LAB1 - ONOS and Mininet Installation

> 310581040 智能所 簡郁宸

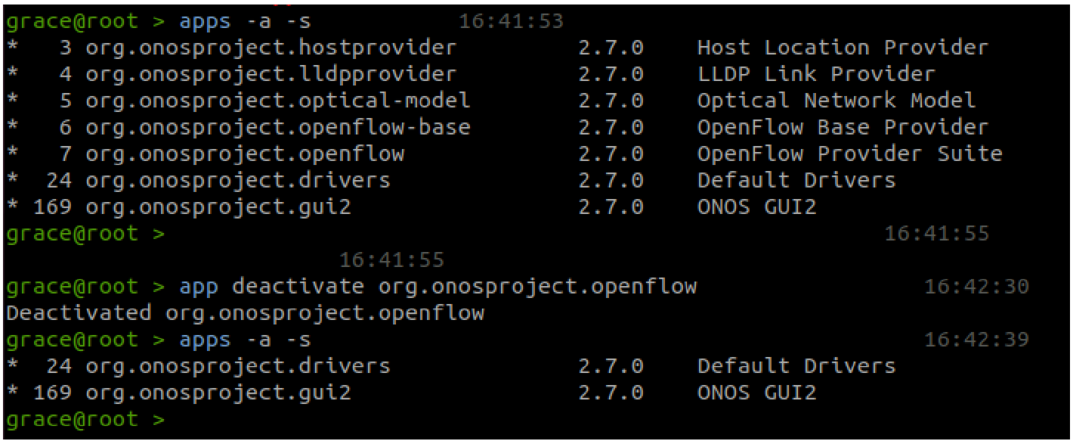
Part 1 Answer Question

- Activate ONOS APPs

  1. **When ONOS activates "org.onosproject.openflow," what are the APPs which it also activates?**

     > - 3 org.onosproject.hostprovider   2.7.0   Host Location Provider
     > - 4 org.onosproject.lldpprovider   2.7.0   LLDP Link Provider
     > - 5 org.onosproject.optical-model   2.7.0   Optical Network Model
     > - 6 org.onosproject.openflow-base   2.7.0   OpenFlow Base Provider

     共額外啟動了 `org.onosproject.hostprovider` 、 `org.onosproject.lldpprovider` 、 `org.onosproject.optical-model` 、 `org.onosproject.openflow-base` 這 4 個 APPs



```
grace@root > apps -a -s              16:41:53
*   3 org.onosproject.hostprovider         2.7.0    Host Location Provider
*   4 org.onosproject.lldpprovider         2.7.0    LLDP Link Provider
*   5 org.onosproject.optical-model        2.7.0    Optical Network Model
*   6 org.onosproject.openflow-base        2.7.0    OpenFlow Base Provider
*   7 org.onosproject.openflow             2.7.0    OpenFlow Provider Suite
*  24 org.onosproject.drivers              2.7.0    Default Drivers
* 169 org.onosproject.gui2                 2.7.0    ONOS GUI2
grace@root >                                                        16:41:55
                         16:41:55
grace@root > app deactivate org.onosproject.openflow                   16:42:30
Deactivated org.onosproject.openflow
grace@root > apps -a -s                                                16:42:39
*  24 org.onosproject.drivers              2.7.0    Default Drivers
* 169 org.onosproject.gui2                 2.7.0    ONOS GUI2
grace@root >
```

  2. **After activating ONOS and running the commands on P.17 and P.20. Will H1 ping H2 successfully?**

     Why or why not?

     > Ref: Basic ONOS tutorial
     >
     > Well, there are no flows installed on the data-plane, which forward the traffic appropriately. ONOS comes with a simple *Reactive Forwarding* app that installs forwarding flows on demand, but this application is not activated by default.

     A: No, 因為在 data-plane 上沒有安裝可以 forward traffic 的 flow。

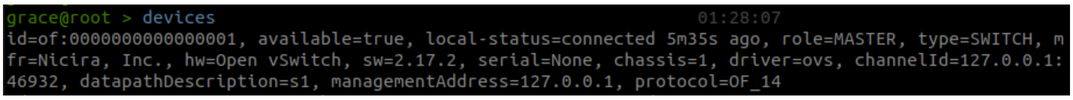
     若要能使的 host 間可以 ping 得到 ，可以激發 ONOS 上的 Reactive Forwarding application (`org.onosproject.fwd`) ，問題便可以解決。

- Observe listening port with terminal command "netstat"

  3. **Which TCP port the controller listens for the OpenFlow connection request from the switch? screenshot**

