

SDN 实验之 Open vSwitch+Floodlight

实验时间：2015 年 2 月

实验目的：模拟实现简单 SDN 场景

学习参考：<http://www.sdnlab.com/>



新疆电信 高鹏

QQ: osoon@vip.qq.com

2015 年 2 月 28 日 星期六

一、实验简介

搭建 SDN 环境少不了 SDN 交换机，SDN 交换机跟普通交换机最大的区别就是将普通交换机的数据平面和控制平面相分离，SDN 交换机只负责数据的转发，而控制指令则由更上一级的控制器下发。由此实现控制层和转发层面的分离。

Open vSwitch（下面简称 OVS）是一个高质量的、多层虚拟交换机。OVS 遵循开源 Apache2.0 许可，通过可编程扩展，OVS 可以实现大规模网络的自动化（配置、管理、维护），同时支持现有标准管理接口和协议（比如 NetFlow、sFlow、SPAN、RSPAN、CLI、LACP、802.1ag 等）。此外 OVS 支持多种 linux 虚拟化技术，包括 Xen/XenServer，KVM，和 VirtualBox 等。

Floodlight 是目前主流的 SDN 控制器之一，它的稳定性、易用性已经得到 SDN 专业人士以及爱好者们的一致好评，并因其完全开源，这让 SDN 网络世界变得更加有活力。控制器作为 SDN 网络中的重要组成部分，能集中地灵活控制 SDN 网络，为核心网络及应用创新提供了良好的扩展平台，本文档主要介绍基于 Java 语言的开源 SDN 控制器 Floodlight 的安装部署。

本文通过 Ubuntu 平台安装部署 OVS 和 Floodlight，并介绍 OVS 和 Floodlight 上的一些常用操作，让读者进一步了解 OpenFlow 交换机和 Floodlight 控制器配合实现简单的控制和转发 SDN 实验的演示。

二、实验环境

★虚拟机 VMware Workstation 11.0，下载地址如下：

<https://download3.vmware.com/software/wkst/file/VMware-workstation-full-11.0.0-2305329.exe>

激活码：1F04Z-6D111-7Z029-AV0Q4-3AEH8

★OVS 系统需求：Ubuntu13.10（内核版本 3.11），Ubuntu Linux 下载地址如下：

<http://releases.ubuntu.com/>

★OVS 选择较新的 Open vSwitch 2.3.0 版本。目前最新版本为 2.3.1；

★Floodlight 系统需求：Ubuntu 10.04（Natty）及以上版本（运行 Ant1.8.1 及以下版本）；

★实验中 OVS 我们选择 ubuntu13.10 桌面版 / 服务版，Floodlight 选择 ubuntu13.10 服务版。

★OVS 安装包和 Floodlight 安装包在试验中下载到虚拟机内，均来自官网下载。

三、Open vSwitch 安装部署和基本操作

1.准备工作

- (1) 安装虚拟机 Vmware Workstation 11.0
- (2) 在虚拟机下安装 Ubuntu13.10（可以在官网下载到）
- (3) 在正式下载安装 OVS 之前我们需要安装一些系统组件及库文件以作为 OVS 正确运行的环境依赖。请切换至 root 用户进行操作。然后更新系统。

