# 基于Floodlight的SDN说明文档

目录

[基于Floodlight的SDN说明文档 1](#_Toc374011746)

[一、程序设计 2](#_Toc374011747)

[1.1 Mininet Virtual Network 2](#_Toc374011748)

[1.1.1 Mininet介绍 2](#_Toc374011749)

[1.1.2 基于Mininet拓扑搭建 3](#_Toc374011750)

[1.2 Floodlight 3](#_Toc374011751)

[1.2.1 Floodlight介绍 3](#_Toc374011752)

[1.3 Command Proxy 3](#_Toc374011753)

[1.4 Web Interface 4](#_Toc374011754)

[二、关键功能简介及截图 4](#_Toc374011755)

[2.1查看Switch列表及各个Switch的信息 4](#_Toc374011756)

[2.2查看Host列表及各个Host的信息 5](#_Toc374011757)

[2.3查看拓扑图 5](#_Toc374011758)

[2.4查看流表 6](#_Toc374011759)

[2.5针对某个Switch添加/删除Static Flow 6](#_Toc374011760)

[2.6查看日志 8](#_Toc374011761)

[三、组员及分工 9](#_Toc374011762)

[钟晓诚MF1332095,靳峥MF1332025. 9](#_Toc374011763)

[吕翔MF1332042,唐毅明MF1332057. 9](#_Toc374011764)

[孟焱MF1332044,倪卫明MF1332045 9](#_Toc374011765)

# 一、程序设计

我们基于Floodlight现有的API实现了一个简单的Web站点，主要功能有：

* 查看Switch列表及各个Switch的信息
* 查看Host列表及各个Host的信息
* 查看拓扑图
* 查看流表
* 针对某个Switch添加/删除Static Flow
* 查看日志

程序的架构设计如下图所示：

Web Interface

Command Proxy

Floodlight

Mininet Virtual Network

底层虚拟机上的操作系统上安装有Mininet，通过Mininet我们可以创建多种类型的拓扑，并且可以把floodlight controller设置为远程，在我们的方案中floodlight安装在虚拟机的宿主机上。Floodlight负责监听和控制虚拟机中拓扑的动态。我们在宿主机上又搭建了一个网站，网站后端即Command Proxy负责转发前端Web Interface的操作请求给Floodlight，Floodlight通过REST API接口开放。前端Web Interface负责发出操作请求并在页面上展示整个拓扑的状态并提供操作接口以便用户能够添加/删除flows。

## 1.1 Mininet Virtual Network

### 1.1.1 Mininet介绍

Mininet是由一些虚拟的终端节点（end-hosts）、交换机、路由器连接而成的一个网络仿真器，它采用轻量级的虚拟化技术使得系统可以和真实网络相媲美。Mininet可以很方便地创建一个支持SDN的网络：host就像真实的电脑一样工作，可以使用ssh登录，启动应用程序，程序可以向以太网端口发送数据包，数据包会被交换机、路由器接收并处理。有了这个网络，就可以灵活地为网络添加新的功能并进行相关测试，然后轻松部署到真实的硬件环境中。

Mininet的特性包括可以简单、迅速地创建一个支持用户自定义的网络拓扑，缩短开发测试周期。可以运行真实的程序，在Linux上运行的程序基本上可以都可以在Mininet上运行，如WiresharkMininet。支持Openflow，在Mininet上运行的代码可以轻松移植到支持OpenFlow的硬件设备上。Mininet可以在自己的电脑，或服务器，或虚拟机，或者云（例如Amazon EC2）上运行。Mininet提供python API，简单易用。Mininet是开源项目，源代码在这里：<https://github.com/mininet>，Mininet的官方网址在这里<http://mininet.org/>。

