

به نام خدا

حدیث غفوری 9825413

Mininet یک شبیه ساز شبکه (emulator) است که شبکه ای از host های مجازی، سوئیچ ها، کنترلرها و لینک ها را ایجاد می کند. میزبان های Mininet نرم افزار استاندارد شبکه لینوکس را اجرا می کنند و سوئیچ های آن از OpenFlow برای مسیریابی custom بسیار انعطاف پذیر است و از شبکه های نرم افزاری تعریف شده پشتیبانی می کنند.

Mininet از تحقیق، توسعه، یادگیری، نمونه سازی اولیه، آزمایش، اشکال زدایی و هر کار دیگری که می توان از داشتن یک شبکه آزمایشی کامل بر روی لپ تاپ یا رایانه شخصی دیگر بهره مند شد، پشتیبانی می کند.

Host ها و سوئیچ های ایجاد شده در mininet هرکدام یک اپی میگیرند.

پس میتوانیم در اون هاست ها مثل هر سیستم لینوکس دیگری برنامه هایی که میخاهیم را اجرا کنیم.

مثلا در یک هاست یک سرور اپاچی و در هاست دیگری که به ان متصل است از یک مرورگر مثل فایرفاکس استفاده کنیم تا به ان سرور متصل شویم.

امکانات Mininet

- بستر آزمایشی شبکه ساده و ارزانی را برای توسعه برنامه های OpenFlow فراهم می کند.
- چندین توسعه دهنده همزمان را قادر می سازد تا به طور مستقل بر روی یک توپولوژی کار کنند.
- از تست های رگرسیون در سطح سیستم پشتیبانی می کند که قابل تکرار هستند و به راحتی بسته بندی می شوند.
- تست توپولوژی پیچیده را بدون نیاز به سیم کشی شبکه فیزیکی فعال می کند.
- شامل یک CLI است که از توپولوژی و OpenFlow آگاه است، برای اشکال زدایی یا اجرای آزمایشات در سطح شبکه

- از توپولوژی های سفارشی دلخواه پشتیبانی می کند و شامل مجموعه ای پایه از توپولوژی های پارامتری شده است.
- بدون برنامه نویسی قابل استفاده است.
- همچنین یک API ساده و قابل توسعه Python برای ایجاد و آزمایش شبکه ارائه می دهد.

Mininet یک راه آسان برای به دست آوردن رفتار صحیح سیستم (و تا حدی که توسط سخت افزار شما پشتیبانی می شود، عملکرد) و آزمایش توپولوژی ها ارائه می دهد.

شبکه های Mininet کد واقعی شامل برنامه های شبکه استاندارد یونیکس/لینوکس و همچنین هسته واقعی لینوکس و پشته شبکه را اجرا می کنند.

به همین دلیل، کدی که بر روی Mininet، برای کنترلر OpenFlow، سوئیچ تغییر یافته یا میزبان آزمایش می کنید، می تواند به یک سیستم واقعی با حداقل تغییرات، برای آزمایش در دنیای واقعی، ارزیابی عملکرد و استقرار منتقل شود.

نحوه ی نصب mininet

```
$ sudo apt install git
$ sudo git clone https://github.com/mininet/mininet
$ cd mininet/util
$ ./install.sh -a
$ sudo mn --test pingall
```

باید از ubuntu 16.04 و از جاوا ۸ استفاده کنیم چون در ورژن های بالاتر یک سری از کتابخانه ها حذف شده اند.

در این شبیه ساز هم میتوان از توپولوژی های دیفالت (مثل single) استفاده کرد هم میتوان توپولوژی دلخواه خود را ایجاد کرد.

```
hadis@ubuntu: ~  
hadis@ubuntu:~$ sudo mn --test pingall  
*** 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 ...  
*** Waiting for switches to connect  
s1  
*** Ping: testing ping reachability  
h1 -> h2  
h2 -> h1  
*** Results: 0% dropped (2/2 received)  
*** Stopping 1 controllers  
c0  
*** Stopping 2 links  
..  
*** Stopping 1 switches  
s1  
*** Stopping 2 hosts  
h1 h2  
*** Done  
completed in 5.794 seconds  
hadis@ubuntu:~$ S
```

```

hadis@ubuntu:~$ 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>

```

از هاست ها به یکدیگر ارتباط برقرار است.

```

mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.13 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.501 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.070 ms
^C
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9177ms
rtt min/avg/max/mdev = 0.062/0.423/3.130/0.911 ms
mininet>

```

هر کدام از هاست های ایجاد شده ایپی جداگانه دارند به شکل زیر:

```
"Node: h1"
root@ubuntu:~# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr 16:e6:43:8a:a7:02
         inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
         inet6 addr: fe80::14e6:43ff:fe8a:a702/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:62 errors:0 dropped:0 overruns:0 frame:0
         TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:6001 (6.0 KB)  TX bytes:2000 (2.0 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@ubuntu:~#
```

از یک هاست میتوان با هاست دیگر ارتباط ایجاد کرد.

```
root@ubuntu:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.979 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.345 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.079 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3031ms
rtt min/avg/max/mdev = 0.079/0.628/1.109/0.428 ms
root@ubuntu:~#
```

با دستور `sudo mn --custom` هم میتوان از توپولوژی دلخواه خود استفاده کرد. (به کمک برنامه ای که با کتابخانه های پایتون نوشته ایم)