第 1 小题：简单网络

一、 实验目的

搭建如图 1.1 所示的简单网络，通过流表操作来实现两台不同主机间的 ping 通与否。

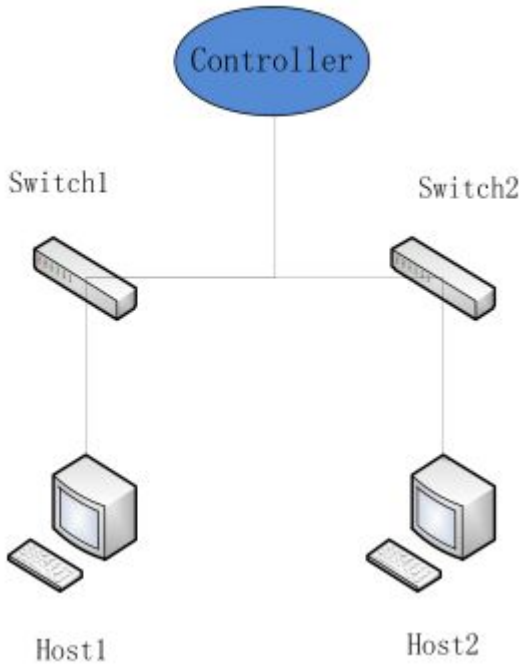

图 1.1：简单网络拓扑

二、 实验环境搭建

（一）设计思路

如图 1.1 所示，简单网络由一台 Controller，两台 switch 以及两台 host 组成。我们通过 Controller 添加流表让两台 host 无法 ping 通。

（二）设备以及平台

我们选择在物理机安装 VMware Workstation 10。下载 SDN Hub（sdnhub.org）构建的 all-in-one tutorial VM（以下称 SDN 虚拟机）并导入到 VMware。这是一个预装了很多 SDN 相关的软件和工具的 64 位的 Ubuntu 12.10 虚拟机映像。内置软件和工具如下：

- SDN 控制器：Opendaylight，Ryu，Floodlight，Pox 和 Trema
- 示例代码：hub，2 层学习型交换机和其它应用
- Open vSwitch 1.11 ：支持 Openflow 1.0，实验性的支持 Openflow 1.2 和 1.3
- Mininet：创建和运行示例拓扑
- Eclipse 和 Maven
- Wireshark：协议数据包分析

我们使用的控制器为 floodlight。

三、 实验过程及结果

（一）初始环境

首次进入 SDN 虚拟机，打开终端，输入以下命令，对 floodlight 进行编译及运行：

>>cd floodlight
>>ant
>>java -jar target/floodlight.jar
floodlight 开始监听交换机和 6633 端口（如图 1.2 和 1.3 所示）。

