Lab4: SDN Open Virtual Switches

# Xinmei Jin: xj573

# Zhaoxi Chen: zc1134

# Objectives

* Emulate a functional SDN network.
* Understand and get familiar with OVS.
* Understand and get familiar with controllers.

# Equipment

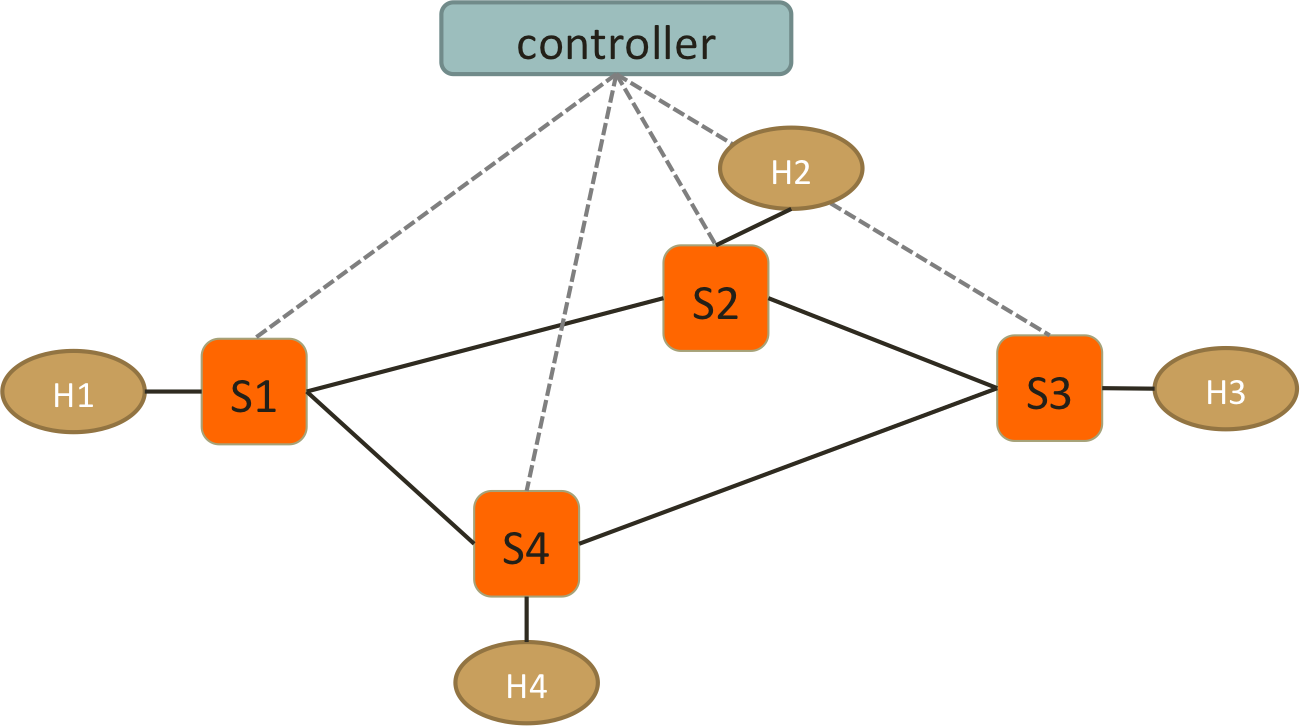
* Computers
* Internet

# References

* RYU programming guide: <http://osrg.github.io/ryu-book/en/html/>

# Experiments

* 1. Use Mininet to create the following topology: (4 Hosts, 4 OVSes ) with a remote controller
  2. Use RYU to implement the controller (you can use other controller such as BEACON, POX, etc...)



