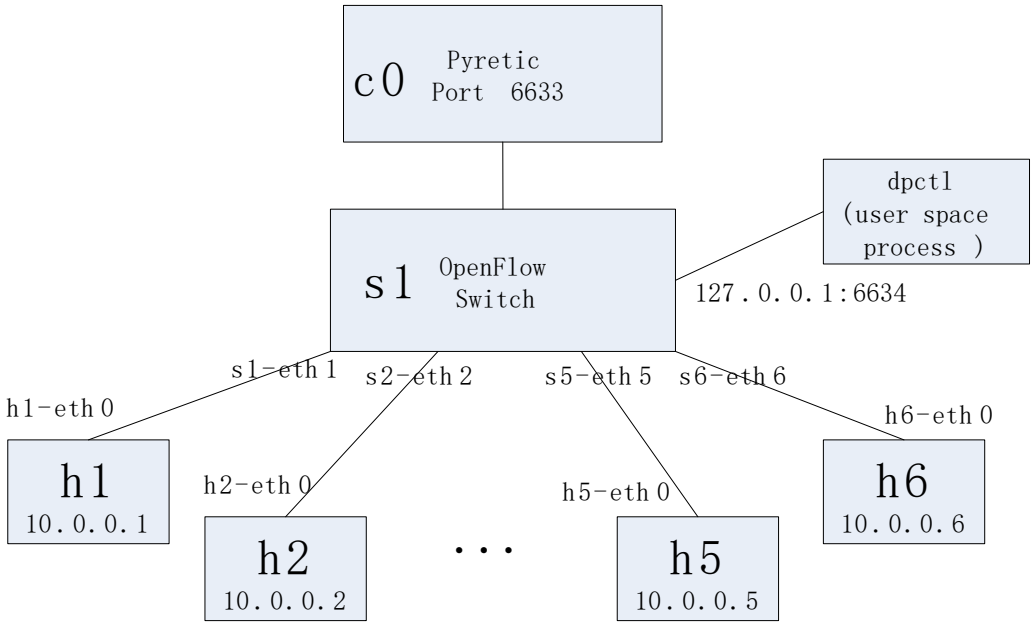
高级计算机网络第三次实验报告

SA16011112 殷康麒

1. 实验目标

使用 Pyretic 实现第二层 Mac 地址上的防火墙。

2. 网络的拓扑结构



由上图所示的拓扑结构可以看出，本次实验的网络拓扑非常简单，6 个主机直接连接到交换机上面，控制器为 Pyretic 控制器。

3. 防火墙代码及相关的解释

#本策略使用的黑名单方法，黑名单中指定禁止的策略，其他的默认通过。

#导入相关的包以及相关模块

```
from pyretic.lib.corelib import *
from pyretic.lib.std import *
from pyretic.examples.pyretic_switch import act_like_switch
import os, csv
```

```
policy_file = "%s/pyretic/pyretic/examples/firewall-policies.csv" % os.environ[ 'HOME' ]


def main():
    #初始化
  not_allowed = none


  #从 csv 读取策略文件
  with open(policy_file, "r") as policy_content:
    dictReader = csv.DictReader(policy_content)
      #添加禁止策略, 双向通信都禁止
    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']))))



  #添加许可策略, 表示为禁止策略取反。
  allowed = ~not_allowed
  #先输出一下当前规则
  print allowed


  # 将允许的路由输入给作为 pyretic_switch 中的 act_like_switch
  return allowed >> act_like_switch()
```

