

Part 1

- When ONOS activates org.onosproject.openflow, what apps does it activate?

When ONOS activates org.onosproject.openflow, it activates the following apps:

- (a) org.onosproject.hostprovider
- (b) org.onosproject.lldpprovider
- (c) org.onosproject.optical-model
- (d) org.onosproject.openflow-base

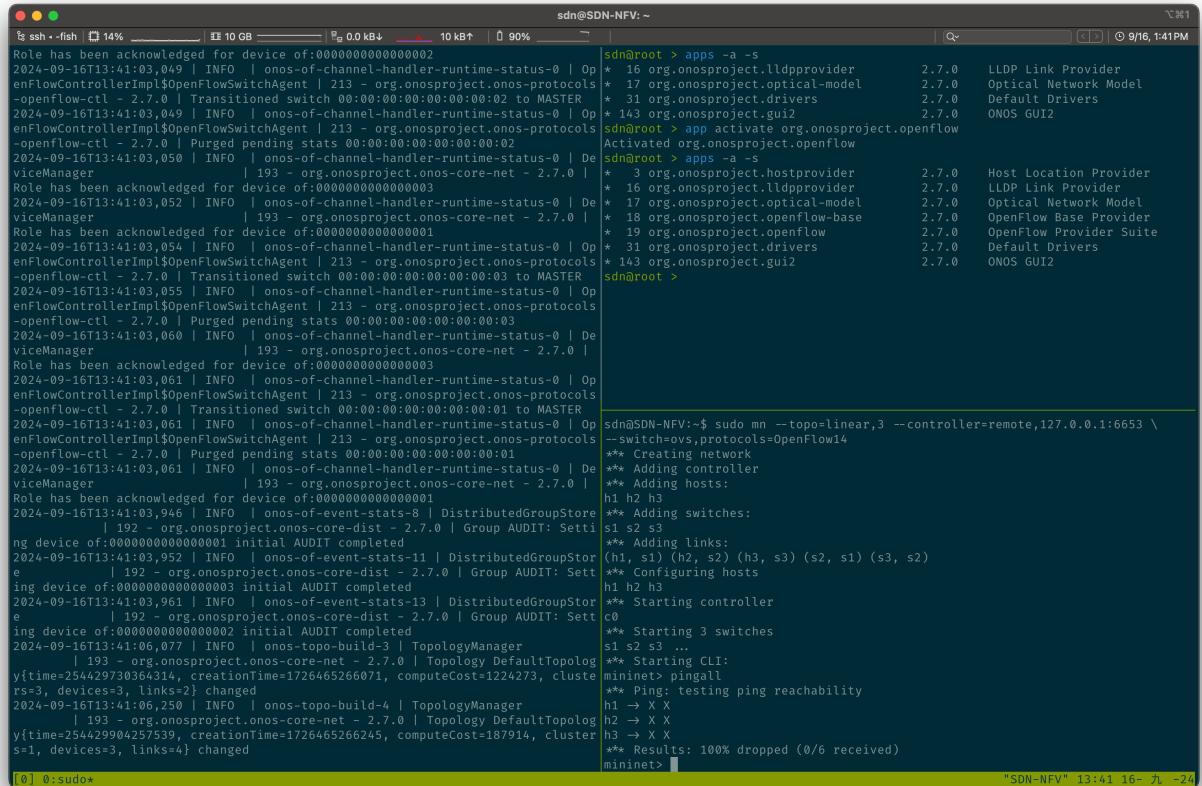
```

sdn@SDN-NFV:~ ssh -l fish | 14% 11 GB 0.0 kB↓ 7.2 kB↑ 0 55% 
2024-09-16T15:09:02,934 | INFO | features-3-thread-1 | OpenFlowDeviceProvider
| 237 - org.onosproject.onos-providers-openflow-device - 2.7.0 | Star
ted
2024-09-16T15:09:02,934 | INFO | features-3-thread-1 | FeaturesServiceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | Done.
2024-09-16T15:09:02,935 | INFO | onos-store-app-app-activation | ApplicationMa
nager
| 193 - org.onosproject.onos-core-net - 2.7.0 | Application
org.onosproject.openflow-base has been activated
2024-09-16T15:09:02,936 | INFO | onos-store-app-app-activation | FeaturesServ
iceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | Adding feat
ures: onos-providers-openflow-app/[2.7.0,2.7.0]
2024-09-16T15:09:03,093 | INFO | features-3-thread-1 | FeaturesServiceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | No deployment change.
2024-09-16T15:09:03,096 | INFO | features-3-thread-1 | FeaturesServiceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | Done.
2024-09-16T15:09:103 | INFO | onos-store-app-app-activation | ApplicationMa
nager
| 193 - org.onosproject.onos-core-net - 2.7.0 | Application
org.onosproject.openflow-base has been activated
