

# Softwarized and Virtualized Mobile Networks Exercises

Vinci Nicolò

10 July 2022

Exercises proposed for the new network topology system. The Docker images are already provided and they are:

- *dev\_host*: image used for a generic host.
- *dev\_server*: image used for running a Docker in Docker virtualization.

## 1 Level easy

### 1.1 Question

Build a network topology yaml file with:

- Automatic MAC addresses enabled.
- Automatic ARP tables enabled.
- Local Controller.
- Switch sw0.
- Client c0 attached to sw0.
- Server s0 attached to sw0.

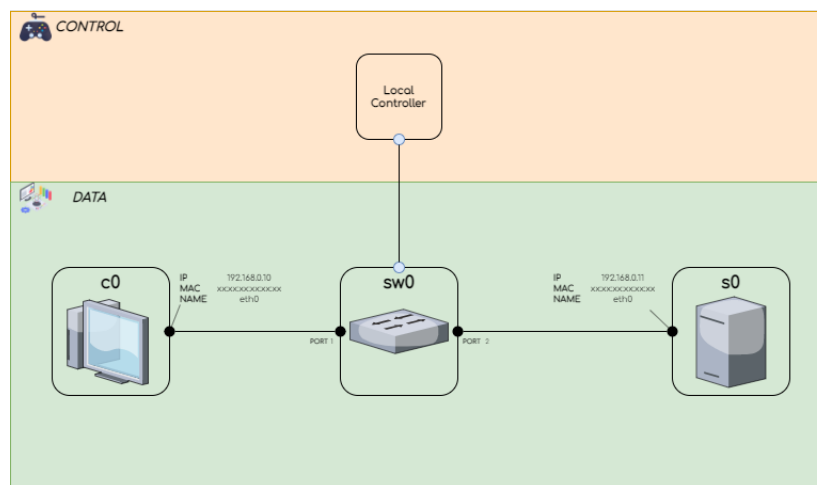


Figure 1: Network easy

## 1.2 Solution

The solution is located at *src/exercises/easy* folder in the file *topology.yaml*.

```
1 network:
2   autoMac: true
3   autoArp: true
4
5 controllers:
6   - name: controller0
7     type: ControllerLocal
8
9 switches:
10  - name: sw0
11    links:
12      - node: c0
13        - node: s0
14
15 hosts:
16   - name: c0
17     ip: 192.168.0.10/24
18   - name: s0
19     ip: 192.168.0.11/24
```

Figure 2: Network topology solution

```
mininet> dump
<DockerHost c0: c0-eth0:192.168.0.10 pid=2599>
<DockerHost s0: s0-eth0:192.168.0.11 pid=2711>
<OVSSwitch sw0: lo:127.0.0.1,sw0-eth1:None,sw0-eth2:None pid=2541>
<Controller controller0: 127.0.0.1:6653 pid=2533>
```

Figure 3: Mininet dump

## 2 Level medium

### 2.1 Question

Build a network topology yaml file with:

- Automatic MAC addresses disabled.
- Automatic ARP tables disabled.
- Local Controller.
- Switch sw0.
- Client c0 with custom MAC address attached to sw0 defining link bandwidth and delay.
- Server s0 with custom MAC address attached to sw0 defining link bandwidth and delay.
- Switch sw1.

- Client c1 with custom MAC address attached to sw1 defining link bandwidth and delay.
- Server s1 with custom MAC addresses attached to sw0 and sw1 defining link bandwidths and delays.

