高级计算机网络实验

Project－3

姓 名: 胡 旻 晗

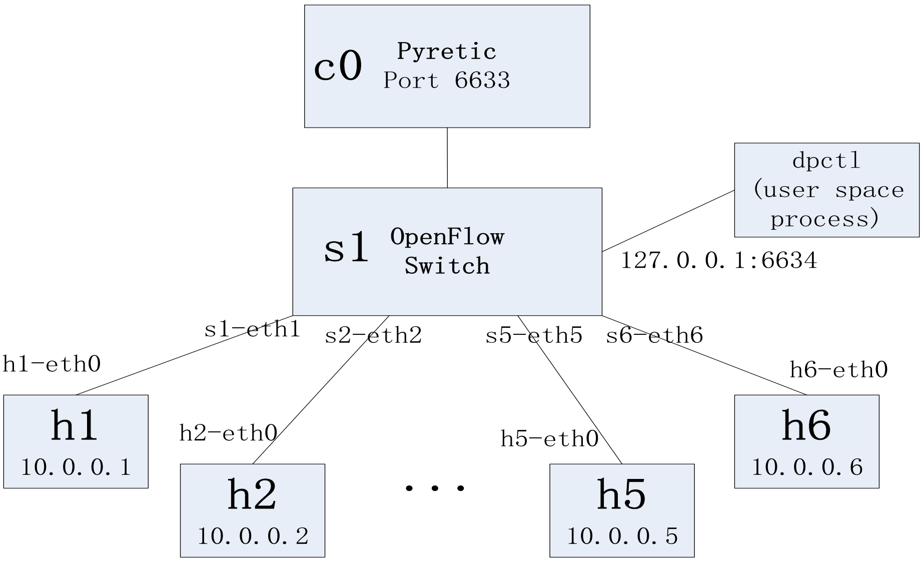
学 号: SA17011118

指导老师: 田 野

一、实验要求

1. 使用 Mininet 模拟拓扑如下图；

2. 在该 中，利用 Pyretic 读取一个自定义的防火墙规则来实现两层防火墙的功能。



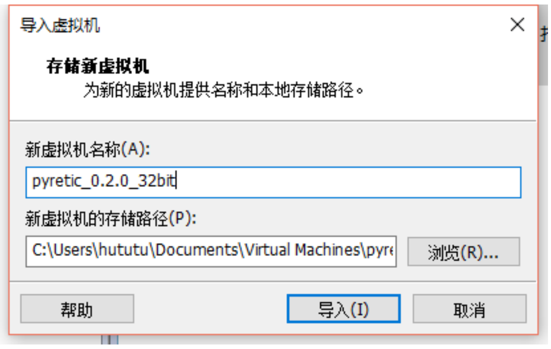
二、实验环境及工具

Vmware 虚拟机, Putty, Xming, Pyretic, Mininet,

三、实验内容

1. 下载安装 VMware 虚拟机

2. 下载 Mininet 虚拟机映像，并在 VMware 中导入



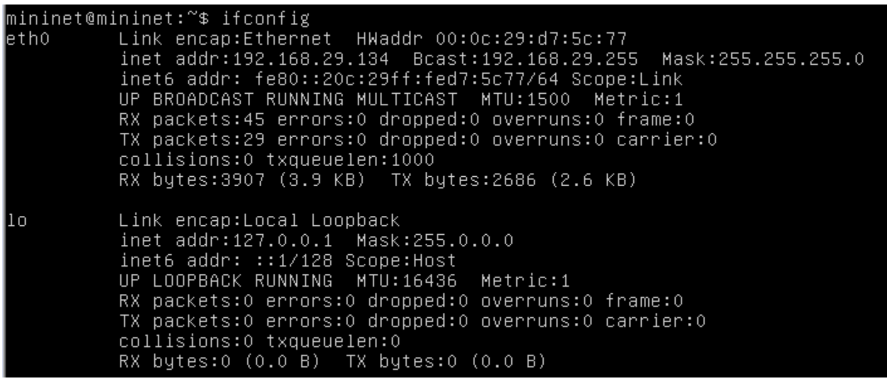
存储新虚拟机

为新的虚拟机提供名称和本地存储路径。

新虚拟机名称(A): pyretic_0.2.0_32bit

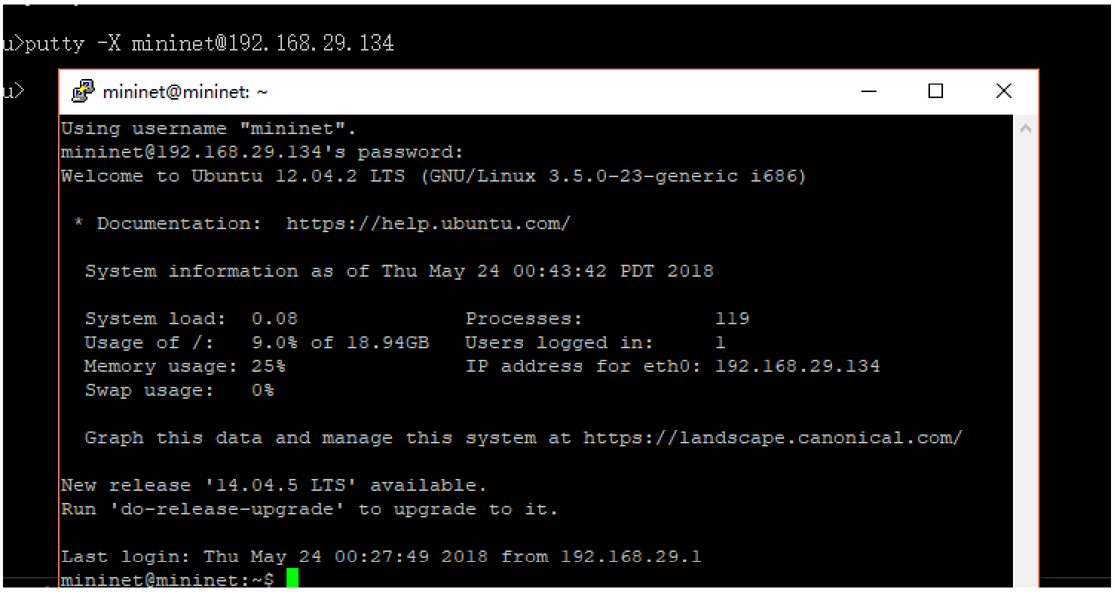
新虚拟机的存储路径(P): C:\Users\hututu\Documents\Virtual Machines\pyre

3. 登录系统并配置网卡



```
mininet@mininet:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:d7:5c:77
