# SDNFV – Lab 1

109550122 王宇晨

Part 1:

1.

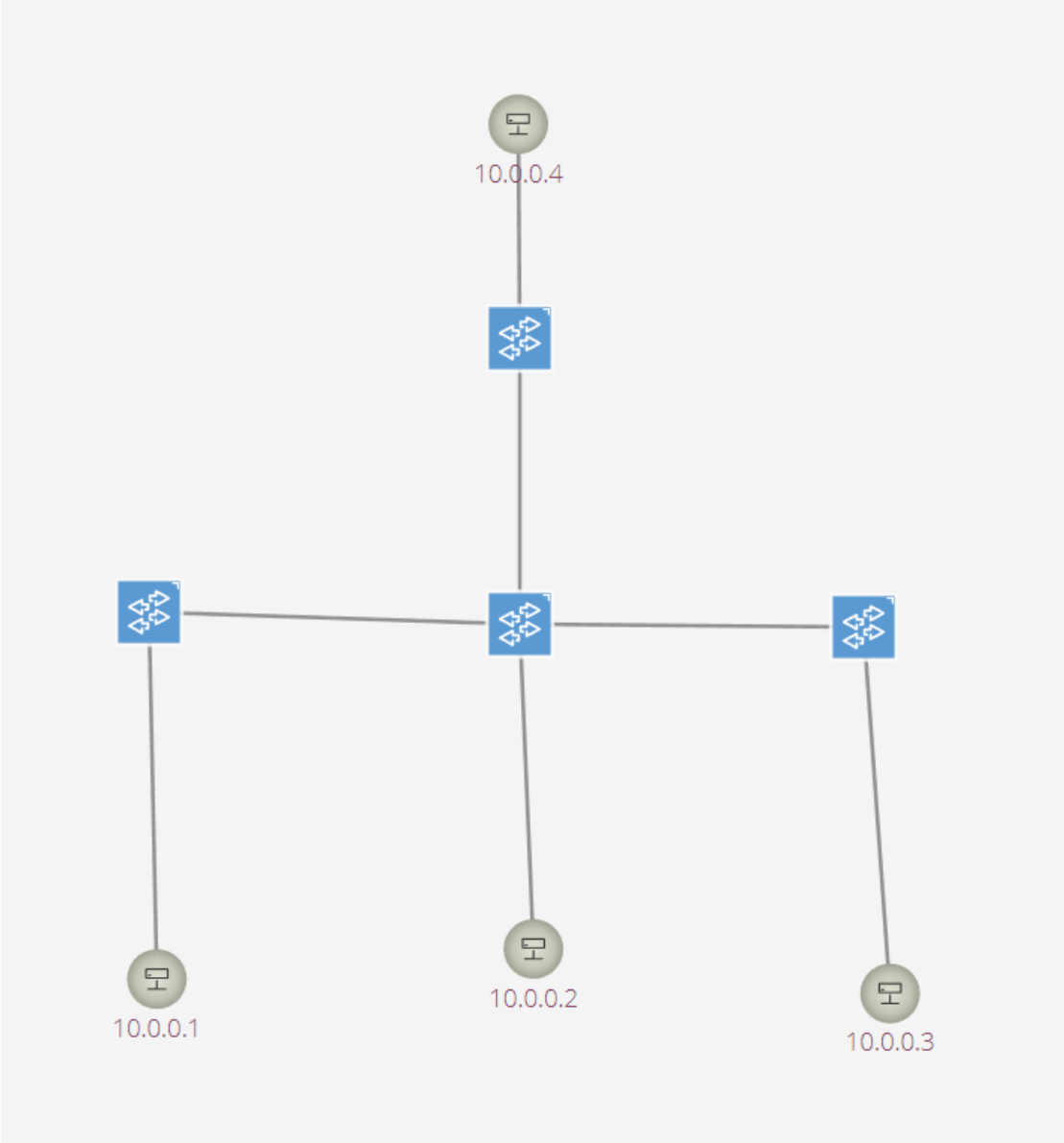
org.onosproject.optical-model
org.onosproject.hostprovider
org.onosproject.lldpprovider
org.onosproject.openflow-base