4. 防火墙规则

```
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
```

如上表所示，为 firewall-policies 的规则，首列为规则编号，后面跟着的两个为不允许通讯的两个主机的 Mac 地址。

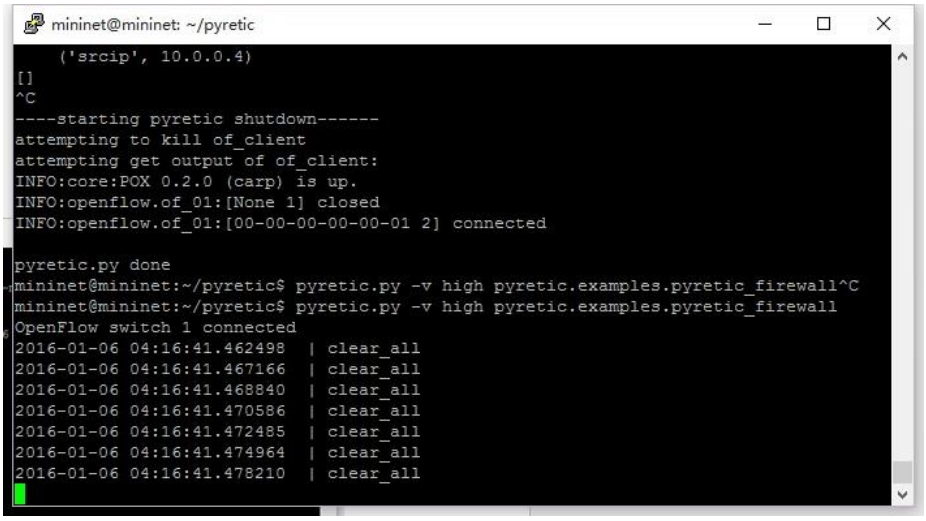
5. 实验及相关结果

5.1 启动控制器

将上述两个文件保存到 Pyretic/examples 的目录下之后，执行以下命令启动控制器。

```
pyretic.py -v high pyretic.examples.pyretic_firewall
```

执行之后可以观察到控制器已经启动（如图 5.1），没有报错说明防火墙程序没有问题。

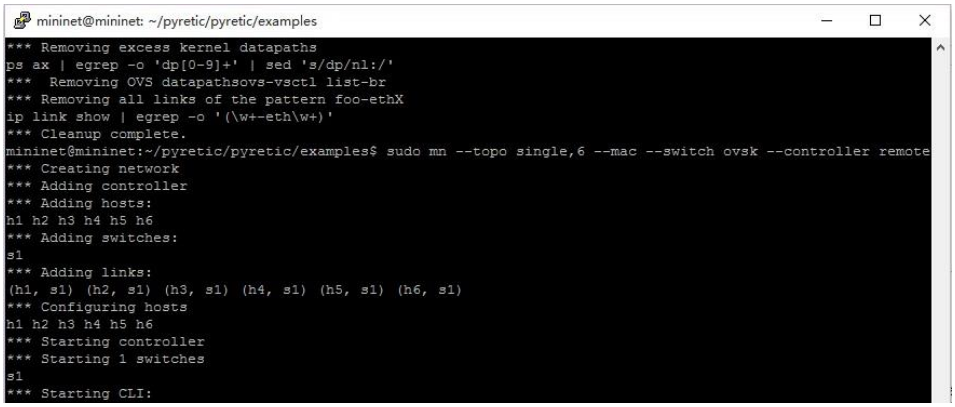


5.2 创建拓扑

执行以下命令创建一个 6 个节点连接到一个交换机上面的网络拓扑。

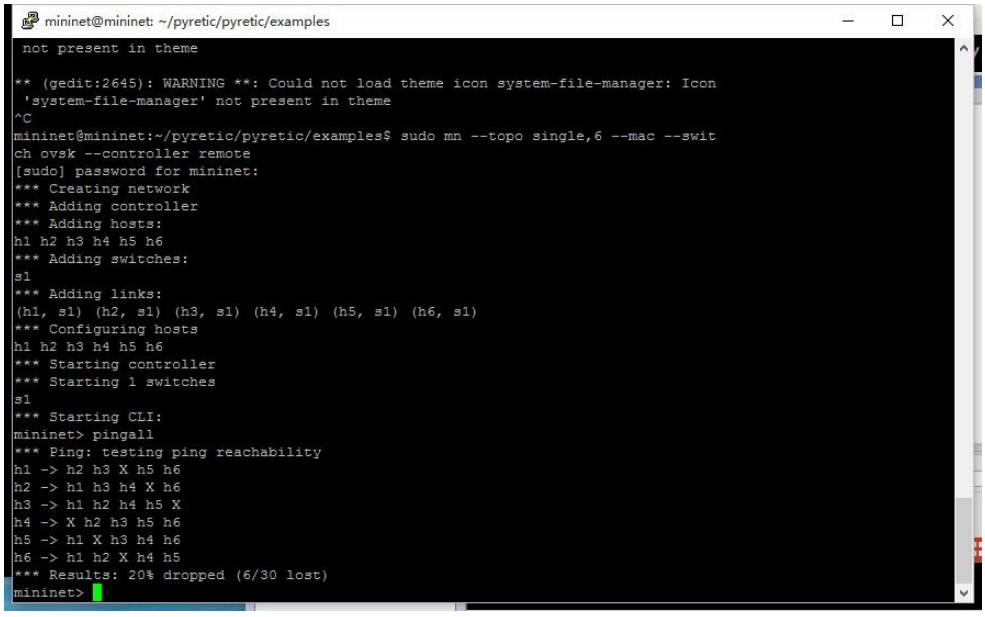
```
sudo mn --topo single,6 --mac --switch ovsk --controller remote
```