### 1.1.2 基于Mininet拓扑搭建

拓扑搭建方面我们使用了Mininet，我们在VirtualBox安装Mininet虚拟机，这里有虚拟机的镜像下载：<http://floodlight-download.projectfloodlight.org/files/floodlight-vm.zip>。利用它可以非常方便的模拟出一个N叉树的网络，其中每一个中间节点都是一个Switch，每一个叶节点都是一个Host。例如，模拟出一棵四层的满二叉树网络的命令为：

**sudo** mn --topo tree,depth=3,fanout=2 --controller=remote --ip=xxx.xxx.xxx.xxx --port=6633

具体的Mininet操作指导在这里：<http://mininet.org/walkthrough/>。Mininet可以连接到Floodlight作为Controller，Controller可以是本地的或者远程的。默认情况下一个Mininet虚拟网络中的Swicth只能使用同一个Controller，但是可以通过编写自定义脚本的方式来构建Mininet拓扑，并且在其中显示的指定多个Controller。

## 1.2 Floodlight

### 1.2.1 Floodlight介绍

Floodlight是Openflow协议的一个实现。它提供了一些简单的API用来查询和控制网络状态，例如查询拓扑环境；罗列Switch和Host；添加/删除静态流；配置防火墙等。目前Floodlight提供的接口还不完善，还有很多工作可以做。Floodlight的官方网站在这里：<http://www.projectfloodlight.org/floodlight/>。Floodlight也是一个开放源代码项目，源码在这里：<https://github.com/floodlight/floodlight>。Floodlight的安装指导在这里：<http://www.openflowhub.org/display/floodlightcontroller/Installation+Guide>。Floodlight提供了一些REST API，我们可以通过这些API来获取整个拓扑的一些状态，具体的API列表在这里：<http://www.openflowhub.org/display/floodlightcontroller/REST+API>。

## 1.3 Command Proxy

由于ajax不能跨域请求，所以我们建立了一个服务器端代理。Command Proxy是我们用Php编写的一个后端脚本，它用来转发Web Interface的请求或者执行Web Interface发送的命令。流程如下图所示：