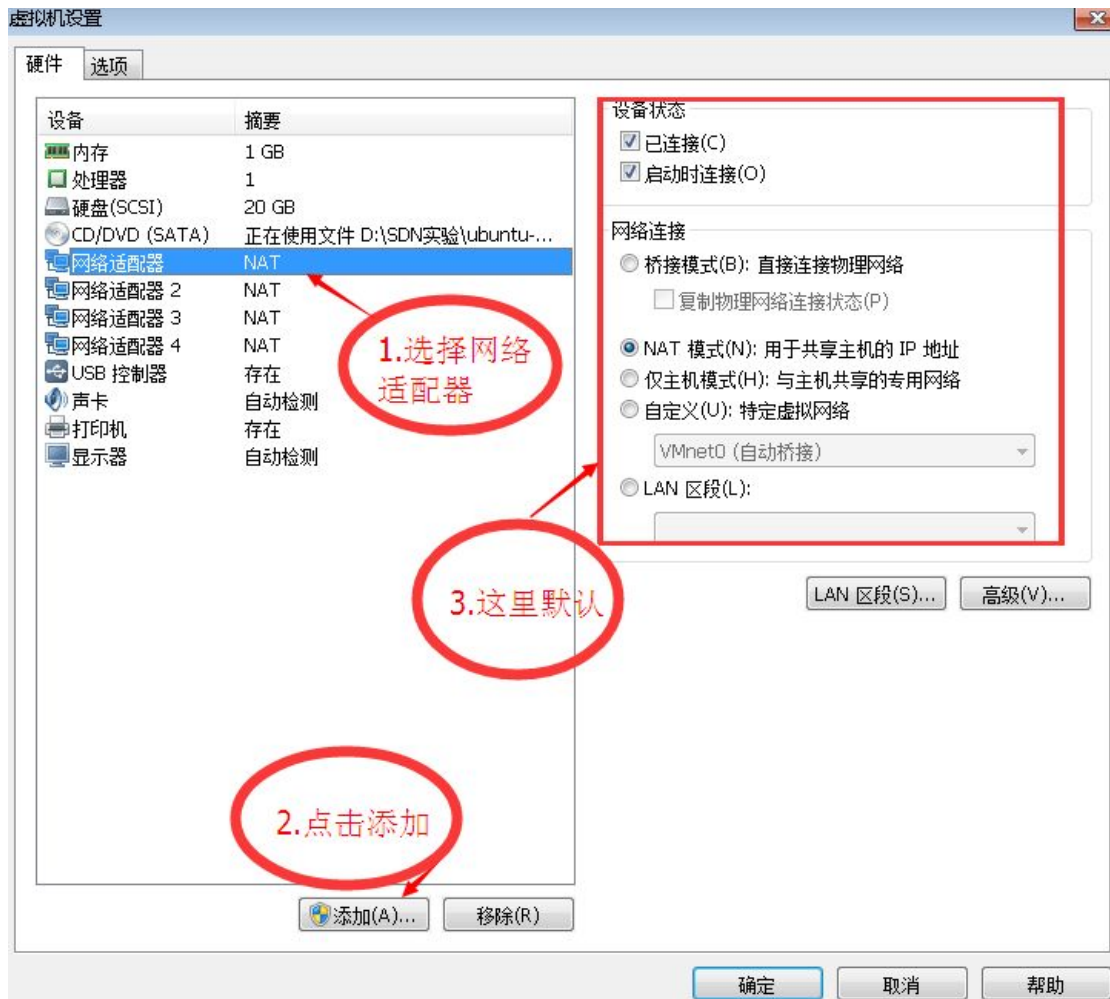
```
# apt-get update
```

```
# apt-get install -y build-essential
```

安装完 ubuntu13.10 桌面版登陆界面



- (4) 做实验之前，在需要安装 ovs（虚拟交换机）的虚拟机上添加几块虚拟网卡 eth1、eth2、eth3,方便后续虚拟出几台主机做实验观察，添加完网卡需要重启虚拟机，添加方法如下图所示：



(5) 添加完网卡，启动虚拟机后，需要在网卡配置文件 `/etc/network/interfaces` 中添加如下内容

```
auto eth0
```

```
iface eth0 inet dhcp
```

```
auto eth1
```

```
iface eth1 inet dhcp
```

```
auto eth2
```

```
iface eth2 inet dhcp
```

```
auto eth3
```

```
iface eth3 inet dhcp
```

2. 下载我之前就准备好的 ovs 安装和部署脚本，令其自动执行，直至安装结束，脚本内容及其步骤如下：

提示：将脚本存放在根目录下的 `ovs.sh` 文件内，然后使用 `source ovs.sh` 指令执行该文件即可完成 `ovs` 的安装，第一次实验建议按照后续的步骤安装。

```

#!/bin/bash

# Make sure only root can run our script
if [ "$(id -u)" != "0" ]; then
    echo "You need to be 'root' dude." 1>&2
    exit 1
fi

apt-get update

apt-get install -y build-essential fakeroot

#apt-get install -y gcc uml-utilities libtool build-essential \
#pkg-config python-qt4 python-twisted-conch debhelper \
#python-all

echo "=====INSTALL
openvswitch-2.3.0=====
if [ -f openvswitch-2.3.0.tar.gz ]
then
    echo "openvswitch-2.3.0.tar.gz has exist"
else
    wget http://openvswitch.org/releases/openvswitch-2.3.0.tar.gz
fi
if [ -d openvswitch-2.3.0 ]
then
    rm -r openvswitch-2.3.0
fi
tar -xzf openvswitch-2.3.0.tar.gz
# Install openvswitch
cd openvswitch-2.3.0
make clean
./configure --with-linux=/lib/modules/`uname -r`/build 2>/dev/null
make && make install

```

```

# install Open vSwitch kernel module

#rmmod openvswitch

#modprobe libcrc32c

#modprobe gre

#insmod datapath/linux/openvswitch.ko

make modules_install

/sbin/modprobe openvswitch

mkdir -p /usr/local/etc/openvswitch

ovsdb-tool          create          /usr/local/etc/openvswitch/conf.db

vswitchd/vswitch.ovsschema  2>/dev/null

#ovsdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock \

#      --private-key=db:Open_vSwitch,SSL,private_key \
#      --certificate=db:Open_vSwitch,SSL,certificate \
#      --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert \
#      --pidfile --detach

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

ovs-vsctl --no-wait init

ovs-vswitchd --pidfile --detach

ovs-vsctl show

