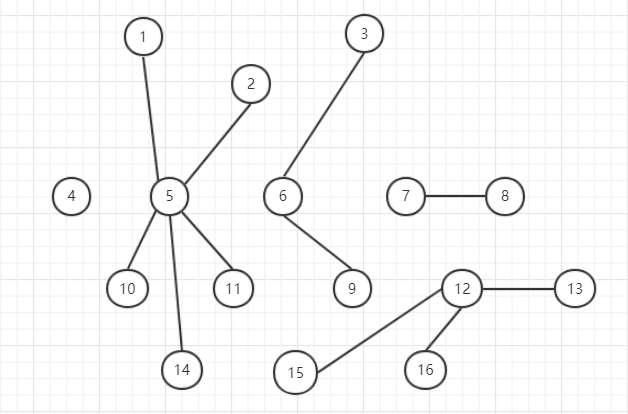
**SDN环境下的虚拟网络搭建**

# 毕设大致思路

通过Mininet工具模拟一个规模在15-20节点的网络拓扑，验证其内部连通性。初始的虚拟网络拓扑搭建完成之后，通过自己对mininet的扩展实现删除若干节点或者链路，证明被删除的节点不会影响剩余节点原来的连通性。

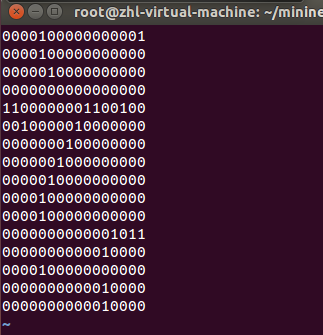
# 进度成果展示

以下图所示拓扑为例：



## 拓扑搭建

将上图所示的拓扑结构装换成矩阵形式如下：



将存放矩阵的topo.txt文件放入自定义的拓扑文件的同级目录，即/mininet/custom。

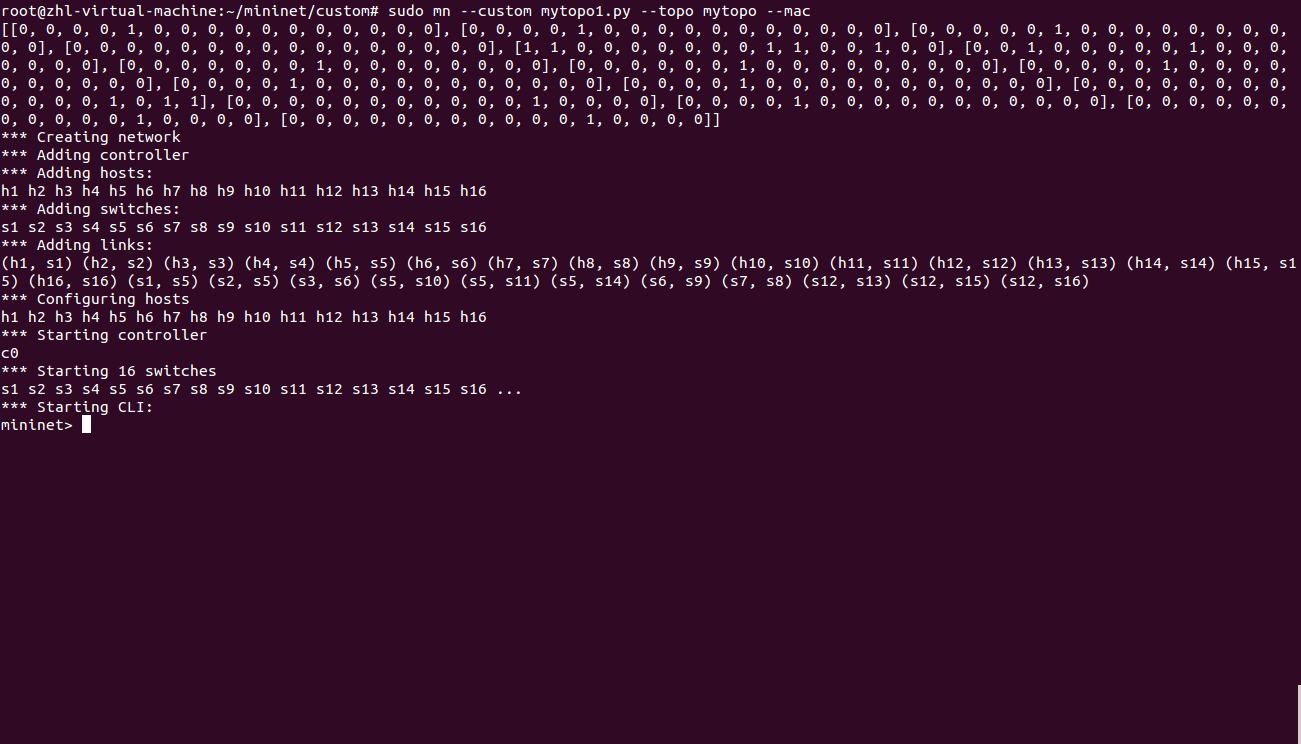
自定义拓扑的文件为mytopo1.py，内容为自己编写的生成自定义拓扑的python代码。