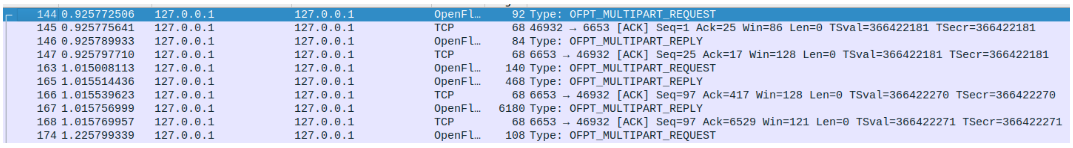
     A: 6653

     從 `devices` 的指令可知 switch 的 port 為 46932 (如下圖)



```
grace@root > devices                                                          01:28:07
id=of:0000000000000001, available=true, local-status=connected 5m35s ago, role=MASTER, type=SWITCH, m
fr=Nicira, Inc., hw=Open vSwitch, sw=2.17.2, serial=None, chassis=1, driver=ovs, channelId=127.0.0.1:
46932, datapathDescription=s1, managementAddress=127.0.0.1, protocol=OF_14
```

接著讓 controller (c0) ping switch 1(s1)，並在 Wireshark 中觀察出和 s1 port 46932 連接的 port 為 6653 (如下圖)， 6653 便為 controller 的 port

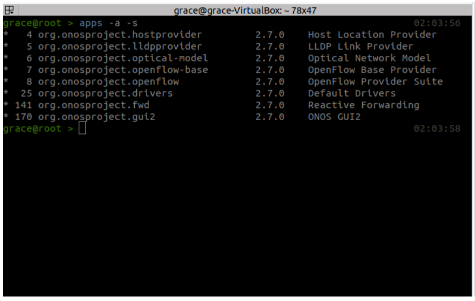


```
144 0.925772506   127.0.0.1   127.0.0.1   OpenFl…    92 Type: OFPT_MULTIPART_REQUEST
145 0.925775641   127.0.0.1   127.0.0.1   TCP        68 46932 → 6653 [ACK] Seq=1 Ack=25 Win=86 Len=0 TSval=366422181 TSecr=366422181
146 0.925789933   127.0.0.1   127.0.0.1   OpenFl…    84 Type: OFPT_MULTIPART_REPLY
147 0.925797710   127.0.0.1   127.0.0.1   TCP        68 6653 → 46932 [ACK] Seq=25 Ack=17 Win=128 Len=0 TSval=366422181 TSecr=366422181
163 1.015008113   127.0.0.1   127.0.0.1   OpenFl…   140 Type: OFPT_MULTIPART_REQUEST
165 1.015514436   127.0.0.1   127.0.0.1   OpenFl…   468 Type: OFPT_MULTIPART_REPLY
166 1.015539623   127.0.0.1   127.0.0.1   TCP        68 6653 → 46932 [ACK] Seq=97 Ack=417 Win=128 Len=0 TSval=366422270 TSecr=366422270
167 1.015756999   127.0.0.1   127.0.0.1   OpenFl…  6180 Type: OFPT_MULTIPART_REPLY
168 1.015769957   127.0.0.1   127.0.0.1   TCP        68 6653 → 46932 [ACK] Seq=97 Ack=6529 Win=121 Len=0 TSval=366422271 TSecr=366422271
174 1.225799339   127.0.0.1   127.0.0.1   OpenFl…   108 Type: OFPT_MULTIPART_REQUEST
```

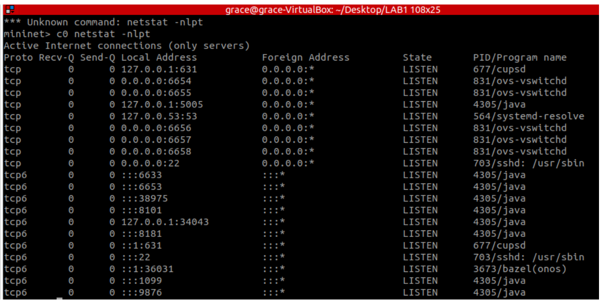
4. **In question 3, which APP enables the controller to listen on the TCP port?**

