: Getting familiar with the environment

1. Open a new terminal window.
2. We'll be using a network performance tool called `iperf3` to generate synthetic traffic in our network. Read the documentation of the tool by typing `man iperf3`.
3. We'll be emulating a real network by using Mininet. Instead of transferring packets on the real wire, Mininet creates virtual network interfaces and emulates the communication between them. The performance in terms of the bitrate of such an emulated network highly depends on ... (see question below) of the host machine.

   Run the command `sudo mn --test iperf`.

   The command first starts as the root user an instance of Mininet with a simple topology comprising two hosts interconnected via a switch. Then, Mininet performs an `iperf` test between the two hosts and outputs the average data rate between the two hosts. Check the result of your test. You should observe a result like:

```
Results: [’x␣bits/sec’, ’y␣bits/sec’]
```

   What is the meaning of the two values? What do they depend on?

   Repeat the above test 5 times. Write your results

   Now try to run in your system any CPU consuming application (e.g., Youtube video or run on a new terminal `stress-ng --cpu 2 --io 2 --timeout 50`) and re-run the five tests.

   Write your results.

Comment the results.

1.4 Step 2: Single switch topology

1. We start with a simple topology. In the terminal, run this chain of commands:

   ```
   sudo mn --clean && sudo mn --topo=single --switch=ovsk --controller=none --mac --arp
   ```

   The startup may take a few seconds. You'll notice that the network has successfully started when you end up with the following line:

   `mininet>`

   This is the command line interface (CLI) of Mininet and we'll be using it to interact with the emulated network.

2. Inside Mininet CLI type `help`, this will bring up a message with all the available commands in Mininet alongside some hints about their usage. In order to get detailed information about a particular Mininet command type `help [command]`.

3. The first topology contains a certain number of devices, hosts and switches, interconnected via links, and no controller. Use the command `nodes` to identify the available devices. Hosts are denoted as `h{ID}` (e.g., `h1`, `h2`) while switches as `s{ID}` (e.g., `s1`, `s2`).

4. Each switch port (e.g. s1-eth1:1) is associated with a name (s1-eth1) and a port ID (an integer number). These IDs are internally used by switches to define the network interfaces used for packet forwarding. Use the command `ports` to find the association between each network interface and the corresponding name/port ID. Report below all the ports:

   Using the `links` command, draw the test topology, including port name/IDs.

5. Use the `ifconfig` command to see infos on the network interfaces for each node. This can be done from the Mininet CLI (command-line interface) through `[node name] ifconfig` or by opening an individual terminal inside each node using the `xterm [node name]` command and running the `ifconfig` command from inside the node xtermNote: each network node, alongside network interfaces, contains also a loopback interface `lo`, for test purposes, which is always associated with the port_ID `0`.

   Discover the IP addresses of all the hosts. What is the corresponding port ID for each network interface of each node? Redraw the full topology with all the ports and all the IP addresses.

6. Inside the Mininet CLI, check the connectivity between `h1` and `h2` by executing either `h1 ping h2` in the Mininet command line or in a terminal for host `h1` opened by the command `xterm h1` and executing `ping [h2 destination IP address]` in the xterm. Press `Ctrl+C` to terminate the test. Was the test successful and why? Compare with `h1 ping h1`.

7. Define the flow table to route the traffic on the basis of traffic arrival port. E.g., the command `sh ovs-ofctl add-flow s1 in_port=2,actions=output:1` (launched within Mininet CLI) will dictate switch s1 to forward all the traffic from port 2 to port 1. How many entries are needed in the switch `s1` flow table to make ping work? Check the installed flow rules (and related infos) for switch `s1`, by using `sh ovs-ofctl dump-flows s1`. Keep executing `ping` and adding rules until it works.

8. Now configure the forwarding table to route the traffic on the basis of IP flow identifiers. E.g., to route on the basis of the destination IP address, you can use the command

```
sh ovs-ofctl add-flow s1 dl_type=0x800,nw_dst=10.0.0.1/32,actions=output:1
```

   Before adding the new entry in the forwarding table, empty the forwarding table using the command: `sh ovs-ofctl del-flows s1`.

   Report the installed flow rules and test host connectivity. Add rules until the test works properly.

9. Use `tshark` command to run a network protocol analyzer that capture packets. Run `xterm h1 h2 s1` to open terminals on the switch and on the two hosts.

