Project 1. ONOS and Mininet

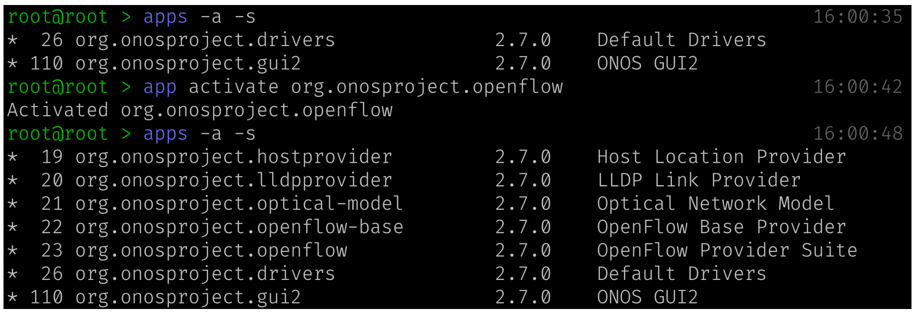
312551015 Sean 韋詠祥

Part 1: Answer Questions

Q1. What are the APPs which it also activates?

It will also activate 4 other apps: `hostprovider`, `lldpprovider`, `optical-model`, `openflow-base` under `org.onosproject.*` namespace.

```
*  19 org.onosproject.hostprovider        2.7.0    Host Location Provider
*  20 org.onosproject.lldpprovider        2.7.0    LLDP Link Provider
*  21 org.onosproject.optical-model       2.7.0    Optical Network Model
*  22 org.onosproject.openflow-base       2.7.0    OpenFlow Base Provider
*  23 org.onosproject.openflow            2.7.0    OpenFlow Provider Suite
```



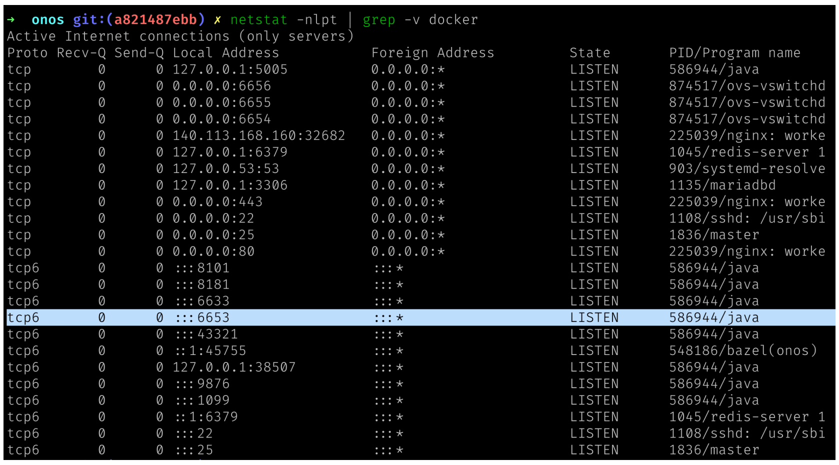
```
root@root > apps -a -s                                                     16:00:35
*  26 org.onosproject.drivers              2.7.0    Default Drivers
* 110 org.onosproject.gui2                 2.7.0    ONOS GUI2
root@root > app activate org.onosproject.openflow                          16:00:42
Activated org.onosproject.openflow
root@root > apps -a -s                                                     16:00:48
*  19 org.onosproject.hostprovider         2.7.0    Host Location Provider
*  20 org.onosproject.lldpprovider         2.7.0    LLDP Link Provider
*  21 org.onosproject.optical-model        2.7.0    Optical Network Model
*  22 org.onosproject.openflow-base        2.7.0    OpenFlow Base Provider
*  23 org.onosproject.openflow             2.7.0    OpenFlow Provider Suite
*  26 org.onosproject.drivers              2.7.0    Default Drivers
* 110 org.onosproject.gui2                 2.7.0    ONOS GUI2
```

Q2. Will H1 ping H2 successfully? Why or why not?

No, we need to also activate `org.onosproject.fwd` app to enable that function.

Q3. Which TCP port the controller listens?

