

مقدمه‌ای بر شبکه‌های نرم افزار بنیان

الهه جلال پور
سپهر کاظمیان

مقدمه

چرا SDN ؟

اجزا تشکیل دهنده SDN

ابزارهای توسعه SDN

فایروال SDWall

مقدمه

- ▶ شبکه‌های کامپیوتری به طور معمول از تعداد زیادی از دستگاه‌های شبکه‌ای مانند روترها، سوئیچ‌ها و انواع متعددی از middlebox ها ساخته شده‌اند.
- ▶ وظیفه‌ی بازرسی ترافیک و یا انتقال آن، به دست middlebox ها می‌باشد.
- ▶ اپراتورهای شبکه باید با استفاده شرایط شبکه با پیکربندی‌ها و سیاست‌هایی که در نظر می‌گیرند اتفاقات شبکه را کنترل و برنامه ریزی کنند.
- ▶ اپراتور شبکه مسئول پیاده‌سازی این سیاست‌ها و پیکربندی‌هاست و باید به صورت دستی سیاست‌های سازمان را به دستورات پیکربندی تبدیل کند.

چرا SDN؟

- ▶ ایده شبکه‌های قابل برنامه ریزی، به عنوان راه حلی برای توسعه آسان‌تر شبکه ارائه شد.
- ▶ وظیفه سخت افزار فقط حمل و نقل!!!!
- ▶ مدیریت شبکه ساده میشود
- ▶ با برنامه نویسی سیاست های خود را در شبکه پیاده سازی
- ▶ با کنترلر: مغز متفکر
- ▶ سطح داده: حمل و نقل ساده بسته ها

مقدمه

چرا SDN ؟

اجزا تشکیل دهنده SDN

ابزارهای توسعه SDN

فایروال SDWall

اجزا تشکیل دهنده SDN

▶ پروتکل ForCES

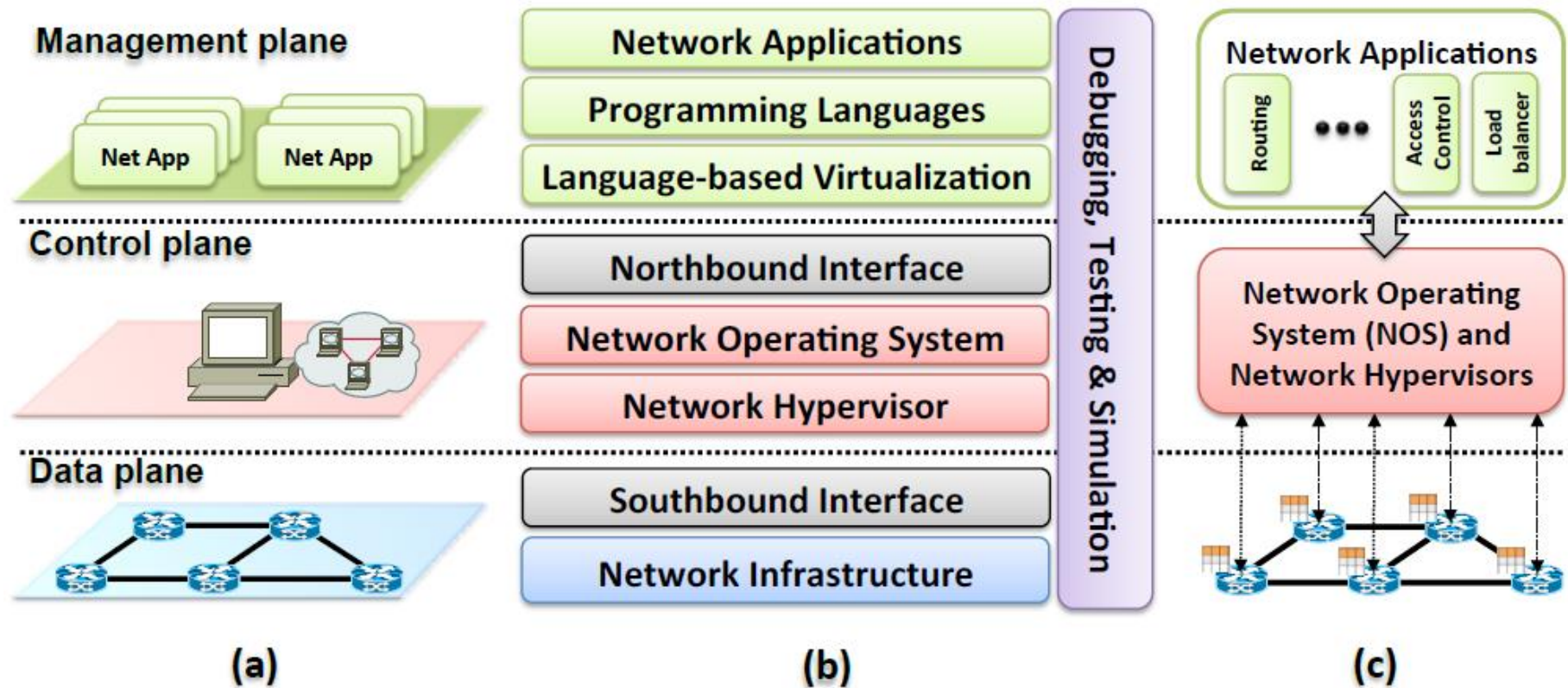
▶ ترکیب دستگاه حمل و نقل جدید با کنترلرهای third-party در یک دستگاه شبکه