A: `org.onosproject.openflow-base`

最原始的 APP 運作狀態以及 port 使用狀況

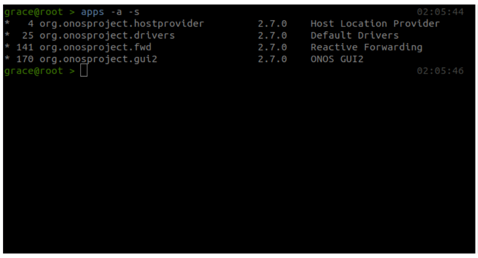


```
grace@root > apps -a -s                                               02:03:56
*   4 org.onosproject.hostprovider        2.7.0    Host Location Provider
*   5 org.onosproject.lldpprovider        2.7.0    LLDP Link Provider
*   6 org.onosproject.optical-model       2.7.0    Optical Network Model
*   7 org.onosproject.openflow-base       2.7.0    OpenFlow Base Provider
*   8 org.onosproject.openflow            2.7.0    OpenFlow Provider Suite
*  25 org.onosproject.drivers             2.7.0    Default Drivers
* 141 org.onosproject.fwd                 2.7.0    Reactive Forwarding
* 170 org.onosproject.gui2                2.7.0    ONOS GUI2
grace@root >                                                          02:03:58
```

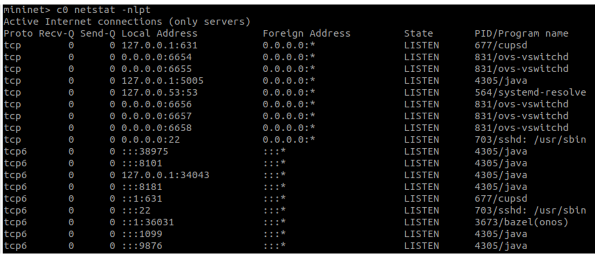


```
*** Unknown command: netstat -nlpt
mininet> c0 netstat -nlpt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      677/cupsd
tcp        0      0 0.0.0.0:6654            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6655            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      4305/java
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      564/systemd-resolve
tcp        0      0 0.0.0.0:6656            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6657            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6658            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      703/sshd: /usr/sbin
tcp6       0      0 :::6633                 :::*                    LISTEN      4305/java
tcp6       0      0 :::6653                 :::*                    LISTEN      4305/java
tcp6       0      0 :::38975                :::*                    LISTEN      4305/java
tcp6       0      0 :::8101                 :::*                    LISTEN      4305/java
tcp6       0      0 127.0.0.1:34043         :::*                    LISTEN      4305/java
tcp6       0      0 :::8181                 :::*                    LISTEN      4305/java
tcp6       0      0 ::1:631                 :::*                    LISTEN      677/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      703/sshd: /usr/sbin
tcp6       0      0 ::1:36031               :::*                    LISTEN      3673/bazel(onos)
tcp6       0      0 :::1099                 :::*                    LISTEN      4305/java
tcp6       0      0 :::9876                 :::*                    LISTEN      4305/java
```

當 deactivate `org.onosproject.openflow` 時，APP 運作狀態以及 port 使用狀況，可看出 port 6633 和 6653 已經消失

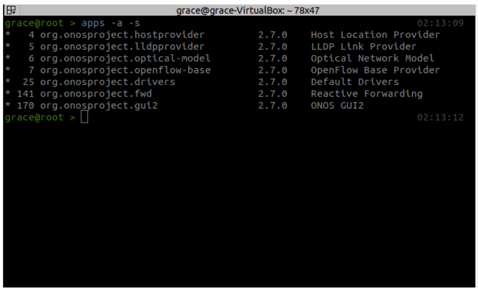


```
grace@root > apps -a -s                                               02:05:44
*   4 org.onosproject.hostprovider        2.7.0    Host Location Provider
*  25 org.onosproject.drivers             2.7.0    Default Drivers
* 141 org.onosproject.fwd                 2.7.0    Reactive Forwarding
* 170 org.onosproject.gui2                2.7.0    ONOS GUI2
grace@root >                                                          02:05:46
```

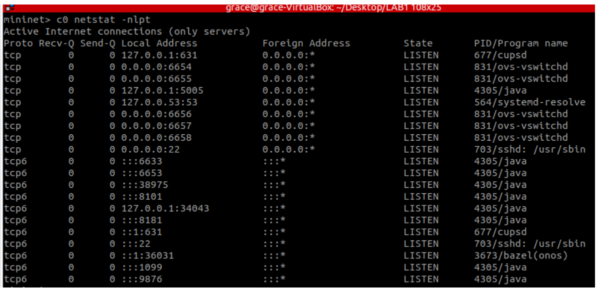


```
mininet> c0 netstat -nlpt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      677/cupsd
tcp        0      0 0.0.0.0:6654            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6655            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      4305/java
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      564/systemd-resolve
tcp        0      0 0.0.0.0:6656            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6657            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6658            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      703/sshd: /usr/sbin
tcp6       0      0 :::38975                :::*                    LISTEN      4305/java
tcp6       0      0 :::8101                 :::*                    LISTEN      4305/java
tcp6       0      0 127.0.0.1:34043         :::*                    LISTEN      4305/java
tcp6       0      0 :::8181                 :::*                    LISTEN      4305/java
tcp6       0      0 ::1:631                 :::*                    LISTEN      677/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      703/sshd: /usr/sbin
tcp6       0      0 ::1:36031               :::*                    LISTEN      3673/bazel(onos)
tcp6       0      0 :::1099                 :::*                    LISTEN      4305/java
tcp6       0      0 :::9876                 :::*                    LISTEN      4305/java
```

