

HKA Assignment 5

Hayder Al Rubaye
T00613298

INTRODUCTION TO OPENSWITCH WITH FLOODLIGHT CONTROLLER

We first login to the GENI using the NCSA account. As seen in Figures 1-3

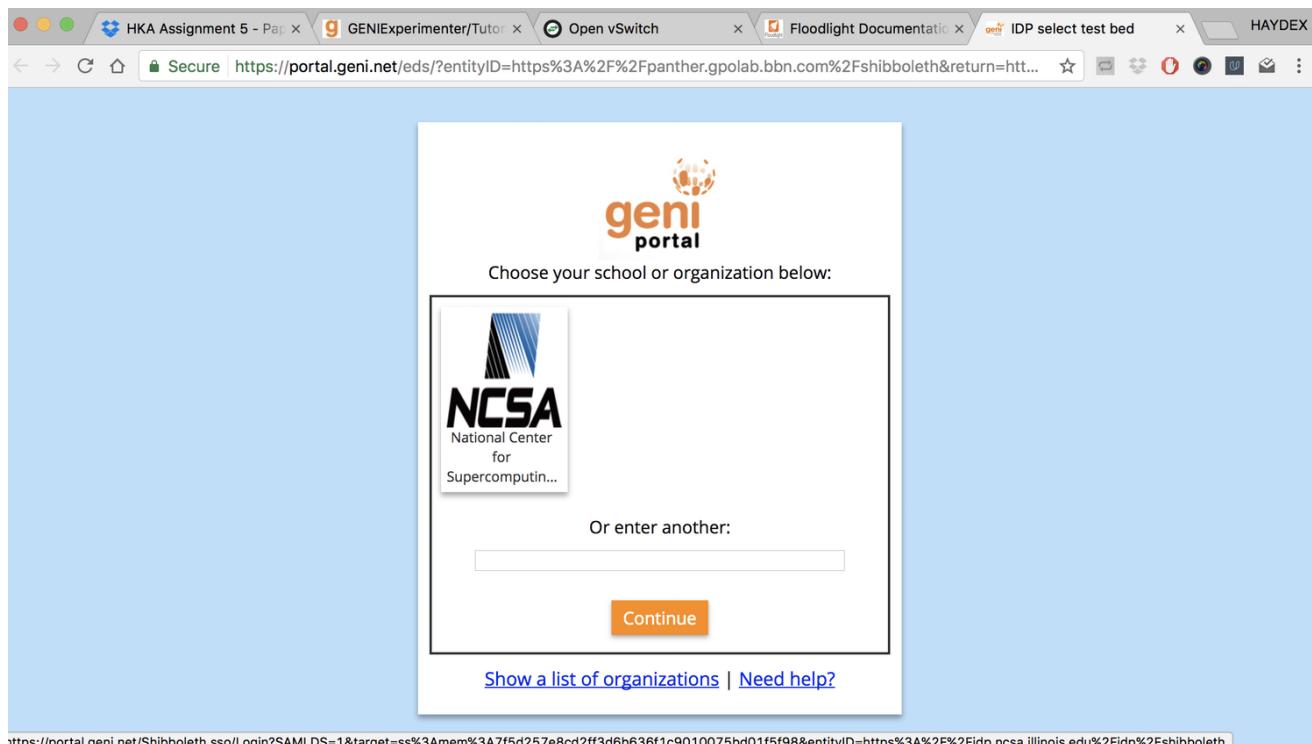


Figure 1

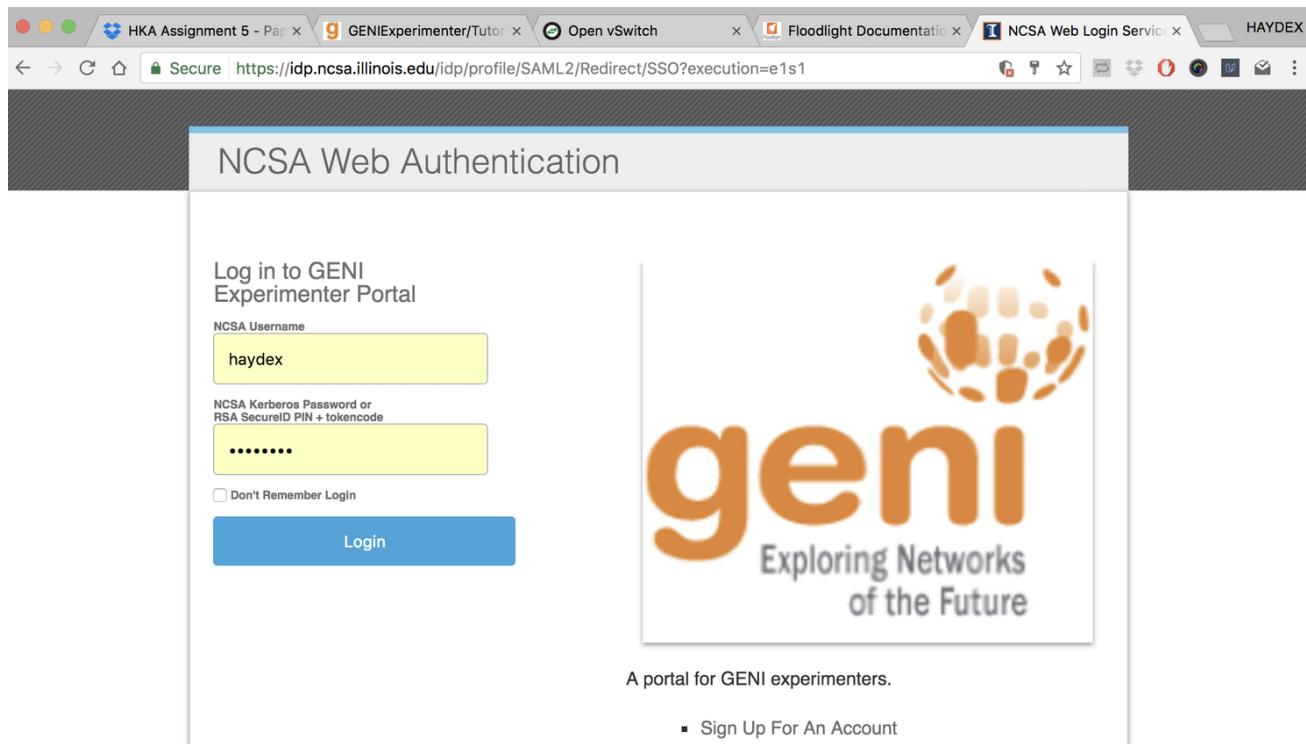


Figure 2

The screenshot shows a web browser window titled "HKA Assignment 5" with the URL <https://portal.geni.net/secure/dashboard.php>. The title bar also includes tabs for "GENIExperimenter/Tutor", "Open vSwitch", "Floodlight Documentation", and "GENI Portal: Home". The main content area is titled "GENI Portal" and shows a navigation bar with "Home", "Tools", "Partners", "Help", and a user profile for "Hayder Al Rubaye". Below the navigation bar is a dark grey header with tabs for "Slices", "Projects", "Logs", "Map", and "Status", where "Slices" is highlighted. The main content area is titled "Slices" and includes filters for "Filter by: UALR-CPSC-7341", "Sort by: Slice name", and "Sort ascending" (with a checked checkbox). It features two buttons: "+ New slice" and "Manage project". A message states "No slices for project UALR-CPSC-7341". At the bottom right, there is a footer with copyright information: "GENI Portal Version 3.23", "Copyright © 2016 Raytheon BBN Technologies", "All Rights Reserved - NSF Award CNS-0714770", and "GENI is sponsored by the National Science Foundation".

Figure 3

Step 1: We obtain the resources.

- a) We create the controller slice and reserve a VM that runs OpenFlow controller by loading its RSpec file from the given link. (Figures 4-8)

Home → Project UALR-CPSC-7341 → Create Slice in project UALR-CPSC-7341

Create New Slice

A GENI slice is a container for reserving and managing a set of GENI resources.

Project name	UALR-CPSC-7341
Slice name	controller -- Required
Slice description	

Note: Slice names must not contain whitespace. Use at most 19 alphanumeric characters or hyphen (no leading hyphen) : "a-zA-Z0-9-".

Note: Slice names are public and must be unique across your project.

Create slice **Cancel**

GENI Portal Version 3.23

Figure 4 Controller Slice

Secure | https://portal.geni.net/secure/slice.php?slice_id=db70115b-41ac-48cf-9610-7fd06c07599d

Notice
Created slice controller

Slice: controller Slice expires in **6 days** ✓
Project: [UALR-CPSC-7341](#) Project expires in **25 days** ✓

Manage Resources

No resources initialized

View Rspec

Figure 5

Figure 6

Figure 7 RSpec loaded

The screenshot shows the GENI Portal interface. At the top, there are several tabs: 'HKA Assignment 5', 'GENIExperimenter', 'Open vSwitch', 'Floodlight Document', 'GENI Portal: Slices', 'GENI Portal: Add Re...', 'YANNI Prelude and ...', and 'HAYDEX'. Below the tabs, the URL is https://portal.geni.net/secure/slice-add-resources-jacks.php?slice_id=db70115b-41ac-48cf-9610-7fd06c07599d#.

The main content area has a title 'Add Resources to GENI Slice controller'. Below it, a sub-section titled 'Add Resources' contains the instruction: 'To add resources you need to draw or choose a Resource Specification (RSpec)'. A modal window titled 'Illinois InstaGENI' is open, showing a list of sites. The 'Illinois InstaGENI' site is selected and highlighted with a red border. To the right of the modal, there is a 'Controller' icon and the text 'Illinois InstaGENI'.

Figure 8

The screenshot shows the GENI Portal interface. At the top, there are several tabs: 'HKA Assignment 5', 'GENIExperimenter', 'Open vSwitch', 'Floodlight Document', 'GENI Portal: Slices', 'GENI Portal: Slices', 'GENI Portal: Slices', and 'HAYDEX'. Below the tabs, the URL is https://portal.geni.net/secure/listresources.php?slice_id=db70115b-41ac-48cf-9610-7fd06c07599d&am_id[]>=267.

The main content area has a title 'Resources on slice: controller'. It shows a message 'Queried 1 of 1 aggregates.' and two buttons: 'Refresh All Details' and 'Refresh Status Only'. Below this, a table shows aggregate status: one entry for 'READY' status under 'Aggregate' column, corresponding to 'Illinois InstaGENI'. Under 'Aggregate Illinois InstaGENI's Resources:', it shows 'Node #1:' with a table of resources. The table has columns: Status, Client ID, Component ID, Expiration, Type, and Hostname. It lists three entries: a 'READY' entry for 'Controller' type with Client ID 'pc1', and two 'Login' entries for 'ssh' type with Client IDs 'mxxie@pcvm1-34.instageni.illinois.edu', 'hkalruba@pcvm1-34.instageni.illinois.edu', and 'txhan@pcvm1-34.instageni.illinois.edu'.

Figure 8 Continued Resources allocated successfully

Slice: controller
Project: UALR-CPSC-7341

Slice expires in 6 days (green)
Project expires in 25 days (green)

Add Resources Renew Update SSH Keys Tools

Manage Resources

Resources on Illinois InstaGENI are ready.

View RSpec

Controller

Renew Renew Date Delete SSH Restart Snapshot Details Add Resources Expand

GENI Portal Version 3.23
Copyright © 2012 Raytheon BBN Technologies
All Rights Reserved - NSF Award CNS-0714770

Figure 8 Continued

b) We create the network topology slice which includes a VM with OVS installed and the endpoint hosts by loading its RSpec file from the given link. (Figures 9-13)

Home → Project UALR-CPSC-7341 → Create Slice in project UALR-CPSC-7341

Create New Slice

A GENI slice is a container for reserving and managing a set of GENI resources.

Project name	UALR-CPSC-7341
Slice name	network -- Required
Slice description	

Note: Slice names must not contain whitespace. Use at most 19 alphanumeric characters or hyphen (no leading hyphen) : "a-zA-Z0-9".

Note: Slice names are public and must be unique across your project.

Create slice Cancel

GENI Portal Version 3.23
Copyright © 2016 Raytheon BBN Technologies
All Rights Reserved - NSF Award CNS-0714770
GENI is sponsored by the National Science Foundation

Figure 9

The screenshot shows the GENI Portal interface. At the top, there are several tabs: "HKA Assignment 5 - Paper", "GENIExperimenter/Tutorials/O", "Open vSwitch", "Floodlight Documentation - Flo", "GENI Portal: Slices", and "HAYDEX". The main content area has a "Notice" box stating "Created slice network". Below this is a navigation bar with links: Resources, Aggregates, Map, Members, Info, and Logs. A summary box displays "Slice: network" and "Project: UALR-CPSC-7341". It also shows "Slice expires in 7 days" and "Project expires in 25 days". Buttons for "Add Resources", "Renew", "Update SSH Keys", and "Tools" are present. The central area is titled "Manage Resources" and contains a message: "Jacks initialized: no resources". A "View RSpec" button is located in the top right of this section. At the bottom, there are buttons for "Renew", "Renew Date", "Delete", "SSH", "Restart", "Snapshot", "Details", "Add Resources", and "Expand".

Figure 10

The screenshot shows the GENI Portal interface for adding resources. The top navigation bar is identical to Figure 10. The main content area features a "Drag to Add" panel on the left with icons for VM, Xen VM, EG VM, Raw PC IG, Raw PC EG, and OF OVS, along with a "New Site" button. To the right, a network diagram titled "Site 1" shows three hosts (Host1, Host2, Host3) connected to a switch. A "Delete All", "Tidy View", and "View RSpec" button is at the top right of the diagram area. Below the diagram, a "Choose RSpec" section allows loading from a URL (with "https://raw.githubusercontent.com" entered), and an "Add To Existing" checkbox. The "Save RSpec" section includes a "Download RSpec" button. The "Editor Ops" section has buttons for "Expand", "Duplicate Nodes only", "Auto IP", and "Add Global Node".

Figure 11 Network RSpec is loaded

The screenshot shows the GENI Portal's 'Add Resources' interface. In the top left, there's a 'Name' input field containing 'Illinois InstaGENI'. To its right is a 'Site' dropdown menu with a list of available sites, including '(any)', Apt, Case Western InstaGENI, CENIC InstaGENI, Clemson InstaGENI, Cornell InstaGENI, Georgia Tech InstaGENI, GPO InstaGENI, Illinois InstaGENI (which is selected and highlighted in black), Kansas InstaGENI, and Kentucky InstaGENI. On the far right of the header are buttons for 'Delete All', 'Tidy View', and 'View RSpec'. Below the header, a message says 'To add resources you need to draw or choose a Resource Specification (RSpec)'. A diagram on the right shows three hosts (Host1, Host2, Host3) connected to a central 'Switch'. At the bottom, there are tabs for 'Choose RSpec' (selected), 'Portal', 'File', 'URL', and 'Text Box', along with a 'Load from URL' input field and an 'Add To Existing' checkbox.

Figure 12 Reserving the resources from Illinois InstaGENI



[Home](#) → Project [UALR-CPSC-7341](#) → Slice network → Resources on Slice network

Resources on slice: network

Queried 1 of 1 aggregates.