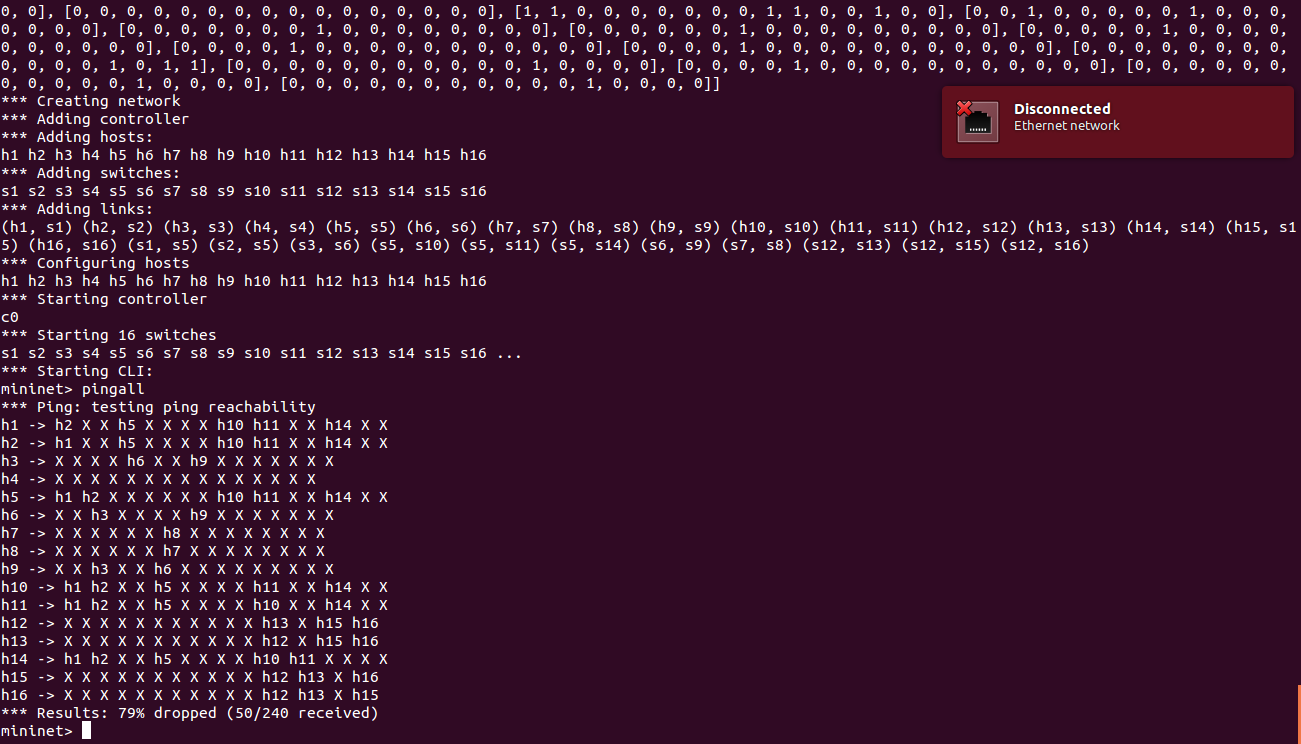
以上是自定义拓扑的核心代码，主要逻辑是读取topo.txt文件中的矩阵，根据读取的内容生成对应交换机为节点组成的虚拟网络拓扑。（为了方便进行测试，每一个switch节点都连接了一个host主机，通过host之间进行评操作反映switch之间的链路是否互通）

mytopo1.py编写完成之后便可以使用其来构建虚拟网络拓扑：

（1）.进入/mininet/custom目录，执行sudo mn --custom mytopo1.py --topo mytopo –mac;