2024-09-16T15:09:185 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.openflow.controller.impl.OpenFlowControllerImpl) | Control
ler
| 236 - org.onosproject.onos-protocols-openflow-ctl -
2.7.0 | Using Nio transport
2024-09-16T15:09:03,186 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.openflow.controller.impl.OpenFlowControllerImpl) | Control
ler
| 236 - org.onosproject.onos-protocols-openflow-ctl -
2.7.0 | OpenFlow TLS Params: TlsParams{tlsMode=disabled, ksLocation=null, tsLo
cation=null}
2024-09-16T15:09:03,189 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.meter.impl.OpenFlowMeterProvider) | OpenFlowMe
terProvider
| 240 - org.onosproject.onos-providers-openflow-meter -
2.7.0 | Configured, forceStatsAfterMeterRemoval is enabled
2024-09-16T15:09:03,190 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.device.impl.OpenFlowDeviceProvider) | OpenFlow
DeviceProvider
| 237 - org.onosproject.onos-providers-openflow-device -
2.7.0 | Settings: portStatsPollFrequency=5
2024-09-16T15:09:03,191 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule
Provider
| 238 - org.onosproject.onos-providers-openflow-flow - 2.7
.0 | Settings: flowPollFrequency=5
2024-09-16T15:09:03,191 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule
Provider
| 238 - org.onosproject.onos-providers-openflow-flow - 2.7
.0 | Settings: adaptiveFlowSampling=false
2024-09-16T15:09:03,191 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule
Provider
| 238 - org.onosproject.onos-providers-openflow-flow - 2.7
.0 | Settings: pollStatsPeriodically=true
[0] 0:bash* "SDN-NFV" 15:09 16- 九 -24

