SDN Project1 Answer Sheet

Name: 陳正宗 Student ID: 0756823 Department: 資訊組

Part 1: Answer Questions

1. Activate ONOS APPs

1-1. When activating **“org.onosproject.openflow”**, what are the APPs which also be activated?

1-2. Which **APP** enables hosts to ping each other?

Hint: Please refer to the reference “Basic ONOS Tutorial”[1] attached at the end of slide

2. Observe listening port

2-1. Openflow protocol defines the TCP port for connection between controller and switch. What is the **number** of this port?

2-2. Regarding to the previous question, which **APP** enables that TCP port be listening?

Hint: Observation of network connection

1. bring up a new terminal
2. use the command “netstat” to print network connection

```
$ netstat -tnlp      #show only listening TCP sockets
```

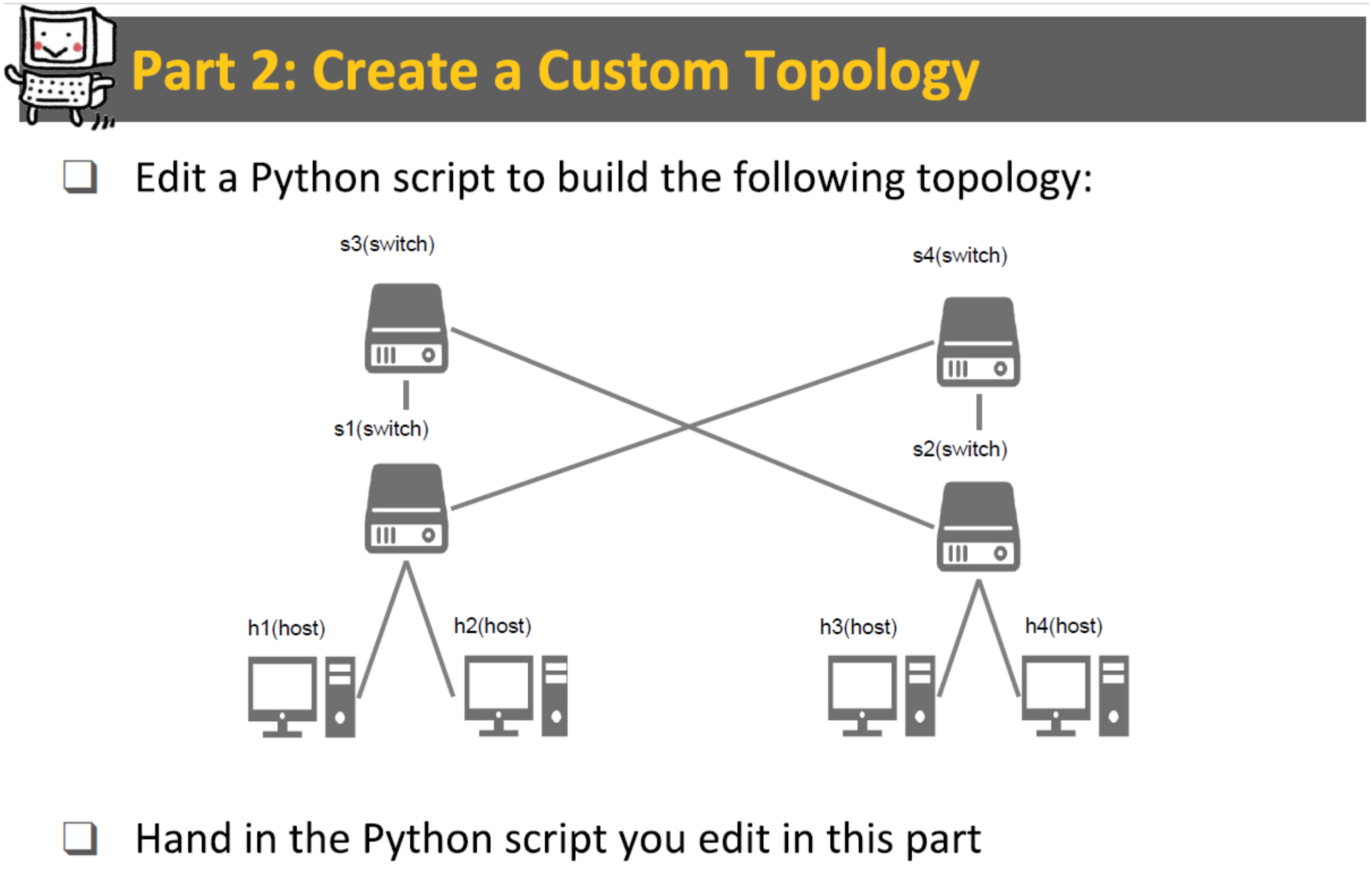
Ans:

