

# mininet踩坑记录

---

## mininet踩坑记录

### 实验2

#### 可能出现的问题

#### 解决办法

### 实验3

## 实验2

---

### 可能出现的问题

1. 拓扑出现环是ping不通的，需要启用生成树协议

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X
h2 -> X X X
h3 -> X X X
h4 -> X X X
*** Results: 100% dropped (12/12 lost)
```

网络上的stp解决方案大都不能用，因为老师给的mininet版本不支持，例如：

[FAQ · mininet/mininet Wiki \(github.com\)](https://github.com/mininet/mininet/wiki/FAQ)

Why does my controller, which implements an Ethernet bridge or learning switch, not work with my network which has loops in it? I can't ping anything!

tl;dr: use `--switch lxbr,stp=1` or `--switch ovsbr,stp=1` and wait for STP to converge.

[mininet - How to enable brctl stp using python script? - Stack Overflow](https://stackoverflow.com/questions/44114111/mininet-how-to-enable-brctl-stp-using-python-script)

# How to enable brctl stp using python script?

Asked 6 years ago   Active 3 years, 11 months ago   Viewed 1k times

1 In mininet I am building a topology using `LinuxBridge` from `mininet.nodelib.LinuxBridge` instead of `OVSKernelSwitch`.

```
s1 = net.addSwitch('s1', cls=LinuxBridge)
```

we can enable the stp of s1 from the command line by

```
brctl stp s1 on
```

How to enable the STP of the bridge `s1` directly from the writing python script instead of CLI.

For example, in `OVSKernelSwitch` we write:

```
s1.cmd('ovs-vsctl set bridge s1 stp-enable=true')
```

to enable the stp to break the loop in the network topology.

What is the `brctl stp enable` script? I have tried with:

```
s1.cmd('brctl stp', s1, 'on')
```

but it is unable to enable the stp.

`python` `mininet` `stp`

[SDN实验（一）：Mininet的安装问题与Fat-Tree的构建 - 知乎 \(zhihu.com\)](#)

其中提到，在这种情况下，应该为网络开启生成树协议（STP），这样可以明确转发路径，进而避免数据包在一个环上来回转发。对应到实现上，需要修改的是 `addSwitch` 部分，为其增加一些参数：

```
switchOpts = {'cls': OVSBridge, 'stp': 1}
addSwitch('sxx', **switchOpts)
```

这里指定使用 `OVSBridge` 类型的交换机，`stp=1` 开启 STP 协议。另外，开启 STP 后需要等待交换机连接，所以脚本中

```
net.waitConnected()
```

是必不可少的，在连接上之前，网络都是不通的。或者也可以在 CLI 下调用这个方法：（启动时创建的 Mininet 实例一般都叫做 `net`）

```
mininet> py net.waitConnected()
```

再启动拓扑进行测试，一切就都符合预期了。

# 解决办法

## 下载最新版mininet, 自定义topo + pox

如果用老师给的pyretic虚拟机, 也能进行下面1、2、3步, 但是没有第4步的命令, 所以查看生成树可能有点困难。你如果找到了别的方法查看生成树可以忽略。

### 1. 首先清理环境

1. `sudo mn -c`
2. `sudo killall ovs-controller`

### 2. 启动pox控制器[Using POX components to create a software defined networking application | Open-Source Routing and Network Simulation \(brianlinkletter.com\)](https://www.brianlinkletter.com/2014/05/20/Using-POX-components-to-create-a-software-defined-networking-application/)

```
sudo ~/pox/pox.py forwarding.12_learning openflow.spanning_tree --no-flood --hold-down  
openflow.discovery host_tracker openflow.of_01 --port=6633
```

如果没启动成功, 可以换个端口试试, 比如6653, 当然这个换了下面启动的拓扑也要跟着换

### 3. 自定义topo, 手敲代码或者miniedit图形编辑 (推荐miniedit, 方便截图, 还不容易出错)

1. 手敲代码启动方式(ppt上有讲解) `sudo mn --custom ./yourtopo.py --topo mytopo --controller=remote,port=6633`
2. miniedit启动方式: `sudo python2 ~/mininet/examples/miniedit.py`, 用python3会出现建立好的topo无法保存的情况, 搭建好网络, 不需要手动设置ip, controller设置成remote模式, 启动就ok了。可以保存成mn文件, 这个是可视化文件, 下次可以直接打开, 也可以保存成python文件, 不需要可视化的情况下, 可以直接建立拓扑。

如果miniedit使用过程中出现PutTY X11 proxy: Authorisation not recognised, 输入 `sudo cp /home/mininet/.Xauthority /root/.Xauthority`

miniedit或者wireshark如果出现打不开的情况, 请下载vcxsrv软件, 并启动。老师给的Xming可能会出现奇怪的报错。

### 4. 查看生成树(带NO\_FLOOD的表示逻辑链路断开)

用 `links` 查看各端口连接情况, 用 `dpctl dump-ports-desc` 查看哪些逻辑链路被禁用了。

```
*** s18 -----
OFPST_PORT_DESC reply (xid=0x2):
 1(s18-eth1): addr:a6:a5:ee:32:03:f5
   config:      0
   state:       0
   current:     10GB-FD COPPER
   speed: 10000 Mbps now, 0 Mbps max
 2(s18-eth2): addr:82:29:d9:f0:db:f0
   config:      NO_FLOOD
   state:       0
   current:     10GB-FD COPPER
   speed: 10000 Mbps now, 0 Mbps max
 3(s18-eth3): addr:5a:2a:7c:d7:6f:c7
   config:      NO_FLOOD
   state:       0
   current:     10GB-FD COPPER
   speed: 10000 Mbps now, 0 Mbps max
 4(s18-eth4): addr:6e:3e:f6:b7:79:f9
   config:      NO_FLOOD
   state:       0
   current:     10GB-FD COPPER
   speed: 10000 Mbps now, 0 Mbps max
LOCAL(s18): addr:fe:95:9c:ac:f9:49
   config:      PORT_DOWN
   state:       LINK_DOWN
   speed: 0 Mbps now, 0 Mbps max
```

## 实验3

直接使用csv文件中的mac地址是不行的，需要将读出来的文本mac地址转换成pyretic认识的MAC类

可以使用 `from pyretic.core.network import MAC`

然后再使用 `match(srcmac=MAC('00:00:00:00:00:01'))` 这样的命令