The controller will listen to TCP 6653 port in this experiment.



```
➜  onos git:(a821487ebb) ✗ netstat -nlpt | grep -v docker
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      586944/java
tcp        0      0 0.0.0.0:6656            0.0.0.0:*               LISTEN      874517/ovs-vswitchd
tcp        0      0 0.0.0.0:6655            0.0.0.0:*               LISTEN      874517/ovs-vswitchd
tcp        0      0 0.0.0.0:6654            0.0.0.0:*               LISTEN      874517/ovs-vswitchd
tcp        0      0 140.113.168.160:32682   0.0.0.0:*               LISTEN      225039/nginx: worke
tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTEN      1045/redis-server 1
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      903/systemd-resolve
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      1135/mariadbd
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      225039/nginx: worke
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1108/sshd: /usr/sbi
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      1836/master
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      225039/nginx: worke
tcp6       0      0 :::8101                 :::*                    LISTEN      586944/java
tcp6       0      0 :::8181                 :::*                    LISTEN      586944/java
tcp6       0      0 :::6633                 :::*                    LISTEN      586944/java
tcp6       0      0 :::6653                 :::*                    LISTEN      586944/java
tcp6       0      0 :::43321                :::*                    LISTEN      586944/java
tcp6       0      0 ::1:45755               :::*                    LISTEN      548186/bazel(onos)
tcp6       0      0 127.0.0.1:38507         :::*                    LISTEN      586944/java
tcp6       0      0 :::9876                 :::*                    LISTEN      586944/java
tcp6       0      0 :::1099                 :::*                    LISTEN      586944/java
tcp6       0      0 ::1:6379                :::*                    LISTEN      1045/redis-server 1
tcp6       0      0 :::22                   :::*                    LISTEN      1108/sshd: /usr/sbi
tcp6       0      0 :::25                   :::*                    LISTEN      1836/master
```

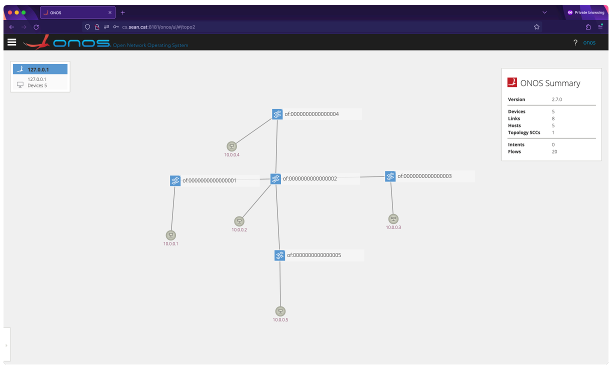
Q4. Which APP enables the controller to listen on the TCP port?

That TCP port will be up/down when we activate "`org.onosproject.openflow`" app and deactivate it.
To be specific, it's "`org.onosproject.openflow-base`" app to enable the controller to listen that port.

```
*  22 org.onosproject.openflow-base        2.7.0    OpenFlow Base Provider
```

Part 2: Create a Custom Topology

Using our customized Python script, we can get the same topology as the spec.

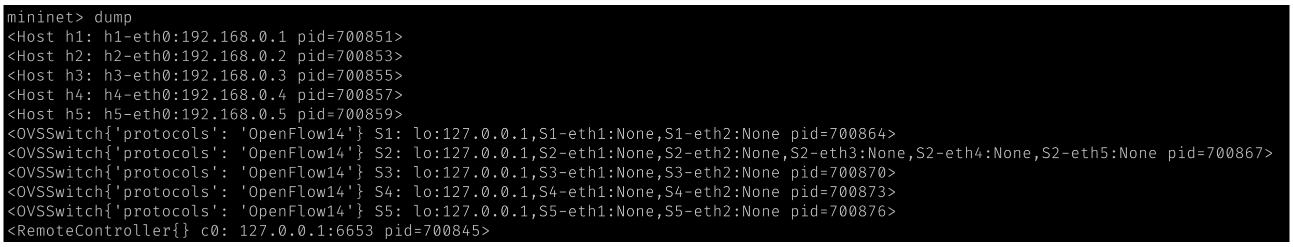


ONOS Summary

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

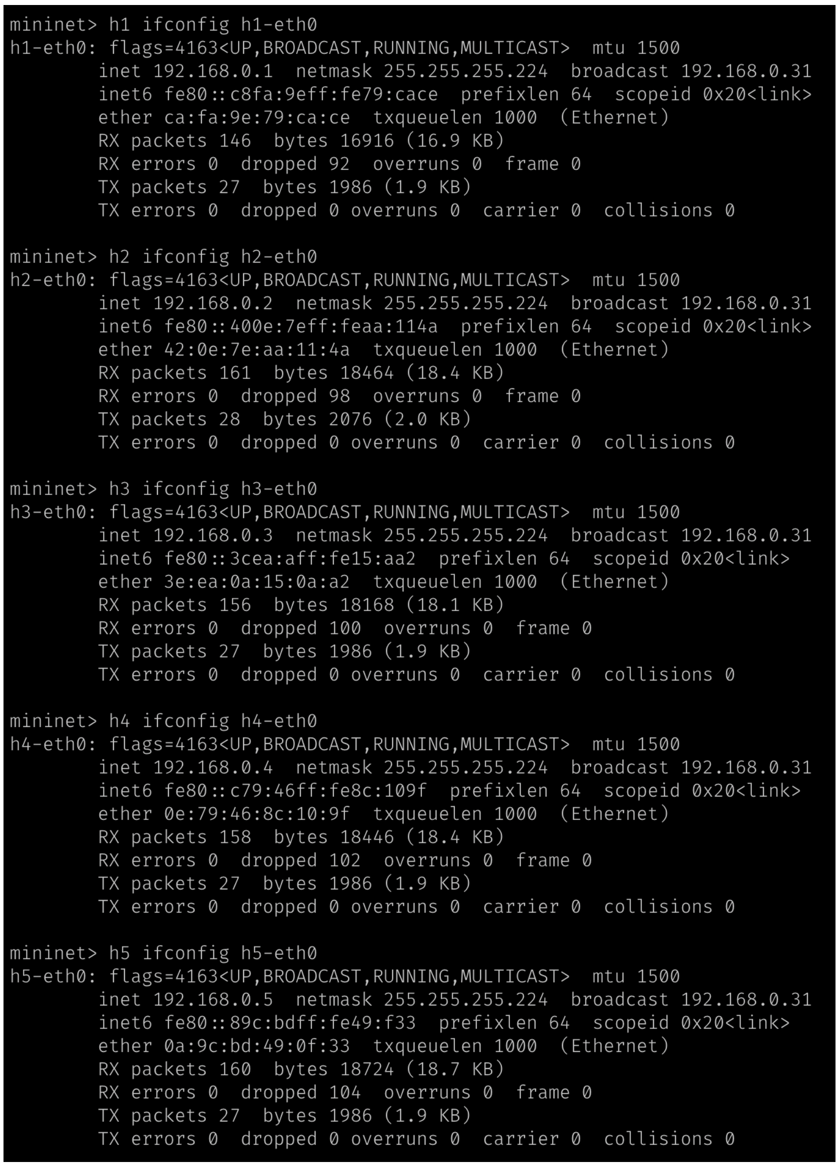
Part 3: Statically Assign Hosts IP Address IP in Mininet