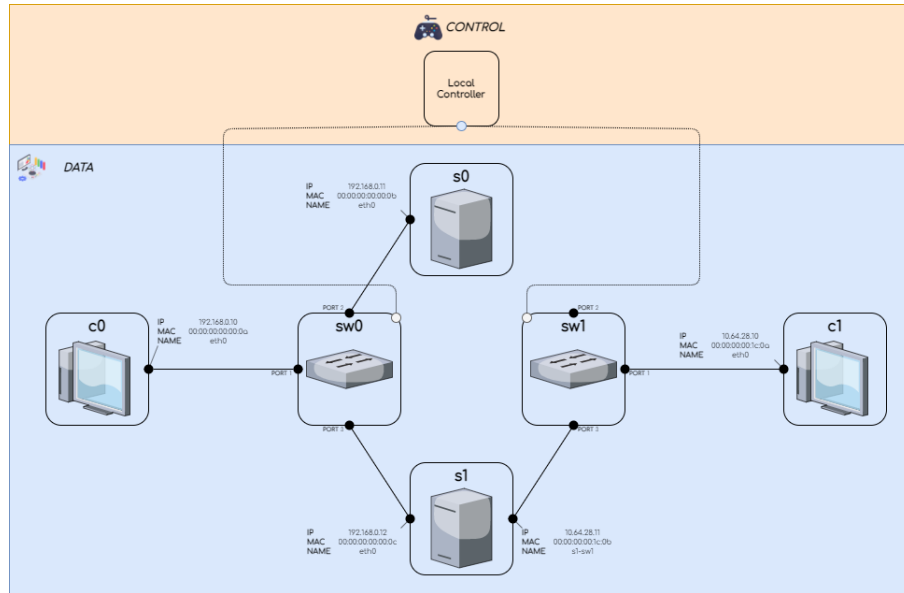


Figure 4: Network medium

## 2.2 Solution

The solution is located at *src/exercises/medium* folder in the file *topology.yaml*.

```

1 network:
2   autoLoc: false
3   autoArp: false
4
5 controllers:
6   - name: controller0
7     type: ControllerLocal
8
9 switches:
10  - name: sw0
11    links:
12      - node: c0
13        bandwidth: 1000
14        delay: 3ms
15      - node: s0
16        bandwidth: 2000
17        delay: 1ms
18      - node: s1
19        bandwidth: 2000
20        delay: 1ms
21  - name: sw1
22    links:
23      - node: c1
24        bandwidth: 1000
25        delay: 3ms
26      - node: s1
27        bandwidth: 2000
28        delay: 1ms
29      fromInterface: sw1-s1
30      toInterface: s1-sw1
31
32 hosts:
33   - name: c0
34     ip: 192.168.0.10/24
35     mac: 00:00:00:00:00:0a
36   - name: c0
37     ip: 192.168.0.11/24
38     mac: 00:00:00:00:00:0b
39   - name: c1
40     ip: 10.64.28.10/24
41     mac: 00:00:00:00:1c:0a
42   - name: s1
43     ip: 192.168.0.12/24
44     mac: 00:00:00:00:00:0c
45     interfaces:
46       - name: s1-sw1
47         ip: 10.64.28.11/24
48         mac: 00:00:00:00:1c:0b

```

Figure 5: Network topology solution

```

mininet> dump
<DockerHost c0: c0-eth0:192.168.0.10 pid=2623>
<DockerHost s0: s0-eth0:192.168.0.11 pid=2732>
<DockerHost c1: c1-eth0:10.64.28.10 pid=2838>
<DockerHost s1: s1-eth0:192.168.0.12,s1-sw1:None pid=2946>
<OVSSwitch sw0: lo:127.0.0.1,sw0-eth1:None,sw0-eth2:None,sw0-eth3:None pid=2563>
<OVSSwitch sw1: lo:127.0.0.1,sw1-eth1:None,sw1-s1:None pid=2566>
<Controller controller0: 127.0.0.1:6653 pid=2555>

```

Figure 6: Mininet dump

## 3 Level difficult

### 3.1 Question

Build a network topology yaml file with:

- Automatic MAC addresses disabled.
- Automatic ARP tables disabled.
- Remote Controller with IP 127.0.0.1 and default port 6653.
- Switch sw0.
- Client c0 with custom MAC address attached to sw0 defining link bandwidth and delay.

- Server s0 with custom MAC address attached to sw0 defining link bandwidth and delay.
- Switch sw1.
- Client c1 with custom MAC address attached to sw1 defining link bandwidth and delay.
- Server s1 with custom MAC addresses attached to sw0 and sw1 defining link bandwidth and delay.
- Server s1 and server s0 run a Docker container (*dev\_server* image).