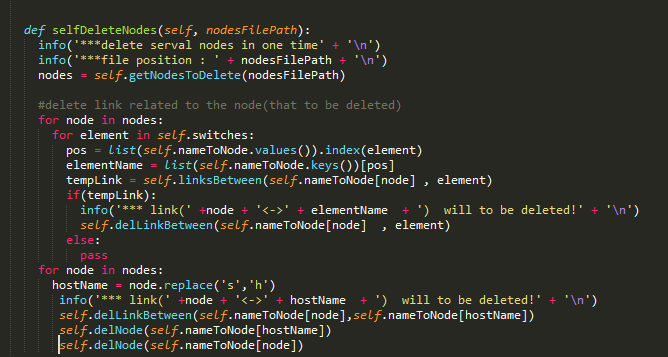
进入mininet>视图，通过pingall命令测试网络拓扑：

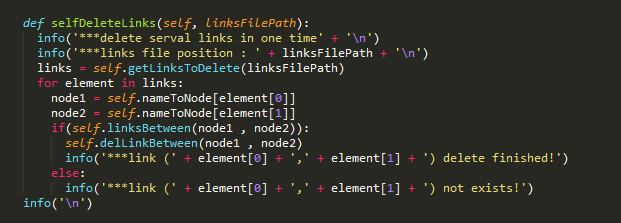


根据mininet源码，虚拟网络拓扑生成之后被一个net对象持有，可以根据对net.py的扩展来实现在mininet>视图中通过文件读取的方式删除节点和链路。

Mininet的cli中提供了py命令以支持在mininet>视图中执行python代码。

