

به نام خدا

پروژه ی اول درس شبکه / حدیث غفوری 9825413

سوال ۱

ابتدا شبیه ساز را نصب میکنیم.

```
hadis@ubuntu:~$ find mininet/  
mininet/  
mininet/examples  
mininet/examples/__init__.py  
mininet/examples/natnet.py  
mininet/examples/sshd.py  
mininet/examples/scratchnetuser.py  
mininet/examples/mtlink.py  
mininet/examples/linuxrouter.py  
mininet/examples/miniedit.py  
mininet/examples/tree1024.py  
mininet/examples/controllers.py  
mininet/examples/bind.py  
mininet/examples/controllers2.py  
mininet/examples/multipoll.py  
mininet/examples/vlanhost.py  
mininet/examples/mobility.py  
mininet/examples/clusterperf.py  
mininet/examples/linearbandwidth.py  
mininet/examples/multitest.py  
mininet/examples/treeping64.py  
mininet/examples/simpleperf.py  
mininet/examples/hwintf.py  
mininet/examples/multiping.py
```

Mininet's logical nodes can be connected into networks. These nodes are sometimes called containers, or more accurately, **network namespaces**. Containers consume sufficiently fewer resources than networks of over a thousand nodes have created, running on a single laptop. A Mininet container is a process (or group of processes) that no longer has access to all the host system's native network interfaces. Containers are then assigned virtual Ethernet interfaces, which are connected to other containers through a virtual switch⁴. Mininet connects a host and a switch using a virtual Ethernet (veth) link. The veth link is analogous to a wire connecting two virtual interfaces, as illustrated below.

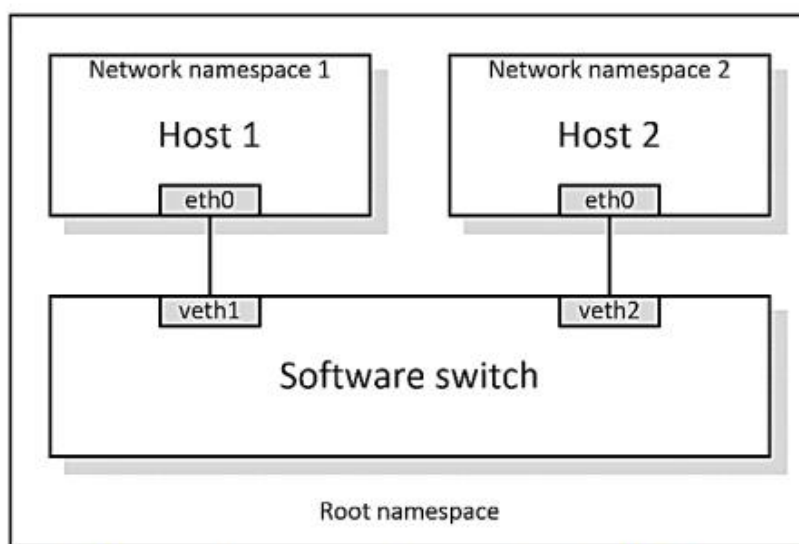


Figure 2. Network namespaces and virtual Ethernet links.

```
hadis@ubuntu: ~  
hadis@ubuntu:~$ sudo mn --topo single,3  
[sudo] password for hadis:  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2 h3  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1) (h3, s1)  
*** Configuring hosts  
h1 h2 h3  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet> █
```

اجرای nodes

```
mininet> nodes  
available nodes are:  
c0 h1 h2 h3 s1  
mininet>
```

خروجی این دستور نشان می دهد که یک کنترلر، دو هاست (host h1 و host h2) و یک سویچ s1 وجود دارد.

Display all the nodes controller, switch and hosts of the topology

اجرای net

```
hadis@ubuntu: ~  
mininet> net  
h1 h1-eth0:s1-eth1  
h2 h2-eth0:s1-eth2  
h3 h3-eth0:s1-eth3  
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0  
c0  
mininet>
```

Show the links connection.