Web Interface

Request

Floodlight

Command Proxy

**Command**

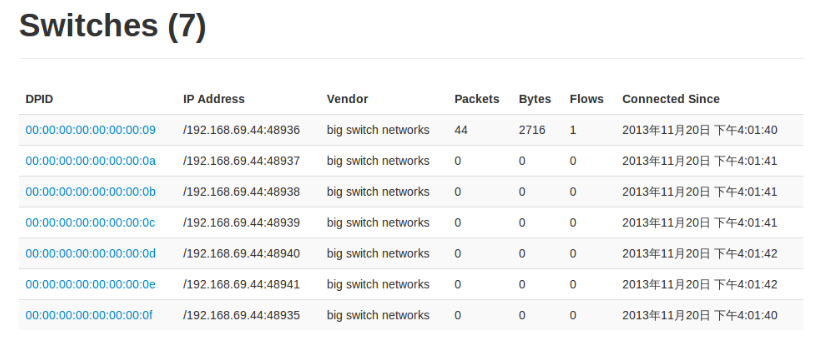
Shell

## 1.4 Web Interface

Web Interface提供了可视化的用户接口，用户可以在浏览器中查看拓扑、添加/删除流、查看日志等操作，并负责发出操作请求到后端commend proxy。

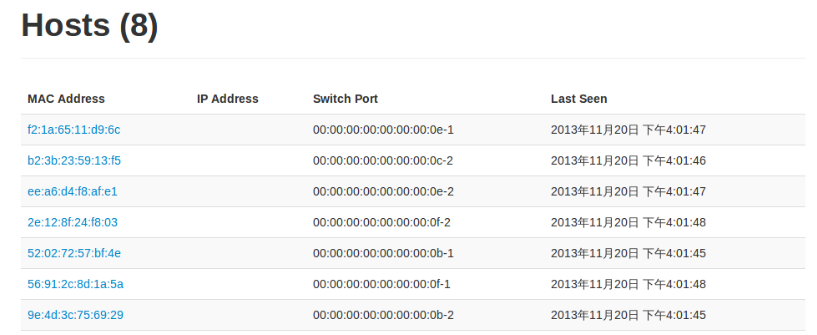
# 二、关键功能简介及截图

## 2.1查看Switch列表及各个Switch的信息



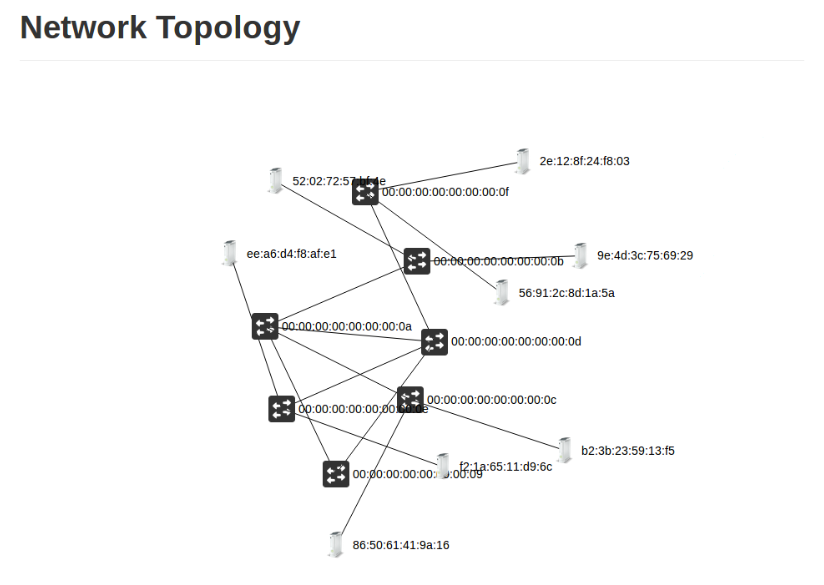
我们首先为用户提供了可视化界面来查看路由的列表，所有路由的IP Adress、Vender、Packets、Bytes、Flows以及Connected Since信息一览无遗。

## 2.2查看Host列表及各个Host的信息



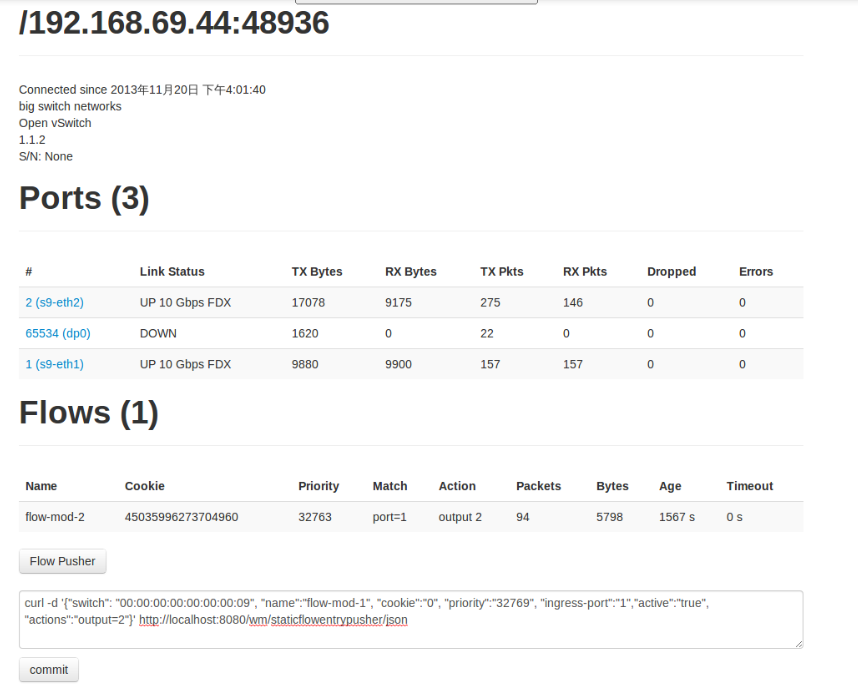
同时我们继续利用floodlight的自带API让各主机的信息也完美地展现在了用户面前。