执行完成之后观察到如图 5.2 所示的界面，表示拓扑创建成功。



5.3 Pingall 测试

拓扑创建完成之后，执行 pingall 命令，观察到如下图 5,3 所示的界面，可以很明显的看出定义的防火墙策略已经生效。

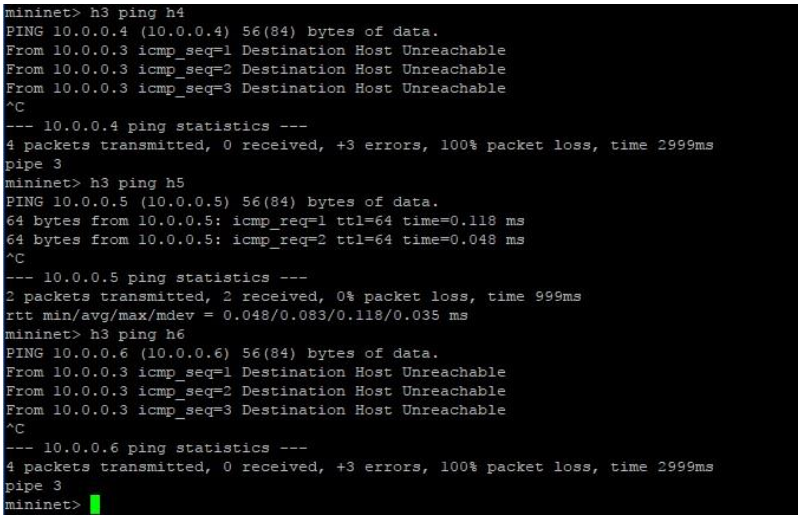


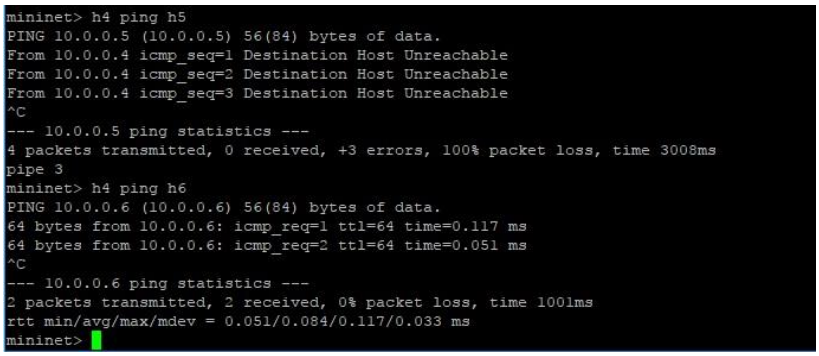
5.4 单个交换机 Ping 测试

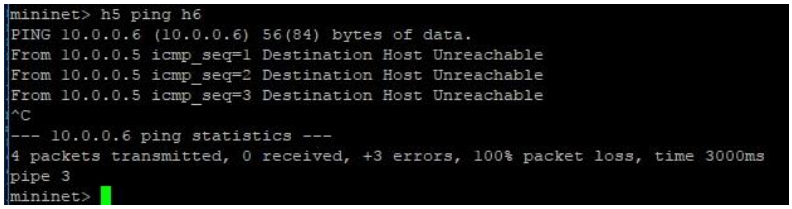
```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable
From 10.0.0.1 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
9 packets transmitted, 0 received, +6 errors, 100% packet loss, time 8046ms
pipe 3
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_req=1 ttl=64 time=227 ms
64 bytes from 10.0.0.3: icmp_req=2 ttl=64 time=0.049 ms
64 bytes from 10.0.0.3: icmp_req=3 ttl=64 time=0.048 ms
64 bytes from 10.0.0.3: icmp_req=4 ttl=64 time=0.152 ms
^C
--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.048/56.860/227.191/98.340 ms
mininet> h1 ping h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_req=1 ttl=64 time=152 ms
64 bytes from 10.0.0.5: icmp_req=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.5: icmp_req=3 ttl=64 time=0.046 ms
^C
--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.046/50.823/152.372/71.805 ms
mininet>
```

可以看出，h1 无法与 h2\h4\h6 通信，可以 ping 通 h3\h5

```
rtt min/avg/max/mdev = 0.046/50.823/152.372/71.805 ms
mininet> h2 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.3 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2999ms
pipe 3
mininet> h2 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_req=1 ttl=64 time=212 ms
64 bytes from 10.0.0.4: icmp_req=2 ttl=64 time=0.051 ms
^C
--- 10.0.0.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.051/106.124/212.197/106.073 ms
mininet> h2 ping h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.5 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2999ms
pipe 3
mininet> h2 ping h6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_req=1 ttl=64 time=100 ms
^C
--- 10.0.0.6 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 100.609/100.609/100.609/0.000 ms
mininet>
```







6. 总结

本次实验使用 Pyretic 控制器，使用控制器实现了 Mac 层上的防火墙相关的策略。也很容易从本次 Project 中看出，使用 SDN 网络实现防火墙策略的管理比传统网络更加容易。