

۱۴ راهاندازی محیط

در فصول گذشته، سوچیج OpenFlow و کنترلرهای آن را معرفی کردیم. حال در این فصل قصد داریم راه اندازی زیرساخت مورد نیاز برای استقرار Net App در شبکه مبتنی بر OpenFlow را بطور کامل شرح خواهیم داد. در فصل ۲، ما با آزمایشگاه OpenFlow مبتنی بر Mininet و کنترل راه دور POX شروع کردیم و سپس به معرفی پروژه OpenDaylight و کنترلر ODL (که یک پروژه از نوع Bootstrap بود) مربوط به آن پروژه، به عنوان یک پلتفرم کنترل SDN پرداختیم. این مباحث می‌تواند برای استقرار و توسعه Net App، در این فصل مورد استفاده قرار گیرد.

شناخت و درک آزمایشگاه OpenFlow

در فصل ۲، پلتفرم Mininet را (که ابزار مدل کننده شبکه است) به عنوان یک آزمایشگاه برای شناخت پروتکل OpenFlow معرفی کردیم. در این فصل به معرفی این آزمایشگاه با جزئیات بیشتر می‌پردازیم، زیرا قرار است بخشی از محیطی باشد که ما سناریوهای مبتنی بر SDN را در آن مستقر می‌کنیم. Mininet، محیط مجازی سبکی را در کرنل لینوکس بکار می‌برد و سیستمی را ارائه می‌دهد که همانند یک شبکه کامل با کلیه عناصر لازم، به نظر برسد. میزبان‌های درون Mininet درست شبیه یک ماشین (کامپیوتر) رفتار می‌کنند؛ یعنی می‌توان با ایجاد یک ارتباط SSH، به این ماشینها متصل شد (با این شرط که برنامه کمکی SSH را راهاندازی کنید و شبکه Mininet را به کامپیوتر خودتان متصل کنید) و سپس برنامه‌های دلخواه خود را اجرا کرد. هر ابزاری که در لینوکس اجرا می‌شود را می‌توانید در محیط Mininet هم اجرا کنید؛ از وب سرورها گرفته تا نرم افزار تحلیلگر Wireshark یا حتی نرم افزار iPerf با این حال، ساختار Mininet بگونه‌ای است که تنها از یک کرنل برای تمام میزبانهای مجازی درون خود

استفاده می‌کند. این بدان معناست که نمی‌توانید نرم افزاری را که متکی بر^۱ BSD، ویندوز یا دیگر سیستم عاملها باشد بکار ببرید. علاوه بر این، میزبانهای مجازی درون Mininet از File System و PID^۲ اشتراکی با سیستم عامل Mininet استفاده می‌کنند. به عبارتی، باید مراقب باشید که آیا از برنامه‌های کمکی نیازمند به پیکربندی در مسیر `/etc`^۳ استفاده می‌کنید یا خیر. همچنین باید مراقب باشید به اشتباه، پردازشها را حذف نکنید، زیرا هرگونه تغییر در مسیر `/etc`^۴ یا حذف پردازشها از این لیست، ممکن است منجر به قطعی سرویس در Mininet شود. از طرفی دیگر، Mininet ویژگیهای را دارد که از آن جمله می‌توان به عدم پشتیبانی از^۵ NAT، بطور پیش فرض نام برد؛ یعنی میزبانهای مجازی درون Mininet، بصورت پیش فرض از محدوده آدرس IP مجازی استفاده کرده و از شبکه LAN شما منفک می‌شوند. معمولاً این ویژگی مطلوبی به حساب آمده که معنی آن این است که میزبانهای مجازی شما، به شبکه بیرونی دسترسی ندارند و با خیالی آسوده، بدون آنکه نگران اختلال در شبکه باشید (از جمله تداخل آدرس IP)، می‌توانید آزمایشگاه خود را در شبکه محلی بروپا کنید.

مشخصه‌های ویژه ای را که بصورت پیش فرض در سیستم عامل لینوکس وجود دارد، مورد استفاده قرار می‌دهد. این مشخصه‌ها به یک سیستم عامل لینوکس اجازه می‌دهد به تعدادی Container^۶ کوچکتر تقسیم شود و به هر Container، سهم ثابتی از قدرت پردازش را در ترکیب با یک کد - لینک مجازی (Virtual Link-Code) می‌دهد. این امکان به Mininet این توانایی را می‌دهد که لینکهایی (به طور مثال، اتصالات اترنت) را با ویژگیها و سرعتهای مختلف (به طور مثال، 100 Mbps یا 1 Gbps) برای میزبانهای مجازی خود ایجاد کند. Mininet ویژگیهای مجازی‌سازی سیک، شامل گروههای پردازشی، جداسازی پهنهای باند CPU و فضای - نام شبکه (Network Namespace) را در کرنل لینوکس بکار می‌گیرد و آنها را با زمانبندی‌های لینک و لینکهای مجازی اترنت (Veth) ترکیب می‌کند. تا بدین بخش، در مورد قابلیت و توانایی Mininet سخن گفتیم. حال به سراغ میزبان مجازی رفته و در ابتدا آن را تعریف کرده و سپس به سراغ امکاناتش می‌رویم. یک میزبان مجازی در Mininet، گروهی از پردازشها سطح کاربر است، که به یک فضای - نام شبکه منتقل شده‌اند (یک Container برای حالت

^۱BSD که اختصار کلمه Berkeley Software Distribution است، سیستم عامل مشتق شده از یونیکس به شمار می‌رود که توسط دانشگاه برکلی منتشر شده است. تا به امروز همچنان نسخه‌های توسعه یافته‌ای از این سیستم عامل در دسترس عموم می‌باشد. (متترجم)

^۲Process ID

^۳NAT قابلیتی در شبکه است که برآساس آن یک مسیریاب یا یک فایروال می‌تواند آدرس‌های IP مبدأ با مقصد را تغییر داده و با اصطلاحاً بازنویسی کند. بیشترین کاربرد آن در اتصال شبکه‌های محلی به اینترنت بوده که آدرس‌های IP خصوصی موجود در شبکه محلی را به آدرس IP عمومی (و بالعکس) تبدیل می‌کند. (متترجم)

^۴Linux Container (Linux Container)، که با نام اختصار LXC نامیده می‌شوند، متدهای مجازی‌سازی سطح سیستم عامل است که برای اجرای چندین سیستم لینوکس مجازاً بر روی یک میزبان اجرا می‌شود. محصول مجازی‌سازی Xen را می‌توان یک نمونه از Linux Container لینوکس نام برد. (متترجم)

شبکه). فضای - نامهای شبکه، گروههای پردازش را با مالکیت انحصاری اینترفیسها، پورتها و جدولهای مسیریابی (مانند ARP و IP) فراهم می‌کند. بطور ساده می‌توان اینگونه بیان کرد که هر میزبان مجازی که در Mininet بعنوان یک ماشین ایفای نقش می‌کند، مجموعه‌ای از موارد ذکر شده در بالا را در خود دارد تا بتواند مشابه یک کامپیوتر عمل کرده و ترافیک داده را انتقال دهد. ترخ داده هر لینک اترنت مدل شده در Mininet، توسط ^۵ TC (که در داخل لینوکس است)، اعمال می‌شود. TC آماری از وضعیت زمانبندیهای بسته دارد تا ترافیک داده مربوط به میزبانهای مجازی را به یک نرخ مورد نیاز، شکل دهن (Shaping) کند. به شما اجازه می‌دهد تا پارامترهای لینک شبکه را تنظیم کنید. حتی می‌توان آن را با خط فرمان زیر بصورت اتوماتیک تنظیم کرد:

```
$ sudo mn --link tc,bw=10,delay=10ms
mininet> iperf
...
mininet> h1 ping -c10 h2
```

این فرمان، پهنای باند لینکها را روی 10 Mbps و تأخیر (Delay) را روی 10 ms تنظیم می‌کند. با این مقدار تأخیر، زمان تأخیر چرخشی^۶ (Round Trip Time) احتمالاً در حدود 40 ms خواهد شد، زیرا درخواست ICMP از دو لینک عبور می‌کند (درخواست در ابتدا از مبدأ آغاز شده و به سوئیچ می‌رسد و سپس از سوئیچ به سمت مقصد ارسال می‌شود و به این صورت از دو لینک عبور می‌کند) و پاسخ نیز در برگشت از دو لینک مذکور دوباره عبور می‌کند و در مجموع بدلیل عبور از ۴ لینک، زمان تأخیر چرخشی معادل 40 ms را ایجاد می‌کند.