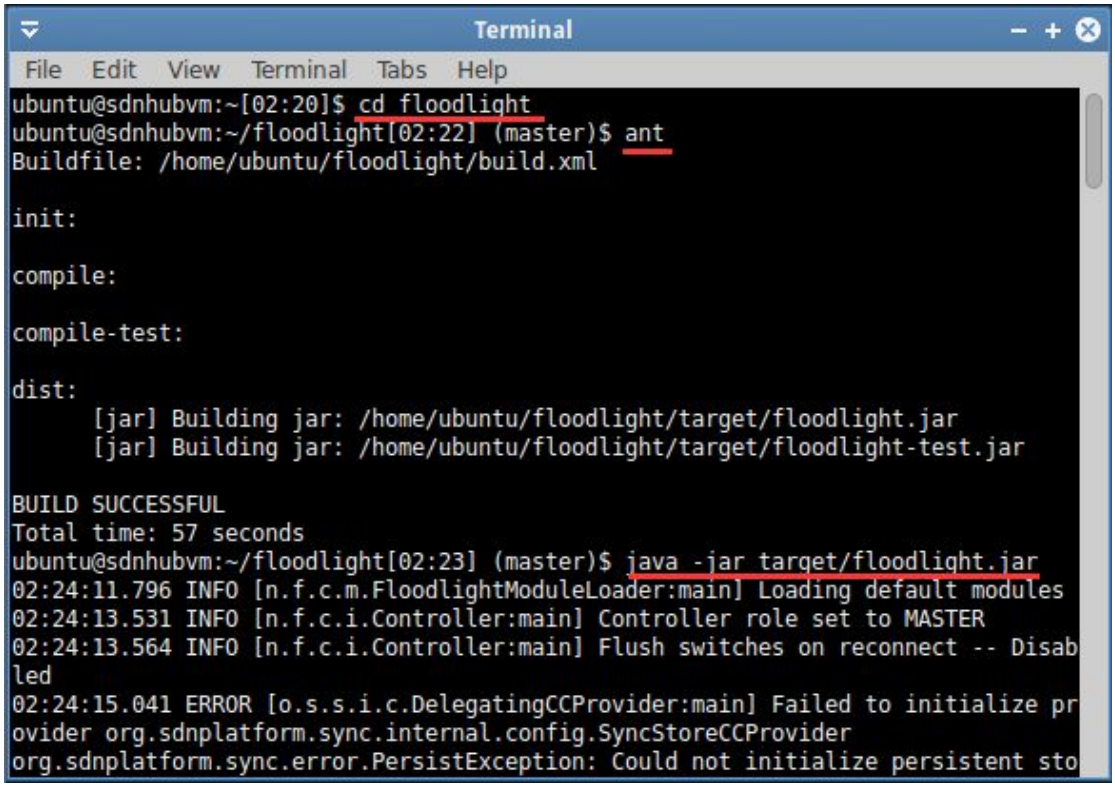

图 1.2：编译并运行 floodlight

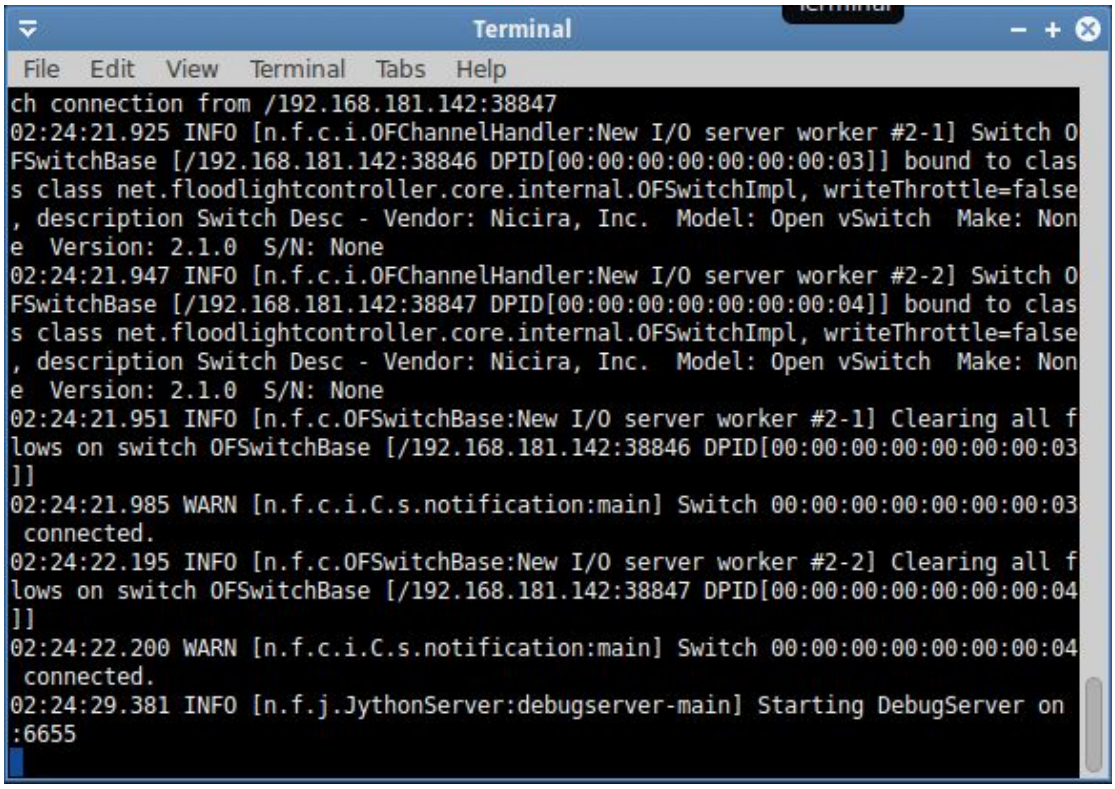

图 1.3：已运行的 floodlight

（二）mininet 创建拓扑

在终端中输入以下命令创建拓扑：

>>sudo mn --custom /home/ubuntu/mininet/custom/topo-2sw-2host.py --topo mytopo --switch ovsk --controller=remote,ip=192.168.181.142,port=6633

