# 基于Floodlight的SDN说明文档

目录

[基于Floodlight的SDN说明文档 1](#_Toc372739916)

[一、程序设计 2](#_Toc372739917)

[1.1 Mininet Virtual Network 2](#_Toc372739918)

[1.2 Floodlight 2](#_Toc372739919)

[1.3 Command Proxy 3](#_Toc372739920)

[1.4 Web Interface 3](#_Toc372739921)

[二、关键功能简介及截图 4](#_Toc372739922)

[2.1查看Switch列表及各个Switch的信息 4](#_Toc372739923)

[2.2查看Host列表及各个Host的信息 4](#_Toc372739924)

[2.3查看拓扑图 5](#_Toc372739925)

[2.4查看流表 5](#_Toc372739926)

[2.5针对某个Switch添加/删除Static Flow 6](#_Toc372739927)

[2.6查看日志 8](#_Toc372739928)

[三、组员及分工 9](#_Toc372739929)

[钟晓诚MF1332095,靳峥MF1332025. 9](#_Toc372739930)

[吕翔MF1332042,唐毅明MF1332057. 9](#_Toc372739931)

[孟焱MF1332044,倪卫明MF1332045 9](#_Toc372739932)

# 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

## 1.1 Mininet Virtual Network

拓扑搭建方面我们使用了Mininet，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

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

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。

## 1.3 Command Proxy

Command Proxy是我们用Php编写的一个脚本，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**

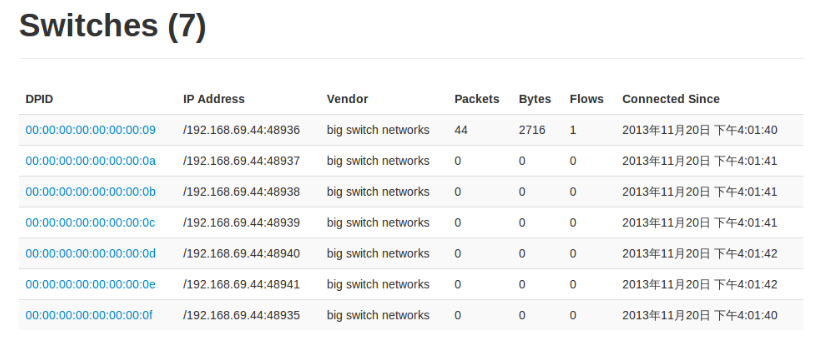
Shell

## 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。

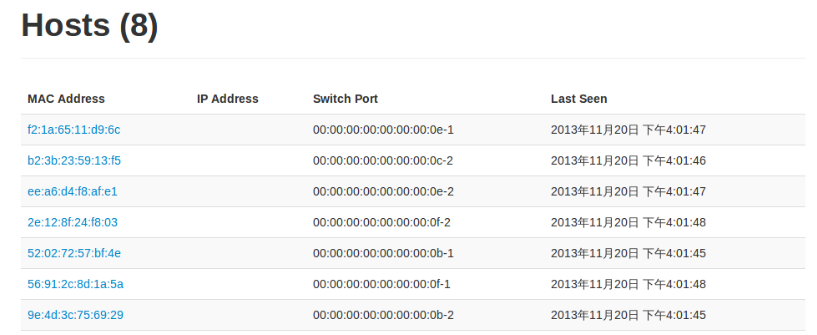
# 二、关键功能简介及截图

## 2.1查看Switch列表及各个Switch的信息



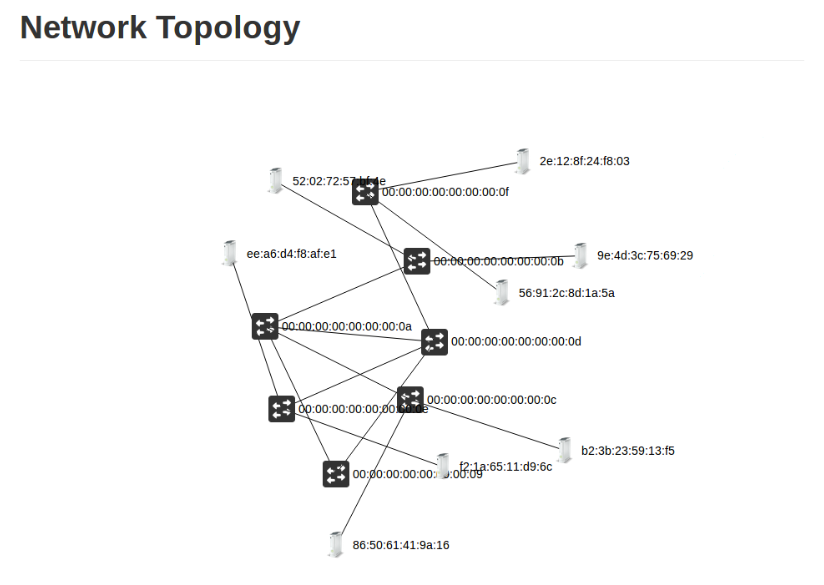
我们首先为用户提供了可视化界面来查看路由的列表，所有路由的IP Adress、Vender、Packets、Bytes、Flows以及Connected Since信息一览无遗。