          inet addr:192.168.29.134  Bcast:192.168.29.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed7:5c77/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3907 (3.9 KB)  TX bytes:2686 (2.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

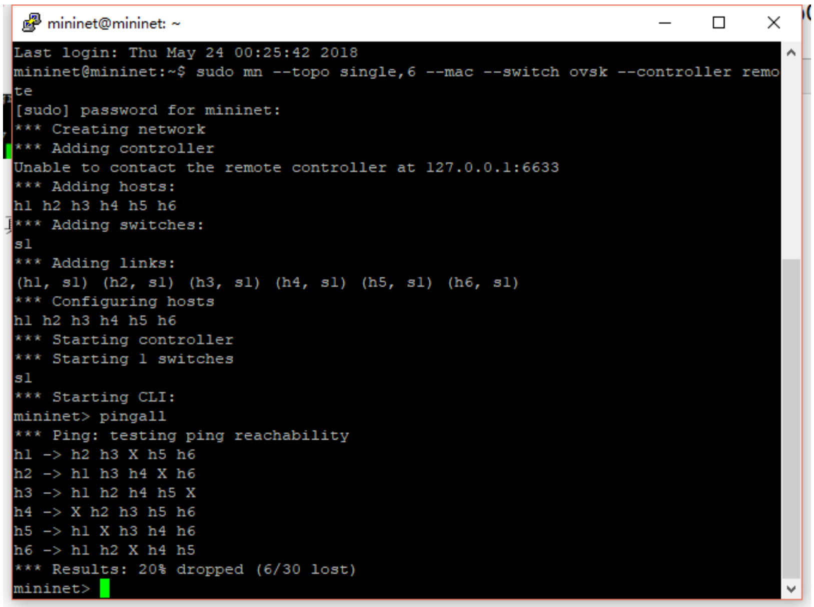
4. 安装 Xming 和 Putty，通过 Putty 远程登录 Mininet



```
u>putty -X mininet@192.168.29.134

u>
```

```
Using username "mininet".
mininet@192.168.29.134's password:
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-23-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Thu May 24 00:43:42 PDT 2018