[Refresh All Details](#) [Refresh Status Only](#)

Status	Aggregate
READY	Illinois InstaGENI

Aggregate Illinois InstaGENI's Resources:

Node #1:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Host1	pc1	2017-05-13T01:34:16.000Z	emulab-xen	Host1.network.ch-geni-net.instageni.illinois.edu
Login			ssh_mxxie@pc1.instageni.illinois.edu -p 25074 ssh_hkalruba@pc1.instageni.illinois.edu -p 25074 ssh_txhan@pc1.instageni.illinois.edu -p 25074		
Interfaces				MAC	Layer 3
interface-6	pc1:lo0	0265d7e95af5			ipv4: 10.0.0.1

Node #2:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Host2	pc1	2017-05-13T01:34:16.000Z	emulab-xen	Host2.network.ch-geni-net.instageni.illinois.edu
Login			ssh_mxxie@pc1.instageni.illinois.edu -p 25075 ssh_hkalruba@pc1.instageni.illinois.edu -p 25075 ssh_txhan@pc1.instageni.illinois.edu -p 25075		
Interfaces				MAC	Layer 3
interface-8	pc1:lo0	02c33308c094			ipv4: 10.0.0.2

Node #3:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Host3	pc1	2017-05-13T01:34:16.000Z	emulab-xen	Host3.network.ch-geni-net.instageni.illinois.edu
Login			ssh_mxxie@pc1.instageni.illinois.edu -p 25076 ssh_hkalruba@pc1.instageni.illinois.edu -p 25076 ssh_txhan@pc1.instageni.illinois.edu -p 25076		
Interfaces				MAC	Layer 3
interface-10	pc1:lo0	02939a4e7789			ipv4: 10.0.0.3

Node #4:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Switch	pc1	2017-05-13T01:34:16.000Z	emulab-xen	Switch.network.ch-geni-net.instageni.illinois.edu
Login			ssh_mxxie@pc1.instageni.illinois.edu -p 25077 ssh_hkalruba@pc1.instageni.illinois.edu -p 25077 ssh_txhan@pc1.instageni.illinois.edu -p 25077		
Interfaces				MAC	Layer 3
interface-7	pc1:lo0	025f8c2737e6			ipv4: 10.10.1.2
interface-9	pc1:lo0	02731746ece7			ipv4: 10.10.2.2
interface-11	pc1:lo0	02ab11b7eeec			ipv4: 10.10.3.2

Link #1:

Client ID	Endpoint #0	Endpoint #1
link-3	interface-6	interface-7

Link #2:

Client ID	Endpoint #0	Endpoint #1
link-4	interface-8	interface-9

Link #3:

Client ID	Endpoint #0	Endpoint #1
link-5	interface-10	interface-11

[Show Raw XML Resource Specification \(Manifest\)](#)

[Show Raw JSON Resource Status](#)

[Show Ansible Inventory](#)

[Back to All slices](#)

[Back to Slice network](#)

Figure 13 Resources allocated successfully

The screenshot shows the GENI Portal interface. At the top, there are several tabs: HKA Assignment 5, GENIExperimenter, Open vSwitch, Floodlight Document, GENI Portal: Slices, GENI Portal: Slices, and GENI Portal: Slices. The URL in the address bar is https://portal.geni.net/secure/slice.php?slice_id=ca588202-474a-435c-b6c7-bd64cde5b91b. The main header says "GENI Portal". Below it, there's a navigation bar with Home, Tools, Partners, Help, and a user name "Hayder Al Rubaye". The main content area has a tab bar with Resources (which is selected), Aggregates, Map, Members, Info, and Logs. It displays information about a slice: "Slice: network" and "Project: UALR-CPSC-7341". It also shows expiration dates: "Slice expires in 6 days" and "Project expires in 25 days". There are buttons for "Add Resources", "Renew", "Update SSH Keys", and "Tools". A section titled "Manage Resources" contains a message "Resources on Illinois InstaGENI are ready." and a network diagram. The diagram shows three hosts (Host1, Host2, Host3) connected to a central "Switch". Below the diagram are buttons for "Renew", "Renew Date", "Delete", "SSH", "Restart", "Snapshot", "Details", "Add Resources", and "Expand". At the bottom right of the main content area, there's a link "View Rspec". The footer of the page includes the text "GENI Portal Version 3.23" and "Copyright © 2016 Raytheon BBN Technologies".

Figure 13 Continued

Step 2: Configuring the Open vSwitch

Although OVS is installed and initialized on the host that acts as a software switch, it has not been configured yet. There are two configuration operations that need to be executed: 1) configure the software switch with the interfaces as ports and 2) point the OVS switch to an OpenFlow controller.

2a. Configure the Software Switch (OVS Window)

- i. We login to the OVS host by using the GENI Desktop, then we connect through SSH in the browser (Figures 14-16)

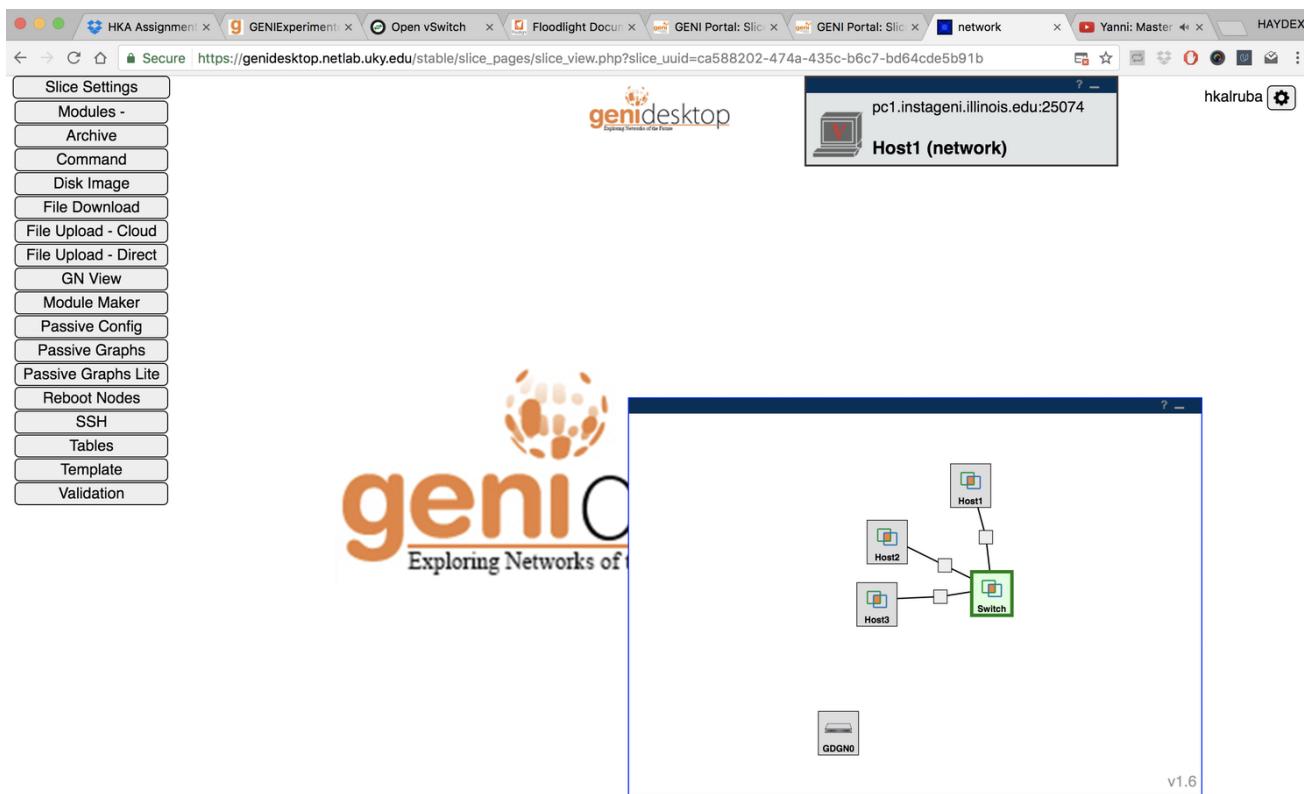


Figure 14

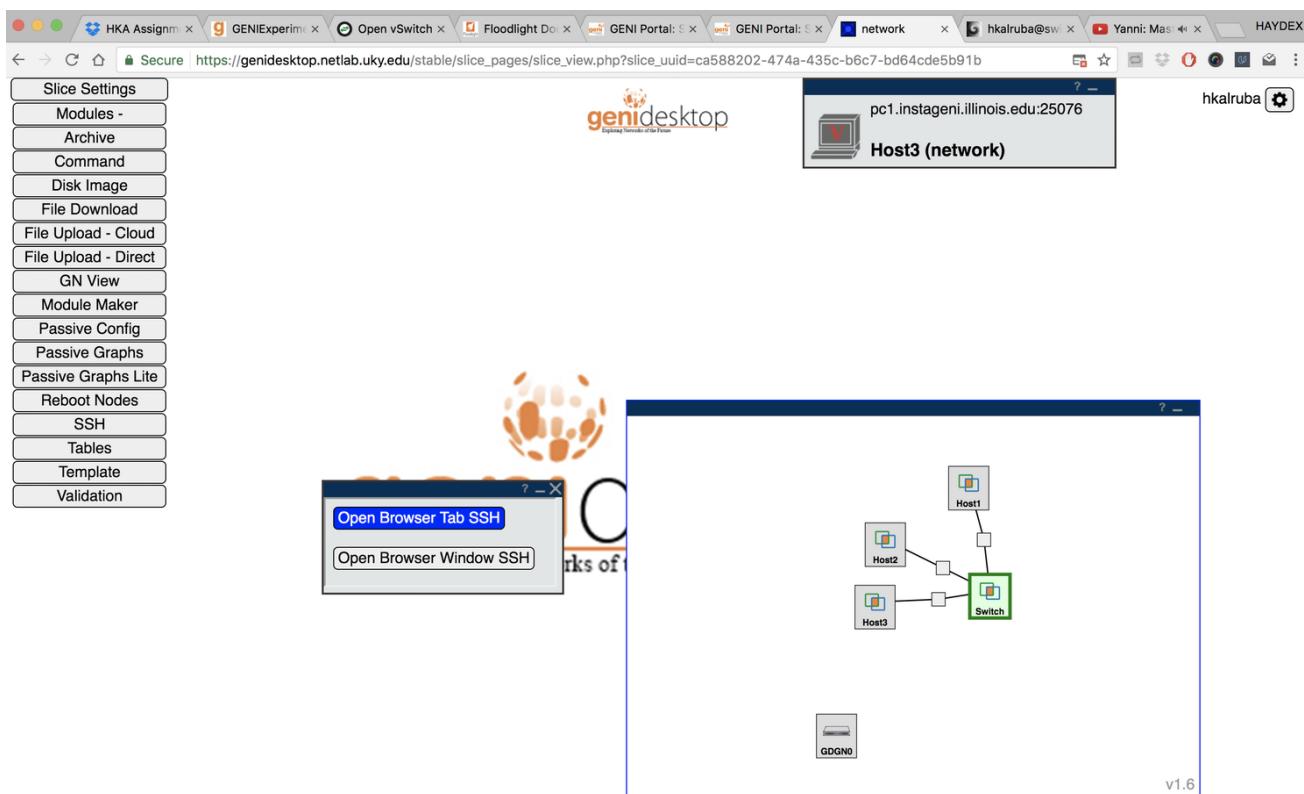


Figure 15

```
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May 5 23:46:08 2017 from genidesktop.netlab.uky.edu
hkalruba@switch:~$ sudo ovs-vsctl add-br br0
hkalruba@switch:~$
```

Figure 16 OVS host SSH connection in the browser

ii. We create an Ethernet bridge that will act as our software switch, Figure 17

```
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May 5 23:46:08 2017 from genidesktop.netlab.uky.edu
hkalruba@switch:~$ sudo ovs-vsctl add-br br0
hkalruba@switch:~$
```

Figure 17

iii. Prepare the interfaces to be added as ports to the OVS switch

The OVS bridge will be a Layer 2 switch and the ports will not need IP addresses. Before we remove them let's keep track of some information

- We Run ifconfig

```
hkalruba@switch:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 02:b9:39:f9:bd:00
          inet addr:172.17.1.33 Bcast:172.31.255.255 Mask:255.240.0.0
          inet6 addr: fe80::b9:39ff:fe9:bd09/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:92386 errors:0 dropped:0 overruns:0 frame:0
            TX packets:76813 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:125448463 (125.4 MB) TX bytes:8717049 (8.7 MB)

eth1      Link encap:Ethernet HWaddr 02:ab:11:b7:ee:ec
          inet addr:10.10.3.2 Bcast:10.10.3.255 Mask:255.255.255.0
          inet6 addr: fe80::ab:11ff:feb7:eeec/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:6 errors:0 dropped:0 overruns:0 frame:0
            TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:384 (384.0 B) TX bytes:1498 (1.4 KB)

eth2      Link encap:Ethernet HWaddr 02:5f:8c:27:37:e6
          inet addr:10.10.1.2 Bcast:10.10.1.255 Mask:255.255.255.0
          inet6 addr: fe80::5f:8cff:fe27:37e6/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:6 errors:0 dropped:0 overruns:0 frame:0
            TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:384 (384.0 B) TX bytes:1408 (1.4 KB)

eth3      Link encap:Ethernet HWaddr 02:73:17:46:ec:e7
          inet addr:10.10.2.2 Bcast:10.10.2.255 Mask:255.255.255.0
          inet6 addr: fe80::73:17ff:fe46:ec7/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:26 errors:0 dropped:0 overruns:0 frame:0
            TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1980 (1.9 KB) TX bytes:1498 (1.4 KB)

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:4 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:342 (342.0 B) TX bytes:342 (342.0 B)
```

Figure 18

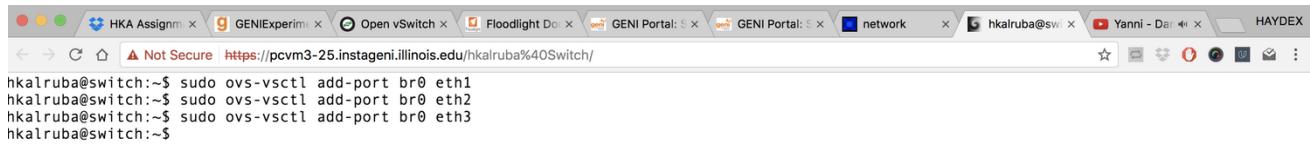
We note down the interfaces that connect to the three hosts we have and remove their IPs. These interfaces are eth1, eth2 and eth3. (Figures 18-19)



```
hkalruba@switch:~$ sudo ifconfig eth1 0
hkalruba@switch:~$ sudo ifconfig eth2 0
hkalruba@switch:~$ sudo ifconfig eth3 0
hkalruba@switch:~$
```

Figure 19

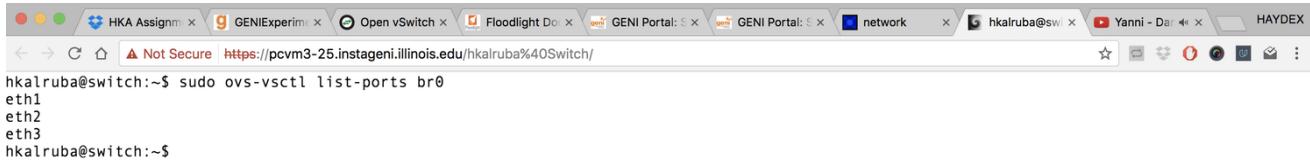
iv. Then we add all the data interfaces to the switch (bridge). (Figure 20)



```
hkalruba@switch:~$ sudo ovs-vsctl add-port br0 eth1
hkalruba@switch:~$ sudo ovs-vsctl add-port br0 eth2
hkalruba@switch:~$ sudo ovs-vsctl add-port br0 eth3
hkalruba@switch:~$
```

Figure 20

v. Now we verify if the three ports are configured correctly. (Figure 21)

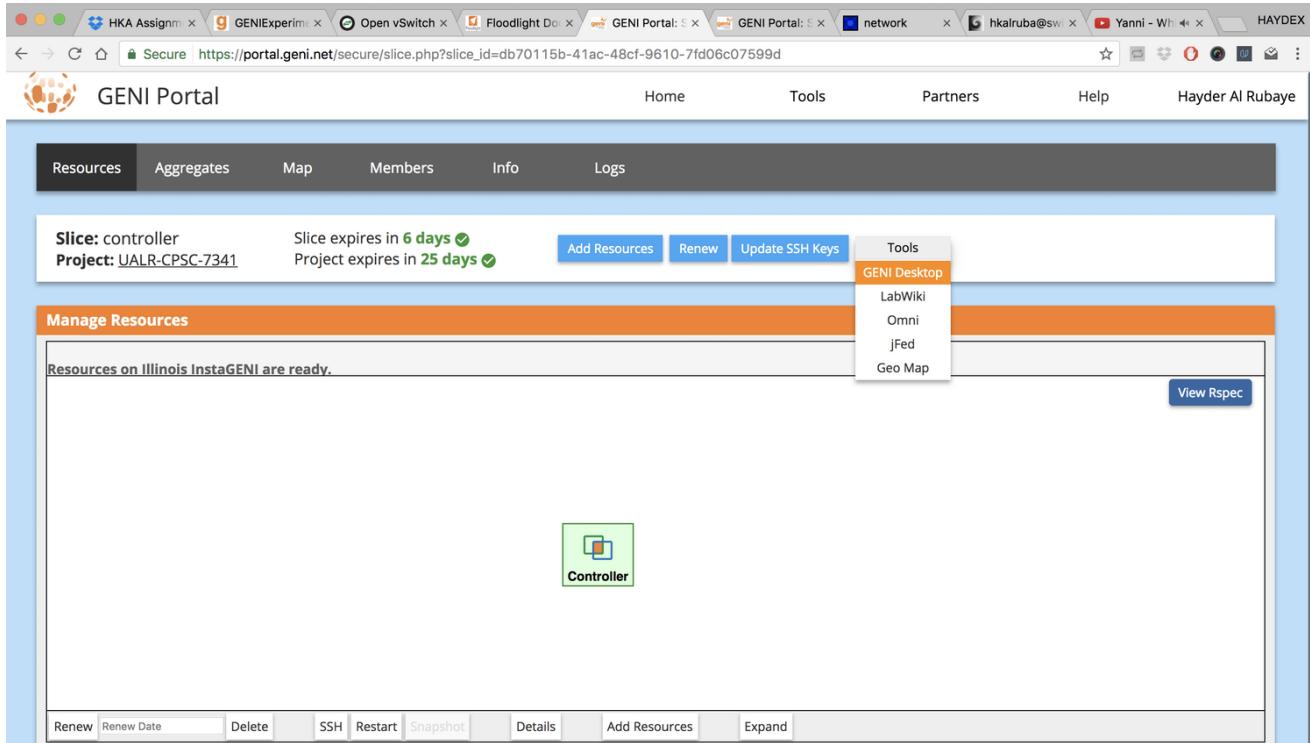


```
hkalruba@switch:~$ sudo ovs-vsctl list-ports br0
eth1
eth2
eth3
hkalruba@switch:~$
```

Figure 21

2b. Now we should point the switch to a controller

i. We login to the controller. (Figures 22-25)



The screenshot shows the GENI Portal interface. At the top, there's a navigation bar with links for Home, Tools, Partners, Help, and Hayder Al Rubaye. Below the navigation bar, there's a header with the text "Slice: controller" and "Project: UALR-CPSC-7341". It also displays "Slice expires in 6 days" and "Project expires in 25 days". There are buttons for "Add Resources", "Renew", and "Update SSH Keys". A dropdown menu under "Tools" shows options: "GENI Desktop" (which is selected and highlighted in orange), "LabWiki", "Omni", "jFed", and "Geo Map". In the main content area, there's a section titled "Manage Resources" with the message "Resources on Illinois InstaGENI are ready." Below this, there's a green square icon labeled "Controller". At the bottom, there are buttons for "Renew", "Renew Date", "Delete", "SSH", "Restart", "Snapshot", "Details", "Add Resources", and "Expand".