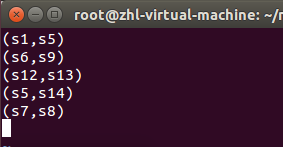
对net.py的扩展的核心内容如下：





首先进行链路删除测试：

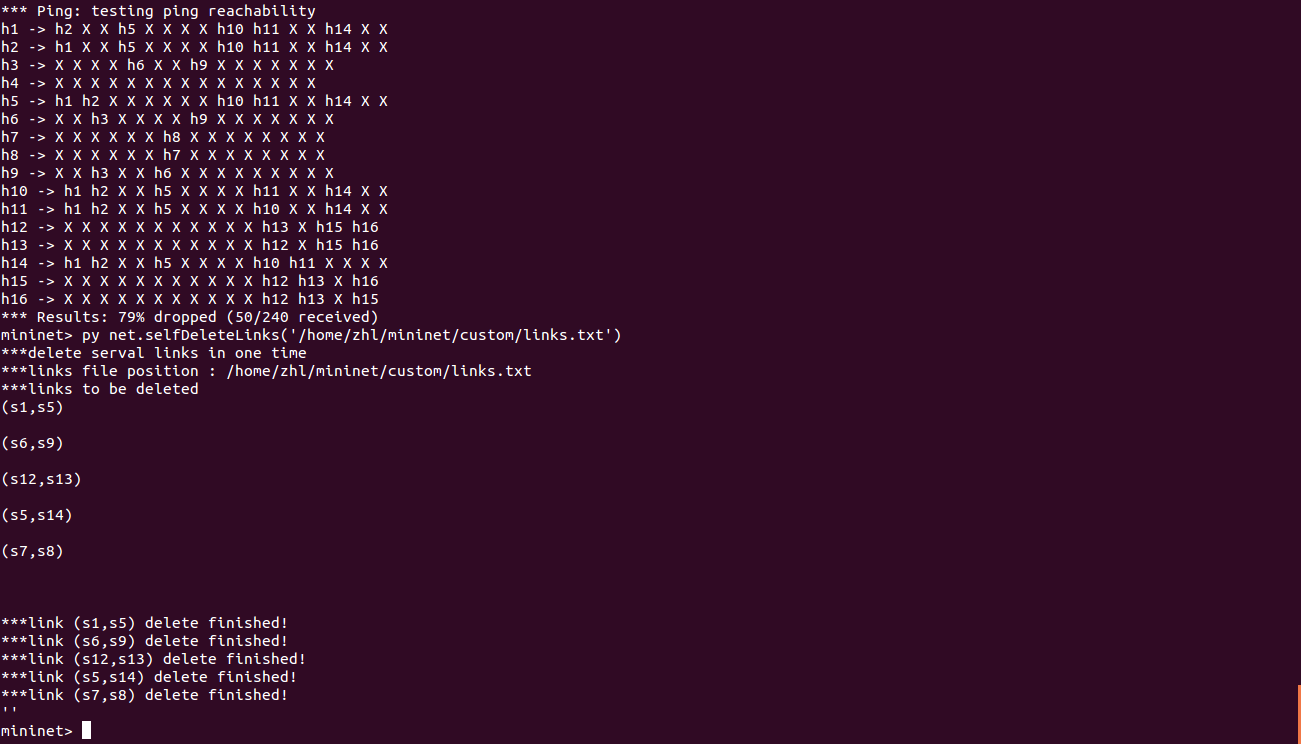
首先要按照下图所示的形式编写links.txt



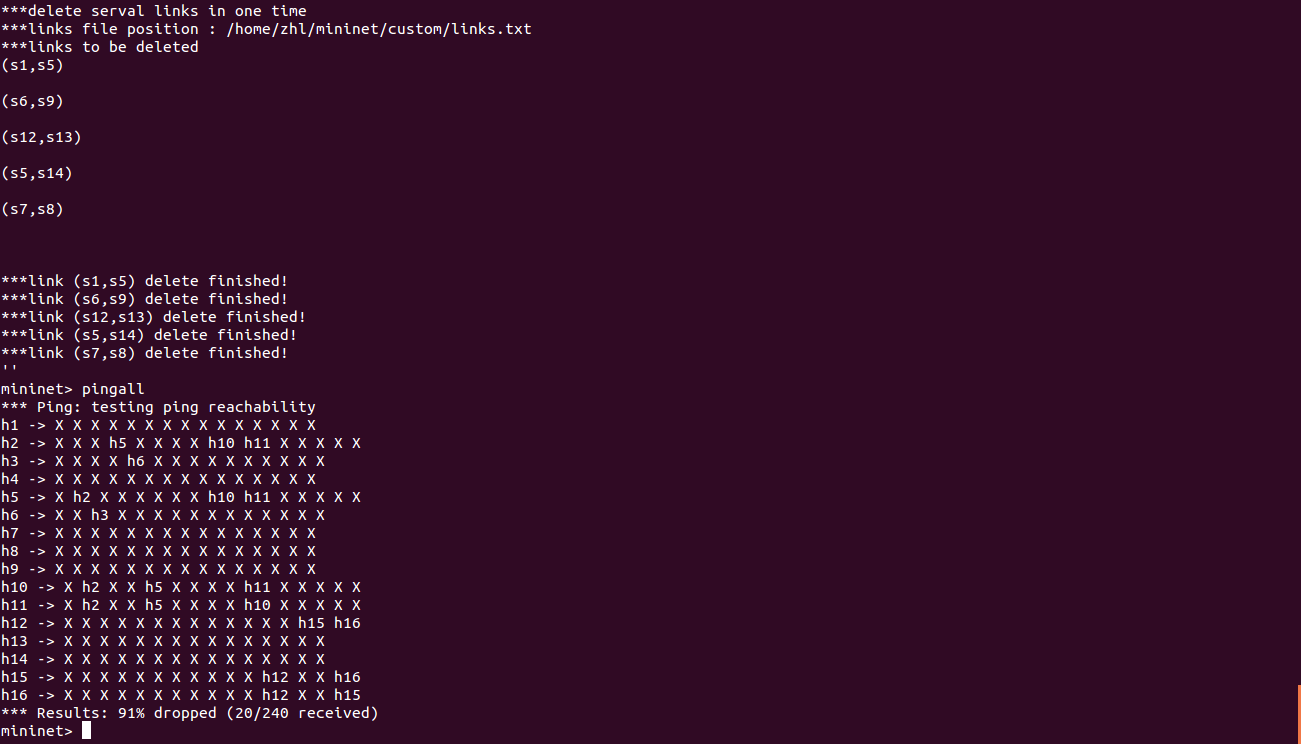
链路编写完成之后，将文本文件放入/mininet/custoim目录下，之后执行命令：

