

Project 1 - ONOS and Mininet Installation

112109045

Part1

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

```
sylvia@root > apps -a -s
* 14 org.onosproject.drivers 2.7.0 Default Drivers
* 128 org.onosproject.gui2 2.7.0 ONOS GUI2
sylvia@root > app activate org.onosproject.openflow
Activated org.onosproject.openflow
sylvia@root > apps -a -s
* 14 org.onosproject.drivers 2.7.0 Default Drivers
* 27 org.onosproject.optical-model 2.7.0 Optical Network Model
* 28 org.onosproject.openflow-base 2.7.0 OpenFlow Base Provider
* 39 org.onosproject.hostprovider 2.7.0 Host Location Provider
* 40 org.onosproject.llpprovider 2.7.0 LLDP Link Provider
* 41 org.onosproject.openflow 2.7.0 OpenFlow Provider Suite
* 128 org.onosproject.gui2 2.7.0 ONOS GUI2
sylvia@root >
```

2 . After activating ONOS and running the commands on P.17 and P.20. Will H1 ping H2 successfully? Why or why not?

H1 is not able to ping H2 without activating “Reactive Forward” because there are no flows installed on the data-plane, which forward the traffic appropriately. According to “Basic ONOS Tutorial” , it is the app for adding flow rules which are needed to correctly route the traffic.

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

After activating openFlow, netstat shows there are two more listening port: 6633 and 6653. It turns out 6653/tcp is the port openFlow controller to listen for switches and the port 6633/tcp is used in ealiar version of openFlow.(Reference: <https://www.speedguide.net/port.php?port=6653>)

- Before activating openflow

```
sylvia@sylvia-VirtualBox:~/onos$ netstat -nltp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6654           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6655           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:5005         0.0.0.0:*               LISTEN      87484/java
tcp        0      0 0.0.0.0:6656           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::1099                :::*                   LISTEN      87484/java
tcp6       0      0 :::134201              :::*                   LISTEN      87200/bazel(onos)
tcp6       0      0 :::9876                :::*                   LISTEN      87484/java
tcp6       0      0 :::631                 :::*                   LISTEN      -
tcp6       0      0 :::22                  :::*                   LISTEN      -
tcp6       0      0 :::8181                :::*                   LISTEN      87484/java
tcp6       0      0 :::8101                :::*                   LISTEN      87484/java
tcp6       0      0 127.0.0.1:41007        :::*                   LISTEN      87484/java
tcp6       0      0 :::43689               :::*                   LISTEN      87484/java
```

- After activating openflow:

```
sylvia@sylvia-VirtualBox:~/onos$ netstat -nlpt
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6654           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6655           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:5005         0.0.0.0:*               LISTEN      87484/java
tcp        0      0 0.0.0.0:6656           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::1099                :::*                    LISTEN      87484/java
tcp6       0      0 :::134201              :::*                    LISTEN      87200/bazel(onos)
tcp6       0      0 :::9876                :::*                    LISTEN      87484/java
tcp6       0      0 :::1631                :::*                    LISTEN      -
tcp6       0      0 :::22                  :::*                    LISTEN      -
tcp6       0      0 :::8181                :::*                    LISTEN      87484/java
tcp6       0      0 :::8101                :::*                    LISTEN      87484/java
tcp6       0      0 :::6633                :::*                    LISTEN      87484/java
tcp6       0      0 :::6653                :::*                    LISTEN      87484/java
tcp6       0      0 127.0.0.1:41007        :::*                    LISTEN      87484/java
tcp6       0      0 :::43689               :::*                    LISTEN      87484/java
```

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

OpenFlow Base Provider is the APP that enables the controller to listen on the TCP port. According to the first screenshot below, openFlow has three dependencies: (1) LLDP Link Provider (2) Host Location Provider (3) OpenFlow Base Provider. The first two should have nothing to do with listening port.