```

2. After we activate ONOS and run the command on page 18 in Mininet, will H1 ping H2 successfully?
Why or why not?

H1 cannot ping H2 successfully because there are **no flows** installed on the data plane, so the traffic cannot be forwarded appropriately.



```

sdn@SDN-NFV:~ ssh -l fish 14% 10 GB 0.0 kB↓ 10 kB↑ 0 90% ~ | Qw | (c) | 9/16, 1:41PM
Role has been acknowledged for device of:00000000000000000000000000000000
2024-09-16T13:41:03,049 | INFO | onos-of-channel-handler-runtime-status-0 | OpenFlowControllerImpl$OpenFlowSwitchAgent | 213 - org.onosproject.onos.protocols
-enopenflow-ctrl - 2.7.0 | Transited switch 00:00:00:00:00:00:02 to MASTER
2024-09-16T13:41:03,049 | INFO | onos-of-channel-handler-runtime-status-0 | OpenFlowControllerImpl$OpenFlowSwitchAgent | 213 - org.onosproject.onos.protocols
-enopenflow-ctrl - 2.7.0 | Purged pending stats 00:00:00:00:00:00:02
2024-09-16T13:41:03,050 | INFO | onos-of-channel-handler-runtime-status-0 | DeviceManager
| 193 - org.onosproject.onos.core.net - 2.7.0 | Role has been acknowledged for device of:00000000000000000000000000000000
2024-09-16T13:41:03,052 | INFO | onos-of-channel-handler-runtime-status-0 | DeviceManager
| 193 - org.onosproject.onos.core.net - 2.7.0 | Role has been acknowledged for device of:00000000000000000000000000000001
2024-09-16T13:41:03,054 | INFO | onos-of-channel-handler-runtime-status-0 | OpenFlowControllerImpl$OpenFlowSwitchAgent | 213 - org.onosproject.onos.protocols
-enopenflow-ctrl - 2.7.0 | Transited switch 00:00:00:00:00:00:03 to MASTER
2024-09-16T13:41:03,055 | INFO | onos-of-channel-handler-runtime-status-0 | OpenFlowControllerImpl$OpenFlowSwitchAgent | 213 - org.onosproject.onos.protocols
-enopenflow-ctrl - 2.7.0 | Purged pending stats 00:00:00:00:00:00:03
2024-09-16T13:41:03,060 | INFO | onos-of-channel-handler-runtime-status-0 | DeviceManager
| 193 - org.onosproject.onos.core.net - 2.7.0 | Role has been acknowledged for device of:00000000000000000000000000000000
2024-09-16T13:41:03,061 | INFO | onos-of-channel-handler-runtime-status-0 | OpenFlowControllerImpl$OpenFlowSwitchAgent | 213 - org.onosproject.onos.protocols
-enopenflow-ctrl - 2.7.0 | Transited switch 00:00:00:00:00:01 to MASTER
2024-09-16T13:41:03,061 | INFO | onos-of-channel-handler-runtime-status-0 | OpenFlowControllerImpl$OpenFlowSwitchAgent | 213 - org.onosproject.onos.protocols
-enopenflow-ctrl - 2.7.0 | Purged pending stats 00:00:00:00:00:01
2024-09-16T13:41:03,061 | INFO | onos-of-channel-handler-runtime-status-0 | DeviceManager
| 193 - org.onosproject.onos.core.net - 2.7.0 | Role has been acknowledged for device of:00000000000000000000000000000001
2024-09-16T13:41:03,946 | INFO | onos-of-event-stats-8 | DistributedGroupStore
| 192 - org.onosproject.onos.core.dist - 2.7.0 | Group AUDIT: Setting device of:0000000000000001 initial AUDIT completed
2024-09-16T13:41:03,952 | INFO | onos-of-event-stats-11 | DistributedGroupStore
| 192 - org.onosproject.onos.core.dist - 2.7.0 | Group AUDIT: Setting device of:0000000000000000 initial AUDIT completed
2024-09-16T13:41:03,961 | INFO | onos-of-event-stats-13 | DistributedGroupStore
| 192 - org.onosproject.onos.core.dist - 2.7.0 | Group AUDIT: Setting device of:0000000000000002 initial AUDIT completed
2024-09-16T13:41:06,077 | INFO | onos-topo-build-3 | TopologyManager
| 193 - org.onosproject.onos.core.net - 2.7.0 | Topology DefaultTopolog
y{time=254429730364314, creationTime=1726465266071, computeCost=1224273, cluster
rs=3, devices=3, links=2} changed
2024-09-16T13:41:06,250 | INFO | onos-topo-build-4 | TopologyManager
| 193 - org.onosproject.onos.core.net - 2.7.0 | Topology DefaultTopolog
y{time=254429904257539, creationTime=1726465266245, computeCost=187914, cluster
s=1, devices=3, links=4} changed
[0] 0:sudo*

```

sdn@SDN-NFV:~ \$ sudo mn --topo=linear,3 --controller=remote,127.0.0.1:6653 \
--switch=ovs,protocols=OpenFlow14
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 → X X
h2 → X X
h3 → X X
*** Results: 100% dropped (0/6 received)
mininet> [0] 0:mn