## 2.3查看拓扑图



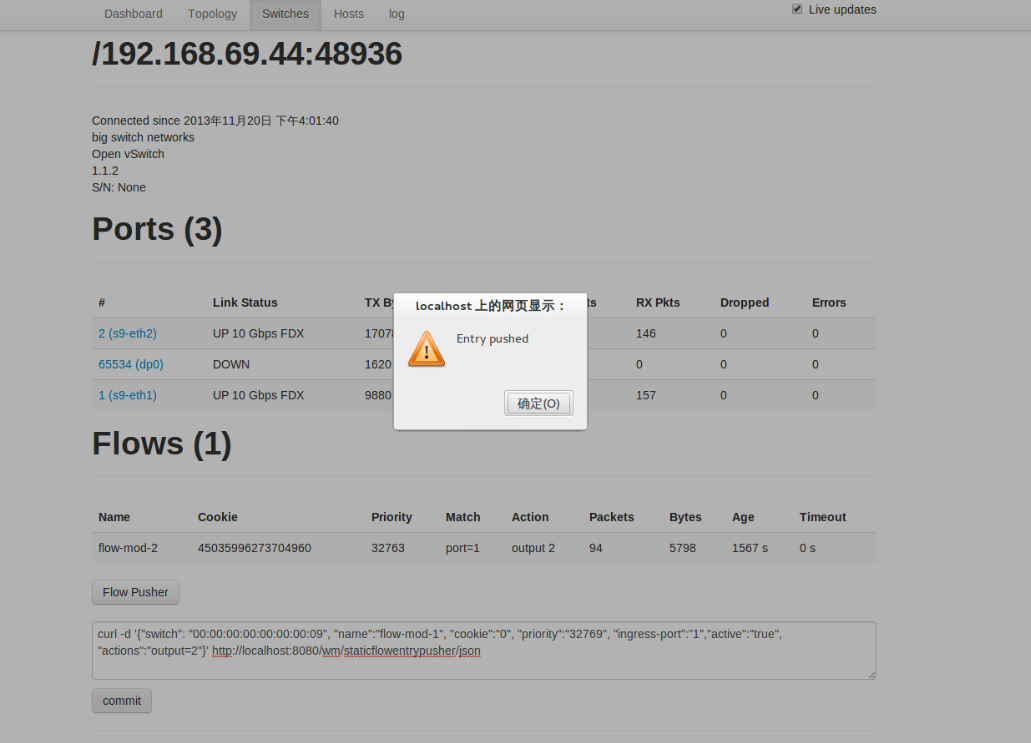
接着我们实现了查看拓扑结构的功能，用户还可以自由拖动图中的设备以达到最好的视觉化效果。