Figure 22



Figure 23



Figure 24



Figure 25

ii. We find the control interface IP of the controller, we use *ifconfig* and note down the IP address of eth0, which is 72.36.65.65 as seen in (Figure 26)

```
* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 00:50:20 2017 from genidesktop.netlab.uky.edu
hkalruba@controller:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 02:de:0f:bf:1f:35
          inet addr:72.36.65.65 Bcast:72.36.65.255 Mask:255.255.255.0
          inet6 addr: fe80::de:febf:1f35/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:66365 errors:0 dropped:0 overruns:0 frame:0
            TX packets:56064 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:122386707 (122.3 MB) TX bytes:5553810 (5.5 MB)

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:8 errors:0 dropped:0 overruns:0 frame:0
            TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:614 (614.0 B) TX bytes:614 (614.0 B)

hkalruba@controller:~$
```

Figure 26

iii. In order to point our software OpenFlow switch to the controller, in the *ovs switch* window, we run the following command (Figure 27)

```
sudo ovs-vsctl set-controller br0 tcp:<controller_ip>:6633
```

```
hkalruba@switch:~$ sudo ovs-vsctl set-controller br0 tcp:72.36.65.65:6633
hkalruba@switch:~$
```

Figure 27

iv. We set the switch to fail-safe-mode. (Figure 28)

```
hkalruba@switch:~$ sudo ovs-vsctl set-fail-mode br0 secure
hkalruba@switch:~$
```

Figure 28

v. We verify the OVS settings. (Figure 29)

```
hkalruba@switch:~$ sudo ovs-vsctl show
89362c3a-1d64-451b-977d-7b2718a1ba20
  Bridge "br0"
    Controller "tcp:72.36.65.65:6633"
    fail_mode: secure
    Port "eth1"
      Interface "eth1"
    Port "eth3"
      Interface "eth3"
    Port "br0"
      Interface "br0"
        type: internal
      Port "eth2"
        Interface "eth2"
    ovs_version: "2.3.1"
hkalruba@switch:~$
```

Figure 29

Step 4. Experiment Execution

4a. Login to the hosts 10.0.0.1, 10.0.0.2, 10.0.0.3 (Figures 30-33)

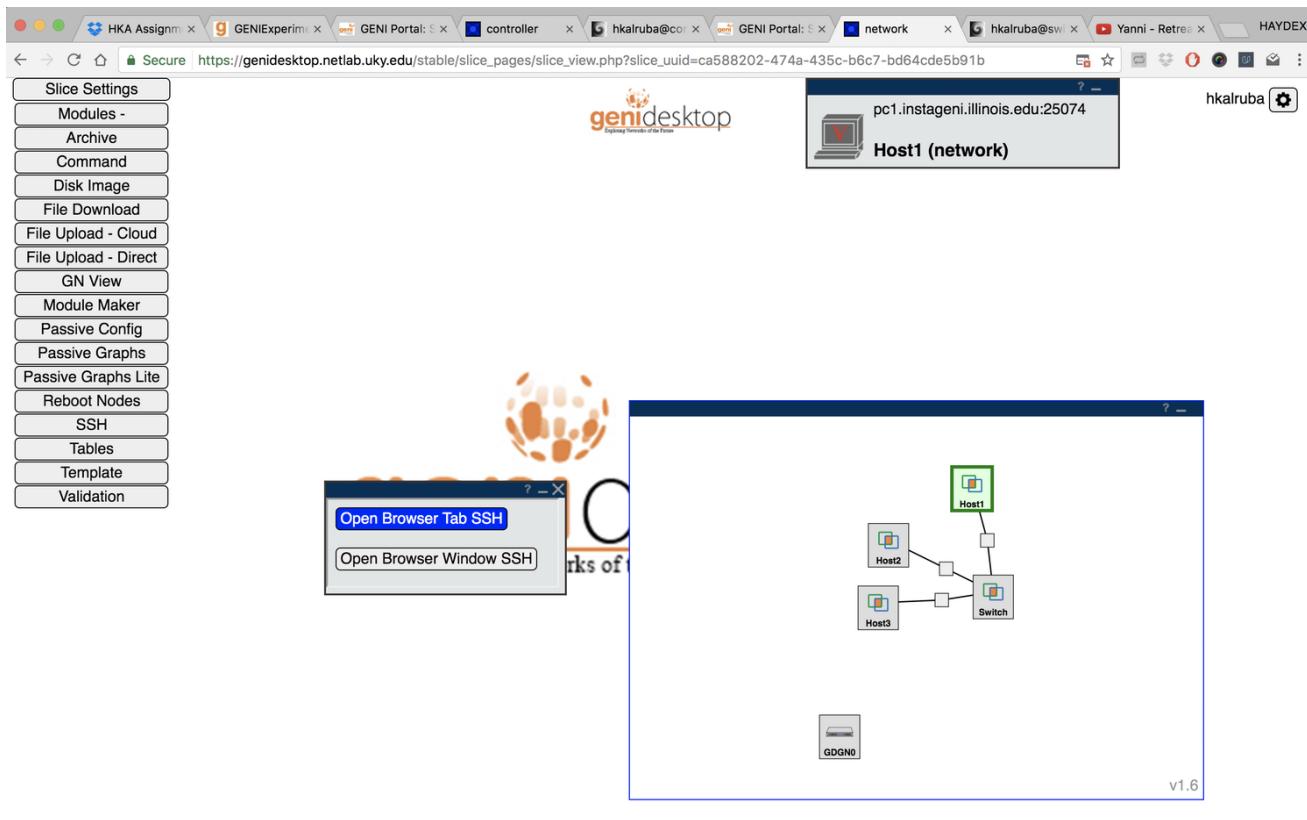


Figure 30

networkXp 6 days 19:22:10 Welcome JACKS Info SSH

Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

```
* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May  5 23:45:33 2017 from genidesktop.netlab.uky.edu
hkalruba@host1:~$
```

Figure 31 Logged to Host1

networkXp 6 days 19:22:10 Welcome JACKS Info SSH

Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

```
* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May  5 23:45:35 2017 from genidesktop.netlab.uky.edu
hkalruba@host2:~$
```

Figure 32 Logged to Host2

networkXp 6 days 19:22:10 Welcome JACKS Info SSH

Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

```
* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May  5 23:45:41 2017 from genidesktop.netlab.uky.edu
hkalruba@host3:~$
```

Figure 33 Logged to Host3

4b. We use a Learning Switch Controller

1. First we start a ping from host1(10.0.0.1) to host2(10.0.0.2), which should timeout, since there is no controller running. (Figure 34)

```

hkalruba@host1:~$ ping 10.0.0.2 -c 10
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable
From 10.0.0.1 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7039ms
pipe 3
hkalruba@host1:~$
```

Figure 34

2. Then we start the Floodlight Controller by running the following commands: (Figure 35)

- 1 cd /local/floodlight
- 2 java -jar target/floodlight.jar

```

hkalruba@controller:~$ cd /local/floodlight
hkalruba@controller:~/local/floodlight$ java -jar target/floodlight.jar
2017-05-06 00:25:21.885 INFO [n.f.c.m.FloodlightModuleLoader] Loading modules from src/main/resources/floodlightdefault.properties
2017-05-06 00:25:22.732 WARN [n.f.r.RestApiServer] HTTPS disabled; HTTPS will not be used to connect to the REST API.
2017-05-06 00:25:22.732 WARN [n.f.r.RestApiServer] HTTP enabled; Allowing unsecure access to REST API on port 8080.
2017-05-06 00:25:22.733 WARN [n.f.r.RestApiServer] CORS access control allow ALL origins: true
2017-05-06 00:25:22.639 WARN [n.f.c.i.OFSwitchManager] SSL disabled. Using unsecure connections between Floodlight and switches.
2017-05-06 00:25:23.640 INFO [n.f.c.i.OFSwitchManager] Clear switch flow tables on initial handshake as master: TRUE
2017-05-06 00:25:23.641 INFO [n.f.c.i.OFSwitchManager] Clear switch flow tables on each transition to master: TRUE
2017-05-06 00:25:23.668 INFO [n.f.c.i.OFSwitchManager] Setting 0x1 as the default max tables to receive table-miss flow
2017-05-06 00:25:23.801 INFO [n.f.c.i.OFSwitchManager] OpenFlow version OF_10 will be advertised to switches. Supported fallback version s [OF_10]
2017-05-06 00:25:23.805 INFO [n.f.c.i.OFSwitchManager] Listening for OpenFlow switches on [0.0.0.0]:6633
2017-05-06 00:25:23.806 INFO [n.f.c.i.OFSwitchManager] OpenFlow socket config: 1 boss thread(s), 16 worker thread(s), 60000 ms TCP connection timeout, max 1600 connection backlog, 4194304 byte TCP send buffer size
2017-05-06 00:25:23.811 INFO [n.f.c.i.Controller] ControllerId set to 1
2017-05-06 00:25:23.812 INFO [n.f.c.i.Controller] Shutdown when controller transitions to STANDBY HA role: true
2017-05-06 00:25:23.812 WARN [n.f.c.i.Controller] Controller will automatically deserialize all Ethernet packet-in messages. Set 'deserializeEighthPacketIns' to 'FALSE' if this feature is not required or when benchmarking core performance
2017-05-06 00:25:23.814 INFO [n.f.c.i.Controller] Controller role set to ACTIVE
2017-05-06 00:25:24.85 INFO [n.f.l.i.LinkDiscoveryManager] Link latency history set to 10 LLDP data points
2017-05-06 00:25:24.91 INFO [n.f.l.i.LinkDiscoveryManager] Latency update threshold set to +/-0.5 (50.0%) of rolling historical average
2017-05-06 00:25:24.94 INFO [n.f.t.TopologyManager] Path metrics set to LATENCY
2017-05-06 00:25:24.94 INFO [n.f.t.TopologyManager] Will compute a max of 3 paths upon topology updates
2017-05-06 00:25:24.126 INFO [n.f.f.Forwarding] Default hard timeout not configured. Using 0.
2017-05-06 00:25:24.127 INFO [n.f.f.Forwarding] Default idle timeout set to 5.
2017-05-06 00:25:24.128 INFO [n.f.f.Forwarding] Default table ID not configured. Using 0x0.
2017-05-06 00:25:24.128 INFO [n.f.f.Forwarding] Default priority not configured. Using 1.
2017-05-06 00:25:24.129 INFO [n.f.f.Forwarding] Default flags will be set to SEND_FLOW_REM false.
2017-05-06 00:25:24.129 INFO [n.f.f.Forwarding] Default flow matches set to: IN_PORT=true, VLAN=true, MAC=true, IP=true, FLAG=true, TPPT=true
2017-05-06 00:25:24.130 INFO [n.f.f.Forwarding] Default detailed flow matches set to: SRC_MAC=true, DST_MAC=true, SRC_IP=true, DST_IP=true, SRC_TPPT=true, DST_TPPT=true
2017-05-06 00:25:24.130 INFO [n.f.f.Forwarding] Not flooding ARP packets. ARP flows will be inserted for known destinations
2017-05-06 00:25:24.131 INFO [n.f.f.Forwarding] Flows will be removed on link/port down events
2017-05-06 00:25:24.132 INFO [n.f.s.StatisticsCollector] Statistics collection disabled
2017-05-06 00:25:24.132 INFO [n.f.s.StatisticsCollector] Port statistics collection interval set to 10s
2017-05-06 00:25:24.430 INFO [o.s.s.i.SyncManager] [1] Updating sync configuration ClusterConfig [allNodes={1=Node [hostname=192.168.1.100, port=6642, nodeId=1, domainId=1], 2=Node [hostname=192.168.1.100, port=6643, nodeId=2, domainId=1]}, authScheme=CHALLENGE_RESPONSE, keyStorePath=/etc/floodlight/key2.jceks, keyStorePassword is set]
2017-05-06 00:25:25.39 INFO [o.s.s.i.r.RPCService] Listening for internal floodlight RPC on 0.0.0.0/0.0.0.0:6642
2017-05-06 00:25:25.890 INFO [o.r.C.I.Server] Starting the Simple [HTTP/1.1] server on port 8080

```

Figure 35

HKA Assignment 5

```

2017-05-06 00:25:24.129 INFO [n.f.f.Forwarding] Default flags will be set to SEND_FLOW_REM false.
2017-05-06 00:25:24.129 INFO [n.f.f.Forwarding] Default flow matches set to: IN_PORT=true, VLAN=true, MAC=true, IP=true, FLAG=true, TPPT=true
2017-05-06 00:25:24.130 INFO [n.f.f.Forwarding] Default detailed flow matches set to: SRC_MAC=true, DST_MAC=true, SRC_IP=true, DST_IP=true, SRC_TPPT=true, DST_TPPT=true
2017-05-06 00:25:24.130 INFO [n.f.f.Forwarding] Not flooding ARP packets. ARP flows will be inserted for known destinations
2017-05-06 00:25:24.131 INFO [n.f.f.Forwarding] Flows will be removed on link/port down events
2017-05-06 00:25:24.132 INFO [n.f.s.StatisticsCollector] Statistics collection disabled
2017-05-06 00:25:24.132 INFO [n.f.s.StatisticsCollector] Port statistics collection interval set to 10s
2017-05-06 00:25:24.430 INFO [o.s.s.i.SyncManager] [1] Updating sync configuration ClusterConfig [allNodes={1=Node [hostname=192.168.1.100, port=6642, nodeId=1, domainId=1], 2=Node [hostname=192.168.1.100, port=6643, nodeId=2, domainId=1]}, authScheme=CHALLENGE_RESPONSE, keyStorePath=/etc/floodlight/key2.jceks, keyStorePassword is set]
2017-05-06 00:25:25.39 INFO [o.s.s.i.r.RPCService] Listening for internal floodlight RPC on 0.0.0.0/0.0.0.0:6642
2017-05-06 00:25:25.890 INFO [o.r.C.I.Server] Starting the Simple [HTTP/1.1] server on port 8080
2017-05-06 00:25:25.892 INFO [org.restlet] Starting net.floodlightcontroller.restserver.RestApiServer$RestApplication application
2017-05-06 00:25:27.910 INFO [n.f.c.i.OFChannelHandler] New switch connection from /172.17.1.33:33723
2017-05-06 00:25:28.1 INFO [n.f.c.i.OFChannelHandler] Negotiated down to controller OpenFlow version of OF_10 for /172.17.1.33:33723 using lesser hello header algorithm.
2017-05-06 00:25:28.435 INFO [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:06:14:8d:84:32:4b] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescription [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.3.1, serialNumber=None, datapathDescription=None]
2017-05-06 00:25:28.524 INFO [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:06:14:8d:84:32:4b on upcoming transition to MASTER.
2017-05-06 00:25:29.142 INFO [n.f.t.TopologyManager] Recomputing topology due to: link-discovery-updates
2017-05-06 00:25:36.576 INFO [n.f.j.JythonServer] Starting DebugServer on :6655
2017-05-06 00:25:40.129 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:25:55.141 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:26:10.146 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:26:25.154 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:26:40.167 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:26:55.177 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:27:10.185 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:27:25.197 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:27:40.206 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:27:55.211 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports

```

Figure 35 Continued

3. In the terminal window of host1, we re-run the ping to host2 i.e. 10.0.0.2 and host3 10.0.0.3 (Figures 36-37)

```

hkalruba@host1:~$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=209 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.92 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=1.72 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.47 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=1.54 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=1.22 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=1.25 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=1.48 ms
^C
--- 10.0.0.2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8011ms
rtt min/avg/max/mdev = 1.037/24.577/209.518/65.387 ms
hkalruba@host1:~$ 

```

Figure 36

```

hkalruba@host1:~$ ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=19.4 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=1.83 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=1.13 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=1.21 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=1.56 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=1.67 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=1.24 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=1.50 ms
^C
--- 10.0.0.3 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7011ms
rtt min/avg/max/mdev = 1.133/3.706/19.491/5.970 ms
hkalruba@host1:~$ 

```

Figure 37

4. We now go to the Open vSwitch host and take a look at the flows. We should see that the controller installed flows based on the mac addresses of the packets. (Figure 38)

```

hkalruba@switch:~$ sudo ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
hkalruba@switch:~$ sudo ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
  cookie=0x20000010000000, duration=6.700s, table=0, n_packets=6, n_bytes=588, idle_timeout=5, priority=1, ip,in_port=2,dl_src=02:65:d7:e9:5a:f5,dl_dst=02:c3:33:08:c0:94,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:3
  cookie=0x20000011000000, duration=6.697s, table=0, n_packets=6, n_bytes=588, idle_timeout=5, priority=1, ip,in_port=3,dl_src=02:c3:33:08:c0:94,dl_dst=02:65:d7:e9:5a:f5,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=output:2
  cookie=0x20000012000000, duration=1.688s, table=0, n_packets=0, n_bytes=0, idle_timeout=5, priority=1, arp,in_port=3,dl_src=02:c3:33:08:c0:94,dl_dst=02:65:d7:e9:5a:f5 actions=output:2
  cookie=0x20000013000000, duration=1.685s, table=0, n_packets=0, n_bytes=0, idle_timeout=5, priority=1, arp,in_port=2,dl_src=02:65:d7:e9:5a:f5,dl_dst=02:c3:33:08:c0:94 actions=output:3
hkalruba@switch:~$ sudo ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
  cookie=0x20000014000000, duration=5.331s, table=0, n_packets=5, n_bytes=490, idle_timeout=5, priority=1, ip,in_port=2,dl_src=02:65:d7:e9:5a:f5,dl_dst=02:93:9a:4e:77:89,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:1
  cookie=0x20000015000000, duration=5.321s, table=0, n_packets=5, n_bytes=490, idle_timeout=5, priority=1, ip,in_port=1,dl_src=02:93:9a:4e:77:89,dl_dst=02:65:d7:e9:5a:f5,nw_src=10.0.0.3,nw_dst=10.0.0.1 actions=output:2
  cookie=0x20000017000000, duration=0.312s, table=0, n_packets=0, n_bytes=0, idle_timeout=5, priority=1, arp,in_port=2,dl_src=02:65:d7:e9:5a:f5,dl_dst=02:93:9a:4e:77:89 actions=output:1
  cookie=0x20000016000000, duration=0.319s, table=0, n_packets=0, n_bytes=0, idle_timeout=5, priority=1, arp,in_port=1,dl_src=02:93:9a:4e:77:89,dl_dst=02:65:d7:e9:5a:f5 actions=output:2
hkalruba@switch:~$
```

Figure 38

4c. We take a look around the OVS switch

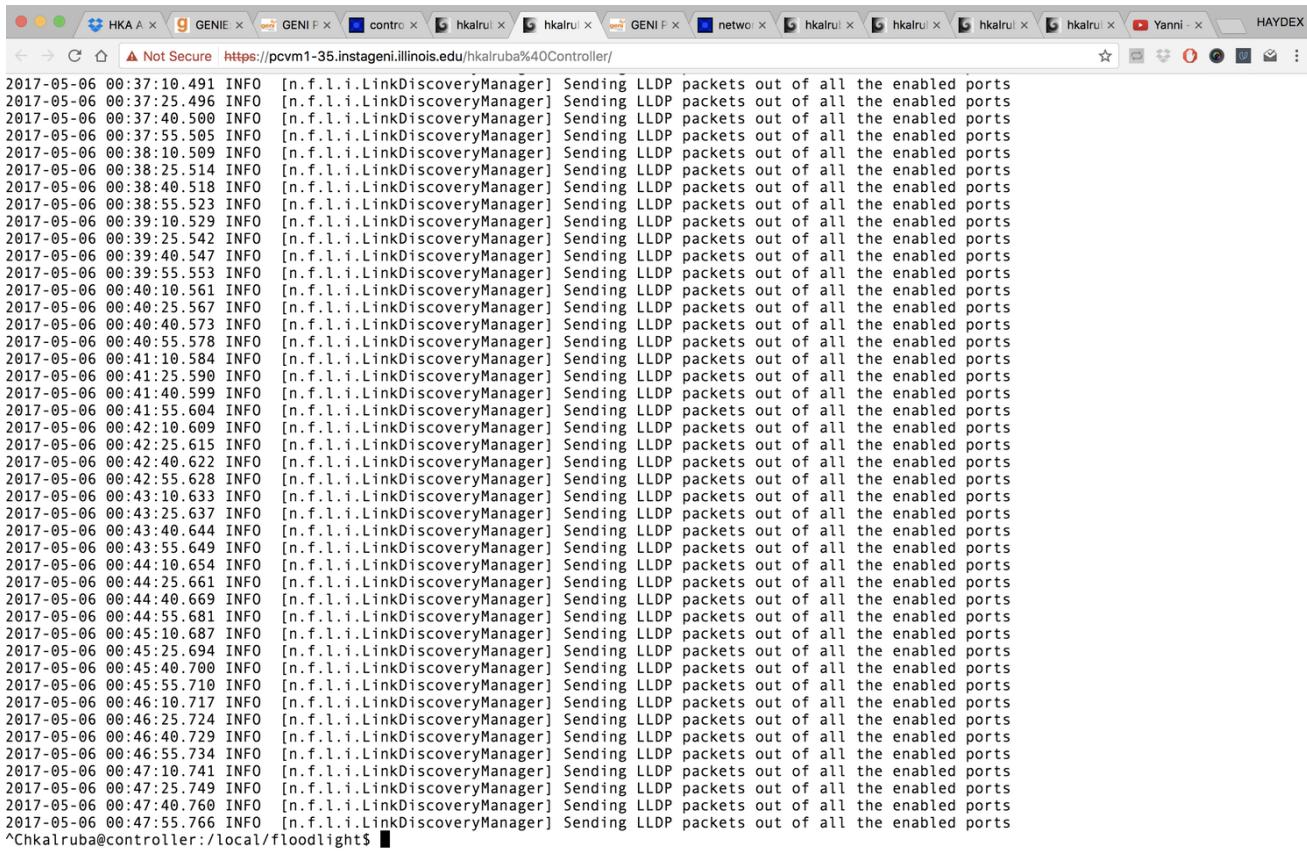
- To see messages go between the switch and the controller, we open a new ssh window to the controller node and run tcpdump on the eth1 interface and on the tcp port that your controller is listening on usually 6633. (Figure 39)

```

hkalruba@controller:~$ sudo tcpdump -i eth0 tcp port 6633
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
01:45:09.695717 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 3237538758:3237538766, a
ck 2942769275, win 252, options [nop,nop,TS val 3984221 ecr 4130223], length 8: OpenFlow
    version 1.0, type ECHO_REQUEST, length 8, xid 0xffffffffdc
01:45:09.696546 IP pcvm1-33.instageni.illinois.edu.33723 > pcvm1-34.instageni.illinois.edu.6633: Flags [P.], seq 1:9, ack 8, win 229, opt
ions [nop,nop,TS val 4130723 ecr 3984221], length 8: OpenFlow
    version 1.0, type ECHO_REPLY, length 8, xid 0xffffffffdc
01:45:09.696577 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 2327538758:3237538766, a
ck 2942769275, win 252, options [nop,nop,TS val 3984221 ecr 4130223], length 8: OpenFlow
    version 1.0, type ECHO_REQUEST, length 8, xid 0xffffffffdc
01:45:10.690712 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 8:107, ack 9, win 252, o
ptions [nop,nop,TS val 3984470 ecr 4130723], length 0
01:45:10.691032 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 206:305, ack 9, win 252,
options [nop,nop,TS val 3984470 ecr 4130723], length 99: OpenFlow
    version 1.0, type PACKET_OUT, length 99, xid 0x00000043a
01:45:10.690893 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 107:206, ack 9, win 252,
options [nop,nop,TS val 3984470 ecr 4130723], length 99: OpenFlow
    version 1.0, type PACKET_OUT, length 99, xid 0x00000043b
01:45:10.691158 IP pcvm1-33.instageni.illinois.edu.33723 > pcvm1-34.instageni.illinois.edu.6633: Flags [P.], ack 206, win 229, options [no
p,nop,TS val 4130972 ecr 3984470], length 0
01:45:10.719340 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 305:412, ack 9, win 252,
options [nop,nop,TS val 3984477 ecr 4130972], length 107: OpenFlow
    version 1.0, type PACKET_OUT, length 107, xid 0x00000043d
01:45:10.719510 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 412:519, ack 9, win 252,
options [nop,nop,TS val 3984477 ecr 4130972], length 107: OpenFlow
    version 1.0, type PACKET_OUT, length 107, xid 0x00000043e
01:45:10.719657 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 519:626, ack 9, win 252,
options [nop,nop,TS val 3984477 ecr 4130972], length 107: OpenFlow
    version 1.0, type PACKET_OUT, length 107, xid 0x00000043f
01:45:10.719734 IP pcvm1-33.instageni.illinois.edu.33723 > pcvm1-34.instageni.illinois.edu.6633: Flags [P.], ack 412, win 229, options [no
p,nop,TS val 4130972 ecr 3984470], length 0
01:45:10.720107 IP pcvm1-33.instageni.illinois.edu.33723 > pcvm1-34.instageni.illinois.edu.6633: Flags [P.], ack 626, win 229, options [no
p,nop,TS val 4130979 ecr 3984477], length 0
01:45:12.721261 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 626:634, ack 9, win 252,
options [nop,nop,TS val 3984477 ecr 4130979], length 8: OpenFlow
    version 1.0, type ECHO_REQUEST, length 8, xid 0xfffffffffdb
01:45:12.722245 IP pcvm1-33.instageni.illinois.edu.33723 > pcvm1-34.instageni.illinois.edu.6633: Flags [P.], seq 9:17, ack 634, win 229,
options [nop,nop,TS val 4131480 ecr 3984477], length 8: OpenFlow
    version 1.0, type ECHO_REPLY, length 8, xid 0xfffffffffdb
01:45:12.722276 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], ack 17, win 252, options [nop
,nop,TS val 39844978 ecr 4131480], length 0
01:45:14.722135 IP pcvm1-34.instageni.illinois.edu.6633 > pcvm1-33.instageni.illinois.edu.33723: Flags [P.], seq 634:642, ack 17, win 252
, options [nop,nop,TS val 3985478 ecr 4131480], length 8: OpenFlow
    version 1.0, type ECHO_REQUEST, length 8, xid 0xfffffffffdb
```

Figure 39

- we now Kill the Floodlight controller by pressing Ctrl-C. (Figure 40)



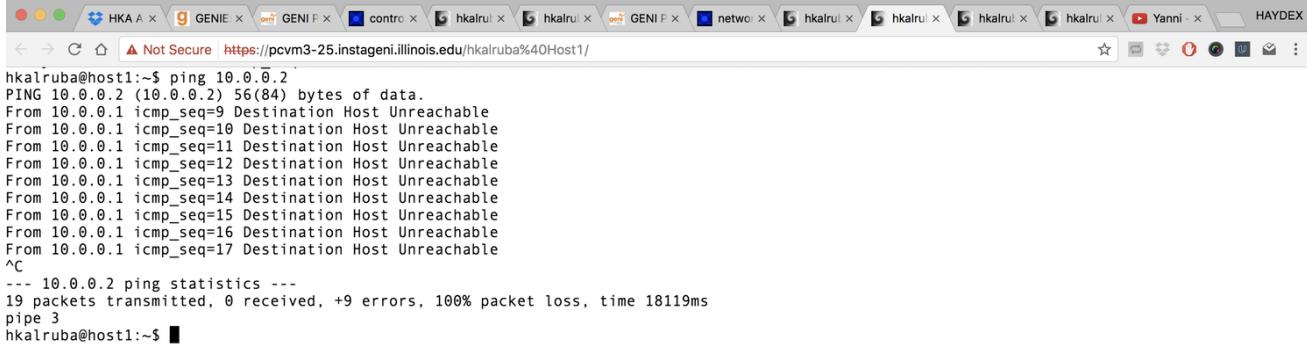
```

2017-05-06 00:37:10.491 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:37:25.496 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:37:40.506 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:37:55.505 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:38:10.509 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:38:25.514 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:38:40.518 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:38:55.523 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:39:10.529 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:39:25.542 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:39:40.547 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:39:55.553 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:40:10.561 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:40:25.567 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:40:40.573 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:40:55.578 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:41:10.584 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:41:25.590 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:41:40.599 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:41:55.604 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:42:10.609 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:42:25.615 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:42:40.622 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:42:55.628 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:43:10.633 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:43:25.637 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:43:40.644 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:43:55.649 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:44:10.654 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:44:25.661 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:44:40.669 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:44:55.681 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:45:10.687 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:45:25.694 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:45:40.700 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:45:55.710 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:46:10.717 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:46:25.724 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:46:40.729 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:46:55.734 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:47:10.741 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:47:25.749 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:47:40.760 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
2017-05-06 00:47:55.766 INFO [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets out of all the enabled ports
^Chkalruba@controller:/local/floodlight$ 

```

Figure 40

4. The ping on host1 has stopped and Host2 and Host3 became unreachable. (Figure 41)



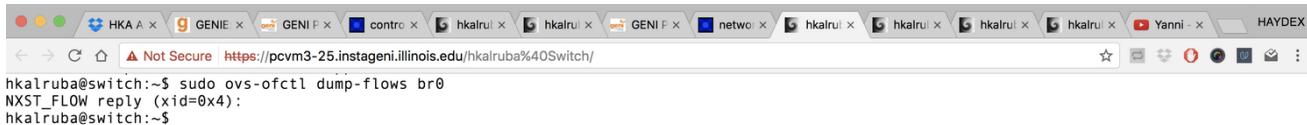
```

hkalruba@host1:~$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=9 Destination Host Unreachable
From 10.0.0.1 icmp_seq=10 Destination Host Unreachable
From 10.0.0.1 icmp_seq=11 Destination Host Unreachable
From 10.0.0.1 icmp_seq=12 Destination Host Unreachable
From 10.0.0.1 icmp_seq=13 Destination Host Unreachable
From 10.0.0.1 icmp_seq=14 Destination Host Unreachable
From 10.0.0.1 icmp_seq=15 Destination Host Unreachable
From 10.0.0.1 icmp_seq=16 Destination Host Unreachable
From 10.0.0.1 icmp_seq=17 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
19 packets transmitted, 0 received, +9 errors, 100% packet loss, time 18119ms
pipe 3
hkalruba@host1:~$ 

```

Figure 41

5. We check the flow table entries on the OVS switch, they are empty. (Figure 42)



```

hkalruba@switch:~$ sudo ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
hkalruba@switch:~$ 

```

Figure 42

Since we set the switch to "secure" mode, i.e. don't forward packets if the controller fails, we will not see flow table entries. (Figure 42)

4d. Debugging the Controller

i. Print messages

ii. Checking the status in the switch (Figure 43)

```

hkalruba@switch:~$ sudo ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
hkalruba@switch:~$ sudo ovs-vsctl show
89362c3a-1d64-451b-977d-7b2718a1ba20
    Bridge "br0"
        Controller "tcp:72.36.65.65:6633"
        fail_mode: secure
        Port "eth1"
            Interface "eth1"
        Port "eth3"
            Interface "eth3"
        Port "br0"
            Interface "br0"
                type: internal
        Port "eth2"
            Interface "eth2"
        ovs_version: "2.3.1"
hkalruba@switch:~$ sudo ovs-vsctl show br0
ovs-vsctl: 'show' command takes at most 0 arguments
hkalruba@switch:~$ sudo ovs-ofctl show br0
OFPT_FEATURES_REPLY (xid=0x2): dpid:000006148d84324b
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
1(eth1): addr:02:ab:11:b7:ee:ec
    config: 0
    state: 0
    speed: 0 Mbps now, 0 Mbps max
2(eth2): addr:02:5f:8c:27:37:e6
    config: 0
    state: 0
    speed: 0 Mbps now, 0 Mbps max
3(eth3): addr:02:73:17:46:ec:e7
    config: 0
    state: 0
    speed: 0 Mbps now, 0 Mbps max
LOCAL(br0): addr:06:14:8d:84:32:4b
    config: PORT_DOWN
    state: LINK_DOWN
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
hkalruba@switch:~$ █

```

Figure 43

iii. Using Wireshark to see the OpenFlow messages

Using the OpenFlow dissector for Wireshark on the reserved controller host. (Figures 44-46)

We first install the XQUARTZ server (Figure 43.1) to get the wireshark to work on the localhost.

