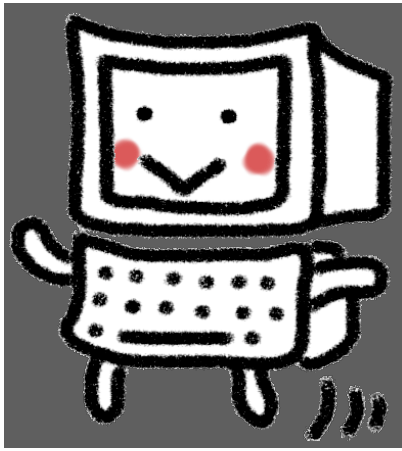


# SDN & NFV
# Lab 5: Group/Meter Entries

*Use Group/Meter in OVS*

Date: 2019/05/02 (Thu.) 15:00

Deadline: 2019/05/12 (Sun.) 23:59

# Group and Meter Types

- Group types
  - ALL
  - INDIRECT
  - SELECT
  - FAILOVER
- Meter types
  - DROP
  - DSCP Remark

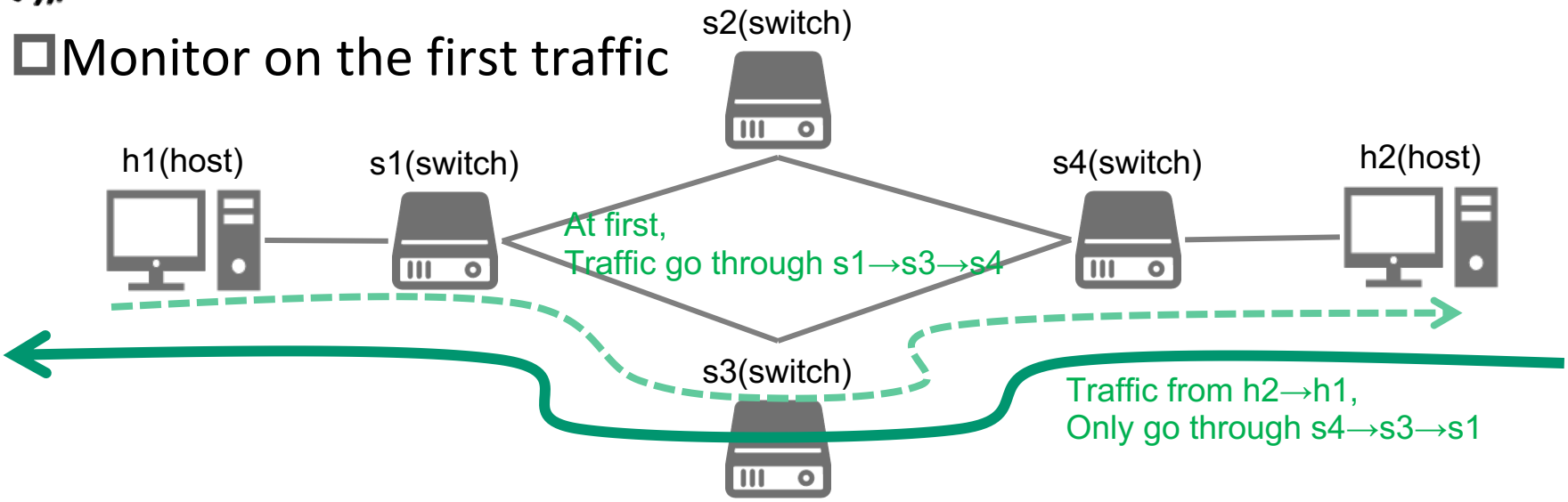
# Outline

- Use group entries in OVS
  - Observe group type “SELECT” behaviors
  - Learn to monitor the traffic
    - Use OVS commands
    - Use ping, iperf udp, iperf tcp tools
- Use meter entries in OVS
  - Prepare a proper environment for testing meter entries
    - Upgrade OVS version
  - Make traffic going through userspace
  - Learn to monitor the traffic
    - Use iperf udp, iperf tcp tools
- Requirements