1-1	org.onosproject.optical-model	(Optical Network Model)
	org.onosproject.hostprovider	(Host Location Provider)
	org.onosproject.lldpprovider	(LLDP Link Provider)
	org.onosproject.openflow-base	(OpenFlow Base Provider)
	org.onosproject.openflow	(OpenFlow Provider Suite)
1-2	org.onosproject.fwd	(Reactive Forwarding)

Ans:

2-1 The port number is 6653 and 6633.

2-2 org.onosproject.openflow-base (OpenFlow Base Provider)



Part 2: Create a Custom Topology

□ Edit a Python script to build the following topology:

□ Hand in the Python script you edit in this part

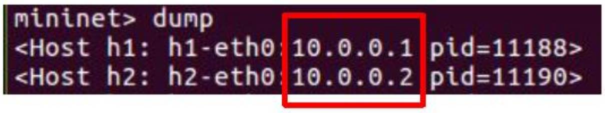
Ans:

Python script as below,

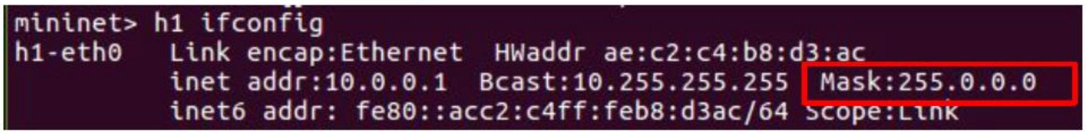
```
from mininet.topo import Topo

class Project1_Topo_0756823( Topo ):
    def __init__( self ):
        Topo.__init__( self )

        # Add hosts
        h1 = self.addHost( 'h1' )
        h2 = self.addHost( 'h2' )
        h3 = self.addHost( 'h3' )
        h4 = self.addHost( 'h4' )

        # Add switches
        s1 = self.addSwitch( 's1' )
        s2 = self.addSwitch( 's2' )
        s3 = self.addSwitch( 's3' )
        s4 = self.addSwitch( 's4' )
        # Add links
        self.addLink( s4, s1 )
        self.addLink( s3, s1 )
        self.addLink( s3, s2 )
        self.addLink( s4, s2 )

        self.addLink( h1, s1 )
        self.addLink( h2, s1 )
        self.addLink( h3, s2 )
        self.addLink( h4, s2 )


topos = { 'topo_0756823': Project1_Topo_0756823 }
```

Bonus: Manually Assign Hosts IP Address In Mininet (I)

- ❑ By default, Mininet will automatically assign an ip address and a subnet mask to each interface of each host
  (i.e. 10.0.0.1/8, 10.0.0.2/8, 10.0.0.3/8)

