reporter

建立拓扑

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

建立防火墙

防火墙规则

```
    for (id, macA, macB) in rules:
        if id == 'id':
            continue
        not_allowed  = not_allowed | match(srcmac=MAC(macA), dstmac=MAC(macB)) |
match(srcmac=MAC(macB), dstmac=MAC(macA))
```

csv中的每一表项取**并集**，只要匹配**srcmac和dstmat(双向)**，就会**丢弃。**不匹配的数据包会进行act_like_switch()匹配。

switch规则

```
    def learn_from_a_packet(pkt):
        # Set the forwarding policy
        self.forward = if_(match(dstmac=pkt['srcmac'],
                                 switch=pkt['switch']), fwd(pkt['inport']),
                           self.policy)  # hint use 'match', '&', 'if_', and
'fwd'
        # Update the policy
        self.policy = self.forward + self.query # hint you've already written
this
        print self.policy
```

默认**flood**，会记录不在policy中的数据包**srcmac**和**inport**，优先匹配mac地址。

rules and pingall screenshots

- firewall rules

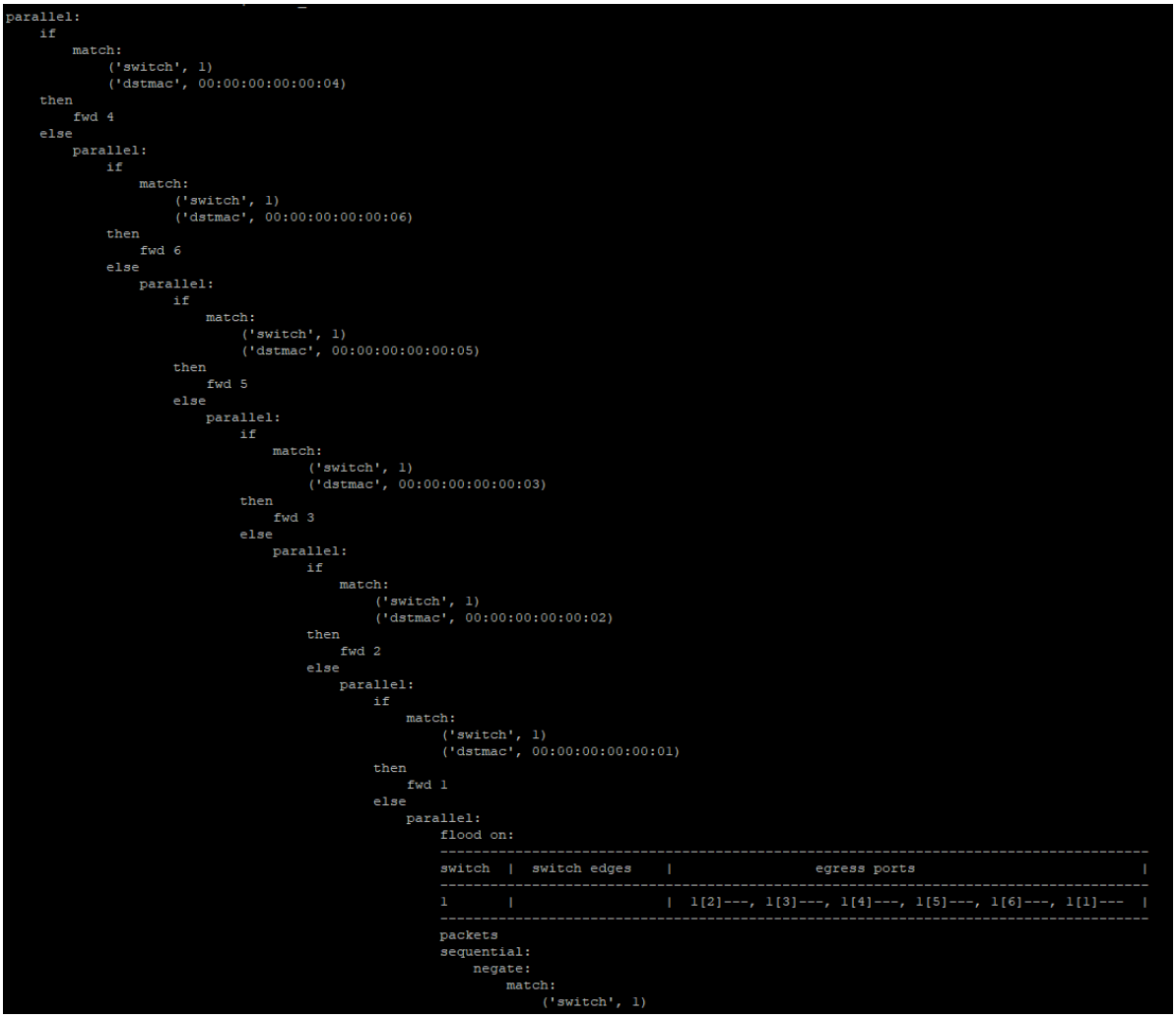
```
allowed:
negate:
    parallel:
        none
        match:
            ('dstmac', 00:00:00:00:00:04)
            ('srcmac', 00:00:00:00:00:01)
        match:
            ('dstmac', 00:00:00:00:00:01)
            ('srcmac', 00:00:00:00:00:04)
        match:
            ('dstmac', 00:00:00:00:00:05)
            ('srcmac', 00:00:00:00:00:02)
        match:
            ('dstmac', 00:00:00:00:00:02)
            ('srcmac', 00:00:00:00:00:05)
        match:
            ('dstmac', 00:00:00:00:00:06)
            ('srcmac', 00:00:00:00:00:03)
        match:
            ('dstmac', 00:00:00:00:00:03)
            ('srcmac', 00:00:00:00:00:06)
OpenFlow switch 1 connected
```

- pingall

```
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>
```

self.policy

pingall以后的policy



按照**目的mac**依次匹配4，6，5，3，2，1，匹配成功则从与mac地址**相同的端口**转发，否则flood。