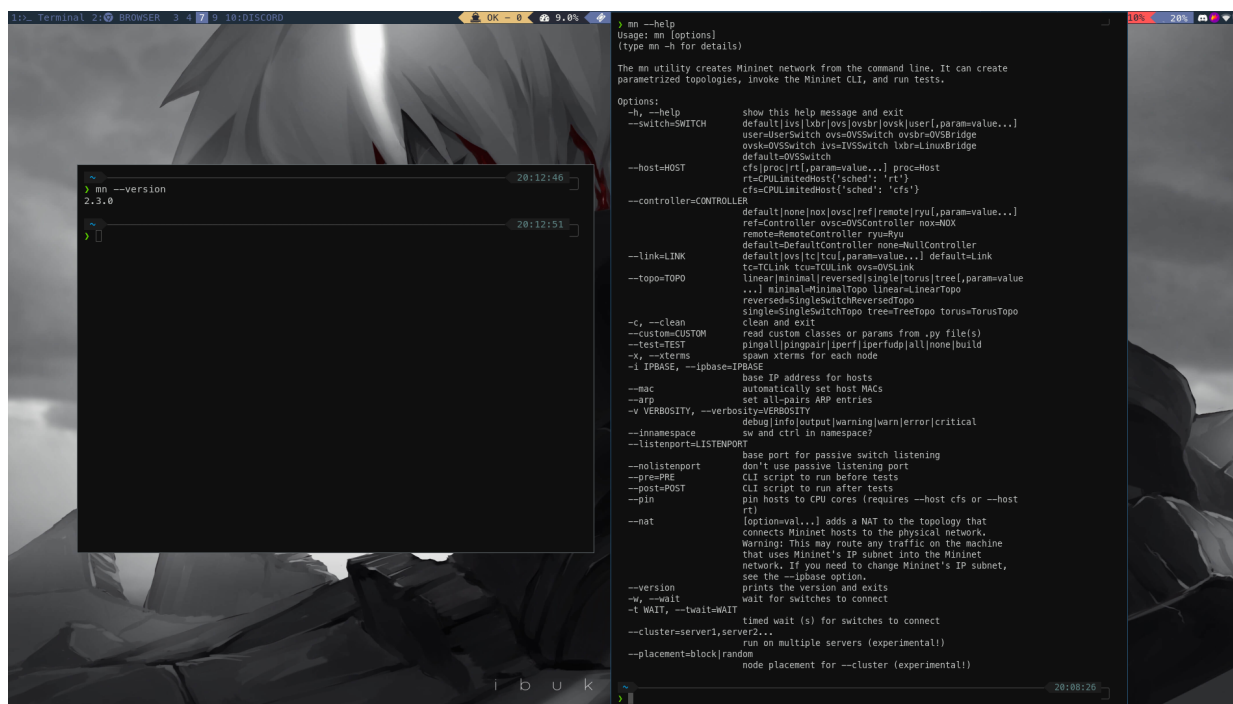


باتوجه به خروجی زیر قسمت اول پروژه یعنی نصب مینی نت به درستی صورت گرفته شده است.



```

> mn --version
2.3.0
> 
> mn --help
Usage: mn [options]
(type mn -h for details)

The mn utility creates Mininet network from the command line. It can create
parametrized topologies, invoke the Mininet CLI, and run tests.

Options:
  -h, --help                show this help message and exit
  --switch=SWITCH           default[linux|vbr|ovs|ovsbr|ovsX[user[,param=value...]]
                             user=usersSwitch ovs=OVSSwitch ovsbr=OVSBridge
                             ovsX=OVSSwitch ixs=IXSvSwitch lxb=LinuxBridge
                             default=OVSSwitch
  --host=HOST               cfs[proc|rt[,param=value...]] proc=Host
                             rt=CPUimitedHost('sched': 'rt')
                             cfs=CPUimitedHost('sched': 'cfs')
  --controller=CONTROLLER  default[none|nox|ovsc|ref|remote|ryu[,param=value...]]
                             ref=Controller ovs=OVSController nox=NOX
                             remote=RemoteController ryu=Ryu
                             default=DefaultController none=NullController
  --link=LINK               default[ovs|tc|tcu[,param=value...]] default=Link
                             tc=Link tcu=CULink ovs=OVSLink
                             linear=Minimal reversed=Single|torus|tree[,param=value
                             ...] minimal=MinimalTopo linear=LinearTopo
                             reversed=SingleSwitchReversedTopo
                             single=SingleSwitchTopo tree=TreeTopo torus=TorusTopo
  -c, --clean              clean and exit
  --custom=CUSTOM          read custom classes or params from .py file(s)
  --test=TEST              pingall|pingpair|iperf|iperfudp|all|none|build
                             spawn xterms for each node
  -x, --xterms             spawn xterms for each node
  -I IPBASE, --ipbase=IPBASE base IP address for hosts
                             automatically set host MACs
  --mac                    set all-pairs ARP entries
  --arp                    debug|info|output|warning|warn|error|critical
  -v VERBOSITY, --verbosity=VERBOSITY sw and ctrl in namespace?
  --namespace              base port for passive switch listening
  --listenport=LISTENPORT don't use passive listening port
  --no_listenport          CLI script to run before tests
  --pre=PRE               CLI script to run after tests
  --post=POST             pin hosts to CPU cores (requires --host cfs or --host
                             rt)