(Reference: <https://wiki.onosproject.org/display/ONOS/Requirements>)

```
sylvia@root > app deactivate org.onosproject.openflow-base
Deactivated org.onosproject.openflow-base
sylvia@root > apps -a -s
* 14 org.onosproject.drivers          2.7.0    Default Drivers
* 125 org.onosproject.fwd             2.7.0    Reactive Forwarding
* 128 org.onosproject.gui2            2.7.0    ONOS GUI2
sylvia@root > app activate org.onosproject.openflow
Activated org.onosproject.openflow
sylvia@root > apps -a -s
* 14 org.onosproject.drivers          2.7.0    Default Drivers
* 27 org.onosproject.optical-model    2.7.0    Optical Network Model
* 28 org.onosproject.openflow-base    2.7.0    OpenFlow Base Provider
* 39 org.onosproject.hostprovider     2.7.0    Host Location Provider
* 40 org.onosproject.lldpprovider     2.7.0    LLDP Link Provider
* 41 org.onosproject.openflow         2.7.0    OpenFlow Provider Suite
* 125 org.onosproject.fwd             2.7.0    Reactive Forwarding
* 128 org.onosproject.gui2            2.7.0    ONOS GUI2
```

- Before activating openflow-base:

```
sylvia@sylvia-VirtualBox:~/onos$ netstat -nlpt
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6654           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6655           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:5005         0.0.0.0:*               LISTEN      87484/java
tcp        0      0 0.0.0.0:6656           0.0.0.0:*               LISTEN      -
```



```

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

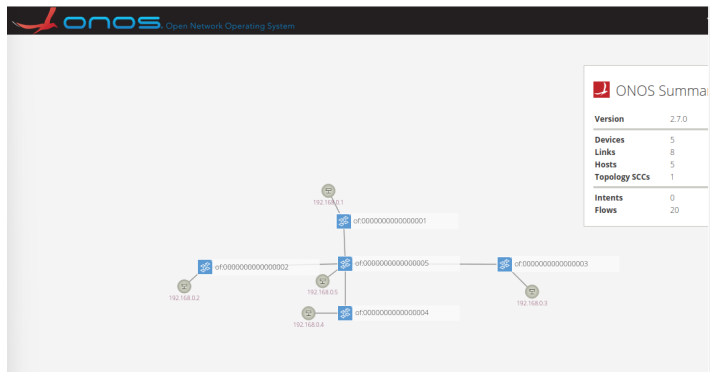
# Add links
self.addLink( h1, s1 )
self.addLink( h2, s2 )
self.addLink( h3, s3 )
self.addLink( h4, s4 )
self.addLink( h5, s5 )
self.addLink( s2, s4 )
self.addLink( s2, s3 )
self.addLink( s2, s1 )
self.addLink( s2, s5 )

topos = { 'topo_part2_112109045': Project1_Topo_112109045 }
~
~
~
"Lab1/project1_part2_112109045.py" 33L, 891B

```

Part3

- Start mn with your Python script and take screenshots with command “dump” and “ifconfig” for all host



```

mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=95491>
<Host h2: h2-eth0:192.168.0.2 pid=95493>
<Host h3: h3-eth0:192.168.0.3 pid=95495>
<Host h4: h4-eth0:192.168.0.4 pid=95497>
<Host h5: h5-eth0:192.168.0.5 pid=95499>

```

```

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::4465:1dff:febb:992a prefixlen 64 scopeid 0x20<link>
ether 46:65:1d:bb:99:2a txqueuelen 1000 (Ethernet)
RX packets 201 bytes 25868 (25.8 KB)
RX errors 0 dropped 152 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: 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::ac9e:46ff:fece:9d5e prefixlen 64 scopeid 0x20<link>
ether ae:9e:46:ce:9d:5e txqueuelen 1000 (Ethernet)
RX packets 262 bytes 34141 (34.1 KB)
RX errors 0 dropped 208 overruns 0 frame 0
TX packets 28 bytes 2056 (2.0 KB)
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::5496:d5ff:feac:274 prefixlen 64 scopeid 0x20<link>
ether 56:96:d5:ac:02:74 txqueuelen 1000 (Ethernet)
RX packets 273 bytes 35711 (35.7 KB)
RX errors 0 dropped 220 overruns 0 frame 0
TX packets 28 bytes 2056 (2.0 KB)
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::969b:b5ff:fe9d:5b6 prefixlen 64 scopeid 0x20<link>
ether 96:9b:b5:fd:9d:5b txqueuelen 1000 (Ethernet)
RX packets 273 bytes 35711 (35.7 KB)
RX errors 0 dropped 220 overruns 0 frame 0
TX packets 28 bytes 2056 (2.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

