Project 3 - My (Painful) Experience

While each of the installations tasks would have been simple to do individually, combining them all in sequence was not a trivial matter. Overall the installation took me a little over 10 hour in total. Being already familiar with using Python and using REST APIs from the command line, the exercises were much easier for me. They only took me 1 hour.

I dumped a lot of time initially into trying to get my environment set up on my Windows 7 laptop. That didn't go too well on account of the the mininet VM just plain refusing to connect to the host network. I don't know how else to describe it really. I followed the instructions from that blog post linked in the instructions and I still couldn't ping the host machine from inside the mininet virtual machine.

So I threw every obscenity I knew at Bill Gates, calmed down, did some real homework for some other class, and then came back to this with the wonderful idea that I'd just get it done on a Linux system.

I began by choosing the system I'd work with: A virtual machine running elementary OS (basically an Ubuntu Linux 64-bit distro) in Virtual Box 4.3.26. The only reason for using this particular VM was because it was a relatively untouched instance that I had already set up previously. The choice of Linux rather than Windows, my host machine's OS, was simply based on my expectation that working with the toolset prescribed in this project may be easier given these are tools built with developers in mind. There should be less compatibility issues as well as more support found in the online communities for working on my chosen platform.

After almost downloading everything, I ran out of disk space on my virtual machine (10GB fixed size). No matter, I went into VirtualBox and tossed it out like yesterday's trash. I cursed Oracle for making the disk size defaults so small, and then created a new shiny VM with exact same operating system, and with 25GB fixed disk size. FYI, there should probably be something included in the instructions next times mentioning the amount space you'll need for this bloated adventure.

I decided I wasn't going to deal with any prebuilt nonsense anymore after that mininet VM went all stupid on me. So I compiled absolutely everything from source. Everything. Sure, it took me a little longer, but I think I saved myself some time on compatibility gotchas. The only thing I didn't compile from source was Eclipse. I wasn't happy to be using Eclipse for this project. I haven't had very good experiences with it in the past. I knew you could probably just compile the Floodlight project and run controller straight from the bin folder. But I wanted to be on the safe side, just in case there was some weird reason that we could only accomplish this through Eclipse. If I didn't need to do so, that should have also been made clear in the instructions. Cringing and uncontrollable eye twitching aside, I pressed onward and used Eclipse, while reminding myself that there is still much good and harmony in our world. The

setup instructions themselves were fairly straightforward but I think they could have been left out and just linked out to. They seemed to have been taken directly from here: http://www.openflowhub.org/display/floodlightcontroller/Installation+Guide

A short tour through the mininet wiki, its topology documentation, and the Floodlight Firewall REST API (which wasn't linked to in the instructions and was just mysteriously referred to as "The Firewall REST API") was all I needed to get my scripts ready for the exercise section. I think I was actually writing my code before everything was set up since the downloads took such a long time, so that also shortened my time considerably. A couple of errors and debuggings here and there, and then I was done. Easy peasy, lemon squeezy.