## 2.2查看Host列表及各个Host的信息



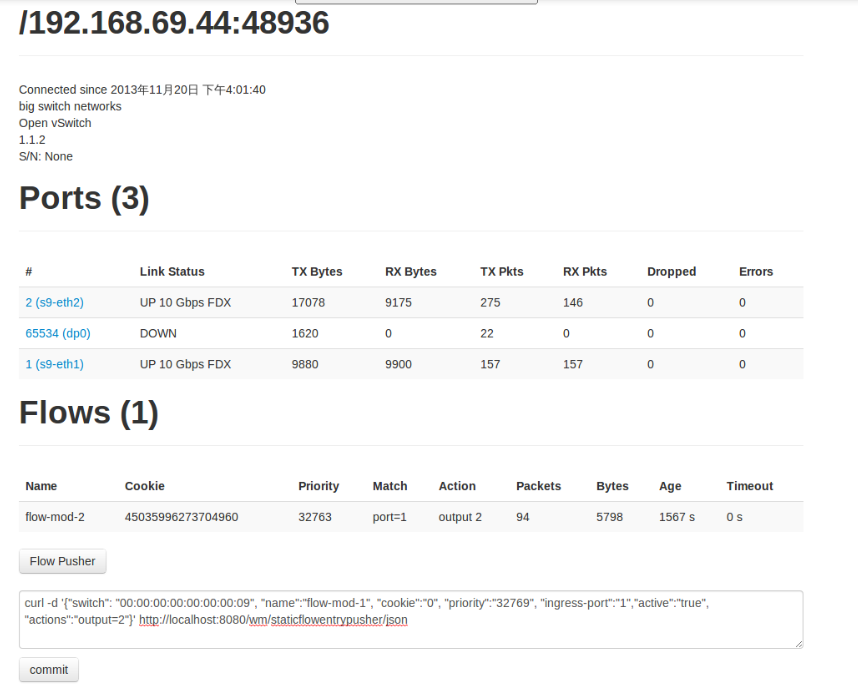
同时我们继续利用floodlight的自带API让各主机的信息也完美地展现在了用户面前。

## 2.3查看拓扑图



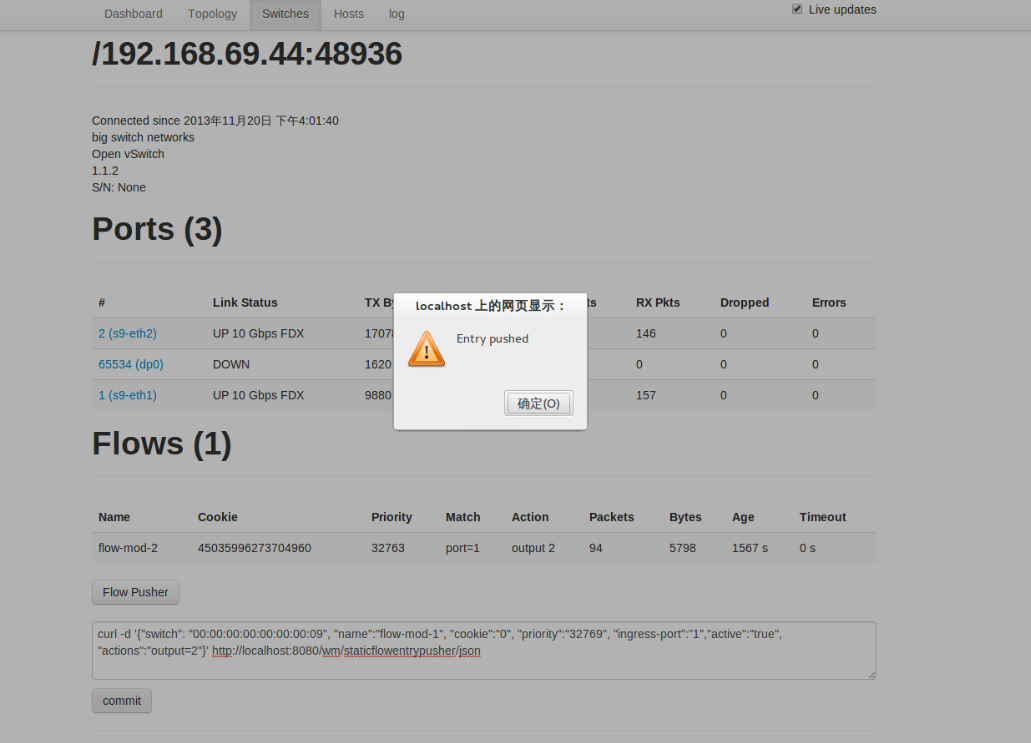
接着我们实现了查看拓扑结构的功能，用户还可以自由拖动图中的设备以达到最好的视觉化效果。

