Lab Basics for SDN Homework

CSCE 665

SDN basics

- SDN controller: allows users to program applications to control SDN/OpenFlow networks.
- SDN switches: process network packets according to instructions from SDN controllers.

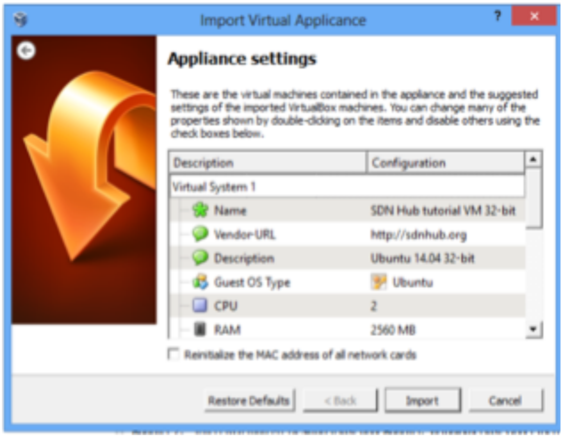
In this homework, you need to write SDN/OpenFlow applications to realize different network management tasks.

Preliminary

- You can use a VM image to simulate the lab environment.
  - –http://sdnhub.org/tutorials/sdn-tutorial-vm/
- In this lab, we will use
  - –Floodlight (SDN controller)
  - –FRESCO (Development Framework)
  - –Mininet (SDN network simulator)

Part1: Import the VM

- Double click the image (.OVA file).
- Click Import



Appliance settings

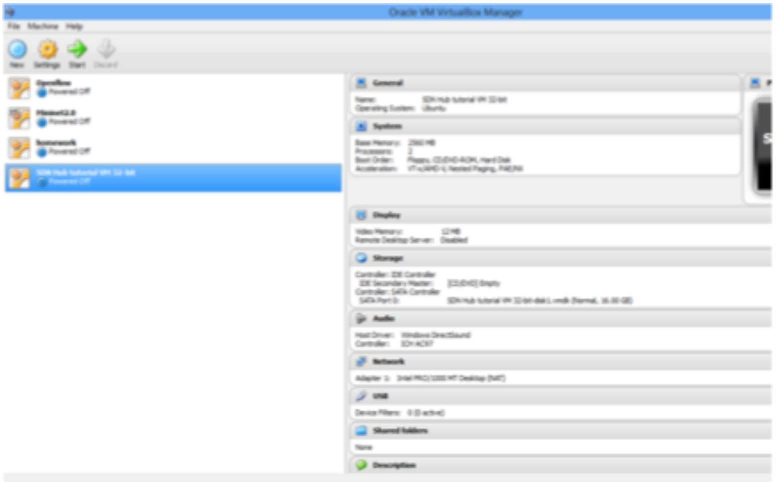
These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Description	Configuration
Virtual System 1	
Name	SDN Hub tutorial VM 32-bit
Vendor-URL	http://sdnhub.org
Description	Ubuntu 14.04 32-bit
Guest OS Type	Ubuntu
CPU	2
RAM	2560 MB

□ Reinitialize the MAC address of all network cards

Part1: Start the VM

- Choose the SDN Hub tutorial VM 32-bit, Click Start.



Name: SDN Hub tutorial VM 32-bit
Operating System: Ubuntu

Base Memory: 2560 MB
Processors: 2
Boot Order: Floppy, CD/DVD-ROM, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX

Video Memory: 12 MB
Remote Desktop Server: Disabled

Controller: IDE Controller
IDE Secondary Master: [CD/DVD] Empty
Controller: SATA Controller
SATA Port 0: SDN Hub tutorial VM 32-bit-disk1.vmdk (Normal, 16.00 GB)

Host Driver: Windows DirectSound
Controller: ICH AC97

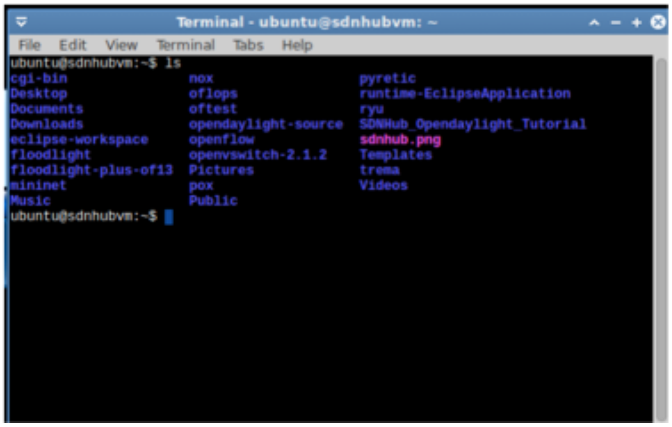
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

Device Filters: 0 (0 active)

None

Part1: About This Image

- In this VM image, many tools are already pre-installed. If you are interested in SDN research. This image is very useful.


Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help

```
ubuntu@sdnhubvm:~$ ls
cgi-bin              nox                    pyretic
Desktop              oflops                 runtime-EclipseApplication
Documents            oftest                 ryu
Downloads            opendaylight-source    SDNHub_Opendaylight_Tutorial
eclipse-workspace    openflow               sdnhub.png
floodlight           openvswitch-2.1.2      Templates
floodlight-plus-of13 Pictures               trema
mininet              pox                    Videos
Music                Public
ubuntu@sdnhubvm:~$ 
```

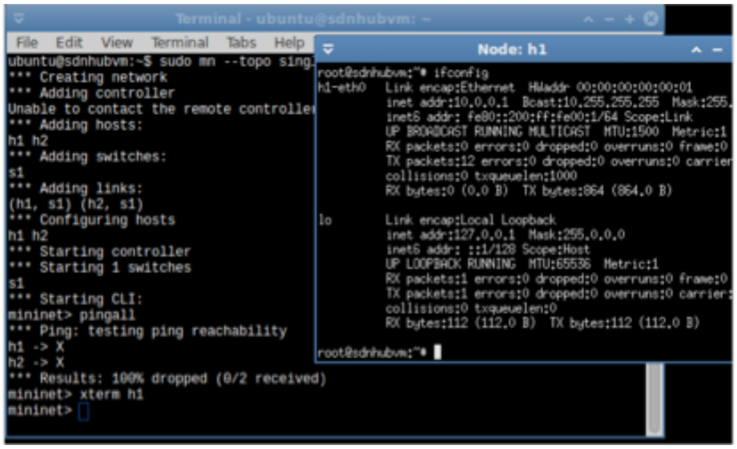
Part2: Simulate an OpenFlow network

- A tool called **Mininet** is pre-installed.
- Open a Terminal and you can use Mininet command.
  - e.g. **sudo mn --topo single,2 --mac --controller=remote, ip=127.0.0.1,port=6653**
  - It is to create a network with one switch and two hosts, the simulated network will contact a remote controller with IP address 127.0.0.1 and port 6653(the default port of Floodlight controller).

- For more Mininet usage, you can refer to the OpenFlow tutorial
  - http://mininet.org/walkthrough/

Part2: More Commands in Mininet

- **pingall**, **h1 ping h2**
- **xterm** (Open a terminal for simulated devices, e.g., switches or hosts)


Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help
Node: h1

```
ubuntu@sdnhubvm:~$ sudo mn --topo sing
*** Creating network
*** Adding controller
Unable to contact the remote controlle
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet> xterm h1
mininet>
```

```
root@sdnhubvm:~# ifconfig
h1-eth0   Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.
          inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:864 (864.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:
          collisions:0 txqueuelen:0
          RX bytes:112 (112.0 B)  TX bytes:112 (112.0 B)

root@sdnhubvm:~#
```

Part3: Install Floodlight

You have two options to install Floodlight SDN controller from source code

- Using git
  - –repo: https://github.com/floodlight/floodlight.git
  - –use command: git clone https://github.com/floodlight/floodlight.git
- Download source code
  - –go to the download page: http://www.projectfloodlight.org/download/
  - –Download Floodlight v1.2 and decompress the source code