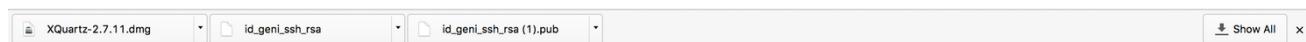
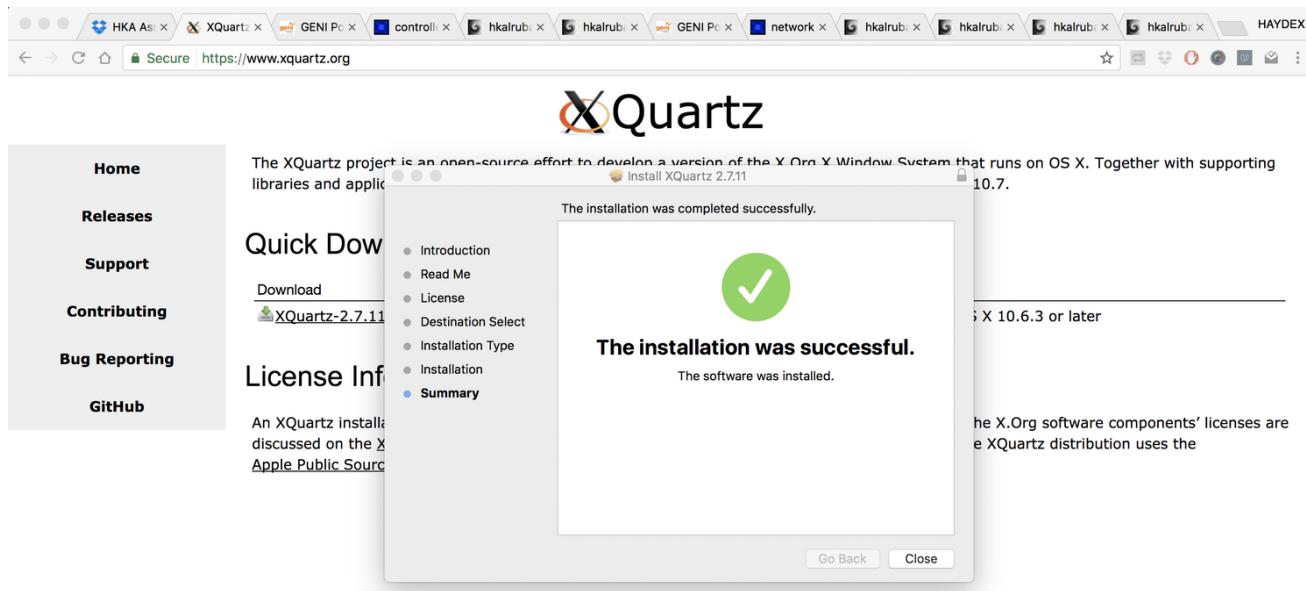


Figure 43.1

To use Wireshark we would use mac's Terminal this time to connect to the controller and the OVS. (Figure 44-46)

Name	Description	Public Key	Private Key	PuTTY	Edit	Delete
id_geni_ssh_rsa a9:27:a5:07:46:6a:71:c1:16:5e:6c:15:ad:52:44:93	Generated SSH keypair	Download Public Key	Download Private Key	Download PuTTY Key	Edit	Delete

On Linux and Mac systems and for most Windows SSH clients (not PuTTY), do:

- Download your private key.
- On Windows, just point your SSH client (not PuTTY) to the downloaded private key.
- On Linux and Mac, open a terminal.
 - Store your key under `~/.ssh/`:
 - If the directory does not exist, create it:

```
mkdir ~/.ssh
```
 - Move the key to `~/.ssh/`:


```
mv ~/Downloads/id_geni_ssh_rsa ~/.ssh/
```
 - Change the file permissions:


```
chmod 0600 ~/.ssh/id_geni_ssh_rsa
```
 - Your SSH command will be something like:

```
ssh -i ~/.ssh/id_geni_ssh_rsa [username]@[hostname] -p [port]
```

Figure 44 We download the public and private keys



```
haydex — -bash — 80x24
Last login: Sat May  6 02:30:38 on ttys000
[HAYDEXs-MBP:~ haydex$ mv ~/Downloads/genikeys/id_geni_ssh_rsa ~/.ssh/
[HAYDEXs-MBP:~ haydex$ chmod 0600 ~/.ssh/id_geni_ssh_rsa
[HAYDEXs-MBP:~ haydex$ ssh-add ~/.ssh/id_geni_ssh_rsa
[Enter passphrase for /Users/haydex/.ssh/id_geni_ssh_rsa:
Identity added: /Users/haydex/.ssh/id_geni_ssh_rsa (/Users/haydex/.ssh/id_geni_s
sh_rsa)
```

Figure 45 Setting up the SSH Keys on my mac computer in Terminal

We connect to the controller through the ssh -Y <username>@<controller> in the terminal as in (Figure 46)

```
[HAYDEXs-MBP:~ haydex$ ssh -Y hkalruba@pcvm1-34.instageni.illinois.edu
The authenticity of host 'pcvm1-34.instageni.illinois.edu (72.36.65.65)' can't b
e established.
RSA key fingerprint is SHA256:NNzk+bYMN/+zrRoUpPKZmDUXmXcylDLfmL5FYlCaF/s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pcvm1-34.instageni.illinois.edu,72.36.65.65' (RSA) t
o the list of known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 01:56:35 2017 from pcvm1-35.instageni.illinois.edu
hkalruba@controller:~$ █
```

Figure 46 Now I am connected to the controller through the Terminal

Assuming that the public IP address on the controller is on eth0, we run Wireshark by typing: (Figure 47)

```
sudo wireshark -i eth0&
```

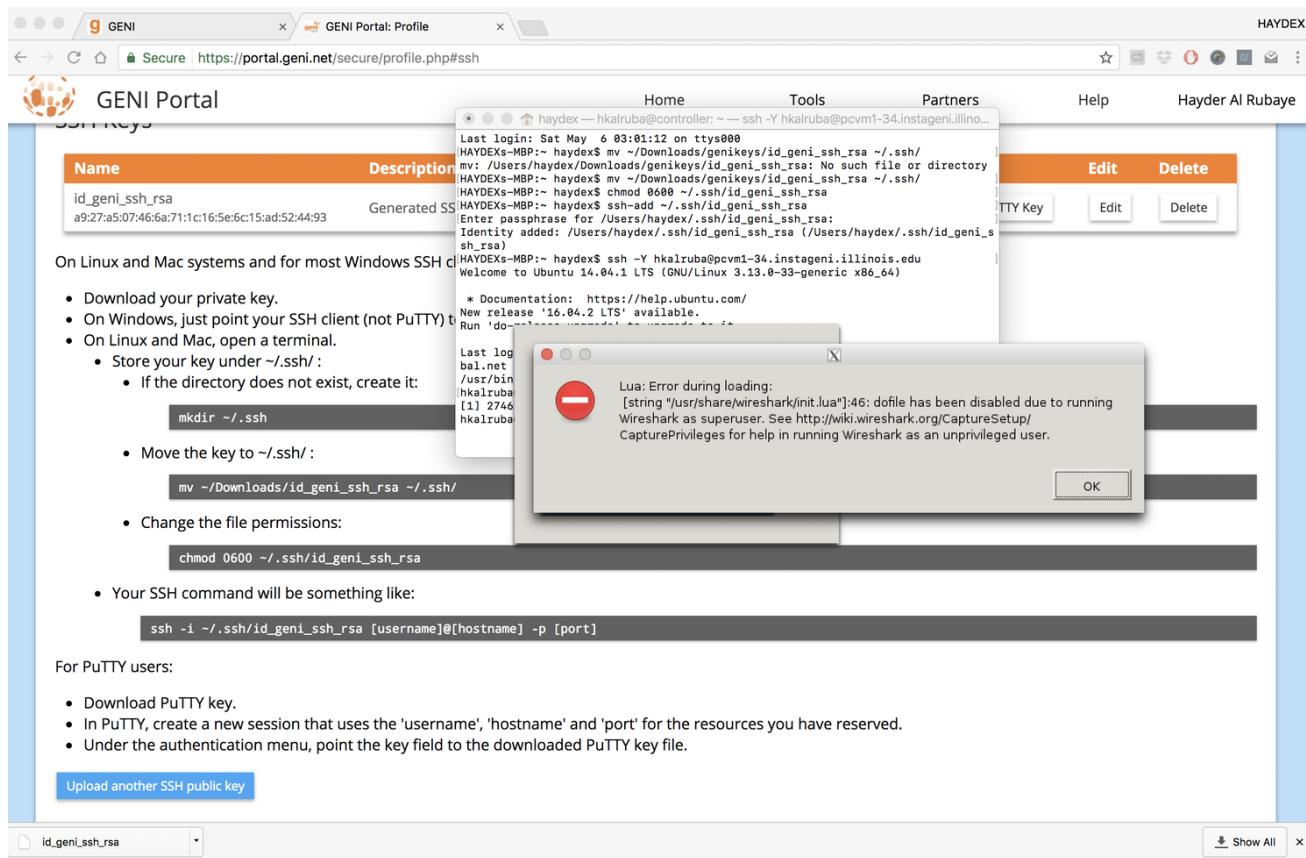


Figure 47 Wireshark in controller

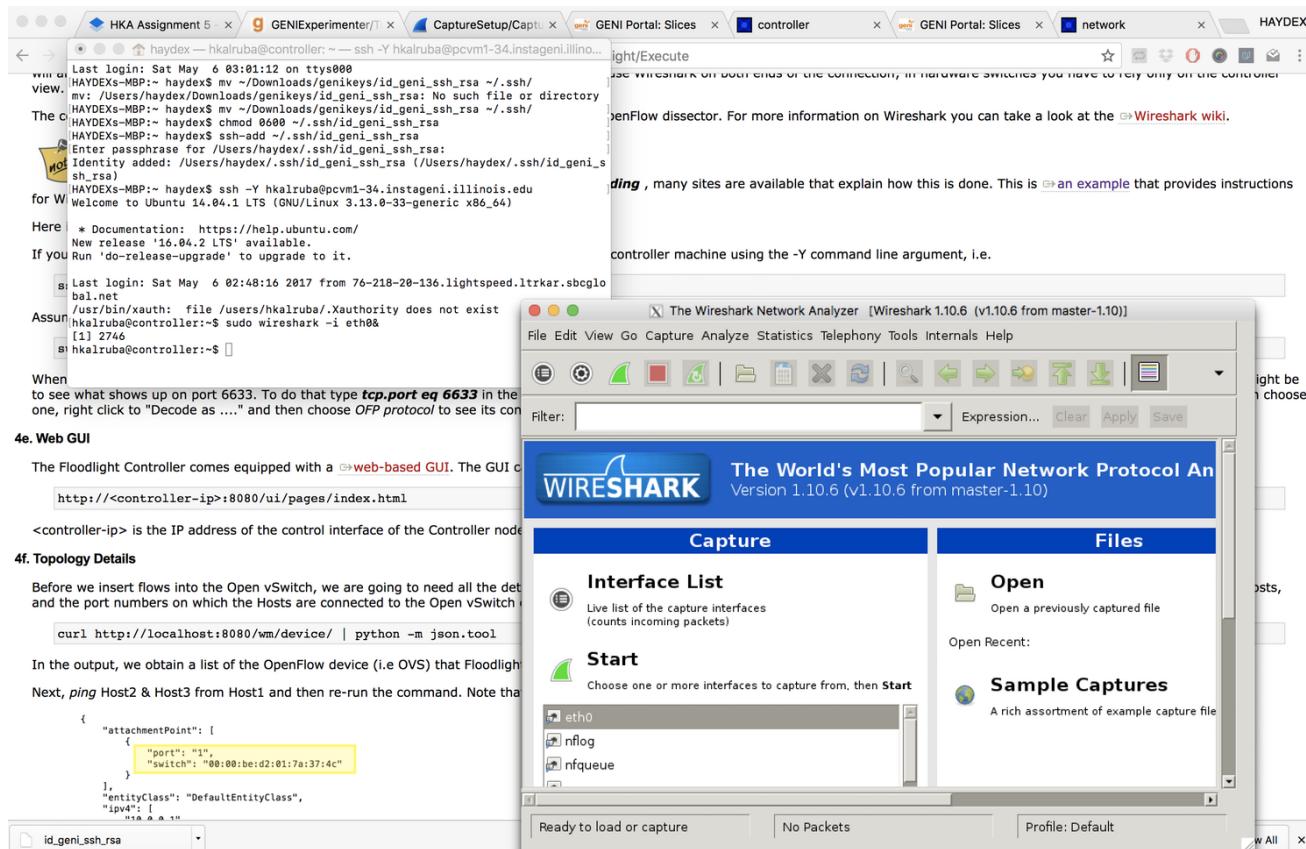
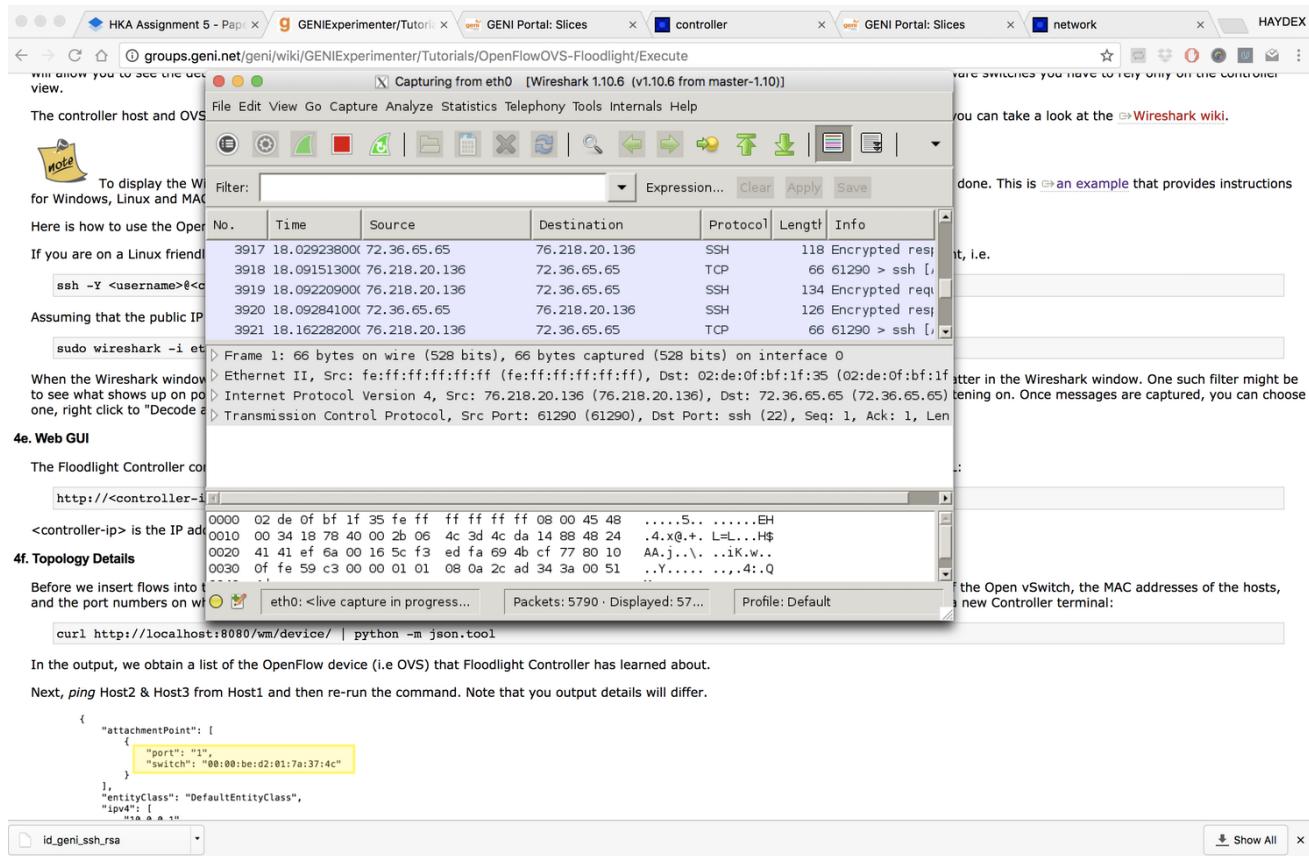


Figure 48



When the Wireshark window pops up, we might still have to choose eth0 for a live capture. (Figure 49). And we will want to use a filter to cut down on the chatter in the Wireshark window. One such filter might be to see what shows up on port 6633. To do that we type **tcp.port eq 6633** in the filter window, assuming that 6633 is the port that the controller is listening on. Once messages are captured, we can choose one, right click to "Decode as" and then we choose *OFP protocol* to see its content.