py net.selfDeleteLinks('/home/zhl/mininet/custom/links.txt')，就可以删除指定的链路。

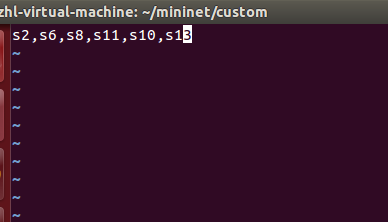
执行结果如下：



然后执行pingall测试，查看链路删除情况：



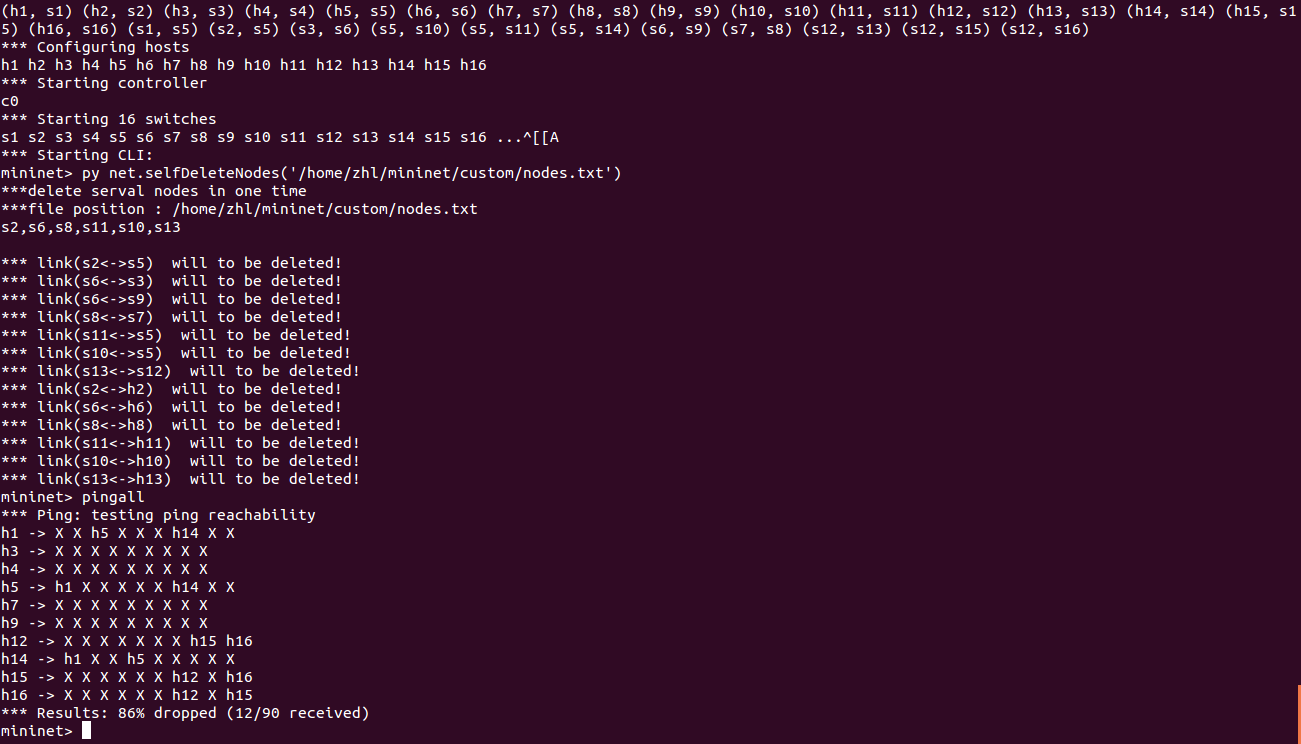
测试完链路删除，接下来是节点删除，删除节点需要将节点相关的链路也删除。同样需要按照如下格式编写nodes.txt：