## 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。

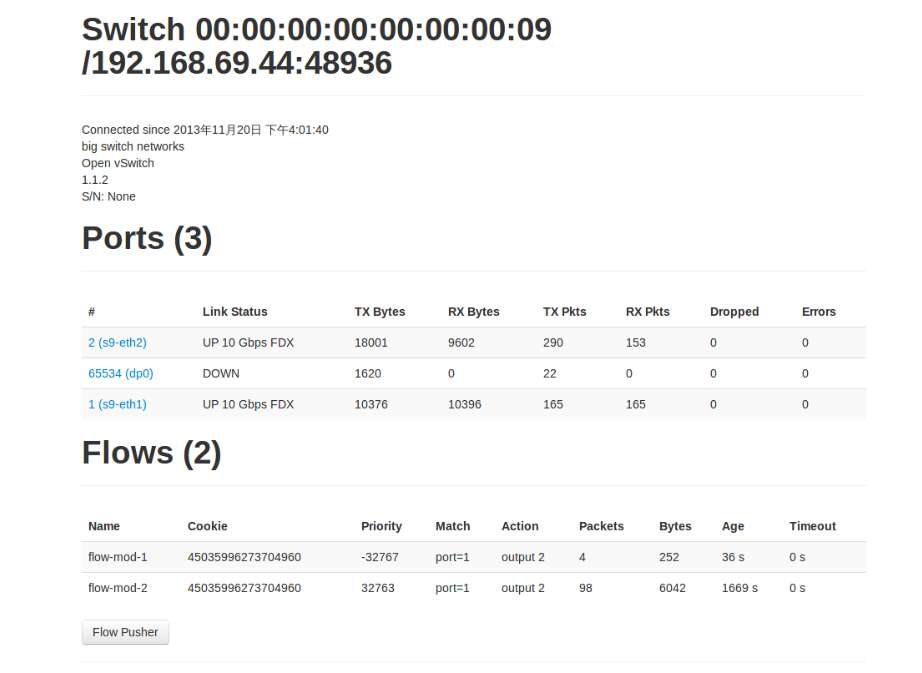
## 2.5针对某个Switch添加/删除Static Flow



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>

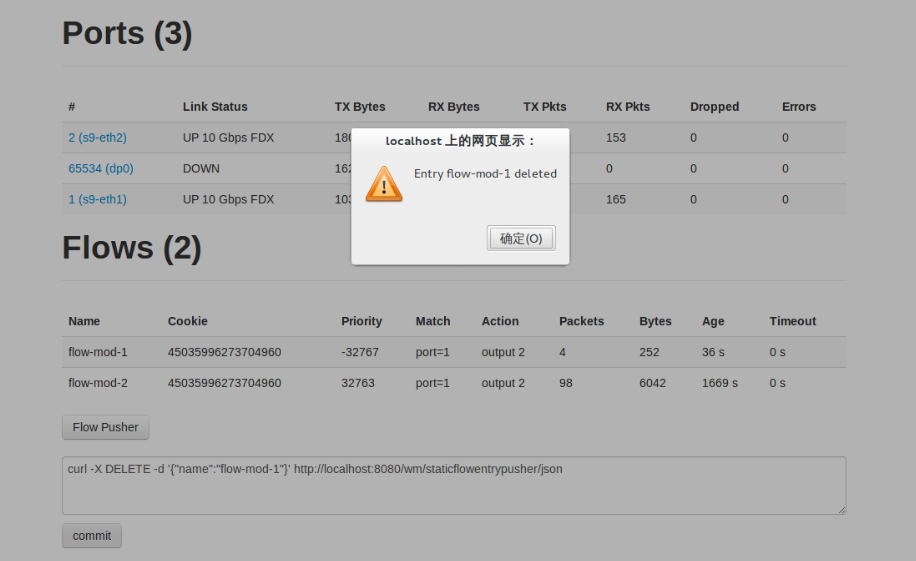
来添加一条流表（具体命令格式详见floodlight REST API说明）。上图是在添加流表，下图则是添加流表成功的样子。我们可以明显看到流表中多了一条记录。