接著啟動一個個被停止運作的 APP ，發現當 `org.onosproject.openflow-base` activate 且 `org.onosproject.openflow-base` deactivate 時，port 6633 和 6653 開始運作，表示 `org.onosproject.openflow-base` 可以使得 controller 能在 tcp port 上監聽



```
grace@root > apps -a -s                                               02:13:09
*   4 org.onosproject.hostprovider        2.7.0    Host Location Provider
*   5 org.onosproject.lldpprovider        2.7.0    LLDP Link Provider
*   6 org.onosproject.optical-model       2.7.0    Optical Network Model
*   7 org.onosproject.openflow-base       2.7.0    OpenFlow Base Provider
*  25 org.onosproject.drivers             2.7.0    Default Drivers
* 141 org.onosproject.fwd                 2.7.0    Reactive Forwarding
* 170 org.onosproject.gui2                2.7.0    ONOS GUI2
grace@root >                                                          02:13:12
```



```
mininet> c0 netstat -nlpt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      677/cupsd
tcp        0      0 0.0.0.0:6654            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6655            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      4305/java
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      564/systemd-resolve
tcp        0      0 0.0.0.0:6656            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6657            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:6658            0.0.0.0:*               LISTEN      831/ovs-vswitchd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      703/sshd: /usr/sbin
tcp6       0      0 :::6633                 :::*                    LISTEN      4305/java
tcp6       0      0 :::6653                 :::*                    LISTEN      4305/java
tcp6       0      0 :::38975                :::*                    LISTEN      4305/java
tcp6       0      0 :::8101                 :::*                    LISTEN      4305/java
tcp6       0      0 127.0.0.1:34043         :::*                    LISTEN      4305/java
tcp6       0      0 :::8181                 :::*                    LISTEN      4305/java
tcp6       0      0 ::1:631                 :::*                    LISTEN      677/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      703/sshd: /usr/sbin
tcp6       0      0 ::1:36031               :::*                    LISTEN      3673/bazel(onos)
tcp6       0      0 :::1099                 :::*                    LISTEN      4305/java
tcp6       0      0 :::9876                 :::*                    LISTEN      4305/java
```

Part 2 Create a custom Topology

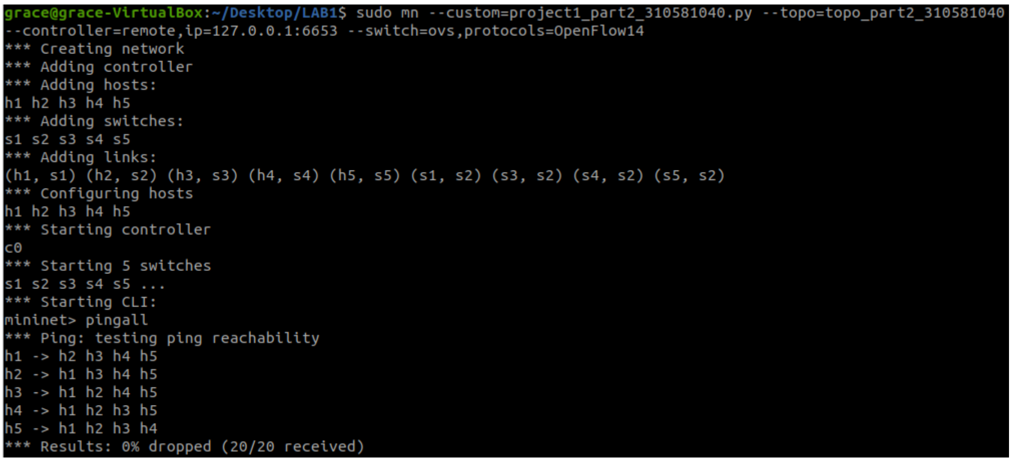
- Write a Python script to build the following topology\

```python
from mininet.topo import Topo


class Project1_Topo_310581040( 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')
        h5 = self.addHost('h5')

        # Add switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')
        s5 = self.addSwitch('s5')

        # Add switch/switch
        self.addLink( s1, s2 )
        self.addLink( s3, s2 )
        self.addLink( s4, s2 )
        self.addLink( s5, s2 )

        # Add host/swtich
        self.addLink( h1, s1 )
        self.addLink( h2, s2 )
        self.addLink( h3, s3 )
        self.addLink( h4, s4 )
        self.addLink( h5, s5 )


topos = {'topo_part2_310581040': Project1_Topo_310581040 }
```

- Run your Python script and use command “pingall”.