After assigned the IP address by `addHost('h1', ip=‘192.168.0.1/27')` statement, we can verify that it is succeed via `dump` command.



```
mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=700851>
<Host h2: h2-eth0:192.168.0.2 pid=700853>
<Host h3: h3-eth0:192.168.0.3 pid=700855>
<Host h4: h4-eth0:192.168.0.4 pid=700857>
<Host h5: h5-eth0:192.168.0.5 pid=700859>
<OVSSwitch{'protocols': 'OpenFlow14'} S1: lo:127.0.0.1,S1-eth1:None,S1-eth2:None pid=700864>
<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=700867>
<OVSSwitch{'protocols': 'OpenFlow14'} S3: lo:127.0.0.1,S3-eth1:None,S3-eth2:None pid=700870>
<OVSSwitch{'protocols': 'OpenFlow14'} S4: lo:127.0.0.1,S4-eth1:None,S4-eth2:None pid=700873>
<OVSSwitch{'protocols': 'OpenFlow14'} S5: lo:127.0.0.1,S5-eth1:None,S5-eth2:None pid=700876>
<RemoteController{} c0: 127.0.0.1:6653 pid=700845>
```

And verified that interface config in `h[1-5]` is correct.



```
mininet> h1 ifconfig h1-eth0
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::c8fa:9eff:fe79:cace  prefixlen 64  scopeid 0x20<link>
        ether ca:fa:9e:79:ca:ce  txqueuelen 1000  (Ethernet)
        RX packets 146  bytes 16916 (16.9 KB)
        RX errors 0  dropped 92  overruns 0  frame 0
        TX packets 27  bytes 1986 (1.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h2 ifconfig h2-eth0
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::400e:7eff:feaa:114a  prefixlen 64  scopeid 0x20<link>
        ether 42:0e:7e:aa:11:4a  txqueuelen 1000  (Ethernet)
        RX packets 161  bytes 18464 (18.4 KB)
        RX errors 0  dropped 98  overruns 0  frame 0
        TX packets 28  bytes 2076 (2.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h3 ifconfig h3-eth0
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::3cea:aff:fe15:aa2  prefixlen 64  scopeid 0x20<link>
        ether 3e:ea:0a:15:0a:a2  txqueuelen 1000  (Ethernet)
        RX packets 156  bytes 18168 (18.1 KB)
        RX errors 0  dropped 100  overruns 0  frame 0
        TX packets 27  bytes 1986 (1.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h4 ifconfig h4-eth0
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::c79:46ff:fe8c:109f  prefixlen 64  scopeid 0x20<link>
        ether 0e:79:46:8c:10:9f  txqueuelen 1000  (Ethernet)
        RX packets 158  bytes 18446 (18.4 KB)
        RX errors 0  dropped 102  overruns 0  frame 0
        TX packets 27  bytes 1986 (1.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> h5 ifconfig h5-eth0
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::89c:bdff:fe49:f33  prefixlen 64  scopeid 0x20<link>
        ether 0a:9c:bd:49:0f:33  txqueuelen 1000  (Ethernet)
        RX packets 160  bytes 18724 (18.7 KB)
        RX errors 0  dropped 104  overruns 0  frame 0
        TX packets 27  bytes 1986 (1.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Part 4: What we learned from this project

It’s not my first time to use Mininet and ONOS controller, but never used the Web GUI version before. I think it’s easier for newbies to learn SDN.