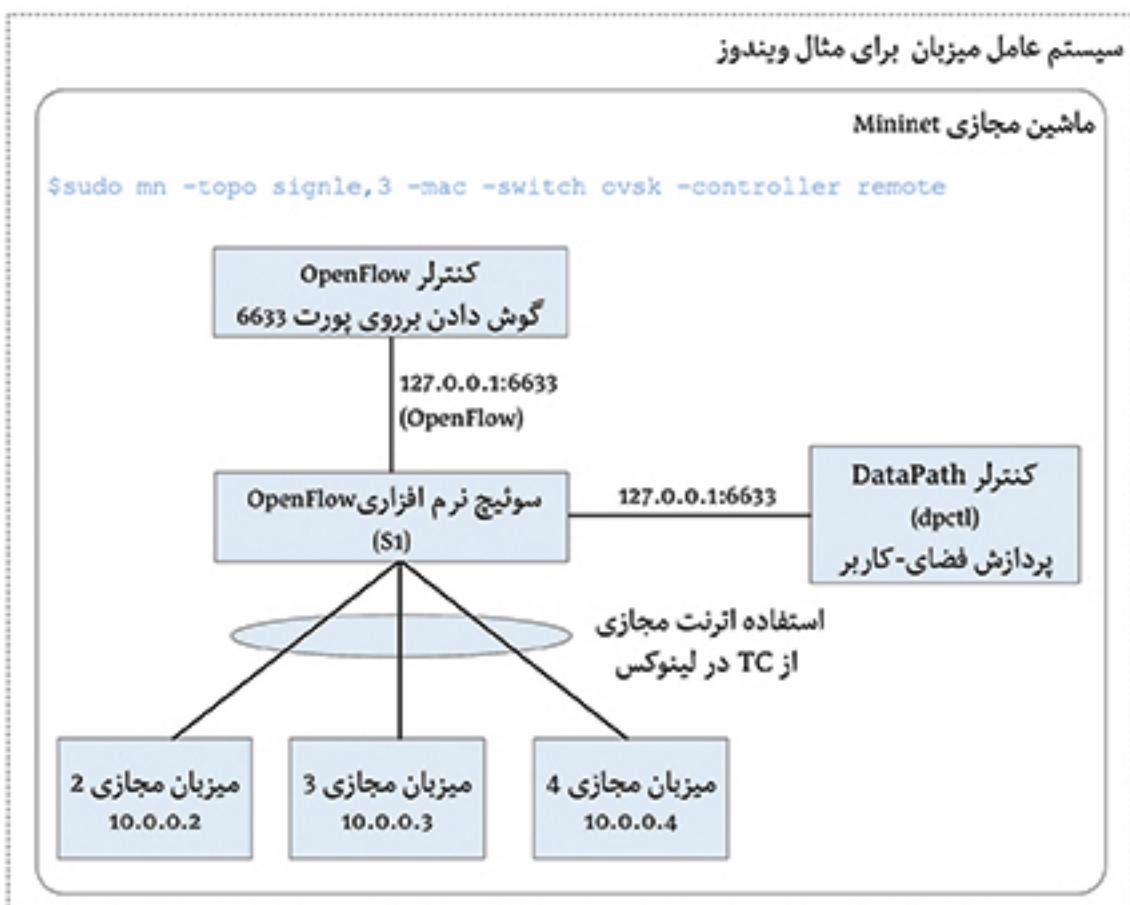
شما می‌توانید با استفاده از Python API که در Mininet وجود دارد، هر کدام از لینکها را سفارشی‌سازی کنید. برای توضیحات بیشتر به آدرس زیر مراجعه کنید:
[!\[\]\(a03a7eb2f4046e1d3c76772003e549ea_img.jpg\) http://github.com/mininet/mininet/wiki/Introduction-to-Mininet](http://github.com/mininet/mininet/wiki/Introduction-to-Mininet)

هر میزبان مجازی، اینترفیس (های) اترنت مجازی خود را دارد. یک جفت اترت مجازی (یا همان Veth) مانند یک سیم عمل کرده که دو اینترفیس مجازی، یا پورتهای سوئیچ مجازی را به هم متصل می‌کند؛ بسته‌هایی که از یک اینترفیس ارسال می‌شوند (این اینترفیس می‌تواند پورت سوئیچ و یا اینترفیس مربوط به یک میزبان مجازی باشد)، به اینترفیس دیگر تحويل داده می‌شود. هر اینترفیس برای تمامی سیستمهای

⁵ Traffic Control

⁶ زمان تأخیر چرخشی، به مدت زمانی گفته می‌شود که یک بسته از مبدأ، به سمت مقصد ارسال شده، مقصد آن را دریافت کرده و به آن پاسخ می‌دهد. پاسخ از سمت مقصد به صورت ACK خواهد بود. (متترجم)

و نرم افزارها، همانند یک پورت اترنت با کلیه قابلیتهای کاربردی، عمل می‌کند. Mininet برای جابجایی بسته‌ها در بین تمام اینترفیسها، بطور معمول از قابلیت پیش فرض Bridge در لینوکس یا سوئیچ مجازی Open vSwitch (که در حالت کرنل اجرا می‌شود) استفاده می‌کند. این وضعیت در شکل زیر نشان داده شده است:



شکل قبل، میزبان‌های مجازی، سوئیچ نرم افزاری و کنترلر OpenFlow را که درون سرور لینوکسی Mininet ایجاد شده‌اند، نمایش داده است. برای ایجاد این توپولوژی شبکه، می‌توانید فرمان زیر را در یک ترمینال SSH که به میزور Mininet متصل شده است، وارد کنید:

\$ sudo mn --topo single,3 --mac --switch ovsk --controller remote

این خط فرمان را می‌توان اینگونه تفسیر کرد که هر میزبان مجازی، آدرس IP جداگانه خود را دارد. میزبانهای مجازی توسط لینک‌های اترنت مجازی به سوئیچ نرم افزاری مجازی، متصل می‌شوند.

آدرس MAC هر میزبان با آدرس IP خودش تنظیم می‌شود. سرانجام سوئیچ نرم افزاری به OpenFlow به یک کنترل راه دور متصل می‌گردد. در ادامه اجزای این فرمان بصورت جزئی تر بررسی می‌شوند.

- بخش `--topo single,3` به Mininet دستور می‌دهد که یک توپولوژی با سه میزبان و یک سوئیچ (سوئیچ فوق، یک سوئیچ Open vSwitch است) راه اندازی کند.
- بخش `--mac` منجر به این تغییر می‌شود که آدرس MAC هر میزبان، متناسب با آدرس IP همان میزبان تنظیم شود. بعنوان مثال اگر آدرس IP میزبان مجازی 10.0.0.1 باشد، آدرس MAC آن نیز 00:00:00:00:00:01 خواهد بود.
- بخش `--controller remote` این فرمان، به Mininet اعلام می‌کند که به یک کنترل راه دور اشاره دارد. در این پیکربندی، کنترل راه دور فوق به آدرس Localhost (که بعنوان پیش فرض انتخاب شده است)، اشاره دارد.

برای مشاهده مثالهایی از چگونگی استفاده از API Python و کدهای بالقوه مفید که در پایگاه کد اصلی Mininet آورده نشده است، می‌توانید به مسیر `~/mininet/examples` در داخل ماشین مجازی Mininet مراجعه کنید.



علاوه بر کامپوننتهای ذکر شده، کامپوننت `dpctl` یک ابزار سودمند است که همراه نسخه مرجع OpenFlow ارائه می‌شود. این ابزار، قابلیت مشاهده و کنترل جدول جریان یک سوئیچ را ارائه می‌دهد. این امکان به ویژه برای اهداف اشکال زدایی و فراهم آوردن قابلیت مشاهده وضعیت جریان و شمارنده‌های جریان یک سوئیچ مفید است. برای دریافت این اطلاعات می‌توانید با استفاده از فرمان `Dump` روی پورت 6634، نمونه برداری از بسته‌ها را روی سوئیچ انجام دهید. با استفاده از فرمان زیر، `dpctl` از طریق یک ارتباط SSH به سوئیچ متصل می‌شود و وضعیت پورت و قابلیتهای آن را نمونه برداری می‌کند:

```
$ dpctl show tcp:127.0.0.1:6634
```

فرمان زیر، جدول جریان سوئیچ نرم افزاری را نمونه برداری می‌کند. در ادامه می‌توانید خروجی فرمان فوق را مشاهده کنید:

```
$ dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0xb5ffalc): flags=none type=1(flow) cookie=0,
duration_sec=1538s, duration_nsec=567000000s, table_id=0,
priority=500, n_packets=0, n_bytes=0,
idle_timeout=0, hard_timeout=0, in_port=1, actions=output:2
```

```
cookie=0, duration_sec=1538s, duration_nsec=567000000s,
table_id=0, priority=500, n_packets=0, n_bytes=0,
idle_timeout=0, hard_timeout=0, in_port=2, actions=output:1
```

همچنین می‌توانید برای نصب دستی جریانهای ضروری در سوئیچ و قرار دادن آنها در جدول جریان سوئیچ، از ابزار `dpctl` استفاده کنید. برای مثال:

```
$ dpctl add-flow tcp:127.0.0.1:6634 in_port=1,actions=output:2
$ dpctl add-flow tcp:127.0.0.1:6634 in_port=2,actions=output:1
```

فرمان بالا بسته‌هایی را که از پورت ۱ سوئیچ وارد می‌شوند، به پورت ۲ ارسال می‌کند و بالعکس. این عمل را می‌توان با استفاده از فرمان زیر و با نموده برداری از جدول جریان بررسی کرد:

```
$ dpctl dump-flows tcp:127.0.0.1:6634
```

بصورت پیش فرض، Open vSwitch می‌تواند OpenFlow-enabled Mininet را در حالت `Open vSwitch` اجرا می‌کند. در این وضعیت سوئیچ OVS به یک کنترلر OpenFlow نیاز دارد. Mininet با استفاده از کلاس‌هایی با نام `Controller()` که از پیش در خود گنجانده است، چندین کنترلر را پشتیبانی می‌کند، از جمله کنترلر مرجع OpenFlow، کنترلر OVS متعلق به سوئیچ Open vSwitch و کنترلر NOX کلاسیک، که این کنترلر امروزه دیگر مورد استفاده قرار نمی‌گیرد (در فصل ۲ در این خصوص صحبت شد). هنگامی که فرمان `mn` را مورد استفاده قرار می‌دهید، می‌توانید انتخاب کنید که کدام کنترلر OpenFlow را می‌خواهید:

```
$ sudo mn --controller ref
$ sudo mn --controller ovsc
$ sudo mn --controller NOX,pyswitch
```

هر کدام از فرمانهای بالا، کنترلری را مورد استفاده قرار می‌دهد که می‌تواند سوئیچ OVS شما را به یک سوئیچ یادگیرنده اترنت تبدیل کنند.