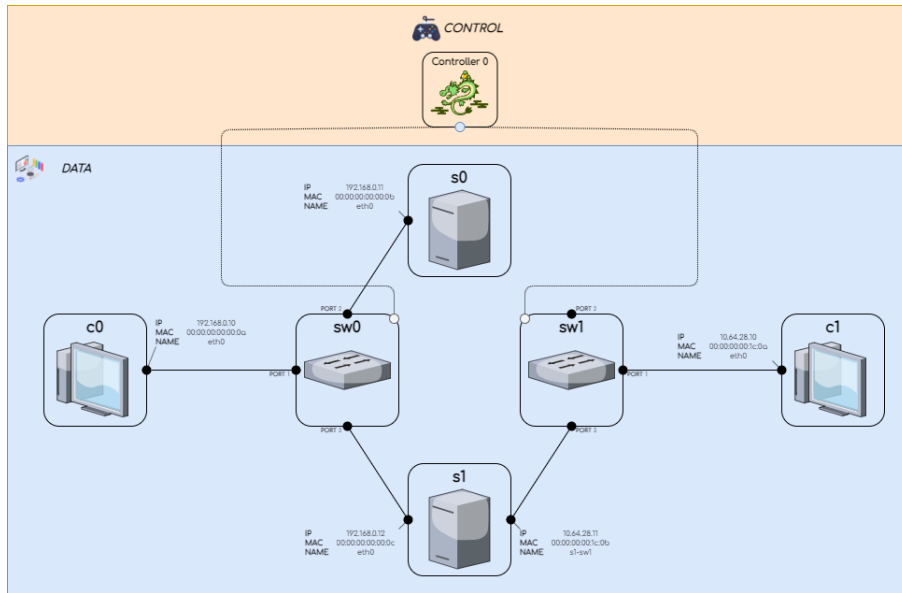


Figure 7: Network difficult

### 3.2 Solution

The solution is located at *src/exercises/difficult* folder in the file *topology.yaml*.

```

1 network:
2   autoMac: false
3   autoArp: false
4
5   controllers:
6     - name: controller0
7       ip: 127.0.0.1
8       port: 6653
9       type: ControllerRemote
10
11   switches:
12     - name: sw0
13       links:
14         - node: c0
15           bandwidth: 1000
16           delay: 1ms
17         - node: s0
18           bandwidth: 2000
19           delay: 1ms
20         - node: s1
21           bandwidth: 2000
22           delay: 1ms
23     - name: sw1
24       links:
25         - node: c1
26           bandwidth: 1000
27           delay: 1ms
28         - node: s1
29           bandwidth: 2000
30           delay: 1ms
31         fromInterface: sw0-s1
32         toInterface: s1-sw1
33
34   hosts:
35     - name: c0
36       ip: 192.168.0.10/24
37       mac: 00:00:00:00:00:0a
38     - name: s0
39       ip: 192.168.0.11/24
40       mac: 00:00:00:00:00:0b
41     containers:
42       - name: server0
43         image: docker:latest
44         cmd: "--host 0.0.0.0 --port 80 --enable"
45         wait: true
46     - name: c1
47       ip: 10.64.28.10/24
48       mac: 00:00:00:00:1c:0a
49     - name: s1
50       ip: 192.168.0.12/24
51       mac: 00:00:00:00:00:0c
52     containers:
53       - name: server1
54         image: docker:latest
55         cmd: "--host 0.0.0.0 --port 80 --enable"
56         wait: true
57     interfaces:
58       - name: s1-sw1
59         ip: 10.64.28.11/24
60         mac: 00:00:00:00:1c:0b

```

Figure 8: Network topology solution

```

mininet> dump
<DockerHost c0: c0-eth0:192.168.0.10 pid=5103>
<DockerHost s0: s0-eth0:192.168.0.11 pid=5217>
<DockerHost c1: c1-eth0:10.64.28.10 pid=5322>
<DockerHost s1: s1-eth0:192.168.0.12, s1-sw1:None pid=5423>
<OVSSwitch sw0: lo:127.0.0.1, sw0-eth1:None, sw0-eth2:None, sw0-eth3:None pid=5048>
<OVSSwitch sw1: lo:127.0.0.1, sw1-eth1:None, sw1-s1:None pid=5051>
<RemoteController controller0: 127.0.0.1:6653 pid=5041>

```

Figure 9: Mininet dump