链路编写完成之后，将文本文件放入/mininet/custoim目录下，之后执行命令：

py net.selfDeleteNodes('/home/zhl/mininet/custom/links.txt')，就可以删除指定的链路。

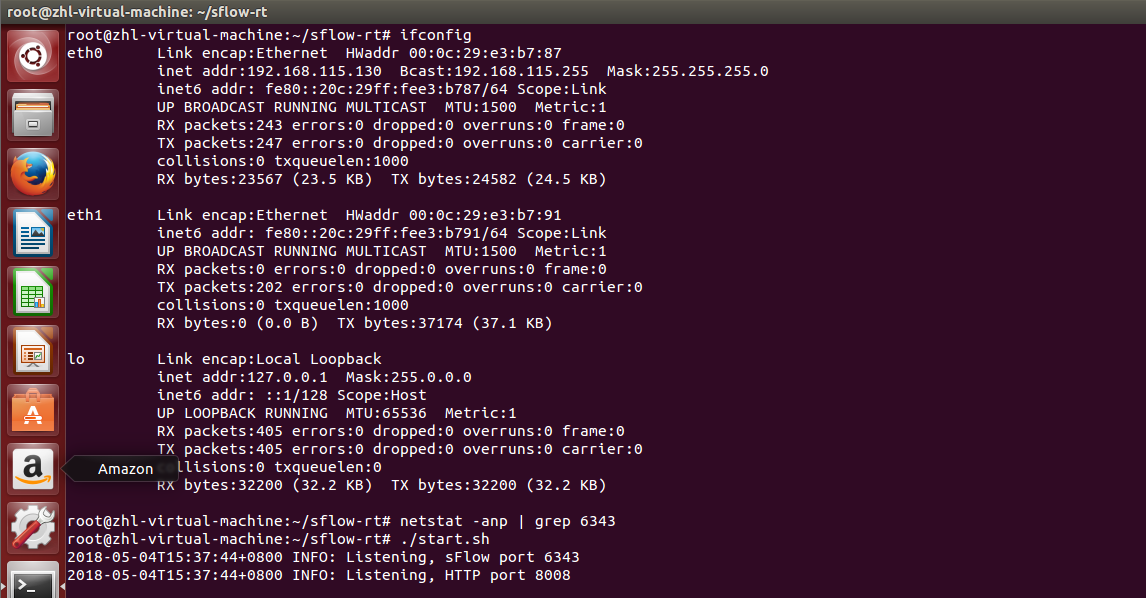
执行结果如下：



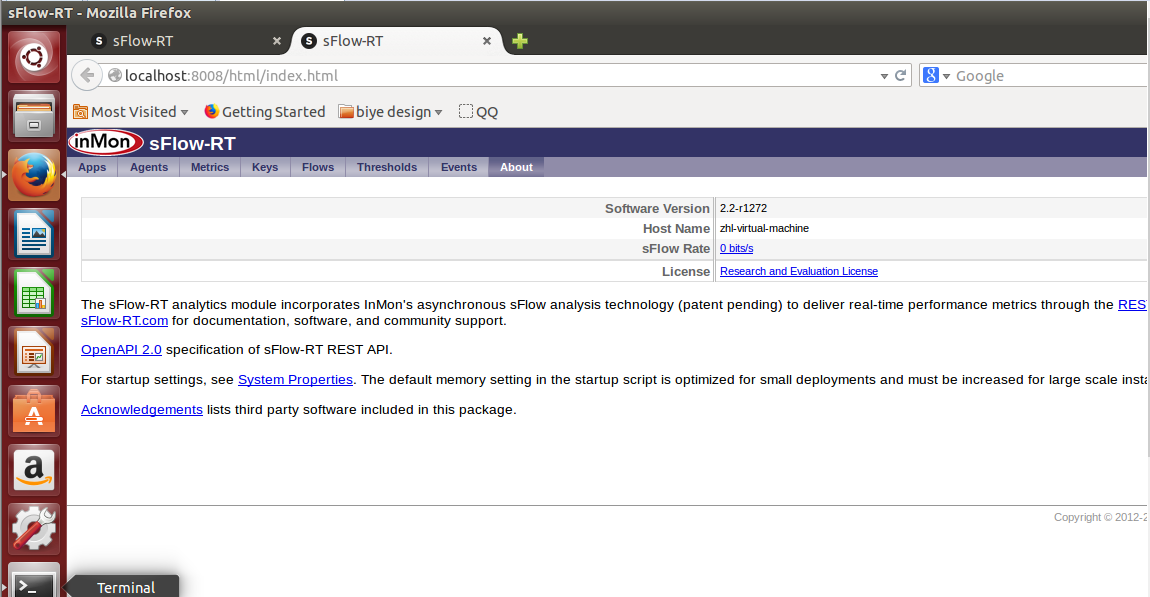
上述为根据4月25日会后完成内容。

流量统计采用sFlow，新开一台虚拟机作为sFlow Collector

虚拟机2：



然后打开浏览器，访问：<http://localhost:8008/html/index.html>

