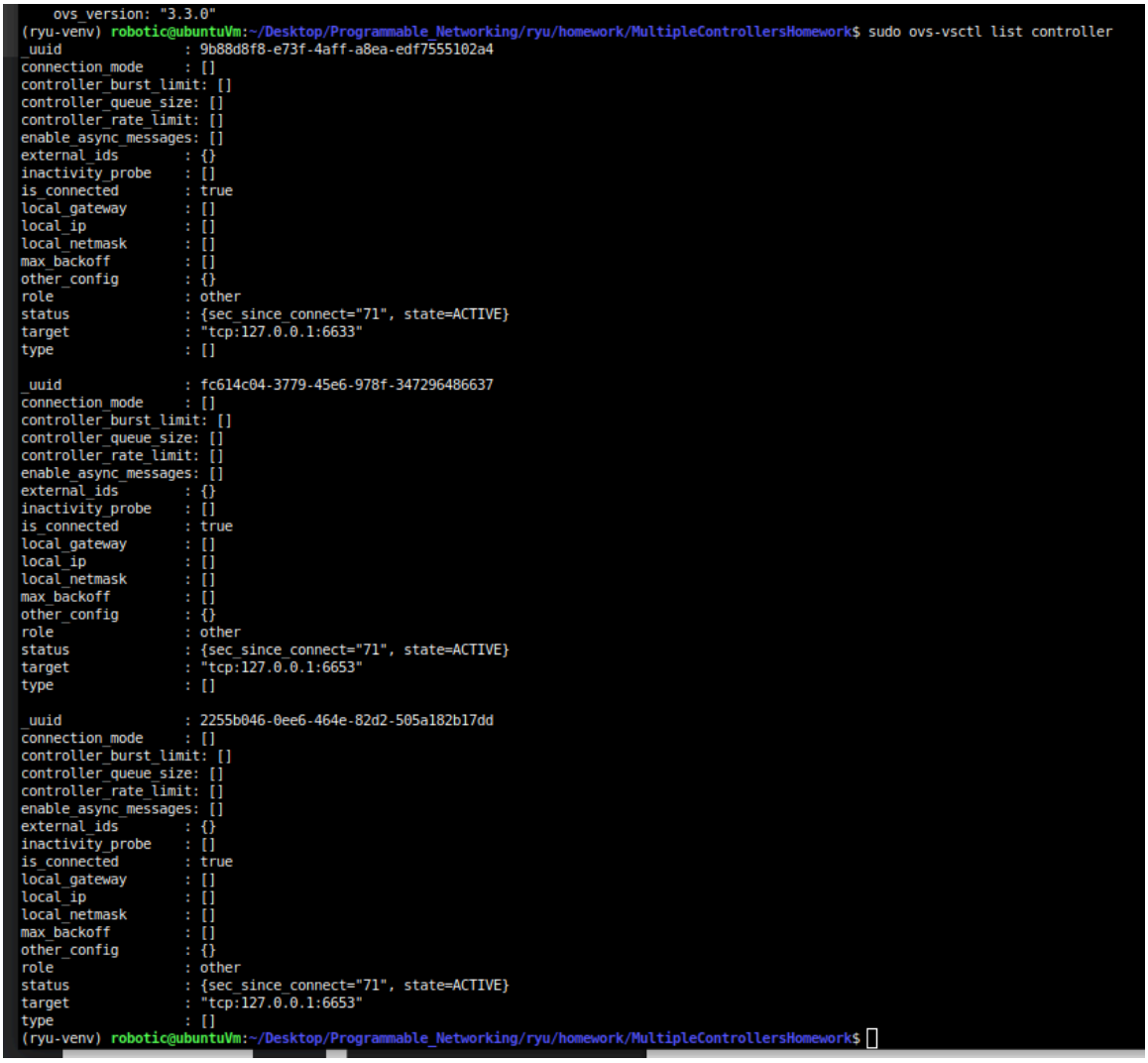
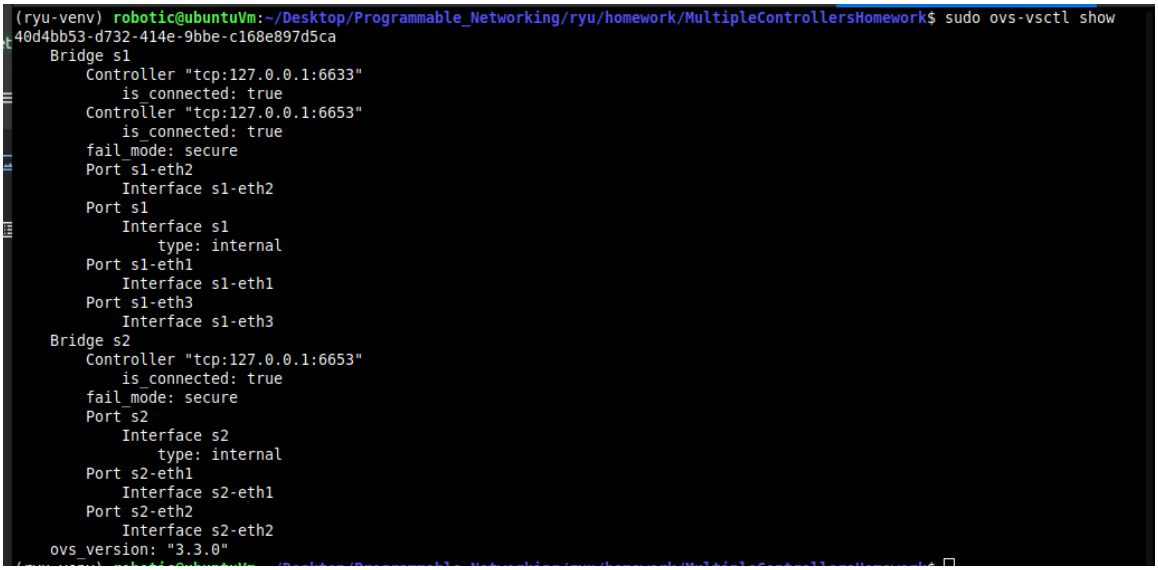
Multiple Controllers Homework