## 2.4查看流表



我们还提供了查看流表的功能。用户可以看到流的名称、Cookie、优先级、数据包数量等信息。

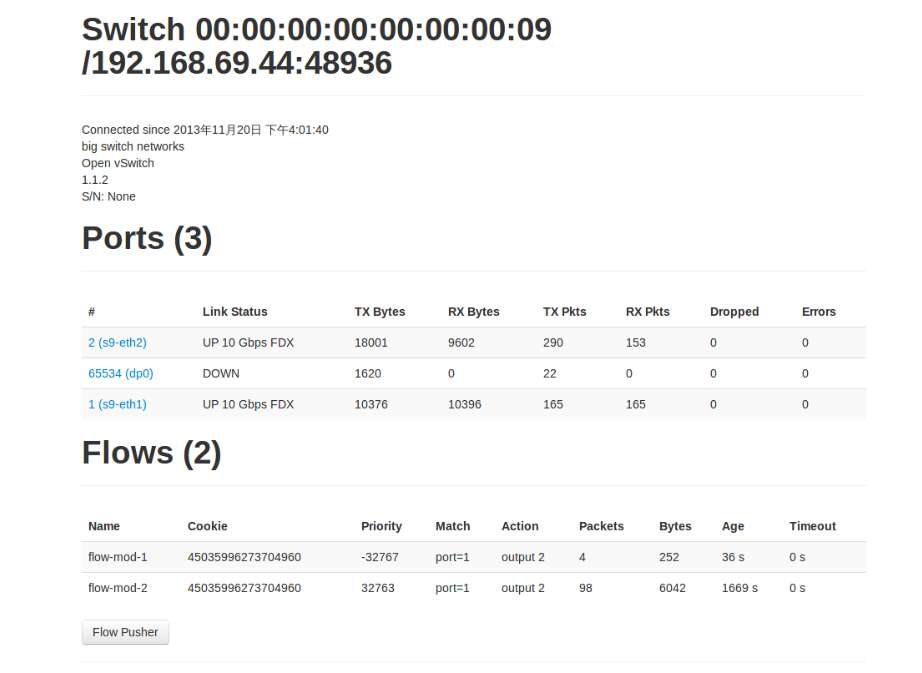
## 2.5针对某个Switch添加/删除Static Flow



我们还给用户提供了增加和删除流表的功能。在流表的下方我们为用户提供了命令输入窗口，在这里用户可以自主输入添加流表命令，比如示例命令如下：

curl -d '{"switch": "00:00:00:00:00:00:00:09", "name":"flow-mod-1", "cookie":"0", "priority":"32768", "ingress-port":"1","active":"true", "actions":"output=2"}' <http://localhost:8080/wm/staticflowentrypusher/json>

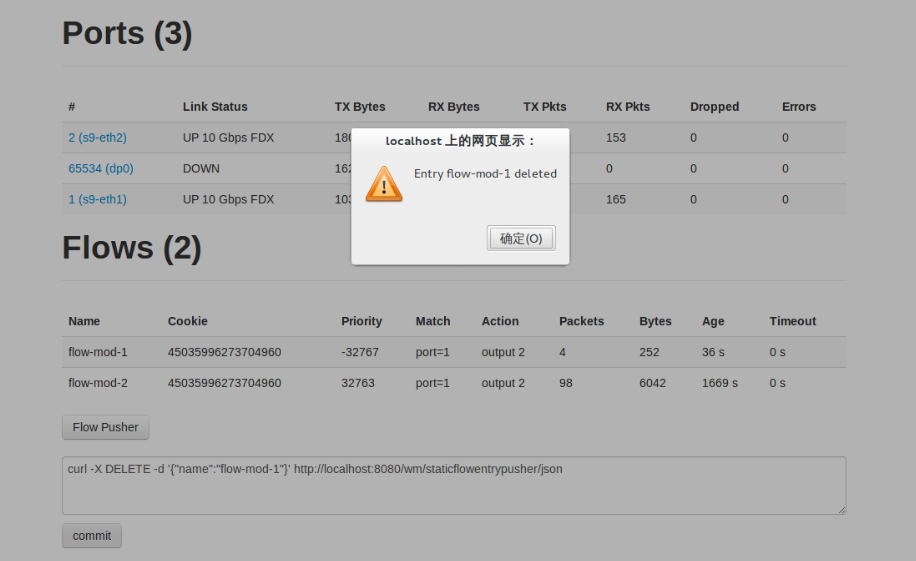
来添加一条流表（具体命令格式详见floodlight REST API说明）。上图是在添加流表，下图则是添加流表成功的样子。我们可以明显看到流表中多了一条记录。