  System load:  0.08              Processes:           119
  Usage of /:   9.0% of 18.94GB   Users logged in:     1
  Memory usage: 25%               IP address for eth0: 192.168.29.134
  Swap usage:   0%

  Graph this data and manage this system at https://landscape.canonical.com/

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu May 24 00:27:49 2018 from 192.168.29.1
mininet@mininet:~$
```

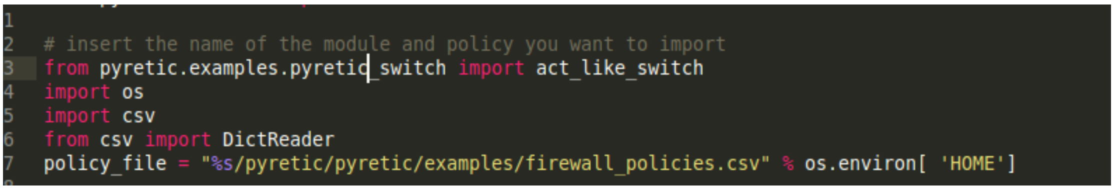
5. 利用 Mininet 模拟仿真实验拓扑结构



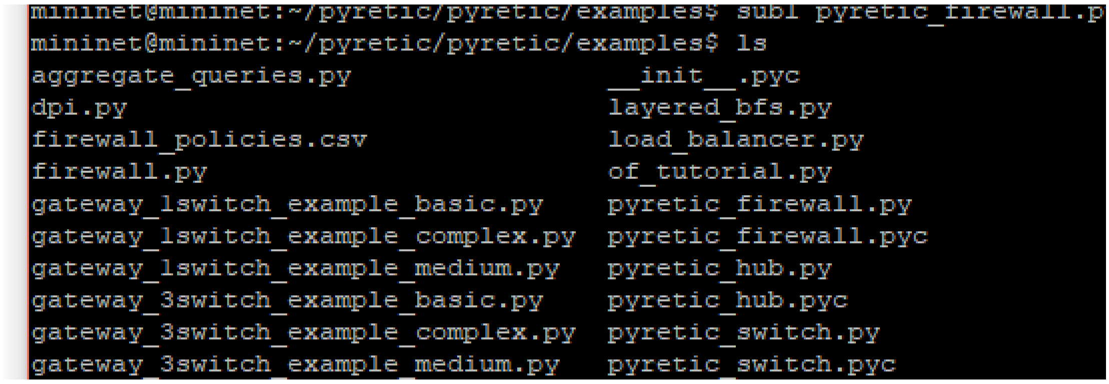
```
Last login: Thu May 24 00:25:42 2018
mininet@mininet:~$ sudo mn --topo single,6 --mac --switch ovsk --controller remo
te
[sudo] password for mininet:
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X h5 h6
h2 -> h1 h3 h4 X h6
h3 -> h1 h2 h4 h5 X
h4 -> X h2 h3 h5 h6
h5 -> h1 X h3 h4 h6
h6 -> h1 h2 X h4 h5
*** Results: 20% dropped (6/30 lost)
mininet>
```

6. 对 Pyretic_firewall.py 进行修改

（1）添加模块和协议规则，声明 policy 文件的路径。并将 Pyretic_switch.py 放入~/pyretic/pyretic/examples/中

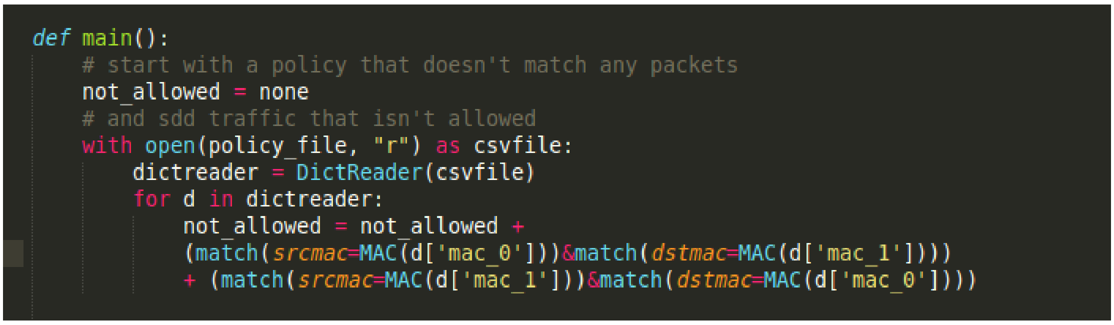