2. No, because I haven't activate the Reactive Forwarding app (org.onosproject.fwd) yet.
3. 6653

```
❯ sudo netstat -nlpt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:42961         0.0.0.0:*               LISTEN      33221/node
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      22312/cupsd
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      48093/java
tcp        0      0 127.0.0.1:44663         0.0.0.0:*               LISTEN      36561/node
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      506/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      645/sshd: /usr/sbin
tcp6       0      0 :::8101                 :::*                    LISTEN      48093/java
tcp6       0      0 :::8181                 :::*                    LISTEN      48093/java
tcp6       0      0 127.0.0.1:46495         :::*                    LISTEN      48093/java
tcp6       0      0 :::35295                :::*                    LISTEN      48093/java
tcp6       0      0 ::1:43175               :::*                    LISTEN      38771/bazel(onos)
tcp6       0      0 :::9876                 :::*                    LISTEN      48093/java
tcp6       0      0 :::1099                 :::*                    LISTEN      48093/java
tcp6       0      0 ::1:631                 :::*                    LISTEN      22312/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      645/sshd: /usr/sbin
❯ sudo netstat -nlpt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:42961         0.0.0.0:*               LISTEN      33221/node
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      22312/cupsd
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      48093/java
tcp        0      0 127.0.0.1:44663         0.0.0.0:*               LISTEN      36561/node
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      506/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      645/sshd: /usr/sbin
tcp6       0      0 :::8101                 :::*                    LISTEN      48093/java
tcp6       0      0 :::8181                 :::*                    LISTEN      48093/java
tcp6       0      0 127.0.0.1:46495         :::*                    LISTEN      48093/java
tcp6       0      0 :::35295                :::*                    LISTEN      48093/java
tcp6       0      0 :::6653                 :::*                    LISTEN      48093/java
tcp6       0      0 :::6633                 :::*                    LISTEN      48093/java
tcp6       0      0 ::1:43175               :::*                    LISTEN      38771/bazel(onos)
tcp6       0      0 :::9876                 :::*                    LISTEN      48093/java
tcp6       0      0 :::1099                 :::*                    LISTEN      48093/java
tcp6       0      0 ::1:631                 :::*                    LISTEN      22312/cupsd
tcp6       0      0 :::22                   :::*                    LISTEN      645/sshd: /usr/sbin
```

4. org.onosproject.openflow-base/org.onosproject.optical-model

Part 2:




I wrote a Python script to build the topology and the hosts can ping each other

Part 3:

```
mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=112752>
<Host h2: h2-eth0:192.168.0.2 pid=112758>
<Host h3: h3-eth0:192.168.0.3 pid=112766>
<Host h4: h4-eth0:192.168.0.4 pid=112771>
<OVSSwitch{'protocols': 'OpenFlow14'} s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=112783>
<OVSSwitch{'protocols': 'OpenFlow14'} s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None pid=112787>
<OVSSwitch{'protocols': 'OpenFlow14'} s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=112790>
<OVSSwitch{'protocols': 'OpenFlow14'} s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=112793>
<RemoteController{'ip': '127.0.0.1:6653'} c0: 127.0.0.1:6653 pid=112735>
```

```
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::e888:caff:feef:356d  prefixlen 64  scopeid 0x20<link>
        ether ea:88:ca:ef:35:6d  txqueuelen 1000  (Ethernet)
        RX packets 138  bytes 17474 (17.4 KB)
        RX errors 0  dropped 96  overruns 0  frame 0
        TX packets 23  bytes 1706 (1.7 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::5430:69ff:fef9:b4dc  prefixlen 64  scopeid 0x20<link>
        ether 56:30:69:f9:b4:dc  txqueuelen 1000  (Ethernet)
        RX packets 178  bytes 23034 (23.0 KB)
        RX errors 0  dropped 136  overruns 0  frame 0
        TX packets 23  bytes 1706 (1.7 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::18e2:b2ff:fe87:609b  prefixlen 64  scopeid 0x20<link>
        ether 1a:e2:b2:87:60:9b  txqueuelen 1000  (Ethernet)
        RX packets 180  bytes 23312 (23.3 KB)
        RX errors 0  dropped 138  overruns 0  frame 0
        TX packets 23  bytes 1706 (1.7 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::e838:10ff:fe35:f99c  prefixlen 64  scopeid 0x20<link>
        ether ea:38:10:35:f9:9c  txqueuelen 1000  (Ethernet)
        RX packets 198  bytes 25814 (25.8 KB)
        RX errors 0  dropped 156  overruns 0  frame 0
        TX packets 23  bytes 1706 (1.7 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
```

I wrote a Python script to build the topology, set the IP addresses and the netmask of the hosts and the hosts can ping each other

What I have learned or solved:

I have learned basic knowledge of ONOS, mininet and creating topology with desired IP addresses/netmask of the hosts.