SDN学习笔记&ubuntu更新下载源

SDN简介

SDN（soft defined network）,软件定义网络。用于代替原有的不可编程的

mininet可视化工具设置拓扑结构

https://www.cnblogs.com/qq952693358/p/5860649.html

ubuntu更新下载源

https://blog.csdn.net/zhangjiahao14/article/details/80554616/

安装

1、网络仿真工具mininet

```
sudo apt-get install mininet
```

完成后，在命令行中输入

```
mn -topo=linear,3
```

```
user1@ubuntu:~/mininet$ sudo mn --topo=linear,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
```

出现如上所示，则说明安装成功。

2、安装openFlow

```
git clone git://github.com/mininet/mininet
cd mininet
./util/install.sh -fnv
```

3、构造并运行拓扑结构

命令行中运行指令：

```
sudo mn --topo=single,4 --switch ovs,protocols=OpenFlow10
```

mn:

mininet的缩写，用于启用mininet程序

--topo=single,4:

topo为设置拓扑结构；single为单一拓扑结构，即只有一个交换机，主机数量自由指定，呈放射状。4为主机数量。

--switch ovs,protocols=OpenFlow10:

ovs:

交换机使用ovs类型。

> 补充：
>
> 一共有三种交换机类型，分别是内核型、user型和OVS型。相比较而言，user型的性能最差，所谓user就是os为每个用户划分一个单独的用户空间，这也就意味着数据包需要额外经历一个内核到用户空间的过程，那就会导致延时增大，吞吐量减小。每个mininet虚拟机中都会预安装一个OVS，OVS型switch和内核型的性能差不多，甚至更好一些。
>
> 参考博客：https://blog.csdn.net/Neo233/article/details/79768904

protocols=OpenFlow10:

用OF1.0模式启动OVS

> 补充：
>
> --topo可指定的类型有：linear、minimal、reversed、single、torus以及tree共六种，另一类是自定义型。
>
> 参考博客：https://blog.csdn.net/wuliangtianzu/article/details/82689347

完成作业部分

dump命令

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=9343>
<Host h2: h2-eth0:10.0.0.2 pid=9345>
<Host h3: h3-eth0:10.0.0.3 pid=9347>
<Host h4: h4-eth0:10.0.0.4 pid=9349>
<OVSSwitch{'protocols': 'OpenFlow10'} s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None
,s1-eth3:None,s1-eth4:None pid=9354>
<Controller c0: 127.0.0.1:6653 pid=9336>
```

pingall命令

```
user1@ubuntu: ~/mininet/examples
File  Edit  View  Search  Terminal  Help
OpenFlow10
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