▶ پروتکل OpenFlow

▶ سطح کنترل به طور کامل از سخت افزارهای شبکه جدا می باشد.

پروتکل OPENFLOW

- ▶ بعد از معماری‌های مختلفی همچون ActiveNetworking، OpenSignaling، DCAN و 4D Project نهایتاً پروتکل OpenFlow مطرح شد.
- ▶ در معماری OpenFlow دستگاه حمل و نقل یا سوئیچ OpenFlow، شامل یک یا چند جدول جریان (Flow Table) و یک لایه انتزاعی است که برای ارتباط ایمن این دستگاه با کنترلر از پروتکل OpenFlow استفاده شده است.



پروتکل OPENFLOW

► عناصر هر Flow

► ۱. فیلدهای تطابق

► هدر بسته ها و یا metadata ها و یا پورت ورودی

► آدرس مبدا + آدرس مقصد + نوع پروتکل و ...

► ۲. شمارنده ها: جمع آوری آمار برای جریان Flow خاص

► تعداد بسته های دریافتی

► تعداد بایت و مدت زمان وجود جریان

پروتکل OPENFLOW

► عناصر هر Flow

► ۳. مجموعه ای از دستورالعمل ها و اقدامات

► مطابقت داره؟ خب عكس العمل x را پیاده سازی كن