```

```
root@gaopeng: /
root@gaopeng:/# wget 61.128.117.227/ovs.sh
--2015-02-25 12:09:04-- http://61.128.117.227/ovs.sh
正在连接 61.128.117.227:80... 已连接。
已发出 HTTP 请求，正在等待响应... 200 OK
长度：1742 (1.7K) [application/x-sh]
正在保存至：“ovs.sh”
100%[=====] 1,742 --K/s 用时 0s
2015-02-25 12:09:04 (19.5 MB/s) - 已保存 “ovs.sh” [1742/1742])
root@gaopeng:/# ls
bin      dev      initrd.img  lost+found  opt      root     srv      usr
boot     etc      initrd.img.old  media      ovs.sh   run      sys      var
cdrom    home     lib         mnt         proc     sbin     tmp      vmlinuz
root@gaopeng:/# cat ovs.sh
#!/bin/bash

# Make sure only root can run our script
if [ "$(id -u)" != "0" ]; then
    echo "You need to be 'root' dude." 1>&2
    exit 1
fi

apt-get update
apt-get install -y build-essential fakeroot

#apt-get install -y gcc uclibc uclibc-dev libtool build-essential \
#pkg-config python-qt4 python-twisted-conch debhelper \
#python-all
```

Ovs 安装脚本步骤如下（可以放在脚本中一次性执行）：

（1）下载 OVS 2.3.0 安装包。

```
# wget
http://openvswitch.org/releases/openvswitch-2.3.0.tar.gz
```

（2）解压 OVS 2.3.0 安装包。

```
# tar -xzf openvswitch-2.3.0.tar.gz
```

（3）构建基于 Linux 内核的交换机。

```
# cd openvswitch-2.3.0
# make clean
# ./configure --with-linux=/lib/modules/$(uname -r)/build
2>/dev/null
```

（4）编译并安装 OVS 2.3.0。

```
# make && make install
```

（5）如果需要 OVS 支持 VLAN 功能，还需要加载 openvswitch.ko 模块，如果不需要，此步可以忽略。

```
# modprobe gre
```



```
# insmod datapath/linux/openvswitch.ko
```

(6) 我们需要安装并加载构建的内核模块。

```
# make modules_install
```

```
# /sbin/modprobe openvswitch
```

(7) 使用 `ovsdb` 工具初始化配置数据库。

```
# mkdir -p /usr/local/etc/openvswitch
```

```
# ovsdb-tool create /usr/local/etc/openvswitch/conf.db
```

```
vswitchd/vswitch.ovsschema 2>/dev/null
```

至此，如果没有报错的话 `OVS` 的部署已经成功完成。如果中间步骤出现问题，请仔细检查是否按步骤进行或有无单词拼写错误。

(8) 在启动 `OVS` 之前，我们需要先启动 `ovsdb-server` 配置数据库。注意后面的命令大部分是由两个短“-”组成的。

```
# ovsdb-server -v
```

```
--remote=punix:/usr/local/var/run/openvswitch/db.sock
```

```
--remote=db:Open_vSwitch,Open_vSwitch,manager_options
```

```
--private-key=db:Open_vSwitch,SSL,private_key
```

```
--certificate=db:Open_vSwitch,SSL,certificate
```

```
--bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile
```

```
--detach
```

(9) 首次用 `ovsdb-tool` 创建数据库时需用 `ovs-vsctl` 命令初始化下数据库。

```
# ovs-vsctl --no-wait init
```

(10) 启动 `OVS` 主进程

```
# ovs-vswitchd --pidfile --detach
```

(11) 查看 `OVS` 进程是否启动。

```
# ps -ef |grep ovs
```

(12) 我们可以通过如下命令查看所安装 `OVS` 的版本号。

```
# ovs-vsctl --version
```

`Ovs` 进程和版本号显示如下图所示：

```
root@gaopeng: /openvswitch-2.3.0
root@gaopeng:/openvswitch-2.3.0# ps -ef | grep ovs
root      13863  1645  0 12:16 ?        00:00:00 ovsdb-server --remote=punix:/usr/
local/var/run/openvswitch/db.sock --remote=db:Open_vSwitch,Open_vSwitch,manager_o
ptions --private-key=db:Open_vSwitch,SSL,private_key --certificate=db:Open_vSwitc
h,SSL,certificate --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert --pidfile --det
ach
root      13866  1645  0 12:16 ?        00:00:00 ovs-vswitchd --pidfile --detach
root      13966  2363  0 12:20 pts/1    00:00:00 grep --color=auto ovs
root@gaopeng:/openvswitch-2.3.0# ovs-vsctl --version
ovs-vsctl (Open vSwitch) 2.3.0
Compiled Feb 25 2015 12:15:17
DB Schema 7.6.0
root@gaopeng:/openvswitch-2.3.0#
```