topo-2sw-2host.py 文件是该系统中已有的文件，可直接使用，且拓扑与题目要求相同。文件内容如下：

```
"""Custom topology example
Two directly connected switches plus a host for each switch:
    host --- switch --- switch --- host
Adding the 'topos' dict with a key/value pair to generate our newly defined
topology enables one to pass in '--topo=mytopo' from the command line.
"""
from mininet.topo import Topo
class MyTopo( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."
```

```
        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        leftHost = self.addHost( 'h1' )
        rightHost = self.addHost( 'h2' )
        leftSwitch = self.addSwitch( 's3' )
        rightSwitch = self.addSwitch( 's4' )

        # Add links
        self.addLink( leftHost, leftSwitch )
        self.addLink( leftSwitch, rightSwitch )
        self.addLink( rightSwitch, rightHost )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

结果如图 1.4 所示，此拓扑由两台主机 h1,h2 和两台交换机 s3,s4 组成，拓扑图如图 1.1 所示。

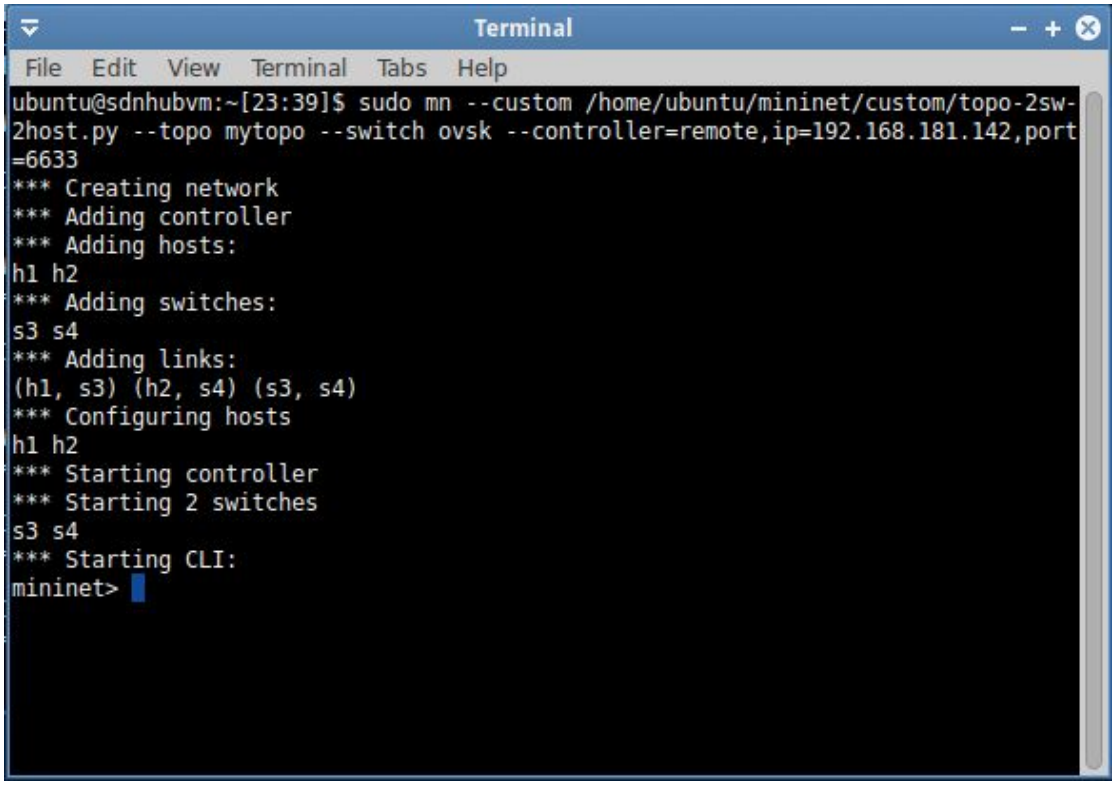

图 1.4：mininet 建立拓扑

打开浏览器输入 http://localhost:8080/ui/index.html，进入图形化的 mininet 可视界面（如图 1.5 和 1.6 所示）。