  --pin                   [optionval...] adds a NAT to the topology that
                             connects Mininet hosts to the physical network.
                             Warning: This may route any traffic on the machine
                             that uses Mininet's IP subnet into the Mininet
                             network. If you need to change Mininet's IP subnet,
                             see the --ipbase option.
  --nat                   prints the version and exits
  --version               wait for switches to connect
  -w, --wait              timed wait (s) for switches to connect
  -t WAIT, --twait=WAIT   run on multiple servers (experimental!)
  --cluster=server1,server2... node placement for --cluster (experimental!)
  --placement=block|random
  
```

در شکل بالا خروجی دو دستور `mn --help` , `mn --version` را مشاهده میکنیم.

نصب آن طبق دستورهای زیر انجام شده است.

```

sudo apt install git
sudo git clone https://github.com/mininet/mininet
cd mininet/util
./install -a
  
```

2:

با استفاده از دستور `sudo mn --topo single,3` از توپولوژی آماده‌ی `single` مینی‌نت با سه میزبان، این شبکه را ایجاد کردیم.

```
~ 20:12:46
> mn --version
2.3.0

~ 20:12:51
> sudo mn --topo single,3
[sudo] password for arya:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
ovs-vsctl: unix:/var/run/openvswitch/db.sock: database connection failed (No such file or directory)
ovs-vsctl exited with code 1
*** Error connecting to ovs-db with ovs-vsctl
Make sure that Open vSwitch is installed, that ovsdb-server is running, and that "ovs-vsctl show" works correctly.
You may wish to try "service openvswitch-switch start".

~ 3s < 20:20:58
> sudo service openvswitch start
Redirecting to /bin/systemctl start openvswitch.service

~ 20:21:16
> sudo mn --topo single,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>
```

```

~ 20:21:16
> sudo mn --topo single,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> nodes
available nodes are:
c0 h1 h2 h3 s1
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> dump
<Host h1: h1-eth0:10.0.0.1 pid=2492926>
<Host h2: h2-eth0:10.0.0.2 pid=2492932>
<Host h3: h3-eth0:10.0.0.3 pid=2492938>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=2492962>
<Controller c0: 127.0.0.1:6653 pid=2492911>
mininet>

```

دستور `nodes` برای نشان دادن تمام نودها در شبکه استفاده می‌شود؛ بدین صورت که تمامی نودها اعم از هاست‌ها، سوئیچ‌ها و کنترلر را نشان می‌دهد. باتوجه به خروجی نشان داده شده که با **قرمز** دور آن خط کشیده شده است، شبکه‌ای ایجاد شده دارای سه هاست یک کنترلر و یک سوئیچ می‌باشد.