HKA Assignment 5

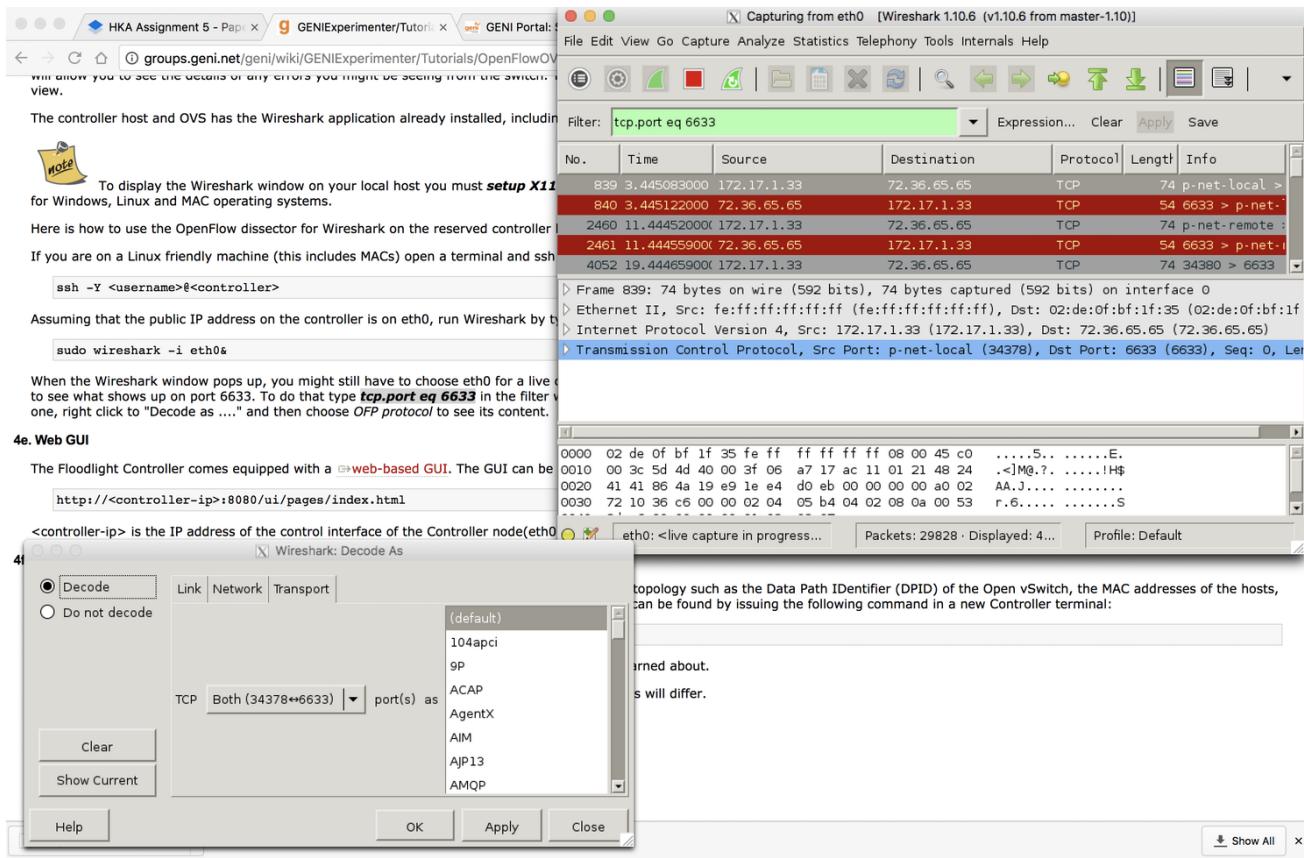


Figure 50 Wireshark applied filter `tcp.port eq 6633` and right clicked on one of the packets and chose `Decode As`

4e. Web GUI

The Floodlight Controller comes equipped with a [web-based GUI](#).

First we start the floodlight controller (Figure 50.1)

```
hkalruba@controller:~$ cd /local/floodlight
hkalruba@controller:/local/floodlight$ dir
apps      debian  findbugs-exclude.xml  floodlight_style_settings.xml  LICENSE.txt  pom.xml    setup-eclipse.sh  target
build.xml  example  floodlight.sh      lib                      NOTICE.txt   README.md   src
hkalruba@controller:/local/floodlight$ java -jar floodlight.jar
Error: Unable to access jarfile floodlight.jar
hkalruba@controller:/local/floodlight$ java -jar target/floodlight.jar
2017-05-06 11:24:24.652 INFO [n.f.c.m.FloodlightModuleLoader] Loading modules from src/main/resources/floodlightdefault.properties
2017-05-06 11:24:25.285 WARN [n.f.r.RestApiServer] HTTPS disabled; HTTPS will not be used to connect to the REST API.
2017-05-06 11:24:25.286 WARN [n.f.r.RestApiServer] HTTP enabled; Allowing unsecure access to REST API on port 8080.
2017-05-06 11:24:25.287 WARN [n.f.r.RestApiServer] CORS access control allow ALL origins: true
2017-05-06 11:24:25.798 WARN [n.f.c.i.OFSwitchManager] SSL disabled. Using unsecure connections between Floodlight and switches.
2017-05-06 11:24:25.799 INFO [n.f.c.i.OFSwitchManager] Clear switch flow tables on initial handshake as master: TRUE
2017-05-06 11:24:25.800 INFO [n.f.c.i.OFSwitchManager] Clear switch flow tables on each transition to master: TRUE
2017-05-06 11:24:25.813 INFO [n.f.c.i.OFSwitchManager] Setting 0x1 as the default max tables to receive table-miss flow
2017-05-06 11:24:25.881 INFO [n.f.c.i.OFSwitchManager] OpenFlow version OF_10 will be advertised to switches. Supported fallback version
s [OF_10]
2017-05-06 11:24:25.883 INFO [n.f.c.i.OFSwitchManager] Listening for OpenFlow switches on [0.0.0.0]:6633
2017-05-06 11:24:25.883 INFO [n.f.c.i.OFSwitchManager] OpenFlow socket config: 1 boss thread(s), 16 worker thread(s), 60000 ms TCP conne
ction timeout, max 1000 connection backlog, 4194304 byte TCP send buffer size
2017-05-06 11:24:25.885 INFO [n.f.c.i.Controller] ControllerId set to 1
2017-05-06 11:24:25.886 INFO [n.f.c.i.Controller] Shutdown when controller transitions to STANDBY HA role: true
2017-05-06 11:24:25.886 WARN [n.f.c.i.Controller] Controller will automatically deserialize all Ethernet packet-in messages. Set 'deseri
alizeEthPacketIns' to 'FALSE' if this feature is not required or when benchmarking core performance
2017-05-06 11:24:25.887 INFO [n.f.c.i.Controller] Controller role set to ACTIVE
2017-05-06 11:24:26.136 INFO [n.f.l.i.LinkDiscoveryManager] Link latency history set to 10 LLDP data points
2017-05-06 11:24:26.142 INFO [n.f.l.i.LinkDiscoveryManager] Latency update threshold set to +/- 0.5 (50.0%) of rolling historical average
2017-05-06 11:24:26.144 INFO [n.f.t.TopologyManager] Path metrics set to LATENCY
2017-05-06 11:24:26.145 INFO [n.f.t.TopologyManager] Will compute a max of 3 paths upon topology updates
2017-05-06 11:24:26.176 INFO [n.f.f.Forwarding] Default hard timeout not configured. Using 0.
2017-05-06 11:24:26.177 INFO [n.f.f.Forwarding] Default idle timeout set to 5.
2017-05-06 11:24:26.177 INFO [n.f.f.Forwarding] Default table ID not configured. Using 0x0.
2017-05-06 11:24:26.177 INFO [n.f.f.Forwarding] Default priority not configured. Using 1.
2017-05-06 11:24:26.177 INFO [n.f.f.Forwarding] Default flags will be set to SEND_FLOW_REM false.
2017-05-06 11:24:26.178 INFO [n.f.f.Forwarding] Default flags will be set to +TM_DRAFT+MAC+TD+ELAC+TDDT
```

Figure 50.1

Then the GUI can be accessed by pointing the browser to the following URL. (Figure 51)

<http://72.36.65.65:8080/ui/pages/index.html>

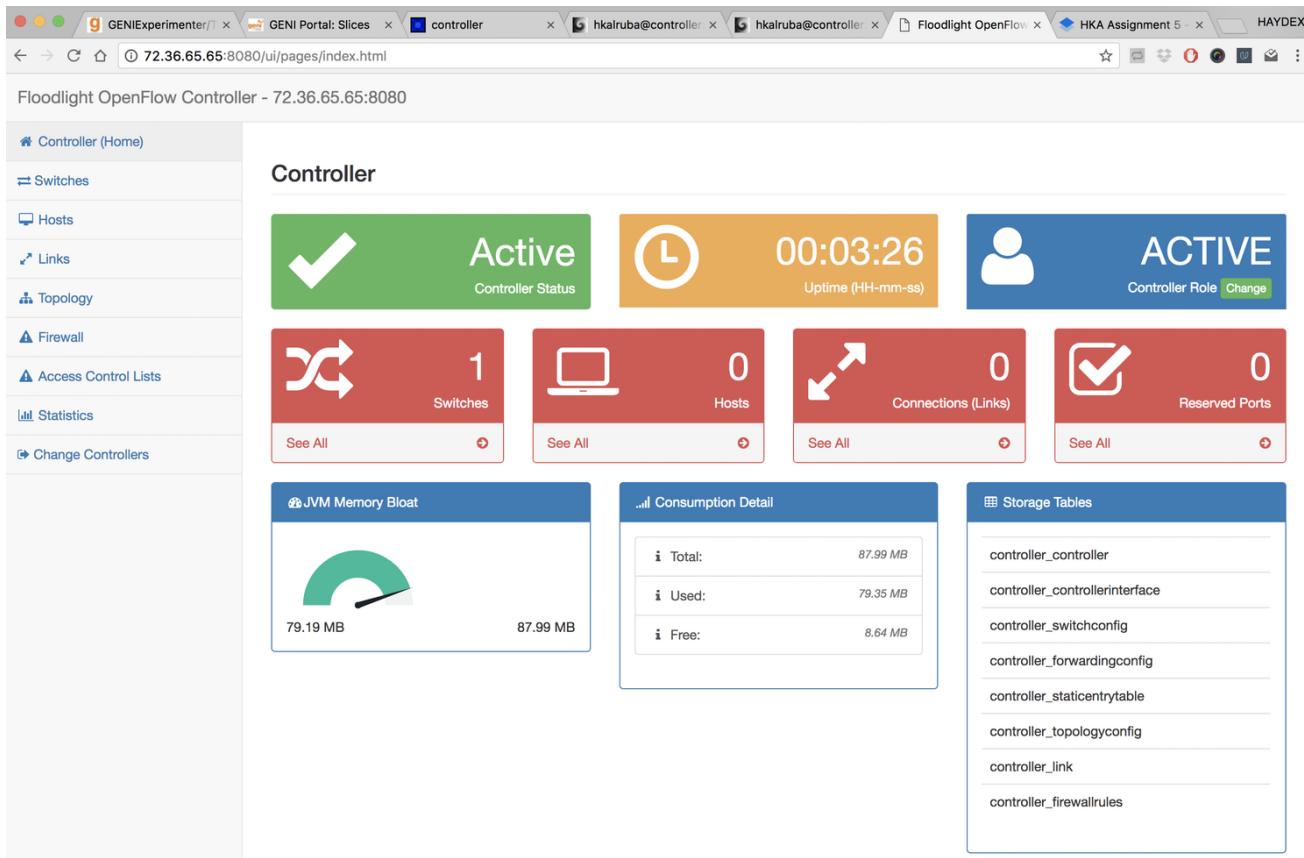


Figure 51

4f. Topology Details

Before we insert flows into the Open vSwitch, we are going to need all the details regarding the topology such as the Data Path IDentifier (DPID) of the Open vSwitch, the MAC addresses of the hosts, and the port numbers on which the Hosts are connected to the Open vSwitch etc. These details can be found by issuing the following command in a new Controller terminal (I have used GENI Desktop in this step): (Figure 52)

```
curl http://localhost:8080/wm/device/ | python -m json.tool
```

In the output, we obtain a list of the OpenFlow device (i.e OVS) that Floodlight Controller has learned about.