کنترلر OVSC نصب آسانی دارد اما تنها از ۱۶ سوئیچ پشتیبانی می‌کند. شما می‌توانید کنترلر اصلی را با استفاده از فرمان `install.sh -f` نصب کنید. همچنین می‌توانید NOX کلاسیک را با استفاده از فرمان `install.sh -x` نصب کنید؛ ولی به یاد داشته باشید NOX کلاسیک منسخ شده است و ممکن است در آینده از آن پشتیبانی نشود.



کنترل‌های بیرونی

هنگامیکه یک شبکه را با استفاده از Mininet راه اندازی می‌کنید، هر سوئیچی در آن شبکه می‌تواند به یک کنترل راه دور وصل شود. این کنترلر می‌تواند در ماشین مجازی Mininet، خارج از ماشین مجازی Mininet، در کامپیوتر شما یا در هر جایی از اینترنت باشد. این تنظیم در صورتی می‌تواند مفید باشد که شما در گذشته یک چهارچوب و ابزار توسعه برای کنترلر فوق را، بر روی کامپیوتر خودتان نصب کرده باشید یا بخواهید یک کنترلر را که بر روی یک دستگاه فیزیکی نصب شده و در حال اجرا است را آزمایش کنید. اگر قصد انجام این عمل را داشته باشید، باید مطمئن شوید که کنترلر شما از ماشین مجازی Mininet قابل دسترس است. بعبارت ساده‌تر، ماشین مجازی Mininet باید بتواند بر روی بستر شبکه به کنترلر مورد نظر دسترسی داشته باشد. فرمان زیر ارتباط بین سوئیچ OpenFlow (که در ماشین مجازی Mininet است) را با یک کنترلر بیرونی برقرار می‌کند. در نظر داشته باشید که در قسمت‌های [controller listening port] و [controller IP] بایستی به ترتیب، آدرس IP کنترلر بیرونی و همچنین پورت TCP را که سوئیچ بایستی بر روی آن به کنترلر متصل شود، مشخص کنید.

```
$ sudo mn --controller=remote,ip=[controller IP],port=[controller listening port]
```

برای نمونه، برای اجرای سوئیچ یادگیرنده آزمایشی POX، می‌توانید مطابق زیر عمل کنید:

```
$ cd ~/pox
$ ./pox.py forwarding.l2_learning
```

با استفاده از نرم افزار SSH، دو ارتباط PuTTY جهت اتصال به Mininet ایجاد کرده و سپس آن را به کنترل راه دور متصل کنید. (در واقع فرمان زیر اتصال به کنترلر را بصورت محلی اجرا می‌کند، اما بیرون از محدوده کنترل (Mininet

```
$ sudo mn --controller=remote,ip=127.0.0.1,port=6633
```

به یاد داشته باشید این آدرس‌های IP و مقادیر پورت‌ها در واقع پیش فرض هستند.

اگر مقداری ترافیک تولید کنید، (به عنوان مثال با استفاده از فرمان

`mininet> h1 ping h2`) باید بتوانید تعدادی خروجی در پنجره مربوط به کنترلر

مشاهده کنید که نشان دهنده آن است که سوئیچ متصل شده و چند ورودی جدول

در جدول جریان نصب شده است.



تکمیل آزمایشگاه OpenFlow

تا به حال فرا گرفتیم، آزمایشگاه OpenFlow که با استفاده از Mininet راه اندازی شده، از چهار جزء کلیدی تشکیل شده است:

- یک نرم افزار مجازی ساز برای میزبانی ماشین مجازی Mininet. برای مثال VirtualBox یا VMware Player.
- یک برنامه ترمینال که از ارتباط SSH پشتیبانی کند. (مانند PuTTY یا SecureCRT)
- یک X Server برای استفاده از قابلیت X11 Forwarding. عنوان مثال Xming یا XQuartz
- (همانطور که در فصل ۲ هم در مورد این ویژگی صحبت شد، X11 این توانایی را می‌دهد تا از قابلیتهای گرافیکی موجود در لینوکس و نرم افزارهای داخل آن استفاده کنیم)
- فایل Image ماشین مجازی Mininet نسخه 2.0

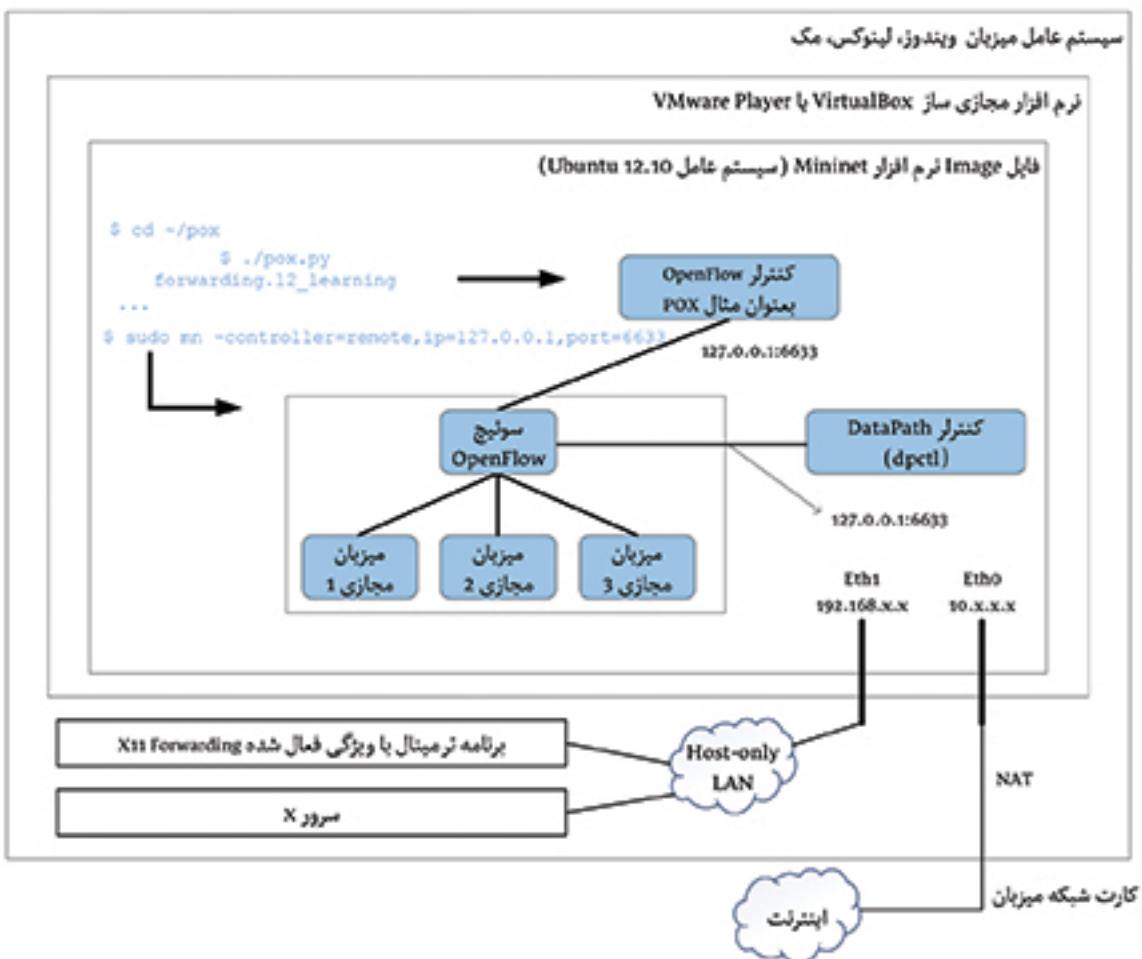
شکل بعد، اجزای سازنده آزمایشگاه OpenFlow و نحوه راه اندازی آن را به طور کامل نشان می‌دهد. این شکل می‌تواند برای راه اندازی و گسترش Net App مفید واقع شود. در Mininet چندین چهارچوب کنترلر OpenFlow (و SDN) به راحتی در دسترس بوده و درست از زمانی که تنظیمات لازم برای اتصال به سویچ (داخل Mininet) را انجام می‌دهیم باید پسادگی با کار کنند (اگر به باد داشته باشید فرمان

```
$ sudo mn -ccontroller=remote, ip=[controller listening port]
```

این اتصال را برقرار می‌کند). در صورتی که با نرم افزار VirtualBox بستر مجازی ساز خود را پیاده سازی کرده اید، باید مطمئن شوید که ماشین مجازی Mininet شما دارای دو اینترفیس شبکه Veth است. یکی از آنها باید اینترفیسی بوده که در حالت NAT قرار داشته باشد. این اینترفیس می‌تواند برای دسترسی به اینترنت و یا شبکه محلی مورد استفاده قرار بگیرد. اینترفیس دوم باید در حالت Host-only بوده تا برای ارتباط با ماشین میزبان (منظور کامپیوتر یا لپ تاپ شخصی) بکار رود. به طور مثال، اینترفیس eth0 در حالت NAT تنظیم شده و یک آدرس IP در رنج 10.x.x.10 داشته باشد و نیز اینترفیس eth1 در حالت Host-only تنظیم شده و یک آدرس IP در رنج 192.168.x.192 داشته باشد. (البته این نکته را در نظر داشته باشید که آدرسهای فوق اختیاری هستند و شما بر حسب نیاز خود می‌توانید آنها را تغییر دهید).

