

Building the Test Bed

Step 1: Set Up Mininet Server:

- 1.1) Install Virtual Box – OpenFlow Tutorial:
 - Download VirtualBox from <https://www.virtualbox.org/wiki/Downloads>
 - Double click the install file and change install settings as desired
- 1.2) Create Mininet Server– OpenFlow Tutorial
 - Download Mininet 2.1.0 from <https://github.com/Mininet/Mininet/wiki/Mininet-VM-Images>
 - Unzip the downloaded, compressed Mininet file, it should become a folder containing two files
 - Open VirtualBox and click New -> Continue -> Name the virtual machine, select Linux/Ubuntu for the operating system, 512MB of memory, Continue -> Use Existing Hard Disk -> Select the Hard Disk -> Add -> Find and select the downloaded Mininet VDMK file, Continue -> Done
 - Select the newly created VM and click Settings -> Network -> Adapter 2 -> Select Enable Adapter and make the adapter a Host-Only Network connection. If you run into problems with this step it may be because VirtualBox doesn't have a host-only connection. If that is the case go File -> Preferences -> Network and add a host-only connection.
 - Start the virtual machine, when the virtual machine has finished booting, the login and password are both "mininet"
 - Enter the command "sudo dhclient eth1"
 - Open a terminal window on your host machine and enter the command "ssh -X mininet@<VM's IP address for eth1>". This should secure shell into the Mininet server and allow an easy connection to the VM from the host machine.
- 1.3) Install MiniEdit - <http://techandtrains.com/category/miniedit/>
 - Download MiniEdit 2.1.0.9 python script from <https://app.box.com/s/c484eu35yiczq2msj45>
 - Move the downloaded python script to the Mininet VM, I accomplished this by using a dropbox account, but any method that moves the python script to the VM is fine.
 - The MiniEdit python script can be run as-is to build Mininet python scripts
- 1.4) Patch iperf - <http://iperf.fr/> | <http://sourceforge.net/p/iperf/patches/8/> | <http://prasadlinuxblog.wordpress.com/2012/08/15/step-by-step-guide-installation-of-iperf-benchmarking-tool/>
 - Download iperf 2.0.5 source code from <http://iperf.fr/> and iperf-detti-poisson-2.0.4.patch from <http://sourceforge.net/p/iperf/patches/8/>
 - Move both files to the Mininet VM
 - On the Mininet VM enter the command "sudo apt-remove iperf"
 - Extract the iperf source code from the compressed file on the Mininet VM
 - Move the patch file into iperf source code directory so it is in the same directory as include, src, etc.
 - Enter the command "sudo patch -p0 < iperf-detti-poisson-2.0.4.patch"
 - Enter the following commands:
 - "make distclean"
 - "./configure"
 - "make"
 - "make install clean"
 - "make install"

Step 2: Install and Modify Floodlight:

- 2.1) Build Floodlight – <http://networkstatic.net/tutorial-to-build-a-floodlight-sdn-openflow-controller-module/>

Run the following commands from the host machine's CLI:

- “sudo apt-get install build-essential default-jdk ant python-dev eclipse”
- “git clone git://github.com/floodlight/floodlight.git”
- “cd floodlight”
- “ant eclipse”

- 2.2) Patch Floodlight – <https://github.com/andi-bigswitch/floodlight-oss/commit/98752d588cbc419566201de3d9dc19140f48c1ba>
- Change Floodlight's classes to reflect the changes shown in the commit at the website shown above, changes to classes in /src/test/java/ do not need to be made unless those tests will be used
 - Lines 191-197 of src/main/java/net/floodlightcontroller/staticflowentry/StaticFlowEntries.java removed by the commit must remain in the class for Floodlight to run properly
- 2.3) Add Custom Modules and Classes – See Attachments | <http://networkstatic.net/tutorial-to-build-a-floodlight-sdn-openflow-controller-module/>
- The source code for the RulePlacer and TopoStats modules as well as their associated classes should be in documents accompanying this Build Guide.
 - These classes should be added to a package in /src/main/java/net/floodlightcontroller/
 - RulePlacer and TopoStats must be added to the Floodlights default startup modules. To do this they must be added to the files: src/main/resources/floodlightdefault.properties and src/main/resources/META-INF/services/net.floodlight.core.module.IFloodlightModule
 - More detailed instructions can be found at the link provided above

Step 3: Create Graphing Program:

- 3.1) Download and Install JFreeChart Libraries - <http://www.jfree.org/jfreechart/> | <http://www.jfree.org/phpBB2/viewtopic.php?t=20743> | <http://www.ivoronline.com/Coding/Languages/JAVA/APIS/JAVA%20-%20JFreeChart%20-%20Install%20On%20-%20Eclipse.php>
- Download JCommon from <http://sourceforge.net/projects/jfreechart/files/3.%20JCommon/>
 - Download JFreeChart from <http://sourceforge.net/projects/jfreechart/files/1.%20JFreeChart/1.0.19/>
 - Unzip the downloads
 - In Eclipse select Window -> Preferences -> expand Java -> expand Build Path -> User Libraries -> New -> name the library “JFreeChart” -> Add External Jar -> ~/jcommon-1.0.23/jcommon-1.0.23.jar -> Edit Source Attachment -> External Location: ~/jcommon-1.0.23/src
 - Add another External Jar file to the JFreeChart user library: ~/jfreechart-1.0.19/lib/jfreechart-1.0.19.jar with the source attachment at ~/jfreechart-1.0.19/src
- 3.2) Add Graphing Classes – See Attachment
- In the same workspace as Floodlight create a new project to contain the graphing classes
 - Add the classes NetGraphs, DataSummarizer, and DataPoint to the new project
 - Right click on the project and select Build Path -> Add Libraries -> User Library -> Select the library that was created with jcommon and jfreechart -> Finish

Additional Considerations

- This guide should only be used for a Linux host machine. Floodlight can also be built on Windows, but it is much more complicated. An easy way to use Floodlight in Windows is to build it on a Linux machine and then export it as an Eclipse archive file. This can then be transferred to a Windows machine, unzipped and imported into eclipse as a file system.
- When link bandwidth is set in MiniEdit, the links it creates in the python script contain errors. Compare the MiniEdit-created python scripts to working python scripts to see how to correct these errors. Corrected python scripts that were originally generated using MiniEdit should be included with this build guide.