1. میزبان h1 با استفاده از رابط شبکه h1-eth0 به سوئیچ s1-eth1 متصل می شود.
2. میزبان h2 با استفاده از رابط شبکه h2-eth0 به سوئیچ s1-eth2 متصل می شود.
3. سوئیچ s1
a. دارای یک رابط Loopback lo.
b. از طریق رابط s1-eth1 به h1-eth0 متصل می شود.
c. از طریق رابط s1-eth2 به h2-eth0 متصل می شود.
کنترلر c0 مغز شبکه است که در آن دانش جهانی دارد.
یک کنترل کننده به سوئیچ ها دستور می دهد چگونه بسته ها را فوروارد یا رها کنند در شبکه.

اجرای dump

```
mininet> dump  
<Host h1: h1-eth0:10.0.0.1 pid=5427>  
<Host h2: h2-eth0:10.0.0.2 pid=5429>  
<Host h3: h3-eth0:10.0.0.3 pid=5431>  
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=5436>  
<Controller c0: 127.0.0.1:6653 pid=5420>  
mininet>
```

Display all the information about the nodes

هر هاست را به همراه ایپی اختصاص یافته و ایپی پروسس و رابط ان نشان میدهد.
نمایش اطلاعات سوئیچ مثل داشتن یک loopback interface و ایپی پروسس.
ایپی و ایپی پروسس و پورت اجرایی کنترلر را هم نشان میدهد.

سوال ۳ معرفی انواع توپولوژی های دیفالت در mininet

Minimal

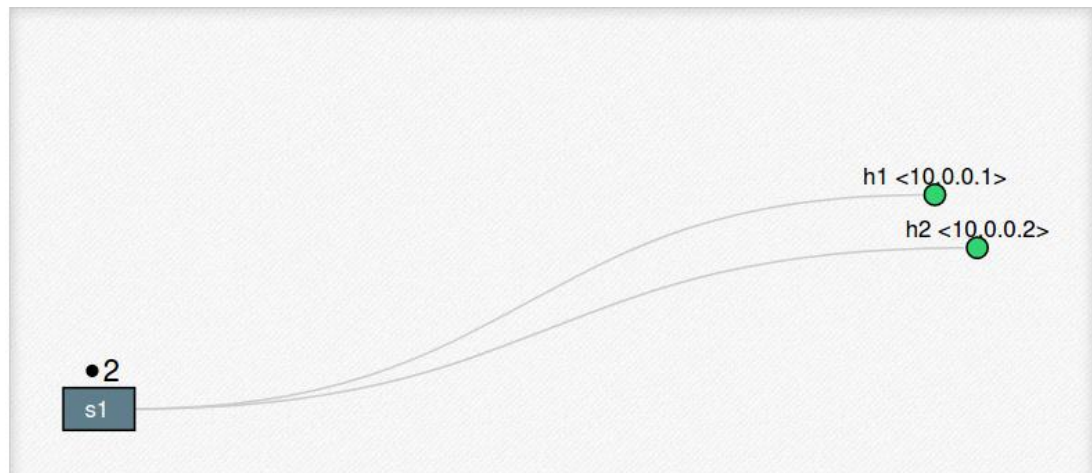
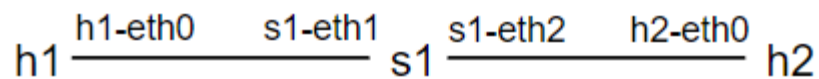
ابتدایی ترین توپولوژی با دو هاست است و یک سوئیچ.

برای اجرای حداقل توپولوژی ما به سادگی دستور زیر را در پنجره ترمینال اجرا می کنیم.