1. Default Setup (Step 1)

- La configurazione iniziale prevede due controller collegati a uno switch in modalità EQUAL.
- Lo switch è collegato:
  - al primo controller sulla porta 6633
  - al secondo controller sulla porta 6653





2. Configurazione Ruoli (Step 2-3)

Esecuzione dei comandi per impostare i ruoli:

```
# Imposta C1 come SLAVE per dpid 1
curl -X POST -d '{
    "dpid": 1,
    "role": "SLAVE"
}' http://localhost:8081/stats/role

# Imposta C0 come MASTER per dpid 1
curl -X POST -d '{
    "dpid": 1,
    "role": "MASTER"
}' http://localhost:8080/stats/role

# Imposta C1 come MASTER per dpid 2
curl -X POST -d '{
    "dpid": 2,
    "role": "MASTER"
}' http://localhost:8081/stats/role
```

```
(ryu-venv) robotic@ubuntuVm:~/Desktop/Programmable_Networking/ryu/homework/MultipleControllersHomework$ sudo ovs-vsctl list controller
[sudo] password for robotic:
_uuid               : 9b88d8f8-e73f-4aff-a8ea-edf7555102a4
connection_mode     : []
controller_burst_limit: []
controller_queue_size: []
controller_rate_limit: []
enable_async_messages: []
external_ids        : {}
inactivity_probe    : []
is_connected        : true
local_gateway       : []
local_ip            : []
local_netmask       : []
max_backoff         : []
other_config        : {}
role                : master
status              : {sec_since_connect="18480", state=ACTIVE}
target              : "tcp:127.0.0.1:6633"
type                : []

_uuid               : fc614c04-3779-45e6-978f-347296486637
connection_mode     : []
controller_burst_limit: []
controller_queue_size: []
controller_rate_limit: []
enable_async_messages: []
external_ids        : {}
inactivity_probe    : []
is_connected        : true
local_gateway       : []
local_ip            : []
local_netmask       : []
max_backoff         : []
other_config        : {}
role                : slave
status              : {sec_since_connect="18480", state=ACTIVE}
target              : "tcp:127.0.0.1:6653"
type                : []

_uuid               : 2255b046-0ee6-464e-82d2-505a182b17dd
connection_mode     : []
controller_burst_limit: []
controller_queue_size: []
controller_rate_limit: []
enable_async_messages: []
external_ids        : {}
inactivity_probe    : []
is_connected        : true
local_gateway       : []
local_ip            : []
local_netmask       : []
max_backoff         : []
other_config        : {}
role                : master
status              : {sec_since_connect="18480", state=ACTIVE}
target              : "tcp:127.0.0.1:6653"
type                : []
(ryu-venv) robotic@ubuntuVm:~/Desktop/Programmable_Networking/ryu/homework/MultipleControllersHomework$
```

4. Test di Connettività (Step 4)

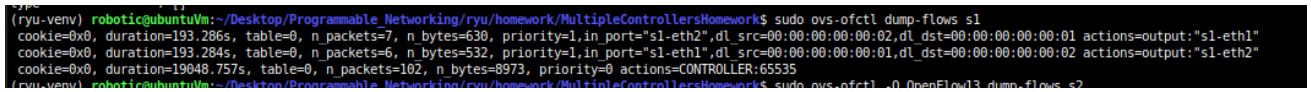
```
*** Configuring hosts
host1 host2 host3
*** Starting controller

*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> host1 ping host2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=6.41 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.162 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.172 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.146 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.159 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5086ms
rtt min/avg/max/mdev = 0.146/1.350/6.414/2.287 ms
mininet>
```

A e b: Ping da H1 a H2 funziona.