"SDN-NFV" 13:41 16- 九 -24

3. Which TCP port does the controller listen to for OpenFlow connection requests from the switch?

Using the `netstat` command, we can determine that the controller listens on ports **6633** and **6653** for OpenFlow connection requests from the switch.

```

sdn@SDN-NFV:~$ ssh -l fish | 13% 11 GB 0.0 kB↑ 0.0 kB↑ 85% | sdn@SDN-NFV:~$ 
2024-09-16T13:51:05,487 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Host re* 16 org.onosproject.lldpprovider 2.7.0 LLDP Link Provider
moval on port/device down events is not configured, using current value of true* 17 org.onosproject.optical-model 2.7.0 Optical Network Model
2024-09-16T13:51:05,487 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Configu* 31 org.onosproject.drivers 2.7.0 Default Drivers
red. Using ARP is enabled* 143 org.onosproject.gui2 2.7.0 ONOS GUI2
2024-09-16T13:51:05,488 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Configu* 16 org.onosproject.lldpprovider 2.7.0 Host Location Provider
red. Using IPv6 NDP Neighbor Solicitation and Advertisement is disabled* 17 org.onosproject.optical-model 2.7.0 LLDP Link Provider
2024-09-16T13:51:05,488 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Configu* 18 org.onosproject.openflow-base 2.7.0 Optical Network Model
red. Using IPv6 NDP Router Solicitation and Advertisement is disabled* 19 org.onosproject.openflow 2.7.0 OpenFlow Base Provider
2024-09-16T13:51:05,488 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Configu* 31 org.onosproject.drivers 2.7.0 Default Drivers
red. Using DHCP is disabled* 143 org.onosproject.gui2 2.7.0 ONOS GUI2
2024-09-16T13:51:05,488 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Configu* 16 org.onosproject.lldpprovider 2.7.0 Host Location Provider
red. Request intercepts is enabled* 17 org.onosproject.optical-model 2.7.0 LLDP Link Provider
2024-09-16T13:51:05,488 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.host.impl.HostlocationProvider) | HostlocationPro
vider | 219 - org.onosproject.onos-providers-host - 2.7.0 | Configu* 18 org.onosproject.openflow-base 2.7.0 Optical Network Model
red. Multihoming is disabled* 19 org.onosproject.openflow 2.7.0 OpenFlow Base Provider
2024-09-16T13:51:05,495 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.of.device.impl.OpenFlowDeviceProvider) | OpenFlow
DeviceProvider | 221 - org.onosproject.onos-providers-openflow-device* 31 org.onosproject.drivers 2.7.0 Default Drivers
- 2.7.0 | Settings: portStatsPollFrequency=5* 143 org.onosproject.gui2 2.7.0 ONOS GUI2
2024-09-16T13:51:05,496 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule
Provider | 222 - org.onosproject.onos-providers-openflow-flow - 2.7* 16 org.onosproject.lldpprovider 2.7.0 Host Location Provider
.0 | Settings: flowPollFrequency=5* 17 org.onosproject.optical-model 2.7.0 LLDP Link Provider
2024-09-16T13:51:05,497 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule
Provider | 222 - org.onosproject.onos-providers-openflow-flow - 2.7* 18 org.onosproject.openflow-base 2.7.0 Optical Network Model
.0 | Settings: adaptiveFlowSampling=false* 19 org.onosproject.openflow 2.7.0 OpenFlow Base Provider
2024-09-16T13:51:05,497 | INFO | CM Event Dispatcher (Fire ConfigurationEvent: pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule
Provider | 222 - org.onosproject.onos-providers-openflow-flow - 2.7* 31 org.onosproject.drivers 2.7.0 Default Drivers
.0 | Settings: pollStatsPeriodically=true* 143 org.onosproject.gui2 2.7.0 ONOS GUI2
[0] 0: bash* sdn@SDN-NFV:~/tmp$ sudo netstat -nlpt > netstat_output_1.txt
sdn@SDN-NFV:~/tmp$ sudo netstat -nlpt > netstat_output_2.txt
sdn@SDN-NFV:~/tmp$ diff netstat_output_1.txt netstat_output_2.txt
12a13,14
> tcp6      0      0 :::6633          ::*:*
46779/java
> tcp6      0      0 :::6653          ::*:*
46779/java
sdn@SDN-NFV:~/tmp$ 