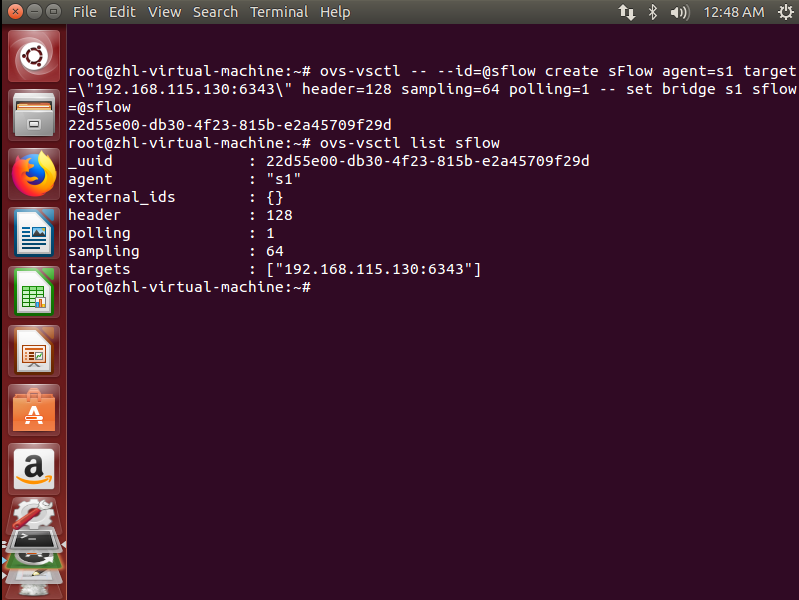


然后在搭建网络拓扑的虚拟机上新开一个ternimal终端，开启sFlow Agent：

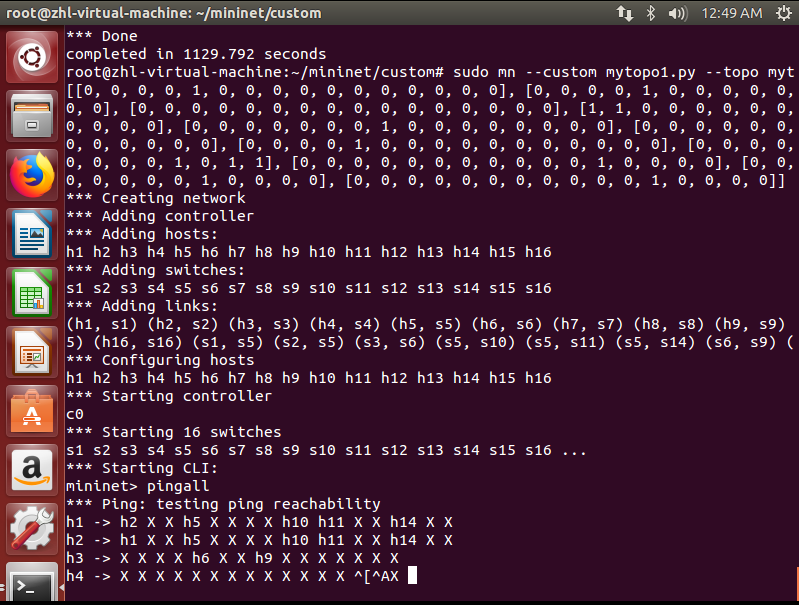
输入命令：

ovs-vsctl -- --id=@sflow create sFlow agent=s1 target=\"10.0.0.107:6343\" header=128 sampling=64 polling=1 -- set bridge s1 sflow=@sflow

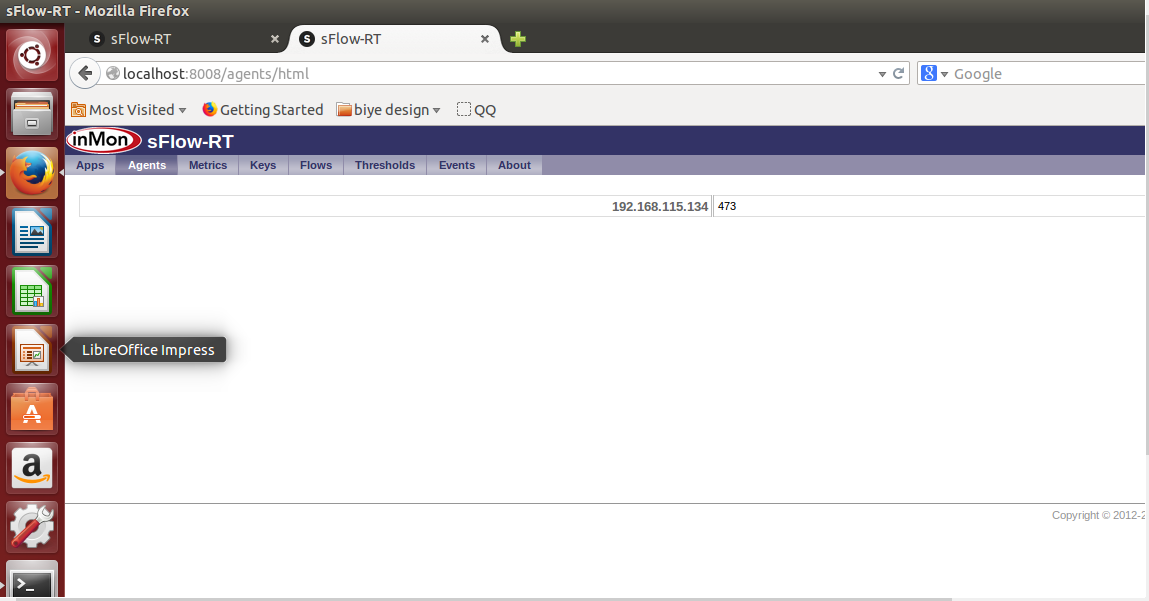
然后通过执行命令：ovs-vsctl list sflow 可见sFlow Agent开启成功。