* 1. Test Connectivity using ping. (Hint: take care of ARP packets in the controller and install proper rules for them.)
  2. Enforce these policies:
* Everything follows shortest path
* When there are two shortest paths available
  + ICMP and TCP packets take the lower/left path
    - S1-S2-S3 and S2-S3-S4
  + UDP packets take the upper/right path
    - S1-S4-S3 and S2-S1-S4
  + H2 and H4 cannot have HTTP traffic (TCP with port:80)
    - New connections are dropped with a TCP RST sent back to H1 or H3
    - To be more specific, when the first TCP packet (SYN) arrives S1 or S3, forwarded it to controller, controller then create a RST packet and send it back to the host.
  + H1 and H4 cannot have UDP traffic
    - simply drop packets at switches

# Reports

1. **Write a pseudo code to implement spanning tree in SDN network.**

a. T (the final spanning tree) is defined to be the empty set;  
b. For each vertex v of G(the original graph), make the empty set out of v;  
c. Sort the edges of G in ascending (non-decreasing) order;  
d. For each edge (u, v) from the sored list of step 3.  
 If u and v belong to different sets:  
 Add (u,v) to T;  
 Get together u and v in one single set;  
e. Return T

1. **List the advantages of using OpenVSwitch and SDN controller compared to IP networks.**

a.The mobility of state

b.Responding to network dynamics

c.Maintenance of logical tags

d.Hardware integration

e. More granular security.

f.Centralized network provisioning

g.Cloud abstraction

h.Guaranteed content delivery

1. **Include the controller’s code.**

See in attachments(mycontroller.py).

1. **Include the topology file**

See in attachments(mytopo.py).

1. **Describe how you generate traffic to test your controller and switch behavior**