```

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

By using the netstat command to check the apps one by one, we found that the app **org.onosproject.openflow-base** enables the controller to listen on the TCP port.

```

sdn@SDN-NFV: ~
ssh -l fish | 14% | 11 GB | 0.0 kB↑ | 0.0 kB↑ | 83% | sdn@SDN-NFV: ~
ted | 227 - org.onosproject.onos-providers-openflow-device - 2.7.0 | Star
2024-09-16T13:56:03,228 | INFO | features-3-thread-1 | FeaturesServiceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | org.onosproject.onos-
s-providers-openflow-meter/2.7.0
2024-09-16T13:56:03,235 | INFO | features-3-thread-1 | OpenFlowMeterProvider
| 230 - org.onosproject.onos-providers-openflow-meter - 2.7.0 | Configured.
forceStatsAfterMeterRemoval is enabled
2024-09-16T13:56:03,239 | INFO | features-3-thread-1 | FeaturesServiceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | org.onosproject.onos-
s-providers-openflow-meter/2.7.0
2024-09-16T13:56:03,246 | INFO | features-3-thread-1 | OpenFlowPacketProvider
| 231 - org.onosproject.onos-providers-openflow-packet - 2.7.0 | Star
ted
2024-09-16T13:56:03,248 | INFO | features-3-thread-1 | FeaturesServiceImpl
| 11 - org.apache.karaf.features.core - 4.2.9 | Done.
2024-09-16T13:56:03,249 | INFO | onos-store-app-app-activation | ApplicationMa-
nager
| 193 - org.onosproject.onos-core-net - 2.7.0 | Application
org.onosproject.openflow-base has been activated
2024-09-16T13:56:03,493 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.openflow.controller.impl.OpenFlowControllerImpl) | Control-
ler
| 226 - org.onosproject.onos-protocols-openflow-ctl -
2.7.0 | Using Nio transport
2024-09-16T13:56:03,494 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.openflow.controller.impl.OpenFlowControllerImpl) | Control-
ler
| 226 - org.onosproject.onos-protocols-openflow-ctl -
2.7.0 | OpenFlow TLS Params: TlsParams{tlsMode=disabled, ksLocation=null, tsLo-
cation=null}
2024-09-16T13:56:03,497 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.meter.impl.OpenFlowMeterProvider) | OpenFlowMe-
terProvider
| 230 - org.onosproject.onos-providers-openflow-meter -
2.7.0 | Configured. forceStatsAfterMeterRemoval is enabled
2024-09-16T13:56:03,499 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.device.impl.OpenFlowDeviceProvider) | OpenFlow-
DeviceProvider
| 227 - org.onosproject.onos-providers-openflow-device -
2.7.0 | Settings: portStatsPollFrequency=5
2024-09-16T13:56:03,500 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule-
Provider
| 228 - org.onosproject.onos-providers-openflow-flow -
2.7.0 | Settings: flowPollFrequency=5
2024-09-16T13:56:03,501 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule-
Provider
| 228 - org.onosproject.onos-providers-openflow-flow -
2.7.0 | Settings: adaptiveFlowSampling=false
2024-09-16T13:56:03,501 | INFO | CM Event Dispatcher (Fire ConfigurationEvent:
pid=org.onosproject.provider.of.flow.impl.OpenFlowRuleProvider) | OpenFlowRule-
Provider
| 228 - org.onosproject.onos-providers-openflow-flow -
2.7.0 | Settings: pollStatsPeriodically=true
[0] 0:bash* sdn@SDN-NFV:~/tmp$ sudo netstat -nlpt > netstat_output_3.txt
sdn@SDN-NFV:~/tmp$ sudo netstat -nlpt > netstat_output_4.txt
sdn@SDN-NFV:~/tmp$ diff netstat_output_3.txt netstat_output_4.txt
12a13,14
> tcp6      0      0 :::6633          ::*:*
16779/java
> tcp6      0      0 :::6653          ::*:*
16779/java
sdn@SDN-NFV:~/tmp$ LISTEN
LISTEN
sdn@SDN-NFV:~/tmp$ "SDN-NFV" 13:56 16- 九 -24