۴-۲ شبکه مجازی با استفاده از Mininet

محیط آزمایشگاه OpenFlow و اجرای سازنده آن



در VirtualBox شما باید اینترفیس دوم شبکه را در حالت Host-only قرار دهید. برای اینکار ابتدا لازم است فایل Image ماشین مجازی خود را انتخاب کرده و سپس به برنامه رفته و نهایتاً به **Network Adapter 2** بروید. گزینه **Enable Network Adapter** را انتخاب و آن را در وضعیت **host-only network** قرار دهید. این تنظیم به شما اجازه می‌دهد از طریق کامپیوتر خود به آسانی به ماشین مجازی دسترسی داشته باشید.



اکنون شما باید برسی کنید که آیا کامپیوتر یا لپ تاپ شخصی خود، از طریق ارتباط SSH به ماشین مجازی (آزمایشگاه OpenFlow) دسترسی دارد یا خیر. در نرم افزار مجازی ساز خود (در اینجا VMware Player) روی ماشین مجازی Mininet کلیک کنید تا بتوانید صفحه Mininet را مشاهده کنید. سپس نام کاربری و رمز عبور را وارد کنید. بصورت پیش فرض نام کاربری [mininet](#) و رمز عبور [mininet](#) است. پس از آن فرمان زیر را وارد کنید:

```
$ ifconfig -a
```

شما باید سه اینترفیس `lo`, `eth0` و `eth1` مشاهده کنید. هر دو اینترفیس `eth0` و `eth1` باید آدرس IP معینی داشته باشند. اگر این گونه نباشد این فرمان را وارد کنید:

```
$ sudo dhclient ethX
```

با این فرمان می‌توانید از طریق پروتکل DHCP بصورت اتوماتیک، یک آدرس IP دریافت کنید. فراموش نکنید که `ethX` را با نام یکی از اینترفیسها جایگزین کنید. به آدرس IP مربوط به `eth1` (به یاد داشته باشید که `eth1` در وضعیت Host-only قرار دارد و احتمالاً آدرس IP دریافتی آن در محدوده `192.168.x.x` است). توجه داشته باشید، بعداً به آن نیاز خواهد داشت. سپس با استفاده از نرم افزار Terminal.app (مانند SSH, PuTTY وغیره) وارد ماشین مجازی Mininet خود شوید. عنوان ترمینال اگر می‌خواهید برای اتصال به Mininet از یک کامپیوتر لینوکس استفاده کنید، فرمان زیر را اجرا کنید:

```
$ ssh -X mininet@[eth1's IP address]
```

به منظور استفاده از اپلیکیشن‌های X11 (منظور Wireshark و Xterm است)، سرویس Xserver بایستی در حال اجرا باشد. (برای اطمینان از اجرایی بودن Xterm می‌توانید به منوی کنار ساعت در ویندوز خود رفته و نرم افزار را مشاهده کرده یا روی Task Manager رفته و در بخش Processes، از در حال اجرا بودن آن اطمینان حاصل کنید). قدم بعدی بررسی قابل دسترسی بودن Xserver است. یک پنجره ترمینال، را با استفاده از فرمان Xterm به شرح زیر بکار بیندازید:

```
$ xterm
```

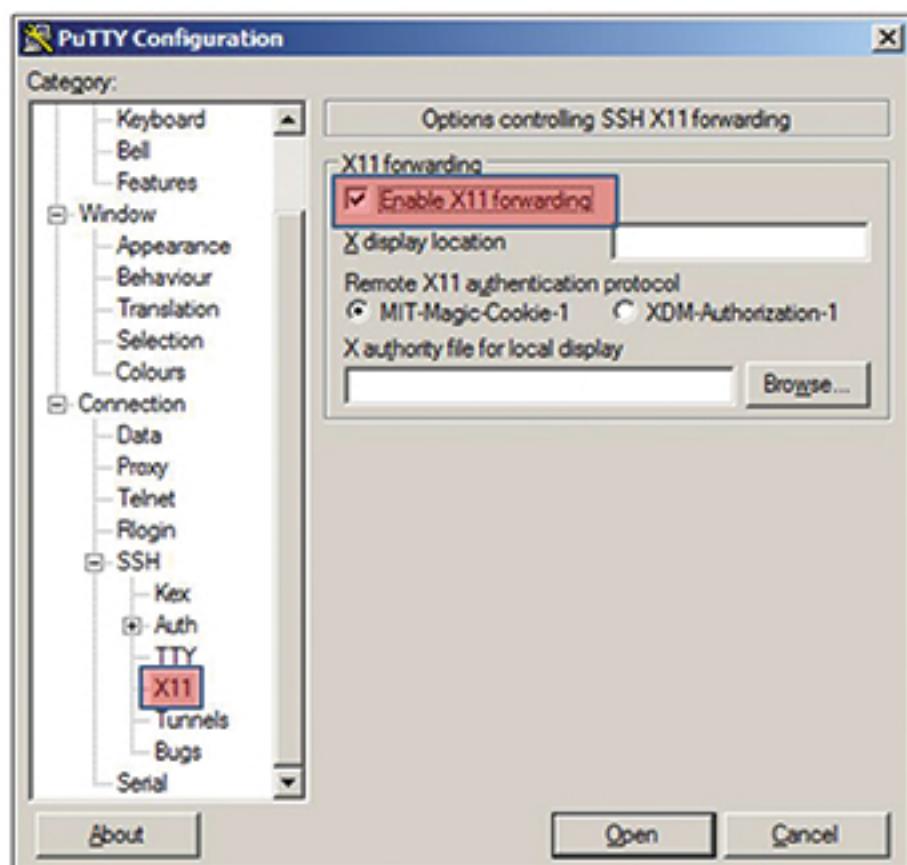
پس از اعمال فرمان بالا و در صورتیکه Xserver به درستی اجرا شود، می‌بایست یک پنجره جدید ترمینال، نمایان شود. در این صورت محیط آزمایشگاه OpenFlow آماده بوده و پس از آزمایش فوق، شما می‌توانید Xterm را مشاهده کنید. حال در صورتی که پیغام خطای `xterm: Xt error: Can't open display` را دریافت کردید (یا خطایی مشابه آن)، مراحل نصب Xserver را مجددآ بررسی کنید.

در ویندوز، سرور Xming باید در حال اجرا باشد و شما باید یک ارتباط SSH را که قابلیت X11 Forwarding در آن فعال شده است، برقرار کنید. ابتدا، سرویس Xming را اجرا کنید. به یاد دارید که Xming پنجره‌ای نشان نمی‌دهد، ولی می‌توانید با بررسی فرایند آن در منوی Taskbar ویندوز از اجرا شدن آن مطمئن شوید. سپس، یک ارتباط SSH را که قابلیت X11 Forwarding در آن فعال شده است،

برقرار کنید. اگر از نرم افزار PuTTY برای ارتباطات SSH خود استفاده می‌کنید، می‌توانید با وارد کردن آدرس IP ماشین مجازی خود (eth1) و فعال کردن X11 Forwarding، به آزمایشگاه OpenFlow (منظور اتصال به سرور Mininet است) متصل شوید. برای فعال کردن X11 Forwarding از طریق محیط نرم افزار PuTTY، به روی گزینه *Enable X11* پس رفته و سپس زیر *PuTTY Connection / SSH / X11* کلیک کنید، (شکل زیر)

شکل ۶-۳

فعال کردن
X11 در
PuTTY



یک راه جایگزین برای استفاده از X11 وجود دارد. شما می‌توانید X11 را در خود ماشین مجازی نصب کنید (که داخل ماشین مجازی آزمایشگاه OpenFlow شما قرار دارد). برای نصب X11 و یک برنامه "مدیریت پنجره ساده" (که برای کنترل پنجره‌های باز شده بر روی صفحه ویندوز مورد استفاده قرار می‌گیرد)، وارد پنجره کنسول ماشین مجازی شوید (نام کاربری: `mininet` و رمز عبور: `mininet`) و فرمانهای زیر را تایپ کنید:

```
$ sudo apt-get update
$ sudo apt-get install xinit flwm
```



اکنون، باید بتوانید یک ارتباط X11 را در پنجره کنسول ماشین مجازی، با فرمان زیر راه اندازی کنید:

```
$ startx
```