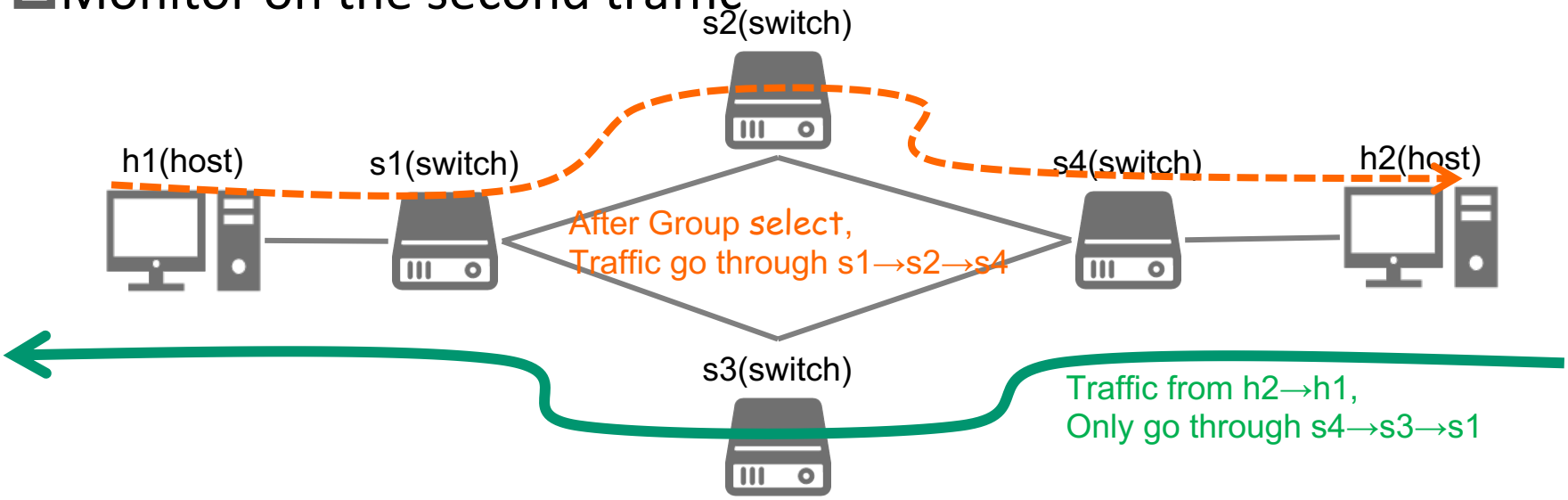
# Use Group Entries in OVS

# Overview

- Monitor on the first traffic

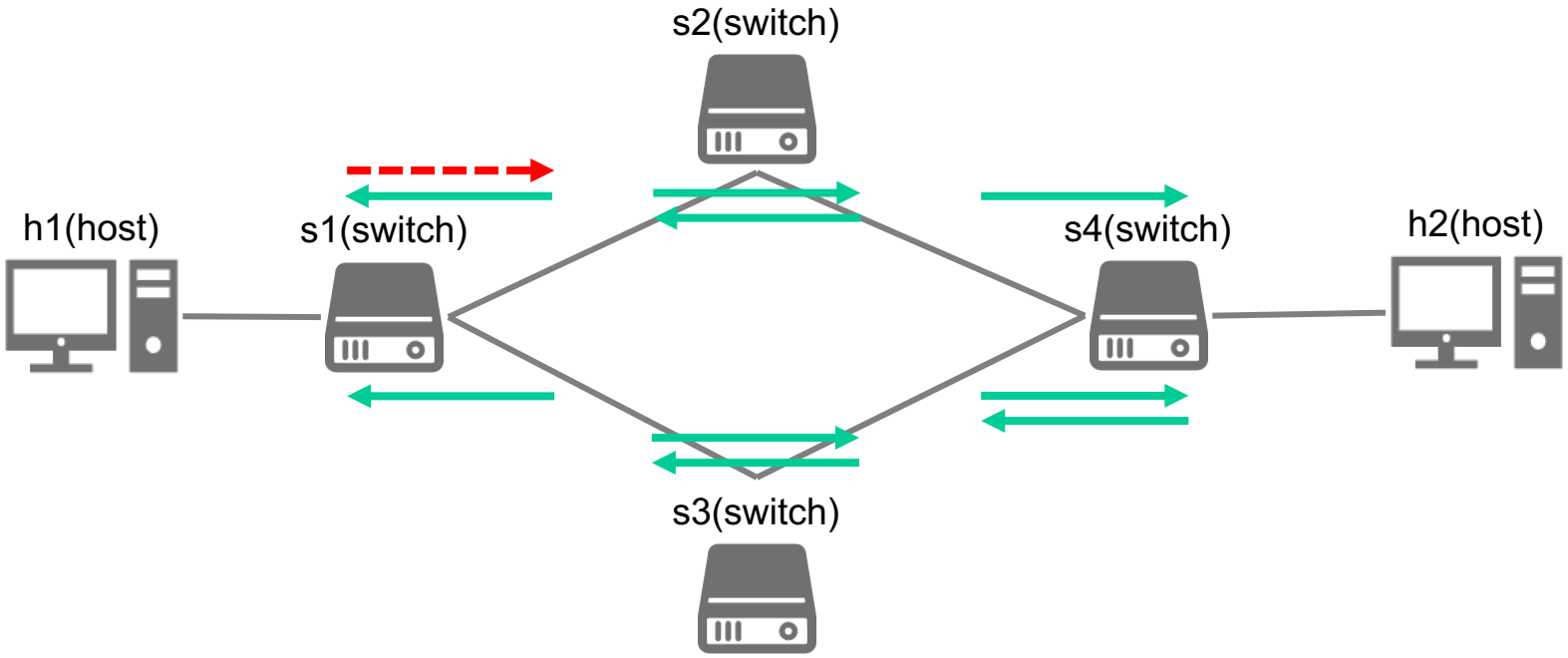



- Monitor on the second traffic



# Create a ring topology

- TA would offer a python script to create a ring topology




- Add Flow and Group Entries in OVS
  - # of flow entries: 9 (——→)
  - # of entries: 1 flow entry redirect to group + 1 group entry (- - - - →)