```

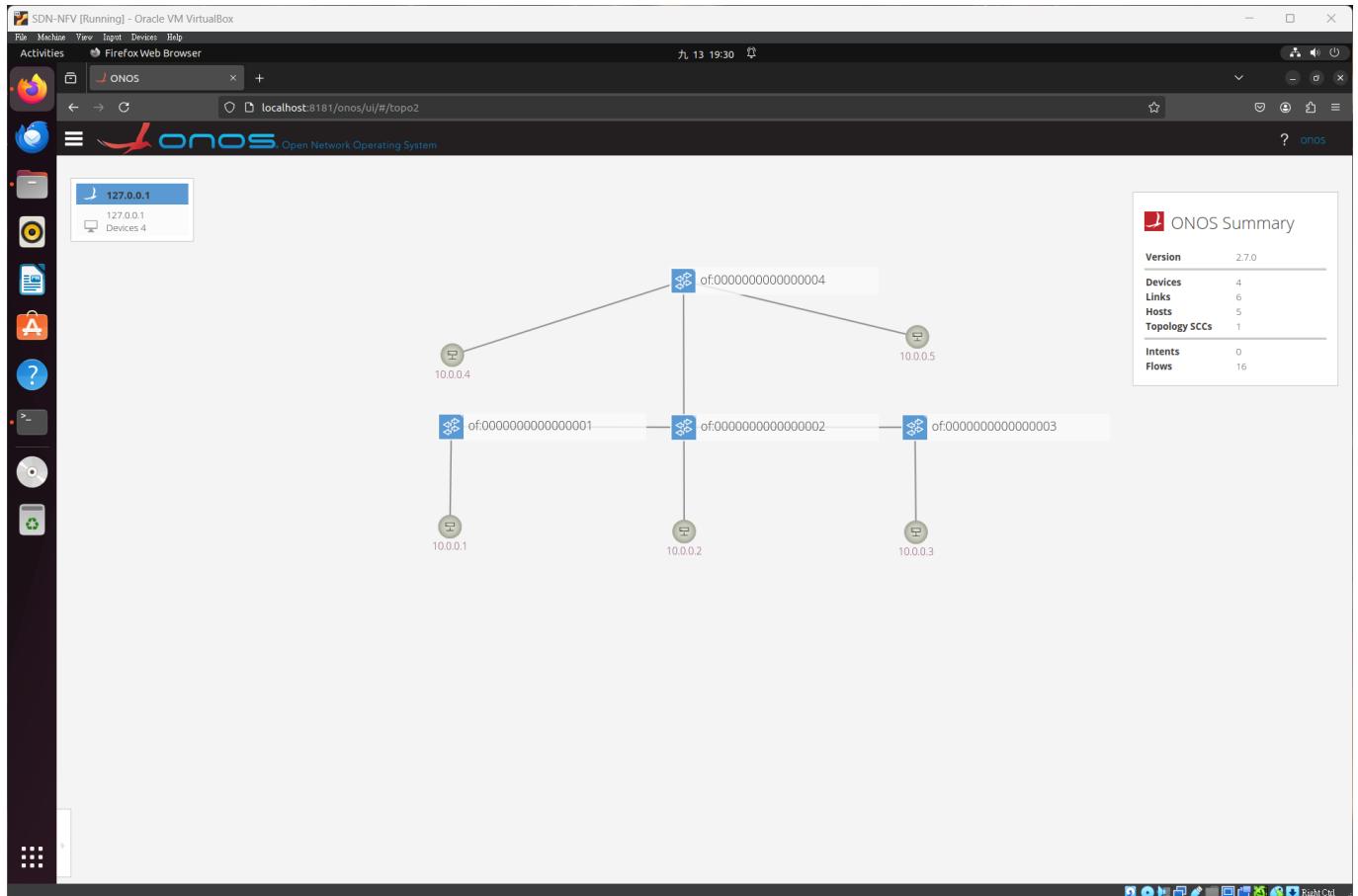
Part 2

```
1 from mininet.topo import Topo
2
3 class Lab1_Topo_112550013( Topo ):
4     def __init__( self ):
5         Topo.__init__( self )
6
7         # Add hosts
8         h1 = self.addHost( 'h1' )
9         h2 = self.addHost( 'h2' )
10        h3 = self.addHost( 'h3' )
11        h4 = self.addHost( 'h4' )
12        h5 = self.addHost( 'h5' )
13
14         # Add switches
15         s1 = self.addSwitch( 's1' )
16         s2 = self.addSwitch( 's2' )
17         s3 = self.addSwitch( 's3' )
18         s4 = self.addSwitch( 's4' )
19
20         # Add links
21         self.addLink( h1, s1 )
22         self.addLink( h2, s2 )
23         self.addLink( h3, s3 )
24         self.addLink( h4, s4 )
25         self.addLink( h5, s4 )
26
27         self.addLink( s1, s2 )
28         self.addLink( s2, s3 )
29         self.addLink( s2, s4 )
30
31
32 topos = { 'topo_part2_112550013': Lab1_Topo_112550013 }
```

To build the topology, we should use the addHost command to add five hosts, the addSwitch command to add four switches, and use the addLink command to connect them. Then, we should use the following command to execute:

```
1| sudo mn --custom=lab1_part2_112550013.py --topo=topo_part2_112550013 --controller=remote,ip=127.0.0.1:6653 --switch=ovs,protocols=OpenFlow14
```

After we build the topology, we can check the topology on the GUI of ONOS.