iperf命令

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['39.6 Gbits/sec', '39.6 Gbits/sec']
```

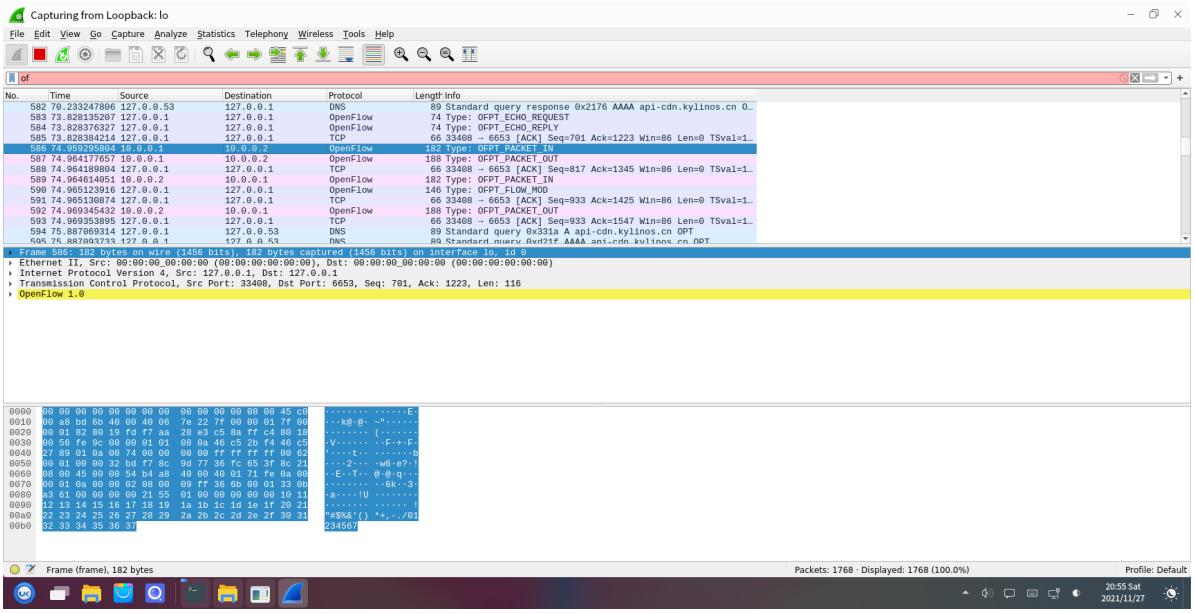
of和icmp && not of 两条指令都报错

of_packet_in

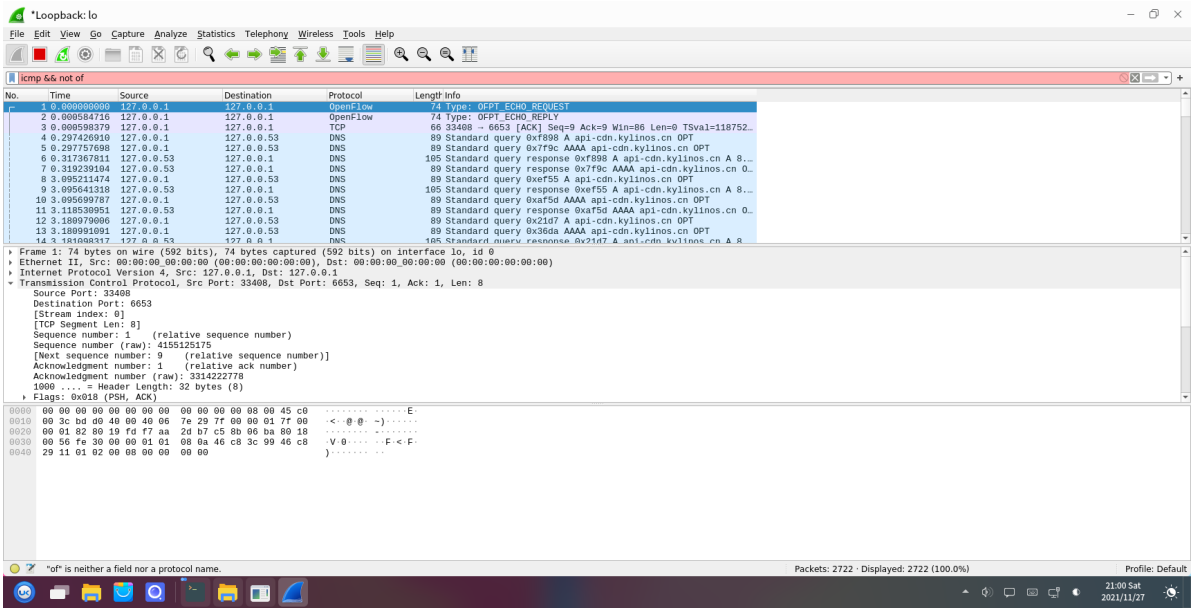
7个

sourceip:10.0.0.1

destip:10.0.0.2



icmp && not of



自定义构造如图拓扑结构（与作业无关）

在/mininet/examples文件夹下运行

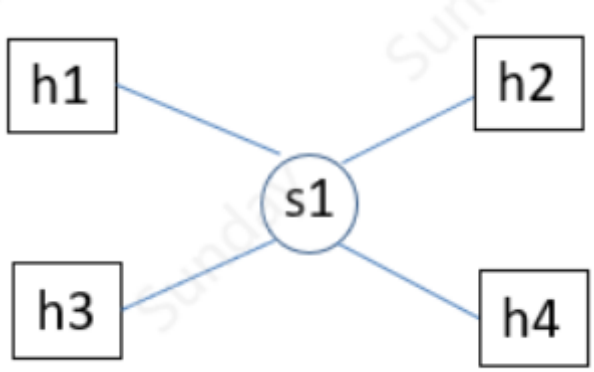
```
sudo ./miniedit.py
```

如果出现

```
Traceback (most recent call last):
  File "./miniedit.py", line 43, in <module>
    from Tkinter import ( Frame, Label, LabelFrame, Entry, OptionMenu,
  File "/usr/lib/python2.7/lib-tk/Tkinter.py", line 42, in <module>
    raise ImportError, str(msg) + ', please install the python-tk package'
ImportError: No module named _tkinter, please install the python-tk package
```

按照要求安装相应包(python-tk)即可。

```
sudo apt install python-tk
```



控制器配置：

鼠标右键s1选择properties,设置

DPID：0000000000000001

switch type(交换机的类型)：ovs交换机

ip地址：10.0.0.101

主机配置：

ip地址：h1,h2,h3,h4分别是10.0.0.1，10.0.0.2，10.0.0.3，10.0.0.4

全局配置：

接下来，就是对全局进行设置，并运行拓扑。

点击界面左上角的edit后，出现properties，点击后进入设计界面，此时需要勾选start CLI，只有勾选这个后，才可以在Linux终端中进行操作，还可以根据需要，选择支持的openflow协议。

保存

点击file->save，保存为python脚本。

运行

点击左下角run运行。

如果出现File exists错误

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python2.7/lib-tk/Tkinter.py", line 1550, in __call__
    return self.func(*args)
  File "./miniedit.py", line 1393, in doRun
    self.start()
  File "./miniedit.py", line 3024, in start
    self.net = self.build()
  File "./miniedit.py", line 2926, in build
    self.buildLinks(net)
  File "./miniedit.py", line 2910, in buildLinks
    net.addLink(srcNode, dstNode)
  File "/usr/lib/python2.7/dist-packages/mininet/net.py", line 366, in addLink
    link = cls( node1, node2, **options )
  File "/usr/lib/python2.7/dist-packages/mininet/link.py", line 449, in __init__
    node1, node2, deleteIntfs=False )
  File "/usr/lib/python2.7/dist-packages/mininet/link.py", line 493, in
makeIntfPair
    deleteIntfs=deleteIntfs )
  File "/usr/lib/python2.7/dist-packages/mininet/util.py", line 194, in
makeIntfPair
    ( intf1, intf2, cmdOutput ) )
Exception: Error creating interface pair (s1-eth1,h3-eth0): RTNETLINK answers:
File exists
```

原因是先前创建了一个拓扑结构，没有清除，所以若在.py文件中构建相同的拓扑图时需要先清除掉先前的拓扑结构。

关闭可视化工具，使用如下命令清除：

```
sudo mn -c
```