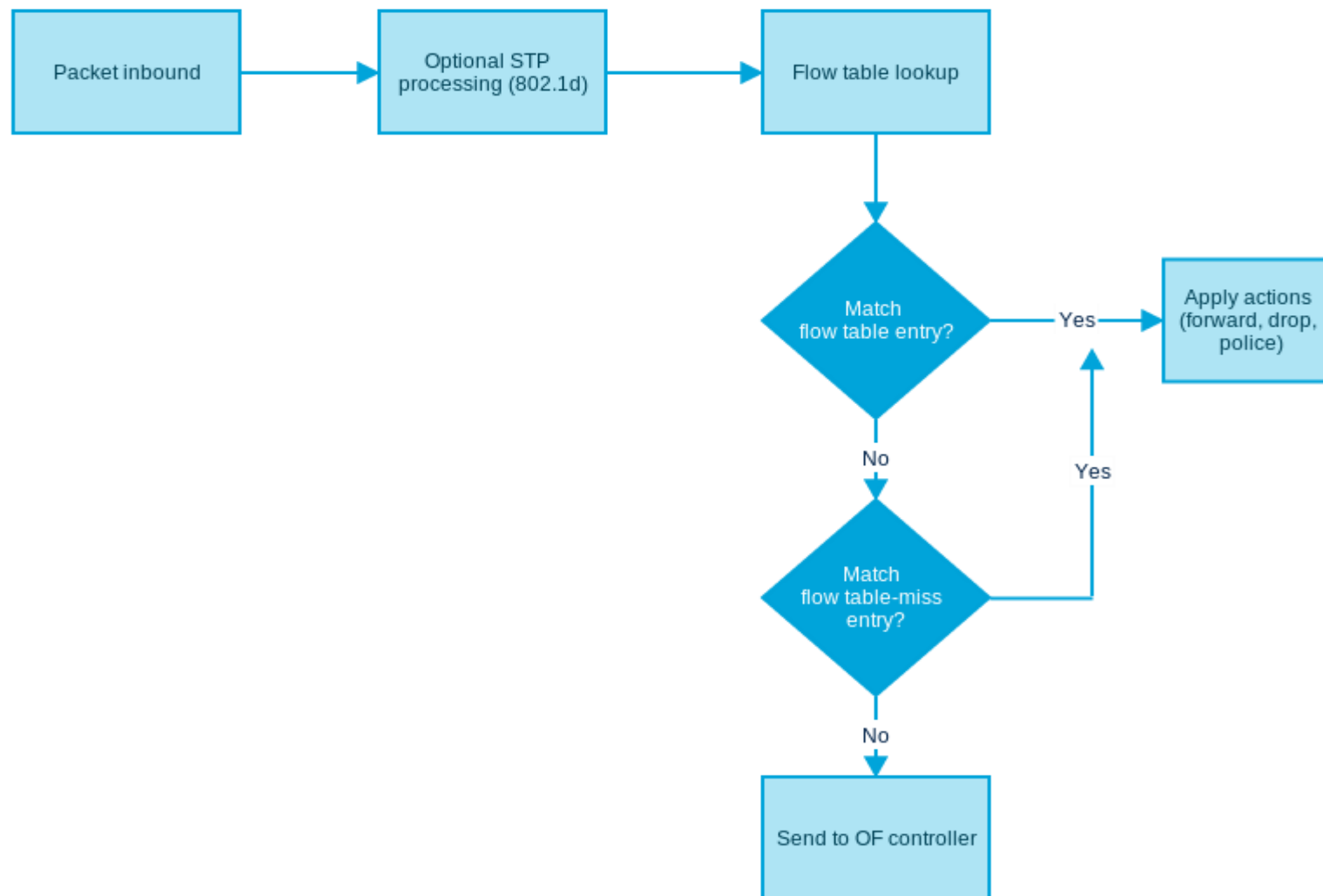
► مطابقت نداره؟؟؟؟

► Table miss

► عكس العمل های y را پیاده سازی كن ...

پروتکل OPENFLOW

- ▶ عکس العمل های y :
- ▶ حالت سنتی:
- ▶ مثلاً حذف بسته ورودی
- ▶ ادامه عمل تطابق روی جدول و ...
- ▶ نسخه های جدید:
- ▶ سویچ های ترکیبی
- ▶ هر دو پورت OF و غیر OF را پشتیبانی میکنند.
- ▶ کنترل از راه دور:
- ▶ اضافه یا حذف کردن از جدول سوئیچ ها و آپدیت آن
- ▶ یا اتوماتیک و proactively
- ▶ یا به صورت passively یا همان واکنشی



تجهیزات حمل و نقل FORWARDING ELEMENTS

► در SDN اغلب این تجهیزات حمل و نقل، برای حمل و نقل پایه‌ای در سطح فیزیکی مورد استفاده قرار می‌گیرند، که این تجهیزات دارای قابلیت OpenFlow هستند و از این طریق با کنترلر در ارتباط هستند.

کنترلر CONTROLLER

► یک کنترل کننده SDN به سوئیچها میگوید که چه عملی را باید انجام دهند.