```
# insert the name of the module and policy you want to import
from pyretic.examples.pyretic_switch import act_like_switch
import os
import csv
from csv import DictReader
policy_file = "%s/pyretic/pyretic/examples/firewall_policies.csv" % os.environ[ 'HOME']
```



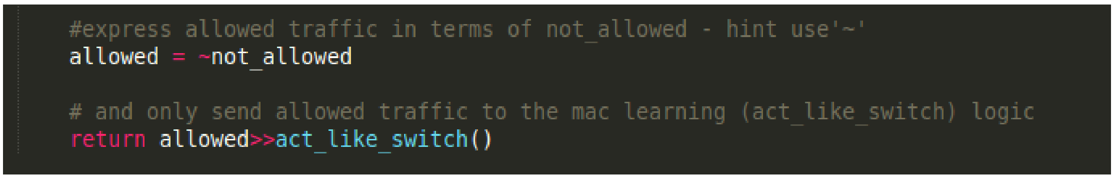
```
mininet@mininet:~/pyretic/pyretic/examples$ subl pyretic_firewall.p
mininet@mininet:~/pyretic/pyretic/examples$ ls
aggregate_queries.py                 __init__.pyc
dpi.py                               layered_bfs.py
firewall_policies.csv                load_balancer.py
firewall.py                          of_tutorial.py
gateway_1switch_example_basic.py     pyretic_firewall.py
gateway_1switch_example_complex.py   pyretic_firewall.pyc
gateway_1switch_example_medium.py    pyretic_hub.py
gateway_3switch_example_basic.py     pyretic_hub.pyc
gateway_3switch_example_complex.py   pyretic_switch.py
gateway_3switch_example_medium.py    pyretic_switch.pyc
```

（2）将禁止条件 not_allowed 赋值为 none，然后根据从文件中读取的相应 MAC 对加入禁止条件中，即对 not_allowed 进行相应赋值



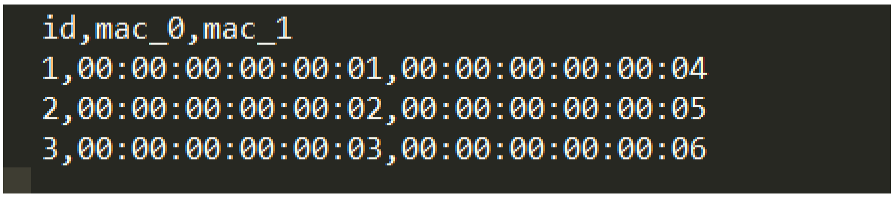
```
def main():
    # start with a policy that doesn't match any packets
    not_allowed = none
    # and sdd traffic that isn't allowed
    with open(policy_file, "r") as csvfile:
        dictreader = DictReader(csvfile)
        for d in dictreader:
            not_allowed = not_allowed +
            (match(srcmac=MAC(d['mac_0']))&match(dstmac=MAC(d['mac_1'])))
            + (match(srcmac=MAC(d['mac_1']))&match(dstmac=MAC(d['mac_0'])))
```

（3）not_allowd 取反获得 allowed，通过>>操作符赋给 act_like_switch()作为输出



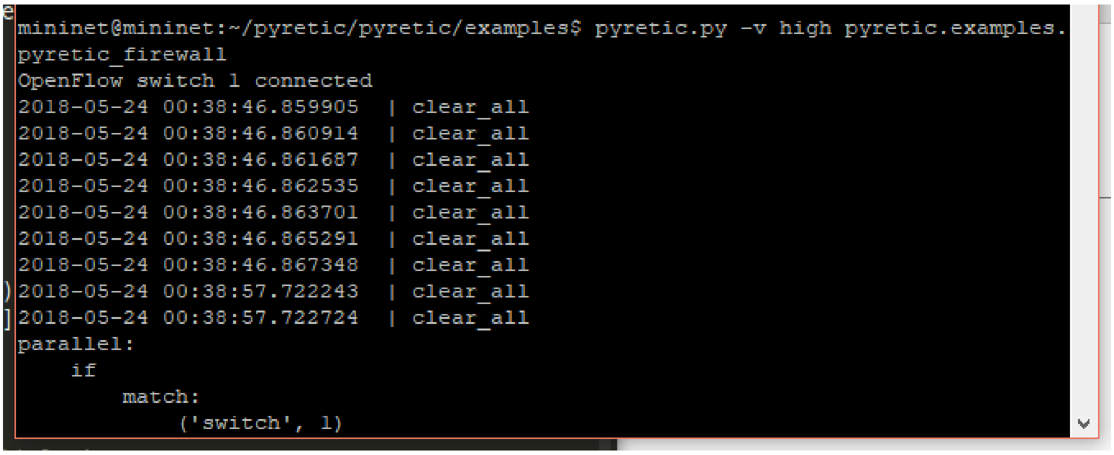
```
#express allowed traffic in terms of not_allowed - hint use'~'
allowed = ~not_allowed