پس از برقراری ارتباط SSH با ماشین مجازی آزمایشگاه OpenFlow و وارد شدن در آن، می‌توانید یک شبکه نمونه Mininet را با وارد کردن خط فرمان زیر، راه اندازی کنید:

```
$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
```

فراموش نکنید از آنجایی که هیچ کنترلر OpenFlow را راه اندازی نکرده اید، با چنین پیغامی مواجه خواهید شد:

```
unable to contact the remote controller at 127.0.0.1:6633
```

از آنجا که گزینه X11 *forwarding* نیز فعال است، شما می‌توانید نرم افزار Wireshark را با استفاده از فرمان زیر، بکار بیاندازید تا ترافیک OpenFlow را دریافت کرده و مورد بررسی قرار دهید. فراموش نکنید که اجرایی شدن Wireshark بدون فعال کردن سرویس X11 Forwarding محقق نخواهد شد.

```
mininet@mininet-vm:~$ wireshark &
```

این فرمان محیط گرافیکی Wireshark را به نمایش می‌گذارد و شما می‌توانید (به همان صورت که در فصل ۲ ذکر شده است) ترافیک شبکه را دریافت کرده و از بین آن، ترافیک OpenFlow را فیلتر کنید. حال می‌توانید کنترلر راه دور خود را راه اندازی کنید. این کنترلر در واقع، داخل ماشین مجازی آزمایشگاه OpenFlow شما اجرا می‌شود. بنابراین باید سراغ کنسول ماشین مجازی خود رفته و فرمانهای زیر را وارد کنید:

```
mininet@mininet-vm:~$ cd pox
mininet@mininet-vm:~/pox$ ./pox.py forwarding.l2_learning
```

پس از چند ثانیه، سوئیچ نرم افزاری POX در Mininet OpenFlow متصل می‌شود. خروجی کنترلر POX شما، باید مانند زیر باشد:

```
POX 0.0.0 / Copyright 2011 James McCauley
DEBUG:core:POX 0.0.0 going up...
DEBUG:core:Running on CPython (2.7.3/Sep 26 2012 21:51:14)
INFO:core:POX 0.0.0 is up.
This program comes with ABSOLUTELY NO WARRANTY. This program is
free software, and you are welcome to redistribute it under
certain conditions.
Type 'help(pox.license)' for details.
DEBUG:openflow.of_01:Listening for connections on 0.0.0.0:6633
INFO:openflow.of_01:[Con 1/1] Connected to 00-00-00-00-00-01
DEBUG:forwarding.12_learning:Connection [Con 1/1]
Ready.
POX>
```

پیغامهای اشکال‌زدایی کنترلر POX نشان می‌دهند که سوئیچ OpenFlow که در آزمایشگاه خود راه اندازی کرده‌اید به کنترلر POX متصل شده است و مانند یک سوئیچ یادگیرنده لایه ۲ رفتار می‌کند. بدین ترتیب راه اندازی آزمایشگاه OpenFlow، در گام نخست خاتمه می‌باید. ما موفق به راه اندازی یک شبکه با استفاده از Mininet شدیم؛ همچنین یک کنترلر OpenFlow از راه دور POX را به عنوان محیطی برای گسترش Net App، بکار انداختیم. در فصل ۵، برای ایجاد و گسترش Net App از محیط آزمایشگاهی Mininet، استفاده خواهیم کرد. در ادامه، با راه اندازی کنترلر OpenDaylight، ویژگی‌های دیگری از این بستر را معرفی خواهیم کرد.

OpenDaylight

در صورتیکه در مورد OpenDaylight در صفحه ۶۵ مطالعه داشته‌اید، مطالبی که در ادامه شرح داده شده‌اند را نادیده گرفته و به صفحه بعد، بخش «کنترلر ODL» مراجعه کنید.

کنترلر OpenDaylight پروژه مشترک بنیاد لینوکس بوده که در آن، یک مجموعه گردhem آمدۀ‌اند تا نیاز مربوط به چهارچوبی مرجع و باز برای برنامه نویسی و کنترل شبکه را، بوسیله یک راه حل SDN متن باز برطرف کنند. این پروژه ترکیبی است از توسعه دهنده‌گان جامعه باز، کدهای متن باز و سیستم نظارت پروژه که فرایند تصمیم‌گیری جمیع و آزاد را در موضوعات فنی و تجارت تضمین می‌کند. OpenDaylight می‌تواند یک کامپوننت مرکزی در انواع معماری‌های SDN محسوب شود. استفاده از یک کنترلر متن باز SDN، کاربران را قادر می‌سازد پیچیدگی عملیات را کاهش دهند، عمر زیر ساخت شبکه موجود را طولانی کنند و خدمات و قابلیتهای جدید را تنها با SDN قابل استفاده کنند. بیانیه

کاری پروژه OpenDaylight به شرح زیر است: OpenDaylight به یک چهارچوب متن باز تحت حمایت صنعت و زیر نظر جامعه مهندسان که شامل کد و معماری بوده کمک می کند که این امر باعث سرعت بخشیدن و پیشرفت این پلتفرم SDN قدرتمند و عمومی شده است." OpenDaylight برای همگان آزاد است. همه می توانند در توسعه کدنویسی آن سهیم بوده و در انتخابات کمیته هدایت فنی TSC شرکت کنند و برای هیأت مدیره انتخاب شوند، یا به روشهای گوناگون در امر نظارت بر پیشرفت پروژه کمک کنند. OpenDaylight ترکیب متنوعی از پروژه ها خواهد بود. هر پروژه دارای همکاران، اعضای کمیته و یک عضو منتخب کمیته خواهد بود که توسط ناظران برای رهبری پروژه انتخاب می شود. اولین کمیته هدایت فنی و رهبران پروژه، مشکل از متخصصان خواهد بود که کدهای اصلی پروژه را ایجاد کرده اند. این امر این اطمینان را خواهد داد که مجموعه، مشکل از متخصصان است که بیشترین آشنایی با کدهای ایجاد شده را داشته تا نگرانی در رابطه با پیشرفت و همچنین ارائه مشاوره به اعضای جدید، وجود نداشته باشد. در میان پروژه های جدید در چهارچوب Bootstrap، کنترلر ODL یکی از پروژه های قدیمی است که در فصل بعد، بیشتر در مورد آن صحبت خواهیم کرد و سپس محیط خود را برای توسعه Net App مبتنی بر ODL آماده می کنیم.

کنترلر ODL

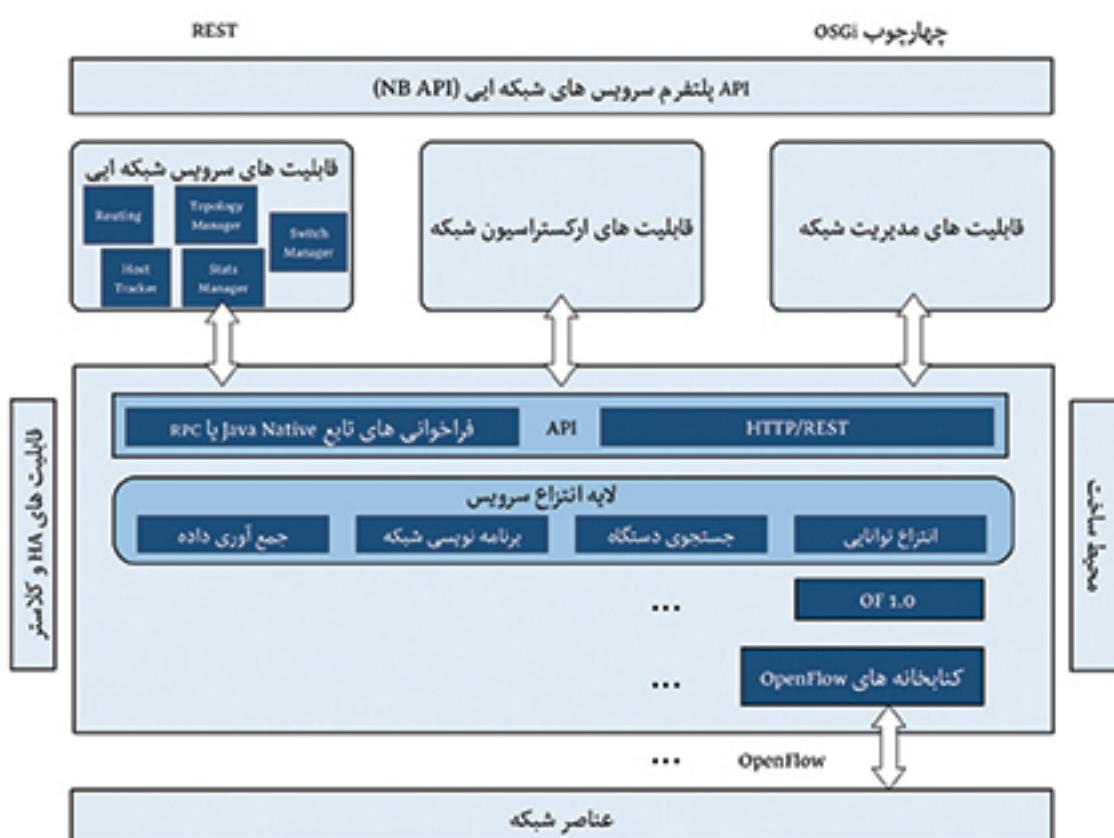
کنترلر ODL زیرساختی با قابلیتهایی همچون دسترسی پذیری بسیار مطلوب، مازولار بودن، توسعه پذیری، مقیاس پذیری و چند - پروتکلی را دارا است. کنترلر ODL به منظور استقرار شبکه ای مبتنی بر SDN روی بستر شبکه های جدیدی ساخته شده است که با طیف گسترده ای از تجهیزات مختلف سازگاری دارد. مدل مبتنی بر ^۷SAL، مفاهیم انتزاعی مورد نیاز برای پشتیبانی پروتکلهای گوناگون روی اینترفیس جنوبی را (به طور مثال پروتکل OpenFlow) از طریق پلاگینها، فراهم می کند. معماری قابل گسترش اینترفیس شمالی (که مبتنی بر اپلیکیشن است)، مجموعه ای غنی از API های شمالی را بوسیله سرویسهای وب RESTfull (برای اپلیکیشنهای با کمترین وابستگی بین کامپوننتی، که به آنها Loosly Coupled App گفته می شود) و سرویسهای OSGi (برای اپلیکیشن های Co-located) فراهم می کند. چهارچوب OSGi که براساس آن پلتفرم کنترلر ساخته می شود، مسئول ماهیت مازولی کنترلر و توسعه پذیری آن است. علاوه بر آن، مدیریت نسخه سازی و چرخه حیات را برای مازولها و خدمات OSGi فراهم می کند. کنترلر OpenDaylight نه تنها از پروتکل OpenFlow، بلکه از دیگر پروتکلهای متن - بازی که برای ارتباط با تجهیزاتی که دارای Agent های OpenFlow هستند، نیز پشتیبانی می کند.