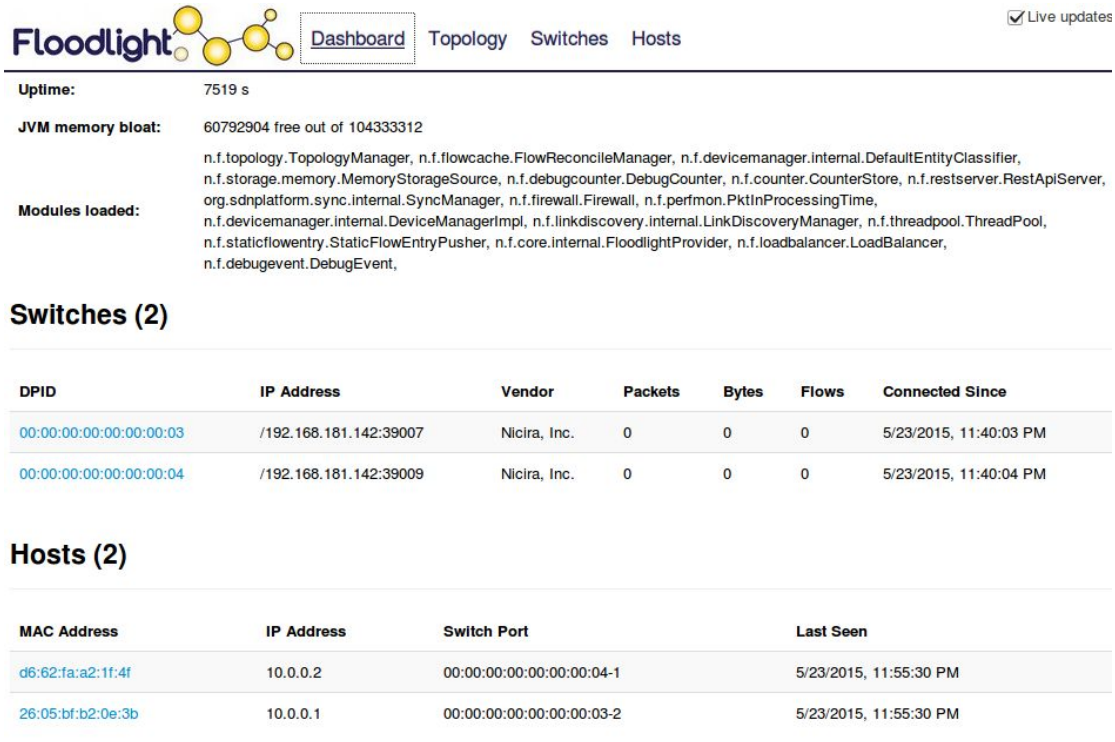

图 1.5：Mininet 图形化界面

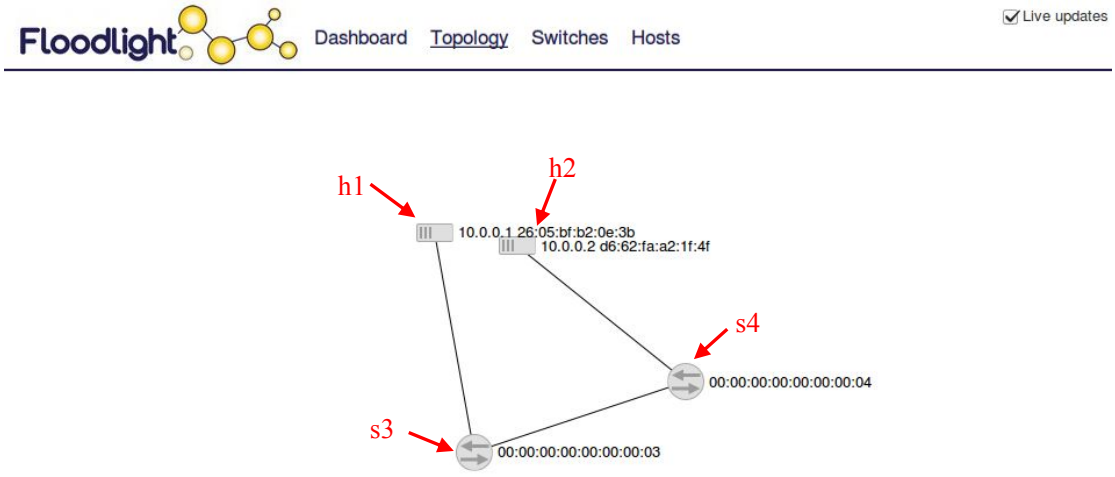

图 1.6：Mininet 图形化拓扑结构界面

（三）测试主机是否 ping 通

在 mininet 中输入以下命令，测试 h1 与 h2 能否 ping 通（如图 1. 7 所示）：

>>pingall

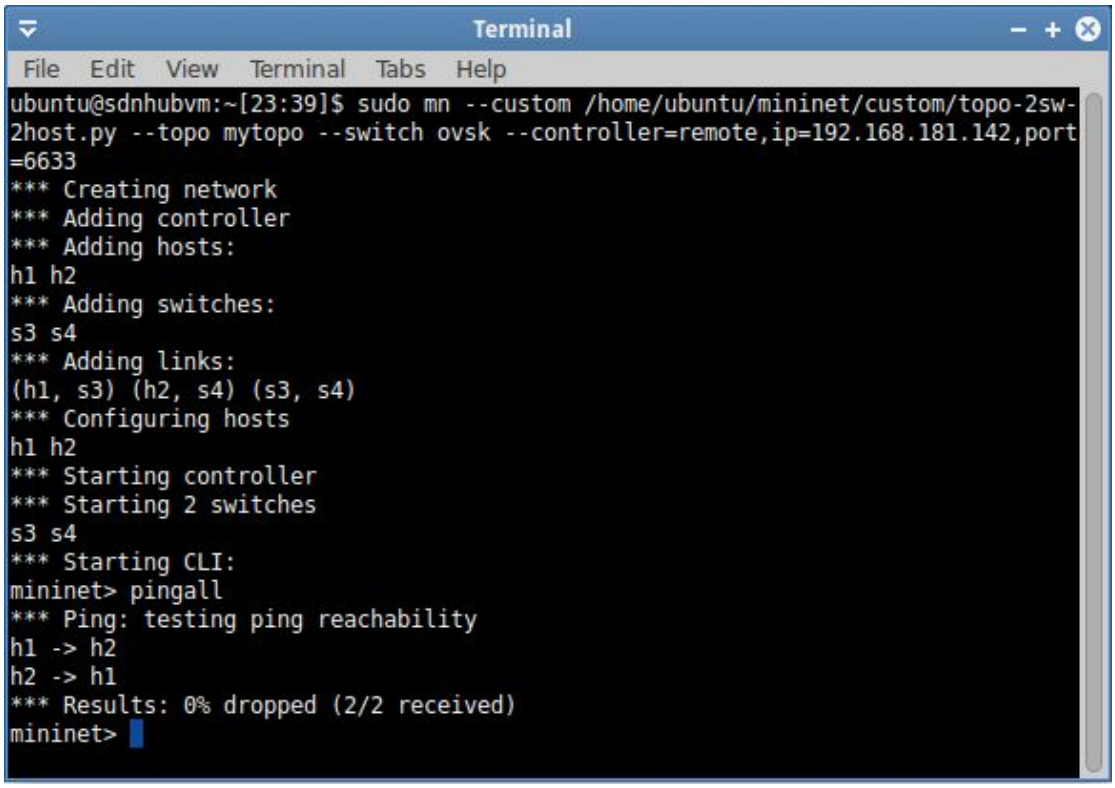

图 1.7：测试 h1 和 h2 的连接

此时丢包率为 0%，即两台主机 h1 与 h2 可以 ping 通。

（四）添加流表使主机间无法 ping 通

在终端中输入以下命令：

>>ovs-ofctl dump-flows s3