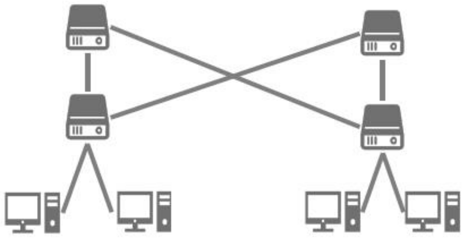


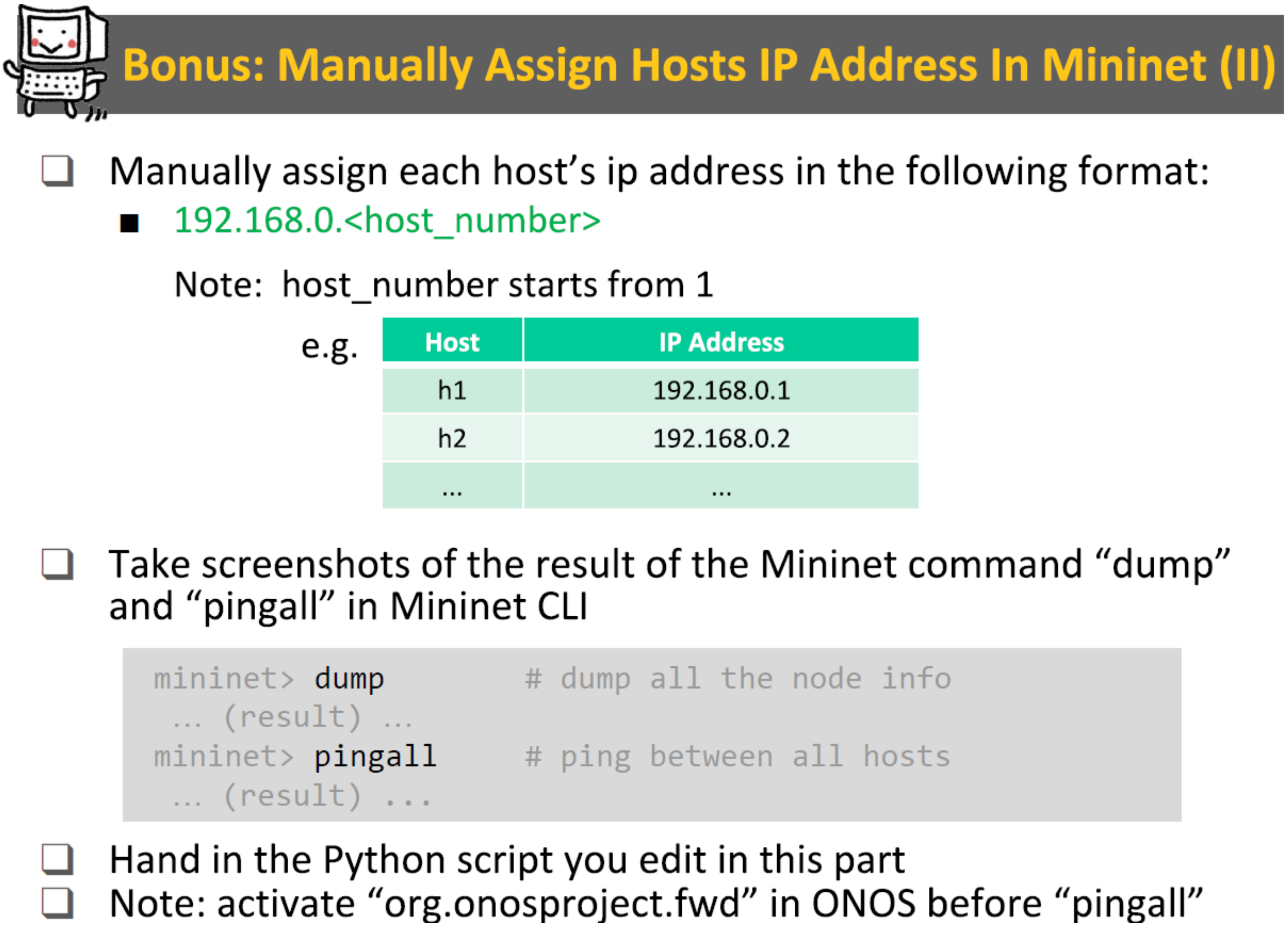
```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=11188>
<Host h2: h2-eth0:10.0.0.2 pid=11190>
```



```
mininet> h1 ifconfig
h1-eth0   Link encap:Ethernet  HWaddr ae:c2:c4:b8:d3:ac
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::acc2:c4ff:feb8:d3ac/64 Scope:Link
```

- ❑ Reuse the topology created in part 2





Bonus: Manually Assign Hosts IP Address In Mininet (II)

- ❑ Manually assign each host's ip address in the following format:
  - ■ 192.168.0.<host_number>

    Note: host_number starts from 1

    e.g.