a.ping all

h2 ping h4

h1 ping h4

b.TCP/ICMP: iperf h1 h3

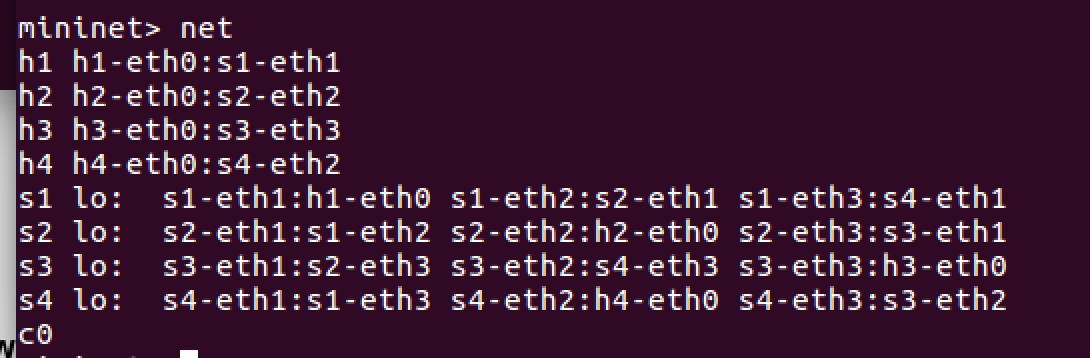
iperf h2 h4

c.UDP: iperfudp bw h1 h3

iperfudp bw h2 h4

1. **Screenshots:**

* **Ping among all the hosts after setting up the platform.**

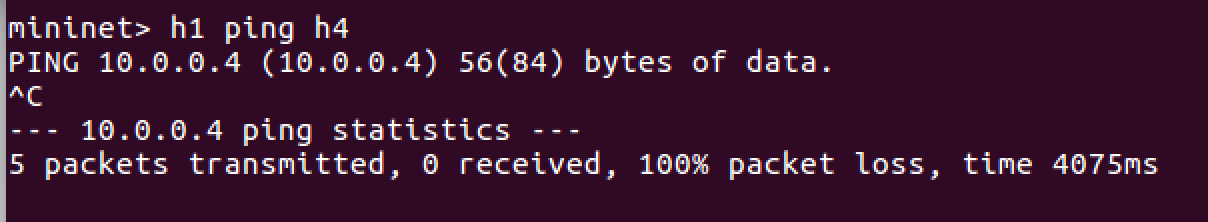
****

****

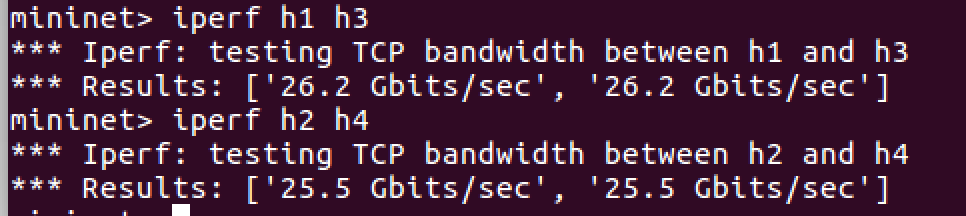
****

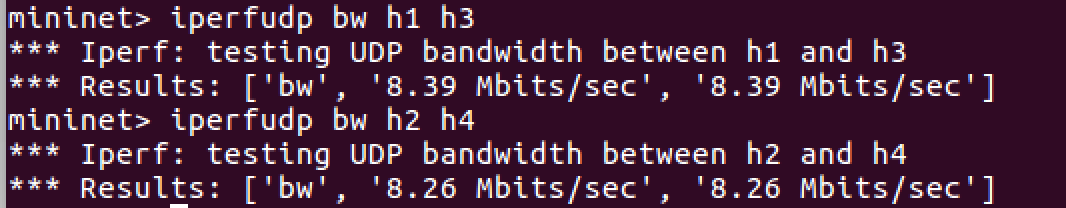
**special case:**

****

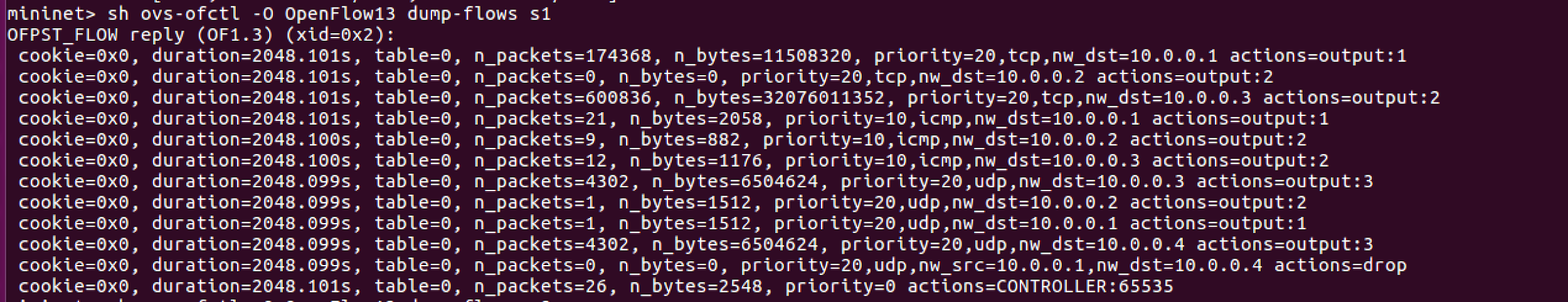
****

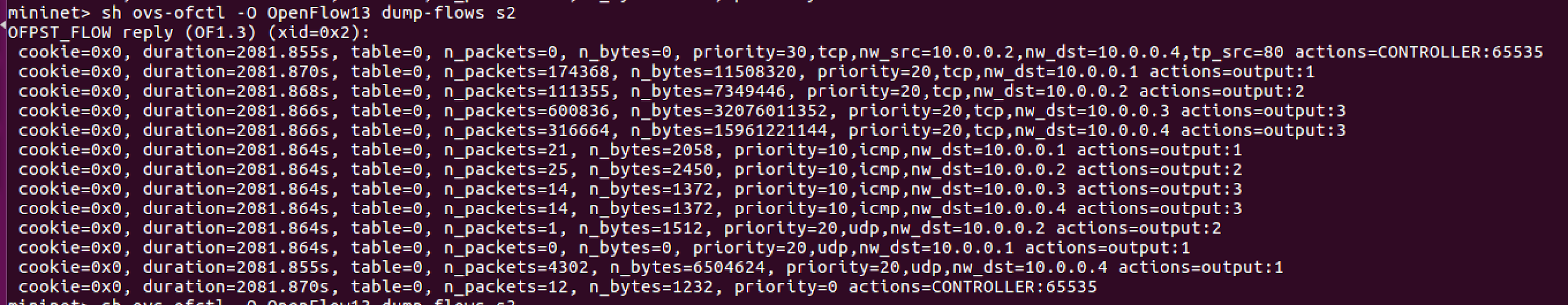
* **TCP, UDP and ICMP packets on their respective paths.**