# Flow/Group Entry

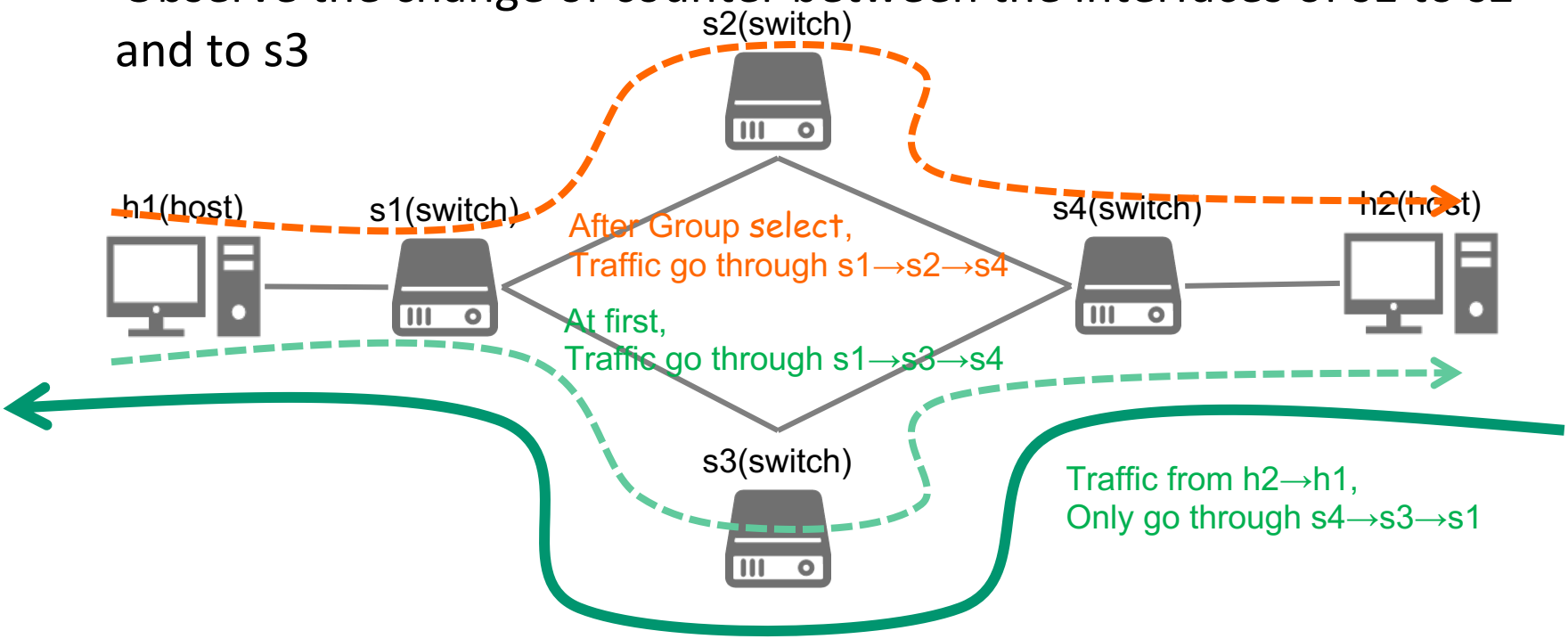
- Flow Entries
  - Use priority carefully, or it would cause flow rule override
  - Before adding a flow entry which outputs to a group entry, make sure the group entry has added already
- Group Entries
  - Group type as "SELECT"
  - In this lab, make group buckets output to s2 and s3 for selecting

