Chapter 8

APPENDIX A - How to Setup

In this chapter, we will explain the detailed procedures to install and configure all necessary components so that, whoever would like to contribute to this work won't have to deal with the hassle of finding and figuring it out by themselves. This guide was designed based on my experience and my setup: Windows 10 on host machine and Ubuntu 16.04 guest machine

8.1 Mininet and OpenvSwitch

First and foremost we need to install the Ubuntu VM configured with Mininet 2.2.2. Simply navigate to:

https://github.com/mininet/mininet/releases

and download the mininet 2.2.2 Ubuntu 14.04.4 zip file. With Virtualbox, before loading the virtual machine, setup the network components as follows:

1. Adapter 1: NAT, Promiscuous mode: Deny
2. Adapter 2: Bridged Adapter, Promiscuous mode: Allow all
3. Adapter 3: Host-Only Adapter, Promiscuous mode: Allow all
4. On VirtualBox: File->Host Network Manager->create, configure however you'd like mine is 192.168.249.1/24 DHCP server enabled

Personally, I also included a shared folder, where I stored all my code and scripts so that i could easily pass them between host and guest machines. After this start up the VM and login using "mininet" as username and password. I would recommend updating Ubuntu to 16+ if implementing RYU is a concern otherwise keep Ubuntu 14 works fine as well.

Now upgrade the OpenvSwitch version, which is already preinstalled in the VM to the OVS-2.13.9 release, 2.13.0 and 2.14.0 should both work fine. The link for the OVS releases is the following:

https://www.openvswitch.org/download/

Then follow the following instructions to upgrade to the new OVS version directly from the VM:

https://github.com/mininet/mininet/wiki/Installing-new-version-of-Open-vSwitch

NOTE: the configure instruction should be like this:

./configure –prefix=/usr –with-linux=/lib/modules/‘uname -r‘/build –enable-Werror

After the installation has been checked time to link gdb to the ovs-vswitchd process using the following commands:

ps -e | grep ovs-vswitchd gdb ovs-vswitchd [pid from command above] run

Now we can start whichever Mininet Topology we want and all the switches will be, if specified by the command, OVS switches.

8.2 Adding custom actions in OVS

As explained in a previous section, there are only 3 files to modify for the userspace implementation of a new action: lib/ofp-actions.c; include/openvswitch/ofp-actions.h; ofproto/ofproto-dpif-xlate.c. The first 2 files deal mainly with the flow reading and parsing, the best advice i can give is to follow the "AGGRS" string and mimic every instance you find of it with the name of the new action you create. It often

occurs that some functions only require you to add the case but not write any actual code, you cannot skip these functions as the project wouldn't compile. The last file, on the other hand, will contain the actual code of your action which should be implemented in the *do_xlate_actions()* function, again take inspiration from the *OFPAT_AGGRS* cases.

Every time a function is modified or added i recommend using *make* on the whole project and see if anything is missing before continuing, without risking to pile up more errors that are hard to track. Additionally, print to gdb by using the *VLOG_ERR()* function, which works like *printf*, and is very useful for debugging.

8.3 Connecting VM to Internet

In the VM use the command *sudo dhclient eth2* to assign an ip address to the bridged adapter. This is also needed if you want to access internet for installing things through the web. Subsequently, start the mininet topology, select the switch to connect to the internet (as of this guide, this is the only way I found works to connect the topology to the internet), open the component up with *xterm s[#ofswitch]*, and give the command:

ovs-vsctl add-port s# eth2

then, for all hosts in the network (no the switches) set the following route:

ip r add default via 10.0.2.2

Ping any site to check connection, first attempt might take some time. **NOTE:** If the host is directly connected to the internet switch everything will work without further setup, if the host is not directly adjacent, flows need to be setup as previously shown in the testing phase section.

8.4 Testing Phase Setup

To setup the Testing phase run the following commands:

1. Go to the project link on github: https://github.com/L-Mancio/ovs

2. Copy the 3 files needed for the new actions from 1. to the appropriate directories in the VM start from /root/ovs. Then run make and make install, check that make doesn't result in errors. Do this step every time you modify code in any of the files in ovs.

3. Link gdb to ovs-vswitchd process and run

4. On VM go to home directory with *cd* command

5. Navigate in 1. to the directory TestingPhase/TestingPhaseSetup-ML/, copy all files from directory home, and the full directories *customTopo, flows* to the current directory in the VM.

6. Execute the command *sh startMynewtopo.sh*, the topology from the testing phase chapter should be generated, and the mininet shell should be usable.

7. In the mininet shell open s7 with *xterm s7*, add eth2 port by executing in the xterminal *ovs-vsctl add-port s7 eth2*

8. In the mininet shell also give the command source setipr.sh (ignore errors displayed), this will set default routes to all hosts in our topology

9. In the mininet shell execute the flow file according to which action we are testing (i.e. aggregation or splitting) with the command *sh sh flows/mynewtopoaggrflows.sh*

10. H1 and H2 should now be able to access the internet using our custom aggregation action, check by giving the command *h1 links google.it* on the mininet shell. Step is the same for splitting action just use the other flow file

11. **NOTE:** if gdb crashes for any reason, or you want to change flows: quit gdb process using *CTR+C*, then run again and give new flows.

NOTE: if any of the scripts mentioned above return errors that aren't discussed in this procedure, then most likely the fault is in the windows formatting of the script file so just use the command "dos2unix nameoffile.sh" and everything should work fine.