四、Floodlight 安装及其部署及其常用指令

1. 安装 JDK, Ant

```
# sudo apt-get install build-essential default-jdk ant python
-dev eclipse

正在设置 libatk-wrapper-java (0.30.4-0ubuntu4) ...
正在设置 libatk-wrapper-java-jni:i386 (0.30.4-0ubuntu4) ...
正在设置 libosgi-compndium-java (4.3.0-1) ...
正在设置 libfelix-gogo-runtime-java (0.10.0-2) ...
正在设置 libfelix-gogo-command-java (0.12.0-2) ...
正在设置 libfelix-gogo-shell-java (0.10.0-2) ...
正在设置 eclipse-rcp (3.8.1-4) ...
正在设置 eclipse-platform (3.8.1-4) ...
正在设置 eclipse-jdt (3.8.1-4) ...
正在设置 eclipse-pde (3.8.1-4) ...
正在设置 eclipse (3.8.1-4) ...
正在设置 libfelix-shell-java (1.4.2-3) ...
正在设置 libfelix-utils-java (1.1.0-3) ...
正在处理用于 libc-bin 的触发器...
正在处理用于 ca-certificates 的触发器...
Updating certificates in /etc/ssl/certs... 0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
done.
done.
root@gaopeng:/# sudo apt-get install build-essential default-jdk ant python-dev eclipse
```

2. 安装 git

```
# sudo apt-get install git-core
```

```

root@gaopeng:/# sudo apt-get install git-core
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会安装下列额外的软件包:
  git git-man liberror-perl
建议安装的软件包:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch
下列【新】软件包将被安装:
  git git-core git-man liberror-perl
升级了 0 个软件包, 新安装了 4 个软件包, 要卸载 0 个软件包, 有 74 个软件包未被升级。
需要下载 8,488 kB 的软件包。
解压缩后会消耗掉 19.7 MB 的额外空间。
您希望继续执行吗? [Y/n]y
获取: 1 http://cn.archive.ubuntu.com/ubuntu/ saucy/main liberror-perl all 0.17-1 [23.8 kB]
获取: 2 http://cn.archive.ubuntu.com/ubuntu/ saucy/main git-man all 1:1.8.3.2-1 [670 kB]
获取: 3 http://cn.archive.ubuntu.com/ubuntu/ saucy/main git i386 1:1.8.3.2-1 [7,793 kB]
获取: 4 http://cn.archive.ubuntu.com/ubuntu/ saucy/main git-core all 1:1.8.3.2-1 [1,386 B]
下载 8,488 kB, 耗时 14秒 (598 kB/s)
Selecting previously unselected package liberror-perl.
(正在读取数据库 ... 系统当前共安装有 70779 个文件和目录。)
正在解压缩 liberror-perl (从 .../liberror-perl_0.17-1_all.deb) ...
Selecting previously unselected package git-man.
正在解压缩 git-man (从 .../git-man_1%3a1.8.3.2-1_all.deb) ...
Selecting previously unselected package git.
正在解压缩 git (从 .../git_1%3a1.8.3.2-1_i386.deb) ...
Selecting previously unselected package git-core.
正在解压缩 git-core (从 .../git-core_1%3a1.8.3.2-1_all.deb) ...
正在处理用于 man-db 的触发器...
正在设置 liberror-perl (0.17-1) ...
正在设置 git-man (1:1.8.3.2-1) ...
正在设置 git (1:1.8.3.2-1) ...
正在设置 git-core (1:1.8.3.2-1) ...
root@gaopeng:/#

```

3. 安装 Floodlight

```

# git clone git://github.com/floodlight/floodlight.git

# cd floodlight

# ant

```

```

root@gaopeng:/# git clone git://github.com/floodlight/floodlight.git
正克隆到 'floodlight'...
remote: Counting objects: 28459, done.
remote: Total 28459 (delta 0), reused 0 (delta 0), pack-reused 28459
接收对象中: 100% (28459/28459), 77.75 MiB | 163.00 KiB/s, done.
处理 delta 中: 100% (14356/14356), done.
Checking connectivity... done
root@gaopeng:/# cd floodlight
root@gaopeng:/floodlight# ant
Buildfile: /floodlight/build.xml

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

compile:
    [javac] Compiling 492 source files to /floodlight/target/bin
    [javac] 注: 某些输入文件使用或覆盖了已过时的 API。
    [javac] 注: 有关详细信息, 请使用 -Xlint:deprecation 重新编译。
    [javac] 注: 某些输入文件使用了未经检查或不安全的操作。
    [javac] 注: 有关详细信息, 请使用 -Xlint:unchecked 重新编译。
    [copy] Copying 54 files to /floodlight/target/bin

compile-test:
    [javac] Compiling 82 source files to /floodlight/target/bin-test

dist:
    [jar] Building jar: /floodlight/target/floodlight.jar
    [jar] Building jar: /floodlight/target/floodlight-test.jar

BUILD SUCCESSFUL
Total time: 17 seconds
root@gaopeng:/floodlight#