Host	IP Address
h1	192.168.0.1
h2	192.168.0.2
…	…

- ❑ Take screenshots of the result of the Mininet command "dump" and "pingall" in Mininet CLI

```
mininet> dump          # dump all the node info
 … (result) …
mininet> pingall       # ping between all hosts
 … (result) ...
```

- ❑ Hand in the Python script you edit in this part
- ❑ Note: activate "org.onosproject.fwd" in ONOS before "pingall"

Ans:

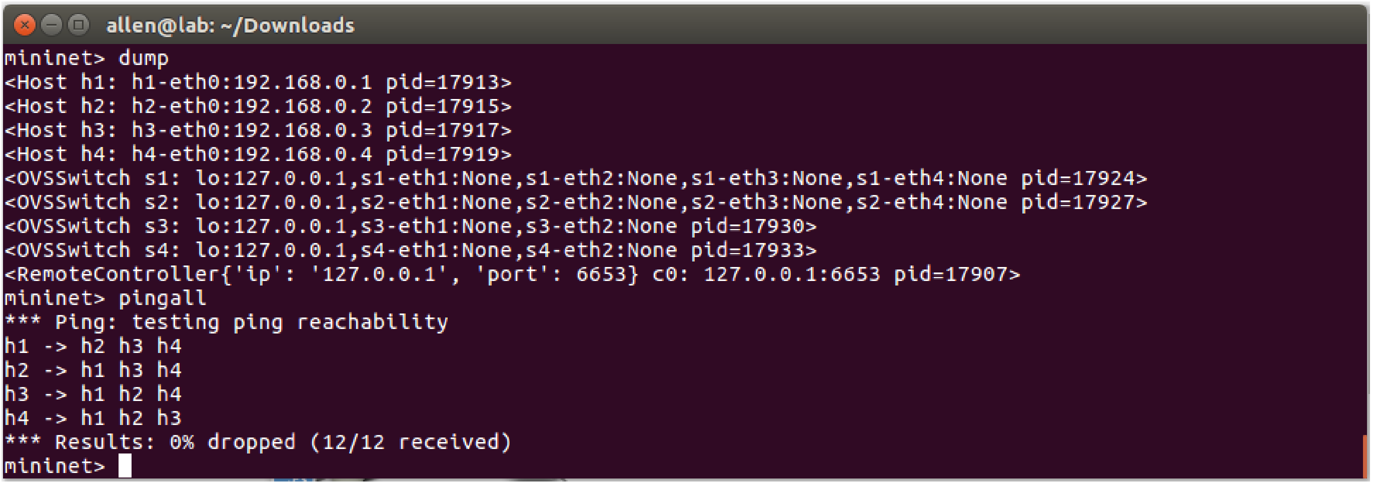
Mininet CLI as below,

Mininet> py h1.setIP('192.168.0.1/24')

Mininet> py h2.setIP('192.168.0.2/24')

Mininet> py h3.setIP('192.168.0.3/24')

Mininet> py h4.setIP('192.168.0.4/24')



```
mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=17913>
<Host h2: h2-eth0:192.168.0.2 pid=17915>
<Host h3: h3-eth0:192.168.0.3 pid=17917>
<Host h4: h4-eth0:192.168.0.4 pid=17919>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None pid=17924>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None pid=17927>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=17930>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=17933>
<RemoteController{'ip': '127.0.0.1', 'port': 6653} c0: 127.0.0.1:6653 pid=17907>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> 
```

Python script as below,

```
from mininet.topo import Topo

class Project1_Topo_Bonus_0756823( Topo ):
    def __init__( self ):
        Topo.__init__( self )

        # Add hosts
        h1 = self.addHost( 'h1', ip='192.168.0.1/24' )
        h2 = self.addHost( 'h2', ip='192.168.0.2/24' )
        h3 = self.addHost( 'h3', ip='192.168.0.3/24' )
        h4 = self.addHost( 'h4', ip='192.168.0.4/24' )

        # Add switches
        s1 = self.addSwitch( 's1' )
         s2 = self.addSwitch( 's2' )
         s3 = self.addSwitch( 's3' )
         s4 = self.addSwitch( 's4' )

        # Add links
        self.addLink( s4, s1 )
         self.addLink( s3, s1 )
         self.addLink( s3, s2 )
         self.addLink( s4, s2 )

        self.addLink( h1, s1 )
        self.addLink( h2, s1 )
        self.addLink( h3, s2 )
        self.addLink( h4, s2 )


topos = { 'topo_bonus_0756823': Project1_Topo_Bonus_0756823 }
```