```

lnet6 fe80::800:b11:fe0:dab0 prefixlen 64 scopeid 0x20<link>
ether 0a:60:0b:c0:da:b0 txqueuelen 1000 (Ethernet)
RX packets 285 bytes 37379 (37.3 KB)
RX errors 0 dropped 232 overruns 0 frame 0
TX packets 28 bytes 2056 (2.0 KB)
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::4c6e:fff:fe08:6641 prefixlen 64 scopeid 0x20<link>
ether 4e:6e:0f:08:66:41 txqueuelen 1000 (Ethernet)
RX packets 297 bytes 39047 (39.0 KB)
RX errors 0 dropped 244 overruns 0 frame 0
TX packets 28 bytes 2056 (2.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- Script:
 - a. Create and add 5 hosts by “addHost” with defined ip
 - b. Create and add 5 switches by “addSwitch”
 - c. Connect the hosts and switches by “addLink” , then assign the relationship between switches

```

from mininet.topo import Topo

class Project1_Topo_112109045( 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 links
        self.addLink( h1, s1 )
        self.addLink( h2, s2 )
        self.addLink( h3, s3 )
        self.addLink( h4, s4 )
        self.addLink( h5, s5 )
        self.addLink( s1, s5 )
        self.addLink( s2, s5 )
        self.addLink( s3, s5 )
        self.addLink( s4, s5 )

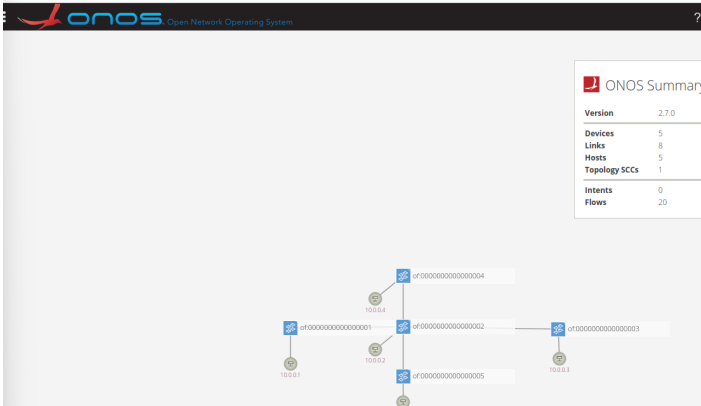
topos = { 'topo_part3_112109045': Project1_Topo_112109045 }
~
~
~
"Lab1/project1_part3_112109045.py" [readonly] 32L, 990B

```

What I’ ve learned or solved

1 . Amount of links:

Q: Why there are 9 links showed at mininet terminal, but ONOS Summary shows 8 links only?



```
sylvia@sylvia-VirtualBox:~/onos/Lab1$ sudo mn --custom=project1_part2_112109045.py --topo=topo_part2_1121090
45 --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) (s2, s1) (s2, s3) (s2, s4) (s2, s5) 9
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
```

A: ONOS Summary only show the links between switches. There are 4 links between 5 switches and all of them are bidirectional which means there are 8 links between all switches.

2 . Mininet Usage:

Q: Why sudo mn failed at root?

```
sylvia@root > sudo mn
Password:
Error executing command: login failed
sylvia@root >
```

A: Should user personal account to use sudo command. (Reference: <https://www.twblogs.net/a/5b7f4e1d2b717767c6aeefb5>)