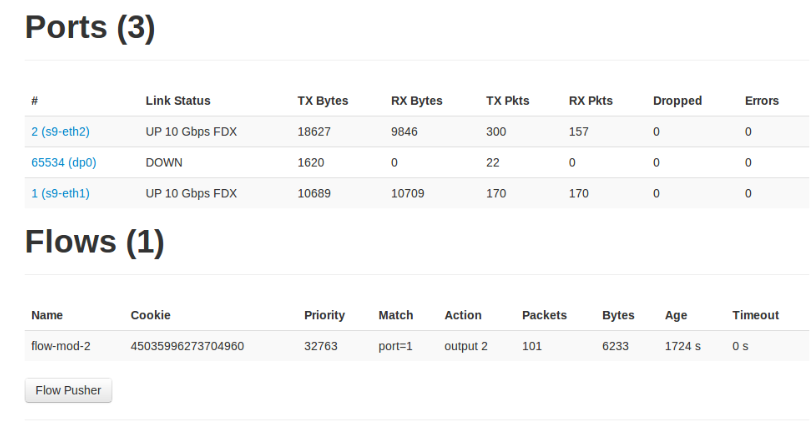
删除和添加基本上一样，区别仅在于命令不同。这是删除上面添加的流表的示例命令：

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。

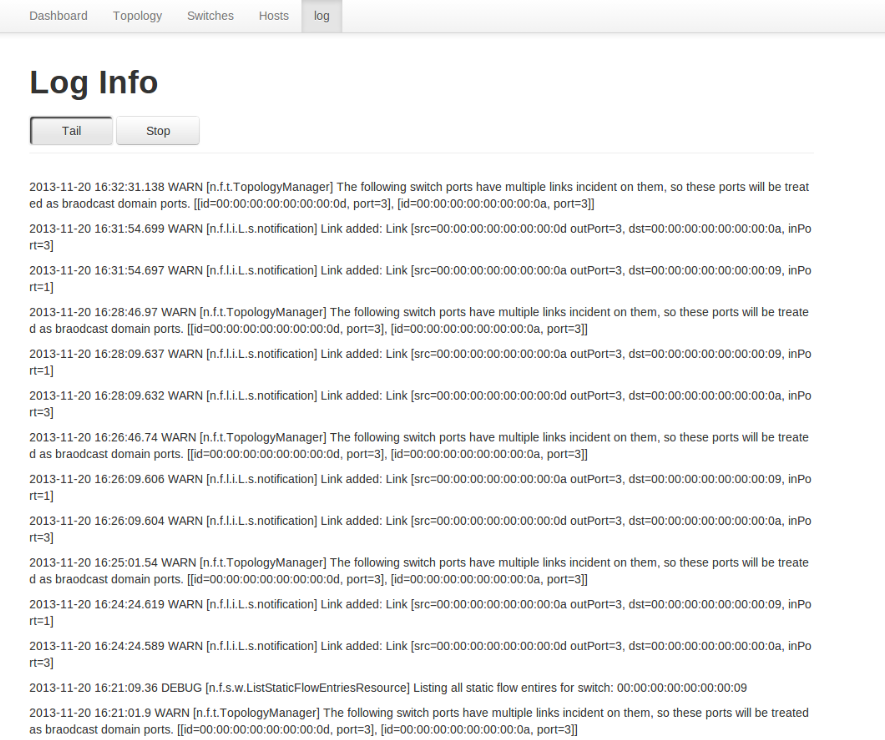


删除完成，点击确认后，哗，流表少了一条~~~



## 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