دستور `net` برای نشان دادن روابط میان نودهاست. (هر نود به چه نودی و از طریق چه چیزی متصل است). برای مثال در شبکه‌ای فوق سه هاست از طریق اینترفیس `eth0` به سوئیچ متصل شده اند و سوئیچ نیز توسط `eth1`, `eth2`, `eth3` به ترتیب به هاست یک تا سه متصل شده؛ در نتیجه سه هاست از طریق این سوئیچ به هم متصل‌اند.

دستور dump نیز برای نمایش نام نود، network interface به همراه آدرس ip آن به علاوه‌ی پروسس آیدی آن در سیستم را نشان می‌دهد.

:3

با توجه به help مینی‌نت طبق عکس زیر توپولوژی‌ها مشخص شده در تصویر را می‌توان استفاده کرد.

```
~ 20:56:58
> sudo mn --help
Usage: mn [options]
(type mn -h for details)

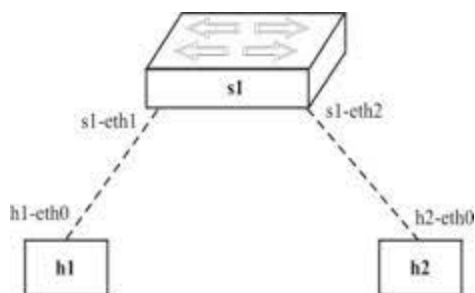
The mn utility creates Mininet network from the command line. It can create
parametrized topologies, invoke the Mininet CLI, and run tests.

Options:
  -h, --help                show this help message and exit
  --switch=SWITCH            default|ivs|lxb|ovs|ovsbr|ovsk[user[,param=value...]]
                             user=UserSwitch ovs=OVSSwitch ovsbr=OVSBridge
                             ovsk=OVSSwitch ivs=IVSSwitch lxb=LinuxBridge
                             default=OVSSwitch
  --host=HOST                cfs|proc|rt[,param=value...] proc=Host
                             rt=CPULimitedHost{'sched': 'rt'}
                             cfs=CPULimitedHost{'sched': 'cfs'}
  --controller=CONTROLLER   default|none|nox|ovsc|ref|remote|ryu[,param=value...]
                             ref=Controller ovsc=OVSController nox=NOX
                             remote=RemoteController ryu=Ryu
                             default=DefaultController none=NullController
  --link=LINK                default|ovs|tc|tcu[,param=value...] default=Link
                             tc=TCLink tcu=TCULink ovs=OVSLink
  --topo=TOPO                linear|minimal|reversed|single|torus|tree[,param=value
                             ...] minimal=MinimalTopo linear=LinearTopo
                             reversed=SingleSwitchReversedTopo
                             single=SingleSwitchTopo tree=TreeTopo torus=TorusTopo
  -c, --clean                clean and exit
  --custom=CUSTOM            read custom classes or params from .py file(s)
  --test=TEST                pingall|pingpair|iperf|iperfudp|all|none|build
  -x, --xterms               spawn xterms for each node
  -i IPBASE, --ipbase=IPBASE base IP address for hosts
  --mac                      automatically set host MACs
  --arp                      set all-pairs ARP entries
  -v VERBOSITY, --verbosity=VERBOSITY debug|info|output|warning|warn|error|critical
  --innamespace              sw and ctrl in namespace?
  --listenport=LISTENPORT    base port for passive switch listening
  --nolistenport             don't use passive listening port
  --pre=PRE                  CLI script to run before tests
  --post=POST                CLI script to run after tests
  --pin                      pin hosts to CPU cores (requires --host cfs or --host
                             rt)
  --nat                      [option=val...] adds a NAT to the topology that
                             connects Mininet hosts to the physical network.
                             Warning: This may route any traffic on the machine
                             that uses Mininet's IP subnet into the Mininet
                             network. If you need to change Mininet's IP subnet,
                             see the --ipbase option.
  --version                  prints the version and exits
  -w, --wait                 wait for switches to connect
```

```

> 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> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet>