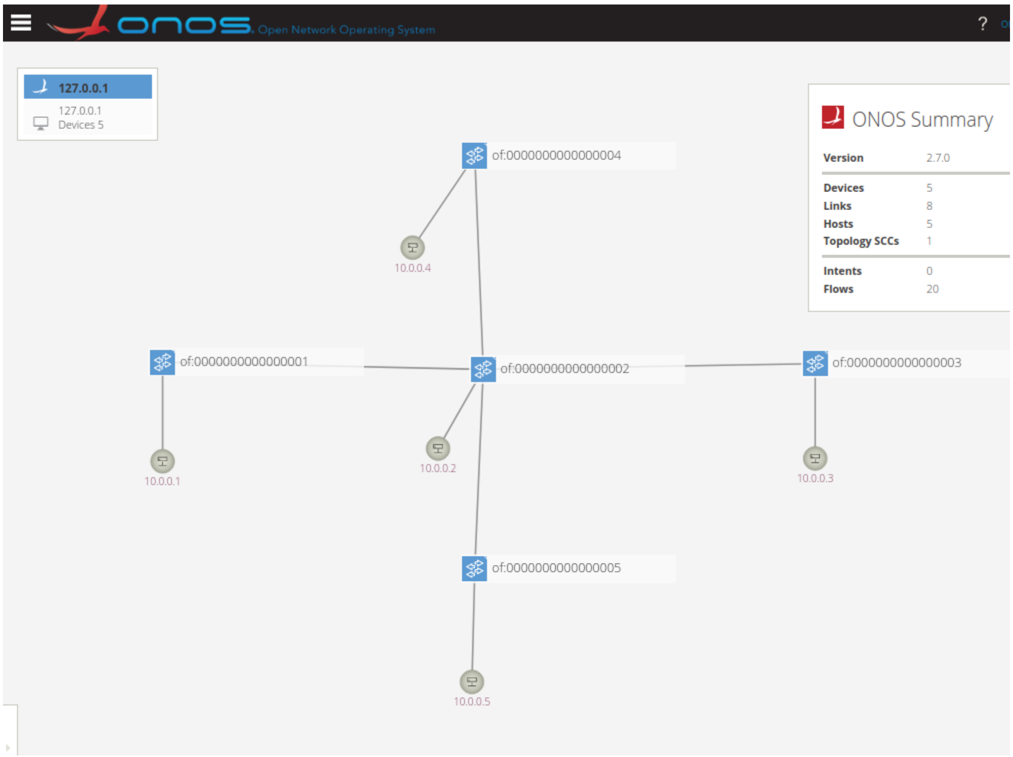
```
$ sudo mn --custom=project1_part2_310581040.py \
--topo=topo_part2_310581040 \
--controller=remote,ip=127.0.0.1:6653 \
--switch=ovs,protocols=OpenFlow14
```

  - 執行過程



```
grace@grace-VirtualBox:~/Desktop/LAB1$ sudo mn --custom=project1_part2_310581040.py --topo=topo_part2_310581040
--controller=remote,ip=127.0.0.1:6653 --switch=ovs,protocols=OpenFlow14
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (h5, s5) (s1, s2) (s3, s2) (s4, s2) (s5, s2)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)
```