# Monitor the traffic

- Choose two ways from ping, iperf tcp, iperf udp tool
  - "SELECT" type would select paths by hashing fields of the flow
- Monitor TX counter on s1 interface

```
Mininet> sh ovs-ofctl dump-ports -O OpenFlow13 s1
```

  - Observe the change of counter between the interfaces of s1 to s2 and to s3



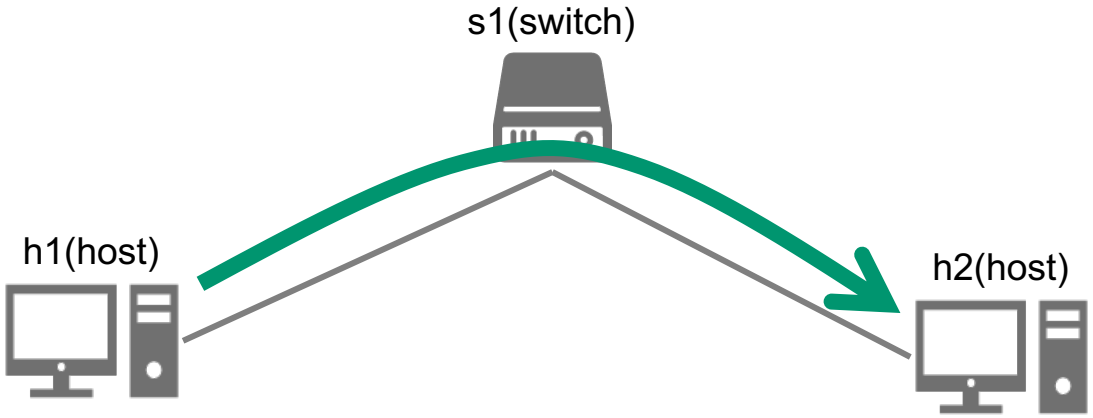
# Reference

- Ryu
  - https://ryu.readthedocs.io/en/latest/app/ofctl_rest.html#add-a-group-entry
- ONOS
  - Use ONOS web Document
  - http://<your-onos-ip>:8181/onos/v1/docs