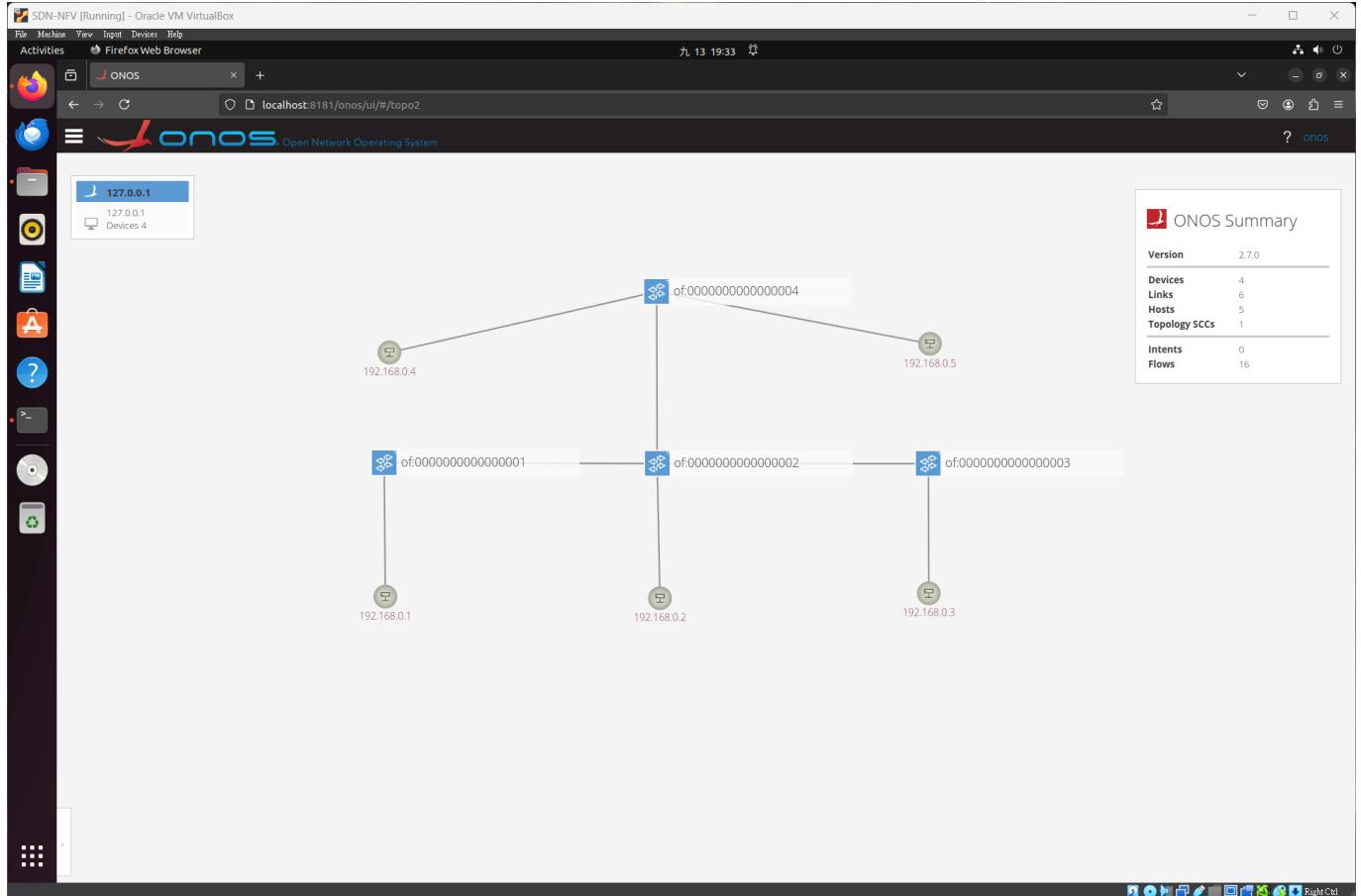


Part 3

```
1 from mininet.topo import Topo
2
3 class Lab1_Topo_112550013( Topo ):
4     def __init__( self ):
5         Topo.__init__( self )
6
7         # Define IP addresses for each host
8         ip_addresses = {
9             'h1': '192.168.0.1/27',
10            'h2': '192.168.0.2/27',
11            'h3': '192.168.0.3/27',
12            'h4': '192.168.0.4/27',
13            'h5': '192.168.0.5/27'
14        }
15
16        # Add hosts with specific IP addresses
17        hosts = { name: self.addHost(name, ip=ip_addresses[name], defaultRoute='
18                                     via 192.168.0.30') for name in ip_addresses }
19
20        # Add switches
21        s1 = self.addSwitch( 's1' )
22        s2 = self.addSwitch( 's2' )
23        s3 = self.addSwitch( 's3' )
24        s4 = self.addSwitch( 's4' )
25
26        # Add links
27        self.addLink(hosts['h1'], s1)
28        self.addLink(hosts['h2'], s2)
29        self.addLink(hosts['h3'], s3)
30        self.addLink(hosts['h4'], s4)
31        self.addLink(hosts['h5'], s4)
32
33        self.addLink( s1, s2 )
34        self.addLink( s2, s3 )
35        self.addLink( s2, s4 )
36
37 topos = { 'topo_part3_112550013': Lab1_Topo_112550013 }
```