```

4. 运行 Floodlight

```
# java -jar target/floodlight.jar
```

```

root@gaopeng:/floodlight# java -jar target/floodlight.jar
12:58:29.401 INFO [n.f.c.m.FloodlightModuleLoader:main] Loading modules from src/main/resources/floodlightdefault.properties
12:58:30.819 INFO [n.f.c.i.Controller:main] Controller role set to ACTIVE
12:58:30.837 INFO [n.f.f.Forwarding:main] Default hard timeout not configured. Using 0.
12:58:30.838 INFO [n.f.f.Forwarding:main] Default idle timeout not configured. Using 5.
12:58:30.838 INFO [n.f.f.Forwarding:main] Default priority not configured. Using 1.
12:58:31.035 INFO [o.s.s.i.c.FallbackCCProvider:main] Cluster not yet configured; using fallback local configuration
12:58:31.040 INFO [o.s.s.i.SyncManager:main] [32767] Updating sync configuration ClusterConfig [allNodes={32767=Node
7}], authScheme=CHALLENGE_RESPONSE, keyStorePath=/etc/floodlight/auth/credentials.jceks, keyStorePassword is unset]
12:58:31.206 INFO [o.s.s.i.r.RPCService:main] Listening for internal Floodlight RPC on localhost/127.0.0.1:6642
12:58:31.219 INFO [n.f.c.i.OFSwitchManager:main] Listening for switch connections on 0.0.0.0/0.0.0.0:6653
12:58:31.222 INFO [n.f.l.i.LinkDiscoveryManager:main] Setting autoportfast feature to OFF
12:58:33.220 INFO [n.f.j.JythonServer:debugserver-main] Starting DebugServer on :6655

```

5. 查看 Floodlight 进程

```
# ps -ef |grep floodlight
```

```
root@gaopeng:/# ps -ef | grep floodlight
root      18831  1207   3  12:58 pts/0    00:00:04 java -jar target/floodlight.jar
root      18991 18937   0  13:00 pts/1    00:00:00 grep --color=auto floodlight
root@gaopeng:/#
```

6. 运行 Floodlight 后可以通过登录 <http://localhost:8080/ui/index.html> 使用 Floodlight 提供的 web 管理界面查看里面的信息，包括交换机、主机等等，因为我们没有接什么设备，页面没有什么信息。

<http://192.168.88.129:8080/ui/index.html>

Controller Status

Hostname: localhost:6633
 Healthy: true
 Uptime: 9693 s
 JVM memory bloat: 4599320 free out of 79273984

Modules loaded:

n.f.debugcounter.DebugCounterServiceImpl, n.f.testmodule.TestModule, n.f.ui.web.StaticWebRouteable, n.f.virtualnetwork.VirtualNetworkFilter, n.f.devicemanager.internal.DeviceManagerImpl, n.f.core.internal.OFSwitchManager, n.f.linkdiscovery.internal.LinkDiscoveryManager, n.f.loadbalancer.LoadBalancer, n.f.topology.TopologyManager, n.f.forwarding.Forwarding, n.f.flowcache.FlowReconcileManager, n.f.devicemanager.internal.DefaultEntityClassifier, n.f.storage.memory.MemoryStorageSource, n.f.jython.JythonDebugInterface, n.f.restserver.RestApiServer, org.sdnplatform.sync.internal.SyncManager, n.f.hub.Hub, n.f.firewall.Firewall, n.f.perfmon.PktInProcessingTime, n.f.core.internal.ShutdownServiceImpl, org.sdnplatform.sync.internal.SyncTorture, n.f.threadpool.ThreadPool, n.f.staticflowentry.StaticFlowEntryPusher, n.f.core.internal.FloodlightProvider, n.f.debugevent.DebugEventService,

Switches (0)

DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
------	------------	--------	---------	-------	-------	-----------------

Hosts (0)

MAC Address	IP Address	Switch Port	Last Seen
-------------	------------	-------------	-----------

7. 配置 Floodlight

Floodlight 提供了两个配置文件：floodlightdefault.properties 和 learningswitch.properties，路径都位于 floodlight/src/main/resources/。通过这两个配置文件可以查看 Floodlight 已经加载的子模块以及控制器的侦听端口、web 端口，默认的侦听端口是 6633，web 端口是 8080。如有必要用户可以自定义加载子模块和修改侦听端口，通过 vi 命令修改配置文件后，在 Floodlight 目录下执行 ant 编译后重启 Floodlight 即可。

8. 常用 Floodlight 命令

Floodlight 将自己的 API 通过 Rest Api 的形式向外暴露，关于 Rest Api，就是将程序的 API 封装成为通用的 http GET/PUT 的形式，这样的话无需关注程序