Next, we *ping* Host2 & Host3 from Host1 and then re-run the command. We will find that the output details will differ.(Figure 52)

We Highlight all of the important details we would need before we start manipulating flows. And we get the DPID by running the following command (Figure 55):

```
sudo ovs-ofctl show br0
```

Open vSwitch

```
DPID = 00:00:06:14:8d:84:32:4b
```

```
IP = 10.0.0.1
```

```
port = 2
```

```
switch = 00:00:06:14:8d:84:32:4b
```

```
mac = 02:65:d7:e9:5a:f5
```

```
IP = 10.0.0.2
```

```
port = 3
```

```
switch = 00:00:06:14:8d:84:32:4b
```

```
mac = 02:c3:33:08:c0:94
```

```
IP = 10.0.0.3
```

```
port = 1
```

```
switch = 00:00:06:14:8d:84:32:4b
```

```
mac = 02:93:9a:4e:77:89
```

As seen in (Figure 52)

```
hkalruba@controller:~$ curl http://localhost:8080/wm/device/ | python -m json.tool
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload   Total   Spent    Left  Speed
100  622     0  622     0      0  10243      0 --:--:-- --:--:-- --:--:-- 10366
{
  "devices": [
    {
      "attachmentPoint": [
        {
          "port": "2",
          "switch": "00:00:06:14:8d:84:32:4b"
        }
      ],
      "entityClass": "DefaultEntityClass",
      "ipv4": [
        "10.0.0.1"
      ],
      "ipv6": [],
      "lastSeen": 1494092466402,
      "mac": [
        "02:65:d7:e9:5a:f5"
      ],
      "vlan": [
        "0x0"
      ]
    },
    {
      "attachmentPoint": [
        {
          "port": "3",
          "switch": "00:00:06:14:8d:84:32:4b"
        }
      ],
      "entityClass": "DefaultEntityClass",
      "ipv4": [
        "10.0.0.2"
      ],
      "ipv6": [],
      "lastSeen": 1494092450734,
      "mac": [
        "02:c3:33:08:c0:94"
      ],
      "vlan": [
        "0x0"
      ]
    }
  ]
}
```

Figure 52

```
},
{
  "attachmentPoint": [
    {
      "port": "1",
      "switch": "00:00:06:14:8d:84:32:4b"
    }
  ],
  "entityClass": "DefaultEntityClass",
  "ipv4": [
    "10.0.0.3"
  ],
  "ipv6": [],
  "lastSeen": 1494092466396,
  "mac": [
    "02:93:9a:4e:77:89"
  ],
  "vlan": [
    "0x0"
  ]
}
hkalruba@controller:~$
```

Figure 52 Continued

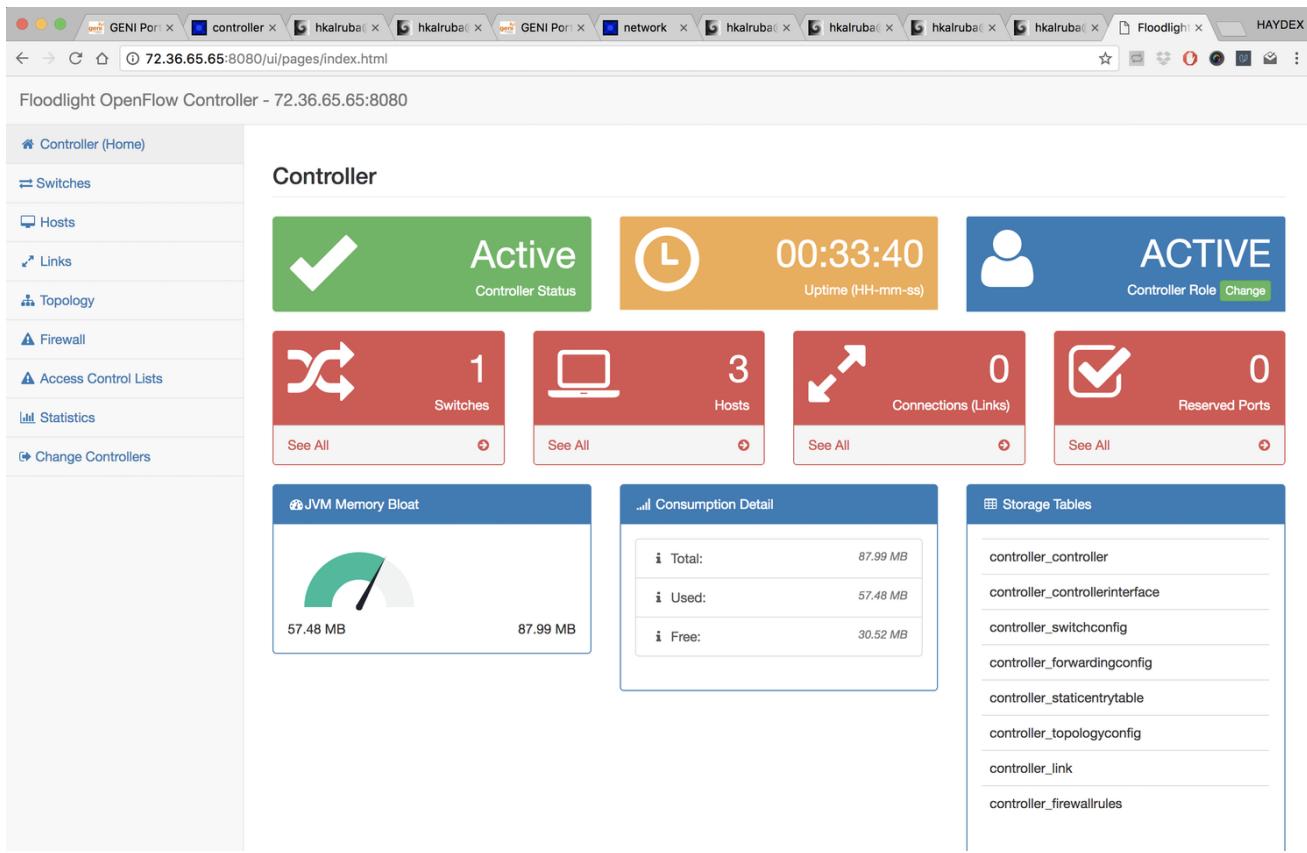


Figure 53 Floodlight detected 3 hosts after pinging them

4f. Now we run the traffic duplication controller

In the above example we ran a very simple learning switch controller. The power of OpenFlow comes from the fact that we can decide to forward the packet any way we want based on the supported OpenFlow actions. A very simple but powerful modification we can do, is to duplicate all the traffic of the switch out a specific port. This is very useful for application and network analysis. At the port where we duplicate traffic we can connect a device that does analysis. For this tutorial we are going to verify the duplication by doing tcpdump on two ports on the OVS switch.

1. We insert the flow to Duplicate Traffic

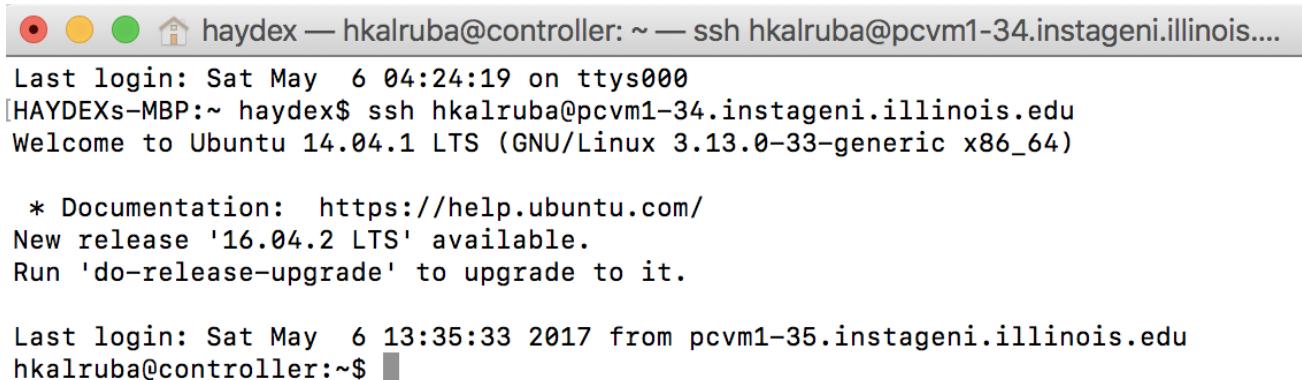
We are going to duplicate IPv4 traffic from Host1 destined to Host2 on Host3. We open a new *Controller* terminal and type the following flow:

```
curl -X POST -d '{"switch": "<DPID OF OPEN vSWITCH>", "name": "flow-1", "priority": "32768", "in_port": "<PORT OF 10.0.0.1>", "active": "true", "eth_type": "0x0800", "eth_src": "<MAC OF 10.0.0.1>", "eth_dst": "<MAC OF 10.0.0.2>", "ipv4_src": "10.0.0.1", "ipv4_dst": "10.0.0.2", "actions": "set_eth_dst=<MAC OF 10.0.0.2>,set_ipv4_dst=10.0.0.2,output=<PORT OF 10.0.0.2>,set_eth_dst=<MAC OF 10.0.0.3>,output=<PORT OF 10.0.0.3>" }' http://localhost:8080/wm/staticflowpusher/json
```

```
curl -X POST -d '{"switch": "00:00:06:14:8d:84:32:4b", "name": "flow-1", "priority": "32768", "in_port": "2", "active": "true", "eth_type": "0x0800", "eth_src": "02:65:d7:e9:5a:f5", "eth_dst": "02:c3:33:08:c0:94", "ipv4_src": "10.0.0.1", "ipv4_dst": "10.0.0.2", "actions": "set_eth_dst=02:c3:33:08:c0:94, set_ipv4_dst=1"}'
```

```
0.0.0.2, output=3, set_eth_dst=02:93:9a:4e:77:89, output=1"}' http://localhost:8080/wm/staticflowpusher/json
```

This time we login to the Controller from the mac Terminal because it is hard for me to write all of the code details in a GENI Desktop Terminal as it does not accept copying and pasting. (Figure 54)

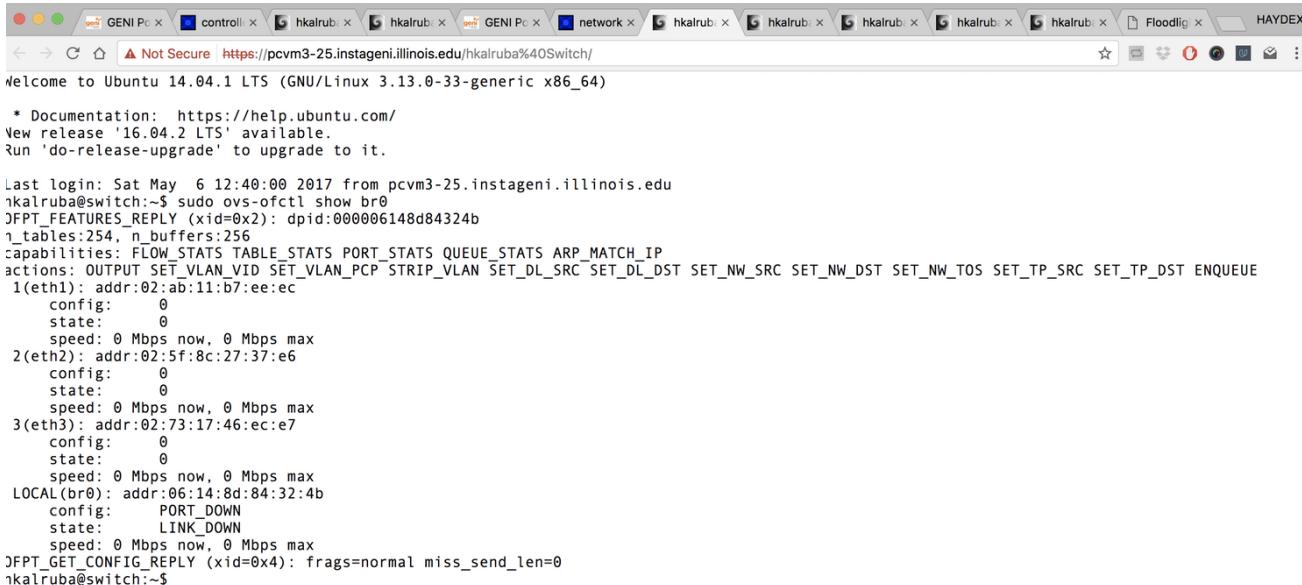


```
Last login: Sat May  6 04:24:19 on ttys000
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pcvm1-34.instageni.illinois.edu
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 13:35:33 2017 from pcvm1-35.instageni.illinois.edu
hkalruba@controller:~$
```

Figure 54



```
* Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 12:40:00 2017 from pcvm3-25.instageni.illinois.edu
hkalruba@switch:~$ sudo ovs-ofctl show br0
DPID_FEATURES_REPLY (xid=0x2): dpid:000006148d84324b
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
1(eth1): addr:02:a5:11:b7:ee:ec
    config: 0
    state: 0
    speed: 0 Mbps now, 0 Mbps max
2(eth2): addr:02:5f:8c:27:37:e6
    config: 0
    state: 0
    speed: 0 Mbps now, 0 Mbps max
3(eth3): addr:02:73:17:46:ec:e7
    config: 0
    state: 0
    speed: 0 Mbps now, 0 Mbps max
LOCAL(br0): addr:06:14:8d:84:32:4b
    config: PORT_DOWN
    state: LINK_DOWN
    speed: 0 Mbps now, 0 Mbps max
DPID_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
hkalruba@switch:~$
```

Figure 55 Show DPID of the OVS

```
[hkalruba@controller:~$ curl -X POST -d '{"switch":"06:14:8d:84:32:4b", "name":"flow-1", "priority":32768, "in_port":2, "active":true, "eth_type":0x0800, "eth_src":"02:65:d7:e9:5a:f5", "eth_dst":"02:c3:33:08:c0:94", "ipv4_src":10.0.0.1, "ipv4_dst":10.0.0.2, "actions":"set_eth_dst=02:c3:33:08:c0:94, set_ipv4_dst=10.0.0.2, output=3, set_eth_dst=02:93:9a:4e:77:89, output=1"}' http://localhost:8080/wm/staticflowpusher/json
{"status": "Entry pushed"}hkalruba@controller:~$
```

Figure 56 Duplication controller

To see that duplication is happening, on Host2 and Host3 , we run: sudo tcpdump -i eth1 (Figure 57)

```
hkalruba@host3:~$ sudo tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
16:19:30.589769 02:ab:11:b7:ee:ec (oui Unknown) > Broadcast, ethertype Unknown (0x8942), length 83:
  0x0000: 2000 0604 0002 0000 0207 0406 148d 8432 .....2
  0x0010: 4b04 0302 0001 0602 0078 fe0c 0026 e100 K.....x...&..
  0x0020: 0000 0614 8d84 324b 1808 05f5 e84b 1898 .....2K....K..
  0x0030: 76e4 e601 01fe 0c00 26e1 0100 0001 5bdf v.....&....[.
  0x0040: a1b0 3100 00 ..1..
16:19:45.622708 02:ab:11:b7:ee:ec (oui Unknown) > Broadcast, ethertype Unknown (0x8942), length 83:
  0x0000: 2000 0604 0002 0000 0207 0406 148d 8432 .....2
  0x0010: 4b04 0302 0001 0602 0078 fe0c 0026 e100 K.....x...&..
  0x0020: 0000 0614 8d84 324b 1808 05f5 e84b 1898 .....2K....K..
  0x0030: 76e4 e601 01fe 0c00 26e1 0100 0001 5bdf v.....&....[.
  0x0040: a1f7 eb00 00 ..
```

Figure 57 Host3 tcpdump

Now we see traffic from host1 to host2 showing up in the tcpdump window for host3 as shown in (Figure 57)

4g. Running a Port Forwarding Controller

Now let's do a slightly more complicated controller. OpenFlow gives the power to overwrite fields of your packets at the switch, for example the TCP source or destination port and do port forwarding. We can have clients trying to contact a server at port 5000, and the OpenFlow switch can redirect your traffic to a service listening on port 6000.

1. To test the controller we are going to use netcat. We open two terminals window on host2. In one terminal we run: (Figure 58)

```
nc -l 5000
```

and in the other terminal we run

```
nc -l 6000
```

```
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075 — 94x48
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
The authenticity of host ' [pc1.instageni.illinois.edu]:25075 ([72.36.65.20]:25075)' can't be established.
RSA key fingerprint is SHA256:NNzK+bYMN/+zrRoUpPKZmDUxmXcy1DLfmL5FY1CaF/s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[pc1.instageni.illinois.edu]:25075,[72.36.65.20]:25075' (RSA) to the list o
he list of known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 17:29:36 2017 from 76-218-20-136.lightspeed.ltrkar.sbcglobal.net
[hkalruba@host2:~$ nc -l 5000
[hkalruba@host2:~$ nc -l 6000

[A] Last login: Sat May  6 13:36:12 on ttys000
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
The authenticity of host ' [pc1.instageni.illinois.edu]:25075 ([72.36.65.20]:25075)' can't be establishe
d.
RSA key fingerprint is SHA256:NNzK+bYMN/+zrRoUpPKZmDUxmXcy1DLfmL5FY1CaF/s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[pc1.instageni.illinois.edu]:25075,[72.36.65.20]:25075' (RSA) to the list o
f known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 16:07:42 2017 from pcvm3-25.instageni.illinois.edu
[hkalruba@host2:~$ nc -l 6000]
```

Figure 58

2. We will check the normal functionality before the flow for a Port Forwarding Controller is inserted. We go to the terminal of host1 and connect to host2 at port 5000:

```
nc 10.0.0.2 5000
```

3. If we type something and we should see it in the terminal on host2 at port 5000. (Figure 59)

HKA Assignment 5

```

HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075 — 94x48
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
The authenticity of host 'pc1.instageni.illinois.edu:[25075] ([72.36.65.20]:25075)' can't be established.
RSA key fingerprint is SHA256:NWvk+bMNk+zrRouPZKmDUxXcy1Dfml5FY1CaF/s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[pc1.instageni.illinois.edu]:25075,[72.36.65.20]:25075' (RSA) to t
he list of known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 17:29:36 2017 from 76-218-20-136.lightspeed.ltrkar.sbcglobal.net
[hkalruba@host2:~$ nc -l 5800
hello world

[1]

```

```

HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075 — 103x48
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
Last login: Sat May  6 13:36:12 on ttys000
[HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
The authenticity of host 'pc1.instageni.illinois.edu:[25075] ([72.36.65.20]:25075)' can't be establishe
d.
RSA key fingerprint is SHA256:NWvk+bMNk+zrRouPZKmDUxXcy1Dfml5FY1CaF/s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[pc1.instageni.illinois.edu]:25075,[72.36.65.20]:25075' (RSA) to the list o
f known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 16:07:42 2017 from pcvm3-25.instageni.illinois.edu
[hkalruba@host2:~$ nc -l 6000