کنترلر OpenDayLight همچنین دارای یک API شمالي است تا به اپليکيشنهای سمت مشتری (نرم افزار) نيز اجازه دهد، که با کنترلر ODL، شبکه را کنترل کند.

کنترلر ODL با استفاده از جاوا توسعه داده می‌شود و به عنوان یک ^۵JVM می‌تواند در هر پلتفرم سخت افزاری و سیستم عاملی که جاوا JVM را ارائه می‌دهد کار کند. کنترلر ODL از JVM نسخه ۱.۷ و نسخه‌های بالاتر پشتيبانی می‌کند. معماری آن در شکل زير نشان داده شده است:

شکل ۶-۶

معماری کنترلر ODL



کنترلر ODL در سمت اينترفيis جنوبی خود می‌تواند پروتوكلهای متنوعی را به عنوان پلاگین پشتيبانی کند (برای مثال، BGP-LS، PCE، OpenFlow 1.0 و غیره). در حال حاضر اين کنترلر، از OpenFlow 1.0 پشتيبانی می‌کند.^۱ دیگر مشارکت کنندگان پروژه OpenDaylight نيز می‌توانند بعنوان همکار یا با هدف اجرای بخشی از پروژه‌های خود، به اين مجموعه اضافه شوند. اين مازولها بطور پویا به يك SAL متصل شده‌اند. SAL سرويسهای را ارائه می‌دهد که مازولهاي لایه بالاتر برای آنها خدماتی

^۵Java Virtual Machine^۱در زمان ترجمه اين کتاب، نسخه ۱.۳ نوسط ODL پشتيبانی می‌شود. (متوجه)

ارائه می‌دهد. SAL بدون توجه به پروتکل اصلی مورد استفاده بین کنترلر و عناصر شبکه (سوئیچ OpenFlow)، تشخیص می‌دهد که چگونه به سرویس‌های درخواست شده پاسخ دهد. این امر هنگامی که OpenFlow و دیگر پروتکلها به مرور تکامل می‌یابند، حاشیه امنی را برای سرمایه گذاران در حوزه اپلیکیشنها فراهم می‌آورد؛ زیرا آسیب‌های احتمالی را که ممکن است اپلیکیشنها به خاطر تغییر ساختار یک پروتکل متحمل شوند، به حداقل می‌رساند. اطلاعات مربوط به قابلیتها و میزان در دسترس بودن تجهیزات شبکه توسط کامپوننت Topology Manager مدیریت و ذخیره می‌شود. کامپوننتهای دیگر (بعنوان مثال Manager Switch و Device Manager، Host Tracker، ARP Handler، Switch Manager API) برای تولید پایگاه داده توپولوژی، به کامپوننت Topology Manager، کمک می‌کنند. جزئیات مربوط به عناصر شبکه را نگه می‌دارد. زمانی که یکی از تجهیزات شبکه توسط این ابزار دیده می‌شود، مشخصه‌های آن دستگاه (مانند، نوع سوئیچ / مسیریاب، نسخه نرم افزاری آن، قابلیتها و غیره) توسط کامپوننت Switch Manager در پایگاه داده، ذخیره می‌شود. کنترلر، API‌های باز شماری را که توسط اپلیکیشنها به کار می‌روند، نمایش می‌دهد. همانطور که پیشتر ذکر شد، کنترلر ODL از چهارچوب OSGi و REST دو طرفه برای API شمالی پشتیبانی می‌کند. چهارچوب OSGi برای اپلیکیشنها بکار می‌رود که در همان فضای آدرسی (منظور رنج آدرس IP) که کنترلر قرار دارد اجرا می‌شوند، در حالی که REST API (مبتنی بر وب) برای اپلیکیشنها بی همچون کنترلر، بکار می‌رود که در فضای آدرس مشابه، پلتفرم سخت افزاری یا حتی پلتفرم نرم افزاری مشابه، اجرا نمی‌شوند. لایه Business Logic و الگوریتم‌های موجود در شبکه‌های مبتنی بر SDN در Net App قراردارند. این Net App‌ها، از کنترلر برای فراهم کردن هوش شبکه، از الگوریتم‌ها برای تحلیل و سپس از کنترلر برای یکپارچه کردن قوانین جدید در سرتاسر شبکه استفاده می‌کنند.

کنترلر ODL، یک مدل مبتنی بر کلاستر با قابلیت دسترسی بالا، برای مدیران شبکه فرآهم می‌کند. چندین نمونه از کنترلر ODL می‌توانند در کنار یکدیگر قرار گرفته و مانند یک کنترلر منطقی واحد عمل می‌کنند. این امر نه تنها افزونگی (Redundancy) با جزئیات فراوانی را فراهم می‌کند، بلکه یک مدل مقیاس افقی^(۱) (Scale Out) را برای مقیاس پذیری خطی، ارائه می‌دهد. کنترلر ODL از یک اینترفیس گرافیکی پشتیبانی می‌کند که بصورت پیش فرض در کنترلر قرار دارد. این اینترفیس گرافیکی تحت

^(۱) Business Logic به بخشی از یک محصول نرم افزاری گفته می‌شود که قوانین حاکم بر کسب و کار یا جریانهای کاری یک سازمان را در قالب کدهای برنامه نویس پیاده‌سازی کرده است. Business Logic توالی بین عملیاتها را می‌تواند در خود قرار داده و آنها را پردازش کند. این لایه را می‌توان بین لایه اینترفیس کاربر و پایگاه داده، یعنوان یک لایه میانی قرار داد. (متترجم)

^(۲) مقیاس پذیری افقی که به آن Scale Out هم گفته می‌شود به وضعیت اطلاق می‌شود که بتوان چندین گره مشابه در کنار یکدیگر ایجاد کرد. در این حالت، بارگذاری روی گره‌های فوق پخش می‌شود. یعنوان مثال فرض کنید که از یک وب سرور، چندین نمونه وجود داشته باشد. این مقیاس پذیری در برایر مقیاس پذیری عمودی با Scale Up قرار دارد که براساس آن، منابع داخلی یک گره (بعنوان مثال میزان CPU، RAM و...) ارتقاء داده می‌شوند. (متترجم)

عنوان اپلیکیشنی پیاده‌سازی شده است که همان API شمالی موجود برای دیگر اپلیکیشن‌های کاربر را، مورد استفاده قرار می‌دهد.

برای اطلاعات بیشتر در خصوص معماری، توسعه زیرساخت، توصیف کتابخانه و مرجعهای API، لطفاً به صفحه Wiki کنترلر ODL مراجعه کنید:

http://wiki.opendaylight.org/view/OpenDaylight_Controller:Programmer_Guide



آزمایشگاه SDN مبتنی بر ODL

در این بخش، آزمایشگاه SDN خود را اینبار با استفاده از کنترلر ODL بروپا می‌کنیم. در این روش، فرض بر این است که شما کنترلر ODL را بر روی سیستم عامل لینوکس نصب می‌کنید و با استفاده از ماشین مجازی Mininet، یک شبکه مجازی ایجاد می‌کنید. سیستم عامل میزبان ما (همان کامپیوتری که VMware را بر روی آن نصب کردید) Windows 7 Enterprise است. بنابراین در سرتاسر این بخش، ما از VMware Player برای میزبانی یک ماشین مجازی دیگر (Ubuntu12.04) که نقش کنترلر ODL را ایفا می‌کند، استفاده می‌کنیم. تنظیمات ماشین مجازی ما بدین قرار است:

- سخت‌افزار مورد نیاز باید از پردازنده دو هسته‌ای، بهمراه 2 GB RAM و 20 GB فضای هارد دیسک تشکیل شده باشد.
- کارت شبکه ماشین مجازی Ubuntu خود را (که قرار است بعنوان کنترلر ODL انجام وظیفه کند)، در حالت Bridge قرار دهید. این وضعیت، ماشین مجازی را بر روی همان شبکه‌ای (همان رنج آدرس IP) قرار می‌دهد که کارت شبکه کامپیوتر خود متصل کنید. بنابراین اگر میزبان می‌توانید به کارت شبکه اینترنت یا بی‌سیم کامپیوتر خود اینترنت کنید، بنابراین اگر میزبان فیزیکی همچون لپ تاپ شما، بر روی آدرس IP برابر 192.168.0.10/24 باشد، یک ماشین مجازی در حالت Bridge، آدرس IP معادل 192.168.0.11/24 را خواهد داشت، یا هر آدرسی که سرور DHCP برای آن تعیین می‌کند. در نهایت می‌توان اینطور بیان کرد که مهمترین نکته در این مرحله، باقی ماندن ماشین مجازی در داخل همان محدوده شبکه‌ای است که کامپیوتر میزبان در آن قرار دارد.