Motivazione:
- h1 e h2 si trovano nello stesso dominio di broadcast (collegati a s1).
- s1 è gestito da C0 che è MASTER.
- Il controller MASTER installa le regole di forwarding, quindi i pacchetti ICMP non devono attraversare s2 né passare per C0.
- La comunicazione resta confinata a s1 sotto il controllo di C0.

OpenFlow Rules su s1:

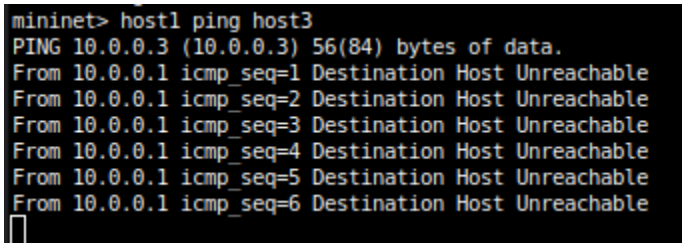


5. Step 5 (stop c0 controller)

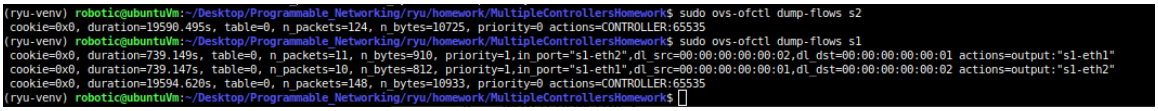
1. Ping funziona
2. Perché: la flow table è già popolata su s1, quindi i pacchetti seguono le regole installate.

6. Step 6

- Ping da h1 a h3: non funziona



Motivazione:
- s1 non ha un controller MASTER nessuna nuova regola installata per inoltrare pacchetti verso s2.
- Le flow table mostrano regole solo per h1–h2 su s1, nessuna per h1–h3.



- s2 non riceve pacchetti da h1.

7. Step 7 – Cambio Ruolo

Setto a MASTER il C1 per S1

```
curl -X POST -d '{
   "dpid": 1,
   "role": "MASTER"
}' http://localhost:8081/stats/role
```

8. Step 8 – Nuovo Test

1. Ping tra h1 e h3 funziona

```
mininet> host1 ping host3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.391 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.262 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.051 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.059 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=0.182 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=0.088 ms
```

2. Motivazione:

- Ora C1 è MASTER su s1 e quindi inietta nuove regole di forwarding su s1..
- I pacchetti vengono correttamente instradati ed infatti nello screenshot si vedono le regole iniettate dal controller

```
(ryu-venv) robotic@ubuntuVm:~/Desktop/Programmable_Networking/ryu/homework/MultipleControllersHomework$ curl -X POST -d '{
    "dpid": 1,
    "role": "MASTER"
 }' http://localhost:8081/stats/role
(ryu-venv) robotic@ubuntuVm:~/Desktop/Programmable_Networking/ryu/homework/MultipleControllersHomework$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=1063.635s, table=0, n_packets=11, n_bytes=910, priority=1,in_port="s1-eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-e
 cookie=0x0, duration=1063.633s, table=0, n_packets=10, n_bytes=812, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-e
 cookie=0x0, duration=44.953s, table=0, n_packets=19, n_bytes=1806, priority=1,in_port="s1-eth3",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s1-et
 cookie=0x0, duration=44.951s, table=0, n_packets=18, n_bytes=1708, priority=1,in_port="s1-eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s1-et
 cookie=0x0, duration=19919.106s, table=0, n_packets=151, n_bytes=11115, priority=0 actions=CONTROLLER:65535
(ryu-venv) robotic@ubuntuVm:~/Desktop/Programmable_Networking/ryu/homework/MultipleControllersHomework$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=48.028s, table=0, n_packets=19, n_bytes=1806, priority=1,in_port="s2-eth1",dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01 actions=output:"s2-et
 cookie=0x0, duration=48.023s, table=0, n_packets=18, n_bytes=1708, priority=1,in_port="s2-eth2",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:"s2-et
 cookie=0x0, duration=19922.170s, table=0, n_packets=128, n_bytes=10977, priority=0 actions=CONTROLLER:65535
(ryu-venv) robotic@ubuntuVm:~/Desktop/Programmable_Networking/ryu/homework/MultipleControllersHomework$
```