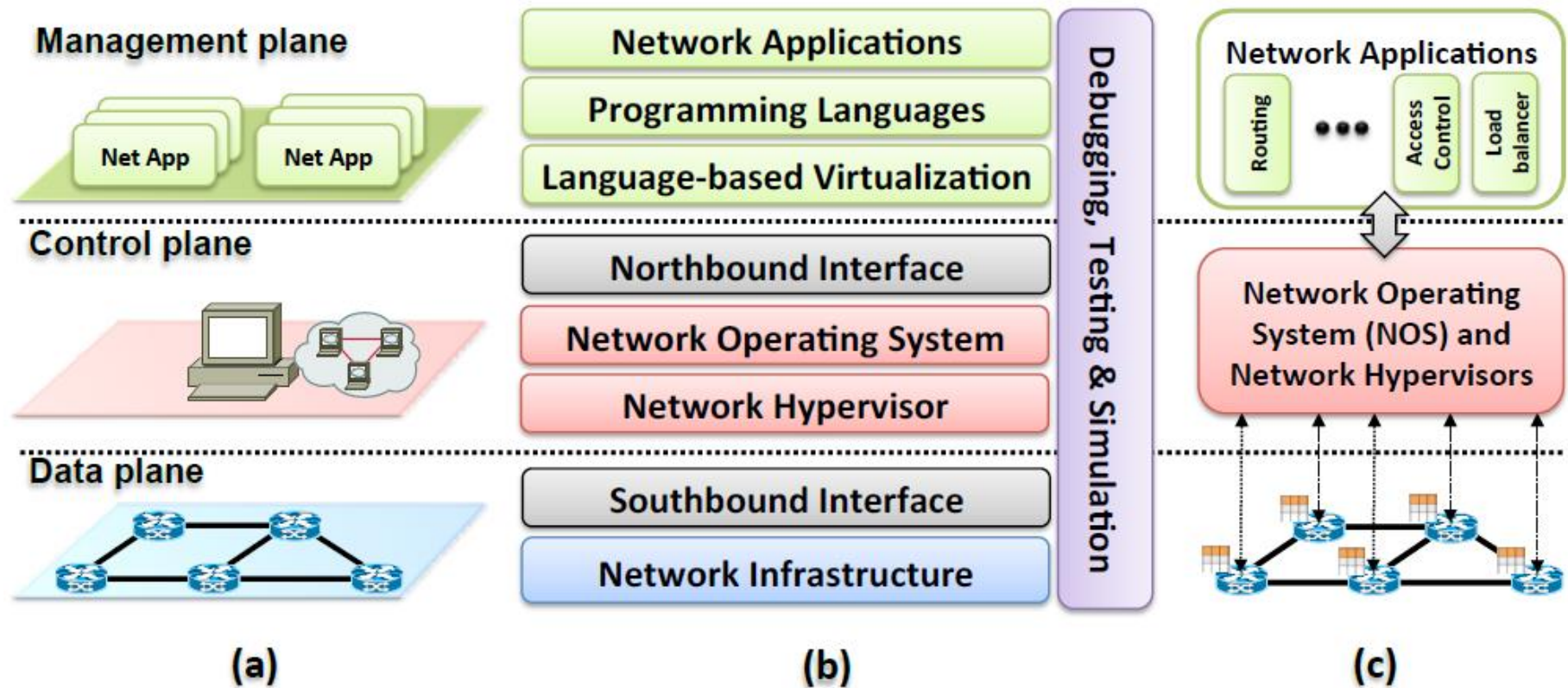
► Southbound API یا واسط جنوبی

► برای ارتباط کنترلر با لایه زیرساخت (سوئیچها) استفاده می شود به عنوان مثال پروتکل OpenFlow به عنوان واسط جنوبی تلقی می شود.

► این واسط جنوبی اغلب مورد توجه توسعه دهندگان OpenFlow و شرکتهای سازنده قطعات است، زیرا مستقیم با سخت افزار ارتباط برقرار می کند.

► Northbound API یا واسط شمالی

► برای ارتباط کنترلر با لایه Application از این API استفاده می شود.