>>ovs-ofctl add-flow s3 priority=1,in_port=2,actions=drop
如图 1.8 和图 1.9 所示。图 1.8 中显示的流表为 Controller 添加的临时流表，5s 后会自动失效。

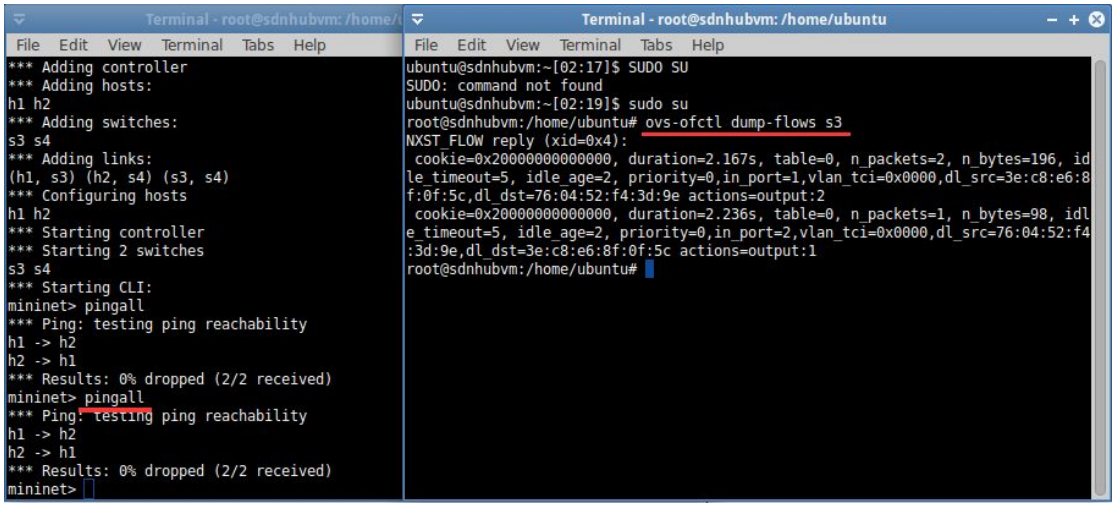

图 1.8：查看交换机 s3 的流表

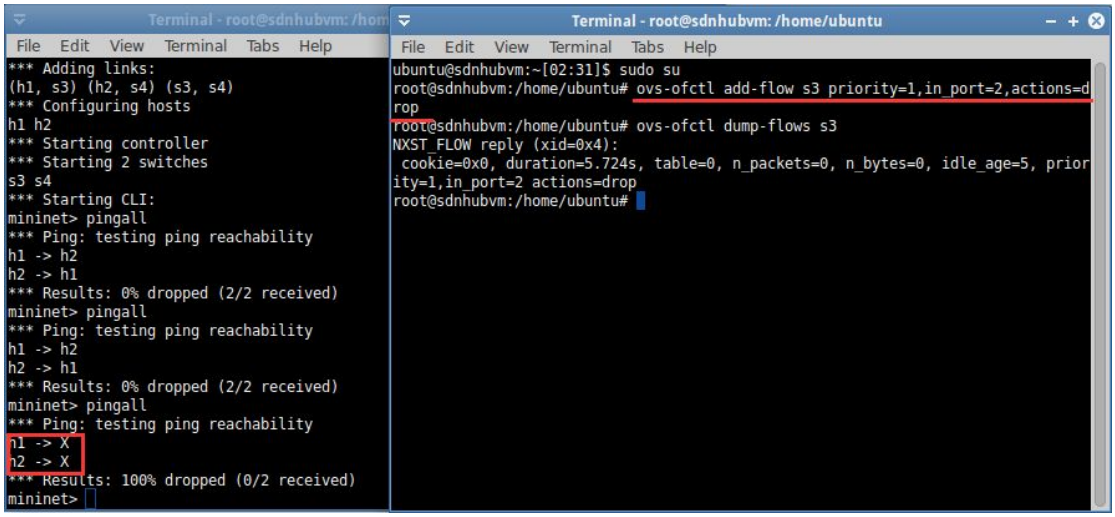

图 1.9：添加流表使 h1，h2 不能连通

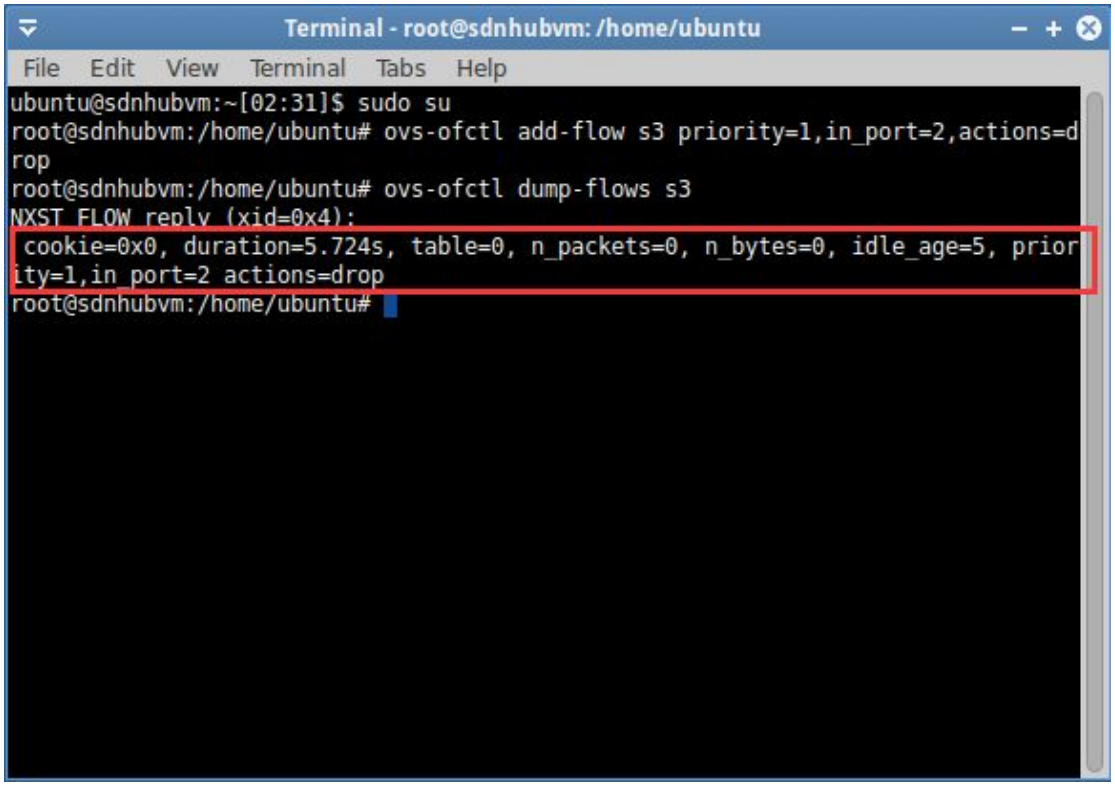

图 1.10：添加的流表信息

由图 1.10 显示，在添加了该流表后，输入 pingall 命令，丢包率达 100%，主机 h1 和 h2 无法 ping 通，实现题目要求。