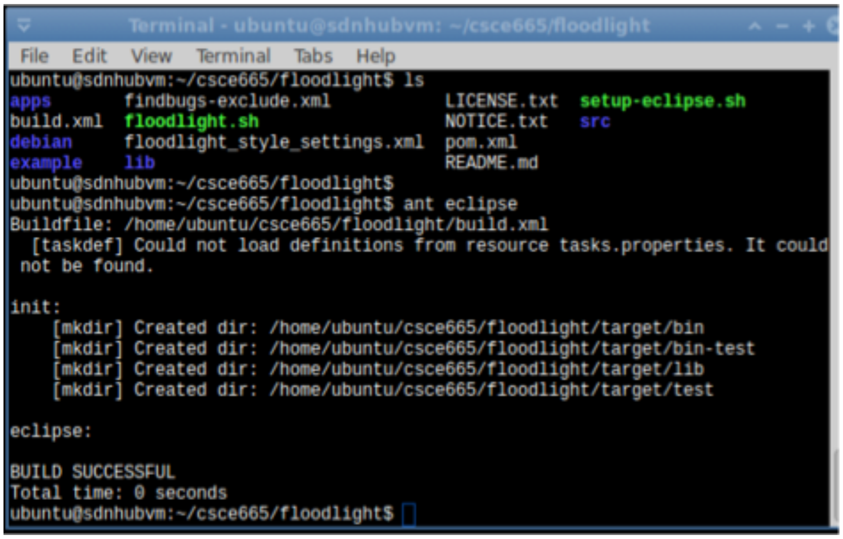
To compile Floodlight, you need to install **ant** tool:

- in the terminal, type in command “**sudo apt-get install ant**”

Compile Floodlight

- go to the folder of Floodlight controller
- type in command “**ant eclipse**”

Install Floodlight


Terminal - ubuntu@sdnhubvm: ~/csce665/floodlight
File Edit View Terminal Tabs Help

```
ubuntu@sdnhubvm:~/csce665/floodlight$ ls
apps       findbugs-exclude.xml            LICENSE.txt  setup-eclipse.sh
build.xml  floodlight.sh                   NOTICE.txt   src
debian     floodlight_style_settings.xml   pom.xml
example    lib                             README.md
ubuntu@sdnhubvm:~/csce665/floodlight$
ubuntu@sdnhubvm:~/csce665/floodlight$ ant eclipse
Buildfile: /home/ubuntu/csce665/floodlight/build.xml
  [taskdef] Could not load definitions from resource tasks.properties. It could
 not be found.

init:
    [mkdir] Created dir: /home/ubuntu/csce665/floodlight/target/bin
    [mkdir] Created dir: /home/ubuntu/csce665/floodlight/target/bin-test
    [mkdir] Created dir: /home/ubuntu/csce665/floodlight/target/lib
    [mkdir] Created dir: /home/ubuntu/csce665/floodlight/target/test

eclipse:

BUILD SUCCESSFUL
Total time: 0 seconds
ubuntu@sdnhubvm:~/csce665/floodlight$
```

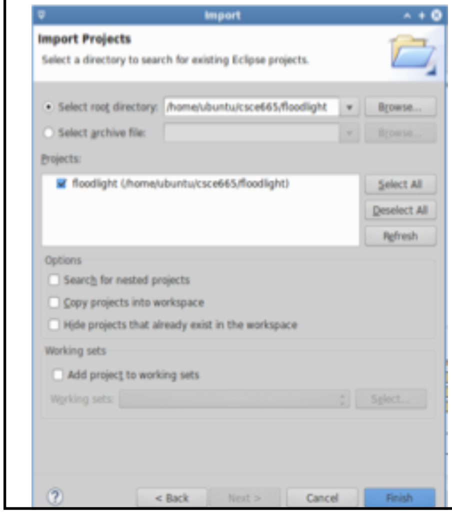
Part3: Run Floodlight

How to run Floodlight Controller in Eclipse.

- First, you should make sure successfully compiling the source code of Floodlight with command “ant eclipse”
- Load Floodlight into Eclipse
  - –Open Eclipse in the desktop.
  - –File -> Switch Workspace -> other -> create a new workspace
  - –File -> Import, choose General -> Existing Projects into Workspace.
  - –Click Next. Select root directory, choose the fold of Floodlight and Click Finish.
  - –Then you can see the floodlight project is in your Eclipse Workspace.
- Run Floodlight
  - –Click Run->Run Configurations>JAVA application> Choose **Floodlight-Default-Conf**
  - –Click Apply.
  - –Click Run, then the floodlight controller is running.

Part3: Run Floodlight (Cont.)