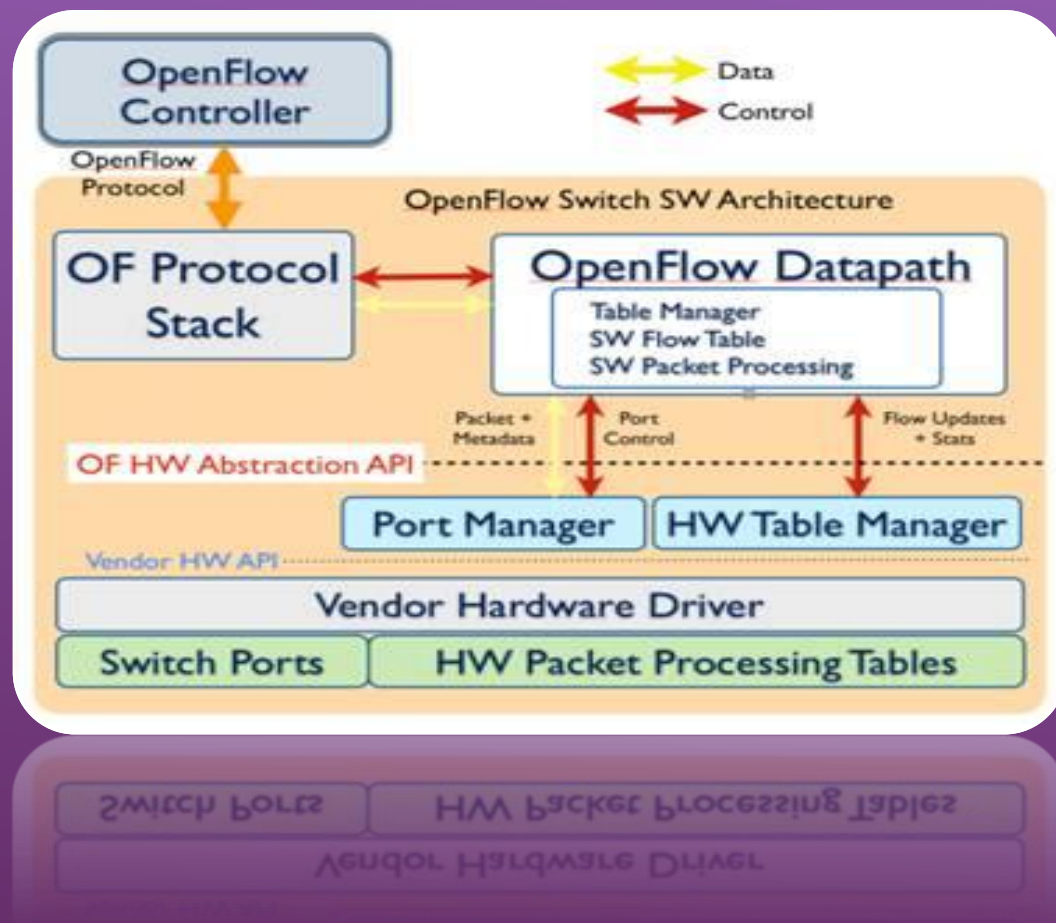


کنترلر CONTROLLER

▶ کنترلر ها:

- RYU (Python-based)
- OpenDayLight (Java-based)
- Floodlight (Java-based)

OPENFLOW ARCHITECTURE



MININET

► **Mininet**: از این ابزار برای شبیه سازی یک شبکه SDN استفاده می شود. این پلتفرم تعدادی سوئیچ، هاست و کنترلر را شبیه سازی میکند و برای کنترل میبایست این پلتفرم به یک کنترلر از سری کنترلرهای موجود متصل گردد.