实现细节,通过发送 http 请求即可完成 API 操作。所以用户可以通过 Floodlight 的 Restful api 来向 Floodlight 请求各种信息,包括交换机状态、能力、拓扑等。需要注意,通过 Floodlight 的 Restful api 返回的信息以 json 格式封装。REST Api 信息用户访问 <http://www.projectfloodlight.org/floodlight/>来获得。

(1) 请求该控制器上所有的 switch 的 DPID,可以参考如下命令,其中 localhost:8080 要换成控制器的 ip 和 port。

```
# curl http://localhost:8080/wm/core/controller/switches/json  
  
root@gaopeng:/# curl http://localhost:8080/wm/core/controller/switches/json  
[{"inetAddress":"/192.168.88.130:60919","connectedSince":1425090522627,"switchDPID":"00:00:2050","switchDPID":"00:00:00:0c:29:51:08:4c"}]root@gaopeng:/#
```

(2) 加入流表项 switch 的 Mac 地址为 00:00:00:0c:29:51:08:4c

```
# curl -d '{"switch":"00:00:00:0c:29:51:08:4c", "name":"Open_vSwitch", "cookie":"0", "priority":"32768","ingress-port":"1", "active":"true","actions":"output=2"}' http://localhost:8080/wm/staticflowentrypusher/json
```

(3) 读取流表项

```
# curl http://localhost:8080/wm/staticflowentrypusher/list/all/json
```

(4) 删除流表项

```
# curl -X DELETE -d '{"name":"Open_vSwitch"}' http://localhost:8080/wm/staticflowentrypusher/json
```

假如要删除所有的流表项,使用如下命令:

```
# curl http://localhost:8080/wm/staticflowentrypusher /clear/<dpid>/json
```


五、Open vSwitch 基本操作

1. ovs 连接 Floodlight 控制器

首先添加一个名为 br0 的网桥。

```
ovs-vsctl add-br br0
```

2. 列出所有网桥。

```
ovs-vsctl list-br
```

3. 启用该网桥

```
ifconfig br0 up
```



```
root@gaopeng: /
br0
root@gaopeng:/# ifconfig br0 up
root@gaopeng:/# ifconfig
br0      Link encap:以太网  硬件地址 ee:aa:16:e8:7d:4e
         inet6 地址: fe80::ecaa:16ff:fee8:7d4e/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
         接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:14  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:0
         接收字节:0 (0.0 B)  发送字节:2952 (2.9 KB)

eth0     Link encap:以太网  硬件地址 00:0c:29:67:04:cb
         inet 地址:192.168.88.128  广播:192.168.88.255  掩码:255.255.255.0
         inet6 地址: fe80::20c:29ff:fe67:4cb/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
         接收数据包:25289  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:14018  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:1000
         接收字节:34104871 (34.1 MB)  发送字节:908521 (908.5 KB)
         中断:19  基地址:0x2000

lo       Link encap:本地环回
         inet 地址:127.0.0.1  掩码:255.0.0.0
         inet6 地址: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  跃点数:1
         接收数据包:550  错误:0  丢弃:0  过载:0  帧数:0
         发送数据包:550  错误:0  丢弃:0  过载:0  载波:0
         碰撞:0  发送队列长度:0
         接收字节:47607 (47.6 KB)  发送字节:47607 (47.6 KB)

root@gaopeng:/#
```

4. 我们可以用下面的命令将 br0 网桥连接至一个指定的控制器。

```
ovs-vsctl set-controller br0 tcp:<controller IP>:<port>
```

这里我们使用 Floodlight 的虚拟机 IP 地址，默认端口号为 6633，由于虚拟机的问题，在我试验中 6633 端口被占用，我使用的是 6653 端口

```
ovs-vsctl set-controller br0 tcp:192.168.88.133:6653
```


5. 查看 OVS 是否连接上控制器。

```
# ovs-vsctl show
```

如下图所示，我们可以看到，ovs 已经连接至 Floodlight 控制器了

```
root@gaopeng:/openvswitch-2.3.0# ovs-vsctl show
2575398d-5834-4b45-9ca9-7e791d62ab46
    Bridge "br0"
        Controller "tcp:192.168.88.133:6653"
            is_connected: true
        Port "br0"
            Interface "br0"
                type: internal
```

6. 此时我们再打开 Floodlight 的 Web 界面能看到控制器、已连接至该控制器的交换机以及主机的信息。Floodlight 控制器 Web 界面的 url 为：

<http://192.168.88.133:8080/ui/index.html>

如下图所示，我们的 ovs 已经连接到了 Floodlight 控制器上

The screenshot shows the Floodlight Web UI at the URL <http://192.168.88.133:8080/ui/index.html>. The interface includes a navigation bar with links for Dashboard, Topology, Switches, and Hosts. The main content area displays the 'Controller Status' section, which includes fields for Hostname (localhost:6633), Healthy (true), Uptime (929 s), and JVM memory bloat (21091408 free out of 83836928). Below this, the 'Modules loaded' section lists various modules. A red box highlights the 'Switches (1)' section, which contains a table with one entry for a switch with DPID 00:00:7e:a0:d8:bf:de:4d, IP Address 192.168.88.130:60917, Vendor Nicira, Inc., and connected since 2015/2/27 上午10:13:18. Below the switches section, the 'Hosts (0)' section is visible, showing a table with columns for MAC Address, IP Address, Switch Port, and Last Seen.

DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
00:00:7e:a0:d8:bf:de:4d	/192.168.88.130:60917	Nicira, Inc.	0	0	254	2015/2/27 上午10:13:18

MAC Address	IP Address	Switch Port	Last Seen
-------------	------------	-------------	-----------

7. 交换机的相关操作

(1) 查看虚拟交换机的信息。

```
# ovs-ofctl show br0
```

```

root@gaopeng:/openvswitch-2.3.0# ovs-ofctl show br0
OFPT_FEATURES_REPLY (xid=0x2): dpid:00007ea0d8bfde4d
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC
LOCAL(br0): addr:7e:a0:d8:bf:de:4d
    config:      0
    state:      0
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0

```

(2) 查看 br0 上各交换机端口的状态。

```
# ovs-ofctl dump-ports br0
```

```

root@gaopeng:/openvswitch-2.3.0# ovs-ofctl dump-ports br0
OFPT_PORT reply (xid=0x2): 1 ports
    port LOCAL: rx pkts=8, bytes=648, drop=0, errs=0, frame=0, over=0, crc=0
                tx pkts=0, bytes=0, drop=0, errs=0, coll=0
root@gaopeng:/openvswitch-2.3.0#

```

(3) 查看 br0 上所有的流规则

```
# ovs-ofctl dump-flows br0
```

```

root@gaopeng:/openvswitch-2.3.0# ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
    cookie=0x0, duration=10387.998s, table=0, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.998s, table=1, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.997s, table=2, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.997s, table=3, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.997s, table=4, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.997s, table=5, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.997s, table=6, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535
    cookie=0x0, duration=10387.997s, table=7, n_packets=0, n_bytes=0, idle_age=10387, priority=0 actions=CONTROLLER:65535

```

8. 网桥管理

(1) 之前已经添加了一个名为 br0 的网桥。

现在可以将网络接口 eth1 挂接到网桥 br0 上。(这里要注意, 如果在虚拟机中将 eth0 挂接到网桥 br0 上虚拟机会断网)

```
# ovs-vsctl add-port br0 eth1
```

(2) 列出挂接到网桥 br0 上的所有网络接口。

```
# ovs-vsctl list-ports br0
```

```

root@gaopeng:/# ovs-vsctl list-ports br0
eth1

```

Eth1 网卡详细信息如下, 请注意观察 Mac 地址

```

eth1  Link encap:以太网 硬件地址 00:0c:29:51:08:4c
      inet 地址:192.168.88.139 广播:192.168.88.255 掩码:255.255.255.0
      inet6 地址: fe80::20c:29ff:fe51:84c/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
      接收数据包:215527746 错误:52 丢弃:0 过载:0 帧数:0
      发送数据包:155309244 错误:0 丢弃:0 过载:0 载波:0
      碰撞:0 发送队列长度:1000
      接收字节:1753472676 (1.7 GB) 发送字节:1944262727 (1.9 GB)
      中断:16 基本地址:0x2080

```

在 Floodlight 控制器页面显示到的主机 Mac 与 Eth1 的 Mac 相同，至此，挂在本台虚拟 ovs 交换机下的主机就显示出来了

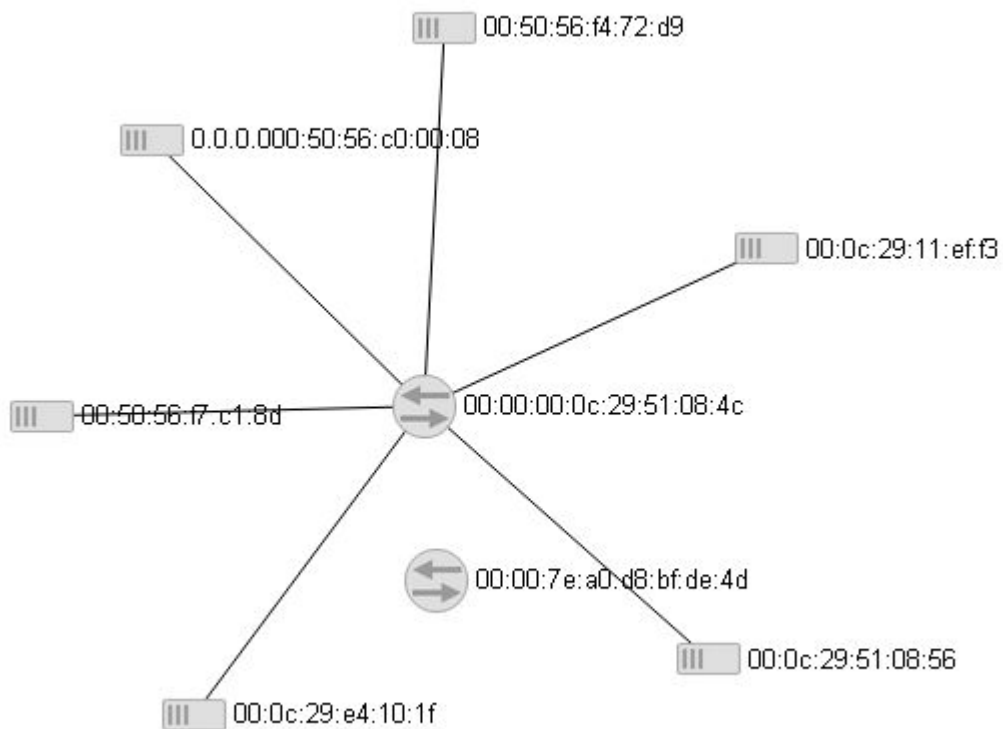
Switches (2)

DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
00:00:7e:a0:d8:bf:de:4d	/192.168.88.130:60919	Nicira, Inc.	0	0	254	2015/2/28 上午10:28:42
00:00:00:0c:29:51:08:4c	/192.168.88.141:54412	Nicira, Inc.	9	2402	254	2015/2/28 下午4:38:02

Hosts (6)

MAC Address	IP Address	Switch Port	Last Seen
00:0c:29:51:08:56		00:00:00:0c:29:51:08:4c-3	2015/2/28 下午4:36:23
00:50:56:c0:00:08	0.0.0.0	00:00:00:0c:29:51:08:4c-3	2015/2/28 下午4:36:23
00:0c:29:e4:10:1f		00:00:00:0c:29:51:08:4c-3 00:00:00:0c:29:51:08:4c-4	2015/2/28 下午4:36:23
00:50:56:f7:c1:8d		00:00:00:0c:29:51:08:4c-3	2015/2/28 下午4:38:10
00:50:56:f4:72:d9		00:00:00:0c:29:51:08:4c-3	2015/2/28 下午4:41:16
00:0c:29:11:ef:f3		00:00:00:0c:29:51:08:4c-3	2015/2/28 下午4:45:41

Floodlight 页面显示拓扑图如下



(3) 查看结果，如图所示：

```
# ovs-vsctl show
```

```
root@gaopeng:/# ovs-vsctl show
87f08cf3-f643-4ca3-a884-6f5eaaee7dc
  Bridge "br0"
    Controller "tcp:192.168.88.133:6653"
      is_connected: true
    Port "br0"
      Interface "br0"
        type: internal
    Port "eth1"
      Interface "eth1"
```

(4) 删除网桥 br0 上挂接的 eth0 网络接口。

```
# ovs-vsctl del-port br0 eth0
```

(5) 删除名为 br0 的网桥。

```
# ovs-vsctl del-br br0
```

9.OVS 其它常用命令及选项如下表所示：

(1) ovs-vsctl 常用命令及选项

命令及选项	含义
init	初始化数据库(前提数据包为空)
show	打印数据库信息摘要
emer-reset	将 ovs 配置复位为空状态
add-br BRIDGE	添加新的网桥 BRIDGE
del-br BRIDGE	删除网桥 BRIDGE
list-br	打印网桥摘要信息
list-ports BRIDGE	打印网桥 BRIDGE 中所有 port 摘要信息
add-port BRIDGE PORT	向网桥 BRIDGE 中添加端口 PORT
del-port [BRIDGE] PORT	删除 BRIDGE 上的端口 PORT
get-controller BRIDGE	获取网桥 BRIDGE 的控制器信息
del-controller BRIDGE	删除网桥 BRIDGE 的控制器信息
set-controller BRIDGE TARGET...	向网桥 BRIDGE 添加控制器 target
-h, --help	打印帮助信息
-V, --version	打印 ovs 的版本信息

(2) ovs-ofctl 常用命令及选项

命令及选项	含义
show SWITCH	打印 OpenFlow 信息
dump-ports SWITCH [PORT]	打印端口统计信息
dump-ports-desc SWITCH	打印端口描述信息
dump-flows SWITCH	打印 SWITCH 中所有的流表项
dump-flows SWITCH FLOW	打印 SWITCH 中匹配的流表项
add-flow SWITCH FLOW	向 SWITCH 添加流表项
add-flows SWITCH FILE	从文件中向 SWITCH 添加流表项
mod-flows SWITCH FLOW	修改 SWITCH 的流表项
del-flows SWITCH [FLOW]	删除 SWITCH 的流表项
-O, --protocols	设置允许的 OpenFlow 版本(默认 : OpenFlow10)
-h, --help	打印帮助信息
-V, --version	打印版本信息