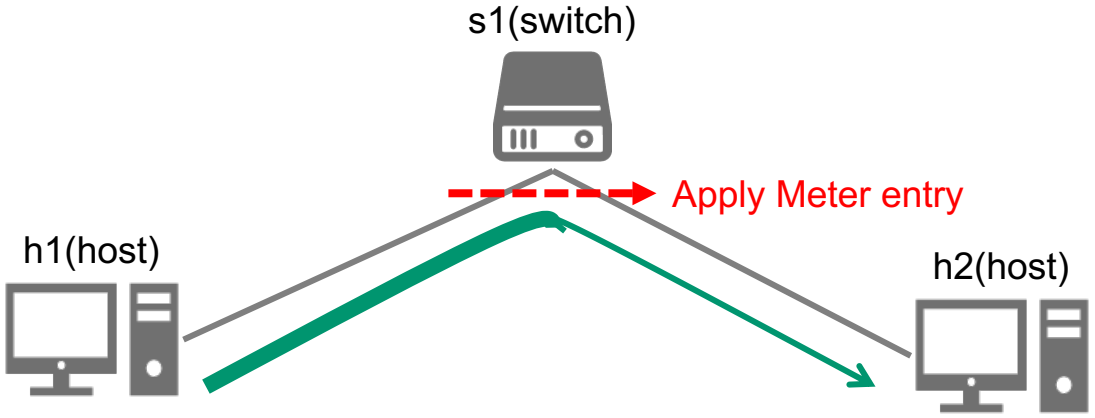
# Use Meter Entries in OVS

# Overview

- h1 sends traffic to h2 without meter rules




- h1 sends traffic to h2 with meter rules



# Upgrade OVS version up to 2.8.1

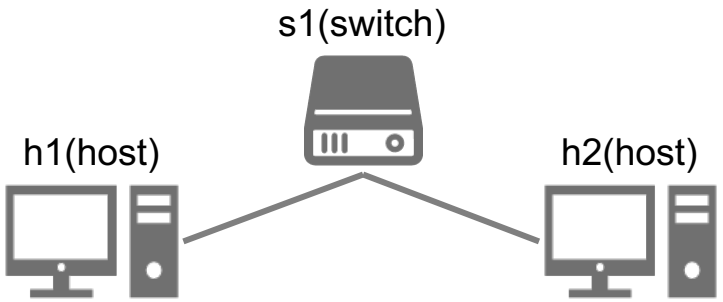
- OVS supports meter after version 2.8.0 (2.8.1 suggested)

```
ohmygodorz@lab5:~$ sudo ovs-vsctl show
5d54c57e-fd77-4019-88c9-193020e41243
        Bridge "s1"
        Controller "ptcp:6654"
        Controller "tcp:192.168.1.161:6653"
        is_connected: true
        fail_mode: secure
        Port "s1"
        Interface "s1"
                type: internal
        Port "s1-eth1"
        Interface "s1-eth1"
        Port "s1-eth2"
        Interface "s1-eth2"
        ovs_version: "2.8.1"
```

- ✓Ref: Installing new version of Open vSwitch

# Create a topology and configure OVS

- Use "--topo single,2" for creating the topology




- Use ovs-vswitchd in userspace mode, create a bridge with datapath_type=netdev in the configuration database

```
mininet> s1 ovs-vsctl set bridge "s1" datapath_type=netdev
```

- Only when max_meter>0, meter entries could be added to OVS

```
mininet> s1 ovs-ofctl -O OpenFlow13 meter-features "s1"
OFPST_METER_FEATURES reply (OF1.3) (xid=0x2):
max_meter:65536 max_bands:8 max_color:0
band_types: drop
capabilities: kbps pktps burst stats
```

# Add rule entries (Take ONOS as example)

- Add a meter entry in OVS to limit rate down to 1Mbps

```
$ curl -X POST --user onos:rocks -H "content-type:application/json"
http://<controller ip>:8181/onos/v1/meters/<device id> -d
@<meterEntry.json>
```

- Then add a flow in OVS to use the above meter entry

```
$ curl -X POST --user onos:rocks -H "content-type:application/json"
http://<controller ip>:8181/onos/v1/flows -d @<flowEntry.json>
```

# Monitor traffic by using “iperf”

- Use h1 as iperf client and h2 as iperf server
  - In order to highlight the effect of speed limit, the client should send a larger traffic to server
- Use UDP packets since TCP would do congestion/flow control
- Observe if the traffic speed is limited down to 1Mbps

# Requirements

# Requirements

- Use group entries in OVS
  - Install “SELECT” group rules and corresponding flow rules
  - Observe if two flows are split into two paths
- Use meter entries in OVS
  - Setup the environment for testing meter entries
  - Install meter rules and corresponding flow rules
  - Observe if the traffic speed is limited

# Submit & Demo

# Submit to e3

- Files
  - A report: **lab5_studentID.pdf**
    - Record your process (especially your command)
    - Take screenshots and explain or mark them
      - Group SELECT: Monitoring TX counter
      - Meter: Limiting bandwidth under 1Mbps
    - Write down what you learn or solve
- Submit
  - Upload the report file to e3
  - Wrong file name or format would not be scored
- Demo
  - Show your results
  - TA would asks some related questions

Q & A

# Thank you