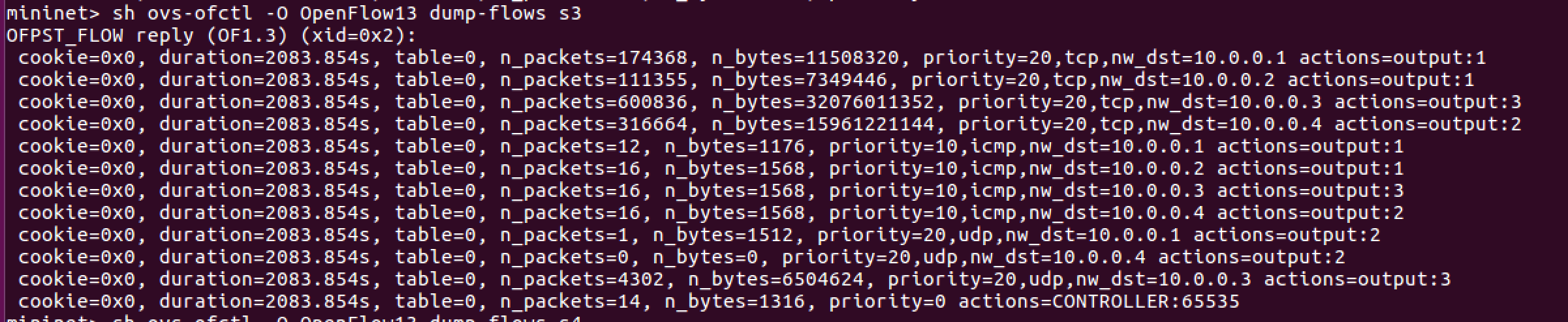
****

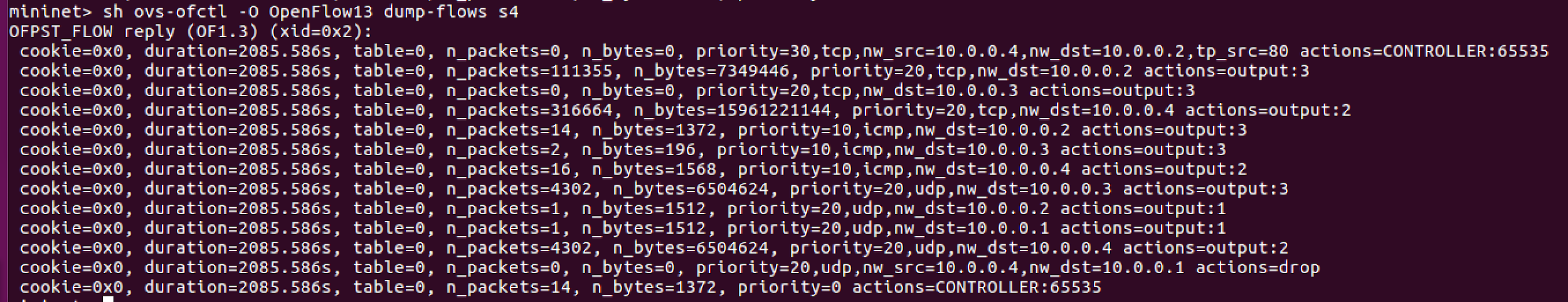
****

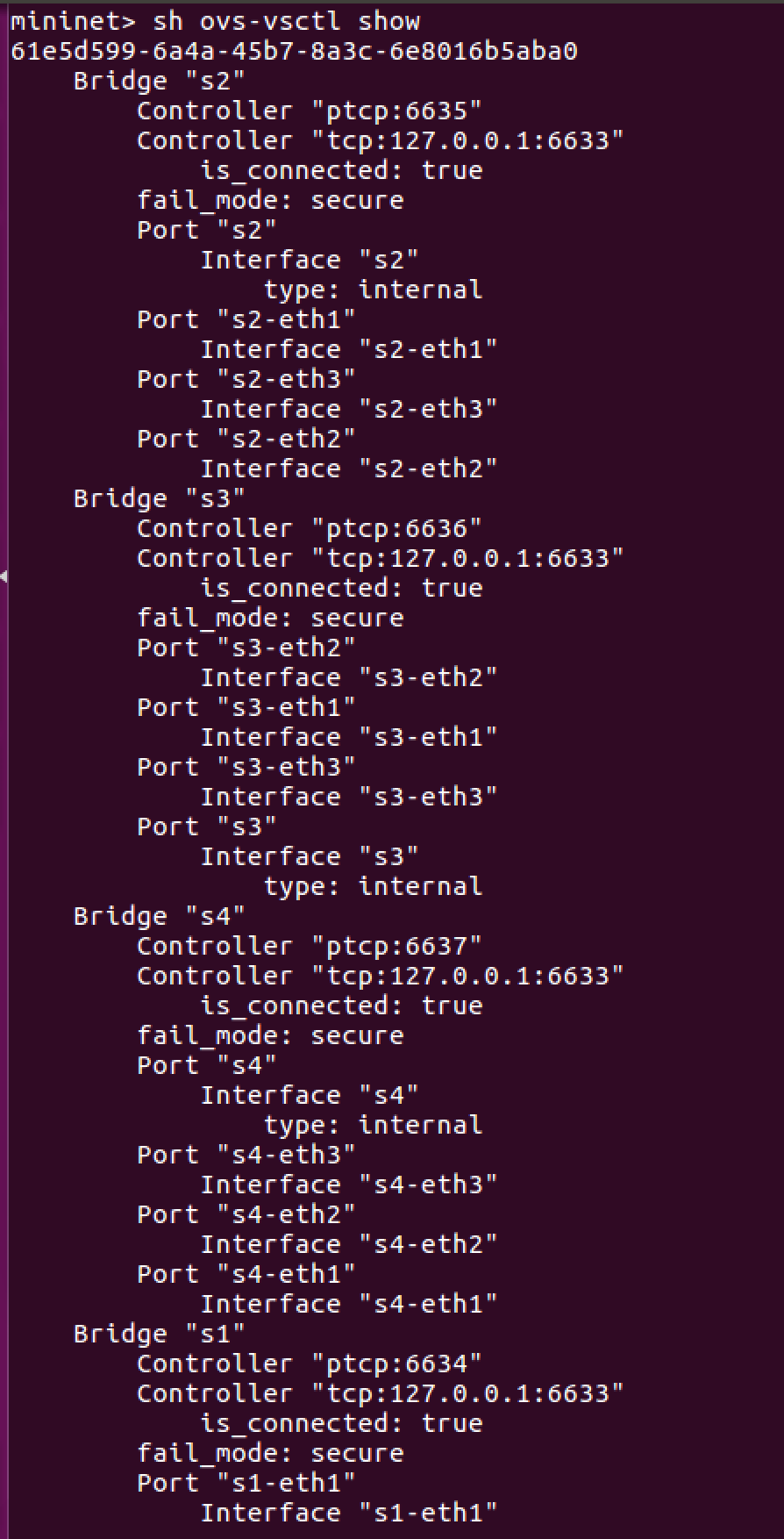
* **Rules installed at each switch.**

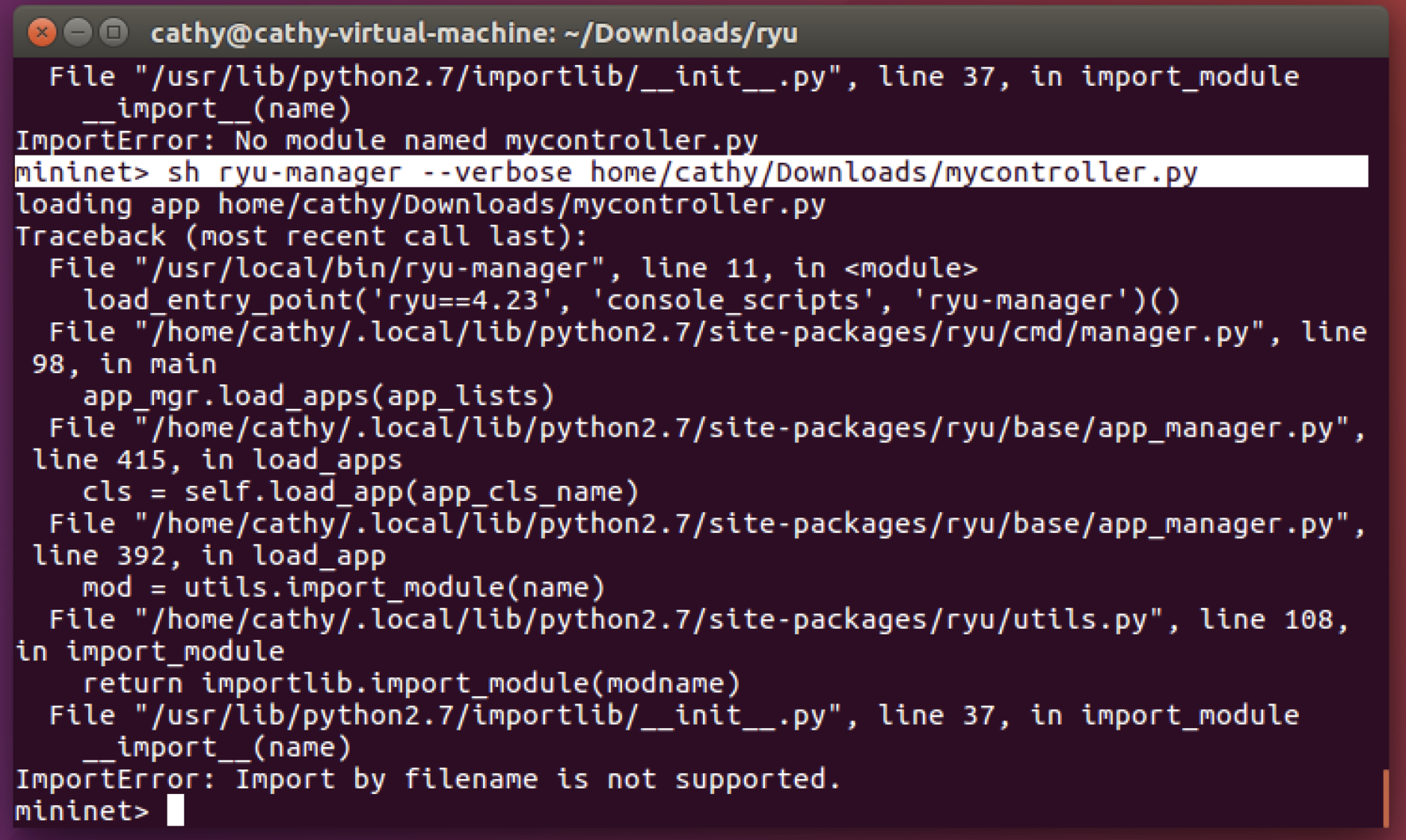
****

****

****

****

****

1. **Challenges you’ve encountered while doing this experiment, and explain how you manage to solve them. If you do not experience any problem, simply say no problems.**

At first, we cannot load our controller. After emailing Frank, we got that it’s easier to load ryu controller in another Terminal, and we need to go to ryu/ryu/app to run it.

**We have zero tolerance to forged or fabricated data!!** A single piece of forged/fabricated data would bring the total score down to zero.