`Sudo mn --topo minimal`

```
hadis@ubuntu: ~  
hadis@ubuntu:~$ sudo mn --topo minimal  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet>
```

```
*** Starting CLI:  
mininet> dump  
<Host h1: h1-eth0:10.0.0.1 pid=7870>  
<Host h2: h2-eth0:10.0.0.2 pid=7872>  
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=7877>  
<Controller c0: 127.0.0.1:6653 pid=7863>  
mininet> links  
h1-eth0<->s1-eth1 (OK OK)  
h2-eth0<->s1-eth2 (OK OK)  
mininet>
```



Reversed Topology

اتصال شبیه حالت single است اما ترتیب اتصال بین هاست و سوئیچ برعکس می شود. برای اجرای توپولوژی معکوس ما از دستور زیر در پنجره ترمینال استفاده می کنیم.

`Sudo mn --topo reversed,3`

```

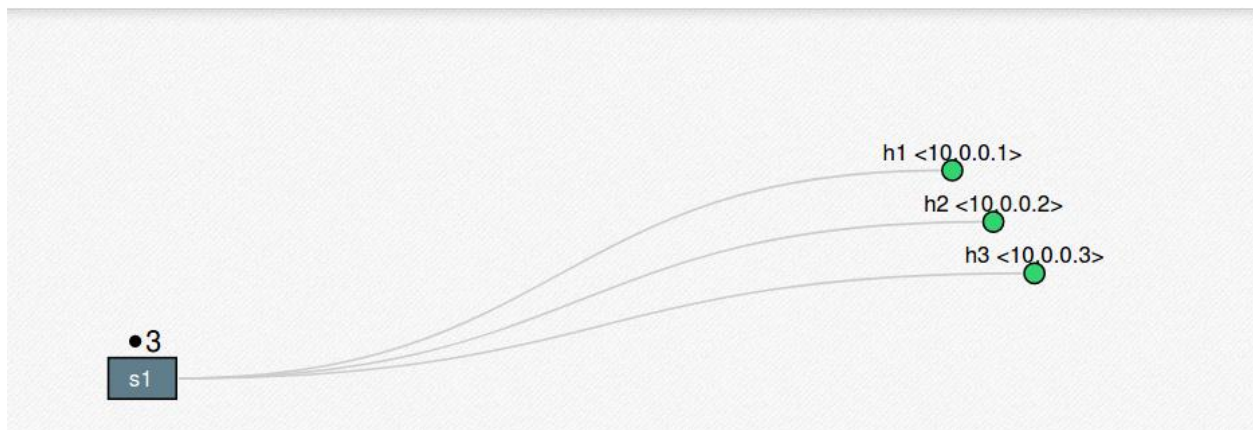
hadis@ubuntu: ~
hadis@ubuntu:~$ sudo mn --topo reversed,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>

```

```

mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=7234>
<Host h2: h2-eth0:10.0.0.2 pid=7236>
<Host h3: h3-eth0:10.0.0.3 pid=7238>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=7243>
<Controller c0: 127.0.0.1:6653 pid=7227>
mininet> links
h1-eth0<->s1-eth3 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s1-eth1 (OK OK)
mininet>

```



Linear Topology

این اتصال بین N میزبان و N سوئیچ است. با نوشتن دستور زیر می توانیم این توپولوژی را اجرا کنیم.

`Sudo mn --topo linear,3`

```

hadis@ubuntu:~$ sudo mn --topo linear,3
*** 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>

```

```

s1-s2-s3-...
*** Starting CLI:
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
h3-eth0<->s3-eth1 (OK OK)
s2-eth2<->s1-eth2 (OK OK)
s3-eth2<->s2-eth3 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=8020>
<Host h2: h2-eth0:10.0.0.2 pid=8022>
<Host h3: h3-eth0:10.0.0.3 pid=8024>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=8029>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=8032>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=8035>
<Controller c0: 127.0.0.1:6653 pid=8013>
mininet>

```

```

h1      h2      h3      hN
|        |        |        |
s1-----s2-----s3--...--sN

```