- Then take a **screenshot** of topology on GUI.
  - GUI



ONOS Summary

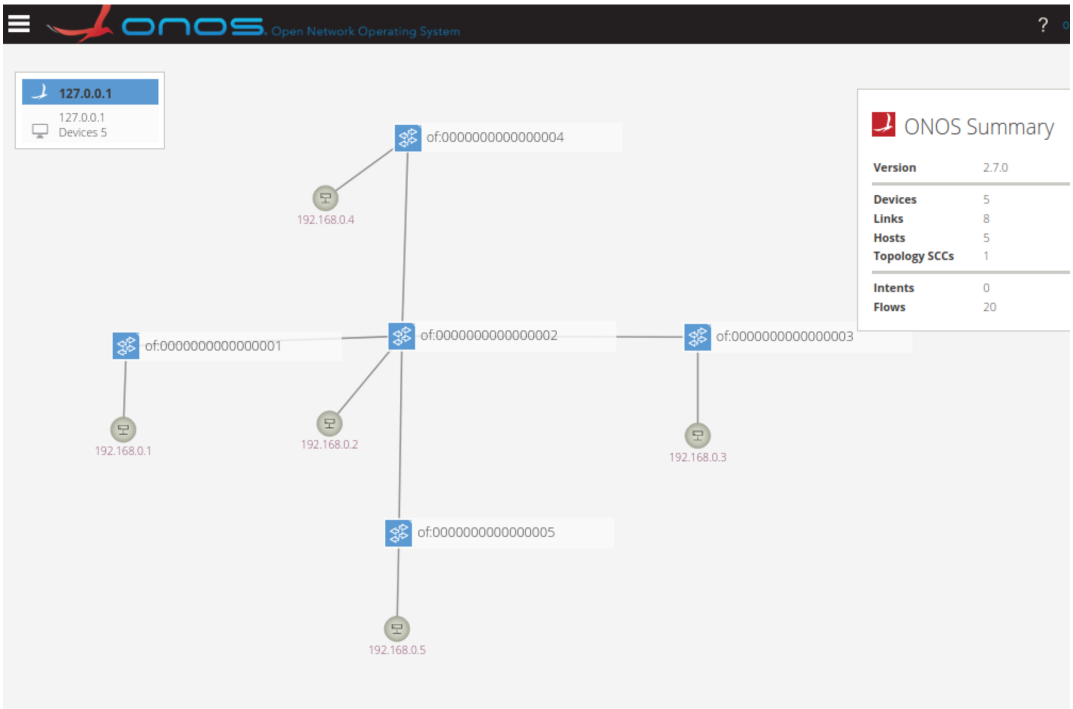
Version	2.7.0
Devices	5
Links	8
Hosts	5
Topology SCCs	1
Intents	0
Flows	20

Part 3 Statically assign Hosts IP Address in Mininet

- Reuse the topology in part 2
- Format for manual assignment of host IP address:
  - – 192.168.0.0/27
  - – netmask 255.255.255.224
- Statically assign IP addresses with Python and hand in the Python script you've edited

```
from mininet.topo import Topo


class Project1_Topo_310581040( Topo ):
    def __init__(self):
        Topo.__init__(self)
        # Add hosts
        h1 = self.addHost('h1', ip = '192.168.0.1/27')
        h2 = self.addHost('h2', ip = '192.168.0.2/27')
        h3 = self.addHost('h3', ip = '192.168.0.3/27')
        h4 = self.addHost('h4', ip = '192.168.0.4/27')
        h5 = self.addHost('h5', ip = '192.168.0.5/27')

        # Add switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')
        s5 = self.addSwitch('s5')

        # Add switch/switch
        self.addLink( s1, s2 )
        self.addLink( s3, s2 )
        self.addLink( s4, s2 )
        self.addLink( s5, s2 )

        # Add links
        self.addLink( h1, s1 )
        self.addLink( h2, s2 )
        self.addLink( h3, s3 )
        self.addLink( h4, s4 )
        self.addLink( h5, s5 )



topos = {'topo_part3_310581040': Project1_Topo_310581040 }
```

- **Screenshots** of manual assignment of host IP address



ONOS Summary

Version 2.7.0

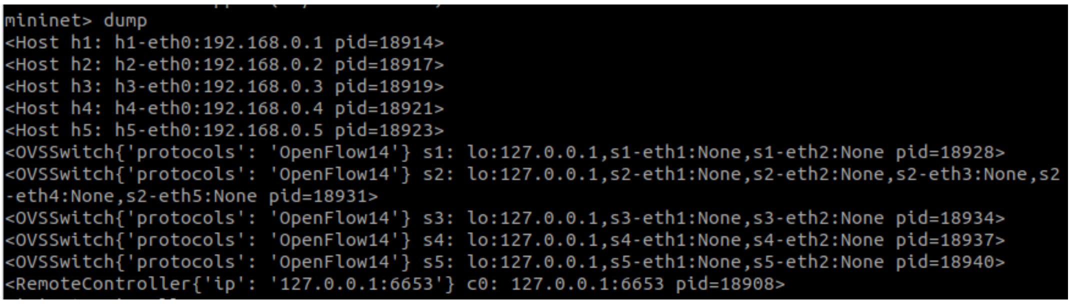
Devices 5
Links 8
Hosts 5
Topology SCCs 1