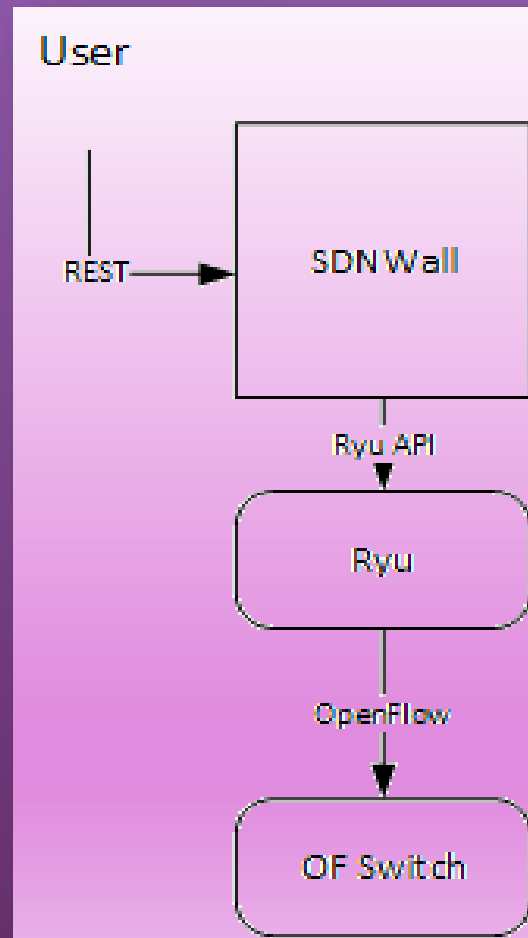
RYU

► **Ryu**: این کنترلر با زبان پایتون پیاده سازی شده است و تا به امروز تا نسخه ۱.۴ از پروتکل OpenFlow را به صورت کامل پشتیبانی می کند. برای استفاده از این کنترلر کافی است که برنامه مورد نظر را با استفاده از کتابخانه مخصوص این کنترلر به زبان پایتون نوشته و روی کنترلر اجرا کنیم.

فایروال SDWALL

- ▶ در گذشته از نودهای سخت افزاری خاص منظوره یا سوئیچ‌هایی که قابلیت فایروال را داشتند تحت عنوان فایروال در شبکه استفاده می‌شد. مشکل اصلی روابط کاربری این قطعات بودند که کاملاً به سازنده آن‌ها وابسته بود و قابلیت اعمال پیکربندی روی آن‌ها از طریق یک رابط یکسان وجود نداشت.
- ▶ در پروژه SDWall ما با پیاده سازی فایروال روی کنترلر SDN قابلیت اعمال پیکربندی از طریق رابط کاربری یکسان REST برای کل شبکه را فراهم آوردیم.

معماری فایروال SDWALL



فایروال SDWALL

► برای اجرای این فایروال نیاز است که مراحل زیر طی شود:

► نصب پایتون نسخه ۳

► نصب پلتفرم EL Ryu

► اجرای SDWall روی پلتفرم EL Ryu و برقراری ارتباط با سوئیچ‌های OpenFlow

► کنترل SDWall از طریق REST API آن به وسیله‌ی یک کلاینت دلخواه

فایروال SDWALL

► نصب پایتون ۳ بر روی یک لینکوس از توزیع‌های Debian

```
% sudo apt-get install python3
```

فایروال SDWALL

► نصب پلتفرم ELRyu

```
% git clone git://github.com/elahejalalpour/elryu.git  
% cd elryu; python3 ./setup.py install
```

فایروال SDWALL

► اجرای SDWall روی پلفرم ELRyu و برقراری ارتباط با سوئیچ‌های OpenFlow

► در این مرحله ما برای شبیه سازی سوئیچ‌های OF از mininet استفاده می‌کنیم.

```
% ryu-manager sdn101/app/rest_firewall.py  
% sudo mn --switch=ovs,protocol=OpenFlow13 --  
controller=remote,x.x.x.x
```

فایروال SDWALL

► کنترل SDWall از طریق REST API آن به وسیله‌ی یک کلاینت دلخواه

```
% python3 sdn101/firewall_client_cli.py
```




با تشکر از زحمات استاد محترم **دکتر مازوچی**، عضو
هیأت علمی پژوهشگاه ارتباطات و فناوری اطلاعات و استاد
دوره‌ی کارآموزی