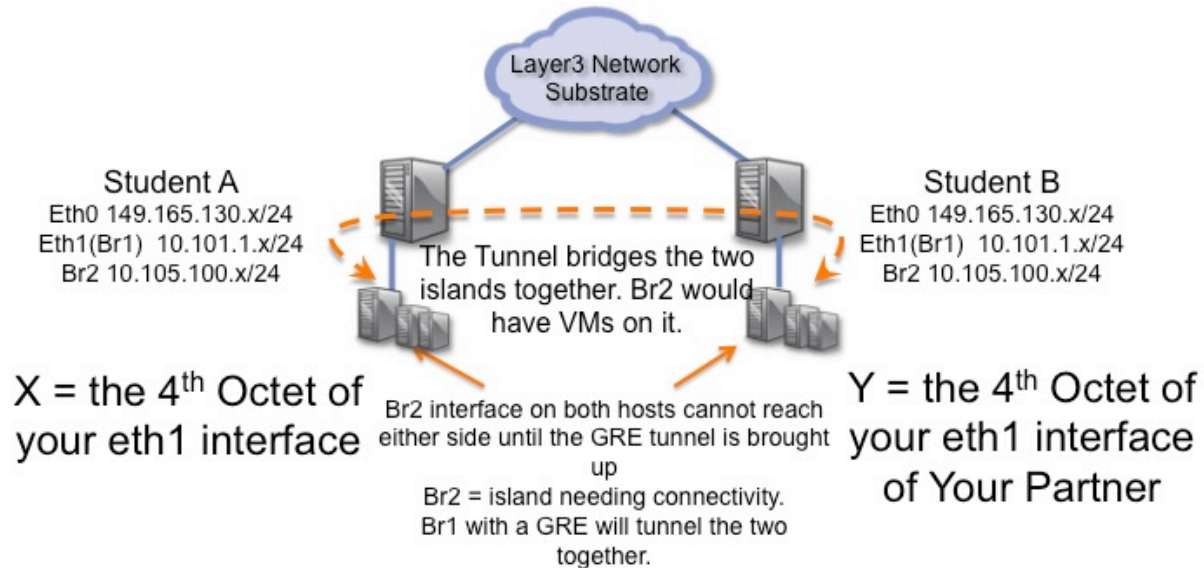


# OpenvSwitch and FloodLight Lab



Brent Salisbury

<http://networkstatic.net>

@networkstatic

## OVS Lab

**Step 1.** With your two terminal windows, one will be for the Floodlight controller and the other will be for OpenvSwitch (OVS). At the end you will open a web page in your web browser when you see a URL. At each "\$" sign there is a command that follows that you will enter into your VM. 10.101.1.x equals the last octet of your eth1 address. Google copy/paste can be weird. Highlight and control+c will copy into clickb

**Step 2.** Create two bridges in OVS.

br1 is the bridge that is reachable from both VMs and will serve as the tunnel endpoints (TEP). Each Bridge creates a unique DPID.

```
$ sudo ovs-vsctl add-br br1
```

br2 is the island that is unreachable from Student A to Student B.

```
$ sudo ovs-vsctl add-br br2
```

View the bridges you created in OVS.

```
$ sudo ovs-vsctl show
```

**Step 3.** Add interface eth1 to the bridge interface br1. Move the address on eth1 to br1. Think of this as adding a physical interface to a Vlan with a Layer3 virtual interface.

(x=your VM).

```
$ sudo ifconfig eth1 0 (zero out the IP)
```

```
$ sudo ifconfig br1 10.101.1.x netmask 255.255.255.0
```

```
$ sudo ifconfig br2 10.105.100.x netmask 255.255.255.0
```

```
$ sudo ovs-vsctl add-port br1 eth1
```

**Step 4.** Attach OpenvSwitch to the controller for both of our bridges (x=your VM). This will enable datapath lookups to punt to the FloodLight OpenFlow controller.

```
$ sudo ovs-vsctl set-controller br1 tcp:10.101.1.x:6633
```

```
$ sudo ovs-vsctl set-controller br2 tcp:10.101.1.x:6633
```

**Step 5.** Build the GRE tunnel to connect the two islands (br2) over the connected bridge (br1). y= your lab partners IP address for interface br1 (formerly eth1). If unsure run '**sudo ifconfig br1**'

```
$ ping 10.105.100.y (Pings should be unreachable and fail)
```

Add the GRE tunnel. Here you create the interface gre0, apply it to br2 (island) interface and point to your neighbors reachable br1 interface.

```
$ sudo ovs-vsctl add-port br2 gre0 -- set interface gre0 type=gre  
options:remote_ip=10.101.1.y
```

```
$ ping 10.105.100.y (Pings should be reachable now)
```

**Step 6.** Start the Floodlight controller. In one of your windows do the following.

```
$ cd ~/floodlight/target/
```

```
$ sudo java -jar floodlight.jar
```