Intents 0
Flows 20

- Start `mn` with your Python script

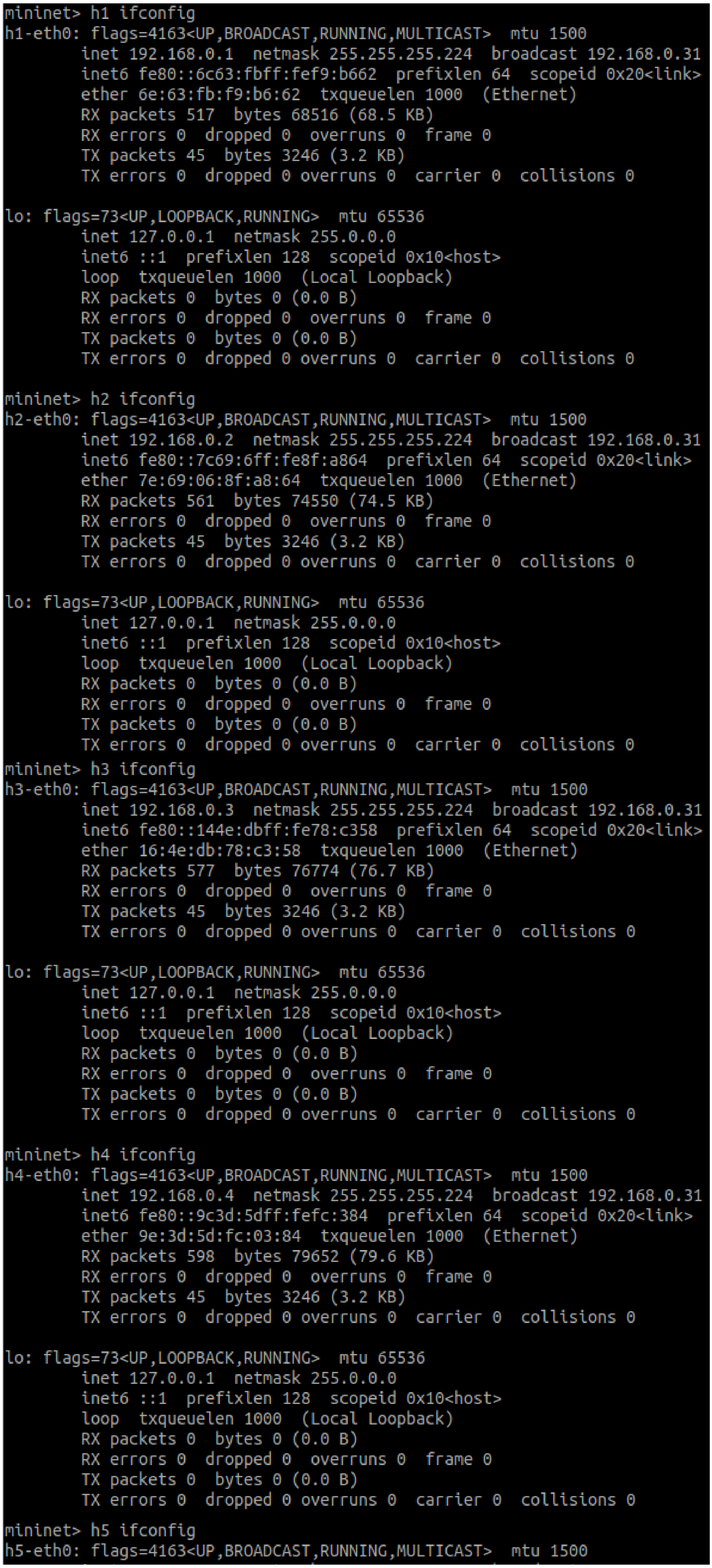
```
$ sudo mn --custom=project1_part3_310581040.py \
--topo=topo_part3_310581040 \
--controller=remote,ip=127.0.0.1:6653 \
--switch=ovs,protocols=OpenFlow14
```