```



تصویر رویداد شمایی از شبکه‌ای ایجاد شده است

```

mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0

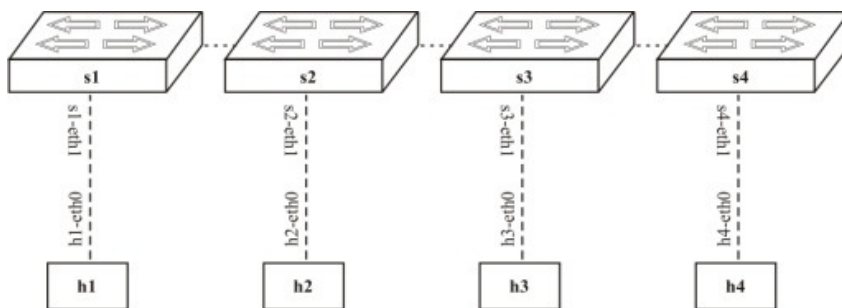
```

توپولوژی **linear**، در این توپولوژی هر هاست به یک سوئیچ متصل می‌شود و سوئیچ‌ها به صورت خطی به یکدیگر متصل می‌شوند. برای مثال مطابق با شکل زیر بنده یک شبکه با ۴ هاست و ۴ سوئیچ طبق این توپولوژی ایجاد کرده‌ام.

```

> sudo mn --topo linear,4
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
h4 h4-eth0:s4-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth3 s3-eth3:s4-eth2
s4 lo: s4-eth1:h4-eth0 s4-eth2:s3-eth3
c0
mininet>

```



```

mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
h4 h4-eth0:s4-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth3 s3-eth3:s4-eth2
s4 lo: s4-eth1:h4-eth0 s4-eth2:s3-eth3
c0

```

شمای توپولوژی ایجاد شده،

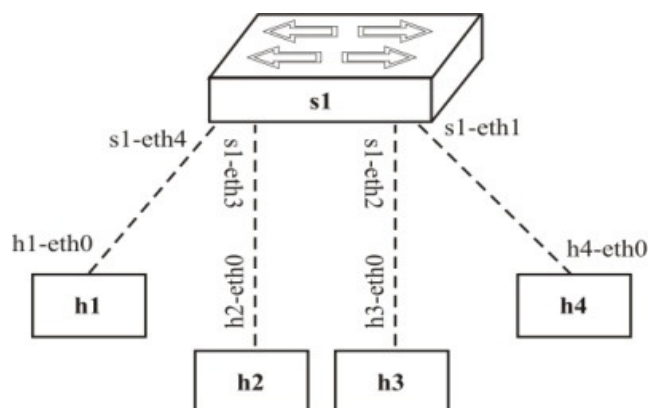
مطابق با شکل روبرو است

توپولوژی reversed، این توپولوژی شبیه به مینیمال است فقط ترتیب اتصال هاست‌ها به پورت‌ها برعکس می‌باشد. مثلاً هاست یک به پورت ۴ متصل است. مطابق با شکل زیر این شبکه را با ۴ هاست ایجاد کرده‌ام.

```

4m 22s < 21:12:15
> sudo mn --topo reversed,4
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth4
h2 h2-eth0:s1-eth3
h3 h3-eth0:s1-eth2
h4 h4-eth0:s1-eth1
s1 lo: s1-eth1:h4-eth0 s1-eth2:h3-eth0 s1-eth3:h2-eth0 s1-eth4:h1-eth0
c0
mininet>

```



شمای توپولوژی ایجاد شده مطابق با شکل روبرو است.

```

mininet> net
h1 h1-eth0:s1-eth4
h2 h2-eth0:s1-eth3
h3 h3-eth0:s1-eth2
h4 h4-eth0:s1-eth1
s1 lo: s1-eth1:h4-eth0 s1-eth2:h3-eth0 s1-eth3:h2-eth0 s1-eth4:h1-eth0
c0

```

باتشکر - یآوری