پس از وارد شدن به ماشین مجازی خود، باید موارد زیر را بعنوان پیشنباز دانلود کنید:

- JVM نسخه ۱.۷ یا نسخه‌های بالاتر، به طور مثال OpenJDK ۱.۷ را می‌توانید مورد استفاده قرار دهید. در نظر داشته باشید، متغیر `JAVA_HOME` را در بخش متغیرهای محیطی (Environment Variable) تنظیم کنید.
 - از مخزن کد Git، برای دریافت کنترلر ODL استفاده کنید.
 - Maven را بروی سیستم عامل مورد نظر اجرا کنید.
- ملزومات را نصب کرده و کد را با استفاده از Git دریافت کنید:

```
$ sudo apt-get update
$ sudo apt-get install maven git openjdk-7-jre openjdk-7-jdk
$ git clone http://git.opendaylight.org/gerrit/p/controller.git
$ cd controller/opendaylight/distribution/opendaylight/
$ mvn clean install
$ cd target/distribution.opendaylight-0.1.0-SNAPSHOT-
osgipackage/opendaylight
```

به این ترتیب ابزارهای مورد نیاز را نصب و کنترلر ODL را از مخزن Git دانلود می‌کنیم. سپس با استفاده از Maven، کنترلر ODL را ساخته (Build) و نصب می‌کنیم. Apache Maven یک ابزار اتوماتیک‌سازی برای عملیات ساخت کدهای منبع است که در ابتدا برای پروژه‌های Java مورد استفاده قرار می‌گرفت. در نظر داشته باشید عملیات ساختن کنترلر ODL، چندین دقیقه به طول می‌انجامد.

اگر هنگام عملیات ساختن در Maven با چنین پیغام خطای زیر مواجه شدید:

`Out Of Memory error: PermGen Space` - با استفاده از سوئیچ `X`-بار

دیگر Maven را اجرا کنید تا عملیات دریافت Log در زمان ساختن، فعال گردد. این ایراد به دلیل مشکل نشست حافظه در بخشی از عملیات ساختن است و به عنوان یک

باغ مطرح شده است. بجای فرمان `maven`، فرمان `mvn clean install`

`clean install -DskipTests` را می‌توانید اجرا کنید. فرمان جدید

آزمایش‌های یکپارچه‌سازی را که به نظر می‌رسد منشاء نشست قسمت Garbage

باشد، نادیده می‌گیرد. همچنین می‌توانید این خطا را با تنظیم گزینه‌های Collector

Maven مطابق زیر برطرف کنید:

```
$ export MAVEN_OPTS="-Xmx512m -XX:MaxPermSize=256m"
```



خلاصه ارائه شده در انتهای عملیات ساختن توسط Maven، ایجاد موفقیت آمیز کنترلر ODL را همراه با زمان سپری شده و حافظه تخصیص داده شده به آن، گزارش می‌دهد. پیش از اجرای کنترلر ODL، باید

متغیر محیطی `JAVA_HOME` را تنظیم کنید. ارزش کنونی `JAVA_HOME` را می‌توانید با فرمان `echo $JAVA_HOME` مشاهده نمایید. این ارزش احتمالاً تعریف نشده و خالی است. از متغیر محیطی `JAVA_HOME`، خروجی گرفته و می‌توانید آن را در بخش `.bashrc` بتوانید تا در هنگام `Reboot` شدن سیستم عامل یا `Login`, ثابت بوده و حذف نشود. متغیر `JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64` خود اضافه کنید یا برای یک بار آن را بصورت زیر تنظیم نمایید:

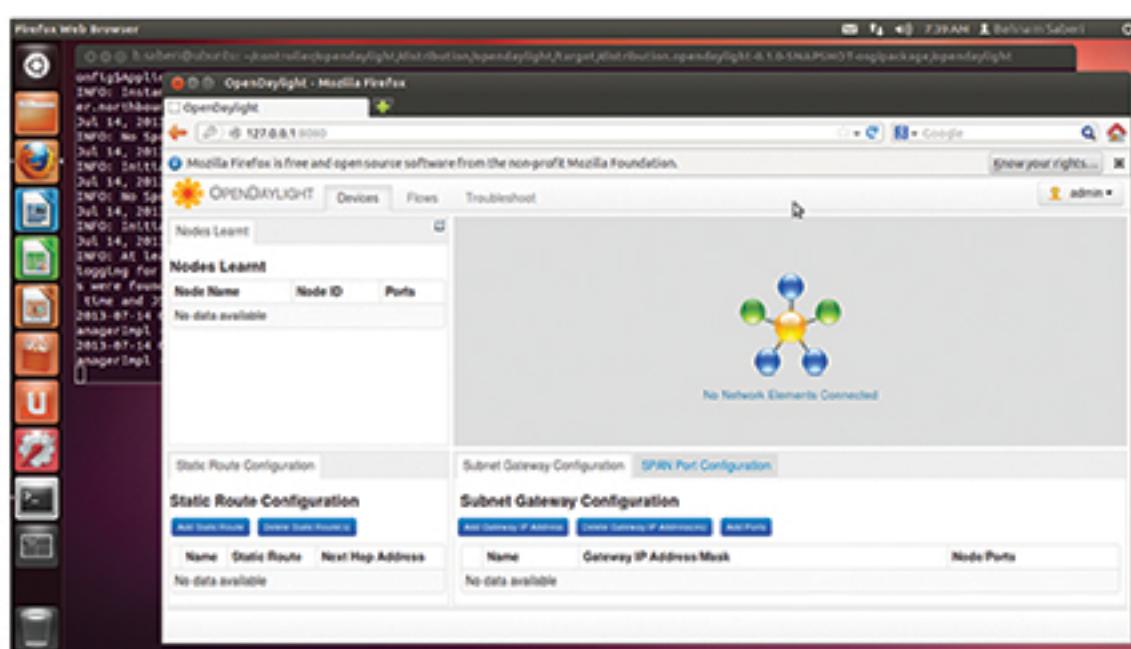
```
$ export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-i386 (or -  
amda64)
```

می‌توانید کنترل ODL را با تغییر دایرکتوری کنونی به مکانی که کدهای باینری ODL در آنجا قرار دارند، راه اندازی کنید. کنترل با فرمان `run.sh` شروع پکار می‌کند:

```
$ cd  
~/controller/opendaylight/distribution/opendaylight/target/distri-  
bution.opendaylight-0.1.0-SNAPSHOT-osgipackage/opendaylight  
$ ./run.sh
```

کنترل ODL نیاز به چند دقیقه زمان دارد تا تمام مأژولها را بارگذاری کند. می‌توانید مروارید خود را روی آدرس 127.0.0.1:8080 تنظیم کنید تا اینترفیس وب کنترل ODL را مشاهده کنید (تصویر زیر را مشاهده کنید). نام کاربری و رمز عبور پیش فرض admin است.