- Take screenshots with command `dump` and `ifconfig` for all host.
  - `dump`

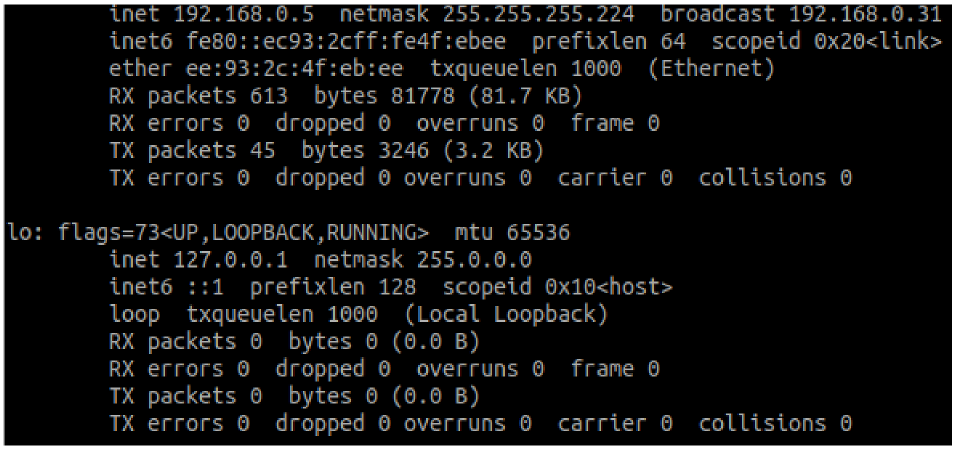


```
mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=18914>
<Host h2: h2-eth0:192.168.0.2 pid=18917>
<Host h3: h3-eth0:192.168.0.3 pid=18919>
<Host h4: h4-eth0:192.168.0.4 pid=18921>
<Host h5: h5-eth0:192.168.0.5 pid=18923>
<OVSSwitch{'protocols': 'OpenFlow14'} s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=18928>
<OVSSwitch{'protocols': 'OpenFlow14'} s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2
-eth4:None,s2-eth5:None pid=18931>
<OVSSwitch{'protocols': 'OpenFlow14'} s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=18934>
<OVSSwitch{'protocols': 'OpenFlow14'} s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=18937>
<OVSSwitch{'protocols': 'OpenFlow14'} s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None pid=18940>
<RemoteController{'ip': '127.0.0.1:6653'} c0: 127.0.0.1:6653 pid=18908>
```

    - `ifconfig`



```
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.1  netmask 255.255.255.224  broadcast 192.168.0.31
        inet6 fe80::6c63:fbff:fef9:b662  prefixlen 64  scopeid 0x20<link>
        ether 6e:63:fb:f9:b6:62  txqueuelen 1000  (Ethernet)
        RX packets 517  bytes 68516 (68.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 3246 (3.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h2 ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.2  netmask 255.255.255.224  broadcast 192.168.0.31
        inet6 fe80::7c69:6ff:fe8f:a864  prefixlen 64  scopeid 0x20<link>
        ether 7e:69:06:8f:a8:64  txqueuelen 1000  (Ethernet)
        RX packets 561  bytes 74550 (74.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 3246 (3.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
mininet> h3 ifconfig
h3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.3  netmask 255.255.255.224  broadcast 192.168.0.31
        inet6 fe80::144e:dbff:fe78:c358  prefixlen 64  scopeid 0x20<link>
        ether 16:4e:db:78:c3:58  txqueuelen 1000  (Ethernet)
        RX packets 577  bytes 76774 (76.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 3246 (3.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h4 ifconfig
h4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.4  netmask 255.255.255.224  broadcast 192.168.0.31
        inet6 fe80::9c3d:5dff:fefc:384  prefixlen 64  scopeid 0x20<link>
        ether 9e:3d:5d:fc:03:84  txqueuelen 1000  (Ethernet)
        RX packets 598  bytes 79652 (79.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 3246 (3.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h5 ifconfig
h5-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```



```
        inet 192.168.0.5  netmask 255.255.255.224  broadcast 192.168.0.31
        inet6 fe80::ec93:2cff:fe4f:ebee  prefixlen 64  scopeid 0x20<link>
        ether ee:93:2c:4f:eb:ee  txqueuelen 1000  (Ethernet)
        RX packets 613  bytes 81778 (81.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 3246 (3.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

What you’ve learned or solved

- 首先在最一開始測試時並沒有發現在複製 tutorial 的指令時，將 terminal 換行符號也複製上去，導致後面指令未執行，直接成為輸出導向，讓 ONOS 設置 host 時失敗。
- 另外，在前面實作 tutorial 時並未發現未啟動`org.onosproject.fwd` 的 APP 時會導致無法使用 pingall，就如第一大題問題所說，在 default 的 data plane 上並沒有連接傳輸的 flow ，故會導致此問題。
- 在第三部分時 ip 遮罩轉換的部分已經忘得差不多了，特別網路上查了一下喚起了記憶:

  遮罩 255.255.255.224 (11111111.11111111.11111111.11100000) -> $256 - 224 = 32 = 2^5$ -> 32 - 5 = 27 為 net address