

Part1

1.

```
try <command> --help for more information.
hcgcarry@root > apps -a -s
* 30 org.onosproject.drivers 2.2.0 Default Drivers
* 31 org.onosproject.optical-model 2.2.0 Optical Network Model
* 66 org.onosproject.hostprovider 2.2.0 Host Location Provider
* 79 org.onosproject.gui2 2.2.0 ONOS GUI2
* 103 org.onosproject.lldpprovider 2.2.0 LLDP Link Provider
* 104 org.onosproject.openflow-base 2.2.0 OpenFlow Base Provider
* 105 org.onosproject.openflow 2.2.0 OpenFlow Provider Suite
* 140 org.onosproject.fwd 2.2.0 Reactive Forwarding
hcgcarry@root >
```

```
hcgcarry@root > apps -a -s
* 30 org.onosproject.drivers 2.2.0 Default Drivers
* 79 org.onosproject.gui2 2.2.0 ONOS GUI2
* 140 org.onosproject.fwd 2.2.0 Reactive Forwarding
hcgcarry@root >
```

Org.onosproject.optical-model

Org.onosproject.hostprovider

org.onosproject.lldpprovider

org.onosproject.openflow-base

org.onosproject.openflow

2. 可以,因為 H1,H2 有 s1 link 起來,並且目前有安裝 org.onosproject.fwd 的情況下 flow 就會被 install 到 data-plane 就可以傳遞訊息了

3. Port :6653

4. Org.onosproject.openflow-base

```
hcgcarry@root > apps -a -s
* 30 org.onosproject.drivers 2.2.0 Default Drivers
* 31 org.onosproject.optical-model 2.2.0 Optical Network Model
* 66 org.onosproject.hostprovider 2.2.0 Host Location Provider
* 79 org.onosproject.gui2 2.2.0 ONOS GUI2
hcgcarry@root >

*** Done
completed in 25.837 seconds
hcgcarry@hcgcarry-SDN:~$ sudo mn --topo=linear,3 --controller=remote,127.0.0.1:6653
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
*** Adding hosts:
```

```

hcgcarry@root > apps -a -s
* 30 org.onosproject.drivers 2.2.0 Default Drivers
* 31 org.onosproject.optical-model 2.2.0 Optical Network Model
* 66 org.onosproject.hostprovider 2.2.0 Host Location Provider
* 79 org.onosproject.gui2 2.2.0 ONOS GUI2
* 104 org.onosproject.openflow-base 2.2.0 OpenFlow Base Provider
hcgcarry@root >

h1 h2 h3
*** Done
completed in 57.358 seconds
hcgcarry@hcgcarry-SDN:~$ sudo mn --topo=linear,3 --controller=remote,127.0.0.1:6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3

```

Part2

```

hcgcarry@hcgcarry-SDN:~/hw/hw1$ sudo mn --custom=project1_part2_309551111.py --topo=topo_part2_309551111 --controller=remote,ip=127.0.0.1:6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s1, s4) (s2, s4) (s3, s4)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=14586>
<Host h2: h2-eth0:10.0.0.2 pid=14588>
<Host h3: h3-eth0:10.0.0.3 pid=14590>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=14595>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=14598>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=14601>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=14604>
<RemoteController{'ip': '127.0.0.1:6653'} c0: 127.0.0.1:6653 pid=14580>

```

```

mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)

```

Part3

```

mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=12905>
<Host h2: h2-eth0:192.168.0.2 pid=12907>
<Host h3: h3-eth0:192.168.0.3 pid=12909>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=12914>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=12917>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=12920>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=12923>
<RemoteController{'ip': '127.0.0.1', 'port': 6653} c0: 127.0.0.1:6653 pid=12899>
mininet>

```

```

mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)

```

Part4:

學到的東西:Onos ,mininet 的熟悉