Load existing Floodlight project



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: /home/ubuntu/csce665/floodlight

○ Select archive file:

Projects:

☑ floodlight (/home/ubuntu/csce665/floodlight)

Options

☐ Search for nested projects

☐ Copy projects into workspace

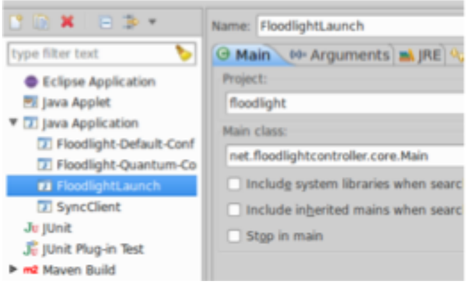
☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

Working sets:

Run Configuration



Name: FloodlightLaunch

Project: floodlight

Main class: net.floodlightcontroller.core.Main

☐ Include system libraries when searc

☐ Include inherited mains when searc

☐ Stop in main

Part4: Install FRESCO

You can install and compile FRESCO from source code

- Using git command:
  git clone https://github.com/xuraylei/fresco_floodlight.git

Compile FRESCO

- go to the folder of FRESCO (called fresco_floodlight)
- type in command “**ant eclipse**”

Enable application in FRESCO

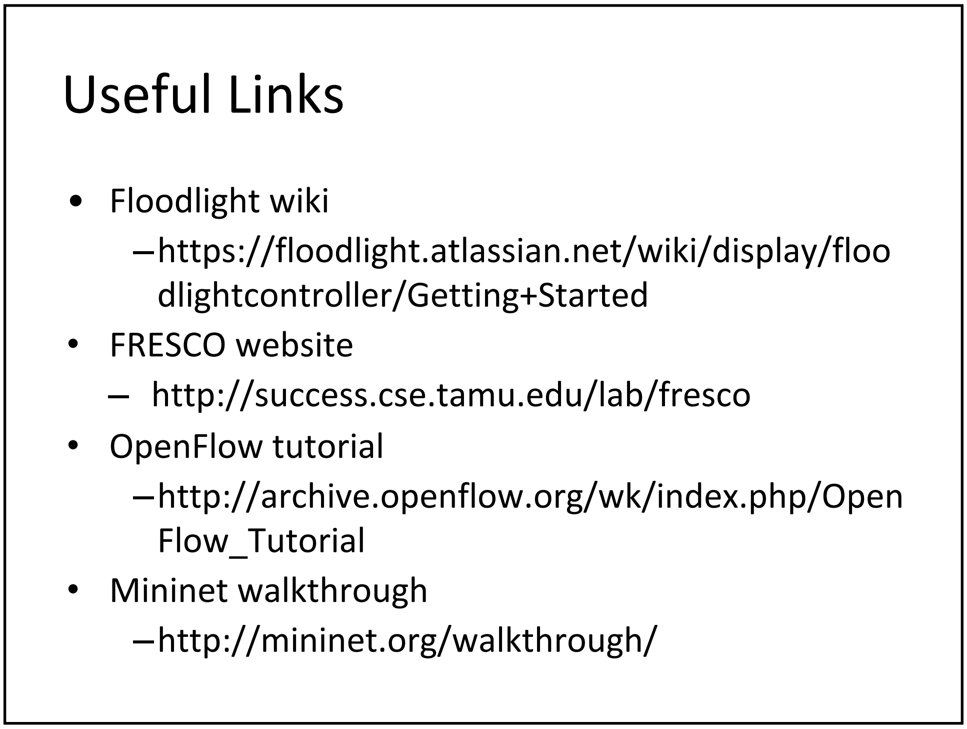
- go to the folder of FRESCO and place the FRESCO application (.fre) you want to run into the folder of *fresco_apps/enable*.

For more installation information, you can refer to:
http://success.cse.tamu.edu/fresco/document/howtoinstall.php

Part3: Run FRESCO

How to run FRESCO in Eclipse.

- First, you should make sure successfully compiling the source code of FRESCO with command “ant eclipse”
- Load FRESCO into Eclipse
  - –Open Eclipse in the desktop.
  - –File -> Switch Workspace -> other -> create a new workspace
  - –File -> Import, choose General -> Existing Projects into Workspace.
  - –Click Next. Select root directory, choose the fold of FRESCO and Click Finish.
  - –Then you can see the FRESCO project is in your Eclipse Workspace.
- Run FRESCO
  - –Click Run->Run Configurations>JAVA application> Choose **Floodlight-Default-Conf**
  - –Click Apply.
  - –Click Run, then the FRESCO is running.



Useful Links

- Floodlight wiki
  - https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Getting+Started
- FRESCO website
  - http://success.cse.tamu.edu/lab/fresco
- OpenFlow tutorial
  - http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial
- Mininet walkthrough
  - http://mininet.org/walkthrough/