# Virtualized Infrastructure in Cloud Computing

*Introduction to Software Defined Networks*

Dino López

The main objective of this Lab is to introduce you to the Software Defined Network (SDN) technologies. In order to learn the basics of SDN, this time we will deploy a virtual network with CORE, and manipulate the Flow Table of SDN software devices through the ovs-ofctl command. The virtual switches that we will be using will Open vSwitch (ovs) switches, frequently employed in production Data Centers.

For the following exercises, you can work in pairs (== 2 people teams).

This lab might be graded. Your mark will depend on the quality of the explanations and/or results you are expected to provide for every exercise. Write all your answers in a Microsoft Word or LibreOffice Writer file. Then, you will submit your report in PDF format through the moodle website (https://lms.univ-cotedazur.fr/course/view.php?id=3606).

## 1. Introduction to the SDN administration commands

First of all, we are going to start familiarizing a bit with the ovs-ofctl command, which allows a manual administration of a SDN equipment.

1. Go to the VICC Moodle website (SDN tab) and complete the "ovs-ofctl" command activity.
    a. You have all the session to complete this activity
    b. be sure to reply correctly by visiting the man pages of the ovs-ofctl command: https://manpages.ubuntu.com/manpages/xenial/man8/ovs-ofctl.8.html

## 2. First contact with a SDN switch

Let's try to use the ovs-ofctl command. For the next exercises, please take into account that all SDN switches have been configured to listen for OpenFlow connections at the 6633 port.

2. Installing the POX controller
    a. In your home directory, clone the POX repository with the command "~$ git clone https://github.com/noxrepo/pox.git"
    b. If you successfully cloned the POX repository, you must have now a "pox" directory in your home directory.
3. Deployment of a network testbed

a. As root, execute in a terminal the command "# core-daemon". You must see a few lines saying that the core-daemon is listening on "localhost:4038". Leave the application running.

    i. A simple "$ sudo core-daemon" command will not work as the application will try to launch with the current user's environment variables and not the ones of root.

b. As a standard user, in another terminal, execute the "$ core-gui" command. This command will open a graphical interface where we will deploy our virtual network

c. Download the "2sdnswitches.inm" file and open it with the CORE gui (File->Open-> single-sdnswitch.inm)

d. Finally, launch the network deployment by clicking on the "play" button (available on the top-left of the CORE interface).

4. In this topology, the SDN switches are expected to work in in-band or out-of-band mode? Note that the hosts "hX" are the users' computers and of course, they are not aware about the SDN network/controller. Provide a brief argument.

5. Double-click on the controller to open a terminal. Get the characteristics of "sdn1" using the appropriate ovs-ofctl command.

    a. Briefly say what is number of Flow Tables supported by the SDN switch, the supported actions by that switch and the characteristics of every vNIC of the switch.

    b. The switch identifier is known as the datapath id (dpid) in OpenFlow. Provide the dpid you have found.

    c. Also, provide the command you will use.

6. At this moment, there is no a controller application running on the controller server. Double-click at the controller server to obtain a terminal. Launch the controller with the command "/home/user/pox/pox.py --verbose forwarding.l2_pairs" and verify that both SDN switches connect to the controller. A few seconds might be needed before the SDN switches connect to the controller.

    a. Here, POX executes the l2_pairs network application. In this Net.App., each PacketIn is used to learn the exact location of the sender at the switch (that is, the @MAC - port association) triggering the PacketIn. Hence, once a pair is known (e.g. if the controller knows that node A is reachable through port 1, and node B through port 2), the controller install the proper Flow Entries at the Flow Table of the switch (e.g. Rule 1: forward every packet coming from A and going to B by port 2. Rule 2: forward every packet coming from B going to A by port 1).

b. If you see that the SDN switches connect to the controller, continue with the following exercises. Otherwise, redo carefully the previous steps to solve the problem.

7. Show the Flow Table at switches sdn1 and sdn2. Verify that the Flow Tables are empty. Which command must you execute at the controller server for this aim?