You should see in OVS being connected to a controller for each bridge now,

```
$ sudo ovs-vsctl show
```

```
Controller "tcp:10.101.1.x:6633"  
    is_connected: true
```

**Step 7.** Find the reactive flow you created. The flows timeout in ~5 seconds so you need to type quick! Up arrow will repeat the previous command in the bash shell.

```
$ sudo ovs-dpctl dump-flows br2
```

If you cannot get the command off in time it will look something like this.

```
in_port(0),eth(src=76:44:9c:aa:3b:45,dst=4e:23:1b:24:81:43),eth_type(0x0800),ipv4(src=10.105  
.100.11,dst=10.105.100.10,proto=1,tos=0,ttl=64,frag=no),icmp(type=8,code=0), packets:1,  
bytes:98, used:4.820s, actions:2
```

The configuration at this point will look like this (x=your eth1, y=neighbors eth1):

```
$ sudo ovs-vsctl show
```

```
c7130ff2-bed3-4787-9a40-cdc3ad888fba  
    Bridge "br1"  
        Controller "tcp:10.101.1.x:6633"  
            is_connected: true  
        Port "eth1"  
            Interface "eth1"  
        Port "br1"  
            Interface "br1"
```

```

        type: internal
Bridge "br2"
  Controller "tcp:10.101.1.x:6633"
    is_connected: true
  Port "br2"
    Interface "br2"
      type: internal
  Port "gre0"
    Interface "gre0"
      type: gre
      options: {remote_ip="10.101.1.y"}

```

**Step 8.** Look at the FloodLight controller through the web UI.

\$ <http://149.165.130.x:8080/ui/index.html> (x=your host eth0 if unsure run 'sudo ifconfig eth0')

You will have two Data Path IDs (DPID) for each bridge. As you start FloodLight you will see your OpenvSwitch attach to the controller.

Explore different pings and look at the counters and how they increment based on traffic.

## *The Quick Build*

Host A

```

sudo ifconfig eth1 0
sudo ovs-vsctl add-br br1
sudo ovs-vsctl add-br br2
sudo ovs-vsctl add-port br1 eth1
sudo ifconfig br1 10.101.1.x netmask 255.255.255.0
sudo ifconfig br2 10.105.100.x netmask 255.255.255.0
sudo ovs-vsctl add-port br2 gre0 -- set interface gre0 type=gre options:remote_ip=10.101.1.1y
sudo ovs-vsctl set-controller br1 tcp:10.101.1.x:6633
sudo ovs-vsctl set-controller br2 tcp:10.101.1.x:6633
cd ~/floodlight/target/
sudo java -jar floodlight.jar &

```

ovs-vsctl show to check if OVS is running.

Will return something like this "36e7a773-fe56-42f6-aec0-25eb682f18eb" DPID+extras.

Host B

```

sudo ifconfig eth1 0
sudo ovs-vsctl add-br br1
sudo ovs-vsctl add-br br2

```

```
sudo ovs-vsctl add-port br1 eth1
sudo ifconfig br1 10.101.1.x netmask 255.255.255.0
sudo ifconfig br2 10.105.100.x netmask 255.255.255.0
sudo ovs-vsctl add-port br2 gre0 -- set interface gre0 type=gre options:remote_ip=10.101.1.y
sudo ovs-vsctl set-controller br1 tcp:10.101.1.x:6633
sudo ovs-vsctl set-controller br2 tcp:10.101.1.x:6633
cd ~/floodlight/target/
sudo java -jar floodlight.jar &
```

**\*\*If you want to cleanup and start over you can paste the following.**

Cleanup Host A

```
sudo killall java
sudo ifconfig eth1 10.101.1.x netmask 255.255.255.0
sudo ifconfig br1 0
sudo ifconfig br2 0
sudo ovs-vsctl del-br br1
sudo ovs-vsctl del-br br2
```

Cleanup Host B

```
sudo killall java
sudo ifconfig eth1 10.101.1.x netmask 255.255.255.0
sudo ifconfig br1 0
sudo ifconfig br2 0
sudo ovs-vsctl del-br br1
sudo ovs-vsctl del-br br2
```

*If you need to start the OpenvSwitch process.*

Insert OVS kernel module

```
% sudo insmod ~/openvswitch/datapath/linux/openvswitch_mod.ko
```

Start ovsdb-server

```
% sudo ovsdb-server /usr/local/etc/openvswitch/conf.db \
--remote=punix:/usr/local/var/run/openvswitch/db.sock \
--remote=db:Open_vSwitch,manager_options \
--private-key=db:SSL,private_key \
--certificate=db:SSL,certificate \
--bootstrap-ca-cert=db:SSL,ca_cert --pidfile --detach --log-file
```

Start vswitch

```
% sudo ovs-vswitchd --pidfile --detach
```