# and only send allowed traffic to the mac learning (act_like_switch) logic
return allowed>>act_like_switch()
```

7. 策略文件内容如下



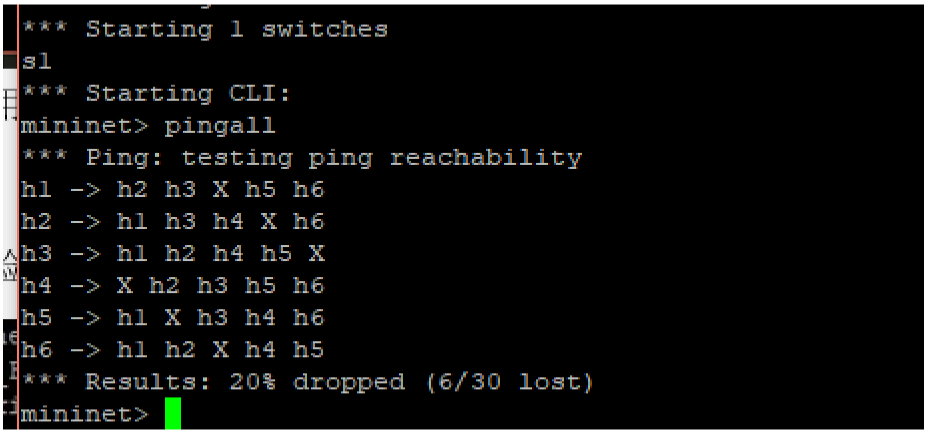
```
id,mac_0,mac_1
1,00:00:00:00:00:01,00:00:00:00:00:04
2,00:00:00:00:00:02,00:00:00:00:00:05
3,00:00:00:00:00:03,00:00:00:00:00:06
```

8. 将防火墙协议运用到 mininet 中



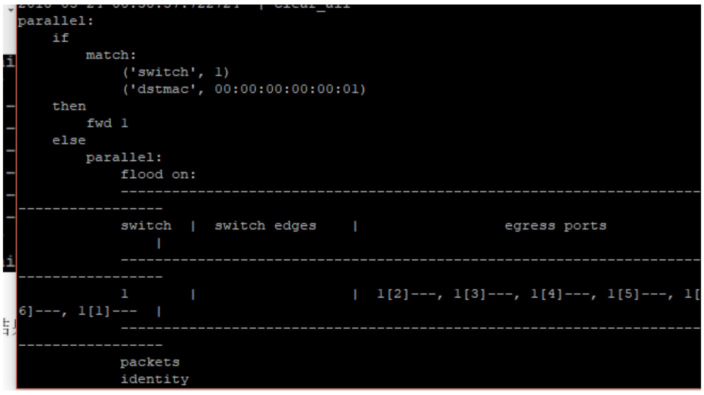
```
mininet@mininet:~/pyretic/pyretic/examples$ pyretic.py -v high pyretic.examples.
pyretic_firewall
OpenFlow switch 1 connected
2018-05-24 00:38:46.859905  | clear_all
2018-05-24 00:38:46.860914  | clear_all
2018-05-24 00:38:46.861687  | clear_all
2018-05-24 00:38:46.862535  | clear_all
2018-05-24 00:38:46.863701  | clear_all
2018-05-24 00:38:46.865291  | clear_all
2018-05-24 00:38:46.867348  | clear_all
2018-05-24 00:38:57.722243  | clear_all
2018-05-24 00:38:57.722724  | clear_all
parallel:
    if
        match:
            ('switch', 1)
```

9. 使用 pingall 测试主机间的拓扑结构连通性

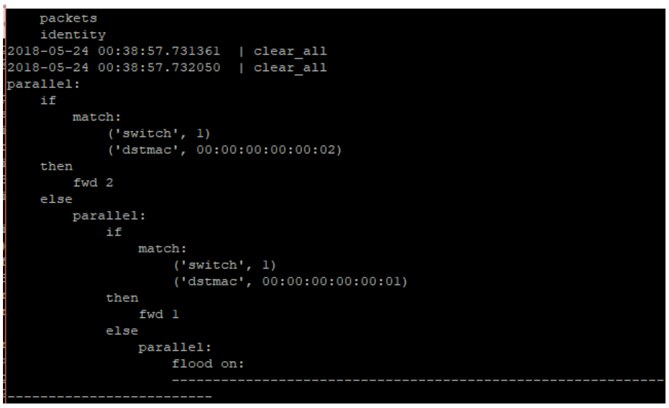