8. Open one terminal per virtual host and with tcpdump, dump the traffic to the terminal (e.g. # tcpdump "arp" -lni eth0, assuming you want to sniff through the eth0 NIC of your host). You must leave tcpdump running. Then, with the arping command (**to install at the VM** if needed "sudo apt install iputils-arping"), execute a request from h1 to h2 (e.g. arping -c1 10.0.0.11) and show the entries at the Flow Table of sdn1 and sdn2.

    a. The arping command above will generate an ARP request (in broadcast mode). Which hosts got the ARP request?

    b. Explain every entry at sdn1 and sdn2, with the deepest details (ignore the cookie parameter however). Feel free to read the ovs-ofctl manpage as needed.

    c. Which host got the ARP reply (unicast mode)?

    d. Based on you understanding of SDN, explain

        i. if the ARP request triggered (and when) a Packet-In, Packet-Out and Flow Mod at sdn1

        ii. if the ARP request triggered (and when) a Packet-In, Packet-Out and Flow Mod at sdn2

        iii. if the ARP reply triggered (and when) a Packet-In, Packet-Out and Flow Mod at sdn1

        iv. if the ARP reply triggered (and when) a Packet-In, Packet-Out and Flow Mod at sdn2

9. Remove the flow entries from sdn1. Which command should you execute?

10. Verify from the controller that all flow entries have been removed. Then, execute 5 ARP requests with the command "arping -c5 10.0.0.11". The output of the command will show the delay between the transmission of the request and the reception of the reply. Explain why the first request gets a much higher delay than the subsequent requests/replies.

If you have finished with this part, execute Ctrl+C to end the execution of POX and close all the terminals from the virtual network, leaving moreover the virtual network running.

## 3. Manual administration of a SDN switch with an external controller

### 1.1. Basics

Now you will be the controller.

11. Download the dummy.py file and save it in /home/user/pox/ext/. Then, execute the POX controller at the controller server: "/home/user/pox/pox.py --verbose dummy".

    a. Open a terminal for h1, h2 and h3, and as for the previous section, run tcpdump.

12. Execute an arping from h1 to h2 with option "-c1". How many packets were captured by h1, h2 and h3? Explain your observations. According to the log messages at the controller, does the arping command generates some PacketIn at the controller?

13. From the controller server, with the ovs-ofctl command, add the following Flow Entry to sdn1 "in_port=1,idle_timeout=30,actions=output:2". Execute again an arping from h1 to h2 with the "-c1" option. How many packets were captured by h1 and h2 and why?

14. How many seconds should you wait for the previous Flow Entry to be removed at sdn1? Wait until the FlowTable is flushed. Then, continue with the next exercise.

15. From the controller server, with the ovs-ofctl command, add the following Flow Entries to sdn1 "in_port=3,actions=output:2" and "in_port=2,actions=output:3". Then, stop the controller with Ctrl+C. Wait a few dozen of seconds, then execute from h3 an arping to h2 ("arping -c1 10.0.0.11").

    a. Explain your observations.

    b. Look at the Flow Table of sdn1 and sdn2. Explain if the switches are configured in secure or standalone mode (Open vSwitch only supports the secure and standalone failures mode), and how you did deduce that.

16. Close all tcpdump processes. Start again the controller with the dummy network application and remove all entries from sdn1.

17. Manually (i.e. using the ovs-ofctl command), emulate a legacy switch at sdn1. A legacy switch builds a Forwarding Data Base (FDB) that maps a port number to a reachable MAC address. This way, when the switch receives a packet with an already learned MAC Address, the frame is immediately forwarded through the appropriate port.

    a. As an example, you must send a rule to forward through port 1 any packet with 00:00:00:00:00:01 destination MAC address.

    b. Broadcasted packets (destination MAC address FF:FF:FF:FF:FF:FF) must be flooded by any port, except the one where the packet is received. Look at the ovs-ofctl manpage how to write a "flood" action and at the website

address http://www.openvswitch.org/support/dist-docs-2.5/ovs-ofctl.8.txt , “Flow Syntax” section, how to match a MAC address (destination or source)

18. Probe with a ping that h1, h2 and h3 are all reachable by each other. Add some screenshots with the flow table rules and the ping output