```

```

HAYDEX-MBP:~ haydex$ nc 10.0.0.2 5800

```

Figure 59

4. Now, we insert the flow for a Port Forwarding Controller: (Figure 60)

```
curl -X POST -d '{"switch":"<DPID OF OPEN vSWITCH>","name":"flow-2","priority":32768,"in_port":"<PORT OF 10.0.0.1>","active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":<MAC OF 10.0.0.1>, "eth_dst":<MAC OF 10.0.0.2>, "tp_dst":5000, "ipv4_src":10.0.0.1, "ipv4_dst":10.0.0.2, "actions":set_tp_dst=6000,output=<PORT OF 10.0.0.2>}' http://localhost:8080/wm/staticflowpusher/json
```

```
curl -X POST -d '{"switch":"00:00:06:14:8d:84:32:4b","name":"flow-2","priority":32768,"in_port":2,"active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":02:65:d7:e9:5a:f5, "eth_dst":02:c3:33:08:c0:94, "tp_dst":5000, "ipv4_src":10.0.0.1, "ipv4_dst":10.0.0.2, "actions":set_tp_dst=6000,output=3}' http://localhost:8080/wm/staticflowpusher/json
```

```

HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075 — 94x48
Last login: Sat May  6 13:36:12 on ttys000
HAYDEX-MBP:~ haydex$ ssh hkalruba@pc1.instageni.illinois.edu -p 25075
The authenticity of host 'pc1.instageni.illinois.edu:25075 ([72.36.65.20]:25075)' can't be established.
RSA key fingerprint is SHA256:NNzK+bVMN/+zrRouPKZmDUxeXcy1Dfml5FY1CaF/s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[pc1.instageni.illinois.edu]:25075,[72.36.65.20]:25075' (RSA) to the list of known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com/
New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May  6 17:29:36 2017 from 76-218-20-136.lightspeed.ltrkar.sbcglobal.net
hkalruba@host2:~$ nc -l 5000
hello world

```

```

hkalruba@host2:~$ nc -l 5000
hello world

```

```

haydex — hkalruba@controller:~ — ssh hkalruba@pcvcm1-34.instageni.illinois.edu -p 25075 — 94x48
hkalruba@controller:~$ curl -X POST -d '{"switch":"00:00:06:14:8d:84:32:4b", "name": "flow-3", "priority": "32768", "in_port": "2", "active": "true", "eth_type": "0x0800", "ip_proto": "0x06", "eth_src": "02:c3:33:08:c0:94", "eth_dst": "02:65:d7:e9:5a:f5", "tp_src": "6000", "tp_dst": "5000", "ipv4_src": "10.0.0.1", "ipv4_dst": "10.0.0.2", "actions": "set_tp_src=5000, output=3"}' http://localhost:8080/wm/staticflowpusher/json
{"status": "Entry pushed"}hkalruba@controller:~$ 

```

Figure 60

5. In the previous step, we inserted a flow to forward TCP traffic from Host1 destined to Host2 at port 5000 to port 6000. But Host1 still thinks it is speaking to Host2 at port 5000. So we need to insert a flow to handle traffic from Host2 Port 6000 for a seamless transition

```
curl -X POST -d '{"switch": "<DPID OF OPEN vSWITCH>", "name": "flow-3", "priority": "32768", "in_port": "<PORT OF 10.0.0.2>", "active": "true", "eth_type": "0x0800", "ip_proto": "0x06", "eth_src": "<MAC OF 10.0.0.2>", "eth_dst": "<MAC OF 10.0.0.1>", "tp_src": "6000", "ipv4_src": "10.0.0.2", "ipv4_dst": "10.0.0.1", "actions": "set_tp_src=5000, output=<PORT OF 10.0.0.1>"}' http://localhost:8080/wm/staticflowpusher/json
```

```
curl -X POST -d '{"switch": "00:00:06:14:8d:84:32:4b", "name": "flow-3", "priority": "32768", "in_port": "3", "active": "true", "eth_type": "0x0800", "ip_proto": "0x06", "eth_src": "02:c3:33:08:c0:94", "eth_dst": "02:65:d7:e9:5a:f5", "tp_src": "6000", "ipv4_src": "10.0.0.2", "ipv4_dst": "10.0.0.1", "actions": "set_tp_src=5000, output=2"}' http://localhost:8080/wm/staticflowpusher/json
```

4h. Running a Server Proxy Controller

3. We need to insert the following flow in the Controller terminal to implement a Server Proxy Controller:

```
curl -X POST -d '{"switch": "<DPID OF OPEN vSWITCH>", "name": "flow-4", "priority": "32768", "in_port": "<PORT OF 10.0.0.1>", "active": "true", "eth_type": "0x0800", "ip_proto": "0x06", "eth_src": "<MAC OF 10.0.0.1>", "eth_dst": "<MAC OF 10.0.0.2>", "tp_dst": "6000", "ipv4_src": "10.0.0.1", "ipv4_dst": "10.0.0.2", "actions": "set_eth_dst=<MAC OF 10.0.0.3>, set_tp_dst=7000, set_ipv4_dst=10.0.0.3, output=<PORT OF 10.0.0.3>"}' http://localhost:8080/wm/staticflowpusher/json
```

```
curl -X POST -d '{"switch":"00:00:06:14:8d:84:32:4b","name":"flow-4","priority":32768,"in_port":2,"active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":02:65:d7:e9:5a:f5, "eth_dst":02:c3:33:08:c0:94, "tp_dst":6000, "ipv4_src":10.0.0.1, "ipv4_dst":10.0.0.2, "actions":set_eth_dst=02:93:9a:4e:77:89, set_tp_dst=7000, set_ipv4_dst=10.0.0.3, output=1}' http://localhost:8080/wm/staticflowpusher/json
```

```
[hkalruba@controller:~$ curl -X POST -d '{"switch":"00:00:06:14:8d:84:32:4b","name":"flow-4","priority":32768,"in_port":2,"active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":02:65:d7:e9:5a:f5, "eth_dst":02:c3:33:08:c0:94, "tp_dst":6000, "ipv4_src":10.0.0.1, "ipv4_dst":10.0.0.2, "actions":set_eth_dst=02:93:9a:4e:77:89, set_tp_dst=7000, set_ipv4_dst=10.0.0.3, output=1}' http://localhost:8080/wm/staticflowpusher/json
```

Figure 61 Flow-4

4. In the previous step, we inserted a flow to forward TCP traffic from Host1 destined to Host2 at port 5000 to Host 3 at port 6000. But Host1 still thinks it is speaking to Host2 at port 5000. So we need to insert a flow to handle traffic from Host3 Port 6000 for a seamless transition.

```
curl -X POST -d '{"switch":<DPID OF OPEN vSWITCH>,"name":"flow-5","priority":32768,"in_port":<PORT OF 10.0.0.3>,"active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":<MAC OF 10.0.0.3>, "eth_dst":<MAC OF 10.0.0.1>, "tp_src":7000, "ipv4_src":10.0.0.3, "ipv4_dst":10.0.0.1, "actions":set_eth_src=<MAC OF 10.0.0.2>, set_ipv4_src=10.0.0.2, set_tp_src=6000, output=<PORT OF 10.0.0.1>}' http://localhost:8080/wm/staticflowpusher/json
```

```
curl -X POST -d '{"switch":"00:00:06:14:8d:84:32:4b","name":"flow-5","priority":32768,"in_port":1,"active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":02:93:9a:4e:77:89, "eth_dst":02:65:d7:e9:5a:f5, "tp_src":7000, "ipv4_src":10.0.0.3, "ipv4_dst":10.0.0.1, "actions":set_eth_src=02:c3:33:08:c0:94, set_ipv4_src=10.0.0.2, set_tp_src=6000, output=2}' http://localhost:8080/wm/staticflowpusher/json
```

```
[hkalruba@controller:~$ curl -X POST -d '{"switch":"00:00:06:14:8d:84:32:4b","name":"flow-5","priority":32768,"in_port":1,"active":true, "eth_type":0x0800, "ip_proto":0x06, "eth_src":02:93:9a:4e:77:89, "eth_dst":02:65:d7:e9:5a:f5, "tp_src":7000, "ipv4_src":10.0.0.3, "ipv4_dst":10.0.0.1, "actions":set_eth_src=02:c3:33:08:c0:94, set_ipv4_src=10.0.0.2, set_tp_src=6000, output=2}' http://localhost:8080/wm/staticflowpusher/json
{"status" : "Entry pushed"}hkalruba@controller:~$
```

Figure 62 Flow-5

5. We go back to the terminal of host1 and try to connect netcat to host2 port 6000 (Figure 63)

```

haydex — hkalruba@host2: ~ — ssh hkalruba@pc1.instageni.illinois.edu -p 2500
^C
hkalruba@Host2:~$ exit
Connection to pc1.instageni.illinois.edu closed.
HAYDEX-MB: ~ haydex — hkalruba@controller: ~ — ssh hkalruba@pcvm1-34.instageni.illinois.edu — 129x49
Welcome to HKA Assignment 5
20:47:51.765698 IP Host3-link->Host2-link-4, length 68
* Document
20:47:51.765617 IP Host1-link->Host2-link-4, length 68
New release
Run 'do-releas
8649, win 29200, options [mss
Length 0
Last login: 20:47:56.767374 ARP, Request v
20:47:56.767487 ARP, Request v
^C
hkalruba@H
length 28
20:47:56.774881 ARP, Reply Host1->Host2-link-4, length 68
^C
hkalruba@H
20:47:56.774928 ARP, Reply Host1->Host2-link-4, length 68
^C
hkalruba@H
length 28
20:47:56.774945 02:ab:11:b7:ea:00
nc: Address 0x8942), length 83:
hkalruba@H
0x0000: 2000 6604 0000
0x0010: 4b94 6302 0000
hkalruba@H
0x0020: 0000 6614 8d8
0x0030: 947c e681 01f
hkalruba@H
0x0040: 97a2 ab00 00
^C
9 packets captured
9 packets received by filter
0 packets dropped by kernel
hkalruba@Host2:~$ nc -l 7000
mac = 02:93:9a:4e:77:89

As seen in (Figure 52)

```

PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=11.4 ms
From 10.0.0.2: icmp_seq=1 ttl=64 time=12.2 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.15 ms
From 10.0.0.2: icmp_seq=2 Redirect Host(New nexthop: 10.0.0.2)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=2.17 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=3 Redirect Host(New nexthop: 10.0.0.2)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.43 ms
From 10.0.0.2: icmp_seq=3 Redirect Host(New nexthop: 10.0.0.2)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.95 ms (DUP!)
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, +3 duplicates, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.158/5.059/12.226/4.799 ms
hkalruba@Host1:~\$ nc 10.0.0.2 5000
[hello world]
hkalruba@Host1:~\$ hello world
The program 'hello' can be found in the following packages:
 * hello
 * hello-debhelper
 * ipv4_src="Ask your administrator to install one of them
output=2") http://hkalruba@Host1:~\$ nc 10.0.0.2 6000
{"status": "EnHello

CHAPTER 5

Figure 63 Testing the flow from host1 to host3

6. If the controller works correctly, we should have seen text appearing on the terminal window of host3. (Unsuccessful! there was no text appearing from host1 to host3)

4i. Deleting the bridge

```

haydex — hkalruba@switch: ~ — ssh hkalruba@pc1.instageni.illinois.edu -p 25...
speed: 0 Mbps now, 0 Mbps max
LOCAL(br0): addr:06:14:8d:84:32:4b
config: PORT_DOWN
state: LINK_DOWN
speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
[hkalruba@switch:~$ sudo ovs-vsctl show
89362c3a-1d64-451b-977d-7b2718a1ba20
Bridge "br0"
    Controller "tcp:72.36.65.65:6633"
        is_connected: true
        fail_mode: secure
    Port "eth1"
        Interface "eth1"
    Port "eth3"
        Interface "eth3"
    Port "br0"
        Interface "br0"
            type: internal
    Port "eth2"
        Interface "eth2"
ovs_version: "2.3.1"
[hkalruba@switch:~$ sudo ovs-vsctl del-br br0
hkalruba@switch:~$ 

```

Figure 64 Deleting the bridge in the OVS

5. FINISH : Deleting the resources in the slices.

The screenshot shows the GENI Portal interface. The top navigation bar includes links for Home, Tools, Partners, Help, and the user's name, Hayder Al Rubaye. The main content area is titled 'Project: UALR-CPSC-7341'. Below this, there are tabs for Slices, Members, Info, and Logs. The 'Slices' tab is active. A table lists four slices: 'controller', 'controller2', 'network', and 'network2'. The 'Actions' column for 'network2' has a dropdown menu open, with 'Delete resources' highlighted. At the bottom of the page, there is a footer with copyright information: 'GENI Portal Version 3.23', 'Copyright © 2016 Raytheon BBN Technologies All Rights Reserved - NSF Award CNS-0714770', and 'GENI is sponsored by the National Science Foundation'.

Name	Project	Owner	Expiration	Next Resource Expiration
controller	UALR-CPSC-7341	Hayder Al Rubaye	In 5 days ✓	In 5 days ✓
controller2	UALR-CPSC-7341	Hayder Al Rubaye	In 5 days ✓	No resources
network	UALR-CPSC-7341	Hayder Al Rubaye	In 5 days ✓	In 5 days ✓
network2	UALR-CPSC-7341	Hayder Al Rubaye	In 5 days ✓	No resources

Figure 65

The screenshot shows a web browser window titled "GENI Portal: Slices". The URL is https://portal.geni.net/secure/sliverdelete.php?slice_id=ca588202-474a-435c-b6c7-bd64cde5b91b&am_id=267. The page header includes "Home", "Tools", "Partners", "Help", and a user "Hayder Al Rubaye". Below the header, the breadcrumb navigation shows "Home → Project UALR-CPSC-7341 → Slice network → Delete Sliver from network". The main content area displays the message "Deleting resources on slice: network" and "Issued delete resources at 1 of 1 aggregates.". At the bottom, there are links to "Back to All slices" and "Back to Slice network". A footer bar at the bottom right contains the text "GENI Portal Version 3.23", "Copyright © 2016 Raytheon BBN Technologies All Rights Reserved - NSF Award CNS-0714770", and "GENI is sponsored by the National Science Foundation".

Figure 66

PROBLEMS

CHAPTER 5

Problem 3:

Solution:

The ASCII code for the word "Networking"

N e t w o r k i n g

078 101 116 119 111 114 107 105 110 103

The binary representation of these numbers is

01001110 01100101 01110100 01110111 01101111 01110010 01101011 01101001 01101110 0110 0111

Checksum Calculation:

0100 1110 0110 0101 + 0111 0100 0111 0111 = 1100 0010 1101 1100

1100 0010 1101 1100 + 0110 1111 0111 0010 = 1 0011 0010 0100 1110

(overflow, wrap around) => 0011 0010 0100 1111

0011 0010 0100 1111 + 0110 1011 0110 1001 = 1001 1101 1011 1000 1001 1101 1011

1000+0110 1110 0110 0111 = 1 0000 1100 0001 1111 =>0000 1100 0001 1111

(overflow)

1's complement of 0000 1100 0001 1111 = 1111 0011 1110 0000.

Problem 5:

Solution:

G=10011

then r = 4

D + r 0s= 1010101010 0000

divide 10011 into 1010101010 0000,

Remainder of R=0100

Problem 10:

Solution:

a) The formula of A's average throughput is $pA(1-pB)$.

Total efficiency is $pA(1-pB) + pB(1-pA)$

b) A's throughput = $pA(1-pB) = 2pB(1-pB) = 2pB - 2(pB)^2$.

B's throughput = $pB(1-pA) = pB(1-2pB) = pB - 2(pB)^2$.

A's throughput is not twice as large as B's.

To make $pA(1-pB) = 2 pB(1-pA)$. $pA = 2 - (pA / pB)$.

c) A's throughput = $2p(1-p)N-1$.

Any other node's throughput = $p(1-p)N- 2(1-2p)$

Problem 17:

Solution:

for a 10Mbps = $K*512*0.0001 = 5.12 \text{ ms}$

for a 100Mbps = $K*512*0.00001 = 0.512\text{ms}$

Problem 23:

Solution:

If all the 11= 9+2 nodes send out data at the maximum possible rate of 100 Mbps, a total aggregate throughput of $11*100 = 1100 \text{ Mbps}$ is possible.