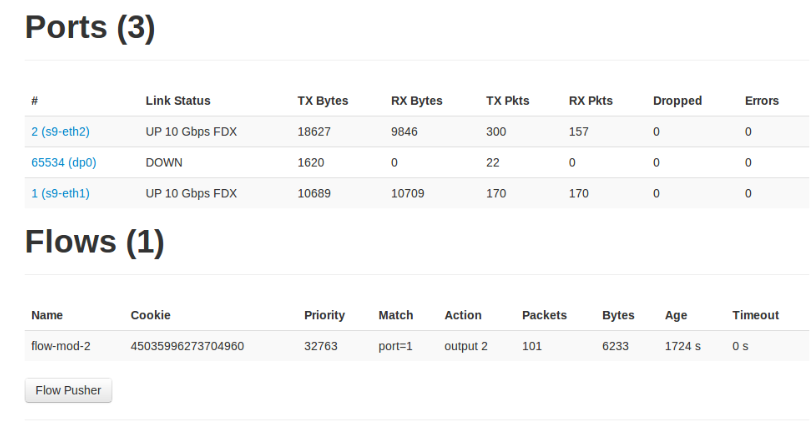
删除和添加基本上一样，区别仅在于命令不同。这是删除上面添加的流表的示例命令：

curl -X DELETE -d '{"name":"flow-mod-1"}' <http://localhost:8080/wm/staticflowentrypusher/json>

（具体命令格式详见floodlight REST API说明）删除的动作如下。

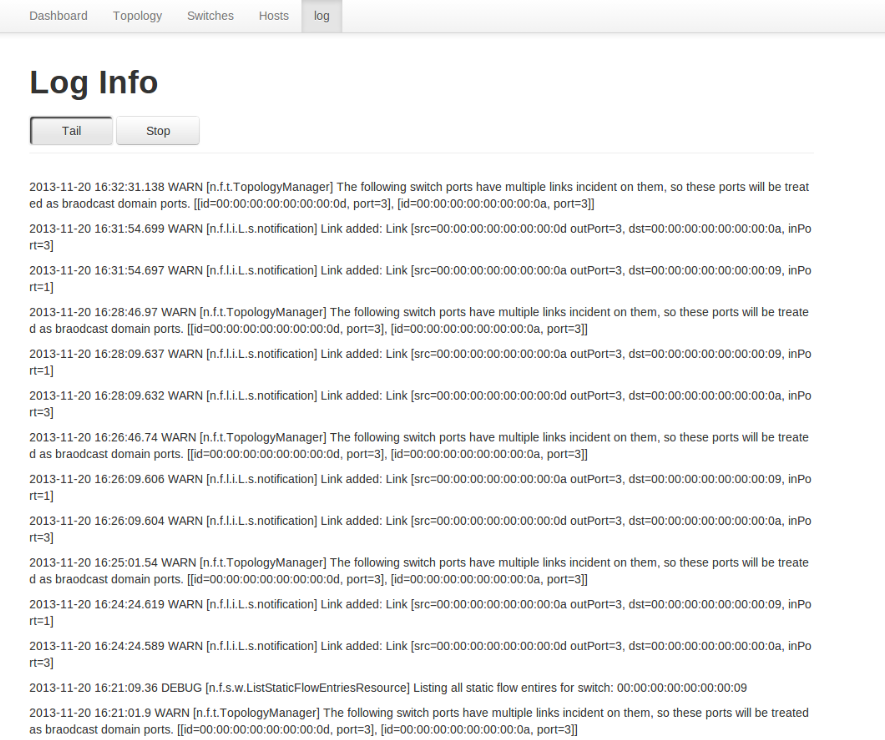


删除完成，点击确认后，哗，流表少了一条~~~



## 2.6查看日志

最后我们还完成了查看日志的功能。整个拓扑网络的日志监听完整地在这里反映出来了，并能够根据日志实时更新页面。页面的样子如下：



# 三、组员及分工

## 钟晓诚MF1332095,靳峥MF1332025:Static Flow Pusher API实现.

## 吕翔MF1332042,唐毅明MF1332057：编写测试用的发包脚本,项目文档编写.

## 孟焱MF1332044,倪卫明MF1332045：拓扑设计与环境搭建,Web页面UI设计,日志功能实现