Different from Part 2, we should first set ip addresses for each host in Part 3. Then, we can use the following command to execute:

```
1 | sudo mn --custom=lab1_part3_112550013.py --topo=topo_part3_112550013 --controller=remote,ip=127.0.0.1:6653 --switch=ovs,protocols=OpenFlow14
```



After we build the topology, we can check the ip addresses of each host on the GUI of ONOS. Also, we can use the following command to check the details of the ip addresses of each host:

```
1 mininet> dump
2 mininet> h1 ifconfig
3 mininet> h2 ifconfig
4 mininet> h3 ifconfig
5 mininet> h4 ifconfig
6 mininet> h5 ifconfig
```

```
sdn@SDN-NFV: ~/Desktop/NYCU_SDN-and-NFV/lab01
File Machine View Input Devices Help
Activities Terminal
sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/onos

mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=7384>
<Host h2: h2-eth0:192.168.0.2 pid=7386>
<Host h3: h3-eth0:192.168.0.3 pid=7388>
<Host h4: h4-eth0:192.168.0.4 pid=7389>
<Host h5: h5-eth0:192.168.0.5 pid=7392>
<OVSSwitch[ 'protocols': 'OpenFlow14' ] s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=7397>
<OVSSwitch[ 'protocols': 'OpenFlow14' ] s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None pid=7400>
<OVSSwitch[ 'protocols': 'OpenFlow14' ] s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=7403>
<OVSSwitch[ 'protocols': 'OpenFlow14' ] s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=7406>
<RemoteController['ip': '127.0.0.1:6653'] c0: 127.0.0.1:6653 pid=7378>
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::2c03:5bff:fe0f:93b0 prefixlen 64 scopedid 0x20<link>
        ether 08:00:2p:03:5b:ff brd ff:ff:ff:ff:ff:ff txqueuelen 1000 (Ethernet)
                RX packets 131 bytes 16035 (16.0 KB)
                RX errors 0 dropped 84 overruns 0 frame 0
                TX packets 27 bytes 1986 (1.9 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 scopedid 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::94c6:97ff:feeb:edff prefixlen 64 scopedid 0x20<link>
        ether 96:c6:97:eb:ed:ff txqueuelen 1000 (Ethernet)
                RX packets 145 bytes 17880 (17.8 KB)
                RX errors 0 dropped 96 overruns 0 frame 0
                TX packets 27 bytes 1986 (1.9 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 scopedid 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::4cf8:a0ff:fed2:ca12 prefixlen 64 scopedid 0x20<link>
        ether 4e:f8:a0:d2:ca:12 txqueuelen 1000 (Ethernet)
                RX packets 148 bytes 18200 (18.2 KB)
                RX errors 0 dropped 98 overruns 0 frame 0
                TX packets 27 bytes 1986 (1.9 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
        lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

```

sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/onos
sdn@SDN-NFV: ~/Desktop/NYCU_SDN-and-NFV/lab01

mininet> h3 ifconfig
h3-eth0: flags=4163 mtu 1500
    inet 192.168.0.3 netmask 255.255.255.224 broadcast 192.168.0.31
        inet6 fe80::4cf8:a0ff:fed2:ca12 prefixlen 64 scopeid 0x20<link>
            ether 4e:f8:a0:d2:ca:12 txqueuelen 1000 (Ethernet)
            RX packets 148 bytes 18200 (18.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 27 bytes 1986 (1.9 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 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 mtu 1500
    inet 192.168.0.31 netmask 255.255.255.224 broadcast 192.168.0.31
        inet6 fe80::a479:50ff:fe43:f59a prefixlen 64 scopeid 0x20<link>
            ether a6:79:50:43:f5:9a txqueuelen 1000 (Ethernet)
            RX packets 156 bytes 19312 (19.3 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

lo: flags=73 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 mtu 1500
    inet 192.168.0.5 netmask 255.255.255.224 broadcast 192.168.0.31
        inet6 fe80::4423:64ff:fe59:33ab prefixlen 64 scopeid 0x20<link>
            ether d6:23:64:59:33:ab txqueuelen 1000 (Ethernet)
            RX packets 164 bytes 20424 (20.4 KB)
            RX errors 0 dropped 114 overruns 0 frame 0
            TX packets 27 bytes 1986 (1.9 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>

```

What you've learned or solved

After this lab, I have learned how to build a topology using Mininet and how to check the details of the topology on the GUI of ONOS. Also, I have learned how to set ip addresses for each host and how to check the details of the ip addresses of each host.