```
*** Starting 1 switches
s1
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X h5 h6
h2 -> h1 h3 h4 X h6
h3 -> h1 h2 h4 h5 X
h4 -> X h2 h3 h5 h6
h5 -> h1 X h3 h4 h6
h6 -> h1 h2 X h4 h5
*** Results: 20% dropped (6/30 lost)
mininet> 
```

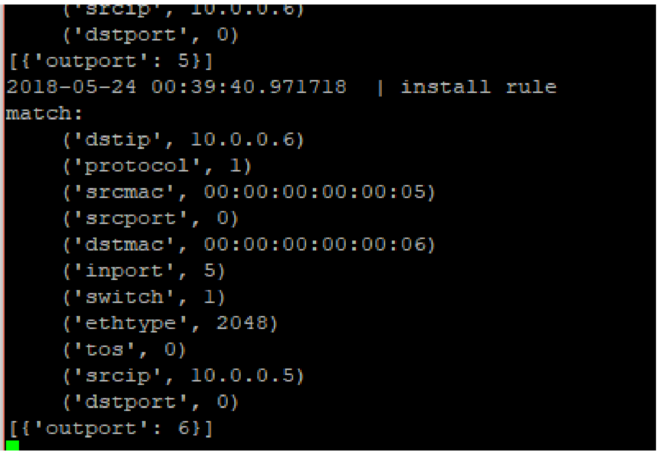
10. 得到的结果如下



```
parallel:
    if
        match:
            ('switch', 1)
            ('dstmac', 00:00:00:00:00:01)
    then
        fwd 1
    else
        parallel:
            flood on:
            ---------------------------------------------------------------------
-----------------
            switch  |  switch edges    |                  egress ports
                |
            ---------------------------------------------------------------------
-----------------
            1       |                  |  1[2]---, 1[3]---, 1[4]---, 1[5]---, 1[
6]---, 1[1]---  |
            ---------------------------------------------------------------------
-----------------
            packets
            identity
```



```
    packets
    identity
2018-05-24 00:38:57.731361  | clear_all
2018-05-24 00:38:57.732050  | clear_all
parallel:
    if
        match:
            ('switch', 1)
            ('dstmac', 00:00:00:00:00:02)
    then
        fwd 2
    else
        parallel:
            if
                match:
                    ('switch', 1)
                    ('dstmac', 00:00:00:00:00:01)
            then
                fwd 1
            else
                parallel:
                    flood on:
                    ------------------------------------------------------------
-------------------------
```



```
    ('srcip', 10.0.0.6)
    ('dstport', 0)
[{'outport': 5}]
2018-05-24 00:39:40.971718  | install rule
match:
    ('dstip', 10.0.0.6)
    ('protocol', 1)
    ('srcmac', 00:00:00:00:00:05)
    ('srcport', 0)
    ('dstmac', 00:00:00:00:00:06)
    ('inport', 5)
    ('switch', 1)
    ('ethtype', 2048)
    ('tos', 0)
    ('srcip', 10.0.0.5)
    ('dstport', 0)
[{'outport': 6}]
```

11. 通过上下文中的实验结果， 很明显的可以看出,h1 和 h4,h2 和 h5， h3 和 h6 之间的丢包率为 100%， 和定义的防火墙协议符合， 因此防火墙实现了预期的功能

四、实验总结

通过本次试验，对 Pyretic 有了一定的了解，能够利用 Pyretic 为模拟出的网络拓扑加入防火墙协议。这个过程实际是对于那些满足协议规定的包的缓存进行清空，以达到防火墙所需功能。这充分体现了 Openflow 在实现数据转发和路由控制的分离的功能，controller 可以通过事先定义好的接口来控制 OpenFlow 交换机中的流表，从而达到控制数据转发的目的。