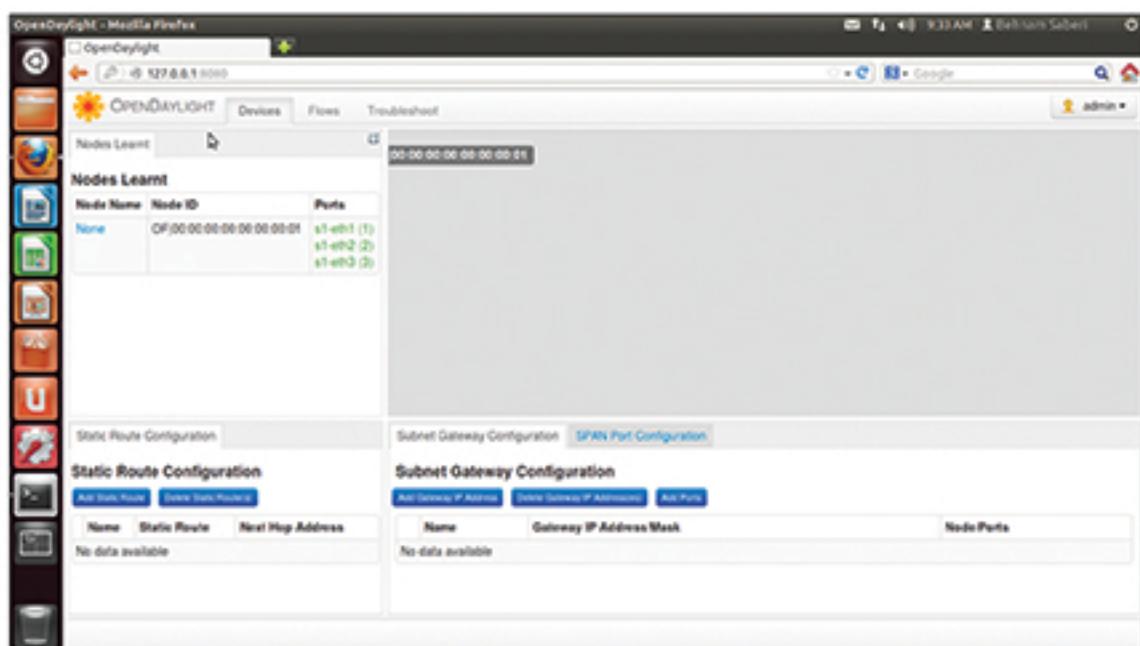
شکل F-۸
محیط گرافیکی
کنترلر OLD



اکنون که کنترل ODL را آماده و در حال اجرا داریم، می‌توانیم آن را به سوئیچ OpenFlow که در آزمایشگاه OpenFlow خود داریم، متصل کنیم. کنترلر ODL همراه با ماشین مجازی Mininet مورد آزمایش قرار گرفته است. به ماشین مجازی Mininet متصل شوید. آدرس IP سروری را که میزبان کنترلر ODL است برای شروع شبکه مجازی خود، در خط فرمان زیر استفاده کنید: (می‌توانید با فرمان `ifconfig -a` آدرس IP سرور را به دست آورید)

```
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=controller-ip --topo single,3
```

به کنترلر OpenDaylight متصل شده و یک سوئیچ و سه میزبان مجازی را که به آن متصل هستند، راهاندازی می‌کند. به تصویر زیر توجه کنید:

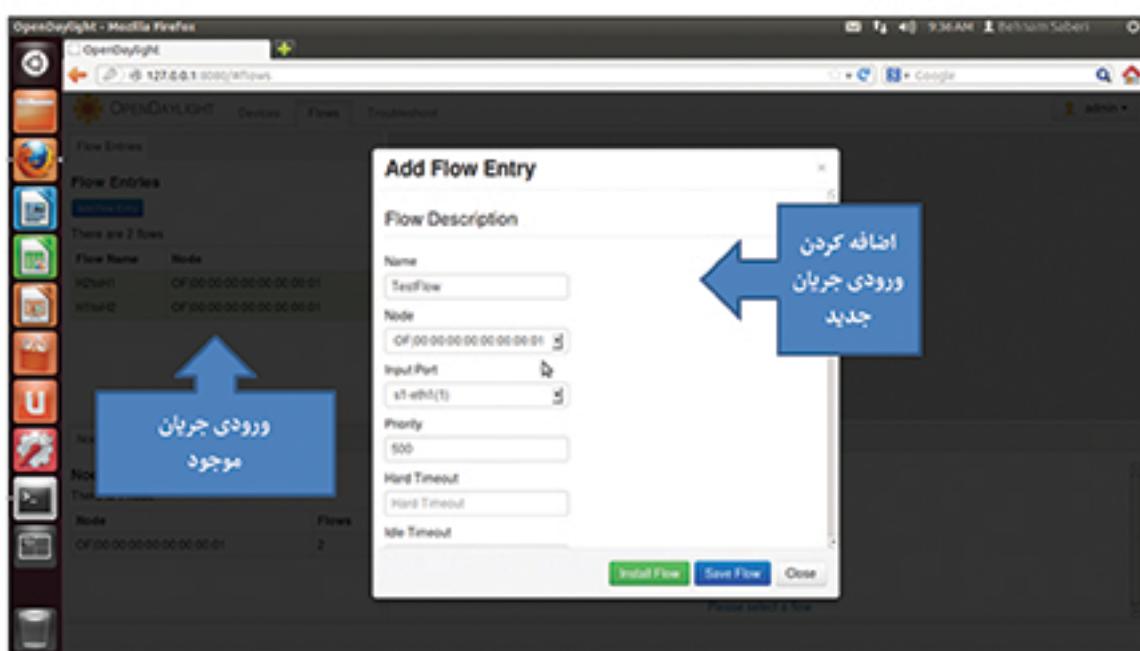


شکل ۴-۶
محیط ODL پس از راهاندازی شبکه‌ای Mininet در

زمانی که در کنترلر OpenDaylight، اشاره‌گر ماوس را روی سوئیچ OpenFlow مورد استفاده در آزمایشگاه خود ببرید، یک Pop Up ظاهر می‌شود که نشان می‌دهد، دستگاه منتظر پیکربندی است. شناسه Data Path ID یا بعبارتی DataPath ID، یک شناسه منحصر بفرد است که از آدرس MAC سوئیچ و یک شناسه تعیین شده توسط کنترلر، ساخته شده است. این شناسه در Mininet از ساختاری تشکیل شده که کلیه اعداد شناسه آن صفر بوده و تنها رقم انتهایی آن ۱ است (شکل بعدی). OpenFlow پروتکل LLDP را برای ردیابی توپولوژی (Topology Discovery) با استفاده از پیغامهای packet_out، بکار می‌برد؛ دستورالعملی که در آن کنترلر به عنصر ارسال کننده یعنی سوئیچ اعلام می‌کند، فعالیتی همچون ارسال پیغام ردیابی LLDP انجام دهد. سپس برای پیغامهای اصلاح جریان (FlowMod) یک

Action را مشخص می‌کند. تصویر بعدی بخشی از یک فرم مبتنی بر وب را نشان می‌دهد که پارامترهای مورد نیاز برای یک ورودی جریان را جمع آوری می‌کند. این ورودی جریان را می‌توان در جدول جریان سوئیچهای OpenFlow نصب نمود. در اینجا پورت خروجی بر روی سوئیچ را انتخاب می‌کنیم. به یاد داشته باشید، OpenFlow تنها آنچه را که دستور می‌دهید ارسال می‌کند، بنابراین یا قوانینی اضافه کنید که بتوان با ترافیک EtherType با ارزش 0x0806 برای درخواستهای Broadcast مربوط به ARP و پاسخهای Unicast بکار رود، یا در زمانی که یک پیغام اصلاح جریان اضافه می‌کنید، ارزش پیش فرض 0x0800 (منظور IPv4) برای فیلد EtherType را حذف کنید. شما باید ورودی جریانی را تعریف کنید که براساس آن، ترافیک های پورت ۱ سوئیچ را هدف قرار داده و Action ایجاد کند که این ترافیک داده را به پورت ۲ سوئیچ ارسال کند. همچنین ورودی جریانی تعریف کنید که با ترافیکهای داده روی پورت ۲ سوئیچ منطبق شده و با استفاده از Action مناسب، آنها را به سمت پورت ۱ سوئیچ ارسال کند. همچنین می‌توانید از پورت‌های رزرو شده Normal، Controller، Flood و تمام Drop دیگری که در بخش مشخصه‌های OpenFlow نسخه ۱.۰، فهرست شده‌اند، در منوی Down List استفاده کنید. Action‌هایی را انتخاب کنید که می‌توانند منطقی یا فیزیکی باشند. رویکرد منطقی باید با نشانه‌های سمبولیک نامگذاری شود، در حالی که رویکرد فیزیکی عددی است. پورت‌ها توسط سوئیچی که اطلاعات پیکربندی را ارسال می‌کند، باد گرفته می‌شوند. همچنین اگر یک پورت یا لینک غیر فعال شود، جدول جریان بروز می‌گردد. با اضافه کردن ورودیهای جریان مناسب در جدول جریان سوئیچ S1، می‌توانید یک مسیر بین میزبانها ایجاد کرده و آن را از طریق فرمان Ping در Mininet بررسی کنید. برای اشکال زدایی و بررسی بسته‌های عبوری، می‌توانید از ابزار dpctl یا Wireshark استفاده کنید، که پیشتر در این فصل و همچنین در فصل ۲ مورد بررسی قرار گرفته شدند.

شکل ۴-۷
منوی ایجاد یک
ورودی جریان جدید



خلاصه

در این فصل توضیحات مفصلی را در خصوص آزمایشگاه OpenFlow مبتنی بر Mininet و نقش آن به عنوان ابزاری برای مدل کردن شبکه‌ای که می‌تواند رابطی برای کنترلرهای راه دور (همچون POX) باشد، ارائه کردیم. این تنظیمات در راستای توسعه زیرساختی به حساب می‌آیند که بر اساس آن می‌توانیم در فصل بعد، API شماли متعلق به کنترلرهای OpenFlow (به طور مثال POX) را، برای توسعه Net App‌ها مورد استفاده قرار دهیم. علاوه بر این ما پروژه OpenDaylight و کنترلر از نوع Bootstrap آن را (منظور کنترلر ODL) معرفی کردیم که می‌تواند به عنوان یک کنترلر SDN برای توسعه محیط SDN بکار رود. کنترلر ODL و پورت شماли آن، که به Mininet مرتبط شده است، محیط مطلوبی را در اختیار ما قرار می‌دهد که در فصل آتی آن را برای توسعه Net App نمونه خود، مورد استفاده قرار خواهیم داد.