   In the `s1` terminal, run:

```
tshark -i s1-eth1 -q -z ip_srcdst,tree -z conv,udp -z conv,eth
```

   Run `iperf3 -s` as a server in `h2` and run `iperf3 -c 10.0.0.2 -u -b 100k` as a client in `h1` When `iperf3` ends in `h2`, you can see a concise summary of the statistics in both `h1` and `h2`. Now, press `ctrl+c` in `s1` terminal to stop `tshark`. Report a summary of the provided infos by `iperf3` and `tshark`.

10. Use `man tshark` and `man iperf3` to figure out the options used for the above tshark and iperf3 commands, and shortly report their meaning.

11. Compare and report the infos above with the ones you can obtain by running `ifconfig` in hosts and switch.

12. Use the `exit` command to stop the network.

1.5 Step 3: Two switches topology

1. In the terminal, to start the new network run this command:

```
sudo mn --clean && sudo mn --topo=linear --switch=ovsk --controller=none --mac --arp
```

2. Discover and draw the topology with all port numbers.

3. Check the connectivity between the hosts. What should be done to enable communication between the hosts? Report the needed commands.

4. Use the `exit` command to stop the network.

1.6 Step 4: Network with multiple switches

1. To start the new network, in the terminal run the following command:

```
sudo mn --clean && sudo mn --custom=Desktop/lab1/topos.py --topo=topo3 --controller=none
     --switch=ovsk --mac --arp
```

   Note: this command assumes that you are in the home directory (if you run `pwd` the output shoud be `/home/netlab`). If you are not, run the command `cd`, and the run the command above.

2. Discover and draw the topology with all port numbers and device addresses.

3. In the previous step, the routing was defined on the basis of either the incoming port or the flow identifier. Do you think that both are equivalent here? Motivate your answer.

4. Define the flow tables to properly route the traffic among hosts h1 and h2. Write flow tables of the involved switches below.

5. Define the flow tables to properly route the traffic among all the available hosts and check host connectivity. Write the complete flow tables below.

6. Use `exit` command to stop the network.

1.7 Step 5: Multipath routing

1. To start the new network, in the terminal run the following command:

```
sudo mn --clean && sudo mn --custom=Desktop/lab1/topos.py --topo=topo4 --controller=none
     --switch=ovsk --mac --arp
```

   Note: this command assumes that you are in the home directory (if you run `pwd` the output shoud be `/home/netlab`). If you are not, run the command `cd`, and the run the command above.

2. Discover and draw the topology with all port numbers.

3. How many paths exist between the hosts? For each switch define the proper flow rules such that the TCP and UDP flows are routed through different distinct paths. (HINT: you need to use `nw_proto=<proto>` in the flow-rule as well to distinguish between TCP and UDP in the switch).

4. How can you prove that the multipath routing work as expected? HINT: Inspect the proper switch interface using `tshark` ot `ifconfig`.

5. Use `exit` command to stop the network.

1.8 Step 6: Dynamic routing

1. To restart the same network as step 5, again in the terminal run the following command:

```
sudo mn --clean && sudo mn --custom=Desktop/lab1/topos.py --topo=topo4 --controller=none
     --switch=ovsk --mac --arp
```

   Note: these command assumes that you are in the home directory (if you run `pwd` the output shoud be `/home/netlab`). If you are not, run the command `cd`, and the run the command above.

2. Run `sh Desktop/lab1/routes.sh` in the Mininet command-line. This script will install the needed flow tables in `s1`, `s2`, and `s3` to provide a bidirectional route on the `H1-S1-S2-S3-H2` path.

3. Check the rules installed in the mentioned switches and report here:

4. Open an xterminal for `h1` and run `ping -i 0.1 10.0.0.2`. Whenever you stop `ping`, you will get statistics about the number of lost packets (if any). What is doing the option `-i`? Do you experience any losses?

5. Now, run `ping` again and in real-time (i.e., when the ICMP traffic flows) install the required flow rules such that the ICMP traffic flow is redirected to the `H1-S1-S4-S3-H2` path.

6. Check the rules installed in the mentioned switches and report here:

7. Did you experience any loss when the route was changed? Report the number of lost packets (if any) and explain whether losses are permanent or only temporary.

8. Now two cases can occur:

   - If you experienced any loss, what is the reason for that? How can the problem be solved? Report here the revised flow commands that avoid such losses. Check that now no losses are experienced.
   - If you did not experience any loss, what is the reason for that? How could be a wrong way to reroute the traffic that would produce losses? Report here the corresponding flow commands. Check that now if any losses are experienced.

9. Use `exit` command to stop the network.