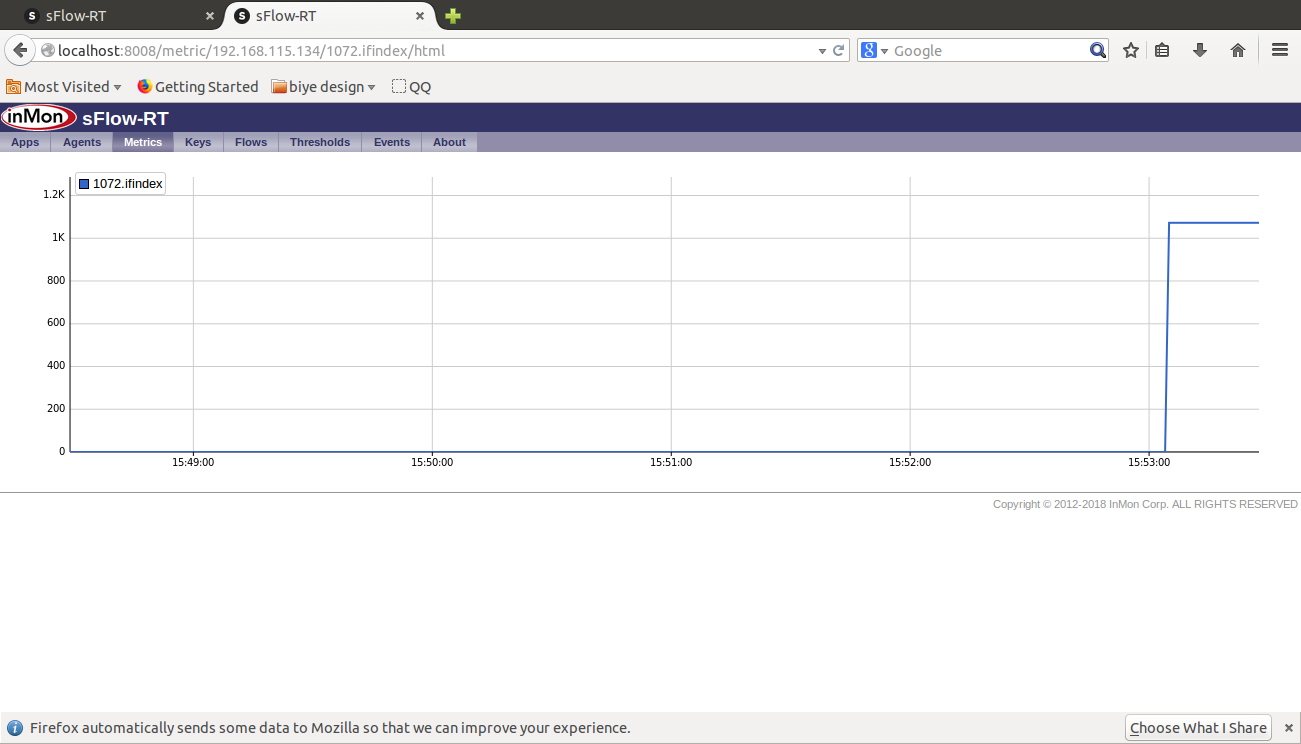


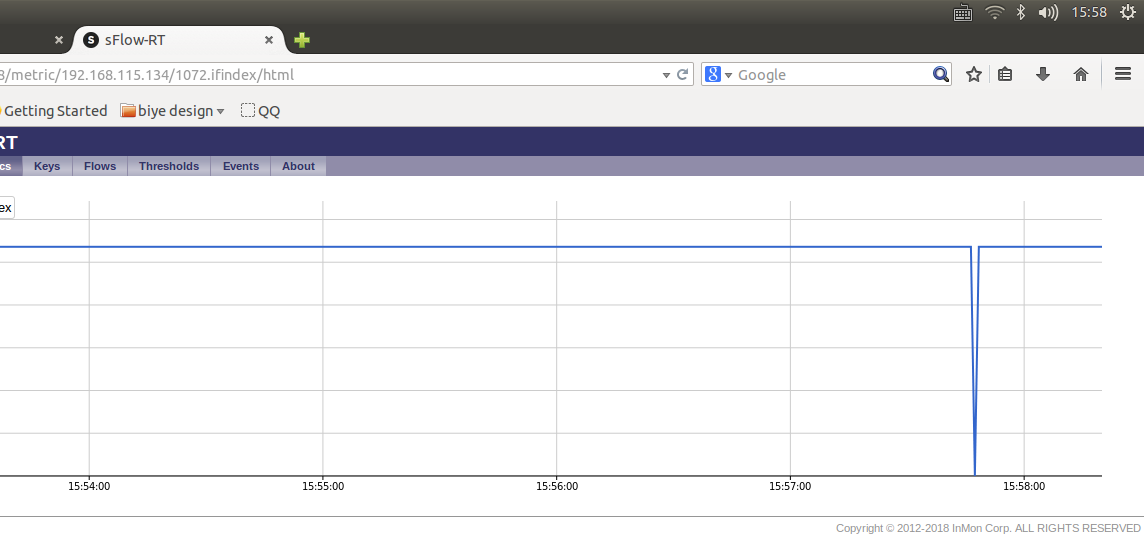
然后在mininet中执行pingall，模拟流量泛洪：



然后进入sFlow Collector发现有新增Agent节点







由上图可见在虚拟机内有真实流量存在